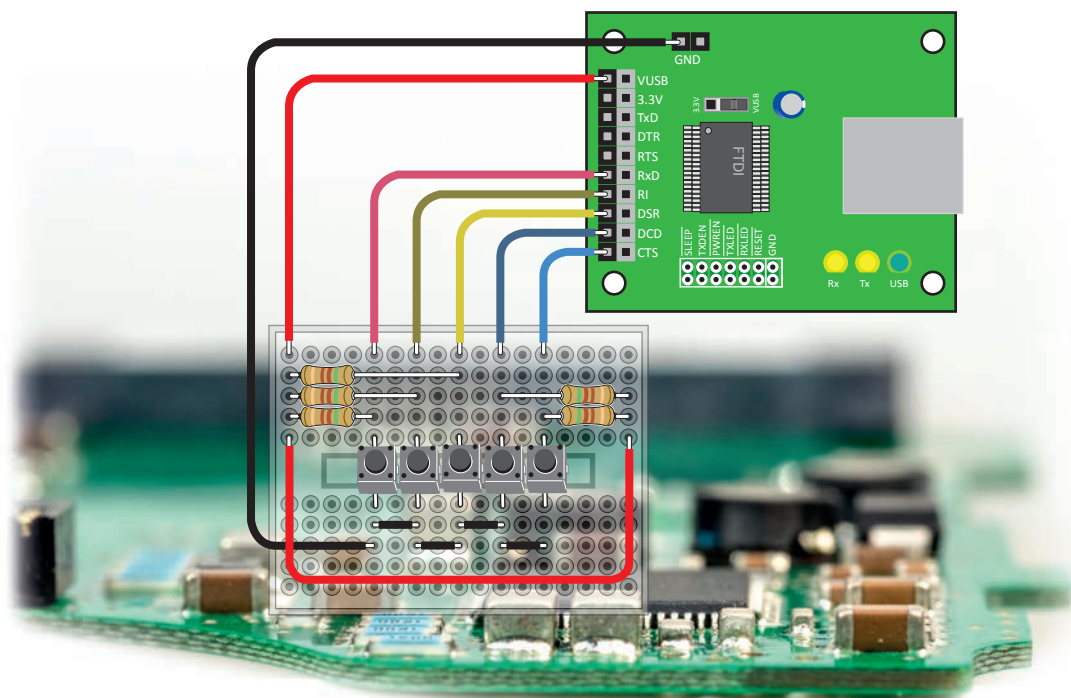


Circuitos Digitales

Problemas y ejercicios resueltos



Antonio Adán Oliver
Inocente Sánchez Ciudad
Blanca Quintana Galera

Circuitos digitales

Problemas y ejercicios resueltos

Circuitos digitales

Problemas y ejercicios resueltos

Antonio Adán Oliver
Inocente Sánchez Ciudad
Blanca Quintana Galera





Circuitos digitales. Problemas y ejercicios resueltos

© Antonio Adán Oliver, Inocente Sánchez Ciudad, Blanca Quintana Galera

© De la edición: Ra-Ma 2018

MARCAS COMERCIALES. Las designaciones utilizadas por las empresas para distinguir sus productos (hardware, software, sistemas operativos, etc.) suelen ser marcas registradas. RA-MA ha intentado a lo largo de este libro distinguir las marcas comerciales de los términos descriptivos, siguiendo el estilo que utiliza el fabricante, sin intención de infringir la marca y solo en beneficio del propietario de la misma. Los datos de los ejemplos y pantallas son ficticios a no ser que se especifique lo contrario.

RA-MA es marca comercial registrada.

Se ha puesto el máximo empeño en ofrecer al lector una información completa y precisa. Sin embargo, RA-MA Editorial no asume ninguna responsabilidad derivada de su uso ni tampoco de cualquier violación de patentes ni otros derechos de terceras partes que pudieran ocurrir. Esta publicación tiene por objeto proporcionar unos conocimientos precisos y acreditados sobre el tema tratado. Su venta no supone para el editor ninguna forma de asistencia legal, administrativa o de ningún otro tipo. En caso de precisarse asesoría legal u otra forma de ayuda experta, deben buscarse los servicios de un profesional competente.

Reservados todos los derechos de publicación en cualquier idioma.

Según lo dispuesto en el Código Penal vigente, ninguna parte de este libro puede ser reproducida, grabada en sistema de almacenamiento o transmitida en forma alguna ni por cualquier procedimiento, ya sea electrónico, mecánico, reprográfico, magnético o cualquier otro sin autorización previa y por escrito de RA-MA; su contenido está protegido por la ley vigente, que establece penas de prisión y/o multas a quienes, intencionadamente, reprodujeran o plagiaran, en todo o en parte, una obra literaria, artística o científica.

Editado por:

RA-MA Editorial

Calle Jarama, 3A, Polígono Industrial Igarsa

28860 PARACUELLOS DE JARAMA, Madrid

Teléfono: 91 658 42 80

Fax: 91 662 81 39

Correo electrónico: editorial@ra-ma.com

Internet: www.ra-ma.es y www.ra-ma.com

ISBN: 978-84-9964-761-6 **eybooks.com**

Depósito legal: M-28957-2018

Maquetación: Antonio García Tomé

Diseño de portada: Antonio García Tomé

Filmación e impresión: Safekat

Impreso en España en octubre de 2018

ÍNDICE

ACERCA DE LOS AUTORES	7
INTRODUCCIÓN	9
CAPÍTULO 1. REPRESENTACIÓN DE LA INFORMACIÓN	11
CONTENIDOS	11
EJERCICIOS RESUELTOS	12
TEST Y PREGUNTAS CORTAS	26
CAPÍTULO 2. FUNCIONES LÓGICAS	33
CONTENIDOS	33
EJERCICIOS RESUELTOS	34
TEST Y PREGUNTAS CORTAS	51
CAPÍTULO 3. SISTEMAS COMBINACIONALES: PUERTAS	55
CONTENIDOS	55
EJERCICIOS RESUELTOS	56
TEST Y PREGUNTAS CORTAS	88
CAPÍTULO 4. SISTEMAS COMBINACIONALES: MÓDULOS.....	93
CONTENIDOS	93
EJERCICIOS RESUELTOS	94
TEST Y PREGUNTAS CORTAS	178
CAPÍTULO 5. CIRCUITOS ARITMÉTICOS Y LÓGICOS.....	189
CONTENIDOS	189
EJERCICIOS RESUELTOS	190
TEST Y PREGUNTAS CORTAS	227

CAPÍTULO 6. SISTEMAS SECUENCIALES: BIESTABLES	231
CONTENIDOS	231
EJERCICIOS RESUELTOS	232
TEST Y PREGUNTAS CORTAS	291
CAPÍTULO 7. SISTEMAS SECUENCIALES: MÓDULOS	305
CONTENIDOS	305
EJERCICIOS RESUELTOS	306
TEST Y PREGUNTAS CORTAS	405
CAPÍTULO 8. SOLUCIONES A TEST Y PREGUNTAS CORTAS	413

ACERCA DE LOS AUTORES

Antonio Adán Oliver

Antonio Adán Oliver es Licenciado en Ciencias Físicas por la Universidad Complutense de Madrid (Astrofísica) y por la UNED (Física Industrial). Es Doctor Ingeniero Industrial con Premio Extraordinario de Doctorado. Se incorporó en 1990 a la Universidad de Castilla La Mancha, siendo actualmente Catedrático de Universidad en Ingeniería de Sistemas y Automática. Es creador y director del grupo de investigación 3D Visual Computing & Robotics de la UCLM (1998), habiendo ejercido como investigador principal de proyectos regionales, nacionales e internacionales.

Durante casi treinta como profesor, su experiencia docente en universidad abarca varias disciplinas relacionadas principalmente con sistemas digitales, computadores, visión por computador y robótica.

Su labor investigadora está centrada dentro del campo de Visión por Computador en los temas: Sistemas sensoriales y digitalización 3D, Creación automática de modelos semánticos 3D con escáneres láser de edificaciones (BIM), Reconocimiento 3D, Interacción de robots en escenas complejas, Sistemas de Visión 3D en Robótica Móvil y Creación de modelos térmicos 3D de edificios. Posee numerosas publicaciones en revistas indexadas (ISI) y congresos internacionales de reconocido prestigio (IEEE, IAPR, ACM), siendo poseedor de varios premios nacionales e internacionales.

Trabajó en el prestigioso Robotics Institute de la Carnegie Mellon University (Pittsburgh, PA, USA) durante un año (2009-2010) como profesor invitado y, posteriormente en el School of Energy, Geoscience, Infrastructure and Society, Heriot-Watt University (UK) en 2016. Ha sido revisor científico y miembro de varias sociedades científicas de carácter nacional e internacional.

Inocente Sánchez Ciudad

Inocente Sánchez Ciudad obtuvo el título de Ingeniero de Telecomunicación a los 23 años por la Universidad Politécnica de Madrid. Realizó los cursos de doctorado en la UNED.

Trabajó como Ingeniero de Desarrollo desde marzo de 1987 hasta enero de 1992 en la empresa TECNOBIT S.A., en su centro de Valdepeñas (Ciudad Real).

Desde marzo de 1992 hasta la actualidad es profesor de la Universidad de Castilla La Mancha en el área de conocimiento de Arquitectura y Tecnología de Computadores, en la Escuela Superior de Informática, antes llamada Escuela Universitaria de Informática, donde desarrolla su actividad docente impartiendo las asignaturas Tecnología de Computadores, Estructura de Computadores y Redes, de primer curso, y las asignaturas optativas, de los planes de estudios ya extinguidos, de Procesamiento de la Señal, Comunicación de Datos y Tecnología del Habla.

Ha desempeñado cargos académicos en el ámbito universitario: Secretario de Centro en la antigua Escuela Universitaria de Informática y Secretario de Departamento en el Departamento de Informática de la UCLM.

Pertenece grupo de investigación MAMI (Modelling Ambient Intelligence) de la UCLM, dirigido por el Dr. José Bravo.

Blanca Quintana Galera

Blanca Quintana es Ingeniera Industrial por la Universidad de Castilla-La Mancha y alumna de doctorado en Ciencias y Tecnologías Aplicadas a la Ingeniería Industrial. Es miembro del grupo de investigación 3D Visual Computing & Robotics (3DVC&R) de la UCLM. Ha sido investigadora visitante en la School of Energy, Geoscience, Infrastructure and Society de la Universidad Heriot-Watt (UK) durante 2018. Sus temas de investigación incluyen: sensores 3D, procesamiento de información 3D, escaneado automático 3D, y generación automática de modelos BIM con escáneres. En relación a estos temas, ha publicado varios artículos en revistas indexadas y en conferencias nacionales e internacionales.

INTRODUCCIÓN

El libro que tiene entre sus manos es el fruto del trabajo de recopilación de ejercicios planteados en los exámenes de la Escuela Superior de Informática de Ciudad Real de la Universidad de Castilla La Mancha (UCLM), en los últimos 28 años, en asignaturas relacionadas con el diseño de sistemas y circuitos digitales. El contenido de este documento correspondería a asignaturas con distintos nombres en otras facultades y escuelas de ingeniería, tales como: Circuitos Digitales, Diseño Digital, Tecnología de Computadores, Diseño Lógico, etc. En definitiva, se trata de una primera asignatura que introduce al alumno en el mundo de los circuitos digitales como base fundamental del hardware de los computadores.

La casi totalidad de los problemas y ejercicios del libro son originales y han sido propuestos en exámenes por los profesores Antonio Adán Oliver (Catedrático de Universidad y Doctor Ingeniero Industrial) e Inocente Sánchez Ciudad (Titular de Escuela Universitaria e Ingeniero de Telecomunicación). Además, se ha contado con la valiosa colaboración de Blanca Quintana Galera (Ingeniero Industrial), quien ha realizado trabajos de revisión y preparación del documento.

El libro consta de problemas y ejercicios de test que cubren gran parte de la temática de este tipo de asignatura. Muchos de los problemas plantean situaciones reales (a veces con un detallado enunciado) en las que se pide la definición e implementación del circuito digital que resuelve dicho problema. El libro está estructurado en siete capítulos que cubren los aspectos de: representación de la información y funciones lógicas, diseño lógico de circuitos combinacionales con puertas lógicas y con módulos combinaciones, y análisis y diseño de circuitos secuenciales basados en autómatas de estados finitos. Por supuesto, se supone conocida la teoría necesaria para realizar estos ejercicios. Dichos ejercicios, junto con las cuestiones tipo test planteadas, y cuya solución también se facilita, suponen

una importante ayuda para afianzar los conocimientos adquiridos por los alumnos en las clases de teoría.

Esperamos que el contenido de este libro resulte de utilidad para los alumnos de primer curso de titulaciones de ingeniería.

LOS AUTORES
Ciudad Real, julio de 2018

1

REPRESENTACIÓN DE LA INFORMACIÓN

CONTENIDOS:

- ▀ Sistemas de numeración y cambios de base.
- ▀ Representación de números enteros.

EJERCICIOS RESUELTOS

1) Escriba el número decimal 250.5 en las bases 3, 4, 7 y 16.

Haremos con detalle el cambio a base 3. Separando las partes entera y decimal tenemos:

Dividendo	Cociente		Resto	
250	83		1	
83	27		2	
27	9		0	
9	3		0	
3	1	→ Último cociente	0	→ Último resto

Parte decimal

$$0.5 \cdot 3 = 1.5 \rightarrow a_{-1}=1$$

$$0.5 \cdot 3 = 1.5 \rightarrow a_{-2}=1$$

.....

$$0.5 \cdot 3 = 1.5 \rightarrow a_{-k}=1$$

Luego $250.5_{(10)} = 100021.11\dots 11_{(3)}$ periódico puro 1

Siguiendo el mismo procedimiento para los demás casos se obtiene:

$$250.5_{(10)} = 3322.20000_{(4)} \text{ sólo hay una cifra decimal no nula (exacto)}$$

$$250.5_{(10)} = 505.333_{(7)} \text{ periódico puro 3}$$

$$250.5_{(10)} = FA.80000_{(16)} \text{ sólo hay una cifra decimal no nula (exacto)}$$

NOTA: el hecho de que un número en base decimal tenga un número limitado, incluso pequeño, de cifras decimales no nulas no implica que su representación en otra base deba tener un número finito de cifras no nulas. Puede darse el caso de que tenga infinitas cifras (periódico puro).

2) Convierta los siguientes números decimales a binarios: 12.0625, 10^4 , 673.23 y 1998.

Igual que en el ejercicio anterior separamos las partes entera y decimal y procedemos de la misma forma.

Dividendo	Cociente		Resto	
12	6		0	
6	3		0	
3	1	→ Último cociente	1	→ Último resto

Dividendo	cociente	resto	
12	6	0	
6	3	0	
3	1	1	→ Último cociente 1

Parte decimal

$$0.0625 \cdot 2 = 0.125 \rightarrow a_1 = 0$$

$$0.125 \cdot 2 = 0.25 \rightarrow a_2 = 0$$

$$0.25 \cdot 2 = 0.5 \rightarrow a_3 = 0$$

$$0.5 \cdot 2 = 1.0 \rightarrow a_4 = 1. \text{ Nótese que } 0.0625 \text{ es } 1/16 = 2^{-4}$$

$$0 \cdot 2 = 0 \rightarrow \text{resto } a_k = 0$$

$$\text{Por consiguiente } 12.0625_{(10)} = 1100.00010000_{(2)}$$

Siguiendo el mismo procedimiento para los demás se obtiene:

$$10^4_{(10)} = 10011100010000_{(2)}$$

$$673.23_{(10)} = 1010100001.00111_{(2)}$$

$$1998_{(10)} = 11111001110_{(2)}$$

También se podría haber hecho en primer lugar el cambio a hexadecimal y después sustituir cada dígito hexadecimal por la cuaterna correspondiente, quedando

$$10^4_{(10)} = 2710_{(16)} = 0010 \ 0111 \ 0001 \ 0000_{(2)}$$

$$673.23_{(10)} = 2A1.3A_{(16)} = 0010 \ 1010 \ 0001 \ . \ 0011 \ 1010_{(2)}$$

$$1998_{(10)} = 7CE_{(16)} = 0111 \ 1100 \ 1110_{(2)}$$

O pasando previamente por octal, en cuyo caso cada dígito octal se sustituye por una terna

$$10^4_{(10)} = 23420_{(8)} = 010 \ 011 \ 100 \ 010 \ 000_{(2)}$$

$$673.23_{(10)} = 1241.16_{(8)} = 001 \ 010 \ 100 \ 001 \ . \ 001 \ 110_{(2)}$$

$$1998_{(10)} = 3716_{(8)} = 011 \ 111 \ 001 \ 110_{(2)}$$

3) Sumar los números en la base dada, sin convertirlos previamente a números decimales.

- a) 1230 y 23 en base 4.
 b) 135.4 y 43.2 en base 6.
 c) 367 y 715 en base 8.
 d) 296 y 57 en base 12.

a) Suma $1230_{(4)} + 23_{(4)} = 1313_{(4)}$

$$\begin{array}{r} 1230_{(4)} \\ +23_{(4)} \\ \hline 1313_{(4)} \end{array}$$

b) Suma $135.4_{(6)} + 43.2_{(6)} = 223.0_{(6)}$

$$\begin{array}{r} 135.4_{(6)} \\ +43.2_{(6)} \\ \hline 223.0_{(6)} \end{array}$$

c) Suma $367_{(8)} + 715_{(8)} = 1304_{(8)}$

$$\begin{array}{r} 367_{(8)} \\ +715_{(8)} \\ \hline 1304_{(8)} \end{array}$$

d) Suma $296_{(12)} + 57_{(12)} = 331_{(12)}$

$$\begin{array}{r} 296_{(12)} \\ +57_{(12)} \\ \hline 331_{(12)} \end{array}$$

4) Convierta los números a las bases que se indican:

- a) 225.225₍₁₀₎ a binario, octal y hexadecimal.
 b) 11010111.110₍₂₎ a decimal, octal y hexadecimal.
 c) 623.77₍₈₎ a binario, decimal y hexadecimal.

a) 225.225₍₁₀₎ = 11100001 . 0011100110011001100...1100...₍₂₎

Tomando grupos de 3 bits a derecha e izquierda del punto decimal obtenemos 011 100 001.001 110 011 por lo que $225.225_{(10)} = 341.163163...163..._{(8)}$

Tomando grupos de 4 bits a derecha e izquierda del punto decimal obtenemos 1110 0001.0011 1001 1001 1001 ... por lo que $225.225_{(10)} = E1.39999...9..._{(16)}$

b) De forma inversa $11010111.110_{(2)} = 1+2+4+16+64+128+1/2+1/4 = 215.75_{(10)}$

Tomando agrupaciones de tres y cuatro dígitos binarios se obtiene:

$$11010111.110_{(2)} = 327.6_{(8)}$$

$$11010111.1100_{(2)} = D7.C_{(16)}$$

$$c) 623.77_{(8)} = 3 \cdot 1 + 2 \cdot 8 + 6 \cdot 64 \cdot 7/8 + 7/64 = 403.984375_{(10)}$$

Expresando cada dígito octal por tres bits se obtiene su representación binaria.

$$623.77_{(8)} = 110010011.111111_{(2)}$$

Tomando agrupaciones de cuatro bits en la representación binaria se obtiene su representación en hexadecimal.

$$623.77_{(8)} = 0001 1001 0011.1111 1100_{(2)} = 193.FC_{(16)}$$

5) Expresar el número decimal 2223.39 en las bases binarias, base 6, octal y hexadecimal.

Base 2

$$1000 1010 1111.1011 0001 1110 1011 1_{(2)}$$

Base 6

Parte entera	2223	6				
	42	370	6			
	03	10	61	6		
	3	4	1	10	6	
				4	1	

Por lo que obtenemos: 14143

Parte decimal

$$0.39 \times 6 = 2.34 \rightarrow 2$$

$$0.34 \times 6 = 2.04 \rightarrow 2$$

$$0.04 \times 6 = 0.24 \rightarrow 0$$

$$0.24 \times 6 = 1.44 \rightarrow 1$$

$$0.44 \times 6 = 2.64 \rightarrow 2$$

$$0.64 \times 6 = 3.84 \rightarrow 3$$

$$0.84 \times 6 = 5.04 \rightarrow 5$$

Finalmente se tiene: $2223.39_{(10)} = 14143.2201235_{(6)}$

Base 8

Parte entera	2223	8	
	62	277	8
	63	37	34
	7	5	2
			4

Por lo que tenemos: 4257

Parte decimal

$$0.39 \times 8 = 3.12 \rightarrow 3$$

$$0.12 \times 8 = 0.96 \rightarrow 0$$

$$0.96 \times 8 = 7.68 \rightarrow 7$$

$$0.68 \times 8 = 5.44 \rightarrow 5$$

$$0.44 \times 8 = 3.52 \rightarrow 3$$

Luego, $2223.39_{(10)} = 4257.30753_{(8)}$

Base 16

Parte entera	2223	16	
	62	138	16
	143	10	8
	15		

Por lo que tenemos: 8AF

Parte decimal

$$0.39 \times 16 = 6.24 \rightarrow 6$$

$$0.24 \times 16 = 3.84 \rightarrow 3$$

$$0.84 \times 16 = 13.44 \rightarrow D$$

$$0.44 \times 16 = 7.04 \rightarrow 7$$

Luego, $2223.39_{(10)} = 8AF.63D7_{(16)}$

- 6) El número 543_x se corresponde con el número 674 en base octal. ¿De qué base se trata x?

Inicialmente puede deducirse que la base buscada es mayor o igual a 6. En base n , siendo $n \leq 6$ se emplean los dígitos entre 0 y $n-1$.

Después se convierte 674_8 a base decimal, siendo igual a: $6 \cdot 64 + 7 \cdot 8 + 4 = 444_{(10)}$

Finalmente, se plantea la ecuación de segundo orden $5x^2 + 4x + 3 = 444$

Resolviendo, se obtiene que $x = 9$ y $x = -9.8$, descartando esta última solución por ser negativa. Por tanto, se puede afirmar que 543 estaba en base 9.

7) Convertir los siguientes números a base hexadecimal.

a) $10000.1_{(2)}$

b) $255.875_{(10)}$

c) $675.03_{(8)}$

d) $345.3_{(5)}$

a) $10000.1_{(2)} = 10.8_{(16)}$

0001 0000 .1000

1 0 8 $\rightarrow$ 10.8

b) $255.875_{(10)} = FF.E_{(16)}$

255 |16

15 15

$\rightarrow$ FF

$0.875 \times 16 = 14.00 \rightarrow E$

c) $675.03_{(8)} = 1BD.0C_{(16)}$

Pasamos el número octal a binario, 0001 1011 1101 . 0000 1100 y de binario a hexadecimal, quedando:

0001 $\rightarrow$ 1

1011 $\rightarrow$ B

1101 $\rightarrow$ D

0000 $\rightarrow$ 0

1100 $\rightarrow$ C

d) $345.3_{(5)} = ?$

No es posible realizar esta operación ya que en el código numérico hay dígitos iguales a la base. Con una base $n = 5$, los dígitos deben estar en el rango $[0,4]$.

8) Convertir los siguientes números a sus binarios equivalentes.

- a) $7.5_{(8)}$
 b) $475_{(10)}$
 c) $475_{(16)}$
 d) $555_{(5)}$

a) $7.5_{(8)} = 111.101_{(2)}$

b) $475_{(10)} = 111011011_{(2)}$

$$\begin{array}{r}
 475 \quad | \quad 2 \quad \underline{\hspace{1cm}} \\
 07 \quad 237 \quad | \quad 2 \quad \underline{\hspace{1cm}} \\
 15 \quad 03 \quad 118 \quad | \quad 2 \quad \underline{\hspace{1cm}} \\
 1 \quad 17 \quad 18 \quad 59 \quad | \quad 2 \quad \underline{\hspace{1cm}} \\
 \quad \quad 1 \quad 0 \quad 19 \quad 29 \quad | \quad 2 \quad \underline{\hspace{1cm}} \\
 \quad \quad \quad \quad 1 \quad 1 \quad 14 \quad | \quad 2 \quad \underline{\hspace{1cm}} \\
 \quad \quad \quad \quad \quad 0 \quad 7 \quad | \quad 2 \quad \underline{\hspace{1cm}} \\
 \quad \quad \quad \quad \quad \quad 1 \quad 3 \quad | \quad 2 \quad \underline{\hspace{1cm}} \\
 \quad \quad \quad \quad \quad \quad \quad 1 \quad 1 \quad | \quad 2 \quad \underline{\hspace{1cm}}
 \end{array}$$

c) $475_{(16)} = 100\ 0111\ 0101_{(2)}$

$4_{(10)} \rightarrow 0100_{(2)}$

$7_{(10)} \rightarrow 0111_{(2)}$

$5_{(10)} \rightarrow 0101_{(2)}$

d) $555_{(5)} = ?$

No es posible realizar esta operación ya que en el código numérico hay dígitos iguales a la base. Con una base $n = 5$, los dígitos deben estar en el rango $[0,4]$.

- 9) Expresa el número $2778_{(10)}$ en base hexadecimal. ¿Te sugiere algo el resultado?
 Comprueba tu respuesta con el código QR.



$$\begin{array}{r}
 126_{(10)} = 1111110_{(2)} \quad 0000\ 0111\ 1110_{(2)} \\
 589_{(10)} = 1001001101_{(2)} \quad +\ 0010\ 0100\ 1101_{(2)} \\
 \hline
 \quad \quad \quad 0010\ 1100\ 1011_{(2)}
 \end{array}$$

Comprobación: $126+589 = 715$. Por otro lado $512+128+64+8+2+1 = 715$

Base 7

$$\begin{array}{r}
 126 \quad |7 \\
 56 \quad 18 \quad |7 \\
 0 \quad 4 \quad 2 \\
 \hline
 589 \quad |7 \\
 29 \quad 84 \quad |7 \\
 1 \quad 0 \quad 12 \quad |7 \\
 \quad \quad \quad 5 \quad 1
 \end{array}$$

$$\begin{array}{r}
 126_{(10)} = 240_{(7)} \quad 0240_{(7)} \\
 589_{(10)} = 1501_{(7)} \quad +\ 1501_{(7)} \\
 \hline
 \quad \quad \quad 2041_{(7)}
 \end{array}$$

Comprobación: $126+589 = 715$. Por otro lado $2*7^3+4*7+1 = 715$

Base 8

$$\begin{array}{r}
 126 \quad |8 \\
 46 \quad 15 \quad |8 \\
 6 \quad 7 \quad 1 \\
 \hline
 589 \quad |8 \\
 29 \quad 73 \quad |8 \\
 5 \quad 1 \quad 9 \quad |8 \\
 \quad \quad \quad 1 \quad 1
 \end{array}$$

$$\begin{array}{r}
 126_{(10)} = 176_{(8)} \quad 176_{(8)} \\
 589_{(10)} = 1115_{(8)} \quad +\ 1115_{(8)} \\
 \hline
 \quad \quad \quad 1313_{(8)}
 \end{array}$$

Comprobación: $126+589 = 715$. Por otro lado $1*8^3+3*64+8+3 = 715$

Base 16

$$\begin{array}{r}
 126 \quad |16 \\
 14 \quad 7 \\
 \hline
 589 \quad |16 \\
 109 \quad 36 \quad |16 \\
 13 \quad 4 \quad 2
 \end{array}$$

$$\begin{array}{r}
 126_{(10)} = 7D_{(16)} \quad 07D_{(16)} \\
 589_{(10)} = 24D_{(16)} \quad +\ 24D_{(16)} \\
 \hline
 \quad \quad \quad 2CB_{(16)}
 \end{array}$$

Comprobación: $126+589 = 715$. Por otro lado $2*16^2+12*16+11 = 715$

11) Realiza las siguientes operaciones con complemento a 1.

- a) $47 + 18$
 b) $-24 - 26$
 c) $344 + 144$
 d) $244 - 25$

a) $47 + 18$ $47_{(10)} = 010\ 1111_{(C1)}$ 7 bits $18_{(10)} = 01\ 0010_{(C1)}$ 6 bits

$$\begin{array}{r}
 010\ 1111 \\
 001\ 0010 \\
 \hline
 0\ 100\ 0001 \\
 \hline
 100\ 0001
 \end{array}$$

El rango en C1 con 7 bits es $2^6 - 1 = 63$ Como se aprecia hay un cambio de signo $\rightarrow$ overflow

$$\begin{array}{r}
 0010\ 1111 \\
 0001\ 0010 \\
 \hline
 0\ 0100\ 0001 \\
 \hline
 0100\ 0001
 \end{array}$$

Como $47 + 18 = 65 \rightarrow$ añadimos un bit de signo 0b) $-24 - 26$ $24 \rightarrow 01\ 1000$ 6 bits $-24 \rightarrow 10\ 0111$ $26 \rightarrow 01\ 1010$ 6 bits $-26 \rightarrow 10\ 0101$

$$\begin{array}{r}
 10\ 0111 \\
 10\ 0101 \\
 \hline
 1\ 00\ 1100 \\
 \hline
 00\ 1101
 \end{array}$$

El rango en C1 con 6 bits es $2^5 - 1 = 31$ Como se aprecia hay un cambio de signo $\rightarrow$ overflow

$$\begin{array}{r}
 110\ 0111 \\
 110\ 0101 \\
 \hline
 1\ 100\ 1100 \\
 \hline
 100\ 1101
 \end{array}$$

Como $-24 - 26 = -50 \rightarrow$ añadimos un bit de signo 1Comprobación: $1001101 (-50) \rightarrow 0110010 (+50)$

c) $344 + 144$

$344 \rightarrow 01\ 0101\ 1000$ 10 bits

$134 \rightarrow 0\ 1000\ 0110$ 9 bits

Con 10 bits el rango es $2^9 - 1 = 511$

$344 + 134 = 478 \rightarrow$ luego no existirá overflow con 10 bits

$$\begin{array}{r}
 01\ 0101\ 1000 \\
 00\ 1000\ 0110 \\
 \hline
 01\ 1101\ 1110 \\
 \hline
 01\ 1101\ 1110
 \end{array}$$

Comprobación: $0111011110 \rightarrow 2+4+8+16+64+128+256 = 478$

d) $241-25$

$241 \rightarrow 0\ 1111\ 0001$

$-25 \rightarrow 10\ 0110$, ya que $+25_{(C1)} = 011001$

$$\begin{array}{r}
 0\ 1111\ 0001 \\
 1\ 1110\ 0110 \\
 \hline
 1\ 0\ 1101\ 0111 \\
 \hline
 0\ 1101\ 1000
 \end{array}$$

Comprobación: $011011000 \rightarrow 8+16+64+128 = 216$

12) Realiza la suma $A+B$ en Ca1 con 8 bits, siendo $A=100$, $B= -23$.

$+23_{(Ca1)} = 00010111$

$$\begin{array}{rcl}
 +100_{(10)} = & 01100100_{(Ca1)} & \\
 -23_{(10)} = & 11101000_{(Ca1)} & \\
 \hline
 & 101001100 & \\
 & +1 & \\
 & 01001101 &
 \end{array}$$

Comprobación: $01001101 \rightarrow 1+4+8+64 = 77$

13) Comprueba si se pueden realizar, sin desbordamiento, las siguientes operaciones con el número de bits que se indican, trabajando en complemento a 2. En los casos en los que sí se pueda, realizar la operación y comprobar el resultado.

- a) $157 + 222$ con 7 bits
- b) $-245 - 112$ con 10 bits
- c) $344 + 134$ con 10 bits
- d) $344 - 220$ con 8 bits
- e) $344 - 569$ con 6 bits
- f) $350 - 533$ con 8 bits

Rango de representaciones:

$$6 \text{ bits} = [-2^5, 2^5 - 1] = [-32, 31]$$

$$7 \text{ bits} = [-2^6, 2^6 - 1] = [-64, 63]$$

$$8 \text{ bits} = [-2^7, 2^7 - 1] = [-128, 127]$$

$$10 \text{ bits} = [-2^9, 2^9 - 1] = [-512, 511]$$

- a) $157 + 222$ con 7 bits

No se puede realizar, ya que sumamos fuera de rango

- b) $-245 - 112$ con 10 bits

Sí se puede realizar, porque tanto operandos como resultado están dentro de rango

$$+245_{(10)} = 1111\ 0101_{(C2)} \rightarrow 00\ 1111\ 0101 \text{ usando 10 bits}$$

$$-245_{(10)} = 11\ 0000\ 1011_{(C2)}$$

$$+112_{(10)} = 111\ 0000_{(C2)} \rightarrow 00\ 0111\ 0000 \text{ usando 10 bits}$$

$$-112_{(10)} = 11\ 1001\ 0000_{(C2)}$$

$$\begin{array}{r} 11\ 0000\ 1011 \\ 11\ 1001\ 0000 \\ \hline 110\ 1001\ 1011 \end{array}$$

Comprobación: $11010011011 \rightarrow -357$, ya que

$$00101100101 \rightarrow 1+4+32+64+256 = 357$$

- c) $344 + 134$ con 10 bits

Sí se puede realizar, ya que tanto operandos como resultado están dentro de rango

$$344_{(10)} = 01\ 0101\ 1000_{(C2)}$$

$$134_{(10)} = 00\ 1000\ 0110_{(C2)}$$

$$\begin{array}{r} 01\ 0101\ 1000 \\ 00\ 1000\ 0110 \\ \hline 01\ 1101\ 1110 \end{array}$$

Comprobación: $0111011110 \rightarrow 2+4+8+16+64+128+256 = 478$

d) $344 - 220$ con 8 bits

No se puede realizar, está fuera de rango alguno de los operandos, aunque no el resultado.

e) $344 - 569$ con 6 bits

No se puede realizar, ya que están fuera de rango todos los operandos e incluso el resultado.

f) $350 - 533$ con 8 bits

No se puede realizar, porque están fuera de rango todos los operandos y el resultado.

14) Realiza los siguientes cambios de base, poniendo en todos los casos dos números decimales.

a) $431.23_{(5)}$ a base 10

b) $1254.25_{(10)}$ a base 8

c) $1011100.111_{(2)}$ a base 16

d) $1011100.111_{(2)}$ a base 4

a) $431.23_{(5)} = 116.52_{(10)}$

b) $1254.25_{(10)} = 2346.20_{(8)}$

c) $1011100.111_{(2)} = 5C.E0_{(16)}$

d) $1011100.111_{(2)} = 1130.32_{(4)}$

15) Realiza las siguientes operaciones en complemento a 2, teniendo en cuenta que en todas ellas se trabaja con 4 bits. ¿Hay desbordamiento (overflow) en algún caso? Indícalo.

$-7 - 1$, $7 + 5$, $1 - 4$

$-7 - 1$	$7 + 5$	$1 - 4$
1001	0111	0001
<u>1111</u>	<u>0101</u>	<u>1100</u>
1000	1100	1101

Overflow

16) Efectuar los siguientes cambios de base:

- a) 10,357 a binario con 11 cifras decimales.**
- b) 110001,001011₍₂₎ a hexadecimal sin pasar por base 10.**
- c) 59026,F5₍₁₆₎ a decimal con todas las cifras decimales.**
- d) 77517,15 a octal con 5 cifras decimales.**

- a) 10,357 a binario con 11 cifras decimales $\rightarrow 1010,01011011011_{(2)}$
- b) 110001,001011₍₂₎ a hexadecimal sin pasar por base 10 $\rightarrow 31,2C_{(16)}$
- c) 59026,F5₍₁₆₎ a decimal con todas las cifras decimales $\rightarrow 364582,95703125$
- d) 77517,15 a octal con 5 cifras decimales $\rightarrow 227315,11463_{(8)}$

TEST Y PREGUNTAS CORTAS

1) En Signo-Magnitud, el rango de representación con 8 bits es:

- ☐ A El mismo que en complemento a 1, es decir, de -255 a +255 ambos incluidos.
- ☐ B El mismo que en complemento a 2, es decir, de -128 a +127 ambos incluidos.
- ☐ C Las dos anteriores son falsas.

2) ¿Qué representa el código binario 00111010?

- ☐ A El número entero +58 en complemento a 1.
- ☐ B El número entero +59 en complemento a 2.
- ☐ C Las dos anteriores son ciertas.

3) De las siguientes afirmaciones, di cuáles son verdaderas y cuáles falsas:

- ☒ V ☐ F 86.77₍₁₀₎ en octal es 55.A
- ☒ V ☐ F El número 10110101.11₍₂₎ en hexadecimal es B5.3
- ☒ V ☐ F El código binario 110001 es la representación binaria del número natural 19 con 6 bits.

4) Di qué afirmaciones son verdaderas y cuáles falsas:

- ☒ V ☐ F El número -106₈ representación *complemento a 2* con 8 bits es 10111010
- ☒ V ☐ F En representación de *complemento a 1* de 9 bits, el rango es [-512, 512]
- ☒ V ☐ F En representación *signo-magnitud* con n bits el rango es [0, 2⁽ⁿ⁻¹⁾]
- ☒ V ☐ F En representación de suma/resta en *complemento a 1* no es necesario recircular el acarreo posterior.

5) Di qué afirmaciones son verdaderas y cuáles falsas

- ☒ V ☐ F El número 106₈ representación *complemento a 2* con 8 bits es 01000110
- ☒ V ☐ F El número 10110101_(C1) en representación *complemento a 2* con 8 bits es 10110110
- ☒ V ☐ F En representación *signo-magnitud* de 10 bits el rango es [-1024, 1024]

6) La representación en *complemento a 2* de un número binario es:

- ☐ A Una forma de representar los números negativos exclusivamente.
- ☐ B Una forma de representar los números enteros con signo, incluyendo al 0.
- ☐ C Igual a la representación en *complemento a 1* cambiando el bit de signo.

7) La representación en *complemento a 1* con 6 bits abarca el intervalo (ambos incluidos):

- ☐ A De -32 a +31
- ☐ B De -31 a +31

☐ De -64 a +63

8) ¿Cuál es el mínimo número de bits que tendríamos que usar para poder trabajar en complemento a 2 sumando y restando números enteros comprendidos entre : -999 y +999?

☐ A 11 bits

☐ B 12 bits

☐ C 13 bits

9) Selecciona V o F:

☐ V ☐ F El número -32 es representable en C a 2 con 7 bits.

☐ V ☐ F El complemento a 2 del complemento a 2 de un número binario es el propio número.

☐ V ☐ F La representación sin signo codifica más números enteros que la representación signo-magnitud con el mismo número de bits.

10) La expresión de $56,1_{(7)}$ en decimal y binario es:

☐ A $56,1_{(7)} = 51,14_{(10)}$ y $56,1_{(7)} = 1010001,0001_{(2)}$

☐ B $56,1_{(7)} = 41,14_{(10)}$ y $56,1_{(7)} = 101001,001_{(2)}$

☐ C Las dos anteriores respuestas son falsas.

11) El valor del número 10π , siendo $\pi = 3.14159265$, en hexadecimal es:

☐ A 1F,6107102

☐ B 31,6A7A2

☐ C 1F,6A7A2

12) Se pretende representar digitalmente la información de un termómetro que mide la temperatura desde -20°C hasta $+50^{\circ}\text{C}$, ambas temperaturas incluidas, con resolución de $0,1^{\circ}\text{C}$. ¿Cuántos bits se necesitan como mínimo representando la temperatura (en décimas de grados) en complemento a 2? ¿Cuáles son los códigos binarios no usados en dicho caso?

☐ A 10 bits. Códigos no usados: de 1000000000 a 1100111000 y de 0111110100 a 0111111111.

☐ B 10 bits. Códigos no usados: de 1000000000 a 1100110111 y de 0111110101 a 0111111111.

☐ C 7 bits. Códigos no usados: de 1000000 a 1101011 y de 0110011 a 0111111.

13) El resultado de la suma de los números en BCD 00100101 y 00100111 expresado en binario natural es:

☐ A 01010010.

☐ B 00110100.

☐ C Ninguna de las anteriores.

- 14) El resultado en complemento a 2 de la operación (00011)-(10100), con ambos operandos expresados también en complemento a 2, es:

- ☐ A 11111
☐ B 01111
☐ C 10111

- 15) Indique el resultado en BCD de la siguiente operación de dos números codificados en complemento a 2: 00100101 + 11111100.

- ☐ A 11011110
☐ B 00110011
☐ C 00100001

- 16) La representación de un número entero positivo utilizando n bits:

- ☐ A Coincide en Signo Magnitud y en Complemento a 2, pero no en Complemento a 1.
☐ B Coincide en Complemento a 1, en Complemento a 2 y en Signo Magnitud.
☐ C Coincide en Signo Magnitud y en Complemento a 1, pero no en Complemento a 2.

- 17) Sean A y B dos números de n bits cada uno, representados en complemento a 2. Sean $A_{n-1}, A_{n-2}, \dots, A_0$ y $B_{n-1}, B_{n-2}, \dots, B_0$ los correspondientes bits de cada número, donde A_{n-1} y B_{n-1} son los respectivos bits de signo. Sea la suma de A y B un número S de n bits que se denominarían $S_{n-1}, \dots, S_0$ siguiendo la misma notación anterior, generándose al mismo tiempo un acarreo final de la suma C_n . Una función lógica que permite detectar el posible desbordamiento producido al sumar A y B tiene por expresión:

- ☐ A $\overline{A_{n-1}} \cdot \overline{B_{n-1}} \cdot S_{n-1} + A_{n-1} \cdot B_{n-1} \cdot \overline{S_{n-1}}$
☐ B $A_{n-1} \cdot \overline{B_{n-1}} \cdot C_n + B_{n-1} \cdot \overline{A_{n-1}} \cdot C_n$
☐ C C_n

- 18) Considere el número 4325. Un sistema (1) asume que el número está codificado en hexadecimal y, para ser procesado en binario lo transforma a binario natural. Otro sistema (2), por el contrario, valora que el número es decimal y lo coloca en su bus en BCD. Los números binarios que procesa cada sistema son:

- ☐ A (1) y (2) 0100001100100101
☐ B (1) 0100001100100101 y (2) 0001000011100101
☐ C (1) 0001000011100101 y (2) 0100001100100101
☐ D (1) y (2) 0001000011100101

- 19) Un ordenador utiliza palabras de 32 bits. ¿Cuántos números enteros en formato decimal pueden representar cada palabra en (1) BCD y en (2) ASCII de 8 bits? No considere signo.

- ☐ A) (1) 2^{32} y (2) 2^8
- ☐ B) (1) 10^8 y (2) 10^4
- ☐ C) (1) 2^8 y (2) 2^4
- ☐ D) (1) 8 y (2) 4

20) En la codificación binaria de números positivos y negativos ¿Mediante qué método de codificación se emplea una única combinación para codificar el 0?

- ☐ A) Complemento a 1.
- ☐ B) Signo Magnitud.
- ☐ C) Complemento a 2.

21) ¿Cuál es el rango para un número binario de 7 bits codificado en complemento a 2?

- ☐ A) [-64, 63].
- ☐ B) [0, 127].
- ☐ C) [-127, 127].

22) Considerando que el número A29 está expresado en hexadecimal, ¿Cómo se expresaría este número en binario y en BCD respectivamente?

- ☐ A) En binario: 1010 0010 1001; en BCD: 1 0000 0010 1001.
- ☐ B) En binario y en BCD se expresarían de igual forma: 1010 0010 1001.
- ☐ C) En binario: 1010 0010 1001 y en BCD: 0010 0110 0000 0001.

23) Indica qué particularidad tiene el código Gray:

- ☐ A) Que resulta muy útil para realizar operaciones aritméticas.
- ☐ B) Que entre dos combinaciones consecutivas sólo cambia un bit.
- ☐ C) Es un estándar para codificar en binario ciertos símbolos como las letras del alfabeto.

24) Indica el número mínimo de bits para representar el número -256 en complemento a 2:

- ☐ A) 10 bits.
- ☐ B) 8 bits.
- ☐ C) 9 bits.

25) De las afirmaciones siguientes:

- a) El número -106_{10} en representación complemento a 2 con 8 bits es 10111010.
- b) El número 10110101₂ en hexadecimal es B5₁₆.
- c) En representación de complemento a 1 de 9 bits el rango es [-512, +512].

Señala lo correcto:

- ☐ A) Las opciones a) y b) son correctas.

- ☐ B) Sólo la opción a) es correcta.
☐ C) La opción c) es correcta.

26) Señala la opción falsa

- ☐ A) $10000,11000_2 = 16,75_{(10)}$ y $10110_2 = 16_{(16)}$
☐ B) $1000011000_2 = 4140_{(8)}$ y $10110_2 = 16_{(16)}$
☐ C) $-24-26 = 1001101_{(C1)}$

27) Se quiere expresar el número 511 en representación de complemento a 2. Señala lo correcto.

- ☐ A) En C1 y en C2 se necesitan 10 bits
☐ B) En C1 se necesitan menos bits que en SM.
☐ C) La representación en C2 empieza por un 1.

28) La representación en complemento a 2 de un número binario es:

- ☐ A) Una forma de representar los números negativos exclusivamente.
☐ B) Una forma de representar los números enteros con signo.
☐ C) La opción b, incluyendo también al 0.

29) El código binario 110001 es:

- ☐ A) La representación binaria del número natural (sin signo) 49 con 6 bits.
☐ B) La representación binaria del número entero (con signo) -14 con 6 bits en complemento a 1.
☐ C) Las dos anteriores son ciertas.

30) ¿Cuál es el mínimo número de bits que tendríamos que usar para poder trabajar en complemento a 2 sumando y restando números enteros comprendidos entre -999 y +999?

- ☐ A) 11 bits.
☐ B) 12 bits.
☐ C) 13 bits.

31) ¿Cuál es el menor número de bits necesario para representar el número entero -16384 en complemento a 1?

- ☐ A) 16 bits.
☐ B) 17 bits.
☐ C) Ninguna de las anteriores.

32) ¿Cuál de las siguientes afirmaciones es falsa respecto a los sistemas de numeración en base hexadecimal?

☐ A Se usan los 10 dígitos decimales habituales (0,1,...9) y las 6 primeras letras del abecedario (A,B,...F) para representar cualquier valor, tanto entero como con parte fraccionaria.

☐ B Es necesario que aparezca alguno de los caracteres A, B,... F para que un número esté expresado en hexadecimal.

☐ C Para expresar cualquier número se necesita un número igual o menor de dígitos que en el sistema decimal.

33) Si realizas la suma en binario de $1001 + 0110$, el resultado que obtienes es 1111. ¿Qué significado tiene?

☐ A Es la suma en binario natural (sin signo) de 9 y 6 que produce el resultado 15.

☐ B Es la suma en complemento a 2 de -7 con +6, que produce el resultado -1.

☐ C Las dos anteriores son ciertas.

34) ¿Cuál es la representación en base 8 del número decimal 59,26?

☐ A 73,26

☐ B 59,205

☐ C 73,205

35) De las afirmaciones siguientes:

a) $274,136_8 = BC,2F_{16}$

b) La suma $A+B$ en Ca_2 con 6 bits, siendo $A = -32$, $B = 14$ es $A+B = 101110_{Ca_2}$

c) $10111100,001_2 = 188,125_{10}$

Elige una de las tres opciones Verdadero y Falso para a) b) y c)

☐ A V F V

☐ B V V F

☐ C V V V

36) La representación en complemento a 2 de un número binario es:

☐ A Una forma de representar los números negativos exclusivamente.

☐ B Una forma de representar los números enteros con signo.

☐ C La opción B, incluyendo también al 0.

2

FUNCIONES LÓGICAS

CONTENIDOS:

- Álgebra de Boole.
- Simplificación algebraica de funciones booleanas.
- Simplificación con mapas de Karnaugh.
- Formas canónicas.

EJERCICIOS RESUELTOS

1) Usando los axiomas de Boole y sus teoremas probar las siguientes igualdades:

a) $\overline{a} + a \cdot \overline{(\overline{a} \cdot b + \overline{b} \cdot c)} = \overline{a} + b + \overline{c}$

b) $a \cdot \overline{b} + \overline{b} \cdot \overline{c} + \overline{a} \cdot \overline{c} = a \cdot \overline{b} + \overline{a} \cdot \overline{c}$

a)

$$\begin{aligned}\overline{a} + a \cdot \overline{(\overline{a} \cdot b + \overline{b} \cdot c)} &= \overline{a} + a \cdot \overline{(\overline{a} \cdot b)} \cdot \overline{(\overline{b} \cdot c)} = \\ &= \overline{a} + a \cdot (a + \overline{b}) \cdot (b + \overline{c}) = \\ &= \overline{a} + (a \cdot a + a \cdot \overline{b})(b + \overline{c}) = \\ &= \overline{a} + a(1 + \overline{b})(b + \overline{c}) = \\ &= \overline{a} + a \cdot (b + \overline{c}) = \\ &= \overline{a} + (b + \overline{c}) = \\ &= \overline{a} + b + \overline{c}\end{aligned}$$

b)

$$\begin{aligned}a \cdot \overline{b} + \overline{b} \cdot \overline{c} + \overline{a} \cdot \overline{c} &= a \cdot \overline{b} + \overline{b} \cdot \overline{c} \cdot (a + \overline{a}) + \overline{a} \cdot \overline{c} = \\ &= a \cdot \overline{b} + a \cdot \overline{b} \cdot \overline{c} + \overline{a} \cdot \overline{b} \cdot \overline{c} + \overline{a} \cdot \overline{c} = \\ &= a \cdot \overline{b}(1 + \overline{c}) + \overline{a} \cdot \overline{c}(\overline{b} + 1) = \\ &= a \cdot \overline{b} + \overline{a} \cdot \overline{c}\end{aligned}$$

2) Demostrar que para las variables binarias a, b y c se cumple lo siguiente:

a) Si $a \cdot b = a \cdot c \rightarrow b = c$

b) Si $a \cdot b = a \cdot c$ y $a + b = a + c \rightarrow b = c$

a) Como hay que demostrar la negación de la implicación, con que se cumpla en un caso, nos basta para demostrar la propiedad.

Siendo $a = 0$ entonces $0 \cdot b = 0 \cdot c$ independientemente de b y c, luego no necesariamente $b = c$, por lo que ya en este caso se cumple la propiedad.

Siendo $a = 1$, $1.b = 1.c \rightarrow b = c$, éste es el caso que no la demuestra.

b) Como hay que demostrar la certeza de una implicación habrá que verificarse para todos los casos.

Siendo $a = 0$, $0.b = 0.c \rightarrow 0 = 0$ y $0 + b = 0 + c \rightarrow b = c$

Siendo $a = 1$, $1.b = 1.c \rightarrow b = c$ y $1 + b = 1 + c \rightarrow 1 = 1$

En los dos casos se llega a que si $a.b = a.c$ y $a + b = a + c$ entonces $b = c$, por lo que la propiedad queda demostrada.

3) Simplifica la función $f(x, y, z) = \overline{x(\bar{x} + y)} + xy$ aplicando Álgebra de Boole.

$$f(x, y, z) = \overline{x(\bar{x} + y)} + xy = (\overline{xx} + \overline{xy}) + xy = \overline{xy} + xy = 1$$

4) Simplifica la función F aplicando Álgebra de Boole.

$$F(x, y, z) = x\bar{y} + y\bar{z} + \bar{x}\bar{y}\bar{z} + \bar{x}y\bar{z} + x\bar{y}z$$

$$\begin{aligned} F(x, y, z) &= x\bar{y} + y\bar{z} + \bar{x}\bar{y}\bar{z} + \bar{x}y\bar{z} + x\bar{y}z = \\ &= x\bar{y} + y\bar{z} + \bar{x}\bar{y}(\bar{z} + z) + x\bar{y}z = \\ &= x\bar{y} + y\bar{z} + \bar{x}\bar{y} + x\bar{y}z = \\ &= y\bar{z} + (x + \bar{x})\bar{y} + x\bar{y}z = \\ &= y\bar{z} + \bar{y} + x\bar{y}z = \bar{y} + y(\bar{z} + xz) = \\ &= \bar{y} + y(\bar{z} + x) = \\ &= \bar{y} + \bar{z} + x = \end{aligned}$$

Finalmente se obtiene:

$$F(x, y, z) = x + \bar{y} + \bar{z}$$

5) Simplifica la función f aplicando Álgebra de Boole

$$f(x, y, z) = \overline{(xy + x)} \cdot \overline{(yxz + z)}$$

$$f(x, y, z) = \overline{(xy + x)} \cdot \overline{(yxz + z)} = \overline{(\bar{x} + \bar{y} + x)} \cdot \overline{(yx + z)} = \overline{(1 + \bar{y})} \cdot \overline{(yx + z)} = \bar{1} \cdot \overline{(yx + z)} = 0 \cdot \overline{(yx + z)} = 0$$

- 6) Dada la función $F(A, B, C, D) = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D}$, encuentra la expresión más simplificada de F aplicando axiomas/teoremas de Álgebra de Boole y utilizando al menos una puerta XOR.

$$\begin{aligned} F(A, B, C, D) &= \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} = \\ &= (\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}\overline{D}) + (\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D}) = \\ &= \overline{A}\overline{B}(\overline{C}\overline{D} + C\overline{D}) + \overline{A}\overline{B}(\overline{C}\overline{D} + C\overline{D}) = \\ &= (\overline{A}\overline{B} + \overline{A}\overline{B})(\overline{C}\overline{D} + C\overline{D}) = \\ &= (A \oplus B)(C \oplus D) \end{aligned}$$

$$F(A, B, C, D) = (A \oplus B)(C \oplus D)$$

- 7) Simplifica la función f aplicando axiomas y teoremas de Álgebra de Boole.

$$f(x, y, z) = x(\overline{xyz} + \overline{xyz} + \overline{yz} + x) + x(y + \bar{y})$$

$$f(x, y, z) = x(\overline{xyz} + \overline{xyz} + \overline{yz} + x) + x(y + \bar{y}) = x(\overline{xyz} + \overline{xyz} + \overline{yz} + x) + x \cdot 1 = x(\dots + 1)$$

$$f(x, y, z) = x$$

- 8) Dada la función $\overline{F}(a, b, c, d) = ab(\overline{cd} + \overline{d}) + b(\overline{ad} + \overline{a}\overline{cd})$, encuentra la expresión más simplificada de F aplicando axiomas/teoremas de Álgebra de Boole y utilizando al menos una puerta XNOR.

$$\begin{aligned}
 \overline{F}(a,b,c,d) &= ab(c\overline{d} + \overline{d}) + b(\overline{a}d + \overline{a}\overline{c}d) = \\
 &= ab\overline{d} + b(\overline{a}d + \overline{a}\overline{c}d) = \\
 &= ab\overline{d} + b\overline{a} \cdot (d + \overline{c}d) = \\
 &= ab\overline{d} + b\overline{a}d = \\
 &= b(\overline{a}d + \overline{a}d) = \\
 &= b(a \oplus d)
 \end{aligned}$$

Finalmente se obtiene:

$$F(a,b,c,d) = \overline{b} + \overline{(a \oplus d)}$$

9) Dada la función $F(x,y,z,w) = x\overline{(z+w)} + \overline{x(y \oplus z)} + \overline{x}$ encuentra la expresión más simplificada de F aplicando axiomas y teoremas de Álgebra de Boole.

$$\begin{aligned}
 F(x,y,z,w) &= x\overline{(z+w)} + \overline{x(y \oplus z)} + \overline{x} = \\
 &= x\overline{(z+w)} + \overline{\overline{x}} + \overline{(y \oplus z)} + \overline{x} = \\
 &= x\overline{(z+w)} + x + \overline{(y \oplus z)} + \overline{x} = \\
 &= x\overline{(z+w)} + \overline{(y \oplus z)} + x + \overline{x} = \\
 &= x\overline{(z+w)} + \overline{(y \oplus z)} + 1 = 1
 \end{aligned}$$

Finalmente, la solución es:

$$F(x,y,z,w) = 1$$

10) Simplifica la función f aplicando Álgebra de Boole.

$$f(x,y,z) = \overline{(x+y)}(x \oplus y) + xy\overline{(x \oplus y)}$$

$$\begin{aligned}
 f(x,y,z) &= \overline{(x+y)}(x \oplus y) + xy\overline{(x \oplus y)} = \\
 &= \overline{x} \cdot \overline{y} (x \oplus y) + (\overline{x} + \overline{y})(x \oplus y) = \\
 &= \overline{x} \cdot \overline{y} (x \oplus y) + \overline{x}(x \oplus y) + \overline{y}(x \oplus y) = \\
 &= \overline{x} \cdot \overline{y} (x \cdot \overline{y} + \overline{x} \cdot y) + \overline{x}(x \oplus y) + \overline{y}(x \oplus y) = \\
 &= 0 + \overline{x}(xy + \overline{x} \cdot \overline{y}) + \overline{y}(xy + \overline{x} \cdot \overline{y}) = \\
 &= \overline{x} \cdot \overline{y} + \overline{x} \cdot \overline{y} = \\
 &= \overline{x} \cdot \overline{y}
 \end{aligned}$$

Finalmente se obtiene:

$$f(x,y,z) = \overline{x} \cdot \overline{y}$$

11) Obtener la tabla de verdad de la siguiente función:

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

Antes de realizar la tabla de verdad procedemos a simplificar la función.

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

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

$$x + \bar{y}z$$

x	y	z	y'	y'z	x+y'z
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

12) Obtén la expresión simplificada como suma de productos de la función f . Utiliza mapas de Karnaugh. $f(a,b,c,d) = \prod M(1,3,4,5,9,11,12,13)$

El mapa de Karnaugh de la función es:

		c d			
		00	01	11	10
a b	00	1	0	0	1
	01	0	0	1	1
	11	0	0	1	1
	10	1	0	0	1

Agrupando unos se llega a $f(a,b,c,d) = \bar{b}\bar{d} + b.c$

13) Encontrar la primera forma canónica de la función:

$$f(x, y, z) = (A + \bar{C})(B + \bar{D})(\bar{B} + \bar{C} + D)(\bar{A} + C + \bar{D})$$

Sabiendo que cada suma corresponde a una agrupación de ceros obtiene el siguiente MK:

		c d				
		00	01	11	10	
F	00	1	0	0	0	$B + \bar{D}$
	01	1	1	0	0	
	11	1	0	1	0	$A + \bar{C}$
	10	1	0	0	1	
						$\bar{B} + \bar{C} + D$
						$\bar{A} + C + \bar{D}$
						$B + \bar{D}$

Luego la primera forma canónica es $f(x, y, z) = \sum m(0, 4, 5, 8, 10, 12, 15)$

14) Dada la función de cinco variables $F(a,b,c,d,e) = \sum m(1, 4, 6, 7, 9, 12, 14, 15)$ expresa la función como suma de productos (SP) y como producto de sumas (PS) utilizando el mapa de Karnaugh.

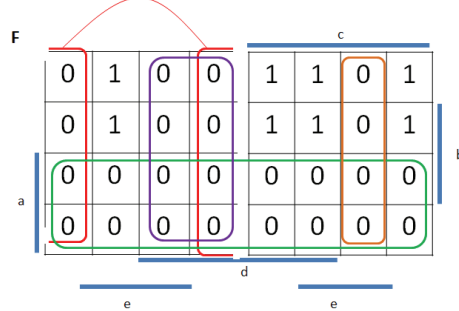
Primero se construye el MK de cinco variables de la función. Agrupando unos se llega a una expresión como SP.

		c d e							
		000	001	011	010	110	111	101	100
F	00	0	1	0	0	1	1	0	1
	01	0	1	0	0	1	1	0	1
	11	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0

$$F(a,b,c,d,e) = \bar{a}cd + \bar{a}\bar{c}\bar{e} + \bar{a}\bar{c}\bar{d}\bar{e}$$

Agrupando ceros se llega a una expresión como PS.

$$F(a,b,c,d,e) = \bar{a}(c+e)(c+\bar{d})(\bar{c}+d+\bar{e})$$



La expresión final es:

$$F(a,b,c,d,e) = \bar{a}(c+e)(c+\bar{d})(\bar{c}+d+\bar{e})$$

- 15) Dada la función $f(a,b,c,d,e) = \bar{a}\bar{c} + ad + abc\bar{d}\bar{e}$ expresa la función inversa como sumas de productos, utilizando el mapa de Karnaugh. Posteriormente negar la función inversa para obtener f .

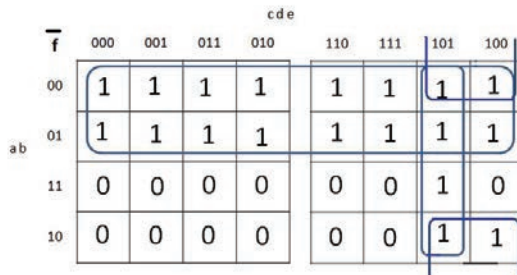
En primer lugar se construye el MK de cinco variables de la función inversa $\bar{f}$

Donde $a=1$ y $c=0$: $f=1$ ($\bar{f}=0$)

Donde $a=1$ y $d=1$: $f=1$ ($\bar{f}=0$)

Donde $a=1$, $b=1$, $c=1$, $d=0$, $e=0$: $f=1$ ($\bar{f}=0$)

A partir de esto se colocan los ceros en el mapa de Karnaugh de $\bar{f}$ directamente.



Agrupando unos se llega a: $\bar{f} = \bar{a} + c\bar{d}e + \bar{b}c\bar{d}$

Negando $\bar{f}$ se obtiene

$$f = \overline{\bar{f}} = \overline{\bar{a} + c\bar{d}e + \bar{b}c\bar{d}} = a[\overline{c\bar{d}e}][\overline{\bar{b}c\bar{d}}]$$

$$f = a(\bar{c} + d + \bar{e})(b + \bar{c} + d)$$

- 16) Dada la función $F(a,b,c,d) = \sum m(1,4,6,7,9,12,14,15)$ expresar la función como suma de productos utilizando el mapa de Karnaugh.

Colocando los minterms en el MK queda:

		cd			
		00	01	11	10
F	00	0	1	0	0
	01	1	0	1	1
	11	1	0	1	1
	10	0	1	0	0

Agrupando unos se obtiene:

$$F(a,b,c,d,e) = bc + b\bar{d} + \bar{b}cd$$

- 17) Calcula, utilizando mapa de Karnaugh, la primera y segunda formas canónicas de la función $H(a,b,c) = F + G$, siendo $F(a,b,c) = \bar{a}\bar{b} + ac + abc$, $G(a,b,c) = a \oplus b$

Los mapas de Karnaugh de F y G son:

		bc			
		00	01	11	10
F	0	1	1	1	0
	1	0	0	0	1

		bc			
		00	01	11	10
G	0	0	0	1	1
	1	1	1	0	0

Si $H(a,b,c) = F + G$, entonces su MK será:

		bc			
H	a	00	01	11	10
	0	1	1	1	1
	1	1	1	0	1

Luego, $H(a,b,c) = \sum m(0,1,2,3,4,5,6) = \prod M(7)$

18) Calcula, utilizando mapa de Karnaugh, la expresión simplificada de $H(a,b,c) = F \oplus G$ siendo $F(a,b,c) = \bar{a}\bar{b} + ac$ y $G(a,b,c) = a\bar{b} + ac + \bar{a}c$

Los mapas de Karnaugh de F y G son:

		bc			
F	a	00	01	11	10
	0	1	1	0	0
	1	0	1	1	0

		bc			
G	a	00	01	11	10
	0	0	1	1	0
	1	1	1	0	1

Si $H(a,b,c) = F \oplus G$, entonces su MK será

		bc			
H	a	00	01	11	10
	0	1	0	1	0
	1	1	0	1	1

Luego, $H(a,b,c) = \bar{b}\bar{c} + bc + ab$

19) Simplifica la expresión de f en mapa de Karnaugh como suma de productos y como producto de sumas $f(a,b,c,d) = \prod M(1,3,4,5,9,11,12,13)$. Esta notación representa los maxitérminos 1, 3, 4, etc...

Colocando los ceros en su lugar, el MK de f es el siguiente:

		c d			
f		00	01	11	10
	00	1	0	0	1
	01	0	0	1	1
	11	0	0	1	1
	10	1	0	0	1

Agrupando unos se tiene:

$$f(a,b,c,d) = \bar{b}\bar{d} + b.c$$

Agrupando ceros se tiene:

		c d			
f		00	01	11	10
	00	1	0	0	1
	01	0	0	1	1
	11	0	0	1	1
	10	1	0	0	1

$$f(a,b,c,d) = (\bar{b} + c)(b + \bar{d})$$

20) Calcula, utilizando mapa de Karnaugh, la expresión más simplificada como suma de productos de la función:

$$f(a,b,c,d,e) = \sum(0,1,2,3,4,5,6,7,9,15,18,19,22,23,26,27,30,31)$$

El MK de la función F es:

		c d e							
F		000	001	011	010	110	111	101	100
00		1	1	1	1	1	1	1	1
01	a b	0	1	0	0	0	1	0	0
11		0	0	1	1	1	1	0	0
10		0	0	1	1	1	1	0	0

Agrupando unos, se llega a $f(a, b, c, d, e) = \bar{a}\bar{b} + ad + \bar{a}\bar{c}\bar{d}e + cde$

21) Dada la función $F(a, b, c, d) = (\bar{a} + \bar{b})(a + d)(\bar{c} + d)$, expresada como producto de sumas, expresa la función como suma de productos utilizando mapas de Karnaugh.

Colocamos los grupos de ceros correspondientes a cada término producto de F y obtenemos el mapa de Karnaugh:

		cd			
F		00	01	11	10
00		0	1	1	0
01	ab	0	1	1	0
11		0	0	0	0
10		1	1	1	0

Agrupando unos obtenemos la expresión como SP:

$$F(a, b, c, d) = \bar{a}d + \bar{b}d + a\bar{b}\bar{c}$$

22) Dada la función $\bar{F}(a, b, c, d) = \sum m(0, 1, 2, 3, 9, 11)$ expresa la función como suma de productos utilizando el mapa de Karnaugh.

A partir de la función $\bar{F}$ se deduce que:

$$F = \bar{\bar{F}} = \prod M(0, 1, 2, 3, 9, 11)$$

Colocando ceros en el MK se tiene:

		cd			
	F	00	01	11	10
ab	00	0	0	0	0
	01	1	1	1	1
	11	1	1	1	1
	10	1	0	0	1

Agrupando unos se tiene:

$$F(a, b, c, d) = b + a\bar{d}$$

23) Demostrar que:

- Los operadores de inhibición e implicación no poseen las propiedades conmutativa ni asociativa.
- No existe la propiedad distributiva con los operadores NAND Y NOR.

a) Operador inhibición.

$$F_{in}(x, y) = \bar{x}y$$

- Es obvio que $\bar{x}y \neq x\bar{y}$ por lo que $F_{in}(x, y) \neq F_{in}(y, x)$

por tanto, no se cumple la conmutatividad.

- Definamos $F_{in}(x, y) = x * y$

$$x * (y * z) = x * (\bar{y}z) =$$

$$\bar{x} * (\bar{y}z) = \bar{x}.\bar{y}z =$$

$$(x * y) * z = (\bar{x}y) * z =$$

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

$\bar{x}.\bar{y}z \neq (x + \bar{y})z$ por lo que no se cumple la propiedad asociativa

Operador implicación.

$$F_{imp}(x, y) = x + \bar{y}$$

- $F_{imp}(x, y) \neq F_{imp}(y, x)$ ya que $x + \bar{y} \neq \bar{x} + y$

- Definimos $F_{imp}(x, y) = x \# y$

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

$x + \bar{y}z \neq x + \bar{y} + \bar{z}$, por lo que no se cumple la propiedad asociativa.

b) Representamos los operadores NAND y NOR como:

$$F_{NAND}(x, y) = x * y = \overline{(xy)}$$

$$F_{NOR}(x, y) = x \# y = \overline{(x + y)}$$

$$\begin{aligned} x * (y \# z) &= x * \overline{(y + z)} = \\ \overline{(x(y + z))} &= \bar{x} + y + z = \\ (x * y) \# (x * z) &= \overline{(xy)} \# \overline{(xz)} = \\ \overline{((\overline{(xy)}) + (\overline{(xz)}))} &= xy \cdot xz = xyz \end{aligned}$$

Como $\bar{x} + y + z \neq xyz$, no existe la ley distributiva de $*$ respecto de $\#$.

24) Realizar las siguientes operaciones:

a) Utilizando expresiones algebraicas expresar en su primera y segunda forma canónica la siguiente expresión:

$$E(a, b, c) = a\bar{b} + ac + bc$$

b) Expresar el complemento de $f(x, y, z) = \prod M(0, 1, 4, 6)$ en sus dos formas canónicas.

a) Un primer procedimiento será el de tipo algebraico (no recurriendo todavía a los mapas de Karnaugh). Multiplicamos cada término por unos, estando cada 1 formado por una variable y su complemento.

$$\begin{aligned}
 E(a, b, c) &= \bar{a}b(c + \bar{c}) + ac(b + \bar{b}) + bc(a + \bar{a}) = \\
 &= \bar{a}bc + \bar{a}b\bar{c} + acb + ac\bar{b} + bca + bc\bar{a} = \\
 &= \bar{a}bc + \bar{a}b\bar{c} + abc + a\bar{b}c
 \end{aligned}$$

Expresando la función como suma de minitérminos obtenemos la primera forma canónica.

$$\begin{aligned}
 E(a, b, c) &= m(011) + m(010) + m(111) + m(101) = \\
 &= \sum m(2, 3, 5, 7)
 \end{aligned}$$

Como la función es de tres variables, hay 8 términos mínimos. La segunda forma canónica se obtiene inmediatamente de la primera viendo la numeración de los minitérminos que faltan en la expresión.

$$E(a, b, c) = \prod M(0, 1, 4, 6)$$

b) La función f se expresa según su primera forma canónica como

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

La función complemento de f es: $\bar{f}(x, y, z) = \sum m(0, 1, 4, 6)$ expresada en su primera forma canónica.

La expresión en su segunda forma es: $\bar{f}(x, y, z) = \prod M(2, 3, 5, 7)$

25) Encontrar la suma de productos y el producto de sumas mínimos que son equivalentes a las siguientes expresiones:

a) $E(w, x, y, z) = \prod M(1, 3, 4, 7, 10, 13, 14, 15)$

b) $E(w, x, y, z) = \sum m(2, 8)$

a) Como se da la función en forma de maxitérminos, se construye el mapa de Karnaugh poniendo ceros en los lugares que se indican.

		yz			
	E	00	01	11	10
wx	00	1	0	0	1
	01	0	1	0	1
	11	1	0	0	0
	10	1	1	1	0

		yz			
	E	00	01	11	10
wx	00	1	0	0	1
	01	0	1	0	1
	11	1	0	0	0
	10	1	1	1	0

Suma de productos: Agrupando unos se obtiene:

$$E(w, x, y, z) = \bar{w}\bar{y}z + wxz + \bar{w}x\bar{z} + \bar{w}yz + wxyz$$

Producto de sumas: Agrupando ceros se obtiene:

$$\bar{E}(w, x, y, z) = wxz + wy\bar{z} + \bar{w}x\bar{z} + \bar{w}yz + \bar{w}x\bar{y}\bar{z}$$

$$E(w, x, y, z) = (\bar{w} + \bar{x} + \bar{z})(\bar{w} + \bar{y} + z)(w + x + \bar{z})(w + \bar{y} + \bar{z})(w + \bar{x} + y + z)$$

b) Del mismo modo, el mapa de Karnaugh es:

		yz			
	E	00	01	11	10
wx	00	0	0	0	1
	01	0	0	0	0
	11	0	0	0	0
	10	1	0	0	0

Annotations in the Karnaugh map:

- Green circle around (00,01) and (00,11) labeled $(w + y)$.
- Red circle around (00,10) and (01,10) labeled $\bar{z}$.
- Orange circle around (00,01), (01,01), (11,01), and (10,01) labeled $\bar{x}$.
- Purple circle around (10,01), (11,01), and (10,11) labeled $(\bar{w} + \bar{y})$.

La expresión como suma de productos es:

$$E = \bar{w}\bar{x}y\bar{z} + wx\bar{y}z$$

Agrupando ceros e invirtiendo las variables, se obtiene la expresión como producto de sumas:

$$E = \bar{x}\bar{z}(w + y)(\bar{w} + \bar{y})$$

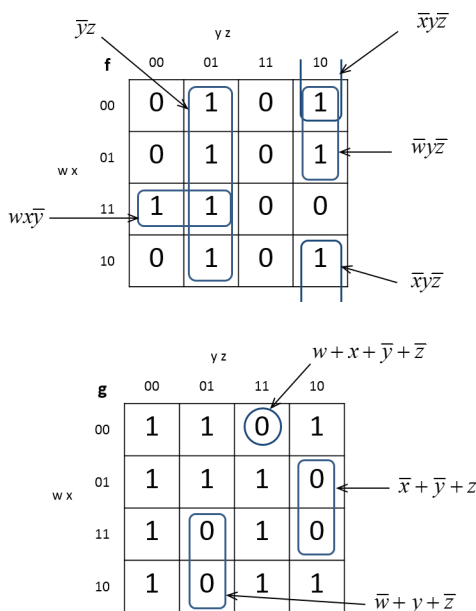
26) Encontrar la forma más simplificada posible en suma de productos de la función $F = f \cdot g$, donde f y g están dadas por:

$$f = wx\bar{y} + \bar{y}z + \bar{w}y\bar{z} + \bar{x}y\bar{z}$$

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

La resolución inmediata de este ejercicio es la siguiente: se obtienen los mapas de Karnaugh de las funciones f y g a partir de sus expresiones, se comparan ambos y se construye el mapa de F poniendo unos en los lugares donde aparecen dos unos en f y g .

Es obvio que, si F se obtiene de multiplicar f por g , los términos mínimos que forman F deben provenir a la vez de f y de g .



Como se ve los términos mínimos comunes son los de los números 1, 2, 5, 10, 12, por lo que el mapa de F es:

		yz			
	F	00	01	11	10
wx	00	0	1	0	1
	01	0	1	0	0
	11	1	0	0	0
	10	0	0	0	1

La expresión simplificada de F es, según este mapa:

$$F = w\bar{x}\bar{y}\bar{z} + \bar{w}\bar{y}z + \bar{x}y\bar{z}$$

TEST Y PREGUNTAS CORTAS

1) La expresión simplificada de $F = \bar{X}Y + XY\bar{Z} + X\bar{Y} + \bar{X}\bar{Y}\bar{Z}$ es:

☐ A $F = X \oplus Y + \bar{Z}$

☐ B $F = \overline{X \oplus Y} + \bar{Z}$

☐ C $F = \bar{Z}$

2) La función simplificada de $F(w, x, y, z) = \prod M(0, 1, 2, 3, 6, 7, 8, 10, 12, 14)$ es:

☐ A $F = x\bar{w}\bar{y} + wxy\bar{z} + w\bar{x}\bar{z}$

☐ B $F = \bar{w}\bar{x} + \bar{w}y + y\bar{z} + w\bar{z}$

☐ C $F = x\bar{w}\bar{y} + wz$

(Sugerencia: utilizar mapas de Karnaugh para resolver este ejercicio).

3) De las afirmaciones siguientes:

a) Es cierto que $\bar{x} \oplus y = \overline{x \oplus y}$

b) El número -1410 se puede representar en C a 2 con 12 bits.

c) Los niveles lógicos de entrada y salida son iguales para todas las tecnologías.

Elige una de las tres opciones Verdadero y Falso para a) b) y c).

☐ A V F V

☐ B V V F

☐ C F V F

4) Señala la respuesta correcta:

☐ A Las funciones $f(a, b, c) = ac + \bar{a}\bar{b}$ y $g(a, b, c) = ac + bc + \bar{a}\bar{b}$ son equivalentes.

☐ B Una puerta XOR con un 1 en una de sus entradas se comporta como una NOR con una entrada a 1.

☐ C Una puerta NAND de dos entradas puede hacerse sólo con dos puertas XOR de dos entradas.

5) La función $f(x, y, z) = \overline{(xy + x)} \cdot \overline{(yxz + z)}$ simplificada utilizando los axiomas y teoremas de Boole es:

☐ A Es siempre 1.

☐ B $f = xy$

☐ C Ni A) ni B) son ciertas.

- 6) La función $f(a,b,c,d,e) = \sum(0,1,2,3,4,5,6,7,9,15,18,19,22,23,26,27,30,31)$ simplificada es (usar mapas de Karnaugh):

- ☐ A $f(a,b,c,d,e) = \bar{a}\bar{b} + ad + \bar{a}\bar{c}\bar{d}e + cde$
☐ B $f(a,b,c,d,e) = (a + \bar{d})(\bar{a} + b + \bar{c} + d)(b + c + \bar{d})$
☐ C Ninguna respuesta anterior es correcta.

- 7) Señala lo falso:

- ☐ A En un mapa de Karnaugh de 5 variables una agrupación de 8 unos representan una variable.
☐ B Un cero en una tabla de verdad representa un maxterm de la función.
☐ C Una función no puede estar expresada en sus dos formas canónicas por minitérminos y maxitérminos de la misma numeración. Por ejemplo, si aparece m_4 en la primera forma, no puede aparecer M_4 en la segunda forma.

- 8) Sabiendo que la función booleana negada $\bar{F} = (A+B)(A+C)(A+B+C)$, la expresión simplificada aplicando Álgebra de Boole de F es:

- ☐ A $F = A + BC$
☐ B $F = \bar{A}\bar{B} + \bar{A}\bar{C}$
☐ C Ninguna de las anteriores.

- 9) La simplificación de la función $F(A,B,C,D) = \overline{\bar{A} + B} + B\bar{C} + \overline{(\bar{A} + B)(C + D)}$ produce el siguiente resultado:

- ☐ A $F = A + B + \bar{C}$
☐ B $F = A$
☐ C $F = 1$

- 10) La idea en la que se basa la simplificación de los Mapas de Karnaugh es:

- ☐ A Aplicar los axiomas y teoremas del Álgebra de Boole de forma sistemática y óptima.
☐ B Obtener la mejor simplificación posible a partir de la tabla de verdad usando sólo puertas AND.
☐ C Las dos anteriores son ciertas.

- 11) ¿Cuál de las siguientes afirmaciones es falsa?

- ☐ A Toda función booleana se puede expresar como producto de sumas.
☐ B Algunas funciones booleanas se pueden expresar como producto de algunas de sus variables (afirmadas o complementadas).

☐ Todas las funciones booleanas se pueden implementar utilizando sólo puertas XOR y XNOR.

12) La función booleana $F = \overline{A \cdot \overline{B} \cdot C \cdot D + \overline{A} \cdot B \cdot C \cdot \overline{D} + A \cdot \overline{B} \cdot \overline{C} \cdot D + \overline{A} \cdot B \cdot \overline{C} \cdot \overline{D}}$ se puede representar por:

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

☐ B $F = 1$

☐ C Ninguna de las anteriores.

13) ¿Cuál de las siguientes expresiones es falsa en Álgebra de Boole?

☐ A $x + y \cdot z = (x + y) \cdot (x + z)$

☐ B $x + x \cdot y = x + y$

☐ C $x + \overline{x} \cdot y + \overline{x} \cdot \overline{y} = 1$

14) El resultado de operar un elemento de un Álgebra de Boole con su complementario con las operaciones OR (suma lógica) y AND (producto lógico) es:

☐ A El propio elemento para cualquiera de las operaciones.

☐ B El elemento identidad de la propia operación.

☐ C El elemento identidad de la otra operación.

15) La función $f = \overline{x \cdot y + \overline{x} \cdot \overline{y} \cdot z} \cdot (x + \overline{y})$ simplificada utilizando los axiomas y teoremas de Boole es

☐ A Es siempre 1.

☐ B $f = xy$

☐ C Ni A) ni B) son ciertas.

16) Si un Mapa de Karnaugh tiene 3 unos y 7 ceros, ¿qué se puede asegurar de la función?

☐ A En principio es una función de al menos 4 variables, siempre que el resto de casos sean condiciones de “no importa”.

☐ B Habrá un producto de 2 variables, si la función es de 4 variables.

☐ C Las dos anteriores son ciertas.

17) Dado el Mapa de Karnaugh adjunto, ¿qué opción produce un circuito con menor cantidad de puertas lógicas, sin contar los inversores y obviando el número de entradas de las puertas?

- ☐ A Agrupar ceros y expresar la función como suma de productos.
☐ B Agrupar unos y expresar la función como suma de productos.
☐ C Agrupar ceros y expresar la función sólo con puertas NOR.

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

18) Dadas las siguientes funciones, ¿cuáles son equivalentes entre sí?

$$f(x, y, z) = \bar{x}z + x\bar{z} + \bar{x}y$$

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

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

- ☐ A $f(x, y, z)$ y $g(x, y, z)$
☐ B $f(x, y, z)$ y $h(x, y, z)$
☐ C Todas son equivalentes

19) Dada la función $F(A, B, C, D) = AB + \bar{B}.D + B.\bar{C}.\bar{D}$, ¿cuál de las siguientes afirmaciones es correcta:

- ☐ A Es equivalente a la función $F(A, B, C, D) = (B + D).(A + \bar{B} + \bar{D}).(A + \bar{B} + \bar{C}).$
☐ B Es equivalente a la función $F(A, B, C, D) = (B + D).(A + \bar{B} + \bar{D}).(A + \bar{C} + D).$
☐ C Es equivalente a ambas

20) Dada la siguiente función lógica de tres variables: $F(A, B, C) = (\bar{C}B + AC\bar{B})(B + \bar{A})$ indicar su expresión equivalente.

- ☐ A $m_2 + m_6$, siendo m_i un minterm de las variables A,B,C.
☐ B $m_0 + m_1 + m_3 + m_4 + m_5 + m_7$, siendo m_i un minterm de las variables A,B,C.
☐ C $M_4.M_5$, siendo M_i un maxterm de las variables A,B,C.

3

SISTEMAS COMBINACIONALES: PUERTAS

CONTENIDOS:

- Especificación de sistemas combinacionales.
- Puertas lógicas. TV y Funciones booleanas.
- Implementaciones AND/OR, OR/AND.
- Implementación con NAND-NOR.
- Análisis de circuitos con puertas lógicas.

EJERCICIOS RESUELTOS

- 1) Una puerta mayorista es un circuito digital cuya salida es 1 si la mayoría de entradas al circuito son unos. Por medio de una tabla de verdad, encontrar una función que lleve a cabo una compuerta mayorista de tres entradas. Simplificar la función obtenida, utilizando los axiomas y teoremas del Álgebra de Boole.

Disponemos de tres variables de entrada x, y, z y una salida F , $F = f(x, y, z)$

Para obtener la función que regula la puerta mayorista, pondremos todas posibles entradas y la salida correspondiente. Es decir, construimos una 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	1

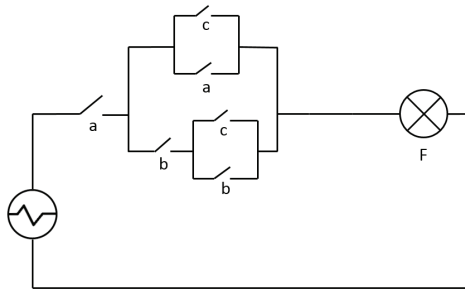
Se observa que F es 1 en cuatro combinaciones de entradas. A partir de aquí es obvio llegar a la expresión de F .

$$F = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz$$

Simplificamos la expresión de F :

$$\begin{aligned}
 F &= \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz = \\
 &= \bar{x}yz + x(\bar{y}z + y) = \\
 &= \bar{x}yz + x(z + y) = \\
 &= \bar{x}yz + xz + xy = \\
 &= z(\bar{x}y + x) + xy = \\
 &= z(y + x) + xy = \\
 &= zy + zx + xy
 \end{aligned}$$

2) Expresar el siguiente circuito de conmutación en notación lógica binaria. Obtenga la función lógica simplificada.



Podemos obtener una tabla de verdad de $F = F(a, b, c)$, obtener su expresión y luego simplificarla. Entenderemos valor 1 como interruptor cerrado y luz encendida, y valor 0 como interruptor abierto y luz apagada.

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

La expresión de F es:

$$\begin{aligned}
 F &= a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c} + abc = \\
 &= a\bar{b} + ab + a
 \end{aligned}$$

Luego, como ya era obvio, $F(a, b, c) = a$

3) Dada la 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	1

- a) Expresar f en producto de términos máximos.
b) Obtener la función simplificada en suma de productos.
c) Obtener la función simplificada en producto de sumas.

- a) Mirando en la tabla de verdad los lugares donde la función vale cero obtenemos:

$$f(x, y, z) = \prod M(0, 1, 2, 4)$$

- b) Como se pide función simplificada, recurrimos al mapa de Karnaugh de tres variables.

		y z			
f		00	01	11	10
x	0	0	0	1	0
	1	0	1	1	1

Agrupando unos se obtiene $f(x, y, z) = xz + xy + yz$

- c) Recurrimos nuevamente al mapa, pero ahora para agrupar los ceros. De esta manera obtenemos el complemento de la función.

		y z			
$\bar{f}$		00	01	11	10
x	0	1	1	0	1
	1	1	0	0	0

Se obtiene:

$$\overline{f}(x, y, z) = \overline{x} \overline{y} + \overline{x} \overline{z} + \overline{y} \overline{z}$$

Complementado otra vez:

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

4) Simplificar las siguientes funciones en suma de productos utilizando las condiciones de “no importa” que se especifican mediante la función d .

a) $f = \overline{y} + \overline{x} \overline{z}$ con $d = yz + xy$

b) $f = \overline{b} \overline{c} \overline{d} + bc \overline{d} + abc \overline{d}$ con $d = \overline{b} \overline{c} \overline{d} + \overline{a} b c d$

- a) La función depende en este caso de tres variables x, y, z . Situemos sobre el mapa el conjunto de unos y '-' correspondientes. Recordemos que, en un mapa de tres variables, el término y genera cuatro unos y el $\overline{x} \overline{z}$ genera dos unos. Por lo que respecta a las condiciones “no importa”, cada una de las expresiones yz xy generan dos condiciones de “no importa”.

		y z			
f		00	01	11	10
x	0	1	1	-	1
	1	1	1	-	-

Es obvio que $f = 1$, ya que se pueden considerar ocho unos agrupados.

- b) Seguimos el mismo procedimiento. Ahora la función depende de 4 variables a, b, c, d .

		c d				
		f	00	01	11	10
a b	00	1	0	0	-	
	01	0	-	0	1	
	11	0	0	0	1	
	10	1	0	0	-	

Se obtiene: $f(a, b, c, d) = \overline{c} \overline{d} + \overline{b} \overline{d}$

- 5) Obtener la expresión de conmutación de la siguiente función aplicando el teorema de Shannon. Descomponer las variables en dos grupos x_3, x_2 y x_1, x_0 . La función está dada por su tabla de verdad.

x_3	x_2	x_1	x_0	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Aplicamos el teorema de Shannon para cuatro variables,

$$E(x_3, x_2, x_1, x_0) = \sum_{i=0}^3 m_i(x_1, x_0) \cdot E(x_3, x_2, i)$$

El mapa de Karnaugh de la función es:

		$x_1 x_0$			
		00	01	11	10
$x_3 x_2$	00	0	0	1	1
	01	1	0	0	1
	11	0	1	0	1
	10	1	0	0	1

En este caso, de cada columna se obtiene la expresión de cada una de las cuatro funciones E_{ij} . ($E_{00}, E_{01}, E_{11}, E_{10}$)

$$E_{00} = \bar{x}_3 x_2 + x_3 \bar{x}_2$$

$$E_{01} = x_3 x_2$$

$$E_{10} = 1$$

$$E_{11} = \bar{x}_3 \bar{x}_2$$

Por lo tanto, la expresión final de f es:

$$E(x_3, x_2, x_1, x_0) = \bar{x}_1 \bar{x}_0 (\bar{x}_3 x_2 + x_3 \bar{x}_2) + \bar{x}_1 x_0 x_3 x_2 + x_1 \bar{x}_0 + x_1 x_0 \bar{x}_3 \bar{x}_2$$

6) Se desea controlar dos motores M_1 y M_2 por medio de tres interruptores a , b y c , de forma que se cumplan las siguientes condiciones:

1.- Si a está pulsado y los otros dos no, se activa M_1 .

2.- Si c está pulsado y los otros dos no, se activa M_2 .

3.- Si los tres interruptores están pulsados se activan los dos motores.

Bajo todas las demás condiciones los dos motores están desactivados. Hacer un estudio de la especificación de este sistema.

Para realizar una especificación del sistema obtendremos, en primer lugar, una tabla de verdad a partir del enunciado. Tendremos en cuenta que existen tres variables, que corresponden a los tres interruptores a , b y c , y que tendremos que hallar las expresiones de conmutación de dos funciones, cada una de ellas referente al funcionamiento de cada motor.

a	b	c	M_1	M_2	
0	0	0	0	0	
0	0	1	0	1	Condición 2
0	1	0	0	0	
0	1	1	0	0	
1	0	0	1	0	Condición 1
1	0	1	0	0	
1	1	0	0	0	
1	1	1	1	1	Condición 3

Se obtiene:

$$M_1 = \bar{a} \bar{b} \bar{c} + abc$$

$$M_2 = \bar{a} \bar{b} c + abc$$

7) Un sistema combinacional tiene una entrada x que representa un dígito decimal. Si la entrada es mayor que 4, la salida z es lo que le falta para llegar a 9; en otro caso la salida coincide con la entrada.

- a) Dar una descripción de alto nivel del sistema.
b) Obtener la expresión de conmutación de la salida.

a) Especificación de alto nivel.

Entrada $x \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Salida

$(z = 9 - x)$ si $x > 4$

$z = x$ si $x < 5$, $z \in \{0, 1, 2, 3, 4\}$

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

- b) Hacemos una codificación binaria de la entrada y la salida. Para la entrada tenemos que disponer al menos de 4 variables ya que hay 10 posibles valores. Para la salida nos basta con 3 variables binarias para codificar 5 valores.

$$x = (x_3, x_2, x_1, x_0)$$

$$z = (z_2, z_1, z_0)$$

A continuación, se construye la tabla de verdad, siguiendo el enunciado.

	x_3	x_2	x_1	x_0	z_2	z_1	z_0
0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	1
2	0	0	1	0	0	1	0
3	0	0	1	1	0	1	1
4	0	1	0	0	1	0	0
5	0	1	0	1	1	0	0
6	0	1	1	0	0	1	1
7	0	1	1	1	0	1	0
8	1	0	0	0	0	0	1
9	1	0	0	1	0	0	0
-	-	-	-	-	-	-	-

Las demás combinaciones no se dan en este caso. Podemos suponer, por tanto, que hay seis condiciones de “no importa” en este problema.

A continuación, no queda más que realizar los mapas de Karnaugh de las tres variables de salida y hallar sus expresiones simplificadas.

