

9/3

80% Asistencia a la práctica.

1 parcial y 1 TP

Final

22 de junio, termina la materia.

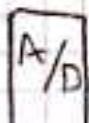
* Práctica 1, introducción.

- Sistema: Objetos que interactúan de forma causal.

- Sistema analógico: Definidos para todo tiempo.

Las señales digitales son muestras de señales analógicas.

Convertidor analógico digital



En el sistema digital existe el error de cuantificación.



Formato mp3 y WAV.

• Compresión:

① Sin pérdidas; ② Con pérdidas.

No pierde información.
pierde información.

• Conceptos eléctricos:

$$P = V \cdot I = \frac{V^2}{R_1 + R_2} = I \cdot \frac{V^2}{R_1 + R_2}$$
$$I = \frac{V}{R_1 + R_2}$$

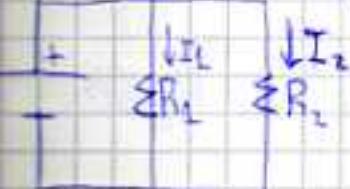
V: Diferencia de potencial (Volts)

: Potencia (Watts)

: Corriente (Amperes)

: Resistencia (Ohms)

Resistencias en paralelo.



$$I = I_1 + I_2$$

$$I_1 = \frac{V}{R_1} \quad I_2 = \frac{V}{R_2}$$

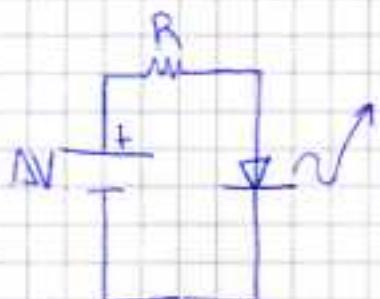
$$I = V \left(\frac{1}{R_1} + \frac{1}{R_2} \right)$$

*LED:

Light

Electroluminescent

Diodo



Suponiendo resistencia del led insignificante, luego:

$I = \frac{V - V_d}{R}$; $V_d = 2,1V \rightarrow$ Se llama V_d a tensión de barrera porque prende el led. Hay que superarla para que prenda.

¶

Transistores:

① Amplificador de tensión.

② Llave portadora o no polar corriente. \rightarrow Vamos a darle este uso.



I) Imprimir prácticas.

1/3 Teórico

Profesor: Jorge Fuchs

Facundo Cataño

• Periodo, Semana 10: Horario de la Teoría:
TP, Circuitos Funcionales; informe.

→ 1º recuperatorio, 2º la práctica
2º recuperatorio, luego de
terminada cursada.

Calculo y examen oral.

Introducción.

Sistemas Numéricos.

Álgebra de Boole.

Códigos.

Circuitos Combinacionales.

Multiplexores y Decodificadores.

Unidades Aritméticas.

PLD - Memorias

Funciones Lógicas

Circuitos Secuenciales

Flip - Flops.

Contadores Síncronos

Contadores Asíncronos

Registros

Transistor y Diodo diodo origen a la electrónica analógica.

- Análogo: Sistema=Combinación de tiempo.

Transistor saturado, abierto con. llave abierta, Transistor cortado o clivagante. Puerta cerrada.

- Digital: Toma valores discretos en el tiempo.

El transistor amplia las señales.

* Circuito Combinacional: La respuesta es función en combinaciones de las variables de entrada, únicamente.

* Circuito Secuencial : La respuesta depende de las Variables de entrada y los estados anteriores de la respuesta anterior.



Sistemas Numéricos:

* Sistemas Numéricos: Permiten ~~representar~~^{otrar} las magnitudes.

Ejemplo s:

Decimal:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Octal:

[0, 1, 2, 3, 4, 5, 6, 7]

Hexadecimal:

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

Binario:

[0, 1]

Formados por Símbolos, y su posición

• Cómo escribir cualquier número en base de un sistema numérico.

$$N = \sum_{i=0}^n a_i \cdot b^i$$

↗ peso
 ↘ base
 ↘ Coeficientes

• Los coeficientes y las bases deben ser enteros.

Ejemplo:

$$458_{10} = 4 \cdot 10^2 + 5 \cdot 10^1 + 8 \cdot 10^0, \quad [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$$

• Cambio de base:

De base 10 a base 3:

$$3^x - 1 = 10^x - 1$$

$$x = \log_{10}(10)$$

$$x \approx 2,95 \Rightarrow \text{Se redondea para } x=3$$

n	m
b_1	b_2

Fórmula cambio de base, solo para número máximo.

A mayor base, menor cantidad de dígitos

Ejemplo:

$$158_{10} \rightarrow 158_2 :$$

$$\begin{array}{r}
 158 \\
 \hline
 10) 158 \\
 10) 15 \\
 10) 5 \\
 \hline
 1
 \end{array}
 \Rightarrow 158_{10} = 1001110_2$$

$$A^9 D^2 5 \rightarrow]$$

$\frac{1}{16} \quad \frac{1}{10}$

$$\underbrace{10 \cdot 16^4} + \underbrace{9 \cdot 16^3} + \underbrace{13 \cdot 16^2} + \underbrace{2 \cdot 16^1} + \underbrace{5 \cdot 16^0} = \\ A \quad 9 \quad D \quad 2 \quad 5$$

$$\begin{array}{r} 1011,101 \\ \times 2 \\ \hline 2022,22 \end{array}$$

Cuando la base aumenta, la cantidad de dígitos es menor. Además, los bases se felicitaban como:

$$\begin{array}{r} 4-2 \\ \times 3 \\ \hline 9-2 \\ 16-2^4 \\ 16-4^2 \\ 9-3^2 \end{array} \rightarrow \text{De base } 2 \text{ a base } 4, \text{ cada número tiene el doble de dígitos}$$

SUMA

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	10
2	2	3	4	10	11
3	3	4	10	11	12
4	4	10	11	12	13

PRODUCTO

*	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	10	13
3	0	3	11	14	22
4	0	4	15	22	31

$$FG\ 75 + 1209 = HOTE, \text{Base } H+1=18$$

$$6+7=13, \text{Base } 12$$

$$5+7=13, \text{Base } 9$$

16/3

HUJA N°

FECHA

~~• Sistema decimal:~~

- Sistemas posicionales
 - Decimal (0,1,2,3,4,5,6,7,8,9)
 - Muyas
- Sistemas no posicionales
 - Romanos I V X C CDM

~~• Diferencias entre binario y decimal.~~

Binarios {⁰₁}

Son simples, para sistemas digitales.

Hexa decimal: [0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F]

Octal: [0,1,2,3,4,5,6,7]

0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

El número más grande que se puede representar es:

$$7 = 2^3 - 1$$

Dígitos usados.

Bási mayor, cantidad de dígitos menor.

Cantidad de dígitos mayor, Básis menor.

Ejercicios de cambio de sistema:

45₁₀ → binario

método:

$$\begin{array}{r}
 45 \\
 \boxed{1} \quad \boxed{0} \quad \boxed{1} \quad \boxed{0} \quad \boxed{1} \\
 \boxed{1} \quad \boxed{1} \quad \boxed{0} \quad \boxed{1} \quad \boxed{0} \\
 \hline
 0 & 1 & 1 & 0 & 1 \\
 \boxed{1} & \boxed{1} & \boxed{0} & \boxed{1} & \boxed{0} \\
 \hline
 1 & 1 & 0 & 1 & 0 \\
 \boxed{1} & \boxed{0} & \boxed{1} & \boxed{0} & \boxed{1} \\
 \hline
 1 & 0 & 1 & 1 & 0
 \end{array}$$

$\Rightarrow 45_{10} = 101101_2$

\rightarrow escribir

restos:

método:

$$32 \ 16 \ 8 \ 4 \ 2 \ 1$$

$$\rightarrow 45 \geq 32 \Rightarrow 45 - 32 = 13$$

$$32 \ 16 \ 8 \ 4 \ 2 \ 1$$

$$1 \quad | \quad \rightarrow 13 \leq 16 \Rightarrow \text{No. } 0$$

$$32 \ 16 \ 8 \ 4 \ 2 \ 1$$

$$1 \quad 0 \quad | \quad \rightarrow 13 \geq 8 \Rightarrow 13 - 8 = 5$$

$$32 \ 16 \ 8 \ 4 \ 2 \ 1$$

$$1 \quad 0 \quad 1 \quad | \quad \rightarrow 5 \geq 4 \Rightarrow 5 - 4 = 1$$

$$32 \ 16 \ 8 \ 4 \ 2 \ 1$$

$$1 \quad 0 \quad 1 \quad 1 \quad | \quad \rightarrow 1 \geq 1 \Rightarrow 1 - 1 = 0$$

$$32 \ 16 \ 8 \ 4 \ 2 \ 1$$

$$1 \quad 0 \quad 1 \quad 1 \quad 0 \quad | \quad \rightarrow 1 = 1 \Rightarrow 1 - 1 = 0$$

$$32 \ 16 \ 8 \ 4 \ 2 \ 1$$

$$1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1$$

• Otro ejemplo: $\begin{array}{r} 524287 \\ \times 892346 \\ \hline \end{array}$

524287
1

$$\begin{array}{r} 262144 & 131072 & 65536 & 32768 & 16384 & 8192 & 4096 & 2048 & 1024 & 512 & 256 \\ \text{①} & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ \hline \end{array}$$

$$\begin{array}{r} 524287 \\ \times 892346 \\ \hline 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ \hline \end{array}$$

$524287 \times 892346 =$

$$\begin{array}{r} 892346 \\ 524287 \\ \hline 367051 \\ -262144 \\ \hline 105114 \\ -65536 \\ \hline 40579 \\ -32768 \\ \hline 7610 \\ -4096 \\ \hline 3514 \\ -2048 \\ \hline 1466 \\ -1024 \\ \hline 442 \\ -312 \\ \hline 30 \\ -256 \\ \hline \end{array}$$

$$\begin{array}{r} 892346 \\ \text{②} \\ \hline \end{array} =$$

• OEP:

$$\begin{array}{r} 435, 593 \\ \text{③} \\ \hline \end{array} \rightarrow \text{binario}$$

$$M = \sum_{i=0}^N C_i b^i + \sum_{d=1}^P C_d \cdot N! b^d$$

④ Se trata la parte entera:

$$\begin{array}{r} 435 \\ 477 \\ 477 \\ \hline 172 \\ 172 \\ \hline 0 \\ \text{NOTA: } 0 \\ \hline 593 \\ 593 \\ \hline 0 \\ \text{NOTA: } 0 \\ \hline 2 \\ 2 \\ \hline \end{array}$$

$$\begin{array}{r} 435 \\ 477 \\ 477 \\ \hline 172 \\ 172 \\ \hline 0 \\ \text{NOTA: } 0 \\ \hline 2 \\ 2 \\ \hline \end{array} \Rightarrow \frac{435}{477} = 1 \frac{1011011}{477}$$

• Extrae la parte decimal:

$$0,543 \times 2 = 1,086$$

$$0,086 \times 2 = 0,172$$

$$0,172 \times 2 = 0,344$$

$$0,344 \times 2 = 0,688$$

$$0,688 \times 2 = 1,376$$

⋮

• Aquí se multiplicó por 2 la parte decimal
número.

Se puede seguir indefinidamente, pero se corta.

$$435,343 = 110110011,10001$$

↓
10

Error:

$$E < 2^{-5}$$

$$\begin{array}{r} 4512 \\ \hline 7 \end{array}$$

$$N = 2 \cdot 7^0 + 1 \cdot 7^1 + 5 \cdot 7^2 + 1 \cdot 7^3 + 2 \cdot 7^4$$

R3

Bases que son potencias de otras bases:

Ejemplo:

$$\begin{array}{r} 100012103,20011212 \\ \hline 10 \quad 7 \quad 2 \end{array} = \cancel{104359}$$

$9 = 3^2$ → Dado: 3 a 9, cada número tiene 9 dígitos en base 3.

Otro ejemplo:

$$\begin{array}{r} 666 \\ \times 2 \\ \hline 1332 \end{array}$$

$$2^3 = 8$$

$$\begin{array}{r} 4 & 2 & 1 \\ \times & 2 & 0 \\ \hline 1 & 0 & 0 & 0 \end{array}$$

$$1110 = 110110110$$

Pasando a Hexadecimal:

$$\begin{array}{r} 110110110 \\ \hline 1 \quad B \quad 6 \end{array}$$

Otro ejemplo:

$$\begin{array}{r} ABCDEF \\ \downarrow \quad \downarrow \quad \downarrow \\ 101010111001101, 1101111 \end{array}$$

• Tabla de suma:

$$\begin{array}{c|cc} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array} \quad \begin{array}{c|cc} \times & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}$$

Ej 12:

$$FG75 + H209 = H07E$$

$$B35 = 19$$

* COMPLEMENTO DE UN NÚMERO:

- Representación de signo y módulo: Para d. representar los negativos de los positivos, si EMPIEZA CON 1 ES NEGATIVO, SI EMPIEZA CON 0 ES POSITIVO.

~~Representación complemento al módulo:~~

+0	0	0	0	0	0
+1	0	0	1	0	0
+2	0	1	0	0	0
+3	0	1	1	0	0
+4	1	0	0	0	0
+5	1	0	0	1	0
+6	1	1	0	0	0
+7	1	1	1	0	0
+8	1	1	1	1	0
+9	1	1	1	1	1

- Representación de complemento al módulo:

0	0	0	0
-1	1	1	0
-2	1	1	0
-3	1	0	1
-4	1	0	0

- Representación de Complemento al módulo menor 1:

$$C_{M-1}(001) = 110$$

Número más negativo decimal base:

$$-b^{n-1}$$

* Operaciones de Físicas:

Ejemplo:

$$\begin{array}{r}
 \text{carry} \quad \text{out carry} \\
 \begin{array}{r}
 \text{0} \quad \text{1} \quad \text{1} \quad \text{1} \quad (7) \\
 + \quad \text{1} \quad \text{0} \quad \text{0} \quad \text{1} \quad (4) \\
 \hline
 \text{1} \quad \text{0} \quad \text{0} \quad \text{0} \quad (11)
 \end{array}
 \end{array}$$

- Carry: Cuando la suma lleva de paso, ese dígito que sobra se llama Carry, se escribe C. (Para números todos positivos).

- Overflow: Cuando el carry in \neq carry out .(V)

- Cero: Si todos los bits del resultado son 0. (\geq)

- Paridad: Si el nº los 1's son par o impar, el que presenta mayor cantidad de 1's manda la paridad. (P) $P=1$ par o $P=0$

- Signo: Indica el signo (N) $N=1$ negativo $N=0$ positivo.

18/3 TRABAJO

EGROUP YAHOO / TECNICA DIGITAL

- Sistema Numérico: Permite representar números.

00 01 1100 101100

Después de la coma también se refiere con coma, y para cambiar de base en binario que con potencias de otras bases, se cambia hacia cada lado a partir de la coma.

$$\begin{array}{r}
 + 0 1 1 \\
 0 0 1 \\
 \hline
 1 1 10
 \end{array}$$

$$\begin{array}{r}
 \times 0 1 \\
 0 0 0 \\
 \hline
 1 0 0 1
 \end{array}$$

Ejemplo de Suma:

$$\begin{array}{r}
 1 1 1 \\
 + 1 0 1 \\
 \hline
 1 1 0 1
 \end{array}$$

11000

• La computadora solo suma, no divide ni resta ni multiplica.

Para la multiplicación, la computadora hace lo siguiente: Cuenta los unos, pone el mismo número en este caso el 110. Si hay 0, pone todos 0.

$$\begin{array}{r}
 1 1 0 \\
 \times 1 0 1 \\
 \hline
 1 1 0 \\
 0 0 0 \\
 \hline
 1 1 1 1 0
 \end{array}$$

★ Complemento de un Número

Lo que le falta a un número para llegar a una potencia de dos.

Complemento(21) = 79 → Lo que a 21 le falta para llegar a 100.

$C_9(21) = 79$ → Lo que falta para llegar a 99

0 → 9

1 → 8

2 → 7

3 → 6

4 → 5

Números
Complementarios

El Complemento del
Complemento es el
mismo número.

• ¿Qué es un B.I.E?

Binary D.I.E

Bit

• ¿Cómo represento números negativos?

Número Positivo	Bit de Signo y Magnitud	Complemento a la base menos 1 C ₁	Complemento a la base C ₂
0000	0	+0	+0
0001	1	+1	+1
0010	2	+2	+2
0011	3	+3	+3
0100	4	+4	+4
0101	5	+5	+5
0110	6	+6	+6
0111	7	+7	+7
1000	8	-0	-8
1001	9	-1	-1
1010	10	-2	-6
1011	11	-3	-5
1100	12	-4	-4
1101	13	-5	-3
1110	14	-6	-2
1111	15	-7	-1

* Número

• Punto: Pueden representar 2^n números.

NO SIRVE PARA RESTAR

* Bit de signo y Magnitud:

Sólo sirve el primer Bit Valor de signo negativo o positivo. Se pueden representar 2^{n-1} números positivos y negativos. NO SIRVE PARA RESTAR

* Complemento a la base menor 1:

Representar a los números negativos por el complemento a la base -1.

SIRVE PARA RESTAR.

* Complemento a la base en binario 2:

Representar a los números negativos por el complemento a la base.

Complemento a la = [Complemento a la base -1] + 1 Puede representar de $-2^{(n-1)}$ hasta $2^{(n-1)} - 1$

SIRVE PARA RESTAR

Ejemplos:

$$\begin{array}{r} +3 \\ -2 \\ \hline 5 \end{array} \quad \rightarrow \quad \begin{array}{r} 001 \\ 001 \\ \hline 010 \end{array}$$

- Datos:
- $C=0$; Carry, es el primer dígito. A veces se suman números y no obtiene otro con más dígitos. En ese caso carry es 1, si no es 0.
 - $V=0$; Vale 1 cuando el resto de la suma es 0.
 - $S=0$; Indica el signo del número. 0 positivo, 1 negativo.
 - $Z=0$; Vale 1, si todos los valores son 0, si no vale 1.
 - $H=1$.
 - $P=0$; Contador de 1 y 0 impares inicia. Se pone el que tiene más.

• Ejemplo 4 y 5 con más números:

$$\begin{array}{r} +4 \\ -5 \\ \hline 9 \end{array} \quad \rightarrow \quad \begin{array}{r} 000100 \\ 000101 \\ \hline 001001 \end{array} \quad \text{No hay overflow!}$$

Resta:

$$C_1 \quad \begin{array}{r} -4 \\ -3 \\ \hline 1 \end{array}$$

$$\begin{array}{r} +0100 \\ -1100 \\ \hline 10000 \end{array}$$

$$C_t=1$$

$$V=0$$

$$S=0$$

$$Z=1$$

$$C_2 \quad \begin{array}{r} -0100 \\ -1101 \\ \hline 10001 \end{array}$$

$$C=1$$

$$V=0$$

$$S=0$$

$$Z=0$$

$$\begin{array}{r} 3 \\ -4 \\ \hline -1 \end{array}$$

C₁

$$\begin{array}{r} 1 & 1 & 1 \\ + 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 0 \end{array}$$

C = 0

V = 0

S = 1

Z = 0

- 1

C₂

$$\begin{array}{r} 0 & 0 & 1 & 1 \\ + 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 1 & 1 \end{array}$$

C = 0

V = 0

S = 1

Z = 0

- 1

$$\begin{array}{r} -4 \\ + -2 \\ \hline -6 \end{array}$$

C₁

$$\begin{array}{r} 1 & 1 & 1 & 1 \\ + 1 & 0 & 1 & 1 \\ \hline 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 \end{array}$$

C = 1

V = 0

S = 1

Z = 0

- 6

C₂

$$\begin{array}{r} 1 & 1 & 0 & 0 & 0 \\ + 1 & 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 \end{array}$$

C = 1

V = 0

S = 1

Z = 0

- 6

+	1	0	0	1
	1	0	1	1
	1	0	1	0

ERROR

$$C=1$$

$$V=1$$

$$S=0$$

$$Z=0$$

+	L	O	L	O
	L	1	O	O
	L	O	L	O

ERROR

$$C=1$$

$$V=1$$

$$S=0$$

$$Z=0$$

* Coma Flotante.

Se utilizan bits para expresar exponentes. Se encuentran en la página de la materia.

* Funciones Lógicas.

Si $Z = F(A, B, C)$:

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

• El circuito lógico binario es Combinacional.

• El circuito es el hecho con componentes que suman la composición real de las funciones lógicas utilizadas en circuitos.

* Algebra de Boole.

(1) Suma lógica: Equivalente al \oplus de lógica.

(2) Producto lógico: Equivalente al \wedge de lógica.

(3) Complemento

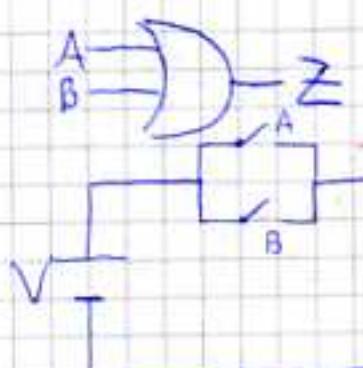
(4) Suma lógica:

$$Z = A + B$$

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

La Suma lógica es como el \vee de lógica.
 $A + B = 1 \Leftrightarrow "A o B" \text{ Valen } 1.$

Sesimboliza:



En inglés **OR**

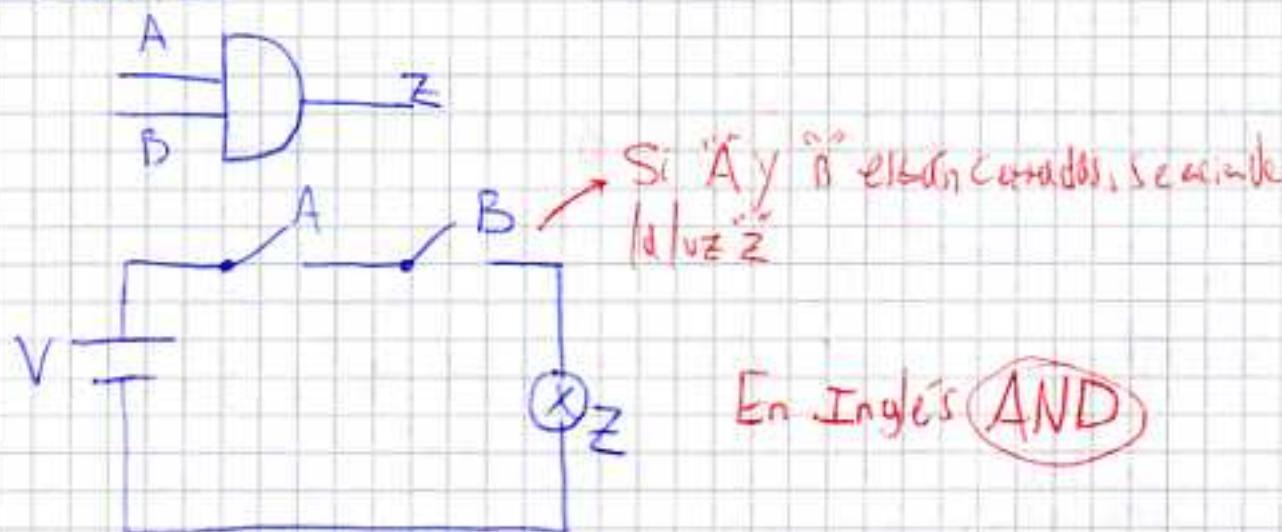
② Producto lógico:

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

$$Z = A \cdot B$$

$A \cdot B = 1 \Leftrightarrow A \text{ y } B \text{ valen } 1$

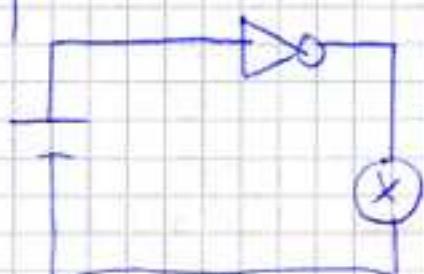
Se simboliza:



③ Complemento:

A	Z
0	1
1	0

$$\rightarrow Z = \bar{A}$$



Se simboliza $\neg A \rightarrow 0 - Z$

En Inglés (NOT)

* De estos 3 se derivan otros como:

• NOB: NOT, OR

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

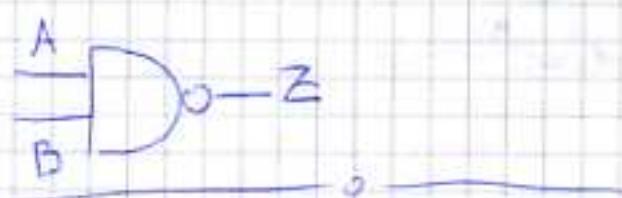
$$Z = \overline{A+B}$$



• NAND: NOT, AND

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

$$Z = \overline{AB}$$



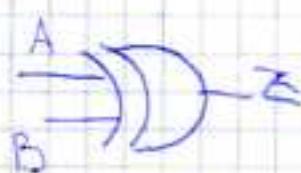
EXISTEN VARIACIONES DEL OR:

④ X-OR:

$$Z = A \oplus B$$

Síntesis de

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0



Un "o" exclusivo. Si los dos entradas son el mismo valor y no los dos al vez.

③ X-NOR:

$$Z = \overline{A \oplus B}$$



A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

/3

Biots = 127



$\frac{E}{32} - 0,15625 \rightarrow \text{Biots} = 127$

$$0,15625 \times 2 = 0,3125$$

$$0,3125 \times 2 = 0,625$$

$$0,625 \times 2 = 1,25$$

$$0,125 \times 2 = 0,25$$

$$0,125 \times 2 = 1$$

$$0,00101 \rightarrow 1,12^3$$

Abito y Biots:

$$1300 - 3111 = 124$$

Expresión Simple en coma Flotante IEEE - 32 bits

Signo Exponente

Mantisa

31 30

1 bit de Signo

43 22

Lubre 10

Alexander

SUMA 127

* ~~Algebra de Boole.~~

• Huntington:

1. $\exists K$ elementos, relación de eq =

2. Reglas de combinación $+$, \cdot , $\bar{}$.

3. ~~$a+b=c$~~
 $a \cdot b=c$

4. $a+0=a$

5. $a \cdot 1=a$

6. $a+b=b+a$

7. $a \cdot b=b \cdot a$

8. $a+(b+c)=(a+b)+c$

9. $a \cdot (b+c)=(a \cdot b)+c$

10. $a+\bar{a}=1$

11. $a \cdot \bar{a}=0$

• Principio de dualidad:

$+ \rightarrow \cdot$ Enunciado Falso, con los ($a \cdot b$) de falso.

$\cdot \rightarrow +$ ~~abacada~~

$0 \rightarrow 1$ $a+\bar{a}=1 \leftarrow a, \bar{a}=\emptyset$

$1 \rightarrow 0$ Justo

• Teorema de De Morgan:

$$\overline{a+b} = \bar{a} \cdot \bar{b}; \quad \overline{a \cdot b} = \bar{a} + \bar{b}$$

Funciones Posibles:

A	B	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}
0	0	0	0	0	0	1	0	0	1	1	1	0	0	1	1	1	L
0	1	0	0	0	1	0	0	1	0	1	0	1	1	0	1	1	1
1	0	0	0	1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	1	0	1	0	0	1	1	0	0	1	0	1	1	1	0	1	1

$\downarrow F=0$ $\downarrow A \cdot B$ $\downarrow \bar{A} \cdot B$ $\downarrow A$ $\downarrow B$ $\downarrow \bar{B}$ $\downarrow \bar{A}$ $\downarrow \bar{A} + B$
 $F = A \cdot B$ (AND) $\bar{A} + B$ (OR)

Mltérminos:

Mdtérminos:

$$F(A, B, C) = \Sigma_m(0, 2, 4)$$

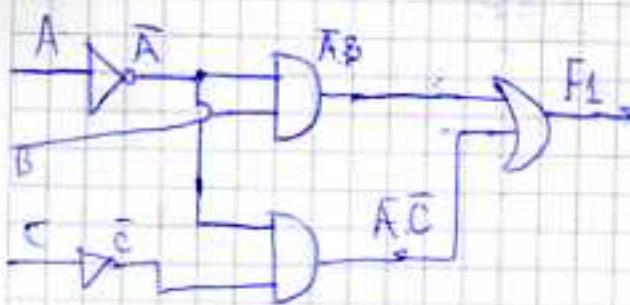
A	B	C	F_1
0	0	0	1
1	0	0	0
0	1	0	0
1	1	1	1
0	1	0	0
1	0	0	0
1	1	0	0
1	1	1	0

Implicante:

Agrupación de 2 mltérminos o mdtérminos.

