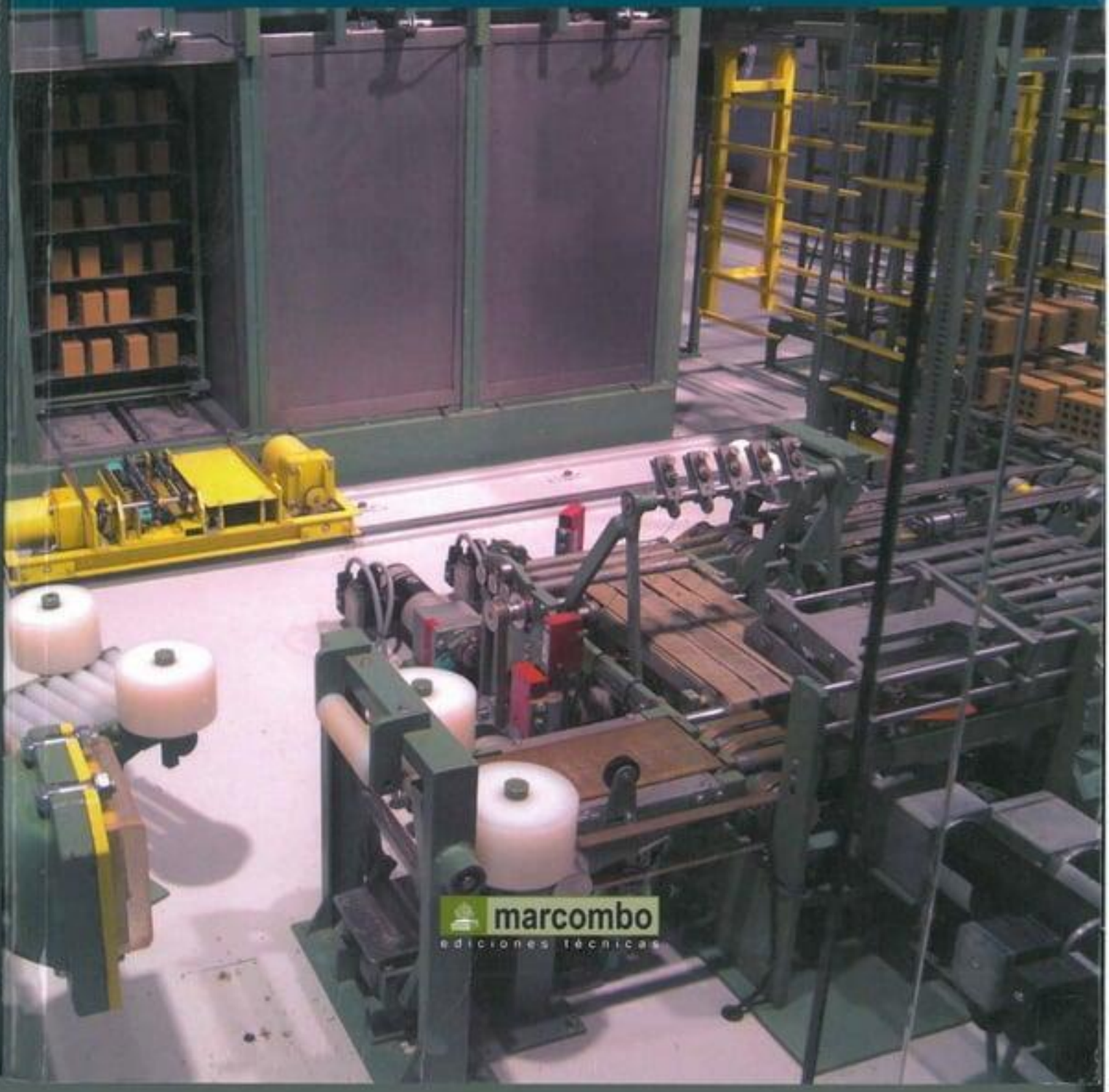
SISTEMAS SECUENCIALES PROGRAMABLES

Sergio Ortiz Sausor
José Manuel Espinosa Malea

marcombo FORMACIÓN



marcombo
ediciones técnicas

Sistemas secuenciales programables

Primera edición, 2014

© 2014 Sergio Ortiz Sausor, José Manuel Espinosa Malea

© 2014 MARCOMBO, S.A.
www.marcombo.com

Maquetación: Pol Creuheras Borda

«Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra solo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley. Diríjase a CEDRO (Centro Español de Derechos Reprográficos, www.cedro.org) si necesita fotocopiar o escanear algún fragmento de esta obra».

ISBN: 978-84-267-2104-4

D.L.: B-27677-2013

Impreso en *España*
Printed in Spain

Índice general

Unidad 1

Sistemas combinacionales con puertas lógicas 1

1.1. Sistemas de numeración. Conversión entre sistemas 2
1.2. Sistemas de codificación 8
1.3. Álgebra de Boole y funciones lógicas 9
1.4. Simplificación de funciones lógicas. Mapas de Karnaugh 19
1.5. Circuitos combinacionales con puertas lógicas 25

Unidad 2

Bloques combinacionales 37

2.1. Codificadores 38
2.2. Multiplexores 46
2.3. Demultiplexores 54
2.4. Decodificadores 55

Unidad 3

Sistemas secuenciales con puertas lógicas 67

3.1. Definición y tipos 68
3.2. Básculas o biestables 68
3.3. Registros de desplazamiento 75
3.4. Contadores 78

Unidad 4

Detectores y preactuadores 87

4.1. Introducción 88
4.2. Entradas digitales a los autómatas programables 88
4.3. Entradas analógicas a los autómatas programables 95
4.4. Salidas digitales de los autómatas programables 106
4.5. Salidas analógicas de los autómatas programables 112

Anexo 1 - variador de velocidad (TECO T-VERTER E2-201-H1F) 114

Unidad 5

Descripción y programación del autómata LOGO 121

5.1. Descripción del módulo programable LOGO! 122
5.2. Conexiones del módulo programable LOGO! 126
5.3. Entrenador para el autómata programable LOGO! 130
5.4. Entorno de programación para el LOGO! 132
5.5. Comunicación LOGO!-PC 134
5.6. Programación de sistemas secuenciales con LOGO! 136

Unidad 6

Descripción y programación del autómata S7-1200 191

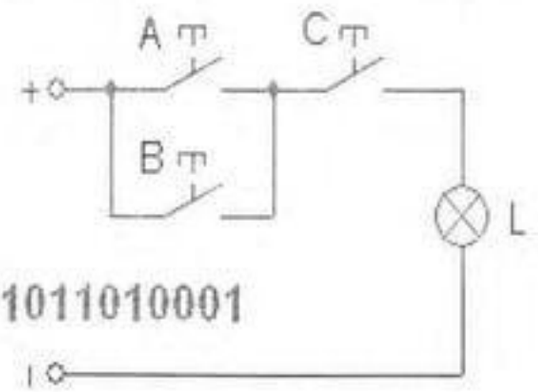
6.1. Descripción del autómata programable S7-1200 192
6.2. Introducción al entorno TIA Portal 196
6.3. Marcas de sistema y de ciclo. Cambio en el direccionamiento E/S 200
6.4. Programación por segmentos en el autómata S7-1200 202
6.5. Entrenador para el autómata S7-1200 204
6.6. Detección automática de la CPU en autómatas S7-1200 206
6.7. Comunicación Ethernet entre un autómata S7-1200 y un ordenador 208
6.8. Programación de sistemas secuenciales con S7-1200 213

Unidad 1 Sistemas combinacionales con puertas lógicas

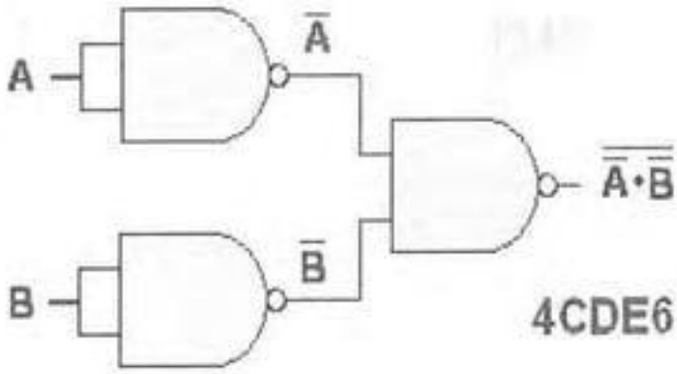
01101110

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

43EDF7



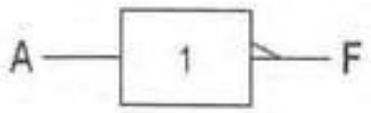
1011010001



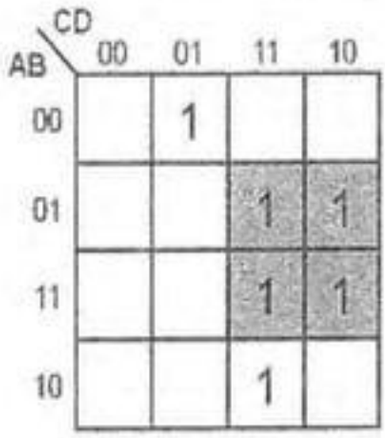
4CDE6

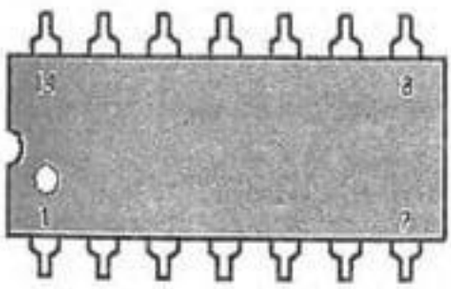
٠١٢٣٤٥٦٧٨٩

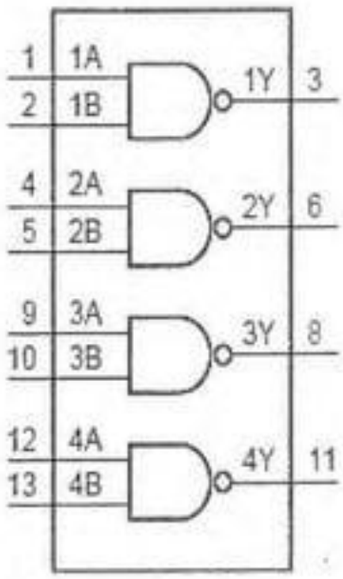
0123456789



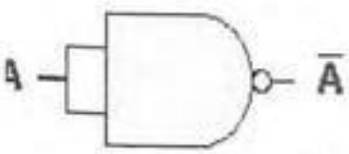
95ABC



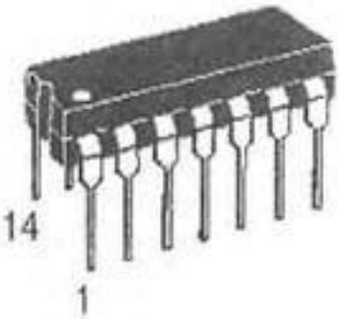




568



111100



En este capítulo:

1.1. Sistemas de numeración. Conversión entre sistemas

1.2. Sistemas de codificación

1.3. Álgebra de Boole y funciones lógicas

1.4. Simplificación de funciones lógicas. Mapas de Karnaugh

1.5. Circuitos combinacionales con puertas lógicas

1.1. Sistemas de numeración
Conversión entre sistemas

Un **sistema de numeración** se puede definir como un conjunto de símbolos permitidos y las reglas que nos permiten generarlos, las cuales además nos van a indicar qué números son válidos y cuáles no dentro del sistema.

Los sistemas de numeración existen desde muy antiguo, pues cada civilización humana a lo largo de la historia (romanos, árabes, griegos, fenicios, mayas, japoneses, etc.) ha ido desarrollando un sistema de numeración acorde a sus necesidades.

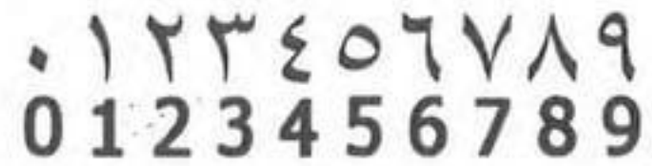


Fig. 1.1. Símbolos de numeración árabes

En la actualidad, con el desarrollo de la tecnología, se han debido crear nuevos sistemas de numeración técnicos pensados para las necesidades de funcionamiento de los equipos tecnológicos. En este sentido, se deben citar el sistema decimal, el sistema binario, el sistema octal, el sistema hexadecimal, entre otros.

El **sistema de numeración decimal**, en base 10, es el más extendido y conocido por la mayoría de la población mundial. Como símbolos permitidos en el sistema se tienen el 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9. Las reglas de utilización son sencillas, de forma que cuando contamos incrementando los símbolos y los agotamos (9), basta añadir una nueva columna a la izquierda (1x) y los reutilizamos empezando otra vez por el cero (10). Pasamos a unidades de segundo orden (decenas) y después, siguiendo esta regla, a las centenas, unidades de millar, decenas de millar, etc.

Al utilizar potencias en base 10, el exponente de los dígitos situados a la izquierda de la coma es positivo (parte entera) mientras el exponente de los dígitos situados a la izquierda de la coma es negativo (parte fraccionaria). De esta forma, se pueden construir números como el 1.564,85 que, expresado en potencias de base 10, sería:

$$1.564,85_{10} = 1 \cdot 10^3 + 5 \cdot 10^2 + 6 \cdot 10^1 + 4 \cdot 10^0 , + 8 \cdot 10^{-1} + 5 \cdot 10^{-2}$$

El **sistema de numeración binario actual**, en base 2, se debe a los trabajos de Leibniz y, sobre todo, a los del matemático George Boole que desarrolló las reglas modernas o un sistema de lógica conocido como el Álgebra de Boole. Este sistema ha sido el fundamento para el desarrollo de los circuitos electrónicos basados en relés y conmutadores, los cuales a su vez han sido la base de múltiples equipos tecnológicos utilizados en la actualidad, como los autómatas programables o plcs.

Recuerda . . .

Los sistemas de numeración son:

- Decimal (del 0 al 9)
- Binario (0 y 1)
- BCD
- Hexadecimal
- Octal

Como símbolos permitidos en el sistema binario solo se tienen dos dígitos o bits (*binary digit*) que son el «0» (cerrado) y el «1» (abierto). Mediante el bit se define la unidad mínima de información empleada. El bit permite representar, pues, solo 2 valores cualesquiera como cerrado o abierto, falso o verdadero, apagado o encendido, etc.

Para poder representar más estados en un dispositivo digital, es necesario utilizar un número mayor de bits. Por ejemplo, con 2 bits y considerando base 2, se pueden representar hasta 4 estados diferentes ($2^n = 2^2 = 4$, con $n = 2$) para interruptores:

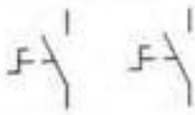
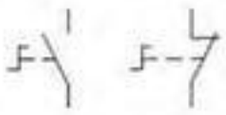
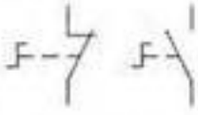
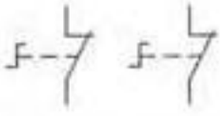
			
0 0 - Los dos están «abiertos»	**0 1** - El primero por la derecha está «cerrado» y el segundo «abierto»	**1 0** - El primero por la derecha está «abierto» y el segundo «cerrado»	**1 1** - Los dos están «cerrados»

Fig. 1.2.
Numeración binaria asociada a diferentes estados de 2 interruptores

Si se trabaja con una secuencia más grande (por ejemplo, 8 bits ordenados que equivalen a 1 byte), se pueden representar hasta $2^8 = 256$ valores diferentes. En general, con un número **n** de bits pueden representarse hasta $\mathbf{2^n}$ combinaciones diferentes.

En un *byte*, se debe considerar la posición que ocupa cada bit y su valor (de derecha a izquierda). El bit más significativo (MSB) es aquel que tiene un mayor peso o valor en el conjunto (situado más a la izquierda) y el menos significativo (LSB) el que menos peso tiene (situado más a la derecha). Como regla, cada vez que un bit se desplaza una posición hacia la izquierda vale el doble y cada vez que se mueve hacia la derecha vale la mitad.

7	6	5	4	3	2	1	0	**Posición del bit**
128	64	32	16	8	4	2	1	**Valor según posición**

Fig. 1.3.
Estructura de un *byte* (8 bits)

Además del *byte*, en el sistema binario se trabaja con palabras (WORD) y con dobles palabras (DWORD). La palabra está formada por 2 *bytes* (16 bits) y la doble palabra por 4 *bytes* (32 bits).

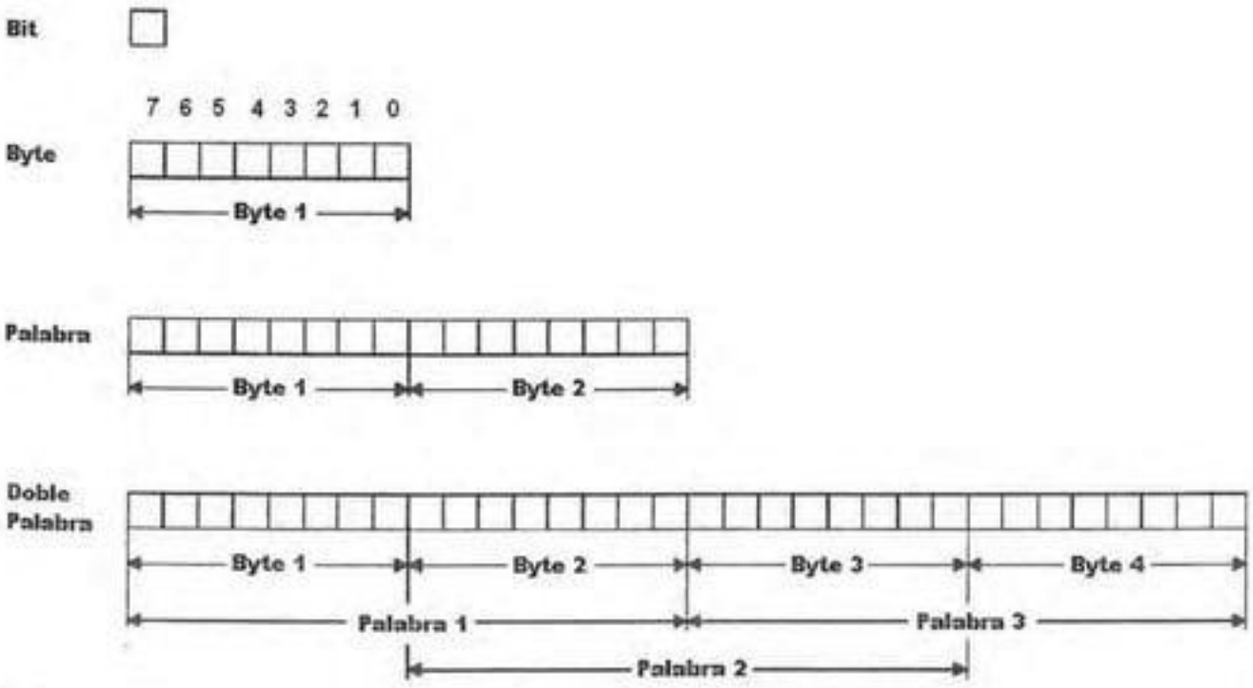


Fig. 1.4.
Estructura de un *byte*, una palabra y una doble palabra

Conversión de decimal a binario

Para realizar la conversión de decimal a binario, se debe dividir sucesivamente el número decimal entre 2, **hasta realizar la última división cuando el número a dividir sea 1. Los restos de las divisiones se ordenan desde el último al primero (en orden inverso).**

Ejemplo 1.1

Transforma el número decimal 87_{10} en binario.

87 ÷ 2 = 43 y el resto es **1**; 43 ÷ 2 = 21 y el resto es **1**; 21 ÷ 2 = 10 y el resto es **1**;

10 ÷ 2 = 5 y el resto es **0**; 5 ÷ 2 = 2 y el resto es **1**; 2 ÷ 2 = 1 y el resto es **0**;

1 ÷ 2 = 0 y el resto es **1**;

Ordenando los restos en orden inverso, el número binario buscado es: 1010111_2

Conversión de binario a decimal

Para realizar la conversión de binario a decimal, empezando por la derecha, se debe desarrollar una suma de potencias en base 2 donde cada cifra (0, 1) multiplica a su potencia respectiva.

Ejemplo 1.2

Transforma todos los números binarios de 4 bits en su valor decimal.

Número binario	Desarrollo	Valor decimal
0000_2	$0\cdot2^3 + 0\cdot2^2 + 0\cdot2^1 + 0\cdot2^0$	0_{10}
0001_2	$0\cdot2^3 + 0\cdot2^2 + 0\cdot2^1 + 1\cdot2^0$	1_{10}
0010_2	$0\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 0\cdot2^0$	2_{10}
0011_2	$0\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0$	3_{10}
0100_2	$0\cdot2^3 + 1\cdot2^2 + 0\cdot2^1 + 0\cdot2^0$	4_{10}
0101_2	$0\cdot2^3 + 1\cdot2^2 + 0\cdot2^1 + 1\cdot2^0$	5_{10}
0110_2	$0\cdot2^3 + 1\cdot2^2 + 1\cdot2^1 + 0\cdot2^0$	6_{10}
0111_2	$0\cdot2^3 + 1\cdot2^2 + 1\cdot2^1 + 1\cdot2^0$	7_{10}
1000_2	$1\cdot2^3 + 0\cdot2^2 + 0\cdot2^1 + 0\cdot2^0$	8_{10}
1001_2	$1\cdot2^3 + 0\cdot2^2 + 0\cdot2^1 + 1\cdot2^0$	9_{10}
1010_2	$1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 0\cdot2^0$	10_{10}
1011_2	$1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0$	11_{10}
1100_2	$1\cdot2^3 + 1\cdot2^2 + 0\cdot2^1 + 0\cdot2^0$	12_{10}
1101_2	$1\cdot2^3 + 1\cdot2^2 + 0\cdot2^1 + 1\cdot2^0$	13_{10}
1110_2	$1\cdot2^3 + 1\cdot2^2 + 1\cdot2^1 + 0\cdot2^0$	14_{10}
1111_2	$1\cdot2^3 + 1\cdot2^2 + 1\cdot2^1 + 1\cdot2^0$	15_{10}

Fig. 1.5. Valores decimales posibles con 4 bits

Un **número codificado en BCD** (*Binary-Coded Decimal* o decimal codificado en binario) es la construcción de un número binario (secuencia de 4 bits), de tal manera que se pueda leer en él directamente un valor decimal. Cuando el número es de más de una cifra, hacen falta tantos **números** binarios como cifras para construir el número BCD. El número BCD no tiene por qué coincidir con el binario puro.

Ejemplo 1.3

Transforma el número 429_{10} en BCD y en binario

4	2	9	DECIMAL
0100	0010	1001	BCD
	110101101		BINARIO

El **sistema de numeración hexadecimal (Hex)**, en base 16, permite representar los números binarios debido a que un *byte* corresponde exactamente a dos dígitos hexadecimales ($2^8 = 2^4 \cdot 2^4 = 16 \cdot 16$). Se debe utilizar un símbolo (número o letra) por cada cuatro bits del sistema binario.

Como símbolos permitidos en el sistema se tienen el 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F (16 símbolos alfanuméricos). Los diez primeros dígitos hexadecimales se corresponden con los decimales y, a continuación, se añaden las seis primeras letras del alfabeto latino hasta completar el total de 16 símbolos.

Decimal	Binario	Hexadecimal	Octal
0_{10}	0000_2	0_{16}	0_8
1_{10}	0001_2	1_{16}	1_8
2_{10}	0010_2	2_{16}	2_8
3_{10}	0011_2	3_{16}	3_8
4_{10}	0100_2	4_{16}	4_8
5_{10}	0101_2	5_{16}	5_8
6_{10}	0110_2	6_{16}	6_8
7_{10}	0111_2	7_{16}	7_8
8_{10}	1000_2	8_{16}	10_8
9_{10}	1001_2	9_{16}	11_8
10_{10}	1010_2	A_{16}	12_8
11_{10}	1011_2	B_{16}	13_8
12_{10}	1100_2	C_{16}	14_8
13_{10}	1101_2	D_{16}	15_8
14_{10}	1110_2	E_{16}	16_8
15_{10}	1111_2	F_{16}	17_8

Fig. 1.6. Correspondencia entre sistemas de numeración

Conversión de binario a hexadecimal

Para realizar la conversión de binario a hexadecimal, se debe agrupar de 4 en 4 el número empezando por el LSB. Si faltan dígitos, se debe añadir ceros a la izquierda. Basta aplicar la tabla de correspondencia y formar el número hexadecimal de izquierda a derecha.

Ejemplo 1.4

Transforma el número 1010001001110001_2 en hexadecimal

1010	0010	0111	0001	BINARIO
A	2	7	1	HEXADECIMAL

El número hexadecimal equivalente es el $A271_{16}$.

Conversión de hexadecimal a binario

Para realizar la conversión de hexadecimal a binario, se debe reemplazar el número hexadecimal por el equivalente de 4 bits.

Ejemplo 1.5

Transforma el número $4CD7_{16}$ en binario y en decimal

4	C	D	7	HEXADECIMAL
0100	1100	1101	0111	BINARIO

El número binario equivalente es el 100110011010111_2.

El número decimal equivalente es 19.671_{10}, pues haciendo el desarrollo se tiene que:

$1 \cdot 2^{14} + 1 \cdot 2^{11} + 1 \cdot 2^{10} + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$

$= 16.384 + 2.048 + 1.024 + 128 + 64 + 16 + 4 + 2 + 1 = 19.671$

El **sistema de numeración octal**, en base 8, utiliza como símbolos permitidos en el sistema el 0, 1, 2, 3, 4, 5, 6 y 7. Cuando los agotamos (7), basta añadir una nueva columna a la izquierda (1x) y formar el número octal de izquierda a derecha.

Conversión de binario a octal

Para realizar la conversión de binario a octal, se debe agrupar de 3 en 3 el número empezando por el LSB. Si faltan dígitos, se deben añadir ceros a la izquierda. Basta aplicar la tabla de correspondencia y formar el número octal de derecha a izquierda.

Ejemplo 1.6

Transforma el número 1111110001_2 en octal

001	111	110	001	BINARIO
1	7	6	1	OCTAL

El número octal equivalente es el 1.761_8.

Conversión de octal a binario

Para realizar la conversión de octal a binario, cada dígito octal se convierte en su binario equivalente de 3 bits y se juntan en el mismo orden.

Ejemplo 1.7

Transforma el número 427_8 en binario y en hexadecimal

4	2	7	OCTAL
100	010	111	BINARIO
	100010111		BINARIO
1	1	7	HEXADECIMAL

El número binario equivalente es el 100010111_2 y el hexadecimal es el 117_{16}.

El **código Gray** o código binario reflejado es un sistema de numeración binario que se construye de forma que dos valores sucesivos solo se diferencian en uno de sus dígitos. Este código se emplea para poder simplificar funciones lógicas mediante mapas de Karnaugh. Estos mapas son una herramienta adecuada y válida para poder diseñar circuitos secuenciales y combinacionales.

1 bit	2 bits	3 bits	4 bits
0	00	000	0000
1	01	001	0001
	11	011	0011
	10	010	0010
		110	0110
		111	0111
		101	0101
		100	0100
			1100
			1101
			1111
			1110
			1010
			1011
			1001
			1000

Fig. 1.7. Códigos de Gray hasta 4 bits

Recuerda . . .

La conversión de sistemas de numeración se utiliza para facilitar la lectura de cifras y para que las máquinas binarias puedan tratar entradas y salidas analógicas.

Como resumen, en la figura 1.8, se consideran los diferentes sistemas de numeración definidos y las conversiones analizadas en los ejemplos.

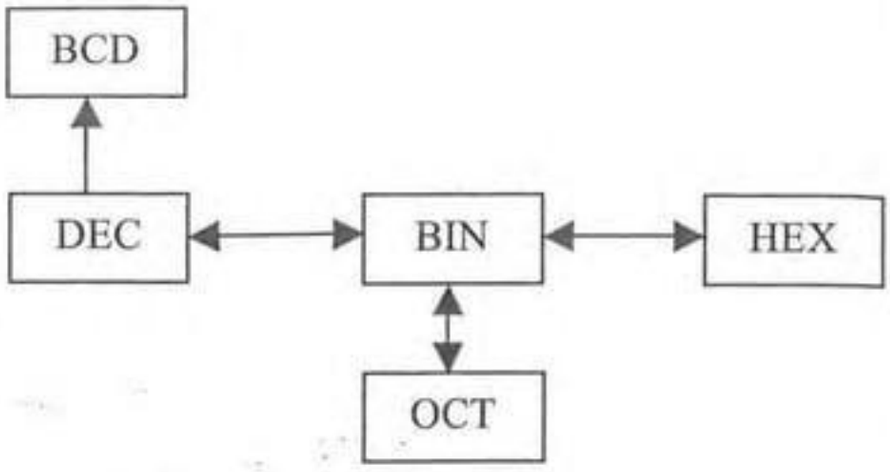


Fig. 1.8. Resumen de conversiones entre sistemas de numeración

1.2. Sistemas de codificación

Un **sistema de codificación** es aquel sistema que permite realizar el proceso de conversión de un sistema de datos inicial a un sistema de datos final (por ejemplo, convertir una señal de tensión analógica a una señal digital codificada en binario o hexadecimal). Una señal analógica es un tipo de señal que presenta valores de forma continua en el tiempo. Esta variación puede venir dada por una función matemática, como en el caso de una onda senoidal, o por una variación aleatoria.

Esta conversión es bastante habitual, pues hay muchos transductores (de temperatura, de humedad, de aceleración, etc.) cuya salida es una señal analógica normalizada de tensión o de intensidad. Las ventajas de esta conversión radican en la inmunidad al ruido de la señal digital, así como en su mayor facilidad para detectar y corregir errores.

La conversión analógico-digital indicada requiere de una etapa inicial de muestreo, de una etapa de discretización y de una etapa final de codificación.

Ejemplo 1.8

Realiza la conversión A-D de una señal analógica de tensión.

En la figura 1.9 se representa una señal analógica de tensión, que presenta diferentes valores a lo largo del tiempo. La conversión A-D requiere definir un tiempo para realizar el muestreo de la señal (en este caso, 1 segundo). En la figura siguiente se realiza la discretización o digitalización de los valores muestreados.

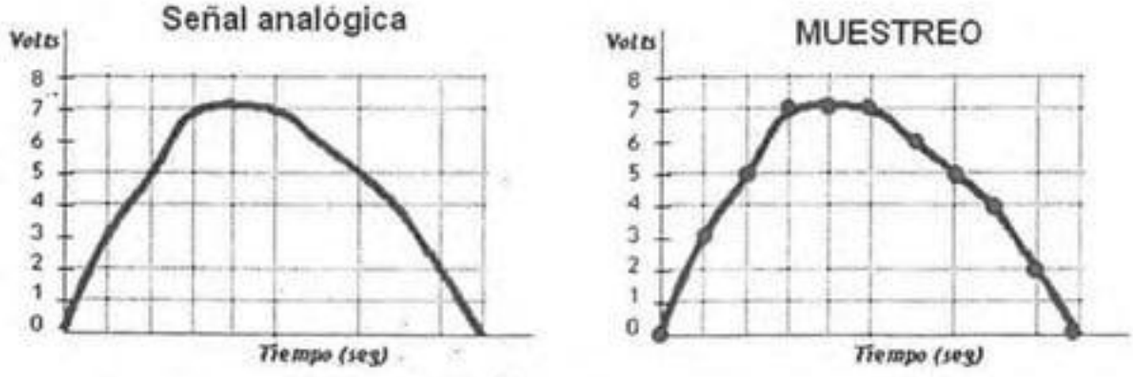


Fig. 1.9. Proceso de conversión analógico-digital

Fig. 1.10. Proceso de conversión analógico-digital

Por último, el proceso de codificación requiere convertir la señal discreta de valores decimales (0, 3, 5, 7, 7, 7, 6, 5, 4, 2, 0) en valores en código binario (BIN) o hexadecimal (HEX). La codificación en binario (3 bits) de la señal digitalizada es 000, 011, 101, 111, 111, 111, 110, 101, 100, 010 y 000.

También se puede realizar el proceso contrario, llamado conversión digital-analógica. Es este caso, la conversión D-A parte de un conjunto de valores codificados en binario o hexadecimal, los convierte a valores decimales discretos y reconstruye la señal analógica correspondiente a partir de dichos valores muestreados.

Actividades propuestas

1. Convierte en binario y en BCD los siguientes números decimales:

$$52_{10};\ 140_{10};\ 592_{10};\ 12_{10}$$

2. Convierte en hexadecimal y en octal los siguientes números binarios:

$$10011010110_2;\ 1010110001_2;\ 100011111_2;\ 1011111110_2$$

3. Convierte en binario y en decimal los siguientes números hexadecimales:

$$AF12_{16};\ 945CD_{16};\ 2213BB_{16};\ 123ABC_{16}$$

4. Convierte en digital y en hexadecimal la señal analógica de tensión dada en la tabla. Esta señal ha sido muestreada de 0,5 en 0,5 seg.

Tiempo (seg)	Tensión (V)	Tiempo (seg)	Tensión (V)
0	0	2,5	22
0,5	54	3	19
1	62	3,5	25
1,5	14	4	36
2	2	4,5	37

1.3. Álgebra de Boole y funciones lógicas

El álgebra de Boole es una metodología lógica para tratar variables binarias. El álgebra está formada por variables booleanas o lógicas, cuyos estados binarios (1 y 0) son estados lógicos, así como únicamente tres tipos de operaciones lógicas (la suma lógica OR, el producto lógico AND y la negación NOT).

La negación NOT es la operación que provoca el cambio de estado de una variable lógica A. Se representa por $\overline{A}$. Como ejemplo se puede considerar, inicialmente, un circuito formado por un pulsador P, con contacto auxiliar normalmente abierto,

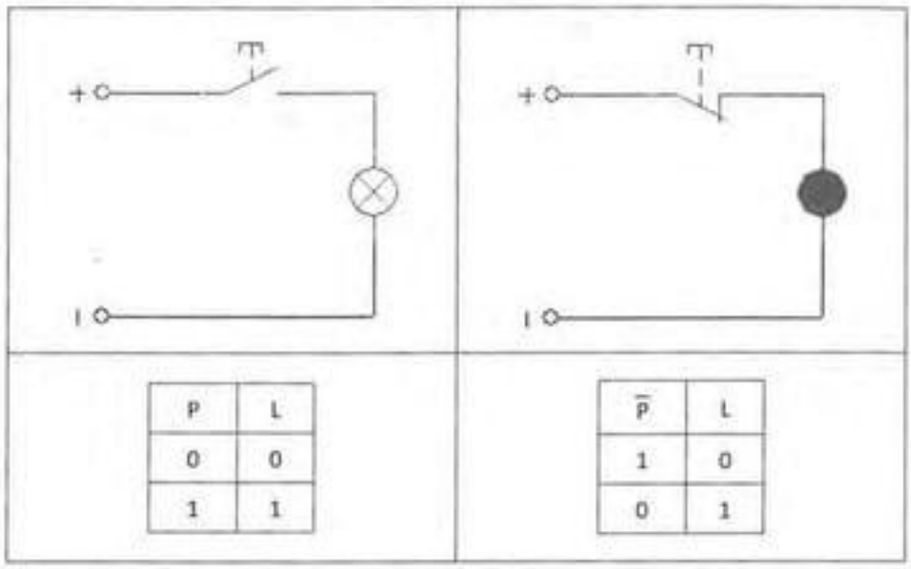
Recuerda • • •

El álgebra de Boole trata las variables binarias. La propiedad del sistema binario nos dice que el número de combinaciones es igual a 2^n, siendo n el número de variables.

y una luz de señalización L. Para aplicar la negación sobre la variable P, basta sustituir el contacto auxiliar inicial por otro que sea normalmente cerrado.

A	$\overline{A}$
0	1
1	0

Fig. 1.11. Tabla de la verdad (negación)

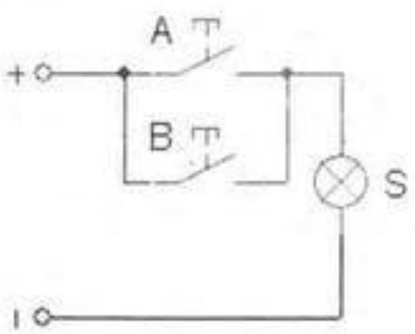


P	L
0	0
1	1

$\overline{P}$	L
1	0
0	1

Fig. 1.12. Circuito eléctrico y tablas de la verdad (negación)

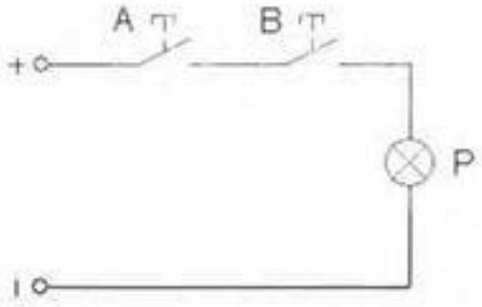
La suma OR es la operación lógica de adición sobre dos o más variables booleanas. Se representa como A + B, para dos variables. El resultado de la operación es 1, siempre que cualquiera de las variables tome el estado 1. Como ejemplo se puede considerar un circuito eléctrico formado por dos pulsadores (A, B) en paralelo, con contactos auxiliares NO, y una luz de señalización S. Se observa que A + 0 = A y que A + 1 = 1.



A	B	S = A + B
0	0	0
0	1	1
1	0	1
1	1	1

Fig. 1.13. Circuito eléctrico y tabla de la verdad (suma)

El producto AND es la operación lógica de multiplicación sobre dos o más variables booleanas. Se representa como A · B, para dos variables. El resultado de la operación es 0, siempre que cualquiera de las variables tome el estado 0. Como ejemplo se puede considerar un circuito eléctrico formado por dos pulsadores (A, B) en serie, con contactos auxiliares NO, y una luz de señalización P. Se observa que A · 1 = A y que A · 0 = 0.



A	B	P = A · B
0	0	0
0	1	0
1	0	0
1	1	1

Fig. 1.14.
Circuito eléctrico y tabla de la verdad (producto)

En general, la relación que existe entre las variables booleanas consideradas se puede expresar mediante la **función lógica F (u otra letra que represente dicha función). La función lógica se puede representar de diferentes formas, siendo las más habituales la tabla de la verdad, las expresiones algebraicas y la forma gráfica mediante símbolos.**

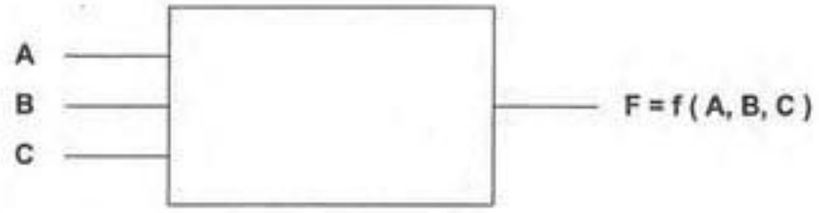


Fig. 1.15.
Función lógica F = f (A, B, C)

Además, para trabajar con las funciones lógicas derivadas de las operaciones del álgebra de Boole, es necesario conocer una serie de reglas, leyes y teoremas adicionales.