		$x_1 x_0$			
	z_2	00	01	11	10
$x_3 x_2$	00	0	0	0	0
	01	1	1	0	0
	11	-	-	-	-
	10	0	0	-	-

$$z_2 = \overline{x_2 x_1}$$

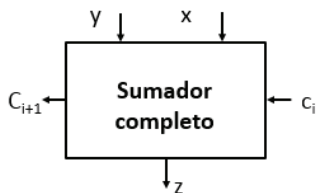
		$x_1 x_0$			
	z_1	00	01	11	10
$x_3 x_2$	00	0	0	1	1
	01	0	0	1	1
	11	-	-	-	-
	10	0	0	-	-

$$z_1 = x_1$$

		$x_1 x_0$			
	z_0	00	01	11	10
$x_3 x_2$	00	0	1	1	0
	01	0	0	0	1
	11	-	-	-	-
	10	1	0	-	-

$$z_0 = \overline{x_3} \overline{x_2} x_0 + \overline{x_2} x_1 \overline{x_0} + x_3 \overline{x_0}$$

- 8) Un sumador completo es un circuito combinacional que realiza la suma aritmética de 3 bits de entrada x, y, c_i . Llamamos c_i al bit de arrastre de una posición previa. Las variables de salida son c_{i+1} y z . Realizar una implementación con puertas OR, AND y XOR.



La tabla de verdad del sumador completo es:

x	y	c_i	c_{i+1}	z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Simplificamos las expresiones de c_{i+1} y z .

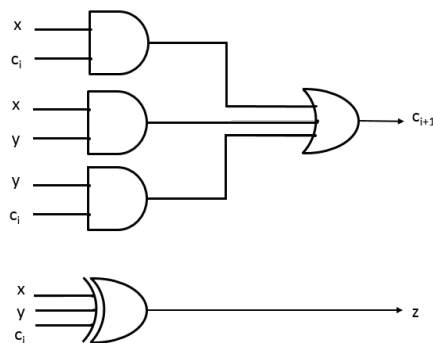
		$C_i y$				
		C_{i+1}	00	01	11	10
x	0	0	0	1	0	
	1	0	1	1	1	

$$c_{i+1} = xy + xc_i + yc_i$$

		$C_i y$			
		00	01	11	10
x	0	0	1	0	1
	1	1	0	1	0

$$z = x \oplus y \oplus c_i$$

La implementación es:

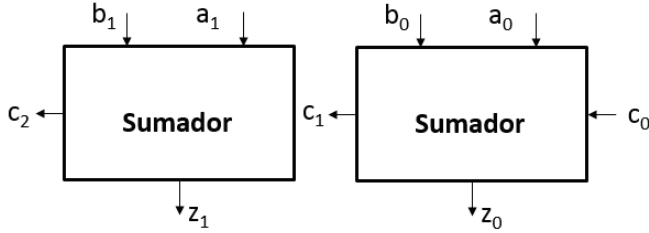


9) Describir un módulo sumador de entradas A y B enteros en el rango 0 a 3.

- Descripción de alto nivel usando expresiones aritméticas.
- Especificación de bajo nivel.
- Expresiones de conmutación reducidas.

Un sumador elemental es aquél que suma 3 bits, dos de ellos proceden de los sumandos y el otro procede de la suma anterior (si la hay) y se le llama bit de arrastre anterior.

Si el circuito combinacional suma enteros del 0 al 4 se necesitarán dos módulos sumadores elementales en cadena (recuérdese que para codificar en número 3 se necesitan al menos 2 bits).



a) Especificación de alto nivel.

Entradas

$$A, B \in \{0, 1, 2, 3\}$$

$c_{in} \in \{0, 1\}$, éste es un bit de arrastre anterior que consideramos de valor inicial 0.

Salidas $z = (A + B + c_0) \bmod 4$

$$c_{out} = 1 \text{ si } (A + B + c_0) \geq 4$$

$$c_{out} = 0 \text{ resto de casos}$$

b) Especificación de bajo nivel.

$$A = \sum_{i=0}^1 a_i \cdot 2^i. \text{ A queda codificado por el vector binario } a = (a_1, a_0)$$

$$B = \sum_{i=0}^1 b_i \cdot 2^i, \text{ quedando codificado por el vector binario } b = (b_1, b_0)$$

$$z = \sum_{i=0}^1 z_i \cdot 2^i, \text{ quedando codificado por el vector binario } z = (z_1, z_0)$$

En general, con esta codificación se tienen las expresiones de las variables de salida y arrastres:

$$\text{Para } i = 0, z_i = (a_i + b_i + c_i) \bmod 2 \text{ siendo } c_0 = c_{in}$$

$$c_{i+1} = 1 \text{ si } a_i + b_i + c_i \geq 2$$

$$c_{i+1} = 0 \text{ para resto de los casos, siendo } c_2 = c_{out}$$

La tabla de verdad que recoge esta especificación es para $i = 0, 1$ es la siguiente:

a_i	b_i	c_i	C_{i+1}	z_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- c) Las expresiones de conmutación las hallamos mediante los mapas de Karnaugh de las funciones generales c_{i+1} y z_i

		$b_i c_i$				
		c_{i+1}	00	01	11	10
a_i	0	0	0	1	0	
	1	0	1	1	1	

$$c_{i+1} = a_i c_i + b_i c_i + a_i b_i$$

		$b_i c_i$				
		z_i	00	01	11	10
a_i	0	0	1	0	1	
	1	1	0	1	0	

$$z_i = \overline{a_i} \overline{b_i} c_i + a_i b_i c_i + \overline{a_i} b_i \overline{c_i} + a_i \overline{b_i} \overline{c_i}$$

Éstas son las expresiones generales para un módulo sumador elemental. En nuestro caso estas expresiones se calculan para $i = 0$ e $i = 1$ teniendo presente que $c_0 = 1$.

- 10) Dado un sistema combinacional que calcula la distancia entre dos unos en el vector de entrada $x = (x_3, x_2, x_1, x_0)$ y asumiendo que $\underline{x}$ tiene exactamente dos unos, hallar las funciones de conmutación de las salidas del sistema.

La salida s se codifica en un bit vector $s(s_1, s_0)$ ya que la máxima distancia entre dos unos es de 3.

x_3	x_2	x_1	x_0	s_1	s_0
0	0	0	0	-	-
0	0	0	1	-	-
0	0	1	0	-	-
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	1	-	-
1	0	0	0	-	-
1	0	0	1	1	1
1	0	1	0	1	0
1	0	1	1	-	-
1	1	0	0	0	1
1	1	0	1	-	-
1	1	1	0	-	-
1	1	1	1	-	-

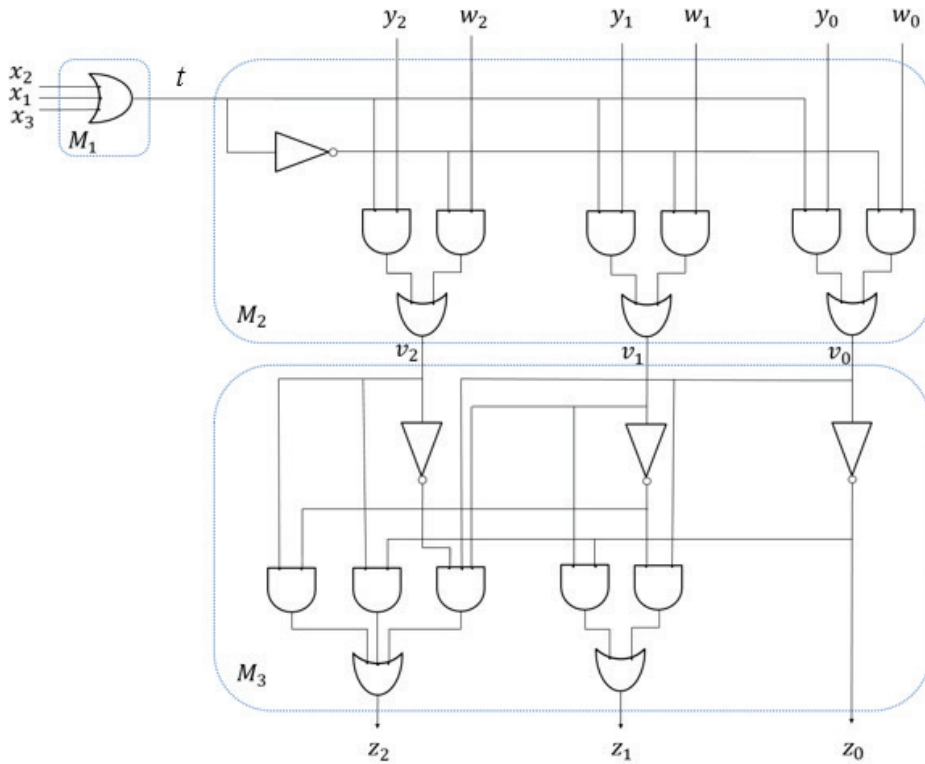
		$x_1 x_0$			
	s_1	00	01	11	10
$x_3 x_2$	00	-	-	0	-
	01	-	1	-	0
	11	0	-	-	-
	10	-	1	-	1

$$s_1 = x_3 x_2 + \overline{x_1} \overline{x_0}$$

		$x_1 x_0$			
	s_0	00	01	11	10
$x_3 x_2$	00	-	-	1	-
	01	-	0	-	1
	11	1	-	-	-
	10	-	1	-	0

$$s_0 = \overline{x_3} x_1 + x_3 \overline{x_1}$$

11) Analizar el circuito combinacional de la figura. Hacerlo de forma modular.



Se va a analizar los módulos señalados M_1 , M_2 , y M_3 por separado y posteriormente se hará un estudio conjunto.

Módulo M_1

$$t = x_2 + x_1 + x_0$$

Módulo M_2

$$v_2 = ty_2 + \bar{t}w_2$$

$$v_1 = ty_1 + \bar{t}w_1$$

$$v_0 = ty_0 + \bar{t}w_0$$

Módulo M_3

$$z_2 = v_2 \bar{v}_1 + v_2 \bar{v}_0 + v_2 v_1 v_0$$

$$z_1 = v_1 \bar{v}_1 + v_1 v_0 = v_1 \oplus v_0$$

$$z_0 = \bar{v}_0$$

A continuación, se realiza un estudio de alto nivel de cada módulo. Los vectores x, y, w y z representan enteros positivos en representación binaria.

M_1

$$x = (x_2, x_1, x_0)$$

$$\text{Resulta obvio que } t = \begin{cases} 1 & \text{si } x \neq 0 \\ 0 & \text{resto} \end{cases}$$

M_2

$$v = (v_2, v_1, v_0)$$

$$v = \begin{cases} y & \text{si } t = 1 \\ w & \text{si } t = 0 \end{cases}$$

M_3

$$z = (z_2, z_1, z_0)$$

La función z no se obtiene de forma inmediata, por lo que recurriremos a realizar su tabla de verdad para inferir su acción.

v_2	v_1	v_0	z_2	z_1	z_0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Como se puede comprobar z se obtiene de sumar una unidad (módulo 8) al valor decimal de v . Comprobar que para $v = 7$ la función de salida es $z = 0$.

$$\text{Luego } z = (v+1) \bmod 8$$

Reuniendo la especificación de los tres módulos se llega a:

$$z = \begin{cases} (y+1) \bmod 8 & \text{si } x \neq 0 \\ (w+1) \bmod 8 & \text{si } x = 0 \end{cases}$$

Por lo tanto, en cualquier caso, se trata de un sistema incrementador en una unidad de alguna de las entradas y o w .

- 12) Diseñar un circuito combinacional que convierta un dígito decimal del código 84-2-1 (con pesos +8, +4, -2 y -1) al código BCD. Utilizar, si es posible, puertas XOR ($\oplus$) y XNOR (Equivalencia).

En primer lugar, verificamos una tabla de conversión del código 84-2-1 al código BCD. Hay que hacer notar que en las combinaciones de las 4 variables que utilizamos existen 6 que no se dan nunca. Éstas corresponden a la representación en binario de los números 10 a 15.

Código 84-2-1 Código BCD

a	b	c	d	w	x	y	z
0	0	0	0	0	0	0	0
0	0	0	1	-	-	-	-
0	0	1	0	-	-	-	-
0	0	1	1	-	-	-	-
0	1	0	0	0	1	0	0
0	1	0	1	0	0	1	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1
1	1	0	0	-	-	-	-
1	1	0	1	-	-	-	-
1	1	1	0	-	-	-	-
1	1	1	1	1	0	0	1

Hallamos las funciones simplificadas de w , x , y , z mediante mapas de Karnaugh.

		c d			
w		00	01	11	10
a b	00	0	-	-	-
	01	0	0	0	0
	11	-	-	1	-
	10	1	0	0	0

$$w = ab + a\bar{c}\bar{d}$$

		c d			
x		00	01	11	10
a b	00	0	-	-	-
	01	1	0	0	0
	11	-	-	0	-
	10	0	1	1	1

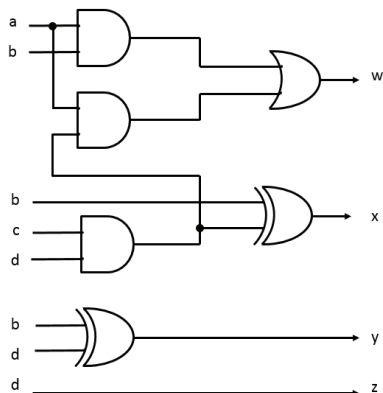
$$\begin{aligned}
 x &= \bar{b}c + \bar{b}d + b\bar{c}\bar{d} = \\
 &= \bar{b}(c+d) + b\bar{c}\bar{d} = \\
 &= \bar{b}(\bar{c}\bar{d}) + b(\bar{c}\bar{d}) = \\
 &= b \quad (\bar{c}\bar{d})
 \end{aligned}$$

		c d			
y		00	01	11	10
a b	00	0	-	-	-
	01	0	1	0	1
	11	-	-	0	-
	10	0	1	0	1

$$y = \bar{c}d + c\bar{d} = c \oplus d$$

La expresión de la variable z es bien sencilla, basta fijarnos en la tabla de verdad de la página anterior para darnos cuenta de que $z = d$.

El esquema lógico se presenta a continuación.



13) Diseñar una red de puertas AND, OR y Equivalencia que implemente un sistema combinacional con tres entradas: a y b binarias y c con tres posibles valores (mayor, igual, menor), y una salida z que puede tomar los mismos valores que c . La función de salida z está dado por:

$z=\text{mayor si } (a = b \text{ y } c = \text{mayor}) \text{ ó } a > b$

$z=\text{igual si } a = b \text{ y } c = \text{igual}$

$z=\text{menor si } (a = b \text{ y } c = \text{menor}) \text{ ó } a < b$

En primer lugar, haremos una codificación de las variables z y c . Nos bastaría utilizar dos variables binarias para poder la codificación de tres valores, pero se van a utilizar 3 variables binarias a efectos de simplificar la red.

La codificación más sencilla con 3 variables es:

C	c_2	c_1	c_0
Z	z_2	z_1	z_0
Mayor	1	0	0
Igual	0	1	0
Menor	0	0	1

En vez de poner la tabla de verdad para las funciones de salida z_2 , z_1 , z_0 con las 5 variables binarias de entrada, vamos a resolver el problema obteniendo las expresiones de estas funciones de forma inmediata a partir del enunciado.

Hay que recordar que:

- La función de igualdad (a igual a b) tiene por expresión de conmutación
 $f(a,b) = ab + \bar{a}\bar{b}$

- La función de comparación (a mayor que b) tiene por expresión de conmutación
 $f(a,b) = a\bar{b}$

- La función de comparación (b mayor que a) tiene por expresión de conmutación
 $f(a,b) = \bar{a}b$

Todo esto es posible saberlo si hacemos la tabla de verdad de estas funciones con las variables a y b :

a	b	$f(=)$	$f(>)$	$f(<)$
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Lo que nos queda es hacer una transcripción del enunciado en funciones de conmutación, sustituyendo los 'y' por productos (puertas AND) y los 'o' por sumas (puertas OR).

- z_2 es únicamente 1 si z es mayor $\rightarrow$ primera condición

$$z_2 = (ab + \bar{a}\bar{b})c_2 + \bar{a}\bar{b} = (a \quad b)c_2 + \bar{a}\bar{b}$$

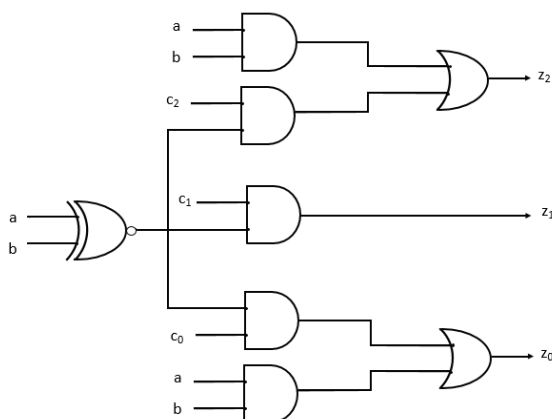
- z_1 es únicamente 1 si z es igual $\rightarrow$ segunda condición

$$z_1 = (ab + \bar{a}\bar{b})c_1 = (a \quad b)c_1$$

- z_0 es únicamente 1 si z es menor $\rightarrow$ tercera condición

$$z_0 = (ab + \bar{a}\bar{b})c_0 + \bar{a}\bar{b} = (a \quad b)c_0 + \bar{a}\bar{b}$$

El diagrama de implementación es:



14) Diseñar una red de puertas de ocho entradas y una salida, que controla un sistema de alarma. La función de la red está especificada por:

La alarma z es activada cuando el sistema está disponible ($e=1$) y alguno de los cuatro sensores de cierre nc_i , $i=1,2,3,4$ está abierto (0) o alguno de los sensores de apertura no_i , $i=1,2,3$ está cerrado (1).

Como vemos existen 7 variables binarias de entrada y una de salida. Sería absurdo construir una tabla de verdad con 128 filas, y obtener a continuación la expresión de z . El camino apropiado, es deducir la expresión de conmutación de la salida de una forma inmediata a partir del enunciado de especificación del sistema.

Tomaremos el acuerdo siguiente:

Estado	Valor binario
<i>Abierto</i>	0
<i>Cerrado</i>	1
<i>Activo</i>	1
<i>Inactivo</i>	0

Siguiendo el enunciado tenemos:

- 'La alarma está activada si...' $\rightarrow z = 1$

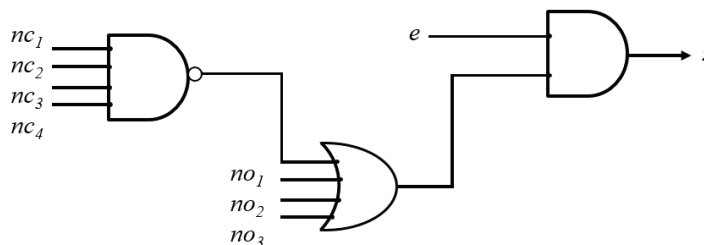
- 'alguno de los nc_i está abierto...' $\rightarrow \overline{nc_1} + \overline{nc_2} + \overline{nc_3} + \overline{nc_4} = \overline{nc_1.nc_2.nc_3.nc_4}$

- 'alguno de los no_i está cerrado...' $\rightarrow no_1 + no_2 + no_3$

Después de estas aclaraciones es obvio concluir que la expresión de z es:

$$z = (\overline{nc_1.nc_2.nc_3.nc_4}) + no_1 + no_2 + no_3$$

La implementación es:



15) Simplificar cada una de las siguientes funciones e implementarlas con puertas NAND.

a) $f_1 = \bar{a}\bar{c} + ace + ace + \bar{a}\bar{c}\bar{d} + \bar{a}\bar{d}\bar{e}$

b) $f_2 = (\bar{b} + \bar{d})(\bar{a} + \bar{c} + d)(a + \bar{b} + \bar{c} + d)(\bar{a} + b + \bar{c} + d)$

- a) Como sabemos para implementar una función con puertas NAND es preciso obtener la expresión de la función en forma de suma de términos producto. En este caso, f_1 está en esta forma, pero no sabemos si está simplificada. Lo primero será intentar simplificar f_1 .

Algebraicamente:

$$\begin{aligned}
 f_1 &= a(\bar{c} + ce + c\bar{e}) + \bar{a}\bar{d}(c + \bar{e}) = \\
 &= a(\bar{c} + c) + \bar{a}\bar{d}(c + \bar{e}) = \\
 &= a + \bar{a}\bar{d}(c + \bar{e}) = \\
 &= a + \bar{d}(c + \bar{e}) = \\
 &= a + c\bar{d} + \bar{d}\bar{e}
 \end{aligned}$$

O bien, calculado el MK se obtiene:

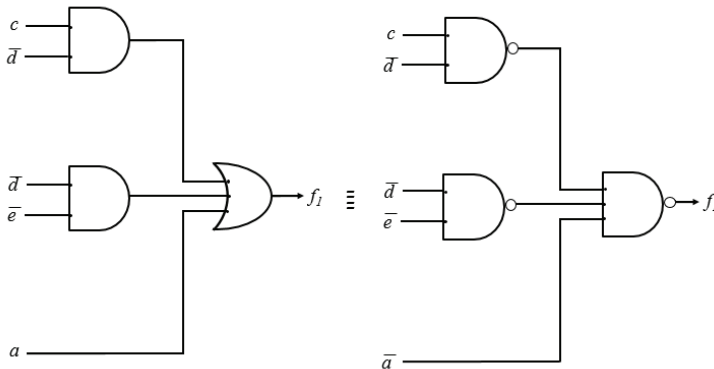
		c d			
	f_1	00	01	11	10
a e	00	1	0	0	1
	01	0	0	0	1
	11	1	1	1	1
	10	1	1	1	1

$$f_1 = a + c\bar{d} + \bar{d}\bar{e}$$

Ciertamente la función f_1 no estaba dada en forma simplificada. Complementando dos veces f_1 se obtiene:

$$f_1 = \overline{(\bar{a}(\bar{c}\bar{d}) \cdot (\bar{d}\bar{e}))}$$

Se necesitan 3 puertas NAND para implementar esta función.



b) En este caso la función está dada en forma de producto de sumas. Representamos f_2 en mapa de Karnaugh y después agrupamos unos.

$$\overline{(f_2)} = bd + ac\bar{d} + \bar{a}bcd + \bar{a}\bar{b}cd$$

Esta expresión complemento nos proporciona los ceros del mapa.

		c d			
	f_2	00	01	11	10
a b	00	1	1	1	1
	01	1	0	0	0
	11	1	0	0	0
	10	1	1	0	0

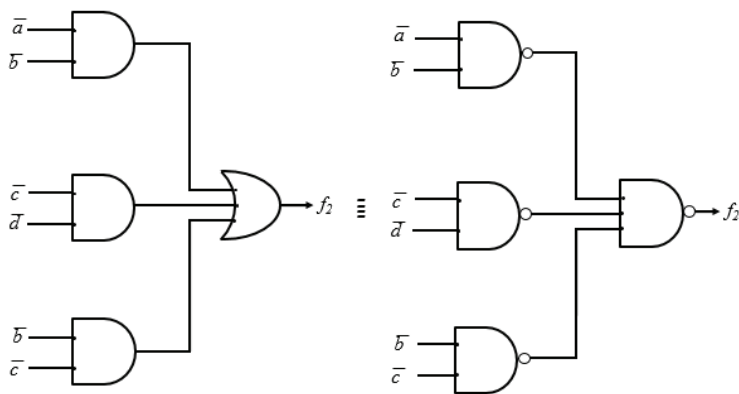
La expresión de f_2 simplificada es:

$$f_2 = \bar{a}\bar{b} + \bar{c}\bar{d} + \bar{b}\bar{c}$$

Complementando dos veces se llega a:

$$f_2 = \overline{(\overline{a\bar{b}}) \cdot (\overline{c\bar{d}}) \cdot (\overline{b\bar{c}})}$$

Se necesitan 4 puertas NAND para implementar el circuito.



16) Implementar las siguientes funciones con puertas NOR:

a) $f_1 = \bar{a}\bar{b} + \bar{c}\bar{d} + \bar{a}\bar{c}\bar{d} + \bar{d}\bar{c}(ab + \bar{a}\bar{b}) + db(\bar{a}\bar{c} + \bar{a}\bar{c})$

b) $f_2 = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}cd + \bar{a}bcd$

El procedimiento es el mismo que en el ejercicio anterior, pero en este caso expresaremos las funciones como producto de sumas de forma simplificada, por lo que haremos agrupaciones de ceros sobre el mapa de Karnaugh.

a) Para la función f_1 :

		c d			
f_1		00	01	11	10
a b	00	1	1	0	1
	01	1	0	1	1
	11	1	1	0	0
	10	1	1	1	1

$$\overline{(f_1)} = abc + \bar{a}\bar{b}cd + \bar{a}\bar{b}cd$$

Luego, $f_1 = (\bar{a} + \bar{b} + \bar{c})(a + \bar{b} + c + \bar{d})(a + b + \bar{c} + \bar{d})$ y la función con puertas NOR será:

$$f_1 = \overline{\overline{(\bar{a} + \bar{b} + \bar{c})} + \overline{(a + \bar{b} + c + \bar{d})} + \overline{(a + b + \bar{c} + \bar{d})}}$$

b) De igual forma para f_2 :

		c d			
f_2		00	01	11	10
a b	00	0	0	0	0
	01	0	1	0	1
	11	0	0	0	0
	10	0	1	0	1

$$\overline{(f_2)} = cd + \bar{c}\bar{d} + ab + \bar{a}\bar{b}$$

$$f_2 = \overline{(\bar{c} + \bar{d})(c + d)(\bar{a} + \bar{b})(a + b)}$$

Con puertas NOR resulta:

$$f_2 = \overline{\overline{(\bar{c} + \bar{d})} + \overline{(c + d)} + \overline{(\bar{a} + \bar{b})} + \overline{(a + b)}}$$

17) Diseñar un circuito combinacional que detecte un error en la representación de un dígito decimal en BCD. Implementar con puertas NOR.

Hay que obtener un diagrama lógico cuya salida f sea 1 en el caso de que las entradas correspondan a una combinación no permitida en el código BCD. Como se sabe en la representación en BCD de un número se necesitan cuatro variables binarias, y las 6 últimas combinaciones no se usan nunca debido a que la codificación es del 0 al 9

decimal. Si aparece alguna de estas combinaciones admitiremos que existe un error y la salida será 1.

Supongamos a, b, c y d entradas, y f salida.

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

Pasando al mapa de Karnaugh:

		c d			
		00	01	11	10
f	00	0	0	0	0
	01	0	0	0	0
	11	1	1	1	1
	10	0	0	1	1
a b					

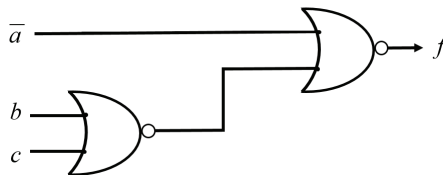
$$\bar{f} = \bar{a} + \bar{b} \bar{c}$$

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

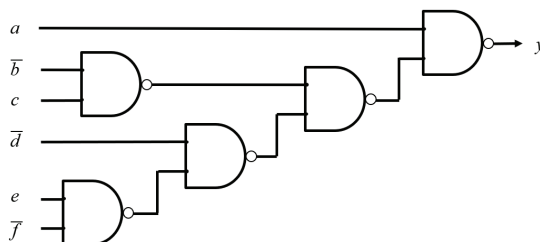
Complementando dos veces se tiene:

$$f = \overline{\overline{a + b + c}}$$

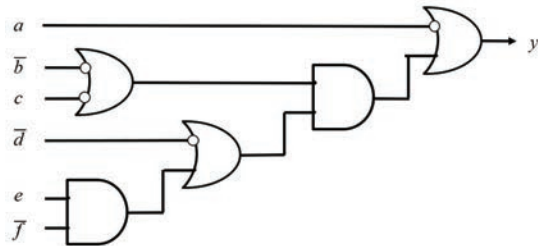
Por lo tanto, se necesitan dos puertas NOR.



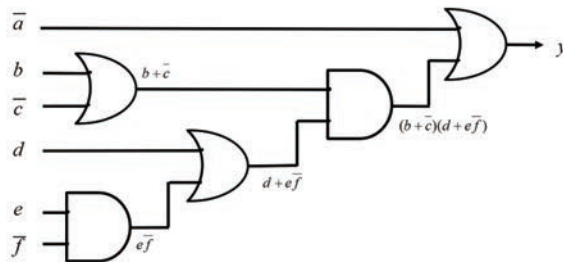
18) Analizar las redes NOR y NAND de la figura siguiente:



Antes de analizar la red intentaremos simplificarla.



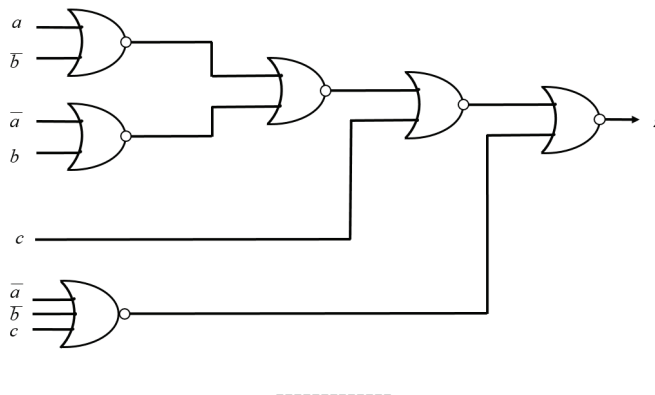
Se pueden eliminar todos los círculos de complemento, quedando el circuito con puertas AND y OR.



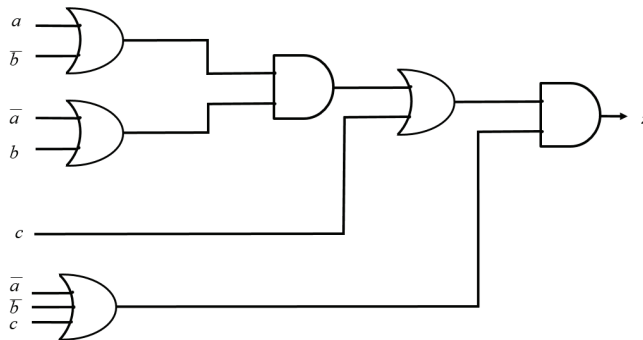
y analizamos esta red se obtiene:

$$y = \bar{a} + (b + \bar{c})(d + e\bar{f})$$

19) Analizar las redes NOR y NAND de la figura siguiente:



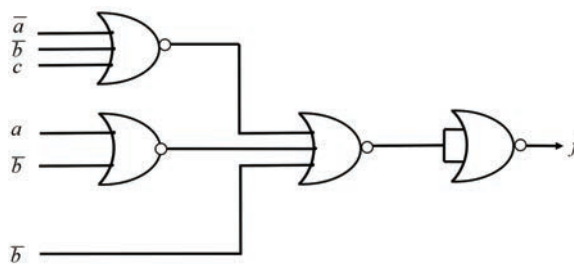
Igual que antes, se intenta llegar a un circuito realizado con puertas AND y OR para, posteriormente, ser fácilmente analizado.



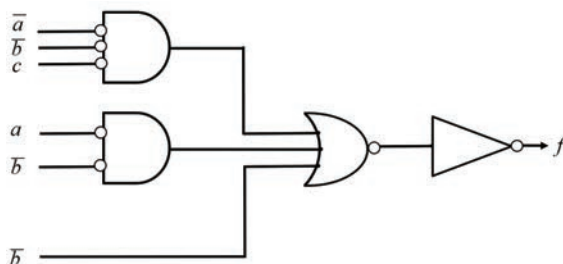
Se obtiene:

$$\begin{aligned}
 z &= [(a + \bar{b}).(\bar{a} + b) + c].[\bar{a} + \bar{b} + c] = \\
 &= (ab + \bar{a}\bar{b} + c).(\bar{a} + \bar{b} + c) = \\
 &= abc + \bar{a}\bar{b} + \bar{a}\bar{b}c + c\bar{a} + c\bar{b} + c = \\
 &= abc + \bar{a}\bar{b}[1 + 1 + c] + c[\bar{a} + \bar{b} + 1] = \\
 &= abc + \bar{a}\bar{b} + c = \\
 &= (ab + 1)c + \bar{a}\bar{b} = \\
 &= c + \bar{a}\bar{b}
 \end{aligned}$$

20) Determinar la función de salida del circuito de la figura. Obtener un circuito equivalente con puertas NOR.



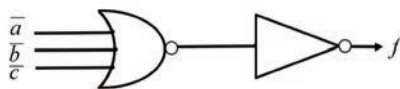
Invirtiendo la representación de las puertas de forma alternada y empezando por la de nivel superior se obtiene una expresión de f .



$$\begin{aligned}
 f &= abc + \bar{a}b + \bar{b} = \\
 &= abc + \bar{a} + \bar{b} = \\
 &= b\bar{c} + \bar{a} + \bar{b} = \\
 &= \bar{a} + \bar{b} + \bar{c}
 \end{aligned}$$

Obteniendo f con puertas NOR queda:

$$f = \overline{(\overline{\bar{a} + \bar{b} + \bar{c}})}$$



- 21) Un almacén utiliza un sistema automático para inventariar los distintos elementos almacenados. Éstos se sitúan en diversos estantes alineados por columnas. Así, el sistema de posicionamiento que debe situar el automatismo encargado de obtener información de un determinado objeto almacenado utiliza un sistema de coordenadas basado en la pareja *columna-estante*.

En cada una de estas posiciones existe una etiqueta magnética con la posición correspondiente. Para ir a una posición distinta, el sistema actúa sobre unos posicionadores (uno horizontal y otro vertical) a los que se debe enviar un valor relativo respecto a la posición actual. Para ello, el sistema recibe la columna y el estante al que debe ir, lee de la etiqueta magnética la posición en la que está y envía a los posicionadores la información que necesita.

Ejemplo: si el sistema está actualmente en la columna 3, estante 4 (información que obtiene de la etiqueta magnética de la posición en la que se encuentra) y recibe la orden de ir a la columna 5, estante 3, la información que se debe enviar a los posicionadores será +2 (posicionador horizontal) y -1 (posicionador vertical).

Este problema presenta una versión muy reducida de este sistema con sólo dos columnas y dos estantes. Teniendo en cuenta todo ello:

a) Obtenga la tabla de verdad que soluciona el problema.

- Para codificar las posiciones utilice el bit más significativo para representar las columnas y el menos significativo para las filas.

- Para codificar las salidas (movimiento de cada uno de los dos posicionadores) deberá utilizar representación en complemento a 2 con el menor número de bits posible.
- b) Obtenga la implementación de la salida correspondiente al bit de signo del posicionador horizontal utilizando sólo puertas NAND.

Codificación de la Posición:

00 – Columna 0 Fila 0

01 – Columna 0 Fila 1

10 – Columna 1 Fila 0

11 – Columna 1 Fila 1

*Movimiento Posicionadores
(complemento a 2).****Hacen falta 2 bits para cada
posicionador:***

De 0 a 0 ó de 1 a 1 = 00 (0)

De 0 a 1 = 01 (+1)

De 1 a 0 = 11 (-1)

a) Tabla de verdad.

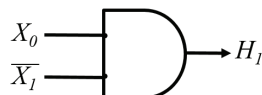
Posición Inicial		Posición Final		Posicionador Horizontal		Posicionador Vertical	
X_0	Y_0	X_1	Y_1	H1(signo)	H0	V1(signo)	V0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	1	0	0
0	0	1	1	0	1	0	1
0	1	0	0	0	0	1	1
0	1	0	1	0	0	0	0
0	1	1	0	0	1	1	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1
1	1	0	0	1	1	1	1
1	1	0	1	1	1	0	0
1	1	1	0	0	0	1	1
1	1	1	1	0	0	0	0

b) El mapa de Karnaugh y expresión de H_1 son:

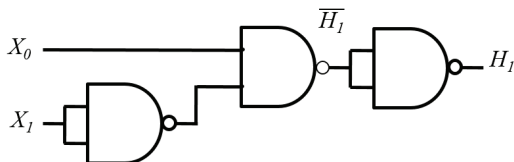
		$X_1 Y_1$			
H_1		00	01	11	10
	00	0	0	0	0
	01	0	0	0	0
$X_0 Y_0$	11	1	1	0	0
	10	1	1	0	0

$$H_1 = X_0 \bar{X}_1$$

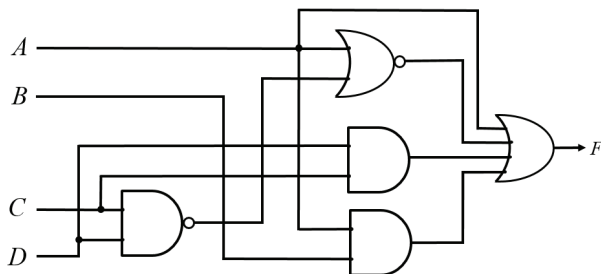
El circuito lógico de H_1 es:



que implementado con NAND es:



- 22) Una pieza de plástico es analizada en función de su resistencia ante presiones externas. Para ello se dispone de cuatro sensores de presión que llamaremos A , B , C y D . Estos sensores están diseñados de forma que, superado cierto umbral de presión, proporcionan una señal de 5 voltios. Se proporciona el circuito digital cuyas entradas son las señales que proporcionan los sensores.



- ¿Cuál es la salida F de dicho circuito digital? Realizar una simplificación de F mediante los axiomas y teoremas de Algebra de Boole y realizar una interpretación del resultado de acuerdo con el enunciado del problema.
- Usando el mapa de Karnaugh de la función, implementar F utilizando exclusivamente lo que se indica.
 - Puerta AND de 3 entradas y OR de 2 entradas en varios niveles.
 - Puertas NOR en dos niveles.

a) Analizando el circuito se llega a que:

$$\begin{aligned}
 F &= A + \overline{(A + \overline{CD})} + CD + AB = \\
 &= A + \overline{A} \cdot CD + CD + AB = \\
 &= A + CD + CD + AB = \\
 &= A + AB + CD = \\
 &= A + CD
 \end{aligned}$$

Interpretación: F se activa cuando se activa A , o bien cuando se activan C y D .

b1) Implementación de F con puertas AND de 3 entradas y OR de 2 entradas.

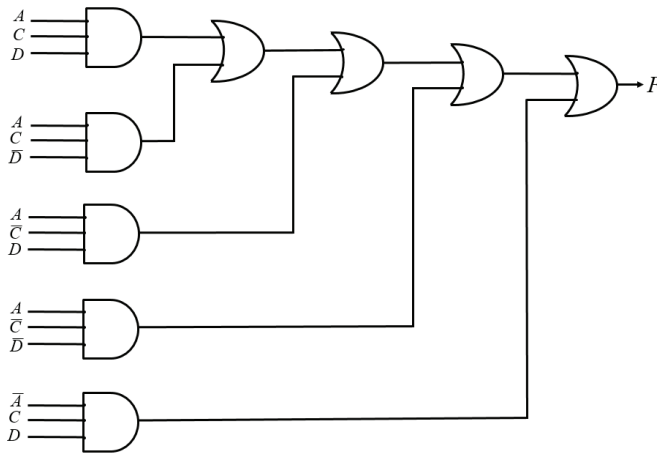
Utilizando el MK, se realizan agrupaciones de 2 unos. Cada grupo está representado por un producto de 3 literales.

		C D			
F		00	01	11	10
	00	0	0	1	0
	01	0	0	1	0
A B	11	1	1	1	1
	10	1	1	1	1

Igualmente se puede resolver de forma algebraica:

$$\begin{aligned}
 F &= A + CD = \\
 &= A(C + \overline{C})(D + \overline{D}) + CD(A + \overline{A}) = \\
 &= ACD + A\overline{C}\overline{D} + A\overline{C}D + A\overline{C}\overline{D} + \overline{A}CD
 \end{aligned}$$

El circuito es el siguiente:

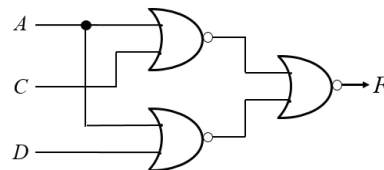


b2) Implementación con puertas NOR en 2 niveles.

Tomando agrupaciones de ceros se obtiene:

$$\begin{aligned}
 F &= (A+C)(A+D); \\
 F &= \overline{\overline{(A+C)(A+D)}} = \\
 &= \overline{(\overline{A+C}) + (\overline{A+D})}
 \end{aligned}$$

El circuito es:



23) Una empresa de ferrocarriles busca a una persona para ofrecerle un puesto de trabajo como maquinista. Para ello exige las siguientes condiciones:

- 1) Ser mayor de edad.
- 2) Tener carnet de conducir o título de mecánico.
- 3) No usar gafas.

Los candidatos tienen que rellenar una encuesta respondiendo con SI/NO a las siguientes preguntas:

- ¿Es Vd. mayor de edad?
- ¿Tiene Vd. carnet de conducir?
- ¿Tiene Vd. título de mecánico?
- ¿Usa Vd. gafas?

Diseñar un circuito combinacional usando sólo puertas NAND que decida si un candidato es admitido o no de acuerdo con la encuesta que rellena al solicitar el empleo.

NOTA: Llamar a las variables booleanas MAY, CON, MEC y GAF, y a la función ADM.

La tabla de verdad y el correspondiente mapa de Karnaugh son:

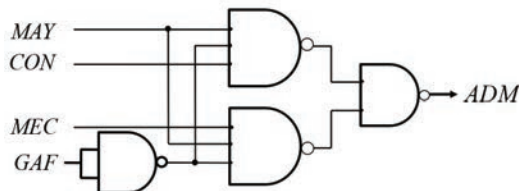
MAY	CON	MEC	GAF	ADM
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

		MEC GAF			
		00	01	11	10
ADM	00	0	0	0	0
	01	0	0	0	0
	11	1	0	0	1
	10	0	0	0	1

Agrupando unos se obtiene $ADM = MAY \cdot CON \cdot \overline{GAF} + MAY \cdot MEC \cdot \overline{GAF}$

Implementando con puertas NAND:

$$ADM = \overline{\overline{ADM}} = \overline{\overline{(MAY \cdot CON \cdot \overline{GAF})} \cdot \overline{(MAY \cdot MEC \cdot \overline{GAF})}}$$



- 24) Tres señores *A*, *B* y *C* leen libros antiguos. El Sr. *A* lee libros sobre política que estén escritos en castellano y novelas en lenguas extranjeras. El Sr. *B* lee todo tipo de libros sobre política y obras en castellano siempre que no sean novelas. El Sr. *C* lee libros que no sean novelas y estén en castellano, y libros sobre política sólo en lenguas extranjeras.

El bibliotecario tiene obligación de comunicarle a cada uno de estos señores la llegada de un libro de su gusto, pero como piensa que el enunciado del problema es muy engorroso, decide encargarle a un estudiante de Informática un circuito combinacional que le diga a qué señores hay que avisar de la llegada de un cierto libro sin más que introducir las características del mismo.

Determinar las variables de entrada y funciones booleanas simplificadas que resuelven el problema, realizando la tabla de verdad e implementar el circuito con puertas NAND.

Variables de entrada:

P: Política (P=1 Política; P=0 No política)

I: Idioma (I=1 Castellano; I=0 Extranjero)

N: Novela (N=1 Novela; N=0 No novela)

Funciones de salida:

A: (A=1 Le interesa; A=0 No le interesa)

B: (B=1 Le interesa; B=0 No le interesa)

C: (C=1 Le interesa; C=0 No le interesa)

Tabla de verdad

P	I	N	A	B	C
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	0

Implementación con NAND:

		I N			
	A	00	01	11	10
P	0	0	1	0	0
	1	0	1	1	1

$$A = \bar{I}N + PI = \overline{\overline{\bar{I}N} \cdot \overline{PI}}$$

		I N			
	B	00	01	11	10
P	0	0	0	0	1
	1	1	1	1	1

$$B = P + \bar{I}\bar{N} = \overline{\overline{P} \cdot \overline{\bar{I}\bar{N}}}$$

		I N			
	C	00	01	11	10
P	0	0	0	0	1
	1	1	1	0	1

$$C = \bar{P}\bar{I} + I\bar{N} = \overline{\overline{\bar{P}\bar{I}} \cdot \overline{I\bar{N}}}$$

TEST Y PREGUNTAS CORTAS

1) Responde verdadero o falso a las siguientes preguntas:

- ☐ V ☐ F Si $x \oplus y = z$, entonces $x \oplus y \oplus z = 0$
- ☐ V ☐ F Para que la salida de una puerta NOR sea 0 es necesario que todas las entradas sean 1.
- ☐ V ☐ F Para que la salida de una puerta NAND sea 0 es necesario que todas las entradas sean 1.

2) Responde verdadero o falso a las siguientes preguntas:

- ☐ V ☐ F Es cierto que $\bar{x} \oplus y = \overline{x \oplus y}$
- ☐ V ☐ F Una puerta XOR con un 1 en una de sus entradas actúa como un inversor de la otra entrada.
- ☐ V ☐ F Una puerta XNOR con un 1 en una de sus entradas actúa como un inversor.

3) Responde verdadero o falso a las siguientes preguntas:

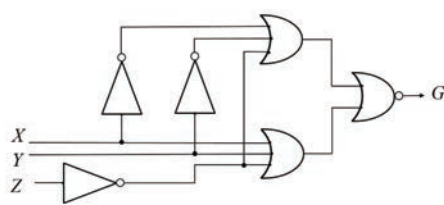
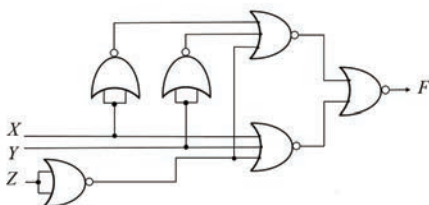
- ☐ V ☐ F Es cierto que $\bar{x} \oplus \bar{y} = x \oplus y$
- ☐ V ☐ F Si $x \oplus y = z$, entonces $x \oplus y \oplus z = 1$
- ☐ V ☐ F El número -65 es representable en C a 2 con 7 bits.

4) Responde verdadero o falso a las siguientes preguntas:

- ☐ V ☐ F Para que la salida de una puerta NAND sea 0 es necesario que todas las entradas sean 1.
- ☐ V ☐ F Un generador de funciones se usa para analizar circuitos digitales.
- ☐ V ☐ F La representación sin signo codifica menos números que la representación signo-magnitud con el mismo número de bits.

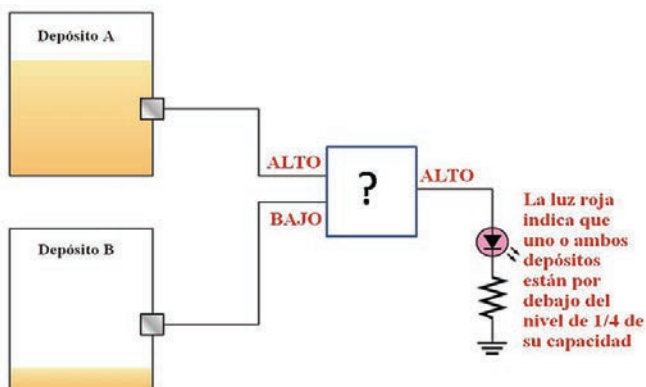
5) Los circuitos de la figura adjunta representan dos circuitos con esquemas NOR (a la izquierda) y OR/AND (derecha). Señala la opción correcta.

- ☐ A $F = \bar{G}$
- ☐ B $F = G$
- ☐ C Ninguna de las anteriores.



6) El circuito que falta en la figura es:

- ☐ A Una puerta XOR.
☐ B Una puerta NOR.
☐ C Una puerta NAND.



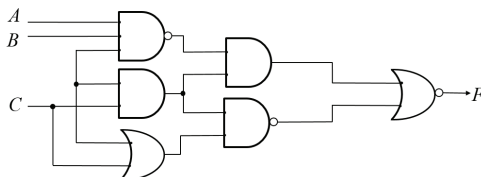
7) Señala lo falso.

- ☐ A $A + CD = (A + C)(A + D)$
☐ B $A \oplus 1 = A$
☐ C $(A \oplus B) \oplus C = (C \oplus B) \oplus A$

8) ¿Cuál de las siguientes afirmaciones es falsa?

- ☐ A En una puerta NAND de dos variables solo se da un único caso en que la salida es 0.
☐ B Las puertas XOR y XNOR no se pueden implementar con puertas NOR.
☐ C $\overline{A \oplus 1} = A \oplus 0$

9) La función del circuito de la figura es:



- ☐ A $F = \overline{\overline{A \cdot B \cdot C} \cdot B \cdot C + \overline{B \cdot C} \cdot (A + C)}$
☐ B $F = \overline{\overline{A \cdot B \cdot C} \cdot B \cdot C + \overline{B \cdot C} \cdot (B + C)}$
☐ C Ninguna de las anteriores.

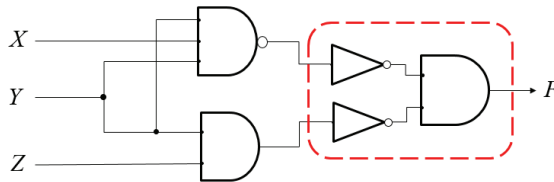
10) Una puerta XNOR con una de sus entradas fija a 1 se comporta como:

- ☐ A Un inversor.
- ☐ B Una XOR con una de sus entradas a 0.
- ☐ C Las dos anteriores son falsas.

11) En la implementación de circuitos lógicos se puede usar un único tipo de puertas:

- ☐ A Siempre que sean AND o bien OR.
- ☐ B Hay que usar puertas NAND y puertas NOR (ambas).
- ☐ C Ambas afirmaciones son falsas.

12) En el circuito de la Figura, la zona marcada con un rectángulo se podría sustituir, realizando la misma función, por:



- ☐ A Una puerta NOR.
- ☐ B Una puerta OR.
- ☐ C Una puerta NAND.

13) ¿Cuál de las siguientes afirmaciones es falsa?

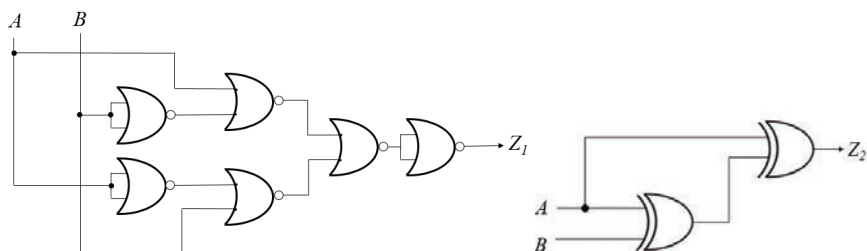
- ☐ A Toda función booleana se puede expresar como producto de sumas.
- ☐ B Algunas funciones booleanas se pueden expresar como producto de algunas de sus variables (afirmadas o complementadas).
- ☐ C Todas las funciones booleanas se pueden implementar utilizando sólo puertas XOR y XNOR.

14) Señala lo que consideres falso:

- ☐ A Una puerta NAND puede ser implementada a base puertas NOR
- ☐ B Una puerta XOR con una de sus entradas a 1 actúa como un inversor
- ☐ C Cualquier circuito puede realizarse con inversores y sólo puertas de uno de los siguientes tipos: AND/OR.

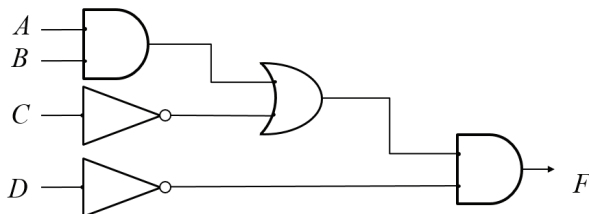
15) Para los dos circuitos de la figura señala lo que consideres falso:

- ☐ A $Z_1 = A \oplus B$
- ☐ B $Z_2 = B$
- ☐ C $Z_1 = Z_2$



- 16) Si cada una de las puertas lógicas de dos entradas tiene un retardo de 3 ns, y los inversores tienen un retardo de 2 ns, ¿cuál es la máxima frecuencia a la que pueden cambiar las entradas para que el circuito de la Figura funcione correctamente?

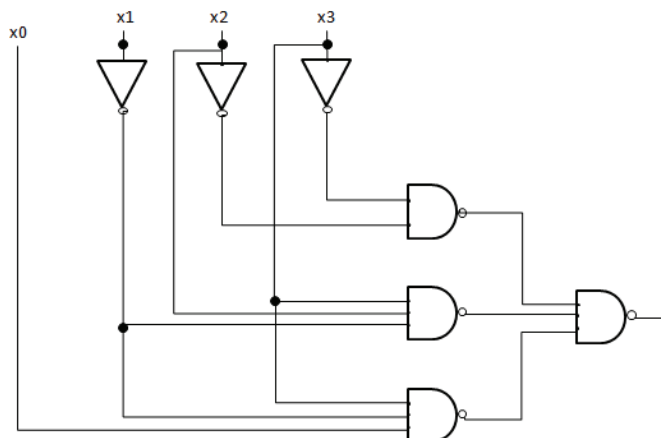
- ☐ A 90 MHz
☐ B 100 MHz
☐ C 111 MHz



- 17) La aplicación de la ley de Morgan permite:

- ☐ A Pasar una función expresada como suma de productos a producto de sumas.
☐ B Pasar una función expresada con puertas NAND a puertas NOR.
☐ C Pasar una función expresada como suma de productos a puertas NAND.

- 18) ¿A qué función corresponde la implementación que se ve en el siguiente circuito?
 Suponer x_3 como variable más significativa.



☐ A $f(x_3, x_2, x_1, x_0) = \sum m(0, 1, 2, 3, 9, 12, 13)$

☐ B $f(x_3, x_2, x_1, x_0) = (x_3 + \bar{x}_2)(\bar{x}_3 + \bar{x}_1)(\bar{x}_3 + x_2 + x_0)$

☐ C Ambas son correctas.

4

SISTEMAS COMBINACIONALES: MÓDULOS

CONTENIDOS:

- Análisis y diseño con módulos combinacionales.
- Análisis y diseño con redes de módulos combinacionales.
- Circuitos completos combinacionales.

EJERCICIOS RESUELTOS

- 1) Encontrar la forma más simplificada en suma de productos de la función $F = f \cdot g$, donde f y g están dadas por:

$$f = wx\bar{y} + \bar{y}z + \bar{w}y\bar{z} + x\bar{y}z$$

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

- a) Obtenida la función F simplificada, implementarla exclusivamente con puertas NOR en dos niveles y realizar un esquema lógico.
 b) Implementar F con una red de decodificadores de dos entradas en forma de árbol. Utilizar a lo sumo una puerta OR.

- a) Utilizando los mapas de Karnaugh de las funciones f y g se forma el mapa de Karnaugh de la función producto y se agrupan los '1' adyacentes. Los '1' comunes en f y g han sido señalados.

		yz			
	f	00	01	11	10
wx	00		1		1
	01		1		1
	11	1	1		
	10		1		1

		yz			
	g	00	01	11	10
wx	00	1	1		1
	01	1	1	1	
	11	1		1	
	10	1		1	1

		yz			
	f.g	00	01	11	10
wx	00		1		1
	01		1		
	11	1			
	10				1

$$F = f \cdot g = wx\bar{y}z + \bar{w}y\bar{z} + x\bar{y}z$$

Expresamos F como producto de sumas.

		y ₂			
	F	00	01	11	10
wx	00	0	1	0	1
	01	0	1	0	0
	11	1	0	0	0
	10	0	0	0	1

Agrupando ceros se obtiene $\bar{F}$:

$$\bar{F} = yz + xy + wz + \bar{w}\bar{y}\bar{z} + w\bar{x}\bar{y}$$

Negando la negada:

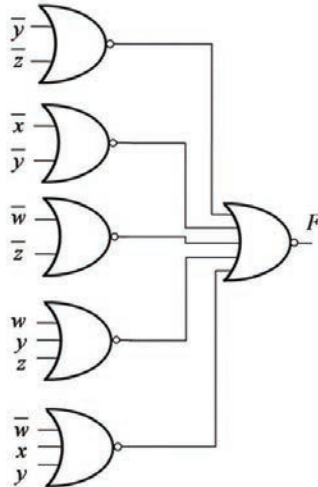
$$F = \overline{yz \cdot xy \cdot wz \cdot \bar{w}\bar{y}\bar{z} \cdot w\bar{x}\bar{y}}$$

Aplicando De Morgan:

$$F = (\bar{y} + \bar{z})(\bar{x} + \bar{y})(\bar{w} + \bar{z})(w + y + z)(\bar{w} + x + y)$$

Agrupando ceros directamente se tiene:

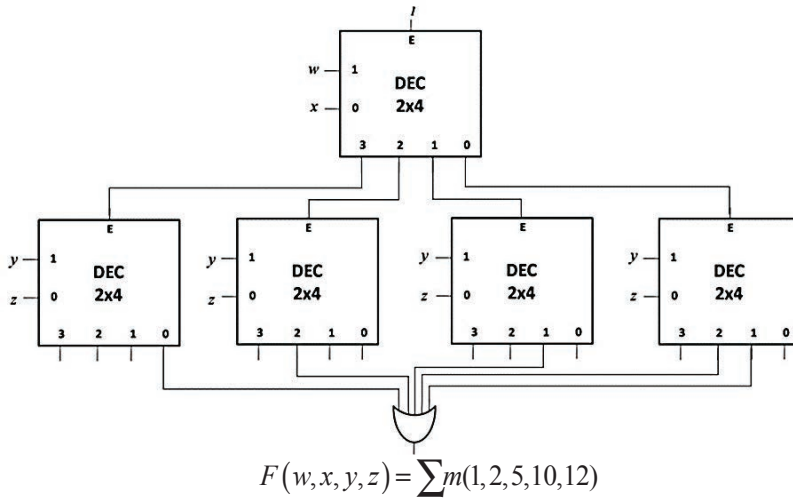
$$F = \bar{F} = \overline{(\bar{y} + \bar{z}) + (\bar{x} + \bar{y}) + (\bar{w} + \bar{z}) + (w + y + z) + (\bar{w} + x + y)}$$



b) La expresión de F en su primera forma canónica es:

$$F(w, x, y, z) = \sum m(1, 2, 5, 10, 12)$$

Dividimos en dos niveles de variables w, x para el primer nivel y y, z para el segundo nivel, resultando:

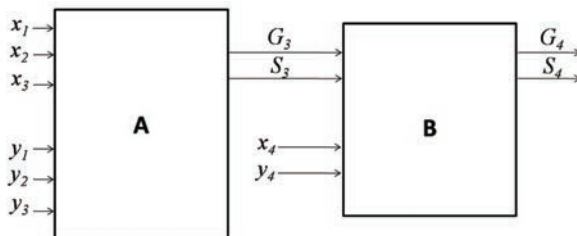


2) El circuito integrado A compara magnitudes de dos números de tres bits, $X_3 = x_1x_2x_3$ y $Y_3 = y_1y_2y_3$, donde x_3 e y_3 son los bits menos significativos. Tienen dos salidas G_3 y S_3 tales que

- $G_3 = 1, S_3 = 0$ si y sólo si $X_3 > Y_3$
- $S_3 = 1, G_3 = 0$ si y sólo si $X_3 < Y_3$
- $G_3 = S_3 = 0$ si y sólo si $X_3 = Y_3$.

Se pide:

- Diseñar un circuito lógico B de manera que en conjunción con el circuito A realice la comparación de dos números de 4 bits, $X_4 = x_1x_2x_3x_4$ y $Y_4 = y_1y_2y_3y_4$ tal como se representa en la figura. Encontrar las expresiones lógicas simplificadas de G_4 y S_4 , que funcionan con el mismo criterio que G_3 y S_3 , en función de las entradas de la unidad B y dar una realización con puertas NAND.
- Diseñar el circuito más sencillo posible de B utilizando los siguientes módulos: decodificadores de dos entradas y multiplexores de cuatro canales de entrada.
- Proporcionar un diseño del circuito A utilizando únicamente unidades de tipo B. Suponer disponibles constantes lógicas 1 y 0.



- a) Se diseña el circuito B a través de su tabla de verdad y se obtiene su expresión simplificada como suma de productos en mapas de Karnaugh:

G_3	S_3	x_4	y_4	G_4	S_4
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X

		$x_4 y_4$			
		00	01	11	10
$G_3 S_3$	00	0	0	0	1
	01	0	0	0	0
	11	X	X	X	X
	10	1	1	1	1

$$G_4 = G_3 + \bar{S}_3 x_4 \bar{y}_4$$

		$x_4 y_4$			
		00	01	11	10
$G_3 S_3$	00	0	1	0	0
	01	1	1	1	1
	11	X	X	X	X
	10	0	0	0	0

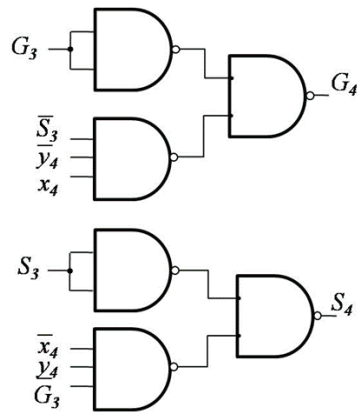
$$S_4 = S_3 + \bar{x}_4 y_4 \bar{G}_3$$

La simplificación como suma de productos nos lleva a una implementación NAND, como se demuestra a continuación.

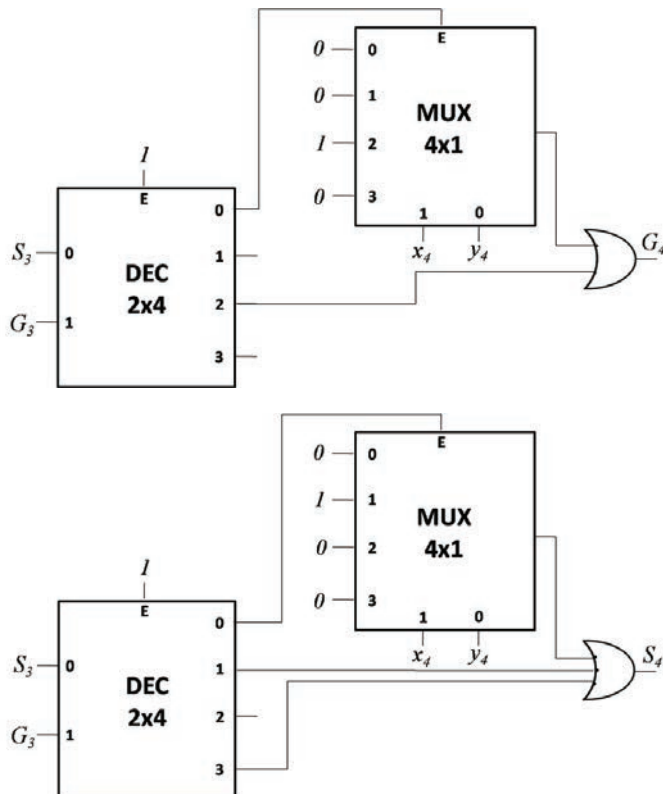
$$G_4 = G_3 + \bar{S}_3 x_4 \bar{y}_4 = \overline{\overline{G_3 + \bar{S}_3 x_4 \bar{y}_4}} = \overline{\overline{G_3} \cdot \overline{\bar{S}_3 x_4 \bar{y}_4}} = \overline{\overline{G_3} \cdot S_3 x_4 y_4}$$

$$S_4 = S_3 + \bar{x}_4 y_4 \bar{G}_3 = \overline{\overline{S_3 + \bar{x}_4 y_4 \bar{G}_3}} = \overline{\overline{S_3} \cdot \overline{\bar{x}_4 y_4 \bar{G}_3}} = \overline{\overline{S_3} \cdot x_4 y_4 G_3}$$

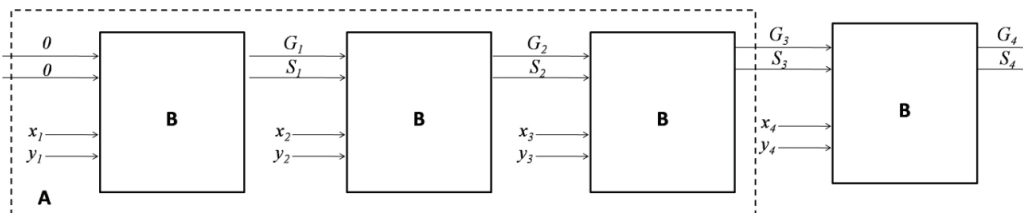
El circuito digital es el siguiente:



- b) En principio se tomarán un DEC 2x4 en el primer nivel y cuatro MUX 4x1 en el segundo nivel para cada función. Sin embargo, como se aprecia en la tabla de verdad varios de ellos pueden ser desechados al ser todas sus entradas las constantes 0 ó 1. En G_4 tomamos las condiciones de “no importa” como ceros y en S_4 como unos.



c) Diseño del circuito A en función de circuitos B.



3) Efectuar la síntesis de un circuito combinacional de cuatro entradas A , B , C y D y una salida F . La salida F toma valor 1 cuando la configuración decimal de que se presenta a la entrada (tómese A como la variable de más peso) es divisible por 4 ó 5.

- Determinar la expresión de F simplificada.
- Implementar F exclusivamente con puertas NOR.
- Implementar F con multiplexores de cuatro entradas y decodificadores de dos entradas.

a) La expresión de la función F como suma de minitérminos es:

$$F = \sum m(0, 4, 5, 8, 10, 12, 15)$$

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

La expresión simplificada es:

$$F = \overline{C} \overline{D} + \overline{A} \overline{B} \overline{C} + \overline{A} \overline{B} \overline{D} + ABCD$$

b) Implementación con NOR

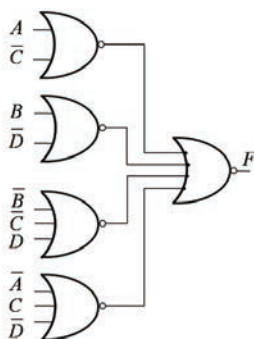
		CD			
	F	00	01	11	10
AB	00	1	0	0	0
	01	1	1	0	0
	11	1	0	1	0
	10	1	0	0	1

Agrupando ceros:

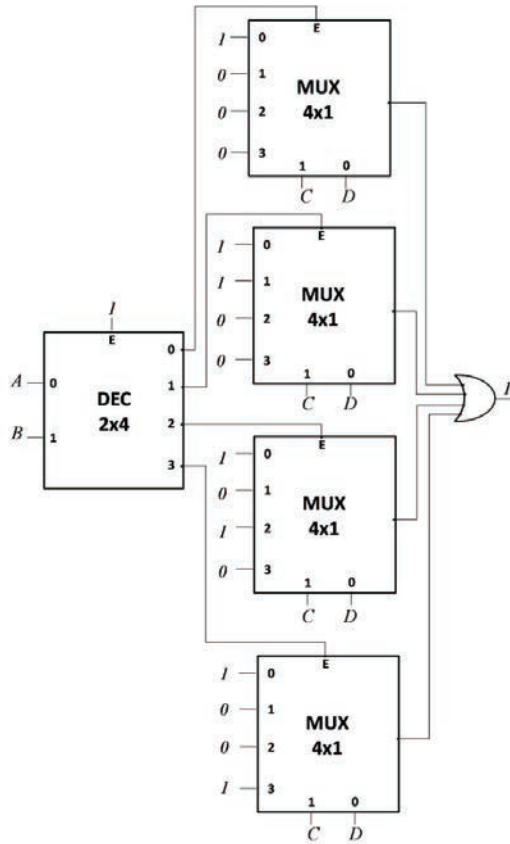
$$\bar{F} = \bar{A}\bar{C} + B\bar{C}\bar{D} + \bar{B}\bar{D} + A\bar{C}\bar{D}$$

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

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

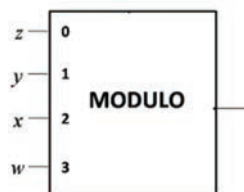


- c) Teniendo en cuenta la primera forma canónica de F , implementamos con una red de decodificadores y multiplexores en dos niveles.



- 4) Se dispone de forma integrada de un circuito combinacional de 4 entradas que denominaremos **MODULO** y que realiza la siguiente función booleana:

$$\text{MODULO}(w, x, y, z) = xy'(w + z)$$



Suponiendo que las variables de entrada están disponibles en formas complementada y no complementada, se pide:

- a) Realizar el diseño de la función $f(w, x, y, z) = \sum m(0, 1, 6, 9, 10, 11, 14, 15)$ utilizando tres MODULOS y una puerta OR.
- b) ¿Pueden sintetizarse cualquier función booleana utilizando únicamente MODULOS y puertas OR? Razonar la respuesta.
-

a) La función MODULO tiene una expresión SP (suma de productos) siguiente:

$$\text{MODULO}(w, x, y, z) = xy(w + z) = xyw + xyz$$

y la apariencia en MK es:

		yz			
		00	01	11	10
M	00	0	0	0	0
	01	0	0	1	0
	11	0	0	1	1
	10	0	0	0	0
wx					

La estrategia a seguir será realizar agrupaciones tipo MODULO en el MK de la función f .

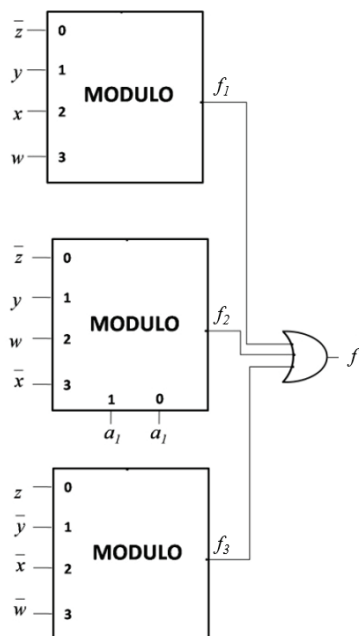
		yz			
		00	01	11	10
f	00	1	1	0	0
	01	0	0	0	1
	11	0	0	1	1
	10	0	1	1	1
wx					

$$f = f_1 + f_2 + f_3$$

$$f_1 = wxy + xy\bar{z} = xy(w + \bar{z}) = M(w, x, y, \bar{z})$$

$$f_2 = w\bar{x}y + wy\bar{z} = wy(\bar{x} + \bar{z}) = M(\bar{x}, w, y, \bar{z})$$

$$f_3 = \bar{w}\bar{x}\bar{y} + \bar{x}\bar{y}z = \bar{x}\bar{y}(\bar{w} + z) = M(\bar{w}, \bar{x}, \bar{y}, z)$$

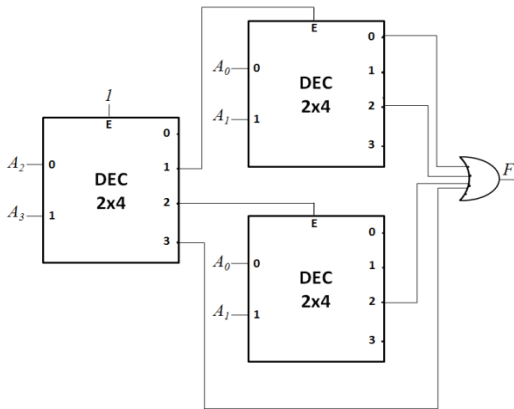


b) No se puede implementar cualquier función con MODULOS, por ejemplo, la función $g(w, x, y, z) = wxyz$

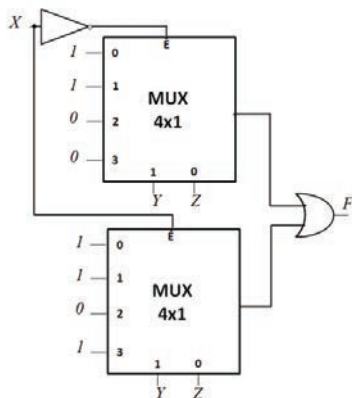
5) Implementa la función F con varios DEC 2x4 en red de dos niveles con salida activa a nivel alto cuya tabla de verdad es la siguiente:

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

Se realiza una implementación en árbol con decodificadores 2x4, estando las variables de más peso en el primer nivel. La salida 0 del decodificador de primer nivel no se utiliza al ser $F=0$ en las cuatro primeras filas de la tabla de verdad. Igualmente, la salida 3 del mismo se conecta directamente a la puerta OR debido a que $F=1$ para las últimas cuatro filas.



6) Obtener la expresión de la función de la figura $F(X,Y,Z)$ como producto de sumas.



Analizando el circuito, podemos hallar el valor de F directamente rellenar el mapa de Karnaugh para todos los valores de las variables de entrada X, Y, Z .

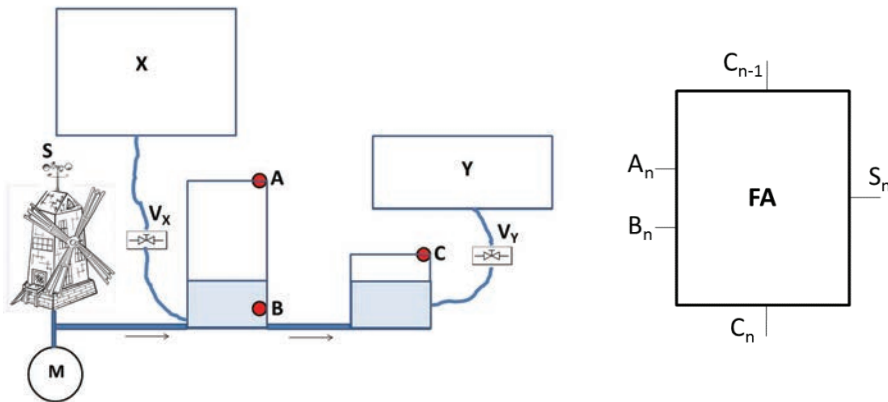
		Y	
		0	1
X	0	1	1
	1	1	0
		Z	
		0	1

Agrupando dos conjuntos de dos ceros se genera la expresión como producto de sumas.

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

7) La figura adjunta (parte izquierda) representa un sistema de regadío de dos parcelas X e Y . Para bombear el agua se cuenta con un molino de viento y un motor M . El agua es bombeada y almacenada en dos depósitos X e Y , uno para cada parcela, que están comunicados por abajo y al mismo nivel del suelo. El primer depósito mide el doble que el segundo. El motor M y las válvulas de los aspersores conectados a los depósitos (que llamaremos V_X e V_Y) se abren en función de los sensores de nivel A, B y C de los depósitos así como de un sensor que mide si hay viento suficiente para que el molino extraiga agua. Si la velocidad del molino es superior a 10 vueltas por segundo, un sensor S queda activado.

El motor M se pone en marcha siempre que no haya suficiente viento y los dos depósitos no estén llenos. La parcela X se riega siempre que su depósito esté lleno y no haya viento suficiente o bien, si hay viento suficiente y su depósito está al menos medio lleno. Por su parte, la parcela Y sólo se riega cuando hay viento suficiente y su depósito supera el nivel mínimo marcado por el sensor de nivel B .



a)

- Determinar las variables de entrada y salida del sistema digital que controla el regadío de las dos parcelas.
- Establecer claramente cuáles son las situaciones de “no importa” en este problema.
- Plantear la tabla de verdad del sistema digital.

b)

- Implementar la función de activación del motor M con un sólo MUX 4x1 y puertas NAND.
- Implementar la función de apertura de la válvula V_X con varios decodificadores 2x4 dispuestos en dos niveles (estructura de árbol) y alguna puerta lógica. Proporcionar el circuito optimizado con el mínimo número de módulos necesarios.
- Implementar la función de apertura de la válvula V_Y con DEC 2x4, MUX 4x1 y alguna puerta lógica. Implementar el circuito optimizado con el mínimo número de módulos necesarios.

- c) El dueño de la finca tiene tres sistemas como el descrito que permiten regar un total de seis parcelas. Para controlar el consumo de agua necesita diseñar un circuito digital en el que se dispare una alarma F cuando haya tres o más parcelas regándose a la vez (recuerda que la parcela se riega cuando la válvula correspondiente se abre). Diseñar este circuito contando exclusivamente con sumadores completos (ver figura a la derecha) y un multiplexor deseado. Numerar las seis válvulas genéricamente como $V_0, V_1, \dots, V_5$. Notar que las salidas

de un FA proporcionan el número de unos de los tres bits de entrada, siendo C_n más significativo que S_n .

a) Variables del sistema y tabla de verdad.

Según el enunciado del problema se concluye que existen 4 entradas: S (sensor de velocidad del viento), A (sensor de nivel del depósito grande), B (sensor de nivel del depósito grande) y C (sensor de nivel del depósito pequeño).

Por su parte las variables de salida son M (motor), V_x (válvula del depósito grande) y V_y (válvula del motor pequeño).

Existen condiciones de “no importa” para incongruencias en las medidas de los sensores de nivel que son reflejadas en la tabla de verdad. Estas incongruencias son:

- Si $A=1$, B y C deben ser 1, por tanto, no puede ocurrir que:

$$A=1, C=0$$

$$A=1, B=0$$

- Si $C=1$, B debe ser 1, por tanto, no puede ocurrir que:

$$C=1, B=0$$

S	A	B	C	M	V _x	V _y
0	0	0	0	1	0	0
0	0	0	1	X	X	X
0	0	1	0	1	0	0
0	0	1	1	0	0	0
0	1	0	0	X	X	X
0	1	0	1	X	X	X
0	1	1	0	X	X	X
0	1	1	1	0	1	0
1	0	0	0	0	0	0
1	0	0	1	X	X	X
1	0	1	0	0	0	1
1	0	1	1	0	1	1
1	1	0	0	X	X	X
1	1	0	1	X	X	X
1	1	1	0	X	X	X
1	1	1	1	0	1	1

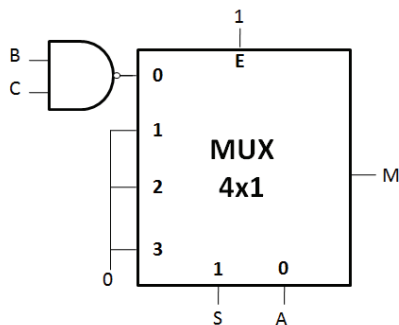
Condiciones de NI

$$A=1, C=0$$

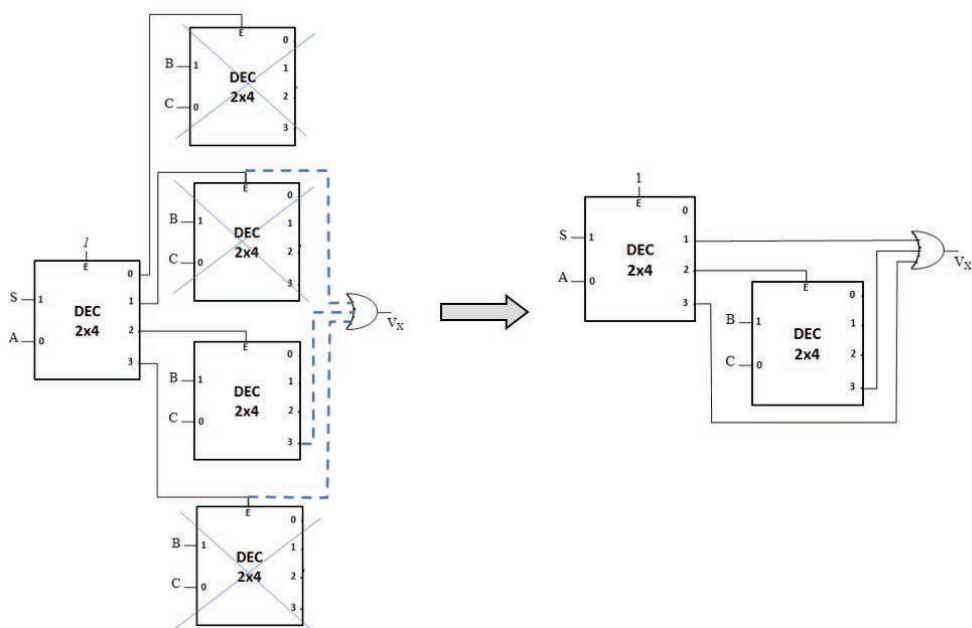
$$A=1, B=0$$

$$C=1, B=0$$

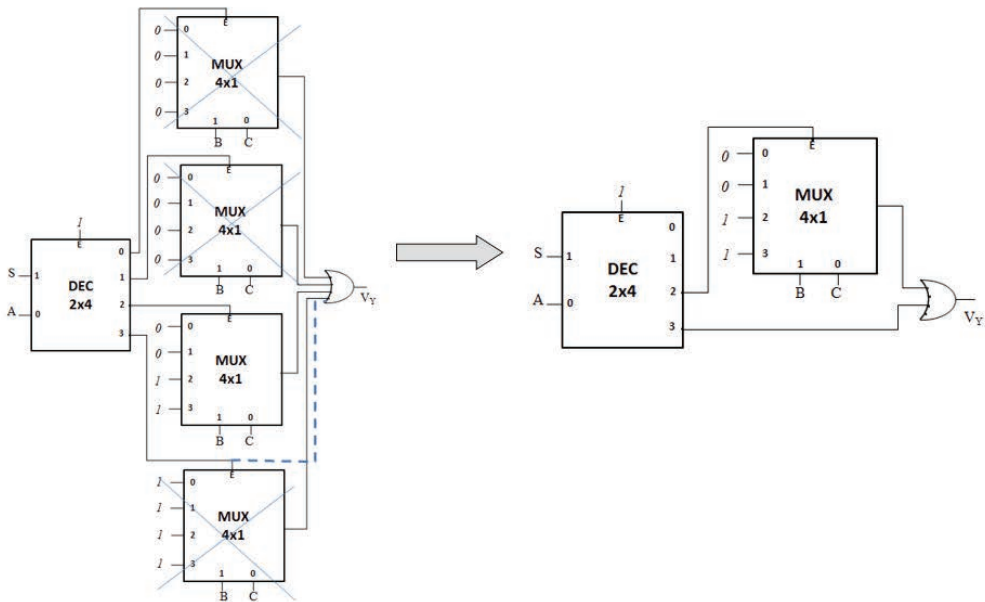
b1) Implementación de M con un MUX 4x1. Se toman por comodidad S y A como entradas de selección del multiplexor.



b2) La implementación con árbol de decodificadores 2x4 queda simplificada al evitar tres de ellos en el segundo nivel, quedando:



b3) En la implementación con DEC 2x4 y MUX 4x1 en árbol se optimiza el diseño con el mínimo número de módulos necesarios, no siendo necesarios tres de los cuatro MUX en el segundo nivel.

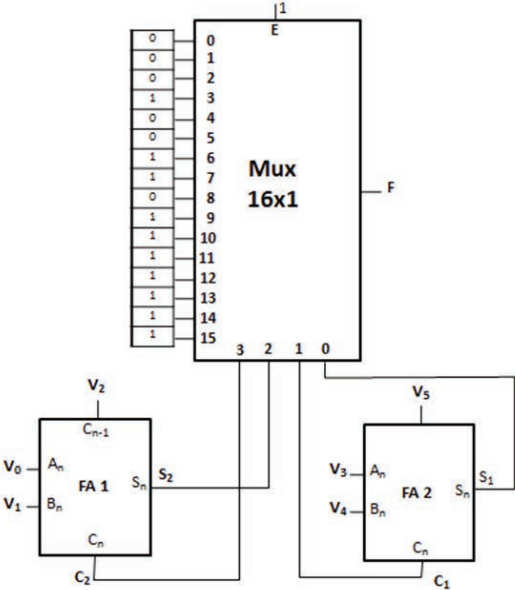


c) Implementación de la función de alarma

Ya que las salidas de un FA proporcionan el número de unos de los tres bits de entrada, repartimos las salidas de las seis válvulas como entradas a los dos FA. Las salidas del primer FA, C_2 y S_2 contabilizarán en número de unos que proporcionan las tres primeras válvulas. Igualmente, las salidas C_1 y S_1 contabilizarán el número de unos que proporcionan las tres últimas válvulas. Según esta especificación, la tabla de verdad de la función de alarma F es clara.

La implementación se realiza fácilmente con un MUX 16x1 introduciendo las variables C_2, S_2, C_1, S_1 por las entradas de selección (en este orden) y el valor de la función F por los canales de entrada.

C ₂	S ₂	C ₁	S ₁	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	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



8) Se desea diseñar un conversor BCD a display de 7 segmentos (Figura 1) usando puertas NAND. Considerar que *A* es la variable más significativa del código BCD y que el segmento se enciende cuando se le aplica un 1 lógico. Tener en cuenta ABCD siempre será un código BCD válido.



Figura 1

El display se verá a través de un espejo, con lo cual los segmentos que se deberán encender no son los mismos que si el display se viera directamente. La selección de segmentos que deberán encenderse aparece en las figuras adjuntas, dependiendo que el espejo se coloque de forma horizontal (Figura 2) o de forma vertical (Figura 3).



Figura 2



Figura 3

Se trata de estudiar cuál de las dos formas de colocar el cristal produce un diseño más sencillo, para lo cual se pide:

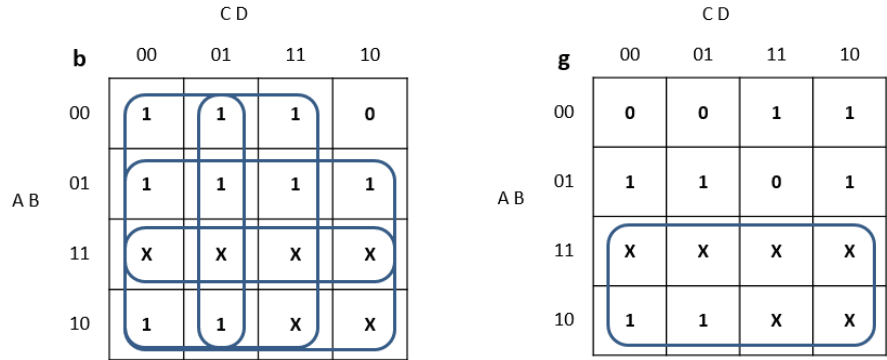
- Realizar la tabla de verdad de los 7 segmentos en cada uno de los dos casos.
- Hacer los Mapas de Karnaugh de las funciones b y g de ambos casos marcando sólo las agrupaciones de 8 unos que sean posibles, diciendo cuál sería el diseño más sencillo para esas dos funciones en ambos casos.
- Implementar c horizontal con un MUX 8x1 y f vertical con un MUX 4x1.

- Tabla de verdad

				Espejo Horizontal								Espejo Vertical							
A	B	C	D	a	b	c	d	e	f	g	a	b	c	d	e	f	g		
0	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0		
0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	1	0		
0	0	1	0	1	0	1	1	0	1	1	1	0	1	1	0	1	1		
0	0	1	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1		
0	1	0	0	0	1	1	0	1	0	1	0	1	0	0	1	1	1		
0	1	0	1	1	1	0	1	1	0	1	1	1	0	1	1	0	1		
0	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1		
0	1	1	1	0	1	1	1	0	0	0	1	0	0	0	1	1	0		
1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
1	0	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1		
1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
1	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
1	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X		

b)

Mapas de Karnaugh: Horizontal (de la “b” y la “g”). Se marcan agrupaciones de ocho unos



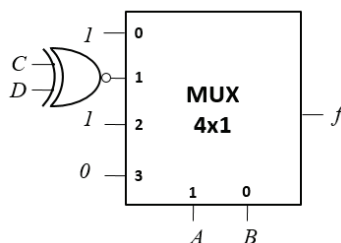
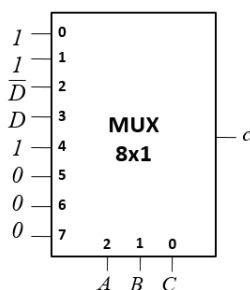
		C D			
		00	01	11	10
A B	00	1	0	0	0
	01	1	1	0	1
	11	X	X	X	X
	10	1	1	X	X

		C D			
		00	01	11	10
A B	00	0	0	1	1
	01	1	1	0	1
	11	X	X	X	X
	10	1	1	X	X

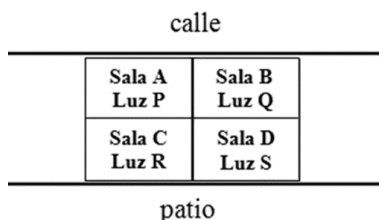
Atendiendo al número de agrupaciones de ocho unos la implementación horizontal sería la más sencilla para la función b . Para la función g es indistinto.

c) Una de las posibles implementaciones de c con MUX 8x1 y f con MUX 4x1 es:

Se toman todas las condiciones de no-importa como ceros.



- 9) Se pretende diseñar un sistema digital combinacional para encender las luces de las salas de un despacho de abogados, según la figura adjunta. El despacho tiene dos salas que dan a la calle (A y B), y otras dos que dan a un patio (C y D). Cada sala tiene un sensor de que se activa, poniéndose a 1, si hay alguien en dicha sala (y mientras esté) y se desactiva, poniéndose a 0, cuando no hay nadie. Por su parte, la luz correspondiente estará apagada si la función booleana es 0 y encendida si es 1.



El problema a resolver tiene las siguientes especificaciones:

- Si no hay nadie en ninguna sala, todas las luces estarán apagadas.
- Si hay alguien en alguna de las dos salas que dan a la calle (o ambas), y nadie en las otras dos (patio), se encienden las luces de las dos salas que dan a la calle.
- Si hay alguien en alguna de las dos salas que dan al patio (o ambas), y nadie en las que dan a la calle, se enciende la luz de la sala en la que haya alguien.
- Si hay tres salas ocupadas y dos de ellas dan a la calle, se encienden las luces de las cuatro salas.
- Si hay tres salas ocupadas y dos de ellas dan al patio, se encienden las luces de las tres salas ocupadas.
- Nunca hay sólo dos salas ocupadas, dando una a la calle y la otra al patio.
- Nunca están las cuatro salas ocupadas.
- Para los casos no previstos expresamente, si los hubiera, se mantendrán todas las luces apagadas.

Diseñar el circuito que resuelve este problema siguiendo los siguientes puntos:

- a) Tabla de verdad.
- b) Implementar la función P con un DEC 3x8 con salidas activas a nivel bajo.
- c) Implementar la función Q con un MUX 8x1.
- d) Implementar la función R usando sólo puertas NAND.
- e) Implementar la función S usando sólo puertas NOR.

a)

Atendiendo a las especificaciones dadas en el enunciado se detectan varios casos de “no importa”.

De la frase “Nunca hay sólo dos salas ocupadas, dando una a la calle y la otra al patio” se deducen los siguientes casos de “no importa”. ABCD = 0101, 0110, 1001, 1010.

