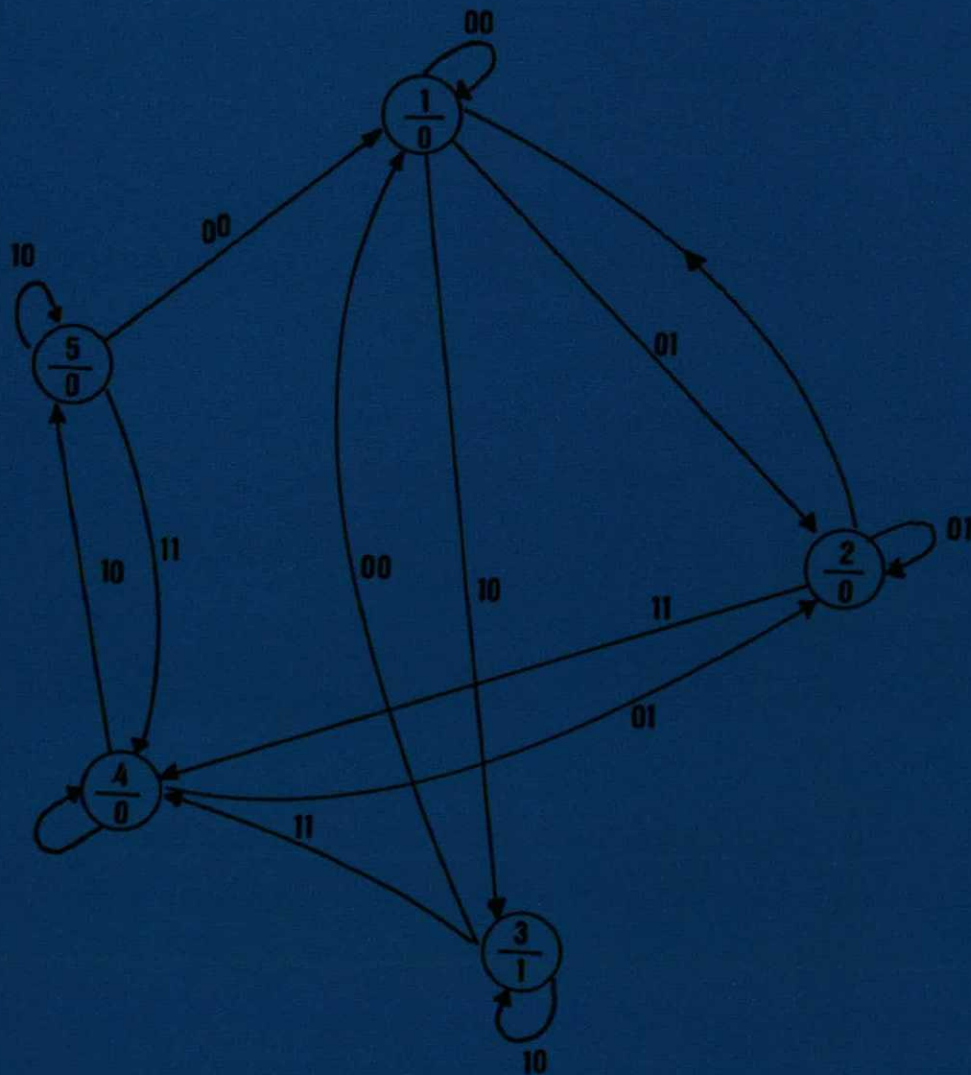


teoría de la conmutación

ALFONSO
GARCIA
RODRIGUEZ



APUNTES DE TEORIA
DE LA
CONMUTACION

APUNTES DE TEORIA DE LA CONMUTACION

Redactados por
ALFONSO GARCIA RODRIGUEZ
Profesor del Instituto de Informática

INSTITUTO DE INFORMATICA

MADRID, 1974

© Servicio de Publicaciones del Ministerio de Educación y Ciencia, 1974

Edita: Servicio de Publicaciones del Ministerio de Educación y Ciencia

Portada: Luis F. del Valle

Imprime: Boletín Oficial del Estado

ISBN: 84-369-0369-2

Depósito legal: M 38103/1974

Printed in Spain.

PROLOGO

Este texto sobre Teoría de la Conmutación se ha dividido, por razones didácticas, en dos partes: la primera dedicada a Circuitos Combinatorios, y la segunda, a Circuitos Secuenciales (síncronos y asíncronos). Se ha pretendido orientarlo, preferentemente, hacia los problemas de síntesis por considerar que este campo presenta un gran interés.

El lector interesado puede complementar este estudio con la Teoría de las Particiones, que proporciona un método de asignación de las variables de estado y facilita la descomposición de circuitos secuenciales complejos en máquinas más sencillas.

Madrid, septiembre de 1973

Primera parte

CIRCUITOS COMBINATORIOS

SUMARIO

Página

PRIMERA PARTE

CIRCUITOS COMBINATORIOS

CAPITULO 1. INTRODUCCION:

1.1	Sistemas de conmutación	15
1.2	Tipos de lógica	15

CAPITULO 2. ALGEBRA DE CONMUTACION:

2.1	Variable de conmutación y valor verdadero	17
2.2	Operaciones en álgebra de conmutación	17
2.3	Postulados y teoremas	18
2.4	Funciones de conmutación y tablas de verdad	19
2.5	Puertas lógicas	21
2.6	Conversión lógica	25
2.6.1	Lógica AND-NOT	25
2.6.2	Lógica OR-NOT	26
2.6.3	Lógica NAND	26
2.6.4	Lógica NOR	27
2.7	Principio de dualidad	27

CAPITULO 3. FORMULACION DE FUNCIONES DE CONMUTACION:

3.1	Productos <i>standards</i> . Forma disyuntiva	29
3.2	Sumas <i>standards</i> . Forma conjuntiva	30
3.3	Conversión entre formas básicas.....	31

CAPITULO 4. SIMPLIFICACION DE FUNCIONES DE CONMUTACION:

4.1	Representación gráfica de variables y funciones: Método de Veitch-Karnaugh.	35
4.2	Simplificación de funciones	37

4.3	Utilización del mapa de Karnaugh para simplificar funciones expresadas en forma de producto de sumas	38
4.4	Mapa de Karnaugh para cinco variables	39
4.5	Método de tabulación de Quine-Mc Cluskey	40
CAPITULO 5. RELE LOGICO:		
5.1	Representación. Puertas AND y OR	45
5.2	Optimización de circuitos	47
5.3	Redes en puente	50
CAPITULO 6. DIODO LOGICO:		
6.1	Representación y características de los diodos de conmutación	53
6.2	Puertas AND y OR	55
6.3	Efecto de una impedancia de carga finita	60
6.4	Optimización de circuitos	63
6.5	Problema de la coincidencia de impulsos	64
CAPITULO 7. TRANSISTOR LOGICO:		
7.1	Introducción	67
7.2	Transistor lógico	74
7.3	Transistor inversor	76
7.4	Cálculo de un inversor	78
7.5	Puertas diodo-transistor (DTL)	81
7.6	Puertas AND y OR (seguidor de emisor)	85
7.7	Transistores directamente acoplados (DCTL)	88
7.8	Tiempos de conmutación del transistor	91
7.9	Diseño con puertas NAND y NOR transistorizadas	93
7.10	Conclusiones	94
APÉNDICE 1. Teorema de Thevenin		95
APÉNDICE 2. Funciones AND y OR cableadas		96
CAPITULO 8. REDES MULTITERMINALES:		
8.1	Introducción	99
8.2	Red en estrella	100
8.3	Red en triángulo	103
8.4	Redes en árbol	106
8.5	Codificadores	112
8.6	Decodificadores	113
8.7	Conversores de código	114
CAPITULO 9. CIRCUITOS SUMADORES Y SUSTRACTORES:		
9.1	Suma y sustracción binarias	117
9.2	Semisumador	119
9.3	Sumador completo	120
9.4	Semisustractor	124
9.5	Sustractor completo	124
9.6	Sumador decimal	127

CAPITULO 10. REDES EN CASCADA:

10.1	Introducción	129
10.2	Células básicas	130
10.3	Fases del diseño de un circuito en cascada	130
10.4	Circuito de comprobación de paridad	132
10.5	Circuito para la determinación del número de «unos»	136
10.6	Sumador binario paralelo	139
10.7	Sumador decimal paralelo	139

BIBLIOGRAFIA		143
--------------------	--	-----

CAPITULO 1
INTRODUCCION

1.1 SISTEMAS DE CONMUTACION

Los sistemas de conmutación son instrumentos indispensables en dos amplios campos: Telecomunicación e Informática.

El mayor sistema de conmutación, actualmente existente, es la red telefónica automática. En el campo de la Informática, es de destacar la importancia de los ordenadores digitales de alta velocidad.

Los sistemas de conmutación se caracterizan por responder a un conjunto de *reglas predeterminadas*, por lo cual se puede establecer una teoría de diseño de los mismos.

El diagrama de bloques de un sistema de conmutación se muestra en la figura 1.1.

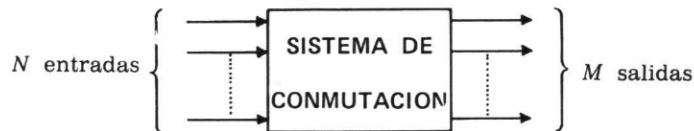


FIGURA 1.1

1.2 TIPOS DE LOGICA

Todos los elementos utilizados se caracterizan por presentar dos estados; de aquí se infiere el uso del Sistema Binario.

Los dos estados de un elemento binario se pueden representar de diferentes formas:

- a) Mediante ausencia o presencia de corriente, como ocurre en el caso de utilizar contactos de relés o conmutadores.



FIG. 1.2

El estado 0 puede corresponder al contacto abierto, y el estado 1, al contacto cerrado (continuidad eléctrica), o viceversa.

b) Mediante niveles lógicos (V_+ y V_-).



FIG. 1.3

c) Mediante impulsos de anchura finita.

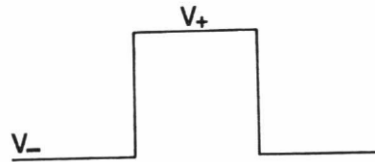


FIG. 1.4

En este caso hay que tener presente el problema de la coincidencia de impulsos (véase capítulo 6).

Las tensiones V_- y V_+ voltios se pueden hacer corresponder con los estados lógicos «0» y «1»:

- Si V_- indica la condición lógica 0, y V_+ la condición lógica 1, se trata de *lógica positiva*, ya que «1» se elige para representar la tensión más alta.
- Si V_- representa la condición lógica 1, y V_+ la condición lógica 0, se trata de *lógica negativa*, por elegirse «1» para representar la tensión más baja.

Inicialmente, sólo se utilizaban contactos de relés, ya que las válvulas presentaban algunos inconvenientes: volumen grande, vida limitada, caldeo de los filamentos, etc.

Los contactos de relés presentan ventajas e inconvenientes. Como ventajas, se pueden citar, entre otras:

- Un relé puede tener muchos contactos.
- Pueden diseñarse de modo que sean rápidos o lentos a la actuación y/o desprendimiento.

Pero los relés presentan inconvenientes, tales como:

- Número limitado de horas de vida.
- Susceptibles a la suciedad, polvo, humedad, grasa, etc., que pueden ocasionar funcionamientos erráticos.

Actualmente, los circuitos lógicos se presentan en pastillas de circuitos integrados, de construcción delicada y compleja, sin posibilidad de manipulación en ellos.

CAPITULO 2

ALGEBRA DE CONMUTACION

2.1 VARIABLE DE CONMUTACION Y VALOR VERDADERO

Algebra de conmutación es la aplicación del álgebra de Boole a los sistemas de conmutación.

Una variable de conmutación es una variable que puede tomar uno u otro de dos valores distintos. A este valor que toma la variable se le denomina valor verdadero.

Una variable de conmutación se representa por una letra mayúscula o minúscula.

Generalmente, los dos valores que toman las variables se designan por 0 y 1.

2.2 OPERACIONES EN ALGEBRA DE CONMUTACION

Existen tres operaciones fundamentales: *AND*, *OR* y *NOT*.

AND: La combinación *AND* de dos variables de conmutación tiene el valor verdadero 1 si, y sólo si, ambas variables tienen el valor verdadero 1.

Sean x e y las dos variables, su combinación *AND* se representa de las siguientes formas:

$$xy, x \cdot y, x \wedge y, x \cap y, x, y$$

En este texto se utiliza la representación xy .

OR: La combinación *OR* de dos variables de conmutación tiene el valor verdadero 1 si, y sólo si, una o ambas variables tienen el valor verdadero 1.

Sean x e y las dos variables, su combinación *OR* se representa de las siguientes formas:

$$x + y, x \vee y, x \cup y$$

En este texto se utiliza la representación $x + y$.

NOT: También recibe los nombres de negación y complemento. El valor verdadero de la negación de una variable de conmutación es 1 si, y sólo si, el valor verdadero de la variable es 0.

Se representa de las siguientes formas:

$$x' , \bar{x} , \sim x$$

En este texto se utiliza x' .

2.3 POSTULADOS Y TEOREMAS

Postulados:

$$\begin{aligned} x = 0 \quad \text{o} \quad x = 1 \\ 0 \cdot 0 = 0 \\ 1 \cdot 1 = 1 \\ 1 \cdot 0 = 0 \cdot 1 = 0 \\ 0 + 0 = 0 \\ 1 + 1 = 1 \\ 1 + 0 = 0 + 1 = 1 \\ 1' = 0 \\ 0' = 1 \end{aligned}$$

A continuación se da un listado de los teoremas más importantes, sin demostración, ya que dicha tarea no es objeto de este curso.

1. Conmutativo:

$$\begin{aligned} x + y = y + x \\ x y = y x \end{aligned}$$

2. Asociativo:

$$\begin{aligned} x + y + z = (x + y) + z = x + (y + z) = (x + z) + y \\ x y z = (x y) z = x (y z) = (x z) y \end{aligned}$$

3. Idempotencia:

$$\begin{aligned} x + x = x \\ x x = x \end{aligned}$$

4. Distributivo:

$$\begin{aligned} x(y + z) = x y + x z \\ (x + y) \cdot (x + z) = x + y z \end{aligned}$$

5. Redundancia:

$$\begin{aligned} x + x y = x \\ x(x + y) = x \end{aligned}$$

6. Negación:

$$(x')' = x$$

7. Complemento:

$$x + x' = 1$$
$$x \cdot x' = 0$$

8. Operaciones con 0 y 1:

$$x + 0 = x$$
$$x + 1 = 1$$
$$0 \cdot x = 0$$
$$1 \cdot x = x$$

9. De Morgan's:

$$(x + y)' = x' y'$$
$$(x y)' = x' + y'$$

10. Otros teoremas:

$$x + x' y = x + y$$
$$x y + x' z + y z = x y + x' z = (x + z) (x' + y)$$
$$(x + y) (x + y') = x$$
$$(x + y) (x' + z) (y + z) = (x + y) (x' + z)$$

2.4 FUNCIONES DE CONMUTACION Y TABLAS DE VERDAD

Una función de conmutación es una combinación de variables y operaciones de conmutación. Ejemplo: $f = x y$.

Para determinar el valor verdadero de la función f se construye la siguiente tabla, llamada «tabla de verdad» (*truth table*):

x	y	$f = x y$
0	0	0
0	1	0
1	0	0
1	1	1

Las columnas x e y indican el conjunto de los posibles valores verdaderos de las variables x e y . Los valores de la columna $f = x y$ se obtienen mediante la aplicación de la definición de la operación *AND*.

Las funciones de conmutación pueden materializarse con conmutadores o contactos de relés, diodos, transistores, núcleos de ferrita, etc. Ya se ha dicho que se va a representar, en este texto, el contacto abierto por «0», y el cerrado, por «1».

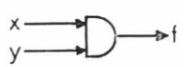
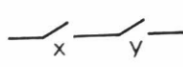
Los diferentes tipos de contactos se representan en la figura 2.1.

Tipo	Representación	Otras denominaciones
Abierto		— de trabajo — de hacer
Cerrado		— de reposo — de romper
De transferencia		— de reposo-trabajo — de conmutación
De transferencia con continuidad		— de trabajo-reposo

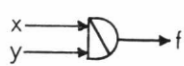
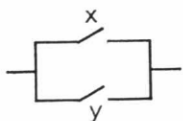
FIG. 2.1

2.5 PUERTAS LOGICAS

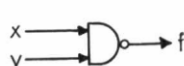
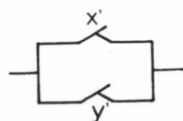
1) AND

Tabla de verdad			Función	REPRESENTACION	
				Lógica	Con contactos
x	y	f			
0	0	0	$f = xy$		
0	1	0			
1	0	0			
1	1	1			

2) OR (OR INCLUSIVA)

x	y	f			
0	0	0	$f = x + y$		
0	1	1			
1	0	1			
1	1	1			

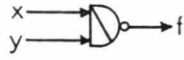
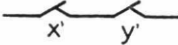
3) NAND

x	y	f			
0	0	1	$f = (xy)' = x' + y'$		
0	1	1			
1	0	1			
1	1	0			

$NAND = (AND)'$

La función *NAND* también se representa como $f = x/y$.


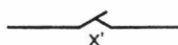
4) **NOR**

Tabla de verdad			Función	REPRESENTACION	
				Lógica	Con contactos
<u>x</u>	<u>y</u>	<u>f</u>			
0	0	1			
0	1	0	$f = (x + y)' = x' y'$		
1	0	0			
1	1	0			


$NOR = (OR)'$

La función **NOR** también se representa como $f = x \downarrow y$

5) **NOT** (INVERSOR, OPERADOR COMPLEMENTO)

<u>x</u>	<u>f</u>			
0	1	$f = x'$		
1	0			

6) **OR EXCLUSIVA**

<u>x</u>	<u>y</u>	<u>f</u>		
0	0	0	$f = x'y + xy' = x \oplus y$	
0	1	1		
1	0	1		
1	1	0		

Se observa que existe un impulso a la salida cuando se aplica un impulso a una u otras de las dos entradas, pero no cuando se aplican simultáneamente a ambas.

También se le denomina, a veces, operador no-equivalente, disyunción o dilema.

OPERADOR EQUIVALENTE

Tabla de verdad			Función	REPRESENTACION	
				Lógica	Con contactos
x	y	f			
0	0	1	$f = x'y' + xy$		
0	1	0			
1	0	0			
1	1	1			

Aplicando el teorema de De Morgan's a la función OR exclusiva, resulta:

$$(x'y + xy')' = (x + y') \cdot (x' + y) = xx' + xy + x'y' + yy' = xy + x'y'$$

ya que

$$\begin{cases} xx' = 0 \\ yy' = 0 \end{cases}$$

(OR EXCLUSIVA)' = OPERADOR EQUIVALENTE

7) AND CON INHIBICIÓN

x	y	f		
0	0	0	$f = x'y$	
0	1	1		
1	0	0		
1	1	0		

En general, es un circuito que contiene $n + 1$ entradas. Suministra una respuesta cuando se presentan simultáneamente n impulsos, no dando lugar a ninguna salida cuando se aplican a sus entradas $n + 1$ impulsos



cuando

$$\begin{aligned} x = 1, y = 1, \dots, t = 0 &\Rightarrow f = 1 \\ x = 1, y = 1, \dots, t = 1 &\Rightarrow f = 0 \end{aligned}$$

SIMBOLOS LOGICOS DE PROCEDENCIAS DIVERSAS

PUERTA	U.S.A.	FRANCIA	INGLATERRA
AND			
OR			
NAND			
NOR			
NOT			

2.6 CONVERSION LOGICA

A veces, es necesario cambiar de un tipo a otro de puertas lógicas por razones de disponibilidad, calidad, coste, uniformidad, etc.

El fundamento teórico de estos cambios se basa en el teorema de De Morgan's:

$$\left. \begin{aligned} (x + y + z)' &= x' y' z' \\ (x y z)' &= x' + y' + z' \end{aligned} \right\}$$

La conversión puede ser aplicada a todos los tipos de puertas (*DTL, TTL, etc.*) y lógicas (positiva o negativa).

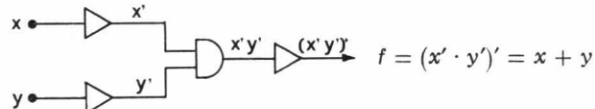
A continuación se desarrollan los cuatro casos posibles:

Puertas lógicas disponibles	Funciones que se desean obtener
1. <i>AND</i> y <i>NOT</i>	<i>OR</i> , <i>NAND</i> , <i>NOR</i>
2. <i>OR</i> y <i>NOT</i>	<i>AND</i> , <i>NAND</i> , <i>NOR</i>
3. <i>NAND</i>	<i>AND</i> , <i>OR</i> , <i>NOR</i>
4. <i>NOR</i>	<i>AND</i> , <i>OR</i> , <i>NAND</i>

2.6.1 Lógica AND-NOT

a) Representación de la función *OR*:

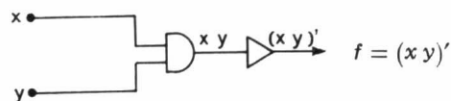
$$f = x + y = (x + y)'' = (x' y')'$$



OR

b) Representación de la función *NAND*:

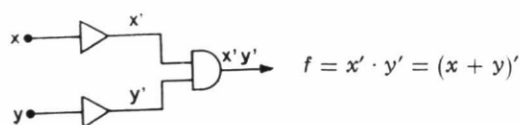
$$f = (x \cdot y)' = (AND)'$$



NAND

c) Representación de la función *NOR*:

$$f = (x + y)' = x' \cdot y'$$

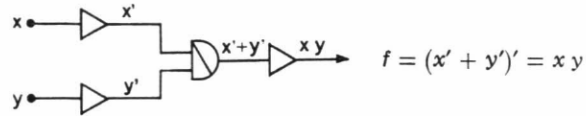


NOR

2.6.2 Lógica OR-NOT

a) Representar la función **AND** usando puertas **OR-NOT**:

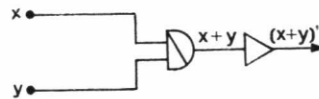
$$f = xy = (xy)'' = (x' + y)'$$



AND

b) Representar la función **NOR** usando puertas **OR-NOT**:

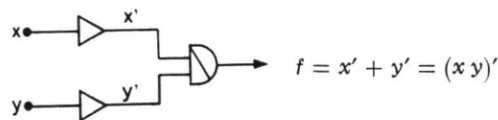
$$f = (x + y)'$$



NOR

c) Representar la función **NAND** usando puertas **OR-NOT**:

$$f = (x \cdot y)' = x' + y'$$

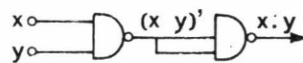


NAND

2.6.3 Lógica NAND

a) Representación de la función **AND**:

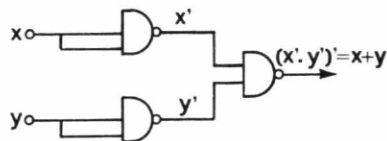
$$f = xy = (xy)'' = (NAND)'$$



AND

b) Representación de la función **OR**:

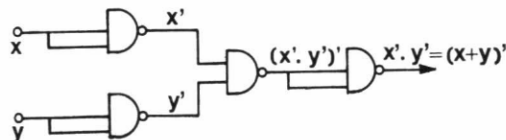
$$f = x + y = (x + y)'' = (x' y')'$$



OR

c) Representación de la función **NOR**:

$$f = (x + y)' = x' y' = (x' y')''$$

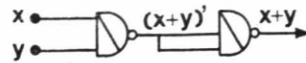


NOR

2.6.4 Lógica NOR

a) Representación de la función **OR**:

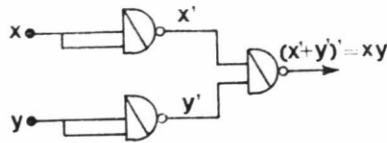
$$f = x + y = [(x + y)']'$$



OR

b) Representación de la función **AND**:

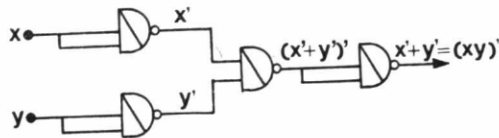
$$f' = x y = (x y)'' = (x' + y)'$$



AND

c) Representación de la función **NAND**:

$$f = (x y)' = (x' + y)''$$



NAND

2.7 PRINCIPIO DE DUALIDAD

Cada teorema o identidad que se deduzca de los postulados del álgebra de Boole permanece válida si las operaciones (+) y (·) y la identidad de los elementos 0 y 1 son intercambiados. Ejemplos:

	$x + 1 = 1$		$x + 0 = x$
dual	$x \cdot 0 = 0$	dual	$x \cdot 1 = x$
	<hr style="width: 50%; margin: 0 auto;"/>		<hr style="width: 50%; margin: 0 auto;"/>
	$x \cdot x = x$		$x \cdot 0 = 0$
dual	$x + x = x$	dual	$x + 1 = 1$
	<hr style="width: 50%; margin: 0 auto;"/>		<hr style="width: 50%; margin: 0 auto;"/>



CAPITULO 3

FORMULACION DE FUNCIONES DE CONMUTACION

3.1 PRODUCTOS *STANDARDS*. FORMA DISYUNTIVA

Es un producto de todas las variables de conmutación, negadas o no negadas.

Recibe el nombre de *canonical minterm* o *minterm*. Supongamos que se tienen n variables; como cada variable puede adoptar dos estados, se pueden formar 2^n productos *standards*.

Para interpretar físicamente esta situación, podemos decir que cada uno de los 2^n productos corresponde a una de las 2^n posibles combinaciones de entrada. Si el valor verdadero de cualquier producto *standard* es 1, significa que su correspondiente combinación de entrada existe.

Dado un producto *standard*, ¿cómo se puede proceder para identificar sus correspondientes combinaciones de entrada, o viceversa? Depende del significado físico que se quiera asignar al producto *standard*.

Se puede establecer la siguiente regla:

- Si una variable aparece sin negar en el producto, el valor verdadero de esta variable en la correspondiente combinación de entrada es 1.
- Si la variable aparece en forma negada en el producto, el valor verdadero de la variable en la correspondiente combinación de entrada es 0.

Se llama forma *disyuntiva* a un conjunto de productos *standards* unidos por operaciones *OR*.

Los productos *standards* pueden representarse con dos notaciones: la notación binaria y la notación m .

EJEMPLO:

Producto <i>standard</i>	Notación binaria	Notación m
$x' y' z'$	0 0 0	m_0
$x' y' z$	0 0 1	m_1
$x' y z'$	0 1 0	m_2
$x' y z$	0 1 1	m_3
$x y' z'$	1 0 0	m_4
$x y' z$	1 0 1	m_5
$x y z'$	1 1 0	m_6
$x y z$	1 1 1	m_7

Utilizando esta notación, una función f puede representarse de la siguiente forma:

$$f = x' y' z + x y' z' + x y z = m_1 + m_4 + m_7 = \Sigma m_{1, 4, 7} = \Sigma 1, 4, 7$$

3.2 SUMAS *STANDARDS*. FORMA CONJUNTIVA

Es una suma de todas las variables de conmutación, negadas o no negadas. Recibe el nombre de *canonical maxterm* o *maxterm*.

Si se tiene n variables, se pueden formar 2^n sumas *standards*.

Para interpretar físicamente esta situación, podemos decir que cada una de las 2^n sumas corresponde a una de las 2^n posibles combinaciones de entrada.

Cuando el valor verdadero de una suma *standard* es cero, significa que su correspondiente combinación de entrada existe.

Dada una suma *standard*, ¿cómo se puede proceder para identificar su correspondiente combinación de entrada? Se puede establecer la siguiente regla:

- Si una variable aparece en la suma en forma no negada, el valor verdadero de esta variable en la correspondiente combinación de entrada será 0.
- Si una variable aparece en forma negada, el valor verdadero de la variable en la correspondiente combinación de entrada será 1.

EJEMPLO:

$$a' + b' + c' = 0 \Rightarrow a = 1, \quad b = 1, \quad c = 1$$

Se llama *forma conjuntiva* a un conjunto de sumas *standards* unidas por operaciones *AND*.

Las sumas *standards* pueden representarse con dos notaciones: la notación binaria y la notación M .

EJEMPLO:

Suma <i>standard</i>	Notación binaria	Notación M
$x' + y' + z'$	0 0 0	M_0
$x' + y' + z$	0 0 1	M_1
$x' + y + z'$	0 1 0	M_2
$x' + y + z$	0 1 1	M_3
$x + y' + z'$	1 0 0	M_4
$x + y' + z$	1 0 1	M_5
$x + y + z'$	1 1 0	M_6
$x + y + z$	1 1 1	M_7

Utilizando esta notación, una función f podría representarse, por ejemplo, de la forma siguiente:

$$f = (x' + y + z')(x + y' + z)(x + y + z) = M_2 M_5 M_7 = \text{II } M \ 2, 5, 7 = \text{II } 2, 5, 7$$

3.3 CONVERSION ENTRE FORMAS BASICAS

Dos formas básicas, disyuntiva y conjuntiva, llamadas también formas canónicas, se han explicado en los puntos precedentes.

Estas formas básicas se caracterizan por los hechos siguientes:

- La función de conmutación de un problema dado puede expresarse en forma disyuntiva o conjuntiva.
- Cada forma básica implica un esquema de realización diferente.
- Es posible la conversión entre formas básicas.

A continuación se presentan dos teoremas relativos a la conversión entre formas básicas, y se resuelven dos ejemplos ilustrativos.

Teorema I

La negación de una función de conmutación puede obtenerse mediante los productos standards, no contenidos en la función original, unidos por operaciones OR.

Sea

n = número de variables.

2^n = número de productos *standards*: $m_0, m_1, \dots, m_{2^n-1}$.

Los 2ⁿ productos pueden ser, arbitrariamente, divididos en dos grupos, a partir de los cuales, mediante operaciones *OR*, se obtienen las funciones *A* y *B*:

$$\begin{aligned} A &= m_a + m_b + m_c + \dots \\ B &= m_\alpha + m_\beta + m_\gamma + \dots \end{aligned}$$

Hay que probar que

$$A' = B \Rightarrow \begin{cases} A + B = 1 & [1] \\ A \cdot B = 0 & [2] \end{cases}$$

En el caso de *n* variables, habrá siempre *en un instante determinado* un producto (y sólo uno) que tenga el valor verdadero 1.

Sea *m_x* este producto; *m_x* puede pertenecer a *A* o *B*.

$$\text{Si } m_x \text{ pertenece a } A \Rightarrow \begin{cases} A = 1 \\ B = 0 \end{cases}$$

$$\text{Si } m_x \text{ pertenece a } B \Rightarrow \begin{cases} A = 0 \\ B = 1 \end{cases}$$

Como se satisfacen las ecuaciones [1] y [2], queda demostrado que $A' = B$.

Teorema II

La negación de una función de conmutación puede obtenerse mediante las sumas standards, no contenidas en la función original, unidas por operaciones AND.

Sea

n = número de variables de conmutación.

2ⁿ = número de sumas *standards*: *M*₀, *M*₁, *M*₂, ..., *M*_{2ⁿ-1}.

Las 2ⁿ sumas pueden ser, arbitrariamente, divididas en dos grupos, a partir de los cuales, mediante operaciones *AND*, se obtienen las funciones *A* y *B*:

$$\begin{aligned} A &= M_a \cdot M_b \cdot M_c \dots \\ B &= M_\alpha \cdot M_\beta \cdot M_\gamma \dots \end{aligned}$$

Hay que demostrar que $A' = B$, o lo que es lo mismo:

$$\begin{aligned} A + B &= 1 & [1] \\ A \cdot B &= 0 & [2] \end{aligned}$$

Sea M_x la suma *standard*, que en un instante determinado tenga el valor verdadero cero. M_x puede pertenecer a A o B .

Si M_x pertenece a A , entonces $A = 0$ y $B = 1$

Si M_x pertenece a B , entonces $B = 0$ y $A = 1$

Como se satisfacen las ecuaciones [1] y [2] queda demostrado que $A' = B$.

EJEMPLO 1: *Conversión de forma disyuntiva a conjuntiva.*

Sea

$$f = x y' z' + x' y' z' + x y' z + x y z$$

1.º Se tiene que:

$$\left. \begin{array}{l} x y' z' = m_4 \\ x' y' z' = m_0 \\ x y' z = m_5 \\ x y z = m_7 \end{array} \right\} f = m_0 + m_4 + m_5 + m_7$$

2.º Se obtiene la negación de la función f :

$$f' = (m_0 + m_4 + m_5 + m_7)' = m'_0 \cdot m'_4 \cdot m'_5 \cdot m'_7$$

$$m'_0 = (x' y' z')' = x + y + z = M_7$$

$$m'_4 = (x y' z')' = x' + y + z = M_3$$

$$m'_5 = (x y' z)' = x' + y + z' = M_2$$

$$m'_7 = (x y z)' = x' + y' + z' = M_0$$

de donde

$$f' = M_0 M_2 M_3 M_7$$

3.º Aplicando el teorema II:

$$f = (f')' = M_1 M_4 M_5 M_6$$

Entre m_x y M_x existe la siguiente relación, que damos sin demostrar:

$$m'_x = M_{2^n - 1 - x}$$

EJEMPLO 2: *Conversión de forma conjuntiva a disyuntiva.*

Sea

$$f = (x' + y' + z)(x + y' + z')(x + y' + z)(x + y + z')$$

1.º Se tiene que:

$$\left. \begin{array}{l} x' + y' + z = M_1 \\ x + y' + z' = M_4 \\ x + y' + z = M_5 \\ x + y + z' = M_6 \end{array} \right\} f = M_1 \cdot M_4 \cdot M_5 \cdot M_6$$

2.º Se obtiene la negación de la función f :

$$f' = (M_1 M_4 M_5 M_6)' = M'_1 + M'_4 + M'_5 + M'_6$$

$$M'_1 = (x' + y' + z)' = x y z' = m_6$$

$$M'_4 = (x + y' + z') = x' y z = m_3$$

$$M'_5 = (x + y' + z) = x' y z' = m_3$$

$$M'_6 = (x + y + z') = x' y' z = m_1$$

de donde

$$f' = m_1 + m_2 + m_3 + m_6$$

3.º Aplicando el teorema I:

$$f = (f)' = m_0 + m_4 + m_5 + m_7$$

Entre M_x y m_x existe la siguiente relación, que damos sin demostrar:

$$M'_x = m_{2^n - 1 - x}$$

CAPITULO 4

SIMPLIFICACION DE FUNCIONES DE CONMUTACION

4.1 REPRESENTACION GRAFICA DE VARIABLES Y FUNCIONES:
METODO DE VEITCH-KARNAUGH

El mapa de Karnaugh es un método utilizado para la representación gráfica de una función de conmutación. Un mapa de Karnaugh para cuatro variables se muestra en la figura 4.1.

$w' x' y' z'$	$w' x' y' z$	$w' x' y z$	$w' x' y z'$
$w' x y' z'$	$w' x y' z$	$w' x y z$	$w' x y z'$
$w x y' z'$	$w x y' z$	$w x y z$	$w x y z'$
$w x' y' z'$	$w x' y' z$	$w x' y z$	$w x' y z'$

FIG. 4.1

El mapa contiene un total de 16 áreas, una por cada posible producto *standard*. Puede observarse que las áreas adyacentes difieren solamente en una variable. También son adyacentes las áreas *correspondientes* de las filas primera y última, y las de las columnas primera y cuarta.

En general, cada mapa debe tener 2^n áreas, donde n es el número de variables que intervienen en el producto.

En la práctica, el mapa de Karnaugh de la figura 4.1 se dispone tal como se indica a continuación:

$w x \backslash y z$		00	01	11	10	(Ejemplo)
		00	0	0	1	
01	0	1	1	0		
11	—	1	1	—		
10	—	0	1	—		

FIG. 4.2

En el mapa de la figura 4.2, las variables se colocan en la esquina superior izquierda, colocándose en cada fila y columna los valores correspondientes de las variables.

Para representar en un mapa de Karnaugh una función f , en este caso de cuatro variables, se procede de la siguiente forma:

1. Se coloca un «1» en las áreas que corresponden a los productos *standards* contenidos en dicha función.
2. Se coloca un «0» en las áreas que corresponden a las productos *standards* no contenidos en la función dada.
3. En el caso de existir condiciones opcionales (*), se coloca un guión (—) en las áreas correspondientes.

Cuando la función dada está especificada en forma de suma de productos, la figura 4.3 es de utilidad para la localización de las áreas apropiadas.

$w x \backslash y z$		00	01	11	10
		00	0	1	3
01	4	5	7	6	
11	12	13	15	14	
10	8	9	11	10	

FIG. 4.3

(*) También llamadas *don't care conditions*, *optional conditions* e *invalid conditions*. Se trata de combinaciones de entrada que no se presentan o que no tienen influencia en la salida.

Los mapas que corresponden a funciones de 1, 2 y 3 variables, constan de 2, 4 y 8 áreas, respectivamente, y se muestran a continuación.

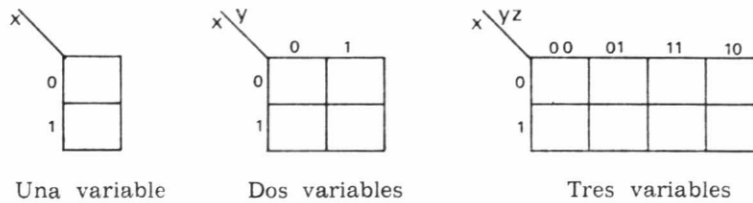


FIG. 4.4

El método de representación explicado es aplicable al caso de funciones de conmutación expresadas en forma de sumas de productos *standards*.

4.2 SIMPLIFICACION DE FUNCIONES

Se pretende obtener una función de conmutación que contenga un número mínimo de sumas de productos o el menor número de términos literales.

Una función de conmutación puede simplificarse:

- a) Por procedimientos algebraicos.
- b) Por medio de mapas de Karnaugh. Se pueden utilizar dos métodos:
 - Agrupamientos de áreas que contengan un 1.
 - Agrupamientos de áreas que contengan un 0.

El fundamento teórico de los agrupamientos de las áreas de un mapa de Karnaugh se basa en propiedades y teoremas del álgebra de Boole:

1. Cuando dos áreas se combinan para formar un grupo, se aplica la propiedad

$$x y + x' y = (x + x') y = y$$

es decir, los productos *standards* considerados difieren solamente en que en uno de ellos aparece la variable negada, y en el otro, sin negar. Esta propiedad es cierta para todas las áreas adyacentes.

2. Un agrupamiento de cuatro áreas puede obtenerse por combinación de dos grupos de dos áreas cada uno. Se pueden presentar varias posibilidades:
 - Que las cuatro áreas constituyan una columna.
 - Que las cuatro áreas constituyan una fila.
 - Que formen un rectángulo.
 - Que un agrupamiento pertenezca a la primera fila (o columna) y el otro a la última fila (o columna), y ambos, a las mismas columnas (o filas).
 - Que las cuatro áreas correspondan a las cuatro esquinas del mapa.

(Lo anterior es aplicable a mapas de Karnaugh de hasta cuatro variables.)

3. Rectángulos que correspondan a un número de áreas que no sea una potencia de dos, no pueden formar agrupamientos.
4. Una misma área puede ser empleada en varios agrupamientos.
5. La elección de agrupamientos puede conducir a diferentes funciones simplificadas, es decir, pueden existir varias alternativas de simplificación para una función dada.

Al usar el método de agrupamiento de áreas que contengan un 1, se obtiene la función f como suma de productos *standards*.

Si se usa el método de agrupamientos de áreas que contengan un 0, se obtiene la función f' como suma de productos *standards* (teorema I). La obtención de f es inmediata, ya que basta aplicar $f = (f)'$.

El conjunto de valores de las variables, para los cuales el valor de la función no está especificado, constituye las condiciones opcionales (*don't care conditions*). Las áreas que corresponden a estas condiciones opcionales pueden ser agrupadas con las áreas marcadas con 1, si con ello se facilita la simplificación de la función. También pueden ser agrupadas con áreas que contengan un 0, si se emplea el método de agrupamientos de áreas marcadas con 0.

4.3 UTILIZACION DEL MAPA DE KARNAUGH PARA SIMPLIFICAR FUNCIONES EXPRESADAS EN FORMA DE PRODUCTO DE SUMAS

Sea la función

$$f = M_6 \cdot M_7 \cdot M_{14} \cdot M_{15}$$

donde

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

En primer lugar se representan las sumas en un mapa de Karnaugh. Para ello, se sabe que:

	Notación binaria
$M_6 = w' + x + y + z'$	0 1 1 0
$M_7 = w' + x + y + z$	0 1 1 1
$M_{14} = w + x + y + z'$	1 1 1 0
$M_{15} = w + x + y + z$	1 1 1 1

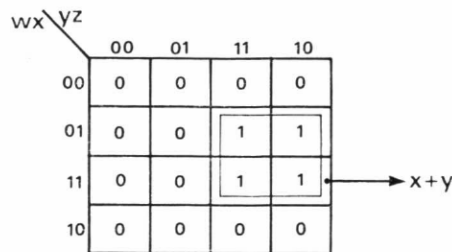


FIG. 4.5

Nótese que se ha representado cada *suma «standard»* con un 1.

Las áreas 11 11 y 11 10 son adyacentes y se verifica que

$$(w + x + y + z)(w + x + y + z') = w + x + y$$

Las áreas 01 11 y 01 10 también son adyacentes:

$$(w' + x + y + z)(w' + x + y + z') = w' + x + y$$

Además,

$$(w' + x + y)(w + x + y) = x + y$$

con lo cual

$$f = x + y$$

4.4 MAPA DE KARNAUGH PARA CINCO VARIABLES

El método del mapa de Karnaugh, expuesto en los puntos anteriores es aplicable solamente en los casos en que las funciones de conmutación sean de hasta cuatro variables.

La extensión del mapa de Karnaugh para funciones de más de cuatro variables presenta algunos problemas. Sabemos que un producto *standard* de cuatro variables de conmutación puede combinarse con uno cualquiera de otros cuatro productos *standards* para eliminar una variable. Esto se debe a que en un mapa de Karnaugh de cuatro variables, cada área tiene otras cuatro áreas adyacentes.

Cuando hay cinco variables, un producto *standard* puede combinarse con uno cualquiera de otros cinco productos para eliminar una variable. La dificultad radica en encontrar un diagrama que suministre cinco áreas adyacentes para cada área del mismo,

En la figura 4.6 se muestra un mapa de Karnaugh para cinco variables.

		wxyz							
		000	001	011	010	110	111	101	100
00		0	1	3	2	6	7	5	4
01		8	9	11	10	14	15	13	12
11		24	25	27	26	30	31	29	28
10		16	17	19	18	22	23	21	20

eje de simetría

FIG. 4.6

La línea central punteada puede considerarse como un espejo o eje de simetría. Cada área es adyacente:

- a) de sus cuatro áreas cercanas;
- b) de su imagen en el espejo.

En dicha figura se indica el área correspondiente a cada uno de los 32 productos *standards*. El área 11 es adyacente de las áreas 3, 9, 10, 27 y 15.

4.5 METODO DE TABULACION DE QUINE - MCCLUSKEY

Este método puede usarse para simplificar funciones de conmutación de muchas variables. Tiene la ventaja de que puede ser programado en ordenador.

Estudiaremos este método mediante dos ejemplos.

EJEMPLO 1. Simplificar una función dada, sin condiciones opcionales.

Sea

$$f(w, x, y, z) = \Sigma 0, 2, 3, 5, 7, 8, 10, 11, 13, 15$$

Primero. Se representan los productos *standards* en notación binaria:

0	0	0	0	0	0
2	0	0	1	0	0
3	0	0	1	1	1
5	0	1	0	1	1
7	0	1	1	1	1
8	1	0	0	0	0
10	1	0	1	0	0
11	1	0	1	1	1
13	1	1	0	1	1
15	1	1	1	1	1

Segundo. Se agrupan las representaciones binarias de acuerdo con el número de «unos» que contengan.

0	0	0	0	0	✓
2	0	0	1	0	✓
8	1	0	0	0	✓
3	0	0	1	1	✓
5	0	1	0	1	✓
10	1	0	1	0	✓
7	0	1	1	1	✓
11	1	0	1	1	✓
13	1	1	0	1	✓
15	1	1	1	1	✓

TABLA 4.1

Tercero. Se pueden realizar eliminaciones solamente entre miembros de grupos adyacentes, que son grupos cuyo número de «unos» difiere solamente en una

unidad. Cada miembro se compara con todos los miembros de su grupo adyacente. Esta tarea comienza lógicamente con el grupo «cero». Por ejemplo: 0000 se compara con 0010. Se elimina el dígito correspondiente a la variable y , y el resultado se representa de la forma: 00—0 (El guión [—] indica que se ha eliminado la variable y .)

A continuación se coloca una marca de comprobación en los términos 0000 y 0010 de la tabla 4.1, y se lleva 00—0 a una nueva tabla. Este proceso continúa hasta que todas las comparaciones se han realizado.

0,2	0	0	—	0	✓
0,8	—	0	0	0	✓
2,3	0	0	1	—	✓
2,10	—	0	1	0	✓
8,10	1	0	—	0	✓
3,7	0	—	1	1	✓
3,11	—	0	1	1	✓
5,7	0	1	—	1	✓
5,13	—	1	0	1	✓
10,11	1	0	1	—	✓
7,15	—	1	1	1	✓
11,15	1	—	1	1	✓
13,15	1	1	—	1	✓

TABLA 4.2

El proceso anterior se vuelve a aplicar a la tabla 4.2, teniendo en cuenta que para que dos términos puedan compararse es necesario que los guiones estén situados en las mismas columnas de ambos. Así se obtiene la tabla 4.3.

A	0, 2, 8, 10	—	0	—	0
B	2, 3, 10, 11	—	0	1	—
C	3, 7, 11, 15	—	—	1	1
D	5, 7, 13, 15	—	1	—	1

TABLA 4.3

El proceso de eliminación termina en la tabla 4.3.

Cuarto. En las tablas 4.1, 4.2 y 4.3 hay términos que tienen marcas de comprobación y otros que no la tienen.

Una versión simplificada de la función dada se puede obtener mediante la combinación *OR* de todos los términos que no tengan marca de comprobación.

Así, pues:

$$f(w, x, y, z) = x'z' + x'y + yz + xz$$

Sin embargo, esta función f puede tener términos redundantes. Para evitarlos, se construye el siguiente diagrama:

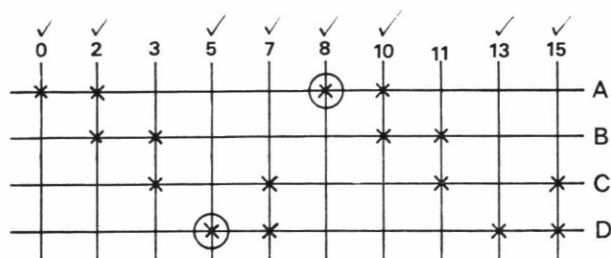


FIG. 4.7

En dicho diagrama hay tantas columnas como productos *standards* tenga la función que se quiere simplificar y tantas filas como términos que no tengan marca de comprobación.

Se ha visto que A resulta de la combinación de m_0 , m_2 , m_8 y m_{10} . Esto se indica colocando una cruz en los cruces correspondientes. Este proceso se repite para B , C y D .

En la figura 4.7 existe un solo cruce debajo del producto *standard* m_5 . La expresión final simplificada debe contener m_5 y, por consiguiente D .

A continuación se encierra en un círculo el cruce $m_5 - D$ y se colocan marcas de comprobación en 5, 7, 13 y 15.

El mismo razonamiento es aplicable al cruce $m_8 - A$. Se colocan marcas de comprobación en 0, 2, 8 y 10.

En la expresión final hay que incluir, además, m_3 y m_{11} . Para ello hay que incluir B o C , pero no ambos simultáneamente.

Así, pues, la solución final puede ser:

$$f(w, x, y, z) = A + D + B = xz + x'y + x'z'$$

o

$$f(w, x, y, z) = A + D + C = xz + yz + x'z'$$

EJEMPLO 2. Simplificar una función dada, con condiciones opcionales.

Cuando aparezcan condiciones opcionales, se utilizarán para simplificar la función dada y se tratarán como si fuesen productos *standards* de ésta.

En el diagrama final de simplificación se ignorarán estas condiciones opcionales, por no ser necesario que aparezcan en dicha expresión.

Sea

$$f(w, x, y, z) = \Sigma 2, 3, 4, 9, 11$$

condiciones opcionales 5, 6, 10, 12, 13 ó 14.

Primero. Representación binaria de los productos *standards*.

2	0	0	1	0	✓
3	0	0	1	1	✓
4	0	1	0	0	✓
5	0	1	0	1	✓
6	0	1	1	0	✓
9	1	0	0	1	✓
10	1	0	1	0	✓
11	1	0	1	1	✓
12	1	1	0	0	✓
13	1	1	0	1	✓
14	1	1	1	0	✓

Segundo. Formación de grupos.

2	0	0	1	0	✓
4	0	1	0	0	✓
<hr/>					
3	0	0	1	1	✓
5	0	1	0	1	✓
6	0	1	1	0	✓
9	1	0	0	1	✓
10	1	0	1	0	✓
12	1	1	0	0	✓
<hr/>					
11	1	0	1	1	✓
13	1	1	0	1	✓
14	1	1	1	0	✓

TABLA 4.4

Tercero. Simplificación.

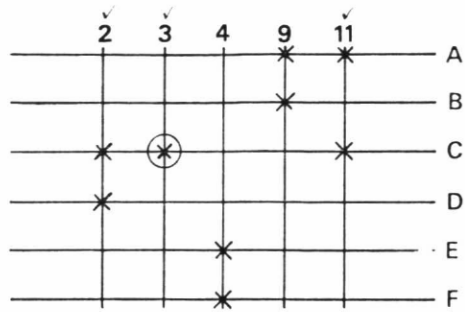
2,3	0	0	1	—	✓
2,6	0	—	1	0	✓
2,10	—	0	1	0	✓
4,5	0	1	0	—	✓
4,6	0	1	—	0	✓
4,12	—	1	0	0	✓
<hr/>					
3,11	—	0	1	1	✓
5,13	—	1	0	1	✓
6,14	—	1	1	0	✓
9,11	1	0	—	1	A
9,13	1	—	0	1	B
10,11	1	0	1	—	✓
10,14	1	—	1	0	✓
12,13	1	1	0	—	✓
12,14	1	1	—	0	✓

TABLA 4.5

TABLA 4.6

2, 3, 10, 11	—	0	1	—	C
2, 6, 10, 14	—	—	1	0	D
4, 5, 12, 13	—	1	0	—	E
4, 6, 12, 14	—	1	—	0	F

Cuarto. Eliminación de redundancias.



En este último diagrama se ignoran las condiciones opcionales, ya que sólo es necesario determinar si se han incluido todos los productos *standards* de la función dada.

La función dada puede expresarse de las siguientes formas:

$$f_1 = C + E + A = x'y + xy' + wx'z$$

$$f_2 = C + E + B = x'y + xy' + wy'z$$

$$f_3 = C + F + A = x'y + xz' + wx'z$$

$$f_4 = C + F + B = x'y + xz' + wy'z$$

CAPITULO 5
RELE LOGICO

5.1 REPRESENTACION. PUERTAS AND Y OR

Los componentes de conmutación deben satisfacer algunos requerimientos, tales como:

- Capacidad de representar sin ambigüedades dos estados distintos.
- La transición de un estado a otro debe ser controlable.
- Características de operación.
- Fiabilidad, etc.

Estos requerimientos se cumplen por los relés electromagnéticos, cuyas partes más importantes son: núcleo, bobina, culata, armadura y contactos.

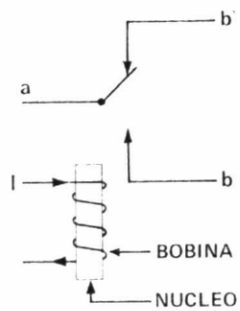


FIG. 5.1

Ya se ha hablado de los diferentes tipos de contactos:

- Contacto *cerrado* o de reposo, que requiere dos láminas.
- Contacto *abierto* o de trabajo, que también requiere dos láminas.
- Contacto de *transferencia* o de reposo-trabajo, que requiere dos láminas fijas y una móvil.
- Contacto de *transferencia con continuidad* o de trabajo-reposo, que requiere dos láminas móviles y una fija.

En la figura 5.1 puede verse un contacto de transferencia.

Un relé puede representarse por una variable X , de forma que:

$X = 1$ implica la existencia de una corriente I en la bobina del relé, así como que el contacto $a b$ está cerrado y $a b'$ abierto.

$X = 0$ implica la ausencia de corriente por la bobina del relé. En este caso, $a b$ está abierto y $a b'$ cerrado.

Los relés pueden utilizarse para instrumentar puertas *AND* y *OR*.

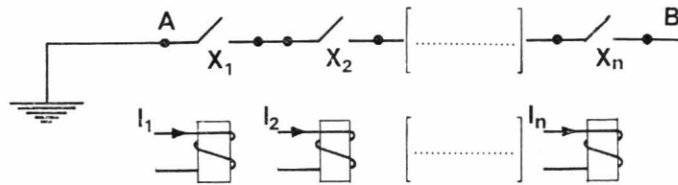


FIG. 5.2.—PUERTA AND

Una puerta *AND* de n entradas, X_1, X_2, \dots, X_n , puede instrumentarse conectando en serie n contactos de relés. Si los terminales A y B están conectados, el producto $X_1 \cdot X_2 \cdot \dots \cdot X_n$ vale 1; en caso contrario, vale 0. Por convenio, el terminal A se conecta a tierra y la aparición en B de dicho potencial indica que el valor verdadero del producto es 1.

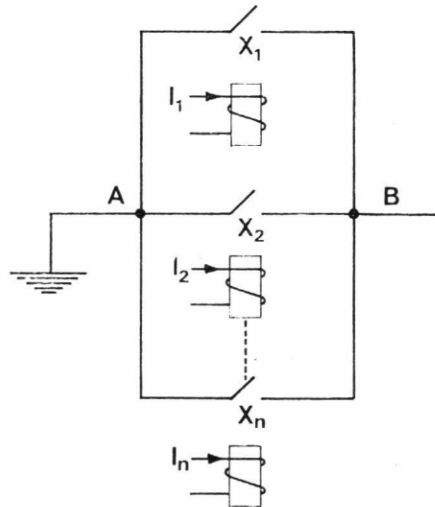


FIG. 5.3.—PUERTA OR

Una puerta *OR* de n entradas, X_1, X_2, \dots, X_n , puede instrumentarse conectando «en paralelo» n contactos de relés. Si los terminales A y B están conectados, la suma $X_1 + X_2 + \dots + X_n$ vale 1; en caso contrario, vale 0. Por convenio, el terminal A se conecta a tierra y la aparición en B de dicho potencial indica que el valor verdadero de la suma es 1.

