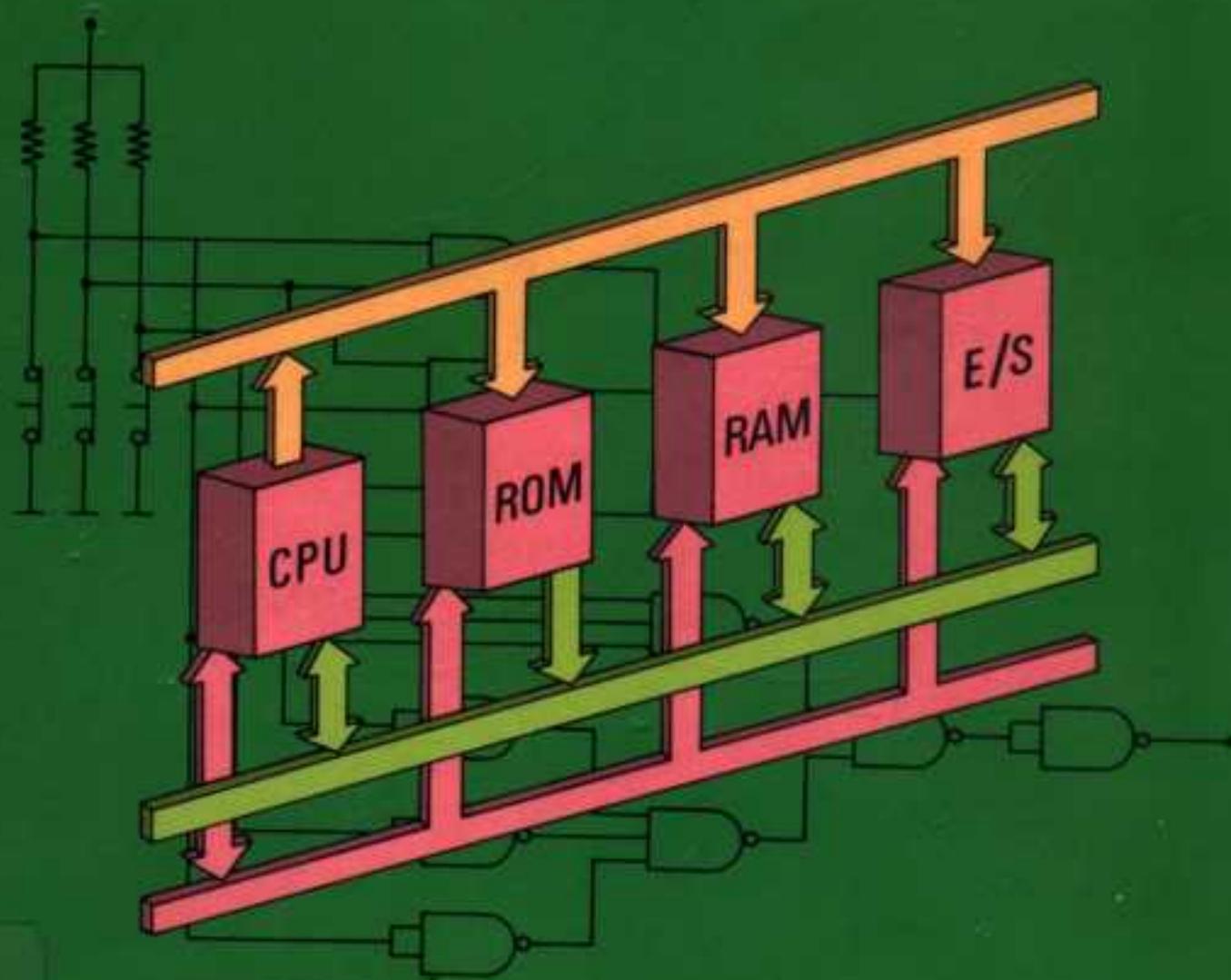


ELECTRONICA GENERAL

1. Dispositivos y sistemas digitales

Antonio J. Gil Padilla



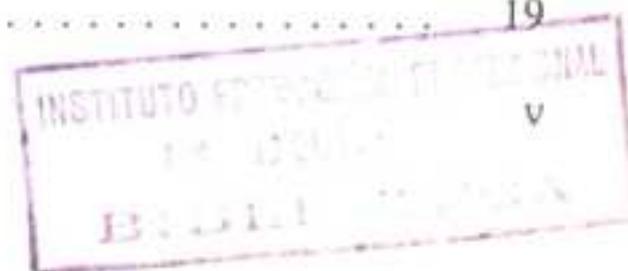
Contenido

Prólogo	xi
---------------	----

Parte Primera

GENERALIDADES Y PUERTAS LOGICAS

Capítulo 1. Naturaleza de los circuitos digitales	3
1.1. Introducción	3
1.2. Electrónica digital	4
1.3. Señales analógicas y digitales	6
1.4. Escalas de integración	7
1.5. Aplicaciones de la electrónica digital	8
Ejercicios propuestos	9
Capítulo 2. Algebra de Boole	10
2.1. El álgebra de Boole y los circuitos digitales	10
2.2. Noción de función lógica o booleana	11
2.3. Tabla de la verdad de una función lógica	12
2.4. Funciones básicas booleanas	13
2.4.1. Función igualdad	13
2.4.2. Función unión	14
2.4.3. Función intersección	14
2.4.4. Función negación	15
2.5. Otras funciones básicas importantes	15
2.6. Postulados, propiedades y teoremas más importantes del álgebra de Boole	15
2.6.1. Postulados	16
2.6.2. Propiedades	19
2.6.3. Teoremas	19



2.7. Forma canónica de una función booleana	20
2.8. Forma de obtener la función lógica a partir de la tabla de la verdad .	20
Ejercicios resueltos	22
Ejercicios propuestos	25
Capítulo 3. Simplificación de funciones	27
3.1. Introducción	27
3.2. Simplificación por el método algebraico	27
3.3. Paso de una función cualquiera a forma canónica	28
3.4. Método gráfico de Karnaugh	30
3.4.1. Método de Karnaugh para más de cuatro variables	33
3.5. Método numérico de Quine-McCluskey	33
Ejercicios resueltos	39
Ejercicios propuestos	45
Capítulo 4. Puertas lógicas	46
4.1. Introducción	46
4.2. Simbología lógica	46
4.3. Constitución de los circuitos integrados que contienen puertas lógicas	47
4.4. Las funciones NAND y NOR como funciones universales	48
4.4.1. Implementación de funciones mediante puertas NAND	49
4.4.2. Implementación de funciones mediante puertas NOR	50
4.5. Características generales de las puertas integradas	51
4.5.1. Familia lógica TTL	51
4.5.2. Familia lógica CMOS	53
4.5.3. Estudio comparativo de las familias TTL y CMOS	54
4.6. Lógica positiva y lógica negativa	55
Ejercicios resueltos	57
Ejercicios propuestos	61
Apéndice: Hojas de características de circuitos integrados de las diferentes tecnologías analizadas: TTL y CMOS	62

Parte Segunda

CIRCUITOS DIGITALES MSI

Capítulo 5. Circuitos combinacionales	73
5.1. Características de los circuitos combinacionales	73
5.2. Codificación y decodificación	73
5.2.1. Sistema binario	74
5.2.2. Códigos	75

5.2.3. Codificadores	81
5.2.3.1. Codificador 74LS148	82
5.2.4. Decodificadores	84
5.2.4.1. Decodificador 74LS42	85
5.3. Multiplexadores y demultiplexadores	85
5.3.1. Multiplexador 74LS151	88
5.4. Comparadores	89
5.4.1. Comparador 7485	90
5.5. Otros circuitos combinacionales	92
Ejercicios resueltos	92
Ejercicios propuestos	95

Capítulo 6. Circuitos operativos	97
6.1. Operaciones matemáticas con circuitos digitales	97
6.2. Suma y resta binaria	97
6.3. Suma y resta con el código BCD natural	100
6.4. Suma y resta en BCD exceso tres	103
6.5. Circuitos sumadores	104
6.5.1. Sumadores y restadores binarios	105
6.5.2. Sumadores y restadores BCD natural	107
6.5.3. Sumadores y restadores BCD exceso tres	109
6.6. Sumador total integrado 7483	112
Ejercicios resueltos	114
Ejercicios propuestos	117

Capítulo 7. Circuitos secuenciales	119
7.1. Definición, características y constitución de los circuitos secuenciales.	119
7.2. Biestables	121
7.2.1. Biestables asíncronos	121
7.2.1.1. Biestable <i>R-S</i> asíncrono	122
7.2.1.2. Biestable <i>J-K</i> asíncrono	123
7.2.1.3. Biestable <i>T</i>	124
7.2.2. Biestables síncronos activados por nivel	125
7.2.2.1. Biestable <i>R-S</i> síncrono activado por nivel	125
7.2.2.2. Biestable <i>J-K</i> síncrono activado por nivel	126
7.2.2.3. Biestable <i>D</i> activado por nivel	126
7.2.3. Biestables síncronos activados por flanco	127
7.2.3.1. Biestable <i>D</i> activado por flanco	128
7.2.3.2. Biestable <i>J-K</i> activado por flanco	129
7.2.3.3. Biestable <i>T</i> activado por flanco	131
7.2.4. Simbología utilizada en este tipo de circuitos	132
7.3. Contadores	132

7.3.1. Contador asíncrono binario 7493	133
7.3.2. Contador síncrono binario 74163	135
7.4. Registros de desplazamiento	139
7.4.1. Registro entrada serie, salida serie	139
7.4.2. Registro de desplazamiento universal 74194	140
Ejercicios resueltos	144
Ejercicios propuestos	151

Parte Tercera

CIRCUITOS DIGITALES LSI PROGRAMABLES

Capítulo 8. Introducción a la lógica programada	155
8.1. El impacto de los circuitos LSI en el diseño de sistemas digitales ...	155
8.2. Tipos de circuitos LSI	157
8.3. Flexibilidad y programabilidad de los dispositivos LSI	158
8.4. Estructura básica de un sistema con microprocesador	160
8.4.1. Buses de conexión y puertas de tres estados	161
8.5. Diseño de sistemas basados en un microprocesador	163
8.6. Descripción del funcionamiento de un sistema digital programable .	164
Ejercicios propuestos	166
Capítulo 9. Memorias	167
9.1. Generalidades	167
9.2. Las características más significativas de las memorias	168
9.3. Capacidad de una memoria	168
9.4. Sistema de numeración hexadecimal	170
9.5. Clases de memorias	171
9.6. Configuración externa de una memoria	174
9.7. Selección de cada uno de los circuitos que constituyen la memoria de un sistema	175
9.8. Organización interna de una memoria	176
9.9. Circuito 2114A de INTEL	179
Ejercicios resueltos	181
Ejercicios propuestos	182
Capítulo 10. Microprocesadores	183
10.1. Introducción	183
10.2. Arquitectura de un microprocesador de 8 bits	184
10.3. Nomenclatura de los terminales del 8085	184
10.4. Arquitectura del microprocesador 8085	187
10.5. Instrucciones del microprocesador 8085	192

10.5.1. Código máquina y nemónicos	192
10.5.2. Tipos de instrucciones	193
10.5.3. Formato de instrucciones	193
10.5.4. Secuencias y tiempos del 8085. Ciclo de instrucción, ciclo de máquina y estado	194
10.5.5. Modos de direccionamiento	199
10.6. Ejemplos de síntesis	199
Ejercicios propuestos	203
Apéndice: Relación completa de las instrucciones del 8085	205
Capítulo 11. Transferencia de datos	221
11.1. Introducción	221
11.2. Instrucciones de transferencia de datos	221
11.2.1. Instrucciones que implican exclusivamente a los registros generales del microprocesador	222
11.2.2. Instrucciones que implican exclusivamente a posiciones de memoria	223
11.2.3. Transferencia entre registros del microprocesador y posiciones de memoria	224
11.2.4. Transferencia entre registros del microprocesador y registros de dispositivos de E/S	225
11.3. Selección de dispositivos de E/S	226
11.4. Puertos de E/S	228
11.4.1. Dispositivo de interconexión periférica programable 8255	233
Ejercicios resueltos	241
Ejercicios propuestos	244
Capítulo 12. Bifurcaciones	245
12.1. Estructuras del tipo secuencia y bifurcaciones	245
12.2. Diagramas de flujo	246
12.3. Instrucciones de salto	247
12.4. Subrutinas	249
12.4.1. Uso de etiquetas en la escritura de programas	250
12.5. Instrucciones de llamada y retorno	251
12.6. El stack, el stack pointer y las instrucciones de stack	252
12.7. Anidamiento de subrutinas	254
12.8. Programación estructurada	254
Ejercicios resueltos	256
Ejercicios propuestos	258
Capítulo 13. Interrupciones y acceso directo a memoria	260
13.1. Concepto y tipos de interrupciones	260
13.2. Las interrupciones del 8085	262

13.2.1. Instrucciones que manejan el sistema de interrupciones . . .	263
13.2.2. Interrupciones provocadas por las entradas RST y TRAP . . .	265
13.2.3. Interrupciones provocadas por la entrada INTR	266
13.3. Prioridad de las interrupciones	268
13.3.1. Controladores de prioridad	269
13.4. Acceso directo a memoria	272
Ejercicios resueltos	273
Ejercicios propuestos	275
Capítulo 14. Operaciones lógicas y aritméticas	276
14.1. Operaciones lógicas	276
14.2. Instrucciones que efectúan operaciones lógicas	276
14.2.1. Instrucciones que efectúan la función AND	277
14.2.2. Instrucciones que efectúan la función OR	278
14.2.3. Instrucciones que efectúan la función O exclusiva	279
14.2.4. Instrucciones que efectúan el complemento	280
14.2.5. Instrucciones que efectúan comparación	280
14.2.6. Instrucciones que efectúan rotación	281
14.3. Operaciones aritméticas	282
14.4. Instrucciones de suma y resta	283
14.4.1. Instrucciones de suma	283
14.4.2. Instrucciones de resta	285
14.5. Otras instrucciones que efectúan operaciones aritméticas	285
Ejercicios resueltos	286
Ejercicios propuestos	289
Glosario	290
Vocabulario de términos en inglés	297
Bibliografía	298
Índice temático	299

Naturaleza de los circuitos digitales

1.1. INTRODUCCION

Dedicaremos la obra, prácticamente en su totalidad, al análisis de las técnicas y dispositivos digitales, así como a sus aplicaciones, pero antes hemos optado por empezar abordando este primer capítulo, de carácter netamente descriptivo, con el fin de situar al lector en esta importante área de los circuitos digitales dentro del amplio campo de la Electrónica General. Pretendemos caracterizar la **Electrónica Digital**, conocida también como *lógica digital* o *circuitos lógicos*, diferenciándola de la **Electrónica Analógica**, así como justificar el enorme crecimiento que se ha producido en el empleo de los componentes digitales. También examinaremos las etapas que se han cubierto hasta llegar a la actual escala de integración. Por último, expondremos cuáles son las aplicaciones más significativas de la Electrónica Digital.

Es necesario, desde el principio, clasificar y diferenciar algunos términos que se repetirán en éste y en siguientes capítulos y que están íntimamente relacionados a vocablos que se utilizan para definir circuitos o conjuntos de elementos electrónicos de diferente orden de magnitud. No es lo mismo hablar de un **sistema** que de un **dispositivo** o de un simple **componente** o **elemento**. El objetivo no es establecer con carácter general una serie de definiciones de índole estrictamente formal, sino, como decíamos anteriormente, diferenciar unos conceptos de otros, conociendo con exactitud a qué nos estamos refiriendo cada vez que utilicemos alguna de estas expresiones a lo largo de todo el texto.

Un sistema es el circuito o conjunto de mayor magnitud de todos los que describiremos, y se puede definir como la asociación de una serie de dispositivos y/o componentes o elementos, todos ellos interrelacionados y entre los que existe una cierta cohesión y unidad de propósito. En todo sistema automático o procesamiento de señal eléctrica es posible apreciar una sucesión de etapas, tales como los sensores o transductores, el control o procesador y la salida o etapa de potencia. Cada una de estas etapas están constituidas por uno o varios dispositivos.

Puede ocurrir que en una misma instalación, máquina o equipo puedan coexistir, junto a un sistema eléctrico-electrónico, otros de naturaleza mecánica y/o

neumohidráulica. La integración de todos ellos da lugar a un sistema de orden superior. Ejemplos de pequeños sistemas electrónicos son: un reloj, un termómetro, un voltímetro, etc.

Un **dispositivo** es un circuito constituido por varios componentes que realiza una operación o materializa una función electrónica por sí mismo. Son ejemplos de dispositivos digitales, un biestable, un contador, un codificador, una etapa de salida, etc.

Un **componente** o **elemento** es cada una de las partes que constituyen un dispositivo. Así un transistor, un diodo, una resistencia, etc., son claros ejemplos de los componentes más simples que se emplean para construir un dispositivo.

Los progresos que se han producido, que se producen en la actualidad y que, presumiblemente, se seguirán produciendo en los procesos de integración de componentes en un solo **chip** (*pastilla*), producen algunas distorsiones en el planteamiento que hemos realizado anteriormente. Un dispositivo puede ser integrado en su totalidad y convertirse en un componente de otro dispositivo de mayor complejidad o, directamente, en un componente de un sistema. Un contador digital puede construirse con varios biestables, pero también se encuentran disponibles de forma integrada en un solo circuito.

La densidad de integración, como veremos más adelante, es hoy día tan elevada que es posible integrar en un solo chip un sistema electrónico en su totalidad. Un ejemplo de esta última afirmación, en grado extremo, lo constituyen los **microcontroladores**, que en una sola pastilla reúnen dispositivos de elevada complejidad, tales como un *microprocesador*, una *memoria* y *unidades de entrada/salida*, para comunicarse con el exterior. A pesar de la ambigüedad existente en esta materia, la descripción realizada, junto a estas últimas aclaraciones, nos parece válida como esquema de trabajo; rechazamos definiciones, recogidas en parte de la bibliografía especializada, tales como las de sistemas combinatoriales o sistemas secuenciales referidas a circuitos codificadores, contadores, etc., que para nosotros son sencillamente dispositivos y en ningún caso sistemas. Por otra parte, hemos de señalar que los catálogos (Databooks) a veces utilizan *dispositivo* (en inglés, *device*) como término genérico para referirse tanto a circuitos, tales como biestables, codificadores, etc., como a simples componentes: diodos, transistores, etc.

1.2. ELECTRONICA DIGITAL

Lo que hoy día conocemos como Electrónica Digital es el conjunto de determinadas técnicas y dispositivos integrados, de distinto grado de complejidad, que se utilizan fundamentalmente para la construcción de circuitos de control de procesos industriales, de equipos informáticos para procesamiento de datos y, en general, de otros equipos y productos electrónicos.

La Electrónica Digital se ha impuesto a la Electrónica Analógica o, más tradicional, en aquellos casos donde la solución a un problema puede efectuarse de ambas formas. Además, su aplicación ha mejorado sistemas y productos ya

existentes y ha dado lugar al desarrollo de otros nuevos que antes no era posible construir.

La utilización y proliferación de las técnicas y circuitos digitales es debido, en gran medida, a la enorme analogía con nuestras mentes, que utilizan de forma continua la lógica para resolver problemas, tomar decisiones, almacenar conocimientos en nuestra memoria, etc. A todo esto hay que añadir las ventajas que toda la amplia gama de dispositivos digitales disponibles en el mercado de la electrónica ofrece frente a los dispositivos analógicos o lineales. Dichas ventajas se concretan en una mayor inmunidad al ruido eléctrico, elevada densidad de integración, facilidad de acoplamiento de unos bloques con otros, etc.

Las técnicas digitales y los circuitos lógicos son, cronológicamente, anteriores a la aparición y posterior desarrollo de la Electrónica Digital integrada. Su origen se remonta a los tiempos en que surgió la necesidad de construir automatismos, optimizando el número de elementos necesarios. Los primeros circuitos lógicos se construyeron con **relés** electromagnéticos, siendo una de sus primeras aplicaciones las redes telefónicas.

La aparición de las válvulas electrónicas sustituyó a los relés en algunas aplicaciones, reduciendo el tamaño del circuito, y, en ese mismo sentido, la aparición de los **semiconductores** supuso un enorme progreso. Sin embargo, el paso más importante se ha producido con el desarrollo de las técnicas de integración de componentes en una sola pastilla, dando lugar a nuevas y complejas aplicaciones y a la sustitución definitiva de los armarios cargados de relés por modernos y reducidos **controladores programables**.

Para el estudio de la Electrónica Digital y el uso de los dispositivos digitales no se requieren conocimientos previos de otras partes de la electrónica ni del funcionamiento de los componentes básicos (*diodos, transistores, etc.*). Los circuitos integrados digitales pueden ser utilizados como *cajas negras*, donde lo importante es conocer la función que realizan, así como sus características eléctricas, y no su funcionamiento interno ni el número de los elementos que lo constituyen.

Desde la aparición del primer circuito integrado hasta estos momentos, el progreso ha sido enorme. Estos avances han dado lugar al empleo de bloques cada vez más complejos. Así, hoy día, siempre que el precio de los chips lo permita, en la mayoría de las aplicaciones se utiliza **lógica programable** con un reducido número de pastillas.

Sin embargo, el conocimiento de la base lógica matemática, así como de las funciones elementales y de la circuitería de menor complejidad, que analizaremos en los siguientes capítulos, es imprescindible para ir desarrollando progresivamente las habilidades y conocimientos necesarios para abordar la última parte, dedicada a los dispositivos más complejos: microprocesadores, memorias, unidades de entrada/salida, etc.

Además, el estudio de la primera parte es también necesario porque, aunque el sistema se resuelva mediante lógica programable, siempre es necesario el uso de circuitos más simples, tales como puertas, codificadores, biestables, etc., para completar el circuito.

1.3. SEÑALES ANALÓGICAS Y DIGITALES

De manera inevitable, en los apartados anteriores hemos tenido que hacer alguna referencia a términos tales como analógico o digital sin aclarar suficientemente la diferencia entre ambos. Este es el momento de hacer tal distinción.

Una señal analógica, y en general cualquier magnitud analógica, *es aquella que puede tomar infinitos valores a lo largo del tiempo*; dicho en otras palabras, es aquella que cambia de forma continua. Sin embargo, una **señal digital** *es aquella que tiene un número finito de valores definidos y cambia de valor por saltos*. La diferencia está reflejada gráficamente en las Figuras 1.1. y 1.2.

Para comprender mejor la desigualdad entre un fenómeno analógico y uno digital expondremos un ejemplo. Supongamos un estudio de TV con un número determinado de focos, la iluminación máxima se podrá conseguir de dos formas distintas.

En primer lugar, imaginemos que cada foco se activa independientemente con un interruptor. El estudio estará iluminado con distintos niveles de intensidad luminosa en función del *número* de focos encendidos, alcanzándose, lógicamente, la máxima iluminación cuando todos los focos estén conectados.

Por otra parte, es también posible conectar todos los focos a un regulador común, construido, por ejemplo, con un potenciómetro. Mediante este método los focos se iluminarán gradualmente. La intensidad aumentará, en este caso, progresivamente cuando el cursor se desplace desde el valor mínimo hasta el máximo, momento en el cual se conseguirá la iluminación total de la sala.

En el primer caso la intensidad aumenta en saltos discretos, mientras que en el segundo varía de manera continua.

La transformación de una señal analógica en digital y de digital en analógica es posible, y en la práctica se consigue con circuitos denominados **convertidores**. La conversión, además de posible es necesaria, ya que, como hemos indicado anteriormente, la mayor parte de los sistemas, en la actualidad, procesan las señales de forma digital; sin embargo, las variables físicas, tales como la temperatura, la

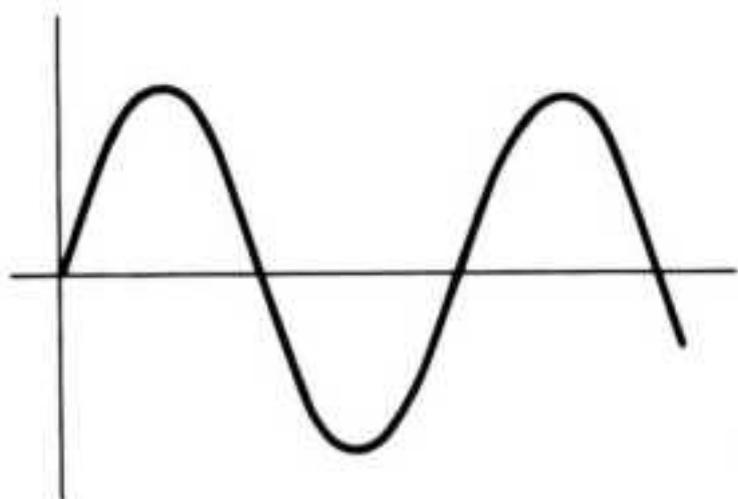


Figura 1.1. Señal analógica.

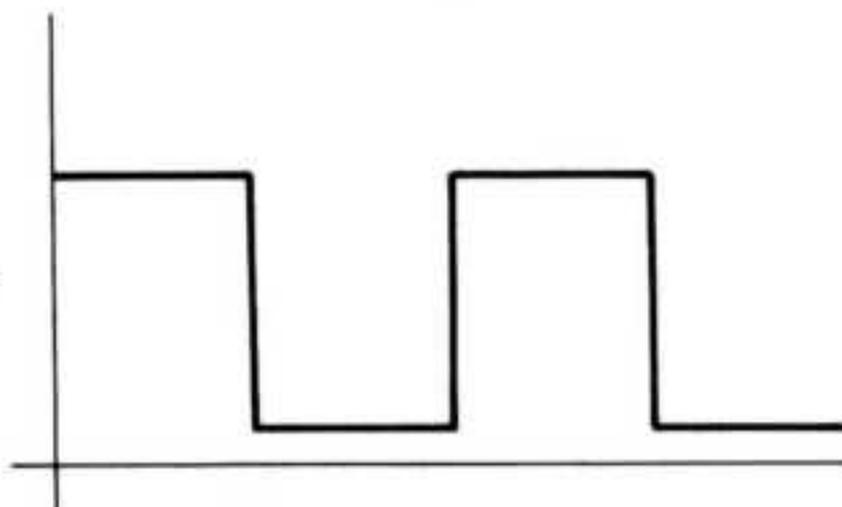


Figura 1.2. Señal digital

presión, la velocidad, etc., y, por tanto, las señales de entrada y salida de dichos sistemas, son generalmente analógicas.

Las señales de tensión y de corriente de los circuitos digitales son extremadamente sencillas, tal como se muestra en la Figura 1.2. Existen solamente dos valores distintos denominados **estados** o **niveles lógicos**. Para diferenciar el uno del otro se utilizan los números *cero* y *uno*.

1.4. ESCALAS DE INTEGRACION

En la actualidad las funciones lógicas se materializan mediante bloques integrados más o menos complejos y con mayor o menor densidad de integración respectivamente.

Uno de los principales objetivos de los fabricantes de componentes electrónicos ha sido la integración en una sola pastilla de distintos elementos. Con ello se ha progresado enormemente en la reducción del tamaño de los circuitos. La rápida escalada en tal reducción ha estado motivada, fundamentalmente, por la aplicación de la electrónica a los grandes proyectos aeroespaciales y militares. De estos progresos se ha beneficiado la industria electrónica en general.

Los motivos que inicialmente impulsaron la integración de componentes fueron la reducción de volumen y de peso. Sin embargo, aparejadas a estas ventajas, mejoraron otras características de gran importancia, tales como la disipación de potencia, las dimensiones de las fuentes de alimentación asociadas, el coste y la fiabilidad.

Desde el punto de vista de la densidad (componentes/mm²), la clasificación de los actuales circuitos integrados disponibles en catálogo, en sus diferentes familias, es la siguiente:

- a) **SSI** (*Small Scale Integration* o *integración a pequeña escala*). En este grupo están comprendidos los circuitos de funciones lógicas elementales y algunos dispositivos algo más complejos, que estudiaremos en próximos capítulos. El número aproximado de componentes por circuito es de 100; el número máximo de puertas lógicas es aproximadamente 10.
- b) **MSI** (*Medium Scale Integration* o *integración a escala media*). Comprende circuitos de aplicación general que realizan funciones lógicas más complejas que las citadas anteriormente. Codificadores, multiplexadores, contadores, etcétera, son buenos ejemplos de circuitos MSI. El número aproximado de componentes por chip está comprendido entre 100 y 1.000. El número máximo de puertas es de, aproximadamente, 100.
- c) **LSI** (*Large Scale Integration* o *integración a gran escala*). Son circuitos que realizan funciones lógicas muy complejas. En este grupo se encuentran los dispositivos propios de la lógica programable: memorias, microprocesadores, etc., y otros más específicos, tales como los empleados en calculadoras. El número de componentes por circuito está comprendido entre 1.000 y 10.000, aproximadamente 1.000 puertas lógicas.

- d) **VLSI** (*Very Large Scale Integration* o *integración a muy gran escala*). Esta es la tecnología de los años 80. (En la actualidad se construyen circuitos con más de 10 millones de componentes, y en el año 2000 la cantidad sobrepasará los 1.000 millones de transistores en pastillas de idéntico tamaño al actual.)

1.5. APLICACIONES DE LA ELECTRONICA DIGITAL

El desarrollo de las técnicas de integración ha sido tal que en la actualidad es posible construir, mediante muy pocos bloques LSI, una **función lógica universal**. El correcto acoplamiento de un microprocesador, una memoria y una unidad de entrada/salida, mediante la adecuada programación, puede efectuar cualquier tipo de operación lógica; y no solamente una, sino que con el programa almacenado en memoria es capaz de realizar un enorme número de ellas en un tiempo relativamente pequeño. Además las ventajas que reporta la utilización de la tecnología LSI (y, por supuesto, de la VLSI) son numerosas. Entre ellas destacan: la disminución del tiempo de diseño del sistema o producto, la disminución del tamaño del circuito, la reducción del consumo de energía, la fiabilidad (debido a la reducción del número de componentes y del cableado), la mayor capacidad en la toma de decisiones y la facilidad de cambiar o aumentar las prestaciones del sistema cambiando exclusivamente el programa de trabajo.

Debido a todas estas razones y al asequible precio de este tipo de componentes, prácticamente la totalidad de los circuitos digitales se construyen utilizando lógica programable mediante el empleo de bloques LSI y VLSI como dispositivos básicos del sistema. La denominada **lógica cableada**, constituida exclusivamente por bloques SSI y MSI, ha quedado prácticamente abandonada.

La Electrónica Digital, como de alguna forma hemos indicado en el apartado 1.2, se aplica a tres amplios y diferentes campos, a saber:

- Sistemas de control industrial.
- Equipos de proceso de datos.
- Otros equipos y productos electrónicos.

Dentro del primer grupo se incluyen todo tipo de automatismos empleados en instalaciones y maquinaria de cualquier clase y en los procesos industriales aplicados a cualquier sector de actividad. El control del sistema se realiza, generalmente, mediante un equipo denominado **controlador** o **autómata programable** que está construido con circuitos LSI y es un dispositivo estándar complejo y modular que emplea un lenguaje de programación muy elemental, propio de cada fabricante.

Ejemplos de sistemas de este tipo son los siguientes: automatización, en general, de líneas de montaje y producción; llenado y empaquetado automático; recuento, prueba, pesaje y clasificación de productos; selección automática de frutos; comprobación de componentes y circuitos electrónicos, etc.

En el grupo de equipos de procesos de datos se incluyen todos los sistemas

empleados, fundamentalmente, para la adquisición, tratamiento y comunicación de datos.

El equipo básico de estos sistemas lo constituye **el ordenador**, cuyo principio de funcionamiento es el mismo que el de los controladores programables anteriormente mencionados, aunque las aplicaciones, así como los dispositivos periféricos a los que se conecta, sean distintos.

Como ejemplos de este grupo indicaremos, en general: equipos de gestión, enseñanza, etc.; control de tráfico; unidades de cuidados intensivos; adquisición y proceso de señales; terminales bancarios y cajeros automáticos; equipos de comunicaciones (transmisión y recepción de datos), etc.

Gracias a la aplicación de dispositivos digitales ha sido posible mejorar considerablemente productos y equipos ya existentes, tanto dentro del grupo de los bienes de consumo como del grupo de los bienes intermedios, y además crear otros que era imposible desarrollar con tecnologías anteriores.

Son claros ejemplos de productos mejorados: toda la gama de electrodomésticos (lavadoras, máquinas de coser, hornos, etc.), los relojes, las máquinas de escribir, los dispositivos de seguridad, las cajas registradoras, las calculadoras, los equipos de navegación y defensa, etc. Como nuevos tipos de productos podemos indicar: órganos musicales, juegos para TV, ordenadores personales, etc.

No queremos terminar este apartado sin hacer especial mención a la aplicación de la Electrónica Digital a los sistemas **CAD/CAM**. Debido a la rapidez y facilidad con que los ordenadores tratan los datos, su campo de aplicación se ha extendido enormemente, incorporándose a la oficina de diseño y a la industria para controlar procesos y cadenas de fabricación. Los sistemas CAD (diseño asistido por ordenador) y CAM (fabricación asistida por ordenador) están formados por una parte física (ordenador, plotters, impresoras, etc.) y unos potentes programas que permiten mejorar y facilitar considerablemente el diseño y controlar y programar la producción.

EJERCICIOS PROPUESTOS

1. Describir hechos donde sea posible diferenciar fenómenos analógicos de digitales.
 2. Establecer una relación de dispositivos digitales (CI) de cada una de las escalas de integración (consultar catálogo).
 3. Presentar una relación de productos, equipos, sistemas y procesos que contengan Electrónica Digital.
-

Algebra de Boole

2.1. EL ALGEBRA DE BOOLE Y LOS CIRCUITOS DIGITALES

George Boole desarrolló en las primeras décadas del siglo XIX el álgebra que lleva su propio nombre para investigar las leyes fundamentales de aquellas operaciones de la mente humana por las que se rigen los razonamientos. En aquellos momentos de ningún modo nadie se podía imaginar hasta qué punto este sistema matemático influiría de manera tan rotunda en el diseño de circuitos electrónicos y, como consecuencia, en el desarrollo de toda la industria.

