

Fundamentos de Diseño Digital

M.C. César Augusto Leal Chapa

UNIVERSIDAD AUTONOMA DE
NUEVO LEON

FACULTAD DE INGENIERIA MECANICA Y ELECTRICA

FUNDAMENTOS DE DISEÑO DIGITAL

P R O L O G O

Este Texto nació como un primer intento para condensar en un solo -- volúmen los aspectos fundamentales del diseño de circuitos digitales y -- que cumpliera además con el programa de la clase de Electrónica Lógica I, impartida a nivel de Licenciatura, en las carreras de Ingeniero en Con--- trol y Computación e Ingeniero en Electrónica y Comunicaciones, de nues-- tra Facultad.

La Terminología usada en este texto aparece en muchas ocasiones en -- inglés. Esto se debe en ciertos casos a la ausencia de algún término en -- nuestro idioma y en otros casos se deberá a la confusión que crearía una -- traducción, cuya interpretación a menudo cambia regionalmente. En cual-- quiera de los dos casos creo que es justificable perder el prejuicio del -- idioma y usar la terminología en su lengua original.

Es conveniente mencionar también que la edición de este Texto respon -- de al Programa de Elaboración de apuntes y textos de clase e instructivos -- para prácticas de laboratorios, promovidos por la Secretaría Técnica de -- nuestra Facultad.

Por último quiero agradecer de una manera muy especial la contribu-- ción del equipo de trabajo formado por, Claudia Lozano, Tomás Sánchez V. -- Vicente de la Garza, Juan Carlos Flores: quienes prestaron su Servicio -- Social como Auxiliares del Departamento de Control y Computación y al es-- fuerzo de Irma Jaime por su labor de Taquimecanografía.

César A. Leal Chapa

C O N T E N I D O

| | | |
|------------|--|----|
| PROLOGO | ----- | |
| CAPITULO 1 | <u>LOS SISTEMAS DIGITALES SE ORIGINARON EN UN MUNDO ANALOGICO.</u> | |
| 1.0 | Conceptos Básicos | 7 |
| 1.1 | Conceptos de Resolución y Exactitud | 9 |
| 1.2 | Que es un Sistema Digital | 10 |
| 1.3 | Sistemas Contínuos y No-Contínuos | 11 |
| 1.4 | Representación de Información y Cantidad | 11 |
| | Problemas Propuestos | 14 |
| CAPITULO 2 | <u>SISTEMAS NUMERICOS</u> | |
| 2.0 | Introducción | 15 |
| 2.1 | Sistemas Numéricos de Notación Posicional | 16 |
| 2.2 | Sistema Numérico Binario | 21 |
| 2.2.1 | Conversión de Binario a Decimal | 22 |
| 2.2.2 | Conversión de Decimal a Binario | 22 |
| 2.3 | Sistema Numérico Octal | 25 |
| 2.3.1 | Conversión Octal a Decimal | 25 |
| 2.3.2 | Conversión de Decimal a Octal | 26 |
| 2.4 | Sistema Numérico Hexadecimal | 26 |
| 2.4.1 | Conversión de Hexadecimal a Decimal | 26 |
| 2.4.2 | Conversión de Decimal a Hexadecimal | 27 |
| 2.5 | Conversión Binario ↔ Octal | 28 |
| 2.6 | Conversión Binario ↔ Hexadecimal | 29 |
| 2.7 | Conversión Octal ↔ Hexadecimal | 30 |
| 2.8 | Aritmética Binaria, Octal y Hexadecimal | 31 |
| 2.8.1 | Suma Binaria | 32 |
| 2.8.2 | Suma Octal | 34 |
| 2.8.3 | Suma Hexadecimal | 34 |
| 2.8.4 | Resta | 35 |
| 2.8.5 | Resta Binaria | 35 |
| 2.8.6 | Dos Complemento | 36 |
| 2.8.7 | Resta Octal | 37 |

| | |
|---|----|
| 2.8.8 Resta Hexadecimal - - - - - | 38 |
| 2.8.9 Multiplicación y División - - - - - | 39 |
| Problemas Propuestos - - - - - | 44 |

CAPITULO 3 ALGEBRA BOOLEANA

| | |
|---|----|
| 3.0 Introducción - - - - - | 47 |
| 3.1 Operadores Lógicos - - - - - | 48 |
| 3.1.1 Operador Lógico AND - - - - - | 48 |
| 3.1.2 Operador Lógico OR - - - - - | 50 |
| 3.1.3 Operador Lógico NOT - - - - - | 51 |
| 3.1.4 Operador Lógico EX-OR (Exclusivo-Or)- - - - - | 52 |
| 3.1.5 Operador Lógico NAND - - - - - | 53 |
| 3.1.6 Operador Lógico NOR - - - - - | 54 |
| 3.1.7 Operador Lógico COINCIDENCE - - - - - | 55 |
| 3.2 Expresiones Booleanas - - - - - | 57 |
| 3.3 Propiedades Fundamentales del Algebra Booleana - - - | 60 |
| 3.4 Teorema de D'Morgan - - - - - | 61 |
| 3.5 La Forma "A.O.N." - AND, OR, NOT - - - - - | 62 |
| 3.6 Expresión de Funciones Booleanas a partir de NAND y - NOR. - - - - - | 63 |
| 3.7 Origen de las Funciones Booleanas, Minitérminos - - - | 68 |
| 3.8 \bar{F} como Alternativa, Maxitérminos - - - - - | 74 |
| 3.9 Las ocho Formas Estandar - - - - - | 76 |
| 3.9 Las ocho Formas Estandar - - - - - | 76 |
| Problemas Propuestos - - - - - | 79 |

CAPITULO 4 CODIGOS Y REPRESENTACION DE INFORMACION

| | |
|---|-----|
| 4.0 Introducción - - - - - | 83 |
| 4.1 Códigos Pesados - - - - - | 85 |
| 4.2 Códigos Numéricos más Usados - - - - - | 88 |
| 4.3 Códigos No-Pesados-Código Gray - - - - - | 90 |
| 4.4 Códigos Alfanuméricos - - - - - | 94 |
| 4.5 Detección de Errores (Paridad) - - - - - | 99 |
| 4.6 Números con Signo - - - - - | 100 |
| 4.7 Sumas y Restas con Números con Signos - - - - - | 101 |
| Problemas Propuestos - - - - - | 103 |

| | | |
|------------|--|-----|
| CAPITULO 5 | <u>MINIMIZACION DE FUNCIONES BOOLEANAS</u> | |
| | 5.0 Introducción - - - - - | 107 |
| | 5.1 Criterio de Costo - - - - - | 108 |
| | 5.2 Manipulación Algebraica - - - - - | 109 |
| | 5.2.1 Factorización - - - - - | 109 |
| | 5.2.2 Duplicando un Término ya Existente - - - - - | 110 |
| | 5.2.3 Multiplicando por un Término del Tipo $(a+\bar{a})$ - - - - - | 111 |
| | 5.2.4 Aplicando la Ley Distributiva - - - - - | 111 |
| | 5.3 Mapas de Karnaugh - - - - - | 111 |
| | 5.3.1 Reducción de Expresiones Booleanas usando el Mapa de Karnaugh. - - - - - | 114 |
| | 5.3.2 Productos de Sumatorias a partir de un Mapa de Karnaugh. - - - - - | 126 |
| | 5.3.3 Mapas de Karnaugh de 5 y 6 Variables - - - - - | 127 |
| | Problemas Propuestos - - - - - | 131 |
| CAPITULO 6 | <u>DISEÑO COMBINACIONAL</u> | |
| | 6.0 Definición de un Bloque Combinacional - - - - - | 133 |
| | 6.1 Metodología de Diseño Combinacional - - - - - | 134 |
| | 6.2 Ejemplos de Diseño - - - - - | 136 |
| | 6.3 Sistemas que no están completamente especificados - - - - - | 142 |
| | 6.4 Display de 7 segmentos - - - - - | 145 |
| | 6.5 Decodificadores - - - - - | 149 |
| | 6.6 Sistemas Combinacionales con salidas Múltiples - - - - - | 157 |
| | Problemas Propuestos - - - - - | 160 |
| CAPITULO 7 | <u>FLIP-FLOPS</u> | |
| | 7.0 Introducción - - - - - | 163 |
| | 7.1 Diagramas de Tiempo - - - - - | 164 |
| | 7.2 Flip-Flops Implementados con Compuertas - - - - - | 165 |
| | 7.3 Flip-Flops con Reloj - - - - - | 171 |
| | 7.4 Flip-Flops RS con Reloj- - - - - | 172 |
| | 7.5 FF JK - - - - - | 173 |
| | 7.6 Flip-Flop JK Disparado en el Filo de una Transición de Reloj - - - - - | 174 |

| | |
|--|-----|
| 7.7 Flip-Flop JK Maestro Esclavo - - - - - | 176 |
| 7.8 Características de los Flip-Flops - - - - - | 177 |
| 7.9 Otras Entradas de Control a un Flip-Flop - - - - - | 183 |
| 7.10 Registros de Corrimiento - - - - - | 185 |
| 7.11 Multivibrador - - - - - | 190 |
| Problemas Propuestos - - - - - | 195 |

CAPITULO 8 DISEÑO SECUENCIAL

| | |
|--|-----|
| 8.0 Introducción - - - - - | 197 |
| 8.1 Tablas de Estados - - - - - | 198 |
| 8.2 Diagramas de Transición - - - - - | 199 |
| 8.3 Metodología de Diseño Secuencial - - - - - | 201 |
| 8.4 Sistemas Secuenciales de más de 2 Estados - - - - - | 208 |
| 8.5 Sistemas Secuenciales que no están Completamente Especificados - - - - - | 212 |
| 8.6 Reducción de Estados - - - - - | 216 |
| 8.7 Contadores - - - - - | 218 |
| 8.8 Reset - - - - - | 220 |
| 8.9 Contadores Auto Restablecientes - - - - - | 221 |
| 8.10 Contadores Ascendente/Descendente - - - - - | 226 |
| 8.11 Ejemplos de Diseño Secuencial - - - - - | 228 |
| 8.12 Contadores a partir de Registros de Corrimiento - - - - - | 243 |
| 8.13 Contadores Asíncronos - - - - - | 245 |
| Problemas Propuestos - - - - - | 246 |

APENDICE A SEMICONDUCTORES

| | |
|---|-----|
| A.0 Introducción - - - - - | 247 |
| A.1 Elementos Semiconductores - - - - - | 247 |

APENDICE B FAMILIAS LOGICAS

| | |
|--|-----|
| B.0 Terminología de los Circuitos Integrados - - - - - | 253 |
| B.1 Familia TTL - - - - - | 262 |
| B.1.0 Series de la Familia TTL - - - - - | 262 |
| B.1.1 Operación Básica - - - - - | 266 |
| B.1.2 Características de la Familia Lógica TTL - - - - - | 270 |

| | |
|---|---------|
| B.1.3 Otros Dispositivo TTL - - - - - | 273 |
| B.1.4 Niveles de Voltaje - - - - - | 279 |
| B.1.5 Lógica Positiva y Negativa - - - - - | 280 |
| B.2 Familia Lógica MOS - - - - - | 284 |
| B.2.1 Características de la Familia Lógica MOS - - | 284 |
| B.2.2 Configuración de Bloques Lógicos - - - - - | 286 |
| B.3 Familia Lógica C-MOS - - - - - | 289 |
| B.3.1 Circuitos Básicos - - - - - | 289 |
| B.3.2 Características de la Serie C-MOS 4000 A - - | 292 |
| B.4 Familia Lógica ECL - - - - - | 295 |
| B.4.1 Características de la Familia Lógica ECL - - | 295 |
| B.4.2 Circuito Básico ECL - - - - - | 296 |
| B.4.3 Compuertas Lógicas ECL - - - - - | 296 |
| B.5 Familia Lógica RTL - - - - - | 298 |
| B.6 Familia Lógica DTL - - - - - | 302 |
| B.7 Familia Lógica HTL - - - - - | 304 |
| B.7.1 Características de la Familia Lógica HTL- - - | 305 |
| B.8 Comparación de las Familias Lógicas - - - - - | 305 |
| BIBLIOGRAFIA - - - - - | 304 |
| INDICE ALFABETICO - - - - - | 309 |

Los Sistemas Digitales se Originaron en un Mundo Analogico

1.0 CONCEPTOS BASICOS

Para establecer una idea clara respecto a la definición de sistemas digitales y analógicos dirigamos nuestra atención hacia el mundo físico en que se originan.

Al referirnos a parámetros físicos como, temperatura, velocidad, -- aceleración, etc. nos topamos frecuentemente con la necesidad de medir-- los, procesar la información medida e incluso controlar tal parámetro.

La medición, manipulación y control de las variables físicas se ha-- bía efectuado tradicionalmente por medio de dispositivos que tienen un -- comportamiento análogo a la variable.

Por este motivo a los parámetros antes mencionados y a sus instru-- mentos de medición y control se les da el nombre de ANALOGICOS. De he-- cho nuestro medio es un mundo cuyas variables físicas son en su mayoría-- analógicas.

Así por ejemplo, en un termómetro, la columna de mercurio que se en-- cuentra dentro de el, aumenta o disminuye dependiendo del aumento o dis-- minución de la temperatura del medio que lo rodea. Fig. 1.1.

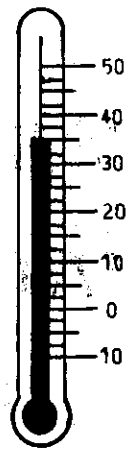
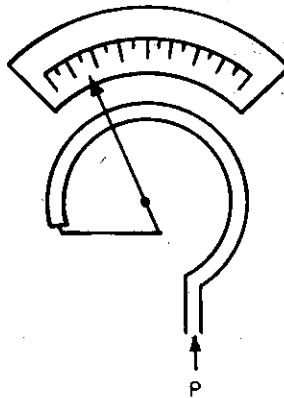


Fig. 1.1 TERMOMETRO DE MERCURIO

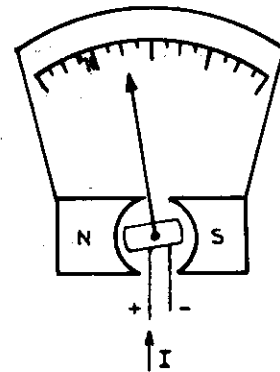
Algo semejante sucede con un dinamómetro, con un manómetro o con un galvanómetro, Fig. 1.2. En cada uno de los casos, la fuerza, presión o corriente eléctrica puede medirse mediante la deflexión de una aguja indicadora sobre la superficie graduada en las unidades correspondientes a cada parámetro.



a)



b)



c)

Fig. 1.2 DISPOSITIVOS DE MEDICION. a) Dinamómetro, b) Manómetro, c) Galvanómetro.

1.1 CONCEPTOS DE RESOLUCION Y EXACTITUD

Establezcamos la definición de dos conceptos importantes, el primero de ellos es la RESOLUCION de un sistema de medición, este término se refiere a la mínima separación de dos valores numéricos sucesivos que pueden resultar del proceso de medición. A esta mínima separación se le llama UNIDAD DE RESOLUCION y limita la exactitud del sistema. Cuando un valor cae entre dos valores numéricos sucesivos de resolución mínima, se le tendrá que dar un valor numérico mayor o menor a su valor real. Por ejemplo si dos personas encuentran una moneda de 5 centavos y se la quieren repartir, a uno de ellos le tocarán 3 centavos y al otro 2 centavos puesto que la UNIDAD MINIMA DE RESOLUCION en nuestro sistema monetario es el centavo. En este caso no es posible una división EXACTA y el error en ambas cantidades es un medio de la unidad de resolución. El término EXACTITUD está relacionado con la calidad del proceso de medición. El incremento de la exactitud usualmente requiere el perfeccionamiento de la técnica o dispositivo de medición. Por ejemplo, de una regla no obtendremos el mismo grado de exactitud que al usar un micrometro. En el ejemplo del termómetro que mencionábamos en el punto anterior pueden apreciarse claramente los conceptos de exactitud y resolución. Fig. 1.3.

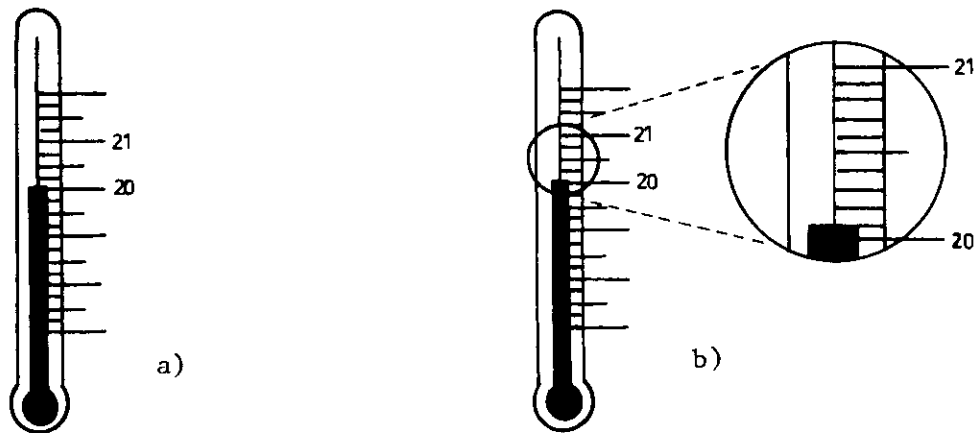


Fig. 1.3 AL DISMINUIR EL INTERVALO ENTRE DOS VALORES NUMERICOS SU-
CESIVOS EN UNA MEDICION SE AUMENTA LA EXACTITUD.

A un observador que se le pregunte la temperatura en el dibujo de la fig. 1.3a, seguramente responderá 20°C. Al ampliar la sección del termómetro entre 20° y 21° e imaginariamente aumentar la escala (fig. 1.3b)- podemos apreciar que un valor más aproximado a la temperatura real será - 20.1°C.

La exactitud de una medición puede incrementarse reduciendo el intervalo entre dos valores numéricos sucesivos. Este incremento de resolución lógicamente aumenta el valor numérico de la medición, para el ejemplo de 20 a 20.1 o sea de 2 a 3 dígitos.

1.2 QUE ES UN SISTEMA DIGITAL?

En la manipulación de una parámetro medido, en su proceso e incluso en la conversación cotidiana es difícil emplear el valor numérico exacto de una variable, y en lugar de el se usa un valor numérico aproximado que es representativo de su valor real. La temperatura en el ejemplo del termómetro leída por un observador, era de 20°C mientras que en realidad es un valor entre 20° y 21°C.

En la adquisición de un dato y en el proceso de medición, intervienen los conceptos de exactitud, resolución y el tiempo en el cual se determina el valor numérico de la variable medida. Comúnmente a este proceso de adquisición se le conoce como "digitalización" de una variable. Este término indica el hecho de que una variable original se reemplaza por un valor numérico cuyos dígitos representan la magnitud de la variable en un tiempo dado. Por ejemplo una vez convertida la altura de la columna de mercurio de un termómetro a un valor digital, la cantidad puede procesarse, almacenarse, controlarse, etc.

Entonces un sistema digital se puede definir como un sistema que procesa información en forma digital (numérica) en vez de procesar a la misma variable en forma analógica.

1.3 SISTEMAS CONTINUOS Y NO CONTINUOS

Para definir estos sistemas comparemos el funcionamiento de un termómetro de mercurio y uno digital. En el primero, cualquier cambio en la temperatura corresponderá a un cambio en la altura de la columna de mercurio. El termómetro digital convertirá periódicamente la temperatura a un valor numérico y lo mostrará en una pantalla. Un cambio en la temperatura no se indicará hasta que sea lo suficientemente grande para cambiar al dígito próximo mayor o menor. Si no sucede esto el valor indicado permanecerá igual.

Por este motivo a un sistema analógico se le asocia con el término "continuo" y a un sistema digital con el término "no continuo".

1.4 REPRESENTACION DE INFORMACION Y CANTIDAD

En la fig. 1.4 se muestran dos formas para detectar e indicar la velocidad de un motor. El primero es un sistema analógico y el segundo es un sistema digital.

En el sistema analógico aparece conectado a la flecha del motor un tacómetro generador, que produce un voltaje proporcional a la velocidad del motor. Este voltaje pasa a un voltmetro, en cuya carátula la graduación está marcada en R.P.M. (Revoluciones por Minuto). En este caso el dato VELOCIDAD, está representado por un voltaje continuo que puede tener un rango de 0 a 10 voltios, manifestado en forma también continua -- por la aguja del voltmetro Fig. 1.4a.

En el sistema digital la flecha del motor tiene una marca reflejante que es detectada por medio de una fotocelda. Cada pulso generado por la fotocelda al pasar la marca equivale a una revolución. Un contador digital cuenta la cantidad de pulsos que por unidad de tiempo en este caso minutos, será igual a las R.P.M. Fig. 1.4b.

En este sistema el dato VELOCIDAD no está representado por un voltaje continuo, sino por pulsos, es decir un voltaje discreto, un nivel alto y un nivel bajo que corresponden a los voltajes típicos de 0 volts y 5 volts de corriente directa.

En ambos casos la INFORMACION se representa por medio de un voltaje. La cantidad de voltaje en el sistema analógico es proporcional a la velocidad. En el sistema digital la velocidad es proporcional a la cantidad de pulsos.

La representación de CANTIDAD puede efectuarse por medio de voltajes, ya sea en forma analógica o en forma digital.

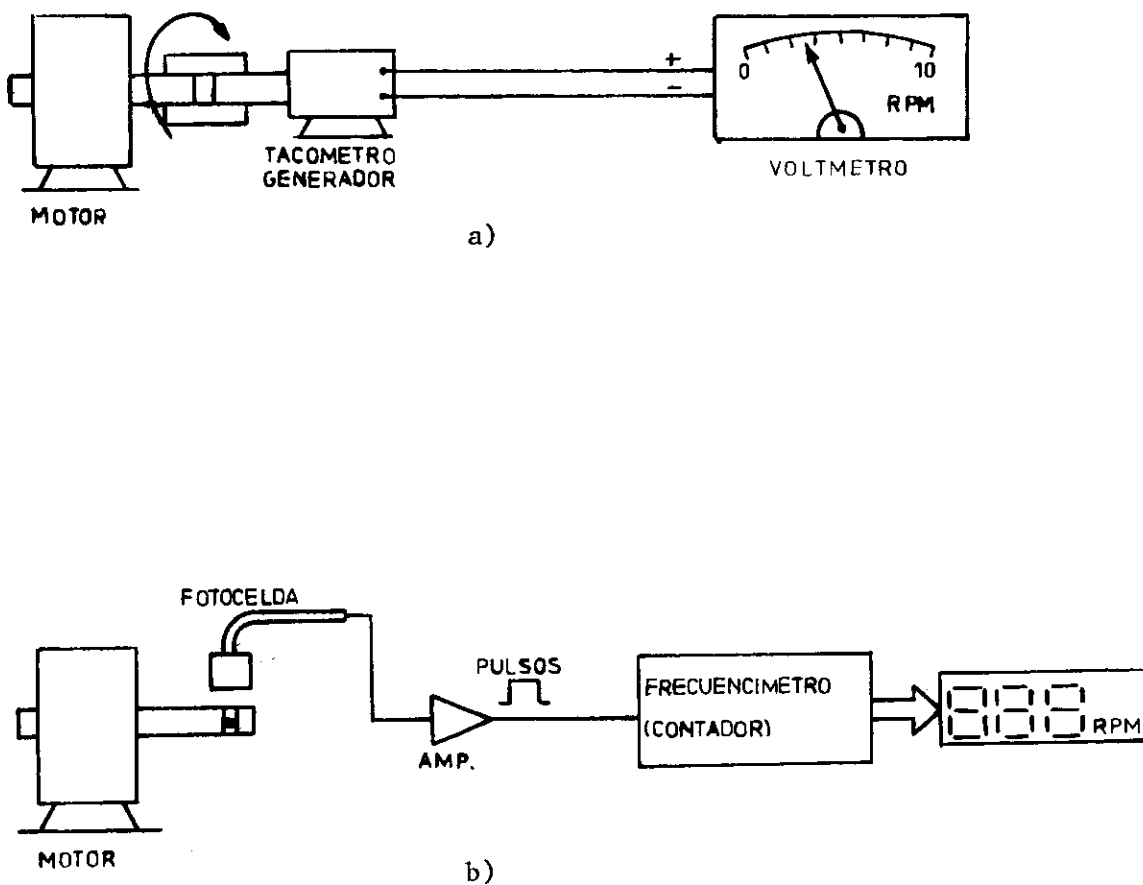


Fig. 1.4 DETECCIÓN Y LECTURA DE VELOCIDAD EN LA FLECHA DE UN MOTOR.
a) Sistema Analógico, Tacómetro-Voltmetro, b) Sistema Digital, Fotocelda-Contador de Pulsos por Unidad de Tiempo.

En la fig. 1.5 se muestra un circuito formado por una fuente, un potenciómetro lineal con una escala de 0 a 9 y un foco.

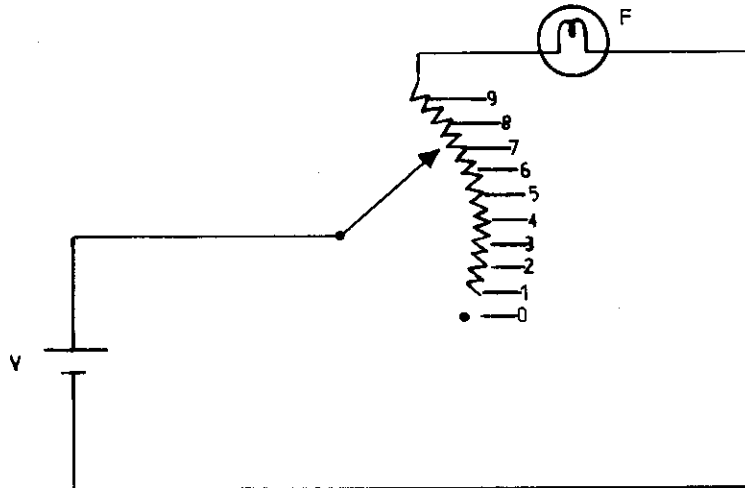


Fig. 1.5 CIRCUITO PARA ILUSTRAR LA REPRESENTACION DE CANTIDAD EN FORMA ANALOGICA.

De acuerdo a la posición en que se encuentre el potenciómetro, existirá una intensidad luminosa proporcional al valor de la resistencia, -- desde "0" (circuito abierto) hasta la máxima posible (circuito cerrado).

Imaginemos que un observador trate de distinguir entre los 10 niveles, con toda seguridad le será difícil apreciar el nivel 4 del 5 o el 5 del 6, sin embargo es simple detectar el foco completamente apagado (posición "0") o completamente encendido (posición "9").

Para un observador humano es difícil detectar niveles analógicos. -- Lo es también para un circuito electrónico, en el cual se elevaría considerablemente el costo y bajaría su confiabilidad.

Por este motivo los circuitos digitales electrónicos trabajan solamente con dos niveles de voltaje.

Un nivel bajo llamado "0" cero lógico y un nivel alto llamado "1" -- uno lógico. Como se muestra en la fig. 1.6

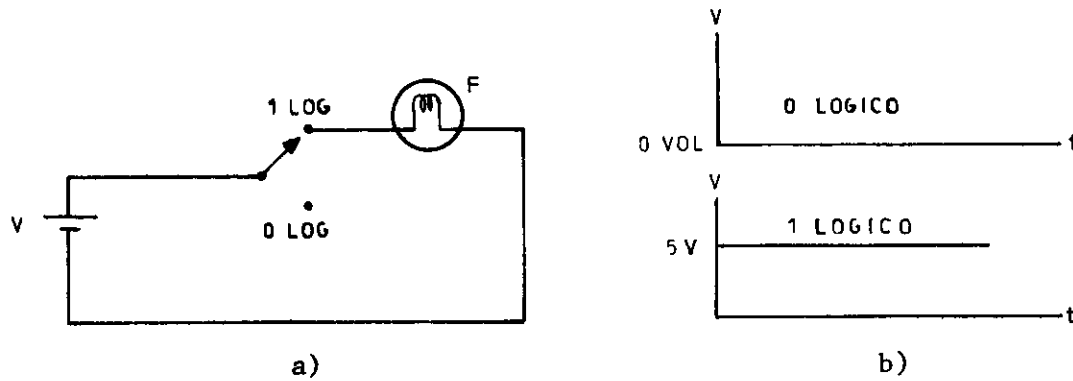


Fig. 1.6 LOS SISTEMAS DIGITALES EMPLEAN SOLAMENTE DOS NIVELES DE VOLTAJE PARA REPRESENTAR CANTIDAD.
 1.6a CIRCUITO SIMPLIFICADO, EL POTENCIOMETRO SE CAMBIO POR UN INTERRUPTOR, Fig. 1.6b NIVELES DE VOLTAJE PARA UN 0 Y UN 1 LOGICOS.

PROBLEMAS PROPUESTOS

- 1.-¿Que diferencia existe entre el funcionamiento de un sistema digital y un sistema analógico?
- 2.-¿En que consiste la conversión analógica digital? Y por que es necesaria?
- 3.- En la fig. 1.3 b) aparecen un termómetro y una sección amplificada del mismo termómetro.
 - a)¿Cual es la unidad de resolución en ambos casos?
 - b)¿Cual graduación puede ofrecer una lectura más precisa ?
- 4.-¿Cual es el concepto de continuidad (o de variable continua)?
- 5.-¿Como se representa la información en un sistema digital y en un sistema analógico?
- 6.-¿Como se representa la cantidad en un sistema digital?

2 *Sistemas Numericos*

2.0 INTRODUCCION

Desde la más remota antigüedad el hombre tuvo la necesidad de contar, fué entonces cuando los números tomaron una gran importancia, aquellos símbolos que representaban cantidades evolucionaron de tal forma -- que estructuraron sistemas numéricos, como es el caso de los numeros romanos, los griegos y los egipcios.

Como seguramente hemos tenido alguna experiencia con el sistema numérico romano lo tomaremos para ilustrar el tipo de notación numérica que empleaba, en la fig. 2.1 aparecen algunos de sus símbolos.

| | |
|--------|----------|
| I — 1 | C — 100 |
| V — 5 | D — 500 |
| X — 10 | M — 1000 |
| L — 50 | |

Fig. 2.1 SIMBOLOS DEL SISTEMA NUMERICO ROMANO Y SU EQUIVALENTE EN DECIMAL.

Existían ciertas reglas, por ejemplo, cuando un I (uno) aparecía antes de un V (cinco), "IV", el símbolo menor era restado al mayor, así el número $IV = (5-1) = 4$. Por el contrario cuando el signo menor aparece delante del mayor se suman, el número "VI" = $(5+1) = 6$. Nótese que en ambos números los símbolos I y V conservan su valor independientemente de la posición en el número, un V (cinco) nunca podrá ser un 50 o un 500.

2.1 SISTEMAS NUMERICOS DE NOTACION POSICIONAL

Con una antigüedad aproximada de 2000 años y originario de la India nuestro actual sistema numérico, el "decimal" fué introducido a Europa por los Arabes, de allí el nombre de números arábigos. A cada uno de sus símbolos del 0 al 9 se les conoce como "dígito" raíz latina que significa dedo. Supuestamente, se usan 10 dígitos porque el hombre posee 10 dedos, que empleaba como herramientas para contar.

El sistema decimal tiene dos características importantes. Una es el concepto del "cero" que indica ausencia de cantidad o valor y la otra es la notación posicional, para explicarla usaremos el siguiente ejemplo.

Imaginemos un conteo en decimal que inicia por supuesto en cero, al llegar a 9 alcanzaremos el dígito de mayor valor, si incrementamos nuestro conteo generaremos un acarreo, como se indica en la fig. 2.2.

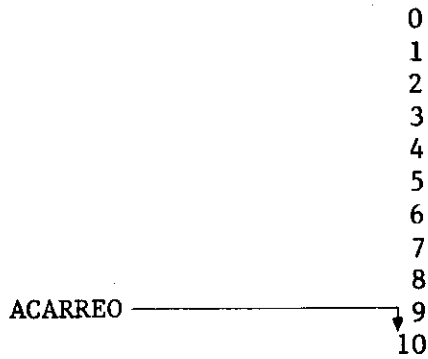


Fig. 2.2 GENERACION DEL ACARREO PARA UN CONTEO EN DECIMAL

Este acarreo forma el número 10 diez, se dice que el "0" cero ocupa la posición de las unidades mientras que el "1" uno ocupa la posición -- de las decenas. Este proceso continuará cada vez que se alcanza el 9 en la posición de las unidades. Cuando aparece un 9 en la posición de las decenas se genera un acarreo a la posición de las centenas y así sucesivamente.

Nótese que un 1 en la posición de las decenas tiene un valor o "peso" 10 veces mayor que un uno de la posición de las unidades. Lo mismo sucede con un 1 de la posición de las centenas, es 10 veces mayor que un 1 de la posición de las decenas.

Definiremos entonces "peso" de un dígito, como el valor que toma -- (ese dígito) según la posición que tenga en el número.

De aquí que el nombre "Sistema numérico de notación posicional" se aplica a los sistemas numéricos donde los dígitos que forman un número -- tienen diferentes pesos de acuerdo a su posición (en el número).

La base del sistema numérico decimal es 10. La base es igual al número de símbolos que posee un sistema numérico. El dígito mayor siempre es una unidad menor que la base. Cada posición multiplica el valor del dígito por la base elevada a esa posición. Además un acarreo de una posición a la próxima mayor, incrementa su peso por base veces. Esto es -- válido para un sistema de notación posicional de cualquier base.

En la fig. 2.3 se muestran los sistemas numéricos de notación posicional más comunes.

| R-BASE | SIST. NUMERICO | R DIGITOS EMPLEADOS |
|--------|----------------|--|
| 2 | BINARIO | 0, 1 |
| 8 | OCTAL | 0, 1, 2, 3, 4, 5, 6, 7. |
| 10 | DECIMAL | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. |
| 16 | HEXADECIMAL | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F |

Fig. 2.3 SISTEMAS NUMERICOS MAS COMUNES

Los sistemas de notación posicional han sido tan ampliamente aceptados que raramente los analizamos. Tomemos un número decimal, por ejemplo el 258. La posición de los dígitos en el número se indica en la fig. 2.4, nótese que la posición inicia en 0.

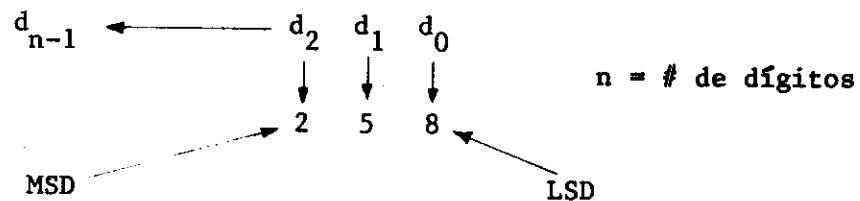


Fig. 2.4 POSICION DE LOS DIGITOS DEL NUMERO 258

El número está formado por tres dígitos 2, 5 y 8. El dígito de menor peso es el 8 y se le conoce como (LSD) siglas en inglés de Least Significant Digit, el dígito de mayor peso es el 2 y se le conoce como (MSD) Most Significant Digit.

El 8 ocupa la posición de las unidades y pesa $8 \times 1 = 8$ unidades. El 5 ocupa la posición de las decenas y pesa $5 \times 10 = 50$ unidades. El 2 ocupa la posición de las centenas y pesa $2 \times 100 = 200$ unidades.

$$(2 \times 100) + (5 \times 10) + (8 \times 1) = 258$$

$$2 (10^2) + 5 (10^1) + 8 (10^0) = 258$$

Entonces un número decimal de N dígitos puede tomarse como una suma toria de sus coeficientes multiplicados por la base elevada a la posi--- ción en que se encuentran.

$$N_{10} = a_{n-1} (10)^{n-1} + a_{n-2} (10)^{n-2} + \dots + a_1 (10)^1 + a_0 (10)^0$$

$$N_{10} = \sum_{i=0}^{n-1} a_i (10)^i$$

Donde:

a = coeficiente

n = cantidad de coeficiente

N = número

A esta ecuación se le conoce como "expresión sumatoria". La nota-- ción posicional de un número es una expresión sumatoria abreviada donde-- se omiten los signos de suma y los pesos de cada posición.

$$N_{10} = a_{n-1} a_{n-2} \dots a_1 a_0$$

$$N_{10} = 258$$

La expresión sumatoria puede generalizarse para cualquier sistema numéri-- co.

$$N_r = \sum_{i=0}^{n-1} a_i (r)^i$$

Al considerar números fraccionarios tenemos.

$$N_r = a_{n-1} (r)^{n-1} + \dots + a_0 (r)^0 + a_{-1} (r)^{-1} + \dots + a_{-m} (r)^{-m}$$

$$N_r = \sum_{i=-m}^{n-1} a_i (r)^i$$

Donde:

r = base del sistema

m = número de dígitos fraccionarios

Ejemplo 2.0

Expresa el número 258.25 de acuerdo a la expresión sumatoria.

$$\begin{aligned} N_{10} &= 258.25 \\ &= 2(10)^2 + 5(10)^1 + 8(10)^0 + 2(10)^{-1} + 5(10)^{-2} \\ &= 200 + 50 + 8 + .2 + .05 \end{aligned}$$

$$N_{10} = 258.25$$

Cuando se trabaje con sistemas numéricos de diferentes bases debe indicarse por medio de un subíndice la base en que se encuentra un número.

Ejemplo 2.1

$$\begin{aligned} 258_{10} \\ 1010_2 \\ 357_8 \\ A32_{16} \end{aligned}$$

2.2 SISTEMA NUMERICO BINARIO

La base del sistema numérico binario es 2, por lo tanto se usan solamente dos dígitos "0" y "1" para la representación de cualquier número o cantidad. Un número mayor que "1" puede representarse empleando el mismo método que en decimal (un número mayor que 9 genera un acarreo que indica una decena). Entonces la representación binaria de 2_{10} es 10_2 , - (uno cero en base 2).

| DECIMAL | BINARIO | OCTAL | HEXADECIMAL |
|---------|---------|-------|-------------|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 10 | 2 | 2 |
| 3 | 11 | 3 | 3 |
| 4 | 100 | 4 | 4 |
| 5 | 101 | 5 | 5 |
| 6 | 110 | 6 | 6 |
| 7 | 111 | 7 | 7 |
| 8 | 1000 | 10 | 8 |
| 9 | 1001 | 11 | 9 |
| 10 | 1010 | 12 | A |
| 11 | 1011 | 13 | B |
| 12 | 1100 | 14 | C |
| 13 | 1101 | 15 | D |
| 14 | 1110 | 16 | E |
| 15 | 1111 | 17 | F |
| 16 | 10000 | 20 | 10 |
| 17 | 10001 | 21 | 11 |
| 18 | 10010 | 22 | 12 |
| 19 | 10011 | 23 | 13 |
| 20 | 10100 | 24 | 14 |

Fig. 2.5 NUMEROS DEL 0 AL 20 EN BASE 10 CON SU EQUIVALENTE EN BINARIO, OCTAL Y HEXADECIMAL.

A los dígitos binarios se les da el nombre de BIT, que es una contracción de los términos en inglés BINARY-DIGIT. Al dígito de mayor peso en un número se le llama MSB del inglés (MOST SIGNIFICANT BIT) y al dígito de menor peso se le llama LSB (LEAST SIGNIFICANT BIT).

2.2.1 CONVERSION DE BINARIO A DECIMAL

La conversión de binario a decimal se efectúa por medio de la expresión sumatoria.

$$N_{10} = \sum_{i=0}^{n-1} a_i (2)^i$$

Ejemplo 2.2

Convierta a base 10 el número binario 111001_2 .

$$111001_2 \longrightarrow N_{10}$$

$$\begin{aligned} N_{10} &= 1(2)^5 + 1(2)^4 + 1(2)^3 + 0(2)^2 + 0(2)^1 + 1(2)^0 \\ &= 32 + 16 + 8 + 0 + 0 + 1 \end{aligned}$$

$$N_{10} = 57_{10}$$

Ejemplo 2.3

Convierta a base 10 el número binario 1101.11

$$1101.11_2 \longrightarrow N_{10}$$

$$\begin{aligned} N_{10} &= 1(2)^3 + 1(2)^2 + 0(2)^1 + 1(2)^0 + 1(2)^{-1} + 1(2)^{-2} \\ &= 8 + 4 + 0 + 1 + 0.5 + 0.25 \end{aligned}$$

$$N_{10} = 13.75_{10}$$

2.2.2 CONVERSION DE DECIMAL A BINARIO

El uso de la expresión sumatoria para convertir un número en base 10 a base 2 no es muy útil puesto que es difícil pensar en otro sistema numérico que no sea decimal.

Ejemplo 2.4

Convertir el número 23_{10} a binario.

$$23_{10} \longrightarrow N_2$$

$$N_2 = 2 (10)^1 + 3 (10)^0$$

$$= 20 + 3$$

$$N_2 \neq 23$$

Aparentemente cometimos un error, sin embargo el problema fué, que es necesario pensar en binario. Nótese que el 2, 3 y 10 están escritos en decimal y no en binario.

Intentemos de nuevo.

$$N_2 = \sum_{i=0}^{n-1} a_i (1010)^i$$

$$= 10 (1010)^1 + 11 (1010)^0$$

$$= 10100 + 11$$

$$N_2 = 10111_2$$

Para comprobar usemos la expresión sumatoria:

$$10111_2 \longrightarrow N_{10}$$

$$N_{10} = 1 (2)^4 + 0 (2)^3 + 1 (2)^2 + 1 (2)^1 + 1 (2)^0$$

$$= 16 + 0 + 4 + 2 + 1$$

$$N_{10} = 23_{10}$$

Existen dos métodos más cómodos para la conversión de decimal a binario. Se les conoce como METODO DE EXTRACCION DE POTENCIAS Y METODO DE LOS RESIDUOS. EL METODO DE EXTRACCION DE POTENCIAS Y EL METODO DE LOS RESIDUOS. El método de extracción de potencias consiste en restar la máxima potencia de 2 que pueda contener el # decimal, repitiendo esta operación con el resultado hasta agotar el # 10. El método es útil solo para números pequeños.

Ejemplo 2.5

$$22_{10} \longrightarrow N_2$$

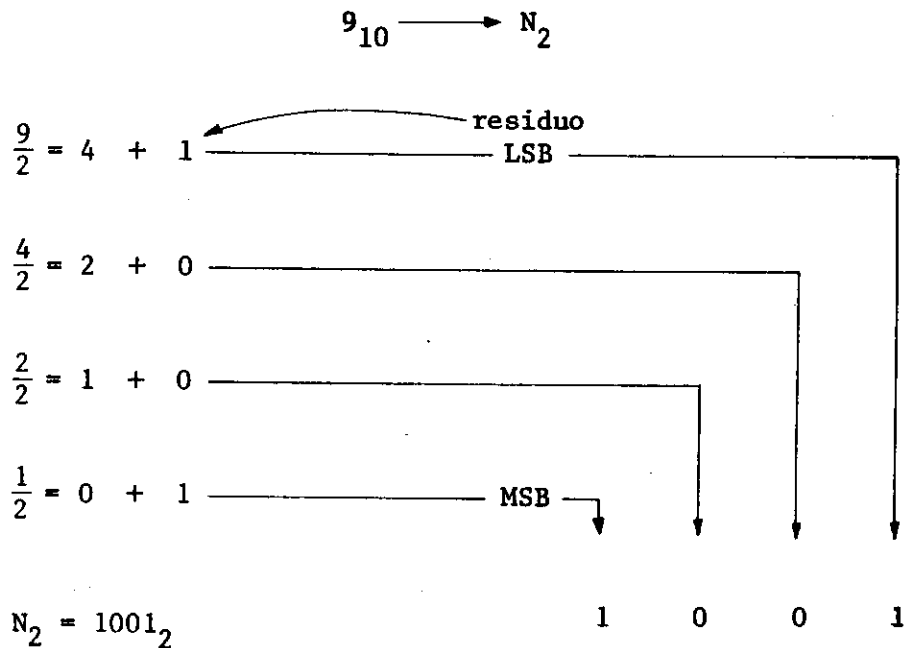
| | |
|-------|-------|
| 22 | |
| - 16 | 2^4 |
| ----- | |
| 6 | |
| - 4 | 2^2 |
| ----- | |
| 2 | |
| - 2 | 2^1 |
| ----- | |
| 0 | |

| | | | | | |
|---------|----|----|----|----|-------|
| | d4 | d3 | d2 | d1 | d0 |
| $N_2 =$ | 1 | 0 | 1 | 1 | 0_2 |

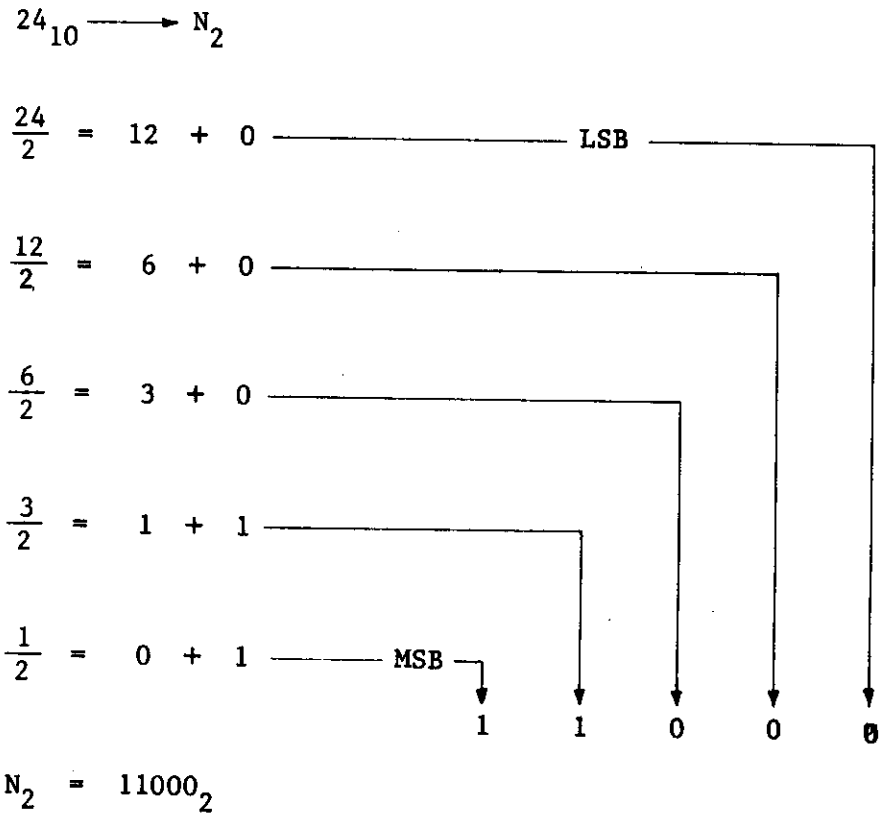
Los coeficientes del número binario son un "1" en la posición de la potencia restada y "0" para la posición no restada.

EL METODO DE LOS RESIDUOS consiste en dividir repetidamente el número decimal entre la base del sistema al que deseamos transformar, e ir registrando sucesivamente los residuos. Estos residuos leídos en orden inverso nos dan el equivalente del número.

Ejemplo 2.6



Ejemplo 2.7



2.3 SISTEMA NUMERICO OCTAL

En este sistema la base es 8, por lo tanto existen solo 8 símbolos que van del 0 al 7. Un número mayor que 7 no puede escribirse (8) puesto que este símbolo no existe en el sistema, la expresión correcta es 10_8 que se lee uno cero base 8. Nótese que un acarreo de un "1" tiene un peso de 8 unidades.

2.3.1 CONVERSION DE OCTAL A DECIMAL

METODO - EXPRESION SUMATORIA

Ejemplo 2.8

$$147_8 \longrightarrow N_{10}$$

$$N_{10} = 1(8)^2 + 4(8)^1 + 7(8)^0$$

$$= 64 + 32 + 7$$

$$N_{10} = 103_{10}$$

2.3.2 CONVERSION DE DECIMAL A OCTAL

METODO DE LOS RESIDUOS

Ejemplo 2.9

$$\begin{array}{l} 103_{10} \longrightarrow N_8 \\ \frac{103}{8} = 12 + 7 \text{ --- LSD} \\ \frac{12}{8} = 1 + 4 \\ \frac{1}{8} = 0 + 1 \text{ --- MSD} \end{array}$$

$N_8 = 147_8$

2.4 SISTEMA NUMERICO HEXADECIMAL

El sistema numérico hexadecimal es un sistema numérico importante - usado en computadoras. Su base es 16 y sus símbolos van del 0 al 9 y de A a F. Como se muestra en la fig. 2.5. Un acarreo de un "1" tiene un peso de 16 unidades. Por lo tanto un 10_{16} (uno cero base 16) no equivale a diez (en decimal).

2.4.1 CONVERSION DE HEXADECIMAL A DECIMAL

METODO: EXPRESION SUMATORIA

Ejemplo 2.10

$$\begin{array}{l} 1A9_{16} \longrightarrow N_{10} \\ N_{10} = 1(16)^2 + 10(16)^1 + 9(16)^0 \\ = 256 + 160 + 9 \\ N_{10} = 425_{10} \end{array}$$

Nótese que la letra "A" se cambia por su equivalente numérico.

2.4.2 CONVERSION DE DECIMAL A HEXADECIMAL

METODO DE LOS RESIDUOS

Ejemplo 2.11

$$1324_{10} \longrightarrow N_{16}$$
$$\frac{1324}{16} = 82 + 12 \quad \text{--- LSB ---}$$
$$\frac{82}{16} = 5 + 2$$
$$\frac{5}{16} = 0 + 5 \quad \text{--- MSB ---}$$

5 2 12

$$N_{16} = 52C_{16}$$

Ejemplo 2.12

$$432_{10} \longrightarrow N_{16}$$
$$\frac{432}{16} = 27 + 0 \quad \text{--- LSB ---}$$
$$\frac{27}{16} = 1 + 11$$
$$\frac{1}{16} = 0 + 1 \quad \text{--- MSB ---}$$

1 11 0

$$N_{16} = 1B0_{16}$$

2.5 CONVERSION BINARIO ↔ OCTAL

El sistema octal puede ser un método conveniente para reducir la longitud de un número binario, esto es muy útil cuando se tienen listados en binario por ejemplo, el contenido de la memoria de una computadora digital. En la fig. 2.6 aparecen los 8 símbolos en octal con su correspondiente en binario. Nótese que para expresar un dígito octal, solo son necesarios 3 bits, esta relación surge de que la base octal 8 es igual a 2^3 .

| OCTAL | BINARIO |
|-------|---------|
| 0 | 000 |
| 1 | 001 |
| 2 | 010 |
| 3 | 011 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

Fig. 2.6 PARA CADA DIGITO OCTAL CORRESPONDEN 3 BIT'S.

La conversión de binario a octal se obtiene dividiendo el número binario en grupos de 3 bits a partir del punto decimal, tanto para la parte entera como la parte fraccionaria.

Ejemplo 2.13 Convierta

$$\underline{11001110.1011}_2 \longrightarrow N_8$$

| | | | | | |
|-----|-----|-----|---|-----|-----|
| 011 | 001 | 110 | . | 101 | 100 |
| ↓ | ↓ | ↓ | | ↓ | ↓ |
| 3 | 1 | 6 | . | 5 | 4 |

$$N_8 = 316.54_8$$

Usando el mismo método podemos convertir un número en base 8 a base 2.

Ejemplo 2.14 Convierta:

$$\begin{array}{ccc}
 345_8 & \longrightarrow & N_2 \\
 \\
 \begin{array}{ccc}
 3 & 4 & 5 \\
 \downarrow & \downarrow & \downarrow \\
 11 & 100 & 101
 \end{array} & & \\
 \\
 N_2 & = & 11100101_2
 \end{array}$$

2.6 CONVERSION BINARIO ↔ HEXADECIMAL

La representación de un número binario en hexadecimal es una mejor alternativa a la representación en octal. La relación parte de que la base hexadecimal 16 es igual a 2^4 . En la figura 2.7 se muestran los dígitos hexadecimales y su correspondiente en binario.

| HEXADECIMAL | BINARIO |
|-------------|---------|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| A | 1010 |
| B | 1011 |
| C | 1100 |
| D | 1101 |
| E | 1110 |
| F | 1111 |

Fig. 2.7 PARA CADA DIGITO HEXADECIMAL CORRESPONDEN 4 BIT'S.

La conversión de binario a hexadecimal se obtiene empleando el mismo método que en octal, solo que aquí se toman 4 bit's por cada dígito - base 16.

Ejemplo 2.15 Convierta:

$$1011110111_2 \longrightarrow N_{16}$$

| | | |
|----|------|------|
| 10 | 1111 | 0111 |
| ↓ | ↓ | ↓ |
| 2 | F | 7 |

$$N_{16} = 2F7_{16}$$

Ejemplo 2.16 Convierta:

$$A51_{16} \longrightarrow N_2$$

| | | |
|------|------|------|
| A | 5 | 1 |
| ↓ | ↓ | ↓ |
| 1010 | 0101 | 0001 |

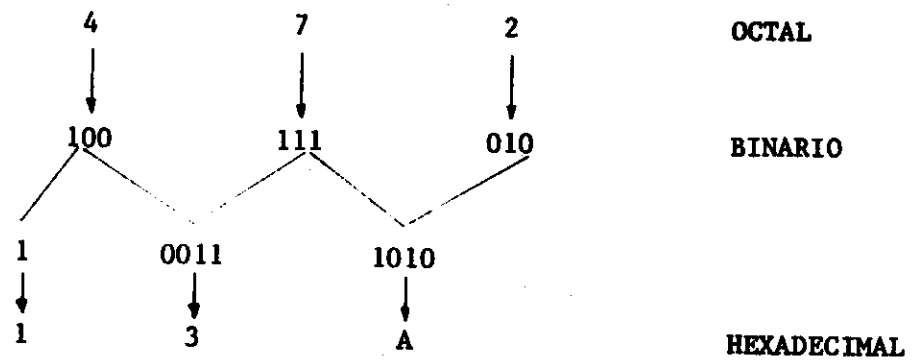
$$N_2 = 101001010001_2$$

2.7 CONVERSION OCTAL ↔ HEXADECIMAL

Un número en base 8 puede convertirse a un número base 16 y viceversa pasando por binario.

Ejemplo 2.17 Convierta:

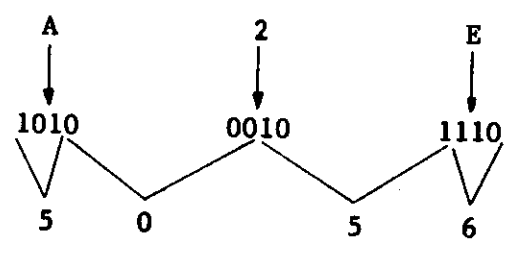
$$472_8 \longrightarrow N_{16}$$



$$472_8 = 13A_{16}$$

Ejemplo 2.18 Convierta:

$$A2E_{16} \longrightarrow N_8$$



$$A2E_{16} = 5056_8$$

2.8 ARITMETICA BINARIA, OCTAL Y HEXADECIMAL

El método para efectuar operaciones aritméticas es básicamente el mismo para todos los sistemas numéricos de notación posicional. Revisemos entonces el procedimiento de la suma "base 10" cuyo método seguramente lo efectuamos en forma mecánica.

Ejemplo:

$$\begin{array}{r} 20 \\ + 46 \\ \hline 66 \end{array}$$

Nótese que en el ejemplo 2.19b) se genera un acarreo igual a 2, debido a que la suma de la columna excedió 2 veces la base.

Existen otros dos métodos para sumar en base 2, en ambos casos es necesario pensar en binario.

METODO 1.- Se basa en el hecho de que $1_2 + 1_2$ es igual a cero y se acarrea 1_2 , si existen varios 1's en la columna, cada par de unos sumados genera un acarreo.

Ejemplo 2.20

$$\begin{array}{r}
 \text{ACARREO} \\
 \downarrow \\
 \begin{array}{r}
 1 \\
 + \quad 1_2 \\
 \quad 1_2 \\
 \hline
 10_2
 \end{array}
 \end{array}
 \qquad
 \begin{array}{r}
 1 \quad \text{ACARREOS} \\
 \diagdown \quad \diagup \\
 \begin{array}{r}
 \text{---} 1 \text{---} 1 \text{---} \text{---} \\
 \quad \quad 1 \quad 1_2 \\
 + \quad \quad 1 \quad 1_2 \\
 \quad \quad \quad 1 \quad 1_2 \\
 \hline
 1 \quad 0 \quad 0 \quad 1_2
 \end{array}
 \end{array}$$

METODO 2.-

En el siguiente método existe menos probabilidad de error, consiste en sumar todos los unos de la columna, dar el resultado en binario, escribir el dígito de menor peso en su columna correspondiente y acarrear a las siguientes columnas los dígitos restantes.

Ejemplo 2.21

$$\begin{array}{r}
 1 \\
 1 \\
 11 \\
 10 \\
 10 \\
 11 \\
 \hline
 1011_2 \\
 1001_2 \\
 + \quad 1111_2 \\
 \quad 1111_2 \\
 \quad 1011_2 \\
 \quad 1101_2 \\
 \hline
 1001010_2
 \end{array}
 \qquad
 \begin{array}{r}
 \diagdown \quad \diagup \\
 \text{ACARREOS}
 \end{array}$$

La suma de la primera columna es igual a seis 110_2 se deja el "0" de menor peso y se acarrea el 11, y así sucesivamente.

2.8.2 SUMA OCTAL

Ejemplo 2.22 Sume los números 547_8 y 323_8

| | | | |
|---|----|----|-----------------------|
| 1 | 1 | ← | ACARREO |
| + | 5 | 4 | 7 |
| | 3 | 2 | 3 |
| | 8 | 7 | 10 |
| | -8 | -8 | ← |
| | 1 | 0 | 7 |
| | | | 2 |
| | | | 8 |
| | | ← | SUMA DECIMAL |
| | | ← | RESTA DEL VALOR DEL - |
| | | | ACARREO |

En la posición de las unidades o LSD se encuentran un $7 + 3 = 10$ este número es mayor que 7, al restarle la base, se genera un acarreo a la siguiente columna con un peso de 8 unidades. En la posición de las unidades queda la diferencia entre $10 - 8 = 2$.

2.8.3 SUMA HEXADECIMAL

Ejemplo 2.23 Sume los números $34B_{16}$ y $F2A_{16}$

| | | | |
|---|-----|-----|----|
| 1 | 1 | ← | |
| + | 3 | 4 | B |
| | F | 2 | A |
| | 18 | 7 | 21 |
| | -16 | -16 | ← |
| | 1 | 2 | 7 |
| | | | 5 |
| | | | 16 |

En la columna de menor peso, la literal se cambia por su valor numérico, $B = 11$, $A = 10$. La suma es mayor que 16 es necesario entonces restarle la base. El acarreo generado tendrá un peso de 16 unidades.

Ejemplo 2.24

$$\begin{array}{r} + \quad 3 \quad C \quad 4 \\ \quad 9 \quad 1 \quad 7 \\ \hline (12) \quad (13) \quad (11) \\ \hline C \quad D \quad B_{16} \end{array}$$

En este ejemplo la suma de cada columna es menor que 16 y mayor que 9, por lo tanto es necesario cambiar los números resultantes por su letra equivalente.

