

Sugerencias Didácticas

- Resulta fundamental en este estudio una clara diferenciación entre los circuitos combinacionales y los secuenciales; se explicarán en cada caso sus aplicaciones prácticas, incidiendo especialmente en su importancia tecnológica.
- Entre los circuitos integrados combinacionales se analizará el funcionamiento de los codificadores, decodificadores, multiplexadores y convertidores de código, en especial los convertidores BCD a 7 segmentos, tan empleados en calculadoras de bolsillo y relojes digitales. El estudio de los circuitos sumadores puede servir de punto de partida para que el alumnado aborde el problema de los circuitos restadores en distintos códigos.
- Se clasificarán los circuitos secuenciales en asíncronos y síncronos. El funcionamiento de los primeros se puede representar mediante las llamadas *tablas de fases*, que permiten implementar el circuito mediante puertas lógicas. Es importante extenderse en la manera de construir estas tablas y en su posterior reducción, que sirve para obtener la tabla de verdad, que relaciona todas las variables del circuito y conseguir, así, su forma combinacional.
- Entre los biestables asíncronos, se estudiará el R-S; los problemas que presenta su sincronización con el reloj que controla el sistema hacen necesario utilizar dos soluciones, que son la edge-triggered y la master-slave. También se considerarán los biestables T, Latch, D y J-K, todos ellos síncronos; en cada caso se buscará siempre relacionar la misión que desempeñan estos biestables y su utilización en circuitos prácticos.

SOLUCIONES a las Actividades propuestas

Pág.

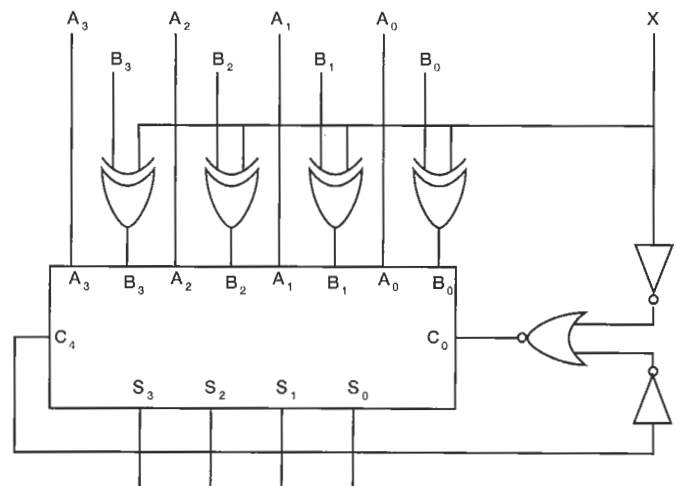
412

1. Ante todo, conviene recordar que la resta binaria se realiza por medio de la adición de complementos (véase Unidad didáctica anterior).

- Si se utiliza el convenio de complemento a uno y el bit de signo, para efectuar la resta se le suma al minuendo el complemento a uno del sustraendo sumando, además, al bit menos significativo del resultado el acarreo de orden superior obtenido. Si la diferencia es positiva, a la salida se obtiene el valor correcto con bit de signo 0; en caso contrario, aparece representada según el convenio del complemento a uno con el bit de signo igual a 1.

En un circuito lógico capaz de efectuar la resta según este procedimiento se introducen el minuendo y el complemento a uno del sustraendo, y el acarreo de orden superior C_4 se realimenta a la entrada C_0 de acarreo inicial.