Implicante de n^o orden: Implicante
Primo

Represento f1:



30/3

Ej. 20:

$$F_1(A+B+C) = A+B+C$$

	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Suma de productos = $\sum m(1, 2, 3, 4, 5, 6, 7)$

Productos de Suma = $\prod M(0)$

Formulas canónicas: { Disyuntiva, Conjuntiva } $(A+B+C)$

$$F_2(A, B, C) = A \cdot B + B \cdot C \bar{D} + \bar{A} \cdot C \bar{D}$$

La Ley de De Morgan:

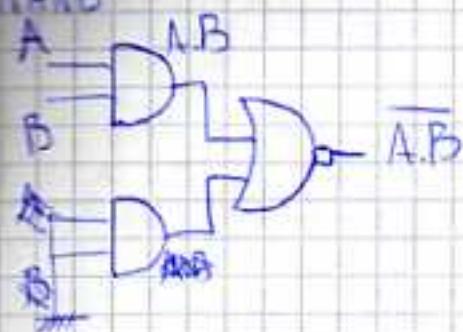
$$= \overline{A \cdot B} \cdot (\overline{B} \cdot \overline{C} \cdot \overline{D}) + \overline{\bar{A}} \cdot \bar{C} \cdot \bar{D}$$

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

$$= \cancel{\bar{A} \cdot \bar{B}} + \bar{A} \cdot \bar{C} + \bar{A} \cdot D + \underbrace{\bar{B} + \bar{B} \cdot \bar{C}}_{\bar{B}} + \bar{B} \cdot D + \bar{A} \cdot C \cdot \bar{D}$$

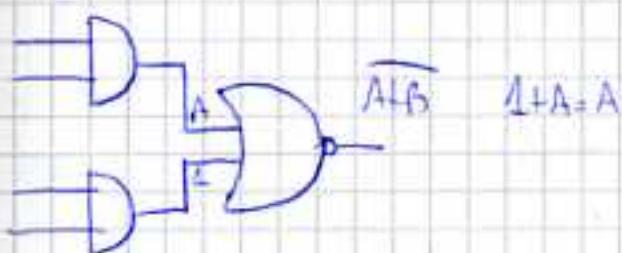
$$= A \cdot \bar{C} + \bar{A} \cdot D + \bar{A} \cdot C \cdot \bar{D} + \bar{B}$$

NAND



$$0+A \cdot B = \overline{A \cdot B}$$

NOR



$$1+A = \overline{A}$$

Ej 32:

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$F = AB \cdot C + \overline{ABC} + \overline{ABC} + ABC$$

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

NO EXISTE XOR de 3 ENTRADAS

Ej 35:

* Mapa de Carnot

			CD
AB	0 0	0 1	1 1
0 0			
0 1		0 0	
1 1			
1 0			

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

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

$$F_1(A; B; C) = \sum m(0, 1, 4, 5, 9, 11, 14, 15) + d(10, 13)$$

→ La d es de donde come. También piden diferentes de la redundancia

	CD	0 0	0 1	1 1	1 0
AB		0	1	3	2
0 0	1				
0 1	1	5			
1 1	1	1	3	15	14
1 0	8	9	11	10	

$$\text{Impl. cte } (0, 1, 4, 5) = \bar{A} \cdot \bar{C}$$

Los impl. ctes

$$\text{Impl. cte } (9, 11, 15, 14) = A \cdot D$$

Se compon de potencias de 2

$$\text{Impl. cte } (10, 13, 16) = A \cdot C$$

2, 4, 8, 16

- *: Impl. cte esencial: Tienen un máx. de min. Baja que no tiene otro impl. cte primo.

0	1	4	5	9	11	14	15
✓	✓	✓	✓	✓	✓	✓	✓
✓	✓						
✓							
✓							

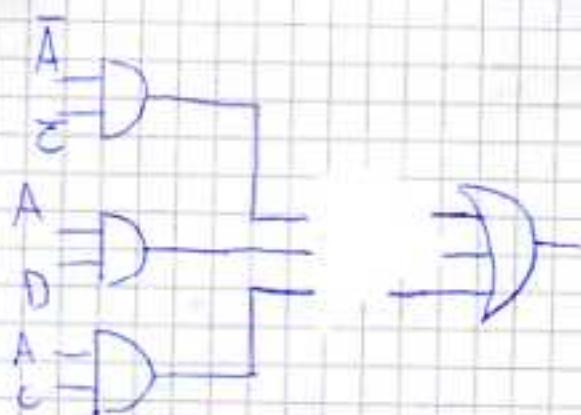
1) Impl. cte (TODO)

2) Impl. cte esencial

3) Verificar los valores mín. o máx. t.

4) Formar la función booleana IPE + IP que cumple

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



40.

	DE	BC	
00	0	1	0
01	1	0	1
11	1	1	1
10	1	0	0

\bar{A}

	DE	BC	
00	0	1	0
01	1	0	1
11	1	1	1
10	1	0	0

A

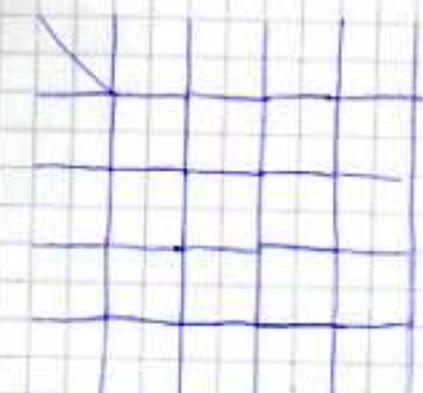
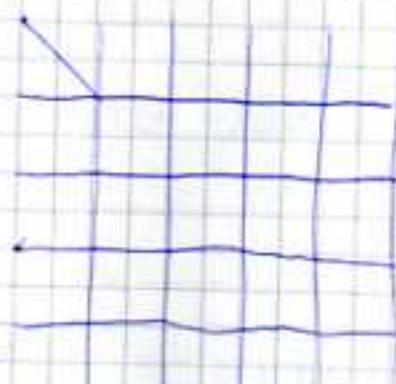
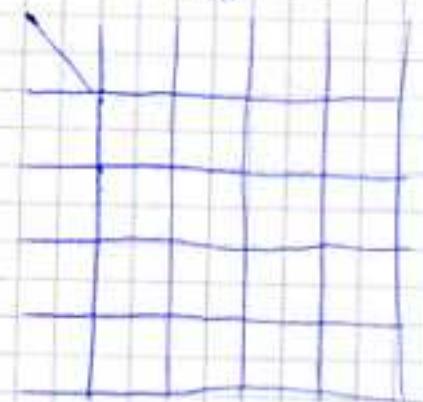
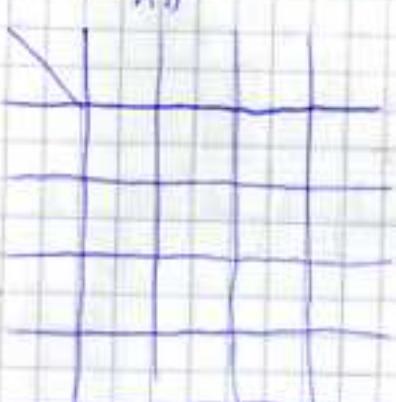
$$I_{pE_1} = \bar{B} \cdot \bar{C}$$

$$I_{pE_2} = \bar{A} \cdot B \cdot C$$

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

+ Common Com. Variables:

* Cartón con 6 Variables:

 $\bar{A}\bar{B}$  $\bar{A}B$  $A\bar{B}$  AB

Quito 2 Variables deven.

	ABCD	F
0	0000	0
1	0001	1
2	0010	1
3	0011	0
4	0100	1
5	0101	1
6	0110	0
7	0111	0
8	1000	1
9	1001	0

	AB	CD	
0	00	00	0
1	00	01	1
2	01	00	0
3	01	01	1
4	10	00	0
5	10	01	1
6	11	00	0
7	11	01	1
8	10	10	0
9	10	11	1
10	11	00	0
11	11	01	1

Todos son esenciales.

$$I_{P_1}(2;10) = \bar{B} \cdot C \cdot \bar{D}$$

$$I_{P_2}(4;14) = B \cdot \bar{C} \cdot \bar{D}$$

$$I_{P_3}(8;11) = A_4 \cdot \bar{C} \cdot \bar{D}$$

$$I_{P_4}(3;13) = B \cdot C \cdot D$$

$$I_{P_5}(1) = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D}$$

	F1	F2	F3	C1	C2	C3
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	0
3	1	1	0	0	0	0
4	0	0	0	0	0	0
5	1	0	1	0	0	0
6	1	0	0	1	0	0
7	1	1	1	0	0	0

$F = F_1 \cdot F_2 + F_1 \cdot F_3 + F_2 \cdot F_3$

	F_1	F_2	F_3	F
0	0	0	1	0
1	0	1	0	1
2	1	0	1	1

F

0	0	1	0
1	0	0	0

C_1

0	0	0	1
0	1	0	0

C_2

0	1	0	0
0	0	0	1

C_3

$$I_{p_1} = (5, 7)$$

$$I_{p_2} = (6, 7) \quad F = F_1 \cdot F_2 + F_1 \cdot F_3 + F_2 \cdot F_3$$

$$I_{p_3} = (3, 7)$$

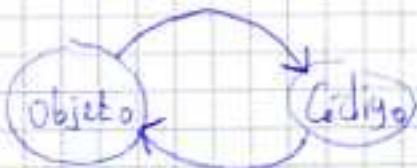
$$C_1 = \overline{F_1} \cdot F_2 \cdot F_3 + F_1 \cdot \overline{F_2} \cdot F_3$$

$$C_2 = \overline{F_1} \cdot F_2 \cdot \overline{F_3} + F_1 \cdot \overline{F_2} \cdot \overline{F_3}$$

$$C_3 = \overline{F_1} \cdot \overline{F_2} \cdot F_3 + F_1 \cdot \overline{F_2} \cdot \overline{F_3}$$

* Código:

El código hace referencia a un objeto. Existe una relación uno a uno entre ellos.



Los sistemas numéricos son códigos.

Ejemplo:

8	4	2	1	5	3	1	1	5	2	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1	0	1	0	1	0
2	0	0	1	0	0	0	1	1	0	0	0	0
3	0	0	1	1	0	0	0	0	0	0	0	0
4	0	1	0	0	0	1	1	0	1	0	0	0
5	0	1	0	1	0	1	1	1	1	0	0	0

Para cada número debe haber un solo código.

Ej. 4:

6	3	2	-2
0	0	0	0
1	0	1	0
2	0	0	1
3	0	1	0
4	1	0	1
5	0	1	1
6	1	0	1
7	1	1	0
8	1	0	0
9	1	1	1

Ej. 5:

6	4	2	1
0	0	0	0
1	0	0	1
2	0	0	0
3	0	0	1
4	0	1	0
5	0	1	0
6	0	1	0
7	0	1	1
8	1	2	0
9	1	0	1

	5	2	1	1
0	0	0	0	0
1	0	0	0	1
2	0	1	0	0
3	0	1	1	0
4	0	1	1	1
5	1	0	0	0
6	1	0	0	1
7	1	0	1	0
8	1	0	1	1
9	1	1	0	0
10	1	1	0	1
11	1	1	1	0
12	1	1	1	1

✓

* Código auto-complementable:

El complementario binario -1 decide si el número es el número bit-a-bit del mismo.

Código X-3:

	X-3
0	0 0 1 1
1	0 1 0 0
2	0 1 0 1
3	0 1 1 0
4	0 1 1 1
5	1 0 0 0
6	1 0 0 1
7	1 0 1 0
8	1 0 1 1
9	1 1 0 0

1/4

* Álgebra de Boole

• Postulados de Huntington (5 en 8)

① Comunitatividad:

- A+B = B+A
- A.B = B.A

② Distributividad:

- A.(B+C) = A.B + A.C
- A+B.C = (A+B)(A+C) → No es álgebra convencional

3) Identidad o inversa:

a) $A + 0 = A$

b) $A \cdot 1 = A$

4) Complemento:

a) $A + \bar{A} = 1$

b) $A \cdot \bar{A} = 0$

• Teoremas o Propiedades:

1) Idempotencia:

a) $A + A = A$

b) $A \cdot A = A$

2) Elementos nulos:

a) $A + 1 = 1$

b) $A \cdot 0 = 0$

3) Invención:

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

4) Absorción:

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

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

5) Asociatividad

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

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

6) Leyes de De Morgan:

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

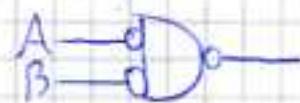
b) $A \cdot B = \overline{\overline{A} + \overline{B}}$

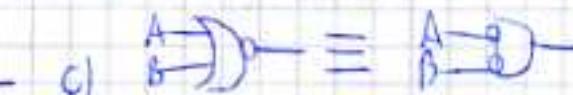
c) $\overline{A + B} = \overline{A} \cdot \overline{B}$

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

• Encuentras (Leyes de De Morgan):

a) 

≡ 

c) 

b) 

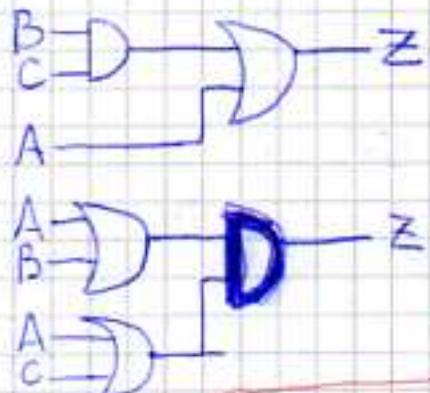
≡ 

d) 

El álgebra de Boole se utiliza para reducir complejidades. Supongamos:

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

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



Ambas funciones son
de 2 niveles.

Coste lógico de una función:

- 1) Menor cantidad de niveles.
- 2) II II II Compuertas.
- 3) II II II Entradas.

* Función Cánónica: expresada como minitermino o
maxitermino. Todas las terminaciones tienen variables.

* Función mínima: Cada término tiene la menor
cantidad de variables posibles.

La función mínima es hasta 2 inclusive. Si tiene 3 ó más, Seguro que no
es la mínima (hay posiciones únicamente de 2 niveles).

• Término:

- Suma de productos (1) $\rightarrow A \cdot B \cdot C \rightarrow$ Minitermino
- Producto de suma (2) $\rightarrow (A + B + C) \rightarrow$ Maxitermino

• Terminos:

Puedes representar una función con tantos unos como tengas en
la función.

Negado cuando vale 0 y sin negar cuando vale 1.

$Z = \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C \Rightarrow$ Tengo cinco 1
 (Si us. 5 and y una or, no es mínima. Tiene 2 niveles pero más compuertas que el anterior)

(1) PSM

(2) ↓ PSM → Negado cuando Val = 1 y sin negar cuando Val = 0.

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

(Us. 4 compuertas en total, 3 or y una and, más barato que para 5Pm)

$$Z = A \cdot B \cdot C$$

$$Z = A \cdot 1 \cdot 1 + 1 \cdot B \cdot C$$

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

$$Z = A \cdot B \cdot C + A \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot C + \cancel{A \cdot \bar{B} \cdot \bar{C}} + \cancel{\bar{A} \cdot B \cdot C}$$

$$Z = \underbrace{(A \cdot A)}_{0} \cdot B \cdot C + A \cdot \bar{B} \cdot \underbrace{(\bar{C} \cdot C)}_{0} + A \cdot B \cdot \underbrace{(\bar{C} \cdot C)}_{0}$$

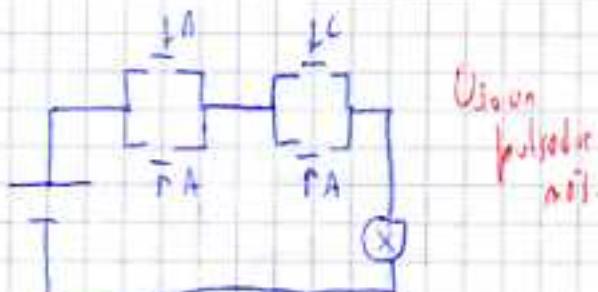
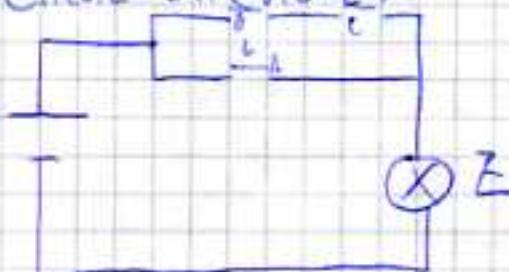
$$Z = B \cdot C + A \cdot \bar{B} + A \cdot A$$

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

$$\boxed{Z = A \cdot B \cdot C}$$

A partir de cualquier función obtendrá la normalizada

Circuito con contacto:

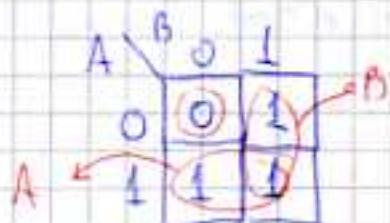


Usa un pulsador NC.

Método gráfico: Veitch-Karnaugh

$$\begin{array}{|c|c|c|c|} \hline A & B & C & Z \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 1 \\ \hline 1 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 \\ \hline \end{array}$$

→



Somando

$$Z = A + B (X \text{ unit})$$

Sol. combina una variable de la celda adyacente.

$$Z = A + B \bar{Y} (X \text{ neg})$$

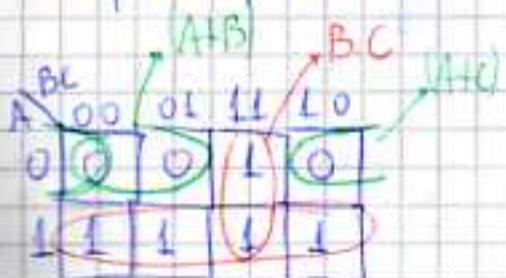
Tengo que buscar los implicantes primos (no puede para ser cubierto por uno de mayor orden)

2^n celdas (agrupan potencia de 2)

Simplifica n Variables.

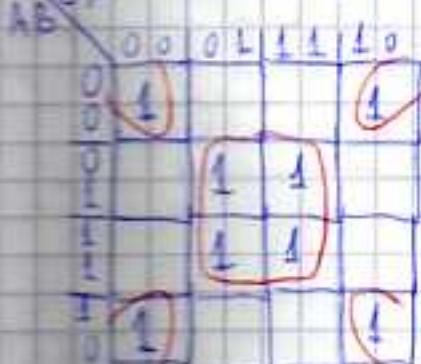
$$Z = \overline{A} \cdot \overline{B} + A \cdot \overline{B} + A \cdot B = \overline{B}(\overline{A} + A) + A(\overline{B} + B) = \overline{A} + B$$

Implicante primo esencial.



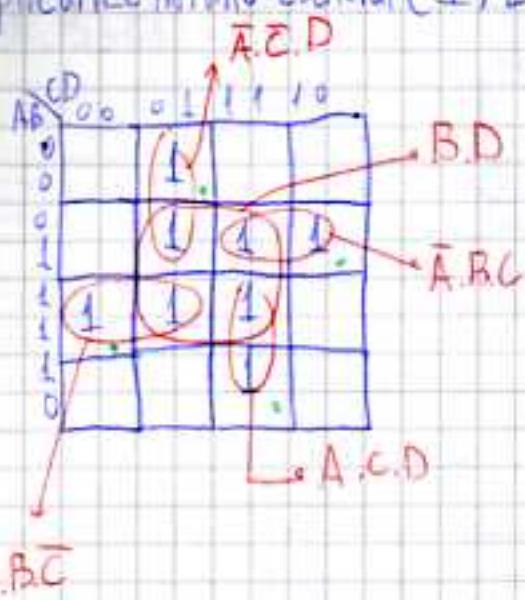
$Z = A + B \cdot C$ \Downarrow <u>Mínima</u>	Potencial Por Celdas $Z = (A+B)(A+C)$
--	--

AB con otras Variables



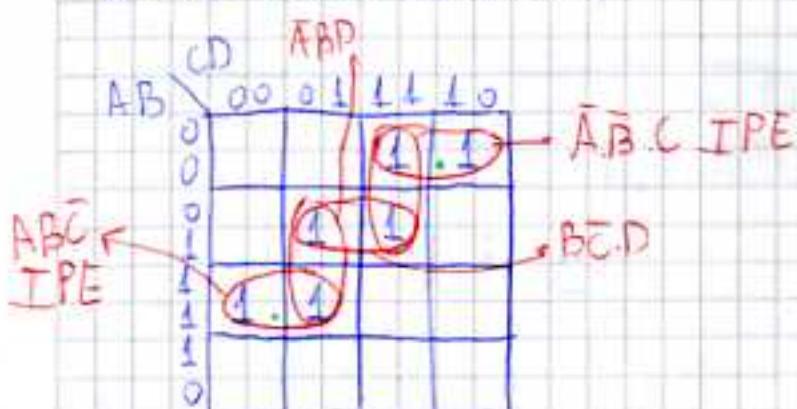
Cuando dos Funciones responden a la misma Tabla de Verdad, se dice que son equivalentes.

Implicante mínimo esencial (IPE) \Rightarrow Cubre uno solo.



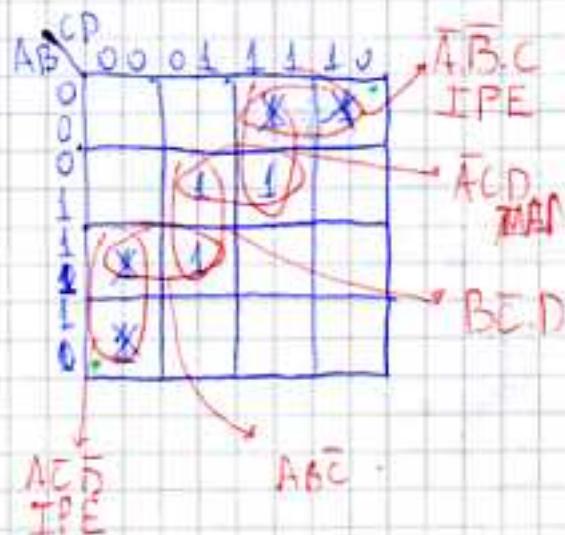
$$Z = \bar{A} \bar{C} D + \bar{A} B C + A C D + A B C$$

Con los demás cubriendo.



$$Z = A B C + A B C + A B D$$

IEP IEP IP

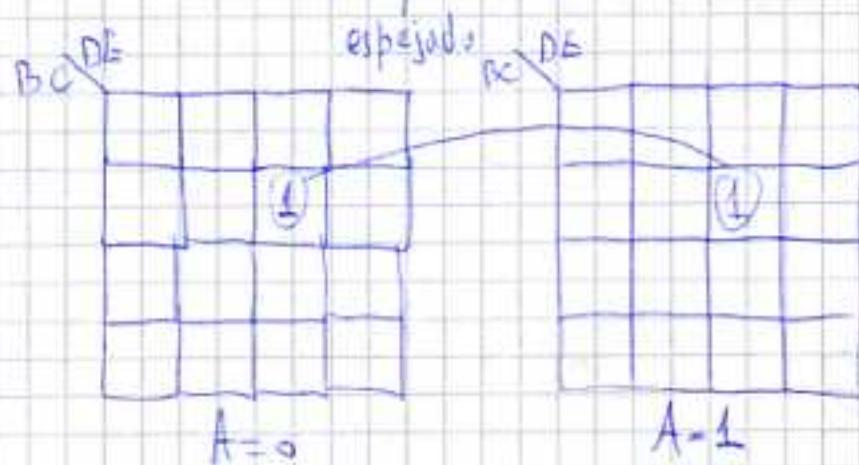


$$Z = A C D + A B C + A C D + B C D ; Z = A C \bar{D} + \bar{A} \bar{B} C + \bar{A} B D + A B C ; Z = \bar{A} C D + \bar{A} \bar{B} C + A B D + B C D$$

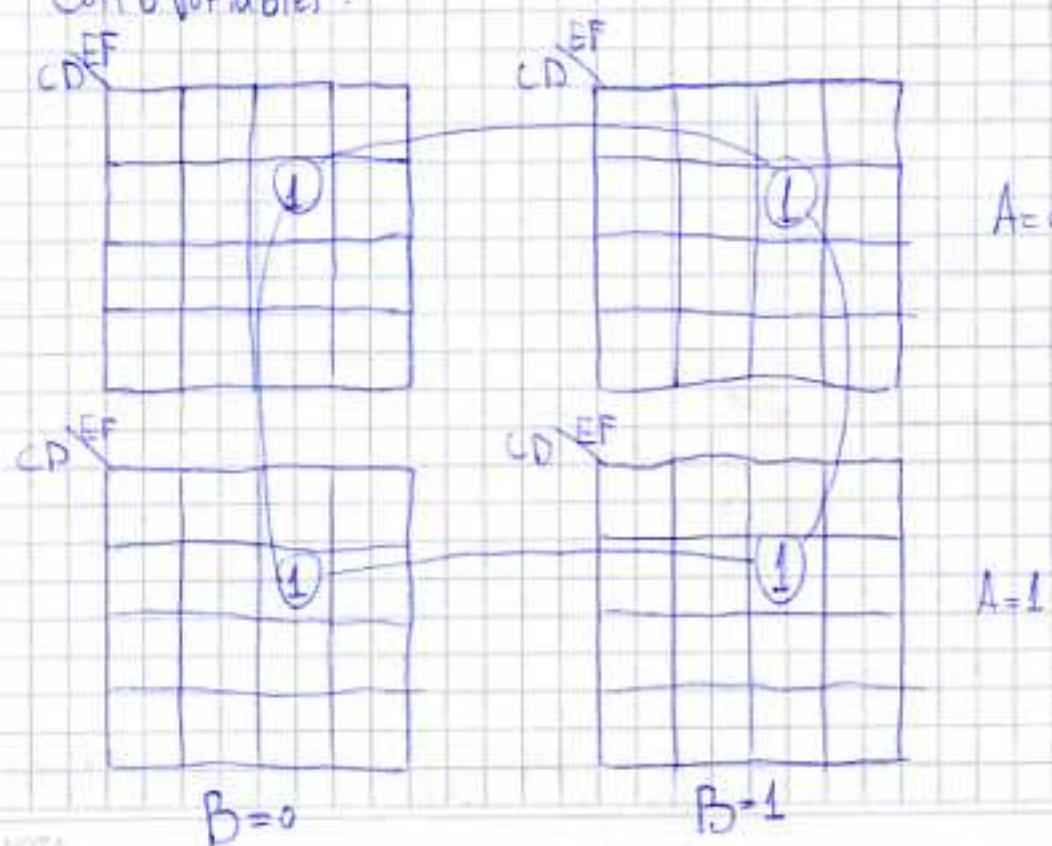
NOTA: Tengo 3 funciones mínimas, con mismo nivel y computado por 0 son menores?

Con 5 variables:

A\B\DE	000	001	011	010	110	111	101	100
0	1							
1								

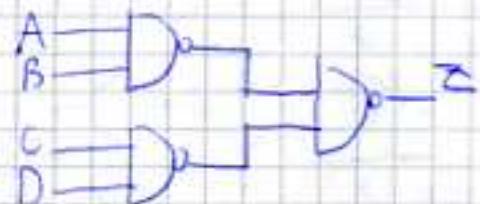
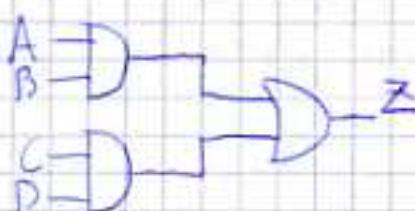


Con 6 variables:



$$Z = \overline{\overline{A} \cdot \overline{B} + \overline{C} \cdot \overline{D}} = \overline{\overline{A} \cdot \overline{B}} \cdot \overline{\overline{C} \cdot \overline{D}}$$

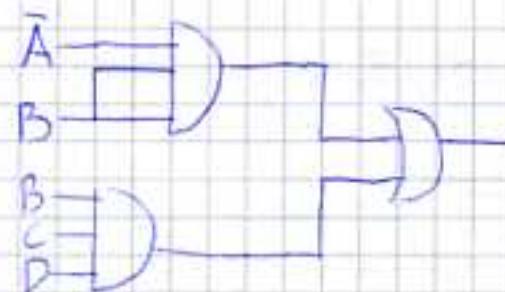
$$\Rightarrow Z = \overline{\overline{A} \cdot \overline{B}}$$



SP → NAND
PS → NOR

(Urg. solo usar integrados)

$$Z = \overline{A} \cdot B + B \cdot C \cdot D$$



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

$$\overline{A} \cdot B \cdot 1 = A \cdot B$$

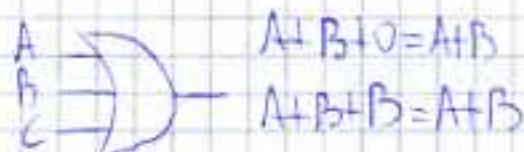
En log. col. us. 1

|| || or || ||



Cuando necesito de

Con 3 entradas, op. lú
os b.:

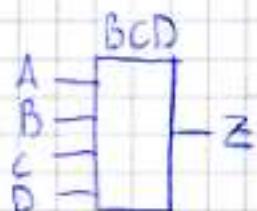


Número primo del 0 al 9:

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

1 1 1 0 X
1 1 1 1 X

Los 1 como
comodines



Por unos:

AB	CD	00	01	11	10
00	00	0	1	1	1
00	01	0	1	1	0
11	X	X	X	V	
11	00	0	0	X	X

$$Z = \bar{A}D + \bar{B}C$$

Función mínima
(unos constitutivos)



Por ceros:

AB	CD	00	01	11	10
00	00	0	1	1	1
00	01	0	1	1	0
11	X	X	X	X	
11	00	0	0	X	X

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



Son equivalentes porque responden a la misma tabla de verdad (en el mismo rectángulo de la función). Pero no son iguales.

