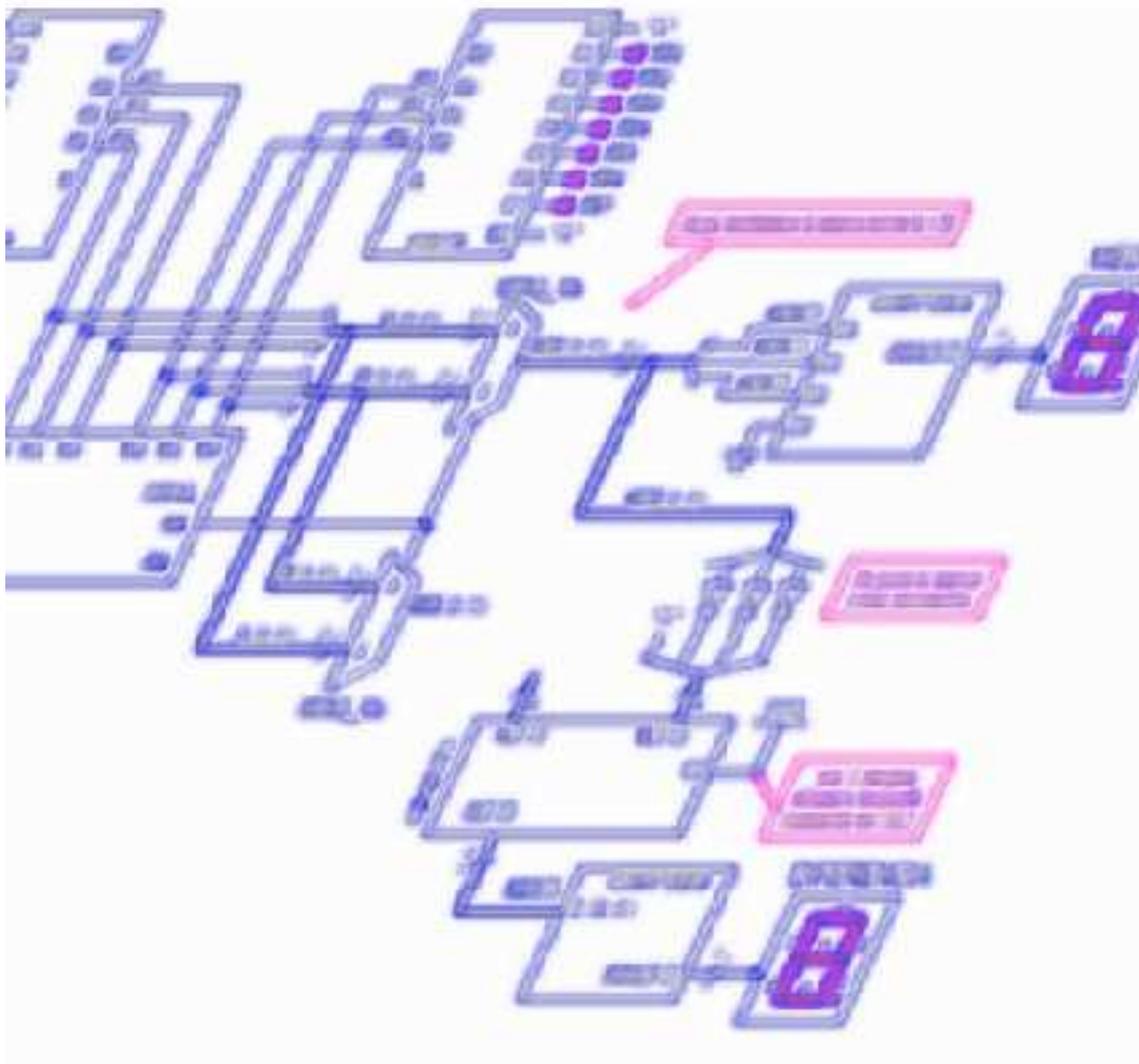


Problemas resueltos de Electrónica Digital



Felipe Machado Sánchez

Departamento de Tecnología Electrónica
Universidad Rey Juan Carlos

Problemas resueltos de electrónica digital

Felipe Machado

Departamento de Tecnología Electrónica de la Universidad Rey Juan Carlos

Versión 1.00 creada el 31 de octubre de 2011



Esta versión digital de *Problemas resueltos de electrónica digital* ha sido creada y licenciada por Felipe Machado Sánchez con una licencia de Creative Commons. Esta licencia permite los usos no comerciales de esta obra en tanto en cuanto se atribuya autoría original. Esta licencia no permite alterar, transformar o generar una obra derivada a partir de esta obra

Con esta licencia eres libre de copiar, distribuir y comunicar públicamente esta obra bajo las condiciones siguientes:

Reconocimiento: debes reconocer y citar a los autores de esta obra

No comercial: no puedes utilizar esta obra para fines comerciales

Sin obras derivadas: no puedes alterar, transformar o generar una obra derivada a partir de esta obra

Para más información sobre la licencia, visita el siguiente enlace:

<http://creativecommons.org/licenses/by-nc-nd/3.0/>

Esta obra está disponible en el archivo abierto de la Universidad Rey Juan Carlos:

<http://eciencia.urjc.es/dspace/handle/10115/5727>

<http://hdl.handle.net/10115/5727>

Para más información sobre los autores:

<http://gtebim.es/>

<http://gtebim.es/~fmachado>

Índice

Índice	3
1. Prefacio	5
2. Sistemas de numeración	6
2.1. Enunciados	6
2.2. Soluciones	8
3. Sistemas de codificación	13
3.1. Enunciados	13
3.2. Soluciones	14
4. Aritmética binaria	16
4.1. Enunciados	16
4.2. Soluciones	17
5. Aritmética en BCD	27
5.1. Enunciados	27
5.2. Soluciones	27
6. Aritmética en BCD XS-3	30
6.1. Enunciados	30
6.2. Soluciones	30
7. Álgebra de Boole y funciones lógicas	33
7.1. Enunciados	33
7.2. Soluciones	35
8. Bloques combinacionales	51
8.1. Enunciados	51
8.2. Soluciones	52
9. Elementos de memoria	56
9.1. Enunciados	56
9.2. Enunciados	56
10. Análisis de circuitos 1	58
10.1. Enunciados	58
10.2. Soluciones	59
11. Toldo automático	62
11.1. Enunciado	62
11.2. Solución	63
12. Riego automático	65
12.1. Enunciado	65
12.2. Solución	66
13. Detector de números primos en BCD-XS-3	67
13.1. Enunciado	67
13.2. Solución	67
14. Alarma de incendios	69
14.1. Enunciado	69
14.2. Solución	69
15. Comparador	71
15.1. Enunciado	71
15.2. Solución	71
16. Comparador BCD-XS3	75
16.1. Enunciado	75
16.2. Solución	75
17. Convertidor de binario a BCD	78
17.1. Enunciado	78
17.2. Solución	78

18. Café electrónico.....	81
18.1. Enunciado.....	81
18.2. Solución.....	81
19. Tuberías.....	83
19.1. Enunciado.....	83
19.2. Solución.....	83
20. Báscula.....	86
20.1. Enunciado.....	86
20.2. Solución.....	87
21. Nivel de depósitos de agua.....	88
21.1. Enunciado.....	88
21.2. Solución.....	88
22. Depósitos de agua comunicados.....	91
22.1. Enunciado.....	91
22.2. Solución.....	91
23. ALU.....	94
23.1. Enunciado.....	94
23.2. Solución.....	94
24. Contador 1.....	97
24.1. Enunciado.....	97
24.2. Solución.....	97
25. Contador 2.....	99
25.1. Enunciado.....	99
25.2. Solución.....	99
26. Contador 3.....	103
26.1. Enunciado.....	103
26.2. Solución.....	103
27. Contador 4.....	109
27.1. Enunciado.....	109
27.2. Solución.....	109
28. Piano electrónico.....	111
28.1. Enunciado.....	111
28.2. Solución.....	111
29. Análisis de circuitos 2.....	116
29.1. Enunciado.....	116
29.2. Solución.....	116
30. Análisis de circuitos 3.....	119
30.1. Enunciado.....	119
30.2. Solución.....	119
31. Análisis de circuitos 4.....	121
31.1. Enunciado.....	121
31.2. Solución.....	121
32. Análisis de circuitos 5.....	123
32.1. Enunciado.....	123
32.2. Solución.....	123

1. Prefacio

En este manual se han recopilado algunos de los problemas propuestos en clase y en exámenes de la asignatura Electrónica Digital I de la titulación de Ingeniería de Telecomunicación de la Universidad Rey Juan Carlos (URJC). Los problemas tratan los sistemas de numeración y su aritmética, el álgebra de Boole, el diseño de circuitos con puertas lógicas y bloques combinacionales, el diseño de contadores y el análisis de circuitos sencillos.

Como los problemas de los sistemas de numeración y álgebra de Boole son más cortos, muchos de estos ejercicios se han agrupado en un mismo capítulo, mientras que los ejercicios más largos, que suelen combinar más de un tema, se han puesto de manera individual en un sólo capítulo.

Si quieres profundizar y aprender a implementar lo aprendido en un dispositivo programable (FPGA) puedes usar el libro *Diseño digital con Esquemáticos y FPGA*, de los autores Felipe Machado, Susana Borromeo y Norberto Malpica.

Para aprender a diseñar circuitos más complejos que incluyen máquinas de estados se puede utilizar el libro *Diseño de circuitos digitales con VHDL*. Este libro principalmente enseña a diseñar con VHDL, que es un lenguaje de descripción de hardware. Sin embargo, en sus capítulos finales incluye problemas de diseño que se propusieron en los exámenes de Electrónica Digital II. Este libro está disponible de manera gratuita en el archivo abierto de la URJC (<http://hdl.handle.net/10115/4045>)

Hay un último libro (*Diseño de sistemas digitales con VHDL*) en donde se enseña a realizar diseños electrónicos digitales más avanzados utilizando también el VHDL y las FPGAs. De igual modo, este libro está disponible en el archivo abierto de la URJC: <http://hdl.handle.net/10115/5700>

Antes de empezar con el libro me gustaría agradecer a los profesores del departamento Norberto Malpica y Susana Borromeo por su colaboración en la propuesta de problemas durante los años que compartimos la asignatura y los años que me precedieron. Asimismo me gustaría agradecerles su estupenda manera de colaborar cuando hemos compartido asignaturas y proyectos.

También me gustaría agradecer a todos los integrantes del Área de Tecnología Electrónica del Departamento de Tecnología Electrónica por su interés por la docencia de calidad y por su apoyo en las tareas docentes, habiéndome permitido *perder el tiempo* en tareas docentes que no están tan reconocidas. De manera especial se lo agradezco al Director del Departamento, Juan Antonio Hernández Tamames, que nos ha dado suficiente libertad en la orientación de nuestra docencia e investigación.

Y cómo no, me gustaría agradecer a los alumnos por su interés generalizado en aprender, su buen humor y sus comentarios sobre la asignatura.

Con el fin de que este manual pueda llegar al mayor número de estudiantes y con el objetivo de generar material educativo abierto, se ha publicado este manual bajo licencia *Creative Commons* que permite copiarlo y distribuirlo. Esperamos que disfrutes de su lectura y que te ayude a aprender la base de la electrónica digital. Para ir mejorando el manual, agradeceremos la comunicación de comentarios, sugerencias y correcciones a la dirección de correo electrónico del autor.

felipe.machado@urjc.es

2. Sistemas de numeración

2.1. Enunciados

2.1.1. Convertir los siguientes números binarios puros a sus equivalentes en base 10

a) 100110	f) 01100110
b) 110011	g) 10110011
c) 010111	h) 0101,11
d) 101110	i) 1001,10
e) 110111	j) 101010110,001

2.1.2. Convertir los siguientes números decimales a sus equivalentes en binario

a) 9	f) 258,75
b) 64	g) 0,75
c) 31	h) 1,625
d) 37	i) 19,3125
e) 131	

2.1.3. Convertir los siguientes números enteros hexadecimales a decimal

a) 13	c) 3F0
b) 65	d) DOCE

2.1.4. Convertir los siguientes números reales hexadecimales a decimal

a) 0,2	c) F1,A
b) 12,9	d) C8,D

2.1.5. Convertir el número hexadecimal $13,4_{16}$ a decimal y a binario:

a) $13,4_{16}$

2.1.6. Convertir los siguientes números a binario, octal y decimal

a) $3,A2_{16}$

b) $1B1,9_{16}$

2.1.7. Convertir los siguientes números a binario (8 dígitos fraccionarios máx), octal y hexadecimal (2 dígitos fraccionarios)

a) $8,9_{10}$

b) $81,1_{10}$

2.1.17. Convertir los números $(245,625)_{10}$ y $(1797,223)_{10}$ a binario, octal y hexadecimal245,625₁₀ :1797,223₁₀ :**2.1.18. Convertir el número $(49403180,AF7)_{16}$ a binario, octal y decimal**49403180,AF7₁₆ :**2.1.19. Dado el número $X=(543,21)_6$, expresarlo en hexadecimal con cuatro dígitos fraccionarios y los dígitos enteros que sea necesario****2.1.20. Convertir los siguientes números de base 10 a base 2**

a. 0,0625 :

b. 43,32 :

c. .51 :

2.1.21. Escribir el equivalente de base octal de los siguientes números en base 2

a. 10111100101 :

b. 1101,101 :

c. 1,0111 :

2.1.22. Calcular para las secuencias de 16 bits dadas su representación octal y hexadecimal

A = 0000 0110 0000 0111 :

B = 0000 0000 1101 0110 :

C = 1100 0001 1111 0011 :

D = 1001 0000 0000 1010 :

2.2. Soluciones**2.2.1. Convertir los siguientes números binarios puros a sus equivalentes en base 10**

a) $100110: 2^5+2^2+2 = 38_{10}$

b) $110011: 2^5+2^4+2+1 = 51_{10}$

c) $010111: 2^4+2^2+2+1 = 23_{10}$

d) $101110: 2^5+2^3+2^2+2 = 46_{10}$

e) $110111: 2^5+2^4+2^2+2+1 = 55_{10}$

f) $01100110: 2^6+2^5+2^2+2 = 102_{10}$

g) $10110011: 2^7+2^5+2^4+2+1 = 179_{10}$

h) $0101,11: 2^2+1+2^{-1}+2^{-2} = 5,75_{10}$

i) $1001,10: 2^3+1+2^{-1} = 9,5_{10}$

j) $101010110,001: 2^8+2^6+2^4+2^2+2+2^{-3} = 342,125_{10}$

2.2.2. Convertir los siguientes números decimales a sus equivalentes en binario

a) 9

f) 258,75

<p>b) 64</p> $\begin{array}{r} 9 \mid 2 \\ 1 \ 4 \mid 2 \\ 0 \ 2 \mid 2 \\ 0 \ 1 \end{array} \quad \boxed{1001_2}$ <p>c) 31</p> $\begin{array}{r} 64 \mid 2 \\ 04 \ 32 \mid 2 \\ 0 \ 12 \ 16 \mid 2 \\ 0 \ 0 \ 8 \mid 2 \\ 0 \ 0 \ 4 \mid 2 \\ 0 \ 0 \ 2 \mid 2 \\ 0 \ 1 \end{array} \quad \boxed{100000_2}$ <p>d) 37</p> $\begin{array}{r} 31 \mid 2 \\ 11 \ 15 \mid 2 \\ 1 \ 1 \ 7 \mid 2 \\ 1 \ 3 \mid 2 \\ 1 \ 1 \end{array} \quad \boxed{11111_2}$ <p>e) 131</p> $\begin{array}{r} 37 \mid 2 \\ 17 \ 18 \mid 2 \\ 1 \ 0 \ 9 \mid 2 \\ 1 \ 4 \mid 2 \\ 0 \ 2 \mid 2 \\ 0 \ 1 \end{array} \quad \boxed{100101_2}$ <p>f) 131</p> $\begin{array}{r} 131 \mid 2 \\ 11 \ 65 \mid 2 \\ 1 \ 05 \ 32 \mid 2 \\ 1 \ 12 \ 16 \mid 2 \\ 0 \ 0 \ 8 \mid 2 \\ 0 \ 4 \mid 2 \\ 0 \ 2 \mid 2 \\ 0 \ 1 \end{array} \quad \boxed{10000011_2}$	<p>Parte entera:</p> $\begin{array}{r} 258 \mid 2 \\ 05 \ 129 \mid 2 \\ 18 \ 09 \ 64 \mid 2 \\ 0 \ 1 \ 04 \ 32 \mid 2 \\ 0 \ 0 \ 12 \ 16 \mid 2 \\ 0 \ 0 \ 0 \ 8 \mid 2 \\ 0 \ 0 \ 4 \mid 2 \\ 0 \ 2 \mid 2 \\ 0 \ 1 \end{array}$ <p>Parte fraccionaria:</p> $\begin{array}{r} 0,75 \\ \times 2 \\ \hline 1,50 \\ \text{fin} \end{array} \quad \begin{array}{r} 0,5 \\ \times 2 \\ \hline 1,0 \\ \text{fin} \end{array}$ <p>g) 0,75</p> $\begin{array}{r} 0,75 \\ \times 2 \\ \hline 1,50 \\ \text{fin} \end{array} \quad \begin{array}{r} 0,5 \\ \times 2 \\ \hline 1,0 \\ \text{fin} \end{array}$ <p>h) 1,625</p> $\begin{array}{r} 0,625 \\ \times 2 \\ \hline 1,250 \\ \text{fin} \end{array} \quad \begin{array}{r} 0,25 \\ \times 2 \\ \hline 0,50 \\ \text{fin} \end{array} \quad \begin{array}{r} 0,5 \\ \times 2 \\ \hline 1,0 \\ \text{fin} \end{array}$ <p>i) 19,3125</p> $\begin{array}{r} 19 \mid 2 \\ 1 \ 9 \mid 2 \\ 1 \ 4 \mid 2 \\ 0 \ 2 \mid 2 \\ 0 \ 1 \end{array} \quad \begin{array}{r} 0,3125 \\ \times 2 \\ \hline 0,6250 \\ \text{fin} \end{array} \quad \begin{array}{r} 0,625 \\ \times 2 \\ \hline 1,250 \\ \text{fin} \end{array} \quad \begin{array}{r} 0,25 \\ \times 2 \\ \hline 0,50 \\ \text{fin} \end{array} \quad \begin{array}{r} 0,5 \\ \times 2 \\ \hline 1,0 \\ \text{fin} \end{array}$
--	--

2.2.3. Convertir los siguientes números enteros hexadecimales a decimal

a) 13 : $1 \cdot 16 + 3 = 19_{10}$	c) 3F0: $3 \cdot 16^2 + 15 \cdot 16^1 = 1008_{10}$
b) 65: $6 \cdot 16 + 5 = 101_{10}$	d) DOCE: $13 \cdot 16^3 + 12 \cdot 16^1 + 14 = 53454_{10}$

2.2.4. Convertir los siguientes números reales hexadecimales a decimal

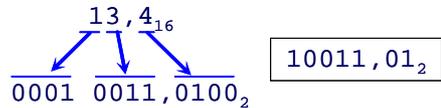
a) 0,2 : $2 \cdot 16^{-1} = 0,125_{10}$	c) F1,A: $15 \cdot 16 + 1 + 10 \cdot 16^{-1} = 241,625_{10}$
b) 12,9: $1 \cdot 16 + 2 + 9 \cdot 16^{-1} = 18,5625_{10}$	d) C8,D: $12 \cdot 16 + 8 + 13 \cdot 16^{-1} = 200,8125_{10}$

2.2.5. Convertir el número hexadecimal 13,4₁₆ a decimal y a binario:

a) Hexadecimal a decimal:

$$1 \cdot 16^1 + 3 \cdot 16^0 + 4 \cdot 16^{-1} = 19,25_{10}$$

Hexadecimal a binario: se sustituye cada cifra por su valor binario



2.2.6. Convertir los siguientes números a binario, octal y decimal

<p>a) 3, A2₁₆</p> <p>Hexadecimal a binario, sustituimos cada cifra por su valor binario</p> <p>Binario a octal: desde la coma agrupamos de 3 en 3 y sustituimos por su valor:</p> <p>hexadecimal a decimal:</p> $3 + 10 \cdot 16^{-1} + 2 \cdot 16^{-2} = 3,6328125_{10}$	<p>b) 1B1, 9₁₆</p> <p>Hexadecimal a binario:</p> <p>Binario a octal: añadiendo ceros a la derecha: 100</p> <p>hexadecimal a decimal:</p> $1 \cdot 16^2 + 11 \cdot 16^1 + 1 + 9 \cdot 16^{-1} = 433,5625_{10}$
--	--

2.2.7. Convertir los siguientes números a binario (8 dígitos fraccionarios máx), octal y hexadecimal (2 dígitos fraccionarios)

<p>a) 8, 9₁₀</p> <p>1000,1110 0110₂ : 8, E₁₆</p> <p>1 000,111 001 100₂ : 10,714₈</p>	<p>b) 81, 1₁₀</p> <p>101 0001,0001 1001₂ : 51,19₁₆</p> <p>1 010 001,000 110 01₂ : 121,062₈</p>
---	---

2.2.8. Convertir el siguiente número a binario, octal y decimal (éste con 5 cifras fraccionarias)

6416213A, 17B₁₆
 0110 0100 0001 0110 0010 0001 0011 1010,0001 0111 1011₂

Agrupamos de tres en tres para convertir e octal:
 001 100 100 000 101 100 010 000 100 111 010,000 101 111 011₂
 14405420472,0573₈

$$6 \cdot 16^7 + 4 \cdot 16^6 + 1 \cdot 16^5 + 6 \cdot 16^4 + 2 \cdot 16^3 + 1 \cdot 16^2 + 3 \cdot 16^1 + 10 + 1 \cdot 16^{-1} + 7 \cdot 16^{-2} + 11 \cdot 16^{-3} = 1679171898,09253_{10}$$

2.2.9. Convertir a base octal

a) 1101110: 156_8	c) 1011001100,11: $1314,6_8$
b) 1001,011: $11,3_8$	d) 101111000,1101: $570,64_8$

2.2.10. Convertir el siguiente número a hexadecimal

$$204231,134_5 = 2 \cdot 5^5 + 4 \cdot 5^3 + 2 \cdot 5^2 + 3 \cdot 5 + 1 + 1 \cdot 5^{-1} + 3 \cdot 5^{-2} + 4 \cdot 5^{-3} = 6816,352_{10}$$

$$6816,352_{10} = 1AA0,5A1CA_{16}$$

2.2.11. Convertir los siguientes números binarios a base hexadecimal y octal

a) 1100 1110 1010 0100 $CEA4_{16}$ 147244_8	b) 1111 0010 0011 1101 $F23D_{16}$ 171075_8	c) 1000 1001 0111 1000 8978_{16} 104570_8
--	--	--

2.2.12. Convertir los siguientes números binarios a sus equivalentes decimales

- | | |
|-----------------------|-----------------------------------|
| a. 001100 : 12_{10} | g. 100001 : 33_{10} |
| b. 000011 : 3_{10} | h. 111000 : 56_{10} |
| c. 011100 : 28_{10} | i. 11110001111 : 1935_{10} |
| d. 111100 : 60_{10} | j. 11100,011 : $28,375_{10}$ |
| e. 101010 : 42_{10} | k. 110011,10011 : $51,59375_{10}$ |
| f. 111111 : 63_{10} | l. 1010101010,1 : $682,5_{10}$ |

2.2.13. Convertir los siguientes números decimales a sus equivalentes binarios

- | | |
|-------------------------|--|
| a. 64 : $100\ 0000_2$ | f. 500 : $1\ 1111\ 0100_2$ |
| b. 100 : $110\ 0100_2$ | g. 34,75 : $10\ 0010,11_2$ |
| c. 111 : $110\ 1111_2$ | h. 25,25 : $1\ 1001,01_2$ |
| d. 145 : $1001\ 0001_2$ | i. 27,1875 : $1\ 1011,0011_2$ |
| e. 255 : $1111\ 1111_2$ | j. 23,1 : $1\ 0111,0\overline{0011}_2$ |

2.2.14. Convertir los siguientes números enteros hexadecimales en sus equivalentes decimales

- | |
|------------------------|
| a. C : 12_{10} |
| b. 9F : 159_{10} |
| c. D52 : 3410_{10} |
| d. 67E : 1662_{10} |
| e. ABCD : 43981_{10} |

2.2.15. Convertir los siguientes números hexadecimales a sus equivalentes decimales

- a) F,4 : 15,25₁₀
 b) D3,E : 211,875₁₀
 c) 111,1 : 273,0625₁₀
 d) 888,8 : 2184,5₁₀
 e) EBA,C : 3770,75₁₀

2.2.16. Convertir los siguientes números a base 10 y base 2

- a) AF315₁₆ : 1010 1111 0011 0001 0101₂ : 717589₁₀
 b) 7326₈ : 111 011 010 110₂ : 3798₁₀

2.2.17. Convertir los números (245,625)₁₀ y (1797,223)₁₀ a binario, octal y hexadecimal

- 245,625₁₀ : 1111 0101,101₂ : 365,5₈ : F5,A₁₆
 1797,223₁₀ : 111 0000 0101,0011 1001 0001...₂ : 3405,1621₈ : 705,391₁₆

2.2.18. Convertir el número (49403180,AF7)₁₆ a binario, octal y decimal

- 49403180,AF7₁₆ :
 0100 1001 0100 0000 0011 0001 1000 0000,1010 1111 0111₂
 11120030600,5367₈ : 1228943744,6853₁₀

2.2.19. Dado el número X=(543,21)₆, expresarlo en hexadecimal con cuatro dígitos fraccionarios y los dígitos enteros que sea necesario

$$207,3611_{10} \rightarrow CF,5C71_{16}$$

2.2.20. Convertir los siguientes números de base 10 a base 2

- a. 0,0625 : 0,0001₂
 b. 43,32 : 10 1011,0101₂
 c. .51 : 0,100000101₂

2.2.21. Escribir el equivalente de base octal de los siguientes números en base 2

- a. 10111100101 : 2745₈
 b. 1101,101 : 15,5₈
 c. 1,0111 : 1,34₈

2.2.22. Calcular para las secuencias de 16 bits dadas su representación octal y hexadecimal

- A = 0000 0110 0000 0111 : 003007₈ : 0607₁₆
 B = 0000 0000 1101 0110 : 000326₈ : 00D6₁₆
 C = 1100 0001 1111 0011 : 140763₈ : C1F3₁₆
 D = 1001 0000 0000 1010 : 110012₈ : 900A₁₆

3. Sistemas de codificación

3.1. Enunciados

3.1.1. Obtener la representación decimal de los siguientes números codificados en BCD

a) 0110 1001 0111 1000

b) 0000 0010 0101 0010 0110

3.1.2. Expresar los números decimales en BCD y en BCD-XS3

a) 88

c) 0

b) 312

d) 1974

3.1.3. Convertir los siguientes números binarios a código Gray

a) 0101

b) 10110

3.1.4. Convertir los siguientes números de código Gray a binario

a) 0101

b) 10110

3.1.5. Obtener la representación decimal de los números (1001 0000 1000 0010) y (1001 0101 0111 0000),(01110001) suponiendo que están codificados en BCD

1001 0000 1000 0010 :

1001 0101 0111 0000,0111 0001 :

3.1.6. Expresar los números decimales 1486, 0, 349 y 37 en código BCD y Exceso-3

1486₁₀:0₁₀:349₁₀:37₁₀:

3.1.7. Convertir los siguientes números binarios a código Gray: (1010),(10010),(10011)

1010:

10010:

10011:

3.1.8. Convertir los siguientes números en código Gray a sus equivalentes binarios (0100),(11111),(011100).

0100_G:11111_G:011100_G:

3.2. Soluciones

3.2.1. Obtener la representación decimal de los siguientes números codificados en BCD

<p>a) 0110 1001 0111 1000</p> <p>BCD a decimal, sustituimos cada cuarteto por su valor decimal</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> $\begin{array}{c} 0110 \\ \downarrow \\ 6 \end{array}$ </div> <div style="text-align: center;"> $\begin{array}{c} 1001 \\ \downarrow \\ 9 \end{array}$ </div> <div style="text-align: center;"> $\begin{array}{c} 0111 \\ \downarrow \\ 7 \end{array}$ </div> <div style="text-align: center;"> $\begin{array}{c} 1000_{BCD} \\ \downarrow \\ 8_{10} \end{array}$ </div> <div style="border: 1px solid black; padding: 2px 5px; margin-left: 20px;">6978₁₀</div> </div>	<p>b) 0000 0010 0101 0010 0110</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> $\begin{array}{c} 0000 \\ \downarrow \\ 0 \end{array}$ </div> <div style="text-align: center;"> $\begin{array}{c} 0010 \\ \downarrow \\ 2 \end{array}$ </div> <div style="text-align: center;"> $\begin{array}{c} 0101 \\ \downarrow \\ 5 \end{array}$ </div> <div style="text-align: center;"> $\begin{array}{c} 0010 \\ \downarrow \\ 2 \end{array}$ </div> <div style="text-align: center;"> $\begin{array}{c} 0110_{BCD} \\ \downarrow \\ 6_{10} \end{array}$ </div> <div style="border: 1px solid black; padding: 2px 5px; margin-left: 20px;">02526₁₀</div> </div>
--	--

3.2.2. Expresar los números decimales en BCD y en BCD-XS3

<p>a) 88</p> <p>Decimal a BCD: sustituimos cada cifra por su equivalente binario</p> <div style="text-align: center;"> $\begin{array}{c} 88 \\ \swarrow \quad \searrow \\ 1000 \quad 1000_{BCD} \end{array}$ </div> <p>BCD a BCD-XS3: se suma 3 a cada cuarteto BCD:</p> <p>1000 1000_{BCD} : 1011 1011_{XS3}</p>	<p>c) 0</p> <p>0000_{BCD}</p> <p>0011_{XS3}</p>
<p>b) 312:</p> <p>0011 0001 0010_{BCD}</p> <p>0110 0100 0101_{XS3}</p>	<p>d) 1974</p> <p>0001 1001 0111 0100_{BCD}</p> <p>0100 1100 1010 0111_{XS3}</p>

3.2.3. Convertir los siguientes números binarios a código Gray

<p>a) 0101</p> <p>Binario a Gray Análisis de izquierda a derecha:</p> <p>1^{er} bit igual en Gray que en binario</p> <div style="text-align: center;"> $\begin{array}{c} 0101_2 \\ \downarrow \\ 0 \end{array}$ </div> <p>En los siguientes bits vemos si el bit del número binario es igual al anterior (del mismo número binario). Si son iguales: 0, si no: 1</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>son distintos</p> $\begin{array}{c} 0101_2 \\ \downarrow \\ 01 \end{array}$ </div> <div style="text-align: center;"> <p>son distintos</p> $\begin{array}{c} 0101_2 \\ \downarrow \\ 011 \end{array}$ </div> <div style="text-align: center;"> <p>son distintos</p> $\begin{array}{c} 0101_2 \\ \downarrow \\ 0111 \end{array}$ </div> <div style="border: 1px solid black; padding: 2px 5px; margin-left: 20px;">0111_G</div> </div>	<p>b) 10110</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>1^{er} bit</p> $\begin{array}{c} 10110_2 \\ \downarrow \\ 1 \end{array}$ </div> <div style="text-align: center;"> <p>distintos</p> $\begin{array}{c} 10110_2 \\ \downarrow \\ 11 \end{array}$ </div> <div style="text-align: center;"> <p>distintos</p> $\begin{array}{c} 10110_2 \\ \downarrow \\ 111 \end{array}$ </div> <div style="text-align: center;"> <p>iguales</p> $\begin{array}{c} 10110_2 \\ \downarrow \\ 1110 \end{array}$ </div> <div style="text-align: center;"> <p>distintos</p> $\begin{array}{c} 10110_2 \\ \downarrow \\ 11101 \end{array}$ </div> </div> <div style="border: 1px solid black; padding: 2px 5px; margin-top: 10px;">11101_G</div>
---	---

3.2.4. Convertir los siguientes números de código Gray a binario

<p>a) 0101</p>	<p>b) 10110</p>
----------------	-----------------

<p>Gray a Binario Análisis de izquierda a derecha:</p> <p>1er bit igual en binario que en Gray</p> $\begin{array}{r} 0101_G \\ \downarrow \\ 0 \end{array} \quad \begin{array}{c} 2 \\ \end{array}$ <p>En los siguientes bits vemos si el último bit binario que hemos puesto es igual al siguiente bit Gray (el de la posición que queremos hallar) Si son iguales: 0, si no: 1</p> <table style="width: 100%; text-align: center;"> <tr> <td style="color: red;">distintos</td> <td>⇨</td> <td style="color: red;">distintos</td> <td>⇨</td> <td style="color: blue;">iguales</td> </tr> <tr> <td>0101_G</td> <td>⇨</td> <td>0101_G</td> <td>⇨</td> <td>0101_B</td> </tr> <tr> <td>$01$₂</td> <td></td> <td>01_B</td> <td></td> <td>0110_B</td> </tr> </table> <div style="border: 1px solid black; display: inline-block; padding: 2px;">0110₂</div>	distintos	⇨	distintos	⇨	iguales	0101_G	⇨	0101_G	⇨	0101_B	01 ₂		01 _B		0110_B	<table style="width: 100%; text-align: center;"> <tr> <td>1er bit</td> <td style="color: red;">distintos</td> <td style="color: blue;">iguales</td> <td style="color: red;">distintos</td> <td style="color: red;">distintos</td> </tr> <tr> <td>10110_G</td> <td>⇨</td> <td>10110_G</td> <td>⇨</td> <td>10110_G</td> <td>⇨</td> <td>10110_G</td> <td>⇨</td> <td>10110_B</td> </tr> <tr> <td>$1$₂</td> <td></td> <td>$11$₂</td> <td></td> <td>$110$₂</td> <td></td> <td>$1101$₂</td> <td></td> <td>$11011$₂</td> </tr> </table> <div style="border: 1px solid black; display: inline-block; padding: 2px;">11011₂</div>	1er bit	distintos	iguales	distintos	distintos	10110_G	⇨	10110_G	⇨	10110_G	⇨	10110_G	⇨	10110_B	1 ₂		11 ₂		110 ₂		1101 ₂		11011 ₂
distintos	⇨	distintos	⇨	iguales																																			
0101_G	⇨	0101_G	⇨	0101_B																																			
01 ₂		01 _B		0110_B																																			
1er bit	distintos	iguales	distintos	distintos																																			
10110_G	⇨	10110_G	⇨	10110_G	⇨	10110_G	⇨	10110_B																															
1 ₂		11 ₂		110 ₂		1101 ₂		11011 ₂																															

3.2.5. Obtener la representación decimal de los números (1001 0000 1000 0010) y (1001 0101 0111 0000),(01110001) suponiendo que están codificados en BCD

1001 0000 1000 0010 : 9082₁₀
 1001 0101 0111 0000,0111 0001 : 9570,71₁₀

3.2.6. Expresar los números decimales 1486, 0, 349 y 37 en código BCD y Exceso-3

1486₁₀: 0001 0100 1000 0110_{BCD} : 0100 0111 1011 1001_{XS3}
 0₁₀: 0000_{BCD} : 0011_{XS3}
 349₁₀: 0011 0100 1001_{BCD} : 0110 0111 1100_{XS3}
 37₁₀: 0011 0111_{BCD} : 0110 1010_{XS3}

3.2.7. Convertir los siguientes números binarios a código Gray: (1010),(10010),(10011)

1010: 1111_{GRAY}
 10010: 11011_{GRAY}
 10011: 11010_{GRAY}

3.2.8. Convertir Convertir los siguientes números en código Gray a sus equivalentes binarios (0100),(11111),(011100).

0100_G: 0111₂
 11111_G: 10101₂
 011100_G: 010111₂

4. Aritmética binaria

4.1. Enunciados

4.1.1. Indicar el rango de un número de 8 bits según las codificaciones siguientes:

- | | |
|--------------------|---------------------|
| a) binario puro: | c) Complemento a 1: |
| b) Signo-Magnitud: | d) Complemento a 2: |

4.1.2. Indicar el resultado de las operaciones y si el resultado de sale de rango (operandos y resultado en Ca2 de 4 bits):

- | | |
|--------------|--------------|
| a) 0011+1100 | f) 0011-1100 |
| b) 0011+0101 | g) 0011-0101 |
| c) 0011+1010 | h) 0011-1010 |
| d) 1011+1111 | i) 1011+0000 |
| e) 1000+1111 | j) 1000-0001 |

4.1.3. Hallar el valor en base 10 de los números $A=01110011$ y $B=11000011$. Calcular también su suma y su diferencia en su misma codificación y en decimal, suponiendo que están codificados en:

- | | |
|---------------------|--------------------|
| a) Magnitud y signo | c) Complemento a 2 |
| b) Complemento a 1 | d) Exceso a 128 |

4.1.4. Utilizando aritmética binaria y habiendo convertido los operandos de base 10 a binario, realizar las siguientes operaciones:

- | | |
|------------|-------------|
| a) 364+112 | c) -364-112 |
| b) 364-112 | d) 121*12 |

4.1.5. Determinar en cuáles de las siguientes operaciones (con operandos representados en Ca2 de 4 bits), el resultado no es correctamente representable, es decir, se produce desbordamiento

- | | |
|--------------|--------------|
| a) 0110+0101 | d) 0100-1110 |
| b) 0000-1111 | e) 1001+1111 |
| c) 1001-1011 | f) 0000+1111 |

4.1.6. Indicar el valor en base 10 de los números $A=1011$ y $B=0101$ suponiendo que están codificados en:

- | | |
|-------------------|--------------------|
| a) Binario puro | c) Complemento a 1 |
| b) Signo magnitud | d) Complemento a 2 |

Para uno de los casos calcular la suma e indicar si hay desbordamiento y por qué

4.1.7. Hallar el valor decimal, la suma y la diferencia de los números binarios $A=11100111$ y $B=10111111$, su suma y diferencia, suponiendo que:

- a) Ambos están representados en Magnitud y signo.
- b) Ambos están representados en Ca2.
- c) Ambos están representados en Ca1.
- d) Ambos están representados en exceso a 128.