El *álgebra de Boole*, como el álgebra convencional, tiene, en principio, como objeto definir una serie de símbolos para representar objetos o fenómenos que encadenados convenientemente dan lugar a expresiones matemáticas más complejas, denominadas *funciones*. Posteriormente, deben ser precisadas las leyes que gobiernan tales funciones, así como las relaciones entre ellas, mediante un conjunto de *enunciados, postulados, teoremas, etc.*

Sin embargo, existen marcadas diferencias entre ambos sistemas. Mientras que el álgebra convencional opera con relaciones cuantitativas, el álgebra de Boole lo hace con **relaciones lógicas**. En el primer caso los signos más (+) y por (\times) representan algoritmos de *suma y producto*, respectivamente, mientras que en el álgebra de Boole representan, como tendremos ocasión de ver más adelante, *relaciones lógicas*. Por otra parte, en el álgebra convencional se utilizan expresiones simbólicas tales como x, y, z , etc., denominadas *variables*, para representar cantidades numéricas. Estas variables pueden tomar infinitos valores y, relacionadas mediante los algoritmos propios de este sistema, dan lugar a las funciones, de las cuales interesa saber la magnitud de ciertas variables cuando cambia el valor de otras de las que dependen.

En el álgebra de Boole las variables, denominadas **binarias**, pueden tomar solamente dos valores distintos: **verdadero** o **falso**. Estos dos valores se representan simbólicamente con los signos **1** y **0**, respectivamente. Los signos 1 y 0 no expresan cantidades, sino **estados** de las variables.

Los componentes electrónicos más elementales (diodos, transistores, etc.) y, como consecuencia, todos los bloques lógicos o circuitos digitales, sea cual sea

la escala de integración, así como otros elementos eléctricos, tales como los motores, las lámparas, etc., presentan dos estados estables de funcionamiento: la lámpara puede estar encendida o apagada, el motor girando o parado.

Estas circunstancias hacen del álgebra de Boole o *álgebra lógica*, así como del *sistema de numeración binario*, el soporte matemático ideal para el diseño y análisis de los circuitos electrónicos digitales, denominados de esta forma precisamente por ajustarse sus señales eléctricas, tanto de entrada como de salida, a dicho sistema, que utiliza los **dígitos 1 y 0** como signos de representación.

Todo este aparato matemático hace posible concretar un problema en una o varias expresiones, que podrán ser manipuladas y simplificadas convenientemente para transformarse posteriormente en un circuito formado por el menor número de elementos posibles.

Queremos indicar, por último, que sea cual sea la complejidad de los elementos o dispositivos que constituyan un sistema digital, su funcionamiento siempre responde al esquema indicado anteriormente para los componentes más básicos. El almacenamiento de datos en una memoria, por ejemplo, se realiza mediante ceros y unos, que físicamente se traduce en que los elementos básicos que forman dicha memoria se encuentran en uno de los dos posibles estados de funcionamiento. Lo mismo ocurre con los microprocesadores y demás dispositivos de la misma complejidad. La traducción a datos de significado real se realiza convenientemente con dispositivos electrónicos, alguno de los cuales tendremos ocasión de estudiar en capítulos posteriores.

2.2. NOCION DE FUNCION LOGICA O BOOLEANA

Se define como **función lógica o booleana** toda variable binaria cuyo valor depende de una expresión algebraica formada por otras variables binarias relacionadas mediante los signos $+$ y/o \times . El significado de estos signos no es el que tienen en la aritmética convencional, sino que, como hemos indicado anteriormente, indican relaciones lógicas, de manera que $+$ deberá interpretarse como la conjunción **o**. De la misma forma el signo \times (también se utiliza un punto \cdot) será equivalente a la conjunción **y**.

Como ejemplo de función lógica podemos indicar la siguiente:

$$S = a \cdot b + b \cdot c$$

que, de la misma manera que en álgebra convencional, se puede representar de forma genérica:

$$S = f(a, b, c)$$

La función o variable dependiente es la S y las **variables binarias** son a , b y c , agrupadas en los términos $a \cdot b$ y $b \cdot c$.

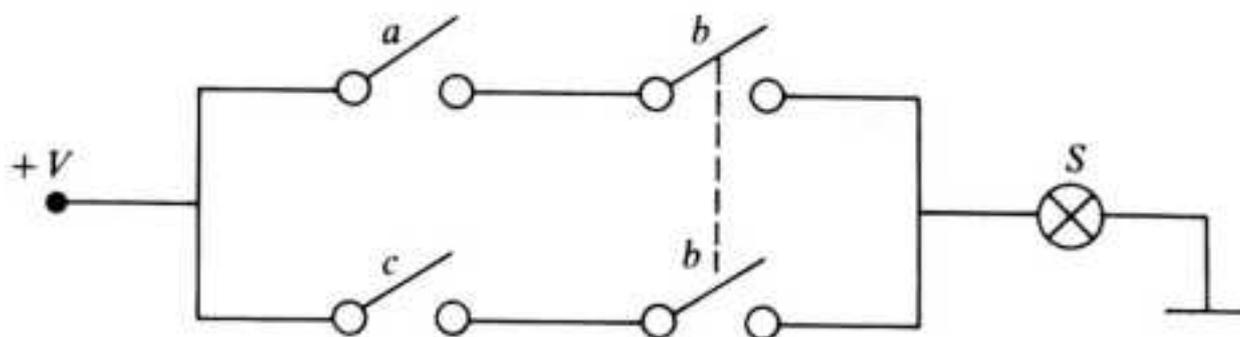


Figura 2.1. Circuito correspondiente a la función $S = a \cdot b + b \cdot c$.

La forma correcta de leer la expresión sería: si a y b o b y c son verdaderas (1), S será verdadera (1). Lógicamente S será también verdadera si lo son ambos términos a la vez.

Para comprobar la relación entre el álgebra de Boole y los circuitos eléctricos-electrónicos, indicaremos que la anterior función lógica recoge en forma de ecuación matemática el siguiente planteamiento: Disponemos de una lámpara (S) y de tres interruptores (a , b , c) y deseamos que la lámpara se ilumine (1) cuando estén cerrados (1) los interruptores a y b , o lo estén b y c . En la Figura 2.1 se muestra el circuito equivalente a la función S que resuelve el problema planteado.

Todas las funciones lógicas están compuestas, generalmente, por funciones lógicas elementales; de ellas nos ocuparemos seguidamente.

2.3. TABLA DE LA VERDAD DE UNA FUNCION LOGICA

Toda función lógica puede ser representada gráficamente mediante una tabla de la verdad.

La **tabla de la verdad**, como se muestra en la Figura 2.2, es un cuadro formado por tantas columnas como variables contenga la función más la correspondiente a ésta y por tantas filas como combinaciones binarias sea posible construir con dichas variables.

a	b	c	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Figura 2.2. Tabla de la verdad de la función $S = a \cdot b + b \cdot c$.

El número de combinaciones posibles será 2^n , siendo n el número de variables. Es conveniente, para evitar repeticiones o confusiones, ordenar las combinaciones binarias de forma creciente.

La tabla de la verdad de la Figura 2.2 corresponde a la función expuesta en el apartado anterior: $S = a \cdot b + b \cdot c$. Como más adelante comprobaremos, en el proceso de diseño el camino a seguir será el contrario, es decir, partiendo de la tabla de la verdad, indicando en ella el estado deseado (1 ó 0) de la función para cada una de las posibles combinaciones de las variables de entrada, obtendremos la función $S = (a, b, c)$, que posteriormente se traducirá en un circuito eléctrico-electrónico.

Es posible construir tablas de la verdad donde existan varias funciones de salida para unas mismas variables de entrada.

Entre la tabla de la verdad y la función que representa existe una *relación biunívoca*, y aquélla es tan importante que figura como uno de los datos más significativos dentro de las características que aparecen en los catálogos de los *dispositivos digitales integrados*. Su conocimiento es fundamental para analizar el funcionamiento y aplicación de *cada bloque*.

2.4. FUNCIONES BASICAS BOOLEANAS

En este apartado describiremos las funciones booleanas o lógicas más elementales, como son: **igualdad**, **unión**, **intersección** y **negación**.

Expondremos para cada una de ellas su expresión matemática, su tabla de la verdad y el circuito que materializa la función, utilizando contactos eléctricos.

2.4.1. FUNCION IGUALDAD

La función igualdad es la más elemental de todas ellas. Interviene exclusivamente una variable. Su expresión matemática es la siguiente:

$$S = a$$

La tabla de la verdad, así como su materialización mediante contactos, se muestra en la Figura 2.3.

$a = 1$ (contacto cerrado)
 $a = 0$ (contacto abierto)
 $S = 1$ (lámpara encendida)
 $S = 0$ (lámpara apagada)

a	S
0	0
1	1

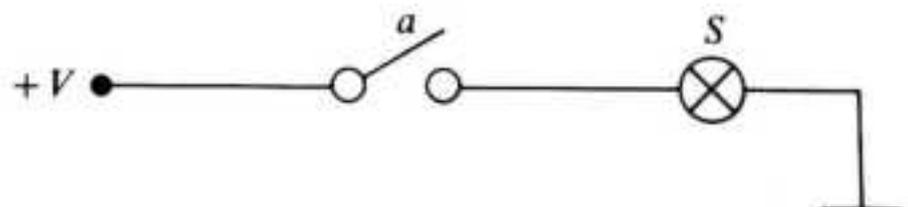


Figura 2.3. Tabla de la verdad y circuito equivalente de la función igualdad.

2.4.2. FUNCION UNION

La función unión es conocida también como función **reunión**, función **suma** o función **O** (**OR** en inglés). Su expresión matemática para dos variables será:

$$S = a + b$$

Su tabla de la verdad y el circuito construido mediante contactos se indica en la Figura 2.4.

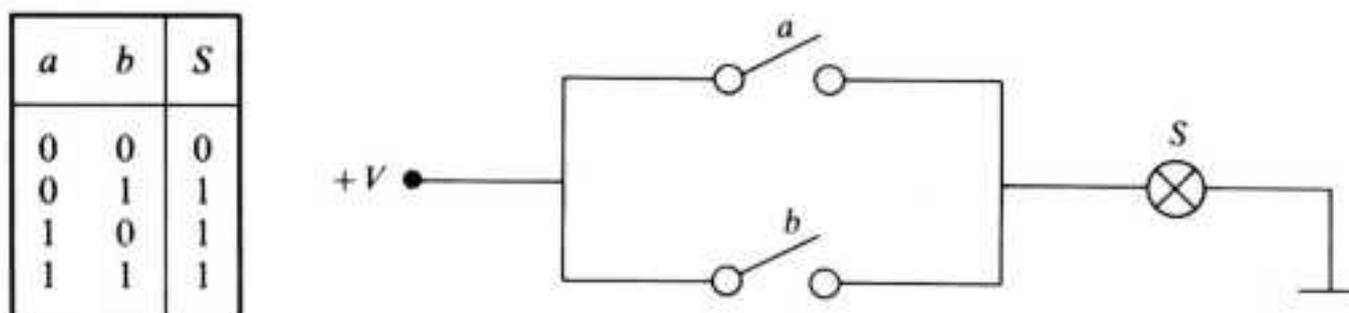


Figura 2.4. Tabla de la verdad y circuito equivalente de la función unión.

2.4.3. FUNCION INTERSECCION

La función intersección se conoce también como **producto** o función **Y** (**AND** en inglés). Su expresión para dos variables es la siguiente:

$$S = a \cdot b$$

La tabla de la verdad y el circuito realizado mediante contactos aparecen en la Figura 2.5.

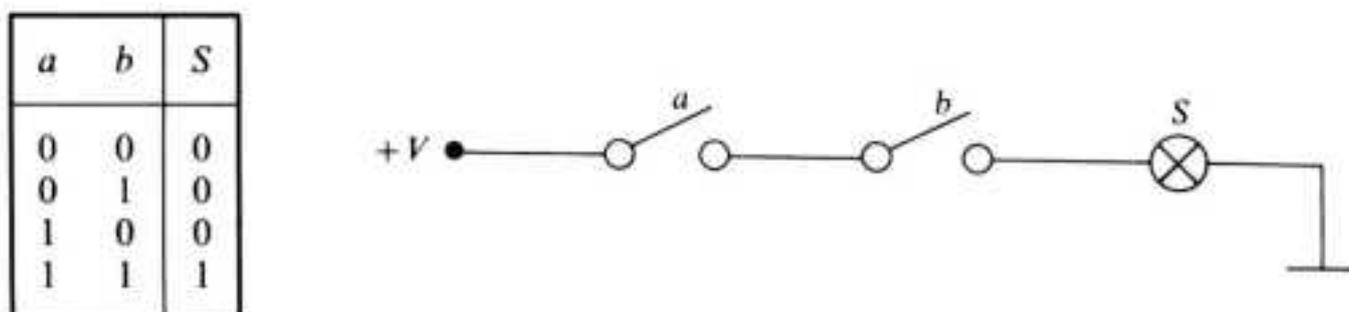


Figura 2.5. Tabla de la verdad y circuito equivalente de la función intersección.

2.4.4. FUNCION NEGACION

La función negación también es conocida como **complemento** o función **NO** (**NOT** en inglés). Su expresión es la siguiente:

$$S = \bar{a}$$

La tabla de la verdad se representa en la Figura 2.6.

a	S
0	1
1	0

Figura 2.6. Tabla de la verdad de la función negación o complemento.

2.5. OTRAS FUNCIONES BASICAS IMPORTANTES

Además de las funciones analizadas, existen otras que también ofrecen un gran interés porque, como veremos más adelante, resulta más fácil fabricar circuitos lógicos integrados con las funciones que describiremos a continuación que con las estudiadas anteriormente.

Las funciones a las que nos referimos son las siguientes:

- Función **NO Y** o **NAND**, que es la función **Y** negada.
- Función **NO O** o **NOR**, que es la función **O** negada.
- Función **O exclusiva** o **exclusive OR** en terminología inglesa.

En la Figura 2.7 aparece un cuadro resumen con las tablas de la verdad y la expresión matemática de cada una de las funciones referidas.

2.6. POSTULADOS, PROPIEDADES Y TEOREMAS MAS IMPORTANTES DEL ALGEBRA DE BOOLE

En este apartado describiremos los postulados, propiedades y teoremas más importantes del álgebra de Boole cuyo conocimiento resulta imprescindible, como veremos en el Capítulo 3, a la hora de simplificar expresiones lógicas por métodos algebraicos.

Denominación	Tabla	Función															
NAND (NO Y)	<table border="1"> <thead> <tr> <th><i>a</i></th> <th><i>b</i></th> <th><i>S</i></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	<i>a</i>	<i>b</i>	<i>S</i>	0	0	1	0	1	1	1	0	1	1	1	0	$S = \overline{a \cdot b}$
<i>a</i>	<i>b</i>	<i>S</i>															
0	0	1															
0	1	1															
1	0	1															
1	1	0															
NOR (NO O)	<table border="1"> <thead> <tr> <th><i>a</i></th> <th><i>b</i></th> <th><i>S</i></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	<i>a</i>	<i>b</i>	<i>S</i>	0	0	1	0	1	0	1	0	0	1	1	0	$S = \overline{a + b}$
<i>a</i>	<i>b</i>	<i>S</i>															
0	0	1															
0	1	0															
1	0	0															
1	1	0															
Exclusive OR (O exclusiva)	<table border="1"> <thead> <tr> <th><i>a</i></th> <th><i>b</i></th> <th><i>S</i></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	<i>a</i>	<i>b</i>	<i>S</i>	0	0	0	0	1	1	1	0	1	1	1	0	$S = 0$ si el n.º de unos es par $S = 1$ si el n.º de unos es impar $S = a \cdot \bar{b} + \bar{a} \cdot b$ $S = a \oplus b$
<i>a</i>	<i>b</i>	<i>S</i>															
0	0	0															
0	1	1															
1	0	1															
1	1	0															

Figura 2.7. Tabla de la verdad y expresión matemática de las funciones NAND, NOR y exclusive OR.

2.6.1. POSTULADOS

Vamos a exponer los postulados más significativos, y, para su mejor comprensión, acompañaremos cada uno de ellos con un circuito eléctrico realizado mediante contactos. En la Figura 2.8 aparece una relación de la mayor parte de los postulados que veremos a continuación:

- *Postulado 1*

La suma lógica de una variable más un 1 lógico equivale a un 1 lógico:

$$a + 1 = 1$$

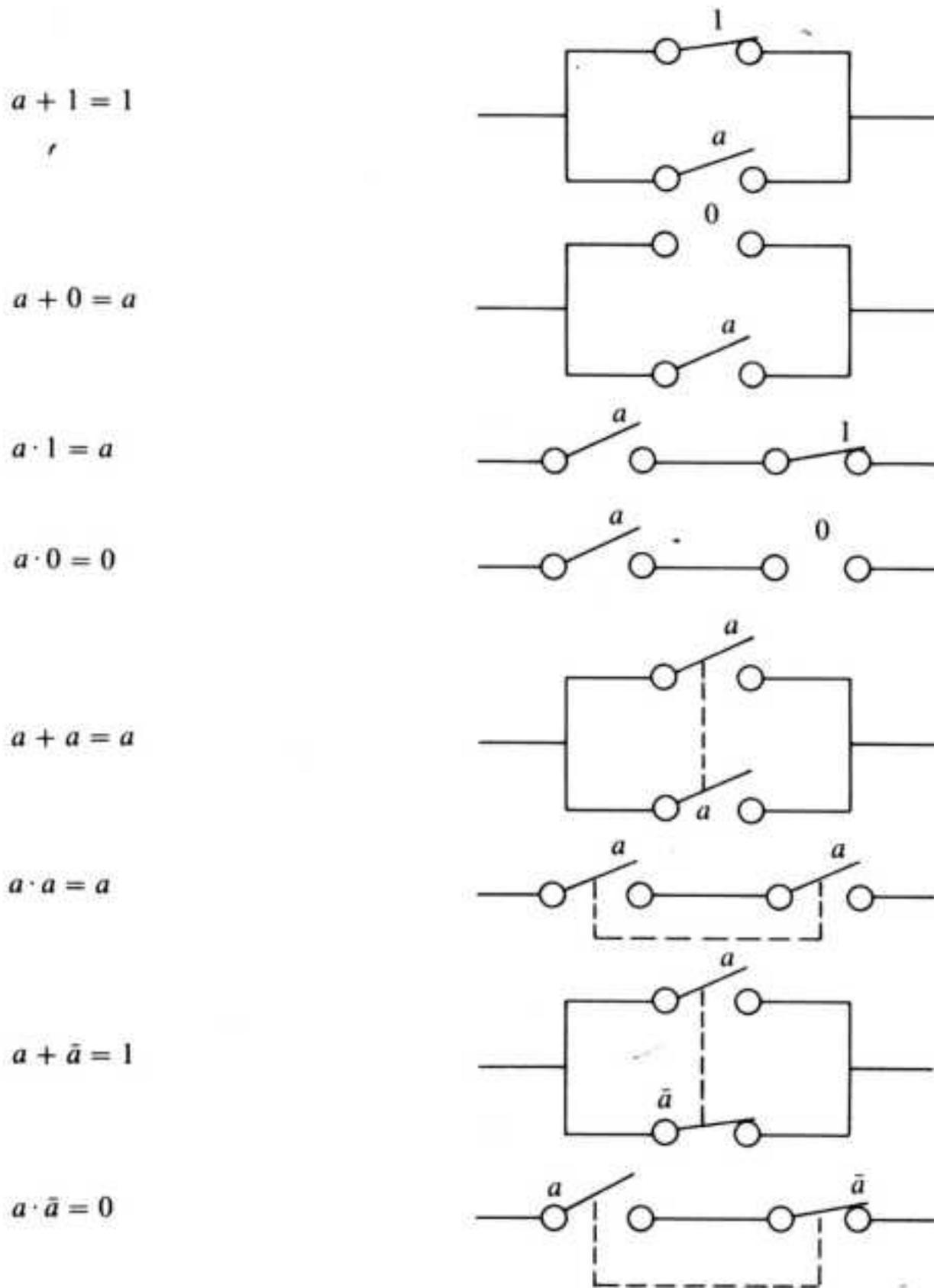


Figura 2.8. Circuito eléctrico equivalente de los postulados más significativos.

● *Postulado 2*

La suma lógica de una variable más un 0 lógico equivale al valor de la variable:

$$a + 0 = a$$

● *Postulado 3*

El producto lógico de una variable por un 1 lógico es igual al valor de la variable:

$$a \cdot 1 = a$$

● *Postulado 4*

El producto lógico de una variable por un 0 lógico es igual a 0:

$$a \cdot 0 = 0$$

● *Postulado 5*

La suma lógica de dos variables iguales equivale al valor de dicha variable:

$$a + a = a$$

● *Postulado 6*

El producto lógico de dos variables iguales equivale al valor de dicha variable:

$$a \cdot a = a$$

● *Postulado 7*

La suma lógica de una variable más la misma variable negada equivale a un 1 lógico:

$$a + \bar{a} = 1$$

● *Postulado 8*

El producto lógico de una variable por la misma variable negada equivale a un 0 lógico:

$$a \cdot \bar{a} = 0$$

● *Postulado 9*

Si una variable es negada dos veces, ésta no varía. Este postulado es válido para cualquier número par de inversiones:

$$\bar{\bar{a}} = a$$

● *Otros postulados*

Si se invierten los dos miembros de una igualdad, ésta no sufre ninguna variación:

$$S = a + b \quad ; \quad \bar{S} = \overline{a + b}$$

$$S = a \cdot b \quad ; \quad \bar{S} = \overline{a \cdot b}$$

2.6.2. PROPIEDADES

De la misma forma que en el sistema convencional, en el álgebra de Boole se cumplen las propiedades que describimos a continuación:

- *Propiedad conmutativa:*

$$a + b = b + a$$

$$a \cdot b = b \cdot a$$

- *Propiedad asociativa:*

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

$$a \cdot b \cdot c = a \cdot (b \cdot c)$$

- *Propiedad distributiva:*

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

$$a + b \cdot c = (a + b) \cdot (a + c)$$

2.6.3. TEOREMAS

Los teoremas que enunciamos seguidamente podrán ser demostrados apoyándonos en los postulados y propiedades descritos anteriormente.

- **Teorema 1. Ley de absorción.**

$$a) \quad a + a \cdot b = a$$

Demostración:

$$a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$$

$$b) \quad a \cdot (a + b) = a$$

Demostración:

$$a \cdot (a + b) = a \cdot a + a \cdot b = a + a \cdot b = a$$

- **Teorema 2.**

$$a) \quad a + \bar{a} \cdot b = a + b$$

Demostración:

$$a + \bar{a} \cdot b = (a + \bar{a}) \cdot (a + b) = 1 \cdot (a + b) = a + b$$

$$b) \quad b \cdot (a + \bar{b}) = a \cdot b$$

Demostración:

$$b \cdot (a + \bar{b}) = b \cdot a + b \cdot \bar{b} = b \cdot a + 0 = b \cdot a$$

● **Teorema 3. Leyes de De Morgan.**

Merecen especial mención las leyes de De Morgan por su gran utilidad en los procesos de simplificación. La comprobación de estos teoremas la podrá realizar el(la) lector(a) construyendo la tabla de la verdad:

$$a) \quad \overline{a + b} = \bar{a} \cdot \bar{b}$$

$$b) \quad \overline{a \cdot b} = \bar{a} + \bar{b}$$

2.7. FORMA CANONICA DE UNA FUNCION BOOLEANA

Se llama **forma canónica** de una función booleana a *todo producto de sumas o suma de productos en los cuales aparecen todas las variables en cada uno de los términos que constituyen la expresión, bien en forma directa, bien en forma complementada*. Son ejemplos de formas canónicas las siguientes funciones:

$$S_1 = a \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot \bar{b} \cdot \bar{c}$$

$$S_2 = (a + b + c) \cdot (a + \bar{b} + \bar{c}) \cdot (\bar{a} + b + \bar{c})$$

Las funciones del tipo de la S_1 reciben el nombre de **primera forma canónica** o **MINTERMS**, y las del tipo de la S_2 se denominan **segunda forma canónica** o **MAXTERMS**.

Toda función lógica puede ser expresada en forma canónica realizando las transformaciones oportunas.

2.8. FORMA DE OBTENER LA FUNCION LOGICA A PARTIR DE LA TABLA DE LA VERDAD

El proceso lógico de diseño comienza generalmente por representar la tabla de la verdad, asignando a la función los valores deseados para cada combinación binaria de las variables de entrada. A partir de la tabla de la verdad es posible, como veremos a continuación, obtener la función en forma canónica en cualquiera de sus dos modalidades.

- La primera forma canónica se obtiene sumando todos los productos lógicos que den a la función el valor 1.

- La segunda forma canónica se obtiene multiplicando todas las sumas lógicas que den a la función el valor 0.

Para comprender con mayor claridad estas últimas expresiones, obtendremos la primera y la segunda forma canónica de la tabla de la verdad contenida en la Figura 2.9.

a	b	c	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Figura 2.9. Tabla de la verdad de una función lógica o booleana.

Para representar la primera forma canónica de la función, debemos detenernos en aquellos puntos donde el valor de la función es 1. El número de términos de la expresión será igual al número de unos que figuran en la columna correspondiente a la función de salida; por tanto, la expresión en este caso será:

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

Como se puede comprobar, las variables aparecen en los distintos términos de forma negada cuando su valor es 0, ya que *para que un término sea un 1 lógico es necesario que cada uno de los factores que lo forman sea también 1*.

Para obtener la segunda forma canónica, debemos fijarnos en los ceros de la columna de la función. El número de términos será igual al número de ceros que aparezcan en dicha columna. En nuestro caso la expresión quedará de la siguiente forma:

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

En este caso las variables aparecen en forma negada cuando su valor en la tabla de la verdad es 1.

El valor de la primera y de la segunda forma canónica obtenido de una misma tabla de la verdad es el mismo. No es necesario representar ambas formas, sino que se obtendrá solamente aquella que tenga menos términos.

EJERCICIOS RESUELTOS

1. Construir mediante contactos el circuito correspondiente a las siguientes funciones:

a) $F_1 = a \cdot \bar{b} + \bar{a} \cdot b$

b) $F_2 = (a \cdot b \cdot \bar{c} + \bar{a} \cdot c) \cdot d$

Representar la tabla de la verdad de cada una de ellas.

Solución

a)

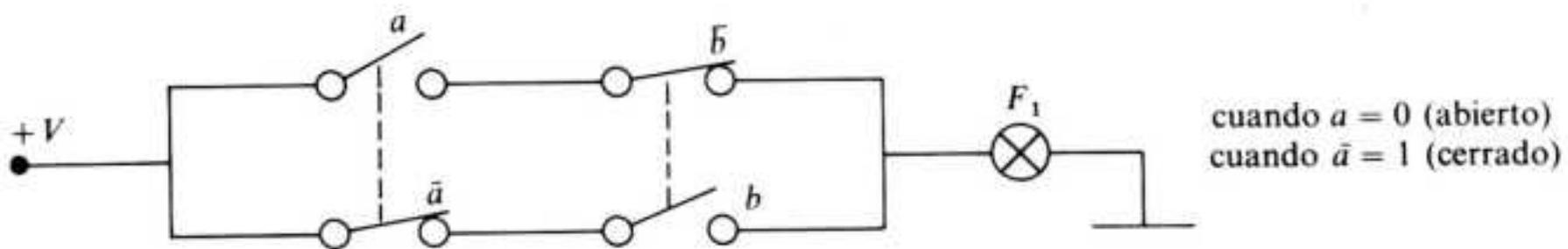


Figura 2.10. Circuito correspondiente a la función F_1 del ejercicio 1.

La tabla de la verdad será:

a	b	F_1
0	0	0
0	1	1
1	0	1
1	1	0

Como podemos comprobar corresponde a la función O exclusiva.

Figura 2.11. Tabla de la verdad de la función F_1 .

b)

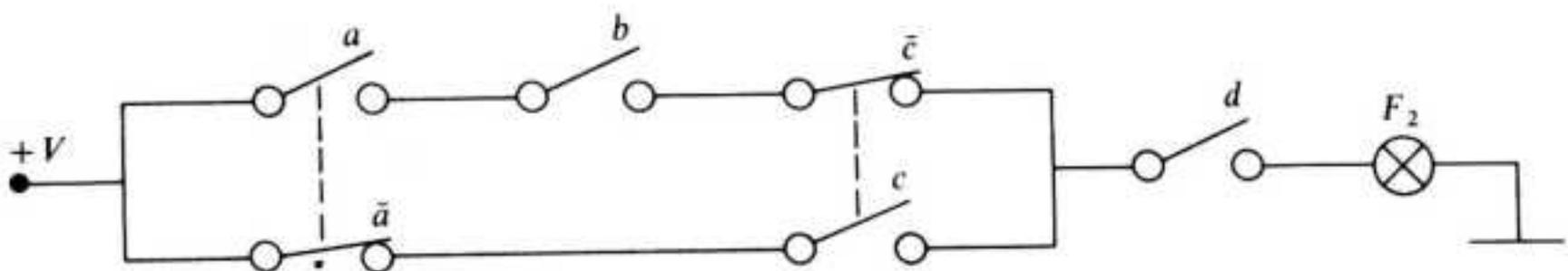


Figura 2.12. Circuito correspondiente a la función F_2 del ejercicio 1.

Su tabla de la verdad se representa en la Figura 2.13.

a	b	c	d	F_2
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Figura 2.13. Tabla de la verdad de la función F_2 .

2. Extraer la primera y la segunda forma canónica de las tablas de las Figuras 2.11 y 2.13.

Solución

Las funciones F_1 y F_2 expresadas en primera forma canónica o suma de productos serán:

$$F_1 = \bar{a} \cdot b + a \cdot \bar{b}$$

$$F_2 = \bar{a} \cdot \bar{b} \cdot c \cdot d + \bar{a} \cdot b \cdot c \cdot d + a \cdot b \cdot \bar{c} \cdot d$$

Las expresiones en forma de producto de sumas serán:

$$F_1 = (a + b) \cdot (\bar{a} + \bar{b})$$

$$F_2 = (a + b + c + d) \cdot (a + b + c + \bar{d}) \cdot (a + b + \bar{c} + d) \cdot (a + \bar{b} + c + d) \cdot (a + \bar{b} + c + \bar{d}) \cdot (a + \bar{b} + \bar{c} + d) \cdot (\bar{a} + b + c + d) \cdot (\bar{a} + b + c + \bar{d}) \cdot (\bar{a} + b + \bar{c} + d) \cdot (\bar{a} + b + \bar{c} + \bar{d}) \cdot (\bar{a} + \bar{b} + c + d) \cdot (\bar{a} + \bar{b} + \bar{c} + d) \cdot (\bar{a} + \bar{b} + \bar{c} + \bar{d})$$

3. Para realizar una primera selección de ingreso en una determinada empresa se establecen los siguientes requisitos:

1. Estar en posesión de título académico y dos años de experiencia en trabajo análogo al ofertado.
2. Acreditar cinco años de experiencia y vivir en la misma localidad.
3. Ser recomendados de la dirección.

En todos los casos será necesario tener coche propio.

Construir, mediante contactos, un circuito que efectúe automáticamente la selección.

Solución

Lás variables del problema son:

A = Título académico.

B = Dos años de experiencia.

C = Cinco años de experiencia.

D = Vivir en la misma localidad.

E = Recomendados.

F = Coche propio.

La expresión lógica será

$$S = (A \cdot B + C \cdot D + E) \cdot F$$

y el circuito equivalente:

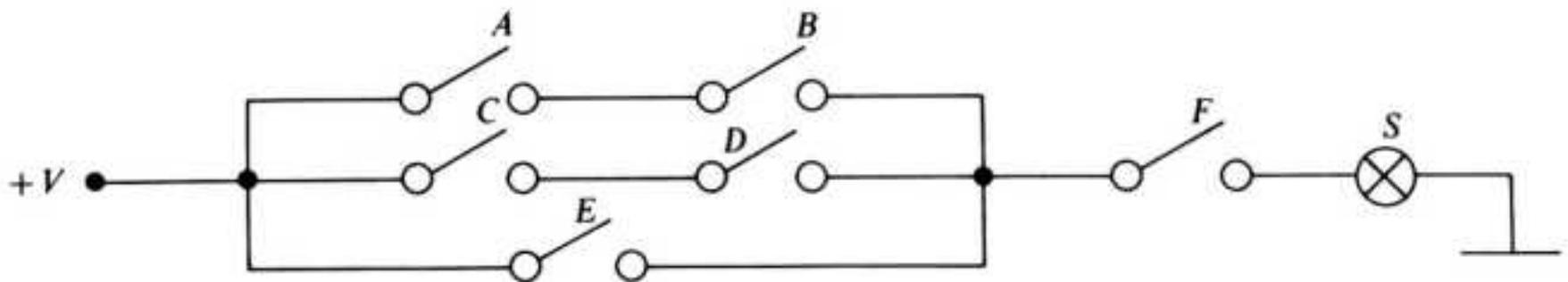


Figura 2.14. Circuito correspondiente a la función S del ejercicio 3.

4. Construir el circuito óptimo para la activación de una lámpara, empleando tres interruptores, de forma que la lámpara solamente se encienda cuando esté activado un solo interruptor o los tres simultáneamente.

Solución

En primer lugar es necesario definir la tabla de la verdad, en la cual cada variable representa a un interruptor:

a	b	c	L
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$L = 0$ (lámpara apagada)

$L = 1$ (lámpara encendida)

Figura 2.15. Tabla de la verdad del ejercicio 4.

La función en forma de suma de productos será

$$L = \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot c$$

Observando la tabla, la función también se puede escribir

$$L = a \oplus b \oplus c$$

El circuito que se obtiene al materializar la función se representa en la Figura 2.16.