2.8.4. RESTA

El procedimiento de la resta en base 10 es el mismo para los sistemas de notación posicional de diferentes bases.

Ejemplo 2.25

Efectúe la siguiente resta decimal $45_{10} - 26_{10}$.

$$\begin{array}{r} 3 \\ \begin{array}{l} \leftarrow \text{PRESTAMO} \\ 4 \quad (10) + 5 \\ - \quad 2 \quad 6 \\ \hline 1 \quad 9 \end{array} \end{array}$$

La resta al igual que la suma, se inicia con el dígito de menor peso, 5 menos 6 no se puede restar entonces pedimos un préstamo al dígito de la siguiente columna (decenas) el 4. La mínima cantidad que nos puede prestar es 1 (una decena), y se le llama "préstamo".

Al sumar el préstamo al 5 tenemos 15 menos 6 es igual a 9. En la siguiente posición $3-2 = 1$.

2.8.5. RESTA BINARIA

El proceso de la resta en base 2 es similar a la resta decimal. En-

este sistema un préstamo de una columna anterior tiene un peso de 2, como se puede observar en el ejemplo:

Ejemplo 2.26

$$\begin{array}{r}
 \\
 \\
 \hline
 1
 \end{array}$$

Frecuentemente no es posible obtener un préstamo de una columna anterior, es necesario en este caso acudir a la próxima columna.

Ejemplo 2.27

$$\begin{array}{r}
 \\
 \\
 \hline
 0
 \end{array}$$

2.8.6 DOS COMPLEMENTO

En una computadora digital, la resta usualmente se desarrolla por medio de sumas. Por consiguiente, no es necesario que la unidad aritmética de la computadora cuente con un circuito que reste, la ventaja de esto es la que la unidad aritmética se reduce. Reducción de los circuitos de la unidad aritmética. El método mas usado para efectuar la resta por medio de sumas es el método del 2 complemento y consiste en sumar el minuendo más el dos complemento del sustraendo. El dos complemento de un número es igual al uno complemento más 1. El uno complemento se encuentra cambiando todos los "unos" por "ceros" y viceversa.

Ejemplo 2.28 .- Reste usando el método del 2 complemento 1101_2 menos 0110_2

$$\begin{array}{r}
 \\
 \\
 \hline
 1
 \end{array}$$

USANDO EL METODO DEL 2 COMPLEMENTO

a) Obtener el uno complemento de 0110_2

$$0110 \longrightarrow 1001 \quad (\text{uno complemento})$$

b) El dos complemento se obtiene sumando 1

$$\begin{array}{r} 1001 \\ + \quad 1 \\ \hline 1010 \end{array} \longleftarrow (\text{dos complemento})$$

c) Sumar el minuendo al dos complemento

$$\begin{array}{r} 1101 \\ + 1010 \\ \hline 10111 \end{array} \longleftarrow \text{resultado}$$

↑
se desprecia

El acarreo que resulta de los bit de mayor peso se desprecia.

2.8.7 RESTA OCTAL

En la resta octal un préstamo de una columna anterior tiene un peso de 8.

Ejemplo 2.29

$$\begin{array}{r} 3 \\ \cancel{4} (8) + 2_8 \\ - 1 \quad 7_8 \\ \hline 2 \quad 3_8 \end{array}$$

2.8.9 MULTIPLICACION Y DIVISION

En los puntos anteriores a este, se puede observar que el mecanismo de la suma y de la resta entre los sistemas numéricos de notación posicional es el mismo. De igual forma el procedimiento para las operaciones de MULTIPLICACION Y DIVISION que se usa en el sistema numérico decimal, funciona en binario, octal y hexadecimal.

Ejemplo 2.33 Efectúe la siguiente multiplicación binaria

$$\begin{array}{r}
 1000_2 \longrightarrow 8_{10} \\
 \times 1000_2 \longrightarrow \times 8_{10} \\
 \hline
 0000 \\
 0000 \\
 0000 \\
 1000 \\
 \hline
 1000000_2 \longrightarrow 64_{10}
 \end{array}$$

Nótese, en el ejemplo 2.33 que el mecanismo de la multiplicación binaria es semejante a la multiplicación decimal, con la variante de que: cualquier dígito del multiplicador que se multiplica con un dígito del multiplicando dará como resultado solamente un CERO o un UNO.

Por otro lado la división binaria consiste en restar al dividendo, el divisor tantas veces como sea posible, como se muestra en el ejemplo 2.34.

Ejemplo 2.34 Efectúe la siguiente división binaria

$$\begin{array}{r}
 10101_2 \left| \begin{array}{l} 11_2 \\ \hline 111_2 \end{array} \right. \longrightarrow \begin{array}{r} 21_{10} \left| \begin{array}{l} 3_{10} \\ \hline 7_{10} \end{array} \right. \\ 0 \end{array} \\
 - \underline{11} \\
 100 \\
 - \underline{11} \\
 11 \\
 - \underline{11} \\
 0
 \end{array}$$

La multiplicación o división $\times 2$ o $\div 2$ de un número binario puede realizarse sin desarrollar operaciones aritméticas, solamente es necesario efectuar un corrimiento a la derecha para multiplicar por 2 y a la izquierda para dividir entre 2.

Ejemplo 2.35 Multiplicación Binaria $\times 2$

$$\begin{array}{lclclcl}
 1 \times 2 = & \overleftarrow{1}_2 & = & 10_2 & = & 2_{10} \\
 2 \times 2 = & \overleftarrow{10}_2 & = & 100_2 & = & 4_{10} \\
 4 \times 2 = & \overleftarrow{100}_2 & = & 1000_2 & = & 8_{10} \\
 8 \times 2 = & \overleftarrow{1000}_2 & = & 10000_2 & = & 16_{10}
 \end{array}$$

Ejemplo 2.36 División Binaria $\div 2$

$$\begin{array}{lclclcl}
 8 \div 2 = & \overrightarrow{1000}_2 & = & 100_2 & = & 4_{10} \\
 4 \div 2 = & \overrightarrow{100}_2 & = & 10_2 & = & 2_{10} \\
 2 \div 2 = & \overrightarrow{10}_2 & = & 1_2 & = & 1_{10}
 \end{array}$$

La MULTIPLICACION EN OCTAL Y HEXADECIMAL debe tomar especial cuidado, puesto que al multiplicar un dígito con otro el resultado puede ser mayor que la base, obviamente este resultado está expresado en decimal. Por consiguiente es necesario efectuar los ajustes pertinentes con el fin de corregir el producto.

Ejemplo 2.37 Efectúe la siguiente multiplicación octal.

$$\begin{array}{r}
 \begin{array}{r}
 1 \leftarrow \\
 25_8 \\
 \times 22_8 \\
 \hline
 10 \\
 - 8 \\
 \hline
 5 \ 2 \\
 \hline
 5 \ 2 \\
 \hline
 5 \ 7 \ 2_8
 \end{array}
 \longrightarrow
 \begin{array}{r}
 2(8)^1 + 5(8)^0 = 21_{10} \\
 2(8)^1 + 2(8)^0 = 18_{10} \\
 \times \\
 \hline
 378_{10}
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 \longrightarrow 5(8)^2 + 7(8)^1 + 2(8)^0 = 378_{10}
 \end{array}$$

↕

Ejemplo 2.38 Efectúe la siguiente multiplicación hexadecimal

$$\begin{array}{r}
 B_{16} \\
 \times 33_{16} \\
 \hline
 33 \\
 14 32 \\
 \hline
 E 1 \\
 E 1 \\
 \hline
 E F 1_{16}
 \end{array}
 \begin{array}{l}
 \longrightarrow 4(16)^1 + 11(16)^0 = 75_{10} \\
 \longrightarrow 3(16)^1 + 3(16)^0 = 51_{10} \\
 \longrightarrow 14(16)^2 + 15(16)^1 + 1(16)^0 = 3825_{10}
 \end{array}$$

La DIVISION en cualquier sistema numérico de notación posicional, -- puede llevarse a cabo por medio de las operaciones básicas de suma, resta y multiplicación, usando el mismo método que en decimal. Cabe mencionar que en la división octal y hexadecimal es necesario tener precaución con el manejo de resultado mayores que la base, los cuales deberán ajustarse a cantidades válidas dentro del sistema en que se esté trabajando.

Ejemplo 2.39 Desarrolle la siguiente división octal

$$\begin{array}{r}
 572_8 \div 25_8 \\
 \hline
 2 5 \\
 \hline
 5 2 \\
 \hline
 0 5
 \end{array}$$

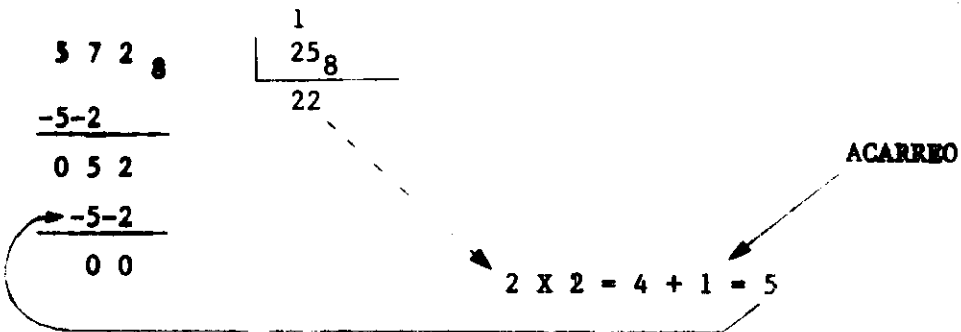
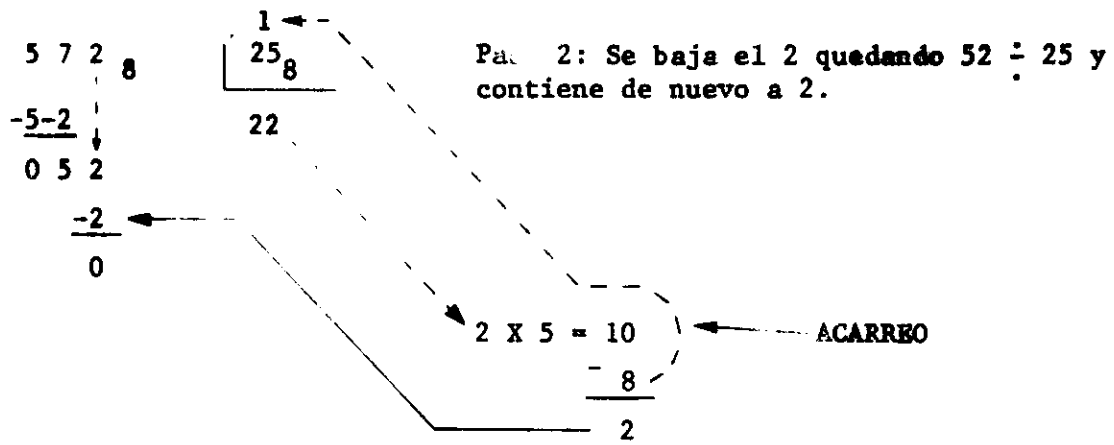
Paso 1: se separan las primeras dos cifras quedando $57 \div 25$, y contiene a 2.

$$\begin{array}{r}
 2 \times 5 = 10 \\
 - 8 \\
 \hline
 2
 \end{array}$$

ACARREO

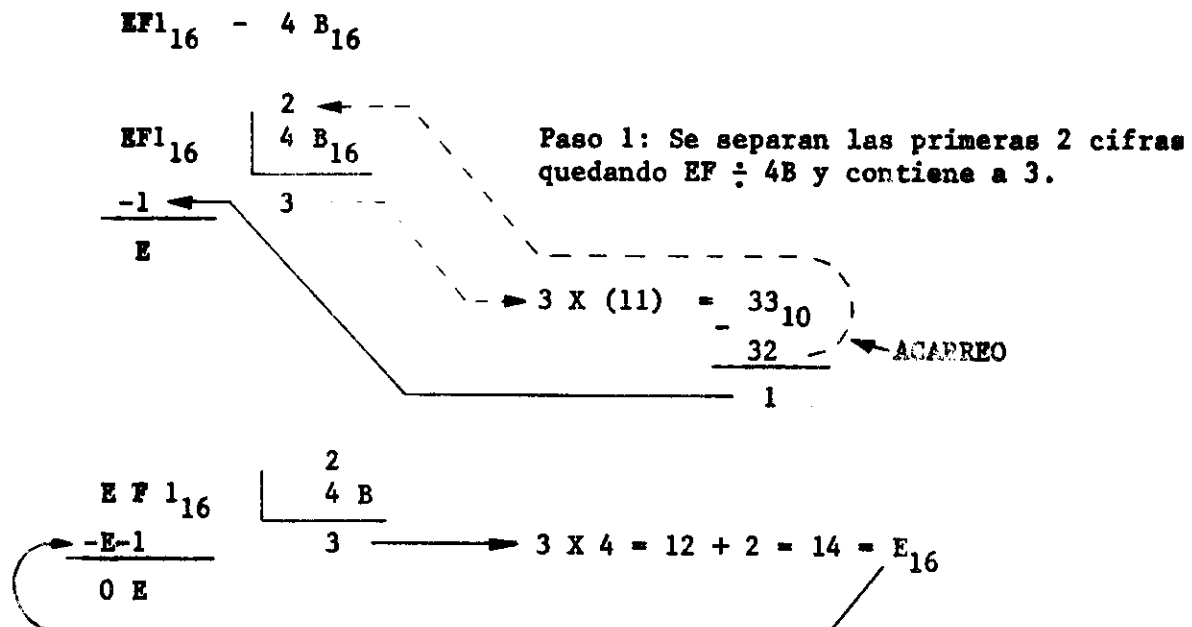
$$\begin{array}{r}
 2 \times 2 = 4 + 1 = 5
 \end{array}$$

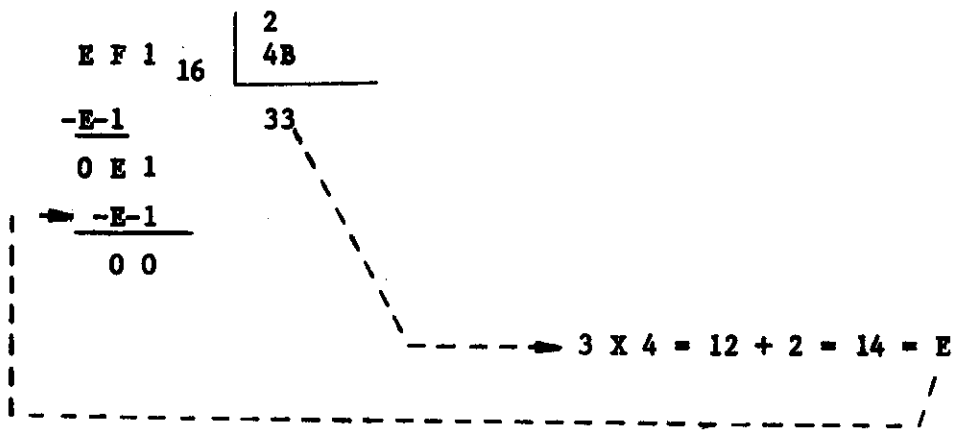
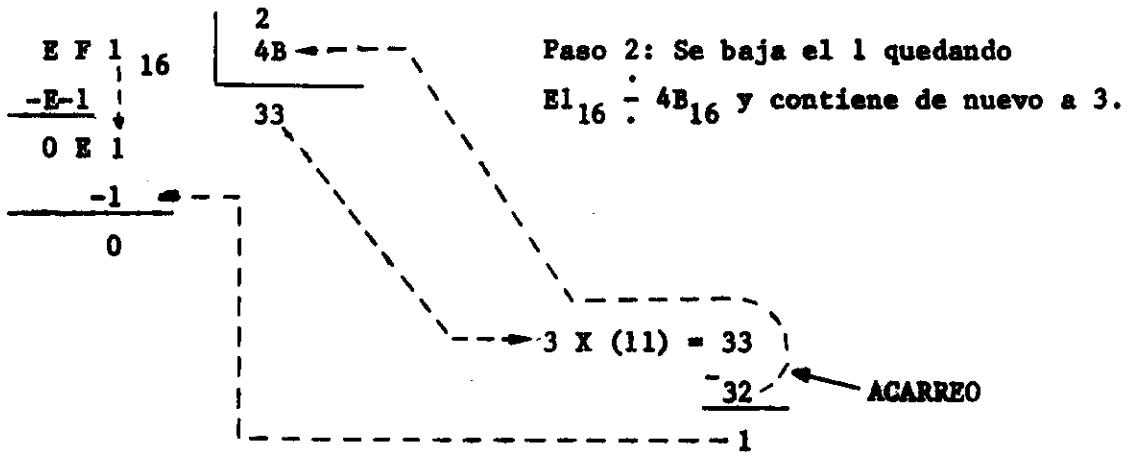
ACARREO



$$\therefore 572_8 \div 25_8 = 22_8$$

Ejemplo 2.40 Desarrolle la siguiente división hexadecimal





$$\therefore EF1_{16} \div 4B_{16} = 33_{16}$$

PROBLEMAS PROPUESTOS

- 1.-¿Que es Dígito?
- 2.- Explique que es Acarreo
- 3.-¿A que se llama "PESO" de un Dígito?
- 4.-¿Que se entiende por la base de un Sistema Numérico?
- 5.-¿Como es el Dígito Mayor de un Sistema de Notación Posicional con respecto a la base? Y escriba el dígito mayor de los siguientes sistemas:

| SISTEMA | DIGITO MAYOR |
|-------------|--------------|
| Binario | _____ |
| Octal | _____ |
| Decimal | _____ |
| Hexadecimal | _____ |

- 6.- En los siguientes ejemplos escriba cual es el dígito de mayor peso y cual es el dígito de menor peso.

| EJEMPLO | MSD | LSD |
|---------|-------|-------|
| 842 | _____ | _____ |
| 523 | _____ | _____ |
| 176 | _____ | _____ |

- 7.-¿Que es un Bit?

- 8.- Efectúe las siguientes conversiones:

| | | |
|-------------------------------------|----------------------------------|--------------------------------|
| $496_{10} \rightarrow N_2$ | $278_{10} \rightarrow N_2$ | $58_{10} \rightarrow N_8$ |
| $525_{10} \rightarrow N_8$ | $1324_{10} \rightarrow N_{16}$ | $4525_{10} \rightarrow N_{16}$ |
| $101011_{10} \rightarrow N_{10}$ | $316_8 \rightarrow N_{10}$ | $524_8 \rightarrow N_{10}$ |
| $12A_{16} \rightarrow N_{10}$ | $34DC_{16} \rightarrow N_{10}$ | $1001110_2 \rightarrow N_8$ |
| $110101011110_2 \rightarrow N_{16}$ | $110101101_2 \rightarrow N_{16}$ | $532_8 \rightarrow N_{16}$ |
| $325_8 \rightarrow N_{16}$ | $5DF_{16} \rightarrow N_8$ | $A85_{16} \rightarrow N_2$ |
| $432_8 \rightarrow N_2$ | $1547_8 \rightarrow N_2$ | |

9.- Efectúe las siguientes operaciones:
 (En la resta binaria además del método tradicional efectuarlas por el método de dos complemento).

$$111_2 + 011_2$$

$$676_8 + 420_8$$

$$9A6_{16} + 697_{16}$$

$$10110_2 + 01110_2 + 10101_2$$

$$01076_8 + 00350_8 + 07764_8$$

$$0849B_{16} + 012C5_{16} + 00D34_{16}$$

$$6523_8 + 7770_8 + 0546_8 + 1010_8$$

$$F56F_{16} + 975B_{16} + 1100_{16} + 0777_{16}$$

$$15236_8 + 07045_8 + 00456_8 + 00017_8$$

$$7FFF8_{16} + BCDE5_{16} + 0AC69_{16} + 0AAAAB_{16}$$

$$110111_2 + 111110_2 + 110001_2 + 101110_2$$

$$111111_2 + 011111_2 + 000100_2 + 010010_2 + 001001_2$$

$$100_2 - 011_2$$

$$200_8 - 067_8$$

$$AAA_{16} - 42C_{16}$$

$$4006_8 - 1657_8$$

$$0001_{16} - 1FFF_{16}$$

$$11101_2 - 10011_2$$

$$63124_8 - 05462_8$$

$$6F3FE_{16} - 22DCE_{16}$$

$$702010_8 - 030567_8$$

$$F01201_{16} - ABCDEF_{16}$$

$$1100001_2 - 0111100_2$$

$$11110000000_2 - 01010101011_2$$

$$427_8 \times 64_8$$

$$CBA_{16} \times 92_{16}$$

$$1011_2 \times 10_2$$

$$123A_{16} \times 3C_{16}$$

$$2534_8 \times 756_8$$

$$10111_2 \times 101_2$$

$$37626_8 \times 405_8$$

$$46247_8 \times 670_8$$

$$101011_2 \times 110_2$$

$$4A9B8C_{16} \times 8AD_{16}$$

$$8F46ED_{16} \times B0F_{16}$$

$$1101010_2 \times 101_2$$

$$7007_8 \div 25_8$$

$$46707_8 \div 3_8$$

$$1FE58_{16} \div 1C_{16}$$

$$FEF10A_{16} \div A_{16}$$

$$3EAF67_{16} \div 2F_{16}$$

$$1FD376_{16} \div A2_{16}$$

$$2021157_8 \div 63_8$$

$$1010111_2 \div 11_2$$

$$1111111_2 \div 100_2$$

$$26051207_8 \div 27_8$$

$$11100111_2 \div 110_2$$

$$10101011_2 \div 1011_2$$

3 *Algebra Booleana*

3.0 INTRODUCCION

Una vez que los circuitos implementados por medio de relevadores electromagnéticos adquirieron popularidad, fué necesario su estudio y sistematización. Eran redes formadas por interruptores y contactos de relevadores que por medio de combinaciones de circuitos abiertos y cerrados - desarrollaban funciones específicas.

Fué entonces cuando se encontró que una de las ramas de la teoría matemática llamada Algebra Booleana desarrollada por el matemático inglés - George Boole podía adaptarse a los circuitos de interrupción.

A diferencia del Algebra normal, las variables booleanas toman únicamente dos valores comúnmente denominados "falso" y "verdadero", que pueden relacionarse a los dos únicos estados de los circuitos de interrupción, circuito "abierto" y "cerrado".

Los símbolos 0 y 1 se usan para expresar los dos posibles valores de las variables booleanas. Si $A = 1$ usualmente significa que A es verdadera y si $A = 0$ significa que es falsa. Regresando a los interruptores, si $A = 1$ significa que el interruptor asociado con A está cerrado y si $A = 0$ significa que el interruptor está abierto.

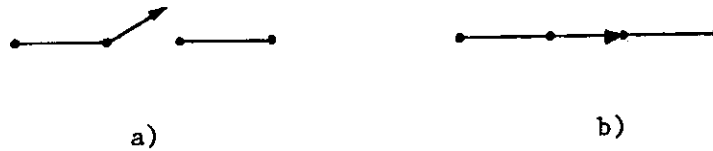


Fig. 3.1 INTERRUPTOR ASOCIADO CON LA VARIABLE "A". a) Interruptor abierto, A = FALSA, A = 0. b) Interruptor Cerrado, A = VERDADERA, A = 1.

3.1 OPERADORES LOGICOS

Las variables booleanas pueden manipularse por medio de operadores similares a los del álgebra normal comúnmente llamados "operadores lógicos".

3.1.1 OPERADOR LOGICO "AND"

Está definido para dos o más argumentos booleanos, y puede ser relacionado con el término "CONDICION", la representación más común para el operador AND es.

$$F(AB) = A.B = AB = A \cap B = A \wedge B.$$

"F" es una función de las variables booleanas A y B. Los primeros dos símbolos son los más empleados y no indican A por B sino "A AND B".

El operador AND es verdadero si y solo si todas sus variables son verdaderas. En otras palabras, es "CONDICION" de que A y B sean ambas verdaderas para que F (AB) sea verdadera.

Una variable booleana puede tomar únicamente los valores de "0" o "1" LOGICOS. Entonces para una función de m variables booleanas existen 2^m posibles combinaciones de estos valores. De aquí que una forma sencilla de expresar el comportamiento de un operador lógico sea por medio de una TABLA DE VERDAD, que consiste de un listado de todas las posibles combinaciones de las variables de entrada a un operador y el valor de la

operación o salida para cada combinación.

| AB | F(AB) = AB |
|----|------------|
| 00 | 0 |
| 01 | 0 |
| 10 | 0 |
| 11 | 1 |

| ABC | F(ABC) = A.B.C. |
|-----|-----------------|
| 000 | 0 |
| 001 | 0 |
| 010 | 0 |
| 011 | 0 |
| 100 | 0 |
| 101 | 0 |
| 110 | 0 |
| 111 | 1 |

Fig. 3.2 TABLAS DE VERDAD PARA UN OPERADOR AND de 2 y 3 VARIABLES BOOLEANAS. F(AB) ES VERDADERA UNICAMENTE CUANDO A Y B SON VERDADERAS.

El operador AND puede relacionarse con 2 o más interruptores conectados en serie con una lámpara. Esta encenderá solamente cuando ambos interruptores estén cerrados.

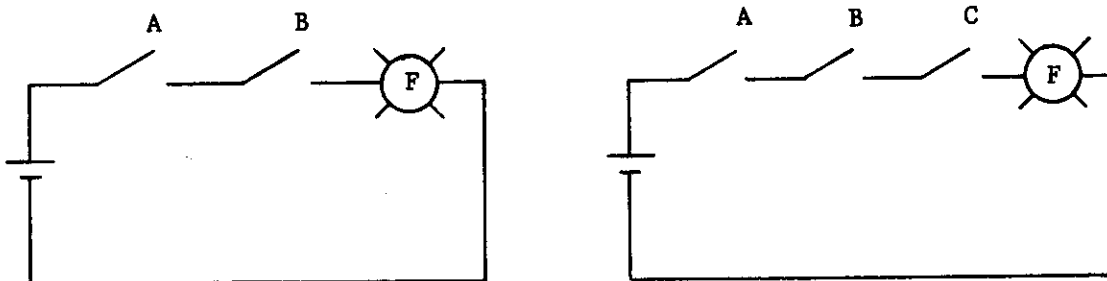


Fig. 3.3 EQUIVALENTE ELECTRICO PARA UN AND DE 2 Y 3 ENTRADAS, O VARIABLES BOOLEANAS.

Al símbolo de un operador lógico usualmente se le llama "COMPUERTA", este término proviene de los antiguos sistemas de interrupción, se decía que el contacto de un relevador, era similar a una compuerta que al abrirse o cerrarse permite el paso de señales eléctricas.



Fig. 3.4 SIMBOLOS PARA UNA COMPUERTA AND. a) Compuerta AND de 2 entradas. b) Compuerta AND de 3 entradas.

3.1.2 OPERADOR LOGICO "OR"

Está definido para dos o más argumentos booleanos y puede ser relacionado con el término "ALTERNATIVA". La representación más común para el operador OR es:

$$F(AB) = A + B = A \cup B = A \vee B$$

El primer símbolo es el más empleado, el signo (+) no significa más-sino OR.

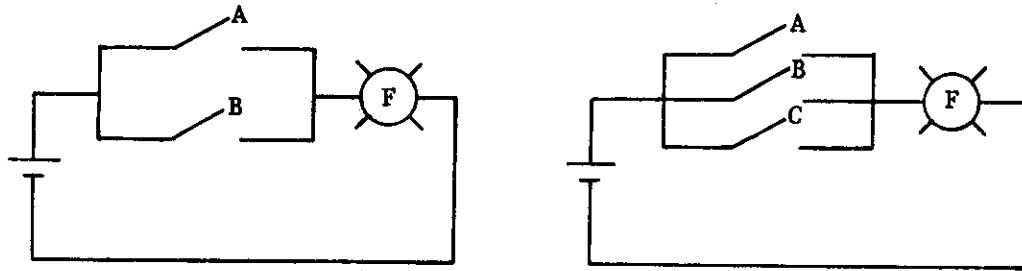
El operador "OR" es verdadero con solo y que una de sus variables sea verdadera. En otras palabras existe la ALTERNATIVA de que alguna de las variables sea verdadera para que el operador sea verdadero.

| AB | $F(AB) = A + B$ |
|----|-----------------|
| 00 | 0 |
| 01 | 1 |
| 10 | 1 |
| 11 | 1 |

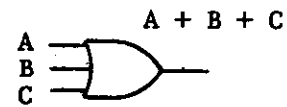
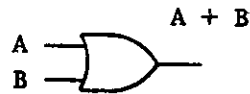
| ABC | $F(ABC) = A + B + C$ |
|-----|----------------------|
| 000 | 0 |
| 001 | 1 |
| 010 | 1 |
| 011 | 1 |
| 100 | 1 |
| 101 | 1 |
| 110 | 1 |
| 111 | 1 |

Fig. 3.5 TABLAS DE VERDAD PARA UN OPERADOR OR, DE 2 Y 3 VARIABLES. F(AB) ES VERDADERO SI A O B SON VERDADERAS.

El operador OR puede relacionarse con 2 o más interruptores conectados en paralelo con una lámpara. Esta encenderá con solo que uno de los interruptores esté cerrado.



a)



b)

Fig. 3.6 a) EQUIVALENTE ELECTRICO PARA UN OR DE 2 Y 3 ENTRADAS O VARIABLES BOOLEANAS.

b) SIMBOLO PARA UN OR DE 2 Y 3 ENTRADAS

3.1.3 OPERADOR LOGICO "NOT"

Está definido para un solo argumento booleano y su función consiste en cambiar el valor de una variable booleana por su complemento. También se le conoce como inversor o complementador. La representación más común para el operador NOT es:

$$F(A) = \bar{A} = A' = A^*$$

La tabla de verdad para un operador NOT es la siguiente:

| A | F(A) = \bar{A} |
|---|------------------|
| 0 | 1 |
| 1 | 0 |

El operador NOT puede relacionarse con un interruptor conectado en paralelo a una lámpara como se muestra en la fig. 3.7a) la lámpara encenderá cuando el interruptor esté abierto.

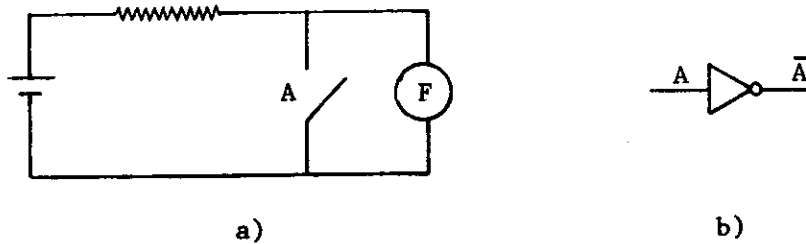


Fig. 3.7 a) EQUIVALENTE ELECTRICICO PARA UNA COMPUERTA NOT.
b) SIMBOLO

3.1.4. OPERADOR LOGICO EX-OR (EXCLUSIVE-OR)

Está definido para 2 o más argumentos booleanos. La representación del operador EX-OR es:

$$F(AB) = A \oplus B$$

El operador EX-OR es verdadero para un número impar de variables verdaderas.

| AB | $F(AB) = A \oplus B$ |
|----|----------------------|
| 00 | 0 |
| 01 | 1 |
| 10 | 1 |
| 11 | 0 |

| ABC | $F(ABC) = A \oplus B \oplus C$ |
|-----|--------------------------------|
| 000 | 0 |
| 001 | 1 |
| 010 | 1 |
| 011 | 0 |
| 100 | 1 |
| 101 | 0 |
| 110 | 0 |
| 111 | 1 |

Fig. 3.8 TABLAS DE VERDAD PARA UN OPERADOR EX-OR DE 2 Y 3 VARIABLES RESPECTIVAMENTE.

El operador EX-OR puede relacionarse con dos interruptores de un polo, dos tiros conectados como se muestra en la figura 3.9a)

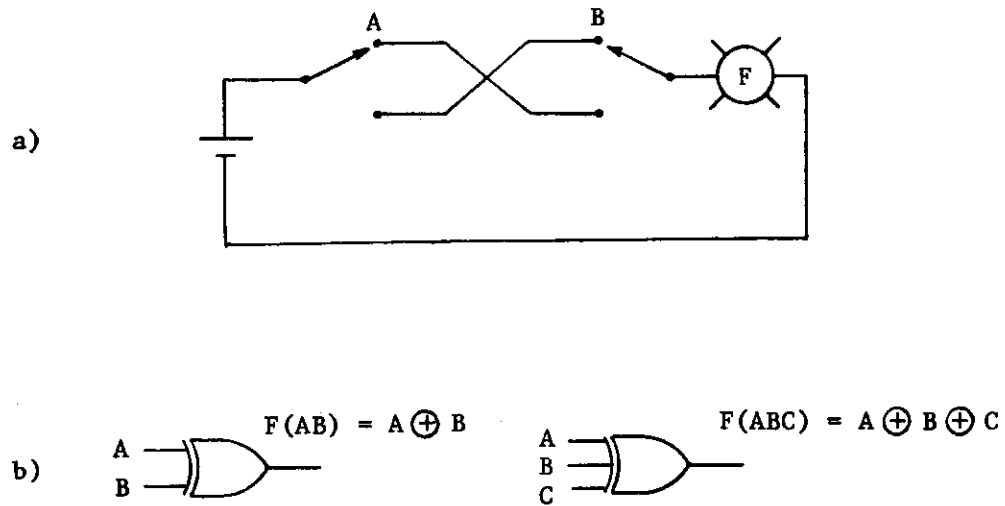


Fig. 3.9 a) EQUIVALENTE ELECTRICO PARA UNA COMPUERTA EX-OR.
b) SIMBOLOS PARA UNA COMPUERTA EX-OR DE 2 Y 3 ENTRADAS RESPECTIVAMENTE.

3.1.5 OPERADOR LOGICO "NAND"

Está definido para 1 o más argumentos booleanos. El operador NAND es la función complemento del AND, su representación es la siguiente:

$$F(AB) = \overline{A \cdot B} = A \uparrow B .$$

El operador NAND es falso si y solo si sus argumentos son verdaderos.

| AB | A.B | $\overline{A \cdot B}$ |
|----|-----|------------------------|
| 00 | 0 | 1 |
| 01 | 0 | 1 |
| 10 | 0 | 1 |
| 11 | 1 | 0 |

Fig. 3.10 TABLA DE VERDAD PARA UN NAND

El operador NAND puede relacionarse con un par de interruptores conectados en paralelo a una lámpara, como se muestra en la fig. 3.11.

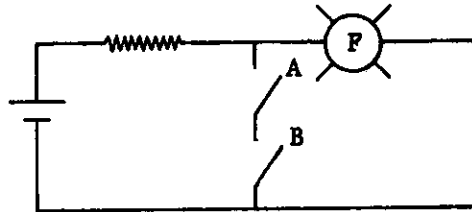


Fig. 3.11 EQUIVALENTE ELECTRICO PARA UN NAND

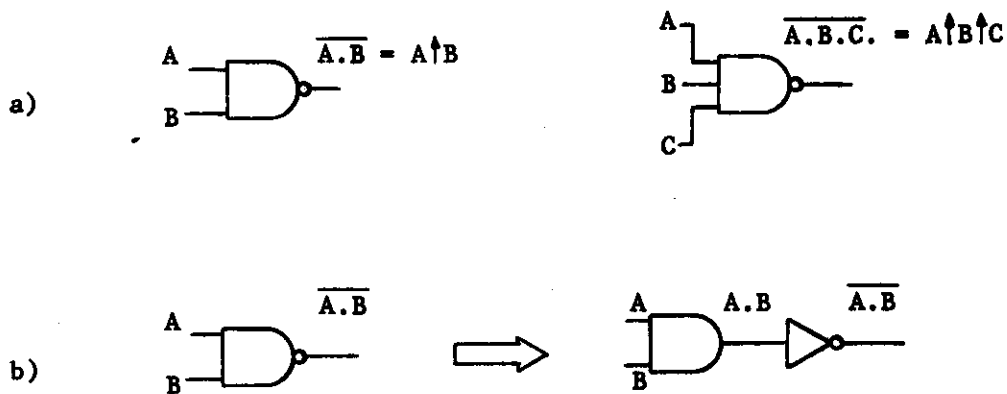


Fig. 3.12 a) SIMBOLOS PARA UNA COMPUERTA NAND DE 2 Y 3 ENTRADAS
b) UN NAND ES IGUAL A UN AND NEGADO.

3.1.6 OPERADOR LOGICO "NOR"

Está definido para 1 o más argumentos booleanos. El operador NOR es la función complemento del OR, su representación es la siguiente:

$$F(AB) = \overline{A + B} = A \downarrow B$$

El operador NOR es verdadero si y solo si todo sus argumentos son falsos.

| AB | A + B | $\overline{A + B}$ |
|----|-------|--------------------|
| 00 | 0 | 1 |
| 01 | 1 | 0 |
| 10 | 1 | 0 |
| 11 | 1 | 0 |

Fig. 3.13 TABLA DE VERDAD PARA UN NOR

El operador NOR puede relacionarse a un par de interruptores conectados en paralelo a una lámpara, fig. 3.14.

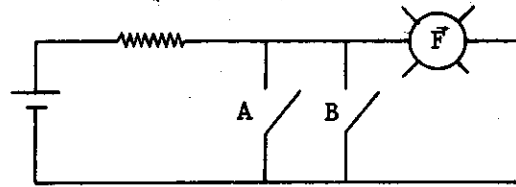


Fig. 3.14 EQUIVALENTE ELECTRICO PARA UN NOR

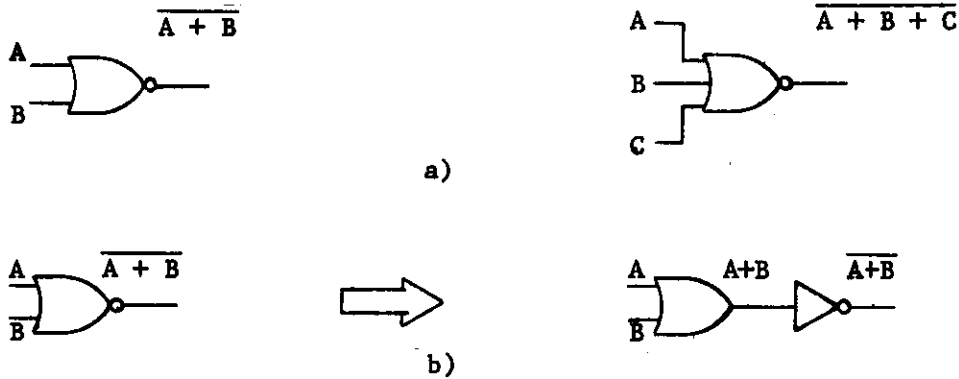


Fig. 3.15 a) SIMBOLO PARA UN NOR DE 2 Y 3 ENTRADAS
b) EL NOR ES IGUAL A UN OR NEGADO.

3.1.7 OPERADOR LOGICO COINCIDENCE.

El operador lógico coincidence es la función complemento del EX-OR, - también se le conoce como EX-NOR. Su representación es la siguiente:

$$F(AB) = A \odot B$$

El coincidence es falso para un número impar de variables verdaderas.

| AB | A \odot B |
|----|-------------|
| 00 | 1 |
| 01 | 0 |
| 10 | 0 |
| 11 | 1 |

Fig. 3.16 TABLA DE VERDAD PARA UN COINCIDENCE

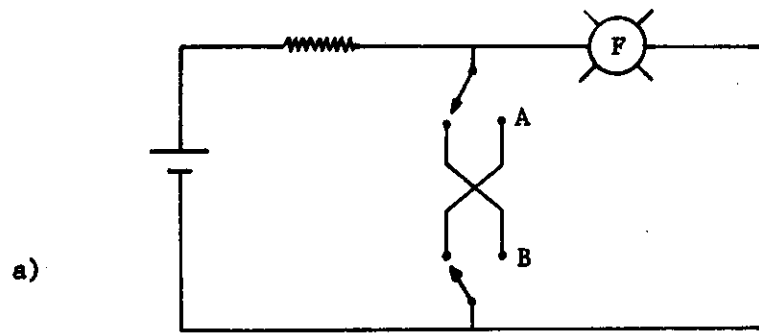


Fig. 3.17 a) EQUIVALENTE ELECTRICO PARA UNA COMPUERTA CONCIDEN-
 CE.
 b) SIMBOLO PARA UNA COMPUERTA COINCIDENCE.

3.2 EXPRESIONES BOOLEANAS

La aplicación de los operadores básicos a una o más variables o constantes forman lo que se conoce como EXPRESIONES BOOLEANAS. Las expresiones booleanas más simples consisten en una sola variable o constante, por ejemplo, A, \bar{B} , 1, etc. La formación de expresiones más complicadas se lleva a cabo combinando expresiones simples por medio de AND'S, OR'S y NOT'S, por ejemplo:

a) $A + BC$

b) $A(\bar{B}+C)$

Los paréntesis se usan para indicar el orden en que se deben ejecutar las operaciones booleanas. Cuando no existen paréntesis en el inciso b) debe ejecutarse primero la complementación, después el AND y por último el OR.

Cada expresión corresponde a un circuito de compuertas lógicas, como se muestra en el ejemplo 3.1.

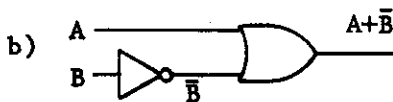
Ejemplo 3.1



La evaluación de una expresión se hace sustituyendo los valores de 0 y 1 para cada variable. Una tabla de verdad es un método útil para este propósito, puesto que muestra todas las posibles combinaciones de los valores de las variables y su salida.

Ejemplo 3.2

a) $A + \bar{B}$



c)

| AB | \bar{B} | $A + \bar{B}$ |
|----|-----------|---------------|
| 00 | 1 | 1 |
| 01 | 0 | 0 |
| 10 | 1 | 1 |
| 11 | 0 | 1 |

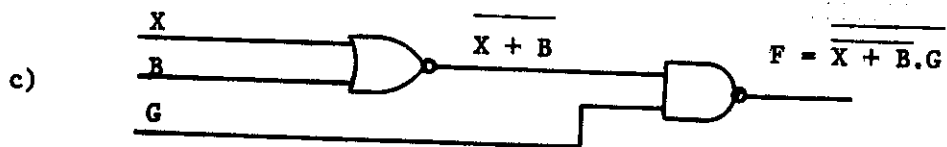
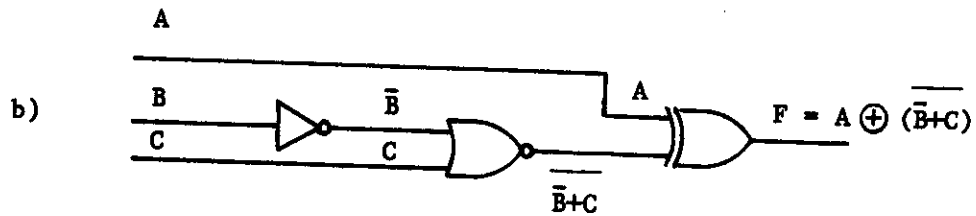
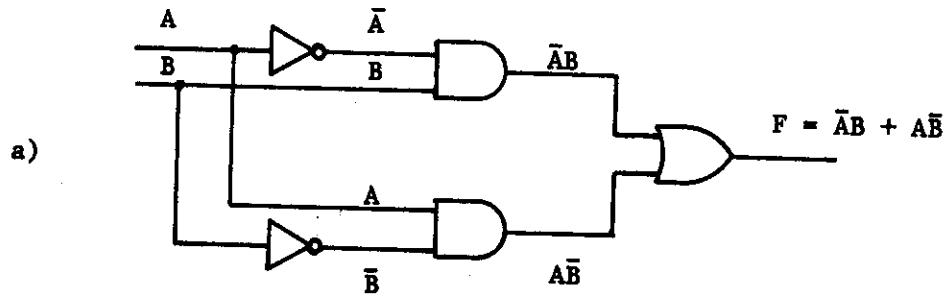
a) Función o expresión booleana, b) Circuito, c) Tabla de verdad,

Encuentre el circuito para las siguientes funciones booleanas

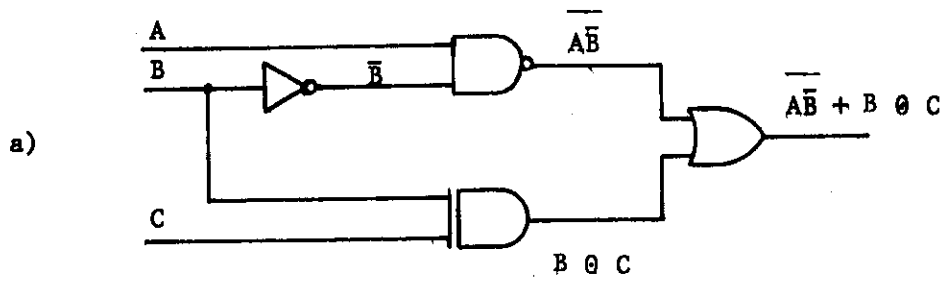
a) $F = \bar{A}\bar{B} + \bar{A}B$

b) $F = \overline{A \oplus (\bar{B} + C)}$

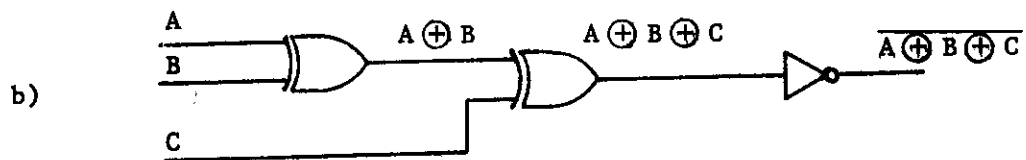
c) $F = \overline{X + B} \cdot G$



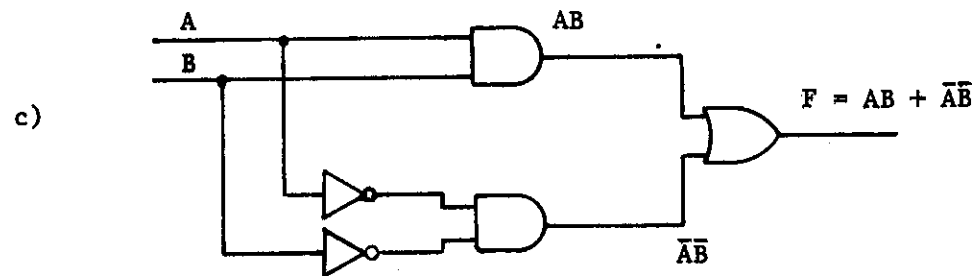
Determine las ecuaciones de los siguientes circuitos:



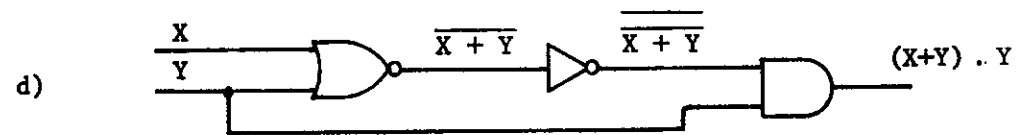
$$F(ABC) = \overline{A\overline{B}} + B \oplus C$$



$$F(ABC) = \overline{A \oplus B \oplus C}$$



$$F(ABC) = AB + \overline{A\overline{B}}$$



$$F(XY) = (X+Y) \cdot Y$$

3.3 PROPIEDADES FUNDAMENTALES DEL ALGEBRA BOOLEANA

Las siguientes proposiciones son las elementales en el álgebra booleana, algunas de ellas no son correctas para el álgebra normal.

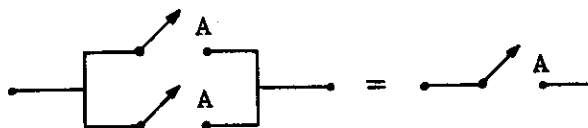
$$\begin{array}{ll}
 a \cdot a = a & a + a = a \\
 a \cdot \bar{a} = 0 & a + \bar{a} = 1 \\
 a \cdot 0 = 0 & a + 0 = a \\
 a \cdot 1 = a & a + 1 = 1
 \end{array}
 \qquad \bar{\bar{a}} = a$$

La comprobación de estas proposiciones se ve obvia por simple inspección sin embargo pueden verificarse usando tablas de verdad o por medio de sus equivalentes eléctricos.

Ejemplo:

Pruebe que $a + a = a$

| a | $a + a = a$ |
|---|-------------|
| 0 | $0 + 0 = 0$ |
| 1 | $1 + 1 = 1$ |



3.3.1 LEYES FUNDAMENTALES

Ley asociativa:

$$\begin{aligned}
 (a + b) + c &= a + (b + c) \\
 (ab)c &= a(bc)
 \end{aligned}$$

Ley conmutativa:

$$\begin{aligned}
 a + b &= b + a \\
 ab &= ba
 \end{aligned}$$

Ley distributiva:

$$\begin{aligned}
 a(b+c) &= ab + ac \\
 a+bc &= (a+b) \cdot (a+c)
 \end{aligned}$$

Aparentemente la última ecuación es incorrecta con respecto al álgebra normal. Puede probarse mediante una tabla de verdad o empleando los postulados anteriormente descritos.

$$\begin{aligned}
 a + bc &= (a+b) (a+c) \\
 &= aa + ac + ba + bc \\
 &= a + ac + ab + bc \\
 &= a (1 + c + b) + bc \\
 &= a (1) + bc \\
 &= a + bc
 \end{aligned}$$

3.4 TEOREMA DE D'MORGAN

Para obtener el complemento o inverso de una expresión booleana se aplica el teorema de "D'MORGAN". En su forma más general establece que para complementar una función booleana expresada por medio de AND, OR y NOT, es necesario:

- 1.- Reemplazar todos los operadores AND por OR.
- 2.- Reemplazar todos los operadores OR por AND.
- 3.- Reemplazar todas las variables por su complemento.

Aplicando el teorema de D'MORGAN para dos argumentos tenemos:

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

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

| ab | $\overline{a+b}$ | $\bar{a} \cdot \bar{b}$ |
|----|----------------------|-----------------------------|
| 00 | $\overline{0+0} = 1$ | $\bar{0} \cdot \bar{0} = 1$ |
| 01 | $\overline{0+1} = 0$ | $\bar{0} \cdot \bar{1} = 0$ |
| 10 | $\overline{1+0} = 0$ | $\bar{1} \cdot \bar{0} = 0$ |
| 11 | $\overline{1+1} = 0$ | $\bar{1} \cdot \bar{1} = 0$ |

| ab | $\overline{a \cdot b}$ | $\bar{a} + \bar{b}$ |
|----|----------------------------|-------------------------|
| 00 | $\overline{0 \cdot 0} = 1$ | $\bar{0} + \bar{0} = 1$ |
| 01 | $\overline{0 \cdot 1} = 1$ | $\bar{0} + \bar{1} = 1$ |
| 10 | $\overline{1 \cdot 0} = 1$ | $\bar{1} + \bar{0} = 1$ |
| 11 | $\overline{1 \cdot 1} = 0$ | $\bar{1} + \bar{1} = 0$ |

Ejemplo 3.3

Complemente la siguiente función:

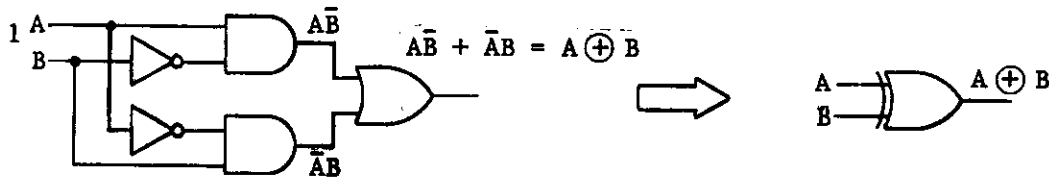
$$\begin{aligned}
 F &= AC + B\bar{D} \\
 \bar{F} &= \overline{AC + B\bar{D}} \\
 \bar{F} &= (\bar{A} + \bar{C}) \cdot (\bar{B} + D)
 \end{aligned}$$

3.5 LA FORMA "A.O.N."-AND, OR, NOT

Todas las funciones booleanas pueden expresarse por medio de los operadores lógicos AND, OR y NOT, tal es el caso de los operadores EX-OR- y coincidence.

$$A \oplus B = A \bar{B} + \bar{A} B$$

La expresión anterior es la forma AON para el EX-OR. El circuito de la expresión anterior es el siguiente:



Para comprobar la expresión anterior usaremos una tabla de verdad.

| AB | $A\bar{B}$ | $\bar{A}B$ | $A\bar{B} + \bar{A}B$ |
|----|------------|------------|-----------------------|
| 00 | 0 | 0 | 0 |
| 01 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 |
| 11 | 0 | 0 | 0 |

\Rightarrow

| AB | $A \oplus B$ |
|----|--------------|
| 00 | 0 |
| 01 | 1 |
| 10 | 1 |
| 11 | 0 |

variables de entrada
 salida del circuito

La expresión para el COINCIDENCE se obtiene complementando, $A \oplus B$

$$\begin{aligned} \overline{A \oplus B} &= \overline{A \bar{B} + \bar{A} B} \\ &= (\bar{A} + B) \cdot (A + \bar{B}) \\ &= \bar{A}A + \bar{A}\bar{B} + BA + B\bar{B} \\ &= 0 + \bar{A}\bar{B} + AB + 0 \end{aligned}$$

$$\overline{A \oplus B} = \bar{A}\bar{B} + AB$$

$$\therefore = A \ominus B = \bar{A}\bar{B} + AB$$

Forma A.O.N. para un coincidence

3.6 EXPRESION DE FUNCIONES BOOLEANAS A PARTIR DE NAND Y NOR

La expresión de funciones booleanas a partir de NAND'S y NOR'S es una alternativa a la forma AON, es decir, con un solo tipo de dispositivo lógico podemos implementar cualquier circuito.

Esta propiedad es de gran utilidad en la práctica, puesto que no hay necesidad de disponer de una gran cantidad de compuertas AND, OR y NOT.

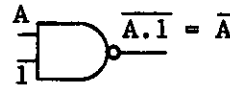
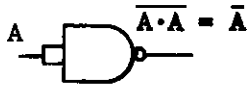
NOT A PARTIR DE NAND

El operador NAND puede hacer la función de un NOT de 2 formas. La primera es efectuando la operación NAND con la misma variable.

$$\overline{A \cdot A} = \bar{A}$$

Y la segunda es combinando la variable con un "1" lógico.

$$\overline{A \cdot 1} = \bar{A}$$

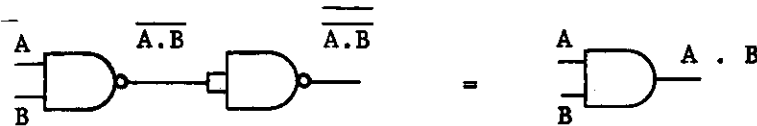


Conexiones para obtener un NOT a partir de un NAND

AND A PARTIR DE NAND

Para obtener un AND es necesario negar la salida del NAND.

$$\begin{aligned} A \cdot B &= \overline{\overline{A \cdot B}} \\ &= \overline{A \uparrow B} \end{aligned}$$

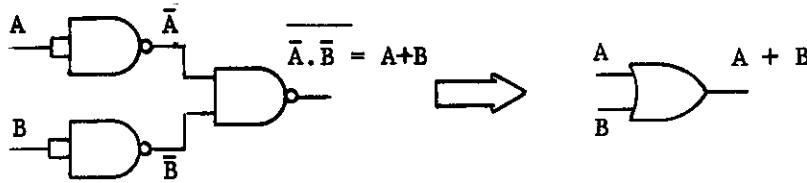


OR A PARTIR DE NAND

Para obtener un OR a partir de NAND'S es necesario cambiar el (+) - OR por un ($\bar{\cdot}$) punto negado o NAND.

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

$$= \overline{\bar{A} \cdot \bar{B}} = \bar{A} \uparrow \bar{B}$$



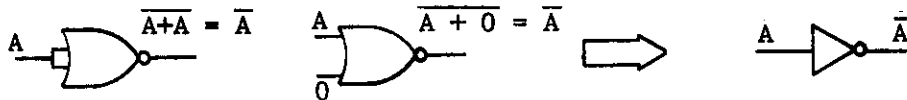
NOT A PARTIR DE NOR

Igual que el NAND el NOR puede actuar como NOT de 2 formas. La primera es efectuando la operación NOR con la misma variable.

$$\overline{A + A} = \bar{A}$$

Y la segunda es combinando la variable con un "0" lógico.

$$\overline{A + 0} = \bar{A}$$

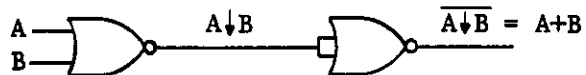


Conexiones para obtener un NOT a partir de un NOR

OR A PARTIR DE UN NOR

Para obtener un OR es necesario negar la salida del NOR

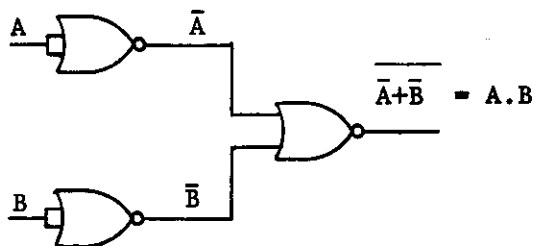
$$\begin{aligned} A + B &= \overline{\overline{A + B}} \\ &= \overline{A \downarrow B} \end{aligned}$$



AND A PARTIR DE UN NOR

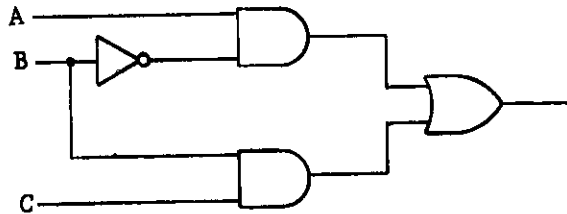
Para obtener un AND a partir de un NOR es necesario cambiar el (.)-AND por un ($\bar{+}$) OR negado.

$$\begin{aligned} A \cdot B &= \overline{\overline{A \cdot B}} \\ &= \overline{\bar{A} + \bar{B}} \\ &= \bar{\bar{A}} \downarrow \bar{\bar{B}} \end{aligned}$$



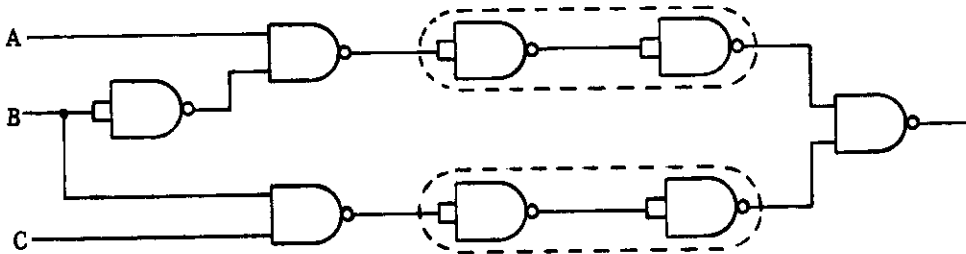
Ejemplo 3.4

Transformar el siguiente circuito implementado con compuertas AND, - OR y NOT a uno que solo contenga compuertas NAND.



PROCEDIMIENTO:

- 1.- Substituir cada elemento por su equivalente en Nand's
- 2.- Dos negaciones seguidas deben eliminarse



Al substituir los elementos del circuito anterior por su equivalente en NAND'S, es necesario eliminar 2 pares de NAND'S consecutivos.