4.1.8. Utilizando la aritmética binaria y habiendo convertido previamente a binario los operandos, realizar las siguientes operaciones:

- a. $(695)_{10} + (272)_{10}$
- b. $(695)_{10} - (272)_{10}$
- c. $(272)_{10} * (23)_{10}$

4.1.9. Realizar las siguientes operaciones, suponiendo primero que los sumandos están representados en MS, luego en Ca2 y Ca1

- a. $100110+000100$
- b. $101101111-010000111$
- c. $000010000+11100001$
- d. $10110.1111-11100.111$
- e. $0000.10000+11.100001$

4.1.10. Utilizando la aritmética binaria y suponiendo que los operandos están representados en complemento a 2, realizar las operaciones

- a. $101101111 - 10000111$
- b. $000010000 + 11100001$

4.1.11. Se dispone de un sistema de representación R de 8 bits. Dadas dos cantidades binarias $A=01100110$ y $B=11011001$, se pide realizar la suma $X=A+B$ en binario y comentar el resultado obtenido

4.2. Soluciones

4.2.1. Indicar el rango de un número de 8 bits según las codificaciones siguientes:

- a) binario puro: $[0, 255]$
- b) Signo-Magnitud: $[-127, 127]$
- c) Complemento a 1: $[-127, 127]$
- d) Complemento a 2: $[-128, 127]$

4.2.2. Indicar el resultado de las operaciones y si el resultado de sale de rango (operandos y resultado en Ca2 de 4 bits)

4.2.3. Hallar el valor en base 10 de los números $A=011110011$ y $B=11000011$. Calcular también su suma y su diferencia en su misma codificación y en decimal, suponiendo que están codificados en:

a) Magnitud y signo

<p>SM $A=011110011 \rightarrow 115$ $B=11000011 \rightarrow -67$</p> <p>Suma $S = A + B$ Suma en S-M con sumandos de distinto signo: Al módulo mayor se le resta el menor y se deja el signo del número de mayor módulo</p> $\begin{array}{r} 1110011 \quad 115 \\ -1000011 \quad -67 \\ \hline 0110000 \quad 48 \end{array}$ <p>Módulo $S=00110000$ $S=48$ Signo positivo añadido</p>	<p>Resta en S-M: se cambia el signo del sustraendo y se suma $A=011110011 \rightarrow 115$ $(-B)=01000011 \rightarrow 67$</p> <p>Suma en S-M con ambos sumandos positivos: Se suman los módulos, se comprueba si hay desbordamiento y se añade el signo + al resultado</p> $\begin{array}{r} 1110011 \quad 115 \\ +1000011 \quad 67 \\ \hline 10110110 \quad 182 \end{array}$ <p>desbordamiento Se necesita un bit más (9 bits): $R=010110110$ Signo positivo añadido</p> <p>No es representable en S-M de 8 bits (-127,+127)</p>
---	--

b) Complemento a 1

<p>Ca1 $A=011110011 \rightarrow 115$ $B=11000011 \rightarrow -60$</p> <p>Suma $S = A + B$</p> $\begin{array}{r} 011110011 \quad 115 \\ +11000011 \quad -60 \\ \hline 100110110 \quad 55 \\ + \quad 1 \quad \text{se suma el acarreo} \\ \hline 00110111 \quad 55 \end{array}$ <p>$S=00110111$ $S=55$ Sumandos con signos diferentes: No habrá desbordamiento</p>	<p>Resta $R = A - B = A + (-B) = A + Ca1(B)$ $Ca1(B) = 00111100$</p> $\begin{array}{r} 011110011 \quad 115 \\ +00111100 \quad 60 \\ \hline 10101111 \quad 175 \end{array}$ <p>cambio de signo: desbordamiento Se necesita un bit más (9 bits): $R=010101111$</p> <p>No es representable en Ca1 de 8 bits (-127,+127)</p>
---	--

c) Complemento a 2

<p>Ca2 $A=011110011 \rightarrow 115$ $B=11000011 \rightarrow -61$</p> <p>Suma $S = A + B$</p> $\begin{array}{r} 011110011 \quad 115 \\ +11000011 \quad +(-61) \\ \hline 000110110 \quad 54 \end{array}$ <p>$S=00110110$ $S=54$ se ignora acarreo Sumandos con signos diferentes: No habrá desbordamiento</p>	<p>Resta $R = A - B = A + (-B) = A + Ca2(B)$ $Ca2(B) = 00111101$</p> $\begin{array}{r} 011110011 \quad 115 \\ +00111101 \quad 61 \\ \hline 10110000 \quad 176 \end{array}$ <p>cambio de signo: desbordamiento Se necesita un bit más (9 bits): $R=010110000$</p> <p>No es representable en Ca2 de 8 bits (-128,+127)</p>
--	--

d) Exceso a 128

XS128 $A=01110011 \rightarrow 115_{XS128} \xrightarrow{-128} -13$
 $B=11000011 \rightarrow 195_{XS128} \rightarrow 67$

Suma $S = A + B - 128$

XS128 binario	XS128 decimal	nº verdaderos
01110011	115	-13
+11000011	+ 195	+ 67
100110110	310	54
- 10000000	- 128	
10110110	182 _{XS128}}	

$S=10110110_{XS128}$ Es representable en 8 bits (0,255) Es representable en XS128 de 8 bits (-128,+127)

$S=182_{XS128}=54_{10}$

Resta $B>A$: Hacemos B-A y el resultado se lo restamos a 128

$R = A - B + 128 = 128 - (B - A)$

B-A XS128 binario	B-A XS128 decimal	nº verdaderos
11000011	195	(-13)
-01110011	- 115	+ (-67)
(B-A): 01010000	80	-80

Quiero A-B: en vez de sumar, restamos 128-(B-A)

10000000	128	Representable en XS128 de 8 bits (-128,+127)
- 01010000	- 80	
00110000	48 _{XS128}}	

$R=00110000_{XS128}$ Representable en 8 bits (0,255)

$R=48_{XS128}=-80_{10}$

4.2.4. Utilizando aritmética binaria y habiendo convertido los operandos de base 10 a binario, realizar las siguientes operaciones:

a) 364+112

```

101101100    364
+ 1110000    +112
=====
111011100    → 476
    
```

b) 364-112

```

101101100    364
- 1110000    -112
=====
011111100    → 252
    
```

c) -364-112

Si lo hago en binario puro, sumo los módulos e indico el signo del resultado (el enunciado no especifica cómo va el signo)

```

101101100    364
+ 1110000    +112
=====
111011100    → 476
    
```

Por tanto,

- 111011100 → -476

d) 121*12

1111001	121
x 1100	x 12
=====	=====
0000000	242
0000000	121
1111001	=====
+ 1111001	1452
=====	
10110101100	

4.2.5. Determinar en cuáles de las siguientes operaciones (con operandos representados en Ca2 de 4 bits), el resultado no es correctamente representable, es decir, se produce desbordamiento

<p>a) 0110+0101</p> $\begin{array}{r} 0110 \\ +0101 \\ \hline 1011 \end{array}$ <p>6 +5 11 → No es representable en Ca2 de 4 bits</p> <p>al estar en Ca2 representa a -6 distintos</p> <p>Cambio de signo → desbordamiento</p>	<p>b) 0000-1111</p> <p>Resta en Ca2: Suma con el sustraendo complementado 0000-1111 → 0000+Ca2(1111) Ca2(1111)=0001</p> $\begin{array}{r} 0000 \\ +0001 \\ \hline 0001 \end{array}$ <p>0₁₀ +1₁₀ 1₁₀ → Es representable en Ca2 de 4 bits</p> <p>iguales</p> <p>Sin cambio de signo → Bien</p>	<p>c) 1001-1011</p> <p>1001-1011 → 1001+Ca2(1011) 1001+0101</p> $\begin{array}{r} 1001 \\ +0101 \\ \hline 1110 \end{array}$ <p>-7 +5 -2 → Es representable en Ca2 de 4 bits</p> <p>iguales</p> <p>Sumandos con signos diferentes: No habrá desbordamiento → Bien</p>
<p>d) 0100-1110</p> <p>0100-1110 → 0100+Ca2(1110)</p> $\begin{array}{r} 0100 \\ +0110 \\ \hline 0110 \end{array}$ <p>4 +2 6 → Es representable en Ca2 de 4 bits</p> <p>iguales</p> <p>Sin cambio de signo → Bien</p>	<p>e) 1001+1111</p> $\begin{array}{r} 1001 \\ +1111 \\ \hline 1000 \end{array}$ <p>-7 +(-1) -8 → Representable en Ca2 de 4 bits</p> <p>iguales</p> <p>Sin cambio de signo → Bien</p> <p>se ignora el acarreo</p>	<p>f) 0000+1111</p> $\begin{array}{r} 0000 \\ +1111 \\ \hline 1111 \end{array}$ <p>0₁₀ +(-1)₁₀ -1₁₀ → Representable</p> <p>iguales</p> <p>Sumandos con signos diferentes: No habrá desbordamiento → Bien</p>

4.2.6. Indicar el valor en base 10 de los números A=1011 y B=0101 suponiendo que están codificados en:

- a) Binario puro
- b) Signo-magintud
- c) Complemento a 1
- d) Complemento a 2

Para uno de los casos calcular la suma e indicar si hay desbordamiento y por qué

<p>a) Binario puro</p> <p>$A = 1011_2 = 2^3 + 2^1 + 2^0 = 11_{10}$ $B = 0101_2 = 2^2 + 2^0 = 5_{10}$</p>	<p>SUMA:</p> $\begin{array}{r} 1011 \\ +0101 \\ \hline 10000 \end{array}$ <p>11 +5 16 → 16 no es representable en binario puro de 4 bits. Con 4 bits sólo puedo representar los números del 0 al 15</p> <p>desbordamiento</p>
<p>b) Signo-magintud</p> <p>$A = 1011_{SM} = -(2^3 + 2^0) = -3_{10}$ $B = 0101_{SM} = 2^2 + 2^0 = 5_{10}$</p>	<p>SUMA:</p> <p>En signo-magintud, como son números de distinto signo, se pone el signo de módulo mayor. Y se restan los valores absolutos.</p> <p>Por tener signos diferentes no se va a desbordar:</p> $\begin{array}{r} 101 \\ -011 \\ \hline 010 \end{array}$ <p>5 -3 2</p> <p>S= 0010_{SM}</p>
<p>c) Complemento a 1</p> <p>$A = 1011_{Ca1} = -4_{10}$ $B = 0101_{Ca1} = 5_{10}$</p>	<p>SUMA: Tienen signos diferentes, por tanto no se van a desbordar.</p> <p>signos distintos: no habrá desbordamiento</p> $\begin{array}{r} 1011 \\ +0101 \\ \hline 10000 \\ + \\ 0001 \\ \hline 0001 \end{array}$ <p>-4 +5 1 → representable en Ca1 de 4 bits</p> <p>se suma el acarreo</p> <p>S= 0001_{Ca1}</p>
<p>d) Complemento a 2</p> <p>$A = 1011_{Ca2} = -2^3 + 2^1 + 2^0 = -5_{10}$ $B = 0101_{Ca2} = 2^2 + 2^0 = 5_{10}$</p>	<p>SUMA: Tienen signos diferentes, por tanto no se van a desbordar.</p> <p>signos distintos: no habrá desbordamiento</p> $\begin{array}{r} 1011 \\ +0101 \\ \hline 10000 \end{array}$ <p>-5 +5 5 → representable en Ca2 de 4 bits</p> <p>no se tiene en cuenta el acarreo</p> <p>S= 0000_{Ca2}</p>

4.2.7. Hallar el valor decimal, la suma y la diferencia de los números binarios A=11100111 y B=10111111, su suma y diferencia, suponiendo que:

a. Ambos están representados en Magnitud y signo.

<p>Signo y Magnitud</p> <p>A=11100111 → -103 B=10111111 → -63</p> <hr/> <p>Suma</p> <p>Suma en S-M con ambos sumandos negativos: Se suman los módulos y se niega el resultado</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: right; padding-right: 20px;">1100111</td> <td style="text-align: right;">103</td> <td></td> </tr> <tr> <td style="text-align: right;">+0111111</td> <td style="text-align: right;">+ 63</td> <td style="color: green;">negamos</td> </tr> <tr style="border-top: 1px solid black;"> <td style="text-align: right;">10100110</td> <td style="text-align: right;">166</td> <td style="color: red;">No es representable en S-M de 8 bits</td> </tr> </table> <p style="color: red;">desbordamiento</p> <p style="color: red;">En S-M de 8 bits quedará el número: -38 → Mal</p> <p>Se necesita un bit más para representar el resultado</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">10100110</div> <p style="color: green;">Signo negativo añadido: -166 → Bien</p>	1100111	103		+0111111	+ 63	negamos	10100110	166	No es representable en S-M de 8 bits	<p>Resta R = A - B</p> <p>Resta en S-M: se cambia el signo del sustraendo y se suma</p> <p>A=11100111 → -103 (-B)=00111111 → +63</p> <p>Suma en S-M con sumandos de distinto signo: Al módulo mayor se le resta el menor. El signo del resultado es el del mayor módulo</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: right; padding-right: 20px;">1100111</td> <td style="text-align: right;">103</td> <td style="color: green;">ponemos el signo del mayor</td> </tr> <tr> <td style="text-align: right;">-0111111</td> <td style="text-align: right;">- 63</td> <td style="color: green;">del mayor</td> </tr> <tr style="border-top: 1px solid black;"> <td style="text-align: right;">0101000</td> <td style="text-align: right;">40</td> <td style="color: green;">ponemos el signo del número que tiene mayor módulo (A)</td> </tr> </table> <div style="border: 1px solid black; padding: 2px; display: inline-block;">10101000</div> → R=-40 <div style="border: 1px solid black; padding: 2px; display: inline-block;">R=10101000</div>	1100111	103	ponemos el signo del mayor	-0111111	- 63	del mayor	0101000	40	ponemos el signo del número que tiene mayor módulo (A)
1100111	103																		
+0111111	+ 63	negamos																	
10100110	166	No es representable en S-M de 8 bits																	
1100111	103	ponemos el signo del mayor																	
-0111111	- 63	del mayor																	
0101000	40	ponemos el signo del número que tiene mayor módulo (A)																	

b. Ambos están representados en Ca2.

<p>Ca2</p> <p>A=11100111 → -25 B=10111111 → -65</p> <hr/> <p>Suma S = A + B</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: right; padding-right: 20px;">11100111</td> <td style="text-align: right;">-25</td> </tr> <tr> <td style="text-align: right;">+10111111</td> <td style="text-align: right;">+ (-65)</td> </tr> <tr style="border-top: 1px solid black;"> <td style="text-align: right;">110100110</td> <td style="text-align: right;">-90</td> </tr> </table> <p>S=-90 <div style="border: 1px solid black; padding: 2px; display: inline-block;">S=10100110</div></p> <p style="color: green;">Sin cambio de signo → Bien se ignora el acarreo</p>	11100111	-25	+10111111	+ (-65)	110100110	-90	<p>Resta R=A - B = A + (-B) = A + Ca2(B)</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: right; padding-right: 20px;">11100111</td> <td style="text-align: right;">-25</td> <td style="text-align: right;">65</td> </tr> <tr> <td style="text-align: right;">+01000001</td> <td style="text-align: right;">+ 65</td> <td style="text-align: right;">-25</td> </tr> <tr style="border-top: 1px solid black;"> <td style="text-align: right;">100101000</td> <td style="text-align: right;">40</td> <td style="text-align: right;">40</td> </tr> </table> <p style="color: red;">se ignora acarreo</p> <p style="color: green;">Sumandos con signos diferentes: No habrá desbordamiento → Bien</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">R=00101000</div> R=40	11100111	-25	65	+01000001	+ 65	-25	100101000	40	40
11100111	-25															
+10111111	+ (-65)															
110100110	-90															
11100111	-25	65														
+01000001	+ 65	-25														
100101000	40	40														

c. Ambos están representados en Ca1.

<p>Ca1</p> <p>A=11100111 → -24 B=10111111 → -64</p> <hr/> <p>Suma S = A + B</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: right; padding-right: 20px;">11100111</td> <td style="text-align: right;">-24</td> </tr> <tr> <td style="text-align: right;">+10111111</td> <td style="text-align: right;">+ (-64)</td> </tr> <tr style="border-top: 1px solid black;"> <td style="text-align: right;">110100110</td> <td style="text-align: right;">-88</td> </tr> </table> <p style="color: red;">se suma el acarreo</p> <p style="color: red;">1</p> <p style="color: red;">10100111 → S=-88</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">S=10100111</div> <p style="color: green;">Sin cambio de signo → Bien</p>	11100111	-24	+10111111	+ (-64)	110100110	-88	<p>Resta R=A - B = A + (-B) = A + Ca1(B)</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: right; padding-right: 20px;">11100111</td> <td style="text-align: right;">-24</td> <td style="text-align: right;">64</td> </tr> <tr> <td style="text-align: right;">+01000000</td> <td style="text-align: right;">+ 64</td> <td style="text-align: right;">-24</td> </tr> <tr style="border-top: 1px solid black;"> <td style="text-align: right;">100100111</td> <td style="text-align: right;">40</td> <td style="text-align: right;">40</td> </tr> </table> <p style="color: red;">se suma el acarreo</p> <p style="color: red;">1</p> <p style="color: red;">00101000 → R=40</p> <p style="color: green;">Sumandos con signos diferentes: No habrá desbordamiento → Bien</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">R=00101000</div> R=40	11100111	-24	64	+01000000	+ 64	-24	100100111	40	40
11100111	-24															
+10111111	+ (-64)															
110100110	-88															
11100111	-24	64														
+01000000	+ 64	-24														
100100111	40	40														

d. Ambos están representados en exceso a 128.

XS128 $A=11100111 \rightarrow 231_{XS128} \xrightarrow{-128} 103$
 $B=10111111 \rightarrow 191_{XS128} \xrightarrow{-128} 63$

Suma $S = A + B - 128$

XS128 binario	XS128 decimal	nº verdaderos
11100111	231	103
+10111111	+ 191	+ 63
110100110	422	166
- 10000000	- 128	
100100110	294 _{XS128}	

desbordamiento

No es representable en 8 bits (0,255) No es representable en XS128 de 8 bits (-128,+127)

En XS128 de 8 bits
 $S=00100110$ $S=-38 \rightarrow$ Mal

En XS128 no se suele hacer extensión de signo, porque implica cambio del valor del exceso. Con 9 bits el exceso valdría 256

Resta A y B son positivos y $A > B$, entonces:
 $R = A - B + 128$

XS128 binario	XS128 decimal	nº verdaderos
11100111	231	103
-10111111	- 191	- 63
00101000	040	40
+ 10000000	+ 128	
10101000	168 _{XS128}	

$R=10101000_{XS128}$

Representable en 8 bits (0,255) Representable en XS128 de 8 bits (-128,+127)

Nota: Si quisiésemos restar A a B: $(B - A)$:
A y B son positivos, y el sustraendo (A) es mayor que el minuendo(B), entonces:
Para evitar restar el mayor al menor, hacemos:
 $R = B - A + 128 = 128 - (A - B)$

A-B en XS128 binario	A-B en XS128 decimal	A-B en decimal
11100111	231	103
-10111111	- 191	- 63
(A-B): 00101000	40	40

Quiero B-A: en vez de sumar, restamos $128-(A-B)$ quiero B-A: cambio de signo
 $R = -40$

10000000	128	
- 00101000	- 40	
01011000	88 _{XS128}	

$R=01011000_{XS128}$

4.2.8. Utilizando la aritmética binaria y habiendo convertido previamente a binario los operandos, realizar las siguientes operaciones:

<p>a. $(695)_{10} + (272)_{10}$</p> <pre> 1010110111 695 + 100010000 + 272 ===== 1111000111 → 967 </pre>	<p>b. $(695)_{10} - (272)_{10}$</p> <pre> 1010110111 695 - 100010000 - 272 ===== 0110100111 → 423 </pre>
<p>c. $(272)_{10} * (23)_{10}$</p> <pre> 100010000 272 x 10111 x 23 ===== 100010000 816 100010000 544 100010000 ===== 000000000 6256 + 100010000 ===== 1100001110000 </pre>	

4.2.9. Realizar las siguientes operaciones, suponiendo primero que los sumandos están representados en MS, luego en Ca2 y Ca1

a. 100110+000100

MS A=100110 → -6
B=000100 → 4

Suma en S-M con sumandos de distinto signo: Al módulo mayor se le resta el menor, y se pone el signo del mayor

$$\begin{array}{r} 00110 \\ -00100 \\ \hline 00010 \end{array} \rightarrow S = -2_{10}$$

$$00010 \rightarrow S = 100010$$

Ca2 A=100110 → -26
B=000100 → 4

Suma en Ca2 con sumandos de distinto signo: No hay problema de desbordamiento

$$\begin{array}{r} 100110 \\ +000100 \\ \hline 101010 \end{array} \rightarrow S = -22_{10}$$

$$101010 \rightarrow S = 100010$$

Ca1 A=100110 → -25
B=000100 → 4

Suma en Ca1 con sumandos de distinto signo: No hay problema de desbordamiento

$$\begin{array}{r} 100110 \\ +000100 \\ \hline 101010 \end{array} \rightarrow S = -21_{10}$$

$$101010 \rightarrow S = 100010$$

b. 101101111-010000111

MS A=101101111 → -111
B=010000111 → 135

Resta en S-M: A - B = A + (-B)
Queda una suma con signos negativos: Suma de módulos, resultado con signo negativo, vigilar desbordamiento

$$\begin{array}{r} 01101111 \\ +10000111 \\ \hline 11110110 \end{array} \quad \begin{array}{r} 111 \\ +135 \\ \hline 246 \end{array}$$

No hay acarreo → Bien Representable en MS de 9bits

Se añade el signo: $R = 11110110 \rightarrow R = -246_{10}$

Ca2 A=101101111 → -145
B=010000111 → 135

Resta: Suma del complemento a 2:
Ca2(B)=101111001 → -135
Sumandos con mismo signo: vigilar desbordamiento

$$\begin{array}{r} 101101111 \\ +101111001 \\ \hline 1011101000 \end{array} \quad \begin{array}{r} (-145) \\ +(-135) \\ \hline -280 \end{array}$$

$-280 < -256 \rightarrow$ No es representable en Ca2 de 9bits

Cambio de signo: desbordamiento

Se necesita un bit más (10bits) para representar el resultado:
 $R = 1011101000 \quad R = -280$

Ca1 A=101101111 → -144
B=010000111 → 135

Resta: suma del complemento a 1:
Ca1(B)=101111000 → -135
Sumandos con mismo signo: vigilar desbordamiento

$$\begin{array}{r} 101101111 \\ +101111000 \\ \hline 1011100111 \\ +1011101000 \\ \hline 1011101000 \end{array} \quad \begin{array}{r} (-144) \\ +(-135) \\ \hline -279 \end{array}$$

$-279 < -256 \rightarrow$ No es representable en Ca1 de 9bits

Cambio de signo: desbordamiento

Se necesita un bit más (10bits) para representar el resultado:
 $R = 1011101000 \quad R = -279$

c. 000010000+11100001

MS A=000010000 → 16
B= 11100001 → -97

Ponemos los sumandos con el mismo n° de bits: del tamaño del mayor: 9bits:

A=000010000 → 16
B=101100001 → -97

Suma en S-M con sumandos de distinto signo: Al módulo mayor se le resta el menor, y se pone el signo del mayor

$$\begin{array}{r} 01100001 \\ -00010000 \\ \hline 01010001 \end{array} \quad \begin{array}{r} 97 \\ -16 \\ \hline 81 \end{array}$$

Se añade el signo: $S = -81_{10}$

9 bits: Representable en MS de 8 y 9bits

$S = 101010001$

(en 8 bits):
 $S = 11010001$

Ca2 A=000010000 → 16
B= 11100001 → -31

Ponemos los sumandos con el mismo n° de bits: del tamaño del mayor: 9bits:

A=000010000 → 16
B=111100001 → -31

$$\begin{array}{r} 000010000 \\ +111100001 \\ \hline 111110001 \end{array} \quad \begin{array}{r} 16 \\ +(-31) \\ \hline -15 \end{array}$$

$S = 111110001 \quad S = -15_{10}$

Ca1 A=000010000 → 16
B= 11100001 → -30

Ponemos los sumandos con el mismo n° de bits: del tamaño del mayor: 9bits:

A=000010000 → 16
B=111100001 → -30

$$\begin{array}{r} 000010000 \\ +111100001 \\ \hline 111110001 \end{array} \quad \begin{array}{r} 16 \\ +(-30) \\ \hline -14 \end{array}$$

$S = 111110001 \quad S = -14_{10}$

d. 10110.1111-11100.111

MS A=10110,1111 → -6,9375
 B=11100,111 → -12,875
 A-B=A+(-B)
 (-B)=01100,111 → 12,875
 Suma en S-M con sumandos de distinto signo: Al módulo mayor se le resta el menor, y se pone el signo del mayor

1100,1110	12,8750
-0110,1111	-6,9375
0101,1111	5,9375

R=00101,1111 R=5,9375₁₀

Ca2 A=10110,1111 → -9,0625
 B=11100,111 → -3,125
 Ca2(A)=01001,0001 → 9,0625
 Ca2(B)=00011,001 → 3,125
 A-B=A+Ca2(B)

10110,1111	-9,0625
+00011,0010	+3,1250
11010,0001	-5,9375

R=11010,0001 R=-5,9375₁₀

Ca1 A=10110,1111 → -9,0
 B=11100,111 → -3,0
 Ca1(A)=01001,0000 → 9,0
 Ca1(B)=00011,000 → 3,0
 A-B = A+Ca1(B)

10110,1111	-9,0
+00011,0000	+3,0
11001,1111	-6,0

R=11001,1111 R=-6,0₁₀

e. 0000.10000+11.100001

MS A=0000,10000 → 0,5
 B=11,100001 → -1,515625
 Suma en S-M con sumandos de distinto signo: Al módulo mayor se le resta el menor, y se pone el signo del mayor

1,100001	1,515625
-0,100000	-0,500000
1,000001	1,015625

Se añade el signo:
S=11,000001 S=-1,015625₁₀
 Con 3 bits para la parte entera (igual que A):
S=1001,000001

Ca2 A=0000,10000 → 0,5
 B=11,100001 → -0,484375
 Ca2(B)=00,011111 → 0,484375

0000,100000	0,500000
+1111,100001	-0,484375
0000,000001	0,015625

se ignora el acarreo
S=0000,000001 S=0,015625

Ca1 A=0000,10000 → 0,5
 B=11,100001 → -0,46875
 Ca1(B)=00,011110 → 0,46875

0000,100000	0,500000
+1111,100001	-0,46875
0000,000001	0,03125

se suma el acarreo
S=0000,000010 S=0,03125

4.2.10. Utilizando la aritmética binaria y suponiendo que los operandos están representados en complemento a 2, realizar las operaciones

<p>a. 101101111 - 10000111</p> <p>Ca2 A=101101111 → -145 B=100001111 → -121 A-B=A+Ca2(B) Ca2(B)=01111001 → 121</p> <table style="margin-left: 40px;"> <tr><td>101101111</td><td>-145</td></tr> <tr><td>+001111001</td><td>+121</td></tr> <tr><td>111101000</td><td>-24</td></tr> </table> <p>R= 111101000 R=-24</p>	101101111	-145	+001111001	+121	111101000	-24	<p>b. 000010000 + 11100001</p> <p>Ca2 A=000010000 → 16 B=11100001 → -31</p> <table style="margin-left: 40px;"> <tr><td>000010000</td><td>16</td></tr> <tr><td>+111100001</td><td>-31</td></tr> <tr><td>111110001</td><td>-15</td></tr> </table> <p>S= 111110001 S=-15</p>	000010000	16	+111100001	-31	111110001	-15
101101111	-145												
+001111001	+121												
111101000	-24												
000010000	16												
+111100001	-31												
111110001	-15												

4.2.11. Se dispone de un sistema de representación R de 8 bits. Dadas dos cantidades binarias A=01100110 y B=11011001, se pide realizar la suma X=A+B en binario y comentar el resultado obtenido

- a. Suponiendo que el sistema R es binario puro.
- b. Suponiendo que el sistema R es magnitud y signo.
- c. Suponiendo que el sistema R es complemento a 2.

Binario puro $X=A+B$

A=01100110 → 102
 B=11011001 → 217

01100110	102
+11011001	+217
100111111	319

Desbordamiento, se necesitan 9 bits
 $319 > 255 (2^8-1) \rightarrow$ No es representable en binario puro de 8 bits

En binario puro de 8 bits:

X= 00111111 X=63 **Mal**

En binario puro de 9 bits:

X=100111111 X=319 **Bien**

MS $X=A+B$

A=01100110 → 102
 B=11011001 → -89

Suma en S-M con sumandos de distinto signo: Al módulo mayor se le resta el menor, y se pone el signo del mayor

1100110	102
-1011001	- 89
0001101	13

Se añade el signo de A $13 < 127 (2^7-1)$: es representable en SM de 8 bits

X= 00001101

Ca2 $X=A+B$

A=01100110 → 102
 B=11011001 → -39

Suma en Ca2 con sumandos de distinto signo: No hay problema de desbordamiento

01100110	102
+11011001	- 39
100111111	63

se ignora el acarreo $53 < 127 (2^7-1)$: es representable en Ca2 de 8 bits

X= 00111111

5. Aritmética en BCD

5.1. Enunciados

5.1.1. Sumar en BCD (código 8-4-2-1) los siguientes números:

- | | |
|------------|----------------|
| a) 4 + 3 | d) 9 + 8 |
| b) 51 + 28 | e) 1489 + 5791 |
| c) 8 + 5 | |

5.1.2. Restar en BCD (código 8-4-2-1) los siguientes números:

- | | |
|------------|----------------|
| a) 4 - 3 | c) 18 - 9 |
| b) 59 - 21 | d) 7340 - 3649 |

5.2. Soluciones

5.2.1. Sumar en BCD (código 8-4-2-1) los siguientes números:

- | | |
|------------|----------------|
| a) 4 + 3 | d) 9 + 8 |
| b) 51 + 28 | e) 1489 + 5791 |
| c) 8 + 5 | |

La suma en BCD (código 8-4-2-1) se realiza de los cuartetos menos significativos a los más significativos (como la suma en decimal: se empieza por las unidades y se va hacia la izquierda).

Cada cuarteto se suma, y si el resultado es menor o igual que 9 (1001_2) y si no hay acarreo, se deja como está.

a) 4 + 3

$$\begin{array}{r}
 S = 4 + 3 \\
 \begin{array}{r}
 \text{BCD} \\
 0100 \\
 +0011 \\
 \hline
 0111
 \end{array}
 \qquad
 \begin{array}{r}
 \text{decimal} \\
 4 \\
 +3 \\
 \hline
 7
 \end{array} \\
 7 \leq 9 \rightarrow \text{correcto} \quad \boxed{S = 0111}
 \end{array}$$

b) 51 + 28

$$\begin{array}{r}
 S = 51 + 28 \\
 \begin{array}{r}
 \text{BCD} \\
 \begin{array}{r}
 \text{decenas} \\
 0101 \\
 +0010 \\
 \hline
 0111
 \end{array}
 \qquad
 \begin{array}{r}
 \text{unidades} \\
 0001 \\
 +1000 \\
 \hline
 1001
 \end{array}
 \end{array}
 \qquad
 \begin{array}{r}
 \text{decimal} \\
 51 \\
 +28 \\
 \hline
 79
 \end{array} \\
 \begin{array}{r}
 7 \qquad 9 \\
 \boxed{S = 0111 \ 1001}
 \end{array}
 \end{array}$$

Sin embargo, si el resultado es mayor que nueve, significa que nos llevamos una, y hay que tenerlo en cuenta, ya que en la suma binaria pura tenemos hasta el número 15 con los cuatro bits del BCD. Entonces, tenemos que sumar 6 al resultado: Esto es, del 1010_2 al 1111_2 (10_{10} al 15_{10} en decimal) pasarán a ser: 0001 0000 al 0001 0101.

a) $4 - 3$ y b) $59 - 21$

<p>$R = 4 - 3$</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: left;">BCD</th> <th style="text-align: left;">decimal</th> </tr> <tr> <td style="border-bottom: 1px solid black;">0100</td> <td style="border-bottom: 1px solid black;">4</td> </tr> <tr> <td style="border-bottom: 1px solid black;">-0011</td> <td style="border-bottom: 1px solid black;">-3</td> </tr> <tr> <td style="border-bottom: 1px solid black;">0001</td> <td style="border-bottom: 1px solid black;">1</td> </tr> <tr> <td>1</td> <td>→ correcto</td> </tr> </table> <p style="text-align: right; border: 1px solid black; display: inline-block; padding: 2px;">R = 0001</p>	BCD	decimal	0100	4	-0011	-3	0001	1	1	→ correcto	<p>$R = 59 + 21$</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="3" style="text-align: center;">BCD</th> </tr> <tr> <th style="text-align: left;">decenas</th> <th style="text-align: left;">unidades</th> <th style="text-align: left;">decimal</th> </tr> <tr> <td style="border-bottom: 1px solid black;">0101</td> <td style="border-bottom: 1px solid black;">1001</td> <td style="border-bottom: 1px solid black;">59</td> </tr> <tr> <td style="border-bottom: 1px solid black;">+0010</td> <td style="border-bottom: 1px solid black;">+0001</td> <td style="border-bottom: 1px solid black;">+21</td> </tr> <tr> <td style="border-bottom: 1px solid black;">0011</td> <td style="border-bottom: 1px solid black;">1000</td> <td style="border-bottom: 1px solid black;">38</td> </tr> <tr> <td>3</td> <td>8</td> <td></td> </tr> </table> <p style="text-align: right; border: 1px solid black; display: inline-block; padding: 2px;">R = 0011 1000</p>	BCD			decenas	unidades	decimal	0101	1001	59	+0010	+0001	+21	0011	1000	38	3	8	
BCD	decimal																												
0100	4																												
-0011	-3																												
0001	1																												
1	→ correcto																												
BCD																													
decenas	unidades	decimal																											
0101	1001	59																											
+0010	+0001	+21																											
0011	1000	38																											
3	8																												

Si hay acarreo se debe de restar 6 al resultado de la resta del cuarteto

c) $18 - 9$

$R = 18 - 9$

BCD		decimal
decenas	unidades	18
0001	1000	- 9
-0000	-1001	09
0001	1111	
-0001	-0110	
0000	1001	
0	9	

R = 0000 1001

El caso de d combina ambas situaciones.

d) $7340 - 3649$

$R = 7340 - 3649$

BCD				decimal
u. millar	centenas	decenas	unidades	7340
0111	0011	0100	0000	-3649
-0011	-0110	-0100	-1001	3691
0100	1101	0000	0111	
-0001	-0110	-0001	-0110	
0011	0111	1111	0001	
3	6	9	1	

R = 0011 0110 1001 0001

El ejemplo c) sí tiene acarreo. Cuando esto sucede se le suma 3 al resultado:

<p>$S = 4 + 6$</p> <p style="text-align: center;">binario XS3</p> <p>El acarreo: 1 se pasa a XS3 (sumándole 3)</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <pre> 0111 +1001 ----- (1)0000 + 0011 ----- 0100 </pre> <p>Acarreo a decenas (1) → acarreo sumar 3</p> <p>$4_{XS3} \rightarrow 1_{10}$ $3_{XS3} \rightarrow 0_{10}$</p> <p>$S_{XS3} = 0100\ 0011$</p> </div> <div style="width: 35%; border-left: 1px solid black; padding-left: 5px;"> <p style="text-align: center;">decimal</p> <p style="text-align: center;">4 +6 ----- 10</p> </div> </div>	<p>$S = 4 + 6$</p> <p style="text-align: center;">binario XS3</p> <p>Como no hay decenas pero hay acarreo, se puede crear una suma 0+0 y se opera igual</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <pre> decenas unidades 0011 0111 +0011 +1001 ----- 0110 0000 +0001 + 0011 ----- 0111 0011 -0011 -0011 ----- 0100 0011 </pre> <p>Acarreo a decenas (1) → acarreo sumar 3</p> <p>no acarreo restar 3 $3_{XS3} \rightarrow 0_{10}$</p> <p>$4_{XS3} \rightarrow 1_{10}$</p> <p>$S_{XS3} = 0100\ 0011$</p> </div> <div style="width: 35%; border-left: 1px solid black; padding-left: 5px;"> <p style="text-align: center;">decimal</p> <p style="text-align: center;">4 +6 ----- 10</p> </div> </div>
---	--

El ejemplo c) se ha tratado el acarreo de dos formas diferentes. Como los sumandos no tienen decenas y hay un acarreo a decenas, se puede, o bien poner directamente el acarreo como un uno en XS3 : 0100. O bien hacer una suma de ceros en las decenas, y operar normalmente (restando 3 al resultado por no haber acarreo en decenas).

e)

$S = 4068 + 1639$

binario XS3

u. millar	centenas	decenas	unidades	decimal
<pre> 0111 +0100 ----- 0 1011 - 0011 ----- 0 1000 </pre> <p>no acarreo restar 3</p> <p>$8_{XS3} \rightarrow 5_{10}$</p>	<pre> 0011 +1001 ----- 0 1100 + 0001 ----- 0 1101 - 0011 ----- 1010 </pre> <p>Acarreo a centenas (1) → acarreo sumar 3</p> <p>no acarreo restar 3</p> <p>$10_{XS3} \rightarrow 7_{10}$</p>	<pre> 1001 +0110 ----- 0 1111 + 0001 ----- 0 0000 + 0011 ----- 0011 </pre> <p>Acarreo a decenas (1) → acarreo sumar 3</p> <p>acarreo sumar 3</p> <p>$3_{XS3} \rightarrow 0_{10}$</p>	<pre> 1011 +1100 ----- 0111 + 0011 ----- 1010 </pre> <p>acarreo sumar 3</p> <p>$10_{XS3} \rightarrow 7_{10}$</p>	<p style="text-align: center;">decimal</p> <p style="text-align: center;">4068 +1639 ----- 5707</p>
$S_{XS3} = 1000\ 1010\ 0011\ 1010$				

6.2.2. Restar en BCD XS-3 los siguientes números:

- a) $4 - 3$
- c) $64 - 15$
- b) $65 - 13$
- d) $705 - 1072$

De la misma manera que en BCD, se debe tomar como minuendo el de mayor módulo, y el resultado tendrá su mismo signo.

Al contrario que en la suma, en la resta en XS3 se suma 3 si no hay acarreo y se resta 3 si lo hay.

a) Este ejemplo no tiene acarreo y por lo tanto se le suma 3:

$R = 4 - 3$

binario XS3

```

0111
-0110
-----
0001
+0011
-----
0100
            
```

No acarreo sumar 3

$4_{XS3} \rightarrow 1_{10}$ $R_{XS3} = 1010$

b) Este ejemplo es igual que el anterior, sólo que con dos cifras decimales:

$R = 65 - 13$

binario XS3

decenas	unidades	decimal
<pre> 1001 -0100 ----- 0101 +0011 ----- 1000 </pre> <p>No acarreo sumar 3</p> <p>$8_{XS3} \rightarrow 5_{10}$</p>	<pre> 1000 -0110 ----- 0010 +0011 ----- 0101 </pre> <p>No acarreo sumar 3</p> <p>$5_{XS3} \rightarrow 2_{10}$</p>	<p style="text-align: center;">decimal</p> <p style="text-align: center;">65 +13 ----- 52</p>
$R_{XS3} = 1000\ 0101$		

c) Este ejemplo tiene acarreo en las unidades. Por tanto al resultado de las unidades se le resta 3 y se pasa el acarreo a las decenas (restando):

R = 64 - 15

binario XS3		decimal
decenas	unidades	64
1001	0111	-15
-0100	-1000	-----
0101	1111	49
-0001	-0011	
0100	1100	
+0011	12 _{XS3} → 9 ₁₀	
0111	7 _{XS3} → 4 ₁₀	
R _{XS3} = 0111 1100		

Annotations: "Acarreo a decenas" with a circled 1 and arrow from units to tens; "no acarreo sumar 3" with arrow to units; "acarreo: restar 3" with arrow from units to tens.

d) Este ejemplo combina ambas situaciones y además el sustraendo es mayor que el minuendo.

R = 705 - 1072

Restamos 1072-705 y se cambia el signo al resultado

binario XS3				decimal
u millar	centenas	decenas	unidades	1072
0100	0011	1010	0101	- 705
-0011	-1010	-0011	-1000	-----
0001	1001	0111	1101	0367
-0001	-0011	-0001	-0011	↓
0000	0110	0110	1010	-367
+0011	6 _{XS3} → 3 ₁₀	+0011	10 _{XS3} → 7 ₁₀	
0011	3 _{XS3} → 0 ₁₀	1001	9 _{XS3} → 6 ₁₀	
R _{XS3} = - 0011 0110 1001 1010				

↑
Se añade signo negativo (el enunciado tendría que especificar el formato)

7. Algebra de Boole y funciones lógicas

7.1. Enunciados

7.1.1. Utilizando las leyes de De Morgan, obtener una expresión en forma de sumas de productos para las siguientes funciones:

$$a) F = \overline{(x + y)(\overline{xy} + z)}$$

$$b) F = \overline{(\overline{\overline{x \cdot \overline{y}} + xz}) \cdot (\overline{\overline{x} + \overline{y} \cdot z})}$$

7.1.2. Aplicando las leyes de De Morgan, obtener el complemento de las siguientes funciones

$$a) f = (x + \overline{y})(yz + x\overline{y})$$

$$b) g = \overline{y}(x + z) + y(\overline{xz} + x\overline{z})$$

$$c) h = x\overline{y}(\overline{x} + z)(yz + x\overline{y})$$

7.1.3. Verificar, mediante manipulaciones algebraicas adecuadas, las siguientes igualdades, justificando cada uno de los pasos haciendo referencia a un postulado o a un teorema

$$a) (x + \overline{y} + xy)(x + \overline{y})\overline{xy} = 0$$

$$b) (x + \overline{y} + x\overline{y})(xy + \overline{xz} + yz) = xy + \overline{x}yz$$

7.1.4. Simplificar la siguiente función lógica por métodos algebraicos

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

7.1.5. Simplificar por Karnaugh la función cuya expresión en términos canónicos es:

$$f(x, y, z) = \sum_3 m(3, 5, 6)$$

7.1.6. Utilizando los mapas de Karnaugh, simplificar las siguientes funciones de conmutación, obtenerlas en función de suma de productos o producto de sumas:

$$a) f(w, x, y, z) = \sum m(5, 6, 9, 10)$$

$$b) f(x, y, z) = \sum m(2, 3, 4, 5, 6, 7)$$

$$c) f(x, y, z) = \sum m(2, 4, 5, 6)$$

$$d) f(w, x, y, z) = \sum m(3, 6, 7, 11, 12, 14, 15)$$

7.1.7. Demostrar que las tres funciones elementales AND, OR y NOT pueden realizarse mediante las funciones NAND y NOR

7.1.8. Comprobar las siguientes relaciones relativas a la función EXOR:

a) $\begin{cases} x \oplus x = 0 \\ x \oplus \bar{x} = 1 \end{cases}$ b) $\begin{cases} x \oplus 0 = x \\ x \oplus 1 = \bar{x} \end{cases}$

c) $x \oplus y = z \Rightarrow x \oplus z = y$ d) $x \oplus y = z \Rightarrow x \oplus y \oplus z = 0$

7.1.9. Obtener la tabla de verdad que corresponde a las siguientes funciones de conmutación expresadas algebraicamente:

a) $F = xy + \bar{x}z + y\bar{z}$ b) $G = (\bar{x} + \bar{z})(y + z)$

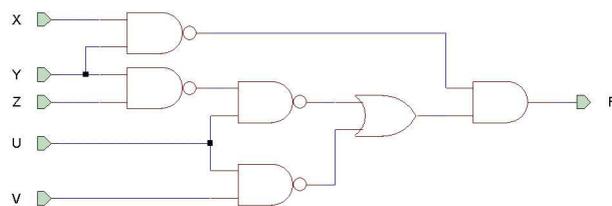
7.1.10. Para cada una de las funciones dadas a continuación, dibujar un circuito con puertas AND, OR Y NOT que la sintetice:

a) $F = \bar{x}yz + \bar{y}(\bar{x}z + z)$
 b) $G = (x + \bar{y} + \bar{z})(\bar{x} + yz)$
 c) $H = (\bar{x}y + xz)(\bar{x} + yz)$

7.1.11. Para cada circuito obtener un circuito equivalente con el mínimo número de puertas lógicas:



7.1.12. Obtener la función de conmutación sintetizada por el circuito de la figura:



7.1.13. Para la siguiente función lógica obtener:

- 1) Obtener mediante operaciones lógicas las formas canónicas en 1ª Forma Normal (1FN) y 2ª Forma Normal (2FN)
- 2) Elaborar la tabla de verdad del circuito
- 3) Hacer el mapa de Karnaugh
- 4) Obtener a partir del mapa de Karnaugh la ecuación simplificada en suma de productos (SOP) y productos de sumas (POS)
- 5) Dibujar las representaciones en puertas lógicas del circuito correspondiente a la 1FN, 2FN, y las correspondientes de SOP y POS obtenidas del mapa de Karnaugh

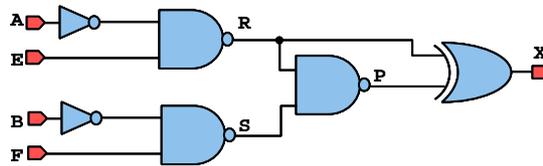
$$F(A, B, C, D) = \left(\left(\overline{\overline{A + \overline{A \cdot (B \cdot C)}}} \right) + B \cdot C \right) \cdot \overline{C \cdot D}$$

7.1.14. Obtener los mismos puntos del apartado anterior para la siguiente función lógica:

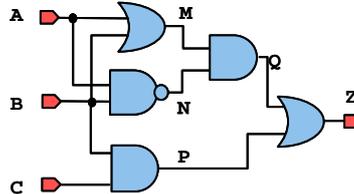
$$F(A, B, C) = \overline{\overline{A \cdot B + \overline{A \cdot C} + B \cdot A + B \cdot C}} + \overline{A + \overline{B}}$$

7.1.15. Del circuito siguiente obtener:

- 1) La ecuación de la salida en suma de productos
- 2) La tabla de verdad del circuito
- 3) Hacer el mapa de Karnaugh
- 4) Obtener a partir del mapa de Karnaugh la ecuación simplificada en suma de productos (SOP) y productos de sumas (POS)
- 5) Dibujar el esquema en puertas de las ecuaciones en suma de productos y producto de sumas simplificados



7.1.16. Obtener los mismos puntos del apartado anterior para el siguiente circuito:



7.2. Soluciones

7.2.1. Utilizando las leyes de De Morgan, obtener una expresión en forma de sumas de productos para las siguientes funciones:

a) $F = \overline{(x + y)(x \cdot \overline{y} + z)} = \overline{(x + y)} + \overline{(x \cdot \overline{y} + z)} = \overline{x} \cdot \overline{y} + x \cdot \overline{y} + z = \overline{y} + z$

b) $F = \overline{(\overline{x \cdot \overline{y} + x \cdot z}) \cdot (\overline{x + \overline{y} \cdot z})} = (\overline{x \cdot \overline{y} + x \cdot z}) + (\overline{x + \overline{y} \cdot z}) = \overline{x \cdot \overline{y}} + \overline{x} + x \cdot z + \overline{y \cdot z} = \overline{x} + x \cdot z + \overline{y} \cdot z$

Hasta aquí es suficiente, aunque todavía se puede simplificar más:

$$= \overline{x} + x \cdot z + 1 \cdot \overline{y} \cdot z = \overline{x} + x \cdot z + (x + \overline{x}) \cdot \overline{y} \cdot z = \overline{x} + x \cdot z + x \cdot \overline{y} \cdot z + \overline{x} \cdot \overline{y} \cdot z =$$

$$= (\overline{x} + \overline{x} \cdot \overline{y} \cdot z) + (x \cdot z + x \cdot \overline{y} \cdot z) = \overline{x}(1 + \overline{y} \cdot z) + x \cdot z \cdot (1 + \overline{y}) = \overline{x} \cdot 1 + x \cdot z \cdot 1 = \overline{x} + x \cdot z = \overline{x} + z$$

El último paso aplicando el teorema de absorción: $\overline{x} + x \cdot z = \overline{x} + z$. Demostración:

$$\overline{x} + x \cdot z = \overline{x} + \overline{x} + x \cdot z = \overline{x} + \overline{x} \cdot 1 + x \cdot z = \overline{x} + \overline{x} \cdot (\overline{z} + z) + x \cdot z = \overline{x} + (\overline{x} \cdot \overline{z} + \overline{x} \cdot z) + x \cdot z =$$

$$= (\overline{x} + \overline{x} \cdot \overline{z}) + (\overline{x} \cdot z + x \cdot z) = \overline{x}(1 + \overline{z}) + z \cdot (\overline{x} + x) = \overline{x} \cdot 1 + z \cdot 1 = \overline{x} + z$$