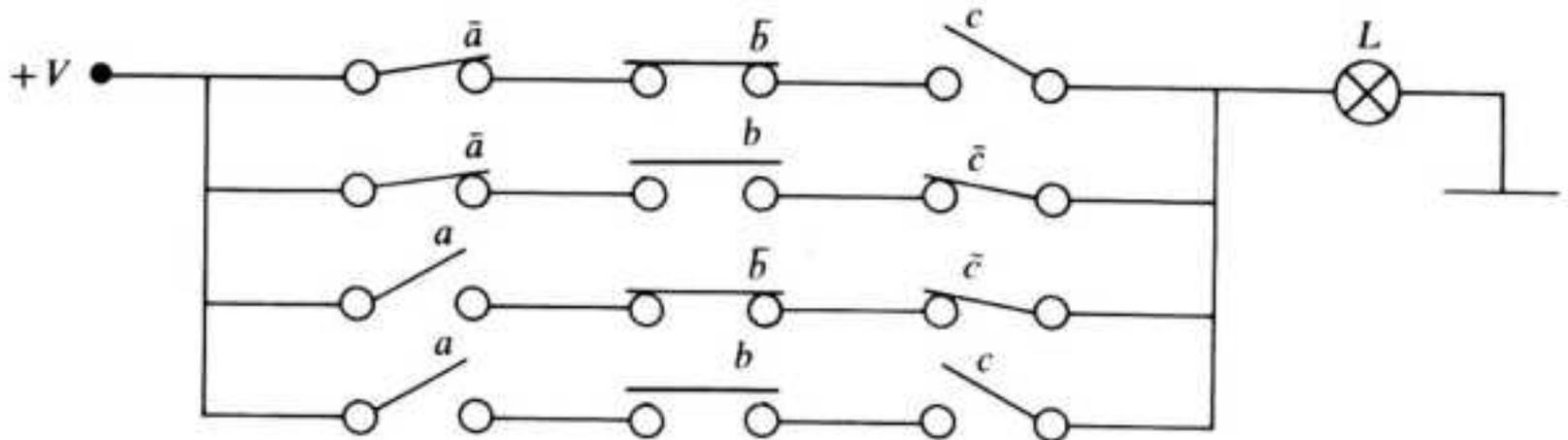


Figura 2.16. Circuito eléctrico correspondiente a la función L del ejercicio 4.

Utilizando conmutadores comerciales se obtiene el circuito representado en la Figura 2.17, que, si se examina detenidamente, es idéntico al anterior.

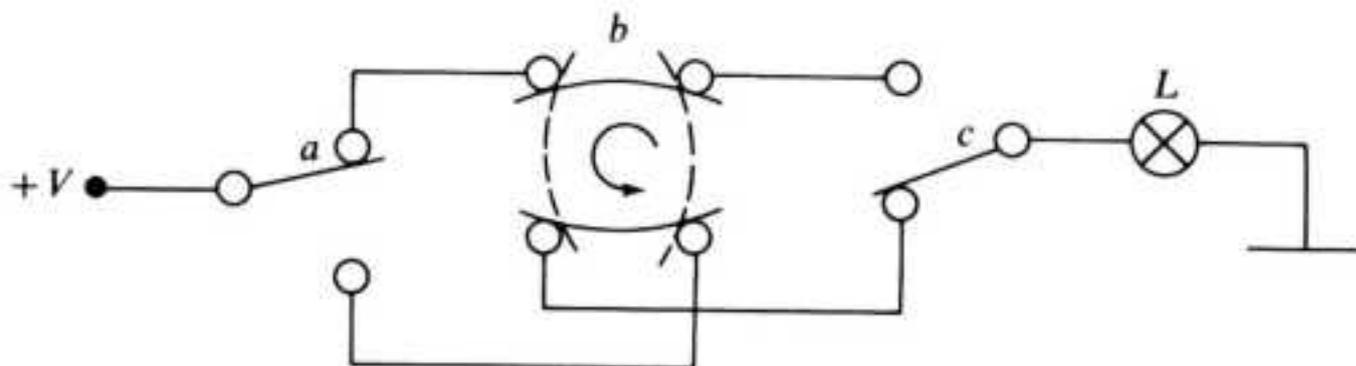


Figura 2.17. Circuito de conexión de tres conmutadores combinados para iluminar una lámpara.

EJERCICIOS PROPUESTOS

1. Para trasladarse desde un punto hasta otro de una gran ciudad existen varias combinaciones:

- Enlazar las líneas 1 y 2 del transporte subterráneo.
- Eligiendo la línea A de autobuses y a continuación de B.
- Coger primero la línea A de autobuses y posteriormente la línea 2 del transporte subterráneo.
- De una manera más cómoda en taxi.

En cualquier caso será necesario disponer del dinero suficiente para cubrir los gastos (supongamos que el gasto es igual en todos los casos).

Se pide un diseño del circuito adecuado que nos indique las combinaciones de los caminos elegidos cuando sean correctos.

2. Las actividades posibles a realizar un día del fin de semana son las siguientes:

- Salir al campo y comer un bocadillo.
- Visitar una ciudad próxima y comer en un restaurante.
- Pasear por el centro de nuestra ciudad y ver una película.

En los dos primeros casos es necesario disponer de coche y de un mapa de carreteras. En cualquier caso será imprescindible que no llueva o que no haga viento.

Escribir la ecuación lógica que recoja las condiciones señaladas.

3. Extraer la función, en sus dos formas canónicas, de la tabla de la verdad de la Figura 2.18:

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>F</i>
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Figura 2.18. Tabla de la verdad de la función *F*.

4. Representar el circuito correspondiente a las funciones:

$$F_1 = [a + (\bar{b} + c)] \cdot \bar{d}$$

$$F_2 = (b + \bar{c}) \cdot [\bar{a} + \bar{b} \cdot (c + \bar{d})]$$

5. Aplicar las propiedades y teoremas adecuados para simplificar las siguientes expresiones:

a) $a + b \cdot c \cdot d$

b) $a_1 \cdot a_2 \cdot (b_1 + b_2)$

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

d) $\overline{(a_1 + a_2) \cdot \bar{b} \cdot \bar{c}}$

Simplificación de funciones

3.1. INTRODUCCION

En el diseño de circuitos digitales resulta de enorme interés simplificar o minimizar las funciones obtenidas de la tabla de la verdad o directamente del enunciado de un problema. Cuanto más simplificadas resulten, menor será el número de componentes necesarios para su materialización en forma de circuito lógico.

La simplificación que se consiga como consecuencia de los métodos que describiremos en este capítulo está encaminada hacia la obtención de una función final con el menor número posible de términos y con el menor número de variables en cada uno de ellos.

Como tendremos ocasión de comprobar en el próximo capítulo, otra forma de simplificar, que se complementa con las que aquí expondremos, consiste en transformar la función, aunque para ello sea necesario aumentar la primitiva, convirtiéndola en otra que haga posible la construcción del circuito con total homogeneidad de componentes.

Una herramienta válida para reducir o simplificar funciones es todo el conjunto de relaciones y leyes relativas al álgebra de Boole expuestas en el capítulo anterior, pero este sistema no es fácil de utilizar, y para hacerlo se requiere una gran experiencia y, sobre todo, suerte.

Por esta razón se han desarrollado diversas técnicas que permiten una simplificación más sistemática y, como consecuencia, más cómoda y eficaz. De todos los métodos que existen, destacan dos: el primero es el denominado **gráfico** o de **KARNAUGH**, y el otro es el conocido como método **numérico** o de **QUINE-McCLUSKEY**. De ellos nos ocuparemos con detenimiento en los próximos apartados.

3.2. SIMPLIFICACION POR EL METODO ALGEBRAICO

No existe ninguna regla fija para efectuar la simplificación por este método. Consiste en reducir una función lógica, recurriendo, en la medida de lo posible, a los postulados, las propiedades y los teoremas del álgebra de Boole.

Sin ánimo de recoger todos los principios del álgebra lógica, ni siquiera los más representativos, analizaremos el proceso seguido para reducir la sencilla función del ejemplo siguiente:

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

Aplicando en primer lugar el teorema de De Morgan al término

$$(\bar{a} \cdot \bar{c})$$

resulta

$$f = a + b \cdot (a + c)$$

Según la propiedad distributiva

$$f = a + b \cdot a + b \cdot c$$

Por aplicación de la ley de absorción, que dice $a + b \cdot a = a$, se obtiene como resultado final

$$f = a + b \cdot c$$

3.3. PASO DE UNA FUNCION CUALQUIERA A FORMA CANONICA

Cualquier función lógica, sea cual sea la forma en que esté expresada inicialmente, puede ser transformada para su representación en forma canónica, bien en suma de productos o bien en producto de sumas.

Tampoco existen reglas fijas para realizar estas transformaciones, aunque, como en el caso anterior, se utilizan también los principios de álgebra lógica o binaria.

Para ilustrar este apartado desarrollaremos algunos ejemplos elegidos al azar e iremos indicando los pasos seguidos para llegar al resultado final en cada caso.

Sea la función:

$$f = a \cdot (b + c)$$

que deseamos transformar en producto de sumas o segunda forma canónica. Como esta función depende de tres variables, en la expresión final debe aparecer con todas ellas en cada término.

En su forma primitiva está constituida por dos términos: (a) y $(b + c)$. En el primero faltan las variables b y c y en el segundo la a .

En primer lugar, sumemos al término (a) los productos $(b \cdot \bar{b})$ y $(c \cdot \bar{c})$, que, como es sabido, valen 0 y, por tanto, no cambian el valor inicial de (a) . Al mismo

tiempo, sumaremos al segundo término el producto $(a \cdot \bar{a})$, ya que es esta variable la que le falta para completar el número total de ellas:

$$f = [a + (b \cdot \bar{b}) + (c \cdot \bar{c})] \cdot [b + c + (a \cdot \bar{a})]$$

Aplicando la propiedad distributiva en dos fases consecutivas obtendremos:

$$f = [(a + b) \cdot (a + \bar{b}) + (c \cdot \bar{c})] \cdot (a + b + c) \cdot (\bar{a} + b + c)$$

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

Como el término $(a + b + c)$ está repetido, se elimina una vez, siendo, en consecuencia, el resultado final:

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

Supongamos que deseamos ahora transformar la misma función a suma de productos o primera forma canónica. En primer lugar será necesario aplicar la propiedad distributiva:

$$f = a \cdot b + a \cdot c$$

En el primer término falta la variable c y en el segundo la b ; por tanto, aplicando criterios semejantes a los del caso anterior:

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

Ahora hemos multiplicado el término primero por $(c + \bar{c})$ y el segundo por $(b + \bar{b})$, que en ambos casos vale 1; por tanto, el valor inicial de cada uno de ellos no sufre variación. Operando se obtiene:

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

El resultado final será:

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

Para poder aplicar cualquiera de los dos métodos de reducción que abordaremos seguidamente, es imprescindible expresar la función en cualquiera de sus formas canónicas. Por este motivo ha sido necesario exponer el presente apartado.

Aunque cuando se obtiene la función directamente de la tabla de la verdad presenta la forma canónica, puede ocurrir que el origen de la expresión lógica sea otro, como, por ejemplo, el enunciado del problema directamente. En este caso, como en otros, la forma inicial no tiene por qué coincidir con la canónica.

3.4. METODO GRAFICO DE KARNAUGH

El método gráfico de Karnaugh es un sistema sencillo y corto para simplificar funciones de hasta cuatro variables. Aunque también es válido para cinco o seis, es prácticamente desechable en estos casos, porque, como veremos, las dificultades que presenta son mayores que las ventajas que ofrece. Por consiguiente, para funciones de más de cuatro variables es aconsejable acudir al método numérico, que analizaremos en el epígrafe siguiente.

Para su aplicación es necesario construir un cuadrilátero, que a su vez se divide en 2^n cuadrados elementales. El exponente n es el número de variables de la función.

En la Figura 3.1 se muestran los gráficos adecuados para reducir funciones de dos, tres y cuatro variables respectivamente. En estos casos es donde encuentra el método su mejor aplicación.

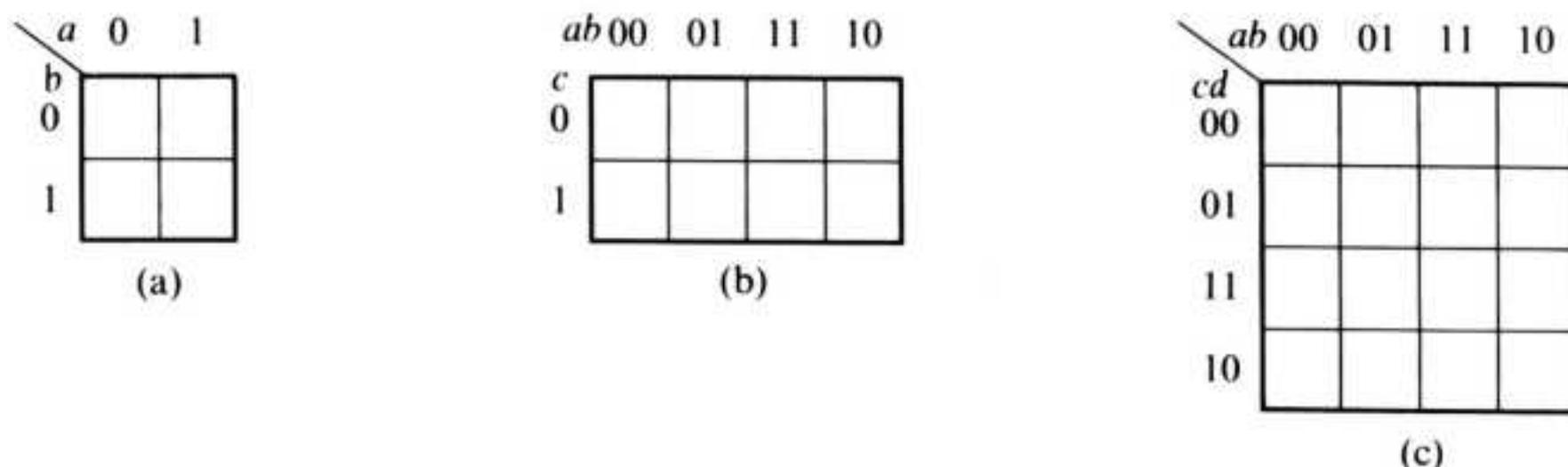


Figura 3.1. Gráficos o mapas de Karnaugh para reducción de funciones de dos (a), tres (b) y cuatro (c) variables.

En los laterales superior e izquierdo, como se representa en los gráficos de la Figura 3.1, deben aparecer todas las combinaciones que se puedan construir al asignar los dos posibles valores que pueden tomar las variables binarias. El orden de colocación de las combinaciones debe ser tal que de una a la siguiente solamente cambie el valor de una variable. Se dice entonces que entre dos casillas consecutivas existe **adyacencia algebraica**. Por consiguiente, en los casos de dos, tres y cuatro variables se produce una total coincidencia entre las adyacencias gráfica y algebraica. La primera casilla, tanto en sentido horizontal como vertical, es adyacente a la última.

Para explicar con mayor detalle los pasos necesarios para reducir una función, partiremos de un caso concreto. Supongamos que partiendo del planteamiento de un problema se llega a la tabla de la verdad de la Figura 3.2.

El primer paso será deducir la función, representándola en cualquiera de las dos posibles formas canónicas. Se optará por la primera o por la segunda, dependiendo del número de ceros o de unos que presente la columna correspondiente a los

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

Figura 3.2. Tabla de la verdad.

valores de función. En este caso elegiríamos la primera forma canónica o suma de productos por existir seis unos frente a diez ceros. Para que la exposición sea más completa, en nuestro ejemplo resolveremos el caso de las dos formas posibles.

La primera forma canónica será:

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

La segunda forma canónica será:

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

Las reglas que es necesario seguir para simplificar son las siguientes:

a) En primer lugar *se coloca un 1 en cada casilla* (tanto si se trata de la primera como de la segunda forma canónica) *donde exista la función*, tal y como se indica en la Figura 3.4. Para saber dónde existe la función, se ha representado en la Figura 3.3 la correspondencia entre términos y casillas.

b) A continuación se agrupan los unos en bloques de 2, 4, 8 ó 16 casillas. *Para formar los grupos es imprescindible que los unos se encuentren en casillas adyacentes.*

Es posible realizar agrupaciones distintas y todas estar bien hechas. El objetivo es construir el menor número de grupos posibles y recoger el mayor número de unos

	<i>ab</i>	00	01	11	10
<i>cd</i>	00	$\bar{a}\bar{b}\bar{c}\bar{d}$	$\bar{a}b\bar{c}\bar{d}$	$a\bar{b}\bar{c}\bar{d}$	$a\bar{b}c\bar{d}$
	01	$\bar{a}\bar{b}c\bar{d}$	$\bar{a}bc\bar{d}$	$a\bar{b}cd$	$a\bar{b}c\bar{d}$
	11	$\bar{a}b\bar{c}d$	$\bar{a}bcd$	$abcd$	$a\bar{b}c\bar{d}$
	10	$\bar{a}b\bar{c}\bar{d}$	$\bar{a}bc\bar{d}$	$a\bar{b}c\bar{d}$	$a\bar{b}cd$

Figura 3.3.

en cada uno de ellos, siempre que se cumpla la condición expuesta en este mismo punto. Un mismo 1 puede pertenecer a varios grupos.

En nuestro caso se ha optado por los agrupamientos que se muestran en los gráficos de la Figura 3.4.

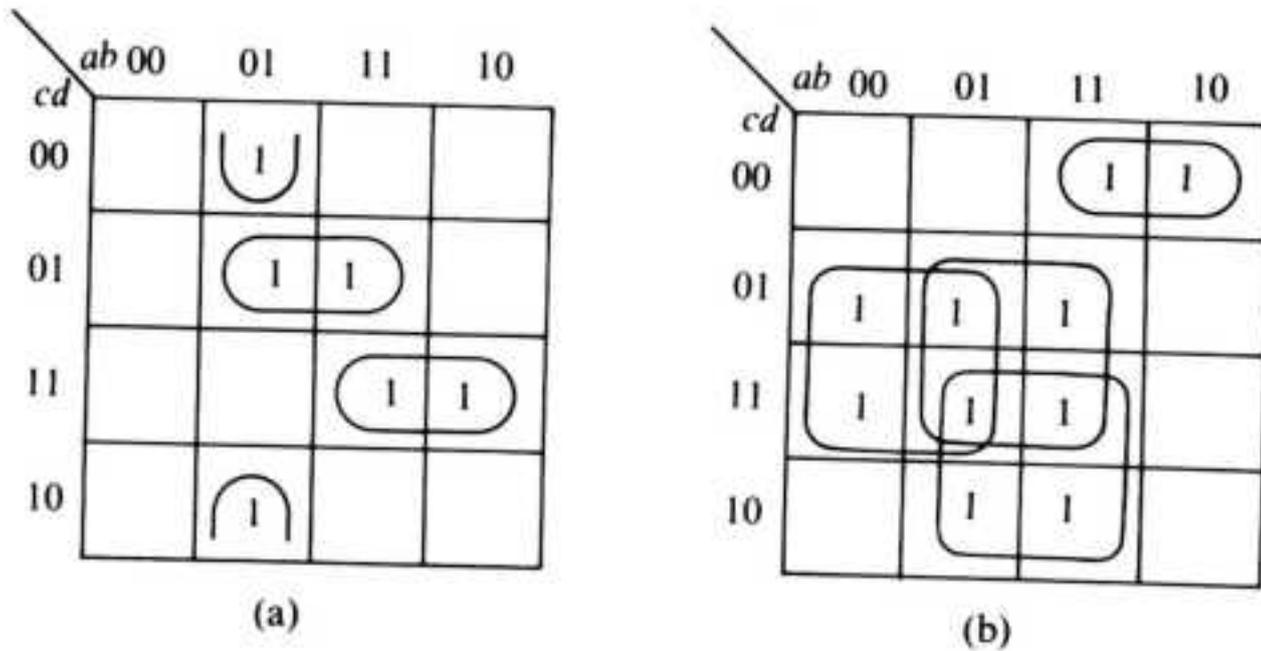


Figura 3.4. Agrupaciones de los términos de la primera forma canónica (a) y de la segunda (b).

c) A cada grupo de unos le corresponde un término. De cada grupo se eliminan las variables que intervienen con su *doble valor* (0 y 1). En el gráfico de la Figura 3.5 se eliminarán las variables *a* y *c*, ya que en algunas casillas del grupo valen 0 y en otras valen 1. Sin embargo, se observa que las variables *b* y *d* siempre valen lo mismo (uno en ambos casos).

d) Para obtener la función reducida, representaremos las variables en *forma negada*, cuando el valor que le corresponda sea un 0. Cuando el valor sea un 1, aparecerán de *forma directa*.

	<i>ab</i>	00	01	11	10
<i>cd</i>	00				
	01		1	1	
	11		1	1	
	10				

Figura 3.5. Agrupamiento de cuatro términos donde es posible eliminar las variables a y c .

Los resultados de nuestro ejemplo serán:

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

obtenida del primer gráfico de la Figura 3.4, y

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

obtenida del segundo gráfico.

Estas dos expresiones son equivalentes a las primitivas, deducidas de la tabla de la verdad.

3.4.1. METODO DE KARNAUGH PARA MAS DE CUATRO VARIABLES

Como hemos señalado anteriormente, el método de Karnaugh resulta muy engorroso para reducir funciones de más de cuatro variables.

En este caso la adyacencia algebraica no coincide con la adyacencia gráfica y esto dificulta enormemente el proceso de reducción. Para agrupar los unos ahora es necesario observar la simetría respecto de los ejes de la tabla. Los ejes vienen definidos por los cambios de valor de las variables.

No es nuestra intención resolver un ejercicio completo, pero si queremos hacer referencia a un caso de seis variables mediante el gráfico de la Figura 3.6, donde se indican los ejes de simetría, la ordenación de las combinaciones y algunas agrupaciones posibles de unos.

3.5. METODO NUMERICO DE QUINE-McCLUSKEY

Como fácilmente se puede comprobar en el apartado anterior, simplificar funciones lógicas con más de cuatro variables mediante el método de Karnaugh resulta

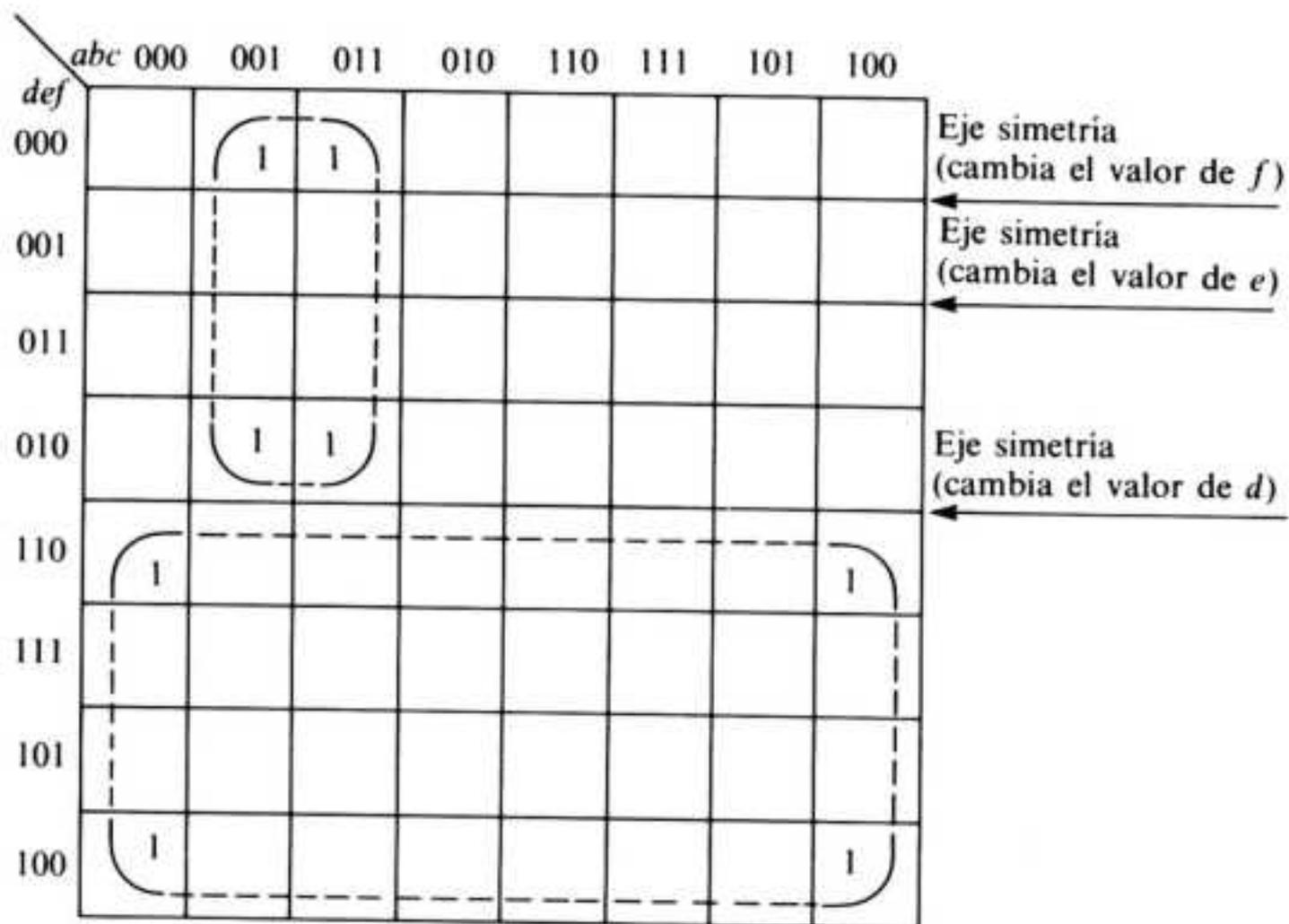


Figura 3.6. Gráfico de Karnaugh para seis variables.

molesto, pudiéndose cometer con relativa facilidad errores que desvirtúen el final del proceso de reducción o simplificación.

El método numérico de Quine-McCluskey es el adecuado para simplificar funciones de más de cuatro variables (también vale para resolver casos de funciones con un menor número de variables), aunque su aplicación implica un proceso más largo que en el caso anterior.

Debido a la sistematización del método, es posible utilizar un ordenador con el programa adecuado para ejecutar el proceso de simplificación. Esto supone, sin duda alguna, una enorme ventaja frente al de Karnaugh.

Para poder aplicar este método, de la misma manera que el analizado anteriormente, es necesario partir de la función expresada en forma canónica en cualquiera de sus dos modos posibles.

De la misma manera que en casos anteriores, para facilitar la explicación del proceso nos apoyaremos en un caso concreto que sea lo más representativo posible. Aunque hemos dicho que el método es el indicado para cinco o más variables, en este caso utilizaremos una función de cuatro para que el/la lector/a pueda contrastar el resultado con el obtenido aplicando el método de Karnaugh.

Sea la función:

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

a) En primer lugar es recomendable colocar en sentido vertical los términos de la función, tal como se indica en la Figura 3.7. En otra columna contigua se sustituirán las variables por unos o ceros, según que éstas aparezcan de forma directa o de forma negada, respectivamente.

$\bar{a}\bar{b}\bar{c}\bar{d}$	0000	Indice 0
$\bar{a}\bar{b}c\bar{d}$ $\bar{a}b\bar{c}\bar{d}$	0010 1000	Indice 1
$\bar{a}b\bar{c}d$ $a\bar{b}c\bar{d}$	0101 1010	Indice 2
$\bar{a}bcd$ $ab\bar{c}d$	0111 1101	Indice 3
$abcd$	1111	Indice 4

Figura 3.7. Agrupamiento de los términos de una función según el número de unos que contiene.

Los términos y combinaciones se organizarán en grupos de manera que en cada uno se encuentren todos aquellos que tengan el *mismo número de unos*.

El orden de colocación dentro de cada grupo debe ser de menor a mayor, siguiendo el valor numérico de las *combinaciones binarias* resultantes.

A cada grupo se le asignará un número de índice distinto que coincidirá con el número de unos de las combinaciones correspondientes.

b) Para facilitar la referencia, a cada término se le asignará el *número decimal equivalente* al valor numérico binario fruto de la combinación correspondiente*. Figura 3.8.

0000	0
0010 1000	2 8
0101 1010	5 10
0111 1101	7 13
1111	15

Figura 3.8. Correspondencia entre cada combinación binaria y su equivalente número decimal.

* Véase apartado 5.2.1 del Capítulo 5, donde se establece la relación entre el sistema de numeración decimal y el binario.

c) La reducción o eliminación de variables, de igual forma que en el método de Karnaugh, se basa en la siguiente propiedad:

$$a \cdot b + a \cdot \bar{b} = a$$

donde resulta fácil comprobar el resultado.

Esta ley, referida a este caso concreto, es generalizable y nos permite concluir enunciando que *una variable puede ser eliminada de dos términos de una función binaria, cuando en uno de ellos aparece en forma directa y en el otro en forma negada*. El resto de las variables de ambos términos deben ser las mismas y tener el mismo valor.

d) Seguidamente deben compararse todos los términos de cada uno de los grupos con todos los del grupo adyacente de índice superior, tal como se indica en la Figura 3.9. Si al comparar las combinaciones binarias correspondientes a dos términos la diferencia entre ellas es solamente de un dígito, éste se sustituirá por un guión y el resto quedará como estaba inicialmente. Puede ocurrir, como en el caso que nos ocupa, que, a veces, al comparar dos términos no se produzca reducción, diremos entonces que esos términos no son *reducibles*.

0000 y 0010	-----	00-0
0000 y 1000	-----	-000
0010 y 0101	-----	no es reducible
0010 y 1010	-----	-010
1000 y 0101	-----	no es reducible
1000 y 1010	-----	10-0
0101 y 0111	-----	01-1
0101 y 1101	-----	-101
1010 y 0111	-----	no es reducible
1010 y 1101	-----	no es reducible
0111 y 1111	-----	-111
1101 y 1111	-----	11-1

Figura 3.9. Comparación entre todos los términos de un grupo y todos los del grupo adyacente.

Los resultados de la comparación darán lugar a la *tabla de reducción de primer orden* de la Figura 3.10. Los números entre paréntesis indican las combinaciones o términos que originaron la reducción.

Indice 0	00-0 -000	(0, 2) (0, 8)
Indice 1	-010 10-0	(2, 10) (8, 10)
Indice 2	01-1 -101	(5, 7) (5, 13)
Indice 3	-111 11-1	(7, 15) (13, 15)

Figura 3.10. Reducción de primer orden.

e) A continuación, contemplando la tabla de la Figura 3.10, se procede, si es posible, a efectuar una *nueva reducción*, denominada en este caso *de segundo orden*. Los criterios son los mismos que los aplicados en el caso anterior. El resultado en el ejemplo propuesto es el contenido de las tablas de las Figuras 3.11 y 3.12.

Indice 0	-0-0 -0-0	(0, 2), (8, 10) (0, 8), (2, 10)
Indice 2	-1-1 -1-1	(5, 7), (13, 15) (5, 13), (7, 15)

Figura 3.11. Reducción de segundo orden.

Sobre las Figuras 3.11 y 3.12 es necesario hacer algunos comentarios. En primer lugar, como se puede observar, cuando aparecen dos términos reducidos que son iguales, se elimina uno de ellos. Por otra parte, puede ocurrir, como en este caso particular, que no aparezca ninguna reducción en un grupo intermedio (en nuestro caso en el índice 1).

El proceso de reducción continuaría, si fuese posible, obteniéndose una nueva tabla de *tercer orden*, *cuarto orden*, etc. En este caso concreto ya no es posible seguir reduciendo variables.

f) Para terminar se construye una tabla como la de la Figura 3.13, en la cual se colocan a la izquierda todas las variables de la función, en el mismo orden que en los términos de la expresión primitiva.

Indice 0	-0-0	(0, 2, 8, 10)
Indice 2	-1-1	(5, 7, 13, 15)

Figura 3.12. Reducción de segundo orden simplificada.

La fila de números situada a la derecha representa, en forma decimal, cada uno de los términos de la función. Son los números que aparecen entre paréntesis en todas las tablas anteriores.

Contemplando, en primer lugar, la Figura 3.12, comenzaremos a construir la tabla de la Figura 3.13, colocando las combinaciones reducidas debajo de las variables; a la vez iremos marcando con una x los números decimales de la tabla que coincidan con los que aparecen entre paréntesis, en línea con cada combinación. En este caso, con los términos reducidos de la Figura 3.12, se completa la serie numérica de la parte derecha de la Figura 3.13.

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	0	2	5	7	8	10	13	15
–	0	–	0	x	x			x	x		
–	1	–	1			x	x			x	x

Figura 3.13.

En aquellos casos donde no se complete la tabla será necesario retroceder a *niveles inferiores de reducción*. Si, aun así, faltase algún término por completar, sería necesario recurrir, en última instancia, a la tabla análoga a la representada en la Figura 3.8.

g) La función resultante la construiremos a partir de las combinaciones reducidas o sin reducir que aparecen en la columna situada en la parte izquierda de la Figura 3.13.

Es necesario elegir tantas combinaciones como sean precisas para recoger al menos una vez las x de cada uno de los términos indicados en forma decimal que aparecen en la zona derecha de la Figura 3.13. Evidentemente, siempre comenzaremos por las combinaciones más reducidas porque darán lugar a términos más simples.

Cada término estará formado por las variables que aparezcan sobre los números binarios que no han sido reducidos. Los unos se sustituirán por las variables en forma directa, y los ceros por las mismas en forma negada. En nuestro caso el resultado final será:

$$f = \bar{b} \cdot \bar{d} + b \cdot d$$

En el apartado siguiente analizaremos otros casos de distinto grado de complejidad con el fin de comprobar todas las aplicaciones del método y, mediante la resolución de nuevos ejercicios, completaremos la explicación que aquí hemos iniciado.