Por último, conviene indicar que cualquier función de conmutación de un circuito combinatorio de conmutación puede ser instrumentada con relés. Un circuito *combinatorio* es un circuito cuyas salidas en un instante determinado de tiempo están solamente determinadas por las entradas en dicho instante de tiempo.

5.2 OPTIMIZACION DE CIRCUITOS

Existen muchos factores que intervienen en la definición de un sistema óptimo de conmutación: características de operación, coste, seguridad, etc. En circuitos de conmutación con relés se emplea un criterio restringido, considerando que el circuito óptimo es aquel que emplea un *número mínimo de láminas de contacto* para instrumentar una función dada.

Para cumplir este requerimiento se realizan tres simplificaciones:

- a) Se reduce el número de «términos literales» en la función de conmutación. Un «término literal» es una variable de conmutación, negada o no negada.
- b) Siempre que sea posible, se emplean contactos de transferencia, lo que representa el ahorro de una lámina de contacto por cada contacto de transferencia utilizado. Esta simplificación se realiza inspeccionando el circuito.
- c) Una vez realizadas las dos simplificaciones anteriores, se intenta una reducción del circuito mediante un proceso de inspección. Si se efectúan reducciones por este procedimiento, es necesario *comprobar* que si se crean nuevos caminos no especificados en la función original, las funciones de conmutación de éstos sean nulas. Ejemplo: Función de conmutación de un nuevo camino: $x y y' y z = 0$.

Si los caminos adicionales introducen *términos no nulos* en la función original (por ejemplo, el término $x y z'$), el circuito resultante puede ser incorrecto.

Conviene indicar que para un problema dado pueden existir *varias* soluciones óptimas.

Los criterios presentados son sencillos, pero no reflejan todas las consideraciones que se deben tener en cuenta. Además del número de láminas de contacto, se deben tener en cuenta otros factores: coste de fabricación, velocidad de conmutación, fiabilidad, etc. Actualmente la optimización de circuitos se resuelve mediante el empleo de programas adecuados en ordenadores.

EJEMPLO: Diseñese un circuito de relés que pueda conmutar una lámpara de encendido a apagado, y viceversa, mediante tres conmutadores, colocados cada uno de ellos en diferentes lugares. La lámpara estará encendida siempre que el número de conmutadores operados sea impar.

El diagrama de bloques es el siguiente:

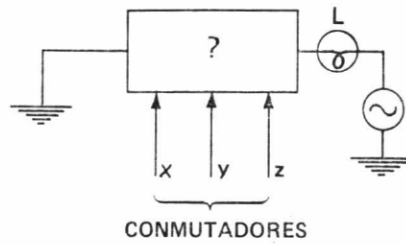


FIG. 5.4

Tabla de verdad

x	y	z	f
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

$$f = x' y' z + x' y z' + x y' z' + x y z$$

[1]

El circuito que corresponde a la ecuación [1] es:

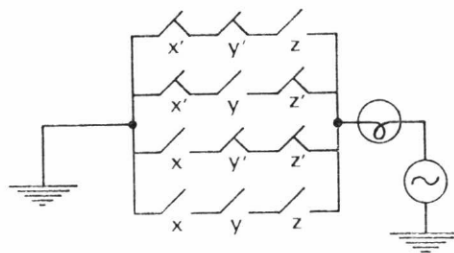


FIG. 5.5

Requiere un total de $12 \times 2 = 24$ láminas.

Optimización:

1.º Reducción del número de «términos literales»:

$$f = x'(y'z + yz') + x(yz + y'z')$$

[2]

Esta función requiere solamente 10 «términos literales»; la función [1] requería 12.

El circuito que corresponde a la función [2] es:

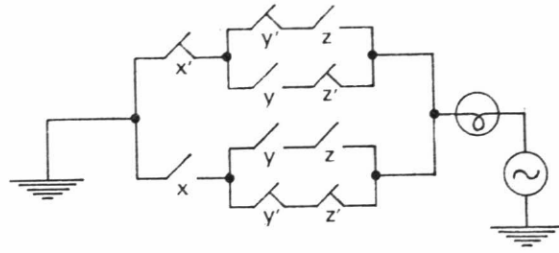


FIG. 5.6

Requiere un total de $10 \times 2 = 20$ láminas.

2.º Uso de contactos de transferencia:

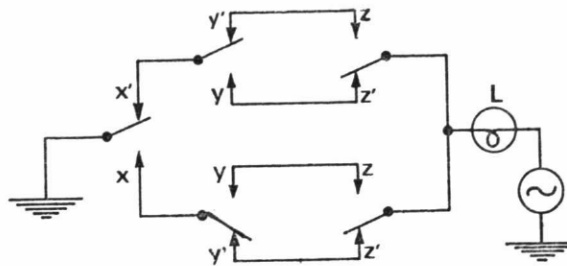


FIG. 5.7

Requiere un total de $5 \times 3 = 15$ láminas.

3.º Inspección del circuito:

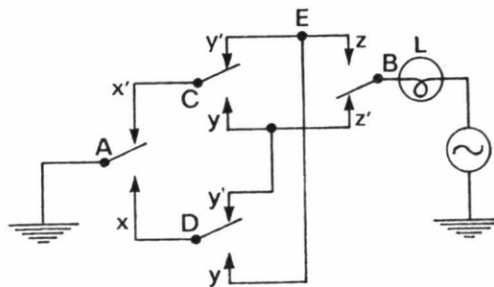


FIG. 5.8.—Se ha eliminado un contacto de transferencia

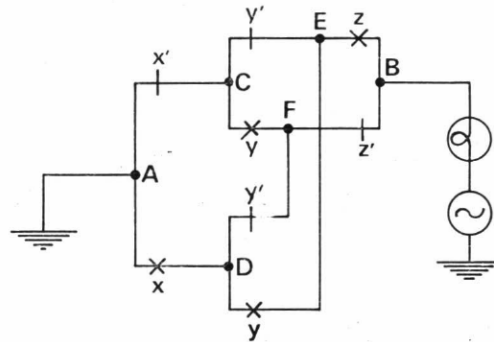
Requiere un total de $4 \times 3 = 12$ láminas.

4.º Comprobación de caminos:

Caminos	Función de conmutación
A - C - E - B	$x' y' z$
A - C - F - B	$x' y z'$
A - D - F - B	$x y' z'$
A - D - E - B	$x y z$
A - C - E - D - F - B	$x' y' y y' z' = 0$
A - C - F - D - E - B	$x' y y' y z = 0$
A - D - F - C - E - B	$x y' y y' z = 0$
A - D - E - C - F - B	$x y y' y z' = 0$

El circuito de la figura 5.8 es la solución óptima.

NOTA.—El circuito de la figura 5.8 también puede representarse de la siguiente forma:



5.3 REDES EN PUENTE

Una propiedad interesante de los contactos de relés es su carácter *bilateral*; es decir, la propiedad de transmitir un potencial determinado en ambos sentidos.

Esta característica se utiliza ventajosamente en los circuitos en puente, que proporcionan generalmente una instrumentación óptima de la función de conmutación dada.

Respecto a los circuitos en puente, conviene tener presente lo siguiente:

1. No es cierto que todas las funciones de conmutación puedan instrumentarse con circuitos en puente.
2. En los casos en que una función pueda materializarse con circuitos de este tipo, no existe un procedimiento sistemático de síntesis que sea fácilmente realizable.

Veamos un ejemplo:

Instrumentar con relés la función

$$f(w, x, y, z, t) = xz't + wx + wyz' + yt$$

Simplificando resulta:

$$f = x(w + z't) + y(wz' + t)$$

cuyo circuito correspondiente se muestra en la figura 5.9.

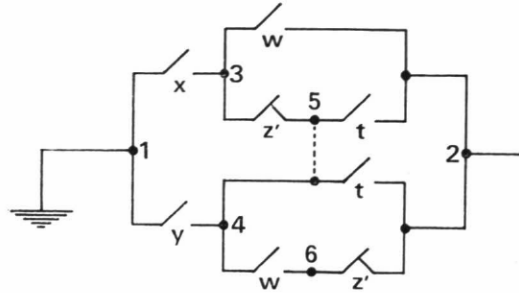


FIG. 5.9

Requiere un total de $8 \times 2 = 16$ láminas de contacto.

Se observa que se requieren dos contactos abiertos para instrumentar la variable t . Para eliminar uno de ellos, se realiza la conexión entre los puntos 4 y 5; el contacto t , situado entre los puntos 5 y 2, quedaría eliminado. Resulta el circuito de la figura 5.10.

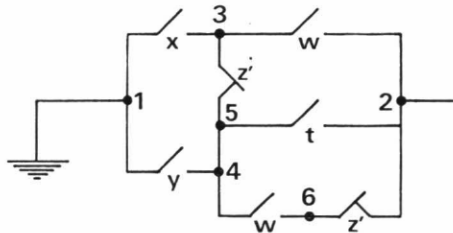


FIG. 5.10

Hay que comprobar que no se ha introducido ningún camino adicional no deseado.

CIRCUITO FIGURA 5.9		CIRCUITO FIGURA 5.10	
Caminos	Funciones	Caminos	Funciones
1-3-2	$w x$	1-3-2	$w x$
1-3-5-2	$x z' t$	1-3-5-2	$x z' t$
1-4-2	$y t$	1-4-5-2	$y t$
1-4-6-2	$w y z'$	1-4-6-2	$w y z'$
		1-3-5-4-6-2	$w x z'$
		1-4-5-3-2	$w y z'$

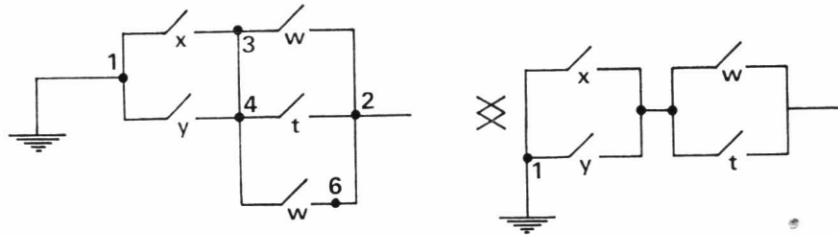
Se han creado los caminos adicionales 1-3-5-4-6-2 y 1-4-5-3-2. La función $w y z'$ ya pertenecía a f , por lo que el camino 1-4-5-3-2 no introduce un funcionamiento incorrecto. La función $w x z'$ está incluida implícitamente en f , ya que:

$$w x = w x(1 + z') = w x + w x z'$$

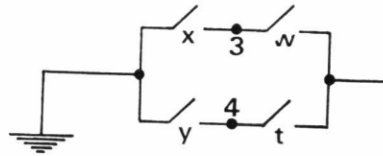
Por este hecho, el camino adicional 1-3-5-4-6-2 tampoco ocasiona un funcionamiento incorrecto del circuito.

La rama 4-6-2 puede eliminarse, ya que:

a) Con z' cerrado, el circuito de la figura 5.10 es equivalente a



b) Con z' abierto, la rama 4-6-2 no tiene influencia.



El circuito final es:

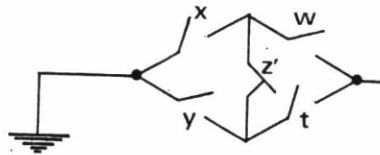


FIG. 5.11

Se requiere un total de $5 \times 2 = 10$ láminas de contacto. Dada la forma en que las variables en conmutación (w, x, y, z, t) intervienen en la función f , éste es el número mínimo de láminas de contacto que se requiere para instrumentar dicha función.

CAPITULO 6
DIODO LOGICO

6.1 REPRESENTACION Y CARACTERISTICAS DE LOS DIODOS DE CONMUTACION

Los diodos semiconductores pueden usarse como elementos de conmutación de dos estados. Estos estados están en relación con la resistencia que ofrece el diodo al paso de la corriente eléctrica:

- 1) En un estado actúa como una alta resistencia, cuando está inversamente polarizado.
- 2) En otro actúa como una baja resistencia, cuando está directamente polarizado.

La transición de un estado a otro se realiza mediante el cambio de la tensión de polarización.

El símbolo usual de un diodo se muestra en la figura 6.1. La flecha señala la dirección de baja resistencia.

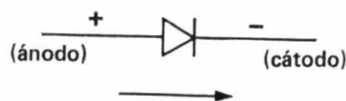


FIG. 6.1

En todos los casos supondremos que la transición dinámica entre ambos estados se realiza en un tiempo suficientemente corto, respecto a las variaciones de

las señales exteriores a las cuales está sometido, como para considerarlo nulo. Así, pues, admitiremos que los tiempos de conmutación entre estados son cero.

Como el diodo está sometido a condiciones cuasiestáticas la mayor parte del tiempo, resumamos su comportamiento estático mediante su curva característica, válida exclusivamente en estas condiciones. En la figura 6.2 se representa la curva característica típica de un diodo semiconductor.

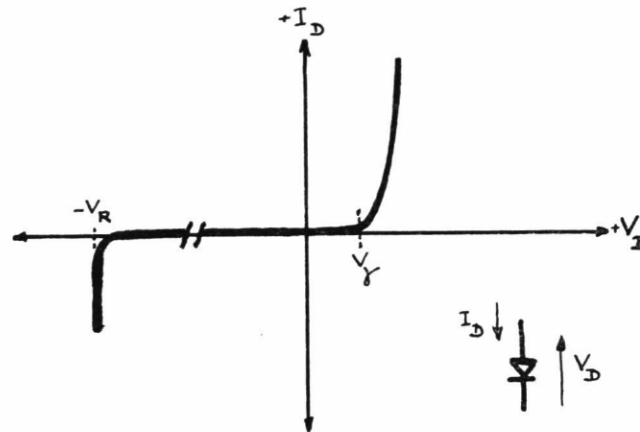


FIG. 6.2

Obsérvese que cuando el diodo está polarizado en sentido directo—primer cuadrante— apenas circula corriente si V_D no supera una cierta tensión umbral, V_γ . V_γ es igual a unos 0,6 voltios para los diodos de silicio, que son casi los exclusivamente utilizados en la actualidad. A partir de dicha tensión, cuando $V_D > V_\gamma$, circula una corriente que crece exponencialmente y que sólo es limitada por el circuito exterior. Ha de procurarse que esta corriente no supere la máxima permitida por el diodo, dato que suministra el fabricante.

Cuando el diodo se polariza inversamente—tercer cuadrante—, apenas circula corriente, si, como es habitual, no se supera la tensión $-V_R$ de ruptura de la unión.

Para nuestros propósitos consideraremos, en todos los casos, el diodo ideal: Cortocircuito en sentido directo y circuito abierto cuando esté polarizado inversamente. Si deseásemos una mejor simulación, el comportamiento en ambos estados vendría representado por las resistencias directa e inversa.

La resistencia directa de un diodo, r_f , disminuye cuando aumenta la tensión. Es del orden de unos cientos de ohmios.

La resistencia inversa de un diodo, r_b , apenas depende del valor absoluto de la tensión inversa. Es el orden de unos cientos de $M\Omega$.

Consideremos una puerta con diodo, de una sola entrada. Esta puerta no realiza ninguna función lógica, pero ayuda a describir el funcionamiento del circuito.

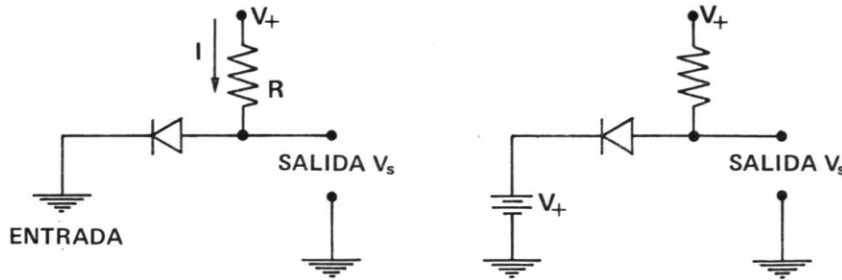


FIG. 6.3

Cuando la entrada está a cero voltios, la tensión de alimentación V_+ polariza el diodo en sentido directo a través de la resistencia R . La caída de tensión en el diodo es de unas décimas de voltio.

$$I = \frac{V_+}{R + r_f}$$

$$V_s = I \cdot r_f \simeq 0 \text{ voltios}$$

Cuando la entrada está a V_+ no fluye corriente a través del diodo, por estar polarizado en sentido inverso. La corriente es $I = 0$, y la tensión de salida es

$$V_s = V_+$$

CONCLUSIÓN: La salida sigue a la entrada.

6.2 PUERTAS AND Y OR

Sea el circuito de la figura 6.4.

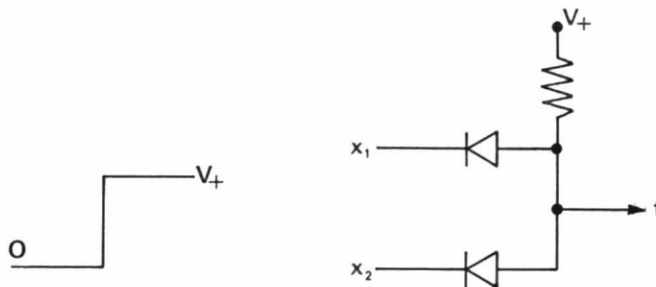


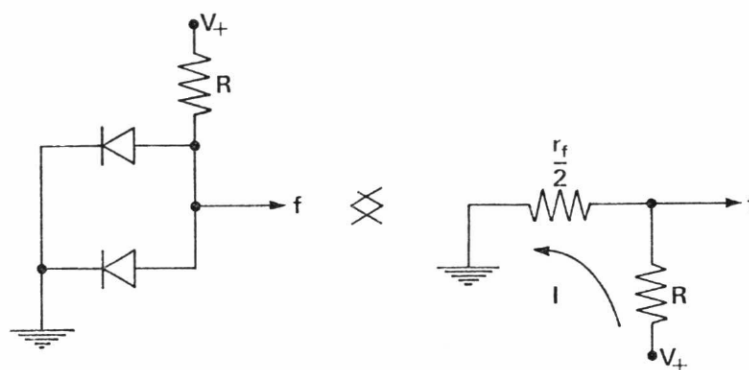
FIG. 6.4

Si ambas entradas están conectadas a masa, la salida será aproximadamente cero voltios. Si un diodo está al potencial de masa y el otro a V_+ , la salida estará a cero voltios y toda la corriente fluirá a través del diodo polarizado en sentido directo.

Si ambas entradas están a V_+ , los dos diodos están polarizados en sentido inverso y no fluye corriente a través de ellos.

Casos	x_1	x_2	f	
a	0	0	0	Las tensiones son ideales, ya que se considera: $V_D \simeq 0$ (caída en el diodo) $I_r \simeq 0$ (corriente inversa)
b	0	V_+	0	
c	V_+	0	0	
d	V_+	V_+	V_+	

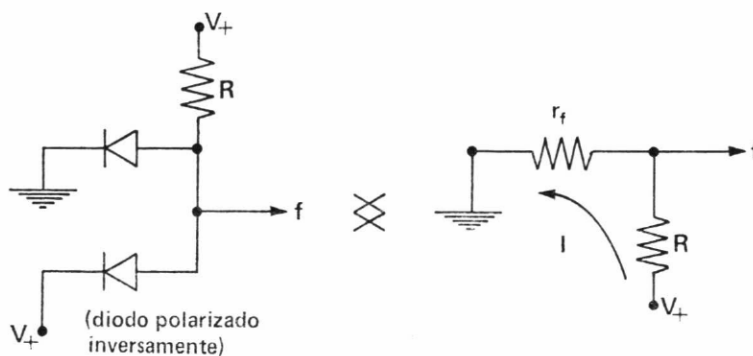
CASO a):



$$I = \frac{V_+}{R + \frac{r_f}{2}} \quad \left. \begin{array}{l} \\ \text{Si } r_f \ll R \end{array} \right\} I \simeq \frac{V_+}{R}$$

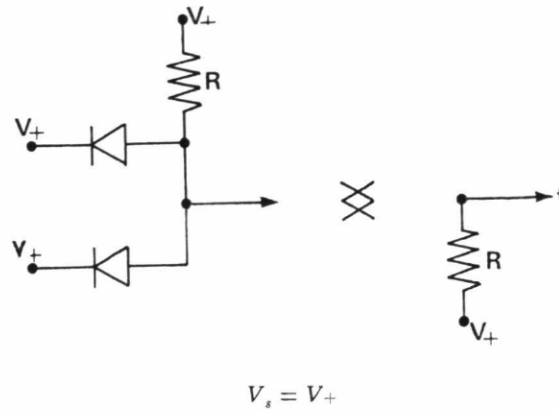
$$V_s = \frac{V_+}{R} \cdot \frac{r_f}{2} = \frac{r_f}{2R} \cdot V_+ \simeq 0 \text{ voltios}$$

CASOS b) Y c):



$$I = \frac{V_+}{R + r_f} \left\{ \begin{array}{l} I \simeq \frac{V_+}{R} \\ \text{Si } r_f \ll R \end{array} \right. \left. \begin{array}{l} V_s = \frac{V_+}{R} \cdot r_f = \frac{r_f}{R} (V_+) \simeq 0 \text{ voltios} \end{array} \right.$$

CASO d):



Las tensiones cero voltios y V_+ voltios pueden emplearse para representar los estados lógicos 0 y 1.

Si cero voltios representa la condición lógica 0, y V_+ voltios representa la condición lógica 1, se trata de *lógica positiva*, ya que 1 se exige para representar la tensión más positiva. La elección inversa corresponde a la *lógica negativa*.

Con lógica positiva se obtiene:

x_1	x_2	f	}	$f = x_1 x_2$ Se trata de una puerta AND
0	0	0		
0	1	0		
1	0	0		
1	1	1		

Con lógica negativa se obtiene:

x_1	x_2	f	}	$f = x_1 + x_2$ Se trata de una puerta OR
1	1	1		
1	0	1		
0	1	1		
0	0	0		

Sea el circuito de la figura 6.5.

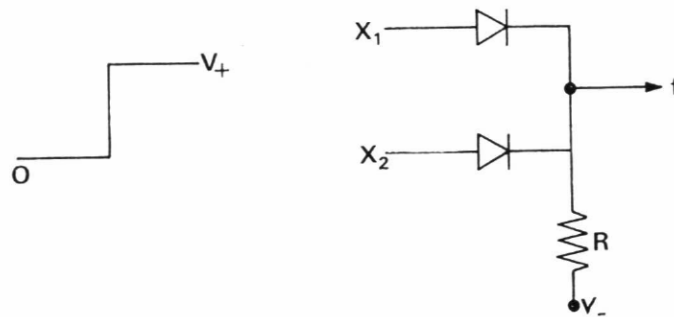


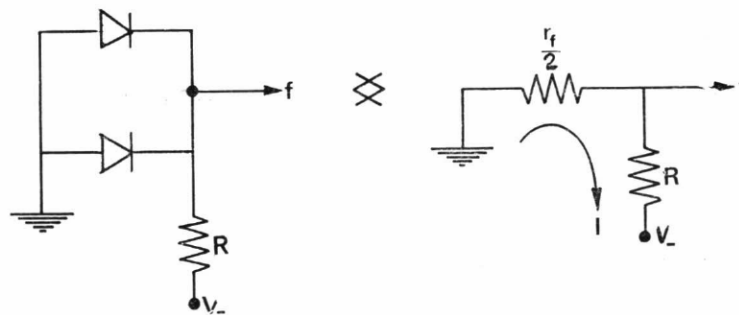
FIG. 6.5

Esta puerta se diferencia de la anterior en que las entradas se aplican a los ánodos de los diodos y la resistencia R se conecta a V_- . Los niveles lógicos son V_+ voltios y 0 voltios. Cuando ambas entradas están a 0 voltios, la salida será aproximadamente 0 voltios.

Con una de las entradas a 0 voltios y otra a V_+ voltios, la salida estará a V_+ voltios, ya que un diodo estará polarizado en sentido directo y el otro en sentido inverso. Cuando ambos diodos tienen sus entradas a V_+ , la salida será también V_+ .

Casos	x_1	x_2	f
a	0	0	0
b	0	V_+	V_+
c	V_+	0	V_+
d	V_+	V_+	V_+

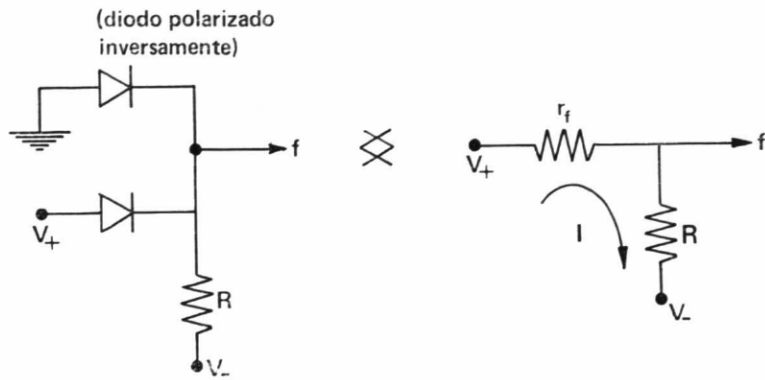
Caso a):



$$I = \frac{V_-}{R + \frac{r_f}{2}} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} I \approx \frac{V_-}{R}$$

$$\text{Como } r_f \ll R \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} V_s = \frac{V_-}{R} \cdot \frac{r_f}{2} = \frac{r_f}{2R} \cdot (V_-) \approx 0 \text{ voltios}$$

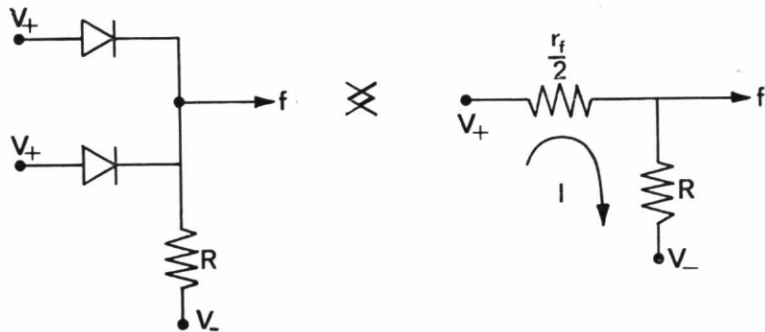
CASOS b) y c):



$$I = \frac{V_+ + V_-}{R + r_f} \approx \frac{V_+ + V_-}{R}$$

$$V_s = V_+ - \underbrace{\frac{V_+ + V_-}{R} \cdot r_f}_{\text{Caída de tensión en el diodo}} \approx V_+ \text{ voltios}$$

CASO d):



$$I = \frac{V_+ + V_-}{R + \frac{r_f}{2}} \approx \frac{V_+ + V_-}{R}$$

$$V_s = V_+ - \frac{V_+ + V_-}{R} \cdot \frac{r_f}{2} \approx V_+ \text{ voltios}$$

Con lógica positiva se obtiene:

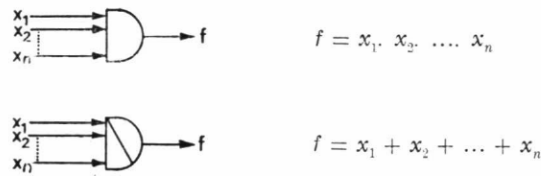
x_1	x_2	f
0	0	0
0	1	1
1	0	1
1	1	1

$\left. \vphantom{\begin{matrix} x_1 & x_2 & f \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{matrix}} \right\} \begin{array}{l} f = x_1 + x_2 \\ \text{Se trata de una puerta OR} \end{array}$

Con lógica negativa se obtiene:

x_1	x_2	f	}	$f = x_1 \cdot x_2$ Se trata de una puerta <i>AND</i>
1	1	1		
1	0	0		
0	1	0		
0	0	0		

Hasta el momento sólo se han considerado dos entradas, x_1 y x_2 . No es necesario restringir el número de entradas de cada puerta, sino que en general éstas serían de la forma:



Es interesante señalar que la *puerta NOT* no puede realizarse con diodos.

6.3 EFECTO DE UNA IMPEDANCIA DE CARGA FINITA

Para estudiar este fenómeno emplearemos las puertas para lógica positiva de la figura 6.6. Se ha conectado una puerta *AND* con una puerta *OR*.

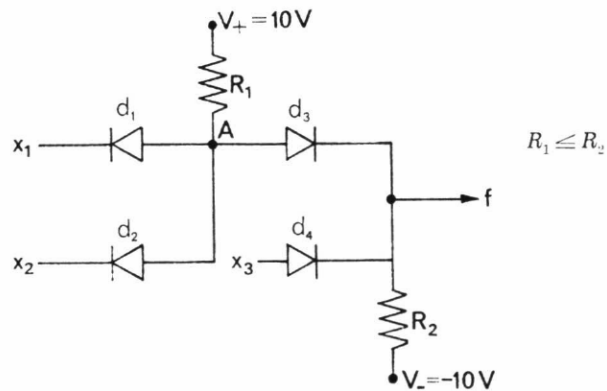


FIG. 6.6

Cuando x_1 y x_2 son 0 voltios, la tensión en el punto A es $+V_{\text{diodo}}$. Si x_3 es 10 voltios, la tensión de salida V_s es

$$V_s = 10 - V_{\text{diodo}} \simeq 10 \text{ voltios}$$

Las puertas funcionan adecuadamente con estas entradas.

Sea $x_1 = x_2 = 10$ voltios y $x_3 = 0$ voltios. En estas condiciones, V_s debería ser 10 voltios, ya que los dos diodos de la puerta AND están inversamente polarizados. ¿Qué ocurre exactamente?

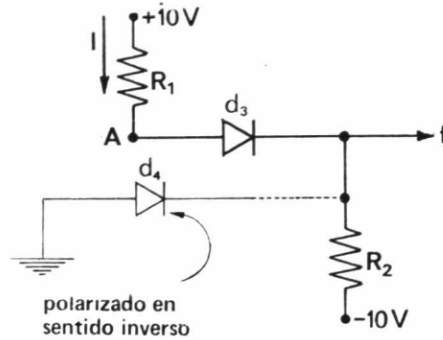


FIG. 6.7

La puerta OR tiene una impedancia de entrada finita y carga sobre el punto A. Se va a despreciar la caída de tensión en el diodo d_3 . El circuito que resulta se muestra en la figura 6.7. En este caso:

$$I = \frac{V_+ + V_-}{R_1 + R_2 + r_f} \simeq \frac{V_+ + V_-}{R_1 + R_2}$$

Si

$$R_1 = R_2 = 1 \text{ K} \Omega \quad , \quad I = \frac{20}{R_1 + R_2 + r_f} \simeq \frac{20}{R_1 + R_2} \quad V_A = V_s = \frac{20}{R_1 + R_2} \cdot R_2 - 10 = \frac{20}{2} - 10 = 0 \text{ voltios}$$

Es evidente que la puerta no funciona, en este caso, del modo adecuado.

Si

$$R_2 = 10 R_1 \quad , \quad V_s = \frac{20}{11 R_1} \cdot 10 R_1 - 10 = 8,2 \text{ voltios}$$

Esta tensión es lo suficientemente alta para que la puerta funcione de modo adecuado.

¿Qué pasaría si la puerta AND alimentase a dos puertas OR con $R_2 = 10 R_1$?

En este caso, la carga equivalente de las puertas sería:

$$R'_2 = \frac{R_2 R_2}{R_2 + R_2} = \frac{10 R_1 \times 10 R_1}{20 R_1} = 5 R_1$$

en cuyo caso

$$V_s = \frac{20}{R_1 + 5 R_1} \cdot 5 R_1 - 10 = 6,6 \text{ voltios}$$

Esta tensión puede no ser adecuada para el correcto funcionamiento del circuito.

Limitación en el valor de R_2

R_2 no puede hacerse arbitrariamente grande porque limitaría la velocidad de funcionamiento del circuito.

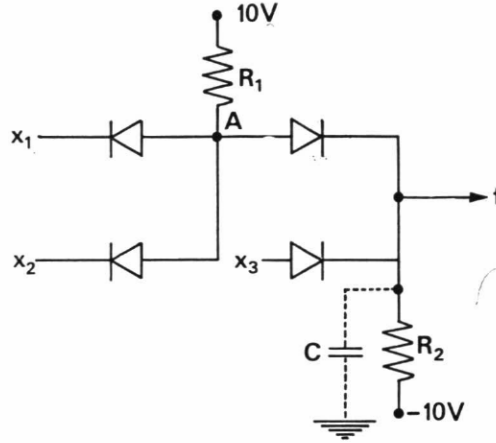


FIG. 6.8

C es una capacidad parásita que existe a la salida de la puerta, debida a conexiones, etc.

Si

$$\left. \begin{array}{l} x_1 = x_2 = 0 \text{ voltios} \\ x_3 = 0 \text{ voltios} \end{array} \right\} \begin{array}{l} V_A = V_{\text{diodo}} \\ y \\ V_s \simeq 0 \text{ voltios} \end{array}$$

Pero si el condensador C está cargado a + 10 voltios, los diodos de la puerta OR están polarizados en sentido inverso y la salida no caerá a 0 voltios hasta que el condensador C se descargue a través de R_2 .

La constante de tiempo $R_2 C$ determina la rapidez de la caída de la tensión de salida.

$$t = R_2 C \quad \text{para} \quad R_2 = 100 \text{ K}\Omega \quad \text{y} \quad C = 100 \text{ pF}$$

$$t = 100 \times 10^3 \times 100 \times 10^{-12} = 10 \cdot 10^{-6} = 10 \mu \text{ seg.}$$

Limitación en el valor de R_1

Si R_1 se hace pequeña en relación con R_2 , también puede mantenerse alto el nivel de la tensión de salida. Esta solución presenta el inconveniente de que al disminuir R_1 , aumenta el valor de la corriente de entrada en los diodos y, con ella, la potencia a disipar, lo cual no es deseable.

En este círculo, las entradas x_1 y x_2 son enviadas a una puerta *AND*; esto constituye un *nivel*. La salida de la puerta *AND* es enviada a una puerta *OR*. Esta puerta se considera que constituye otro nivel. El circuito de la figura 6.6 es un circuito de conmutación de dos niveles.

A causa de la interacción entre puertas con diodos, es necesario restaurar la salida del circuito.

Se presentan los siguientes problemas:

- 1) Retardos introducidos por las puertas. Este retardo es el intervalo de tiempo comprendido entre el instante en que las entradas son aplicadas y el instante en que las salidas son detectadas.
- 2) Degradación del valor de la señal (atenuación).
- 3) Problemas de coincidencias de impulsos.

6.4 OPTIMIZACION DE CIRCUITOS

No existe un criterio bien definido sobre optimización de circuitos de conmutación con diodos. Se deben considerar:

- coste del circuito;
- posible interacción entre puertas, y
- retardo desarrollado,

y alcanzar un compromiso.

El coste puede evaluarse en relación con el número de diodos que se utilicen. El retardo y la interacción están en función del número de niveles empleados.

En la práctica, se adopta el siguiente criterio: «El circuito puede tener, como máximo, dos niveles, y el número de diodos requerido debe ser el menor posible.»

Para lograr un circuito de conmutación de dos niveles, la función debe ser:

- a) Una combinación *OR* de productos.
- b) Una combinación *AND* de sumas.

El número de diodos requeridos puede calcularse mediante la suma de:

- a) Número de términos literales.
- b) Número de términos de que conste la función de conmutación dada.

EJEMPLO: Sea

$$f(w, x, y, z) = x' y' z' + x y' z + w' y z + w x z + w x' y + x' y z' \quad [1]$$

que requiere 24 diodos para ser instrumentada.

Simplificando resulta:

$$f(w, x, y, z) = x z + y z + x' z' \quad [2]$$

que solamente requiere 9 diodos para su realización.

6.5 PROBLEMA DE LA COINCIDENCIA DE IMPULSOS

Los valores verdaderos de las variables pueden representarse físicamente:

- Mediante niveles lógicos (V_+ y V_-). En este caso, la salida no se detecta hasta que se haya alcanzado un estado constante.
- Mediante impulsos de anchura finita.

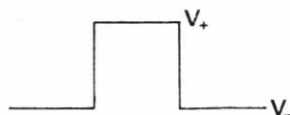


FIG. 6.9

Sea el circuito siguiente:

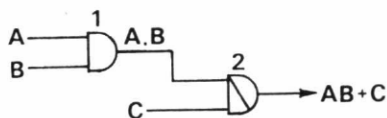


FIG. 6.10

Si las condiciones de entrada son $A = 1$, $B = 1$ y $C = 0$, y se usan impulsos de anchura finita, las entradas del circuito son las indicadas en la figura 6.11. En dicha figura se muestran también las salidas del circuito, en los casos de retardo nulo y retardo $t_1 - t_0$. Si las condiciones de entrada fueran $A = 0$, $B = 0$ y $C = 1$, el retardo sería $t'_1 - t_0$.

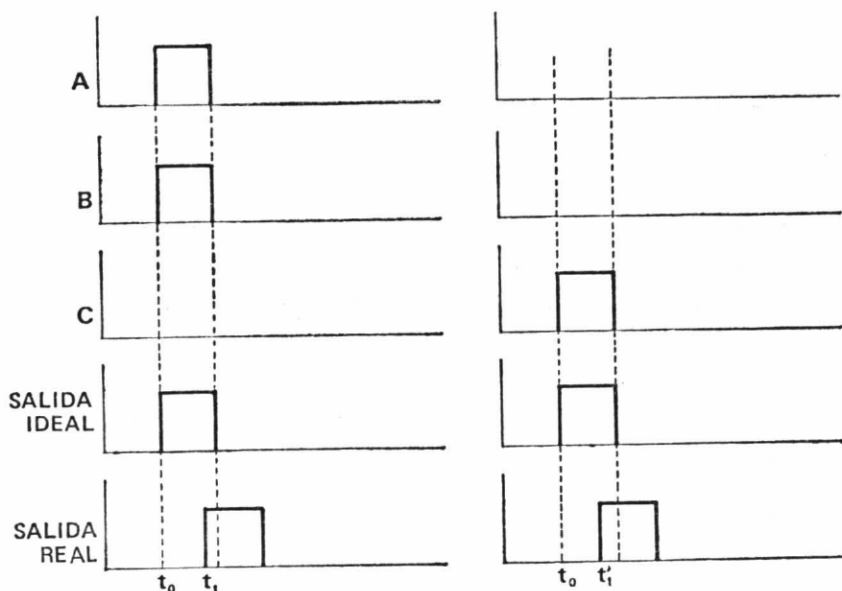


FIG. 6.11

$t_1 - t_0$ es el retardo introducido por la puerta *AND* y por la puerta *OR* (suma de ambos).

$t'_1 - t_0$ es el retardo introducido por la puerta *OR* solamente.

Se observa que $t'_1 - t_0 < t_1 - t_0$.

Se puede concluir diciendo que las salidas aparecen en diferentes instantes para diferentes condiciones de entrada.

Examinemos ahora el circuito de la figura 6.12.

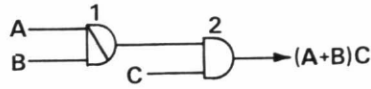
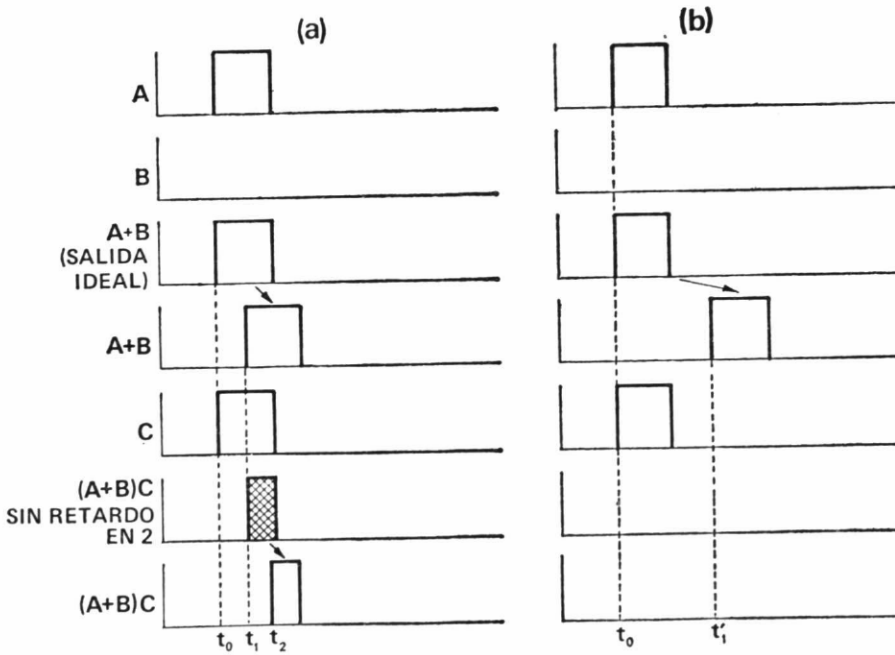


FIG. 6.12



$t_1 - t_0 =$ retardo de la puerta *OR*
 $t_2 - t_1 =$ retardo de la puerta *AND*

$t'_1 - t_0 =$ retardo de la puerta *OR*

FIG. 6.13

Si las condiciones de entrada son $A = 1$, $B = 0$, $C = 1$, y se emplean impulsos de anchura finita, las entradas y salidas del circuito son las indicadas en la figura 6.13. La salida de la puerta *OR* comienza a aparecer en t_1 , debido al retardo desarrollado en dicha puerta. Por otra parte, la entrada *C* aparece en t_0 .

Por esta causa, se reduce la anchura del impulso de salida, lo cual no es deseable.

Si el retardo producido por la puerta *OR* tuviera una duración superior a la anchura del impulso, el circuito puede generar una salida incorrecta. En la figura 6.13 (b), se produce un 0 a la salida, en vez de un 1.

Esta situación puede corregirse introduciendo una línea de retardo en la entrada *C*.

Por último, conviene tener presente que al emplear impulsos de anchura finita se deben considerar los problemas de coincidencia de impulsos.

CAPITULO 7
TRANSISTOR LOGICO

7.1 INTRODUCCION

Un transistor de unión es un cristal de silicio (o germanio) en el que una capa de silicio tipo n está colocada entre otras dos de tipo p (transistor PNP).

Otra posibilidad es que una capa tipo p esté entre otras dos tipo n (transistor NPN).

El conjunto formado es pequeño y se cierra herméticamente, para prevenirlo de los agentes exteriores, en una cápsula metálica o de plástico.

Las regiones se denominan: emisor, base y colector.

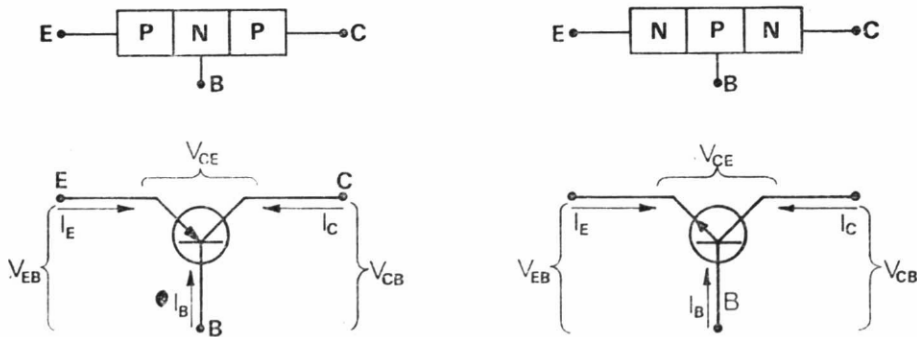


FIG. 7.1

El emisor y el colector son siempre del mismo tipo de semiconductor, y la base, de otro tipo.

La flecha en el terminal de emisor indica el sentido de circulación de corriente cuando la unión emisor-base se polariza directamente.

En la figura 7.1, las corrientes I_E , I_B y I_C se consideran positivas cuando entran en el transistor.

Existen tres posibilidades de montaje:

- a) Emisor común.
- b) Base común.
- c) Colector común.

En la configuración emisor común, distinguiremos tres regiones de funcionamiento:

1. *Zona activa.*—La unión emisor-base se polariza directamente, y la unión colector-base en sentido inverso. Cuando un transistor trabaja como dispositivo amplificador debe procurarse que trabaje en esta zona.

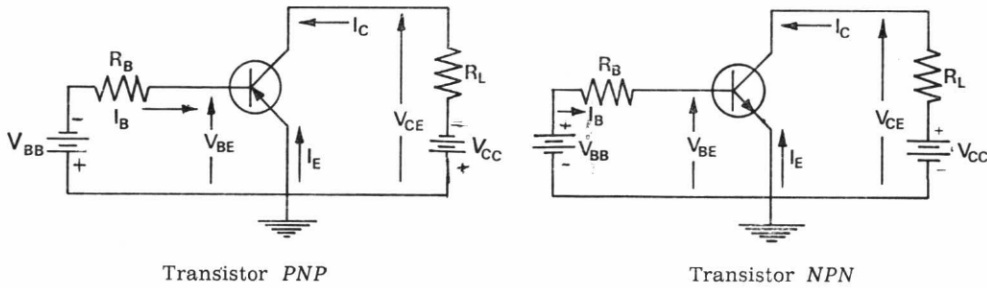


FIG. 7.2.—Polarizaciones. Zona activa. Emisor común

Para que el transistor conduzca, al colector y a la base se les da siempre una tensión del mismo signo con relación al emisor:

- a) Si es del tipo PNP, el colector y la base son negativos con relación al emisor.
- b) Si es del tipo NPN, el colector y la base son positivos con relación al emisor.

2. *Zona de saturación.*—Las uniones emisor-base y colector-base están directamente polarizadas.

Para que un transistor esté en saturación se ha de verificar:

$$I_B > \frac{(I_C)_{SAT}}{h_{FE}}$$

donde

I_B = corriente de base.

$(I_C)_{SAT}$ = corriente de colector en saturación.

h_{FE} = ganancia de corriente en continua. Varía entre 10 y 150, aproximadamente.

3. *Zona de corte.*—Las uniones emisor-base y colector-base están inversamente polarizadas.

El corte se define como la condición en la cual la corriente de colector es igual a la corriente inversa de saturación, y la corriente de emisor es igual a cero. El corte significa:

$$I_E = 0.$$

$$I_C = I_{C_0}.$$

$$I_B = -I_C = -I_{C_0}.$$

V_{BE} es una tensión inversa, cuya magnitud es del orden de 0,1 voltios para el transistor de germanio y de 0 voltios para el silicio.

Los parámetros que definen el funcionamiento del transistor son diferentes según el montaje utilizado.

A continuación se estudian las curvas características del transistor en montaje de emisor común.

Los símbolos utilizados son los siguientes:

a) *Régimen estático:*

V_{BE} = tensión colector-emisor.

V_{BE} = tensión base-emisor.

V_{BB} = tensión de polarización de la base

V_{CC} = tensión de polarización del colector.

I_C = corriente de colector.

I_B = corriente de base.

I_E = corriente de emisor.

b) *Régimen dinámico:*

v_{ce} = tensión colector emisor.

v_{be} = tensión base-emisor.

i_c = corriente de colector.

i_b = corriente de base.

i_e = corriente de emisor.

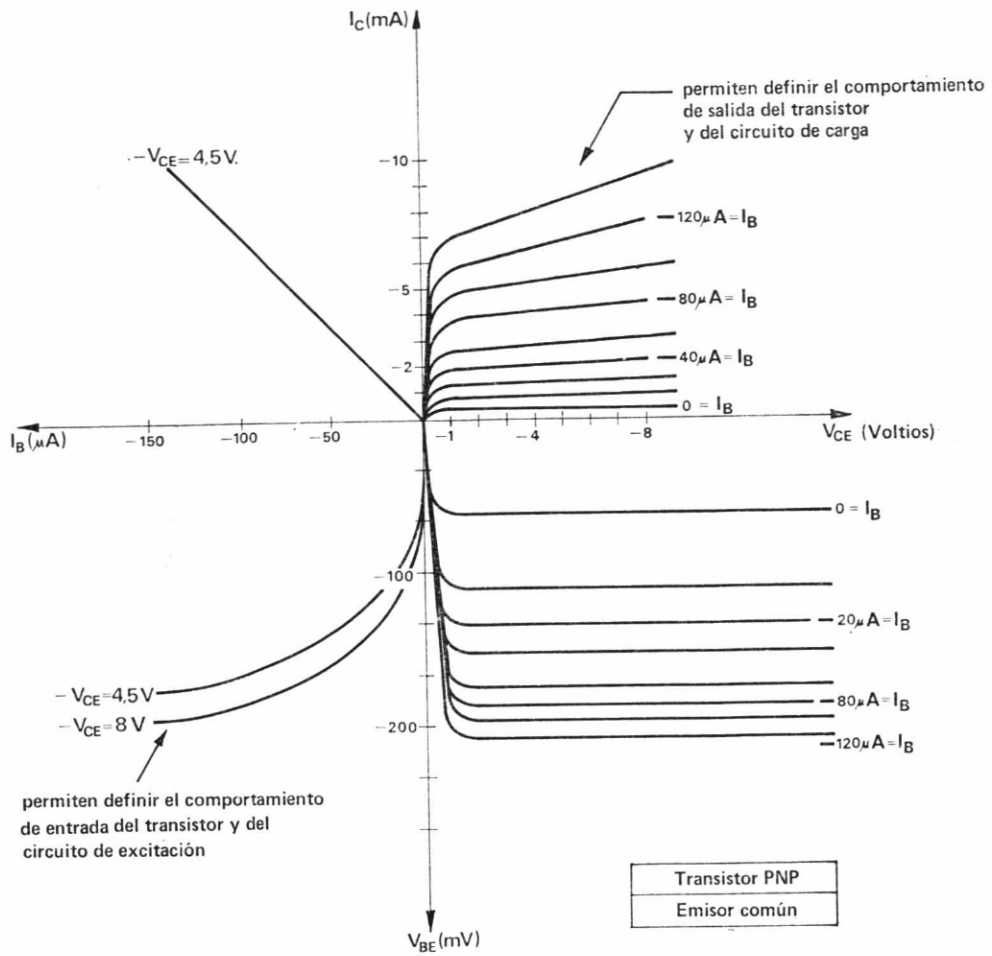


FIG. 7.3.—Ejemplo

RECTA DE CARGA

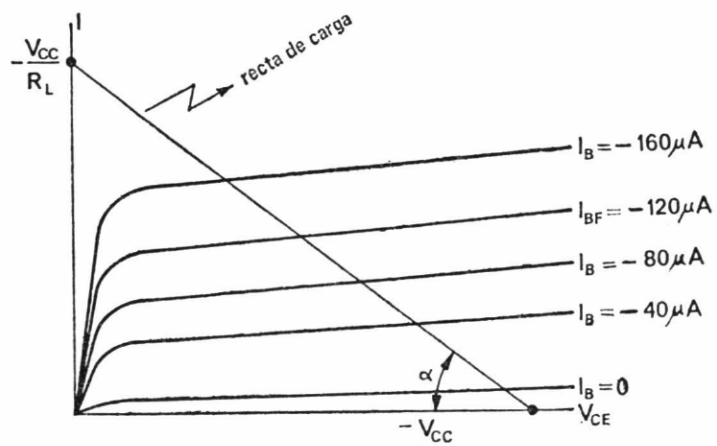
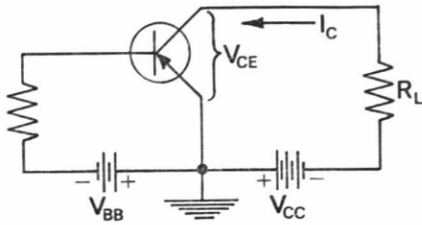


FIG. 7.4

Dado el montaje de la figura 7.5, se trata de determinar la recta de carga del transistor.



R_L : resistencia de carga.
 V_{CC} : número positivo que representa la magnitud de la tensión de alimentación.

FIG. 7.5

En la figura 7.4 se observa que

$$V_{CC} + I_C \cdot R_L + V_{CE} = 0$$

de donde

$$I_C = -\frac{V_{CE}}{R_L} - \frac{V_{CC}}{R_L} \quad (\text{ecuación de una recta})$$

Para

$$I_C = 0 \quad , \quad \boxed{V_{CE} = -V_{CC}}$$

Para

$$V_{CE} = 0 \quad , \quad \boxed{I_C = -\frac{V_{CC}}{R_L}}$$

El ángulo que forma la recta de carga con el eje horizontal tiene la propiedad de que su tangente es la inversa de la resistencia de carga.

$$\text{tang } \alpha = \frac{V_{CC}/R_L}{V_{CC}} = \frac{1}{R_L}$$

Si V_{CE} se expresa en voltios e I_C en miliamperios, R_L viene dado en kiloohmios.

ESTUDIO DE LOS PARÁMETROS DE UN TRANSISTOR

Sea el cuadripolo de la figura 7.6.

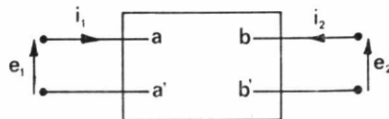


FIG. 7.6

En este cuadripolo se pueden definir cuatro magnitudes eléctricas y, entre ellas, varias relaciones.

Si se tomaran otras bornas para la entrada y la salida, se obtendrían otras dos ecuaciones.