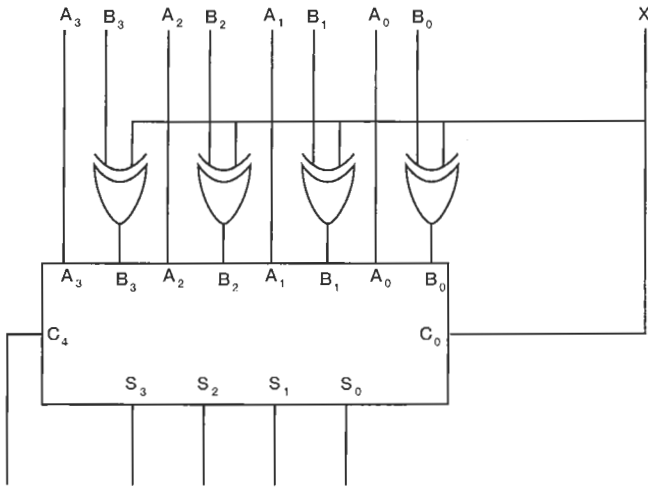
En la figura siguiente se representa un circuito que puede funcionar como sumador o como restador, según el valor de una entrada X de control. El circuito funciona como sumador cuando $X = 0$; cuando $X = 1$, se introduce el sustraendo B complementado y el acarreo C_4 se realimenta a la entrada C_0 .



- Si se utiliza el convenio de complemento a dos y el bit de signo, para efectuar la resta se le suma al minuendo el complemento a dos del sustraendo. A este respecto, conviene recordar que el complemento a dos de un número se puede obtener sumando 1 al dígito menos significativo del complemento a uno de dicho número. Por consiguiente, la resta se podrá llevar a cabo si se introducen en un sumador el minuendo por una parte, y por otra el complemento a uno del sustraendo; haciendo, además, siempre igual a 1 la entrada C_0 de acarreo inferior.

Si la diferencia es positiva, a la salida se obtiene el valor correcto; en caso contrario, aparece representada según el convenio del complemento a dos.

En la figura siguiente se representa un circuito que puede funcionar también como sumador o como restador, según el valor de la entrada X de control. Si $X = 0$ funciona como sumador; si $X = 1$, como restador.



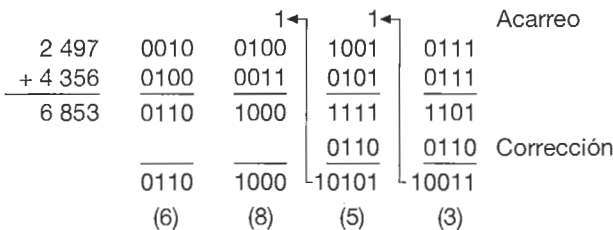
2. • **Aritmética en el código BCD natural.** Este código se suele utilizar cuando el sistema es de dimensiones reducidas y se deben generar salidas decimales. En estos casos, la ausencia de conversiones de código binario-BCD para presentar datos de salida compensa sobradamente la mayor longitud del código BCD en comparación con el binario natural.

El código BCD resulta muy cómodo para la presentación visual de números mediante dispositivos de 7 fragmentos y, en general, cualquier sistema de entrada o salida que convierta cada dígito decimal, por separado, en un código binario.

• **Suma en el código BCD natural.** La suma se realiza por medio de la adición binaria, para cada dígito decimal, de los bits correspondientes. Si el resultado de la suma es mayor que nueve, se produce un acarreo que se suma al dígito decimal siguiente. Pero como en el código BCD natural no se utilizan más que diez de las dieciséis combinaciones posibles de cuatro bits, para obtener el acarreo necesario y conseguir, a la vez, el resultado correcto, es necesario sumar $16 - 10 = 6$ al resultado.

En términos generales, el proceso que se sigue para la suma es el siguiente: se suman bit a bit las dos codificaciones correspondientes al mismo dígito decimal; si el resultado obtenido equivale a un número decimal mayor que nueve, se le suma el número binario 0110; de esta forma se obtiene el valor correcto del dígito decimal y en su caso el acarreo necesario para llevar a cabo la suma del dígito decimal siguiente.

Ejemplo:



Cuando se lleva a cabo la suma en este código de dos números de cuatro dígitos cada uno, más un bit representativo del acarreo de una etapa anterior, a la salida se pueden generar 19 combinaciones binarias diferentes: de ellas sólo 10 representan correctamente la suma en el código BCD natural; las otras 9 precisan ser corregidas.