REGLAS, LEYES Y TEOREMAS	EQUIVALENCIAS	OBSERVACIONES
Derivada de la negación	$\overline{\overline{A}} = A;$	
Derivadas de la suma (paralelo)	$A + 0 = A;\ A + A = A;$	0 es el elemento neutro
	$A + 1 = 1;\ A + \overline{A} = 1;$	
Derivadas del producto (serie)	$A \cdot 0 = 0;\ A \cdot A = A;$	1 es el elemento neutro
	$A \cdot 1 = A;\ A \cdot \overline{A} = 0;$	
Ley conmutativa	$A + B = B + A;$	Para la suma
	$A \cdot B = B \cdot A;$	Para el producto
Ley asociativa	$A + (B + C) = (A + B) + C = A + B + C;$	Para la suma
	$A \cdot (B \cdot C) = (A \cdot B) \cdot C = A \cdot B \cdot C;$	Para el producto
Ley distributiva	$A \cdot (B + C) = A \cdot B + A \cdot C;$	
Ley de absorción	$A + A \cdot B = A;$	
	$A \cdot (A + B) = A;$	
Teorema de De Morgan	$\overline{A \cdot B} = \overline{A} + \overline{B};$	Conversión de producto en suma
	$\overline{A + B} = \overline{A} \cdot \overline{B};$	Conversión de suma en producto

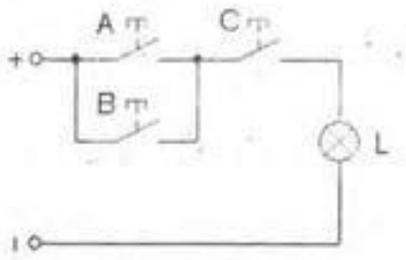
Fig. 1.16.
Reglas, leyes y teoremas

1.3.1. Tabla de la verdad

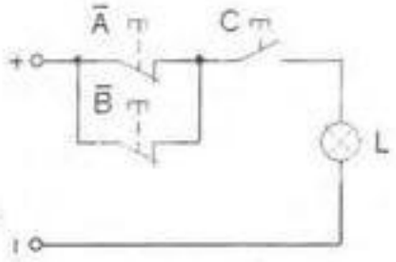
La tabla de la verdad es un método útil cuando el número de variables booleanas a considerar es reducido, pero deja de serlo si hay muchas variables (2^n posibilidades, siendo **n** el número de variables). La tabla es única y contiene todos los valores posibles de la función lógica, dependiendo del valor de las variables booleanas.

Ejemplo 1.9

Elabora la tabla de la verdad del circuito considerado, formado por 3 pulsadores (variables booleanas A, B y C) y una luz de señalización L.



Circuito 1.9.a.



Circuito 1.9.b.

Tabla de la verdad:

A	B	C	L
0	0	0	0
0	0	1	0
0	1	1	1
0	1	0	0
1	1	0	0
1	1	1	1
1	0	1	1
1	0	0	0

Circuito 1.9.a.

$\overline{A}$	$\overline{B}$	C	L
0	0	0	0
0	0	1	1
0	1	1	1
0	1	0	0
1	1	0	0
1	1	1	0
1	0	1	1
1	0	0	0

Circuito 1.9.b.

1.3.2. Expresión algebraica

Una **expresión algebraica** es la representación de una función lógica a través de sumas y productos lógicos de las variables de entrada de la función. La manera más sencilla de realizar la equivalencia es mediante la forma canónica, en la que se incluyen todas las variables de la función, usando los conceptos de **minitérmino (m)** y **maxitérmino (M):**

- El minitérmino (m) es un **producto lógico** donde cada variable aparece una sola vez (negada o sin negar). Ejemplos: $A \cdot \overline{B} \cdot C$; $A \cdot B \cdot \overline{C}$; $\overline{A} \cdot B \cdot \overline{C}$;
- El maxitérmino (M) es una **suma lógica** donde cada variable aparece una sola vez (negada o sin negar). Ejemplos: $(A + B + \overline{C})$;$(\overline{A} + B + C)$; $(A + \overline{B} + \overline{C})$;

Ejemplo 1.10

Escribe las expresiones algebraicas equivalentes, en forma canónica, de la función lógica L del ejemplo 1.9.

Mediante la **forma canónica disyuntiva,** la expresión algebraica de la función lógica L se puede expresar como suma de todos los minitérminos (o sea, como suma de todos los productos lógicos) que valen 1.

$$L = f(A,B,C) = \sum_i m_i(=1);$$

En este caso, la regla para obtener la expresión algebraica de la función lógica L es:

Un cero se convierte en una variable negada.

Un uno se convierte en una variable sin negar.

Para el ejemplo 1.9, la función lógica L es:

Circuito 1.9.a: $L = \bar{A} \cdot B \cdot C + A \cdot B \cdot C + A \cdot \bar{B} \cdot C;$

Circuito 1.9.b: $L = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot \bar{B} \cdot C;$

Mediante la **forma canónica conjuntiva,** la expresión algebraica de la función lógica L se puede expresar como producto de todos los maxitérminos (o sea, como producto de todas las sumas lógicas) que valen 0.

$$L = f(A,B,C) = \prod_i M_i\,(=0)$$

En este caso, la regla para obtener la expresión algebraica de la función lógica L es:

Un cero se convierte en una variable sin negar.

Un uno se convierte en una variable negada.

Para el ejemplo 1.9, la función lógica L es:

Circuito 1.9.a:

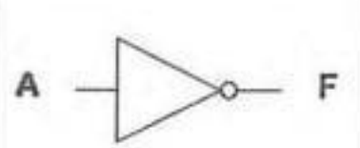
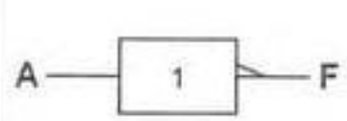
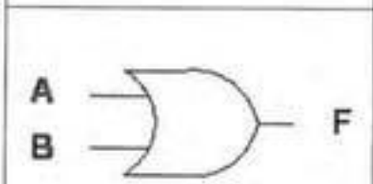
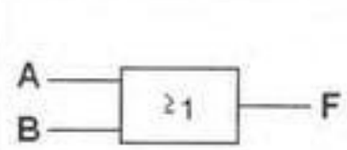
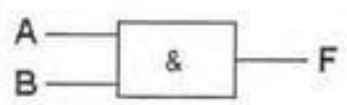
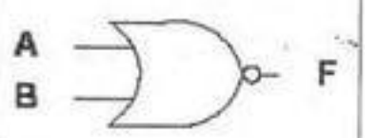
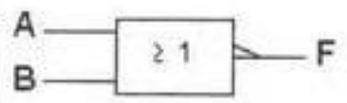
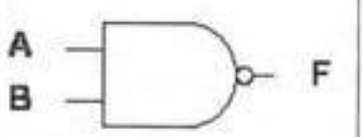
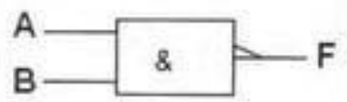
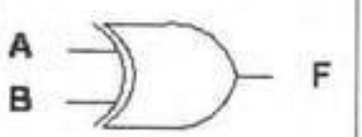
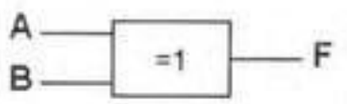
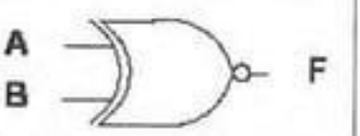
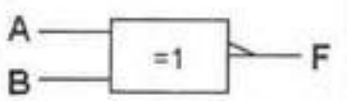
$$L = (A+B+C)\cdot(A+B+\bar{C})\cdot(A+\bar{B}+C)\cdot(\bar{A}+\bar{B}+C)\cdot(\bar{A}+B+C)$$

Circuito 1.9.b:

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

1.3.3. Puertas lógicas básicas

Una función lógica F también se puede representar de **forma gráfica** mediante circuitos eléctricos, neumáticos o electrónicos. Para circuitos electrónicos, se utilizan **puertas lógicas** que son la expresión física de un operador lógico. Las puertas lógicas están formadas por circuitos de conmutación integrados en un chip. Las puertas lógicas utilizadas (con simbología clásica y normativa ANSI/IEEE) son la NOT, la OR, la AND, la NOR, la NAND, la XOR y la NXOR

Símbolo clásico	Símbolo ANSI/IEEE	Función lógica	Tabla de la verdad	Nombre
		NOT $F = \overline{A}$	A | F 0 | 1 1 | 0	Negación
		OR (O) F = A + B	A B | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1	Suma
		AND (Y) F = A · B	A B | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1	Producto
		NOR $F = \overline{A + B}$	A B | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0	Suma negada
		NAND $F = \overline{A \cdot B}$	A B | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0	Producto negado
		XOR $F = \overline{A} \cdot B + A \cdot \overline{B}$	A B | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0	Semisuma
		XNOR $F = \overline{\overline{A} \cdot B + A \cdot \overline{B}}$	A B | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1	Semisuma negada

Nota: En una puerta XOR, la salida vale 1 cuando A = 1 o cuando B = 1 pero no cuando ambas valen 1, es decir, una entrada a 1 excluye a la otra.

Fig. 1.17.
Puertas lógicas básicas utilizadas en sistemas digitales

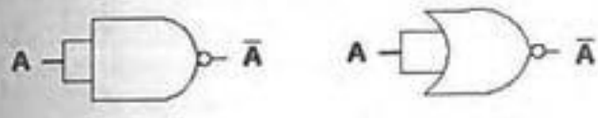
Como resumen de las puertas lógicas consideradas, es interesante recalcar que:

	Valor de salida cero si:	Valor de salida uno si:
Puerta AND	Alguna entrada es cero.	Todas las entradas son uno.
Puerta OR	Todas las entradas son cero.	Alguna entrada es uno.
Puerta NAND	Todas las entradas son uno.	Alguna entrada es cero.
Puerta NOR	Alguna entrada es uno.	Todas las entradas son cero.
Puerta XOR	Hay un número par de entradas igual a uno	Hay un número impar de entradas igual a uno
Puerta XNOR	Hay un número impar de entradas igual a uno	Hay un número par de entradas igual a uno

Ejemplo 1.11

Construye las puertas lógicas elementales NOT, OR y AND utilizando las puertas lógicas NAND y NOR.

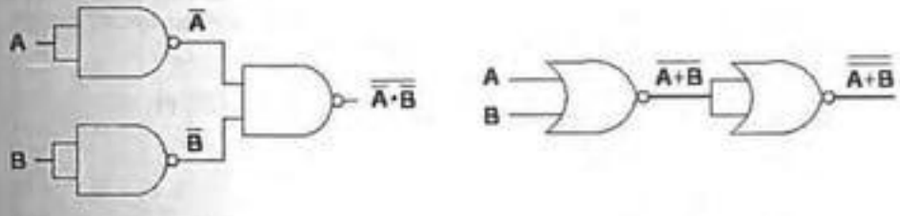
1. Puerta lógica NOT (NEGACIÓN): Se considera A = B



A	A	F
0	0	1
1	1	0

Con una puerta NAND Con una puerta NOR

2. Puerta lógica OR (SUMA):



Con 3 puertas NAND:

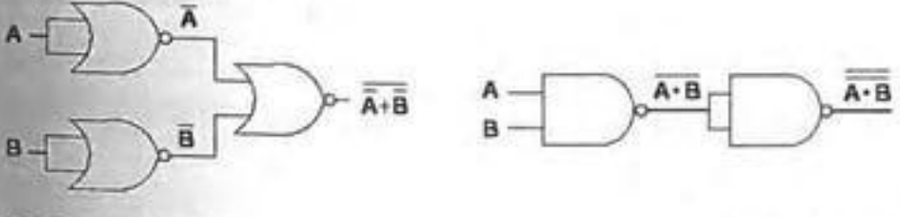
T. de De Morgan $\overline{A+B} = \overline{A} \cdot \overline{B}$;

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

Con 2 puertas NOR:

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

3. Puerta lógica AND (PRODUCTO):



Con 3 puertas NOR:

T. de De Morgan $\overline{A \cdot B} = \overline{A} + \overline{B}$;

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

Con 2 puertas NAND:

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

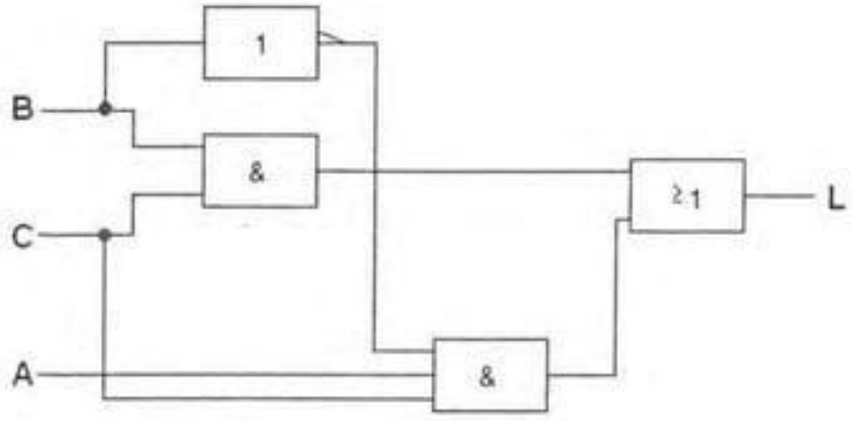
Ejemplo 1.12

Representa de forma gráfica mediante puertas lógicas básicas (símbolos ANSI/IEEE) la función lógica L del ejemplo 1.9. Realiza la simplificación de la forma canónica disyuntiva.

Para el circuito 1.9.a, la simplificación de la expresión algebraica de función lógica L es:

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

La representación de forma gráfica mediante puertas lógicas básicas (NOT, AND, OR) es:

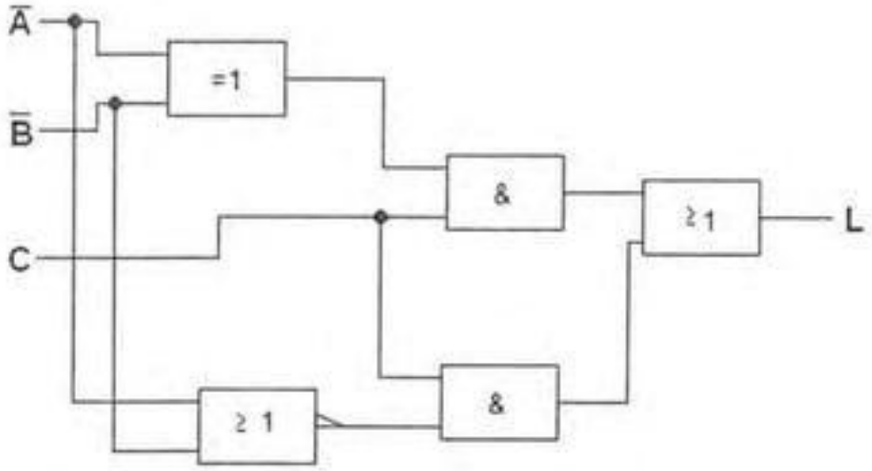


Para el circuito 1.9.b, la simplificación de la expresión algebraica de función lógica L es:

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

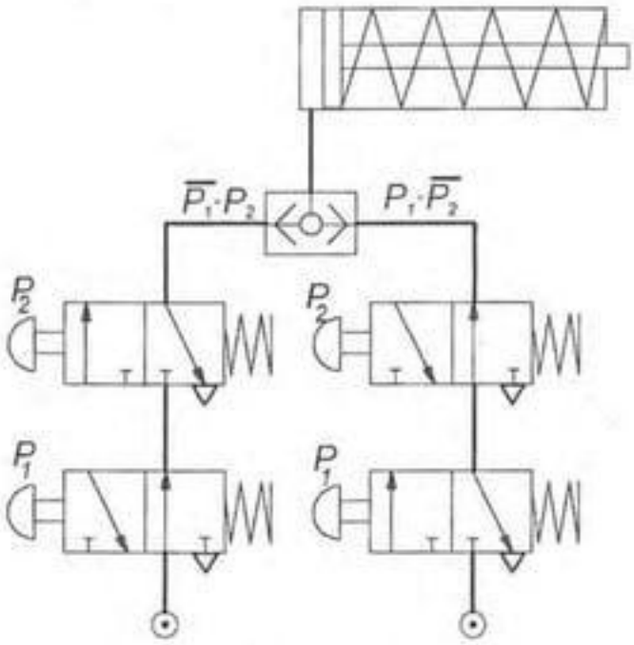
En la primera simplificación se debe aplicar el teorema de De Morgan, mientras en la segunda simplificación se considera una semisuma de las variables $\overline{A}$ y $\overline{B}$. En ambos casos el término del paréntesis aparece multiplicado por la variable C.

La representación de forma gráfica mediante puertas lógicas básicas (NOR, XOR, AND, OR) es:



Ejemplo 1.13

Dado un circuito neumático formado por 2 pulsadores con válvulas 3/2 NC y NO que actúan sobre un cilindro de simple efecto con retorno por muelle, se pide obtener la función lógica F de las tres formas posibles (tabla de la verdad, expresión algebraica y en forma gráfica mediante puertas lógicas).



Circuito 1.13.

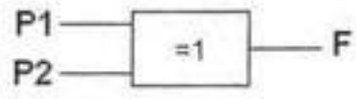
La tabla de la verdad de la función lógica F es:

P1	P2	F
0	0	0
0	1	1
1	0	1
1	1	0

La expresión algebraica de la función lógica F es una semisuma pues:

$$F = \overline{P1}\cdot P2 + P1\cdot\overline{P2};$$

Por tanto, la implementación mediante puertas lógicas se reduce al uso de una puerta XOR.



Actividades propuestas

1. Dibuja una tabla de números binarios del 0 al 30.

2. Dibuja una tabla de números hexadecimales del 0 al 30.

3. Realiza la conversión sin calculadora de los diferentes números indicados en las tablas siguientes.

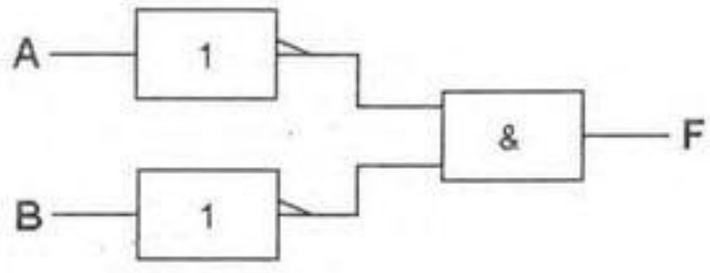
BINARIO	DECIMAL	HEXADECIMAL	DECIMAL	BCD	DECIMAL
00111011		FF		01100101	
11001110		E0		01110001	
01011011		21		10001001	
11111111		B7		00100111	

DECIMAL	BINARIO	HEXADECIMAL	BINARIO		
90		4B			
34		EE			
381		E5			
91		17			

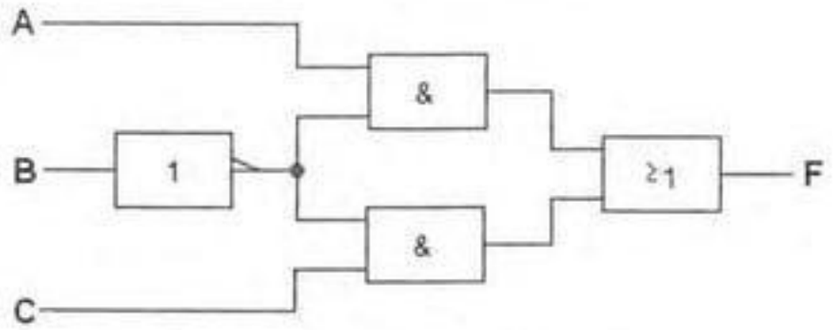
OCTAL	BINARIO	OCTAL	BINARIO	OCTAL	BINARIO
5		40		345	
10		55		500	
25		60		625	
30		100		700	

BINARIO	OCTAL	BINARIO	OCTAL	BINARIO	OCTAL
1000		101010		10	
1010		111111		101	
1001001		100001		110	
101		11001100		111	

4. Obtener la función lógica F indicando a qué tipo de puerta lógica básica corresponde.



5. Obtener la función lógica F y la tabla de la verdad correspondiente.



6. El portón de un garaje (P) se abre cuando se activa un pulsador (entrada S1) y se detecta simultáneamente la presencia de un vehículo (entrada B1). Se pide obtener la tabla de la verdad de la función lógica de control y el circuito lógico implementado mediante puertas NAND de 2 entradas.

7. Un circuito digital posee una entrada de señal E, un interruptor de conmutación S y 2 salidas de señal Q1 y Q2. El funcionamiento del circuito viene dado por:

- Si S = 1, Q1 = E y Q2 = 0;
- Si S = 0, Q1 = 0 y Q2 = E;

Se pide obtener la tabla de la verdad de la función lógica de control y el circuito lógico implementado mediante puertas lógicas.

8. El sistema de seguridad de una prensa hidráulica es un circuito lógico formado por 3 entradas (pulsadores S1 y S2, pedal P) y 2 salidas (motor de la máquina M, señal de alarma AL). El motor se debe poner en marcha si se acciona el pedal y, al menos, uno de los 2 pulsadores. La activación de la alarma se produce cuando el motor está en marcha y no se pulsan simultáneamente los pulsadores. Se pide la tabla de la verdad, la expresión algebraica y el circuito realizado con puertas lógicas para cada función lógica de salida (M y AL).

9. Se dispone de 3 interruptores para activar una sirena de alarma. El dispositivo dará la señal de aviso si se activa un único interruptor cualquiera o si se activan los tres interruptores de forma simultánea. Se pide obtener la función lógica F de las tres formas posibles (tabla de la verdad, expresión algebraica y en forma gráfica mediante puertas lógicas).

1.4. Simplificación de funciones lógicas Mapas de Karnaugh

Cuando la función lógica obtenida en forma canónica es demasiado compleja, se debe aplicar un **método de simplificación** para obtener una función equivalente reducida a su mínima expresión, es decir, que contenga la mínima cantidad de operaciones y de variables. Los métodos más habituales son el **método algebraico** y el método o **mapa de Karnaugh**.

La aplicación del **método algebraico** no es intuitivo, pues requiere desarrollar una cierta habilidad lógico-matemática para aplicar las reglas, leyes y teoremas del álgebra booleana.

Ejemplo 1.14

Realiza la simplificación por el método algebraico de la siguiente función lógica F.

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

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

La simplificación permite obtener una función de 3 sumandos, dos de ellos con dos variables y el tercero con tres variables.

El método o **mapa de Karnaugh** es un diagrama utilizado para la simplificación de funciones lógicas escritas en forma algebraica. Consiste en una representación bidimensional (cuadrícula en forma de cuadrado o rectángulo) de la tabla de la verdad de la función lógica que se desea simplificar. El mapa de Karnaugh, para N variables, debe tener 2^N casillas.

Para utilizar este método, es habitual partir de una función lógica en forma canónica expresada en minitérminos. En el caso que algún minitérmino no contenga todas las variables utilizadas en la función lógica, se debe completar con estas. A continuación, se puede realizar la tabla de la verdad.

Para representar en el mapa los términos de la tabla, se debe construir una matriz nxm donde filas y columnas son las variables consideradas expresadas en **código Gray.** En una fila o columna, entre una casilla y otra consecutiva, existe adyacencia lógica, es decir, que solo cambia una variable. Para 2 variables, desde la casilla 00, si se considera la casilla adyacente en la fila se pasa a la casilla 01. En caso de considerar la casilla adyacente en la columna, se pasa a la casilla 10.

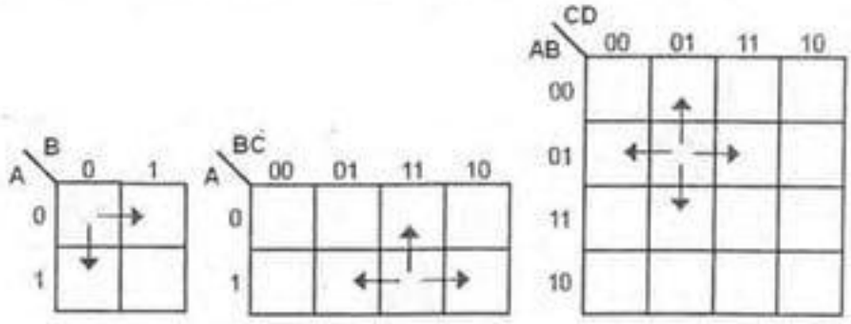


Fig. 1.18. Mapas de Karnaugh para 2, 3 y 4 variables

Es conveniente realizar la numeración de las casillas del mapa de Karnaugh hasta 4 variables en código Gray. Así es más sencillo marcar en el mapa de Karnaugh, con un 1, las casillas correspondientes de los minitérminos y, con un 0, las casillas correspondientes de los maxitérminos de la función lógica.

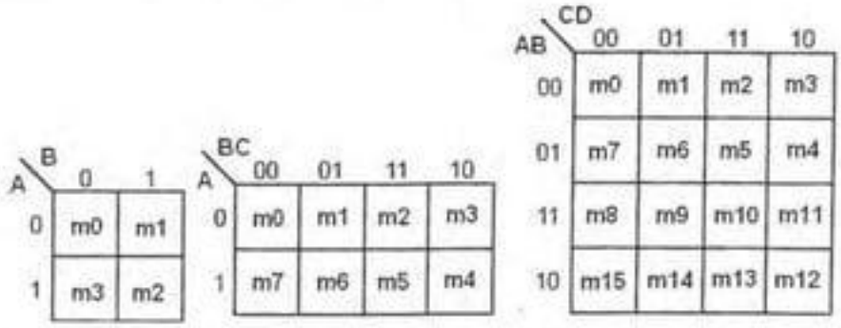


Fig. 1.19. Numeración de casillas en mapas K para 2, 3 y 4 variables

Ejemplo 1.15

Representa directamente la función lógica F en un mapa de Karnaugh de 3 variables.

$$F = f(A,B,C) = \sum_s m\,(0,1,2,3,5);$$

Basta considerar la tabla de la verdad en código Gray de la función lógica F para ver que los minitérminos que se deben marcar con un 1 son aquellos cuyo orden es 0, 1, 2, 3 y 5. La escritura en el mapa de Karnaugh es directa tras identificar las casillas 0, 1, 2, 3 y 5.

3 bits	F
000	1
001	1
011	1
010	1
110	
111	1
101	
100	

A \ BC	00	01	11	10
0	1	1	1	1
1			1	

La estrategia para llevar a cabo la simplificación sobre el mapa de Karnaugh pasa por cubrir todos los minitérminos con los grupos más grandes posibles y con el menor número de grupos. Se procede agrupando «unos» adyacentes en el mapa, en bloques con potencia de base 2 (1, 2, 4, 8, etc.) según los ejes coordenados (nunca en diagonal).

El algoritmo sistemático para simplificar el mapa de Karnaugh puede ser:

1. Cubrir las casillas que no pueden formar grupos de 2.
2. Cubrir las casillas que pueden formar grupos de 2, pero no de 4.
3. Cubrir las casillas que pueden formar grupos de 4, pero no de 8.
4. Cubrir las casillas que pueden formar grupos de 8, pero no de 16.
5. Y así sucesivamente.

Si en algún paso hay más de una opción, es recomendable empezar siempre cubriendo las casillas que tienen menos opciones.

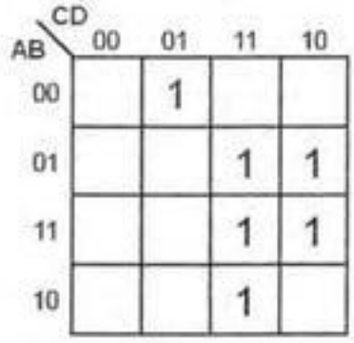
Al final, cada grupo de «unos» es un término producto que se debe escribir con las variables que son comunes. Se pueden repetir términos, si es necesario (propiedad de absorción). Finalmente, la expresión final será la suma lógica OR de todos los términos producto.

Para obtener la expresión lógica simplificada de cada grupo formado de «unos», se debe considerar que:

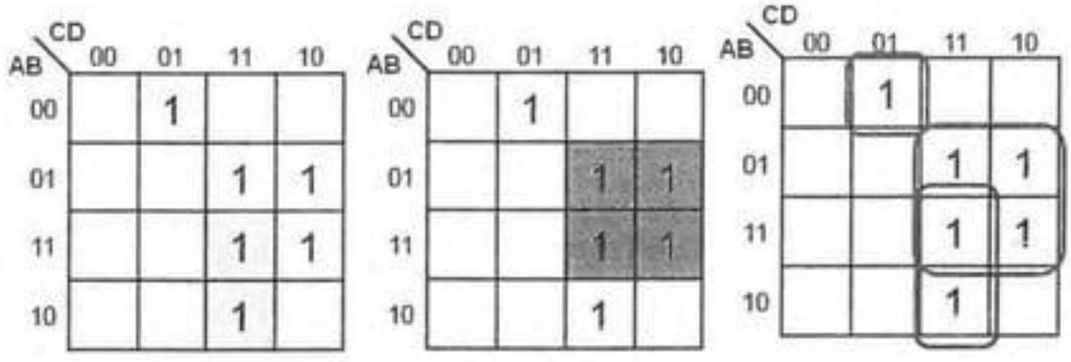
1. En los grupos de 1 celda no desaparece ninguna variable.
2. En los grupos de 2 celdas desaparece 1 variable, la que cambia.
3. En los grupos de 4 celdas desaparecen 2 variables, las que cambian.
4. Y así sucesivamente.

Ejemplo 1.16

Realiza la simplificación de la función lógica dada en su correspondiente mapa de Karnaugh de 4 variables (minitérminos marcados con 1).



Por aplicación del algoritmo propuesto, inicialmente se debe cubrir la casilla suelta (minitérmino en la casilla numerada 1). A continuación, el grupo de 2 formado por las casillas 10 y 13 y, por último, el grupo de 4 formado por las casillas 5, 4, 10 y 11.



Por último, basta escribir la expresión de cada grupo identificado (producto) y la función final total como suma de las expresiones obtenidas en forma de producto.

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

Es interesante consultar en Internet los diferentes programas disponibles que realizan la minimización de los mapas de Karnaugh. De esta manera, se puede comprobar que la simplificación realizada a mano es correcta.

Ejemplo 1.17

Ejemplo 1.17: Realiza la simplificación por mapa de Karnaugh de la función lógica S.

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

La función lógica S, de 4 variables booleanas, está expresada como suma de 5 términos. Existen 2 términos que no están expresados en forma de minitérminos, pues en ellos no están presentes todas las variables. Se deben completar dichos términos con las variables que faltan (negadas y no negadas, con todas sus combinaciones posibles).

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

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

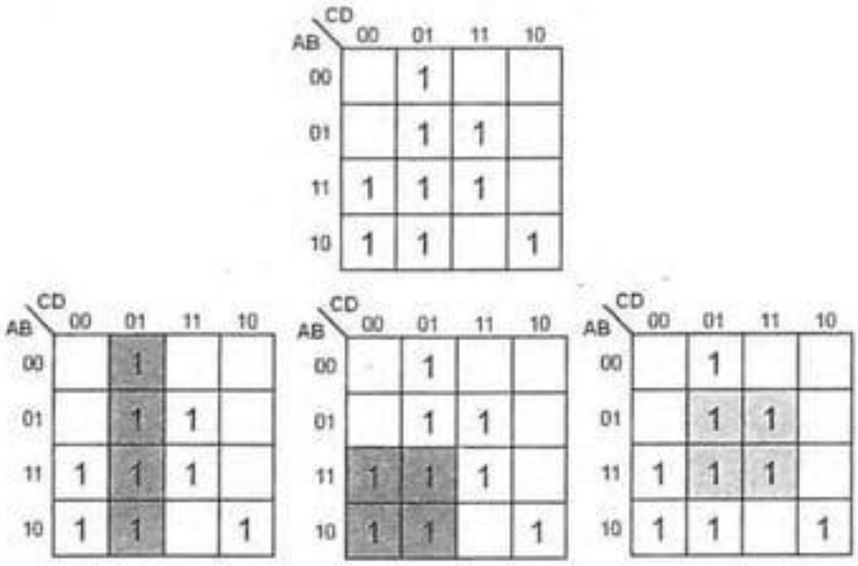
Considerando todas las variables, la tabla de la verdad de la función lógica S es:

A	B	C	D	S
0	0	0	0	0
0	**0**	**0**	**1**	**1**
0	0	1	1	0
0	0	1	0	0
0	1	1	0	0
0	**1**	**1**	**1**	**1**
0	**1**	**0**	**1**	**1**
0	1	0	0	0
1	**1**	**0**	**0**	**1**
1	**1**	**0**	**1**	**1**
1	**1**	**1**	**1**	**1**
1	1	1	0	0
1	**0**	**1**	**0**	**1**
1	0	1	1	0
1	**0**	**0**	**1**	**1**
1	**0**	**0**	**0**	**1**

Sobre el mapa de Karnaugh de 4 variables, se deben representar los «unos» de los minitérminos proporcionados por la función lógica S. Se deben identificar los grupos posibles (de 1, 2 o 4 casillas) de acuerdo al algoritmo proporcionado.

$$F = f(A,B,C,D) = \sum_9 m\ (1,5,6,8,9,10,12,14,15);$$

Se puede formar un grupo de 2 con las casillas 12 y 15 y hasta 3 grupos de 4. Se pueden agrupar las casillas 1, 6, 9 y 14. Otro grupo lo forman las casillas 8, 9, 15 y 14. El último grupo de 4 está compuesto por las casillas centrales (6, 5, 9 y 10).

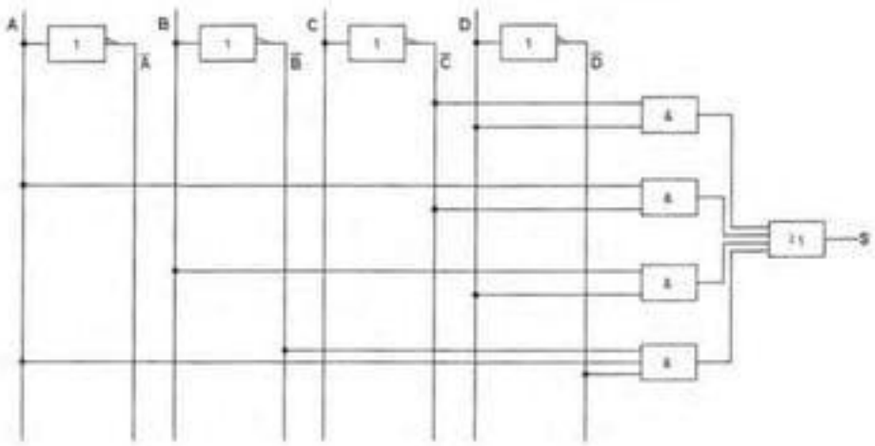


Basta escribir la expresión de cada grupo identificado, desapareciendo la variable o variables que cambian en el grupo.

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

Ejemplo 1.18

Representa de forma gráfica mediante puertas lógicas básicas (símbolos ANSI/IEEE) la función lógica S, simplificada por Karnaugh, del ejemplo anterior.



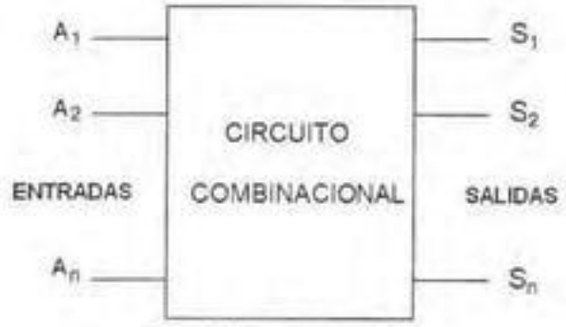
Recuerda • • •

En un sistema combinacional la salida solo depende del estado de las entradas.

1.5. Circuitos combinacionales con puertas lógicas

1.5.1. Definición

Un circuito combinacional es aquel en el que las salidas solo dependen de las distintas combinaciones que se puedan dar con sus variables de entrada en un momento determinado. Por tanto, un circuito combinacional no depende del estado previo de las salidas (diferencia con circuitos secuenciales) y, además, no interviene el factor tiempo.



$$S_i = f(A_1, A_2, \ldots, A_n)$$

Fig. 1.20. Definición de un circuito combinacional

El proceso para llevar a cabo el diseño de un circuito combinacional mediante puertas lógicas requiere de los siguientes pasos:

1. Analizar el enunciado del caso práctico planteado.
2. Escribir la tabla de la verdad del caso práctico planteado.
2. Obtener la función lógica en forma canónica para cada salida que dé como resultado un «uno» lógico.
4. Simplificar la función lógica anterior.
5. Si interesa, realizar la conversión de la función lógica simplificada para utilizar de forma exclusiva puertas de tipo NAND o puertas de tipo NOR.
6. Implementar el diagrama lógico con puertas.
7. Seleccionar los circuitos integrados comerciales necesarios.
8. Montar de forma práctica el circuito y comprobar su funcionamiento.

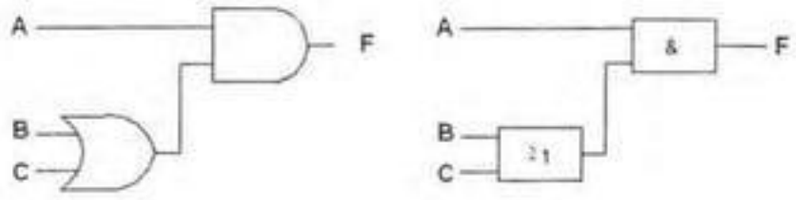
Como se ha comentado, suele ser habitual optar por utilizar un único tipo de puerta lógica (la NAND o la NOR) en el diseño de circuitos combinacionales. La ventaja de este proceder estriba en poder utilizar todas las puertas lógicas integradas en el chip (circuito integrado) comercial. De esta forma, aunque aumente el número de puertas necesarias, no es necesario disponer de todos los tipos de puertas.

Como ya se vio en las reglas gráficas del ejemplo 1.11, la negación NOT () se puede realizar con una puerta NAND o con una puerta NOR. La suma OR () se puede traducir mediante 3 puertas NAND en forma de triángulo o utilizando 2 puertas NOR en serie. Por último, el producto AND () se puede traducir con 2 puertas NAND en serie o mediante 3 puertas NOR en forma de triángulo.