Riesgos

Una función de 2 niveles tiene libertad de 4 tipos distintos

1	1
1	1

- unos:

AB	CD	00	01	11	10
00	0	1	1	1	1
01	0	1	1	0	0
11	X	X	X	V	
10	0	0	X	X	

$$Z = \bar{A}D + \bar{B}C$$

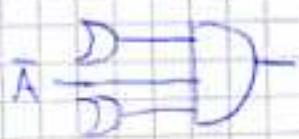
Función mínima
(unos cumplidos)



Por ceros:

AB	CD	00	01	11	10
00	0	1	1	1	1
01	0	1	1	0	0
11	X	X	X	X	
10	0	0	X	X	

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



Son equivalentes porque responden a la misma tabla de verdad (en el mismo rectángulo de la función). Pero no son iguales.

Riesgos

Una función de 2 niveles tiene libertad de 4 posibilidades

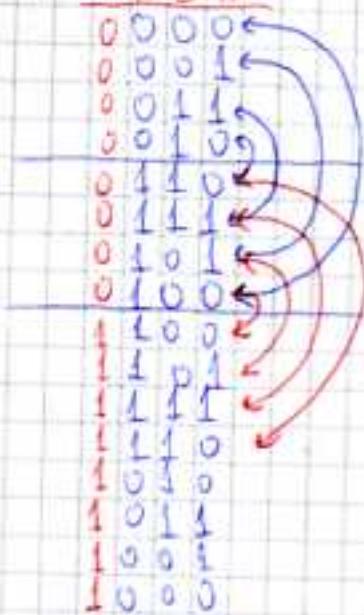
1	1
1	1

6/4

* Código Reflejado.

Ej. 8.

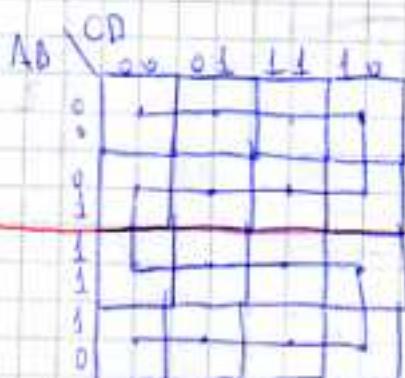
A B C D	0 0 0 0	0 0 0 1	0 0 1 1	0 0 1 0	0 1 0 2	0 1 1 1	0 1 0 1	0 1 0 0	1 0 0 0	1 1 P 1	1 1 1 0	1 0 1 0	1 0 0 1	1 0 0 0
------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------	------------------



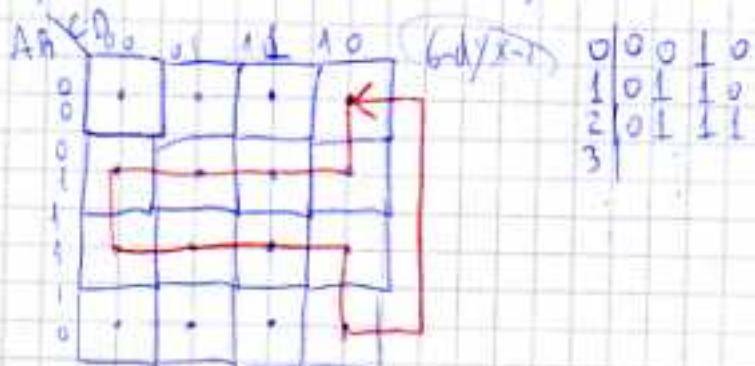
• Código hexadecim Gray.

- 1) Es un código reflejado.
- 2) Código progresivo.
- 3) El código es cerrado, pues entre el último dígito y el primero hay un bit.
- 4) El código no es pesoado.

ENEL MAPA DE CARNOT:



Si quisieras ver el código en el 3d (Gray).



Distancia entre Códigos:

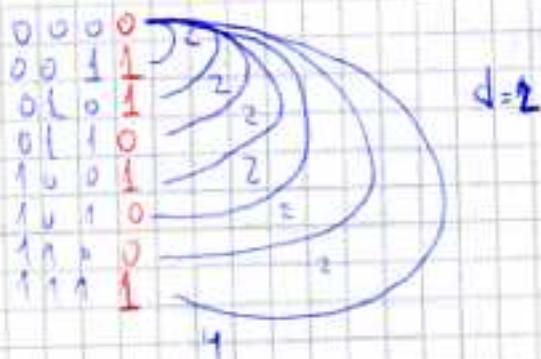
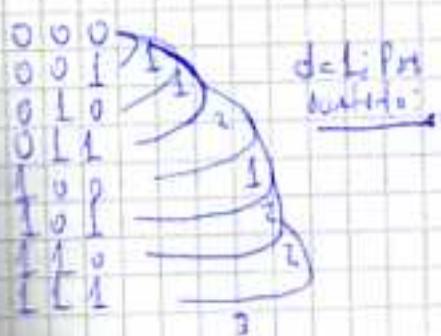
Cada Variación de 0 a 1 es la diferencia entre códigos.

Ej:

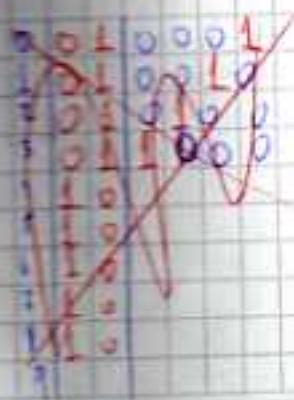


La distancia mínima es 3.

- Si tengo distancia mínima y quiero disminuirlo, digo, un bit de probabilidad de cada dígito:



Biquintario:



• Código Biquintario:

0	0	1	0	0	0	0
1	0	1	0	0	0	1
2	0	1	0	0	1	0
3	0	1	0	1	0	0
4	0	1	1	0	0	0
5	1	0	0	0	0	1
6	1	0	0	0	1	0
7	1	0	0	1	0	0
8	1	0	1	0	0	0
9	1	0	1	0	0	0

• Código Dos DE Cinco:

0	1	1	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	0	0
4	0	1	1	0
5	0	1	0	0
6	0	0	1	0
7	1	0	0	1
8	1	0	1	0
9	1	0	0	0

• Código de Johnson:

0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	1
4	0	1	1	1
5	1	1	1	1
6	1	1	1	0
7	1	1	0	0
8	1	0	0	0

+ Método de recepción difusa.

~~✓ Mito~~ Algoritmo

Transmision:

$$P_1 = m_1 \oplus m_2 \oplus m_3$$

$$P_2 = m_1 \oplus m_3 \oplus m_4$$

$$P_3 = m_2 \oplus m_4 \oplus m_5$$

Recepcion:

$$\check{V}_1 = P_1 \oplus m_1 \oplus m_2 \oplus m_3$$

$$\check{V}_2 = P_2 \oplus m_2 \oplus m_3 \oplus m_4$$

$$\check{V}_3 = P_3 \oplus m_3 \oplus m_4 \oplus m_5$$

Ejemplo:

P_1 Punto de suministro

1100100

ENTRADA

P_2 Punto de suministro

1100100

100

$$P_1 = 0 \oplus 1D_0 = 1$$

$$P_2 = 0 \oplus 1D_0 = 1$$

$$P_4 = 1 \oplus 1D_0 = 0$$

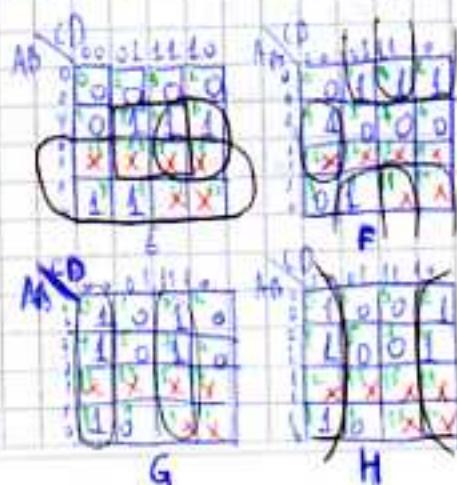
$$V_L = 1D_0 \oplus 1D_0 = 0$$

$$V_3 = 1D_0 \oplus 0D_0 = 1$$

$$V_4 = 0 \oplus 1D_0 = 0$$

• Passar de códigos:

A B C D	E F G H
8 1 4 5 2 1	X-3
0 0 0 0 0	0 0 0 1 1
1 0 0 0 1	1 0 1 0 0
2 0 0 1 0	2 0 1 0 1
3 0 0 1 1	3 0 1 1 0
4 0 1 0 0	4 0 1 1 1
5 0 1 0 1	5 1 0 0 0
6 0 1 1 0	6 1 0 0 1
7 0 1 1 1	7 1 0 1 0
8 1 0 0 0	8 1 0 1 1
9 1 0 0 1	9 1 1 0 0



E:

$$I_{p_L}(1, 2, 4, 5, 13, 14, 15) = A$$

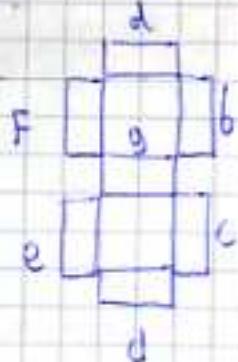
$$I_{p_L}(5, 2, 13, 15) = BD$$

$$I_{p_L}(6, 7, 14, 15) = BC$$

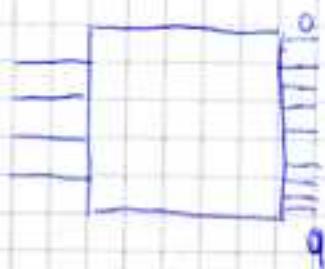
$$E = A + BD + BC$$

F:

Ej:



	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	0	0	1	1



* Consultar: Johnson.

A	B	C	D	E	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	0	0	0
2	0	0	0	1	1	0	0	1	0	0	0	0	0	0
3	0	0	1	1										
4	0	1	1	1										
5	1	1	1	1										
6	1	1	1	0										
7	1	1	0	0										
8	1	0	0	0										
9	1	0	0	0										

8/4

CÓDIGOS < NUMÉRICOS — BCD: Binary Coded Digit.
ALFANUMÉRICOS — ASCII

CARACTERÍSTICAS / PESADO O PONDERADO
BCD / AUTO COMPLEMENTARIO
PROGRESIVO
CERRADO
REFLEJADO

* Código BCD:

Tienen 4 bits. El más bajo es el 8421 (binario natural):

Para pasar a BCD, se pasa dígito a dígito:

2 | 4 | 3
0 0 1 0 0 1 0 0 1 1

8	4	2	1	→ Código pesado, peso - 8421
0	0	0	0	
1	0	0	0	
2	0	0	1	
3	0	0	1	
4	0	1	0	
5	0	1	0	
6	0	1	1	
7	0	1	1	
8	1	0	0	
9	1	0	0	

Otro ejemplo:

7 4 2 1 → Código pesado:
0 0 0 0 0
1 0 0 0 1
2 0 0 1 0
3 0 0 1 1
4 0 1 0 0
5 0 1 0 1
6 0 1 1 0
7 0 1 1 1
8 0 0 0 0
9 1 0 0 1
NOTA 9 1 0 1 0

■ Para que el código sea válido, debe sumar por lo menos 4, lo que

* Código: El 0 y el 9 son complementarios, el 1 y el 8, el 2 y el 7, el 3 y el 6, el 4 y el 5 y el 5 con el 9, y viceversa. No necesariamente son pares.

* Código Reflejado:

Gray:

Para 1 bit:

0
1

Para 2 bit

Completo
con 0
0 0
0 1
1 0
Completo
con 1
1 1

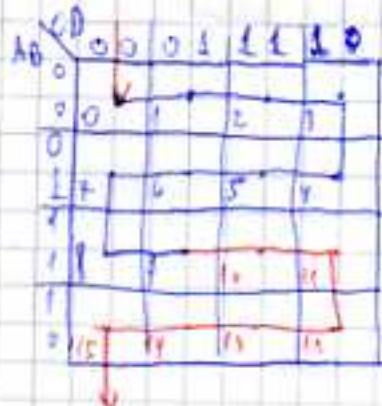
Para 3 bit

Completo
con 0
0 0 0
0 0 1
0 1 1
0 1 0
1 1 0
Completo
con 1
1 1 1
1 1 0
1 0 1
1 0 0

Para 4 bit

Completo
con 0
0 0 0 0
0 0 0 1
0 0 1 1
0 0 1 0
0 1 1 0
0 1 1 1
0 1 0 1
0 1 0 0
1 1 0 0
1 1 0 1
1 1 1 1
1 1 1 0
1 0 1 0
Completo
con 1
1 0 1 1
1 0 0 1
1 0 0 0
0 0 0 0

* El código reflejado, entre cada dígito cambia solo un bit. En el mapa de Karnaugh:



El código Gray es:

* Riflesado: Para construcción.

* Prueba: Cambio de un bit.

* Cerrado: Entre el último y el primero cambia 1 bit.

El código Gray BCD No es cerrado. Para que lo sea, puede empezar por el 3:

	00	01	10	11	
0					✓
0					0
1	4	3	*	1	
1	*	1	*	7	
1	5	6	*	*	
0					*

Gray BCD X-3:

0	0	0	1	0
1	0	1	1	0
1	0	1	0	1
1	0	1	0	0
1	1	1	0	0
1	1	1	1	1
1	1	1	1	0
1	1	0	1	0

Ahora sí es cerrado.

	00	01	10	11	0
0	0	1	3	2	
0	4	5	7	6	
1					
1	8	9			
0					

* Distancia minima = distanciamax
dist = 3, igual a 1, 16 bits

*Error de bits:

Sedgrega un bit más para lo anterior.

	A	B	C	D	P
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0

Bié depende del: Si hay 1 impares,
Vale 1, Si no vale 0.

En karnaugh:

AB	CD	00	01	11	10
00	00	0	3		
01	00	5	6		
11	00				
10	00	9			

$$P=0$$

AB	CD	00	01	11	10
00	00	-	1	-	2
01	00	4	-	7	-
11	00	-	-	-	-
10	00	9	-	-	-

$$P=1$$

Distancia mínima = Aquí es 2, entre cada digito hay 2 bits.

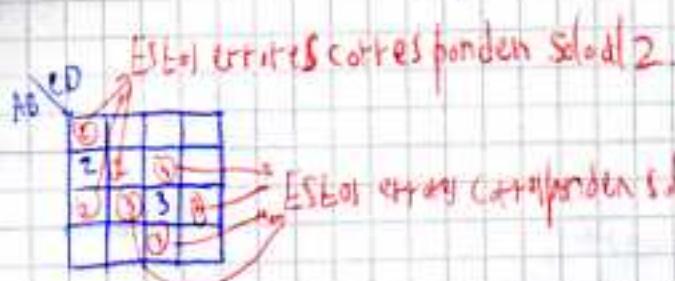
*Con distancia mínima 2, puedes corregir el error, pero no saber de donde viene.

AB	CD	00	01	11	10
00	00	0	1	-	-
01	00	5	6	-	-
11	00	-	-	-	-
10	00	9	-	-	-

Este código puede ser error del 3,5 ó 6.

* Si pongo distancia mínima 3, puedo categorizar el error y ver de donde viene:

Vino:



$$D_{\min} = 3$$

④ Fórmula para la detección o corrección de errores:

$$d_{\min} = \text{detector} + \text{Corrector} + 1$$

Ej:

$$1 = 0 + 0 + 1$$

→ No puedo detectar ni corregir ningún error.

$$2 = 1 + 0 + 1$$

→ Puedo detectar 1 error y no corregir ninguno.

⑤ Código de Johnson (BCD):

	0	0	0	0	0
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	1	1
3	0	0	1	1	1
4	0	1	1	1	1
5	1	1	1	1	1
6	1	1	1	1	0
7	1	1	1	0	0
8	1	1	0	0	0
9	1	0	0	0	0

Método de codificación:

* Para la detección y corrección de errores de forma sistemática:

① Hamming

② Reed-Solomon

Método de Hamming:

$$k_1, k_2, m_1, m_2, m_3, m_4, m_5, m_6, m_7$$

$$k_1 = m_1 \oplus m_3 \oplus m_7$$

$$k_2 = m_1 \oplus m_2 \oplus m_7$$

$$k_4 = m_3 \oplus m_6 \oplus m_7$$

Se asume un solo error.



* En el envío:

$$k_1, k_2, m_1, m_2, m_3, m_4, m_5$$

0	1	1	1	1	1	1
---	---	---	---	---	---	---

0 1 1 1 → Es el mensaje original.

k ₁	k ₂	k ₄
1	1	1
1	0	0
1	0	1
0	1	1

$$k_1 = 0 \oplus 1 \oplus 1 = 0$$

$$k_2 = 0 \oplus 1 \oplus 1 = 0 \implies$$

0	0	1	0	1	1	1
---	---	---	---	---	---	---

Viaja desde el emisor.

$$k_4 = 1 \oplus 1 \oplus 1 = 1$$

• En la recepción del mensaje:

$$P_1 = k_1 \oplus m_3 \oplus m_5 \oplus m_7 = 0$$

$$P_2 = k_2 \oplus m_1 \oplus m_6 \oplus m_7 = 0$$

$$P_4 = k_4 \oplus m_3 \oplus m_6 \oplus m_7 = 0$$

Las paridades deben valer 0. Luego NO HAY

ERRORES.

S. 8.6. E 6 sección forman U durante el envío:

$$\begin{matrix} K_1 & K_2 & m_1 & K_3 & m_2 \\ 0 & 0 & 0 & 1 & 0 \end{matrix}$$

$$P_1 = K_1 \oplus m_1 \oplus m_2 \oplus m_3 = 0$$

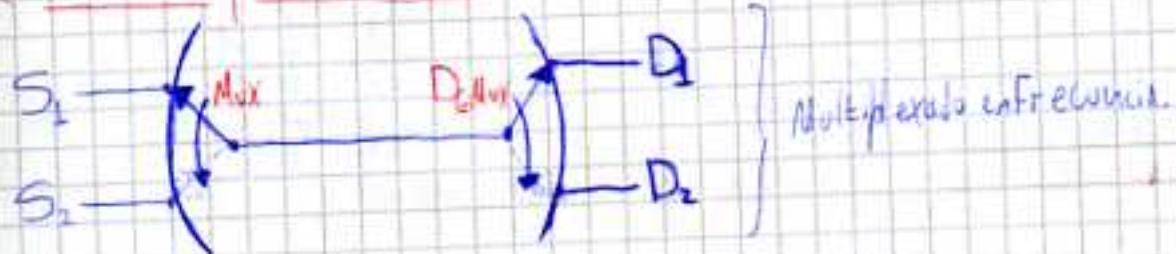
$$P_2 = K_2 \oplus m_2 \oplus m_3 = 0 \rightarrow \text{Error de transmisión.}$$

$$P_3 = K_3 \oplus m_1 \oplus m_2 \oplus m_3 = 0$$

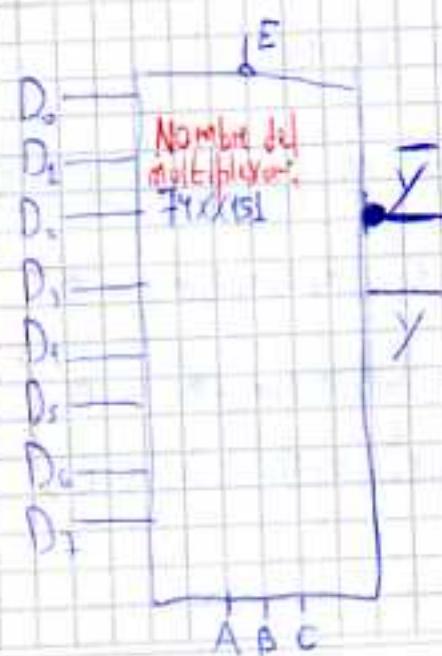
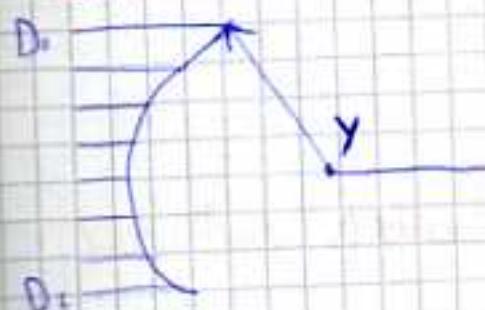
Cuando el error es de un bit, el error es de en K.

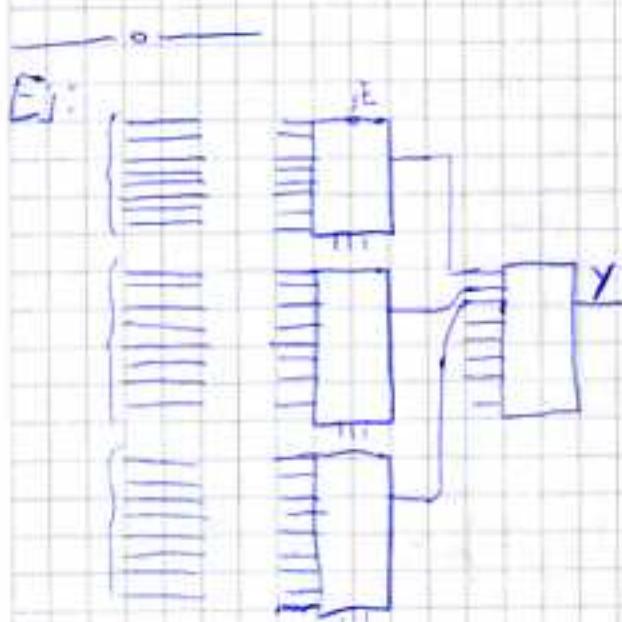
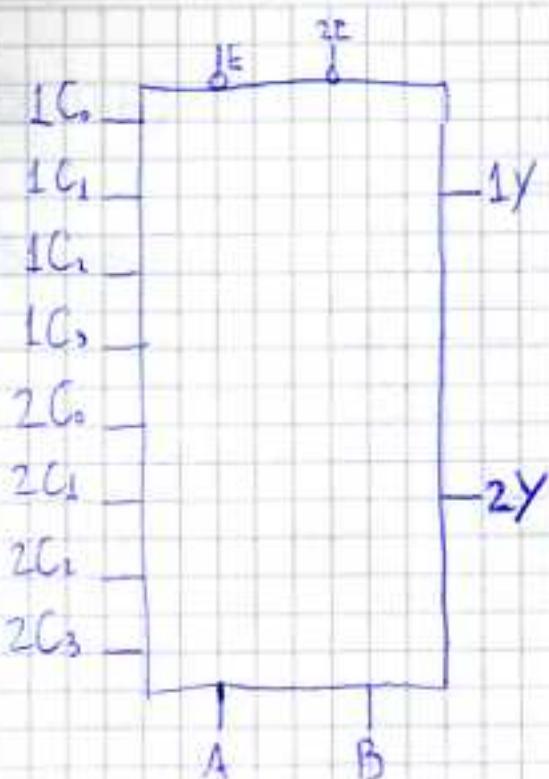
13/4

* Multiplexores:



- Se simboliza así.
- Definirlos como Multiplexores.
- Los datos vienen sincronizados.

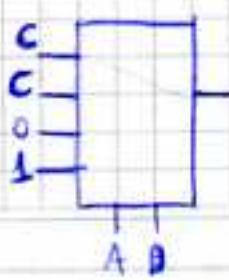




* El multiplexor puede implementarse en funciones.

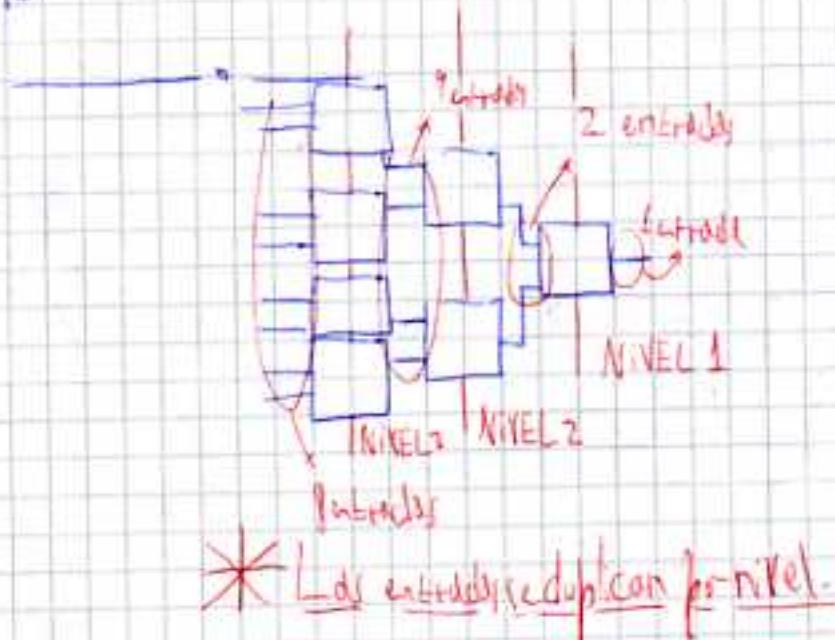
$$F(A,B,C) = \sum m(4,5,6,7)$$

A	B	C	F
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



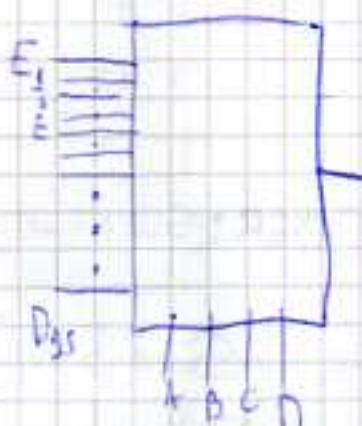
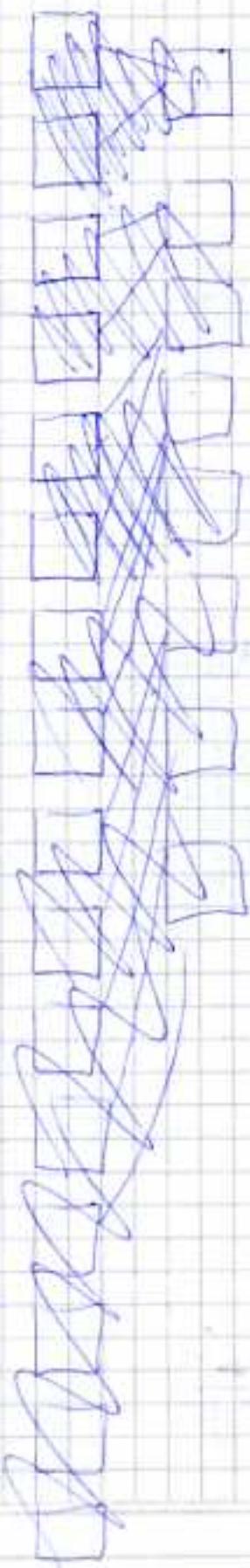
5.5.