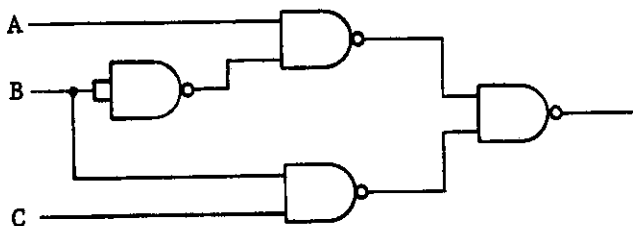
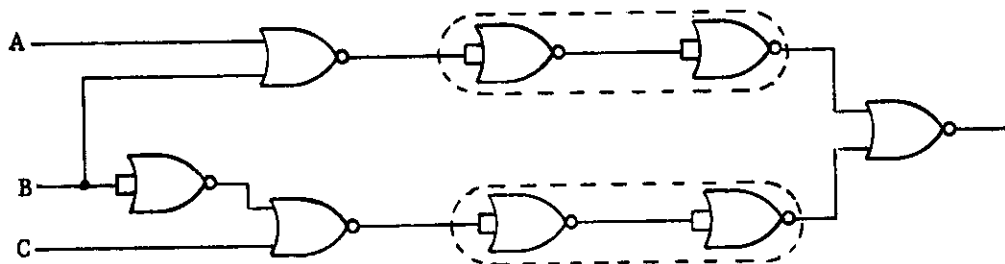
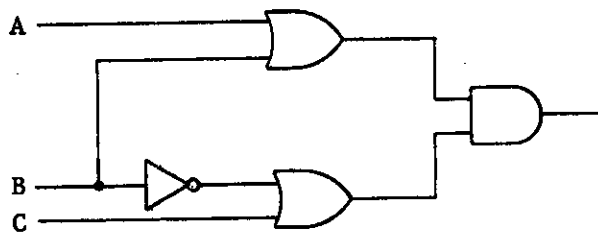


Fig. 3.18 CIRCUITO RESULTANTE

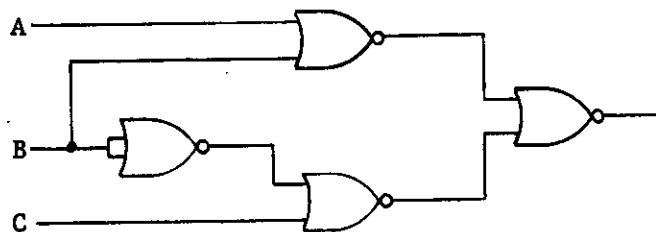
Las compuertas electrónicas se fabrican en paquetes llamados CIRCUI TOS INTEGRADOS que generalmente tienen solo compuertas del mismo tipo, - por ejemplo un circuito con compuertas AND contiene solamente compuertas AND. Para implementar el circuito equivalente anterior es necesario un solo CIRCUITO INTEGRADO, mientras que el circuito original necesita 3.

Ejemplo 3.5

Con el procedimiento del ejemplo anterior transforme el siguiente - circuito a uno que solo contenga compuertas NOR.



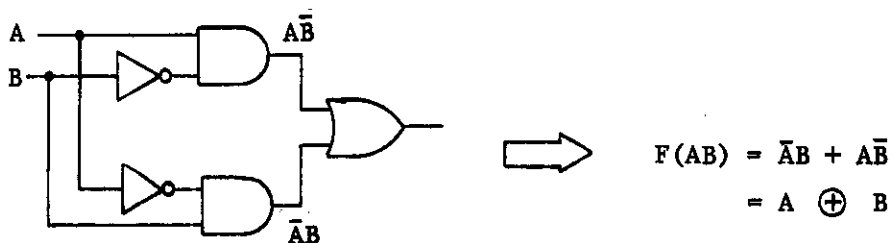
SUSTITUCION A COMPUERTAS NOR



CIRCUITO EQUIVALENTE

3.7 ORIGEN DE LAS FUNCIONES BOOLEANAS, MINITERMINOS

Como se discutió en el punto 3.2 para cada expresión booleana corresponde un circuito implementado por compuertas lógicas. Esa expresión es comúnmente llamada FUNCION BOOLEANA y representa el comportamiento de un circuito determinado. En el punto 3.5 podemos observar un ejemplo bastante ilustrativo. La ecuación booleana se obtiene haciendo pasar las variables a través de cada compuerta. La salida es una función de las variables de entrada, en este caso es una función de A y B, $F(AB)$. En esta forma se puede obtener una función (ecuación) a partir de un circuito.

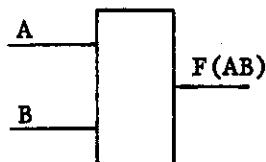


El proceso inverso, obtener un circuito a partir de una función booleana también discutido en el punto 3.2 no tiene el menor problema. Veamos ahora como obtener la función de un bloque cuyo circuito no conocemos. Imaginemos para el caso un bloque con 2 entradas y una sola salida.

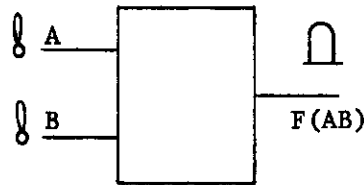
Procedimiento a seguir:

1.- Las entradas obviamente son variables booleanas

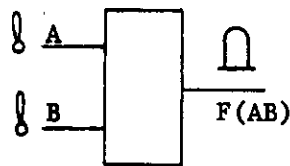
Asignémosle pues un nombre, por ejemplo A y B. La salida tendrá que llamarse $F(AB)$, F de AB.



2.- Por medio de un par de interruptores cuyas salidas sean niveles lógicos, substituiremos las variables por UNOS y CEROS LOGICOS, la función de salida será monitoreada por una lámpara, si $F(AB) = 1$ (verdadera) la lámpara encenderá, si $F(AB) = 0$, (falsa), la lámpara no encenderá.



3. Como la función de salida es una ecuación que representa el comportamiento del bloque, obtengamos entonces su comportamiento substituyendo los valores de A y B por 1'S y 0'S, indiquemos en una tabla el valor de la salida para cada combinación.



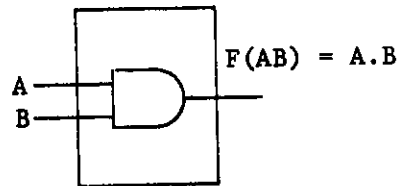
| AB | F(AB) = ? |
|----|-----------|
| 00 | 0 |
| 01 | 0 |
| 10 | 0 |
| 11 | 1 |

4. Supongamos que los valores que aparecen en la tabla anterior son los correspondientes al bloque. $F(AB)$ es VERDADERA solo una vez, cuando A y B son verdaderas y es falsa $\bar{F}(AB)$, en las restantes 3 combinaciones. Por lo tanto para que $F(AB)$ sea verdadera es CONDICION de que A y B sean "ambas" verdaderas, de aquí que:

$$F(AB) = A \cdot B$$

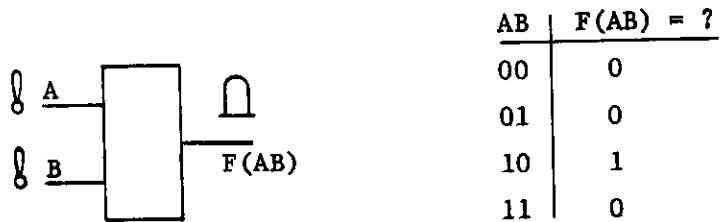
$F(AB)$ es igual a A AND B

5. Podemos concluir que el circuito que se encuentra en el bloque que -- analizamos tiene el comportamiento de una compuerta AND.



Ejemplo 3.6

Tomemos otro bloque cuya tabla de verdad sea:

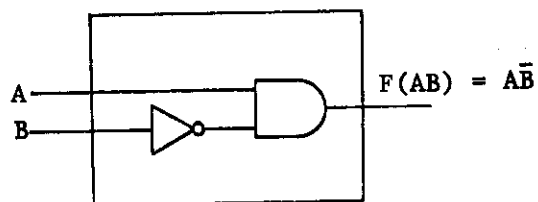


En este caso para que F(AB) sea verdadera es CONDICION de que A sea verdadera y B falsa.

$$F(AB) = A \cdot \bar{B}$$

F(AB) es igual a A AND \bar{B}

El circuito dentro del bloque es el siguiente:



Podemos observar que $A.B$ para el primer ejemplo y que $A.\bar{B}$ para el segundo, son verdaderas solo una vez, es decir existe un solo uno para todas las combinaciones.

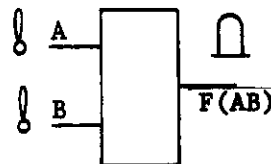
| AB | $F(AB) = A.B$ |
|----|---------------|
| 00 | 0 |
| 01 | 0 |
| 10 | 0 |
| 11 | 1 |

En esta tabla el término $A.B$ tiene un número mínimo de uno's en su salida, por este motivo se le conoce como MINITERMINO.

El MINITERMINO es un término producto que contiene todas las variables de la función ya sea en su forma normal o complementada.

Ejemplo 3.7

Tomemos un tercer bloque, cuya tabla de verdad sea la siguiente:



| AB | $F(AB) = ?$ |
|----|-------------|
| 00 | 0 |
| 01 | 1 |
| 10 | 1 |
| 11 | 0 |

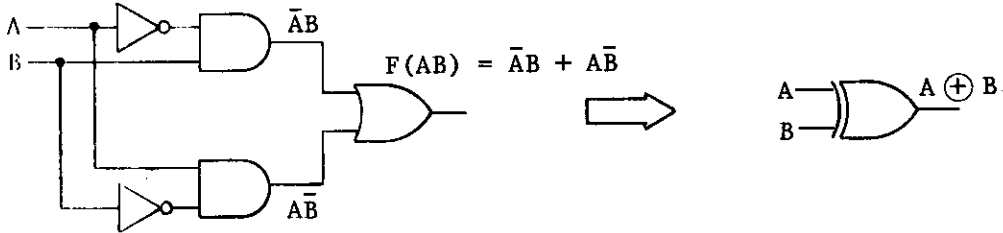
En este caso $F(AB)$ es verdadera en 2 ocasiones, es decir existe la ALTERNATIVA de que $F(AB)$ sea 2 veces verdadera, una cuando se presente la CONDICION de \bar{A} y B, y la otra cuando se presente la condición de A y \bar{B} .

De aquí que:

$$F(AB) = \bar{A}.B + A.\bar{B}$$

$$F(AB) \text{ es igual a } \bar{A} \text{ AND } B, \text{ OR, } A \text{ AND } \bar{B}$$

El circuito dentro del bloque es un EX-OR



En este ejemplo existen 2 minitérminos, el $\bar{A}B$ y el $A\bar{B}$ y la función de AB , $F(AB)$ es igual a la suma (OR) de estos 2 minitérminos.

$$F = \Sigma \text{ de los minitérminos}$$

Un sistema de n variables de entrada tendrá 2^n diferentes minitérminos. Los minitérminos pueden expresarse por medio de una "m" minúscula con un sub-índice decimal correspondiente al número binario que representa el minitérmino.

Ejemplo 3.8

Enuncie los minitérminos para un sistema de 2 variables.

| AB | MINITERMINOS |
|----|------------------------|
| 00 | $\bar{A}\bar{B} = m_0$ |
| 01 | $\bar{A}B = m_1$ |
| 10 | $A\bar{B} = m_2$ |
| 11 | $AB = m_3$ |

Puesto que una función es igual a la sumatoria de sus minterminos tenemos:

| AB | F(AB) |
|----|-------|
| 00 | 1 |
| 01 | 1 |
| 10 | 1 |
| 11 | 0 |

$$F(AB) = \bar{A}\bar{B} + \bar{A}B + A\bar{B}$$

$$F(AB) = m_0 + m_1 + m_2$$

$$F(AB) = \Sigma m_0, m_1, m_2$$

$$F(AB) = \Sigma 0, 1, 2 \quad \text{FORMA CANONICA}$$

A esta forma minimizada se le conoce como SUMATORIA DE PRODUCTOS (SOP) o forma canónica.

Ejemplo 3.9

Obtenga la forma canónica de la expresión booleana para el sistema cuya tabla de verdad se muestra a continuación.

| ABC | F(ABC) |
|-----|--------|
| 000 | 1 |
| 001 | 1 |
| 010 | 0 |
| 011 | 0 |
| 100 | 0 |
| 101 | 0 |
| 110 | 0 |
| 111 | 1 |

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

$$F(ABC) = m_0 + m_1 + m_7$$

$$F(ABC) = \Sigma 0, 1, 7$$

Ejemplo 3.10

Obtenga la tabla de verdad de la siguiente función booleana expresada en su forma canónica.

$$F(ABC) = \Sigma 1, 4, 5, 7$$

| ABC | F(ABC) = $\Sigma 1, 4, 5, 7$ |
|-----|------------------------------|
| 000 | 0 |
| 001 | 1 |
| 010 | 0 |
| 011 | 0 |
| 100 | 1 |
| 101 | 1 |
| 110 | 0 |
| 111 | 1 |

3.8 \bar{F} COMO ALTERNATIVA, MAXITERMINOS

Para obtener una expresión booleana a partir de una tabla de verdad se hace uso de la F AFIRMADA, o VERDADERA sin embargo \bar{F} o (FALSA), puede ser una alternativa muy útil, sobre todo cuando se tienen pocos "0" - - CEROS en la función.

Ejemplo 3.11

| ABC | F(ABC) |
|-----|--------|
| 000 | 0 |
| 001 | 0 |
| 010 | 1 |
| 011 | 1 |
| 100 | 1 |
| 101 | 1 |
| 110 | 1 |
| 111 | 1 |

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

$$\bar{F}(ABC) = \sum 0, 1$$

Pero realmente no nos interesa \bar{F} (NEGADA) sino F (AFIRMADA). Aplicando el teorema de D'Morgan tenemos:

$$F(ABC) = \overline{\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C}$$

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

A esta forma se le conoce como PRODUCTO DE SUMATORIAS (POS) y es una alternativa al (SOP) sumatoria de productos. A los términos de la forma $(A+B+C)$ se les llama maxitérminos y al igual que los minitérminos deben - contener todas las variables de la función ya sea en su forma normal o -- complementada.

El nombre de maxitérmino surge de la tabulación de un solo maxitérmi no.

| ABC | (A+B+C) |
|-----|---------|
| 000 | 0 |
| 001 | 1 |
| 010 | 1 |
| 011 | 1 |
| 100 | 1 |
| 101 | 1 |
| 110 | 1 |
| 111 | 1 |

La salida contiene únicamente un "0" CERO, o sea un número MAXIMO de UNOS.

La expresión del ejemplo 3.11 para maxitérminos queda:

$$F(ABC) = (A+B+C) \cdot (A+B+C)$$

$$F(ABC) = M_7 \cdot M_6$$

$$F(ABC) = \Pi 6, 7$$

A esta forma se le conoce también como forma canónica conjuntiva. - Algunos autores no coinciden con nombrar a los maxitérminos en esta forma, el término (A+B+C) lo toman como (000) M₀ en lugar de M₇ (111).

3.9 LAS OCHO FORMAS ESTANDARD

En los puntos 3.7 y 3.8 se vió como una expresión booleana que representa el comportamiento de un bloque, puede expresarse por medio de la sumatoria de sus minitérminos. También llamada forma AND/OR (debido a que las variables pasan primero a través de compuertas AND y después a una compuerta OR), o también por medio del producto de sus maxitérminos llamado forma OR/AND.

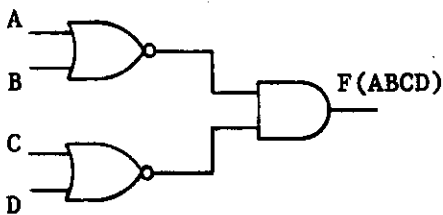
En el punto 3.6 se discutió el método para transformar un circuito a compuertas NAND, forma NAND/NAND y a compuertas NOR o forma NOR/NOR.

Con el propósito de disponer de una mayor versatilidad a la hora de implementar un determinado circuito por medio de compuertas, podemos combinar los 4 operadores, AND, OR, NAND y NOR, con lo cual lograríamos 16 posibles combinaciones, sin embargo solo se usan 8.

- 1.- AND/OR
- 2.- NAND/NAND
- 3.- OR/NAND
- 4.- NOR/OR
- 5.- AND/NOR
- 6.- NAND/AND
- 7.- OR/AND
- 8.- NOR/NOR

Esto es debido a que las 8 restantes no configuran una función de acuerdo a la sumatoria de productos o al producto de sumatorias.

Ejemplo 3.12



$$\begin{aligned} F(ABCD) &= \overline{A+B} \cdot \overline{C+D} \\ &= \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \end{aligned}$$

FORMA NOR/AND

Considerando que se dispone de las variables y sus complementos, podemos obtener las 8 formas a partir de F y \bar{F} . En 2 grupos.

Ejemplo 3.13

Desarrolle las 8 formas estandar para la función definida por la siguiente tabla de verdad.

| AB | F(AB) | |
|----|-------|-------------------------------------|
| 00 | 1 | $F(AB) = \bar{A}\bar{B} + AB$ |
| 01 | 0 | |
| 10 | 0 | $\bar{F}(AB) = \bar{A}B + A\bar{B}$ |
| 11 | 1 | |

1. A PARTIR DE F , GRUPO AND/OR

$$\begin{aligned}
 F(AB) &= \bar{A}\bar{B} + AB && \text{FORMA (AND/OR)} \\
 &= \overline{\overline{\bar{A}\bar{B} + AB}} \\
 &= \overline{\bar{A}\bar{B}} \cdot \overline{AB} = (\bar{A} \uparrow \bar{B}) \uparrow (A \uparrow B) && \text{FORMA (NAND/NAND)} \\
 &= (A+B) \cdot (\bar{A}+\bar{B}) = (A+B) \uparrow (\bar{A}+\bar{B}) && \text{FORMA (OR/NAND)} \\
 &= \overline{(A+B)} + \overline{(\bar{A}+\bar{B})} = (A \downarrow B) + (\bar{A} \downarrow \bar{B}) && \text{FORMA (NOR/OR)}
 \end{aligned}$$

2. A PARTIR DE \bar{F} , GRUPO OR/AND

$$\begin{aligned}
 \bar{F}(AB) &= A\bar{B} + \bar{A}B \\
 F(AB) &= \overline{A\bar{B} + \bar{A}B} = (\bar{A}B) \downarrow (A\bar{B}) && \text{FORMA (AND/NOR)} \\
 &= \overline{(\bar{A}B)} \cdot \overline{(A\bar{B})} = (\bar{A} \uparrow B) \cdot (A \uparrow \bar{B}) && \text{FORMA (NAND/AND)} \\
 &= (A+\bar{B}) \cdot (\bar{A}+B) && \text{FORMA (OR/AND)} \\
 &= \overline{(A+\bar{B})} + \overline{(\bar{A}+B)} \\
 &= (A \downarrow \bar{B}) + (\bar{A} \downarrow B) = (A \downarrow \bar{B}) \downarrow (\bar{A} \downarrow B) && \text{FORMA (NOR/NOR)}
 \end{aligned}$$

Para obtener las 3 formas restantes, estando en el grupo AND/OR, o el grupo OR/AND, basta con aplicar sucesivamente el teorema de D'MORGAN, como se puede observar en los 2 casos anteriores.

En el ejemplo siguiente se muestra la forma de cambiar de un grupo a otro.

Ejemplo 3.14

Convertir, $F = (A + B) (C + D)$ de la forma OR/AND a la forma AND/OR.

$$F = (A+B) (C+D) \qquad \text{DESARROLLANDO EL PRODUCTO}$$

$$F = AC + AD + BC + BD \qquad \text{FORMA (AND/OR)}$$

Ejemplo 3.15

Convertir $F = A\bar{B} + \bar{A}B$ de la forma AND/OR a la forma OR/AND.

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

$$= (A\bar{B}) + \bar{A}B \qquad \text{APLICANDO LA LEY DISTRIBUTIVA}$$

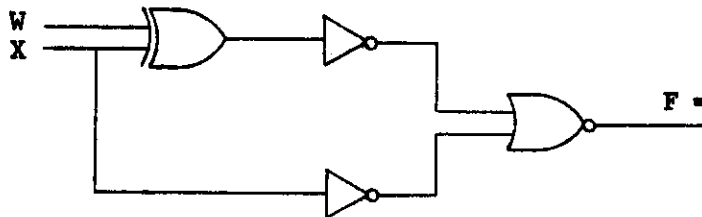
$$= (A\bar{B} + \bar{A}) (A\bar{B} + B) \qquad \text{DE NUEVO LA LEY DISTRIBUTIVA}$$

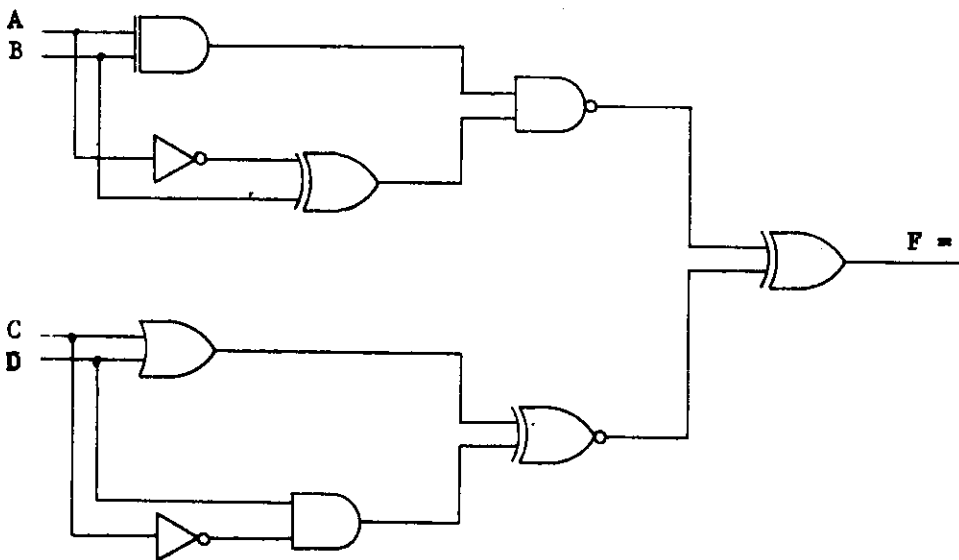
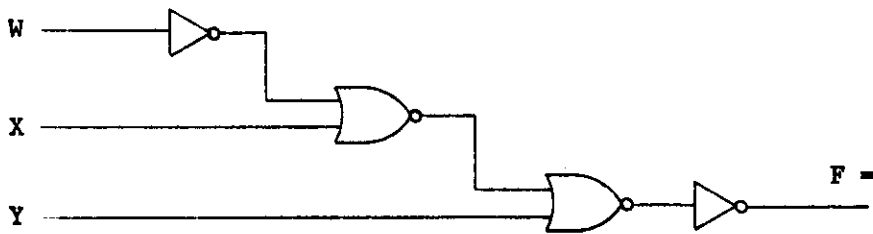
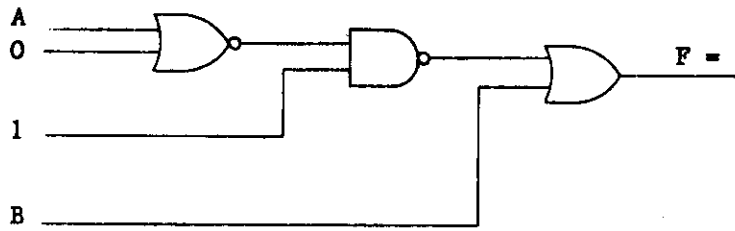
$$= (A + \bar{A}) (\bar{B} + \bar{A}) (A+B) (\bar{B}+B) \qquad a + \bar{a} = 1$$

$$F = (\bar{B} + \bar{A}) (A + B) \qquad \text{FORMA (OR/AND)}$$

PROBLEMAS PROPUESTOS

- 1.-¿Cual es la diferencia entre el Algebra Normal y el Algebra Booleana?
- 2.-¿Cuando es verdadero el resultado de una operación AND?
- 3.-¿Cuántas combinaciones de entrada puede tener una función Si n = al número de variables de entrada?
- 4.-¿Que es una compuerta?
- 5.-¿Cuando es verdadero el resultado de una operación OR?
- 6.-¿Cual es la función de un inversor? Y escriba su símbolo.
- 7.-¿Cuando se cumple una función?
 - a) EX-OR
 - b) NAND
 - c) NOR
- 8.- Explique el funcionamiento de un operador CONCIDENCE.
- 9.-¿Como se complementa una función por medio del Teorema de D'MORGAN?
- 10.-¿A que se le llama minitérmino y cuantos minitérminos tiene una función de n variables de entrada?
- 11.-¿A que se le llama maxitérminos?
- 12.- Encontrar el circuito de las siguientes ecuaciones:
 - a) $F = X \oplus (y \odot Z)$
 - b) $F = \bar{W} + (X + Y)$
 - c) $F = (A.B) + (C+\bar{B}) \cdot (\bar{A}.C) + (B+\bar{C})$
 - d) $F = (\overline{PE} + \overline{PE}) \cdot (\bar{P} + \bar{A}) \oplus (E \odot A)$
- 13.- Encontrar las ecuaciones de los siguientes circuitos:





14.- Implementar un circuito EX-OR y un CONCIDENCE con compuertas

a) NAND

b) NOR

15.- Indique cual de las funciones está expresada en miniterminos.

$$F(ABC) = AC + \bar{A}B + C\bar{B} + ABC$$

$$F(ABC) = B + A\bar{B} + \bar{B}C + \bar{A}BC$$

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

16.- a) Representar la Tabla de Verdad de las siguientes funciones:

$$F_1 = AB + A\bar{B}C + \bar{A}\bar{B}$$

$$F_2 = ABC\bar{D} + A\bar{B}\bar{C} + \bar{A}BD$$

b) Hallar la forma canónica de suma de productos y producto de sumas de las dos funciones del inciso a).

17.- Dada la función F (ABCD) representada mediante la forma canónica -- de suma de productos.

$$F(ABCD) = \sum (0, 1, 2, 3, 12, 15)$$

a) Representar la tabla de verdad de esta función

b) Obtener la forma canónica de producto de sumas

c) Obtener las dos formas canónicas algebraicas de esta función

18.- La función F(ABCD) cumple la siguiente tabla de verdad.

| D | C | B | A | F |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

a) Obtener las formas canónicas de suma de productos y productos de sumas

b) Obtener las formas canónicas de esta función

19.- Una función de tres variables $F(ABC)$ ha de tomar el valor cero — cuando la variable B se encuentre en estado uno y la variable A no esté en estado uno.

- a) Realizar la tabla de verdad de esta función
- b) Obtener las formas canónicas de suma de productos y producto de sumas.

4 *Codigos y Representacion de Informacion*

4.0 INTRODUCCION

En el capítulo 1 vimos como la información y la cantidad se pueden representar por medio de UNOS y CEROS.

Conforme aumenta la complejidad de la información y de los datos -- se hace necesario el uso de CODIGOS que faciliten su representación.

El término CODIGO se usa aquí para designar a un conjunto de símbolos o combinaciones de UNOS y CEROS que sirven para representar información numérica o alfabética.

Los sistemas digitales generalmente representan la información numérica y efectúan sus operaciones internas en CODIGO BINARIO. Sin embargo -- para poder entablar protocolos que interactúen con el mundo exterior se -- recurre al uso de otros códigos.

En la fig. 4.0 se indican los códigos más comunes empleados en la comunicación de un sistema digital con el mundo exterior.

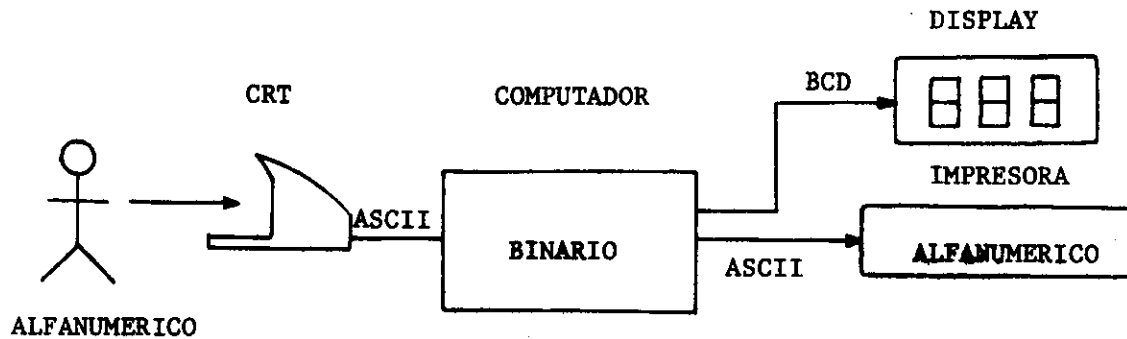


Fig. 4.0 COMUNICACION DE UN SISTEMA DIGITAL CON EL MUNDO EXTERIOR

Antes de continuar daremos las definiciones de algunos términos que usaremos en este capítulo.

BIT Contracción de BINARY-DIGIT - dígito binario

BYTE Grupo de 8 bits.

CARACTER Cualquier letra, número o símbolo que un computador pueda entender, almacenar o procesar.

WORD PALABRA- Grupo de bits utilizados para representar una información. No existe restricción para la cantidad de bits que forman una palabra.

4.1 CODIGOS PESADOS

Muchas veces no es posible usar el código binario para la representación de una cantidad en este caso se utilizan los llamados CODIGOS PESADOS. Se dice que un código es PESADO cuando en correspondencia con la posición de cada bit en una palabra existen valores numéricos, que observan la siguiente propiedad:

LA SUMA DE LOS PRODUCTOS DE CADA BIT POR SU CORRESPONDIENTE VALOR DE POSICION w , ES IGUAL AL VALOR EQUIVALENTE DE LA PALABRA. Esto puede representarse mediante la siguiente expresión:

$$N = \sum_{i=1}^n w_i a_i + C \dots \dots \dots (4.0)$$

Donde N es la cantidad

- n = Número de bits
- w = Peso de cada bit
- a_i = coeficientes
- C = Base constante del código

EJEMPLO 4.0

Determine si el siguiente código es un código pesado.

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

El primer paso es encontrar los valores de w y c para el código y aplicar la ecuación (4.0) a cada combinación. Si la ecuación es válida para todas las combinaciones el código es un código pesado.

De la primera combinación podemos determinar el valor de C aplicando la ecuación (4.0) $N = \sum_{i=1}^n w_i a_i + C$

$$0.w_A + 0.w_B + 0.w_C + 0.w_D + C = 0$$

$$\therefore C = 0$$

Los valores de los pesos w_A , w_B , w_C y w_D se pueden determinar de las siguientes combinaciones.

DE LA SEGUNDA COMBINACION

$$0.w_A + 0.w_B + 0.w_C + 1.w_D + 0 = 1$$

$$\therefore w_D = 1$$

DE LA NOVENA COMBINACION

$$1.w_A + 0(4) + 0(2) + 0(1) + 0 = 8$$

$$\therefore w_A = 8$$

DE LA TERCERA COMBINACION

$$0.w_A + 0.w_B + 1.w_C + 0.(1) + 0 = 2$$

$$\therefore w_C = 2$$

DE LA QUINTA COMBINACION

$$0.w_A + 1.w_B + 0.(2) + 0(1) + 0 = 4$$

$$\therefore w_B = 4$$

Los pesos de este código son 8, 4, 2, 1,, el siguiente paso es aplicar estos pesos en todas las demás combinaciones por ejemplo:

ABCD

$$1100 = 1(8) + 1(4) + 0(2) + 0(1) = 12.$$

$$1111 = 1(8) + 1(4) + 1(2) + 1(1) = 15.$$

En este caso todos los códigos coinciden, podemos decir que se trata de un código pesado.

EMPLO 4.1

Determine si el siguiente código es un código pesado.

| ABCD | CANTIDAD |
|------|----------|
| 0000 | 0 |
| 0001 | 1 |
| 0010 | 2 |
| 0011 | 3 |
| 0100 | 4 |
| 1011 | 5 |
| 1100 | 6 |
| 1101 | 7 |
| 1110 | 8 |
| 1111 | 9 |

PASO 1

Determinar w_A , w_B , w_C , w_D y C .

$$0 \cdot w_A + 0 \cdot w_B + 0 \cdot w_C + 0 \cdot w_D + C = 0$$

$$\therefore C = 0$$

$$0 \cdot w_A + 0 \cdot w_B + 0 \cdot w_C + 1(w_D) + 0 = 1$$

$$C = 0$$

$$\therefore w_D = 1$$

$$w_A = 2$$

$$0 \cdot w_A + 0 \cdot w_B + 1(w_C) + 0(1) + 0 = 2$$

$$w_B = 4$$

$$\therefore w_C = 2$$

$$w_C = 2$$

$$0 \cdot w_A + 1 \cdot (w_B) + 0(2) + 0(1) + 0 = 4$$

$$w_D = 1$$

$$\therefore w_B = 4$$

$$1(w_A) + 0(4) + 1(2) + 1(1) + 0 = 5$$

$$\therefore w_A = 2$$

Aplicamos los valores de los pesos 2, 4, 2, 1 en otras combinaciones.

ABCD

$$1110 = 1(2) + 1(4) + 1(2) + 0(1) = 8$$

$$1111 = 1(2) + 1(4) + 1(2) + 1(1) = 9$$

El código 2, 4, 2, 1 es un código pesado

4.2 CODIGOS NUMERICOS MAS USADOS

En la siguiente tabla se listan algunos de los códigos numéricos de 4 bits más utilizados. Fig. 4.1

| VALOR | BCD 8421 | 2,4,2,1 | EXCESO 3 | GRAY | BINARIO NATURAL 8421 |
|-------|-------------|---------|-------------|------|----------------------------|
| 0 | 0000 | 0000 | 0011 | 0000 | 0000 |
| 1 | 0001 | 0001 | 0100 | 0001 | 0001 |
| 2 | 0010 | 0010 | 0101 | 0011 | 0010 |
| 3 | 0011 | 0011 | 0110 | 0010 | 0011 |
| 4 | 0100 | 0100 | 0111 | 0110 | 0100 |
| 5 | 0101 | 1011 | 1000 | 0111 | 0101 |
| 6 | 0110 | 1100 | 1001 | 0101 | 0110 |
| 7 | 0111 | 1101 | 1010 | 0100 | 0111 |
| 8 | 1000 | 1110 | 1011 | 1100 | 1000 |
| 9 | 1001 | 1111 | 1100 | 1101 | 1001 |
| 10 | | | | 1111 | 1010 |
| 11 | | | | 1110 | 1011 |
| 12 | | | | 1010 | 1100 |
| 13 | | | | 1011 | 1101 |
| 14 | | | | 1001 | 1110 |
| 15 | | | | 1000 | 1111 |

Fig. 4.1 CODIGOS NUMERICOS MAS USADOS.

CODIGO BCD

El código BCD cuyas siglas tienen su origen del nombre en inglés (Binary, Coded, Decimal) DECIMAL CODIFICADO EN BINARIO, es precisamente eso -- un número decimal del 0 al 9 representado en 4 bits. Los números del 10 al 15 no se incluyen este código.

Es importante notar que un número codificado en BCD no es lo mismo -- que un número codificado en binario natural como se puede observar en la fig. 4.1.

Para expresar un número de 2 dígitos decimales en BCD es necesario -- usar 2 DECADAS de BCD como se muestra en el ejemplo 4.2.

EJEMPLO 4.2

Represente en BCD el número 10_{10} .



El código BCD se usa en dispositivos digitales en donde los datos de entrada se generan en un teclado decimal y las salidas se muestran en una pantalla numérica. Por ejemplo en calculadoras digitales, relojes, multímetros, contadores de frecuencia, etc.

Las computadoras digitales modernas no procesan en BCD por dos motivos: El primero es que para representar un número en BCD se requieren más bits que un número representado en binario natural.

Y el segundo motivo es que las operaciones aritméticas son más complicadas que en binario. Imaginemos una suma de $0110 + 0111$, $6 + 7$.

$$\begin{array}{r} 0110 \longrightarrow 6 \\ + 0111 \longrightarrow + 7 \\ \hline 1101 \qquad \qquad 13 \end{array}$$

El número 1101 no existe en BCD, por lo tanto es necesario una operación extra para corregir el resultado, un método simple es sumarle 6 -0110 que es el número de combinaciones que no existen en BCD.

Entonces:

$$\begin{array}{r}
 + \quad 0110 \\
 \hline
 \quad 0111 \\
 + \quad 1101 \longrightarrow \text{SUMA BINARIA} \\
 \hline
 \quad 0110 \\
 1 \quad 0011 \longrightarrow \text{CORRECCION A BCD}
 \end{array}$$

CODIGO 2421

El código 2421 es un código BCD que tiene un paso diferente al usual. En vez de que la posición del bit de mayor peso MSB tenga un peso de 8, como sucede en el BCD 8421, tiene un peso de 2.

EXCESO-3

Es otro código BCD común, a menudo se abrevia como XS3. Este código representa a un número decimal en 4 bits, solo que se le añade 3 a cada dígito decimal antes de efectuar la conversión, por ejemplo el cero se -- encodifica en EXCESO-3 como 0011. Este código tiene propiedades aritméticas útiles, para encontrar el 9 complemento de un número solo se cambian los UNOS por CEROS y viceversa. El método del 9 complemento sirve para hacer restas base 10 y es semejante al método del 2 complemento. En la fig. 4.1 aparece el código XS3.

4.3 CODIGOS NO PESADOS-CODIGO GRAY

En la tabla de la fig. 4.1 aparece el código GRAY. En este código existe solo un cambio de un bit entre dos números sucesivos. Los códigos que tienen esta característica generalmente son CODIGOS NO PESADOS y su aplicación se extiende en los campos de la instrumentación, transductores, -- convertidores analógico/digital, encodificadores de desplazamiento lineal y angular, etc.

En la fig. 4.2 se muestra parcialmente el disco de un encodificador -- de posición angular. Cada uno de los 4 anillos concéntricos representa -

un peso binario y las partes oscuras y blancas representan CEROS y UNOS - respectivamente. Sobre el disco se hayan colocados radialmente 4 transductores mecánicos u ópticos que detectan cada combinación binaria correspondiente a una posición del 0 al 15.

Supongamos que el detector está leyendo el número 8 (1000_2) y la posición que sigue según el movimiento del disco es la 7 (0111_2). Por más delgada que pueda ser la zona sensora del detector al pasar del 1000 al 0111 detectará un 1111, que para este caso es precisamente el número del extremo opuesto del disco.

La decisión de usar el código gray en vez del binario es la mejor solución a el problema de la ambigüedad de lectura en un encodificador óptico. En la fig. 4.3 se muestra parcialmente un disco codificado en código GRAY.

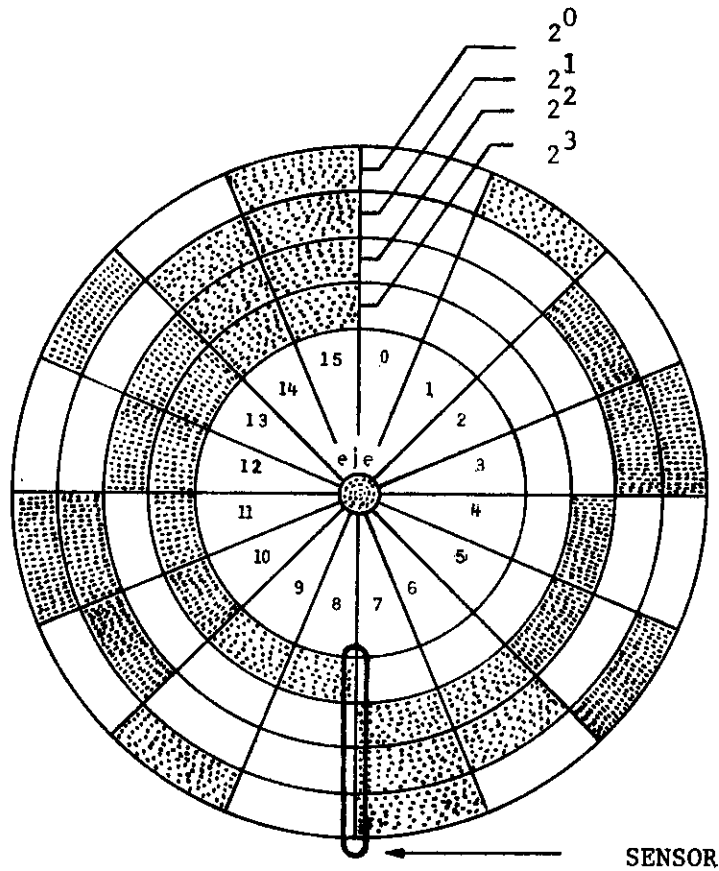


Fig. 4.2 ENCODIFICADOR DE DESPLAZAMIENTO ANGULAR CODIFICADO EN BINARIO.

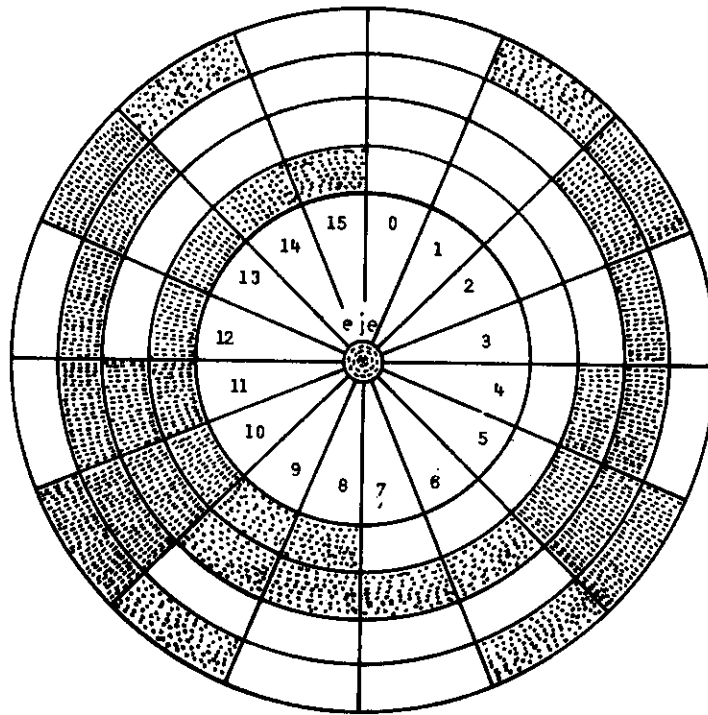


Fig. 4.3 ENCODIFICADOR DE DESPLAZAMIENTO ANGULAR CODIFICADO EN GRAY.

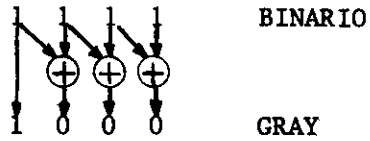
CONVERSION DE CODIGO GRAY \longleftrightarrow BINARIO

Pasos para la conversión de binario a código gray.

- 1- El bit de mayor peso del código GRAY es el mismo que el de código binario.
- 2- El segundo bit del código GRAY es igual a la operación EX-OR del primer y segundo bits binarios.
- 3- El tercer bit del código GRAY es igual al EX-OR del segundo y tercer bits del número binario y así sucesivamente.

EJEMPLO 4.3

Convierta el número binario 1111_2 a código GRAY

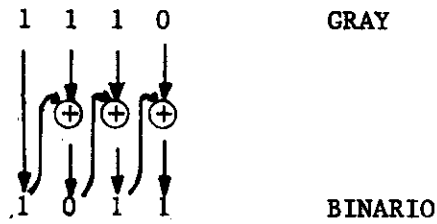


Pasos para la conversión del código GRAY a BINARIO.

- 1- El bit de mayor peso BINARIO es el mismo que el de código GRAY.
- 2- El segundo bit de código binario es igual a la operación EX-OR del primer bit de código binario y el siguiente GRAY y así sucesivamente.

EJEMPLO 4.4

Convierta el número Gray 1110 a Binario



4.4 CODIGOS ALFANUMERICOS

Un computador utiliza tanto datos alfabéticos, como caracteres especiales, tales como signos de puntuación y símbolos matemáticos. A los códigos que representan letras, caracteres y números se les llama CODIGOS ALFANUMERICOS. Generalmente estos códigos tienen un promedio de 64 caracteres, para representarlos en forma binaria se necesitan 6 bits, $2^6 = 64$.

CODIGO EBCDIC

El código EBCDIC (Extended-Binary-Coded-Decimal-Interchange-Code) Código de Intercambio Decimal Codificado En Binario Extendido, puede representar hasta 256 diferentes caracteres. Todos los caracteres están representados por 8 bits o dos números hexadecimales. Este código permite el uso de letras mayúsculas y minúsculas así como caracteres especiales y de control tales como NULL y PF. Estos caracteres de control los interpretan -- los dispositivos periféricos como las impresoras y terminales de video. Muchas combinaciones no tienen asignado un caracter. En la fig. 4.4 aparece la tabulación del código EBCDIC.

| EBCDIC | Bit Con- figuration | Hex | EBCDIC | Bit Con- figuration | Hex | EBCDIC | Bit Con- figuration | Hex | EBCDIC | Bit Con- figuration | Hex |
|--------|------------------------|-----|-----------|------------------------|-----|--------|------------------------|-----|--------|------------------------|-----|
| NULL | 0000 0000 | | b (blank) | 0100 0000 | 40 | | 1000 0000 | . | A | 1100 0000 | C0 |
| | 0000 0001 | | | 0100 0001 | | a | 1000 0001 | 81 | B | 1100 0001 | C1 |
| | 0000 0010 | | | 0100 0010 | | b | 1000 0010 | 82 | C | 1100 0010 | C2 |
| | 0000 0011 | | | 0100 0011 | | c | 1000 0011 | 83 | D | 1100 0011 | C3 |
| PF | 0000 0100 | 04 | | 0100 0100 | | d | 1000 0100 | 84 | E | 1100 0100 | C4 |
| HT | 0000 0101 | 05 | | 0100 0101 | | e | 1000 0101 | 85 | F | 1100 0101 | C5 |
| LC | 0000 0110 | 06 | | 0100 0110 | | f | 1000 0110 | 86 | G | 1100 0110 | C6 |
| DEL | 0000 0111 | 07 | | 0100 0111 | | g | 1000 0111 | 87 | H | 1100 0111 | C7 |
| | 0000 1000 | | C | 0100 1000 | | h | 1000 1000 | 88 | I | 1100 1000 | C8 |
| | 0000 1001 | | | 0100 1001 | 49 | i | 1000 1001 | 89 | | 1100 1001 | C9 |
| | 0000 1010 | | ↓ | 0100 1010 | 4A | | 1000 1010 | | | 1100 1010 | |
| | 0000 1011 | | | 0100 1011 | 4B | | 1000 1011 | | | 1100 1011 | |
| | 0000 1100 | | < | 0100 1100 | 4C | | 1000 1100 | | | 1100 1100 | |
| | 0000 1101 | | ! | 0100 1101 | 4D | | 1000 1101 | | | 1100 1101 | |
| | 0000 1110 | | + | 0100 1110 | 4E | | 1000 1110 | | | 1100 1110 | |
| | 0000 1111 | | ! | 0100 1111 | 4F | | 1000 1111 | | | 1100 1111 | |
| | 0001 0000 | | & | 0101 0000 | | | 1001 0000 | | J | 1101 0000 | D0 |
| | 0001 0001 | | | 0101 0001 | | i | 1001 0001 | 91 | K | 1101 0001 | D1 |
| | 0001 0010 | | | 0101 0010 | | k | 1001 0010 | 92 | L | 1101 0010 | D2 |
| | 0001 0011 | | | 0101 0011 | | l | 1001 0011 | 93 | M | 1101 0011 | D3 |
| RES | 0001 0100 | 14 | | 0101 0100 | | m | 1001 0100 | 94 | N | 1101 0100 | D4 |
| NL | 0001 0101 | 15 | | 0101 0101 | | n | 1001 0101 | 95 | O | 1101 0101 | D5 |
| BS | 0001 0110 | 16 | | 0101 0110 | | o | 1001 0110 | 96 | P | 1101 0110 | D6 |
| IDL | 0001 0111 | 17 | | 0101 0111 | | p | 1001 0111 | 97 | Q | 1101 0111 | D7 |
| | 0001 1000 | | | 0101 1000 | | q | 1001 1000 | 98 | R | 1101 1000 | D8 |
| | 0001 1001 | | | 0101 1001 | | r | 1001 1001 | 99 | | 1101 1001 | D9 |
| | 0001 1010 | | ! | 0101 1010 | 5A | | 1001 1010 | | | 1101 1010 | |
| | 0001 1011 | | \$ | 0101 1011 | 5B | | 1001 1011 | | | 1101 1011 | |
| | 0001 1100 | | * | 0101 1100 | 5C | | 1001 1100 | | | 1101 1100 | |
| | 0001 1101 | | ! | 0101 1101 | 5D | | 1001 1101 | | | 1101 1101 | |
| | 0001 1110 | | : | 0101 1110 | 5E | | 1001 1110 | | | 1101 1110 | |
| | 0001 1111 | | ? | 0101 1111 | 5F | | 1001 1111 | | | 1101 1111 | |
| | 0010 0000 | | / | 0110 0000 | 60 | | 1010 0000 | | S | 1110 0000 | E0 |
| | 0010 0001 | | | 0110 0001 | 61 | s | 1010 0001 | | T | 1110 0001 | E1 |
| | 0010 0010 | | | 0110 0010 | | t | 1010 0010 | A2 | U | 1110 0010 | E2 |
| | 0010 0011 | | | 0110 0011 | | u | 1010 0011 | A3 | V | 1110 0011 | E3 |
| BYP | 0010 0100 | 24 | | 0110 0100 | | v | 1010 0100 | A4 | W | 1110 0100 | E4 |
| LF | 0010 0101 | 25 | | 0110 0101 | | w | 1010 0101 | A5 | X | 1110 0101 | E5 |
| EOB | 0010 0110 | 26 | | 0110 0110 | | x | 1010 0110 | A6 | Y | 1110 0110 | E6 |
| PRE | 0010 0111 | 27 | | 0110 0111 | | y | 1010 0111 | A7 | Z | 1110 0111 | E7 |
| | 0010 1000 | | | 0110 1000 | | z | 1010 1000 | A8 | | 1110 1000 | E8 |
| | 0010 1001 | | | 0110 1001 | | | 1010 1001 | A9 | | 1110 1001 | E9 |
| | 0010 1010 | | | 0110 1010 | | | 1010 1010 | | | 1110 1010 | |
| | 0010 1011 | | . | 0110 1011 | 6B | | 1010 1011 | | | 1110 1011 | |
| | 0010 1100 | | % | 0110 1100 | 6C | | 1010 1100 | | | 1110 1100 | |
| | 0010 1101 | | - | 0110 1101 | 6D | | 1010 1101 | | | 1110 1101 | |
| | 0010 1110 | | ~ | 0110 1110 | 6E | | 1010 1110 | | | 1110 1110 | |
| | 0010 1111 | | ? | 0110 1111 | 6F | | 1010 1111 | | | 1110 1111 | |
| | 0011 0000 | | | 0111 0000 | | | 1011 0000 | | 0 | 1111 0000 | F0 |
| | 0011 0001 | | | 0111 0001 | | | 1011 0001 | | 1 | 1111 0001 | F1 |
| | 0011 0010 | | | 0111 0010 | | | 1011 0010 | | 2 | 1111 0010 | F2 |
| | 0011 0011 | | | 0111 0011 | | | 1011 0011 | | 3 | 1111 0011 | F3 |
| PN | 0011 0100 | 34 | | 0111 0100 | | | 1011 0100 | | 4 | 1111 0100 | F4 |
| RS | 0011 0101 | 35 | | 0111 0101 | | | 1011 0101 | | 5 | 1111 0101 | F5 |
| UC | 0011 0110 | 36 | | 0111 0110 | | | 1011 0110 | | 6 | 1111 0110 | F6 |
| EOT | 0011 0111 | 37 | | 0111 0111 | | | 1011 0111 | | 7 | 1111 0111 | F7 |
| | 0011 1000 | | | 0111 1000 | | | 1011 1000 | | 8 | 1111 1000 | F8 |
| | 0011 1001 | | | 0111 1001 | 79 | | 1011 1001 | | 9 | 1111 1001 | F9 |
| | 0011 1010 | | | 0111 1010 | 7A | | 1011 1010 | | | 1111 1010 | |
| | 0011 1011 | | # | 0111 1011 | 7B | | 1011 1011 | | | 1111 1011 | |
| | 0011 1100 | | @ | 0111 1100 | 7C | | 1011 1100 | | | 1111 1100 | |
| | 0011 1101 | | * | 0111 1101 | 7D | | 1011 1101 | | | 1111 1101 | |
| | 0011 1110 | | = | 0111 1110 | 7E | | 1011 1110 | | | 1111 1110 | |
| | 0011 1111 | | " | 0111 1111 | 7F | | 1011 1111 | | | 1111 1111 | |

Fig. 4.4 CODIGO (EBCDIC)

CODIGO ASCII

En un esfuerzo por estandarizar los códigos de intercambio de información los fabricantes de equipo relacionado a esta rama acordaron usar el código ASCII, siglas del inglés (American Standard Code for Information In-

terchange). Este código puede representar hasta 128 caracteres diferentes y usa 7 bits. El listado está dividido en zonas, por ejemplo la zona 011 (de los bit de mayor peso) contiene todos los caracteres numéricos más 6 caracteres especiales así el número 0 es un 30 HEX o 011 0000.

La letra A es un 41 HEX o un 100 0001, este código también incluye los caracteres de control. En la fig. 4.5 aparece el listado del código ASCII y el significado de las abreviaciones para los caracteres de control.

NOTA. No se acostumbra usar traducción para estos términos

| | CONTROL | | CARACTERES ALFANUMERICOS | | | | | | |
|-----------------------|---------|-----|--------------------------|-----|-----|-----|-----|-----|--|
| 765 ↓ 4321 → | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 | |
| 0000 | NUL | DLE | SP | 0 | @ | P | ' | p | |
| 0001 | SOH | DC1 | ! | 1 | A | Q | a | q | |
| 0010 | STX | DC2 | " | 2 | B | R | b | r | |
| 0011 | ETX | DC3 | # | 3 | C | S | c | s | |
| 0100 | EOT | DC4 | \$ | 4 | D | T | d | t | |
| 0101 | ENQ | NAK | % | 5 | E | U | e | u | |
| 0110 | ACK | SYN | & | 6 | F | V | f | v | |
| 0111 | BEEP | ETB | ' | 7 | G | W | g | w | |
| 1000 | BS | CAN | (| 8 | H | X | h | x | |
| 1001 | HT | EM |) | 9 | I | Y | i | y | |
| 1010 | LF | SUB | * | : | J | Z | j | z | |
| 1011 | VT | ESC | + | ; | K | [| k | { | |
| 1100 | FF | FS | , | < | L | \ | l | : | |
| 1101 | CR | GS | - | = | M |] | m | } | |
| 1110 | SO | RS | . | > | N | ^ | n | ~ | |
| 1111 | SI | US | / | ? | O | _ | o | DEL | |

Fig. 4.5 LISTADO DEL CODIGO ASCII

| | | | |
|-------|-----------------------|----------------------------------|---|
| NULL | Null Idle | CR | Carriage return |
| SOM | Start of message | SO | Shift out |
| EOA | End of address | SI | Shift in |
| EOM | End of message | DC ₀ | Device control ① Reserved for data link escape |
| EOT | End of transmission | DC ₁ -DC ₃ | Device control |
| WRU | "Who are you?" | ERR | Error |
| RU | "Are you ...?" | SYNC | Synchronous idle |
| BELL | Audible signal | LEM | Logical end of media |
| FE | Format effector | SO ₀ -SO ₇ | Separator (information) Word separator (blank, normally non-printing) |
| HT | Horizontal tabulation | ACK | Acknowledge |
| SK | Skip (punched card) | ② | Unassigned control |
| LF | Line feed | ESC | Escape |
| V/TAB | Vertical tabulation | DEL | Delete Idle |
| FF | Form feed | | |

Fig. 4.5 LISTADO DEL CODIGO ASCII

CODIGO TTY

El código TTY (Tele Type) a menudo llamado BAUDOT usa solamente 5 bit por carácter. Como resultado algunas palabras del código se emplean para representar más de un carácter. El código TTY ha sido extendido para representar 32 caracteres diferentes usando el carácter especial MODE-CHANGE.

El transmisor y el receptor que manejan este tipo de código deben comenzar con el mismo modo, generalmente el modo alfabético. Los cambios en el modo se introducen en la secuencia de las palabras del código siempre que sea necesario.

El número efectivo de bits por carácter se incrementa por encima de 5, desde que los caracteres de Mode-Change adicionan los bits al dato que se está transmitiendo o almacenando y que sin embargo no llevan información. Una variedad del código de 5 bits, usa dos caracteres de Mode-Change, uno para hacer la transferencia a un modo alfabético y otro para la transfe--

rencia al modo numérico. En la fig. 4.6 aparece el listado para el código TTY.

| PALABRA | MODO ALFABETICO | MODO NUMERICO |
|---------|-----------------|---------------|
| 00 | Blank | Blank |
| 01 | E | 3 |
| 02 | | |
| 03 | A | - |
| 04 | | |
| 05 | S | |
| 06 | I | 8 |
| 07 | U | 7 |
| 10 | | |
| 11 | D | \$ |
| 12 | R | 4 |
| 13 | J | . |
| 14 | N | |
| 15 | F | |
| 16 | C | |
| 17 | K | (|
| 20 | T | 5 |
| 21 | Z | |
| 22 | L |) |
| 23 | W | 2 |
| 24 | H | |
| 25 | Y | 6 |
| 26 | P | 0 |
| 27 | Q | 1 |
| 30 | O | 9 |
| 31 | B | |
| 32 | G | |
| 33 | | |
| 34 | M | . |
| 35 | X | / |
| 36 | V | |
| 37 | Mode change | Mode change |

Fig. 4.6 CODIGO TTY

4.5 DETECCION DE ERRORES (PARIDAD)

Una de las propiedades de los códigos que hemos discutido en este capítulo es la capacidad que tienen para detectar errores cuando alguna información codificada se transmite de un dispositivo a otro, incluso cuando esa información se almacena en memoria. Los errores consisten en la pérdida o alteración de uno o más bits de una palabra manipulada, o transmitida.

Uno de los formatos más utilizados para la detección de errores es el método de PARIDAD. Este método consiste en agregar a la palabra codificada un bit extra llamado precisamente BIT DE PARIDAD que se usa para determinar si el dato transmitido ha sido alterado durante el proceso de transmisión. El bit de paridad se establece como 0 o 1 dependiendo del número de UNOS que contiene la palabra. Este bit se usa en 2 formas diferentes, una para indicar una PARIDAD PAR y otra para indicar una PARIDAD-IMPAR.

