

TEMA 4. MÓDULOS COMBINACIONALES.

4.1. Módulos combinacionales básicos MSI.

Los circuitos combinacionales realizados con puertas lógicas implementan funciones booleanas, pero no son los únicos elementos capaces de ello. En este tema veremos que los llamados *módulos combinacionales* pueden implementar funciones booleanas y cumplir otras misiones específicas más.

Se tratarán como bloques funcionales que realizan una función determinada, si bien se comprobará cómo es posible realizar su función específica por medio de puertas lógicas.

Las siglas MSI (**M**edium **S**ize of **I**ntegration) se refieren a aquellos circuitos de Media Escala de Integración con un número de puertas entre 10 y 100.

Por tanto, primeramente veremos los módulos combinacionales como bloques funcionales, posteriormente comprobaremos que un conjunto de puertas lógicas debidamente conectadas realizan la función del módulo, y, por último, estudiaremos cómo se pueden implementar funciones booleanas usando módulos combinacionales.

Los módulos combinacionales disponen además de las llamadas *señales de control*, que permiten que el circuito funcione normalmente, si están activadas, o permanezca inactivo si las señales de control están desactivadas.

4.1.1. Codificador.

El Codificador es un módulo con 2^n entradas y n salidas de modo que en las n líneas de salida aparece codificado en binario aquella entrada que se activa, que puede ser a nivel alto o a nivel bajo. En la Figura 1 aparece la representación de su bloque funcional, su realización con puertas lógicas y su tabla de verdad, para $n=3$. Se denominan abreviadamente COD $2^n \times n$, siendo $n \in \mathbb{N}$. 2^n es el número de entradas, mientras que n es el número de salidas. Este tipo de denominación abreviada la veremos en otros módulos más adelante.

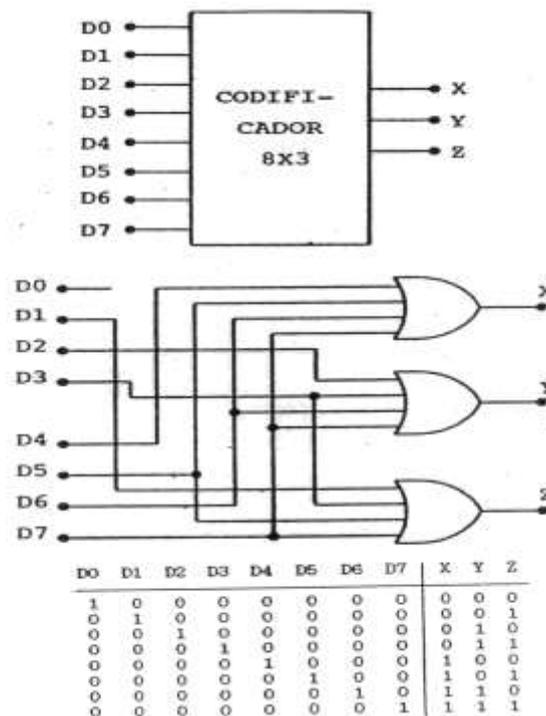


Figura 1

La tabla de verdad del codificador permite encontrar las ecuaciones lógicas de las funciones de salida X, Y, Z, que son:

$$X = D4 + D5 + D6 + D7 \quad (1)$$

$$Y = D2 + D3 + D6 + D7 \quad (2)$$

$$Z = D1 + D3 + D5 + D7 \quad (3)$$

que corresponde a la implementación del codificador con puertas. Este codificador, denominado COD 8x3, tiene tanto las entradas como las salidas activas a nivel alto.

Obsérvese que tal como está concebido este codificador, sólo se puede activar una señal de entrada. ¿Qué ocurriría si se activaran más de una señal en el codificador de la Figura 1? La respuesta se puede extraer de las ecuaciones 1, 2 y 3. Supongamos, por ejemplo, que D1=1 y D4=1. En este caso, resultaría que X=1, Y=0, Z=1, que corresponde con el caso de que D5 fuera 1 y las restantes entradas fueran 0. Por tanto, si la salida XYZ = 101, podemos asegurar que la entrada activada es D5 sólo si se ha activado una entrada, pero si se activan más entradas, queda la duda de saber cuáles han sido las entradas activadas. Este codificador tan sencillo, pero con este inconveniente, se conoce como *codificador sin prioridad*.

Para resolver este problema, existen codificadores para los que, en caso de activar más de una entrada, la salida será normalmente la que corresponde a la mayor de ellas, independientemente de las restantes inferiores. A estos codificadores se les denomina *codificadores con prioridad*. En las Figura 2a y 2b aparecen codificadores con prioridad, en el primer caso con su tabla de verdad y expresiones booleanas, y en el segundo caso con una implementación con puertas lógicas junto con tu tabla de verdad.

D3	D2	D1	D0	X	Y
0	0	0	0	-	-
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
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

$$X = D2 + D3$$

$$Y = D3 + D1 \cdot \overline{D2}$$

Figura 2a

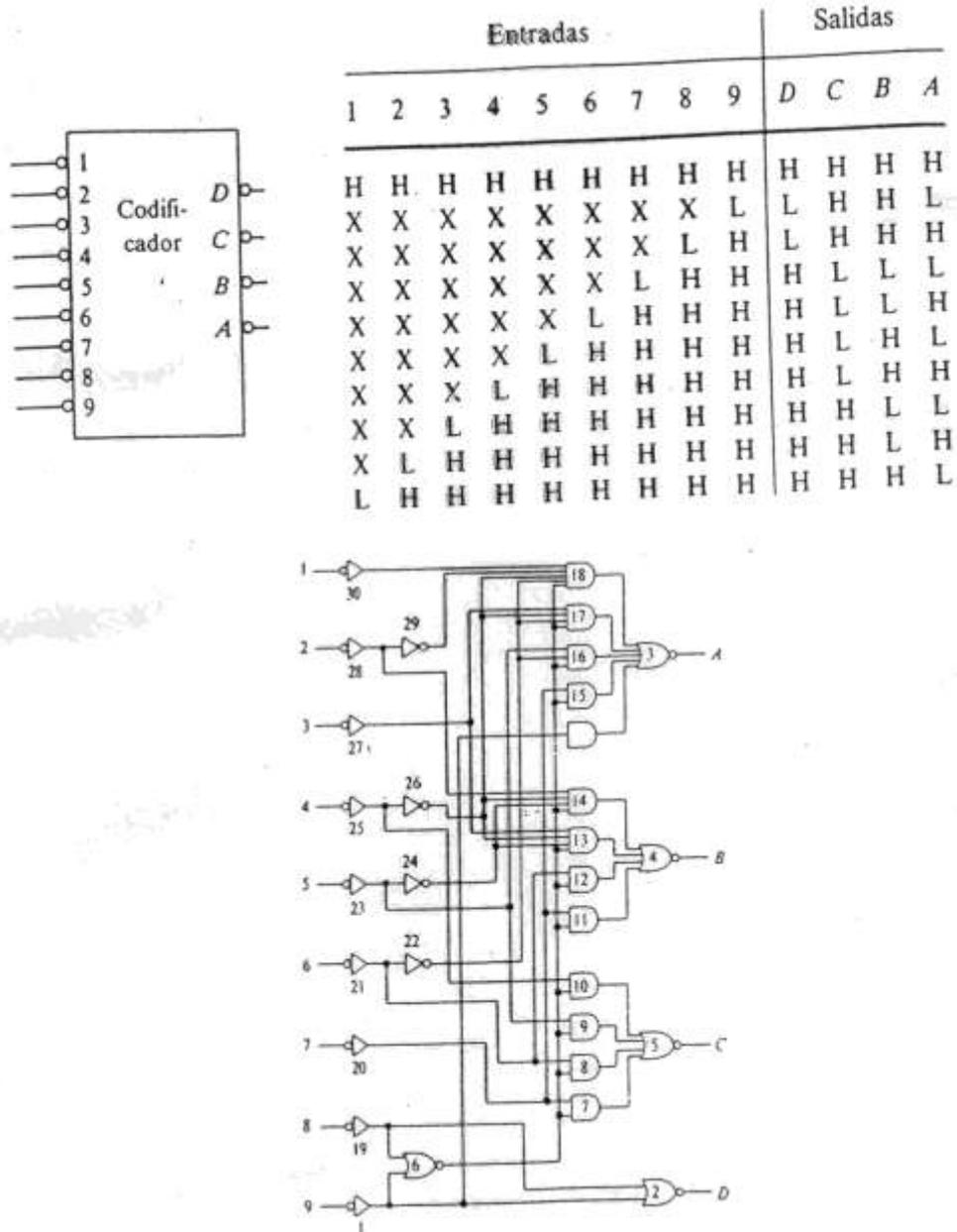


Figura 2b

4.1.2. Decodificador.

El Decodificador es un módulo combinacional con n entradas y 2^n salidas, de forma que dependiendo de qué valor binario se coloque en las entradas, se activará una, y sólo una, salida mientras que las demás estarán desactivadas. La activación puede ser por nivel alto o por nivel bajo. Realiza la función inversa del codificador.