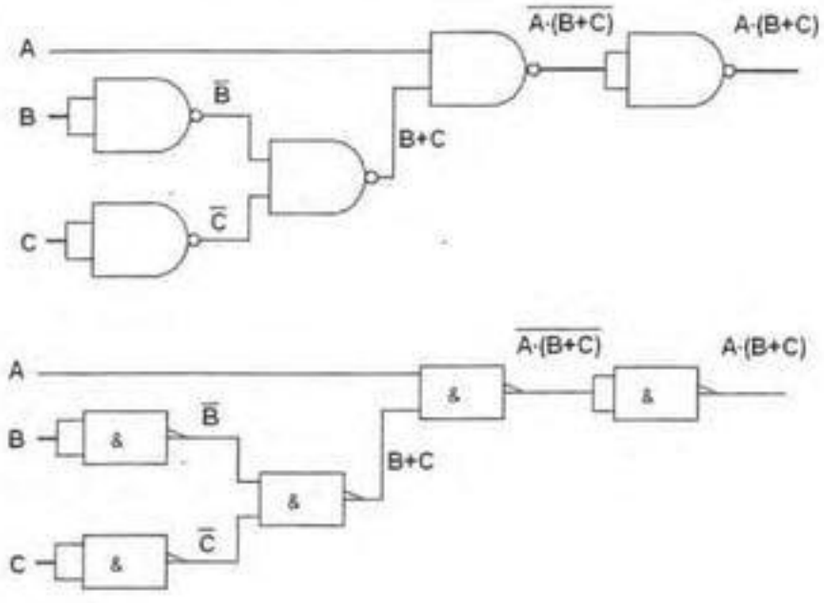
Ejemplo 1.19

Diseña el circuito lógico dado por la función lógica simplificada F mediante solo puertas NAND.

$$F = f(A, B, C) = A \cdot (B + C);$$



De acuerdo con las reglas gráficas definidas, la suma se debe traducir por 3 puertas NAND de 2 entradas dispuestas en forma de triángulo. El producto se debe traducir por 2 puertas NAND de 2 entradas en serie.



Mediante reglas algebraicas, el algoritmo que puede utilizarse para implement una función lógica solo con puertas lógicas de un tipo (NAND o NOR) puede ser:

a) Cuando se desea sintetizar solo con puertas NAND se deben hacer los siguie tes pasos:

1º Negar la función dos veces.

2º Si la operación más externa es una suma, convertirla en un producto por MO GAN. Si la operación más externa es un producto, se deja como está.

3º Si internamente en la expresión hay una suma, se debe negar dos veces y co vertirla en un producto.

b) Cuando se desea sintetizar solo con puertas NOR se deben hacer los siguient pasos:

1º Negar la función dos veces.

2º Si la operación más externa es un producto, convertirla en una suma por MO GAN. Si la operación más externa es una suma, se deja como está.

3º Si internamente en la expresión hay un producto, se debe negar dos vece y convertirlo en una suma.

1.5.2. Circuitos integrados

Para llevar a cabo la implementación del esquema electrónico de un circuito combinacional se deben seleccionar **circuitos integrados CI** (chips) disponibles comercialmente. Los circuitos integrados están formados por un conjunto de componentes electrónicos (resistencias, diodos, transistores) integrados en una sola pieza de material semiconductor a base de silicio e insertada en el interior de un encapsulado. Los encapsulados más habituales son el DIP, el PLCC, el SOIC y el SSOP.

El encapsulado DIP tiene 2 hileras de patillas numeradas que se conectan a través de un zócalo al circuito de forma sencilla. El terminal o patilla marcada con el 1 se sitúa en un pequeño agujero situado en un extremo. Como ejemplo se puede considerar el integrado 74HCT08 de Philips que integra 4 puertas lógicas AND de 2 entradas.

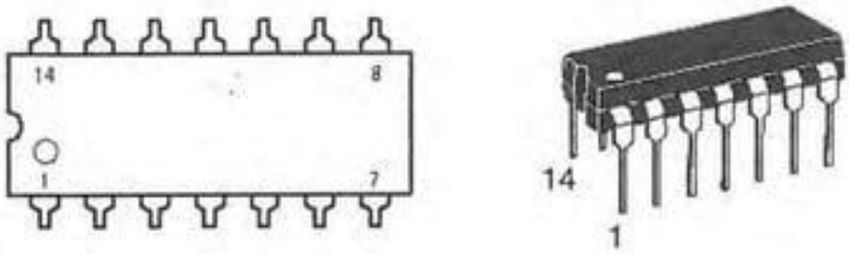


Fig. 1.21. Numeración de patillas en un CI con encapsulado DIP

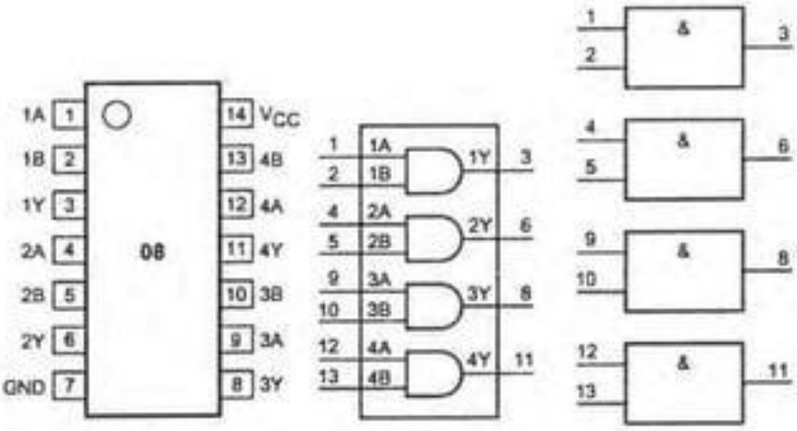


Fig. 1.22. Integrado 74HCT08 de Philips (4 puertas AND)

La fabricación de los circuitos integrados digitales puede hacerse mediante la aplicación de diferentes tecnologías. Una **familia lógica** es el conjunto de todos los componentes lógicos fabricados con la misma tecnología. Actualmente, las 2 familias lógicas más utilizadas son la TTL (basada en transistores bipolares) y la CMOS (basada en transistores unipolares MOSFET de canal N y de canal P).

En cualquiera de las familias lógicas, es interesante conocer los principales parámetros y características de un circuito lógico integrado (tensión de alimentación Vcc, niveles de tensión de entrada y salida, inmunidad al ruido, disipación de potencia, tiempo de propagación, etc.).

	TTL 74	CMOS 74HC
Tensión de alimentación Vcc	4,5-5,5 V	3-15 V
Niveles de tensión de entrada	0,8-2 V	1-3,5 V
Niveles de tensión de salida	0,4-2,4 V	0,1-4,9 V
Margen de ruido	0,4 V	0,9-1,4 V
Potencia consumida	10 mW	0,0025 mW
Tiempo de propagación	9 ns	8 ns

Fig. 1.23. Comparativa entre familias lógicas

La **familia TTL** (lógica de transistor a transistor) ha sido durante más de 2(años la más utilizada debido a su bajo coste y gran variedad de circuitos fabricados En la actualidad, se utilizan en los CI de pequeña y mediana escala y van siendo sus tituidas por otras tecnologías como la CMOS. La primera serie que apareció en e mercado fue la TTL estándar o serie 74. A partir de esta, se han desarrollado nueva series para mejorar sus prestaciones (serie 74S para mejorar el tiempo de propa gación, serie 74LS con menor consumo de potencia, serie 74AS con bajo tiempo d propagación, serie 74F con buen equilibrio entre potencia y tiempo, etc.).

Normalmente, junto a la denominación de la serie, se añaden un par de dígito con los que se indica el tipo y el número de puertas lógicas que van incluidas er el integrado. Así, como ejemplo, el 08 indica un integrado con 4 puertas AND d 2 entradas.

00	4 puertas NAND de 2 entradas
02	4 puertas NOR de 2 entradas
04	6 puertas NOT
08	4 puertas AND de 2 entradas
10	3 puertas NAND de 2 entradas
11	3 puertas AND de 2 entradas
20	2 puertas NAND de 4 entradas
21	2 puertas AND de 4 entradas
27	3 puertas NOR de 3 entradas
30	1 puerta AND de 8 entradas
32	4 puertas OR de 2 entradas
86	4 puertas XOR de 2 entradas
133	1 puerta NAND de 13 entradas

Fig. 1.24. Significado de los últimos dígitos de un CI

La **familia CMOS** (*Complementary Metal-Oxide Semiconductor*) se ha ido imponiendo en el mercado gracias a la mejora de alguna de sus características, si bien tienen un tiempo de propagación algo mayor y son muy sensibles a la electricidad estática. La primera serie que apareció en el mercado fue la 4000. A partir de esta, se han desarrollado nuevas series (serie 74HC para mejorar la velocidad de conmutación, serie 74HCT que es eléctricamente compatible con la TTL, serie 74LV con menor tamaño y potencia consumida, etc.). La serie 74HCT, con el mismo orden en los terminales que los TTL, posee un menor tiempo de propagación y menor potencia de disipación y es la más utilizada en la actualidad.

Ejemplo 1.20

Diseña un circuito combinacional que sea capaz de activar dos motores «M1 y M2» en las siguientes condiciones de los interruptores S1, S2, S3 y S4.

El motor M1 se activará cuando:

Todos los interruptores estén a cero.

S1 esté conectado y los otros no.

S3 esté conectado y los otros no.

S1 y S3 estén conectados y los otros no.

El motor M2 se activará cuando:

-S4 esté conectado y los otros no.

-S3 y S4 estén conectados y los otros no.

-S1 y S4 estén conectados y los otros no.

-S1, S3 y S4 estén conectados y los otros no.

La tabla de la verdad para la función lógica del motor M1 es:

S1	S2	S3	S4	M1
0	0	0	0	1
1	0	0	0	1
0	0	1	0	1
1	0	1	0	1

La función lógica para el motor M1, expresada en forma algebraica, consta de 4 términos formados por combinaciones de las 4 variables de entrada utilizadas. La forma simplificada puede obtenerse por aplicación del mapa de Karnaugh para 4 variables.

$$M1 = \overline{S1} \cdot \overline{S2} \cdot \overline{S3} \cdot \overline{S4} + S1 \cdot \overline{S2} \cdot \overline{S3} \cdot \overline{S4} + \overline{S1} \cdot \overline{S2} \cdot S3 \cdot \overline{S4} + S1 \cdot \overline{S2} \cdot S3 \cdot \overline{S4} ;$$

El mapa de Karnaugh y la función lógica simplificada para el motor M1 es:

S1S2 \ S3S4	00	01	11	10
00	1			1
01				
11				
10	1			1

$$M1 = \overline{S1} \cdot \overline{S2} \cdot \overline{S4} + S1 \cdot \overline{S2} \cdot \overline{S4} = \overline{S2} \cdot \overline{S4} ;$$

La tabla de la verdad para la función lógica del motor M2 es:

S1	S2	S3	S4	M2
0	0	0	1	1
0	0	1	1	1
1	0	0	1	1
1	0	1	1	1

La función lógica para el motor M2 consta de 4 términos con las 4 variables de entrada utilizadas. La forma simplificada puede obtenerse por aplicación del mapa de Karnaugh para 4 variables.

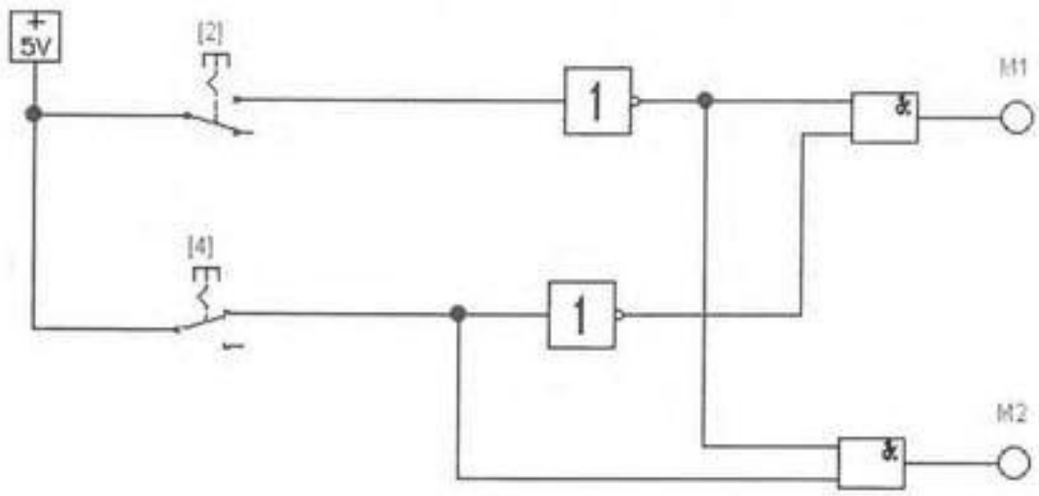
$$M2 = \overline{S1} \cdot \overline{S2} \cdot \overline{S3} \cdot S4 + \overline{S1} \cdot \overline{S2} \cdot S3 \cdot S4 + S1 \cdot \overline{S2} \cdot \overline{S3} \cdot S4 + S1 \cdot \overline{S2} \cdot S3 \cdot S4 ;$$

El mapa de Karnaugh y la función lógica simplificada para el motor M2 es:

S1S2 \ S3S4	00	01	11	10
00		1	1	
01				
11				
10		1	1	

$$M2 = \overline{S2} \cdot S4 ;$$

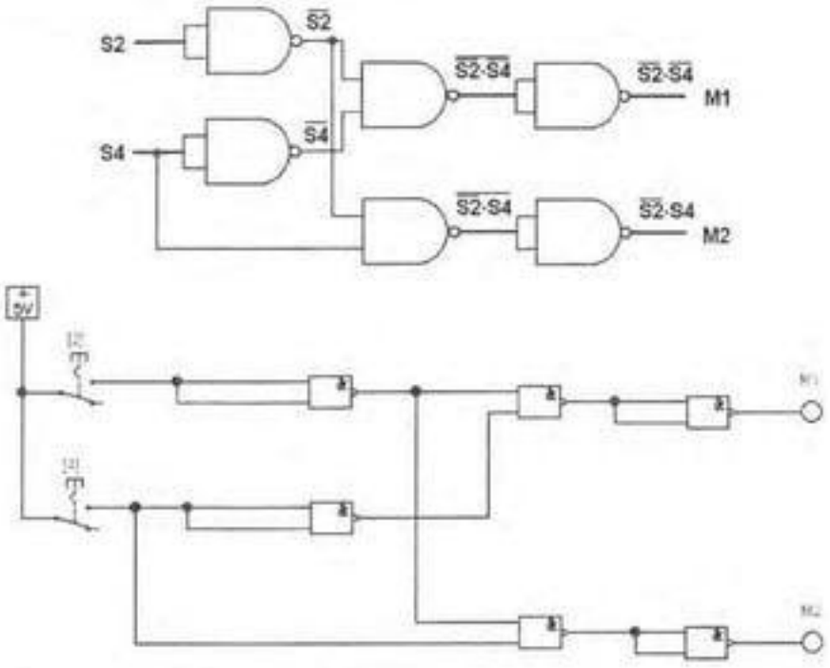
De forma gráfica, el circuito combinacional simplificado necesario para activar los motores M1 y M2 está formado por las entradas S2 y S4, 2 puertas NOT, 2 puertas AND y las salidas M1 y M2.



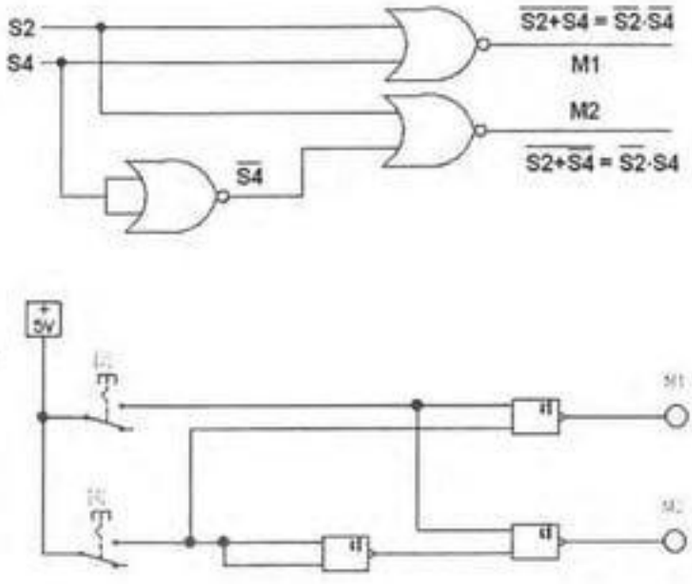
Ejemplo 1.21

Diseña el circuito combinacional del ejemplo anterior utilizando solo puertas NAND o solo puertas NOR.

Puertas NAND: De forma gráfica, la negación de cada variable (S2, S4) requiere de una puerta NAND. El producto AND se puede traducir con 2 puertas NAND en serie para la salida M1 y con otras 2 puertas NAND en serie para la salida M2.



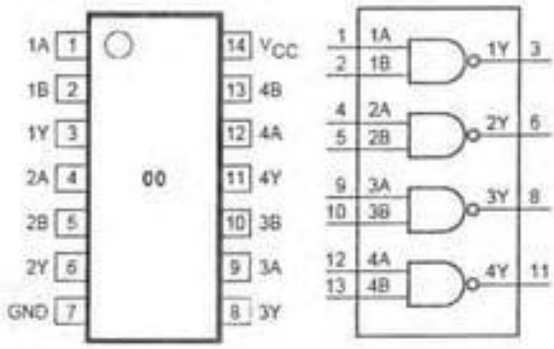
Puertas NOR: De forma gráfica, la negación de cada variable (S2, S4) requiere de una puerta NOR. Para la salida M1, el producto AND necesita 3 puertas NOR en forma de triángulo. No obstante, de las 5 puertas indicadas, basta con utilizar una puerta NOR (no tiene sentido negar una entrada para después negar la entrada negada). Para la salida M2, se debe negar la entrada S4 y luego utilizar una única puerta NOR adicional.

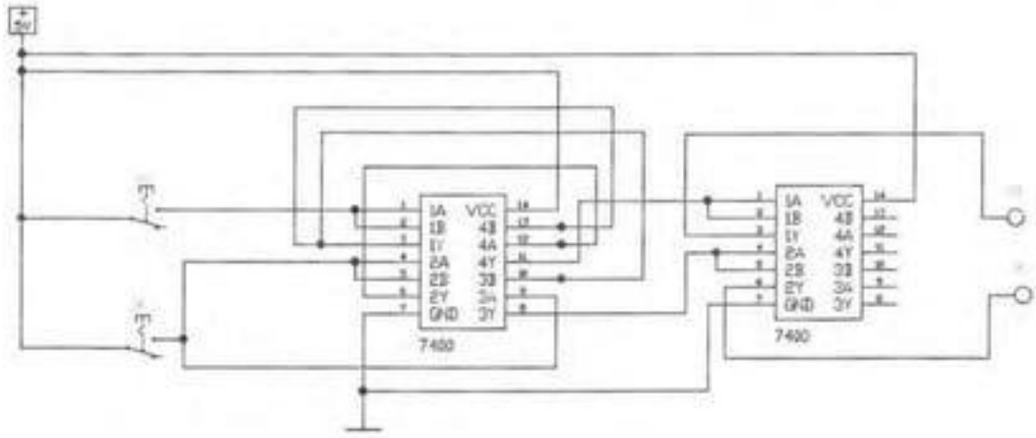


Ejemplo 1.22

Implementa el circuito combinacional del ejemplo anterior utilizando circuitos integrados comerciales solo con puertas NAND o solo con puertas NOR.

SERIE 74HCT00 (4 puertas NAND de 2 entradas)





En el circuito integrado CI1, el patillaje que se ha conectado es:

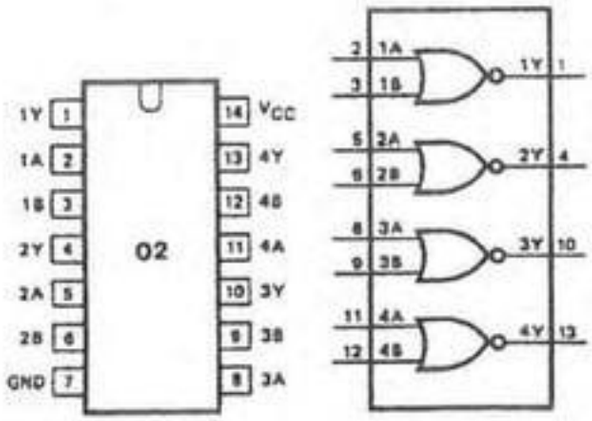
1A, 1B= $S2$;	4A= 2Y; 4B= 1Y ;
1Y= $\overline{S2}$;	4Y= $\overline{\overline{S2} \cdot \overline{S4}}$;

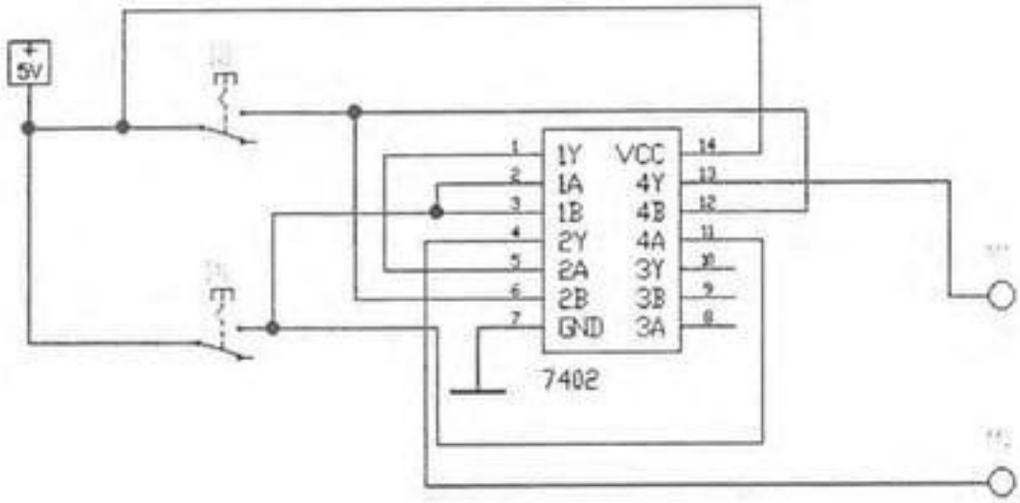
2A, 2B= $S4$;	3B= 1Y ; 3A= 2A ;
2Y= $\overline{S4}$;	3Y= $\overline{\overline{S2} \cdot \overline{S4}}$;

En el circuito integrado CI2, el patillaje que se ha conectado es:

1A, 1B= 4Y= $\overline{\overline{S2} \cdot \overline{S4}}$;	2A, 2B= 3Y= $\overline{\overline{S2} \cdot \overline{S4}}$;
1Y= M1 ;	2Y= M2 ;

SERIE 74HCT02 (4 puertas NOR de 2 entradas)





En el circuito integrado CI1, el patillaje que se ha conectado es:

4B= $S2$;	1A, 1B= $S4$;
4A= $S4$;	1Y= $\overline{S4}$ = 2A ; 2B= $S2$;
4Y= M1 ;	2Y= M2 ;

Actividades propuestas

1. Diseña un sistema combinacional que sea capaz de detectar cuándo a sus entradas le introducimos una combinación que exprese una cantidad que sea en decimal 0, 3 o 7. Se pide:
   - Ecuaciones simplificadas. Implementación de la ecuación con las puertas correspondientes.
   - Implementación de la ecuación con relés y prueba en CADe_SIMU.
   - Implementación con puertas NAND y NOT en Workbench.
   - Implementación con puertas NOR y NOT en Workbench.

2. Diseña un circuito constituido por tres pulsadores, A, B, C, y una lámpara que funcione de forma que esta se encienda cuando se pulsen los tres pulsadores a la vez o uno cualquiera solamente. Se pide:
   - Ecuaciones simplificadas.
   - Implementación con puertas NOR en Workbench.
3. Se desea gobernar un motor eléctrico por medio de tres interruptores A, B y C. Se pondrá el motor en marcha en los siguientes casos:
   a) Cuando se accione solamente el interruptor A.
   b) Cuando se accionen a la vez los interruptores A y C.
   c) Cuando se accionen a la vez los interruptores B y C.

   Se pide:
   - Ecuaciones simplificadas.
   - Implementación de la ecuación con relés y prueba en CADe_SIMU.
   - Implementación con puertas NOR en Workbench.
4. Se desean gobernar dos motores M1 y M2 por medio de tres interruptores A, B y C, según:
   a) Si se acciona solamente el interruptor A, funciona solamente el motor M1.
   b) Si se acciona solamente el interruptor B, funcionan los motores M1 y M2.
   c) Si se acciona solamente el interruptor C, funciona el motor M2.
   d) Si se accionan a la vez los interruptores A y C, funciona el motor M1.

   Se pide:
   - Ecuaciones simplificadas.
   - Implementación de la ecuación con relés y prueba en CADe_SIMU.
   - Implementación con puertas NOR y NOT en Workbench.
5. Desde dos puntos de accionamiento S1 e S2 (de contactos indeterminado) se desea controlar 5 lámparas L1, L2, L3, L4 y L5:
   1. Si se acciona solo el punto S1 se encienden las lámparas L1, L2, L3.
   2. Si se acciona solo el punto S2 se encienden las lámparas L4, L2, L5.
   3. Si se acciona a la vez S1 y S2 se enciende solo la lámpara L2.

   Se pide:
   - Ecuaciones simplificadas.
   - Implementación con puertas NAND y NOT
6. Diseña un automatismo combinacional que sea capaz de activar dos motores M1 y M2 en las siguientes condiciones de los interruptores NO S1, S2, S3 y S4.

   M1 se activará cuando:
   - S2 y S4 están conectados y los demás no.
   - S1, S2 y S4 están conectados y los demás no.
   - S2, S3 y S4 están conectados y los demás no.
   - Todos están conectados.

M2 se activará cuando:

- S2 conectado y los demás no.
- S1 y S2 conectados y los demás no.
- S2 y S3 conectados y los demás no.
- S1, S2 y S3 conectados y S4 desconectado.

Se pide:

a) Ecuaciones simplificadas por Karnaugh de la función para cada motor.

b) Implementar las ecuaciones de M1 y M2 con puertas NOT y NAND en Workbench.

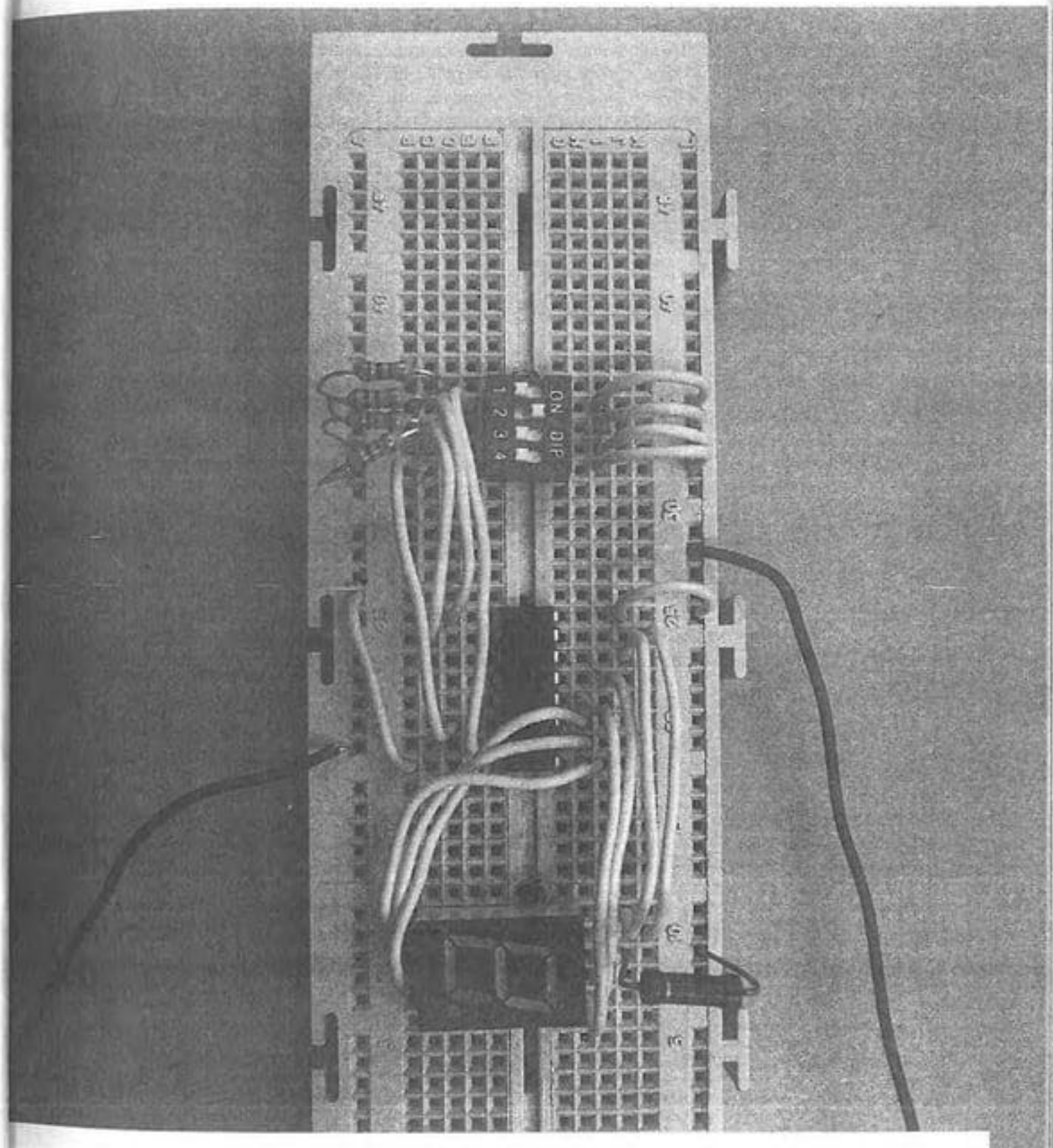
c) Implementar las ecuaciones de M1 y M2 con puertas NOT y NOR en Workbench.

d) Buscar las características de los circuitos integrados TTL y CMOS 7404 y 7400.

e) Implementar las ecuaciones de M1 y M2 con circuito integrado 7404 de puertas NOT y 7400 de puertas NAND en Workbench.

f) Implementar las ecuaciones de M1 y M2 con puertas NAND de dos entradas en Workbench.

Unidad 2 Bloques combinacionales



En este capítulo:

2.1. Codificadores

2.2. Multiplexores

2.3. Demultiplexores

2.4. Decodificadores

Recuerda • • •

Para implementar un sistema combinacional con puertas lógicas, se debe simplificar al máximo y tener en cuenta que los circuitos integrados de puertas lógicas son de 2 entradas.

Una vez definido lo que se entiende por un circuito combinacional, es inte sante abordar el estudio de aquellos elementos y bloques combinacionales u lizados comúnmente. De forma genérica se deben definir los codificadores, multiplexores, los decodificadores y los demultiplexores.

2.1. Codificadores

Un codificador es un circuito combinacional que permite codificar en forma bi ria la información numérica o alfanumérica aplicada a su entrada. Es decir, el co ficador produce un **código de salida de N bits** según la entrada que se encuen activa en cada momento (estado lógico 1). Para cada entrada (E_0, E_1, E_2,....) exi un código distinto asociado a la salida (S_0, S_1,...), de forma que solo una de las tradas puede estar activa cada vez.

Fig. 2.1. Esquema de un circuito combinacional codificador

El ejemplo típico es un codificador decimal a BCD con 10 entradas, en las que s consideran los números decimales del 0 al 9, y 4 salidas binarias que permite codificar los datos decimales en BCD en función de la combinación de bits (0, 1) d cada salida. La tabla de la verdad nos indica el código BCD devuelto en cada casc

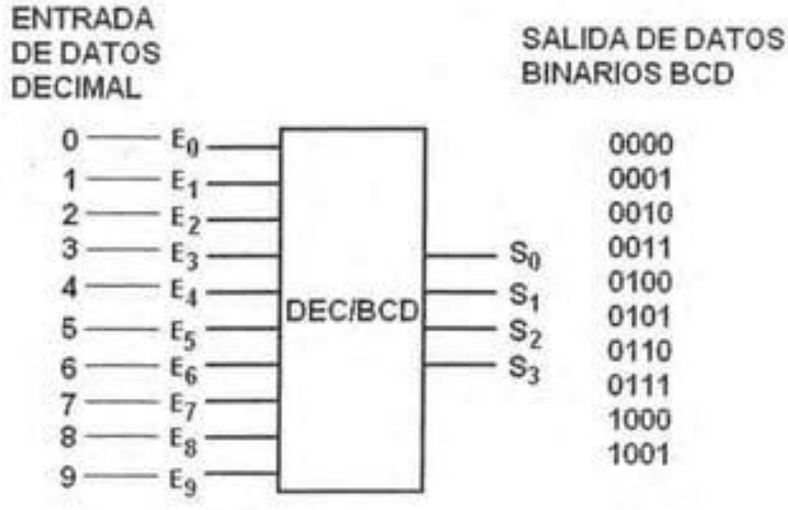


Fig. 2.2. Esquema de un codificador decimal a BCD

DEC	E_9	E_8	E_7	E_6	E_5	E_4	E_3	E_2	E_1	E_0	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
2	0	0	0	0	0	0	0	1	0	0	0	0	1	0
3	0	0	0	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	0	1	0	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0	0	0	1	0	1
6	0	0	0	1	0	0	0	0	0	0	0	1	1	0
7	0	0	1	0	0	0	0	0	0	0	0	1	1	1
8	0	1	0	0	0	0	0	0	0	0	1	0	0	0
9	1	0	0	0	0	0	0	0	0	0	1	0	0	1

Fig. 2.3.
Tabla de la verdad para un codificador decimal a BCD

Ejemplo 2.1

Realiza la tabla de la verdad y la implementación mediante puertas lógicas de un codificador decimal a BCD sin prioridad de 4 entradas y 2 salidas.

Tabla de la verdad:

DEC	A_0	A_1	A_2	A_3	S_1	S_0
0	1	0	0	0	0	0
1	0	1	0	0	0	1
2	0	0	1	0	1	0
3	0	0	0	1	1	1

Funciones lógicas iniciales y simplificadas por Karnaugh:

Las casillas que tienen una X en el mapa de Karnaugh representan salidas que nunca se producirán. Se les puede asignar el valor que más convenga para obtener la expresión lógica más simplificada.

Para la salida S_0:

$$S_0 = \overline{A_0} \cdot A_1 \cdot \overline{A_2} \cdot \overline{A_3} + \overline{A_0} \cdot \overline{A_1} \cdot \overline{A_2} \cdot A_3 \; ;$$

A0·A1 \ A2·A3	00	01	11	10
00		1		
01	1	X		
11				
10				

A0·A1 \ A2·A3	00	01	11	10
00		1		
01	1	X		
11				
10				

$$S_0 = \overline{A_0} \cdot A_1 \cdot \overline{A_2} + \overline{A_0} \cdot \overline{A_2} \cdot A_3 = \overline{A_0} \cdot \overline{A_2} \cdot (A_1 + A_3) \; ;$$

Para la salida S_1:

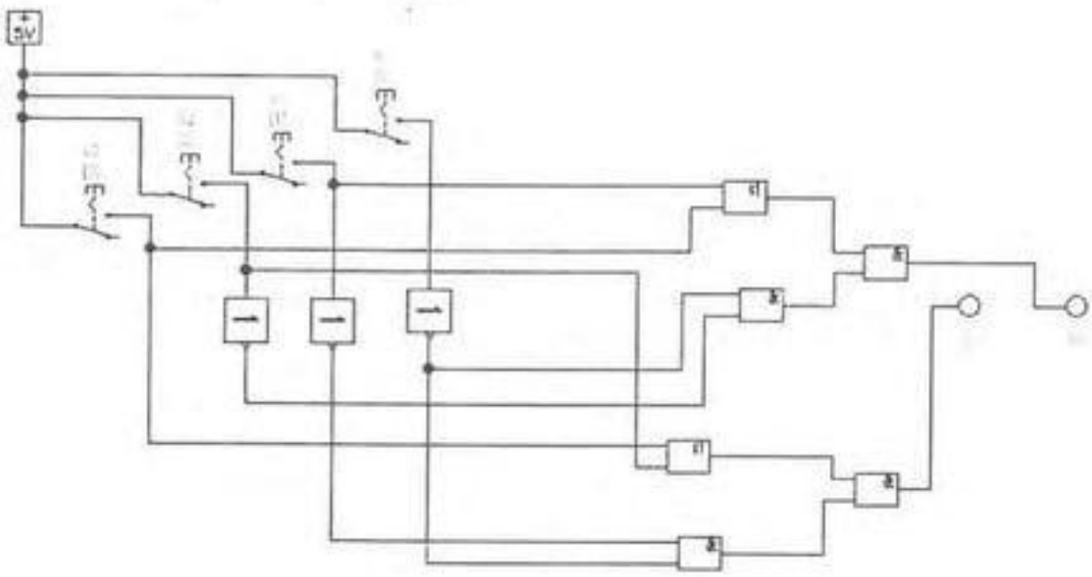
$$S_1 = \overline{A_0} \cdot \overline{A_1} \cdot A_2 \cdot \overline{A_3} + \overline{A_0} \cdot \overline{A_1} \cdot \overline{A_2} \cdot A_3 \; ;$$

A0·A1 \ A2·A3	00	01	11	10
00		1	X	1
01				
11				
10				

A0·A1 \ A2·A3	00	01	11	10
00		1	X	1
01				
11				
10				

$$S_1 = \overline{A_0} \cdot \overline{A_1} \cdot A_3 + \overline{A_0} \cdot \overline{A_1} \cdot A_2 = \overline{A_0} \cdot \overline{A_1} \cdot (A_2 + A_3) \; ;$$

Esquema de puertas lógicas (NOT, AND y OR):



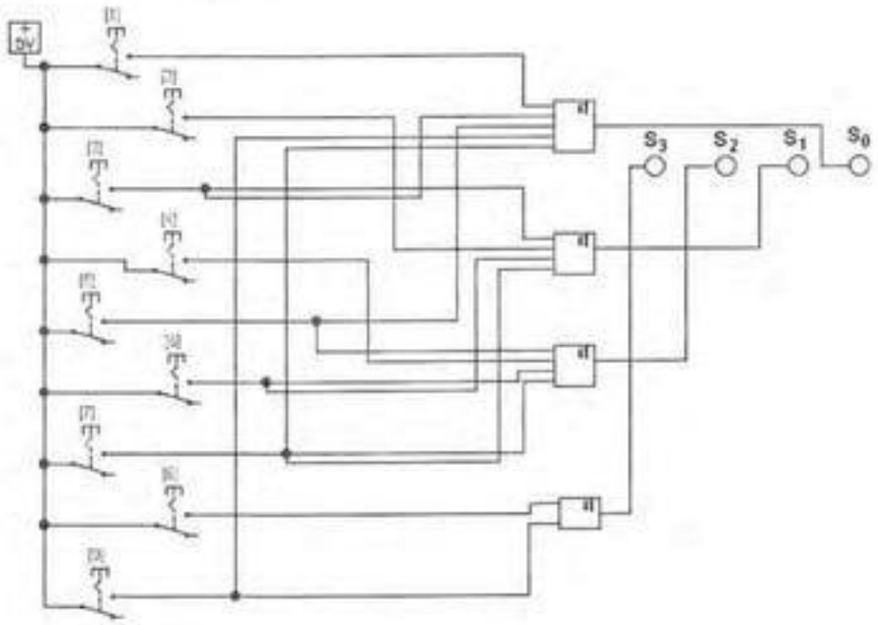
Ejemplo 2.2

Realiza la implementación mediante puertas lógicas de un codificador decimal a BCD sin prioridad de 10 entradas y 4 salidas (S_3, S_2, S_1 y S_0).