Las relaciones que pueden establecerse son:

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} \quad [1]$$

$$\begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \quad [2]$$

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} \quad [3]$$

$$\begin{bmatrix} i_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ i_2 \end{bmatrix} \quad [4]$$

Desarrollando:

$$\left. \begin{aligned} e_1 &= z_{11} i_1 + z_{12} i_2 \\ e_2 &= z_{21} i_1 + z_{22} i_2 \end{aligned} \right\} \quad [5]$$

$$\left. \begin{aligned} e_1 &= h_{11} i_1 + h_{12} e_2 \\ i_2 &= h_{21} i_1 + h_{22} e_2 \end{aligned} \right\} \quad [7]$$

$$\left. \begin{aligned} i_1 &= y_{11} e_1 + y_{12} e_2 \\ i_2 &= y_{21} e_1 + y_{22} e_2 \end{aligned} \right\} \quad [6]$$

$$\left. \begin{aligned} i_1 &= g_{11} e_1 + g_{12} i_2 \\ e_2 &= g_{21} e_1 + g_{22} i_2 \end{aligned} \right\} \quad [8]$$

En algunos libros, los subíndices se sustituyen por letras: 11 por *i*, 12 por *r*, 21 por *f* y 22 por *o*.

$$\begin{aligned} \Delta z &= z_{11} z_{22} - z_{12} z_{21} \\ \Delta h &= h_{11} h_{22} - h_{12} h_{21} \end{aligned}$$

$$\begin{aligned} \Delta y &= y_{11} y_{22} - y_{12} y_{21} \\ \Delta g &= g_{11} g_{22} - g_{12} g_{21} \end{aligned}$$

Se tiene:

$$[\Delta z]^{-1} = \frac{1}{\Delta z} \cdot \begin{bmatrix} z_{22} & -z_{12} \\ -z_{21} & z_{11} \end{bmatrix} = [\Delta y]$$

con lo cual

$$y_{11} = \frac{z_{22}}{\Delta z}$$

$$y_{12} = \frac{-z_{12}}{\Delta z}$$

$$y_{21} = \frac{-z_{21}}{\Delta z}$$

$$y_{22} = \frac{z_{11}}{\Delta z}$$

A continuación se analizan cada uno de los parámetros.

a) Parámetros z :

$$z_{11} = \left(\frac{e_1}{i_1} \right)_{i_2=0}$$

Impedancia de entrada para la salida en circuito abierto.

$$z_{12} = \left(\frac{e_1}{i_2} \right)_{i_1=0}$$

Impedancia de transferencia para la entrada en circuito abierto.

$$z_{21} = \left(\frac{e_2}{i_1} \right)_{i_2=0}$$

Impedancia de transferencia para la salida en circuito abierto.

$$z_{22} = \left(\frac{e_2}{i_2} \right)_{i_1=0}$$

Impedancia de salida para la entrada en circuito abierto.

b) Parámetros y :

$$y_{11} = \left(\frac{i_1}{e_1} \right)_{e_2=0}$$

Admitancia de entrada para la salida en cortocircuito.

$$y_{12} = \left(\frac{i_1}{e_2} \right)_{e_1=0}$$

Admitancia de transferencia para la entrada en cortocircuito.

$$y_{21} = \left(\frac{i_2}{e_1} \right)_{e_2=0}$$

Admitancia de transferencia para la entrada en cortocircuito.

$$y_{22} = \left(\frac{i_2}{e_2} \right)_{e_1=0}$$

Admitancia de salida para la entrada en cortocircuito.

c) Parámetros h :

$$h_{11} = \left(\frac{e_1}{i_1} \right)_{e_2=0}$$

Impedancia de entrada para la salida en cortocircuito.

$$h_{12} = \left(\frac{e_1}{e_2} \right)_{i_1=0}$$

Relación entre las tensiones de entrada y salida, para la entrada en circuito abierto.

$$h_{21} = \left(\frac{i_2}{i_1} \right)_{e_2=0}$$

Ganancia de corriente, con la salida en cortocircuito.

$$h_{22} = \left(\frac{i_2}{e_2} \right)_{i_1=0}$$

Admitancia de salida para la entrada en circuito abierto.

d) Parámetros g :

$$g_{11} = \left(\frac{i_1}{e_1} \right)_{i_2=0}$$

Admitancia de entrada para la salida en circuito abierto.

$$g_{12} = \left(\frac{i_1}{i_2} \right)_{e_1=0}$$

Relación entre las corrientes de entrada y salida, con la entrada en cortocircuito.

$$g_{21} = \left(\frac{e_2}{e_1} \right)_{i_2=0}$$

Ganancia de tensión, con la salida en circuito abierto.

$$g_{22} = \left(\frac{e_2}{i_2} \right)_{e_1=0}$$

Impedancia de salida para la entrada en cortocircuito.

De las ecuaciones [5], [6], [7] y [8] se deducen los circuitos equivalentes siguientes:

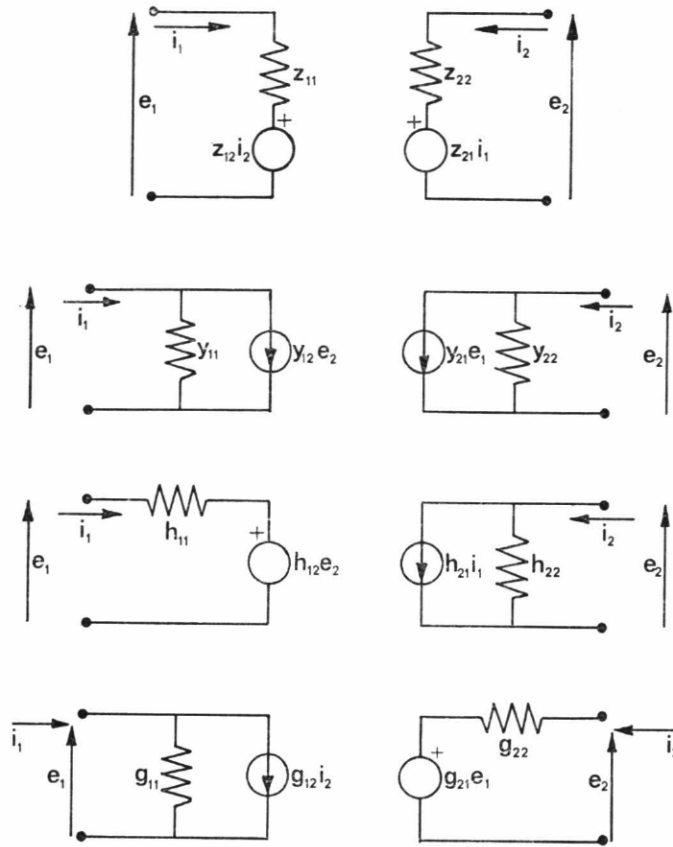


FIG. 7.7

Un circuito muy utilizado es el siguiente:

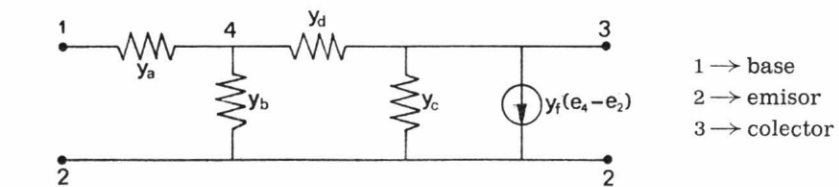


FIG. 7.8

7.2 TRANSISTOR LOGICO

Los transistores empleados en los circuitos lógicos han de representar los estados 0 y 1 de forma que resulten perfectamente diferenciables. Las condiciones idóneas para conseguirlo se logran cuando el transistor está totalmente bloqueado y cuando está en estado de conducción total.

Conviene tener presente:

- a) Que cuando el transistor está en estado de conducción total, existe una caída de tensión entre los terminales colector-emisor. Esta tensión se representa por $(V_{CE})_{SAT}$ y es del orden de unas décimas de voltio.
 - b) Que cuando el transistor está en estado de bloqueo o corte, existe una corriente minoritaria I_{CO} , llamada corriente inversa de saturación, que varía entre varios nanoamperios y varios microamperios.
- La caída de tensión $I_{CO} \cdot R_L$, originada por esta corriente en la resistencia de carga R_L , suele ser despreciable (*).

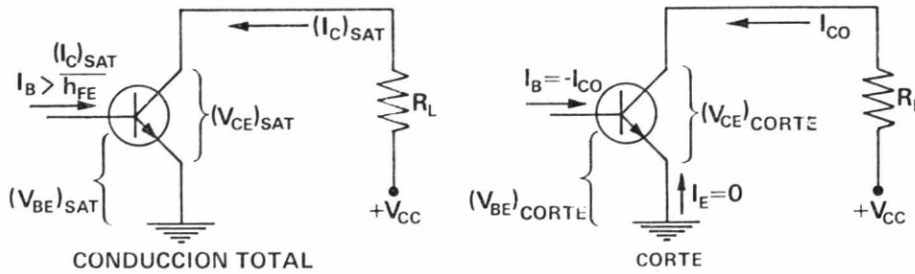


FIG. 7.9

Frente a estos inconvenientes, el transistor presenta notables ventajas:

- 1) Larga vida (miles de millones de operaciones).
- 2) Breve tiempo de conmutación, inferior a un microsegundo.
- 3) La señal aplicada al circuito base-emisor se obtiene invertida y amplificada en el circuito colector-emisor (montaje emisor común).

El funcionamiento básico y las desviaciones de un transistor respecto a un modelo ideal se pueden estudiar en las curvas características que relacionan I_C con V_{CE} .

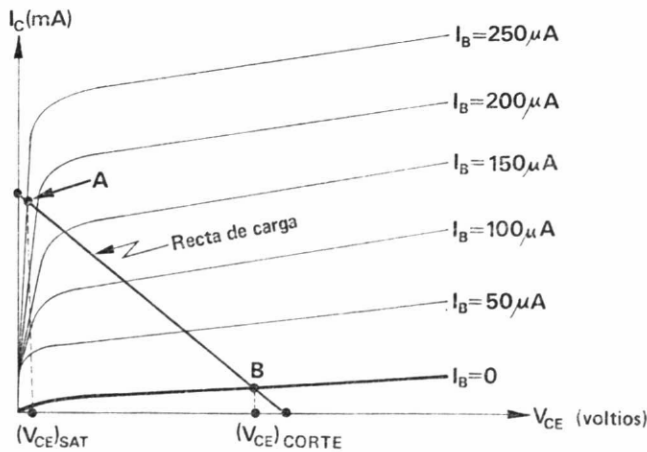


FIG. 7.10.—Transistor NPN

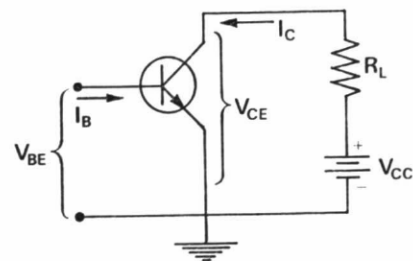


FIG. 7.11

(*) Para un transistor PNP, I_{CO} es negativa, y para un transistor NPN, positiva.

La conexión de una resistencia de carga R_L define las zonas de funcionamiento del transistor.

1) *Zona de saturación.*—Cuando el transistor está en estado de conducción total, la corriente I_C aumenta hasta un valor de saturación. En estas condiciones, la tensión colector-emisor se llama $(V_{CE})_{SAT}$.

El punto de funcionamiento, intersección de la recta de carga con la curva característica del transistor, es el punto A.

2) *Zona de corte.*—La condición de corte corresponde aproximadamente a $I_B = 0$. El punto B de intersección de la recta de carga con la curva correspondiente a la condición de corte, muestra que la tensión de corte es más baja que la tensión de alimentación:

$$(V_{CE})_{CORTE} \simeq V_{CC} - I_{C0} R_L$$

3) *Zona activa.*—Situada entre las zonas de corte y saturación.

La diferencia entre $(V_{CE})_{CORTE}$ y $(V_{CE})_{SAT}$ es perfectamente distinguible. Generalmente nos referimos a la tensión de saturación como si fuera el valor de 0 voltios y la tensión de corte como si fuese el valor V_{CC} de alimentación (*).

Al conmutar de un estado a otro, el transistor atraviesa la región lineal de la característica. En esta zona, el tiempo de transición es tan pequeño que cuando el transistor actúa como conmutador se dice que sólo actúa en las regiones de saturación o corte. El tiempo de conmutación suele fluctuar entre algunos microsegundos y algunos nanosegundos.

7.3 TRANSISTOR INVERSOR

En la figura 7.12 se muestra un circuito inversor.

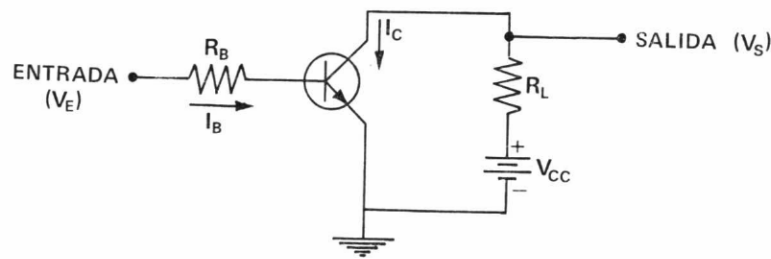


FIG. 7.12

(*) La familia lógica ECL utiliza como zonas de trabajo la zona de corte y la zona activa. No trabaja en la zona de saturación. (ECL: *Emitter-coupled logic*).

La resistencia de entrada se elige de forma que establezca un valor para la corriente de base que sea suficiente para colocar el transistor en estado de saturación.

R_L no debe ser demasiado elevada para evitar una caída de tensión $R_L \cdot I_{C0}$ que sea apreciable. Es del orden de varios $K \Omega$.

Cuando la entrada está a 0 voltios, la unión emisor-base está polarizada en sentido inverso y la corriente de base I_B vale cero. En este caso, la tensión de salida es:

$$V_S = V_{CC} - I_{C0} R_L = (V_{CE})_{\text{CORTE}} \simeq V_{CC}$$

Cuando la entrada está a $+V_E$ voltios, el transistor se excita al estado de conducción. Cuando se alcanza el estado de conducción total la caída de tensión entre los terminales de la unión emisor-base, polarizada en sentido directo, es: $(V_{BE})_{\text{SAT}} \simeq 0,7$ voltios. Esta tensión varía ligeramente cuando se producen cambios notables en el valor de I_B :

$$V_E = I_B \cdot R_B + (V_{BE})_{\text{SAT}}$$

$$I_B = \frac{V_E - (V_{BE})_{\text{SAT}}}{R_B}$$

La tensión de salida es $V_S = (V_{CE})_{\text{SAT}} \simeq 0$ voltios.

*

Para valores típicos de

$$V_E = 10 \text{ voltios} \qquad V_{CC} = 10 \text{ voltios}$$

$$R_B = 10 \text{ K } \Omega$$

$$R_L = 2 \text{ K } \Omega$$

$$(V_{CE})_{\text{SAT}} = 0,3 \text{ voltios}$$

$$(V_{BE})_{\text{SAT}} = 0,5 \text{ voltios}$$

se obtiene:

$$I_B = \frac{10 - 0,5}{10 \times 10^3} = 0,95 \text{ mA}$$

$$(I_C)_{\text{SAT}} = \frac{V_{CC} - (V_{CE})_{\text{SAT}}}{R_L} = \frac{10 - 0,3}{2 \times 10^3} = 4,85 \text{ mA} \text{ (*)}$$

*

(*) El valor máximo de la corriente de colector en saturación es:

$$(I_C)_{\text{SAT}} \simeq \frac{V_{CC}}{R_L}$$

Las tablas de tensiones y de verdad son las siguientes:

V_E	V_S	V_E	V_S	V_E	V_S
0	+ 10	0	1	1	0
+ 10	0	1	0	0	1
Tabla de tensiones		Tabla de verdad Lógica positiva		Tabla de verdad Lógica negativa	

7.4 CALCULO DE UN INVERSOR

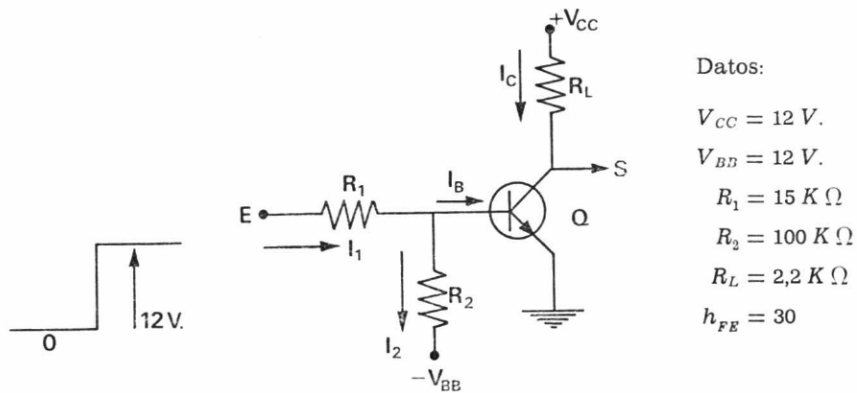


FIG. 7.13

Los valores típicos de un transistor *NPN*, de silicio, a $25^\circ C$ son:

$$\frac{(V_{CE})_{SAT}}{0,3 V} \quad \frac{(V_{BE})_{CORTE}}{0,0 V} \quad \frac{(V_{BE})_{SAT}}{0,7 V}$$

Con una polarización de 0 voltios queda cortada la unión emisor-base del transistor de silicio. Para esta tensión, *Q* estará cortado.

1.º Comprobación de que el transistor se satura para una tensión de entrada de 12 voltios:

$$(I_C)_{SAT} = \frac{V_{CC} - (V_{CE})_{SAT}}{R_L} = \frac{12 - 0,3}{2,2} = 5,31 \text{ mA}$$

Dado que

$$h_{FE} = \left(\frac{I_C}{I_B} \right) v_s = 0$$

aplicando los valores $h_{FE} = 30$ e $I_C = 5,31 \text{ mA}$, se obtiene el valor mínimo de I_B necesario para saturar el transistor *Q*:

$$(I_B)_{min} = \frac{5,31}{30} = 0,18 \text{ mA}$$

A partir del circuito de la figura 7.13 se pueden calcular los valores de I_1 , I_2 e I_B .

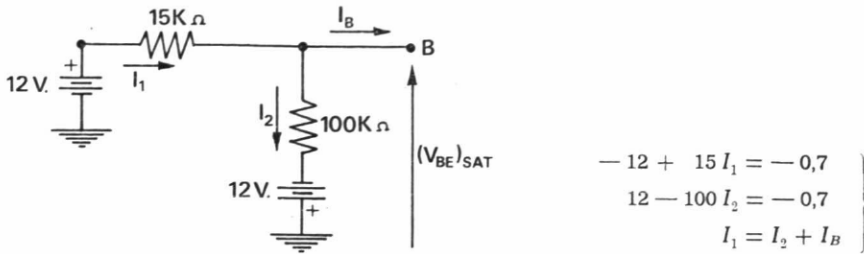


FIG. 7.14

$$I_1 = \frac{12 - 0,7}{15} = 0,75 \text{ m A} \quad ; \quad I_2 = \frac{12 + 0,7}{100} = 0,13 \text{ m A} \quad ; \quad I_B = 0,75 - 0,13 = 0,62 \text{ m A}$$

Como $I_B > (I_B)_{\min}$, es evidente que Q está en saturación (véase la nota 1).

2.º Comprobación de que el transistor se corta para una tensión de entrada de 0 voltios.

De la tabla de valores se observa que para una entrada de 0 voltios el transistor está al corte (véase la nota 2).

3.º Rehaciendo los cálculos para $(V_{CE})_{\text{SAT}} = (V_{BE})_{\text{SAT}} = 0$ se obtienen los siguientes valores:

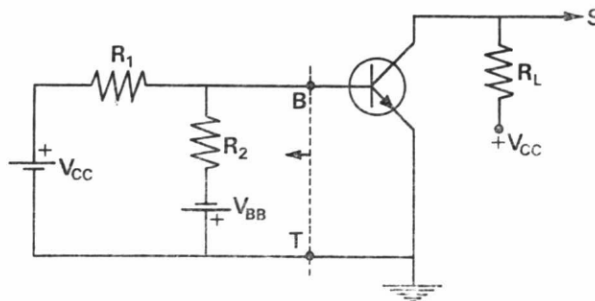
$$\begin{aligned} (I_C)_{\text{SAT}} &= 5,45 \text{ m A} \\ (I_B)_{\min} &= 0,18 \text{ m A} \\ I_1 &= 0,80 \text{ m A} \\ I_2 &= 0,12 \text{ m A} \\ I_B &= 0,68 \text{ m A} \end{aligned}$$

que no difieren grandemente de los hallados.

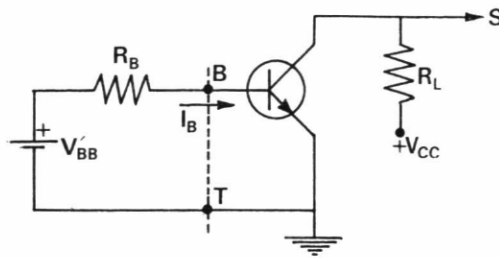
4.º Se añade un condensador C en paralelo con R_1 para mejorar la respuesta del transistor en régimen transitorio. Es del orden de 100 pF.

Un estudio detallado de la influencia de la carga de los portadores minoritarios, tiempos de subida y caída y retardo de propagación, puede verse en el capítulo 20 del libro *Circuitos de pulsos, digitales y de conmutación*, por Millman-Taub (Ediciones del Castillo, 1969).

NOTA 1.—El valor de I_B también puede calcularse de la siguiente forma:



Aplicando el teorema de Thevenin entre los puntos B y T , se obtiene:



$$R_B = \frac{R_1 R_2}{R_1 + R_2}$$

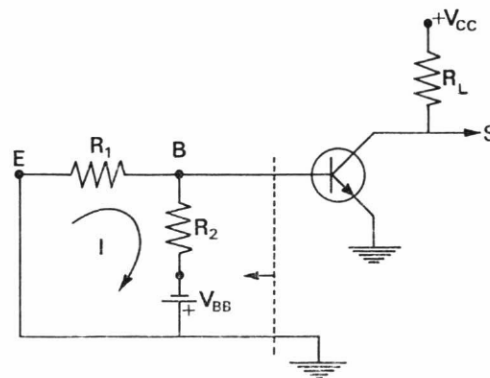
$$V_{BB} = V_{CC} - \frac{V_{CC} + V_{BB}}{R_1 + R_2} \cdot R_1 \quad \boxed{I_B = \frac{V_{BB} - V_{BE}}{R_B}}$$

Comprobación:

$$I_B = \frac{V_{CC} - \frac{V_{CC} + V_{BB}}{R_1 + R_2} R_1 - V_{BE}}{\frac{R_1 R_2}{R_1 + R_2}} = \frac{V_{CC} R_1 + V_{CC} R_2 - V_{CC} R_1 - V_{BB} R_1 - (R_1 + R_2) V_{BE}}{R_1 R_2}$$

$$I_B = \frac{(V_{CC} - V_{BE}) R_2}{R_1 \cdot R_2} - \frac{(V_{BB} + V_{BE}) R_1}{R_1 R_2} = \frac{V_{CC} - V_{BE}}{R_1} - \frac{V_{BB} + V_{BE}}{R_2} = I_1 - I_2 \quad \text{c. s. q. d.}$$

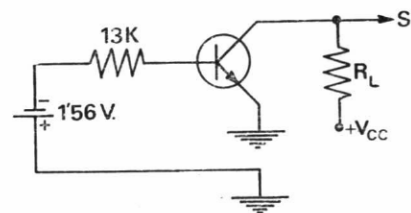
NOTA 2.—Comprobación de que el transistor se corta para una tensión de entrada de 0 voltios:



Aplicando el teorema de Thevenin, se obtiene:

$$R_B = \frac{R_1 R_2}{R_1 + R_2} = \frac{15 \times 100}{100 + 15} \approx 13 \text{ K } \Omega$$

$$V_{BB} = \frac{-V_{BB}}{R_1 + R_2} R_1 = -\frac{12}{100 + 15} \cdot 15 = -1,56$$



lo que quiere decir que la unión emisor-base está polarizada en sentido inverso.

7.5 PUERTAS DIODO-TRANSISTOR (DTL) *

Hasta ahora se han considerado separadamente las puertas *AND* y *OR* de diodos y el inversor transistorizado. Estos dos tipos pueden combinarse para formar una puerta compuesta.

Al emplear puertas con diodos, hay que tener presente:

- a) Que existe una degradación de la señal.
- b) Que no pueden realizar la operación *NOT*.
- c) Que el número de puertas excitadas a partir de una puerta sencilla está limitado en la práctica.

Para realizar la amplificación o adaptación de cargas, se requiere el empleo de inversores transistorizados para cada una de las puertas con diodos. La puerta básica puede lograrse añadiendo un inversor después de una puerta lógica.

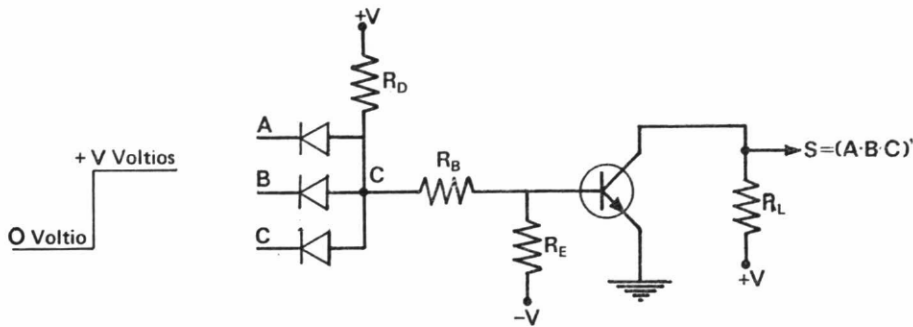


FIG. 7.15.—Puerta *NAND* con lógica positiva (o puerta *NOR* con lógica negativa)

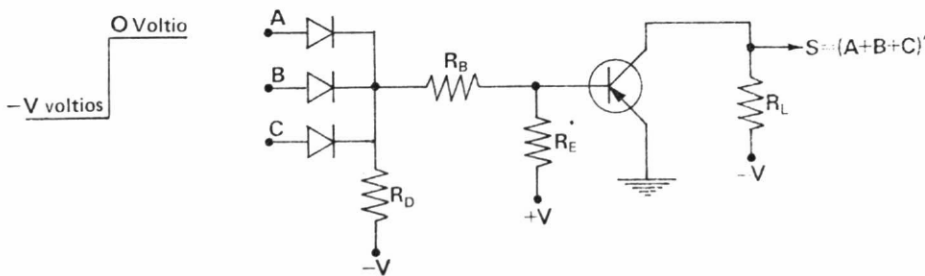


FIG. 7.16.—Puerta *NOR* con lógica positiva (o puerta *NAND* con lógica negativa)

A continuación se realiza un análisis del funcionamiento de la puerta *NAND* con lógica positiva, para un circuito de dos entradas e inversor *NPN*.

* DTL: diode transistor logic.

Este análisis se puede desdoblar en dos casos:

1) Transistor en estado de *conducción*.

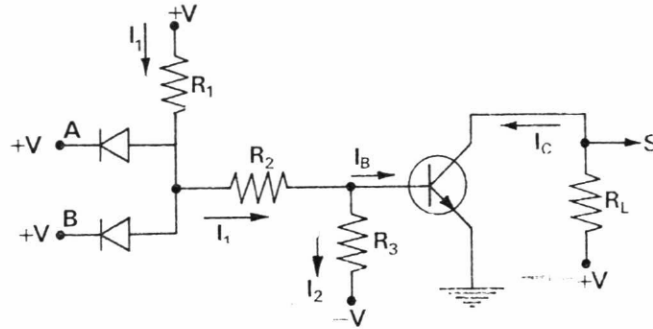


FIG. 7.17

El transistor *NPN* está en estado de conducción total; por lo cual $V_{CE} = (V_{CE})_{SAT}$. En este caso, las entradas *A* y *B* de la puerta *AND* deben ser $+V$ voltios, con lo cual los dos diodos están inversamente polarizados.

Se verifica:

$$I_1 = I_2 + I_B$$

Para calcular I_2 e I_B se recurre al circuito de la figura 7.18.

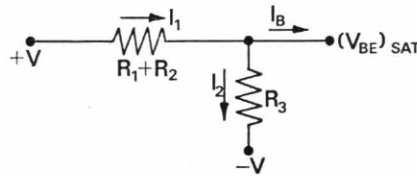


FIG. 7.18

$$\left. \begin{aligned} V - I_1(R_1 + R_2) &= (V_{BE})_{SAT} \\ -V + I_2 R_3 &= (V_{BE})_{SAT} \end{aligned} \right\} \begin{aligned} I_1 &= \frac{V - (V_{BE})_{SAT}}{R_1 + R_2} \\ I_2 &= \frac{V + (V_{BE})_{SAT}}{R_3} \end{aligned}$$

$$I_B = I_1 - I_2$$

Además

$$I_C = \frac{V - (V_{CE})_{SAT}}{R_L}$$

En condiciones de saturación se verifica:

$$h_{FE} = h_{21E} = \left(\frac{i_2}{i_1} \right)_{e_2=0} > \left(\frac{I_C}{I_B} \right)_{SAT}$$

es decir, $h_{FE} > \left(\frac{I_C}{I_B} \right)_{SAT}$ [1]

Si la condición [1] se cumple, el transistor se saturará adecuadamente cuando pase al estado de conducción. Hay que tener presente que [1] puede no cumplirse:

- a) Por haber elegido un transistor con un valor menor de h_{FE} que el necesario.
- b) Por haber aumentado las necesidades de carga extrayendo más I_C .
- c) Por disminuir la temperatura.

El parámetro h_{FE} depende de la temperatura y decrece al disminuir ésta.

2) Transistor en estado de corte.

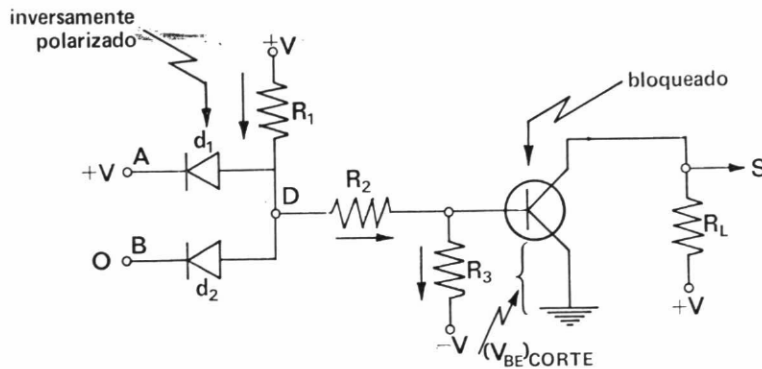


FIG. 7.19

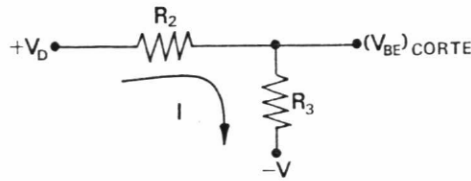
El transistor está en estado de corte, por lo cual

$$V_{BE} = (V_{BE})_{CORTE}$$

En este caso, las entradas de la puerta AND son $+V$ voltios y 0 voltios, con lo cual el diodo d_1 está polarizado en sentido inverso. La tensión en el punto D es:

$$V_D \simeq 0 \text{ voltios}$$

¿Cuál es la tensión en la base del transistor? Sea la figura 7.18:



ya que

$$\left. \begin{aligned} I_B = 0 \quad , \quad I &= \frac{V + V_D}{R_2 + R_3} \\ (V_{BE})_{CORTE} &= V_D - \frac{V + V_D}{R_2 + R_3} \cdot R_2 \end{aligned} \right\}$$

Admitiendo que $V_D \simeq 0$ resulta:

$$(V_{BE})_{CORTE} = -\frac{R_2}{R_2 + R_3} \cdot V$$

Para valores típicos de

$$\begin{array}{ll} R_1 = 10 \text{ K } \Omega & V = 10 \text{ V} \\ R_2 = 10 \text{ K } \Omega & (V_{CE})_{SAT} = 0,2 \text{ V} \\ R_3 = 100 \text{ K } \Omega & (V_{BE})_{SAT} = 0,5 \text{ V} \\ R_L = 10 \text{ K } \Omega & h_{FE} = 20 \end{array}$$

resulta:

$$(V_{BE})_{CORTE} = -\frac{10}{10 + 100} \cdot 10 = -0,91 \text{ voltios}$$

lo que quiere decir que la unión emisor-base está polarizada en sentido inverso en casi un voltio, lo que garantiza que el transistor está en el estado de *corte*.

¿Se mantiene bloqueado el transistor si se presenta un impulso de ruido de determinada amplitud?

El transistor pasará al estado de conducción cuando $V_{BE} = + 0,5$ voltios.

La tensión de ruido que no afectará al circuito es:

$$V_r < | (V_{BE})_{CORTE} | + (V_{BE})_{SAT} = 0,91 + 0,5 = 1,41 \text{ voltios}$$

¿Cuál es la caída de tensión en R_L ?

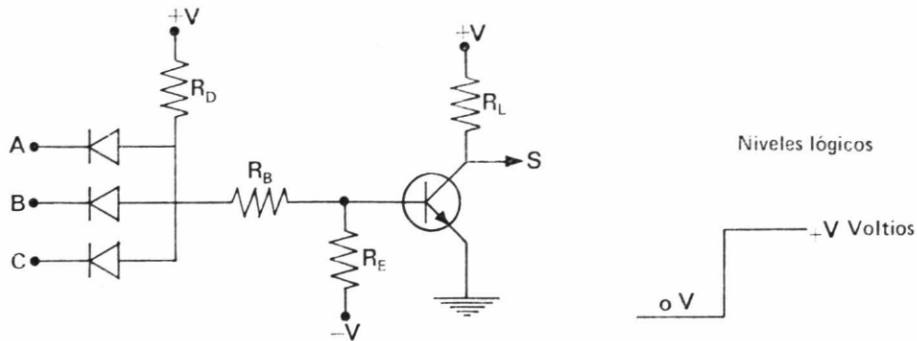
Para los valores típicos de $R_L = 10 \text{ K } \Omega$ e $I_{C0} = 1 \mu \text{ A}$, se tiene:

$$V_{R_L} = I_{C0} \cdot R_L = 10^{-6} \times 10 \times 10^3 = 10^{-2} = 10 \text{ m V}$$

valor despreciable frente a $V = 10$ voltios.

LOGICA POSITIVA

PUERTA «NAND»



PUERTA «NOR»

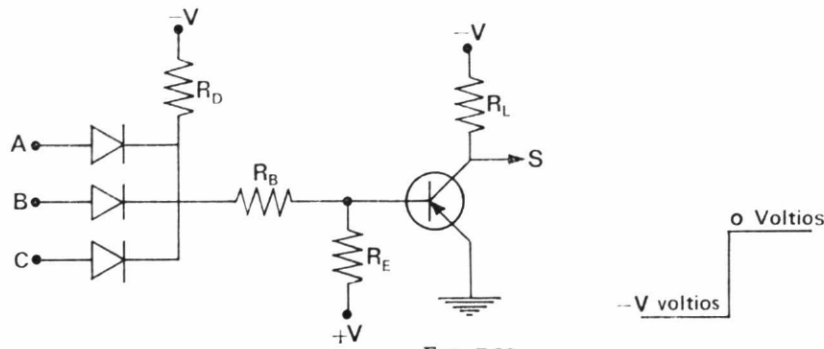


FIG. 7.20

Las puertas *NAND* y *NOR* con diodos no son compatibles:

- Emplean distintos niveles de tensión en sus entradas.
- Utilizan transistores opuestos.

7.6 PUERTAS *AND* Y *OR* (SEGUIDOR DE EMISOR)

Sea el circuito de la figura 7.21.

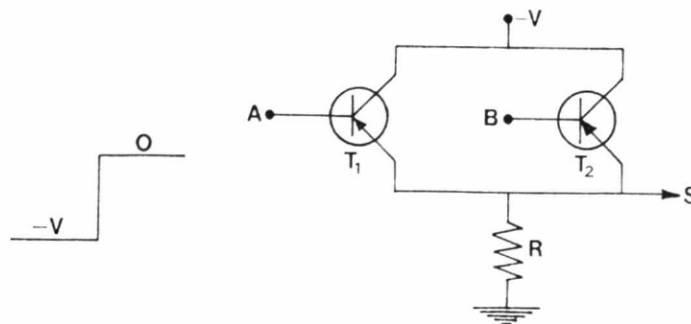


FIG. 7.21

Si A y B están al potencial de masa, los dos transistores están virtualmente en el corte, y S está, también, al potencial de masa (se puede demostrar que con la unión emisor-base en cortocircuito el transistor está virtualmente en el corte):

$$V_s = 0 \text{ voltios}$$

Si A y B son potenciales negativos, los dos transistores se encuentran en estado de conducción, y el potencial en S es

$$V_s = -V + V_{BE} \simeq -V$$

ya que V_{BE} es negativa y próxima a cero.

Si A y B son, respectivamente, 0 voltios y $-V$ voltios, T_1 está cortado y T_2 en estado de conducción; en cuyo caso:

$$V_s \simeq -V + V_{BE} \simeq -V$$

La tabla de funcionamiento del circuito sería:

A	B	S
0	0	0
0	$-V$	$-V$
$-V$	0	$-V$
$-V$	$-V$	$-V$

Empleando lógica positiva, se obtiene la siguiente tabla de verdad:

A	B	S	
1	1	1	$0V \equiv 1$
1	0	0	$-V \equiv 0$
0	1	0	
0	0	0	

de donde $S = A \cdot B$ Se trata de una función AND.

Empleando lógica negativa, se obtiene:

A	B	S	
0	0	0	$0V \equiv 0$
0	1	1	$-V \equiv 1$
1	0	1	
1	1	1	

de donde $S = A + B$ Se trata de una puerta OR.

Realizando el montaje de la figura 7.22, con transistores NPN, se obtienen los resultados inversos de los anteriores.

- a) Con lógica positiva: $S = A + B$ OR.
- b) Con lógica negativa: $S = A \cdot B$ AND.

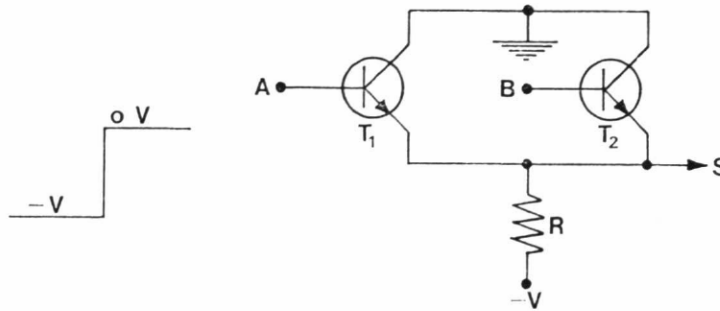


FIG. 7.22

Si A y B están al potencial de masa, los dos transistores están en estado de conducción, y el potencial en S es

$$V_S = -V_{BE} \approx 0 \text{ voltios}$$

Si A y B son potenciales $-V$ voltios, los dos transistores están virtualmente en el corte, y S está, también, al potencial $-V$.

$$V_S = -V$$

Si A y B son, respectivamente, 0 voltios y $-V$ voltios, T_1 está en estado de conducción y T_2 en el corte, en cuyo caso

$$V_S = -V_{BE} \approx 0 \text{ voltios}$$

La tabla de funcionamiento del circuito sería:

A	B	S
0	0	0
0	$-V$	0
$-V$	0	0
$-V$	$-V$	$-V$

Empleando lógica positiva, se obtiene la siguiente tabla de verdad:

A	B	S
1	1	1
1	0	1
0	1	1
0	0	0

de donde $S = A + B$ Puerta OR.
 $0 V \equiv 1$
 $-V \equiv 0$

Empleando lógica negativa, se obtiene:

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

de donde $S = A \cdot B$ Puerta AND.
 $0 V \equiv 0$
 $-V \equiv 1$

NOTA.—Las tensiones en las uniones de un transistor típico NPN a 25° C son:

	$(V_{CE})_{SAT}$	$(V_{BE})_{SAT}$	$(V_{BE})_{ACT}$	$(V_{BE})_{UMBRAL}$	$(V_{BE})_{CORTE}$
Silicio	0,3 V	0,7 V	0,6 V	0,5 V	0,0 V
Germanio	0,1 V	0,3 V	0,2 V	0,1 V	-0,1 V

Para un transistor PNP deben cambiarse de signo todos los valores.

7.7 TRANSISTORES DIRECTAMENTE ACOPLADOS (DCTL) *

Este tipo de lógica emplea el transistor como elemento de entrada.
 Sea el circuito de la figura 7.23.

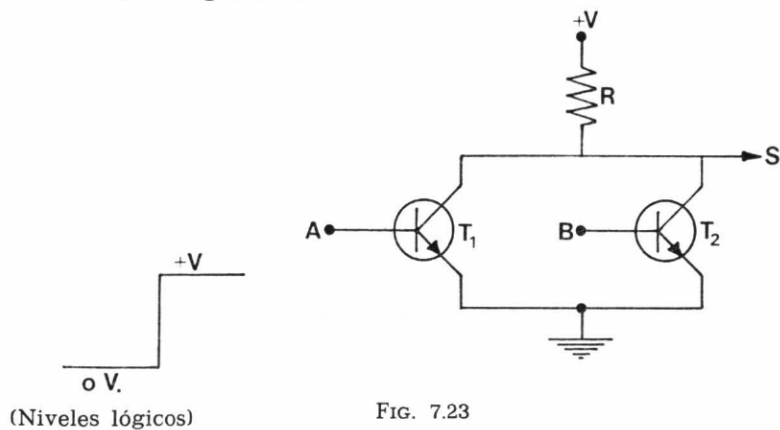


FIG. 7.23

* DCTL: *direct coupled transistor logic*.

Cuando A y B son potenciales de 0 voltios, los transistores T_1 y T_2 están en estado de corte. En este caso, la salida en S es

$$V_S = +V$$

Si A y B son potenciales de $+V$ voltios, los dos transistores T_1 y T_2 están en estado de saturación, en cuyo caso

$$V_S = (V_{CE})_{SAT} \simeq 0 \text{ voltios}$$

Si A y B son, respectivamente, 0 voltios y $+V$ voltios, el transistor T_1 se encuentra en estado de corte y T_2 en estado de saturación. La tensión de salida es

$$V_S = (V_{CE})_{SAT} \simeq 0 \text{ voltios}$$

Las tablas de tensiones y verdad son las siguientes:

A	B	S
0 voltios	0 voltios	$+V$ voltios
0 voltios	$+V$ voltios	0 voltios
$+V$ voltios	0 voltios	0 voltios
$+V$ voltios	$+V$ voltios	0 voltios

A	B	S	
0	0	1	$0V \equiv 0$
0	1	0	$+V \equiv 1$
1	0	0	Lógica positiva
1	1	0	positiva

A	B	S	
1	1	0	$+V \equiv 0$
1	0	1	$0V \equiv 1$
0	1	1	Lógica negativa
0	0	1	negativa

$$S = A' B' = (A + B)'$$

Puerta NOR con lógica positiva.

$$S = (A B)'$$

Puerta NAND con lógica negativa.

Otro circuito que puede estudiarse se muestra en la figura 7.24.

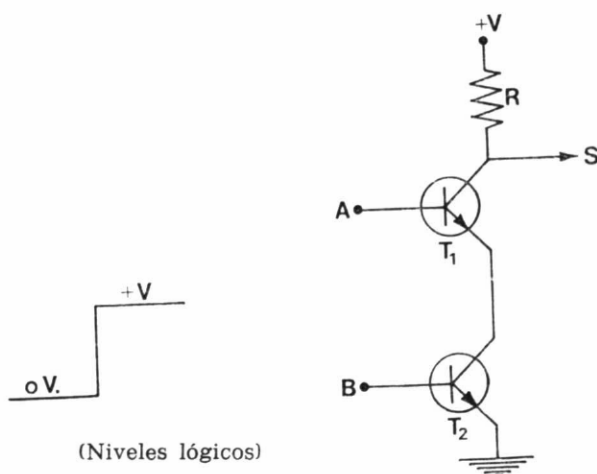


FIG. 7.24

Si A y B son potenciales de 0 voltios, los dos transistores están en estado de corte y la salida es

$$V_s = +V$$

Si A y B son, respectivamente, 0 y $+V$ voltios, el transistor T_1 está en estado de corte, en cuyo caso

$$V_s = +V$$

Si A y B son potenciales de $+V$ voltios, los dos transistores, T_1 y T_2 , están en estado de saturación y la tensión de salida es

$$V_s = (V_{CE})_{SAT-T_1} + (V_{CE})_{SAT-T_2} \simeq 0 \text{ voltios}$$

Con transistores de silicio epitaxiales se pueden conseguir valores para $(V_{CE})_{SAT}$ del orden de 0,1 voltios para cada transistor.

En el caso de una puerta con cinco entradas, la tensión de salida, para el estado 0, sería:

$$V_s = 5(V_{CE})_{SAT} \simeq 0,5 \text{ voltios}$$

Para representar el estado 1 se puede elegir un valor claramente diferente de 0,5 voltios; por ejemplo, $+3$ voltios.

Las tablas de tensiones y verdad del circuito de la figura 7.24 son:

A	B	S
0 voltios	0 voltios	$+V$ voltios
0 voltios	$+V$ voltios	$+V$ voltios
$+V$ voltios	0 voltios	$+V$ voltios
$+V$ voltios	$+V$ voltios	0 voltios

A	B	S		A	B	S	
0	0	1	$0V \equiv 0$	1	1	0	$0V \equiv 1$
0	1	1	$+V \equiv 1$	1	0	0	$+V \equiv 0$
1	0	1	Lógica positiva	0	1	0	Lógica negativa
1	1	0		0	0	1	

$$S = (AB)'$$

Puerta *NAND* con lógica positiva

$$S = A'B' = (A+B)'$$

Puerta *NOR* con lógica negativa

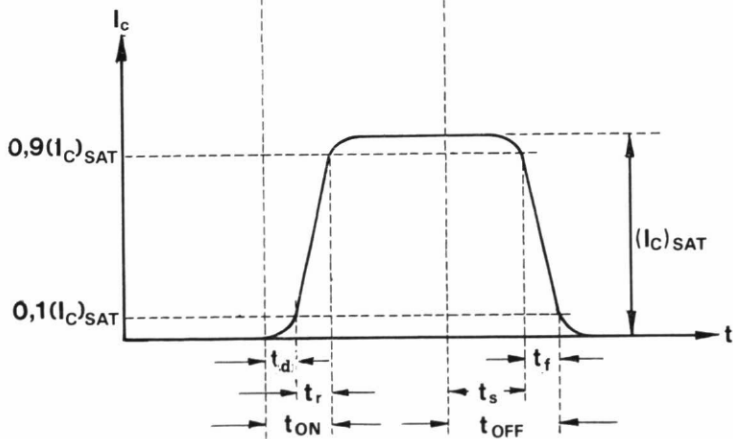
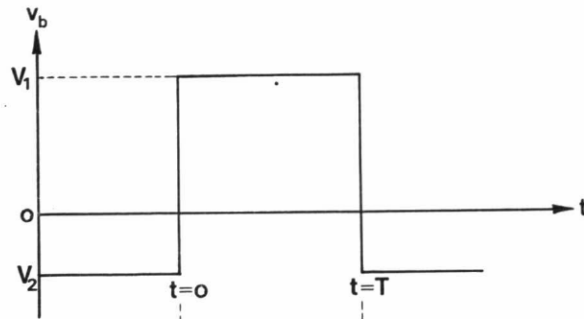
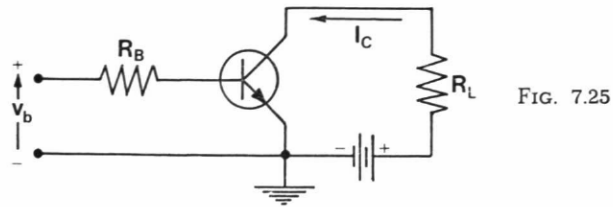
Entre las ventajas de este tipo de puertas, pueden citarse:

- Las tensiones de funcionamiento son menores que con lógica *DTL*.
- Se produce menos disipación en cada puerta.
- Son compatibles las puertas *NAND* y *NOR*.

En circuitos convencionales, la lógica *DCTL* es más cara que la *DTL*, ya que la primera requiere un transistor por cada término literal.

7.8 TIEMPOS DE CONMUTACION DEL TRANSISTOR

Consideremos el circuito de la figura 7.25, excitado por un impulso cuya forma de onda se representa en la figura 7.26.



La forma de onda de v_b varía entre los niveles de tensión V_2 y V_1 . En V_2 el transistor está al corte y en V_1 está en saturación.

v_b se aplica entre la base y el emisor a través de una resistencia R_B que puede incluirse explícitamente en el circuito o que puede representar la impedancia de salida de la fuente que suministra la señal de entrada.

En la figura 7.27 se representa la corriente de colector, como respuesta a la señal de entrada. La corriente no responde inmediatamente a la señal de entrada. A continuación se definen los tiempos indicados en la figura 7.27:

$t_d =$ tiempo de retardo: Tiempo necesario para que la corriente alcance el 10 por 100 de su valor máximo de saturación.

$$(I_C)_{SAT} \simeq \frac{V_{CC}}{R_L}$$

$t_r =$ tiempo de subida: Tiempo necesario para que la corriente pase del 10 al 90 por 100 del valor $(I_C)_{SAT}$.

$t_{ON} =$ tiempo de cierre:

$$t_{ON} = t_d + t_r$$

$t_s =$ tiempo de almacenamiento: Cuando la señal de entrada vuelve a su estado inicial en $t = T$, la corriente no varía instantáneamente.

El intervalo de tiempo que transcurre entre la transición de la forma de onda de entrada y el instante en que I_C ha caído al 90 por 100 de su valor máximo, se llama tiempo de almacenamiento.

$t_f =$ tiempo de caída: Tiempo que tarda I_C en caer del 90 al 10 por 100 de $(I_C)_{SAT}$.

$t_{OFF} =$ tiempo de apertura:

$$t_{OFF} = t_s + t_f$$

EJEMPLO: Para el transistor 2N3830 NPN, los valores anteriores pueden ser:

$$t_d = 10 \text{ n seg.}$$

$$t_r = 50 \text{ n seg.}$$

$$t_s = 40 \text{ n seg.}$$

$$t_f = 30 \text{ n seg.}$$

Se pueden disminuir los tiempos de subida y caída, colocando una capacidad en paralelo con la resistencia de base R_B . Este método disminuye también el tiempo de almacenamiento t_s .

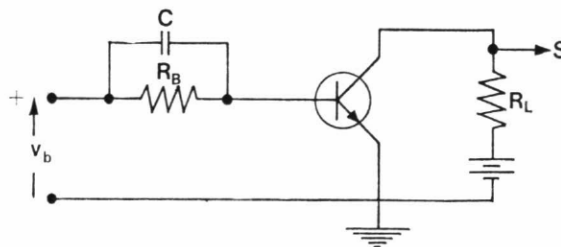


FIG. 7.28

En la práctica, se utilizan transistores de conmutación rápida (es decir, que posean buenas propiedades a las frecuencias elevadas) en lugar de emplear condensadores, ya que éstos introducen efectos no deseados en ciertos circuitos.

7.9 DISEÑO CON PUERTAS NAND Y NOR TRANSISTORIZADAS

Al usar puertas *DCTL*, el diseñador debe tener presente dos consideraciones:

- a) Las puertas *DCTL* realizan las operaciones *NAND* y *NOR*.
- b) Los criterios de optimización para circuitos *DCTL* no han sido todavía claramente definidos. El número de transistores utilizados sirve como indicación aproximada del coste.

Otros factores que deben tenerse en cuenta son:

1. Coste de fabricación.
2. Seguridad.
3. Retardos.
4. Carga de cada puerta (*fan-out*).
5. No hay restricción en el número de niveles, con tal de no sobrepasar el retardo permitido.
6. La mínima suma de productos y el mínimo producto de sumas no representan, necesariamente, el circuito optimizado.
7. Las puertas *NAND* y *NOR* son compatibles en el mismo circuito.
8. Es necesario especificar para cada entrada su disponibilidad en forma negada o no negada.

Es de utilidad tener en cuenta las equivalencias de la figura 7.29.

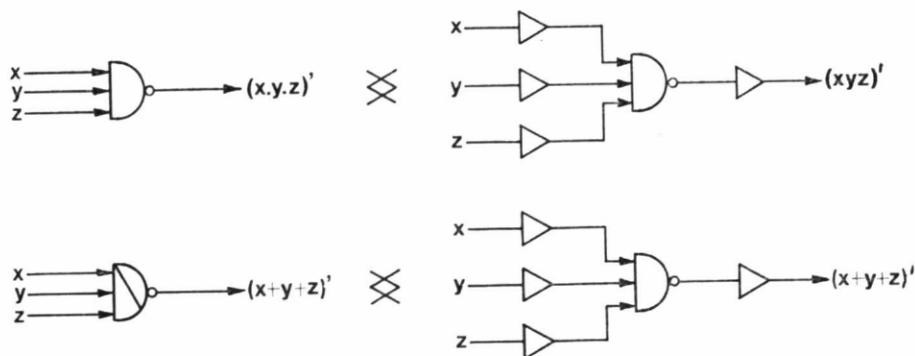


FIG. 7.29

EJEMPLO: Instrumentar con circuitos lógicos *DCTL* la función $S = A \cdot B'$. Las entradas A y B sólo se tienen disponibles en forma no negada.

1.º Realización con tres niveles:

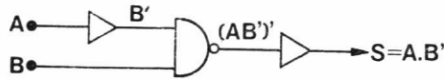


FIG. 7.30

Requiere cuatro transistores.

2.º Realización con dos niveles:

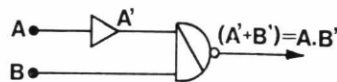


FIG. 7.31

Requiere tres transistores.

Conviene recordar que las puertas *NAND* y *NOR* de una sola entrada realizan la función *NOT*.

7.10 CONCLUSIONES

En el dominio de los circuitos integrados, la tendencia se orienta hacia la utilización de los transistores *NPN* de silicio, ya que éste es prácticamente el único semiconductor utilizable; posee las cualidades requeridas para la fabricación en serie de elementos diversos, activos y pasivos, interconectados.

Los parámetros que caracterizan un circuito lógico transistorizado son:

1. Las necesidades de alimentación.
2. Los niveles lógicos requeridos.
3. La potencia disipada.
4. Las tolerancias de los niveles lógicos, para que no haya transición de un estado a otro.
5. La sensibilidad a la temperatura; es decir, la zona de temperatura en la cual el circuito funciona correctamente.
6. La velocidad de funcionamiento.
7. Los factores de entrada (*fan-in*) y salida (*fan-out*) del circuito.
8. El factor de mérito: $FM = \text{potencia disipada} \times \text{tiempo de propagación}$ que representa la energía necesaria para producir una señal identificable.