La representación modular y el circuito con puertas lógicas de un decodificador de 2 entradas y 4 salidas aparecen en la Figura 3, junto con su tabla de verdad. Se denominan DEC $n \times 2^n$.

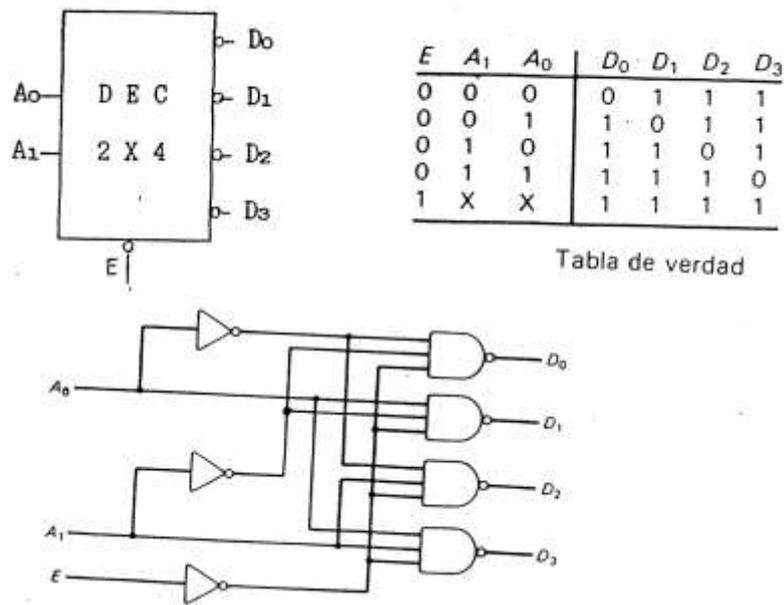


Figura 3

Vemos que aparece una nueva señal de entrada llamada E. Dicha señal permite que el módulo funcione como tal o que no funcione. Se denomina entrada de HABILITACION (ENABLE), y puede ser activa a nivel alto o a nivel bajo.

Esta entrada permite una de las siguientes actuaciones:

- 1) Inhibir el funcionamiento del módulo, de modo que su comportamiento sea similar a que el módulo no esté conectado.
- 2) Permitir que el módulo funcione con normalidad, cumpliendo sus funciones específicas.

En el caso concreto de la Figura 3, si la entrada E toma el valor 1 inhibe el funcionamiento del módulo, ya que al estar conectada a una entrada de todas las puerta NAND a través de un inversor, sus correspondientes salidas serían 1.

Sin embargo, si E se pone a 0, no tiene ningún efecto sobre el normal funcionamiento del módulo, ya que el 1 es el elemento neutro de la multiplicación lógica u operación AND/NAND.

4.1.3. Multiplexor.

El multiplexor es un módulo combinacional que selecciona a su salida una de las 2^n posibles entradas usando para ello las n líneas de selección.

Se comportan como si fuera un selector digital, colocando a la salida el nivel lógico que hubiera a la entrada seleccionada.

En la Figura 4 aparece su realización con puertas lógicas y su representación modular para $n=2$. Se denominan MUX $2^n \times 1$.

La tabla de verdad, con entrada y salida activa a nivel bajo, viene dada por:

C1	C0	E0	E1	E2	E3	S
0	0	K	X	X	X	K
0	1	X	K	X	X	K
1	0	X	X	K	X	K
1	1	X	X	X	K	K

donde K puede tomar cualquier valor binario 0, 1. X significa indefinido.

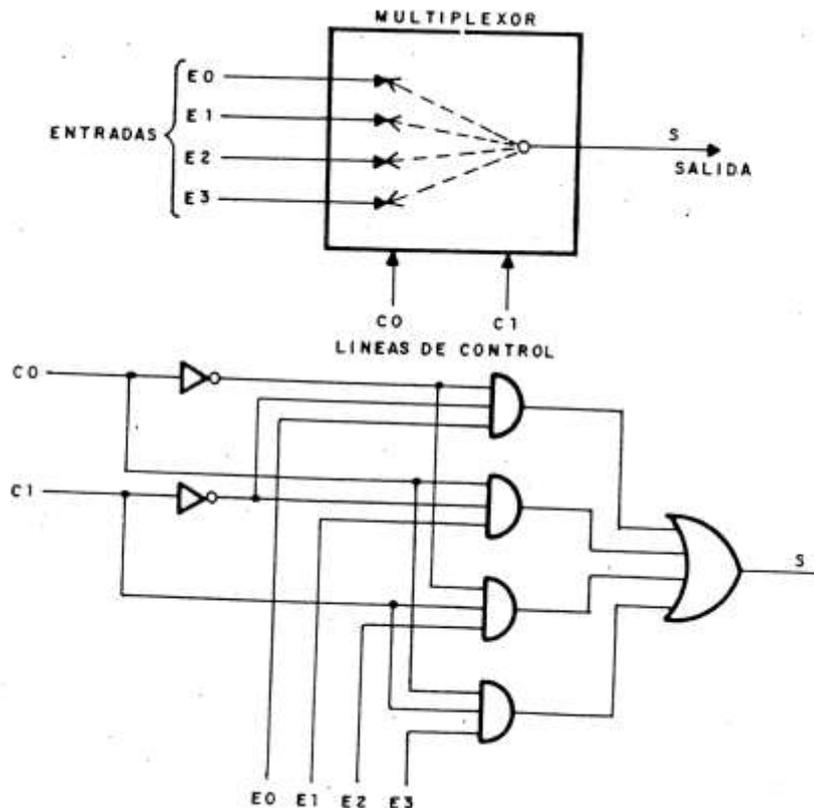


Figura 4

4.1.4. Demultiplexor.

El Demultiplexor es un módulo combinacional que envían el valor binario de su entrada a la salida que se seleccione con las líneas de selección. Realiza la función inversa del multiplexor.

Tiene sólo una entrada y 2^n salidas, con n líneas de selección.

En la Figura 5 aparece la implementación con puertas lógicas y su representación modular, así como la tabla de verdad, para $n=2$. Se denominan DEMUX 1×2^n .