Utilizando la tabla de la verdad de la figura 2.3 se aprecia que las combinaciones lógicas que hacen que valga 1 cada una de las salidas del codificador son:

- Para la salida S_0: Suma de las entradas (números decimales) 1 + 3 + 5 + 7 + 9;
- Para la salida S_1: Suma de las entradas (números decimales) 2 + 3 + 6 + 7;
- Para la salida S_2: Suma de las entradas (números decimales) 4 + 5 + 6 + 7;
- Para la salida S_3: Suma de las entradas (números decimales) 8 + 9;

Esquema de puertas lógicas (solo OR):



Es interesante comentar que, para asegurar que solo una entrada está activa a la vez, se han desarrollado **codificadores con prioridad**. Estos circuitos producen solo la salida BCD del dígito decimal de entrada de más alto orden que esté activo. Por ejemplo, si se activan a la vez las entradas 2 y 6, el codificador devuelve a la salida el código BCD 0110, que corresponde al dígito de más peso que es el 6.

El circuito integrado 74HC/HCT147 de la familia lógica CMOS es un codificador con prioridad con entradas a nivel bajo (números decimales del 1 al 9) y 4 salidas BCD activas a nivel bajo.

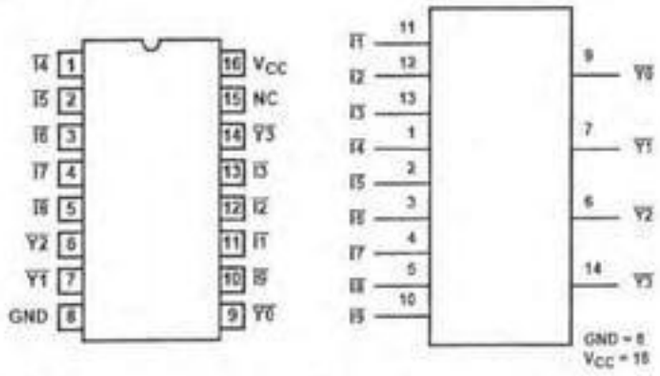


Fig. 2.4. Patillaje y símbolo lógico de un codificador 74HC147

El integrado 74HC147 tiene las entradas y salidas negadas. Si se desea introducir el número decimal 5, se debe poner un 0 en el pin correspondiente (pin 2) y un 1 en los pines de las entradas de la 6 a la 9, ambas incluidas. De esta forma, en el CI entra lo contrario de lo introducido (un 1 en la entrada 5 y un 0 en las entradas de la 6 a la 9 incluidas). Para evitar confusiones, es adecuado poner una puerta NOT en cada entrada que controla un número y así, para introducir el número decimal 5, se pone un 1 antes de la puerta NOT que ataca la entrada 5 del integrado.

Como las salidas del integrado son negadas, se deberá volver a negarlas para obtener el código BCD 0101 equivalente. En caso de no negarlas, el código BCD devuelto sería el 1010 (que es el número decimal 10).

INPUTS									OUTPUTS			
1	2	3	4	5	6	7	8	9	3	2	1	0
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

H = Nivel lógico alto (1); L = Nivel lógico bajo (0); X = Irrelevante;

Fig. 2.5. Tabla de la verdad del codificador 74HC147

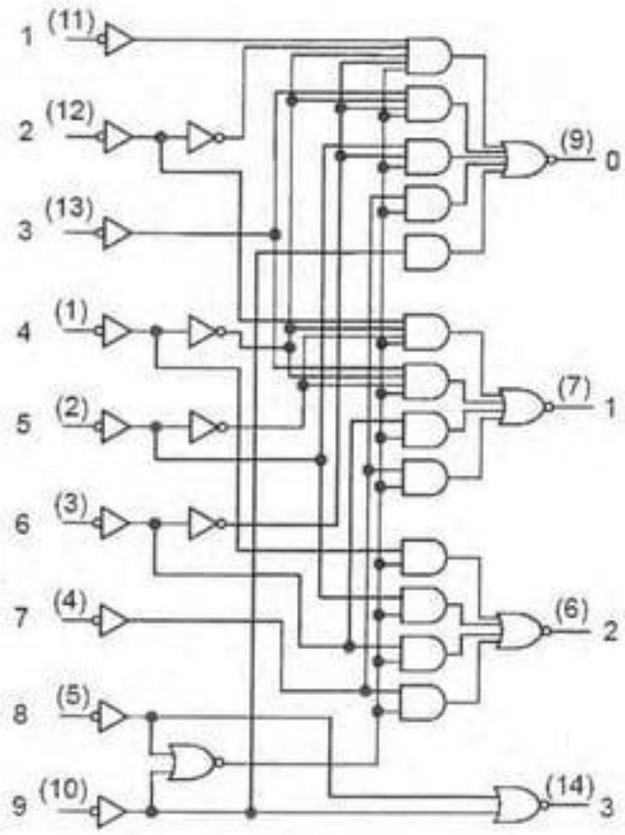
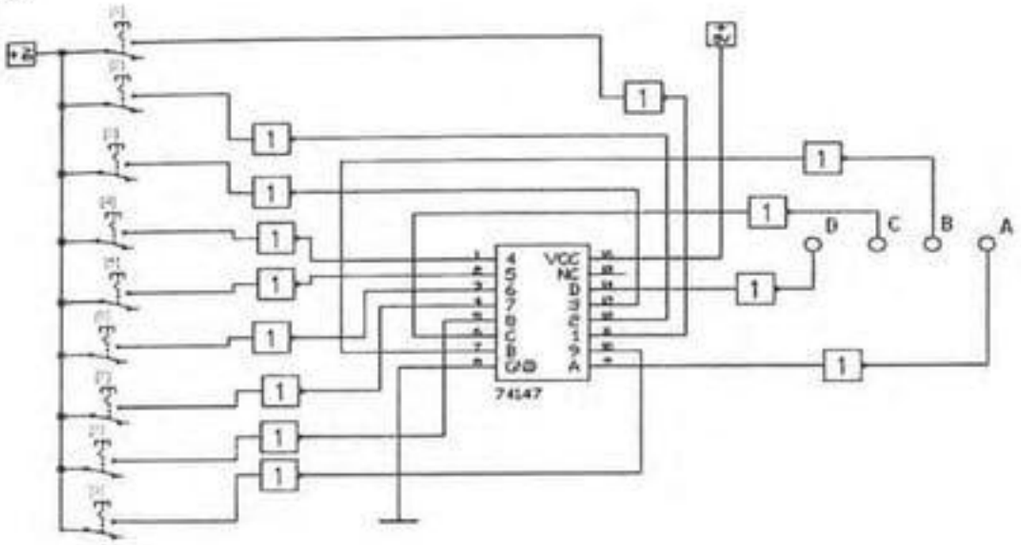


Fig. 2.6. Diagrama lógico del codificador 74HC147

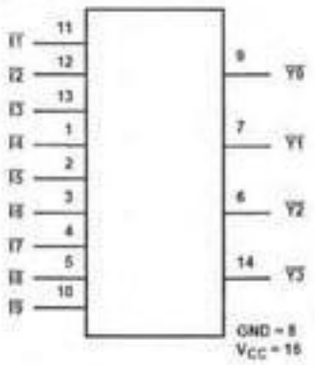
Ejemplo 2.3

Realiza la implementación mediante el CI 74HC147 de un codificador decimal a BCD con prioridad de 10 entradas y 4 salidas (D, C, B y A). Recuerda que se deben negar tanto las entradas como las salidas al CI indicado.



Actividades propuestas

1. Obtén la función lógica para que mediante dos interruptores S1 y S8 se cumplan los siguientes accionamientos en las salidas del circuito integrado. Cablea en Workbench el circuito integrado para que funcione como indica la tabla.



S1	S8	Y0	Y3
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	0

Resultado:

Como las entradas y salidas del codificador son negadas.

La salida Y0 se activará cuando S1 y S8 no son iguales y tengo S1.

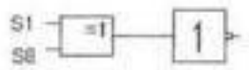
La salida Y3 se activará cuando S1 y S8 no son iguales y tengo S8.

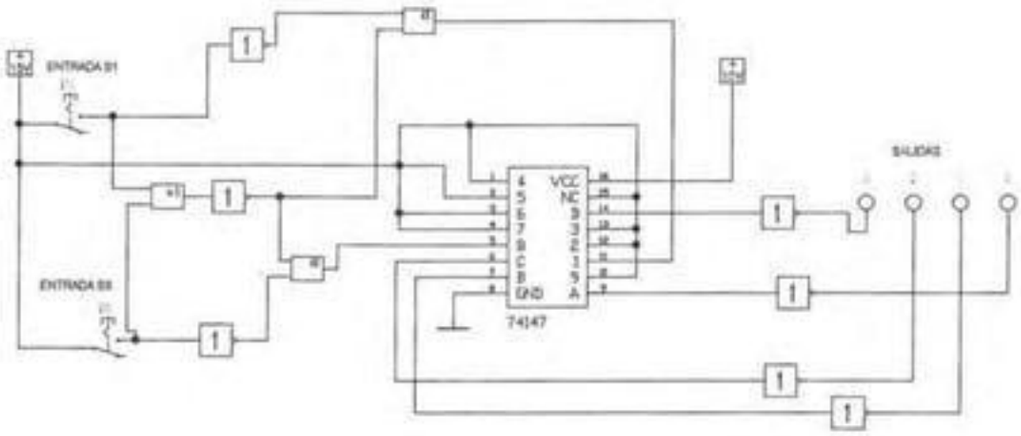
S1	S8	ACTIVACION DE Y0 E Y3	ENTRADA NEGADA AL INTEGRADO
0	0	NO	1
0	1	SI	0
1	0	SI	0
1	1	NO	1

Esto es una puerta XOR negada.

$$Y0 = \overline{\left(\left(\overline{S1} \times S8\right) + \left(S1 \times \overline{S8}\right)\right)} + \overline{S1}$$

$$Y3 = \overline{\left(\left(\overline{S1} \times S8\right) + \left(S1 \times \overline{S8}\right)\right)} + \overline{S8}$$





2. Se desea gobernar cuatro motores eléctricos D_C_B_A. por medio de tres interruptores S3_S7_S9 según la tabla siguiente:

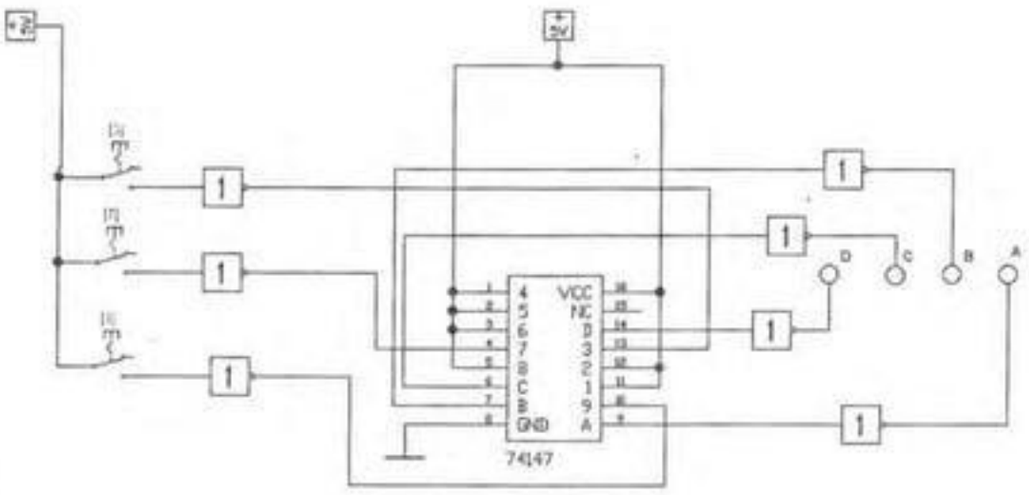
S3	S7	S9	MD	MC	MB	MA
1	0	0	0	0	1	1
X	1	0	0	1	1	1
X	X	1	1	0	0	1

Se pide el circuito combinacional con codificador 74147 que funcione correctamente.

Las entradas de la 1 a la 8 son negadas, + 5V Desactivadas, $\overline{+5V}$ Activadas.

Las salidas son negadas.

Resultado:

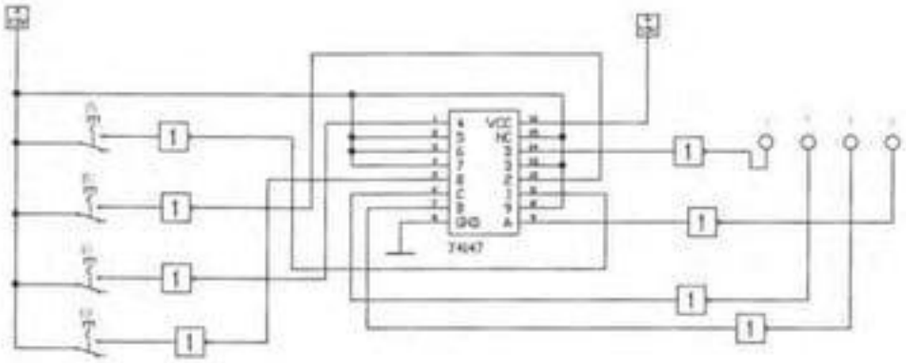


3. Mediante cuatro pulsadores NO S0, S1, S2 y S3 se desea activar las salidas del circuito integrado como indica la tabla. 0 = Pulsador abierto. 1 = pulsador cerrado.

S0	S1	S2	S3	Y0	Y1	Y2	Y3
1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1

Se pide la función lógica de activación de cada salida y el montaje del circuito en Workbench.

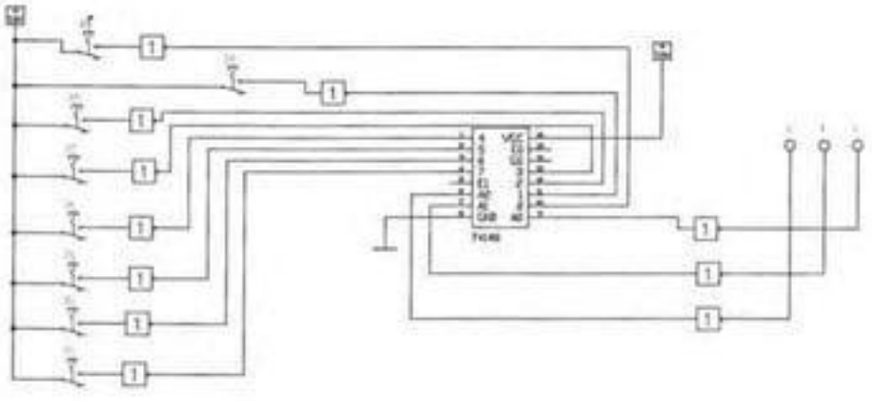
Resultado:



4. Monta el circuito combinacional en Workbench con el codificador 74148 para codificar la numeración decimal a binario que indica la tabla.

Entradas decimal	Salidas en binario		
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Resultado:



2.2. Multiplexores

Un multiplexor es un circuito que permite seleccionar cuál de varias líneas de e tradas de datos (E_0, E_1, E_2...) va a aparecer en una única línea de salida S. Para el se deben configurar unas líneas de control o entradas de selección A. Se de cumplir que por cada **n** entradas de selección haya **2^n** entradas de datos (por ejer plo, se necesitan 2 entradas de selección para 4 entradas de datos).

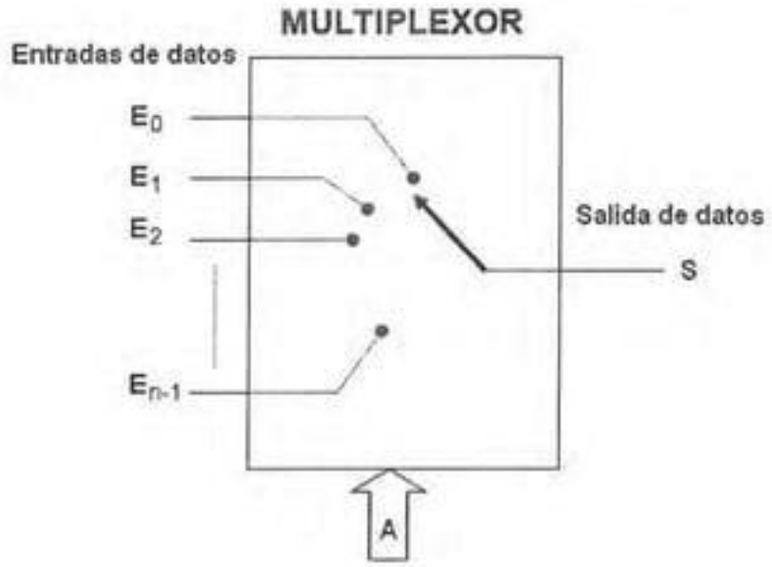
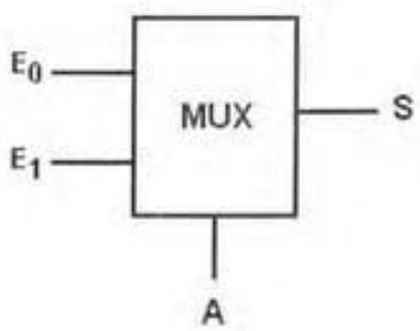


Fig. 2.7.
Esquema de un circuito combinacional multiplexor

Por tanto, se puede decir que un multiplexor es un conmutador de varias posici nes con diferentes entradas y una salida. Cuando se sitúa el selector en una de l entradas, solo esta aparece en la salida. Es un circuito lógico que permite envi en serie y de forma ordenada a través de una sola línea de salida, varias inform ciones digitales que pueden aparecer simultáneamente en distintas entradas.

Ejemplo 2.4

Realiza la tabla de la verdad y la implementación mediante puertas lógicas NAND de un multiplexor de 2 entradas.



Para realizar un multiplexor de 2 entradas (E_0, E_1), se necesita considerar una entrada de selección A. En la tabla de la verdad, se debe tener en cuenta que:

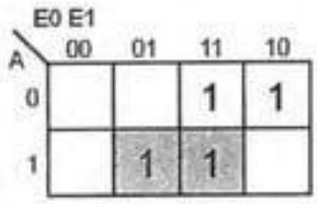
- Cuando la entrada de selección A vale 0, se considera el valor de la entrada E_0 en la salida S.
- Cuando la entrada de selección A vale 1, el valor de la entrada E_1 alcanza la salida S.

Tabla de la verdad:

A	E_0	E_1	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

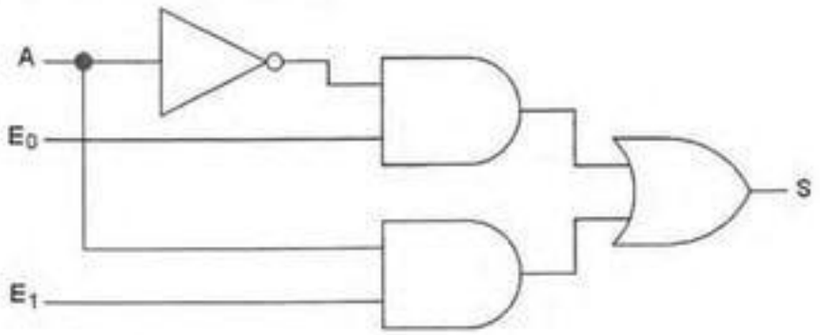
Función lógica inicial y simplificada por Karnaugh:

$$S = \bar{A} \cdot E_0 \cdot \overline{E_1} + \bar{A} \cdot E_0 \cdot E_1 + A \cdot \overline{E_0} \cdot E_1 + A \cdot E_0 \cdot E_1;$$

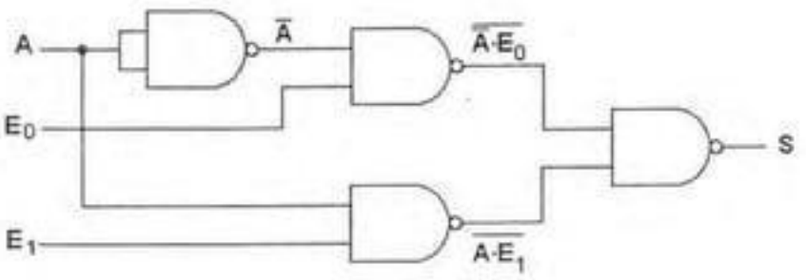


$$S = \bar{A} \cdot E_0 + A \cdot E_1;$$

Esquema de puertas lógicas (NOT, AND y OR):



Para implementar el circuito lógico anterior solo con puertas NAND, se debe cambiar la negación por una puerta NAND y las puertas AND por 2 puertas NAND en serie. Además, la suma se debe traducir por 3 puertas NAND en forma de triángulo. El resultado permite simplificar 4 puertas NAND (2 en serie en la que hay una negación negada).

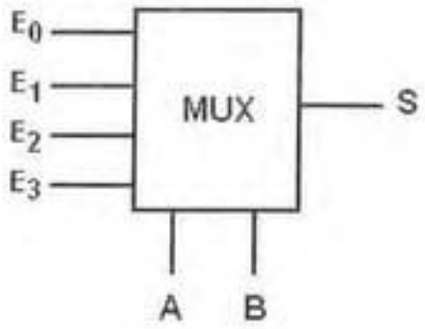


Recuerda • • •

Como bloques combinacionales o circuitos integrados combinacionales tenemos:

- Codificadores
- Decodificadores
- Multiplexores
- Demultiplexores

Para llevar a cabo el diseño de un **multiplexor de 4 entradas** (E_0, E_1, E_2, E_3) se cesitan 2 entradas de selección (A, B). Como en la salida S solo aparece la entr que haya sido seleccionada con las entradas de selección, la tabla de la ver (para 6 bits) solo debe reflejar esas combinaciones pudiendo indicar con ur aquellas entradas que son indiferentes.



A	B	E_0	E_1	E_2	E_3	S
0	0	0	X	X	X	0
0	0	1	X	X	X	1
0	1	X	0	X	X	0
0	1	X	1	X	X	1
1	0	X	X	0	X	0
1	0	X	X	1	X	1
1	1	X	X	X	0	0
1	1	X	X	X	1	1

Fig. 2.8. Tabla de la verdad para un multiplexor de 4 entradas

De esta forma, si los 2 bits de la entrada de selección AB son:

- 00, aparece el valor de la entrada E_0 en la salida S;
- 01, aparece el valor de la entrada E_1 en la salida S;
- 10, aparece el valor de la entrada E_2 en la salida S;
- 11, aparece el valor de la entrada E_3 en la salida S;

La función lógica de un multiplexor de 4 entradas está formada por 4 términos 3 variables y se requieren 7 puertas NAND para llevar a cabo su implementació

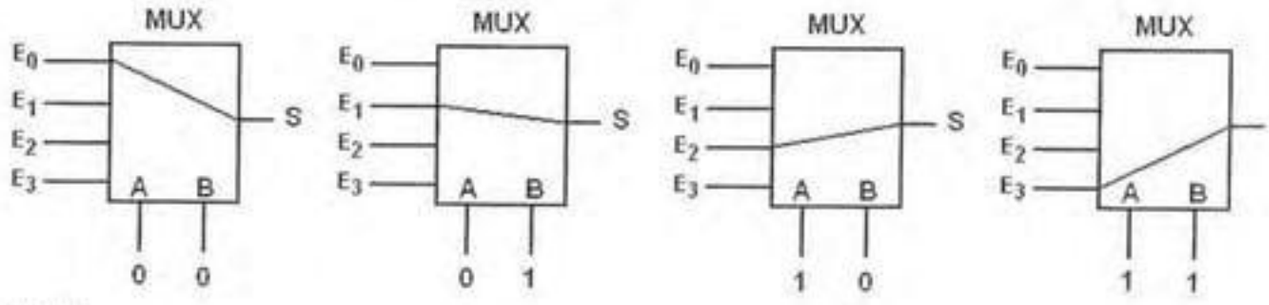


Fig. 2.9.
Conmutación de las entradas en un multiplexor

Recuerda • • •

Un multiplexor es un circuito combinacional que dispone de 2^n entradas de datos, n entradas de selección y una única salida de datos. Es posible interconectar varios entre sí.

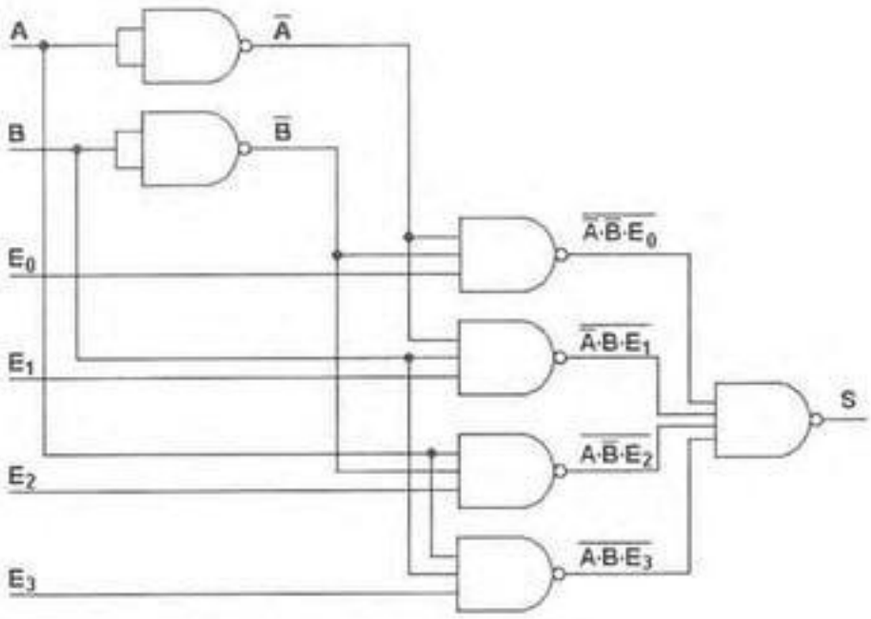


Fig. 2.10. Multiplexor de 4 entradas con puertas NAND

El circuito integrado 74HC/HCT151 de la familia lógica CMOS es un multiplexor de 8 entradas de datos (I_0 hasta I_7) y 3 líneas de entrada de selección de datos (S_0, S_1 y S_2). Además, dispone de un entrada de habilitación EN a modo de interruptor de encendido. El integrado 74HC151 tiene una salida de datos Y y su negada $\overline{Y}$, de forma que se pueda utilizar la que más convenga en función del diseño. Vcc debe ser +5 V.

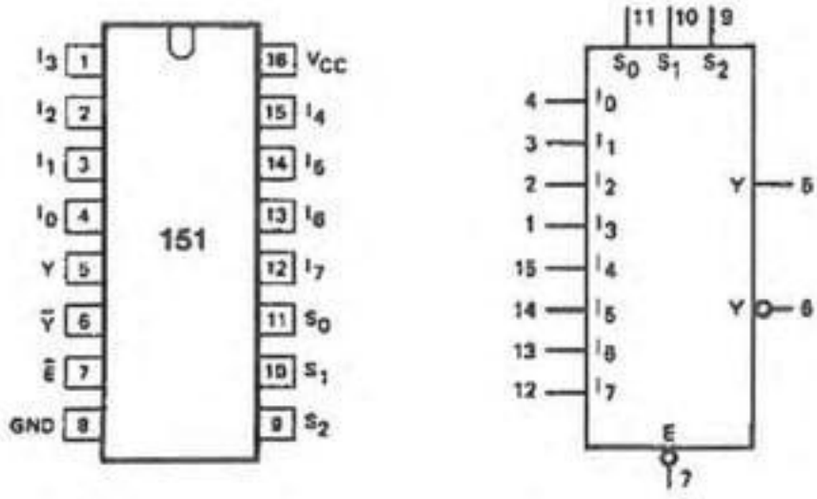


Fig. 2.11. Patillaje y símbolo lógico de un multiplexor 74HC151

INPUTS												OUTPUTS	
$\overline{E}$	S_2	S_1	S_0	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	$\overline{Y}$	Y
H	X	X	X	X	X	X	X	X	X	X	X	H	L
L	L	L	L	L	X	X	X	X	X	X	X	H	L
L	L	L	L	H	X	X	X	X	X	X	X	L	H
L	L	L	H	X	L	X	X	X	X	X	X	H	L
L	L	L	H	X	H	X	X	X	X	X	X	L	H
L	L	H	L	X	X	L	X	X	X	X	X	H	L
L	L	H	L	X	X	H	X	X	X	X	X	L	H
L	L	H	H	X	X	X	L	X	X	X	X	H	L
L	L	H	H	X	X	X	H	X	X	X	X	L	H
L	H	L	L	X	X	X	X	L	X	X	X	H	L
L	H	L	L	X	X	X	X	H	X	X	X	L	H
L	H	L	H	X	X	X	X	X	L	X	X	H	L
L	H	L	H	X	X	X	X	X	H	X	X	L	H
L	H	H	L	X	X	X	X	X	X	L	X	H	L
L	H	H	L	X	X	X	X	X	X	H	X	L	H
L	H	H	H	X	X	X	X	X	X	X	L	H	L
L	H	H	H	X	X	X	X	X	X	X	H	L	H

Fig. 2.12.
Tabla de la verdad del multiplexor 74HC151

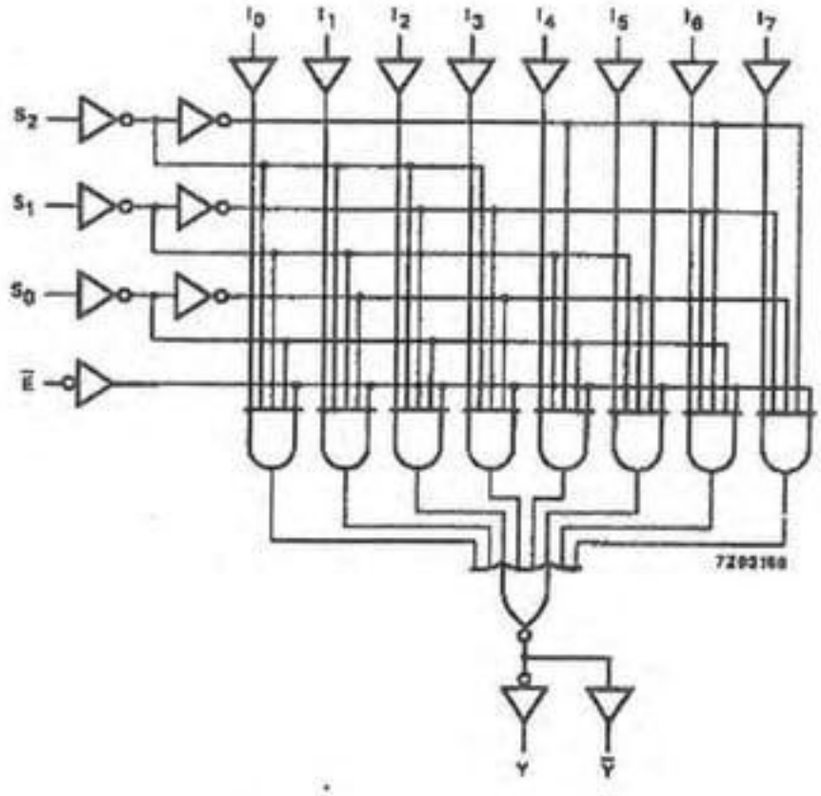


Fig. 2.13. Diagrama lógico del multiplexor 74HC151

Para aumentar la capacidad de un multiplexor, se pueden interconectar varios entre sí. Por ejemplo, se puede conseguir un multiplexor de 16 entradas de datos mediante 5 multiplexores de 4 entradas de datos, de forma que la salida de datos de cada uno de los 4 primeros sea entradas de datos en el último multiplexor. Los 4 circuitos iniciales deben tener 2 entradas de selección (A, B), mientras el último multiplexor tendrá 2 entradas de selección diferentes (C, D).

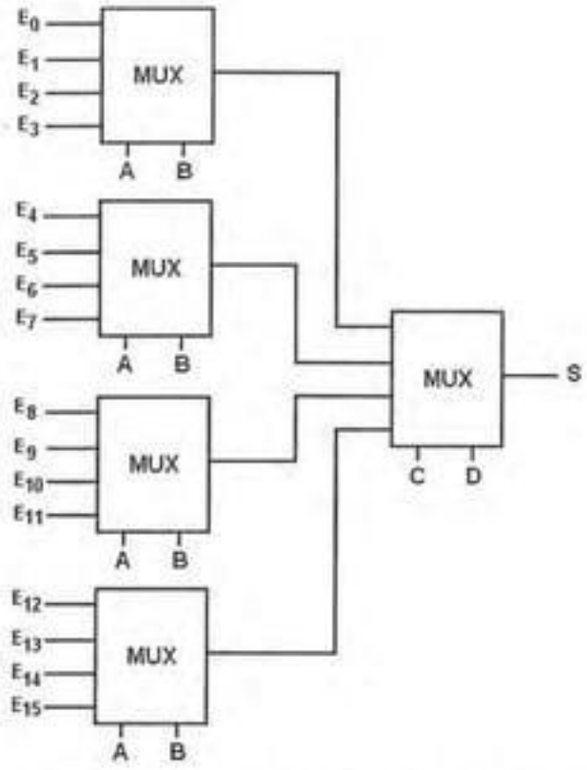


Fig. 2.14.
Multiplexor de 16 entradas de datos y 4 entradas de selección de datos

Además del CI 74151, se pueden encontrar en el mercado otros multiplexores en forma de integrados. Son el CI 74153 y el CI 74157.

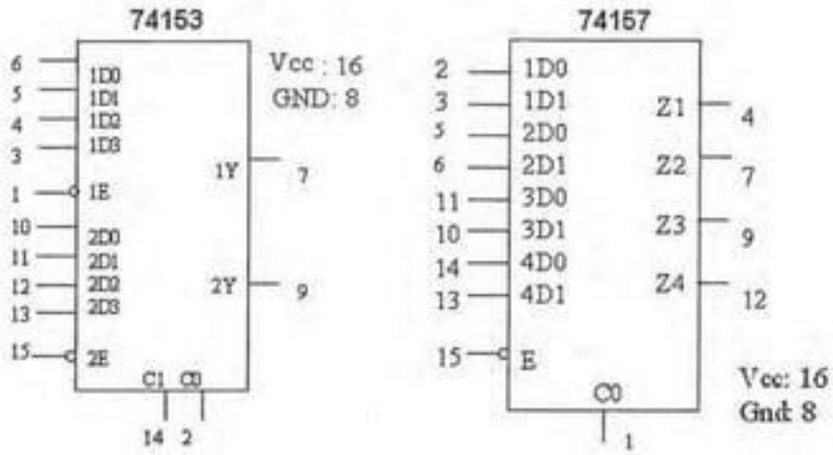


Fig. 2.15.
Otros multiplexores en forma de circuitos integrados

El multiplexor CI 74153 es un multiplexor doble con las entradas de selección C1 y C0 (LSB) comunes para ambos multiplexores. Las entradas de habilitación del Mux1 y Mux2 son 1E y 2E, respectivamente. Las salidas de los multiplexores son 1Y y 2Y. Las entradas del Mux1 son 1D0, 1D1, 1D2 y 1D3; las entradas del Mux2 son 2D0, 2D1, 2D2 y 2D3. El integrado CI 74157 contiene cuatro multiplexores con sus dos entradas de datos (1D0, 1D1; 2D0, 2D1; 3D0, 3D1; 4D0, 4D1) y su salida para cada uno (Z1, Z2, Z3 y Z4). Tiene una entrada de habilitación E y una entrada de selección C0 comunes a los cuatro multiplexores.

2.2.1. Generación de funciones lógicas mediante multiplexores

Mediante multiplexores es posible generar fácilmente una función lógica de un circuito combinacional, siempre que dicha función lógica esté expresada como suma de productos. Por ejemplo, mediante el multiplexor de 8 entradas CI 74HC151, se puede implementar cualquier función lógica de 3 variables. El proceso requiere conectar las variables a las entradas de selección y conectar a cada entrada de datos el nivel lógico (0, 1) que nos indique la tabla de la verdad de la función lógica.

Ejemplo 2.5

Implementa, mediante un multiplexor de 8 entradas, el circuito combinacional de activación de una alarma (salida S). Se dispone de 3 sensores (A, B y C) que controlan diferentes zonas de una vivienda. La activación de la alarma se produce cuando:

- El sensor A está activado, el B desactivado y el C en cualquier posición.
- El sensor C está activado y los sensores A y B están desactivados.
- Todos los sensores están activados.

La tabla de la verdad del circuito propuesto nos indica que las entradas E_0, E_2, E_3 y E_6 están al nivel lógico 0. Esto se puede implementar llevando dichas entradas de datos en el multiplexor a masa. Por otro lado, la tabla de la verdad nos indica que las entradas E_1, E_4, E_5 y E_7 están al nivel lógico 1. Esto se puede realizar conectando dichas entradas de datos en el multiplexor a la tensión de alimentación Vcc.

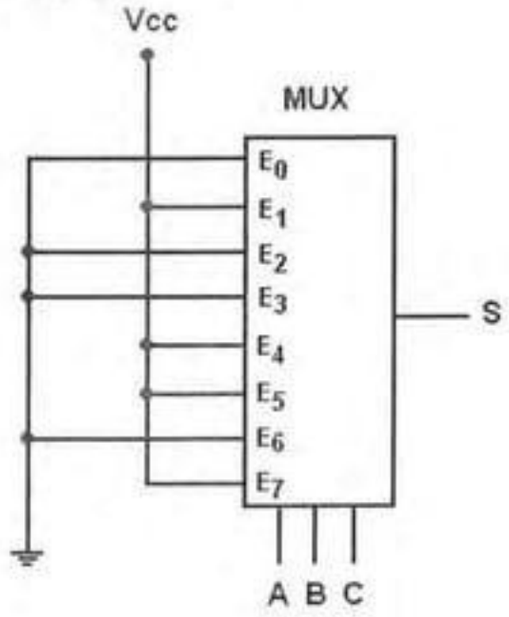
Tabla de la verdad:

A	B	C	S	
0	0	0	0	E_0
0	**0**	**1**	**1**	**E_1**
0	1	0	0	E_2
0	1	1	0	E_3
1	**0**	**0**	**1**	**E_4**
1	**0**	**1**	**1**	**E_5**
1	1	0	0	E_6
1	**1**	**1**	**1**	**E_7**

Función lógica en forma canónica (suma de productos):

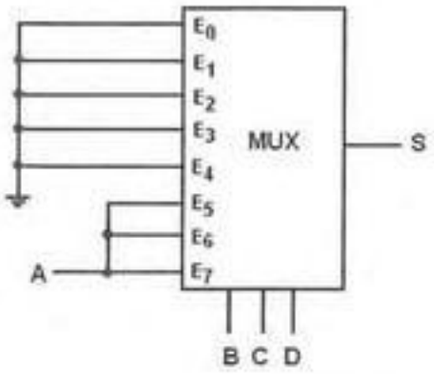
$$S = \bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot B \cdot C;$$

Diagrama lógico del multiplexor de 8 entradas:



Ejemplo 2.6

Se pide obtener la función lógica canónica que representa el circuito lógico dado por un multiplexor. Se deben considerar 4 variables en la función lógica (A, B, C y D).