7.2.2. Aplicando las leyes de De Morgan, obtener el complemento de las siguientes funciones

a) $f = (x + \bar{y})(yz + x\bar{y})$

$$\begin{aligned} \bar{f} &= \overline{(x + \bar{y})(yz + x\bar{y})} = \overline{(x + \bar{y})} + \overline{(yz + x\bar{y})} = \bar{x} \cdot y + \overline{yz \cdot x\bar{y}} = \bar{x} \cdot y + (\bar{y} + \bar{z}) \cdot (\bar{x} + y) = \\ &= \bar{x} \cdot y + (\bar{x} \cdot \bar{y} + y \cdot \bar{y} + \bar{x} \cdot \bar{z} + y \cdot \bar{z}) = (\bar{x} \cdot y + \bar{x} \cdot \bar{y}) + 0 + \bar{x} \cdot \bar{z} + y \cdot \bar{z} = \\ &= \bar{x} + \bar{x} \cdot \bar{z} + y \cdot \bar{z} = \bar{x} + y \cdot \bar{z} \end{aligned}$$

b) $g = \bar{y}(x + z) + y(\bar{x}z + x\bar{z})$

$$\begin{aligned} \bar{g} &= \overline{\bar{y}(x + z) + y(\bar{x}z + x\bar{z})} = \overline{\bar{y}(x + z)} \cdot \overline{y(\bar{x}z + x\bar{z})} = (\bar{\bar{y}} + \overline{(x + z)}) \cdot (\bar{y} + \overline{(\bar{x}z + x\bar{z})}) = \\ &= (y + \bar{x} \cdot \bar{z}) \cdot (\bar{y} + (\bar{x}z \cdot \bar{x}\bar{z})) = (y + \bar{x} \cdot \bar{z}) \cdot (\bar{y} + (x + \bar{z}) \cdot (\bar{x} + z)) = \\ &= (y + \bar{x} \cdot \bar{z}) \cdot (\bar{y} + x \cdot \bar{x} + x \cdot z + \bar{x} \cdot \bar{z} + z \cdot \bar{z}) = (y + \bar{x} \cdot \bar{z}) \cdot (\bar{y} + 0 + x \cdot z + \bar{x} \cdot \bar{z} + 0) = \\ &= (y + \bar{x} \cdot \bar{z}) \cdot (\bar{y} + x \cdot z + \bar{x} \cdot \bar{z}) = y \cdot \bar{y} + x \cdot y \cdot z + \bar{x} \cdot y \cdot \bar{z} + \bar{x} \cdot \bar{y} \cdot \bar{z} + x \cdot \bar{x} \cdot z \cdot \bar{z} + \bar{x} \cdot \bar{x} \cdot \bar{z} \cdot \bar{z} = \\ &= 0 + x \cdot y \cdot z + (\bar{x} \cdot y \cdot \bar{z} + \bar{x} \cdot \bar{y} \cdot \bar{z}) + 0 + \bar{x} \cdot \bar{z} = x \cdot y \cdot z + (\bar{x} \cdot \bar{z}) + \bar{x} \cdot \bar{z} = x \cdot y \cdot z + \bar{x} \cdot \bar{z} \end{aligned}$$

c) $h = x\bar{y}(\bar{x} + z)(yz + x\bar{y})$

$$\begin{aligned} \bar{h} &= \overline{x\bar{y}(\bar{x} + z)(yz + x\bar{y})} = \bar{x} + y + \overline{(\bar{x} + z)} + \overline{(yz + x\bar{y})} = \bar{x} + y + x \cdot \bar{z} + \overline{yz \cdot x\bar{y}} = \\ &= \bar{x} + y + x \cdot \bar{z} + (\bar{y} + \bar{z}) \cdot (\bar{x} + y) = \bar{x} + y + x \cdot \bar{z} + (\bar{x} \cdot \bar{y} + y \cdot \bar{y} + \bar{x} \cdot \bar{z} + y \cdot \bar{z}) = \\ &= \bar{x} + y + x \cdot \bar{z} + \bar{x} \cdot \bar{y} + \bar{x} \cdot \bar{z} + y \cdot \bar{z} = (\bar{x} + \bar{x} \cdot \bar{y} + \bar{x} \cdot \bar{z}) + (y + y \cdot \bar{z}) + x \cdot \bar{z} = \bar{x} + y + x\bar{z} \end{aligned}$$

7.2.3. Verificar, mediante manipulaciones algebraicas adecuadas, las siguientes igualdades, justificando cada uno de los pasos haciendo referencia a un postulado o a un teorema

<p>a) $(x + \bar{y} + xy)(x + \bar{y})\bar{x}y = 0$</p> $\begin{aligned} &(x + \bar{y} + xy) \cdot (x + \bar{y}) \cdot \bar{x} \cdot y \\ &\downarrow \text{teorema de absorción } (x + \bar{y} + xy) \cdot (x + \bar{y}) = x + \bar{y} \\ &(x + \bar{y}) \cdot (x + \bar{y}) \cdot \bar{x} \cdot y \\ &\downarrow \text{teorema de idempotencia } (x + \bar{y}) \cdot (x + \bar{y}) = (x + \bar{y}) \\ &(x + \bar{y}) \cdot \bar{x} \cdot y = \underbrace{x \cdot \bar{x}}_0 \cdot y + \underbrace{\bar{x} \cdot y}_0 = 0 \\ &\downarrow \text{teorema de identidad } 0 \cdot y = 0 \end{aligned}$	<p>b) $(x + \bar{y} + x\bar{y})(xy + \bar{x}z + yz) = xy + \bar{x} \cdot \bar{y}z$</p> $\begin{aligned} &(x + \bar{y} + x\bar{y}) \cdot (xy + \bar{x}z + yz) \\ &\downarrow \text{teorema de absorción } (x + \bar{y} + x\bar{y}) = x \\ &x \cdot (xy + \bar{x}z + yz) \\ &\downarrow \text{teorema de identidad } x \cdot yz = (x + x)yz = \\ &\text{elemento neutro } yz = 1 \cdot yz = (x + x)yz = \\ &\text{propiedad distributiva } = xy\bar{z} + xyz \\ &(x + \bar{y}) \cdot (xy + \bar{x}z + \bar{x}yz + xyz) \\ &\downarrow \text{teorema de absorción } \begin{cases} xy + x \cdot \bar{x}yz = xy \\ \bar{x}z + \bar{x}yz = \bar{x}z \end{cases} \\ &(x + \bar{y}) \cdot (xy + \bar{x}z) \\ &\downarrow \text{propiedad distributiva} \\ &\underbrace{xx}_x xy + \underbrace{x\bar{x}}_0 z + \underbrace{xy\bar{y}}_0 + \bar{x}yz \Rightarrow xy + \bar{x}yz \end{aligned}$
--	---

7.2.4. Simplificar la siguiente función lógica por métodos algebraicos

$$\begin{aligned} f(A, B, C, D) &= \overline{A \cdot (\bar{B} + C) + \bar{B} \cdot \bar{D} + A \cdot (\bar{C} + \bar{D})} \cdot \bar{B} \cdot (C + B) \cdot A = \\ &= \overline{A \cdot (\bar{B} + C)} \cdot \bar{B} \cdot \bar{D} \cdot A \cdot (\bar{C} + \bar{D}) \cdot \bar{B} \cdot ((C + B) + \bar{A}) = \\ &= (\bar{A} + \overline{(\bar{B} + C)}) \cdot (B + D) \cdot (\bar{A} + \overline{(\bar{C} + \bar{D})} + B) \cdot (\bar{C} \cdot \bar{B} + \bar{A}) = \\ &= (\bar{A} + B \cdot \bar{C}) \cdot (B + D) \cdot (\bar{A} + B + C \cdot D) \cdot (\bar{B} \cdot \bar{C} + \bar{A}) = \\ &= (\bar{A} + B) \cdot (\bar{A} + \bar{C}) \cdot (B + D) \cdot (\bar{A} + B + C) \cdot (\bar{A} + B + D) \cdot (\bar{A} + \bar{B}) \cdot (\bar{A} + \bar{C}) = \\ &= [(\bar{A} + B) \cdot (\bar{A} + B + C) \cdot (\bar{A} + B + D)] \cdot [(\bar{A} + \bar{C}) \cdot (\bar{A} + \bar{C})] \cdot (\bar{A} + \bar{B}) \cdot (B + D) = \end{aligned}$$

$$\begin{aligned}
 &= [\bar{A} + B][\bar{A} + \bar{C}](\bar{A} + \bar{B}) \cdot (B + D) = \\
 &= [(\bar{A} + B) \cdot (\bar{A} + \bar{B})](\bar{A} + \bar{C}) \cdot (B + D) = \\
 &= [\bar{A}] \cdot (\bar{A} + \bar{C}) \cdot (B + D) = \bar{A} \cdot (B + D) = \bar{A} \cdot B + \bar{A} \cdot D
 \end{aligned}$$

7.2.5. Simplificar por Karnaugh la función cuya expresión en términos canónicos es:

$$f(x, y, z) = \sum_3 m(3,5,6)$$

tabla de verdad

X	Y	Z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

m_3
 m_5
 m_6

		YZ			
		00	01	11	10
x	0	0	0	1	0
	1	0	1	0	1

$$f(X,Y,Z) = X\bar{Y}Z + \bar{X}YZ + XY\bar{Z}$$

7.2.6. Utilizando los mapas de Karnaugh, simplificar las siguientes funciones de conmutación, obtenerlas en función de suma de productos o producto de sumas

a) $f(w,x,y,z) = \sum m(5,6,9,10)$

tabla de verdad

W	X	Y	Z	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

m_5
 m_6
 m_9
 m_{10}

SOP

		YZ			
		00	01	11	10
wx	00	0	0	0	0
	01	0	1	0	1
	11	0	0	0	0
	10	0	1	0	1

$$f(W,X,Y,Z) = \bar{W}X\bar{Y}Z + \bar{W}XY\bar{Z} + W\bar{X}\bar{Y}Z + W\bar{X}Y\bar{Z}$$

POS

		YZ			
		00	01	11	10
wx	00	0	0	0	0
	01	0	1	0	1
	11	0	0	0	0
	10	0	1	0	1

$$f(W,X,Y,Z) = (W+X) \cdot (\bar{W}+\bar{X}) \cdot (Y+Z) \cdot (\bar{Y}+\bar{Z})$$

b) $f(x,y,z) = \sum m(2,3,4,5,6,7)$

tabla de verdad

X	Y	Z	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

m_2
 m_3
 m_4
 m_5
 m_6
 m_7

SOP

		YZ			
		00	01	11	10
x	0	0	0	1	1
	1	1	1	1	1

$$f(X,Y,Z) = X + Y$$

POS

		YZ			
		00	01	11	10
x	0	0	0	1	1
	1	1	1	1	1

$$f(X,Y,Z) = X + Y$$

La Z no interviene en la función

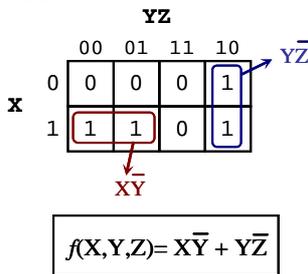
c) $f(x,y,z) = \sum m(2,4,5,6)$

tabla de verdad

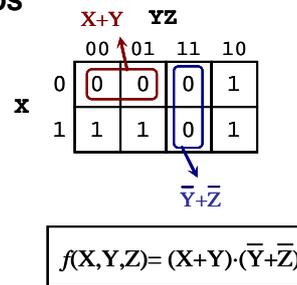
X	Y	Z	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

m_2
 m_4
 m_5
 m_6

SOP

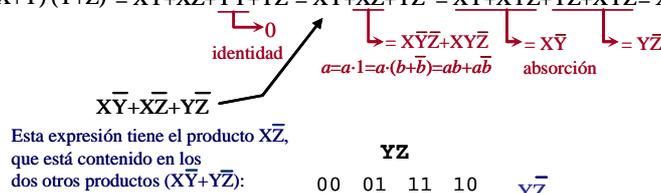


POS

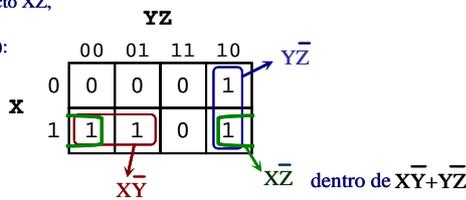


Comprobación de igualdad (no es necesaria):

$$f(X,Y,Z) = (X+Y) \cdot (\bar{Y}+\bar{Z}) = X\bar{Y} + X\bar{Z} + Y\bar{Y} + Y\bar{Z} = X\bar{Y} + X\bar{Z} + Y\bar{Z} = X\bar{Y} + X\bar{Y}\bar{Z} + Y\bar{Z} + XY\bar{Z} = X\bar{Y} + Y\bar{Z}$$



Esta expresión tiene el producto $X\bar{Z}$, que está contenido en los dos otros productos ($X\bar{Y} + Y\bar{Z}$):



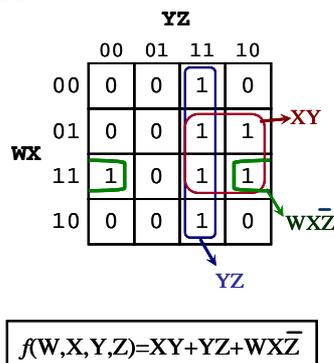
d) $f(w,x,y,z) = \sum m(3,6,7,11,12,14,15)$

tabla de verdad

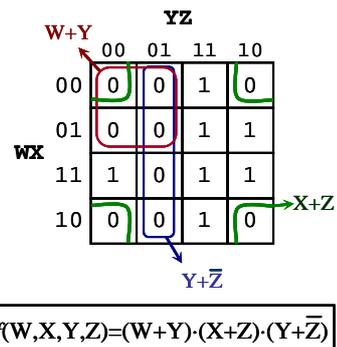
W	X	Y	Z	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

m_3
 m_6
 m_7
 m_{11}
 m_{12}
 m_{14}
 m_{15}

SOP

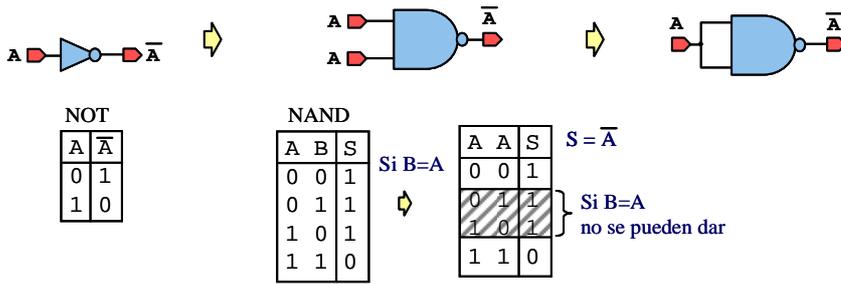


POS

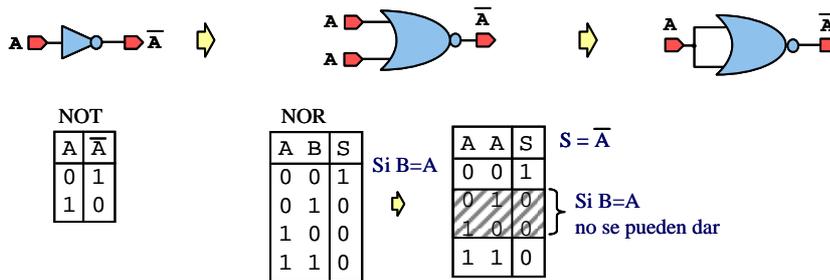


7.2.7. Demostrar que las tres funciones elementales AND, OR y NOT pueden realizarse mediante las funciones NAND y NOR

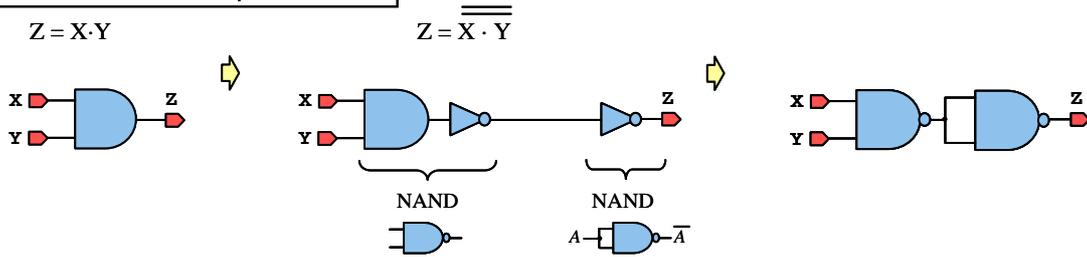
Puerta NOT (inversor) hecha con puertas NAND



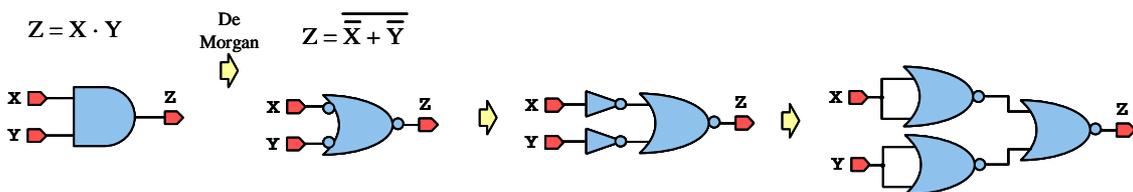
Puerta NOT (inversor) hecha con puertas NOR



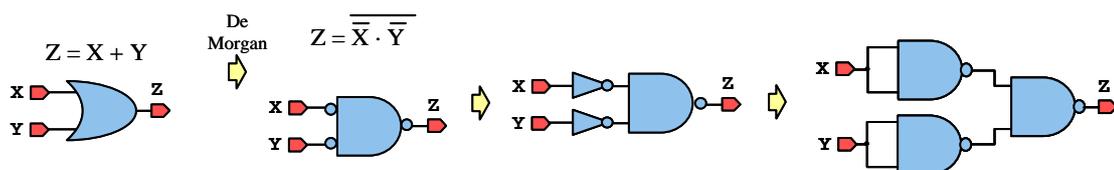
Puerta AND hecha con puertas NAND



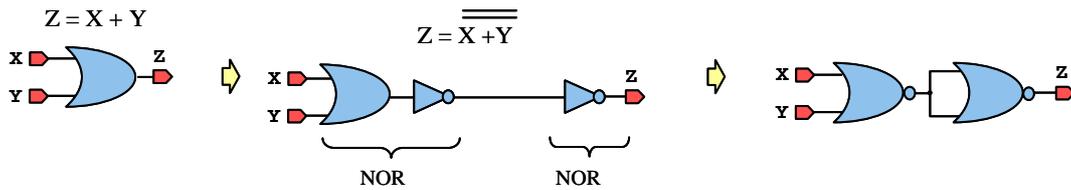
Puerta AND hecha con puertas NOR



Puerta OR hecha con puertas NAND



Puerta OR hecha con puertas NOR



7.2.8. Comprobar las siguientes relaciones relativas a la función EXOR:

a) $x \oplus x = 0$

XOR		
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

$x \oplus x$
 $A=x$ y $B=x$
 entonces hay 2 posibilidades:
 $A=B=0 \rightarrow S=0$
 $A=B=1 \rightarrow S=0$
 Por tanto $x \oplus x = 0$

Otra manera: Se extrae la 1FN de la XOR:
 $A \oplus B = \overline{A}B + A\overline{B}$
 Se sustituye $A=x$, $B=x$
 $x \oplus x = \overline{x}x + x\overline{x} = 0$
 0 0 teorema de identidad

$x \oplus \overline{x} = 1$

XOR		
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

$x \oplus \overline{x}$
 $A=x$ y $B=\overline{x}$
 entonces hay 2 posibilidades:
 $A=0$; $B=1 \rightarrow S=1$
 $A=1$; $B=0 \rightarrow S=1$
 Por tanto $x \oplus \overline{x} = 1$

Otra manera: Se extrae la 1FN de la XOR:
 $A \oplus B = \overline{A}B + A\overline{B}$
 Se sustituye $A=x$, $B=\overline{x}$
 $x \oplus \overline{x} = \overline{x}\overline{x} + x\overline{\overline{x}} = \overline{x} + x = 1$
 $\overline{x} \cdot \overline{x} = \overline{x}$

b) $x \oplus 0 = x$

XOR		
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

$x \oplus 0$
 $A=x$ y $B=0$
 entonces hay 2 posibilidades:
 $A=0 \rightarrow S=0$
 $A=1 \rightarrow S=1$
 $S=A$
 Por tanto $x \oplus 0 = x$

Otra manera: Se extrae la 1FN de la XOR:
 $A \oplus B = \overline{A}B + A\overline{B}$
 Se sustituye $A=x$, $B=0$
 $x \oplus 0 = \overline{x} \cdot 0 + x \cdot \overline{0} = x \cdot 1 = x$
 0

$x \oplus 1 = \overline{x}$

XOR		
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

$x \oplus 1$
 $A=x$ y $B=1$
 entonces hay 2 posibilidades:
 $A=0 \rightarrow S=1$
 $A=1 \rightarrow S=0$
 $S=\overline{A}$
 Por tanto $x \oplus 1 = \overline{x}$

Otra manera: Se extrae la 1FN de la XOR:
 $A \oplus B = \overline{A}B + A\overline{B}$
 Se sustituye $A=x$, $B=1$
 $x \oplus 1 = \overline{x} \cdot 1 + x \cdot \overline{1} = \overline{x} + x \cdot 0 = \overline{x}$
 0

c) $x \oplus y = z \Rightarrow x \oplus z = y$

d) $x \oplus y = z \Rightarrow x \oplus y \oplus z = 0$

$x \oplus y = z \Rightarrow x \oplus y \oplus z = 0$

x	y	z=x⊕y	w=(x⊕y)⊕z
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	0

$w = x \oplus y \oplus z = \underbrace{(x \oplus y)}_z \oplus z = z \oplus z = 0$

x	y	$z=x\oplus y$	$w=x\oplus z$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

$y = w$

$z = x \oplus y = \bar{x}y + x\bar{y}$
 $w = x \oplus z = \bar{x}z + x\bar{z} = \bar{x}(\bar{x}y + x\bar{y}) + x(\bar{x}\bar{y} + x\bar{y})$
 $= \bar{x}\bar{x}y + \bar{x}x\bar{y} + x\bar{x}\bar{y} + xx\bar{y} = \bar{x}y + x(x+\bar{y})\cdot(\bar{x}+\bar{y})$
 $= \bar{x}y + x(\bar{x}\bar{y} + x\bar{y} + \bar{x}y + y\bar{y}) = \bar{x}y + x\bar{y} + x\bar{y} + x\bar{y} = \bar{x}y + xy = (\bar{x}+x)y = y$

Por tanto

Es equivalente a:

7.2.9. Obtener la tabla de verdad que corresponde a las siguientes funciones de conmutación expresadas algebraicamente:

a) $F = xy + \bar{x}z + y\bar{z}$

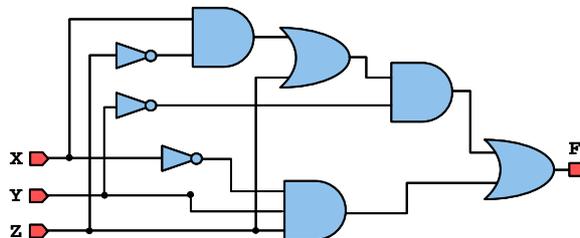
X	Y	Z	XY	$\bar{X}Z$	$Y\bar{Z}$	F
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	0	1	1
0	1	1	0	1	0	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	1	1
1	1	1	1	0	0	1

b) $G = (\bar{x} + \bar{z})(y + z)$

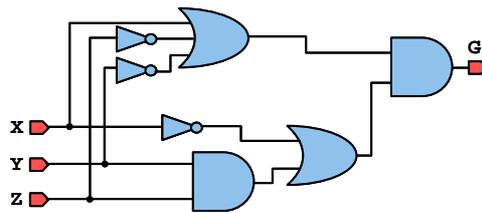
X	Y	Z	$\bar{X} + \bar{Z}$	$Y + Z$	G
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	1	1	1
1	1	1	0	1	0

7.2.10. Para cada una de las funciones dadas a continuación, dibujar un circuito con puertas AND, OR Y NOT que la sintetice:

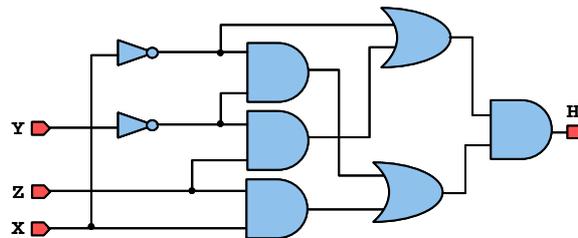
a) $F = \bar{x}yz + \bar{y}(xz + z)$



b) $G = (x + \bar{y} + \bar{z})(\bar{x} + yz)$

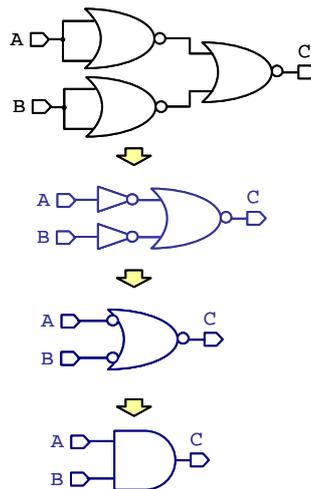


c) $H = (\bar{x}\bar{y} + xz)(\bar{x} + \bar{y}z)$

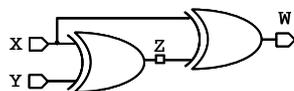


7.2.11. Para cada circuito obtener un circuito equivalente con el mínimo número de puertas lógicas:

a)



b)



analizando la tabla de verdad vemos que $W=Y$

X	Y	Z = X ⊕ Y	W = X ⊕ Z
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

$y = w$

Por tanto no se necesita ninguna puerta lógica

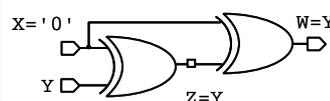


También se puede analizar de otra manera :

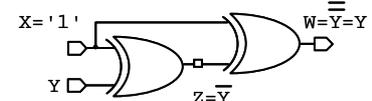
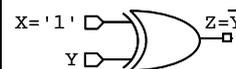
En la XOR,
si $X=0$ ' entonces $Z=Y$
(deja pasar la otra señal)



Por tanto

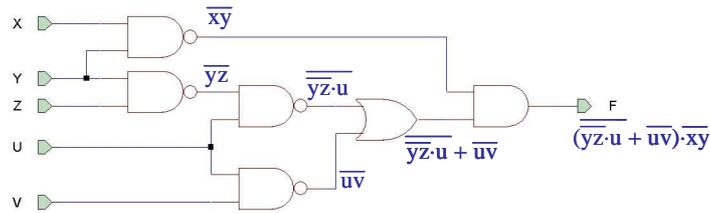


Si $X=1$ ' entonces $Z=\bar{Y}$
(niega la otra señal)



Independientemente de X la salida siempre será Y:
 $W = Y$

7.2.12. Obtener la función de conmutación sintetizada por el circuito de la figura:



Por tanto la función de conmutación es: $F = (\overline{yz \cdot u} + \overline{uv}) \cdot \overline{xy}$

7.2.13. Para la siguiente función lógica obtener:

- 1) Obtener mediante operaciones lógicas las formas canónicas en 1ª Forma Normal (1FN) y 2ª Forma Normal (2FN)
- 2) Elaborar la tabla de verdad del circuito
- 3) Hacer el mapa de Karnaugh
- 4) Obtener a partir del mapa de Karnaugh la ecuación simplificada en suma de productos (SOP) y productos de sumas (POS)
- 5) Dibujar las representaciones en puertas lógicas del circuito correspondiente a la 1FN, 2FN, y las correspondientes de SOP y POS obtenidas del mapa de Karnaugh

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

1) Obtención de la 1FN y 2FN

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

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

Por el teorema de absorción $A + \overline{A} \cdot \overline{B} = A + \overline{B}$

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

Tenemos la ecuación en Producto de Sumas (POS), ahora para cada producto vemos qué variable falta, y le añadimos el producto de dicha variable por su complemento (que es igual a 0) para obtener la 2FN:

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

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

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

Quitando los repetidos obtenemos la 2FN:

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

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

Para obtener la 1FN multiplicamos el producto de sumas:

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

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

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

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

Ya tenemos la ecuación en Suma de Productos (SOP), ahora para cada suma, vemos qué variable le falta y le multiplicamos la suma de dicha variable por su complemento (que da 1), obteniendo así la 1FN:

$$\begin{aligned} \overline{B} \cdot \overline{C} + \overline{B} \cdot \overline{D} + A \cdot \overline{C} &= (A + \overline{A}) \cdot \overline{B} \cdot \overline{C} \cdot (D + \overline{D}) + (A + \overline{A}) \cdot \overline{B} \cdot (C + \overline{C}) \cdot \overline{D} + A \cdot (B + \overline{B}) \cdot \overline{C} \cdot (D + \overline{D}) = \\ &= A \cdot \overline{B} \cdot \overline{C} \cdot D + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot C \cdot \overline{D} + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + \\ &\quad + A \cdot B \cdot \overline{C} \cdot D + A \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot C \cdot D + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \end{aligned}$$

Quitando los repetidos obtenemos la 1FN:

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

2) Elaboración de la tabla de verdad

La tabla de verdad la obtenemos directamente de la 1FN ó 2FN. Si tomamos la 1FN, cada uno de los minterminos aporta un '1' a la tabla de verdad. Las posiciones que no se hayan rellenado irán con '0'.

Si tomamos la 2FN, cada uno de los maxiterminos aporta un '0' a la tabla de verdad. Se debe recordar, que en la tabla de verdad a partir de maxiterminos se ponen los literales complementados. En el resto de posiciones se ponen '1'.

El resultado debe ser el mismo haciéndolo por la 1FN y la 2FN. Y por tanto, la suma del número de minterminos y maxiterminos debe ser igual al número de posibilidades: 2^N (siendo N el número de variables). Para mostrarlo, en la figura se muestra la obtención de la tabla de verdad por la 1FN y la 2FN (no hace falta hacer dos tablas de verdad).

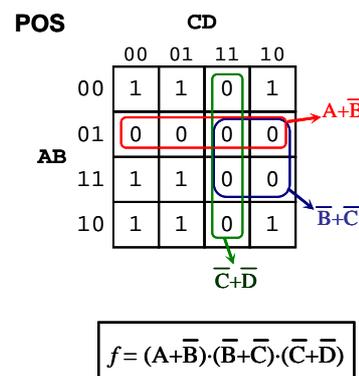
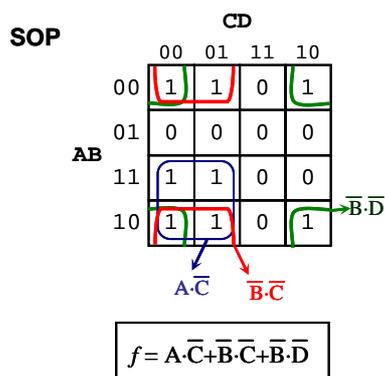
Por la 1FN

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

Por la 2FN

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

3) Mapa de Karnaugh

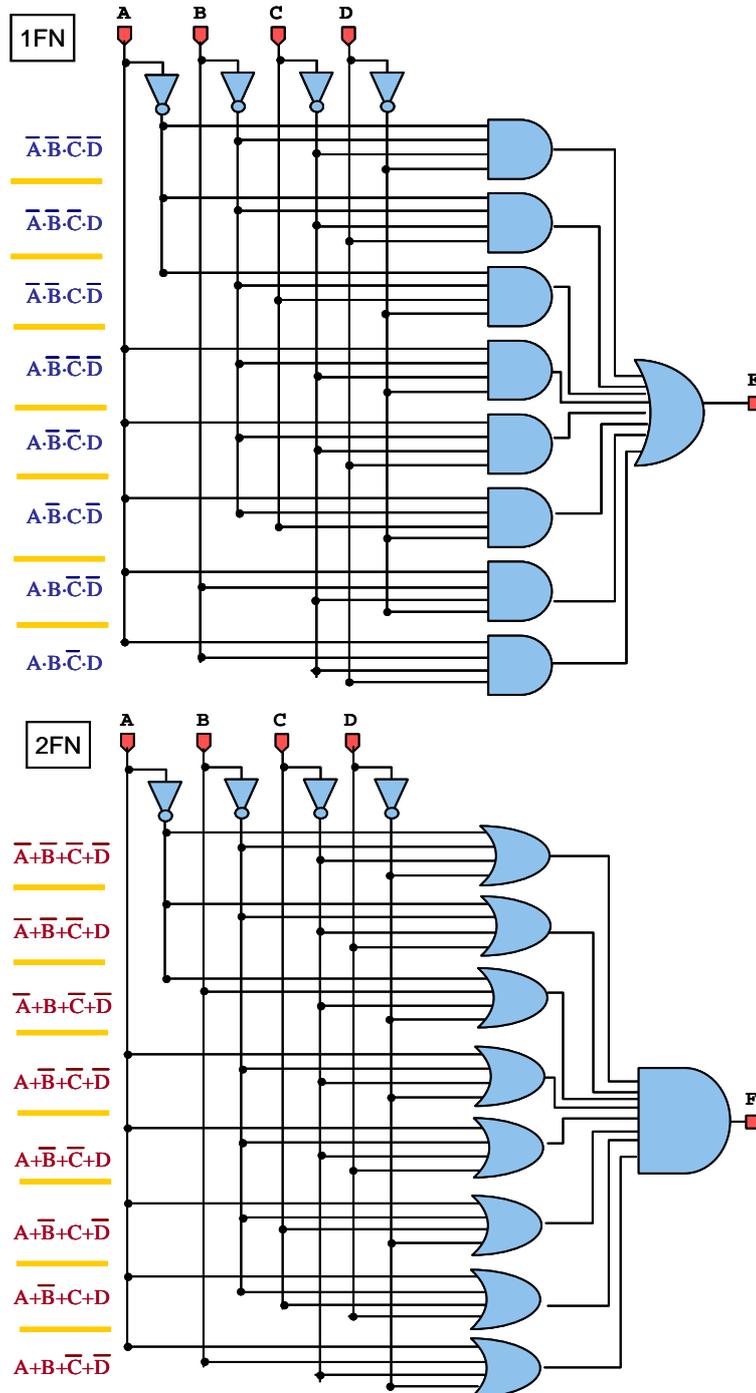


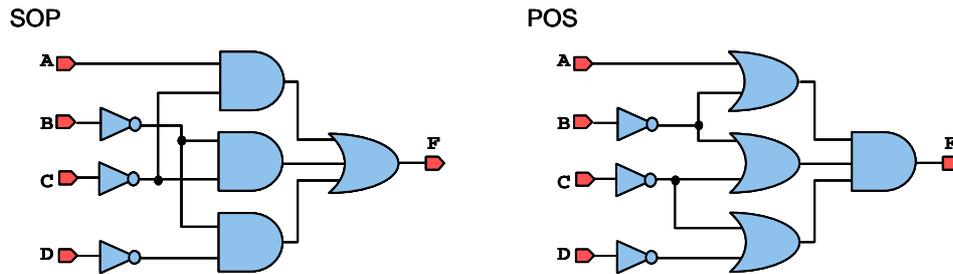
4) Ecuaciones:

En suma de productos $f = A\bar{C} + \bar{B}\bar{C} + \bar{B}\bar{D}$

En producto de sumas $f = (A + \bar{B}) \cdot (\bar{B} + \bar{C}) \cdot (\bar{C} + \bar{D})$

5) Esquemas en puertas lógicas:





Se ve que las formas normales son mucho más complejas que las SOP y POS obtenidas por Karnaugh

7.2.14. Para la siguiente función lógica obtener:

- 1) Obtener mediante operaciones lógicas las formas canónicas en 1ª Forma Normal (1FN) y 2ª Forma Normal (2FN)
- 2) Elaborar la tabla de verdad del circuito
- 3) Hacer el mapa de Karnaugh
- 4) Obtener a partir del mapa de Karnaugh la ecuación simplificada en suma de productos (SOP) y productos de sumas (POS)
- 5) Dibujar las representaciones en puertas lógicas del circuito correspondiente a la 1FN, 2FN, y las correspondientes de SOP y POS obtenidas del mapa de Karnaugh

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

1) Obtención de la 1FN y 2FN

$$F = \left((\overline{\overline{A} \cdot B}) \cdot (\overline{\overline{A} \cdot C}) \cdot (\overline{B \cdot A}) \cdot (\overline{B \cdot C}) \right) + \overline{A + B} = (A + \overline{B}) \cdot (A + \overline{C}) \cdot (\overline{A} + \overline{B}) \cdot (\overline{B} + \overline{C}) + \overline{A + B} =$$

$$\left((A + \overline{B}) \cdot (\overline{A} + \overline{B}) \right) \cdot (A + \overline{C}) \cdot (\overline{B} + \overline{C}) + \overline{A + B} =$$

Como $(A + \overline{B}) \cdot (\overline{A} + \overline{B}) = (A \cdot \overline{A}) + \overline{B} = 0 + \overline{B} = \overline{B}$ entonces:

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

Para ponerla en SOP:

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

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

Obtenemos la 1FN:

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

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

1FN: $A \cdot \overline{B} \cdot C + A \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A + B} \cdot C$

Para la 2FN obtenemos la ecuación en POS:

$$(\overline{B} \cdot (A + \overline{C}) \cdot (\overline{B} + \overline{C})) + \overline{A + B} = \left((\overline{B} \cdot (A + \overline{C}) \cdot (\overline{B} + \overline{C})) + \overline{A} \right) \left((\overline{B} \cdot (A + \overline{C}) \cdot (\overline{B} + \overline{C})) + B \right) =$$

$$= \left((\overline{A} + \overline{B}) \cdot (A + \overline{A} + \overline{C}) \cdot (\overline{A} + \overline{B} + \overline{C}) \right) \left((B + \overline{B}) \cdot (A + B + \overline{C}) \cdot (B + \overline{B} + \overline{C}) \right) =$$

$$= \left((\overline{A} + \overline{B}) \cdot (1 + \overline{C}) \cdot (\overline{A} + \overline{B} + \overline{C}) \right) \left(1 \cdot (A + B + \overline{C}) \cdot (1 + \overline{C}) \right) = (\overline{A} + \overline{B}) \cdot (\overline{A} + \overline{B} + \overline{C}) \cdot (A + B + \overline{C}) =$$

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

Obtenemos la 2FN:

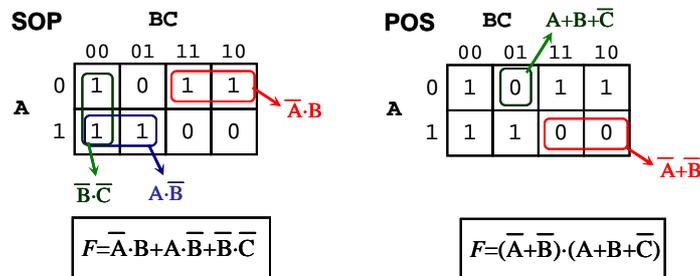
$$= (\bar{A} + \bar{B} + (C \cdot \bar{C})) \cdot (A + B + \bar{C}) = (\bar{A} + \bar{B} + C) \cdot (\bar{A} + \bar{B} + \bar{C}) \cdot (A + B + \bar{C})$$

2FN: $(\bar{A} + \bar{B} + \bar{C}) \cdot (\bar{A} + \bar{B} + C) \cdot (A + B + \bar{C})$

2) Elaboración de la tabla de verdad

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

3) Mapa de Karnaugh

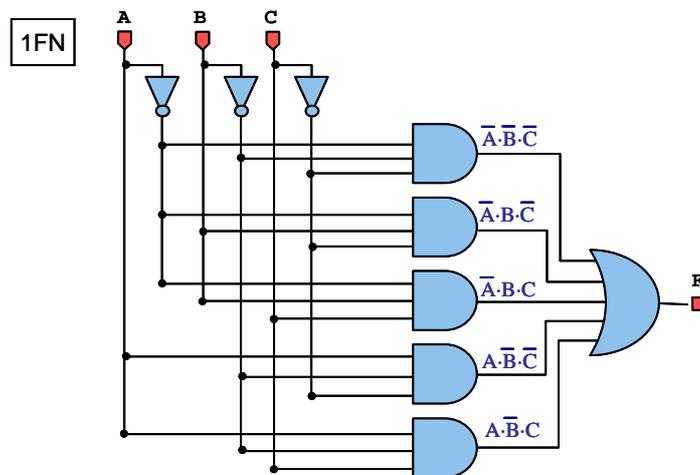


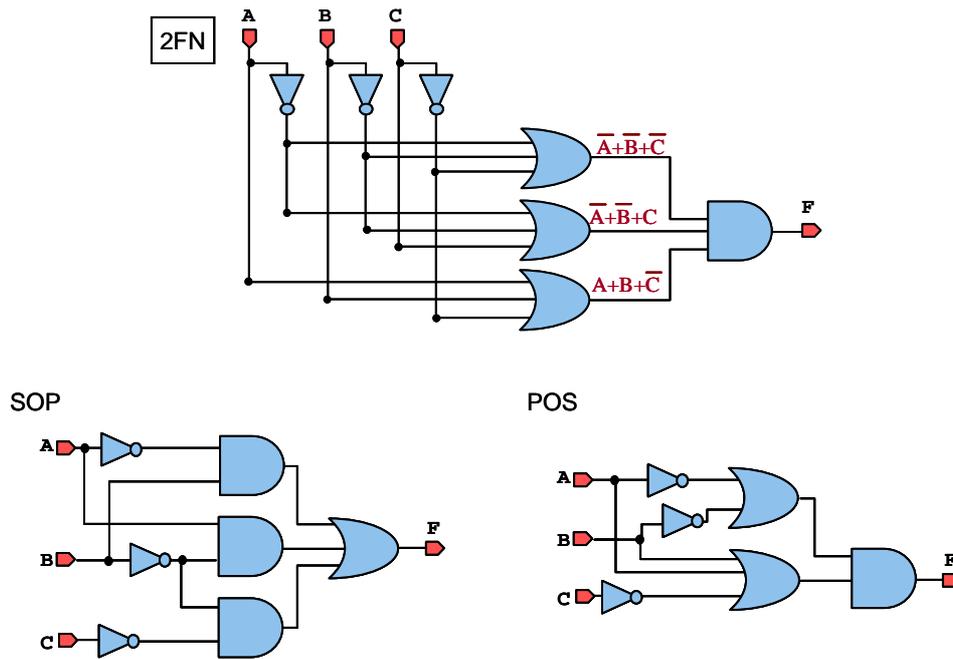
4) Ecuaciones

En suma de productos $f = \bar{A} \cdot B + A \cdot \bar{B} + \bar{B} \cdot C$

En producto de sumas $f = (\bar{A} + \bar{B}) \cdot (A + B + \bar{C})$

5) Esquemas en puertas lógicas:





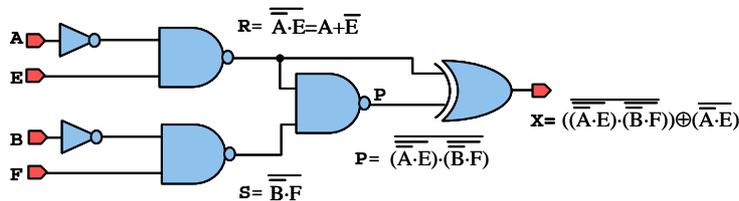
7.2.15. Del circuito siguiente obtener:

- 1) La ecuación de la salida en suma de productos
- 2) La tabla de verdad del circuito
- 3) Hacer el mapa de Karnaugh
- 4) Obtener a partir del mapa de Karnaugh la ecuación simplificada en suma de productos (SOP) y productos de sumas (POS)
- 5) Dibujar el esquema en puertas de las ecuaciones en suma de productos y producto de sumas simplificados

1) Ecuación de la salida en suma de productos

Identificamos la ecuación de cada señal en el esquema:

Y obtenemos la ecuación en SOP:



$$\begin{aligned}
 X &= ((\overline{\overline{A \cdot E}}) \cdot \overline{\overline{B \cdot F}}) \oplus (\overline{\overline{A \cdot E}}) \\
 &= (\overline{\overline{A \cdot E}} + \overline{\overline{B \cdot F}}) \oplus (A \cdot E) = \\
 &= (\overline{\overline{A \cdot E}} + \overline{\overline{B \cdot F}}) \cdot (A + \overline{E}) + (\overline{\overline{A \cdot E}} + \overline{\overline{B \cdot F}}) \cdot (A + \overline{E}) = \\
 &= (\overline{\overline{A \cdot E}} + \overline{\overline{B \cdot F}}) \cdot \overline{\overline{A \cdot E}} + (\overline{\overline{A \cdot E}} + \overline{\overline{B \cdot F}}) \cdot (A + \overline{E}) = \\
 &= \overline{\overline{A \cdot E}} \cdot \overline{\overline{A \cdot E}} + (\overline{\overline{A \cdot E}} + \overline{\overline{B \cdot F}}) \cdot (A + \overline{E}) = \\
 &= \overline{\overline{A \cdot E}} + (\overline{\overline{A \cdot E}} + \overline{\overline{B \cdot F}}) \cdot (B + \overline{F}) = \\
 &= \overline{\overline{A \cdot E}} + A \cdot B + A \cdot \overline{F} + B \cdot \overline{E} + \overline{E} \cdot \overline{F}
 \end{aligned}$$

$$X = \overline{\overline{A \cdot E}} + A \cdot B + A \cdot \overline{F} + B \cdot \overline{E} + \overline{E} \cdot \overline{F}$$

2) Tabla de verdad

Una de las maneras de elaborar la tabla de verdad es ir evaluando los valores a las salidas de cada una de las puertas.