De la frase “Nunca están las cuatro salas ocupadas” se deduce la condición de “no importa” ABCD=1111

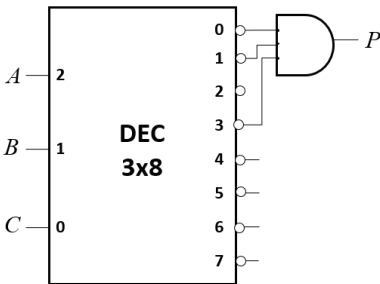
				Solución principal				Solución alternativa (*)			
A	B	C	D	P	Q	R	S	P	Q	R	S
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1	0	0	0	0
0	1	0	0	1	1	0	0	1	1	0	0
0	1	0	1	X	X	X	X	X	X	X	X
0	1	1	0	X	X	X	X	X	X	X	X
0	1	1	1	0	1	1	1	0	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0
1	0	0	1	X	X	X	X	X	X	X	X
1	0	1	0	X	X	X	X	X	X	X	X
1	0	1	1	1	0	1	1	1	0	1	1
1	1	0	0	1	1	0	0	0	0	0	0
1	1	0	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1
1	1	1	1	X	X	X	X	X	X	X	X

(*) La solución alternativa se ha dado también como válida. Sólo discrepa en los casos marcados de otro color y está basada en excluir implícitamente la opción “ambas” cuando dice “en alguna de las dos salas que dan a calle/patio... y nadie en las otras dos”, es decir, se ha tomado “alguna de las dos salas” estrictamente en singular. Los casos distintos a la solución principal estarían englobados en el apartado de “casos no previstos expresamente”.

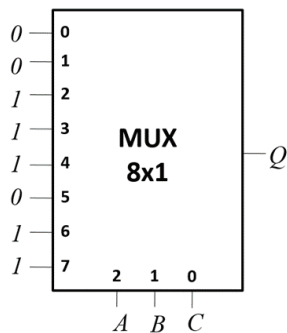
b) Implementación de *P* con un DEC3x8 con salida invertida.

Los circuitos que se presentan a continuación están hechos para la solución principal.

De la tabla de verdad, se puede ver que la función *P* depende únicamente de las variables *A*, *B* y *C* si se toma la condición no importa número 6 como 0 y el resto como unos. Al implementar con DEC con salida invertida podemos usar una puerta NAND en caso de tomar unos o AND en caso de tomar ceros. En este caso, es más sencillo tomar ceros.



c) Implementación de Q con un MUX 8x1



d) Función R con puertas NAND

		C D			
	R	00	01	11	10
A B	00	0	0	1	1
	01	0	X	1	X
	11	0	1	X	1
	10	0	X	1	X

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

e) Función S con puertas NOR

		C D			
	S	00	01	11	10
A B	00	0	1	1	0
	01	0	X	1	X
	11	0	1	X	1
	10	0	X	1	X

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

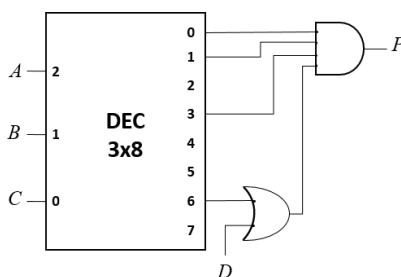
Los circuitos que se presentan a continuación están hechos para la solución alternativa.

b) Función S :

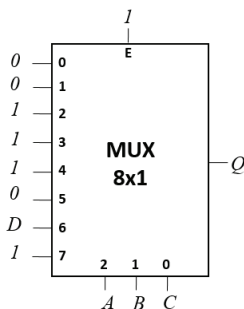
Tomando los mismos valores para las condiciones de “no importa” como en el caso de la solución principal, se llega a que la segunda forma canónica de la función P , que puede ser expresada como:

$$S = \prod M_{(A,B,C)}(0,1,3) \cdot M_{(A,B,C,D)}(12) = \prod M_{(A,B,C)}(0,1,3) \cdot [M_{(A,B,C)}(6) + D]$$

$$\text{ya que } S = M_{(A,B,C,D)}(12) = \bar{A} + \bar{B} + C + D = (\bar{A} + \bar{B} + C) + D = M_{(A,B,C)}(6) + D$$



c) Función Q



Se omiten los Mapas de Karnaugh de la solución alternativa.

d) $R = C \cdot \bar{D} + B \cdot D + A \cdot D = \overline{\overline{C \cdot \bar{D} + B \cdot D + A \cdot D}}$ o bien, tomando otra agrupación de 4 unos posible en vez una de las tomadas, $R = C \cdot \bar{D} + B \cdot D + A \cdot C = \overline{\overline{C \cdot \bar{D} + B \cdot D + A \cdot C}}$

e) $S = (C + D) \cdot (A + B + \bar{C}) = \overline{\overline{(C + D)} + \overline{(A + B + \bar{C})}}$. En este caso, no hay otra solución posible.

- 10) Diseñar un circuito combinacional para resolver el siguiente problema de diseño lógico. El Presidente de un club de fútbol debe decidir al final de la temporada si renovar ($F=1$) o no ($F=0$) el contrato al entrenador. Para ello, tiene que valorar varios aspectos. Si el equipo queda campeón y obtiene más de 70 puntos, renueva contrato, pero si, quedando campeón no supera los 70 puntos, el entrenador correrá la misma suerte que el entrenador del equipo rival, el que le disputa el campeonato. Si no queda campeón pero supera los 70 puntos, correrá la suerte contraria que el entrenador del equipo rival. Si no queda campeón y no supera los 70 puntos, no se le renovará.

Se pide:

- Definir las variables booleanas del problema, codificándolas con el valor 1 el caso descrito en el párrafo anterior y con el valor 0 el caso contrario. Llamar F a la función booleana, siendo 0 en caso de no renovar.
- Tabla de verdad.
- Implementación del problema usando sólo puertas NOR.

Al objeto de tomar una decisión más acertada, el Presidente decide dejarse aconsejar por su Junta Directiva. Se aplican los mismos criterios que en el apartado anterior (incluyendo la renovación del entrenador del equipo rival) si queda campeón, pero en caso contrario, la decisión dependerá exclusivamente de lo que diga la Junta Directiva.

- Repetir la tabla de verdad en este segundo caso.
- Implementar la nueva función usando un MUX 4x1, colocando en las líneas de selección las variables que se consideren oportunas. Llamar G a la función booleana, siendo 0 en caso de no renovar.

- Definición de variables

$C=1$, queda campeón

$C=0$, no queda campeón

$P=1$, obtiene más de 70 puntos

$P=0$, no obtiene más de 70 puntos

$R=1$, renueva entrenador equipo rival

$R=0$, no renueva entrenador equipo rival

- Tabla de verdad

C	P	R	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- Implementación con puertas NOR

		P R			
F		00	01	11	10
	C	0	0	0	1
	1	0	1	1	1

$$F = (P + R) \cdot (C + \overline{R}) = (P + R) \cdot (C + \overline{R}) = (P + R) + (C + \overline{R})$$

d) Nueva tabla de verdad (y su correspondiente codificación)

$C=1$, queda campeón

$P=1$, obtiene más de 70 puntos

$R=1$, renueva entrenador equipo rival

$J=1$, la Junta Directiva propone renovar

$C=0$, no queda campeón

$P=0$, no obtiene más de 70 puntos

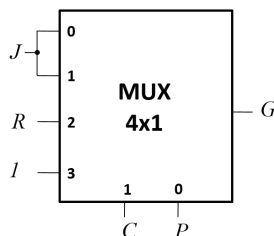
$R=0$, no renueva entrenador equipo rival

$J=0$, la Junta Directiva propone no renovar

C	P	R	J	G
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

C	P	R	J	G
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

e) Nueva implementación usando un MUX 4x1 con entradas de selección C y P . La tabla de verdad es dividida en cuatro pequeñas tablas de verdad de cuatro filas para $CP=00$, $CP=01$, $CP=10$ y $CP=11$ (ver imagen derecha). Fácilmente se obtienen las entradas a los canales de entrada.



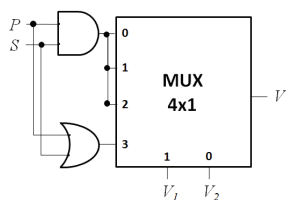
11) Un Tribunal de Justicia está compuesto por cuatro miembros, el Presidente (*P*), el Secretario (*S*), el Vocal 1 (*V1*) y el Vocal 2 (*V2*). A la hora de dictar el veredicto (*V*), cada uno individualmente toma una decisión sobre el acusado sin conocer el veredicto de los restantes miembros del Tribunal. Para tomar decisiones han acordado tener en cuenta el veredicto de la mayoría con el siguiente criterio:

- Si el Presidente considera que es inocente, ocurre lo siguiente: si al menos uno de los restantes miembros declara inocente al acusado, éste es declarado inocente. Si todos los demás miembros consideran que es culpable, éste es declarado culpable.
- Por el contrario, si el Presidente lo declara culpable, el veredicto se toma de la siguiente forma: si los dos Vocales lo declaran inocente, el veredicto final será el del Secretario. Si los dos Vocales discrepan, el veredicto final será el del Secretario, siempre que coincida con el del Presidente, y en caso contrario (si Presidente y Secretario discrepan) el acusado será declarado inocente. Por último, si los dos Vocales coinciden en declararlo culpable, será declarado culpable, independientemente de lo que diga el Secretario.

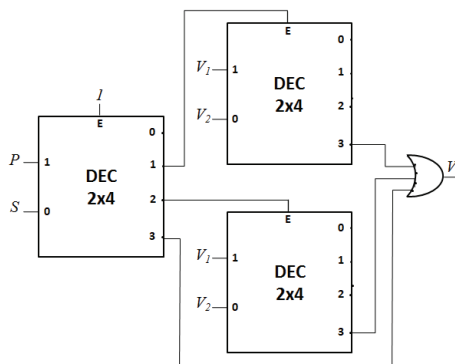
El abogado podrá recurrir la sentencia (*R*), sólo en caso de que lo declaren culpable, a otro tribunal de mayor rango si ha contado con: al menos dos votos favorables (inocente) de entre los cuatro del tribunal o bien si el presidente votó favorablemente.

Como los magistrados no saben nada de Informática, le encargan a un estudiante de 1º de la carrera que les diseñe un circuito para resolver este problema.

- Hallar la tabla de verdad que especifica el problema propuesto. Explicar si existen condiciones de No importa y para qué función de salida. La codificación tomada será como 0=inocente, 1=culpable y la ordenación de la tabla de verdad será *P, S, V1, V2*.
- Implementar la función de salida del veredicto (*V*) con un único MUX 4x1 y puertas necesarias, tomando como entradas de selección por orden de peso las decisiones del Vocal 1 y Vocal 2.
- Implementar la función de salida del veredicto (*V*) con una red de decodificadores DEC 4x1 en dos niveles (estructura de árbol) y puertas necesarias.
- Implementar la función del recurso de sentencia (*R*) con puertas NOR. Dibujar el diagrama.



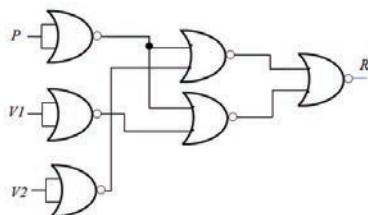
c) Red de decodificadores DEC 2x4 en dos niveles



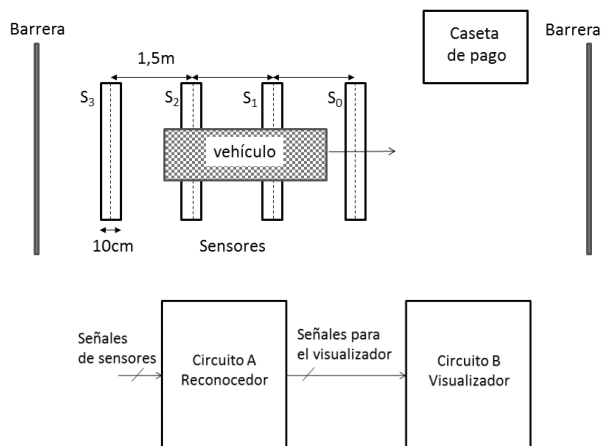
d) Implementación de la función del recurso de sentencia (R) con puertas NOR:

		V_1				
R		X	X	X	X	
		X	X	1	X	
		1	0	0	0	S
P		X	X	0	X	
		V_2				

$$R = (\overline{P} + \overline{V_2}) \cdot (\overline{P} + \overline{V_1}) = \overline{(\overline{P} + \overline{V_2}) + (\overline{P} + \overline{V_1})}$$



- 12) En una entrada de peaje de autopista, existe un carril exclusivo de vehículos ligeros. En este carril, existe un sistema de reconocimiento de vehículos que consta de cuatro bandas que actúan como sensores cuando son presionadas. Si la banda es presionada se produce un 1 lógico, mientras que si no es presionada la salida del sensor es 0. Llamar a estas variables S_3 , S_2 , S_1 y S_0 . Las bandas tienen una anchura de 10 cm y la separación entre los centros de dos bandas contiguas es de 1,5 metros (ver figura).



El sistema reconoce el tipo de vehículos que puede entrar por este carril en función de la señal que proporcionan las bandas cuando son presionadas por las ruedas del vehículo. Los tipos de vehículos reconocidos y el intervalo de separación posible entre ruedas son los siguientes:

Tipo 1: motocicletas/bicicletas. Separación entre ruedas: 145 a 160 cm

Tipo 2: coche pequeño. Separación entre ruedas 290 a 305 cm

Tipo 3: coche berlina. Separación entre ruedas. 447 a 457 cm

El coche es reconocido cuando la distancia entre dos sensores activos entra dentro del rango del tipo correspondiente. Para que no haya solapes, en el momento en el que un coche es reconocido, se bajan dos barreras, una por detrás y otra por delante que imposibilitan el acceso de un posible coche por detrás y que el coche pase sin pagar. Cuando el coche realiza el pago, se levantan dichas barreras.

El sistema tiene dos displays de 7 segmentos para mostrar diversa información. El display 1 está apagado mientras no haya ningún sensor presionado. Si el coche es reconocido, aparecerá en el display el número del tipo de coche (1, 2 ó 3). Si, aun habiendo algún sensor presionado, el sistema no reconoce el tipo, el display mostrará un 0 (como Tipo 0 de vehículo no reconocido). Considérese que cualquier otro tipo de vehículo (furgonetas, camiones, etc.) no puede entrar en este carril de peaje y que los sensores funcionan siempre correctamente.

Por su parte, el display 2 sólo está encendido cuando no hay ningún sensor presionado y, en este caso, muestra todos los segmentos del display iluminados.

Para hacer este circuito distinguir dos partes *A* y *B*. El circuito *A* es el que tiene por entradas la señal proveniente de los sensores y como salidas las funciones booleanas necesarias para generar el número del tipo (llamarlas variables T_1 y T_0) y la función que recoge la circunstancia de que no haya ningún sensor activado (variable V). El circuito *B*, se encarga de controlar los displays.

Obviar los demás elementos del sistema: barreras y caseta de pago.

Se pide:

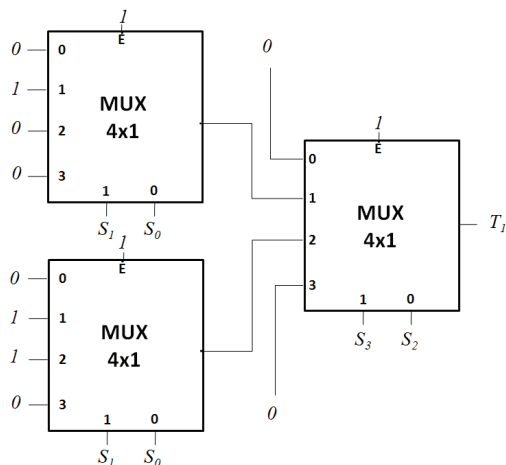
- a) Mostrar la tabla de verdad del circuito *A*. Explicar las variables de salida que se hayan tomado. Identificar los casos de “no importa”.
- b) Diseñar la función T_1 usando únicamente el módulo multiplexor 4x1 (uno o varios, los que se necesiten) y sin utilizar ninguna puerta, ni inversores.
- c) Diseñar la función T_0 con un DEC con salidas activas a nivel bajo y la función V usando puertas lógicas.
- d) Diseñar el circuito *B* utilizando módulos display de 7 segmentos (no hay que diseñarlos), los módulos adicionales que se necesiten y el mínimo número de puertas.

a) Tabla de verdad

S3	S2	S1	S0	T1	T0	V
0	0	0	0	x	x	1
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	0	1	1	0	0
0	1	1	0	0	1	0
0	1	1	1	x	x	0
1	0	0	0	0	0	0
1	0	0	1	1	1	0
1	0	1	0	1	0	0
1	0	1	1	x	x	0
1	1	0	0	0	1	0
1	1	0	1	x	x	0
1	1	1	0	x	x	0
1	1	1	1	x	x	0

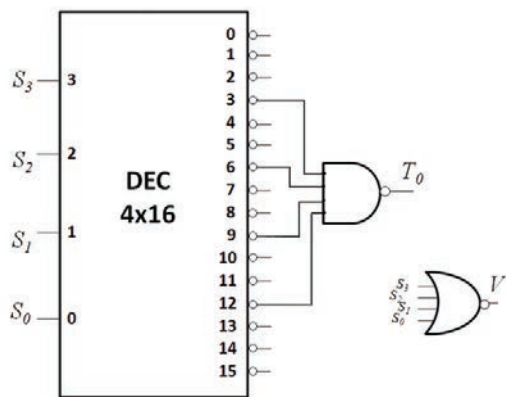
b) Implementación de T_1 usando únicamente módulos multiplexores 4x1.

En el primer nivel de multiplexores se toman las variables de menos peso como variables de control. En el segundo nivel se toma un único multiplexor tomando las variables de más peso como entradas de control. Se toman ceros en las condiciones de “no importa”.



c) Implementación de T_0 con un DEC con salidas activas a nivel bajo y V con puertas lógicas.

$V=1$ cuando no hay ningún sensor presionado.

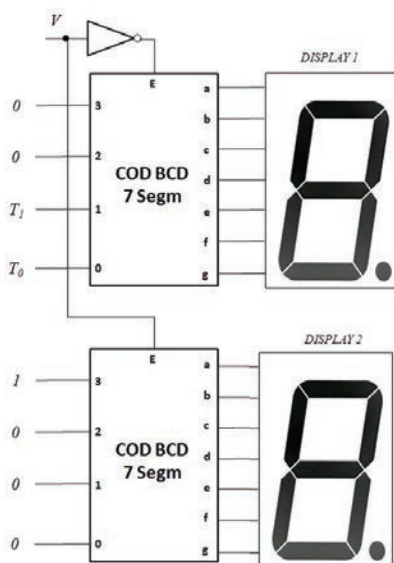


d) Diseñar el circuito B .

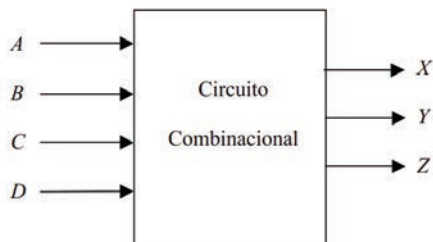
V selecciona el display a ser utilizado.

Si $V=0$, algún sensor es activado. El display muestra el tipo de coche.

Si $V=1$ ningún sensor es presionado y el display 2 está encendido mostrando el número 8 (todos los segmentos del display iluminados).



- 13) Diseñar un circuito combinacional de 4 entradas (A , B , C y D) que proporcione a la salida el número de unos que hay en las entradas, codificado en binario natural de 3 bits mediante las salidas X , Y y Z . Ejemplo: si las entradas toman los valores: $A=1$, $B=0$, $C=1$, $D=1$, las salidas deben tomar los valores: $X=0$, $Y=1$ y $Z=1$.



Se pide:

- La tabla de la verdad del circuito combinacional a diseñar.
- Realizar la función de salida X mediante un decodificador 3×8 con salidas activas a nivel alto y las puertas lógicas que se necesiten.
- Realizar la función de salida Y mediante un MUX 4×1 , usando A y B como líneas de selección.
- Realizar la función de salida Z mediante puertas que se consideren oportunas.

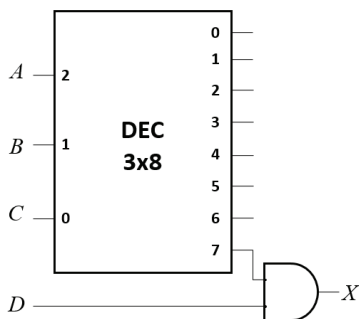
- e) Realizar el circuito completo mediante un convertidor de código, empleando un decodificador 4x16, un codificador 8x3 y las puertas lógicas necesarias.

- a) Tabla de verdad

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

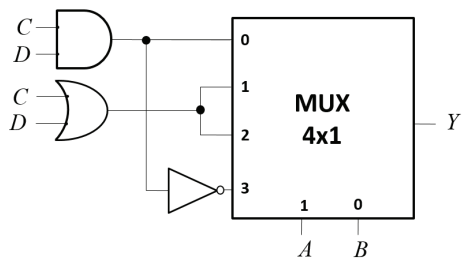
- b) Implementación de X mediante un decodificador 3x8.

La función X únicamente es 1 para $ABCD=1111$. Ya que $X = ABCD = m_7(A, B, C).D$, podemos usar un DEC 3x8 de entradas A , B y C y una puerta AND.



- c) Salida Y mediante un MUX 4x1, usando A y B como líneas de selección.

La tabla de verdad es dividida en cuatro pequeñas tablas de verdad de cuatro filas para cada caso $AB=00$, $AB=01$, $AB=10$ y $AB=11$, cuyas funciones corresponden a los canales de entrada. Fácilmente se obtienen las entradas.



d) Función de salida Z mediante puertas.

Pasando la tabla de verdad de Z se reconoce el esquema en tablero de ajedrez correspondiente a la función generador de paridad par de 4 bits.

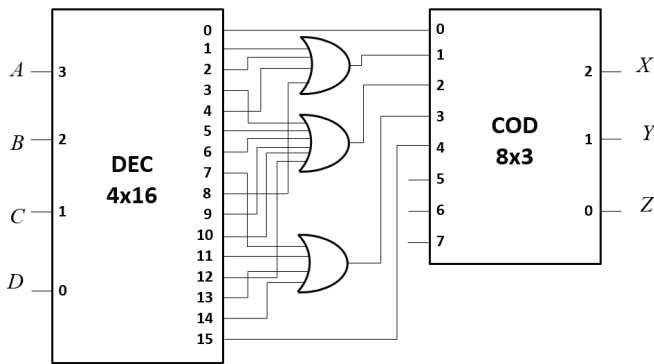
		C D			
A B	Z	00	01	11	10
	00	0	1	0	1
	01	1	0	1	0
	11	0	1	0	1
	10	1	0	1	0

$$Z = (A \oplus B) \oplus (C \oplus D)$$

e) Circuito completo mediante un convertidor de código.

Observar que la salida XYZ=000 sólo se espera para la combinación de las entradas ABCD=0000. Sin embargo, la salida XYZ=001 se debe generar en los casos ABCD=0001, ABCD=0010, ABCD=0100 y ABCD=1000. Por lo tanto, se toma una puerta OR de cuatro entradas a la salida del DEC 4x16 y su salida se lleva al canal 1 de COD 8x3.

De igual manera se razona para los restantes casos



- 14) La Figura 1 muestra el plano de un barrio donde hay vehículos aparcados en las plazas marcadas como *A*, *B* y *C*. Todas las calles son de un solo sentido, según se ve en la citada figura. Se trata de diseñar un circuito combinacional que active las señales que controlan las luces de los semáforos existentes (verde permitida, roja prohibida) para poder ir desde cada una de las plazas a las restantes, con la restricción de que, además de respetar el sentido de circulación, siempre hay que pasar por el TÚNEL en el sentido marcado.

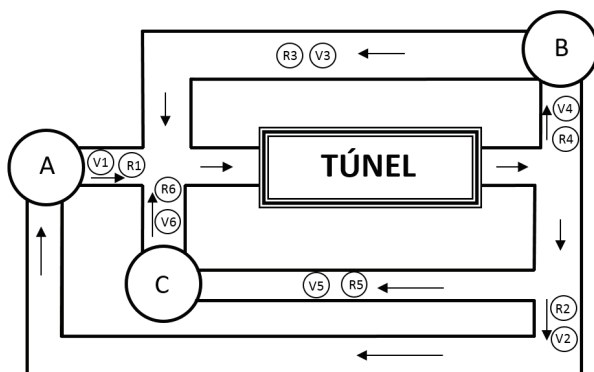


Figura 1

Un supervisor (Figura 2) gestiona el encendido y apagado de los semáforos, pulsando una botonera numerada de 0 a 7, que activa la entrada del mismo número del COD 8x3, con salidas activas a nivel alto, de la Figura 2:

- Para dejar libre el camino de *A* a *B*, pulsa un botón número 1.
- Para dejar libre el camino de *A* a *C*, pulsa un botón número 3.
- Para dejar libre el camino de *B* a *A*, pulsa un botón número 4.
- Para dejar libre el camino de *B* a *C*, pulsa un botón número 2.
- Para dejar libre el camino de *C* a *A*, pulsa un botón número 6.
- Para dejar libre el camino de *C* a *B*, pulsa un botón número 7.
- Para prohibir la circulación en todo el barrio, poniendo todos los semáforos en rojo, pulsa un botón número 5.
- Para apagar todas las luces, pulsa el botón número 0.



Figura 2

Se pide:

- a) **Tabla de verdad que resuelve el problema con TODAS las funciones necesarias.**
NOTA: Usar las últimas letras de abecedario para referirse a las variables independientes, siendo la Z la variable menos significativa.
- b) **Implementar las funciones de luz verde 1 y 2 con MUX 4x1 cada una, usando las dos variables más significativas como líneas de selección en el primer caso y las dos variables menos significativas en el segundo caso.**
- c) **Implementar las funciones de luz verde 3 y 4 con DEC3x8 con salidas activas a nivel bajo y las puertas necesarias.**
- d) **Implementar la función de luz roja 5 sólo con puertas NAND. No es necesario el dibujo.**
- e) **Implementar la función de luz roja 6 sólo con puertas NOR. No es necesario el dibujo.**

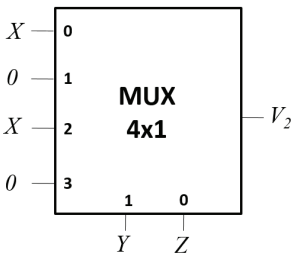
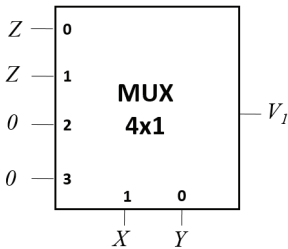
a) Tabla de verdad.

Se presenta el camino seguido Inicio → Final, el código XYZ del botón presionado y el estado de cada uno de los semáforos en la trayectoria desde Inicio a Final.

SIGNIFICADO	X Y Z	V ₁ R ₁	V ₂ R ₂	V ₃ R ₃	V ₄ R ₄	V ₅ R ₅	V ₆ R ₆
Apagado	0 0 0	0 0	0 0	0 0	0 0	0 0	0 0
A→B	0 0 1	1 0	0 1	0 1	1 0	0 1	0 1
B→C	0 1 0	0 1	0 1	1 0	0 1	1 0	0 1
A→C	0 1 1	1 0	0 1	0 1	0 1	1 0	0 1
B→A	1 0 0	0 1	1 0	1 0	0 1	0 1	0 1
Prohibido	1 0 1	0 1	0 1	0 1	0 1	0 1	0 1
C→A	1 1 0	0 1	1 0	0 1	0 1	0 1	1 0
C→B	1 1 1	0 1	0 1	0 1	1 0	0 1	1 0

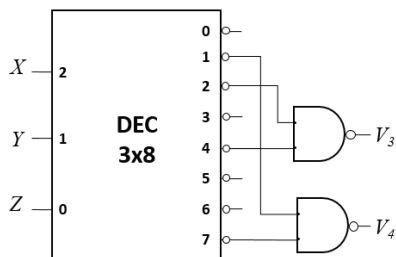
b) Implementación de V_1 y V_2 con MUX 4x1

Se usan las dos variables más significativas como líneas de selección para V_1 y las dos variables menos significativas para V_2 .



c) Implementación de V_3 y V_4 con DEC 3x8 con salida activa a nivel bajo.

Se toman los unos de la tabla de verdad y se llevan a sendas puertas NAND.



d) Implementación de R_5 con sólo NAND:

		YZ			
	R_5	00	01	11	10
X	0	0	1	0	0
	1	1	1	1	1

$$R_5 = X + \bar{Y} \cdot Z = \overline{\overline{X + \bar{Y} \cdot Z}} = \overline{\overline{X} \cdot \overline{\bar{Y} \cdot Z}} = \overline{\overline{X} \cdot (Y \cdot Z)}$$

e) Implementación de R_6 con sólo NOR:

		YZ			
	R_6	00	01	11	10
X	0	0	1	1	1
	1	1	1	0	0

$$R_6 = (\overline{X + \bar{Y}}) \cdot (X + Y + Z) = \overline{\overline{\overline{X + \bar{Y}}}} \cdot (X + Y + Z) = \overline{\overline{\overline{X + \bar{Y}}}} + \overline{\overline{\overline{X + \bar{Y}}}} \cdot \overline{\overline{\overline{X + \bar{Y}}}} \cdot (X + Y + Z)$$

15) Los ciudadanos acuden a una oficina de la Administración para realizar varios trámites de su interés. Para ello deberán rellenar una encuesta con cuatro preguntas:

- ¿Tiene gafas? (G)
- ¿Tiene reflejos, capacidad de reaccionar rápidamente ante imprevistos? (R)
- ¿Tiene título de Bachiller? (B)

- ¿Es mayor de 25 años? (E)

Si la respuesta es afirmativa, dichas variables se codificarán con un 1 y en caso negativo con un 0.

Se trata de diseñar un circuito tal que, introduciendo dichos datos, responda diciendo si se acepta o no la renovación del Permiso de Conducir (C), la renovación o no del Permiso de Armas (A) y la posibilidad de presentarse o no a unas oposiciones a la Policía (P). Dichas funciones toman el valor lógico 1 cuando es aceptada y 0 en el caso contrario. Para ello, la oficina tiene los siguientes criterios:

- Para poder renovar el Permiso de Conducir, deben tener reflejos y gafas, o bien título de Bachiller. En ese último caso, no deben ser mayor de 25 años.
- Para poder renovar el Permiso de Armas, deben tener reflejos y no tener gafas, o bien ser mayores de 25 años.
- Para presentarse a las oposiciones a la Policía, deben tener Bachiller o ser mayores de 25 años. En ese último caso, no deben tener gafas.

Diseñar un circuito que resuelva el problema. Para ello se pide:

- Tabla de verdad.
- Implementación de la función C sólo con puertas NAND.
- Implementación de la función A sólo con puertas NOR.
- Implementación de la función P con un MUX 4x1.
- Implementación de la función C con un DEC 3x8 con salidas activas a nivel alto y las puertas lógicas que se consideren necesarias.

- Tabla de verdad.

Variables				Funciones		
G	R	B	E	C	A	P
0	0	0	0	0	0	0
0	0	0	1	0	1	1
0	0	1	0	1	0	1
0	0	1	1	0	1	1
0	1	0	0	0	1	0
0	1	0	1	0	1	1
0	1	1	0	1	1	1
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	0	1	0	1	0
1	0	1	0	1	0	1
1	0	1	1	0	1	1
1	1	0	0	1	0	0
1	1	0	1	1	1	0
1	1	1	0	1	0	1
1	1	1	1	1	1	1

b) C sólo con puertas NAND.

		B E			
		00	01	11	10
G R	C	00	01	11	10
	00	0	0	0	1
	01	0	0	0	1
	11	1	1	1	1
	10	0	0	0	1

$$C = G \cdot R + B \cdot \overline{E} = \overline{\overline{G \cdot R} \cdot \overline{B \cdot \overline{E}}}$$

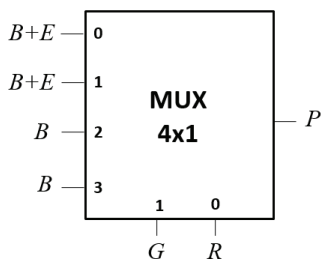
c) A sólo con puertas NOR.

		B E			
		00	01	11	10
G R	C	00	01	11	10
	00	0	1	1	0
	01	1	1	1	1
	11	0	1	1	0
	10	0	1	1	0

$$A = (\overline{G} + E) \cdot (R + E) = \overline{\overline{\overline{G} + E} \cdot \overline{R + E}}$$

d) P con MUX 4x1.

Tomando G y R como entradas de selección, La tabla de verdad de P es dividida en cuatro pequeñas tablas de verdad de cuatro filas para cada caso $GR=00$, $GR=01$, $GR=10$ y $GR=11$, cuyas funciones corresponden a los canales de entrada. Fácilmente se obtienen las entradas:

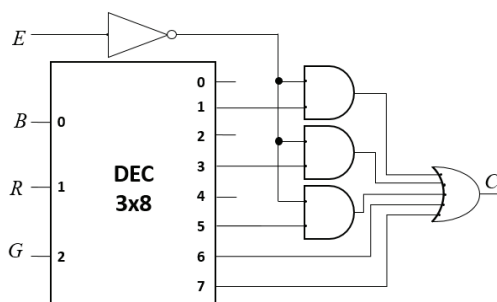


- e) C con un DEC 3x8 con salidas activas a nivel alto. Hay múltiples soluciones válidas. Se muestran dos posibles soluciones:

Tomando G , R y B como variables de entrada del decodificador, recorremos la tabla de verdad de C de dos en dos filas obteniendo la expresión en función de los minterms de G , R y B .

$$C = \bar{G} \bar{R} B \bar{E} + \bar{G} R B \bar{E} + G \bar{R} B \bar{E} + G R \bar{B} + G R B = (\bar{G} \bar{R} B + \bar{G} R B + G \bar{R} B) \bar{E} + G R \bar{B} + G R B =$$

$$= (m_{1(G,R,B)} + m_{3(G,R,B)} + m_{5(G,R,B)}) \bar{E} + m_{6(G,R,B)} + m_{7(G,R,B)}$$

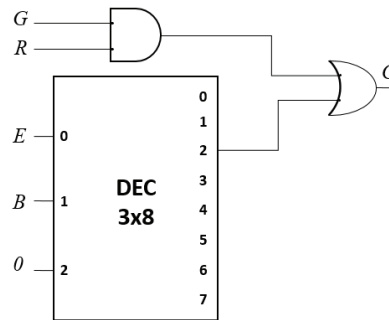


En la segunda forma calculamos la expresión simplificada de C como suma de productos. Se obtiene fácilmente la expresión

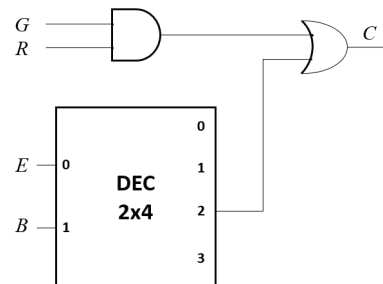
		B E			
		00	01	11	10
G R	00	0	0	0	1
	01	0	0	0	1
	11	1	1	1	1
	10	0	0	0	1

$$C = GR + B\bar{E}$$

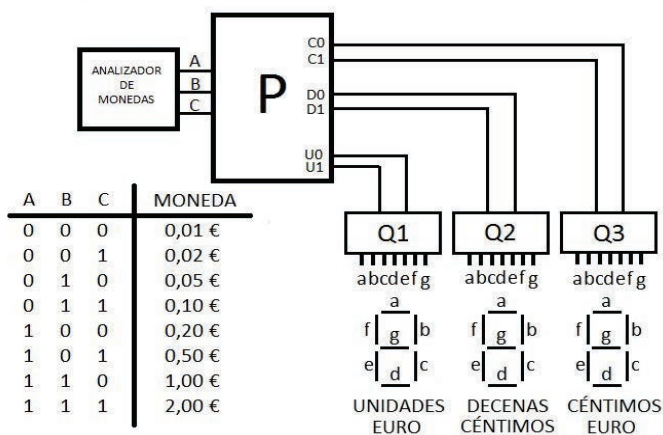
Tomando variables de entrada del decodificador como 0 , B y E , tenemos $C = GR + m_{2(B,E)}$



O bien con un DEC 2x4:



- 16) Se desea diseñar un circuito combinacional que presente en tres displays de 7 segmentos el valor en euros de una moneda previamente analizada según la tabla de la figura adjunta:



NOTA 1: No se contempla la posibilidad “sin moneda”

El circuito tiene cuatro bloques, uno de ellos llamado *P* y tres bloques iguales llamados *Q1*, *Q2* y *Q3*. El bloque *P* convierte el código de moneda ABC a un código de 6 bits (3 parejas) en los que se codifica las unidades de euro (pareja de bits *U1* y *U0*), las decenas de céntimos (pareja de bits *D1* y *D0*) y los céntimos de euro (pareja de bits *C1* y *C0*). Se emplea una codificación en binario natural, excepto para el 5, que emplea el código 11. Los tres bloques *Q1*, *Q2* y *Q3*, iguales, convierten cada uno de los códigos binarios de dos bits obtenidos en el bloque *P*, en un display de siete segmentos para la representación decimal del número correspondiente al código binario de entrada. Por ejemplo, la moneda de 0,50 € tendría 0 unidades de euro, 5 decenas de céntimos y 0 céntimos de euro. Las variables con el subíndice 1 son las más significativas.

- Realizar la tabla de verdad para las seis funciones *U1*, *U0*, *D1*, *D0*, *C1* y *C0*, dependientes de *A*, *B* y *C*. Minimizar las expresiones de dichas funciones y dibujar el esquema de sus circuitos lógicos, expresando la función *D1* sólo con puertas NOR y la función *D0* sólo con puertas NAND.
- Implementar las funciones *b*, *e*, *f* y *g* del bloque *Q1* con puertas lógicas en el modo más simple posible. El resto de funciones, *a*, *c* y *d*, con módulos DEC 2x4.

NOTA 2: El apartado b se puede realizar sin haber hecho previamente el apartado a.

- Tabla de verdad y expresiones de todas las salidas

A	B	C	U1	U0	D1	D0	C1	C0
0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	1	0
0	1	0	0	0	0	0	1	1
0	1	1	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0
1	0	1	0	0	1	1	0	0
1	1	0	0	1	0	0	0	0
1	1	1	1	0	0	0	0	0

Las expresiones pueden deducirse directamente de la tabla o mediante mapas de Karnaugh de tres variables.

$U1 = A \cdot B \cdot C$, es obvio ya que *U1* sólo vale 1 en ese caso.

$$U0 = A \cdot B \cdot \bar{C}$$

$D1 = A \cdot \bar{B} = \overline{\bar{A} + B}$, ya que *D1* toma valor 1 para los casos *A*=1, *B*=0, independientemente de *C*.

$D0 = A \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C = \overline{\overline{A \cdot \overline{B} \cdot C} \cdot \overline{\overline{A} \cdot B \cdot C}}$. Implementación con NAND.

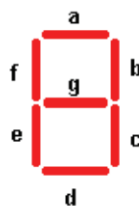
$C1 = \overline{A} \cdot (B + C) \cdot (\overline{B} + \overline{C})$. Implementación con NOR. Mirar cuando $C1$ toma valor 0.

o bien $C1 = \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot \overline{C} = \overline{A} \cdot (\overline{B} \cdot C + B \cdot \overline{C}) = \overline{A} \cdot (B \oplus C)$

$C0 = \overline{A} \cdot \overline{C}$, ya que $C0=1$ cuando $A=0$ y $C=0$, independientemente de B .

b) Bloque $Q1$

U1	U0	a	b	c	d	e	f	g
0	0	1	1	1	1	1	1	0
0	1	0	1	1	0	0	0	0
1	0	1	1	0	1	1	0	1
1	1	1	0	1	1	0	1	1

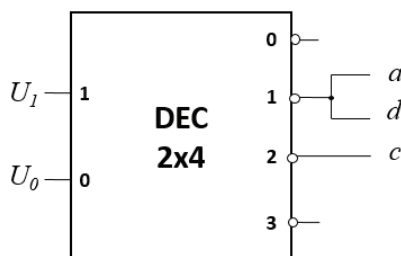


Implementaciones con puertas de b, e, f, g

$$b = \overline{U_1} + \overline{U_0} = \overline{U_1 \cdot U_0} \quad e = \overline{U_0} \quad f = \overline{U_1 \oplus U_0} \quad g = U_1$$

Serían posibles otras soluciones (peores) usando DEC activo a nivel alto y puertas, o usando un DEC para cada función, o ambas.

Implementación de a, c y d con un único DEC 2x4. Tomo el decodificador con salida negada.



- 17) La Figura 1 muestra parte del plano de una ciudad donde aparecen cuatro calles llamadas A , B , C y D .

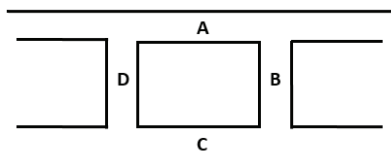


Figura 1

Cada calle tiene un sensor del mismo nombre que la calle que indica a un circuito combinacional a diseñar (Figura 2) si todas las farolas de la calle están encendidas o están apagadas. Por ejemplo, para la calle A , el sensor $A=1$ si las farolas están encendidas y $A=0$ si las farolas están apagadas.

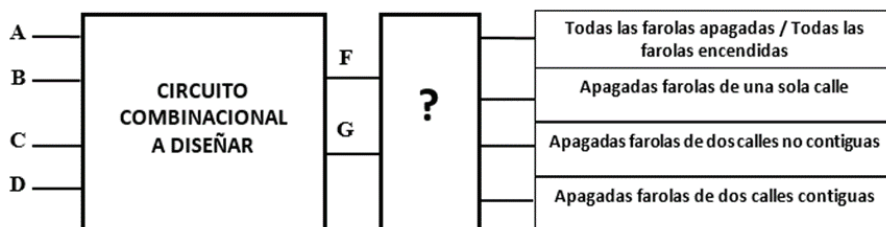


Figura 2

Se trata de diseñar un circuito combinacional con dos salidas F y G que cumpla las siguientes especificaciones:

- Si las farolas de las cuatro calles están apagadas, las salidas F y G deben ser 0. Lo mismo ocurrirá si las farolas de las cuatro calles están encendidas.
- Si las farolas de una única calle están apagadas, y las farolas de las otras tres calles están encendidas, F será 0 y G será 1.
- Si las farolas de dos calles no contiguas (y en nuestro caso, paralelas) están apagadas y las farolas de las otras dos calles están encendidas, F será 1 y G será 0.
- Si las farolas de dos calles contiguas están apagadas y las farolas de las otras dos calles están encendidas, F será 1 y G será 1.
- Nunca ocurrirá que estén apagadas las farolas de tres calles y encendidas las farolas de la otra calle.

Se pide diseñar el circuito combinacional de la siguiente forma:

- a) Tabla de verdad que resuelve el problema siendo A la variable más significativa.

- b) Implementar F sólo con puertas NOR.
- c) Implementar G sólo con puertas NAND.
- d) Implementar G de forma que no se necesiten más de 3 puertas, incluidas las XOR.
- e) Tal como dice el enunciado, no está previsto que haya tres calles con las farolas apagadas, pero ¿qué salida se produciría si las calles A , B y C tuvieran sus farolas apagadas en la solución que se ha adoptado?
- f) ¿Qué módulo combinacional habría en el interior del bloque marcado con una interrogación de la Figura 2 para activar cada uno de los luminosos que aparecen en la citada figura? Dibujarlo con todo detalle, incluyendo los pesos de las entradas y salidas.

- a) Tabla de verdad.

A	B	C	D	F	G
0	0	0	0	0	0
0	0	0	1	X	X
0	0	1	0	X	X
0	0	1	1	1	1
0	1	0	0	X	X
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	0	1
1	0	0	0	X	X
1	0	0	1	1	1
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	0

- b) F sólo con puertas NOR.

		C D			
		00	01	11	10
A B	F	0	X	1	X
	00				
	01	X	1	0	1
	11	1	0	0	0
	10	X	1	0	1

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

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

El 0 de la posición 0000 también se podría haber agrupado con la X que tiene debajo, o con la X de la posición 0010, o con la X de la posición 1000 produciendo en esos casos los términos $(A + C + D)$, $(A + B + D)$ o $(B + C + D)$, respectivamente, en vez del término $(A + B + C)$.

c) G sólo con puertas NAND.

		C D			
		00	01	11	10
A B	G	0	X	1	X
	00				
	01	X	0	1	1
	11	1	1	0	1
	10	X	1	1	0

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

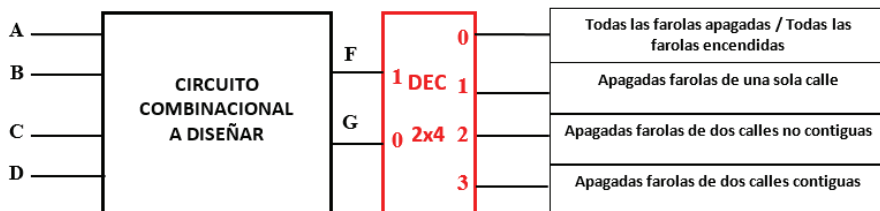
d) G con no más de 3 puertas (incluyendo XOR).

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

e) Valores dados a las condiciones de “no importa”

$F(0,0,0,1)=0$ y $G(0,0,0,1)=1$. Por tanto, la entrada no prevista ABCD=0001 produciría una salida FG=01.

f) Sería un decodificador como el que aparecen en la Figura adjunta



18) Se quiere diseñar un codificador de prioridad con tres líneas de entrada (3 bits), X_2 , X_1 , X_0 donde X_0 tiene la máxima prioridad, la siguiente línea en prioridad es X_2 y la mínima prioridad es X_1 . El codificador es activo al nivel alto a la entrada y a la salida. La salida del codificador está implementada con dos líneas (2 bits) Y_1Y_0 de la siguiente forma:

- La salida es 11 para la máxima prioridad
- La salida es 10 para la prioridad intermedia
- La salida es 01 para la mínima prioridad
- La salida es 00 cuando $X_2=X_1=X_0=0$.

- a) Hallar la tabla de verdad de este codificador y las expresiones simplificadas de Y_1 e Y_0 , escribiendo las variables de entrada en el orden $X_2 X_1 X_0$.
- b) Implementar Y_1 con NAND e Y_0 con NOR.

a) Tabla de verdad y las expresiones simplificadas de Y_1 e Y_0 :

X_2	X_1	X_0	Y_1	Y_0
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

Y_1	X_1			
	0	1	1	0
X_2	1	1	1	1
	X_0			

$$Y_1 = X_2 + X_0$$

Y_0	X_1			
	0	1	1	1
X_2	0	1	1	0
	X_0			

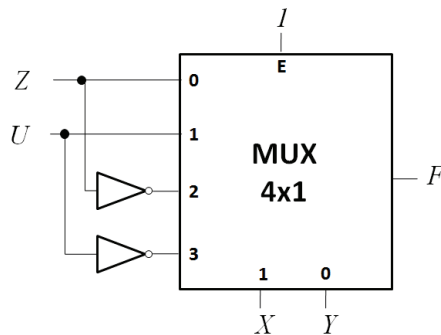
$$Y_0 = X_0 + \bar{X}_2 X_1$$

b) Y_1 con NAND e Y_0 con NOR:

$$Y_1 = X_2 + X_0 = \overline{\overline{X_2 + X_0}} = \overline{\overline{X_2} \overline{X_0}}$$

$$Y_0 = X_0 + \overline{X_2} X_1 = (X_0 + \overline{X_2})(X_0 + X_1) = \overline{\overline{(X_0 + \overline{X_2})} \overline{(X_0 + X_1)}} = \overline{\overline{(X_0 + \overline{X_2})} + \overline{(X_0 + X_1)}}$$

19) Dado el circuito de la figura:



- Obtener la expresión de F como suma de productos.
- Implementar F utilizando puertas XOR y cualquier otro tipo de puertas.
- Implementar F con MUXs y DECs en estructura de árbol en dos niveles.
- Suponer que $U=1$. Implementar F con un DEC con salida invertida.

- Por la misma definición de un MUX se obtiene directamente:

$$F(X, Y, Z, U) = \overline{X}\overline{Y}Z + \overline{X}YU + X\overline{Y}\overline{Z} + XY\overline{U}$$

También se podría haber hecho la tabla de verdad y expresarlo en forma canónica.

- Implementar F utilizando puertas XOR

Se recurre a utilizar manipulación booleana.

$$\begin{aligned} F(X, Y, Z, U) &= \overline{X}\overline{Y}Z + \overline{X}YU + X\overline{Y}\overline{Z} + XY\overline{U} = \overline{X}\overline{Y}Z + X\overline{Y}\overline{Z} + \overline{X}YU + XY\overline{U} = \\ &= \overline{Y}(\overline{X}Z + X\overline{Z}) + Y(\overline{X}U + X\overline{U}) = \overline{Y}(X \oplus Z) + Y(X \oplus U) \end{aligned}$$

- F con MUXs y DECs en estructura de árbol en dos niveles

Hay que hallar la expresión canónica de F . Podemos recurrir al mapa de Karnaugh para obtener la tabla de verdad de la función $F(X, Y, Z, U) = \overline{X}\overline{Y}Z + \overline{X}YU + X\overline{Y}\overline{Z} + XY\overline{U}$

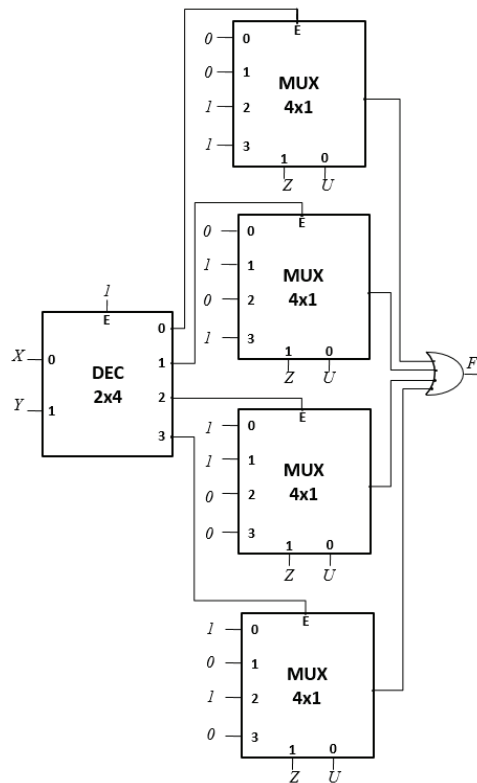
F	Z				
	0	0	1	1	
	0	1	1	0	
	1	0	0	1	
	X	1	1	0	0
		U			

Según el mapa de Karnaugh obtenemos que:

$$F(X,Y,Z,U) = \sum_4 m(2,3,5,7,8,9,12,14)$$

Luego la tabla de verdad y la implementación pedida son:

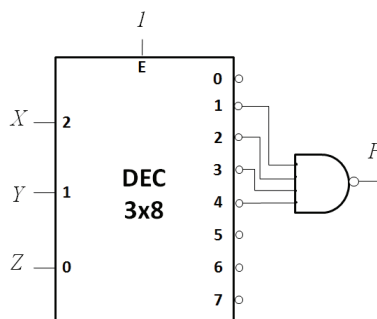
X	Y	Z	U	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0



d) Para $U=1$. Implementar F con un DEC con salida invertida. La función F sería:

$$F(X, Y, Z) = \bar{X}\bar{Y}Z + \bar{X}Y + X\bar{Y}\bar{Z} = \bar{X}\bar{Y}Z + \bar{X}Y(Z + \bar{Z}) + X\bar{Y}\bar{Z} = \bar{X}\bar{Y}Z + \bar{X}YZ + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} = \sum m(1, 2, 3, 4)$$

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0



20) Implementar un 3-desplazador de 4 bits utilizando el tipo de multiplexores necesario. Suponer por separado los casos unidireccional y bidireccional.

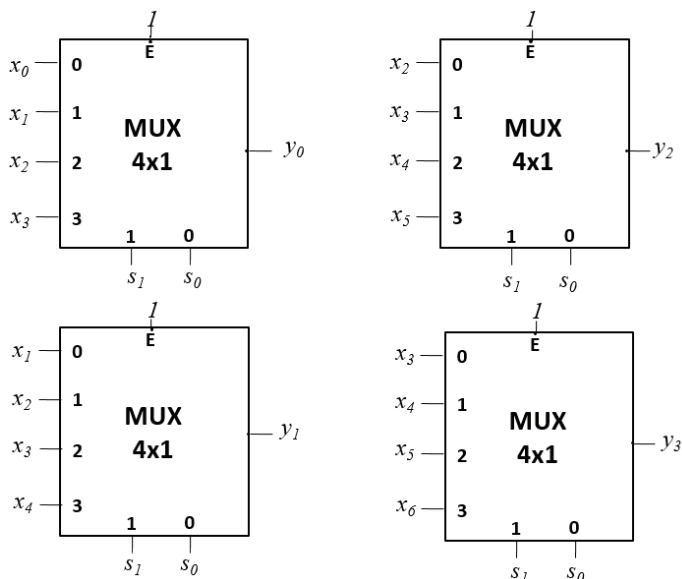
a) Caso unidireccional.

Máximo desplazamiento $p=3$, número de bits que desplaza $n=4$

El número de entradas de selección es 2 ya que $\log_2(p+1) = \log_2 4 = 2$, y no existe una entrada de selección de dirección. Por tanto, los multiplexores que utilizamos son MUX 4x1.

El número total de entradas al desplazador es $n+p=7$, las llamamos $x_6, x_5, x_4, x_3, x_2, x_1, x_0$

Como el número de bits de salida es 4, necesitamos 4 multiplexores. Cada multiplexor tiene un grupo de 4 entradas de subíndices ordenados y desfasados en una unidad con respecto al siguiente grupo. Si, por ejemplo, las entradas de selección marcan un desplazamiento en 2 ($s_1 = 1$ y $s_0 = 0$), se seleccionarían las entradas x_2, x_3, x_4 y x_5 correspondientes a la entrada 2ª de los respectivos multiplexores.



Como se puede apreciar, las entradas de activación están todas a 1.

b) Caso bidireccional.

Igual que antes $p=3$ y $n=4$. Ahora el número total de entradas al desplazador es $n+2p = 10$. Llamaremos a las variables de entrada

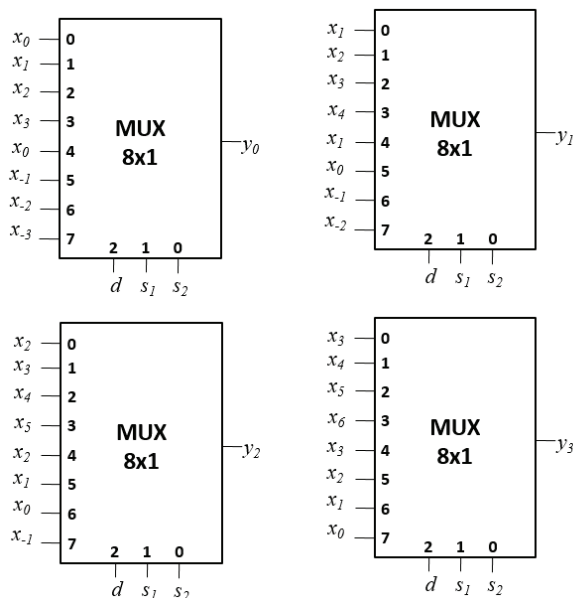
$$x_6, x_5, x_4, x_3, x_2, x_1, x_0, x_{-1}, x_{-2}, x_{-3}$$

También necesitaremos 4 multiplexores, esta vez con una entrada añadida de control de dirección y dos entradas de control de desplazamiento sobre las entradas de selección. Se concluye que los multiplexores han de ser de 8 entradas.

Sobre las entradas 0 y 4 se mantienen los bits centrales x_0, x_1, x_2, x_3 respectivamente. A su izquierda se introducen los consecutivos para desplazamientos a derecha o izquierda, ordenados de forma inversa.

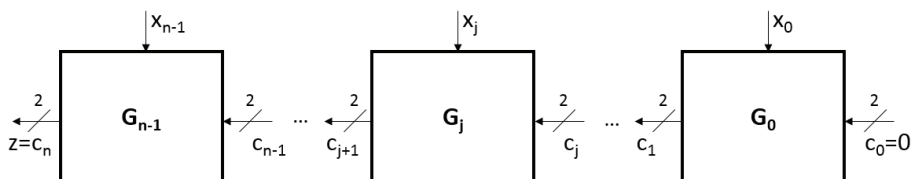
Por ejemplo, suponiendo un desplazamiento a derechas ($d=1$) en dos lugares 2 ($s_1=1$ y $s_0=0$), se obtiene un valor de entrada de selección 110, por lo que la salida corresponderá a la entrada 6ª de cada multiplexor. La salida será x_1, x_0, x_{-1}, x_{-2} , como cabía esperar.

El esquema es el siguiente:



21) Diseñar una red iterativa para un sistema de n entradas binarias $(x_{n-1}, \dots, x_0)$ y una salida binaria z . La salida vale 1 cuando existe un índice j , $0 \leq j \leq n-3$ tal que $x_j = 1$, $x_{j+1} = 0$, $x_{j+2} = 1$. Esto es, reconoce el patrón de bits 101.

1.- Disponemos de n módulos y cada entrada binaria x_j es la entrada externa de cada módulo G_j . La salida de la red se obtiene de la salida interna del último módulo. Es decir, todos los módulos son, salvo una pequeña modificación en el último, iguales.



2.- Las variables internas del módulo G_j pueden tener 4 valores diferentes.

(a) El patrón 101 ya ha ocurrido, existe algún t tal que $t < j-3$ cumpliéndose $x_j = 1$, $x_{j+1} = 0$, $x_{j+2} = 1$

(b) No sucede a) y $x_{j-2} = 1$, $x_{j-1} = 0$, es decir las dos entradas de los dos módulos anteriores han sido 0 y 1.

(c) No sucede a) y $x_{j-1} = 1$, la entrada del módulo anterior ha sido 1.

(d) No se está en ningún caso de los anteriores.

3.- La especificación de alto nivel sobre cuál es la información que se transmite entre módulos es, según las variables definidas:

x_j	c_j	c_{j+1}
0	a	a
0	b	d
0	c	b
0	d	d
1	a	a
1	b	a
1	c	c
1	d	c

4.- Especificación de bajo nivel.

Codificando las variables internas como:

c_j	$c1_j$	$c2_j$
a	0	0
b	0	1
c	1	0
d	1	1

se construye una tabla de verdad del módulo G_j .

x_j	$c1_j$	$c0_j$	$c1_{j+1}$	$c0_{j+1}$
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	0

Se obtiene

$$c1_{j+1} = \sum m(1,3,6,7)$$

$$c0_{j+1} = \sum m(1,2,3)$$

5.- Simplificamos las funciones.

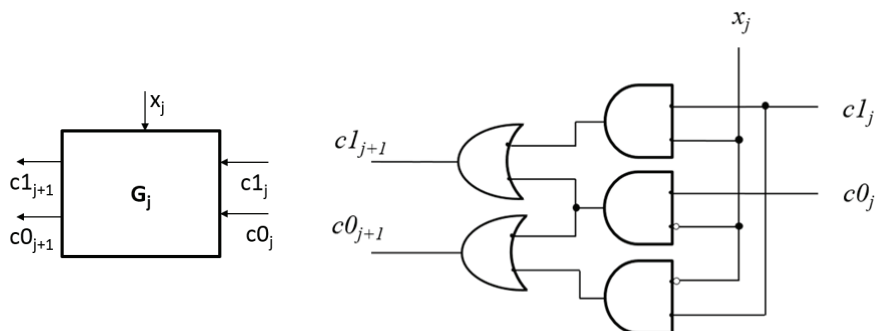
		$c_1 c_0$				
		c_{1+j}	00	01	11	10
x_j	0	0	1	1	0	
	1	0	0	1	1	

$$c1_{j+1} = x_j c1_j + \overline{x_j} c0_j$$

		$c1_j c0_j$				
		$c0_{j+1}$	00	01	11	10
x_j	0	0	1	1	1	
	1	0	0	0	0	

$$c0_{j+1} = \overline{x_j} c1_j + \overline{x_j} c0_j$$

La implementación de un módulo es la siguiente:

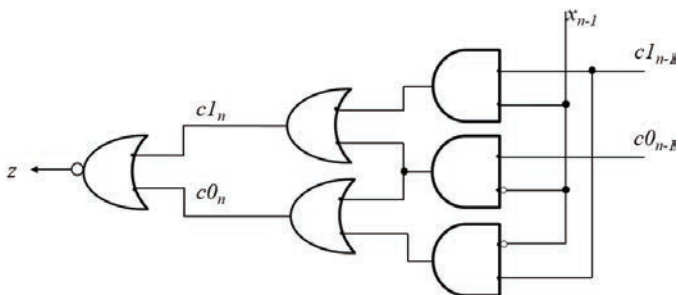


$c1_j$ y $c0_j$ del primer módulo, o sea, $c1_0$ y $c0_0$, son 11.

La implementación del último módulo es igual a los anteriores, pero se le añade una puerta NOR, ya que la salida z es:

$$z = \overline{(c1_n + c0_n)} = \overline{c1_n} \cdot \overline{c0_n}$$

Si se ha reconocido la secuencia 101, estaríamos en el caso a (00), y en ese caso la salida debe ser 1.

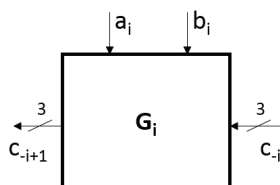


22) Diseñar, mediante una red iterativa, un comparador de dos números naturales de n bits.

Disponemos de dos entradas A y B , representadas por n bits cada una de ellas, $A(a_{n-1}, \dots, a_0)$ y $B(b_{n-1}, \dots, b_0)$.

1.- Cada módulo recibe dos entradas, correspondientes a un bit por cada uno de los dos números. El valor de k es 2, como en realidad hay $2*n$ variables binarias de entrada, necesitamos n módulos.

El i -ésimo módulo tiene por entradas externas a_i y b_i y no existe salida externa.



2.- Determinación de variables internas.

Llamando A_i y B_i a las cantidades formadas por los i primeros bits de cada operando A y B , la información que sobre ellas puede ser transferida al módulo G_i puede ser la siguiente:

$$c_i = M \text{ (Mayor) } A_i > B_i$$

$$c_i = i \text{ (Igual) } A_i = B_i$$

$$c_i = m \text{ (Menor) } A_i < B_i$$

3.- Las salidas internas se definen en función de las entradas internas y externas de la siguiente manera:

c_j	a_j	b_j	c_{j+1}
m	0	0	m
m	0	1	m
m	1	0	M -
m	1	1	m
i	0	0	i *
i	0	1	m
i	1	0	M -
i	1	1	i *
M	0	0	M -
M	0	1	m
M	1	0	M -
M	1	1	M -

4.- Adoptamos una codificación de tres variables binarias para las salidas internas. Esto soluciona el problema de añadir algo al último módulo.

c_j	g_i	e_i	s_i
M	1	0	0
i	0	1	0
m	0	0	1

Las expresiones de g_{i+1} , e_{i+1} y s_{i+1} se obtienen directamente de la tabla de alto nivel para evitarnos una tabla de verdad de 8 columnas con cinco variables de entrada.

- Para el caso en que $c_{i+1} = M$ (están señalados con -)

$$g_{i+1} \begin{cases} 1 & \text{si } (a_i = 1, b_i = 0) \text{ ó } (a_i = b_i = 0, g_i = 1) \text{ ó } (a_i = b_i = 1, g_i = 1) \\ 0 & \text{resto} \end{cases}$$

$$\text{luego } g_{i+1} = a_i \bar{b}_i + g_i (\bar{a}_i \bar{b}_i + a_i b_i)$$

- Para el caso en que $c_{i+1} = i$ (están señalados con *)

$$e_{i+1} \begin{cases} 1 & \text{si } (a_i = b_i = 0, e_i = 1) \text{ ó } (a_i = b_i = 1, e_i = 1) \\ 0 & \text{resto} \end{cases}$$

$$\text{luego } e_{i+1} = e_i (\bar{a}_i \bar{b}_i + a_i b_i)$$

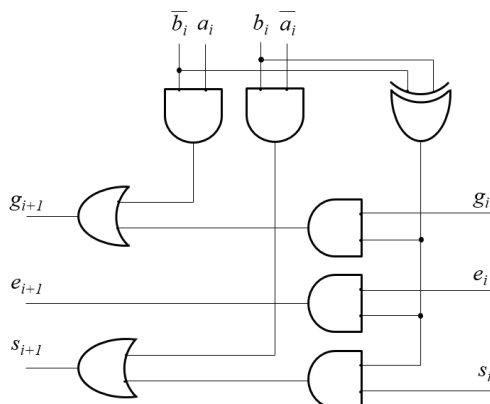
- Para el caso en que $c_{i+1} = m$

$$s_{i+1} \begin{cases} 1 & \text{si } (a_i = 0, b_i = 1) \text{ ó } (a_i = b_i = 0, s_i = 1) \text{ ó } (a_i = b_i = 1, s_i = 1) \\ 0 & \text{resto} \end{cases}$$

$$\text{luego } s_{i+1} = \bar{a}_i b_i + s_i (\bar{a}_i \bar{b}_i + a_i b_i)$$

En las tres funciones se nota que existe una función de equivalencia $a_i \quad b_i = \bar{a}_i \bar{b}_i + a_i b_i$

5.- La implementación del bloque g_i es:



23) Implementar la función

$$z = \begin{cases} 1 & \text{si } a = (b-1) \bmod 8 \\ 2 & \text{si } a = b \\ 0 & \text{resto} \end{cases}$$

donde $0 \leq a, b \leq 7$, utilizando multiplexores y decodificadores.

Hacemos una codificación de la salida z con dos variables binarias z_1 y z_0 .

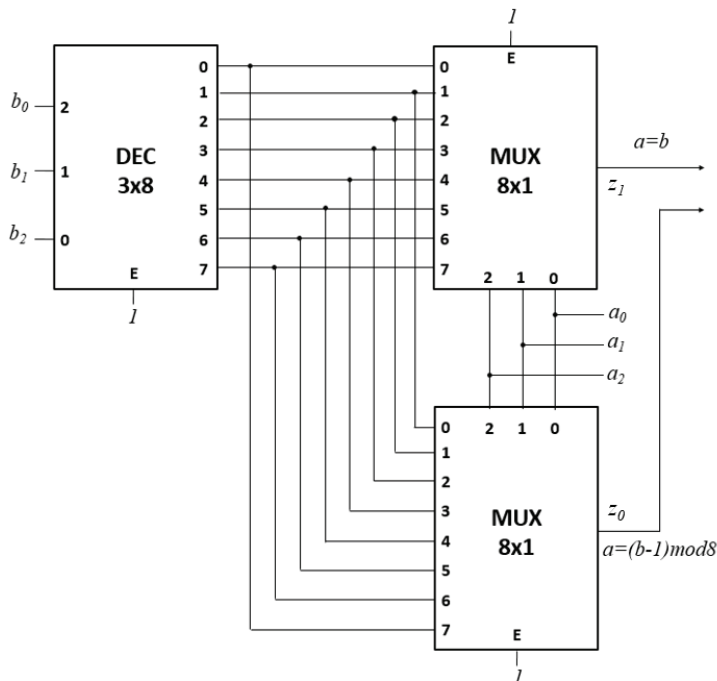
z_j	z_1	z_0
0	0	0
1	0	1
2	1	0

Igualmente codificamos los dos números a y b , mediante tres variables cada uno: $a(a_2, a_1, a_0)$ y $b(b_2, b_1, b_0)$. Como hay dos variables binarias de salida, estas las podemos implementar con dos multiplexores 8x1.

Las variables b_2, b_1 y b_0 se utilizan como entradas a un decodificador, y las variables a_2, a_1 y a_0 como entradas de selección de los multiplexores.

El caso más fácil es si $a = b$, ya que se unen directamente las salidas del DEC a las entradas del MUX, la salida z_1 será 1 en ese caso.

En caso de que $a = (b-1) \bmod 8$ la disposición es parecida, pero uniendo salidas a entradas desfasadas en una unidad. Por ejemplo, si $b=5$ y $a=4$, la salida z_0 debe ser 1, esto se logra si la salida 5ª del DEC está unida a la entrada 4ª del MUX.



- 24) Dadas las funciones f y g , implementar la función $f \cdot g$ mediante una red de decodificadores de dos entradas en forma de árbol. Utilizar a lo sumo una puerta OR.

$$f = wx\bar{y} + \bar{y}z + \bar{w}y\bar{z} + \bar{x}y\bar{z}$$

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

Realizaremos el producto $f \cdot g$ mediante los mapas de Karnaugh de las dos funciones. Ya que la implementación es a base de decodificadores, no buscamos una expresión simplificada, sino más bien la expresión en suma de minterminos.

		y z			
		00	01	11	10
w x	f	00	01	11	10
	00	0	1	0	1
	01	0	1	0	1
	11	1	1	0	0
	10	0	1	0	1

		y z			
		00	01	11	10
w x	g	00	01	11	10
	00	1	1	0	1
	01	1	1	1	0
	11	1	0	1	0
	10	1	0	1	1

El mapa del producto $f \cdot g$ tendrá unos en los lugares que procedan de dos unos en los mapas de f y g . En los demás lugares el producto $f \cdot g$ valdrá 0.

		y z			
		00	01	11	10
w x	f.g	00	01	11	10
	00	0	1	0	1
	01	0	1	0	0
	11	1	0	0	0
	10	0	0	0	1

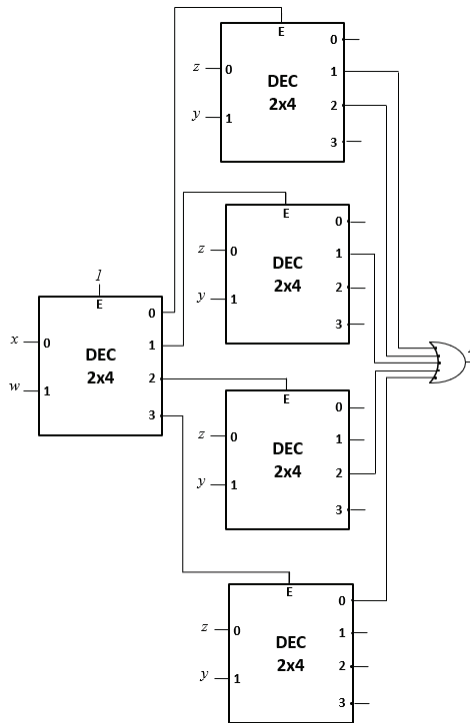
Expresamos $f \cdot g$ en suma de minterminos,

$$f \cdot g = \sum m(1, 2, 5, 10, 12)$$

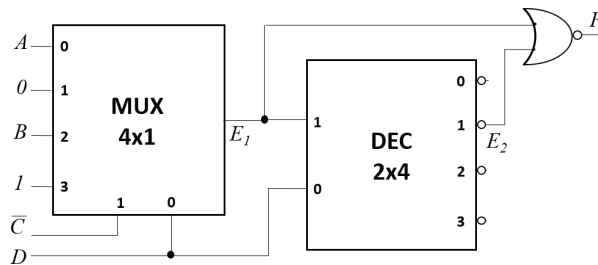
Como la implementación debe hacerse en forma de árbol, dividimos el vector de entrada en dos subvectores (w, x) para entradas al decodificador del primer nivel, y (y, z) como entradas a los decodificadores del segundo nivel.

Nº MINTERMS	W	X	Y	Z
1	0	0	0	1
2	0	0	1	0
5	0	1	0	1
10	1	0	1	0
12	1	1	0	0

La implementación es la siguiente:



25) Calcular la expresión booleana de F en el circuito de la Figura adjunta, haciendo su tabla de verdad previamente. Suponer que A es la variable más significativa.



La primera forma de hacer el ejercicio es calcular el valor de la salida para combinación de las variables de entrada y rellenar la tabla de verdad.

A	B	C	D	E ₁	E ₂	F
0	0	0	0	0	1	0
0	0	0	1	1	1	0
0	0	1	0	0	1	0
0	0	1	1	0	0	1
0	1	0	0	1	1	0
0	1	0	1	1	1	0
0	1	1	0	0	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	0
1	0	0	1	1	1	0
1	0	1	0	1	1	0
1	0	1	1	0	0	1
1	1	0	0	1	1	0
1	1	0	1	1	1	0
1	1	1	0	1	1	0
1	1	1	1	0	0	1

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

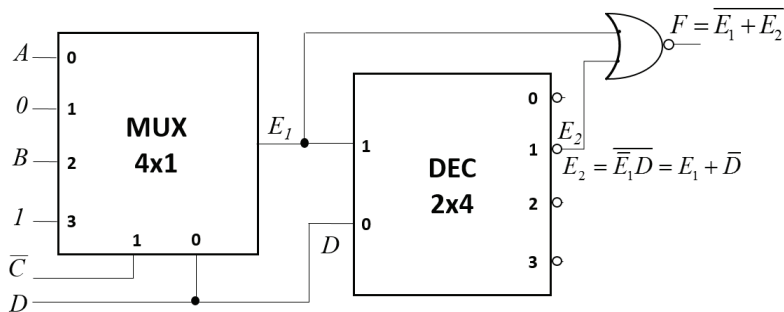
$$F = CD$$

La segunda forma es mediante cálculo de las expresiones booleanas que salen de cada componente del circuito. En la figura adjunta se muestra que la función F es $F = \overline{E_1} + E_2$ siendo E_1 la salida del multiplexor y E_2 la salida 1 del decodificador.

$$E_1 = A\overline{C}\overline{D} + 0 \cdot \overline{C}D + B\overline{C}\overline{D} + 1 \cdot \overline{C}D = A\overline{C}\overline{D} + B\overline{C}\overline{D} + \overline{C}D;$$

$$E_2 = \overline{\overline{E_1}D} = E_1 + \overline{D};$$

$$\begin{aligned} F &= \overline{E_1 + E_2} = \overline{E_1 + \overline{D}} = \overline{A\overline{C}\overline{D} + B\overline{C}\overline{D} + \overline{C}D + \overline{D}} = \overline{\overline{D}(AC + BC + 1) + \overline{C}D} = \\ &= \overline{\overline{D} + \overline{C}D} = \overline{\overline{D} + \overline{C}} = DC = CD \end{aligned}$$

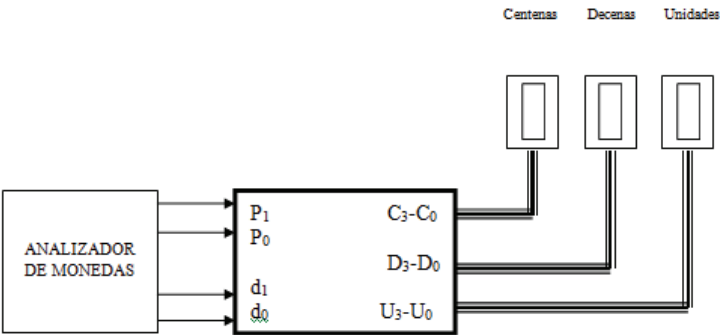


$$F = \overline{E_1 + E_2} = \overline{E_1 + E_1 + \overline{D}} = \overline{E_1 + \overline{D}}$$

26) Se desea diseñar un circuito combinacional que active tres displays de 7 segmentos en los que se representa el valor en pesetas de diversos tipos de monedas. Para ello, existe un dispositivo de entrada al circuito que analiza el peso y el diámetro de las monedas (ANALIZADOR DE MONEDAS) y proporciona una salida codificada digitalmente según la tabla adjunta:

CODIFICACIÓN PESO			CODIFICACIÓN DIÁMETRO		
PESO (gr)	P ₁	P ₀	DIÁMETRO (mm)	d ₁	d ₀
Menor de 1	0	0	Menor de 16	0	0
De 1 a 4	0	1	De 16 a 20	0	1
De 4 a 8	1	0	De 20 a 25	1	0
Mayor de 8	1	1	Mayor de 25	1	1

El circuito a diseñar admite como entrada las líneas P_1 , P_0 , d_1 y d_0 , y tiene líneas de salida para actuar sobre los tres displays, con cuatro líneas cada uno, para ver respectivamente las centenas, las decenas y las unidades.



Teniendo en cuenta que los pesos y diámetros de las monedas de interés son:

TIPO DE MONEDA	PESO (gr)	DIÁMETRO (mm)
1 peseta	0.6	14
5 pesetas	3.1	17.5
25 pesetas	4.4	19.5
100 pesetas	9.3	24.5

Diseñar dicho circuito suponiendo que sólo se van a introducir las cuatro monedas descritas en la tabla anterior y realizar la implementación más simplificada posible usando puertas lógicas.

TIPO DE MONEDA	CODIFICACIÓN DE LAS MONEDAS	
	P_1P_0	d_1d_0
1 peseta	0 0	0 0
5 pesetas	0 1	0 1
25 pesetas	1 0	0 1
100 pesetas	1 1	1 0

La tabla de verdad con entradas P_1 , P_0 , d_1 y d_0 es la siguiente:

P ₁	P ₀	d ₁	d ₀	C ₃	C ₂	C ₁	C ₀	D ₃	D ₂	D ₁	D ₀	U ₃	U ₂	U ₁	U ₀
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X
0	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X
0	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X
0	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X
0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1
0	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X
0	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X
1	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X
1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	1
1	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X
1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X

Si hacemos que las X sean 0, muchas funciones son 0. Implementamos las que no son 0.

		d ₁ d ₀			
	C ₀	00	01	11	10
P ₁ P ₀	00	0	X	X	X
	01	X	0	X	X
	11	X	X	X	1
	10	X	0	X	X
		d ₁ d ₀			
	U ₂	00	01	11	10
P ₁ P ₀	00	0	X	X	X
	01	X	1	X	X
	11	X	X	X	0
	10	x	1	X	X
		d ₁ d ₀			
	D ₁	00	01	11	10
P ₁ P ₀	00	0	X	X	X
	01	X	0	X	X
	11	X	X	X	0
	10	X	1	X	X
		d ₁ d ₀			
	U ₀	00	01	11	10
P ₁ P ₀	00	1	X	X	X
	01	X	1	X	X
	11	X	X	X	0
	10	X	1	X	X

$$C_0 = d_1 \quad U_2 = d_0 \quad D_1 = P_1 \bar{P}_0 \quad U_0 = \bar{d}_1$$

El resto de funciones tienen valor 0.

27) Diseñar un circuito combinacional (Figura 1) que regule las luces de dos semáforos de tráfico colocados en una carretera en obras (Figura 2). El funcionamiento es el siguiente:

- Cuando los operarios abandonan la obra, dejan el interruptor $E=0$ y el interruptor F indiferente, y se quedan encendidas las luces ámbar de ambos semáforos de forma intermitente, si bien las funciones M_1 y M_2 (Figura 1) están permanentemente a 1.
- Cuando los operarios regulan el tráfico manualmente, ponen $E=1$:
 - Para dar paso a los vehículos que se encuentran el semáforo 1, ponen $F=0$.
 - Para dar paso a los vehículos que se encuentran el semáforo 2, ponen $F=1$.

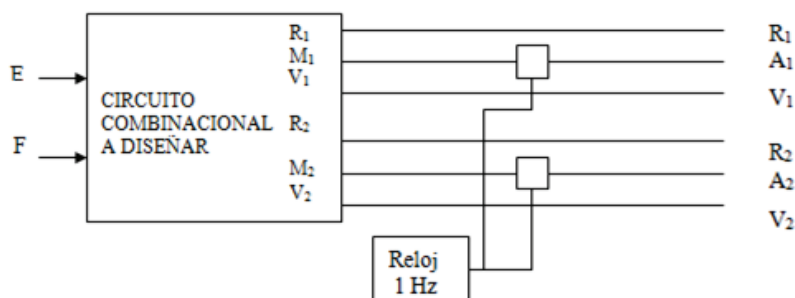


Figura 1

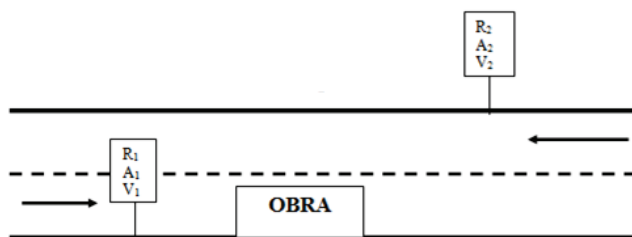


Figura 2

Se pide:

- Tabla de verdad.
- ¿Qué relación hay entre las luces ámbar de los semáforos y las salidas M_1 y M_2 del circuito combinacional?
- ¿Cuántas funciones booleanas realmente distintas hay en el problema?
- Implementar las funciones que se hayan usado con sólo puertas lógicas.

- e) Implementar las funciones que se hayan usado con decodificadores con salidas activas a nivel alto.
- f) Implementar las funciones que se hayan usado con decodificadores con salidas activas a nivel bajo.
- g) Implementar las funciones que se hayan usado con multiplexores con 1 línea de selección y conectando la entrada E a dicha línea de selección.
- h) Implementar las funciones que se hayan usado con multiplexores con 1 línea de selección y conectando la entrada F a dicha línea de selección.

- a) La tabla de verdad es la siguiente:

E	F	V_2	M_2	R_2	V_1	M_1	R_1
0	0	0	1	0	0	1	0
0	1	0	1	0	0	1	0
1	0	0	0	1	1	0	0
1	1	1	0	0	0	0	1

- b) Llamando A_1 y A_2 a las luces ámbar, $A_1 = M_1$ AND Reloj y $A_2 = M_2$ AND Reloj.

- c) Según la tabla de verdad se puede ver que realmente hay únicamente tres funciones distintas: 1ª) $R_1 = V_2$, 2ª) $R_2 = V_1$ y 3ª) $M_1 = M_2$

$$d) R_1 = V_2 = E \cdot F \quad R_2 = V_1 = E \cdot \overline{F} \quad M_1 = M_2 = \overline{E}$$

- e) $M_1 = M_2$ es la OR de las salidas 0 y 1 de un DEC 2x4 en cuyas entradas se colocan E en 1 y F en 0.

$V_1 = R_2$ es la salida 2 de un DEC 2x4 en cuyas entradas se colocan E en 1 y F en 0.

$R_1 = V_2$ es la salida 3 de un DEC 2x4 en cuyas entradas se colocan E en 1 y F en 0.

- f) $M_1 = M_2$ es la NAND de las salidas 0 y 1 de un DEC 2x4 con entradas E en 1 y F en 0.

$V_1 = R_2$ es la salida 2 de un DEC 2x4 seguida de un inversor con entradas E en 1 y F en 0.

$R_1 = V_2$ es la salida 3 de un DEC 2x4 seguida de un inversor con entradas E en 1 y F en 0.

- g) $M_1 = M_2$ es la salida de un MUX 2x1 a la que llega un 1 por la entrada 0 y 0 por la entrada 1.

$V_1 = R_2$ es la salida de un MUX 2x1 a la que llega un 0 por la entrada 0 y $\overline{F}$ por la entrada 1.

$R_1 = V_2$ es la salida de un MUX 2x1 a la que llega un 0 por la entrada 0 y F por la entrada 1.

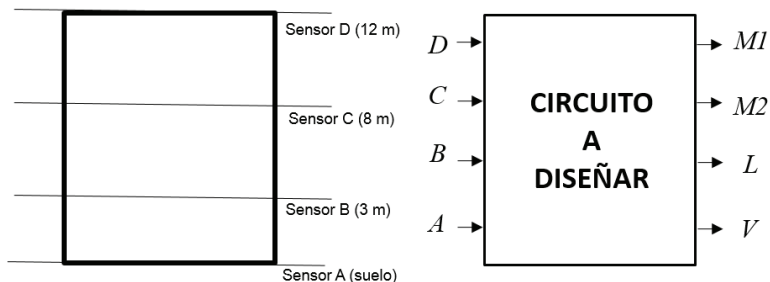
h) $M_1 = M_2$ es la salida de un MUX 2x1 a la que llega $\overline{E}$ por las entradas 0 y 1.

$V_1 = R_2$ es la salida de un MUX 2x1 a la que llega E por la entrada 0 y un 0 por la entrada 1.

$R_1 = V_2$ es la salida de un MUX 2x1 a la que llega un 0 por la entrada 0 y E por la entrada 1.

- 28) Un depósito de agua tiene cuatro sensores colocados verticalmente que indican si el nivel del líquido supera la posición del sensor. El sensor más bajo (*A*) está a nivel del suelo; el siguiente (*B*) a tres metros de altura, el siguiente (*C*) a ocho metros y el más alto (*D*) a doce metros. Cuando el nivel de agua iguala o supera la posición de un sensor, éste da salida lógica 1 y en caso contrario la salida lógica es 0.

Del depósito está constantemente saliendo agua y tiene dos motores *M1* y *M2* que se puede activar de forma independiente para llenar el depósito. Además, tiene dos alarmas, una que se activa cuando el depósito está totalmente lleno (*L*) y otra que se activa cuando el depósito está totalmente vacío (*V*).



Se trata de diseñar la lógica de control que regule el funcionamiento de los motores y las alarmas en función de la salida de los sensores, siguiendo las siguientes especificaciones:

- Cuando se alcanza el nivel máximo de doce metros, ambos motores deben apagarse y activarse la alarma *L*.
- Cuando el nivel se encuentra entre ocho y doce metros, debe activarse sólo el motor *M1*.

- Cuando el nivel se encuentra entre tres y ocho metros, debe activarse sólo el motor $M2$.
- Cuando el nivel descende por debajo de los tres metros, deben activarse ambos motores. Además, si el depósito queda totalmente vacío deben seguir activados los dos motores y la alarma V .

El encendido de los motores se consigue poniendo la salida correspondiente de la lógica de control a 1, mientras que el apagado tiene lugar cuando dicha salida se pone a 0. Se pide:

- Tabla de verdad que resuelve el problema.
- Implementación de las funciones $M1$ y $M2$ con puertas lógicas con estructura AND-OR (suma de productos).
- Implementación de las funciones L y V con puertas lógicas con estructura OR-AND (producto de sumas).
- Implementación de $M1$ usando un MUX 4x1 colocando en las líneas de selección las variables D y C , tomando C como la variable más significativa.
- Implementación de $M2$ con un DEC 2x4 colocando en las entradas las variables D y C , tomando D como la variable más significativa.

NOTA: En los apartados b) y c) no es necesario dibujar el circuito. Es suficiente con escribir la expresión algebraica de las funciones pedidas. En los apartados d) y e) hay que encontrar la solución más sencilla posible.

- Tabla de verdad:

D	C	B	A	M1	M2	L	V
0	0	0	0	1	1	0	1
0	0	0	1	1	1	0	0
0	0	1	0	X	X	X	X
0	0	1	1	0	1	0	0
0	1	0	0	X	X	X	X
0	1	0	1	X	X	X	X
0	1	1	0	X	X	X	X
0	1	1	1	1	0	0	0
1	0	0	0	X	X	X	X
1	0	0	1	X	X	X	X
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	0	0	1	0

b) y c)

Implementación de las funciones $M1$ y $M2$ con AND-OR y L y V con puertas lógicas con estructura OR-AND.

		BA			
M1		00	01	11	10
DC	00	1	1	0	X
	01	X	X	1	X
	11	X	X	0	X
	10	X	X	X	X

		BA			
M2		00	01	11	10
DC	00	1	1	1	X
	01	X	X	0	X
	11	X	X	0	X
	10	X	X	X	X

		BA			
L		00	01	11	10
DC	00	0	0	0	X
	01	X	X	0	X
	11	X	X	1	X
	10	X	X	X	X

		BA			
V		00	01	11	10
DC	00	1	0	0	X
	01	X	X	0	X
	11	X	X	0	X
	10	X	X	X	X

$$M1 = \bar{B} + C \cdot \bar{D} \quad M2 = \bar{C}$$

$$L = D \quad V = \bar{A}$$

d) Implementación de $M1$ usando un MUX 4x1, tomando C y D como variables de selección.

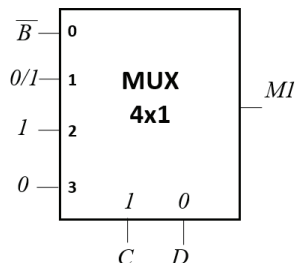
Formando las siguientes tablas se obtienen las expresiones de las entradas al multiplexor.

C=0 D=0		
B	A	E ₀
0	0	1
0	1	1
1	0	X
1	1	0

C=0 D=1		
B	A	E ₁
0	0	X
0	1	X
1	0	X
1	1	X

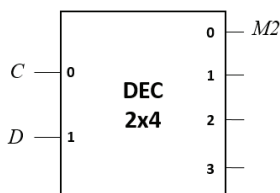
C=1 D=0		
B	A	E ₂
0	0	X
0	1	X
1	0	X
1	1	1

C=1 D=1		
B	A	E ₃
0	0	X
0	1	X
1	0	X
1	1	0



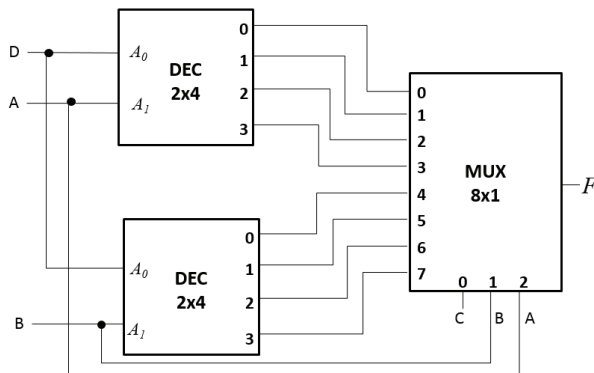
e) Implementación de $M2$ con un DEC 2x4, colocando en las entradas las variables D y C , tomando D como la variable más significativa.

Mirando el mapa de Karnaugh de M_2 y tomando valor 1 en la casilla número 2, podemos llegar a que $M_2 = \overline{C} \overline{D}$. Luego la solución es:



29) Dado el circuito de la figura, compuesto por dos decodificadores 2x4 y un multiplexor 8x1:

- Escribir la tabla de verdad que define el comportamiento del circuito. La entrada A es la más significativa y la D la menos significativa.
- Obtener la expresión algebraica equivalente con puertas NOR.



a) Tabla de verdad.