A	B	C	D	F
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0



$F(A, B, C, D, E)$

	A	B	C	D	E	F
0	0	0	0	0	0	0
1	0	0	0	0	1	1
2	0	0	0	1	0	1
3	0	0	0	1	1	1
4	0	0	1	0	0	0
5	0	0	1	0	1	0
6	0	0	1	1	0	1
7	0	0	1	1	1	0
8	0	1	0	0	0	0
9	0	1	0	0	1	1
10	0	1	0	1	0	0
11	0	1	0	1	1	1
12	0	1	1	0	0	0
13	0	1	1	0	1	0
14	0	1	1	1	0	0
15	0	1	1	1	1	1
16	1	0	0	0	0	0
17	1	0	0	0	1	1
18	1	0	0	1	0	0
19	1	0	1	0	1	1
20	1	0	1	1	0	0
21	1	0	1	1	1	1
22	1	1	0	0	0	0
23	1	1	0	0	1	1
24	1	1	0	1	0	0
25	1	1	1	0	0	0
26	1	1	1	0	1	1
27	1	1	1	1	0	0
28	1	1	1	1	1	1



* Suma en binario:

El unificador es | El unificador se pone, el otro carry in
carry out $\begin{array}{r} 1 \\ + 0 \end{array} \Rightarrow \begin{array}{r} 0 \\ 1 \end{array}$

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array}$$

* HALF-ADDER: (Suma-Sumador)

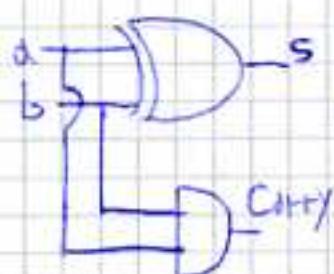
a	b	S	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = \bar{a} \cdot b + a \cdot \bar{b}$$

$$S = a \oplus b$$

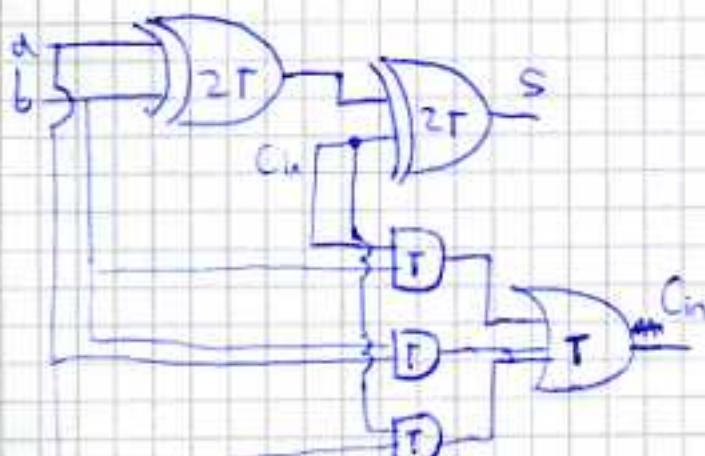
$$\text{Carry} = a \cdot b$$

En circuito:



a	b	S	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

* FULL-ADDER: (Sumador-Completo)



15/4

Codificador

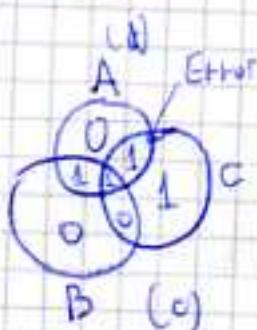
Decoder

Universo de Codigos

MUX

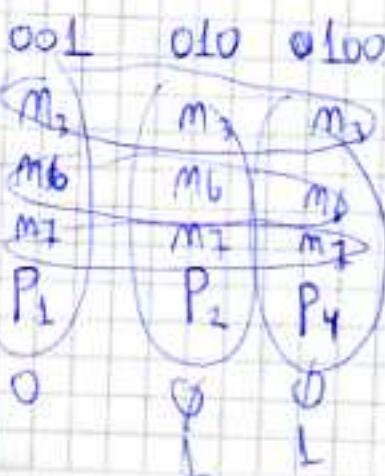
DEMUX

~~★~~ MÉTODO de HUMMING



(a) Codificación de 0011

(b) Adición de paridad por (c) Error en A.C.



* Métodos alternativos de implementación de funciones lógicas:

- Usando Decodificadores y Demultiplexores.
- Usando Multiplexores

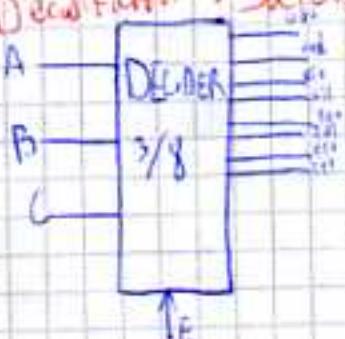
• Codificador: Salen menos entradas que entradas que salidas.



* Los codificadores tienen 2^n entradas.

S_3	S_2	S_1	S_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1

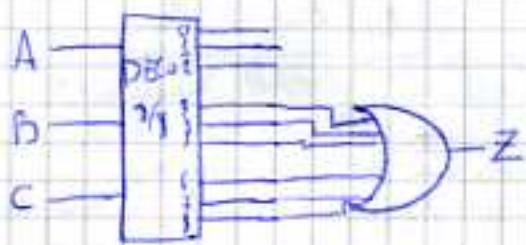
• Decodificador: Salen más entradas que salidas.



ABCSSSSSSSS

0 0 0	1 0 0	0 0 0	0 0 0	0 0 0
0 0 1	0 1 0	0 0 1	0 0 0	0 0 0
0 1 0	0 0 1	0 0 0	0 0 0	0 0 0
0 1 1	0 0 0	1	0 0 0	0 0 0
1 0 0	0 0 0	0 1 0	0 0 0	0 0 0
1 0 1	0 0 0	0 0 1	0 0 0	0 0 0
1 1 0	0 0 0	0 0 0	0 1 0	0 0 0
1 1 1	0 0 0	0 0 0	0 0 0	1 0 0

• S: Salida



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

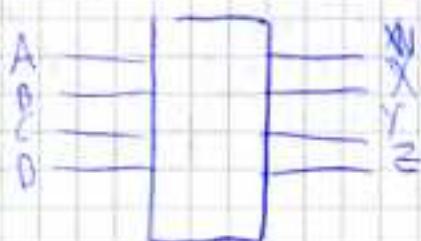
$$Z = A + B \cdot C$$

* Conversor de Código:

Polinomio X-3 a 8421:

X-3
0 0 0 1 1
1 0 1 0 0
2 0 1 0 1
3 0 1 1 0
4 0 1 1 1
5 1 0 0 0
6 1 0 0 1
7 1 0 1 0
8 1 0 1 1
9 1 1 0 0
A B C D

BCD 8421
0 0 0 0 0 0
1 0 0 0 0 1
2 0 0 0 1 0
3 0 0 0 1 1
4 0 0 1 0 0
5 0 0 1 0 1
6 0 1 1 0 0
7 0 1 1 1 1
8 1 0 0 0 0
9 1 0 0 0 1
X Y Z



A	B	C	D	W	X	Y	Z
0	0	0	0	x x x x			
0	0	0	1	x x x x			
0	0	1	0	x x x x			
0	0	1	1	0 0 0 0			
0	1	0	0	0 0 0 1			
0	1	0	1	0 0 1 0			
0	1	1	0	0 0 1 1			
0	1	1	1	0 1 0 0			
1	0	0	0	0 1 0 1			
1	0	0	1	0 1 1 0			
1	0	1	0	0 1 1 1			
1	0	1	1	0 0 0 0			
1	1	0	0	0 0 0 1			
1	1	0	1	x x x x			
1	1	1	0	x x x x			
1	1	1	1	x x x x			

	CD	AB	W	X	Y	Z
0	00	01	11	10	0	x
0	01	10	11	00	1	
1	10	01	11	01	0	
1	11	10	11	10	x	x
1	11	11	11	11	0	1

$$Z = \overline{D}$$

$$Y =$$

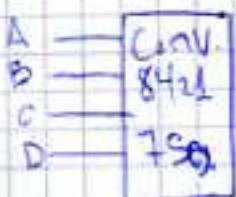
$$X =$$

$$W =$$

Led 7 segmentos:

* Led de 7 segmentos:

F / g / h
e / d / k.

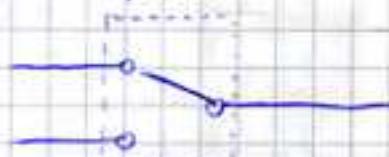


A	B	C	D	E	F	G	H	K
0	0	0	1	1	1	1	1	0
0	0	1	0	1	1	0	0	1
0	1	0	1	1	0	1	0	1
0	0	1	1	1	1	0	0	1
0	1	0	0	1	1	0	1	1
0	1	0	1	0	1	1	0	1
0	1	1	0	1	0	1	1	1
0	1	1	1	1	1	1	1	0
1	0	0	0	1	1	1	1	1
1	0	0	1	1	1	1	0	1
1	0	1	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0
1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0

Todos los 8
números. Los que
tienen 0 por defecto
son los que
nunca se activan.

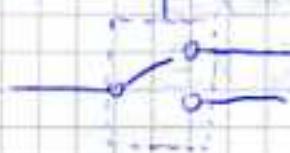
* Multiplexor.

Un multiplexor analógico es implemente una llave:

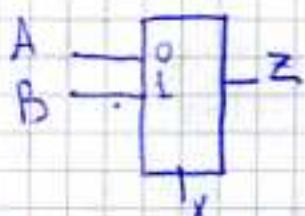


Un multiplexor es un selector.

El multiplexor analógico es igual al multiplexor:



Multiplexor digital:



X \ Z	0 A	1 B
0 A	0	
1 B		0

Si $X=0, Z=A$.

Si $X=1, Z=B$.

A, B: Entradas de datos.

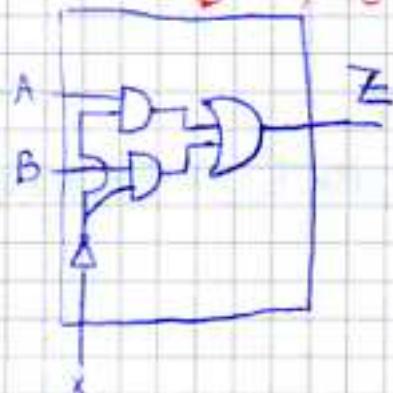
X: Entrada de control.

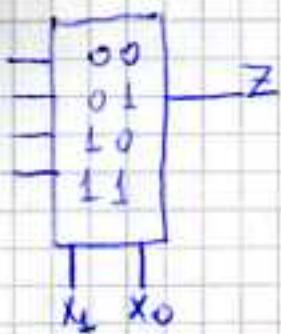
X \ A \ B \ Z	0000	0010	0100	0111	1000	1011	1100	1111
0000	0				0			
0010		0				0		
0100			0				0	
0111				1				
1000					0			
1011						1		
1100							0	
1111								1

$$Z = \bar{X} \cdot A + B$$

Dentro de este multiplexor hay este.

X \ AB	00	01	11	10
0000	0	1	1	0
1010	1	1	0	0

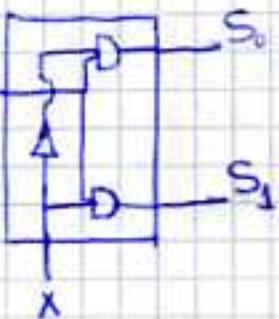




~~Entrada de datos~~
control

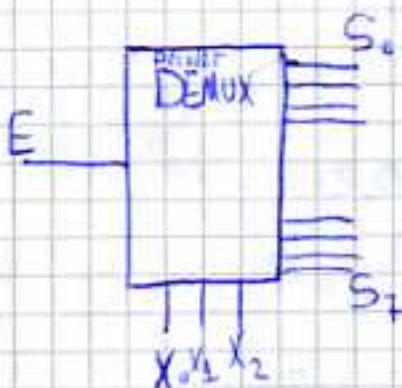
2 = Entrada de datos

~~Demultiplexor:~~



X	E	S ₀	S ₁
0	0	0	0
0	1	0	0
1	0	0	0
1	1	0	1

$$\begin{cases} S_0 = \bar{X} \cdot E \\ S_1 = X \cdot E \end{cases}$$



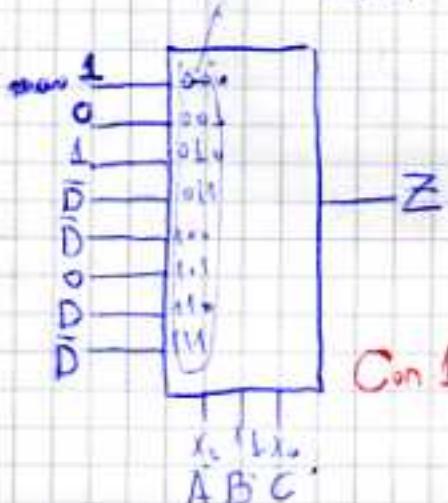
X ₃	X ₂	X ₁	S ₀	S ₁	S ₂
0	0	0	E	0	0
0	0	1	0	E	0
0	1	0	0	0	E
0	1	1	0	0	0
1	0	0	0	0	0
1	1	0	0	0	0
1	1	1	0	0	E

* Decodificador tiene el mismo circuito que demultiplexor.

* Implementar función el con multiplexores:

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

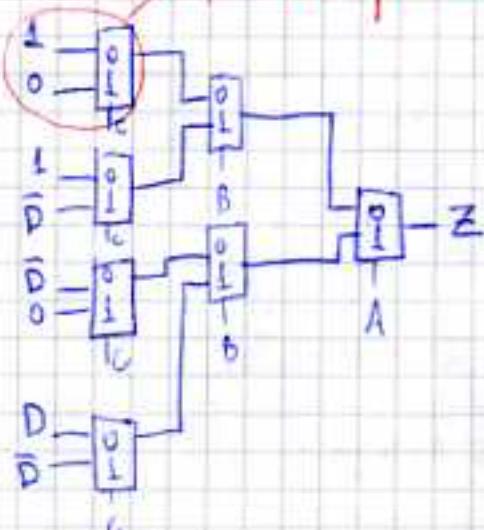
Función de D.



Con 1 multiplicador.

* Implementación en árbol:

Equivalente a C, pero si entra un 1, devolverá 0 y si entra un 0, devolverá 1.



29/4

$$\begin{array}{r}
 \text{Carry} \\
 \text{a}_1 \text{ a}_0 \text{ c}_1 \\
 \text{---} \\
 \text{b}_1 \text{ b}_0 \text{ c}_0 \\
 \text{---} \\
 \text{d}_1 \text{ d}_0 \text{ c}_1 \text{ c}_0
 \end{array}$$

* Hay Carry (Carry Summarization).

$$b_1 = a_1 \oplus b_1 \oplus c_1$$

$$b_0 = a_0 \oplus b_0 \oplus c_0$$

$$g_i = d_i \cdot b_i \rightarrow \text{Función del carry, } d_i \text{ y } b_i$$

* Propagación:

$$\begin{aligned}
 p_i &= d_i \cdot b_i, C_{i,\text{IN}} \rightarrow \text{Función de propagación.} \\
 &\quad \text{a/carry: } (d_i \cdot b_i) \text{ y } C_{i,\text{IN}}
 \end{aligned}$$

• En el menor peso, solo para haber generación de carry.

• El Carryout $C_{\text{out}} = C_{\text{out}} + g_i + p_i \cdot C_i$

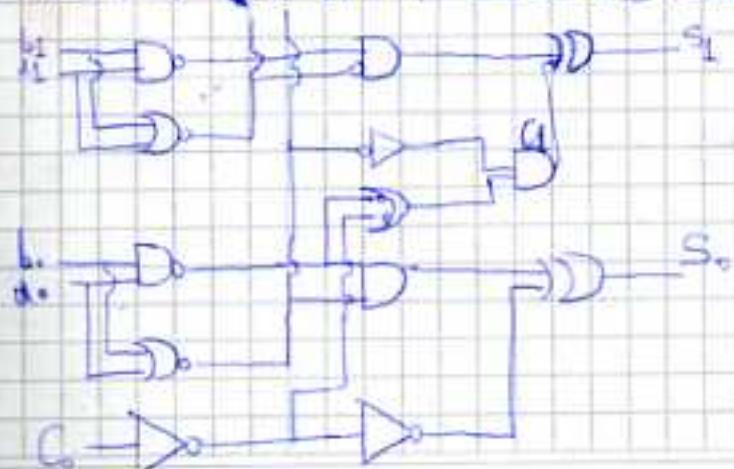
$$C_{\text{out}} = a_1 \cdot b_1 + (a_1 \oplus b_1) \cdot d_0 \cdot b_0$$

$$C_{\text{out}} = a_1 \cdot b_1 + (a_1 \oplus b_1) \cdot [a_1 \cdot b_1 + (a_1 \oplus b_1) \cdot d_0 \cdot b_0]$$

$$C_{\text{out}} = a_1 \cdot b_1 + (a_1 \oplus b_1) \cdot [d_1 \cdot b_1 + (a_1 \oplus b_1) \cdot (d_1 \cdot b_1 + (a_1 \oplus b_1) \cdot d_0 \cdot b_0)]$$

* Sumador del tipo Carry Look AHEAD:

Circuito 74283 (VERIFICAR SI CORRESPONDE A ESE)



+ Obras Civiles:

TILSIT

* R estadores:

$$\begin{array}{c} \text{Invert} \\ \text{I} \\ \text{U} \\ \text{O} \\ \text{O} \\ \text{L} \\ \text{L} \end{array} \quad \begin{array}{c} \text{O} \\ \text{O} \\ \text{O} \\ \text{I} \\ \text{I} \\ \text{L} \\ \text{L} \end{array}$$

Binaria: cada vez un número más redondeado, hasta por ceros.

(Ceros).

a	b	c	Binario
0	0	0	000
0	1	1	111
1	0	1	101
1	1	0	110
1	1	1	111

$$B = a.b + 0.b$$

$$B_{binario} = 101$$

~~a/b/B/B_n~~

0	0	0	0
0	1	0	1
1	1	0	1
1	0	1	0
1	0	1	0
1	1	0	0
1	1	1	1

$$\begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{array}$$

a	b	c	
0	0	0	0
0	1	0	1
1	1	0	1
1	0	1	0
1	0	1	0
1	1	0	0
1	1	1	1

a	b	c	
0	0	0	0
0	1	1	1
1	1	0	1
1	0	0	0
1	0	0	0

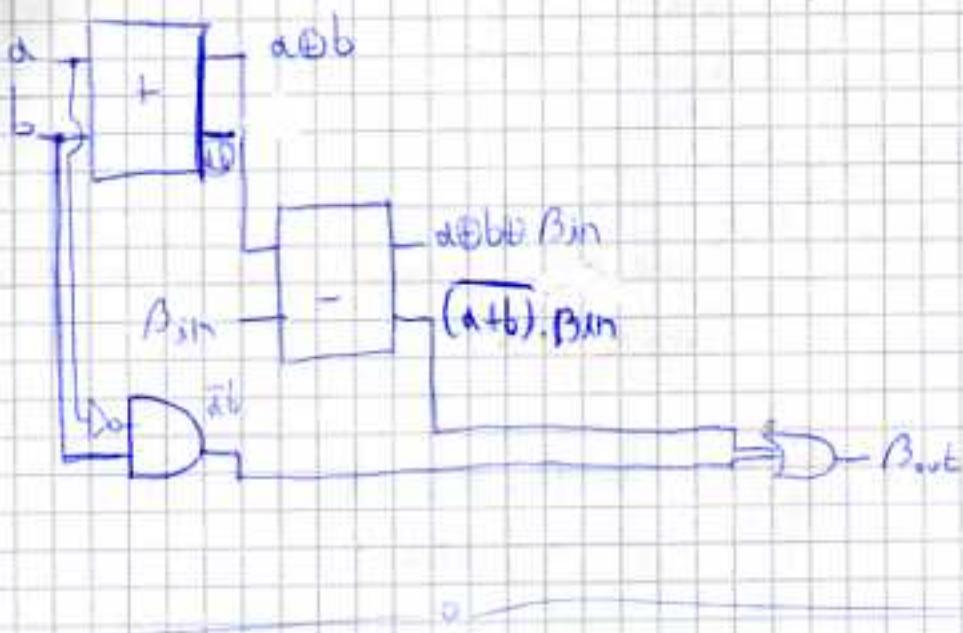
R_{redondo}B_{red}

$$B = a \oplus b \oplus B_{n-1}$$

$$B_{n-1} \rightarrow 1 \oplus B_{n-2} \oplus B_{n-3}$$

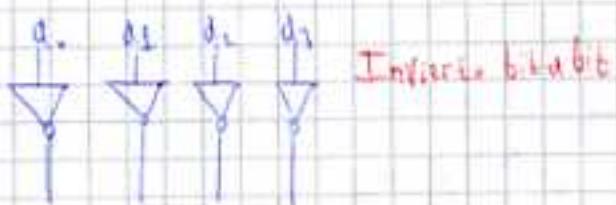
10

$A \oplus B$	B_{out}
0 0 0	0 0
0 0 1	1 1
0 1 0	1 0
0 1 1	0 1

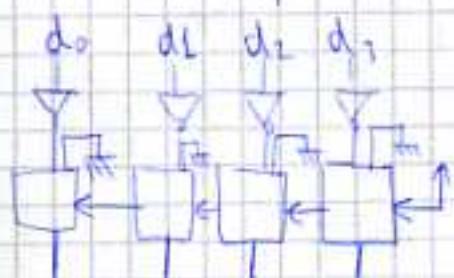


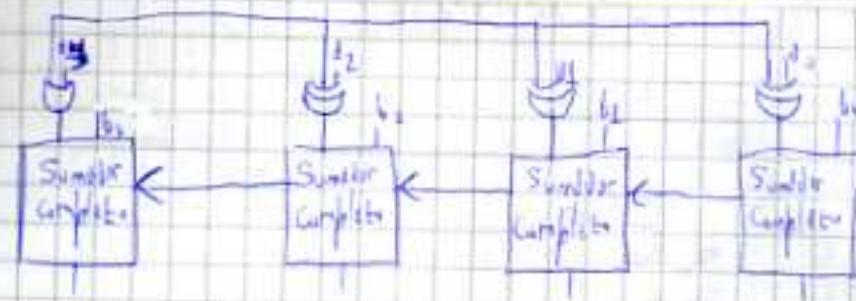
* Circuitos Combinacionales:

Para calcular complemento de $a+b = -1$:



Para calcular el complemento de $a+b$:





* Comparadores

a	b	$a = b$	$a > b$	$a < b$	$a \geq b$	$a \leq b$
0	0	1	0	0	1	1
0	1	0	0	1	0	1
1	0	0	1	0	1	0
1	1	1	0	0	1	1

$$F(a=b) = a \oplus b$$

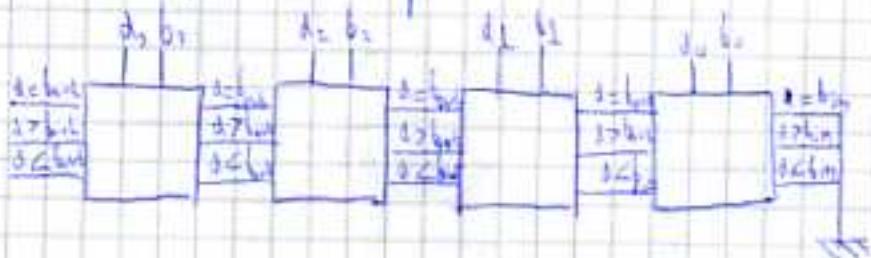
$$F(a > b) = a \cdot \bar{b}$$

$$F(a < b) = \bar{a} \cdot b$$

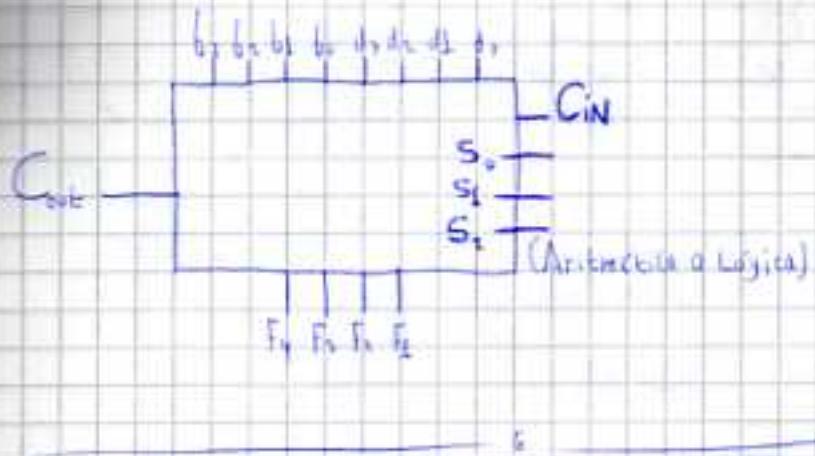
$$F(a \geq b) = a + \bar{b}$$

$$F(a \leq b) = \bar{a} + b$$

• Si queremos comparar muchos paralelos de bits:



* ALU (Unidad Aritmética Lógica)



22/4

* UAL

Unidade Aritmética → SUMADOR
(RESTADOR)
COMPARADOR
DE MAGNITUD

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = A \oplus B$
 $C = A \cdot B$



HALF ADDER
Sum de 2 NÚM
de 1 BIT C/U

$$\begin{array}{r} 111 \\ + 101 \\ \hline 1000 \end{array}$$

A _i	B _i	C _{i-1}	S _i	C _i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

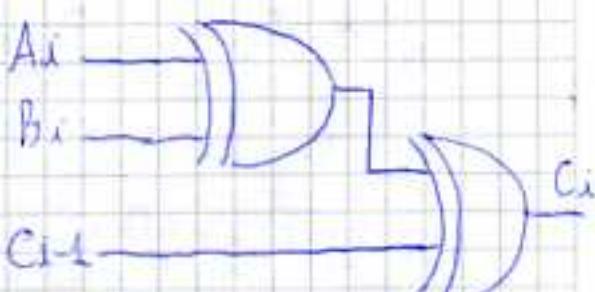


-1
 10
 11
 +10

FULL ADDER ó SUM. DE 3 NOS DE 1BIT C/U

A _i	B _i	01	11	10
0	0	1	0	1
1	1	0	1	0

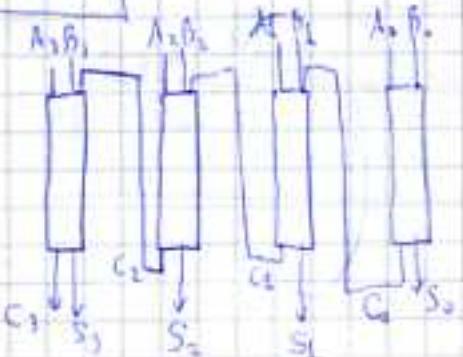
A _i	B _i	01	11	10
0	0	0	1	0
1	0	1	1	1



Sumador Serie:

$$C_i = A_i \cdot B_i + A_i \cdot C_{i-1} + B_i \cdot C_{i-1}$$

$$\begin{aligned}
 & A_3 \ A_2 \ A_1 \ A_0 \\
 & + B_3 \ B_2 \ B_1 \ B_0 \\
 & \hline
 & C_3 \ S_3 \ S_2 \ S_1 \ S_0
 \end{aligned}$$



A mds contiene
de bits durante
el tiempo del
ciclo.

*Carry Look Ahead

$$C_i = \underbrace{A_i \cdot B_i}_{S_{\text{columna}}} + \underbrace{C_{i-1} \cdot (A_i + B_i)}_{P_i}$$

S: columna

g_i , porque

S: suma

sum,

genera sum,

C_{i-1} .