En el método de PARIDAD PAR el bit de paridad se escoge de tal manera que el número total de UNOS en la palabra, incluyendo el bit de paridad sea un número par. Por ejemplo supongamos una C, codificada en ASCII como 100 0011 el grupo tiene 3 UNOS por lo tanto añadiremos un bit de paridad igual a 1 para hacer que el número total de UNOS tenga un valor PAR. Entonces la palabra quedaría como:

100 0011 1
 ↑
 Bit de PARIDAD

Supongamos ahora que deseamos incluir un bit de paridad par en una A codificada en ASCII como 100 0001. El grupo tiene 2 UNOS por lo tanto el bit de paridad par debe ser igual a 0 para hacer que el número total de unos tenga un valor PAR entonces la palabra quedaría:

100 0001 0
 ↑
 Bit de PARIDAD

Los números negativos pueden expresarse en forma afirmada como el -16 de la fig. 4-7, sin embargo muchas computadoras manejan los números negativos en la forma de UNO y DOS COMPLEMENTO. En la fig. 4.8 se muestra un -16 expresado en la forma del UNO y DOS complemento.

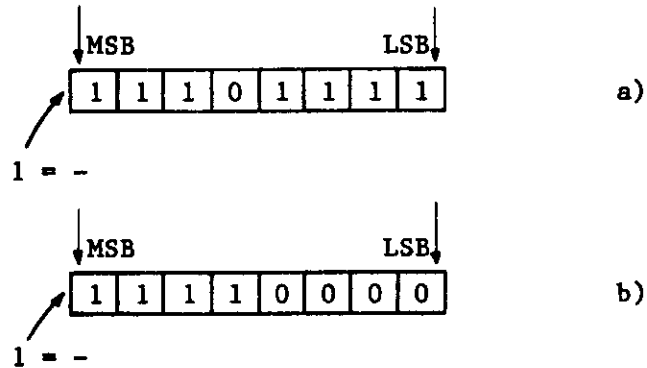


Fig. 4.8 NUMEROS NEGATIVOS EXPRESADOS POR MEDIO
a) UNO COMPLEMENTO, b) DOS COMPLEMENTO..

4.7 SUMAS Y RESTAS CON NUMEROS CON SIGNO

Las reglas para sumar o restar números binarios con signo son las mismas que las que se usan en decimal.

- 1- Para sumar números con el mismo signo, se le da al resultado el mismo signo, es decir, dos números positivos generan una suma positiva, dos números negativos generan una suma negativa.
- 2- Para sumar números con signos diferentes, se obtiene la diferencia entre ambos y el signo del resultado es el del número mayor.
- 3- Para restar números con signos, se cambia el signo del sustraendo y se suma el sustraendo al minuendo de acuerdo con las reglas 1 y 2.

EJEMPLO 4.5

SUMAS

$$\begin{array}{r}
 0000\ 0011 \quad 3 \\
 + \underline{0000\ 1000} \quad +\ 8 \\
 +\ 0000\ 1011 \quad 11
 \end{array}$$

Números Positivos

$$\begin{array}{r}
 1111\ 1101 \quad -\ 3 \\
 \underline{1111\ 1000} \quad -\ 8 \\
 1111\ 0101 \quad -\ 11
 \end{array}$$

Números Negativos en
2 complemento.
Resultado en 2 complemento

$$\begin{array}{r}
 1111\ 1101 \quad -\ 3 \\
 \underline{0000\ 1000} \quad +\ 8 \\
 0000\ 0101 \quad +\ 5
 \end{array}$$

2 Complemento
Afirmado
Afirmado

$$\begin{array}{r}
 0000\ 0011 \quad +\ 3 \\
 \underline{1111\ 1000} \quad -\ 8 \\
 1111\ 1011 \quad -\ 5
 \end{array}$$

Afirmado
2 Complemento
2 Complemento

RESTAS

$$8 - 3 = +5$$

$$\begin{array}{r}
 0000\ 1000 \quad +\ 8 \\
 \underline{1111\ 1101} \quad (-)\ +\ 3 \\
 0000\ 0101 \quad +\ 5
 \end{array}$$

Afirmado
2 Complemento
Afirmado

$$3 - 8 = -5$$

$$\begin{array}{r}
 0000\ 0011 \quad +\ 3 \\
 \underline{1111\ 1000} \quad (-)\ +\ 8 \\
 1111\ 1011 \quad -\ 5
 \end{array}$$

Afirmado
2 Complemento
2 Complemento

Generalmente un computador efectúa la suma y resta con un circuito que únicamente sume, la multiplicación y la división se desarrollan con subrutinas que usan la suma y resta.

CAPITULO 4

PROBLEMAS PROPUESTOS

- 1.-¿Que es Código?
- 2.-¿Que es un Byte y que es un Caracter?
- 3.-¿Cuantos tipos de códigos hay?
- 4.-¿Que es un Código pesado?
- 5.-¿Cual es la diferencia entre un código BCD y un 2421?
- 6.-¿Para que nos puede servir un Código de Exceso-3?
- 7.-¿Que es un Código no pesado?
- 8.-¿Como se convierte de un Código Gray a un Código Binario?
- 9.-¿Como se convierte de un Código Binario a un Código Gray?
- 10.-¿Que son los Códigos Alfanuméricos?
- 11.-¿Cuantos caracteres y cuantos bits representa un Código EBCDIC?
- 12.- Como se representan los siguientes caracteres en Código ASCII.

| | | | |
|------|-------|---|-------|
| Y | _____ | δ | _____ |
| EM | _____ | δ | _____ |
| SP | _____ | δ | _____ |
| a | _____ | δ | _____ |
| NULL | _____ | δ | _____ |
| y | _____ | δ | _____ |
| 0 | _____ | δ | _____ |
| : | _____ | δ | _____ |
| 4 | _____ | δ | _____ |
| ? | _____ | δ | _____ |
| (| _____ | δ | _____ |

- 13.- Como se representan los siguientes caracteres en Código.TTY.

| | |
|----|-------|
| \$ | _____ |
| G | _____ |
| - | _____ |
| D | _____ |
| / | _____ |

14.- ¿Para que nos sirve el Método de Paridad y en que consiste?

15.- Determine si el siguiente Código es un Código pesado.

| ABCD | CANTIDAD |
|------|----------|
| 0000 | 0 |
| 0001 | 1 |
| 0010 | 2 |
| 0011 | 3 |
| 0100 | 4 |
| 1000 | 5 |
| 1001 | 6 |
| 1010 | 7 |
| 1011 | 8 |
| 1100 | 9 |

16.- Convierta los siguientes números binarios al Código Gray.

1100₂ _____
0111₂ _____
1010₂ _____
1000₂ _____
0100₂ _____

17.- Convierta los siguientes números Gray a Binario

1001 _____
1110 _____
0101 _____
0010 _____
1011 _____

18.- Obtener el número decimal equivalente al número 0110 1000 0100 en BCD 8421.

19.- Obtener a partir de el Código 2, 4, 2, 1, un Código de Paridad Par.

20.- Convertir el número 1100 1000 0011 perteneciente al Código BCD --
exceso tres a:

- a) El código BCD 8421
- b) El código BCD 2,4,2,1
- c) El sistema binario natural
- d) El sistema decimal

21.- Efectúe las sig. operaciones de números con signo.

$$\begin{array}{r} 0000\ 1101 \\ + \underline{0000\ 0101} \end{array} \qquad \begin{array}{r} 0011\ 0000 \\ - \underline{0100\ 1111} \end{array}$$

$$\begin{array}{r} 1110\ 0011 \\ + \underline{1001\ 1110} \end{array} \qquad \begin{array}{r} 1001\ 1111 \\ - \underline{1111\ 1111} \end{array}$$

$$\begin{array}{r} 0100\ 0011 \\ + \underline{1111\ 1101} \end{array} \qquad \begin{array}{r} 1110\ 1010 \\ - \underline{1111\ 1000} \end{array}$$

5 *Minimización de Funciones Booleanas*

5.0 INTRODUCCION

En el capítulo tres observamos como, a partir de una tabla de verdad, se puede obtener la expresión booleana, que representa el comportamiento de un bloque digital. Esta expresión no siempre está en su forma más simple.

Ejemplo 5.0

Obtenga la función del bloque digital cuya tabla de verdad se muestra a continuación.

| AB | F(AB) |
|----|-------|
| 00 | 0 |
| 01 | 1 |
| 10 | 0 |
| 11 | 1 |

$$F(AB) = \bar{A}B + AB$$

La expresión $F(AB) = \bar{A}B + AB$ no está en su forma más reducida. Por simple inspección podemos notar que los valores de $F(AB)$ son iguales a los valores de la variable B.

En ambos minitérminos B permanece constante, si la tomamos como factor común y aplicamos después la propiedad del álgebra booleana que dice $\bar{a} + a = 1$, tenemos:

$$F(AB) = \bar{A}B + AB$$

$$F(AB) = B(\bar{A} + A)$$

$$F(AB) = B$$

En este capítulo discutiremos las técnicas de minimización que nos lleven a reducir el número de los componentes necesarios para implementar una función booleana. Esta minimización es importante debido a que la cantidad de elementos impacta en el costo, complejidad, y mantenimiento de un circuito digital.

5.1 CRITERIO DE COSTO

Cuando se implementaban las compuertas lógicas por medio de elementos discretos como, diodos, resistencias y transistores, el COSTO estaba relacionado con la cantidad de entradas a las compuertas que formaban un circuito.

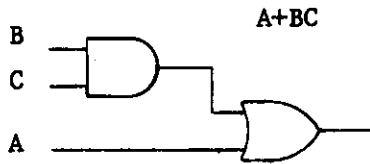
EL COSTO UNITARIO C.U. = # DE ENTRADAS A LAS COMPUERTAS DEL CIRCUITO

En la actualidad el costo de un CIRCUITO INTEGRADO no depende tanto del bloque que se encuentra en su interior, sino en gran parte del número de entradas y salidas, que determinan la cantidad de terminales en el chip. Por lo tanto el costo de un bloque digital depende del costo de los circuitos integrados empleados para su implementación. Sin embargo es recomendable comparar el C.U. con el costo total de los circuitos integrados.

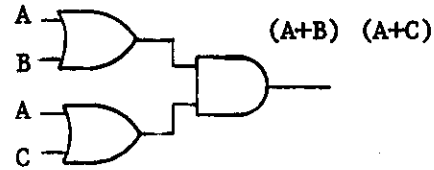
Ejemplo 5.1

De la siguiente igualdad. Compare las dos expresiones en términos del costo unitario necesario para implementar ambas partes.

$$A+BC = (A+B) (A+C)$$



$$CU = 4$$



$$CU = 6$$

5.2 MANIPULACION ALGEBRAICA

En este punto aplicaremos las propiedades y las leyes del álgebra booleana para la simplificación de expresiones booleanas llamadas también funciones de interrupción.

Existen básicamente cuatro métodos de minimización algebraica que consisten en 1, Factorizar términos para lograr las formas $a + \bar{a} = 1$ y $a + 1 = 1$, 2, Duplicado de un término ya existente, 3, Multiplicar por un término del tipo $(a + \bar{a})$ y 4, Aplicar la Ley Distributiva.

5.2.1 FACTORIZACION

Cuando una expresión booleana en la forma de sumatoria de productos contiene dos minitérminos que difieren solo en una variable, esta puede eliminarse factorizando los términos comunes.

Ejemplo 5.2

Simplifique la función $F(ABC) = ABC + AB\bar{C}$

$$F(ABC) = ABC + AB\bar{C}$$

$$F(ABC) = AB (C + \bar{C}) \quad \text{Dado} \quad C + \bar{C} = 1$$

$$F(ABC) = AB$$

La expresión $A(B + \bar{B}) = A$, tiene su equivalente en la forma de maxi términos.

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

Ejemplo 5.3

Simplifique la siguiente función:

$$\begin{aligned} F(ABC) &= (A+B+C) (A+B+\bar{C}) \\ F(ABC) &= \left[(A+B)+C \right] \left[(A+B)+\bar{C} \right] && (A+B) (A+\bar{B}) = A \\ F(ABC) &= A+B \end{aligned}$$

Al usar el método de factorización pueden aparecer minitérminos que compartan una misma literal y que tomen la forma $a + 1 = 1$.

Ejemplo 5.4

Simplifique la siguiente expresión:

$$\begin{aligned} F(ABC) &= A + AB + A\bar{C} + A\bar{B}C \\ F(ABC) &= A (1 + B + \bar{C} + \bar{B}C) && \text{Dado } a + 1 = 1 \\ F(ABC) &= A \end{aligned}$$

5.2.2 DUPLICANDO UN TERMINO YA EXISTENTE

La propiedad del álgebra booleana que dice $a = a + a + \dots + a$, puede usarse en la simplificación de funciones booleanas. Un término que a parece en una suma de minitérminos o un producto de sumatorias puede duplicarse tantas veces como sea necesario, para su combinación con otros términos.

Ejemplo 5.5

Minimizar la siguiente expresión duplicando un término

$$\begin{aligned} F &= A\bar{B}\bar{C} + ABC + A\bar{B}C \\ F &= A\bar{B}\bar{C} + ABC + ABC + A\bar{B}C && \text{Duplicando } ABC \\ F &= AB (\bar{C}+C) + AC (B+\bar{B}) && \text{Dado } a + \bar{a} = 1 \\ F &= AB + AC \end{aligned}$$

5.2.3 MULTIPLICANDO POR UN TERMINO DEL TIPO (a + ā)

En ciertas ocasiones se presentan funciones tales como:

$$F = XY + \bar{X}Z + YZ$$

Aquí no se visualiza una posible simplificación por factorización. -
En tal caso podemos multiplicar el término YZ por $(\bar{X}+X)$ donde $(\bar{X}+X) = 1$.

Ejemplo 5.6

$$F = XY + \bar{X}Z + YZ$$

$$F = XY + \bar{X}Z + (X+\bar{X})YZ \quad \text{Dado} \quad \text{multiplicando por } (\bar{X} + X)$$

$$F = XY + \bar{X}Z + XYZ + \bar{X}YZ$$

$$F = XY + XYZ + \bar{X}Z + \bar{X}YZ$$

$$F = XY(1+Z) + \bar{X}Z(1+Y)$$

$$F = XY + \bar{X}Z$$

5.2.4 APLICANDO LA LEY DISTRIBUTIVA

La ley Distributiva $a + bc = (a+b)(a+c)$, se presenta como una posible solución para expresiones de la forma, $a + \bar{a}b$.

Ejemplo 5.7

Simplifique la expresión $F = A + \bar{A}B$

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

$$F = (A + \bar{A})(A + B) \quad a + \bar{a}b \text{ Ley Distributiva}$$

$$F = 1(A+B) \quad \text{Dado} \quad a + \bar{a} = 1$$

$$F = A + B$$

5.3 MAPAS DE KARNAUGH

El mapa de Karnaugh es un método gráfico para la representación y -
minimización de funciones booleanas. Se usa para simplificar funciones-
de 2, 3 y 4 variables, pero puede extenderse satisfactoriamente a funcio-
nes de 5 y 6 variables.

Su operación se basa en la combinación de minitérminos los cuales -
difieren en solo una variable, como $A\bar{B} + AB = A(\bar{B} + B) = A$

Un mapa para una función de N variables consiste de 2^N cuadros. Donde cada cuadro representa a un minitérmino, además entre los minitérminos de cuadros adyacentes debe haber un solo cambio en una de sus variables.

Un mapa para una función de 2 variables tiene $2^2 = 4$ cuadros, para 3 variables $2^3 = 8$ cuadros, para 4 variables $2^4 = 16$ cuadros, y así sucesivamente. El mapa de Karnaugh para una función de dos variables se muestra en la fig. 5.1

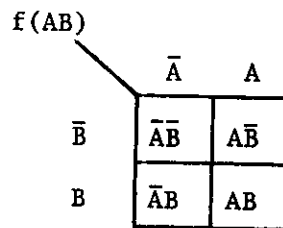


Fig. 5.1 MAPA DE KARNAUGH PARA UNA FUNCION DE DOS VARIABLES

La función se encuentra graficada en una cuadrícula donde las coordenadas son A y B . En el eje horizontal la mitad derecha del mapa corresponde a la variable afirmada A y la izquierda a su complemento \bar{A} . Lo mismo sucede con la variable B graficada en el eje vertical. Generalmente se acostumbra marcar la zona para cada variable con su etiqueta correspondiente fig. 5.1

Si se desea graficar la expresión $\bar{A}\bar{B}$ en el mapa se indica escribiendo un 1 en el cuadro donde las variables \bar{A} y \bar{B} son comunes, como se muestra en la fig. 5.2 en general cada cuadro impreso representa un término formado por el producto de las variables comunes al cuadro.

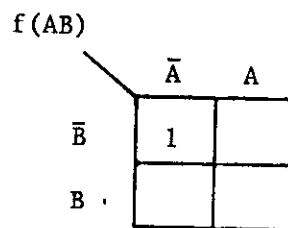


Fig. 5.2 GRAFICA EN UN MAPA PARA LA EXPRESION $\bar{A}\bar{B}$.

Para simplificar el acomodo de las etiquetas correspondientes a cada zona se indican las variables alfabéticas en la parte superior izquierda del mapa, para el caso de una función de dos variables, A se grafica en el eje horizontal y B en el eje vertical. Por último las zonas se marcan con un número 0 o 1 como se muestra en la fig. 5.3

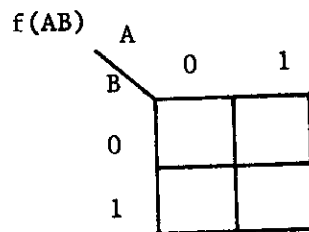


Fig. 5.3 PARA INDICAR CADA ZONA, LAS VARIABLES PUEDEN SUSTITUIRSE POR NUMEROS.

Un mapa para una función de tres variables se muestra en la fig. 5.4 se puede observar que existe únicamente una variable modificada entre dos cuadros adyacentes.

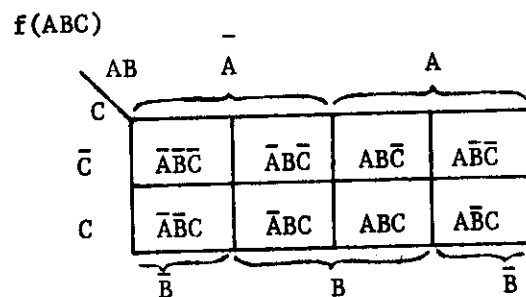


Fig. 5.4 MAPA DE KARNAUGH PARA UNA FUNCION DE 3 VARIABLES INDICANDO LA ZONA CORRESPONDIENTE A CADA VARIABLE.

En la fig. 5.5 aparece el mapa de Karnaugh con la distribución acostumbrada. En el eje horizontal se grafican simultáneamente las variables A y B por este motivo la etiqueta que aparece en la parte superior de cada columna es de dos dígitos y dan las combinaciones 00, 01, 11, y 10.

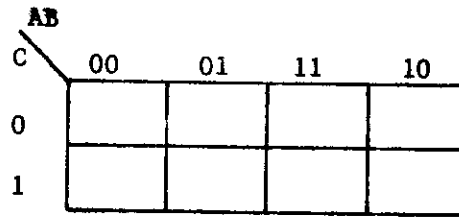


Fig. 5.5 MAPA DE KARNAUGH PARA UNA FUNCION DE 3 VARIABLES

Para graficar una expresión de cuatro variables tenemos que utilizar un mapa de $2^4 = 16$ cuadros. En el eje horizontal se colocan las variables A y B, y en el eje vertical las variables C y D. En la fig. 5.6 se muestra un mapa de Karnaugh para esta función, indicando la zona correspondiente a cada variable y sus etiquetas numéricas.

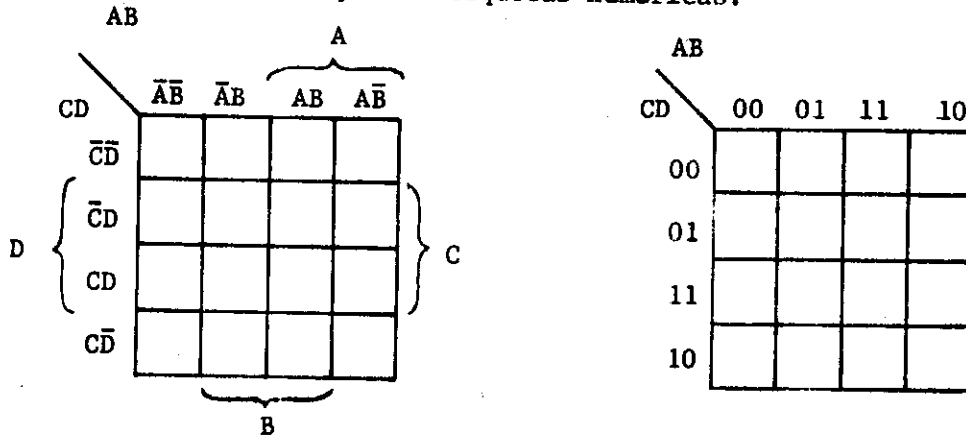


Fig. 5.6 MAPAS DE KARNAUGH PARA UNA FUNCION DE 4 VARIABLES

5.3.1 REDUCCION DE EXPRESIONES BOOLEANAS USANDO EL MAPA DE KARNAUGH

La utilidad del mapa de Karnaugh se basa en que el acomodo de las áreas para cada variable, permite minimizar una expresión lógica por simple inspección. Veamos que relación existe entre un mapa de Karnaugh y una tabla de verdad.

En la fig. 5.7 se muestra una tabla de verdad para una función de 2 variables y el acomodo para cada minitérmino de la función en el mapa.

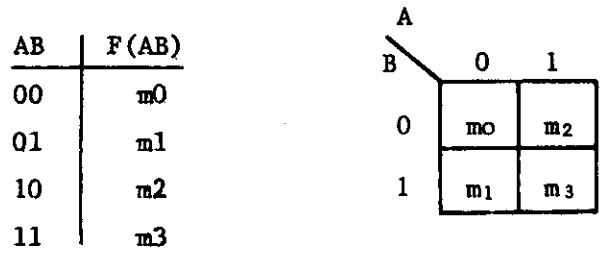


Fig. 5.7 TABLA DE VERDAD PARA UNA FUNCION DE 2 VARIABLES Y MAPA DE KARNAUGH CONTENIENDO LOS VALORES DE LA TABLA.

A cada cuadro se le asigna un número en decimal que corresponde al nombre de cada minitérmino. Generalmente se escriben estos números en la parte superior derecha del cuadro para facilitar la transferencia de los datos de la tabla, ver fig. 5.8

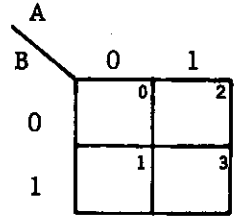


Fig. 5.8 A CADA CUADRO SE LE ASIGNA UN NUMERO DECIMAL CORRESPONDIENTE AL MINITERMINO DE ESE CUADRO.

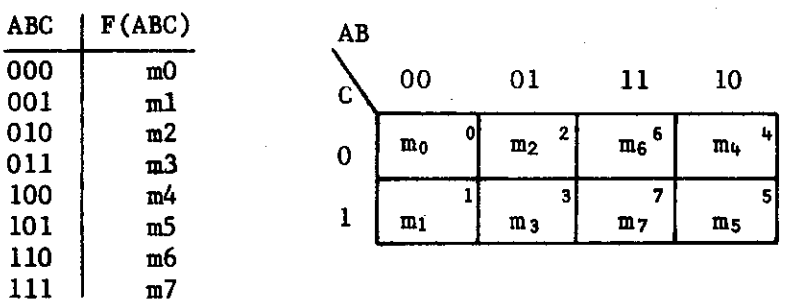


Fig. 5.9 DISTRIBUCION DE LOS MINITERMINOS PARA UN MAPA DE KARNAUGH DE 3 VARIABLES.

| ABCD | F |
|------|-----|
| 0000 | m0 |
| 0001 | m1 |
| 0010 | m2 |
| 0011 | m3 |
| 0100 | m4 |
| 0101 | m5 |
| 0110 | m6 |
| 0111 | m7 |
| 1000 | m8 |
| 1001 | m9 |
| 1010 | m10 |
| 1011 | m11 |
| 1100 | m12 |
| 1101 | m13 |
| 1110 | m14 |
| 1111 | m15 |

| | | AB | | | |
|----|----|----|----|----|----|
| | | 00 | 01 | 11 | 10 |
| CD | 00 | 0 | 4 | 12 | 8 |
| | 01 | 1 | 5 | 13 | 9 |
| | 11 | 3 | 7 | 15 | 11 |
| | 10 | 2 | 6 | 14 | 10 |

Fig. 5.10 ACOMODO PARA LOS MINITERMINOS DE UN MAPA DE KARNAUGH DE 4 VARIABLES.

Para transferir el contenido de la tabla al mapa de Karnaugh se colocan en su cuadro correspondiente los minitérminos para los cuales la función es verdadera. Con el propósito de facilitar la transferencia, estos minitérminos se sustituyen por 1's unos. Los cuadros restantes pueden llenarse con ceros e indican los minitérminos que no aparecen en la función. Los ceros pueden omitirse si se desea.

Ejemplo 5.8

Transfiera el contenido de la siguiente tabla de verdad a su mapa de Karnaugh.

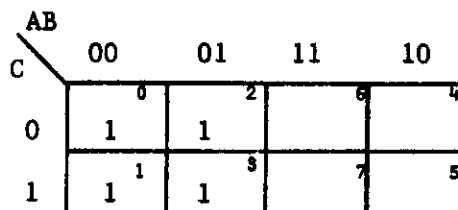
| AB | F(AB) |
|----|-------|
| 00 | 1 |
| 01 | 1 |
| 10 | 0 |
| 11 | 0 |

| | | A | |
|---|---|---|---|
| | | 0 | 1 |
| B | 0 | 1 | |
| | 1 | 1 | |

Ejemplo 5.9

Transfiera el contenido de la siguiente tabla de verdad, de una función de 3 variables a su mapa de Karnaugh correspondiente.

| ABC | F(ABC) |
|-----|--------|
| 000 | 1 |
| 001 | 1 |
| 010 | 1 |
| 011 | 1 |
| 100 | 0 |
| 101 | 0 |
| 110 | 0 |
| 111 | 0 |

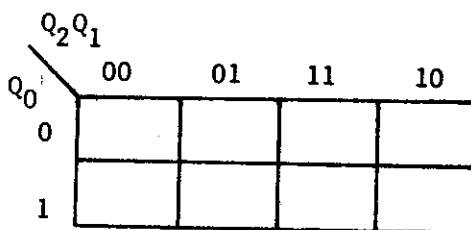


En los arreglos que hemos visto anteriormente la letra A representa la variable de mayor peso. Obviamente el orden de las variables puede cambiarse, por ejemplo que sea A la variable con un peso de 2^0 (menor peso).

Cuando se usan nombres con subíndices numéricos para las variables de un sistema, esto cambia. La variable X_0 siempre será la de menor peso, 2^0 como se muestra en el ejemplo 5.10

Ejemplo 5.10

| Q_2 | Q_1 | Q_0 | F |
|-------|-------|-------|---|
| 000 | | | |
| 001 | | | |
| 010 | | | |
| 011 | | | |
| 100 | | | |
| 101 | | | |
| 110 | | | |
| 111 | | | |



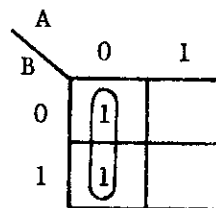
Cuando aparecen 1's (unos) en cuadros adyacentes significa que existe entre ellos una variable redundante, es decir que al agruparlos se elimina una variable, usando la siguiente propiedad del álgebra booleana:

$$A\bar{B} + AB = A(\bar{B}+B) = A$$

Ejemplo 5.11

Simplificar la función cuya tabla de verdad aparece a continuación.

| AB | F |
|----|---|
| 00 | 1 |
| 01 | 1 |
| 10 | 0 |
| 11 | 0 |



Del grupo formado se observa que la variable B es redundante ya que adquiere el valor de \bar{B} y B a lo largo del grupo, mientras que A permanece constante.

Por lo tanto:

$$F(AB) = A$$

De este ejemplo podemos deducir que el nombre que toma un grupo es igual al de la variable o variables que no cambian.

Un mismo UNO puede agruparse una o varias veces con diferentes unos-adyacentes, y así sintetizar el método de "DUPLICACION DE UN MINITERMINO YA EXISTENTE" discutido en la sección 5.2.2. de este mismo capítulo.

Ejemplo 5.12

Obtenga la forma más simple de la función cuya tabla de verdad aparece a continuación.

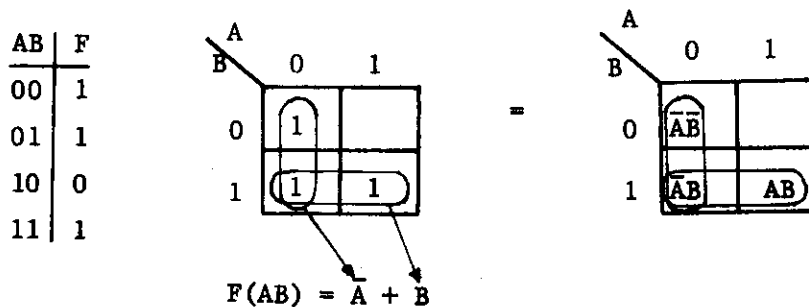
| AB | F |
|----|---|
| 00 | 1 |
| 01 | 1 |
| 10 | 0 |
| 11 | 1 |

$$F(AB) = \bar{A}\bar{B} + \bar{A}B + AB$$

1- ALGEBRAICAMENTE se puede lograr una máxima simplificación duplicando el término $\bar{A}B$.

$$\begin{aligned} F(AB) &= \bar{A}\bar{B} + \bar{A}B + \bar{A}B + AB \\ &= \bar{A}(\bar{B}+B) + B(\bar{A}+A) \\ F(AB) &= \bar{A}+B \end{aligned}$$

2- Por medio del MAPA DE KARNAUGH el minitérmino $\bar{A}B$ puede formar parte de dos grupos distintos.



Ejemplo 5.13

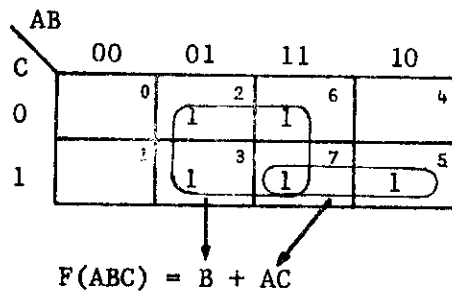
Simplifique la siguiente función a partir de su tabla de verdad.

| ABC | F |
|-----|---|
| 000 | 0 |
| 001 | 0 |
| 010 | 1 |
| 011 | 1 |
| 100 | 0 |
| 101 | 1 |
| 110 | 1 |
| 111 | 1 |

$$\begin{aligned}
 F &= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + \bar{A}BC \\
 &= \boxed{\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C}} + \boxed{A\bar{B}\bar{C} + A\bar{B}C} + \boxed{\bar{A}BC} \\
 &= \boxed{\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C}} + \boxed{A\bar{B}\bar{C} + A\bar{B}C} + \boxed{\bar{A}BC} \\
 &= \bar{A}B(\bar{C}+C) + AB(\bar{C}+C) + AC(\bar{B}+B) \\
 &= \boxed{\bar{A}B} + \boxed{AB} + AC \\
 &= B(\bar{A} + A) + AC \\
 F &= B + AC
 \end{aligned}$$

En la simplificación algebraica anterior podemos observar que los minterminos 2, 3, 6 y 7 se agruparon en forma separada para posteriormente llegar a solo la variable B. Por otro lado los minterminos 7 y 5 se agruparon reduciéndose a la combinación AC.

Veamos la solución por medio del mapa de Karnaugh.



El nombre del grupo formado por los minterminos 2, 3, 6 y 7 es igual al de la variable que no cambia B. Y el nombre del grupo formado por los minterminos 5 y 7 es AC.

REGLAS PARA EL USO DEL MAPA DE KARNAUGH

- 1- Formar el menor número de grupos
- 2- Formar cada grupo con la mayor cantidad de unos posible.
- 3- Todos los unos deberán agruparse, tomando en cuenta que un solo mínimo término puede formar un grupo.
- 4- El número de unos agrupados en un lazo debe ser una cantidad potencia de 2, (2^n) por ejemplo: 1, 2, 4, 8, 16, etc.

Estas reglas se acompañan de los llamados, grupos típicos de UNOS -- adyacentes. Un par de unos se consideran adyacentes entre sí, cuando -- son contiguos en forma horizontal o vertical, pero no diagonalmente.

Todos los cuadros de un mapa de Karnaugh son adyacentes entre sí, -- esto puede manifestarse en forma más clara en un mapa de 3 variables en -- adelante.

Para un mapa de 3 variables 0 y 1 y 4,5 son adyacentes entre sí puesto que existe un solo cambio de un extremo a otro del mapa (La variable - A).

Para indicar un grupo de un extremo a otro de un mapa se marca como se muestra en la fig. 5.11.

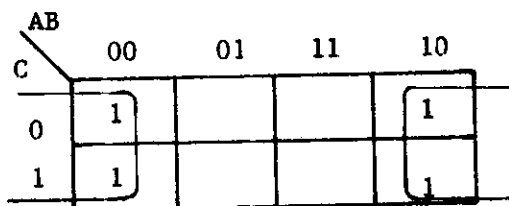
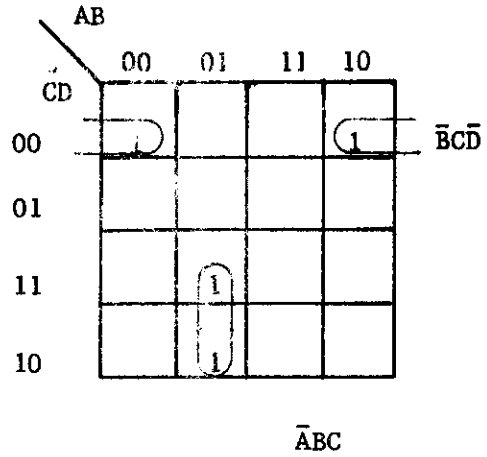
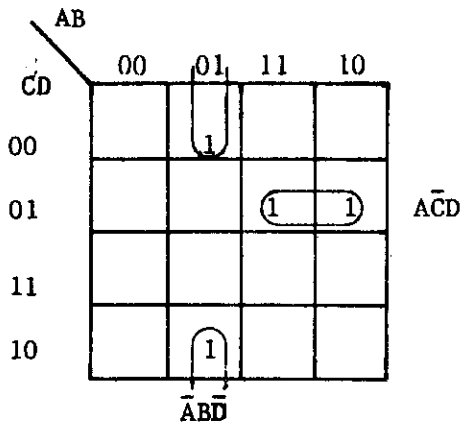
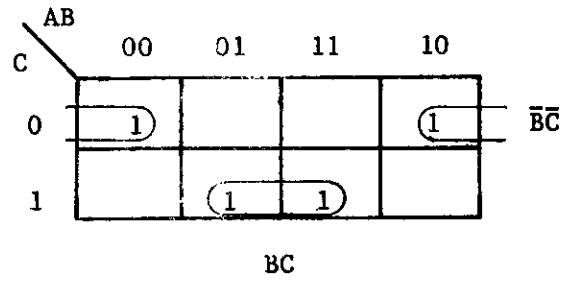
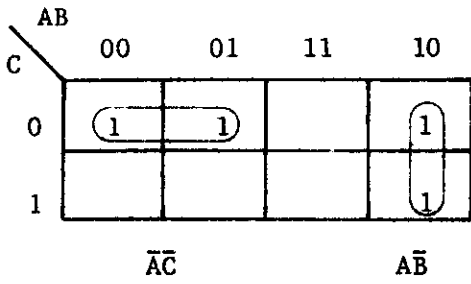
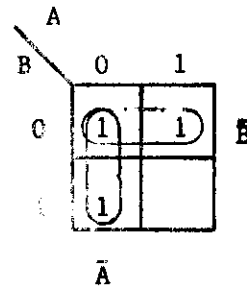
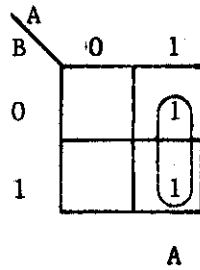
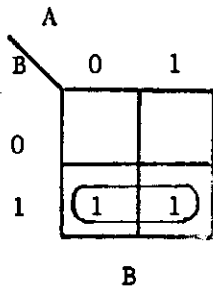
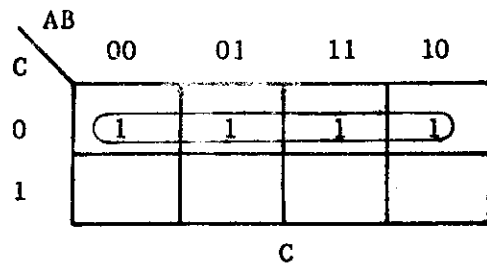
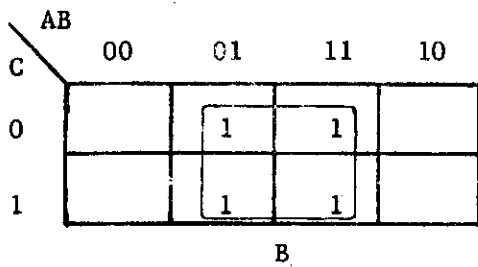


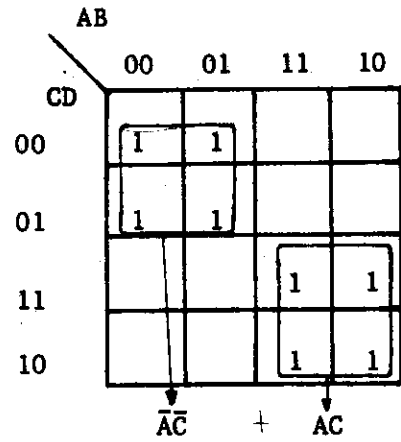
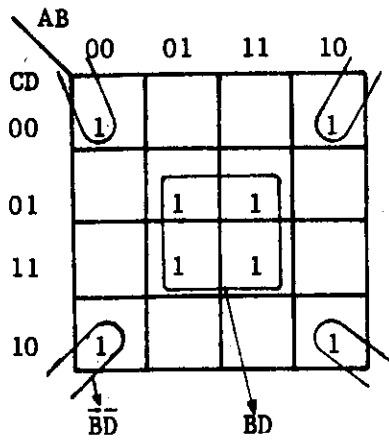
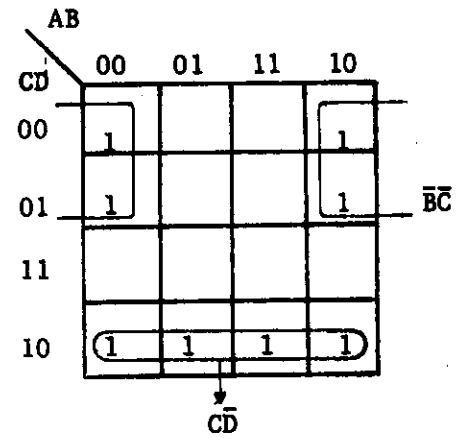
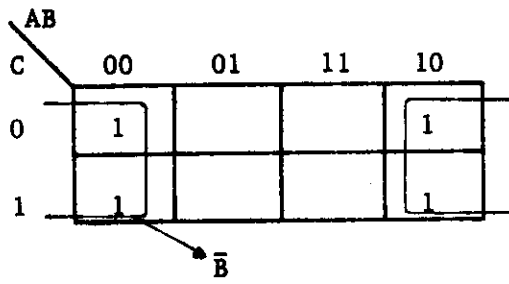
Fig. 5.11 LOS GRUPOS DE UN EXTREMO A OTRO DE UN MAPA SE MARCAN CON LAZOS ABIERTOS

Grupos Típicos de 2 minitérminos

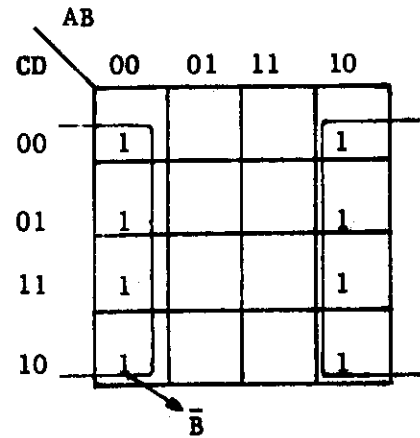
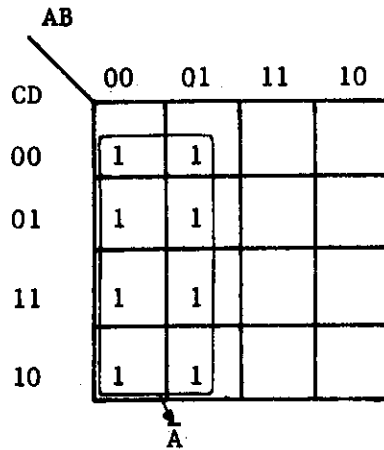


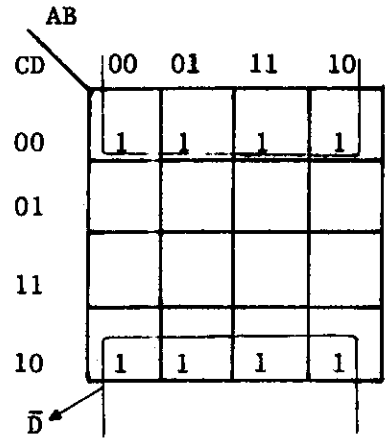
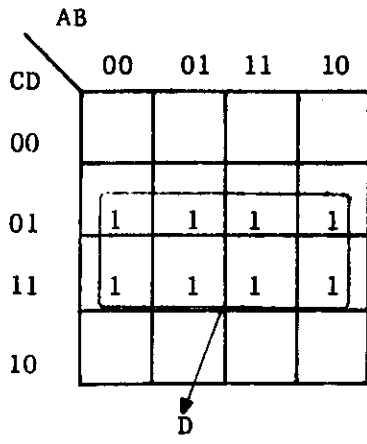
Grupos Típicos de 4 minitérminos



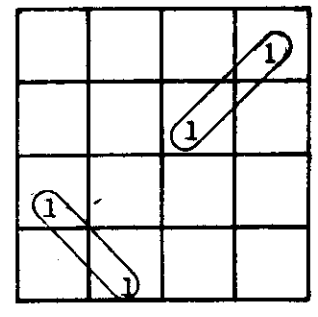
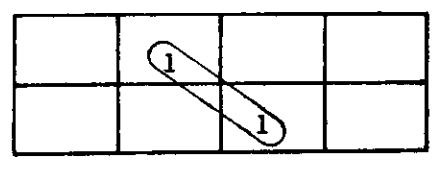
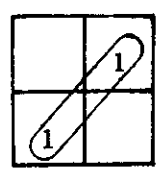


Grupos de 8 Minitérminos

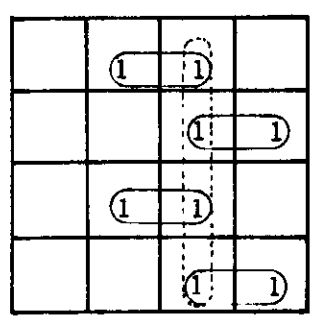




Grupos no permitidos



Lazo redundante

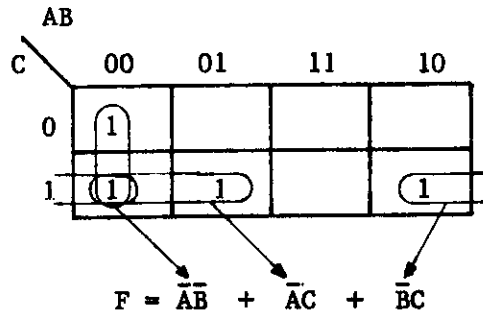


Ejemplo 5.14

Simplifique la siguiente función booleana, por medio del mapa de Karnaugh.

$$F(ABC) = \Sigma 0, 1, 3, 5$$

| ABC | F |
|-----|---|
| 000 | 1 |
| 001 | 1 |
| 010 | 0 |
| 011 | 1 |
| 100 | 0 |
| 101 | 1 |
| 110 | 0 |
| 111 | 0 |

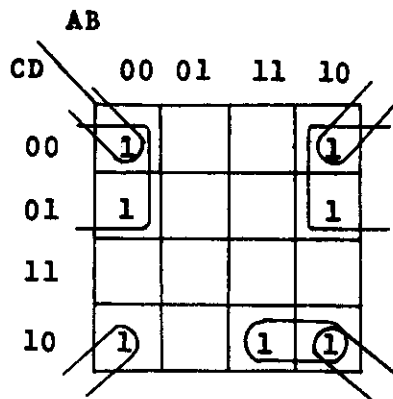


Ejemplo 5.15

Simplifique la siguiente función utilizando el mapa de Karnaugh.

$$F(ABCD) = \Sigma 0, 1, 2, 8, 9, 10, 14$$

| ABCD | F |
|------|---|
| 0000 | 1 |
| 0001 | 1 |
| 0010 | 1 |
| 0011 | 0 |
| 0100 | 0 |
| 0101 | 0 |
| 0110 | 0 |
| 0111 | 0 |
| 1000 | 1 |
| 1001 | 1 |
| 1010 | 1 |
| 1011 | 0 |
| 1100 | 0 |
| 1101 | 0 |
| 1110 | 1 |
| 1111 | 0 |



$$F(ABC) = \bar{B}\bar{C} + \bar{B}\bar{D} + AC\bar{D}$$

5.3.2 PRODUCTOS DE SUMATORIAS A PARTIR DE UN MAPA DE KARNAUGH

A partir de un mapa de Karnaugh, es posible obtener una función booleana expresada en la forma de productos de sumatorias. El primer paso es encontrar la forma de sumatoria de productos para \bar{F} . Esto se logra agrupando CEROS en vez de UNOS en el mapa y después aplicar el teorema de D' Morgan para convertir la \bar{F} en F . En algunas ocasiones la función expresada a partir de \bar{F} es más compacta.

Ejemplo 5.16

Dada $F_{(ABC)} = \Sigma 4, 5, 7$. Simplificar F en la forma de sumatoria de productos y productos de sumatorias a partir del mapa de Karnaugh.

a) A partir de F .

| | | | | |
|---|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |

$$F_{(ABC)} = A\bar{B} + AC \dots CU = 7$$

b) A partir de \bar{F} .

| | | | | |
|---|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |

$$\bar{F}_{(ABC)} = \bar{A} + B\bar{C}$$

$$F_{(ABC)} = \overline{\bar{A} + B\bar{C}}$$

$$F_{(ABC)} = A(\bar{B} + C) \dots CU = 5$$

5.3.3 MAPAS DE KARNAUGH DE 5 Y 6 VARIABLES

Un mapa de Karnaugh de 5 variables puede construirse en tres dimensiones. Colocando un mapa de 4 variables sobre otro mapa también de 4 variables, los términos de la parte inferior se numeran del 0 al 15 y corresponden a la zona de la variable de mayor peso negada, \bar{A} . los términos de la parte superior, corresponden a la variable A afirmada y se numeran del 16 al 31.

Para representar el mapa en dos dimensiones, se dividen los cuadros de un mapa de 4 variables por medio de una línea diagonal. Para colocar en la parte inferior de la línea, el mapa de \bar{A} y en la parte superior el mapa de A, como se muestra en la figura 5.12.

Los términos de la parte superior e inferior de la línea diagonal combinan igual que un mapa de 4 variables.

Los términos de un mismo cuadro pueden combinarse puesto que difieren únicamente en una variable.

Los términos que aparentemente son adyacentes, no lo son. Por ejemplo los términos 4 y 28 no son adyacentes, porque aparecen en diferentes columnas y en diferente posición respecto a la diagonal, además existe más de un cambio entre sus variables.

Cada término puede ser adyacente a otros 5 términos, 4 en la misma posición respecto a la diagonal y uno en el mismo cuadro. Como aparece en la fig. 5.13.

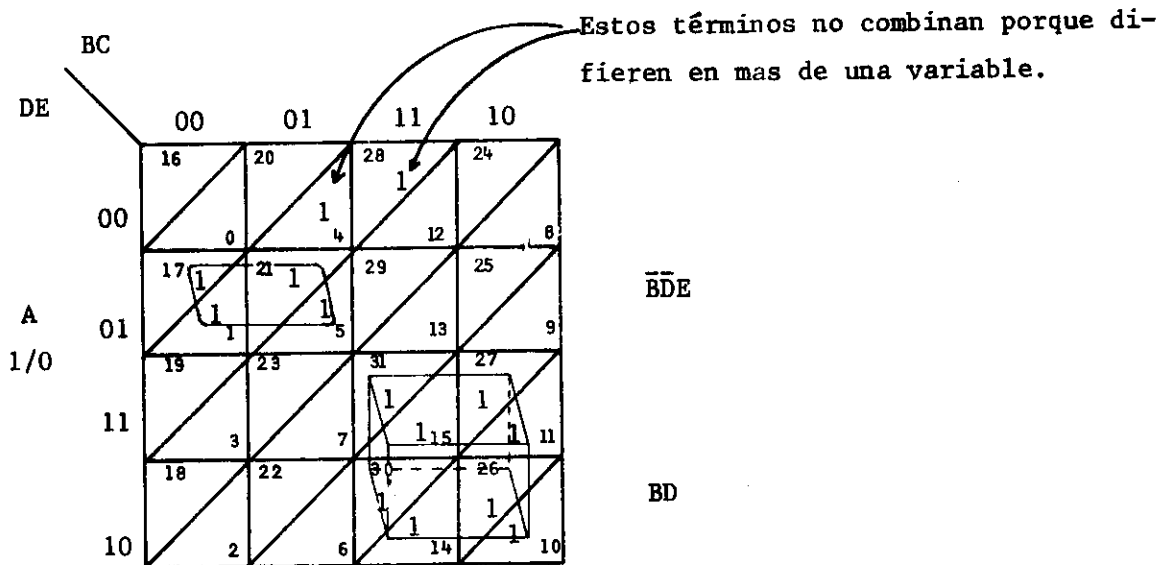


Fig. 5.12 MAPA DE KARNAUGH PARA UNA FUNCION DE 5 VARIABLES

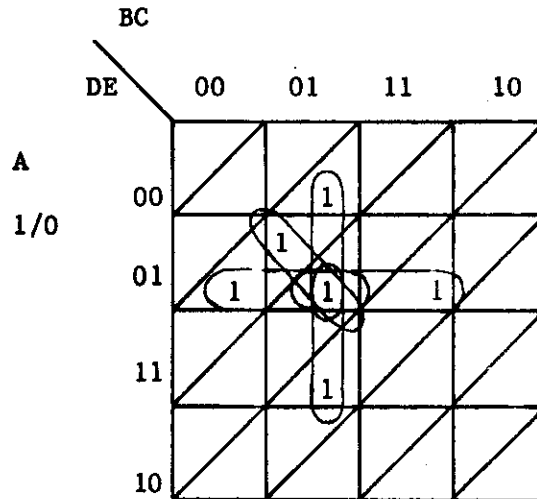
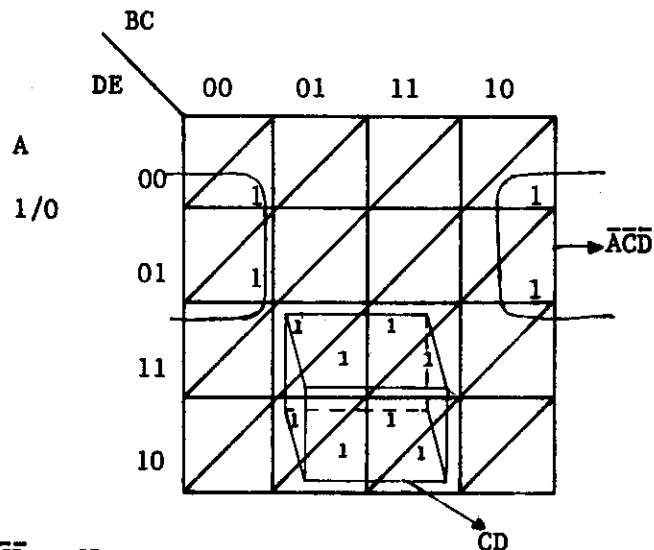


Fig. 5-13. UN MISMO UNO PUEDE SER ADYACENTE A OTROS 5 UNOS

Ejemplo 5.17

Simplifique la siguiente función usando un mapa de Karnaugh de 5 variables.

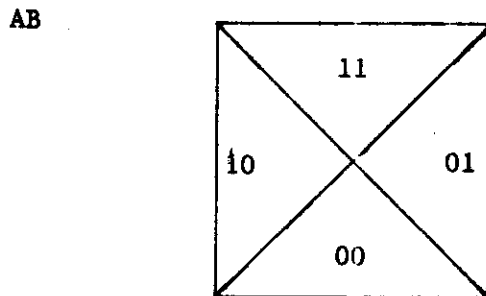
$$F_{(ABCDE)} = \Sigma 0, 1, 6, 7, 8, 9, 14, 15, 22, 23, 30, 31.$$



$$F_{(ABCDE)} = \overline{A}CD + CD$$

Un mapa de Karnaugh de 6 variables puede construirse dividiendo el cuadro de un mapa de 4 variables en 4 partes como se muestra en la fig. 5.14, asignando los valores de A y B (variables de mayor peso) a cada parte de un cuadro y las variables C D E y F a las hileras y columnas.

A y B se distribuyen en la siguiente forma



Los minitérminos del 0 — 15 se grafican en la posición AB = 00.

Los minitérminos del 16 — 31 se grafican en la posición AB = 01.

Los minitérminos del 32 — 47 se grafican en la posición AB = 10.

Los minitérminos del 48 — 63 se grafican en la posición AB = 11.

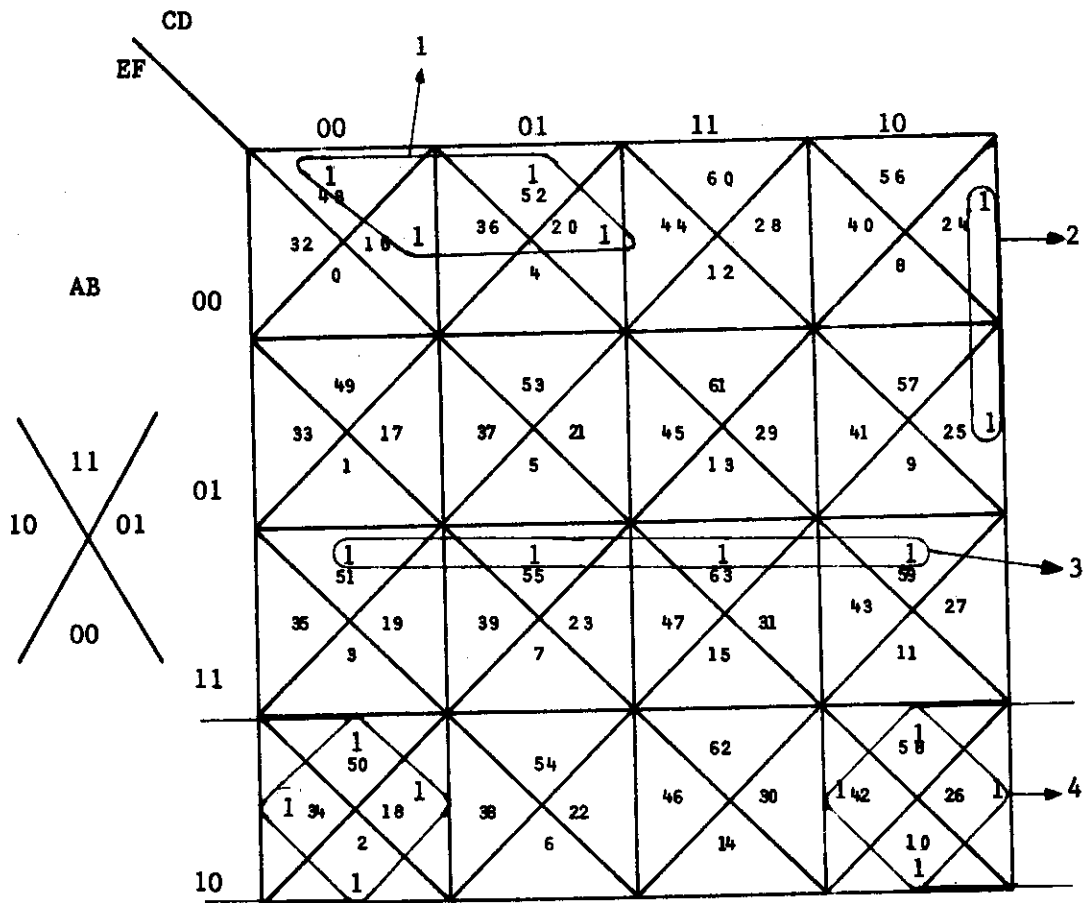


Fig. 5.14 MAPA DE KARNAUGH PARA UNA FUNCION DE 6 VARIABLES.

Los grupos que aparecen en el mapa de la fig. 5.13 son los siguientes:

- 1 - $\overline{B}\overline{C}\overline{E}\overline{F}$
- 2 - $\overline{A}B\overline{C}\overline{D}\overline{E}$
- 3 - $A\overline{B}E\overline{F}$
- 4 - $\overline{D}\overline{E}\overline{F}$

PROBLEMAS PROPUESTOS

CAPITULO 5

- 1.- ¿Para que nos sirve la minimización de las funciones booleanas?
- 2.- Mencione los cuatro métodos de minimización.
- 3.- ¿Para que nos sirven los Mapas de Karnaugh y cuantos cuadros tiene - un Mapa de N variables?
- 4.- ¿Cuales son las reglas para usar el mapa de Karnaugh?
- 5.- Minimice algebraicamente las siguientes funciones:

$$F = XY(X+Y)$$

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

$$F = \bar{A}BC + A\bar{B}C + ABC$$

$$F = A(A+B)$$

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

- 6.- Simplifique las siguientes funciones utilizando el Mapa de Karnaugh.

$$F(ABC) = \sum 1,2,6,$$

$$F(ABCD) = \sum 1,2,3,8,9,12,14$$

$$F(ABC) = \sum 1,3,4,5,$$

$$F(ABC) = \sum 0,1,4,5,6,7$$

$$F(ABC) = \sum 0,1,4,5,7$$

$$F(ABCD) = \sum 1,3,6,7,10,11,13,15$$

$$F(ABC) = \sum 0,1,2,3,4,5$$

$$F(ABCD) = \sum 0,1,2,8,9,10,11,14,15$$

$$F(ABCD) = \sum 2,3,4,5,6,8,9,10,11,12,13,14$$

$$F(ABCD) = \sum 0,1,2,3,8,9,10,11,12,13,14,15$$

$$F(ABCDE) = \sum 0,2,5,7,8,10,13,15,16,18,21,23,24,26,29,31$$

$$F(ABCDE) = \sum 1,3,4,6,9,11,12,14,17,19,22,30$$

$$F(ABCDEF) = \sum 0,2,8,10,16,18,24,26,32,34,40,42,48,50,53,55,56,58,61,63$$

$$F(ABCDEF) = \sum 16,17,18,19,20,23,24,25,26,27,28,31,32,33,34,35,36,40,41,42,43,44,55,63$$

6 *Diseño Combinacional*

6.0 DEFINICION DE UN BLOQUE COMBINACIONAL

El término SISTEMA COMBINACIONAL describe a un bloque digital cuya salida es una función booleana de sus entradas. En otras palabras, los valores de la salida (0 ó 1) de un bloque combinacional dependen únicamente de la COMBINACION que tomen los valores (0 ó 1) de sus variables de entrada.