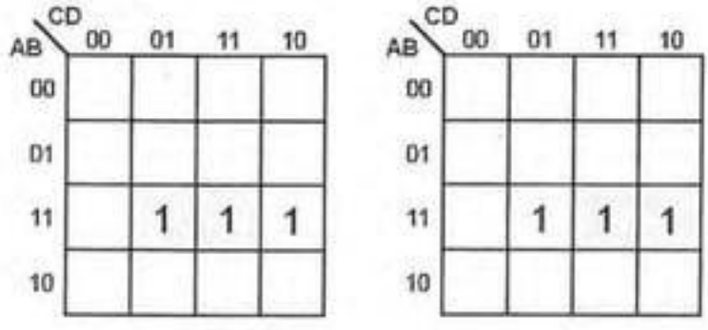


La tabla de la verdad del circuito propuesto debe tener 4 columnas para las entradas de selección (variables A, B, C y D), 2 columnas para las entradas de datos y una columna más para la salida de datos.

A	B	C	D			S
X	0	0	0	E_0	0	0
X	0	0	1	E_1	0	0
X	0	1	0	E_2	0	0
X	0	1	1	E_3	0	0
X	1	0	0	E_4	0	0
0	1	0	1	E_5	A	0
1	**1**	**0**	**1**	E_5	**A**	**1**
0	1	1	0	E_6	A	0
1	**1**	**1**	**0**	E_6	**A**	**1**
0	1	1	1	E_7	A	0
1	**1**	**1**	**1**	E_7	**A**	**1**

Función lógica inicial y simplificada por Karnaugh:

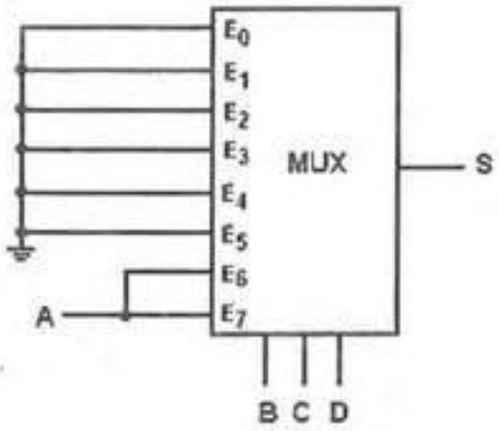
$$S = A \cdot B \cdot \bar{C} \cdot D + A \cdot B \cdot C \cdot \bar{D} + A \cdot B \cdot C \cdot D;$$



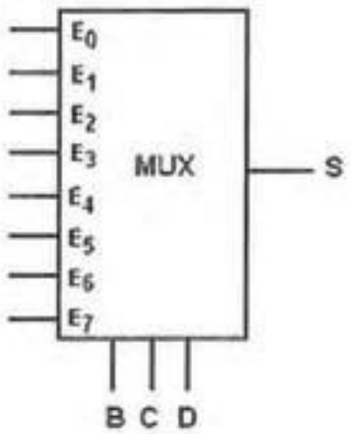
$$S = A \cdot B \cdot D + A \cdot B \cdot C;$$

Actividades propuestas

1. Obtén la función lógica del circuito de la figura.



2. Dibuja la tabla de la verdad para el funcionamiento del multiplexor de la figura.



2.3. Demultiplexores

Un demultiplexor es un circuito que realiza la operación contraria al multiplexo
Posee una única entrada de datos E y 2^n salidas de datos (S_0, S_1, S_2,...) con **n** entr
das de selección.

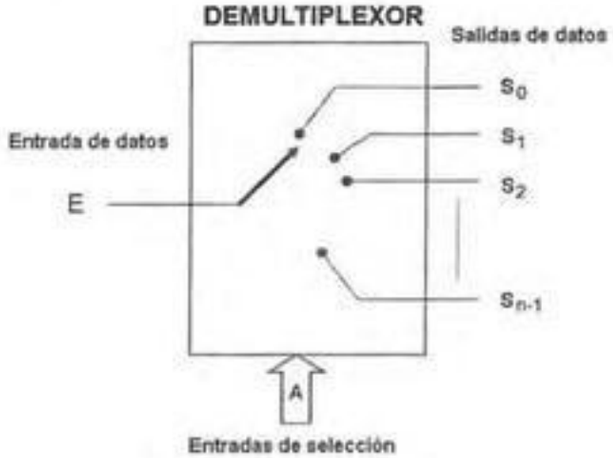


Fig. 2.16.
Esquema de un circuito combinacional demultiplexor

Recuerda • • •

El demultiplexor funciona a nivel lógico de forma inversa al multiplexor.

El circuito integrado 74HC/HCT138 es un demultiplexor de 8 salidas de datos (Y_0 hasta Y_7) y 3 líneas de entrada de selección de datos (A_0, A_1 y A_2). Los datos de la única entrada E_3 saldrán por las salidas negadas (Y_0 hasta Y_7) en función de la combinación de bits aplicados a las entradas de selección (A_0, A_1 y A_2). Este CI tiene 2 entradas de habilitación negadas E_1 y E_2.

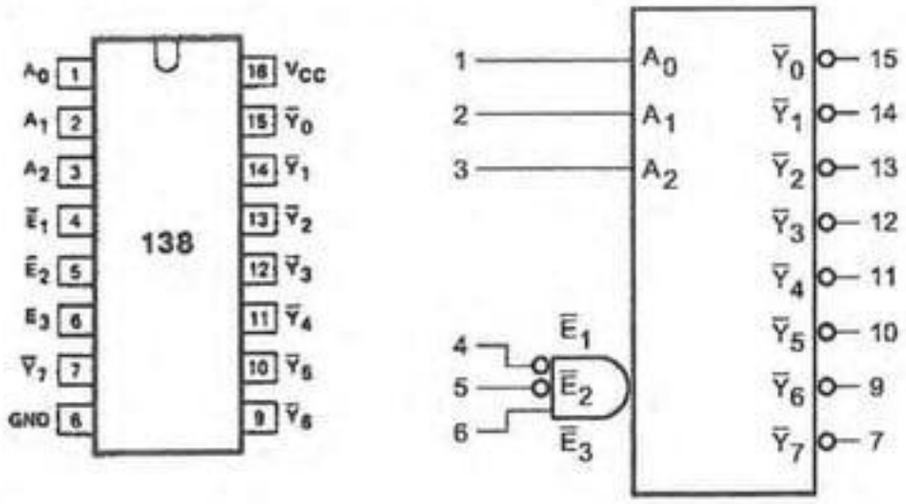


Fig. 2.17. Patillaje y símbolo lógico de un demultiplexor 74HC138

2.4. Decodificadores

Un decodificador es un circuito combinacional que permite traducir una información codificada en lenguaje binario (**código de entrada de N bits**) a otro tipo de código como el decimal. Es decir, por cada una de las combinaciones de entradas de datos binarios (E_0, E_1, E_2...) se tiene solo una salida de datos decimal (S_0, S_1...) distinta activada (valor lógico 1) cada vez.

Fig. 2.18. Esquema de un circuito combinacional decodificador

Existen 2 tipos de decodificadores, excitadores y no excitadores, en función de si las salidas pueden o no poner en funcionamiento un indicador numérico como un *display* de 7 segmentos.

El ejemplo típico es un decodificador BCD a decimal con 4 entradas binarias (E_0, E_1, E_2, E_3) y 10 salidas decimales (números del 0 al 9). Al aplicar a las entradas un dato binario en BCD, solo una de las salidas toma el valor lógico 1 y se le asigna el número decimal correspondiente. La tabla de la verdad nos indica el número decimal devuelto en cada caso y la salida activa alta. Cuando la combinación binaria de entradas no corresponde a un número decimal, se ponen a cero todas las salidas (nivel bajo).

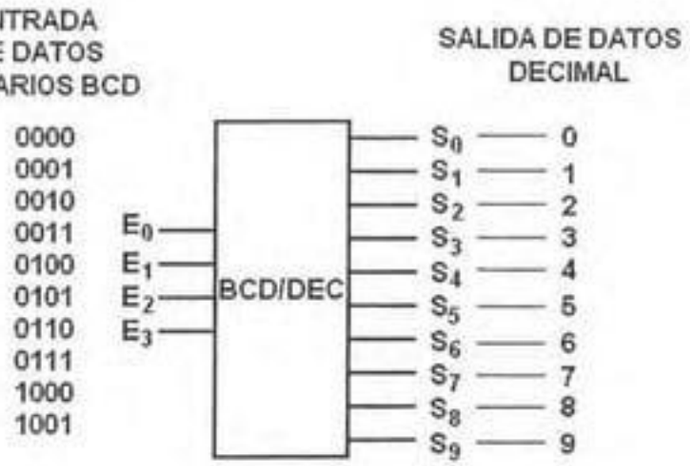


Fig. 2.19. Esquema de un decodificador BCD a decimal

DEC	E_3	E_2	E_1	E_0	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	**1**	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	**1**	0	0	0	0	0	0	0
3	0	0	1	1	0	0	0	**1**	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	**1**	0	0	0	0	0
5	0	1	0	1	0	0	0	0	0	**1**	0	0	0	0
6	0	1	1	0	0	0	0	0	0	0	**1**	0	0	0
7	0	1	1	1	0	0	0	0	0	0	0	**1**	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	**1**	0
9	1	0	0	1	0	0	0	0	0	0	0	0	0	**1**

Fig. 2.20. Tabla de la verdad para un decodificador BCD a decimal

Ejemplo 2.7

Realiza la tabla de la verdad y la implementación mediante puertas lógicas de un decodificador BCD a decimal de 2 entradas y 4 salidas.

Tabla de la verdad:

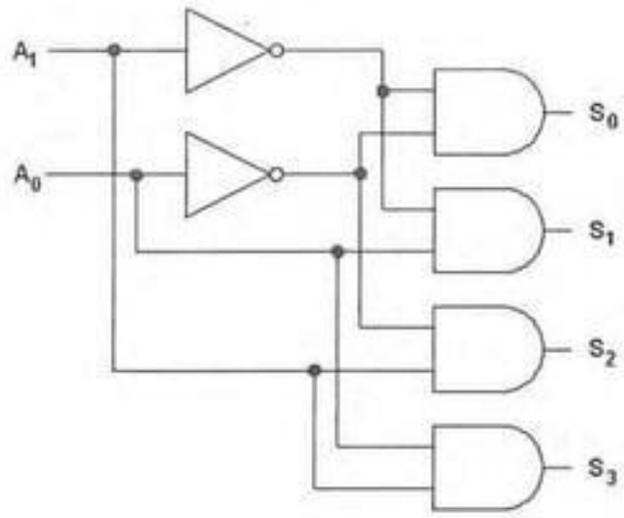
DEC	A_1	A_0	S_0	S_1	S_2	S_3
0	0	0	1	0	0	0
1	0	1	0	1	0	0
2	1	0	0	0	1	0
3	1	1	0	0	0	1

Las funciones lógicas, para cada salida de datos, se obtienen directamente de la tabla de la verdad.

$$S_0 = \overline{A_0} \cdot \overline{A_1}\,; \qquad S_1 = A_0 \cdot \overline{A_1}\,;$$

$$S_2 = \overline{A_0} \cdot A_1\,; \qquad S_3 = A_0 \cdot A_1\,;$$

Esquema de puertas lógicas (NOT y OR):



El circuito integrado 74HC/HCT42 de la familia lógica CMOS es un decodificador de 4 entradas binarias (A_0, A_1, A_2 y A_3) y 10 salidas activas a nivel bajo (desde Y_0 hasta Y_9). Esto significa que las salidas toman el valor lógico 0 en vez de 1 en la función de decodificación.

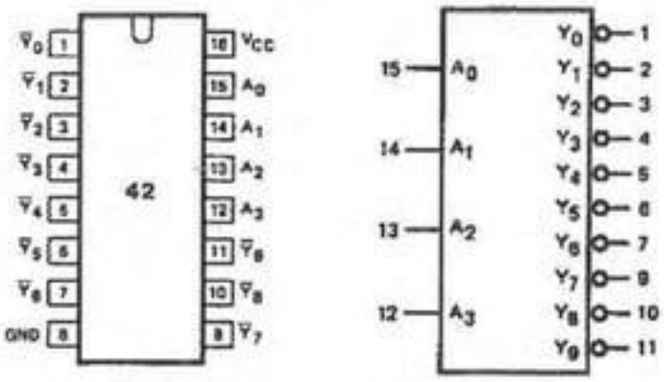


Fig. 2.21. Patillaje y símbolo lógico de un codificador 74HC42

INPUTS				OUTPUTS									
A_3	A_2	A_1	A_0	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$	$\overline{Y}_8$	$\overline{Y}_9$
L	L	L	L	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	H	H	L	H	H	H	H	H	H	H
L	L	H	H	H	H	H	L	H	H	H	H	H	H
L	H	L	L	H	H	H	H	L	H	H	H	H	H
L	H	L	H	H	H	H	H	H	L	H	H	H	H
L	H	H	L	H	H	H	H	H	H	L	H	H	H
L	H	H	H	H	H	H	H	H	H	H	L	H	H
H	L	L	L	H	H	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L
H	L	H	L	H	H	H	H	H	H	H	H	H	H
H	L	H	H	H	H	H	H	H	H	H	H	H	H
H	H	L	L	H	H	H	H	H	H	H	H	H	H
H	H	L	H	H	H	H	H	H	H	H	H	H	H
H	H	H	L	H	H	H	H	H	H	H	H	H	H
H	H	H	H	H	H	H	H	H	H	H	H	H	H

H = Nivel lógico alto (1); L = Nivel lógico bajo (0);

Fig. 2.22. Tabla de la verdad del decodificador 74HC42

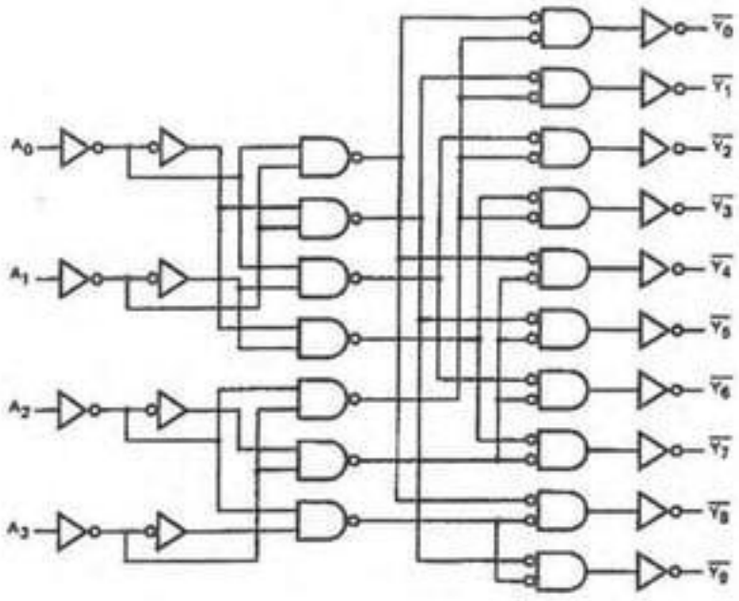


Fig. 2.23. Diagrama lógico del decodificador 74HC42

Ejemplo 2.8

Se pide obtener la función lógica canónica que representa el circuito lógico dado por un decodificador. Se deben considerar 3 variables en la función lógica (A, B y C).

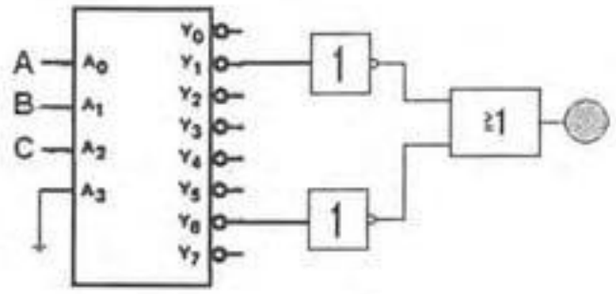


Tabla de la verdad:

DEC	C	B	A	
0	0	0	0	S_0
1	0	0	1	S_1
2	0	1	0	S_2
3	0	1	1	S_3
4	1	0	0	S_4
5	1	0	1	S_5
6	1	1	0	S_6
7	1	1	1	S_7

La función lógica se obtiene directamente de la tabla de la verdad:

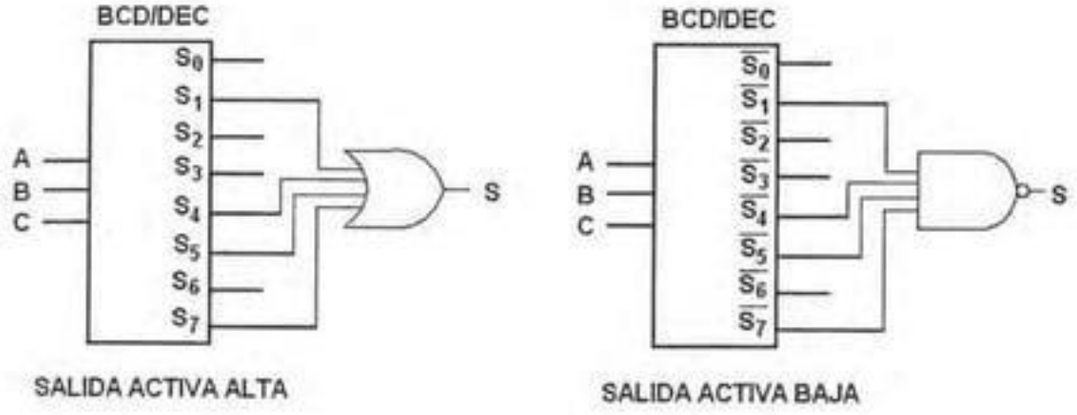
$$F = \bar{C} \cdot \bar{B} \cdot A + C \cdot B \cdot \bar{A};$$

2.4.1. Generación de funciones lógicas mediante decodificadores

Mediante decodificadores es posible generar fácilmente una función lógica de un circuito combinacional. Partiendo de la tabla de la verdad y para un decodificador con salida activa alta, basta conectar una puerta suma OR a las salidas de datos decimal en aquellas combinaciones que tienen el estado lógico 1. Para un decodificador de salida activa baja, el procedimiento es el mismo pero conectando una puerta NAND.

Ejemplo 2.9

Implementa, mediante un decodificador de 3 entradas y 8 salidas, el circuito combinacional de activación de una alarma (salida S) del ejemplo 2.5. Considera las 2 opciones posibles de decodificadores (primero con salida activa alta y, después, con otro con salida activa baja).



El circuito integrado 74HC/HCT154 es un decodificador de 4 entradas binarias (A_0, A_1, A_2 y A_3) y 16 salidas activas a nivel bajo (desde Y_0 hasta Y_{15}). Dispone de 2 entradas de habilitación, E_0 y E_1, activas a nivel bajo."

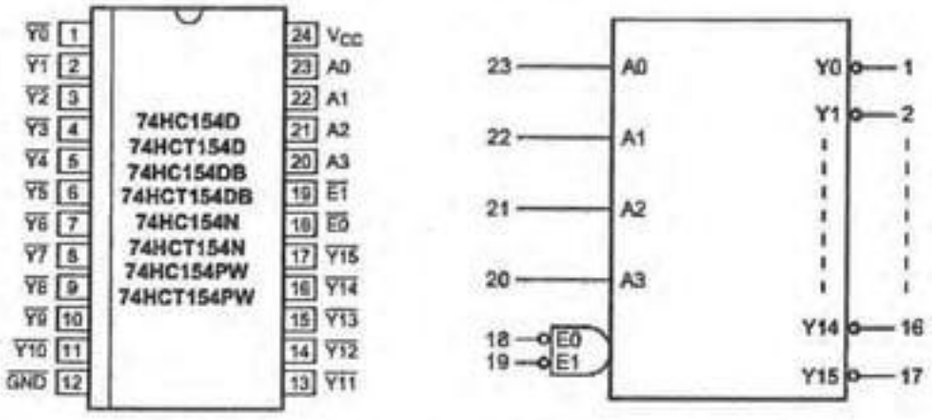


Fig. 2.24. Patillaje y símbolo lógico de un codificador 74HC154

Input						Output															
$\overline{E0}$	$\overline{E1}$	A0	A1	A2	A3	$\overline{Y0}$	$\overline{Y1}$	$\overline{Y2}$	$\overline{Y3}$	$\overline{Y4}$	$\overline{Y5}$	$\overline{Y6}$	$\overline{Y7}$	$\overline{Y8}$	$\overline{Y9}$	$\overline{Y10}$	$\overline{Y11}$	$\overline{Y12}$	$\overline{Y13}$	$\overline{Y14}$	$\overline{Y}$
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
		H	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
		L	H	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
		H	H	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
		L	L	H	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
		H	L	H	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
		L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
		H	H	H	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
		L	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
		H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
		L	H	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
		H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
		L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
		H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
		L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
		H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L

H = Nivel lógico alto (1); L = Nivel lógico bajo (0); X = Irrelevante;

Fig. 2.25. Tabla de la verdad del decodificador 74HC154

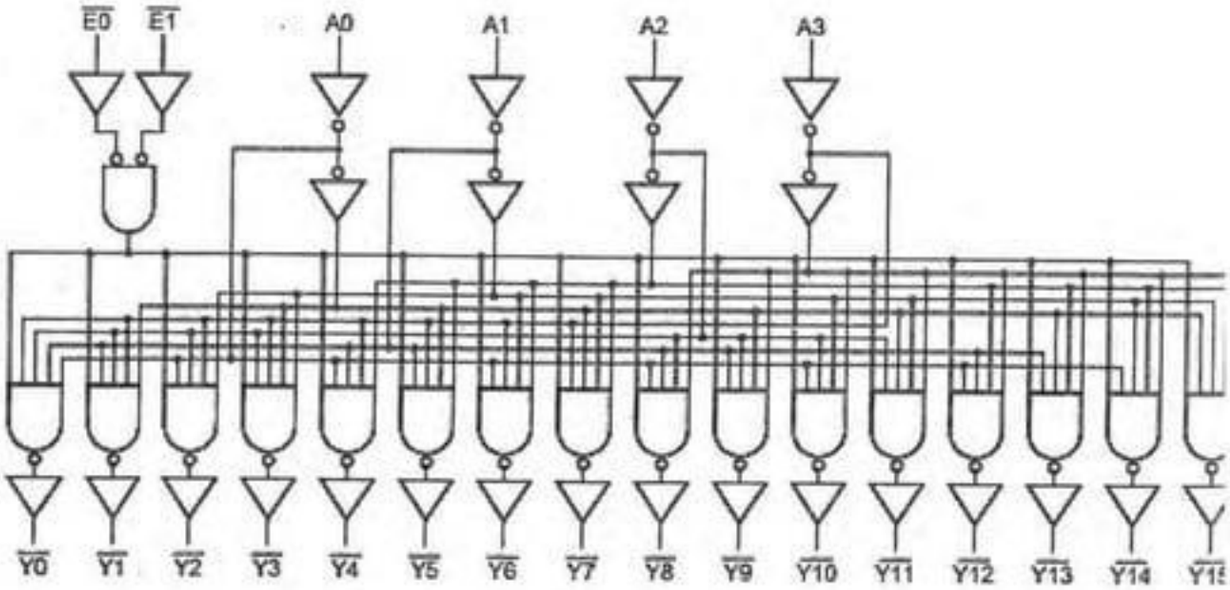


Fig. 2.26. Diagrama lógico del decodificador 74HC154

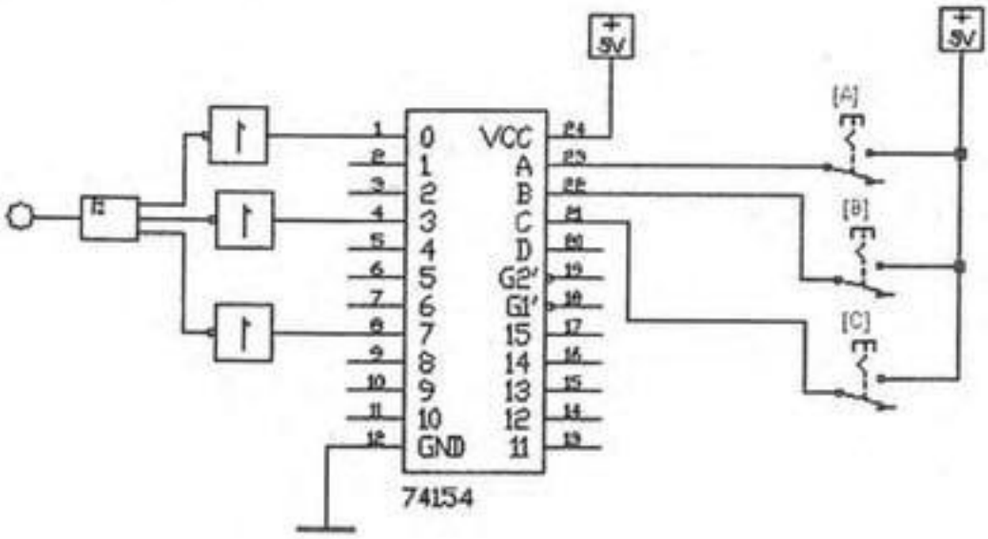
Ejemplo 2.10

Diseña un sistema combinacional que sea capaz de detectar cuándo a sus entradas se le introduce una combinación en binario que exprese una cantidad que sea en decimal 0, 3 o 7.

Tabla de la verdad:

DEC	C	B	A	
0	0	0	0	S_0
1	0	0	1	S_1
2	0	1	0	S_2
3	0	1	1	S_3
4	1	0	0	S_4
5	1	0	1	S_5
6	1	1	0	S_6
7	1	1	1	S_7

Utilizando un decodificador CI 74154, con 3 entradas de datos binarios (C, B y A) y considerando las salidas decimales necesarias (S_0, S_3 y S_7), se tiene que:



Ejemplo 2.11

Desde dos puntos de accionamiento I_1 e I_2 (de número de contactos indeterminado) se desea controlar 5 lámparas L_1, L_2, L_3, L_4, L_5 con la siguiente secuencia:

- Si se acciona solo el punto I_1 se encienden las lámparas L_1, L_2 y L_3.
- Si se acciona solo el punto I_2 se encienden las lámparas L_4, L_2 y L_5.
- Si se acciona a la vez I_1 e I_2 se enciende solo la lámpara L_2.

Tabla de la verdad:

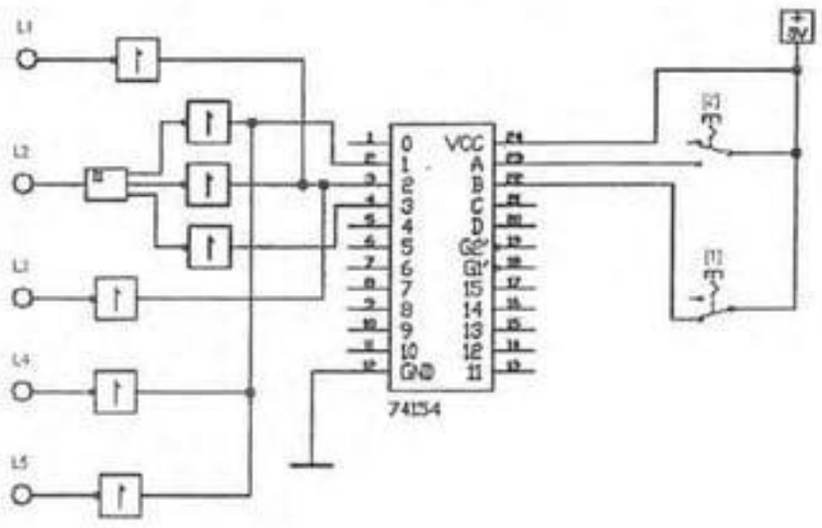
DEC	I_1(B)	I_2(A)		L_1	L_2	L_3	L_4	L_5
0	0	0	S_0	0	0	0	0	0
1	0	1	S_1	0	1	0	1	1
2	1	0	S_2	1	1	1	0	0
3	1	1	S_3	0	1	0	0	0

Analizando la tabla de la verdad se tiene que:

- Si se activa solo el punto de accionamiento I_1, la entrada de datos binario al decodificador es 10. Por tanto, se debe activar a nivel bajo solo la salida S_2.
- Si se activa solo el punto de accionamiento I_2, la entrada de datos binario al decodificador es 01. Por tanto, se debe activar a nivel bajo solo la salida S_1.
- Si se activa tanto I_1 como I_2, la entrada de datos binario al decodificador es 11. Por tanto, se debe activar a nivel bajo solo la salida S_3.

Utilizando un decodificador CI 74154, con 2 entradas de datos binarios (I_1 e I_2) y considerando las salidas decimales necesarias (S_1, S_2 y S_3), se tiene que:

- Para que se encienda la lámpara L_1 se necesita tener activada a nivel bajo la salida S_2 del decodificador.
- Para que se encienda la lámpara L_2 se necesita tener activada a nivel bajo las salidas S_1, S_2 y S_3 del decodificador.
- Para que se encienda la lámpara L_3 se necesita tener activada a nivel bajo la salida S_2 del decodificador.
- Para que se encienda la lámpara L_4 se necesita tener activada a nivel bajo la salida S_1 del decodificador.
- Para que se encienda la lámpara L_5 se necesita tener activada a nivel bajo la salida S_1 del decodificador.

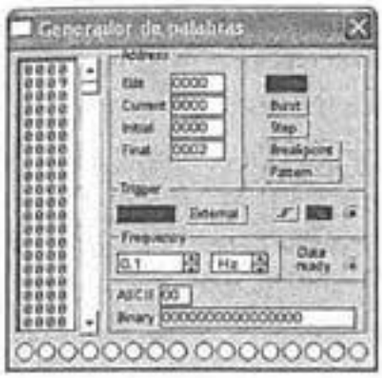


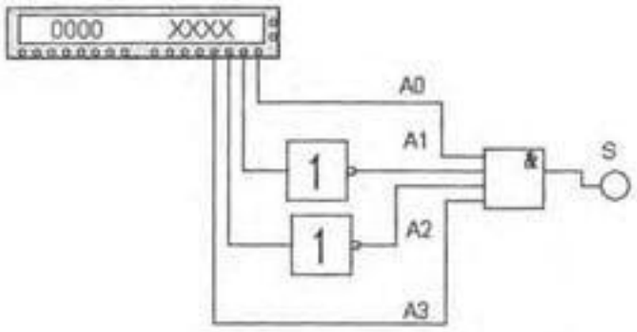
Actividades propuestas

1. Diseña un decodificador con puertas lógicas de cuatro entradas y una salida para que se cumplan los valores de la tabla. Monta el circuito en Workbench con un generador de palabras para las entradas.

Entradas				Salida
A0	A1	A2	A3	S
1	0	0	1	1

Resultado:

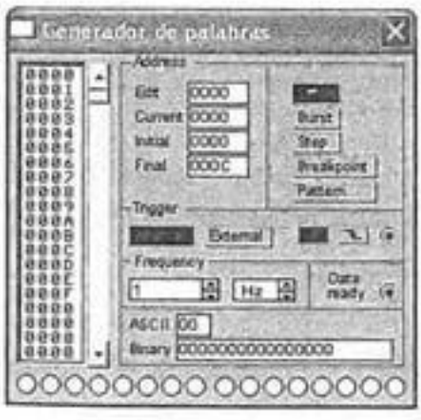


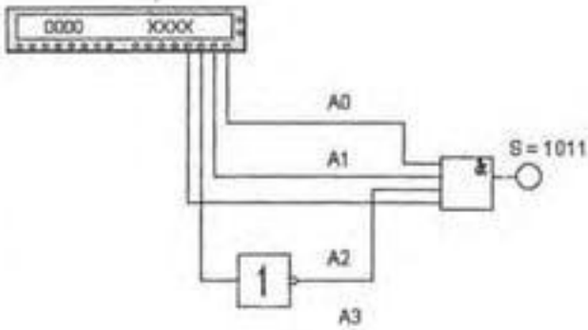


2. Diseña un decodificador con puertas lógicas de cuatro entradas y una salida para que se cumplan los valores de la tabla. Monta el circuito en Workbench con un generador de palabras para las entradas.

Entradas				Salida
A0	A1	A2	A3	S
1	0	1	1	1

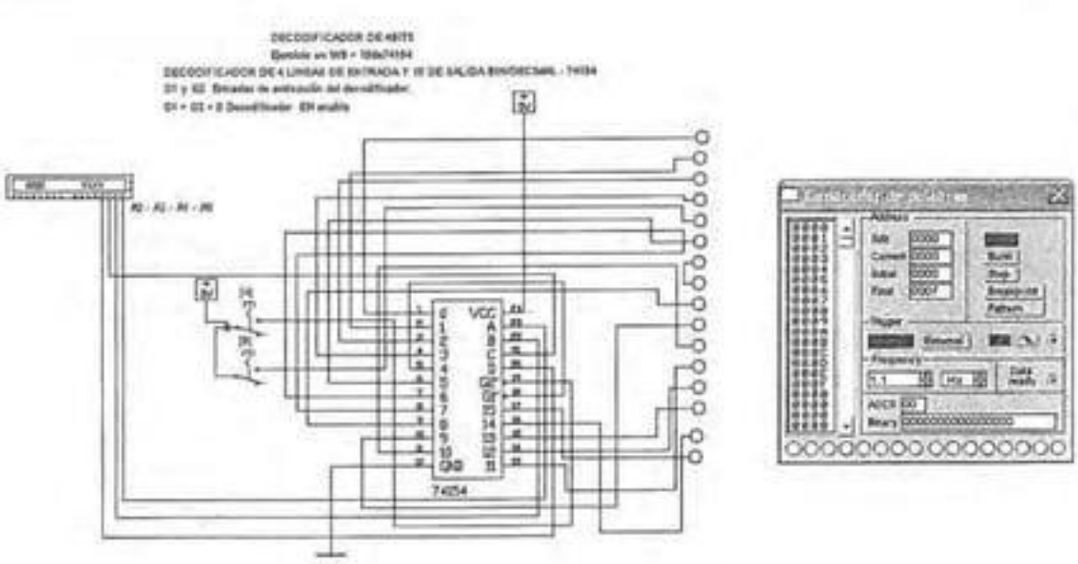
Resultado:





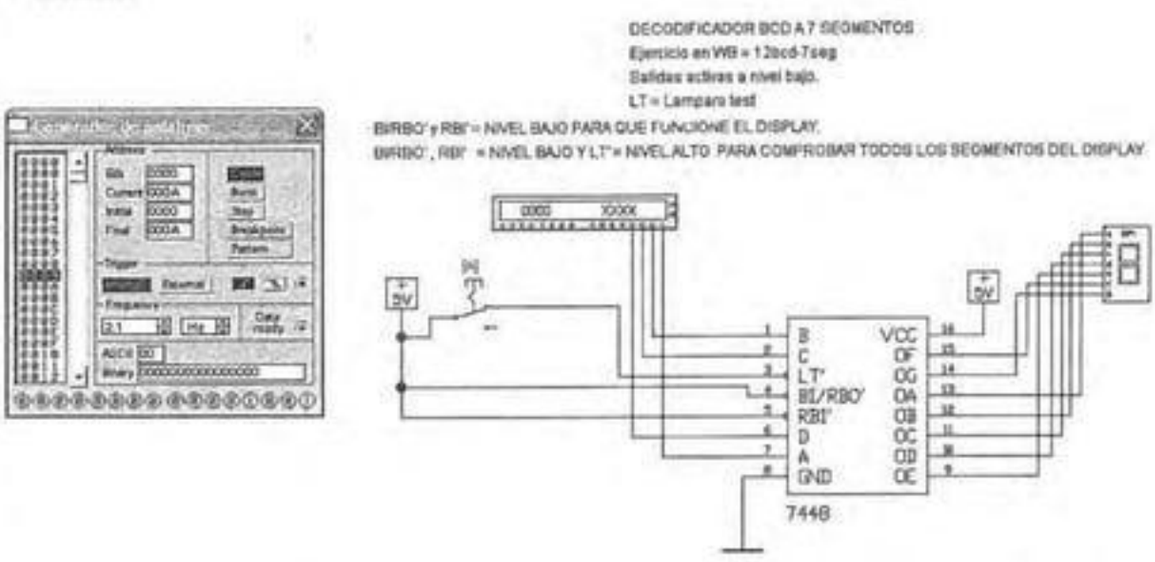
3. Monta en Workbench el decodificador de cuatro líneas de entrada y dieciséis de salida 74154, con un generador de palabras para las entradas y dieciséis pilotos para las salidas. Prueba su funcionamiento.

Resultado:



4. Monta en Workbench un decodificador BCD a 7 segmentos que cuente hasta nueve.

Resultado:



5. Monta en Workbench un circuito con dos decodificadores BCD a 7 segmentos que cuente hasta 99.

Resultado:

6. Se desea gobernar un motor eléctrico por medio de tres interruptores A, B y C. Se pondrá el motor en marcha en los siguientes casos:

   a) Cuando se accione solamente el interruptor A.

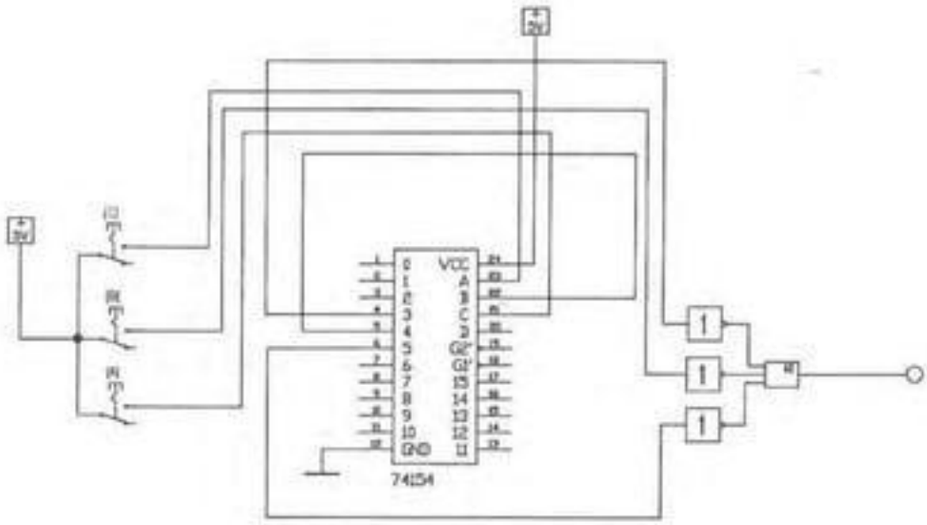
   b) Cuando se accionen a la vez los interruptores A y C.

   c) Cuando se accionen a la vez los interruptores B y C.

   Se pide:

   - Planteamiento del automatismo e implementación con un decodificador.
   - Montaje en Workbench y prueba de su funcionamiento.

   Resultado:



7. Se desean gobernar dos motores M1 y M2 por medio de tres interruptores A, B y C, según el siguiente programa:

   a. Si se acciona solamente el interruptor A, funciona solamente el motor M1.

   b. Si se acciona solamente el interruptor B, funcionan los motores M1 y M2.