EJERCICIOS RESUELTOS

1. Simplificar por el método algebraico las siguientes expresiones:

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

d) $a \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c} \cdot d$

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

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

c) $[(a \cdot b) \cdot a] + (b \cdot \bar{a})$

Solución

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

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

c) $[(a \cdot b) \cdot a] + (b \cdot \bar{a}) = a \cdot b + \bar{a} \cdot b = b(a + \bar{a}) = b$

d) $a \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c} \cdot d = a \cdot c(b + \bar{b}) + a \cdot b \cdot \bar{c} \cdot d = a \cdot c + a \cdot b \cdot \bar{c} \cdot d =$
 $= a \cdot (c + b \cdot \bar{c} \cdot d) = a[(c + b) \cdot (c + \bar{c}) \cdot (c + d)] = a \cdot (c + b \cdot d) = a \cdot c + a \cdot b \cdot d$

e) $a \cdot b \cdot c \cdot d + a \cdot b + a \cdot b \cdot d(\bar{e} + \bar{f}) = a \cdot b[c \cdot d + 1 + d(\bar{e} + \bar{f})] = a \cdot b$

2. Transformar en primera forma canónica la expresión

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

Solución

$$\begin{aligned} F &= (a + b) \cdot (b + \bar{c}) \cdot (\bar{a} + \bar{c}) = (a \cdot b + a \cdot \bar{c} + b \cdot b + b \cdot \bar{c}) \cdot (\bar{a} + \bar{c}) = \\ &= \cancel{a \cdot \bar{a} \cdot b} + a \cdot b \cdot \bar{c} + \cancel{a \cdot \bar{a} \cdot \bar{c}} + a \cdot \bar{c} \cdot \bar{c} + \bar{a} \cdot b + b \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} + b \cdot \bar{c} \cdot \bar{c} = \\ &= a \cdot b \cdot \bar{c} + a \cdot \bar{c} + \bar{a} \cdot b + b \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} + \cancel{b \cdot \bar{c}} = \\ &= a \cdot b \cdot \bar{c} + \cancel{a \cdot \bar{b} \cdot \bar{c}} + a \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot c + \cancel{a \cdot \bar{b} \cdot \bar{c}} + \cancel{a \cdot b \cdot \bar{c}} + \bar{a} \cdot b \cdot \bar{c} + \cancel{\bar{a} \cdot \bar{b} \cdot \bar{c}} = \\ &= a \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot c + \bar{a} \cdot b \cdot \bar{c} \end{aligned}$$

3. Reducir, mediante el método gráfico de Karnaugh, las funciones obtenidas de las tablas de la verdad de las Figuras 3.14 y 3.15.

a	b	c	F_1
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Figura 3.14. Tabla de la verdad de la función F_1 .

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	F_2
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Figura 3.15. Tabla de la verdad de la función F_2 .

Solución

a) Representamos la función F_1 en forma de suma de productos:

$$F_1 = \bar{a} \cdot \bar{b} \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot \bar{b} \cdot c$$

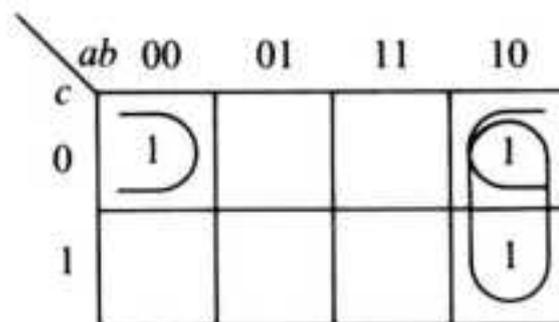


Figura 3.16. Método de Karnaugh para F_1 .

La función reducida será:

$$F_1 = a \cdot \bar{b} + \bar{b} \cdot \bar{c} = \bar{b} \cdot (a + \bar{c})$$

b) Fijándonos en las combinaciones de la Figura 3.15, para las cuales la función vale 1, podemos representar estos valores directamente en el mapa de Karnaugh sin necesidad de escribir la función.

	ab	00	01	11	10
cd	00	1	1		1
	01		1		
	11		1		
	10	1	1		1

Figura 3.17. Método de Karnaugh para F_2 .

Como resultado obtendremos

$$F_2 = \bar{b} \cdot \bar{d} + \bar{a} \cdot b$$

4. Simplificar, utilizando el método de Quine-McCluskey, la siguiente función:

$$F = a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d + \bar{a} \cdot b \cdot c \cdot \bar{d} + a \cdot b \cdot c \cdot d$$

Solución

$\bar{a}\bar{b}\bar{c}\bar{d}$	0000	Indice 0
$\bar{a}\bar{b}\bar{c}d$ $\bar{a}\bar{b}c\bar{d}$	0001 1000	Indice 1
$\bar{a}bc\bar{d}$	0110	Indice 2
$abcd$	1111	Indice 4

Figura 3.18.

0000	0
0001 1000	1 8
0110	6
1111	15

Figura 3.19.

Comparando las combinaciones de cada grupo con las del adyacente, obtenemos las reducciones de la Figura 3.20.

Indice 0	000- -000	(0, 1) (0, 8)
----------	--------------	------------------

Figura 3.20. Reducción de primer orden.

No es posible efectuar más reducciones de orden superior.

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	0	1	6	8	15
0	0	0	-	x	x			
-	0	0	0	x			x	
0	1	1	0			x		
1	1	1	1					x

Figura 3.21.

Como se puede observar, para completar los términos de la función primitiva ha sido necesario añadir las combinaciones 0110 para el término 6 y 1111 para el término 15.

La función resultante será:

$$F = a \cdot b \cdot c \cdot d + \bar{a} \cdot b \cdot c \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{b} \cdot \bar{c} \cdot \bar{d}$$

5. Reducir por el método Quine-McCluskey la función obtenida de la tabla de la verdad de la Figura 3.22.

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>F</i>
0	0	0	0	0	1
0	0	0	0	1	1
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	1	1
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	1
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	1
0	1	1	1	1	1

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

Figura 3.22. Tabla de la verdad de la función *F*.

La función en forma de producto de sumas será:

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

Los términos agrupados y ordenados se muestran en la Figura 3.23. También se indica el número decimal equivalente a cada combinación.

En la Figura 3.24 se muestran las reducciones de primer orden.

Indice 0	$\bar{a}\bar{b}\bar{c}\bar{d}\bar{e}$	00000	0
Indice 1	$\bar{a}\bar{b}\bar{c}\bar{d}e$	00001	1
	$\bar{a}\bar{b}\bar{c}d\bar{e}$	00010	2
Indice 2	$\bar{a}\bar{b}cde$	00011	3
	$\bar{a}\bar{b}c\bar{d}\bar{e}$	00110	6
	$\bar{a}b\bar{c}d\bar{e}$	01010	10
	$\bar{a}bc\bar{d}\bar{e}$	01100	12
	$\bar{a}bcde$	10010	18
Indice 3	$\bar{a}bcde$	00111	7
	$\bar{a}b\bar{c}de$	01011	11
	$\bar{a}bc\bar{d}e$	01101	13
	$\bar{a}b\bar{c}d\bar{e}$	10011	19
	$\bar{a}bc\bar{d}\bar{e}$	10101	21
	$\bar{a}bcde$	11010	26
	$\bar{a}bc\bar{d}\bar{e}$	11100	28
Indice 4	$ab\bar{c}de$	11011	27
	$abc\bar{d}\bar{e}$	11101	29

Figura 3.23.

Indice 0	0000-	(0, 1)
	000-0	(0, 2)
Indice 1	000-1	(1, 3)
	0001-	(2, 3)
	00-10	(2, 6)
	0-010	(2, 10)
Indice 2	-0010	(2, 18)
	00-11	(3, 7)
	0-011	(3, 11)
	-0011	(3, 19)
	0011-	(6, 7)
	0101-	(10, 18)
	-1010	(10, 26)
	0110-	(12, 13)
	-1100	(12, 28)
	1001-	(18, 19)
1-010	(18, 26)	
Indice 3	-1011	(11, 27)
	1-011	(19, 27)
	1101-	(26, 27)
	-1101	(13, 29)
	1-101	(21, 29)
	1110-	(28, 29)

Figura 3.24. Reducción de primer orden.

En la Figura 3.25 aparece la tabla con las reducciones de segundo orden. En ella se han eliminado las combinaciones repetidas.

Indice 0	000—	(0, 1, 2, 3)
Indice 1	00-1-	(2, 3, 6, 7)
	0-01-	(2, 3, 10, 11)
	-001-	(2, 3, 18, 19)
	--010	(2, 10, 18, 26)
Indice 2	--011	(3, 11, 19, 27)
	-101-	(10, 11, 26, 27)
	-110-	(12, 13, 28, 29)
	1-01-	(18, 19, 26, 27)

Figura 3.25. Reducción de segundo orden.

Si observamos atentamente la Figura 3.25, comprobaremos que no es posible seguir reduciendo.

En la tabla de la Figura 3.26 colocaremos, en primer lugar, todas las combinaciones o bloques reducidos de la Figura 3.25. Como el número 21 no queda cubierto, es necesario añadir una combinación de la tabla reducida de primer orden de la Figura 3.24 que se haya obtenido a partir de este término. Este bloque es el 1-101, que cubre además el término 29. Para construir la función simplificada es obligatorio elegir el bloque 1-101.

Hemos seleccionado para cubrir el resto de los términos las siguientes combinaciones:

- 000-- que cubre los términos 0, 1, 2 y 3
- 00-1- que cubre los términos 2, 3, 6 y 7
- 001- que cubre los términos 2, 3, 18 y 19
- 101- que cubre los términos 10, 11, 26 y 27
- 110- que cubre los términos 12, 13, 28 y 29

La función se obtendrá a partir de todos estos bloques más el 1-101.

a	b	c	d	e	0	1	2	3	6	7	10	11	12	13	18	19	21	26	27	28	29	
→	0	0	0	-	X	X	X	X														
→	0	0	-	1			X	X	X	X												
	0	-	0	1			X	X			X	X										
→	-	0	0	1			X	X							X	X						
	-	-	0	1	0		X				X				X			X				
	-	-	0	1	1			X				X				X				X		
→	-	1	0	1							X	X						X	X			
→	-	1	1	0								X	X							X	X	
	1	-	0	1									X	X				X	X			
→	1	-	1	0	1													X			X	

Figura 3.26.

Como se puede comprobar en la Figura 3.26, podríamos haber elegido otras combinaciones distintas para cubrir todos los términos. La función resultante, aun siendo diferente a la que aquí indicamos, también sería válida.

La función simplificada será:

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

EJERCICIOS PROPUESTOS

1. Simplificar las siguientes expresiones:

a) $[(a + b) + c] + (\bar{a} + b)$

b) $\overline{(\bar{a} + b)} \cdot \overline{[c \cdot (b + \bar{c})]}$

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

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

2. Calcular el complemento de F si

$$F = a + b + c + d$$

donde $a = x \cdot y \cdot z$, $b = \bar{x} \cdot y \cdot z$, $c = x \cdot \bar{y} \cdot z$ y $d = x \cdot y \cdot \bar{z}$.

3. Transformar en primera y segunda forma canónica:

$$F_1 = a \cdot \bar{b} + a \cdot c + b \cdot \bar{c}$$

$$F_2 = \bar{a} + \overline{(b \cdot \bar{c})}$$

4. Extraer las funciones lógicas de las tablas de las Figuras 3.14 y 3.15 en forma de producto de sumas y reducir mediante el método gráfico de Karnaugh.

5. Simplificar por el método de Quine-McCluskey la función

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

6. Extraer la función de la tabla de la Figura 3.22 en forma de suma de productos y simplificar utilizando el método de Quine-McCluskey.

Puertas lógicas

4.1. INTRODUCCION

Después de haber visto los capítulos anteriores nos encontramos en condiciones de exponer cuáles son los pasos necesarios para diseñar un sencillo circuito digital. En general, son éstos:

- a) Traducir el problema en una tabla de la verdad.
- b) Extraer de la tabla la función en forma canónica.
- c) Simplificar por el método más oportuno.
- d) Construir el circuito con los dispositivos más adecuados.

La ejecución de un circuito lógico se puede realizar, como hemos hecho en capítulos anteriores, mediante contactos, interruptores, etc. También se pueden utilizar elementos electrónicos básicos tales como diodos, transistores, etc.

Sin embargo, lo más práctico y fiable es la materialización utilizando **puertas lógicas integradas** construidas en tecnología SSI. Cada una de estas puertas realiza una función lógica básica de las estudiadas en el Capítulo 2.

Existe un amplio número de puertas distintas, clasificadas por **familias**, que, como veremos más adelante, presentan características distintas, dependiendo de la tecnología con que estén construidas.

4.2. SIMBOLOGIA LOGICA

En los casos anteriores, cuando representábamos un circuito, asociábamos las variables del problema con contactos o interruptores. A partir de ahora las variables serán las entradas de los bloques lógicos integrados que iremos abordando progresivamente. Por tanto, las variables de entrada serán señales eléctricas digitales, de la misma manera que lo serán las funciones o salidas de los circuitos.

En la Figura 4.1 se muestra la representación simbólica más generalizada de cada una de las puertas integradas que efectúan funciones lógicas básicas.

Estos son los elementos más simples de los que se parte en Electrónica Digital para la construcción de otros bloques de mayor complejidad.

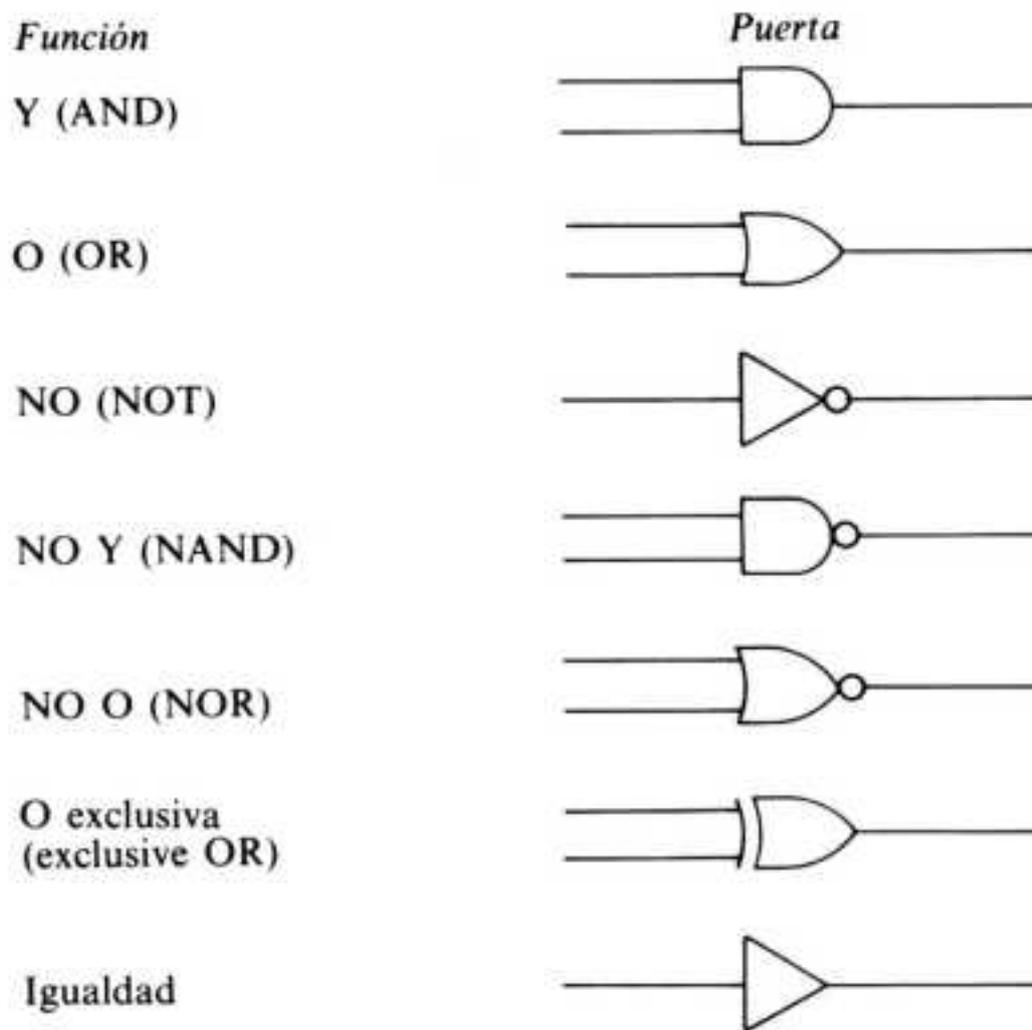


Figura 4.1. Representación simbólica de las funciones lógicas básicas.

4.3. CONSTITUCION DE LOS CIRCUITOS INTEGRADOS QUE CONTIENEN PUERTAS LOGICAS

Cada bloque o circuito integrado está formado, generalmente, por varias puertas de un mismo tipo.

El número de puertas por circuito depende del número de entradas que tenga cada una de ellas, estableciéndose una relación inversa entre entradas y puertas.

En la Figura 4.2 se muestra un circuito muy normal formado por cuatro puertas de dos entradas cada una.

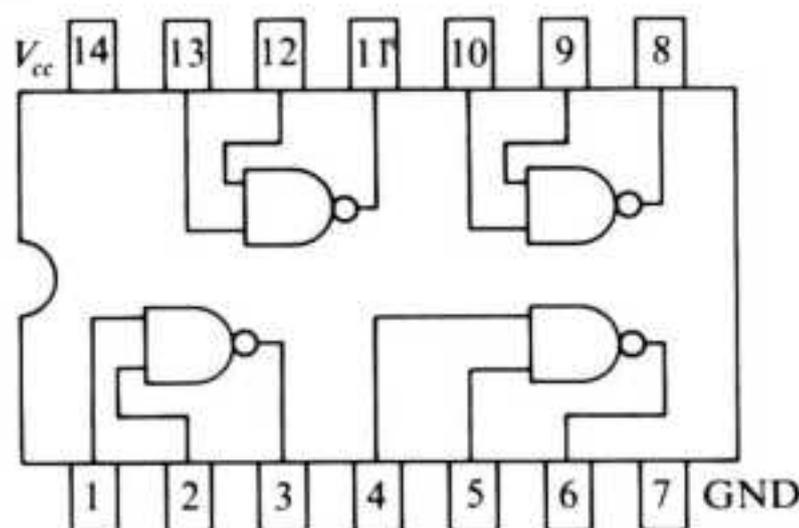


Figura 4.2. Diagrama de conexión de un circuito integrado con sistema de encapsulado DIL (Dual In Line).

El número de terminales (conocidos también como **pins**) de cada circuito es de 14 para el caso de las puertas.

El encapsulado más normal es el denominado **DIL** (Dual In Line). Los dos terminales de alimentación (7 y 14) son comunes para todas las puertas de un mismo circuito.

Cada una de las puertas o funciones está constituida por elementos electrónicos básicos (*resistencias, diodos y transistores* fundamentalmente). El número total de componentes depende del tipo de función que realice y también de la tecnología empleada. En la Figura 4.3 aparece la configuración interna de una puerta lógica.

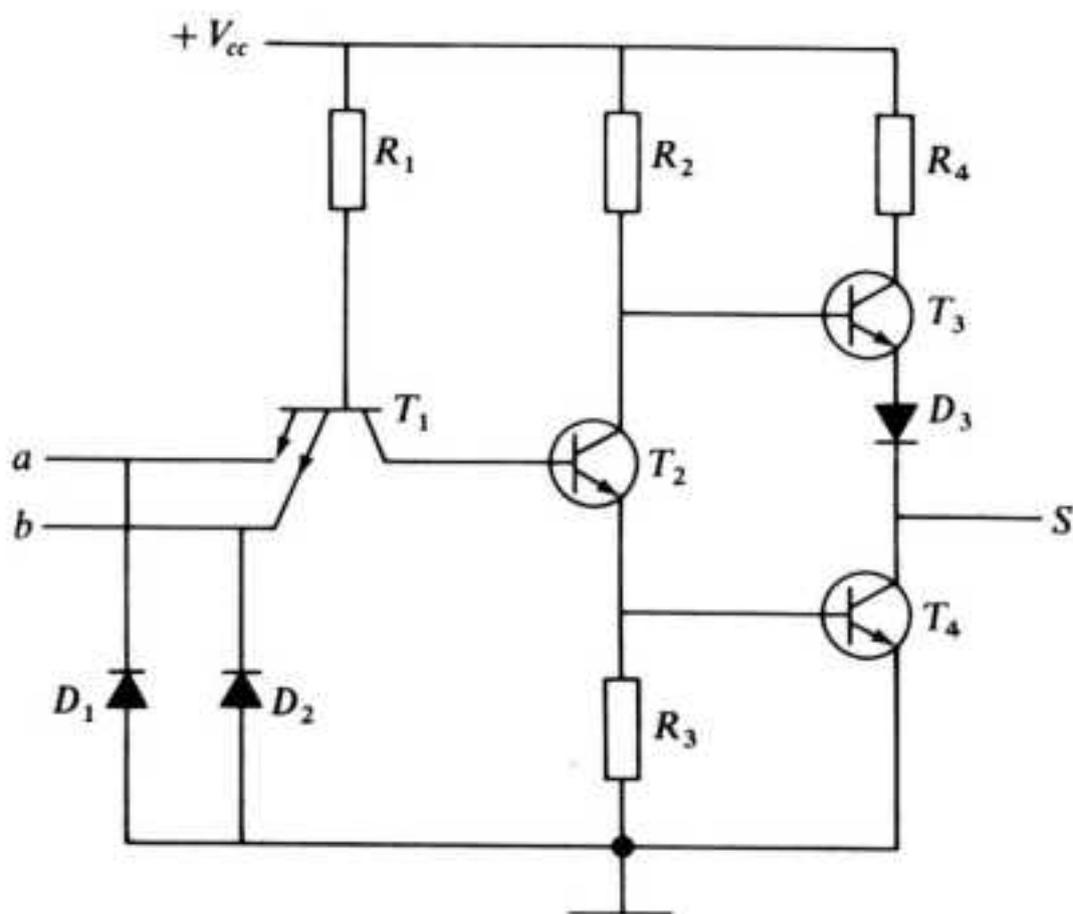


Figura 4.3. Configuración interna de una puerta lógica.

En general, son más sencillas de construir las funciones NAND y NOR. Por esta razón estas puertas, fundamentalmente la NAND, están consideradas como funciones universales. En consecuencia, es conveniente, con el fin de emplear el menor número de bloques y además los más baratos, transformar la función, después de reducida, para que puedan utilizarse puertas de un solo tipo.

4.4. LAS FUNCIONES NAND Y NOR COMO FUNCIONES UNIVERSALES

Todas las funciones se pueden construir utilizando exclusivamente puertas NAND o puertas NOR.

En la Figura 4.4 se muestran las funciones básicas Y, O y NO materializadas mediante estos dos tipos de puertas.

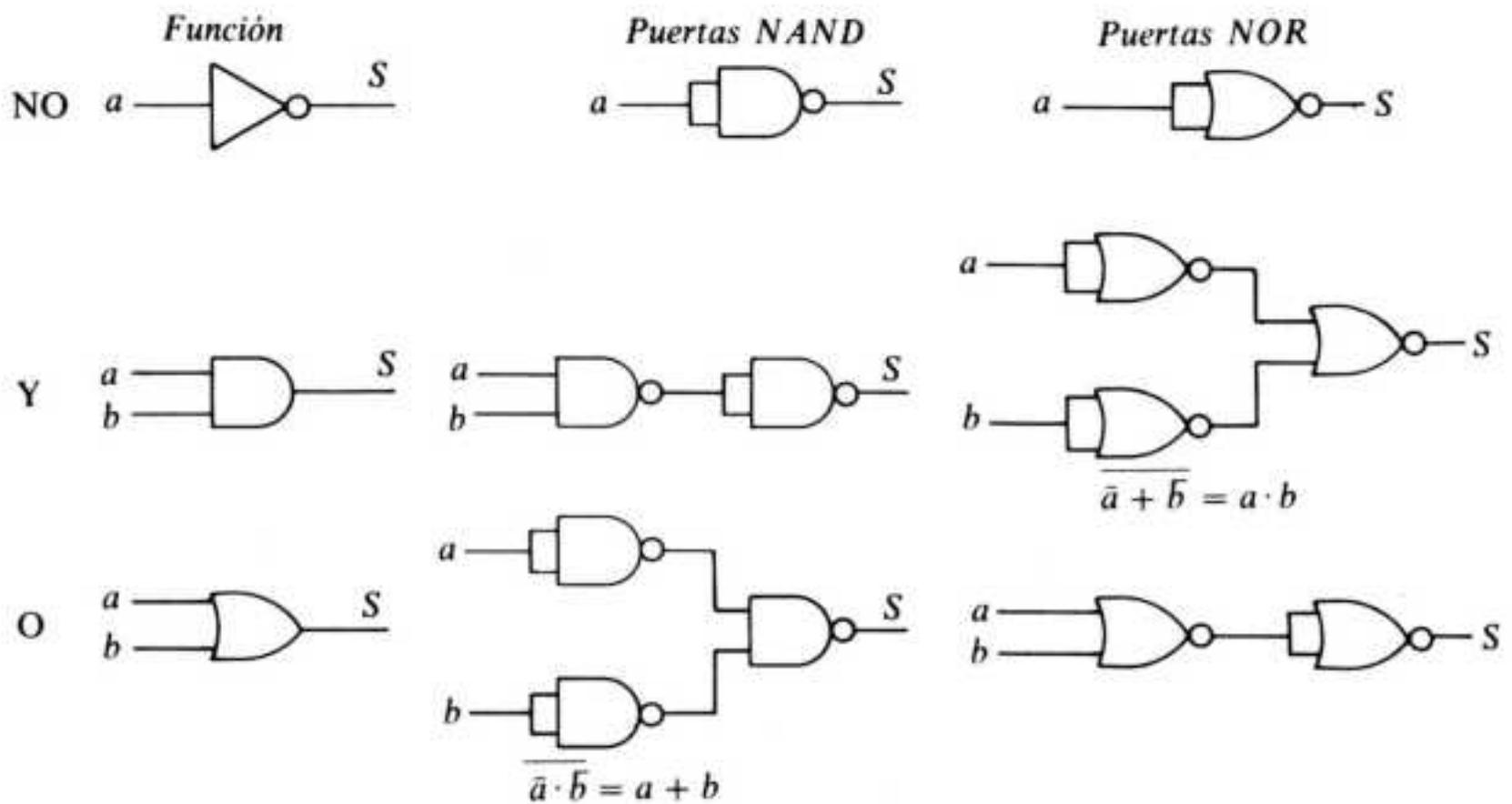


Figura 4.4. Representación de las funciones básicas NO, Y y O con puertas NAND y NOR.

4.4.1. IMPLEMENTACION DE FUNCIONES MEDIANTE PUERTAS NAND

El proceso que se debe seguir para transformar cualquier tipo de función en una expresión algebraica tal que se pueda implementar con puertas NAND solamente es el siguiente:

- En primer lugar debe aplicarse a la expresión en su conjunto una *doble inversión*.
- Si la función es un *producto*, las dos negaciones deben dejarse tal cual. Si es una *suma*, se elimina una de ellas mediante la aplicación del *teorema de De Morgan*.
- Se continúa invirtiendo doblemente los términos o partes de la función hasta que todas las *sumas* y *productos* se conviertan en productos negados.

Ejemplos:

$$b \cdot c + d + \bar{e} = \overline{\overline{b \cdot c + d + \bar{e}}} \quad \overline{\overline{b \cdot c} \cdot \overline{\overline{d + \bar{e}}}}$$

$$1. S_1 = b \cdot c + d + \bar{e}; S_1 = \overline{\overline{b \cdot c + d + \bar{e}}} = \overline{\overline{b \cdot c} \cdot \overline{\overline{d + \bar{e}}}}$$

$$2. S_2 = (a + b) \cdot (c + \bar{a}); S_2 = \overline{\overline{(a + b) \cdot (c + \bar{a})}} = \overline{\overline{\overline{a + b}} \cdot \overline{\overline{c + \bar{a}}}} = \overline{\overline{\overline{a + b}} \cdot \overline{\overline{c + \bar{a}}}}$$

En la Figura 4.5 se representan los esquemas lógicos de las funciones después de ser transformadas.

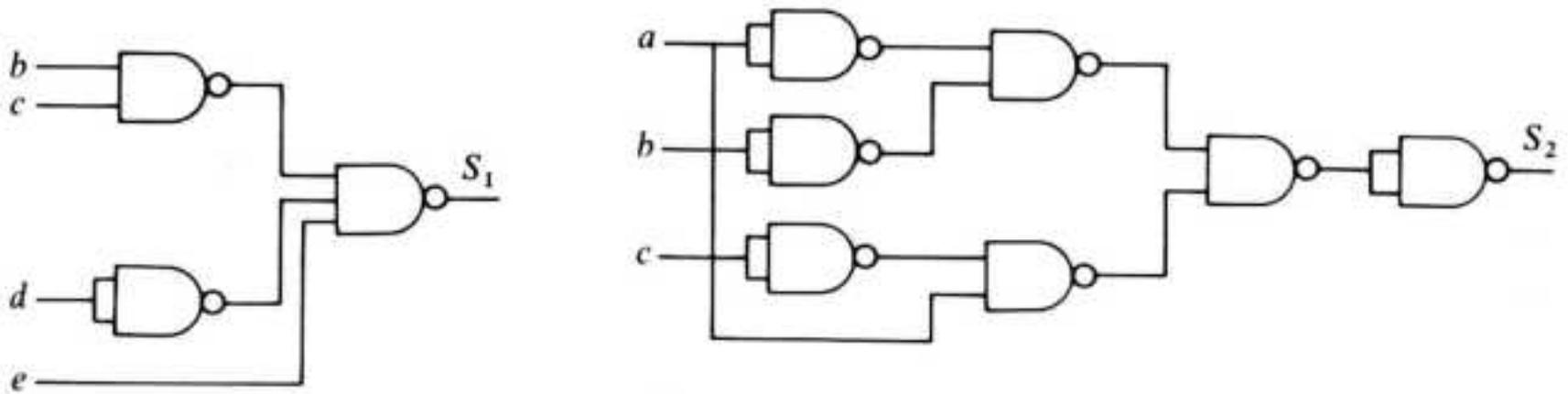


Figura 4.5. Diagramas lógicos de las funciones S_1 y S_2 formados por puertas NAND.

4.4.2. IMPLEMENTACION DE FUNCIONES MEDIANTE PUERTAS NOR

El proceso que debe seguirse es semejante al expuesto en el apartado anterior:

- Se aplica una *doble inversión*.
- Si la expresión original es una *suma* lógica, no se opera ninguna inversión. Si es un *producto*, se elimina una de ellas por aplicación del *teorema de De Morgan*.
- Se continúa invirtiendo doblemente los términos hasta que todas las *sumas* y *productos* se hayan convertido en sumas negadas.

Ejemplos:

$$1. S_1 = b \cdot c + d + \bar{e}; S_1 = \overline{\overline{b \cdot c + d + \bar{e}}} = \overline{\overline{b \cdot c} + \overline{d + \bar{e}}} = \overline{\overline{b} + \overline{c} + d + \bar{e}}$$

$$2. S_2 = (a + b) \cdot (c + \bar{a}); S_2 = \overline{\overline{(a + b) \cdot (c + \bar{a})}} = \overline{\overline{a + b} + \overline{c + \bar{a}}}$$

En la Figura 4.6 se representan los circuitos correspondientes.

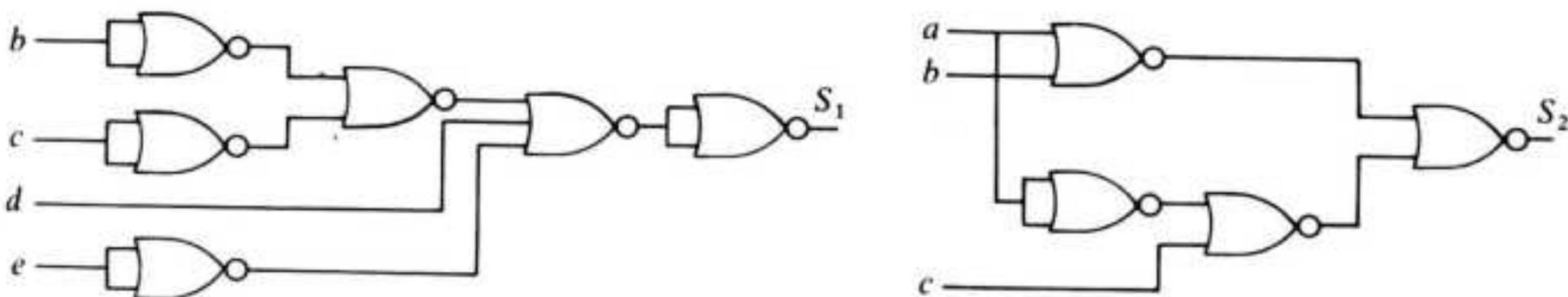


Figura 4.6. Diagramas lógicos de las funciones S_1 y S_2 formados por puertas NOR.

4.5. CARACTERÍSTICAS GENERALES DE LAS PUERTAS INTEGRADAS

Son muchas las tecnologías de fabricación de circuitos digitales integrados. Sin embargo, nos vamos a ocupar exclusivamente de las dos más importantes: la **TTL** y la **CMOS**. Estas son las más utilizadas y las que ofrecen una mayor variedad de bloques.

Los principales parámetros que los fabricantes indican en las hojas de características para todas las familias son:

- a) *La tensión de alimentación y su tolerancia.*
- b) *La temperatura máxima de trabajo.*
- c) *Fan-out o abanico de salida.* Consiste en el número máximo de entradas de otras puertas que se pueden conectar a una salida.
- d) *Niveles de entrada y salida.* Se indican los valores de tensión de los estados lógicos 1 y 0.
 - V_{IL} es la tensión de entrada a nivel bajo.
 - V_{IH} es la tensión de entrada a nivel alto.
 - V_{OL} es la tensión de salida a nivel bajo.
 - V_{OH} es la tensión de salida a nivel alto.
- e) *Margen de ruido* en los estados lógicos 1 y 0. Indica las variaciones máximas que se pueden producir a la entrada sin que la salida varíe su estado.
- f) *Tiempo de propagación medio.* Es el retraso, es decir, el tiempo que transcurre desde que se produce el cambio lógico a la entrada hasta que lo hace a la salida.
- g) *Disipación de potencia.* Normalmente se indica la disipación por función.