En la tabla siguiente se representan estas 19 combinaciones, así como su equivalente correcto en el código BCD natural.

	Binario					BCD natural				
	C ₄	S ₃	S ₂	S ₁	S ₀	C ₄	S ₃	S ₂	S ₁	S ₀
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	1	0	0	1
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0

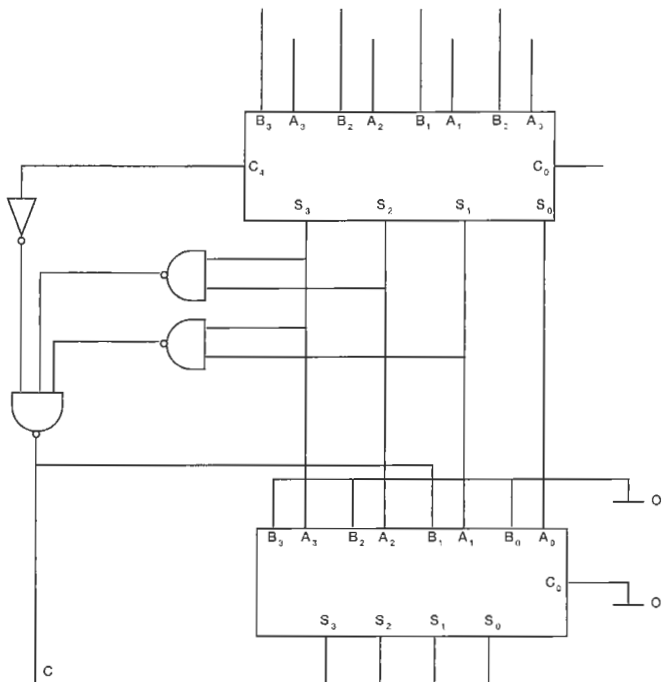
Si el resultado binario es menor que 10, es correcto en código BCD natural. Por el contrario, si es igual o mayor que 10, para obtener el valor correcto en dicho código se deberá sumar 6. En la tabla siguiente se obtiene la variable de control X que, si es cero, indica que el resultado es menor que 10. Si es 1, el resultado es mayor o igual que 10. Por lo tanto, esta variable X se podrá utilizar como el acarreo BCD que se produce en la suma, puesto que son coincidentes.

S ₁ S ₀ \ C ₄ S ₃ S ₂	000	001	011	010	110	111	101	100
00	0	0	1	0	X	X	X	1
01	0	0	1	0	X	X	X	1
11	0	0	1	1	X	X	X	X
10	0	0	1	1	X	X	X	1

La suma BCD natural correcta se puede generar con el circuito que se representa en la figura de la página siguiente: la suma binaria se obtiene en el primer sumador, y se genera la variable de control X. Si esta variable vale 1, al resultado binario anterior se le suma en un segundo sumador el valor 6 (0110). Si X = 0, el resultado binario del primer sumador queda inalterado.

• **Resta en el código BCD natural.** La resta se suele efectuar sumando al minuendo el complemento a nueve o a diez del sustraendo. De esta manera se pueden obtener la suma y la resta con un único circuito básico. Mediante el método del complemento a nueve, el circuito global necesario resulta más sencillo que con el método de complemento a diez.

A efectos prácticos, el procedimiento que se sigue es el siguiente: Para cada dígito decimal se suman el minuendo y el complemento a nueve del sustraendo, siguiendo las normas indicadas anteriormente para la suma. Si se origina



158	0	0001	0101	1000	Acarreo
- 347	1	0110	0101	0010	
- 189	1	1000	1011	1010	
			0110	0110	Corrección
	1	1000	10001	10000	
	-	(1)	(8)	(9)	

Teniendo en cuenta las consideraciones que se acaban de hacer, la resta de dos números se puede realizar introduciendo en un sumador BCD: por una parte el minuendo; por otra, el complemento a nueve del sustraendo, que deberá generarse en un circuito especial; y, por último, realimentando el acarreo C_4 de orden superior a la entrada C_0 de acarreo de orden inferior.