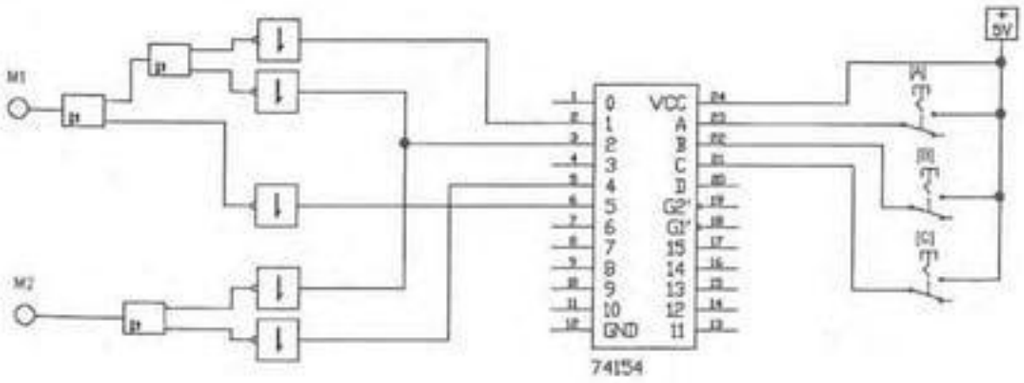
   c. Si se acciona solamente el interruptor C, funciona el motor M2.

   d. Si se accionan a la vez los interruptores A y C, funciona el motor M1.

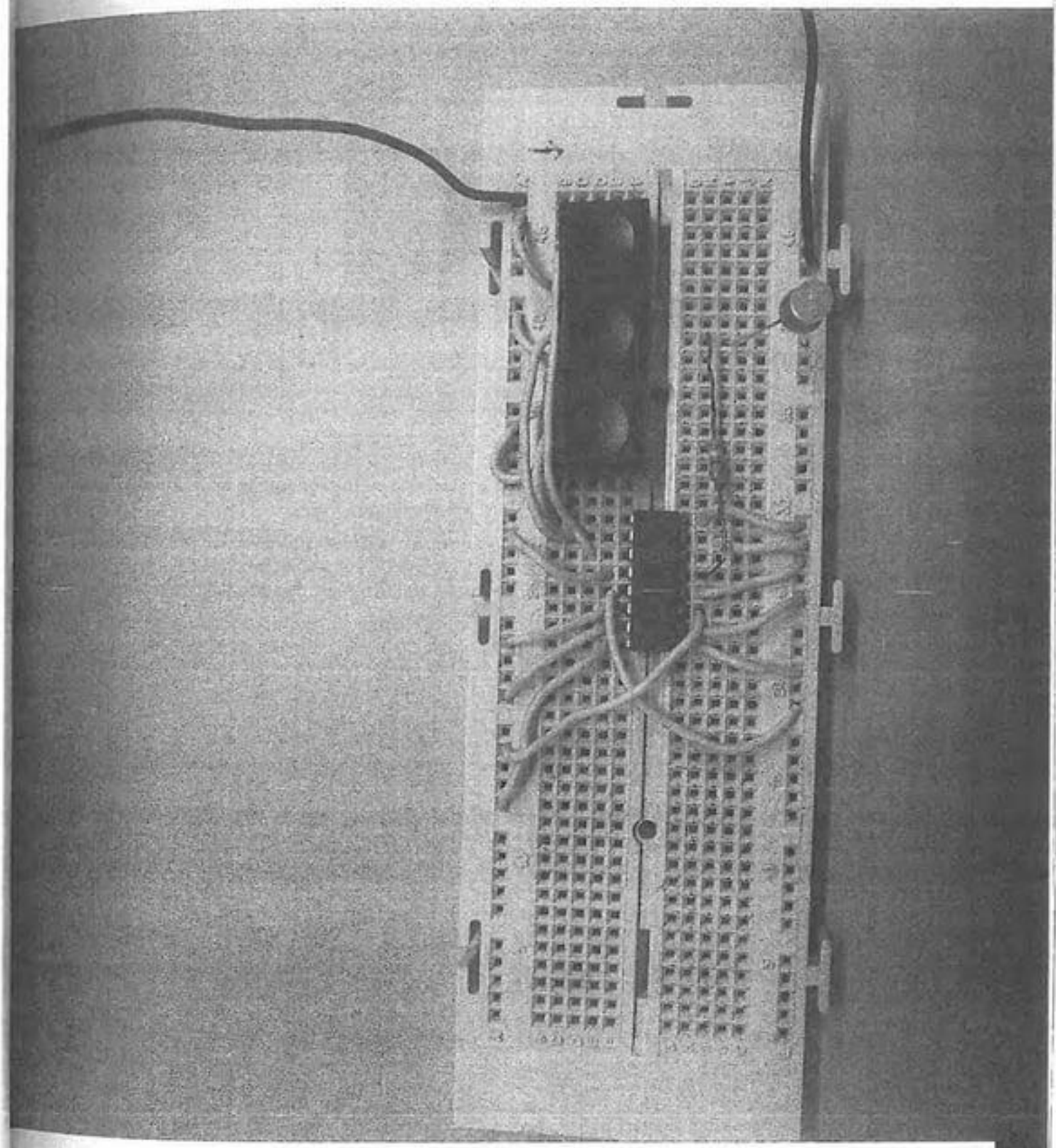
   Se pide:

   - Planteamiento del automatismo e implementación con un decodificador.
   - Montaje en Workbench y prueba de su funcionamiento.

   Resultado:



Unidad 3 Sistemas secuenciales con puertas lógicas



En este capítulo:

3.1. Definición y tipos

3.2. Básculas o biestables

3.3. Registros de desplazamiento

3.4. Contadores

3.1. Definición y tipos

Un **sistema secuencial** es aquel en el que, en un determinado instante, las sa das no dependen solo del estado de las variables de entradas, sino que, adem dependen de los estados anteriores que haya adquirido el sistema. Por tanto, requiere una **memoria** en la que el sistema secuencial guarde el estado de l salidas anteriores y las convierta en un estado interno del mismo sistema secue cial. Un sistema secuencial posee 2^n estados para n entradas ($X_1, X_2, ..., X_n$). Posee además, 2^p estados para p salidas ($Z_1, ..., Z_{p-m}, ..., Z_p$) y un número finito de estad internos ($y_1, ..., y_m$).

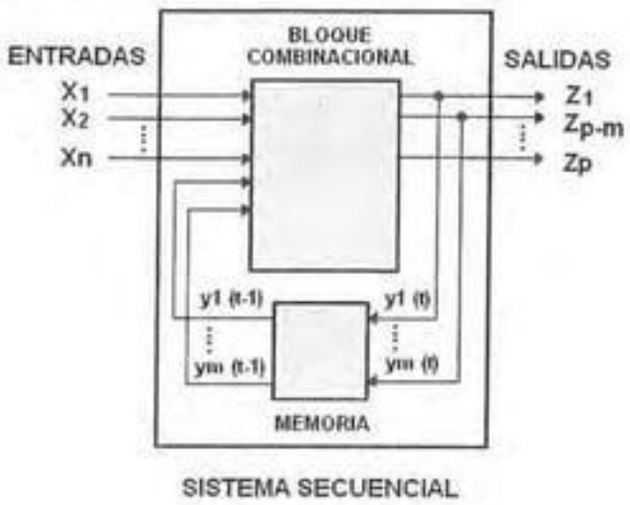


Fig. 3.1.
Esquema de un sistema secuencial

Según la forma de realizar el elemento de memoria hay 2 tipos de sistemas se cuenciales:

- Sistemas Secuenciales Síncronos: Se requiere la sincronización de los elemento del sistema mediante una señal de reloj (tren de pulsos periódico). Las variable internas no se modifican hasta que no llega un pulso del reloj.
- Sistemas Secuenciales Asíncronos: Actúan de forma continua en el tiempo, d modo que un cambio en las entradas provoca cambios en las variables interna sin esperar a la intervención de un reloj.

3.2. Básculas o biestables

Como ejemplo inicial de un sistema secuencial se considera un biestable. Este cir cuito está formado por puertas lógicas y tiene la realimentación de la salida com una entrada más. Posee 2 estados posibles (biestable), uno para el encendido d una lámpara y otro para el apagado.

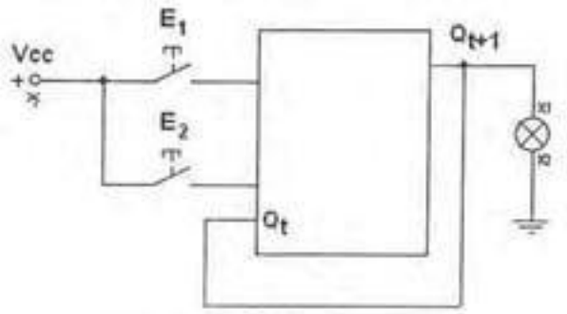


Fig. 3.2.
Diagrama de bloques para un biestable

La tabla de la verdad del circuito biestable propuesto, para 3 variables de entrada (E_1, E_2 y Q_t) y una de salida (Q_{t+1}) es:

Recuerda • • •

Un sistema secuencial es un sistema combinacional en el que, además, interviene el tiempo y el estado de las salidas que interactúan como entradas para procesar la información y dar resultados en las salidas.

E_1	E_2	Q_t	Q_{t+1}	Estado de la salida
0	0	0	0	Luz apagada
1	0	0	1	Se pulsa E_1 y se enciende la luz
0	0	1	1	Aunque no se pulse E_1, la luz sigue encendida
1	0	1	1	Si se vuelve a pulsar E_1, la luz sigue encendida
0	1	1	0	Se pulsa E_2 y se apaga la luz
0	1	0	0	Si se vuelve a pulsar E_2, la luz sigue apagada

Fig. 3.3. Tabla de la verdad para un biestable

donde Q_t es la salida en el momento de accionar uno de los pulsadores y Q_{t+1} es la salida un tiempo después de accionar uno de los pulsadores. La función lógica S, simplificada por Karnaugh, del circuito biestable es:

$$S = Q_{t+1} = E_1 + \overline{E_2} \cdot Q_t;$$

En la práctica, el circuito interno del biestable propuesto se implementa con puertas NOT y NAND o con puertas NOR.

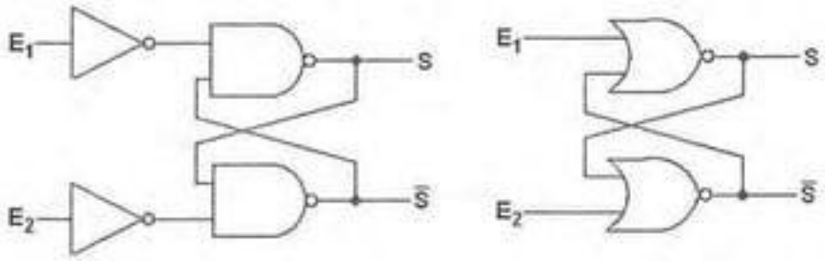


Fig. 3.4. Circuito interno de un biestable

Los circuitos biestables de 2 entradas más utilizados son el asíncrono R-S, el síncrono R-S activado por nivel, el síncrono R-S activado por flancos de reloj, el asíncrono J-K, el síncrono J-K y el J-K maestro-esclavo. Los circuitos biestables de 1 entrada más utilizados son el síncrono D activado por flanco y el síncrono T. Las aplicaciones más habituales de los biestables son los registros de desplazamiento, los contadores y las memorias.

3.2.1. Biestable R-S asíncrono

Es un circuito secuencial que posee 2 entradas (R-reset para borrar y S-set para grabar) y 2 salidas (Q y $\overline{Q}$) que son complementarias (cuando una tiene un valor lógico 1, la otra debe estar a 0).

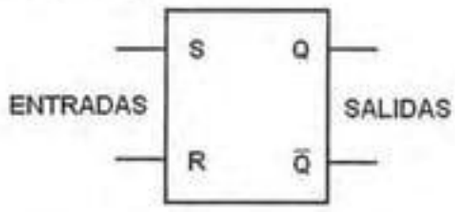


Fig. 3.5. Biestable asíncrono R-S

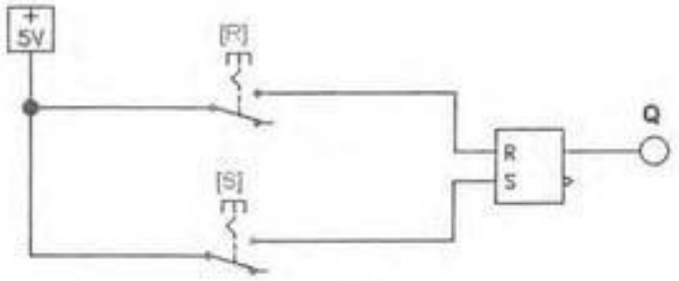
La figura siguiente muestra la tabla de la verdad de un biestable R-S asíncrono, donde X indica un estado prohibido.

R	S	Q	$\overline{Q}$
0	0	0	1
0	1	1	0
1	0	0	1
1	1	X	X

Fig. 3.6. Tabla de la verdad para un biestable R-S asíncrono

Ejemplo 3.1

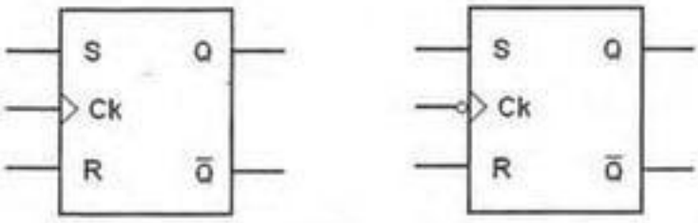
Explica el funcionamiento de la báscula R-S asíncrona en el circuito de la figura.



El circuito está formado por 2 pulsadores normalmente abiertos, un biestable R-S y una salida Q. Al activarse la entrada S (grabado), la salida Q pasa de forma permanente a nivel alto (uno lógico). Si, pasado un tiempo, se activa la entrada R (reseteado), la salida Q se pone permanentemente a nivel bajo (cero lógico). Nunca deben activarse las 2 entradas a la vez, porque no puede asegurarse cuál será el estado lógico de la salida.

3.2.2. Biestable R-S síncrono activado por flancos de reloj

Es un circuito secuencial que requiere de la existencia de una señal de reloj (en trada Ck) para que las entradas R-S provoquen una variación en el estado de l salida Q. En este caso, solo se tienen en cuenta los flancos de subida (positivos) de bajada (negativos) de la señal de sincronismo (reloj) para activar el biestable.



a) Activado por flanco de subida b) Activado por flanco de bajada

Fig. 3.7.
Biestable R-S síncrono activado por flanco de reloj

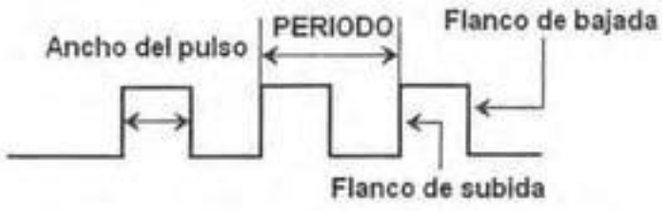


Fig. 3.8.
Señal de reloj de sincronismo

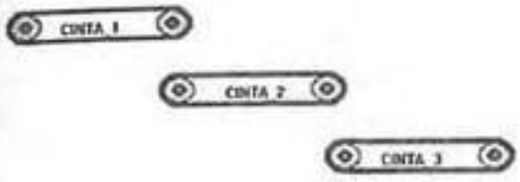
La figura siguiente muestra la tabla de la verdad de un biestable R-S síncrono, don de X indica un estado prohibido y Q_t es el nivel de salida anterior al pulso de reloj

Ck	R	S	Q
0	--	--	Q_t
1	0	0	Q_t
1	0	1	1
1	1	0	0
1	1	1	X

Fig. 3.9.
Tabla de la verdad para un biestable R-S síncrono

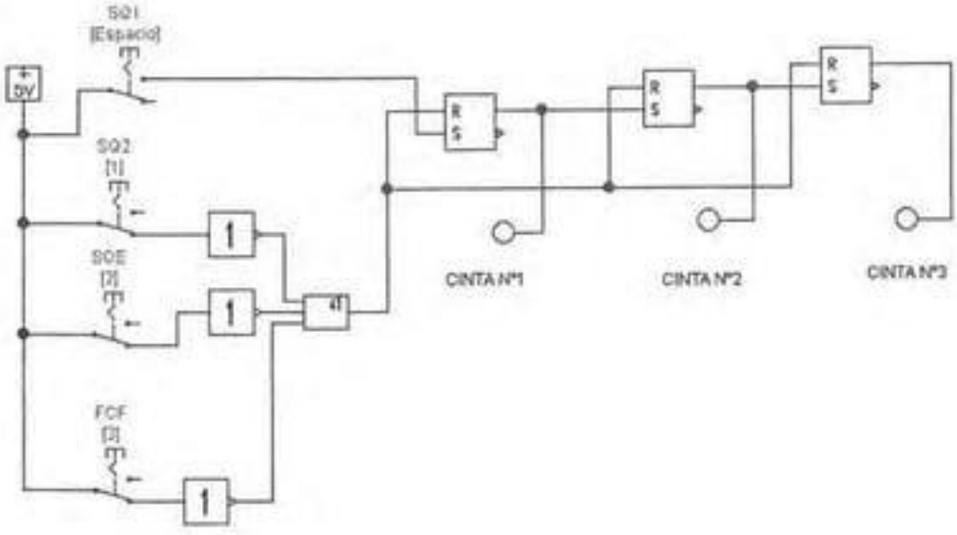
Ejemplo 3.2

Se pretende diseñar y montar el automatismo, con electrónica digital, de una cadena de montaje formada por tres cintas transportadoras. Se desea que al activar el pulsador de marcha SQ1 con la instalación sin anomalías, se ponga en marcha la cinta n.º 1 y, después de esta arranque la cinta n.º 2 para, por último, arrancar la cinta n.º 3. Si accionamos el pulsador SQ2 (pulsador de paro), el pulsador SQE (paro de emergencia) o se produce un defecto térmico FCF en cualquier motor se debe parar instantáneamente la instalación. Tres led (H1, H2 y H3) indicarán la marcha de los motores de las cintas.



APARATO	DENOMINACIÓN
SQ1	Pulsador de marcha
SQ2	Pulsador de paro
SQE	Pulsador de paro de emergencia
FCF	Defecto térmico de las cintas
H1	Led Motor Cinta nº 1
H2	Led Motor Cinta nº 2
H3	Led Motor Cinta nº 3

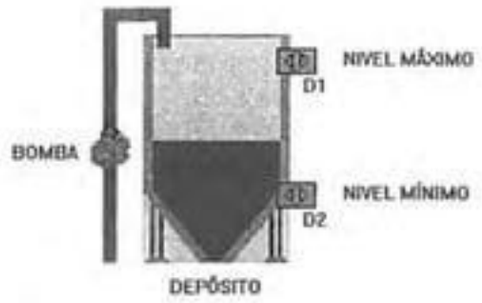
Se requieren 3 biestables R-S asíncronos para definir las condiciones de activación (set) y desactivación (reset) de cada uno de los leds de salida (motores de las cintas). La condición de set del biestable 1 viene dada por el pulsador SQ1, mientras la de los siguientes viene dada por la salida del precedente. Las condiciones de reset son comunes a los 3 biestables y requieren la negación de los pulsadores y una puerta OR para las 3 condiciones en paralelo.



Ejemplo 3.3

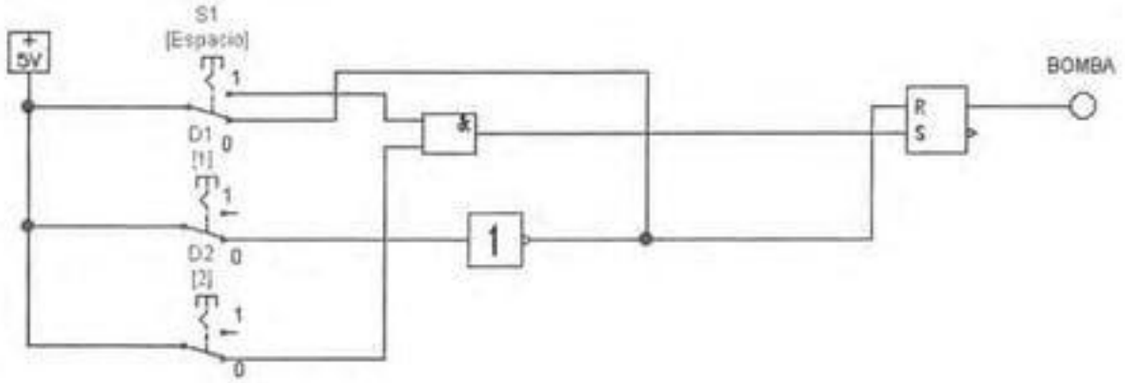
Se pretende diseñar y montar con electrónica digital el control de llenado de agua de un depósito de abonados, mediante una bomba de extracción de agua de un pozo. El automatismo funcionará como sigue:

- La instalación se pondrá en funcionamiento automático mediante un interruptor.
- Si el nivel de agua está por debajo del detector de nivel mínimo debe arrancar la bomba para llenar el depósito.
- Cuando el nivel de agua llegue al nivel máximo, se debe parar la bomba.



APARATO	DENOMINACIÓN
S1	Interruptor de marcha automática
D1	Detector de nivel máximo
D2	Detector de nivel mínimo
B	Bomba de agua

Se requiere un único biestable R-S asíncrono para definir las condiciones de activación (*set*) y desactivación (*reset*) de la bomba del depósito. La condición de set del biestable viene dada por una puerta AND para que se den en serie la condición del interruptor de marcha automático S1 y del detector de nivel mínimo D2. La condición de reset viene dada por la activación del detector de nivel máximo D1. Se debe considerar que los detectores sin agua equivalen a un contacto NC.



3.2.3. Biestable J-K asíncrono

Es un biestable muy similar al R-S asíncrono, pero permite eliminar la condición de salida no permitida cuando las 2 entradas están activadas (uno lógico). Para ello basta conectar 2 puertas AND a las entradas S y R.

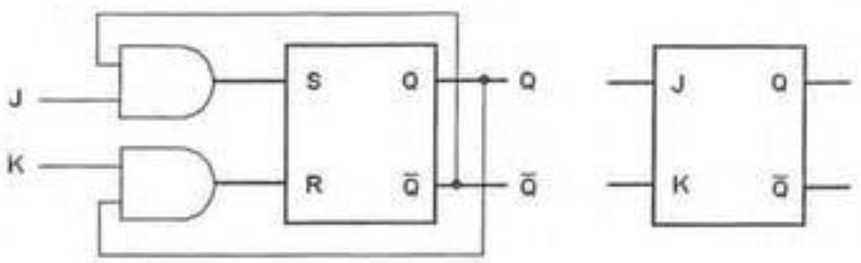


Fig. 3.10.
Biestable asíncrono J-K

3.2.4. Biestable J-K síncrono activado por flancos de reloj

Basta añadir una entrada de reloj Ck para pasar a tener un biestable en modo síncrono. Al igual que antes, el biestable puede activarse por flancos de subida o flancos de bajada.

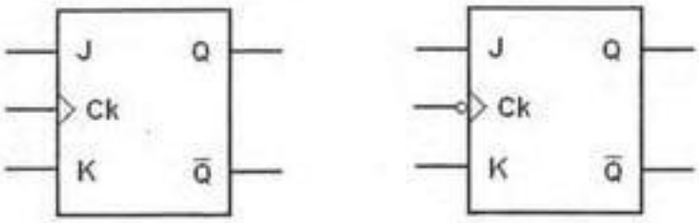


Fig. 3.11.
Biestable J-K síncrono activado por flanco de reloj

a) Activado por flanco de subida b) Activado por flanco de bajada

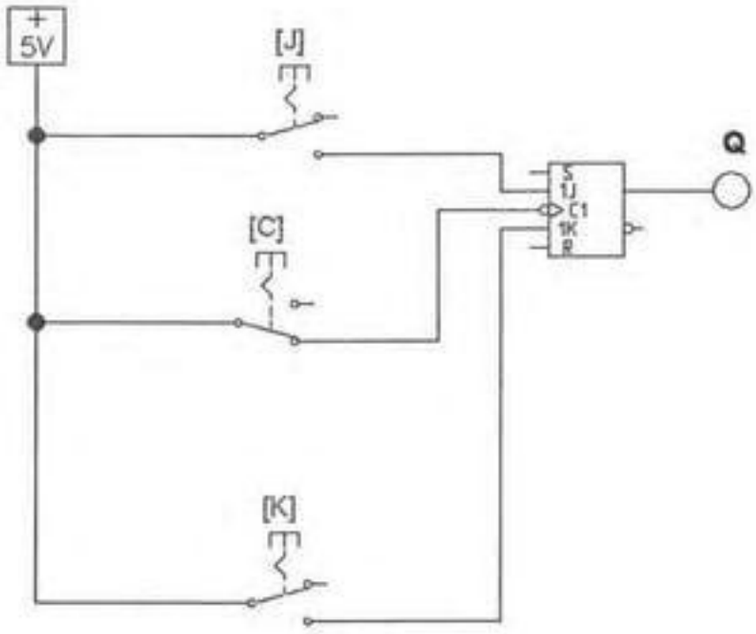
La figura siguiente muestra la tabla de la verdad de un biestable J-K síncrono, donde Q_t es el nivel de salida anterior al pulso de reloj.

Ck	J	K	Q
0	--	--	Q_t
1	0	0	Q_t
1	0	1	0
1	1	0	1
1	1	1	Q_t

Fig. 3.12.
Tabla de la verdad para un biestable J-K síncrono activado por flanco de reloj

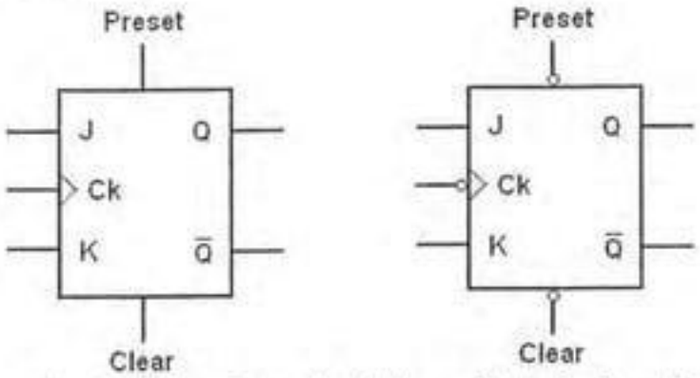
Ejemplo 3.4

Explica el funcionamiento de la báscula J-K síncrona en el circuito de la figura.



El circuito está formado por 2 pulsadores normalmente abiertos (J y K), un biestable J-K síncrono activado por flanco de bajada y una salida Q. Ante la presencia de una señal de reloj C, si se activa el pulsador J y hay un flanco negativo en C, la salida Q pasa de forma permanente a nivel alto (uno lógico). Si, pasado un tiempo, se activa la entrada K y hay un flanco negativo en el reloj C, la salida Q se pone permanentemente a nivel bajo (cero lógico).

Dado que los biestables síncronos se suelen integrar en un solo chip o CI, es tual añadir un par de entradas asíncronas más (*Preset* y *Clear*) que permiten el biestable en estado 1 (*Preset*) o en estado 0 (*Clear*) independientemente señales de reloj. Ambas entradas pueden ser disparadas o por flanco de sub por flanco de bajada.

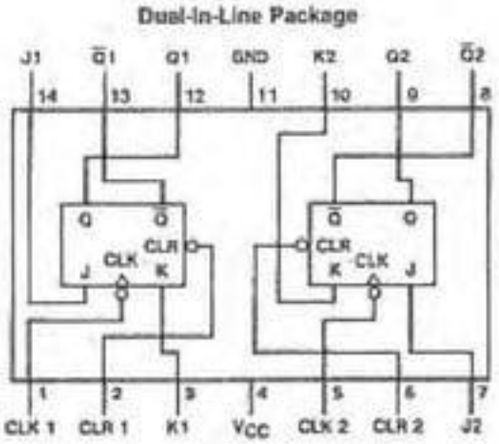


a) Activado por flanco de subida b) Activado por flanco de ba

Fig. 3.13.
Biestable J-K síncrono con entradas Preset y Clear

A nivel comercial, se pueden destacar los siguientes circuitos integrados CI de tables de la serie 74 de la familia lógica TTL:

- Circuito integrado 74LS73: Doble biestable J-K con Clear.

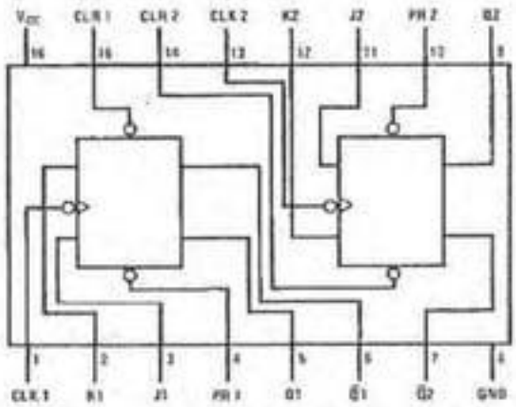


Inputs				Outputs
CLR	CLK	J	K	Q
L	X	X	X	L
H	↓	L	L	Q_0
H	↓	H	L	H
H	↓	L	H	L
H	↓	H	H	Toggle
H	H	X	X	Q_0

H = Nivel lógico alto (1);
L = Nivel lógico bajo (0); X = Irrelevar

Fig. 3.14.
Diagrama de conexiones y tabla de la verdad 74LS73

- Circuito integrado 74LS112: Doble biestable J-K con Preset y Clear disparad flanco de bajada.

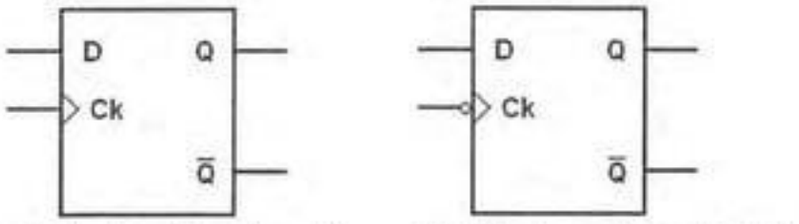


Inputs					Outputs	
PR	CLR	CLK	J	K	Q	$\overline{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H (Note 1)	H (No
H	H	↓	L	L	Q_0	$\overline{Q}_0$
H	H	↓	H	L	H	L
H	H	↓	L	H	L	H
H	H	↓	H	H	Toggle	
H	H	H	X	X	Q_0	$\overline{Q}_0$

Fig. 3.15.
Diagrama de conexiones y tabla de la verdad 74LS112

3.2.5. Biestable D síncrono activado por flancos de reloj

Es un biestable que solo posee una entrada de datos D y una de reloj Ck. Se obtiene a partir de un biestable R-S síncrono de forma que $S = D$ y $R = \overline{D}$. Se puede activar por flanco de subida (positivo) o de bajada (negativo) de la señal del reloj.



a) Activado por flanco de subida b) Activado por flanco de bajada

Fig. 3.16. Biestable D síncrono activado por flanco de reloj

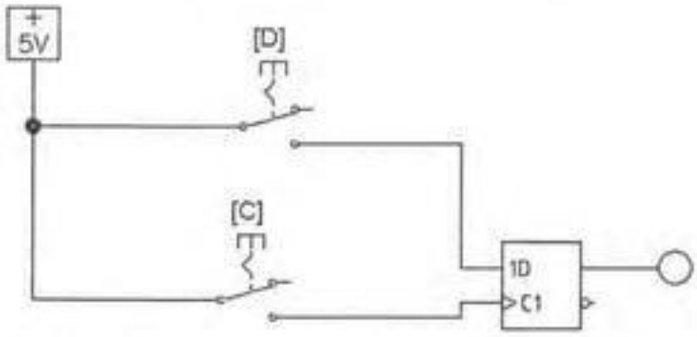
La figura siguiente muestra la tabla de la verdad de un biestable D síncrono.

Ck	D	Q
1	1	1
1	0	0

Fig. 3.17. Tabla de la verdad para un biestable D síncrono

Ejemplo 3.5

Explica el funcionamiento de la báscula D síncrona en el circuito de la figura.



El circuito está formado por 2 pulsadores normalmente abiertos (D y C) y un biestable D síncrono activado por flanco de subida y una salida Q. Ante la presencia de una señal de reloj C, si se activa el pulsador D y hay un flanco positivo en C, la salida Q pasa de forma permanente a nivel alto (uno lógico). Si en el siguiente flanco de subida se desactiva el pulsador D, la salida Q se pone permanentemente a nivel bajo (cero lógico).

3.3. Registros de desplazamiento

Los registros de desplazamiento son circuitos secuenciales que tienen la capacidad de almacenar un dato de n bits. Están formados por biestables de tipo D conectados en cascada y una circuitería adicional que permite controlar los modos de escritura y la lectura de los datos almacenados. Se utilizan como almacenamiento temporal de un conjunto de bits que están siendo procesados.

En un registro de desplazamiento, la información de entrada se va desplazand
las diferentes salidas mediante una señal de sincronismo; si se deja de enviar l
ñal de sincronismo, en la salida se mantiene la información anterior (memoriza

La información de entrada se puede escribir en serie o en paralelo. La informa
de salida también se puede leer en serie o en paralelo. Combinando estas for
se puede hablar de diferentes tipos: 1) entrada o escritura en serie, salida o lec
en serie; 2) entrada o escritura en serie, salida o lectura en paralelo; 3) entra
escritura en paralelo, salida o lectura en serie; 4) entrada o escritura en paral
salida o lectura en paralelo.

En serie, los datos se escriben/leen bit a bit, en secuencia, a través de una ú
línea. En paralelo, todos los bits del dato se leen/escriben a la vez, a través
múltiples líneas.

En un registro de desplazamiento de entrada serie y salida serie, solo la entr
del primer biestable y la salida del último son accesibles externamente. Se
plean como líneas de retardo digitales y en tareas de sincronización.

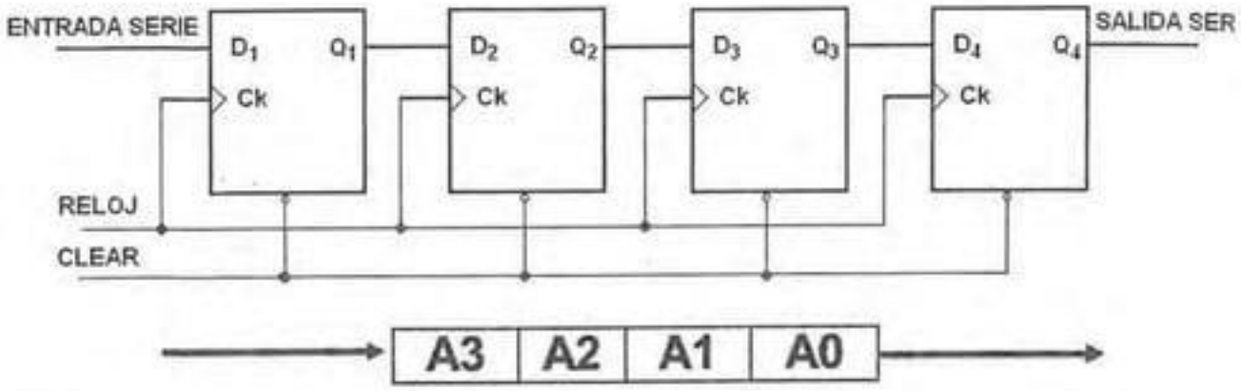


Fig. 3.18.
Registro de desplazamiento con entrada y salida serie

En cada flanco de subida del reloj, se produce un desplazamiento hacia la dere
de 1 bit. Por A_0 saldrá la misma secuencia que entró por la entrada serie.

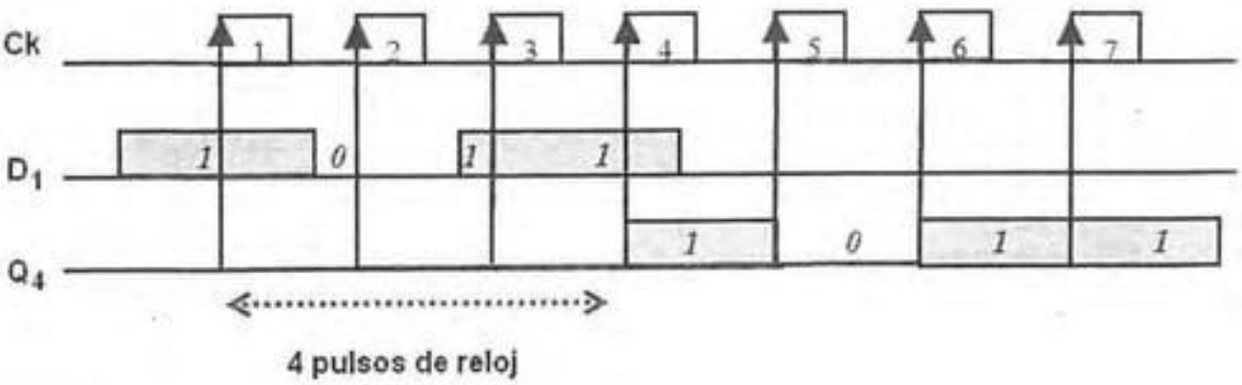


Fig. 3.19.
Cronograma de un registro con entrada y salida serie

En un registro de desplazamiento de entrada serie y salida paralelo, son accesib
las salidas de todos los biestables y la entrada del primero. Permiten conve
datos serie en datos paralelo como en los puertos RS232.

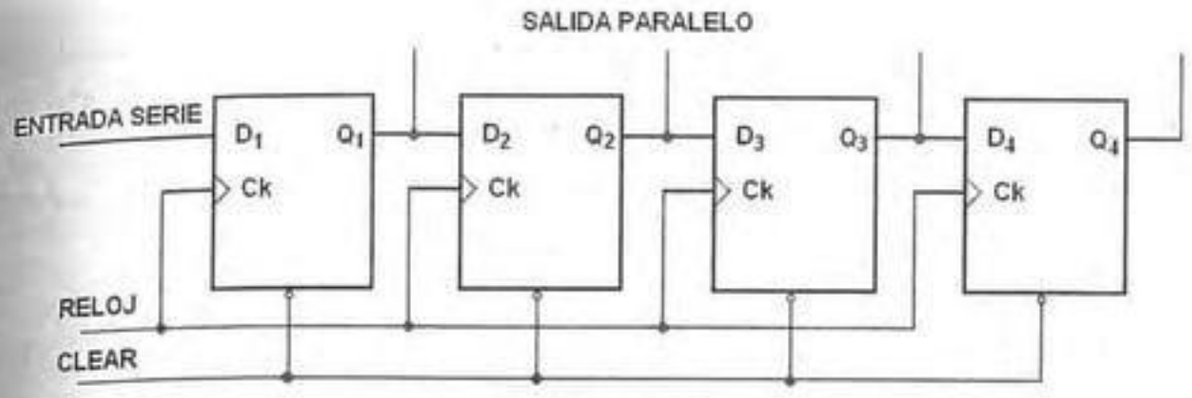


Fig. 3.20.
Registro de desplazamiento con entrada serie y salida paralelo

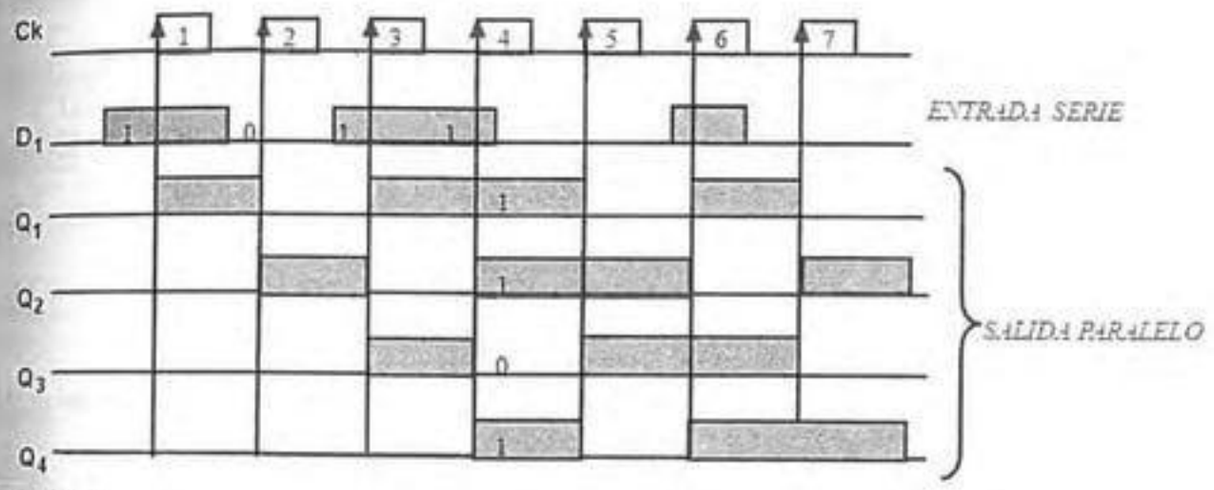


Fig. 3.21.
Cronograma de un registro con entrada serie y salida paralelo

A nivel comercial se puede encontrar el integrado de la familia CMOS 74HC164, que es un registro de desplazamiento de entrada serie y salida paralelo de 8 bits.