También se proporcionan los consumos de corriente de alimentación y los de entrada y salida para los dos valores lógicos.

Cada una de las familias lógicas tiene sus ventajas y sus inconvenientes frente a las demás, por este motivo, en cada caso, se elegirá la más adecuada al diseño que se vaya a desarrollar.

Las características ideales de una familia lógica integrada serían las siguientes:

- Gran densidad de integración.
- Alta velocidad de conmutación (tiempo de propagación bajo).
- Mínimo consumo.
- Máxima inmunidad al ruido y a las variaciones de temperatura.
- Compatibilidad con otras familias lógicas.
- Bajo coste.

4.5.1. FAMILIA LOGICA TTL

Las siglas TTL son las iniciales de *Transistor-Transistor-Logic*, que traducido quiere decir *Lógica Transistor, Transistor*. Las puertas están construidas mediante resis-

tencias, diodos y transistores bipolares. El nombre de Transistor-Transistor le viene dado por el transistor multiemisor que posee.

Esta familia es aún la más popular, debido a su bajo coste y a la gran variedad de circuitos que se han desarrollado por la práctica totalidad de los fabricantes de semiconductores.

En esta tecnología se fabrican, además de las puertas lógicas estudiadas en el presente capítulo, otros circuitos de mayor complejidad, tales como decodificadores, contadores, etc., que analizaremos en capítulos posteriores. La escala de integración en estos casos es la MSI.

La familia TTL comprende varias series que han sido desarrolladas progresivamente para mejorar algunas de las características de las fabricadas con anterioridad.

La primera serie que se creó fue la denominada **TTL Standard**, que es conocida por la mayoría de los fabricantes como la **serie 54/74**, y cuyas principales características son:

- *Tensión de alimentación* comprendida entre 4,5 y 5,5 V (5 V nominales).
- *Temperatura de trabajo* de 0 a 70 °C.
- *Fan-out* igual a 10.
- *Niveles de tensión*:
 - V_{IH} mín. = 2,0 V.
 - V_{IL} máx. = 0,8 V.
 - V_{OH} mín. = 2,4 V.
 - V_{OL} máx. = 0,4 V.
- *Margen de ruido* en ambos niveles, 0,4 V.
- *Tiempo de propagación medio*, 10 nsgs.
- *Disipación de potencia*, 10 mW por función.

La serie 54 presenta prácticamente las mismas características. Se diferencia fundamentalmente en la temperatura de trabajo, que en este caso está comprendida entre -55°C y 125°C . Esta serie se reserva para aplicaciones especiales. Su precio es más elevado que el de la serie 74.

Tomando como referencia la serie estándar y con el fin de mejorar, principalmente, los tiempos de conmutación o la disipación de potencia o ambas cosas a la vez, los fabricantes de circuitos integrados lógicos han desarrollado las siguientes series:

- a) **Serie 54/74L (Low-power)**, obteniéndose menor consumo (1 mW por función) a costa de sacrificar el tiempo de propagación, que en este caso pasa a ser de 33 nsgs.
- b) Mediante la incorporación de un componente denominado diodo Schottky se crea la **serie 54/74S (Schottky)**, que mejora el tiempo de conmutación (3 nsgs), obteniéndose una disipación de potencia por puerta de 19 mW aproximadamente.
- c) Posteriormente se desarrolla la **serie 54/74LS (Low-power, Schottky)**, cuyas características son:

- Potencia disipada por puerta, 2 mW.
 - Tiempo de conmutación, 5 nsgs.
- d) Por último, SIGNETIC ha sacado al mercado, como familia de los años 80, la **serie 54/74F (Fast)**, con una disipación por puerta de 4 mW y un tiempo de propagación de 3 nsgs.

4.5.2. FAMILIA LOGICA CMOS

Su nombre se debe a la utilización de un componente básico denominado transistor **MOS (Metal-Oxide-Semiconductor)**.

Los circuitos integrados de la familia CMOS ofrecen una fuerte competencia a los de tecnología TTL, debido a las mejores características que presentan en algunos aspectos. La principal ventaja es la menor disipación de potencia por función, lo que supone una mayor densidad de integración. Por otra parte, esta familia tiene una mayor inmunidad al ruido eléctrico que la TTL. Por el contrario, los tiempos de propagación, en general, son superiores y el número de bloques integrados disponibles es menor.

La tecnología CMOS es la síntesis de otras dos familias que utilizan también el transistor MOS como elemento básico. Estas son la **NMOS**, constituida por transistores de canal N, y la **PMOS**, cuyo elemento fundamental es el transistor MOS de canal P.

La letra C, que forma parte de las siglas CMOS, es la abreviatura de COS (Complementary Symetry). Esto quiere decir que la familia de circuitos que estamos analizando utiliza una estructura heterogénea, mezcla de la NMOS y de la PMOS.

Las características más significativas de esta familia son:

- *Tensión de alimentación* variable entre 3 y 18 V.
- *Rango de temperatura* comprendido entre -40°C y 85°C .
- *Fan-out* generalmente superior a 50.
- *Niveles de tensión* (para una tensión de alimentación de 5 V):
 - V_{IH} mín. = 3,5 V.
 - V_{IL} máx. = 1,5 V.
 - V_{OH} mín. = 4,95 V.
 - V_{OL} máx. = 0,05 V.
- *Gran inmunidad al ruido*; no le afectan impulsos del 30 por 100 de la tensión de alimentación.
- *Los tiempos de propagación* varían inversamente a la tensión de alimentación, siendo de 125 nsgs para 5 V y de 45 nsgs para 15 V.
- *La potencia disipada por puerta* es de 10 nW.

La familia CMOS básica aparece en los catálogos como **serie 4000**, y en ella se incluyen, además de puertas lógicas, otros dispositivos de mayor complejidad, tales

como contadores, registros, memorias, microprocesadores, etc. Las escalas de integración en estos casos son la MSI, la LSI y la VLSI.

Como en la familia estudiada anteriormente, dentro de la tecnología CMOS se incluye un conjunto de series desarrolladas con posterioridad a la serie básica. Todas ellas son compatibles con las de tecnología TTL e intentan aproximarse a ésta en cuanto a tiempo de propagación, sin perder las características propias de la familia CMOS. De entre ellas destaca la **serie High-Speed CMOS**, que para una alimentación de 4,5 V presenta un tiempo de propagación de 7 nsgs aproximadamente.

4.5.3. ESTUDIO COMPARATIVO DE LAS FAMILIAS TTL Y CMOS

A lo largo de los apartados anteriores ya se han establecido comparaciones entre ambas familias. Trataremos de resumir aquí, mediante una tabla, las diferencias fundamentales entre las *familias TTL y la CMOS*.

Como hemos tenido ocasión de comprobar, los fabricantes han puesto todo su empeño en mejorar el tiempo de propagación o conmutación y la potencia disipada por cada función, con el fin de integrar en un solo chip el máximo número de componentes y de reducir los tiempos de operación de los cada vez más complejos circuitos.

En la tabla de la Figura 4.7 se muestran exclusivamente estas dos características referidas a las principales series de cada una de las dos familias examinadas.

<i>Serie</i>	<i>Potencia disipada</i>	<i>Tiempo de conmutación</i>
TTL	10 mW	10 nsgs
TTL LP	1 mW	33 nsgs
TTL S	19 mW	3 nsgs
TTL LS	2 mW	5 nsgs
TTL F	4 mW	3 nsgs
CMOS	10 nW	125 nsgs
H-S CMOS	—	7 nsgs

Figura 4.7. Tabla comparativa de las principales series de las familias TTL y MOS.

Los valores que se ofrecen en esta tabla están obtenidos, en todos los casos, en las mismas condiciones de tensión, alimentación, temperatura, etc.

En resumen, la tecnología CMOS ofrece, en general, ventajas en cuanto a la tensión de alimentación, la temperatura de trabajo, la inmunidad al ruido, el *fan-out* y la potencia disipada. En cuanto a la tecnología TTL, hay que indicar que por

ser más antigua se encuentra más desarrollada, contando con un mayor número de circuitos. Además, los precios de éstos se mantienen a un nivel más bajo. La ventaja principal de esta familia respecto a la CMOS es, sin embargo, el reducido tiempo de propagación.

4.6. LOGICA POSITIVA Y LOGICA NEGATIVA

La función lógica que realiza una puerta integrada depende de que se trabaje con lógica positiva o con lógica negativa.

Lógica positiva es aquella en la cual el nivel de tensión para el estado lógico uno es mayor que para el estado cero. Por el contrario, en **lógica negativa** el nivel de tensión correspondiente al estado uno es menor que el del estado cero (Figura 4.8).

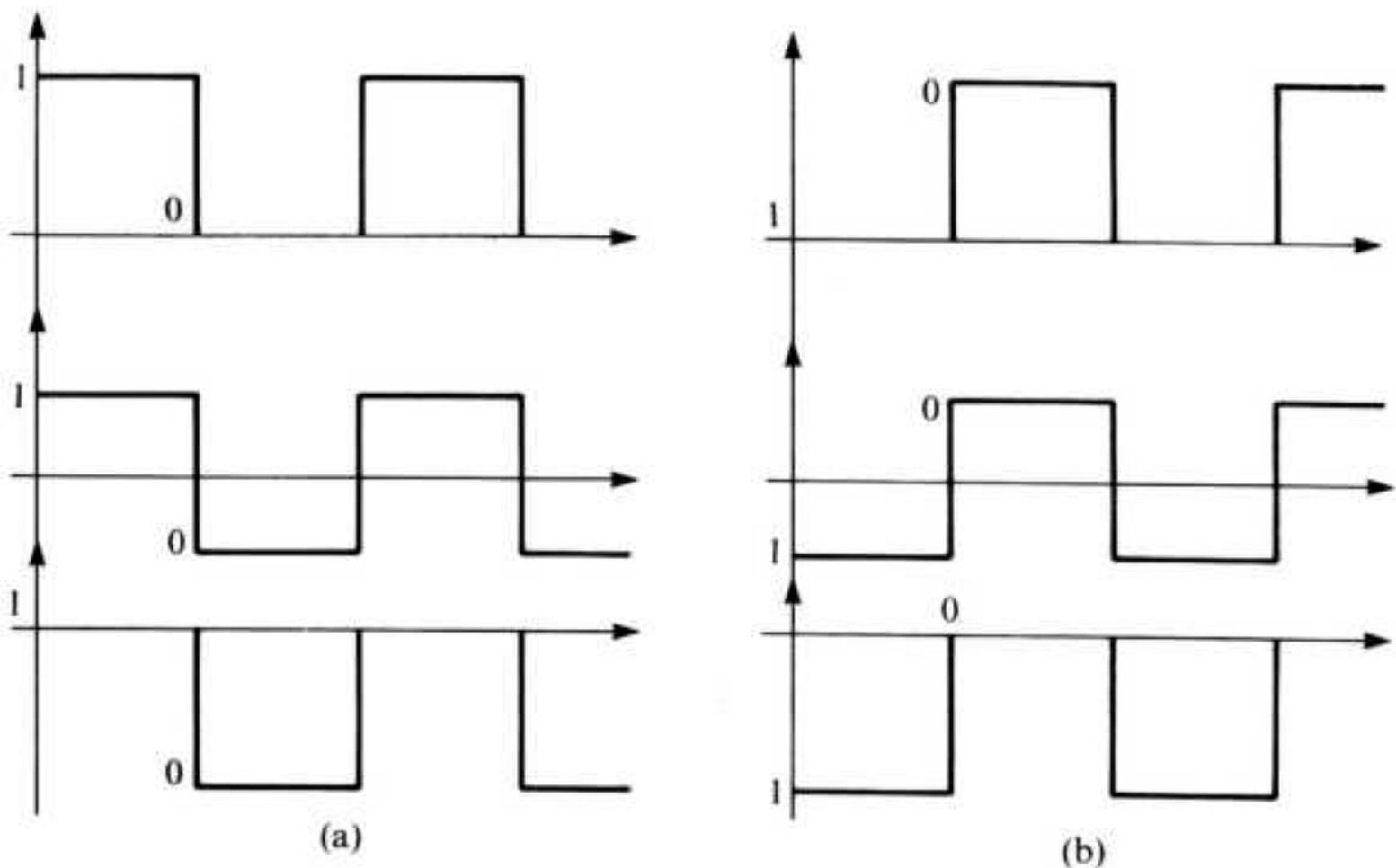


Figura 4.8. Diversas señales digitales y sus correspondientes niveles en lógica positiva (a) y en lógica negativa (b).

Para ilustrar la diferencia entre ambas formas de trabajo expondremos un ejemplo: supongamos que los niveles de tensión con los que vamos a trabajar son 0 y 5 V, y que la salida de una puerta se comporta conforme a los valores indicados en la Figura 4.9.

En lógica positiva, los estados uno y cero se corresponden con los niveles de 5 y 0 V respectivamente. Por tanto, si sustituimos los valores de tensión de la

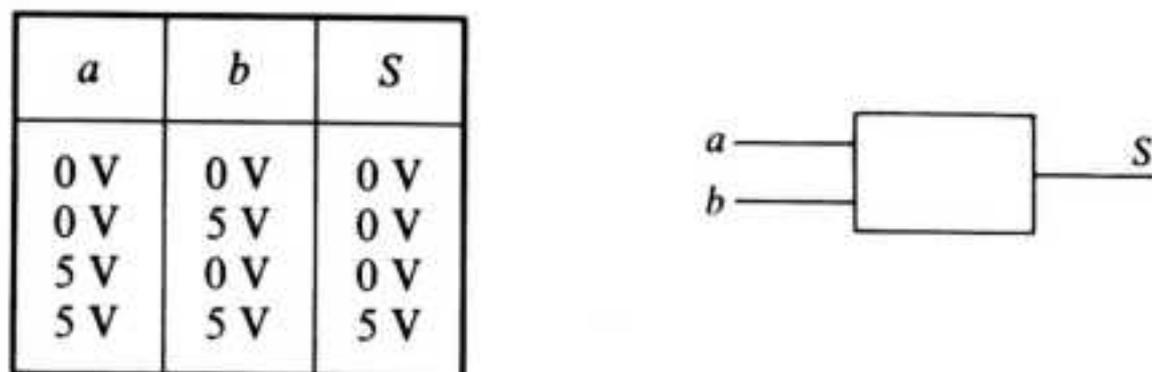


Figura 4.9. Tabla de la verdad y representación simbólica de una puerta de dos entradas.

Figura 4.9 por los estados correspondientes, obtenemos la tabla de la Figura 4.10, que, como podemos comprobar, es la equivalente a una función AND.

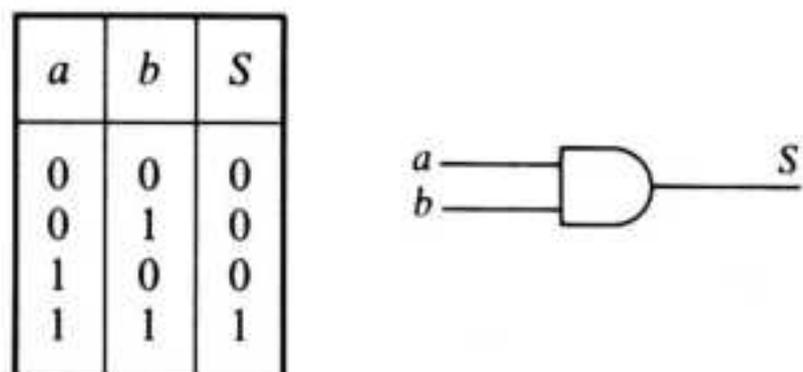


Figura 4.10. Tabla de la verdad y símbolo lógico de una puerta AND.

En lógica negativa tenemos las siguientes correspondencias entre niveles de tensión y estados:

- Estado uno < > 0 V.
- Estado cero < > 5 V.

Sustituyendo estos valores en la tabla de la Figura 4.9 obtenemos la tabla de la Figura 4.11, que en este caso corresponde a una función OR.

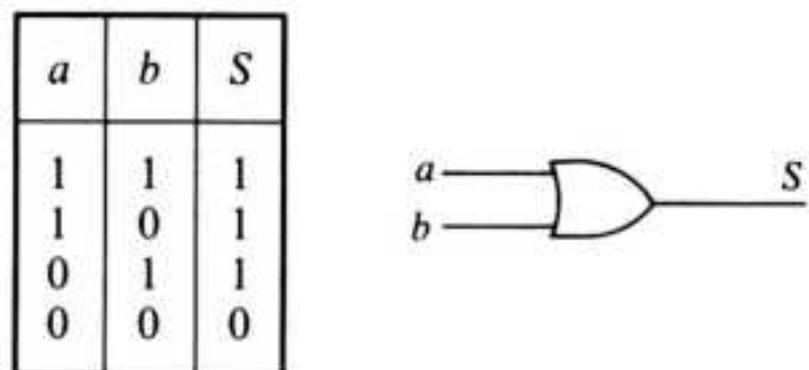


Figura 4.11. Tabla de la verdad y símbolo lógico de una puerta OR.

Por esta razón, para que las tablas de la verdad que aparecen en los catálogos sean válidas, tanto si se trabaja en lógica positiva como en lógica negativa, tienen la forma de la Figura 4.12, donde la L (Low) indica el nivel bajo de tensión y la H (High) el nivel alto.

a	b	S
L	L	L
L	H	L
H	L	L
H	H	H

L: bajo
H: alto

Figura 4.12. Tabla de la verdad válida para lógica positiva y para lógica negativa.

EJERCICIOS RESUELTOS

1. Dibujar el circuito lógico de las siguientes funciones:

a) $F_1 = (a \cdot b \cdot \bar{c} + \bar{a} \cdot c) \cdot d$

b) $F_2 = \overline{(a_1 + a_2)} \cdot \bar{b} \cdot \bar{c}$

c) $F_3 = \overline{[(a + b) \cdot (\bar{a} + \bar{b}) \cdot (b + c)]}$

Solución

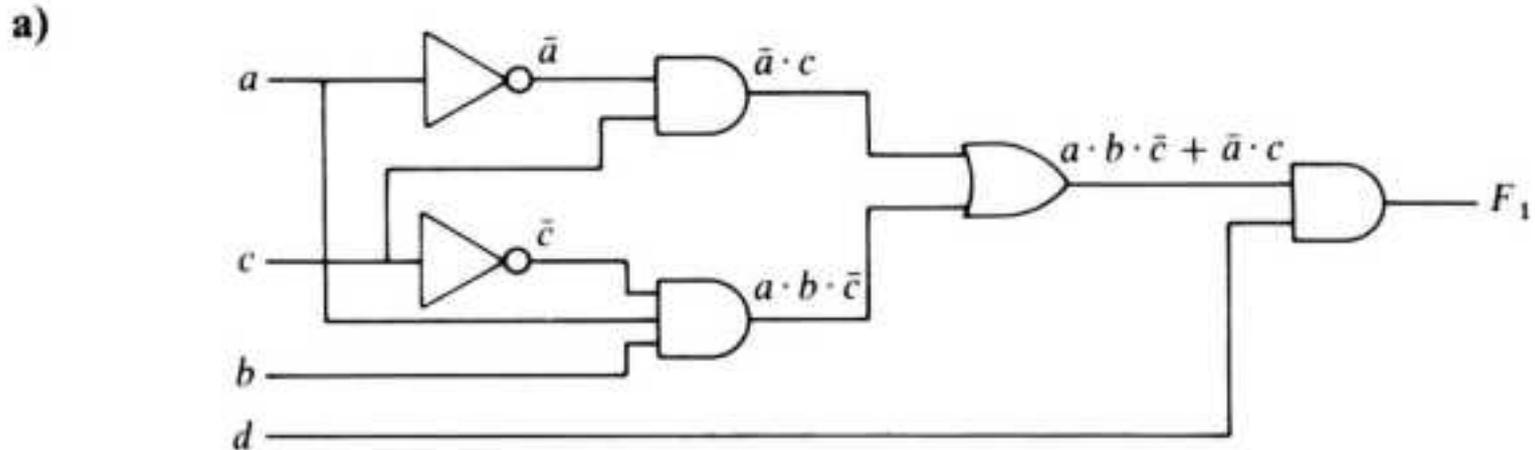


Figura 4.13. Diagrama lógico de la función F_1 del ejercicio 1.

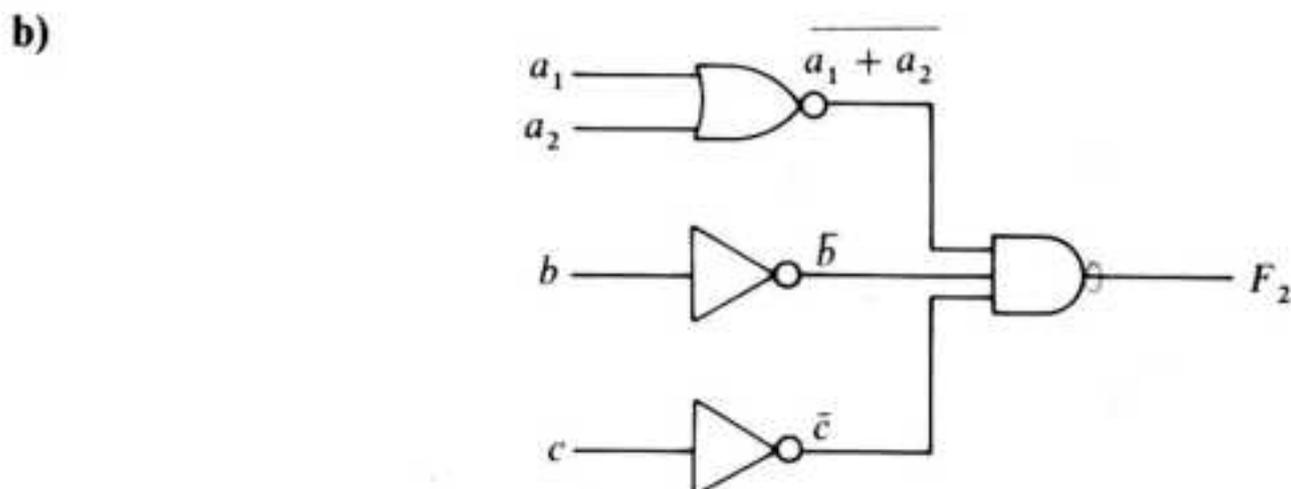


Figura 4.14. Diagrama lógico de la función F_2 del ejercicio 1.

c) Obsérvese que $(a + b) \cdot (\bar{a} + \bar{b}) = a \oplus b$, es decir, es la función O exclusiva expresada en forma de producto de sumas; por tanto:

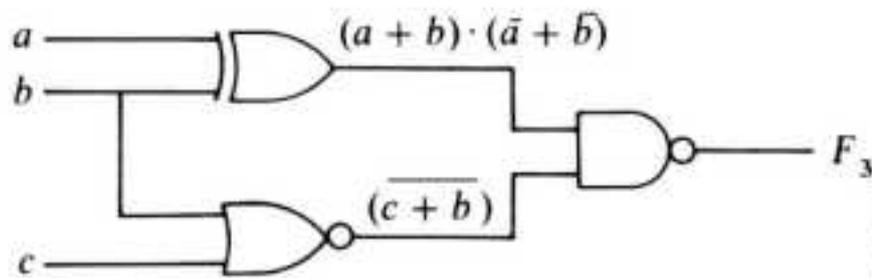


Figura 4.15. Diagrama lógico de la función F_3 del ejercicio 1.

2. Implementar con puertas NAND de dos entradas las siguientes funciones:

a) $F_1 = \overline{a \cdot b \cdot \bar{c}} + \overline{a \cdot \bar{b} \cdot c}$

b) $F_2 = \overline{(\bar{a} + b) \cdot (\bar{a} + \bar{b} + c)}$

Solución

a) $F_1 = \overline{a \cdot b \cdot \bar{c}} + \overline{a \cdot \bar{b} \cdot c} = \overline{a \cdot b \cdot \bar{c} \cdot a \cdot \bar{b} \cdot c} = \overline{\overline{\overline{a \cdot b \cdot \bar{c} \cdot a \cdot \bar{b} \cdot c}}}$

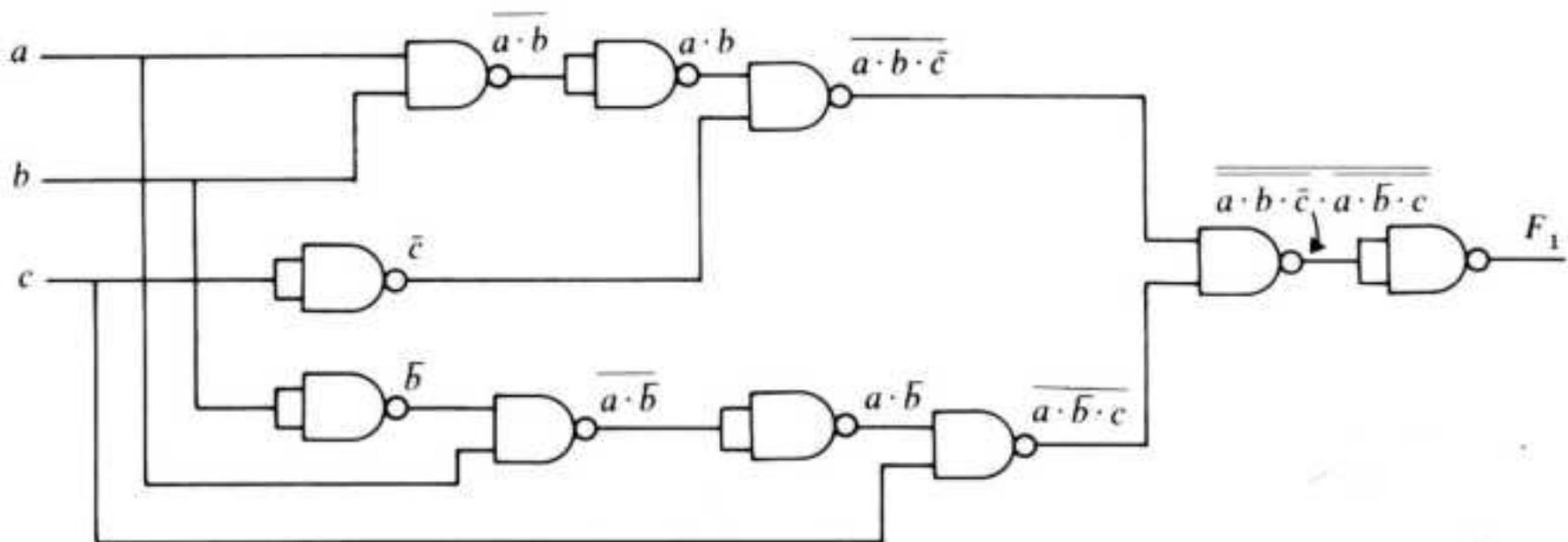


Figura 4.16. Diagrama lógico de la función F_1 del ejercicio 2.

b) $F_2 = \overline{(\bar{a} + b) \cdot (\bar{a} + \bar{b} + c)} = \overline{\overline{\overline{(\bar{a} + b) \cdot (\bar{a} + \bar{b} + c)}}} = \overline{\overline{a \cdot \bar{b} \cdot a \cdot b \cdot \bar{c}}}$

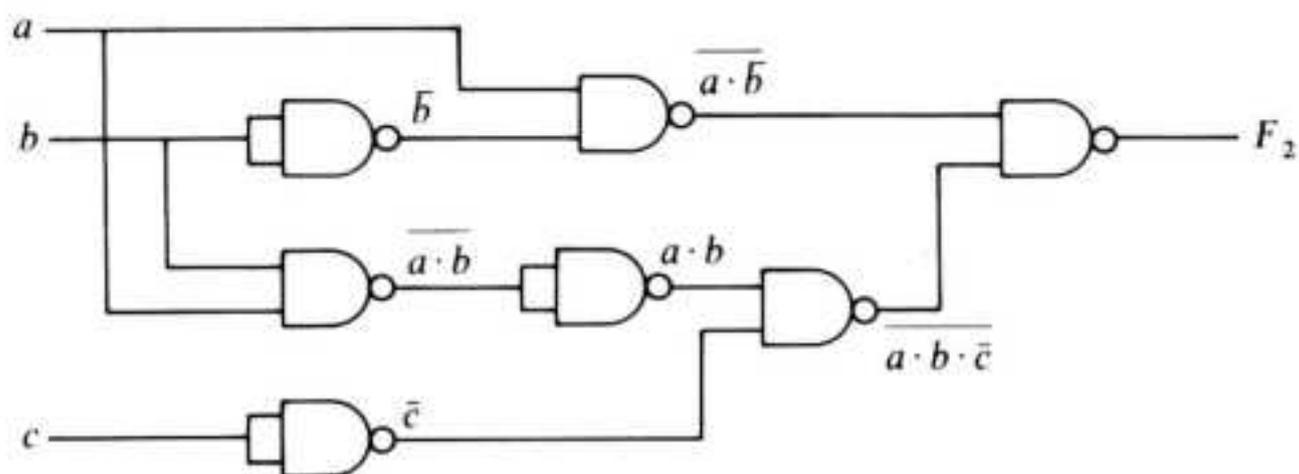


Figura 4.17. Diagrama lógico de la función F_2 del ejercicio 2.

3. Implementar con puertas NOR de dos entradas las siguientes funciones:

a) $F_1 = \overline{a \cdot b \cdot c \cdot \bar{a} \cdot \bar{c}}$

b) $F_2 = \overline{(a + b) \cdot (a + b + c)}$

Solución

a) $F_1 = \overline{a \cdot b \cdot c \cdot \bar{a} \cdot \bar{c}} = \overline{\overline{\overline{\overline{a \cdot b \cdot c \cdot \bar{a} \cdot \bar{c}}}}} = \overline{\overline{a \cdot b \cdot c} + \overline{\bar{a} \cdot \bar{c}}} =$
 $= \overline{\overline{a \cdot b \cdot c} + \overline{\bar{a} \cdot \bar{c}}} = \overline{\bar{a} + b + \bar{c} + a + c}$

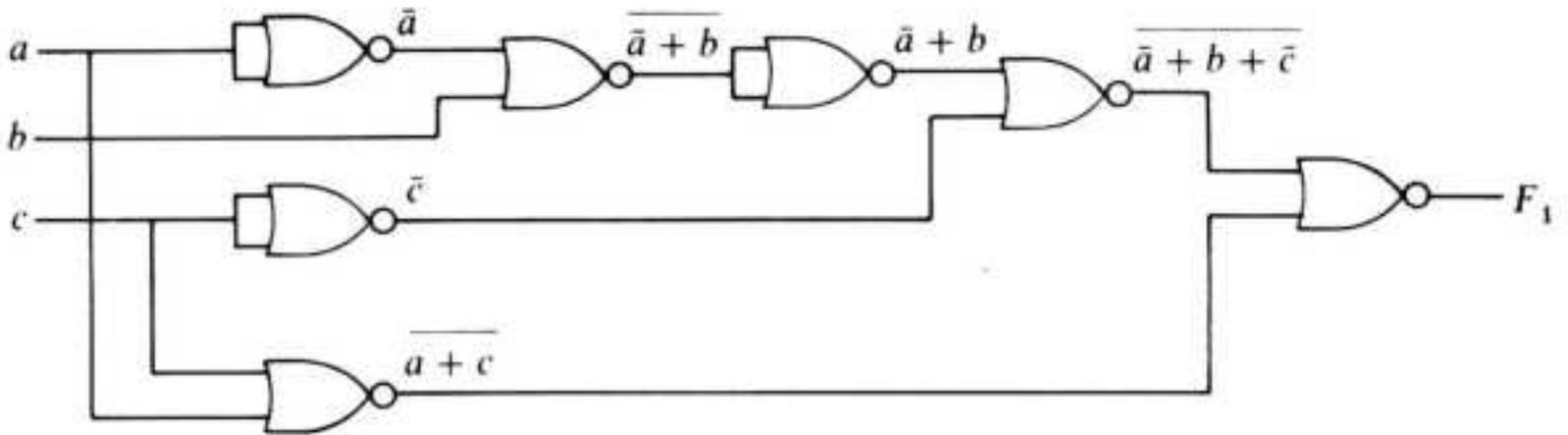


Figura 4.18. Diagrama lógico de la función F_1 del ejercicio 3.

b) $F_2 = \overline{(a + b) \cdot (a + b + c)} = \overline{\overline{\overline{\overline{(a + b) \cdot (a + b + c)}}}} = \overline{\overline{\bar{a} + b} + \overline{\bar{a} + b + c}}$

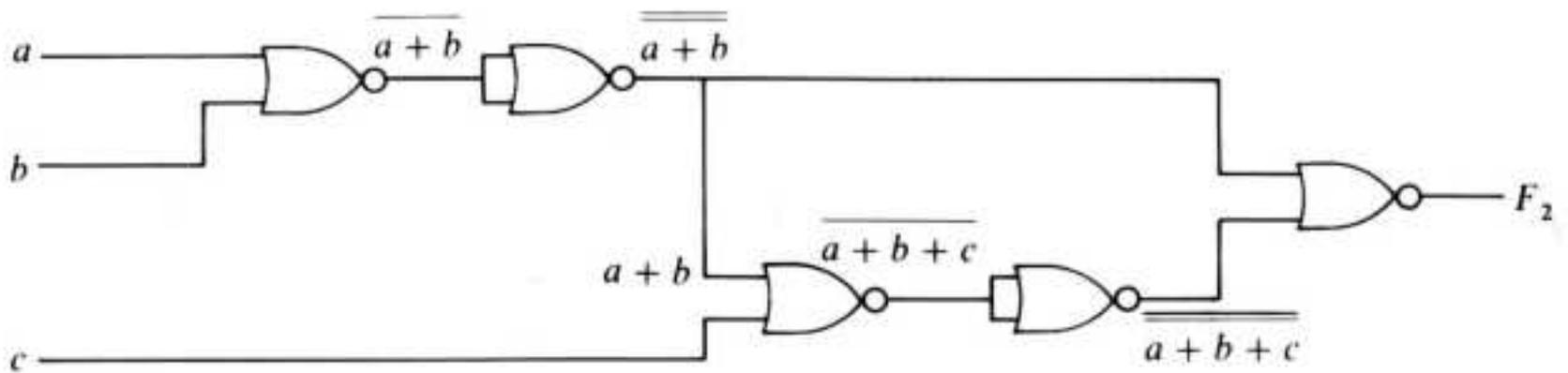


Figura 4.19. Diagrama lógico de la función F_2 del ejercicio 3.