S: columna

P_i , por

propagación

de carry.

$$C_i = g_i + C_{i-1} \cdot P_i$$

. Para el Carry gen.

$$C_0 = g_0 + P_0 \cdot C_{-1}$$

. Parallel 1:

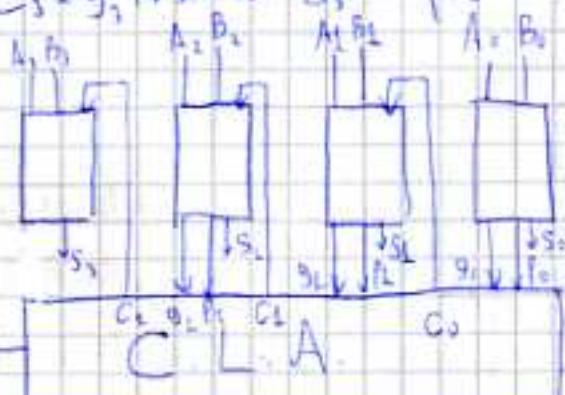
$$C_1 = g_1 + P_1 \cdot C_0 = g_1 + P_1 \cdot (g_0 + P_0 \cdot C_{-1})$$

. Parallel 2:

$$C_2 = g_2 + P_2 \cdot C_1 = g_2 + P_2 \cdot [g_1 + P_1 \cdot (g_0 + P_0 \cdot C_{-1})]$$

. Parallel 3:

$$C_3 = g_3 + P_3 \cdot C_2 = g_3 + P_3 \cdot \{ g_2 + P_2 \cdot [g_1 + P_1 \cdot (g_0 + P_0 \cdot C_{-1})] \}$$



* Restador

* SEMI-RESTADOR. RESTANDO DE 2^{NO} DE 1BIT C/U

A_i	B_i	R_i	B_{wi}
0 0	0 0		
0 1	1 1		
1 0	1 0		
1 1	0 0		

$$R_i = A_i \oplus B_i$$

$$B_{wi} = \overline{A_i} \cdot B_i$$

- $A_3\ A_2\ A_1\ A_0$
 $B_3\ B_2\ B_1\ B_0$

* RESTADOR

RESTADOR DE 3 - N^o DE 1BIT C/U

A_i	B_i	B_{wi-1}	R_i	B_{wi}
0 0 0	0 0 0	0 0		
0 0 1	1 1 1	1 1		
0 1 0	1 1 0	1 1		
0 1 1	0 1 1	0 1		
1 0 0	1 0 0	1 0		
1 0 1	0 0 0	0 0		
1 1 0	0 0 0	0 0		
1 1 1	1 1 1	1 1		

A_i	B_{wi-1}	R_i	B_{wi}
0 0	0	0	0
1 0	1	1	1
1 1	0	0	0

$$B_{wi} = \overline{A_i} \cdot B_{xi} + \\ \overline{A_i} \cdot B_{wi-1} + \\ B_i \cdot B_{wi-1}$$

$$\begin{array}{r} 10 \\ - 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 101 \\ - 10 \\ \hline 10 \end{array}$$

* Comparador de magnitud.

• 2^N^{es} de 1 BIT:

A	B	$S_{A \leq B}$	$S_{A \geq B}$	$S_{A=B}$	$S_{A > B}$	$S_{A >= B}$	$S_{A \neq B}$
0	0	1	0	1	0	0	0
0	1	0	1	0	1	0	1
1	0	1	0	0	1	1	0
1	1	0	1	1	0	0	1

$$S_{A \leq B} = S_{A \leq B} + S_{A=B}$$

~~$S_{A \geq B} = \overline{S_{A \leq B}}$~~

~~$S_{A \neq B} = \overline{S_{A=B}}$~~

~~$S_{A > B} = \overline{S_{A \leq B}}$~~

$$S_{A \leq B} = \overline{A \cdot B} \quad | \text{ El comparador}$$

$S_{A=B} = \overline{A \oplus B}$ | Hace esto para
que A sea 1 y de
esta forma obtener
dicho comparador

• 2^N^{es} de 2 BITS:

A_1	A_0	B_1	B_0	$S_{A \leq B}$	$S_{A=B}$
0	0	0	0	1	0
0	0	0	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
1	0	0	0	0	1
1	0	0	1	1	0
1	1	0	0	0	1
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	0	1

• 2^{16} N° de 4 BE:

A₃, A₂, A₁, A₀

B₃, B₂, B₁, B₀

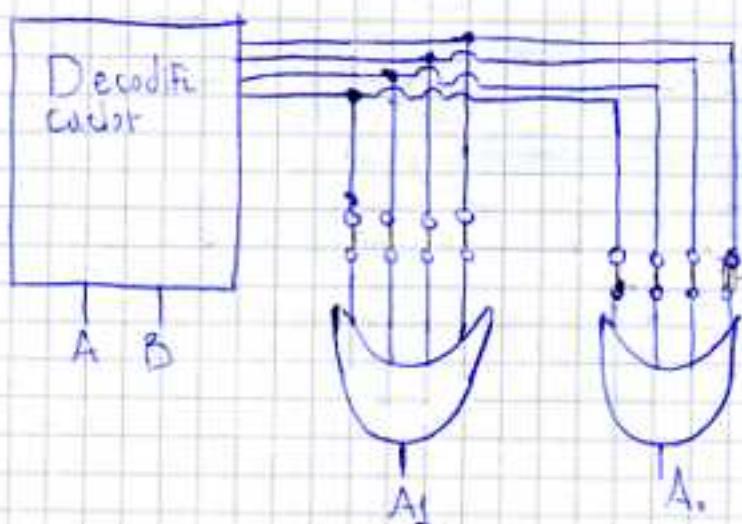
$$S_{A=B} = (\overline{A_3 \oplus B_3}) \cdot (\overline{A_2 \oplus B_2}) \cdot (\overline{A_1 \oplus B_1}) \cdot (\overline{A_0 \oplus B_0})$$

$$S_{A < B} = (\overline{A_3 \cdot B_3}) + (\overline{A_3 \oplus B_2}) \cdot (\overline{A_2 \cdot B_2}) + (\overline{A_3 \oplus B_1}) \cdot (\overline{A_1 \cdot B_1}) + (\overline{A_3 \oplus B_0}) \cdot (\overline{A_2 \oplus B_2}) + (\overline{A_2 \oplus B_1}) \cdot (\overline{A_1 \oplus B_1}) + (\overline{A_1 \oplus B_0}) \cdot (\overline{A_0 \cdot B_0})$$

27/4

* PLD (Programmable Logic Devices).

• ROM: Read Only Memory (Memoria de solo lectura).

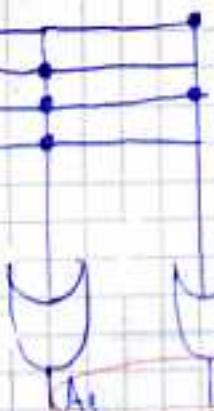


Fusibles, que se queman, y la memoria luego no puede escribirse más.



D·Co

(A) (B)



También se simboliza así.

La combinación de los salidas se llama palabra.

La combinación de las entradas se llaman direcciones.

* EEPROM - Memoria que puede borrar y reprogramarse

* EEPROM - Memoria que puede borrar y reprogramarse eléctricamente

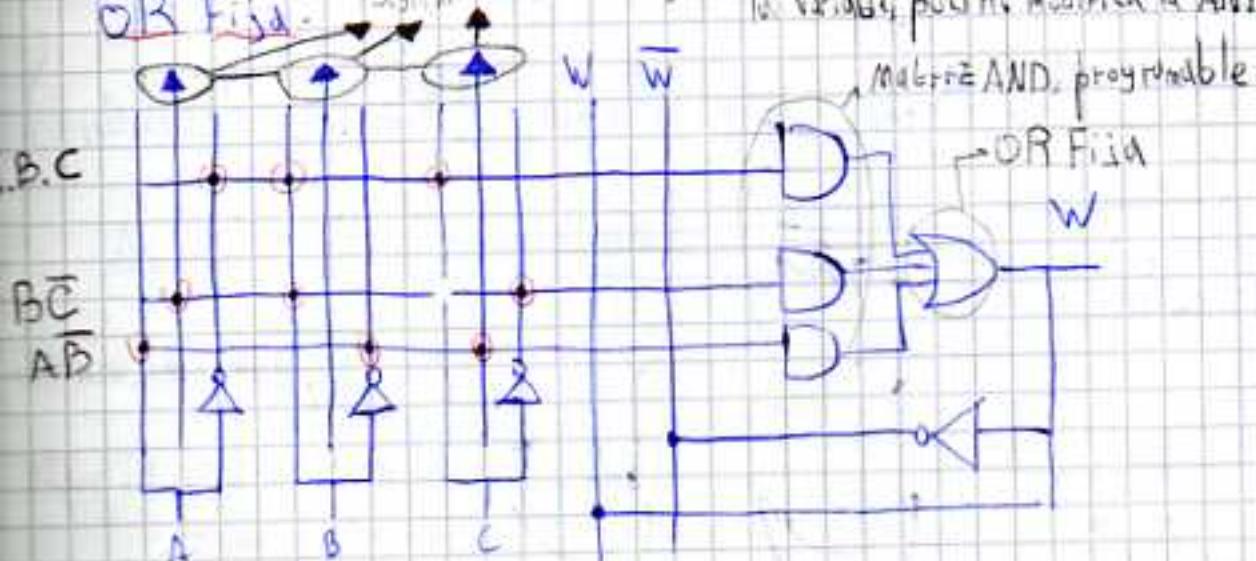
* PAL (Programmable Array Logic)

No tiene que ver con la blue bin

• Se trata de una matriz AND, programable

Y linea

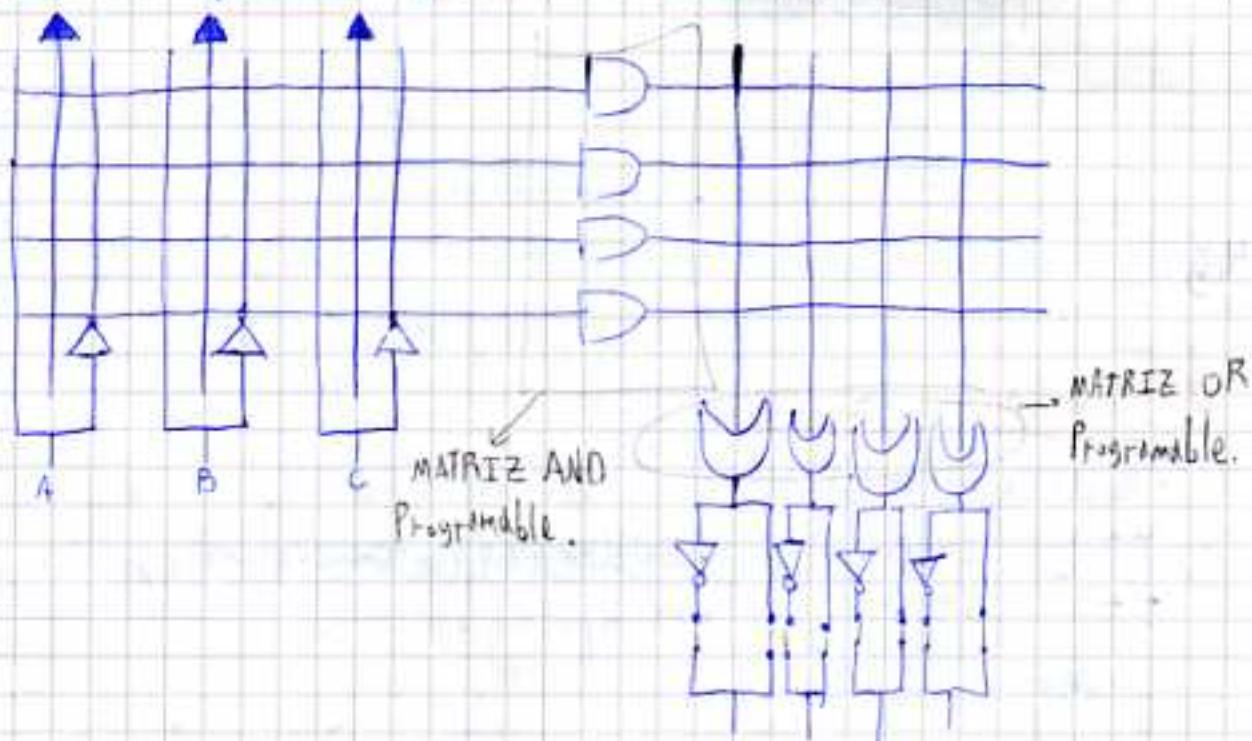
OR Fija - Significa 1 en el lugar de las variables, para no modificar la AND.



* GAL, Generic Array Logic: Son las que se usan ahora, son reprogramables.

* PLA: Programable logic Array.

- Matriz AND programable y matriz OR también programable.



* Circuito programable, viene programado de fábrica.

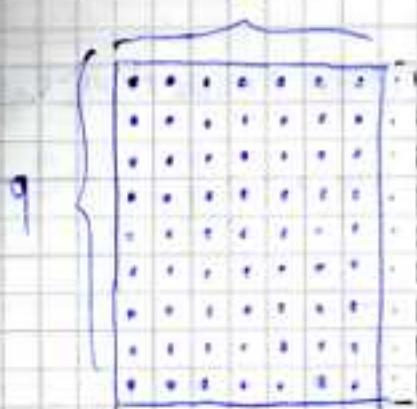
* Circuito programable, puede programarse.

Ej 7:

→ Armar una función para cada uno e implementar en F16, PAL y PLA.



Ej 8:



8 bits \rightarrow 256

$\text{En este caso: } 9 \times 8 = 256 \rightarrow \text{(Contado)}$

de casillas

Altura

Ancho

polilla

cont. bits.

Debe imprimir 256 caracteres.

45

* Familias Lógicas

• Series de integrados:

① TTL 7400
TTL 5400

- Se busca que el consumo sea lo mejor posible.
- Es importante la velocidad.

Transistor

Transistor

Logic

TTL 74H00; TTL de bajo consumo.

TTL 74L00; TTL de bajo consumo.

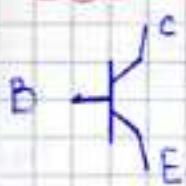
② CMOS:

Consumen menos pero son más lentas. Con el tiempo se hacen más rápidas.

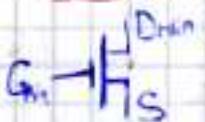
CMOS = MOS Complementario

HCmos \rightarrow CMOS de alta velocidad.

* Transistor TTL:



* Transistor CMOS:



La pata Source tiene una tensión de trabajo.

Tensión VGS.



* El uso del Transistor es como sigue:

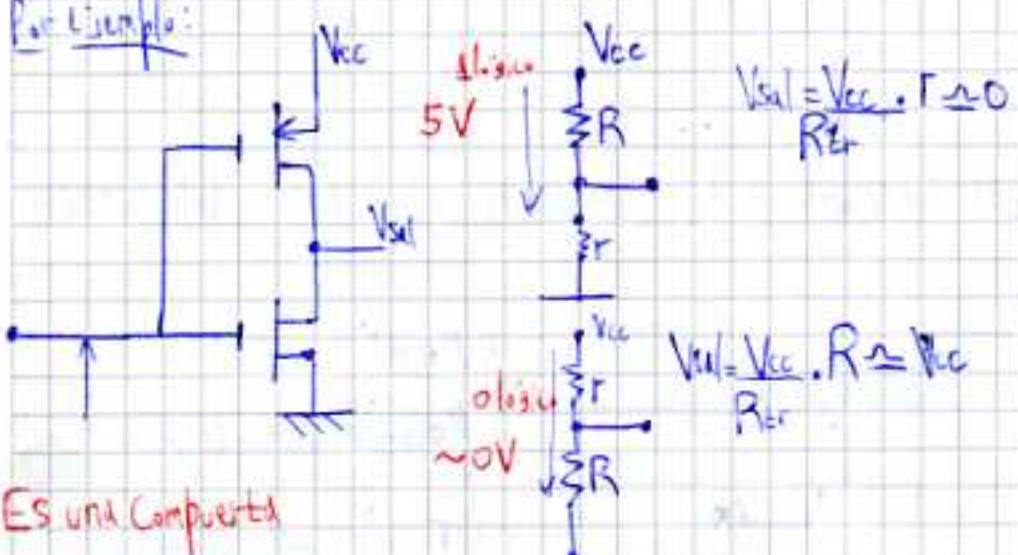


Dependiendo de la tensión Señal que se aplica. Para esto considera la fuente como una resistencia variable:

S. $V_{GS} > 0 \Rightarrow r_{ds} \approx 0$, fuerte abierto

S. $V_{GS} \approx 0 \Rightarrow r_{ds} \gg 1 \text{ M} \Omega$, fuerte cerrado.

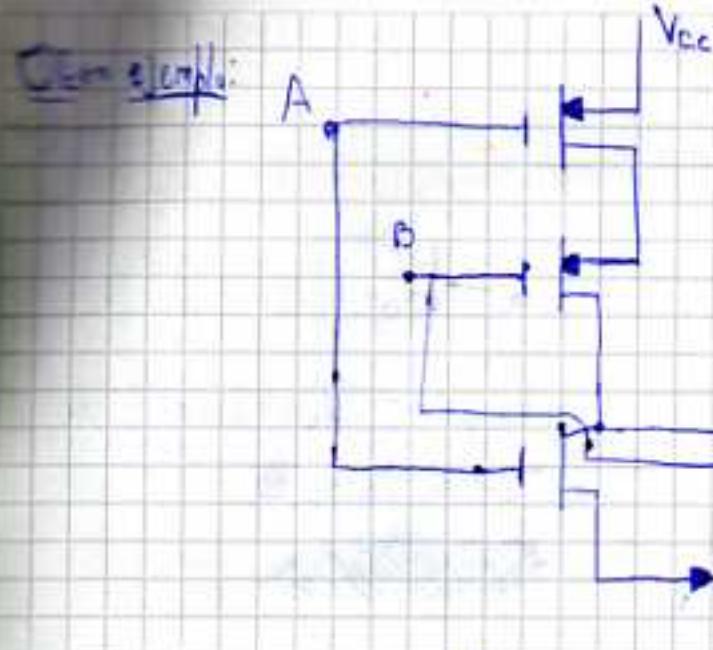
Por ejemplo:



Es una Compuerta

CMOS Invertida:



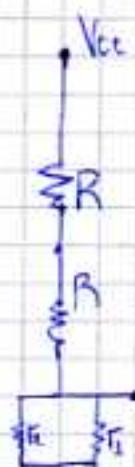


$A \rightarrow 1 \quad \text{Sale un "0"}$

$B \rightarrow 1$

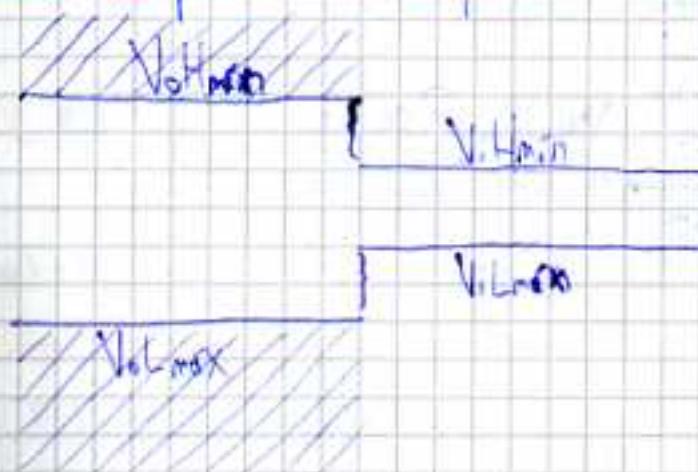
$A \rightarrow 1 \quad \left\{ \begin{array}{l} \text{Sal} \rightarrow 0 \\ B \rightarrow 0 \end{array} \right.$

$A \rightarrow 0 \quad \left\{ \begin{array}{l} \text{Sal} \rightarrow 1 \\ B \rightarrow 0 \end{array} \right.$



Compuerta NOR, Cmos.

* Las Compuertas se Comportan bien de acuerdo a la señal que tienen.



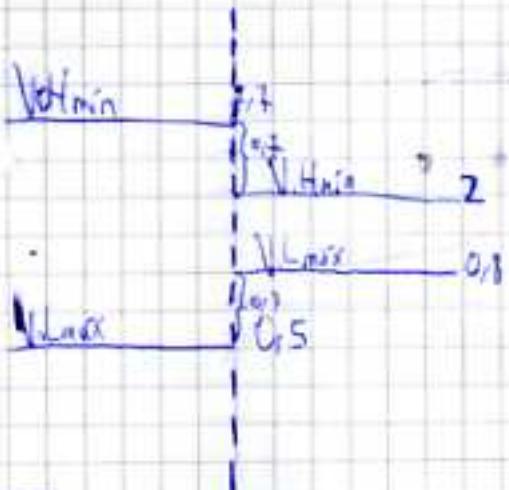
* Niveles de Ruido:

$$M_{RH} = V_{Hmax} - V_{Hmin}$$

$$M_{RL} = V_{Lmax} - V_{Lmin}$$

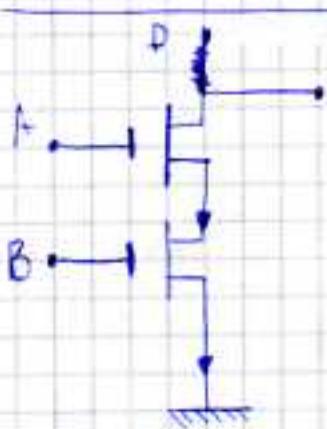
Ej:

	74LS00	4011
V _{Hmin}	2,4	4,4
V _{Lmax}	0,5	0,1
V _{Hmin}	2	3,45
V _{Lmax}	0,8	1,35



*DIP

*Dual in line package

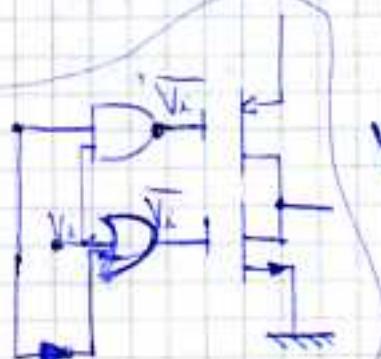


*OPEN DRAIN

Compuerta tri-state:

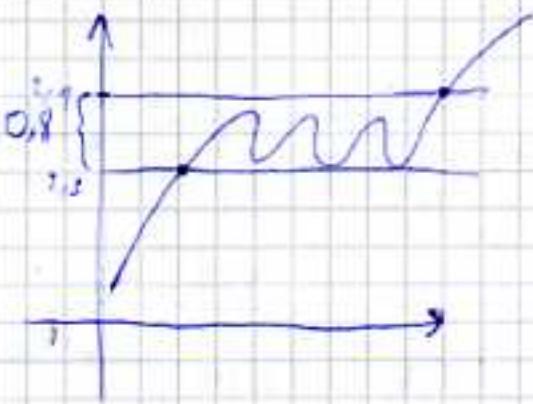
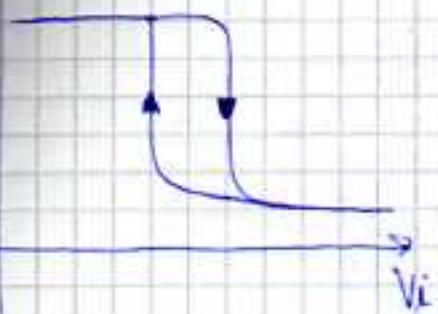


EA

 $V_i = 0 \rightarrow V_{SD} = 0$ 

* Comportamiento de histeresis (cond. hay ruido):

No.



* Número Fan Out:

(número de componentes que un solo dispositivo controla).

Caso difícil: $\frac{I_{out}}{I_{in}}$ (una salida).

Caso fácil: $\frac{I_{out\min}}{I_{in\min}}$

* Número Fan In:

Dato que el fabricante da para Verona la tecnología NMOS de 1.8 voltios.

Ej3

Siempre se resuelve.

Amboos son típicos.

* Comportamientos detectivos: tienen una puerta para la discriminación.

* Comportamiento polivalente: No tienen discriminación.

Todos los comportamientos deben estar definidos, de otra forma funcionarán mal.

(2)

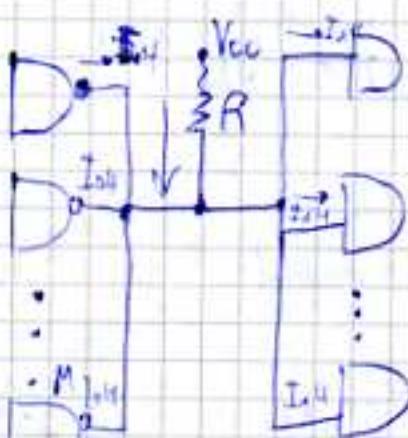
Pull-up: Resistencia alta > Load R

(3)

Pull-down: Resistor baja < load

4) Tri-state: una sola linea controlada a Vcc o GND.
 Ofrece la misma resistencia de Pull-up.

(4)

***Proyecto:**

* Comprar Testar y Soldador.

* Alambres de Colores para mejor comprensión del circuito

* Se buelva probando en un simulador decíduo.

* 4 Puentes de Control

* 2 displays: Número de mesa y número de orden

* Se recomienda comprar el CMOS.

* **[ES INDIVIDUAL]***** V.A Con informe**

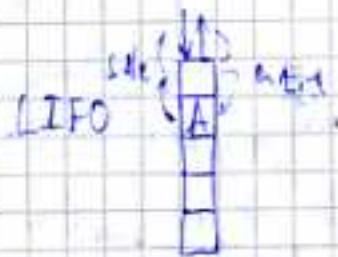
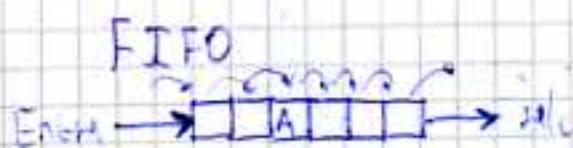
- Objetivo
- Diagrama de Flujo
- Dificultades y como las superan
- Circuito lógico
- Conclusión.

6/5

~~Memoria~~

* Memoria de acceso: El tiempo de acceso es el tiempo que tarda en traer el dato. El tiempo secuencial. La memoria entra en el programar.

de acceso depende de la posición. Se llama memoria FIFO o Registro de desplazamiento. A diferencia de la memoria LIFO o Stack, que el último en entrar es el primero en salir.



* Memoria de acceso aleatorio: El tiempo de acceso NO depende de la ubicación. Memorias RAM.

① Memoria de lectura escritura. ② Memoria ROM, solo lectura. La ROM también es RAM.

③ Memoria, PLD's. Son
Roms programables

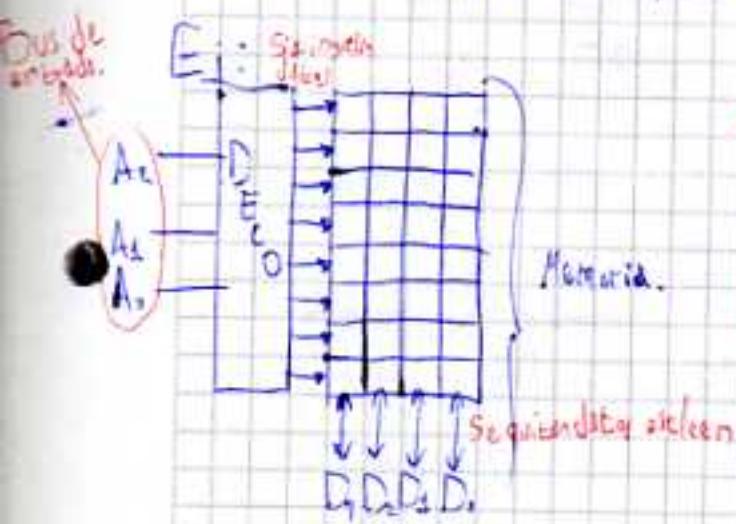
1. Memoria Volátil: Cuando se le saca la alimentación pierden los datos. 2 Tipos:

- RAM dinámica: El almacenamiento se da en pequeños capacitores que guardan 1 o 0. Sin más cambios. Es un trabajo lento.

- RAM estática: Claro, es más complicada, más estable, con encendido y apagado. Muy rápida pero más caras. Es un Flip-Flop.

1. ROM

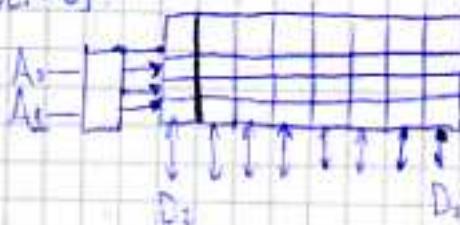
2. Tienen programación básica de una computadora. Es como una tabla de verdad almacenada. Puede programarse solo en fábrica. La **PROM** puede ser programada por el usuario, pero, luego, si se quiere reprogramar se necesita sobre PROM. La **EPROM** puede reprogramarse, y para borrar se hace por radiación ultravioleta. La **EEPROM** puede reprogramarse por el electricidad. La **Mosca de Flait** puede borrarle por bloques.

• Las tablas de verdad organizadas en bloques.

* Selección de salidas: 2^n salidas, con n líneas de entrada (Bus de direcciones)

* Capacidad de memoria: $2^n \times d$ celdas fijas.

Otro ej.:



→ Se puede ver como una tabla de verdad de 2 entradas y 8 funciones.