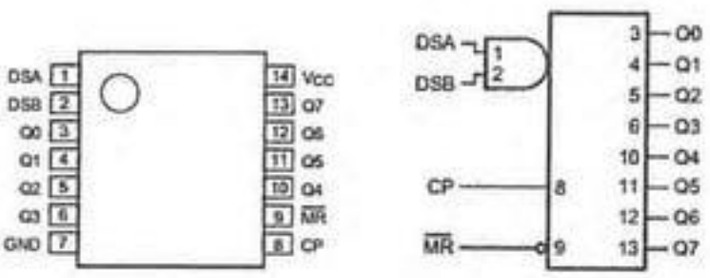


Fig. 3.22. Patillaje y símbolo lógico de un registro de desplazamiento 74HC164

El integrado 74164 tiene una entrada de reloj (CP) y otra de reset general (MR), activa a nivel bajo, y que pone a cero todas las salidas. Para la entrada de datos dispone de 2 entradas (DSA, DSB) unidas internamente a través de una puerta AND. Las salidas del integrado están numeradas desde Q_0 hasta Q_7.

Cuando el pin de puesta a cero tiene un nivel lógico bajo, independientemente del estado del resto de entradas, las salidas pasarán a estar todas a cero. Si está a nivel lógico alto, el resultado del AND de las dos entradas A y B será desplazado por cada salida desde Q_0 hasta Q_7 a cada flanco ascendente de la señal de reloj.

Recuerda . . .

Una báscula S/R almacena el estado de un bit (0 o 1). Las básculas más utilizadas en los PLC o autómatas son:

- S/R: Preferencia al Reset.
- R/S: Preferencia al Set.

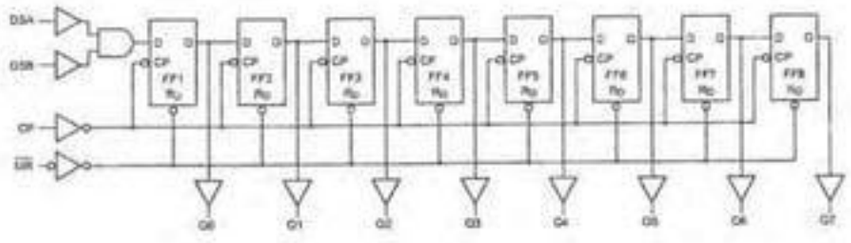
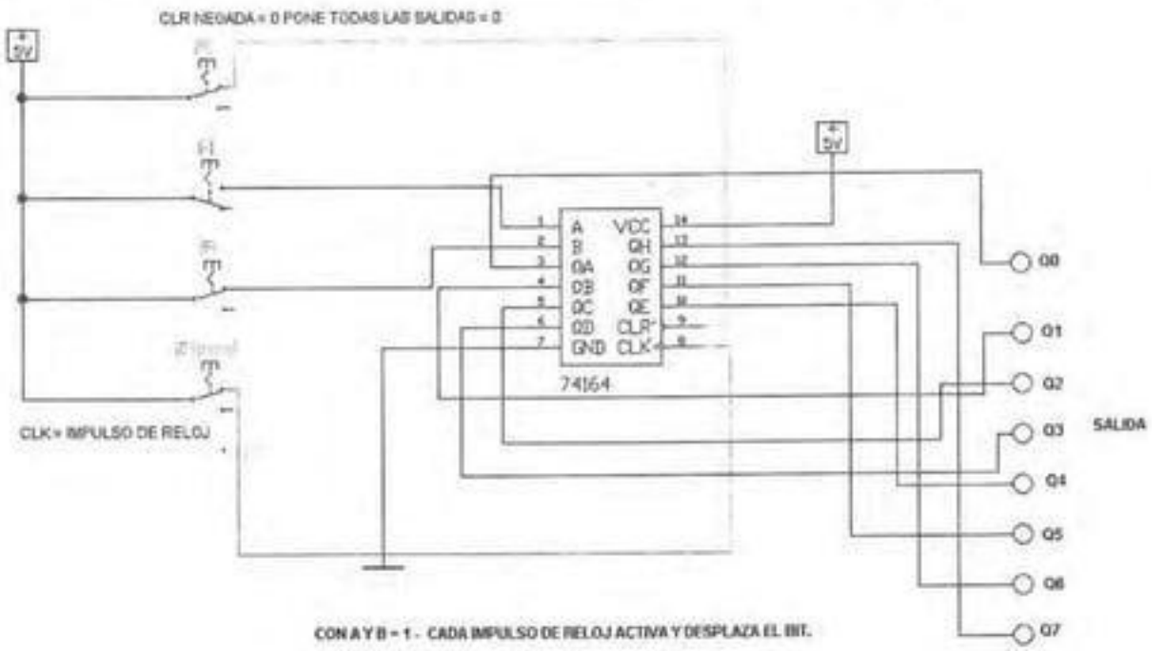


Fig. 3.23.
Diagrama funcional de un registro de desplazamiento 74HC164

Ejemplo 3.6

Explica el funcionamiento del registro de desplazamiento 74164 en el circuito de la figura.



El circuito está formado por 2 pulsadores para las entradas A y B y 2 pulsadores más para activar el pulso de reloj (CLK) y para poder resetear a cero (CLR = Clear) todas las salidas, junto con un integrado 74164. Las entradas A y B son serie, mientras las salidas son paralelo. Cuando se introduce una señal de sincronismo en el integrado y las dos entradas A y B están activas en nivel alto (puerta AND), para cada flanco positivo de la señal de reloj, se desplaza el dato resultante de las entradas desde la salida Q_A hasta la salida Q_H.

A nivel comercial también se puede encontrar el integrado de la familia CMOS 74HC174, que es un registro universal de 4 bits bidireccional (desplazamiento hacia la derecha o hacia la izquierda).

3.4. Contadores

Un contador es un circuito secuencial formado por biestables y puertas lógicas que permite realizar la cuenta de los impulsos recibidos en su entrada (estado interno en cada instante). El contaje se puede realizar en binario (para n bits, cuentan desde 0 hasta 2^n -1) o en BCD (cuentan del 0 al 9). El contaje puede ser solo ascendente, solo descendente o ascendente-descendente.

En el caso que los impulsos recibidos ocurran a intervalos de tiempo conocidos, un contador permite realizar también la medida de tiempos (temporizador). Según su modo de funcionamiento, los contadores pueden ser síncronos o asíncronos.

3.4.1. Contadores asíncronos

Son aquellos en los que los impulsos de contaje no se aplican a las entradas de reloj de todos los biestables J-K, sino generalmente solo al primer biestable A. Por tanto, las entradas de reloj del resto de biestables están marcadas por las salidas del biestable precedente. Las entradas de los biestables J-K se unen entre sí a un uno lógico (biestable de tipo T).

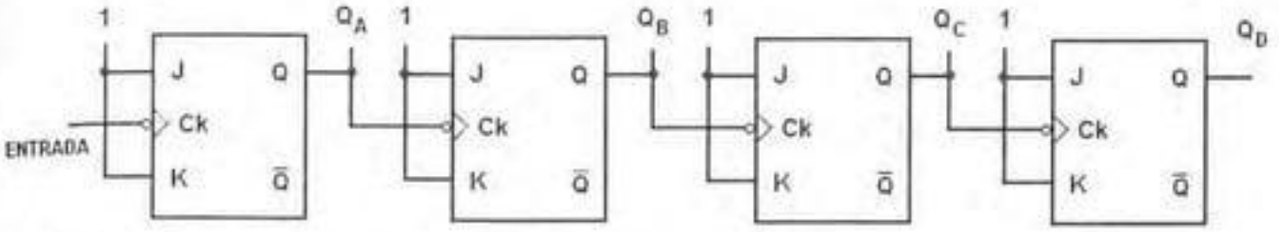
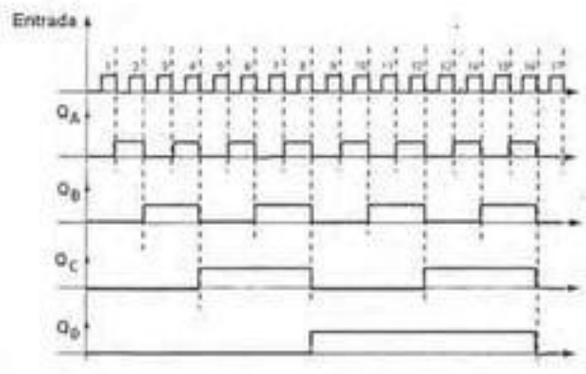


Fig. 3.24.
Contador asíncrono con básculas J-K

En el ejemplo, la entrada del contador es la entrada de reloj del primer biestable A, de forma que su salida Q_A se conecta a la entrada de reloj del siguiente biestable y así sucesivamente. Además, como las entradas están a nivel alto lógico, las salidas cambian de estado en los flancos de bajada de la entrada de reloj de cada uno de ellos.



EN	QD	QC	QB	QA
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Fig. 3.25.
Cronograma de un contador asíncrono de 4 bits (binario de 0 a 15)

En el caso de desear un contador asíncrono ascendente para contar un número que no sea potencia de 2, por ejemplo de 0 a 9 lo que implica N = 10 estados, se necesita conectar en cascada un número X = 4 de biestables J-K que cumplan:

$$2^{X-1} \leq N \leq 2^X; \qquad 2^3 = 8 \leq N = 10 \leq 2^4 = 16;$$

En este contador MOD10 de décadas es necesario forzar un nuevo inicio del ciclo (1010 en binario, que equivale a 10) antes de haber pasado por todos los estados normales. Cuando el contador adquiera el estado 1010 ($Q_D Q_C Q_B Q_A$), se obliga en el siguiente flanco de subida del reloj a que todas las salidas que estén a nivel alto se pongan a nivel bajo usando las entradas asíncronas de RESET de los biestables. Se requiere para ello una puerta NAND (decodificador parcial) que tenga conectadas como entradas los biestables a nivel alto del número binario 1010.

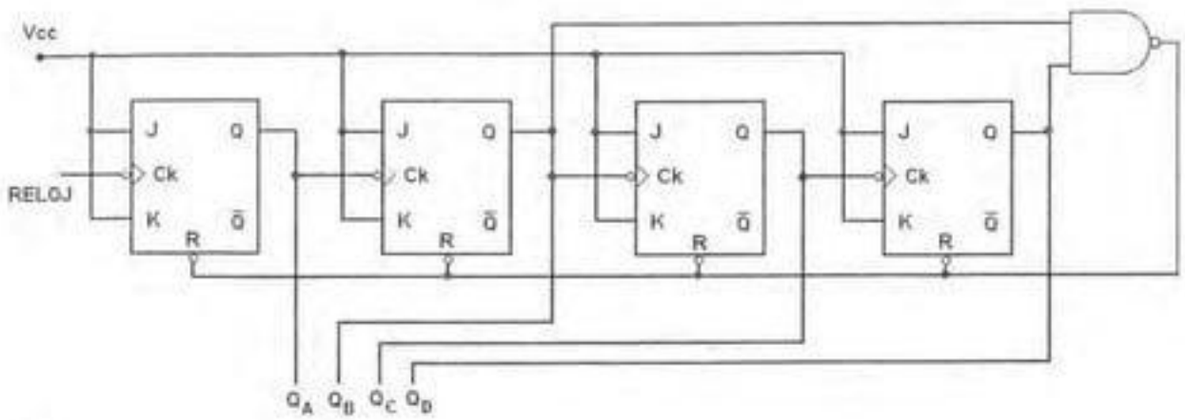


Fig. 3.26.
Contador asíncrono de décadas (de 0 a 9)

El circuito se completaría con un decodificador y un *display* BCD de 7 segmentos de ánodo común, por ejemplo, para poder visualizar el número decimal del contaje.

A nivel comercial se puede encontrar el integrado de la familia TTL 74LS293, que es un contador asíncrono binario de 4 bits.

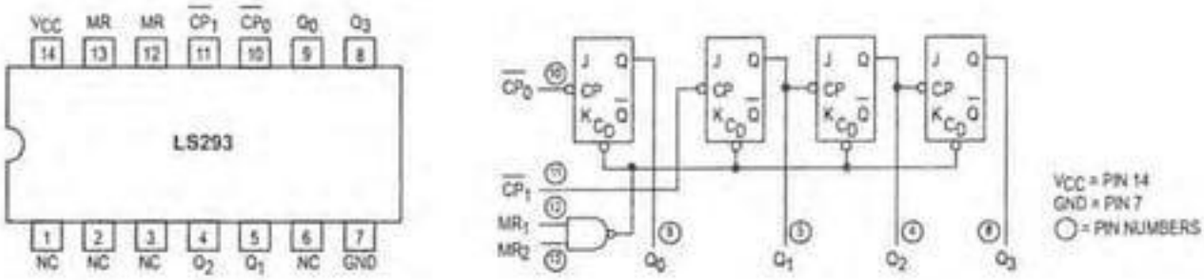


Fig. 3.27.
Patillaje y diagrama lógico de un contador binario de 4 bits 74LS293

Este circuito integrado tiene 4 biestables (salidas Q_0, Q_1, Q_2 y Q_3), donde Q_0 corresponde al LSB y Q_3 al MSB. El orden de los biestables J-K es Q_0, Q_1, Q_2 y Q_3. Cada biestable tiene una entrada CP para el pulso de reloj CLK. Las entradas de reloj CP_0 y CP_1 son de acceso externo. Además, cada biestable tiene una entrada asíncrona CD para borrar o resetear el estado (CLEAR). Las entradas MR1 y MR2 de la puerta NAND permiten reiniciar el contador si ambas entradas están en nivel lógico alto.

3.4.2. Contadores síncronos

Los contadores síncronos son aquellos en los que todos los biestables conmutan a la vez, es decir, con una señal de reloj común para todos. Se requiere una lógica adicional con puertas básicas (AND, NOT, OR...) conectada a las entradas de los biestables. Esta lógica se debe diseñar en cada caso.

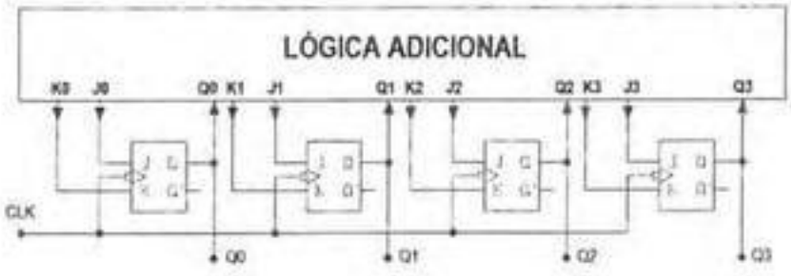
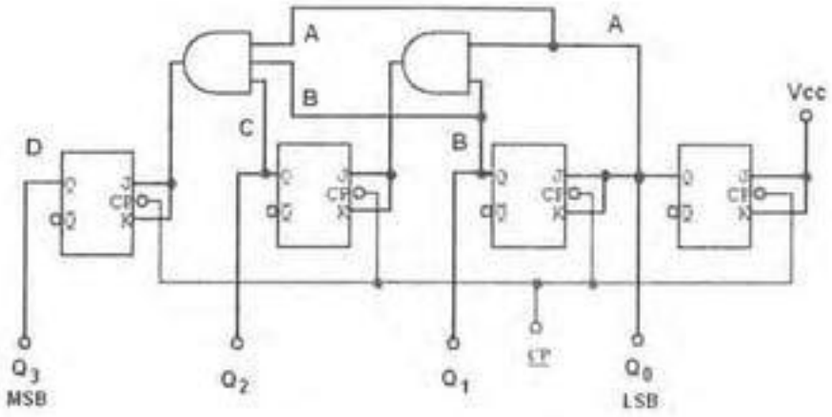


Fig. 3.28.
Contador síncrono de 4 bits

Recuerda • • •

Las básculas pueden ser asíncronas (no interviene el tiempo) o síncronas (interviene el tiempo). Con básculas tipo D síncronas podemos hacer registros de desplazamiento. Con básculas tipo J/K síncronas podemos hacer contadores síncronos y asíncronos.

Como ejemplo se puede considerar un contador BCD síncrono ascendente de 4 bits con biestables J-K (T). En este contador, como se aprecia en la secuencia de conteo, el primer bit siempre cambia de estado con un pulso de reloj, el segundo bit cambia cada dos pulsos, el tercer bit cada cuatro pulsos y el cuarto bit cada ocho pulsos de reloj.



Q_3 (MSB)	Q_2	Q_1	Q_0 (LSB)	Número de pulsos
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	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Fig. 3.29. Tabla de conteo de un contador síncrono de 4 bits

El contador síncrono se debe construir sabiendo que cuando $Q_0 = 1$, Q_1 debe bascular entre el 0 y el 1. Si se tiene un contador con 3 biestables, la basculación de Q_2 debe producirse si $Q_0 = Q_1 = 1$ y así sucesivamente. Para ello se requieren 2 puertas AND, de forma que la entrada al biestable 2 sea el producto de las salidas Q_0 y Q_1 (puerta 1) y la entrada al biestable 3 sea el producto de las salidas Q_0, Q_1 y Q_2 (puerta 2).

El circuito se completaría con un decodificador y un *display* BCD de 7 segmentos de ánodo común, por ejemplo, para poder visualizar el número decimal del contaje.

A nivel comercial se puede encontrar el integrado de la familia TTL 74LS193, que es un contador binario síncrono de 4 bits ascendente/descendente. Este contador síncrono cuenta con relojes separados para el contaje ascendente y para el contaje descendente. El cambio de estado síncrónico a la salida del 74LS193 se obtiene con transiciones de nivel bajo a nivel alto de la señal de reloj.

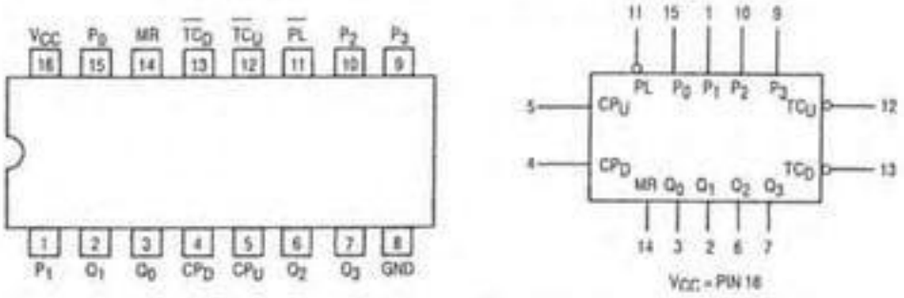
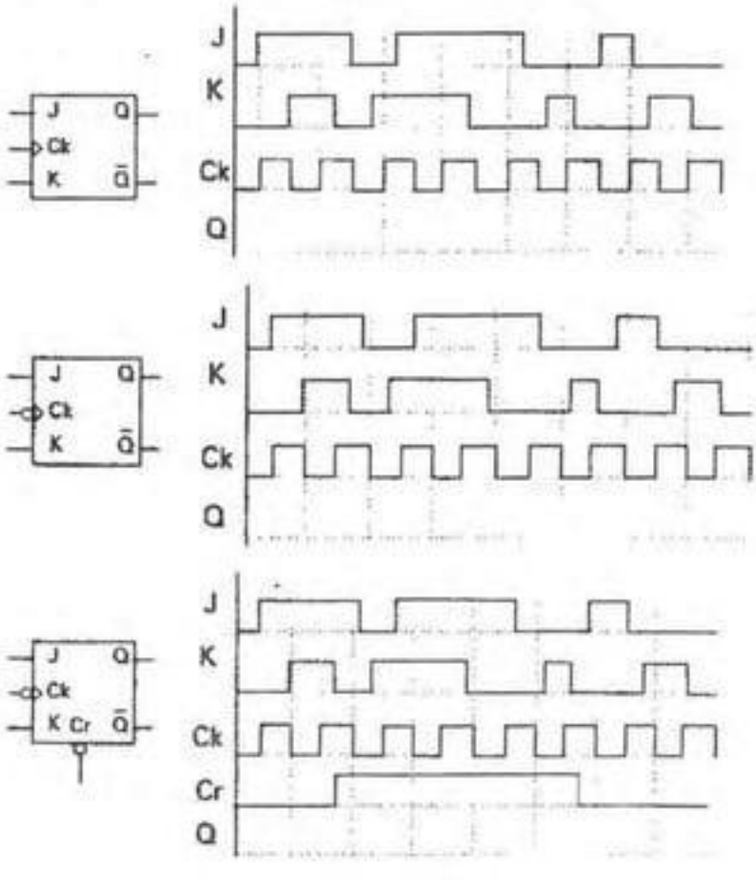


Fig. 3.30. Patillaje y símbolo lógico de un contador binario de 4 bits 74LS193

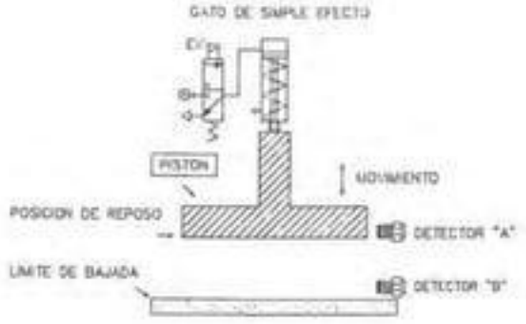
Este circuito integrado tiene 4 biestables (salidas Q_0, Q_1, Q_2 y Q_3), donde Q_0 corresponde al LSB y Q_3 al MSB. Cada biestable tiene una entrada CPU para el pulso de reloj de contaje ascendente y una entrada CPD para el pulso de reloj de contaje descendente. La entrada PL permite activar la carga de datos paralelo. La entrada MR permite reiniciar el contador si está en nivel lógico alto. Las entradas P_n corresponden a las entradas de datos para el número que se va a cargar mientras Q_n son las salidas del contador síncrono. Por último, la salida TCU enviará un pulso cada vez que el contador se recicle en contaje ascendente. De forma análoga, se incluye la salida TCD para el contaje descendente.

Actividades propuestas

1. Completa los cronogramas correspondientes que se indican a continuación:



2. La operación de prensado la realiza un pistón, que gobernado por gato de simple brazo baja hasta la posición **B** e inmediatamente sube hasta la posición de reposo, determinada por el detector de posición **A** de la figura. Al cerrar brevemente el interruptor **PM** se ejecuta una operación de prensado, bajando y subiendo el pistón, (activación y desactivación de «EV») deteniéndose en la posición **A** de partida. Obsérvese que cada vez que se pulsa **PM** el pistón efectúa un ciclo completo. Los detectores A y B son con contacto NO.

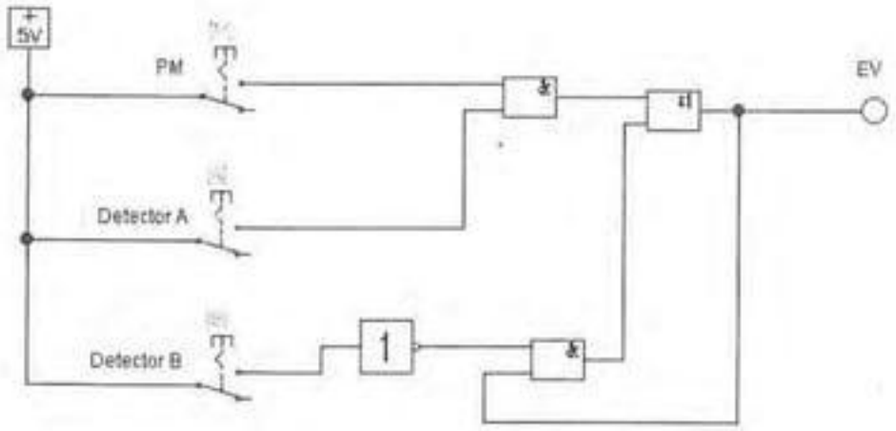


Se pide la ecuación de funcionamiento del automatismo y circuito con puertas lógicas y básculas S/R.

RESULTADO:

Con puertas lógicas.

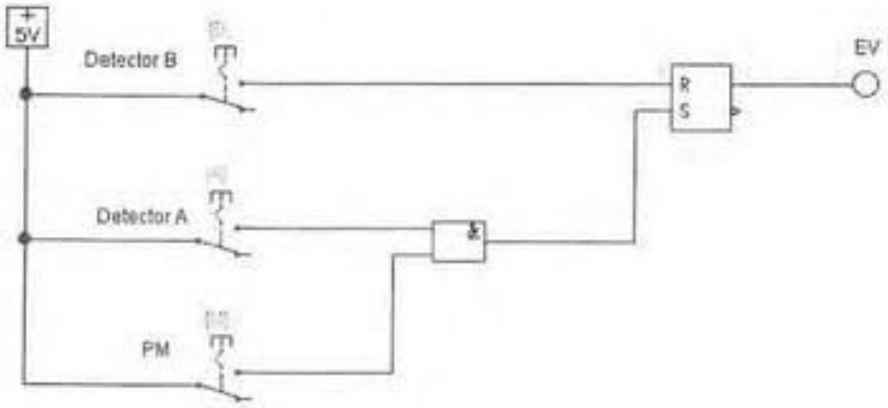
$$EV = (PM \times A) + (EV \times \overline{B})$$



Con puertas lógicas y báscula S/R.

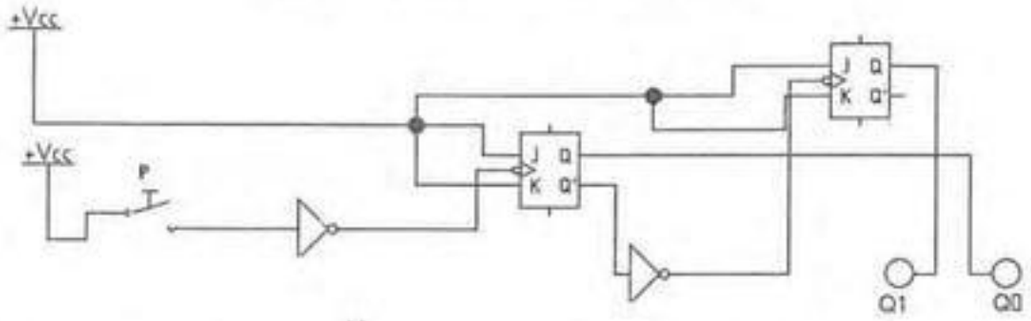
EV Set = PM x A

EV Reset = B



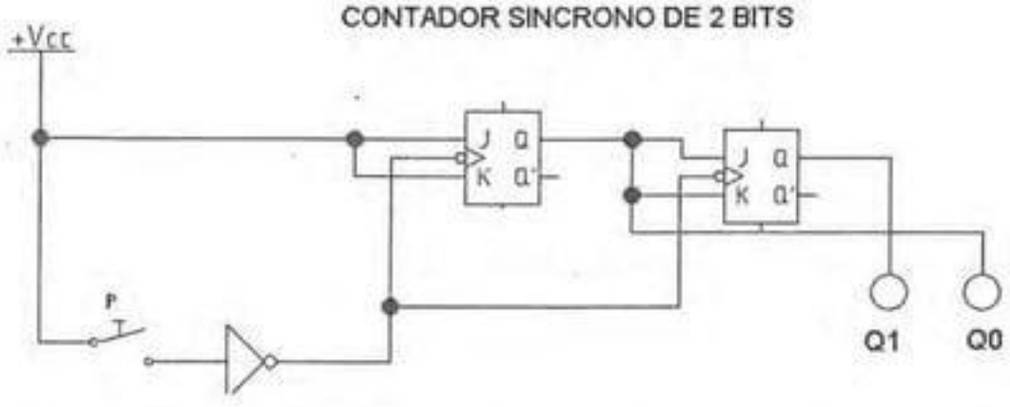
3. Con básculas J/K realiza un contador asíncrono binario de dos bits, móntalo y pruébalo en Workbench.

CONTADOR ASINCRONO BINARIO DE DOS BITS



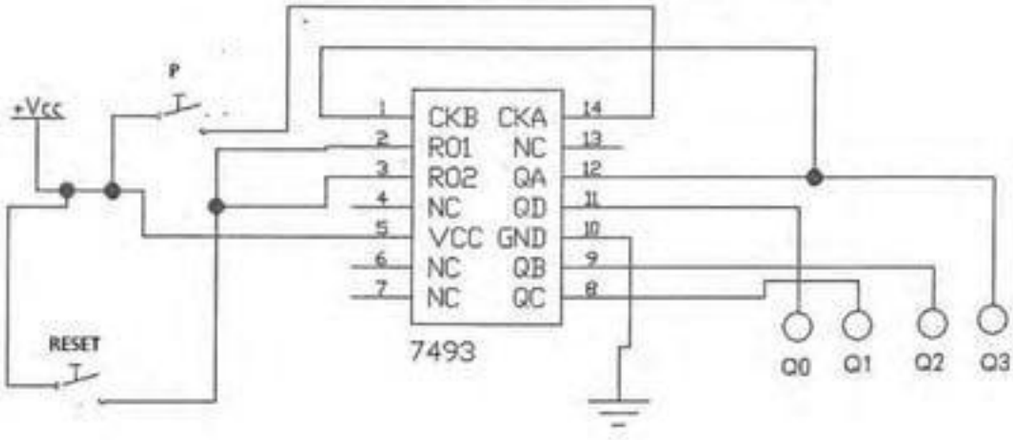
4. Con básculas J/K realiza un contador síncrono binario de dos bits, móntalo y pruébalo en Workbench.

CONTADOR SINCRONO DE 2 BITS

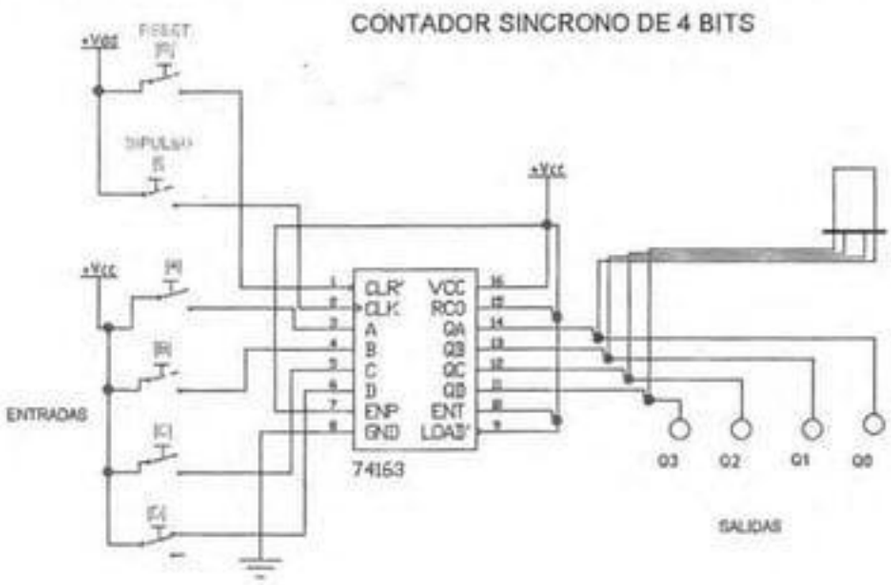


5. Monta y prueba en Workbench un contador asíncrono de cuatro bits con el circuito integrado 7493.

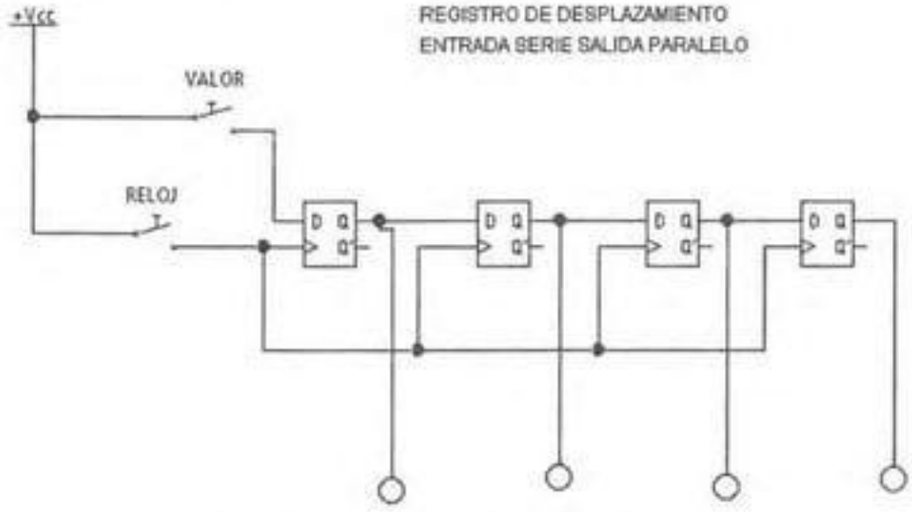
CONTADOR BINARIO ASINCRONO DE 4 BITS



6. Monta y prueba en Workbench un contador síncrono de cuatro bits con el circuito integrado 74163.

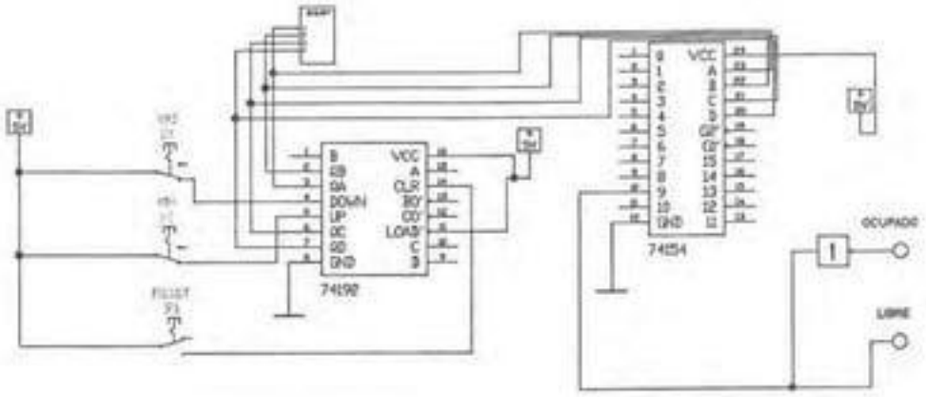


7. Se desea hacer un contador asíncrono de 4 bits con cuatro básculas «T». Conectar al mismo un decodificador 74154 binario – decimal 1 de 16. Cuando el valor actual del contador sea 15 se activará un diodo led. Diseña y monta el circuito con electrónica digital.



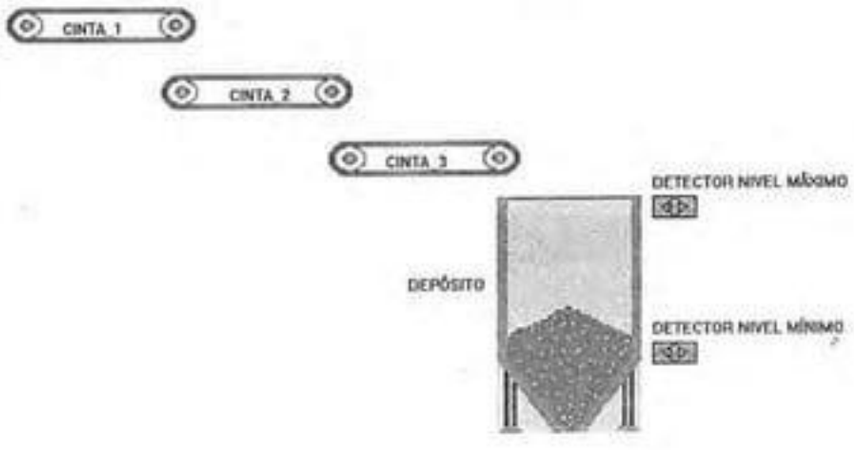
8. Se pretende realizar el control de un aparcamiento de tal forma que cuando en él tengamos igual o menos de 8 coches, esté encendido un piloto verde de libre y cuando tengamos 9 coches se encienda un piloto de ocupado. Diseña el circuito con electrónica digital y móntalo con Workbench.