4. El siguiente ejemplo sintetiza la teoría expuesta en los Capítulos 2, 3 y 4. Supongamos una prensa que se pone en marcha mediante la actuación simultánea de tres pulsadores: P_1 , P_2 y P_3 (un solo contacto cada uno). Si se pulsaran solamente dos cualesquiera, la prensa funcionará, pero se activará una lámpara indicando una manipulación incorrecta. Cuando se pulse un solo dispositivo, también se encenderá la lámpara, pero no se activará la prensa.

Diseñar el circuito de control correspondiente mediante puertas NAND de tecnología TTL.

Solución

El primer paso es traducir el enunciado del problema en una tabla de la verdad. Figura 4.20.

<i>a</i>	<i>b</i>	<i>c</i>	<i>P</i>	<i>L</i>
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	0

P = Prensa
L = Lámpara

a, *b* y *c* son las variables del problema. Estas dependerán del estado de los pulsadores *P*₁, *P*₂ y *P*₃.

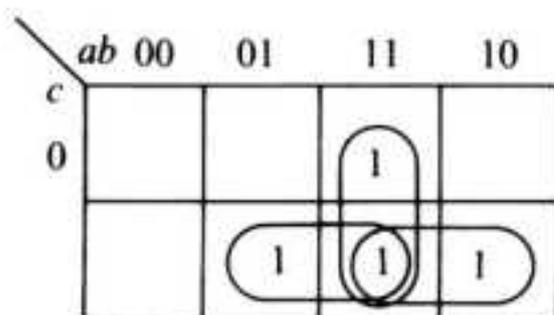
Figura 4.20. Tabla de la verdad del ejercicio 4.

Las funciones obtenidas de la tabla serán:

$$P = \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c} + a \cdot b \cdot c$$

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

y la simplificación por el método de Karnaugh:



$$P = a \cdot b + b \cdot c + a \cdot c$$

Figura 4.21. Simplificación por el método de Karnaugh de la función *P*.

La función *L* no se puede simplificar.

Para utilizar puertas NAND exclusivamente transformaremos las funciones:

$$P = \overline{\overline{a \cdot b \cdot b \cdot c \cdot a \cdot c}}$$

$$L = \overline{\overline{\overline{\bar{a} \cdot \bar{b} \cdot \bar{c} \cdot a \cdot b \cdot c}}}$$

Por último, dibujaremos (Figura 4.22) el circuito lógico. Para su construcción serán suficientes tres bloques integrados de la familia TTL, dos 7400 y uno 7410.

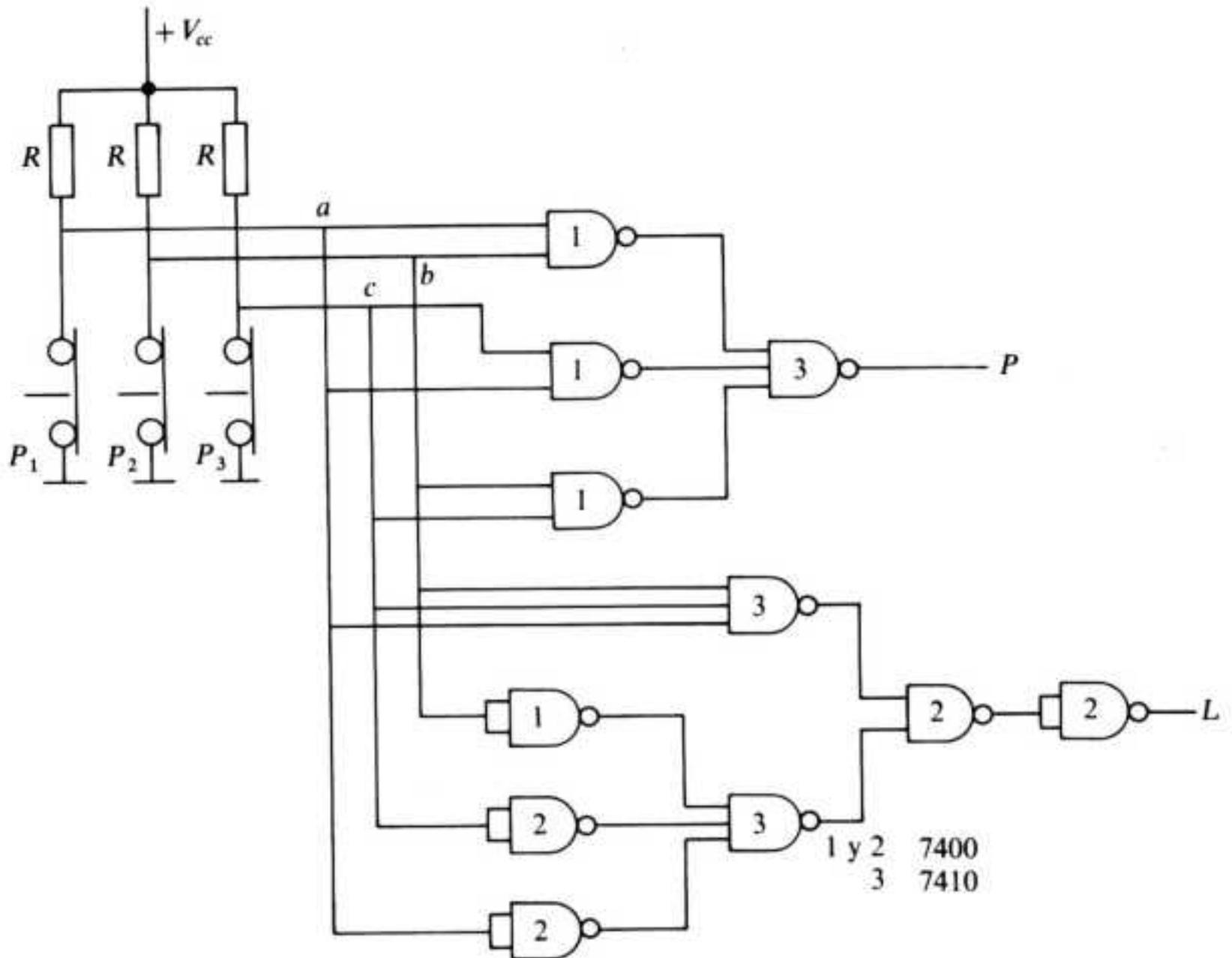


Figura 4.22. Diagrama lógico del ejercicio 4.

EJERCICIOS PROPUESTOS

1. Simplificar, utilizando el método más adecuado, y después dibujar el circuito lógico con puertas integradas:

a) $a \cdot b \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot b \cdot \bar{c} \cdot \bar{d} + a \cdot b \cdot \bar{c} \cdot d + \bar{a} \cdot b \cdot \bar{c} \cdot d$

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

c) $a \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot \bar{b} \cdot c + a \cdot b \cdot c$

2. Construir en primer lugar con puertas NAND y después con puertas NOR las siguientes funciones:

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

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

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

3. En la Figura 4.23 se representa un sistema de apertura de la puerta de un garaje particular.

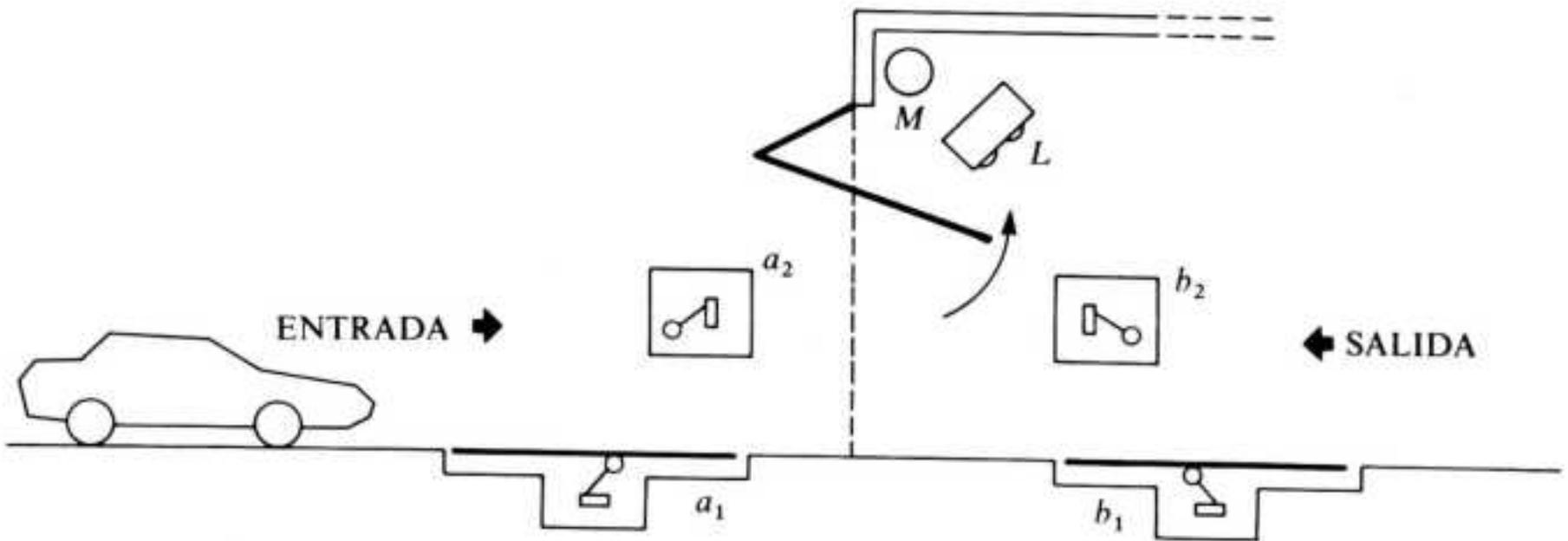


Figura 4.23.

Para que la puerta se abra es necesario que estén activados los interruptores a_1 y a_2 o los b_1 y b_2 simultáneamente.

El cierre de la puerta se produce automáticamente, transcurrido un tiempo. Existe una protección para evitar que la puerta baje cuando haya un coche debajo. La lámpara L permanecerá encendida cuando la puerta esté cerrada.

Diseñar el circuito necesario para la apertura de la puerta, exclusivamente, y para el control de la lámpara, utilizando puertas integradas.

Apéndice: Hojas de características de circuitos integrados de las diferentes tecnologías analizadas: TTL y CMOS

Incluimos en este apéndice las hojas de características extraídas de catálogo de la firma NATIONAL SEMICONDUCTOR de los siguientes dispositivos o bloques integrados:

- Tecnología TTL:
 - DM5400/DM7400. Cuádruple puerta NAND de dos entradas.
 - DM5402/DM7402. Cuádruple puerta NOR de dos entradas.
- Tecnología CMOS:
 - CD4001M/CD4001C. Cuádruple puerta NOR de dos entradas.
 - CD4011M/CD4011C. Cuádruple puerta NAND de dos entradas.

En las ocho páginas siguientes se reproducen tal y como figuran en el databook las características de los dispositivos mencionados.



DM5400/DM7400 Quad 2-Input NAND Gates

General Description

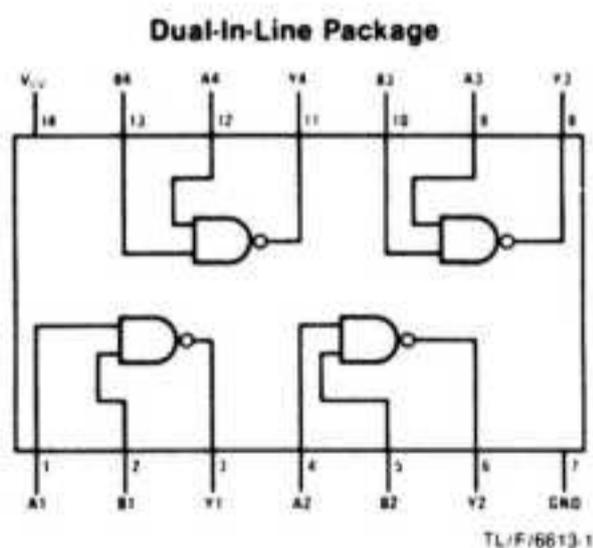
This device contains four independent gates each of which performs the logic NAND function.

Absolute Maximum Ratings (Note 1)

Supply Voltage	7V
Input Voltage	5.5V
Storage Temperature Range	- 65 °C to 150 °C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device can not be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Connection Diagram



DM5400 (J) DM7400 (N)

Function Table

$$Y = \overline{AB}$$

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

H = High Logic Level
L = Low Logic Level

Recommended Operating Conditions

Symbol	Parameter	DM5400			DM7400			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.8			0.8	V
I_{OH}	High Level Output Current			-0.4			-0.4	mA
I_{OL}	Low Level Output Current			16			16	mA
T_A	Free Air Operating Temperature	-55		125	0		70	°C

Electrical Characteristics over recommended operating free air temperature (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 1)	Max	Units	
V_I	Input Clamp Voltage	$V_{CC} = \text{Min}$, $I_I = -12 \text{ mA}$			-1.5	V	
V_{OH}	High Level Output Voltage	$V_{CC} = \text{Min}$, $I_{OH} = \text{Max}$ $V_{IL} = \text{Max}$	2.4	3.4		V	
V_{OL}	Low Level Output Voltage	$V_{CC} = \text{Min}$, $I_{OL} = \text{Max}$ $V_{IH} = \text{Min}$		0.2	0.4	V	
I_I	Input Current @ Max Input Voltage	$V_{CC} = \text{Max}$, $V_I = 5.5 \text{ V}$			1	mA	
I_{IH}	High Level Input Current	$V_{CC} = \text{Max}$, $V_I = 2.4 \text{ V}$			40	μA	
I_{IL}	Low Level Input Current	$V_{CC} = \text{Max}$, $V_I = 0.4 \text{ V}$			-1.6	mA	
I_{OS}	Short Circuit Output Current	$V_{CC} = \text{Max}$ (Note 2)	DM54	-20		-55	mA
			DM74	-18		-55	
I_{CCH}	Supply Current With Outputs High	$V_{CC} = \text{Max}$		4	8	mA	
I_{CCL}	Supply Current With Outputs Low	$V_{CC} = \text{Max}$		12	22	mA	

Switching Characteristics at $V_{CC} = 5 \text{ V}$ and $T_A = 25^\circ\text{C}$ (See Section 1 for Test Waveforms and Output Load)

Parameter	Conditions	$C_L = 15 \text{ pF}$ $R_L = 400\Omega$			Units
		Min	Typ	Max	
t_{PLH} Propagation Delay Time Low to High Level Output			12	22	ns
t_{PHL} Propagation Delay Time High to Low Level Output			7	15	ns

Note 1: All typicals are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

Note 2: Not more than one output should be shorted at a time.



DM5402/DM7402 Quad 2-Input NOR Gates

General Description

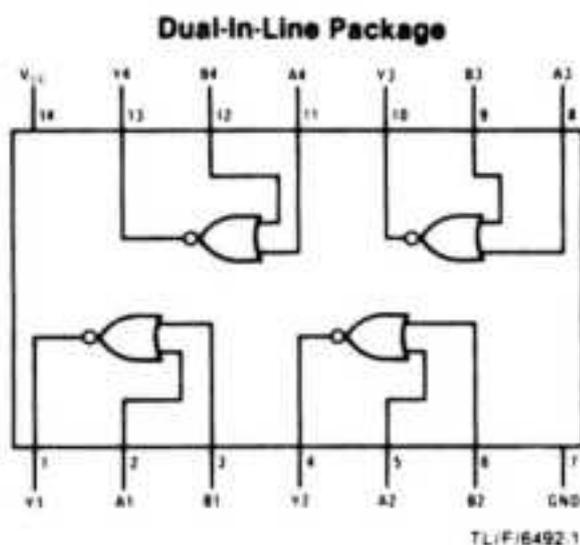
This device contains four independent gates each of which performs the logic NOR function.

Absolute Maximum Ratings (Note 1)

Supply Voltage	7V
Input Voltage	5.5V
Storage Temperature Range	- 65°C to 150°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device can not be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Connection Diagram



DM5402 (J) DM7402 (N)

Function Table

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

Inputs		Output
A	B	Y
L	L	H
L	H	L
H	L	L
H	H	L

H = High Logic Level
L = Low Logic Level

Recommended Operating Conditions

Symbol	Parameter	DM5402			DM7402			Units
		Min	Nom	Max	Min	Nom	Max	
V_{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V_{IH}	High Level Input Voltage	2			2			V
V_{IL}	Low Level Input Voltage			0.8			0.8	V
I_{OH}	High Level Output Current			-0.4			-0.4	mA
I_{OL}	Low Level Output Current			16			16	mA
T_A	Free Air Operating Temperature	-55		125	0		70	°C

Electrical Characteristics over recommended operating free air temperature (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 1)	Max	Units	
V_I	Input Clamp Voltage	$V_{CC} = \text{Min}, I_I = -12 \text{ mA}$			-1.5	V	
V_{OH}	High Level Output Voltage	$V_{CC} = \text{Min}, I_{OH} = \text{Max}$ $V_{IL} = \text{Max}$	2.4	3.4		V	
V_{OL}	Low Level Output Voltage	$V_{CC} = \text{Min}, I_{OL} = \text{Max}$ $V_{IH} = \text{Min}$		0.2	0.4	V	
I_I	Input Current @ Max Input Voltage	$V_{CC} = \text{Max}, V_I = 5.5 \text{ V}$			1	mA	
I_{IH}	High Level Input Current	$V_{CC} = \text{Max}, V_I = 2.4 \text{ V}$			40	μA	
I_{IL}	Low Level Input Current	$V_{CC} = \text{Max}, V_I = 0.4 \text{ V}$			-1.6	mA	
I_{OS}	Short Circuit Output Current	$V_{CC} = \text{Max}$ (Note 2)	DM54	-20		-55	mA
			DM74	-18		-55	
I_{CCH}	Supply Current With Outputs High	$V_{CC} = \text{Max}$		8	16	mA	
I_{CCL}	Supply Current With Outputs Low	$V_{CC} = \text{Max}$		14	27	mA	

Switching Characteristics at $V_{CC} = 5 \text{ V}$ and $T_A = 25^\circ\text{C}$ (See Section 1 for Test Waveforms and Output Load)

Parameter	Conditions	$C_L = 15 \text{ pF}$ $R_L = 400\Omega$			Units
		Min	Typ	Max	
t_{PLH} Propagation Delay Time Low to High Level Output			12	22	ns
t_{PHL} Propagation Delay Time High to Low Level Output			8	15	ns

Note 1: All typicals are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

Note 2: Not more than one output should be shorted at a time.



CD4001M/CD4001C Quadruple 2-Input NOR Gate CD4011M/CD4011C Quadruple 2-Input NAND Gate

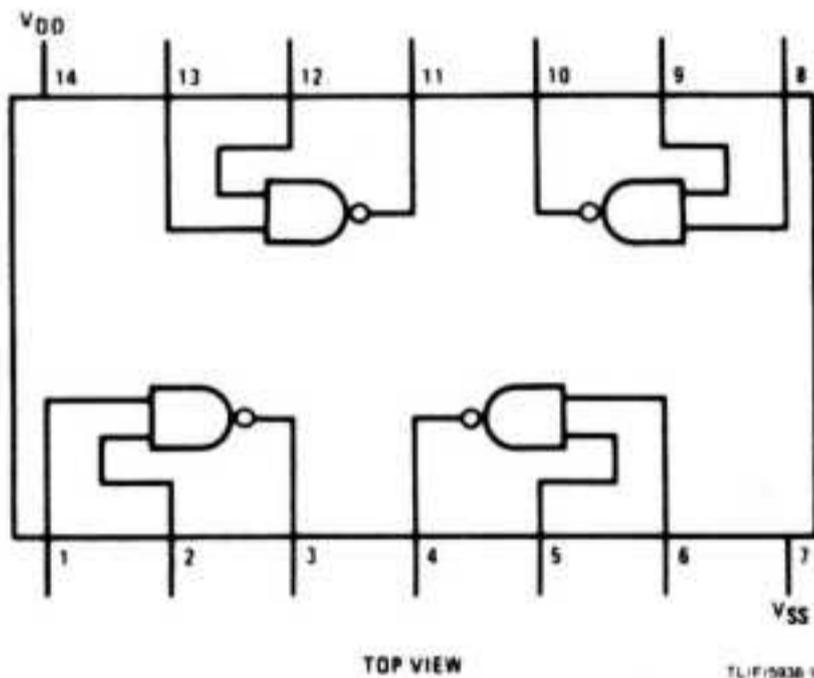
General Description

The CD4001M/CD4001C, CD4011M/CD4011C are monolithic complementary MOS (CMOS) quadruple two-input NOR and NAND gate integrated circuits. N- and P-channel enhancement mode transistors provide a symmetrical circuit with output swings essentially equal to the supply voltage. This results in high noise immunity over a wide supply voltage range. No DC power other than that caused by leakage current is consumed during static conditions. All inputs are protected against static discharge and latching conditions.

Features

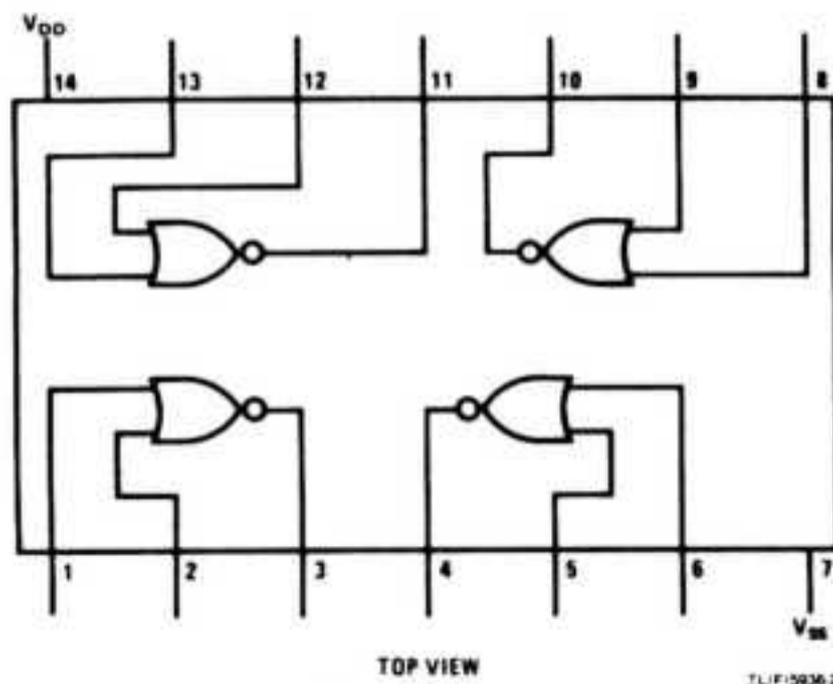
- Wide supply voltage range 3.0 V to 15 V
- Low power 10 nW (typ.)
- High noise immunity 0.45 V_{DD} (typ.)

Connection Diagrams Dual-In-Line Packages



Order Number CD4001MJ, CD4001CJ,
CD4011MJ or CD4011CJ
See NS Package J14A

Order Number CD4001MN, CD4001CN,
CD4011MN or CD4011CN
See NS Package N14A



Absolute Maximum Ratings (Note 1)

Voltage an Any Pin $V_{SS} - 0.3V$ to $V_{DD} + 0.3V$
 Operating Temperature Range
 CD4001M, CD4011M $-55^{\circ}C$ to $+125^{\circ}C$
 CD4001C, CD4011C $-40^{\circ}C$ to $+85^{\circ}C$

Storage Temperature Range $-65^{\circ}C$ to $+150^{\circ}C$
 Package Dissipation 500 mW
 Operating V_{DD} Range $V_{SS} + 3.0V$ to $V_{SS} + 15V$
 Lead Temperature (Soldering, 10 seconds) $260^{\circ}C$

DC Electrical Characteristics — CD4001M, CD4011M

Sym	Parameter	Conditions	Limits						Units	
			$-55^{\circ}C$		$25^{\circ}C$			$125^{\circ}C$		
			Min	Max	Min	Typ	Max	Min		Max
I_L	Quiescent Device Current	$V_{DD} = 5.0V$ $V_{DD} = 10V$		0.05		0.001	0.05		3.0	μA
				0.1		0.001	0.1		6.0	μA
P_D	Quiescent Device Dissipation/Package	$V_{DD} = 5.0V$ $V_{DD} = 10V$		0.25		0.005	0.25		15	μW
				1.0		0.01	1.0		60	μW
V_{OL}	Output Voltage Low Level	$V_{DD} = 5.0V, V_I = V_{DD}, I_O = 0A$ $V_{DD} = 10V, V_I = V_{DD}, I_O = 0A$		0.05		0	0.05		0.05	V
				0.05		0	0.05		0.05	V
V_{OH}	Output Voltage High Level	$V_{DD} = 5.0V, V_I = V_{SS}, I_O = 0A$ $V_{DD} = 10V, V_I = V_{SS}, I_O = 0A$	4.95		4.95	5.0		4.95		V
			9.95		9.95	10		9.95		V
V_{NL}	Noise Immunity (All Inputs)	$V_{DD} = 5.0V, V_O = 3.6V, I_O = 0A$ $V_{DD} = 10V, V_O = 7.2V, I_O = 0A$	1.5		1.5	2.25		1.4		V
			3.0		3.0	4.5		2.9		V
V_{NH}	Noise Immunity (All Inputs)	$V_{DD} = 5.0V, V_O = 0.95V, I_O = 0A$ $V_{DD} = 10V, V_O = 2.9V, I_O = 0A$	1.4		1.5	2.25		1.5		V
			2.9		3.0	4.5		3.0		V
I_{DN}	Output Drive Current N-Channel (4001) (Note 2)	$V_{DD} = 5.0V, V_O = 0.4V, V_I = V_{DD}$ $V_{DD} = 10V, V_O = 0.5V, V_I = V_{DD}$	0.5		0.40	1.0		0.28		mA
			1.1		0.9	2.5		0.65		mA
I_{DP}	Output Drive Current P-Channel (4001) (Note 2)	$V_{DD} = 5.0V, V_O = 2.5V, V_I = V_{SS}$ $V_{DD} = 10V, V_O = 9.5V, V_I = V_{SS}$	-0.62		-0.5	-2.0		-0.35		mA
			-0.62		-0.5	-1.0		-0.35		mA
I_{DN}	Output Drive Current N-Channel (4011) (Note 2)	$V_{DD} = 5.0V, V_O = 0.4V, V_I = V_{DD}$ $V_{DD} = 10V, V_O = 0.5V, V_I = V_{DD}$	0.31		0.25	0.5		0.175		mA
			0.63		0.5	0.6		0.35		mA
I_{DP}	Output Drive Current P-Channel (4011) (Note 2)	$V_{DD} = 5.0V, V_O = 2.5V, V_I = V_{SS}$ $V_{DD} = 10V, V_O = 9.5V, V_I = V_{SS}$	-0.31		-0.25	-0.5		-0.175		mA
			-0.75		-0.6	-1.2		-0.4		mA
I_i	Input Current				10				pA	

DC Electrical Characteristics — CD4001C, CD4011C

Sym	Parameter	Conditions	Limits						Units	
			-40°C		25°C			80°C		
			Min	Max	Min	Typ	Max	Min		Max
I_L	Quiescent Device Current	$V_{DD} = 5.0V$ $V_{DD} = 10V$		0.5 5.0		0.005 0.005	0.5 5.0		15 30	μA μA
P_D	Quiescent Device Dissipation/Package	$V_{DD} = 5.0V$ $V_{DD} = 10V$		2.5 50		0.025 0.05	2.5 50		75 300	μW μW
V_{OL}	Output Voltage Low Level	$V_{DD} = 5.0V, V_I = V_{DD}, I_O = 0A$ $V_{DD} = 10V, V_I = V_{DD}, I_O = 0A$		0.05 0.05		0 0	0.05 0.05		0.05 0.05	V V
V_{OH}	Output Voltage High Level	$V_{DD} = 5.0V, V_I = V_{SS}, I_O = 0A$ $V_{DD} = 10V, V_I = V_{SS}, I_O = 0A$	4.95 9.95		4.95 9.95	5.0 10		4.95 9.95		V V
V_{NL}	Noise Immunity (All Inputs)	$V_{DD} = 5.0V, V_O = 3.6V, I_O = 0A$ $V_{DD} = 10V, V_O = 7.2V, I_O = 0A$	1.5 3.0		1.5 3.0	2.25 4.5		1.4 2.9		V V
V_{NH}	Noise Immunity (All Inputs)	$V_{DD} = 5.0V, V_O = 0.95V, I_O = 0A$ $V_{DD} = 10V, V_O = 2.9V, I_O = 0A$	1.4 2.9		1.5 3.0	2.25 4.5		1.5 3.0		V V
I_{DN}	Output Drive Current N-Channel (4001) (Note 2)	$V_{DD} = 5.0V, V_O = 0.4V, V_I = V_{DD}$ $V_{DD} = 10V, V_O = 0.5V, V_I = V_{DD}$	0.35 0.72		0.3 0.6	1.0 2.5		0.24 0.48		mA mA
I_{DP}	Output Drive Current P-Channel (4001) (Note 2)	$V_{DD} = 5.0V, V_O = 2.5V, V_I = V_{SS}$ $V_{DD} = 10V, V_O = 9.5V, V_I = V_{SS}$	-0.35 -0.3		-0.3 -0.25	-2.0 -1.0		-0.24 -0.2		mA mA
I_{DN}	Output Drive Current N-Channel (4011) (Note 2)	$V_{DD} = 5.0V, V_O = 0.4V, V_I = V_{DD}$ $V_{DD} = 10V, V_O = 0.5V, V_I = V_{DD}$	0.145 0.3		0.12 0.25	0.5 0.8		0.095 0.2		mA mA
I_{DP}	Output Drive Current P-Channel (4011) (Note 2)	$V_{DD} = 5.0V, V_O = 2.5V, V_I = V_{SS}$ $V_{DD} = 10V, V_O = 9.5V, V_I = V_{SS}$	-0.145 -0.35		-0.12 -0.3	-0.5 -1.2		-0.095 -0.24		mA mA
I_I	Input Current					10				μA

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: I_{DN} and I_{DP} are tested one output at a time.

AC Electrical Characteristics $T_A = 25^\circ\text{C}$, $C_L = 15\text{ pF}$, and input rise and fall times = 20 ns.
 Typical temperature coefficient for all values of $V_{DD} = 0.3\%/^\circ\text{C}$

Sym	Parameter	Conditions	Min	Typ	Max	Units
CD4001M						
t_{PHL}	Propagation Delay Time High to Low Level	$V_{DD} = 5.0\text{V}$		35	50	ns
		$V_{DD} = 10\text{V}$		25	40	ns
t_{PLH}	Propagation Delay Time Low to High Level	$V_{DD} = 5.0\text{V}$		35	65	ns
		$V_{DD} = 10\text{V}$		25	40	ns
t_{THL}	Transition Time High to Low Level	$V_{DD} = 5.0\text{V}$		65	125	ns
		$V_{DD} = 10\text{V}$		35	70	ns
t_{TLH}	Transition Time Low to High Level	$V_{DD} = 5.0\text{V}$		65	175	ns
		$V_{DD} = 10\text{V}$		35	75	ns
C_{IN}	Input Capacitance	Any Input		5.0		pF
CD4001C						
t_{PHL}	Propagation Delay Time High to Low Level	$V_{DD} = 5.0\text{V}$		35	80	ns
		$V_{DD} = 10\text{V}$		25	55	ns
t_{PLH}	Propagation Delay Time Low to High Level	$V_{DD} = 5.0\text{V}$		35	120	ns
		$V_{DD} = 10\text{V}$		25	65	ns
t_{THL}	Transition Time High to Low Level	$V_{DD} = 5.0\text{V}$		65	200	ns
		$V_{DD} = 10\text{V}$		35	115	ns
t_{TLH}	Transition Time Low to High Level	$V_{DD} = 5.0\text{V}$		65	300	ns
		$V_{DD} = 10\text{V}$		35	125	ns
C_{IN}	Input Capacitance	Any Input		5.0		pF

AC Electrical Characteristics $T_A = 25^\circ\text{C}$, $C_L = 15\text{ pF}$, and input rise and fall times = 20 ns.
 Typical temperature coefficient for all values of $V_{DD} = 0.3\%/^\circ\text{C}$

Sym	Parameter	Conditions	Min	Typ	Max	Units
CD4011M						
t_{PHL}	Propagation Delay Time High to Low Level	$V_{DD} = 5.0\text{V}$		50	75	ns
		$V_{DD} = 10\text{V}$		25	40	ns
t_{PLH}	Propagation Delay Time Low to High Level	$V_{DD} = 5.0\text{V}$		50	75	ns
		$V_{DD} = 10\text{V}$		25	40	ns
t_{THL}	Transition Time High to Low Level	$V_{DD} = 5.0\text{V}$		75	125	ns
		$V_{DD} = 10\text{V}$		50	75	ns
t_{TLH}	Transition Time Low to High Level	$V_{DD} = 5.0\text{V}$		75	100	ns
		$V_{DD} = 10\text{V}$		40	60	ns
C_{IN}	Input Capacitance	Any Input		5.0		pF
CD4011C						
t_{PHL}	Propagation Delay Time High to Low Level	$V_{DD} = 5.0\text{V}$		50	100	ns
		$V_{DD} = 10\text{V}$		25	50	ns
t_{PLH}	Propagation Delay Time Low to High Level	$V_{DD} = 5.0\text{V}$		50	100	ns
		$V_{DD} = 10\text{V}$		25	50	ns
t_{THL}	Transition Time High to Low Level	$V_{DD} = 5.0\text{V}$		75	150	ns
		$V_{DD} = 10\text{V}$		50	100	ns
t_{TLH}	Transition Time Low to High Level	$V_{DD} = 5.0\text{V}$		75	125	ns
		$V_{DD} = 10\text{V}$		40	75	ns
C_{IN}	Input Capacitance	Any Input		5.0		pF

PARTE
SEGUNDA

Circuitos digitales MSI

Circuitos combinacionales

5.1. CARACTERISTICAS DE LOS CIRCUITOS COMBINACIONALES

En este capítulo estudiaremos exclusivamente los circuitos combinacionales que son de aplicación general. Sin embargo, cualquier diagrama lógico, construido a partir de una o varias funciones como las expuestas en capítulos anteriores, también es un circuito de idénticas características. Los ejemplos de síntesis del tema anterior son una buena muestra de esta última afirmación.