La tabla de verdad del circuito combinacional que genera el complemento a nueve de un número B es la siguiente:

B	9-B	B_3	B_2	B_1	B_0	B'_3	B'_2	B'_1	B'_0
0	9	0	0	0	0	1	0	0	1
1	8	0	0	0	1	1	0	0	0
2	7	0	0	1	0	0	1	1	1
3	6	0	0	1	1	0	1	1	0
4	5	0	1	0	0	0	1	0	1
5	4	0	1	0	1	0	1	0	0
6	3	0	1	1	0	0	0	1	1
7	2	0	1	1	1	0	0	1	0
8	1	1	0	0	0	0	0	0	1
9	0	1	0	0	1	0	0	0	0

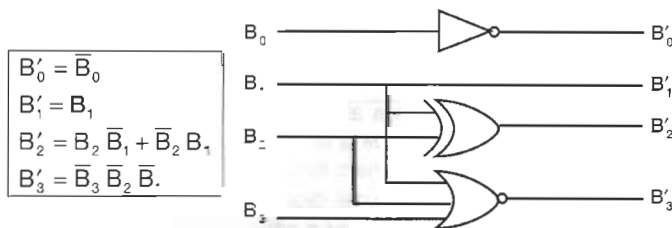
acarreo, éste se suma al dígito decimal siguiente. Si se produce acarreo en el dígito decimal más significativo, ello indica que el resultado es positivo y este acarreo se suma al dígito decimal menos significativo; si no se produce acarreo en este dígito, el resultado es negativo y representa el complemento a nueve del número resultante de la operación.

Ejemplos:

2 362	0010	0011	0110	0010	Acarreo
- 1 829	1000	0001	0111	0000	
0 533	1010	0101	1101	0011	
	0110		0110		Corrección
	10000	0101	10011	0011	
	+ (0)	(5)	(3)	(3)	

1 829	0001	1000	0010	1001	Acarreo
- 2 362	0111	0110	0011	0111	
- 0 533	1001	1110	0110	10000	
		0110		0110	Corrección
	1001	10100	0110	10110	
	- (0)	(5)	(3)	(3)	

y la realización de este circuito:

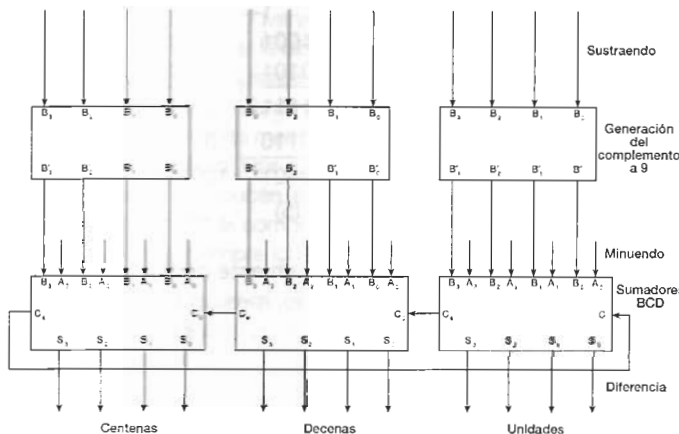


Por lo tanto, un restador en código BCD puede construirse según el esquema siguiente:

Si los números se representan con bit de signo (0 para positivo y 1 para negativo), la resta se efectúa de forma semejante, utilizando el bit de signo como si fuera el dígito decimal $n + 1$.

Ejemplos:

347	0	0011	0100	0111	Acarreo
- 158	1	1000	0100	0001	
189	10	1011	1000	1001	
		0110			Corrección
	10	10001	1000	1001	
	+	(1)	(8)	(9)	



• **Aritmética en el código BCD exceso tres.**

Con este código las operaciones aritméticas se simplifican considerablemente.

• **Suma en el código BCD exceso tres.** Se realiza de forma semejante a la empleada en el código BCD natural.