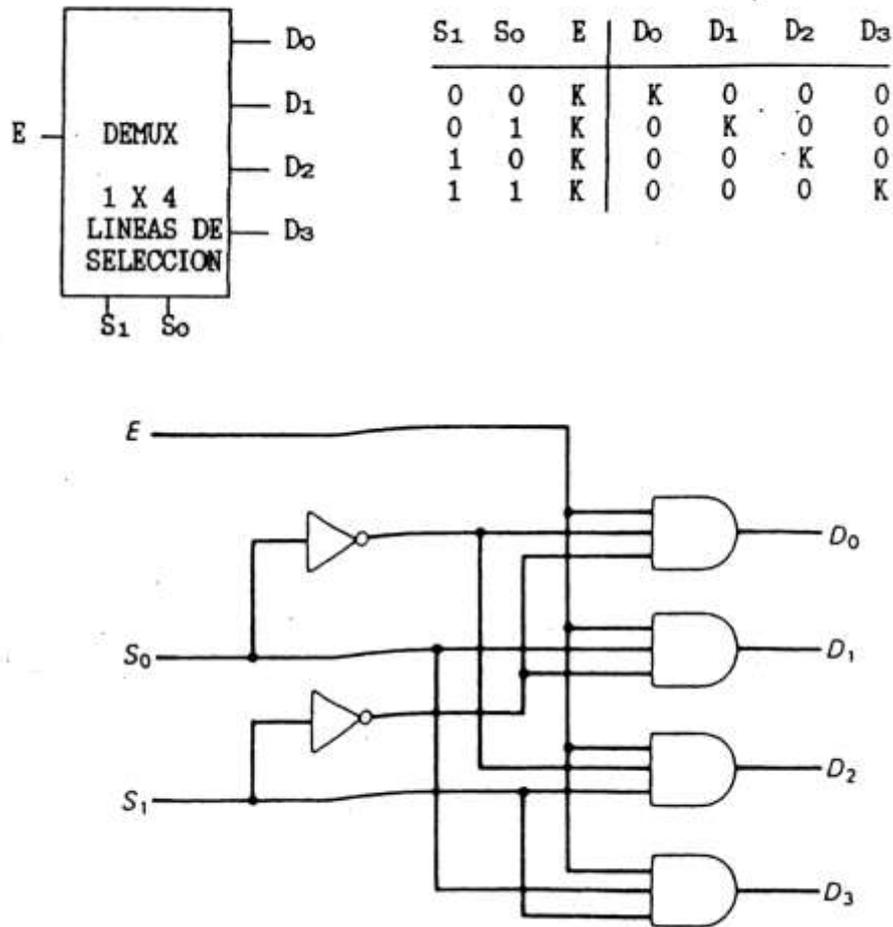


Figura 5

Llegado este momento conviene señalar que la diferencia entre un DEC 2x4 y un DEMUX con 2 líneas de selección, por ejemplo, es más conceptual que funcional. Un DEMUX 1x4 se puede ver como un DEC 2x4 en donde las 2 líneas de selección del DEMUX equivalen a las 2 entradas del DEC 2x4, mientras que la entrada del DEMUX 1x4 se corresponde con la entrada de habilitación del DEC 2x4. Los circuitos lógicos de ambos módulos son idénticos. Por tal motivo, a veces se habla de decodificador/demultiplexor. Sin embargo, a nivel conceptual conviene diferenciarlos.

4.2. Redes modulares.

Son redes de módulos combinacionales utilizadas para asociar dichos módulos y obtener otros de mayores prestaciones.

4.2.1. Redes de codificadores.

Asociar dos codificadores de 4x2 (más uno de 2x1) para construir un codificador de 8x3. S es una salida de control que se activa cuando se usa un determinado COD 4x2.

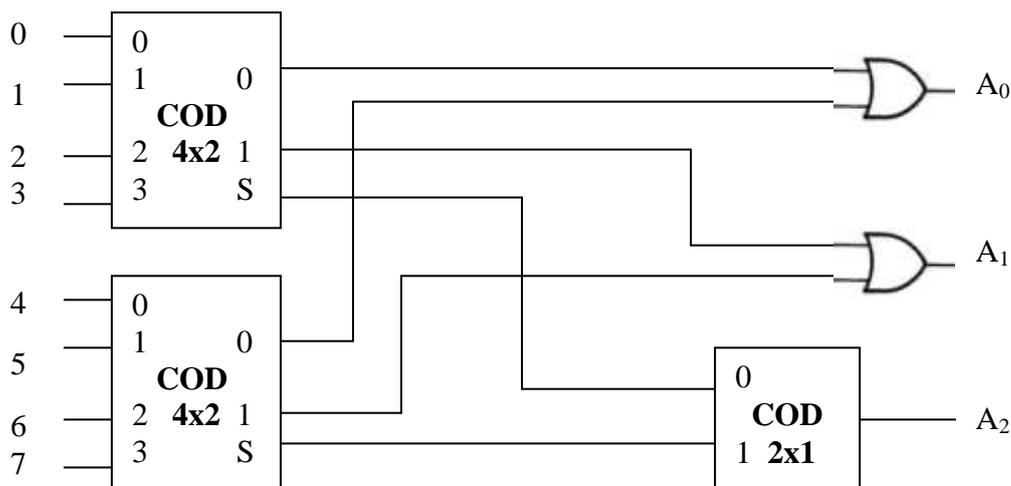


Figura 6

4.2.2. Redes de decodificadores.

La Figura 7 muestra una asociación de dos DEC 2 x 4.

El conjunto se puede suponer un decodificador de 3 x 8, ya que tiene 3 líneas de entradas, las dos comunes más la línea A_2 , y las 8 salidas, 4 de cada uno de ellos.

Cuando A_2 está a nivel alto el decodificador de la parte inferior de la figura funciona con normalidad, siempre que la entrada habilitadora sea activa a nivel alto, mientras que el decodificador de la parte superior está inhibido. En este caso, alguna de las líneas de salida $D_4 - D_7$ se activarán dependiendo de la entrada binaria en A_0 y A_1 .

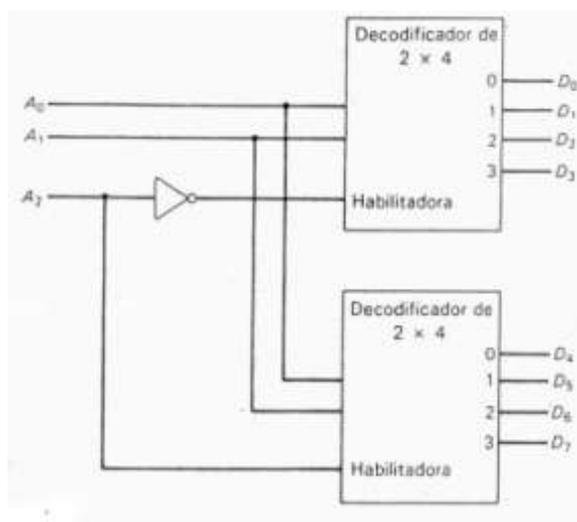


Figura 7

Cuando A_2 está a nivel bajo el decodificador de la parte superior de la figura está habilitado para funcionar, mientras que el decodificador de la parte inferior de la figura está inhibido. Alguna de las salidas $D_0 - D_3$ se activarán dependiendo de la entrada binaria en A_0 y A_1 .

De esta forma siempre habrá un decodificador habilitado y otro inhibido.

El conjunto se comporta como un DEC 3 x 8, en donde la línea de entrada de mayor peso, A_2 , es la que determina según su valor lógico, cuál de los dos decodificadores está inhibido o habilitado. El inversor asegura que mientras uno de ellos esté habilitado, el otro esté inhibido y viceversa.

Existe una forma de asociar decodificadores en forma de árbol. La implementación se realiza en dos niveles y las variables de entrada se dividen en dos grupos. El primer grupo entra al decodificador del primer nivel, mientras que el segundo grupo actúa como entrada al conjunto de decodificadores que están en el segundo nivel.

El primer decodificador proporciona un conjunto de salidas que se dirigen a las entradas de habilitación (Enable) de los decodificadores del segundo nivel. En la Figura 8 se puede ver un DEC 4x16 contruido a partir de cinco DEC 2x4. Las variables de mayor peso se conectan en el decodificador del primer nivel.