Un sistema combinacional puede tener una o más entradas y una o más salidas. Estas salidas no pueden ser retroalimentadas a la entrada.

En la fig. 6.0 aparece la representación de un sistema combinacional generalizado.

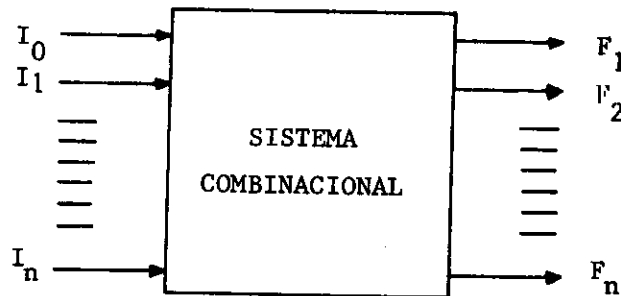


Fig. 6.0 REPRESENTACION DE UN SISTEMA COMBINACIONAL

Cuando se presenta un cambio en la combinación de las variables de entrada de un sistema combinacional, las salidas toman nuevos valores, - estos nuevos valores aparecen con un intervalo de tiempo, determinado -- por los tiempos de propagación inherentes a cada compuerta usada para implementar el circuito.

El hecho de no tener retroalimentación asegura que los cambios en - las entradas produzcan cambios en las salidas sin generar inestabilidad- en el bloque.

Las aplicaciones típicas de los sistemas combinacionales son, en -- bloques de control digital, convertidores de código, circuitos que efectúan operaciones aritméticas, como sumas, comparaciones, etc. y que forman la estructura básica de calculadoras y computadoras digitales. Algunos de estos ejemplos se discuten en el punto 6.2 de este capítulo.

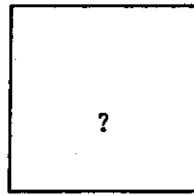
6.1 METODOLOGIA DE DISEÑO COMBINACIONAL

El diseño de un sistema combinacional se puede resumir básicamente- en los siguientes pasos.

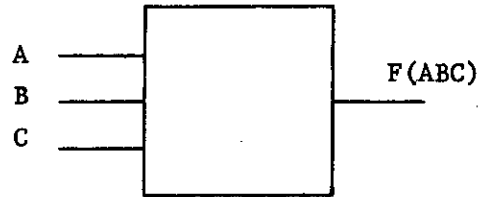
1. Establecer las funciones específicas del bloque combinacional.
2. Determinar la cantidad de entradas y salidas al sistema.
3. Representar el comportamiento del sistema por medio de una tabla de verdad.
4. Obtener la función booleana de salida del sistema a partir de la tabla de verdad, usando el método de minimización algebraica o - del mapa de Karnaugh.
5. Implementar el sistema con elementos lógicos.

Estos primeros pasos de diseño nos conducirán a la obtención del -- prototipo de prueba en el laboratorio. Posteriormente se discutirán los detalles para efectuar las pruebas de campo del prototipo.

1)



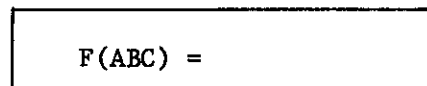
2)



3)

| ABC | F |
|-----|---|
| — | — |
| — | — |
| — | — |
| — | — |

4)



5)

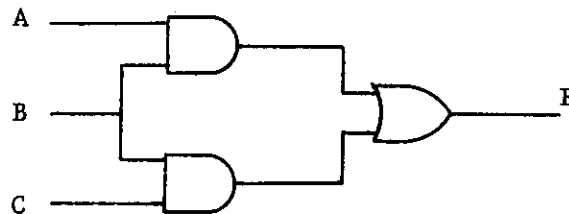


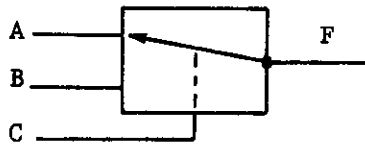
Fig. 6.1 PASOS DEL DISEÑO COMBINACIONAL

6.2 EJEMPLOS DE DISEÑO

Ejemplo 6.0 SELECTOR (MULTIPLEXER)

Diseñe un circuito combinacional que tenga tres entradas denominadas A, B y C, una salida denominada F. Si la entrada C es igual a cero lógico, la salida debe ser igual a la entrada A y si $C = 1$, la salida debe ser igual a B.

1. La descripción anterior cumple con el primer paso de diseño.
2. Determinar el # de entradas y salidas.



| C | F |
|---|---|
| 0 | A |
| 1 | B |

3. Se puede tener una mayor visualización del problema si se acomoda la tabla de verdad en la siguiente forma.

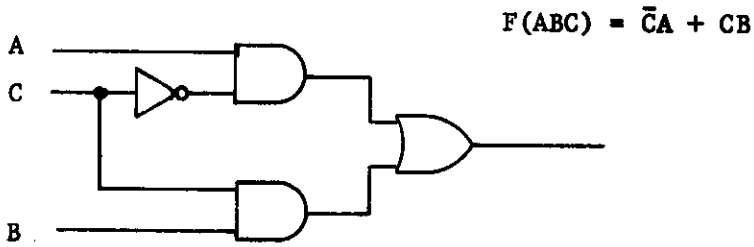
| CBA | F |
|-----|---|
| 000 | 0 |
| 001 | 1 |
| 010 | 0 |
| 011 | 1 |
| 100 | 0 |
| 101 | 0 |
| 110 | 1 |
| 111 | 1 |

4. Obtener $F(CBA)$

| | | CB | | | |
|---|---|----|----|----|----|
| | | 00 | 01 | 11 | 10 |
| A | 0 | 0 | 0 | 1 | 0 |
| | 1 | 1 | 1 | 1 | 0 |

$$F(CBA) = \bar{C}A + CB$$

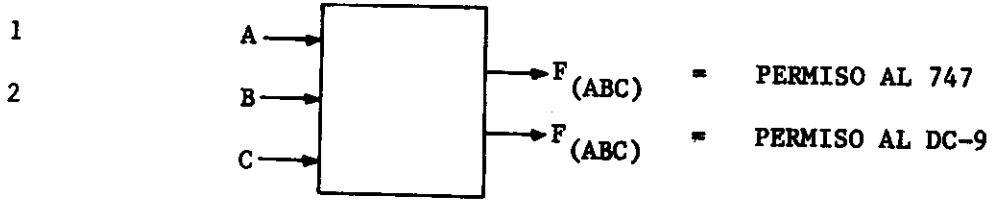
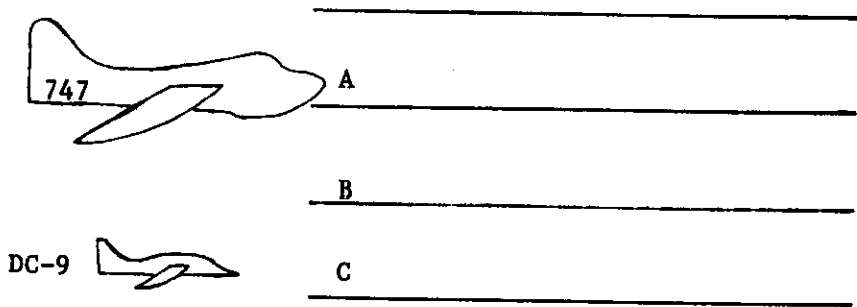
Implementación



Ejemplo 6.1

Diseñe un circuito que indique al operador de la torre de control - de un aeropuerto, que avión puede aterrizar cuando alguna de las pistas - esté ocupada.

- a) El aeropuerto tiene 3 pistas, A, B y C
- b) Pueden aterrizar jumbos 747 ó DC-9's
- c) Un 747 necesita dos pistas contiguas para aterrizar y un DC-9 -- solo una.
- d) El 747 tiene mayor prioridad que el DC-9



3

| ABC | 747 | DC-9 |
|-----|-----|------|
| 000 | 0 | 0 |
| 001 | 0 | 1 |
| 010 | 0 | 1 |
| 011 | 1 | 0 |
| 100 | 0 | 1 |
| 101 | 0 | 1 |
| 110 | 1 | 0 |
| 111 | 1 | 1 |

un cero en A, B o C indican que la pista no está disponible.

4

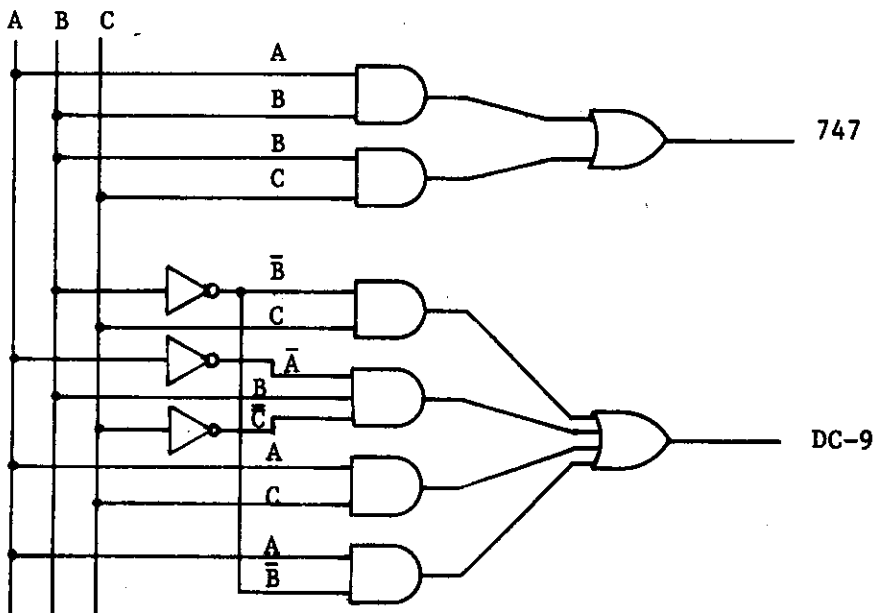
| | | AB | | | |
|---|---|----|----|----|----|
| C | | 00 | 01 | 11 | 10 |
| | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |

$$747 (ABC) = AB + BC$$

| | | AB | | | |
|---|---|----|----|----|----|
| C | | 00 | 01 | 11 | 10 |
| | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |

$$DC-9 (ABC) = \bar{B}C + \bar{A}\bar{B}C + AC + A\bar{B}$$

5 Implementación



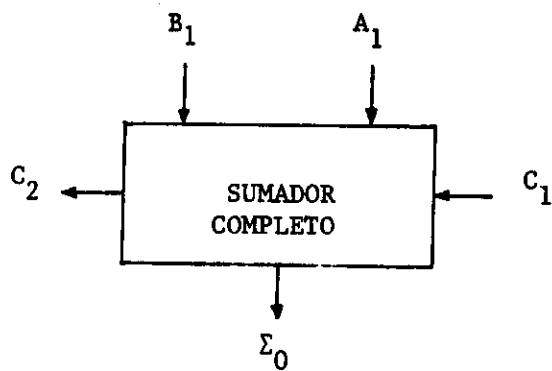


Fig. 6.3 SIMBOLO PARA UN SUMADOR COMPLETO

Para efectuar la suma de una palabra de 4 bit's por ejemplo se puede usar 3 sumadores completos y un medio sumador como se muestra en la - fig. 6.4

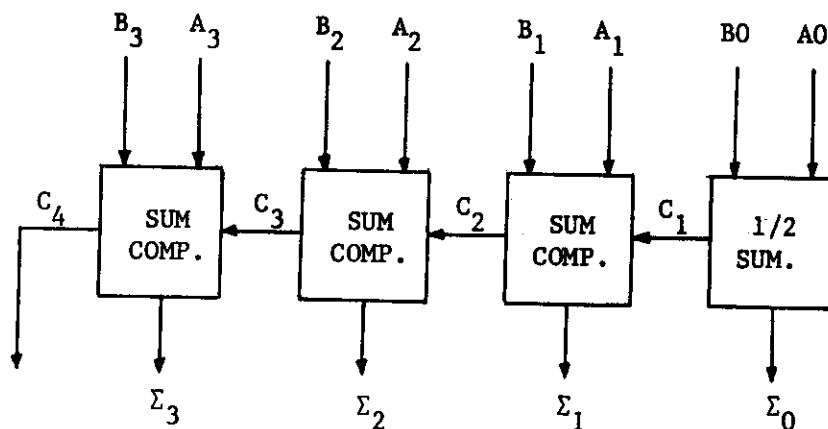
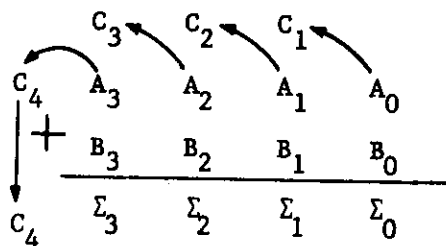


Fig. 6.4 SUMA BINARIA DE 4 BITS. AL SUMAR LOS BITS DE MENOR PESO NO SE TOMA EN CUENTA EL ACARREO ANTERIOR PUESTO QUE NO--EXISTEN BITS ANTERIORES.

Entonces el bloque combinacional a diseñar tiene 3 entradas A, B y C y 2 salidas Σ_1 y C_2 . La tabla de verdad para el sumador aparece a continuación.

| $A_1 B_1 C_1$ | Σ_1 | C_2 |
|---------------|------------|-------|
| 000 | 0 | 0 |
| 001 | 1 | 0 |
| 010 | 1 | 0 |
| 011 | 0 | 1 |
| 100 | 1 | 0 |
| 101 | 0 | 1 |
| 110 | 0 | 1 |
| 111 | 1 | 1 |

| | | AB | | | |
|---|---|----|----|----|----|
| C | | 00 | 01 | 11 | 10 |
| | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | |

| | | AB | | | |
|---|---|----|----|----|----|
| C | | 00 | 01 | 11 | 10 |
| | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | |

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

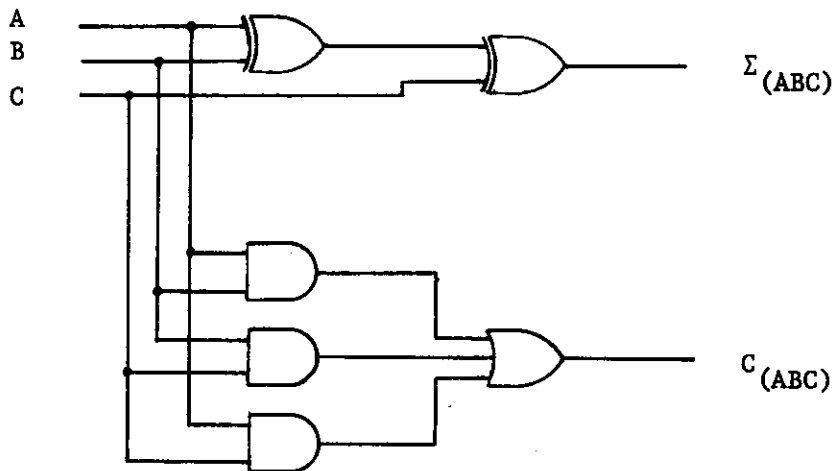
$$\Sigma_{(ABC)} = \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}C + B\bar{C})$$

$$\Sigma_{(ABC)} = \bar{A}(B \oplus C) + A(\overline{B \oplus C})$$

$$\Sigma_{(ABC)} = A \oplus (B \oplus C)$$

$$\Sigma_{(ABC)} = A \oplus B \oplus C$$

$$C_{(ABC)} = AB + BC + AC$$



6.3 SISTEMAS QUE NO ESTAN COMPLETAMENTE ESPECIFICADOS

En los ejemplos de diseño combinacional que hemos visto anteriormente podemos notar que para cada combinación de las variables de entrada - existe un valor definido para la salida o salidas, a este tipo de sistemas combinacionales se les da el nombre de SISTEMAS COMPLETAMENTE ESPECIFICADOS.

En muchos casos se pueden presentar bloques en que sus combinaciones de entrada por alguna u otra razón no requieren a su salida un valor específico, es decir para esa combinación de entrada la salida puede ser CERO o UNO, no importa cual.

Estos casos se clasifican de dos formas. La primera se conoce como DON'T CARE y describe a una combinación de las variables de entrada para la cual no interesa que valor pueda tomar la salida.

El segundo caso es el CAN'T HAPPEN y se refiere a una salida cuya combinación de entrada jamás llega a presentarse.

A un sistema que contenga don't care's o can't happen's se le denomina SISTEMA COMBINACIONAL QUE NO ESTA COMPLETAMENTE ESPECIFICADO.

Para motivos de diseño el don't care o can't happen puede tomarse - como CERO o como UNO según convenga a la solución del mapa de Karnaugh, - y se indican con una X. Esto es muy útil, en la fig. 6.5 se muestra - - cuando se toma una X como "1" o como "0".

| ABC | F |
|-----|---|
| 000 | 0 |
| 001 | 0 |
| 010 | 1 |
| 011 | 1 |
| 100 | 0 |
| 101 | X |
| 110 | 1 |
| 111 | X |

| | | | | |
|---|----|----|----|----|
| | AB | | | |
| C | 00 | 01 | 11 | 10 |
| 0 | | 1 | 1 | |
| 1 | | 1 | X | X |

Fig. 6.5 LA "X" EN 7 CONVIENE TOMARLA COMO UNO LA QUE ESTA EN 5 CONVIENE TOMARLA COMO CERO

Si por algún motivo la combinación de entrada considerada como - - don't care o can't happen llegará a presentarse, el valor de la salida para esa combinación será igual al valor que se le impuso a la X en el mapa en el ejemplo de la fig. 6.5 si se llegara presentar la combinación ABC la salida será igual a UNO porque esa X fué tomada como UNO, - por el contrario si se presenta la combinación $\bar{A}\bar{B}C$ la salida será igual a CERO porque la X fué tomada como CERO.

Una última observación sobre las X's en un mapa, es que pueden formar grupos tomados como unos o como ceros, pero no se deben formar grupos de X'S solas.

Ejemplo 6.3

En la fig. 6.6 se muestra un dispositivo empleado para la detección de tres tipos de monedas que pasan por un plano inclinado. Consta de -- tres acas de luz que inciden sobre tres fotoceldas marcadas como A, B y C. Al incidir un rayo de luz sobre una fotocelda se genera un CERO LOGICO a su salida, al interrumpirse un az de luz la fotocelda genera un UNO LOGICO.

El problema es entonces diseñar un circuito cuyas entradas sean A - B y C y sus salidas indiquen si paso una moneda de .20, .50 ó 1.00.

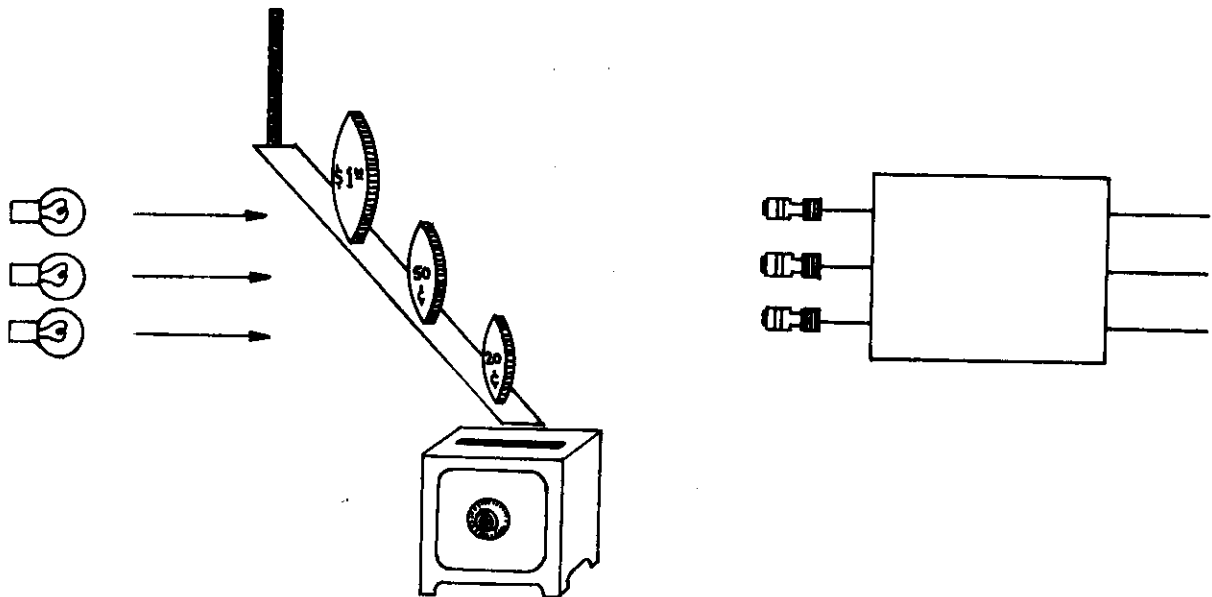
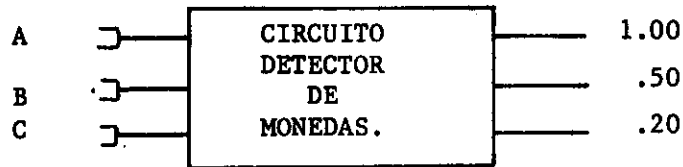


Fig. 6.6 SISTEMA DETECTOR DE TRES TIPOS DE MONEDAS.

1. El bloque queda descrito por la redacción anterior.
2. Determinar el # de entradas y salidas

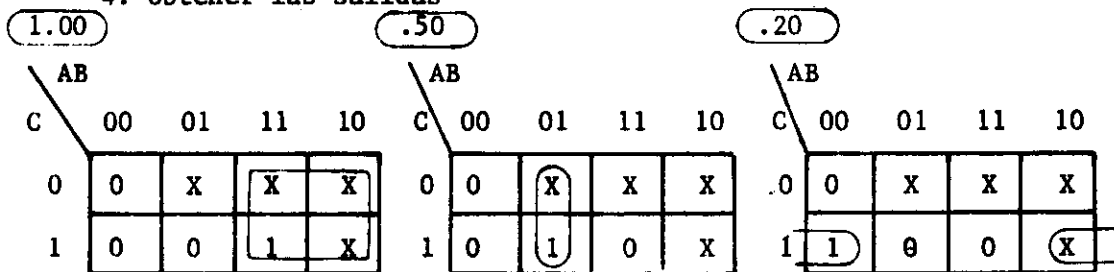


3. Tablas de verdad

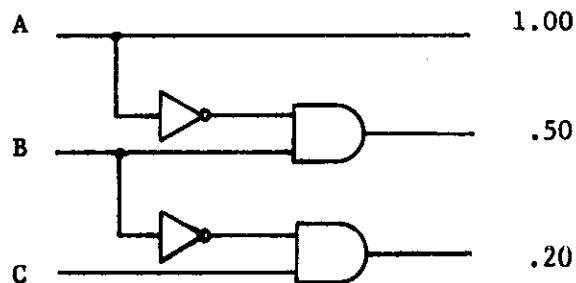
| ABC | 1.00 | .50 | .20 |
|-----|------|-----|-----|
| 000 | 0 | 0 | 0 |
| 001 | 0 | 0 | 1 |
| 010 | X | X | X |
| 011 | 0 | 1 | 0 |
| 100 | X | X | X |
| 101 | X | X | X |
| 110 | X | X | X |
| 111 | 1 | 0 | 0 |

ciertas combinaciones no pueden presentarse puesto que no es posible que una moneda se despegue del plano inclinado al bajar. En esta caso esas combinaciones son can't happen's.

4. Obtener las salidas



5. IMPLEMENTACION



6.4 DISPLAY DE 7 SEGMENTOS

El término DISPLAY se usa aquí, como otros tantos sin traducción. -- Una definición aproximada sería la de un dispositivo OPTICO empleado para visualizar en forma alfanumérica o gráfica una información expresada generalmente a partir de UNOS y CEROS.

El uso principal de un display se presenta cuando es necesario mostrar información a partir de un dispositivo digital. En este caso discutiremos un arreglo típico formado por siete segmentos con capacidad de -- mostrar caracteres numéricos decimales y algunos caracteres alfabéticos -- el nombre que recibe es precisamente DISPLAY DE 7 SEGMENTOS. Fig. 6.7

Estos display's se presentan comercialmente en una amplia variedad -- de tamaños, colores y tipos. La iluminación de cada segmento la producen, focos incandescentes, efectos fluorescentes sobre segmentos móviles, diodos, emisores de luz (led's), cristal líquido de cuarzo y otras técnicas.

Para mostrar información en el display es necesario diseñar un sistema combinacional que convierta un código BCD a un juego de 7 salidas que enciendan o apaguen cada segmento a fin de desplegar el carácter apropiado.

A este tipo de bloques convertidores de código se les llama también-DECODIFICADORES.

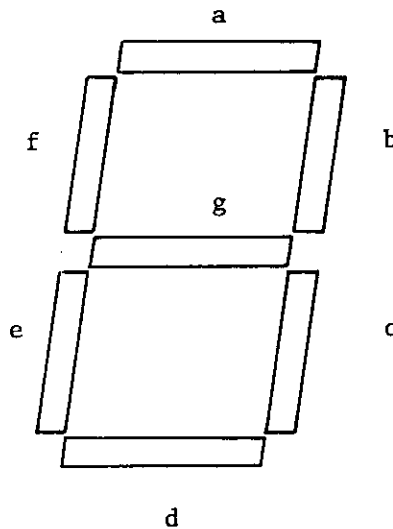


Fig. 6.7 DISPLAY DE 7 SEGMENTOS. CADA SEGMENTO SE MARCA CON UNA LETRA MINUSCULA DE LA a A LA g

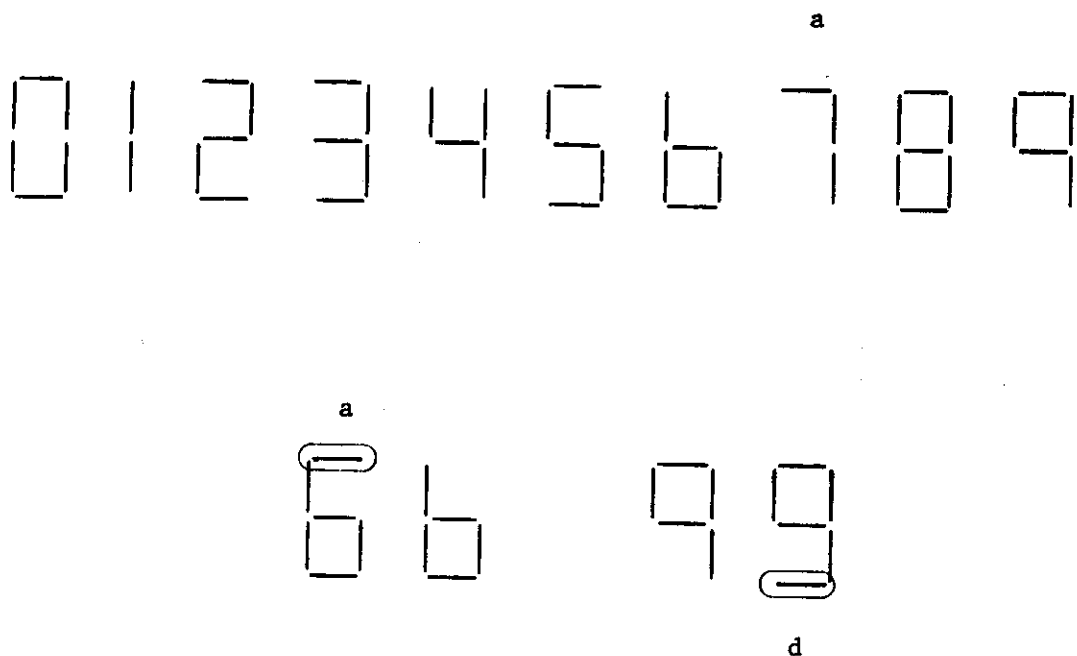
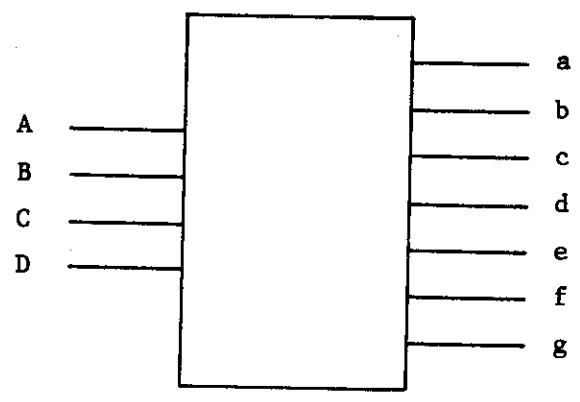


Fig. 6.8 CONFIGURACION PARA LOS NUMEROS DEL 0 AL 9 EN UN DISPLAY DE 7 SEGMENTOS. EXISTEN CARACTERES COMO EL 6 Y 9 QUE PUEDEN CONFIGURARSE INDISTINTAMENTE.

Ejemplo 6.4

Diseñe un decodificador de BCD a 7 SEGMENTOS

1. El bloque es un convertidor de código cuya entrada es BCD de 0 a 9, y la salida un código para manejar 7 segmentos.
2. # de entradas y salidas.



3. TABLA DE VERDAD

| ABCD | a | b | c | d | e | f | g |
|------|---|--------------|---|---|---|---|---|
| 0000 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0001 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0010 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0011 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0100 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0101 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0110 | X | 0 | 1 | 1 | 1 | 1 | 1 |
| 0111 | 1 | 1 | 1 | 0 | 0 | X | 0 |
| 1000 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1001 | 1 | 1 | 1 | X | 0 | 1 | 1 |
| 1010 | X | X | X | X | X | X | X |
| 1011 | X | | | | | | X |
| 1100 | X | | | | | | X |
| 1101 | X | CAN'T HAPPEN | | | | | X |
| 1110 | X | | | | | | X |
| 1111 | X | X | X | X | X | X | X |

4. MAPAS

a

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | 1 | | X | 1 |
| 01 | | 1 | X | 1 |
| 11 | 1 | 1 | X | X |
| 10 | 1 | X | X | X |

b

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | X | 1 |
| 01 | 1 | | X | 1 |
| 11 | 1 | 1 | X | X |
| 10 | 1 | | X | X |

c

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | X | 1 |
| 01 | 1 | 1 | X | 1 |
| 11 | 1 | 1 | X | X |
| 10 | | 1 | X | X |

d

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | 1 | | X | 1 |
| 01 | | 1 | X | X |
| 11 | 1 | | X | X |
| 10 | 1 | 1 | X | X |

e

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | 1 | | X | 1 |
| 01 | | | X | |
| 11 | | | X | X |
| 10 | 1 | 1 | X | X |

f

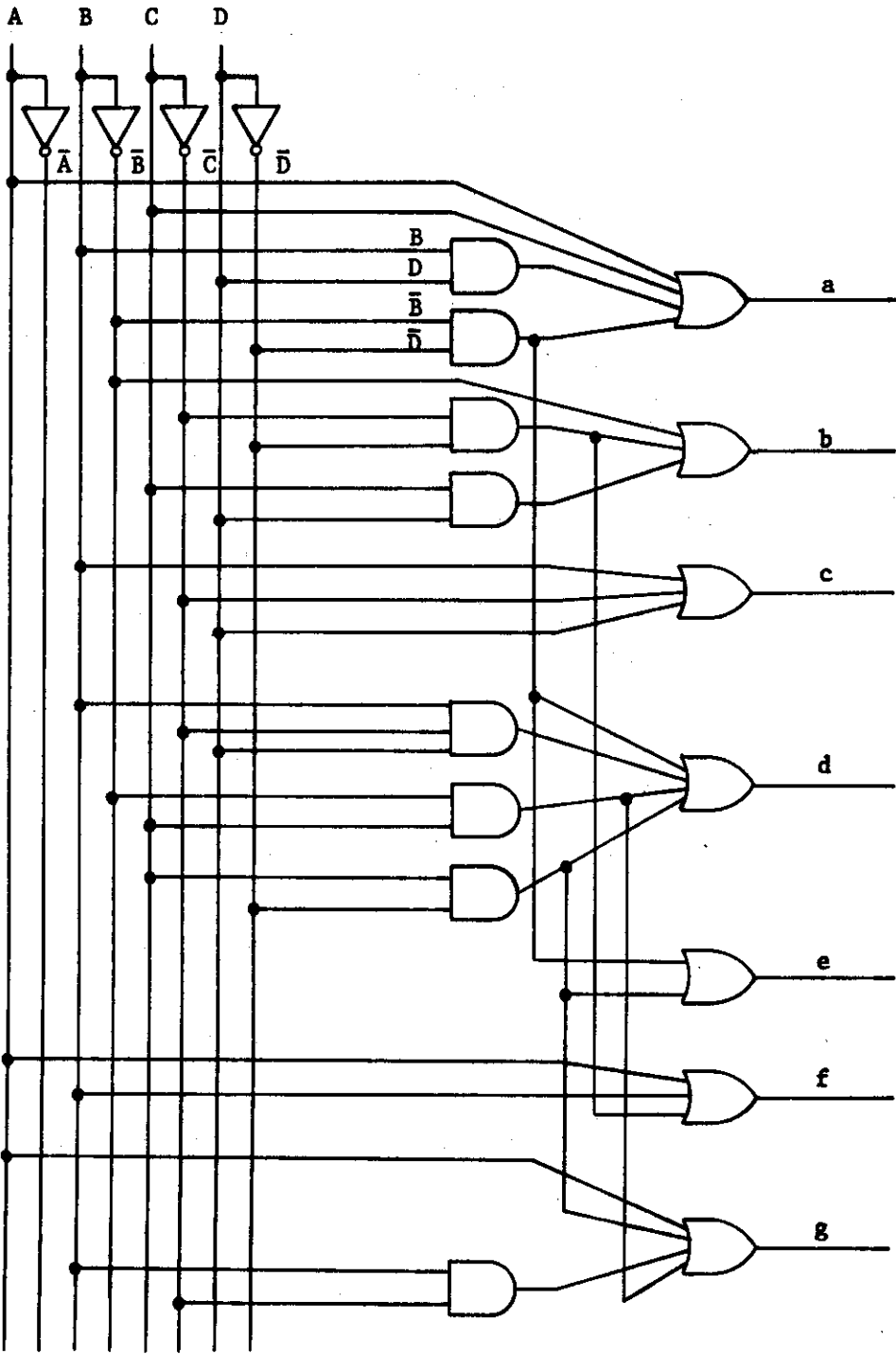
| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | X | 1 |
| 01 | | 1 | X | 1 |
| 11 | | X | X | X |
| 10 | | 1 | X | X |

g

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | | 1 | X | 1 |
| 01 | | 1 | X | 1 |
| 11 | 1 | | X | X |
| 10 | 1 | 1 | X | X |

$a = A + C + BD + \bar{B}\bar{D}$
 $b = \bar{B} + \bar{C}D + CD$
 $c = \bar{B} + \bar{C} + D$
 $d = \bar{B}\bar{D} + \bar{B}CD + \bar{B}C + C\bar{D}$
 $e = \bar{B}\bar{D} + C\bar{D}$
 $f = A + B + \bar{C}\bar{D}$
 $g = A + C\bar{D} + \bar{B}C + \bar{B}C$

5. IMPLEMENTACION



6.5 DECODIFICADORES

Un decodificador es un circuito lógico que convierte un código de entrada de n bits a un código de salida de una cantidad menor o igual -- a 2^n bits. La cantidad de combinaciones que se pueden presentar a la salida es igual a 2^n bits de entrada.

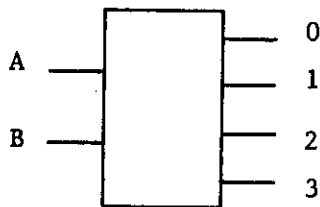
El decodificador de BCD a 7 segmentos es un bloque donde la salida no tiene una relación directa con la entrada sin embargo existen decodificadores donde sí se presenta esta relación. Este tipo de codificadores son los llamados de "n líneas de entrada a 2^n líneas de salida" para cada combinación de las líneas de entrada se habilita una sola salida a la vez, ya sea con niveles altos o bajos.

Ejemplo 6.5

Diseñar un decodificador de 2 a 4 líneas con salidas activadas a nivel alto.

1 - ok

2 - dos entradas y 4 salidas



3 - Tabla de Verdad

| AB | 0 | 1 | 2 | 3 |
|----|---|---|---|---|
| 00 | 1 | 0 | 0 | 0 |
| 01 | 0 | 1 | 0 | 0 |
| 10 | 0 | 0 | 1 | 0 |
| 11 | 0 | 0 | 0 | 1 |

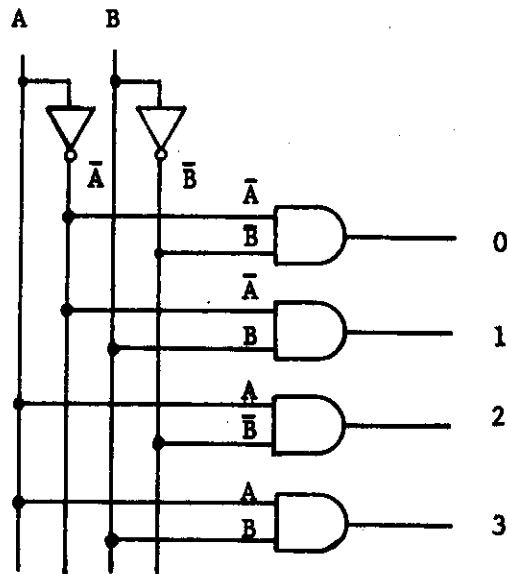
4 - Cada salida es igual a un minitérmino y no existen X's es innecesario recurrir a los mapas.

$$0_{(AB)} = \bar{A}\bar{B}$$

$$1_{(AB)} = \bar{A}B$$

$$2_{(AB)} = A\bar{B}$$

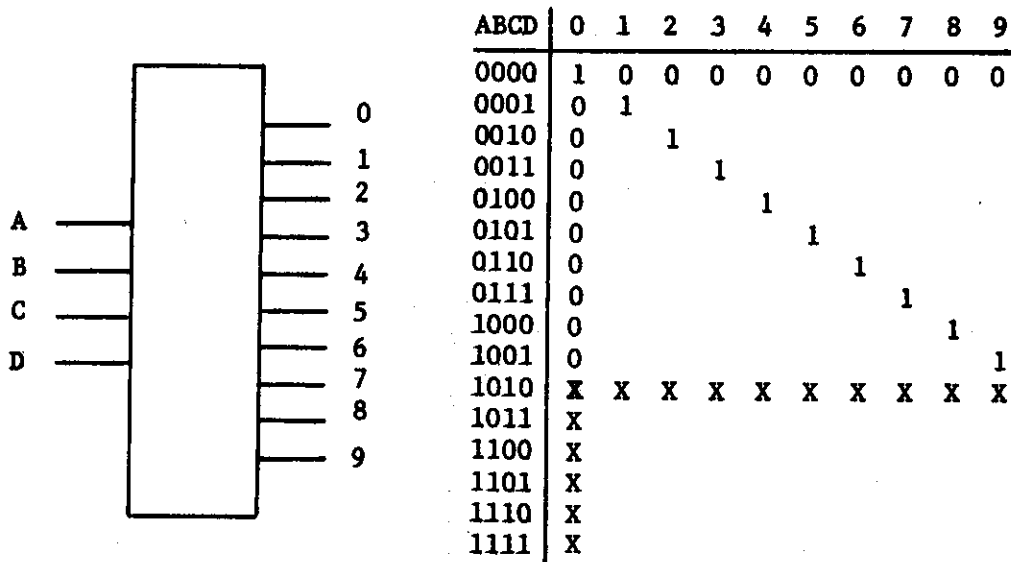
$$3_{(AB)} = AB$$



Algunos decodificadores no usan todos los 2^n posibles códigos de salida, sino solo algunos. Por ejemplo, un decodificador de BCD a DECIMAL tiene un código de entrada de 4 bits y 10 salidas. Las cuales tienen valores solo para las combinaciones de entrada del 0 al 9.

Ejemplo 6.6

Diseñe un decodificador de 4 a 10 líneas



0

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | 1 | | X | |
| 01 | | | X | |
| 11 | | | X | X |
| 10 | | | X | X |

1

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | | | X | |
| 01 | 1 | | X | |
| 11 | | | X | X |
| 10 | | | X | X |

2

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | | | X | |
| 01 | | | X | |
| 11 | | | X | X |
| 10 | 1 | | X | X |

3

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | | | X | |
| 01 | | | X | |
| 11 | 1 | | X | X |
| 10 | | | X | X |

4

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | | 1 | X | |
| 01 | | | X | |
| 11 | | | X | X |
| 10 | | | X | X |

5

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | | | X | |
| 01 | | 1 | X | |
| 11 | | | X | X |
| 10 | | | X | X |

6

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | | | X | |
| 01 | | | X | |
| 11 | | | X | X |
| 10 | | 1 | X | X |

7

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | | | X | |
| 01 | | | X | |
| 11 | | 1 | X | X |
| 10 | | | X | X |

8

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | | | X | 1 |
| 01 | | | X | |
| 11 | | | X | X |
| 10 | | | X | X |

9

| | | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | | | X | |
| 01 | | | X | 1 |
| 11 | | | X | X |
| 10 | | | X | X |

0 (ABCD) = $\bar{A}\bar{B}\bar{C}\bar{D}$

1 (ABCD) = $\bar{A}\bar{B}\bar{C}D$

2 (ABCD) = $\bar{B}\bar{C}\bar{D}$

3 (ABCD) = $\bar{B}\bar{C}D$

4 (ABCD) = $\bar{B}C\bar{D}$

5 (ABCD) = $\bar{B}CD$

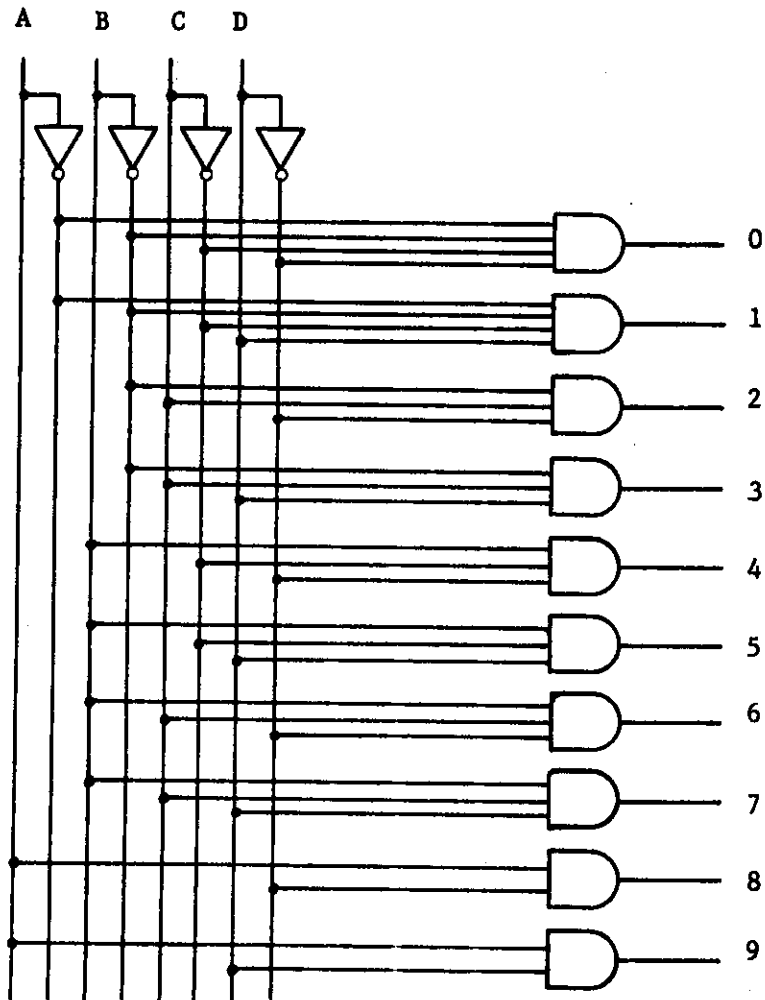
6 (ABCD) = $BC\bar{D}$

7 (ABCD) = BCD

8 (ABCD) = $A\bar{D}$

9 (ABCD) = AD

5. IMPLEMENTACION



Antes de que aparecieran los displays de 7 segmentos en el mercado se fabricaba un display encapsulado en un bulbo de cristal transparente. Cada dígito se configuraba por un filamento muy delgado. Todos los filamentos con forma de números del 0 al 9 estaban colocados uno detrás de otro. Si un filamento se iluminaba, opacaba a todos los demás, notándose claramente un dígito.

Un decodificador de 4 a 10 líneas puede servir como un decodificador de BCD a DECIMAL para manejar este tipo de display, como se muestra en la fig. 6.9.

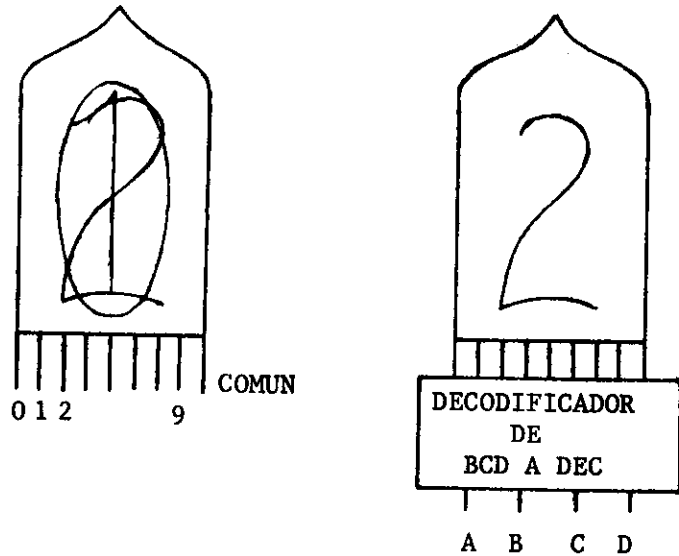


Fig. 6.9 a) DISPLAY DECIMAL, b) DECODIFICADOR DE BCD A DECIMAL MANEJANDO UN DISPLAY DECIMAL.

Ejemplo 6.7

Diseñe un circuito que efectúe la comparación en magnitud de 2 palabras binarias de un solo bit. Además que pueda ser expandido para comparar palabras de más bits.

- Un comparador completo. A diferencia de un medio comparador, es un bloque que puede conectarse en cascada para efectuar comparaciones multibit y que considera el resultado de la comparación de los bits anteriores. Como se muestra en la fig. 6.10.

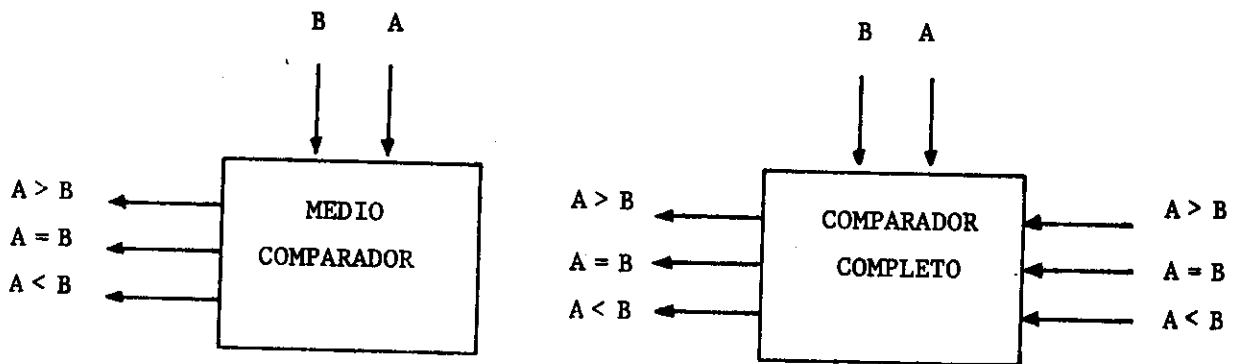


Fig. 6.10 a) MEDIO COMPARADOR, b) COMPARADOR COMPLETO

Como podemos observar en la fig. 6.10 b) el bloque tiene 5 entradas y 3 salidas. Para reducir el número de entradas es preferible sustituir las entradas a solamente 2, cuyas combinaciones formen un código que de los 3 valores de comparación, $A > B$, $A = B$, $A < B$. Fig. 6.11

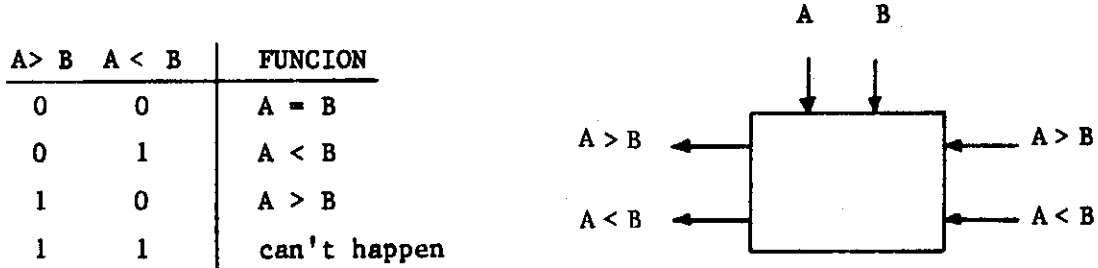
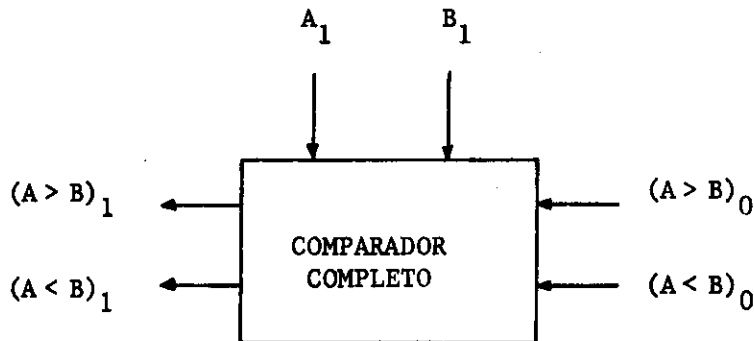


Fig. 6.11 LAS ENTRADAS Y SALIDAS A UN COMPARADOR PUEDEN REDUCIRSE.

Entonces el bloque a diseñar queda de la siguiente forma:



Los valores de $(A > B)_0$ y $(A < B)_0$ son el resultado de la comparación del bit anterior.

| $(A > B)_0$ | $(A < B)_0$ | $A_1 B_1$ | $(A > B)_1$ | $(A < B)_1$ |
|-------------|-------------|-----------|-------------|-------------|
| 0 | 0 | 0 0 | 0 | 0 |
| 0 | 0 | 0 1 | 0 | 1 |
| 0 | 0 | 1 0 | 1 | 0 |
| 0 | 0 | 1 1 | 0 | 0 |
| 0 | 1 | 0 0 | 0 | 1 |
| 0 | 1 | 0 1 | 0 | 1 |
| 0 | 1 | 1 0 | 1 | 0 |
| 0 | 1 | 1 1 | 0 | 1 |
| 1 | 0 | 0 0 | 1 | 0 |
| 1 | 0 | 0 1 | 0 | 1 |
| 1 | 0 | 1 0 | 1 | 0 |
| 1 | 0 | 1 1 | 1 | 0 |
| 1 | 1 | 0 0 | X | X |
| 1 | 1 | 0 1 | X | X |
| 1 | 1 | 1 0 | X | X |
| 1 | 1 | 1 1 | X | X |

$A > B_1$

| | 00 | 01 | 11 | 10 |
|----|----|----|----|----|
| 00 | 0 | 0 | X | 1 |
| 01 | 0 | 0 | X | 0 |
| 11 | 0 | 0 | X | 1 |
| 10 | 1 | 1 | X | 1 |

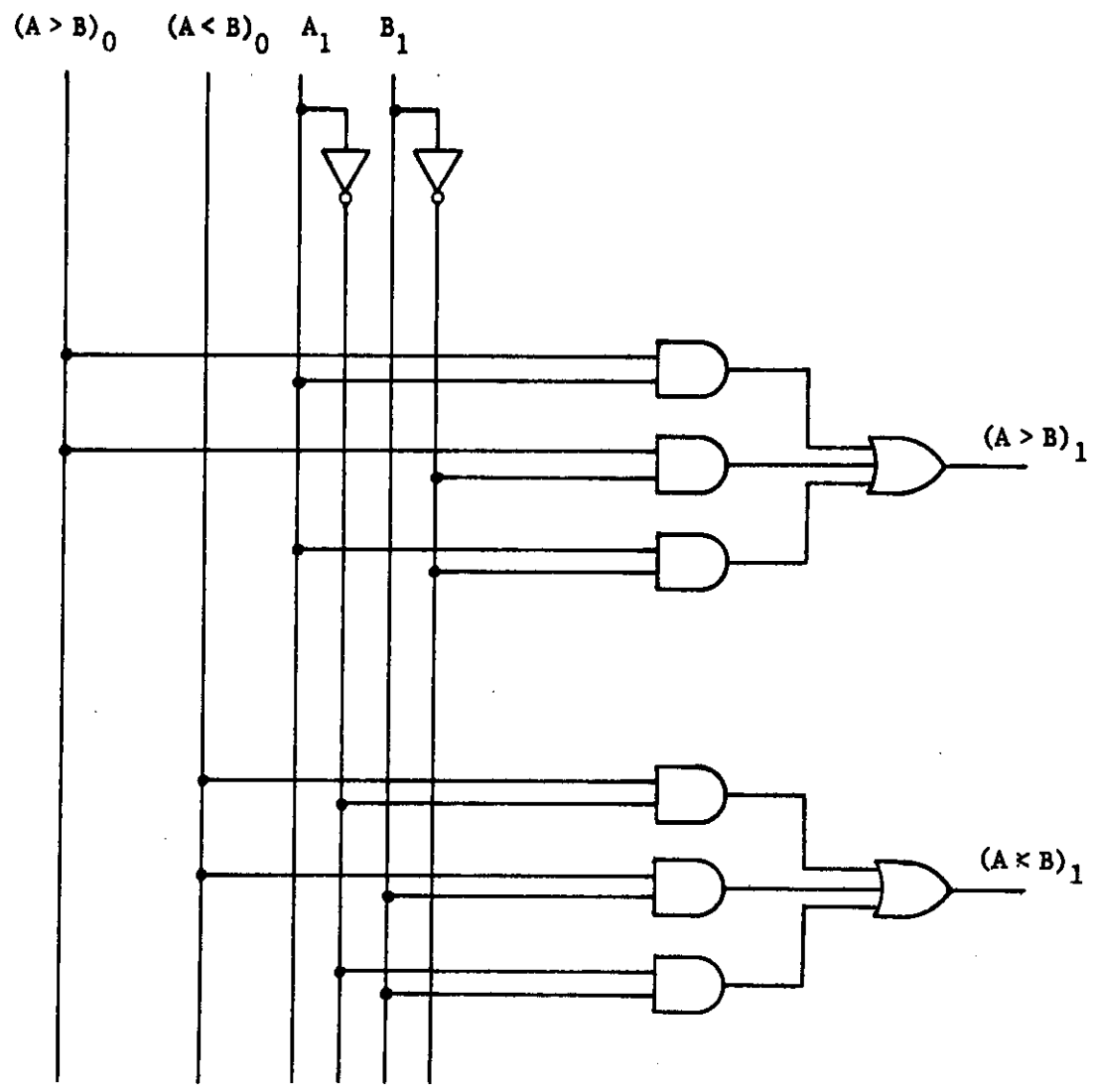
$A < B_1$

| | 00 | 01 | 11 | 10 |
|----|----|----|----|----|
| 00 | 0 | 1 | X | 0 |
| 01 | 1 | 1 | X | 1 |
| 11 | 0 | 1 | X | 0 |
| 10 | 0 | 0 | X | 0 |

$$(A > B)_1 = (A > B)_0 \cdot \bar{B}_1 + (A > B)_0 \cdot A_1 + A_1 \bar{B}_1$$

$$(A < B)_1 = \bar{A}_1 B_1 + (A < B)_0 \cdot \bar{A}_1 + (A < B)_0 \cdot B_1$$

5. IMPLEMENTACION



6.6 SISTEMAS COMBINACIONALES CON SALIDAS MÚLTIPLES

En los ejemplos de diseño combinacional que hemos tratado en los -- puntos anteriores de este mismo capítulo, observamos que existen bloques con varias salidas que se usan en forma simultánea, es decir sistemas -- constituidos por varias funciones lógicas que dependen de las mismas variables booleanas de entrada. A estos bloques se les da el nombre de -- sistemas combinacionales con salidas múltiples.

La minimización de una función múltiple se puede efectuar tratando -- cada una de las funciones en forma independiente como en los ejemplos -- del punto 6.2, sin embargo no se tiene la completa seguridad de obtener -- la forma más simple del circuito.

Un método sencillo de minimización consiste en -- comparar los mapas de Karnaugh de todas las funciones e identificar los grupos que sean comunes a más de una función. Estos términos comunes o repetidos tienen que ser implementados una sola vez.

Una de las precauciones que es necesario considerar es que la salida de una compuerta puede conectarse a un número limitado de entradas -- (Fan-out, ver terminología de los circuitos integrados, apéndice B).

Pasos para la minimización de funciones múltiples por medio del mapa de Karnaugh.

- 1.- Configurar el mapa de Karnaugh para cada función.
- 2.- Buscar el grupo más pequeño que aparece en cada uno de los mapas e -- indicarlo con un círculo.
- 3.- Continúa el proceso de agrupación partiendo del grupo más pequeño a -- el más grande.
- 4.- Al seleccionar un grupo considerar su utilidad de acuerdo a que sea -- común a la mayoría de las funciones.
- 5.- Seleccionar el mejor juego de grupos comunes a cada función.