Si el resultado es mayor que nueve, como éste está expresado en código BCD exceso seis, se genera acarreo. Por esto mismo, ya se ha realizado la corrección necesaria de sumar seis, siendo sólo necesario sumar tres (0011) para representar este resultado en código BCD exceso tres.

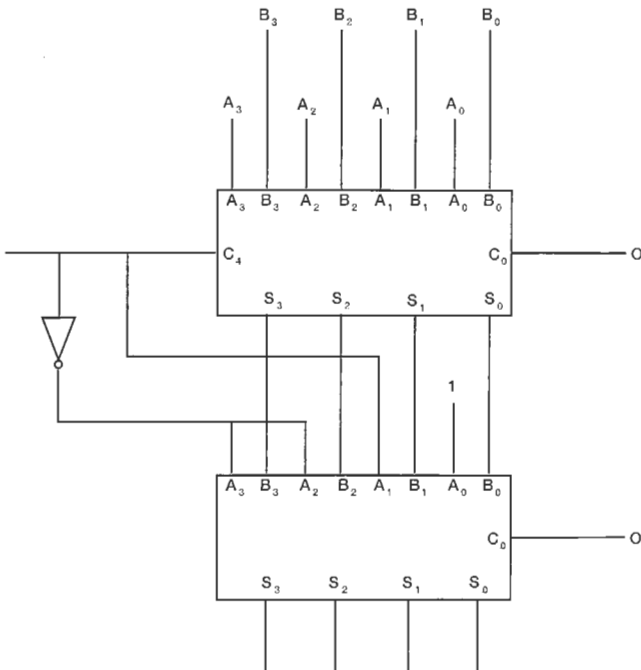
Si el resultado es menor o igual que nueve, no se produce acarreo. Para volver a representar el resultado en BCD exceso tres, como éste está en BCD exceso seis, habrá que restar al mismo el número tres (0011). Esta sustracción del número tres puede realizarse más cómodamente sumando su complemento a dos —es decir, trece (1101)—, y no considerando el acarreo producido.

En términos generales, el proceso que se sigue para la suma es el siguiente: se suman bit a bit las dos codificaciones correspondientes al mismo dígito decimal; si se origina acarreo es necesario sumar el número 0011 y el acarreo producido se suma al siguiente dígito decimal; si no se origina acarreo, la corrección que es necesario efectuar consiste en sumar el número 1101, sin considerar el acarreo que se produce.

Ejemplo:

2 496	0101	0111	1100	1001		Acarreo
+ 4 357	0111	0110	1000	1010		
6 853	1100	1110	10101	10011		Corrección
	1101	1101	0011	0011		
	1001	1011	1000	0110		
	(6)	(8)	(5)	(3)		

La figura que se incluye a continuación representa un circuito compuesto por dos sumadores. En el primero se introducen los dos sumandos y se efectúa su suma según las reglas anteriores. El segundo sumador se utiliza para añadir al resultado del primero el número 0011 o el 1101, según que exista o no el acarreo de orden superior C_4 .



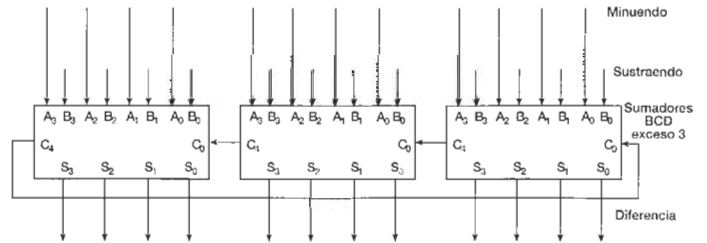
• **Resta en el código BCD exceso tres.** Se realiza de forma semejante a la empleada en el código BCD natural. El código BCD exceso tres ofrece la ventaja para la resta de que el complemento a nueve del número se obtiene de forma inmediata: no hay más que complementar todos los bits de su codificación.