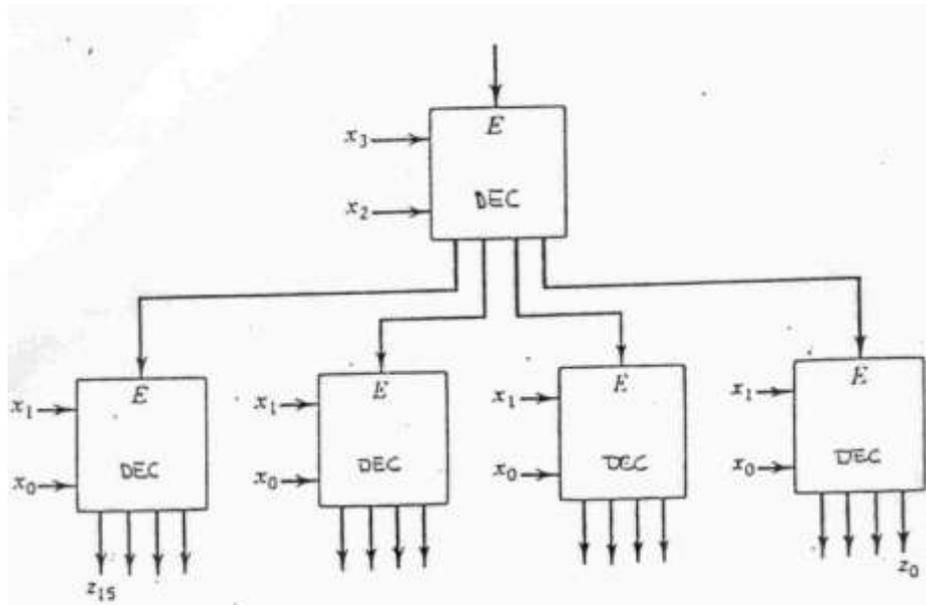


Figura 8

4.2.3. Redes de multiplexores.

Los multiplexores también se pueden asociar para ampliar sus prestaciones. La Figura 9 muestra una asociación de dos multiplexores de 16x1 para constituir uno de 32x1. Las salidas de cada multiplexor de 16x1 son las entradas del MUX de 2x1, mientras que la línea de selección de más peso, E_4 , es precisamente la línea de selección de MUX de 2x1. Suponemos que E es activa a nivel bajo. Por dicho motivo la colocamos a nivel bajo (masa).

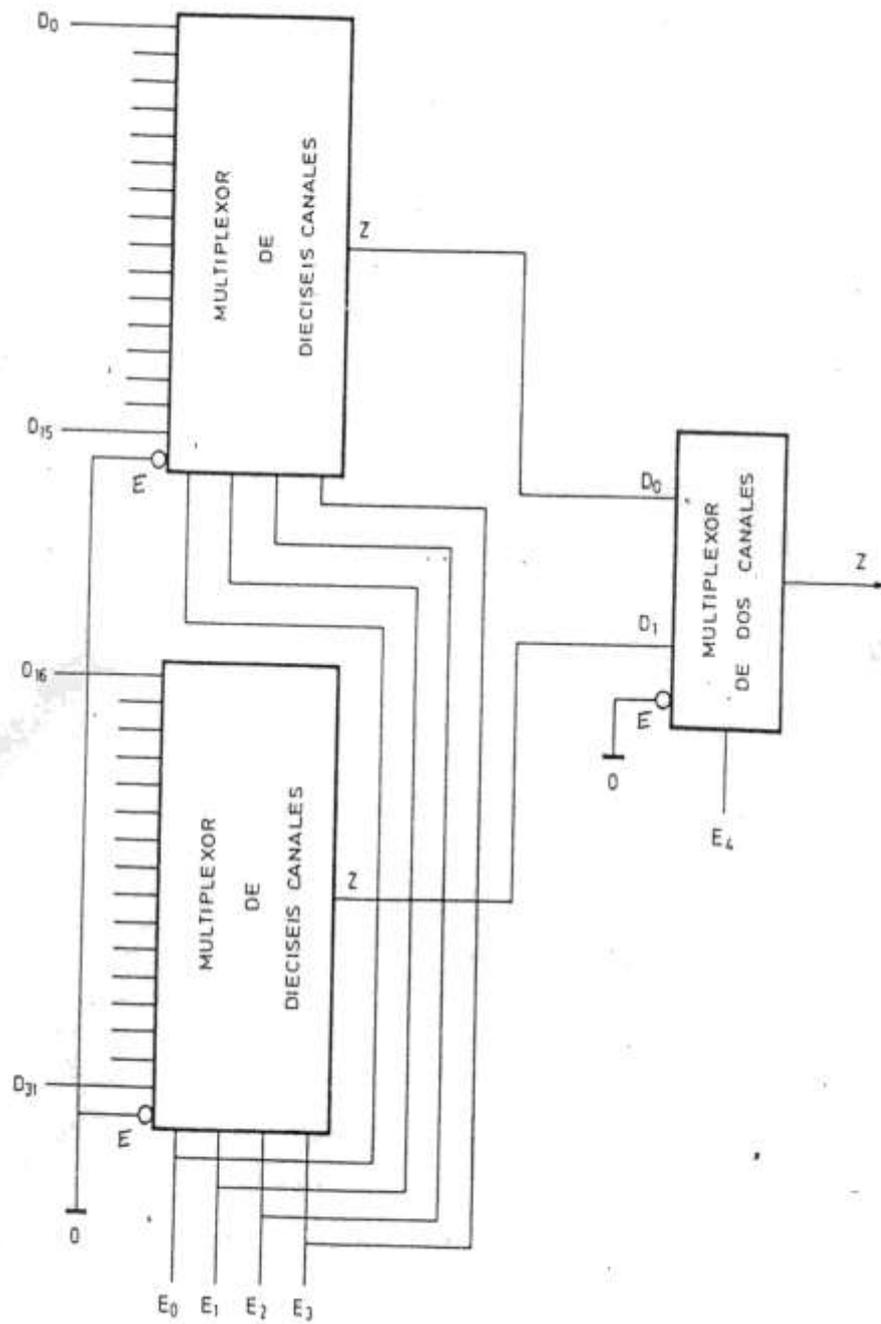


Figura 9

También es posible construir un MUX 64x1 a partir de cuatro MUX 16x1 y un MUX 4x1, como lo muestra la Figura 10. El principio de funcionamiento es el mismo que en el caso anterior.

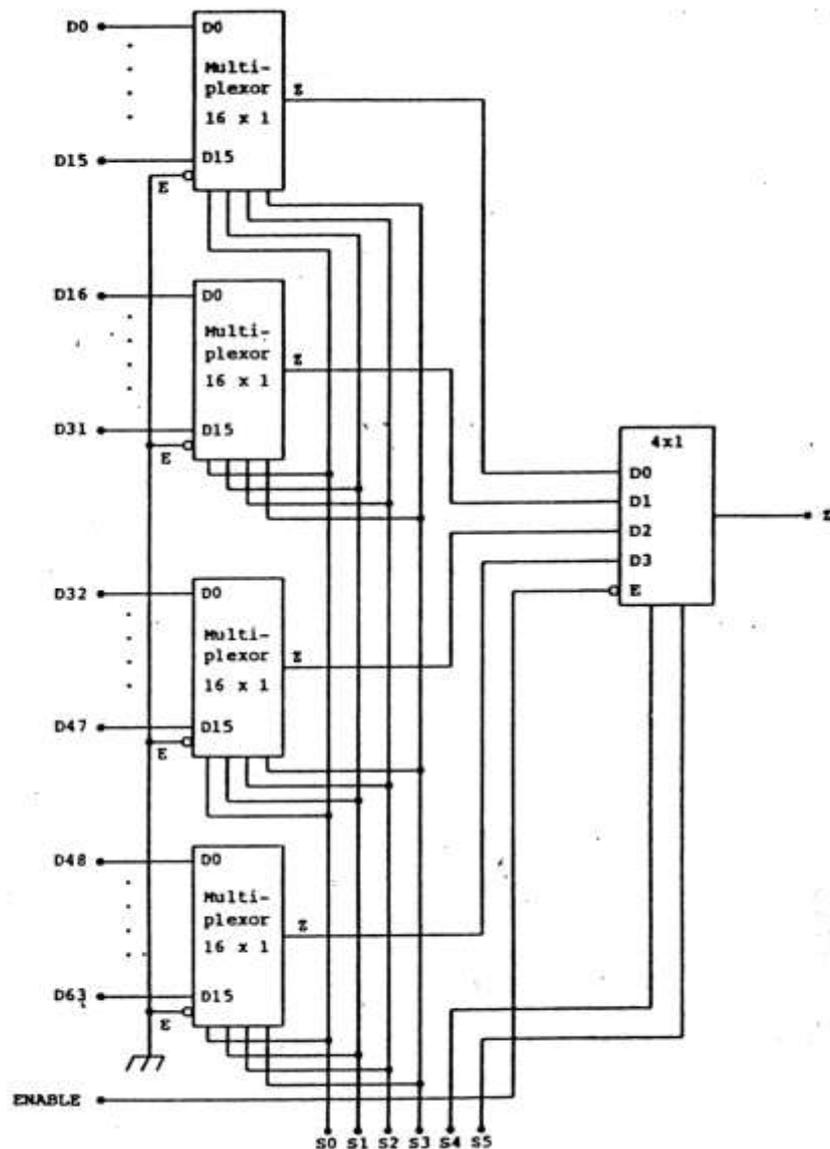


Figura 10

Con esta idea de colocar multiplexores en dos niveles es posible asociar multiplexores de más entradas de cuantas formas se nos ocurran.