A	B	E	F	A+E	B+F	R.S	POR
				R	S	P	X
0	0	0	0	1	1	0	1
0	0	0	1	1	0	1	0
0	0	1	0	0	1	1	1
0	0	1	1	0	0	1	1
0	1	0	0	1	1	0	1
0	1	0	1	1	1	0	1
0	1	1	0	0	1	1	1
0	1	1	1	0	1	1	1
1	0	0	0	1	1	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	1	0	1
1	0	1	1	1	0	1	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	0	1	1	0	1
1	1	1	1	1	1	0	1

3) Mapa de Karnaugh

SOP

		EF			
		00	01	11	10
AB	00	1	0	1	1
	01	1	1	1	1
	11	1	1	1	1
	10	1	0	0	1

$X = \bar{A} \cdot E + B + \bar{F}$

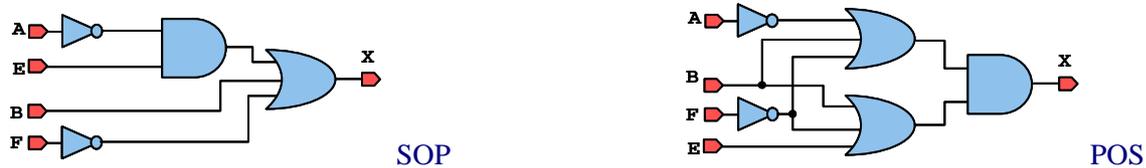
POS

		EF			
		00	01	11	10
AB	00	1	0	1	1
	01	1	1	1	1
	11	1	1	1	1
	10	1	0	0	1

$X = (B + E + \bar{F}) \cdot (\bar{A} + B + \bar{F})$

4) Ecuaciones en SOP y POS

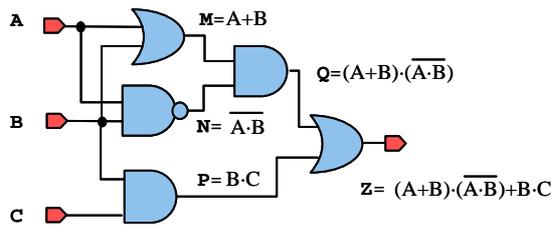
5) Esquema de la ecuaciones en SOP y POS simplificadas



7.2.16. Del circuito siguiente obtener:

- 1) La ecuación de la salida en suma de productos
- 2) La tabla de verdad del circuito
- 3) Hacer el mapa de Karnaugh
- 4) Obtener a partir del mapa de Karnaugh la ecuación simplificada en suma de productos (SOP) y productos de sumas (POS)
- 5) Dibujar el esquema en puertas de las ecuaciones en suma de productos y producto de sumas simplificadas

1) Ecuación de la salida en suma de productos



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

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

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

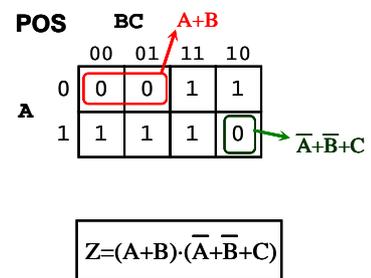
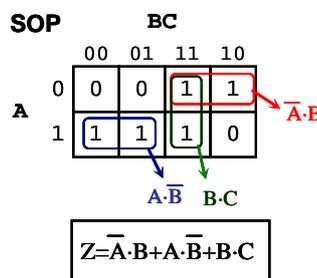
$$Z = A \cdot \overline{B} + \overline{A} \cdot B + B \cdot C$$

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

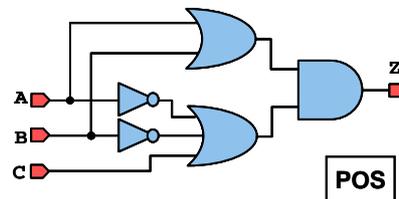
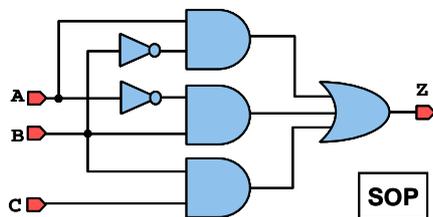
2) Tabla de verdad minimizadas

A+B			$\overline{A \cdot B}$	M·N	B·C	Q+P	
A	B	C	M	N	Q	P	Z
0	0	0	0	1	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	1	1	0	1
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	0	0	0	0
1	1	1	1	0	0	1	1

3 y 4) Mapa de Karnaugh y ecuaciones



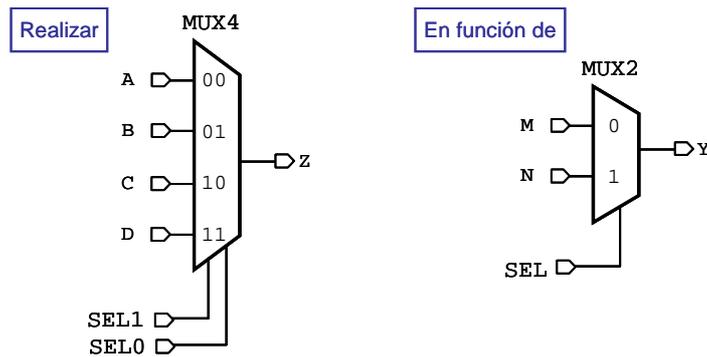
5) Esquema en puertas de la ecuaciones en SOP y POS simplificadas:



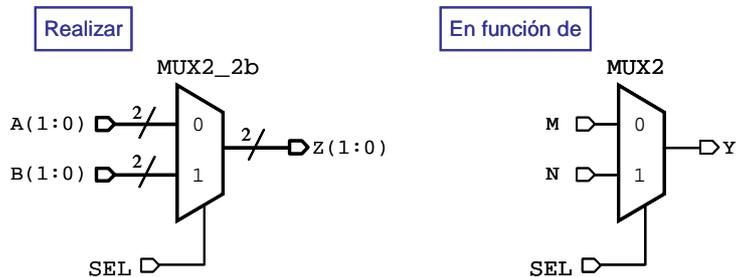
8. Bloques combinacionales

8.1. Enunciados

8.1.1. Realizar un multiplexor de 4 entradas de datos en función de multiplexores de 2 entradas de datos



8.1.2. Realizar un multiplexor de 2 entradas de datos de dos bits de ancho, en función de multiplexores de 2 entradas de datos de un bit de ancho.

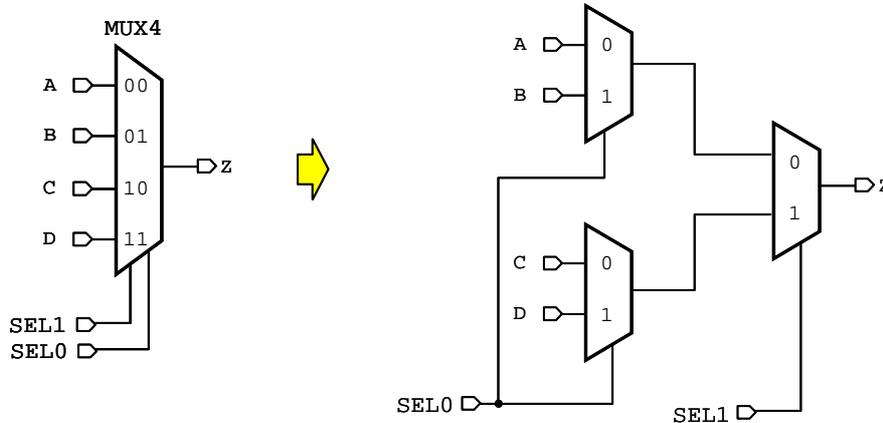


8.1.3. Realizar un decodificador de 4 entradas y 16 salidas a partir de decodificadores de 2 entradas y 4 salidas.

8.1.4. Explicar **brevemente** la implementación de una función lógica mediante un multiplexor y un decodificador. ¿Cuál de las dos opciones es preferible?

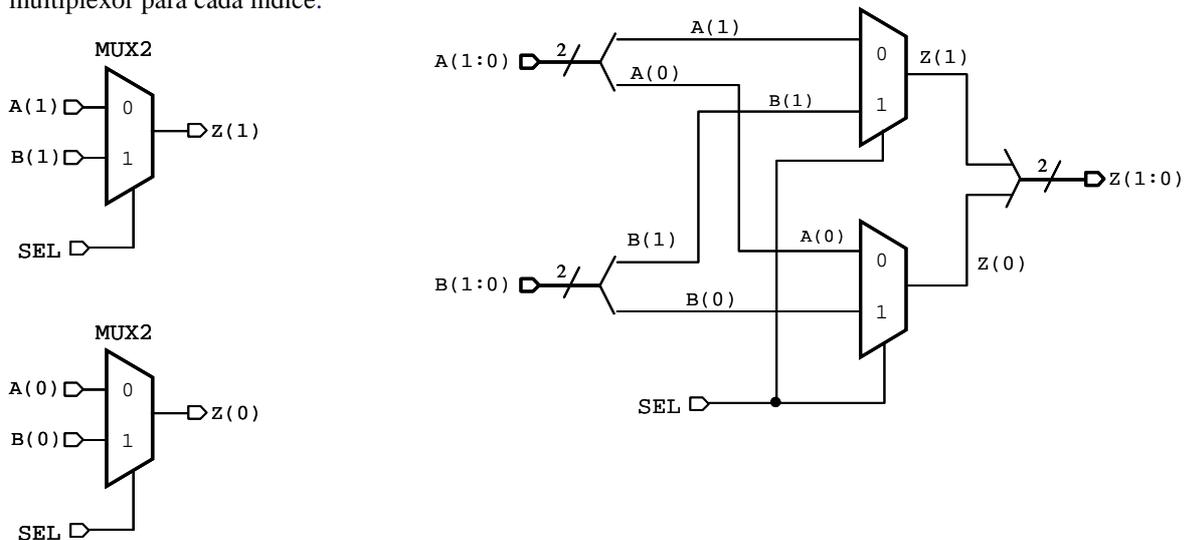
8.2. Soluciones

8.2.1. Realizar un multiplexor de 4 entradas de datos en función de multiplexores de 2 entradas de datos



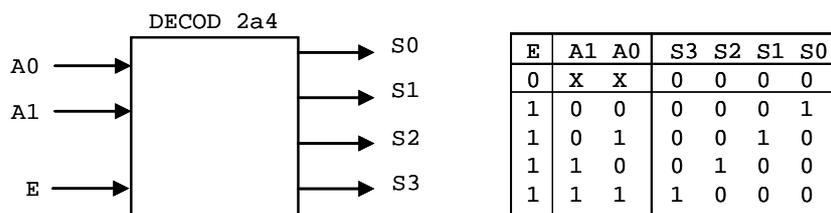
8.2.2. Realizar un multiplexor de 2 entradas de datos de dos bits de ancho, en función de multiplexores de 2 entradas de datos de un bit de ancho.

Simplemente hay que tomar un Dibujo todo junto queda: multiplexor para cada índice:

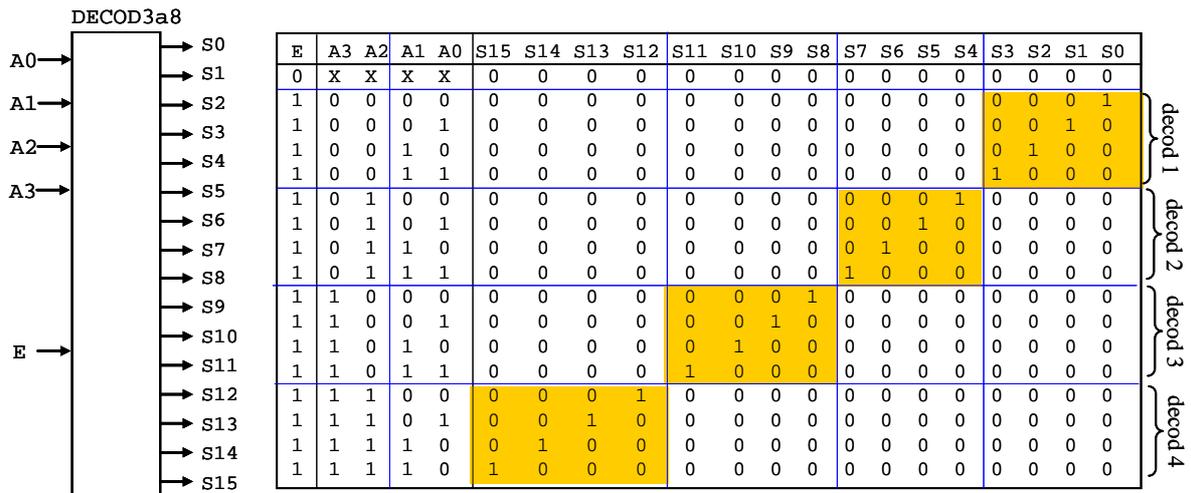


8.2.3. Realizar un decodificador de 4 entradas y 16 salidas a partir de decodificadores de 2 entradas y 4 salidas.

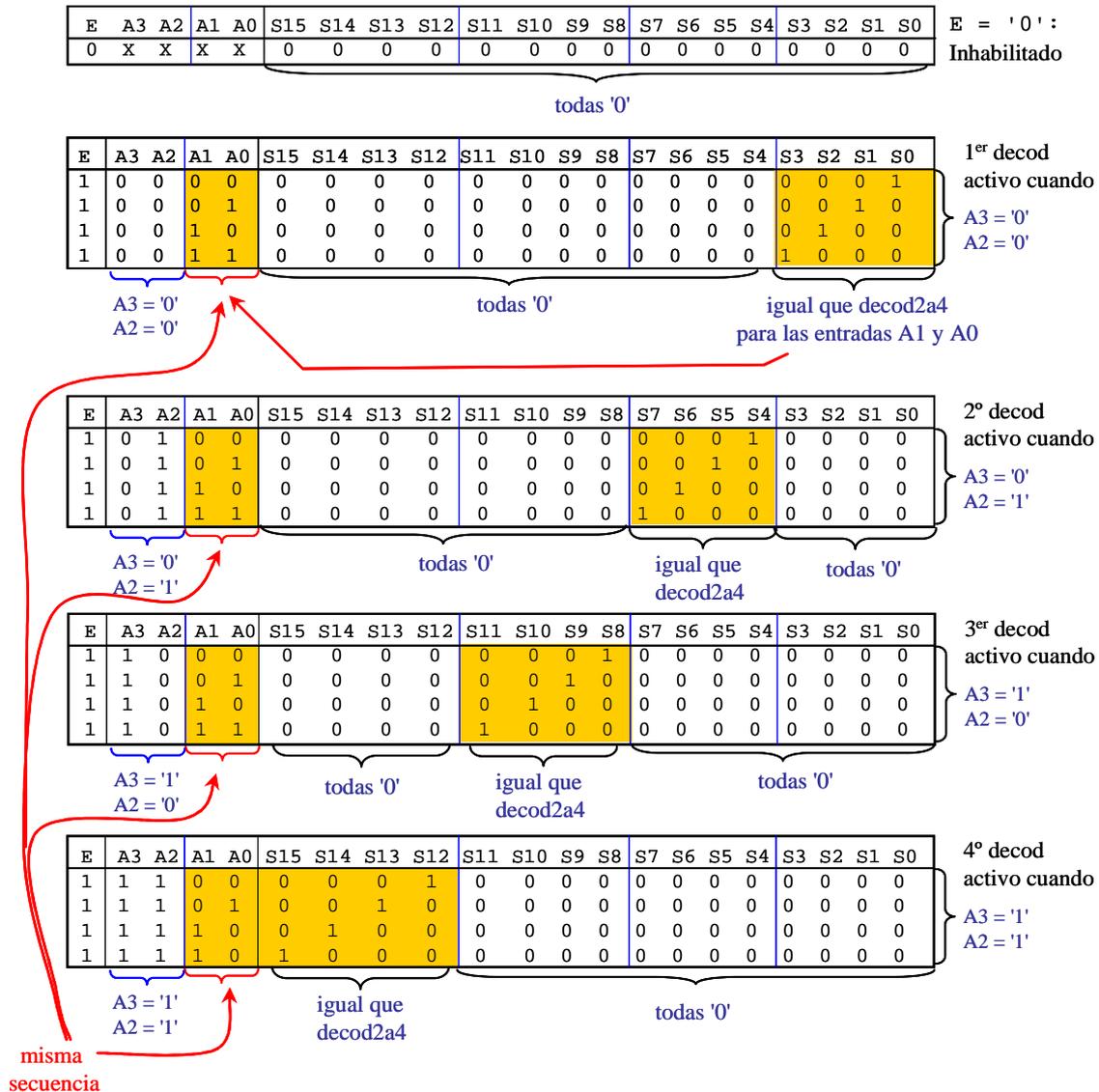
La tabla de verdad del decodificador de 2 a 4 es la siguiente:



Y la tabla de verdad del decodificador que queremos obteneres la siguiente:



En la tabla de verdad vemos que las señales de salida se han agrupado de cuatro en cuatro. En la siguiente figura se han separado estos grupos más el caso en que E = '0'.



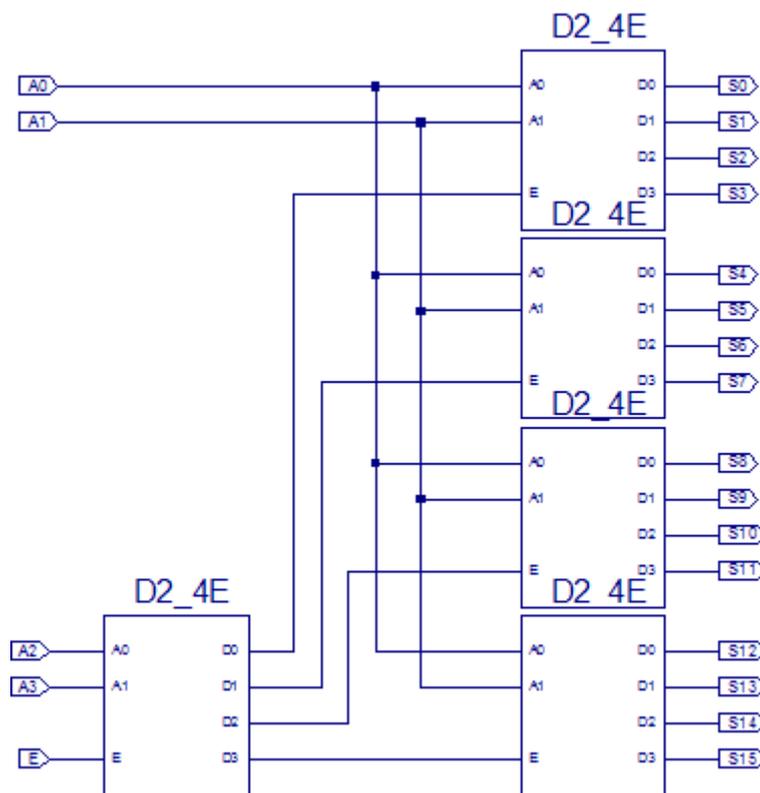
Podemos ver que para cada grupo se puede utilizar un decodificador de 2 a 4. Todos los decodificadores tendrán las mismas entradas A1 y A0. La única diferencia está en las salidas a las que se conecta cada decodificador, y la habilitación. Por ejemplo, el primer decodificador tendrá como salidas S0, S1, S2 y S3 y sólo estará habilitado cuando A3='0' y A2='0'; el segundo decodificador estará habilitado cuando A3='0' y A2='1'. Y de manera similar para el tercer y cuarto decodificador (como lo muestra la figura).

Así que usaremos cuatro decodificadores de 2 a 4 para las 16 salidas. Además necesitaremos crear la lógica que se encargue de habilitar cada decodificador. Si nos fijamos en las señales A3 y A2 de la tabla de verdad, y de cómo habilitan a cada decodificador, podemos extraer la tabla de verdad de la figura. En ella se ponen las señales de habilitación de cada decodificador (E4, E3, E2, E1) en función de la habilitación general (E) y las entradas A3 y A2. Podemos ver que la tabla de verdad se corresponde con la tabla de verdad de un decodificador de 2 a 4, así que para las señales de habilitación de los cuatro decodificadores usaremos otro decodificador de 2 a 4.

Señales de habilitación de los 4 decodificadores

E	A3	A2	E4	E3	E2	E1	
0	X	X	0	0	0	0	→ Ningún decodificador habilitado
1	0	0	0	0	0	1	→ 1 ^{er} decodificador habilitado
1	0	1	0	0	1	0	→ 2 ^o decodificador habilitado
1	1	0	0	1	0	0	→ 3 ^{er} decodificador habilitado
1	1	1	1	0	0	0	→ 4 ^o decodificador habilitado

Y conectamos los decodificadores:



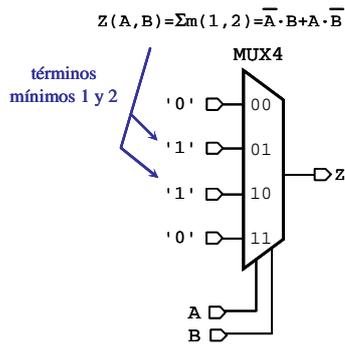
8.2.4. Explicar brevemente la implementación de una función lógica mediante un multiplexor y un decodificador. ¿Cuál de las dos opciones es preferible?

Con un multiplexor:

Si tenemos el mismo número de variables que señales de selección, simplemente hay que poner unos o ceros en las entradas del multiplexor según los términos mínimos de la función a realizar.

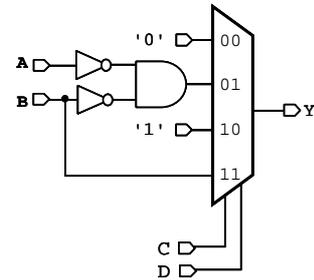
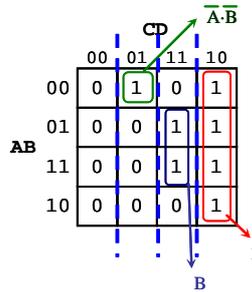
Si tenemos mayor número de variable, se puede obtener la función lógica o el mapa de Karnaugh, agrupando las variables según las variables que se hayan escogido como señales de selección.

Mismo número de variables que entradas de selección



Mayor número de variables que entradas de selección

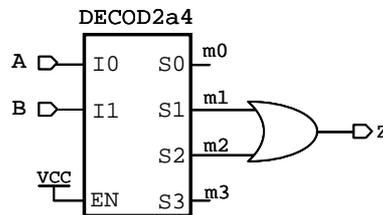
$Y(A, B, C, D) = \sum m(1, 2, 6, 7, 10, 14, 15)$



Con un decodificador:

Si tenemos el mismo número de variables que entradas del decodificador, se unen con una puerta OR todas las salidas correspondientes a los términos mínimos de la función.

$Z(A, B) = \sum m(1, 2) = \bar{A} \cdot B + A \cdot \bar{B}$



Ventajas:

Si se tiene el mismo número de variables que entradas de selección, la ventaja del multiplexor es que no hace falta añadir ninguna puerta externa. Sin embargo con el decodificador necesitamos una puerta OR.

Por otro lado, con más variables que entradas de selección el multiplexor reduce la complejidad del mapa de Karnaugh.

El decodificador también podría utilizarse cuando hubiese más variables que señales de entrada, aunque el método de diseño no es tan fácil como el empleado en el multiplexor

9. Elementos de memoria

9.1. Enunciados

9.1.1. Realizar un biestable J-K a partir de un biestable T. Explicar el procedimiento

9.2. Enunciados

9.2.1. Realizar un biestable J-K a partir de un biestable T. Explicar el procedimiento

1) Ponemos la tabla de verdad completa de lo que queremos: el biestable J-K

J	K	Q(t-1)	Q(t)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

2) Incluimos en la tabla una columna que nos diga cómo obtener el paso de Q(t-1) a Q(t) con lo que partimos: el biestable T

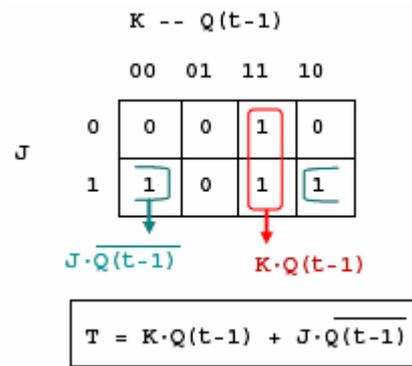
J	K	Q(t-1)	Q(t)	T
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	1

Como es un biestable T, cuando Q(t) sea igual a Q(t-1), T será '0'. Mientras que cuando sean distintos T tendrá que ser '1'

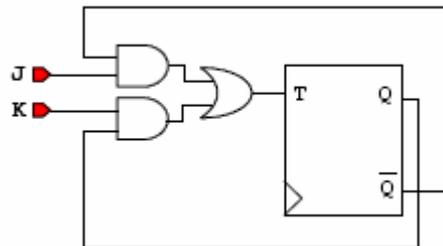
3) Ahora, con las entradas J, K y Q(t-1) hallamos la lógica necesaria para obtener T

J	K	Q(t-1)	T
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

4) Esto lo hacemos mediante su mapa de Karnaugh



5) Y dibujamos el circuito final

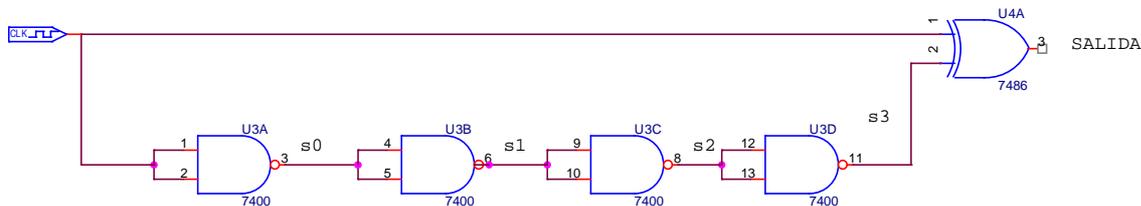


10. Análisis de circuitos 1

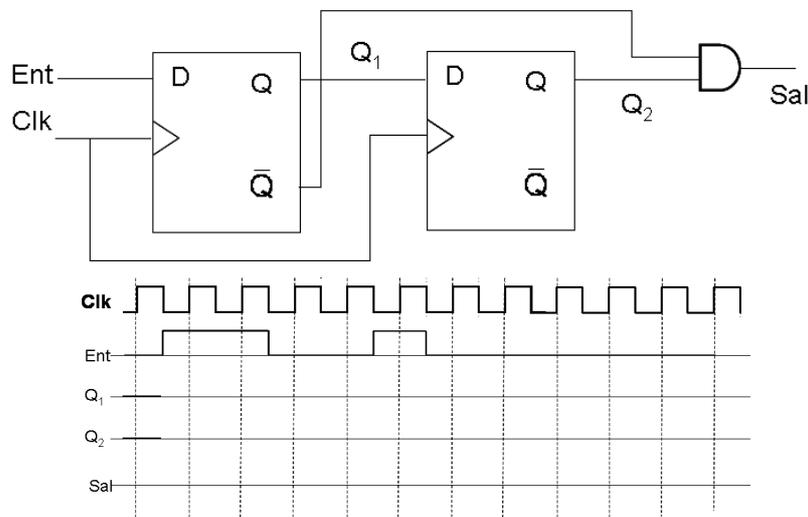
10.1. Enunciados

10.1.1. Dibujar la forma de onda a la salida del circuito de la figura durante 6 ciclos de la señal de reloj CLK, considerando

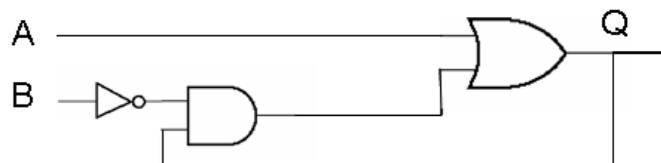
- El circuito ideal, es decir, los tiempos de propagación de las puertas son 0 ns.
- Tiempo de propagación de las puertas vale $\delta=10\text{ns}$. La frecuencia de la señal de reloj es de 1 MHz.



10.1.2. Completar el cronograma adjunto para el circuito de la figura. ¿Cuál es el cometido de este circuito lógico?



10.1.3. Completar la tabla de verdad que describe el siguiente circuito lógico. Indicar a qué biestable corresponde el circuito, y qué señales son A y B

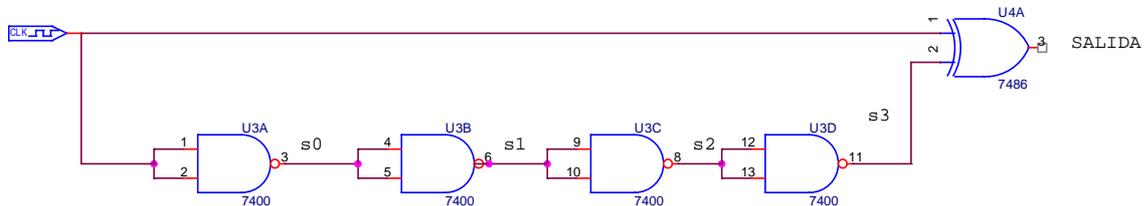


A	B	Q(t)	Operación

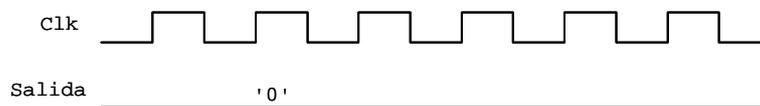
10.2. Soluciones

10.2.1. Dibujar la forma de onda a la salida del circuito de la figura durante 6 ciclos de la señal de reloj CLK, considerando

- a) El circuito ideal, es decir, los tiempos de propagación de las puertas son 0 ns.
- b) Tiempo de propagación de las puertas vale $\delta=10$ ns. La frecuencia de la señal de reloj es de 1 MHz.

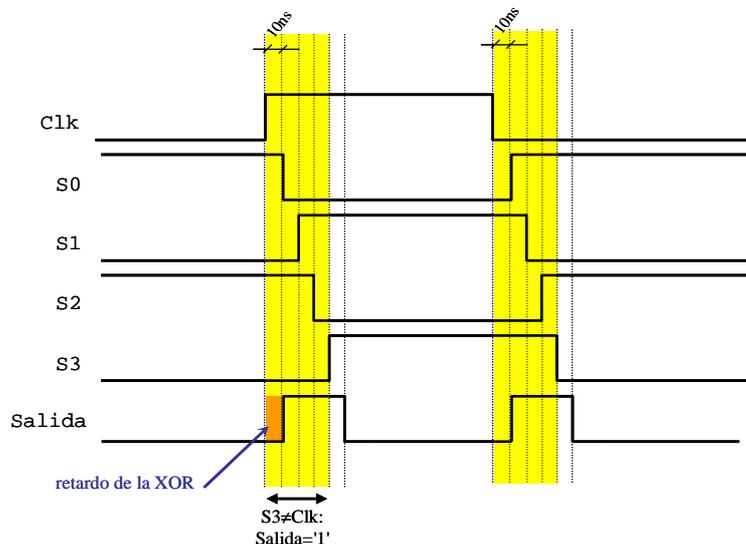


a) Si el circuito fuese ideal, el efecto de las puertas NAND, que funcionan como inversores, se anula, y por tanto en ambas entradas de la XOR tendríamos la misma señal CLK. Por tanto, la salida sería '0' durante todo el tiempo.

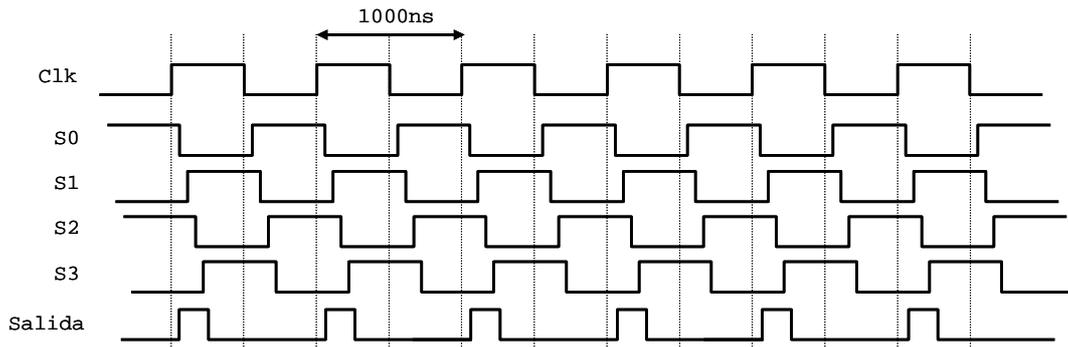


b) El circuito es real, y tanto las puertas NAND como la XOR introducen retardo. Así que transcurre un tiempo desde que hay una transición en la señal CLK hasta que se estabiliza SALIDA.

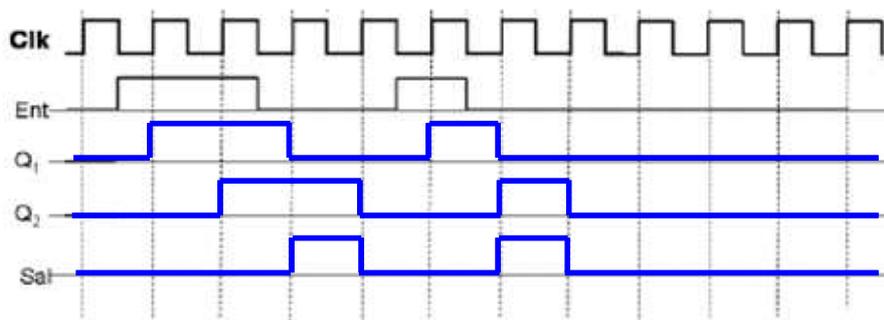
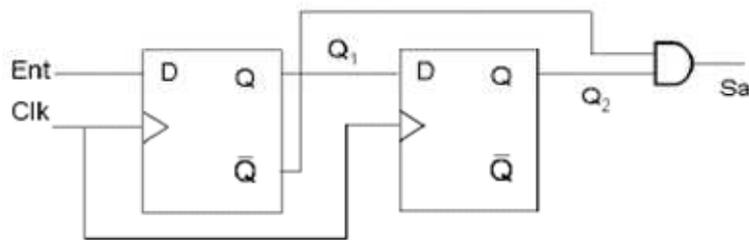
Cada puerta hace que haya un retardo y por lo tanto se van acumulando como muestra la siguiente figura (no está a escala). Al final la salida es 1 en los instantes cuando $CLK \neq S3$, aunque también ésta salida sufre el retardo debido a la XOR.



En 6 ciclos de reloj, el cronograma sería como el siguiente (no está a escala)



10.2.2. Completar el cronograma adjunto para el circuito de la figura. ¿Cuál es el cometido de este circuito lógico?

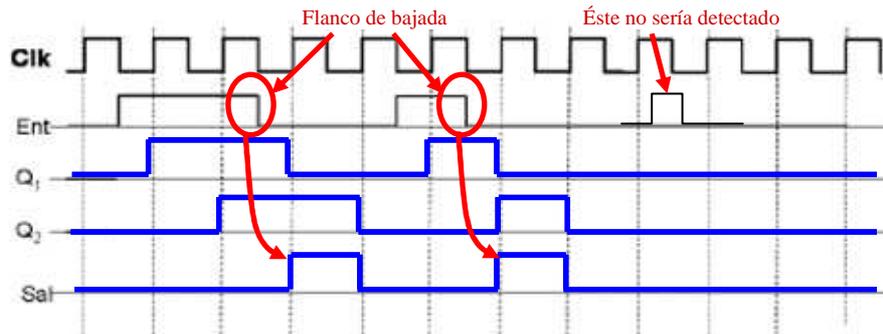


El circuito detecta flancos de bajada de la señal de entrada Ent, es decir, cuando en la señal Ent hay una transición de 1 a 0, por medio de la salida sal el circuito da un pulso que dura un ciclo de reloj.

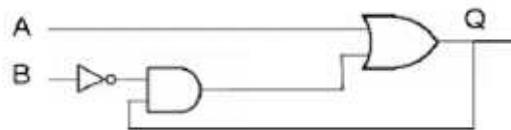
¿Por qué? Primero, el biestable Q1 copia el valor de la señal Ent después de cada flanco de subida del reloj. Luego, en el siguiente flanco de reloj el biestable Q2, copia el valor de Q1, y por tanto, Q2 es exactamente igual que la señal Q1, salvo que está retrasada un ciclo de reloj.

Así que para cada pulso (subida y bajada) de la señal Ent, sólo habrá un ciclo de reloj en el que $Q1 = 0$ y $Q2 = 1$, en dicho ciclo de reloj es cuando se activa la señal Sal

Para que las transiciones de Ent sean detectadas, el pulso debe durar como mínimo un ciclo de reloj.



10.2.3. Completar la tabla de verdad que describe el siguiente circuito lógico. Indicar a qué biestable corresponde el circuito, y qué señales son A y B



A	B	Q(t)	Operación
0	0	Q(t-1)	Mantiene Q
0	1	0	Puesta a 0 (Reset)
1	0	1	Puesta a 1 (Set)
1	1	1	

Es equivalente a un latch S-R asíncrono, en el que:

- A es equivalente a S (Set)
- B es equivalente a R (Reset)
- Y en el que nunca se debe dar la condición A=B='1' (condición prohibida).

11. Toldo automático

11.1. Enunciado

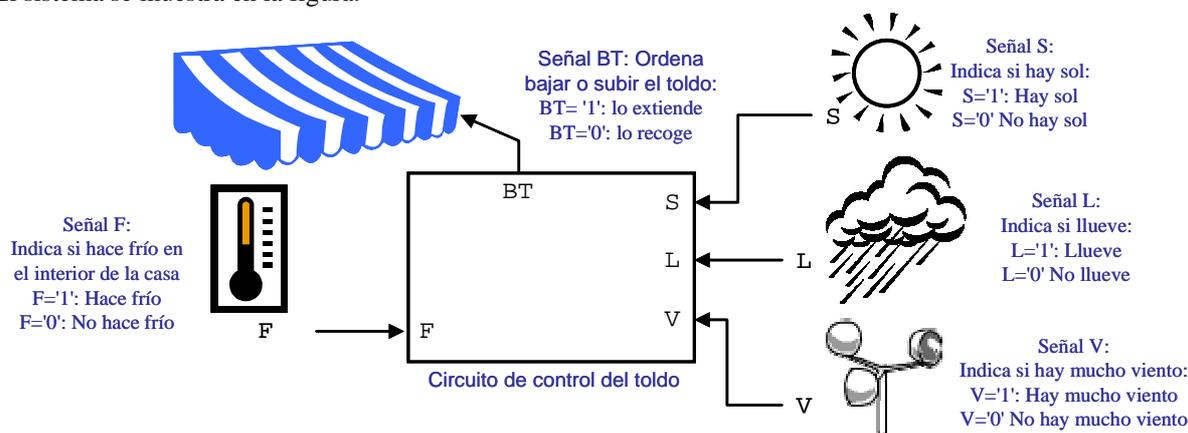
Se desea realizar un circuito de control para el toldo de una terraza de una vivienda. El toldo tiene la función tanto de dar sombra como de proteger del viento y de la lluvia. Así que es un toldo resistente al viento y a la lluvia, manteniendo la terraza seca en los días de lluvia.

Para el circuito de control tenemos las siguientes entradas:

- Señal S: Indica si hay sol
- Señal L: Indica si llueve
- Señal V: Indica si hay mucho viento
- Señal F: Indica si hace frío en el interior de la casa.

Según los valores de estas entradas se bajará o subirá el toldo. Esto se realizará mediante la señal de salida BT (Bajar Toldo). Si BT='1' indica que el toldo debe estar extendido (bajado) y si BT='0' indica que el toldo debe estar recogido (subido).

El sistema se muestra en la figura.



El circuito que acciona el toldo que debe funcionar según las siguientes características:

- Independientemente del resto de señales de entrada, siempre que llueva se debe de extender el toldo para evitar que se moje la terraza. No se considerará posible que simultáneamente llueva y haga sol.
- Si hace viento se debe extender el toldo para evitar que el viento moleste. Sin embargo, hay una excepción: aún cuando haya viento, si el día está soleado y hace frío en la casa, se recogerá el toldo para que el sol caliente la casa.
- Por último, si no hace viento ni llueve, sólo se bajará el toldo en los días de sol y cuando haga calor en el interior, para evitar que se caliente mucho la casa.

Se pide

- a) Realizar la tabla de verdad de la señal que controla el toldo (BT) a partir de las señales S, L, V y F.
- b) Obtener la expresión reducida en suma de productos, y producto de sumas
- c) Dibujar el esquema en puertas de estas expresiones
- d) Dibujar el diagrama de bloques para una variante del circuito en el que mediante un interruptor seleccionemos que el control del toldo sea manual o automático. De modo que:
 - Cuando el control sea automático, funcionará como se ha descrito hasta ahora.
 - Cuando el control sea manual, podremos hacer que el toldo suba o baje por medio de dos pulsadores. Un pulsador dará la orden de recoger el toldo y el otro de bajarlo. En este modo manual, el circuito deberá mantener la última orden pulsada después de soltar los pulsadores. Si se pulsan ambos pulsadores a la vez el comportamiento no será predecible.