Un **circuito combinacional** es *aquel que está formado por funciones lógicas elementales que tiene un número de entradas y otro de salidas y los valores de éstas dependen exclusivamente del estado que adopten las entradas y de su constitución interna*. El proceso para el diseño de un circuito combinacional es el descrito en el apartado 4.1. Siempre hemos de comenzar por definir la tabla de la verdad.

Nos ocuparemos en este capítulo del desarrollo, mediante puertas, de sencillos circuitos combinacionales de aplicación general. Con ello pretendemos desarrollar la capacidad de diseño.

Analizaremos también los circuitos combinacionales construidos en tecnología MSI, con el fin de familiarizar al lector con las hojas de características de estos bloques integrados.

A la hora de diseñar un sistema o circuito complejo, dado el gran número de bloques integrados MSI disponibles en catálogo, es preferible utilizar estos circuitos directamente antes que construirlos mediante simples puertas.

Son ejemplos de circuitos combinacionales de aplicación general: los codificadores, decodificadores, multiplexadores, demultiplexadores, comparadores, generadores y detectores de paridad y convertidores de código. De todo este tipo de funciones, así como de los dispositivos que las realizan, nos ocupamos a continuación.

5.2. CODIFICACION Y DECODIFICACION

Todos los circuitos digitales funcionan mediante la aplicación a sus entradas de señales digitales. Las salidas también proporcionan señales eléctricas de la misma forma.

Este tipo de señales están formadas exclusivamente, como vimos en el Capítulo 1, por dos *niveles* de tensión (*nivel alto* y *nivel bajo*) que se corresponden con los dos posibles estados estables (corte y saturación) de los elementos electrónicos básicos que constituyen las puertas y demás bloques integrados.

Cualquier información que se desee tratar, procesar o almacenar mediante sistemas digitales, deberá ser traducida o **codificada** en un tipo de lenguaje apropiado. La forma correcta de hacerlo es convertir cualquier número, letra, signo, instrucción u operación en un conjunto de señales eléctricas digitales, que será diferente en cada caso. Cada uno de los datos estará constituido por una serie de unos y ceros que indicarán niveles altos o bajos de tensión, respectivamente.

El número 9 (decimal), por ejemplo, podría representarse por 1001, que es su equivalente en el sistema binario. De la misma forma, a la hora de interpretar un resultado a la salida de un circuito digital, es necesario traducir o **decodificar** el resultado transformando los ceros y los unos en datos que sean inteligibles.

La codificación y la decodificación serán siempre operaciones necesarias en sistemas digitales que traten información, o en procesos industriales donde sea necesario suministrar datos o presentar resultados. En algunos sistemas cibernéticos o de control es posible prescindir de este tipo de operaciones, siendo suficiente la aplicación de señales digitales mediante traductores y aplicar las salidas del circuito sobre elementos de potencia tales como lámparas, motores, etc.

5.2.1. SISTEMA BINARIO

Aunque anteriormente hemos hecho alguna referencia a la equivalencia entre el *sistema decimal* y el *sistema binario*, este es el momento de efectuar su análisis, porque su conocimiento es fundamental para estudiar cualquier tipo de codificación.

El sistema en *base dos* utiliza únicamente los símbolos 0 y 1. A cada cifra o dígito de un número binario se le denomina **bit**, abreviatura de **Binary Digit**.

Un número binario puede presentarse de forma polinómica:

$$a_n b^n + a_{n-1} b^{n-1} + \dots + a_1 b^1 + a_0 b^0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \dots$$

Los coeficientes $a_n, a_{n-1}, \dots, a_1, a_0$, etc., representan ordenadamente las cifras del número, y b es la base del sistema de numeración, en nuestro caso $b = 2$.

Si los términos se expresan en base 10 y se suman todos ellos, se puede obtener el equivalente decimal del número binario al que representa.

Ejemplos:

a) Pasar el número 101101 en base dos a decimal:

$$1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 0 + 8 + 4 + 0 + 1 = 45$$

b) Pasar 0,1001 a decimal:

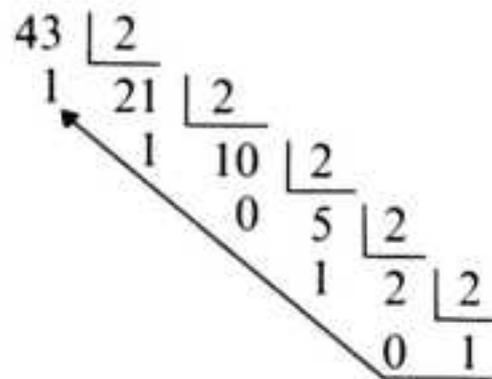
$$1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = \frac{1}{2^1} + \frac{1}{2^4} = 0,5625$$

Para pasar un número decimal entero a binario se realizan divisiones sucesivas por dos hasta que el último cociente sea inferior a dos.

El número binario será el formado por el *último cociente*, que será el bit de mayor peso, y los *restos* de cada división.

Ejemplo:

Pasar 43 decimal a binario:



El número expresado en binario será 101011.

Para convertir a binario un número decimal fraccionario se multiplica éste por dos. La parte decimal del resultado se vuelve a multiplicar por dos, y así sucesivamente hasta que el resultado del producto sea un valor entero o hasta que se obtenga la precisión deseada. El número binario quedará formado por la sucesión de las partes enteras resultantes de los productos.

Ejemplo:

Pasar 0,5625 a binario:

$$\begin{array}{l} 0,5625 \cdot 2 = 1,1250 \\ 0,125 \cdot 2 = 0,250 \\ 0,25 \cdot 2 = 0,5 \\ 0,5 \cdot 2 = 1,0 \end{array} \quad \downarrow$$

El número binario equivalente será 0,1001.

De la suma y resta binaria nos ocuparemos en el próximo capítulo, cuando examinemos los circuitos sumadores.

Otro sistema de numeración muy utilizado también en circuitos digitales es el **hexadecimal**, que desarrollaremos en el Capítulo 9.

5.2.2. CODIGOS

Como hemos indicado anteriormente, la información que tenga que ser procesada mediante circuitos digitales debe ser previamente codificada.

Un **código** es, en general, *un conjunto de unidades de información relacionadas de forma sistemática y biunívoca con otro conjunto de signos y símbolos según unas determinadas reglas de traducción fijadas de antemano*. Los códigos que se utilizan en los sistemas digitales son binarios, es decir, combinaciones de unos y de ceros.

La información que se puede codificar no se limita exclusivamente a cantidades numéricas, sino que se extiende a signos operativos, letras, instrucciones, etc.

Cuando decíamos en el apartado 5.2 que el número decimal 9 se representa en sistema binario como 1001, debe entenderse que el número 9 se codifica de esa manera para ser aplicada la información a un circuito lógico. Cuando establecemos una relación análoga entre los números 0, 1, 2, 3, 4, 5, etc., y un conjunto de combinaciones binarias como la indicada, estamos creando uno de los múltiples códigos posibles.

Los códigos más comunes son:

- a) Binario natural.
- b) BCD (Decimal Codificado en Binario):

- Natural.
- Exceso tres.
- Aiken.

- c) Códigos progresivos.
- d) Códigos detectores y correctores de error.
- e) Códigos alfanuméricos.

a) **Binario natural**. Consiste simplemente en representar, por el método indicado en el apartado 5.2.1, cualquier número decimal mediante la combinación binaria correspondiente.

b) **Códigos BCD**. Esta familia de códigos es la más utilizada para representar información numérica.

Para codificar un número decimal mediante este sistema se representa por separado cada una de sus cifras. La cantidad de bits necesaria para representar cada cifra es de cuatro. Con ellos se pueden efectuar $2^4 = 16$ combinaciones distintas. Como el máximo de signos diferentes en el sistema decimal es de 10, quedan siempre seis combinaciones inutilizadas.

La diferencia entre los códigos de esta familia, Natural, Exceso tres y Aiken, reside en las diez (de las dieciséis) combinaciones distintas que emplea cada uno de ellos.

- En **BCD natural** se utilizan las diez primeras en orden creciente.
- En **BCD exceso tres** no se utilizan las tres primeras ni las tres últimas.
- En **BCD Aiken** se emplean las cinco primeras y las cinco últimas.

En la Figura 5.1 se representan todos ellos y su equivalencia con el sistema decimal.

c) **Códigos progresivos**. La característica fundamental de estos códigos es que una combinación difiere de la anterior y de la siguiente exclusivamente en un solo

<i>Sistema decimal</i>	<i>BCD natural</i>	<i>BCD exceso tres</i>	<i>BCD Aiken</i>
0	0000	0011	0000
1	0001	0100	0001
2	0010	0101	0010
3	0011	0110	0011
4	0100	0111	0100
5	0101	1000	1011
6	0110	1001	1100
7	0111	1010	1101
8	1000	1011	1110
9	1001	1100	1111

Figura 5.1. Equivalencia entre el sistema decimal y los códigos de la familia BCD.

bit. Se emplean en procesos industriales para transformar magnitudes analógicas en digitales. Los códigos más utilizados son los de *Gray*. La diferencia entre unos y otros reside en el número de bit de cada combinación. Para mayores precisiones se emplean códigos con combinaciones más largas. En la Figura 5.2 se muestra el código Gray para cuatro bits.

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

Figura 5.2. Código Gray para cuatro bits.

d) Códigos detectores y correctores de error. Existen códigos más complejos que *detectan*, y en algunos casos *corrigen*, el error en la información cuando ésta es transmitida a través de una red. El error se detecta o corrige si se produce en un solo bit de la combinación. La posibilidad de que se origine en dos bits a la vez es muy remota.

El número mínimo de bits por combinación es de cinco.

Los códigos **detectores** más comunes son los de **paridad** y los de dos entre cinco y dos entre siete o **biquinario**.

Los dos últimos están formados por combinaciones de cinco y siete bits respectivamente, siendo dos el número de unos lógicos en cada combinación en

ambos casos. El error se detecta contando, mediante el circuito adecuado, el número de unos de cada combinación.

Los códigos de paridad se forman añadiendo un bit más a los de la familia BCD. Pueden ser de *paridad par* o *impar*. En el primer caso el número de unos, incluido el de paridad, debe ser par y en el segundo impar. En la Figura 5.3 se muestra el código de paridad impar formado a partir del BCD exceso tres.

0	0011	1
1	0100	0
2	0101	1
3	0110	1
4	0111	0
5	1000	0
6	1001	1
7	1010	1
8	1011	0
9	1100	1

Figura 5.3. Código de paridad impar construido con el BCD exceso tres.

El bit de paridad se genera mediante un circuito combinacional muy sencillo denominado **generador de paridad**, que se construye con puertas O exclusiva.

La detección se realiza comprobando (mediante un *circuito detector*, formado también por puertas O exclusiva) que el número de unos en cada combinación es siempre par o impar, según el caso.

Los códigos **correctores** proporcionan el lugar que ocupa el bit erróneo. Mediante el circuito adecuado se puede corregir automáticamente el fallo detectado en la información recibida. Se utilizan fundamentalmente en procesos industriales.

El código corrector más utilizado es el **HAMMING**, en el cual cada combinación está formada por siete bits, y para su construcción se parte también de los de la familia BCD. En la Figura 5.4 aparece el código HAMMING, formado a partir del BCD natural.

	b_7	b_6	b_5	b_4	b_3	b_2	b_1
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	0	1	1	0	0	1
3	0	0	1	1	1	1	0
4	0	1	0	1	0	1	0
5	0	1	0	1	1	0	1
6	0	1	1	0	0	1	1
7	0	1	1	0	1	0	0
8	1	0	0	1	0	1	1
9	1	0	0	1	1	0	0

Figura 5.4. Código Hamming formado a partir del BCD natural.

Los siete bits están relacionados mediante las siguientes ecuaciones:

$$C_1 = b_1 \oplus b_3 \oplus b_5 \oplus b_7$$

$$C_2 = b_2 \oplus b_3 \oplus b_6 \oplus b_7$$

$$C_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7$$

El número decimal equivalente a la combinación binaria $C_3C_2C_1$ indicará el bit con error.

Las columnas b_7 , b_6 , b_5 y b_3 corresponden al código BCD natural. Las b_1 , b_2 y b_4 se construyen de forma que en cada combinación:

$$b_1 - b_3 - b_5 - b_7$$

$$b_2 - b_3 - b_6 - b_7$$

$$b_4 - b_5 - b_6 - b_7$$

el número de unos sea par, es decir, $C_1 = C_2 = C_3 = 0$.

Cuando no existe error, el valor de las funciones C_1 , C_2 y C_3 será cero.

Para comprender la forma de operar, supongamos que al transmitir el número tres cuyo valor codificado es 0011110 se produce un error y la combinación recibida es la 0011010. El valor de C_1 , C_2 y C_3 será:

$$C_1 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$

$$C_2 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$C_3 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

La combinación $C_3C_2C_1$ será en este caso 011, que equivale al número tres decimal. El bit equivocado es el tercero por la derecha.

e) **Códigos alfanuméricos.** Estos códigos se emplean para representar información de letras, números y signos especiales.

Los más utilizados son los códigos ASCII (American Standard Code for Information Interchange). Existen códigos de 6 y 7 bits. Permite representar cifras decimales, caracteres alfabéticos, signos especiales y diversas órdenes de control para periféricos (impresoras, pantallas, etc.).

En la Figura 5.5 se muestra el código ASCII de 7 bits. El bit b_7 es el más significativo o de mayor peso. Normalmente se antepone a los siete bits propios del código otro de paridad, quedando constituido el formato de cada combinación:

P	b_7	b_6	b_5	b_4	b_3	b_2	b_1
-----	-------	-------	-------	-------	-------	-------	-------

				b_7	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1
b_4	b_3	b_2	b_1		0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	'	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(8	H	X	h	x
1	0	0	1	9	HT	EM)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	
1	1	0	1	13	CR	GS	-	=	M]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	-	o	DEL

Figura 5.5. Código ASCII.

5.2.3. CODIFICADORES

Un **codificador** es un circuito combinacional formado por 2^n entradas y n salidas cuya función es tal que cuando una sola entrada adopta un determinado valor lógico (0 ó 1, según las propiedades del circuito) las salidas representan en binario el número de orden de la entrada que adopte el valor activo.

En la Figura 5.6 aparece la tabla de la verdad de un codificador de $2^3 = 8$ entradas y de tres salidas. El valor activo es el cero.

Entradas								Salidas		
a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	S_2	S_1	S_0
1	1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	0	1	0	0	1
1	1	1	1	1	0	1	1	0	1	0
1	1	1	1	0	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	0	0
1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1

Figura 5.6. Tabla de la verdad de un codificador de ocho entradas y tres salidas.

Los codificadores comerciales construidos en tecnología **MSI** son *prioritarios*, esto quiere decir que la combinación presente a la salida será la correspondiente a la entrada activa de mayor valor decimal.

El valor binario de las salidas puede ser el de cualquiera de los códigos analizados en los apartados anteriores.

El diseño de un codificador se realiza como el de cualquier circuito combinacional. Como ejemplo construiremos el diagrama lógico de un sencillo codificador de cuatro entradas y, por consiguiente, de dos salidas. Comenzaremos por la tabla de la verdad (Figura 5.7).

a_3	a_2	a_1	a_0	S_1	S_0
x	x	x	1	0	0
x	x	1	0	0	1
x	1	0	0	1	0
1	0	0	0	1	1

Figura 5.7. Tabla de la verdad de un codificador de cuatro entradas y dos salidas.

Los valores de las entradas representadas con una x se llaman **términos indiferentes**. Esto quiere decir que tanto si la entrada correspondiente vale cero como si vale uno, las salidas deberán tener el mismo valor en ambos casos. Para obtener las funciones S_0 y S_1 , los términos indiferentes deben tomarse como valor cero y uno.

Si los términos indiferentes aparecen en las columnas de las salidas, pueden adoptar el valor cero o uno según convenga.

Obtenidas las funciones y simplificadas por el método de Karnaugh obtendremos:

$$S_1 = \bar{a}_0 \bar{a}_1 a_2 + \bar{a}_0 \bar{a}_1 a_3 = \bar{a}_0 \bar{a}_1 (a_2 + a_3)$$

$$S_0 = \bar{a}_0 a_1 + \bar{a}_0 \bar{a}_2 a_3 = \bar{a}_0 (a_1 + \bar{a}_2 a_3)$$

El circuito será el de la Figura 5.8.

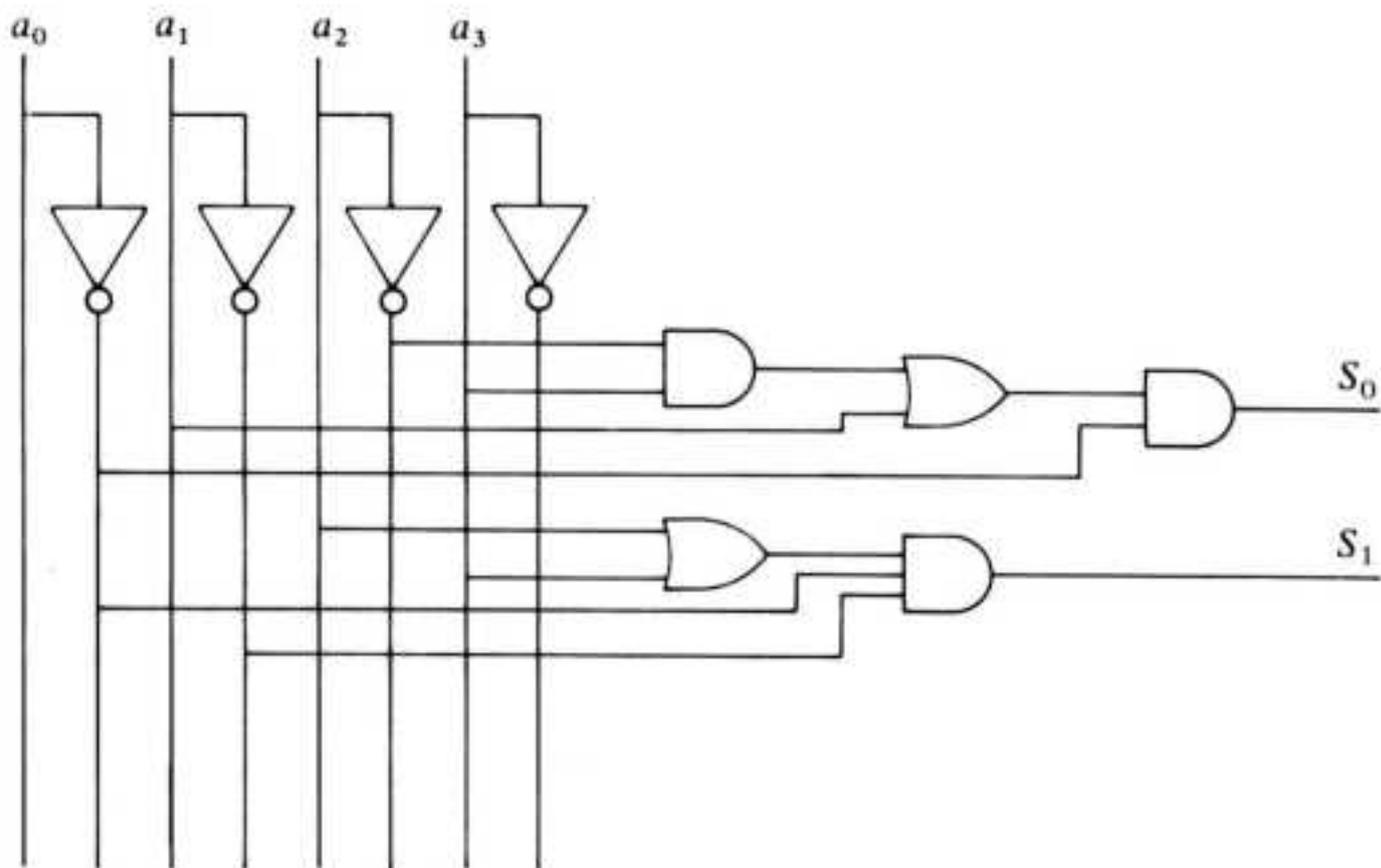


Figura 5.8. Diagrama lógico de un codificador 4/2.

5.2.3.1. Codificador 74LS148

Este circuito construido en tecnología TTL es un codificador que tiene *ocho líneas de entrada y tres de salida*. La principal aplicación es la obtención de un código binario a partir de las líneas procedentes de un teclado.

En la Figura 5.9 se muestra la distribución de terminales y la denominación de cada uno de ellos. Además de las líneas de entrada y salida de datos, dispone de una entrada de *inhibición* \bar{E}_1 de tal forma que su valor debe ser cero para que codifique. Tiene también dos salidas: EO y GS . La primera indica mediante un nivel

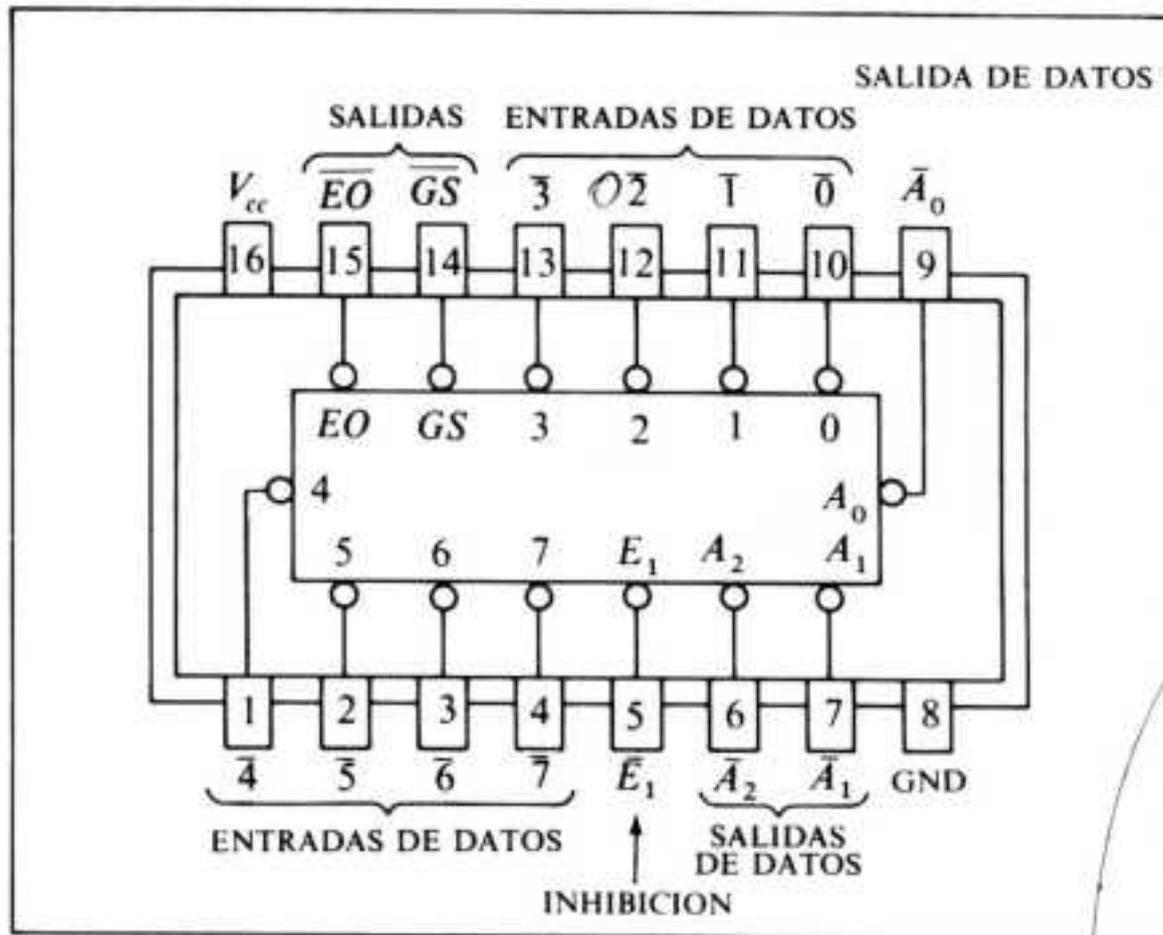


Figura 5.9. Diagrama de conexión del 74LS148.

bajo (*L*) que todas las entradas están a nivel alto (*H*), y la segunda pasa a nivel bajo cuando alguna de las entradas de datos es activa, es decir, cuando el circuito está codificando.

En la Figura 5.10 se representa la tabla de la verdad. Como se puede comprobar, *el nivel activo es el cero*, y las salidas indican, en forma negada, el valor binario correspondiente a la entrada activa.

Entradas									Salidas				
\bar{E}_1	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$	\bar{A}_2	\bar{A}_1	\bar{A}_0	\overline{GS}	\overline{EO}
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

L (bajo)
H (alto)

Figura 5.10. Tabla de la verdad del 74LS148.

5.2.4. DECODIFICADORES

Los decodificadores realizan la función inversa a los codificadores. Un decodificador *selecciona una de las salidas dependiendo de la combinación binaria presente a la entrada.*

En la Figura 5.11 aparece la tabla de la verdad de un decodificador de dos entradas y cuatro salidas.

a_1	a_0	S_3	S_2	S_1	S_0
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

Figura 5.11. Tabla de la verdad de un decodificador de dos entradas y cuatro salidas.

De la tabla obtenemos las siguientes ecuaciones:

$$\bar{S}_0 = \bar{a}_0 \cdot \bar{a}_1 \quad ; \quad S_0 = \overline{\bar{a}_0 \cdot \bar{a}_1}$$

$$\bar{S}_1 = a_0 \cdot \bar{a}_1 \quad ; \quad S_1 = \overline{a_0 \cdot \bar{a}_1}$$

$$\bar{S}_2 = \bar{a}_0 \cdot a_1 \quad ; \quad S_2 = \overline{\bar{a}_0 \cdot a_1}$$

$$\bar{S}_3 = a_0 \cdot a_1 \quad ; \quad S_3 = \overline{a_0 \cdot a_1}$$

El circuito resultante se muestra en la Figura 5.12.

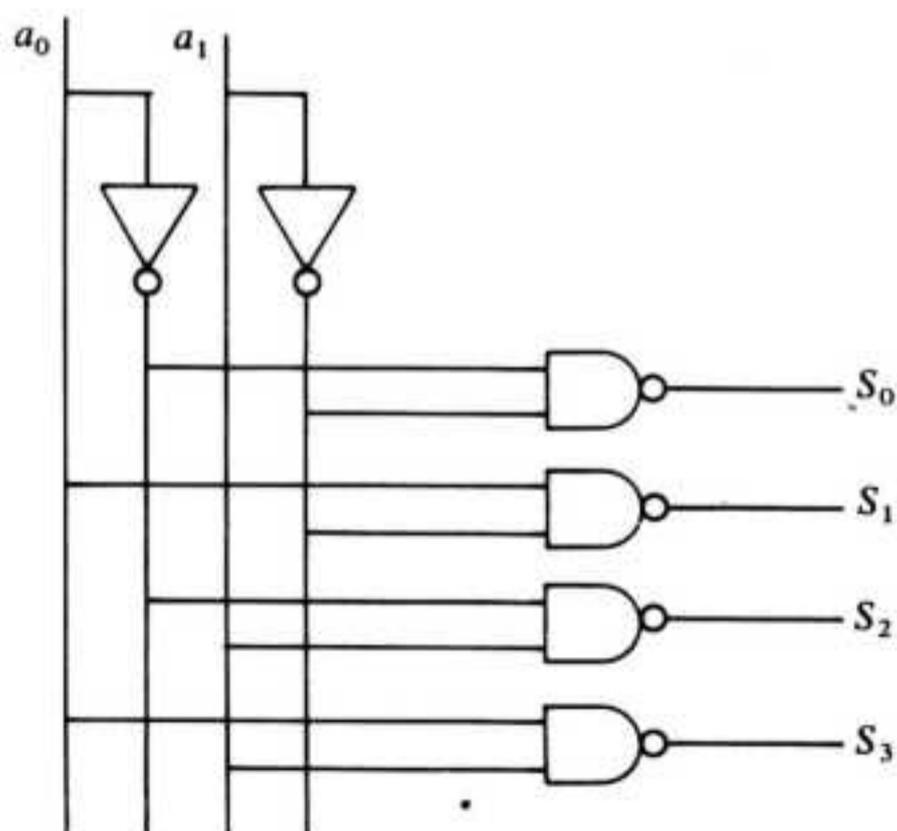


Figura 5.12. Diagrama lógico de un decodificador de dos entradas y cuatro salidas.

5.2.4.1. Decodificador 74LS42

Es un circuito construido en tecnología TTL. Tiene *cuatro líneas de entrada y diez de salida*. Aplicando una combinación BCD a su entrada, activa la correspondiente línea de salida.

En la Figura 5.13 aparece el diagrama de conexión y la denominación de cada terminal, y en la Figura 5.14 se muestra la tabla de la verdad. *El nivel activo a la salida es el cero* (trabajando en lógica positiva).

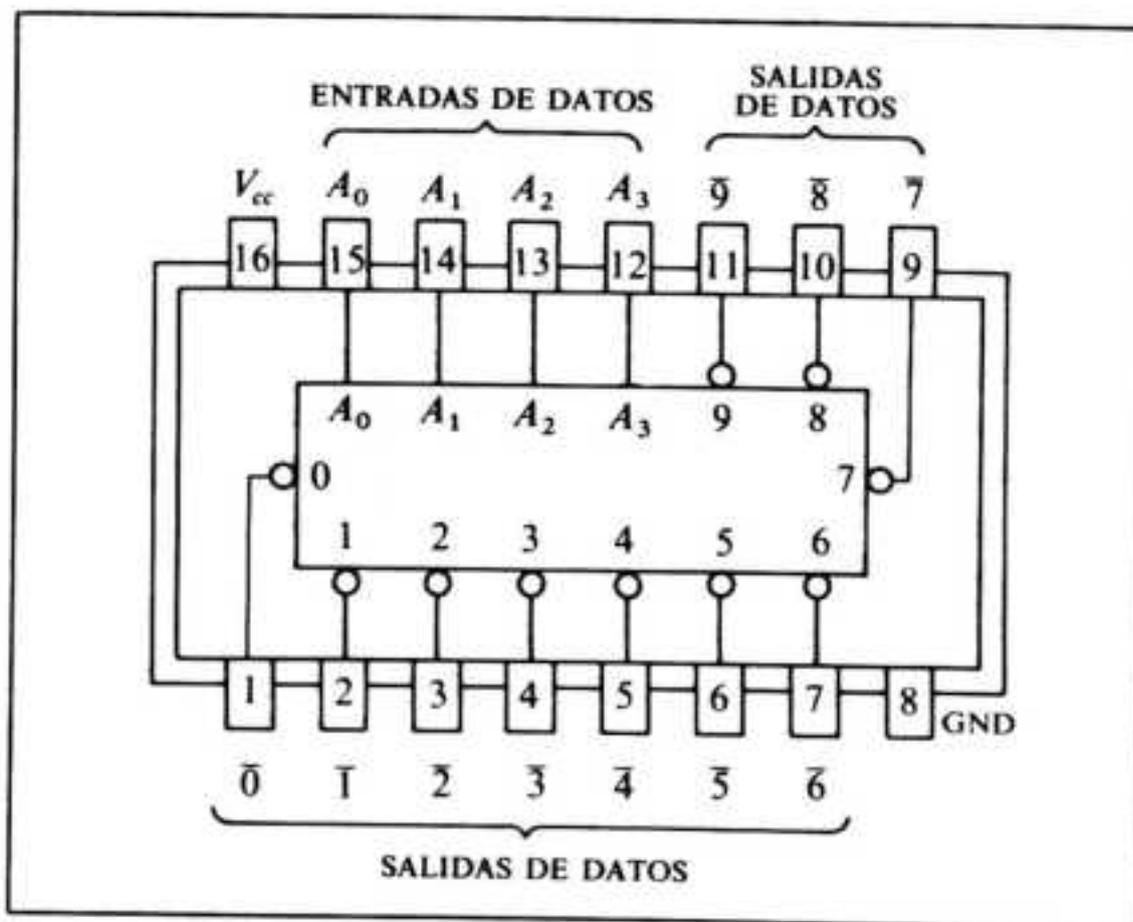


Figura 5.13. Diagrama de conexión del CI 74LS42.

5.3. MULTIPLEXADORES Y DEMULTIPLEXADORES

La función de **multiplexar** consiste en *enviar a voluntad por un solo canal de salida alguna de las informaciones presentes en varias líneas de entrada*. El dispositivo más elemental que realiza esta función es el conmutador.

Los circuitos combinacionales que realizan esta función se llaman **multiplexadores** y están formados por *N líneas de entrada de información, una salida y n entradas de control*. La relación entre las entradas de información y las de control es la siguiente: $N = 2^n$.