4.3. Implementación de funciones booleanas con módulos combinacionales.

Veamos cómo implementar funciones booleanas con módulos combinacionales.

Para implementar una función dada su tabla de verdad con un decodificador, se toma un decodificador que tenga tantas entradas como variables booleanas tenga la función, y se suman con una puerta OR las salidas que corresponden a los términos con valor 1.

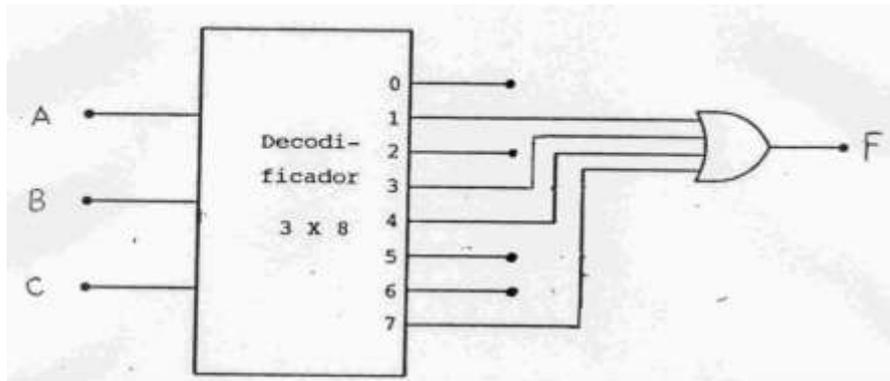


Figura 11

Para todas las combinaciones de las variables que dan lugar a salida 1, y sólo para ellas, aparece un 1 en alguna de las entradas de la puerta OR, lo que da lugar a que $F=1$. En los restantes casos, los 1 aparecen en distintas salidas de las conectadas a la puerta OR, con lo que la puerta OR tendrá todas sus entradas a 0, y su salida a 0. Por tanto, basta con sumar con una puerta OR aquellas salidas que toman el valor 1 en la tabla de verdad. Se puede decir que un decodificador es un generador de minterminos. Genera todos los minterminos posibles, y con la puerta OR se suman los que se necesiten (Figura 11).

Si las salidas del decodificador fueran activas a nivel bajo, se utilizaría una puerta NAND en aquellas salidas correspondientes a combinaciones que dieran lugar al valor 1. En la Figura 12 aparece la función implementada tanto con puertas NAND (con salidas activas a nivel bajo) como con puertas OR (con salidas activas a nivel alto):

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

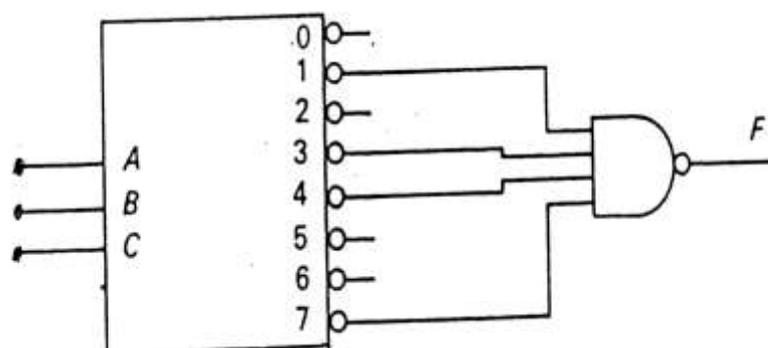


Fig 12

También se pueden implementar funciones booleanas con multiplexores. Para ello se procede según los casos:

1) Si el número de variables booleanas es igual que el número de líneas de selección del multiplexor. Se conectan las variables booleanas directamente a las líneas de selección en el mismo orden en cuanto al peso. En las entradas del multiplexor se conectan directamente las constantes lógicas obtenidas de la tabla de verdad.

De esta forma, cada vez que en las líneas de selección se coloque una de las posibles combinaciones binarias, el dato que se encuentre en la entrada correspondiente aparece en la salida.

Veamos un ejemplo con una función de tres variables de entrada en la Figura 13.

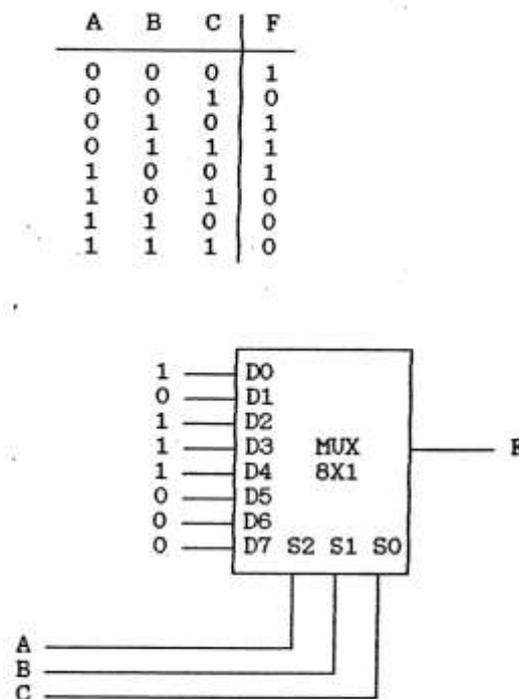


Fig 13

2) Si el número de variables booleanas es mayor que el número de líneas de selección del multiplexor. Se toma un multiplexor que tenga un número de entradas de selección menor en una unidad al número de variables booleanas. Se deja una variable booleana, normalmente la de menor peso, (aunque puede ser cualquier otra), sin conectar a las entradas de selección del MUX, mientras que las restantes variables se conectan a las líneas de las entradas de selección, en el mismo orden de pesos.

Posteriormente, para una cierta pareja de valores de F se compara el valor que toma la función F con los valores que toma la variable booleana que no se conectó a las entradas de selección. Se pueden dar sólo 4 casos:

- 1) Que $F=0$ para los dos valores de la variable booleana no conectada.
- 2) Que $F=1$ para los dos valores de la variable booleana no conectada.
- 3) Que F sea igual a los valores de la variable.
- 4) Que F sea el negado de los valores de la variable.

Según cuál de estos casos se dé en cada pareja, la entrada que quedó sin conectar a las líneas de selección se pondrá, respectivamente, a:

- 1) Valor constante 0.
- 2) Valor constante 1.
- 3) Valor de la variable no conectada.
- 4) Valor negado de la variable no conectada.

Veamos un ejemplo a partir de la tabla de verdad en la Figura 14:

Cuando $A=0$, $B=0$ y $C=0$, la salida es la misma que la variable D , luego a la entrada 0 le ponemos la variable D .

Cuando $A=0$, $B=1$ y $C=0$, la salida es la negada de la variable D , luego a la entrada 2 le ponemos la negada de la variable D negada.

Cuando $A=0$, $B=1$ y $C=1$, la salida es 0 independientemente de D , luego a la entrada 3 le ponemos la constante lógica 0.

Cuando $A=1$, $B=1$ y $C=1$, la salida es 1 independientemente de D , luego a la entrada 7 le ponemos la constante lógica 1.