Cuanto más pequeño sea este parámetro, tanto mejor es el circuito.

APENDICE 1
Teorema de Thevenin

Cualquier red lineal de dos terminales puede reemplazarse por un generador igual a la tensión en circuito abierto que aparece entre los terminales, en serie con la impedancia de salida vista desde estas bornas.

La impedancia de salida es la que se ve entre los terminales de salida cuando se sustituyen todas las fuentes de energía independientes por sus impedancias internas.

Como ejemplo, véase la nota 2 del apartado 7.4.

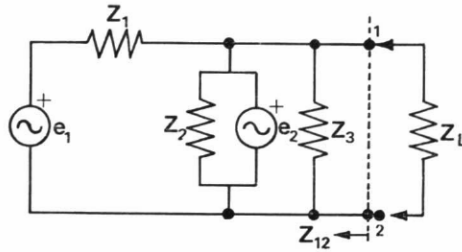


FIG. A.1

$Z_L \equiv$ impedancia de carga.

$v_{12} \equiv$ diferencia de potencial entre los terminales 1 y 2 en circuito abierto.

$Z_{12} \equiv$ impedancia de salida.

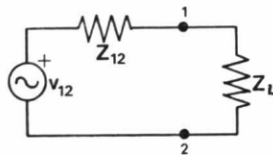


FIG. A.2

APENDICE 2

Funciones AND y OR cableadas

Sea la figura siguiente, que representa dos puertas *NAND*, de cuatro entradas, realizadas con lógica *DCTL*, cuyas salidas se han conectado directamente.

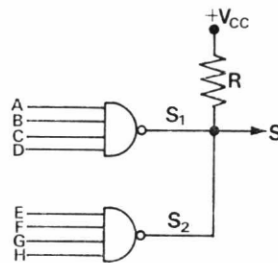


FIG. B.1

La tabla de verdad del circuito anterior es:

$S_1 = (A B C D)'$	$S_2 = (E F G H)'$	S
0	0	0
0	1	0
1	0	0
1	1	1

El examen de las señales de salida de las puertas *NAND* y de la señal de salida resultante del conjunto muestra que esta última es 1 si, y sólo si las señales de salida de todas las puertas *NAND* son 1.

La señal de salida es cero, cuando sea cero una cualquiera de las salidas de las puertas *NAND*.

Se observa que la conexión directa de los colectores (*collector-dotting*) realiza la operación *AND* con las salidas de las puertas *NAND*.

El circuito de la figura B.1 puede representarse mediante el logigrama de la figura B.2.

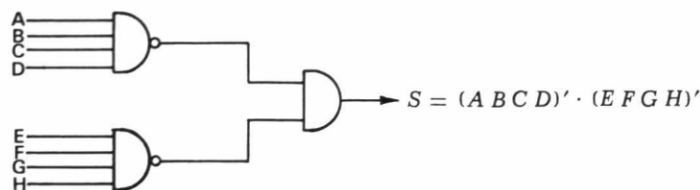


FIG. B.2

La función realizada por el cableado recibe el nombre de *AND cableada* (*wire-AND logic* o *dot-AND logic*) y se representa de la forma siguiente:

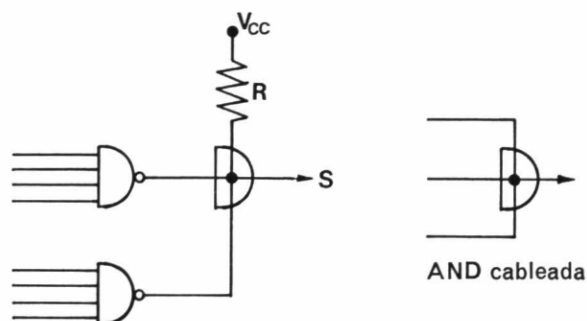


FIG. B.3

El circuito de la figura B.2 puede representarse también de la forma indicada en la figura B.4

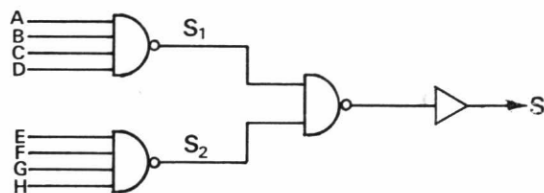


FIG. B.4

y realiza la operación *OR-AND* de las variables de entrada negadas, ya que

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

que también puede escribirse como:

$$S = [(A' + B' + C' + D') (E' + F' + G' + H')]'$$

$$S = [(A' + B' + C' + D')' + (E' + F' + G' + H')']'$$

$$S = (A B C D + E F G H)'$$

La última expresión indica que se realiza la operación *AND-OR-INVERSION* de las variables de entrada. Por este hecho se ha extendido erróneamente el término *OR cableada* (*wire-OR logic* o *dot-OR-logic*) para la operación realizada por el cableado directo de los colectores de las puertas *NAND*.

El método de obtención de la función *AND* mediante el cableado permite ahorrar dos niveles y evitar los retardos debidos a los mismos.

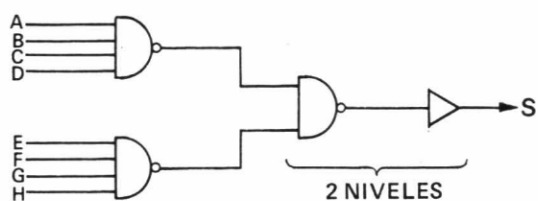


FIG. B.5

Para evitar ambigüedades en los logigramas conviene indicar la operación realizada por el cableado. Como puede verse en la figura B.3, las operaciones que se realizan son:

- a) Dos funciones mediante puertas *NAND*, y
- b) Otra función mediante la simple unión de dos salidas.

CAPITULO 8
REDES MULTITERMINALES

8.1 INTRODUCCION

Una red multiterminal es un circuito combinatorio de conmutación constituido por n entradas y m salidas.

Un circuito de este tipo puede realizarse diseñando un circuito combinatorio de una salida para cada una de las m salidas requeridas. Sin embargo, al diseñar un circuito con m salidas se pueden efectuar optimizaciones que no son realizables si se diseñan m circuitos separadamente.

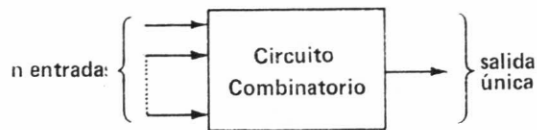


FIG. 8.1

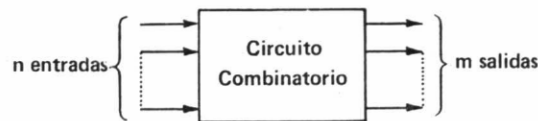


FIG. 8.2

Otro tipo de redes multiterminales son las constituidas por los circuitos de selección. En ellos, un conjunto específico de variables de entrada determina solamente una salida específica. Hay tantos conjuntos posibles de variables de entrada como salidas. Este problema puede resolverse:

- a) Mediante el diseño de m circuitos de conmutación independientes de una salida.
- b) Mediante el diseño de una red multiterminal.

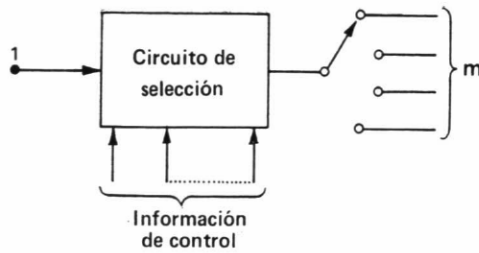


FIG. 8.3

Como se observa, un circuito de selección es capaz de seleccionar 1 entre m posibilidades, de acuerdo con la información de control recibida. (Ejemplo: establecimiento de una conexión.)

8.2 RED EN ESTRELLA

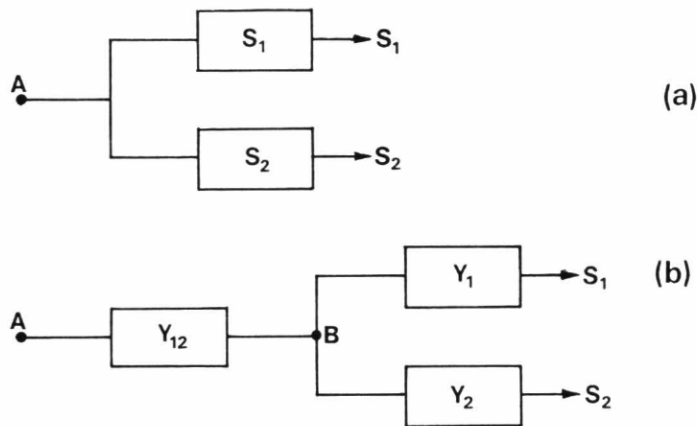


FIG. 8.4

Sean dos funciones lógicas, S_1 y S_2 , en las cuales aparecen simultáneamente algunas variables. Se pretende transformar la red de la figura 8.4 (a) en una red en estrella, tal como la de la figura 8.4 (b).

Para resolver este problema, se realizan los siguientes pasos:

- a) Establecimiento de una matriz común para S_1 y S_2 .
- b) Determinación de las matrices representativas de las funciones Y_{12} , Y_1 e Y_2 a partir de la matriz $S_1 - S_2$.

MATRIZ $S_1 - S_2$

Esta matriz contiene cuatro clases de condiciones de acuerdo con el contenido de sus áreas:

1. Areas que corresponden a $S_1 = 0$ y $S_2 = 0$. Estas áreas contienen 00.
2. Areas que corresponden a $S_1 = 1$ y $S_2 = 0$. Estas áreas contienen 10.
3. Areas que corresponden a $S_1 = 0$ y $S_2 = 1$. Estas áreas contienen 02.
4. Areas que corresponden a $S_1 = 1$ y $S_2 = 1$. Estas áreas contienen 12.

MATRIZ Y_{12}

Es evidente que Y_{12} debe tomar el valor binario 1 cuando:

- a) $S_1 = 1$,
- b) $S_2 = 1$, y
- c) $S_1 = 1$ y $S_2 = 1$.

Todas las condiciones restantes son indiferentes, y pueden ser utilizadas adecuadamente para simplificar la función Y_{12} . Sin embargo, conviene tener presente que las condiciones indiferentes de Y_{12} no utilizadas constituyen condiciones indiferentes para Y_1 e Y_2 .

MATRIZ Y_1

La matriz Y_1 debe tomar el valor binario 1 en todos los casos en que S_1 valga 1. En todos los casos en que Y_{12} valga 0, Y_1 puede tomar cualquier valor.

MATRIZ Y_2

La matriz Y_2 debe tomar el valor binario 1 en todos los casos en que S_2 valga 1. Cuando Y_{12} valga 0, el valor de Y_2 es indiferente.

Ejemplo:

$$S_1 = A(B + CD') + BC' = AB + ACD' + BC'$$

$$S_2 = DE'(A + BC') + AC = ADE' + BC'DE' + AC$$

DE \ ABC		000	001	011	010	110	111	101	100
		00	00	00	10	10	12	12	00
01		00	00	00	10	10	12	12	00
11		00	00	00	10	10	12	02	00
10		00	00	00	12	12	12	02	02

Matriz $S_1 - S_2$

DE \ ABC		000	001	011	010	110	111	101	100
		0	0	0	1	1	1	1	1
01		0	0	0	1	1	1	1	1
11		0	0	0	1	1	1	1	1
10		0	0	0	1	1	1	1	1

$Y_{12} = A + BC'$

DE \ ABC		000	001	011	010	110	111	101	100
		-	-	-	1	1	1	1	0
01		-	-	-	1	1	1	1	0
11		-	-	-	1	1	1	0	0
10		-	-	-	1	1	1	0	0

$Y_1 = B + CD'$

DE \ ABC		000	001	011	010	110	111	101	100
		-	-	-	0	0	1	1	0
01		-	-	-	0	0	1	1	0
11		-	-	-	0	0	1	1	0
10		-	-	-	1	1	1	1	1

$Y_2 = C + DE'$

En la matriz Y_{12} se ha colocado un 1 en las áreas rayadas de la última columna. Esto implica la colocación de ceros en las áreas homólogas de Y_1 e Y_2 . En las restantes áreas de Y_{12} se han colocado ceros.

El circuito resultante se muestra en la figura 8.5.

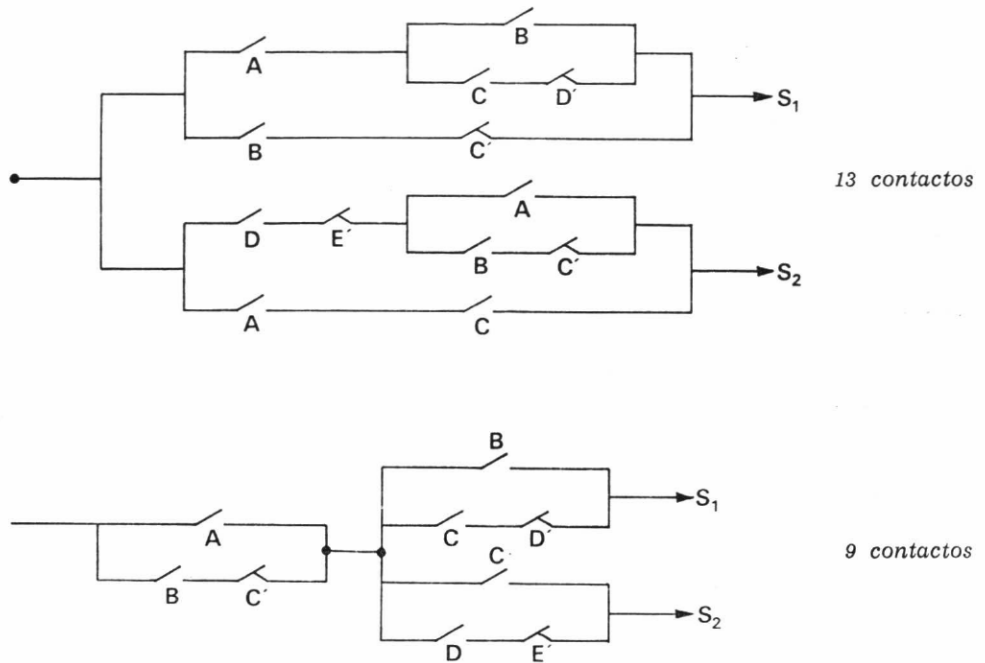


FIG. 8.5

Las correspondencias que se pueden establecer entre las áreas de $S_1 - S_2$ y las áreas de Y_{12} , Y_1 e Y_2 se resumen en el siguiente cuadro:

A R E A S

$S_1 - S_2$	Y_{12}	Y_1	Y_2
1-2	1	1	1
1-0	1	1	0
0-2	1	0	1
0-0	{ — 0	{ 0 — (*)	{ 0 —

(*) El guión (—) significa una condición opcional.

Veamos qué ocurre cuando $S_1 = S_2'$. La tabla anterior queda reducida a:

$S_1 - S_2$	Y_{12}	Y_1	Y_2
1-0	1	1	0
0-2	1	0	1

Es decir:

$$\begin{cases} Y_{12} = 1 \\ Y_1 = S_1 \\ Y_2 = S_2 \end{cases}$$

No existe una red en estrella que sea equivalente al circuito dado.

8.3 RED EN TRIANGULO

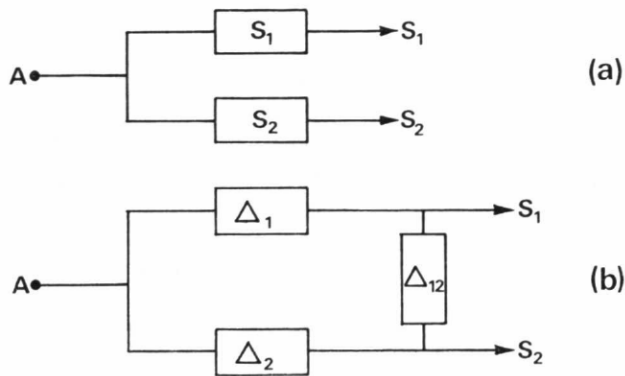


FIG. 8.6

Sean dos funciones lógicas, S_1 y S_2 , en las cuales aparecen simultáneamente algunas variables. Se pretende transformar la red de la figura 8.6 (a) en una red en triángulo, tal como la de la figura 8.6 (b).

Para resolver este problema, se realizan los siguientes pasos:

- Establecimiento de una matriz común para S_1 y S_2 .
- Determinación de las matrices representativas de las funciones Δ_1 , Δ_2 y Δ_{12} a partir de la matriz $S_1 - S_2$.

MATRIZ $S_1 - S_2$

Esta matriz ya ha sido descrita en el punto 8.2.

MATRIZ Δ_1

La función Δ_1 debe tomar:

- El valor 1 en los casos en que la matriz $S_1 - S_2$ contenga el valor 1—0.
- El valor 0 en los casos en que la matriz $S_1 - S_2$ contenga los valores 0—0 y 0—2.
- Un valor cualquiera cuando la matriz $S_1 - S_2$ contenga el valor 1—2.

MATRIZ Δ_2

La función Δ_2 debe tomar:

- a) El valor 1 en los casos en que la matriz $S_1 - S_2$ contenga el valor 0—2.
- b) El valor 0 en los casos en que la matriz $S_1 - S_2$ contenga los valores 0—0 y 1—0.
- c) Un valor cualquiera cuando la matriz $S_1 - S_2$ contenga el valor 1—2.

MATRIZ Δ_{12}

La función Δ_{12} debe tomar:

- a) El valor 0 en los casos en que la matriz $S_1 - S_2$ contenga los valores 1—0 y 0—2.
- b) Un valor cualquiera cuando la matriz $S_1 - S_2$ contenga el valor 0—0.
- c) En los restantes casos puede tomar los valores «1» u «opcional» de acuerdo con los valores correspondientes de Δ_1 y Δ_2 .

Las correspondencias que se pueden establecer entre las áreas de $S_1 - S_2$ y las de Δ_{12} , Δ_1 y Δ_2 se resumen en la siguiente tabla:

$S_1 - S_2$	Δ_1	Δ_2	Δ_{12}
0—0	0	0	—
1—0	1	0	0
0—2	0	1	0
1—2	$\left\{ \begin{array}{l} 1 \\ 1 \\ - \end{array} \right.$	$\left\{ \begin{array}{l} 1 \\ - \\ 1 \end{array} \right.$	$\left\{ \begin{array}{l} - \\ 1 \\ 1 \end{array} \right.$

Ejemplo:

$$S_1 = AC' + A'C + BCD' + B'C'D$$

$$S_2 = BD' + B'D + AC'D + A'CD'$$

		CD			
		00	01	11	10
AB	00	00	12	12	12
	01	02	00	10	12
	11	12	12	00	12
	10	10	12	02	00

Matriz $S_1 - S_2$

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

Matriz Δ_1
 $\Delta_1 = AC' + A'C$

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

Matriz Δ_2
 $\Delta_2 = BD' + B'D$

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

Matriz Δ_{12}
 $\Delta_{12} = C'D + CD'$
 $\Delta_{12} = A'B' + AB$

Para Δ_{12} existen dos soluciones.

Los circuitos correspondientes se muestran en las figuras 8.7 y 8.8.

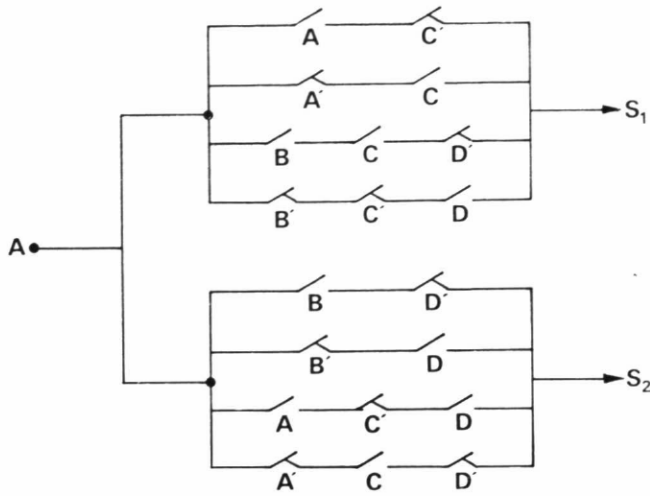


FIG. 8.7

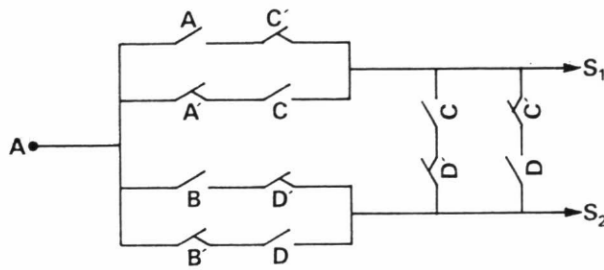


FIG. 8.8

☆

Veamos qué ocurre cuando $S_1 = S_2'$. En este caso, las correspondencias que se pueden establecer entre las áreas de $S_1 - S_2$, Δ_1 , Δ_2 y Δ_{12} son las siguientes:

$S_1 - S_2$	Δ_1	Δ_2	Δ_{12}
1-0	1	0	0
0-2	0	1	0

Es decir:

$$\begin{cases} \Delta_{12} = 0 \\ \Delta_1 = S_1 \\ \Delta_2 = S_2 \end{cases}$$

No existe una red en triángulo que sea equivalente al circuito dado

8.4 REDES EN ARBOL

Ya se ha visto que una función de conmutación puede expresarse como sumas de productos *standards*. También se ha comprobado que el empleo de contactos de transferencia contribuye a optimizar las redes de conmutación.

¿Puede realizarse una red de conmutación para instrumentar todos los posibles productos *standards*, utilizando únicamente contactos de transferencia?

La red multiterminal de la figura 8.9 está realizada solamente con contactos de transferencia.

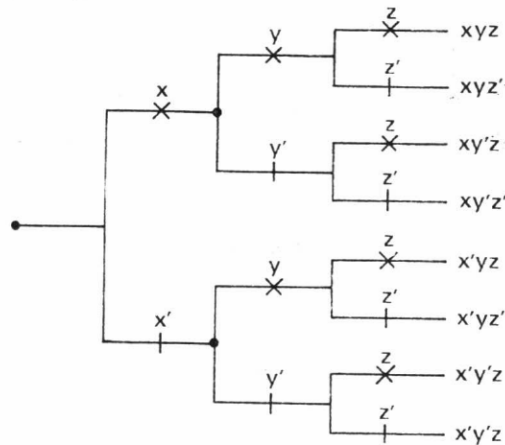


FIG. 8.9

Cada uno de los ocho terminales proporciona uno de los ocho productos *standards* de tres variables, x , y , z . Este tipo de red se conoce con el nombre de «redes en árbol» (o árbol de contactos). Con este ejemplo se muestra que una red de conmutación, materializada a base de contactos de transferencia, puede proporcionar todos los posibles productos *standards*.

Para concluir se puede decir que una red en árbol:

- a) Está realizada con contactos de transferencia.
- b) Proporciona todos los productos *standards*.
- c) Puede ser aprovechada para obtener una red de conmutación uniterminal o multiterminal mediante un proceso conocido como «poda del árbol» (*trimming a tree*).

EJEMPLO 1: Se pretende instrumentar con relés la siguiente función:

$$S = A B C' + A B' C + A' B C + A B C$$

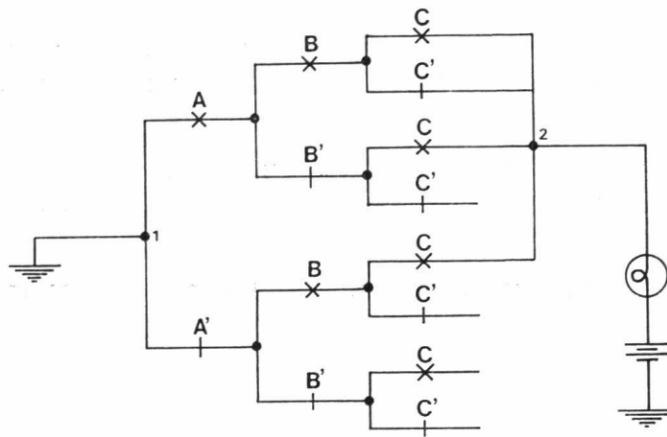


FIG. 8.10

La función S tiene tres variables de conmutación, por lo cual se parte de una red en árbol de tres variables.

Los terminales no conectados al punto 2 pueden eliminarse. También puede eliminarse el contacto de transferencia $C - C'$.

El circuito resultante es el siguiente:

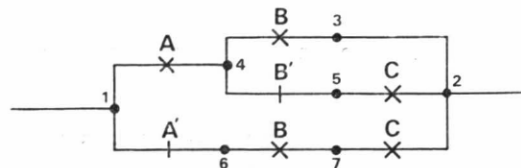


FIG. 8.11

El contacto B' de la rama 4-5-2 se puede eliminar, ya que:

$$B + B'C = B + C$$

Si se cortocircuitan los puntos 5 y 7 se obtiene el siguiente circuito:

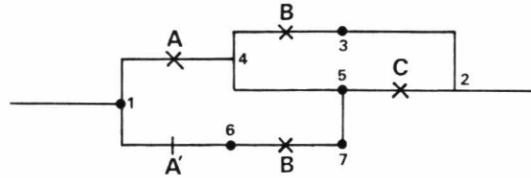


FIG. 8.12

Sin embargo, se ha creado el camino 1-6-7-5-4-3-2, cuya función de conmutación es $A'B B = A'B$, no especificada en la función S original. Los puntos 5 y 7 no pueden cortocircuitarse.

Pero si no se elimina el contacto B' de la figura 8.11, el camino creado 1-6-7-5-4-3-2 correspondería a la función $A'B B' B = 0$. En este caso, los puntos 5 y 7 pueden cortocircuitarse.

El circuito final se muestra en la figura 8.13 y requiere solamente 10 láminas de contacto.

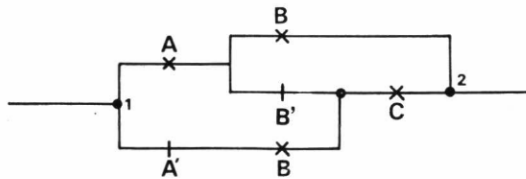


FIG. 8.13

☆

Sean dos funciones, S_1 y S_2 , que se van a instrumentar mediante el procedimiento de redes en árbol.

Si S_1 y S_2 cumplen la condición $S_1 = S_2'$, ya se ha visto que no existen redes en estrella ni en triángulo que sean equivalentes.

Cuando $S_1 = S_2'$ se puede utilizar el procedimiento de redes en árbol.

EJEMPLO 2:

$$S_1 = A B + C D$$

$$S_2 = (A' + B') \cdot (C' + D') = A' C' + A' D' + B' C' + B' D'$$

$$S_1 = S_2'$$

Para instrumentar estas funciones se construye una red en árbol de cuatro variables. Esta red se muestra en la figura 8.14.

Los terminales que representan productos *standards* contenidos en S_1 se conectan al terminal 2, y los que representan productos *standards* contenidos en S_2 se conectan al terminal 3.

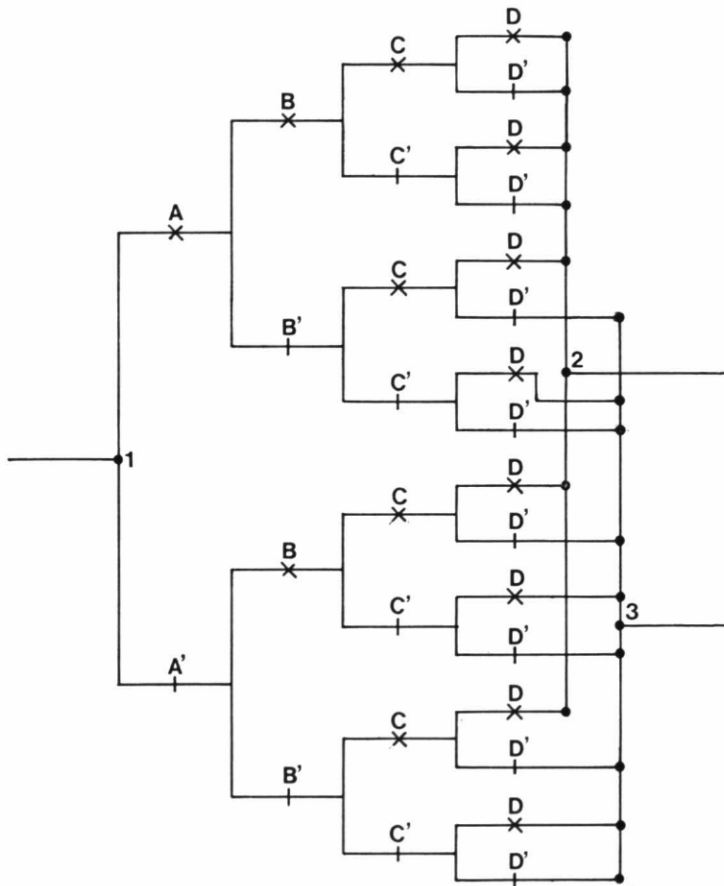


FIG. 8.14

La simplificación de S_1 y S_2 se realiza separadamente.

Los circuitos que corresponden a S_1 separadamente se muestran en la figura 8.15.

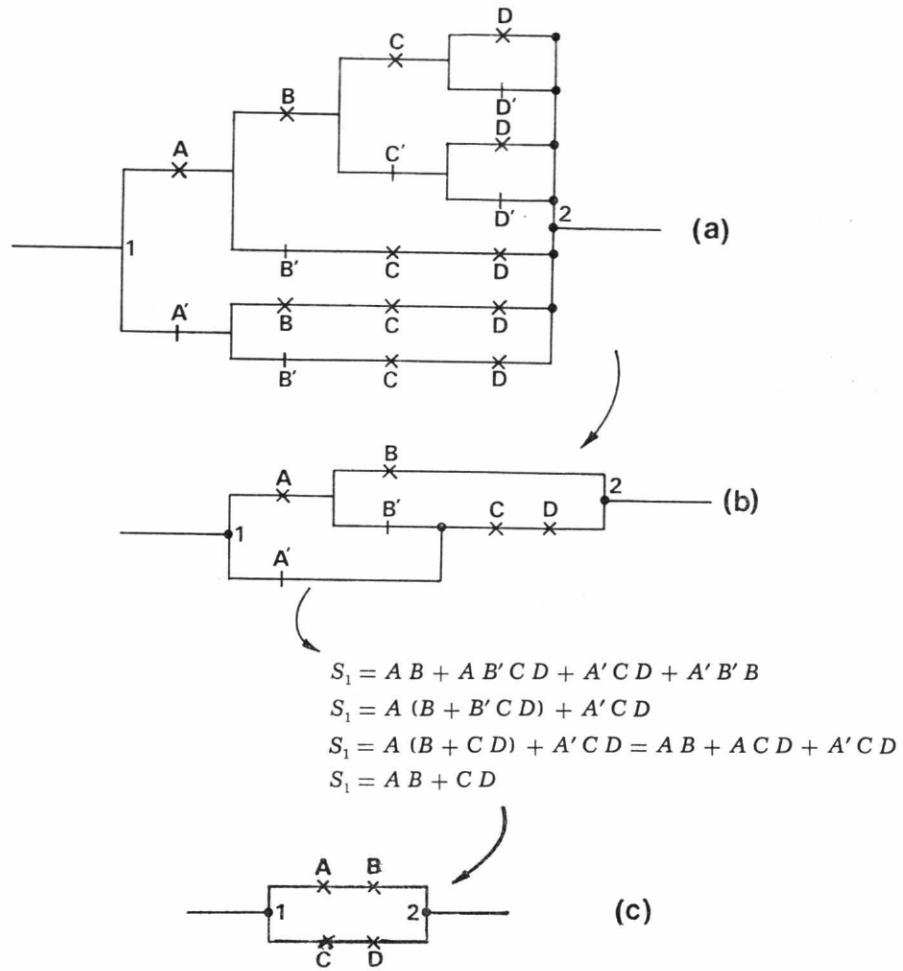


FIG. 8.15

Este circuito puede obtenerse de la función S_1 dada. Requiere ocho láminas.

Los circuitos que corresponden a S_2 son:

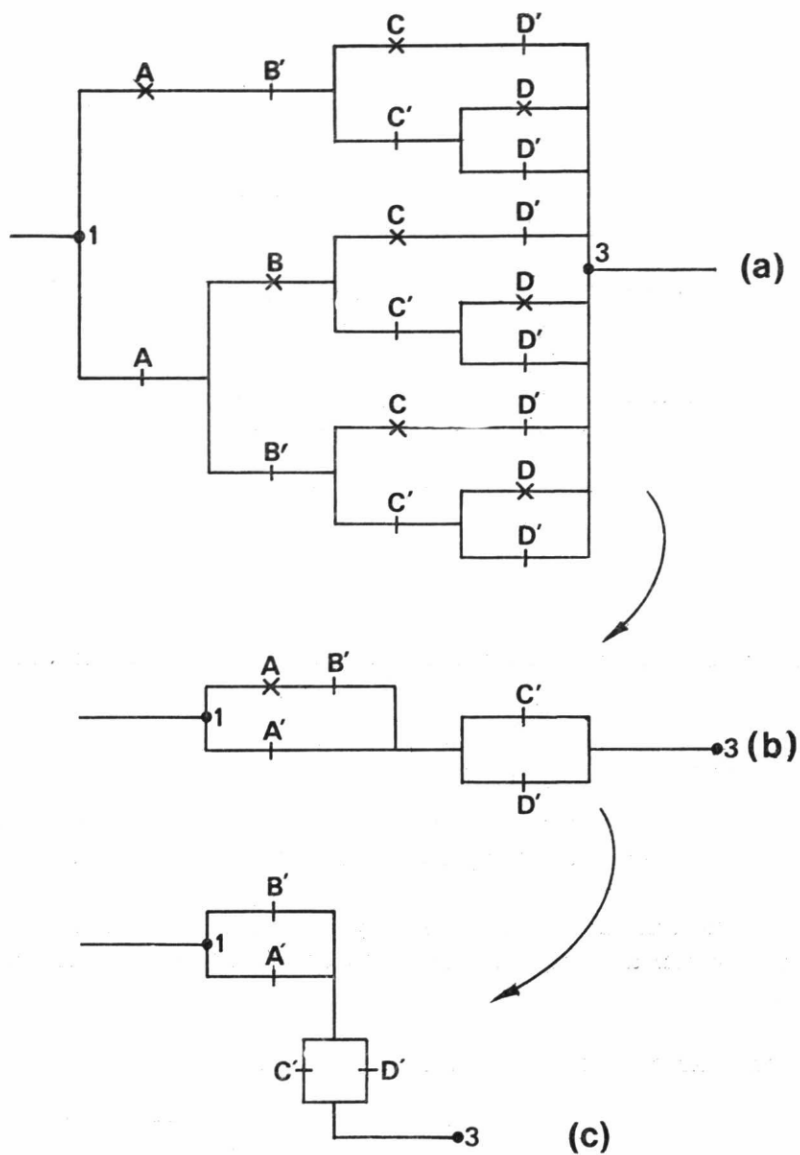


FIG. 8.16

Requiere ocho láminas.

Si las funciones S_1 y S_2 son instrumentadas por dos circuitos separados, conviene elegir los de las figuras 8.15 (c) y 8.16 (c), que solamente requieren 16 láminas.

Si se examinan los circuitos de las figuras 8.15 (b) y 8.16 (b), se ve que pueden combinarse ventajosamente para constituir un circuito más simplificado.

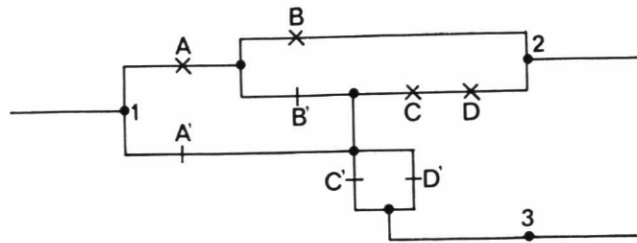


FIG. 8.17

Este circuito puede dibujarse de la forma siguiente:

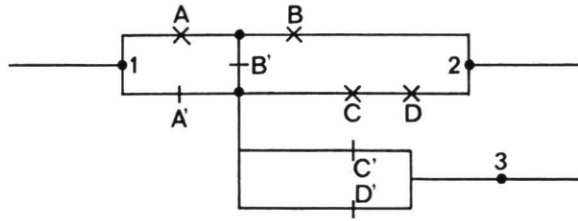


FIG. 8.18

Este circuito puede instrumentarse con $3 \times 3 + 2 \times 2 = 13$ láminas.

8.5 CODIFICADORES

Un *codificador* es una red multiterminal que posee tantas entradas como símbolos diferentes se quieren codificar y tantas salidas como dígitos tenga el código empleado.

Cuando se excita una línea cualquiera, y sólo una, se obtienen señales en las líneas que corresponden al código elegido. Los codificadores se pueden realizar por medio de circuitos *OR*.

EJEMPLO: Codificador 8 4 2 1 para las cifras 0 a 9.

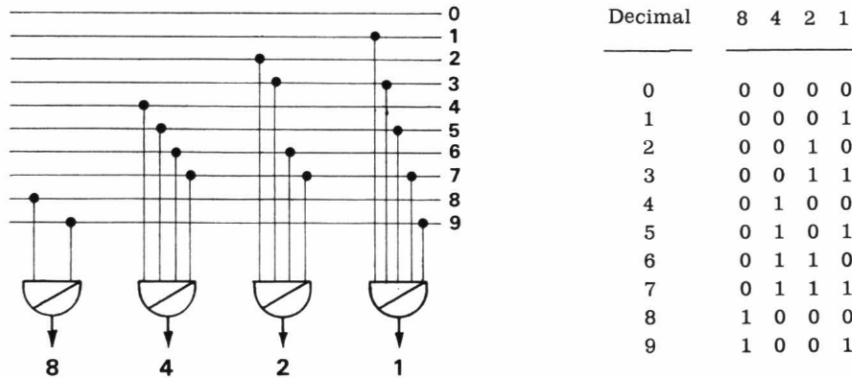


FIG. 8.19

8.6 DECODIFICADORES

Un decodificador es una red multiterminal que tiene tantas entradas como dígitos tenga el código y tantas salidas como símbolos diferentes se empleen.

Estos circuitos son muy utilizados.

Veamos un ejemplo: Sea un ordenador, cuyo formato de instrucción tenga la estructura siguiente:

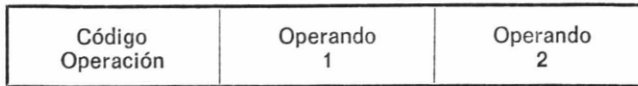


FIG. 8.20

Supongamos que el ordenador está diseñado para realizar 16 operaciones básicas. Se requiere un decodificador:

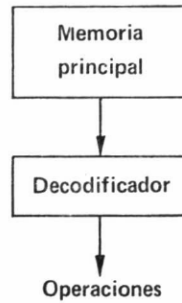


FIG. 8.21

Este decodificador se puede realizar mediante circuitos de conmutación de uno o dos niveles.

Decodificador de un nivel:

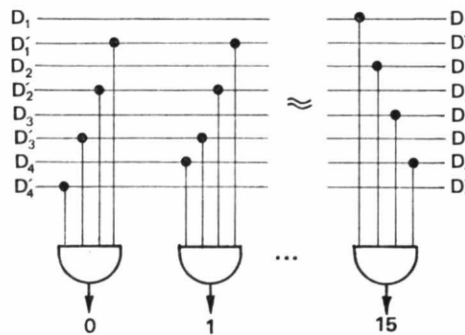


FIG. 8.22

Con diodos, se requieren $16 \times 4 = 64$ diodos.

Decodificador de dos niveles:

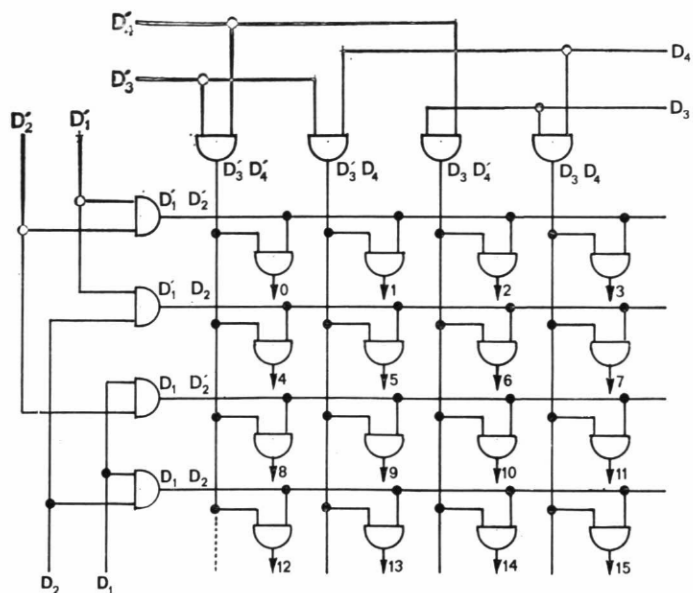


FIG. 8.23

Se requieren $16 + 8 = 24$ puertas *AND*, con un total de 48 diodos.

8.7 CONVERSORES DE CODIGO (*)

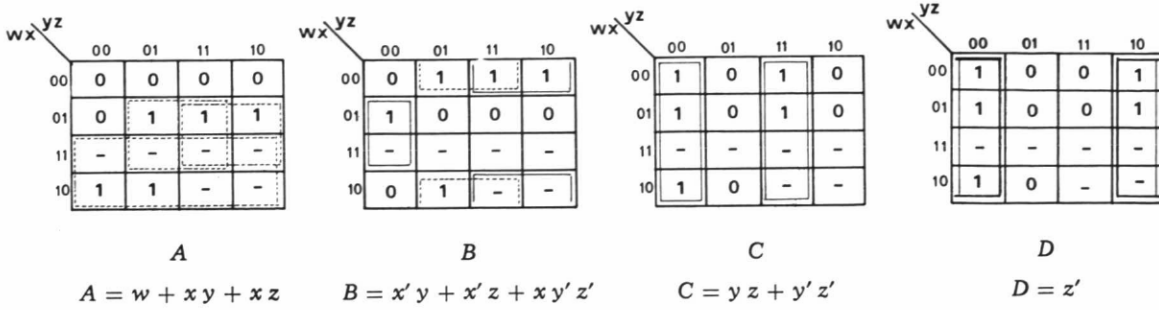
Un conversor de código es una red multiterminal que permite convertir un código en otro. Ejemplo: convertir el código *B C D* en código de exceso en tres.

La correspondencia entre estos dos códigos es:

Decimal	Código <i>B C D</i>				Código exceso en 3			
	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

(*) También reciben el nombre de transcodificadores o circuitos de traducción.

Para cada variable de salida, A , B , C y D , se puede construir un mapa de Karnaugh en función de las cuatro variables de entrada w , x , y , z .



El diagrama lógico del convertor de código es:

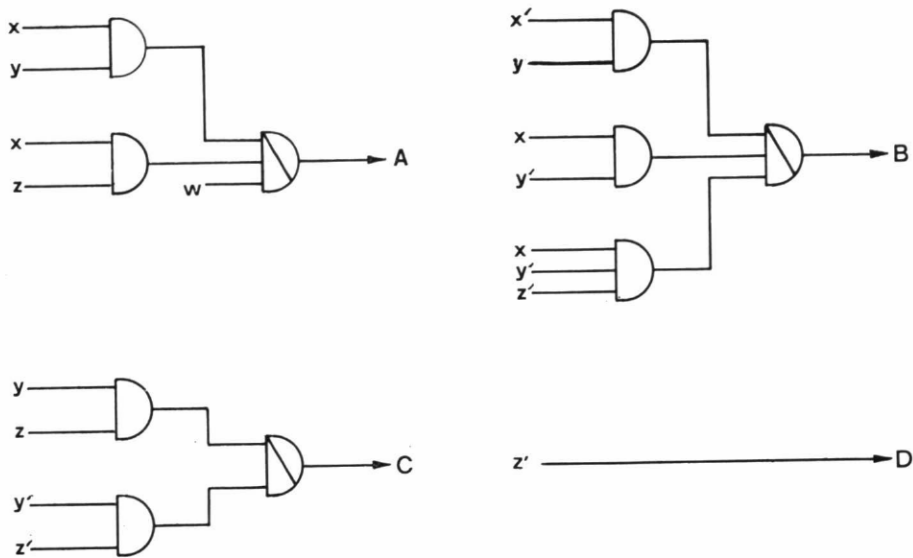


FIG. 8.24

CAPITULO 9

CIRCUITOS SUMADORES Y SUSTRADORES

9.1 SUMA Y SUSTRACCION BINARIAS

Existen varios procedimientos para realizar una suma binaria. En este apartado se van a describir someramente tres de ellos.

Sean dos números, x e y , cuyas expresiones son las siguientes:

$$x = a_n a_{n-1} a_{n-2} \dots a_2 a_1 a_0$$

$$y = b_n b_{n-1} b_{n-2} \dots b_2 b_1 b_0$$

Si se utiliza el sistema binario, los valores que pueden adoptar a_i y b_i son 0 y 1.

a) *Modo paralelo*

En este caso, las sumas de los dígitos binarios se efectúan al mismo tiempo; esto significa que las siguientes operaciones son realizadas simultáneamente:

$$a_0 + b_0$$

$$a_1 + b_1$$

$$\vdots$$

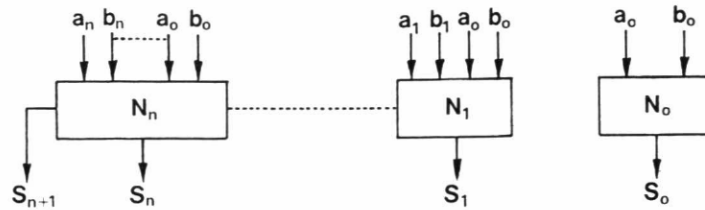
$$a_n + b_n$$

La suma de $a_0 + b_0$ puede producir un acarreo que debe ser considerado en la suma de $a_1 + b_1$. De acuerdo con esto, en la suma de dos dígitos binarios hay que considerar siempre el acarreo que ha podido producirse en una operación anterior. Es decir:

$$S_i = a_i + b_i + c_{i-1}$$

con lo cual, la suma de $a_{i+1} + b_{i+1}$ no puede realizarse hasta que se conozca el acarreo producido en la suma de $a_i + b_i + c_{i-1}$.

Para realizar todas las sumas simultáneamente, se recurre al siguiente diagrama de bloques:



El diagrama de bloques anterior corresponde a un sumador paralelo puro, también llamado sumador de acarreo simultáneos.

a_0 y b_0 determinan el dígito menos significativo de la suma (S_0).

El dígito S_1 se determina no solamente por a_1 y b_1 , sino también por a_0 y b_0 . La inclusión de a_0 y b_0 en el circuito N_1 asegura la anticipación del acarreo, pero incrementa el número de entradas del bloque.

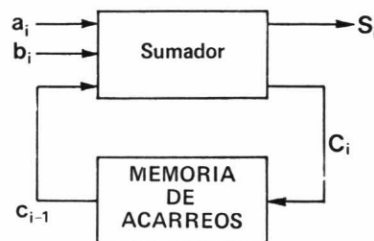
Los circuitos N_i son de tipo combinatorio, ya que las salidas S_i de los mismos, en cualquier instante, sólo están determinadas por las condiciones de entrada en ese instante.

Este procedimiento es muy rápido, pero requiere circuitos de conmutación muy complejos.

b) *Modo serie*

En este caso, las sumas de los dígitos binarios se realizan de una forma secuencial, teniendo en cuenta el acarreo producido en la operación anterior.

Un diagrama de bloques de un sumador serie es el siguiente:



La realización de estas operaciones requiere más tiempo que el necesitado en el modo paralelo; pero tiene la ventaja de usar el mismo circuito de conmutación repetidas veces.

Como se observa, la salida del circuito depende:

- 1) de las condiciones de entrada en ese instante;
- 2) de las entradas anteriores del circuito.

Se trata, pues, de un circuito secuencial.

c) *Modo iterativo o en cascada*

Los circuitos que realizan la suma se llaman también sumadores de acarreo horizontal. Se estudian en el capítulo 10.

Consideraciones análogas a las anteriores se pueden aplicar a la *sustracción binaria*.

9.2 SEMISUMADOR

Un sumador que solamente suma dos entradas se denomina semisumador (*half adder*). La tabla de verdad de este circuito es:

Entradas		Salidas	
A	B	C ₀	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Las funciones correspondientes para la suma y el acarreo son:

$$S = A' B + A B'$$

$$C_0 = A B$$

El diagrama lógico y símbolo de un semisumador se muestran en la figura 9.1.

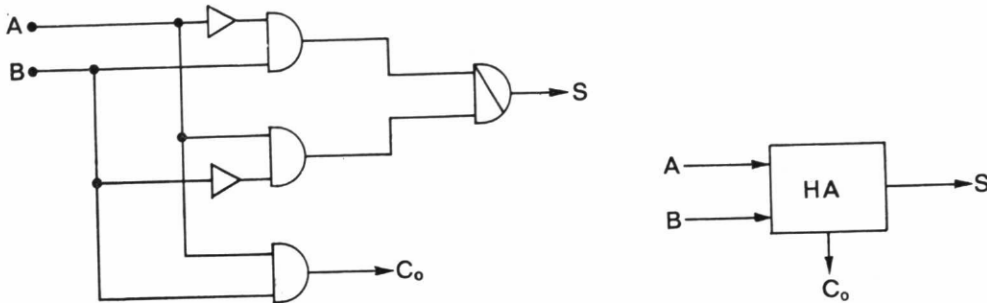


FIG. 9.1

El semisumador no es más que una puerta AND y un circuito OR exclusivo (figura 9.2).

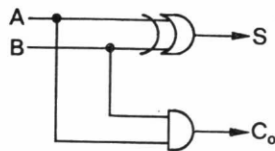


FIG. 9.2

Un circuito semisumador más sencillo se obtiene utilizando la función

$$S = (A B)' (A + B) = (A' + B') (A + B) = A' B + A B'$$

La función $C_0 = A B$ no necesita ser sintetizada por separado.

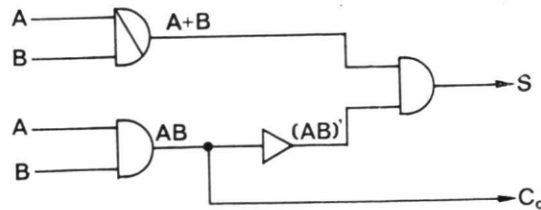


FIG. 9.3

Los circuitos de un semisumador obtenidos a partir de puertas NOR y NAND, se muestran en las figuras 9.4 y 9.5.

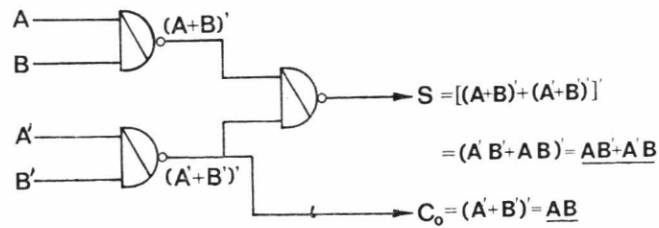


FIG. 9.4

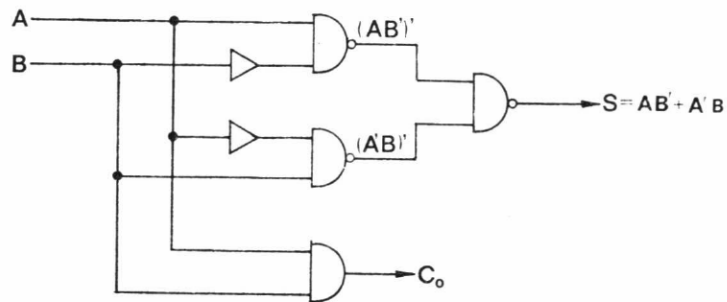


FIG. 9.5

9.3 SUMADOR COMPLETO

Un sumador completo (*full adder*) es un circuito que requiere tres entradas: A, B y el acarreo de la operación anterior.

La tabla de verdad del circuito es:

Entradas			Salidas	
A	B	C _i	S	C ₀
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

La expresión de la suma es la siguiente:

$$S = A' B' C_i + A' B C_i + A B' C_i + A B C_i$$

[1]

$$S = A' (B' C_i + B C_i) + A (B' C_i + B C_i)$$

o bien

$$S = C_i (A' B' + A B) + C_i' (A' B + A B')$$

La expresión que corresponde al acarreo es:

$$\begin{aligned}
 C_0 &= A' B C_i + A B' C_i + A B C_i' + A B C_i \\
 C_0 &= B C_i (A' + A) + A B (C_i' + C_i) + A C_i (B + B') \\
 C_0 &= B C_i + A B + A C_i \\
 C_0 &= A B + C_i (A + B)
 \end{aligned}$$

[2]

El diagrama lógico y el símbolo del circuito se muestran en la figura 9.6.