Ejemplo 6.8

Implemente la función lógica que se presenta a continuación. a) En forma independiente, b) Como una función múltiple.

$$F_1(ABCD) = \sum 1, 4, 5, 6, 7, 8, 9, 13$$

$$F_2(ABCD) = \sum 6, 7$$

$$F_3(ABCD) = \sum 4, 5, 8, 9$$

a) En forma independiente

F1 AB

| CD | 00 | 01 | 11 | 10 |
|----|----|----|----|----|
| 00 | | 1 | | 1 |
| 01 | 1 | 1 | 1 | 1 |
| 11 | | 1 | | |
| 10 | | 1 | | |

$$F1 = \bar{A}B + \bar{C}D + A\bar{B}\bar{C}$$

F2 AB

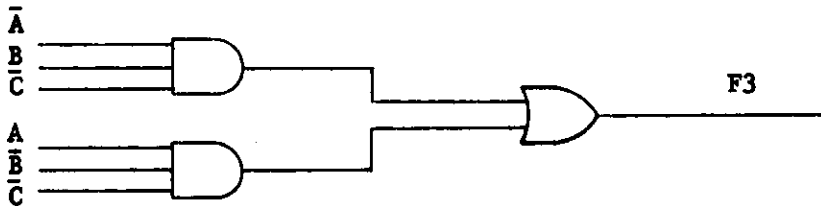
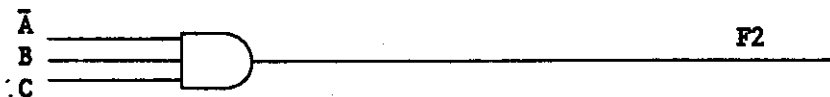
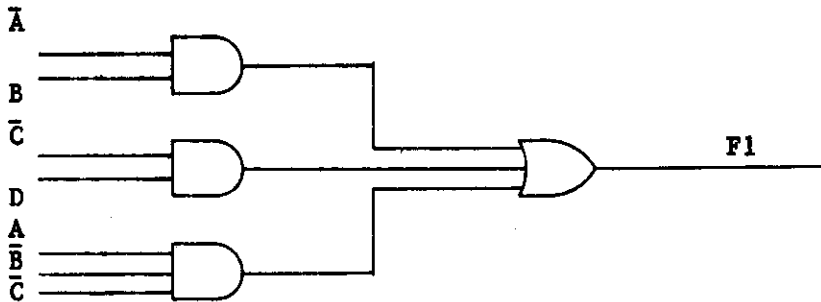
| CD | 00 | 01 | 11 | 10 |
|----|----|----|----|----|
| 00 | | | | |
| 01 | | | | |
| 11 | | 1 | | |
| 10 | | 1 | | |

$$F2 = \bar{A}BC$$

F3 AB

| CD | 00 | 01 | 11 | 10 |
|----|----|----|----|----|
| 00 | | 1 | | 1 |
| 01 | | 1 | | 1 |
| 11 | | | | |
| 10 | | | | |

$$F3 = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C}$$



Sin tomar en cuenta los NOT'S de entrada el costo unitario $CU = 21$

b) Como una Función Múltiple:

| F1 | AB | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| CD | | 1 | | 1 |
| 00 | | 1 | | 1 |
| 01 | 1 | 1 | 1 | 1 |
| 11 | | 1 | | |
| 10 | | 1 | | |

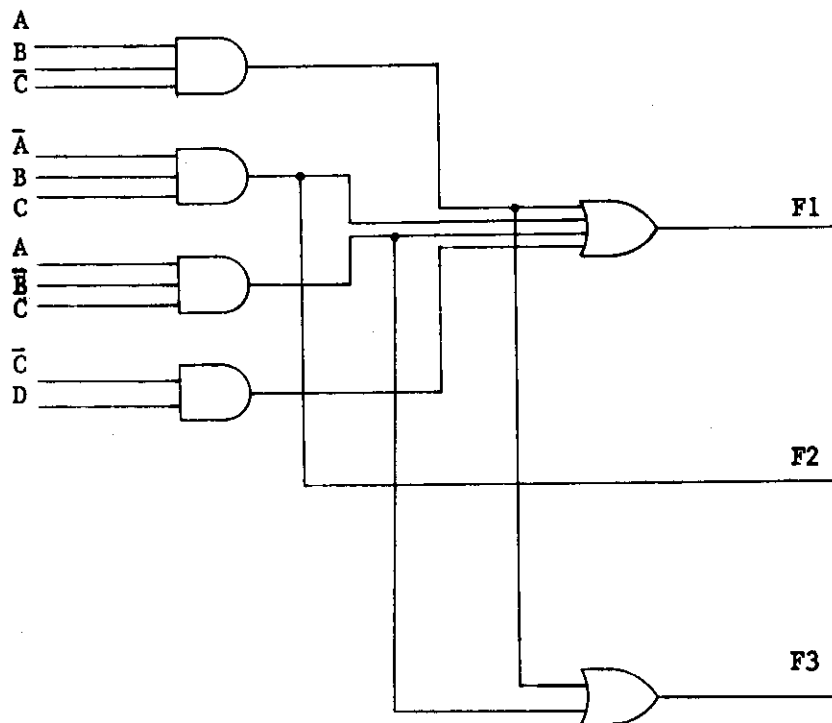
| F2 | AB | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| CD | | | | |
| 00 | | | | |
| 01 | | | | |
| 11 | | 1 | | |
| 10 | | 1 | | |

| F3 | AB | | | |
|----|----|----|----|----|
| | 00 | 01 | 11 | 10 |
| CD | | 1 | | 1 |
| 00 | | 1 | | 1 |
| 01 | | 1 | | 1 |
| 11 | | | | |
| 10 | | | | |

$$F1 = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + \bar{C}D$$

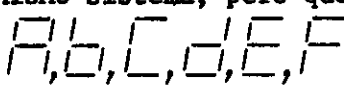
$$F2 = \bar{A}BC.*$$

$$F3 = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C}.*$$



Sin tomar en cuenta los NOT'S de entrada el costo unitario CU = 17

PROBLEMAS PROPUESTOS

- 1.- Defina los Sistemas Completamente Específicos.
- 2.-¿Que es el DON'T CARE?
- 3.-¿Que es el CAN'T HAPPEN?
- 4.- Defina los sistemas que no están completamente especificados.
- 5.-¿Para que nos sirve un DISPLAY?
- 6.-¿Que son los Decodificadores?
- 7.- Diseñe un circuito combinacional que desarrolle la multiplicación binaria de 2 palabras de 2 bits. La palabra A, (A_1A_0) y la palabra B, (B_1B_0) . El resultado o salida del circuito marcarlo con la letra M $(M_3M_2M_1M_0)$.
- 8.- Diseñar un circuito combinacional al que lleguen 4 líneas de entrada-codificadas en código binario natural y cuya salida esté codificada - en BCD.
- 9.- Detector de errores en BCD. Diseñar un circuito al que lleguen 4 líneas de entrada e indique cuando alguna de las combinaciones no sea - BCD.
- 10.- Cuando se requiere extrema confianza en el control de algún proceso,- se usan 2 o 3 sistemas de control que operen simultáneamente. Tal -- es el caso de un sistema que opera por triplicado, nuestro interés -- es que cuando menos 2 de los 3 sistemas operen satisfactoriamente. -- Por lo tanto se requiere señalar la confiabilidad del sistema de -- control, por medio de una sola salida.
- 11.- En el ejemplo 6.4 diseño de un decodificador de BCD a 7 segmentos, -- las combinaciones de 10 a 15 se tomaron como CAN'T HAPPEN.
 - a) Diseñe el mismo sistema con la variante de que una combinación que no sea BCD muestre una E de ERROR.
 - b) Diseñe el MISMO sistema, pero que muestre además los siguientes -- caracteres  para las combinaciones del 10 al 15.
- 12.- a) Diseñe un circuito que tenga 4 entradas I_3, I_2, I_1, I_0 y 2 salidas O_1, O_0 . El estado de las salidas mostrará cual línea de entrada - tiene en uno lógico, es decir la salida es un código en binario -- natural correspondiente a cada una de las entradas.
 - b) Diseñe el mismo circuito tomando en cuenta que pueden presentarse

unos en todas las entradas a la vez. En este caso la salida será un código que represente a la entrada de mayor peso. Este circuito recibe el nombre de codificador de prioridad.

- 13.- Diseñe un circuito combinacional que convierta código gray de 4 - - bits a código binario natural.

7

Flip - Flops

7.0 INTRODUCCION

Los circuitos digitales discutidos hasta ahora han sido circuitos - combinacionales cuyos niveles de salida en cualquier instante de tiempo - dependen de los niveles que tienen las entradas en ese momento. Cual- - quier entrada anterior no tiene efecto en las salidas actuales, por este motivo se dice que los circuitos combinacionales no tienen MEMORIA. Una gran cantidad de bloques digitales están contruidos de una parte combi- nacional y otra parte de memoria. Este tipo de bloques reciben el nom- bre de CIRCUITOS SECUENCIALES.

Un circuito secuencial se define como un bloque digital cuyas sali- das dependan no únicamente de sus entradas actuales sino también de en- tradas anteriores. En la fig. 7.0 aparece la representación de un siste- ma secuencial.

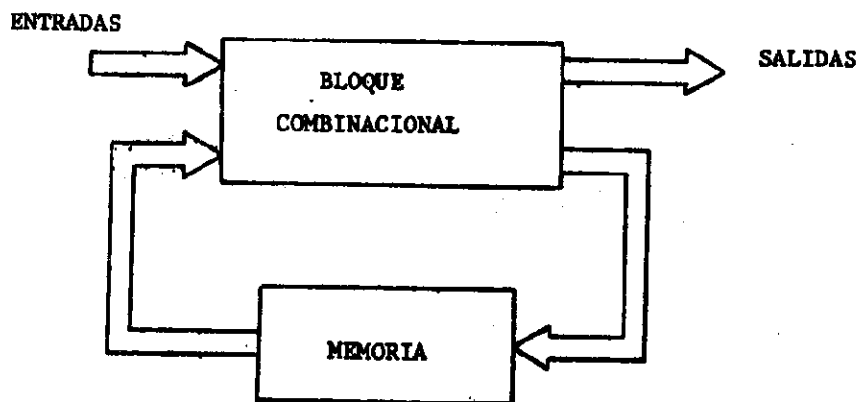


Fig. 7.0 REPRESENTACION DE UN SISTEMA SECUENCIAL GENERALIZADO

Las entradas que ya habían entrado se almacenan en un bloque de memoria y se retroalimentan a la entrada del bloque combinacional para formar parte de la expresión booleana de la salida.

A diferencia de un sistema puramente combinacional, el sistema secuencial tiene una retroalimentación entre la entrada y la salida. Además el bloque de memoria tiene la capacidad de mantener información, de tal forma que los valores de entrada y salida previos puedan usarse para determinar las salidas actuales y futuras del sistema.

Los sistemas secuenciales tienen muy variadas aplicaciones, pueden usarse como elementos de memoria, por ejemplo los registros de una calculadora o un computador digital, y sobre todo donde sea necesario tener noción de condiciones anteriores, por ejemplo los contadores digitales cuyo uso se extiende desde aplicaciones industriales de conteo de piezas o productos hasta los populares relojes digitales, también podemos considerar a los circuitos generadores de secuencia para la automatización de máquinas numéricas, secuenciadores para controladores de tráfico, etc.

7.1 DIAGRAMAS DE TIEMPO

Frecuentemente es necesario efectuar análisis de circuitos secuenciales, para este propósito un diagrama de tiempo es una herramienta muy útil, consiste de una gráfica de voltaje con respecto a tiempo.

En la fig. 7.1 se muestra un diagrama de tiempo para un circuito formado por una compuerta AND y un circuito de retardo de t_d segundo. La entrada A consiste de un pulso con una duración de $2t_d$ segundos. El circuito de retardo tiene una salida B exactamente igual a la entrada con la diferencia de que se presenta con un retardo de t_d segundo. La salida de la compuerta AND debe ser igual a 1 durante el intervalo de tiempo en el cual A y B son 1.

Considerando un pequeño tiempo de propagación de la compuerta t_p , la salida aparecerá como se muestra en el diagrama.

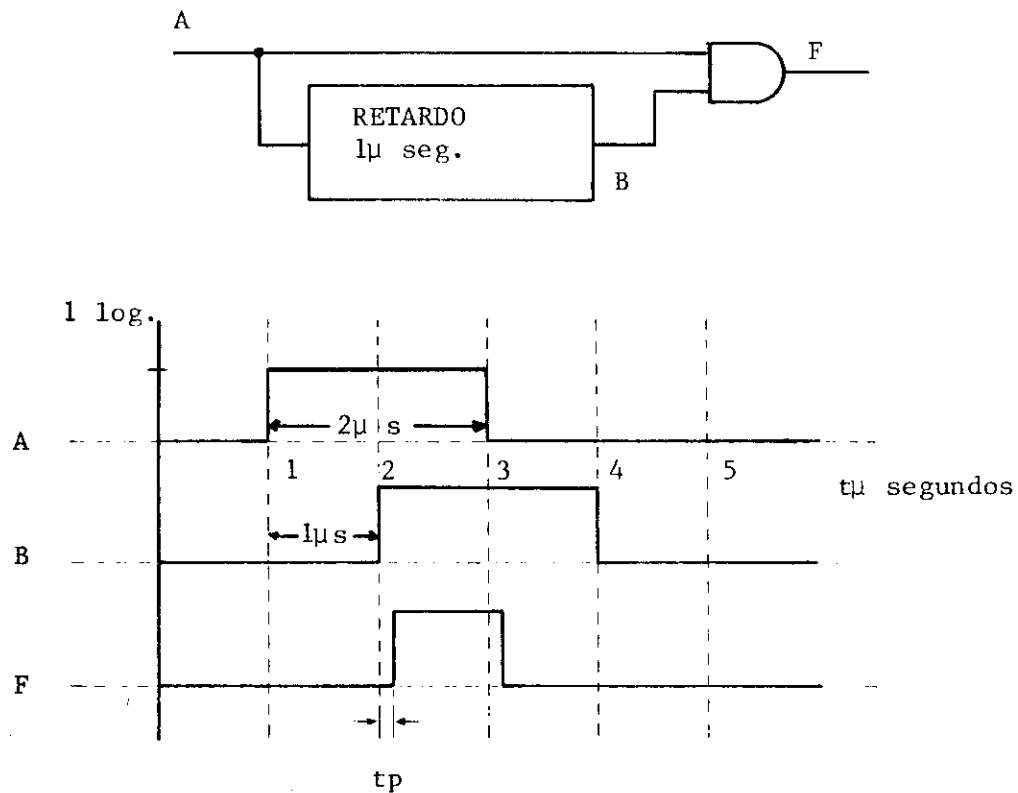


Fig. 7.1 DIAGRAMA DE TIEMPO PARA UN CIRCUITO FORMADO POR UNA COMPUERTA AND Y UN BLOQUE DE RETARDO.

7.2 FLIP-FLOPS IMPLEMENTADOS CON COMPUERTAS

Uno de los elementos de memoria más simple es el flip-flop abreviado FF, es un circuito lógico con dos salidas, una afirmada y la otra negada, generalmente marcadas como Q y \bar{Q} , las entradas a un FF se les llama entradas de control y posteriormente discutiremos su cantidad y tipo.

Cuando hablamos de la salida de un FF nos referimos siempre a la Q afirmada. Al valor (0 o 1) de la salida (Q) a diferencia de una compuerta no se le llama nivel sino ESTADO. Entonces podemos decir que un estado es un NIVEL almacenado en una memoria (FF).

Con las aclaraciones anteriores podemos definir a un flip-flop como un circuito lógico con dos estados estables (0 y 1) en el cual las **transiciones** entre esos dos estados pueden efectuarse por medio de sus entradas de control y datos. En la fig. 7.2 se muestra el símbolo para un FF.

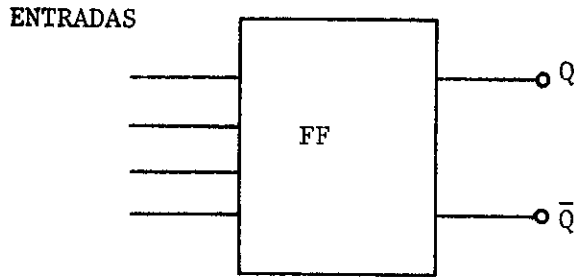


Fig. 7.2 SIMBOLO PARA UN FLIP - FLOP

El circuito para un flip-flop básico puede implementarse con un par de compuertas NOR interconectadas como se muestra en la fig. 7.3 a).

Obsérvese que las entradas aparecen marcadas como R y S y las salidas como Q y \bar{Q} . Este FLIP-FLOP recibe el nombre de FF set-reset.

SET = Establecer, significa colocar un uno lógico en Q.

RESET = Reestablecer, significa colocar un 0 lógico en Q, también se usa el término RESET para indicar regreso al estado original o de inicio.

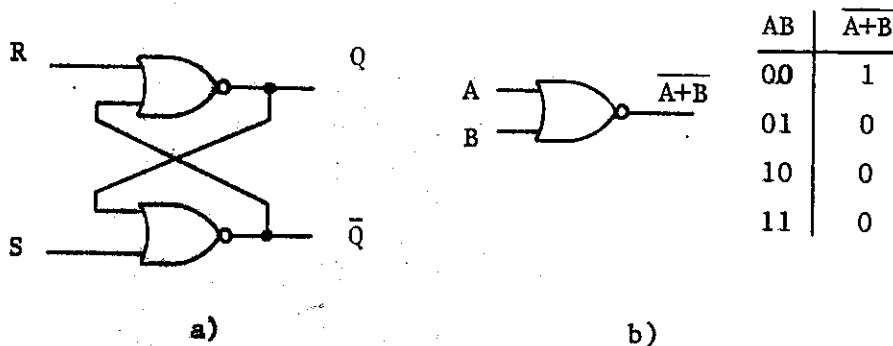
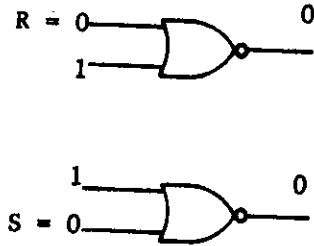


Fig. 7.3 a) FLIP-FLOP BASICO IMPLEMENTADO CON COMPUERTAS NOR,
b) COMPUERTA NOR Y SU TABLA DE VERDAD.

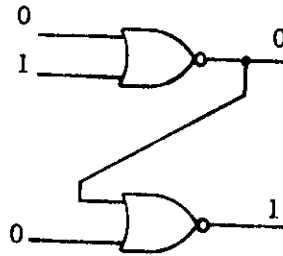
De la tabla de verdad que aparece en la fig. 7.3b) podemos deducir - que con el simple hecho de que una de las entradas al NOR sea "1", la salida será incondicionalmente "0" (sin importar lo que haya en la otra entrada).

Analizamos por pasos el circuito de la fig. 7.3a) antes de ser interconectado.

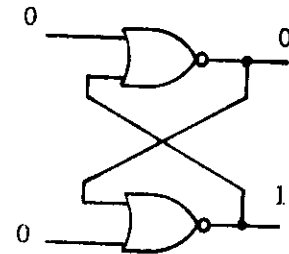
PASO 1



PASO 2



PASO 3



PASO 1

Las entradas R y S se colocan a un nivel de 0 lógico y las entradas restantes se dejan a un uno lógico. Con esto se logra que la salida para cada NOR sea igual a un 0 lógico.

PASO 2

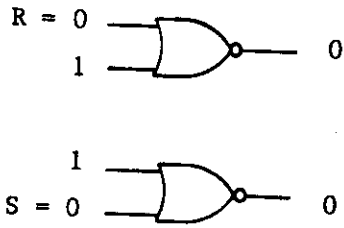
Conservando R y S en cero, se conecta la salida de la compuerta superior a la entrada libre de la compuerta inferior, cuya salida cambia a 1.

PASO 3

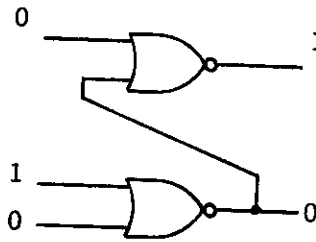
La salida de la compuerta inferior con un (1 lógico) se conecta a la entrada libre de la compuerta superior (que también tiene como entrada un 1 lógico) de tal forma que la salida permanece sin cambiar (en 0 lógico).

Al concluir el paso 3 el FF se encuentra en el estado 0 (cero) -- $Q = 0, \bar{Q} = 1$ que es uno de sus estados estables. El estado 1 se deduce similarmente invirtiendo el orden de la conexión.

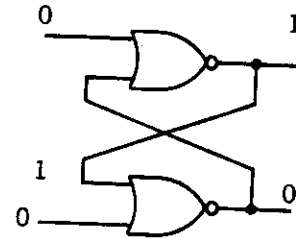
PASO 1



PASO 2



PASO 3



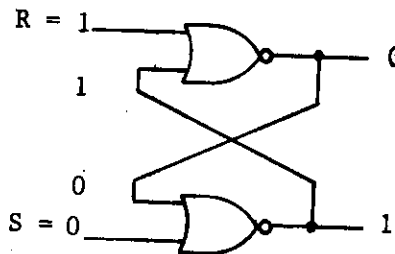
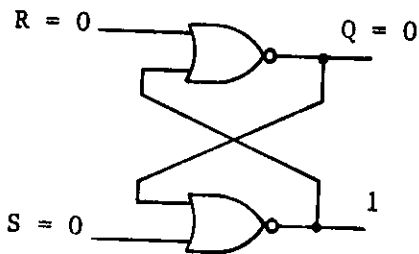
Para continuar con el análisis del FF RS implementado con NOR'S usaremos una tabla donde se muestren todas las combinaciones de entrada y las salidas para cada combinación. Para $R = 0$ y $S = 0$ la salida toma dos diferentes valores en Q y \bar{Q} es decir los dos estados estables (0 y 1).

| RS | $Q\bar{Q}$ |
|----|------------|
| 00 | 01 |
| | 10 |
| 01 | |
| 10 | |
| 11 | |

Obviamente las combinaciones restantes de R y S causan cambios en Q y \bar{Q} . La comprobación de estos cambios se efectúa a partir de ciertos estados determinados como se muestra en los pasos 4, 5 y 6.

PASO 4

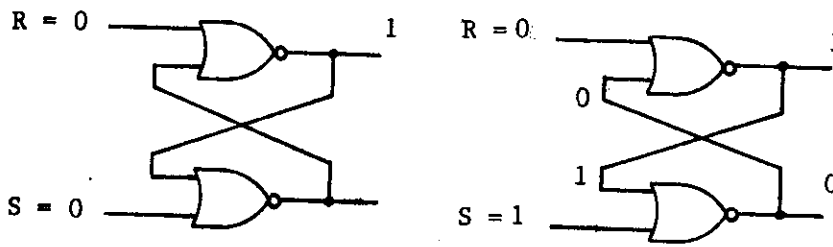
A partir del estado cero



| SR | $Q\bar{Q}$ |
|------|--|
| 00 | $\begin{bmatrix} 01 \\ 10 \end{bmatrix}$ |
| * 01 | 01 |
| 10 | |
| 11 | |

PASO 5

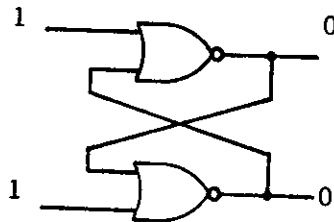
A partir del estado 1



| SR | Q \bar{Q} |
|------|-------------|
| 00 | 01 |
| | 10 |
| 01 | 01 |
| * 10 | 10 |
| 11 | |

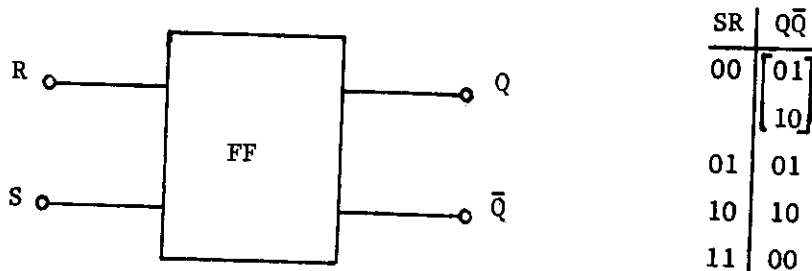
PASO 6

A partir de cualquier estado



| SR | Q \bar{Q} |
|------|-------------|
| 00 | 01 |
| | 10 |
| 01 | 01 |
| 10 | 10 |
| * 11 | 00 |

Al estar el FLIP-FLOP en el estado "00" para entradas $R = 1$ y $S = 1$ y cambiar simultáneamente las entradas R y S a "0" puede caer en el estado "01" o "10" indistintamente. Se dice que el circuito de la fig. 7.3.a. FF implementado con NORs tiene MEMORIA por que el cambio de estado no depende únicamente de las entradas actuales sino también de la secuencia anterior de las entradas. En la fig. 7.4 se muestra el símbolo para el FF RS y su tabla de estados.



| SR | Q \bar{Q} |
|----|-------------|
| 00 | 01 |
| | 10 |
| 01 | 01 |
| 10 | 10 |
| 11 | 00 |

Fig. 7.4 SIMBOLO PARA UN FLIP - FLOP RS

En la fig. 7.5 aparece el diagrama de tiempo para el FF RS. Nótese que cuando S cambia a 1 en t_1 , Q cambia un instante después. (tpff que es el tiempo de propagación del FF). Cuando S regresa a 0 (t_2), Q permanece en 1 (no cambia). Cuando R cambia a 1 (t_3) Q cambia a 0 un tpff - después. Los tiempos de duración para R o S deben ser más grandes que los tpff.

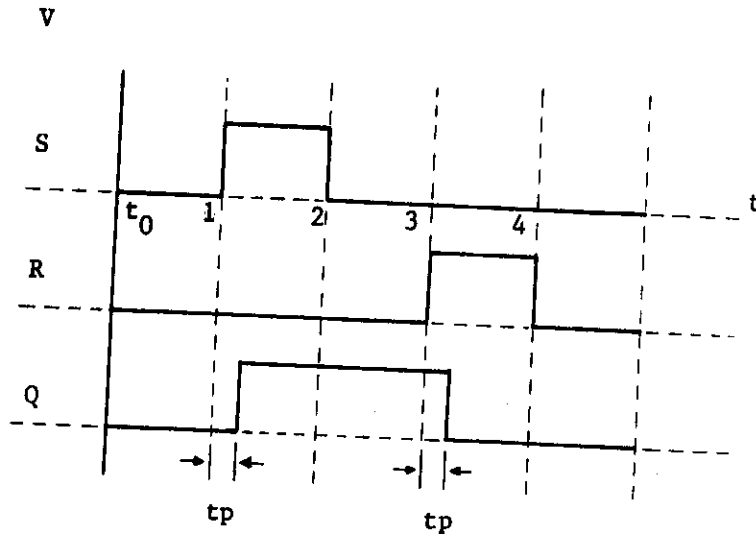


Fig. 7.5 DIAGRAMA DE TIEMPO PARA UN FF - RS

Es muy frecuente encontrar un FLIP-FLOP RS implementado por medio de compuertas NAND. La operación de este FF puede analizarse de la misma manera empleada para el FF implementado con compuertas NOR. En la fig. 7.6 se muestra un FLIP-FLOP y su tabla de estados. Puede notarse que su funcionamiento es igual al FF de compuertas NOR excepto que responde a niveles bajos en las entradas R y S mientras que los niveles altos no tienen efecto alguno.

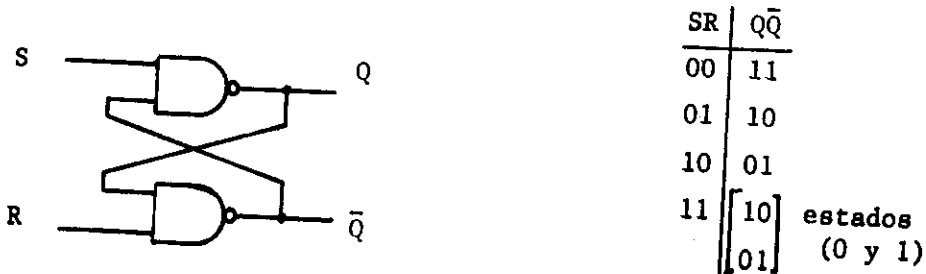


Fig. 7.6 FLIP-FLOP RS IMPLEMENTADO CON NAND Y SU TABLA DE ESTADOS

7.3 FLIP-FLOPS CON RELOJ

Existe una tercer entrada de control a un FF llamada RELOJ, con este nombre se denomina a una señal de onda cuadrada positiva, generalmente simétrica en período y con una magnitud mínima de un cero lógico y -- máxima de un uno lógico.

La señal de reloj permite al FF cambiar de estado. Estos cambios-- de estado pueden efectuarse en la TRANSICION POSITIVA del reloj (cuando cambia de 0 a 1) o en la TRANSICION NEGATIVA (cuando cambia de 1 a 0) en la Fig. 7.7 se observan 2 señales de reloj, no todas las señales de reloj son simétricas.

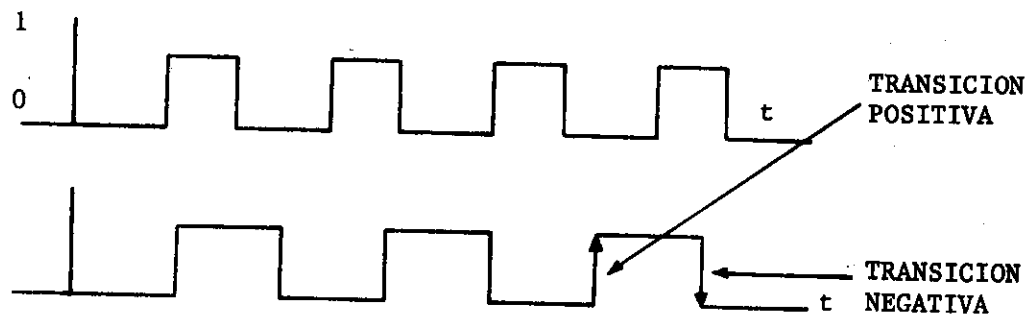


Fig. 7.7 SEÑALES DE RELOJ

A menudo se presentan sistemas secuenciales configurados por varios FLIP-FLOPS que operan como SISTEMAS SECUENCIALES SINCRONOS, el término -- síncrono se refiere a un sistema donde la secuencia de transiciones en -- los diferentes FLIP-FLOPS está controlada por una misma señal de reloj.-- Por otro lado los sistemas secuenciales que no tienen las característi-- cas discutidas anteriormente se les llama SISTEMAS ASINCRONOS.

La señal de reloj usualmente se representa por las iniciales del -- término en inglés CK (CLOCK).

7.4 FLIP-FLOP RS CON RELOJ

En la fig. 7.8 se muestra un FF construido con NOR'S al que se le han agregado un par de compuertas AND. Aparece además de R y S una tercer entrada de control CK (reloj).

Cuando la señal de reloj CK es igual a CERO se asegura un nivel de 0 log. en la salida de cada compuerta AND, esto condiciona al FF a permanecer en alguno de los dos estados estables (0 o 1). Cuando la señal de reloj es igual a UNO los valores de R y S pasan a las entradas del FF y lo cambian de estado (dependiendo del valor R y S). Cuando $R = 1$ y $S = 1$ provocan un estado indefinido al bajar la señal de reloj CK.

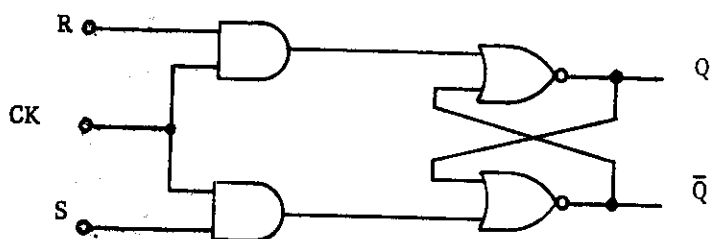


Fig. 7.8 FLIP-FLOP RS CON RELOJ

Para representar el comportamiento de un FF con reloj se usa una tabla llamada TABLA CARACTERISTICA. Aquí aparecen nuevas nomenclaturas para las Q. El valor de Q antes de que se presente el pulso de reloj se llama Q_n o ESTADO ACTUAL y al valor de Q después del pulso de reloj se le llama Q_{n+1} o estado próximo. Por ejemplo si $R = 0$ y $S = 0$ y se presenta un pulso de reloj, el próximo estado es el mismo, $Q_{n+1} = Q_n$, como se muestra en la tabla de la fig. 7.9.

| RS | Q_{n+1} |
|----|-----------|
| 00 | Q_n |
| 01 | 1 |
| 10 | 0 |
| 11 | ? |

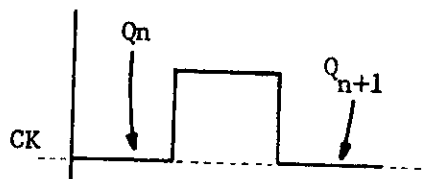


Fig. 7.9 TABLA CARACTERISTICA PARA UN FF RS.

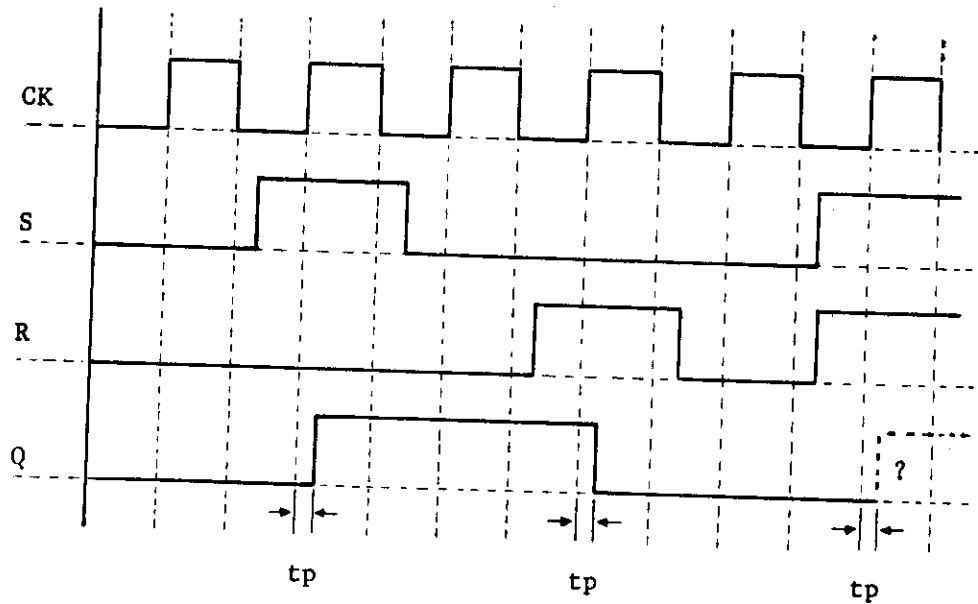


Fig. 7.10 DIAGRAMA DE TIEMPO PARA UN FF RS CON RELOJ

7.5 FF JK

El estado indefinido del FLIP-FLOP RS de la fig. 7.8 puede eliminarse usando una retroalimentación adicional como se muestra en la fig. 7.11. La salida Q regresa a través de una compuerta AND de tres entradas a la entrada R del FF y lo mismo sucede con \bar{Q} . Las entradas libres a las compuertas AND se marcan como J y K. La operación del circuito aparece graficada en la fig. 7.12. Cuando J y K son 0, R y S permanecen en 0 y los pulsos de reloj no cambian el estado del FF. Cuando el estado inicial del FF es $Q = 0$, la salida Q retroalimentada a la entrada K inhibe el cambio de esa compuerta cuya salida permanece en 0. Por lo tanto una entrada de $K = 1$ servirá únicamente para restablecer el FF (o sea $Q = 0$). En este instante una entrada en $J = 1$ puede lograr una $Q = 1$. Cuando el estado inicial es igual a 1 el FF puede reestablecerse por medio de la entrada K. Cuando J y K son 1 se presenta una característica especial, si el estado inicial es $Q = 0$ la K se encuentra inhibida por $Q = 0$ por lo tanto el FF no puede reestablecerse, sin embargo como $\bar{Q} = 1$ y $J = 1$ el FF cambia al estado 1. Cuando se presenta un nuevo pulso de reloj el estado próximo será el complemento del estado actual. Si el pulso de reloj es -

más corto que los tiempos de retardo t_p o mucho más grande que la suma de estos el FF no operará correctamente.

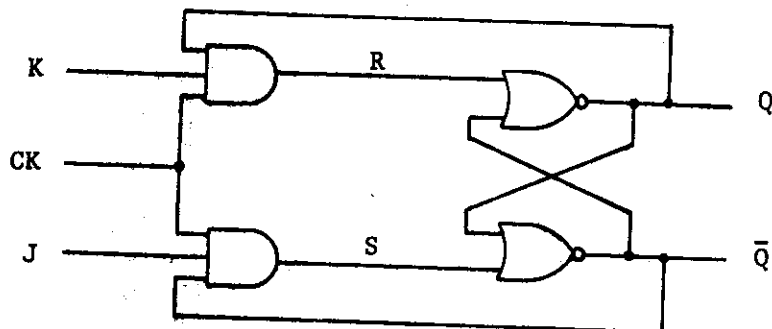


Fig. 7.11 FLIP - FLOP JK

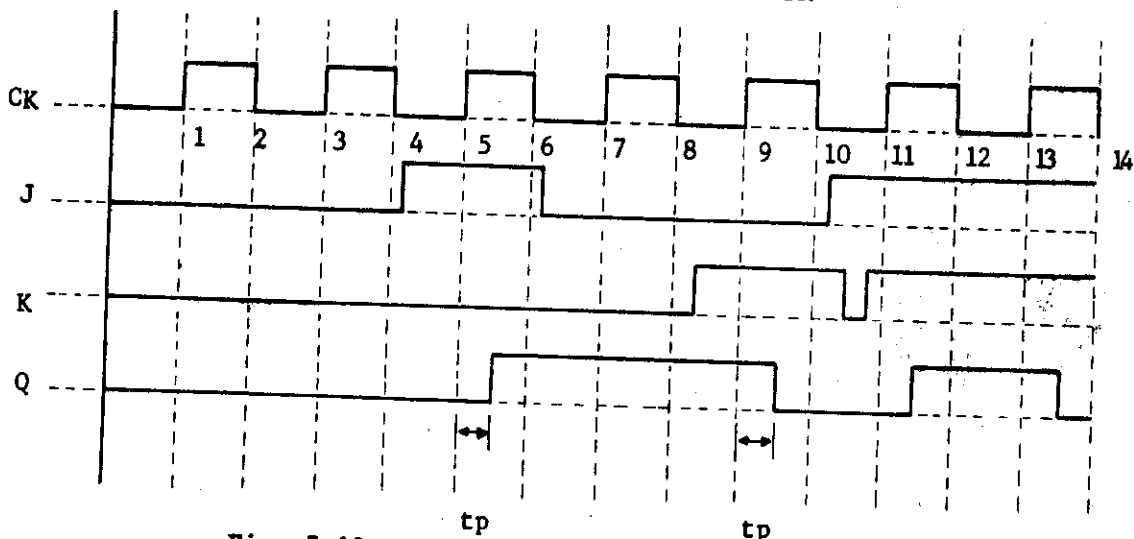


Fig. 7.12 DIAGRAMA DE TIEMPOS PARA UN FF JK

7.6 FLIP-FLOP JK DISPARADO EN EL FILO DE UNA TRANSICIÓN DE RELOJ

Cuando la operación del FF JK depende de un pulso de reloj de un ancho específico, se puede usar un circuito que recorte el pulso de reloj-- tal como se muestra en la fig. 7.13. La señal de reloj se aplica a una compuerta AND de 2 entradas. La otra entrada del AND es la misma señal - del reloj invertida y retrasada 3 tiempos de propagación t_p . La opera--- ción del circuito se muestra en la fig. 7.14

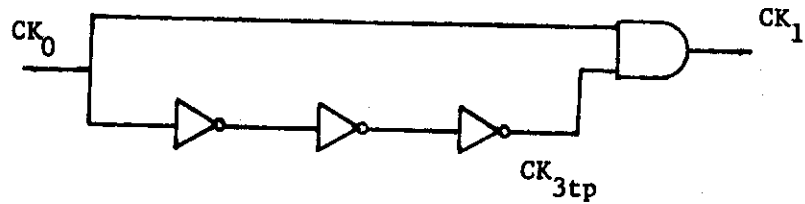


Fig. 7.13 CIRCUITO RECORTADOR DE PULSO DE CK

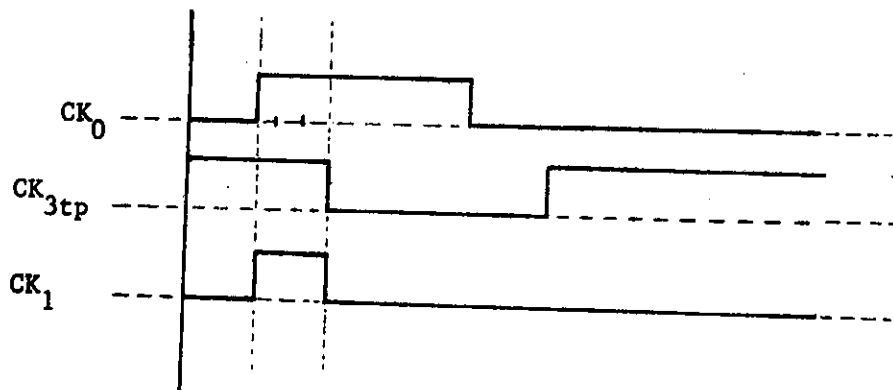


Fig. 7.14 DIAGRAMA DE TIEMPO PARA ILUSTRAR LA OPERACION DEL RECORTADOR DE PULSO.

El FLIP-FLOP JK con un circuito de entrada de reloj cuyo comportamiento sea semejante al de la fig. 7.13 se le conoce como FLIP-FLOP JK -- con disparo en el filo de la transición del CK. En la fig. 7.15 aparece su símbolo y su tabla caracterfística.

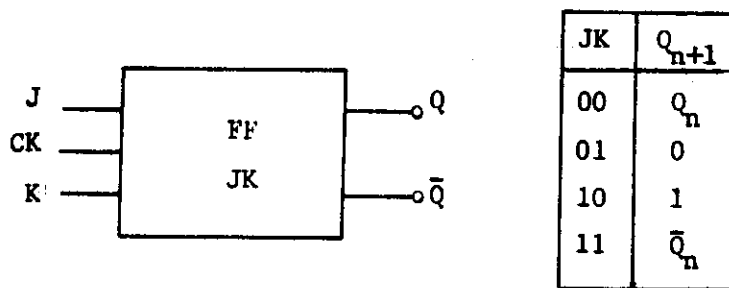


Fig. 7.15 SIMBOLO Y TABLA CARACTERISTICA PARA UN FF JK

7.7 FLIP-FLOP JK MAESTRO ESCLAVO

El circuito recortador de CK tiene un pequeño defecto, cuando se presentan señales de reloj lentas con respecto al tiempo, con un tiempo de subida de la pendiente, promedio de 10 tpd, simplemente no habría salida del circuito de la fig. 7.13.

El problema de recibir una señal de CK lenta puede superarse con el uso del un FLIP-FLOP MAESTRO-ESCLAVO. Este circuito incorpora 2 FF RS y con un juego de compuertas como se muestra en la fig. 7.16.

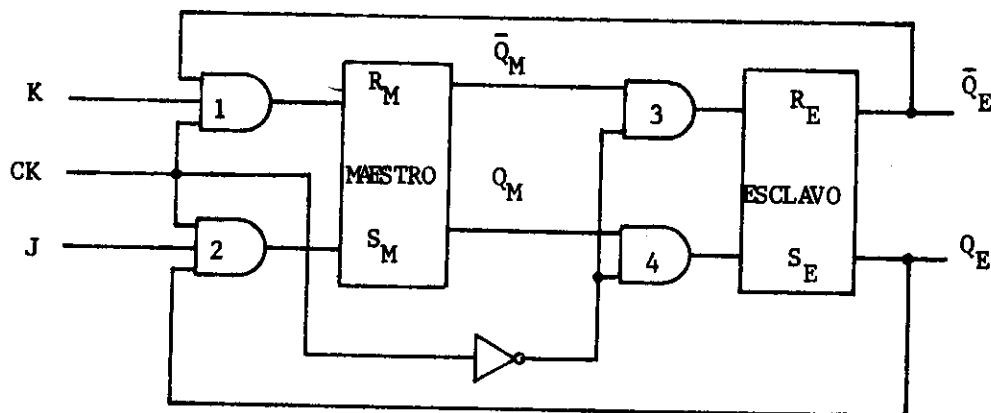


Fig. 7.16 FLIP-FLOP JK MAESTRO-ESCLAVO

La fig. 7.17 describe detalladamente el diagrama de tiempo para un FF JK MAESTRO ESCLAVO. A grandes rasgos la explicación es como sigue: - partiendo de $CK = 0$, al subir el pulso de CK las compuertas 3 y 4 aíslan las entradas del FF ESCLAVO en otras palabras R_E y S_E son 0 y Q_m y \bar{Q}_m no pasan a través de las compuertas por lo tanto el FF ESCLAVO no cambia de estado. En este instante las entradas en J y K condicionan el estado -- del FF MAESTRO. Al bajar el pulso de reloj las compuertas 1 y 2 aíslan las entradas R_M y S_M obligando al FF MAESTRO a permanecer en el mismo estado. A su vez las compuertas 3 y 4 permiten el paso de Q_m y \bar{Q}_m a R_E y S_E , entonces el FF ESCLAVO copia el estado del FF MAESTRO

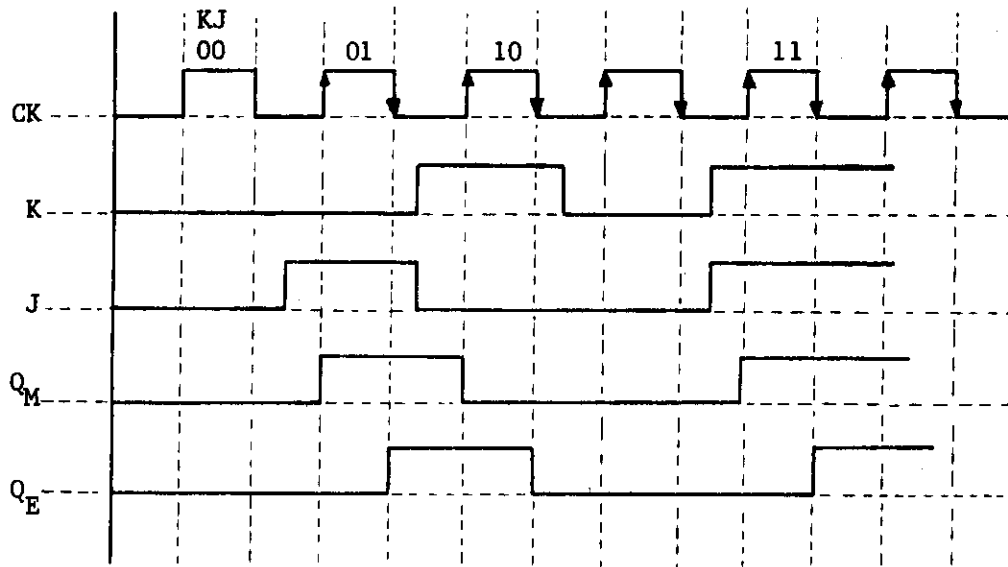


Fig. 7.17 DIAGRAMAS DE TIEMPO PARA UN FF JK MAESTRO ESCLAVO

7.8 CARACTERISTICAS DE LOS FLIP-FLOPS

En los puntos anteriores de este capítulo hemos trabajado con FLIP-FLOPS RS y JK y también hemos descrito su funcionamiento a partir de una TABLA CARACTERISTICA y un DIAGRAMA DE TIEMPOS. Sin embargo, existen otros dos tipos de FF, el FF-D y el FFT, también existen otras tablas que incrementan la cantidad de información de un FF determinado para su aplicación. Además de la tabla característica existen la TABLA DE EXCITACION y LA TABLA DE ESTADOS.

TABLA CARACTERISTICA

Como lo hemos visto anteriormente la tabla característica es un listado de las entradas a un FF, para cada combinación existe una salida (en este caso el estado próximo Q_{n+1}). Tomando como ejemplo la tabla del FF - JK se lee de la siguiente forma: para una entrada de J y K la salida es Q_{n+1} .

TABLA DE ESTADOS

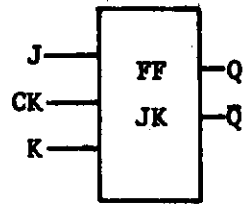
La tabla de estados es un listado de las combinaciones del estado presente Q_n y las entradas al FF, para cada una de estas combinaciones existe un estado próximo Q_{n+1} . Y se lee de la siguiente forma: Si el estado presente y la entrada tienen cierto valor, ¿cuál es el estado próximo?.

TABLAS DE EXCITACION

La tabla de excitación es un listado de las combinaciones del estado presente Q_n y el estado próximo Q_{n+1} . Para cada combinación debe de existir un valor de las ENTRADAS que condicione la combinación, esta tabla se lee así: Si tenemos un estado actual Q_n y queremos cambiar a un estado próximo Q_{n+1} ¿Cuál debe ser el valor de la entrada?

La tabla de estados se obtiene a partir de la **tabla característica** — y la tabla de excitación se obtiene a partir de la **tabla de estados**.

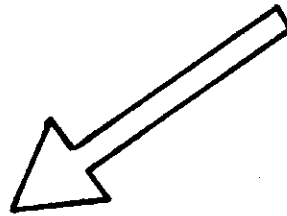
FLIP-FLOP JK



a)

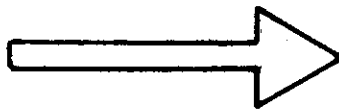
| JK | Q_{n+1} |
|----|-------------|
| 00 | Q_n |
| 01 | 0 |
| 10 | 1 |
| 11 | \bar{Q}_n |

b)



| Q_n JK | Q_{n+1} |
|----------|-----------|
| 000 | 0 |
| 001 | 0 |
| 010 | 1 |
| 011 | 1 |
| 100 | 1 |
| 101 | 0 |
| 110 | 1 |
| 111 | 0 |

c)

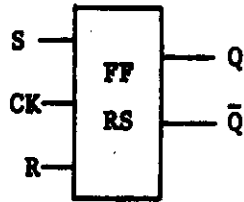


| Q_n Q_{n+1} | JK |
|-----------------|----|
| 00 | 0X |
| 01 | 1X |
| 10 | X1 |
| 11 | X0 |

d)

Fig. 7.18 FLIP-FLOP JK a) SIMBOLO, b) TABLA CARACTERISTICA
c) TABLA DE ESTADOS
d) TABLA DE EXCITACION

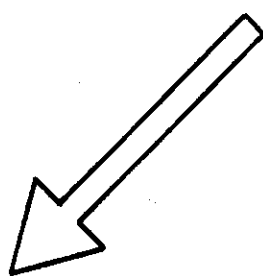
FLIP-FLOP RS



a)

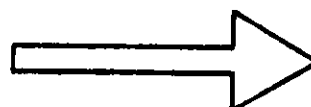
| RS | Q_{n+1} |
|----|-----------|
| 00 | Q_n |
| 01 | 1 |
| 10 | 0 |
| 11 | ? |

b)



| Q_n | R | S | Q_{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 | ? |

c)



| Q_n | Q_{n+1} | RS |
|-------|-----------|----|
| 0 | 0 | X0 |
| 0 | 1 | 01 |
| 1 | 0 | 10 |
| 1 | 1 | 0X |

d)

Fig. 7.19 FLIP-FLOPS RS a) SIMBOLO, b) TABLA CARACTERISTICA
 c) TABLA DE ESTADOS
 d) TABLA DE EXCITACION

LOS FLIP-FLOP D Y T

El FF JK es un dispositivo muy versátil, sin embargo en ocasiones -- es necesario reducir el número de entradas de control a un FF e incluso -- restringir su funcionamiento a solo unos cuantos estados.

Una de las opciones que presenta el FF JK es hacer una conexión entre J y K, $K = J$. El comportamiento de este FF se describe en la fig. 7.20 Se puede ver en su tabla característica que cuando $T = 1$ el estado del -- FF se complementa, el nombre de este FF en inglés es toggle, o simplemente FLIP-FLOP T. La otra opción es conectar a J lo que entra a K pero negado o $K = \bar{J}$, la característica de este FF aparece en la fig. 7.21. El nombre que recibe es FLIP-FLOP D del término en inglés delay.

Las tablas características del FFT y FFD se obtienen de la tabla característica del FFJK.

FLIP-FLOP

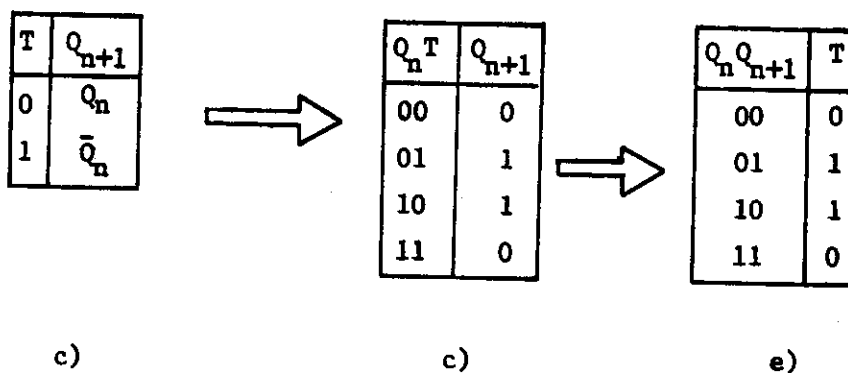
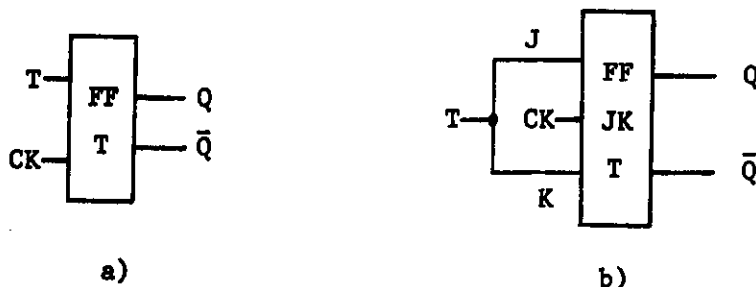
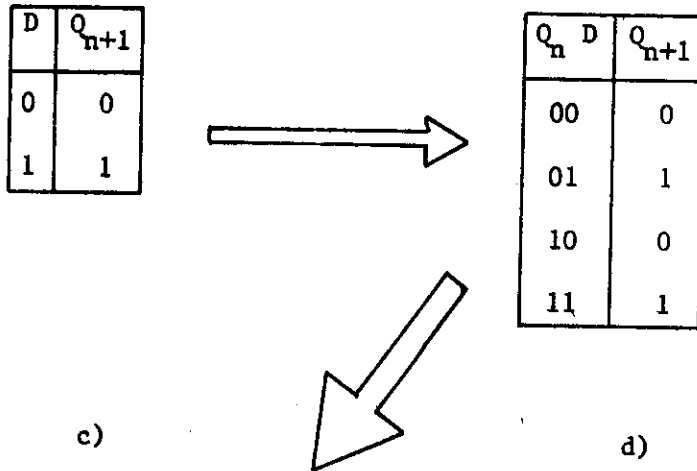
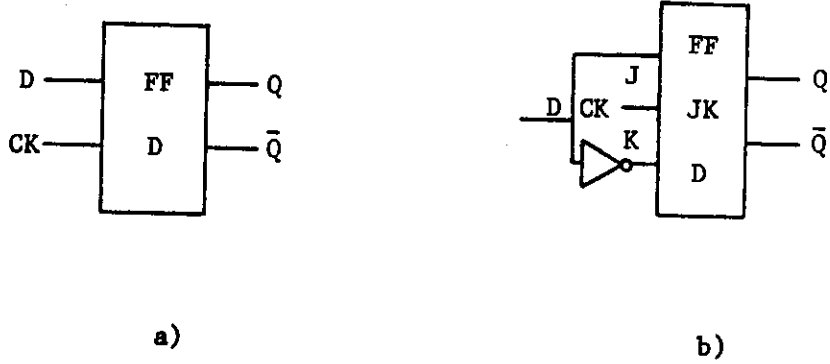


Fig. 7.20 FLIP FLOP T. a) SIMBOLO, b) IMPLEMENTACION DE UN FFT A PARTIR DE UN FF JK. c) TABLA CARACTERISTICA d) TABLA DE ESTADOS e) TABLA DE EXCITACION

FLIP-FLOP D



| $Q_n Q_{n+1}$ | D |
|---------------|---|
| 00 | 0 |
| 01 | 1 |
| 10 | 0 |
| 11 | 1 |

e)

Fig. 7.21 FLIP-FLOP D. a) SIMBOLO, b) IMPLEMENTACION DE UN FFD A PARTIR - DE UN FFJK, c) TABLA CARACTERISTICA, d) TABLA DE ESTADOS, e) TABLA DE EX-CITACION.

7.9 OTRAS ENTRADAS DE CONTROL A UN FLIP-FLOP

Las entradas a los FF que hemos discutido en los puntos anteriores de este capítulo y que denominamos entradas de control (RS, J, K, T, D), caen en la categoría de entradas SINCRONAS porque el efecto que tienen sobre el estado del FF depende de que se presente el pulso de reloj, en otras palabras están sincronizadas al CK.

Con el propósito de aumentar la versatilidad de los FLIP-FLOPS se les dota de otras dos entradas de control que operan en forma ASINCRONA estas entradas son el CLEAR y el PRESET. La operación asíncrona se refiere a que no necesitan del pulso de reloj para actuar sobre el FF. Otra característica de estas entradas es que son entradas prioritarias es decir actúan sobre el comportamiento del FF sin permitir la acción de otra entrada, incluso el reloj CK.

CLEAR

Esta entrada asíncrona condiciona al FLIP-FLOP a cargar un 0 lógico en Q. CLEAR = BORRAR, el concepto de borrar una memoria se asocia con la escritura de un 0 lógico.

PRESET

Esta entrada asíncrona condiciona al FLIP-FLOP a cargar un 1 lógico en Q. CLEAR = $\overline{\text{PRESET}}$.

El CLEAR y PRESET pueden actuar sobre el FF presentándose como niveles altos (1 log.) o bajos (0 log.) A diferencia de las demás entradas al FF el reloj se considera una entrada DINAMICA puesto que los cambios del FF se presentan en las transiciones ya sea positivas o negativas. Las entradas ACTIVAS de tipo alto o bajo lo mismo las entradas dinámicas se representan gráficamente como se muestran en la fig. 7.22

En la fig. 7.23 aparece una tabla característica que muestra la acción del CLEAR y PRESET sobre un FFD, la señal de reloj también aparece en esta tabla que corresponde a un FFD SN 7474.

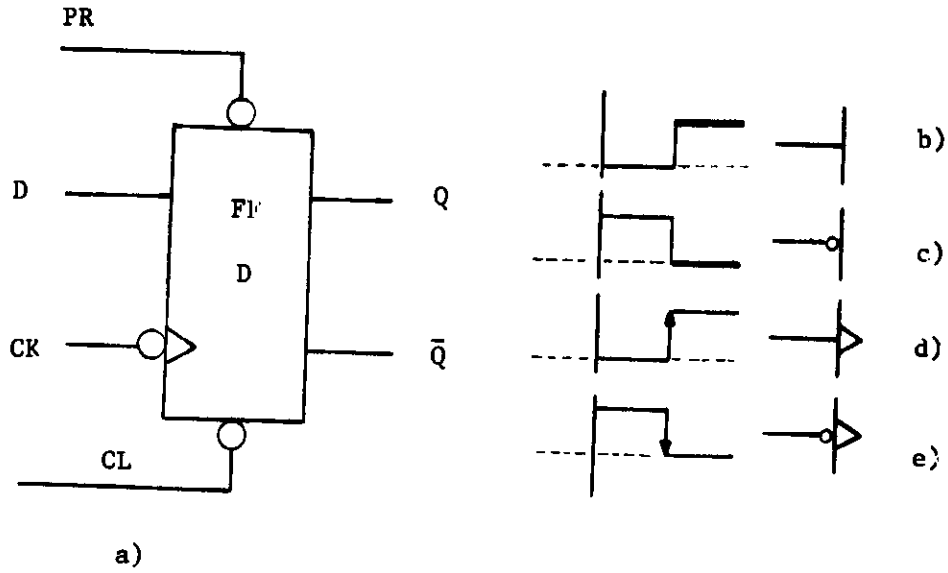


Fig. 7.22 a) FFD CON PRESET Y CLEAR, b) ENTRADA ACTIVA ALTA, c) ENTRADA ACTIVA BAJA, d) ENTRADA DINAMICA ALTA - e) ENTRADA DINAMICA BAJA.

| ENTRADAS | | | | SALIDAS | |
|----------|-------|----|---|---------|-------------|
| PRESET | CLEAR | CK | D | Q | \bar{Q} |
| L | H | X | X | H | L |
| H | L | X | X | L | H |
| L | L | X | X | H* | H* |
| H | H | ↑ | H | H | L |
| H | H | ↑ | L | L | H |
| H | H | L | X | Q_0 | \bar{Q}_0 |

a)

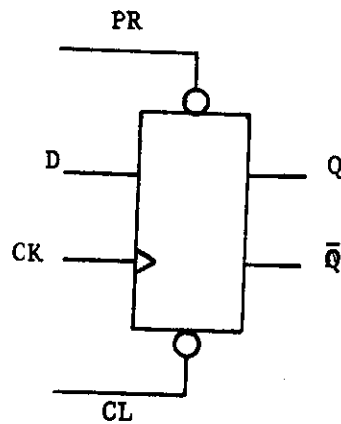


Fig. 7.23 a) TABLA CARACTERISTICA (DE FUNCIONAMIENTO) PARA UN FFD SN 7474 LAS H INDICAN UN 1 LOGICO LAS L INDICAN UN 0 LOGICO Y LAS X -- DON'T CARE. b) FF. *ESTA CONFIGURACION NO ES ESTABLE, NO PERSISTE CUANDO EL PRESET Y CLEAR REGRESAN A SU NIVEL INACTIVO H.