Lo mismo se puede hacer con las restantes no citadas.

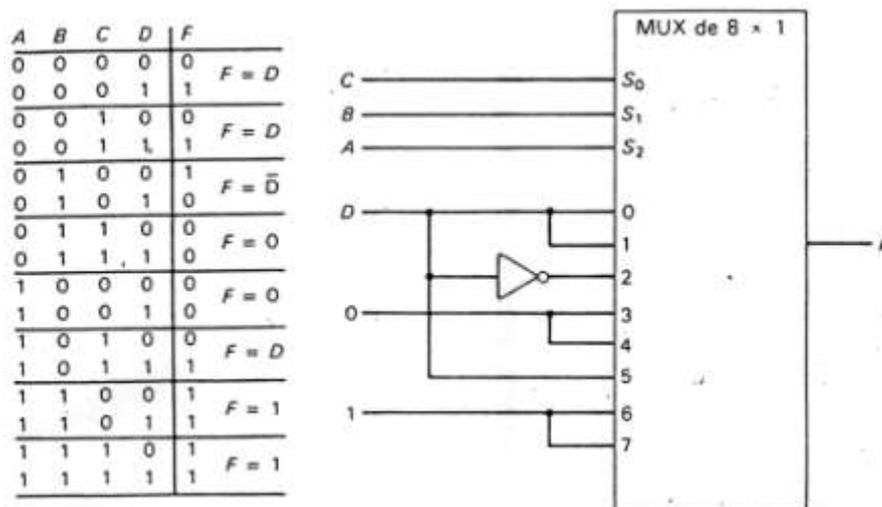


Figura 14

Finalmente, el criterio para decidir cuándo usar un decodificador y cuándo usar un multiplexor para implementar una función booleana con módulos, podría ser el siguiente:

El método del decodificador necesita una puerta para cada función de salida, pero sólo se necesita un decodificador para generar todos los minterminos.

El método del multiplexor usa unidades de tamaño más pequeño, pero requiere un multiplexor para cada función de salida.

Por tanto, parece razonable suponer que para circuitos combinacionales con un número pequeño de salidas deberían utilizarse multiplexores, mientras que para el caso de un número elevado de funciones de salida probablemente se usarían menos circuitos integrados con el método del decodificador.

Supuesto práctico.

Se pretende diseñar un sistema de auxilio para un barco que se encuentra en alta mar. Dicho sistema estará compuesto por un circuito digital a diseñar que activará un transmisor de radio con una llamada de socorro S.O.S. en Morse.

El código Morse de S.O.S. está compuesto por tres puntos, seguido de tres rayas y acabado en tres puntos. El punto se corresponde con un pulso a nivel alto, la raya se corresponde con tres pulsos consecutivos a nivel alto, y el tiempo intermedio con niveles bajos. Es decir, un S.O.S. sería una señal digital de la forma de la Figura 15:

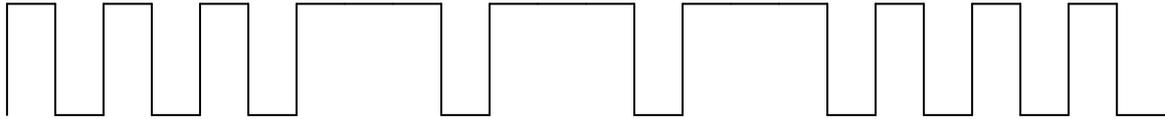


Figura 15

Para ello se dispone de un contador de 5 bits que cuenta en sentido ascendente cuando se activa la señal de socorro.

Diseñar el circuito combinacional que acepta como entrada los cinco bits de salida del contador y cuya salida genera la señal digital de la Figura 15 que permite activar al transmisor de radio que lanza el S.O.S. al aire.

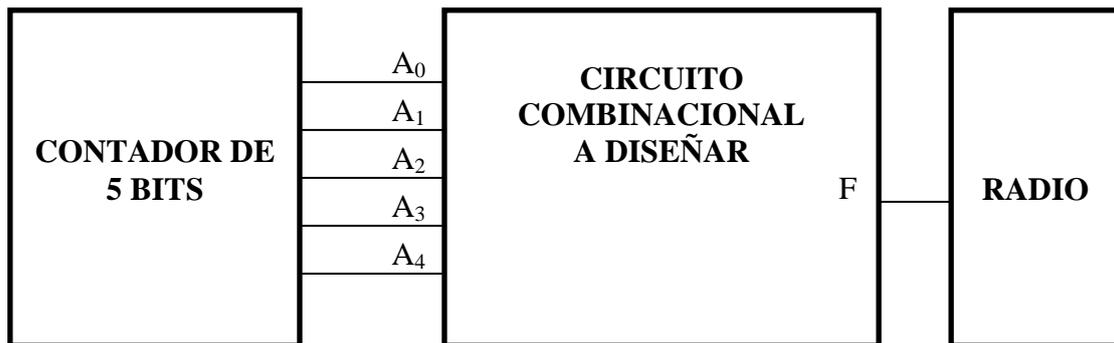


Figura 16

NOTA: Tomar A4 como bit más significativo.