11.2. Solución

a) Tabla de verdad

(Otro orden de las variables en la tabla de verdad es igualmente válido)

L	V	S	F	BT
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	1
1	1	0	1	1
1	1	1	0	X
1	1	1	1	X

No llueve ni hace viento: sólo se baja el toldo cuando haga calor (\bar{F}) y haya sol (S), entonces en el resto de casos se sube el toldo (BT=0)

→ No llueve ni hay viento, hace calor y hay sol: se baja el toldo (BT=1)

Hace viento, se baja el toldo (menos cuando hace frío y haya sol)

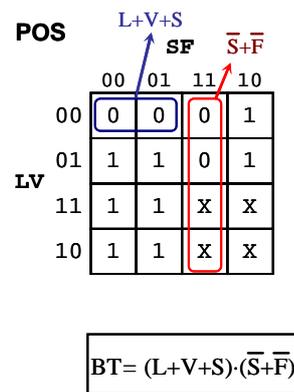
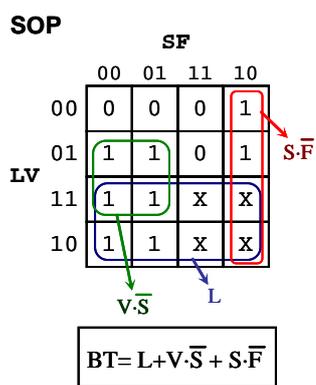
→ Hace frío y sol, se sube el toldo (BT=0)

→ Lluvia y sol: no se considera posible

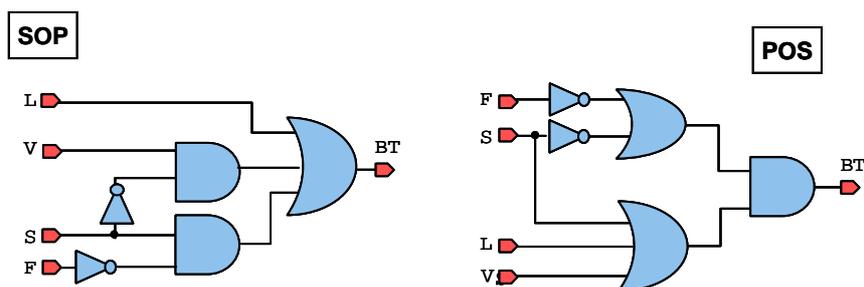
Llueve: se baja el toldo (BT=1)

→ Lluvia y sol: no se considera posible

c) Expresión reducida en suma de productos y productos de sumas

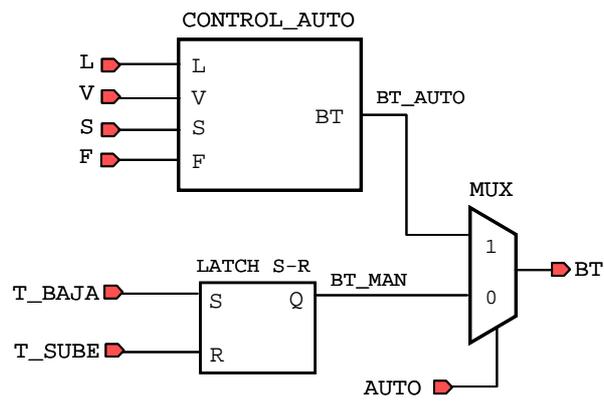


d) Esquema en puertas



e) Variante

La selección la realizamos con un multiplexor, que va a escoger la señal del circuito combinacional que acabamos de realizar en caso de que AUTO=1', en otro caso deberá tomar la selección que se haya tomado con los pulsadores. Debido a que son pulsadores, tenemos que guardar el valor del último pulsador. Lo podemos hacer con un latch S-R. Se podía haber hecho también con un biestable.

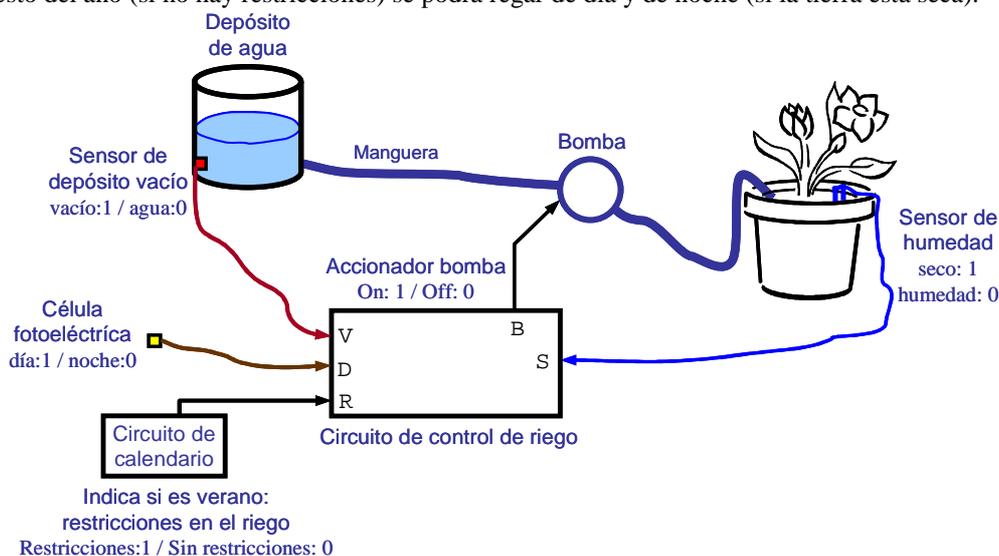


12. Riego automático

12.1. Enunciado

Se desea hacer un circuito de riego automático como el mostrado en la figura. El circuito deberá accionar la bomba en las siguientes condiciones

- El circuito accionará la bomba solamente cuando la tierra esté seca, pero antes debe comprobar las siguientes condiciones:
- Para evitar que la bomba se estropee por funcionar en vacío, nunca se accionará la bomba cuando el depósito de agua esté vacío.
- Si hay restricciones en el riego (época de verano), sólo se podrá regar de noche.
- En el resto del año (si no hay restricciones) se podrá regar de día y de noche (si la tierra está seca).



Para la implementación del circuito se dispone de las siguientes entradas:

- S: Señal que indica si la tierra está seca.
Tierra seca: S=1 ; Tierra húmeda: S=0
- R: Señal que indica si hay restricciones en el riego (es verano):
Hay restricciones: R=1 No hay restricciones: R=0
- D: Señal que indica si es de día o de noche:
Día: D=1 ; Noche: D=0
- V: Señal que indica si el depósito de agua está vacío:
Vacío: V=1 ; Hay agua: V=0

Y la salida B, que accionará la bomba para regar: *Bomba funcionando: B=1 ; Bomba apagada B=0*

Con esta información se debe:

- Elaborar la tabla de verdad del circuito
- Obtener la ecuación en la primera forma normal
- Hacer el mapa de Karnaugh
- Obtener la ecuación simplificada en suma de productos y productos de suma
- Representar ambas ecuaciones simplificadas (SOP y POS) en puertas lógicas

12.2. Solución

a y b) Tabla de verdad y 1FN

V	S	D	R	B
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Tanque con agua,
tierra seca

Tanque con agua,
tierra no está seca:
no se riega

De noche: se riega
(independientemente de la época)

De día en invierno: se riega

De día en verano: no se riega

Tanque vacío: no se riega

1FN: m4 + m5 + m6:

$$B = \bar{V} \cdot S \cdot \bar{D} \cdot \bar{R} + \bar{V} \cdot S \cdot \bar{D} \cdot R + \bar{V} \cdot S \cdot D \cdot \bar{R}$$

c) Mapa de Karnaugh

SOP

	DR			
	00	01	11	10
00	0	0	0	0
01	1	1	0	1
11	0	0	0	0
10	0	0	0	0

vs

$\bar{V} \cdot S \cdot \bar{R}$

$\bar{V} \cdot S \cdot \bar{D}$

POS

	DR			
	00	01	11	10
00	0	0	0	0
01	1	1	0	1
11	0	0	0	0
10	0	0	0	0

vs

\bar{V}

$\bar{D} + \bar{R}$

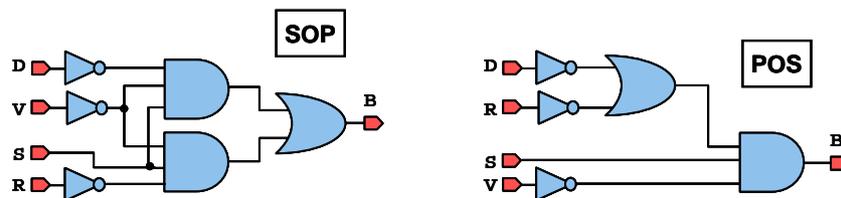
S

d) Ecuaciones simplificadas:

$$B = \bar{V} \cdot S \cdot \bar{D} + \bar{V} \cdot S \cdot \bar{R}$$

$$B = \bar{V} \cdot S \cdot (\bar{D} + \bar{R})$$

e) Esquema en puertas de la ecuaciones en SOP y POS simplificadas:



13. Detector de números primos en BCD-XS-3

13.1. Enunciado

Se quiere realizar un circuito que reciba un número BCD-XS3 (4 bits) y devuelva '1' si el número recibido es primo, y devuelva '0' si no lo es.

Se considerará el número 1 como número primo. El cero no es un número primo.

En ningún caso el circuito recibirá números que no estén codificados en BCD-XS3.

Se pide:

- a) Realizar la tabla de verdad de la señal de salida.
- b) Obtener la expresión reducida en suma de productos, y producto de sumas
- c) Dibujar el esquema en puertas de estas expresiones

13.2. Solución

Como los BCD-XS3 van de 0 a 9, los números primos son: 1,2,3,5,7 (ya que se especifica que el uno se considera como primo).

Entonces la tabla de verdad para la señal de salida P es

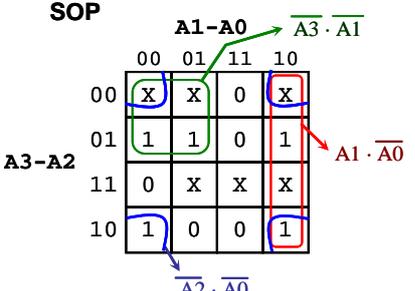
A3	A2	A1	A0	Num	P
0	0	0	0	-	X
0	0	0	1	-	X
0	0	1	0	-	X
0	0	1	1	0	0
0	1	0	0	1	1
0	1	0	1	2	1
0	1	1	0	3	1
0	1	1	1	4	0
1	0	0	0	5	1
1	0	0	1	6	0
1	0	1	0	7	1
1	0	1	1	8	0
1	1	0	0	9	0
1	1	0	1	-	X
1	1	1	0	-	X
1	1	1	1	-	X

← número que representa
} No son posibles, ya que no son un números BCD-XS3

} No son posibles, ya que no son un números BCD-XS3

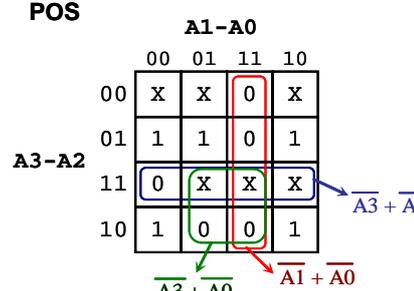
Usamos los mapas de Karnaugh para obtener las expresiones:

SOP



$P = A1 \cdot \overline{A0} + \overline{A3} \cdot \overline{A1} + \overline{A2} \cdot \overline{A0}$

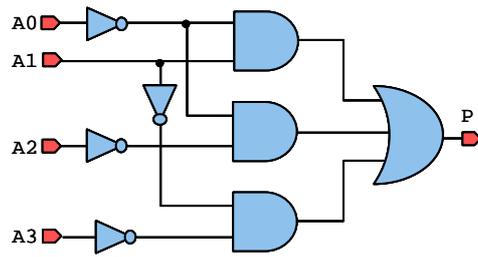
POS



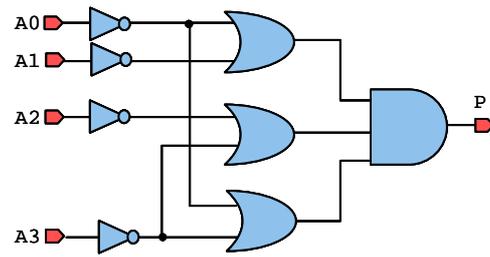
$P = (\overline{A1} + \overline{A0}) \cdot (\overline{A3} + \overline{A2}) \cdot (\overline{A3} + \overline{A0})$

La representación en puertas queda:

SOP



POS



14. Alarma de incendios

14.1. Enunciado

Se quiere realizar un circuito para activar la alarma de incendios (**A**) para la evacuación de un edificio. Para ello se tiene un sensor de gases (**G**), un sensor de humos (**H**), y dos señales procedentes de un termómetro que indican si la temperatura es mayor de 45°C (**T45**) y si la temperatura es mayor de 60°C (**T60**).

Debido a que a veces los sensores detectan humos y gases que no siempre proceden de incendios (por ejemplo de los cigarrillos o las cocinas), para evitar falsas alarmas, la señal **A** se activará cuando se cumplan las siguientes condiciones:

- Si la temperatura es mayor de 60°C siempre se activará la alarma
- Si la temperatura está entre 45°C y 60°C se activará la alarma sólo si han detectado gases o humos (o ambos).
- Si la temperatura es menor de 45°C se activará la alarma sólo si se detectan gases y humos

Resumiendo, las 4 señales binarias de entrada y la salida:

- **G**: vale '1' si se detecta **GAS** resultante de la combustión.
- **H**: vale '1' si se detecta **HUMO**.
- **T45**: vale '1' si la temperatura es superior a 45°C
- **T60**: vale '1' si la temperatura es superior a 60°C

La señal de salida **A** (alarma) se activará a nivel alto

Se pide

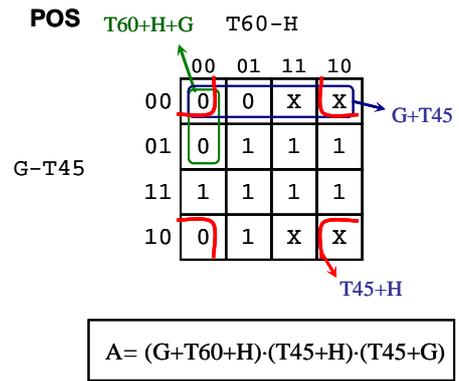
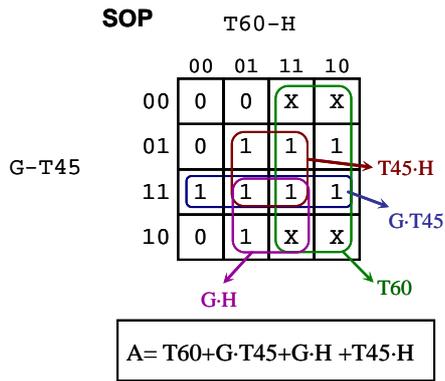
- a) Realizar la tabla de verdad de la señal de alarma (**A**) a partir de las señales de entrada (**G, H, T45, T60**). Explicarla brevemente.
- b) Obtener la expresión reducida en suma de productos y producto de sumas
- c) Dibujar el esquema en puertas de estas expresiones

14.2. Solución

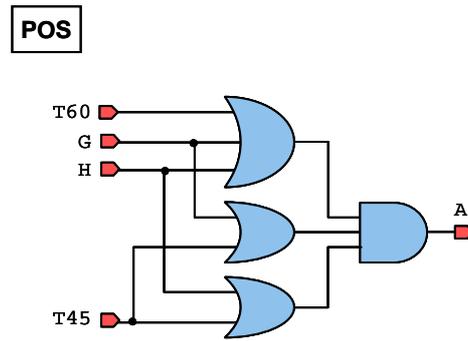
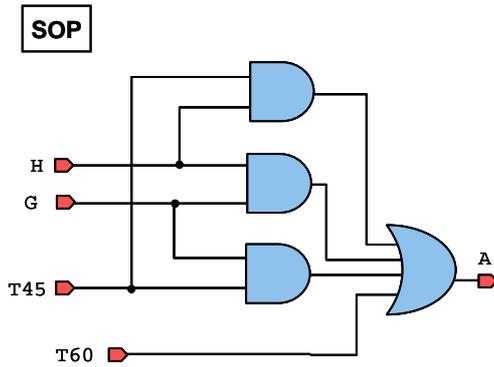
a) Tabla de verdad

G	T45	T60	H	A	
0	0	0	0	0	→ Condiciones normales: NO Alarma
0	0	0	1	0	→ Sólo humo: NO Alarma
0	0	1	0	X	} → La temperatura no puede ser mayor que 60° y menor que 45°
0	0	1	1	X	
0	1	0	0	0	→ temperatura mayor que 45°, sin humo ni gas: NO Alarma
0	1	0	1	1	→ Hay humo y temperatura mayor que 45° : Alarma
0	1	1	0	1	} → Más de 60°: alarma
0	1	1	1	1	
1	0	0	0	0	→ Sólo gas: NO Alarma
1	0	0	1	1	→ Gas y Humo aunque temperatura sea menor que 45°: alarma
1	0	1	0	X	} → La temperatura no puede ser mayor que 60° y menor que 45°
1	0	1	1	X	
1	1	0	0	1	} → Hay gas y temperatura mayor que 45° : Alarma
1	1	0	1	1	
1	1	1	0	1	} → Más de 60°: alarma
1	1	1	1	1	

b) Expresión reducida en suma de productos (SOP) y producto de sumas (POS)



c) Esquema en puertas de estas expresiones



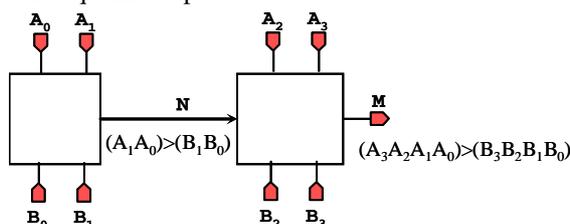
15. Comparador

15.1. Enunciado

Realizar un circuito que recibe dos números binarios puros (sin signo): A[3,0], B[3,0] e indica por la salida S si A es mayor que B

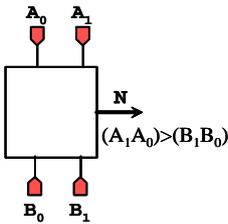
Especificaciones y notas de ayuda al diseño:

- La salida S='1' cuando A>B, en otro caso S='0'.
- El bit más significativo es el bit 3.
- Debe obtenerse la función mínima mediante mapas de Karnaugh, pero debido a que el circuito tiene 8 entradas y resulta difícil de hacer el mapa de Karnaugh con tantas variables, el circuito se implementará en dos módulos. La implementación se muestra en la figura del final.
- Primero se deberá realizar un circuito que compare los dos bits menos significativos de A y B, y devuelva '1' si el número (A₁A₀) es mayor que el número (B₁B₀). Esta salida (N) será la entrada del segundo comparador.
- Después de haber hecho el primer comparador (de 4 entradas) se realizará el segundo comparador (de 5 entradas). Este comparador recibe los dos bits más significativos de A y B, y la señal N que indica si (A₁A₀)>(B₁B₀).
- El segundo comparador es muy similar al primero, únicamente cuando (A₃A₂)= (B₃B₂) será necesario utilizar la señal N.
- Para el mapa de Karnaugh del segundo comparador se recomienda tomar la señal N como la 5ª variable.
- Representar el circuito final en esquema de puertas

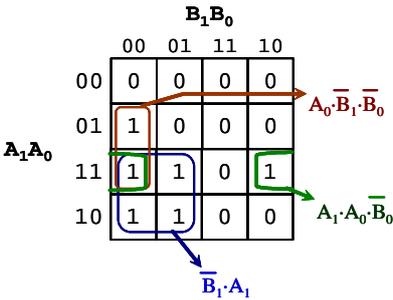


15.2. Solución

Primero representamos la tabla de verdad del primer comparador y dibujamos su mapa de Karnaugh, obteniendo a partir de él la ecuación lógica en suma de productos (pues sale más pequeña que en producto de sumas).

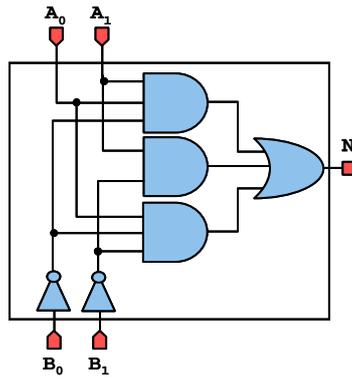


A1	A0	B1	B0	N
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0



$$N = A_1 \cdot A_0 \cdot \bar{B}_0 + A_1 \cdot \bar{B}_1 + A_0 \cdot \bar{B}_1 \cdot \bar{B}_0$$

Una vez que tenemos la ecuación lógica del primer comparador, ya podemos representarlo en puertas lógicas:



El segundo comparador, tiene una entrada más (N), cuando N es cero la tabla de verdad es igual que la del primer comparador. Mientras que cuando N=1, M valdrá 1 cuando $A_3A_2=B_3B_2$

N=0 igual que al anterior

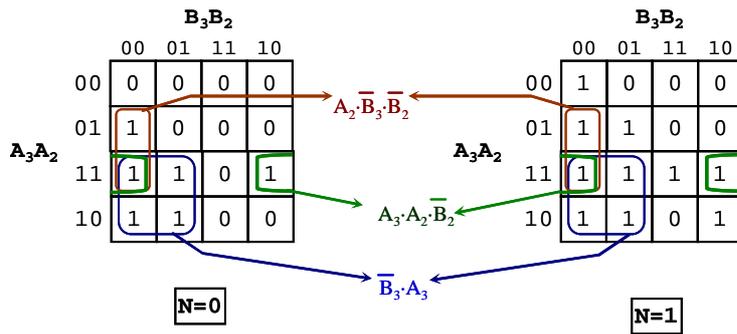
N	A3	A2	B3	B2	M
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	0	1	1
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0

N=1 cuando $A_3=B_3$ y $A_2=B_2 \rightarrow M=1$

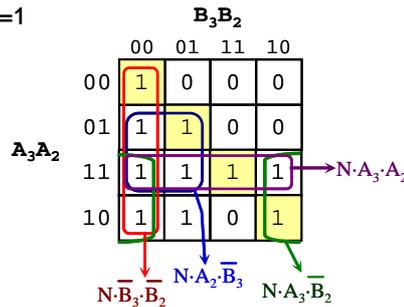
N	A3	A2	B3	B2	M
1	0	0	0	0	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	1
1	0	1	1	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

Como la tabla de verdad es igual cuando N=0, también lo será el mapa de Karnaugh para N=0, y por eso tomamos la variable N como la variable que se saca fuera, quedando:

Cubos que son iguales en N=0 y N=1



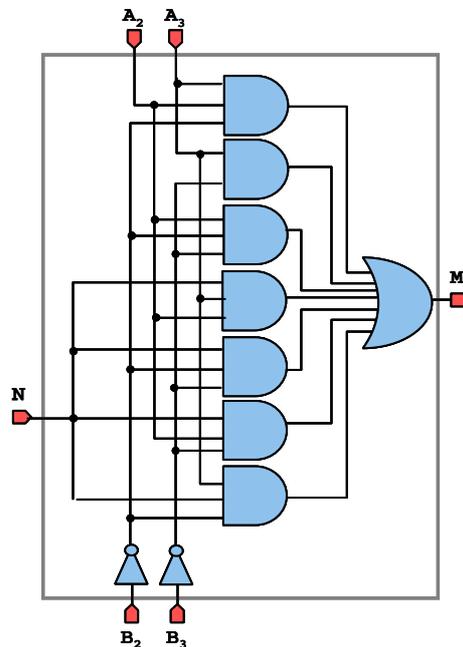
Cubos que sólo están en N=1



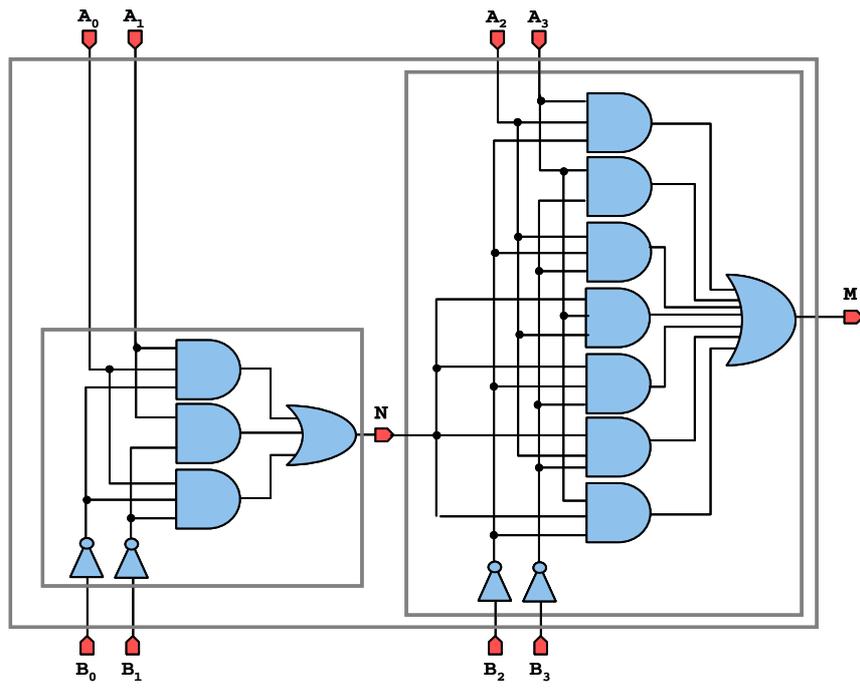
$$M = A_3 \cdot A_2 \cdot \bar{B}_2 + A_3 \cdot \bar{B}_3 + A_2 \cdot \bar{B}_3 \cdot \bar{B}_2 + N \cdot \bar{B}_3 \cdot \bar{B}_2 + N \cdot A_2 \cdot \bar{B}_3 + N \cdot A_3 \cdot \bar{B}_2 + N \cdot A_3 \cdot A_2$$

Para no complicar las figuras se han puesto primero los grupos (cubos) que son iguales para N=0 y N=1, y debajo se han puesto aquellos los necesarios para cubrir los que no están en el mapa de N=0 (los unos de la diagonal), todos estos grupos incluirán la variable N.

La suma de todos los grupos nos da la ecuación de la salida, que su esquema en puertas queda representado así:



Juntando ambos módulos, el circuito final queda:



16. Comparador BCD-XS3

16.1. Enunciado

Repetir el ejercicio anterior suponiendo que los números están codificados en BCD-XS3

16.2. Solución

Primero recordamos que 4 bits de un número BCD-XS3 representa una cifra decimal entre 0 y 9. Por tanto, de las 16 combinaciones posibles para cada uno de los números que entra en nuestro comparador (A y B) sólo serán posibles 10.

En la siguiente tabla se muestran los valores decimales que representan los números BCD-XS3

número en BCD-XS3				XS3 (número decimal que representa)	
A3	A2	A1	A0		
0	0	0	0	-	} No son números BCD-XS3
0	0	0	1	-	
0	0	1	0	-	
0	0	1	1	0	Si $(A_3=0 \text{ y } A_2=0) \Rightarrow A_1=1 \text{ y } A_0=1$
0	1	0	0	1	} Si $(A_3=0 \text{ y } A_2=1) \Rightarrow$ cuanto mayor sea A_1A_0 mayor será el número
0	1	0	1	2	
0	1	1	0	3	
0	1	1	1	4	
1	0	0	0	5	} Si $(A_3=1 \text{ y } A_2=0) \Rightarrow$ cuanto mayor sea A_1A_0 mayor será el número
1	0	0	1	6	
1	0	1	0	7	
1	0	1	1	8	
1	1	0	0	9	Si $(A_3=1 \text{ y } A_2=1) \Rightarrow A_1=0 \text{ y } A_0=0$
1	1	0	1	-	} No son números BCD-XS3
1	1	1	0	-	
1	1	1	1	-	

Viendo la tabla podemos deducir que:

- Hay 6 números que no son posibles
- Si los dos bits más significativos son 00 ($A_3=0, A_2=0$) los dos bits menos significativos serán 11 ($A_1=1, A_0=1$), que se corresponde con el 0 en decimal.
- Si los dos bits más significativos son 11 ($A_3=1, A_2=1$) los dos bits menos significativos serán 00 ($A_1=0, A_0=0$), que se corresponde con el 9 en decimal.
- Por tanto en estos dos casos no hace falta saber los dos bits menos significativos (bit 1 y bit 0). Es decir, el segundo comparador no necesita la señal N.
- Si los dos bits más significativos son 01 ($A_3=0, A_2=1$), entonces recurriremos a los dos bits menos significativos para saber qué número es. En este caso sí que necesitamos saber cuanto valen. Y el segundo comparador necesita la señal N. Lo mismo ocurre cuando los dos bits más significativos son 10 ($A_3=1, A_2=0$)
- El primer comparador será igual que el del circuito anterior, y determinará quién es mayor cuando ($A_3=0, B_3=0, A_2=1, B_2=1$) ó cuando ($A_3=1, B_3=1, A_2=0, B_2=0$)

La tabla de verdad queda

N=0					
N	A3	A2	B3	B2	M
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	0	1	1
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0

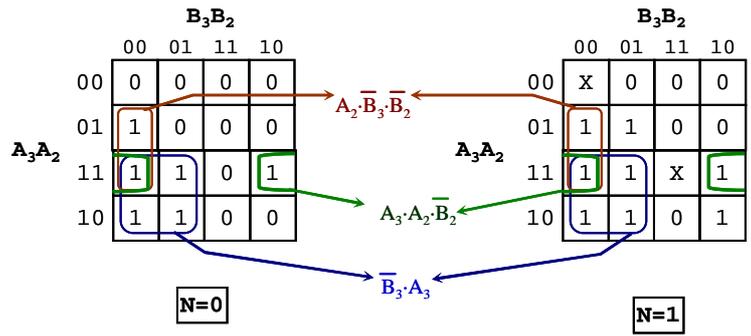
N=1					
N	A3	A2	B3	B2	M
1	0	0	0	0	X
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	1
1	0	1	1	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	X

→ imposible: Si (A₃=0 y A₂=0) => A₁=1 y A₀=1 ;
y Si (B₃=0 y B₂=0) => B₁=1 y B₀=1
entonces

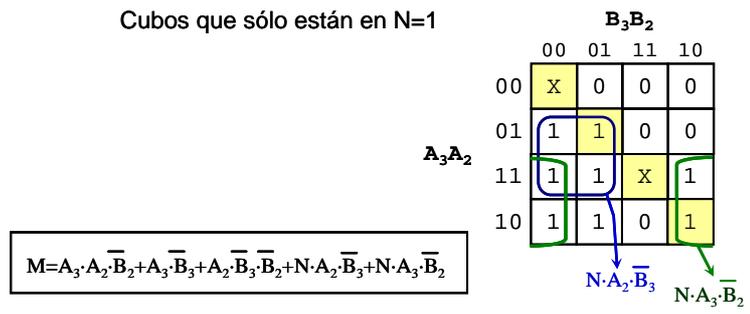
→ imposible: Si (A₃=1 y A₂=1) => A₁=0 y A₀=0 ;
y Si (B₃=1 y B₂=1) => B₁=0 y B₀=0
entonces N = 0

Esta tabla de verdad es igual a la del problema anterior salvo los dos casos señalados que no pueden darse. Estos dos casos nos permitirán simplificar los mapas de Karnaugh, que serán muy parecidos a los del problema anterior.

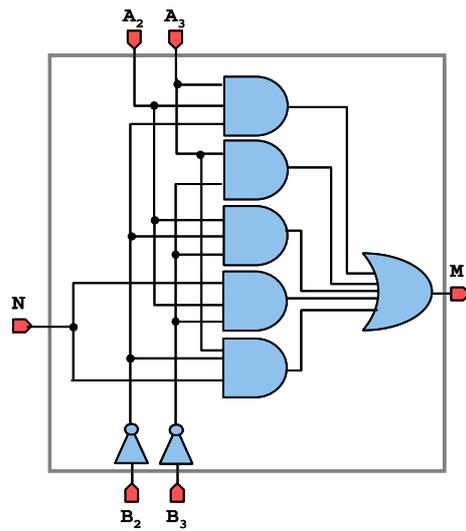
Cubos que son iguales en N=0 y N=1



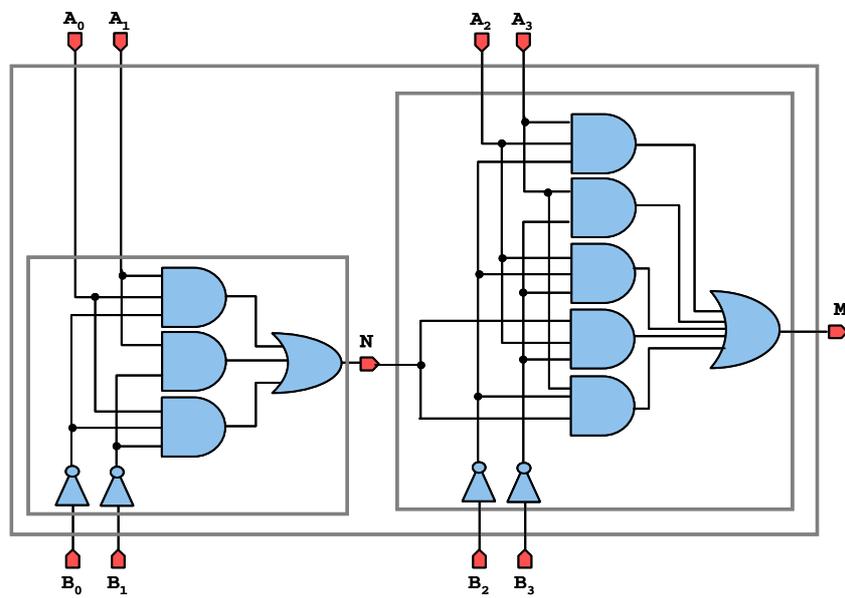
Cubos que sólo están en N=1



Así pues, hemos logrado quitar dos sumandos a la ecuación. El esquema en puertas queda representado:



Juntando ambos módulos, el circuito final queda:



17. Convertidor de binario a BCD

17.1. Enunciado

Realizar un circuito que a partir de un número binario puro de 4 bits obtenga el número BCD equivalente. El circuito **se debe realizar mediante bloques combinacionales** y no mediante mapas de Karnaugh o sólo puertas lógicas. El resultado será un número BCD que representa dos cifras decimales (8 bits en total).

Para realizar el circuito se pueden usar los siguientes bloques combinacionales: decodificadores, codificadores, comparadores, multiplexores, demultiplexores y sumadores de cualquier número de bits. Así como puertas lógicas. Cualquier otro bloque se deberá realizar en función de los anteriores o en puertas lógicas. Para cada bloque utilizado se debe especificar claramente qué bloque es, las señales de entrada y salida, y el ancho de bus.

17.2. Solución

Primero estudiamos la relación entre los dos sistemas de numeración, el binario y el BCD, para números binarios de 4 bits (rango de 0 a 15). Esta relación se muestra a continuación:

decimal	binario	BCD	
		decenas	unidades
0	0000	0000	0000
1	0001	0000	0001
2	0010	0000	0010
3	0011	0000	0011
4	0100	0000	0100
5	0101	0000	0101
6	0110	0000	0110
7	0111	0000	0111
8	1000	0000	1000
9	1001	0000	1001
10	1010	0001	0000
11	1011	0001	0001
12	1100	0001	0010
13	1101	0001	0011
14	1110	0001	0100
15	1111	0001	0101

Número binario menor o igual a 9:
- unidades igual que en binario,
- decenas igual a 0000

Número binario mayor que 9:
- unidades: número binario
 menos 10
- decenas igual a 0001

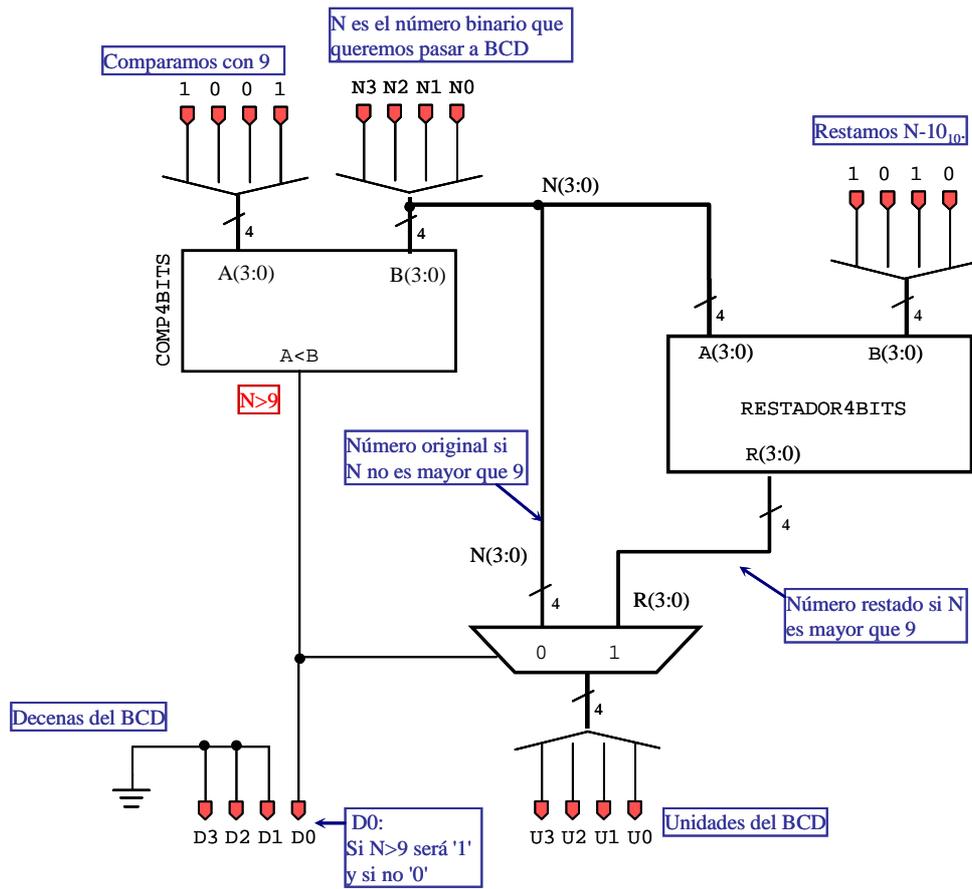
Por ejemplo, si tenemos el número 14,
le restamos 10, y obtenemos las unidades

1110	14
-1010	-10
0100	4

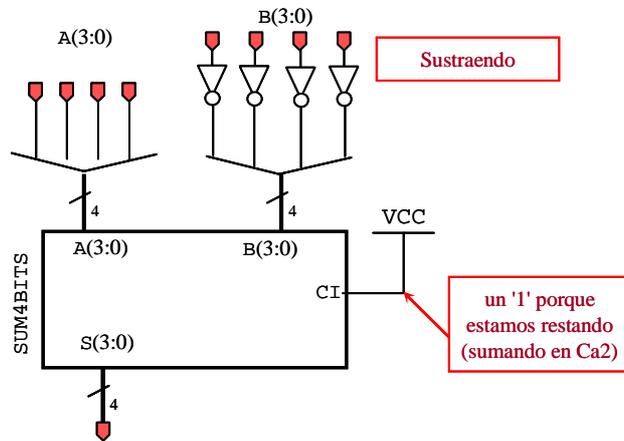
Nos fijamos que hay dos grupos, los números menores o iguales a 9, y los mayores que 9.

Por lo tanto, si el número binario es menor o igual a nueve, dejamos la unidades igual que el número binario y las decenas a cero.

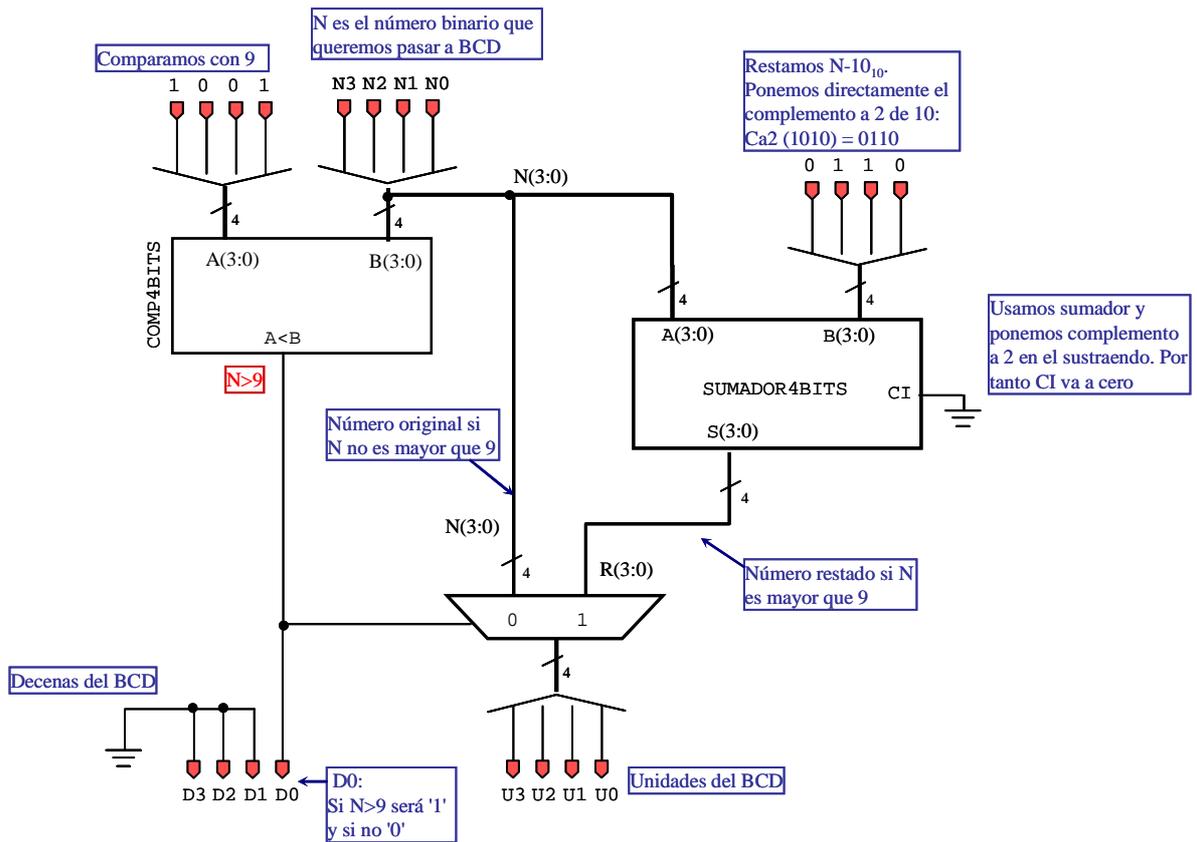
Y si el número binario es mayor a nueve, dejamos la unidades igual que el número binario y las decenas a cero.



Sólo nos quedaría especificar cómo se hace el restador de 4 bits:



Por último, como curiosidad, fíjate que como el sustraendo es un número fijo (1010_2), el circuito se podría optimizar un poco, usando directamente un sumador habiendo hecho previamente el complemento a 2 de 1010_2 , que es 0110 . Por tanto es como sumar 6 y desprestigiar el acarreo. Lo que es otra manera de hacerlo, ya que por ejemplo, $13+6=19$ que en binario es 10011_2 . Tomando los 4 primeros bits nos da el 0011 que es el tres de las unidades del 13. El circuito quedará:



18. Café electrónico

18.1. Enunciado

Para disminuir la ingesta de cafeína durante el periodo de exámenes, un grupo de alumnos de 1º de Ingeniería de Telecomunicación de la URJC deciden realizar el "café electrónico".

Lo que quieren hacer es detectar si el alumno se queda dormido mientras estudia por la noche, en tal caso y según la hora que sea, hacer sonar una alarma para despertarle.

Para detectar si está dormido ponen un circuito detector de movimiento en su muñeca, de modo que si la muñeca está quieta durante más de 10 minutos, se activará la señal **Q10** (*Quieto 10 minutos*). Esto será una señal inequívoca de que se ha quedado dormido, ya que durante 10 minutos el alumno no ha movido la mano para escribir, ni para de pasar de página, ni para rascarse.