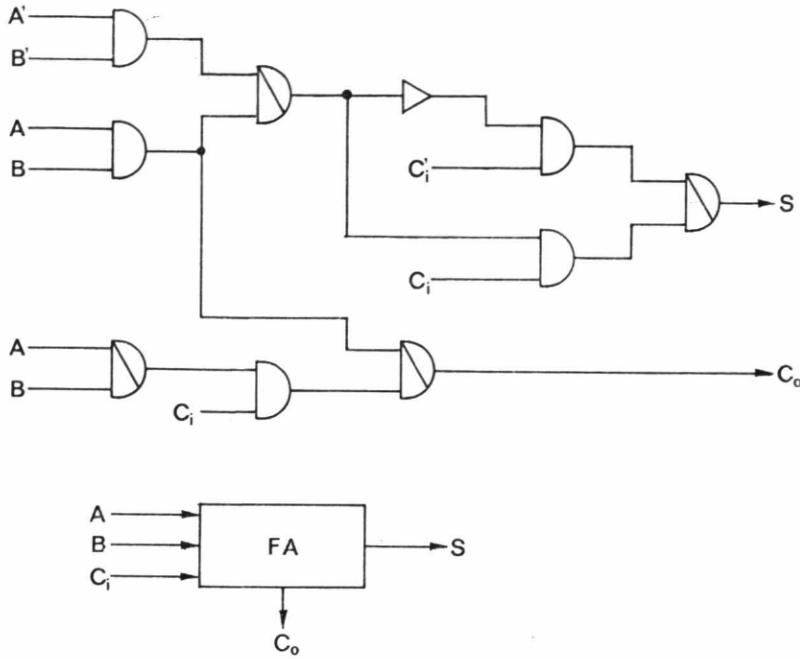


FIG. 9.6

A continuación se va a obtener el diagrama de un sumador completo utilizando puertas NAND:

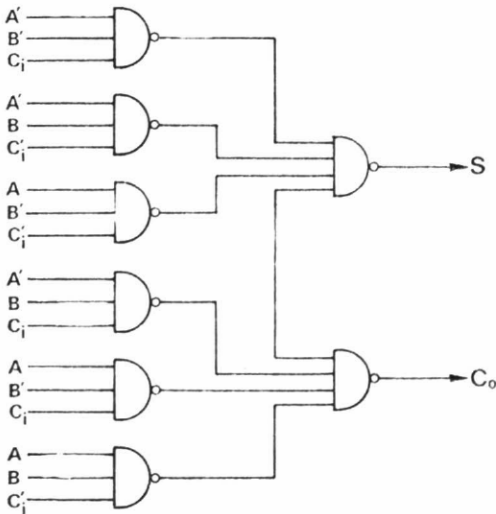


FIG. 9.7

$$\begin{aligned}
 S &= [(A' B' C_i + A' B C_i' + A B' C_i' + A B C_i)'] \\
 S &= [(A' B' C_i) \cdot (A' B C_i') \cdot (A B' C_i') \cdot (A B C_i)'] \\
 C_0 &= [(A' B C_i + A B' C_i + A B C_i' + A B C_i)'] \\
 C_0 &= [(A' B C_i)' (A B' C_i)' (A B C_i')' (A B C_i)']
 \end{aligned}$$

Si se instrumentan las funciones S y C_0 con lógica *DTL* de dos niveles, se utilizarían las expresiones [1] y [2], respectivamente. El número de diodos y transistores requeridos es:

Función	Diodos	Transistores
S	16	3
C_0	9	—
Totales	25	3

Otra realización que puede emplearse es la siguiente:

$$S = C_i (A' B' + A B) + C_i' (A' B + A B')$$

$$C_0 = A B + C_i (A' B + A B')$$

Se hace

$$M = A' B + A B' \quad [3]$$

con lo cual

$$M' = A B + A' B' \quad [4]$$

Las funciones S y C_0 resultan ser las siguientes:

$$S = C_i M' + C_i' M \quad [5]$$

$$C_0 = A B + C_i M \quad [6]$$

La expresión [3], que corresponde a M , indica que requiere dos inversores. Dado que interesa reducir este número, se realiza lo siguiente:

$$M = A' B + A B' = (A + B) (A' + B') = (A + B) (A B)' \quad [7]$$

La expresión [7] sólo requiere un inversor.

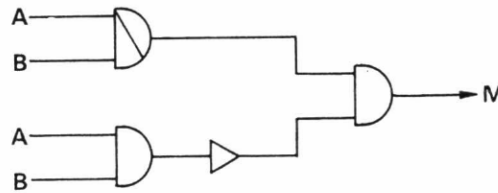


FIG. 9.8

El circuito de la figura 9.8 es un circuito de conmutación de diodos de dos niveles. Para generar S_i se requiere restaurar la señal de salida. Se obtiene, de esta forma, el circuito de la figura 9.9.

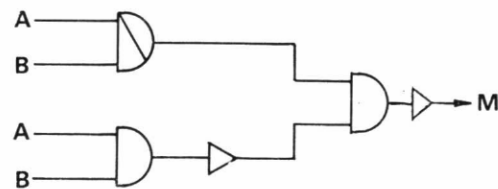


FIG. 9.9

Requiere seis diodos y dos transistores.

En la realización de la función S , interesa utilizar solamente la expresión de M' , por lo cual

$$S = C'_i M + C_i M' = (C_i + M)(C'_i + M') = (C'_i + M')(C'_i M')$$

que corresponde al circuito siguiente:

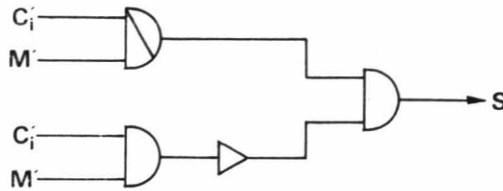


FIG. 9.10

Requiere seis diodos y un transistor.

Se observa que el acarreo debe ser generado en forma negada:

$$C_0 = A B + C_i M = A B + (C'_i + M')$$

$$C'_0 = [A B + (C'_i + M')]'$$

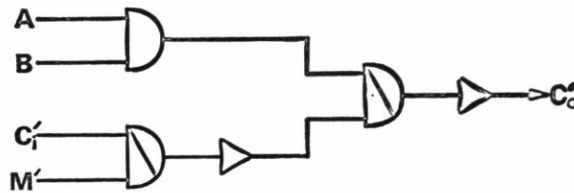


FIG. 9.11

Requiere seis diodos y dos transistores.

Como puede observarse, el circuito que se ha diseñado responde al siguiente diagrama de bloques:

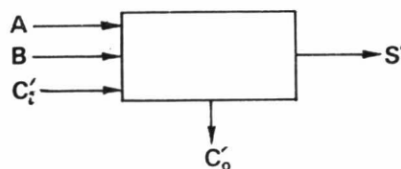


FIG. 9.12

y requiere un total de dieciocho diodos y cinco transistores.

9.4 SEMISUSTRADOR

Un sustractor que conste de dos entradas se denomina semisustractor (*half subtractor*).

La tabla de verdad de este circuito es la siguiente:

Entradas		Salidas	
x	y	B_0	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

(Tabla de verdad del semisustractor $x - y$)

En la tabla anterior se indica el resultado de la diferencia de dos números binarios x e y . Se denomina B_0 (*borrow*), a lo que puede llevarse al realizar la resta si el dígito y es mayor que el x .

Las ecuaciones de la diferencia (D) y del acarreo (B_0) son las siguientes:

$$D = x'y + xy'$$

$$B_0 = x'y$$

La realización lógica se indica en la figura 9.13.

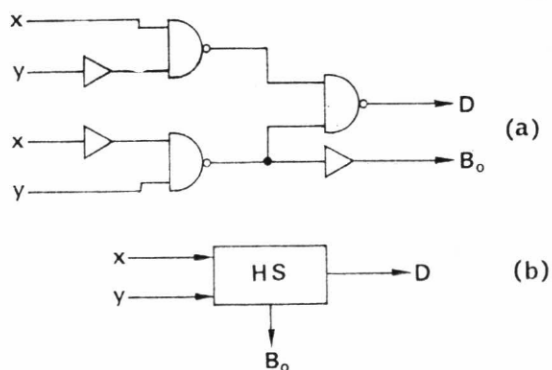


FIG. 9.13

9.5 SUSTRADOR COMPLETO

Un sustractor completo (*full subtractor*) es un circuito que requiere tres entradas x , y y el acarreo de la operación anterior.

La tabla de verdad del circuito es la siguiente:

Entradas			Salidas	
x	y	B_i	B_0	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

(Tabla de verdad del sustractor completo $(x - y) - B_i$)

Las ecuaciones booleanas serían:

$$D = x' y' B_i + x' y B_i' + x y' B_i' + x y B_i$$

$$B_0 = x' y' B_i + x' y B_i' + x' y B_i + x y B_i$$

$$B_0 = x' B_i + x' y + y B_i$$

Estas funciones se pueden realizar con puertas *NAND*, como se muestra en la figura 9.14.

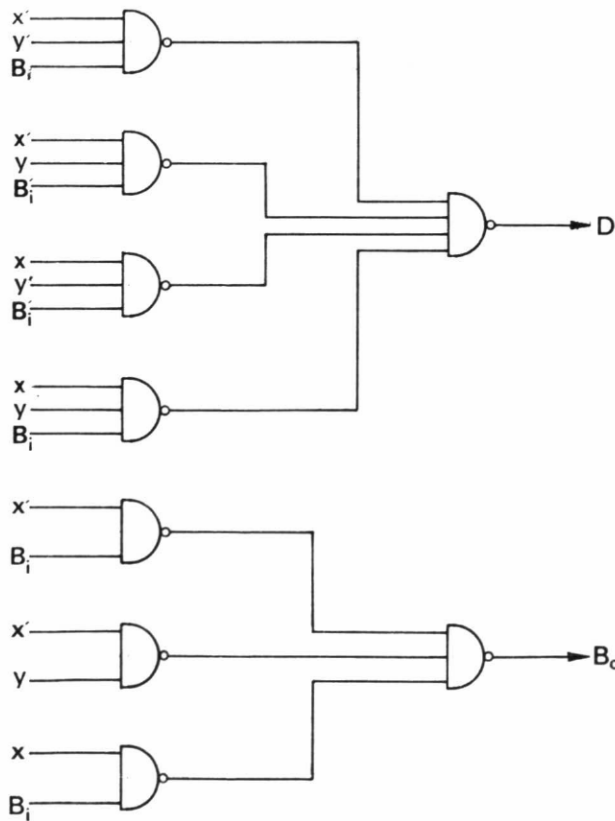


FIG. 9.14

El símbolo lógico normalmente utilizado para el sustractor completo es el siguiente:

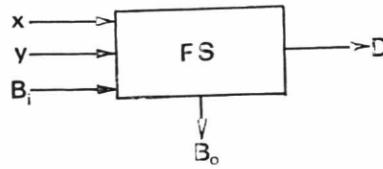


FIG. 9.15

☆

Otra realización alternativa que puede emplearse es la siguiente:

$$D = B_i(x'y' + xy) + B_i'(x'y + xy')$$

$$B_o = x'y(B_i' + B_i) + B_i(xy + x'y') = x'y + B_i(xy + x'y')$$

Haciendo $M = x'y + x'y'$, [7] se obtiene:

$$D = B_i M' + B_i' M \tag{8}$$

$$B_o = x'y + B_i M' \tag{9}$$

La expresión [7] que corresponde a M es análoga a la empleada para el circuito sumador completo. El circuito correspondiente a M' se muestra en la figura 9.9. Como ya se ha visto, requiere para su realización seis diodos y dos transistores.

La expresión [8], $D = B_i M' + B_i' M$, es análoga a la obtenida para la función S de un sumador completo. El circuito correspondiente a D se muestra en la figura 9.10, que requiere seis diodos y un transistor para su realización.

Se observa que el acarreo debe ser generado en forma negada:

$$B_o = x'y + B_i M'$$

$$B_o' = (x'y + B_i M)'$$

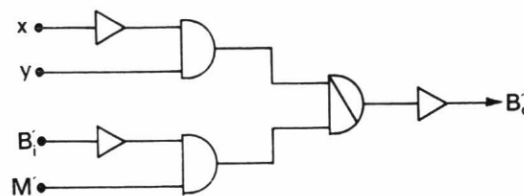


FIG. 9.16

Requiere seis diodos y tres transistores.

Como se puede observar, el circuito que se ha diseñado responde al siguiente diagrama de bloques:

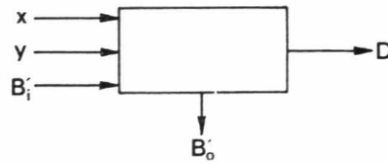


FIG. 9.17

y requiere 18 diodos y 6 transistores.

9.6 SUMADOR DECIMAL

Para diseñar un sumador decimal es necesario codificar los dígitos decimales en un código binario, por ejemplo, en BCD .

El diagrama de bloques de un sumador decimal se muestra en la figura 9.18.

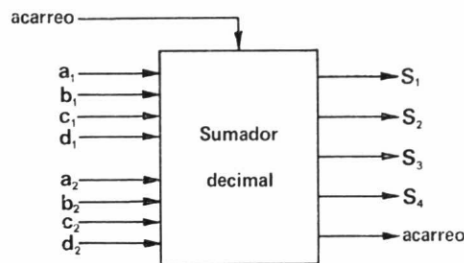


FIG. 9.18

El circuito consta de nueve entradas y cinco salidas:

- Las entradas a_1 , b_1 , c_1 y d_1 corresponden a un dígito decimal.
- Las entradas a_2 , b_2 , c_2 y d_2 corresponden al otro dígito decimal.
- Existe una entrada que corresponde al acarreo de una operación anterior.
- Las cinco salidas S_1 , S_2 , S_3 , S_4 y acarreo corresponden a la salida codificada y al posible acarreo que haya podido producirse.

Para realizar este diseño, se puede establecer un conjunto de cinco funciones de conmutación que expresen las cinco salidas del circuito. Este problema es laborioso, ya que al existir nueve variables de entrada, cada tabla de verdad consta de $2^9 = 512$ filas.

Se pueden utilizar dos métodos de simplificación:

- a) Emplear el método de tabulación de Quine-McCluskey mediante un programa adecuado tratado en un ordenador.
- b) Emplear el método iterativo que se explica en el capítulo 10.

CAPITULO 10
REDES EN CASCADA

10.1 INTRODUCCION

En el capítulo 9 se ha presentado el diagrama de bloques de un sumador paralelo puro. Este circuito se vuelve a repetir en la figura 10.1, para el caso de $n = 3$.

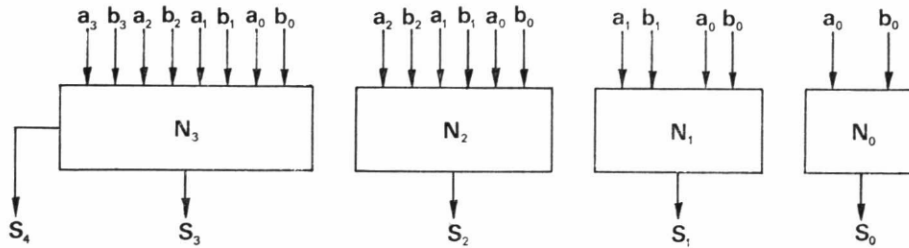


FIG. 10.1

En el caso anterior, los dos números binarios que se van a sumar son:

$$A = a_3, a_2, a_1, a_0$$

$$B = b_3, b_2, b_1, b_0$$

Se requieren cuatro circuitos separados, N_0 , N_1 , N_2 y N_3 . Para N_3 existen ocho variables de entrada, por lo cual la tabla de verdad correspondiente constará de $2^8 = 256$ filas. El diseño de este circuito es muy laborioso.

Es conveniente investigar la posibilidad de conectar en cascada varios circuitos más simples, a fin de formar el sistema deseado. Las variables de entrada de estos circuitos se reducen en número.

Un diagrama de bloques revisado se muestra en la figura 10.2.

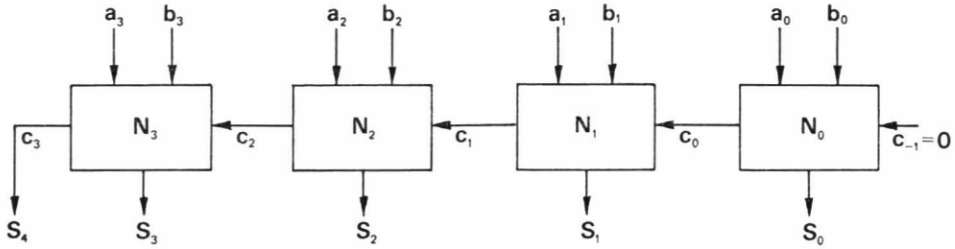


FIG. 10.2

En este diagrama, los dos bits menos significativos son procesados por N_0 para generar la suma (S_0) y el acarreo (c_0) que se envía a la siguiente etapa. Una explicación similar es aplicable a los circuitos N_1 , N_2 y N_3 .

Puede observarse que este circuito es más sencillo, pero más lento.

Los circuitos N_0 , N_1 , N_2 y N_3 son idénticos, cada uno tiene tres variables de entrada y dos variables de salida; reciben el nombre de células básicas (*typical cells*).

El circuito resultante de conectar en cascada varias células básicas se llama red en cascada o red iterativa.

10.2 CELULAS BASICAS

Son los circuitos idénticos que constituyen una red en cascada. Todos tienen el mismo número de entradas y el mismo número de salidas.

Ejemplo:

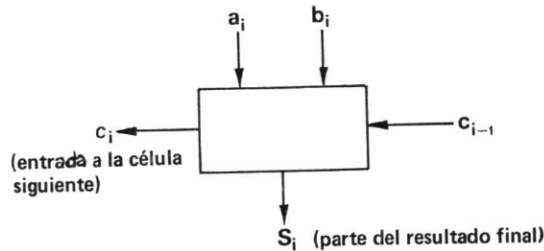


FIG. 10.3

Algunas de las salidas de la célula son consideradas como parte del resultado final del circuito completo, y otras se usan como entradas para otra célula.

10.3 FASES DEL DISEÑO DE UN CIRCUITO EN CASCADA

A continuación se indican las fases principales del diseño de una red en cascada:

1. *Identificación de la célula básica*, es decir, una distribución de las n variables de entrada y las m variables de salida entre un número determinado de células.

Generalmente, son posibles varias elecciones. Cada esquema de asignación diferente conduce a diferentes células básicas. Es responsabilidad del diseñador realizar una elección inteligente.

2. *Diseño de la célula básica.* Para ello, se debe tener en cuenta lo siguiente:

2.1 Información a transmitir entre células. La comunicación entre células de una red en cascada se consigue mediante ciertas entradas y salidas de cada célula. Algunas de las salidas de una célula son parte del resultado final del circuito completo y otras constituyen entradas para otras células. De este modo se logra una red en cascada; pero hay que tener presente el problema de la identificación de estados.

Reciben el nombre de *estados* las diferentes situaciones que deben ser detectadas. En la determinación del número de estados, se ha de cumplir que éstos sean necesarios y suficientes, a fin de evitar la redundancia y consiguiente complicación de la célula básica.

2.2 Definición del tipo de elemento de conmutación a utilizar:

a) *Relés*

Se asigna un hilo de interconexión entre células a cada estado, por lo cual se requieren tantos hilos como estados. Esta operación se llama asignación de estados. Por convenio, si un estado es válido en un cierto punto, en el hilo de estado correspondiente aparece un potencial característico, que en este texto se designa por V_+ .

b) *Diodos*

La realización física de estados en una red en cascada con diodos tiene los mismos condicionantes que en una red en cascada con relés. Esto se debe a que tanto los diodos como los relés no tienen capacidad de inversión. La aparición del voltaje V_+ en un punto específico, indica que el estado correspondiente es verdadero. La única diferencia entre ambas redes se debe a que la propagación del voltaje V_+ en un sentido se realiza mejor con diodos que con relés.

Se impone la condición de que cada célula tenga como máximo dos niveles. Dado que la salida de una célula se conecta a la entrada de la siguiente, se requiere restaurar la señal atenuada y aislar componentes que se interaccionan.

c) *Transistores*

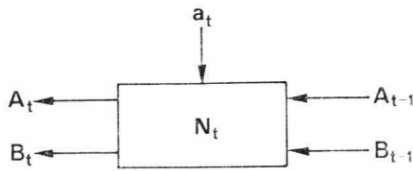
Debido a la capacidad de inversión de los transistores, para representar dos estados se puede usar dos niveles de voltaje en un solo hilo. Por este hecho, cada variable de conmutación X puede representar dos estados diferentes, A o B . Cuando $X = 1$, A es verdadero, y cuando $X = 0$, B es verdadero.

2.3 Establecimiento del diagrama de bloques de la célula básica.

2.4 Establecimiento de una tabla de transferencia de estados.

Ejemplo:

Estados de entrada	a_t	0	1
	A_{t-1}	A_t	B_t
	B_{t-1}	B_t	A_t



a_t : variable binaria de entrada de la célula N_t .

A_t y B_t : estados a la salida de la célula básica N_t .

A_{t-1} y B_{t-1} : estados a la entrada de la célula básica N_t .

- 2.5 Establecimiento de las funciones de conmutación, a partir de la tabla de transferencia de estados.
- 2.6 Establecimiento de la célula básica.
- 2.7 Establecimiento de la primera y última células.
- 2.8 Interconexión de células.

10.4 CIRCUITO DE COMPROBACION DE PARIDAD

Dada una secuencia de n dígitos binarios

$$a_{n-1}, a_{n-2}, \dots, a_2, a_1, a_0$$

diseñar un circuito que indique si el número de «unos» contenidos en esta secuencia es par o impar.

Para realizar este diseño, se va a usar el procedimiento iterativo.

1. *Identificación de la célula básica.*—Comencemos distribuyendo una variable de entrada a cada una de n células básicas. El esquema de interconexión de las n células sería el siguiente:

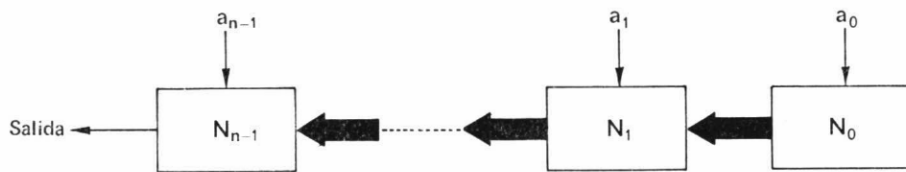


FIG. 10.4

2. *Información a transmitir entre células.*—En cada etapa es necesario conocer si el número de «unos» contados es par o impar y no hace falta conocer cuántos «unos» se han contado. Existen, pues, dos situaciones diferentes que deben ser detectadas. Para este problema, estos dos estados son necesarios y suficientes. No es

posible reducirlos a un solo estado ni conviene definir más de dos, ya que existiría redundancia en la célula básica. Sea:

A, estado para el cual el número de «unos» es par.

B, estado para el cual el número de «unos» es impar.

3. *Realización con relés.*—En la salida de la célula básica, solamente uno de los dos estados es verdadero; es decir, la situación en ese punto puede describirse por uno de los dos estados, pero no por ambos.

En el presente problema se requieren dos hilos, uno para cada estado.

El diagrama de bloques de la célula básica sería el siguiente:

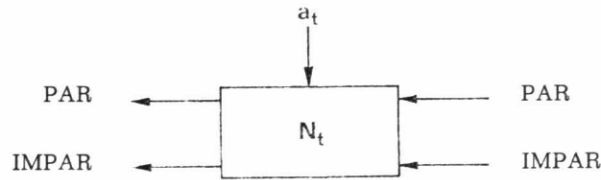


FIG. 10.5

La célula consta de tres entradas y de dos salidas.

La tabla de transferencia de estados se muestra a continuación:

	a_t	
	0	1
A_{t-1}	A_t	B_t
B_{t-1}	B_t	A_t

donde

a_t indica el dígito binario examinado por la célula N_t .

t indica la célula t .

$t-1$ indica la célula $t-1$.

Esta tabla consta de dos filas, una por cada estado, y de dos columnas, una para $a_t = 0$ y otra para $a_t = 1$.

Cuando el estado de entrada es A_{t-1} , el número de «unos» contados por todas las células precedentes es par. En este caso, si $a_t = 0$, el número de «unos» contados por N_t y todas las células precedentes seguirá siendo *par*. Sin embargo, si $a_t = 1$, el número de «unos» pasará a ser *impar*.

La segunda fila puede explicarse de la misma forma.

Se pueden establecer las siguientes ecuaciones:

$$A_t = A_{t-1} \cdot a'_t + B_{t-1} \cdot a_t$$

$$B_t = A_{t-1} \cdot a_t + B_{t-1} \cdot a'_t$$

De acuerdo con estas ecuaciones, la célula básica quedaría constituida de la forma:

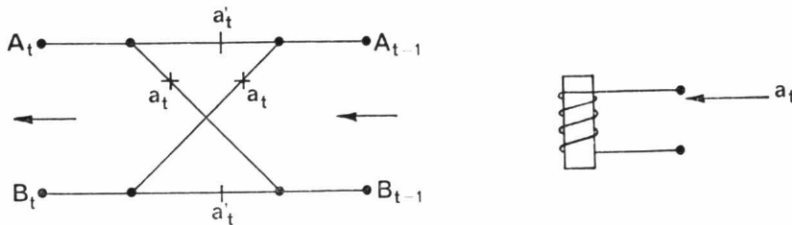


FIG. 10.6

El bit a_i se emplea para controlar la corriente en la bobina del relé. Si $a_i = 1$, circula una corriente por la bobina, y si $a_i = 0$, no existe dicha corriente. Los contactos del relé a_i se emplean para transmitir el nivel de voltaje V_+ .

Examinemos cómo debe ser la primera célula. A la izquierda de la primera célula no se examina ningún dígito; es decir, el número de «unos» contados es cero, que puede considerarse como par. Por ello, se conecta V_+ al hilo de entrada que representa el estado que corresponde al número de «unos» contados par.

Si el primer dígito es 0, en A_0 se tiene V_+ , y si el primer dígito es 1, en B_0 se tiene V_+ .

La primera célula tiene, pues, la configuración indicada en la figura 10.7.

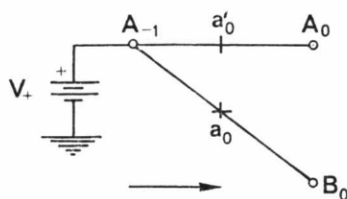


FIG. 10.7

El circuito completo resultante es el siguiente:

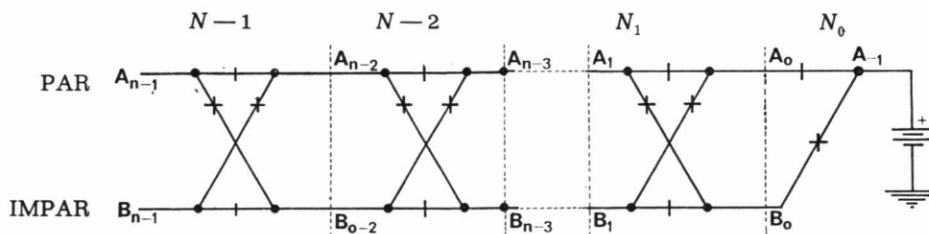


FIG. 10.8

En el diseño realizado se han utilizado dos hilos: A y B . El estado A es verdadero si V_+ aparece en el hilo A , y el estado B es verdadero cuando V_+ aparece en el hilo B .

¿Podría usarse un solo hilo para representar estos dos estados? Por ejemplo:

- Si V_+ aparece en el hilo, A es verdadero, y
- Si V_+ no aparece, B es verdadero.

En este caso, la red en cascada resultante sería:



FIG. 10.9

Puede observarse que la propagación del potencial V_+ puede ser parada en cualquier célula, y una vez detenida, nunca puede aparecer a la salida, aunque la paridad sea par.

Este hecho se debe a que los relés no poseen capacidad de regeneración del voltaje.

Se puede concluir afirmando que se requieren dos hilos, uno para cada estado.

En el diseño realizado se ha asignado una variable de entrada a cada célula básica. Veamos, ahora, cómo resulta dicha célula asignando dos variables a cada una de ellas.

En este caso, la tabla de transferencia de estados resultante es:

	$a_\alpha \ a_\beta$			
	00	01	11	10
A_{t-1}	A_t	B_t	A_t	B_t
B_{t-1}	B_t	A_t	B_t	A_t

Las funciones de conmutación correspondientes son:

$$A_t = A_{t-1} (a'_\alpha a'_\beta + a_\alpha a_\beta) + B_{t-1} (a'_\alpha a_\beta + a_\alpha a'_\beta)$$

$$B_t = A_{t-1} (a'_\alpha a_\beta + a_\alpha a'_\beta) + B_{t-1} (a'_\alpha a'_\beta + a_\alpha a_\beta)$$

Haciendo

$$M = a'_\alpha a'_\beta + a_\alpha a_\beta$$

$$N = a'_\alpha a_\beta + a_\alpha a'_\beta$$

se obtiene

$$A_t = M A_{t-1} + N B_{t-1}$$

$$B_t = N A_{t-1} + M B_{t-1}$$

La célula básica resultante es:

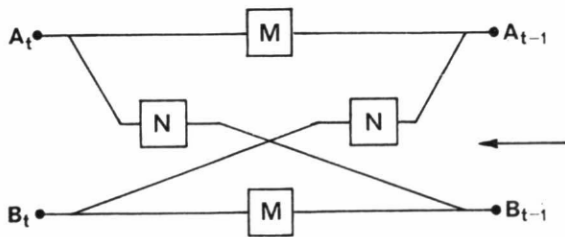


FIG. 10.10

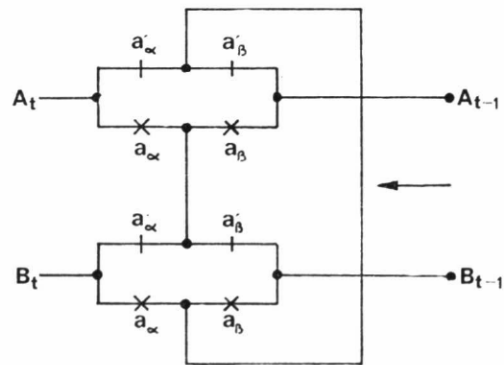


FIG. 10.11

que requiere cuatro contactos de transferencia (12 láminas). La célula básica de la figura 10.6 requiere dos contactos de transferencia (seis láminas).

Se observa que cada célula requiere seis láminas por cada variable de entrada.

4. *Realización con circuitos DCTL.*—Debido a la capacidad de inversión de los transistores, para representar dos estados se pueden usar dos niveles de voltaje en un solo hilo. Por este hecho se puede emplear una variable de conmutación para representar dos estados.

En el diseño presente, la variable de conmutación J_t puede utilizarse para representar los dos estados A y B :

- Si $J_t = 1$, entonces A_t es verdadero, y
- Si $J_t = 0$, entonces B_t es verdadero.

De acuerdo con esto, las ecuaciones

$$\begin{aligned} A_t &= A_{t-1} \cdot a'_t + B_{t-1} \cdot a_t \\ B_t &= A_{t-1} \cdot a_t + B_{t-1} \cdot a'_t \end{aligned} \quad [1]$$

pueden escribirse de la siguiente forma:

$$\begin{aligned} J_t &= J_{t-1} a'_t + J'_{t-1} a_t \\ J'_t &= J_{t-1} a_t + J'_{t-1} a'_t \end{aligned} \quad [2]$$

Los circuitos que corresponden a las ecuaciones [2] son:

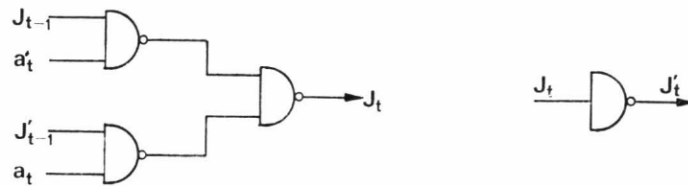


FIG. 10.12

Un diseño alternativo de la célula básica es:

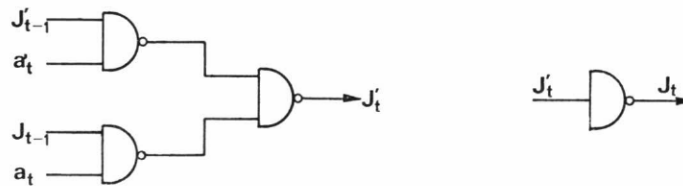


FIG. 10.13

10.5 CIRCUITO PARA LA DETERMINACION DEL NUMERO DE «UNOS»

Dado un mensaje de nueve dígitos binarios, se pretende diseñar un circuito de conmutación de relés que indique si el número de «unos» contenidos en el mensaje es exactamente cuatro.

Para realizar este diseño se va a utilizar el procedimiento iterativo.

1. *Identificación de la célula básica.*—Comencemos distribuyendo una variable de entrada a_t a cada una de las nueve células básicas.

2. *Información a transmitir entre células.*—Se pueden definir los siguientes estados:

- A: ningún bit es 1.
- B: un bit es 1.
- C: dos bits son 1.
- D: tres bits son 1.
- E: cuatro bits son 1.
- F: más de cuatro bits son 1.

Cuando más de cuatro bits contados son 1, el circuito, de acuerdo con la especificación, no debe dar señal de salida. Se podría pensar que son necesarios seis hilos para representar estos seis estados; sin embargo, no hay necesidad de suministrar el hilo F para propagar V_+ , ya que su aparición en la salida no es requerida.

3. *Realización con relés.*—El diagrama de bloques de la célula básica es el siguiente:

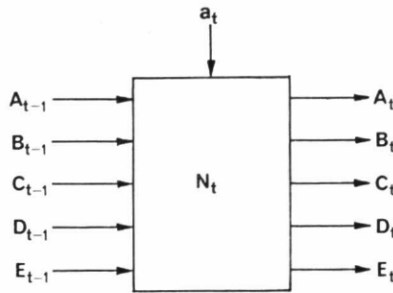


FIG. 10.14

La tabla de transferencia de estados se muestra a continuación:

a_t	0	1
A_{t-1}	A_t	A_t
B_{t-1}	B_t	B_t
C_{t-1}	C_t	C_t
D_{t-1}	D_t	D_t
E_{t-1}	E_t	F_t

De la tabla anterior se pueden establecer las siguientes ecuaciones:

$$\left. \begin{aligned}
 A_t &= A_{t-1} \cdot a'_t \\
 B_t &= A_{t-1} \cdot a_t + B_{t-1} a'_t \\
 C_t &= B_{t-1} \cdot a_t + C_{t-1} a'_t \\
 D_t &= C_{t-1} \cdot a_t + D_{t-1} a'_t \\
 E_t &= D_{t-1} \cdot a_t + E_{t-1} a'_t
 \end{aligned} \right\}$$

De acuerdo con estas ecuaciones, la célula básica quedaría constituida de la forma:

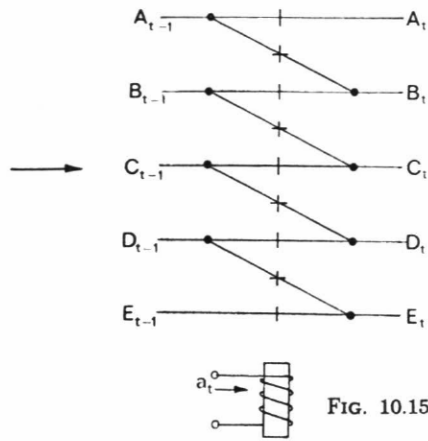


FIG. 10.15

Examinemos la situación en las primeras células. No existe dígito 1 antes de la primera célula. El voltaje V_+ se debe, pues, conectar en el terminal A_0 para indicar la validez del estado A . La simplificación obtenida en esta primera célula repercute en las tres siguientes, que también resultan simplificadas.

Examinemos, ahora, la situación en las últimas células. El circuito requiere que el potencial V_+ aparezca en el terminal de salida E_9 si el número de dígitos es exactamente cuatro. Por esta causa, sólo debe aparecer este terminal en la última célula. La simplificación realizada en esta última célula repercute en las tres precedentes, que también resultan simplificadas.

El circuito completo se muestra en la figura 10.16.

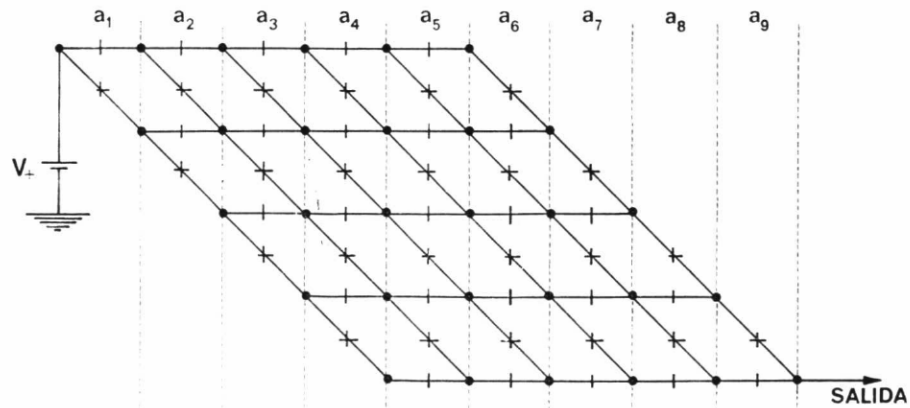


FIG. 10.16

10.6 SUMADOR BINARIO PARALELO

Un sumador binario es un circuito de conmutación capaz de generar la suma de dos números expresados en el sistema binario.

En el punto 10.1 se muestra el diagrama de bloques de un sumador binario paralelo que realiza la suma de dos números binarios:

$$A = a_3 a_2 a_1 a_0$$

$$B = b_3 b_2 b_1 b_0$$

El circuito consta de cuatro células básicas, que son sumadores completos.

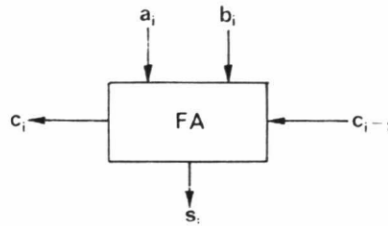


FIG. 10.17

Es más adecuado emplear, por las razones explicadas en el punto 9.3, como célula básica el circuito diseñado para un sumador completo, cuyo diagrama de bloques se muestra en la figura 9.12 y que se vuelve a repetir en la figura 10.18.

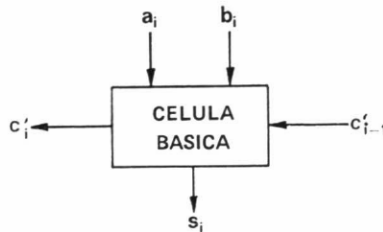


FIG. 10.18

10.7 SUMADOR DECIMAL PARALELO

Se va a diseñar un sumador decimal paralelo, usando el código *BCD*.

El diagrama de bloques del circuito se indica en la figura 10.19.

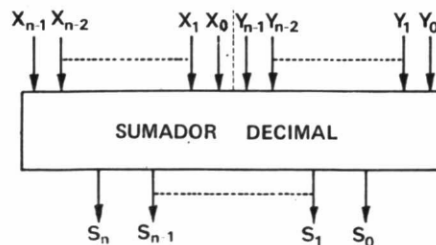


FIG. 10.19

Es decir, se pretende realizar la suma

$$S = A + B$$

donde

$$A = X_{n-1} X_{n-2} \dots X_1 X_0$$

$$B = Y_{n-1} Y_{n-2} \dots Y_1 Y_0$$

$$S = S_n S_{n-1} \dots S_1 S_0$$

pudiendo ser X_t , Y_t y S_t los números decimales 0, 1, 2, ..., 9.

Al emplear el procedimiento iterativo, se requiere diseñar la célula básica de la figura 10.20.

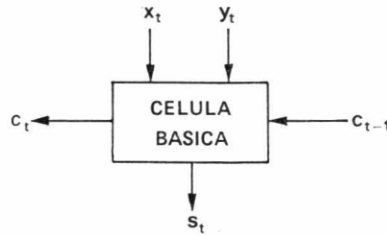


FIG. 10.20

El diseño de la célula básica parece sencillo, pero realmente no lo es, ya que X_t , Y_t , S_t , C_t y C_{t-1} son números decimales.

Al utilizar componentes de conmutación, es necesario codificar estas cinco variables en el sistema binario. En este diseño, el código utilizado es el *BCD*; en cuyo caso se tiene:

$$\left\{ \begin{array}{l} X_t = a_{3t} a_{2t} a_{1t} a_{0t} \\ Y_t = b_{3t} b_{2t} b_{1t} b_{0t} \\ S_t = u_{3t} u_{2t} u_{1t} u_{0t} \end{array} \right.$$

A continuación se van a estudiar dos aspectos:

- a) Análisis del código empleado.
- b) Problema del acarreo.

ANÁLISIS DEL CÓDIGO EMPLEADO

Se deben considerar dos casos:

Caso a) La suma $X_t + Y_t < 10$, por lo cual $C_t = 0$. Este caso no introduce complicación; por ejemplo, si X_t e Y_t son iguales, respectivamente, a 3 y 4, la suma 7 representa una combinación del código *BCD*:

$$\begin{array}{r} 3 \\ + 4 \\ \hline 7 \end{array} \qquad \begin{array}{r} 0011 \\ + 0100 \\ \hline 0111 \end{array}$$

Caso b) La suma $X_t + Y_t \geq 10$, por lo cual $C_t = 1$; es decir, es necesario generar la suma correcta y producir un acarreo para su envío a la siguiente etapa. Este caso puede dividirse en tres subcasos:

- b - 1) La suma $X_t + Y_t \geq 16$. Al realizar sumas binarias para procesar los códigos de los operandos, es necesario efectuar una corrección para obtener el acarreo y la suma correcta. Ejemplo:

$$\begin{array}{r}
 8 \qquad \qquad 1000 \\
 + 9 \qquad + \quad 1001 \\
 \hline
 17 \qquad \qquad 10001
 \end{array}$$
$$\begin{array}{r}
 0001 \\
 + 0110 \quad \leftarrow \text{Corrección} \\
 \hline
 0111 \quad \leftarrow \text{Suma correcta}
 \end{array}$$

- b - 2) La suma $X_t + Y_t \geq 12$, pero menor que 16. En este caso, también es necesario efectuar una corrección para obtener la suma correcta. Ejemplo:

$$\begin{array}{r}
 6 \qquad \qquad 0110 \\
 + 7 \qquad + \quad 0111 \\
 \hline
 13 \qquad \qquad 1101 \quad \leftarrow \text{Suma no correcta}
 \end{array}$$

$$\begin{array}{r}
 1101 \\
 + 0110 \quad \leftarrow \text{Corrección} \\
 \hline
 10011
 \end{array}$$

Obsérvese que si los bits de peso 8 y 4 de la suma son «unos», es necesario sumar al resultado el código 0110.

- b - 3) La suma $X_t + Y_t$ cumple: $10 \leq X_t + Y_t < 12$. Este caso es similar al b - 2). Ejemplo:

$$\begin{array}{r}
 6 \qquad \qquad 0110 \\
 + 5 \qquad + \quad 0101 \\
 \hline
 11 \qquad \qquad 1011 \quad \leftarrow \text{Suma no correcta}
 \end{array}$$

$$\begin{array}{r}
 1011 \\
 + 0110 \quad \leftarrow \text{Corrección} \\
 \hline
 10001
 \end{array}$$

Obsérvese que si los bits de peso 8 y 2 de la suma son «unos», es necesario sumar al resultado el código 0110.

El análisis anterior muestra que, si se emplea el código *BCD* para un sumador decimal, los dos códigos de los operandos se pueden sumar binariamente. Los cuatro bits resultantes representan el código de la suma de los dos operandos, siempre que no se presenten ninguna de estas tres condiciones:

- ocurre un «overflow»;
- los bits de peso 8 y 4 de la suma son «unos»;
- los bits de peso 8 y 2 de la suma son «unos»,

en cuyo caso se debe enviar un acarreo de 1 (decimal) a la siguiente célula y sumar el código 0110 a los cuatro bits resultantes para obtener el código que corresponde a la suma correcta.

PROBLEMA DEL ACARREO

El análisis anterior no indica el tratamiento de C_{t-1} . Si $C_{t-1} = 1$ (decimal), el código *BCD* correspondiente 0001 tiene que sumarse a los códigos de X_t e Y_t . Como puede verse, esto sólo afecta a la suma de a_{0t} y b_{0t} .

El diagrama de la célula básica es:

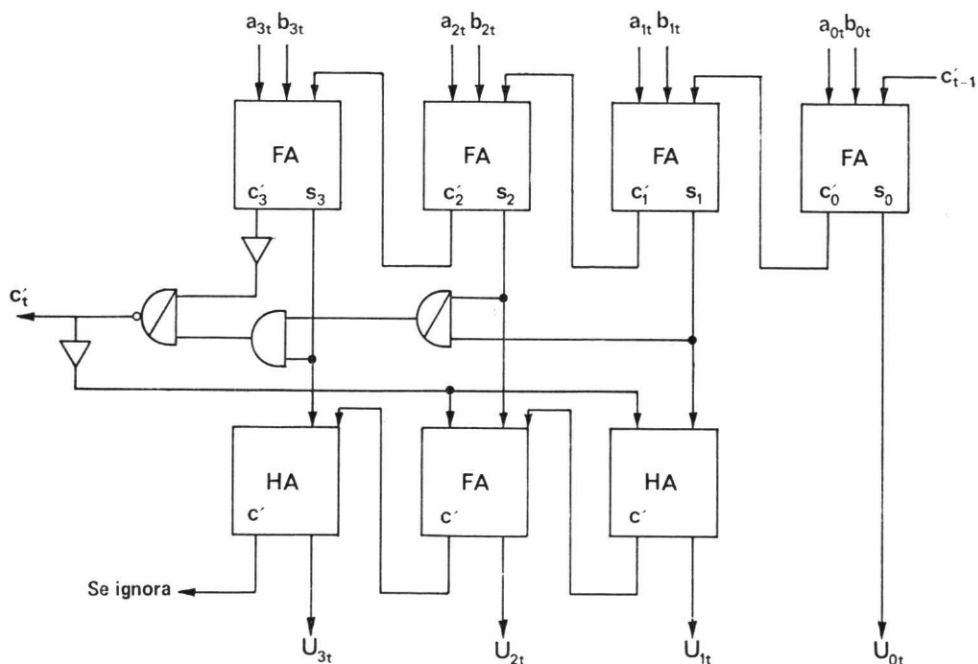


FIG. 10.21

Una desventaja de este circuito es el retardo con que se produce el acarreo, en algunos casos. Esto repercute en la célula básica siguiente.

Puede mejorarse el diseño mediante el empleo del código de exceso en 3, en lugar del código *BCD*.

BIBLIOGRAFIA

1. JACOB MILLMAN y CHRISTOS HALKIAS: *Dispositivos y circuitos electrónicos*. Anaya, 1971.
2. L. NASKELSKY: *Teoría de las calculadoras numéricas automáticas*. Editorial Alhambra, 1970.
3. H. C. TORNG: *Introduction to the logical desing of switching systems*. Addison, Wesley, 1964.
4. MARCOVITZ y PUGSLEY: *An introduction to the switching system design*. Wiley & Sons, 1971.
5. DEBRAINE: *Machines de traitement de l'information (Circuits et programmes)*. Tome I: *Ètude logique et construction des circuits*. Maisson et Cie., 1967.
6. MILLMAN-TAUB: *Circuitos de pulsos, digitales y de conmutación*. Ediciones del Castillo, 1969.
7. TAYLOR L. BOOTH: *Digital Networks and Computer Systems*. Wiley & Sons, 1971.
8. ENRIQUE MANDADO: *Sistemas electrónicos digitales*. Boixareu Editores, 1973.
9. MILLMAN y HALKIAS: *Integrated electronics*. McGraw-Hill Book Company, 1972.

Segunda parte
CIRCUITOS SECUENCIALES

SUMARIO

Página

SEGUNDA PARTE

CIRCUITOS SECUENCIALES

CAPITULO 1. INTRODUCCION:

1.1	Circuitos combinatorios	149
1.2	Circuitos secuenciales	150
1.3	Circuitos secuenciales síncronos	151
1.4	Circuitos secuenciales asíncronos.....	152

CAPITULO 2. MEMORIAS:

2.1	Memoria binaria («flip-flop»)	153
2.2	«Flip-flop» tipo T	154
2.3	«Flip-flop» tipo RS	155
2.4	«Flip-flop» tipo JK	157
2.5	«Flip-flop» tipo RST	158
2.6	«Flip-flop» tipo RT	160
2.7	«Flip-flop» tipo ST	161
2.8	Memoria transicional (<i>transition memory</i>)	161

CAPITULO 3. CIRCUITOS SECUENCIALES ASINCRONOS:

3.1	Diagrama de bloques	163
3.2	Diseño de circuitos secuenciales asíncronos	164
3.3	Matriz de salida	165
3.4	Matriz de direcciones o tabla de estados	165
3.5	Algunas tablas de estados	167
3.6	Matriz de excitaciones	168
3.6.1	Análisis de la matriz de excitaciones	169
3.6.2	Carreras (<i>racing problem</i>).....	170

	Página
3.7 Diagramas de estados (<i>state diagram o flow diagram</i>)	171
3.7.1 Diagrama de Mealy	171
3.7.2 Diagrama de Moore	173
3.8 Fusión de filas (<i>row merging</i>)	174
3.9 Diagrama de fusión (<i>merger diagram</i>)	177
3.10 Codificación de la matriz de fases.....	178
3.11 Obtención de las funciones que definen las excitaciones secundarias	180
3.12 El azar en los circuitos de conmutación	182
3.13 Resumen	183
3.14 Ejemplos	184
CAPITULO 4. CIRCUITOS SECUENCIALES SINCRONOS:	
4.1 Diagrama de bloques	191
4.2 Diseño de circuitos secuenciales sincronos	193
4.3 Estados equivalentes	194
4.4 Estados equivalentes representados por filas no idénticas	196
4.5 Método de tabulación para la reducción de tablas de estados	198
4.6 Diseño de la memoria	202
4.7 Diseño de contadores	203
4.7.1 Contador binario hexadecimal	203
4.7.2 Contador binario decimal	208
4.8 Diseño de un generador de bit de paridad	211
4.9 Diseño de un circuito secuencial sin estado inicial	220
4.10 Registro de desplazamiento (<i>shift register</i>)	225
4.11 Multiplicador binario	227
4.11.1 Generación de productos parciales	227
4.11.2 Diagrama de bloques de un multiplicador binario	228
4.12 Ejemplo	230
APÉNDICE: Símbolos lógicos utilizados	232
BIBLIOGRAFIA	233

CAPITULO 1
INTRODUCCION

1.1 CIRCUITOS COMBINATORIOS

Un circuito combinatorio es un circuito caracterizado por tener n variables binarias de entrada y m variables binarias de salida.

Un diagrama de bloque de este tipo de circuito se muestra en la figura 1.1.

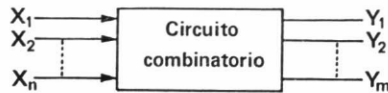


FIG. 1.1

En este tipo de circuitos no se considera explícitamente el papel del tiempo. Esto no implica que las entradas y salidas del sistema no sean función de esta variable; lo que ocurre es que las salidas en el instante t son función únicamente de las variables de entrada en el instante t .

Así, pues, las ecuaciones que definen el sistema son:

$$\begin{aligned} Y_1 &= f_1(X_1, X_2, \dots, X_n) \\ Y_2 &= f_2(X_1, X_2, \dots, X_n) \\ &\vdots \\ Y_m &= f_m(X_1, X_2, \dots, X_n) \end{aligned}$$

Las variables de entrada pueden tomar 2^n valores diferentes y las de salida 2^m . Los valores decimales de estas combinaciones binarias se suelen representar por X^i e Y^j . Es decir:

- X^i representa el valor decimal de cada una de las 2^n posibles combinaciones de entrada.
- Y^j representa el valor decimal de cada una de las 2^m combinaciones de salida que corresponden a las entradas X^i .

En un circuito combinatorio a cada valor de entrada X^i corresponde una combinación de salida Y^j ; lo recíproco no es cierto, ya que un mismo valor Y^j puede corresponder a diferentes valores de X^i .

1.2 CIRCUITOS SECUENCIALES

La diferencia fundamental entre circuitos combinatorios y secuenciales radica en la dependencia que estos últimos tienen de la variable tiempo. Las salidas presentes de un circuito secuencial están determinadas por las entradas presentes y por la secuencia de los valores anteriores de éstas. Esta información sobre la historia del circuito se almacena en una memoria y se designa como estado interno del circuito.

El diagrama de bloques de un circuito secuencial se muestra en la figura 1.2.

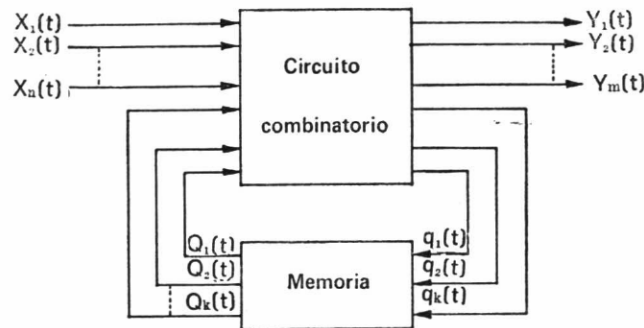


FIG. 1.2

Como se puede observar, el diagrama de bloque consta de dos partes:

- Una memoria de K salidas y K' entradas. Esta memoria está formada usualmente por elementos binarios de almacenamiento.
- Un circuito lógico combinatorio requerido para controlar la memoria y producir las salidas deseadas.

En la figura 1.2:

$X_i(t)$ representa las variables primarias o de entrada.

$Y_j(t)$ representa las variables de salida.

$q_i(t)$ representa las excitaciones secundarias $q_1(t), q_2(t), \dots, q_k(t)$. Define el próximo estado interno del circuito; es decir, el valor $Q(t + \tau)$.

$Q_s(t)$ representa las variables internas o secundarias $Q_1(t), Q_2(t), \dots, Q_{kl}(t)$. Define el estado interno presente del circuito.

El próximo valor de $Q_s(t)$ depende del valor presente de las variables de entrada y de la información presente almacenada; es decir:

$$Q_s(t + \tau) = f[X_i(t), Q_s(t)]$$

para $s = 1, 2, \dots, k$ e $i = 1, 2, \dots, n$.

Se puede describir el comportamiento del circuito mediante un sistema de $m + k'$ ecuaciones booleanas de $n + k$ variables.

$$Y_j = g [X_i(t), Q_s(t)] \quad \text{para } j = 1, 2, \dots, m$$

$$q_l = h [X_i(t), Q_s(t)] \quad \text{para } l = 1, 2, \dots, k'$$

Tradicionalmente se suelen considerar dos tipos de circuitos secuenciales; síncronos y asíncronos. A continuación se explican las características básicas de cada uno de ellos.

1.3 CIRCUITOS SECUENCIALES SINCRONOS

En estos sistemas, se aplican las entradas y se observan las salidas en instantes discretos de tiempo t_1, t_2, \dots , llamados instantes de muestreo. Los intervalos de tiempo $t_2 - t_1, t_3 - t_2, \dots$, pueden ser iguales o diferentes.

Las variables del sistema se muestran en instantes discretos de tiempo; por lo que éstas sólo pueden definirse en los instantes de muestreo t .