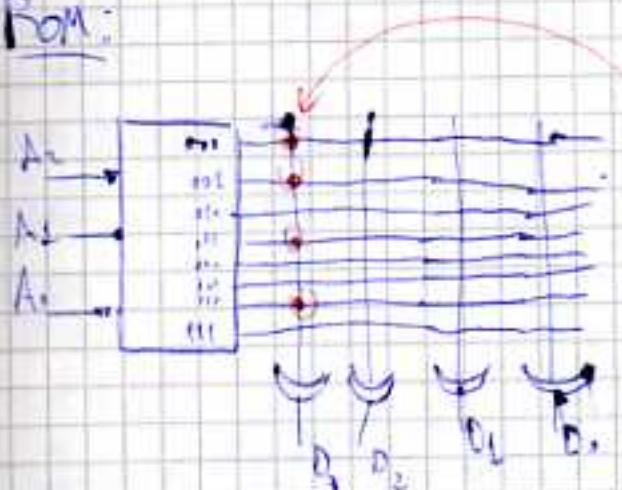
• A mayor ancho de bus, mayor anchos de banda.

PLD's:

~~Programmable logic devices~~, de niveles AND/OR.

Suma de productos minterminos

ROM:



- Si queremos calcular:
 $Z = \sum m(1, 3, 5)$, se conectan en orbitas
y para programar se calcan fuses
se funden.



PLA:

El arreglo de ANDS es programable. Se tratan funciones más simples, por lo que
tienen menor ancho.

* En PLA se programan los Arreglos de AND Y OR.

	AND	OR
PRoM	Fijo	programable
FAL	programable	Fijo
PLA	programable	programable

* PAL:

Se programan máx. 2 AND, pero fija la OR. Son más sencillas de programar y más baratas. También hay que simplificar.

Ex:

11/5. Flip-Flop RS

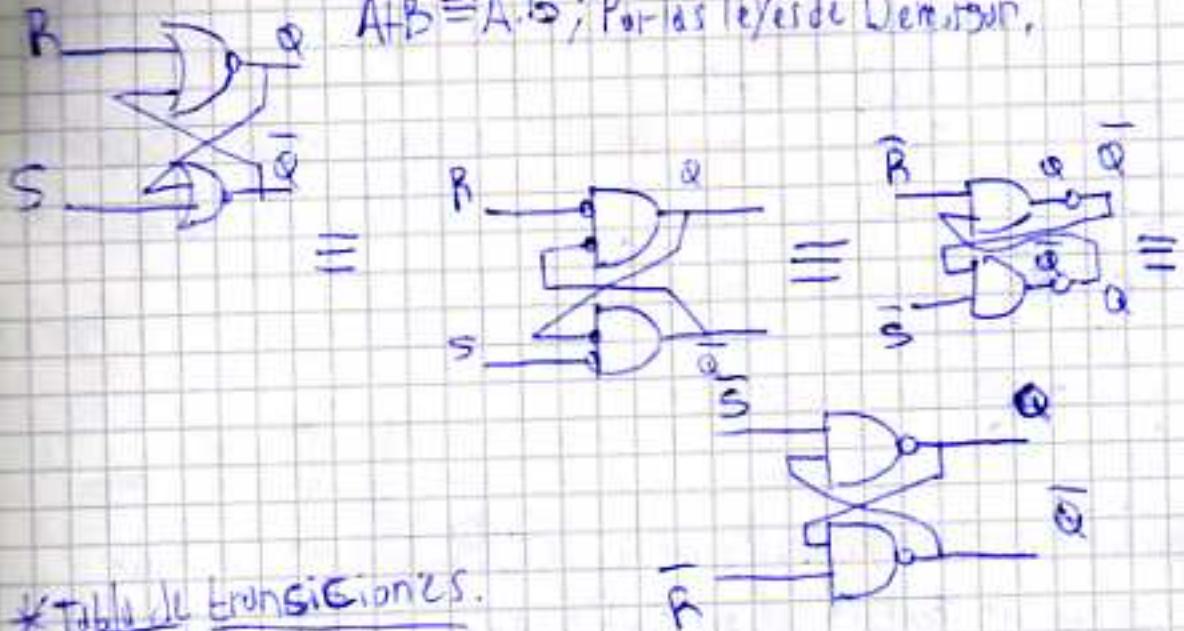
* Tabla de estados ampliada.

R	S	D	Q ⁿ⁺¹
0	0	0	Q ⁿ
0	1	0	Q ⁿ
1	0	1	1
1	1	0	0
1	1	1	-

* Tabla de estados reducida.

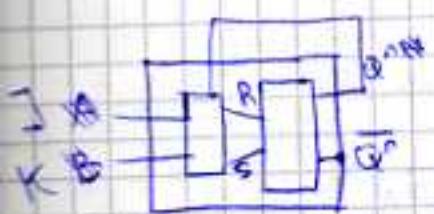
R	S	D	Q ⁿ⁺¹
0	0	0	Q ⁿ
0	1	1	1
1	0	0	0
1	1	-	-

$$\overline{A+B} = \overline{A} \cdot \overline{B}; \text{ Por las leyes de De Morgan.}$$



* Tabla de transiciones.

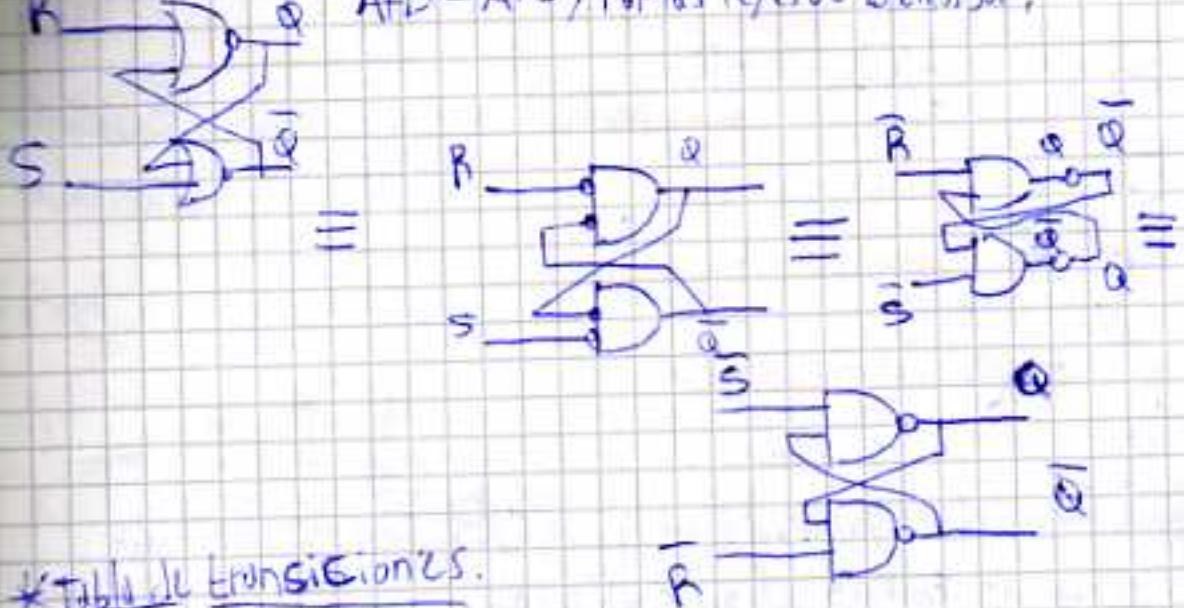
Q^n	Q^{n+1}	$R S$
0 0	x 0	
0 1	0 1	
1 0	1 0	
1 1	0 x	



J	K	I	S	R	S'
0	0	0	0	x	0
0	0	1	1	0	x
0	1	0	0	0	x
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	1	1	0
1	1	0	1	0	x
1	1	1	0	1	0

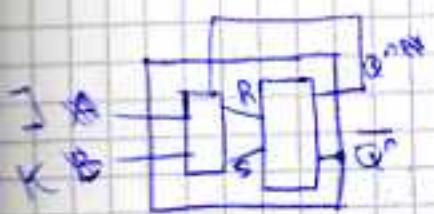
- Quiero hallar:
- $R = R(J, K, Q^n)$
- $S = S(J, K, Q^n)$
- Hago Karnaugh:

$$\overline{A+B} = \overline{A} \cdot \overline{B}; \text{ Por las leyes de De Morgan.}$$



* Tableau de Transiciones.

Q^n	Q^{n+1}	RS
0 0	x 0	
0 1	0 1	
1 0	1 0	
1 1	0 x	



J	K	R	S	RS
0 0	0 0	x 0		
0 0	1 1	0 x		
0 1	0 0	x 0		
0 1	1 0	0 0		
1 0	0 0	0 1		
1 0	1 1	0 x		
1 1	0 0	0 1		
1 1	1 0	1 0		

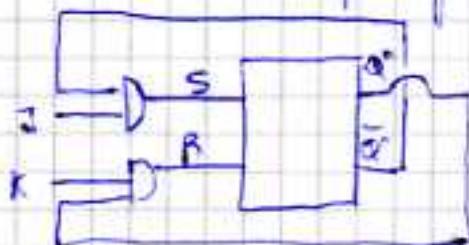
- Quiero hallar:
- $R = R(J, K, Q^n)$
- $S = S(J, K, Q^n)$
- Todos los estados:

J	K	Q^n
0	0	0
0	1	1
1	0	1
1	1	X

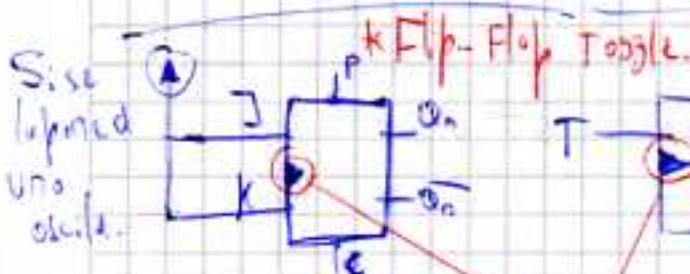
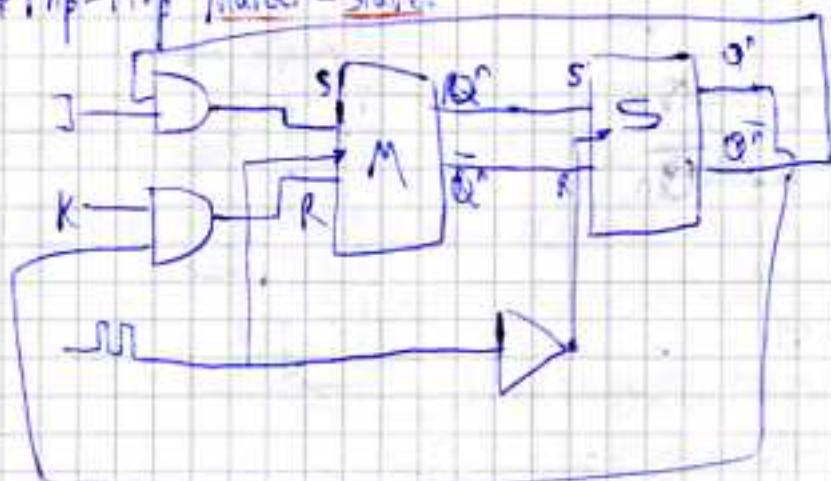
J	K	Q^n
0	0	0
0	1	0
1	0	0
1	1	0

$$R = K Q^n \quad S = J \bar{Q}_n$$

Entonces, el Flip-Flop resultante es:



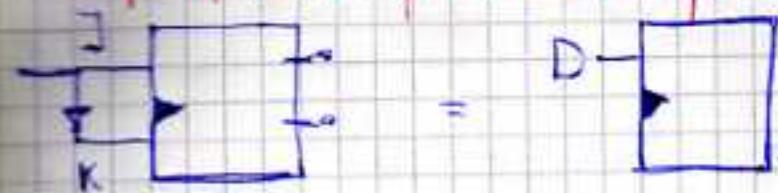
Flip-Flop Master-Slave:



T	Q^n	Q^{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

$$Q^{n+1} = \bar{T} \cdot Q^n + T \cdot \bar{Q}^n$$

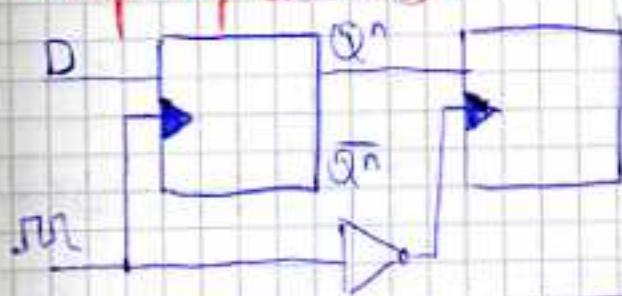
→ Flip-Flop delay: rechte (ausgetragene) Zeitpunkte der Umlade.



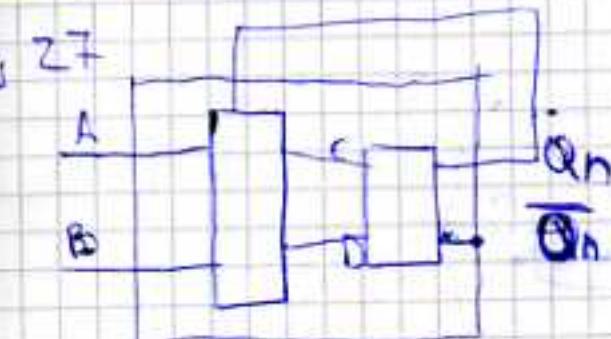
D	Q^n	Q^{n+1}
0	0	0
0	0	1
1	0	1
1	1	1

$$Q^{n+1} = D$$

→ Flip-Flop edge-Trigger.



Ej 27



• Tabla auxiliar.

C	D	Q^n	Q^{n+1}
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Siguiente

S. Q^n+1st values 0, 1
by definition 10, 01.

Unmittelbar herabsetzen

oder XX weitergereicht
00, 01, 10, 11, 1, 0000

oder XX weitergereicht
00, 01, 10, 11, 1, 0000

A	B	Q^n	Q^{n+1}	C	D
0	0	0	XX		
0	0	1	00		
0	1	0	00		
0	1	1	01		
1	0	0	01		
1	0	1	00		
1	1	0	00		
1	1	1	1d		

<u>Op. Bin.</u>	<u>CD</u>
0 0	X X
0 1	X b
1 0	b X
1 1	C P

* Se tienen 2 Karnaugh, que quieren la función de los resultados, por igualmente se llega a la misma función.

* Repaso parcial.

El resultado del 4 al 5 bit se llama half carry.

• Suma de binarios y Flags:

$$\begin{array}{r}
 \text{Dado:} \\
 \begin{array}{r}
 01101010 \\
 + 01101010 \\
 \hline
 10111110
 \end{array}
 \end{array}$$

$$\begin{array}{l}
 C=0 \\
 V=1 \\
 Z=0 \\
 S=1
 \end{array}$$

(Sirve para trabajar con bolas positivas)

Carry: No forma parte del número, es el que llevan los dígitos de más. Significativo.

Overflow: Cuando el bit Carry, es el bit más significativo es distinto del Carry que el bit más significativo.
(Sirve para trabajar con números con signo)

S, C = 1, Es Overflow, en positivo.

S, V = 1, Es Overflow con signo.

Zero: Vale 1, si todos los bits son 0, menos el Carry.

Signo: Si una significatividad es 1, vale 1.

Punto decimal: Vale 1, si la cantidad de ceros es impar, y 0, si es par.

base:

$$\begin{array}{r} 666 \\ \hline 9 \end{array}$$

Como $9 = 3^2$, basta expresar cada cifra como 2^k de la base 3.

$$\begin{array}{r} 6 \quad | \quad 6 \quad | \quad 6 \\ \hline 2 \quad 0 \quad 2 \quad 0 \quad 2 \quad 0 \end{array}$$

$$\begin{array}{r} 666 \\ \hline 9 \quad 3 \end{array} = 202020$$

Complemento a la base:

$$6 = 842121$$

$$C_1(0110) = C_1(0110) + 1 = 100111 = 1010$$

Completa el resultado usando los métodos propuestos.

Ejemplo:

$$4523 \rightarrow \begin{array}{r} 10^{-23} \\ 10 \quad 3 \end{array}$$

$$\begin{array}{r} 4523 \\ 9 \quad 45 \quad 3 \\ 9 \quad 5 \quad 1 \quad 3 \\ 4 \quad 1 \quad 0 \end{array} \Rightarrow 45 \leftarrow 1200 \begin{array}{r} 10^{-23} \\ 10 \quad 3 \end{array}$$

$0_{123} \cdot 3 = 0,69$ → La cantidad de veces que
 $0,69 \cdot 3 = 2,07$ → mult. por 3
 $0,1 \cdot 3 = 0,21$ → es el error:
 $0_{123} = 0,0201$ → b^{-n} , n cantidad de veces
 $\begin{array}{r} 10^{-23} \\ 10 \quad 3 \end{array}$ → multiplicar

$$3^{-n} < 10^{-3}$$

$$-n \log_{10}(3) < -3$$

$$\boxed{n=7}$$

Ej 4.

$b_1^{m+1} = 1$; Máximo número de b_1

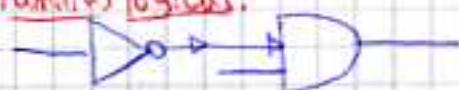
b_2

$$\boxed{b_2^m \geq b_1^n}$$

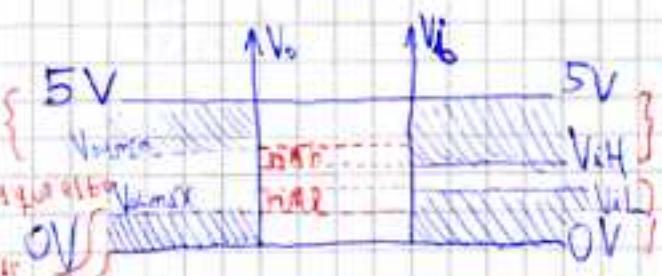
13/5

Aula 201, parcial teórica

*Familias lógicas.



No puede
dejarse en
ningún nivel que esté
entre los
dos niveles
que tiene.



5V } No puede dejarlo en ningún punto

ViH

0V } No puede dejarlo en ningún punto.

• $nMh = mJg$ donde

g es doble.

• $nMl = mJg$ donde $|S_{in}|$

Integrados que pueden interactuar entre sí sin problemas.

- Lógica positiva el nVd resulta en el l.
- Interface adapta niveles de tensión entre integrados.

$$\begin{aligned} * N_AH = V_{OH\min} - V_{IH} \\ * N_AL = V_{IL\max} - V_{OL} \end{aligned}$$

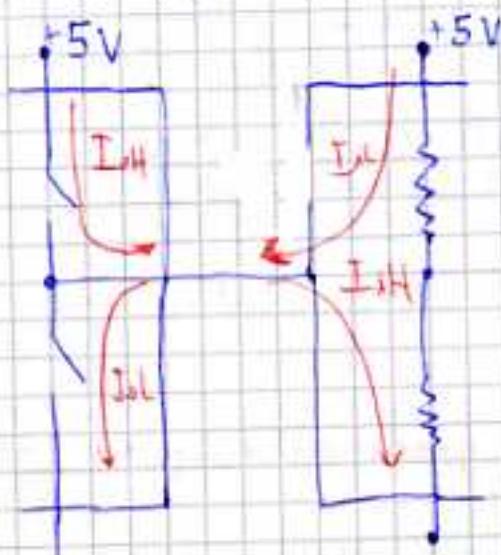
márgen de trabajo
estático.

Márgen de trabajo: Márgen de trabajo de corta duración.
dinámico

* I_{OH} : Corriente de salida en el lado alto.

* I_{OL} : Corriente de salida en el lado bajo.

* La entrada tiene una corriente:



* Fanout:

$$F_{OH} = \left| \frac{I_{OH}}{I_{IH}} \right| ; F_{OL} = \left| \frac{I_{OL}}{I_{IL}} \right|$$

Capacidad de entradas que puede controlar una salida.

* Fan In

- Contador de estados sencillo:
Comparte \rightarrow que equivale una determinada.

Si una entrada binaria $Fan\ In = 2$, consume como 2 entradas consumo.

- Retardo: tiempo que el cambio de variable se refleja en la salida. Epd

- Potencia consumida por una combinación: potencia que consume la combinación de binaria propuesta.

Una familia familiar: pWd

Si Epd es alto, pWd es bajo.

- FM: Factor de familia, $FM = Epd \cdot pWd$

Si Epd es bajo, pWd es alto.

10

* Familia lógica ideal:

$$V_{OH} = V_{cc} \text{ (Tiempo de Fuerza)}$$

$$V_{OL} = 0V$$

$$V_{IH} = V_{IL} = \frac{V_{cc}}{2}$$

$$V_{IL} = V_{cc}/2$$

$$NM = V_{cc}/2$$

$$FO = \rightarrow \infty$$

$$Z_i = \rightarrow \infty$$

$$Z_o = 0$$

$$I_{OH} = \rightarrow \infty$$

$$I_{IH} = \rightarrow \infty$$

$$I_{OL} = 0A$$

$$I_{IL} = 0A$$

$$Epd = 0$$

$$pWd = 0$$

$$FM = 0$$

Punto 6

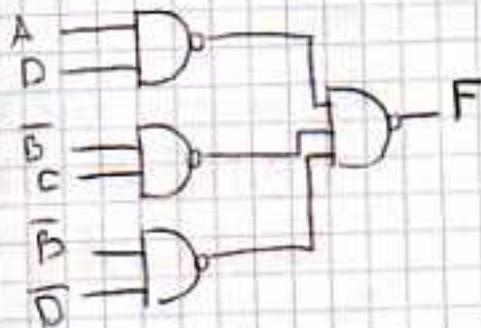
	0	2	3	8	9	10	11	15
I _{p1}	V	V		V	V			
I _{p2}			V		V	V		
I _{p3}	V	V		V	V			
I _{p4}	V	V	V	V	V	V	V	V

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

Por Demorgan:

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

$$F(A, B, C, D) = \overline{\overline{A} \cdot \overline{D}} \cdot \overline{\overline{B} \cdot C} \cdot \overline{\overline{B} \cdot \overline{D}}$$



Por OS:

	CD	AB	00	01	11	10
			0	1	3	2
			0	1	5	6
			0	X	0	0
			1	0	1	1
			1	X	X	0
			1	1	11	10

Implicantes primas

$$I_{p1}(1; 5; 6; 7) = A + \bar{B}$$

$$I_{p2}(4; 5; 10; 11) = \bar{B} + C$$

$$I_{p3}(4; 6; 11; 15) = \bar{B} + D$$

$$I_{p4}(1, 5) = A + C + \bar{D}$$

Implicantes primos esenciales

$$I_{p1}; I_{p2}; I_{p3}$$

Petr. C

	1	4	6	7	8	9
I_{p_1}		V	V	V		
I_{p_2}			V	V		
I_{p_3}	V				V	V

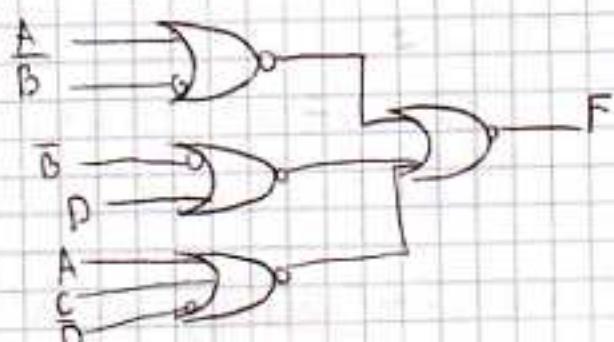
Ley de De Morgan

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

$$\text{Por De Morgan } A + \overline{B} = \overline{A} \cdot \overline{B} \Rightarrow A + B = \overline{\overline{A} \cdot \overline{B}}$$

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

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



Ej 45, guia 2:

	S	S	S	E	E	E	G	G
0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	
2	0	1	0	0	0	1	0	
3	0	1	1	1	0	0	0	
4	1	0	0	0	1	0	0	
5	1	1	1	0	1	0	0	
6	1	1	0	1	0	0	1	
7	1	1	1	1	0	0	0	0

Ejemplo 3.

A	B	C	D	E	F
0	0	0	1	1	0
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	0	0

Pseudotabla con redundancia

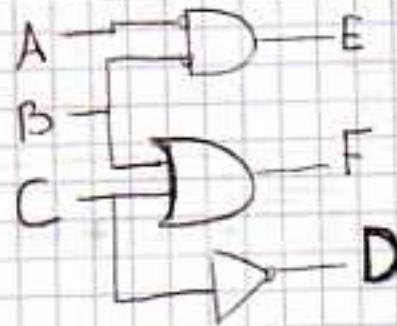
A	B	C	D	E	F
0	0	0	1	0	
0	0	1	0	1	
0	1	0	1	0	
0	1	1	0	0	
1	0	0	1	0	

$D = \bar{C}$

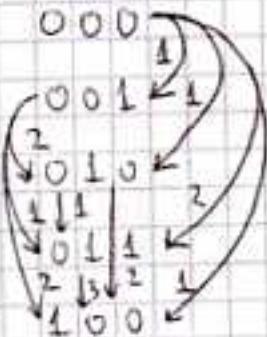
A	B	C	D	E	F
0	0	0	1	1	
0	0	1	0	0	
0	1	0	1	0	
0	1	1	0	0	
1	0	0	1	0	

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

$F = C + B$

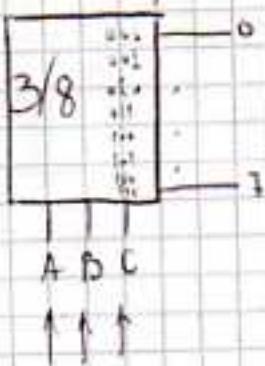


Dist. minima = 1

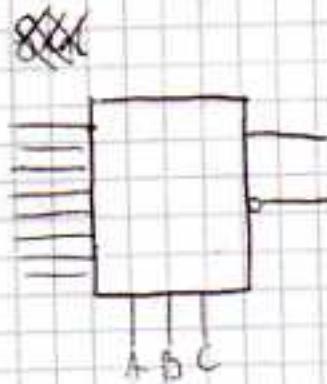


+ Decodificador:

Se entra un código y se decodifica:



+ Multiplexor:



* Problema de multiplexor:

- Sumador Paralelo, con dígitos en fútil de 2 números de 2 bits C/M usando multiplexor de 2 entradas de control.

Full-Adder:

A	B	Cin	Cout	Sout
0	0	0	0	0
1	0	1	0	1
2	0	1	0	1
3	1	1	1	0
4	1	0	0	1
5	1	1	1	0
6	1	0	1	0
7	1	1	1	1

Karnaugh para C/M:

AB	Cin	
	0	1
00	1	
01	0	1
10	1	0
11	0	1

Cin		0	1
AB	0	1	0
00		1	
01		0	1
10		1	0
11		0	1

Suma

Cout

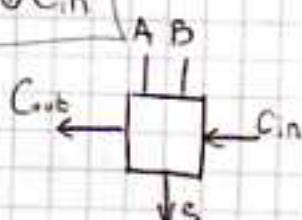
$$S = \bar{A}\bar{B}C_{in} + \bar{A}\bar{B}\bar{C}_{in} + A\bar{B}C_{in} + A\bar{B}\bar{C}_{in}$$

$$S = C_{in}(\bar{A}\bar{B} + A\bar{B}) + \bar{C}_{in}(\bar{A}B + A\bar{B})$$

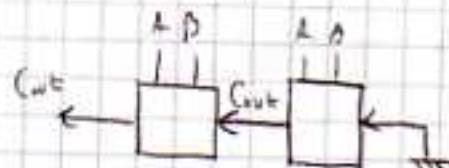
$$Cout = A\bar{B}C_{in} + \bar{A}\bar{B}C_{in} + A\bar{B}C_{in}$$

$$Cout = A\bar{B} + B\bar{C}_{in} + AC_{in}$$

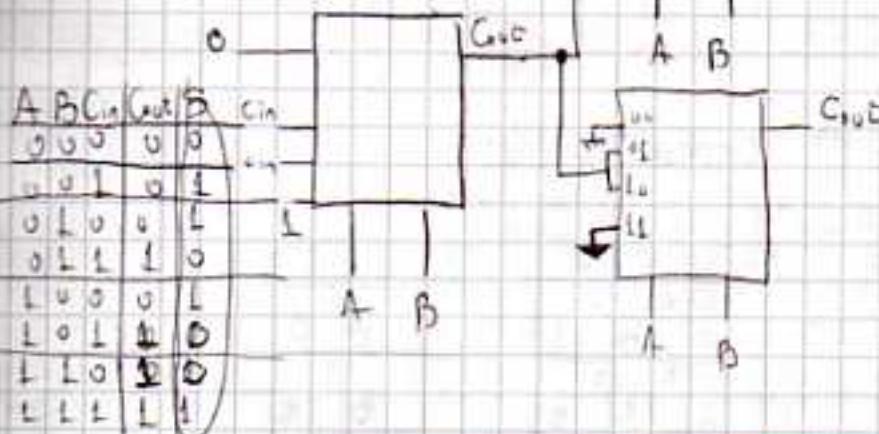
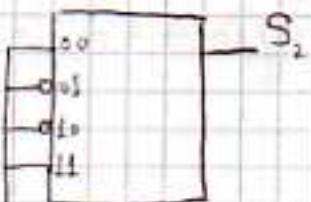
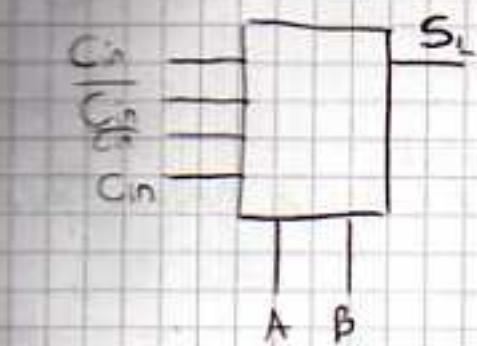
$$S = A \oplus B \oplus C_{in}$$



Para 2 bits:



* Porque el menor Sea Serie, deben ser multiplicadores en paralelo:



El multiplicador tiene
Solo 1 Salida

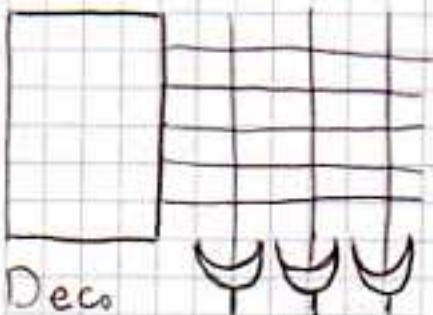
Los decodificadores comerciales tienen salidas negadas.