En la Figura 5.15 representamos simbólicamente un multiplexador con ocho entradas de información y tres de control. La información presente a la salida será

Entradas				Salidas									
A_0	A_1	A_2	A_3	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$	$\bar{8}$	$\bar{9}$
L	L	L	L	L	H	H	H	H	H	H	H	H	H
H	L	L	L	H	L	H	H	H	H	H	H	H	H
L	H	L	L	H	H	L	H	H	H	H	H	H	H
H	H	L	L	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	H	H	H	L	H	H	H	H	H
H	L	H	L	H	H	H	H	H	L	H	H	H	H
L	H	H	L	H	H	H	H	H	H	L	H	H	H
H	H	H	L	H	H	H	H	H	H	H	L	H	H
L	L	L	H	H	H	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L
L	H	L	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
L	L	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
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	H	H

L (bajo)
H (alto)

Figura 5.14. Tabla de la verdad del CI 74LS42.

la de la entrada seleccionada mediante los terminales de control a , b y c . La tabla de la verdad se muestra en la Figura 5.16.

La ecuación que se obtiene de la tabla de la verdad es:

$$S = \bar{a}\bar{b}\bar{c}d_0 + \bar{a}\bar{b}cd_1 + \bar{a}b\bar{c}d_2 + \bar{a}bcd_3 + a\bar{b}\bar{c}d_4 + a\bar{b}cd_5 + ab\bar{c}d_6 + abcd_7$$

El diagrama lógico correspondiente es el de la Figura 5.17.

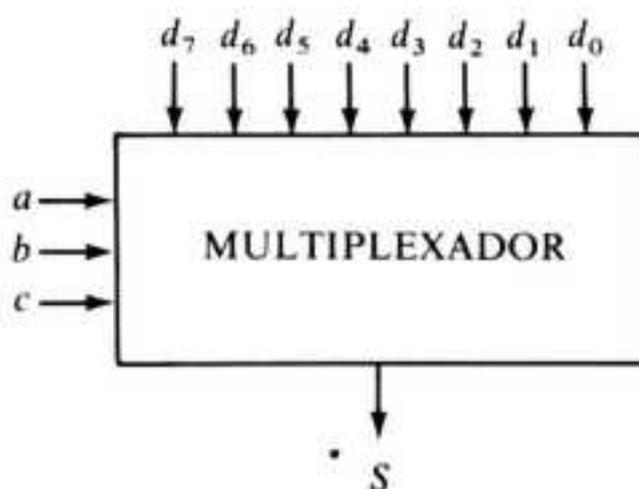


Figura 5.15. Símbolo lógico de un multiplexador de ocho líneas.

Entradas control			
<i>a</i>	<i>b</i>	<i>c</i>	<i>S</i>
0	0	0	d_0
0	0	1	d_1
0	1	0	d_2
0	1	1	d_3
1	0	0	d_4
1	0	1	d_5
1	1	0	d_6
1	1	1	d_7

Figura 5.16. Tabla de la verdad de un multiplexador de ocho líneas.

Los **demultiplexadores** son circuitos con *una sola entrada, N salidas y n entradas de control*. La información de la entrada se transmite a la línea de salida seleccionada mediante las entradas de control. La tabla de la Figura 5.18 corresponde a un demultiplexador de cuatro líneas de salida.

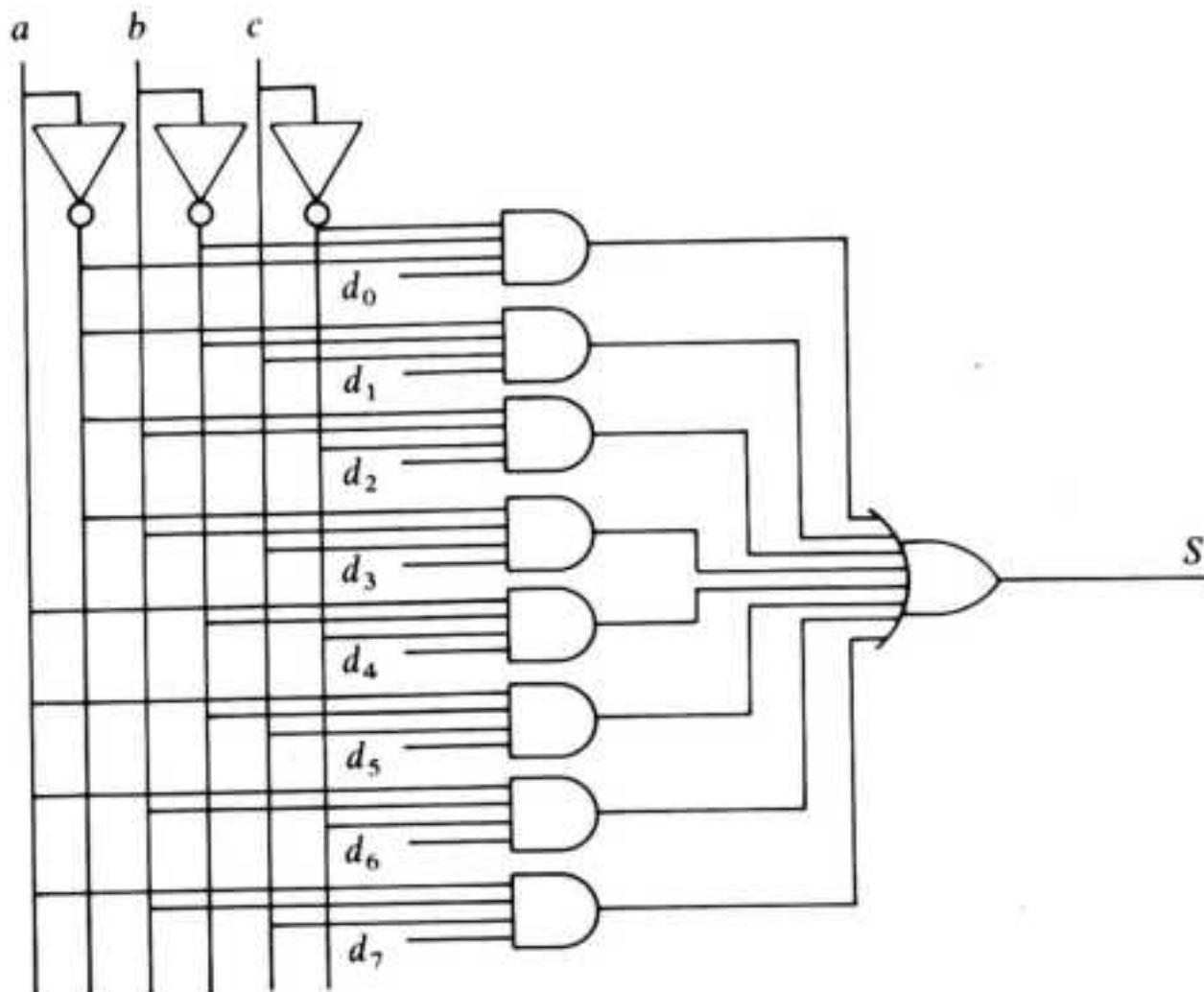


Figura 5.17. Diagrama lógico de un multiplexador de ocho líneas.

<i>a</i>	<i>b</i>	S_0	S_1	S_2	S_3
0	0	<i>d</i>	0	0	0
0	1	0	<i>d</i>	0	0
1	0	0	0	<i>d</i>	0
1	1	0	0	0	<i>d</i>

Las ecuaciones de salida son:

$$S_0 = \bar{a}\bar{b}d$$

$$S_1 = \bar{a}bd$$

$$S_2 = a\bar{b}d$$

$$S_3 = abd$$

Figura 5.18.

Técnicamente los circuitos integrados comerciales se pueden utilizar como demultiplexadores o como decodificadores. Así el circuito 74LS42 examinado en el apartado anterior funciona como demultiplexador, utilizando el terminal A_3 como entrada de datos y los A_0 , A_1 y A_2 como entradas de control. En este caso obtendremos un demultiplexador de ocho salidas (de 0 a 7).

5.3.1. MULTIPLEXADOR 74LS151

Es un circuito de *ocho líneas de entrada* (de D_0 a D_7), *tres entradas de selección*, A , B y C , y *una de inhibición*, S . Dispone también de *dos salidas complementarias*: Y y W .

En las Figuras 5.19 y 5.20 se muestran el diagrama de conexión y la tabla de la verdad, respectivamente.

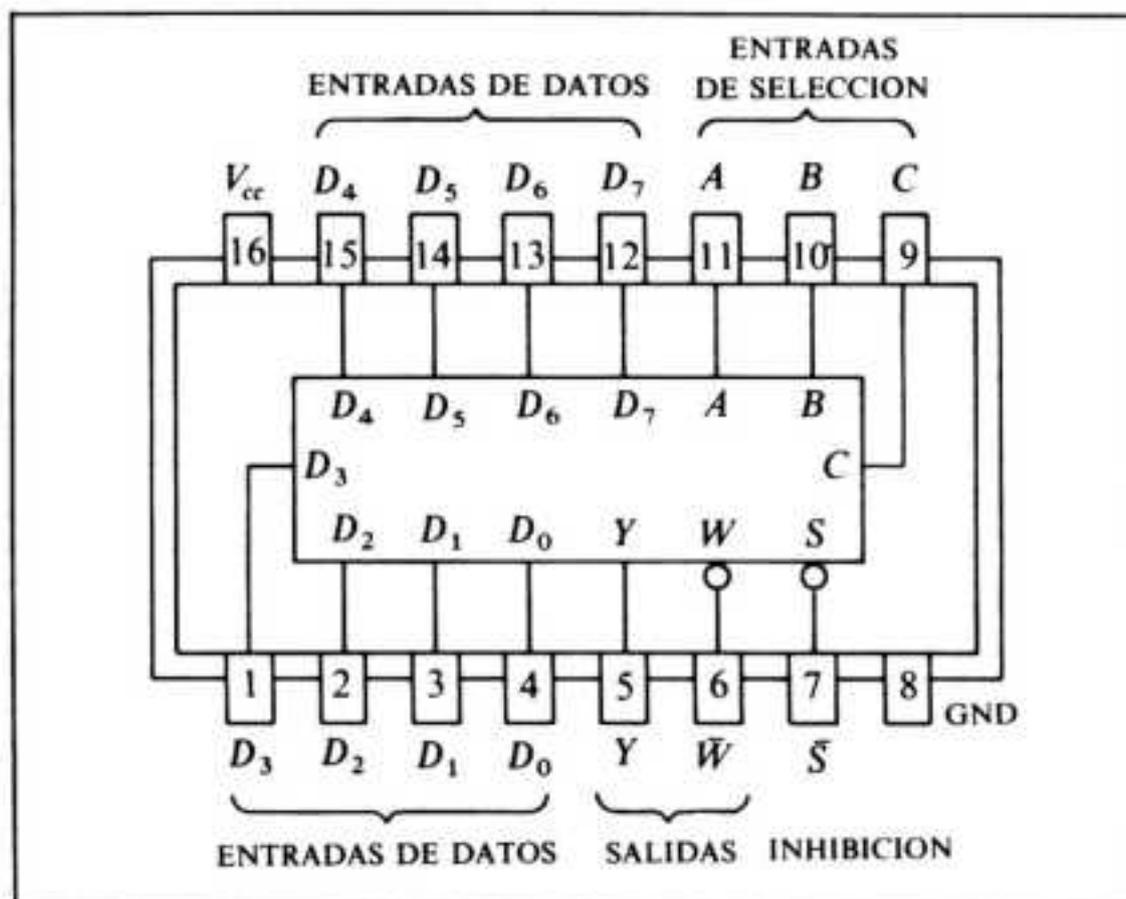


Figura 5.19. Diagrama de conexión del CI 74LS151.

Entradas				Salidas	
Selección			Inhibición	Y	\bar{W}
C	B	A	\bar{S}		
X	X	X	H	L	H
L	L	L	L	D_0	\bar{D}_0
L	L	H	L	D_1	\bar{D}_1
L	H	L	L	D_2	\bar{D}_2
L	H	H	L	D_3	\bar{D}_3
H	L	L	L	D_4	\bar{D}_4
H	L	H	L	D_5	\bar{D}_5
H	H	L	L	D_6	\bar{D}_6
H	H	H	L	D_7	\bar{D}_7

Figura 5.20. Tabla de la verdad del CI 74LS151.

Como se puede observar, la entrada de inhibición S a nivel alto fuerza las salidas Y y W a nivel bajo y alto respectivamente, sea cual sea el valor de las entradas de información y de selección.

5.4. COMPARADORES

Los **comparadores** son circuitos combinacionales que, al presentar en sus entradas dos palabras* de n bits, detectan si son o no iguales, y en este caso cuál de las dos es mayor o menor.

La puerta O exclusiva es una célula elemental comparadora (Figura 5.21).

a	b	S
0	0	0
0	1	1
1	0	1
1	1	0

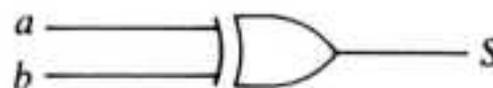


Figura 5.21. Tabla de la verdad y símbolo de la función O exclusiva.

* Palabra es un conjunto de bits que en un formato determinado (4, 6, 8, 16, 32 bits) forman un código para facilitar el intercambio de información en sistemas digitales.

El proceso de diseño de un comparador se realiza igual que en los casos anteriores. En la Figura 5.22 se indica la tabla de la verdad de un comparador completo de dos palabras de un bit cada una. El camino seguido para su diseño es válido para otros con un número de bits mayor por palabra.

a	b	S_0	S_1	S_2
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

S_0 es la salida $a > b$

S_1 es la salida $a < b$

S_2 es la salida $a = b$

Figura 5.22. Tabla de la verdad de un comparador de dos palabras de un bit cada una.

Las funciones S_0 , S_1 y S_2 valen:

$$S_0 = a \cdot \bar{b} \quad S_1 = \bar{a} \cdot b \quad S_2 = \overline{a \oplus b}$$

El diagrama correspondiente es el de la Figura 5.23.

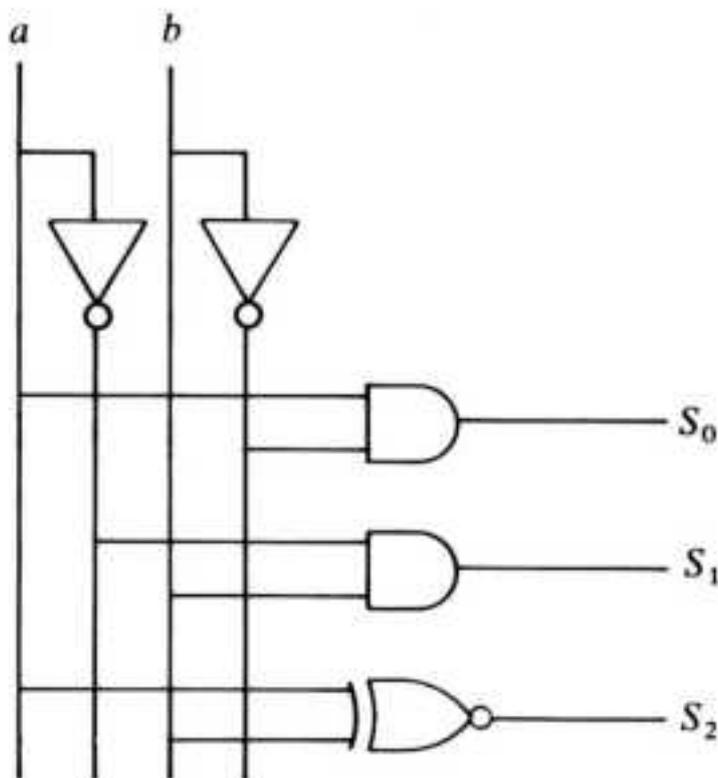


Figura 5.23. Diagrama lógico de un comparador de dos palabras de un solo bit.

5.4.1. COMPARADOR 7485

Es un comparador de palabras de *cuatro bits cada una*. Este dispositivo se puede conectar en paralelo con otros de idénticas características para comparar palabras de más bits. Aplicando las salidas $A > B$, $A = B$ y $A < B$ de un dispositivo a las entradas del mismo nombre de otro obtenemos un comparador de 8 bits.

En las Figuras 5.24 y 5.25 se representan el diagrama de conexión y la tabla de la verdad, respectivamente del 7485.

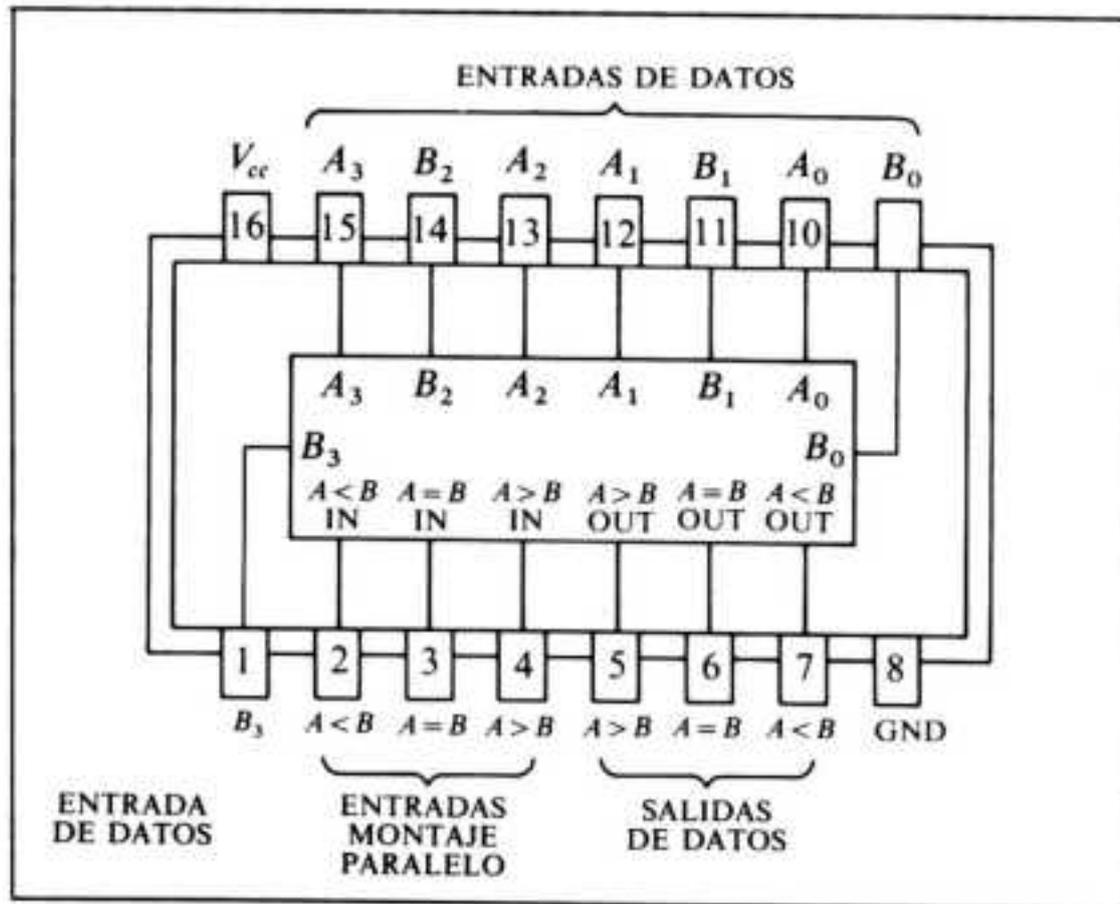


Figura 5.24. Diagrama de conexión del CI 7485.

Entradas				Entradas montaje paralelo			Salidas		
A_3, B_3	A_2, B_2	A_1, B_1	A_0, B_0	$A > B$	$A < B$	$A = B$	$A > B$	$A < B$	$A = B$
$A_3 > B_3$	X	X	X	X	X	X	H	L	L
$A_3 < B_3$	X	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	L	L	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	H	L	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	L	H	L	L	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	X	X	H	L	L	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	H	L	L	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	L	L	H	H	L

Figura 5.25. Tabla de la verdad del CI 7485.

5.5. OTROS CIRCUITOS COMBINACIONALES

Existen otros muchos tipos de circuitos combinacionales que están disponibles en bloques integrados o que se pueden construir mediante puertas lógicas. Destacaremos entre ellos los *detectores y generadores de paridad* y los *circuitos convertidores de código*. En el siguiente apartado reservado, como en el resto de los capítulos, a la resolución de casos prácticos analizaremos algunos de estos circuitos.

Por último, hemos de señalar que cuando surja la necesidad de incorporar un circuito combinacional de aplicación general a un sistema electrónico, se utilizará un bloque MSI de cualquiera de las tecnologías estudiadas en el Capítulo 4. La construcción mediante puertas lógicas quedará reservada para aquellos casos en los que no existan circuitos estándar en catálogo.

EJERCICIOS RESUELTOS

La explicación de los circuitos combinacionales más significativos (codificadores, decodificadores, multiplexadores, demultiplexadores y comparadores) descritos en los apartados anteriores se ha realizado utilizando un ejemplo concreto en cada caso. Por consiguiente, este apartado lo dedicaremos al diseño y construcción de los circuitos que únicamente se han relacionado en el apartado 5.5, con ello reforzaremos los conocimientos adquiridos sobre el proceso de diseño de circuitos digitales.

1. Dibujar el diagrama lógico de un generador de paridad par para una palabra de 8 bits.

Solución

El diseño de este tipo de circuitos (generadores y detectores) es distinto al descrito en apartados anteriores.

La puerta O exclusiva, en éste como en otros casos, es el elemento básico para la construcción de este tipo de circuitos. Esta función es un generador elemental de paridad par para una palabra de 2 bits.

Para palabras de más de dos bits bastará con construir una puerta O exclusiva con un mayor número de entradas mediante la ramificación de puertas de dos entradas.

Para un generador de 8 bits el circuito será tal como se representa en la Figura 5.26.

Las salidas de dos puertas sucesivas se aplican a otra de idénticas características. De esta manera las siguientes etapas están formadas por un número menor de puertas que las anteriores. Para construir un generador de paridad impar se coloca una puerta inversora a la salida.

La función del circuito será:

$$BP = [(a \oplus b) \oplus (c \oplus d)] \oplus [(e \oplus f) \oplus (g \oplus h)]$$

El(la) lector(a) podrá ensayar, particularizando, la ecuación para combinaciones numéricas concretas y comprobará el resultado, observando que en todos los casos se obtiene un

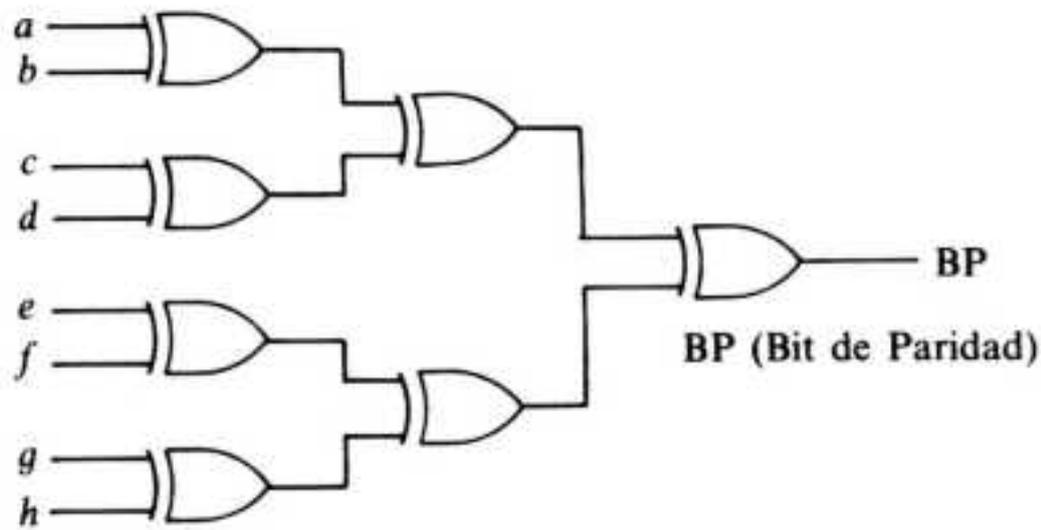


Figura 5.26. Diagrama de un generador de paridad par para una palabra de 8 bits.

valor de BP que hace par el número de unos de la palabra, incluyendo en ella el propio bit de paridad.

2. Construir un detector de paridad para palabras de 7 bits, incluido el bit de paridad.

Solución

La estructura del circuito es idéntica al caso anterior. El circuito en este caso será tal como se representa en la Figura 5.27.

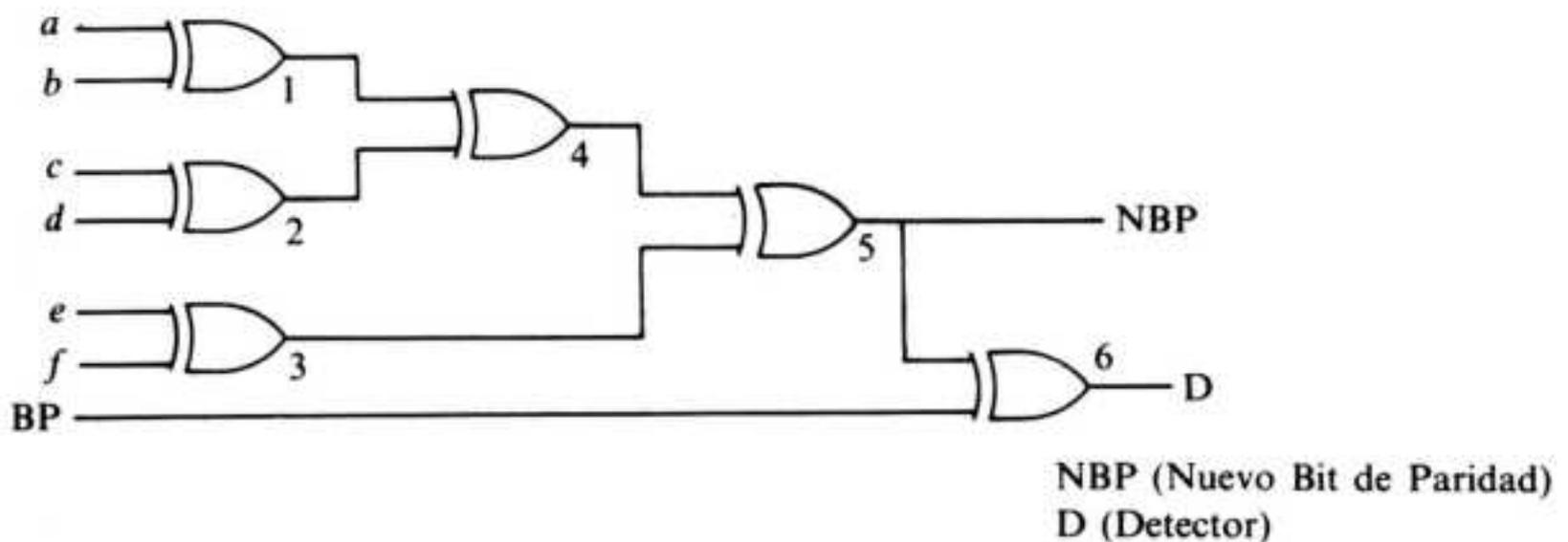


Figura 5.27. Diagrama de un detector de paridad para palabras de 7 bits (incluido el bit de paridad, BP).

Como se puede comprobar, se vuelve a generar otra vez el bit de paridad (NBP) y se compara éste con el generado anteriormente mediante la puerta 6. Si la salida D es 0, es porque ambos bits (BP y NBP) son iguales y, por tanto, la transmisión fue correcta. Si la salida es 1, hubo error en la transmisión.

El circuito es también un generador de bit de paridad para una palabra de 7 bits.

3. Diseñar un circuito combinacional para convertir el código Aiken en BCD natural.

Solución

La tabla de la verdad será la indicada en la Figura 5.28.

	<i>Aiken</i>				<i>BCD</i>			
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	1	0	1	1	0	1	0	1
6	1	1	0	0	0	1	1	0
7	1	1	0	1	0	1	1	1
8	1	1	1	0	1	0	0	0
9	1	1	1	1	1	0	0	1

Figura 5.28.

Ecuaciones:

$$A = abc\bar{d} + abcd = abc(\bar{d} + d) = \boxed{abc}$$

$$B = \bar{a}b\bar{c}\bar{d} + \bar{a}bcd + ab\bar{c}\bar{d} + ab\bar{c}d$$

Reduciendo *B* por el método de Karnaugh:

$$B = \boxed{b\bar{c}\bar{d} + ab\bar{c} + a\bar{b}cd}$$

$$C = \bar{a}b\bar{c}\bar{d} + \bar{a}bcd + ab\bar{c}\bar{d} + ab\bar{c}d$$

Aplicando el método de Karnaugh a *C* obtendremos:

$$C = \boxed{\bar{a}\bar{b}c + ab\bar{c}}$$

Observando la tabla de la verdad de la Figura 5.28 se comprueba que:

$$D = \boxed{d}$$

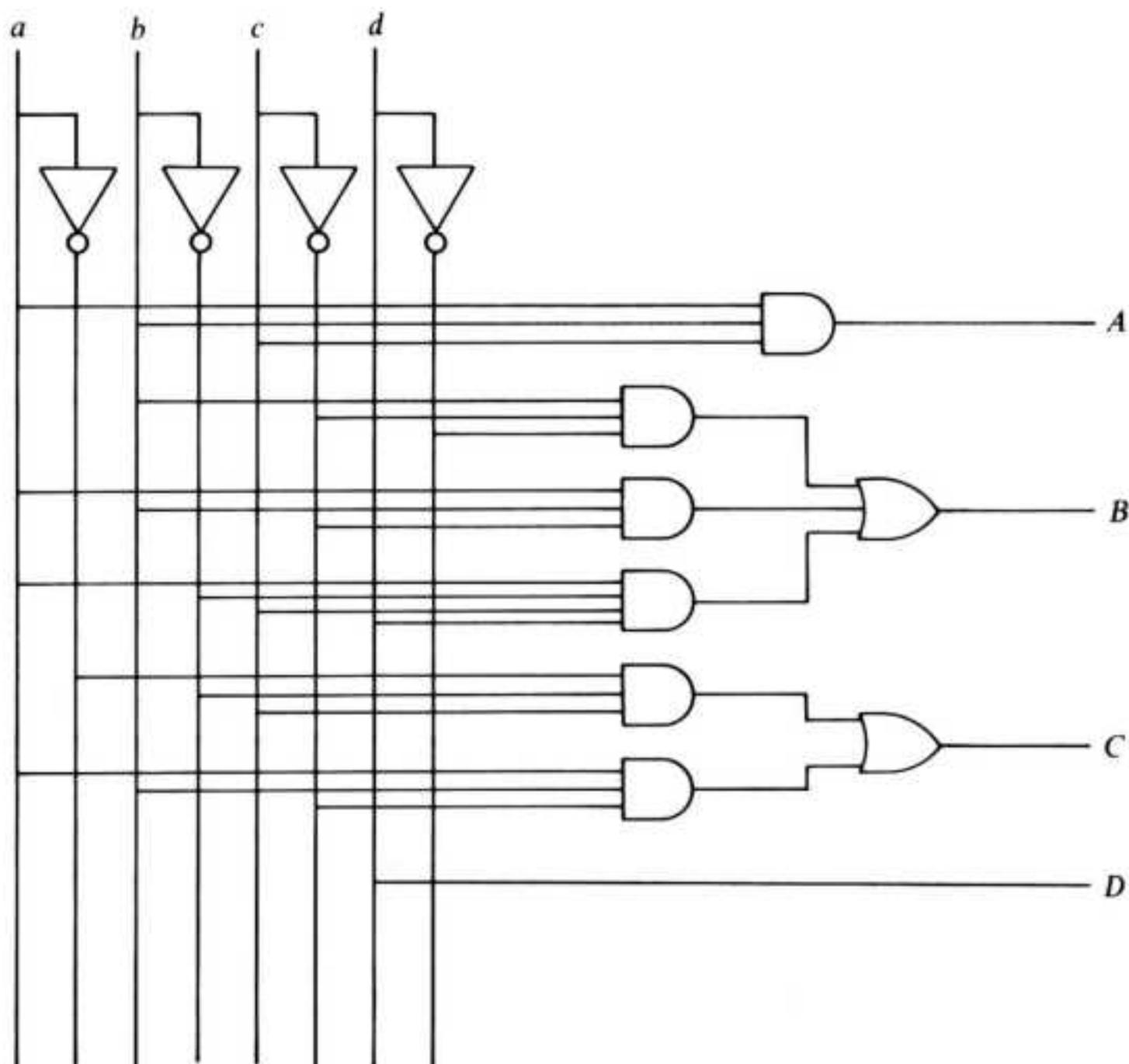


Figura 5.29. Convertidor del código Aiken en BCD natural.

EJERCICIOS PROPUESTOS

- Pasar de decimal a binario los siguientes números:
 - 36,75
 - 12,525
 - 102,125
 - 50,225
- Calcular los números decimales equivalentes a los siguientes binarios:
 - 110011
 - 10011110
 - 001000,0101
 - 11110000,1110
- Codificar en BCD natural, exceso tres y Aiken las siguientes cantidades:
 - 4725
 - 839
 - 10.637
 - 235

4. Construir un código de paridad par a partir del BCD Aiken.
 5. Formar el código Hamming utilizando el BCD exceso tres.
 6. Expresar en código ASCII:
 - a) *B*
 - b) *7*
 - c) *h*
 - d) *LF*
 - e) *DEL*
 - f) ***
 - g) :
 7. Diseñar un generador de paridad impar para una palabra de 10 bits.
 8. Construir un detector de paridad impar para palabras de 5 bits sin incluir el bit de paridad.
 9. Dibujar el circuito combinacional necesario para convertir el código BCD natural en BCD exceso tres.
 10. Obtener las funciones a partir de la tabla de la Figura 5.14 y representar el circuito lógico correspondiente.
-

Puerta