7.10 REGISTROS DE CORRIMIENTO

Se le da el nombre de REGISTRO DE CORRIMIENTO a un grupo de FLIP-FLOPS conectados en cascada. Es decir, un registro es un circuito en el cual las salidas de cada FF se conectan a la entrada del siguiente. Otra característica importante es que todos los FF de un registro comparten el mismo reloj, con esto se asegura que todas las transiciones sean simultáneas.

Los registros de corrimiento se clasifican de acuerdo a la forma en que cargan y entregan su información, existen 5 categorías.

- 1- Entrada serie-salida serie
- 2- Entrada serie-salida paralelo
- 3- Entrada paralelo-salida serie
- 4- Entrada paralelo-salida paralelo
- 5- Entrada paralelo-salida paralelo bidireccional

El comportamiento de cada FF en un registro debe ser igual al del FF-D, también se usan FLIP-FLOPS JK o RS conectados como FF-D.

En la fig. 7.24 a) se muestra un registro de corrimiento de 4 bits - formado por 4 FF-RS configurados como FF-D, el CK es común a todos los FF lo mismo que el CLEAR. En el inciso b) de esta misma figura aparece - un diagrama de tiempo, donde se puede apreciar el corrimiento que experimentan 2 UNOS a través del registro.

ENTRADA SERIE-SALIDA SERIE

Prácticamente no existe límite en cuanto a la cantidad de bits de un registro ENTRADA SERIE-SALIDA SERIE puesto que un circuito integrado precisa de únicamente 5 terminales, 1, entrada, 2, salida, 3, reloj, 4 y 5 -- VCC y GND. (alimentación). Este tipo de registros se fabrican usualmente en 8 bit como se muestra en la fig. 7.25. La entrada serie es el producto de A y B, la salida es la del último FF Q_H y \bar{Q}_H .

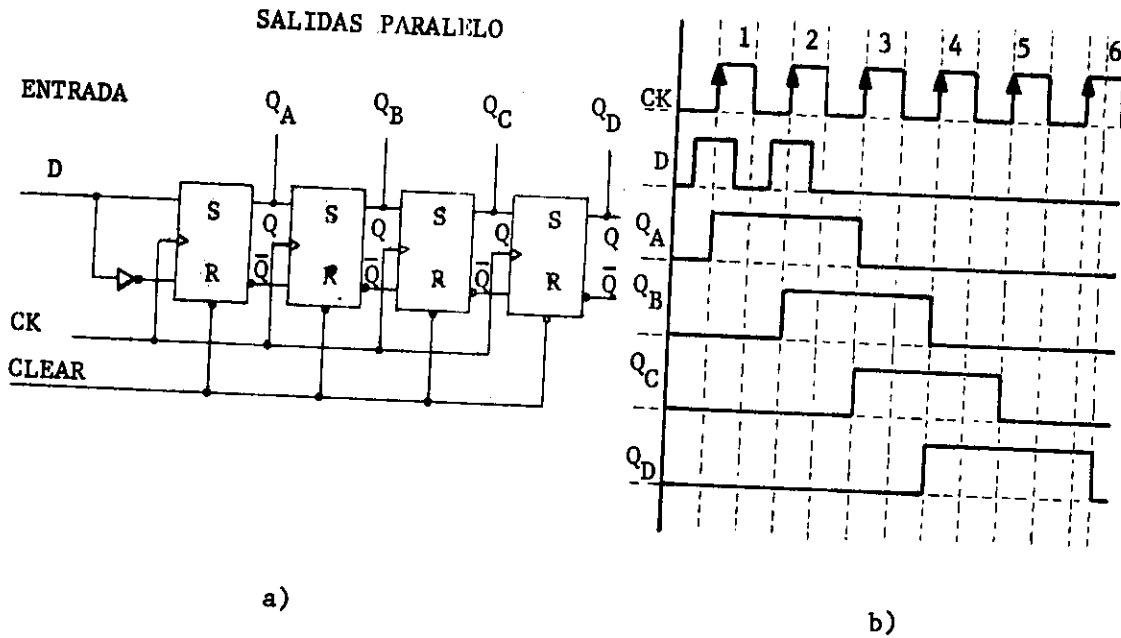


Fig. 7.24 a) REGISTRO DE CORRIMIENTO DE 4 BITS ENTRADA SERIE-SALIDA SERIE, SALIDA PARALELO. b) DIAGRAMA DE TIEMPO.

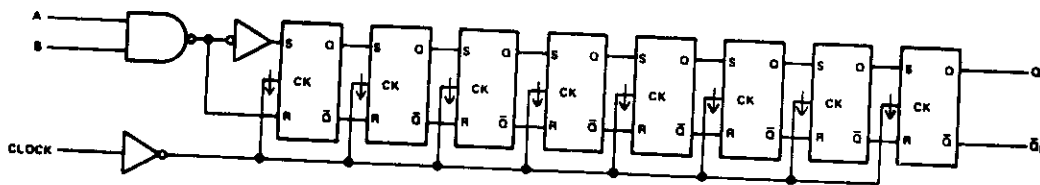


Fig. 7.25 REGISTRO DE CORRIMIENTO ENTRADA SERIE-SALIDA SERIE DE 8 BIT.

ENTRADA SERIE-SALIDA PARALELO

En este registro de corrimiento existe la limitante de la cantidad de terminales del circuito integrado, de tal forma que se fabrican en cantidades reducidas de bits. En la fig. 7.26, se muestra un registro de corrimiento que carga la información serie, la entrada serie es igual a A.B existe acceso a la salida de cada FF, además el circuito cuenta con una entrada de CLEAR común a cada FF para borrar todo el registro.

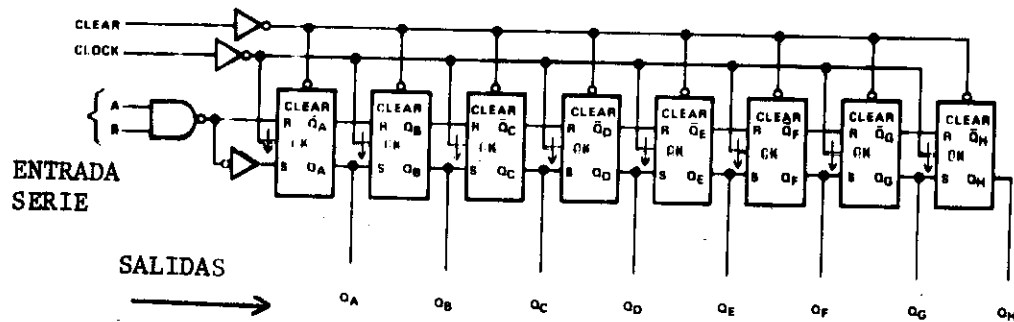


Fig. 7.26 REGISTRO DE CORRIMIENTO ENTRADA SERIE-SALIDA PARALELO.

ENTRADA PARALELO-SALIDA SERIE

Existen dos formas de cargar datos en paralelo. La primera es usando las entradas de PRESET y CLEAR de cada FF, como se muestra en la fig.- 7.27. Este tipo de carga es asincrónica. Para explicar la carga en paralelo, tomemos una de las entradas. Al habilitarse la línea de carga en paralelo y entrar un 1 por A la salida del NAND superior es un 0 lógico y la del NAND inferior es un 1 lógico, por lo tanto se acciona el PRESET y $Q = 1$. Si la entrada es 0 lógico, la salida del primer NAND es un 0 y la del segundo es un 1 por lo tanto se acciona el CLEAR y $Q = 0$, cuando se cambia el modo de operación $LOAD = 1$ los datos corren en serie.

ENTRADA PARALELO-SALIDA PARALELO

La segunda forma de cargar datos en paralelo es por medio de un selector semejante al discutido en el punto 6.2 del capítulo 6. La salida del selector se conecta a la entrada de cada FF, una de las entradas del selector es la entrada paralelo y la otra es la Q del FF anterior. En la fig. 7.28 aparece este circuito. La carga de datos es SINCRONA es decir que para cargar datos en serie o paralelo es necesario el pulso del reloj.

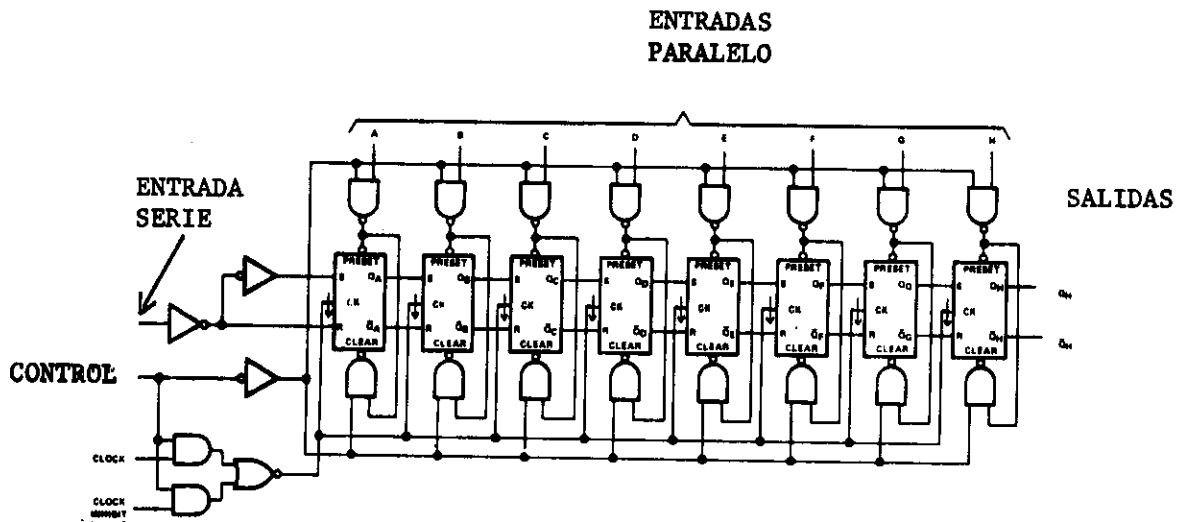


Fig. 7.27 REGISTRO DE CORRIMIENTO ENTRADA PARALELO-SALIDA
SERIE

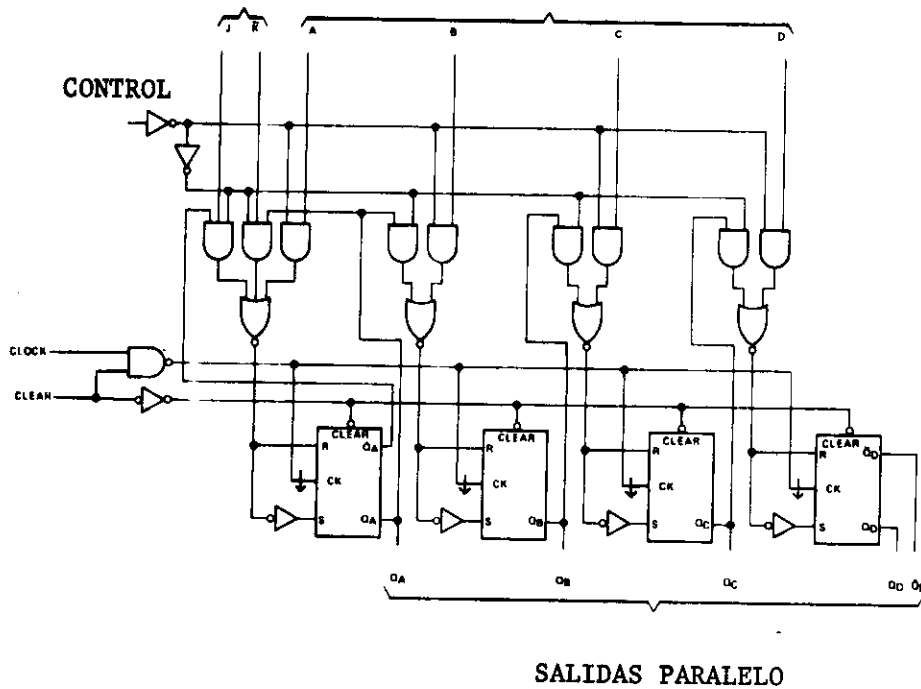


Fig. 7.28 REGISTRO DE CORRIMIENTO ENTRADA PARALELO-SALIDA
PARALELO.

ENTRADA PARALELO-SALIDA PARALELO BIDIRECCIONAL

Al registro bidireccional con todas las demás funciones se le conoce generalmente como REGISTRO UNIVERSAL, su operación es semejante al registro con carga en paralelo que utiliza un selector, solo que aquí las entradas del selector son 3. Una viene del FF anterior otra de la entrada paralelo y la última del FF posterior. Si se desea correr datos a la derecha se usa la primer compuerta, si se desea correr datos a la izquierda se usa la última compuerta para cargar los datos del FF posterior. En la fig. 7.29 se muestra este registro.

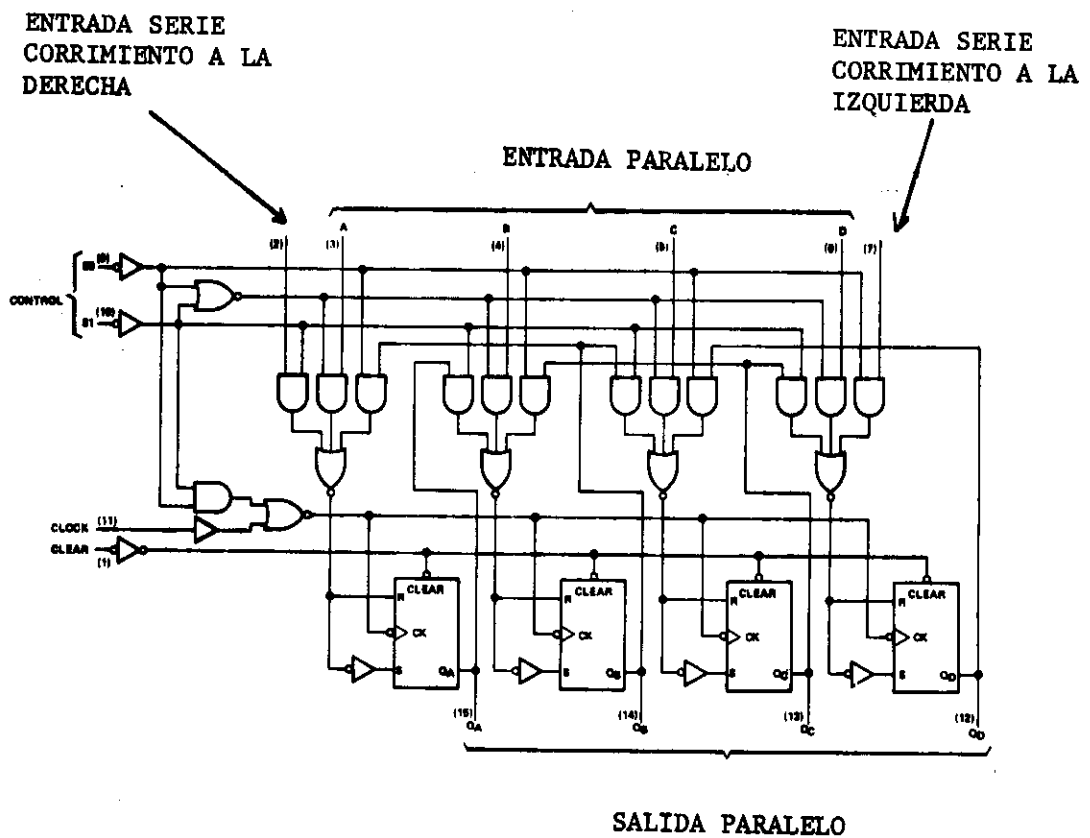


Fig. 7.29 REGISTRO DE CORRIMIENTO BIDIRECCIONAL (UNIVERSAL)

7.11 MULTIVIBRADORES

La clasificación de los FLIP-FLOP como elementos que tienen 2 estados es la siguiente:

- 1- ELEMENTOS BIESTABLES, tienen 2 estados estables. Los FLIP-FLOPS estudiados en los puntos anteriores de este mismo capítulo están incluidos en esta clasificación.
- 2- ELEMENTOS MONOESTABLES. Son circuitos con un solo estado estable, del cual pueden cambiar por un predeterminado intervalo de tiempo. Después regresa a su estado original.
- 3- ELEMENTOS ASTABLES, se refiere a un circuito que tiene 2 estados y no permanece estable en ninguno de ellos.

MULTIVIBRADOR MONOESTABLE

También conocido como ONE-SHOT, SINGLE-SHOT o START-STOP, es un FF - empleado para producir pulsos con una duración que puede ser independiente a la frecuencia de entrada. El FF se dispara con una entrada transitoria, causando un cambio al estado no estable, por un tiempo determinado - por un circuito externo RC y después regresa a su estado original. En la fig. 7.31 aparece el 74121 multivibrador monoestable, la exactitud y duración de la salida dependen de la resistencia R_X y capacitor C_X externos - y están determinados por la siguiente expresión:

$$t = K_D R_X C_X \left(1 + \frac{0.7}{R_X} \right)$$

Donde

$$C_X = \text{pF}$$

$$R_X = \text{ohms}$$

$$t = n \text{ seg.}$$

$$\begin{aligned} k &= 0.28 \text{ para un } 74123 \\ &= 0.32 \text{ para un } 74122 \\ &= 0.33 \text{ para un } 74L123 \\ &= 0.37 \text{ para un } 74L122 \end{aligned}$$

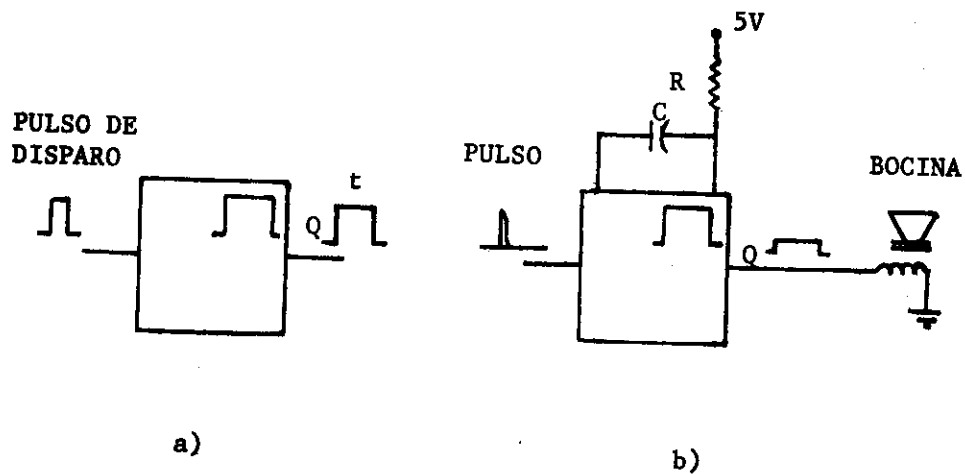


Fig. 7.30 MULTIVIBRADOR MONOESTABLE a) SIMBOLO LOGICO
b) APLICACION DETECTOR DE UN PULSO INTERMITENTE.

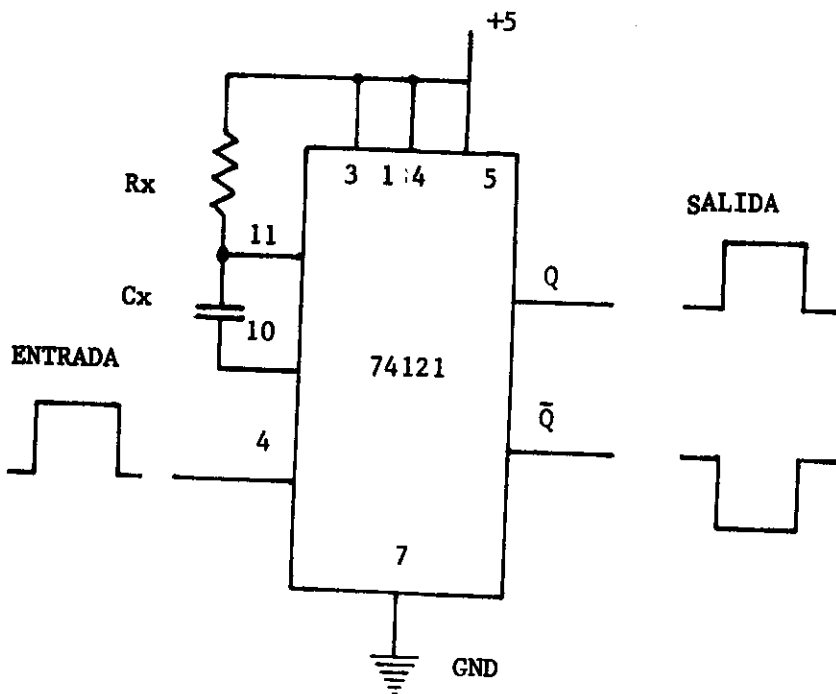


Fig. 7.31 CONEXION PARA UN 74121 MULTIVIBRADOR MONOESTABLE.

MONOESTABLES REDISPARABLES Y NO REDISPARABLES

En un multivibrador monoestable existe un retardo de tiempo entre la entrada y la salida determinado por la constante RC. En la fig. 7.32 a) se muestra el comportamiento del multivibrador monoestable NO REDISPARABLE. La salida inicia en su estado original "0" al presentarse el pulso A cambia al estado "1", y permanece en este estado por un tiempo "t" y los pulsos B y C se ignoran a lo largo del "t", el pulso D dispara de nuevo al circuito y el pulso E no afecta al monoestable por que no es una transición de "0" a "1" mientras la salida está en "0".

El multivibrador monoestable REDISPARABLE puede redispararse antes de que el pulso de salida termine, esto hace que el tiempo del pulso de salida se extienda. En la fig. 7.32 b) aparece el comportamiento de este multivibrador. El pulso A cambia de estado al circuito y el pulso B lo redispara logrando así un tiempo doble para el pulso de salida.

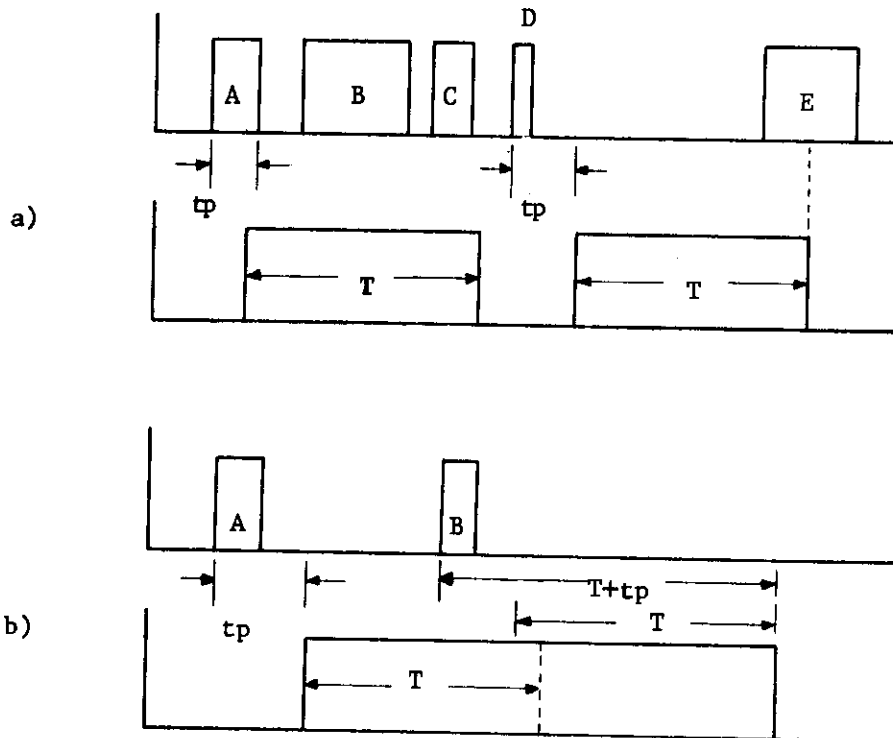


Fig. 7.32 DIAGRAMAS DE TIEMPO PARA a) UN MULTIVIBRADOR MONOESTABLE NO REDISPARABLE Y b) UN MULTIVIBRADOR MONOESTABLE REDISPARABLE.

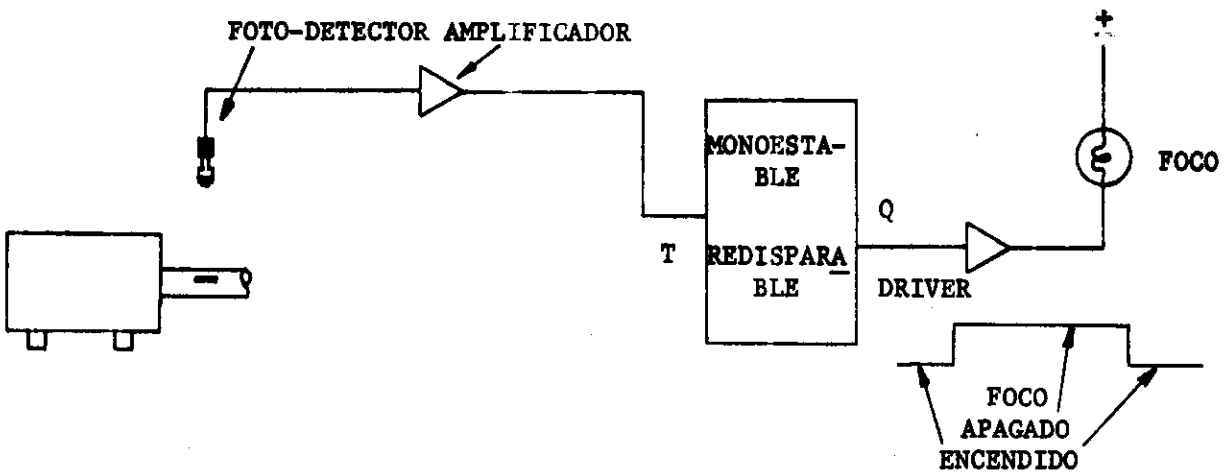


Fig. 7.33 Detector de Cero Movimiento de la Flecha de un Motor

En la figura 7.33 aparece una aplicación de un multivibrador monoestable-redisparable, se trata de un detector de cero movimiento de la flecha de un motor.

Este circuito es muy útil sobre todo en motores grandes a los que continuamente se les cambia el sentido de giro. Su objetivo principal es evitar la demanda excesiva de corriente de armadura para vencer la inercia de giro, en sentido contrario.

FUNCIONAMIENTO

En la flecha del motor se colocan una o varias marcas reflejantes que se captan por un foto-detector, cuya señal acondicionada mediante un amplificador, dispara a un multivibrador monoestable-redisparable.

La constante de tiempo se ajusta a un valor tal que al pasar la primera marca dispara al monoestable y continúe este redisparándose con las siguientes marcas.

Mientras el monoestable permanezca disparado el foco se mantendrá apagado.

Al no detectarse más el paso de las marcas, el retardo de tiempo concluirá, cambiará de estado el multivibrador y el foco encenderá indicando que el motor ha dejado de girar.

MULTIVIBRADOR ASTABLE

También conocido como FREE-RUNNING, es un dispositivo que tiene como salida dos estados y ninguno de ellos es estable, el resultado es que la salida oscila entre esos dos estados obteniéndose una onda cuadrada - positiva. El Timer 555 se usa satisfactoriamente como multivibrador astable, tiene un ciclo de trabajo ajustable y opera con frecuencias de -- 0.1 Hz a 100 KHz. en la fig. 7.34 se muestra su conexión como astable.

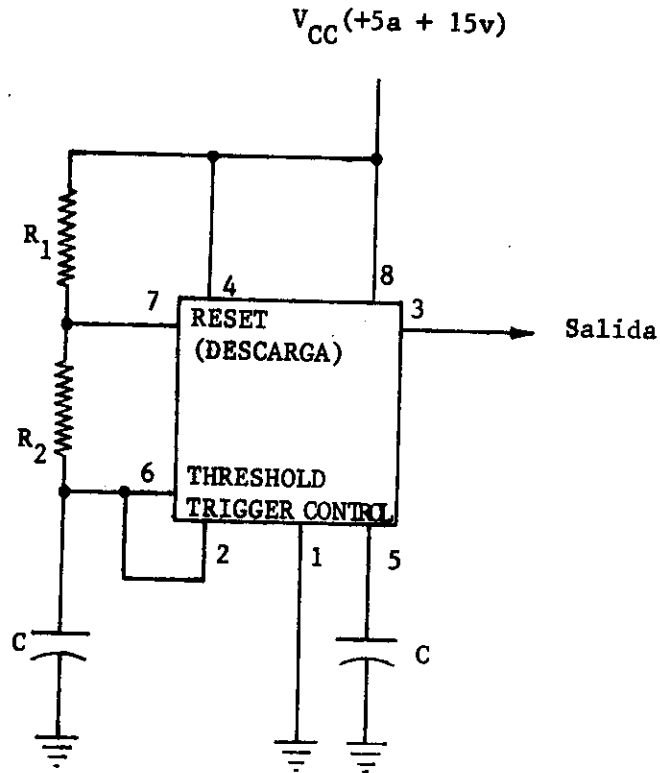


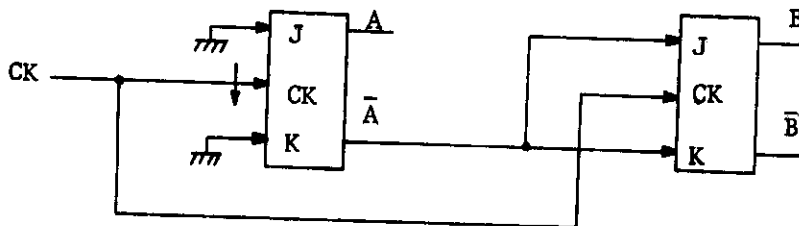
Fig. 7.34 TIMER 555 CONECTADO COMO MULTIVIBRADOR ASTABLE.

PROBLEMAS PROPUESTOS

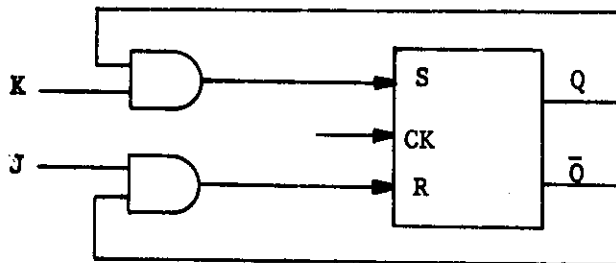
- 1.- ¿Cual es la finalidad de un Diagrama de Tiempo?
- 2.- Defina lo que es un Flip - Flop
- 3.- Describa la diferencia entre un sistema síncrono y un asíncrono.
- 4.- Que función tiene el reloj en un Flip - Flop.
- 5.- complete las tablas de excitación para cada tipo de Flip - Flop, en los siguientes estados.

| Q_n | Q_{n+1} | FF D | FF T | FF R S | FF J K |
|---------|-----------|---------|---------|-----------|-----------|
| 0 0 0 0 | 1 1 1 1 | | | | |
| 0 0 0 1 | 0 0 1 1 | | | | |
| 0 0 1 0 | 1 1 0 1 | | | | |
| 0 0 1 1 | 1 1 0 0 | | | | |
| 0 1 0 0 | 1 0 1 1 | | | | |
| 0 1 0 1 | 1 0 1 0 | | | | |
| 0 1 1 0 | 1 0 0 1 | | | | |
| 0 1 1 1 | 1 0 0 0 | | | | |
| 1 0 0 0 | 0 1 1 1 | | | | |
| 1 0 0 1 | 0 1 1 0 | | | | |
| 1 0 1 0 | 0 1 0 1 | | | | |
| 1 0 1 1 | 0 1 0 0 | | | | |
| 1 1 0 0 | 0 0 1 1 | | | | |
| 1 1 0 1 | 0 0 1 0 | | | | |
| 1 1 1 0 | 0 0 0 1 | | | | |
| 1 1 1 1 | 0 0 0 0 | | | | |

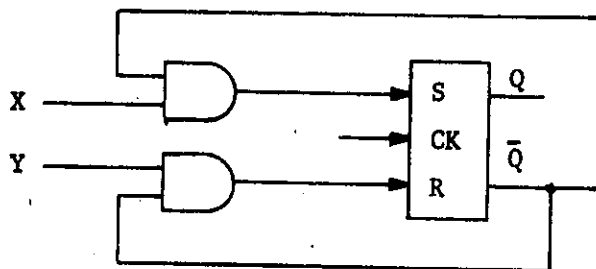
- 6.- ¿En que consiste un Registro de Corrimiento, cuantos tipos hay y defina brevemente cada uno de ellos,
- 7.- Dos FF - JK son interconectados como se muestra en la fig. Determine la tabla de excitación para los dos FF, con entrada de reloj.- Comience con AB = 00.



- 8.- Defina brevemente, los multivibradores. a) Biestables, b) Monoestables (Redisparable y no Redisparable) y c) Astables.
- 9.- Compruebe, realizando la tabla característica, si la siguiente configuración corresponde a un FF - JK



- 10.- Determine que tipo de FF representa la siguiente configuración realizando la tabla característica.



8

Diseño Secuencial

8.0 INTRODUCCION

Cuando hicimos referencia a un SISTEMA SECUENCIAL en el punto 7.0 -- del capítulo anterior, representamos a estos sistemas como un bloque que tiene una parte COMBINACIONAL y una parte de memoria formada por FLIP -- FLOPS como se muestra en la fig. 8.0

Para establecer una metodología de diseño secuencial partiremos del hecho de que un FF ya está diseñado, por ejemplo los FLIP-FLOPS JK, RS, D y T que también fueron discutidos en el capítulo anterior, entonces lo único que realmente queda por diseñar es la parte combinacional.

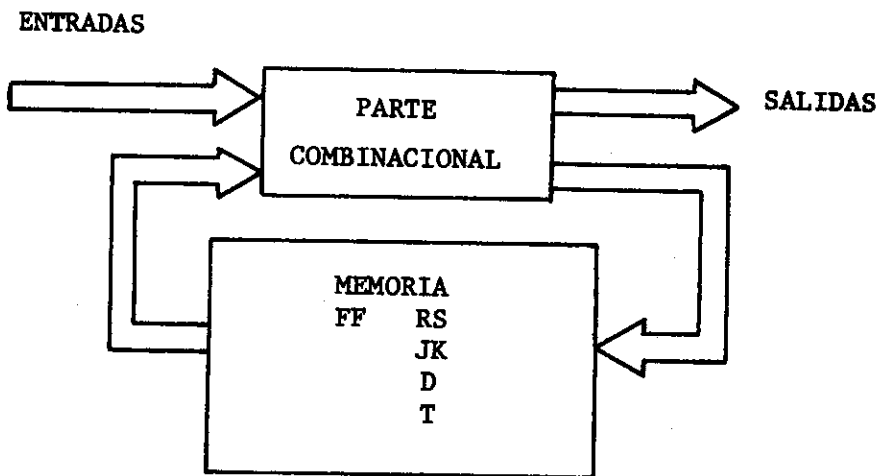


Fig. 8.0 LA PARTE DE MEMORIA EN UN SISTEMA SECUENCIAL ESTA FORMADA POR FLIP-FLOPS.

8.1 TABLAS DE ESTADOS

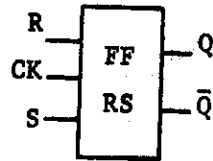
Una TABLA DE ESTADOS también conocida como tabla de transición es un listado de los estados de un sistema digital.

El máximo número de estados de un sistema digital está limitado por el número de FLIP-FLOPS. Si el sistema tiene n FLIP-FLOPS el número de estados es igual a 2^n . El número de entradas o salidas a un sistema digital NO está relacionado con el número de estados.

Una tabla de estados debe contener la siguiente información:

- Los estados presentes Q_n indicados por un número binario.
- Los estados próximos Q_{n+1} para todas las COMBINACIONES de Q_n y la entrada al sistema.
- Las salidas correspondientes a cada una de estas combinaciones.

En la fig. 8.1 aparece un FF-RS y su tabla de estados distribuída en dos formas diferentes. La forma del inciso c) es más práctica para la representación de sistemas secuenciales.



a)

| Q_n SR | Q_{n+1} |
|----------|-----------|
| 000 | 0 |
| 001 | 0 |
| 010 | 1 |
| 011 | ? |
| 100 | 1 |
| 101 | 0 |
| 110 | 1 |
| 111 | ? |

b)

| ESTADO PRESENTE Q_n | ENTRADAS S Y R | | | |
|--------------------------|---------------------------|-------------------------|-------------------------|-------------------------|
| | 00 | 01 | 10 | 11 |
| | ESTADO PROX. Q_{n+1} | EST. PROX. Q_{n+1} | EST. PROX. Q_{n+1} | EST. PROX. Q_{n+1} |
| 0 | 0 | 0 | 1 | ? |
| 1 | 1 | 0 | 1 | ? |

c)

Fig. 8.1 FF RS a) SIMBOLO b) y c) TABLA DE ESTADOS.

8.2 DIAGRAMAS DE TRANSICION

El análisis de un FF se hace generalmente usando un diagrama de tiempo, que es un método útil incluso para bloques secuenciales pequeños. Sin embargo es preferible usar un DIAGRAMA DE TRANSICION de estados.

Un DIAGRAMA DE TRANSICION es la representación gráfica del comportamiento de un sistema secuencial. La información de una tabla de estados puede trasladarse a un diagrama de este tipo.

En un DIAGRAMA DE TRANSICION cada estado se representa por un círculo y cada transición se representa por una flecha que parte de un estado y termina en otro o en el mismo. Las entradas o salidas del sistema se marcan sobre cada flecha para indicar su valor en cada transición.

En la fig. 8.2 se muestra un diagrama de transición generalizado para un sistema secuencial que tiene 2 estados, una entrada y una salida.

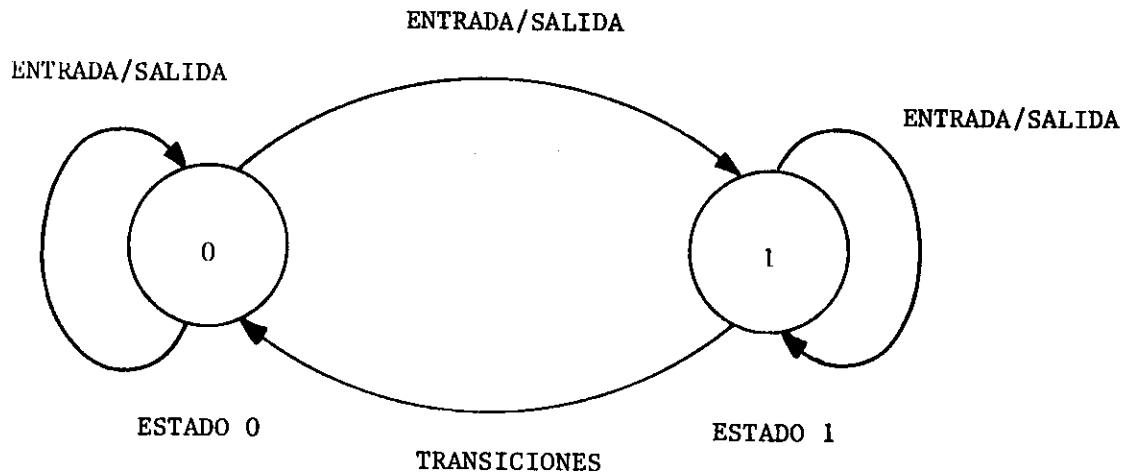


Fig. 8.2 DIAGRAMA DE TRANSICION GENERALIZADO

Una flecha que sale de un estado y regresa a el mismo estado indica que el sistema no experimentó ninguna transición, fig. 8.3a). Un estado transitorio se indica por una flecha que sale de el, fig. 8.3b). En este tipo de estados generalmente no se puede entrar pero si se puede salir. Y por último los estados aislados fig. 8.3c) se puede entrar a ellos pero no se puede salir.

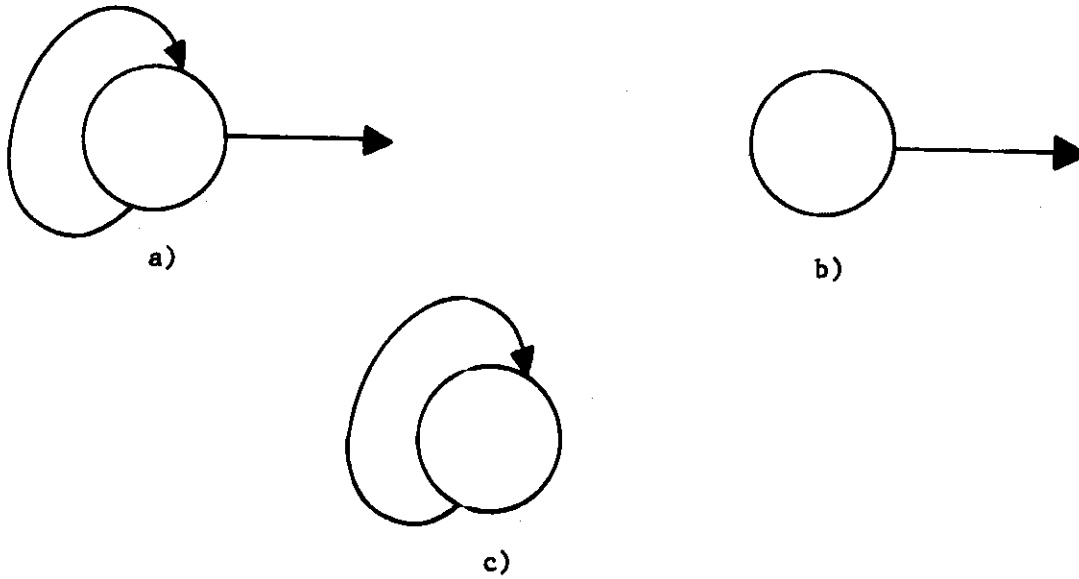


Fig. 8.3 VARIOS TIPOS DE ESTADOS a) PERMANENCIA, b) ESTADO TRANSITORIO, c) ESTADO AISLADO.

8.3 METODOLOGIA DE DISEÑO SECUENCIAL

El método de diseño secuencial es semejante al de diseño combinacional, y aunque algunos autores sofistican mucho este proceso el método básico se reduce a los siguientes pasos.

- 1- Especificación del sistema.
- 2- Determinar el número de entradas y salidas.
- 3- Seleccionar el tipo de FLIP-FLOPS
- 4- Trasladar el comportamiento del sistema una tabla de estados.
- 5- Reducción de estados.
- 6- Implementación de los bloques combinacionales de entrada y salida.

EJEMPLO 8.0

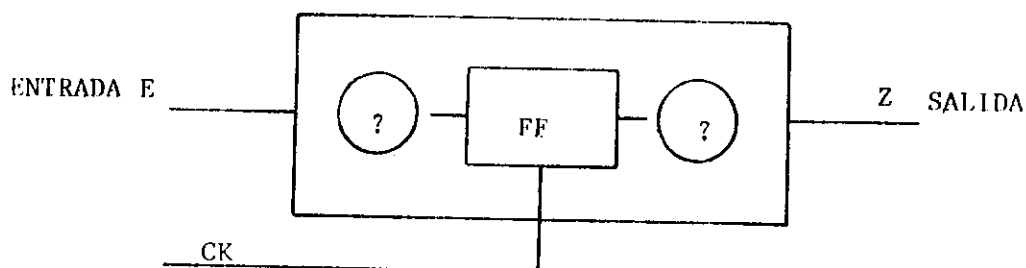
Diseñe un sistema digital que tenga dos estados (0 y 1) además -- una entrada E y una salida Z, si la entrada E = 0 el sistema debe permanecer en el mismo estado, si la entrada E = 1 debe cambiar de estado y la salida Z = 1 solamente cuando la entrada E = 1, y el sistema está en el estado "0".

1- La descripción anterior cumple con el primer paso de diseño.

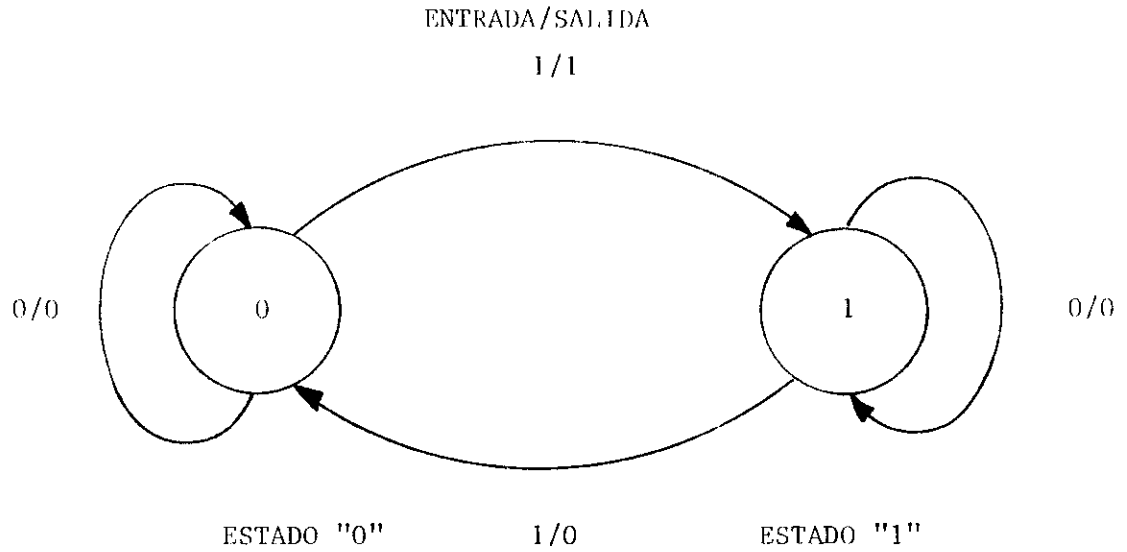
2- Determinar el # de entradas y salidas al sistema.

3- El sistema será diseñado por los 4 tipos de FF, JK, RS, D y T.

Nótese que una vez seleccionado el tipo de FF resta la parte combinacional, que conecta la entrada A y la salida Z con el bloque de memoria.



4- Para trasladar el comportamiento del sistema a una tabla de verdad haremos uso de un diagrama de transición.



| ESTADO PRESENTE Q_n | E = 0 | | E = 1 | |
|--------------------------|-----------------------------|-------------|-----------------------------|-------------|
| | ESTADO PROXIMO Q_{n+1} | SALIDA Z | ESTADO PROXIMO Q_{n+1} | SALIDA Z |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |

La tabla de estados debe incluir la entrada E y el o los FLIP-FLOPS que se usarán para implementar el sistema. Por lo tanto lo único que cambia en los ejemplos subsecuentes es precisamente la entrada al FF.

5- REDUCCION DE ESTADOS

- a) No existen estados redundantes.

DISEÑO CON UN FF T

| ESTADO PRESENTE Q_n | E = 0 | | | E = 1 | | |
|--------------------------|-----------------------------|-------------|--------------------|-----------------------------|-------------|--------------------|
| | ESTADO PROXIMO Q_{n+1} | SALIDA Z | ENTRADA AL FF T | ESTADO PROXIMO Q_{n+1} | SALIDA Z | ENTRADA AL FF T |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 |

Para llenar la tabla anterior podemos auxiliarnos de la tabla de excitación del FF T

| Q_n | Q_{n+1} | T |
|-------|-----------|---|
| 00 | | 0 |
| 01 | | 1 |
| 10 | | 1 |
| 11 | | 0 |

Como podemos observar en la tabla de estados, la salida Z y la entrada al FF T son función de E y Q_n y además están completamente especificadas. En la parte superior del mapa aparecen las dos combinaciones de E y en el extremo izquierdo los valores de Q_n , esto nos hace recordar el acomodo de un mapa de Karnaugh para una función de 2 variables.

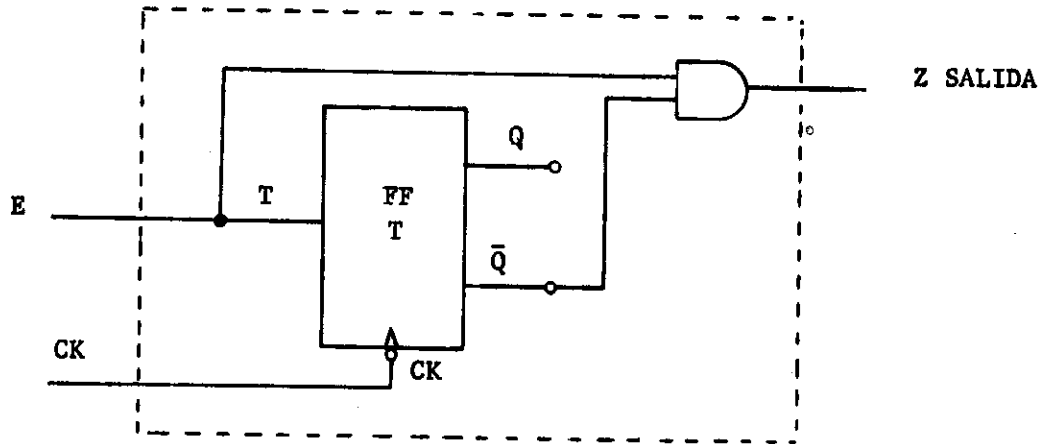
| | | |
|-------|---|---|
| | E | |
| | 0 | 1 |
| Q_n | | |
| 0 | 0 | 1 |
| 1 | 0 | 0 |

| | | |
|-------|---|---|
| | E | |
| | 0 | 1 |
| Q_n | | |
| 0 | 0 | 1 |
| 1 | 0 | 1 |

$$Z(E, Q_n) = E \cdot \bar{Q}_n$$

$$T(E, Q_n) = E$$

6- IMPLEMENTACION



NOTA: LA SEÑAL DE RELOJ NO INTERVIENE EN EL DISEÑO.

DISEÑO CON UN FF JK

El único cambio de la tabla de estados es la entrada al FF en este caso son J y K.

| ESTADO PRESENTE Q_n | E = 0 | | | E = 1 | | | | |
|--------------------------|-----------------------------|-------------|---------------|-------|-----------------------------|-------------|---------------|---|
| | ESTADO PROXIMO Q_{n+1} | SALIDA Z | ENTRADA AL FF | | ESTADO PROXIMO Q_{n+1} | SALIDA Z | ENTRADA AL FF | |
| | | | J | K | | | J | K |
| 0 | 0 | 0 | 0 | X | 1 | 1 | 1 | X |
| 1 | 1 | 0 | X | 0 | 0 | X | X | 1 |

| $Q_n Q_{n+1}$ | J K |
|---------------|-----|
| 00 | 0 X |
| 01 | 1 X |
| 10 | X 1 |
| 11 | X 0 |

Tabla de excitación del FF JK.

| | | |
|-------|---|---|
| | 0 | 1 |
| Q_n | 0 | 1 |
| 0 | 0 | 1 |
| 1 | X | X |

$$J(E, Q_n) = E$$

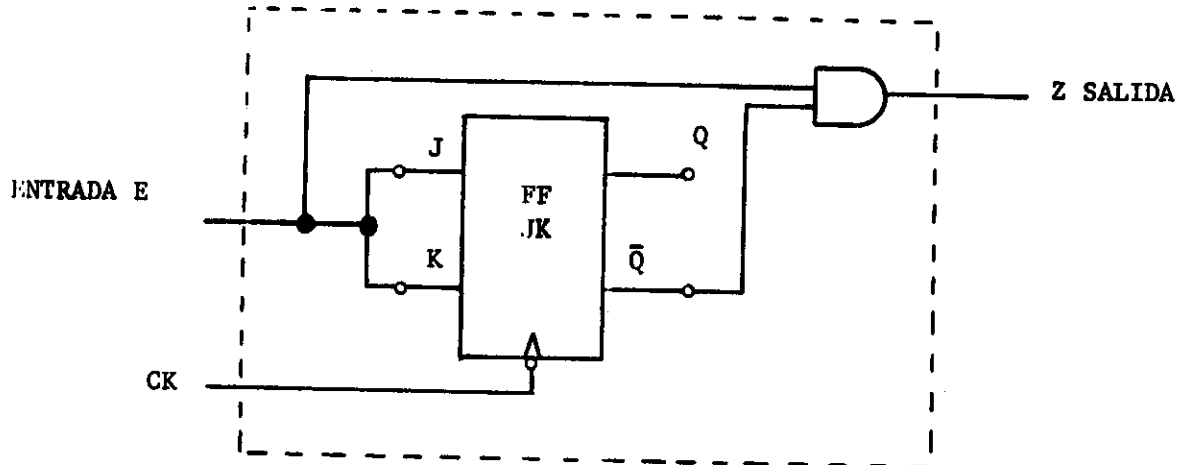
| | | |
|-------|---|---|
| | 0 | 1 |
| Q_n | 0 | 1 |
| 0 | X | X |
| 1 | 0 | 1 |

$$K(E, Q_n) = E$$

$$Z(E, Q_n) = E \cdot \bar{Q}_n$$

La salida permanece igual - puesto que depende solamente de (E y Q_n).

IMPLEMENTACION



DISEÑO CON UN FF D

El único cambio de la tabla de estados es la entrada al FF en este caso es solo una, D.

| ESTADO PRESENTE Q_n | E = 0 | | | E = 1 | | | Q_n Q_{n+1} | D |
|--------------------------|-----------------------------|----------|-----------------|-----------------------------|----------|-----------------|-----------------|---|
| | ESTADO PROXIMO Q_{n+1} | SALIDA Z | ENTRADA AL FF D | ESTADO PROXIMO Q_{n+1} | SALIDA Z | ENTRADA AL FF D | | |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 00 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 01 | 1 |
| | | | | | | | 10 | 0 |
| | | | | | | | 11 | 1 |

Tabla de excitación para un FF D.

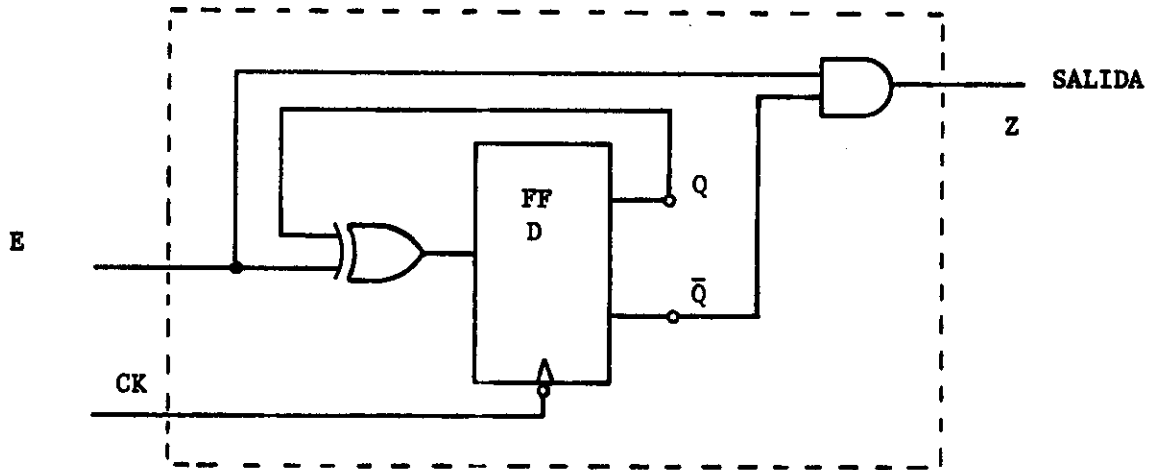
| | | | |
|---|---|---|---|
| | | E | |
| | | 0 | 1 |
| Q | 0 | 0 | 1 |
| | 1 | 1 | 0 |

$$Z(E, Q_n) = E \cdot \bar{Q}_n$$

$$D(E, Q_n) = E \cdot \bar{Q}_n + \bar{E} \cdot Q_n$$

$$D(E, Q_n) = E \oplus Q_n$$

IMPLEMENTACION



DISEÑO CON UN FF RS

| ESTADO PRESENTE Q_n | E = 0 | | | | E = 1 | | | | Q_n Q_{n+1} | SR |
|--------------------------|-----------------------------|----------|----------------------|---|-----------------------------|----------|----------------------|---|-----------------|----|
| | ESTADO PROXIMO Q_{n+1} | SALIDA Z | ENTRADA AL FF S R | | ESTADO PROXIMO Q_{n+1} | SALIDA Z | ENTRADA AL FF S R | | | |
| 0 | 0 | 0 | 0 | X | 1 | 1 | 1 | 0 | 00 | 0X |
| 1 | 1 | 0 | X | 0 | 0 | 0 | 0 | 1 | 01 | 10 |
| | | | | | | | | | 10 | 01 |
| | | | | | | | | | 11 | X0 |

| | | | |
|-------|---|---|---|
| | S | E | |
| | | 0 | 1 |
| Q_n | 0 | 0 | 1 |
| | 1 | X | 0 |

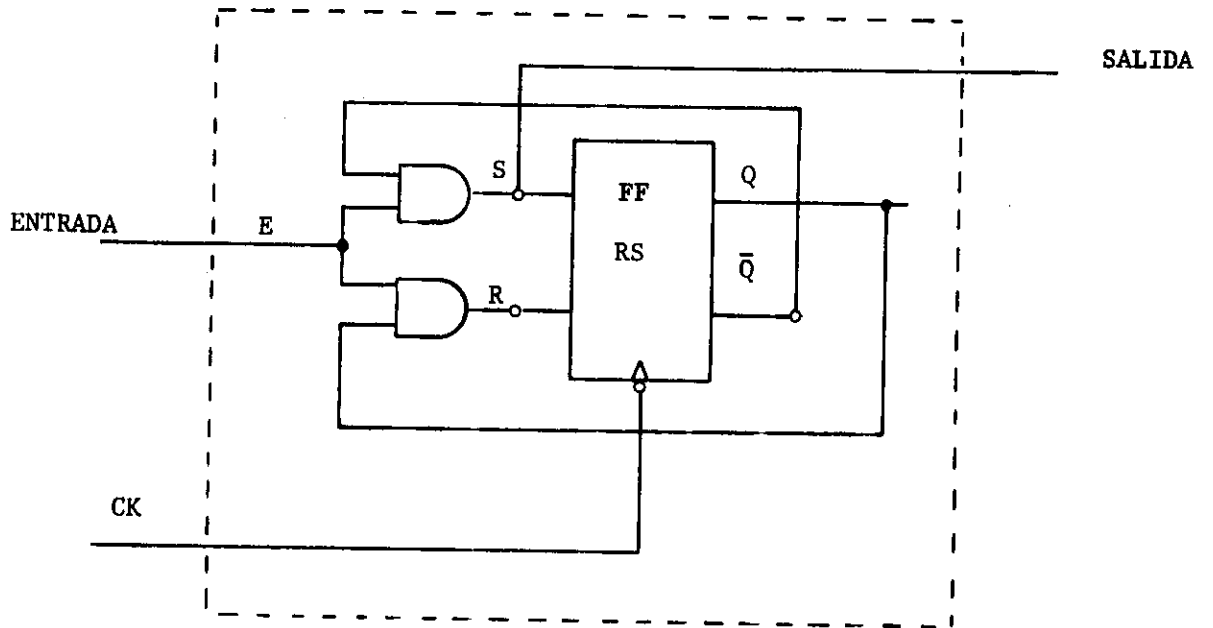
| | | | |
|-------|---|---|---|
| | R | E | |
| | | 0 | 1 |
| Q_n | 0 | X | 0 |
| | 1 | 0 | 1 |

$$Z(E, Q_n) = E \cdot \overline{Q_n}$$

$$S(E, Q_n) = E \cdot \overline{Q_n}$$

$$R(E, Q_n) = E \cdot Q_n$$

IMPLEMENTACION



8.4 SISTEMAS SECUENCIALES DE MAS DE 2 ESTADOS

El diseño secuencial de más de 2 estados implica además de la reducción de estados que se discutirá posteriormente, la asignación de los estados. Conforme aumenta la cantidad de estados, aumenta también la posibilidad de selección de un estado apropiadamente.