SOLUCION: Con Decodificadores.
Tabla de verdad

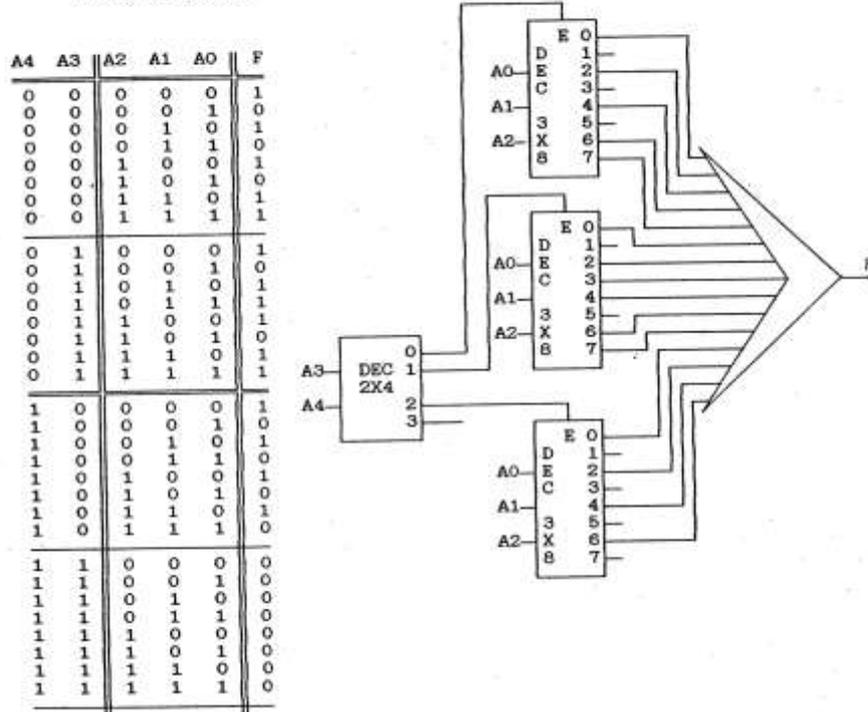


Figura 17

SOLUCION: Con Multiplexores

Tabla de verdad

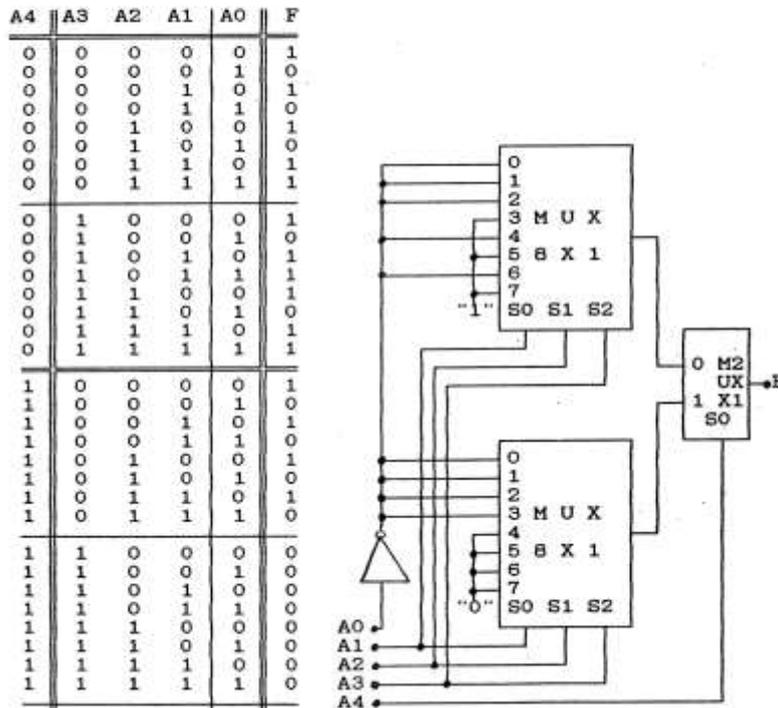


Figura 18

4.4. Descripción de algunos circuitos integrados SSI y MSI comunes.

SSI

Dispositivo	Descripción	Número de puertas	Número de entradas
7400	NAND	4	2
7402	NOR	4	2
7404	NOT	6	1
7408	AND	4	2
7410	NAND	3	3
7411	AND	3	3
7420	NAND	2	4
7421	AND	2	4
7427	NOR	3	3
7430	NAND	1	8
7432	OR	4	2
7486	XOR	4	2

MSI

Dispositivo	Descripción	Observaciones
74148	Codificador con prioridad 8x3	Entrada y salida activa a nivel bajo
74147	Codificador con prioridad 10x4	Entrada y salida activa a nivel bajo
7442	Decodificador 4x10	Entrada activa a nivel alto y salida activa a nivel bajo
7447	Decodificador BCD 7 segmentos	Entrada activa a nivel alto y salida activa a nivel bajo
74150	Multiplexor 16x1	Entrada activa a nivel alto y salida activa a nivel bajo
74151	Multiplexor 8x1	Entrada activa a nivel alto y salida activa a nivel alto y bajo
74153	2 multiplexores 4x1	Entrada y salida activa a nivel alto
74157	4 multiplexores 2x1	Entrada y salida activa a nivel alto
74138	Decodificador 3x8 y demultiplexor	Entrada activa a nivel alto y salida activa a nivel alto y bajo
74154	Decodificador 4x16 y demultiplexor	Entrada activa a nivel alto y salida activa a nivel alto y bajo
74155	Decodificador 3x8 (o dos de 2x4) y demultiplexor	Entrada activa a nivel alto y salida activa a nivel alto y bajo

4.5. Aplicaciones de los módulos combinacionales en los sistemas informáticos.

Codificadores: codificar en binario entradas de periféricos como teclados.

Decodificador: visualizar resultados, direccionamiento memoria o dispositivos CPU, implementación de funciones, implementación de demultiplexores.

Multiplexor / demultiplexor: implementación de funciones, selector digital, conversión paralelo-serie para multiplexación en el tiempo.

EJERCICIOS PROPUESTOS

1) 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:

$$1^a) \overline{a_i} \quad 2^a) \overline{a_i + b_i} \quad 3^a) \overline{a_i \cdot b_i} \quad 4^a) \overline{a_i \cdot b_i}$$

Implementar este sistema mediante una red de módulos en árbol.

2) 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 solo en el escenario debe activar únicamente dos interruptores cualesquiera que no sean consecutivos, de forma que se enciendan todas las luces mientras este actor se encuentre solo 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 sólo 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 (1,0 punto) y llevar a cabo la implementación usando:

- Sólo puertas NOR para la función V (0,25 puntos).
- Sólo puertas NAND para la función A (0,25 puntos).
- Un MUX 4x1 y las puertas necesarias para la función R , usando I_1 e I_2 como señales de control, de las que I_1 es la más significativa (0,25 puntos).
- Un DEC 4X16 para la función M (0,25 puntos).

NOTA: las funciones en a) y b) deben utilizar el mínimo número de puertas posible.

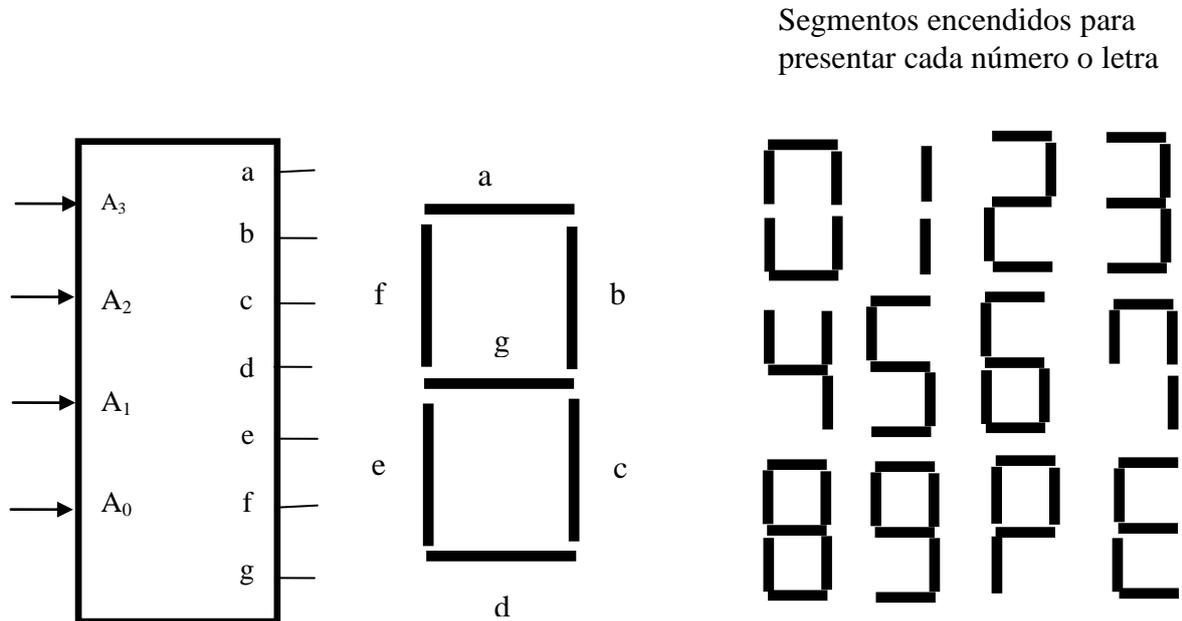
3) Se desea diseñar un circuito combinacional que active un display de 7 segmentos.

El circuito a diseñar admite como entrada las líneas A_3, A_2, A_1 y A_0 , que representan un número binario de 4 bits, siendo A_3 el bit más significativo. Las líneas de salida actúan directamente sobre los siete segmentos del display. Las salidas y segmentos con la misma letra están conectados.

La presentación en el display debe ser la siguiente:

- a) Si el número no es primo y es mayor que 5, debe aparecer el mayor de sus divisores, excluido el propio número, es decir, no considerar que un número es el mayor divisor de sí mismo. Ejemplo: si el número fuera 36, sus divisores son 2, 3, 4, 6, 9, 12 y 18. El mayor sería 18.

- b) Si el número es primo o menor que 4, debe aparecer la letra "P".
c) En el resto del caso debe aparecer la letra "E".



1. Encontrar la tabla de verdad que resuelve el circuito (0,7p).
2. Implementar las funciones de la siguiente forma:
 - a) La función **a** usando sólo una puerta lógica e inversores (0,3p).
 - b) La función **b** usando sólo puertas NAND (0,3p).
 - c) La función **c** usando sólo puertas NOR (0,3p).
 - d) La función **d** usando un MUX 4x1 tomando A_1 y A_0 como líneas de selección S_1 y S_0 , respectivamente (0,5p).
 - e) La función **e** con puertas OR y una puerta AND (0,3p)
 - f) La función **f** con puertas AND y una puerta OR (0,3p).
 - g) La función **g** con un Decodificador con salidas activas a nivel bajo (0,3p).