Considerar P/partial.

ALU:

*PLDs

* ROM: Read Only Memory.



0	0	0	0	
0	0	0	1	
0	0	1	1	
0	0	1	0	
0	1	0	0	
0	1	1	1	
0	1	1	0	
1	1	0	1	
1	1	1	1	
1	1	1	0	
1	0	1	0	
1	0	1	1	
1	0	0	1	
1	0	0	0	
1	0	0	0	Gray

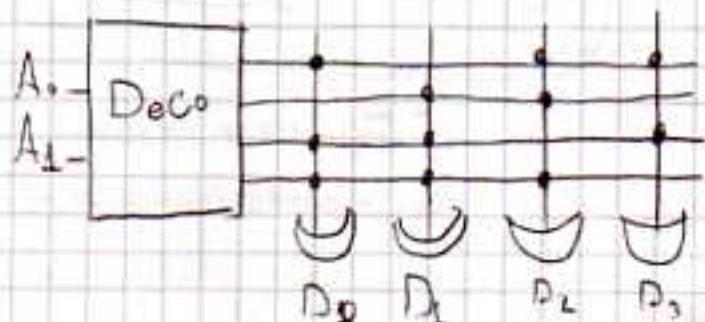
Tienen solo los que contemplan

Objetos contemplados:

2^n, unidimensional.

Ej:

A ₀	A ₁	D ₀	D ₁	D ₂	D ₃
0	0	1	0	1	1
0	1	0	1	1	0
1	0	1	1	0	1
1	1	1	1	1	0



MATRIZ Fija
MATRIZ OR PROGRAMABLE

*PAL

MATRIZ
AND
PROGRAMABLE



MATRIZ
OR
FIJA

27/5 *

* Circuitos Secuenciales.



$$F = F(\text{entradas, estado actual})$$

$$G = G(\text{estado, Clk})$$

- Combinacional:

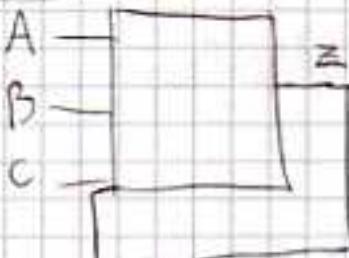
$$Z = F(A, B, C)$$

Combinacional de los
entradas.



- Secuencial:

$$Z = F(A, B, C, Z)$$



Retro-alimentación

* Tabla de verdad del Flip-Flop:

TABLA DE VERDAD AMPLIADA	R'	S'	Q'	Q'^{n+1}
	0	0	0	0
	0	1	1	1
	0	1	0	1
	1	0	0	0
	1	0	1	X
	1	1	1	X
	1	1	0	X



TABLA
DE VERDAD
REDUCIDA

R'	S'	Q'^{n+1}
0	0	0
0	1	1
1	0	0
1	1	X



En Karnaugh:

$S'Q'$	00	01	11	10
R'	0	0	1	1
	0	1	1	0

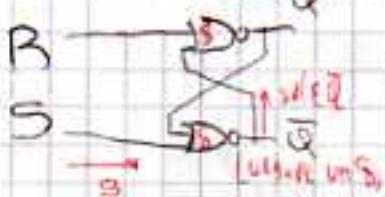
$$Q'^{n+1} = S + \bar{R} \cdot Q'$$

Ec. característica.

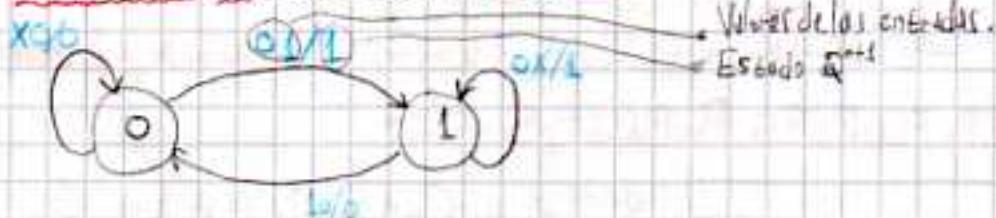
El tiempo de pulsación debe ser mayor a $2S$, donde S es el retraso:

Salida

$\rightarrow Q$



* Diagrama de transición de estados.

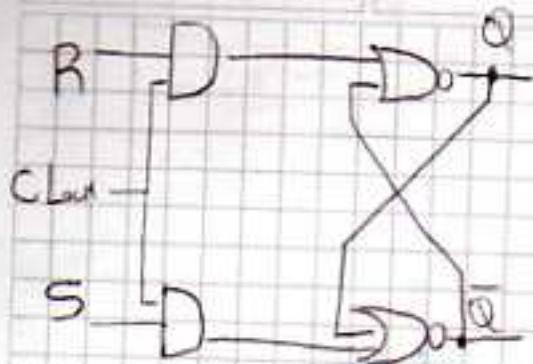


Vales de las entradas.

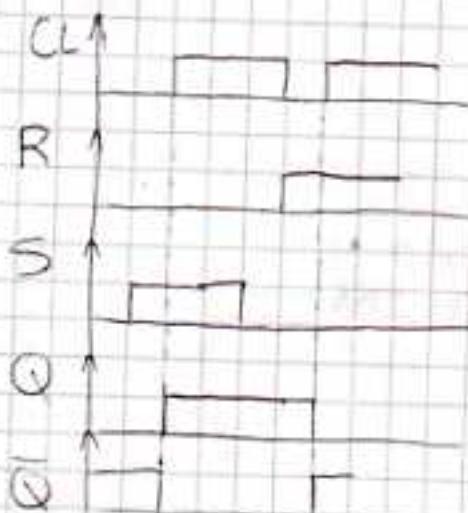
Estado Q'^{n+1}

* Tabla de transiciones del FLIP-FLOP.

Q'	Q' ⁿ⁺¹	R	S
0	0	X	0
0	1	0	0
1	0	0	0
1	X	0	X



De las formas se dispone del clock.



* FLIP-FLOP JK:

J	K	Q^n	Q^{n+1}
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	1
1	1	0	0

TABLA DE VERDAD AMPLIADA

J	K	Q^n	Q^{n+1}
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

TABLA DE VERDAD REDUCIDA

• En Karnaugh.

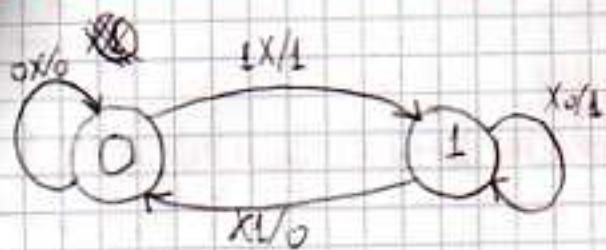
J	K	Q^n	Q^{n+1}
0	0	00	01
1	0	11	10

$$Q^{n+1} = J \cdot \bar{Q}^n + \bar{K} \cdot Q^n$$

Q^n	Q^{n+1}	J	K
0	0	0	X
1	1	X	1

TABLA DE TRANSICIÓN DEL FLIP-FLOP

DIAGRAMA DE ESTADOS



* FLIP-FLOP D

- Se llama "D", por dato. Almacenar dato en D.

TABLA DE VERDAD AMPLIADA		$D Q^n Q^{n+1}$
0	0	0
0	1	0
1	0	1
1	1	1

$D Q^n Q^{n+1}$		TABLA DE VERDAD REDUCIDA
0	0	0

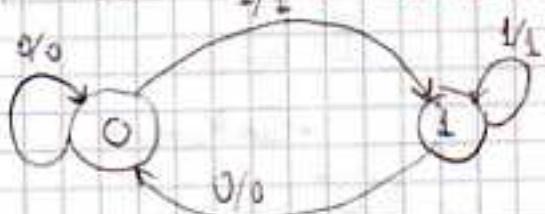
- En Karnaugh.

Q^n		0	1
0	0	0	0
1	1	1	1
$Q^{n+1} = D$			

TABLA DE TRANSICIÓN

Q^n	Q^{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

DIAGRAMA DE ESTADOS



* FLIP-FLOP Toggle

TABLA
DE VERDAD
AMPLIADA

T	Q^n	Q^{n+1}
0	0	0
0	1	1
1	0	1

TABLA
DE VERDAD
REDUCIDA

T	Q^n	Q^{n+1}
0	0	0
1	1	0

TABLA
DE
TRANSICIÓN

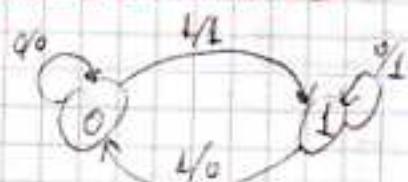
Q^n	Q^{n+1}	T
0	0	0
0	1	1
1	0	0

• En Karnaugh

T	Q^n	0	1
0	0	0	1
1	0	1	0

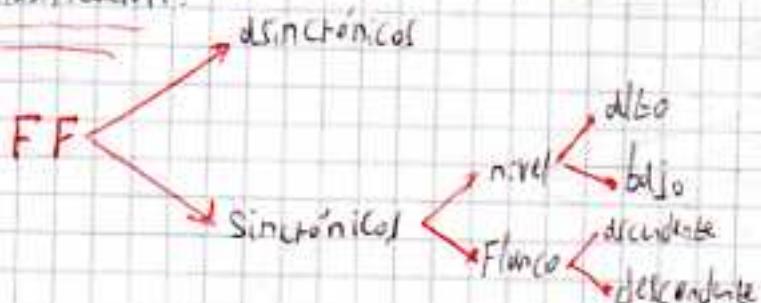
MARINDE

DIAGRAMA DE ESTADOS

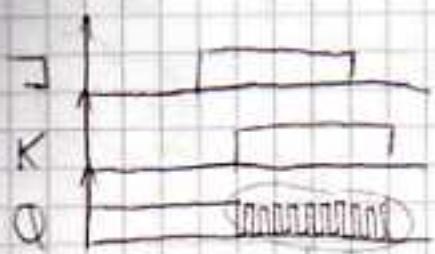


$$Q^{n+1} = T \cdot \bar{Q}^n + \bar{T} \cdot Q^n$$

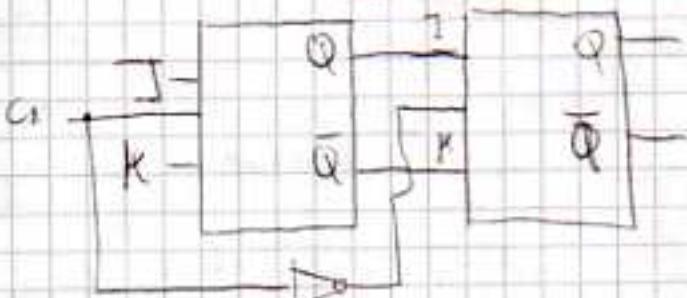
* Clasificación:



En el FF JK:

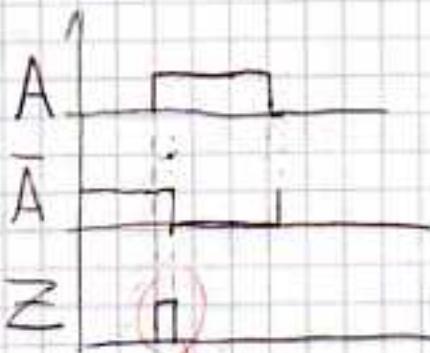


→ Alta latencia. Se solucionó como:

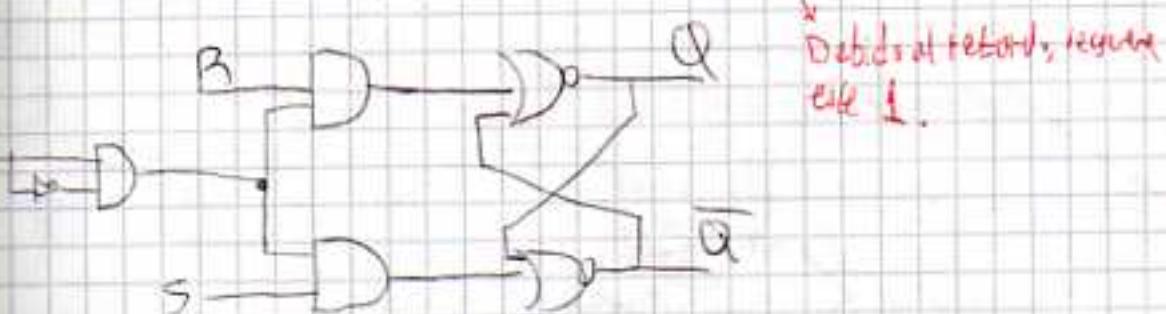


Sol. lenta FFMS,
mucha latencia
y latido oscilante

* Hay en dir si usó FF de flanco:

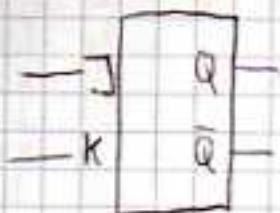


En el RS:



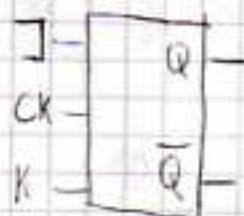
Dibujar el retardo, regular
este 1.

* Asincrónico:

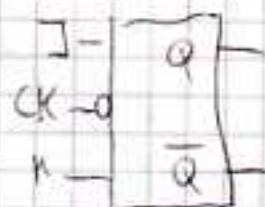


* Sincrónico

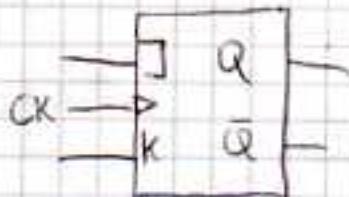
• nivel alto



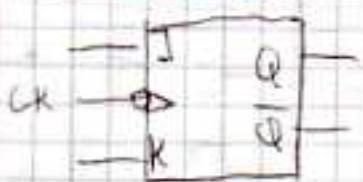
• nivel bajo



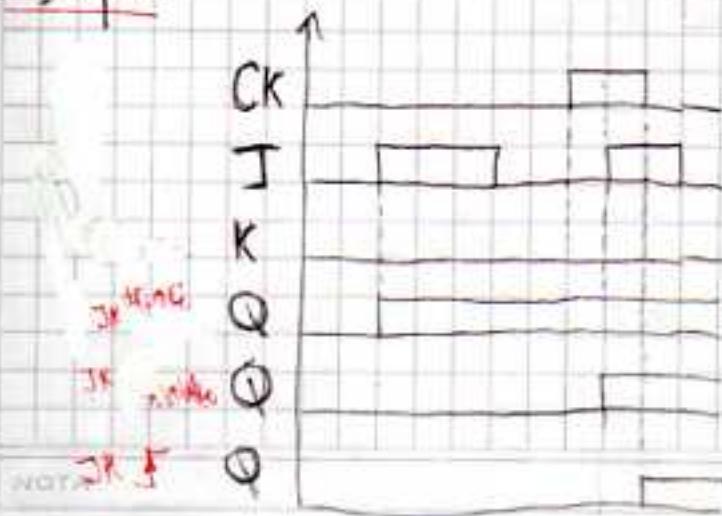
• Flanco alto.



• Flanco bajo.

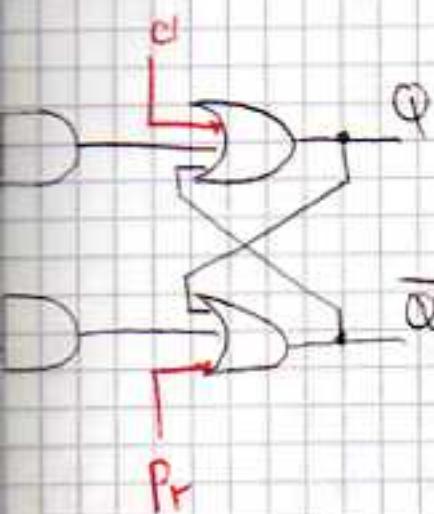


* Ejemplo:

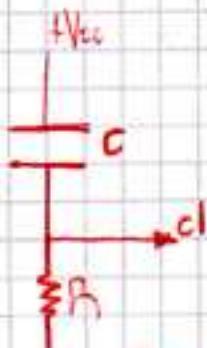


Ck	P	D	Q^n	Q^{n+1}
L	L	0	0	0
L	L	0	1	0
L	L	1	0	L
L	L	1	1	1
L	X	X	0	0

Ck	P	D	Q^n	Q^{n+1}
X	L	H	X	X
X	H	L	X	X
X	H	H	X	X

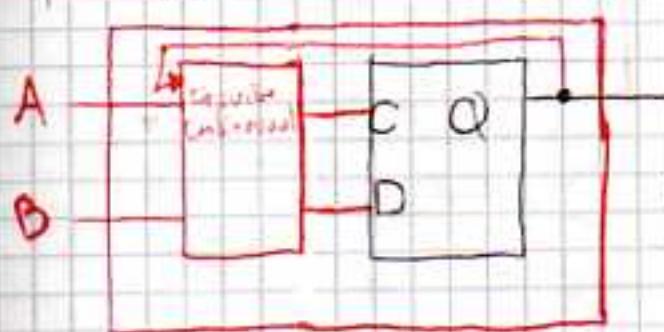


Clk y Preset siempre inician en 0, en 1.



Power-on Reset

* Métodos para FF → Transición } Funcionamiento Sincrónico y
d. borde de alta → Ec. características dinámico(s).



$$C = F_1(A, B, Q^n)$$

$$D = F_2(A, B, Q^n)$$

* Para un FF JK con FFRS:

Q^n Q^{n+1} R S

000	0	X	0
001	1	X	X
010	0	X	0
011	0	1	0
100	1	0	1
101	1	0	X
110	1	0	1
111	0	1	0

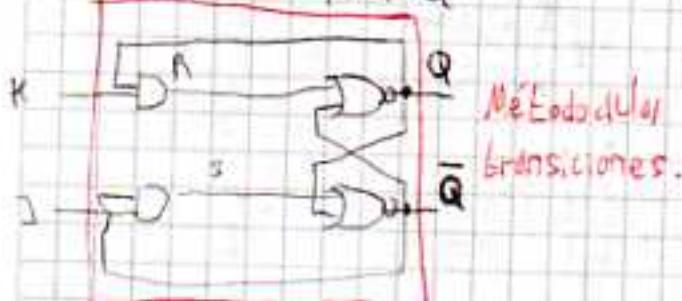
Q^n Q^{n+1} R S

0	0	X	0
0	1	0	1
1	1	0	0
1	0	X	1

J	00	01	11	10
0	X	0	L	X
1	0	0	L	0

J	00	01	11	10
0	0	X	0	0
1	D	x	0	L

$$R = K \cdot Q^n$$



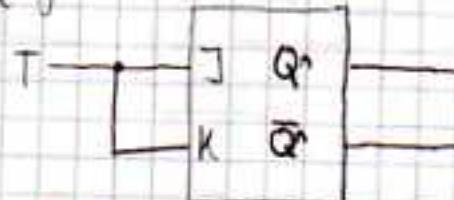
$$S = J \cdot Q^{\bar{n}}$$

* Para un FF JK con FFT:

$$Q^{n+1} = J \cdot Q^n + \bar{K} \cdot Q^n \quad J=T$$

$$Q^{n+1} = T \cdot Q^n + \bar{T} \cdot Q^n \quad K=T$$

Método de la función característica

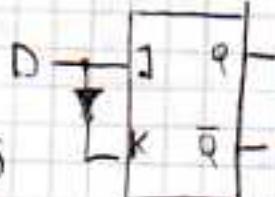


* Para un FF JK con FFD:

$$Q^{n+1} = J \cdot \bar{Q}^n + \bar{K} \cdot Q^n$$

$$Q^{n+1} = D - D(Q_n + \bar{Q}_n) = D \cdot Q_n + D \cdot \bar{Q}_n$$

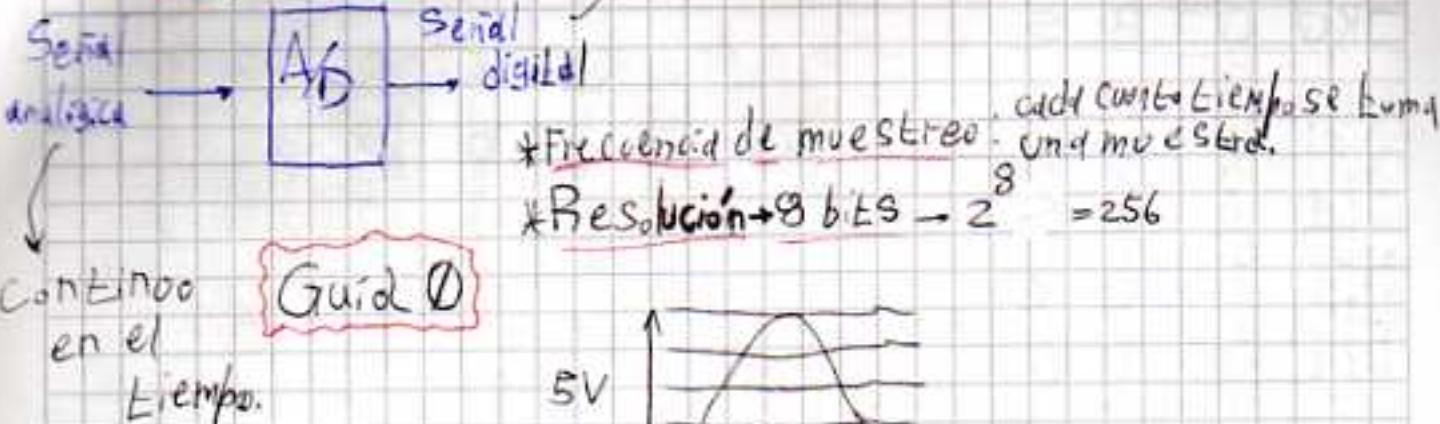
$$\begin{aligned} D &= J && \text{AK-RS} \\ D &= \bar{K} && \rightarrow K = \bar{D} \end{aligned}$$



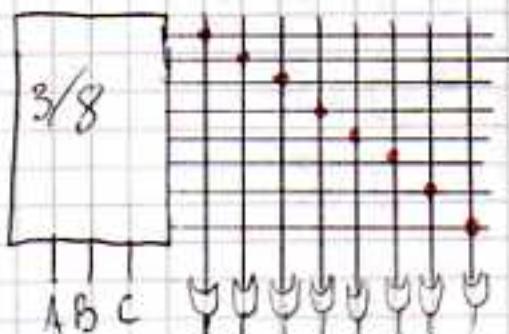
Revisión

1/6 + ~~revisar~~ parte d).

Muestras de la señal analógica, debido a pasarlo por el conversor.

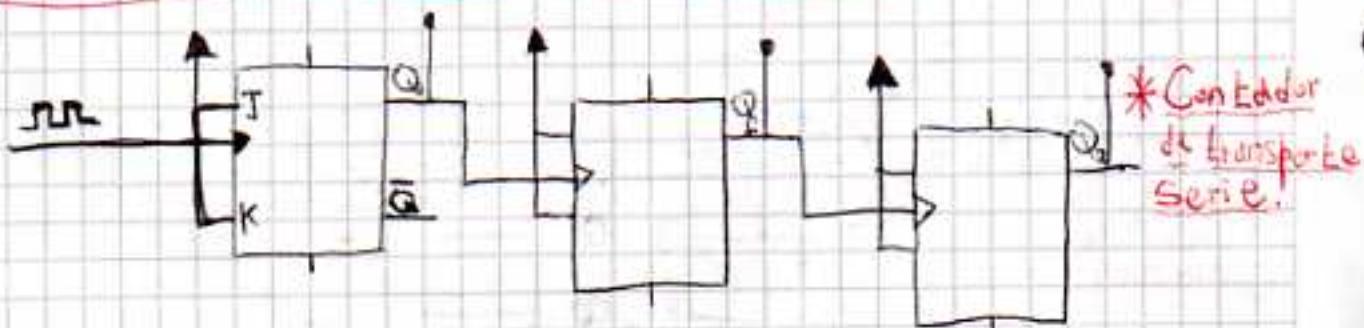


Ej 5:

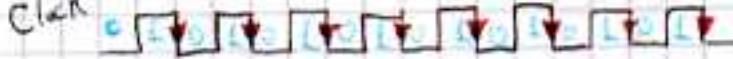


A	B	C	D	E	F	G	H	I	J	K
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0
0	1	1	0	0	1	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	1	0	0	0	0
1	1	1	0	0	0	0	1	0	0	0

Recuperar ejercicios de multiplicador decodificador Y PLD's.

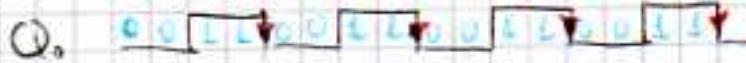


clock



* Son fáciles de hacer, pero el retraso es una desventaja

Q₀



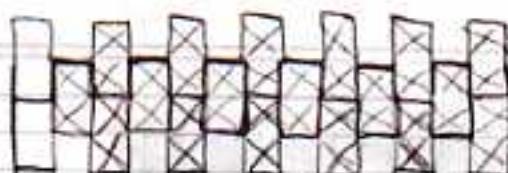
Q₁



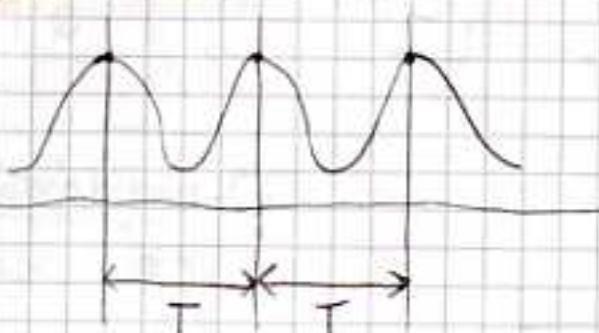
Q₂



*Contadores.