Sin embargo, la señal **Q10** no siempre se usará para despertar al alumno. Si estamos entre las 4am y las 6am, no le despertaremos para que descanse un poco.

Por otro lado, independientemente de la hora, si el usuario lleva media hora durmiendo, siempre se le despertará para que decida si quiere seguir estudiando o realmente quiere irse a dormir en la cama y no en la mesa. La señal que indica que lleva media hora quieto se llamará **Q30** (*Quieto 30 minutos*).

Resumiendo, las señales que entran a nuestro sistema son:

- **Q10**: vale '1' si el alumno lleva 10 minutos o más quieto, si no Q10='0';
- **Q30**: vale '1' si el alumno lleva 30 minutos o más quieto, si no Q30='0';
- **M4**: vale '1' si son más de las 4am, si no M4='0';
- **M6**: vale '1' si son más de las 6am, si no M6='0';

La señal de salida **A** (alarma) se activará a nivel alto.

Se pide

- a) Realizar la tabla de verdad de la señal que controla la alarma (**A**) a partir de las señales de entrada **Q30**, **Q10**, **M4** y **M6**.
- b) Obtener la expresión reducida en suma de productos, y producto de sumas
- c) Dibujar el esquema en puertas de estas expresiones

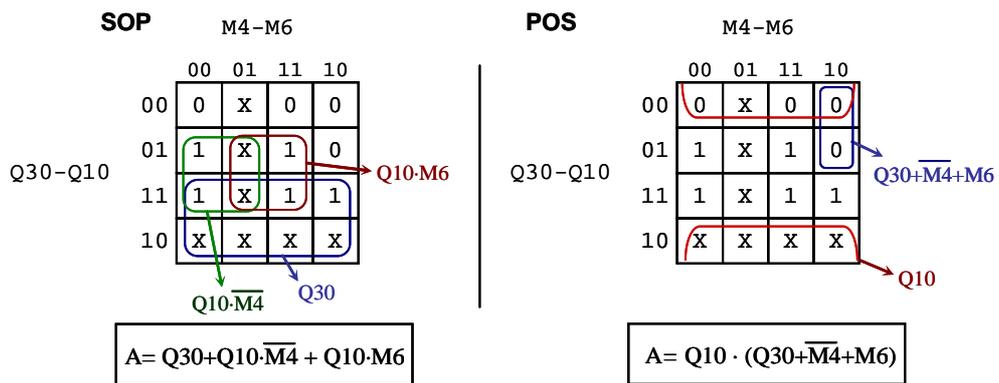
18.2. Solución

a) Tabla de verdad

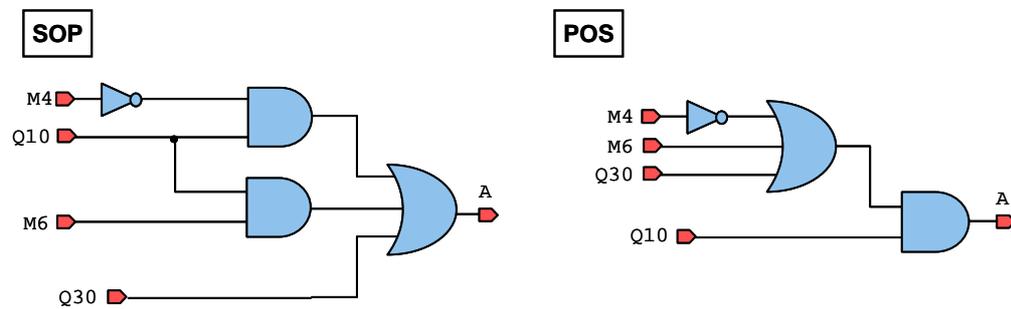
(Otro orden de las variables en la tabla de verdad es igualmente válido)

Q30	Q10	M4	M6	A	
0	0	0	0	0	} Est \acute{a} moviéndose: no est \acute{a} dormido: no hay alarma
0	0	0	1	X	
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	1	} Se ha quedado dormido, son menos de las 4am: se le despierta
0	1	0	1	X	
0	1	1	0	0	} 10 minutos quieto, pero son m \acute{a} s de las 4am y menos de las 6am: se le deja dormir un poco
0	1	1	1	1	
1	0	0	0	X	} No es posible, si lleva 30 minutos quieto, tambi \acute{e} n llevar \acute{a} 10 minutos quieto
1	0	0	1	X	
1	0	1	0	X	
1	0	1	1	X	
1	1	0	0	1	} No pueden ser menos de las 4am y m \acute{a} s de las 6am } M \acute{a} s de media hora quieto: se le despierta
1	1	0	1	X	
1	1	1	0	1	
1	1	1	1	1	

b) Expresión reducida en suma de productos y producto de sumas



c) Esquema en puertas

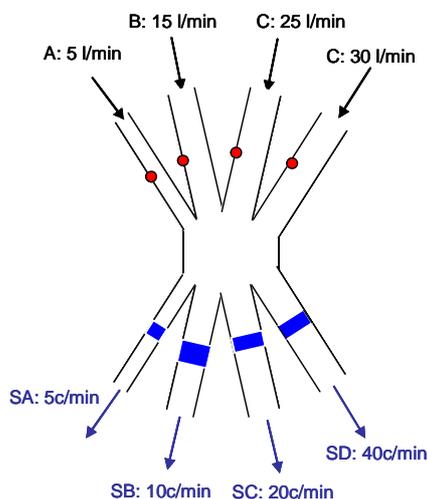


19. Tuberías

19.1. Enunciado

Supongamos que hay un nudo de tuberías, 4 de entrada y 4 de salidas. La tubería A aporta de media 5 litros por minuto, la B 15 litros/minuto, la C 25 litros/minuto y la D 30 litros/minuto. Cuatro sensores, uno por tubería de entrada, nos indican por qué tubería está circulando el agua. Las tuberías de salida son SA, SB, SC y SD y pueden recoger 5, 10, 20 y 40 litros por minuto respectivamente. Cada tubería de salida está regulada por una válvula que únicamente tiene dos estados: cerrada (un cero lógico) o abierta (un uno lógico). Teniendo en cuenta que sólo puede circular agua en dos tuberías de entrada simultáneamente, activar las válvulas de las tuberías de salida necesarias para que salga tanto caudal de agua como entra.

- Representar la tabla de verdad de la función
- Obtener las funciones lógicas simplificadas para las cuatro válvulas
- Implementar el circuito de control de la válvula de la tubería SB utilizando sólo puertas NAND.
- Implementar el circuito de control de la válvula de la tubería SC empleando decodificadores de 3 a 8.

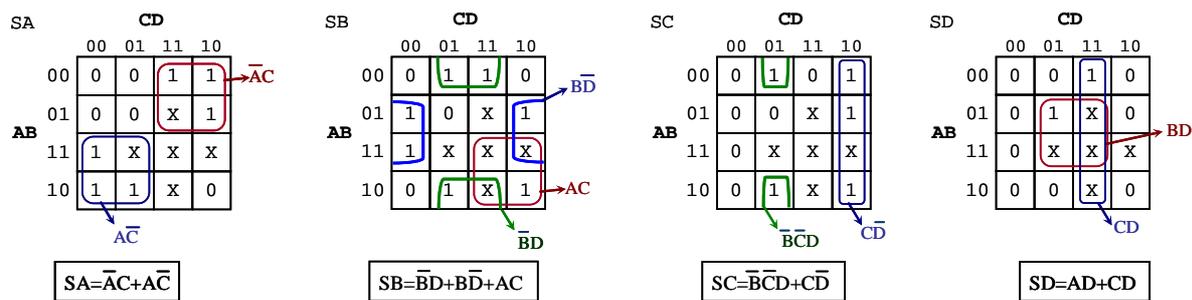


19.2. Solución

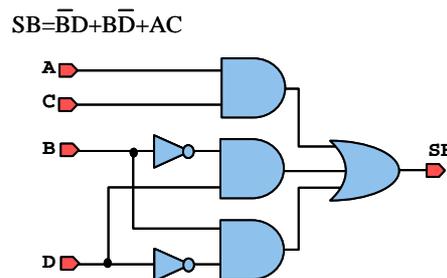
a) Dibujamos la tabla de verdad sabiendo que no va a haber más de dos tuberías por las que entren agua (que son "1" en la tabla de verdad). En los casos que impliquen más de dos tuberías de entrada con agua pondremos una X en las salidas, ya que ese caso nunca se va a dar. Esto nos permitirá simplificar el mapa de Karnaugh. En la tabla se han incluido dos columnas que indican en número de litros/minuto que entran, para facilitarnos el cálculo de las válvulas que debemos de abrir de modo que salga el mismo caudal que entra

entradas					salidas				
5	10	25	30		5	10	20	40	
A	B	C	D	entran	SA	SB	SC	SD	salen
0	0	0	0	0	0	0	0	0	0
0	0	0	1	30	0	1	1	0	30
0	0	1	0	25	1	0	1	0	25
0	0	1	1	55	1	1	0	1	55
0	1	0	0	10	0	1	0	0	10
0	1	0	1	40	0	0	0	1	40
0	1	1	0	35	1	1	1	0	35
0	1	1	1	--	X	X	X	X	--
1	0	0	0	5	1	0	0	0	5
1	0	0	1	35	1	1	1	0	35
1	0	1	0	30	0	1	1	0	30
1	0	1	1	--	X	X	X	X	--
1	1	0	0	15	1	1	0	0	15
1	1	0	1	--	X	X	X	X	--
1	1	1	0	--	X	X	X	X	--
1	1	1	1	--	X	X	X	X	--

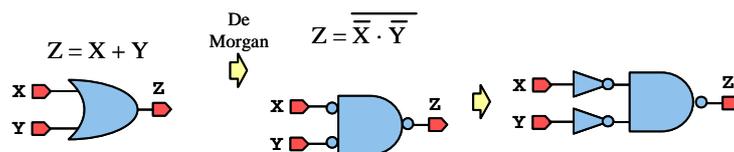
b) A continuación se muestran las funciones simplificadas en suma de productos. Se podía haber realizado también en producto de sumas.



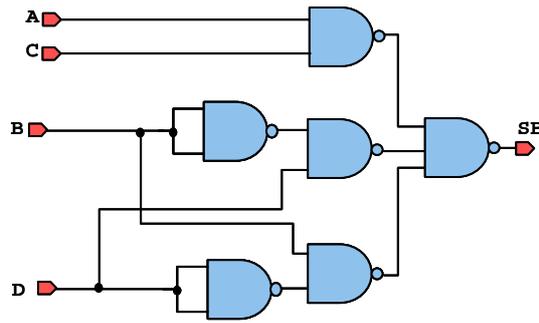
c) A continuación se muestra la función lógica SB en puertas:



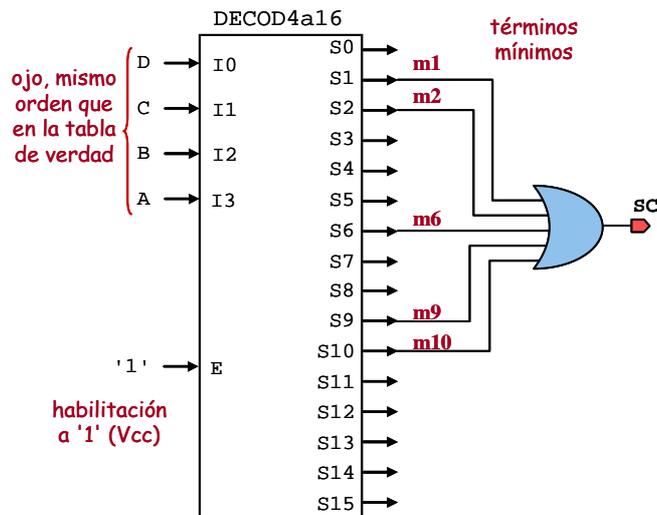
Sabiendo que una puerta OR se puede poner en función de una puerta NAND:



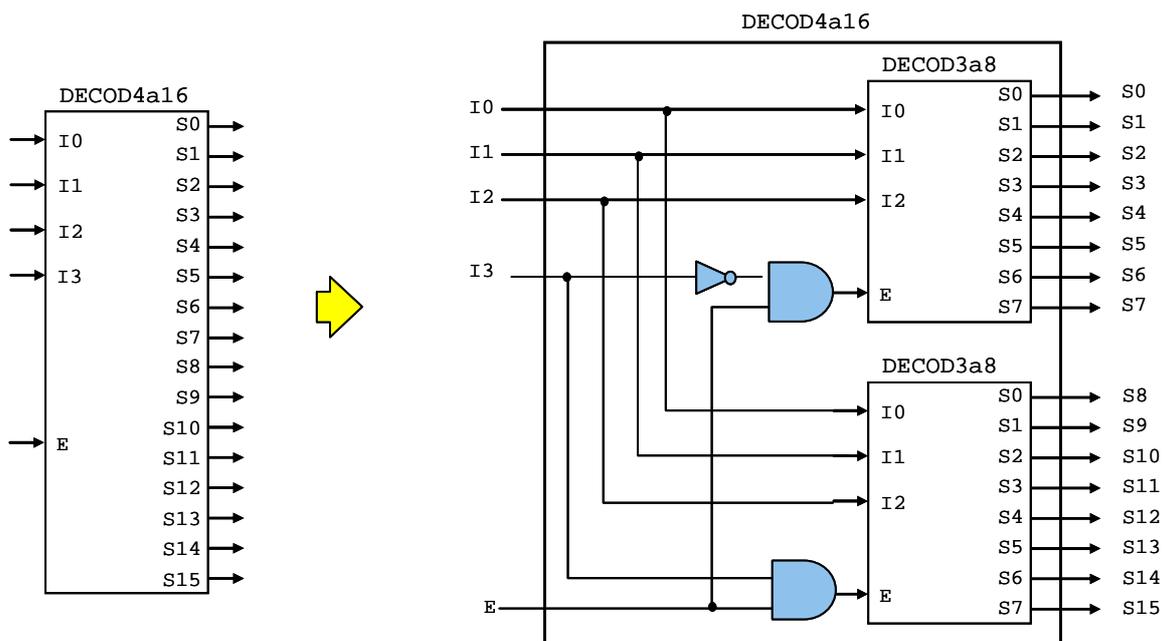
Y que un inversor se puede realizar uniendo las dos entradas de una NAND, sustituyendo el diseño quedará:



d) Nos piden realizar el diseño con decodificadores de 3 a 8. Como tenemos 4 entradas, en realidad necesitamos un decodificador de 4 a 16. El diseño con decodificadores se realiza los términos mínimos de la función lógica y uniéndolos con una puerta OR.:



Como tenemos decodificadores de 3 a 8, tenemos que realizar indicar cómo hacemos el decodificador de 4 a 16:



20. Báscula

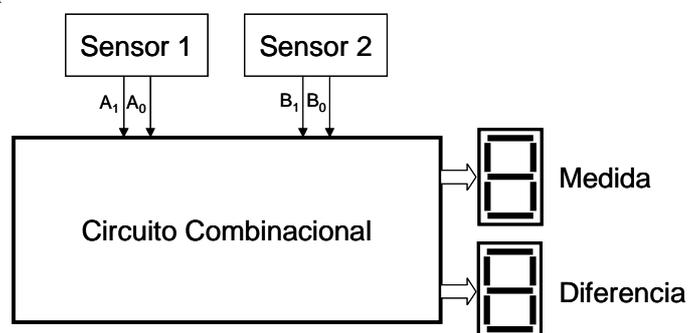
20.1. Enunciado

Una báscula utiliza dos sensores. Estos sensores son idénticos y están midiendo la misma pieza, aunque su medida puede diferir.

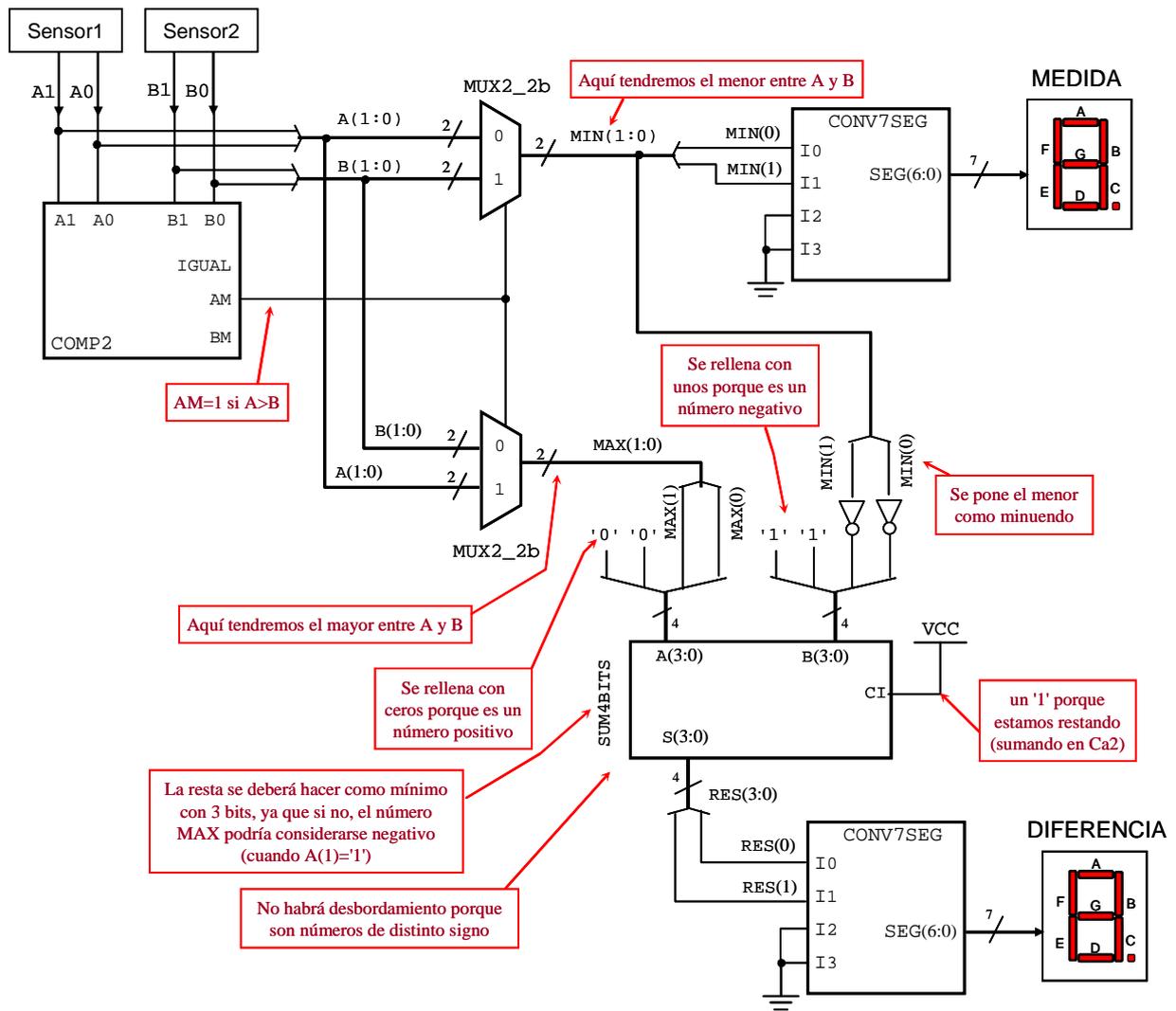
Las dos medidas se entregan a un sistema digital codificadas cada una con 2 bits (A_1A_0 para el sensor 1 y B_1B_0 para el sensor 2).

Para visualizar la medida se dispone de dos *displays* de 7 segmentos. En uno de ellos se visualizará la diferencia entre los sensores y en el otro la menor de las medidas en el caso de que no coincidan las lecturas de los dos sensores.

Diseñar mediante bloques combinatoriales estándares (multiplexores, decodificadores, sumadores, comparadores...) el esquema del circuito combinatorial.



20.2. Solución



21. Nivel de depósitos de agua

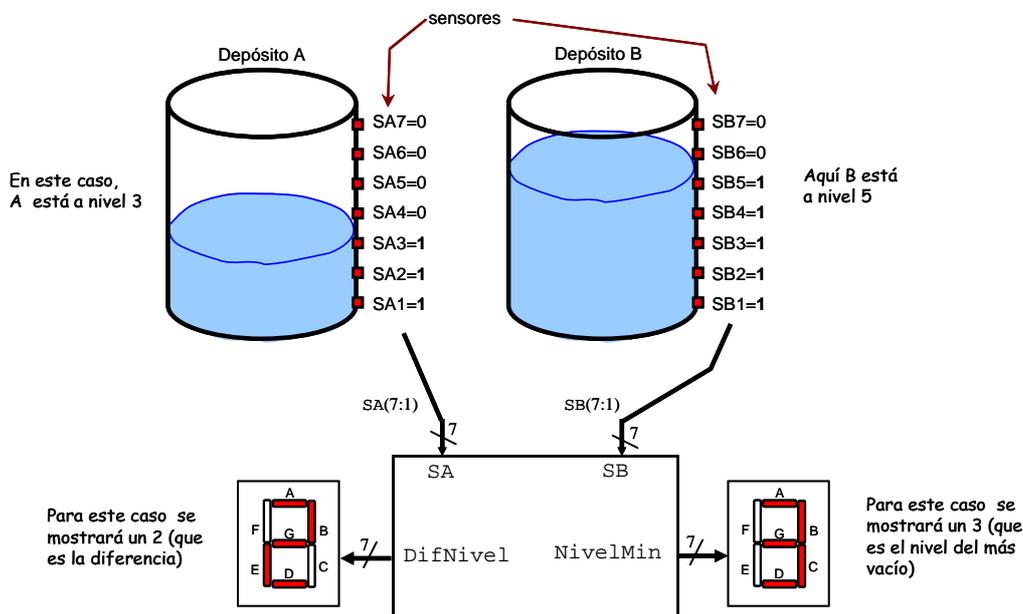
21.1. Enunciado

Se tienen dos depósitos de agua de los que se quiere conocer en cada momento el nivel de agua del que esté más vacío, y la diferencia de nivel respecto al mayor (en valor absoluto).

Para conocer el nivel, cada depósito tiene 7 sensores distribuidos a lo alto del depósito. Cada sensor dará un '1' lógico si está cubierto de agua, y un '0' lógico si está al aire.

La salida se dará mediante dos displays de 7 segmentos, uno para indicar el nivel del más vacío, y otro para indicar la diferencia.

Como se tienen 7 sensores para conocer el nivel, el rango de valores va desde 0 a 7. Se supone que ningún sensor va a fallar, por lo tanto, si un sensor indica un '1' lógico, todos los sensores que estén debajo de él darán un '1' lógico (pues el agua los cubrirá también).



Se pide realizar el diagrama de bloques del circuito:

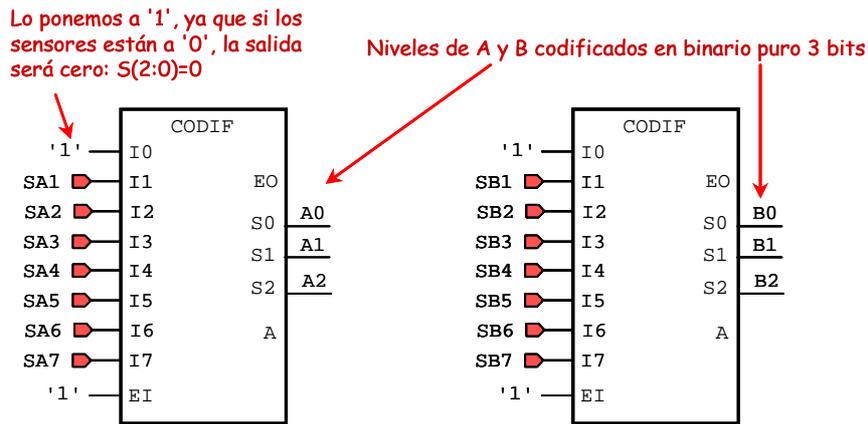
Para el diseño se podrá emplear cualquiera de los siguientes bloques sin necesidad de describirlos en puertas (no todos son necesarios): multiplexores, sumadores, codificadores, decodificadores, demultiplexores, comparadores y decodificadores de 7 segmentos. Todos ellos de uno o varios bits. Además se podrá emplear cualquier tipo de puertas lógicas. Cualquier otro bloque deberá ser descrito en función de los bloques citados o en puertas.

Es muy importante indicar todos los nombres de cada señal y su índice (o peso lógico).

Se valorará la reducción de componentes del circuito y la sencillez.

21.2. Solución

Para empezar, una buena opción será codificar las señales de entrada (las que provienen de los sensores). Ya que tenemos 7 señales, y como el rango va de 0 a 7, las podemos codificar en 3 bits. Necesitamos por tanto un codificador con prioridad de 8 a 3 para cada depósito.



Esto se puede hacer directamente, aunque si tienes dudas, lo puedes comprobar con la tabla de verdad.

EI	I7	I6	I5	I4	I3	I2	I1	I0	S2	S1	S0	A	EO	Nivel
0	X	X	X	X	X	X	X	X	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	0	0	0	1	1	0	0	1	1	0	1
1	0	0	0	0	0	1	1	1	0	1	0	1	0	2
1	0	0	0	0	1	1	1	1	0	1	1	1	0	3
1	0	0	0	1	1	1	1	1	1	0	0	1	0	4
1	0	0	1	1	1	1	1	1	1	0	1	1	0	5
1	0	1	1	1	1	1	1	1	1	1	0	1	0	6
1	1	1	1	1	1	1	1	1	1	1	0	1	0	7
1	el resto de combinaciones son imposibles								X	X	X	X	X	X

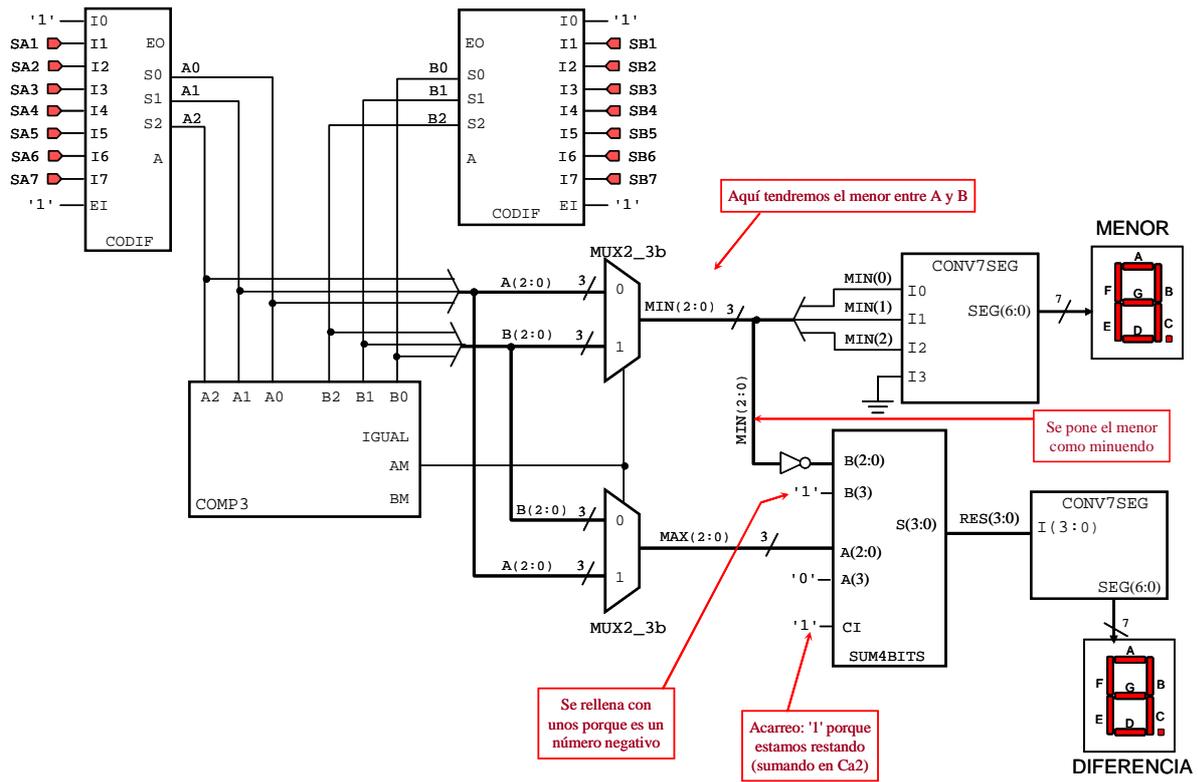
Estos '1' en realidad en el codificador pueden ser X, pero en nuestro caso no es posible.

De la tabla de verdad se puede comprobar que para el nivel 0, la entrada I0 vale '0'.

Debido a que cuando un sensor está en contacto con el agua todos los sensores que están debajo también lo están, se rellenan con '1' y no con X (se muestra en un triángulo).

El resto de combinaciones no son posibles.

Una vez codificadas las señales de los depósitos, el resto del circuito se puede hacer como sigue (similar al problema 20 de la báscula):



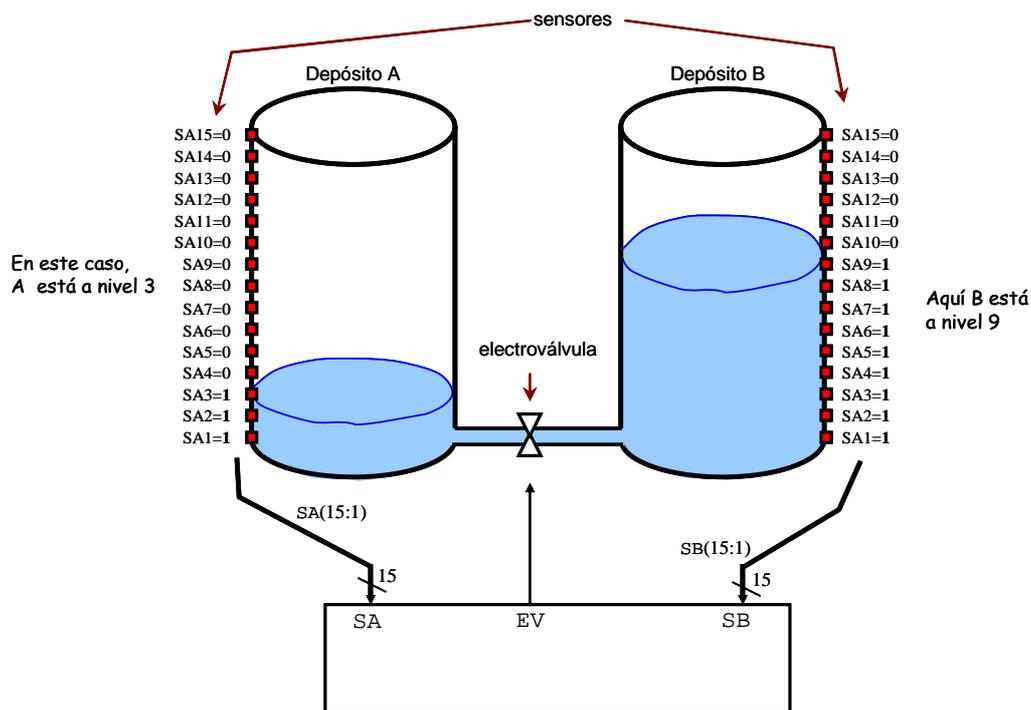
22. Depósitos de agua comunicados

22.1. Enunciado

Se tienen dos depósitos de agua comunicados por una tubería que tiene una electroválvula (ver figura). El nivel de agua de cada depósito se mide con 15 sensores distribuidos uniformemente a lo alto de cada depósito. Cada sensor dará un '1' lógico si está cubierto de agua, y un '0' lógico si está al aire. Como se tienen 15 sensores para conocer el nivel, el rango de valores va desde 0 a 15. Se supone que ningún sensor va a fallar, por lo tanto, si un sensor indica un '1' lógico, todos los sensores que estén debajo de él darán un '1' lógico (pues el agua los cubrirá también).

Se quiere realizar un circuito que accione la electroválvula cuando uno de los depósitos contenga el doble o más de agua que el otro, en este caso la salida **EV** será '1'. Mientras que **EV** será '0' cuando la diferencia del contenido de agua entre los depósitos sea menor del doble. Al accionar la válvula el depósito de mayor cantidad de agua se vaciará y el de menor cantidad de agua se llenará por ser un sistema de vasos comunicantes.

Para realizar el circuito se pueden usar los siguientes bloques combinacionales: decodificadores, codificadores, comparadores, multiplexores, demultiplexores y sumadores de cualquier número de bits. Así como puertas lógicas. Cualquier otro bloque se deberá realizar en función de los anteriores o en puertas lógicas. Para cada bloque utilizado se debe especificar claramente qué bloque es, las señales de entrada y salida, y el ancho de bus.



22.2. Solución

Lo primero es codificar las señales de los sensores de cada uno de los depósitos. Se hace de la misma manera que el problema 21, con la única diferencia que ahora se usa un codificador de 16 a 4, en vez de uno de 8 a 3.

Las salidas codificadas (**A** y **B**) se comparan y se obtiene la mayor (**MAX**) y la menor (**MIN**) (ver el problema 21).

La salida menor se multiplica por dos, obteniendo (**MINX2**), y se compara con la mayor (**MAX**).

Si la salida menor multiplicada por dos (**MINX2**) es mayor o igual que la mayor (**MAX**) accionaremos la electroválvula (**EV=1**), si no, la mantendremos cerrada (**EV=0**).

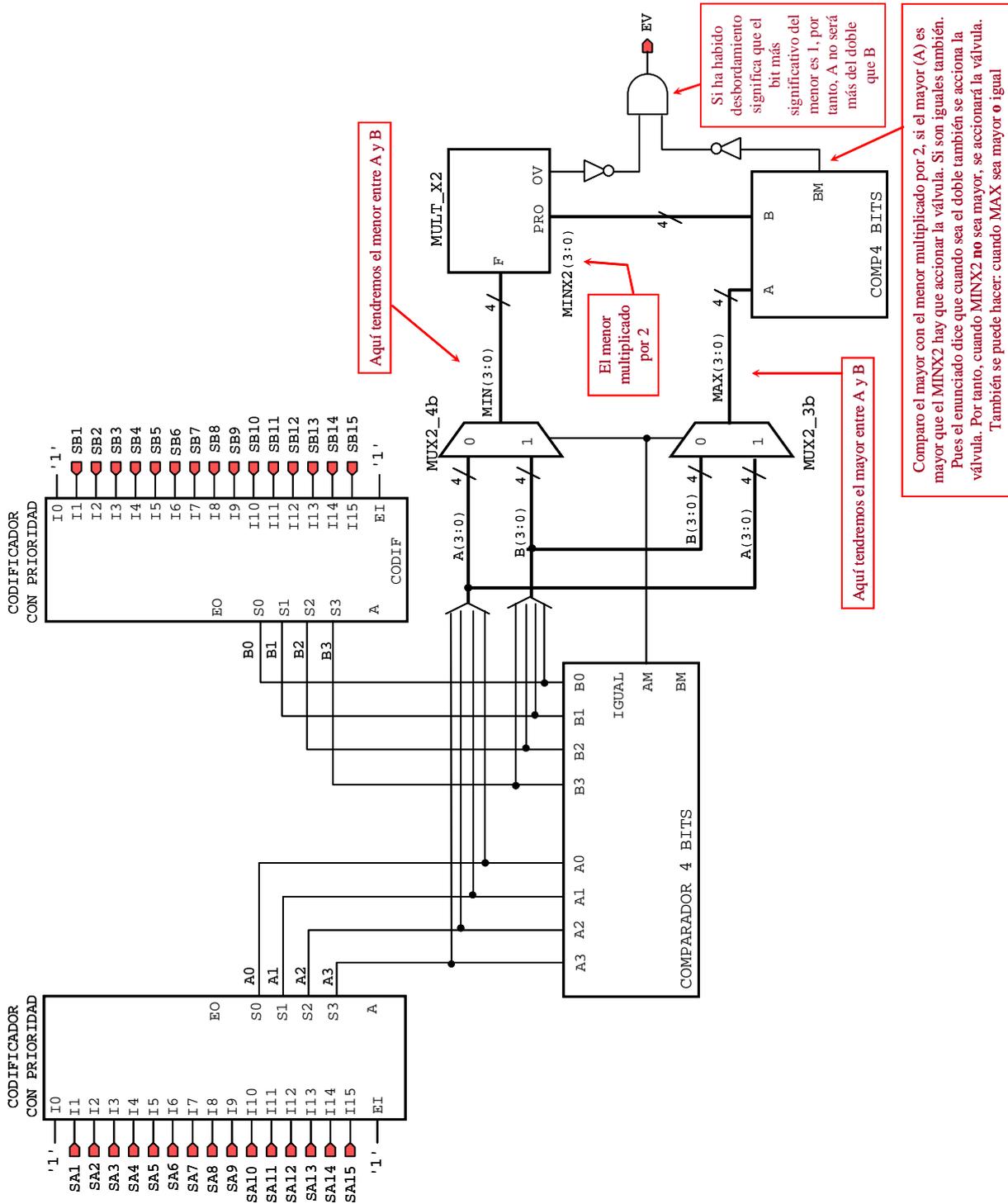
Sin embargo hay que considerar si ha habido desbordamiento. Si la multiplicación se desborda, significa que ese depósito tiene un nivel de 8 o mayor. Ya que $8 \times 2 = 16$, y 16 es mayor que 15 (el valor máximo posible). Por lo

tanto, si se desborda la multiplicación significa que el depósito de mayor cantidad de agua no podrá tener más del doble de agua que el otro (porque no puede tener más de 15).

Por eso se debe poner una puerta AND antes de la salida **EV**: Si es mayor del doble y no se ha desbordado.

Otra manera de hacerlo es mirar el bit más significativo del menor (**MIN**). Si **MIN(3) = 1**, significa que tiene un nivel de agua igual o mayor que 8, por lo tanto, el otro depósito no puede tener más del doble de agua.

En la figura se muestra el diseño. Todos los bloques se pueden usar tal cual menos el del multiplicador, ya que el enunciado no dice que lo tengamos.



El bloque multiplicador x2 se puede hacer de dos maneras:

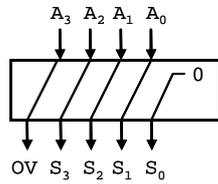
1. Desplazando a la izquierda

2. Sumando el número a sí mismo

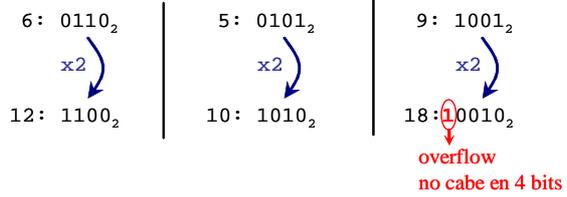
Evidentemente, la primera forma es más sencilla y más barata, aunque los dos dan el mismo resultado.

A continuación se muestra el multiplicador hecho con el desplazador a la izquierda, y tres ejemplos de multiplicación. El último ejemplo, al multiplicar 9 (1001) se desborda y vemos también que el bit 3 es uno.

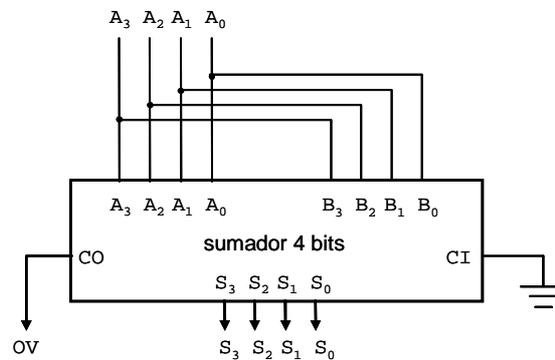
Multiplicador x2



Ejemplos:



El multiplicador realizado con un sumador de cuatro bits se muestra a continuación:



23. ALU

23.1. Enunciado

Realizar una unidad aritmético lógica con dos operandos de 4 bits (OPA y OPB) y 4 operaciones. Las operaciones que se realizarán son: suma, resta, complemento a 2 (Ca2) y complemento a 1 (Ca1). Para seleccionar una operación se dispone de 4 pulsadores:

- BTN0: realiza la suma $S=OPA+OPB$
- BTN1: realiza la resta. $S=OPA-OPB$
- BTN2: devuelve el complemento a 2 de OPB
- BTN3: devuelve el complemento a 1 de OPB.

Siendo BTN0 la operación de mayor prioridad, y BTN3 la de menor (para el caso en que se pulsen más de un botón a la vez). Por ejemplo, si se pulsan a la vez BTN1 y BTN2 el circuito devolverá la resta y no el Ca2. Como salida se tendrá la señal RESUL de 4 bits.

a) Realizar el diagrama de bloques del circuito:

Para el diseño se podrá emplear cualquiera de los siguientes bloques sin necesidad de describirlos en puertas (no todos son necesarios): multiplexores, sumadores, codificadores, decodificadores, demultiplexores y comparadores. Todos ellos de uno o varios bits. Además se podrá emplear cualquier tipo de puertas lógicas. Cualquier otro bloque deberá ser descrito en función de los bloques citados o en puertas.

b) Incluir la señal de desbordamiento, sabiendo que:

Para el caso de la suma se considera que los operandos están en binario puro.

Para la resta y el Ca2, los operandos de entrada están codificados en Ca2

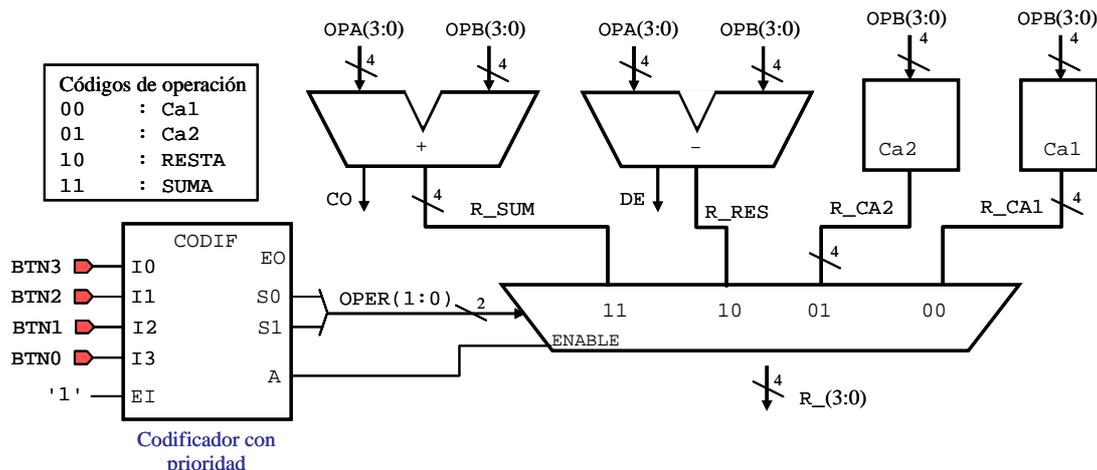
Para el Ca1 el operando de entrada está codificado en Ca1.

Se valorará la reducción de componentes del circuito y la sencillez.