A	B	C	D	Salida activa DEC arriba	=	Entrada MUX activada por el DEC de arriba	Salida activa DEC abajo	Entrada MUX Activada por el DEC abajo	Entrada MUX seleccionada	F
0	0	0	0	0		0	0	4	0	1
0	0	0	1	1		1	1	5	0	0
0	0	1	0	0		0	0	4	1	0
0	0	1	1	1		1	1	5	1	1
0	1	0	0	0		0	2	6	2	0
0	1	0	1	1		1	3	7	2	0
0	1	1	0	0		0	2	6	3	0
0	1	1	1	1		1	3	7	3	0
1	0	0	0	2		2	0	4	4	1
1	0	0	1	3		3	1	5	4	0
1	0	1	0	2		2	0	4	5	0
1	0	1	1	3		3	1	5	5	1
1	1	0	0	2		2	2	6	6	1
1	1	0	1	3		3	3	7	6	0
1	1	1	0	2		2	2	6	7	0
1	1	1	1	3		3	3	7	7	1

Mapa de Karnaugh. Variable A más significativa y D menos significativa.

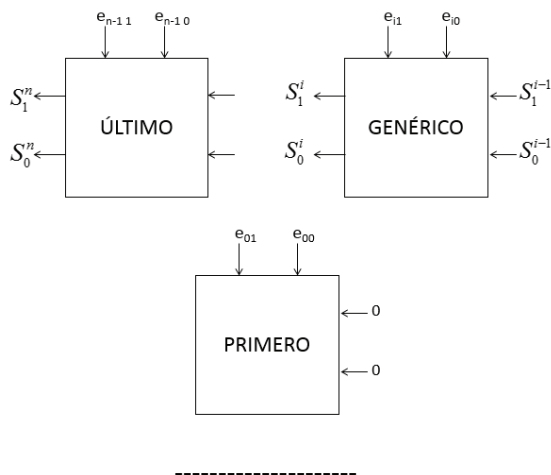
		CD			
		00	01	11	10
AB	00	1	0	1	0
	01	0	0	0	0
	11	1	0	1	0
	10	1	0	1	0

b) Expresión algebraica

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

30) Suponer un sistema de n entradas $e_{n-1}, \dots, e_1, e_0$, siendo e_i un elemento del conjunto $E=\{0,1,2,3\}$. Diseñar mediante una red iterativa un sistema cuya salida proporcione la mayor de las n entradas. Hazlo de forma detallada indicando:

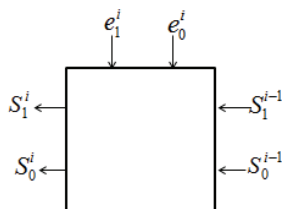
- Número de módulos y especificación de alto nivel del módulo i -ésimo y del último módulo.
- Especificación de bajo nivel con la codificación adoptada.
- Expresiones de las salidas del módulo.
- Diagrama lógico.



En realidad el sistema proporciona el mayor de los números introducidos, $E=\{0,1,2,3\}$, luego las entradas e_i se codifican en $e_{i,1}$ $e_{i,0}$.

- Hay n módulos iguales.
- Entrada: El número correspondiente e_i y la información procedente que proporcionará el mayor de los números hasta ese módulo S_{i-1} .

Salida: Resultado de comparar S_{i-1} con e_i , la salida será el mayor de los 2 números.



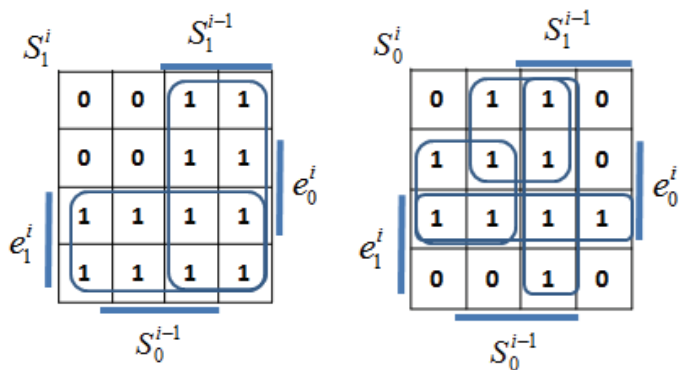
e^i	$e_1^i \ e_0^i$	S^i	$S_1^i S_0^i$
0	00	0	00
1	01	1	01
2	10	2	10
3	11	3	11

c) Expresiones de las salidas:

La tabla de verdad es la siguiente:

e_1^i	e_0^i	S_1^{i-1}	S_0^{i-1}	S_1^i	S_0^i
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

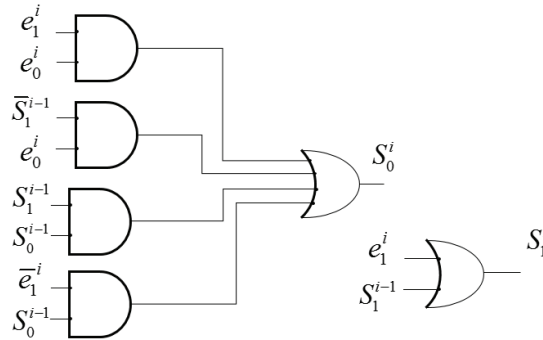
Utilizando MK se obtienen las expresiones de las salidas.



$$S_1^i = e_1^i + S_1^{i-1};$$

$$S_0^i = e_1^i e_0^i + \bar{S}_1^{i-1} e_0^i + S_1^{i-1} S_0^{i-1} + \bar{e}_1^i S_0^{i-1}$$

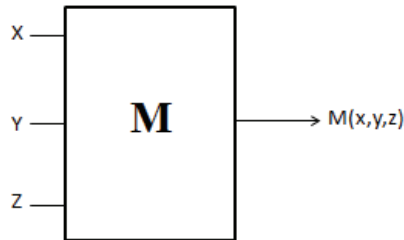
d) Diagrama lógico:



31) La función mayoría sobre tres variables viene dada por:

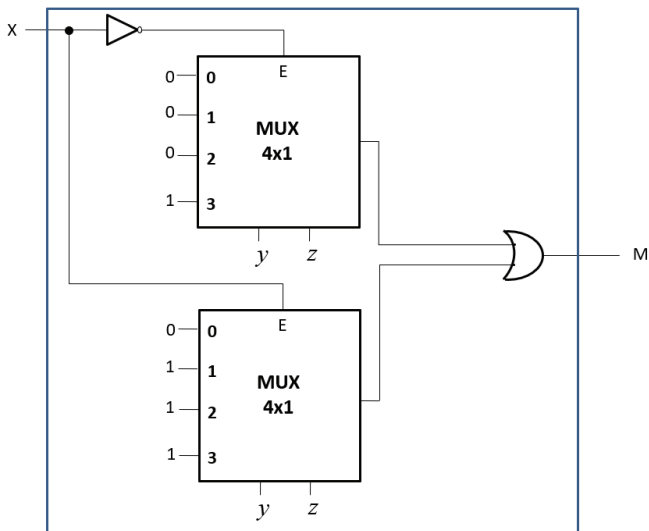
$$M(x, y, z) = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz$$

(Tomará valor 0 si hay mayoría de ceros, y 1 si hay mayoría de unos). Siendo un bloque funcional:



- Implementar las funciones AND y OR de dos variables con bloques M y las constantes lógicas 0 y 1.
- Diseñar el bloque M mediante multiplexores de 8 y de 4 entradas y el mínimo número de puertas. (Considerar que el multiplexor está activo cuando su entrada de validación E es 1).

- Simplificando la función se llega a:



- 32) Se desea diseñar un circuito combinacional (Figura 1) excitado por un contador de 3 bits y cuyas salidas se conecten a tres displays de 7 segmentos, cada uno de los cuales presenta el carácter BCD que se le aplique a su entrada, para que en ellos aparezca la siguiente presentación que se aprecia en la Figura 2: con cuenta 0 todos apagados, con cuenta 1 se enciende el 1 en DIS1, con cuenta 2 se enciende 1 en DIS2 y un 2 en DIS1, etc... Se supone que cada display lleva incorporado su propio codificador, que, por tanto, no es necesario diseñar. Cualquier número binario mayor o igual que 12 mantiene el display de 7 segmentos totalmente apagado, mientras que los números 10 y 11 presentan respectivamente las letras *A* y *U*.

Realizar el diseño usando decodificadores de 3 entradas (uno con salidas activas a nivel alto y otro con salidas activas a nivel bajo, conectadas sus entradas en paralelo), 6 puertas OR y 6 puertas AND. Ninguna de las puertas puede tener más de 3 entradas. Para implementar cada función booleana se usará sólo una puerta.

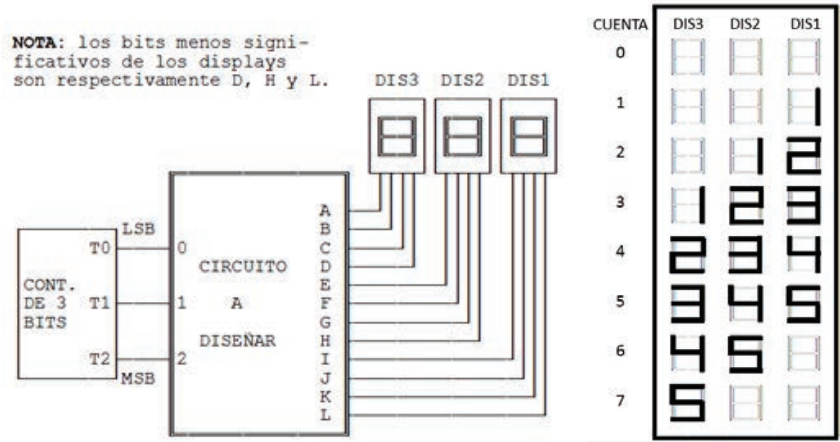
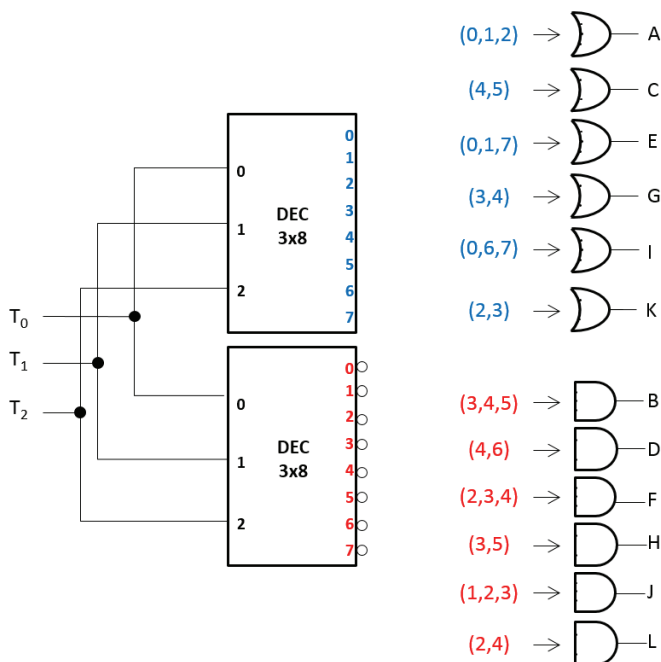


Figura 1

Figura 2

T ₂	T ₁	T ₀	A	B	C	D	E	F	G	H	I	J	K	L
0	0	0	1	1	X	X	1	1	X	X	1	1	X	X
0	0	1	1	1	X	X	1	1	X	X	0	0	0	1
0	1	0	1	1	X	X	0	0	0	1	0	0	1	0
0	1	1	0	0	0	1	0	0	1	0	0	0	1	1
1	0	0	0	0	1	0	0	0	1	1	0	1	0	0
1	0	1	0	0	1	1	0	1	0	0	0	1	0	1
1	1	0	0	1	0	0	0	1	0	1	1	1	X	X
1	1	1	0	1	0	1	1	1	X	X	1	1	X	X

Según el número de ceros y unos de cada función, se opta por implementar con el codificador de salida a nivel alto con puertas OR o con salida activa a nivel bajo con puertas AND



33) Un técnico de luminotecnia debe encargarse de encender los focos de iluminación del escenario de un teatro. Para ello dispone de un circuito combinacional a diseñar con 4 interruptores de entrada I_1 , I_2 , I_3 , I_4 , y cuatro salidas R , V , A , M que encienden los correspondientes focos de colores respectivos Rojo, Verde, Azul y Marrón. Las especificaciones de dicho circuito combinacional son las siguientes:

- Cuando esté el actor principal en el escenario debe activar dos interruptores cualesquiera que no sean consecutivos, de forma que se enciendan todas las luces mientras este actor se encuentre sólo en el escenario.
- Cuando suban los restantes actores debe haber tres interruptores cualesquiera activados para que se encienda las luces Roja, Verde y Azul.
- Si todos los interruptores están apagados o si se activan dos interruptores consecutivos, todos focos están apagados.
- Cuando el escenario se quede sin actores, entre actos o antes de empezar la obra, debe encenderse sólo la luz roja, para lo cual es necesario tener sólo un interruptor activado, el que sea.
- Cuando finalice la obra y salgan todos los actores a saludar al público, deben encender todos los focos excepto el rojo. Para ello deberán estar todos los interruptores activados.

Se pide diseñar el circuito combinacional, escribiendo previamente la tabla de verdad y llevar a cabo la implementación usando:

- a) Sólo puertas NOR para la función V .
- b) Sólo puertas NAND para la función A .
- c) Un MUX 4x1 y las puertas necesarias para la función R , usando $I1$ e $I2$ como señales de control, de las que $I1$ es la más significativa.
- d) Un DEC 4X16 para la función M .

NOTA: Las funciones en a) y b) deben utilizar el mínimo número de puertas posible.

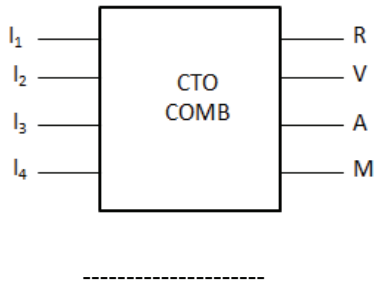


Tabla de verdad

ESTADO	CODIFICACIÓN				DEFINICIÓN
	R	A	V	M	
A	0	0	0	0	Todos los focos apagados
S	1	0	0	0	Sin actores
P	1	1	1	1	Actor principal
R	1	1	1	0	Resto de actores
F	0	1	1	1	Fin de obra

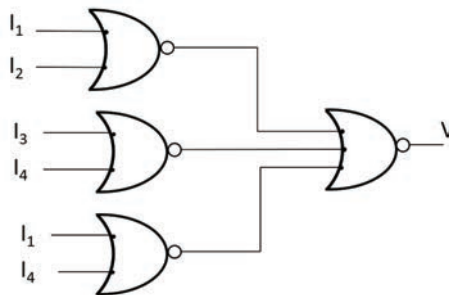
I_1	I_2	I_3	I_4	R	V	A	M	Código Situación
0	0	0	0	0	0	0	0	A
0	0	0	1	1	0	0	0	S
0	0	1	0	1	0	0	0	S
0	0	1	1	0	0	0	0	A
0	1	0	0	1	0	0	0	S
0	1	0	1	1	1	1	1	P
0	1	1	0	0	0	0	0	A
0	1	1	1	1	1	1	0	R
1	0	0	0	1	0	0	0	S
1	0	0	1	1	1	1	1	P
1	0	1	0	1	1	1	1	P
1	0	1	1	1	1	1	0	R
1	1	0	0	0	0	0	0	A
1	1	0	1	1	1	1	0	R
1	1	1	0	1	1	1	0	R
1	1	1	1	0	1	1	1	F

a) Puertas NOR para la función V :

		$I_3 I_4$			
	V	00	01	11	10
$I_1 I_2$	00	0	0	0	0
	01	0	1	1	0
	11	0	1	1	1
	10	0	1	1	1

$$V = (I_3 + I_4) \cdot (I_1 + I_2) \cdot (I_1 + I_4)$$

$$V = \overline{\overline{(I_3 + I_4)} + \overline{(I_1 + I_2)} + \overline{(I_1 + I_4)}}$$

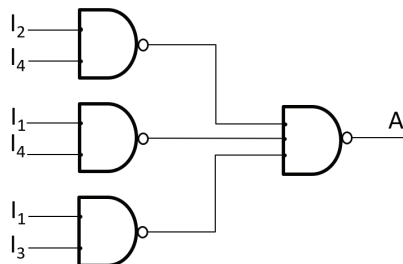


b) Puertas NAND para la función A :

		$I_3 I_4$			
A		00	01	11	10
00		0	0	0	0
$I_1 I_2$	01	0	1	1	0
	11	0	1	1	1
	10	0	1	1	1

$$V = I_2 I_4 + I_1 I_4 + I_1 I_3$$

$$V = \overline{I_2 I_4} \cdot \overline{I_1 I_4} \cdot \overline{I_1 I_3}$$

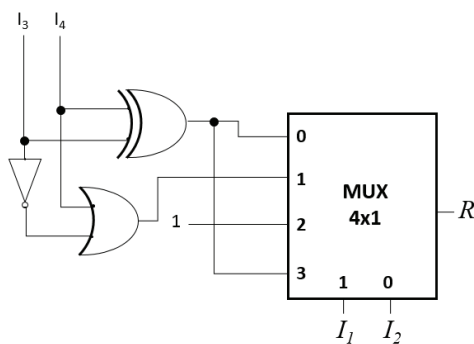


c) Implementación de R con un MUX 4x1, usando I_1 e I_2 como señales de control.

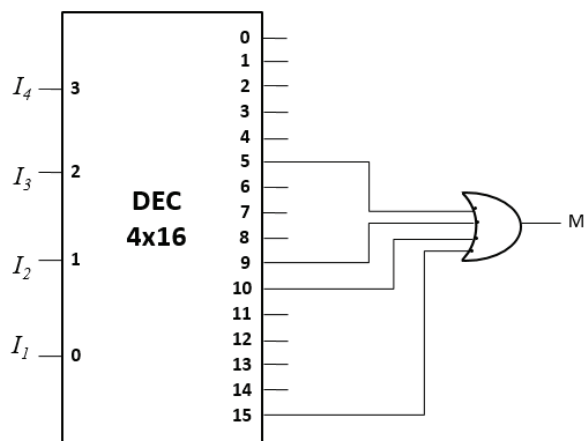
Las entradas al multiplexor se obtienen a partir de las tablas siguientes:

$I_1=0 \ I_2=0$			$I_1=0 \ I_2=1$			$I_1=1 \ I_2=0$			$I_1=1 \ I_2=1$		
I_3	I_4	E_0	I_3	I_4	E_1	I_3	I_4	E_2	I_3	I_4	E_3
0	0	0	0	0	1	0	0	1	0	0	0
0	1	1	0	1	1	0	1	1	0	1	1
1	0	1	1	0	0	1	0	1	1	0	1
1	1	0	1	1	1	1	1	1	1	1	0

$$E_0 = I_3 \oplus I_4, \ E_1 = \overline{I_3} + I_4, \ E_2 = 1, \ E_3 = I_3 \oplus I_4$$



d) Un DEC 4x16 para la función M :



TEST Y PREGUNTAS CORTAS

1) ¿Cuál de la siguiente pareja de módulos combinacionales son realmente lo mismo (a nivel de puertas)?

- ☐ A El codificador y el demultiplexor.
- ☐ B El decodificador y el multiplexor.
- ☐ C El decodificador y el demultiplexor.

2) Si una función booleana de 4 variables tiene en su tabla de verdad 12 ceros y 4 unos, ¿cómo sería su implementación más sencilla?

- ☐ A Con un DEC con salidas activas a nivel alto y una puerta NOR.
- ☐ B Con un DEC con salidas activas a nivel bajo y una puerta AND.
- ☐ C Con un DEC con salidas activas a nivel bajo y una puerta NAND.

3) ¿Cuál de los siguientes elementos es el adecuado para elegir una información digital de un origen determinado?

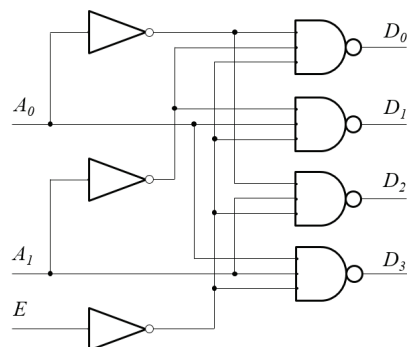
- ☐ A Un multiplexor.
- ☐ B Un demultiplexor.
- ☐ C Un decodificador.

4) Señala la opción correcta

- ☐ A Un DEC 4x16 tiene el mismo número de puertas AND que un MUX 16x1.
- ☐ B Un conversor de código es normalmente implementado únicamente con módulos Multiplexores.
- ☐ C Un demultiplexor es utilizado habitualmente como selector de información.

5) Identifica el circuito de la figura

- ☐ A Detector de paridad de 4 bits
- ☐ B Decodificador 2x4 con salida invertida y con enable
- ☐ C Decodificador 3x4

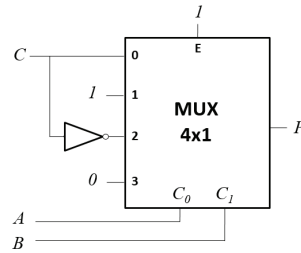


6) Indica la función de salida del circuito de la figura:

☐ A $F = \overline{B}C + \overline{A}B + \overline{A}B\overline{C}$

☐ B $F = \overline{B}A + \overline{C}B + \overline{A}C$

☐ C $F = \overline{C}A + B$

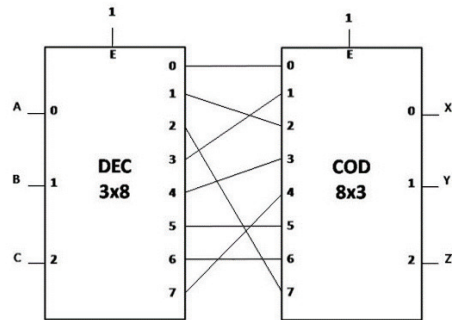


7) El circuito de la figura señala la opción correcta.

☐ A $X=1, Y=1, Z=1$ siempre.

☐ B Si $A=0, B=1, C=1$ entonces $X=0, Y=1, Z=1$

☐ C Si $A=1, B=1, C=0$ entonces $X=0, Y=1, Z=1$

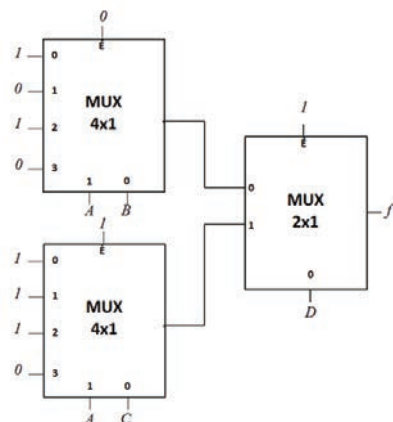


8) En el circuito de la figura señala la opción correcta.

☐ A La función f no depende de la variable B .

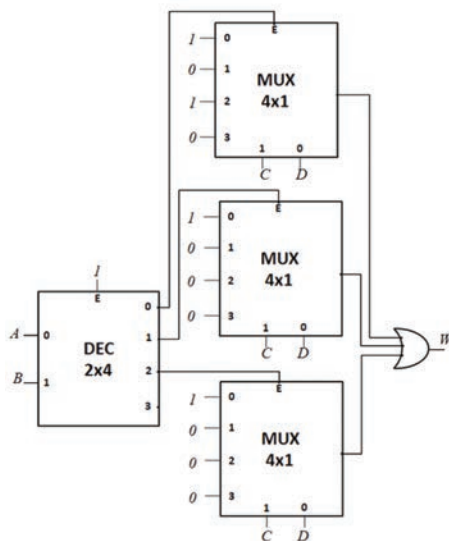
☐ B $f = ACD$

☐ C $f = 0$



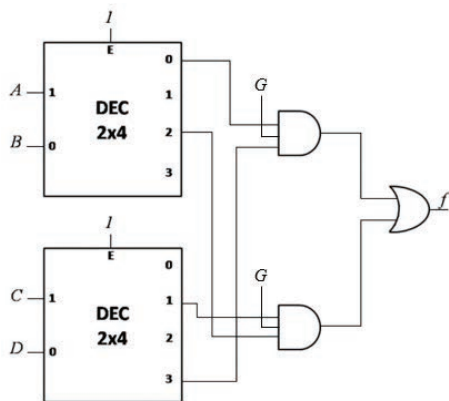
9) En la figura, señala la opción correcta.

- ☐ A La función de salida W vale 0 si A , B , C y D son 0.
☐ B La función de salida W vale 0 si A y B son 1.
☐ C La función de salida W vale siempre 1 si $A=1$ y $B=0$.



10) . En el circuito de la figura, para $f(A, B, C, D, G)$, marca la que consideres correcto.

- ☐ A $f(0,0,1,1,1) = 0$
☐ B $f(1,1,0,0,1) = 0$
☐ C La A) y la B) son incorrectas.

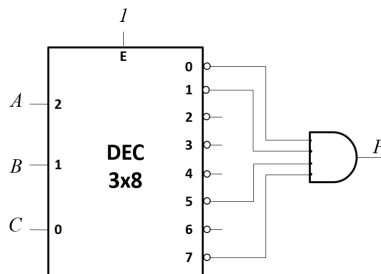


11) La primera forma canónica de la función F de la figura es:

☐ A $F = \sum m(2, 3, 4, 6)$

☐ B $F = \sum m(0, 1, 5, 7)$

☐ C $F = \prod M(2, 3, 4, 6)$

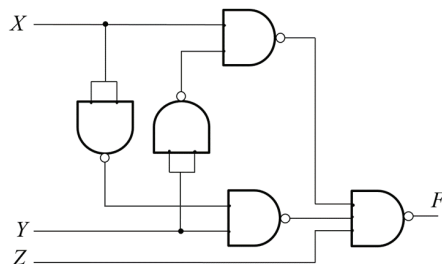


12) La función de salida del sistema de la figura adjunta es:

☐ A $F = \bar{X}\bar{Y} + \bar{X}Y + \bar{Z}$

☐ B $F = X\bar{Y} + \bar{X}Y + \bar{Z}$

☐ C Ninguna de las anteriores.

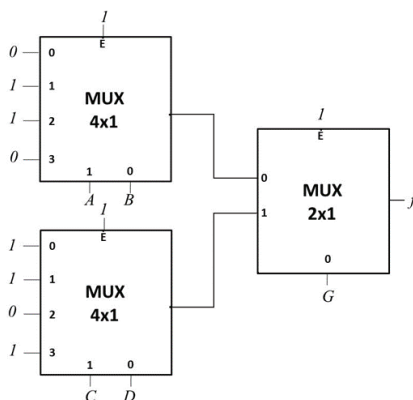


13) En el circuito de la figura, para $f(A, B, C, D, G)$, marca la opción que consideres correcta.

☐ A $f(0, 0, 1, 1, 1) = 0$

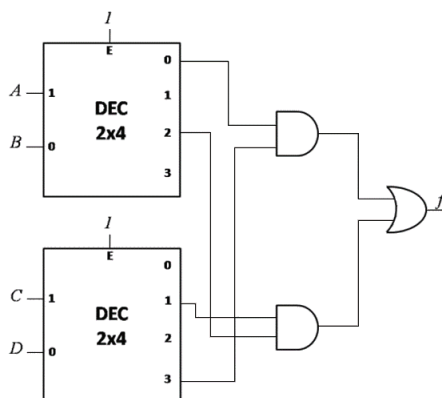
☐ B $f(1, 1, 0, 0, 1) = 0$

☐ C La ☐ A y la ☐ B son incorrectas.



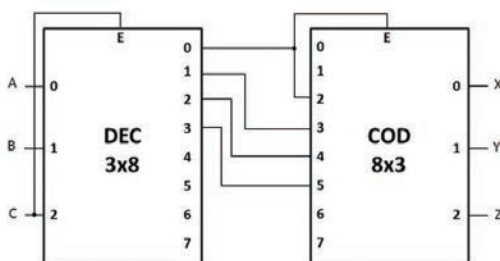
14) En el circuito de la figura adjunta marca la opción correcta.

- ☐ A $f(A, B, C, D) = \sum m(0, 2, 5, 7)$
☐ B $f(A, B, C, D) = 0$
☐ C $f(A, B, C, D) = \sum m(3, 9)$



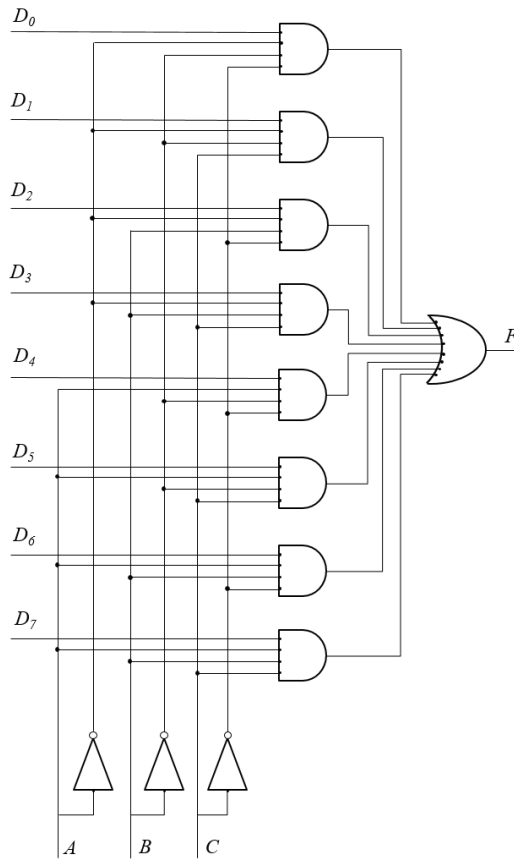
15) Señala la respuesta correcta en referencia al circuito de la figura adjunta.

- ☐ A La salida es la representación en binario del número de entrada más dos unidades si números de entrada son menores que 4.
☐ B Las salidas son siempre 0 0 0.
☐ C Las salidas son siempre 0 1 0.



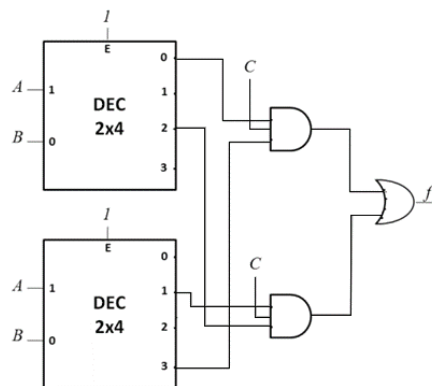
16) ¿Qué módulo representa la siguiente figura?

- ☐ A Un multiplexor de tres entradas de selección.
☐ B Un decodificador de 8 canales de entrada y salida no invertida.
☐ C Un multiplexor de 3 entradas de datos y salida no invertida.



17) En la figura adjunta, señala la opción correcta.

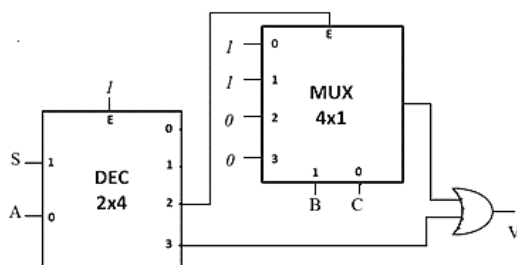
- ☐ A El circuito corresponde a un DEC 3x8.
- ☐ B La función de salida es $f = AB + C$
- ☐ C La función de salida es $f=0$



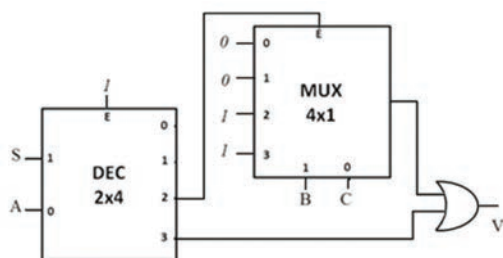
18) Señala cuál de los circuitos implementa la función

$$V(S, A, B, C) = \sum m(8, 9, 12, 13, 14, 15)$$

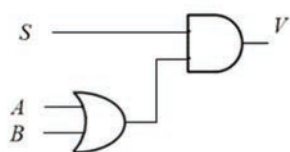
A



B

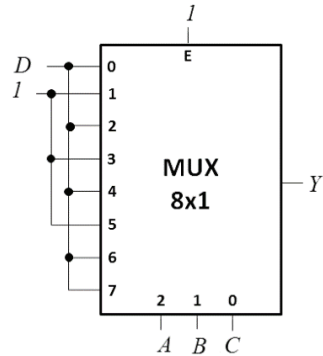


C



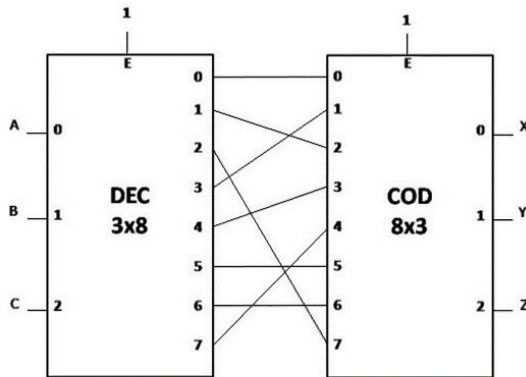
19) Sobre el circuito de la figura adjunta, señala la opción correcta:

- ☐ A La salida Y es 0 si $D = 0$.
- ☐ B $Y(A, B, C, D) = \prod M(0, 4, 8, 12, 14)$
- ☐ C $Y(A, B, C, D) = \prod M(0, 2, 4, 6, 7)$



20) El circuito de la figura adjunta señala la opción correcta

- ☐ A $X=0, Y=1, Z=1$ siempre.
- ☐ B Si $A=0, B=1, C=1$ entonces $X=0, Y=0, Z=1$.
- ☐ C Si $A=1, B=0, C=0$ entonces $X=0, Y=1, Z=0$.



21) ¿Con cuál de las siguientes opciones se consigue realizar un MUX 16x1?

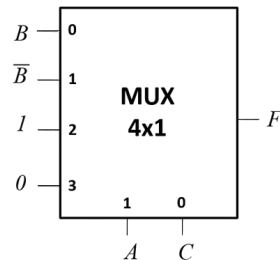
- ☐ A Con 8 MUX 2x1.
- ☐ B Con 4 MUX 4x1.
- ☐ C Con 5 MUX 4x1.

22) ¿Cuál de los siguientes pares de módulos combinacionales se usarían para transmitir por un único cable un dato de varios bits, en distintos instantes de tiempo?

- ☐ A Codificador-Decodificador.
☐ B Decodificador-Demultiplexor.
☐ C Multiplexor-Demultiplexor.

23) Dada la figura adjunta, ¿cuál es la expresión de la salida F ?

- ☐ A $F = \bar{B} \cdot C + A \cdot C + \bar{A} \cdot B \cdot C$
☐ B $F = A \cdot \bar{C} + \bar{B} \cdot \bar{C} + \bar{A} \cdot B \cdot C$
☐ C Ninguna de las anteriores.

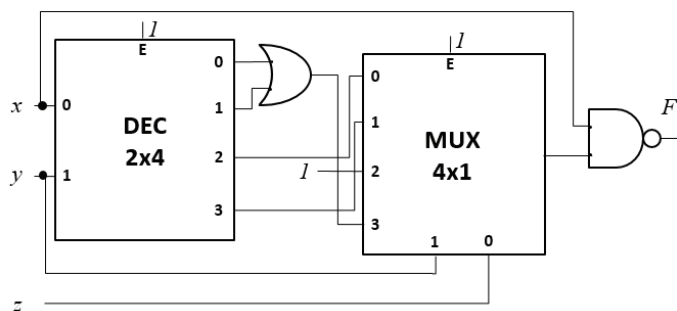


24) Indicar qué tipo de puerta conviene utilizar (por tener menos entradas) para implementar una función booleana cuya tabla de verdad tiene pocos ceros, si se usa un decodificador con salidas activas a nivel bajo.

- ☐ A NOR.
☐ B NAND.
☐ C AND.

25) Analiza el siguiente circuito y di cuál es la expresión booleana de salida.

- ☐ A $F = 1$
☐ B $F = x \cdot y \cdot \bar{z}$
☐ C $F = \bar{x} + \bar{y} + z$



26) ¿Cuántos multiplexores 2x1 son necesarios para construir un multiplexor 16x1?

- ☐ A 15
- ☐ B 16
- ☐ C 9

5

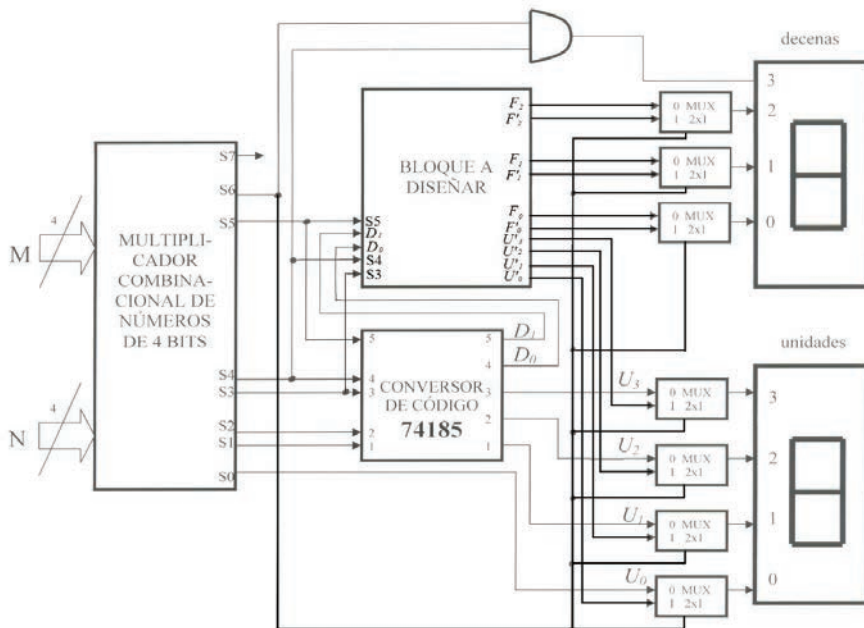
CIRCUITOS ARITMÉTICOS Y LÓGICOS

CONTENIDOS:

- Sumadores: binarios, BCD.
- Restadores.
- Otros circuitos: comparadores, generadores de paridad, conversores de código y multiplicadores.
- ALUs.

EJERCICIOS RESUELTOS

- 1) Se trata de diseñar parte de un circuito combinacional para enseñar la tabla de multiplicar a niños. El circuito en su conjunto multiplica dos números M y N de una cifra decimal, expresados ambos en BCD, y proporciona el resultado con dos cifras decimales, decenas y unidades expresados también en BCD en sendos displays de 7 segmentos.



Para ello se dispone de un circuito cuyo diagrama general es el de la Figura adjunta. En la resolución del problema, hay que diseñar el bloque que excita el display de las decenas en función de algunas de las salidas del multiplicador directamente o después de pasar por un convertor de código. El bloque a diseñar tiene 10 salidas, $F_2, F_2', F_1, F_1', F_0, F_0', U_3, U_2, U_1$ y U_0 . El multiplicador combinacional acepta dos dígitos en BCD y pone el resultado en sus 8 líneas de salida. Como el mayor resultado posible es 81, la salida S_7 siempre es 0 y no se usa.

El convertor de código 74185 convierte la palabra binaria de 6 bits S_5, S_4, S_3, S_2, S_1 y S_0 en dos palabras, una de dos bits (D_1 y D_0), que codifica las decenas, y otra de cuatro bits (U_3, U_2, U_1 y U_0), que codifica las unidades del valor decimal de la palabra binaria de entrada. Por ejemplo, si entra 24, sale un 2 por las decenas

y un 4 por las unidades. El mayor número que se puede representar es, por tanto, 39.

Si el número es mayor o igual a 40 y menor o igual que 63, codifican en D_1 y D_0 las decenas disminuidas en 4 unidades, con dos bits. Es decir, codifica el número binario introducido disminuido en 40. Por ejemplo, si se introduce el número 56, sale un 1 por las decenas y un 6 por las unidades.

Por último, si el número es mayor o igual a 64, el número binario que entra al conversor de código se minora en 64, ya que sus seis bits menos significativos son iguales. Por ejemplo, si el resultado es 81, es igual que si le introducimos $81-64 = 17$, con lo que sale un 1 por las decenas y un 7 por las unidades. En este último caso, cuando el resultado iguala o supera 64, las entradas a los multiplexores de los displays son las funciones $F_2', F_1', F_0', U_3', U_2', U_1'$ y U_0' en vez de las funciones $F_2, F_1, F_0, U_3, U_2, U_1$ y U_0 .

Se pide:

- Rellenar la tabla 1, poniendo en binario los números pedidos y viendo qué valores aparecen en A y B para todos los casos.
- Rellenar las tablas 2, 3 y 4 que permiten especificar completamente el circuito combinacional a diseñar en su conjunto.
- Implementar las funciones $F_2, F_1, F_0, U_3, U_2, U_1$ y U_0 de la forma más sencilla posible en función de las salidas del multiplicador.

a) Tabla 1

Resultado	S6	C=S5	S4	S3	S2	S1	S0	B=D ₁	A=D ₀	
0 0	0	0	0	0	0	0	0	0	0	
0 1	0	0	0	0	0	0	1	0	0	
0 2	0	0	0	0	0	1	0	0	0	
0 3	0	0	0	0	0	1	1	0	0	
0 4	0	0	0	0	1	0	0	0	0	
0 5	0	0	0	0	1	0	1	0	0	
0 6	0	0	0	0	1	1	0	0	0	
0 7	0	0	0	0	1	1	1	0	0	
0 8	0	0	0	1	0	0	0	0	0	
0 9	0	0	0	1	0	0	1	0	0	
1 0	0	0	0	1	0	1	0	0	1	
1 2	0	0	0	1	1	0	0	0	1	
1 4	0	0	0	1	1	1	0	0	1	
1 5	0	0	0	1	1	1	1	0	1	
1 6	0	0	1	0	0	0	0	0	1	
1 8	0	0	1	0	0	1	0	0	1	
2 0	0	0	1	0	1	0	0	1	0	
2 1	0	0	1	0	1	0	1	1	0	
2 4	0	0	1	1	0	0	0	1	0	
2 5	0	0	1	1	0	0	1	1	0	
2 7	0	0	1	1	0	1	1	1	0	
2 8	0	0	1	1	1	0	0	1	0	
3 0	0	0	1	1	1	1	0	1	1	
3 2	0	1	0	0	0	0	0	1	1	
3 5	0	1	0	0	0	1	1	1	1	
3 6	0	1	0	0	1	0	0	1	1	
4 0	0	1	0	1	0	0	0	0	0	
4 2	0	1	0	1	0	1	0	0	0	
4 5	0	1	0	1	1	0	1	0	0	
4 8	0	1	1	0	0	0	0	0	0	
4 9	0	1	1	0	0	0	1	0	0	
5 4	0	1	1	0	1	1	0	0	1	
5 6	0	1	1	1	0	0	0	0	1	
6 3	0	1	1	1	1	1	1	1	0	
6 4	1	0	0	0	0	0	0	0	0	→ 64-64=0 (decenas=0)
7 2	1	0	0	1	0	0	0	0	0	→ 72-64=8 (decenas=0)
8 1	1	0	1	0	0	0	1	0	1	→ 81-64=17 (decenas=1)

b)

Tabla 2

C	B	A	F2	F1	F0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	0	1	1

Tabla 3

S4	S3	F'2	F'1	F'0
0	0	1	1	0
0	1	1	1	1
1	0	0	0	0
1	1	X	X	X

Tabla 4

S4	S3	U'3	U'2	U'1	U'0
0	0	0	1	0	0
0	1	0	0	1	0
1	0	0	0	0	1
1	1	X	X	X	X

c) Simplificando, queda:

$$F_2 = C \cdot \overline{A \cdot B}$$

$$F_1 = B$$

$$F_0 = A$$

$$F'_2 = \overline{S_4}$$

$$F'_1 = \overline{S_4}$$

$$F'_0 = S_3$$

$$U'_3 = 0$$

$$U'_2 = \overline{S_3 + S_4}$$

$$U'_1 = S_3$$

$$U'_0 = S_4$$

Se puede observar que todos los casos en que las decenas valen 5, por ejemplo, tiene

Se puede observar que todos los casos cuyas decenas son 5, por ejemplo, tiene $S_5=C=1$, $B=0$ y $A=1$. Por tanto, para $CBA=101$, $F_2F_1F_0=101$. Y así se haría con los restantes casos.

Se produce un desdoblamiento sólo cuando el valor de las decenas vale 3. Por eso hay dos casos que dan salida 011, las entradas 011 y 111.

Cuando el resultado es mayor o igual que 64, el display de las decenas debe visualizar 6, 7 o 0. En este último caso la puerta AND hace llegar el 1000 (8) al display.

Cuando el resultado es mayor o igual que 64, el display de las unidades debe presentar, respectivamente, 4 (para 64), 2 (para 72) y 1 (para 81).

2) Se dispone de una ALU descrita en la tabla adjunta con tres entradas S_2, S_1, S_0 que seleccionan la función a realizar y una cuarta entrada M que distingue si la función es lógica o aritmética. Diseñar un circuito que realice las cuatro funciones siguientes de forma secuencial y cíclica: $F1=A+B$; $F2=A+B'$; $F3=(A+B)$ MAS AB' ; $F4=A$ XOR B con las siguientes herramientas disponiendo siempre de una ALU y de módulos contadores necesarios.

NOTA: "MAS" significa SUMA ARITMETICA, mientras que "+" significa SUMA LOGICA. El signo "-" es la resta aritmética.

- a) Con puertas lógicas.
- b) Con módulos combinacionales: decodificadores y codificadores.

$S_2S_1S_0$	M=1 Funciones Lógicas	M=0 Funciones Aritméticas
000	$F=A'$	$F=A$
001	$F=(A+B)'$	$F=A+B$
010	$F=A' B$	$F=A+B'$
011	$F=0$	$F=A-1$
100	$F=(AB)'$	$F=A$ MAS AB'
101	$F=B'$	$F=(A+B)$ MAS (AB')
110	$F=A$ XOR B	$F= A-B-1$
111	$F= AB'$	$F= AB' -1$

ENTRADA DE DATOS

A

B

S2

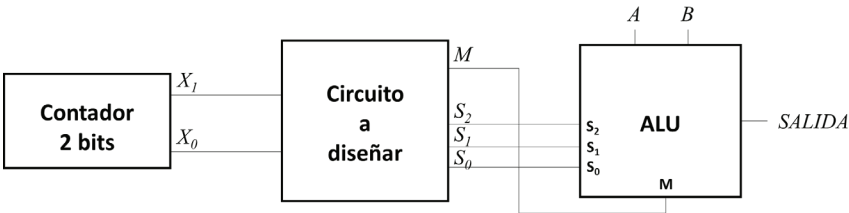
S1

S0

M

SALIDA

a) El circuito global que soluciona el problema es el siguiente:

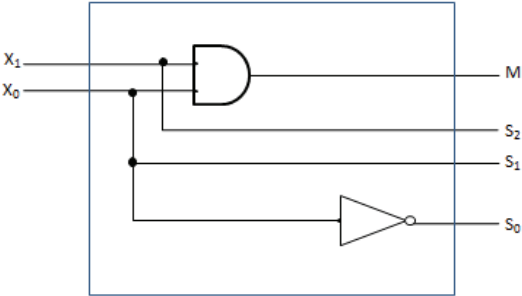


El circuito a diseñar tiene por entradas X_1, X_0 y salidas M, S_2, S_1, S_0 . De acuerdo con la tabla de la ALU, la tabla de verdad es:

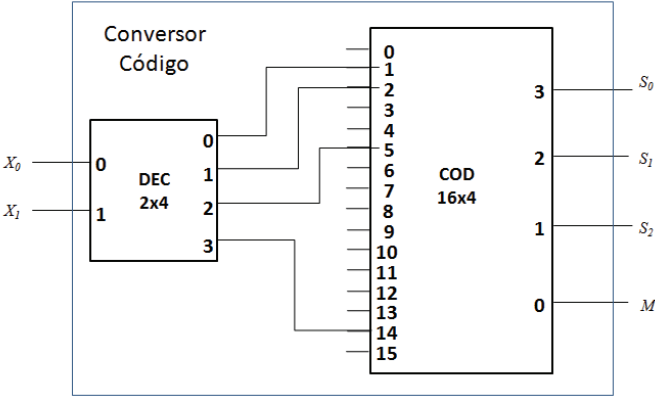
$X_1 X_0$	Operación	$M S_2 S_1 S_0$
0 0	$A+B$	0 0 0 1
0 1	$A+\overline{B}$	0 0 1 0
1 0	$(A+B)+A\overline{B}$	0 1 0 1
1 1	$A\oplus B$	1 1 1 0

Las expresiones de las funciones de salida y el circuito lógico son:

$$M = X_1 X_0$$
$$S_2 = X_1$$
$$S_1 = X_0$$
$$S_0 = \overline{X_0}$$



b) Implementación utilizando módulos combinacionales, decodificadores y codificadores:



3) Diseñar una Unidad Aritmético Lógica con las siguientes características: las entradas serán dos números $A(a_1, a_0)$ y $B(b_1, b_0)$ de 2 bits cada uno.

Las salidas $S_3 S_2 S_1 S_0$ serán:

- Caso a: Si $A > B$, ($F_1=1, F_0=0$), entonces $S_3 S_2 S_1 S_0$ es el producto aritmético $A \cdot B$.
- Caso b: Si $A = B$, ($F_1=0, F_0=0$), entonces $S_3 S_2 S_1 S_0$ igual que en el caso anterior.
- Caso c: Si $A < B$, ($F_1=0, F_0=1$), entonces $S_0 = \overline{a_0 \cdot b_1}$, $S_1 = \overline{a_1 + b_0}$, $S_2 = a_0 \cdot b_0$ y $S_3 = a_1 + b_1$.

(“+” la suma lógica, “ $\cdot$ ” el producto lógico).

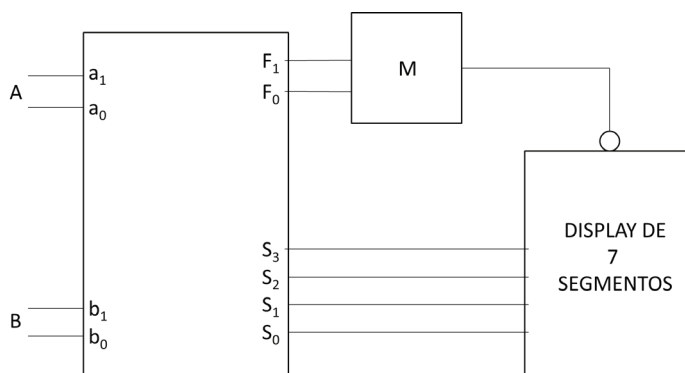
a) Encontrar la tabla de verdad que resuelve el problema.

b) Implementar las funciones con los elementos solicitados:

- F_1 usando sólo puertas NAND.
- F_0 como suma de productos.
- S_3 usando sólo puertas NOR.
- S_2 como producto de sumas.
- S_1 con un DEC 4x16 con salidas activas a nivel alto.
- S_0 con un MUX 8x1.

c) Contenido del bloque M sabiendo que el número se debe ver cuando la operación realizada haya sido una multiplicación aritmética, siendo E la señal de habilitación, activa a nivel bajo.

a) El circuito de bloques de la ALU es:



La tabla de la verdad que resuelve el problema es:

Caso	a_1	a_0	b_1	b_0	F_1	F_0	S_3	S_2	S_1	S_0	E
b	0	0	0	0	0	0	0	0	0	0	0
c	0	0	0	1	0	1	0	0	0	1	1
c	0	0	1	0	0	1	1	0	1	1	1
c	0	0	1	1	0	1	1	0	0	1	1
a	0	1	0	0	1	0	0	0	0	0	0
b	0	1	0	1	0	0	0	0	0	1	0
c	0	1	1	0	0	1	1	0	1	0	1
c	0	1	1	1	0	1	1	1	0	0	1
a	1	0	0	0	1	0	0	0	0	0	0
a	1	0	0	1	1	0	0	0	1	0	0
b	1	0	1	0	0	0	0	1	0	0	0
c	1	0	1	1	0	1	1	0	0	1	1
a	1	1	0	0	1	0	0	0	0	0	0
a	1	1	0	1	1	0	0	0	1	1	0
a	1	1	1	0	1	0	0	1	1	0	0
b	1	1	1	1	0	0	1	0	0	1	0

b) Implementación de funciones:

F_1

	b_1b_0			
	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

a_1a_0

F_0

	b_1b_0			
	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

a_1a_0

S_3

	b_1b_0			
	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	1	0
10	0	0	1	0

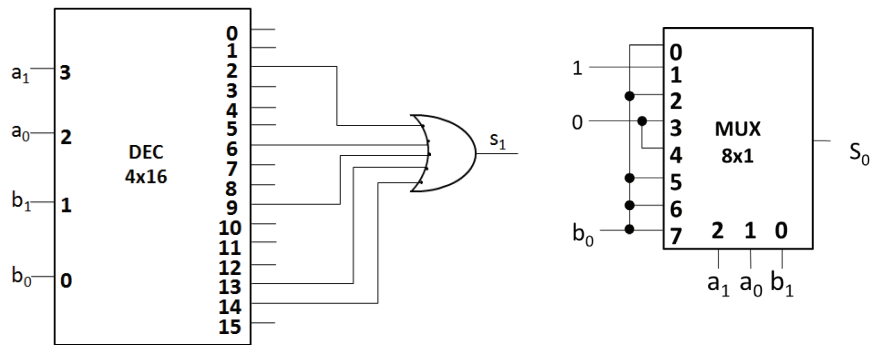
a_1a_0

S_2

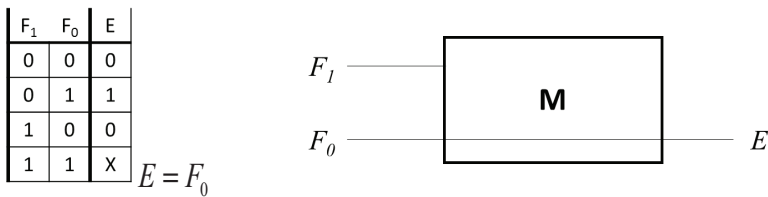
	b_1b_0			
	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	0	1
10	0	0	0	1

a_1a_0

$$F_1 = a_1\overline{b_1} + a_0\overline{b_1}\overline{b_0} + a_1a_0\overline{b_0} = \overline{\overline{a_1}\overline{b_1}\overline{b_0} \cdot \overline{a_1a_0b_0}} \quad ; \quad F_0 = \overline{a_1}b_1 + \overline{a_1}\overline{a_0}b_0 + \overline{a_0}b_1b_0$$
$$S_3 = b_1 \cdot (\overline{a_1} + b_0) = \overline{b_1} + (\overline{a_1} + b_0) \quad ; \quad S_2 = b_1(\overline{a_1} + \overline{b_0}) \cdot (a_1 + b_0)(a_1 + a_0)$$



c) Bloque M:

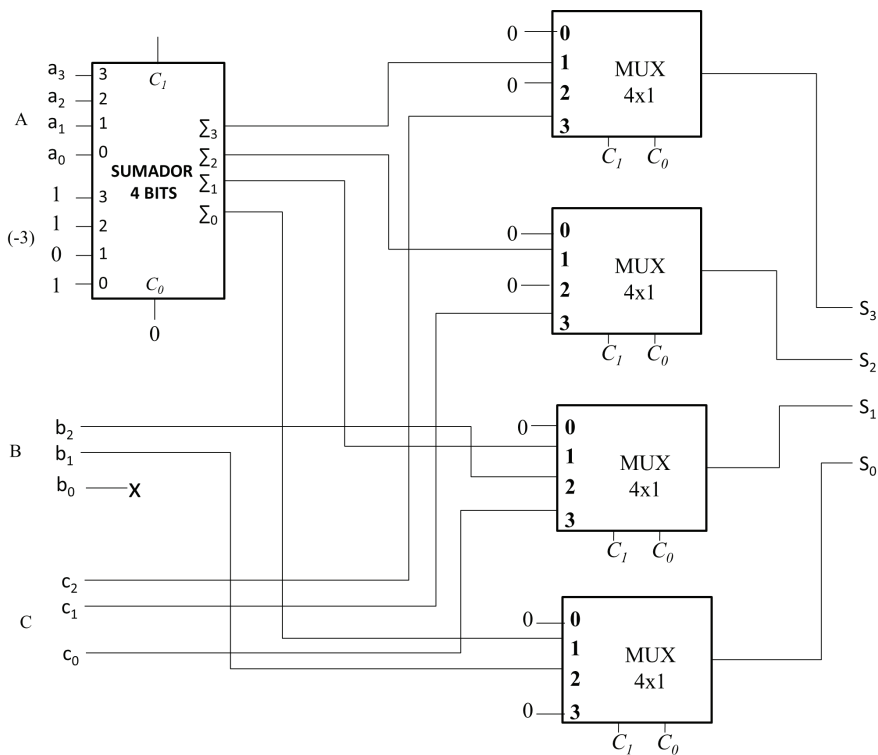


4) Un sistema digital con dos señales de control C_1 y C_0 tiene en su entrada 3 buses A , B y C de 4, 3 y 3 bits, respectivamente. A representa un número expresado en BCD exceso 3, B y C representan un número en binario natural.

La salida del sistema es un bus de 4 bits según la tabla:

$C_1 C_0$	SALIDA
00	Todos Ceros
01	A codificado en binario natural
10	La mitad de B (si es impar, la parte entera de la mitad)
11	El doble de C.

Diseñar el circuito usando sólo multiplexores y un sumador completo de 4 bits, y explicando la solución del diseño.



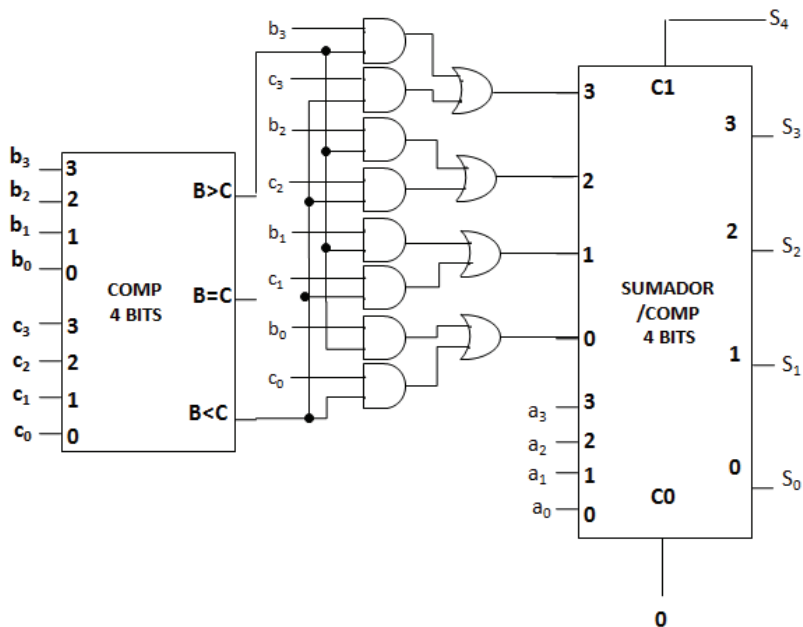
Para pasar A en BCD expreso 3 a binario, se le resta 3, o se suma el complemento a 2 de 3, que es 1101.

Para multiplicar un número binario por 2, se desplaza un bit hacia la izquierda su código binario.

Para dividir un número binario entre 2, se desplaza un bit hacia la derecha su código binario. Si el número fuese impar, se obtendría la parte entera de la mitad.

NOTA: No confundir las líneas de selección de los multiplexores C_1 y C_0 con los bits c_2, c_1 y c_0 del número C .

- 5) Dados tres números binarios de 4 bits, A , B y C , codificados en binario natural, diseñar un circuito que realice la suma de A con el mayor de B y C . Si $B = C$, el resultado debe ser A . Para ello usar sumadores binarios y comparadores y las puertas lógicas que sean necesarias.

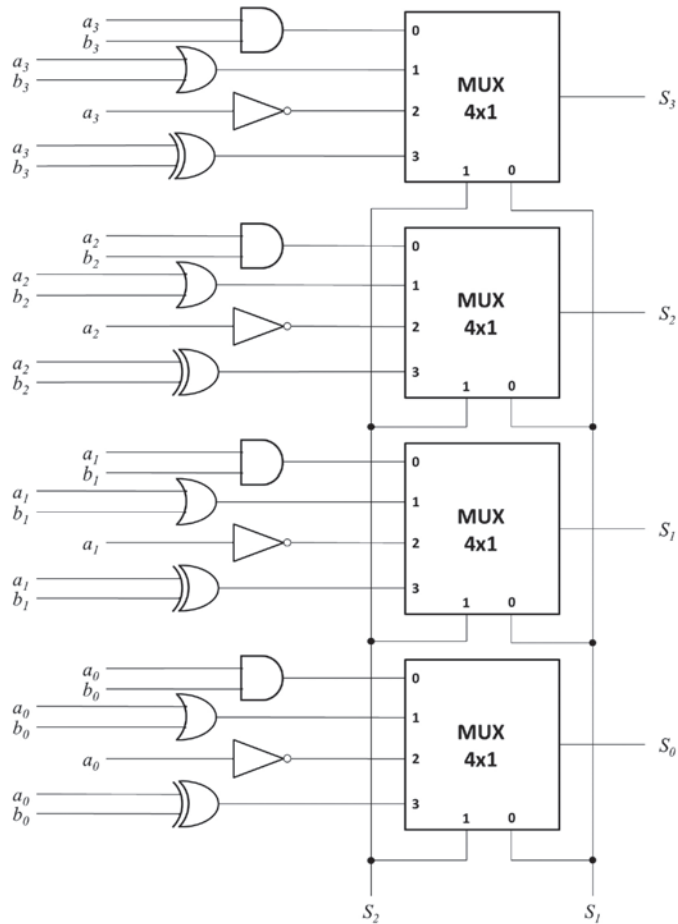


- 6) Diseñar una Unidad Lógica de 4 bits que permita realizar las siguientes operaciones lógicas sobre dos números A y B .

A and B A or B NOT A A xor B

El sistema tendrá, además de los números A (a_3, a_2, a_1, a_0) y B (b_3, b_2, b_1, b_0), dos entradas de selección S_2 y S_1 para definir cuál de las operaciones se realizará. Utilizar como componente básico un multiplexor adecuado y puertas lógicas. Se recomienda diseñar en primer lugar el circuito básico para un bit A_i y B_i .

El diseño de la Unidad Lógica es:

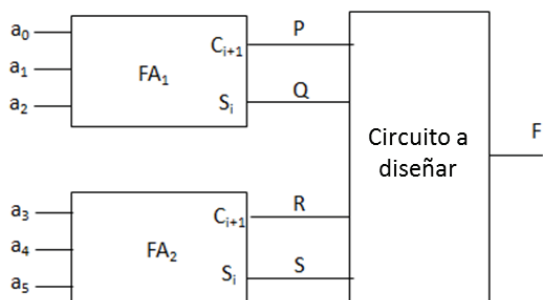


- 7) Se dispone de dos circuitos integrados que contienen, respectivamente, dos sumadores completos de 1 bit y un multiplexor de 8x1. Utilizando éstos e inversores, diseñar un circuito capaz de detectar la presencia de 3 bits a nivel "1" en una palabra de 6 bits en paralelo.

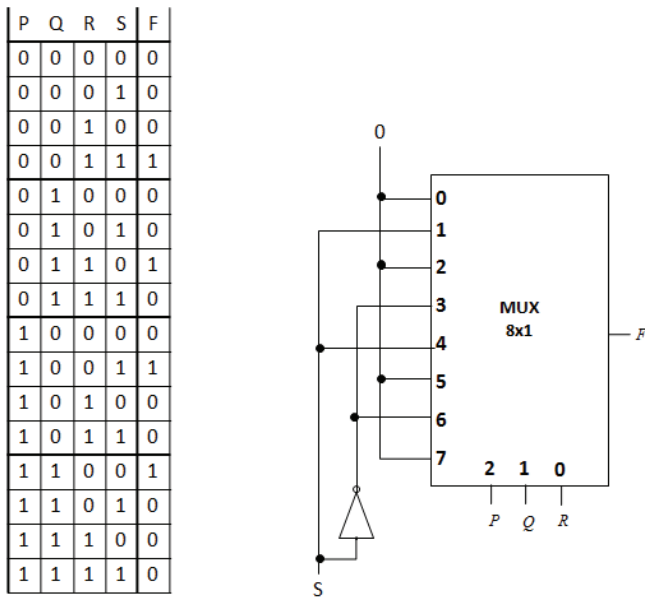
a_i	b_i	c_i	C_{i+1}	S_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Expresa el número de 1's en binario

En el diseño se usa el sumador completo (Full Adder FA) como “contador de unos”.

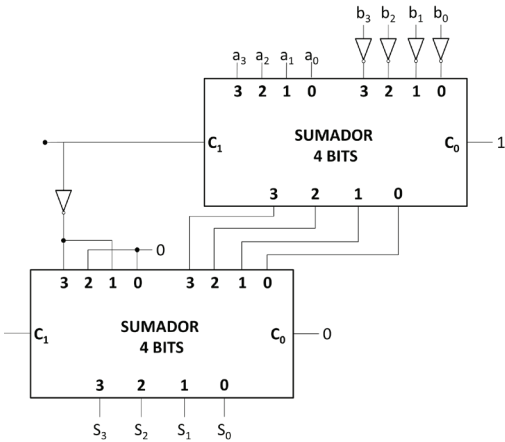


P Q indican cuántos unos hay en a_0 a_1 a_2 mientras que R S hacen lo mismo en a_3 a_4 a_5 .



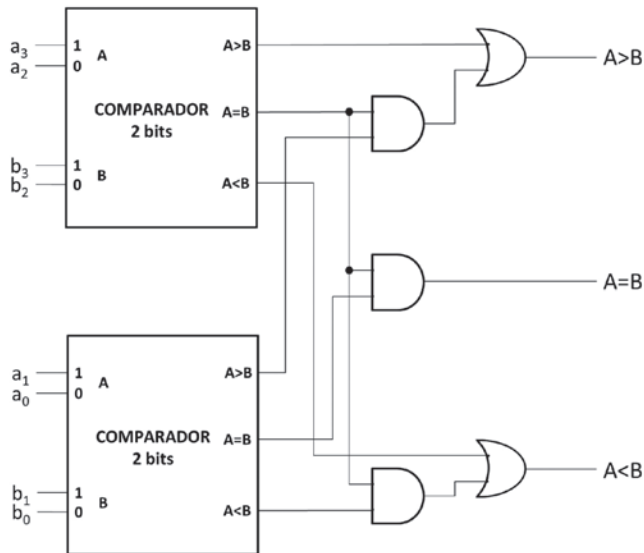
8) Basándose en sumadores binarios de 4 bits en paralelo (circuito integrado 7483), diseñar un restador BCD de un dígito con las entradas y salidas necesarias para poder realizar la resta de dos números BCD de varios dígitos.

El diseño del circuito es:



- 9) Diseñar un circuito que realice la comparación de dos números $A(a_3, a_2, a_1, a_0)$ y $B(b_3, b_2, b_1, b_0)$. Para ello se dispone únicamente de circuitos comparadores de 2 bits sin entradas para conexión en paralelo y puertas lógicas.

El diseño del circuito que realiza la comparación de dos números es:



- 10) Suponer que se quiere diseñar un sistema digital que realice operaciones de tipo lógico sobre parejas de bits de dos palabras que entran de forma serie $a_0, a_1, a_2 \dots$ y $b_0, b_1, b_2 \dots$. El sistema podrá realizar 4 tipos de operaciones sobre los 2 bits que estén a la entrada del circuito, y serán seleccionadas mediante dos entradas de control C_1 y C_0 . Las operaciones posibles son:

$$1^a) \bar{a}_i \quad 2^a) \overline{(a_i + b_i)} \quad 3^a) \bar{a}_i b_i \quad 4^a) \overline{a_i b_i}$$

Implementar este sistema mediante una red de módulos en árbol.

La tabla de verdad es la siguiente:

c_1	c_0	a_i	b_i	F
0	0	0	0	1
0	0	0	1	1
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	0
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

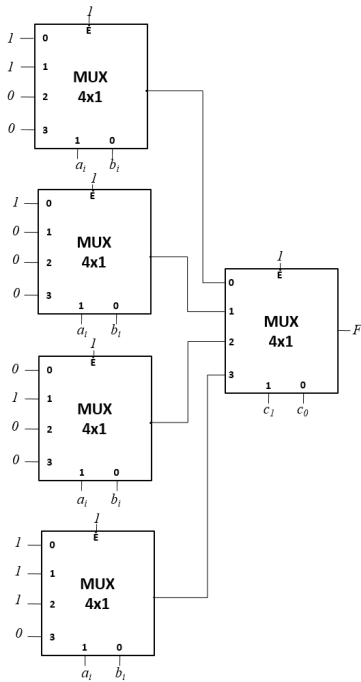
$F = \overline{a_i}$

$F = \overline{a_i + b_i}$

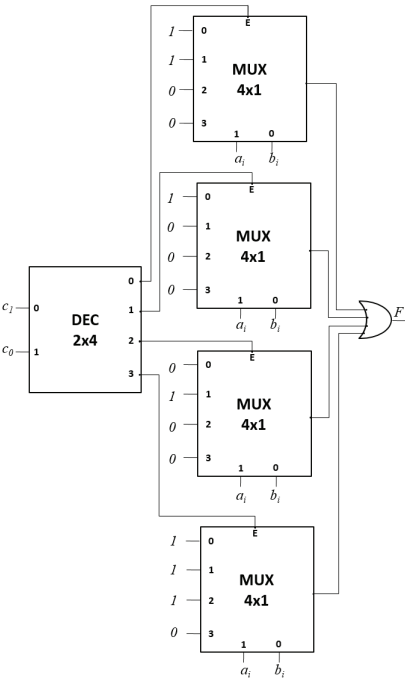
$F = \overline{a_i}b_i$

$F = \overline{a_i b_i}$

Solución 1

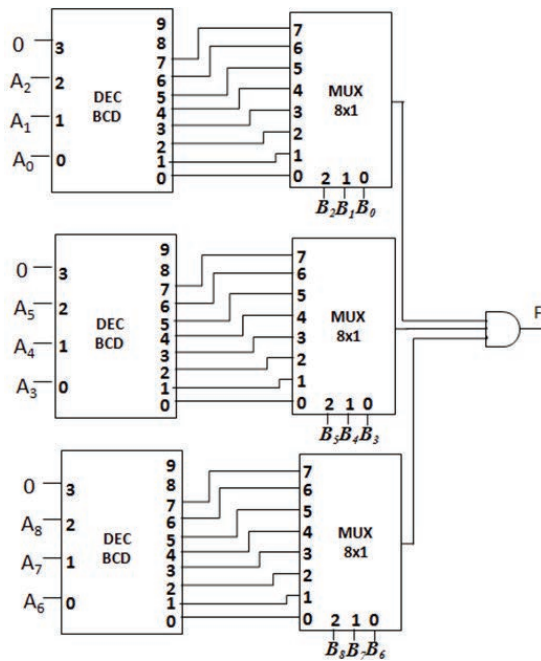


Solución 2

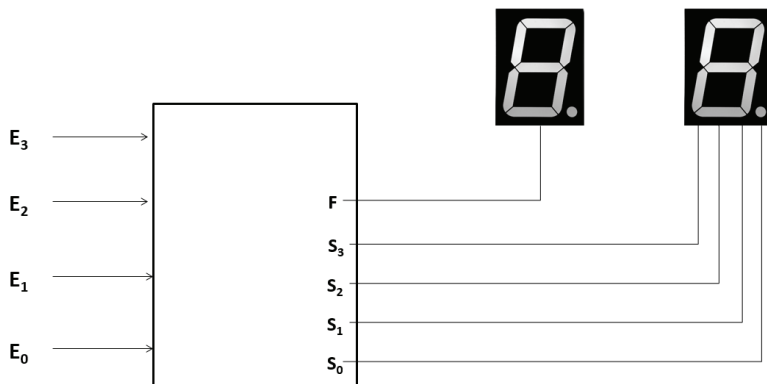


- 11) Se dispone de decodificadores BCD a decimal y de multiplexores de 3 entradas de selección. Diseñar un circuito combinacional capaz de detectar la igualdad o desigualdad de dos números binarios de 9 bits.

El diseño del circuito es:



- 12) Diseñar, usando un comparador, un sumador y los multiplexores que sean necesarios, un circuito que admite como entrada un número en binario natural de 4 bits (E_3, E_2, E_1, E_0) siendo E_3 el bit más significativo, y su salida ataca a los displays de 7 segmentos, uno de ellos, el D , representa las decenas del valor decimal del número, y el otro de ellos, el U representa el valor de las unidades, de acuerdo con la Figura adjunta.



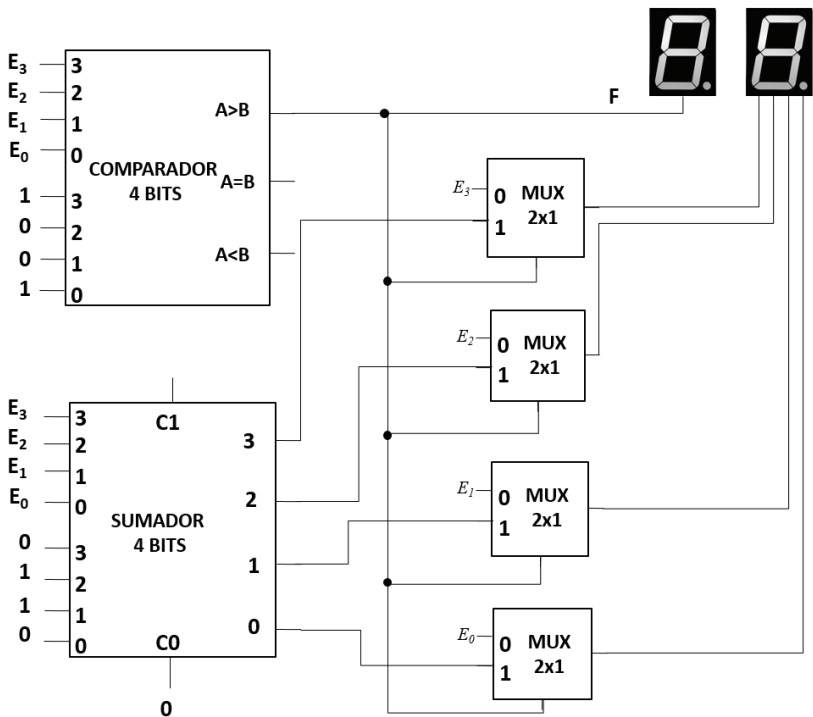
Para atacar a las decenas, sólo se necesita una línea F que activa, cuando se pone a 1, los segmentos de la derecha, quedando apagados todos cuando F es 0. Para atacar al display de las unidades se utilizan 4 líneas que expresan en BCD el valor que debe verse en el display (la codificación de BCD a 7 segmentos está incluida dentro del display y no es objeto de nuestro problema).

Para resolver correctamente este problema es necesario utilizar módulos combinacionales y subsistemas aritméticos y lógicos.

Todo depende de si el número es mayor que 9 o no. Por tanto, se comparará el número con 9 (en binario 1001).

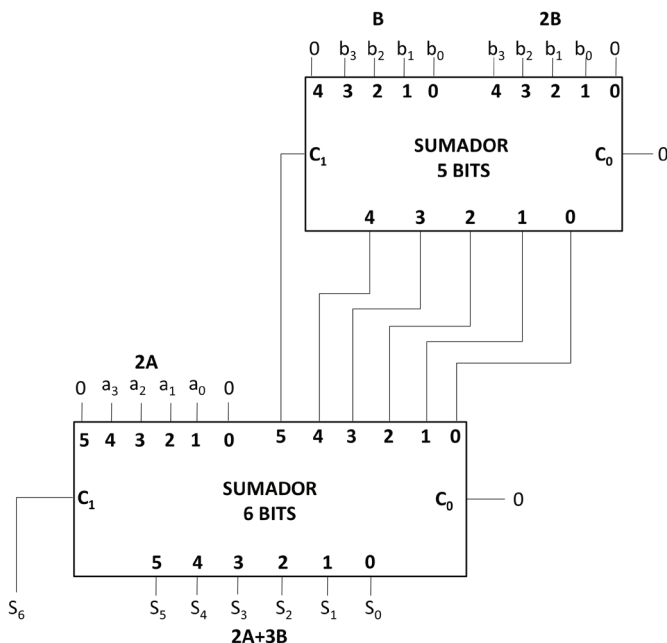
Si el número es mayor que 9, se activa la salida $A > B$ y en ese caso en D debe aparecer $F=1$, cuando el número es mayor que 9 y $F=0$ en caso contrario. Por tanto, F es directamente la salida $A > B$. Además, en este caso debe restarse 10 del número binario, o lo que es lo mismo, sumar 0110, igual que en el sumador BCD, para obtener las unidades.

Si el número es menor que 10, $F=0$ y directamente el número binario debe llegar al display de las unidades. La propia salida $A > B$, que coincide con F , controla los MUX 2x1 que eligen entre las dos opciones.



13) Sean A y B dos números binarios de 4 bits cada uno. Diseñar el circuito aritmético que genere la suma $2A+3B$. Deben utilizarse para ello el menor número posible de módulos aritméticos del número de bits que sea necesario.

Peso	4	3	2	1	0
A:		a3	a2	a1	a0
B:		b3	b2	b1	b0
2A:	a3	a2	a1	a0	0
2B:	b3	b2	b1	b0	0

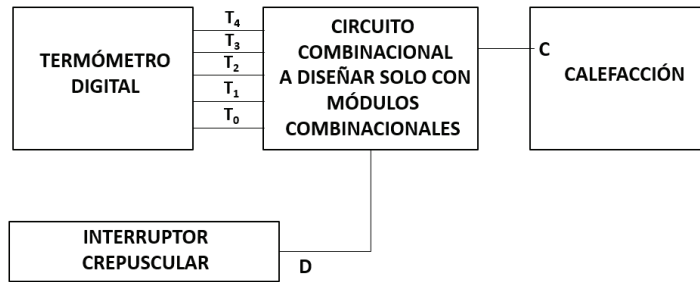


- 14) Diseñar un sistema combinacional utilizando necesariamente al menos uno de los siguientes módulos: comparador y multiplexor. No se puede usar ninguna puerta.

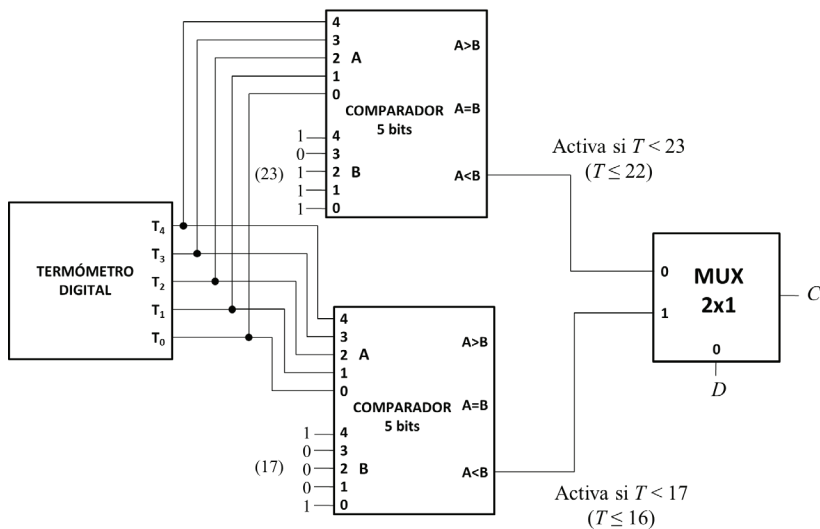
El sistema a diseñar debe activar la calefacción de una oficina cuando la temperatura, medida por un termómetro digital, sea igual o inferior a 22°C durante el día, y cuando la temperatura sea igual o inferior a 16°C durante la noche. La calefacción se activa poniendo a 1 su entrada C . Cuando la temperatura supera los 22°C durante el día o los 16°C durante la noche, el sistema combinacional diseñado manda apagar la calefacción poniendo C a 0.

Se dispone también de un interruptor crepuscular, que proporciona salida $D=0$ durante el día y salida $D=1$ durante la noche. También se disponen de tomas fijas a nivel lógico 1 y a nivel lógico 0 en cantidad suficiente.

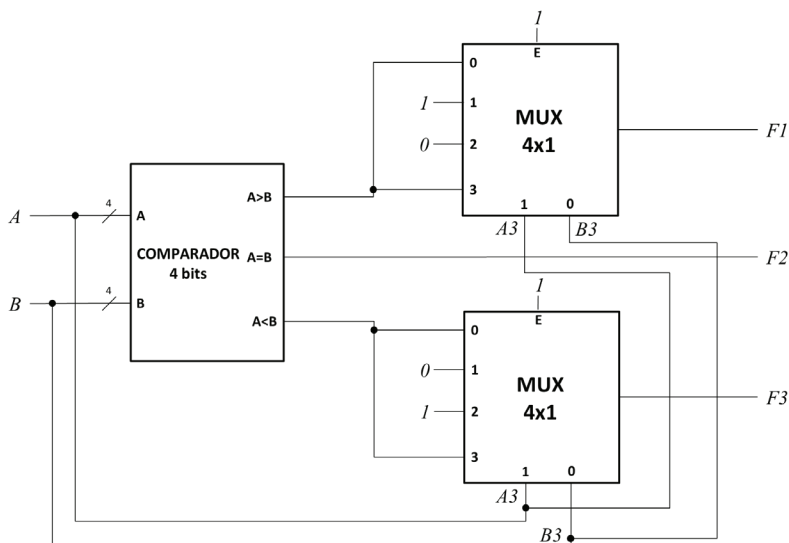
NOTA: La temperatura en $^\circ\text{C}$ del termómetro digital está codificada binario natural, siendo T_4 el bit más significativo. Se supone que siempre es positiva.



El diseño del sistema combinacional es:



- 15) Analiza el funcionamiento del circuito combinacional representado en la figura. Se compone de un comparador binario de números de 4 bits (entradas A y B) y de dos multiplexores 4×1 . Para deducir cuál es el comportamiento de las funciones de salida $F1$, $F2$ y $F3$ indica el valor de las mismas para una secuencia de valores de entrada como sigue: A : 3, -7, -5, 4, -2, 0, -1 y B : 1, -7, -3, 6, -3, 0, 4. Los números en las entradas se encuentran representados en complemento a 2. Observando los valores que toman las funciones de salida $F1$, $F2$ y $F3$, escribe cuál es su significado y el del circuito en general.



Valores de las salidas:

A	B	F1	F2	F3
3 (0011)	1 (0001)	1	0	0
-7 (1001)	-7 (1001)	0	1	0
-5 (1011)	-3 (1101)	0	0	1
4 (0100)	6 (0110)	0	0	1
-2 (1110)	-3 (1101)	1	0	0
0 (0000)	0 (0000)	0	1	0
-1 (1111)	4 (0100)	0	0	1*

* Nótese como en este caso, aunque el valor en binario de -1 representado en complemento a 2 (entrada para el comparador simple) es mayor que la entrada B . La lógica adicional hace que en los multiplexores se seleccionen otras entradas distintas a las que provienen del comparador binario simple (erróneas para el propósito de nuestro circuito). Así seleccionando '01' de ambos MUX, tenemos las salidas mostradas en la tabla. Análisis de las funciones $F1$, $F2$, $F3$. $F1 = 1$ cuando $A > B$ $F2 = 1$ cuando $A = B$ $F3 = 1$ cuando $A < B$.

Análisis del circuito. El circuito se comporta como un comparador de números binarios de 4 bits representados en complemento a 2. No es un simple comparador ya que éste no distinguiría casos como el comentado en el punto anterior. La red de multiplexores a la salida permite, analizar el signo de ambos números y seleccionar la salida correcta

ya que, un número negativo (bit más significativo a 1) sería siempre mayor que un número positivo (bit más significativo a 0).

- 16) Diseñar un semi restador que reciba dos bits de entrada A_i y B_i , y genere un bit con el resultado de la resta y otro con el acarreo para la siguiente etapa (denominado borrow).

A_i	B_i	R_i	BR_{out}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$R_i = A_i \oplus B_i$$

$$BR_{out} = \bar{A}_i \cdot B_i$$

- 17) Diseñar un restador completo cuyas entradas serán los 2 bits a restar (A_i y B_i), y el borrow de entrada (BR_{in}). Las salidas serán los bits de resultado (S_i) y borrow para la siguiente etapa (BR_{out}).

A_i	B_i	BR_{in}	R_i	BR_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$R_i = A_i \oplus B_i \oplus BR_{in}$$

$$BR_{out} = \bar{A}_i BR_{in} + \bar{A}_i B_i + B_i BR_{in}$$

18) Utilizando un comparador, multiplexores 2x1 y un sumador/restador, diseña un sumador signo-magnitud de 4 bits. Incluye un indicador para las condiciones de *overflow*.

Sea $A = a_n \ a_{n-1} \dots a_0$

Sea $B = b_n \ b_{n-1} \dots b_0$

Siendo a_n y b_n los bits de signo de A y B respectivamente.

Casos:

- 1) Si $a_n = 0$ y $b_n = 0$, se suman ($S/R=0$) y se pone 0 como bit de signo.
- 2) Si $a_n = 0$ y $b_n = 1$ y $a_{n-1} \dots a_0 \geq b_{n-1} \dots b_0$, se resta $A-B$ ($S/R=1$) y se pone 0 como bit de signo.
- 3) Si $a_n = 0$ y $b_n = 1$ y $a_{n-1} \dots a_0 < b_{n-1} \dots b_0$, se resta $B-A$ ($S/R=1$, $IP=1$) y se pone 1 como bit de signo.
- 4) Si $a_n = 1$ y $b_n = 0$ y $a_{n-1} \dots a_0 \geq b_{n-1} \dots b_0$, se resta $A-B$ ($S/R=1$) y se pone 1 como bit de signo.
- 5) Si $a_n = 1$ y $b_n = 0$ y $a_{n-1} \dots a_0 < b_{n-1} \dots b_0$, se resta $B-A$ ($S/R=1$, $IP=1$) y se pone 0 como bit de signo.
- 6) Si $a_n = 1$ y $b_n = 1$, se suman ($S/R=0$) y se pone 1 como bit de signo.

Casos	a _n	b _n	M	m	S/R	IP	BS	Ejemplo
X	0	0	0	0	X	X	X	-
1	0	0	0	1	0	0	0	+2+3=+5
1	0	0	1	0	0	0	0	+3+1=+4
X	0	0	1	1	X	X	X	-
X	0	1	0	0	X	X	X	-
3	0	1	0	1	1	1	1	+2-5=-3
2	0	1	1	0	1	0	0	+5-2=+3
X	0	1	1	1	X	X	X	-
X	1	0	0	0	X	X	X	-
5	1	0	0	1	1	1	0	-4+6=+2
4	1	0	1	0	1	0	1	-6+4=-2
X	1	0	1	1	X	X	X	-
X	1	1	0	0	X	X	X	-
6	1	1	0	1	0	0	1	-2-3=-5
6	1	1	1	0	0	0	1	-3-1=-4
X	1	1	1	1	X	X	X	-

		M m			
	S/R	00	01	11	10
a _n b _n	00	X	0	X	0
	01	X	1	X	1
	11	X	0	X	0
	10	X	1	X	1

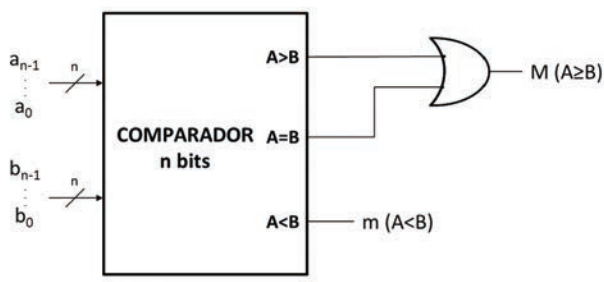
$S/R = a_n \oplus b_n$

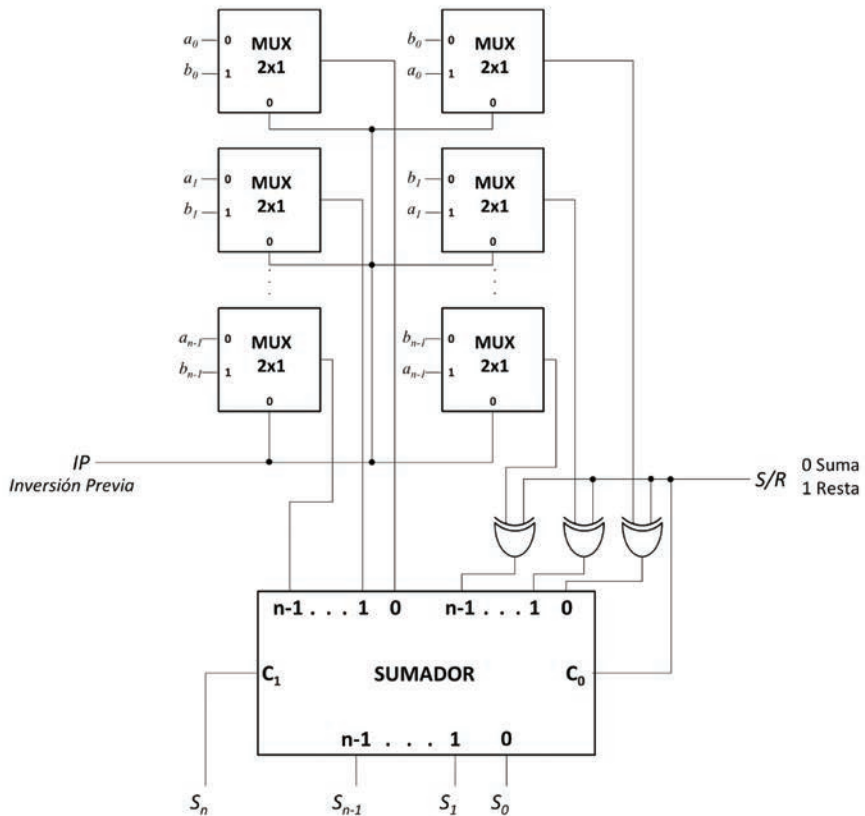
		M m			
	IP	00	01	11	10
a _n b _n	00	X	0	X	0
	01	X	1	X	0
	11	X	0	X	0
	10	X	1	X	0

$IP = \bar{a}_n b_n m + a_n \bar{b}_n m = m(a_n \oplus b_n)$

BS	M m			
	00	01	11	10
00	X	0	X	0
01	X	1	X	0
11	X	1	X	1
10	X	0	X	1

$BS = b_n m + a_n \bar{m}$





19) Se desea diseñar un circuito combinacional que admita dos números binarios A y B de 4 bits cada uno y cuya salida sea:

Si $A > B$, el complemento a 1 de A .