• Función periódica: Cuando pasa un período de tiempo el valor de la función se repite.



$$f = \frac{1}{T}, \text{ Frecuencia} = \frac{1}{\text{período}}$$



$$T = t_1 + t_2$$

Ej:
⑤

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

$$D_b = \bar{B}$$

$$D_a = \bar{A}$$

outputs

$$Q_0 Q_1 Q_2$$

$$0 \ 0 \ 0$$

$$0 \ 0 \ 1$$

$$0 \ 1 \ 0$$

$$0 \ 1 \ 1$$

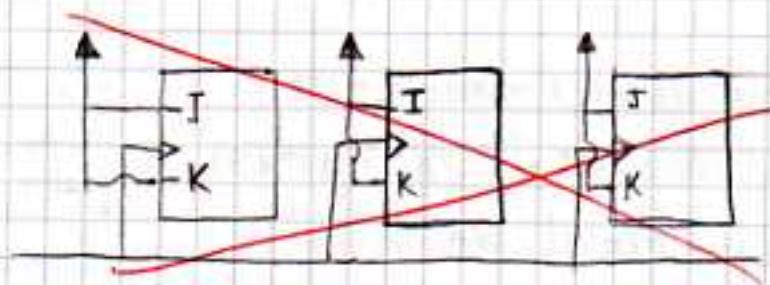
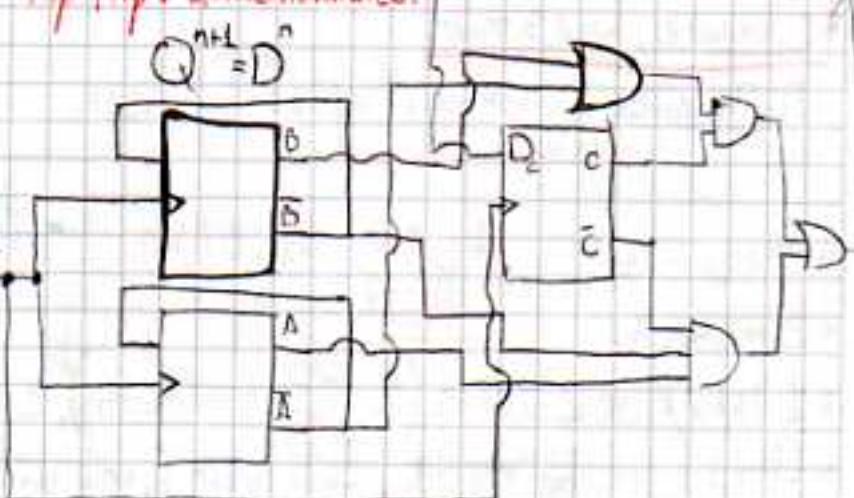
$$1 \ 0 \ 0$$

$$1 \ 0 \ 1$$

$$1 \ 1 \ 0$$

$$1 \ 1 \ 1$$

• Contador sincrónico: el clock llega a todos los flip-flops simultáneamente.



Lugares para comprar componentes.

- Radio Acebo: Uruguay y 300-400.
- Micro electrónico: Perón, entre Paraná y Uruguay.
- Electrocomponentes: Calle Paraná.
- SyC: Calle Paraná.

* Alimentación

- Batería de 9V, por el bártir.
- Fuente.

3/6

* Contador

ABCD

0	0	0	0	a	0
0	0	1	0	0	x
0	1	0	0	c	c
0	1	1	0	l	x
1	0	0	1	b	b
1	0	1	0	0	x
1	1	0	1	d	d
1	1	1	0	l	x

A	B	Q ⁿ	Q ⁿ⁺¹	C	D
0	0	0	0	a	0
0	0	1	1	0	x
0	1	0	1	c	c
0	1	1	0	l	x
1	0	0	1	b	b
1	0	1	0	0	x
1	1	0	1	d	d
1	1	1	0	l	x

* Módulo: cantidad de estados que permanecen/cambian.

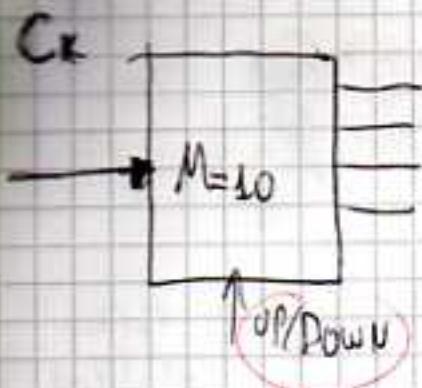
* Contador tiene memoria.

A	B	Q ⁿ	Q ⁿ⁺¹	C	D
0	0	0	0	0	0
0	0	1	1	0	0
1	0	0	1	0	0

$$C = B + Q^n$$

A	B	Q ⁿ	Q ⁿ⁺¹	C	D
0	0	0	0	0	0
0	0	1	1	0	0

$$D = A$$



Permite control hacia arriba o hacia abajo.

ASÍNCRONICOS

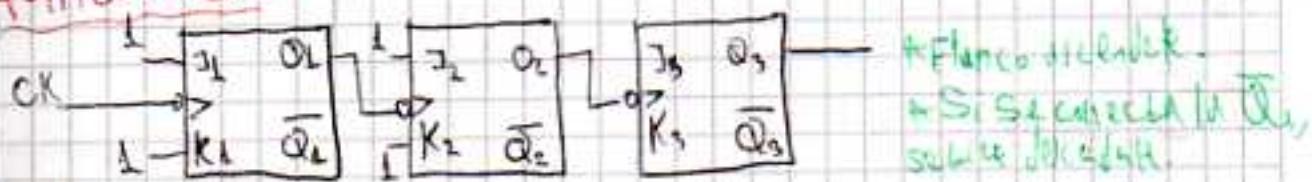
ANÁLISIS BINARIO, PERIODICO
SÍNTESIS $M \leq 2^n$

* Contadores

SÍNCRONICOS \leftarrow ANÁLISIS SÍNTESIS

Síncronico: Todos los Flip-Flops cambian al mismo tiempo.

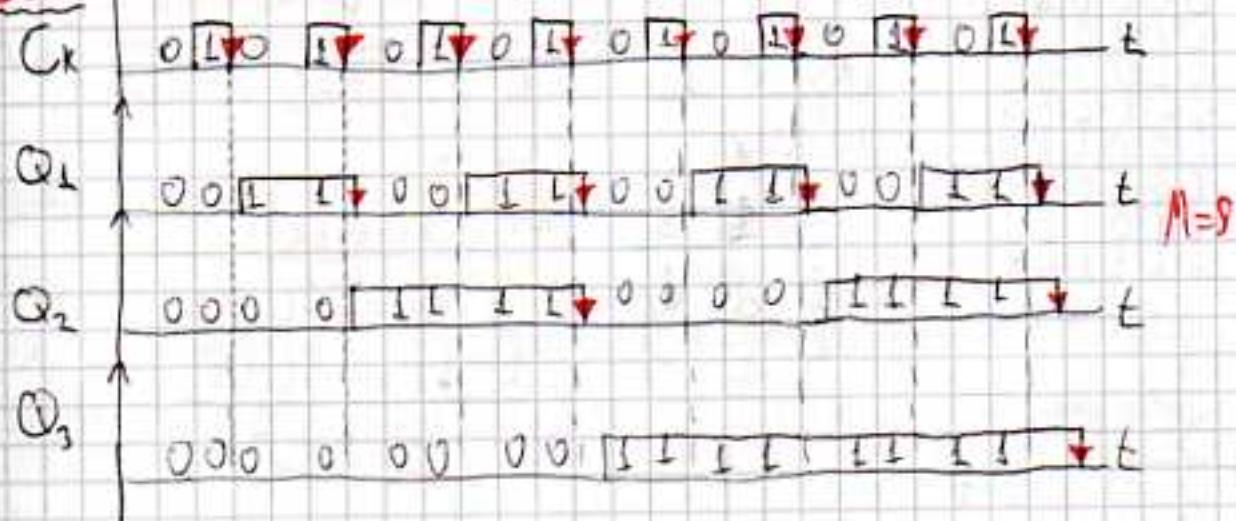
Asíncronico:



* Flanco ascendente.

* Si se cambia la Q1, sube el Q2.

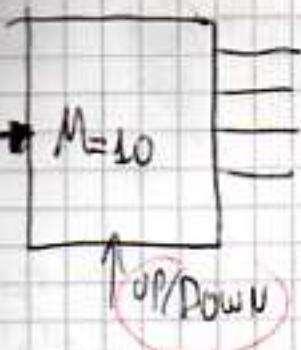
* Análisis



→ Es un divisor de frecuencia entre la entrada y la salida por su módulo.

* Estados espurios: Estados NO Deseados. de uno por vez.

Ck



Permite control hacia arriba o hacia abajo.

ASÍNCRONICOS

ANÁLISIS BINARIO
SÍNTESIS

Asíncrono
pero tiene

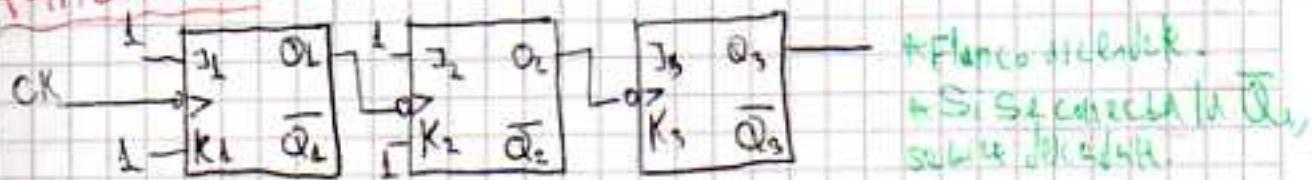
$M \leq 2^n$

* Contadores

SÍNCRONICOS < ANALISIS
SÍNTESIS

• Síncronico: Todos los Flip-Flops cambian al mismo tiempo.

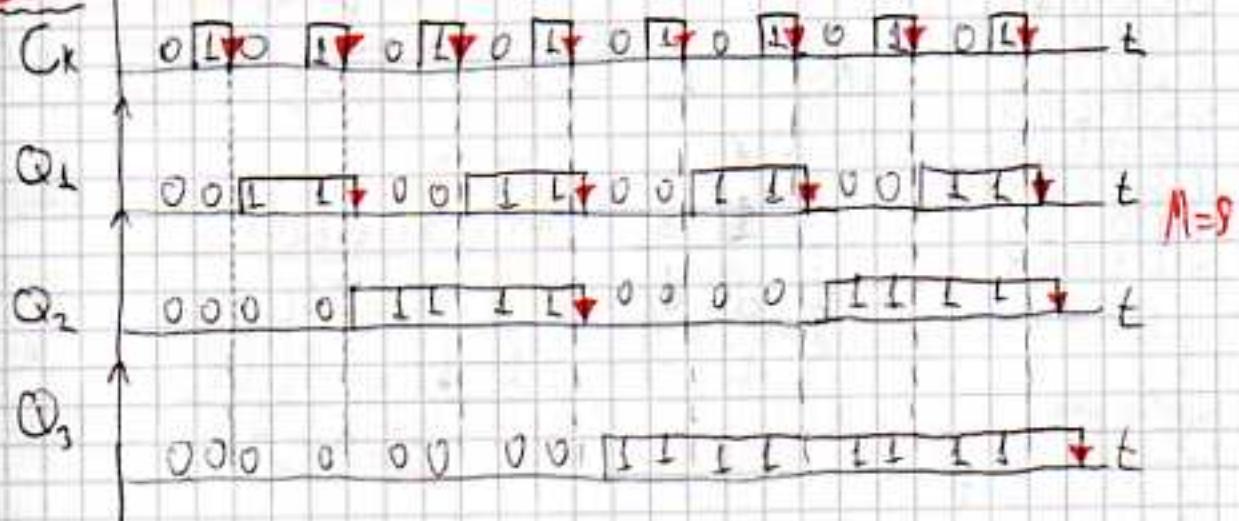
• Asíncronico:



Flanco ascendente.

Si se cambia la Q_1 , suben Q_2 y Q_3 .

• Análisis



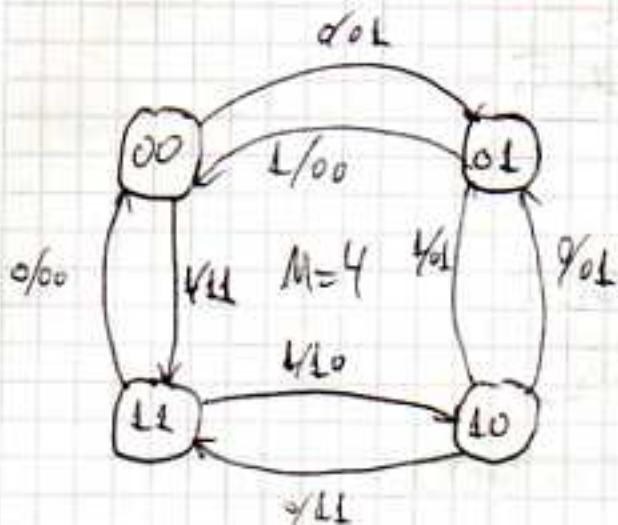
• Es un divisor de frecuencia entre la entrada y la salida por su módulo.

• Estados espurios: Estados NO Deseados. de uno por vez.

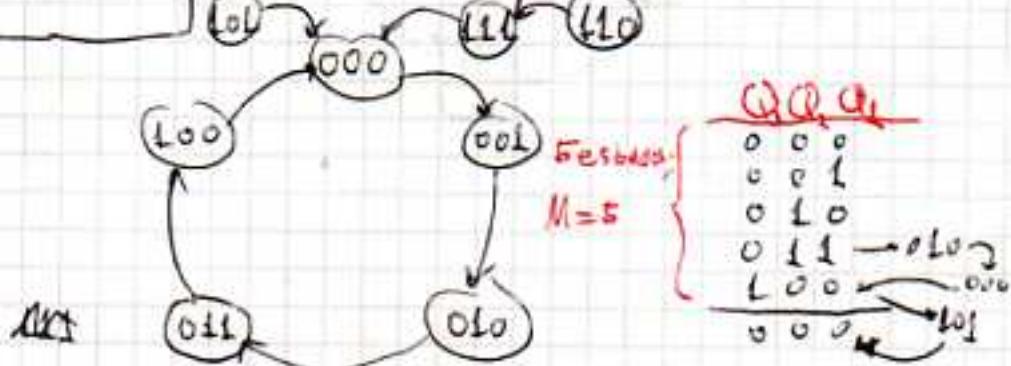
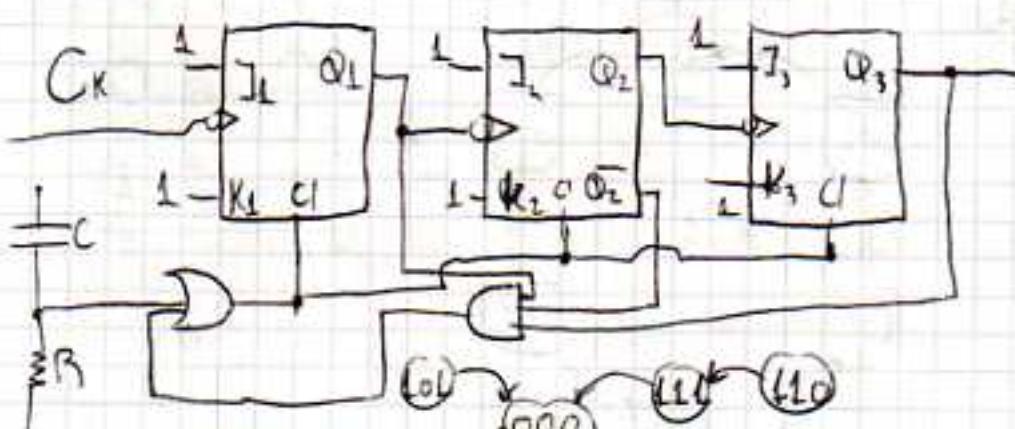
+ DIVISOR $F_{\text{MAX}} = \frac{f}{2 \cdot E_{\text{PD}}}$

$E_{\text{PD}} = \text{Tiempo de muestreo}$

+ CONTADOR $F_{\text{MAX}} = \frac{1}{2 \cdot n \cdot E_{\text{PD}} + E_{\text{d}}$

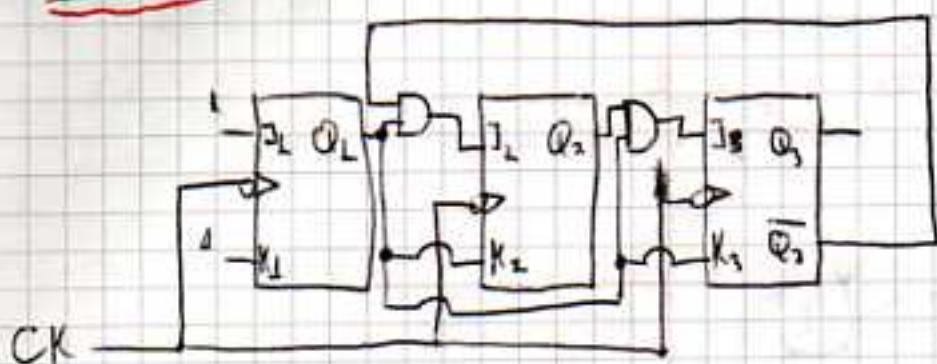


~~M=5~~ M=5



Síntesis

Análisis



$$M=8$$

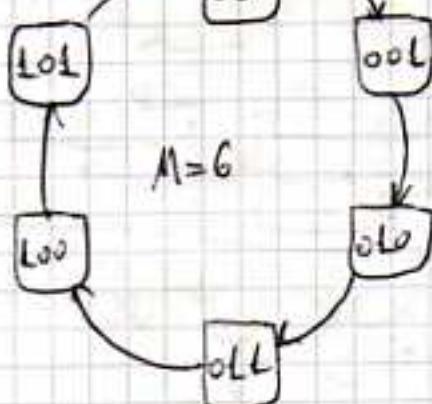
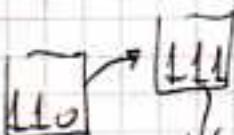
$$J_1 = 1 \quad J_2 = Q_1 \cdot \bar{Q}_2 \quad J_3 = Q_2 \cdot Q_3$$

$$K_1 = 1 \quad K_2 = Q_1 \quad K_3 = Q_2$$

$Q_3 Q_2 Q_1$ Sólo los que incluyen 000

$$M=6$$

<u>$Q_3 Q_2 Q_1$</u>
0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1

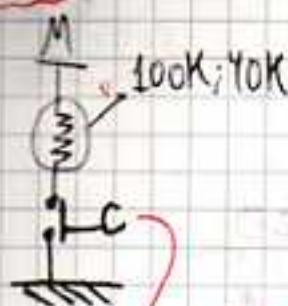


Estatutos prohibidos

<u>$Q_3 Q_2 Q_1$</u>
0 1 1
1 1 1
0 0 0

8/6

* Pulsador.



Contador up/down 45-10

Se puede hacer el decreto de
Clock con 2 pulsadores

Se codifica

Con = salida del código

* Las entradas no pueden estar abiertas, deben ir con una resistencia, para asegurar el 0.

Contador decimal 40-29

CMOS

19/6

* Síntesis Cons. Síncronos

$M=13$ binario

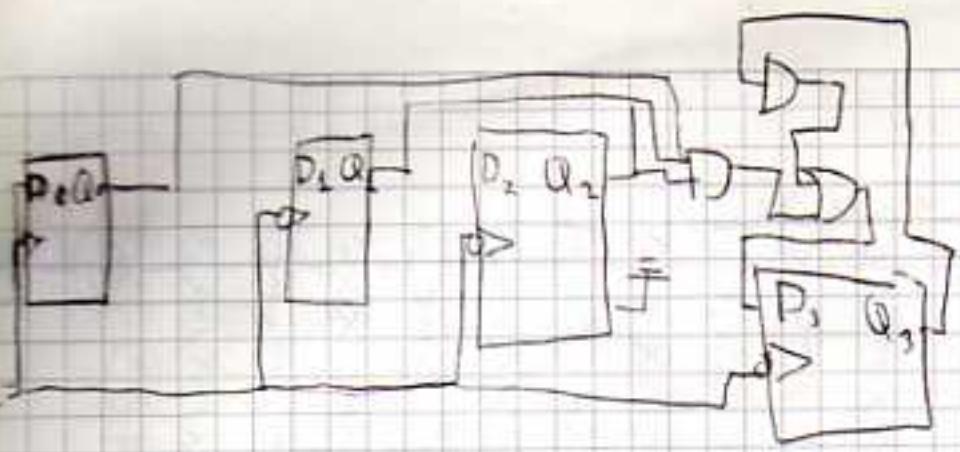
JK Función de Sincrono

D Nivel de multiplicación

* Transiciones

A^n	Q^{n+1}	JK
0	0	0 X
0	1	L X
1	0	X 1
1	1	X 0

Q^n	Q^{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1



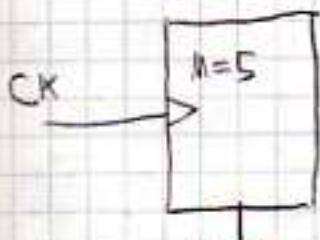
BCD GRAY XS-3

Q_3^n	Q_2^n	Q_1^n	Q_0^n	Q_3^{n+1}	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}
0	0	0	0	X	X	X	X
0	0	0	1	X	X	X	0
0	0	1	0	0	1	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	1	1
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	X
1	0	0	0	X	X	X	X
1	0	0	1	X	X	0	0
1	0	1	0	0	0	0	1
1	0	1	1	0	1	1	1
1	1	0	0	1	0	1	0
1	1	0	1	1	1	0	0
1	1	1	0	1	0	0	0

BINARIO

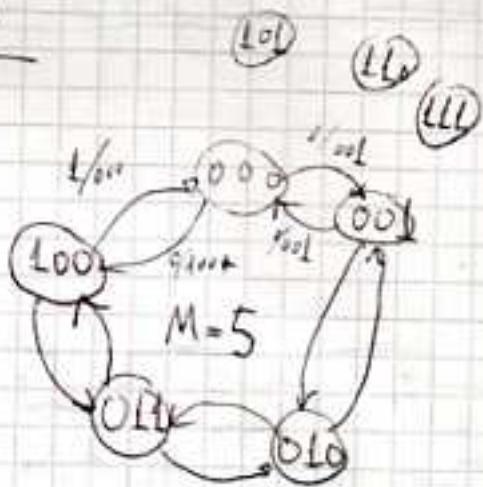
GRAY XS-3

→ UP Down Binario (ultimo probará o para abajo)



Up/Down
permite contar
haciendo lo hacia
el otro.

<u>Up</u>	Q_0^*	Q_1^*	Q_2^*	Q_3^*	Q_4^{n+1}	Q_5^{n+1}	Q_6^{n+1}
0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	1	0
0	0	1	0	1	0	1	0
0	1	0	0	0	X	X	X
0	1	0	0	1	X	X	X
0	1	0	1	0	X	X	X
0	1	0	1	0	0	1	0
0	1	0	1	0	0	0	1
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0



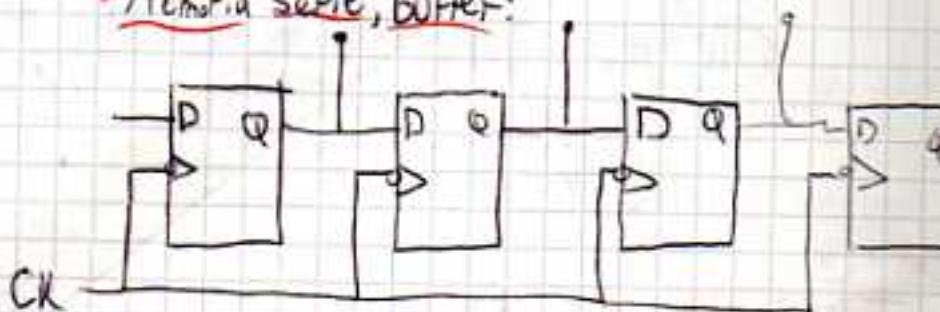
* Memoria de acceso secuencial

FIFO → →

LIFO

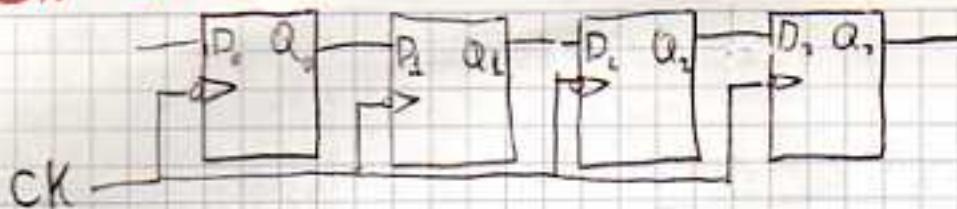


* Memoria serie, buffer:



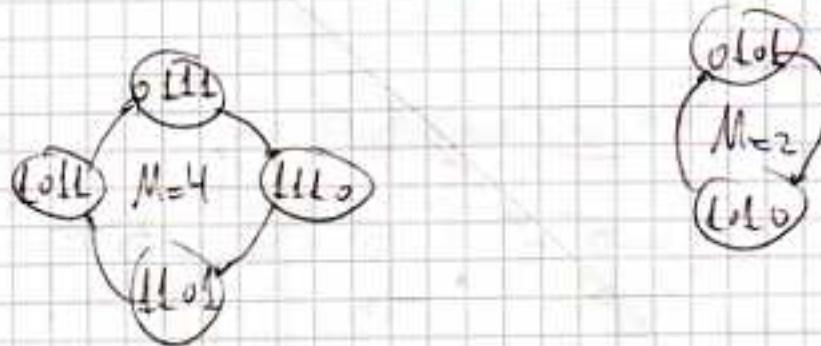
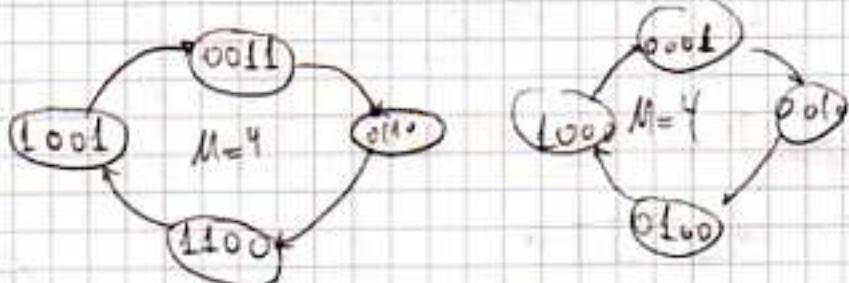
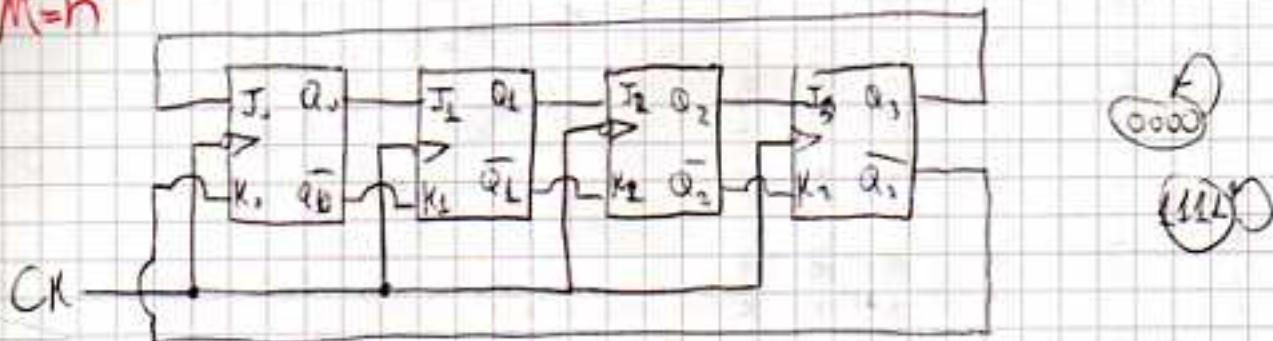
SHIFT REGISTER
ENTRA COMO SERIE,
SALE EN PARALELO

ENTRASERIE, SALE SERIE



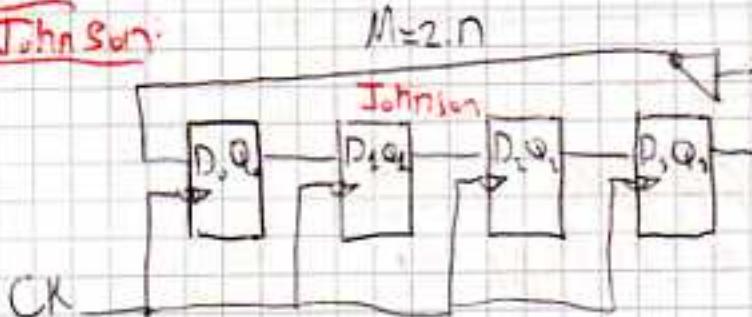
Anillo:

$M=n$



Contador

Johnson



Hacer puente sobre en la puerta.

positivo con positivo

negativo con negativo.

* Informe:

1 - Colección: nombre, apellido, padrón
profesor
fecha

2 - Objetivo: 364 reglones

3 - Diagrama en bloques

4 - Dibujos, a partir del diagrama

5 - Esquema, circuito lógico, si se quiere, listo de componentes

6 - Conclusión

$Q_4 Q_3 Q_2 Q_1$	$D C B A$
0 0 0 0	0 0 0 0
0 1 0 0	1 0 0 0
0 0 1 0	0 1 0 0