APARATO	DENOMINACIÓN
SB1	Detector de entrada
SB2	Detector de salida
H1	Piloto verde libre
H2	Piloto rojo ocupado



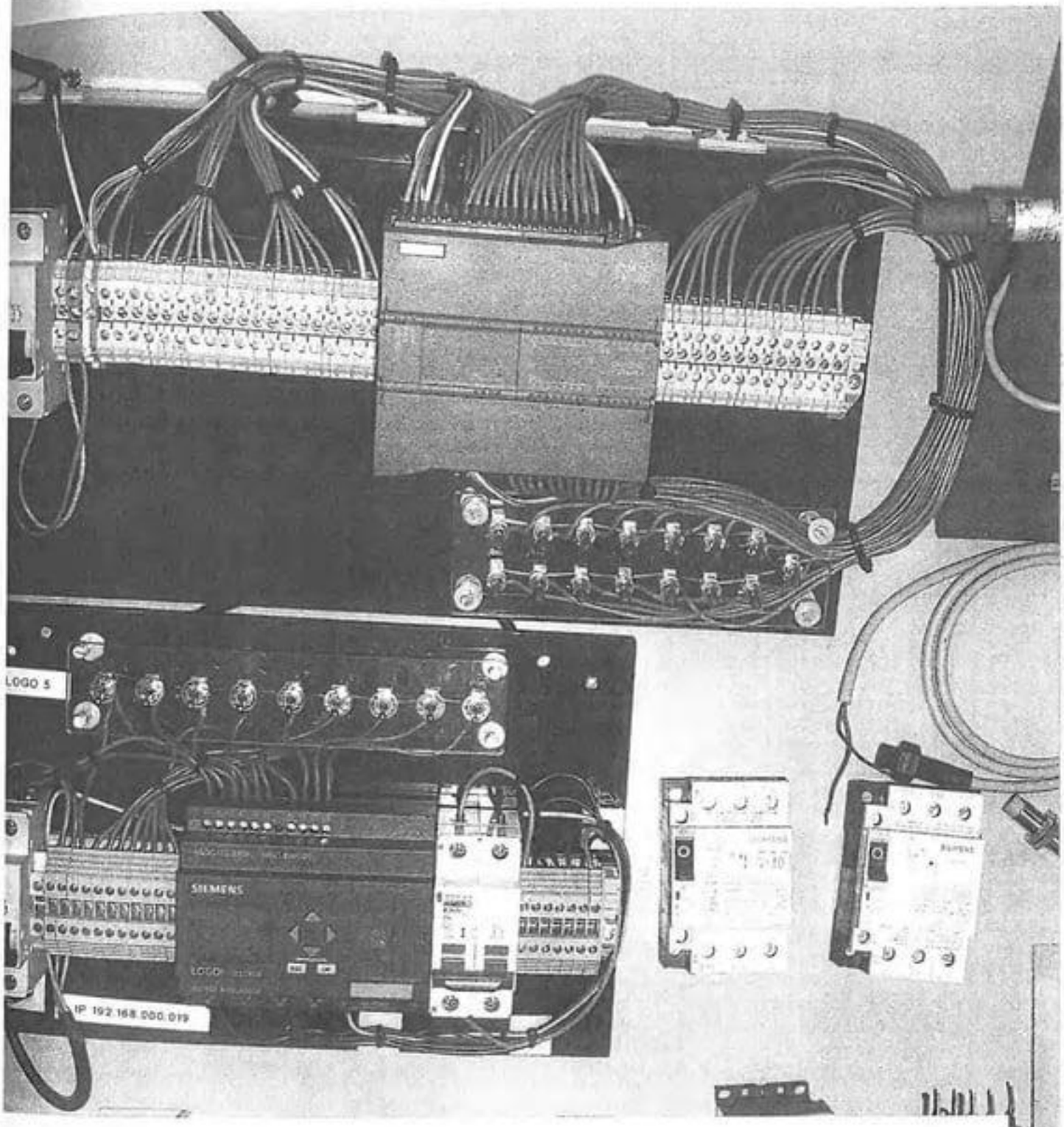
9. Se pretende diseñar y montar el automatismo para una cadena de montaje formada por tres cintas transportadoras y un silo de áridos. Se desea que al activar el pulsador de marcha SQ1 con la instalación sin anomalías y el silo vacío, se ponga en marcha la cinta n.º 1 y, después de esta, arranque la cinta n.º 2 para y por último mover la cinta n.º 3. Si accionamos el pulsador SQ2 (pulsador de paro), el pulsador SQE (paro de emergencia), se produce un defecto térmico FCF en cualquier motor o el silo se llena, se debe parar instantáneamente la instalación. Tres leds (H1, H2 y H3) indicarán la marcha de los motores de las cintas.

APARATO	DENOMINACIÓN
SQ1	Pulsador de marcha NO
SQ2	Pulsador de paro NC
SQE	Pulsador de paro de emergencia NC
FR	Defecto térmico de motores de cintas 1, 2 y 3. NC
D. Mínimo	Detector de nivel mínimo. NO
D. Máximo	Detector de nivel máximo. NO
H1	Señalización marcha cinta 1.
H2	Señalización marcha cinta 2.
H3	Señalización marcha cinta 3.



Se pide el diseño del circuito con electrónica digital y el montaje y prueba en Workbench.

Unidad 4 Detectores y preactuadores



En este capítulo:

4.1. Introducción

4.2. Entradas digitales a los autómatas programables

4.3. Entradas analógicas a los autómatas programables

4.4. Salidas digitales de los autómatas programables

4.5. Salidas analógicas de los autómatas programables

4.1. Introducción

Este tema pretende abordar, de forma resumida, todos aquellos elementos que son necesarios en todo proceso automatizado debido a su conexión en los sistemas secuenciales programables o autómatas programables. Como se desarrollará en temas posteriores, los autómatas requieren de entradas del proceso automatizado y de salidas al mismo proceso automatizado.

Las entradas están previstas para recoger información del estado de la automatización y permitir la interacción del hombre en el proceso automatizado. En este grupo se deben considerar entradas digitales como finales de carrera, detectores de posición, pulsadores, relés térmicos, etc., así como entradas analógicas mediante transductores que son capaces de medir de forma continua en el tiempo aquellas magnitudes físicas que son importantes para el proceso automatizado.

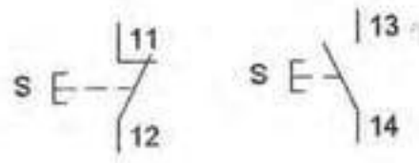
Las salidas están previstas para poder actuar sobre el proceso modificándolo para que este se pueda ajustar a los requerimientos impuestos por el diagrama de flujo del proceso productivo automatizado. Esto puede implicar, en un determinado momento, conectar o desconectar un actuador eléctrico (mayoritariamente, un motor asíncrono trifásico) o neumático (cilindro de simple o de doble efecto). Además, muchas veces es necesario señalizar un determinado estado del proceso; para tal fin se utilizan señalizaciones ópticas (pilotos de señalización) o acústicas (sirenas).

Estos actuadores nunca se conectan directamente al autómata programable, pues cada uno de ellos necesita de preactuadores digitales conectados al autómata. En el caso de un actuador eléctrico, este preactuador puede ser electromecánico (contactor) o electrónico (relé de estado sólido). En el caso de un actuador neumático, se requieren las correspondientes electroválvulas o preactuadores neumáticos. Además, para contemplar los procesos de reducción de la corriente de arranque de los motores asíncronos trifásicos, suele ser necesario utilizarsalidas digitales en las que conectar arrancadores estáticos. También se utilizan salidas analógicas con las que controlar un variador de velocidad de un motor eléctrico.

4.2. Entradas digitales a los autómatas programables

4.2.1. Pulsadores

Como entradas al autómata se suelen emplear botoneras con pulsadores que permiten al operario interactuar con el proceso. Debajo de los pulsadores se colocan los contactos de cámara (verde para la marcha con contacto NO; negro para el paro con contacto NC), según lo deseado.



PULSADOR NC PULSADOR NO

Fig. 4.1. Botonera con pulsadores (2NO+1NC)

Recuerda • • •

Los detectores y sensores digitales y analógicos actúan como entradas en los PLC o autómatas.

Recuerda • • •

Un relé térmico diferencial protege a los motores contra las sobrecargas y el fallo de una fase. Además del relé térmico, debemos utilizar una protección contra cortocircuitos (fusibles AM o interruptores magnetotérmicos de curvas C o D).

4.2.2. Relé térmico

El relé térmico controla el calentamiento de los devanados de los motores y provoca la apertura del **contacto** asociado cuando se alcanza el calentamiento límite debido a una sobrecarga. Su funcionamiento se basa en un bimetal, que se calienta y se dobla a causa del calor producido por la corriente eléctrica. El bimetal acciona unos contactos auxiliares retardados (NC, 95-96; NO, 97-98) **que deben provocar la desconexión del motor eléctrico.**

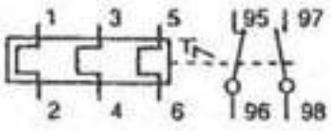


Fig. 4.2. Relé térmico de protección de un motor

4.2.3. Detectores de posición

Finales de carrera

Un interruptor o sensor final de carrera es un interruptor de posición que se acciona de forma automática cuando un elemento móvil (por ejemplo, una pieza situada sobre una cinta transportadora) llega al final de su recorrido. Son muy usados en la industria y en otras aplicaciones (ascensores, montacargas, robots, etc.)

Fig. 4.3. Ejemplos de finales de carrera

Generalmente están compuestos de dos partes, un cuerpo donde se sitúan los contactos auxiliares (NC+NO) y una cabeza de formas muy variadas que detecta el movimiento. Se tienen, por ejemplo, los de lengüeta, bisagra, palanca con rodillo, varilla, palanca metálica con muelle, de pulsador, etc.

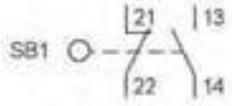


Fig. 4.4. Final de carrera dotado de contactos auxiliares (NC+NO)

Detectores de proximidad magnéticos

Los **detectores magnéticos** se emplean para la detección sin contacto y sin desgaste de la posición de un objeto (por ejemplo, un bote), que debe tener un imán o estar imantado, ya que solo ante este puede reaccionar el sensor. Como el campo magnético es capaz de atravesar todos los materiales no magnetizables,

los sensores son capaces de detectar imanes u objetos imantados a través de superficies de metal no ferroso, de acero inoxidable, de aluminio, de plástico o de madera. Son habituales alcances de hasta 70 milímetros.

Los detectores magnéticos pueden ser de 2 hilos (con contacto auxiliar abierto o cerrado) y de 3 hilos (con contacto auxiliar abierto o cerrado) con salida PNP o NPN a relé de continua de 24 V DC (marrón: positivo, azul: negativo, negro: contacto auxiliar abierto). Se utilizan habitualmente en los cilindros neumáticos.

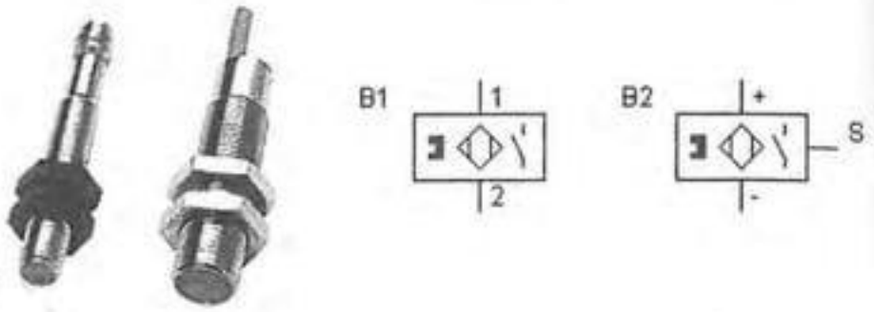


Fig. 4.5.
Detector magnético a 2 o 3 hilos

Detector de proximidad inductivo

El detector de proximidad inductivo utiliza como elemento detector una bobina concéntrica, en la que se genera un campo magnético B alterno. Al acercar un objeto metálico a la bobina, se produce una alteración de las líneas de campo magnético.

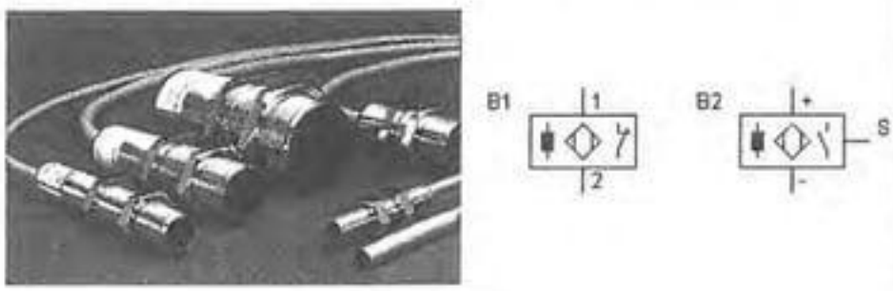
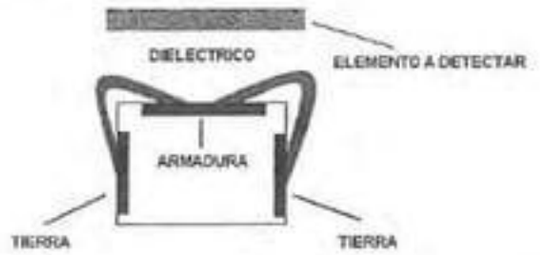


Fig. 4.6.
Detectores de proximidad inductivos a 2 o 3 hilos

Detector de proximidad capacitivo

El detector de proximidad inductivo utiliza como elemento detector un electrodo (armadura metálica) en lugar de una bobina. Este electrodo puede considerarse como una de las armaduras del condensador que forma el sensor y la tierra. Dependiendo del objeto a detectar, se cambia el dieléctrico del condensador (detección de casi cualquier material).



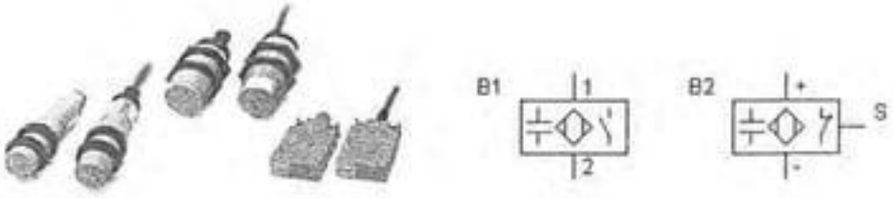


Fig. 4.7.
Detectores de proximidad capacitivos a 2 o 3 hilos

Recuerda • • •

El detector de proximidad magnético detecta imanes o materiales imantados. El detector de proximidad inductivo detecta materiales férricos. El detector de proximidad capacitivo detecta todo tipo de materiales.

Conexionado y simbología

El color asignado por el fabricante a cada hilo del detector, junto con el esquema de conexionado, permite su correcta conexión como entradas al autómata programable utilizado. Es habitual utilizar el código de colores alemán.

Tensión Alterna	Tensión continua	Alemán	Japonés	Italiano
Fase	Positivo	Marrón	Rojo	Marrón
Neutro	Negativo	Azul	Negro	Blanco
Salida		Negro	Blanco	Verde

Fig. 4.8. Código de colores para detectores

El conexionado y la simbología del detector dependerá del tipo de alimentación requerido (continua; alterna), del valor de tensión necesario (24V; 230 V), del número de hilos del detector (2 o 3 hilos, aunque hay detectores con 4 o 5 hilos) y del tipo de transistor utilizado (PNP o NPN) en el detector.

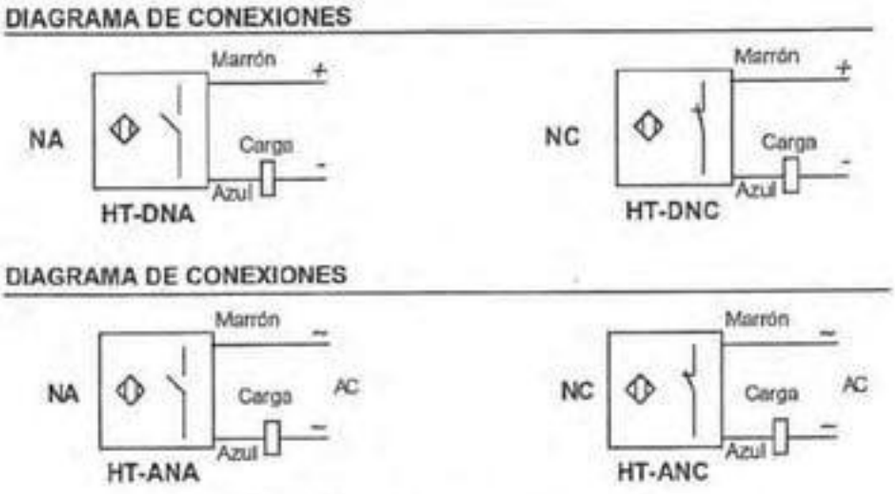


Fig. 4.9. Detectores a 2 hilos DC y AC

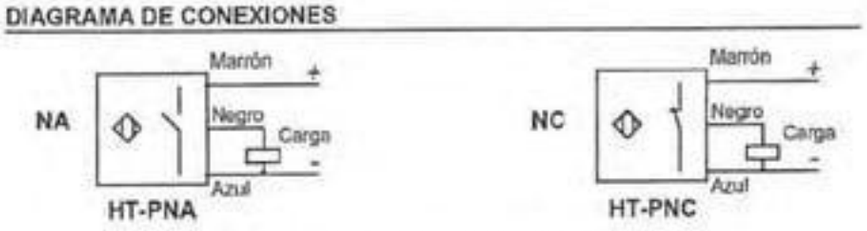


Fig. 4.10. Detectores a 3 hilos tipo PNP

Los detectores a 3 hilos PNP están formados por un elemento detector, una resistencia y un transistor PNP. La alimentación positiva se debe conectar al emisor E del transistor mientras la salida del detector al autómata parte del colector C del transistor.

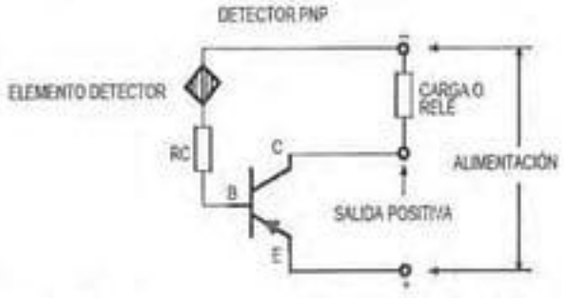


Fig. 4.11. Esquema de un detector a 3 hilos tipo PNP

DIAGRAMA DE CONEXIONES

Fig. 4.12.
Detectores a 3 hilos tipo NPN

Los detectores a 3 hilos NPN están formados por un elemento detector, una resistencia y un transistor NPN. La alimentación negativa se debe conectar al emisor E del transistor mientras la salida del detector al autómata parte del colector C del transistor.

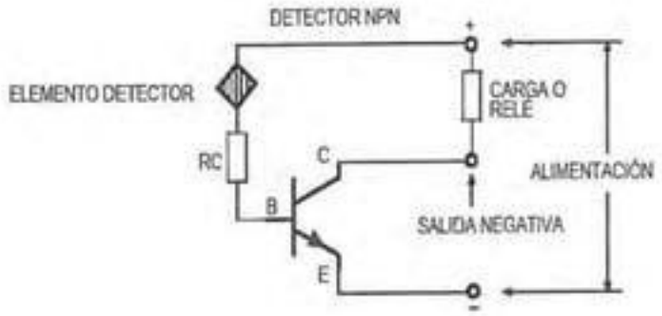
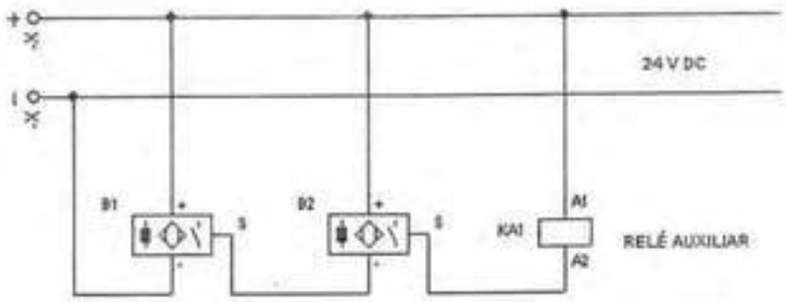


Fig. 4.13.
Esquema de un detector a 3 hilos tipo NPN

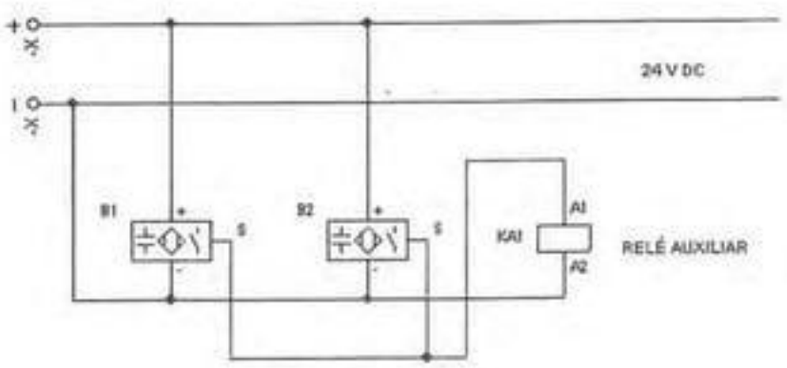
Ejemplo 4.1

Realiza la conexión serie (AND) de 2 detectores inductivos a 3 hilos tipo NPN. La salida se debe llevar a un relé auxiliar KA1.



Ejemplo 4.2

Realiza la conexión paralelo (OR) de 2 detectores capacitivos a 3 hilos tipo PNP. La salida se debe llevar a un relé auxiliar KA1.



Detector fotoeléctrico a 4 hilos

Los detectores o sensores fotoeléctricos son elementos muy utilizados en automatizaciones, junto con otros tipos de detectores (inductivos, capacitivos y magnéticos) y finales de carrera. Sirven para detectar el paso de una pieza u objeto que se mueve sobre una cinta transportadora por un determinado punto (por ejemplo, un bote o una caja). Gracias a ellos, se puede realizar la parada del motor asíncrono que acciona la cinta transportadora para permitir el llenado del bote o la caja.

Los detectores fotoeléctricos a 4 hilos detectan el objeto por reflexión difusa (sin necesidad de receptor del haz), son de cuerpo cilíndrico plástico, con ajuste de sensibilidad y alcance determinado. La salida del detector es de tipo PNP, con 2 contactos seleccionables (NO o NC). Requieren alimentación en continua 10-30 V.

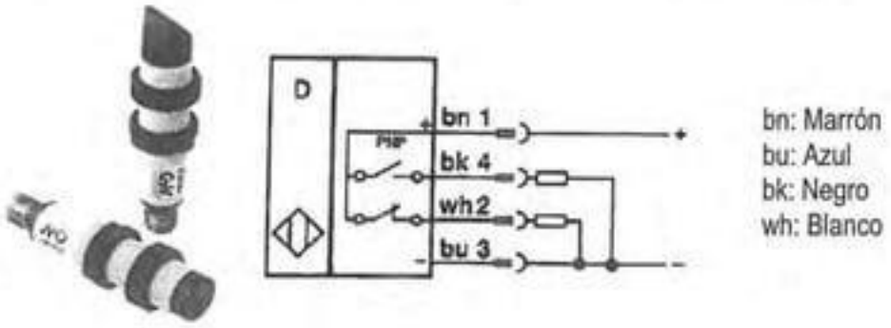


Fig. 4.14.
Esquema de detectores a 4 hilos tipo PNP

Detector de barrera a 5 hilos

Los detectores de barrera permiten la supervisión de puntos de cierre de ascensores y puertas industriales automáticas, el radio de giro de portones automáticos y la detección y seguimiento de objetos durante la manipulación de materiales.

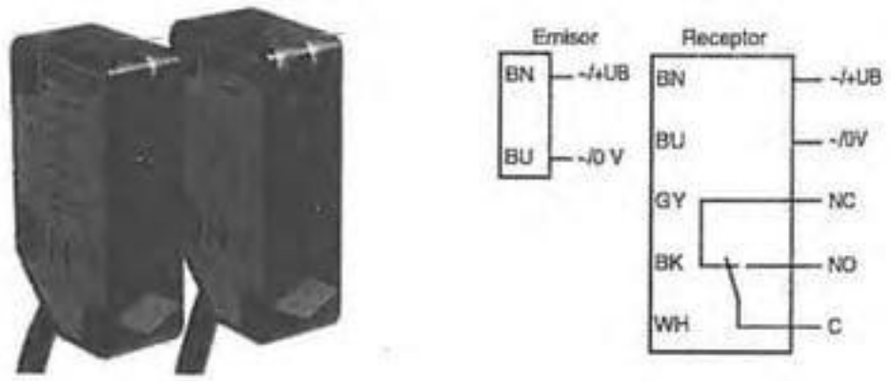


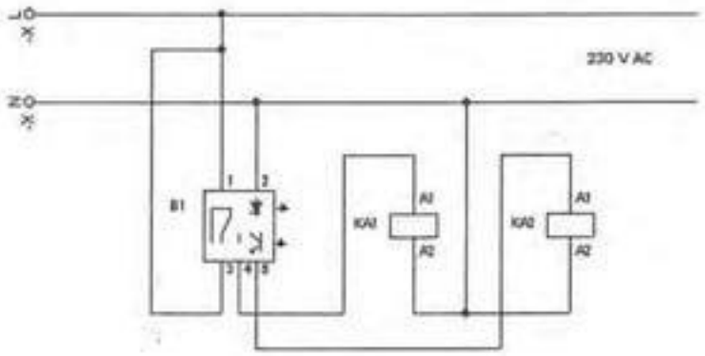
Fig. 4.15.
Esquema de un detector de barrera E-R a 5 hilos

Requieren de un emisor E de luz alterna roja y de un receptor R a 5 hilos en donde 2 hilos se utilizan para la alimentación y los otros 3 hilos restantes corresponden a un juego de contacto conmutado (NC, común, NO). Hay modelos donde tanto el emisor como el receptor están integrados en la misma carcasa y solo se precisa un reflector o catadióptrico.

Fig. 4.16.
Detector de barrera con catadióptrico a 5 hilos

Ejemplo 4.3

Explica el esquema de conexión de un detector de barrera réflex a 5 hilos junto con 2 relés auxiliares KA1 y KA2.



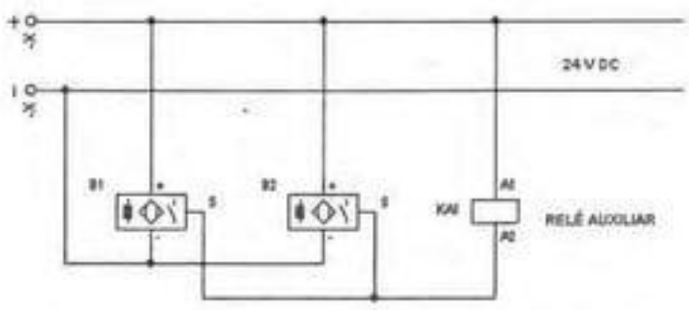
Se ha conectado un relé auxiliar KA1 que tiene tensión sin la presencia de objetos en el detector de barrera fotoeléctrico réflex. Ello es debido a que se lleva la fase al contacto común 3 y se pasa la misma al relé KA1 desde el contacto NC 4.

Se ha conectado en paralelo otro relé auxiliar KA2 que requiere la presencia de objetos en el detector B1 para dar tensión a su bobina. Ello es debido a que se lleva la fase al contacto común 3 y se pasa al relé KA2 desde el contacto NO 5.

Actividades propuestas

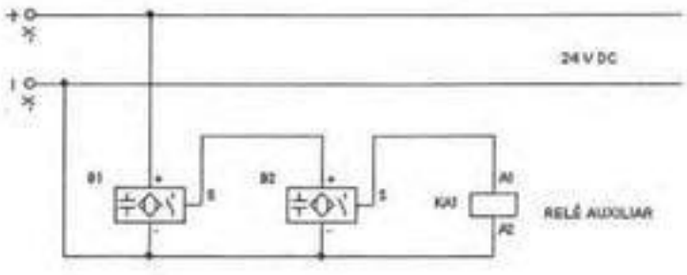
1. Realiza la conexión paralelo (OR) de 2 detectores inductivos a 3 hilos tipo NPN. La salida se debe llevar a un relé auxiliar KA1.

   Solución:



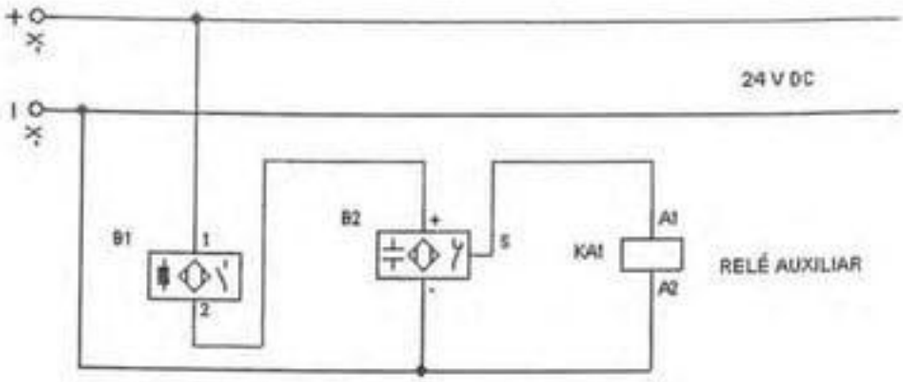
2. Realiza la conexión serie (AND) de 2 detectores capacitivos a 3 hilos tipo PNP. La salida se debe llevar a un relé auxiliar KA1.

   Solución:

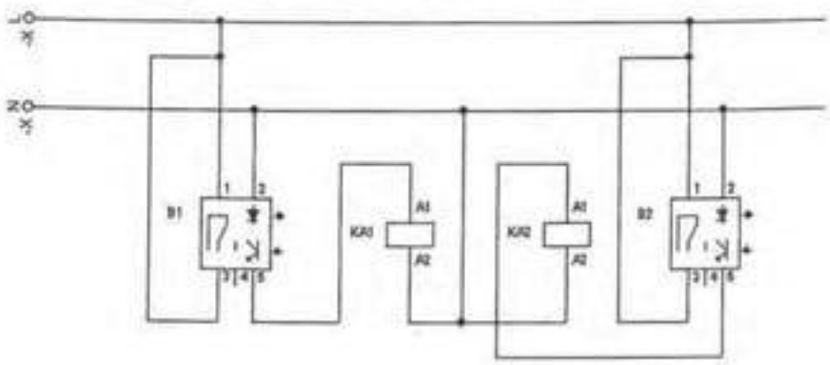


3. Realiza el esquema de conexión de 2 detectores de forma que si el detector 1 detecta presencia de objeto se active el relé auxiliar KA1 y si, además, en el mismo punto el detector 2 detecta también el objeto se desactive el relé auxiliar KA1. El detector 1 es magnético a 2 hilos DC y el detector 2 es capacitivo PNP a 3 hilos a 24 VDC.

Solución:



4.Explica el funcionamiento del siguiente circuito formado por detectores de barrera réflex y relés auxiliares.



Solución:

La presencia de un objeto en B1 activa el relé auxiliar KA1, mientras cuando dicho objeto alcance el detector B2 se activará el relé auxiliar KA2.

4.3. Entradas analógicas a los autómatas programables

Una señal analógica es aquella cuyo valor, que representa a una magnitud física como la velocidad, nivel, volumen, presión, temperatura..., evoluciona variando en todo momento en el tiempo.

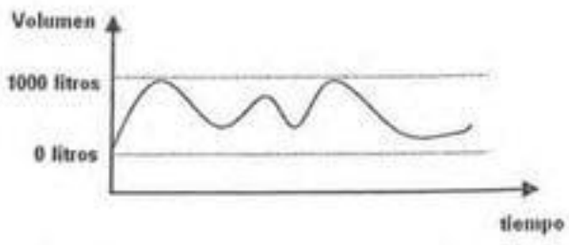


Fig. 4.17. Entrada analógica de volumen

Los elementos que intervienen en el tratamiento de la entrada analógica de un proceso industrial por parte de un autómata programable son:

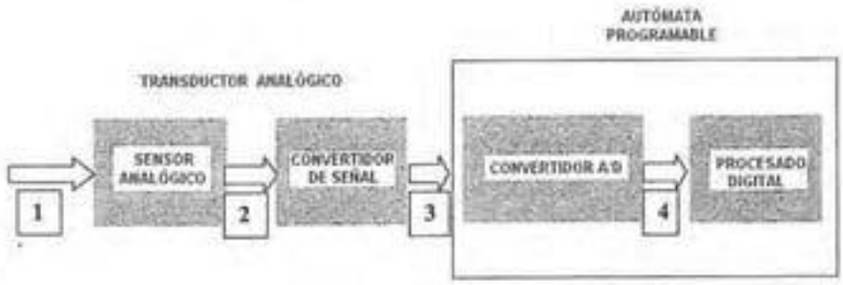


Fig. 4.18.
Tratamiento de una entrada analógica

donde:

1. Magnitud física (presión, nivel, temperatura...)
2. Señal analógica eléctrica (tensión, intensidad o resistencia)
3. Señal analógica eléctrica normalizada (tensión o intensidad)
4. Señal digital cuantificada y codificada

Un sensor es un dispositivo que detecta las modificaciones que sufre una magnitud física (por ejemplo: presión, temperatura, humedad, etc.). Las variaciones detectadas se convierten en señales eléctricas variables como pueden ser resistencia, tensión o intensidad de corriente.

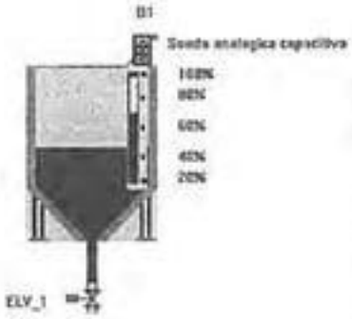


Fig. 4.19.
Entrada analógica de nivel en un depósito

En el caso de un depósito, se desea controlar el nivel de líquido existente (entrada analógica del plc). Si se instala en la parte superior del depósito un medidor de nivel analógico (sonda capacitiva), la magnitud física variará entre una distancia mínima (por ejemplo, 50 mm que corresponde a 1.000 litros) y una distancia máxima (por ejemplo, 500 mm que corresponde a 0 litros).

El transductor analógico de nivel proporciona una señal eléctrica analógica normalizada en tensión (de 0 a 10 V DC) o en intensidad (de 4 a 20 mA DC). Dependiendo del valor de nivel, la señal normalizada al autómata puede incrementarse o reducirse en la misma medida que lo haga el nivel del depósito.

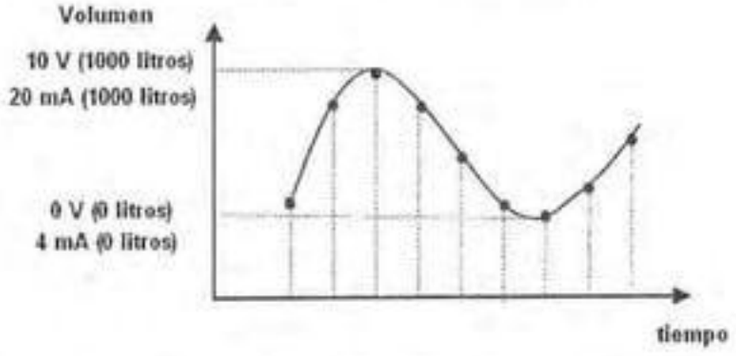


Fig. 4.20.
Entrada analógica muestreada de nivel en un depósito

Para trabajar con medidas analógicas en un autómata, se han de convertir los valores analógicos de tensión o intensidad en información digital mediante un convertidor analógico-digital (convertidor A/D), que en cada ciclo muestrea un valor y después lo traduce a un bit. Cuantos más bits, más definición y precisión.

El convertidor A/D realiza la conversión de la entrada muestreada en digital en 2 etapas que son la cuantificación de la señal y su posterior codificación:

- La cuantificación de la señal consiste en representar la amplitud de la señal normalizada (tensión o intensidad) en un número finito de valores distintos en instantes determinados. Si el convertidor A/D es de n bits, hay 2^n valores o estados posibles.
- La codificación es la representación del valor asignado a la señal normalizada mediante un número binario de los posibles según el número de bits de resolución del convertidor A/D.

Esto significa que, por ejemplo, un valor de 4,25 V medido como entrada analógica se debe convertir y guardar en el autómata en una posición binaria como informa-

ción. Cuanto mayor sea el número de bits o posiciones binarias disponibles para la representación digital, más precisa será la resolución.

Si se tiene una resolución de 3 bits en la entrada analógica para el rango de tensión de 0 a 10 V DC, este rango tan solo se puede dividir en 8 estados posibles ($2^3 = 8$). Esto solo permite considerar valores de 0 a 1,25 V; de 1,25 a 2,5 V; de 2,5 a 3,75 V; de 3,75 a 5 V; de 5 a 6,25 V; de 6,25 a 7,5 V; de 7,5 a 8,75 V y de 8,75 a 10V (valores binarios 000, 001, 010, 011, 100, 101, 110 y 111).

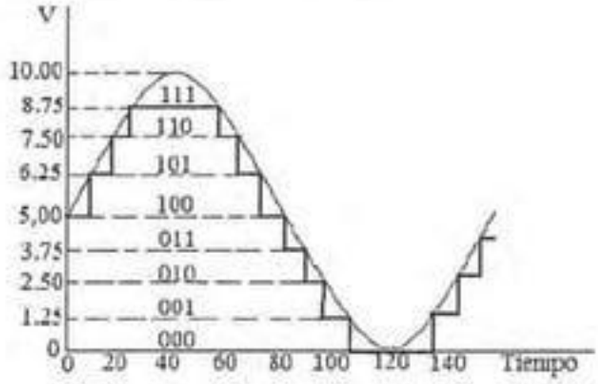


Fig. 4.21. Entrada analógica codificada de nivel en un depósito

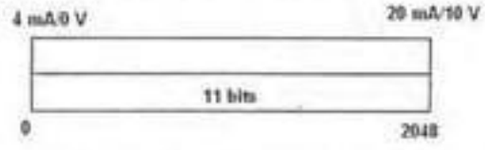
Resolución de la entrada analógica (tensión) = 10 V/8 = 1,25 V;

Resolución de la entrada analógica (intensidad) = 16 mA/8 = 2 mA;

Los convertidores A/D en las entradas analógicas de los autómatas LOGO! y S7-1200 realizan la conversión con resoluciones de 8 y 10 bits, respectivamente. Para una resolución de 8 bits se puede dividir la señal normalizada en 256 partes ($2^8 = 256$); en el caso de 11 bits de resolución, la señal normalizada se puede dividir en 2048 partes ($2^{11} = 2048$).

Resolución de la entrada analógica (tensión) = 10 V/2048 = 0,00488 V;

Resolución de la entrada analógica (intensidad) = 16 mA/2048 = 0,0078 mA;



Con la resolución de 11 bits, se pueden distinguir valores diferentes en tensión menores de 5 mV y en intensidad menores de 8 µA.

Así pues, lo más interesante es tener una mayor resolución en las entradas analógicas del autómata pues mayor será la exactitud de la lectura y menor el error entre el valor real y el almacenado en el autómata. Por contra, también será mayor el tiempo de conversión A/D y, por tanto, las variaciones en el proceso tardarán más en reflejarse en el autómata.

Si bien se pueden encontrar en el mercado, como mínimo, tantos transductores o entradas analógicas como magnitudes físicas se desean medir, es interesante centrarse en aquellas más utilizadas como son los transductores de nivel, los de presión y los de temperatura.

4.3.1. Transductores de nivel

Proximidad por ultrasonidos

Es un transductor de nivel por ultrasonidos de formato cilíndrico que permite la detección de cualquier objeto sin contacto sea cual sea el material (metal, plástico, madera, cartón...), la naturaleza (sólido, líquido, polvo...), el color, el grado de transparencia, etc. Permiten una salida analógica normalizada en tensión (0-10 V) o en intensidad (4-20 mA). Tienen un alcance de trabajo Sn, según modelo y fabricante.

Recuerda • • •

Los detectores pueden ser:

- A 2 hilos, carga en serie.
- A 3 hilos, tipo NPN o PNP, con un solo contacto auxiliar.
- A 4 hilos, tipo NPN o PNP, con contactos auxiliares NO y NC.
- A 5 hilos, con contacto conmutado.

Los detectores fotoeléctricos pueden ser:

- Réflex, polarizados o no.
- De barrera.
- De proximidad.

La entrada de una señal analógica en un PLC se trata a través de un convertidor A/D.