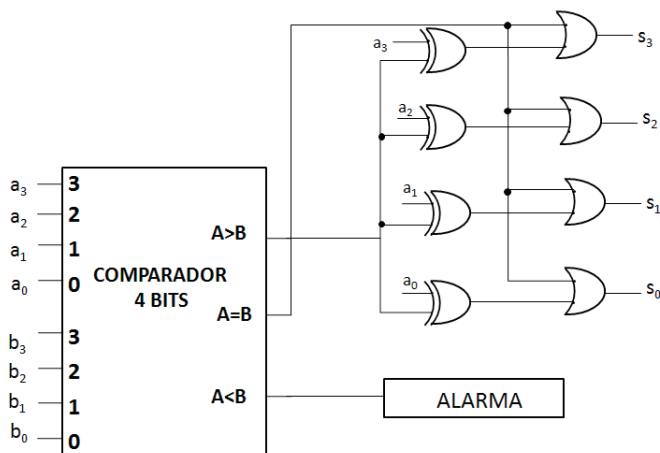
Si $A = B$, el valor binario 1111.

Si $A < B$, en número binario A . En este último caso debe activarse además una alarma.

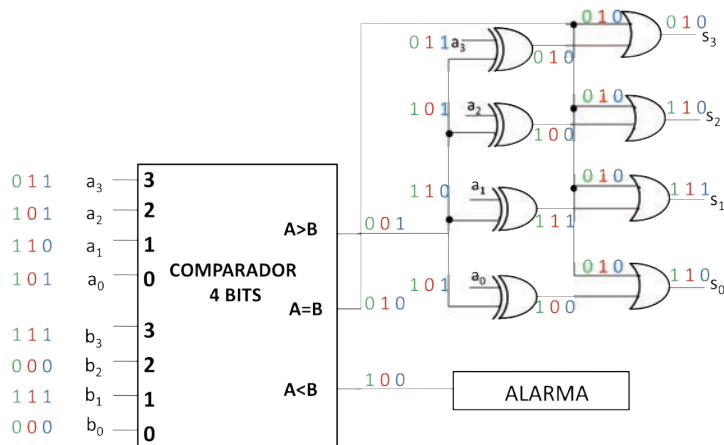
Usa para ello los circuitos combinacionales y puertas que consideres necesarias.

Una vez diseñado particulariza el diseño para $B=1010$ y dale a A los valores 0111, 1010, 1101, comprobando que el circuito funciona correctamente en los tres casos.

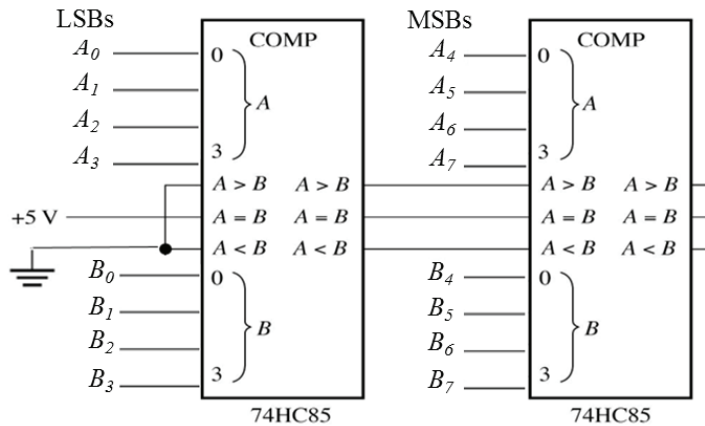
Diseño del circuito combinacional:



Particularización para $B=1010$ y A con los valores 0111, 1010, 1101.



- 20) Se pretende realizar un comparador de 8 bits. Para ello, se dispone de 2 módulos 7485 (comparadores de 4 bits). El circuito comparador tendrá las entradas de los dos números de 8 bits ($A7...A0$ y $B7...B0$) y como salidas: si A es menor que B , si son iguales o si A es mayor que B .



21) Sean B (B_2, B_1, B_0) y C (C_2, C_1, C_0) números de tres bits cada uno, que representan cantidades codificadas en binario. Diseñar un circuito combinacional que proporcione tres salidas tales que:

$F_1=1$ cuando $B \neq C$

$F_2=1$ cuando $B = C \neq 0$

$F_3=1$ cuando $B = C = 0$

Se ha de utilizar única y exclusivamente un comparador de palabras de 4 bits y el mínimo número de puertas que necesite. Realizar previamente una especificación de alto nivel para deducir las funciones de salida. Mostrar, por último, el circuito con todas sus conexiones.

(El comparador tendrá tres salidas m =menos, i =igual, M =mayor).

- Especificación y Funciones de Salida

Como utilizaremos un comparador la especificación se basará en términos m , i , M de las variables de entrada.

$F_1=1$ cuando $B \neq C \leftrightarrow F=1$ cuando $B > C$ o $B < C$.

Luego $F=1$ cuando $M=1$ ó $m=1$.

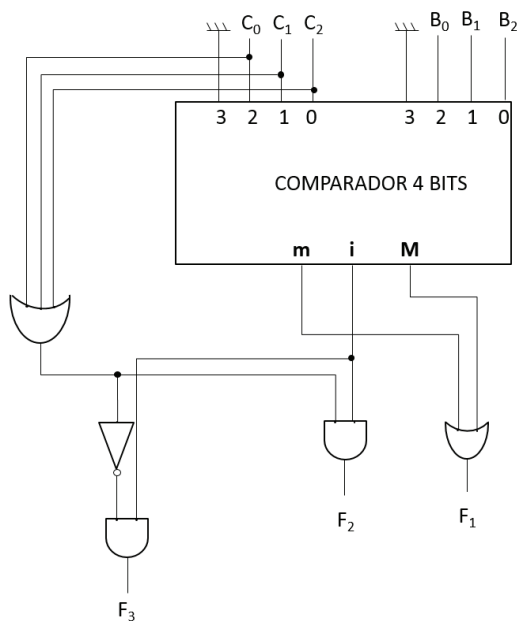
Luego $F_1 = m + M$. También podría ser $F=1$, cuando $i=0$, $F = \bar{i}$.

$F_2=1$ cuando $B = C \neq 0 \leftrightarrow F_2=1$ cuando $i=1$ y $C \neq 0$.

Luego $F_2=1$ cuando $I=1$ y $C_0+C_1+C_2=1$.

Luego $F_2 = I(C_0 + C_1 + C_2)$

$F_3=1$ cuando $B = C = 0 \leftrightarrow F_3 = \overline{i(C_0 + C_1 + C_2)}$.



22) Diseñar un circuito combinacional que acepte como entradas dos números binarios de dos bits D y d , y cuyas salidas sean el Cociente C y el Resto R resultantes de dividir D (dividendo) entre d (divisor). Suponer las siguientes condiciones:

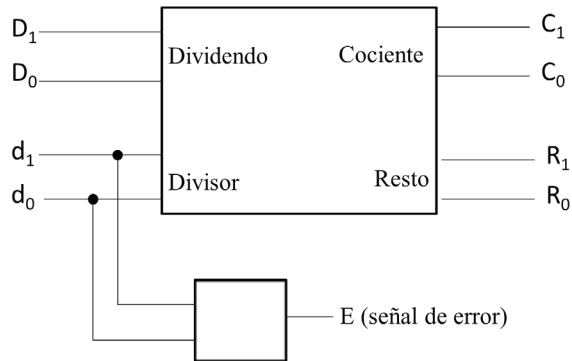
$$0 \leq D < 4$$

$$0 < d < 4$$

siendo D y d números naturales. Si $d=0$, se activa una señal de error E , y las salidas Cociente y Resto se ponen a cero.

- Realizar la tabla de verdad que especifica el problema.
- Diseñar el circuito de la señal de error con el mínimo número de puertas.
- Implementar la salida Cociente con MUX 8x1, y la salida Resto con Decodificadores y puertas OR de 2 entradas.

a) La tabla de verdad es:



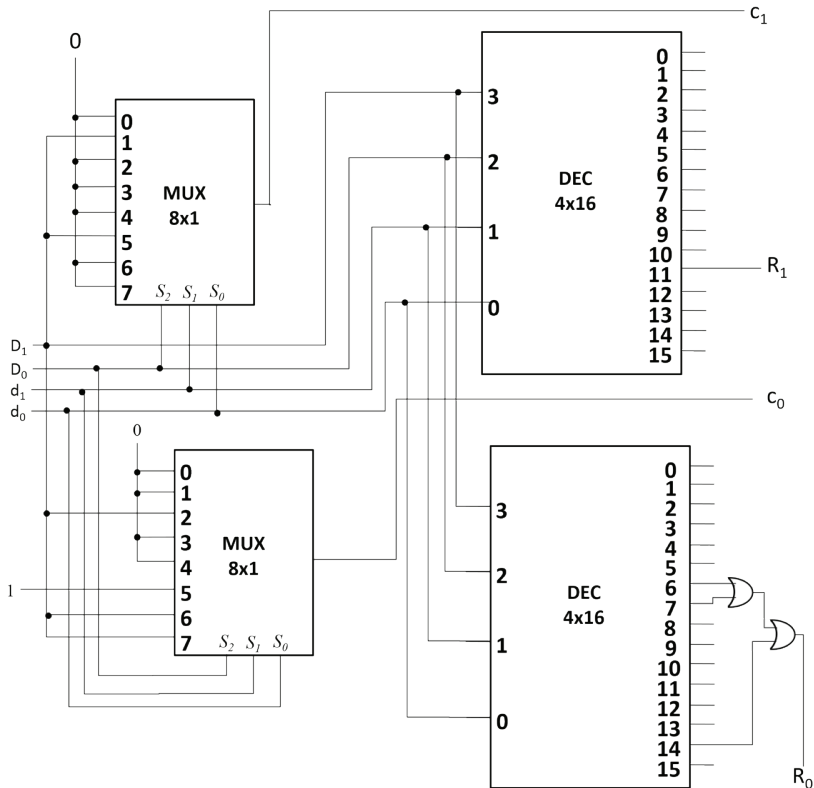
D_1	D_0	d_1	d_0	C_1	C_0	R_1	R_0	E
0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0
0	1	0	0	0	0	0	0	1
0	1	0	1	0	1	0	0	0
0	1	1	0	0	0	0	1	0
0	1	1	1	0	0	0	1	0
1	0	0	0	0	0	0	0	1
1	0	0	1	1	0	0	0	0
1	0	1	0	0	1	0	0	0
1	0	1	1	0	0	1	0	0
1	1	0	0	0	0	0	0	1
1	1	0	1	1	1	0	0	0
1	1	1	0	0	1	0	1	0
1	1	1	1	0	1	0	0	0

b) El circuito de la señal de error es:

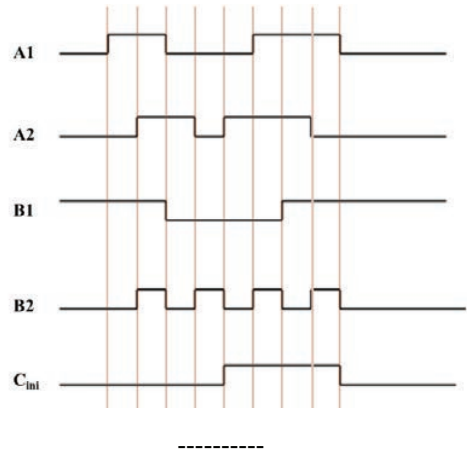


$$E = \overline{d_1 + d_0}$$

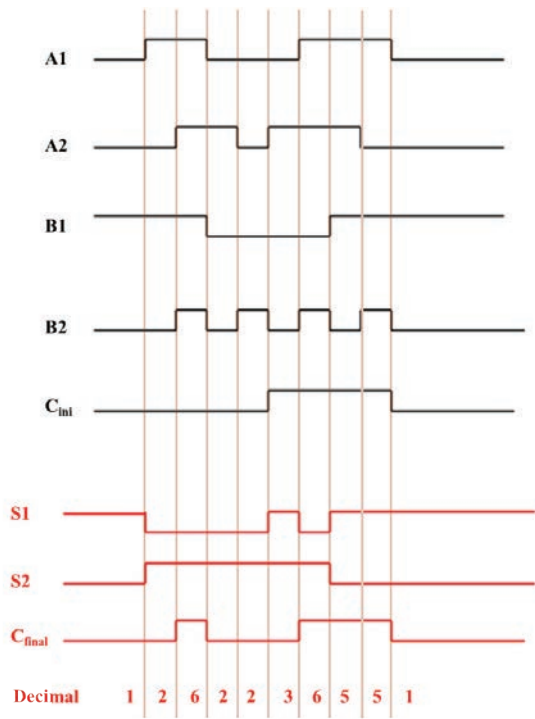
c) La implementación del cociente (C_1C_0) y el resto (R_1R_0) es la siguiente:



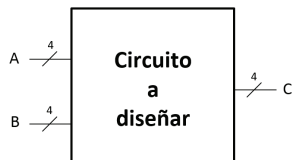
- 23) Las formas de onda de la figura se aplican a la entrada de un sumador completo de dos bits en el que se suman el número A , compuesto por los bits $A1$ (bit menos significativo) y bit $A2$, y el número B , compuesto por los bits $B1$ (bit menos significativo) y $B2$. C_{ini} es el acarreo de entrada, mientras que C_{final} es el acarreo de salida. Completar el cronograma, despreciando los retardos. Suponer que $S1$ es el bit menos significativo de la suma.



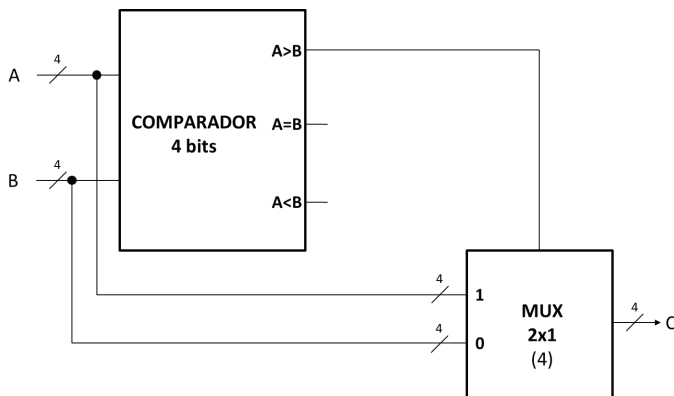
El cronograma completo es:



- 24) Diseñe un circuito que seleccione el menor de dos números. De acuerdo con la figura, la salida C debe ser el menor de A y B , o cualquiera de ellos en el caso de que sean iguales. Utilice para ello módulos combinacionales (multiplexores/demultiplexores, codificadores/decodificadores, comparadores, sumadores...).



El diseño del circuito es:

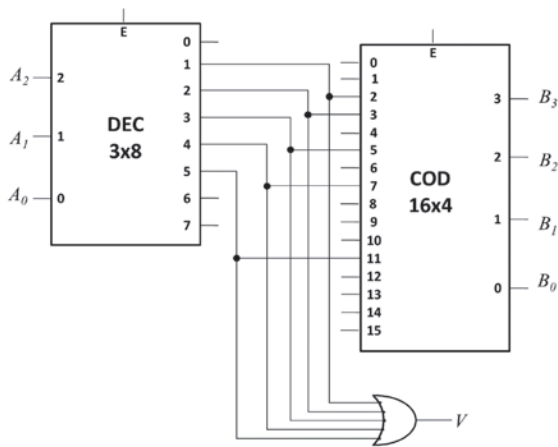


- 25) Diseña un circuito combinacional que permita realizar la conversión entre el código A (con valores 1, 2, 3, 4 y 5) y el código B (2, 3, 5, 7, 11). El circuito debe disponer además de una señal adicional de salida V que indique si el código obtenido es válido o no. Será válido en el caso de que la entrada tome cualquiera de los valores del código A .

La tabla de verdad de conversión de código es:

Código A				Código B			
A2	A1	A0	V	B3	B2	B1	B0
0	0	0	0	X	X	X	X
0	0	1	1	0	0	1	0
0	1	0	1	0	0	1	1
0	1	1	1	0	1	0	1
1	0	0	1	0	1	1	1
1	0	1	1	1	0	1	1
1	1	0	0	X	X	X	X
1	1	1	0	X	X	X	X

El circuito conversor de código se implementa con un DEC 3x8 y un COD 16x4 según la tabla anterior.



		A ₁ A ₀			
V		00	01	11	10
	A ₂	0	1	1	1
	1	1	1	0	0

$V = A_2 \cdot \overline{A_1} + \overline{A_2} \cdot A_0 + \overline{A_2} \cdot A_1$, o bien $V = \overline{B_3} \cdot B_2 \cdot B_0 + \overline{B_3} \cdot \overline{B_2} \cdot B_1 + \overline{B_2} \cdot B_1 \cdot B_0$, o bien poniendo una puerta OR cuyas entradas sean todas y cada una de las líneas que comunican ambos módulos.

- 26) Diseñar un circuito que sume dos números de dos bits cada uno, expresados en complemento a 2, utilizando únicamente puertas NAND y MUX 4x1. (Indicar las funciones de salida y tabla de verdad del circuito).

A1	A0	B1	B0	S2	S1	S0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	1	1	0
0	0	1	1	1	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	1	1
0	1	1	1	0	0	0
1	0	0	0	1	1	0
1	0	0	1	1	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	1	1	1
1	1	0	1	0	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

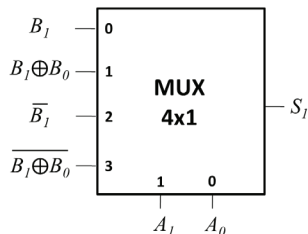
Los operandos $A(A_1, A_0)$ y $B(B_1, B_0)$ están en complemento a 2, luego $00=0_{(2)}$, $01=1_{(2)}$, $10=-2_{(2)}$, $11=-1_{(2)}$.

Utilizando el mapa de Karnaugh se calculan S_2 y S_0 .

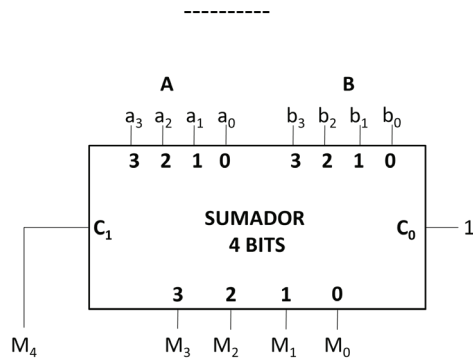
$$S_2 = \overline{(A_1 + B_1)} + \overline{(A_1 + \overline{A_0} + B_0)} + \overline{(\overline{A_0} + B_1 + \overline{B_0})}$$

$$S_0 = \overline{(\overline{A_0} + B_0)} + \overline{(\overline{A_0} + B_0)}$$

La implementación de S_1 con un único MUX 4x1 es:



- 27) Con un sumador de 4 bits, diseñe un circuito que permita obtener la media aritmética de dos números (codificados con 4 bits en binario natural) redondeada por defecto.



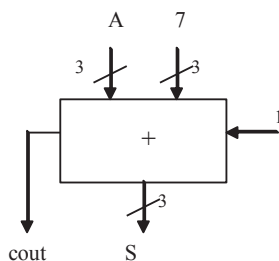
Siendo $M_4 \ M_3 \ M_2 \ M_1 \ M_0$ las salidas del sumador de 4 bits. La media de los dos números es el número expresado por el código $M_4 \ M_3 \ M_2 \ M_1$. Esto es debido a que si en el resultado de la suma $S = M_4 \ M_3 \ M_2 \ M_1 \ M_0$ desplazamos a la derecha una posición, obtenemos la parte entera de su mitad $\text{int}(S/2) = M_4 \ M_3 \ M_2 \ M_1$.

TEST Y PREGUNTAS CORTAS

1) ¿Qué afirmación es cierta?

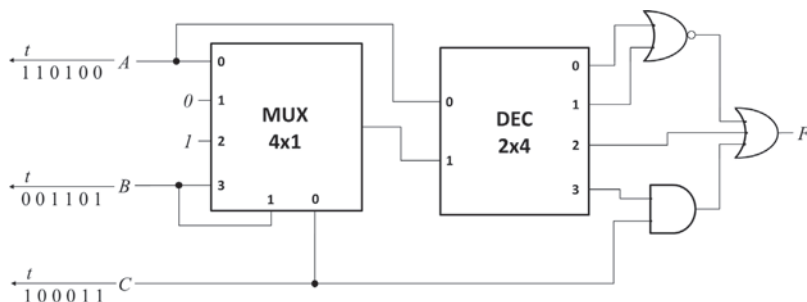
- ☐ A Para implementar un sumador/restador de 4 bits se usa un sumador de 4 bits en el que uno de los operandos está en complemento a 2.
- ☐ B Un generador de paridad par genera siempre un 1 y lo añade a la información antes de ser mandada por un bus.
- ☐ C Un sumador completo tiene dos entradas y dos salidas.

2) ¿Cuál es el valor de salida del siguiente circuito?



- ☐ A A
- ☐ B $A + 8$
- ☐ C $A - 1$

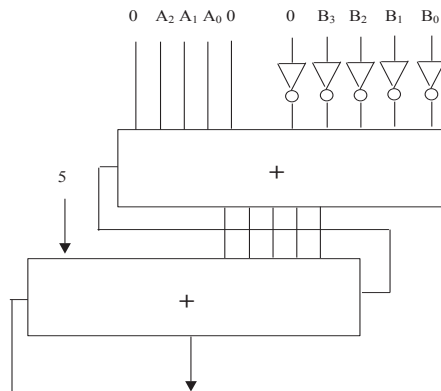
- 3) Dado el circuito de la Figura adjunta, por la entrada A se introduce el número 52 expresado en binario con 6 bits, entrando primero el bit menos significativo. Por la entrada B se introduce el número 44, expresado en binario con 6 bits, de forma sincronizada con el número A , y entrando en primer lugar el bit más significativo. La entrada C , también sincronizada con A y B , es el número 35 comenzando por el bit menos significativo.



Si la salida sincronizada por F se considera un número binario en donde el bit menos significativo sale en primer lugar, ¿qué número en decimal representaría dicha salida?

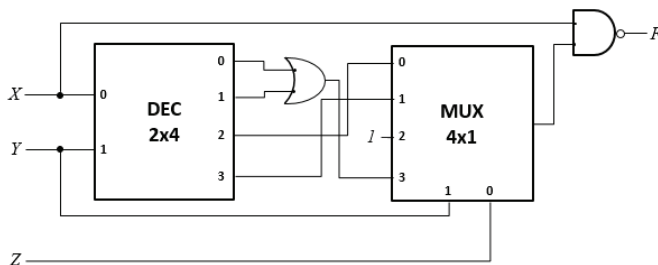
- ☐ A El número 46
☐ B El número 29.
☐ C El número 37.

4) ¿Cuál de las siguientes expresiones corresponde a la expresión de la salida aritmética del circuito de la figura?



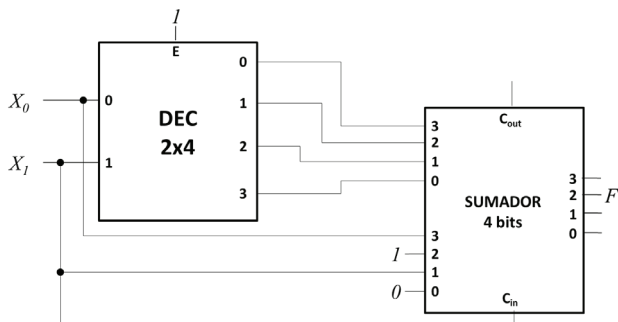
- ☐ A $(2A - B) + 5$
☐ B $2(A - B) + 5$
☐ C $2A - B + 6$

5) Analiza el siguiente circuito y di cuál es la expresión booleana de salida.



- ☐ A $F = 1$
☐ B $F = x \cdot y \cdot \bar{z}$
☐ C $F = \bar{x} + \bar{y} + z$

- 6) Dado el circuito de la figura, ¿cuál es relación entre la función booleana de salida F y las variables de entrada x_1 y x_0 ?

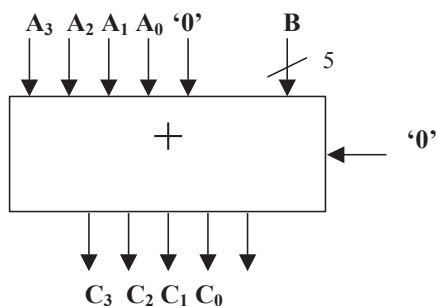


- ☐ A $x_0 \cdot x_1$
☐ B $\overline{x_0 \cdot x_1}$
☐ C $\overline{x_0} \cdot \overline{x_1}$

- 7) Dado el sumador de números binarios de la figura, si se realizan 2 sumas, con los valores que se indican a continuación, ¿cuáles serán los resultados de la suma siendo $C=C_3C_2C_1C_0$?

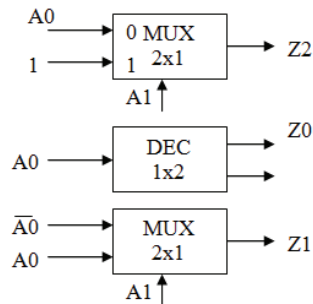
Suma 1: $A = 0101$, $B = 10100$

Suma 2: $A = 1001$, $B = 10100$



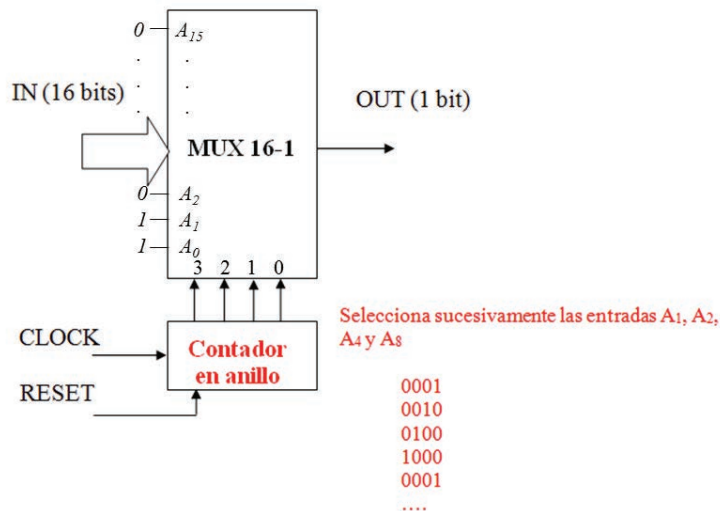
- ☐ A $C = 1110$ y $C = 0110$
☐ B $C = 1111$ y overflow
☐ C $C = 1111$ y $C = 0011$

8) Indique la operación que realiza el siguiente circuito:



- ☐ A $Z = A$.
☐ B $Z = A + 3$.
☐ C Ninguna de las anteriores.

9) El circuito de la figura responde a un conversor paralelo serie que en cada ciclo de reloj pone en la salida un bit de los de la palabra de entrada. Para la palabra de entrada “0110011101010011” ($A_{15}, A_{14}, \dots, A_1, A_0$), ¿cuál sería la secuencia de salida? Suponga que el contador comienza la cuenta en el número “0001” (C_3, C_2, C_1, C_0) y que el sentido de circulación del contador en anillo es de derecha a izquierda.



- ☐ A 1011101110111011
☐ B 1001010111001101
☐ C 1100110011001100

6

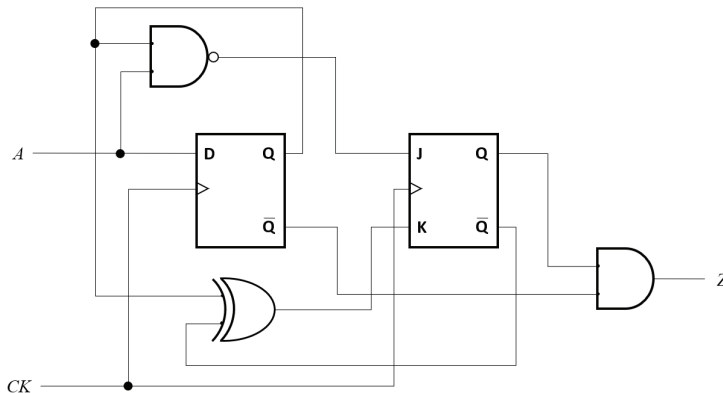
SISTEMAS SECUENCIALES: BIESTABLES

CONTENIDOS:

- Especificación de sistemas secuenciales: Diagrama de estados.
- Biestables. Análisis, Funcionamiento.
- Análisis de sistemas secuenciales síncronos. Autómatas.
- Diseño de SSS. Autómatas.
- Sistemas secuenciales básicos.

EJERCICIOS RESUELTOS

1) Dado el circuito de la figura adjunta:



Calcular:

- Ecuaciones de excitación de los biestables y de la salida Z .
- Tabla de verdad completa a partir de la entrada y las variables de estado, indicando próximos estados.
- Diagrama de estados.
- A la vista del diagrama de estados, ¿tiene este circuito alguna singularidad digna de destacar?
- Evolución temporal de las excitaciones de los biestables y las salidas de los mismos si inicialmente ($t=0$) $A=0$, $Q_1=0$ y $Q_0=0$ y la evolución temporal de la entrada A es 1110.

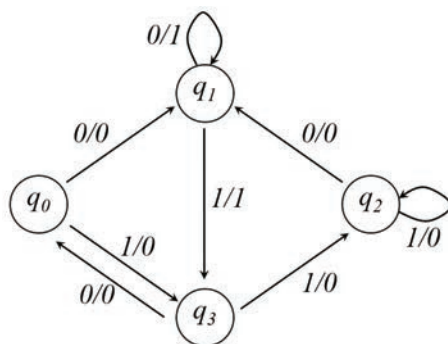
Ecuaciones de entrada a biestables y salida:

$$D_1 = A \quad J_0 = \overline{A \cdot Q_1} \quad K_0 = Q_1 \oplus \overline{Q_0} \quad Z = \overline{Q_1} Q_0$$

Tabla de transición de estados y salida:

A	$Q_1(t)$	$Q_0(t)$	D_1	J_0	K_0	$Q_1(t+1)$	$Q_0(t+1)$	z
0	0	0	0	1	1	0	1	0
0	0	1	0	1	0	0	1	1
0	1	0	0	1	0	0	1	0
0	1	1	0	1	1	0	0	0
1	0	0	1	1	1	1	1	0
1	0	1	1	1	0	1	1	1
1	1	0	1	0	0	1	0	0
1	1	1	1	0	1	1	0	0

Diagrama de estados:



Codificación de estados:

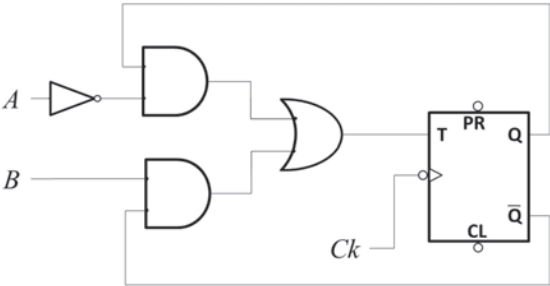
	Q_1	Q_0
q_0	0	0
q_1	0	1
q_2	1	0
q_3	1	1

El diagrama de estados no presenta ningún comportamiento atípico digno de resaltar. Con las entradas adecuadas se puede alcanzar todos los estados.

La evolución temporal es la siguiente:

	t=0	t=1	t=2	t=3	t=4	t=5
Q ₁	0	0	1	1	1	0
Q ₀	0	1	1	0	0	1
A	0	1	1	1	0	
D ₁	0	1	1	1	0	
J ₀	1	1	0	0	1	
K ₀	1	0	1	0	0	

2) El circuito de la Figura es un biestable genérico AB realizado a partir de un biestable T. Indicar cuál es su comportamiento ante las 4 posibles entradas. ¿Qué modificaciones habría que hacerle para que se comporte como un biestable JK?



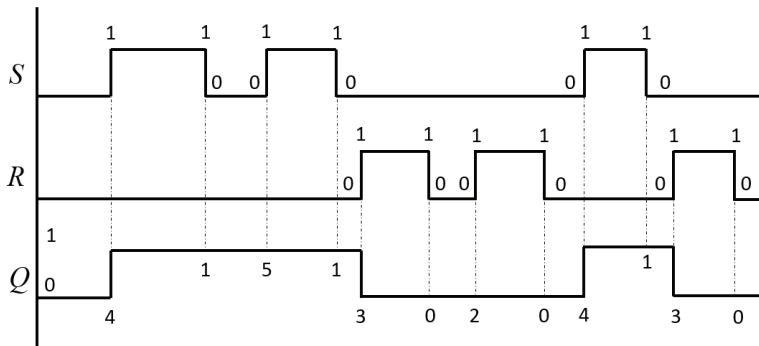
$$T = \overline{A}Q + B\overline{Q}$$

A	B	Q(t)	T	Q(t+1)	Significado
0	0	0	0	0	Fuerza a 0 desde cualquier estado
0	0	1	1	0	
0	1	0	1	1	Cambia al otro estado
0	1	1	1	0	
1	0	0	0	0	Mantiene el estado actual
1	0	1	0	1	
1	1	0	1	1	Fuerza a 1 desde cualquier estado
1	1	1	0	1	

Modificación para que se comporte como un biestable JK.

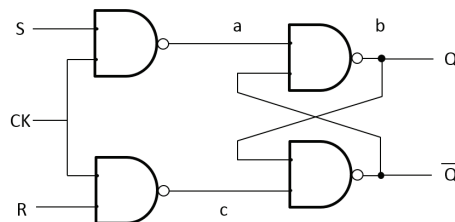
S	R	Q(t)	Q(t+1)	
0	0	0	0	0
0	0	1	1	1
0	1	0	0	2
0	1	1	0	3
1	0	0	1	4
1	0	1	1	5
1	1	0	-	6
1	1	1	-	7

En la figura aparecen numerados los casos en que existe cambio de algunas de las entradas S o R . El número que se indica, corresponde al número de la transición de estado que se produce.

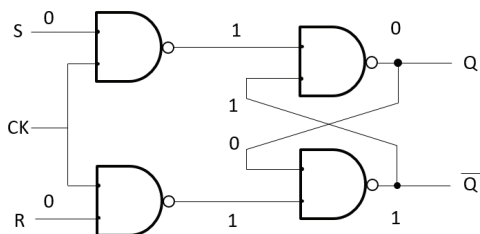


4) Un biestable SR construido a partir de puertas NAND se muestra en la figura. Determinar los niveles lógicos en los puntos a , b , y c bajo las siguientes condiciones:

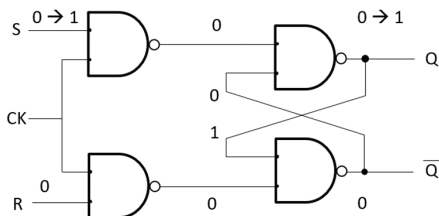
- $S=0, R=0, Q=0$.
- Como en a) pero S cambia de 0 a 1.
- $S=0, R=0, Q=1, R$ cambia de 0 a 1.



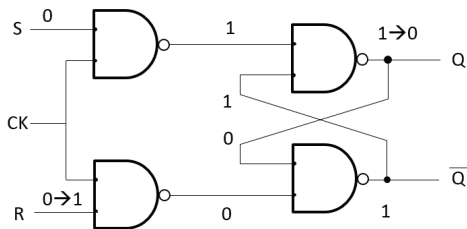
- a) $S=0$, $R=0$ y $Q=0$. Recordamos que la salida de una puerta NAND solo es 0 si ambas entradas son 1. En este primer caso hay una situación estable, con $a=1$, $b=0$ y $c=1$.



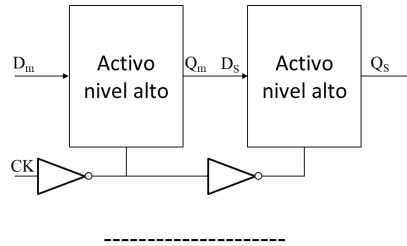
- b) Si en esta situación S cambia de 0 a 1, se produce un cambio a la salida del flip-flop, pasando Q de 0 a 1 en cuanto $CK=1$. Se produce un cambio de estado, $a=0$, $b=1$ y $c=1$.



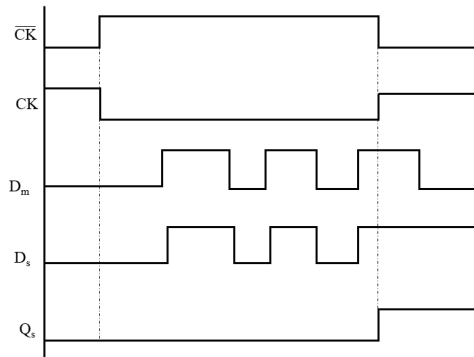
- c) $S=0$, $R=0$, $Q=1$ y R cambia de 0 a 1. Se produce un cambio de estado en cuanto $CK=1$ de 1 a 0. $a=1$, $b=0$ y $c=0$.



- 5) Un flip-flop disparado por flanco combina dos cerrojos, tal como se muestra en la figura. Con la ayuda de un diagrama de tiempos mostrar que el flip-flop detecta los datos de entrada presentes en el flanco positivo de reloj y genera una salida que corresponde a éstos.



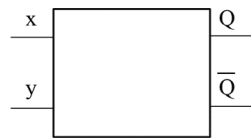
El flip-flop de tipo D tiene por ecuación de transición $Q(t+1) = D(t)$. Por tanto, si CK está a 0, sigue la evolución de la entrada D_m ($Q_m = D_s$), y si CK está en 1, Q_s sigue la evolución de D_s .



6) Dibujar el diagrama de estados del biestable cuya ecuación característica es:

a) $Q(t+1) = (x \oplus y \oplus Q)$

b) $Q(t+1) = (x \quad y \quad Q)$



En primer lugar, y para no duplicar trabajo innecesario, se demuestra que $(x \oplus y \oplus Q) = (x \quad y \quad Q)$ mediante sus tablas de verdad.

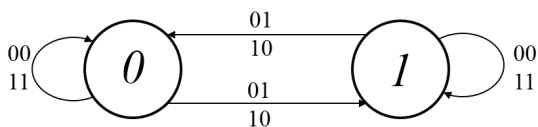
x	y	Q(t)	$x \oplus y$	$x \oplus y \oplus Q$	$x \odot y$	$x \odot y \odot Q$
0	0	0	0	0	1	0
0	0	1	0	1	1	1
0	1	0	1	1	0	1
0	1	1	1	0	0	0
1	0	0	1	1	0	1
1	0	1	1	0	0	0
1	1	0	0	0	1	0
1	1	1	0	1	1	1

Podemos poner esta tabla de otra forma, como tabla de estado de dos dimensiones.

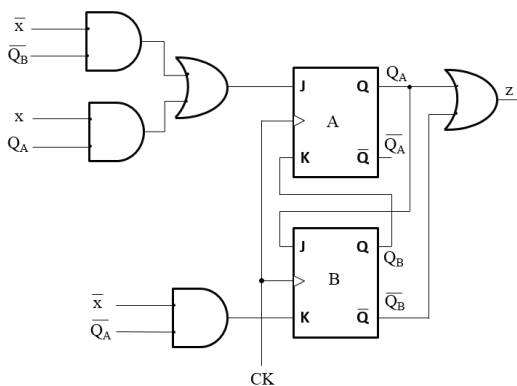
Estado actual Q(t)	Entradas (x,y)			
	00	01	10	11
0	0	1	1	0
1	1	0	0	1

Estado siguiente

El diagrama de estado para los dos casos a) y b), es el siguiente:



7) Analiza la siguiente red secuencial



Existen dos variables de estado Q_A y Q_B que pueden proporcionar 4 estados.

Analizando el circuito combinacional previo se obtienen las expresiones de entrada a los flip-flop:

$$J_A = \bar{x} \bar{Q}_B + x Q_A$$

$$J_B = Q_A$$

$$K_A = Q_B$$

$$K_B = \bar{x} \bar{Q}_A$$

La función de salida es $z = Q_A + \bar{Q}_B$

A continuación, nos proponemos obtener las transiciones y la función de salida, teniendo en cuenta la tabla de transición de estado del flip-flop tipo JK.

J(t)	K(t)	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

La tabla de transición de estados y salida puede representarse de varias formas. En la primera de ellas se incluyen columnas distintas para distintos valores de la entrada x .

E. Presente (t)		Entradas a FF			E. Próximo (t+1)		Salida z
		x=0	x=1				
$Q_A Q_B$	$J_A K_A$	$J_B K_B$	$J_A K_A$	$J_B K_B$	$Q_A Q_B$	$Q_A Q_B$	
00	10	01	00	00	10	00	1
01	01	01	01	00	00	01	0
10	10	10	10	10	11	11	1
11	01	10	11	10	01	01	1

Para obtener las tablas de entradas a flip-flop y salida, recurrimos a las expresiones que hemos obtenido del análisis de la red. Para obtener la tabla del Estado próximo ($t+1$) utilizamos la tabla de transición de estado del flip-flop particular. Por ejemplo, para el

primer caso se tiene $Q_A = 0, J_A = 1, K_A = 0$, el valor correspondiente a esta combinación es $Q_A(t+1) = 1$

El siguiente paso es dar una descripción de alto nivel tomando cuatro estados S_0, S_1, S_2, S_3 según la siguiente codificación:

Q_A	Q_B	S
0	0	S_0
0	1	S_1
1	0	S_2
1	1	S_3

Las correspondientes funciones de transición y salida están dadas en forma de tabla.

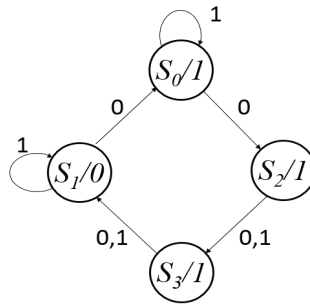
Estado actual	$x=0$	$x=1$	Salida
S_0	S_2	S_0	1
S_1	S_0	S_1	0
S_2	S_3	S_3	1
S_3	S_1	S_1	1

Estado próximo

Otra forma de resolver el problema es utilizar una sola tabla de transición de estados y salida como la siguiente:

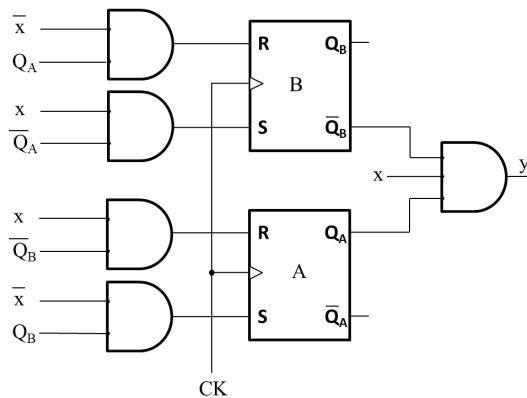
x	$Q_A(t)$	$Q_B(t)$	J_A	K_A	J_B	K_B	$Q_A(t+1)$	$Q_B(t+1)$	Z
0	0	0	1	0	0	1	1	0	1
0	0	1	0	1	0	1	0	0	0
0	1	0	1	0	1	0	1	1	1
0	1	1	0	1	1	0	0	1	1
1	0	0	0	0	0	0	0	0	1
1	0	1	0	1	0	0	0	1	0
1	1	0	1	0	1	0	1	1	1
1	1	1	1	1	1	0	0	1	1

El diagrama de estados es:



A la vista de la tabla se ve que es un autómata de Moore, en donde cada estado tiene asociada una salida.

8) Analizar el circuito secuencial de la figura adjunta y obtener las ecuaciones de transición de estados.



Existen: una variable de entrada x , una variable de salida y , y dos biestables SR temporizados llamados A y B .

Las expresiones de las entradas a los flip-flop son:

$$R_B = x\overline{Q_A}$$

$$S_B = x\overline{Q_A}$$

$$R_A = x\overline{Q_B}$$

$$S_A = x\overline{Q_B}$$

La salida es $y = xQ_A\overline{Q_B}$. Según estas expresiones, la tabla de estado es la siguiente:

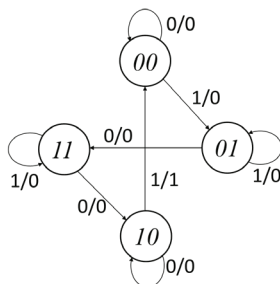
Entradas al cc.			Entradas a FF		Estado sig.		Salida
Q_A	Q_B	x	$S_A R_A$	$S_B R_B$	Q_A	Q_B	y
0	0	0	00	00	0	0	0
0	0	1	01	10	0	1	0
0	1	0	10	00	1	1	0
0	1	1	00	10	0	1	0
1	0	0	00	01	1	0	0
1	0	1	01	00	0	0	1
1	1	0	10	01	1	0	0
1	1	1	00	00	1	1	0

Se recuerda la tabla de transición de estado del flip-flop SR.

Estado actual $Q(t)$	Entradas		$S(t)$		$R(t)$
	00	01	10	11	
0	0	0	1	-	
1	1	0	1	-	

Estado siguiente $Q(t+1)$

El diagrama de estado obtenido a partir de la tabla de estado es:



A continuación, se obtienen las ecuaciones de transición y salida.

$$D_B = x_2 Q_A$$

Ya que la ecuación de transición de un flip-flop tipo D es $Q(t+1) = D(t)$, podemos obtener de forma directa las funciones de transición de ambos:

$$Q_A(t+1) = x_1 Q_B(t)$$

$$Q_B(t+1) = x_2 Q_A(t)$$

La salida es $z = x_1 \overline{Q_B}$

A partir de estas expresiones podemos obtener una tabla de estado binaria. En este caso, se formará una tabla de dos dimensiones cuyas variables de entrada son:

- Variables de entrada x_1, x_2 .
- Variables de estado Q_A, Q_B .

Este tipo de tabla es, en ocasiones, más cómoda de construir que la tabla de una dimensión, como la formada en el problema 5. Nos proponemos obtener la tabla de verdad del estado siguiente $Q_A(t+1)$ y $Q_B(t+1)$, y de la salida z , en función de las 4 entradas.

Estado presente	Entradas ($x_1 x_2$)				Entradas ($x_1 x_2$)			
$Q_A Q_B$	00	01	10	11	00	01	10	11
00	00	00	00	00	0	0	1	1
01	00	00	10	10	0	0	0	0
10	00	01	00	01	0	0	1	0
11	00	01	10	11	0	0	0	0

Estado siguiente		Salida	
$Q_A(t+1)$	$Q_B(t+1)$	z	

Para pasar a una descripción de alto nivel realizamos la siguiente decodificación:

Q_A	Q_B	S
0	0	S_0
0	1	S_1
1	0	S_2
1	1	S_3

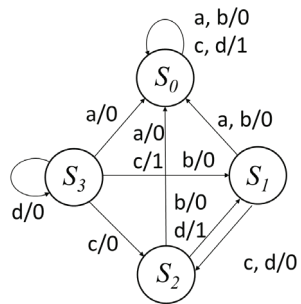
x_1	x_2	x
0	0	a
0	1	b
1	0	c
1	1	d

Se obtiene la siguiente tabla de estado de alto nivel:

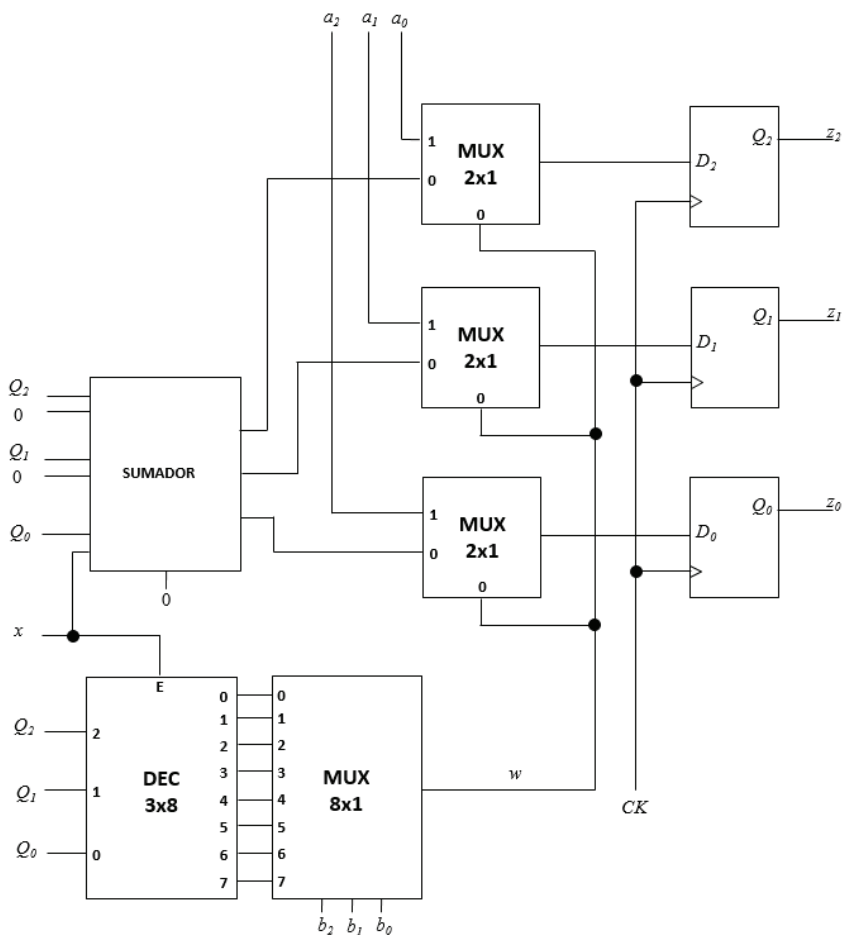
Estado actual	Entrada (x)			
	a	b	c	d
S_0	$S_0, 0$	$S_0, 0$	$S_0, 1$	$S_0, 1$
S_1	$S_0, 0$	$S_0, 0$	$S_2, 0$	$S_2, 0$
S_2	$S_0, 0$	$S_1, 0$	$S_0, 1$	$S_1, 1$
S_3	$S_0, 0$	$S_1, 0$	$S_2, 0$	$S_3, 0$

Estado siguiente, salida

El diagrama de estado es el siguiente:



10) Analizar el circuito secuencial de la figura.



En primer lugar, analizamos todas las variables de entrada, salida y estado que posee el circuito. Se observa una agrupación de variables binarias en vectores de variables. A partir de ahora una letra subrayada es un vector.

- $\underline{a} = (a_2, a_1, a_0)$ van a las entradas 1 de los 3 multiplexores.
- $\underline{D} = (D_2, D_1, D_0)$ son las salidas de los multiplexores que van a las entradas de los flip-flop.
- $\underline{b} = (b_2, b_1, b_0)$ son las entradas de selección del multiplexor que está acoplado al decodificador.

- $\underline{Q} = (Q_2, Q_1, Q_0)$ son las variables de estado, coinciden con las salidas del sistema y constituyen un operando del sumador.
- $\underline{x} = (0, 0, x)$ es el otro operando del sumador.
- $\underline{z} = (z_2, z_1, z_0)$ son las salidas del sistema.

Analicemos en primer lugar las funciones implementadas por el decodificador y multiplexor. Se dispone de una variable auxiliar w .

$$w = \begin{cases} 1 & \text{si } (x = 1) \text{ y } (\underline{Q} = \underline{b}) \\ 0 & \text{en caso contrario} \end{cases}$$

En el sistema, los 3 multiplexores proporcionan una salida $\underline{D}$, especificada como:

$$\underline{D} = \begin{cases} \underline{a} & \text{si } w = 1 \\ \text{Suma}(\underline{Q}, \underline{x}) & \text{si } w = 0 \end{cases}$$

Sustituyendo la especificación de w en $\underline{D}$ se obtiene:

$$\underline{D} = \begin{cases} \underline{a} & \text{si } (x = 1) \text{ y } (\underline{Q} = \underline{b}) \\ \text{Suma}(\underline{Q}, \underline{x}) & \text{en caso contrario} \end{cases}$$

Ya que en una báscula D se cumple $\underline{Q}(t+1) = \underline{D}(t)$, la función de transición del sistema digital que estamos analizando será:

$$\underline{Q}(t+1) = \begin{cases} \underline{a} & \text{si } (x = 1) \text{ y } (\underline{Q}(t) = \underline{b}) \\ \text{Suma}(\underline{Q}(t), \underline{x}(t)) & \text{en caso contrario} \end{cases}$$

La salida es $\underline{z} = \underline{Q}$

11) Diseñar un sistema secuencial con la siguiente tabla de estados. Hacerlo con flip-flop SR. Se asume que la combinación $x_1=0, x_2=0$ no ocurre nunca.

Estado actual	Entradas ($x_1 x_2$)		
	01	10	11
a	b,0	c,1	c,0
b	a,0	d,1	d,0
c	d,0	c,0	a,1
d	c,0	a,0	d,1

E. siguiente, salida

Las entradas x_1 y x_2 , así como la salida ya están codificadas en binario. Como el número de estados es 4, podemos codificarlos con 2 variables binarias Q_1 y Q_2 . Cada una de ellas será la salida de un flip-flop SR. Utilizamos la siguiente codificación:

Estado	Q_1	Q_2
a	0	0
b	0	1
c	1	0
d	1	1

Tendremos que determinar las entradas a los flip-flop S_1, R_1, S_2, R_2 en función de x_1, x_2, Q_1 y Q_2 , utilizando las tablas de excitación correspondiente al flip-flop SR. Con esto determinaremos el circuito combinacional previo a los flip-flops.

Entradas al cc.				E. siguiente		Entradas a FF		Salida
x_1	x_2	Q_1	Q_2	Q_1	Q_2	S_1R_1	S_2R_2	Z
0	0	0	0	-	-	--	-	-
0	0	0	1	-	-	--	-	-
0	0	1	0	-	-	--	-	-
0	0	1	1	-	-	--	-	-
0	1	0	0	0	1	0-	10	0
0	1	0	1	0	0	0-	01	0
0	1	1	0	1	1	-0	10	0
0	1	1	1	1	0	-0	01	0
1	0	0	0	1	0	10	0-	1
1	0	0	1	1	1	10	-0	1
1	0	1	0	1	0	-0	0-	0
1	0	1	1	0	0	01	01	0
1	1	0	0	1	0	10	0-	0
1	1	0	1	1	1	10	-0	0
1	1	1	0	0	0	01	0-	1
1	1	1	1	1	1	-0	-0	1

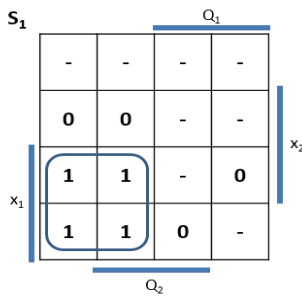
Recordamos la tabla de excitación de los flip-flop tipo SR.

Q(t)	Q(t+1)	S(t)	R(t)
0	0	0	-
0	1	1	0
1	0	0	1
1	1	-	0

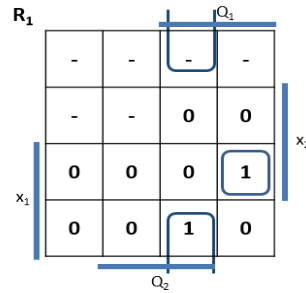
Tabla 2

Las columnas de Estado siguiente y de salida se obtienen directamente de la tabla 1. Las columnas de Entradas a los FF se obtienen con la ayuda de la tabla 2.

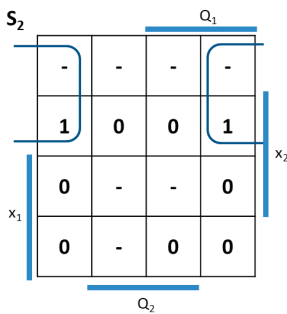
El paso siguiente será determinar las expresiones de conmutación de las entradas a los flip-flop y salida. Para ello, utilizamos de los mapas de Karnaugh.



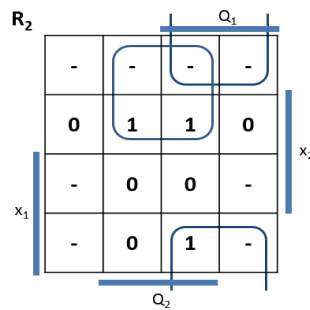
$$S_1 = x_1 \bar{Q}_1$$



$$R_1 = \bar{x}_2 Q_2 Q_1 + x_1 x_2 \bar{Q}_2 Q_1$$



$$S_2 = \bar{x}_1 \bar{Q}_2$$

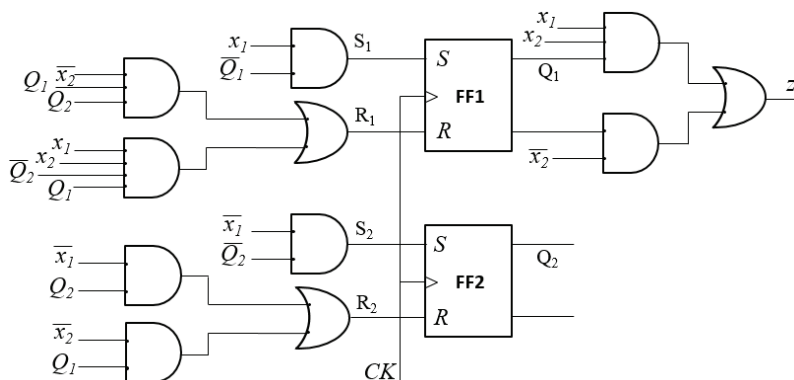


$$R_2 = \bar{x}_1 Q_2 + x_2 \bar{Q}_1$$

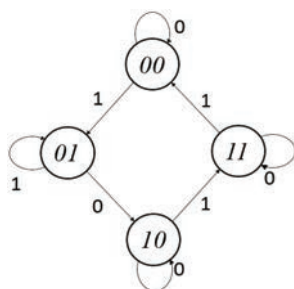
z	Q_1			
	-	-	-	-
	0	0	0	0
	0	0	1	1
x_1	Q_2			
	1	1	0	0
x_2				

$$z = \overline{x_2} \overline{Q_1} + x_1 x_2 Q_1$$

Por último, se realiza el esquema gráfico de la implementación.



- 12) Diseñar un circuito secuencial síncrono, usando biestables JK, cuyo diagrama de estados es:



Como se puede ver, disponemos de 4 estados con codificación binaria ya asignada. Hay una variable de entrada binaria y no hay ninguna variable binaria de salida, por lo que se pueden considerar como salidas los estados presentes de los biestables.

Para implementar 4 estados dispondremos de 2 biestables.

Utilizamos la tabla de excitación de los flip-flop JK para obtener la tabla de excitación del sistema.

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

Estado actual		Entrada	Estado siguiente		Entradas a FF	
Q _A	Q _B	x	Q _A (t+1)	Q _B (t+1)	J _A K _A	J _B K _B
0	0	0	0	0	0-	0-
0	0	1	0	1	0-	1-
0	1	0	1	0	1-	-1
0	1	1	0	1	0-	-0
1	0	0	1	0	-0	0-
1	0	1	1	1	-0	1-
1	1	0	1	1	-0	-0
1	1	1	0	0	-1	-1

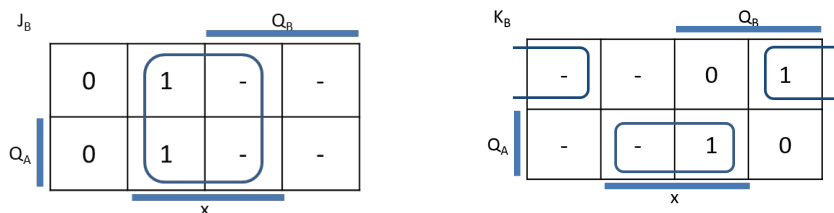
Queda obtener las funciones de conmutación de las entradas a los flip-flop.

J _A	Q _B			
	0	0	0	1
Q _A	-	-	-	-
x				

$$J_A = Q_B \bar{x}$$

K _A	Q _B			
	-	-	-	-
Q _A	0	0	1	0
x				

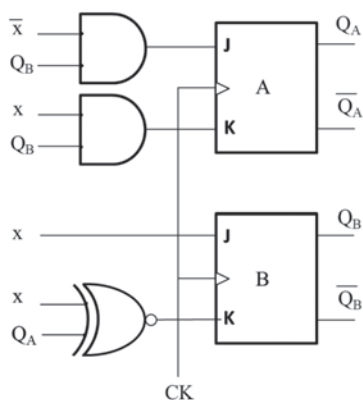
$$K_A = Q_B x$$



$$J_B = x$$

$$K_B = Q_A \quad x = \overline{Q_A} \oplus x$$

El diagrama lógico es el siguiente:



13) Diseñar usando biestables D una red secuencial con cuatro flip-flop *A*, *B*, *C*, *D*, de forma que:

- Los estados siguientes de *B*, *C* y *D* son iguales a los estados presentes de *A*, *B* y *C*, respectivamente.
- El estado siguiente de *A* es igual a la función XOR de los estados presentes de *C* y *D*.

Según el enunciado del problema, las ecuaciones de estado del sistema son:

$$Q_A(t+1) = Q_C \oplus Q_D$$

$$Q_B(t+1) = Q_A(t)$$

$$Q_C(t+1) = Q_B(t)$$

$$Q_D(t+1) = Q_C(t)$$

Cada flip-flop transfiere o desplaza su contenido al siguiente a cada pulso de reloj. Como las ecuaciones de estado son muy sencillas, podemos utilizar biestables del tipo D para que la implementación sea inmediata.

Las expresiones de las entradas a los biestables son:

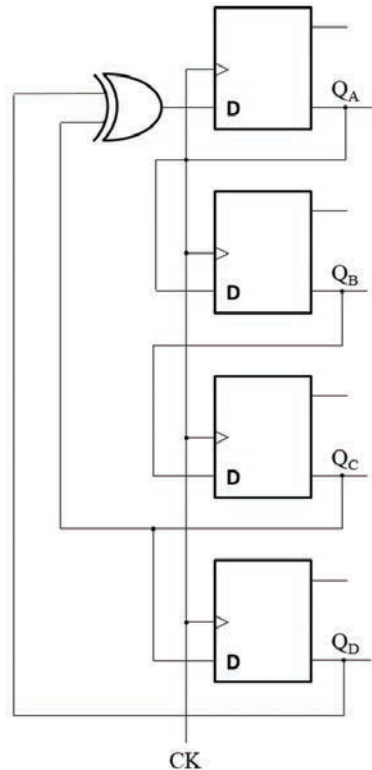
$$D_A = Q_C \oplus Q_D$$

$$D_B = Q_A$$

$$D_C = Q_B$$

$$D_D = Q_C$$

El diagrama lógico correspondiente es:



El ejercicio podría haberse resuelto realizando la tabla de excitación del circuito y calculando las expresiones de las entradas a los biestables.

$Q_A(t)$	$Q_B(t)$	$Q_C(t)$	$Q_D(t)$	D_A $Q_A(t+1)$	D_B $Q_B(t+1)$	D_C $Q_C(t+1)$	D_D $Q_D(t+1)$
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	1	0	0	1
0	0	1	1	0	0	0	1
0	1	0	0	0	0	1	0
0	1	0	1	1	0	1	0
0	1	1	0	1	0	1	1
0	1	1	1	0	0	1	1
1	0	0	0	0	1	0	0
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	0	1	0	1
1	1	0	0	0	1	1	0
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	1

$$D_A = Q_C \oplus Q_D$$

$$D_B = Q_A$$

$$D_C = Q_B$$

$$D_D = Q_C$$

- 14) Una máquina expendedora de refrescos está controlada por un circuito secuencial que recibe como entradas las monedas que va introduciendo el usuario, codificadas según la tabla adjunta. El usuario debe introducir monedas hasta alcanzar la cantidad de 1,00€, que será el precio del refresco. Por simplicidad, suponer que no existen las monedas que no se citan aquí. Sabiendo que los estados se definen en función del dinero acumulado que tiene la máquina en un determinado momento, se pide dibujar el diagrama de estados. ¿Cuántos biestables se necesitan para resolver el problema? Suponer que cuando se alcanza el importe de 1,00 € o superior, hay una salida que se pone a 1, y queda a 0 en el resto de los casos.

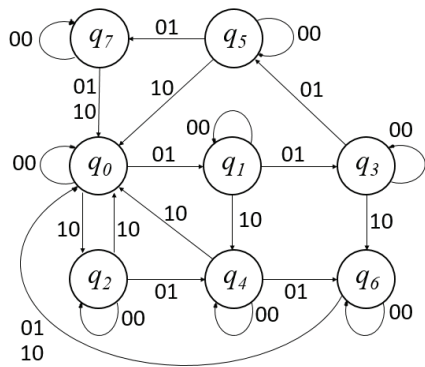
NOTA: Escribir en cada flecha de transición la entrada que se necesita. Cuando se alcanza la cantidad de dinero de 1,00 € o superior, se vuelve al estado inicial y no procede la devolución de dinero.

Codificación de las monedas de entrada	
0 0	No se ha introducido ninguna moneda
0 1	Moneda introducida: 0,20€
1 0	Moneda introducida: 0,50€
1 1	No utilizado

Los estados se definen según el dinero que contiene la máquina expendedora.

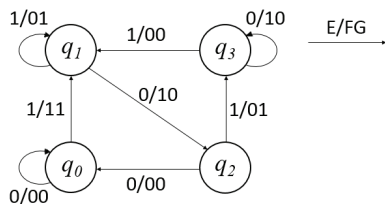
Estado	Dinero
q_0	No hay dinero
q_1	0,20€
q_2	0,50€
q_3	0,40€
q_4	0,70€
q_5	0,60€
q_6	0,90€
q_7	0,80€

El diagrama de estados es el siguiente:



Ya que existen 8 estados el número de biestables necesarios para diseñar el sistema es 3.

- 15) Dado el diagrama de estados de la figura, con una entrada E y dos salidas F y G , se pide encontrar la tabla de transición y excitación usando biestables T , D y JK . ¿Qué biestables usarías para hacer el diseño más sencillo si la excitación de los mismos se hace con puertas lógicas? Razona la respuesta.

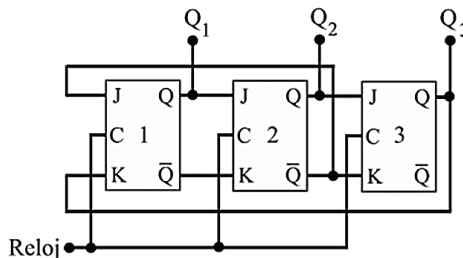


Dado el diagrama de estados se obtiene la tabla de excitación del sistema:

E	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	F	G	T_1	T_0	D_1	D_0	J_1K_1	J_0K_0
0	0	0	0	0	0	0	0	0	0	0	0X	0X
0	0	1	1	0	1	0	1	1	1	0	1X	X1
0	1	0	0	0	0	0	1	0	0	0	X1	0X
0	1	1	1	1	1	0	0	0	1	1	X0	X0
1	0	0	0	1	1	1	0	1	0	1	0X	1X
1	0	1	0	1	0	1	0	0	0	1	0X	X0
1	1	0	1	1	0	1	0	1	1	1	X0	1X
1	1	1	0	1	0	0	1	0	0	1	X1	X0

La mejor opción son los biestables JK . Bastaría ver la expresión algebraica (Karnaugh) de las funciones.

- 16) Suponer que el circuito de la figura adjunta se inicializa a $Q_1=0$, $Q_2=1$, $Q_3=0$. Calcula la secuencia de estados para los cinco primeros ciclos de reloj:



Las expresiones de las entradas a los biestables son:

$$J_1 = \bar{Q}_2 \quad K_1 = Q_3$$

$$J_2 = \bar{K}_2 = D_2 = Q_1$$

$$J_3 = \bar{K}_3 = D_3 = Q_2$$

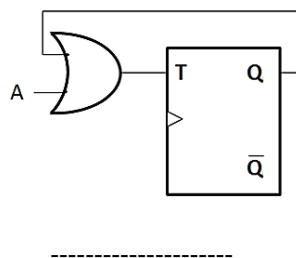
La tabla de transición de estados es la siguiente:

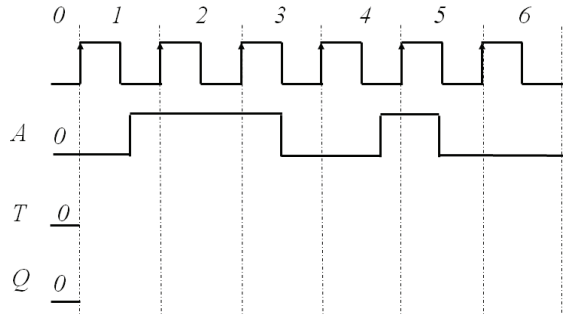
$Q_1(t)$	$Q_2(t)$	$Q_3(t)$	J_1	K_1	D_2	D_3	$Q_1(t+1)$	$Q_2(t+1)$	$Q_3(t+1)$
0	0	0	1	0	0	0	1	0	0
0	0	1	1	1	0	0	1	0	0
0	1	0	0	0	0	1	0	0	1
0	1	1	0	1	0	1	0	0	1
1	0	0	1	0	1	0	1	1	0
1	0	1	1	1	1	0	0	1	0
1	1	0	0	0	1	1	1	1	1
1	1	1	0	1	1	1	0	1	1

Sabiendo que el estado inicial es 010, la secuencia es la siguiente:

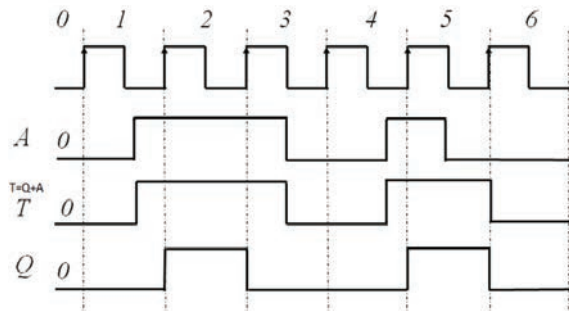
010, 001, 100, 110, 111, 011....

17) Dibuja el cronograma concerniente a la siguiente figura:

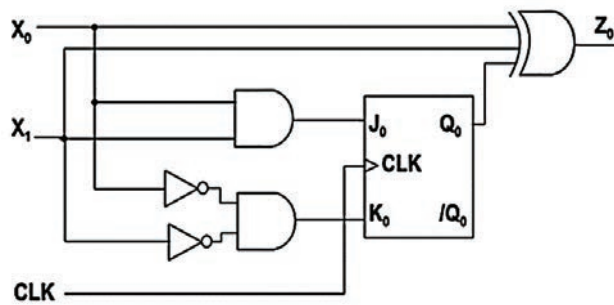




Teniendo en cuenta que $T_0 = A + Q_0$ se obtiene:



18) Calcula el diagrama de estados del circuito de la figura.



Las expresiones de las entradas al biestable y salida son:

$$Z_0 = x_0 \oplus x_1 \oplus Q_0$$

$$J_0 = x_0 x_1$$

$$K_0 = \bar{x}_0 \bar{x}_1$$

La tabla de transición de estados y salida es:

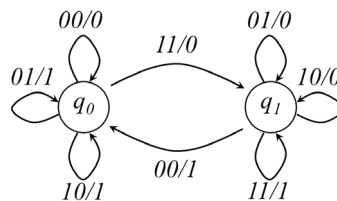
x_1	x_0	$Q_{(t)}$	J_0	K_0	$Q_{(t+1)}$	Z_0
0	0	0	0	1	0	0
0	0	1	0	1	0	1
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	1	0	1	0
1	1	1	1	0	1	1

Suponiendo la codificación:

$$q_0 : (Q=0)$$

$$q_1 : (Q=1)$$

El diagrama de estados es:



19) Un circuito secuencial tiene una entrada E , una salida S y queda implementado con las siguientes funciones lógicas:

$$J_1 = \bar{E} \cdot Q_0 \quad K_1 = K_0 = 1$$

$$J_0 = \bar{E} \cdot \bar{Q}_1 \quad S = \bar{E} \cdot Q_1$$

donde J_i , K_i y Q_i corresponden a las entradas y estado del biestable i . El sistema se inicializa a $Q_1=0$ y $Q_0=0$.

Para dicha implementación, obtenga el diagrama de estados de la máquina. Sobre ese diagrama realice las optimizaciones que considere oportunas y explique la función que realiza el circuito.

Codificación de estados:

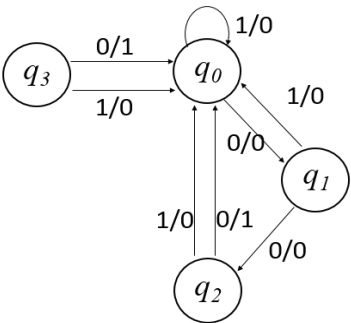
$J_1 = \overline{E}Q_0$
 $J_0 = \overline{E}\overline{Q_1}$
 $K_1 = K_0 = 1$
 $S = \overline{E}Q_1$

	Q_1	Q_0
q_0	0	0
q_1	0	1
q_2	1	0
q_3	1	1

La tabla de transición de estados y salida es

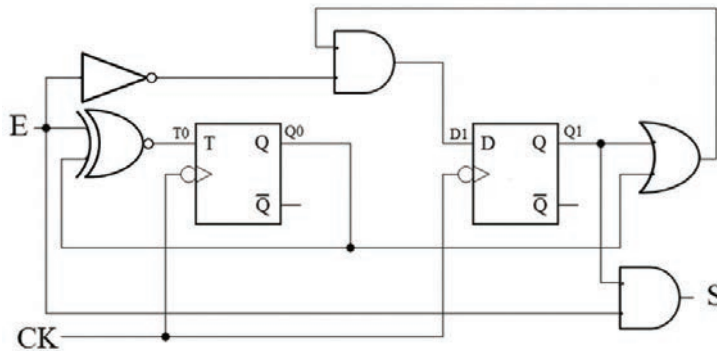
E	$Q_1(t)$	$Q_0(t)$	J_1K_1	J_0K_0	$Q_1(t+1)$	$Q_0(t+1)$	S
0	0	0	0 1	1 1	0	1	0
0	0	1	1 1	1 1	1	0	0
0	1	0	0 1	0 1	0	0	1
0	1	1	1 1	0 1	0	0	1
1	0	0	0 1	0 1	0	0	0
1	0	1	0 1	0 1	0	0	0
1	1	0	0 1	0 1	0	0	0
1	1	1	0 1	0 1	0	0	0

Y el diagrama de estados será el siguiente:



El estado q_3 es inalcanzable, y desde él sólo se va al estado q_0 inicial. Se podría eliminar. Según la salida del sistema se trata de un reconocedor de tres ceros consecutivos.

20) Halla la tabla de transición de estados y salidas del circuito siguiente. El orden de las variables de entrada al sistema secuencial será EQ_1Q_0 .



Las funciones de entrada a los biestables y la salida que realiza el circuito son:

$$T_0 = \overline{(E \oplus Q_0)}$$

$$D_1 = E(Q_1 + Q_0)$$

$$S = EQ_1$$

La tabla de verdad será la siguiente (los valores del estado siguiente se han calculado teniendo en cuenta el estado actual y las entradas D de los biestables):

E	$Q_1(t)$	$Q_0(t)$	D_1	T_0	$Q_1(t+1)$	$Q_0(t+1)$	S
0	0	0	0	1	0	1	0
0	0	1	1	0	1	1	0
0	1	0	1	1	1	1	0
0	1	1	1	0	1	1	0
1	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0
1	1	0	0	0	0	0	1
1	1	1	0	1	0	0	1

Se trata de un registro de desplazamiento serie-serie

Ciclo	Input	Q_2	Q_1	Q_0
0	1	0	0	0
1	1	1	0	0
2	0	1	1	0
3	1	0	1	1
4	0	1	0	1
5	1	0	1	0
6	-	1	0	1

23) Diseñar un contador síncrono que siga la secuencia: 0, 1, 2, 4, 5, 6, 0, 1, ... utilizando biestables JK.

Se trata de un contador de 3 bits, que pasa por seis estados. Necesitamos 3 biestables.

La secuencia del contador y la tabla de excitación son:

Estado actual			Estado próximo			Entradas a FF		
Q_A	Q_B	Q_C	Q_A	Q_B	Q_C	$J_A K_A$	$J_B K_B$	$J_C K_C$
0	0	0	0	0	1	0 -	0 -	1 -
0	0	1	0	1	0	0 -	1 -	- 1
0	1	0	1	0	0	1 -	- 1	0 -
0	1	1	-	-	-	--	--	--
1	0	0	1	0	1	- 0	0 -	1 -
1	0	1	1	1	0	- 0	1 -	- 1
1	1	0	0	0	0	- 1	- 1	0 -
1	1	1	-	-	-	--	--	--

La tabla de excitación de un flip-flop JK es:

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

No hace falta utilizar los mapas de Karnaugh para obtener las expresiones de conmutación de las entradas a los biestables, ya que por observación de la tabla se obtienen directamente las expresiones:

$$J_A = Q_B$$

$$J_B = Q_C$$

$$J_C = \overline{Q_B}$$

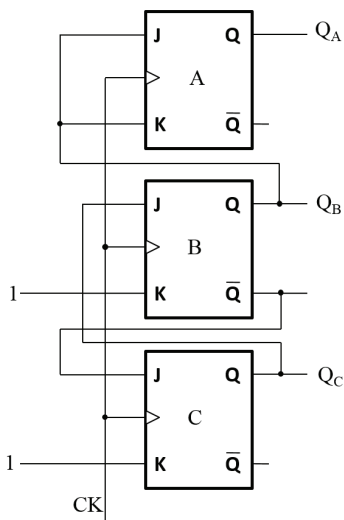
$$K_A = Q_B$$

$$K_B = 1$$

$$K_C = 1$$

Suponemos que el estado inicial del contador es 000.

El diagrama lógico es el siguiente.



24) Diseñar un contador que cuente dígitos decimales de acuerdo al código BCD con pesos 2421. Utilizar biestables T.

Nota: En este código, dos dígitos que suman 9 tienen códigos binarios complementarios.

La codificación 2421 de los 10 dígitos decimales es:

Dígito decimal	Código 2421
0	0000
1	0001
2	0010
3	0011
4	0100
5	1011
6	1100
7	1101
8	1110
9	1111

Esto quiere decir que, en realidad, hay que diseñar un contador que siga la secuencia decimal 0, 1, 2, 3, 4, 11, 12, 13, 14, 15.

La tabla de excitación será:

Estado actual				Estado próximo				Entradas a FF			
Q _A	Q _B	Q _C	Q _D	Q _A	Q _B	Q _C	Q _D	T _A	T _B	T _C	T _D
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	1	0	1	1	1	1	1	1
-	-	-	-	-	-	-	-	-	-	-	-
Condiciones de no-importa del 5 al 10											
-	-	-	-	-	-	-	-	-	-	-	-
1	0	1	1	1	1	0	0	0	1	1	1
1	1	0	0	1	1	0	1	0	0	0	1
1	1	0	1	1	1	1	0	0	0	1	1
1	1	1	0	1	1	1	1	0	0	0	1
1	1	1	1	0	0	0	0	1	1	1	1

Obtenemos las expresiones de conmutación de las entradas a los biestables utilizando los mapas de Karnaugh de 4 variables.

T_A

		Q_C			
		0	0	0	0
	1	-	-	-	-
Q_A	0	0	1	0	
	-	-	0	-	
		Q_D			

Q_B

$$T_A = \overline{Q_A}Q_B + Q_BQ_CQ_D$$

T_B

		Q_c			
		0	0	1	0
	1	-	-	-	-
Q_A	0	0	1	0	
	-	-	1	-	
		Q_D			

Q_B

$$T_B = \overline{Q_A}Q_B + Q_CQ_D$$

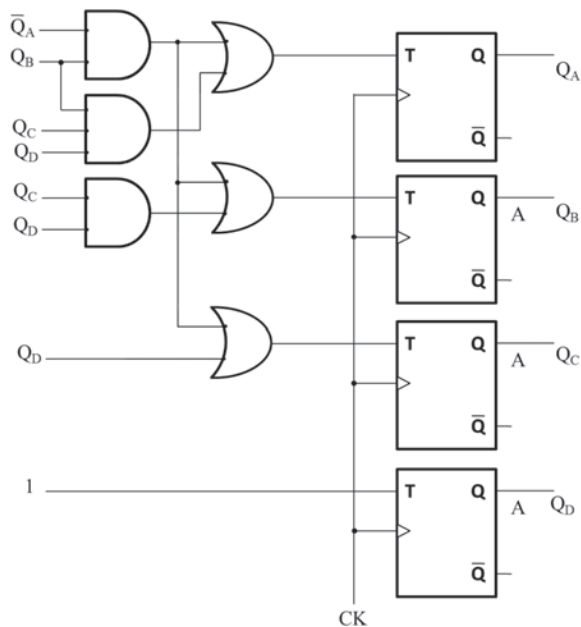
T_C

		Q_C	
		0	1
	1	-	-
Q_A	0	1	0
	-	-	1
		Q_D	

Q_B

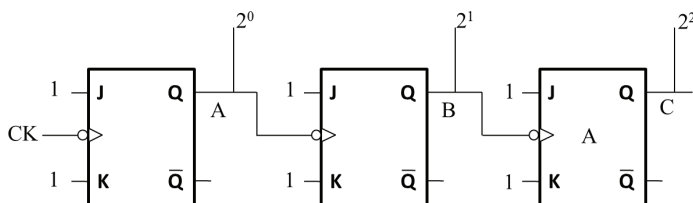
$$T_C = Q_D + \overline{Q_A}Q_B \text{ y } T_D = 1$$

El diagrama lógico es el siguiente:



- 25) Diseñar un contador creciente módulo-8 mediante un sistema contador de rizado asíncrono, utilizando biestables JK. Suponer que los cambios de estado se producen en salto de flanco negativo. Hacer una representación temporal de los biestables.

Necesitaremos 3 biestables JK dispuestos de tal manera, que la salida de uno sea la entrada de reloj del siguiente. Todas las entradas J y K de los biestables se mantienen a 1. La salida está codificada en binario y corresponde a los estados presentes de los biestables. El biestable A proporciona el bit menos significativo (peso 2^0), y el C proporciona el más significativo (peso 2^2): El esquema es:

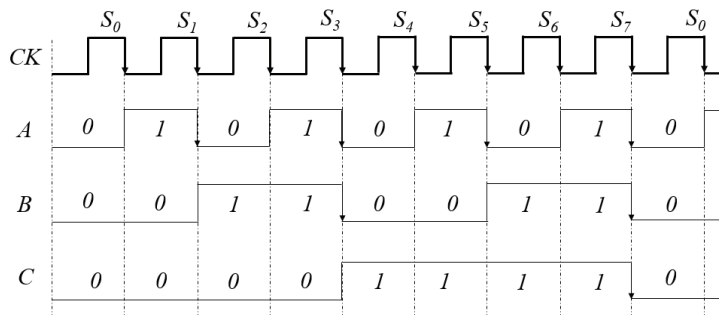


Recordemos la tabla de transición de estados para un biestable JK.

$J(t)$	$K(t)$	$Q(t)$	$Q(t+1)$
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 *

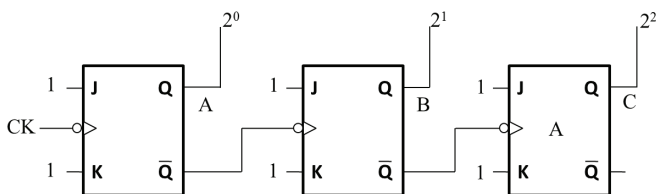
Se comprueba que, según esta tabla, se producen transiciones de 0 a 1, o de 1 a 0, cuando haya un salto de flanco negativo. Se recuerda que las entradas J y K están conectadas permanentemente a 1. En la tabla se han marcado con la señal * las dos transiciones que pueden tener lugar.

Teniendo en cuenta lo anterior, a continuación, se muestra una representación temporal del registro contador.



Se comprueba que el registro toma de forma autónoma el estado 0 después del 7, por lo que no hay que añadir ningún circuito de puesta a cero cuando se llegue al final del conteo.

Para diseñar el mismo circuito decreciente, se dispone el registro tomando las salidas $\overline{Q}_A$ y $\overline{Q}_B$ en vez de Q_A y Q_B conectadas a las entradas de reloj de los biestables B y C, según se muestra en la siguiente figura. $\overline{Q}_A$



26) Diseñar un contador asíncrono módulo-10 con biestables JK sin actuar sobre las entradas asíncronas.

Después de realizar la implementación del contador módulo-8 con biestables JK, realizar un contador módulo-10 es sencillo. Tendremos que disponer de 4 biestables y habrá que diseñar un pequeño circuito que haga cargar 0000 en el registro cuando se haya llegado a 1001.

En primer lugar, realizamos un diagrama temporal del registro.

27) Diseñar un generador cíclico de la siguiente secuencia, utilizando flip-flops JK y puertas NAND.

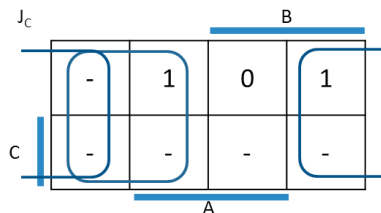
Impulso del reloj	C	B	A
1	0	0	1
2	1	0	0
3	0	1	0
4	1	0	1
5	1	1	0
6	0	1	1

Es un sistema contador cíclico. Consideramos la salida como el estado presente del registro.

La tabla de excitación del sistema es:

Estado actual			Estado próximo			Entradas a FF		
C	B	A	C	B	A	$J_C K_C$	$J_B K_B$	$J_A K_A$
0	0	0	-	-	-	--	--	--
0	0	1	1	0	0	1-	0-	-1
0	1	0	1	0	1	1-	-1	1-
0	1	1	0	0	1	0-	-1	-0
1	0	0	0	1	0	-1	1-	0-
1	0	1	1	1	0	-0	1-	-1
1	1	0	0	1	1	-1	-0	1-
1	1	1	-	-	-	--	--	--

Veamos las expresiones de conmutación de las entradas a los biestables:



$$J_C = \overline{A} + \overline{B} = \overline{(AB)}$$

Las demás expresiones son inmediatas y se obtienen directamente de la tabla.

$$K_C = \overline{A}$$

$$J_B = C$$

$$J_A = B$$

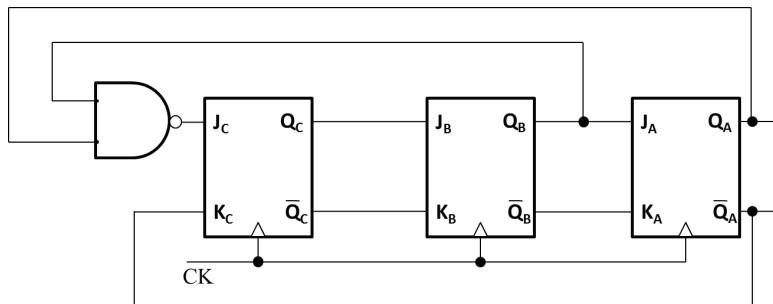
$$K_B = \overline{C}$$

$$K_A = \overline{B}$$

Está claro que se trata de un registro de desplazamiento a la derecha, en el que la información se traspa de un biestable al siguiente, y con una función de realimentación dada por $\overline{(AB)}$. Se pueden visualizar los desplazamientos a la derecha.

Impulso del reloj	C	B	A
1	0	0	1
2	1	0	0
3	0	1	0
4	1	0	1
5	1	1	0
6	0	1	1

El esquema es:



28) Diseñar un sumador serie.

Recibe los dos sumandos en serie, de menor a mayor peso, bit a bit en cada período de reloj. Al mismo tiempo genera las sumas parciales y los acarreo. La necesidad de

memoria está en recordar si para la etapa siguiente hay o no acarreo. Lo mismo que cuando sumamos mentalmente recordamos si "nos llevamos una" o no, el estado que hay que recordar es si hay o no acarreo.

Por tanto, en circuito secuencial que implementa un sumador de 4 bits debe tener una tabla de verdad del siguiente tipo:

Entradas y Estado actual			Salida y E. próximo		Excitación biestable	
E_1	E_2	$Q_0(t)$	SUMA	$Q_0(t+1)$	J_0	K_0
0	0	0	0	0	0	X
0	0	1	1	0	X	1
0	1	0	1	0	0	X
0	1	1	0	1	X	0
1	0	0	1	0	0	X
1	0	1	0	1	X	0
1	1	0	0	1	1	X
1	1	1	1	1	X	0

El circuito que resuelve esta situación será el siguiente, cuyas ecuaciones son:

		$E_2 Q_0$				
		J_0	00	01	11	10
E_1	0	0	X	X	0	
	1	0	X	X	1	

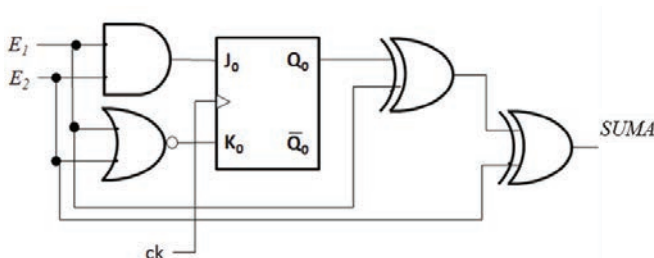
$$J_0 = E_1 \cdot E_2$$

		E_2Q_0				
		K_0	00	01	11	10
E_1	0		X	1	0	X
	1		X	0	0	X

$$K_0 = \overline{E_1} \cdot \overline{E_2} = \overline{E_1 + E_2}$$

$$SUMA = Q_0(t) \oplus E_1 \oplus E_2$$

El diagrama lógico es:



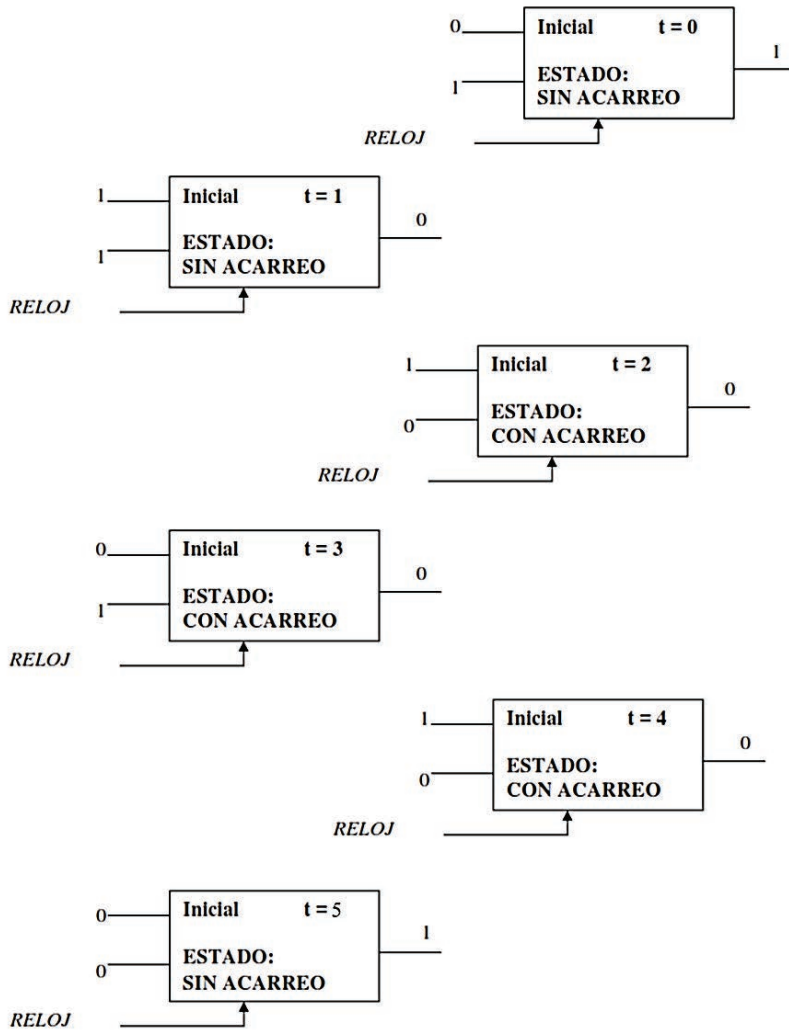
Ejemplo: sumar $22 + 11$

$E1: 000010110$

$E2: 000001011$

$SUMA: 000100001$

Evolución temporal del sumador serie de 1 bit



29) Diseñar un circuito reconocedor de la secuencia serie 1101 con biestables JK. Cuando reconozca la secuencia debe volver al estado inicial. Diseñarlo como autómata de Mealy.