Ejemplos:

2 362	0101	0110	1001	0101		Acarreo
- 1 829	1011	0100	1010	0011		
0 533	10000	1011	10011	1001		Corrección
	0011	1101	0011	1101		
	0011	1000	0110	0110		
	+ (0)	(5)	(3)	(3)		

1 829	0100	1011	0101	1100		Acarreo
- 2 362	1010	1001	0110	1010		
- 0 533	1111	10100	1100	10110		Corrección
	1101	0011	1101	0011		
	1100	0111	1001	1001		
	- (0)	(5)	(3)	(3)		

Para restar dos números en el código BCD exceso tres basta introducir en un circuito que genere la suma correcta según este código el minuendo y el resultado de complementar bit a bit el sustraendo (con lo cual se obtiene ya directamente el complemento a nueve en código BCD exceso tres), y realimentar el acarreo de orden superior hasta el acarreo de orden inferior. A continuación se representa una cadena de sumadores BCD exceso tres conectados de manera que generen la resta de dos números de tres dígitos cada uno expresados según este código.



Pág. 420

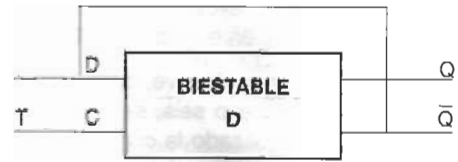
- Recordemos que los biestables T (toggle) cambian de estado durante los pasos de 0 a 1 y de 1 a 0 de una señal de entrada que se designa como T. Su tabla de fases es la siguiente:

0	1	S
①	2	0
3	②	0
③	4	1
1	④	1

Esta tabla no admite fusión, y se asignan las variables internas Q_1 y Q_2 conforme se indica en la tabla siguiente, con objeto de evitar transiciones no adyacentes:

Q_1	Q_2
0	0
1	0
1	1
0	1

2. Los biestables T no se encuentran disponibles comercialmente, siendo necesario formarlos a partir de otros. En el esquema siguiente se muestra un biestable T obtenido a partir de uno tipo D:



SOLUCIONES a las Actividades de Síntesis

1. Un circuito *combinacional* es aquel sistema lógico cuya salida depende en todo momento de los valores binarios que adopten las variables de entrada. En estos circuitos no es necesario tener en cuenta el tiempo. Son funciones lógicas, que se pueden representar en una tabla de verdad, y simplificar mediante la lógica booleana o por métodos como el de Karnaugh. Los sistemas lógicos combinacionales no permiten almacenar el estado de las entradas en un instante determinado y utilizarlo para una posterior toma de decisiones.

Un circuito *secuencial* es aquél cuya salida en cualquier instante depende no solamente de la entrada al circuito en ese momento, sino también de la evolución anterior; es decir, de la secuencia de entradas a que estuvo sometido. Con otras palabras, memoriza el estado de las entradas y lo convierte en un estado interno del propio sistema. Un sistema lógico secuencial reacciona ante secuencias de estados de entrada de una forma determinada que depende de su construcción física; por lo tanto, es un sistema automático, que cambia sin que intervenga la pauta de un reloj (*asíncrono*: más rápido y complejo), o bien está controlado por medio de un reloj (*síncrono*: lento, pero muy sencillo de diseñar).

Un circuito combinacional presenta una tabla de verdad como la que figura a la derecha. Esto es: la salida depende de las entradas en cada instante, y cada par de entradas sólo puede dar una salida.

Entradas		Salida
E_1	E_2	S
0	0	0
0	1	1
1	0	1
1	1	1

Un circuito secuencial, en cambio, podría presentar una tabla de verdad como la que se reproduce en segundo lugar. Esto es: la salida no sólo depende de las entradas en cada instante, sino de la secuencia de entradas anteriores. De esta manera,

Entradas		Salida
E_1	E_2	S
1	1	1
0	1	1
0	0	0
0	1	0

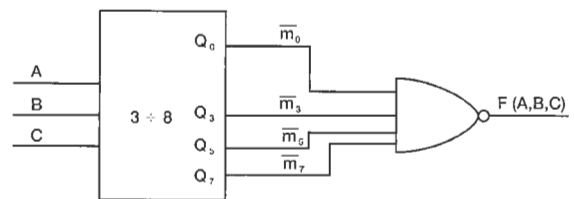
para la pareja de entradas (0,1) la salida es distinta según la secuencia anterior (1,1) o (0,0). Es decir, si antes entró (1,1), la pareja (0,1) da un 1 en la salida. En cambio, si entró la (0,0), la pareja (0,1) tiene como salida un 0.