23.2. Solución

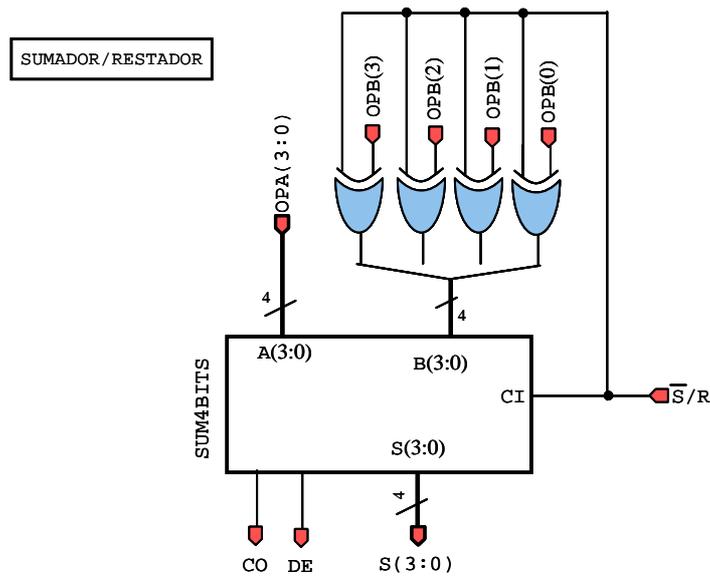
Hay muchas posibles soluciones para este diseño.

Inicialmente el circuito podría diseñarse de la siguiente manera.



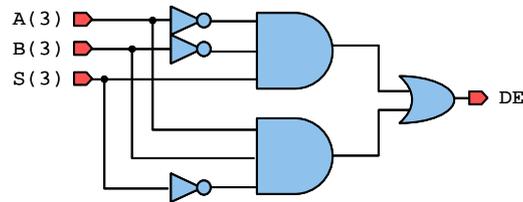
Según el enunciado, no disponemos del restador y de los bloques que realizan el Ca2 y Ca1.

El restador se puede hacer con un sumador, haciendo el complemento a dos del minuendo. Y que como nunca tendremos que hacer una suma y resta a la vez, podemos realizar un bloque sumador/restador:



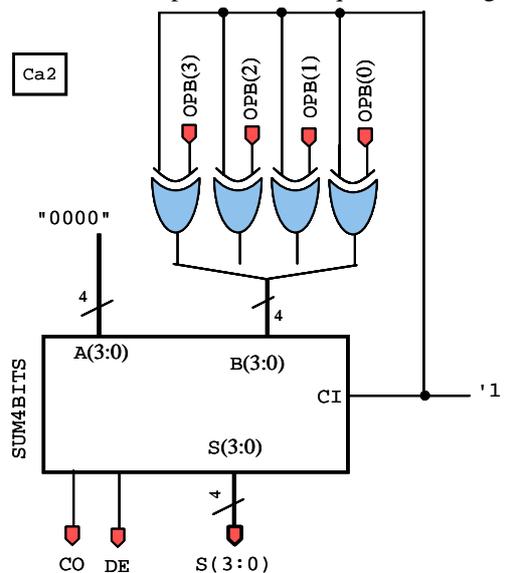
La señal DE corresponde al desbordamiento de la resta (hecha como suma) en complemento a dos, y se puede calcular de distintas maneras, una de ellas es (condición de desbordamiento en la suma en Ca2):

A3	B3	S3	DE
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

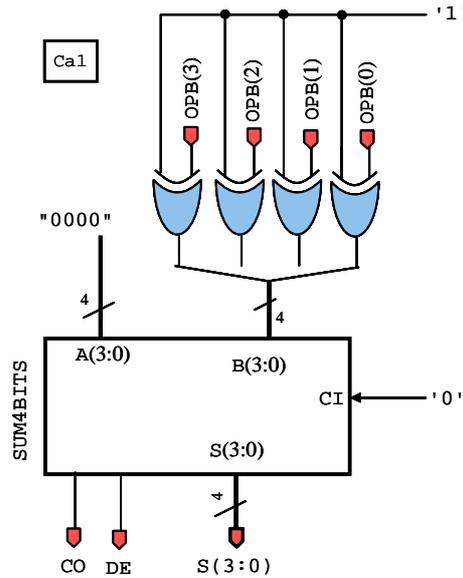


Por lo tanto podemos sustituir los bloques sumador y restador por el nuevo bloque.

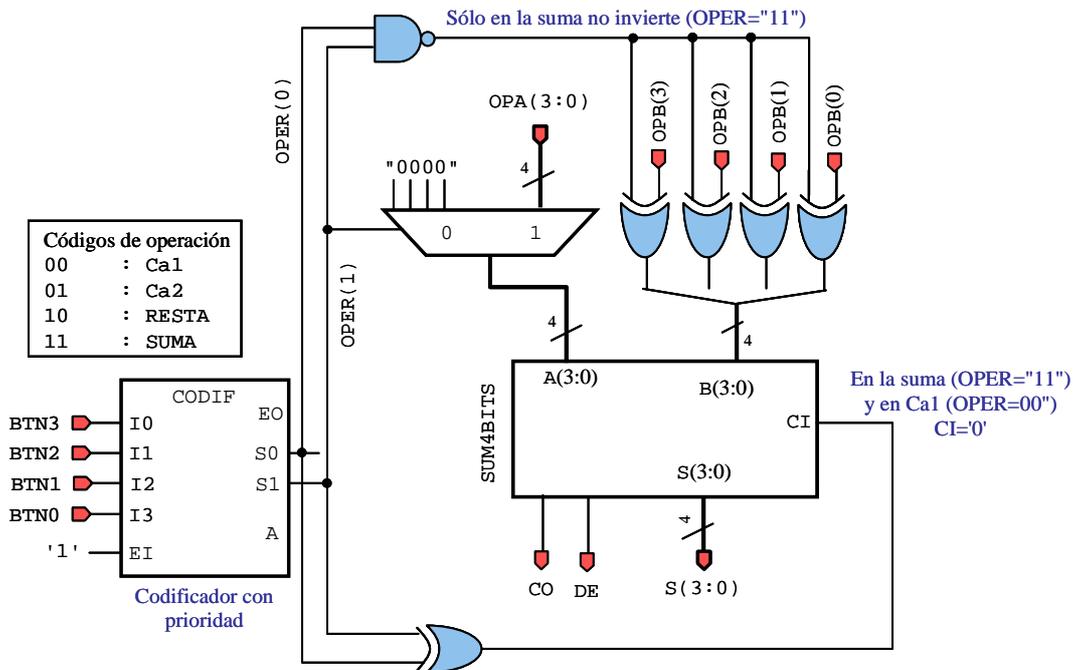
Hacer el bloque que calcula el complemento a dos de B, va a ser sencillo, simplemente tenemos que invertir B y añadirle uno. Como necesitamos añadir uno, podemos utilizar el mismo bloque sumador. De hecho podemos hacer igual que si fuésemos a realizar una resta, pero haciendo que OPA sea igual a cero.



Y por último, podemos hacer el complemento a uno simplemente invirtiendo B. Para ello podemos usar un bloque a parte, o utilizar lo mismo que hemos usado, sólo que no añadimos el acarreo de entrada.

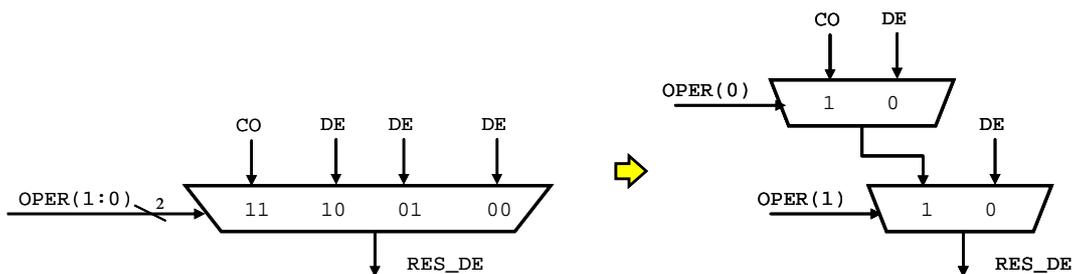


Evidentemente esta una forma ineficiente de hacer el complemento a uno si sólo necesitásemos realizar esta operación, pero al tener que realizar sumas y restas, nos quedará un diseño muy compacto.



b) Desbordamiento:

Y por último queda la señal de desbordamiento. En la suma se selecciona el acarreo de salida (CO), ya que están en binario puro; en la resta y Ca2 se selecciona DE, y en Ca1 cualquiera de ellas ya que siempre será cero (porque se le suma 0). Esta señal se quedaría así.

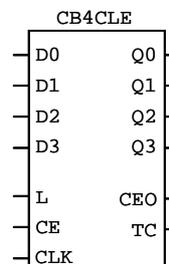


24. Contador 1

24.1. Enunciado

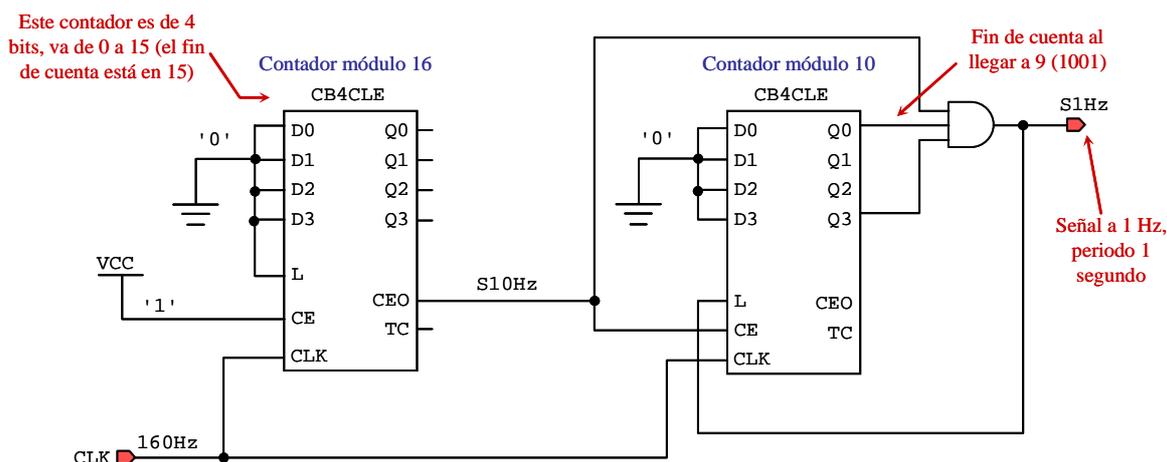
Mediante contadores síncronos de 4 bits como el mostrado en la figura,

- Diseñar un circuito que a partir de una señal de reloj CLK de frecuencia 160Hz proporcione una señal de 1Hz
- Realizar un segundero: A partir de la salida anterior obtener un circuito que periódicamente encienda y apague un LED durante un segundo. Esto es: 1 segundo encendido, el siguiente segundo apagado, el siguiente segundo encendido, ...
- Añadir al circuito anterior otro LED que esté encendido mientras el otro LED esté apagado, y que se apague mientras el otro esté encendido.

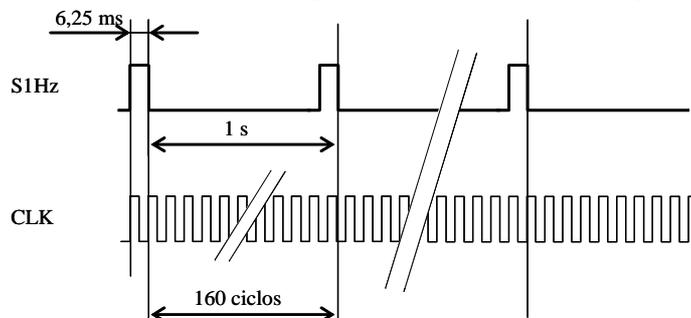


24.2. Solución

- Tenemos un reloj que va a 160Hz, y queremos una salida a 1Hz. Para ello tenemos que dividir la frecuencia entre 160. Con contadores de 4 bits, podemos dividir primero entre 16 y obtener una señal de 10Hz y dividir ésta entre 10 y obtener la salida de 1Hz.

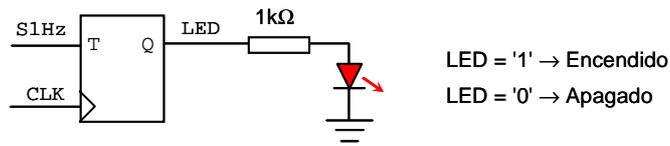


- La señal que hemos obtenido es de periodo 1 segundo. Y en cada segundo sólo estará un ciclo de reloj de Clk a '1' y el resto estará a cero. Cada ciclo de reloj dura 6,25 ms ($1s/160=0,00625s$). Ya que 160 ciclos de 6,25ms hacen 1 segundo. Así que la señal de salida S1Hz tiene la siguiente forma de onda (la imagen no está a escala).

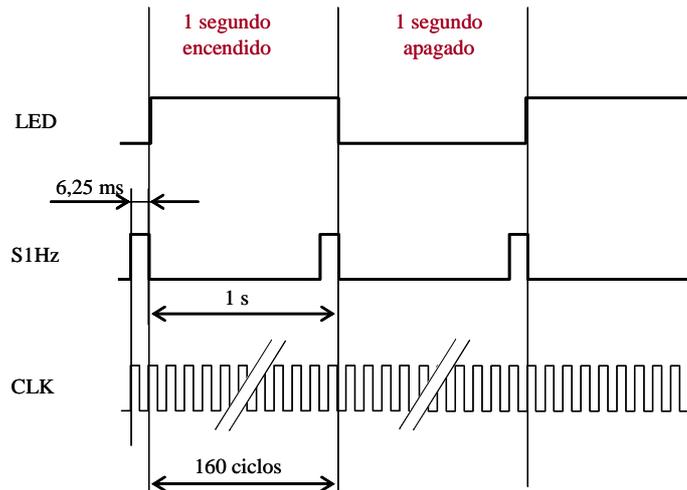


Ahora queremos que cada segundo se encienda o apague un LED. Como cada segundo tenemos un pulso de la señal S1Hz, podemos hacer que esta señal se encargue de dar la orden de encendido o apagado. Para mantener

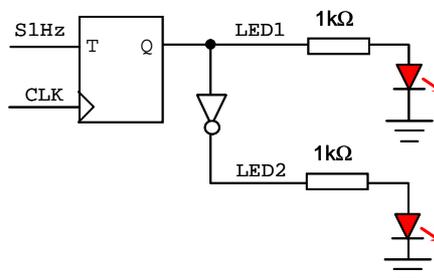
(memorizar) la orden después de que se termine el pulso de S1Hz usaremos un biestable. Podemos usar el biestable T, que cambia de valor cada vez que recibe un pulso.



La forma de onda resultante que va al LED será:



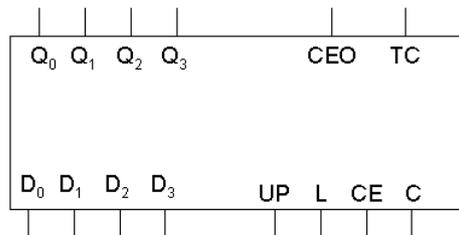
c) Si queremos dos LED que se enciendan alternativamente, bastará con conectar el otro LED a la salida negada del biestable.



25. Contador 2

25.1. Enunciado

- a) Necesitamos obtener una señal de 20 KHz a partir de una señal de reloj de 1 MHz. Para ello disponemos de dos contadores síncronos BCD 74HC160 como el de la figura. Dibujar el diagrama lógico necesario.



- b) Se quiere implementar un reloj digital que muestre la hora (de 1 a 12) y los minutos (de 0 a 59) en el formato H H : M M.
Para ello se dispone de 4 contadores BCD 74HC160, 4 decodificadores BCD- 7 segmentos y 4 displays de 7 segmentos. Indique las conexiones necesarias de todos los componentes.

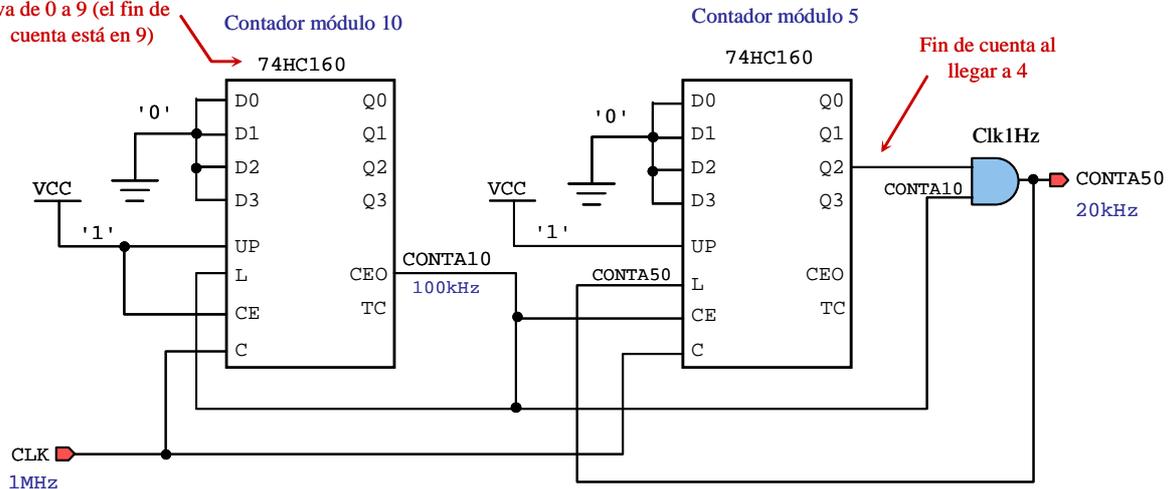
25.2. Solución

a)

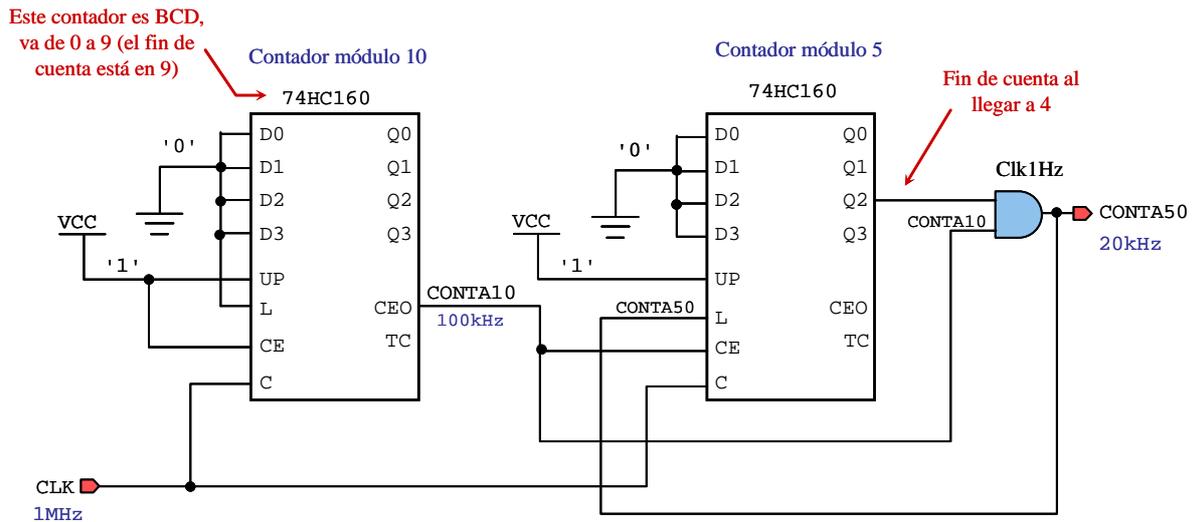
La frecuencia tiene que disminuir desde 1MHz hasta 20kHz, lo que supone que necesitamos dividir la frecuencia entre 50. Como disponemos de contadores BCD, podemos dividir primero entre 10 y luego entre 5. Como es un contador BCD, cuenta de 0 a 9, y por tanto para el primer contador no hace falta calcular el fin de cuenta pues ya lo hacen las señales CEO y TC.

A continuación se muestra una posible solución, aunque hay muchas otras posibles

Este contador es BCD,
va de 0 a 9 (el fin de
cuenta está en 9)



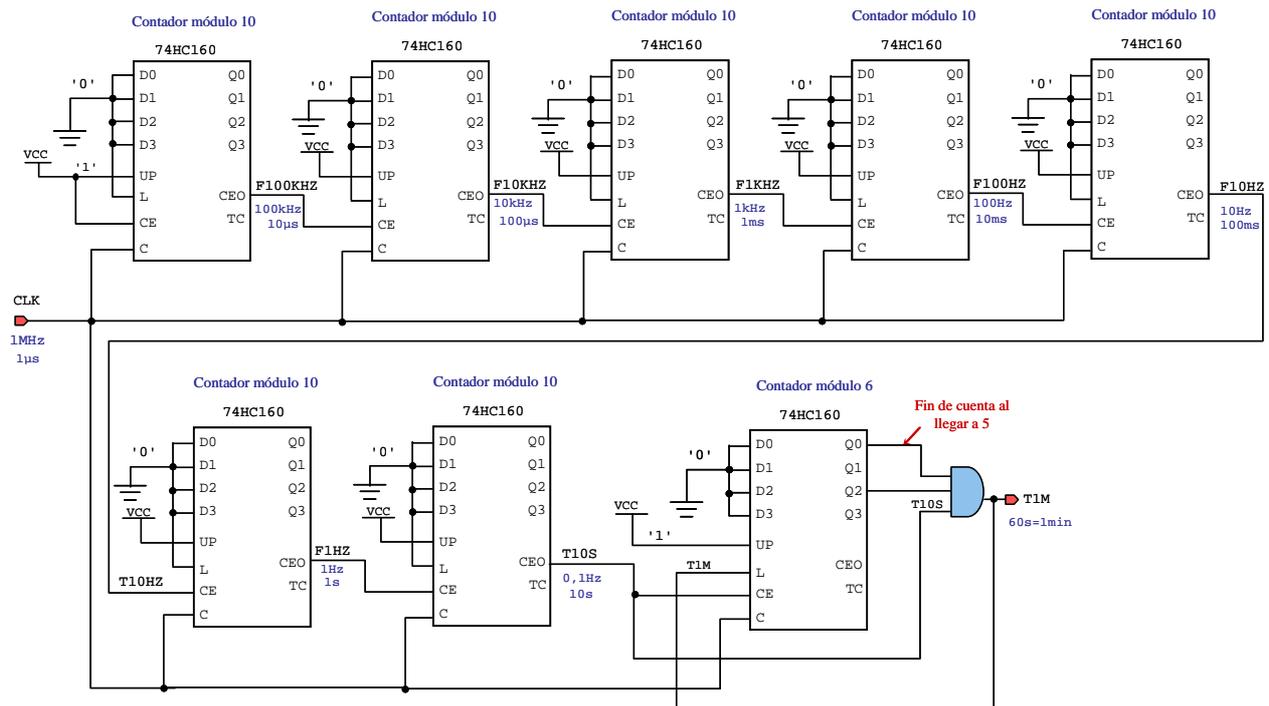
Por ejemplo, el primer contador realizar la cuenta de 0 a 9, que es todo el rango del contador (pues es BCD), por tanto no es necesario cargar ningún dato al llegar al fin de cuenta, él sólo irá de 9 a 0. Pero si fuese un contador hexadecimal (de 0 a 15) entonces sí que sería necesario. Así que podría quedar así:



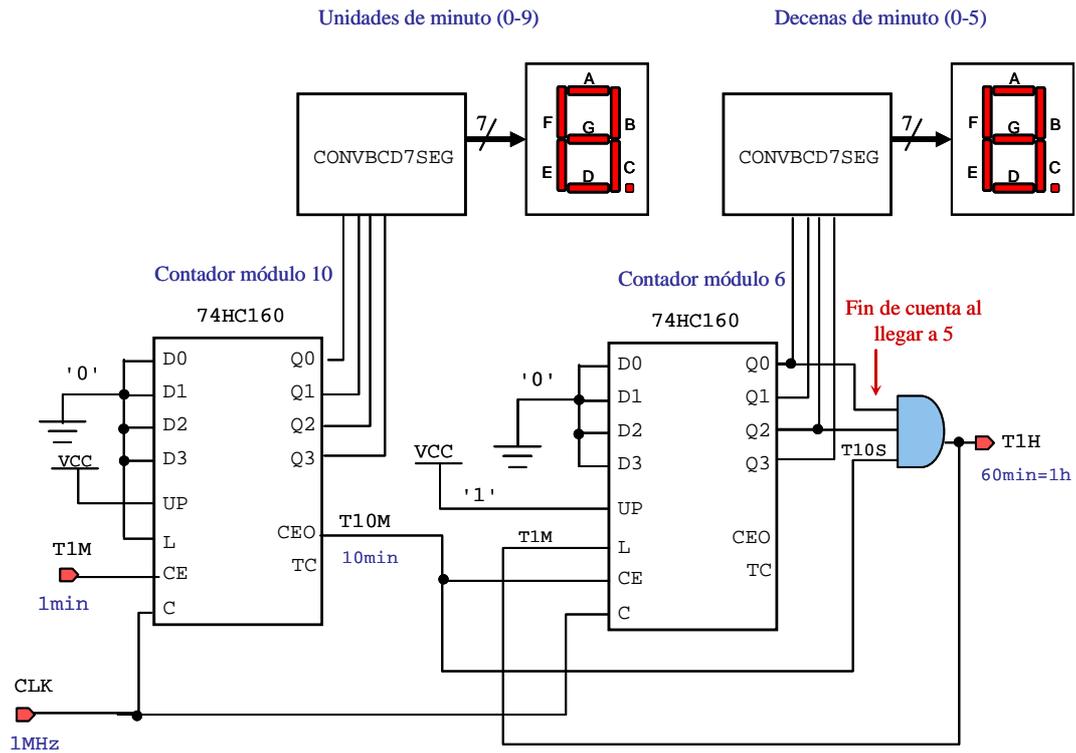
b)

Podemos suponer que tenemos una señal de entrada de periodo 1 minuto, sin embargo, aprovechamos para explicar cómo se generaría a partir del reloj de 1MHz. Vamos a usar contadores BCD y por tanto no hace falta usar el final de cuenta ni la carga para todos aquellos que cuenten de 0 a 9 (ver apartado anterior, **si fuese contadores módulo 16 sí sería necesario**). Para el último contador, debido que cuenta de 0 a 5, sí que necesitamos generar un fin de cuenta cuando $Q=0101$, y $CE=1$, que realizará la carga (L) a cero.

Fíjate como van cambiando las frecuencias y los periodos a la salida de cada contador.

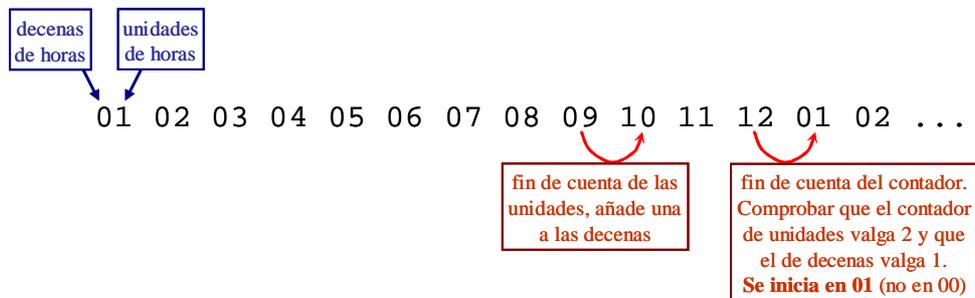


Ahora la salida que hemos generado, que nos indica cada minuto que pasa, será la entrada de nuestro reloj. Y generaremos un contador de 60 minutos:



Y ya tenemos una señal que nos da un pulso cada hora, y con ella generaremos el contador de 12 horas. Este contador de horas es especial, ya que la cuenta es 01, 02, ..., 08, 09, 10, 11, 12 y vuelta a 01. Tendremos un contador para las unidades y otro para las decenas de horas. Las decenas de horas sólo van a ser 0 ó 1. Ya que el formato es de 12 horas y no de 24 horas. El contador de unidades realizará una cuenta de 1 a 9 cuando las decenas valgan 0, y otra de 0 a 2 cuando las decenas valgan 1, al terminar esta última cuenta, las unidades volverán a 1 (después de las doce viene la una, no existe la hora cero en formato de 12 horas).

La secuencia de la cuenta será:



Así que tendremos el paso de 09 a 10 que es el paso normal de añadir una decena. Y por otro lado tenemos el paso del fin de cuenta, de 12 a 01, que tiene dos particularidades:

- Por usar contadores BCD para poder mostrar fácilmente la cuenta de horas por los *displays*, se nos complica la cuenta, y como 12 no es divisible entre 10, tenemos que comprobar el fin de cuenta no sólo con el valor de las decenas, sino también con el de las unidades.
- Por otro lado, para el contador de unidades, el inicio de cuenta, desde 12 a 01 es distinto que de 09 a 10. Si usamos un contador BCD, la cuenta de 09 a 10 se inicia sola por el propio contador y no tenemos que preocuparnos. Para el caso de 12 a 01 tenemos que cargar un 1

El contador nos quedaría:

26. Contador 3

26.1. Enunciado

Realizar el circuito que obtiene las señales **Q10** y **Q30** del "café electrónico" descrito en el problema 18.

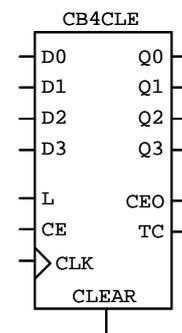
Para ello se dispone un reloj (**CLK**) de 12kHz y una señal de entrada (**MOV**), que indica si el alumno se ha movido. Cuando la señal de entrada **MOV** vale '1' indica que hay movimiento, cuando vale '0' indica que el alumno está quieto.

El funcionamiento de las señales de salida es el siguiente:

- La señal **Q10** se debe poner a '1' cuando hayan transcurrido 10 minutos durante los que la señal **MOV** se ha mantenido a '0'. Antes de que transcurran 10 minutos, **Q10** valdrá '0'.
- La señal **Q30** se debe poner a '1' cuando hayan transcurrido 30 minutos durante los que la señal **MOV** se ha mantenido a '0'. Antes de que transcurran 30 minutos, **Q30** valdrá '0'.

Para realizar el circuito se pueden usar todos los bloques combinatoriales estudiados (comparadores, multiplexores, codificadores, sumadores,...) ,así como puertas lógicas y biestables. También se pueden utilizar contadores módulo 16 como el de la figura.

Para cada bloque utilizado se debe especificar claramente el tipo de bloque, las señales de entrada y salida, y el ancho de bus.



26.2. Solución

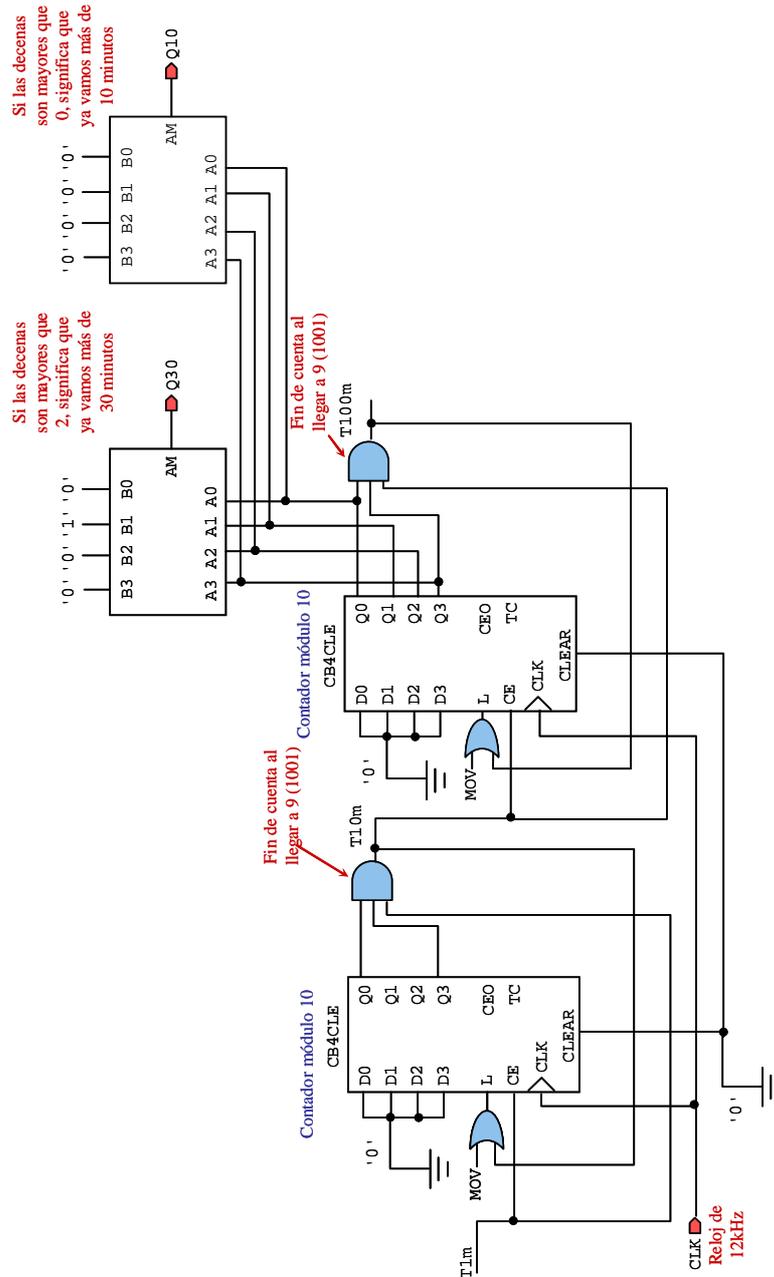
Tenemos que realizar un contador de minutos, cuando la cuenta sea mayor que 10 minutos, se activará la señal **Q10** (se podrá a uno) y cuando la cuenta sea mayor que 30 se activará **Q30**. (se podrá a uno).

La única diferencia con un contador normal es que cuando la señal **MOV** valga 1, empezaremos la cuenta desde cero. Para empezar la cuenta se puede usar el *load* (**L**) del contador y lo haríamos de manera síncrona cargando ceros en el dato, o bien lo podemos realizar de manera asíncrona utilizando el **CLEAR** de cada contador.

Como queremos contar minutos, lo primero que tenemos que hacer es obtener una señal periódica de un minuto. Para ello, pasamos primero los 12kHz a 1Hz, obteniendo una señal T1S de un segundo de periodo (dividiendo por 12000) y de esta señal, pasamos de una señal de un minuto (dividiendo por 60).

En cualquier momento que la señal **MOV** valga 1, el contador volverá al principio. En la figura de la izquierda se muestra cómo obtener la señal de periodo un segundo. En este caso el contador se inicializa asíncronamente cada vez que **MOV=1** , utilizando la señal **CLEAR**.

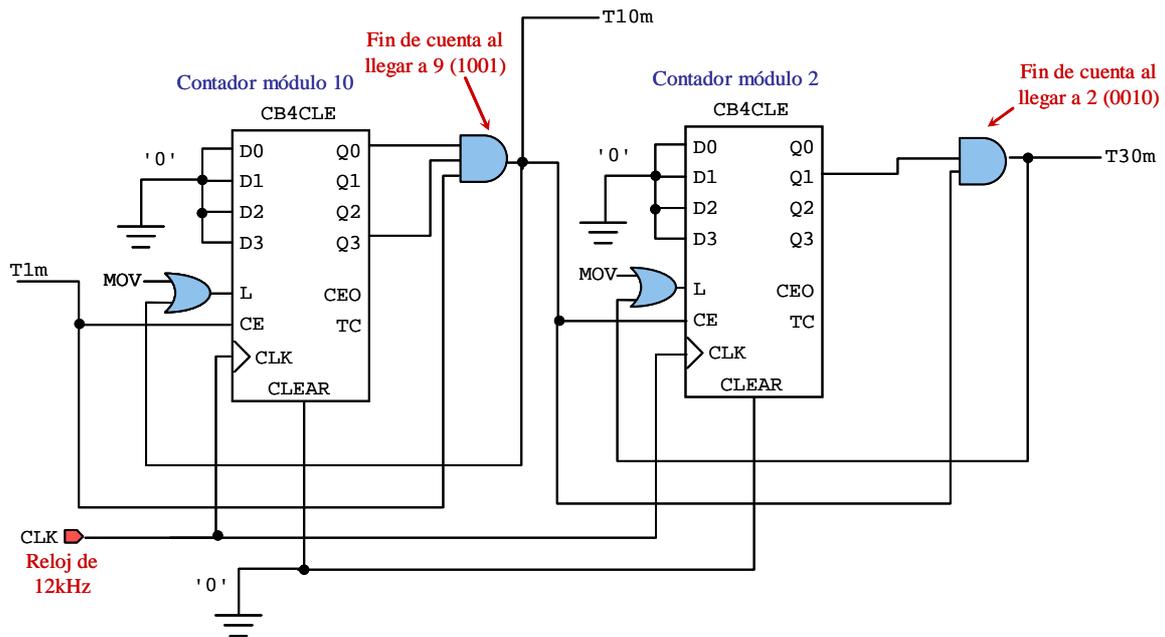
La otra opción es utilizar el **LOAD**, habría que hacer la OR de load que ya tenemos con la inicialización (esta opción se muestra en la figura de la derecha):



Otra alternativa sería hacer un contador de 30 minutos y un contador de 10 minutos. Como las unidades de minuto van a ser las mismas para los dos, podemos realizar un contador de 3 decenas de minuto y otro de 1 decena de minuto. De hecho, este contador lo tenemos ya hecho, es la señal T10m.

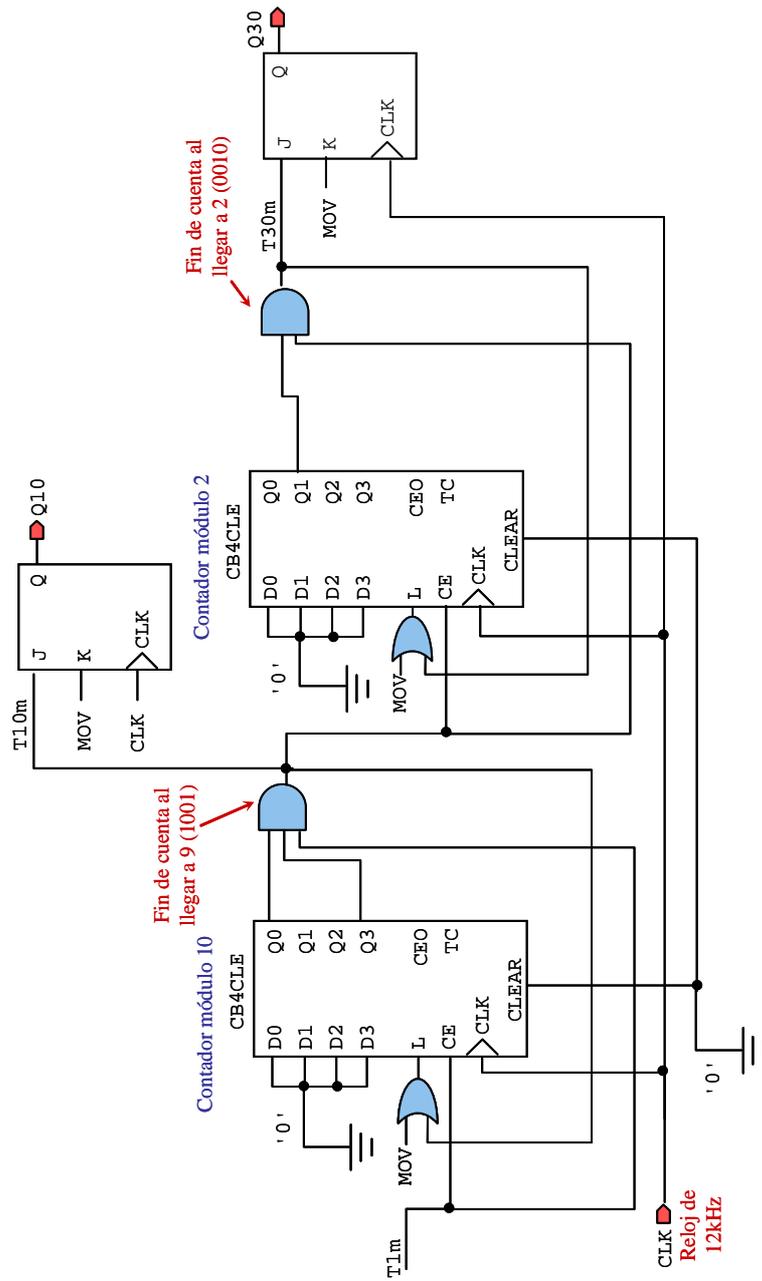
Con el contador de 30 minutos obtendremos la señal T30m.

Así pues, queremos obtener una señal de periodo de 10 minutos y otra de periodo de 30 minutos. Éstas señales se muestran en la siguiente figura:



Y ¿cómo conseguimos que una vez que hemos llegado a los 10 ó los 30 minutos se mantenga la señal activa para que suene la alarma? Podemos usar un biestable, por ejemplo un J-K, poniendo la señal T10m ó T30m en la entrada J, hará que el biestable se mantenga a 1 hasta que K se ponga a '1'. ¿Cuándo queremos que el biestable se ponga a 0? cuando se mueve, por tanto, en la entrada K del biestable ponemos la señal **MOV**. ¿Qué pasaría si MOV y T10m se ponen a uno simultáneamente? el biestable cambiaría de valor, pero no es importante, porque T10m (y T30m) se ponen a uno sólo durante un ciclo de reloj, por tanto será inapreciable, y en el siguiente ciclo de reloj tendremos que $J=0$ y $K=1$, por tanto el biestable volverá a 0. Además de que sería mucha casualidad que el alumno se moviese en el preciso instante en que el contador llega a su fin.

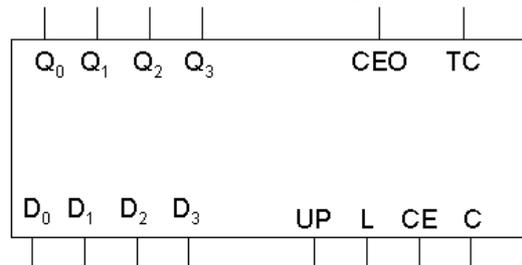
El circuito quedaría como sigue:



27. Contador 4

27.1. Enunciado

Empleando contadores síncronos BCD CI 74160 como el de la figura:

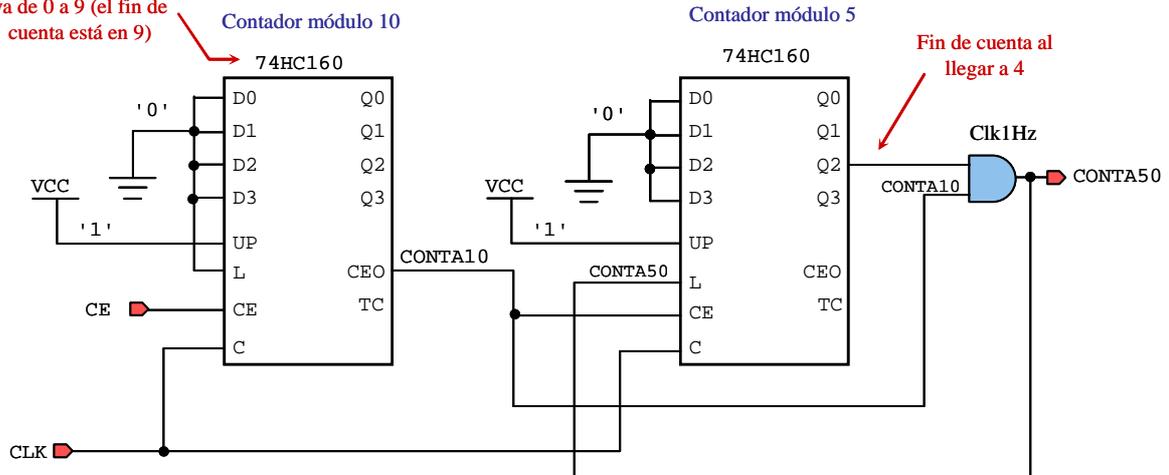


- Diseñar un contador síncrono módulo 50
- Diseñar, mediante bloques combinatoriales estándar, un sistema que cuente hasta un cierto número variable entre 0 y 99 marcado por el usuario, y que presente la cuenta en 2 displays de 7 segmentos.