El comportamiento del sistema queda definido por las ecuaciones

$$Y_{jt} = F (X_{1t}, X_{2t}, \dots, X_{nt}; Q_{1t}, Q_{2t}, \dots, Q_{kt})$$

$$Q_s(t + 1) = G (X_{1t}, X_{2t}, \dots, X_{nt}; Q_{1t}, Q_{2t}, \dots, Q_{kt})$$

o abreviadamente:

$$Y_{jt} = F (X_{it}, Q_{st})$$

$$Q_s(t + 1) = G (X_{it}, Q_{st})$$

donde

- i : es un valor entero tal que $1 \leq i \leq n$.
- j : es un valor entero tal que $1 \leq j \leq m$.
- s : es un valor entero tal que $1 \leq s \leq k$.
- t : instante de tiempo presente.
- $t + 1$: instante de tiempo próximo.
- X_{it} : representa el valor de las n variables de entrada en el instante de muestreo t .
- Y_{jt} : representa el valor de las m variables de salida en el instante de muestreo t .
- Q_{st} : representa el valor de las k variables de estado (estado interno) en el instante de muestreo t .
- F y G : son las funciones que definen unívocamente la salida en el instante t y el estado interno en el instante $t + 1$.

Estudiaremos los sistemas cuyas señales están sincronizadas mediante *impulsos de reloj (clock pulses)*. Con las señales del reloj se pueden sincronizar niveles o impulsos.

El diagrama de bloques de un circuito secuencial síncrono puede verse en la figura 1.3.

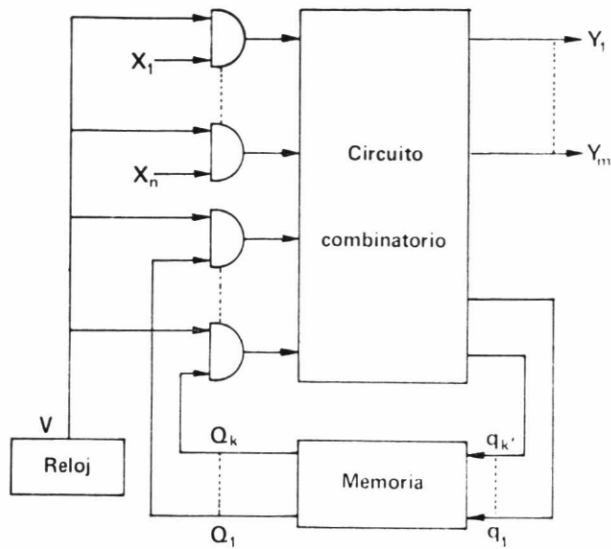


FIG. 1.3

El reloj es un generador de impulsos. Su salida es la representada en la figura 1.4.

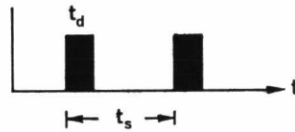


FIG. 1.4

Las puertas *AND* sirven para sincronizar las $n + k$ entradas con los impulsos del reloj.

1.4 CIRCUITOS SECUENCIALES ASINCRONOS

El diagrama de bloque de este tipo de circuito puede verse en la figura 1.2.

Son circuitos secuenciales que no utilizan impulsos de reloj para controlar los cambios de estados. En ellos es primordial tener en cuenta el orden en que sus variables cambian. Estas variables deben definirse continuamente, como se indica en la figura 1.5.

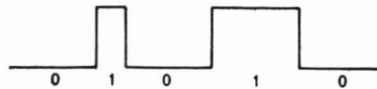


FIG. 1.5

CAPITULO 2
MEMORIAS

2.1 MEMORIA BINARIA («FLIP-FLOP»)

Es un circuito secuencial que posee dos estados estables (1) y es susceptible de cambiar de estado cuando es sometido a la acción de señales de control apropiadas.

Desde el punto de vista lógico, es un circuito que presenta las siguientes características:

- a) Sólo son posibles dos estados internos: 0 y 1.
- b) El número de entradas varía normalmente de 1 a 3.
- c) El número de salidas es, normalmente, dos: B_n y B'_n .
- d) Las señales de salida son complementarias: $B_n = (B'_n)'$.
- e) Por convenio, el estado del «flip-flop» es el que corresponde a su salida afirmativa.
- f) El estado interno del «flip-flop» se manifiesta en las señales de salida.
El estado presente de salida depende exclusivamente del estado interno en ese momento y éste, a su vez, de los estados de entrada e internos inmediatamente anteriores.
- g) La duración de los impulsos de entrada debe ser inferior al tiempo de transición de la memoria (2).
- h) Si todas las entradas son 0, el estado del «flip-flop» no cambia.

(1) El concepto de estado estable se introduce en el punto 3.4.

(2) Tiempo que tarda en alcanzar su nuevo estado.

Estudiaremos los siguientes tipos de «flip-flops»:

TIPO	Número de entradas	Ecuación característica
T	1	$B_{n+1} = B'_n T_n + B_n T'_n$
RS	2	$B_{n+1} = S_n + B_n R'_n$
JK	2	$B_{n+1} = B_n K'_n + B'_n J_n$
RST	3	$B_{n+1} = S_n + B'_n T_n + B_n R'_n T'_n$
RT	2	$B_{n+1} = B'_n T_n + B_n R'_n T'_n$
ST	2	$B_{n+1} = S_n + B'_n T_n + B_n T'_n$

2.2 «FLIP-FLOP» TIPO T

Un «flip-flop» tipo *T* (*trigger memory*) es un dispositivo equipado con una entrada (T_n) y dos salidas complementarias (B_n y B'_n). Cada vez que se aplica un impulso a la entrada del «flip-flop», éste cambia de estado y, por consiguiente, su salida cambia, permaneciendo en este estado hasta que un nuevo impulso de control sea aplicado.

En la figura 2.1 se muestra el diagrama de bloque del «flip-flop».

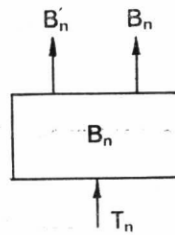


FIG. 2.1

La tabla de verdad de este dispositivo es la siguiente:

B_n	T_n	B_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

La ecuación característica que define el comportamiento de este elemento es:

$$B_{n+1} = B'_n T_n + B_n T'_n$$

2.3 «FLIP-FLOP» TIPO RS

Una memoria RS es un dispositivo que tiene dos entradas, llamadas R_n (Reset) y S_n (Set), y dos salidas, B_n y B'_n .

El comportamiento del «flip-flop» queda reflejado en las siguientes reglas:

- a) B_{n+1} es 1 si $S_n = 1$, y este estado se mantiene aun después de que S_n se haga 0.
- b) B_{n+1} es 0 si $R_n = 1$, y este estado se mantiene aun después de que R_n se haga 0.
- c) Para la combinación de variables de entrada $R_n = 1$ y $S_n = 1$, el comportamiento del «flip-flop» es indeterminado; es decir, no se puede, en general, predecir el próximo estado del «flip-flop». Esta indeterminación se debe a la peculiaridad del diseño electrónico del «flip-flop» tipo RS. El diseño de este circuito se realiza de forma que no aparezcan simultáneamente las señales $R_n = 1$ y $S_n = 1$.

El diagrama de bloque de este elemento se muestra en la figura 2.2.

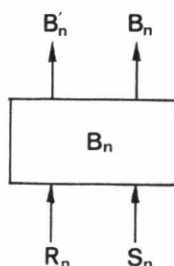
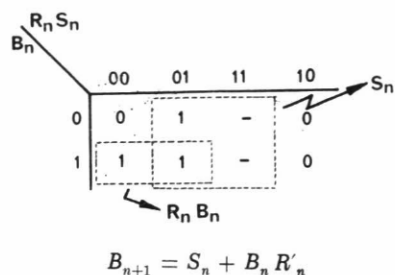


FIG. 2.2

De acuerdo con las reglas anteriores, se puede establecer la siguiente tabla de verdad:

B_n	R_n	S_n	B_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	—
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	—

De donde la matriz de excitaciones secundarias (1) y la ecuación característica del «flip-flop» resultan ser:



con la condición $R_n S_n = 0$.

Un método para construir un «flip-flop» tipo RS consiste en usar el circuito de la figura 2.3. La materialización de «flip-flops» RS se usa, a menudo, en la práctica, ya que muchos fabricantes suministran puertas NOR , pero no «flip-flops» tipo RS .

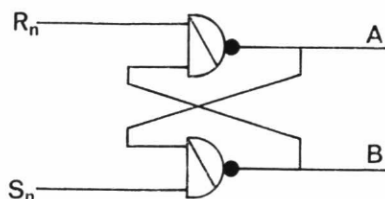


FIG. 2.3

Este «flip»flop» responde a niveles.

Las ecuaciones lógicas son:

$$A = (R_n + B)'$$

$$B = (S_n + A)'$$

Para $R_n = 0$ y $S_n = 1$ se tiene:

$$B = (1 + A)' = 0$$

$$A = (0 + B)' = 1$$

Del mismo modo, para $R_n = 1$ y $S_n = 0$:

$$A = (1 + B)' = 0$$

$$B = (0 + A)' = 1$$

Examinemos el comportamiento del circuito para $R_n = S_n = 0$:

$$A = (0 + B)' = 0'$$

$$B = (0 + A)' = A'$$

esto es, las salidas son diferentes, pero no hay especificación de la combinación de salida. Puede ser $A = 0, B = 1$ ó $A = 1, B = 0$. Como ocurre en la mayoría de los

(1) El concepto de matriz de excitaciones se introduce en el punto 3.6.

circuitos electrónicos, se cumple el principio de la mínima energía. Así, si la última entrada 1 fue R_n , entonces el circuito permanecerá en el estado $A = 0, B = 1$. Del mismo modo, si la última entrada 1 fue S_n , el circuito permanecerá en el estado $A = 1, B = 0$.

Finalmente, si $R_n = S_n = 1$, se obtiene:

$$A = (1 + B)' = 0$$

$$B = (1 + A)' = 0$$

con lo cual las dos salidas serían las mismas. Si ambas entradas R_n y S_n desaparecen simultáneamente, no se puede predecir, en general, el estado final del «flip-flop». Así, pues, esta combinación de entrada no es válida.

Las entradas de este «flip-flop» pueden controlarse con puertas *AND*, tal como se indica en la figura 2.4.

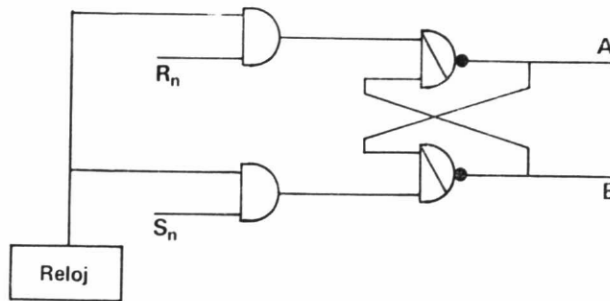


FIG. 2.4

En este último caso, las entradas R_n y S_n se activarán en los intervalos en que existen impulsos de reloj. Estos impulsos deben ser bastante estrechos para que las salidas del «flip-flop» no cambien hasta que hayan desaparecido los impulsos de sincronización; de otra forma, múltiples cambios pueden ocurrir con un solo impulso.

2.4 «FLIP-FLOP» TIPO *JK*

Este tipo, muy común, se comporta de acuerdo con la siguiente tabla de verdad:

B_n	K_n	J_n	B_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

El diagrama de bloque del dispositivo se muestra en la figura 2.5.

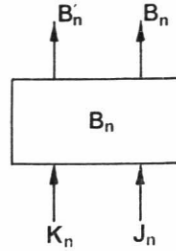
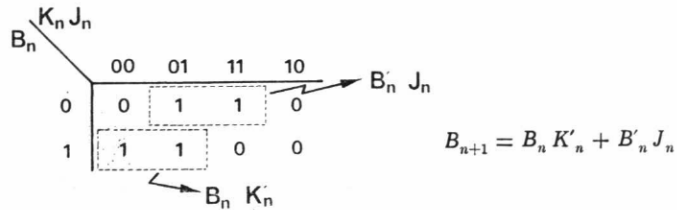


FIG. 2.5

Comparando las tablas de verdad de los «flip-flops» *RS* y *JK*, se observan tres hechos importantes:

1. El comportamiento del «flip-flop» *JK* está definido para todas las posibles combinaciones de entrada.
2. La presencia simultánea de $J_n = 1$ y $K_n = 1$ provoca siempre una transición.
3. La entrada J_n es una entrada «set» y la entrada K_n es una entrada «reset». Por esta causa sería más lógico denominar a este dispositivo «flip-flop» tipo *KJ*.

La matriz de excitaciones secundarias y la ecuación característica del «flip-flop» son:



no existiendo, en este caso, ninguna condición adicional.

2.5 «FLIP-FLOP» TIPO *RST*

Es un dispositivo que tiene tres entradas, llamadas *R* (*Reset*), *S* (*Set*) y *T* (*Trigger*), y dos salidas.

Para este «flip-flop» existe la restricción de que no puede presentarse simultáneamente en las entradas más de un 1.

El diagrama de bloque de este elemento es:

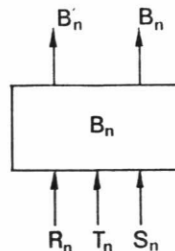
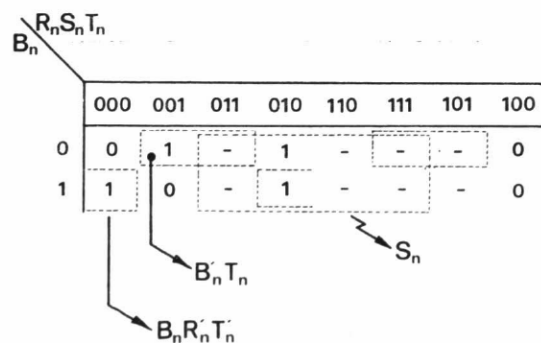


FIG. 2.6

El comportamiento del «flip-flop» se refleja en la siguiente tabla de verdad:

B_n	R_n	S_n	T_n	B_{n+1}
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	—
0	1	0	0	0
0	1	0	1	—
0	1	1	0	—
0	1	1	1	—
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	—
1	1	0	0	0
1	1	0	1	—
1	1	1	0	—
1	1	1	1	—

La matriz de excitaciones secundarias y la ecuación característica del «flip-flop» son:



$$B_{n+1} = S_n + B'_n T_n + B_n R'_n T'_n$$

con la condición

$$R_n S_n + R_n T_n + S_n T_n = 0$$

En cierto modo, el «flip-flop» JK puede hacer lo mismo que el «flip-flop» RST , ya que:

- a) $K_n J_n = 11$ tiene el mismo efecto que $T_n = 1$.
- b) $K_n J_n = 10$ tiene el mismo efecto que $R_n S_n T_n = 100$.
- c) $K_n J_n = 01$ tiene el mismo efecto que $R_n S_n T_n = 010$.

Cualquier «flip-flop» RST puede usarse como «flip-flop» RS sin más que hacer $T = 0$. También se puede usar como «flip-flop» T haciendo $R = S = 0$.

2.6 «FLIP-FLOP» TIPO RT

Resulta de suprimir la entrada S_n en un «flip-flop» tipo RST .

La tabla de verdad, matriz de excitaciones secundarias y ecuación característica de este «flip-flop» son:

B_n	R_n	T_n	B_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	—
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	—

		$R_n T_n$			
B_n		00	01	11	10
0		0	1	-	0
1		1	0	-	0

$B'_n T_n$

$B_n R'_n T'_n$

$$B_{n+1} = B'_n T_n + B_n R'_n T'_n$$

con la condición

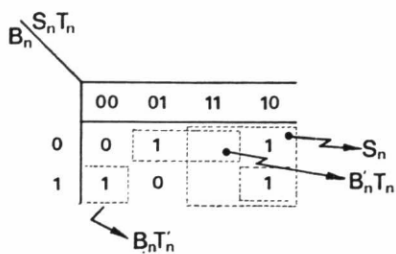
$$R_n T_n = 0$$

2.7 «FLIP-FLOP» TIPO ST

Resulta de suprimir la entrada R_n en un «flip-flop» tipo RST .

La tabla de verdad, matriz de excitaciones secundarias y ecuación característica de este elemento son:

B_n	S_n	T_n	B_{n+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	—
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	—



$$B_{n+1} = S_n + B'_n T_n + B_n T'_n$$

con la condición

$$S_n T_n = 0$$

2.8 MEMORIA TRANSICIONAL («TRANSITION MEMORY»)

Es una línea de retardo (1), elemento normalmente pasivo con una entrada y una salida, cuyo diagrama de bloque es:

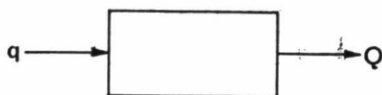


FIG. 2.7

(1) Puede estar constituida por cualquier elemento que transmita una señal con un cierto retraso; por ejemplo: relé electromagnético, célula de retardo, etc.

Si la excitación es q , la salida adquiere el mismo valor que la entrada, al cabo de un intervalo de tiempo τ .

La ecuación que define el próximo estado de este circuito es:

$$Q_{n+1} = q_n$$

Nótese que, para este elemento, el próximo estado es realmente independiente del estado presente. La línea de retardo debe operar en un sistema síncrono con un período de los impulsos de reloj exactamente igual al retraso. La línea de retardo debe tener una entrada con cada impulso de reloj; de otro modo la salida desaparecerá.

CAPITULO 3

CIRCUITOS SECUENCIALES ASINCRONOS

3.1 DIAGRAMA DE BLOQUES

Ya se ha indicado anteriormente en la figura 1.2 el diagrama de bloques de un circuito secuencial asíncrono.

El diseño de un circuito de este tipo implica:

- a) Diseño de la memoria.
- b) Diseño del circuito combinatorio correspondiente.

En esta clase de circuitos no existe reloj, por lo cual la memoria puede ser diseñada a base de componentes de conmutación sencillos.

En la figura 3.1 se muestra el diagrama de bloques de un circuitos secuencial asíncrono de relés.

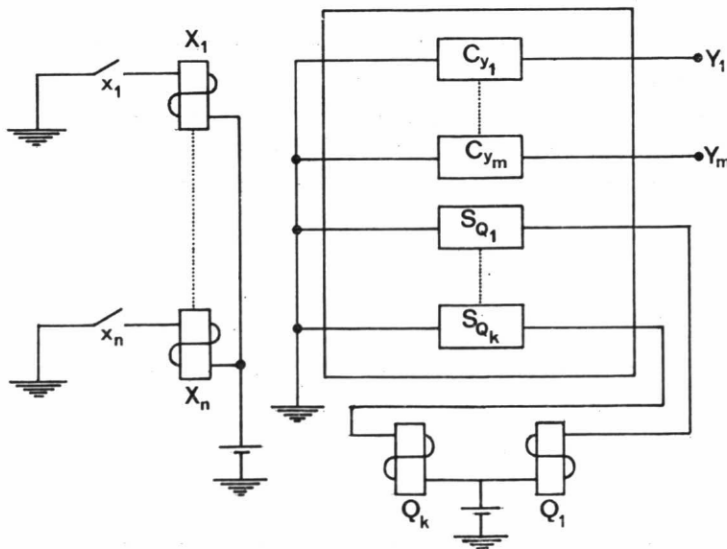


FIG. 3.1

Como puede verse en la figura 3.1:

- a) Variables primarias: son los relés X . Una entrada es 1 si su correspondiente contacto está cerrado; en caso contrario se dice que la entrada es cero.
- b) Variables secundarias: son los relés Q . Cada bobina Q se emplea para almacenar un dígito binario. Si una bobina está excitada, se interpreta como que tiene almacenado un 1; en caso contrario tiene almacenado un cero.

Los contactos de los relés X («relés primarios») y Q («relés secundarios») se usan:

- a) Para la realización de los circuitos $C_{y_1}, C_{y_2}, \dots, C_{y_m}$, que originan las salidas y_1, y_2, \dots, y_m , respectivamente.
- b) Para la realización de los circuitos $S_{Q_1}, S_{Q_2}, \dots, S_{Q_k}$, que controlan las bobinas de los relés Q .

3.2 DISEÑO DE CIRCUITOS SECUENCIALES ASINCRONOS

A continuación se indica un procedimiento sistemático para llevar a cabo el diseño de un circuito secuencial asíncrono. Se parte de la especificación del problema, en forma proposicional, con lo cual el proceso a seguir puede dividirse en las siguientes etapas:

1. Obtención de un diagrama de estados.
2. Obtención de una tabla de estados.
3. Reducción del número de filas de la tabla de estados; para ello es necesario identificar los llamados estados equivalentes.
4. Codificación de los estados internos que aparecen en la tabla de estados reducida; para ello se asigna a cada fila una combinación binaria de las variables secundarias. Para evitar el problema de las carreras (*racing problem*), que estudiaremos más adelante, se limitan las transiciones a cambios entre estados representados por combinaciones adyacentes.
5. Obtención de la matriz de excitaciones secundarias y matriz de salida.
6. Obtención de las funciones de excitación y de salida a partir de las matrices correspondientes.
7. Estudio de los fenómenos de azar, en el caso de emplear relés para el diseño del circuito.
8. Materialización del circuito.

En los apartados que siguen se altera este orden por razones didácticas, estudiándose en primer lugar la matriz de salida y las tablas de estados.

3.3 MATRIZ DE SALIDA

Es una matriz que relaciona las variables primarias y los estados internos presentes con las salidas del sistema.

Se suele representar de la siguiente forma:

$Q \setminus X$	X^1	X^2	X^i	X^N
Q^1	Y^{11}	Y^{12}	Y^{1i}	Y^{1N}
Q^2	Y^{21}	Y^{22}	Y^{2i}	Y^{2N}
.....
Q^k	Y^{k1}	Y^{k2}	Y^{ki}	Y^{kN}

En esta matriz:

X^i : representa el equivalente decimal de cada una de las 2^n posibles combinaciones de las n variables primarias.

Q^j : representa el equivalente decimal de cada una de las 2^k posibles combinaciones de las k variables de estado.

Y^{ij} : representa el equivalente decimal de la combinación de las m variables de salida. El número total de combinaciones puede ser menor que 2^m .

$N = 2^n$.

$K = 2^k$.

Las variables de salida (Y) dependen, a veces, solamente de las variables secundarias, con lo cual la matriz de salida se reduce a una sola columna. En este caso hay sólo K salidas de las 2^m combinaciones posibles.

El movimiento en sentido horizontal implica un cambio en las variables primarias. El movimiento en sentido vertical implica un cambio en las variables secundarias.

3.4 MATRIZ DE DIRECCIONES O TABLA DE ESTADOS

Es una matriz que relaciona las variables primarias y estados internos presentes con los estados futuros.

Se suele representar de la siguiente forma:

$Q \setminus X$	X^1	X^2	X^i	X^N
Q^1	q^{11}	q^{12}	q^{1i}	q^{1N}
Q^2	q^{21}	q^{22}	q^{2i}	q^{2N}
.....
Q^k	q^{k1}	q^{k2}	q^{ki}	q^{kN}

En esta matriz:

X^i : representa el equivalente decimal de cada una de las 2^n posibles combinaciones de las n variables primarias.

Q^j : representa el equivalente decimal de cada una de las 2^k posibles combinaciones de las k variables de estado.

q^{ij} : representa el equivalente decimal de la próxima combinación de las k excitaciones secundarias (indica el estado final).

Se llaman *estados totales* a los estados definidos por el conjunto de las variables primarias y secundarias. Existen tantos estados totales como elementos posee la tabla de estados; es decir, $2^n \times 2^k$ estados.

Se llaman estados internos o secundarios a los estados que dependen solamente de las variables secundarias. Hay tantos estados internos como filas en la tabla de estados; es decir, 2^k estados.

Las *excitaciones secundarias* (q) son siempre función de las variables primarias y secundarias.

En la tabla de estados siguiente se muestran, como ejemplo, unas matrices de direcciones y de salida para el caso $n = 1, m = 2, k = 2$.

Q^x	0		1	
	0	1	0	1
0	①	1	0	3
1	2	3	1	2
2	②	3	1	2
3	0	③	0	2

q
y

Un estado total es ESTABLE si tiene el mismo número de orden que el de la fila en que está situado; en caso contrario, se dice que el estado es INESTABLE. Los estados estables se representan encerrándolos en un círculo.

Una notación que puede utilizarse para indicar los estados del sistema es:

$$(a, b) (c)$$

donde

- a) representa la combinación de las variables primarias presentes;
- b) representa la combinación de las variables secundarias presentes;
- c) representa la combinación que corresponde al próximo estado del circuito.

Cuando b y c coinciden el estado es estable; en caso contrario, el estado es inestable; es decir:

- a) Si $Q(t) = Q(t + \tau)$, el estado es estable, y
- b) Si $Q(t) \neq Q(t + \tau)$, el estado es inestable.

La tabla anterior define completamente el sistema. Supongamos que partimos del estado estable (0, 0) (0) y que cambiamos el valor de la variable de entrada con lo que este nuevo estado pasa a (1, 0) (1), que es inestable.

Al cabo de un tiempo Δ_1 pasamos al nuevo estado inestable (1, 1) (3). Al cabo de Δ_2 pasamos al estado (1, 3) (3), que es estable.

Los tiempos Δ_1 y Δ_2 no suelen ser iguales, pues dependen del tiempo que tarden las señales en propagarse a través de la memoria.

3.5 ALGUNAS TABLAS DE ESTADOS

A continuación se presentan dos tablas de estados extensamente utilizadas:

a) *Matriz primitiva de fases (primitive flow table)*.—Esta matriz se obtiene del diagrama de estados de Moore, que estudiaremos más adelante. Se caracteriza por presentar un solo estado estable por fila.

Ejemplo:

Estado presente	Próximo estado				Salida presente
	00	01	11	10	
1	①	6	3	2	0
2	4	②	4	3	1
3	③	4	-	2	0
4	④	5	-	6	1
5	-	3	⑤	6	1
6	4	⑥	4	-	1
7	4	-	⑦	3	0

La matriz primitiva de fases, como puede verse en el ejemplo anterior, viene siempre relacionada con la matriz de salida correspondiente.

Al reducir el número de filas de una matriz primitiva de fases utilizando los métodos que oportunamente se explicarán, se obtiene una matriz de fases (*flow table*).

En dicha matriz aparecen numerados individualmente los estados totales estables y los inestables con el mismo número, sin círculo, que el del estado estable al que conducen.

Ejemplo:

Q_1, Q_2	X_1, X_2			
	0	1	3	2
0	①	4	5	⑧
1	②	③	6	7
3	2	④	⑤	8
2	1	3	⑥	⑦

En la tabla anterior se han encerrado en un círculo cada uno de los seis estados estables existentes y se han representado con flechas las transiciones entre estados.

Para representar los resultados del análisis de la matriz de excitaciones se puede utilizar un procedimiento gráfico conocido como *gráfico de Moore*. Los estados del sistema, las combinaciones de variables aplicadas y las transiciones entre estados se indican tal como se muestra en la figura 3.2.

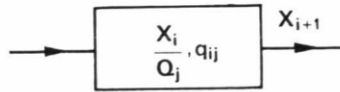


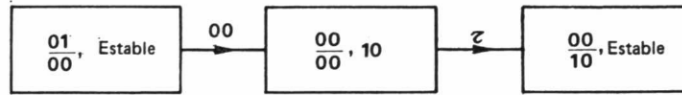
FIG. 3.2

donde

- X_i representa el equivalente binario de la combinación de las variables primarias.
- Q_j representa el equivalente binario de la combinación de las variables de estado.
- q_{ij} representa el equivalente binario de la próxima combinación de las variables de estado.
- X_{i+1} representa el equivalente binario de la nueva combinación de las variables primarias.

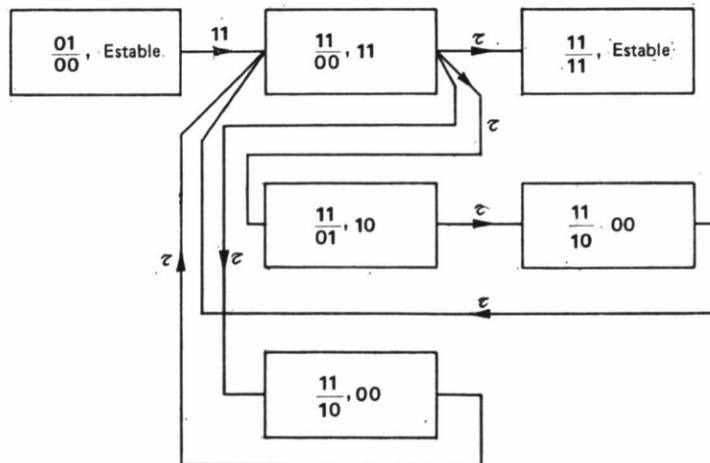
3.6.1 Análisis de la matriz de excitaciones

1.º La evolución del sistema al partir del estado estable (01, 00) (00) y modificar la combinación de las variables primarias para que correspondan a 00 es la siguiente:



Partiendo del estado estable (01, 00) (00), se alcanza al cabo de un tiempo τ el nuevo estado estable (00, 10) (10).

2.º Sea 11 la nueva combinación de entrada:



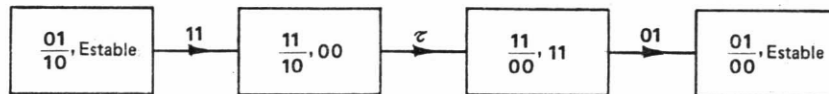
En este caso varían simultáneamente las dos excitaciones secundarias q_1 y q_2 . Si se trata de un circuito de relés, esto es poco probable. Parece más verosímil que un relé lo haga antes que el otro (más adelante se volverá a insistir sobre este hecho). Hay que considerar tres casos:

a) Cambian simultáneamente q_1 y q_2 . Se pasa de un estado estable a otro estable a través de una transición.

b) Cambia en primer lugar q_2 . Se recorre un ciclo automantenido que comprende dos estados de transición (*Buzzer circle*), ya que partiendo de un estado inestable no se alcanza ningún estado estable y se vuelve al punto de partida, repitiéndose indefinidamente el proceso.

c) Cambia en primer lugar q_1 . Se recorre un ciclo automantenido con un estado de transición.

3.º A continuación se estudia cómo partiendo del estado estable (01, 10) (10) se puede alcanzar el estado estable (01, 00) (00).



Se modifican las variables de entrada para tener la combinación 01, tan pronto se llegue al estado inestable (11, 00) (11).

3.6.2 Carreras («racing problem»)

Sea el relé de la figura 3.3.

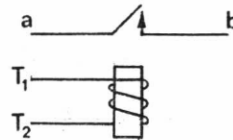


FIG. 3.3

En el instante $t = t_0$ se aplica un voltaje constante a los terminales T_1 y T_2 de la bobina.

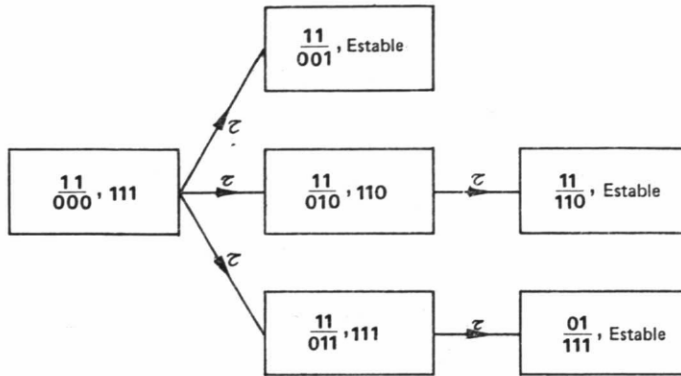
El contacto no se cerrará en el instante $t = t_0$, sino en $t = t_0 + \Delta$; es decir, existe un retraso entre el instante de aplicación del voltaje a los terminales de la bobina del relé y la apertura o cierre de los contactos del mismo.

No hay ninguna razón para admitir que el retraso Δ pudiera ser el mismo en todos los relés. Este hecho ocasiona, en los circuitos secuenciales asíncronos, el fenómeno denominado carreras.

Las carreras que se producen en los circuitos secuenciales asíncronos, ocasionadas por el hecho de requerirse el cambio simultáneo de varias excitaciones secundarias, son de tres tipos: críticas, seguras y bucles automantendidos.

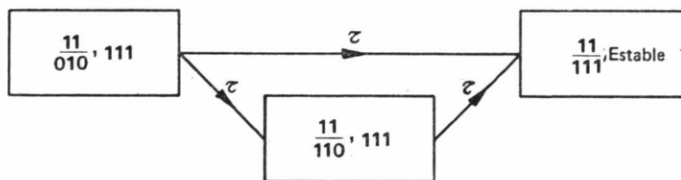
Las carreras *críticas* (*critical races*) pueden llevar el circuito a distintos estados estables, según las variables secundarias que cambien en primer lugar.

Ejemplo:



La carrera es *segura* (*safe race*) cuando el estado final alcanzado es siempre el mismo.

Ejemplo:



Los bucles automantenidos (*Buzzer circle*) ya se han explicado anteriormente en el punto 3.6.1.

3.7 DIAGRAMAS DE ESTADOS («STATE DIAGRAM» O «FLOW DIAGRAM»)

El primer paso en el proceso de diseño de un circuito es la obtención de un diagrama de estados a partir de las especificaciones del problema.

A continuación se estudian los diagramas de Mealy y de Moore.

3.7.1 Diagrama de Mealy

Este diagrama, obtenido a partir de las especificaciones de diseño del circuito, se ajusta a las siguientes reglas:

1. Cada estado interno general del sistema se representa por una letra, normalmente mayúscula, o por un número. Este estado interno general puede corresponder a uno o más estados estables.

2. A cada letra o número se asigna un rectángulo o una circunferencia, que representa uno o más estados estables, de acuerdo con el número de lazos que lleve asociados.
3. Las transiciones entre estados se representan mediante flechas. Con cada flecha se asocia:
 - a) La combinación de las variables primarias, y
 - b) La próxima combinación de las variables de salida correspondiente.
4. Los *estados iniciales* se representan mediante dos rectángulos o dos circunferencias concéntricas, y se caracterizan porque una vez que el sistema ha dejado uno de ellos, ya no puede volver al mismo. Las flechas se alejan de ellos.
5. *Estado final* es el que sólo recibe transiciones dirigidas hacia él. No se puede abandonar una vez alcanzado.
6. Normalmente, se admite que sólo una variable primaria puede cambiar a la vez. (Circuitos asíncronos de relés.)

Un diagrama que puede presentarse como ejemplo se muestra en la figura 3.4.

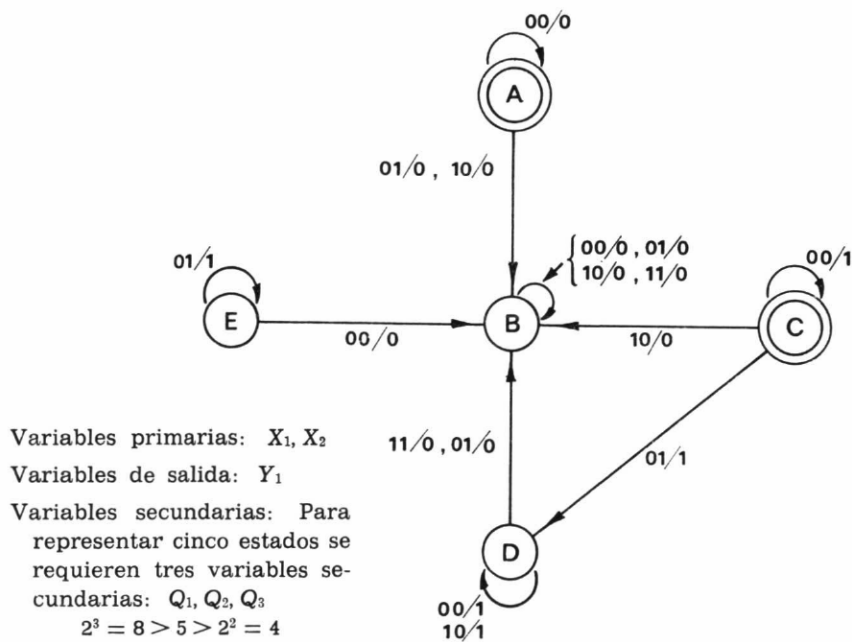


FIG. 3.4

Los resultados del diagrama de Mealy se pueden representar en una tabla de transición de estados en la que se considera que sólo una variable de entrada puede cambiar cada vez.

Las combinaciones de entrada que no pueden presentarse en estas condiciones dan lugar a condiciones opcionales.

Estado presente	Próximo estado				Próxima salida			
	00	01	11	10	00	01	11	10
A	(A)	B	-	B	0	0	-	0
B	(B)	(B)	(B)	(B)	0	0	0	0
C	(C)	D	-	B	1	1	-	0
D	(D)	B	B	(D)	1	0	0	1
E	B	(E)	-	-	0	1	-	-

3.7.2 Diagrama de Moore

Las diferencias existentes entre este diagrama y el de Mealy son:

1. Cada estado estable del sistema se representa por una letra o por un número.
2. Se asigna una circunferencia o un rectángulo a cada estado estable.
3. Cada estado interno estable se asocia con su salida presente correspondiente, no con las transiciones.

Un diagrama que puede presentarse como ejemplo se muestra en la figura 3.5.

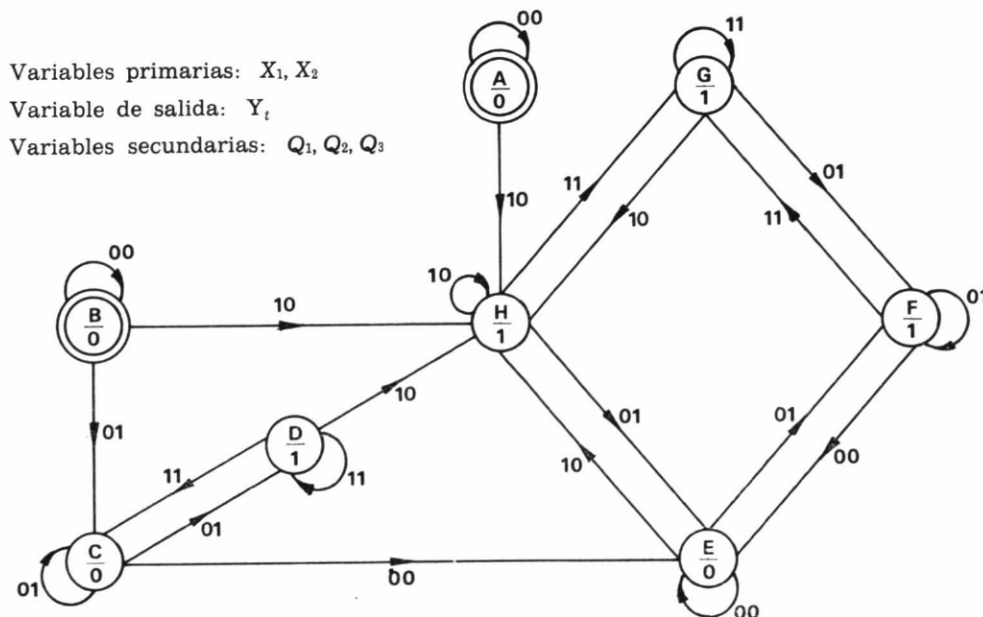


FIG. 3.5

Los resultados del diagrama de Moore se pueden representar en una matriz primitiva de fases, en la que se considera que sólo una variable de entrada puede cambiar cada vez.

Estado presente	Próximo estado				Salida presente
	00	01	11	10	
A	Ⓐ	-	-	H	0
B	Ⓑ	C	-	H	0
C	E	Ⓒ	D	-	0
D	-	C	Ⓓ	H	1
E	Ⓔ	F	-	H	0
F	E	Ⓕ	G	-	1
G	-	F	Ⓖ	H	1
H	-	E	G	Ⓗ	1

3.8 FUSION DE FILAS («ROW MERGING») (1)

Consiste en combinar las entradas de dos o más filas de una matriz primitiva de fases para formar una sola fila.

La única condición que limita estas fusiones es que para un observador externo al sistema no se detecte ningún cambio en el mismo.

Se aplicarán las dos reglas siguientes:

1. Dos filas de una matriz pueden fusionarse si los próximos estados son iguales en las mismas columnas, o si una de las filas tiene un estado sin especificar en la columna correspondiente.
2. Cuando uno de los próximos estados, de igual número o letra, tiene una circunferencia, el estado resultante también la lleva.

Ejemplo: Sea la matriz primitiva de fases siguiente:

Estado presente	Próximo estado				Salida presente
	00	01	11	10	
A	Ⓐ	B	-	F	0
B	A	Ⓑ	D	-	1
C	Ⓒ	B	-	-	1
D	-	B	Ⓓ	E	0
E	C	-	D	Ⓔ	1
F	A	-	D	Ⓕ	0

(1) En el capítulo dedicado a circuitos secuenciales síncronos se explican varios métodos de simplificación de tablas de estado.

En la matriz primitiva de fases anterior se pueden fusionar las filas que se indican a continuación:

Estados	Filas fusionables	Resultado
AyB	(A) B - F A (B) D -	(A) (B) D F
AyF	(A) E - F A - D (F)	(A) B D (F)
ByD	A (B) D - - B (D) E	A (B) (D) E
ByF	A (B) D - A - D (F)	A (B) D (F)
CyD	(C) B - - - B (D) E	(C) B (D) E
CyE	(C) B - - C - D (E)	(C) B D (E)
DyE	- B (D) E C - D (E)	C B (D) (E)

No es necesario construir la tabla anterior para expresar las diferentes posibilidades de fusión. Para ello puede usarse el método siguiente:

Fila	Fusionable con	Parejas de agrupamiento
A	B, F	(A, B), (A, F)
B	D, F	(B, D), (B, F)
C	D, E	(C, D), (C, E)
D	E	(D, E)
E	-	-

Las parejas (A, B), (A, F), (B, F) sugieren la formación del agrupamiento (A, E, F). Lo mismo ocurre con (C, D), (C, E), (D, E), que pueden ser agrupadas en la forma (C, D, E).

Como se observa, existen varias posibilidades de fusión. Entre otros, se pueden formar los siguientes agrupamientos:

- a) (A, B), (C, D), (E), (F)
- b) (A, B, F), (C, D, E)
- c) (A), (B, D), (C, E), (F)
- d) (A, F), (B, D), (C, E)

En las distintas soluciones indicadas para reducir el número de filas de la matriz primitiva de fases, puede observarse que, hasta ahora, no se han tenido en cuenta las *salidas del circuito secuencial*, ya que en la práctica aquéllas se observan cuando éste se encuentra en estado estable.

A continuación se muestra el resultado de la reducción de la matriz primitiva de fases para la solución b).

Estado presente	Próximo estado				Próxima salida			
	00	01	11	10	00	01	11	10
A, B, F	(A)	(B)	D	(F)	0	1	0	0
C, D, E	(C)	B	(D)	(E)	1	1	0	1

Se ha obtenido una matriz de fases de dos filas; pero la matriz de salida se ha complicado, ya que por haber fusionado filas con diferentes salidas, el sistema ya no es representable gráficamente por un diagrama de Moore, requiriéndose para este fin un diagrama de Mealy.

Para obtener una *matriz de salida más simplificada* es conveniente emplear las siguientes normas:

1. Generalmente, es preferible fusionar sólo las filas que tengan las mismas salidas, pues resultan funciones de salida que sólo dependen de las variables secundarias.
2. No interesa reducir el número de filas a un número inferior a la menor potencia posible de dos (ejemplo: si hay que elegir entre dos fusiones que llevan a una matriz de fases de tres o cuatro filas, se elegirá la solución de cuatro filas. La matriz obtenida contiene un mayor número de *condiciones opcionales*, lo que facilita la codificación de las variables secundarias).

Para evaluar la importancia de la reducción del número de filas, conviene considerar la función que define el coste del circuito:

$$C_t = \sum_{i=1}^m C_{Y_i} + \sum_{j=1}^{k'} C_{Q_j} + k M$$

donde

C_t = coste total del circuito.

C_{Y_i} = coste de las funciones de salida.

C_{Q_j} = coste de las funciones internas.

M = coste de la unidad de memoria.

Se admite:

$$C_t \simeq M k$$

por lo cual se intentará minimizar el número de memorias. Para ello se procurará obtener una matriz de fases con el menor número de filas.

3.9 DIAGRAMA DE FUSION («MERGER DIAGRAM»)

Se llama también diagrama de Cadwell.

Cada fila de la matriz primitiva de fases se representa por un punto al que se asigna el número o letra correspondiente de la fila de la matriz.

Cuando dos filas pueden fusionarse, se traza una línea entre los dos puntos correspondientes del diagrama. Esta línea es:

- a) Continua, si tienen la misma combinación de salida.
- b) Discontinua, en caso contrario.

Ejemplo: A partir de la matriz primitiva de fases utilizada en el punto 3.8 se obtiene el diagrama de fusión que se indica en la figura 3.6.

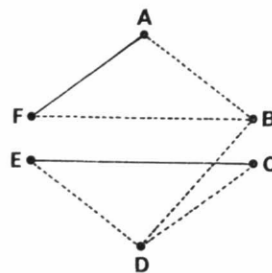


FIG. 3.6

Dos posibles soluciones se muestran en la figura 3.7.

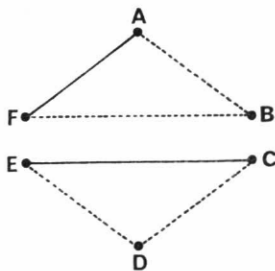


FIG. 3.7 (a)

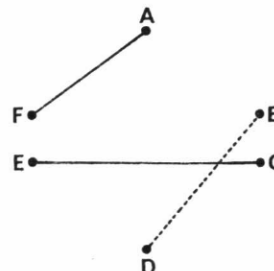


FIG. 3.7 (b)

Las matrices que corresponden a estas soluciones son:

Estado presente	Próximo estado				Próxima salida			
	00	01	11	10	00	01	11	10
A, B, F	(A)	(B)	D	(F)	0	1	0	0
C, D, E	(C)	B	(D)	(E)	1	1	0	1

Estado presente	Próximo estado				Próxima salida			
	00	01	11	10	00	01	11	10
A, F	(A)	B	D	(F)	0	1	0	0
B, D	A	(B)	(D)	E	0	1	0	1
C, E	(C)	B	D	(E)	1	1	0	1

Como puede observarse, la matriz de salida depende de las variables primarias y secundarias.

3.10 CODIFICACION DE LA MATRIZ DE FASES

Sea la siguiente matriz primitiva de fases:

Estado presente	Próximo estado	
	0	1
0	0	1
1	2	1
2	2	3
3	4	3
4	4	5
5	6	5
6	6	7
7	0	7

El análisis de esta matriz indica que no puede ser reducida.

Para materializar con relés la matriz de fases hay que decidir en primer lugar el número de relés secundarios que van a ser usados.

Cada fila tiene que ser representada por una combinación específica de las condiciones de operación de los relés secundarios. Se necesitan, por lo menos, tres relés.

La asignación de combinaciones a las filas de la matriz de fases, constituye el problema de *asignación secundaria*.

Un primer intento es el siguiente:

Fila	Q ₁	Q ₂	Q ₃
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Cada combinación es el equivalente binario de su correspondiente designación de fila. Cada representación indica las condiciones presentes de operación de los relés secundarios.

Para completar este método, a cada estado se le asigna el mismo código que tiene la fila en la cual es estable.

Se sustituye el esquema de asignación en la matriz de fases, con lo cual se obtiene una matriz de excitaciones.

Q ₁ Q ₂ Q ₃	X	
	0	1
000	000	001
001	010	001
010	010	011
011	100	011
100	100	101
101	110	101
110	110	111
111	000	111

El diseñador debe evitar la creación de carreras críticas. Para alcanzar este requerimiento, la transición entre estados estables se efectúa cuando solamente uno de los relés secundarios cambia en un instante determinado de tiempo. En caso contrario, se impone que las carreras que existan no sean críticas, sino seguras.

La codificación elegida crea carreras críticas; para evitar este problema se realiza una nueva asignación.

Fila	Q ₁	Q ₂	Q ₃
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	1	1
6	1	0	1
7	1	0	0

Se trata de un código continuo y cíclico, con el que resulta la matriz de excitaciones siguiente:

Q ₁ Q ₂ Q ₃	X	
	0	1
000	000	001
001	011	001
011	011	010
010	110	010
110	110	111
111	101	111
101	101	100
100	000	100

Un análisis de esta matriz indica:

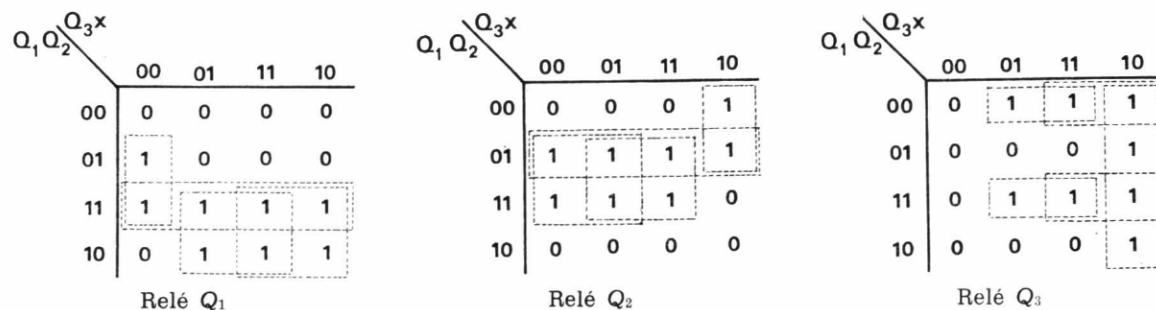
1. Que no existen carreras críticas.
2. Que cualquier transición entre *dos estados estables* originará un cambio de estado en uno solo de los tres relés secundarios.

Por último recuérdese que:

- a) La codificación que se elija debe evitar las carreras críticas con un mínimo de transiciones entre estados estables.
- b) La velocidad de respuesta del circuito depende del número de transiciones.
- c) La codificación elegida influye en la complejidad de las funciones de salida.

3.11 OBTENCIÓN DE LAS FUNCIONES QUE DEFINEN LAS EXCITACIONES SECUNDARIAS

A partir de la última matriz de excitaciones, obtenida en el punto 3.10, se pueden establecer los siguientes mapas:



de donde las excitaciones secundarias resultan ser:

$$q_1 = Q_2 Q_3' x' + Q_1 x + Q_1 Q_3$$

$$q_2 = Q_1' Q_3 x' + Q_2 x + Q_2 Q_3'$$

$$q_3 = Q_3 x' + Q_1 Q_2 x + Q_1' Q_2' x$$

Antes de seguir adelante se va a explicar la notación que emplearemos; para ello conviene tener en cuenta el diagrama de la figura 3.8.

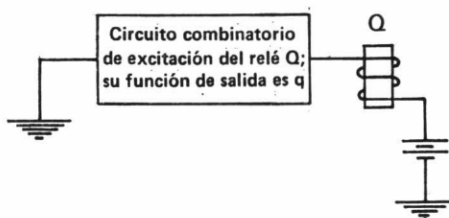


FIG. 3.8

El relé Q, como ya se ha dicho, es un elemento utilizado como memoria. Representa una variable interna o secundaria. Cuando $Q = 1$, el relé está actuado por circular una corriente de excitación por su bobina. Cuando $Q = 0$, el relé está desexcitado y, por consiguiente, $i = 0$.

Los relés pueden tener varios tipos de contactos:

- a) Contactos de trabajo. Son contactos abiertos cuando el relé está desexcitado. Se representan en este texto por la misma variable que define al relé del cual dependen (*).
- b) Contactos de reposo. Son contactos cerrados cuando el relé está desexcitado. Se representan por la variable negada que define el relé del cual dependen.
- c) Contactos de reposo-trabajo o de transferencia.
- d) Contactos de trabajo-reposo o de transferencia con continuidad, que se representan como se indica en la figura 3.9.

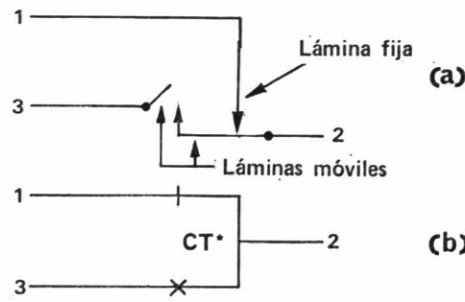


FIG. 3.9

Una vez obtenidas las funciones q_1 , q_2 y q_3 se pueden representar los circuitos de excitación de los relés Q_1 , Q_2 y Q_3 . En la figura 3.10 se indica el correspondiente a Q_1 .

$$q_1 = Q_2 Q_3 x' + Q_1 x + Q_1 Q_3$$

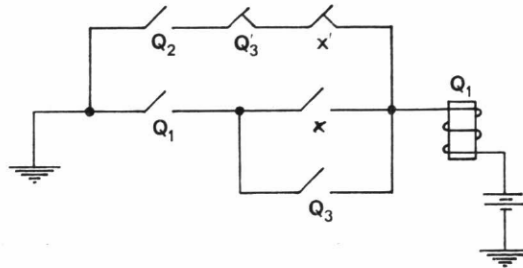


FIG. 3.10

Sustituyendo los contactos x y x' por un contacto de transferencia, se obtiene el circuito equivalente de la figura 3.11.

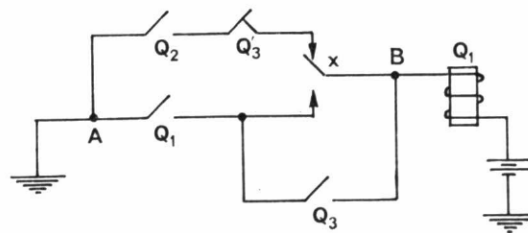


FIG. 3.11

Circuitos análogos se pueden obtener para los relés Q_2 y Q_3 .

(*) Contacto de trabajo: —X— Contacto de reposo: —|—

3.12 EL AZAR EN LOS CIRCUITOS DE CONMUTACION

Nos basaremos en el circuito de la figura 3.11. Se admitirá que el dispositivo está en el estado $\frac{0}{110}$, estable (fig. 3.12).

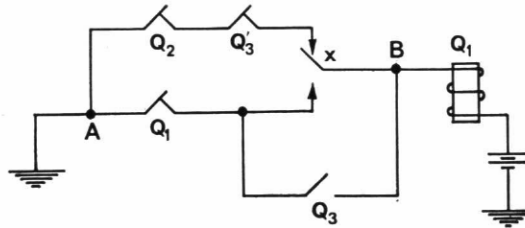


FIG. 3.12

El relé Q_1 está excitado a través del camino:

$$\text{tierra} - Q_2 - Q_3' - x' - \text{batería} - \text{tierra.}$$

Cuando X cambia de 0 a 1, el relé Q_3 se debe excitar. Los terminales A y B permanecerán conectados todo el tiempo solamente si el contacto x' abre después de que el contacto x se cierre. Hay un intervalo de tiempo en que los terminales A y B no están conectados. Si este intervalo es lo suficientemente grande, el relé Q_1 se desexcita y el contacto Q_1 se abre. Esto ocasionará que Q_1 esté desexcitado a pesar de que x se cierre. Esta condición de azar puede impedir alcanzar el estado estable deseado.