Supongamos que se quiere detectar la subsecuencia 1101 dentro de una secuencia de bits de longitud indefinida.

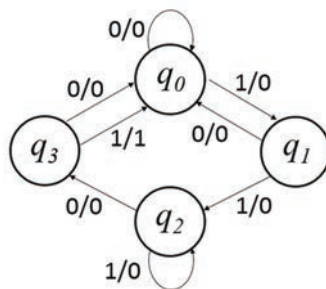
Vamos a hacerlo usando el diseño de circuitos secuenciales.

La definición de estados es:

- q_0 : estado inicial. Resto de estados.
- q_1 : se ha recibido un 1 en el último bit.
- q_2 : se han recibidos dos 1's en los dos últimos bits.
- q_3 : se ha recibido un 0 en el último bit después de que se hayan recibidos dos 1's en los dos anteriores bits.

Tomando la codificación de la figura se obtiene el siguiente diagrama de estado

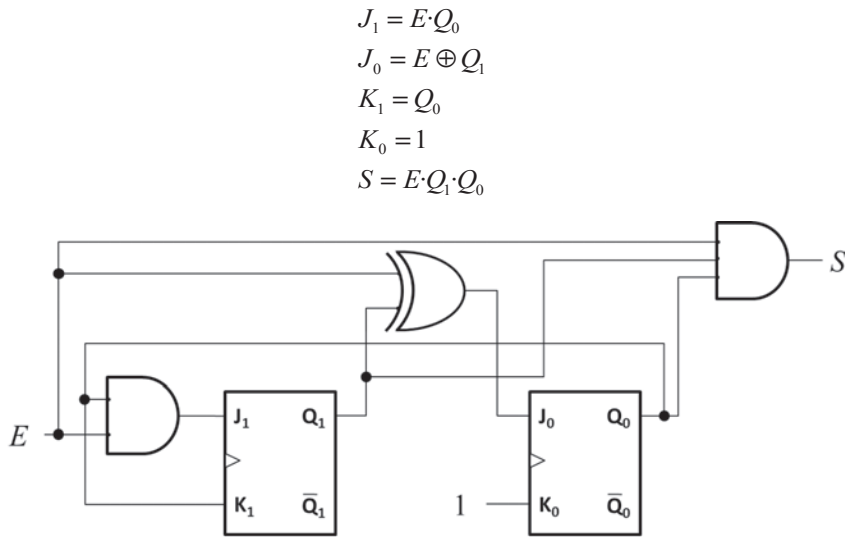
	Q_1	Q_0
q_0	0	0
q_1	0	1
q_2	1	0
q_3	1	1



La tabla de excitación es la siguiente:

E	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	J_1	K_1	J_0	K_0	S
0	0	0	0	0	0	X	0	X	0
0	0	1	0	0	0	X	X	1	0
0	1	0	1	1	X	0	1	X	0
0	1	1	0	0	X	1	X	1	0
1	0	0	0	1	0	X	1	X	0
1	0	1	1	0	1	X	X	1	0
1	1	0	1	0	X	0	0	X	0
1	1	1	0	0	X	1	X	1	1

Las expresiones de entradas a biestables y salida son:



30) Diseñar un circuito generador de paridad par de 3 bits. Proporcionar:

- Paso de las especificaciones verbales al diagrama de estados.
- Construcción de la tabla de transición de estados y tabla de excitación.
- Minimización de las funciones e implementación del circuito.

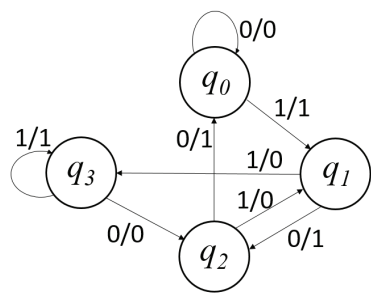
Se trata de diseñar un generador secuencial de paridad par de cadenas de 3 bits serie. Por ejemplo, si la secuencia de entrada es $X = 000110101\dots$ la salida deberá ser $Z = 0100010\dots$ Utilizar para ello biestables tipo D.

a) Diagrama de estados:

Como puede apreciarse, para poder calcular cada paridad de una serie de tres bits es necesario conservar los dos últimos bits, más el bit que se introduce en la entrada. Es necesario por lo tanto que el circuito secuencial sea capaz de recordar cuáles fueron estos dos últimos bits. De esta manera el número de estados diferentes será 4, correspondientes a las cuatro posibles combinaciones de 2 bits (00, 01, 10 y 11).

En nuestro caso el diagrama resultante será el de la figura adjunta, en la que los diferentes estados q_0 a q_3 corresponden a la descripción de la tabla adjunta.

Estado	Definición de estados
q_0	Dos últimos bits = 00
q_1	Dos últimos bits = 01
q_2	Dos últimos bits = 10
q_3	Dos últimos bits = 11



Supongamos por ejemplo que nos hallamos en el estado q_1 , lo que significa que después del último cálculo de paridad tenemos almacenada la secuencia 01. Como puede apreciarse en el diagrama, un 1 en la entrada daría lugar a la combinación 011, cuya paridad es par. La salida por lo tanto será 0, y el siguiente estado q_3 , puesto que ahora los dos últimos bits serán 11.

b) Tabla de transición de estados y tabla de excitación:

Tomando el diagrama de estados de la figura anterior debe generarse la tabla de estados, en la que a partir de las entradas y el estado actual se obtendrá el siguiente estado y la salida.

E	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	z
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	0	0	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	1	0
1	1	0	0	1	0
1	1	1	1	1	1

Para calcular la tabla de excitación es necesario disponer de la tabla de excitación del biestable (del D en este caso):

$Q(t)$	$Q(t+1)$	D
0	0	0
0	1	1
1	0	0
1	1	1

A partir de esta tabla, conociendo $Q(t)$ y $Q(t+1)$ para ambos bits, podemos calcular los valores de entrada a los biestables D_1 y D_0 . La nueva tabla quedará de la siguiente manera.

x	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	z	D_1	D_0
0	0	0	0	0	0	0	0
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	0
0	1	1	1	0	0	1	0
1	0	0	0	1	1	0	1
1	0	1	1	1	0	1	1
1	1	0	0	1	0	0	1
1	1	1	1	1	1	1	1

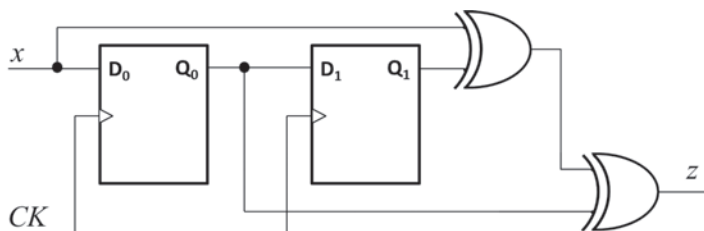
c) Minimización de las funciones e implementación del circuito:

Simplificando por Karnaugh obtendremos las funciones de entrada a los biestables (D_1 y D_0) para el cálculo del nuevo estado, y la función de salida Z , implementadas en la figura siguiente:

$$D_1 = Q_0$$

$$D_0 = x$$

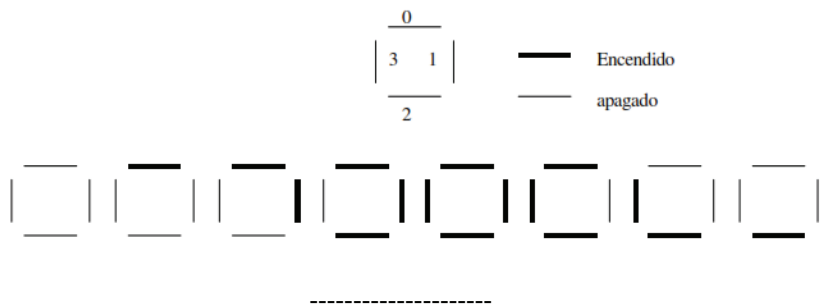
$$Z = x \oplus Q_1 \oplus Q_0$$



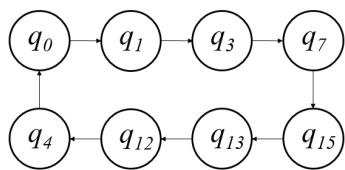
- 31) Se dispone de 4 barras luminosas dispuestas y numeradas según la figura. Se quiere diseñar un circuito que realice una secuencia cíclica de iluminación controlada por una señal de reloj de frecuencia baja. La secuencia requerida es cíclica y es la que se representa en la figura.

Implementar un circuito digital que consiga realizar la secuencia descrita y que comience con todos los segmentos apagados, teniendo en cuenta que la iluminación de un segmento se logra proporcionando un nivel lógico 1. Disponer de biestables tipo T, decodificador/es para implementar las entradas a los biestables, y puertas

OR necesarias. Respetar la numeración de los luminosos a la hora de codificar los estados.



Como muestra la figura del enunciado, pueden presentarse 8 configuraciones diferentes en las barras luminosas, por lo que el número de estados necesario será también 8, una para cada configuración:



Como son 8 estados, podríamos asignar un código consecutivo del 000 al 111 y utilizar 3 bits para codificar cada uno de ellos. Sin embargo, como se verá en el circuito resultante puede resultar más sencillo utilizar un biestable para controlar el encendido o apagado de cada barra. De esta manera utilizaríamos 4 biestables. Aunque existen 16 posibles combinaciones, en realidad sólo vamos a utilizar 8. El resto no se presentará nunca, por lo que pueden utilizarse como condiciones de “no importa” para simplificar las funciones resultantes. La codificación de los estados será la siguiente:

Secuencia específica de cuenta	Codificación y activación de luces
0	0000
1	0001
3	0011
7	0111
15	1111
13	1101
12	1100
4	0100

Construcción de la tabla de estados:

Cuenta	$Q_3(t)$	$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	$Q_3(t+1)$	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$	T_3	T_2	T_1	T_0
0	0	0	0	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	0	1	1	0	0	1	0
	0	0	1	0	X	X	X	X	X	X	X	X
3	0	0	1	1	0	1	1	1	0	1	0	0
4	0	1	0	0	0	0	0	0	0	1	0	0
	0	1	0	1	X	X	X	X	X	X	X	X
	0	1	1	0	X	X	X	X	X	X	X	X
7	0	1	1	1	1	1	1	1	1	0	0	0
	1	0	0	0	X	X	X	X	X	X	X	X
	1	0	0	1	X	X	X	X	X	X	X	X
	1	0	1	0	X	X	X	X	X	X	X	X
	1	0	1	1	X	X	X	X	X	X	X	X
12	1	1	0	0	0	1	0	0	1	0	0	0
13	1	1	0	1	1	1	0	0	0	0	0	1
	1	1	1	0	X	X	X	X	X	X	X	X
15	1	1	1	1	1	1	0	1	0	0	1	0

Las expresiones de las entradas a los biestables son (si se considera $X=0$):

$$T_3 = m_7 + m_{12}$$

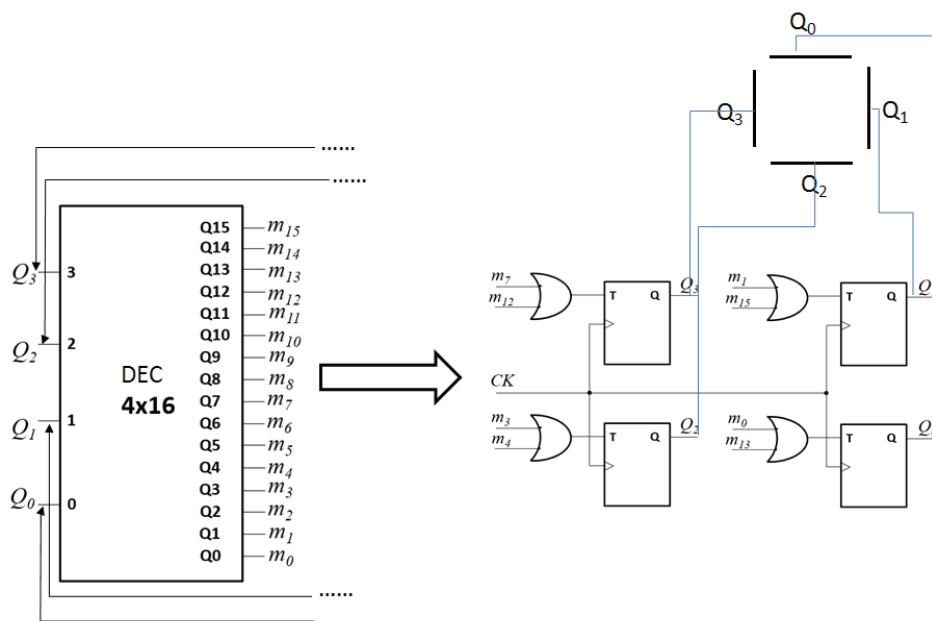
$$T_2 = m_3 + m_4$$

$$T_1 = m_1 + m_{15}$$

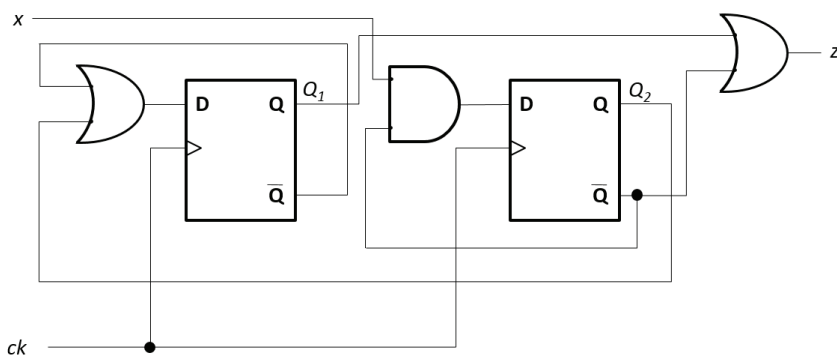
$$T_0 = m_0 + m_{13}$$

Puesto que el enunciado pide utilizar puertas OR y decodificadores, lo que haremos será utilizar un decodificador 4x16, al que conectaremos las salidas de los biestables (Q_3 , Q_2 , Q_1 , Q_0), de manera que para la implementación de T_3 , por ejemplo, sólo tendremos que sumar las salidas 7 y 12 del decodificador.

La función de salida, como se ha comentado, consiste simplemente en conectar cada salida de los biestables a la señal de activación de cada una de las 4 barras luminosas, tal y como se muestra en la figura siguiente.



- 32) Analiza el circuito de la figura adjunta, escribiendo las ecuaciones booleanas de todos los puntos del circuito, así como los cambios de estados. Escribe la tabla de verdad que describe el análisis del circuito. Calcula la evolución de la salida en al menos 5 ciclos de reloj si la entrada es $x = 0$ siempre y el estado inicial está definido con los dos biestables con salida a 1.



Las expresiones de las entradas a los biestables y salida son:

$$D_1 = \overline{Q_1} + Q_2$$

$$D_2 = x \cdot \overline{Q_2}$$

$$z = Q_1 + \overline{Q_2}$$

Ya que $Q_1(t+1) = D_1$ y $Q_2(t+1) = D_2$ la tabla de transición de estados es

x	$Q_1(t)$	$Q_2(t)$	D_1	D_2	$Q_1(t+1)$	$Q_2(t+1)$	Z
0	0	0	1	0	1	0	1
0	0	1	1	0	1	0	0
0	1	0	0	0	0	0	1
0	1	1	1	0	1	0	1
1	0	0	1	1	1	1	1
1	0	1	1	0	1	0	0
1	1	0	0	1	0	1	1
1	1	1	1	0	1	0	1

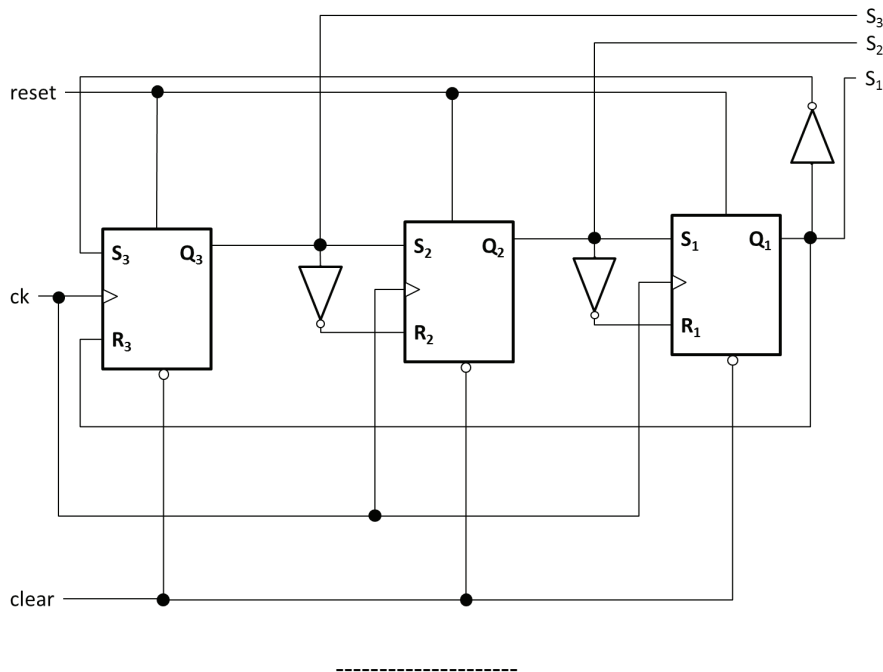
La evolución de la salida z para los cinco primeros con las condiciones de inicio indicadas puede ser obtenida a través de la siguiente tabla:

Reloj	Inicio	1º ciclo	2º ciclo	3º ciclo	4º ciclo	5º ciclo
x	0	0	0	0	0	0
Q_1	1	1	0	1	0	1
Q_2	1	0	0	0	0	0
D_1	1	0	1	0	1	0
D_2	0	0	0	0	0	0
z	1	1	1	1	1	1

Como se ve la salida permanece constante a 1: $1 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1$

- 33) Analiza el circuito de la figura adjunta y escribe las ecuaciones booleanas de todos los puntos del circuito, así como los cambios de estado. Escribe la tabla de verdad que describe el análisis del circuito. Calcula la evolución de la salida en al menos 6 ciclos de reloj.**

NOTA: El valor inicial de las salidas de los biestables RS es 000.



Distinguimos en el circuito entre s (salida) y S (entrada a biestables).

Las expresiones de las entradas a los biestables y salidas son:

$$\begin{aligned} R_1 &= \overline{Q_2} & R_2 &= \overline{Q_3} & R_3 &= Q_1 \\ S_1 &= Q_2 & S_2 &= Q_3 & S_3 &= \overline{Q_1} \end{aligned}$$

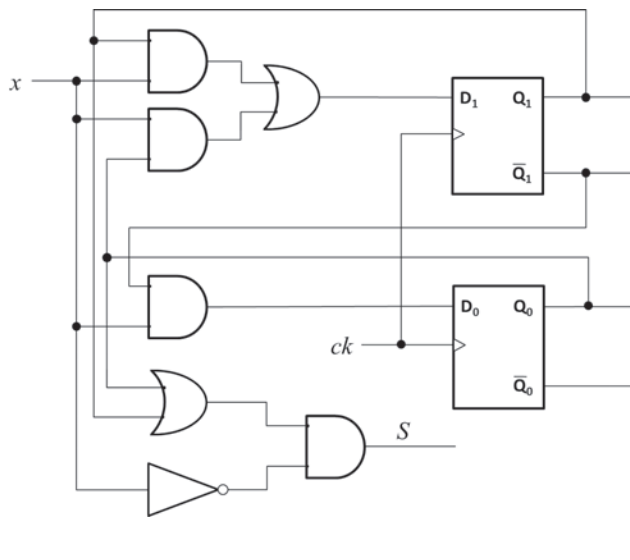
Teniendo en cuenta las tablas de transición del biestable RS, la tabla de transición de estados es:

Q_3	Q_2	Q_1	R_3	S_3	R_2	S_2	R_1	S_1	$Q_3(t+1)$	$Q_2(t+1)$	$Q_1(t+1)$
0	0	0	0	1	1	0	1	0	1	0	0
0	0	1	1	0	1	0	1	0	0	0	0
0	1	0	0	1	1	0	0	1	1	0	1
0	1	1	1	0	1	0	0	1	0	0	1
1	0	0	0	1	0	1	1	0	1	1	0
1	0	1	1	0	0	1	1	0	0	1	0
1	1	0	0	1	0	1	0	1	1	1	1
1	1	1	1	0	0	1	0	1	0	1	1

La evolución de la salida $Q_3Q_2Q_1$ para los seis primeros con las condiciones de inicio de todos los biestables a cero es: $0 \rightarrow 4 \rightarrow 6 \rightarrow 7 \rightarrow 3 \rightarrow 1 \rightarrow 0$

- 34) Hallar el diagrama de estados del circuito de la siguiente figura: Si x es constante y vale 1 durante los primeros 5 ciclos de reloj, ¿cuál sería la secuencia de salida (valor de la señal S) en los primeros 5 ciclos de reloj?

NOTA: Los biestables D se encuentran inicialmente a 1.



Las expresiones de entrada a los biestables y la salida S son:

$$D_1 = x \cdot Q_1 + x \cdot Q_0 = x \cdot (Q_1 + Q_0)$$

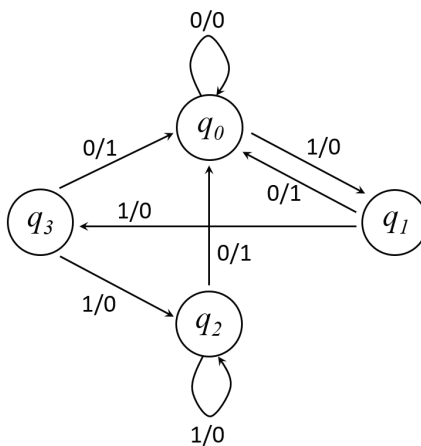
$$D_0 = \overline{Q_1} \cdot x$$

$$S = (Q_0 + Q_1) \cdot \overline{x}$$

La tabla de excitación del circuito es:

x	$Q_1(t)$	$Q_0(t)$	D_1	D_0	$Q_1(t+1)$	$Q_0(t+1)$	S
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1
0	1	0	0	0	0	0	1
0	1	1	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	1	1	1	1	1	0
1	1	0	1	0	1	0	0
1	1	1	1	0	1	0	0

Con lo que el diagrama de estado resulta ser:

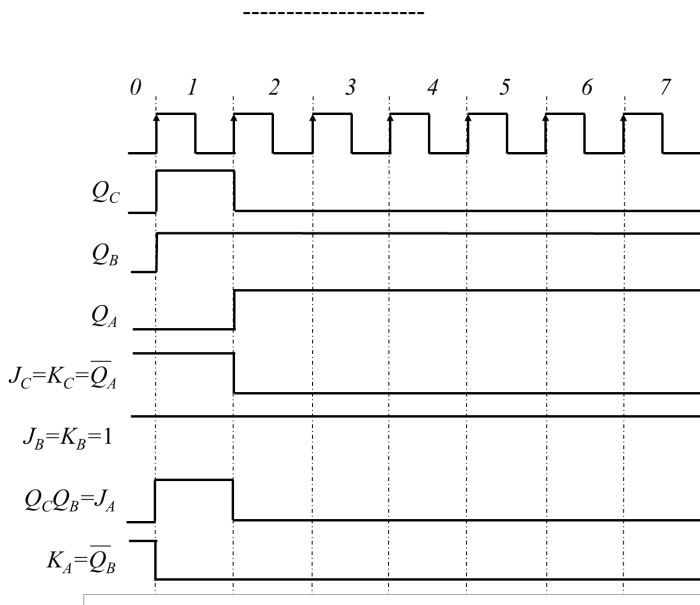


Considerando el estado inicial $Q_1=1, Q_0=1$ y la entrada fija $x=1$, la secuencia de estados y salida en los 5 primeros ciclos es (estado/salida):

$q_3 \rightarrow q_2/0 \rightarrow q_2/0 \rightarrow q_2/0 \rightarrow q_2/0$

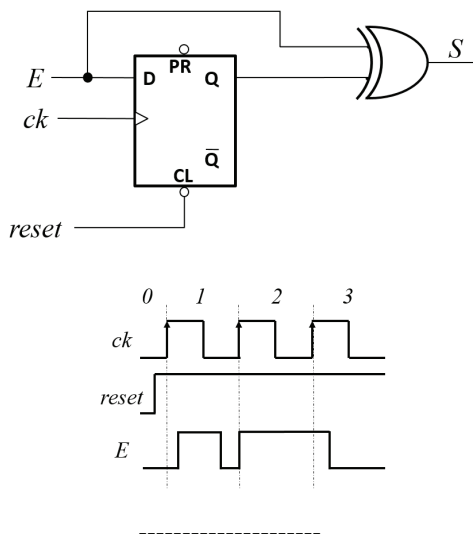
Luego las salidas durante los 5 primeros ciclos de reloj son cinco ceros.

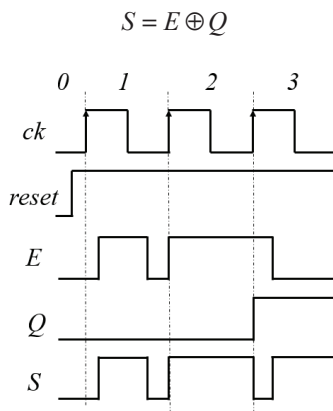
35) El circuito de la figura adjunta representa un biestable implementado a partir de otro tipo D (latch). Calcula la tabla de transición e indica el tipo de biestable implementado.



Los biestables C y A son síncronos con el reloj, B no.

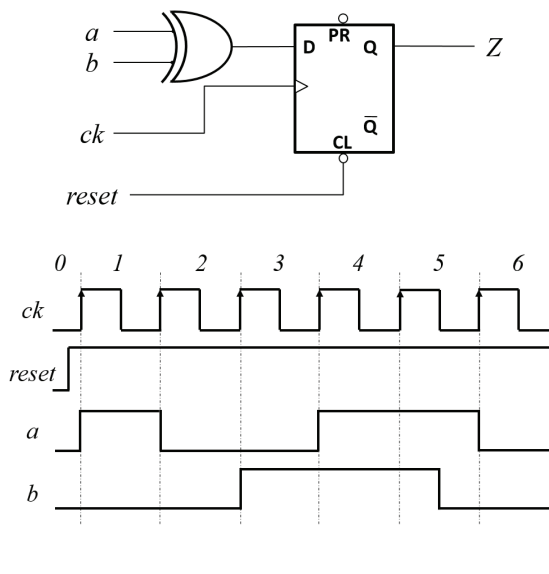
37) Dado el circuito secuencial de la figura, en el que el biestable es síncrono por flanco de subida, completa la señal de salida en el cronograma.



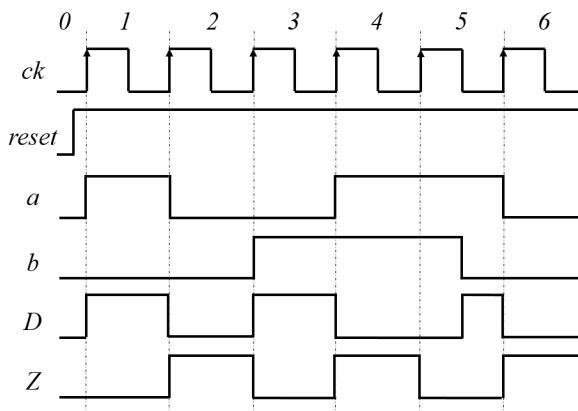


- 38) Dado el circuito secuencial de la siguiente figura, completa el cronograma correspondiente. Ten en cuenta que la señal D se refiere a la salida combinacional de la puerta XOR, mientras que Z es la salida secuencial de biestable D, síncrono por flanco de subida. Suponer que inicialmente $Q=0$.

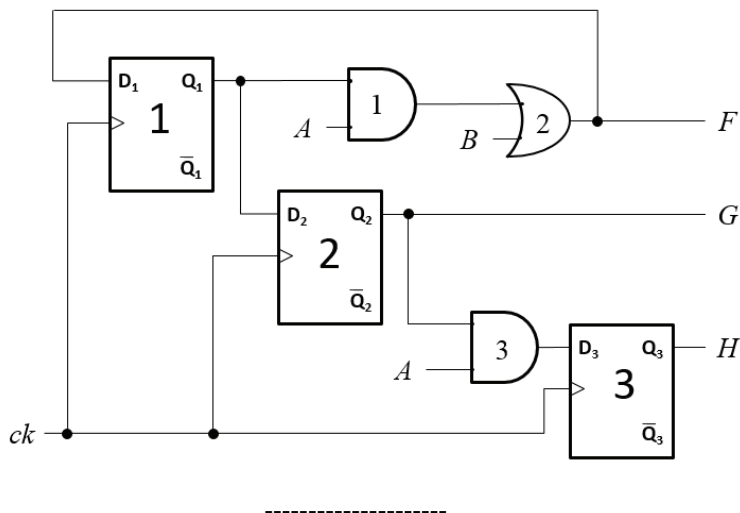
NOTA: te recomendamos que en primer lugar completes en el cronograma los valores de la salida combinacional, y posteriormente utilices ésta para determinar los valores de la salida Z .



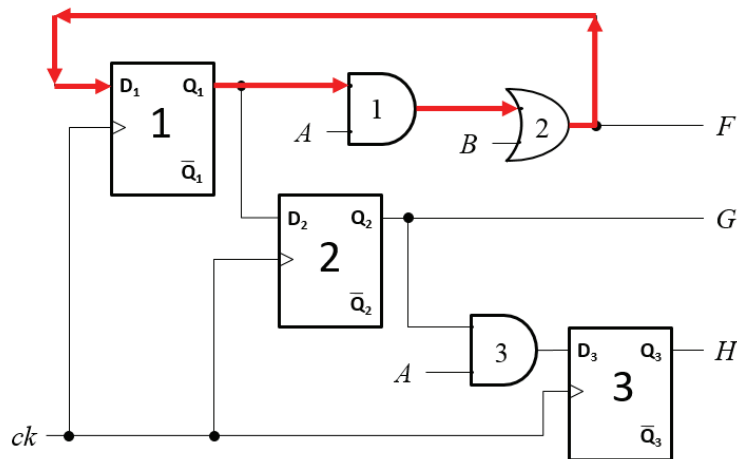
Teniendo en cuenta que $D = a \oplus b$, el cronograma es el siguiente:



- 39) Considerando que el tiempo de propagación de cada biestable es igual a 6 ns, su tiempo de set-up es de 5 ns y el tiempo de propagación de cada puerta es igual a 2 ns, determina la máxima frecuencia a la que está asegurado el funcionamiento de este circuito. Supón que la anchura del pulso no impone ninguna limitación (siempre es suficiente).



Según el diagrama del circuito, el camino más largo realimentado sería el que pasa por el biestable 1 y las puertas 1 y 2 (señalado con flechas).



$$f_{\max} \leq \frac{1}{T_{\min}}$$

$$T_{\min} \geq t_{\text{prop-biastable}} + 2 \cdot t_{\text{prop_puerta}} + t_{\text{setup}} = 6\text{ns} + 4\text{ns} + 5\text{ns} = 15\text{ns}$$

$$f_{\max} = \frac{1}{15 \cdot 10^{-9}} = 66,6 \text{ MHz}$$

40) La máxima frecuencia de funcionamiento de un sistema es de 41,6 MHz. Si el camino más largo entre dos biestables está compuesto por 3 puertas lógicas y los biestables tienen un tiempo de *setup* de 4 ns. ¿Cuál es el tiempo de propagación de cada puerta lógica? *NOTA: las puertas lógicas y el biestable tienen el mismo tiempo de propagación.*

Justificación: 41,6 MHz corresponden a un periodo de 24 ns. Este tiempo entre ciclos debe ser igual a $t_{\text{prop-biastable}} + 3 \cdot t_{\text{prop-puerta}} + t_{\text{setup}}$

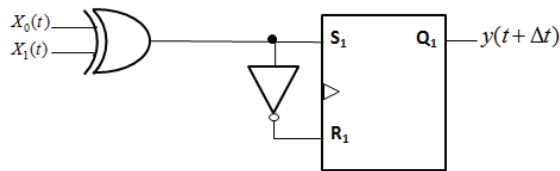
Dado que $t_{\text{prop-biastable}} = t_{\text{prop-puerta}}$, nos queda que $4 \cdot t_{\text{prop-puerta}} + 4 = 24$

Por tanto, $t_{\text{prop-puerta}} = 5 \text{ ns}$.

TEST Y PREGUNTAS CORTAS

- 1) Dada las entradas $x_0=111111.....$ y $x_1=011001...$ e inicializado el biestable a 0, la secuencia de salida será:

- ☐ A $y=0/000000...$
☐ B $y=0/100110...$
☐ C $y=0/011001...$

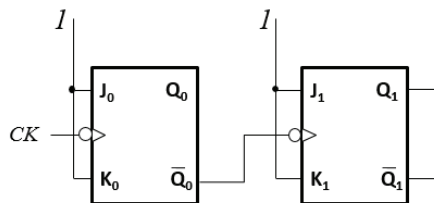


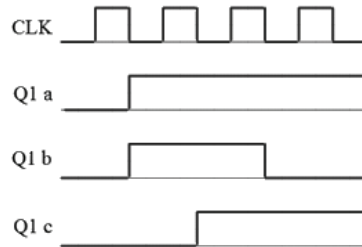
- 2) Indica la respuesta incorrecta. En un diagrama de estados de un sistema secuencial:

- ☐ A El número de estados depende del número de salidas.
☐ B Si se representa un autómata de Moore las salidas solo dependen del estado.
☐ C El número de flechas que sale de un estado depende del número de entradas.

- 3) El contador de la figura, ¿es síncrono o asíncrono? ¿Cuál es la forma de onda de la señal $Q1$, asumiendo que se parte de $Q0=0$, $Q1=0$?

- ☐ A Síncrono, a
☐ B Asíncrono, b
☐ C Asíncrono, c





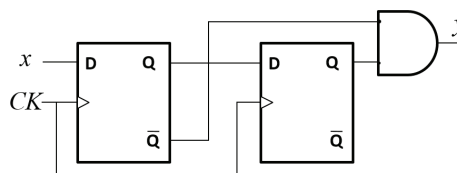
4) Señala cuál de las siguientes afirmaciones es verdadera:

- ☐ A En un autómata de Mealy, cada estado tiene asociada una salida.
- ☐ B En un autómata de Moore la salida depende exclusivamente del estado.
- ☐ C Las dos anteriores son ciertas.

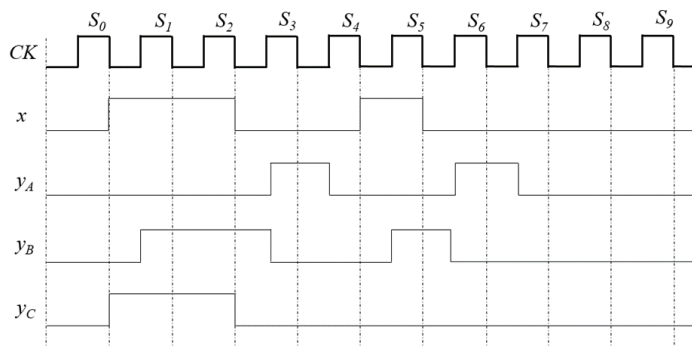
5) Indica la respuesta incorrecta.

- ☐ A Si a un biestable RS se le introduce una entrada $RS=00$, mantiene el estado.
- ☐ B Si a un biestable RS se le introduce una entrada $RS=11$, cambia de estado.
- ☐ C La función de transición de estados de un biestable JK es $Q(t+1) = JQ'(t) + K'Q(t)$

6) En el circuito de la figura adjunta los biestables se inicializan a cero. Señala la onda correcta de salida:

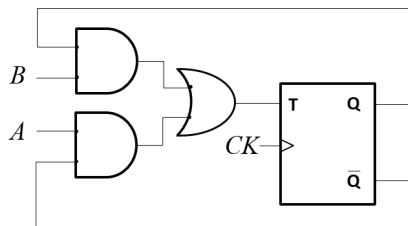


- ☐ A y_A
- ☐ B y_B
- ☐ C y_C



- 7) Suponer que el circuito adjunto la situación inicial es $Q=0$ y que A y B permanecen siempre a 1. El contenido del biestable Q durante los primeros ciclos de reloj será:

- ☐ A $Q=00000000$
☐ B $Q=01010101$
☐ C $Q=00110011$



- 8) Señala cuál de las afirmaciones siguientes es la falsa:

- ☐ A Un biestable maestro-esclavo puede cambiar hasta dos veces en el mismo ciclo de reloj.
☐ B El tiempo de mantenimiento (t_{hold}) es el tiempo mínimo que el dato debe estar estable después del flanco activo de reloj.
☐ C En el sincronismo por nivel, cuando el reloj se encuentra en el nivel activo, las salidas responden a todas las variaciones de las entradas.

- 9) Suponer que en un sistema secuencial el tiempo de nivel alto de reloj son 3 ns y el tiempo de retardo del biestable es de 9ns. Señala la opción correcta.

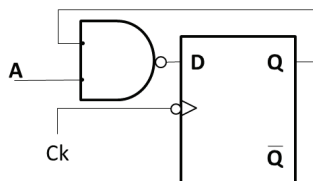
- ☐ A Se produce el efecto de “glitche”.
☐ B Nunca puede pasar esa circunstancia.
☐ C Se produce el efecto de “carrera”.

- 10) Señala la opción verdadera

- ☐ A La salida de un sistema secuencial no depende de las entradas.
☐ B El tiempo de mantenimiento de un biestable es el tiempo mínimo que el dato debe estar estable después del flanco activo del reloj.
☐ C Un biestable de tipo D se consigue a partir de un JK haciendo $J=K$.

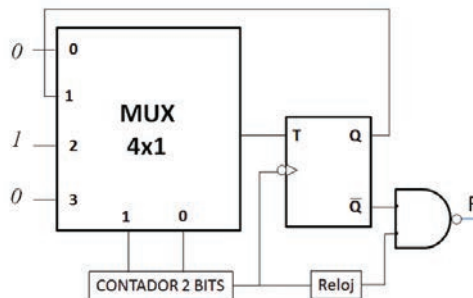
- 11) Determina la salida Q , suponiendo que A se mantiene a 1 e inicialmente $Q=1$.

- ☐ A 1001001
☐ B 1000000
☐ C 1010101



12) En la figura, el biestable está inicialmente a 1. Señala la respuesta correcta:

- ☐ A F está a 1 sólo durante las dos primeras cuentas
☐ B F está a 1 siempre, excepto medio ciclo de reloj durante el ciclo de cuenta número 2
☐ C F se pone a cero cuando el reloj esté a 1



13) Marcar la afirmación correcta sobre biestables:

- ☐ A En un biestable de tipo JK, si se unen J y K se obtiene un biestable de tipo T.
☐ B En un biestable de tipo JK, si se unen J y K con un inversor, se obtiene un biestable de tipo D entrando por la entrada K .
☐ C La salida de un biestable SR para el caso $S=1$ y $R=1$ pone a cero el biestable.

14) El tiempo de establecimiento (t_{setup}) en un biestable es:

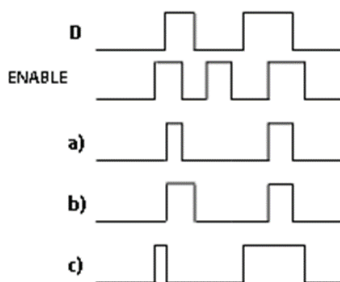
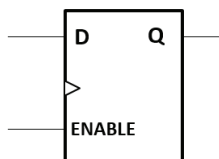
- ☐ A El tiempo de propagación del biestable.
☐ B El tiempo mínimo que el dato debe estar estable antes del flanco activo del reloj.
☐ C El tiempo mínimo que el dato debe estar estable después del flanco activo del reloj.

15) Referente a la tabla de excitación de un biestable, señala la opción correcta:

- ☐ A Sirve para saber cuál es el próximo estado conocido el estado actual y las entradas (actuales).
☐ B Sirve para saber qué valores hay que poner en sus entradas para que se produzcan las transiciones deseadas.
☐ C Se utiliza cuando queremos analizar un circuito secuencial.

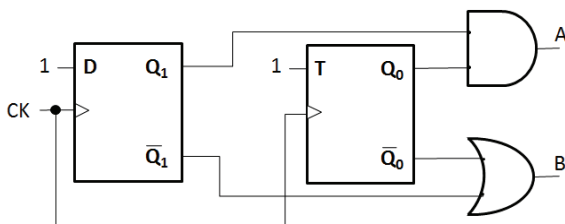
16) En la figura adjunta se representa un biestable D. Para las entradas D y $ENABLE$ del cronograma, ¿cuál es la salida Q correcta? Suponer que inicialmente $Q = 0$. Observa que es un sistema asíncrono.

- ☐ A Figura a
☐ B Figura b
☐ C Figura c



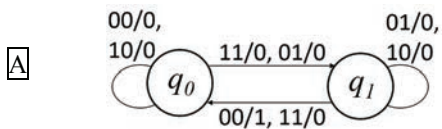
17) Dado el circuito de la figura adjunta, donde inicialmente ambos biestables están a cero, las salidas para los primeros ciclos, contando el instante inicial, son:

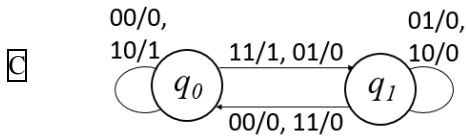
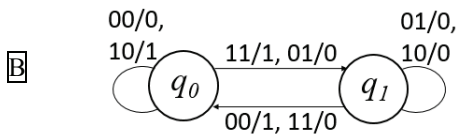
- ☐ A $A=010101, B=101010$
☐ B $A=000000, B=111111$
☐ C $A=010101, B=111111$



18) Dada la tabla de verdad adjunta, indicar cuál es el diagrama de estados correcto.

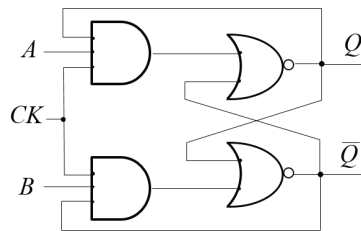
A	B	Q(t)	Q(t+1)	S
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0



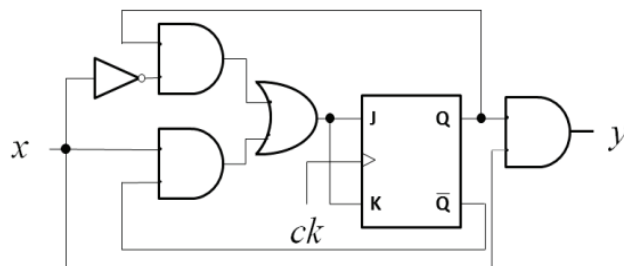


19) Considera el circuito de la figura adjunta, en el que inicialmente $Q=0$. En el instante siguiente se dan entradas $A=0$ y $B=1$. Señala la respuesta correcta:

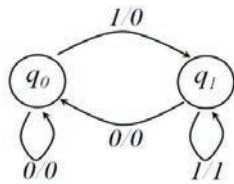
- ☐ A Se trata de un biestable JK con $K=B$ y $J=A$, por lo que Q continúa a 0.
☐ B No es un biestable JK y Q pasará a valer 0.
☐ C Se trata de un biestable JK con $K=A$ y $J=B$, por lo que Q pasará a valer 1.



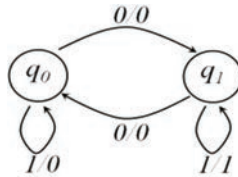
20) Para el circuito de la figura señala el diagrama de estado correcto.



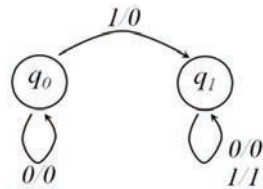
A



B

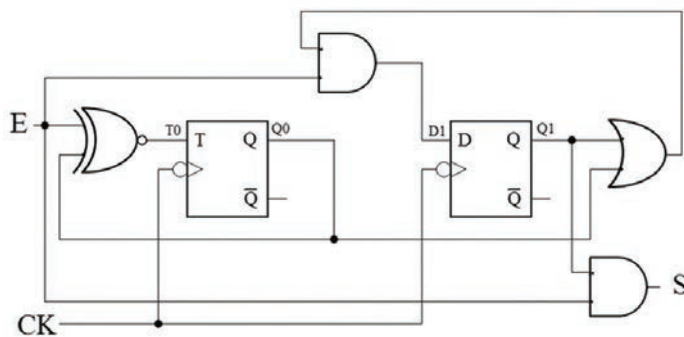


C



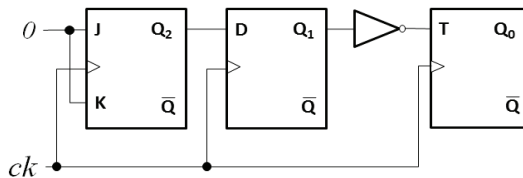
21) Sabiendo que $T_{prop NOT} = 2 \text{ ns}$, $T_{prop AND/OR} = 5 \text{ ns}$, $T_{prop XOR} = 7 \text{ ns}$, $T_{prop D/T} = 5 \text{ ns}$, $T_{setup D/T} = 3 \text{ ns}$, ¿Cuál será la frecuencia máxima de funcionamiento del circuito de la figura adjunta?

- A 76.9 MHz
- B 50 MHz
- C 55.5 MHz

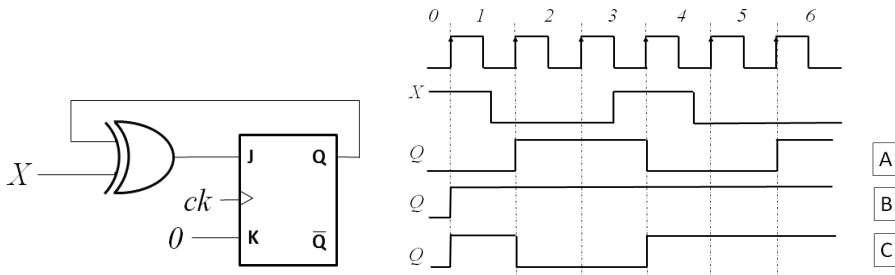


22) El circuito de la figura se inicializa con $Q_2=0$, $Q_1=0$, $Q_0=0$. El número representado en binario por $Q_2Q_1Q_0$ de salida será: (Q_2 es el bit más significativo).

- ☐ A 0, 4, 2, 1, 0, 4, 2, ...
☐ B 0, 1, 0, 1, 0, 1, 0, ...
☐ C 0, 0, 0, 0, 0, 0, 0, 0

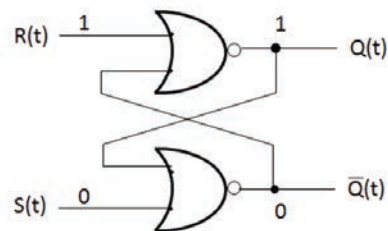


23) Señala el cronograma correcto para Q . Observa que Q se inicializa a 0.



24) Ante una situación como la de la figura, elige lo correcto

- ☐ A $Q(t+1)=1$
☐ B $Q(t+1)=0$
☐ C Es una situación prohibida.



25) Si una señal Preset de un biestable es activa a nivel bajo:

- ☐ A Debemos mantener la señal a 1 excepto cuando queramos inicializar el biestable a 1, en cuyo caso generaremos un pulso 101.
☐ B Debemos mantener la señal a 0 excepto cuando queramos inicializar el biestable a 1, en cuyo caso generaremos un pulso 010.
☐ C Debemos mantener la señal a 1 cuando queramos inicializar el biestable a 0.

26) Las ecuaciones características de los FFs de tipos T y D son:

- ☐ A $Q(t+1) = T$ y $Q(t+1) = D \oplus Q(t)$
- ☐ B $Q(t+1) = TQ(t)$ y $Q(t+1) = D$
- ☐ C $Q(t+1) = T \oplus Q(t)$ y $Q(t+1) = D$

27) Señala la opción falsa:

- ☐ A El biestable SR no admite entradas $S=1$, $R=1$.
- ☐ B Un biestable disparado por nivel puede cambiar su estado cuando el reloj pasa de 0 a 1 o de 1 a 0.
- ☐ C El Hold time (t_h) es el tiempo que tiene que mantenerse la entrada al biestable después de ocurrido el flanco de reloj.

28) Marcar la afirmación correcta sobre un biestable de tipo JK:

- ☐ A Está construido con puertas lógicas y, por lo tanto, es un circuito combinacional.
- ☐ B Nunca se pueden unir las entradas J y K porque produciría cortocircuito.
- ☐ C Si se unen J y K se obtiene un biestable de tipo T.

29) Marcar la afirmación correcta:

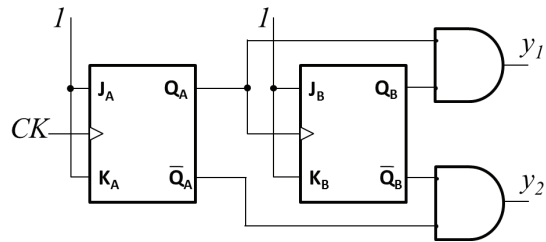
- ☐ A La salida de un contador asíncrono depende de las entradas.
- ☐ B Es posible diseñar un contador en anillo utilizando biestables de tipo D o biestables de tipo SR, sin ningún tipo de puertas.
- ☐ C Un registro de desplazamiento paralelo-serie está diseñado con biestables tipo T.

30) El tiempo de mantenimiento (t_{hold}) en un biestable es:

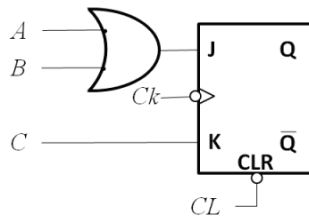
- ☐ A El tiempo de propagación del biestable.
- ☐ B El tiempo mínimo que el dato debe estar estable después del flanco activo del reloj.
- ☐ C El tiempo de subida en un flanco activo del biestable.

31) Dado el circuito de la figura adjunta, donde inicialmente ambos biestables están a cero, las salidas para los primeros ciclos, contando el instante inicial, son:

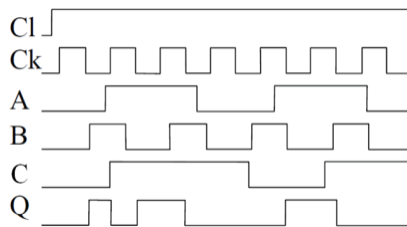
- ☐ A $y_1=000000, y_2=111111$
☐ B $y_1=010001, y_2=100010$
☐ C $y_1=000100, y_2=100010$



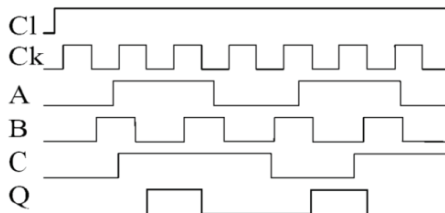
32) Dado el circuito de la figura, marcar el cronograma correspondiente.



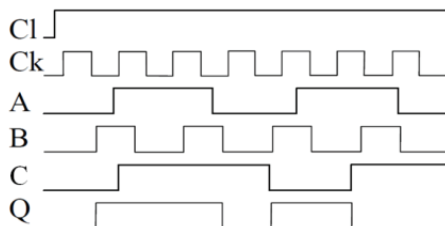
☐ A



☐ B

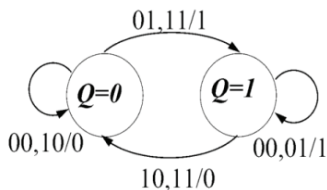


☐ C

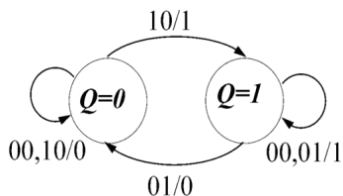


33) ¿Cuál de estos diagramas de estado corresponde a un biestable JK?

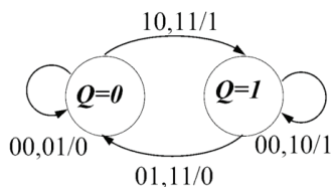
☐ A



☐ B



☐ C



34) Señala cuál de las siguientes afirmaciones es verdadera:

- ☐ A En un autómata de Mealy, cada estado tiene asociada una salida.
- ☐ B Dado un autómata de Mealy, siempre es posible encontrar su autómata de Moore equivalente.
- ☐ C Los autómatas de Moore y Mealy equivalentes tienen siempre el mismo número de estados.

35) Se pretende convertir un biestable C en un biestable AB. ¿Qué se debe hacer? El biestable AB funciona de la siguiente forma: $AB = 00$ cambia de estado, $AB = 01$ fuerza a 0, $AB = 10$ fuerza a 1 y $AB = 11$ mantiene estado. ¿El biestable C cambia el estado cuando $C = 0$ y mantiene el estado cuando $C = 1$?

- ☐ A Introducir en C la función $A\overline{Q}(t) + \overline{B}Q(t)$
- ☐ B Introducir en C la función $AQ(t) + B\overline{Q}(t)$
- ☐ C Introducir en C la función $AB + B\overline{Q}(t)$

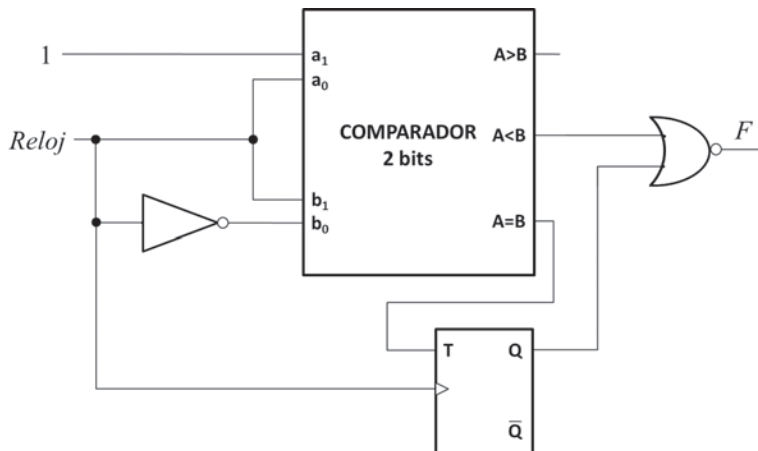
36) ¿Cuál de las siguientes expresiones permite convertir un biestable tipo D a otro tipo JK?

☐ A $D = JQ + KQ$

☐ B $D = J\bar{Q} + KQ$

☐ C $D = J\bar{Q} + \bar{K}Q$

37) Dado el circuito de la figura, formado por un comparador de 2 bits y un biestable T, ¿cuál es la salida que corresponde a F ? Suponer que inicialmente $Q = 0$ y $Reloj = 0$.

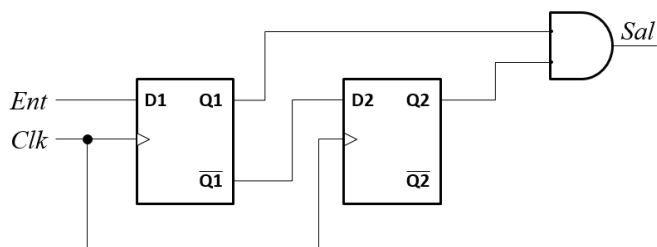


☐ A Reloj invertido.

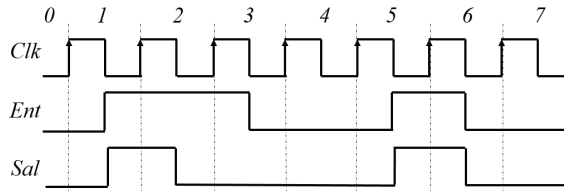
☐ B Siempre a 1.

☐ C Ninguna de las anteriores es correcta.

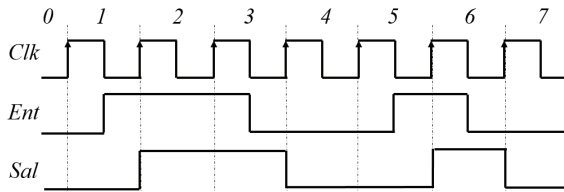
38) ¿Qué diagrama corresponde con el funcionamiento de este circuito lógico? Suponer que inicialmente $Q1=0$ y $Q2=0$. Los biestables son activos en flancos de subida.



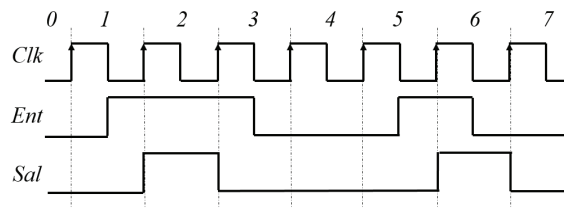
A



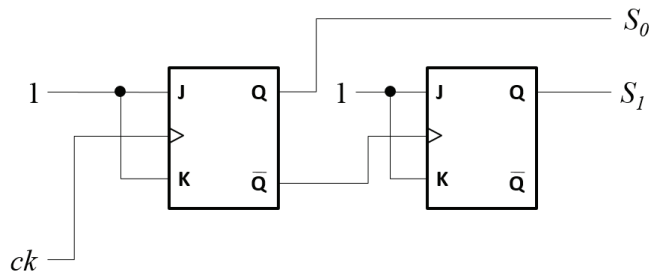
B



C



39) ¿Qué tipo de circuito implementa la figura?



- A Un contador síncrono ascendente de 2 bits
- B Un contador asíncrono ascendente de 2 bits
- C Un contador asíncrono descendente de 2 bits

SISTEMAS SECUENCIALES: MÓDULOS

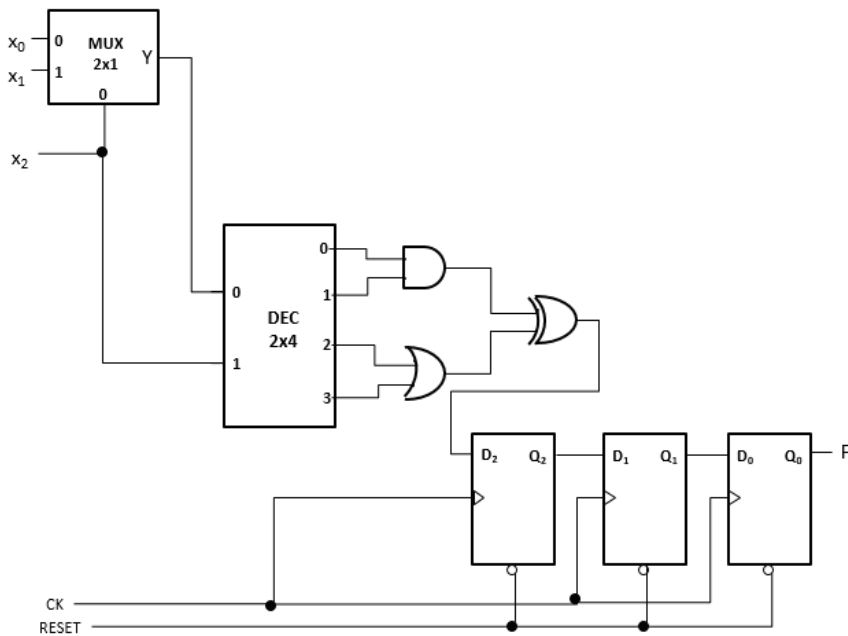
CONTENIDOS:

- Circuitos de contadores y registros.
- Circuitos secuenciales complejos.

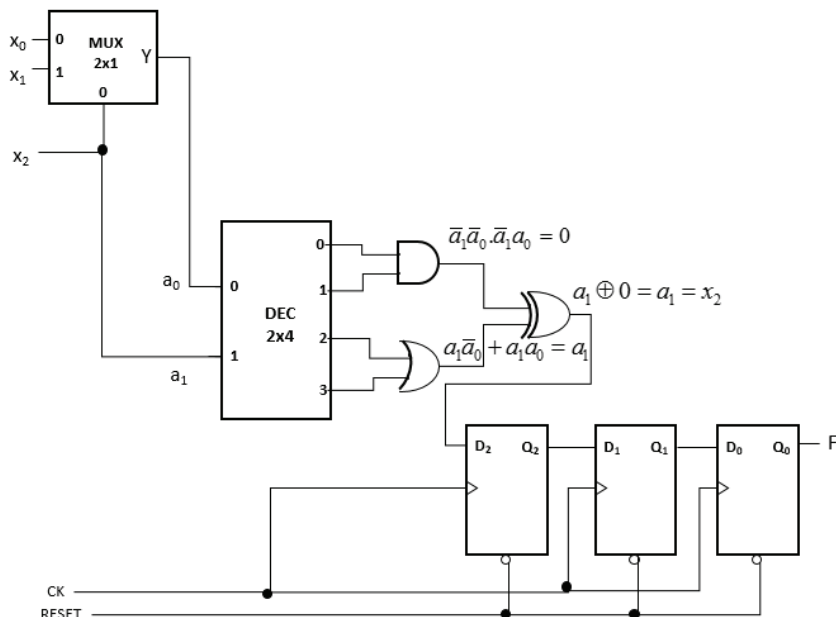
EJERCICIOS RESUELTOS

- 1) Dado el circuito de la figura, ¿cuál será el valor almacenado en el registro después del 3º ciclo de reloj, dada la siguiente secuencia de entradas en los valores x_2 , x_1 e x_0 ? Suponer que inicialmente $Q_2 = Q_1 = Q_0 = 0$.

Inicialmente: $x_0 \ x_1 \ x_2 = 0 \ 1 \ 0$
 Después del 1º ciclo de reloj: $x_2 \ x_1 \ x_0 = 1 \ 0 \ 1$
 Después del 2º ciclo de reloj: $x_2 \ x_1 \ x_0 = 1 \ 0 \ 0$
 Después del 3º ciclo de reloj: $x_2 \ x_1 \ x_0 = 1 \ 1 \ 1$



Realizando un análisis del circuito, exploramos la salida de las puertas que hay detrás del decodificador, que resulta ser la entrada del primer biestable.

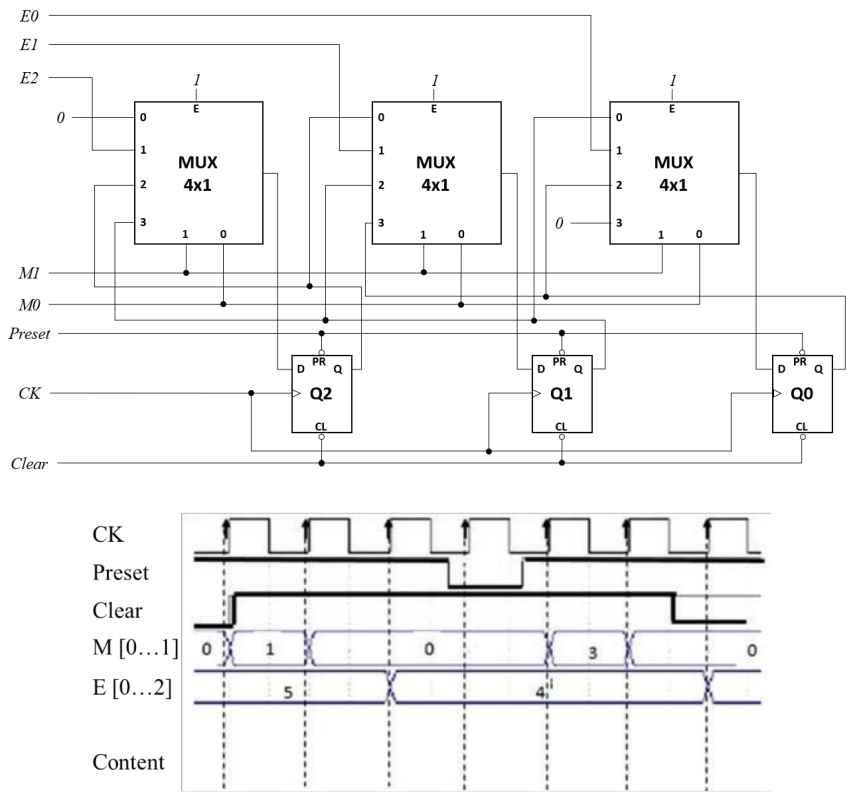


Simplificando el circuito detrás del DEC se llega a que $D2(t) = x2(t)$. Como $Q2(t+1) = D2(t)$ se tiene que $Q2(t+1) = x2(t)$.

Para los demás biestables es un registro de desplazamiento y se cumple $Q1(t+1) = Q2(t)$ y $Q0(t+1) = Q1(t)$

Clock cycle	x2	x1	x0	D2	Q2	Q1	Q0
0	0	1	0	0	0	0	0
1	1	0	1	1	0	0	0
2	1	0	0	1	1	0	0
3	1	1	1	1	1	1	0

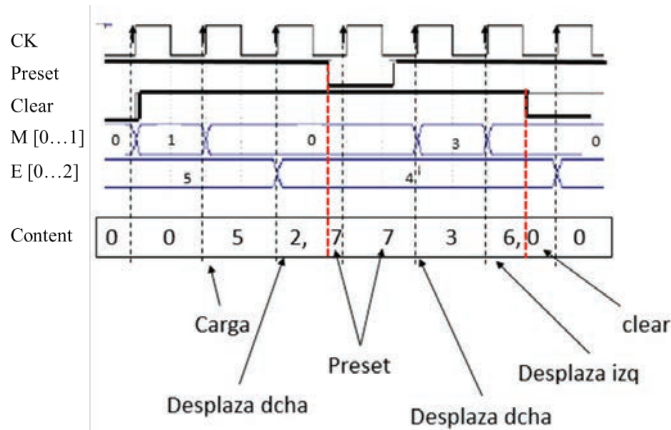
- 2) Dado el circuito de la figura adjunta, dibujar la señal “Content” (contenido $Q2Q1Q0$) especificando su valor decimal de modo similar a como se hace con las variables de entrada $M1M0$ y $E2E1E0$.



Se trata de un registro universal de desplazamiento. Teniendo en cuenta que el funcionamiento depende de las entradas de control según la tabla:

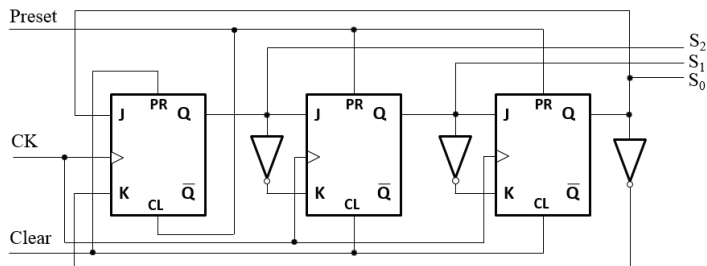
M1	M0	Funcionamiento
0	0	Desplazamiento a la derecha
0	1	Carga de datos
1	0	No desplazamiento
1	1	Desplazamiento a la izquierda

La señal de Content es la siguiente:

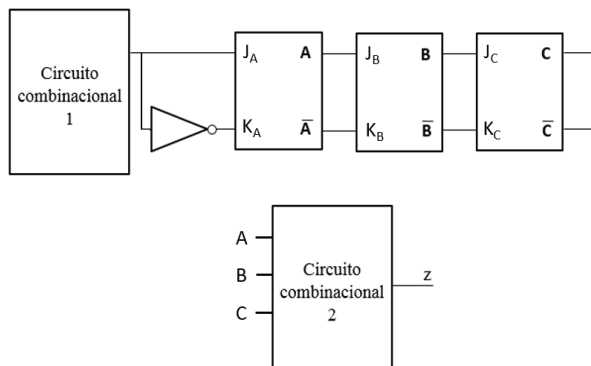


- 3) Haz un circuito que realice la cuenta cíclica de 5, 6, 3, 5, 6, 3, ...únicamente usando biestables JK. Suponer que el circuito es inicializado a 5.

El problema consiste en diseñar un contador en anillo inicializado a 101, por lo tanto, diseñamos un registro de desplazamiento con los biestables JK y las conexiones de realimentación entre el ultimo y primer biestable. El circuito es el siguiente:



- 4) Realizar el diseño de un circuito que genere la secuencia binaria 1-0-1-1-0-1-1-0, con un registro de desplazamiento.



Como hay 8 bits en la secuencia binaria, se necesitará un registro de desplazamiento de tres etapas. Se supone el estado inicial 000.

Debemos recorrer una secuencia de estados que, ciñéndonos a la condición de desplazamiento a la derecha y empezando en el estado 000, vuelva después de 8 transiciones al estado primero 000. Se puede comprobar que una secuencia de este tipo es la siguiente: 0-1-2-5-3-7-6-4-0... La función de salida z que se precisa en cada estado del registro es respectivamente

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

Todo esto lo podemos visualizar mejor en la tabla siguiente.

C	B	A	z
0	0	0	1
0	0	1	0
0	1	0	1
1	0	1	1
0	1	1	0
1	1	1	1
1	1	0	1
1	0	0	0
0	0	0	

El problema queda reducido a construir dos circuitos. El circuito (1) es un contador con la secuencia 0-1-2-5-3-7-6-4. El circuito (2) genera, tomando las salidas de los biestables, una configuración de salida 1-0-1-1-0-1-1-0.

Como se trata de un registro de desplazamiento, todas las entradas de los biestables son complementadas. Además, las entradas a los biestables B y C son conocidas (son las salidas de los biestables A y B , respectivamente). Tendremos que hallar las expresiones de J_A , exigiendo que sea complemento de la expresión K_A y de la salida z .

$$J_B = A$$

$$J_C = B$$

$$K_B = \bar{A}$$

$$K_C = \bar{B}$$

	C	B	A	A(t+1)	J _A	K _A	z
0→1	0	0	0	1	1	X	1
1→2	0	0	1	0	X	1	0
2→5	0	1	0	1	1	X	1
3→7	0	1	1	1	X	0	0
4→0	1	0	0	0	0	X	0
5→3	1	0	1	1	X	0	1
6→4	1	1	0	0	0	X	1
7→6	1	1	1	0	X	1	1

Tenemos los dos mapas de Karnaugh de J_A y K_A .

J_A

	B			
	0	1	0	1
C	0	-	-	0
	A			

K_A

	B			
	0	1	0	-
C	-	0	1	-
	A			

Para que J_A sea función complemento de K_A , las condiciones de no-importa se transforman en condiciones de sí-importa.

J_A

	B			
	1	0	1	1
C	0	1	0	0
	A			

z

	B			
	1	0	0	1
C	0	1	1	1
	A			

$$J_A = \bar{C}B + \bar{C}\bar{A} + C\bar{B}A$$

$$z = \bar{C}\bar{A} + CA + CB$$

$$K_A = \bar{J}_A$$

La implementación de estas tres funciones constituye los circuitos combinacionales 1 y 2.

- 5) Diseñar un sistema secuencial con una entrada y una salida binaria tal que su salida $z(t)$ valga 1 sólo cuando las cuatro últimas entradas hasta t coincidan con alguno de los siguientes patrones: 1011 ó 0110. Utilizar biestables JK.

Se implementarán por separado dos reconocedores de patrones: uno para 1011 y otro para 0110, y las salidas se llevarán a una puerta OR, de modo que, si se reconoce alguno de los dos, la salida sea 1.

Patrón 1011

Definimos los estados;

- S_0 = no se está en ninguno de los siguientes estados
- S_1 = la última entrada fue 1.
- S_2 = las dos últimas entradas han sido 10.
- S_3 = las tres últimas entradas han sido 101.

La tabla de transición de estados y salida será:

Estado actual	Entradas (x)	
	0	1
S_0	$S_0, 0$	$S_1, 0$
S_1	$S_2, 0$	$S_1, 0$
S_2	$S_0, 0$	$S_3, 0$
S_3	$S_2, 0$	$S_1, 1$

E. siguiente, salida

Haciendo una codificación usual para las variables de estado, se llega a la siguiente tabla de excitación:

Entrada	Estado actual		Estado siguiente		Entradas a FF		Salida
x	Q_2	Q_1	Q_2	Q_1	J_1K_1	J_2K_2	z_1
0	0	0	0	0	0 -	0 -	0
0	0	1	1	0	1 -	- 1	0
0	1	0	0	0	- 1	0 -	0
0	1	1	1	0	- 0	- 1	0
1	0	0	0	1	0 -	1 -	0
1	0	1	0	1	0 -	- 0	0
1	1	0	1	1	- 0	1 -	0
1	1	1	0	1	- 1	- 0	1

J_1			Q_2	
	0	1	-	-
x	0	0	-	-
		Q_1		

$$J_1 = \bar{x} Q_1$$

K_1			Q_2	
	-	-	0	1
x	-	-	1	0
		Q_1		

$$K_1 = x \oplus Q_1$$

Directamente obtenemos:

$$J_2 = x$$

$$K_2 = \bar{x}$$

La salida es:

z_1			Q_2	
	0	0	0	0
x	0	0	1	0
		Q_1		

$$z_1 = x Q_1 Q_2$$

Patrón 0110

Se definen los siguientes estados:

- S_0 = no se está en ninguno de los siguientes estados.
- S_1 = la última entrada fue 0.
- S_2 = las dos últimas entradas han sido 01.
- S_3 = las tres últimas entradas han sido 011.

La tabla de transición de estados y salida es:

Estado actual	Entradas (x)	
	0	1
S_0	$S_{1, 0}$	$S_{0, 0}$
S_1	$S_{1, 0}$	$S_{2, 0}$
S_2	$S_{1, 0}$	$S_{3, 0}$
S_3	$S_{1, 1}$	$S_{0, 0}$

E. siguiente, salida

Haciendo una codificación usual para las variables de estado, se llega a la siguiente tabla de excitación:

Entrada	Estado actual		Estado siguiente		Entradas a FF		Salida
x	Q_2	Q_1	Q_2	Q_1	J_1K_1	J_2K_2	z_1
0	0	0	0	1	0 -	1 -	0
0	0	0	0	1	0 -	- 0	0
0	1	0	0	1	- 1	1 -	0
0	1	0	0	1	- 1	- 0	1
1	0	0	0	0	0 -	0 -	0
1	0	1	1	0	1 -	- 1	0
1	1	0	1	1	- 0	1 -	0
1	1	1	0	0	- 1	- 1	0

Veamos las expresiones de conmutación de entradas a biestables y salida.

J_1	Q_2			
	0	0	-	-
x	0	1	-	-
Q_1				

$$J_1 = xQ_1$$

K_1	Q_2			
	-	-	1	1
x	-	-	1	0
Q_1				

$$K_1 = Q_1 + \bar{x}$$

J_1		Q_2	
		1	0
x	1	1	0
	0	0	1
		Q_1	

$$J_2 = \bar{x} + Q_2$$

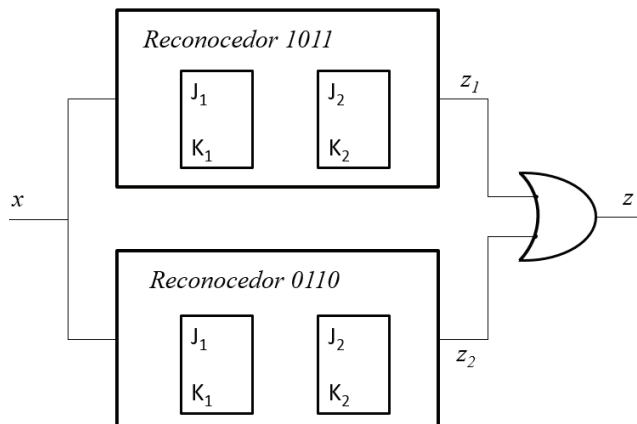
$$K_2 = x$$

La salida del segundo reconocedor es:

z_2		Q_2	
		1	0
x	1	0	0
	0	0	0
		Q_1	

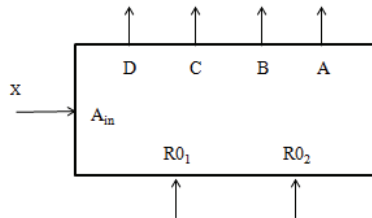
$$z_2 = \bar{x}Q_1Q_2$$

El diagrama en bloques de este doble reconocedor de patrones es:



6) Se dispone de contadores de década (7490), que cuentan del 0 al 9 y que se inicializa a 0. Su esquema es el de la figura.

- El terminal A_{in} recibe la señal de reloj x , activa por flanco de bajada.
- D, C, B, A son las salidas codificadas en BCD, siendo D la más significativa.
- $R0(1)$ y $R0(2)$ son dos terminales que deben recibir dos unos para que el contador se inicialice a 0000.

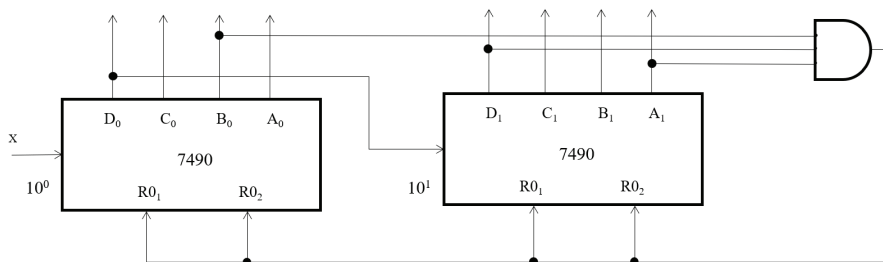


Diseñar con contadores 7490 un contador módulo-92.

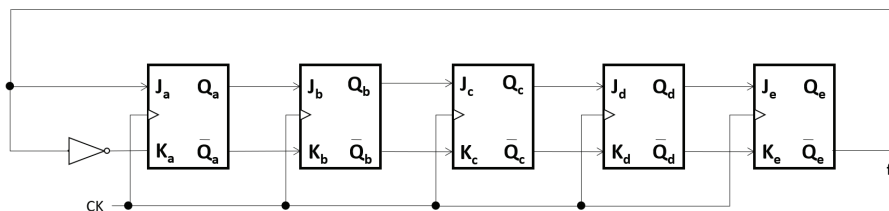
Se utilizarán dos contadores de década 7490 de tal manera, que el primero proporcione las unidades y el segundo las decenas (que son los contadores que aparecen a la izquierda y derecha en la figura adjunta). Ambos dígitos estarán codificados en BCD.

El contador de las unidades (10^0) realizará en el décimo pulso de reloj x , la transición 1001 a 0000. En esta transición debe ponerse en marcha el contador de las decenas (10^1). Para lograrlo, utilizamos el salto de 1 a 0 de D_0 como flanco de bajada del reloj del contador 10^1 , de modo que este pasará de 0000 a 0001. De esta manera, cada vez que se pase de 9 a 0 en el primer contador, existe un flanco de bajada de reloj para el segundo contador, y éste contará una decena más.

El único problema que queda por resolver, es inicializar los dos contadores a 0 cuando se llegue a 92 (0010 1001). Si ponemos una puerta AND con entradas B_0, D_1 Y A_1 , la única vez que la salida de la puerta tendrá valor 1 será cuando se llegue a 92. Consecuentemente, conectamos la salida de la puerta AND a todas las entradas $R0(1)$ y $R0(2)$.



- 7) Un contador Johnson es un contador de anillo invertido, donde la conexión de realimentación va de la salida complementada de la última etapa a la primera etapa, tal como muestra la figura.



Diseñar con este tipo de circuito un contador de diez estados, suponiendo que inicialmente todos los biestables se inicializan a 0.

Si queremos diseñar un contador de 10 estados, nos basta con utilizar un contador Johnson de 5 etapas. Según se ve en la tabla siguiente, en 10 pulsos de reloj se logran 10 configuraciones distintas en las 5 etapas. Cada una de ellas va a representar, en esa codificación particular, los dígitos del 0 al 9. La función f representa la realimentación, que, como ya sabemos, es $\bar{Q}_e$, y se supone que inicialmente el contador está a cero.

Se recuerda que se trata de un contador de desplazamiento.

Pulso de reloj	Q_e	Q_d	Q_c	Q_b	Q_a	f
0	0	0	0	0	0	1
1	0	0	0	0	1	1
2	0	0	0	1	1	1
3	0	0	1	1	1	1
4	0	1	1	1	1	1
5	1	1	1	1	1	0
6	1	1	1	1	0	0
7	1	1	1	0	0	0
8	1	1	0	0	0	0
9	1	0	0	0	0	0
10	0	0	0	0	0	1

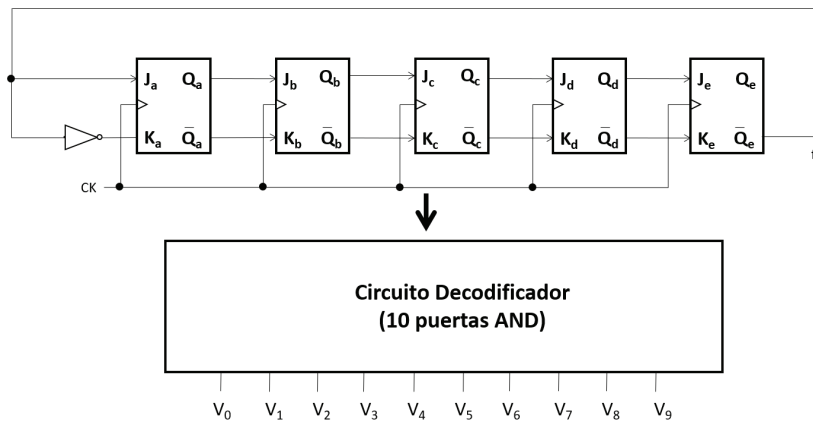
Se necesitará una lógica de decodificación para obtener un conteo decimal. Esta lógica se hará disponiendo de 10 funciones de salida $v_0, v_1, \dots, v_9$. Cuando la función v_i valga 1, se dirá que el estado del contador es i .

Para obtener la expresión de estas funciones, nos basta con estudiar con detenimiento la tabla. Se implementarán con puertas AND de dos entradas, siendo las dos entradas 1 solamente una vez, y haciendo la operación AND con dos variables consecutivas.

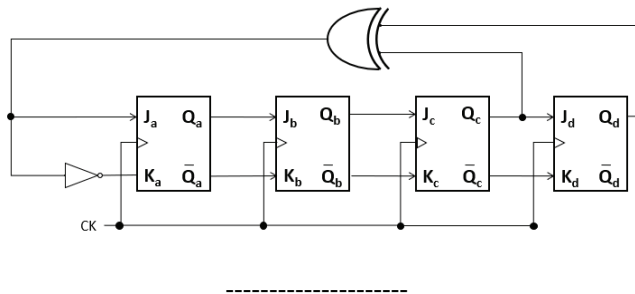
Por ejemplo, para implementar la función v_0 basta elegir como entradas A' y E' , ya que solo en una oportunidad coinciden ambas en ser 1. Para v_1 se escogen A y B' , y así sucesivamente.

$$\begin{aligned} v_0 &= \bar{Q}_a \bar{Q}_e & v_5 &= Q_a Q_e \\ v_1 &= Q_a \bar{Q}_b & v_6 &= \bar{Q}_a Q_b \\ v_2 &= Q_b \bar{Q}_c & v_7 &= \bar{Q}_b Q_c \\ v_3 &= Q_c \bar{Q}_d & v_8 &= \bar{Q}_c Q_d \\ v_4 &= Q_d \bar{Q}_e & v_9 &= \bar{Q}_d Q_e \end{aligned}$$

La implementación queda de la siguiente manera:



- 8) Analizar el circuito de la figura. Suponer que el estado inicial es $Q_A=1, Q_B=0, Q_C=0, Q_D=0$.



La realimentación se produce desde las etapas C y D , de forma que $J_A = C \oplus D$.

Como la situación inicial es $Q_A=1, Q_B=0, Q_C=0, Q_D=0$, la primera realimentación es $J_A = 0$. Al recibir el siguiente pulso de reloj habrá un desplazamiento a la derecha, pasando al estado $A=0, B=1, C=0, D=0$.

La secuencia completa de estados es la siguiente:

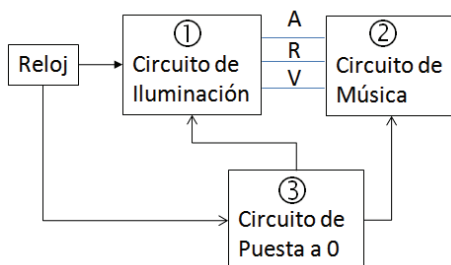
Estado	D	C	B	A	$f = C \oplus D$
S_1	0	0	0	1	0
S_2	0	0	1	0	0
S_4	0	1	0	0	1
S_9	1	0	0	1	1
S_3	0	0	1	1	0
S_6	0	1	1	0	1
S_{13}	1	1	0	1	0
S_{10}	1	0	1	0	1
S_5	0	1	0	1	1
S_{11}	1	0	1	1	1
S_7	0	1	1	1	1
S_{15}	1	1	1	1	0
S_{14}	1	1	1	0	0
S_{12}	1	1	0	0	0
S_8	1	0	0	0	1
S_1	0	0	0	1	0

Como se ve, hay 15 estados hasta volver al estado inicial. El estado 0000 no está incluido debido a que es un estado sin salida. Si el registro se encontrara en ese estado, quedaría atrapado en él.

9) Una fuente de agua posee tres focos luminosos (azul (A), rojo (R), verde (V)) y una caja musical que tiene 7 canales digitales de entrada, cada uno de ellos para activar cada una de las 7 notas musicales (Do, Re, Mi, Fa, Sol, La, Si). Por ejemplo, si se proporciona un 1 lógico por el canal de entrada de la nota "Mi" sonará esa nota durante un ciclo de reloj. La secuencia cíclica que se quiere seguir en la iluminación con los focos es: todos apagados, rojo, azul, rojo y azul, verde, verde y rojo, verde y azul, todos encendidos. Al mismo tiempo la secuencia de notas musicales debe ser: Re, Mi, Fa, Sol, Do, La, Si, La. Diseñar un circuito secuencial que implemente esta secuencia durante 15 segundos (suponer un reloj de frecuencia 1 Hz) y que termine con todos sus componentes apagados: luces y sonido.

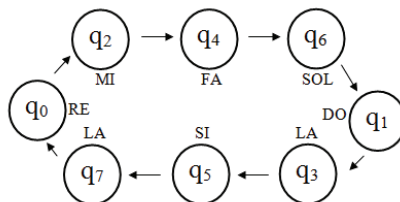
a) Resolver el sistema de iluminación (Circuito 1) tomando la codificación más adecuada al enunciado del problema. Mostrar la codificación de estados y el diagrama de estados.

- b) Tabla de excitación del circuito 1. Implementar el sistema de iluminación de la forma más sencilla posible utilizando biestables JK.
- c) Resolver el circuito 2, asociado al diseño anterior, que excite la caja musical según la secuencia indicada utilizando módulos.
- d) Resolver el sistema de apagado (circuito 3) de los dos sistemas anteriores utilizando los módulos y puertas lógicas necesarias.



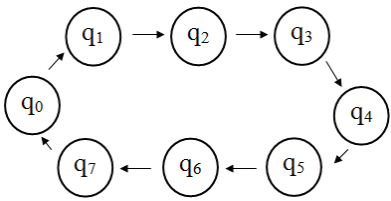
- a) La definición/codificación de estados y el diagrama de estados son.

	Q_A	Q_R	Q_V
q0: todos apagados	0	0	0
q1: encendido verde	0	0	1
q2: encendido rojo	0	1	0
q3: encendido v y r	0	1	1
q4: encendido azul	1	0	0
q5: encendido a y v	1	0	1
q6: encendido a y r	1	1	0
q7: encendidos todos	1	1	1



También se podría tomar otra solución como la representada a continuación:

	Q_V	Q_A	Q_R
q0: todos apagados	0	0	0
q1: encendido rojo	0	0	1
q2: encendido azul	0	1	0
q3: encendido rojo y azul	0	1	1
q4: encendido verde	1	0	0
q5: encendido verde y rojo	1	0	1
q6: encendido verde y azul	1	1	0
q7: encendidos todos	1	1	1



En lo que sigue se toma la primera solución.