Como ejemplos de circuitos combinacionales se pueden citar el control de un ascensor o una mesa de votaciones (sumador). Circuitos secuenciales son, por ejemplo, un contador o un control de luces de desplazamiento.

2. Un *decodificador* es un circuito integrado combinacional que presenta n entradas o líneas de selección y N ($\leq 2^n$) salidas o líneas seleccionadas.

Por lo tanto, la respuesta correcta es la a).

3. Es un decodificador de nivel activo bajo:



Se cumple: $f = \overline{m_0} \cdot \overline{m_3} \cdot \overline{m_5} \cdot \overline{m_7}$

y teniendo en cuenta las leyes de Morgan, resulta:

$$f(A, B, C) = m_0 + m_3 + m_5 + m_7$$

4. a) *Utilizando puertas simples.*

Designemos por DCBA a las entradas correspondientes al número decimal codificado en BCD, siendo A el bit de menor peso.

En la tabla se recogen todas las combinaciones posibles junto con su significado en código BCD, representando por E (error) aquellas combinaciones (en total 6) que carezcan de significado decimal. El problema que se propone quedará resuelto si se obtiene una función de error E que tome el valor lógico «1» cuando en las entradas aparezca alguna de esas seis combinaciones.

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

La función E valdrá:

$$E = D\bar{C}\bar{B}\bar{A} + D\bar{C}B\bar{A} + DC\bar{B}\bar{A} + DC\bar{B}A + DCB\bar{A} + DCBA$$

y su representación en un mapa de Karnaugh conduce a:

DC \ BA	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Para el cálculo de la expresión lógica mínima, tengamos en cuenta que, una vez efectuadas las correspondientes asociaciones:

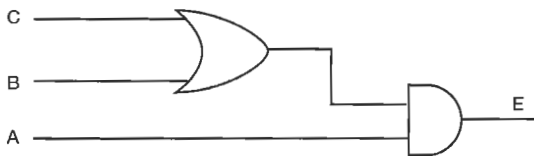
- Las casillas 3, 7, 11 y 15 eliminan las variables A y B. El resultado es: D·C.
- Las casillas 10, 11, 14 y 15 eliminan las variables A y C. El resultado es: D·B.

De esta manera la función lógica mínima que se obtiene es:

$$E = D \cdot C + D \cdot B = D \cdot (C + B)$$

Se puede comprobar que esta función es equivalente a la anterior construyendo su tabla de verdad y viendo que coincide con la anteriormente obtenida.

El diagrama lógico $E = D \cdot (C + B)$ se puede implementar utilizando puertas OR y AND de la forma siguiente:



b) Por medio de un decodificador y una puerta simple.

De la tabla de verdad se obtiene directamente:

$$E = D \cdot (\bar{C}\bar{B}\bar{A} + \bar{C}B\bar{A} + C\bar{B}\bar{A} + C\bar{B}A + CB\bar{A} + CBA) = D \cdot E'$$

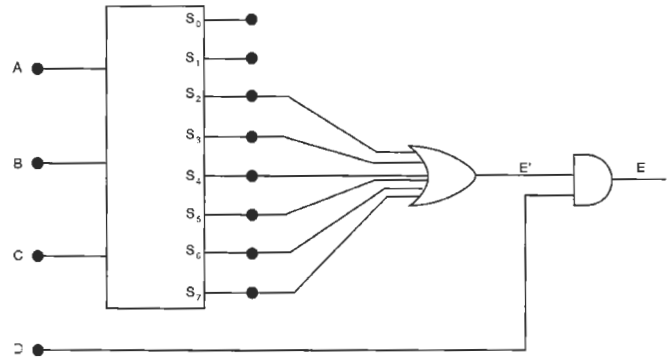
representando

$$E' = \bar{C}\bar{B}\bar{A} + \bar{C}B\bar{A} + C\bar{B}\bar{A} + C\bar{B}A + CB\bar{A} + CBA$$

el OR de las últimas salidas de un decodificador de 3 entradas y 8 salidas, conforme se aprecia en la siguiente tabla:

C	B	A	
0	0	0	S_0
0	0	1	S_1
0	1	0	S_2
0	1	1	S_3
1	0	0	S_4
1	0	1	S_5
1	1	0	S_6
1	1	1	S_7

Por lo tanto, la función E se puede implementar fácilmente utilizando un decodificador 3+8 y una puerta AND, de la forma:



5. Un **codificador** es un circuito integrado combinacional que presenta $N (\leq 2^n)$ líneas de entrada y n líneas de salida, en las que aparece el código binario correspondiente a la línea de entrada activa en ese momento.

Por lo tanto, la respuesta correcta es la c).

6. Un **multiplexador** es un circuito combinacional que presenta n líneas de control, $N (\leq 2^n)$ entradas y una salida. En la salida aparece el valor de la línea de entrada seleccionada por el código binario de las líneas de control.

Por lo tanto, la respuesta correcta es la b).

7. El circuito es un multiplexador de 8 entradas y 3 controles.

D \ ABC	000	001	010	011	100	101	110	111
0	1	0	1	0	0	0	1	0
1	1	0	0	0	1	1	1	1
	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7

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

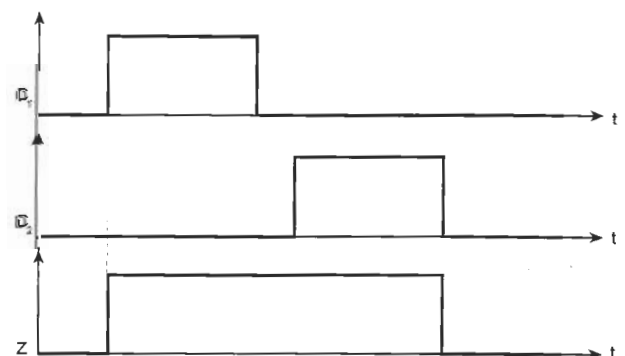
$$f = \sum D_i \cdot m_i =$$

$$1 \cdot m_0 + 0 \cdot m_1 + \bar{D} \cdot m_2 + 0 \cdot m_3 + D \cdot m_4 + D \cdot m_5 + 1 \cdot m_6 + D \cdot m_7 =$$

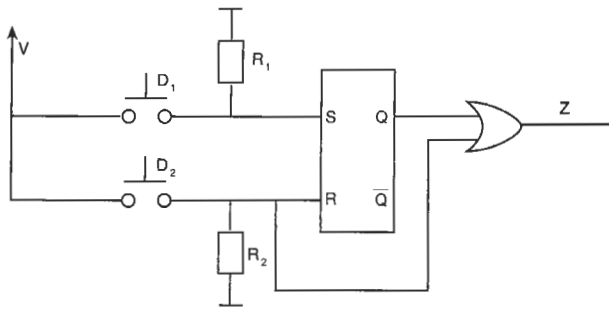
$$= \boxed{m_0 + \bar{D} \cdot m_2 + D \cdot m_4 + D \cdot m_5 + m_6 + D \cdot m_7}$$

8. Designemos los dos detectores por D_1 (el de cabeza) y D_2 (el de cola), y consideremos que $Z_1 \rightarrow 1$ indica la barrera bajada y $Z_0 \rightarrow 0$, cuando está subida.

El diagrama lógico de tiempos es:



y el circuito:



9. El circuito del enunciado no es igual a un biestable T. Para que lo fuese debería llevar un reloj (*clock*) (véase diagrama adjunto).