4) Se desea diseñar un sistema de detección de incendios para un edificio de 2 plantas. Cada planta dispone de 1 pulsador de alarma (P1 para la planta 1 y P2 para la segunda). Además, desde el centro de control puede activarse una señal de inhibición para cada planta (I1 e I2) una vez que se tiene conocimiento de la alarma. Como respuesta a la pulsación, el sistema generará 3 salidas: dos para indicar en qué piso se activó un pulsador (A_1 y A_2) y una tercera de alarma general (A). El mecanismo detallado de funcionamiento del sistema será el siguiente:

- Si se activa el pulsador de un piso se activará la señal de alarma de dicho piso.
- Si la señal de inhibición de un piso determinado está a '1' no se debe activar la alarma de ese piso.
- Si se produce alarma en cualquier piso debe activarse la señal A de alarma general.

Se pide:

- 1) Completar la tabla de verdad.
- 2) Implementar la salida A en forma de producto de sumas.
- 3) Implementar la salida A2 mediante un decodificador activo a nivel alto.
- 4) Implementar la salida A1 mediante un multiplexor 4x1 más la lógica combinacional necesaria, teniendo en cuenta que las señales de control deben ser I1 e I2.

5) Se trata de diseñar un circuito combinacional (Figura 1) que regule las luces de dos semáforos de tráfico colocados en un carretera en obras (Figura 2). El funcionamiento es el siguiente:

1. 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.
2. Cuando los operarios regulan el tráfico manualmente, ponen $E=1$ y
 - 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 $F=1$.

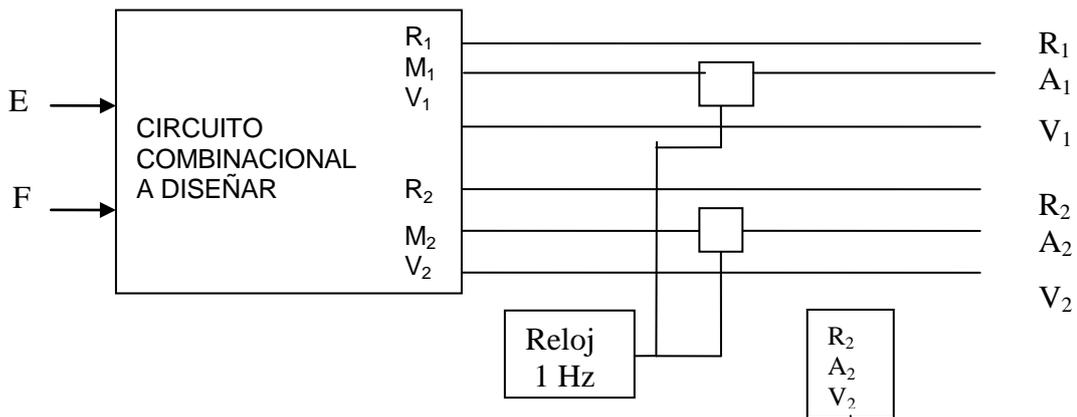


Figura 1

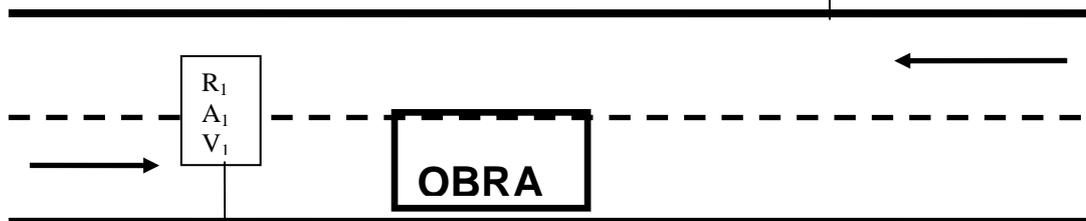


Figura 2

Se pide:

- a) Tabla de verdad (0,3 p)
- b) ¿Qué relación hay entre las luces ámbar de los semáforos y las salidas M_1 y M_2 del circuito combinacional? (0,2 p).
- c) ¿Cuántas funciones booleanas realmente distintas hay en el problema? (0,3 p).
- d) Implementar las funciones que haya usando sólo puertas lógicas. (0,3 p).
- e) Implementar las funciones que haya usando decodificadores con salidas activas a nivel alto (0,3 p).
- f) Implementar las funciones que haya usando decodificadores con salidas activas a nivel bajo (0,3 p).
- g) Implementar las funciones que haya usando multiplexores con 1 línea de selección y conectando la entrada E a dicha línea de selección (0,3 p).
- h) Implementar las funciones que haya usando multiplexores con 1 línea de selección y conectando la entrada F a dicha línea de selección (0,5 p).

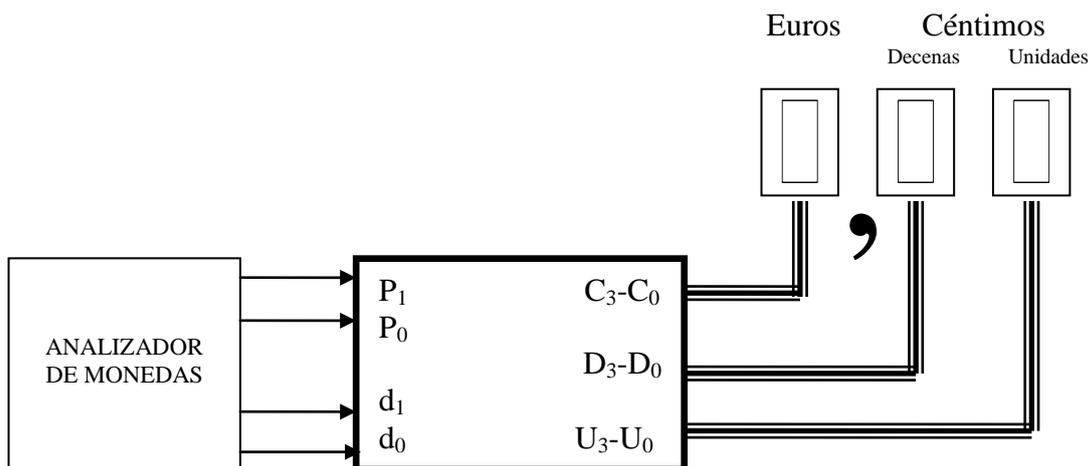
6) Se desea diseñar un circuito combinacional que active tres displays de 7 segmentos en los que se representa el valor en euros de diversos tipos de monedas. Para ello, existe un dispositivo de entrada al circuito que analiza el peso y el diámetro de las monedas (ANALIZADOR DE MONEDAS) y proporciona una salida codificada digitalmente según la tabla adjunta:

PESO (gr)	CODIFICACIÓN PESO	
	P ₁	P ₀
Menor de 0,5	0	0
De 0,5 a 1	0	1
De 1 a 1,2	1	0
Mayor de 1,2	1	1

DIÁMETRO (mm)	CODIFICACIÓN DIÁMETRO	
	d ₁	d ₀
19	0	0
23	0	1
24	1	0
25	1	1

El circuito a diseñar admite como entrada las líneas P₁, P₀, d₁ y d₀, 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 de céntimos de euros.

Las líneas de salida del circuito deben codificar en BCD el valor de la moneda en euros, décimas y centésimas de euros. Por ejemplo, la moneda de 50 céntimos debe codificarse como 0,50.



Teniendo en cuenta que los pesos y diámetros de las monedas de interés son:

TIPO DE MONEDA	PESO (gr)	DIÁMETRO (mm)
5 céntimos	0.6	19
50 céntimos	1.1	24
1 euro	1.1	23
2 euros	1.3	25

Diseñar dicho circuito suponiendo que sólo se van a introducir las cuatro monedas descritas en la tabla anterior y realizando la implementación más simplificada posible usando módulos combinacionales.

NOTA: En el caso de monedas de 1 y 2 euros, los displays de los céntimos deben ser 00.