Existen varios métodos para evitar estos fenómenos. Algunos de ellos son:

1. Utilizar contactos de transferencia con continuidad.
2. Hacer el relé Q_1 lento al desprendimiento.
3. Proveer un camino redundante, independiente de la variable que cambia. En este caso concreto se añadiría el camino adicional $Q_1 Q_2$ (fig. 3.13).

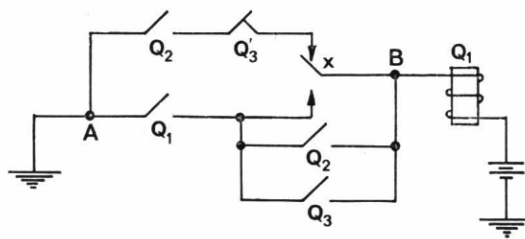


FIG. 3.13

Algunas características de este camino $Q_1 Q_2$ son:

- a) Es totalmente redundante, pero tiene la ventaja de que la conexión entre los terminales A y B no depende del contacto de transferencia, x , durante su transición.

b) Para obtener el camino $Q_1 Q_2$ es necesario analizar el mapa de Karnaugh de la figura 3.14.

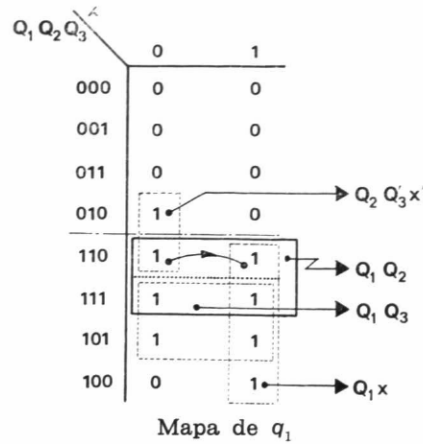


FIG. 3.14

La transición que se estudia comienza en 1100 y termina en 1101. Desafortunadamente, estas dos cuadrículas pertenecen a agrupamientos distintos, lo que significa que la conexión entre los terminales A y B pasa desde 1100 (camino $Q_2 Q_3 x'$) a 1101 (camino $Q_1 x$)

Ambos caminos están influenciados por x , produciéndose una condición de azar. Para evitarla se añade el agrupamiento $Q_1 Q_2$.

En este caso, al cambiar x de 0 a 1, las cuadrículas inicial y final están contenidas en el mismo agrupamiento.

c) ¿Existen otras transiciones en las que ocurran condiciones de azar?

Para averiguarlo es necesario examinar la figura 3.14. Se aplica la siguiente regla:

Si las cuadrículas inicial y final contienen un 1 y pertenecen al mismo agrupamiento, no ocurre ninguna condición de azar.

El mismo razonamiento se puede aplicar para las funciones q_2 y q_3 , con lo cual las excitaciones resultan ser:

$$q_1 = Q_2 Q_3' x' + Q_1 x + Q_1 Q_3 + \underline{Q_1 Q_2}$$

$$q_2 = Q_1' Q_3 x' + Q_2 x + Q_2 Q_3' + \underline{Q_1' Q_2}$$

$$q_3 = Q_1 x' + Q_1 Q_2 x + Q_1' Q_2' x + \underline{Q_1' Q_2' Q_3} + \underline{Q_1 Q_2 Q_3} = Q_3 x' + (Q_1 Q_2 + Q_1' Q_2')(x + Q_3)$$

3.13 RESUMEN

Se han ido desglosando en los diferentes apartados de este capítulo los pasos necesarios para llevar a cabo el proceso de diseño de un circuito secuencial asíncrono, tomando como base de partida las especificaciones del problema.

Como recordatorio, se listan a continuación las etapas necesarias, que deben ser efectuadas secuencialmente:

1. Obtención de un diagrama de estados, normalmente el de Moore.
2. Obtención de una tabla de estados. (Normalmente se obtendrá una matriz primitiva de fases.)
3. Obtención de un diagrama de fusión.
4. Obtención de una matriz de fases.
5. Codificación de la matriz de fases, con lo cual se obtendrá una matriz de excitaciones asociada con su matriz de salida correspondiente.
6. Análisis de la matriz de excitaciones, comprobando que no existen carreras críticas.
7. Obtención de las funciones de excitación y de salida a partir de las matrices correspondientes.
8. Materialización del circuito.

En el caso de utilizarse relés para la instrumentación de las funciones es necesario, además:

1. Estudio de las condiciones de azar.
2. Obtención de las funciones de excitación q_i definitivas.
3. Materialización del circuito.

3.14 EJEMPLOS

1. Un circuito consta de una entrada, X , y de dos salidas, Y_1 e Y_2 ; recibe una señal binaria que puede cambiar de nivel en cualquier instante. Se desea que por la salida Y_1 sólo aparezcan las señales 1 pares y por la Y_2 sólo las impares.

Diséñese un circuito de relés que realice la función indicada.

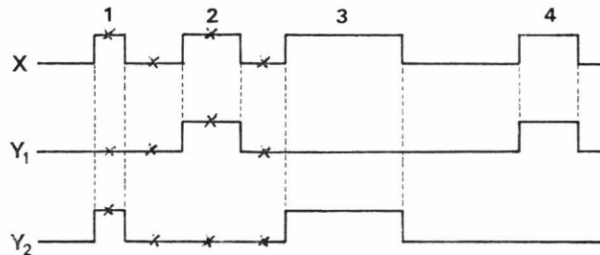


FIG. 3.15

a) DIAGRAMA DE MOORE

Existen cuatro estados:

- Señal cero entre un uno impar y un uno par.
- Señal cero entre un uno par y un uno impar.
- Número de unos par.
- Número de unos impar.

Con lo cual se obtiene el diagrama de Moore de la figura 3.16.

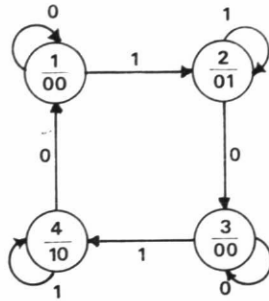


FIG. 3.16

b) MATRIZ PRIMITIVA DE FASES

Estado presente	Próximo estado		Salida presente
	0	1	
1	①	2	00
2	3	②	01
3	③	4	00
4	1	④	10

En este caso, la matriz primitiva de fases no puede ser reducida.

c) MATRIZ DE EXCITACIONES. MATRIZ DE SALIDA

Se impone la condición adicional de que el estado inicial 1 corresponda a la combinación $Q_1 Q_2 = 00$. Por otra parte:

Número de filas $\leq 2^n$ de relés secundarios

Estado	Q_1	Q_2
1	0	0
2	0	1
3	1	1
4	1	0

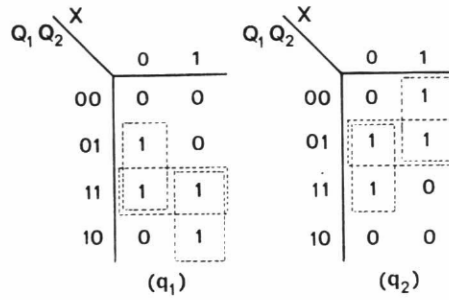
con lo cual:

$Q_1 Q_2 \backslash X$	0	1	Y_1	Y_2
00	00	01	0	0
01	11	01	0	1
11	11	10	0	0
10	00	10	1	0

El análisis de esta matriz de excitaciones demuestra que no existen carreras críticas.

d) FUNCIONES q_1 Y q_2

Los mapas que definen estas funciones son:



Teniendo en cuenta las condiciones de azar se obtiene:

$$q_1 = Q_2 X' + Q_1 X + Q_1 Q_2$$

$$q_2 = Q_2 X' + Q_1' X + Q_1' Q_2$$

e) FUNCIONES DE SALIDA

$$Y_1 = Q_1 Q_2'$$

$$Y_2 = Q_1' Q_2$$

f) CIRCUITOS

Para q_1 :

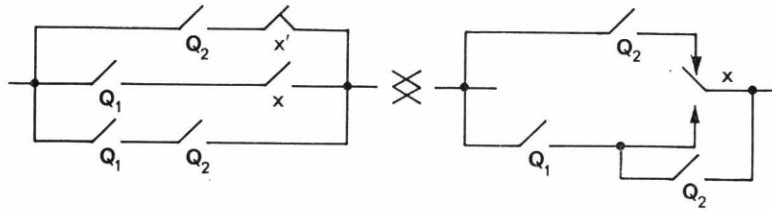


FIG. 3.17

Para q_2 :

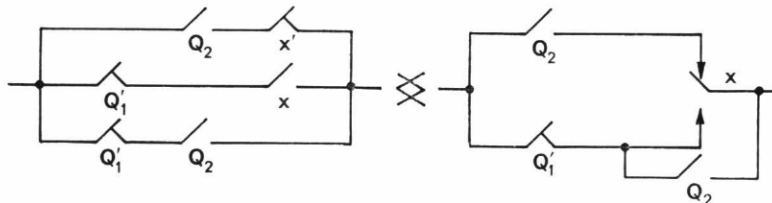


FIG. 3.18

Circuito simplificado:

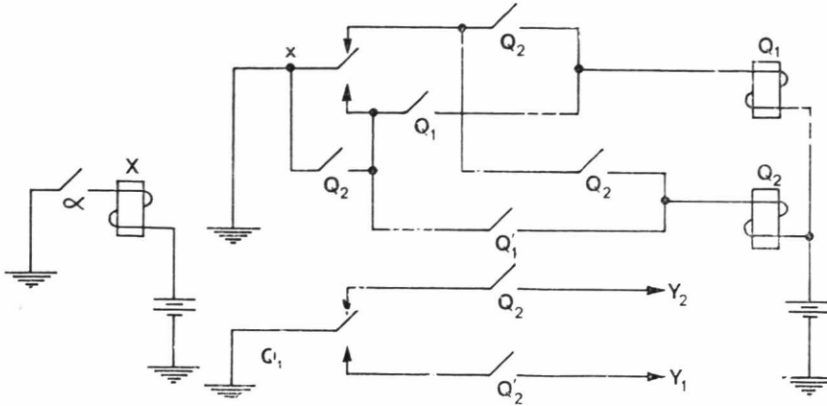


FIG. 3.19

2. Dado el diagrama lógico de la figura 3.20, determinar:

- Funciones R y S de cada «flip-flop» y función de salida Y .
- Matriz de excitaciones secundarias y matriz de salida.
- Diagrama de estado (Moore).
- ¿Qué función realiza este circuito?

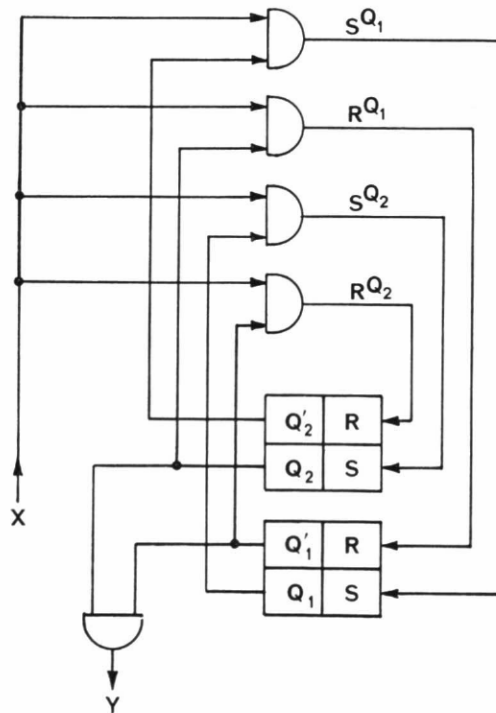


FIG. 3.20

Las funciones R y S de los «flip-flops» Q_1 y Q_2 , así como la función de salida Y , son las siguientes:

$$\begin{aligned} R^{Q_1} &= X Q_2 \\ S^{Q_1} &= X Q_2' \\ R^{Q_2} &= X Q_1' \\ S^{Q_2} &= X Q_1 \\ Y &= Q_1' Q_2 \end{aligned}$$

Mediante un análisis del diagrama lógico del circuito se obtienen las matrices de excitaciones y salida que corresponden al mismo. Se admite como estado inicial $Q_1 Q_2 = 00$ y que la duración de los impulsos de entrada es menor que el tiempo de transición de la memoria.

Q_1 Q_2	Próximo estado		Salida
	$X = 0$	$X = 1$	Y
0 0	00	10	0
1 0	10	11	0
1 1	11	01	0
0 1	01	00	1

Es fácil obtener la función de salida Y :

$$Y = Q_1' Q_2$$

que, como se ve, corresponde a la indicada en el logigrama.

Admitiendo que los estados internos 00, 10, 11 y 01 corresponden a 1, 2, 3 y 4, respectivamente, se obtiene la siguiente tabla de estados:

Estado presente	Próximo estado		Salida
	$X = 0$	$X = 1$	
1	1	2	0
2	2	3	0
3	3	4	0
4	4	1	1

que procede del diagrama de Moore indicado en la figura 3.21.

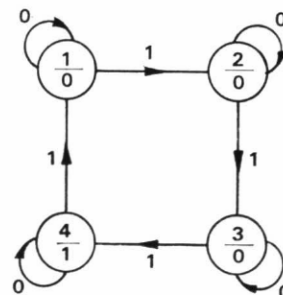


FIG. 3.21

Este circuito funciona como un divisor 4/1, con excepción del primer ciclo.

NOTA: Como comprobación, se deducen a continuación las funciones R^{Q_1} , S^{Q_1} , R^{Q_2} y S^{Q_2} , cuyos mapas obtenidos a partir de la matriz de excitaciones son:

$Q_1 Q_2 \backslash X$	0	1
00	0	1
01	0	0
11	-	0
10	-	-

$$S^{Q_1} = X Q_2'$$

$Q_1 Q_2 \backslash X$	0	1
00	-	0
01	-	-
11	0	1
10	0	0

$$R^{Q_1} = X Q_2$$

$Q_1 Q_2 \backslash X$	0	1
00	0	0
01	-	0
11	-	-
10	0	1

$$S^{Q_2} = X Q_1$$

$Q_1 Q_2 \backslash X$	0	1
00	-	-
01	0	1
11	0	0
10	-	0

$$R^{Q_2} = X Q_1'$$

CAPITULO 4

CIRCUITOS SECUENCIALES SINCRONOS

4.1 DIAGRAMA DE BLOQUES

El diagrama de bloques de un circuito secuencial síncrono ya ha sido representado en la figura 1.3. Para mayor facilidad se repite en la figura 4.1.

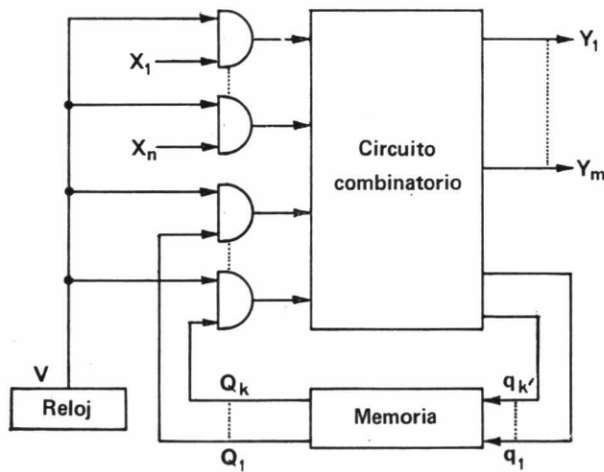


FIG. 4.1

Puede verse que un circuito secuencial síncrono es un circuito combinatorio con una memoria. El circuito combinatorio tiene $n + k$ entradas: $X_1, X_2, \dots, X_n, Q_1, Q_2, \dots, Q_k$.

Las variables X_1, X_2, \dots, X_n son las entradas o variables primarias. Q_1, Q_2, \dots, Q_k indican los bits que pueden obtenerse de la memoria.

Las $m + k'$ salidas son: Y_1, Y_2, \dots, Y_m (que representan las salidas del circuito secuencial) y $q_1, q_2, \dots, q_{k'}$ (que representan los bits que van a ser almacenados en la memoria).

El órgano que diferencia a los circuitos secuenciales síncronos de los asíncronos es el reloj.

Por la presencia de puertas *AND* a la entrada del circuito combinatorio, éstas tienen que estar sincronizadas con los impulsos de reloj.

Para ilustrar la forma de operación de un circuito secuencial síncrono, se va a examinar el diagrama de bloques de un sumador serie binario.

El diagrama de la figura 4.2 es una versión específica y concreta del indicado en la figura 4.1.

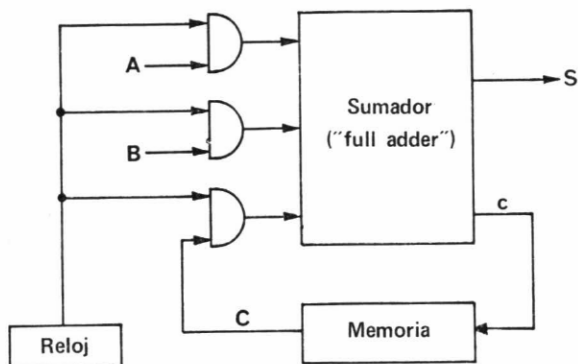


FIG. 4.2

Sean los números *A* y *B* que van a ser sumados en serie y que se expresan en el sistema binario como:

$$A = a_n a_{n-1} \dots a_1 a_0$$

$$B = b_n b_{n-1} \dots b_1 b_0$$

Se admite que cada bit de *A* y *B*, la suma *S* y el «acarreo» *c* son todos impulsos, mientras que la salida de la memoria, *C*, que representa el «acarreo» almacenado, está en forma de un nivel de voltaje.

Supóngase que en el instante $t = t_0$ comienza la adición en serie; aparecerán simultáneamente en las puertas *AND* los impulsos que representan a_0 , b_0 con un impulso de reloj; también se aplica un voltaje que representa C_0 en el instante t_0 . En la salida obtendremos s_0 y c_0 , que funcionalmente pueden representarse como:

$$s_0 = f(a_0, b_0, C_0)$$

$$c_0 = g(a_0, b_0, C_0)$$

Estas funciones implican que se ha despreciado el retraso del circuito combinatorio en producir las salidas s_0 y c_0 .

En el instante $t = t_0$, un impulso que representa c_0 se envía a la memoria; si este impulso apareciese sin retardo a la salida de ésta, se sumaría de nuevo con a_0 y b_0 . Para evitar este peligro de carreras se introduce un cierto retardo en la memoria, que se define como el intervalo entre el instante en que se envía un bit a la misma y el instante en que puede sentirse este bit a su salida. Recordando la forma

idealizada de los impulsos de reloj (fig. 1.4), se ve que el tiempo de retardo t_r de la memoria debe cumplir:

$$t_d < t_r < t_s$$

o lo que es lo mismo,

$$C_{a+1} = c_a$$

siendo c_a el «acarreo» en un instante t_a en general.

En el instante $t = t_1$ ($t_1 = t_0 + t_s$) aparecen simultáneamente el segundo impulso de reloj y los impulsos a_1 y b_1 , junto con la salida C_1 de la memoria, representando el «acarreo» de la operación anterior ($C_1 = c_0$).

Sin retraso, en el circuito combinatorio se obtiene:

$$s_1 = f(a_1, b_1, C_1)$$

$$c_1 = g(a_1, b_1, C_1)$$

Esta operación continúa hasta que se han procesado todos los bits de A y B .

Del razonamiento anterior se pueden inducir las ecuaciones generales de un circuito secuencial síncrono expresadas en el punto 1.3.

4.2 DISEÑO DE CIRCUITOS SECUENCIALES SINCRONOS

El diseño de un circuito secuencial síncrono seguirá pasos muy similares a los del diseño de un circuito secuencial asíncrono:

1. Establecimiento de un diagrama de estados y/o de una tabla de transición de estados.
2. Determinación del número de «flip-flops» que se van a utilizar para la memoria.
3. Asignación de posibles combinaciones a los estados establecidos.
4. Determinación y simplificación de las funciones de conmutación.
5. Materialización de las funciones.

En la optimización de circuitos secuenciales hay que tener muy en cuenta el número de «flip-flops» que compondrán la memoria, el cual, junto con la asignación de estados, está muy relacionado con la complejidad del circuito.

4.3 ESTADOS EQUIVALENTES

Sea el siguiente diagrama de bloque de un circuito secuencial síncrono:

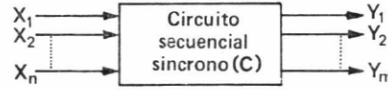


FIG. 4.3

El circuito secuencial C tiene un número finito de estados internos. Sean dos de ellos S_1 y S_2 .

Si C está inicialmente en el estado S_1 y se aplica una secuencia arbitraria, X , de combinaciones de entrada, se obtiene la secuencia $Y_{S_1}(X)$ de combinaciones de salida.

Ejemplos: Sea

$$\begin{aligned} n &= 1 \\ m &= 1 \\ X: & 0, 0, 1, 1 \\ Y(X): & 1, 0, 0, 0 \end{aligned}$$

Si C está inicialmente en el estado S_2 y se aplica la misma secuencia X de combinaciones de entrada, se obtiene la secuencia $Y_{S_2}(X)$ de combinaciones de salida.

Por definición, S_1 y S_2 son estados equivalentes si $Y_{S_1}(X)$ e $Y_{S_2}(X)$ son idénticos. En este caso, S_1 puede reemplazar a S_2 , o viceversa.

A fin de reducir el número de filas de una tabla de transición de estados, se deben localizar parejas de estados equivalentes.

El principal inconveniente de utilizar el método anterior radica en la necesidad de satisfacer la especificación de que *dos estados son equivalentes si generan la misma secuencia de combinaciones de salida para cualquier secuencia arbitraria de combinaciones de entrada*. El número de secuencias de combinaciones de entrada puede ser infinito, ya que la longitud de cada secuencia (o número de combinaciones de entrada contenidas en cada secuencia) puede ser cualquier entero.

Para resolver este problema se reduce la longitud de cada secuencia a 1. De esta forma el número de secuencias es finito e igual a 2^n .

Se observa, con esta aproximación, que dos estados cualesquiera que satisfagan la condición anterior no pueden ser declarados equivalentes, ya que no satisfacen los requerimientos establecidos en la definición. Para satisfacer estos requerimientos se establece:

1. Que las combinaciones de salida que correspondan a cada una de las 2^n combinaciones de entrada sean las mismas.
2. Que el circuito se transforme en el mismo próximo estado, cuando el estado de partida sea S_1 o S_2 .

Ejemplo: Sea la tabla de transición de estados:

Estado presente	Próximo estado		Próxima salida	
	0	1	0	1
1	2	3	00	00
2	4	5	00	00
3	8 5	7 4	00	00
4	8	9	00	00
5	10 9	11 9	00	00
6	12 9	13 8	00	00
7	14 8	15 9	00	00
8	1	1	10	01
9	1	1	01	10
10	1	1	01	10
11	1	1	10	01
12	1	1	01	10
13	1	1	10	01
14	1	1	10	01
15	1	1	01	10

Al intentar localizar estados equivalentes se buscan filas cuyas combinaciones de salida y próximos estados sean idénticos para cualquier posible combinación de entrada. De acuerdo con esto:

- a) Las filas 9, 10, 12 y 15 son idénticas y pueden reducirse a una sola. Sea ésta, por ejemplo, la fila 9. Las filas 10, 12 y 15 se cruzan con una raya.
- b) Las filas 8, 11, 13 y 14 son también idénticas y pueden reducirse a la fila 8.
- c) El siguiente paso consiste en sustituir en las columnas de próximos estados 10, 12 y 15 por 9 y 11, 13 y 14 por 8.
- d) Por consiguiente, las filas 5 y 6, 4 y 7 son idénticas.

Al desaparecer las filas 6 y 7 y sustituir en las columnas de próximos estados éstos por sus equivalentes, la tabla de estados queda reducida a siete filas.

Estado presente	Próximo estado		Próxima salida	
	0	1	0	1
1	2	3	00	00
2	4	5	00	00
3	5	4	00	00
4	8	9	00	00
5	9	8	00	00
8	1	1	10	01
9	1	1	01	10

4.4 ESTADOS EQUIVALENTES REPRESENTADOS POR FILAS NO IDENTICAS

Sea la siguiente tabla de transición de estados:

Estado presente	Próximo estado		Próxima salida	
	0	1	0	1
1	2	3	0	0
2	5	3	1	0
3	4	1	0	0
4	5	1	1	0
5	5	5	1	0

Ya se ha dicho que para localizar filas idénticas en una tabla de transición se buscan estados cuyas combinaciones de salida y próximos estados sean idénticos para cualquier posible combinación de entrada. *Sin embargo, no es necesario que los próximos estados sean los mismos.*

En vez de comparar filas enteras, se examinan solamente las columnas de salida de dos estados y se agrupan éstos de la forma siguiente:

I (0, 0)	II (1, 0)
1, 3	2, 4, 5

Quiere esto decir que la condición necesaria, pero no suficiente, para que los estados 1 y 3 sean equivalentes es que la longitud de la secuencia de entrada sea 1.

De acuerdo con lo anterior se obtiene la siguiente tabla:

Nuevo estado	Próximo estado	
	0	1
I	II	I
II	II	I
I	II	I
II	II	I
II	II	II

Se omite la columna de salida, puesto que las salidas de los nuevos estados ya han sido fijadas.

El proceso siguiente es una comprobación del número de nuevos estados para ver si es adecuado:

1. Las filas 1 y 3 corresponden al nuevo estado I. Los próximos estados son II y I cuando las entradas son 0 y 1, respectivamente. No se observan contradicciones.
2. Las filas 2, 4 y 5 corresponden al nuevo estado II. Para las filas 2 y 4, las columnas de la derecha son idénticas. Al comparar 5 con 2 ó 4 se encuentra una contradicción; cuando la entrada es 1 el próximo estado de II es II, de acuerdo con la fila 5, o I, de acuerdo con las filas 2 y 4. Esta contradicción muestra que 5 no puede agruparse con 2 y 4.

Se crea un nuevo estado III y se realiza un nuevo agrupamiento:

I (0, 0)	II (1, 0)	III (1, 0)
1, 3	2, 4	5

Se obtiene en este caso la tabla:

Nuevo estado	Próximo estado	
	0	1
I	II	I
II	III	I
I	II	I
II	III	I
III	III	III

que no presenta ninguna contradicción; por lo que el número final de estados viene representado por la tabla siguiente:

Estado presente	Próximo estado		Próxima salida	
	0	1	0	1
I	II	I	0	0
II	III	I	1	0
III	III	III	1	0

Empleando este método, se observa que el número inicial de nuevos estados es igual al número que existe de combinaciones de salida y no se introduce ningún estado adicional, excepto cuando se encuentra una contradicción.

4.5 METODO DE TABULACION PARA LA REDUCCION DE TABLAS DE ESTADOS

Sea la siguiente tabla de estados:

TABLA 4.5.1


Estado presente	Próximo estado		Próxima salida	
	0	1	0	1
1	2	3	00	00
2	4	5	00	00
3	6	7	00	00
4	8	9	00	00
5	10	11	00	00
6	12	13	00	00
7	14	15	00	00
8	1	1	10	01
9	1	—	01	—
10	1	—	01	—
11	1	—	10	—
12	1	1	01	10
13	1	—	10	—
14	1	—	10	—
15	1	—	01	—

Esta tabla puede ser simplificada utilizando el método de localización de estados equivalentes. Es fácil ver, por ejemplo, que los estados 8 y 11 son equivalentes si se asignan en la fila 11 valores a las entradas no especificadas. Pero existe la posibilidad de que otras asignaciones a las entradas no especificadas proporcionen una reducción más idónea.

Para investigar todas las equivalencias conviene emplear un procedimiento sistemático, llamado método de tabulación, cuyos pasos son los siguientes:

1. CONSTRUCCIÓN DE UNA TABLA DE IMPLICACIONES

TABLA 4.5.2

2	$\begin{matrix} 2-4 \\ 3-5 \end{matrix}$																
3	$\begin{matrix} 2-6 \\ 3-7 \end{matrix}$	$\begin{matrix} 4-6 \\ 5-7 \end{matrix}$															
4	$\begin{matrix} 2-8 \\ 3-9 \end{matrix}$	$\begin{matrix} 4-8 \\ 5-9 \end{matrix}$	$\begin{matrix} 6-8 \\ 7-9 \end{matrix}$														
5	$\begin{matrix} 2-10 \\ 3-11 \end{matrix}$	$\begin{matrix} 4-10 \\ 5-11 \end{matrix}$	$\begin{matrix} 6-10 \\ 7-11 \end{matrix}$	$\begin{matrix} 8-10 \\ 9-11 \end{matrix}$													
6	$\begin{matrix} 2-12 \\ 3-13 \end{matrix}$	$\begin{matrix} 4-12 \\ 5-13 \end{matrix}$	$\begin{matrix} 6-12 \\ 7-13 \end{matrix}$	$\begin{matrix} 8-12 \\ 9-13 \end{matrix}$	$\begin{matrix} 10-12 \\ 11-13 \end{matrix}$												
7	$\begin{matrix} 2-14 \\ 3-15 \end{matrix}$	$\begin{matrix} 4-14 \\ 5-15 \end{matrix}$	$\begin{matrix} 6-14 \\ 7-15 \end{matrix}$	$\begin{matrix} 8-14 \\ 9-15 \end{matrix}$	$\begin{matrix} 10-14 \\ 11-15 \end{matrix}$	$\begin{matrix} 12-14 \\ 13-15 \end{matrix}$											
8	xx	xx	xx	xx	xx	xx	xx	xx									
9	xx	xx	xx	xx	xx	xx	xx	xx	xx								
10	xx	xx	xx	xx	xx	xx	xx	xx	xx	✓							
11	xx	xx	xx	xx	xx	xx	xx	xx	✓	xx	xx						
12	xx	xx	xx	xx	xx	xx	xx	xx	✓	✓	xx						
13	xx	xx	xx	xx	xx	xx	xx	✓	xx	xx	✓	xx					
14	xx	xx	xx	xx	xx	xx	xx	✓	xx	xx	✓	xx	✓				
15	xx	xx	xx	xx	xx	xx	xx	xx	✓	✓	xx	✓	xx	xx	xx		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14			

En esta tabla hay tantas casillas como pares de estados se puedan obtener de la tabla de estados 4.5.1. En cada una se indica si es o no posible la equivalencia entre estados.

Consideremos la casilla 2-1. Al ser (00, 00) las salidas de las filas 1 y 2, es posible la equivalencia entre ambos estados; para ello, 2 y 4, y 3 y 5 deben ser equivalentes. Esta información se refleja en dicha casilla.

Comparando las filas 8 y 1, se observa que sus salidas son diferentes y no es posible agruparlas. Este hecho se indica colocando una cruz en la casilla 8-1.

Las filas 10 y 9 son idénticas, por lo cual los dos estados son equivalentes. Esto se indica colocando una marca de comprobación en la casilla 10-9.

Este mismo procedimiento se seguiría para completar las restantes casillas de la tabla de implicaciones.

2. INVESTIGACIÓN DE LAS EQUIVALENCIAS ENTRE PARES DE ESTADOS

Para representar toda la información sobre las posibles equivalencias entre pares de estados se examina primero la casilla 15-14. De ella se deduce que 15 y 14 no pueden ser equivalentes. A continuación se examinan todas las casillas restantes para averiguar si la equivalencia entre pares de estados depende de la equiva-

lencia entre 15 y 14. Caso de encontrarse, se coloca una cruz en la casilla correspondiente para indicar que no es posible la equivalencia entre sus dos estados. En el presente problema no existe ningún caso. Cuando se termina la investigación de todas las casillas se coloca una cruz adicional en la casilla 15-14.

A continuación se examina la columna 13. En la casilla 14-13 existe una marca de comprobación; esto significa que los estados 13 y 14 son equivalentes. La casilla 15-13 está marcada con una cruz. Por este hecho, en la casilla 7-6 se coloca una cruz, y en la 15-13, otra.

Este proceso continúa hasta que se terminan de examinar todas las posibilidades.

3. OBTENCIÓN DE LOS POSIBLES AGRUPAMIENTOS

El próximo paso en el proceso de reducción consiste en extraer todos los agrupamientos posibles existentes entre los estados. Cada agrupamiento consta de estados que pueden ser equivalentes.

TABLA 4.5.3

14	
13	13-14
12	13-14, 12-15
11	11-13-14, 12-15
10	11-13-14, 10-12-15
9	11-13-14, 9-10-12-15
8	8-11-13-14, 9-10-12-15
7	8-11-13-14, 9-10-12-15
6	8-11-13-14, 9-10-12-15
5	5-6, 8-11-13-14, 9-10-12-15
4	4-7, 5-6, 8-11-13-14, 9-10-12-15
3	4-7, 5-6, 8-11-13-14, 9-10-12-15
2	4-7, 5-6, 8-11-13-14, 9-10-12-15
1	4-7, 5-6, 8-11-13-14, 9-10-12-15
FINAL	1, 2, 3, 4-7, 5-6, 8-11-13-14, 9-10-12-15

Se comienza por la columna 14 de la tabla 4.5.2. Ningún agrupamiento puede establecerse. De acuerdo con esto, la fila 14 de la tabla 4.5.3 debe quedar en blanco.

Columna 13: Agrupamiento posible, 13-14.

Columna 12: Agrupamiento posible, 12-15. Las entradas de la fila 12 de la tabla 4.5.3 son 13-14 y 12-15.

Columna 11: Existen dos agrupamientos posibles, 11-13 y 11-14. Las entradas de la fila 11 podrían ser 11-13, 11-14, 13-14 y 12-15. Sin embargo, el primero de los agrupamientos sugiere 11-13-14, por lo cual las entradas de la fila 11 son 11-13-14 y 12-15.

Columna 10: Agrupamientos posibles, 10-12 y 10-15. Las entradas de la fila 10 podrían ser 11-13-14, 12-15, 10-12 y 10-15. Agrupando las tres últimas se obtendría 11-13-14 y 10-12-15.

Columna 9: Agrupamientos posibles, 9-10, 9-12 y 12-15. Se obtiene 11-13-14 y 9-10-12-15.

Usando este razonamiento se completa la tabla 4.5.3.

Las entradas de la fila FINAL se obtienen listando todas las entradas de la fila 1 y todos los estados de la tabla de estados 4.5.1 dada, que hayan sido omitidos.

4. SELECCIÓN DE AGRUPAMIENTOS

A continuación se seleccionan agrupamientos de entre los listados en la fila FINAL. Se aplican dos reglas:

- a) Cada estado de la tabla de estado original tiene que estar incluido, por lo menos, en uno de los agrupamientos seleccionados (cada agrupamiento seleccionado implica un estado en la nueva tabla de estado). Este requerimiento asegura que todos los estados de la tabla dada serán considerados. Es interesante señalar que un estado de la tabla original puede aparecer en más de un agrupamiento seleccionado.
- b) Los agrupamientos que se seleccionen deben ser escogidos de forma que todas las implicaciones necesarias sean satisfechas. Esto significa que si la equivalencia entre los estados en un agrupamiento depende de la equivalencia de otros estados, estos últimos pertenecerán a un agrupamiento seleccionado. Por ejemplo, si el agrupamiento 5-6 se incluye en la selección, los estados 10 y 12 deben aparecer en un agrupamiento de la selección; esto se debe a que la equivalencia entre los estados 5 y 6 depende de la equivalencia entre los estados 10 y 12. Por la misma razón, los estados 11 y 13 deben también pertenecer a un agrupamiento en la selección.

Aplicando estas dos reglas a la tabla dada vemos que:

- a) La selección incluirá todos los agrupamientos listados en la fila FINAL (por la primera regla).
- b) Examinando la tabla original se observa que la equivalencia de los estados 4 y 7 depende de la equivalencia entre 8 y 14 y entre 9 y 15. La segunda regla establece que los estados 9 y 15 deben aparecer en un agrupamiento y también los estados 8 y 14. Puesto que 9 y 15 aparecen en el agrupamiento 9-10-12-15, y 8 y 14 aparecen en el agrupamiento 8-11-13-14, las implicaciones necesarias se satisfacen para los estados 4 y 7.

- c) Un examen similar muestra que las implicaciones de los estados 5 y 6 también se satisfacen.

La selección es, pues: 1, 2, 3, 4-7, 5-6, 8-11-13-14, 9-10-12-15.

5. OBTENCIÓN DE LA TABLA DE ESTADOS REDUCIDA

Cada agrupamiento se representa como un nuevo estado, obteniéndose una tabla de estados reducida.

TABLA 4.5.4

Estado presente	Próximo estado		Próxima salida	
	0	1	0	1
1	2	3	00	00
2	4	5	00	00
3	5	4	00	00
4	8	9	00	00
5	9	8	00	00
8	1	1	10	01
9	1	1	01	10

→ Aparecen en la fila 12 de la TABLA 4.5.1

Mediante este procedimiento exhaustivo se puede asegurar que las entradas no especificadas de la tabla de estados se usan de una forma idónea.

El método de tabulación explicado puede aplicarse a la reducción de tablas de estados, especificadas o no completamente.

4.6 DISEÑO DE LA MEMORIA

El primer paso en el diseño de la memoria es la determinación del número necesario de elementos binarios de almacenamiento.

Este número puede calcularse a partir del número de estados que aparecen en la tabla de estados reducida, es decir, de lo que tiene que almacenar la memoria.

Se podría pensar que con el conocimiento de este número mínimo se completa el primer paso en el diseño de la memoria, lo que sería correcto si el coste y la seguridad del circuito estuvieran solamente determinados por el número de elementos binarios de almacenamiento; pero se debe tener también en cuenta el coste del circuito combinatorio.

No es evidente que el número mínimo de elementos de almacenamiento implique la obtención del circuito combinatorio más sencillo. Para la determinación de este número se utiliza un método de ensayos sucesivos.

4.7 DISEÑO DE CONTADORES

El diseño de contadores es de interés no sólo por su enorme importancia, sino también por el hecho de que ilustra las consideraciones básicas que atañen al diseño general de un circuito secuencial síncrono.

4.7.1 Contador binario hexadecimal

Se va a diseñar un contador que cuente impulsos y represente este número en binario, regresando a su estado inicial cada vez que cuente 16 impulsos. Se utilizarán «flip-flops» tipo *RS*.

De acuerdo con las especificaciones dadas, el contador deberá representar el número de impulsos contados. Así, el estado del circuito en cualquier instante es el número de impulsos contados en ese instante.

Sea *A* el estado en que no se ha contado ningún impulso; *B*, el estado en que se ha contado uno; etc. La segunda condición de la especificación fija el número de estados, ya que cuando se cuenten 16 impulsos se regresa al estado inicial. Por tanto, el número de estados es 16.

La tabla 4.7.1 representa la tabla de transición de estados.

TABLA 4.7.1

Estado presente	Estado próximo	
	$V_n = 0$	$V_n = 1$
A_n	A_{n+1}	B_{n+1}
B_n	B_{n+1}	C_{n+1}
C_n	C_{n+1}	D_{n+1}
D_n	D_{n+1}	E_{n+1}
E_n	E_{n+1}	F_{n+1}
F_n	F_{n+1}	G_{n+1}
G_n	G_{n+1}	H_{n+1}
H_n	H_{n+1}	I_{n+1}
I_n	I_{n+1}	J_{n+1}
J_n	J_{n+1}	K_{n+1}
K_n	K_{n+1}	L_{n+1}
L_n	L_{n+1}	M_{n+1}
M_n	M_{n+1}	N_{n+1}
N_n	N_{n+1}	O_{n+1}
O_n	O_{n+1}	P_{n+1}
P_n	P_{n+1}	A_{n+1}

Los subíndices indican el instante $t = t_n$ y V la señal de entrada procedente del generador de impulsos.

Es conveniente que únicamente sean identificados los estados necesarios, pues de ellos depende el número de «flip-flops». Puesto que el contenido de un «flip-flop» puede ser 0 ó 1, para almacenar 16 combinaciones se necesitarán cuatro «flip-flops».

La siguiente cuestión es: ¿Cómo se asignan las 16 posibles combinaciones a los 16 estados? En general, esta cuestión exige mucho cuidado, por su influencia en la complejidad del circuito combinatorio. En este caso, queda fijada sin alternativa posible por la necesidad de representar en binario el número de impulsos contados. Esto significa que el estado A queda representado por 0000; el B , por 0001, etc.

El diagrama de bloques del contador se muestra en la figura 4.4.

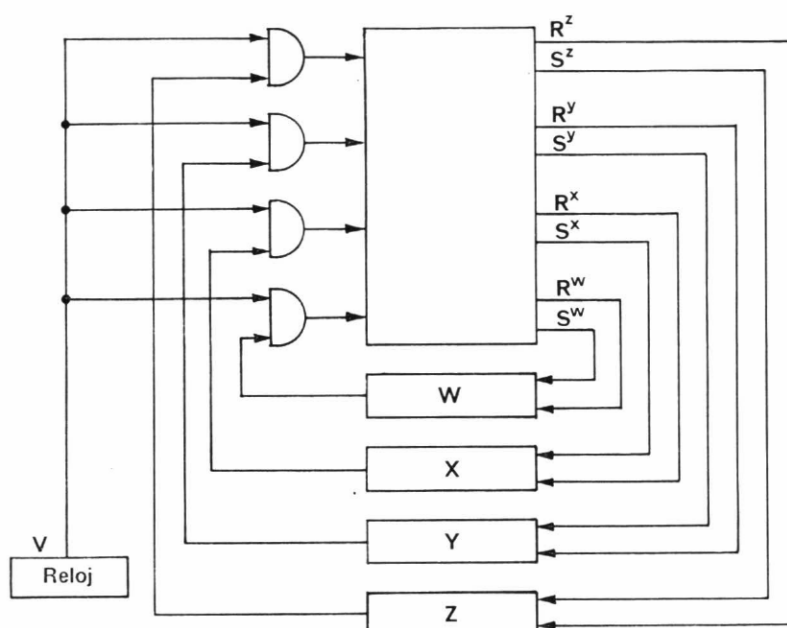


FIG. 4.4

En el diagrama anterior hay que hacer notar que los impulsos de entrada deben estar sincronizados con los impulsos de reloj. La memoria cambiará, y debe cambiar, únicamente cuando llegue un impulso de entrada. Esto explica que en la figura 4.4 la fuente de impulsos de entrada reemplaza al reloj.

El trabajo ahora es diseñar un circuito combinatorio que tiene cuatro variables de entrada y ocho de salida.

Las funciones de conmutación de las ocho salidas se pueden establecer a partir de la tabla de transición de estados 4.7.2.

TABLA 4.7.2

Estado presente				Próximo estado			
				$V_n = 1$			
W_n	X_n	Y_n	Z_n	W_{n+1}	X_{n+1}	Y_{n+1}	Z_{n+1}
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

Esta tabla se obtiene de la tabla 4.7.1, reemplazando los estados por sus respectivas combinaciones de «flip-flops». La columna $V_n = 0$ se omite, porque no hay transiciones de estados.

Se puede considerar la tabla 4.7.2 como cuatro tablas de verdad separadas para W_{n+1} , X_{n+1} , Y_{n+1} y Z_{n+1} , respectivamente.

Quizá parezca extraño que siendo S^w , R^w , S^x , R^x , S^y , R^y , S^z y R^z las variables que se tienen que calcular, no aparezcan en la tabla de verdad. Pero, por tratarse de «flip-flops» RS, entre ellas y las variables W , X , Y y Z existen las siguientes relaciones (*):

$$S_n^B R_n^B = 0$$

$$B_{n+1} = S_n^B + B_n (R_n^B)$$

(*) Como se recordará, la ecuación característica del «flip-flops» RS es: $B_{n+1} = S_n + B_n (R_n)$, con la condición $S_n R_n = 0$, que también se puede indicar de la forma siguiente:

B_n	B_{n+1}	S_n	R_n
0	0	0	—
0	1	1	0
1	1	—	0
1	0	0	1

A continuación se va a obtener la función S_n^w . Obsérvese que

a) Cuando W_n es cero y W_{n+1} es 1, S_n^w será 1. Con este criterio en mente y examinando las columnas W_n y W_{n+1} de la tabla 4.7.2 se obtiene:

$$S_n^w (W_n, X_n, Y_n, Z_n) = W'_n X_n Y_n Z_n = m_7$$

b) Cuando W_n es 1 y W_{n+1} es 1 S_n^w podrá ser 0 ó 1. En estos casos se pueden establecer las siguientes condiciones opcionales:

$$m_8, m_9, m_{10}, m_{11}, m_{12}, m_{13} \text{ ó } m_{14}$$

Antes de simplificar la función establecida para S_n^w , conviene señalar que el circuito combinatorio que se está diseñando es una red multiterminal, cuya simplificación se debe hacer de tal manera que las ocho funciones $S_n^w, R_n^w, S_n^x, R_n^x, S_n^y, R_n^y, S_n^z$ y R_n^z se consideren al mismo tiempo.

Existen dos métodos para obtener estas funciones:

- a) El método de tabulación.
- b) Mapas de Karnaugh.

En este texto se empleará el segundo método.

Las restantes siete funciones se pueden deducir directamente de la tabla de transición.

Los mapas que resultan para las ocho funciones son los siguientes:

«FLIP-FLOP» w :

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	-	-	0	-
10	-	-	-	-

$$S_n^w = W'_n X_n Y_n Z_n$$

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	-	-	-	-
01	-	-	0	-
11	0	0	1	0
10	0	0	0	0

$$R_n^w = W_n X_n Y_n Z_n$$

«FLIP-FLOP» x :

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	0	0	1	0
01	-	-	0	-
11	-	-	0	-
10	0	0	1	0

$$S_n^x = X'_n Y_n Z_n$$

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	-	-	0	-
01	0	0	1	0
11	0	0	1	0
10	-	-	0	-

$$R_n^x = X_n Y_n Z_n$$

«FLIP-FLOP» y :

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	0	1	0	-
01	0	1	0	-
11	0	1	0	-
10	0	1	0	-

$$S_n^y = Y'_n Z_n$$

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	-	0	1	0
01	-	0	1	0
11	-	0	1	0
10	-	0	1	0

$$R_n^y = Y_n Z_n$$

«FLIP-FLOP» z :

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

$$S_n^z = Z'_n$$

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	1	1	0
10	0	1	1	0

$$R_n^z = Z_n$$

Del agrupamiento adecuado en los mapas de Karnaugh, se obtiene:

$$\begin{aligned}
 S_n^w &= W'_n X_n Y_n Z_n & S_n^y &= Y'_n Z_n \\
 R_n^w &= W_n X_n Y_n Z_n & R_n^y &= Y_n Z_n \\
 S_n^x &= X'_n Y_n Z_n & S_n &= Z'_n \\
 R_n^x &= X_n Y_n Z_n & R_n^z &= Z_n
 \end{aligned}$$

Ya para completar el problema no queda más que instrumentar estas funciones.

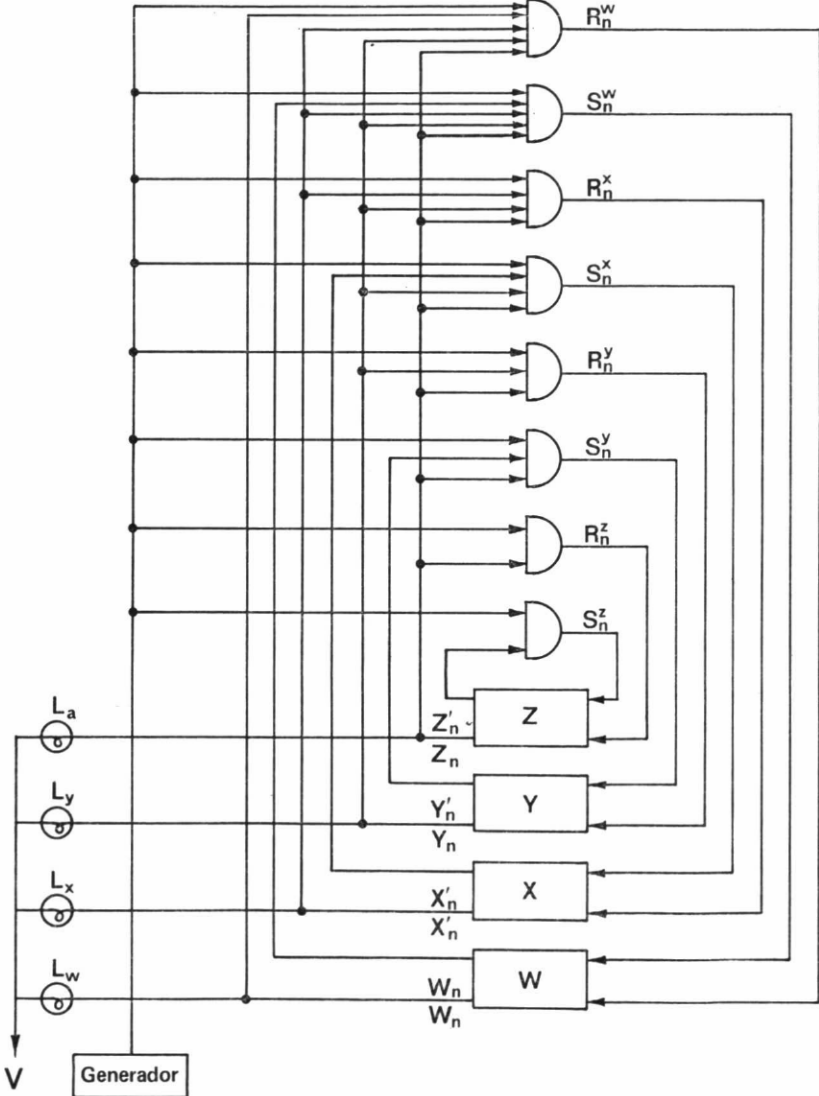


FIG. 4.5

4.7.2 Contador binario decimal

Se pretende diseñar un contador que cuente el número de impulsos recibidos y los represente en el sistema binario. El contador vuelve a su estado inicial cada vez que cuenta 10 impulsos. Se utilizarán «flip-flops» tipo T .

La ecuación característica y la matriz de excitaciones secundarias de un «flip-flop» tipo T son las siguientes:

$$\begin{array}{c|cc}
 & \begin{array}{c} T \\ 0 \quad 1 \end{array} & \\
 \begin{array}{c} B_n \\ 0 \\ 1 \end{array} & \begin{array}{cc} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{array} &
 \end{array}
 \quad B_{n+1} = B'_n T_n + B_n T'_n$$

Se necesitan representar 10 estados. La letra A indica el estado en el cual el número de impulsos contados es $10 \cdot l$ (l es un entero, 0 incluido), la letra B indica el estado en el cual el número de impulsos contados es $10 \cdot l + 1$, etc.

La tabla de transición de estados que se obtiene es la siguiente:

Estado presente	Próximo estado	
	$V_n = 0$	$V_n = 1$
A_n	A_{n+1}	B_{n+1}
B_n	B_{n+1}	C_{n+1}
C_n	C_{n+1}	D_{n+1}
D_n	D_{n+1}	E_{n+1}
E_n	E_{n+1}	F_{n+1}
F_n	F_{n+1}	G_{n+1}
G_n	G_{n+1}	H_{n+1}
H_n	H_{n+1}	I_{n+1}
I_n	I_{n+1}	J_{n+1}
J_n	J_{n+1}	A_{n+1}

Para representar estos 10 estados se requieren cuatro «flip-flops»: W , X , Y y Z . En el presente ejemplo la codificación viene impuesta por las especificaciones del problema.

Estados	Codificación			
	W	X	Y	Z
A	0	0	0	0
B	0	0	0	1
C	0	0	1	0
D	0	0	1	1
E	0	1	0	0
F	0	1	0	1
G	0	1	1	0
H	0	1	1	1
I	1	0	0	0
J	1	0	0	1

El diagrama de bloques del contador se muestra en la figura 4.6.

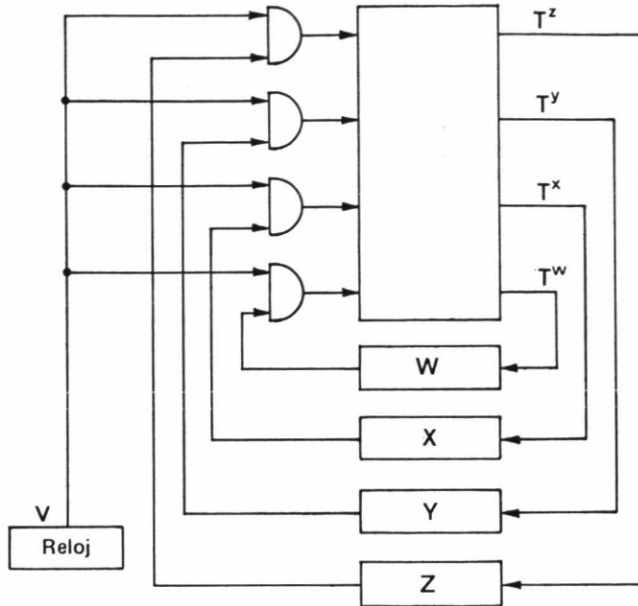


FIG. 4.6

El circuito combinatorio presenta cuatro variables de entrada (W, X, Y, Z) y cuatro variables de salida (T^w, T^x, T^y, T^z).

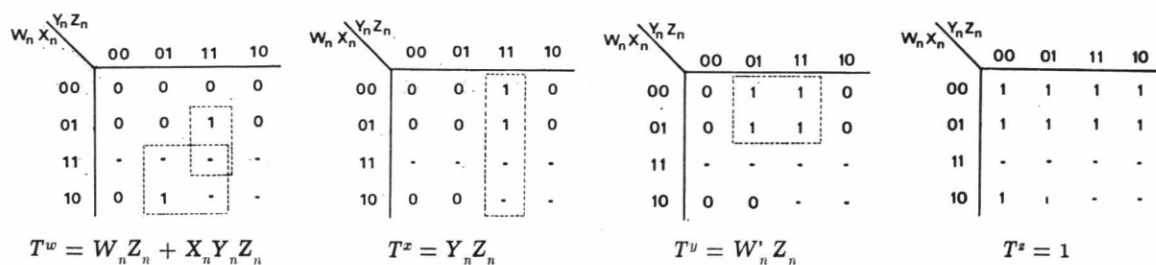
Una nueva tabla de transición de estados que puede obtenerse es la siguiente:

Estado presente				Próximo estado			
				$V_n = 1$			
W_n	X_n	Y_n	Z_n	W_{n+1}	X_{n+1}	Y_{n+1}	Z_{n+1}
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0

De esta tabla se pueden obtener las funciones: T^w , T^x , T^y y T^z . Para ello, se utilizará la siguiente tabla obtenida de la ecuación característica del «flip-flop» tipo T.