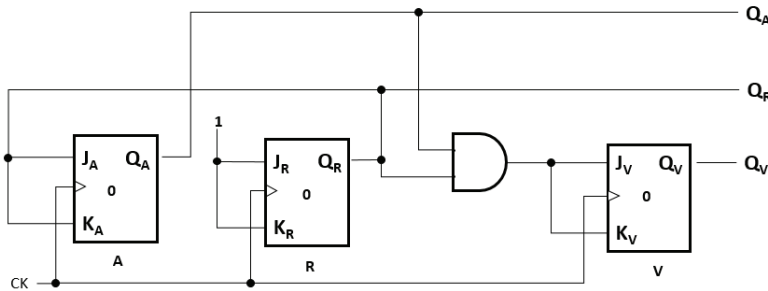
b) Tabla de excitación del circuito 1:

Q_A	Q_R	Q_V	Q_2 (t+1)	Q_1 (t+1)	Q_0 (t+1)	J_A	K_A	J_R	K_R	J_V	K_V
0	0	0	0	1	0	0	X	1	X	0	X
0	0	1	0	1	1	0	X	1	X	X	0
0	1	0	1	0	0	1	X	X	1	0	X
0	1	1	1	0	1	1	X	X	1	X	0
1	0	0	1	1	0	X	0	1	X	0	X
1	0	1	1	1	1	X	0	1	X	X	0
1	1	0	0	0	1	X	1	X	1	1	X
1	1	1	0	0	0	X	1	X	1	X	1

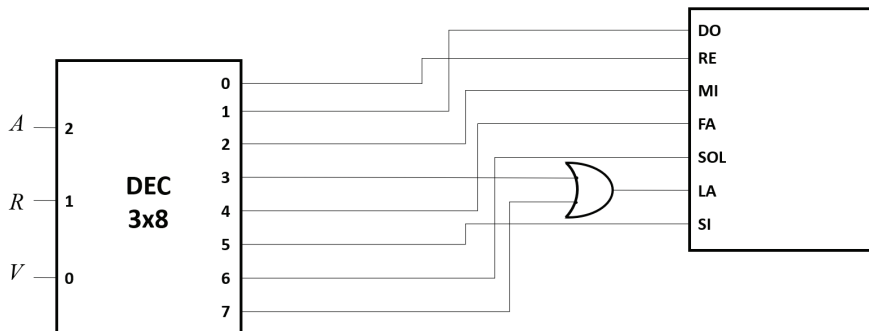
$$J_A = K_A = Q_R$$

$$J_R = K_R = 1$$

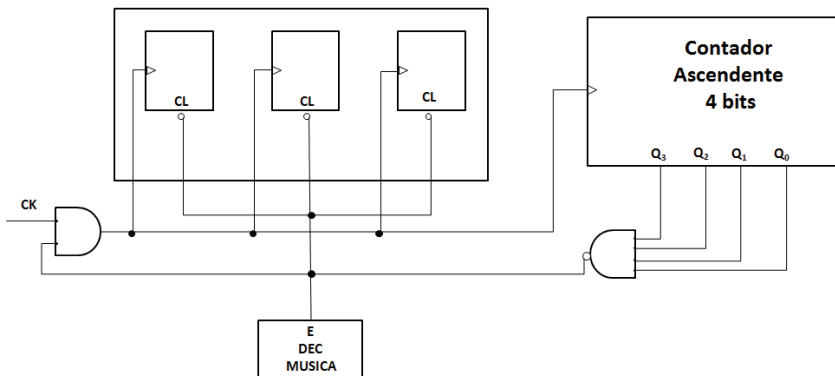
$$J_V = K_V = Q_A Q_R$$



c) Diseño de circuito 2.



d) Diseño de circuito 3.

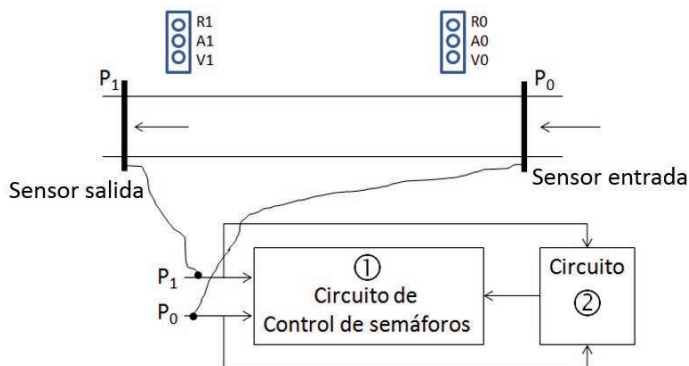


10) Se pretende diseñar un circuito secuencial (1) que active las 6 luces correspondientes a dos semáforos, uno para la entrada y otro para la salida de vehículos a un garaje. Cada semáforo cuenta con tres luces ROJA, AMBAR y VERDE ($R1$, $A1$, $V1$ y $R0$, $A0$ y $V0$ respectivamente) de forma que los vehículos sólo pueden circular cuando su semáforo de entrada esté en verde. Cada luz del semáforo se enciende poniendo a nivel 1 su correspondiente función booleana. El funcionamiento de los semáforos es el usual (el paso de verde a rojo se realiza a través del ámbar automáticamente y el paso de rojo a verde es directo).

Unos metros antes de llegar al semáforo, el vehículo pisa un sensor $P1$ (cuando se sale del garaje) o $P0$ (cuando se entra en el garaje), que actuará como entrada al circuito a diseñar (suponiendo un 1 lógico) y permitirá cambiar, si procediera, la situación actual de los semáforos, de modo que se eviten accidentes. En cualquier caso, siempre deberá haber un único semáforo en rojo. Nunca se dará el caso de que

los sensores sean activados simultáneamente y mientras no se pulse ningún sensor, la situación se mantendrá indefinidamente.

Diseñar el circuito que resuelva el problema siguiendo las siguientes fases.



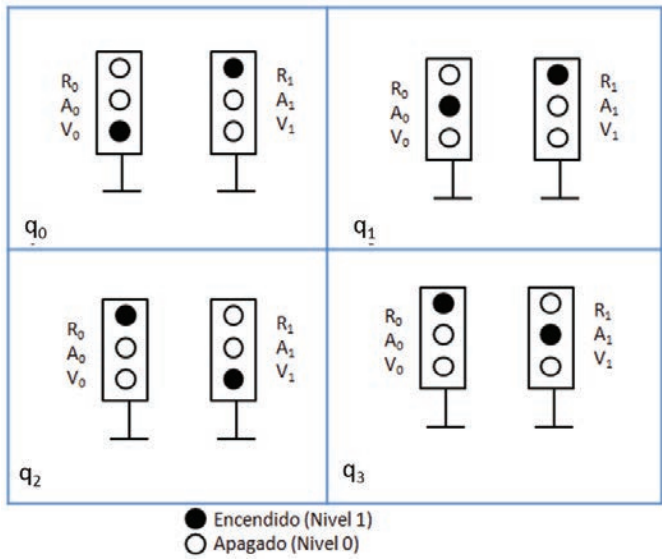
- Definir y codificar los estados del circuito 1 con el mínimo número de biestables. Realizar el diagrama de estados y establecer la tabla de transición y excitación del sistema. El orden de las variables será: variables externas y estado actual.
- Implementar el circuito secuencial utilizando biestables tipo D excitados con puertas lógicas.
- Diseñar la circuitería combinacional adicional en el circuito 1 de modo que se actúe sobre las luces de los semáforos a partir de las salidas de los biestables usando un DEC 2x4 y puertas lógicas.
- Suponer que el estacionamiento sólo puede acoger 6 coches. Diseñar el circuito adicional 2 para que cuando haya 6 coches en el garaje el sistema mantenga el semáforo de entrada en rojo y el de salida en verde permanentemente. Realízalo con los módulos y puertas necesarias.

a) Definición de estados y codificación.

Entradas

- $P_1P_0 = 00$ no se pisa ningún sensor
- $P_1P_0 = 01$ se pisa sólo el sensor de entrada
- $P_1P_0 = 10$ se pisa sólo el sensor de salida
- $P_1P_0 = 11$ se pisan ambos sensores (no se da el caso)

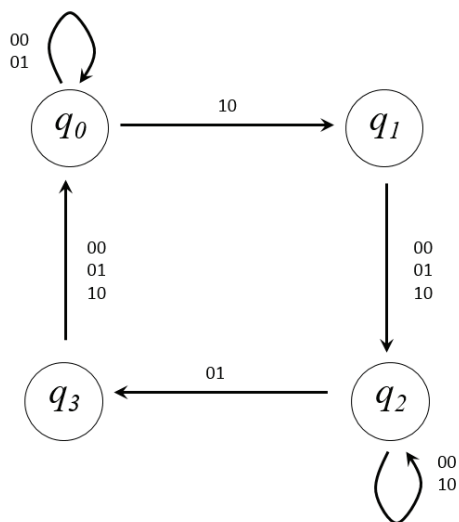
Una representación de los estados es la de la figura.



En la tabla siguiente aparecen los estados y las salidas asociadas a los estados (Autómata de Moore).

Estado	Descripción	Q_1	Q_0	V_1	A_1	R_1	V_0	A_0	R_0
q_0	Se puede entrar al garaje tranquilamente y no se puede salir	0	0	0	0	1	1	0	0
q_1	Se puede entrar al garaje pero sin descuidarse y no se puede salir	0	1	0	0	1	0	1	0
q_2	Se puede salir del garaje tranquilamente y no se puede entrar	1	0	1	0	0	0	0	1
q_3	Se puede salir del garaje pero sin descuidarse y no se puede entrar	1	1	0	1	0	0	0	1

El diagrama de estado y tabla del sistema son:



P1	P0	Q ₁	Q ₀	Q ₁ (t+1)	Q ₀ (t+1)
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	0	0
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X

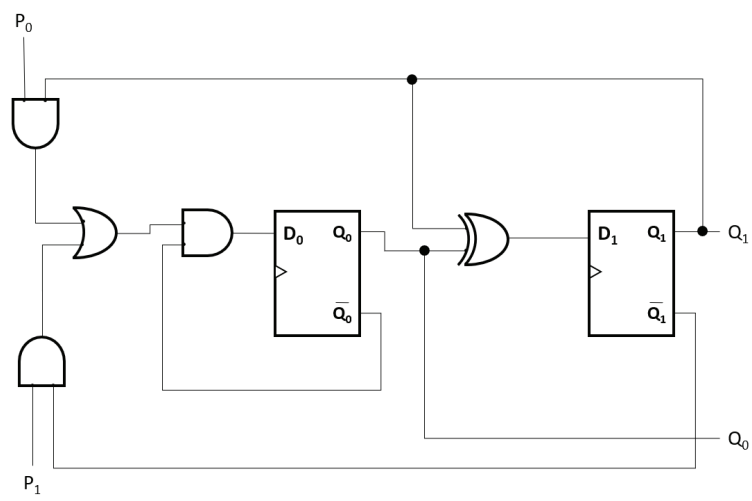
b) Esquema del circuito 1 con puertas:

		Q ₁ Q ₀			
	D₁	00	01	11	10
P₁P₀	00	0	1	0	1
	01	0	1	0	1
	11	X	X	X	X
	10	0	1	0	1

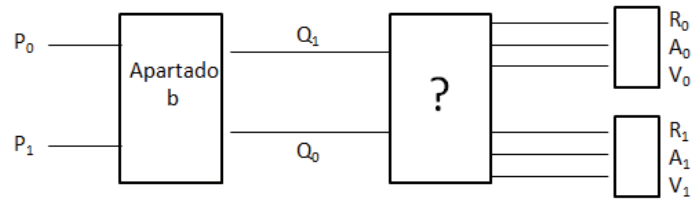
		Q ₁ Q ₀			
	D₀	00	01	11	10
P₁P₀	00	0	0	0	0
	01	0	0	0	1
	11	X	X	X	X
	10	1	0	0	0

$$D_1 = Q_1 \overline{Q_0} + \overline{Q_1} Q_0 = Q_1 \oplus Q_0$$

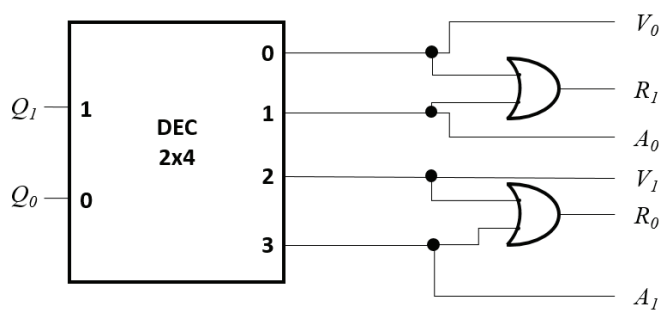
$$D_0 = P_1 \overline{Q_0} \overline{Q_1} + P_0 Q_1 \overline{Q_0} = \overline{Q_0} [P_1 \overline{Q_1} + P_0 Q_1]$$



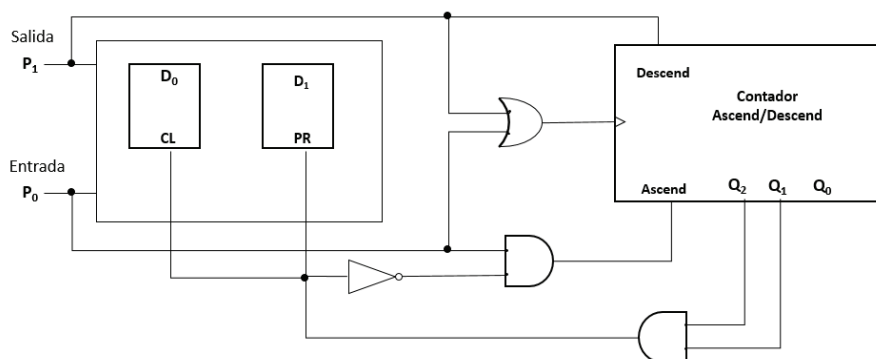
c) Circuito combinacional adicional:



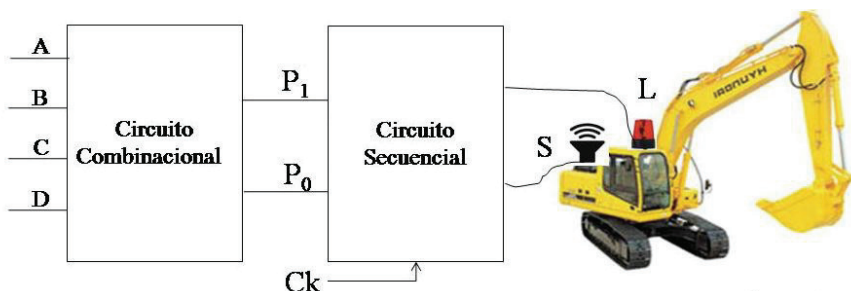
Q_1	Q_0	V_1	A_1	R_1	V_0	A_0	R_0
0	0	0	0	1	1	0	0
0	1	0	0	1	0	1	0
1	0	1	0	0	0	0	1
1	1	0	1	0	0	0	1



- d) Circuito 2. Cuando la cuenta de coches llegue a 6, el circuito debe ir al estado q_2 ($Q_1=1$, $Q_0=0$), e impedir que el contador ascendente siga contando.



- 11) Se pretende diseñar el sistema de encendido y apagado de una luz (L) y un efecto sonoro (S) de emergencia de una excavadora. Dicha luz y efecto sonoro de emergencia son activados dependiendo de la acción que esté realizado la excavadora. El circuito cumple las siguientes especificaciones, de acuerdo con la figura adjunta.



Cuando el operario acciona la marcha adelante, se genera un nivel 1 en la variable A y se deberá encender y apagar de forma intermitente una luz de emergencia que porta la excavadora en su techo. Esta intermitencia marcará las transiciones de la máquina secuencial, estableciendo el encendido y apagado en cada transición.

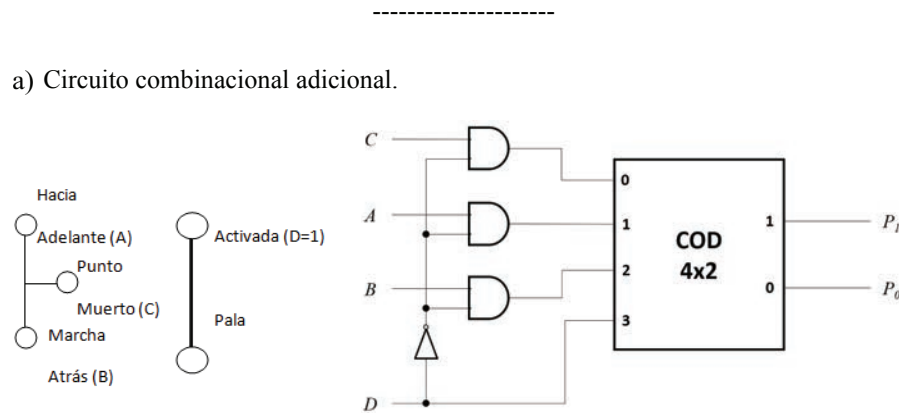
Cuando el operario acciona la marcha hacia atrás, se genera un nivel 1 en la variable B y se deberá producir un sonido intermitente de emergencia.

Cuando el operario pone en punto muerto la excavadora, se genera un nivel 1 en la variable C y no se encenderá ninguna señal luminosa ni sonora.

Finalmente, cuando el operario accione la pala de la excavadora, se genera un nivel 1 en la variable D y se deberá encender y apagar las dos señales (luminosa y

sonora). Además, la entrada D tiene prioridad absoluta independientemente de la marcha de movimiento en la que esté funcionando la excavadora.

- a) Diseñar primero la circuitería adicional necesaria (circuito combinacional de la figura) para reducir las cuatro entradas binarias (A, B, C y D) a sólo dos entradas binarias P_1 y P_0 , que serán las que actuarán sobre el sistema secuencial. En este circuito se deberá resolver la prioridad de la entrada D . Utilizar puertas y alguno de estos módulos: COD sin prioridad, DEC o MUX.
- b) Definir claramente las entradas, los estados y las salidas del sistema secuencial. Mostrar su significado exacto y cuál es la codificación usada. Usar la codificación más sencilla posible.
- c) Presentar el diagrama de estados del sistema secuencial. El orden de las variables será entradas, estado actual.
- d) Realizar la tabla de excitación del sistema secuencial usando biestables T.
- e) Implementar el circuito secuencial utilizando decodificadores con salida a nivel bajo y puertas de 2 y 3 entradas. Representar el diagrama lógico.

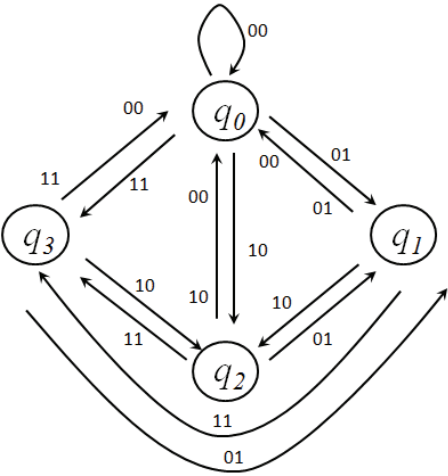


b) Definición de las entradas, los estados y las salidas:

Entradas del secuencial		Estados		Salidas del Secuencial
P_1P_0	Significado	Definición y codificación	Q_1Q_0	
00	Punto muerto	q_0 : Todo apagado	00	
01	Hacia Adelante	q_1 : Luz ON, sonido OFF	01	$L=Q_0$
10	Marcha Atrás	q_2 : Luz OFF, sonido ON	10	$S=Q_1$
11	Uso de la pala	q_3 : Todo Encendido	11	

c) Diagrama de estados.

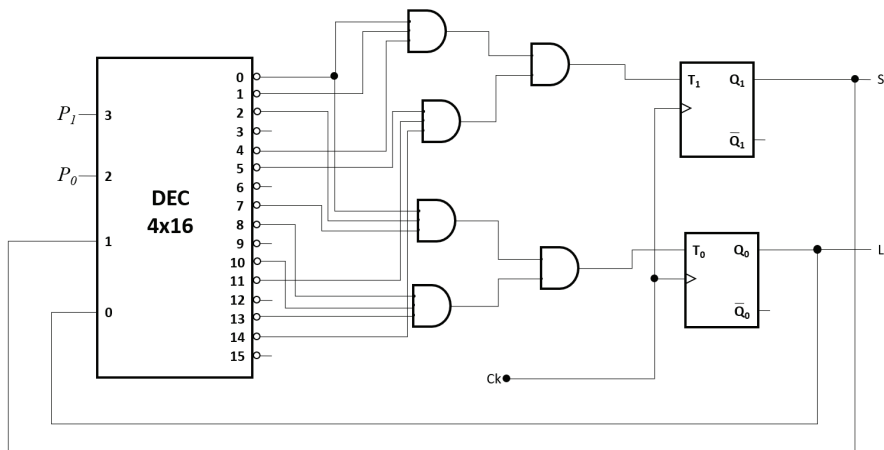
Es un autómata de Moore. Cada estado tiene asociada una salida de forma inequívoca.



d) Tabla de verdad

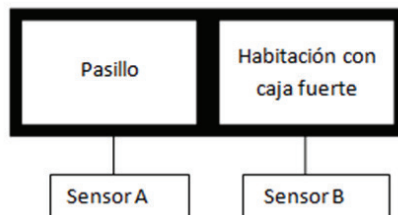
P_1	P_0	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	T_1	T_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	1
0	1	0	1	0	0	0	1
0	1	1	0	0	1	1	1
0	1	1	1	0	1	1	0
1	0	0	0	1	0	1	0
1	0	0	1	1	0	1	1
1	0	1	0	0	0	1	0
1	0	1	1	1	0	0	1
1	1	0	0	1	1	1	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	0	1
1	1	1	1	0	0	1	1

e) Implementación del circuito con decodificadores:



- 12) Se desea diseñar un sistema secuencial que sirva para avisar a un Guarda de Seguridad (G) e incluso a la Policía (P) de un intento de robo en una sucursal bancaria.

Para ello supondremos que la caja fuerte está en una habitación interior tal que para acceder a ella hay que pasar necesariamente por un pasillo. Dicho pasillo tiene un detector de presencia (sensor A) que se pone a 1 cuando hay una persona en el mismo y a 0 cuando no hay nadie. La habitación de la caja fuerte tiene otro detector (sensor B) que funciona de la misma manera. Suponemos que el ladrón nunca irá acompañado.



El sistema funciona de la siguiente manera. Mientras no entre nadie, el sistema permanecerá en reposo sin avisar ni al Guarda de Seguridad ni a la Policía. Si entra un ladrón al pasillo, se avisará al Guarda de Seguridad, pero no a la Policía. Si el ladrón avanza y llega hasta la habitación con la caja fuerte, se avisará, además, a la Policía. Si habiendo entrado el ladrón al pasillo decide salirse, dejarán de producirse los avisos, pero el sistema “recordará” que alguien ya estuvo en el pasillo, de modo que, si se vuelve a producir una entrada en el pasillo, se avisará al Guarda de Seguridad y a la Policía, exactamente igual que ocurriría si entra por primera vez

en la habitación de la caja fuerte, según se ha descrito anteriormente. Suponer que, en este último caso, el ladrón no pasa del pasillo a la habitación de la caja fuerte.

Si se llega a avisar a la Policía en cualquiera de los dos casos comentados, el sistema secuencial se queda en el estado alcanzado sin posibilidad de volver a otro estado, sea cual sea la entrada. Se necesitaría inicializar el sistema de forma “manual”, pero eso no es objeto de este problema. Se pide:

- Definir las entradas y salidas del sistema secuencial.
- Definir y codificar los estados.
- Realizar el diagrama de transición de estados.
- ¿Es un autómata de Moore o de Mealy? ¿Por qué?
- Realizar la tabla de transición y excitación usando un biestable T para la variable de excitación más significativa, un biestable D para la variable de excitación menos significativa y biestables JK para las restantes variables, si las hubiera.
- Implementar la excitación más significativa sólo con puertas NOR.
- Implementar la excitación menos significativa sólo con puertas NAND.
- Implementar las funciones de salida con puertas lógicas en dos niveles de la forma más sencilla.

a) Definiciones:

- Definiciones de entradas:

A=1 hay alguien en el pasillo

A=0 no hay nadie en el pasillo

B=1 hay alguien en la habitación

B=0 no hay nadie en la habitación

- Definición de salidas:

G=1 se le avisa al Guarda de Seguridad

G=0 no se le avisa al Guarda de Seguridad

P=1 se le avisa a la Policía

P=0 no se le avisa a la Policía

b) Definición de estados:

q_0 : estado inicial (nunca ha entrado nadie)

$Q_1 = 0 \quad Q_0 = 0$ TRANQUILIDAD TOTAL

q_1 : hay alguien en el pasillo por primera vez (sólo se avisa al Guarda de Seguridad)

$Q_1 = 0 \quad Q_0 = 1$ PREOCUPACIÓN

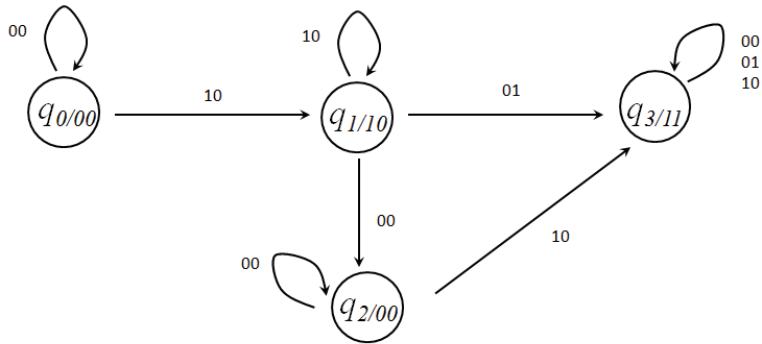
q_2 : no hay nadie en el pasillo, pero hubo alguien en algún momento (no se avisa a nadie)

$Q_1 = 1 \quad Q_0 = 0$ VIOLACIÓN PARCIAL DE SEGURIDAD

q_3 : Se ha alcanzado la habitación de la caja fuerte o alguien vuelve a estar en el pasillo (se avisa al Guarda y a la Policía)

$Q_1 = 1 \quad Q_0 = 1$ VIOLACIÓN TOTAL DE SEGURIDAD

c) Diagrama de estados:



d) Tipo de autómata: Autómata de Moore, porque cada estado tiene asociada de forma inequívoca una salida.

Estado		Salidas	
q_0	$(Q_1=0, Q_0=0)$	$G = 0$	$P = 0$
q_1	$(Q_1=0, Q_0=1)$	$G = 1$	$P = 0$
q_2	$(Q_1=1, Q_0=0)$	$G = 0$	$P = 0$
q_3	$(Q_1=1, Q_0=1)$	$G = 1$	$P = 1$

e) Tabla de verdad

A	B	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	T_1	D_0	G	P
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	1	0	1	0
0	0	1	0	1	0	0	0	0	0
0	0	1	1	1	1	0	1	1	1
0	1	0	0	X	X	X	X	X	X
0	1	0	1	1	1	1	1	1	0
0	1	1	0	X	X	X	X	X	X
0	1	1	1	1	1	0	1	1	1
1	0	0	0	0	1	0	1	0	0
1	0	0	1	0	1	0	1	1	0
1	0	1	0	1	1	0	1	0	0
1	0	1	1	1	1	0	1	1	1
1	1	0	0	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X

La salida G se corresponde directamente con Q_0 , mientras que la función P es la AND de Q_1 y Q_0 . Respecto a la excitación de los biestables, T_1 y D_0 son:

$$T_1 = \overline{A + Q_1 + \overline{Q_0}}$$

$$D_0 = A + B + Q_1 Q_0 = \overline{\overline{A} \overline{B} \overline{Q_1} \overline{Q_0}}$$

- 13) El sistema de apertura de una caja fuerte está compuesto por dos teclas A y B , un circuito secuencial ① y un circuito temporizador ②.

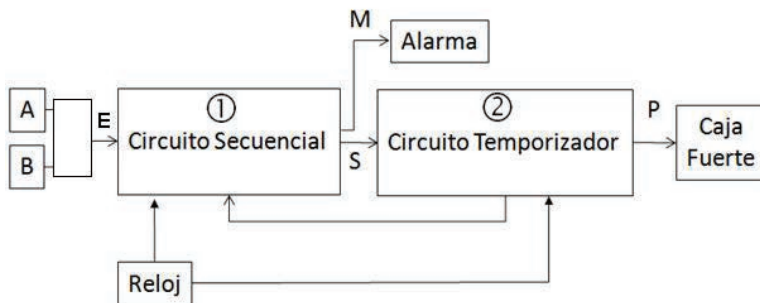
El circuito ① funciona de la siguiente manera: cuando se pulsa la tecla A , se produce un nivel lógico 1 que entra al circuito secuencial ($E=1$), mientras que cuando se pulsa la tecla B se produce un nivel lógico 0 de entrada al circuito a diseñar ($E=0$). Mientras no se pulse ninguna tecla, no se genera ningún nivel lógico de entrada al circuito secuencial. Nunca se pulsarán las dos teclas a la vez.

Para abrir la caja fuerte, la combinación secreta de cuatro pulsaciones es: pulsar dos veces seguidas la tecla A , a continuación, pulsar una vez la tecla B , y finalmente pulsar una vez la tecla A . Si se hace de esta manera, el circuito secuencial dará una salida S a nivel lógico 1 que actuará sobre el temporizador ②. Si en cualquier momento se introdujera un error al pulsar la secuencia secreta saltará una alarma sonora M y en el siguiente ciclo de reloj todos los biestables se pondrán a cero (el

sistema pasará al estado inicial), y la secuencia debe volver a introducirse desde el principio.

Cuando la salida S es 1, el circuito ② hace que se genere una salida $P=1$ durante 12 segundos, tiempo durante el que la caja fuerte estará abierta. Transcurrido ese tiempo la caja fuerte se cierra y todo el sistema se reinicia.

- Dibujar el diagrama de estados para el circuito ① explicando claramente entradas, salidas y estados.
- Realizar la tabla de verdad para el circuito ① y diseñarlo con biestables JK y las puertas necesarias. El orden de las variables será: entrada externa, variables de estado.
- Diseñar el circuito temporizador ② y acoplarlo con el resto del sistema usando los módulos y puertas que se considere oportuno. Suponer que cuando se alcanza $S=1$ el pulsado de las teclas A y B está bloqueado.



a) Circuito ①

- $q_0(t)$: estado inicial.
- $q_1(t)$: Se pulsó A en $(t-1)$ (se ha recibido un 1 en $t-1$)
- $q_2(t)$: Se pulsó A en $(t-2)$ y en $(t-1)$ (se han recibido 11 en $t-2$ y $t-1$)
- $q_3(t)$: Se pulsó A en $(t-3)$ y $(t-2)$ y B en $(t-1)$, (se han recibido 110 en $t-3$, $t-2$ y $t-1$)

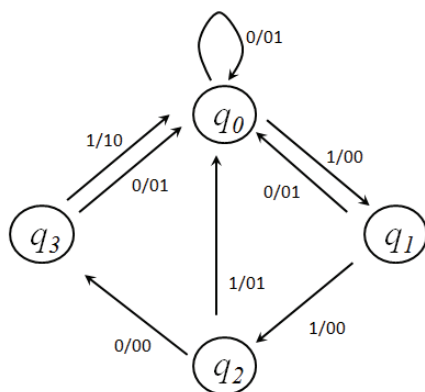
Entrada: E

Salidas:

- $S(t)=1$, si se reconoce el patrón 1101 (en $t-3$, $t-2$, $t-1$, t)
- $S(t)=0$ en caso contrario
- $M(t)=1$, si se pierde la secuencia del patrón en t
- $M(t)=0$ en caso contrario

La codificación de estados y diagrama de estados es el siguiente: (En las transiciones del diagrama de estados aparecen los valores E/SM)

Q_1	Q_0	
0	0	q_0
0	1	q_1
1	0	q_2
1	1	q_3



b) Tabla de excitación y transición:

E	Q_1	Q_0	Q_1 (t+1)	Q_0 (t+1)	J_1	K_1	J_0	K_0	S	M
0	0	0	0	0	0	X	0	X	0	1
0	0	1	0	0	0	X	X	1	0	1
0	1	0	1	1	X	0	1	X	0	0
0	1	1	0	0	X	1	X	1	0	1
1	0	0	0	1	0	X	1	X	0	0
1	0	1	1	0	1	X	X	1	0	0
1	1	0	0	0	X	1	0	X	0	1
1	1	1	0	0	X	1	X	1	1	0

Expresiones de entrada a los biestables y salida:

$$J_1 = EQ_0$$

$$K_1 = E + Q_0$$

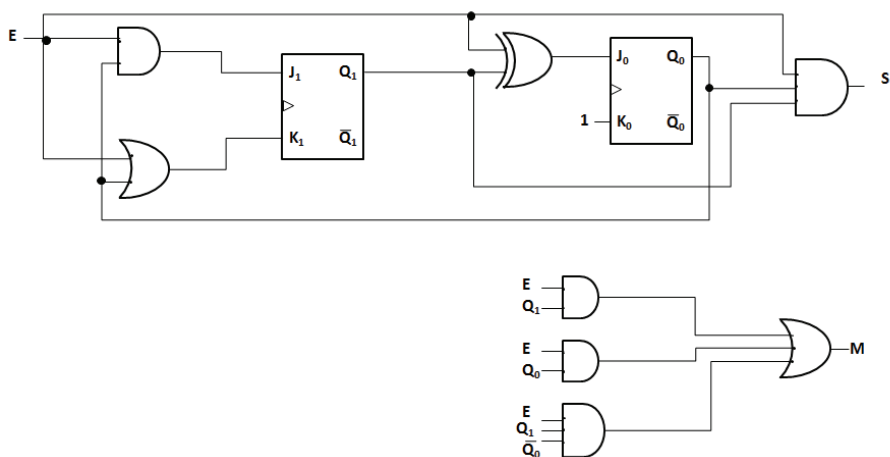
$$J_0 = E \oplus Q_1$$

$$K_0 = 1$$

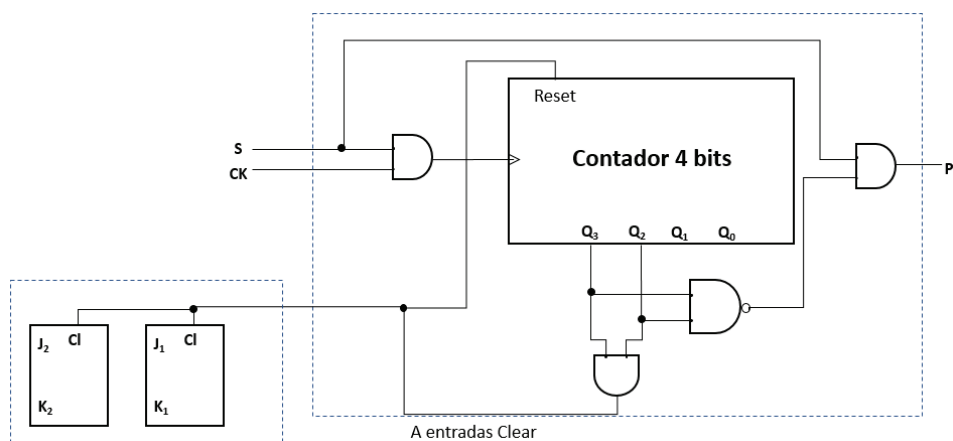
$$S = EQ_1Q_0$$

$$M = \overline{E}\overline{Q}_1 + \overline{E}Q_0 + EQ_1\overline{Q}_0$$

Circuito 1:



c) Diseño circuito 2:

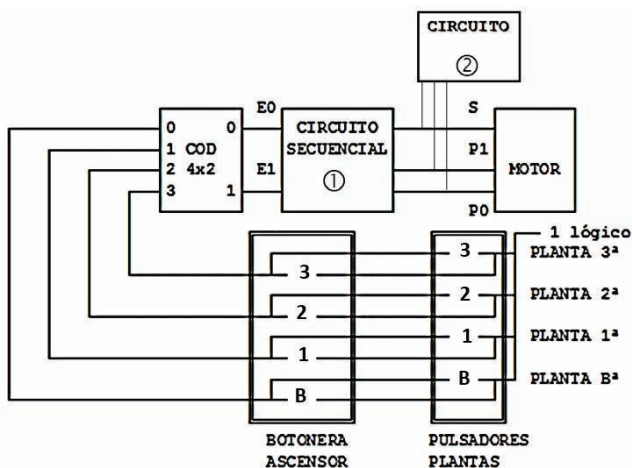


Como se puede ver, al comienzo del segundo 12 se inicializa.

- 14) Se pretende diseñar un circuito secuencial ① que controle el motor de un ascensor en un edificio de 3 plantas más planta baja. El motor del ascensor tiene una entrada S que indica si el ascensor debe subir ($S=0$) o bajar ($S=1$), y dos entradas P_1 y P_0 que le indican al motor cuántas plantas debe subir o bajar. El circuito secuencial a diseñar tendrá por entradas las dos salidas de un codificador cuyas entradas aparecen como 0, 1, 2 y 3 en la figura, y corresponden a los botones de las plantas o de la botonera del ascensor.

El funcionamiento del ascensor es el siguiente: cuando desde la propia botonera del ascensor o desde una cierta planta se pulsa el botón de llamada, el motor del ascensor se pondrá en marcha para ir a esa planta.

Utilizar las salidas S , P_1 y P_0 para diseñar circuito ② en el que se visualice en un display rojo o azul el número de plantas que debe subir o bajar (según el caso) el ascensor.

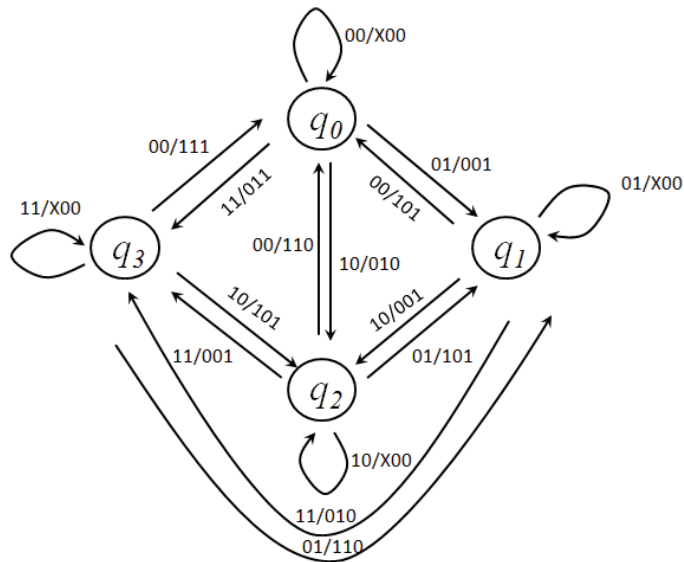


- Sistema ①. Definir las entradas, salidas y estados, explicando claramente en qué consiste cada uno.
- Sistema ①. Encontrar el diagrama de estados y las tablas de transición y excitación usando biestables RS. El orden de las variables será: entrada externa, variables de estado.
- Sistema ①. Diseñar el circuito secuencial pedido usando biestables RS, un MUX 8x1 para S , un decodificador para P_1 y una puerta XOR para P_0 .
- Diseñar el circuito ② con los módulos y puertas que consideres oportuno.

a) Sistema ①: Definición de entradas, salidas y estados.

Entradas		Estados		Salidas			
E_1E_0		Q_1Q_0		P_1P_0		S	
00	Ir a planta 0	00	Está en planta 0	00	Mueve 0 plantas	0	Sube
01	Ir a planta 1	01	Está en planta 1	01	Mueve 1 planta	1	Baja
10	Ir a planta 2	10	Está en planta 2	10	Mueve 2 plantas		
11	Ir a planta 3	11	Está en planta 3	11	Mueve 3 plantas		

b) Diagrama de estados.



Sabiendo la tabla de excitación del biestable SR la siguiente

$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

se obtiene la siguiente tabla de transición y excitación del sistema.

E1	E0	Q ₁	Q ₀	Q ₁ (t+1)	Q ₀ (t+1)	S ₁	R ₁	S ₀	R ₀	S	P1	P0
0	0	0	0	0	0	0	X	0	X	X	0	0
0	0	0	1	0	0	0	X	0	1	1	0	1
0	0	1	0	0	0	0	1	0	X	1	1	0
0	0	1	1	0	0	0	1	0	1	1	1	1
0	1	0	0	0	1	0	X	1	0	0	0	1
0	1	0	1	0	1	0	X	X	0	X	0	0
0	1	1	0	0	1	0	1	1	0	1	0	1
0	1	1	1	0	1	0	1	X	0	1	1	0
1	0	0	0	1	0	1	0	0	X	0	1	0
1	0	0	1	1	0	1	0	0	1	0	0	1
1	0	1	0	1	0	X	0	0	X	X	0	0
1	0	1	1	1	0	X	0	0	1	1	0	1
1	1	0	0	1	1	1	0	1	0	0	1	1
1	1	0	1	1	1	1	0	X	0	0	1	0
1	1	1	0	1	1	X	0	1	0	0	0	1
1	1	1	1	1	1	X	0	X	0	X	0	0

c) Diseño circuito 1.

Las expresiones de las salidas son:

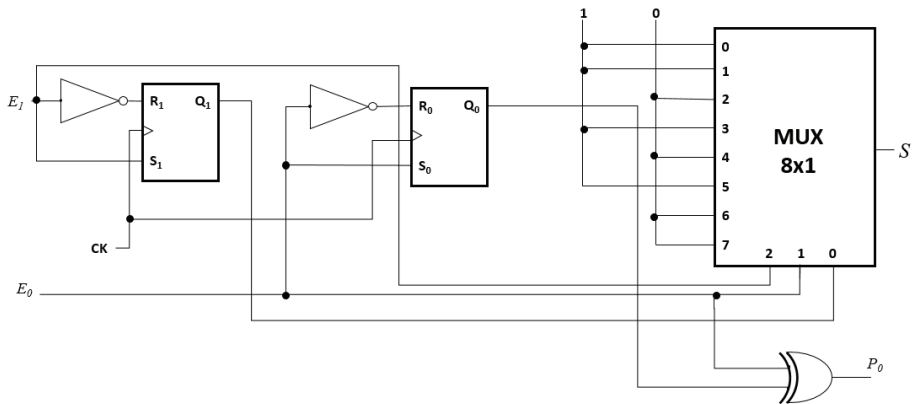
$$R_1 = \overline{E_1}; S_1 = E_1$$

$$R_0 = \overline{E_0}; S_0 = E_0$$

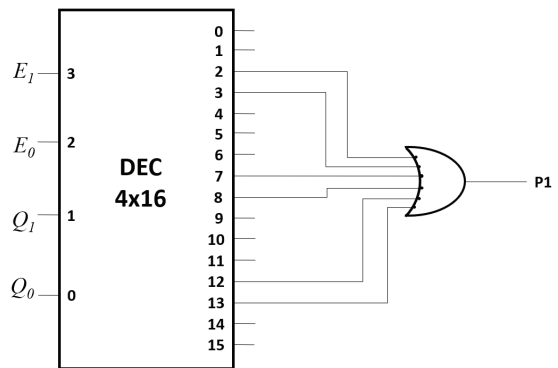
$$P_0 = E_0 \oplus Q_0$$

$$P_1 = \sum (2, 3, 7, 8, 12, 13)$$

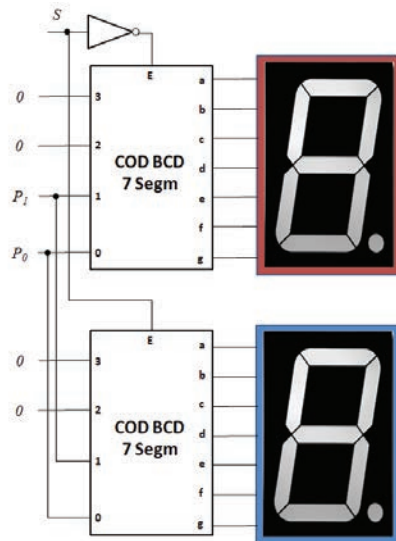
Las implementaciones de P_0 y S son:



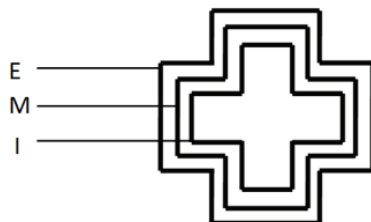
La implementación de P_1 es:



d) Diseño circuito 2. El display superior es de subida, mientras que el inferior, es de bajada



- 15) Se trata de diseñar un circuito que controle el funcionamiento del luminoso de una farmacia según la figura adjunta. Como se ve, tiene tres aros luminosos cerrados en forma de cruz, uno interior (*I*), otro medio (*M*) y otro exterior (*E*).



Cuando se aplica un 1 lógico sobre cualquier punto del contorno, se enciende dicho contorno, y cuando se aplican 0 lógico permanece apagado.

La secuencia de encendido debe ser la siguiente y en este orden: todos apagados, encendido sólo el interior, encendido sólo el intermedio, encendido sólo el exterior, encendido sólo el intermedio, encendido sólo el interior, todos apagados, todos encendidos. Y a partir de ahí se vuelve a repetir.

El problema se puede resolver de dos formas distintas e independientes.

Se pide:

- 1ª forma. Diseñar un contador de 3 bits ascendente con biestables T y la lógica necesaria que toma como entrada las salidas del contador y cuyas salidas activan los aros luminosos *E*, *M*, *I*. A cada número de cuenta le corresponde cada uno de los ciclos indicados, asignando el número de cuenta 000 al momento inicial del ciclo descrito, el número de cuenta 001 al siguiente, y así sucesivamente. Implementar *I* y *E* con puertas y *M* con un MUX 4x1.
- 2ª forma. Diseñar un contador de 3 bits, de manera que la salida de los biestables *Q2*, *Q1* y *Q0* activen, respectivamente, *E*, *M*, *I*. Usar para ellos biestables T para *Q2*, *Q1* y *Q0*, y si se necesitasen más biestables, deberán ser tipo D. Para la excitación de los biestables usar puertas lógicas.

- Primera solución. Para implementar las salidas, se necesitará un circuito decodificador a la salida de los biestables.

Tabla y mapas:

Estado actual t			Estado siguiente t+1					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	T_2	T_1	T_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

		$Q_1 Q_0$			
T_2	Q_2	00	01	11	10
0	0	0	0	1	0
1	0	0	0	1	0

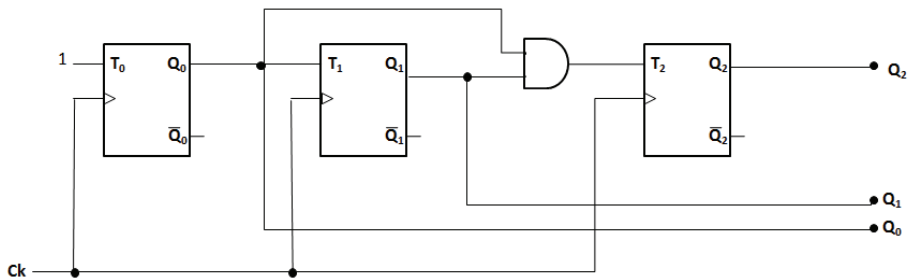
$$T_2 = Q_1 Q_0$$

		$Q_1 Q_0$			
T_1	Q_2	00	01	11	10
0	0	0	1	1	0
1	0	0	1	1	0

$$T_1 = Q_0$$

$$T_0 = 1$$

El circuito de excitación de los biestables es el siguiente:



El circuito decodificador que conecta las salidas de los biestables con el sistema de luces se implementa a continuación.

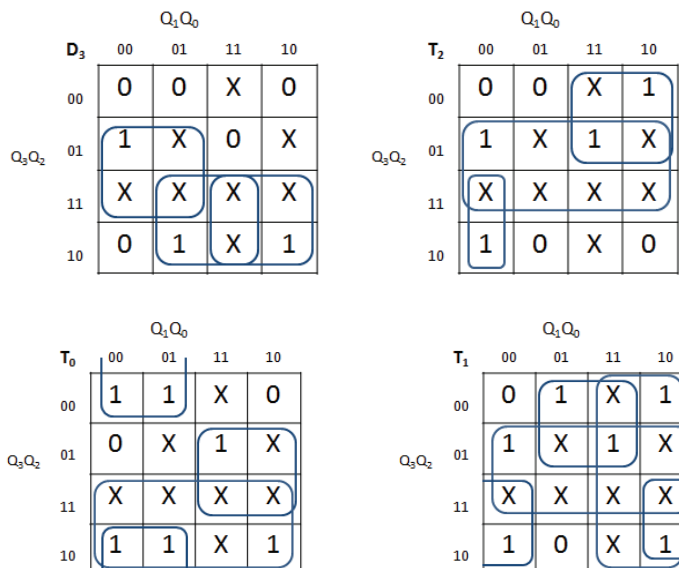
Tabla y mapas de Karnaugh del circuito decodificador:

Definición de Estados		Codificación de Estados			
		Q ₃	Q ₂	Q ₁	Q ₀
q _{0A}	Número de cuenta 0 (1ª Vez)	0	0	0	0
q _{1A}	Número de cuenta 1 (1ª Vez)	0	0	0	1
q _{2A}	Número de cuenta 2 (1ª Vez)	0	0	1	0
q _{4A}	Número de cuenta 4 (Única vez)	0	1	0	0
q _{2B}	Número de cuenta 2 (2ª Vez)	1	0	1	0
q _{1B}	Número de cuenta 1 (2ª Vez)	1	0	0	1
q _{0B}	Número de cuenta 0 (2ª Vez)	1	0	0	0
q _{7B}	Número de cuenta 7 (Única vez)	0	1	1	1

Tabla de excitación:

Q ₃	Q ₂	Q ₁	Q ₀	Q ₃	Q ₂	Q ₁	Q ₀	D	T ₂	T ₁	T ₀
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	1	0	0	0	1	1	0
0	0	1	1	X	X	X	X	X	X	X	X
0	1	0	0	1	0	1	0	1	1	1	0
0	1	0	1	X	X	X	X	X	X	X	X
0	1	1	0	X	X	X	X	X	X	X	X
0	1	1	1	0	0	0	0	0	1	1	1
1	0	0	0	0	1	1	1	0	1	1	1
1	0	0	1	1	0	0	0	1	0	0	1
1	0	1	0	1	0	0	1	1	0	1	1
1	0	1	1	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X

Mapas de Karnaugh y expresiones de las entradas a los biestables:



$$D_3 = Q_2 \overline{Q_1} + Q_3 Q_0 + Q_3 Q_1$$

$$T_2 = Q_2 + \overline{Q_3} Q_1 + Q_3 \overline{Q_1} \overline{Q_0}$$

$$T_1 = Q_2 + Q_1 + \overline{Q_3} Q_0 + Q_3 \overline{Q_0} = Q_2 + Q_1 + Q_3 \oplus Q_0$$

$$T_0 = Q_3 + \overline{Q_2} \overline{Q_1} + Q_2 Q_1$$

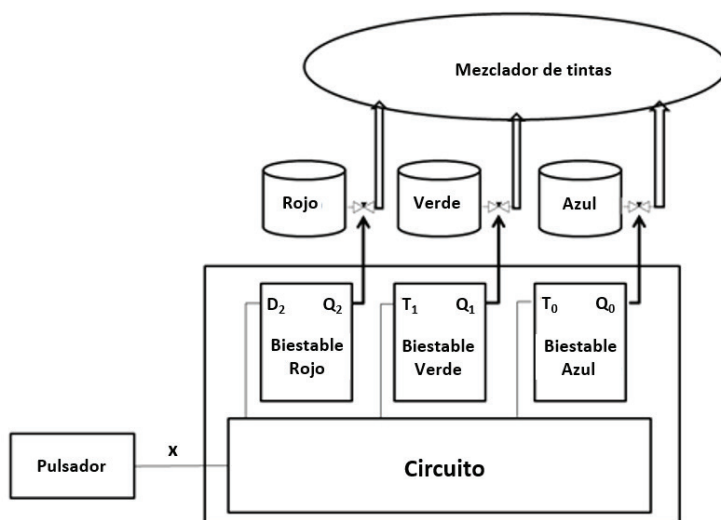
- 16) Se quiere diseñar un circuito digital que controle un sistema de mezcla de tres cartuchos de tinta con los colores básicos (Rojo, Verde y Azul). Como se aprecia en la figura, la salida de cada cartucho está activada por su correspondiente biestable. Si el biestable está a 1, se abre la válvula correspondiente y la tinta de ese color entra al sistema mezclador. La tabla adjunta muestra el color que se consigue para cada mezcla posible. Para el caso $Q_2Q_1Q_0=000$ se considera que no existe tinta en el mezclador ya que todas las válvulas estarían cerradas.

Se desea generar una secuencia concreta de colores que salgan del mezclador. La secuencia es: Azul, Cian, Magenta, Blanco, Rojo. El circuito funciona de la siguiente forma. El sistema inicialmente tiene las tres válvulas cerradas y no hay pintura en el mezclador. Este es el estado inicial. Para introducir cada mezcla se actúa sobre el pulsador x (cuando pulsamos, $x=1$, y si no pulsamos, $x=0$). Cada vez que se presiona el pulsador ($x=1$) se abren las válvulas correspondientes a la mezcla de colores. Esto ocurre hasta que se vuelve a presionar de nuevo, en cuyo caso la siguiente mezcla se genera. Por lo tanto, si en cualquier estado no se pulsa el pulsador, se mantiene la

mezcla de color en la que se esté. Una vez alcanzado el último color de la secuencia (en este caso el Rojo), si se vuelve a pulsar el pulsador el sistema vuelve al estado de inicial $Q_2Q_1Q_0=000$.

Se pide:

- Definición de estados y codificación de estados. En la codificación, tener presente que cada biestable actúa sobre su cartucho de tinta.
- Diagrama de estados.
- Tabla de verdad usando un biestable D para el más significativo y biestables T para el resto. La ordenación de variables en la tabla debe ser así: $xQ_2Q_1Q_0$.
- Implementación con puertas NAND para la excitación del biestable D, con puertas NOR para T_1 y con un DEC para T_0 .
- Explica cómo cambiarías el diseño (añadir y/o quitar cosas) para que las mezclas no se hicieran bajo órdenes de pulsador, sino que se hicieran de forma automática y que la duración de cada mezcla fuera durante 10 segundos. Puedes utilizar cualquier tipo de componente secuencial o combinacional. Incluye el circuito lógico con las modificaciones realizadas.



Se adjunta una tabla que indica cómo se obtienen los colores a partir de las componentes básicas: rojo, verde y azul.

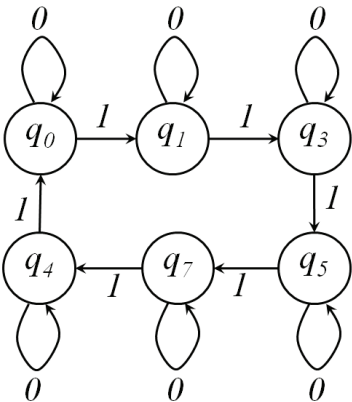
R	G	B	Color
0	0	0	Sin color
0	0	1	Azul
0	1	0	Verde
0	1	1	Cian
1	0	0	Rojo
1	0	1	Magenta
1	1	0	Amarillo
1	1	1	Blanco

a) Estados y Codificación:

- $q_0(t)$: Estado inicial. Válvulas cerradas en $(t-1)$
- $q_1(t)$: El color de la mezcla es azul en $(t-1)$
- $q_3(t)$: El color de la mezcla es cian en $(t-1)$
- $q_5(t)$: El color de la mezcla es magenta en $(t-1)$
- $q_7(t)$: El color de la mezcla es blanco en $(t-1)$
- $q_4(t)$: El color de la mezcla es rojo en $(t-1)$

Estado	Q_2	Q_1	Q_0
q_0	0	0	0
q_1	0	0	1
q_3	0	1	1
q_5	1	0	1
q_7	1	1	1
q_4	1	0	0

b) Diagrama de estados. Es un autómata de Moore:



c) Tabla de verdad.

x	Q ₂	Q ₁	Q ₀	Q ₂	Q ₁	Q ₀	D ₂	T ₁	T ₀
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0
0	0	1	0	X	X	X	x	x	X
0	0	1	1	0	1	1	0	0	0
0	1	0	0	1	0	0	1	0	0
0	1	0	1	1	0	1	1	0	0
0	1	1	0	X	x	X	x	x	X
0	1	1	1	1	1	1	1	0	0
1	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	1	0	1	0
1	0	1	0	X	x	X	x	x	X
1	0	1	1	1	0	1	1	1	0
1	1	0	0	0	0	0	0	0	0
1	1	0	1	1	1	1	1	1	0
1	1	1	0	X	x	X	x	x	X
1	1	1	1	1	0	0	1	1	1

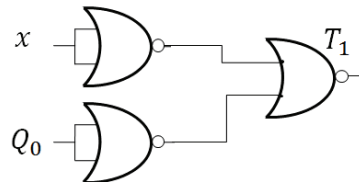
d) Implementación con NAND y NOR:

		Q ₁ Q ₀			
	D ₂	00	01	11	10
x Q ₂	00	0	0	0	X
	01	1	1	1	X
	11	0	1	1	X
	10	0	0	1	X

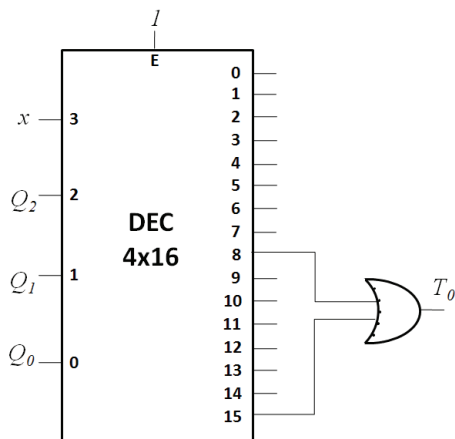
$$D_2 = \overline{xQ_1} \cdot \overline{Q_2Q_0} \cdot \overline{xQ_2}$$

		Q ₁ Q ₀			
	T ₁	00	01	11	10
X Q ₂	00	0	0	0	X
	01	0	0	0	X
	11	0	1	1	X
	10	0	1	1	x

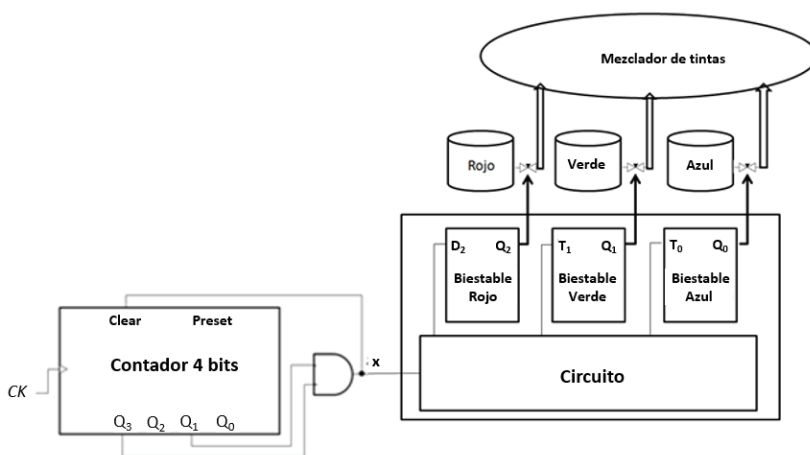
$$T_1 = xQ_0 = \overline{x} + \overline{Q_0}$$



Implementación de T₀ con decodificador:

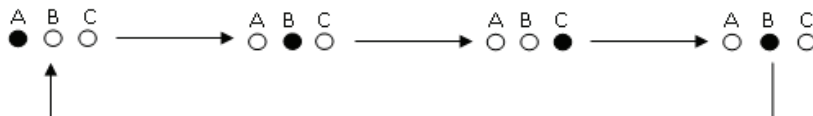


- e) Inicializando a cero un contador de 4 bits con una entrada de reloj de 1 Hz. Al principio de la cuenta 10 el contador se pone a cero. La entrada *Clear* pone a cero todo el contador.



- 17) En los años 80 se hizo famosa una serie televisiva que llevaba el nombre de “El coche fantástico”. Una de las características que lo distinguía era que en el frontal tenía un juego de luces que se encendían de izquierda a derecha y llegado al extremo cambiaba de sentido hasta llegar a la izquierda, y después repetía la secuencia sucesivamente. Pues bien, en este ejercicio se trata de diseñar un circuito secuencial que realice esa función. Por simplicidad, se va a hacer sólo con tres luces *A*, *B* y *C*. La luz que aparece en negro está encendida (nivel lógico 1).

Se pide diseñar un circuito secuencial síncrono, basado en un autómata de Moore, que encienda de forma repetitiva la siguiente secuencia de luces, mostradas en la figura adjunta, al ritmo del reloj. Las flechas marcan el orden temporal.



Para el diseño del circuito es necesario que las salidas Q_2 , Q_1 y Q_0 de los biestables se conecten directamente a A , B y C , respectivamente.

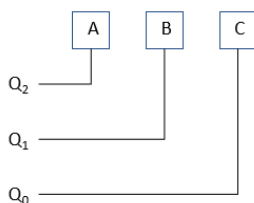
Se recomienda definir los estados en función del número binario que en cada momento representan las tres luces, y usar un biestable adicional Q_3 para el caso en que un mismo número pueda ser representado por dos estados distintos.

El biestable 0 debe ser de tipo JK, el biestable 1 debe ser tipo T y el resto de biestables deben ser tipo D.

Se pide:

- Obtener la tabla de transición y excitación con los biestables citados.
- Implementar las entradas a los biestables con puertas lógicas.
- Dibujar el circuito.

- Se definen primero los estados y su codificación.



Secuencia	Estados	Codificación			
		Q_3	Q_2	Q_1	Q_0
1º	Encendida A	0	1	0	0
2º	Encendida B	0	0	1	0
3º	Encendida C	0	0	0	1
4º	Encendida B de nuevo	0	0	1	0

La tabla de transición y excitación es la siguiente:

Q_3	Q_2	Q_1	Q_0	Q_3 (t+1)	Q_2 (t+1)	Q_1 (t+1)	Q_0 (t+1)	D_3	D_2	T_1	J_0	K_0
0	0	0	0	X	X	X	X	X	X	X	X	X
0	0	0	1	1	0	1	0	1	0	1	X	1
0	0	1	0	0	0	0	1	0	0	1	1	X
0	0	1	1	X	X	X	X	X	X	X	X	X
0	1	0	0	0	0	1	0	0	0	1	0	X
0	1	0	1	X	X	X	X	X	X	X	X	X
0	1	1	0	X	X	X	X	X	X	X	X	X
0	1	1	1	X	X	X	X	X	X	X	X	X
1	0	0	0	X	X	X	X	X	X	X	X	X
1	0	0	1	X	X	X	X	X	X	X	X	X
1	0	1	0	0	1	0	0	0	1	1	0	X
1	0	1	1	X	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X	X

b) Implementación de las entradas a los biestables:

D_3

	$Q_1 Q_0$	00	01	11	10
$Q_3 Q_2$	00	X	1	X	0
	01	0	X	X	X
	11	X	X	X	X
	10	X	X	X	0

D_2

	$Q_1 Q_0$	00	01	11	10
$Q_3 Q_2$	00	X	0	X	0
	01	0	X	X	X
	11	X	X	X	X
	10	X	X	X	1

T_1

	$Q_1 Q_0$	00	01	11	10
$Q_3 Q_2$	00	X	1	X	1
	01	1	X	X	X
	11	X	X	X	X
	10	X	X	X	1

J_0

	$Q_1 Q_0$	00	01	11	10
$Q_3 Q_2$	00	X	X	X	1
	01	0	X	X	X
	11	X	X	X	X
	10	X	X	X	0

K_0

	$Q_1 Q_0$	00	01	11	10
$Q_3 Q_2$	00	X	1	X	X
	01	X	X	X	X
	11	X	X	X	X
	10	X	X	X	X

$$D_3 = Q_0$$

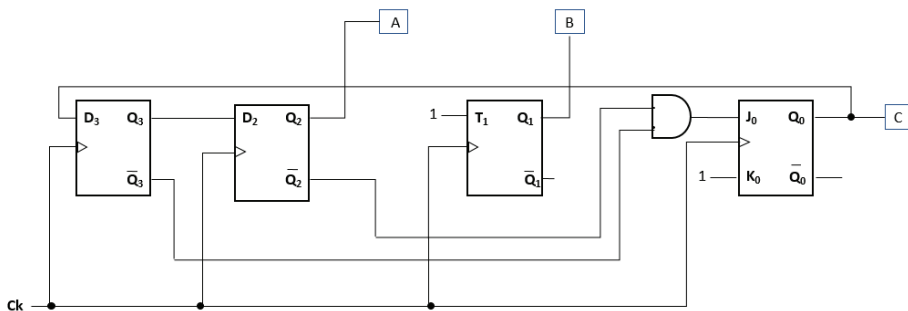
$$D_2 = Q_3$$

$$T_1 = 1$$

$$J_0 = \overline{Q_3}Q_1 \quad \text{o} \quad J_0 = \overline{Q_3}\overline{Q_2} = \overline{Q_3 + Q_2}$$

$$K_1 = 1$$

c) Diseño del circuito:



18) Se desea diseñar una máquina secuencial que dispense tickets de aparcamiento para la zona azul, que es una modalidad de prepago de tiempo de aparcamiento. Las características establecidas por la empresa que gestiona el servicio son las siguientes:

- La máquina sólo admite monedas de 0,50 €. Prever como una entrada más el caso de no echar monedas (moneda de 0 €).
- En cualquier momento el cliente puede cancelar su operación, pulsando un botón rojo, y se le devuelve todo el dinero introducido, sólo en monedas de 0,50 €.
- El precio máximo que se puede pagar es de 1,50 €. Incluso habiendo introducido todo el dinero, se puede cancelar la operación como en el apartado anterior.
- En cualquier momento, con el dinero introducido, el cliente puede confirmar que quiere el ticket, pulsando un botón verde, y en ese caso se imprimirá el ticket con una hora de finalización que será la suma de la hora actual y el tiempo “comprado”.
- Desde que se introduce la primera moneda, un display presenta la cantidad de dinero introducido hasta el momento y la hora de finalización del tiempo de aparcamiento. Este apartado es irrelevante para la resolución de nuestro problema.

- Las salidas del sistema son el dinero devuelto en caso de cancelación de la operación y la orden de impresión del ticket (SI/NO). Los detalles que deban imprimirse en el ticket son ajenos a nuestro problema.
- Suponer que nunca se pulsará el botón rojo (CANCELAR) ni el botón verde (CONFIRMAR) antes de haber introducido monedas. Suponer también que cuando se ha introducido el máximo dinero (1,50 €) no se introduce ninguna moneda más.

Se pide:

- Definir y codificar las entradas, incluyendo los botones rojo y verde.
- Definir y codificar los estados.
- Definir y codificar las salidas. Usar el mismo criterio para codificar el dinero que se usó al codificar la entrada de monedas.
- Realizar el diagrama de estados.
- Realizar la tabla de verdad completa, usando biestables JK.
- Implementar todas las salidas y excitaciones de los biestables usando puertas lógicas. No es necesario dibujar el circuito.

- Definición y codificación de entradas.

E_1	E_0	Significado
0	0	No se introduce moneda (0€) ni se pulsa ningún botón
0	1	Se introduce la moneda (única) de 0,50€
1	0	Se pulsa el botón rojo para CANCELAR
1	1	Se pulsa el botón verde para CONFIRMAR

- Definición y codificación de las salidas.

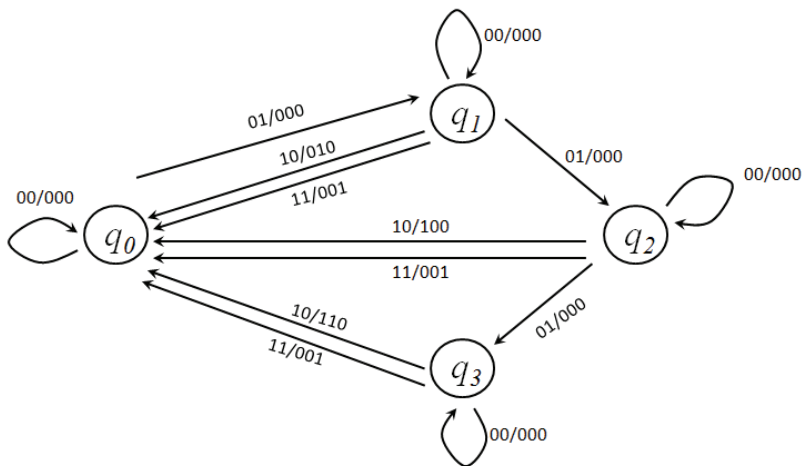
Significado	S_1	S_0
No se devuelve dinero alguno (0 €)	0	0
Se devuelve 0,50 €	0	1
Se devuelve 1,00 €	1	0
Se devuelve 1,50 €	1	1

I	Significado
0	No se imprime ticket
1	Se imprime ticket

c) Definición y codificación de los estados:

Estado	Q_1	Q_0	Significado
q_0	0	0	No tiene crédito
q_1	0	1	Tiene 0,50 € de crédito
q_2	1	0	Tiene 1,00 € de crédito
q_3	1	1	Tiene 1,50 € de crédito

d) Diagrama de estados: (En las transiciones se representan las señales E_1E_0/S_1S_0I)



e) Tabla de excitación del sistema.

E_1	E_0	Q_1	Q_0	Q_1	Q_0	J_1K_1	J_0K_0	S_1	S_0	I
0	0	0	0	0	0	0X	0X	0	0	0
0	0	0	1	0	1	0X	X0	0	0	0
0	0	1	0	1	0	X0	0X	0	0	0
0	0	1	1	1	1	X0	X0	0	0	0
0	1	0	0	0	1	0X	1X	0	0	0
0	1	0	1	1	0	1X	X1	0	0	0
0	1	1	0	1	1	X0	1X	0	0	0
0	1	1	1	X	X	XX	XX	X	X	X
1	0	0	0	X	X	XX	XX	X	X	X
1	0	0	1	0	0	0X	X1	0	1	0
1	0	1	0	0	0	X1	0X	1	0	0
1	0	1	1	0	0	X1	X1	1	1	0
1	1	0	0	X	X	XX	XX	X	X	X
1	1	0	1	0	0	0X	X1	0	0	1
1	1	1	0	0	0	X1	0X	0	0	1
1	1	1	1	0	0	X1	X1	0	0	1

f) Implementación de las salidas y excitaciones de los biestables.

		Q_1Q_0			
		00	01	11	10
E_1E_0	00	0	0	X	X
	01	0	1	X	X
	11	X	0	X	X
	10	X	0	X	X

		Q_1Q_0			
		00	01	11	10
E_1E_0	00	X	X	0	0
	01	X	X	X	0
	11	X	X	1	1
	10	X	X	1	1

		Q_1Q_0			
		00	01	11	10
E_1E_0	00	0	X	X	0
	01	1	X	X	1
	11	X	X	X	0
	10	X	X	X	0

		Q_1Q_0			
		00	01	11	10
E_1E_0	00	X	0	0	X
	01	X	1	X	X
	11	X	1	1	X
	10	X	1	1	X

$$J_1 = \overline{E_1}E_0Q_0$$

$$K_1 = E_1$$

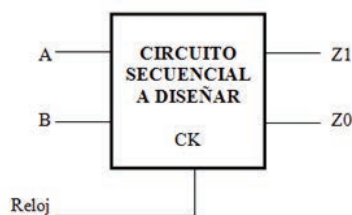
$$J_0 = \overline{E_1}E_0$$

$$K_0 = E_1 + E_0$$

19) En la figura adjunta, A y B representan dos líneas serie por la que entran bits de forma síncrona con el *Reloj*. Ambas líneas proceden de una misma fuente generadora de datos, por lo que deberían ser siempre iguales. Sin embargo, existen errores en la transmisión, de modo que puede ocurrir que las entradas A y B sean diferentes en algún momento. Se trata de diseñar un circuito secuencial que detecte dichos errores y avise al operador cuando el número de errores supere un cierto valor. Para ello se dispone de dos Leds de salida $Z1$ (ámbar) y $Z0$ (rojo) que advierten al operador de la gravedad de la situación de error.

Inicialmente y mientras no haya errores, tanto $Z1$ como $Z0$ están a apagados. En el momento que se haya producido algún error, la luz ámbar queda encendida permanentemente como señal de aviso leve. Cuando el número de errores llegue a ser 3, la luz roja se pondrá a 1 permanentemente como señal de aviso grave. El operario volverá a inicializar el sistema de Leds a través de las entradas asíncronas del sistema, por lo tanto, no considerar la acción del operario como una entrada externa al autómata de estados. La acción de reinicio del sistema por el operario no es objeto del problema a resolver.

Utilizar biestables de tipo D (MSB) para el biestable de más peso y T para el resto de biestables.



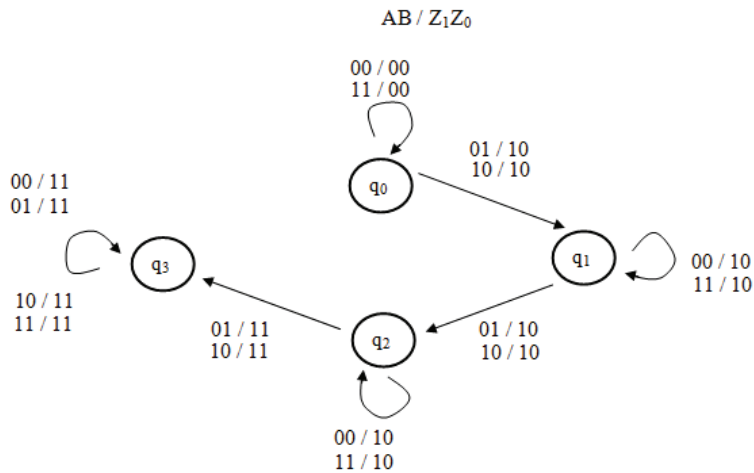
Se pide:

- Definición y codificación de estados y diagrama de estados.
- Tabla de excitación del sistema.
- Implementación de las entradas a biestables. Para el biestable D usar DEC con salida activa a nivel bajo y el resto usar MUX 4x1 para cada una.
- Implementación de las salidas: puertas lógicas para $Z1$ y puertas NOR para $Z0$.

a) Definición y codificación de estados:

Estado	Definición y codificación de estados	Q_1	Q_0
q_0	Inicial, sin errores	0	0
q_1	1 error	0	1
q_2	2 errores	1	0
q_3	3 o más errores	1	1

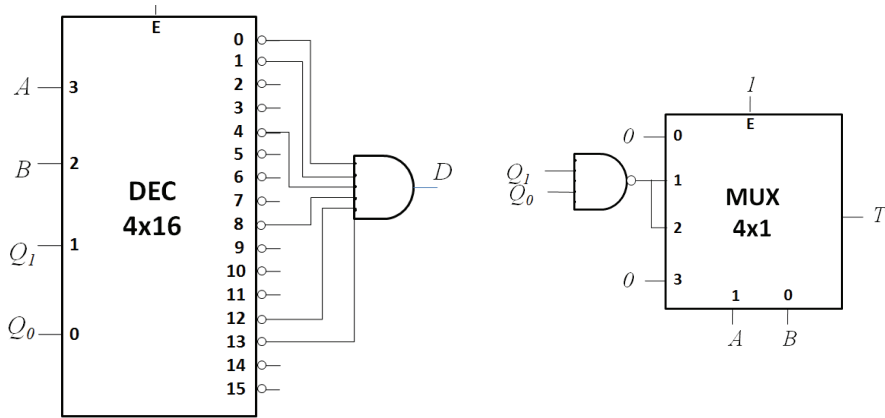
Diagrama de estados:



b) Tabla de excitación del sistema:

[illegible]

c) Implementación de D con DEC y T con MUX 4x1:



d) Implementación de salidas:

Tomando los dos ceros:

$$Z_1 = (A + B + Q_1 + Q_0)(\bar{A} + \bar{B} + Q_1 + Q_0)$$

O agrupando unos:

		$Q_1 Q_0$			
	Z_1	00	01	11	10
$A B$	00	0	1	1	1
	01	1	1	1	1
	11	0	1	1	1
	10	1	1	1	1

$$Z_1 = Q_1 + Q_0 + \bar{A} \cdot B + A \cdot \bar{B} = Q_1 + Q_0 + A \oplus B$$

		$Q_1 Q_0$			
Z_0		00	01	11	10
00		0	0	1	0
01	$A \ B$	0	0	1	1
11		0	0	1	0
10		0	0	1	1

$$\begin{aligned}
 Z_0 &= Q_1 \cdot (A + B + Q_0) \cdot (\bar{A} + \bar{B} + Q_0) = \\
 &= \overline{Q_1 + (A + B + Q_0) + (\bar{A} + \bar{B} + Q_0)}
 \end{aligned}$$

- 20) Un sistema de empaquetado de quesos puede preparar dos tipos de paquetes de forma automática: el paquete Tipo 1 contienen 2 quesos grandes mientras que el Tipo 2 contiene 3 quesos pequeños. El sistema de automatizado consiste en una cinta transportadora con compartimentos estancos donde los quesos vienen colocados al azar, por lo tanto, la secuencia del tipo de queso en la cinta transportadora es aleatoria. A la entrada del sistema de empaquetado existe una cámara que detecta, en cada ciclo de reloj, el tipo de queso que está pasando a la etapa de empaquetado. La empaquetadora es capaz de empaquetar automáticamente si encuentra consecutivos 2 quesos grandes o 3 quesos pequeños.

Diseño del Circuito A

Se desea diseñar un circuito secuencial, que llamaremos circuito A, que proporcionen dos señales digitales G y P de modo que cuando sean activas (sean 1), activen los sistemas de empaquetado de Tipo 1 ó Tipo 2, según el caso.

- Proporcionar una especificación del circuito A, indicando claramente las entradas, salidas y estados definidos. De acuerdo con lo anterior, realizar el diagrama de estados y salidas, indicando la codificación tomada. Los estados deben identificarse como $S0, S1, \dots$ etc.
- Proporcionar la tabla de excitación del circuito utilizando biestables T. Ordenar las variables en la tabla poniendo primero la entrada externa y después las variables de estado.
- Realizar el diseño con decodificadores y puertas NOR de 3 entradas.

Diseño del Circuito B

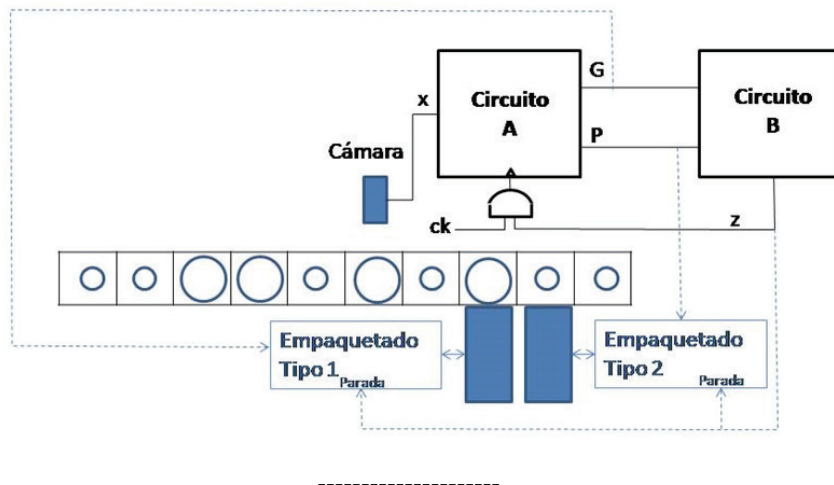
Se quiere completar el circuito anterior con otro circuito, que llamaremos Circuito B, de modo que la empaquetadora tenga dos displays donde se pueda ver el número de paquetes realizados en cada momento de cada tipo. Además, se quiere

activar un sistema luminoso de tres luces de semáforo que compare los números de paquetes de los dos tipos a medida que se van realizando. La especificación del semáforo es la siguiente: luces verdes (*V*), azul (*A*) y roja (*R*) para los respectivos casos: número de paquetes Tipo 1 menor, igual y mayor que número de paquetes de Tipo 2. Además, se debe generar una salida *z* que detenga el sistema de empaquetado y que bloquee el circuito A, cuando alguna de las dos cuentas de paquetes llega a diez.

Para diseñar el circuito B se puede elegir cualquier tipo de módulo combinacional y secuencial y sólo algunas puertas accesorias que sean necesarias. Tener en cuenta que el circuito B tiene por entradas las salidas del circuito A. Describir en primer lugar los módulos que se van a utilizar y su función en el circuito B. Después, realizar el esquema lógico del circuito de forma clara.

Recomendaciones:

- Centrarse en la parte del diseño de los circuitos digitales que se piden (los marcados en trazo continuo), abstrayéndose de otros detalles del sistema.
- Tener en cuenta que el diseño de los circuitos A y B son totalmente independientes.



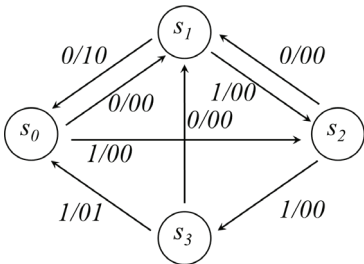
SOLUCIÓN CIRCUITO A

a) Definición de las entradas, salidas y estados:

Codificación Q ₁ Q ₀	
s0: Inicial	0 0
s1: El último queso que llegó fue grande	0 1
s2: El último queso que llegó fue pequeño	1 0
s3: Los dos últimos quesos que llegaron fueron pequeños	1 1

Entradas			Salidas		
X	0	Queso Grande	G	0	Lo contrario a G=1
				1	Se detectan dos quesos grandes seguidos
	1	Queso Pequeño	P	0	Lo contrario a P=1
				1	Se detectan tres quesos pequeños seguidos

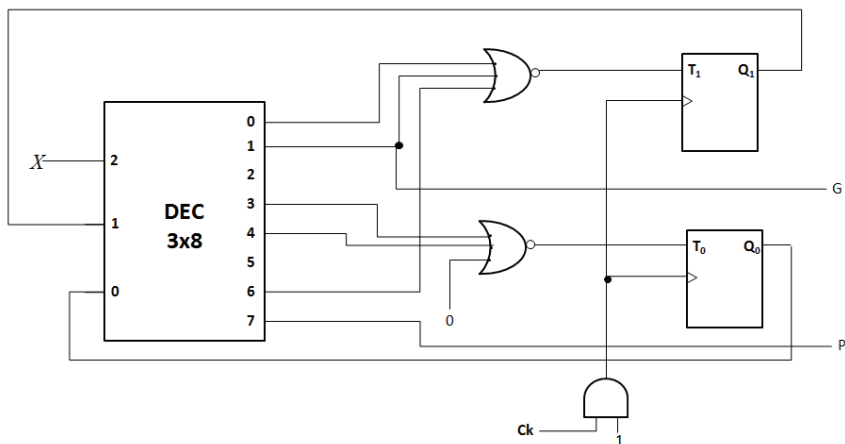
Diagrama de estados:



b) Tabla de excitación:

Entrada	Estado actual		Próximo estado		Excita. biestable		Salidas	
x	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	T_1	T_0	G	P
0	0	0	0	1	0	1	0	0
0	0	1	0	0	0	1	1	0
0	1	0	0	1	1	1	0	0
0	1	1	0	1	1	0	0	0
1	0	0	1	0	1	0	0	0
1	0	1	1	0	1	1	0	0
1	1	0	1	1	0	1	0	0
1	1	1	0	0	1	1	0	1

c) Diseño del circuito A:

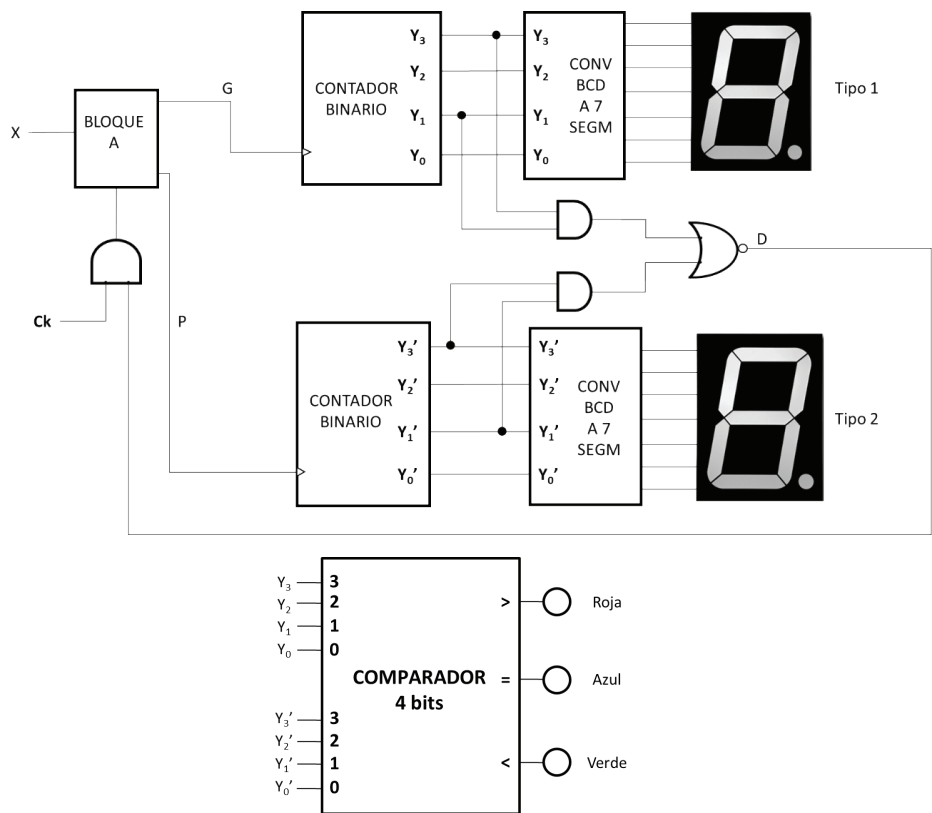


SOLUCIÓN CIRCUITO B

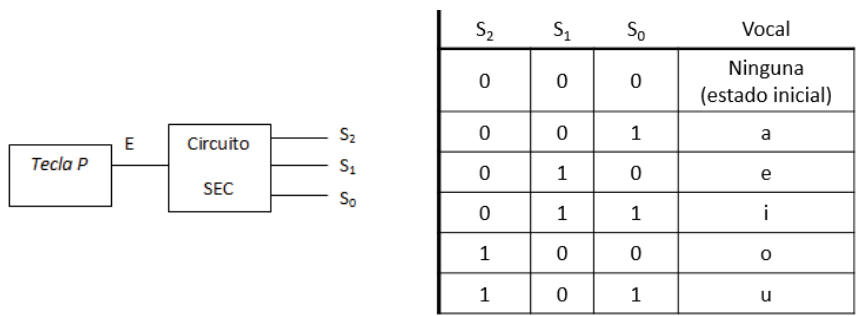
Diseño del circuito B.

La señal D se activa cuando alguna cuenta es 10.

$$D = \overline{Y_3 Y_1} + Y_3 ' Y_1 '$$



21) La Figura muestra un circuito secuencial a diseñar al que se le introduce la entrada E , y que tiene 3 salidas, S_2 , S_1 y S_0 , que codifican las cinco vocales castellanas según la siguiente tabla:



El circuito funciona de la siguiente forma. Cuando se inicializa con una señal asíncrona ajena a nuestro diseño, no se codifica ninguna vocal ($S_2S_1S_0=000$). Mientras no se pulsa la *Tecla P*, $E=0$ y se mantiene dicha situación inicial. Cada vez que se pulsa la tecla un tiempo determinado, $E=1$, y se cambia a la siguiente vocal en orden alfabético ($a \rightarrow e \rightarrow i \rightarrow o \rightarrow u$). Si en cualquier vocal no se pulsa la *Tecla P*, se mantiene dicha vocal. Una vez alcanzada la letra *u*, si se pulsa la *Tecla P* se vuelve al estado de inicial, cuya salida es $S_2S_1S_0=000$.

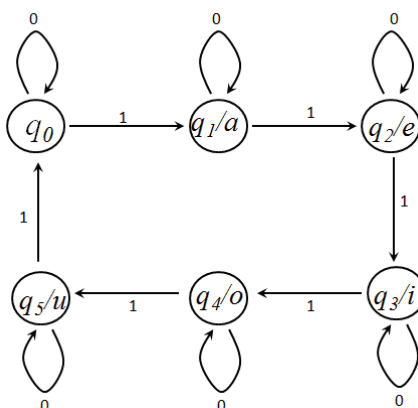
Se pide:

- Definición y codificación de estados.
- Diagrama de estados.
- Tabla de verdad usando un biestable D para el más significativo y biestables T para el resto.
- Implementación con puertas NAND para la excitación del biestable D y puertas NOR para el resto de excitación de biestables.

- Definición y codificación de estados:

	Q_2	Q_1	Q_0
q_0 : Inicial	0	0	0
q_1 : Letra a	0	0	1
q_2 : Letra e	0	1	0
q_3 : Letra i	0	1	1
q_4 : Letra o	1	0	0
q_5 : Letra u	1	0	1

- Diagrama de estados.



c) Tabla de excitación del sistema:

$S_2 \ S_1 \ S_0$				D_2				
E	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	T_1	T_0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0
0	0	1	0	0	1	0	0	0
0	0	1	1	0	1	1	0	0
0	1	0	0	1	0	0	0	0
0	1	0	1	1	0	1	0	0
0	1	1	0	X	X	X	X	X
0	1	1	1	X	X	X	X	X
1	0	0	0	0	0	1	0	1
1	0	0	1	0	1	0	1	1
1	0	1	0	0	1	1	0	1
1	0	1	1	1	0	0	1	1
1	1	0	0	1	0	1	0	1
1	1	0	1	0	0	0	0	1
1	1	1	0	X	X	X	X	X
1	1	1	1	X	X	X	X	X

d) Implementación con puertas NAND y NOR.

	$Q_1 Q_0$			
D_2	00	01	11	10
00	0	0	0	0
01	1	1	X	X
11	1	0	X	X
10	0	0	1	0

	$Q_1 Q_0$			
T_1	00	01	11	10
00	0	0	0	0
01	0	0	X	X
11	0	0	X	X
10	0	1	1	0

	$Q_1 Q_0$			
T_0	00	01	11	10
00	0	0	0	0
01	0	0	X	X
11	1	1	X	X
10	1	1	1	1

$$D_2 = \overline{E}Q_2 + Q_2\overline{Q_0} + EQ_1Q_0;$$

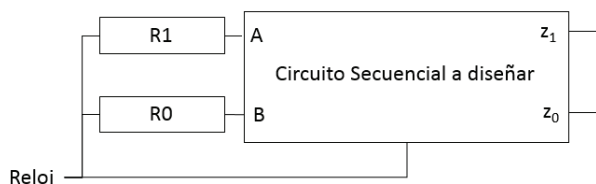
$$D_2 = \overline{\overline{E}Q_2 \cdot Q_2\overline{Q_0} \cdot EQ_1Q_0};$$

$$T_1 = E \cdot \overline{Q_2} \cdot Q_0;$$

$$T_1 = \overline{E + Q_2 + Q_0};$$

$$T_0 = E$$

- 22) Se trata de diseñar un circuito secuencial que admite como entradas dos líneas *A* y *B*, procedentes de sendos registros de desplazamientos, de forma sincronizada con un reloj, por las que entran 1 bit por cada línea en cada ciclo de reloj.