27.2. Solución

a)

Este contador es BCD, va de 0 a 9 (el fin de cuenta está en 9)

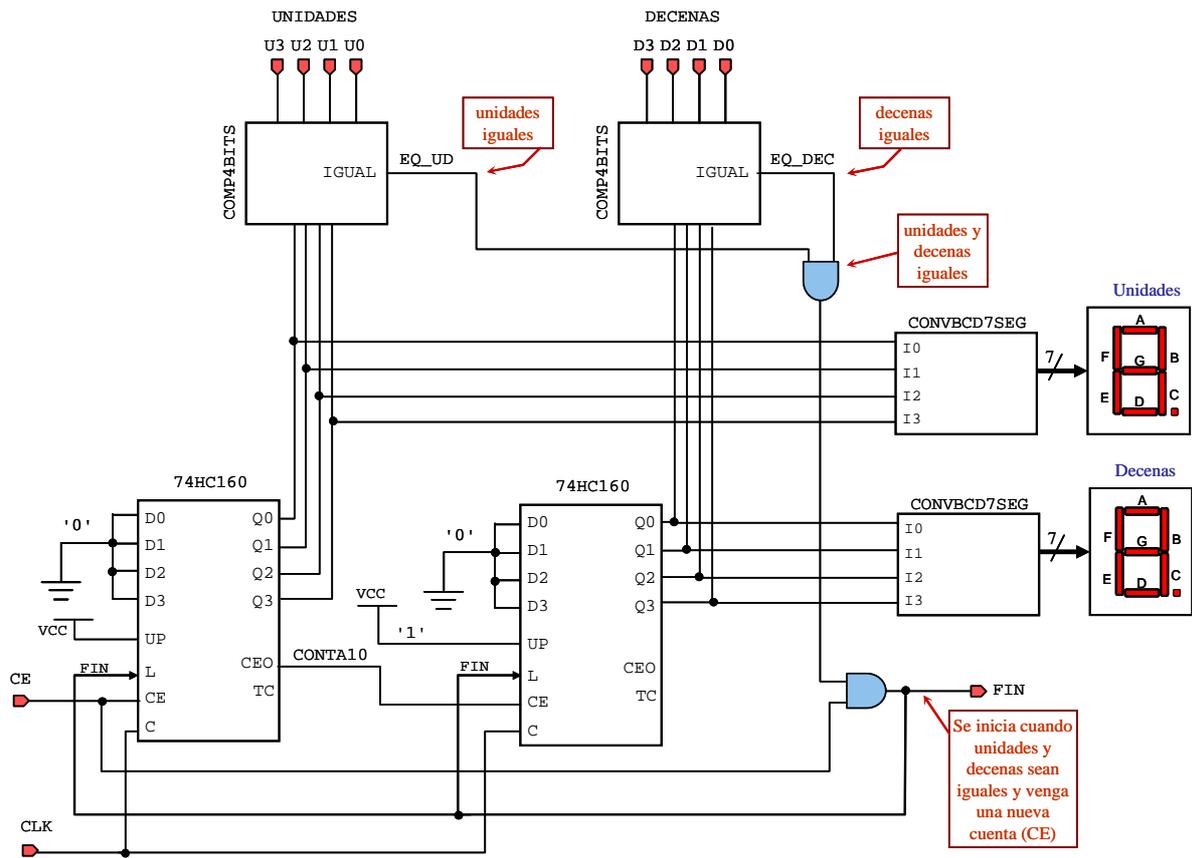


b)

El usuario introduce el número en BCD, con dos cifras BCD (cada una con 4 bits), una para las unidades y otra para las decenas.

La cuenta irá desde 0 hasta el número marcado, utilizándose un contador para unidades y el otro para decenas.

Utilizaremos dos comparadores, y cuando los contadores coincidan (sean iguales) al número introducido por el usuario, en la siguiente cuenta los dos contadores pasarán a cero.



28. Piano electrónico

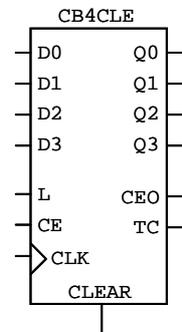
28.1. Enunciado

Se tiene un reloj (**CLK**) de 30 MHz, y contadores hexadecimales como el de la figura, se pide:

a) Realizar un circuito que a partir de la señal de reloj obtenga una señal (**F10KHZ**) de frecuencia 10 kHz. Esta señal estará un sólo ciclo de reloj a '1' y el resto a '0'.

b) Realizar un circuito que cuente entre 1 y 100 pulsos de señal anterior (**F10KHZ**). La salida se llamará (**FOUT**). Esto es, si cuenta un pulso, la FOUT tendrá el mismo número de pulsos que F10KHZ. Mientras que si cuenta 100 pulsos, FOUT tendrá un pulso por cada 100 de F10KHZ.

La cuenta máxima vendrá fijada por un número BCD de dos cifras introducido por el usuario.



c) ¿Cuáles serán las frecuencias y periodos mínimos y máximos que podrá tener la señal **FOUT**?

d) A partir de lo anterior, realizar un sintetizador digital simple que genere cuatro notas musicales (DO, RE, MI y FA). De manera resumida, el sintetizador de sonido genera señales periódicas cuya frecuencia viene determinada por la nota musical correspondiente. Aproximadamente, las frecuencias que queremos generar son:

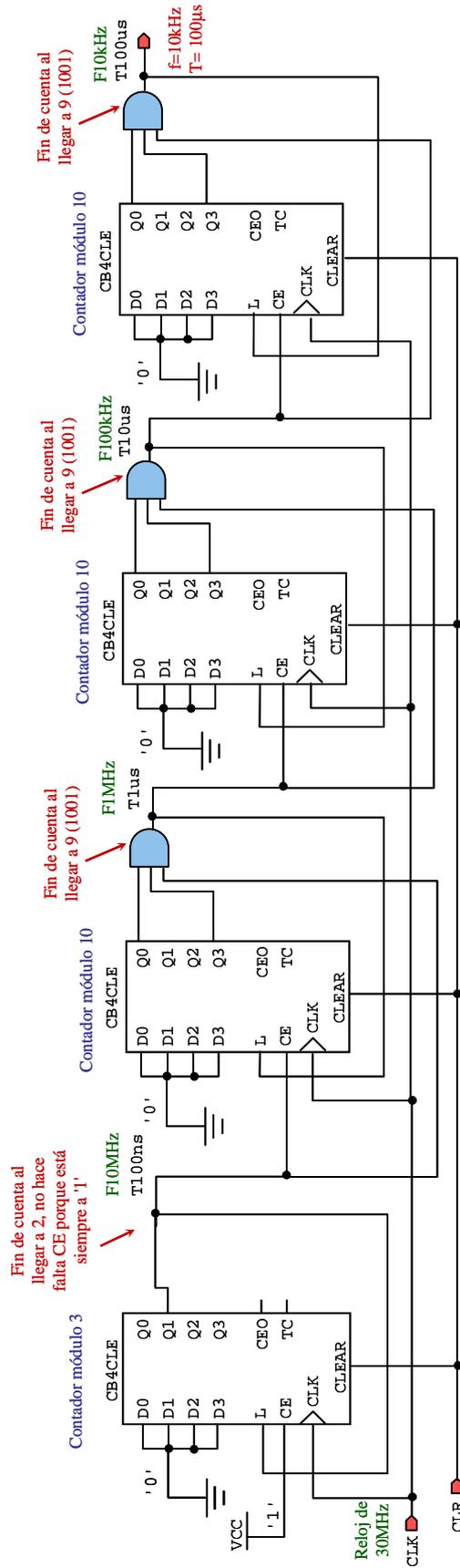
	DO	RE	MI	FA
frecuencia	263 Hz	294 Hz	333 Hz	345 Hz
periodo	3,8 ms	3,4 ms	3,0 ms	2,9 ms

Para facilitar el cálculo, se recomienda utilizar los periodos (no las frecuencias). La selección de la nota se realizará mediante cuatro pulsadores. Si se pulsan dos pulsadores a la vez, la nota más aguda tendrá prioridad (FA es la de mayor prioridad, DO es la de menor prioridad)

Para realizar el circuito se pueden usar todos los bloques combinatoriales estudiados, así como contadores, puertas lógicas y biestables. Para cada bloque utilizado se debe especificar claramente qué es, las señales de entrada y salida, y el ancho de bus.

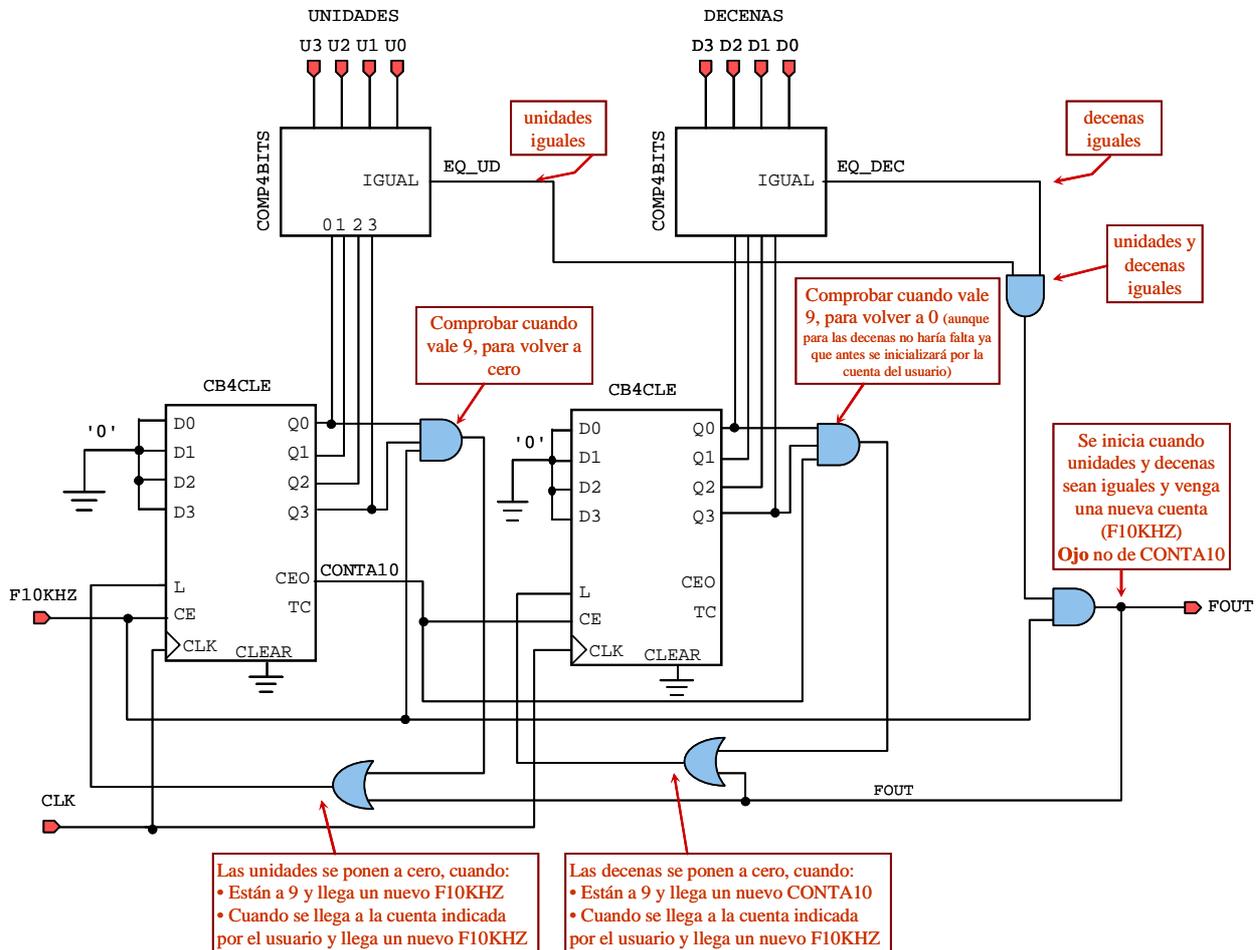
28.2. Solución

a) A partir de una señal de 30 MHz queremos obtener una de 10 kHz, por lo tanto, tenemos que contar 3000 ciclos de reloj, que cosas similares ya hemos hecho en los problemas de contadores anteriores.



b) A partir de la señal F10kHz (T100us) queremos contar entre 1 y 100 ciclos de esta señal. Este problema es similar al de junio de 2007. La diferencia con aquel problema es que ahora tenemos contadores hexadecimales. Por lo tanto, ahora la inicialización de los contadores (L: Load) se hace cuando se cumplen dos condiciones:

- Cuando el contador está a 9 y le llega una nueva cuenta
- Cuando se llega a la cuenta indicada por el usuario.



c) No hace falta haber hecho el aparatado anterior para saber la frecuencia y periodos máximos y mínimos de FOUT, ya que se puede deducir del enunciado:

- La frecuencia máxima y periodo mínimo corresponden con la de la propia señal F10KHZ. Cuando se realiza una sola cuenta. Por tanto:

$$f_{max} = 10\text{kHz} \quad ; \quad T_{min} = 100 \mu\text{s}$$

- La frecuencia mínima y periodo máximo corresponden a cuando se cuentan 100 pulsos. Por tanto la frecuencia se divide entre 100 y el periodo se multiplica por 100. Por tanto:

$$f_{min} = 100\text{Hz} \quad ; \quad T_{max} = 10 \text{ ms}$$

d) La señal inicial F10KHZ tiene $f = 10\text{kHz}$; $T = 100 \mu\text{s}$

Para obtener una señal de una frecuencia mayor tengo que ver la relación de frecuencias o periodos entre ambas. Por ejemplo.

Si F10KHZ tiene $f = 10\text{kHz}$ y quiero un DO de $f = 263 \text{ Hz}$ tengo que dividir 10kHz entre 263 Hz :

$$\frac{10000}{263} = 38$$

Esto también se puede hacer dividiendo los periodos, pero colocados al revés: dividir 3,8 ms entre 100µs, esto es

$$\frac{3800}{100} = 38$$

En este caso, la división de los periodos resulta más fácil, y por eso se recomendaba en el enunciado.

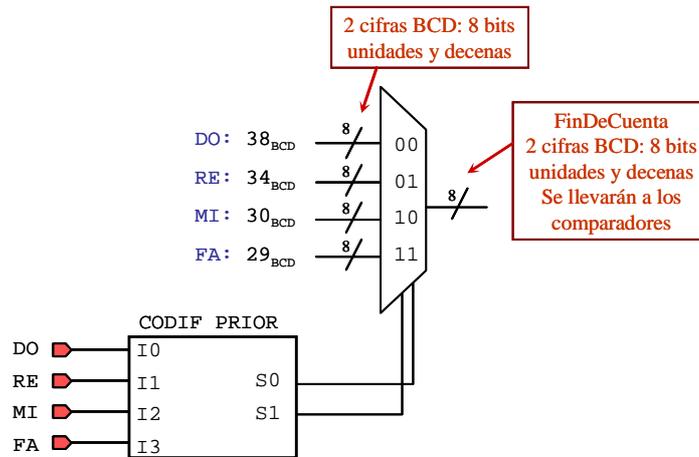
Para las otras notas se realiza igual:

	DO	RE	MI	FA
relación	38	34	30	29

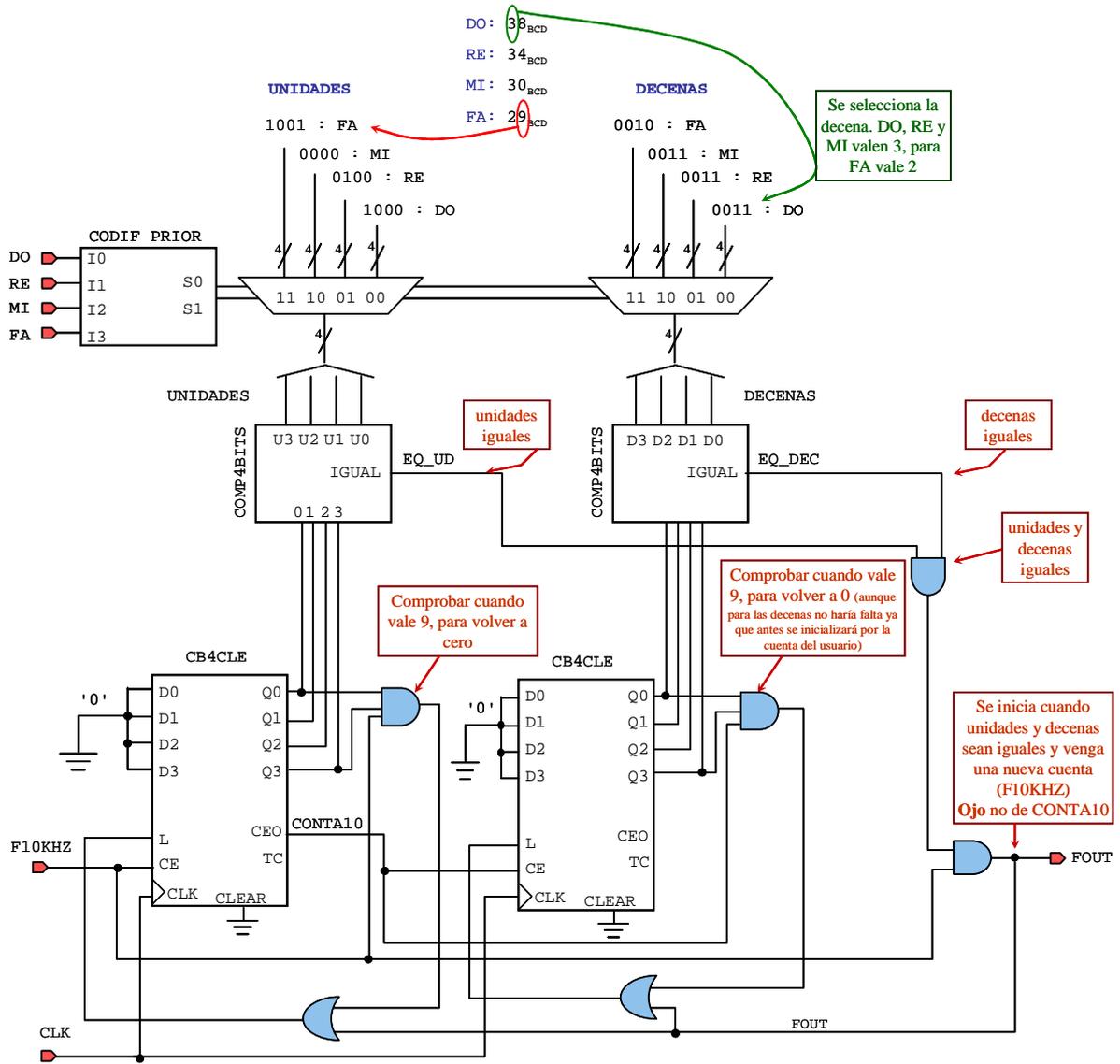
Por tanto, para realizar el circuito, bastará con contar el número indicado de cuentas según la nota.

Según el pulsador que esté presionado seleccionaremos una cuenta distinta. Como ya sabemos, para seleccionar se usan multiplexores, y como hay 4 pulsadores, y necesitamos 2 señales de selección, las tenemos que codificar.

Esto que queremos hacer se muestra esquemáticamente a continuación



Que unido al circuito anterior nos quedaría el circuito final:



29. Análisis de circuitos 2

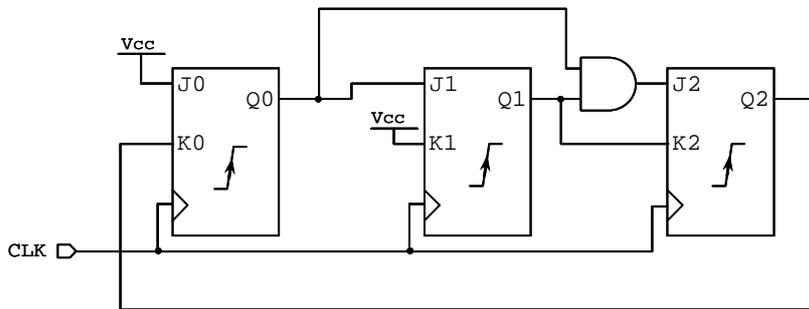
29.1. Enunciado

Para el siguiente contador suponiendo que el estado inicial es **000** ($Q_2=Q_1=Q_0=0$)

Dibujar el cronograma durante los ciclos de reloj que sean necesarios para mostrar la secuencia completa. En el cronograma se deben incluir al menos las señales CLK, Q0, Q1 y Q2.

Indicar la secuencia de salida.

Nota: es muy importante indicar los pasos que indiquen cómo se ha obtenido la secuencia. Aunque la secuencia sea la correcta, no se contabilizarán los ejercicios que sólo dibujen las formas de onda sin explicar nada.



29.2. Solución

Tenemos biestables J-K y una puerta AND.

Ponemos sus tablas de verdad para evitar confundirnos durante el ejercicio.

J	K	Q _t	Q _{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Q1	Q0	J2
0	0	0
0	1	0
1	0	0
1	1	1

Las ecuaciones de las entradas de los biestables son:

$J_2 = Q_1 \cdot Q_0$	$J_1 = Q_0$	$J_0 = 1$
$K_2 = Q_1$	$K_1 = 1$	$K_0 = Q_2$

En el tiempo inicial ($t=0$), $Q_2=Q_1=Q_0=0$

En estas condiciones, las entradas J-K de cada biestable son:

t=0 ($Q_2=Q_1=Q_0=0$)		
$J_2 = 0$	$J_1 = 0$	$J_0 = 1$
$K_2 = 0$	$K_1 = 1$	$K_0 = 0$

Así que podemos empezar a rellenar la tabla de transición de estados.

tiempo	Estado Actual			Entradas para el siguiente estado					
t	Q2	Q1	Q0	J2	K2	J1	K1	J0	K0
t=0	0	0	0	0	0	0	1	1	0
t=1									

Con estas condiciones, cuando llegue el siguiente flanco de reloj. Los estados de los biestables cambiaran a consecuencia de los valores de las entadas (J-K)

Así que en $t=1$, Q2 se mantendrá en el mismo valor (0) porque $J2=0$ y $K2=0$. Q1 se está poniendo a 0, por tanto, mantiene su valor (0). Mientras que Q0 se pone a uno, cambiando de 0 a 1 (esto lo puedes obtener de la tabla del J-K que hemos hecho al principio).

Entonces: en $t=1 \rightarrow Q2=0; Q1=0; Q0=1$, y lo incluimos en la tabla:

tiempo	Estado Actual			Entradas para el siguiente estado					
t	Q2	Q1	Q0	J2	K2	J1	K1	J0	K0
t=0	0	0	0	0	0	0	1	1	0
t=1	0	0	1						

→ El siguiente estado será 001

Para calcular los valores de J y K operamos de la misma manera que antes. A partir de los estados de los biestables y las ecuaciones de las J y K, obtenemos los siguientes valores para $t=1$:

t=1 (Q2=0; Q1=0; Q0=1)		
J2 = 0	J1 = 1	J0 = 1
K2 = 0	K1 = 1	K0 = 0

Los ponemos en la tabla:

tiempo	Estado Actual			Entradas para el siguiente estado					
t	Q2	Q1	Q0	J2	K2	J1	K1	J0	K0
t=0	0	0	0	0	0	0	1	1	0
t=1	0	0	1	0	0	1	1	1	0

Igual que antes: Con estas condiciones, cuando llegue el siguiente flanco de reloj. Los estados de los biestables cambiaran a consecuencia de los valores de las entadas (J-K)

Así que en $t=2$, Q2 se mantendrá en el mismo valor (0) porque $J2=0$ y $K2=0$. Q1 cambiará de valor porque $J1=1$ y $K1=1$, por tanto se pone a 1. Mientras que Q0 se pone a uno, por lo que se mantiene en el mismo valor. Rellenamos la tabla:

tiempo	Estado Actual			Entradas para el siguiente estado					
t	Q2	Q1	Q0	J2	K2	J1	K1	J0	K0
t=0	0	0	0	0	0	0	1	1	0
t=1	0	0	1	0	0	1	1	1	0
t=2	0	1	1						

→ El siguiente estado será 011

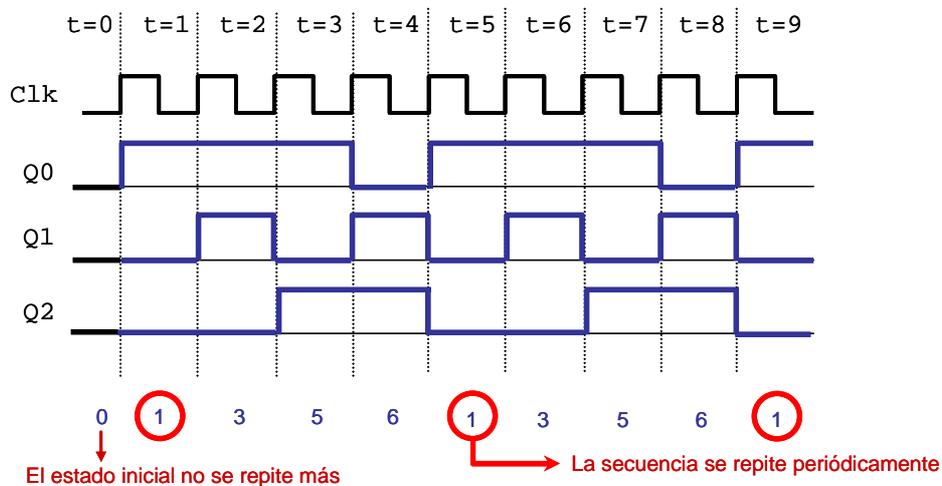
Y se continúa realizando el mismo proceso. La tabla queda:

tiempo	Estado Actual			Entradas para el siguiente estado					
t	Q2	Q1	Q0	J2	K2	J1	K1	J0	K0
t=0	0	0	0	0	0	0	1	1	0
t=1	0	0	1	0	0	1	1	1	0
t=2	0	1	1	1	1	1	1	1	0
t=3	1	0	1	0	0	1	1	1	1
t=4	1	1	0	0	1	0	1	1	1
t=5	0	0	1	0	0	1	1	1	0
t=6	0	1	1	...					

Vuelve al estado de t=1. el estado de t=0 no se repite más

Por tanto, la secuencia de salida es 000, 001, 011, 101, 110, 001, 011, ..., que si son números binarios la secuencia sería 0, 1, 3, 5, 6, 1, 3, 5, 6, 1, ... Como se puede apreciar, el 0 no vuelve a aparecer más.

El cronograma queda entonces:



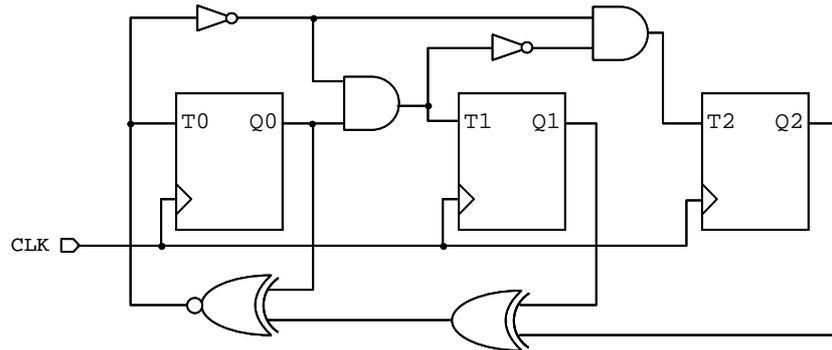
30. Análisis de circuitos 3

30.1. Enunciado

Indicar la secuencia del siguiente contador suponiendo que el estado inicial es $Q_0=Q_1=Q_2=0$.

Dibujar las formas de ondas para 9 ciclos de reloj

¿Con qué secuencia se corresponde?



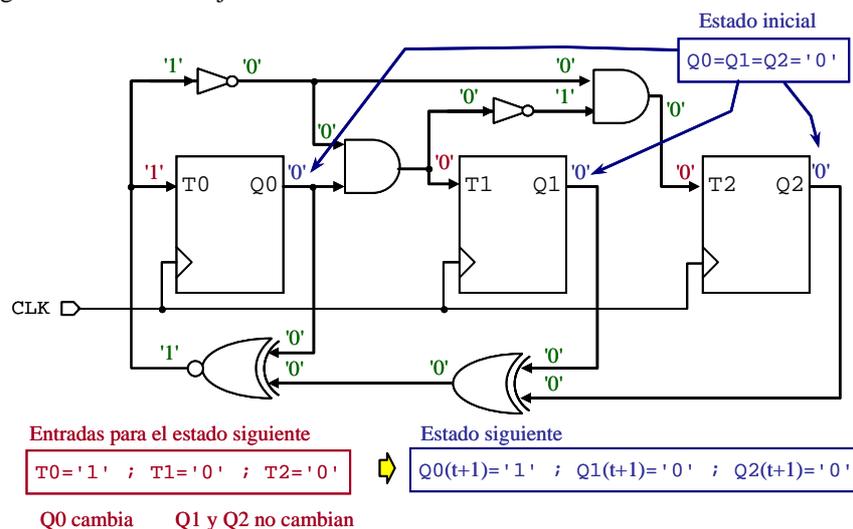
30.2. Solución

Tenemos biestables T, si la entrada es '1' en el siguiente ciclo de reloj el estado del biestable será el estado complementario al actual. Si la entrada es '0' se mantendrá el mismo estado.

Para hallar la secuencia podemos ir yendo estado a estado:

Inicialmente partimos de "000" ($Q_2=0, Q_1=0, Q_0=0$).

Con estos valores hallamos los valores que tendremos en las entradas T2, T1 y T0, y estas entradas harán cambiar el estado en el siguiente ciclo de reloj.



Haciendo esto para cada estado obtenemos la siguiente tabla:

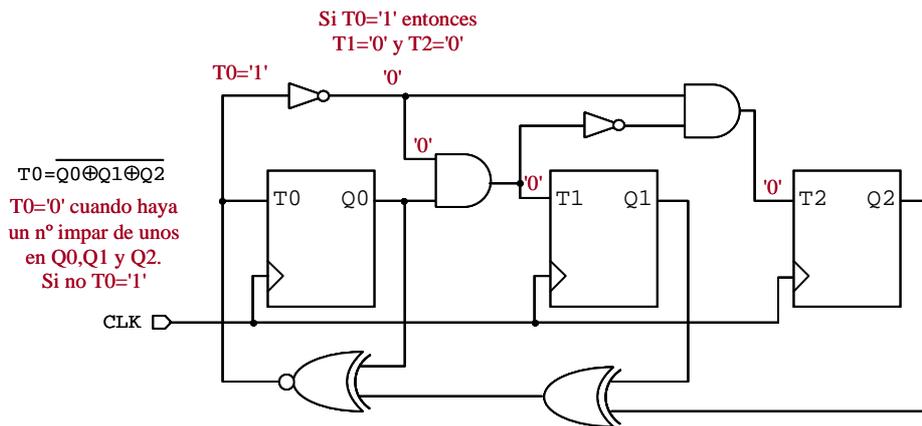
Estado Actual			Entradas para el siguiente estado		
Q2	Q1	Q0	T2	T1	T0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	1	0	0	1
0	1	0	1	0	0
1	1	0	0	0	1
1	1	1	0	1	0
1	0	1	0	0	1
1	0	0	1	0	0
0	0	0			

Vuelta al estado inicial

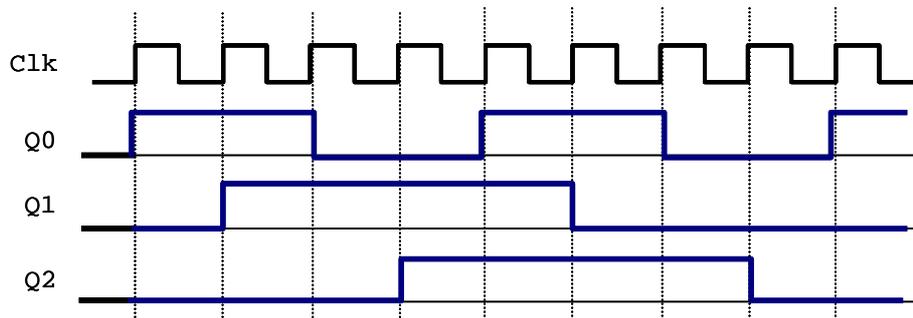
Dedicar un tiempo a analizar previamente el circuito puede ayudarnos a obtener la tabla, por ejemplo analizando la siguiente figura vemos que $T0=0$ cuando hay un número impar de unos en $Q2, Q1$ y $Q0$. Si hay un número par de unos $T0=1$.

Además, cuando $T0=1$, entonces $T1=0$ y $T2=0$.

Y de la misma manera, si $T1=1$ entonces $T2=0$. Por lo tanto, en cada ciclo de reloj sólo va a haber una T que sea '1', y por tanto, sólo va a cambiar de estado un biestable.



A partir de la secuencia se pueden dibujar las formas de onda:



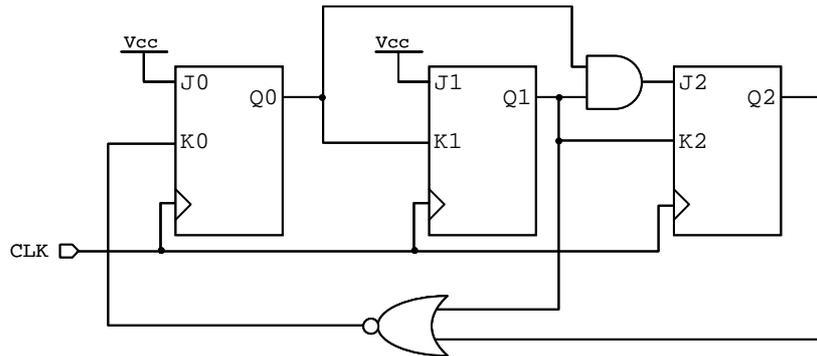
Secuencia de salida: 000-001-011-010-110-111-101-100

Esta secuencia coincide con el código Gray de 3 bits. Se puede apreciar más fácilmente desde la tabla. Y como hemos podido comprobar con el valor de las entradas T de los biestables, en cada ciclo de reloj sólo cambia un bit.

31. Análisis de circuitos 4

31.1. Enunciado

Para el siguiente contador suponiendo que el estado inicial es **001** ($Q_0=1$; $Q_1=Q_2=0$)
 Dibujar las formas de ondas durante los ciclos de reloj que sean necesarios para mostrar la secuencia completa.
 Indicar la secuencia de salida.



31.2. Solución

Tenemos biestables J-K, y puertas AND y NOR.

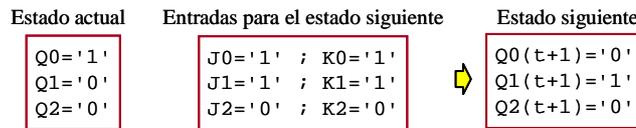
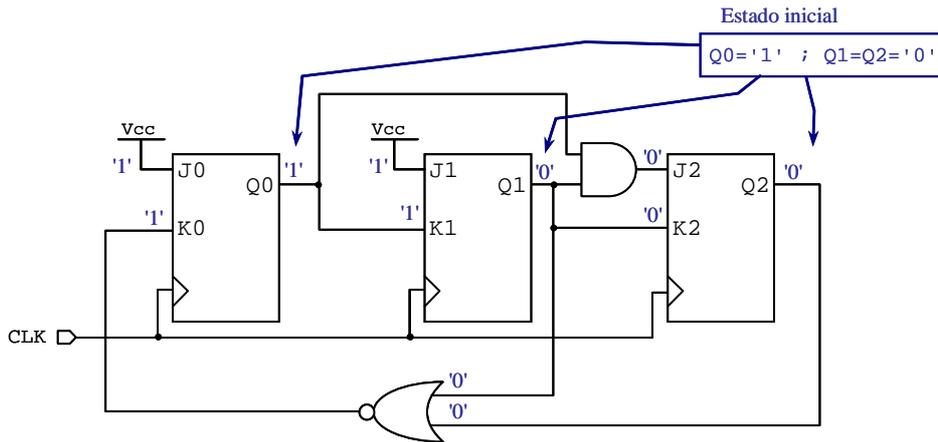
Ponemos sus tablas de verdad para evitar confundirnos durante el ejercicio.

Q_t	J	K	Q_{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Q_1	Q_0	J2
0	0	0
0	1	0
1	0	0
1	1	1

Q_2	Q_1	K0
0	0	1
0	1	0
1	0	0
1	1	0

Partimos de $Q_0=1$; $Q_1=Q_2=0$. Entonces las entradas para el siguiente ciclo de reloj serán.



Realizamos este mismo cálculo para cada estado nuevo que aparece y lo ponemos en una tabla.

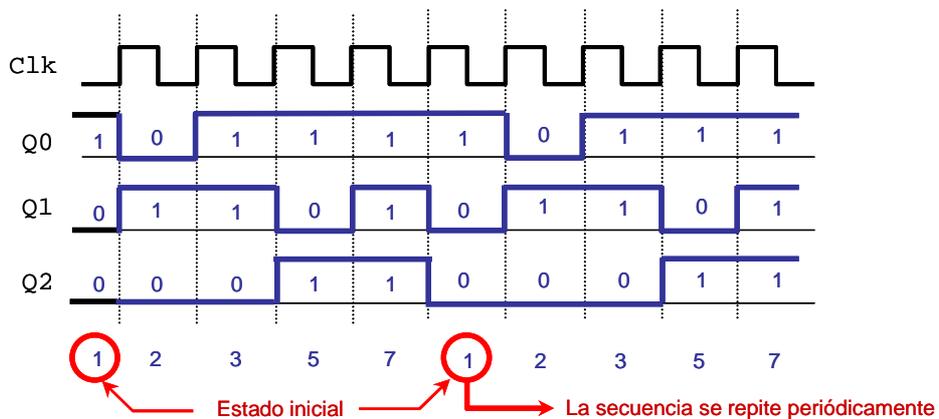
Estado Actual			Entradas para el siguiente estado					
Q2	Q1	Q0	J2	K2	J1	K1	J0	K0
0	0	1	0	0	1	1	1	1
0	1	0	0	1	1	0	1	0
0	1	1	1	1	1	1	1	0
1	0	1	0	0	1	1	1	0
1	1	1	1	1	1	1	1	0
0	0	1	0	0	1	1	1	0

El siguiente estado será 010

Vuelta al estado inicial

La secuencia de salida es 001, 010, 011, 101, 111, que son los números primos del 1 al 7, (incluyendo el 1): 1,2,3,5,7

El cronograma es:



32. Análisis de circuitos 5

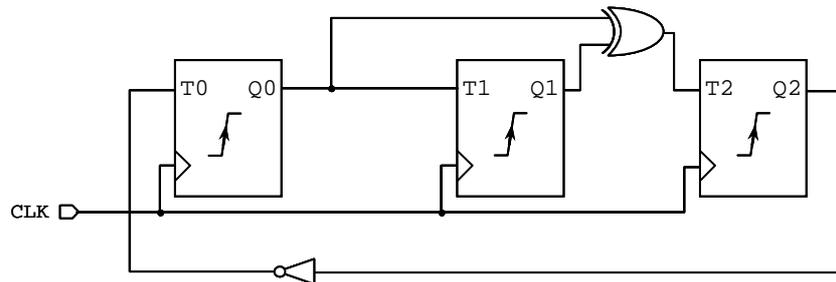
32.1. Enunciado

Para el siguiente circuito secuencial suponiendo que el estado inicial es **000** ($Q_0=Q_1=Q_2=0$)

Dibujar el cronograma durante los ciclos de reloj que sean necesarios para mostrar la secuencia completa. En el cronograma se deben incluir al menos las señales CLK, Q0, Q1 y Q2.

Indicar la secuencia de salida.

Nota: es muy importante indicar los pasos que indiquen cómo se ha obtenido la secuencia. Aunque la secuencia sea la correcta, no se contabilizarán los ejercicios que sólo dibujen las formas de onda sin explicar nada.



32.2. Solución

Tenemos biestables T y una puerta XOR.

Ponemos sus tablas de verdad para evitar confundirnos durante el ejercicio.

T		
T	Qt	Qt+1
0	0	0
0	1	1
1	0	1
1	1	0

XOR		
Q1	Q0	T2
0	0	0
0	1	1
1	0	1
1	1	0

Las ecuaciones de las entradas de los biestables son:

$$T_2 = Q_1 \oplus Q_0 = \overline{Q_1} \cdot Q_0 + Q_1 \cdot \overline{Q_0} \quad T_1 = Q_0 \quad T_0 = \overline{Q_2}$$

En el tiempo inicial ($t=0$), $Q_2=Q_1=Q_0=0$

En estas condiciones, la entrada T de cada biestable es

$t=0$ ($Q_2=Q_1=Q_0=0$)		
$T_2 = 0$	$T_1 = 0$	$T_0 = 1$

Así que sólo Q0 va a cambiar con el siguiente flanco de reloj, teniendo entonces para $t=1$:

$$Q_2=0; Q_1=0; Q_0=1$$

Este proceso se realiza igual para los siguientes ciclos de reloj hasta que se obtenga un estado por el que se ha pasado, a partir de entonces la secuencia empieza a ser periódica (en el problema 29 el proceso está más detallado).

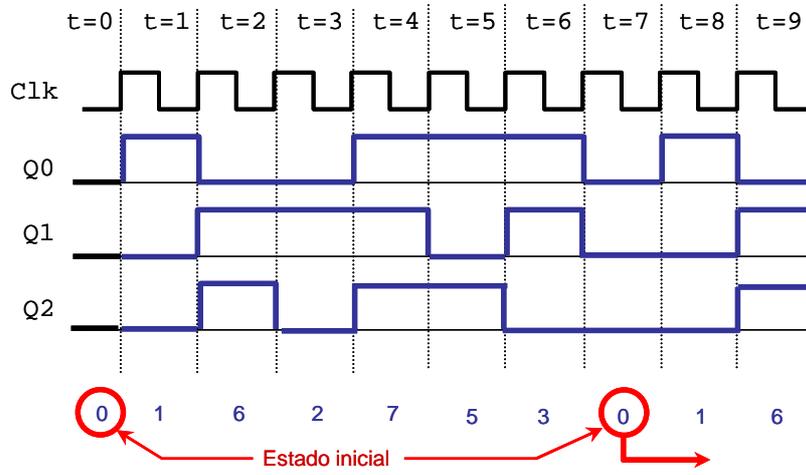
La tabla de transición de estado se muestra a continuación:

tiempo	Estado Actual			Entradas para el siguiente estado		
	Q2	Q1	Q0	T2	T1	T0
t=0	0	0	0	0	0	1
t=1	0	0	1	1	1	1
t=2	1	1	0	1	0	0
t=3	0	1	0	1	0	1
t=4	1	1	1	0	1	0
t=5	1	0	1	1	1	0
t=6	0	1	1	0	1	1
t=7	0	0	0	0	0	1

Vuelta al estado inicial (t=0)

Por tanto, la secuencia de salida es: 000, 001, 110, 010, 111, 101, 011, 000, que si se consideran números binarios es: 0, 1, 6, 2, 7, 5, 3, 0,

El cronograma se dibuja a continuación:



La secuencia se repite periódicamente