B_n	B_{n+1}	T_n
0	0	0
0	1	1
1	1	0
1	0	1

Los mapas que se obtienen para las funciones T^w , T^x , T^y y T^z son:



La materialización de las funciones:

$$T_n^w = W_n Z_n + X_n Y_n Z_n$$

$$T_n^x = Y_n Z_n$$

$$T_n^y = W_n' Z_n$$

$$T_n^z = 1$$

es bastante sencilla y se muestra en la figura 4.7.

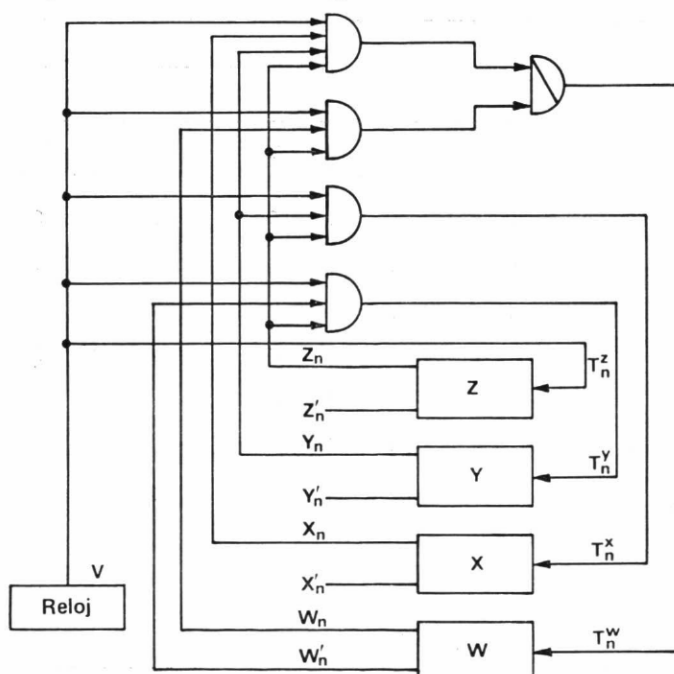


FIG. 4.7

4.8 DISEÑO DE UN GENERADOR DE BIT DE PARIDAD

Se pretende diseñar un generador de bit de paridad, tipo serie, para código *BCD*. Se utilizarán «flip-flops» tipo *RS* y puertas lógicas de diodos. Se empleará paridad *impar*.

El diagrama de bloques del circuito a diseñar se muestra en la figura 4.8.

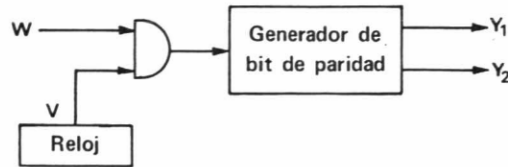


FIG. 4.8

Los cuatro bits de una secuencia se envían a través de la entrada *W*. Los impulsos que representan los bits se sincronizan con los impulsos del reloj y se envían consecutivamente al circuito. Se supone que el primer bit del ciclo que se recibe es el de orden inferior.

Cuando el circuito ha recibido el cuarto bit, se genera un bit de paridad que se añade a la secuencia original.

En otras palabras, el circuito examina cuatro bits sucesivos, produce una salida y se prepara para recibir una nueva secuencia de cuatro bits.

Existe, pues, una operación cíclica con un estado inicial.

1. DIAGRAMA DE MEALY

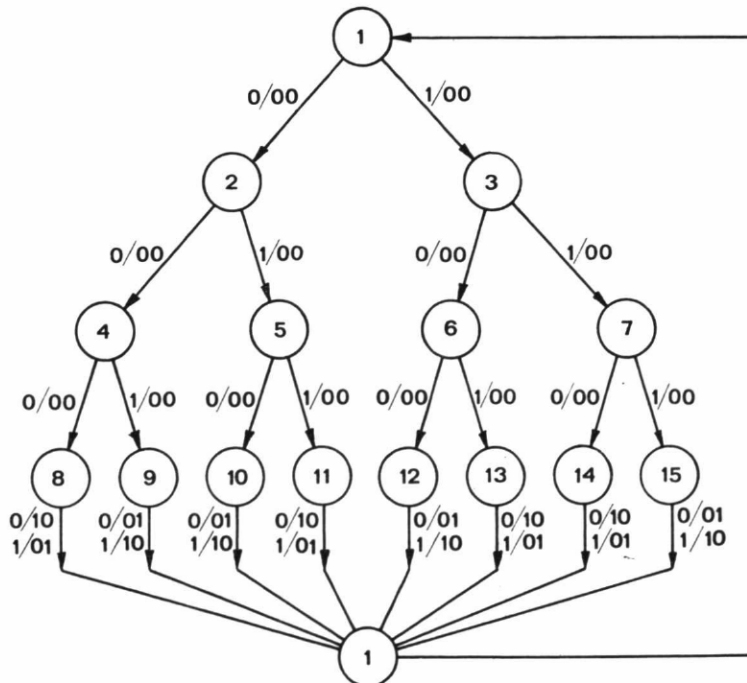


FIG. 4.9

El estado inicial se designa por 1. El circuito se encuentra en este estado cuando ha terminado de examinar cuatro bits y está preparado para comenzar un nuevo ciclo.

El primer bit del ciclo puede ser 0 ó 1. Cuando se recibe un 0 el estado del circuito cambia de 1 a 2; cuando recibe un 1, el estado del circuito pasa de 1 a 3.

El segundo bit del ciclo puede ser 1 ó 0. Con este segundo bit, el circuito alcanzará uno de los estados 4, 5, 6 ó 7.

El tercer bit transfiere al circuito a uno de los siguientes ocho estados: 8, 9, 10, 11, 12, 13, 14 ó 15.

Después de recibir el cuarto bit, el circuito vuelve al estado 1, ya que cuatro bits constituyen un ciclo.

Existen tres clases de salidas:

$Y_1 Y_2 = 00$, para indicar que no se ha recibido la secuencia de cuatro bits.

$Y_1 Y_2 = 10$, que denota que el bit de paridad que debe generarse es 1.

$Y_1 Y_2 = 01$, que indica que el bit de paridad para la secuencia de cuatro bits considerada es 0.

Por ejemplo, si el estado del circuito es 10 y recibe un 0, se tiene:

Secuencia	Salida	Bit de paridad
0 0 1 0	01	0
1 0 1 0	10	1

Al utilizar el código *BCD*, las combinaciones 1010, 1011, 1100, 1101, 1110, 1111 no se presentan.

El diagrama de estados definitivos se indica en la figura 4.10.

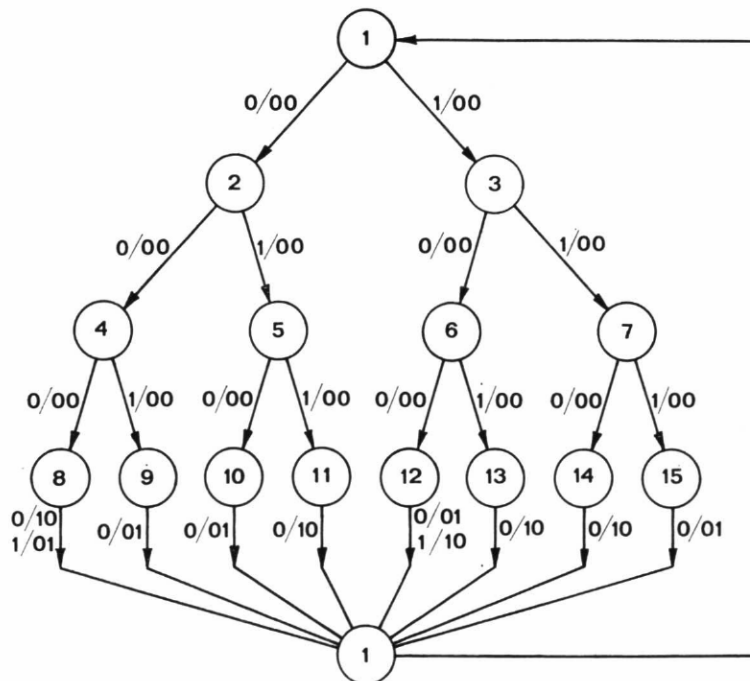


FIG. 4.10

2. TABLA DE TRANSICIÓN DE ESTADOS

Estado presente	Próximo estado		Próxima salida	
	0	1	0	1
1	2	3	00	00
2	4	5	00	00
3	6	7	00	00
4	8	9	00	00
5	10	11	00	00
6	12	13	00	00
7	14	15	00	00
8	1	1	10	01
9	1	1	01	10
10	1	1	01	10
11	1	1	10	01
12	1	1	01	10
13	1	1	10	01
14	1	1	10	01
15	1	1	01	10

La tabla de transición que se obtiene al utilizar el código *BCD* es:

Estado presente	Próximo estado		Próxima salida	
	0	1	0	1
1	2	3	00	00
2	4	5	00	00
3	6	7	00	00
4	8	9	00	00
5	10	11	00	00
6	12	13	00	00
7	14	15	00	00
8	1	1	10	01
9	1	—	01	—
10	1	—	01	—
11	1	—	10	—
12	1	1	01	10
13	1	—	10	—
14	1	—	10	—
15	1	—	01	—

Las condiciones no especificadas (—) pueden usarse para simplificar el diseño resultante.

3. TABLA DE ESTADOS REDUCIDA

La tabla de estados reducida coincide con la tabla 4.5.4.

Estado presente	Próximo estado		Salida	
	$W = 0$	$W = 1$	0	1
1	2	3	00	00
2	4	5	00	00
3	5	4	00	00
4	8	9	00	00
5	9	8	00	00
8	1	1	10	01
9	1	1	01	10

4. ASIGNACIÓN DE ESTADOS

Ya se ha visto que la tabla de estados reducida tiene solamente siete estados. Para almacenarlos se requieren, como mínimo, tres flip-flops». Sean estos X, Y y Z.

Con estos «flip-flops» se pueden obtener las siguientes combinaciones:

X	Y	Z
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

¿Cómo asignar *siete* de las *ocho* combinaciones anteriores a los *siete* estados que aparecen en la tabla de estados?

Hay que tener en cuenta que con cada esquema de asignación específica se obtendrá un circuito combinatorio.

Se debe intentar una asignación que proporcione el circuito combinatorio más idóneo. No ha sido, todavía, establecido un método sistemático para obtener el esquema de asignación idóneo.

Se ensayará, en primer lugar, la siguiente asignación:

	X	Y	Z
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
8	1	0	1
9	1	1	0
No usada	1	1	1

5. MATRIZ DE EXCITACIONES SECUNDARIAS. MATRIZ DE SALIDA

Mediante la aplicación del esquema anterior, se obtiene la siguiente tabla:

Estado presente			Próximo estado						Salida Y ₁ Y ₂	
			W _n = 0			W _n = 1				
X _n	Y _n	Z _n	X _{n+1}	Y _{n+1}	Z _{n+1}	X _{n+1}	Y _{n+1}	Z _{n+1}	W _n = 0	W _n = 1
0	0	0	0	0	1	0	1	0	00	00
0	0	1	0	1	1	1	0	0	00	00
0	1	0	1	0	0	0	1	1	00	00
0	1	1	1	0	1	1	1	0	00	00
1	0	0	1	1	0	1	0	1	00	00
1	0	1	0	0	0	0	0	0	10	01
1	1	0	0	0	0	0	0	0	01	10
1	1	1	—	—	—	—	—	—	—	—

El problema se limita al diseño de un circuito combinatorio con cuatro entradas (W_n, X_n, Y_n, Z_n) y ocho salidas (R^x, S^x, R^y, S^y, R^z, S^z, Y₁ e Y₂).

6. OBTENCIÓN DE LAS FUNCIONES

Aplicando el método del mapa se obtiene:

«FLIP-FLOP» x :

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	0	0	1	1
01	-	0	-	0
11	-	0	-	0
10	0	1	1	0

$$S^x = W_n X_n' Z_n + W_n' X_n' Y_n$$

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	-	-	0	0
01	0	1	-	1
11	0	1	-	1
10	-	0	0	-

$$R^x = X_n Z_n + X_n Y_n$$

«FLIP-FLOP» y :

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	0	1	0	0
01	1	0	-	0
11	0	0	-	0
10	1	0	-	-

$$S^y = W_n' X_n' Y_n' Z_n + W_n' X_n Y_n' Z_n + W_n X_n Z_n$$

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	-	0	1	1
01	0	-	-	1
11	-	-	-	1
10	0	-	0	0

$$R^y = X_n Y_n + W_n' Y_n$$

«FLIP-FLOP» z :

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	1	-	-	0
01	0	0	-	0
11	1	0	-	0
10	0	0	0	1

$$S^z = W_n' X_n' Y_n' + W_n X_n Y_n' Z_n + W_n X_n Y_n Z_n$$

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	0	0	0	-
01	-	1	-	-
11	0	1	-	-
10	-	1	1	0

$$R^z = X_n Z_n + W_n Z_n$$

Para obtener las funciones Y_1 e Y_2 hay que tener en cuenta la matriz de salida.

X_n	Y_n	Z_n	$Y_1 Y_2$	
			$W_n = 0$	$W_n = 1$
0	0	0	00	00
0	0	1	00	00
0	1	0	00	00
0	1	1	00	00
1	0	0	00	00
1	0	1	10	01
1	1	0	01	10

$$Y_1 = W_n' X_n Y_n' Z_n + W_n X_n Y_n Z_n$$

$$Y_2 = W_n' X_n Y_n Z_n' + W_n X_n Y_n' Z_n$$

Así, pues, las funciones que definen el comportamiento del circuito son:

$$\begin{aligned}
 S^z &= W_n X'_n Z_n + W'_n X'_n Y_n \\
 R^z &= X_n Z_n + X_n Y_n \\
 S^y &= W'_n X'_n Y'_n Z_n + W'_n X_n Y'_n Z'_n + W_n X'_n Z'_n \\
 R^y &= W'_n Y_n + X_n Y_n \\
 S^x &= W'_n X'_n Y'_n + W_n X_n Y'_n Z'_n + W_n X'_n Y_n Z'_n \\
 R^x &= X_n Z_n + W_n Z_n \\
 Y_1 &= W'_n X_n Y'_n Z_n + W_n X_n Y_n Z'_n \\
 Y_2 &= W_n X_n Y'_n Z_n + W'_n X_n Y_n Z'_n
 \end{aligned}$$

7. CONSIDERACIONES SOBRE EL ESQUEMA DE ASIGNACIÓN

Con la codificación realizada en el punto 4.8.4, se han obtenido las funciones que permiten materializar el circuito. Pero no se tiene seguridad de haber empleado el mejor esquema de asignación de estados. Se pretende encontrar una asignación de estados que conduzca a funciones más sencillas. Para ello, se construyen los dos mapas siguientes, llamados mapa P y mapa N.

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	1	2	4	3
01	5	8	-	9
11	5	8	-	9
10	1	2	4	3

Mapa de estados presentes.
Mapa P (*Present Map*)

$W_n X_n \backslash Y_n Z_n$	00	01	11	10
00	2	4	8	5
01	9	1	-	1
11	8	1	-	1
10	3	5	9	4

Mapa de estados futuros.
Mapa N (*Next Map*)

El mapa P es una representación gráfica del esquema de asignación realizado en 4.8.4.

	X	Y	Z
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
8	1	0	1
9	1	1	0
No usada	1	1	1

Por ejemplo, la combinación 000 se usa para representar el estado 1; por lo cual dos áreas (0000 y 1000) van a corresponder a dicho estado.

La combinación 111 no se usa; por lo cual las áreas 0111 y 1111 corresponden a condiciones opcionales.

La variable W se introduce en el mapa P para poderlo relacionar con el mapa N.

El mapa N se deriva del mapa P. Sus entradas están determinadas por la tabla de estados del problema.

Estado presente	Próximo estado		Salida	
	0	1	0	1
1	2	3	00	00
2	4	5	00	00
3	5	4	00	00
4	8	9	00	00
5	9	8	00	00
8	1	1	10	01
9	1	1	01	10

Por ejemplo, el área 0000 en el mapa P es 1; en la tabla de estado se ve que el próximo estado, para $W_n = 0$, es 2. Por este hecho, en el mapa N en el área 0000 aparece un 2.

En el mapa N, el estado 1 aparece en cuatro áreas. Cada área representa una combinación de entrada, en $t = t_n$, que transferirá el estado del circuito a 1, en $t = t_{n+1}$.

Si se reexaminan los mapas de R^x , R^y y R^z se ve que las cuatro áreas 0101, 0110, 1101 y 1110 contienen 1 ó —.

Si las cuatro áreas del mapa N donde 1 aparece son adyacentes y forman un rectángulo, las funciones R^x , R^y y R^z se simplifican. El esquema de asignación que se ha empleado no coloca estas cuatro áreas dentro de un rectángulo.

¿Qué se debe hacer para que estas cuatro áreas formen un rectángulo? En el mapa N, 1 aparece en las áreas 0101 y 1101, por ser 1 el próximo estado de 8; 1 aparece en las áreas 0110 y 1110, ya que 1 es el próximo estado de 9. Se tiene que lograr que las áreas 8 y 9, del mapa P, sean adyacentes y formen un rectángulo. Puede conseguirse haciendo que 8 y 9 sean adyacentes.

Generalizando, se pueden establecer las siguientes reglas:

a) REGLA 1.^a: *Dos o más estados que tienen el mismo próximo estado, se representarán por combinaciones adyacentes.*

Para hallarlos, se construye la siguiente tabla:

Próximo estado	Estados presentes
1	8,9
2	1
3	1
4	2,3
5	2,3
8	4,5
9	4,5

De acuerdo con esto, deben ser adyacentes:

- a) 2 y 3.
- b) 4 y 5.
- c) 8 y 9.

lo que representa: (2,3), (4,5), (8,9).

b) REGLA 2.^a *Dos o más estados que son los próximos estados de un estado, se representarán por combinaciones adyacentes.*

Estos estados son adyacentes en el mapa N.

Los estados que se buscan pueden ser detectados examinando las filas de la tabla de estados.

Se obtiene: (2,3), (4,5), (8,9).

En este ejemplo particular, los agrupamientos recomendados por las reglas 1 y 2 son los mismos. Esto no ocurre en todos los casos. Cuando ambos agrupamientos difieren, se adopta el obtenido a partir de la regla 1.

A continuación se establece un nuevo esquema de codificación.

	X	Y	Z	
1	0	0	0	
2	0	0	1	} Adyacentes
3	0	1	1	
4	1	1	1	} Adyacentes
5	1	1	0	
8	1	0	0	} Adyacentes
9	1	0	1	
NO USADA	0	1	0	

Resolviendo el problema para esta codificación, se obtienen las funciones:

$$S^x = X'_n Z'_n$$

$$R^x = X_n Y'_n$$

$$S^y = W_n X'_n + X'_n Z'_n$$

$$R^y = X_n$$

$$S^z = X'_n Z'_n + W'_n Y_n Z'_n$$

$$R^z = X_n Y'_n + W_n Y'_n Z_n + W'_n Y_n Z_n$$

$$Y_1 = W'_n X_n Y'_n Z'_n + W_n X_n Y'_n Z_n$$

$$Y_2 = W_n X_n Y'_n Z'_n + W'_n X_n Y'_n Z_n$$

que son más sencillas que las obtenidas en el punto 4.8.6; el esquema de asignación empleado es mejor.

Conviene tener presente que por aplicación de las reglas 1 y 2 no se consigue un esquema de asignación único. A continuación se va a emplear otro nuevo esquema de asignación.

	X	Y	Z	
1	0	0	0	
2	0	1	0	} Adyacentes
3	0	1	1	
4	1	1	1	} Adyacentes
5	1	1	0	
8	1	0	0	} Adyacentes
9	1	0	1	
NO USADA	0	0	1	

Para esta codificación se obtienen las funciones:

$$S^x = Y_n$$

$$R^x = Y'_n$$

$$S^y = X'_n$$

$$R^y = X_n$$

$$S^z = W_n X'_n Y'_n + W'_n Y_n Z'_n$$

$$R^z = X_n Y'_n + W'_n Z_n$$

$$Y_1 = W'_n X_n Y'_n Z'_n + W_n X_n Y'_n Z_n$$

$$Y_2 = W_n X_n Y'_n Z' + W'_n X_n Y'_n Z_n$$

Como puede observarse, este nuevo esquema de asignación es más adecuado que los dos anteriores.

4.9. DISEÑO DE UN CIRCUITO SECUENCIAL SIN ESTADO INICIAL

Hay circuitos secuenciales cuyas operaciones no son cíclicas; en ellos no existen estados iniciales. Esto, en principio, no complica la tarea de establecer la tabla de estados.

Se pretende diseñar un circuito secuencial para examinar un tren de dígitos binarios. La salida del circuito debe ser 1 cuando los cuatro últimos bits recibidos sean todos 1. Se emplearán «flip-flops» tipo *T* y puertas *NAND* y *NOR*.

1. DIAGRAMA DE MEALY

Supóngase que un hipotético tren de bits y las salidas del circuito son:

Instantes de tiempo	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	...
Bit de entrada	0	0	1	1	1	1	1	0	0	...
Salida	0	0	0	0	0	1	1	0	0	...

Existe, además, una demora antes de que el circuito produzca una salida. Por ejemplo, en t_6 no aparecerá exactamente la salida correspondiente.

Examinando las especificaciones del problema, se ve:

a) Que no existe estado inicial.

b) Que el circuito debe tener la capacidad de recordar o almacenar los últimos tres bits recibidos. Así, el circuito, al recibir el cuarto bit, podrá decidir cuál debe ser la salida. Por esto, los estados del circuito deben indicar las ocho posibles combinaciones de los tres últimos bits recibidos. Se requieren, pues, ocho estados.

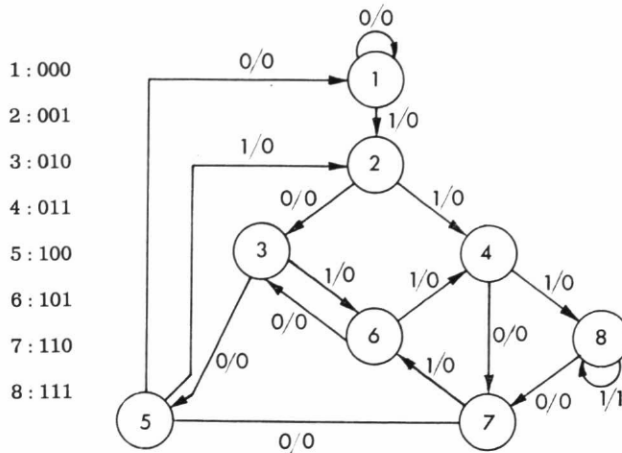


FIG. 4.11

Cada estado se designa por la combinación de los tres últimos bits recibidos.

2. TABLA DE TRANSICIÓN DE ESTADOS

Estado presente	Próximo estado		Próxima salida		
	0	1	0	1	
1	1	2	0	0	
2	1	4	0	0	
3	1	2	0	0	reducible con 1
4	1	8	0	0	
5	1	2	0	0	reducible con 1
6	3	4	0	0	reducible con 2
7	1	2	0	0	reducible con 3
8	1	8	0	1	

3. TABLA DE ESTADOS REDUCIDOS

El proceso de reducción se inicia localizando filas idénticas. Esto elimina los estados 3, 5, 6 y 7.

Estado presente	Próximo estado		Próxima salida	
	0	1	0	1
1	1	2	0	0
2	1	4	0	0
4	1	8	0	0
8	1	8	0	1

Examinando la tabla anterior, se observa que existen solamente dos diferentes modelos de salida: 00 y 01.

A continuación se intenta con dos estados cumplir lo indicado por las filas 1, 2, 4, 8:

I (0,0)	II (0,1)
1, 2, 4	8

Sustituyendo, resulta:

Estado presente	Próximo estado	
	0	1
I	I	I
I	I	I
I	I	II
II	I	II

En la tercera fila existe una contradicción. Se crea un nuevo estado III.

I (0,0)	II (0,1)	III (0,0)
1, 2	8	4

Sustituyendo, resulta:

Estado presente	Próximo estado	
	0	1
I	I	I
I	I	III
III	I	II
II	I	II

Una contradicción existe en la segunda fila. Se crea otro nuevo estado: IV.

I (0,0)	II (0,1)	III (0,0)	IV (0,0)
1	8	4	2

En otras palabras, la tabla resultante consta de cuatro filas. A este mismo resultado se llegaría utilizando el método de tabulación.

4. ASIGNACIÓN DE ESTADOS

Para ello se va a aplicar la regla primera («Dos o más estados que tienen el mismo próximo estado, se representarán por combinaciones adyacentes»).

Próximo estado	Estados presentes
1	1, 2, 4, 8
2	1
4	2
8	4, 8

El esquema de asignación a emplear puede ser:

	x	
	0	1
y	0	1
1	2	8

Solamente se requieren dos «flip-flops». El efecto de considerar la regla primera no es apreciable.

5. MATRIZ DE EXCITACIONES. MATRIZ DE SALIDA

Estado presente		Próximo estado				Salida	
		$W_n = 0$		$W_n = 1$		$W_n = 0$	$W_n = 1$
X_n	Y_n	X_{n+1}	Y_{n+1}	X_{n+1}	Y_{n+1}		
0	0	0	0	0	1	0	0
0	1	0	0	1	0	0	0
1	0	0	0	1	1	0	0
1	1	0	0	1	1	0	1

6. OBTENCIÓN DE LAS FUNCIONES

El problema se limita al diseño de un circuito combinatorio que tiene tres variables de entrada (W_n , X_n e Y_n) y cinco salidas (R^x , S^x , R^y , S^y y Z) caso de usar «flip-flops» tipo RS . En el caso de usar «flip-flops» tipo T , el circuito tendrá tres salidas (T^x , T^y y Z).

Para este último se sabe que:

$$B_{n+1} = B'_n T_n + B_n T'_n$$

B_n	B_{n+1}	T_n
0	0	0
0	1	1
1	1	0
1	0	1

Los mapas que se obtienen son los siguientes:

$W_n \backslash X_n Y_n$	00	01	11	10
0	0	0	1	1
1	0	1	0	0

$$T^x = W'_n X_n + W_n X'_n Y_n$$

$W_n \backslash X_n Y_n$	00	01	11	10
0	0	1	1	0
1	1	1	0	1

$$T^y = W'_n Y_n + W_n X'_n + W_n Y'_n$$

Para obtener la función de salida hay que tener en cuenta:

Estado presente		Z	
X_n	Y_n	$W_n = 0$	$W_n = 1$
0	0	0	0
0	1	0	0
1	0	0	0
1	1	0	1

con lo cual, $Z_n = W'_n X_n Y_n$.

7. DIAGRAMA LÓGICO

El diagrama lógico del circuito se muestra en la figura 4.12.

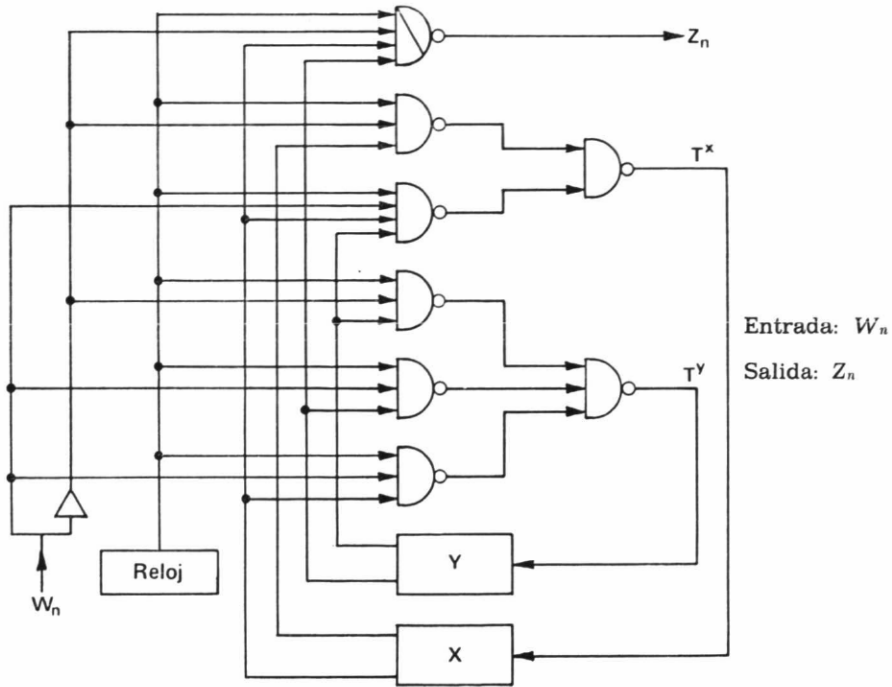


FIG 4.12

4.10. REGISTRO DE DESPLAZAMIENTO («SHIFT REGISTER»)

El registro de desplazamiento es un tipo de circuito secuencial muy sencillo e importante. Es una cadena de «flip-flops» conectados de tal manera que al recibir un impulso de desplazamiento, el contenido de un «flip-flop» se transmite al de su derecha (o izquierda), mientras recibe el contenido del situado a su izquierda (o derecha).

En la figura 4.13 se representa la forma idealizada de los impulsos de desplazamiento.

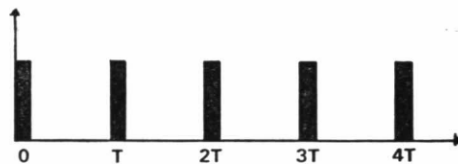


FIG. 4.13

Un diagrama de bloques de un registro de desplazamiento a la derecha, constituido por cuatro células, se muestra en la figura 4.14.

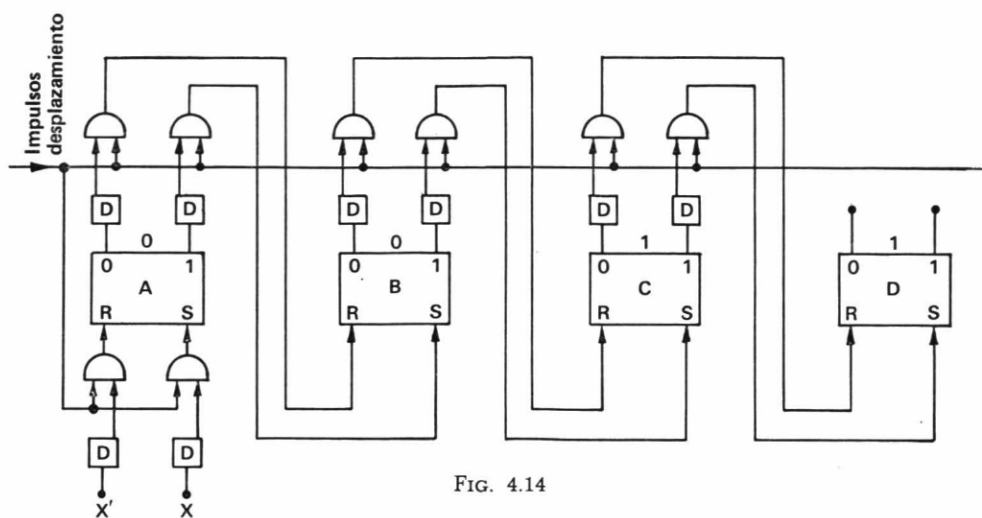


FIG. 4.14

Las células de retardo *D*, conectadas en los terminales de salida de los «flip-flops», permiten que cada uno transmita su contenido a la célula situada a su derecha, mientras está recibiendo el contenido de la célula situada a su izquierda; es decir, es posible aplicar una entrada a un «flip-flop» y al mismo tiempo usar su salida. La entrada presente no afecta a la salida presente, sino solamente al próximo estado.

Por ejemplo, el «flip-flop» *B* es «set» cada vez que ocurre un impulso de desplazamiento si el «flip-flop» *A* está en el estado 1 y es «reset» cada vez que se presenta un impulso de desplazamiento si el «flip-flop» *A* está en el estado 0. Admitase que en $t = 0$: $A = 0, B = 0, C = 1, D = 1$ y que los impulsos de desplazamiento se aplican en $t = 0, 1, 2, \dots$. El comportamiento del dispositivo para los valores de entrada X ($t = 0$) = 1, X ($t = 1$) = 0, X ($t = 2$) = 1, etc., es el siguiente:

t	X	A	B	C	D
0	1	0	0	1	1
1	0	1	0	0	1
2	1	0	1	0	0
3	1	1	0	1	0
4	—	1	1	0	1

4.11 MULTIPLICADOR BINARIO

4.11.1 Generación de productos parciales

Los dos operandos de una multiplicación pueden ser definidos como sigue:

$$\begin{aligned} \text{Multiplicando: } N_1 &= a_{n-1} a_{n-2} a_{n-3} \dots a_1 a_0 \\ \text{Multiplicador: } N_2 &= b_{n-1} b_{n-2} b_{n-3} \dots b_1 b_0 \end{aligned}$$

Básicamente, la multiplicación es un proceso de adición repetido. Sin embargo, a fin de aumentar la velocidad de operación, en los ordenadores modernos se utilizan otros esquemas. Uno de ellos es muy similar al realizado cuando se resuelve una multiplicación con papel y lápiz.

Para ello, supóngase que el multiplicando es 1101 y el multiplicador 1010. El primer paso consiste en multiplicar el multiplicando por el dígito menos significativo del multiplicador, en este caso cero. Así se obtiene el primer producto parcial 0000.

El segundo dígito contado a partir de la derecha del multiplicador, 1, es multiplicado por el multiplicando y se obtiene como segundo producto parcial 1101.

El segundo producto parcial es desplazado un dígito hacia la izquierda y añadido al primer producto parcial para producir la primera suma parcial.

Este proceso se repite hasta obtener el resultado final.

$$\begin{array}{r} 1101 \\ 1010 \\ \hline 0000 \quad \leftarrow \text{Primer producto parcial.} \\ 1101 \quad \leftarrow \text{Segundo producto parcial desplazado un dígito hacia la izquierda.} \\ \hline 11010 \quad \leftarrow \text{Primera suma parcial.} \\ 1101 \\ \hline 10000010 \quad \leftarrow \text{Resultado.} \end{array}$$

El proceso de multiplicación implica:

- a) La operación de desplazamiento a la izquierda.
- b) Las operaciones de suma subsiguientes.

Está claro que un multiplicador binario necesita un circuito de conmutación para generar los productos parciales, unidades que los almacenen y los desplacen a la izquierda, y un sumador binario.

El diagrama esquemático de un circuito de conmutación para generar el producto parcial se indica en la figura 4.15.

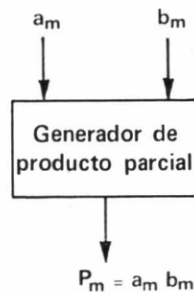


FIG. 4.15

4.11.2 Diagrama de bloques de un multiplicador binario

El diagrama de bloques de un multiplicador binario que es capaz de procesar un multiplicando y un multiplicador de tres bits cada uno, se muestra en la figura 4.16.

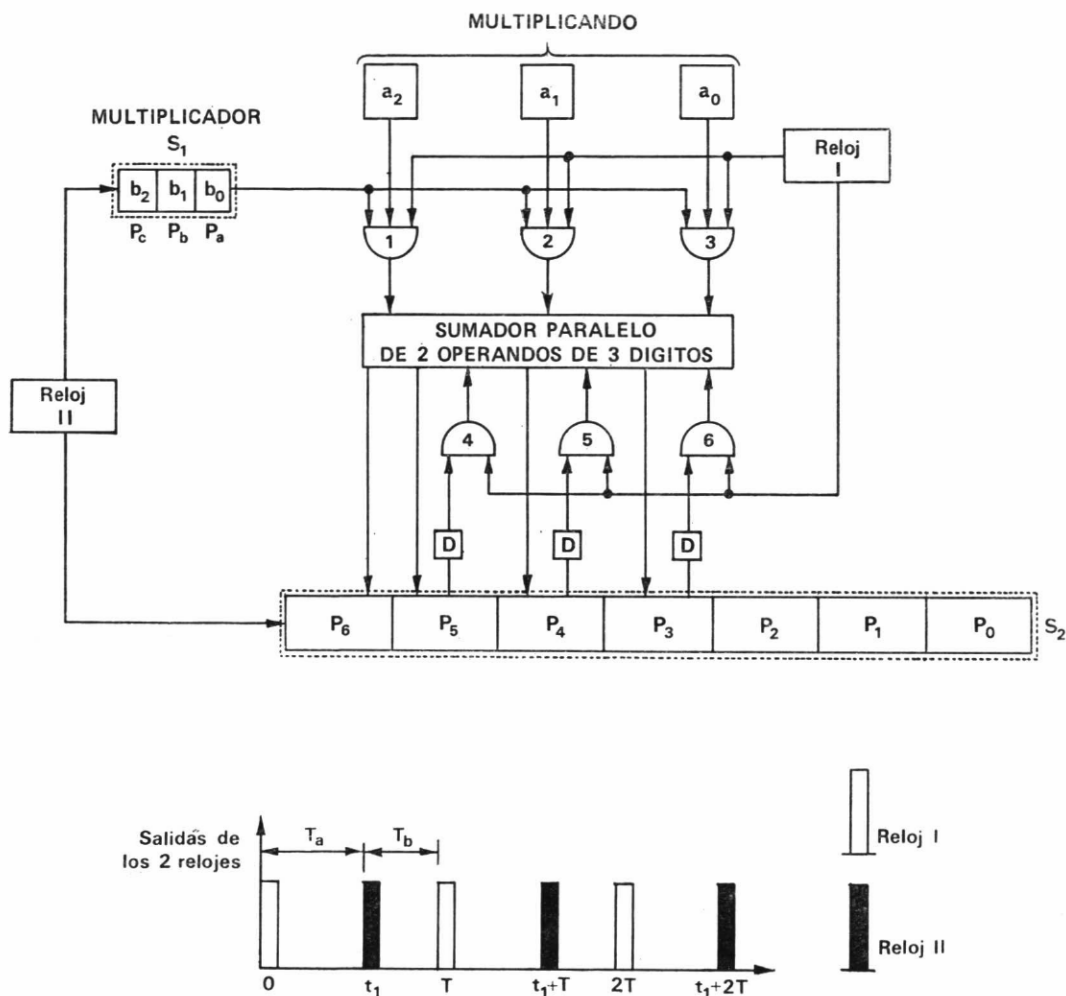


FIG. 4.16

Al comienzo de la operación de multiplicación, los tres bits del multiplicando (a_2, a_1, a_0) están almacenados en tres «flip-flops». Los tres bits del multiplicador están almacenados en los tres «flip-flops» de un registro de desplazamiento (S_1).

S_2 es un registro de desplazamiento de siete bits. Al comienzo de la operación todos los bits almacenados en él son 0.

La operación de multiplicación puede estudiarse mejor con un ejemplo. Sea el multiplicando 111 y el multiplicador 101. El proceso que se sigue se expresa en la tabla siguiente:

	Tiempo	Contenido de S_2
<i>Primer ciclo:</i>		$P_6 P_5 P_4 P_3 P_2 P_1 P_0$
Multiplicando a_2, a_1, a_0 1 1 1	<div style="border: 1px solid black; padding: 2px;">$t = 0$</div>	0 0 0 0 0 0
Multiplicador b_2, b_1, b_0 1 0 1		
Primer producto parcial 1 1 1	<div style="border: 1px solid black; padding: 2px;">$t = t_1$</div>	0 1 1 1 0 0 0
Contenido de P_5, P_4, P_3 0 0 0		
Primera suma acumulada 1 1 1	<div style="border: 1px solid black; padding: 2px;">$t = T$</div>	0 0 1 1 1 0 0
Desplazamiento a la derecha de un bit en S_1 y S_2		
<i>Segundo ciclo:</i>	<div style="border: 1px solid black; padding: 2px;">$t = t_1 + T$</div>	0 0 0 1 1 1 0
Segundo producto parcial 0 0 0		
Contenido de P_5, P_4, P_3 0 1 1	<div style="border: 1px solid black; padding: 2px;">$t = 2T$</div>	0 0 0 1 1 1 0
Segunda suma acumulada 0 1 1		
Desplazamiento a la derecha de un bit en S_1 y S_2	<div style="border: 1px solid black; padding: 2px;">$t = t_1 + 2T$</div>	0 0 0 1 1 1 0
<i>Tercer ciclo:</i>		
Tercer producto parcial 1 1 1	<div style="border: 1px solid black; padding: 2px;">$t = t_1 + 2T$</div>	1 0 0 0 1 1 0
Contenido de P_5, P_4, P_3 0 0 1		
Tercera suma acumulada 1 0 0 0		0 1 0 0 0 1 1
Desplazamiento a la derecha de un bit en S_1 y S_2 (S_1 queda en blanco y S_2 contiene el producto.)		

Hay varios puntos que necesitan aclaración.

El tiempo indicado en el lado derecho de la tabla corresponde al expresado en la figura 4.16 para la salida de los dos relojes.

En $t = 0$, un impulso procedente del reloj I se aplica a las seis puertas *AND*. A través de las puertas *AND* 1, 2 y 3, se obtiene el primer producto parcial y se envía al sumador. Los bits almacenados en P_5, P_4 y P_3 de S_2 se envían al sumador a través de las puertas *AND* 4, 5 y 6. La primera suma acumulada, 111, es reenviada a las células P_5, P_4 y P_3 . Para evitar que estos bits produzcan errores, se introducen células de retardo en las salidas de P_5, P_4 y P_3 . (La anchura finita de los impulsos del reloj I también contribuye a evitar errores.)

En $t = t_1$, un impulso del reloj II se envía a todos los «flip-flops» de S_1 y S_2 . Cuando S_1 desplaza a la derecha los bits contenidos en él, en los tres «flip-flops» P_c , P_b y P_a se tiene almacenado 010, respectivamente. El propósito de realizar este desplazamiento a la derecha, es mover b_1 para generar el segundo producto parcial. El contenido de S_2 , al ser desplazado a la derecha, es ahora 0011100. ¿Por qué es necesario desplazar el contenido de S_2 a la derecha? Para colocar la primera suma acumulada en una posición tal que permita que sea sumada al segundo producto parcial. (Cuando se realiza la multiplicación con lápiz y papel, se desplaza a la izquierda el producto parcial.) Esto completa un ciclo.

Se requieren dos ciclos más para completar la multiplicación pedida. El producto final se obtiene en S_2 .

Cuando el sumador paralelo y el registro de desplazamiento S_2 se combinan en un único circuito, se obtiene un acumulador.

4.12 EJEMPLO

Reducir, por el método de tabulación, la siguiente tabla de estados:

Estado presente	Próximo estado		Salida	
	0	1	0	1
1	1	2	0	0
2	3	4	0	0
3	5	6	0	0
4	7	8	0	0
5	1	2	0	0
6	3	4	0	0
7	5	6	0	0
8	7	8	0	1

Para averiguar las equivalencias entre estados, se construye la tabla de implicaciones:

2	$\frac{1 \times 3}{2 \times 4}$						
3	$\frac{1 \times 5}{2 \times 6}$	$\frac{3 \times 5}{4 \times 6}$					
4	$\frac{1 \times 7}{2 \times 8}$	$\frac{3 \times 7}{4 \times 8}$	$\frac{5 \times 7}{6 \times 8}$				
5	✓	$\frac{1 \times 3}{2 \times 4}$	$\frac{1 \times 5}{2 \times 6}$	$\frac{1 \times 7}{2 \times 4}$			
6	$\frac{1 \times 3}{2 \times 4}$	✓	$\frac{3 \times 5}{4 \times 6}$	$\frac{3 \times 7}{4 \times 8}$	$\frac{1 \times 3}{2 \times 4}$		
7	$\frac{1 \times 5}{2 \times 6}$	$\frac{3 \times 5}{4 \times 6}$	✓	$\frac{5 \times 7}{6 \times 8}$	$\frac{1 \times 5}{2 \times 6}$	$\frac{3 \times 5}{4 \times 6}$	
8	xx	xx	xx	xx	xx	xx	xx
	1	2	3	4	5	6	7

Agrupamientos posibles:

7	
6	
5	5-7
4	5-7
3	3-5-7
2	2-6, 3-5-7
1	2-6, 1-3-5-7
<i>Final</i>	2-6, 1-3-5-7, 4-8

La columna 3 sugiere 3-5 y 3-7; la fila 3 tendría 3-5, 3-7 y 5-7. Se agrupan en 3-5-7. La columna 1 sugiere 1-3, 1-5 y 1-7. La fila 1 tendría 2-6, 1-3, 1-5, 1-7 y 3-5-7. Se agrupan en 2-6 y 1-3-5-7.

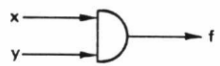

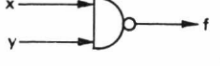


Selección de agrupamientos:

- a) Cada estado de la tabla de estado original tiene que estar incluido, por lo menos, en uno de los agrupamientos seleccionados.
- b) Los agrupamientos que se seleccionen deber ser escogidos de forma que todas las implicaciones necesarias sean satisfechas:
 - 2-6 no requiere ninguna implicación.
 - 1-3 requiere que 1-5 y 2-6 pertenezcan a agrupamientos seleccionados.

La selección es, pues: 2-6, 1-3-5-7, 4 y 8.

APENDICE

Símbolos lógicos utilizados

PUERTAS	SIMBOLOS	FUNCIONES
<i>AND</i>		$f = xy$
<i>OR</i>		$f = x + y$
<i>NAND</i>		$f = (xy)' = x' + y'$
<i>NOR</i>		$f = (x + y)' = x' y'$
<i>NOT</i> <i>(Inversor)</i>		$f = x'$

BIBLIOGRAFIA

1. SAMUEL H. CADWELL: *Les circuits de conmutation et leur conception*. Editions Eyrolles, 1967.
2. MARCOVITS y PUGSLEY: *An Introduction to switching system design*. Wiley and Sons, 1971.
3. TAYLOR L. BOOTH: *Digital networks and computer systems*. Wiley and Sons, 1971.
4. H. C. TORNG: *Introduction of the logical design of switching systems*. Addison Wesley, 1964.
5. DOUGLAS LEWIN: *Theory and design of digital computers*. Nelson and Sons, 1972.
6. G. BOULAYE: *Logique et organes des calculatrices numériques*. Dunod, 1970.
7. JUAN KAGI REYMANN: *Teoria de la conmutación. Circuitos secuenciales*. E.T.S. de Ingenieros de Telecomunicación, 1970.
8. *Fundamental principles of switching circuits and systems*. American Telephone and Telegraph, 1961.
9. L. NASHELSKY: *Teoria de las calculadoras numéricas automáticas*. Editorial Alhambra, 1970.
10. SCHOLTEN: *Logical Circuitry I*. Philips International Institute, 1968.
11. RAMOS y PÉREZ-POLO: *Circuitos de conmutación*. E.T.S. de Ingenieros Industriales, 1969.
12. GERARD SCHMITT: *Introduction to switching technology*. R. Oldenbourg, 1965.
13. M. KRIEGER: *Basic switching theory*. McMillan Co., 1967.
14. TAYLOR L. BOOTH: *Sequential machines and automata theory*. Wiley and Sons, 1967.
15. MATTHEW MANDI: *Circuitos electrónicos de conmutación*. Marcombo, S. A., 1972.
16. DEBRAINE: *Machines de traitement de l'information* (Circuits et programmes). Tome I: Étude logique et construction des circuits. Maisson et Cie, Editeurs, 1967.
17. J. P. OEHMINCHEN: *Emploi rationnel des circuits intégrés*. Editions Radio, 1971.

INDICE

Página

PRIMERA PARTE

CIRCUITOS COMBINATORIOS

CAPITULO 1. INTRODUCCION:

1.1	Sistemas de conmutación	15
1.2	Tipos de lógica	15

CAPITULO 2. ALGEBRA DE CONMUTACION:

2.1	Variable de conmutación y valor verdadero	17
2.2	Operaciones en álgebra de conmutación	17
2.3	Postulados y teoremas	18
2.4	Funciones de conmutación y tablas de verdad	19
2.5	Puertas lógicas	21
2.6	Conversión lógica	25
2.6.1	Lógica AND-NOT	25
2.6.2	Lógica OR-NOT	26
2.6.3	Lógica NAND	26
2.6.4	Lógica NOR	27
2.7	Principio de dualidad	27

CAPITULO 3. FORMULACION DE FUNCIONES DE CONMUTACION:

3.1	Productos <i>standards</i> . Forma disyuntiva	29
3.2	Sumas <i>standards</i> . Forma conjuntiva	30
3.3	Conversión entre formas básicas.....	31

CAPITULO 4. SIMPLIFICACION DE FUNCIONES DE CONMUTACION:

4.1	Representación gráfica de variables y funciones: Método de Veitch-Karnaugh.....	35
4.2	Simplificación de funciones	37

	Página
4.3 Utilización del mapa de Karnaugh para simplificar funciones expresadas en forma de producto de sumas	38
4.4 Mapa de Karnaugh para cinco variables	39
4.5 Método de tabulación de Quine-Mc Cluskey	40
 CAPITULO 5. RELE LOGICO:	
5.1 Representación. Puertas AND y OR	45
5.2 Optimización de circuitos	47
5.3 Redes en puente	50
 CAPITULO 6. DIODO LOGICO:	
6.1 Representación y características de los diodos de conmutación	53
6.2 Puertas AND y OR	55
6.3 Efecto de una impedancia de carga finita	60
6.4 Optimización de circuitos	63
6.5 Problema de la coincidencia de impulsos	64
 CAPITULO 7. TRANSISTOR LOGICO:	
7.1 Introducción	67
7.2 Transistor lógico	74
7.3 Transistor inversor	76
7.4 Cálculo de un inversor	78
7.5 Puertas diodo-transistor (DTL)	81
7.6 Puertas AND y OR (seguidor de emisor)	85
7.7 Transistores directamente acoplados (DCTL)	88
7.8 Tiempos de conmutación del transistor	91
7.9 Diseño con puertas NAND y NOR transistorizadas	93
7.10 Conclusiones	94
APÉNDICE 1. Teorema de Thevenin	95
APÉNDICE 2. Funciones AND y OR cableadas	96
 CAPITULO 8. REDES MULTITERMINALES:	
8.1 Introducción	99
8.2 Red en estrella	100
8.3 Red en triángulo	103
8.4 Redes en árbol	106
8.5 Codificadores	112
8.6 Decodificadores	113
8.7 Conversores de código	114
 CAPITULO 9. CIRCUITOS SUMADORES Y SUSTRACTORES:	
9.1 Suma y sustracción binarias	117
9.2 Semisumador	119
9.3 Sumador completo	120
9.4 Semisustractor	124
9.5 Sustractor completo	124
9.6 Sumador decimal	127

CAPITULO 10. REDES EN CASCADA:

10.1	Introducción	129
10.2	Células básicas	130
10.3	Fases del diseño de un circuito en cascada	130
10.4	Circuito de comprobación de paridad	132
10.5	Circuito para la determinación del número de «unos»	136
10.6	Sumador binario paralelo	139
10.7	Sumador decimal paralelo	139

BIBLIOGRAFIA	143
--------------------	-----

SEGUNDA PARTE

CIRCUITOS SECUENCIALES

CAPITULO 1. INTRODUCCION:

1.1	Circuitos combinatorios	149
1.2	Circuitos secuenciales	150
1.3	Circuitos secuenciales síncronos	151
1.4	Circuitos secuenciales asíncronos	152

CAPITULO 2. MEMORIAS:

2.1	Memoria binaria («flip-flop»)	153
2.2	«Flip-flop» tipo T	154
2.3	«Flip-flop» tipo RS	155
2.4	«Flip-flop» tipo JK	157
2.5	«Flip-flop» tipo RST	158
2.6	«Flip-flop» tipo RT	160
2.7	«Flip-flop» tipo ST	161
2.8	Memoria transicional (<i>transition memory</i>)	161

CAPITULO 3. CIRCUITOS SECUENCIALES ASINCRONOS:

3.1	Diagrama de bloques	163
3.2	Diseño de circuitos secuenciales asincronos	164
3.3	Matriz de salida	165
3.4	Matriz de direcciones o tabla de estados	165
3.5	Algunas tablas de estados	167
3.6	Matriz de excitaciones	168
3.6.1	Análisis de la matriz de excitaciones	169
3.6.2	Carreras (<i>racing problem</i>)	170

	Página
3.7 Diagramas de estados (<i>state diagram o flow diagram</i>)	171
3.7.1 Diagrama de Mealy	171
3.7.2 Diagrama de Moore	173
3.8 Fusión de filas (<i>row merging</i>)	174
3.9 Diagrama de fusión (<i>merger diagram</i>)	177
3.10 Codificación de la matriz de fases.....	178
3.11 Obtención de las funciones que definen las excitaciones secundarias	180
3.12 El azar en los circuitos de conmutación	182
3.13 Resumen	183
3.14 Ejemplos	184
CAPITULO 4. CIRCUITOS SECUENCIALES SINCRONOS:	
4.1 Diagrama de bloques	191
4.2 Diseño de circuitos secuenciales síncronos	193
4.3 Estados equivalentes	194
4.4 Estados equivalentes representados por filas no idénticas	196
4.5 Método de tabulación para la reducción de tablas de estados	198
4.6 Diseño de la memoria	202
4.7 Diseño de contadores	203
4.7.1 Contador binario hexadecimal	203
4.7.2 Contador binario decimal	208
4.8 Diseño de un generador de bit de paridad	211
4.9 Diseño de un circuito secuencial sin estado inicial	220
4.10 Registro de desplazamiento (<i>shift register</i>)	225
4.11 Multiplicador binario	227
4.11.1 Generación de productos parciales	227
4.11.2 Diagrama de bloques de un multiplicador binario	228
4.12 Ejemplo	230
APÉNDICE: Símbolos lógicos utilizados	232
BIBLIOGRAFIA	233