En cada momento se debe ver en la salida *z1 z0*, codificado en binario, el número de ceros que faltan por introducir hasta que éste (el número de ceros introducidos) sea un múltiplo de 4. Suponer que *z1* es la línea más significativa de la salida y que el estado inicial, es decir, cuando todavía no se ha introducido ningún cero, es el mismo que el estado correspondiente a introducir un número de ceros que sea múltiplo de 4.

Se pide:

- Definición de estados, codificación de entradas, estados y salidas.
- Implementación del circuito usando biestables T.
- Si se usasen biestables D, ¿qué relación algebraica habría entre la excitación de dichos biestables y las salidas del circuito?

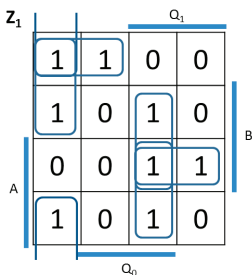
- Definición de estados y codificación de entradas.

Codificación	
q0: inicial, falta por introducir 0 ceros	0 0
q1: falta por introducir 1 cero	0 1
q2: falta por introducir 2 ceros	1 0
q3: falta por introducir 3 ceros	1 1

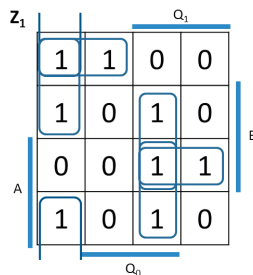
b) Tabla de excitación e implementación del circuito con puertas lógicas:

A	B	Q ₁	Q ₀	Q ₁ /D ₁	Q ₀ /D ₀	Z ₁	Z ₀	T ₁	T ₀
0	0	0	0	1	0	1	0	1	0
0	0	0	1	1	1	1	1	1	0
0	0	1	0	0	0	0	0	1	0
0	0	1	1	0	1	0	1	1	0
0	1	0	0	1	1	1	1	1	1
0	1	0	1	0	0	0	0	0	1
0	1	1	0	0	1	0	1	1	1
0	1	1	1	1	0	1	0	0	1
1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	1
1	0	1	0	0	1	0	1	1	1
1	0	1	1	1	0	1	0	0	1
1	1	0	0	0	0	0	0	0	0
1	1	0	1	0	1	0	1	0	0
1	1	1	0	1	0	1	0	0	0
1	1	1	1	1	1	1	1	0	0

Mapas:



$$Z_1 = \bar{A}\bar{B}\bar{Q}_1 + \bar{A}\bar{Q}_1\bar{Q}_0 + \bar{B}\bar{Q}_1\bar{Q}_0 + BQ_1Q_0 + AQ_1Q_0 + ABQ_1$$



$$T_1 = \bar{B}\bar{Q}_0 + \bar{A}\bar{B} + \bar{A}\bar{Q}_0$$

	Q_1			
Z_0	0	1	1	0
	1	0	0	1
	0	1	1	0
A	1	0	0	1
	Q_0			

$$Z_0 = \bar{A}\bar{B}Q_0 + ABQ_0 + \bar{A}B\bar{Q}_0 + \bar{A}\bar{B}\bar{Q}_0$$

	Q_1			
T_0	0	0	0	0
	1	1	1	1
	0	0	0	0
A	1	1	1	1
	Q_0			

$$T_0 = \bar{A}B + A\bar{B} = A \oplus B$$

c) Como se puede observar en la tabla de excitación, si se utilizan biestables D, la excitación de los biestables es igual a la salida del circuito.

- 23) Una máquina de bebidas refrescantes funciona con monedas en euros y en pesetas. Proporciona una lata por valor de 0,80 € o 133 ptas. Sólo admite una moneda de 25, 50 y 100 ptas. y de 0,20 € para adquirir la lata y no devuelve cambio. Si la cantidad introducida es mayor que la necesaria, proporciona la lata, poniendo a 1 la salida S y activa un panel luminoso que pone “Gracias por renunciar al cambio” poniendo a 1 la salida L . Si se introduce el precio exacto, sólo proporciona la lata, sin activar el panel luminoso.

No se puede cambiar el tipo de monedas con el que se empezó (si se comenzó con pesetas, se seguirá con pesetas, y si se comenzó con euros se seguirá con euros) salvo que se alcance el valor de 100 ptas. ó 0,60 €, en cuyo caso se puede cambiar de moneda. Cada vez que entregue una lata, volverá al estado inicial, preparada para funcionar de nuevo.

NOTA: Se recuerda que el valor de 1 € es de 166,386 ptas. y que 100 ptas. son exactamente 0,60 €.

Codificar los estados de forma que el valor numérico que indica el código generado con los biestables de mayor a menor peso sea mayor cuanto mayor sea la cantidad de dinero introducida en la máquina hasta ese momento, independientemente de la moneda introducida.

Se pide:

- Definición y codificación de estados.
- Diagrama de estados.
- Tabla de transición y excitación usando biestables T.
- Circuitaría combinacional adicional que admitiendo como entradas las salidas de los biestables ponga a 1 dos salidas, una P que activa un panel luminoso que pone “PESETAS” y otra E , que activa un panel luminoso que pone “EUROS” indicándole al usuario qué tipo de moneda debe introducir. Es posible que en algún caso puede introducir dos, y se encenderán ambos paneles poniendo a 1

sus respectivas entradas. Implementar P con un decodificador con salidas activas a nivel bajo y una puerta lógica con el mínimo número de entradas. ¿Cuál es la forma más sencilla de obtener E ?

OBSERVACIÓN: el apartado d) se puede realizar sin necesidad de hacer los 3 anteriores.

Solución 1

a) Definición y codificación de estados.

Entradas: hay 4 posibles opciones que se codifican con dos variables binarias A, B .

Entrada	A B
25 ptas	0 0
50 ptas	0 1
100 ptas	1 0
0,20 €	1 1

Salidas:

$S=1$ si la cantidad es mayor o igual que la requerida. Proporciona la lata.

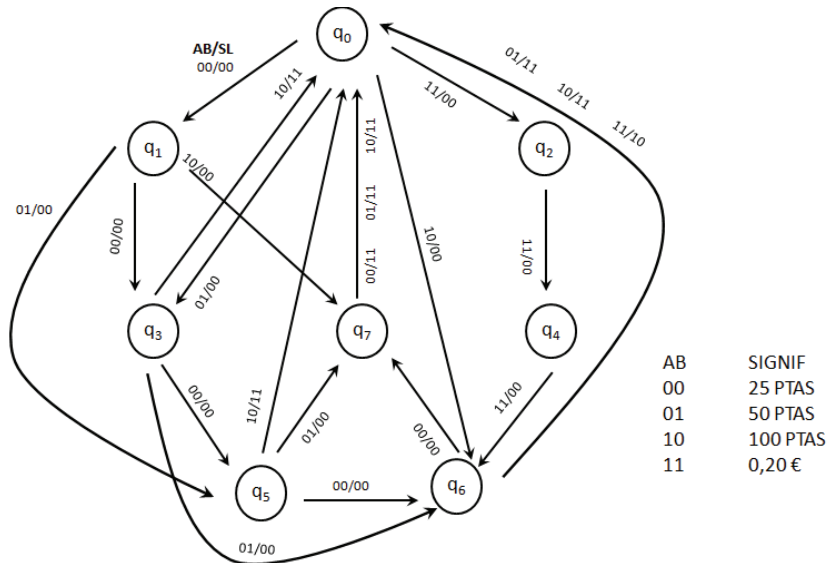
$S=0$ si la cantidad es inferior a la requerida.

$L=1$ si la cantidad es mayor que la requerida. Luminoso encendido.

$L=0$ si la cantidad es igual o menor que la requerida.

ESTADO	CODIFICACIÓN			DEFINICIÓN
	$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	
q_0	0	0	0	Inicial. Final. No tiene dinero
q_1	0	0	1	Tiene 25 ptas (0,15€)
q_2	0	1	0	Tiene 0,20€ (33 ptas)
q_3	0	1	1	Tiene 50 ptas (0,30€)
q_4	1	0	0	Tiene 0,40€ (66 ptas)
q_5	1	0	1	Tiene 75 ptas (0,45€)
q_6	1	1	0	Tiene 100 ptas (0,60€)
q_7	1	1	1	Tiene 125 ptas (0,75€)

b) Diagrama de estados.

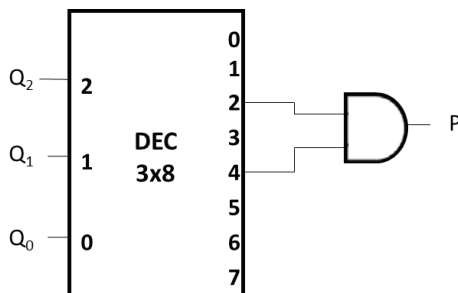


c) Tabla de excitación y transición:

A	B	$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$	T_2	T_1	T_0	S	L
0	0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	0	1	0	1	1	0	1	0	0	0
0	0	0	1	0	X	X	X	X	X	X	X	X
0	0	0	1	1	1	0	1	1	1	0	0	0
0	0	1	0	0	X	X	X	X	X	X	X	X
0	0	1	0	1	1	1	0	0	1	1	0	0
0	0	1	1	0	1	1	1	0	0	1	0	0
0	0	1	1	1	0	0	0	1	1	1	1	1
0	1	0	0	0	0	1	1	0	1	1	0	0
0	1	0	0	1	1	0	1	1	0	0	0	0
0	1	0	1	0	X	X	X	X	X	X	X	X
0	1	0	1	1	1	1	0	1	0	1	0	0
0	1	1	0	0	X	X	X	X	X	X	X	X
0	1	1	0	1	1	1	1	0	1	0	0	0
0	1	1	1	0	0	0	0	1	1	0	1	1
0	1	1	1	1	0	0	0	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	0	0	0
1	0	0	0	1	1	1	1	1	1	0	0	0
1	0	0	1	0	X	X	X	X	X	X	X	X
1	0	0	1	1	0	0	0	0	1	1	1	1
1	0	1	0	0	X	X	X	X	X	X	X	X
1	0	1	0	1	0	0	0	1	0	1	1	1
1	0	1	1	0	0	0	0	1	1	1	1	1
1	0	1	1	1	0	0	0	1	1	1	1	1
1	1	0	0	0	0	1	0	0	1	0	0	0
1	1	0	0	1	X	X	X	X	X	X	X	X
1	1	0	1	0	1	0	0	1	1	0	0	0
1	1	0	1	1	X	X	X	X	X	X	X	X
1	1	1	0	0	1	1	0	0	1	0	0	0
1	1	1	0	1	X	X	X	X	X	X	X	X
1	1	1	1	0	0	0	0	1	1	0	1	0
1	1	1	1	1	X	X	X	X	X	X	X	X

d) Circuitería combinacional adicional para las funciones P y E :

Q_2	Q_1	Q_0	P	E
0	0	0	1	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	1	0



$$E = \bar{Q}_0$$

24) Diseñar un reconocedor de secuencias que permita reconocer alguna de las tres secuencias:

Secuencia 1: 0101, Secuencia 2: 1000, Secuencia 3: 110

Nota: En todos los casos, el bit escrito a la izquierda entra antes (en el tiempo) que el bit escrito a la derecha.

Al circuito le llegará un bit sincronizado con un reloj por una línea de entrada E . La salida serán dos líneas Z_1 y Z_0 que indiquen si se ha reconocido o no alguna de las tres secuencias y se codificará de la siguiente manera:

$Z_1Z_0 = 00$	No se ha reconocido (todavía) ninguna de las tres secuencias
$Z_1Z_0 = 01$	Se ha reconocido la secuencia 1
$Z_1Z_0 = 10$	Se ha reconocido la secuencia 2
$Z_1Z_0 = 11$	Se ha reconocido la secuencia 3



Suponer que el sistema parte de un estado inicial que corresponde al caso en que todavía no se ha introducido ningún bit. Suponer también que siendo reconocida cualquier secuencia, los últimos bits introducidos se tienen en cuenta para reconocer

otra posible secuencia, es decir, el sistema no se inicializa, excepto si la secuencia reconocida es la 3, en cuyo caso sí se inicializa.

Se pide:

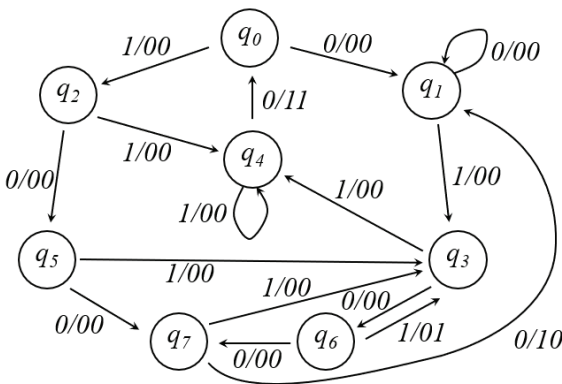
- a) Definir claramente los estados y su codificación.
- b) Dibujar el diagrama de transición de estados.
- c) Tabla de transición y excitación, usando biestables T para el biestable más significativo y D para el resto.
- d) Expresiones booleanas de las funciones de salida.
- e) Implementar la excitación de los biestables con MUX a discreción.

a) Definición y codificación de estados:

Estado	Codificación			Definición
	$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	
q_0	0	0	0	Inicial
q_1	0	0	1	Último bit 0
q_2	0	1	0	Último bit 1
q_3	0	1	1	Dos últimos bits 01 (1 último y 0 penúltimo)
q_4	1	0	0	Dos últimos bits 11
q_5	1	0	1	Dos últimos bits 10 (0 último y 1 penúltimo)
q_6	1	1	0	Tres últimos bits 010
q_7	1	1	1	Tres últimos bits 100 (0 último)

La definición de estados podría haber sido otra similar.

b) Diagrama de estados:



c) Tabla de transición, excitación y salidas:

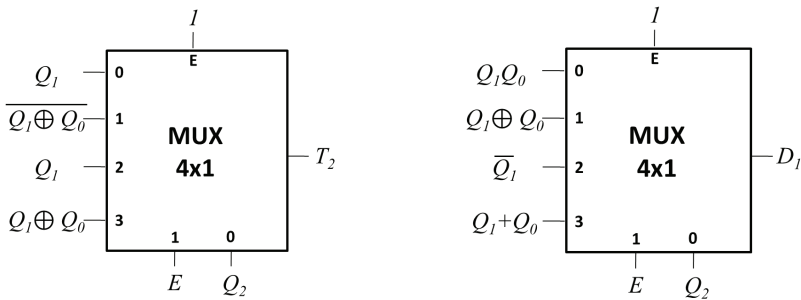
E	$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$	T_2	D_1	D_0	Z_1	Z_0
0	0	0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	1	0	0	1	0	0
0	0	1	0	1	0	1	1	0	1	0	0
0	0	1	1	1	1	0	1	1	0	0	0
0	1	0	0	0	0	0	1	0	0	1	1
0	1	0	1	1	1	1	0	1	1	0	0
0	1	1	0	1	1	1	0	1	1	0	0
0	1	1	1	0	0	1	1	0	1	1	0
1	0	0	0	0	1	0	0	1	0	0	0
1	0	0	1	0	1	1	0	1	1	0	0
1	0	1	0	1	0	0	1	0	0	0	0
1	0	1	1	1	0	0	1	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	0
1	1	0	1	0	1	1	1	1	1	0	0
1	1	1	0	0	1	1	1	1	1	0	1
1	1	1	1	0	1	1	1	1	1	0	0

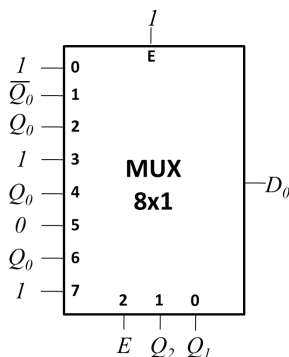
d) Expresiones algebraicas de las funciones de salida:

$$Z_1 = \bar{E}Q_2(\overline{Q_1 \oplus Q_0})$$

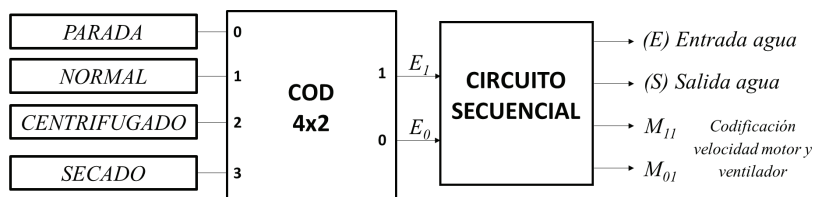
$$Z_0 = \bar{Q_2}Q_0(\overline{E \oplus Q_1})$$

e) Implementación de las funciones de excitación de los biestables:





- 25) Diseñar la parte secuencial de la circuitería lógica de una lavadora de acuerdo con las siguientes especificaciones. Hay 4 botones llamados PARADA, que detiene el funcionamiento de la lavadora, dejándola en reposo, NORMAL, que permite que funcione normalmente, CENTRIFUGADO, que consiste en que el motor gire rápidamente, y finalmente SECADO, que consiste en que el motor gire rápidamente junto con un ventilador adicional para secar la ropa. Cuando la lavadora está parada no se permite la entrada ni la salida de agua.



Nunca se pueden pulsar a la vez más de un botón de los 4 descritos anteriormente. Cuando la lavadora está parada, se puede poner en marcha pulsando el botón NORMAL y en este caso se permitirá la entrada de agua, pero no la salida de la misma y el motor girará a velocidad lenta. Cuando el usuario lo considere oportuno, estando en este modo, pulsará el botón CENTRIFUGADO, se cierra la entrada de entrada de agua, se abre la salida de agua, y el giro del motor a velocidad rápida. Para detener el motor desde CENTRIFUGADO, se debe pulsar el botón PARADA, se parará el motor y no será posible la entrada ni la salida de agua.

Por otro lado, desde PARADA se puede directamente pasar a SECADO, pulsando el botón correspondiente. En este caso, se cortará la entrada y salida de agua, la velocidad del motor será rápida y se encenderá el ventilador. Para finalizar SECADO, se debe pulsar la tecla PARADA.

La codificación de la velocidad del motor y funcionamiento del ventilador se describen en la siguiente tabla:

M_1	M_0	Velocidad del motor	Ventilador
0	0	Parado	Parado
0	1	Lento	Parado
1	0	Rápido	Parado
1	1	Rápido	Funcionando

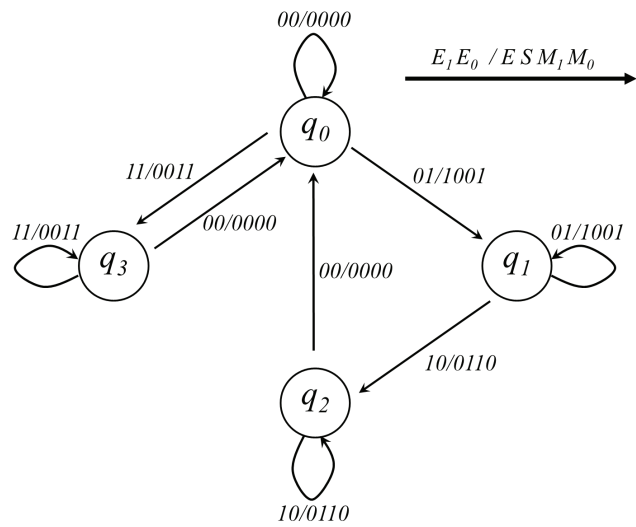
Se pide:

- Definir y codificar los estados del circuito secuencial.
- Realizar el diagrama de estados.
- Obtener la tabla de verdad completa que permite diseñar el circuito usando biestables T.
- Se quiere diseñar una circuitería combinacional adicional que tomando como entrada la salida de los biestables, sirva para encender unas luces que hay dentro de cada botón de entrada. Las luces que se encenderán (poniendo a 1 la función booleana correspondiente) serán sólo aquellas que correspondan a los botones que en cada momento se puedan pulsar. Ejemplo, cuando está en reposo, sólo se encenderán las luces *N* y *S*, porque desde el reposo sólo se puede comenzar a lavar Normal o Secado. Llamar a las funciones booleanas de las luces con la letra inicial de la palabra correspondiente.
- Diseñar la circuitería adicional de un indicador luminoso que se active (puesto a 1) cuando el motor está en marcha, independientemente de la velocidad y otro que se active (puesto a 1) cuando funcione el ventilador, tomando como entradas en ambos casos *M1* y *M0*.

- Definición y codificación de estados

Definición de estados		Codificación de estados	
		Q_1	Q_0
q_0	Reposo	0	0
q_1	Normal	0	1
q_2	Centrifugado	1	0
q_3	Secado	1	1

b) Diagrama de estados:



c) Tabla de excitación del sistema:

E_1	E_0	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	E	S	M_1	M_0	T_1	T_0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	X	X	X	X	X	X	X	X
0	0	1	0	0	0	0	0	0	0	1	0
0	0	1	1	0	0	0	0	0	0	1	1
0	1	0	0	0	1	1	0	0	1	0	1
0	1	0	1	0	1	1	0	0	1	0	0
0	1	1	0	X	X	X	X	X	X	X	X
0	1	1	1	X	X	X	X	X	X	X	X
1	0	0	0	X	X	X	X	X	X	X	X
1	0	0	1	1	0	0	1	1	0	1	1
1	0	1	0	1	0	0	1	1	0	0	0
1	0	1	1	X	X	X	X	X	X	X	X
1	1	0	0	1	1	0	0	1	1	1	1
1	1	0	1	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X
1	1	1	1	1	1	0	0	1	1	0	0

d) Indicadores luminosos de los botones:

$Q_1(t)$	$Q_0(t)$	P	N	C	S
0	0	0	1	0	1
0	1	0	0	1	0
1	0	1	0	0	0
1	1	1	0	0	0

$$P = Q_1(t) \quad N = S = \overline{Q_1(t)} + \overline{Q_0(t)} \quad C = \overline{Q_1(t)} Q_0(t)$$

e) Indicadores luminosos adicionales del ventilador y el motor:

M_1	M_0	M	V
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

$$M = M_1 + M_0 \quad V = M_1 M_0$$

26) Diseñar un circuito secuencial utilizando FF-JK que dispone de una entrada x , y dos salidas y_1 e y_2 que indican si la secuencia de entrada es menor que, igual a o mayor que la representación binaria de 1/3.

- Definición del mínimo número de estados.
- Diagrama de estados.
- Tabla de estado.
- Tabla de excitación.
- Expresiones simplificadas de las entradas a biestables y salidas.

Adoptar la siguiente codificación:

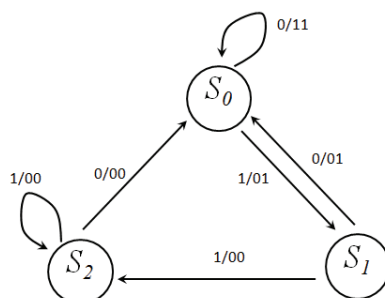
y_1	y_2	Definición
0	0	La secuencia de entrada es $>1/3$
0	1	La secuencia de entrada todavía coincide con 1/3
1	1	La secuencia de entrada es $<1/3$

Teniendo en cuenta que el valor binario de $1/3$ es $0,010101\dots$

a) Definición del mínimo número de estados:

S_0	Se ha introducido un 0 en el instante t-1
S_1	Se ha introducido 01 en los inst. t-2 y t-1
S_2	Ninguno de los anteriores

b) Diagrama de estados:



c) Tabla de estado.

	Q_1	Q_0
S_0	0	0
S_1	0	1
S_2	1	0

d) Tabla de excitación.

Entradas a C.C.			Estado Siguiente		Entradas a los FF				Salidas	
x	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	J_1	K_1	J_0	K_0	Y_1	Y_2
0	0	0	0	0	0	X	0	X	1	1
0	0	1	0	0	0	X	X	1	0	1
0	1	0	0	0	X	1	0	X	0	0
0	1	1	X	X	X	X	X	X	X	X
1	0	0	0	1	0	X	1	X	0	1
1	0	1	1	0	1	X	X	1	0	0
1	1	0	1	0	X	0	0	X	0	0
1	1	1	X	X	X	X	X	X	X	X

e) Expresiones simplificadas de las entradas a biestables y salidas:

J_1			Q_1	
	0	0	-	-
x	0	1	-	-
			Q_0	

$$J_1 = xQ_0$$

K_1			Q_1	
	-	-	-	1
x	-	-	-	0
			Q_0	

$$K_1 = \bar{x}$$

J_0			Q_1	
	0	-	-	0
x	1	-	-	0
			Q_0	

$$J_0 = x\bar{Q}_1$$

K_0			Q_1	
	-	1	-	-
x	-	1	-	-
			Q_0	

$$K_0 = 1$$

Y_1			Q_1	
	1	0	-	0
x	0	0	-	0
			Q_0	

$$Y_1 = \bar{x}\bar{Q}_0\bar{Q}_1 = \bar{x} + Q_0 + Q_1$$

Y_2			Q_1	
	1	1	-	0
x	1	0	-	0
			Q_0	

$$Y_2 = \bar{x}\bar{Q}_1 + \bar{Q}_0\bar{Q}_1$$

27) Un circuito secuencial síncrono dispone de dos entradas A y B por las que recibe en forma serie dos números binarios de n bits (los dos primeros bits en llegar por cada entrada son los menos significativos), y una salida z que va dando en forma de serie la suma aritmética de los dos números. Se dispone de biestables ficticios tipo PR; su funcionamiento es tal que sólo cambia de estado cuando sus entradas P y R son distintas.

- Proporcionar la tabla de transición del biestable PR. A partir de ella deducir la tabla de excitación suponiendo que R se mantiene a cero.
- Diseñar el circuito secuencial especificado, tomando la entrada R a cero. Proporcionar:
 - Especificación del sistema definiendo con claridad cuáles son los estados considerados.
 - Diagrama de estado
 - Diseño

c) Construir el biestable ficticio PR a partir de un JK.

a) Tabla de transición y excitación:

Tabla de Transición			
P	R	Q	$Q_{(t+1)}$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Si $R=0$ la tabla quedaría:

P	R	Q	$Q_{(t+1)}$
0	0	0	0
0	0	1	1
1	0	0	1
1	0	1	0

De ahí se deduce:

Tabla de Excitación			
Q_t	$Q_{(t+1)}$	P	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	0

b) Lo único que necesitamos saber de este sistema es si tiene un bit de acarreo proveniente de la suma anterior: consideramos dos estados.

S_0 = El bit de arrastre anterior es 0

S_1 = El bit de arrastre anterior es 1

Definición y codificación de estados:

Estado Presente S	Entradas XY			
	00	01	10	11
S_0	$S_0/0$	$S_0/1$	$S_0/1$	$S_1/0$
S_1	$S_0/1$	$S_1/0$	$S_1/0$	$S_1/1$

Estado S / Salida

Diagrama de estados:

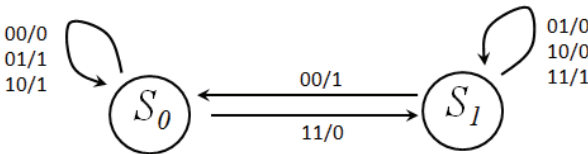


Tabla de excitación:

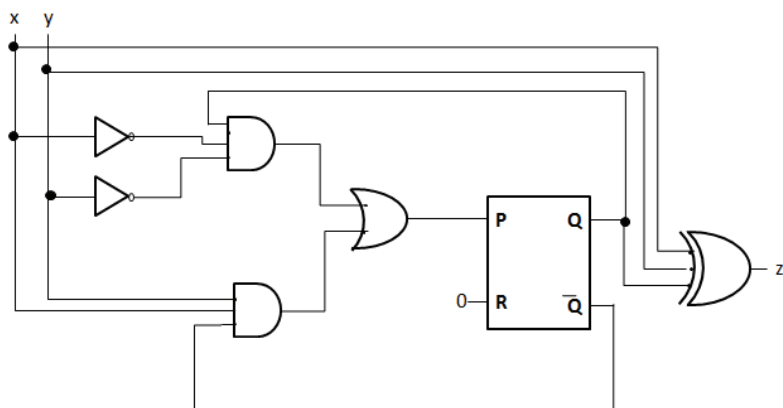
Entradas			Estado Siguiente	Entradas a los FF		Salida
X	Y	Q		P	R	
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	0	0	1
0	1	1	1	0	0	0
1	0	0	0	0	0	1
1	0	1	1	0	0	0
1	1	0	1	1	0	0
1	1	1	1	0	0	1

$$P = \bar{x} \bar{y} Q + xy \bar{Q}$$
$$R = 0$$

	Z		Y	
	0	1	0	1
x	1	0	1	0
			Q	

$$z = x \oplus y \oplus Q$$

Diseño del circuito:



c) Construcción del biestable ficticio PR:

P	R	Q	$Q_{(t+1)}$	J	K
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	1	1	X
0	1	1	0	X	1
1	0	0	1	1	X
1	0	1	0	X	1
1	1	0	0	0	X
1	1	1	1	X	0

a) Análisis del circuito:

$$D_A = E$$

$$D_B = Q_A$$

$$D_C = Q_B$$

$$Z = \overline{Q_A} Q_B Q_C + Q_A \overline{Q_B} Q_C + Q_A Q_B \overline{Q_C}$$

Para FF tipo D; $Q(t+1) = D(t)$ luego:

$$Q_A(t+1) = E$$

$$Q_B(t+1) = D_B(t) = Q_A(t)$$

$$Q_C(t+1) = D_C(t) = Q_B(t)$$

Entradas al C.C.				Estado Siguierte (t+1)			Salida
E	Q _C	Q _B	Q _A	Q _C	Q _B	Q _A	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0
0	0	1	1	1	1	0	1
0	1	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	1	1	0	1	0	0	1
0	1	1	1	1	1	0	0
1	0	0	0	0	0	1	0
1	0	0	1	0	1	1	0
1	0	1	0	1	0	1	0
1	0	1	1	1	1	1	1
1	1	0	0	0	0	1	0
1	1	0	1	0	1	1	1
1	1	1	0	1	0	1	1
1	1	1	1	1	1	1	0

b) A la vista de la expresión de Z:

$$Z = \overline{Q_A} Q_B Q_C + Q_A \overline{Q_B} Q_C + Q_A Q_B \overline{Q_C}$$

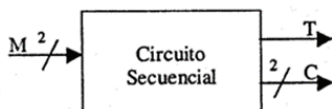
Es obvio que Z es independiente de la entrada E y sólo depende del contenido de los biestables y será 1 siempre que haya dos biestables con contenido 1 y el restante con contenido 0.

Reconoce secuencias de 2 unos en los últimos 3 bits recibidos independientemente del orden, es decir, puede reconocer las secuencias:

t-3	t-2	t-1
0	1	1
1	0	1
1	1	0

- 29) Una máquina expendedora de tabaco está controlada por un circuito secuencial como el de la figura. Dicho circuito recibe como entradas el código de dos bits (tabla 1) de las monedas que va introduciendo el usuario. Como salidas, el circuito debe activar 2 señales, correspondientes a si se debe o no entregar el tabaco (T), y la cantidad de cambio a reintegrar (CI y CO), codificadas según las tablas 2 y 3 respectivamente.

El funcionamiento del circuito es el siguiente. El usuario debe introducir monedas hasta alcanzar la cantidad de 100, que será el precio del tabaco. Una vez que la suma de las monedas alcance dicho importe, automáticamente la máquina servirá el tabaco, retornando además el cambio correspondiente. Por ejemplo, si el usuario ha introducido 1 moneda de 25 y 2 monedas de 50, la máquina debe servirle el paquete y una moneda de 25 (salidas $T\ CI\ 0=101$).



$M1\ M0$	Significado
0 0	No hay nuevas monedas
0 1	Nueva moneda de 25
1 0	Nueva moneda de 50
1 1	Nueva moneda de 100

Tabla 1: Codificación de la entrada M

T	Significado
0	No servir tabaco
1	Servir tabaco

Tabla 2: Codificación de la salida T

$CI\ CO$	Significado
0 0	No hay cambio
0 1	Cambio = 25
1 0	Cambio = 50
1 1	Cambio = 75

Tabla 3: Codificación de la salida

Diseña el circuito secuencial correspondiente, indicando claramente:

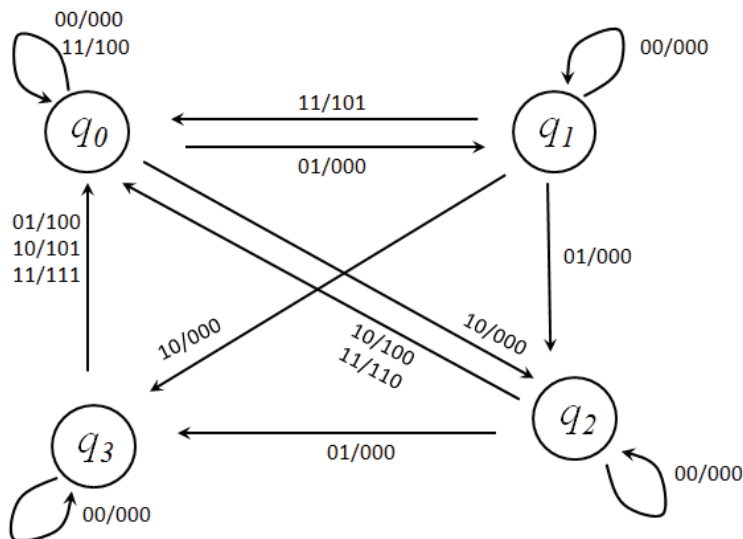
- El número y codificación de los estados
- El diagrama de estados.

- c) La tabla de transiciones y salidas del circuito, utilizando para ello un biestable T (para el bit más significativo de los que almacenan el estado), y biestables JK para el resto.
- d) La implementación del circuito.

a) Definición y codificación de estados:

	Q_1	Q_0	Significado
q_0	0	0	Tiene 0 ptas
q_1	0	1	Tiene 25 ptas
q_2	1	0	Tiene 50 ptas
q_3	1	1	Tiene 75 ptas

b) Diagrama de estados: (en las transiciones se representan las señales $M_i M_0 / TC_i C_0$)



c) Tabla de excitación del sistema:

M_1	M_0	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	T	C_1	C_0	T_1	J_0	K_0
0	0	0	0	0	0	0	0	0	0	0	X
0	0	0	1	0	1	0	0	0	0	X	0
0	0	1	0	1	0	0	0	0	0	0	X
0	0	1	1	1	1	0	0	0	0	X	0
0	1	0	0	0	1	0	0	0	0	1	X
0	1	0	1	1	0	0	0	0	1	X	1
0	1	1	0	1	1	0	0	0	0	1	X
0	1	1	1	0	0	1	0	0	1	X	1
1	0	0	0	1	0	0	0	0	1	0	X
1	0	0	1	1	1	0	0	0	1	X	0
1	0	1	0	0	0	1	0	0	1	0	X
1	0	1	1	0	0	1	0	1	1	X	1
1	1	0	0	0	0	1	0	0	0	0	X
1	1	0	1	0	0	1	0	1	0	X	1
1	1	1	0	0	0	1	1	0	1	0	X
1	1	1	1	0	0	1	1	1	1	X	1

d) Implementación del circuito:

		Q_1Q_0			
	J_0	00	01	11	10
M_1M_0	00	0	X	X	0
	01	1	X	X	1
	11	0	X	X	0
	10	0	X	X	0

		Q_1Q_0			
	K_0	00	01	11	10
M_1M_0	00	X	0	0	X
	01	X	1	1	X
	11	X	1	1	X
	10	X	0	1	X

		Q_1Q_0			
C_1		00	01	11	10
M_1M_0	00	0	0	0	0
	01	0	0	0	0
	11	0	0	1	1
	10	0	0	0	0

		Q_1Q_0			
C_0		00	01	11	10
M_1M_0	00	0	0	0	0
	01	0	0	0	0
	11	0	1	1	0
	10	0	0	1	0

		Q_1Q_0			
T		00	01	11	10
M_1M_0	00	0	0	0	0
	01	0	0	1	0
	11	1	1	1	1
	10	0	0	1	1

$$J_0 = \overline{M_1}M_0$$

$$K_0 = M_0 + M_1Q_1$$

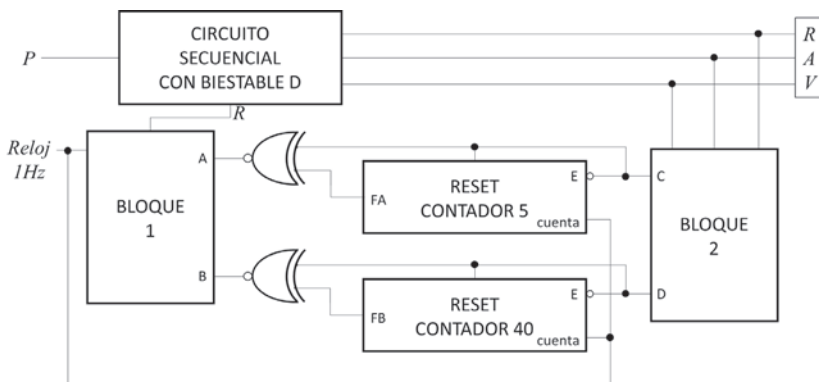
$$C_1 = M_1M_0Q_1$$

$$C_0 = M_1M_0Q_0 + M_1Q_1Q_0$$

$$T = M_1M_0 + M_1Q_1 + M_0Q_1Q_0$$

- 30) Se desea diseñar el circuito de control de un semáforo de peatones, cuyo funcionamiento se describe a continuación.

Mientras no se active el pulsador ($P=0$), el semáforo permanecerá por tiempo indefinido en VERDE. Cuando se pulse P , se encenderá en el siguiente ciclo de reloj la luz AMBAR, sin apagarse la VERDE y transcurridos 5 segundos, se apagará ambas y se encenderá la ROJA durante 40 segundos, finalizados los cuales se volverá a la situación inicial con sólo la luz VERDE encendida. Se supone que P se pulsa sólo cuando esté encendida la luz VERDE, y nunca en los restantes casos. El sistema completo funciona según el esquema de la figura adjunta.



Cuando se encienden las luces VERDE y AMBAR el bloque 2 se pone la línea *C* a nivel 0, mientras que cuando se enciende sólo la luz ROJA pone la línea *D* a nivel 0. Mientras alguna de estas líneas (*C/D*) está a nivel 0, queda inhibida la entrada del reloj al circuito secuencial. Mientras el semáforo esté VERDE, los pulsos de reloj entran al circuito secuencial.

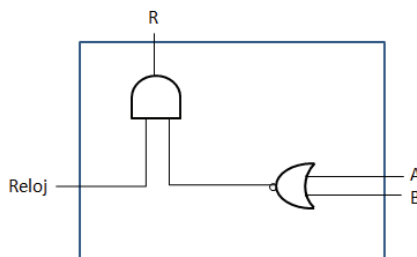
Los contadores tienen una entrada de RESET, activa a nivel alto, y una entrada de HABILITACION (*E*), activa a nivel bajo. Cuando los contadores cuentan el último número previsto presentan en su salida respectiva FA/FB un nivel 1, señalando que se ha llegado al final de la cuenta. NOTA: *R* está conectado a las entradas de reloj de los tres biestables *D* que hay en el circuito secuencial a diseñar.

- ¿Cuál es la implementación más sencilla del bloque 1 considerando que la entrada de reloj al circuito secuencial debe inhibirse ($R=0$) cuando *A* o *B* sean 1?
- Implementar el bloque 2, considerando que cuando estén encendidas las luces VERDE y AMBAR, *C* debe ser 0, y cuando esté encendida la luz ROJA, *D* debe ser 0. Para ello usar sólo una puerta lógica y un inversor.
- Diseñar el circuito secuencial usando 3 biestables tipo *D* con salidas conectadas directamente a las luces del semáforo.

a) Implementación del bloque 1:

A	B	R
0	0	Reloj
0	1	0
1	0	0
1	1	X

$$R = \overline{A} \overline{B} \text{ Reloj} = (\overline{A+B}) \text{ Reloj}$$



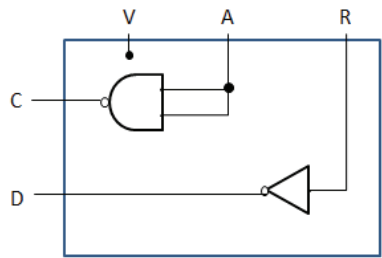
b) Implementación del bloque 2.

V	A	R	C	D
0	0	0	X	X
0	0	1	1	0
0	1	0	X	X
0	1	1	X	X
1	0	0	1	1
1	0	1	X	X
1	1	0	0	1
1	1	1	X	X

		AR				
		C	00	01	11	10
V	0	X	1	X	X	
	1	1	X	X	0	

		AR				
		D	00	01	11	10
V	0	X	0	X	X	
	1	1	X	X	1	

$$C = \overline{A}$$
$$D = \overline{R}$$



c) Diseño del circuito secuencial:

Definición de estados:

Definición Estados		Codificación
Estado	Significado	$Q_V Q_A Q_R$
q0	Verde	1 0 0
q1	Verde y ámbar	1 1 0
q2	Rojo	0 0 1

Diagrama de estados:

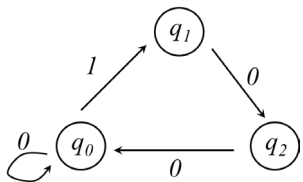
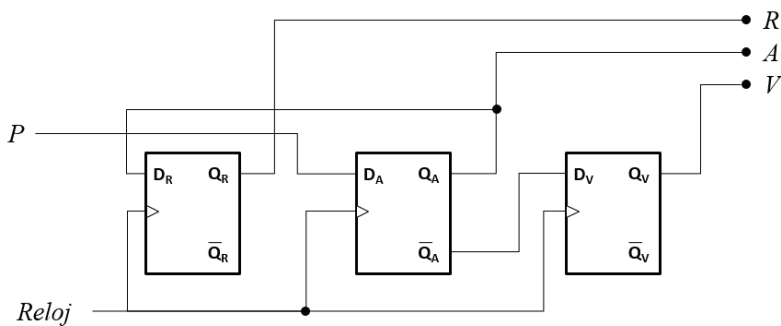


Tabla de verdad:

P	$Q_V(t)$	$Q_A(t)$	$Q_R(t)$	$Q_V(t+1)$	$Q_A(t+1)$	$Q_R(t+1)$	D_V	D_A	D_R
0	0	0	0	X	X	X	X	X	X
0	0	0	1	1	0	0	1	0	0
0	0	1	0	X	X	X	X	X	X
0	0	1	1	X	X	X	X	X	X
0	1	0	0	1	0	0	1	0	0
0	1	0	1	X	X	X	X	X	X
0	1	1	0	0	0	1	0	0	1
0	1	1	1	X	X	X	X	X	X
1	0	0	0	X	X	X	X	X	X
1	0	0	1	X	X	X	X	X	X
1	0	1	0	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X
1	1	0	0	1	1	0	1	1	0
1	1	0	1	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X

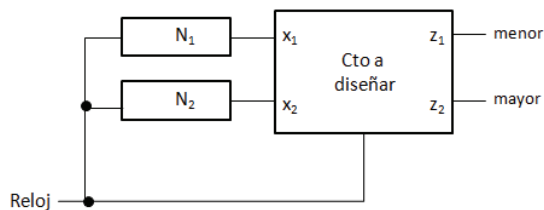
Ecuaciones: $D_V = \bar{Q}_A$ $D_A = P$ $D_R = Q_A$

Diseño del circuito:



31) $R1$ y $R2$ son dos registros de desplazamiento de 8 bits, en cada uno de los cuales están almacenados dos números binarios $N1$ y $N2$ respectivamente. Se trata de diseñar un circuito secuencial síncrono con dos entradas $X1$ y $X2$ por las que entran bit a bit en cada ciclo de reloj los números $N1$ y $N2$, comenzando por el bit más significativo (MSB), y dos salidas $Z1$ y $Z2$ (por $Z1$ sale el menor de los dos números y por $Z2$ el mayor).

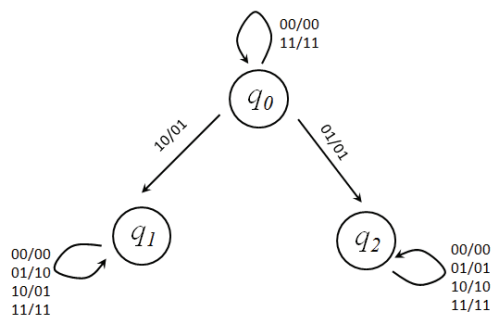
- Definir claramente los estados del sistema y su codificación.
- Encontrar el diagrama de estados y las tablas de transición y excitación usando biestables T.
- Encontrar las ecuaciones de salida $Z1$ y $Z2$ teniendo en cuenta que sólo se dispone de puertas NAND y las ecuaciones de excitación de los biestables para implementar con puertas AND e inversores.



a) Definición y codificación de estados:

Estado	Codificación		Definición
	$Q_1(t)$	$Q_0(t)$	
q_0	0	0	Inicial. Hasta ahora ambos números son iguales
q_1	0	1	N_1 es mayor que N_2
q_2	1	0	N_2 es mayor que N_1

b) Diagrama de estados:



Tablas de excitación y transición:

X_1	X_0	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	Z_1	Z_2	T_1	T_0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	0
0	0	1	1	X	X	X	X	X	X
0	1	0	0	1	0	0	1	1	0
0	1	0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1	0	0
0	1	1	1	X	X	X	X	X	X
1	0	0	0	0	1	0	1	0	1
1	0	0	1	0	1	0	1	0	0
1	0	1	0	1	0	1	0	0	0
1	0	1	1	X	X	X	X	X	X
1	1	0	0	0	0	1	1	0	0
1	1	0	1	0	1	1	1	0	0
1	1	1	0	1	0	1	1	0	0
1	1	1	1	X	X	X	X	X	X

c) Ecuaciones de salida:

		Q_1Q_0			
Z_1		00	01	11	10
	00	0	0	X	0
	01	0	1	X	0
X_1X_2	11	1	1	X	1
	10	0	0	X	1

		Q_1Q_0			
Z_2		00	01	11	10
	00	0	0	X	0
	01	1	0	X	1
X_1X_2	11	1	1	X	1
	10	1	1	X	0

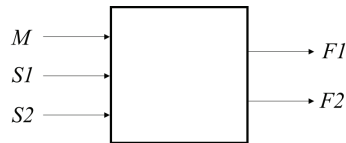
$$Z_1 = X_2Q_0 + X_1X_2 + X_1Q_1 = \overline{X_2Q_0} \cdot \overline{X_1X_2} \cdot \overline{X_1Q_1}$$

$$Z_2 = X_2\overline{Q_0} + X_1\overline{Q_1} = \overline{X_2Q_0} \cdot \overline{X_1Q_1}$$

$$T_1 = \overline{X_1} \cdot X_2 \cdot \overline{Q_1} \cdot \overline{Q_0}$$

$$T_0 = X_1 \cdot \overline{X_2} \cdot \overline{Q_1} \cdot \overline{Q_0}$$

- 32) El circuito de la figura corresponde al sistema de apertura de una puerta de garaje controlada por un mando a distancia.



El sensor M detecta la orden del mando, mientras que los sensores $S1$ y $S2$ indican si la puerta está completamente cerrada o completamente abierta, respectivamente. M toma el valor lógico '1' cuando recibe la orden del mando, y '0' en caso contrario. Los sensores $S1$ y $S2$ toman el valor lógico '1' cuando la puerta está completamente cerrada o completamente abierta, respectivamente, mientras que su valor es '0' durante las operaciones de apertura o cierre.

Las salidas $F1$ y $F2$ indican si está o no activado el motor de la puerta, y el sentido en caso de estar activado, respectivamente. Es decir, si $F1$ está activada ('1' lógico) entonces el motor se pone en marcha, mientras que al desactivarla ('0' lógico) éste se para. Mientras, $F2$ indica si el sentido del motor debe ser abrir ('1' lógico) o cerrar ('0' lógico).

El circuito que gestiona la apertura de la puerta sigue los siguientes criterios:

- Cuando la puerta está cerrada y recibe una orden del mando (M) se pone en marcha la apertura de la puerta, que finalizará cuando el sensor $S2$ indique que está totalmente abierta (cuando tome el valor '1' lógico).
- Cuando la puerta está abierta y recibe una orden del mando (M) se pone en marcha el cierre de la puerta, que finalizará cuando el sensor $S1$ indique que está totalmente cerrada (cuando tome el valor '1' lógico).
- Si durante la apertura o cierre de la puerta, y antes de que se haya completado la operación, se recibe una nueva orden del mando, ésta será ignorada.

Para ello, se pide:

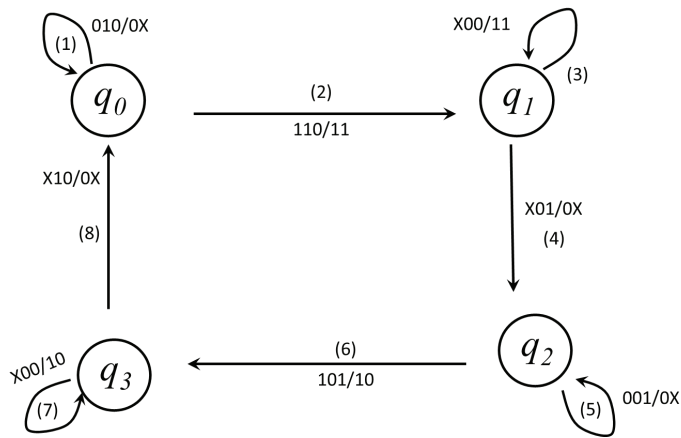
- Definir claramente los estados, entradas y salidas.
- Dibujar el diagrama de estados.
- Hacer la tabla de verdad del circuito.
- Si se quiere hacer que la puerta se cierre sola transcurridos 32 segundos desde que se abrió totalmente y no habiendo recibido la orden del mando a distancia, ¿qué circuitería se podría añadir al circuito anterior para conseguir este objetivo?

a) Definición y codificación de estados:

Definición Estados	Codificación
	Q_1Q_0
q0: puerta cerrada	0 0
q1: puerta abriéndose	0 1
q2: puerta abierta	1 0
q3: puerta cerrándose	1 1

Entradas			Salidas		
M	0	Ninguna orden	F ₁	1	Activado motor
	1	Orden del mando		0	No activado motor
S ₁	1	Puerta cerrada	F ₂	1	Abriendo puerta
	0	Puerta no cerrada		0	Cerrando puerta
S ₂	1	Puerta abierta			
	0	Puerta no abierta			

b) Diagrama de estados: (en las transiciones aparecen las señales MS_1S_2/F_1F_2)



Los casos posibles aparecen numerados entre paréntesis en cada transición, y son:

- (1) Puerta cerrada sin recibir orden del mando.
- (2) Puerta cerrada y se recibe orden del mando.

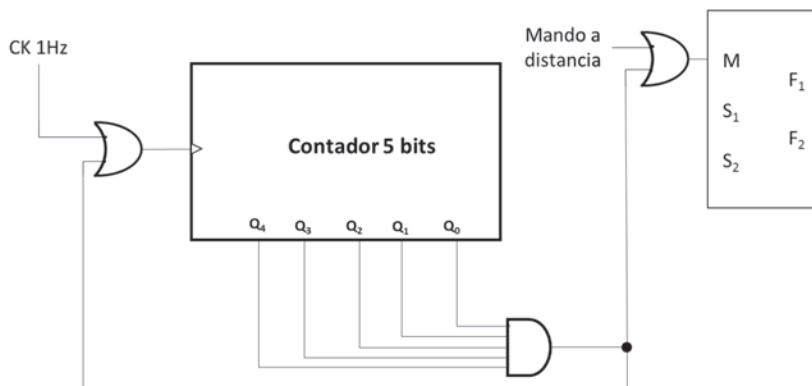
- (3) Puerta abriéndose (reciba o no orden).
- (4) Puerta abriéndose llega al final del recorrido y se para (reciba o no orden).
- (5) Puerta abierta sin recibir orden del mando.
- (6) Puerta abierta y se recibe orden del mando.
- (7) Puerta cerrándose (reciba o no orden).
- (8) Puerta cerrándose llega al final del recorrido y se para (reciba o no orden).

c) Tabla de verdad.

NOTA: De las 32 líneas que tiene la tabla de verdad, muchas de ellas no se pueden dar; por ejemplo, no pueden estar activados los dos sensores S_1 y S_2 simultáneamente. Ello supone poner condiciones de “no importa” en la tabla. Sólo hay 12 de los 32 casos a considerar, los marcados en el diagrama y en la tabla.

M	S ₁	S ₂	Q ₁ (t)	Q ₂ (t)	Q ₁ (t+1)	Q ₂ (t+1)	T ₁	T ₂	F ₁	F ₂	CASO
0	0	0	0	0	X	X	X	X	X	X	NO
0	0	0	0	1	0	1	0	0	1	1	(3)
0	0	0	1	0	X	X	X	X	X	X	NO
0	0	0	1	1	1	1	0	0	1	0	(7)
0	0	1	0	0	X	X	X	X	X	X	NO
0	0	1	0	1	1	0	1	1	0	X	(4)
0	0	1	1	0	1	0	0	0	0	X	(5)
0	0	1	1	1	X	X	X	X	X	X	NO
0	1	0	0	0	0	0	0	0	0	X	(1)
0	1	0	0	1	X	X	X	X	X	X	NO
0	1	0	1	0	X	X	X	X	X	X	NO
0	1	0	1	1	0	0	1	1	0	X	(8)
0	1	1	0	0	X	X	X	X	X	X	NO
0	1	1	0	1	X	X	X	X	X	X	NO
0	1	1	1	0	X	X	X	X	X	X	NO
0	1	1	1	1	X	X	X	X	X	X	NO
1	0	0	0	0	X	X	X	X	X	X	NO
1	0	0	0	1	0	1	0	0	1	1	(3)
1	0	0	1	0	X	X	X	X	X	X	NO
1	0	0	1	1	1	1	0	0	1	0	(7)
1	0	1	0	0	X	X	X	X	X	X	NO
1	0	1	0	1	1	0	1	1	0	X	(4)
1	0	1	1	0	1	1	0	1	1	0	(6)
1	0	1	1	1	X	X	X	X	X	X	NO
1	1	0	0	0	0	1	0	1	1	1	(2)
1	1	0	0	1	X	X	X	X	X	X	NO
1	1	0	1	0	X	X	X	X	X	X	NO
1	1	0	1	1	0	0	1	1	0	X	(8)
1	1	1	0	0	X	X	X	X	X	X	NO
1	1	1	0	1	X	X	X	X	X	X	NO
1	1	1	1	0	X	X	X	X	X	X	NO
1	1	1	1	1	X	X	X	X	X	X	NO

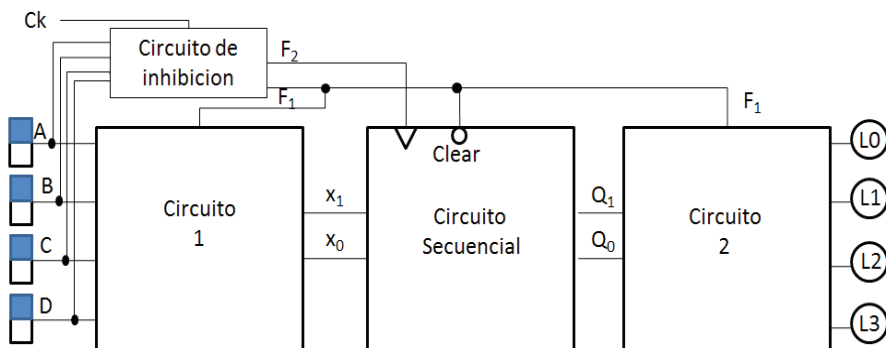
- d) Poner un contador de 5 bits y un reloj de 1Hz de modo que cuando llegue al final de cuenta se bloquee la entrada de reloj y se mande a M un valor 1.



33) Suponer un sistema de cuatro luces $L3, L2, L1$ y $L0$ en línea. La secuencia de luces es tal que siempre hay una y sólo una luz iluminada en cada ciclo de reloj. Tener en cuenta que las luces se activan a nivel bajo. Esta secuencia es controlada a través de 4 interruptores A, B, C y D con sus dos posiciones ON (1) y OFF (0). La secuencia de luces es como sigue:

- Si A está en ON, la secuencia de luces queda parada.
- Si B está en ON, la secuencia de iluminación avanza hacia abajo (ver el dibujo adjunto) y es cíclica (la última transición es de $L3$ a $L0$).
- Si C está en ON, la secuencia de iluminación avanza hacia arriba y es cíclica (la última transición es de $L0$ a $L3$).
- Si D está en ON, la secuencia se inicializa poniendo $L0$ a ON.

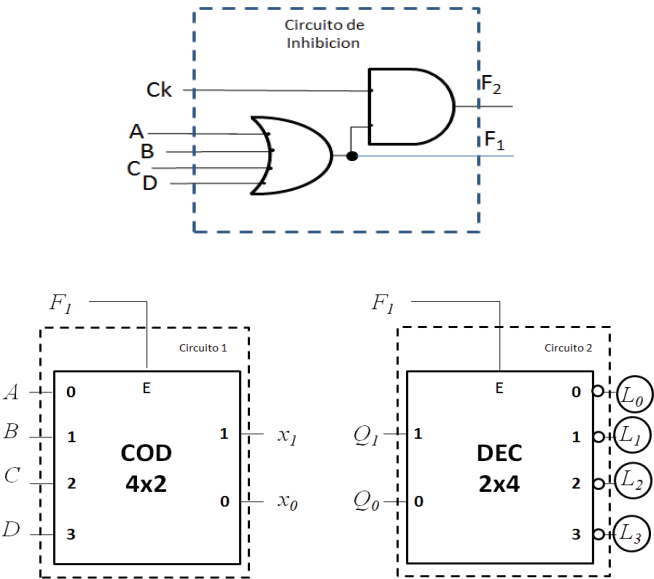
Asumir que el sistema sólo permite poner un interruptor en ON.



Si ningún interruptor está en ON, el sistema queda inhabilitado bloqueándose la señal de reloj, inhabilitando los circuitos 1 y 2, y reseteando los biestables del circuito secuencial. Notar que al deshabilitar el circuito 2, todas las luces quedarán apagadas.

- a) Hallar el interior de los circuitos: circuito de inhibición (circuito que implementa las funciones $F1$ y $F2$), circuito 1 y circuito 2. Utilizar cualquiera herramienta de diseño (puertas o módulos que se deseen). Dar una breve explicación.
- b) Explicar cuáles son las entradas y salidas del circuito secuencial. Explicar los estados, su significado en el contexto del sistema global y su codificación. Hallar únicamente el diagrama de estados del circuito secuencial. Llamar a los estados con la letra “ q ” y un subíndice (por ejemplo: $q0$, $q1$, ...).

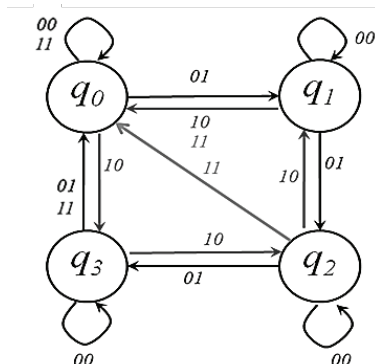
a) Interior de los circuitos de inhibición, circuito 1 y circuito 2:



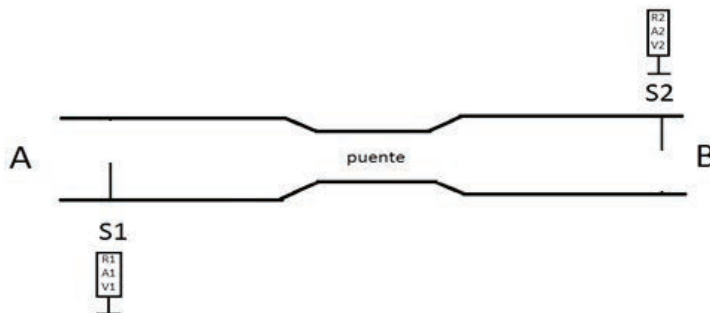
b) Entradas y salidas del circuito secuencial. (x_1x_0 entradas, Q_1Q_0 salidas):

Definición Estados		Codificación
Estado	Significado	Q_1Q_0
q0	L0 está iluminado	0 0
q1	L1 está iluminado	0 1
q2	L2 está iluminado	1 0
q3	L3 está iluminado	1 1

Diagrama de estados. En las transiciones aparecen x_1x_0 .



- 34) Diseñar un circuito secuencial para regular el tráfico en una carretera que tiene un estrechamiento en un puente al cruzar por encima de un río. Para ello, se colocan semáforos en ambos sentidos, según muestra la figura.



El estado inicial es el de “No circulación” por el puente en cualquier sentido, en cuyo caso los semáforos permanecen en ámbar (precaución). Cuando alguno de los sensores $S1$ o $S2$ detecta un coche, su semáforo se pone a verde y el contrario se pone a rojo. Este estado de semáforos permanece durante 30 segundos y, de forma asíncrona, se vuelven a poner ambos semáforos en ámbar. Se supone que en 30 segundos el coche tiene tiempo suficiente para atravesar el puente. Se supone también que la separación temporal entre coches en el mismo sentido es de al menos 120 segundos, por lo que nos aseguramos de que nunca habrá dos coches seguidos en el mismo sentido atravesando el puente. Finalmente, se supone también que nunca llegarán dos coches en sentido contrario a la vez a sus respectivos sensores.

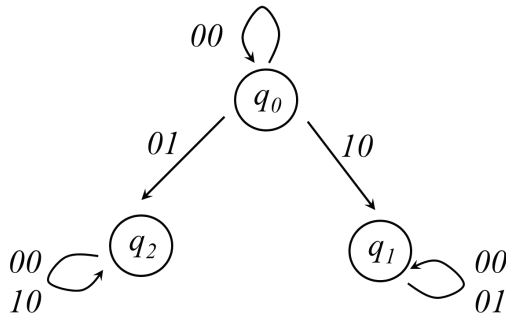
Se pide:

- Definir y codificar los estados.
- Realizar el diagrama de transición de estados.
- Hacer la tabla de transición y excitación usando biestables T para el biestable más significativo y D para el menos significativo. El resto de biestables, si los hubiera, serán JK.
- Encontrar las ecuaciones booleanas de las funciones de excitación de los biestables y de las salidas de los semáforos a partir de las salidas de los biestables.
- Integrar un contador como bloque con frecuencia de reloj de 0,5 Hz al circuito anterior para inicializar los semáforos al estado inicial.

- Definición y codificación de estados.

Definición Estados (autómata de Moore)		Codificación Q_1Q_0	Salidas					
Estado	Significado		V1	A1	R1	V2	A2	R2
q0	Estado inicial, encendidas luces ámbar de ambos semáforos	0 0	0	1	0	0	1	0
q1	Circulación permitida desde A hasta B, con luz verde V1	0 1	1	0	0	0	0	1
q2	Circulación permitida desde B hasta A, con luz verde V2	1 0	0	0	1	1	0	0

- Diagrama de estados. En las transiciones aparece S_1S_2 :



c) Tabla de excitación del sistema:

S1	S2	Q ₁	Q ₀	Q ₁ (t+1)	Q ₀ (t+1)	T ₁	D ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	1
0	0	1	0	1	0	0	0
0	0	1	1	X	X	X	X
0	1	0	0	1	0	1	0
0	1	0	1	0	1	0	1
0	1	1	0	X	X	X	X
0	1	1	1	X	X	X	X
1	0	0	0	0	1	0	1
1	0	0	1	X	X	X	X
1	0	1	0	1	0	0	0
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

d) Ecuaciones de las entradas a biestables y salidas:

Q ₁	Q ₀	V1	R1	V2	R2	A
0	0	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	1	1	0	0
1	1	X	X	X	X	X

		Q ₁ Q ₀			
T ₁		00	01	11	10
00		0	0	X	0
01		1	0	X	X
11		X	X	X	X
10		0	X	X	0

		Q ₁ Q ₀			
D ₀		00	01	11	10
00		0	1	X	0
01		0	1	X	X
11		X	X	X	X
10		1	0	X	0

$$T_1 = S_2 \overline{Q_0} \quad D_0 = \overline{S_1} Q_0 + S_1 \overline{Q_1} \overline{Q_0} \quad V1 = R2 = Q_0$$
$$V2 = R1 = Q_1 \quad A = \overline{Q_1 + Q_0}$$

e) Circuito de reinicio:

Es un bloque contador de 4 salidas (30 seg. a ½ Hz es la salida 15), cuyas salidas van directamente a una puerta AND de 4 entradas y la salida de la misma, a las entradas *Clear* de los biestables, si son activas a nivel alto. En caso contrario, la puerta sería NAND.

La entrada de disparo del Contador es una puerta OR cuyas entradas son los dos sensores *S1* y *S2*.

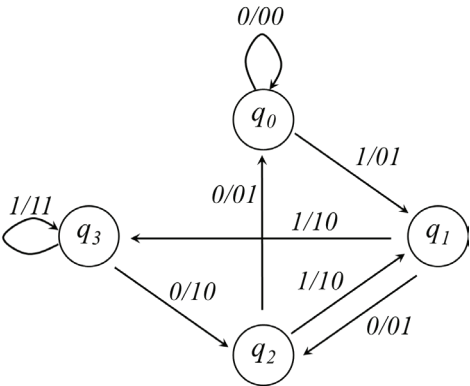
35) Se desea diseñar un circuito secuencial que acepta como entrada la variable *E*, que vale 1 cuando una cámara de video reconoce un cuadrado dibujado en una cinta de transporte que se va moviendo, y 0 cuando reconoce un círculo. En cada ciclo de reloj la cámara reconocerá una de estas dos figuras geométricas. El circuito secuencial debe proporcionar como salida en cada momento la cantidad de cuadrados visualizados, incluyendo la entrada actual y las dos anteriores. Para ello, usar las salidas que sean necesarias nombrándolas como *S1*, *S2*, etc. ... Se pide:

- a) Definición y codificación de estados.
- b) Diagrama de estados.
- c) Tabla de transición y excitación, usando biestables JK para el más significativo, D para el menos, y T para el resto si los hubiera.
- d) Implementación de las funciones de excitación de los biestables usando puertas lógicas.
- e) Implementación de las funciones de salida usando un MUX 2x1 para cada una de las variables.

a) Definición y codificación de estados

Definición Estados		Codificación
Estado	Significado	Q ₁ Q ₀
q0	Última figura recibida CÍRCULO y penúltima figura recibida CÍRCULO	0 0
q1	Última figura recibida CUADRADO y penúltima figura recibida CÍRCULO	0 1
q2	Última figura recibida CÍRCULO y penúltima figura recibida CUADRADO	1 0
q3	Última figura recibida CUADRADO y penúltima figura recibida CUADRADO	1 1

b) Diagrama de estados:



c) Tabla de excitación y salidas

E	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	J_1	K_1	D_0	S_1	S_2
0	0	0	0	0	0	X	0	0	0
0	0	1	1	0	1	X	0	0	1
0	1	0	0	0	X	1	0	0	1
0	1	1	1	0	X	0	0	1	0
1	0	0	0	1	0	X	1	0	1
1	0	1	1	1	1	X	1	1	0
1	1	0	0	1	X	1	1	1	0
1	1	1	1	1	X	0	1	1	1

d) Implementación de las entradas a biestables:

		Q_1Q_0			
J_1		00	01	11	10
E	0	0	1	X	X
	1	0	1	X	X

$$J_1 = Q_0$$

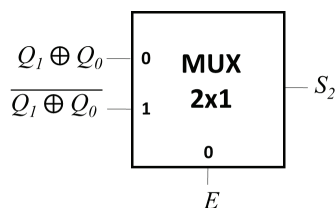
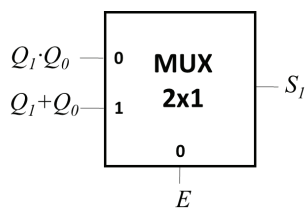
		Q_1Q_0			
K_1		00	01	11	10
E	0	X	X	0	1
	1	X	X	0	1

$$K_1 = \bar{Q}_0$$

		Q_1Q_0			
D_0		00	01	11	10
E	0	0	0	0	0
	1	1	1	1	1

$$D_0 = E$$

e) Implementación de las salidas:



TEST Y PREGUNTAS CORTAS

1) Un contador síncrono ascendente de 0 a 11 es:

- ☐ A Un sistema secuencial con 12 estados, y necesita 4 biestables
- ☐ B Un sistema secuencial con 11 estados, y necesita 4 biestables
- ☐ C No es una máquina secuencial, pero necesita 4 biestables

2) Señala la opción verdadera:

- ☐ A Un contador Johnson de 6 bits cuenta 64 estados.
- ☐ B El período mínimo de reloj determina la frecuencia mínima de trabajo que depende del tiempo de propagación del biestable.
- ☐ C Las *carreras* ocurren cuando el nivel activo es demasiado amplio permitiendo realimentación de la salida con la entrada.

3) Indica la opción correcta:

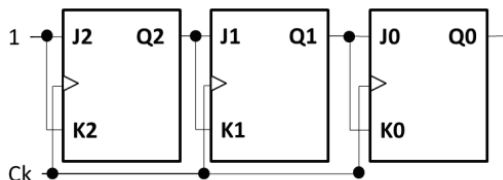
- ☐ A Un reconocedor de secuencias puede diseñarse como un sistema combinacional ya que no necesita elementos de memoria.
- ☐ B Un contador asíncrono de tres bits consta de tres biestables T donde las entradas de reloj en dos de ellos proceden del estado del biestable anterior.
- ☐ C Tres biestables tipo T en serie constituyen un registro de desplazamiento de tres bits.

4) Señala lo que se considere cierto:

- ☐ A En un autómata de Moore la salida depende del estado actual y de la entrada.
- ☐ B En un autómata de Mealy puede que no haya elementos de memoria.
- ☐ C En un contador en anillo la salida Q del último biestable RS se conecta a la entrada S del primero.

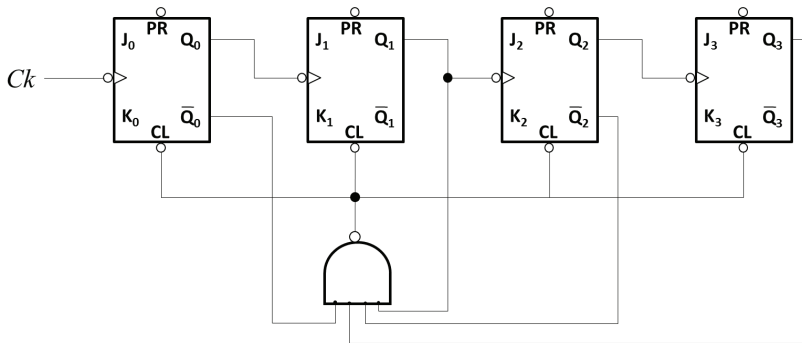
5) Sobre la figura, inicializado a 000 señala, la opción correcta:

- ☐ A Cuenta 0, 4, 2, 7, ...
- ☐ B Es un registro de desplazamiento.
- ☐ C Es un contador en anillo.



6) En el circuito de abajo las entradas J y K de todos los biestables están conectadas a 5 voltios. El circuito de la figura es:

- ☐ A Registro de desplazamiento serie.
☐ B Contador síncrono de 4 bits.
☐ C Contador asíncrono BCD ascendente.

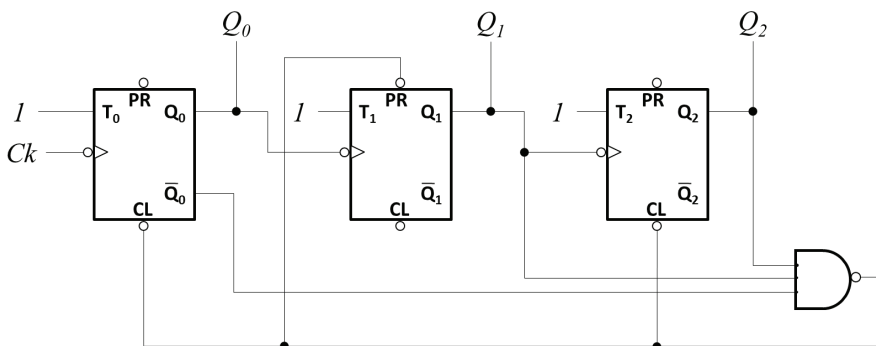


7) Señala la respuesta falsa.

- ☐ A Un contador Johnson de 7 bits cuenta 14 estados.
☐ B Un biestable de tipo T se consigue a partir de un JK haciendo $J=K$.
☐ C Los registros de desplazamiento suelen implementarse con biestables tipo T.

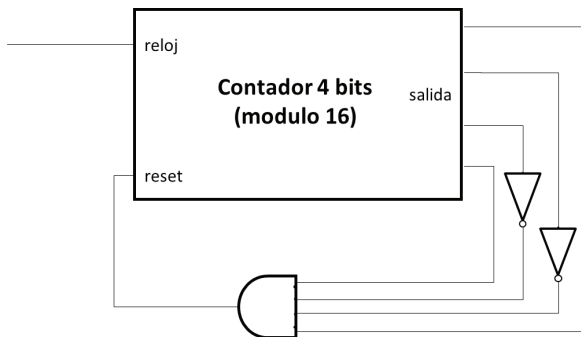
8) ¿Cuál es la secuencia de cuenta del circuito asíncrono de la figura, si el valor inicial almacenado es 5?

- ☐ A 5,6,2,3,4,5,6,...
☐ B 5,0,1,2,3,4,5,...
☐ C 5,2,3,4,5,...



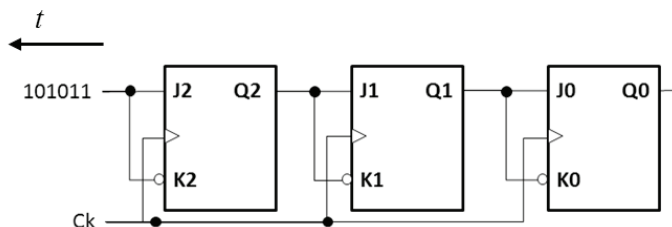
- 9) En el circuito de la figura adjunta se ha modificado un contador de 4 bits (módulo 16) para que en lugar de la secuencia de cuenta de 0 a 15 se obtenga la secuencia:

- ☐ A 9 a 15
☐ B 0 a 9
☐ C 0 a 8

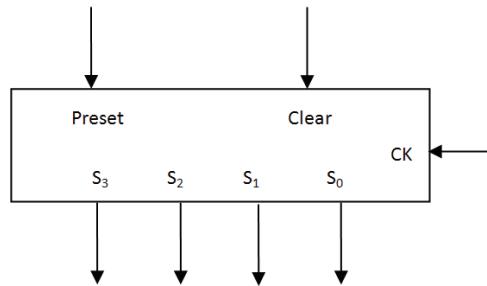


- 10) En la figura los tres biestables se inician a 0 y la secuencia de entrada es por orden de tiempos 1 1 0 1 0 1. La codificación binaria que contienen los biestables corresponde a la secuencia de números:

- ☐ A No es un registro de desplazamiento y cuenta 0, 1, 3, 5, 2, 5, ...
☐ B Es un contador ascendente y cuenta 0, 1, 2, 3, 4, ...
☐ C Es un registro de desplazamiento y cuenta 0, 4, 6, 3, 5, 2, ...



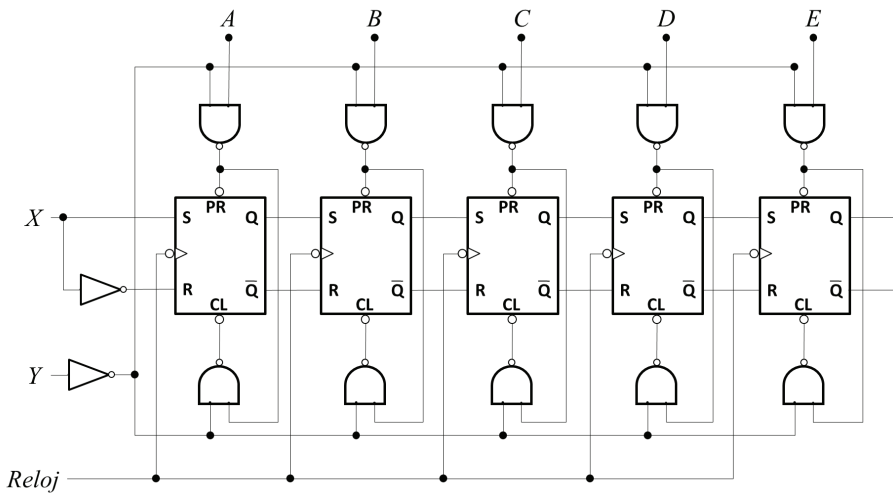
- 11) El circuito de la figura representa un contador de 4 bits. A partir de él se desea implementar un circuito que divida por 4 la frecuencia del reloj CK. Teniendo en cuenta que se realiza una cuenta en cada flanco de subida del reloj, selecciona la opción que permite obtener el circuito deseado.



- ☐ A Tomando como salida la señal S_1 .
- ☐ B Tomando como salida la señal S_2 .
- ☐ C Llevando las señales S_2 y S_0 a una puerta AND y la salida al *Clear* (borrar al llegar a 5).

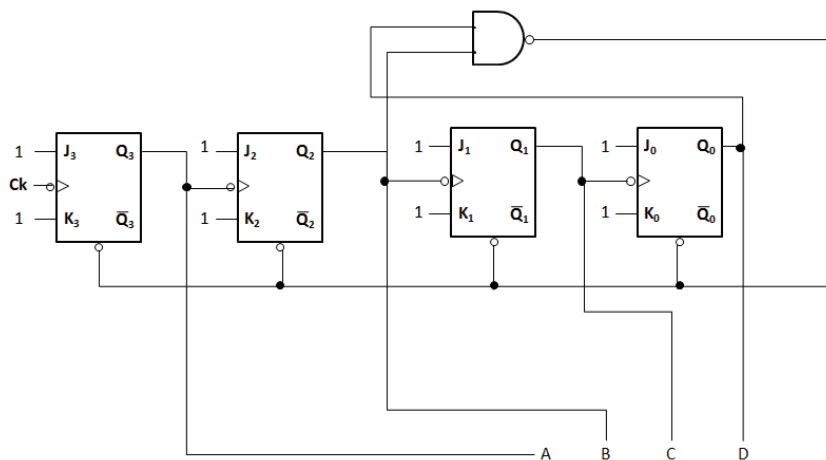
12) Señala la opción correcta. El circuito de la figura de abajo es:

- ☐ A Un registro de desplazamiento universal reversible.
- ☐ B Un registro de desplazamiento con carga paralelo y salida serie.
- ☐ C Un contador en anillo que cuenta 8 estados.



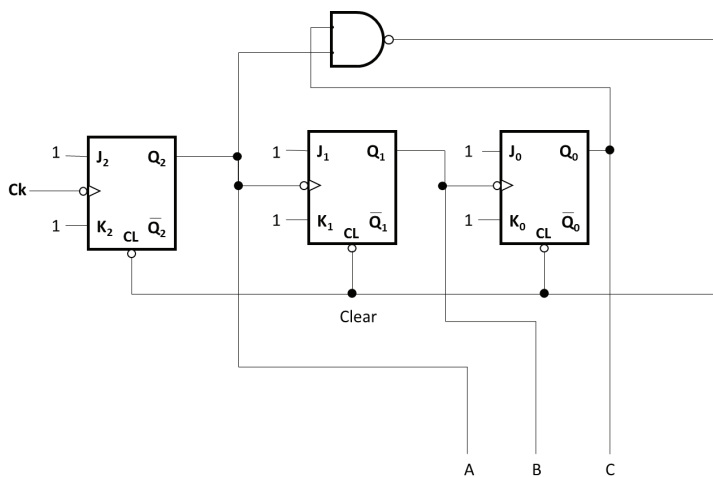
13) Señala la opción correcta. El circuito de la figura de abajo:

- ☐ A Es un circuito divisor de frecuencia por 10.
- ☐ B Es un contador descendente.
- ☐ C Es un contador ascendente síncrono.



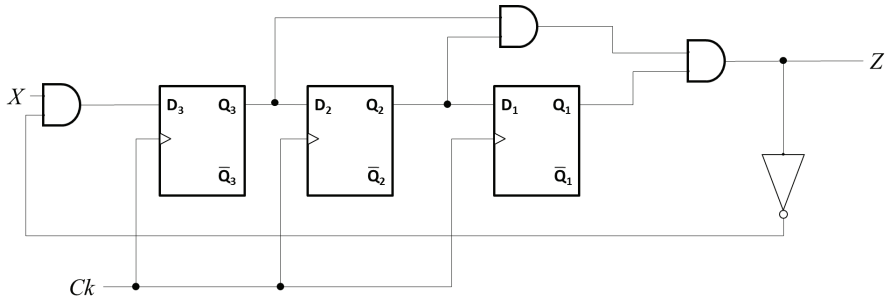
14) Identifica este circuito. Supón que se han inicializado los biestables a 1 1 0.

- ☐ A Es un contador módulo-5 asíncrono cuyo primer estado es la cuenta 3.
- ☐ B Es un registro de desplazamiento 3 bits con salida paralelo que cuenta la secuencia 110, 011, 001, 000
- ☐ C Es un contador síncrono de 0 a 7 cuyo primer estado es la cuenta 6.



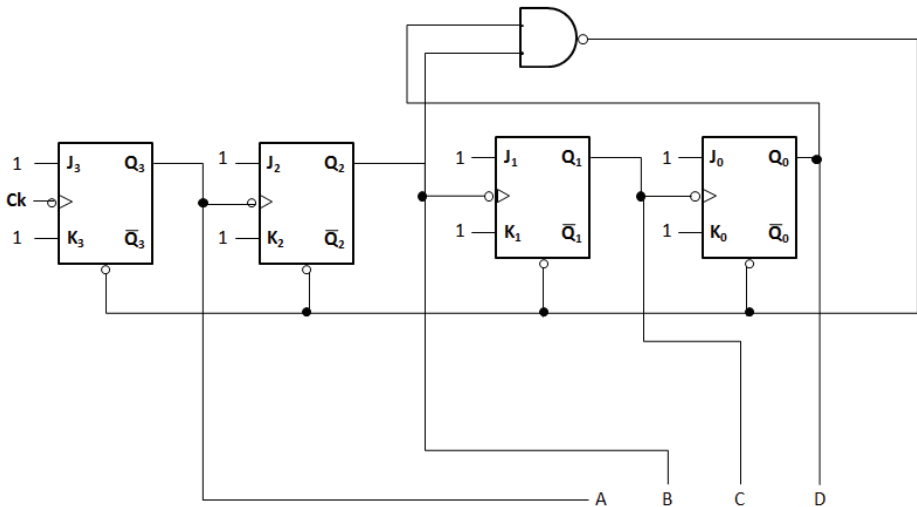
15) ¿Qué función se realiza en el circuito de la figura?

- ☐ A Es un reconocedor de secuencias 111 aisladas.
- ☐ B Es un contador de 3 bits ascendente/descendente, según el valor de x .
- ☐ C Es un contador de Johnson, independientemente del valor de x .



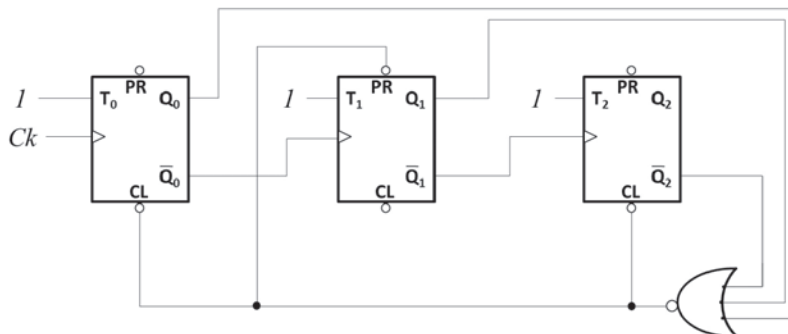
16) Identifica el siguiente circuito:

- ☐ A Circuito divisor por 9.
☐ B Contador módulo-10 asíncrono, ya que cuenta de 0 a 9.
☐ C Contador binario síncrono descendente.

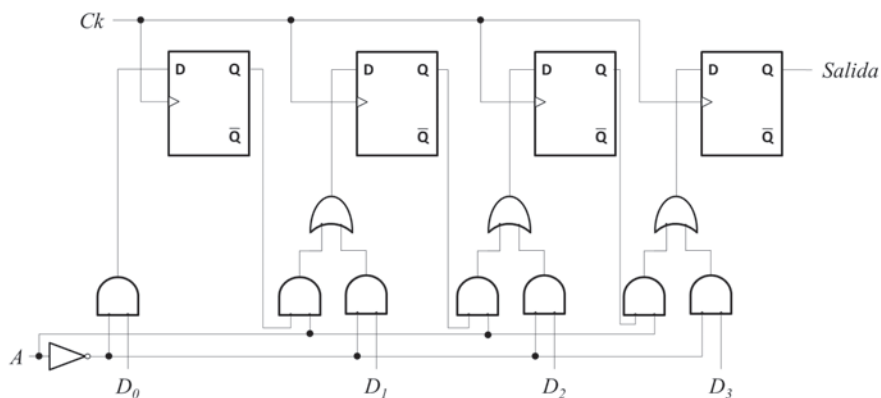


17) ¿Cuál es la secuencia generada por el circuito de la figura adjunta si se parte de 0 (Q_2 es el bit más significativo)?

- ☐ A 0, 7, 6, 5, 4, 0, 7, ...
☐ B 0, 1, 2, 3, 2, 3, ...
☐ C 0, 1, 2, 2, 2, ...

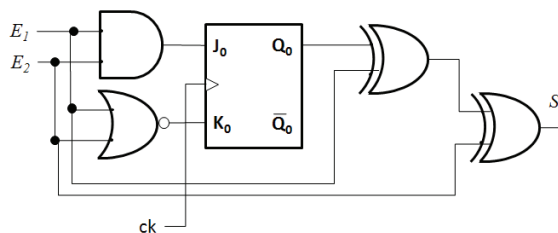


18) Explicar qué función realiza la señal A (para $A=0$ y $A=1$) en el circuito de la figura adjunta.



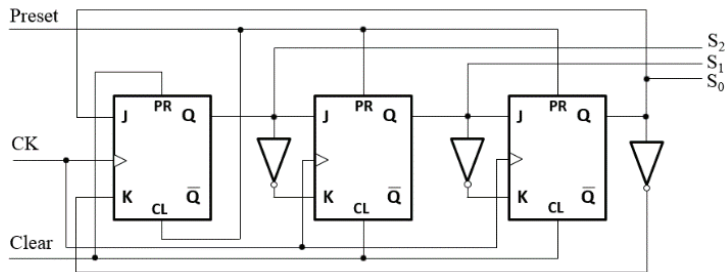
- ☐ A Se trata de un registro con entrada en paralelo y salida en serie, en el que la señal A tiene la función de carga y desplazamiento.
- ☐ B Es un contador síncrono reversible en función del valor de A .
- ☐ C Es un contador asíncrono módulo-16 donde A no tiene ninguna influencia.

19) Señala la opción correcta en el circuito de la figura:



- ☐ A Es un sumador serie en el que S es la suma de bits.
☐ B Es un circuito combinacional.
☐ C Si $E_1(t)=0$, $E_2(t)=1$, $Q(t)=1$, entonces $Q(t+1)=0$

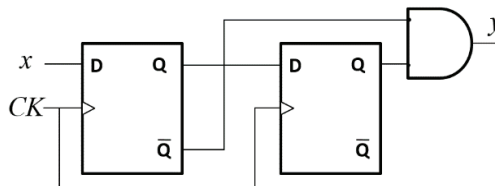
20) En la figura adjunta el estado inicial de los biestables es 110 (6). Selecciona la respuesta correcta (tomar S_2 como variable más significativa):



- ☐ A La secuencia seguida en los biestables es 6, 5, 4, 3, 2, 1, ...
☐ B La secuencia seguida en los biestables es 6, 2, 0, 6, 2, 0, ...
☐ C Las respuestas A y B son falsas.

21) Marca la opción correcta respecto del circuito implementado en la figura.

- ☐ A El circuito puede considerarse la implementación de un autómata de Mealy ya que posee una entrada x .
☐ B La salida en el instante t es $y(t)=0$ cuando la entrada en el instante $t-1$ es $x(t-1)=1$.
☐ C Tanto A como B son falsas.



SOLUCIONES A TEST Y PREGUNTAS CORTAS

CAPÍTULO 1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C	A	FFV	VFFF	VVF	B	B	B	VVV	B	C	B	B	B	B

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
B	A	A	B	C	A	C	B	C	B	B	A	C	C	B

31	32	33	34	35	36
C	B	C	C	C	C

CAPÍTULO 2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	C	B	A	C	A	A	B	C	A	C	C	B	C	B	A	B	C	C	A

CAPÍTULO 3

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
VFV	VVF	VFF	VFF	C	C	B	B	C	B	C	A	C	C	C

16	17	18
C	C	C

CAPÍTULO 4

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C	C	A	A	B	A	B	A	B	C	A	B	C	C	B

16	17	18	19	20	21	22	23	24	25	26
A	C	A	B	C	C	C	C	C	C	A

CAPÍTULO 5

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

CAPÍTULO 6

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C	A	B	B	B	A	B	A	A	B	C	B	A	B	B

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
A	A	B	C	A	C	B	B	B	A	C	B	C	B	B

31	32	33	34	35	36	37	38	39
B	B	C	B	B	C	B	C	B

CAPÍTULO 7

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	C	B	C	A	C	C	C	C	C	A	B	A	A	A

16	17	18	19	20	21
B	B	A	A	C	B

Circuitos Digitales

Problemas y ejercicios resueltos

Este libro es el fruto del trabajo de recopilación, durante los últimos 28 años, de ejercicios planteados en los exámenes de la Escuela Superior de Informática de Ciudad Real, (UCLM), en asignaturas como: Circuitos Digitales, Diseño Digital, Tecnología de Computadores, Diseño Lógico, etc. Tiene como finalidad, facilitar al lector introducirse en el mundo de los circuitos digitales como base fundamental del hardware de los computadores.

El libro consta de problemas y ejercicios que cubren gran parte de la temática de este tipo de asignaturas, planteando situaciones reales en las que se pide la definición e implementación del circuito digital que resuelve dicho problema.

El libro está estructurado en siete capítulos que cubren los aspectos generales de:

- Representación de la información y funciones lógicas.
- Diseño lógico de circuitos combinacionales con puertas lógicas y con módulos combinaciones.
- Análisis y diseño de circuitos secuenciales basados en autómatas de estados finitos.

Estos ejercicios, junto con las cuestiones tipo test planteadas y cuya solución también se facilita, suponen una importante ayuda para afianzar los conocimientos teóricos adquiridos.

Esperamos que el contenido de este libro resulte de utilidad para los alumnos de los primeros cursos de ingeniería.



9 788499 647616