Imaginemos un sistema secuencial con 4 estados cuyo diagrama de transición aparece en la fig. 8.4, la asignación de los siguientes estados conducirá a diseñar un sistema secuencial donde su parte combinacional sea simple o muy complicada. Este sistema no tiene más entrada que el CK, por lo tanto en la diagonal que se dibuja sobre cada flecha no se indica la entrada.

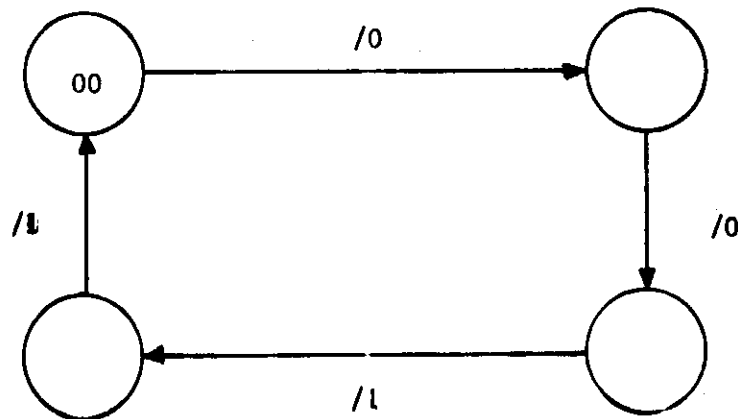


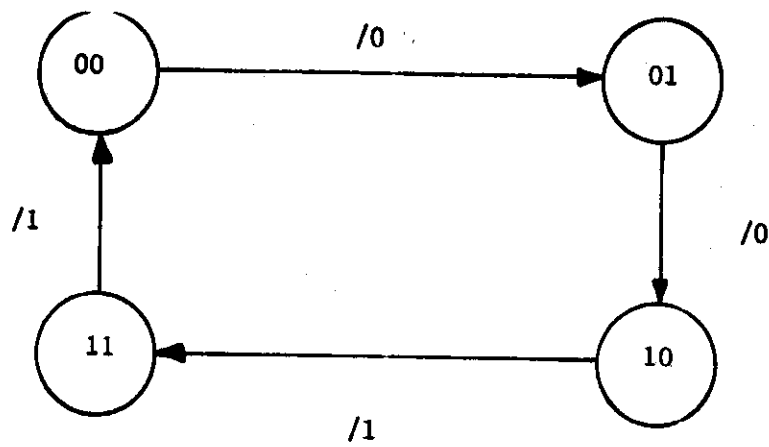
Fig. 8.4 DIAGRAMA DE TRANSICION AL QUE NO SE HA ASIGNADO TODOS LOS ESTADOS.

EJEMPLO 8.1

Diseñe el sistema secuencial que aparece en la fig. 8.4 asignando los estados restantes en código binario natural ascendente.

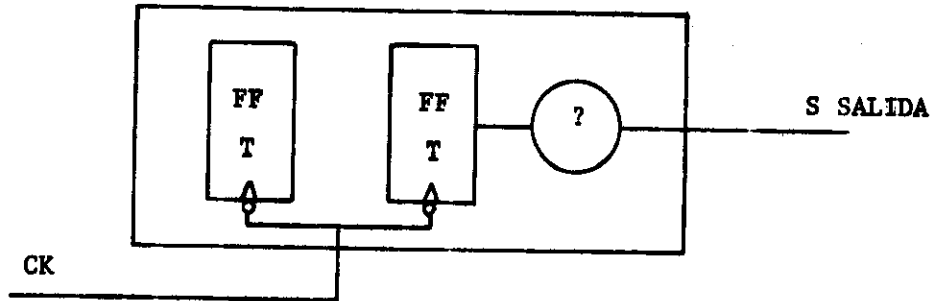
1- ESPECIFICACION DEL SISTEMA

a) Para especificar el sistema usaremos un diagrama de transición.



2- DETERMINAR EL NUMERO DE ENTRADAS Y SALIDAS

- a) No existe más entrada que el CK
- b) Las salidas es una sola y es verdadera en 2 ocasiones



3- SE USARAN FF T

Dejaremos al lector la implementación con otros tipos de FF.

4- TABLA DE ESTADOS

Los subíndices de los estados próximos Q_{1n+1} se omiten por simplicidad.

| ESTADOS PRESENTES $Q_1 Q_0$ | ESTADOS PROXIMOS $Q_1 Q_0$ | SALIDA S | ENTRADAS A LOS FF $T_1 T_0$ |
|--------------------------------|-------------------------------|-------------|--------------------------------|
| 00 | 01 | 0 | 01 |
| 01 | 10 | 0 | 11 |
| 10 | 11 | 1 | 01 |
| 11 | 00 | 1 | 11 |

| $Q_n Q_{n+1}$ | T |
|---------------|---|
| 00 | 0 |
| 01 | 1 |
| 10 | 1 |
| 11 | 0 |

Tabla de excitación.

| | | |
|----|----|---|
| | Q1 | |
| | 0 | 1 |
| Q0 | 0 | 0 |
| | 1 | 1 |

a)

| | | |
|----|----|---|
| | Q1 | |
| | 0 | 1 |
| Q0 | 0 | 1 |
| | 1 | 1 |

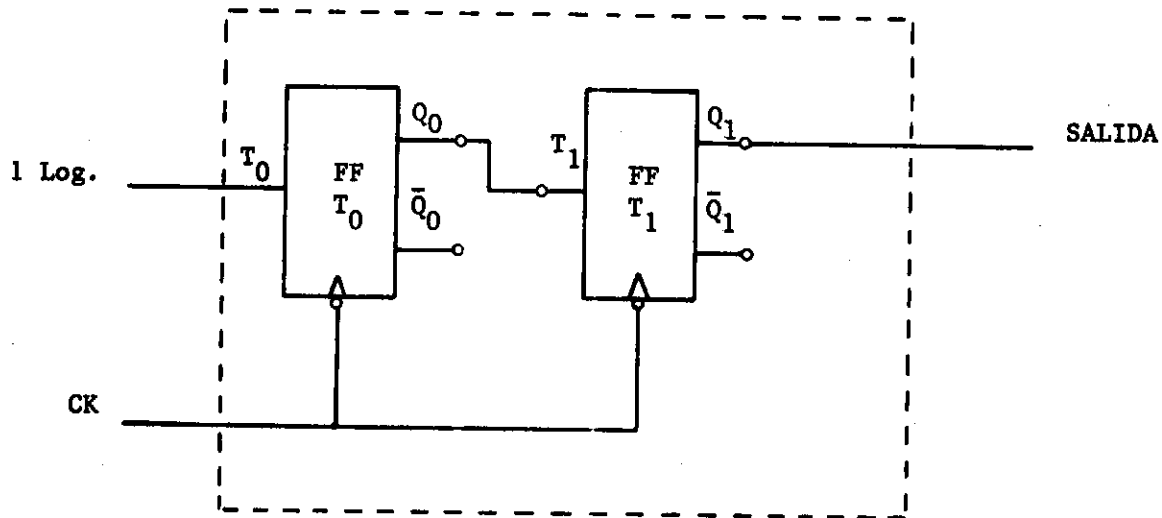
b)

$$a) T_1(Q_1 Q_0) = Q_0$$

$$T_0(Q_1 Q_0) = 1 \text{ log}$$

$$b) S(Q_1 Q_0) = Q_1$$

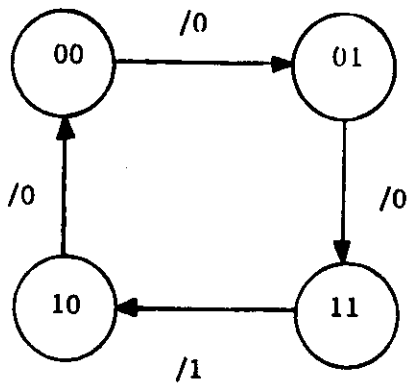
IMPLEMENTACION



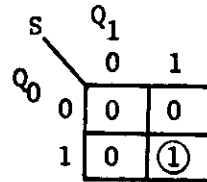
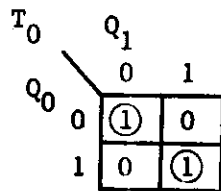
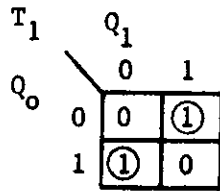
No existe una reglamentación formal para el acomodo de los FLIP-FLOPS y compuertas en un sistema secuencial, la distribución anterior se hizo -- para la simplificación del circuito.

EJEMPLO 8.2

Dé una asignación diferente al sistema secuencial de 4 estados de la fig. 8.4



| ESTADOS PRESENTES Q_1 Q_0 | | ESTADOS PROXIMOS Q_1 Q_0 | | SALIDA S | ENTRADAS A LOS FF T_1 T_0 | |
|----------------------------------|--|---------------------------------|--|-------------|----------------------------------|---|
| 00 | | 01 | | 0 | 0 | 1 |
| 01 | | 11 | | 0 | 1 | 0 |
| 10 | | 00 | | 0 | 1 | 0 |
| 11 | | 10 | | 1 | 0 | 1 |

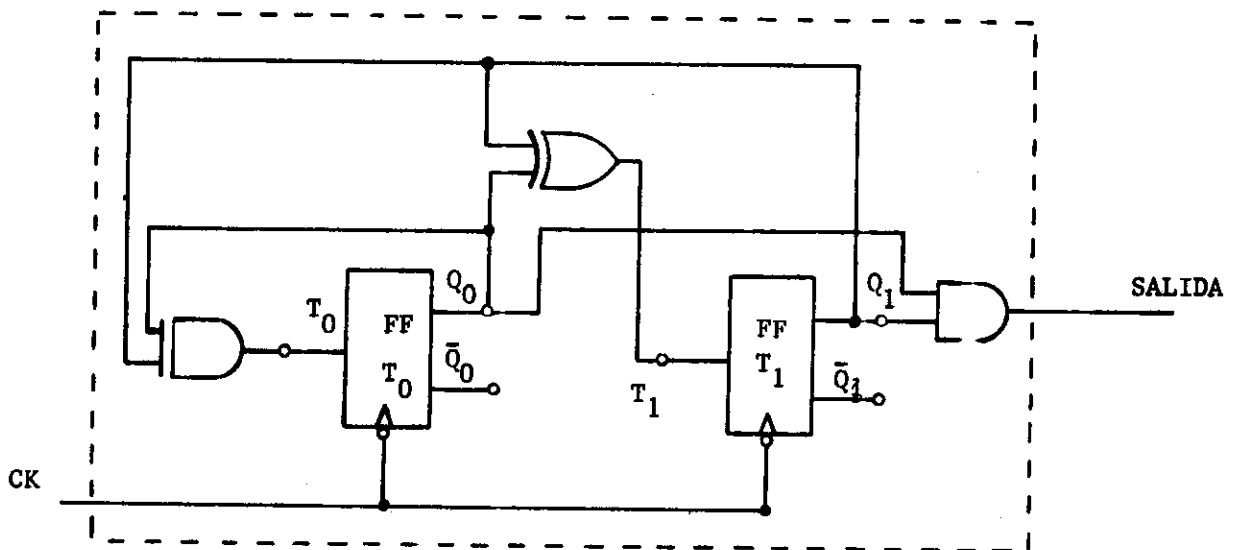


$$T_1(Q_1, Q_0) = Q_1 \bar{Q}_0 + \bar{Q}_1 Q_0 = Q_1 \oplus Q_0$$

$$T_0(Q_1, Q_0) = \bar{Q}_1 \bar{Q}_0 + Q_1 Q_0 = Q_1 \odot Q_0$$

$$S(Q_1, Q_0) = Q_1 \cdot Q_0$$

IMPLEMENTACION



8.5 SISTEMAS SECUENCIALES QUE NO ESTAN COMPLETAMENTE ESPECIFICADOS

La no completa especificación también se presenta en los sistemas secuenciales. Cuando el número de estados NO es un entero potencia de 2 se dice que el sistema no está completamente especificado. En este caso la asignación de algún estado no se encuentra especificada por dos motivos.- El primero es que el estado en cuestión sea un estado aislado y el segundo que se trate de un estado transitorio (un estado transitorio es aquel al que no se puede entrar por medio de las entradas de control del sistema pero si se puede salir de él). En cualquier caso un estado no especificado se manifiesta como un don't care para el diseño del sistema.

En la fig. 8.5 se presentan dos diagramas de transición en uno de los cuales el estado 11 no se presenta y en el otro aparece como un estado transitorio.

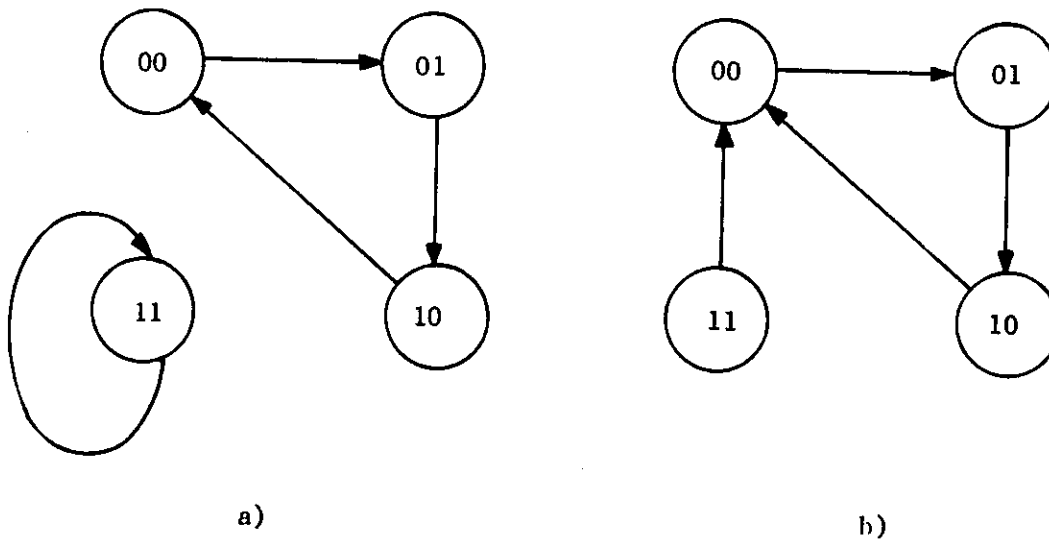
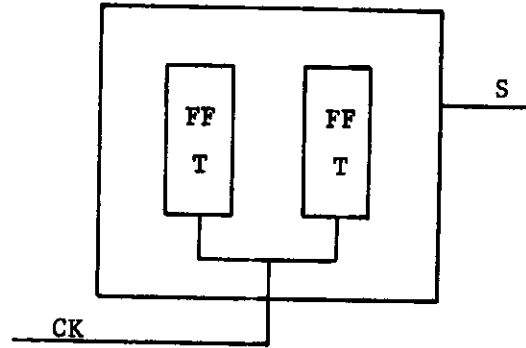
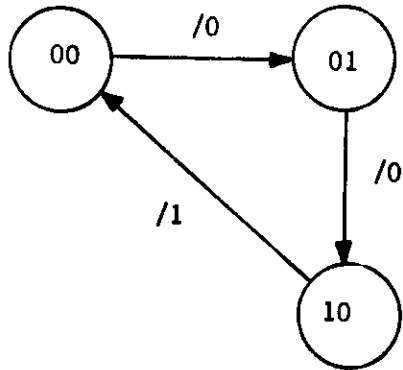


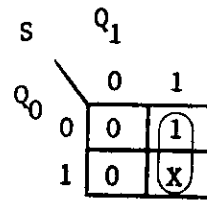
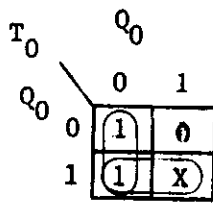
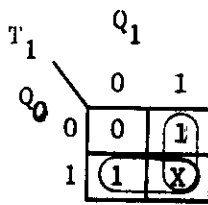
Fig. 8.5 SISTEMAS SECUENCIALES QUE NO ESTAN COMPLETAMENTE ESPECIFICADOS-
a) ESTADO AISLADO , b) ESTADO TRANSITORIO.

EJEMPLO 8.3

Diseñe el siguiente sistema secuencial que no está completamente especificado.



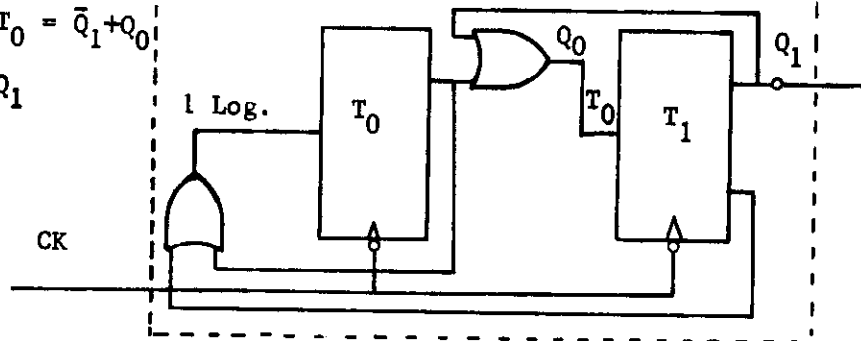
| ESTADOS PRESENTES $Q_1 Q_0$ | ESTADOS PROXIMOS $Q_1 Q_0$ | SALIDA | ENTRADA A LOS FF $T_1 T_0$ |
|--------------------------------|-------------------------------|--------|-------------------------------|
| 00 | 01 | 0 | 01 |
| 01 | 10 | 0 | 11 |
| 10 | 00 | 1 | 10 |
| 11 | 00 | X | XX |



$$T_1(Q_1 Q_0) = T_1 = Q_1 + Q_0$$

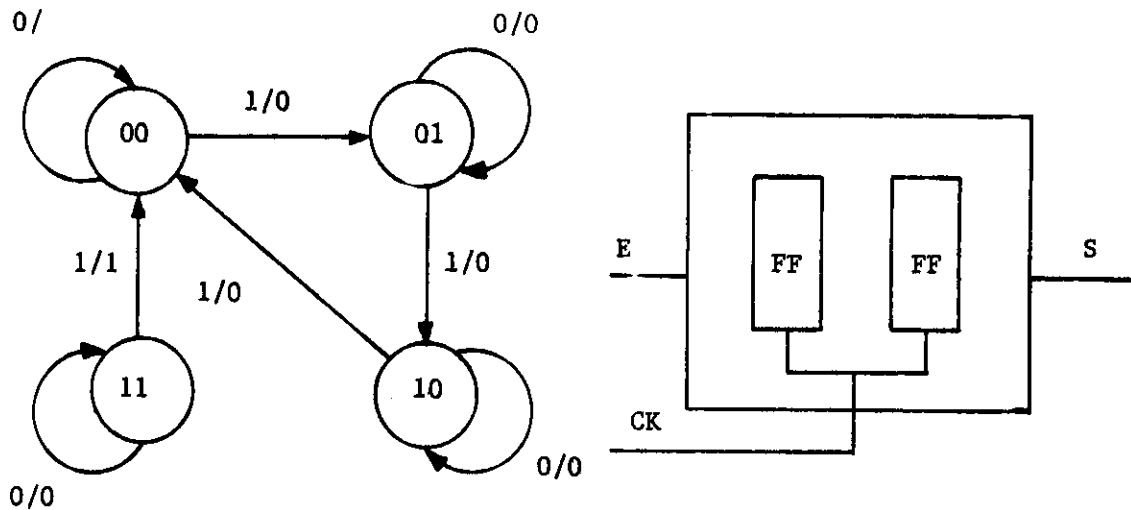
$$T_0(Q_1 Q_0) = T_0 = \bar{Q}_1 + Q_0$$

$$S(Q_1 Q_0) = S = Q_1$$



EJEMPLO 8.4

Diseñe el sistema secuencial descrito mediante el siguiente diagrama de transición.



| ESTADOS PRESENTES $Q_1 Q_0$ | E = 0 | | E = 1 | | | |
|--------------------------------|-------------------------------|-------------|-----------------------------|-------------------------------|-------------|--------------------------------|
| | ESTADOS PROXIMOS $Q_1 Q_0$ | SALIDA S | ENTRADAS AL FF $T_1 T_0$ | Estados Próximos $Q_1 Q_0$ | SALIDA S | ENTRADAS A LOS FF $T_1 T_0$ |
| 00 | 00 | 0 | 00 | 01 | 0 | 01 |
| 01 | 01 | 0 | 00 | 10 | 0 | 11 |
| 10 | 10 | 0 | 00 | 00 | 0 | 10 |
| 11 | 11 | 0 | 00 | 00 | 1 | 11 |

| T_1 | $Q_1 Q_0$ | | | |
|-------|-----------|----|----|----|
| | 00 | 01 | 11 | 10 |
| E 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |

| T_0 | $Q_1 Q_0$ | | | |
|-------|-----------|----|----|----|
| | 00 | 01 | 11 | 10 |
| E 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

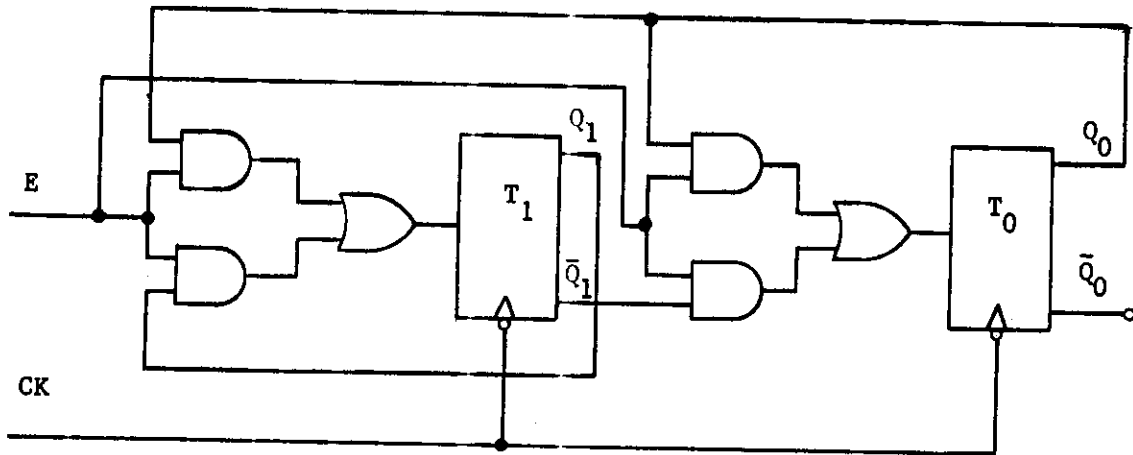
| S | $Q_1 Q_0$ | | | |
|-----|-----------|----|----|----|
| | 00 | 01 | 11 | 10 |
| E 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |

$$T_1(Q_1 Q_0 E) = Q_0 E + Q_1 E$$

$$T_0(Q_1 Q_0 E) = \bar{Q}_1 E + Q_0 E$$

$$S(Q_1 Q_0 E) = Q_1 Q_0 E$$

IMPLEMENTACION



Existen ocasiones en que se incrementa considerablemente la cantidad de líneas que van hacia un FF, en ese caso las entradas pueden quedar indicadas sobre la parte combinacional solamente como se muestra en la fig. 8.6.

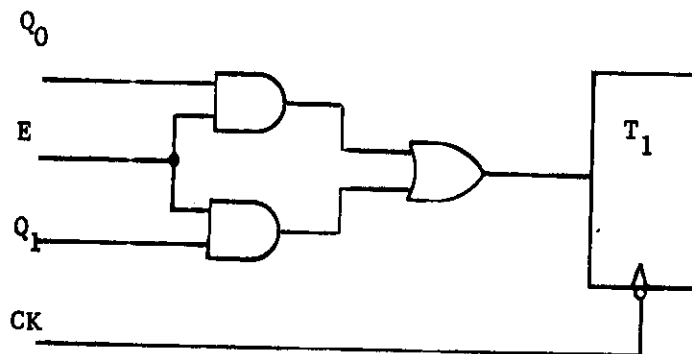


Fig. 8.6 PARA SIMPLIFICAR EL DIBUJO DE LAS CONEXIONES A UNA ENTRADA DE UN FF, PUEDEN INDICARSE SOBRE LAS COMPUERTAS.

8.6 REDUCCION DE ESTADOS

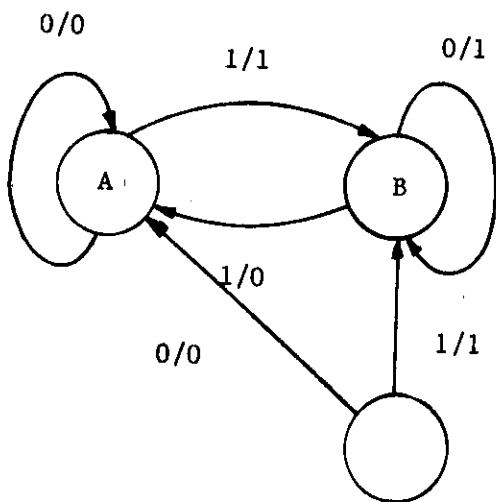
En un sistema digital suelen presentarse estados redundantes o prescindibles que al eliminarlos se pueden reducir la cantidad de FLIP-FLOPS - necesarios para su implementación. La técnica de reducción se basa precisamente en estos estados redundantes a los que se les da el nombre de estados equivalentes. Al eliminar estos estados el sistema deberá trabajar con la misma eficiencia. Los estados equivalentes no son fácil de identificar.

El primer paso para determinar la posibilidad de alguna reducción es formar la tabla de estados que describa el comportamiento del sistema. Además de los estados es necesario incluir las salidas del sistema.

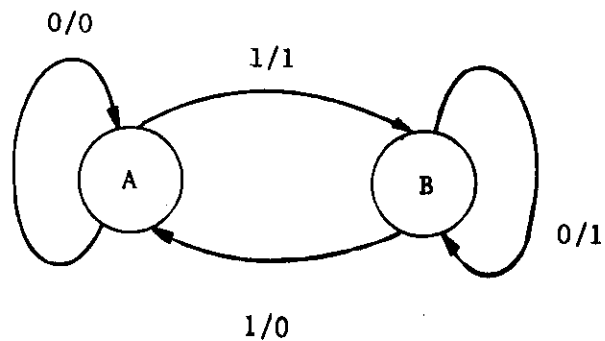
Una vez formada la tabla, hay que identificar a los estados equivalentes. DOS ESTADOS SON EQUIVALENTES SI CUALQUIERA DE SUS ESTADOS ESPECIFICADOS Y SEÑALES DE SALIDA, CORRESPONDEN AL MISMO ESTADO, PARA TODAS LAS COMBINACIONES DE LAS ENTRADAS.

EJEMPLO 8.5

Determine si los dos sistemas secuenciales cuyos diagramas de transición se muestra a continuación son equivalentes.



a)



b)

| ESTADO PRESENTE | E = 0 | | E = 1 | |
|-----------------|----------------|--------|------------------|--------|
| | ESTADO PROXIMO | SALIDA | ESTADOS PROXIMOS | SALIDA |
| A | A | 0 | B | 1 |
| B | B | 1 | A | 0 |
| C | A | 0 | B | 1 |

TABLA DE ESTADOS PARA EL SISTEMA a)

| ESTADO PRESENTE | E = 0 | | E = 1 | |
|-----------------|----------------|--------|----------------|--------|
| | ESTADO PROXIMO | SALIDA | ESTADO PROXIMO | SALIDA |
| A | A | 0 | B | 1 |
| B | B | 1 | A | 0 |

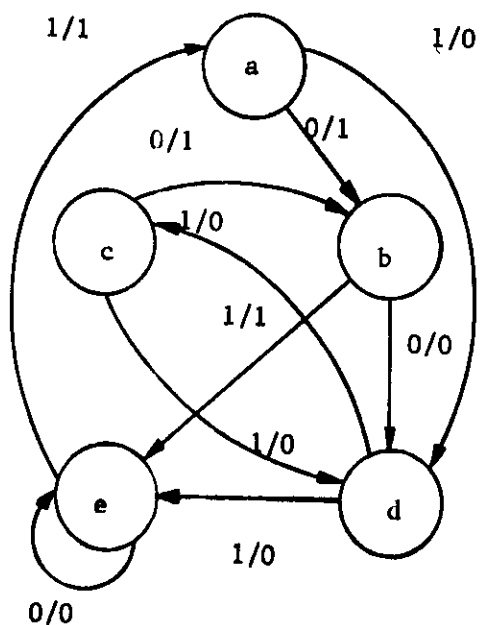
TABLA DE ESTADOS DEL SISTEMA b)

Se puede observar en la tabla de estados de a) que el estado A es idéntico al estado C (sus salidas y estados próximos son los mismos). Por lo tanto este estado es un estado equivalente al estado A. Y es necesario.

El sistema a) necesita 2 FF para implementarse y el sistema b) solo 1 FF.

EJEMPLO 8.6

Analice el siguiente sistema secuencial y determine si tiene estados equivalentes.



| EDO. PRESENTE | E = 0 | | E = 1 | |
|---------------|------------|------|------------|------|
| | EDO. PROX. | SAL. | EDO. PROX. | SAL. |
| a | b | 1 | d | 0 |
| b | d | 0 | e | 1 |
| c | b | 1 | d | 0 |
| d | c | 0 | e | 0 |
| e | e | 0 | a | 1 |

Inspeccionando la tabla de estados podemos notar que el estado a y c son equivalentes puesto que tienen el mismo estado próximo y las mismas salidas para cada combinación de la entrada E.

8.7 CONTADORES

Una de las principales aplicaciones de un sistema secuencial es la de los CONTADORES. Un CONTADOR se define como un sistema digital de propósito especial diseñado para contar el número de transiciones a la entrada del bloque (CK). Estos sistemas se usan para desarrollar tareas tales como, un simple conteo, controladores de secuencia de algunos eventos, divisores de frecuencia, relojes digitales, etc. La salida suele representarse en algún código binario, BINARIO NATURAL (8-4-2-1), GRAY, exceso-3, o BCD, el conteo puede ser ascendente o descendente, sincrónico o asincrónico y pueden también ser programables.

Las señales de entrada a un contador son pocas, pueden tener como entrada únicamente el reloj (CK), así mismo el RESET (reestablecer), esta señal es asincrónica y se usa para forzar al contador a regresar su estado de inicio. Las salidas de un contador generalmente son las Q de los FLIP-FLOPS.

Partiendo de que un contador puede tener como única entrada el CK, el diagrama de transición en sistema del ejemplo 8.1 es precisamente un-

contador de 2 bit que cuenta en binario natural ascendente desde 00 hasta 11. (de 0_{10} a 3_{10}).

A continuación veremos algunos ejemplos de diseño de contadores sín cronos.

EJEMPLO 8.7

Diseñe un contador, cuya entrada sea únicamente el reloj y su salida las Q de los FLIP-FLOPS, que cuente en código binario natural ascendente de 3 bit. (De 0_{10} a 7_{10}).

- 1- Un diagrama de transición puede ayudarnos a especificar el sistema.
- 2- No existe más entrada que el CK y las salidas son las Q.
- 3- El sistema tiene 3 FF.

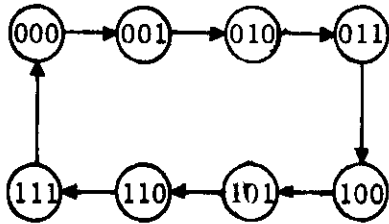


DIAGRAMA DE TRANSICION

| ESTADOS PRESENTES $Q_2 Q_1 Q_0$ | ESTADOS PROXIMOS $Q_2 Q_1 Q_0$ | ENTRADAS A LOS FF $T_2 T_1 T_0$ | | |
|------------------------------------|-----------------------------------|------------------------------------|-------|-------|
| | | T_2 | T_1 | T_0 |
| 000 | 001 | 0 | 0 | 1 |
| 001 | 010 | 0 | 0 | 1 |
| 010 | 011 | 0 | 0 | 1 |
| 011 | 100 | 0 | 1 | 1 |
| 100 | 101 | 0 | 1 | 1 |
| 101 | 110 | 0 | 1 | 1 |
| 110 | 111 | 0 | 1 | 1 |
| 111 | 000 | 1 | 1 | 1 |

TABLA DE ESTADOS

| T_2 | $Q_2 Q_1$ | | | |
|-------|-----------|----|----|----|
| | 00 | 01 | 11 | 10 |
| Q_0 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 1 | 0 |

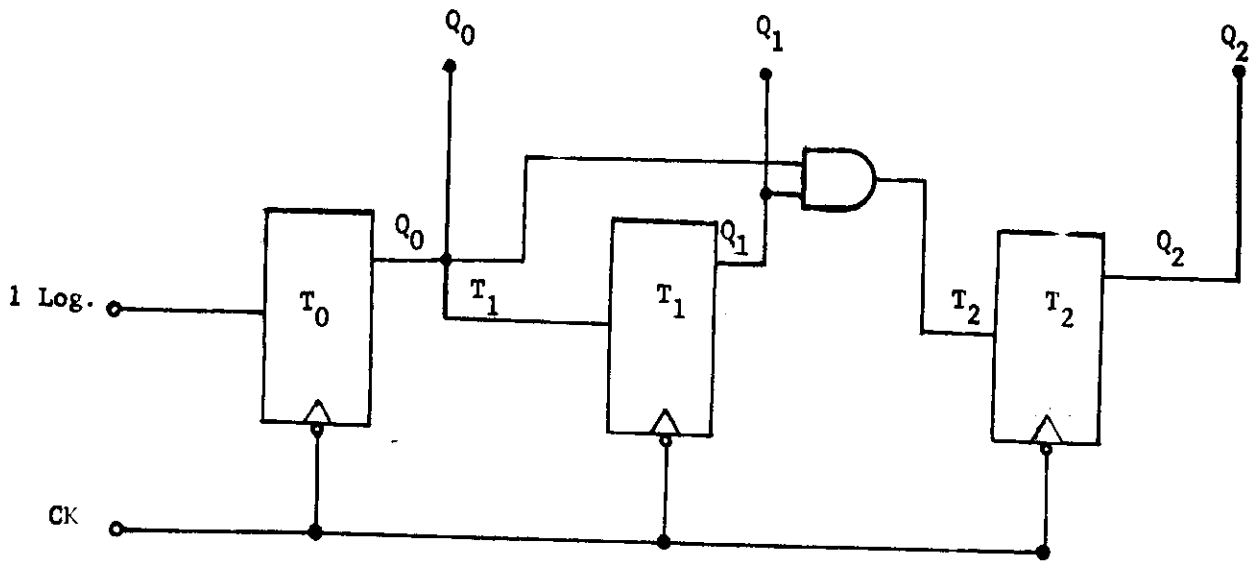
$$T_2(Q_2 Q_1 Q_0) = Q_1 Q_0$$

| T_1 | $Q_2 Q_1$ | | | |
|-------|-----------|----|----|----|
| | 00 | 01 | 11 | 10 |
| Q_0 | 0 | 0 | 0 | 0 |
| | 1 | 1 | 1 | 1 |

$$T_1(Q_2 Q_1 Q_0) = Q_0$$

$$T_0(Q_2 Q_1 Q_0) = 1 \text{ log.}$$

IMPLEMENTACION



8.8 RESET

La entrada de RESET a un contador lo obliga a regresar al estado -- original en forma asincrónica (independientemente al CK).

Suponiendo que el estado de inicio u origen de sistema anterior es el 000, entonces la señal de RESET debe conectarse a los CLEAR de cada FF para lograr que carguen ceros. Como se muestra en la fig. 8.7

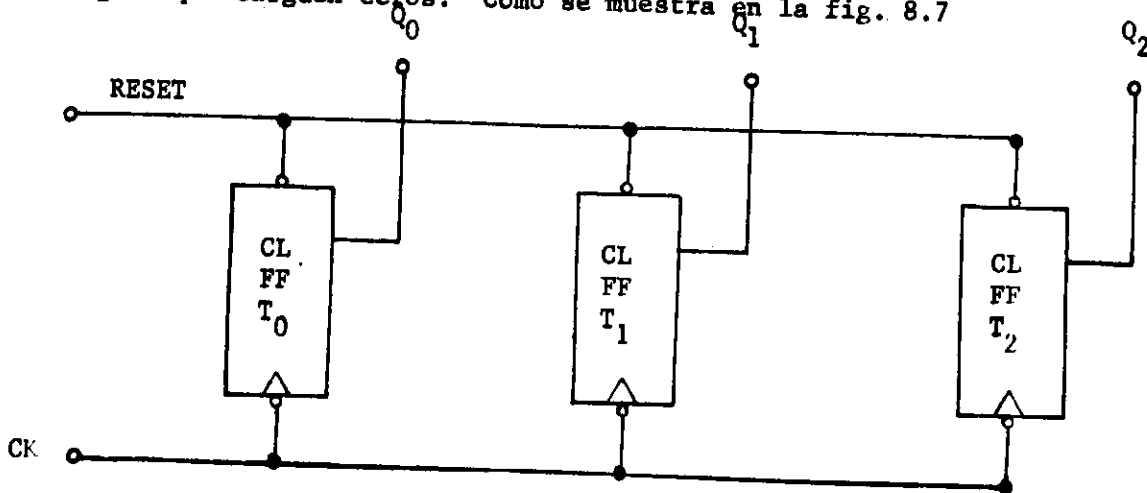


Fig. 8.7 CONEXION DE LA LINEA DE RESET PARA EL CONTADOR DEL EJEMPLO 8.7

Si el estado de inicio es el 111 la línea de RESET se conecta a los PRESET de cada FF. Si el estado de inicio tiene unos y ceros por ejemplo el estado 110, la entrada de RESET de los FF de mayor peso va al PRESET - y la del FF de menor peso al CLEAR. Fig. 8.8

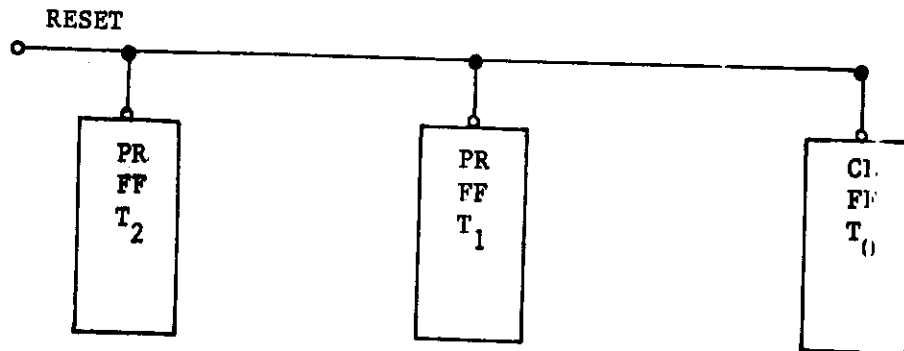


Fig. 8.8 CONEXION PARA RESTABLECER EN EL ESTADO 110.

8.9 CONTADORES AUTO RESTABLECIENTES

Puede presentarse el caso de un contador que no tenga 2^n FF-ESTADOS es decir que alguno de sus estados se encuentre aislado. Al diseñar este tipo de contadores debemos tomar en cuenta que podemos caer en ese estado aislado, por perturbaciones externas o al momento de energizar el circuito. En tal caso el contador se puede quedar "atorado" en ese estado.

Una solución a este problema es tomar el estado aislado como un estado TRANSITORIO, de tal forma que si llega a caer en estos estados (por un motivo no aclarado en el diseño) regresa al lazo principal de conteo - sin necesidad de aplicar RESET. A estos sistemas se les conoce como contadores auto restablecientes. Fig. 8.9

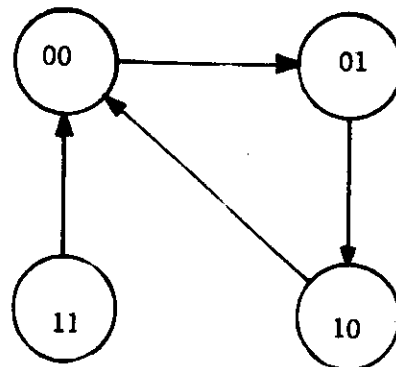


Fig. 8.9 CONTADOR AUTO RESTABLECIENTE

EJEMPLO 8.8

Diseñe un contador que cuente en código BCD este contador tiene 10-estados del 0000_2 al 1001_2 , los demás estados para las combinaciones de 4 bits del estado 1010_2 al 1111_2 podemos tomarlos como estados aislados e incluir DON'T CARES en los mapas, al tomarse esas Xs como UNOS o como CEROS existe la posibilidad de que los estados del 10 al 15 no queden del todo aislados y regresen al lazo principal de conteo. Esto es sin que se declaren como estados transitorios.

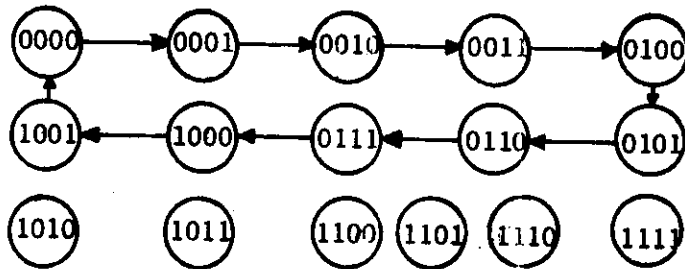
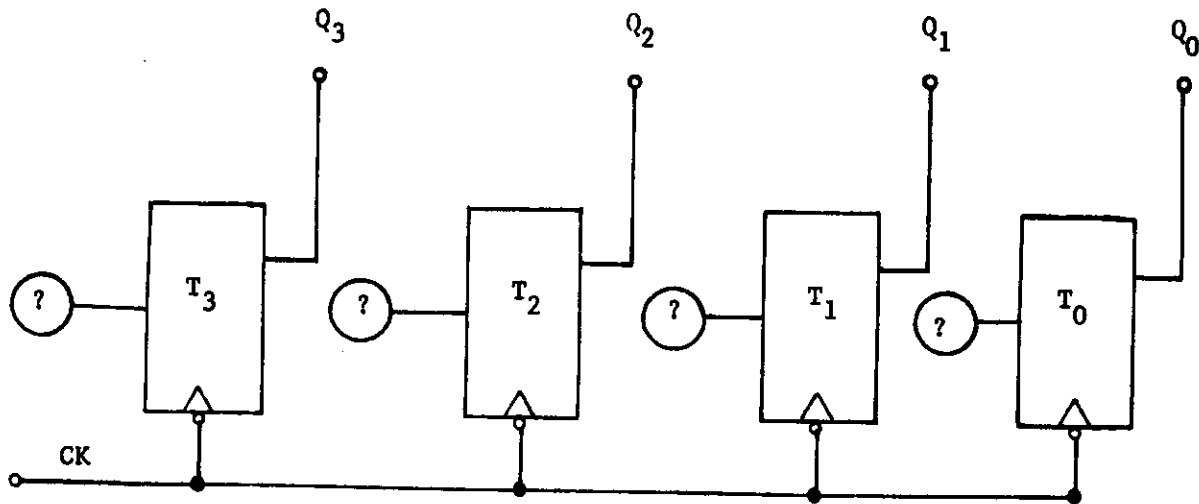


DIAGRAMA DE TRANSICION



EL SISTEMA NO TIENE MAS ENTRADA QUE EL CK

| ESTADOS PRESENTES $Q_3Q_2Q_1Q_0$ | ESTADOS PROXIMOS $Q_3Q_2Q_1Q_0$ | ENTRADAS A LOS FF $T_3T_2T_1T_0$ |
|-------------------------------------|------------------------------------|-------------------------------------|
| 0000 | 0001 | 0 0 0 1 |
| 0001 | 0010 | 0 0 1 1 |
| 0010 | 0011 | 0 0 0 1 |
| 0011 | 0100 | 0 1 1 1 |
| 0100 | 0101 | 0 0 0 1 |
| 0101 | 0110 | 0 0 1 1 |
| 0110 | 0111 | 0 0 0 1 |
| 0111 | 1000 | 1 1 1 1 |
| 1000 | 1001 | 0 0 0 1 |
| 1001 | 0000 | 1 0 0 1 |
| 1010 | | X X X X |
| 1011 | | X X X X |
| 1100 | | X X X X |
| 1101 | | X X X X |
| 1110 | | X X X X |
| 1111 | | X X X X |

TABLA DE ESTADOS

| T_3 | Q_3Q_2 | | | |
|----------|----------|----|----|----|
| Q_1Q_0 | 00 | 01 | 11 | 10 |
| 00 | | | X | |
| 01 | | | X | 1 |
| 11 | | 1 | X | X |
| 10 | | | X | X |

| T_2 | Q_3Q_2 | | | |
|----------|----------|----|----|----|
| Q_1Q_0 | 00 | 01 | 11 | 10 |
| 00 | | | X | |
| 01 | | | X | |
| 11 | 1 | 1 | X | X |
| 10 | | | X | X |

$$T_3(Q_3Q_2Q_1Q_0) = Q_3Q_0 + Q_2Q_1Q_0$$

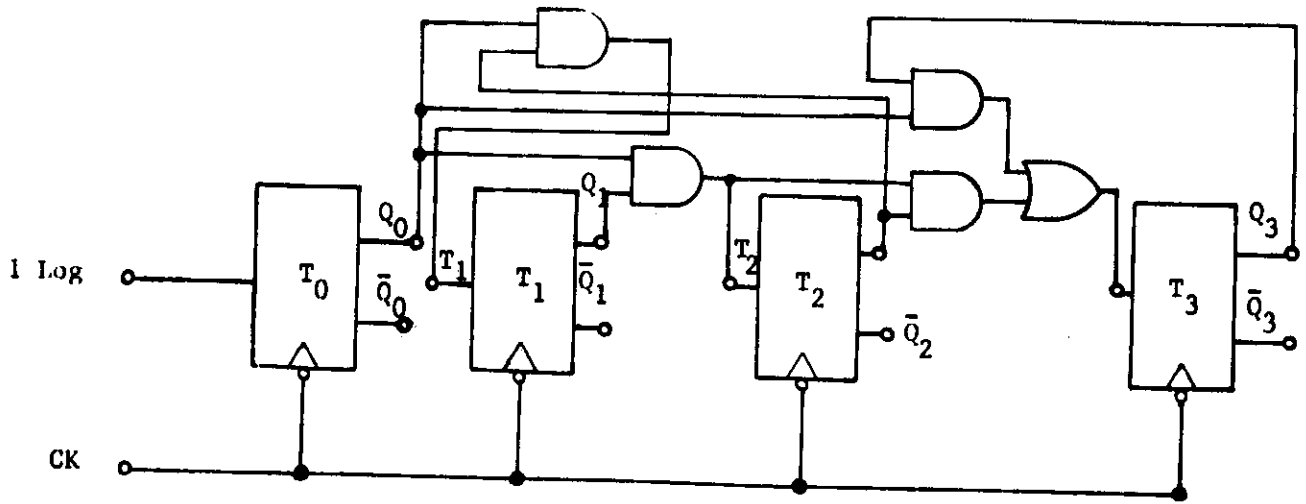
$$T_2(Q_3Q_2Q_1Q_0) = Q_1Q_0$$

$$T_1(Q_3Q_2Q_1Q_0) = Q_0Q_2$$

$$T_0(Q_3Q_2Q_1Q_0) = 1 \text{ log.}$$

| T_1 | Q_2Q_3 | | | |
|----------|----------|----|----|----|
| Q_1Q_0 | 00 | 01 | 11 | 10 |
| 00 | | | X | |
| 01 | 1 | 1 | X | |
| 11 | 1 | 1 | X | X |
| 10 | | | X | X |

IMPLEMENTACION



EJEMPLO 8.9

Diseñar un contador que cuente en código binario natural ascendente de 4 bits.

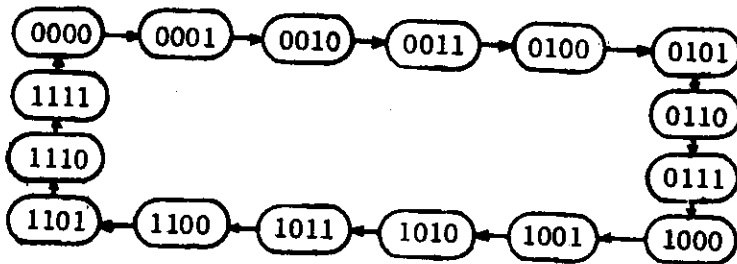


DIAGRAMA DE TRANSICION

| ESTADOS PRESENTES $Q_3Q_2Q_1Q_0$ | ESTADOS PROXIMOS $Q_3Q_2Q_1Q_0$ | ENTRADAS A LOS FF $T_3T_2T_1T_0$ |
|-------------------------------------|------------------------------------|-------------------------------------|
| 0000 | 0001 | 0001 |
| 0001 | 0010 | 0011 |
| 0010 | 0011 | 0001 |
| 0011 | 0100 | 0111 |
| 0100 | 0101 | 0001 |
| 0101 | 0110 | 0011 |
| 0110 | 0111 | 0001 |
| 0111 | 1000 | 1111 |
| 1000 | 1001 | 0001 |
| 1001 | 1010 | 0011 |
| 1010 | 1011 | 0001 |
| 1011 | 1100 | 0111 |
| 1100 | 1101 | 0001 |
| 1101 | 1110 | 0011 |
| 1110 | 1111 | 0001 |
| 1111 | 0000 | 1111 |

TABLA DE ESTADOS

| T_3 | Q_3Q_2 | 00 | 01 | 11 | 10 |
|----------|----------|----|----|----|----|
| Q_1Q_0 | 00 | | | | |
| 01 | | | | | |
| 11 | | | 1 | 1 | |
| 10 | | | | | |

| T_2 | Q_3Q_2 | 00 | 01 | 11 | 10 |
|----------|----------|----|----|----|----|
| Q_1Q_0 | 00 | | | | |
| 01 | | | | | |
| 11 | | 1 | 1 | 1 | 1 |
| 10 | | | | | |

| T_1 | Q_3Q_2 | 00 | 01 | 11 | 10 |
|----------|----------|----|----|----|----|
| Q_1Q_0 | 00 | | | | |
| 01 | | 1 | 1 | 1 | 1 |
| 11 | | 1 | 1 | 1 | 1 |
| 10 | | | | | |

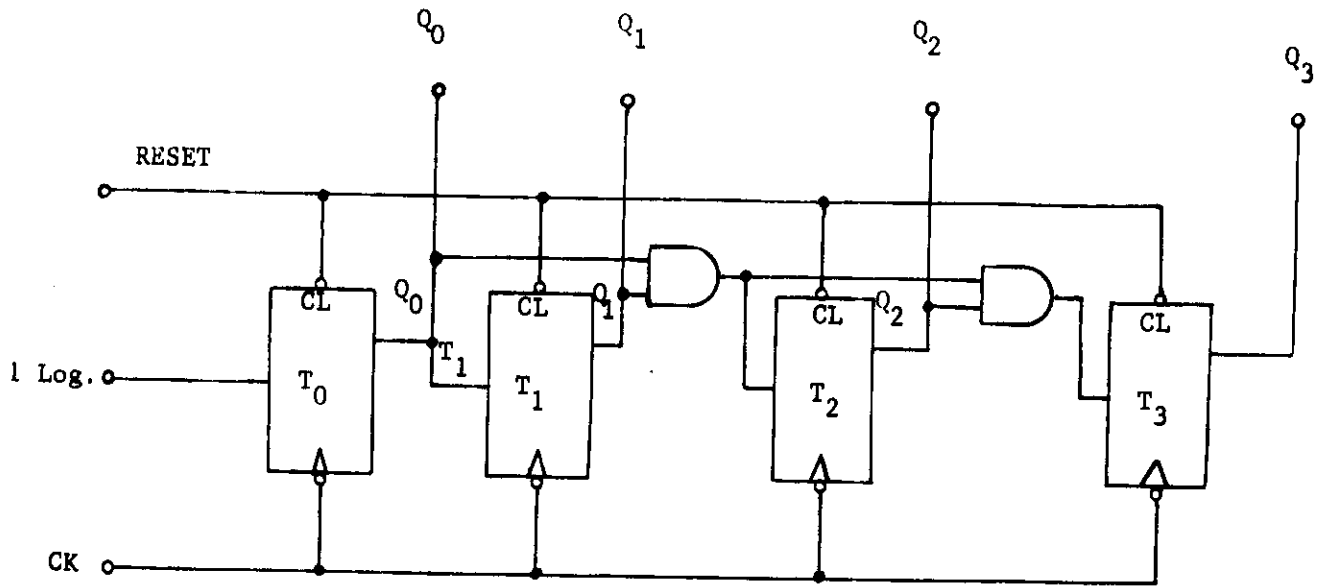
$$T_3(Q_3Q_2Q_1Q_0) = Q_2Q_1Q_0$$

$$T_2(Q_3Q_2Q_1Q_0) = Q_1Q_0$$

$$T_1(Q_3Q_2Q_1Q_0) = Q_0$$

$$T_0(Q_3Q_2Q_1Q_0) = 1 \text{ log.}$$

IMPLEMENTACION

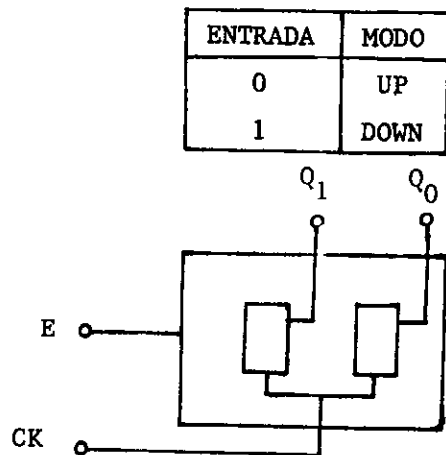
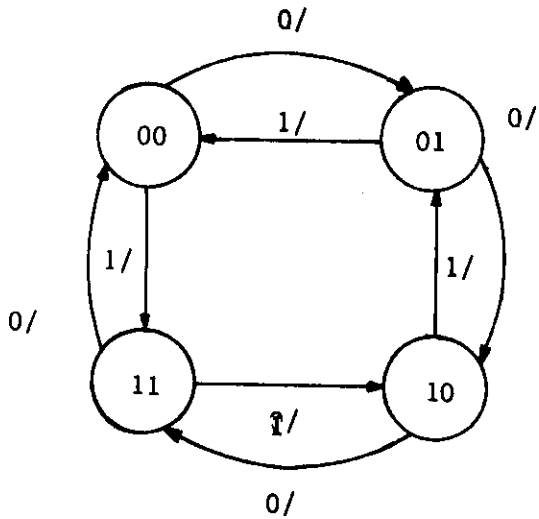


8.10 CONTADORES ASCENDENTE/DESCENDENTE

Los contadores ascendente/descendente también conocidos como UP/DOWN son contadores con una o más entradas de control que determinan el sentido de la generación de la secuencia de salida. Para diseñar este contador es necesario incluir en el diagrama de transición la entrada como se muestra en el ejemplo 8.10.

EJEMPLO 8.10

Diseñar un contador UP/DOWN de 2 bit, en código binario natural. El estado inicial es 00.



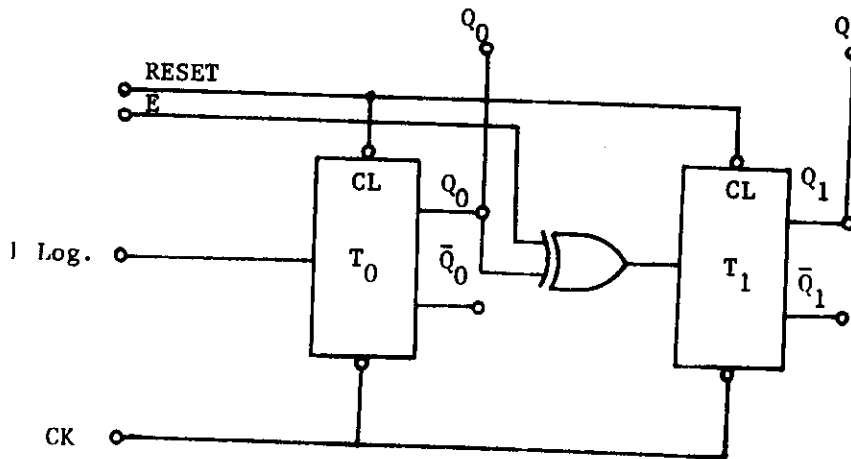
| ESTADOS PRESENTES $Q_1 Q_0$ | $E = 0$ | | $E = 1$ | |
|--------------------------------|-------------------------------|--------------------------------|-------------------------------|--------------------------------|
| | ESTADOS PROXIMOS $Q_1 Q_0$ | ENTRADAS A LOS FF $T_1 T_0$ | ESTADOS PROXIMOS $Q_1 Q_0$ | ENTRADAS A LOS FF $T_1 T_0$ |
| 00 | 01 | 01 | 11 | 11 |
| 01 | 10 | 11 | 00 | 01 |
| 10 | 11 | 01 | 01 | 11 |
| 11 | 00 | 11 | 10 | 01 |

| T_1 | $Q_1 Q_0$ | | | |
|-------|-----------|----|----|----|
| E | 00 | 01 | 11 | 10 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |

$$T_1(Q_1, Q_0, E) = Q_0 \bar{E} + \bar{Q}_0 E$$

$$T_0(Q_1, Q_0, E) = 1 \text{ log.}$$

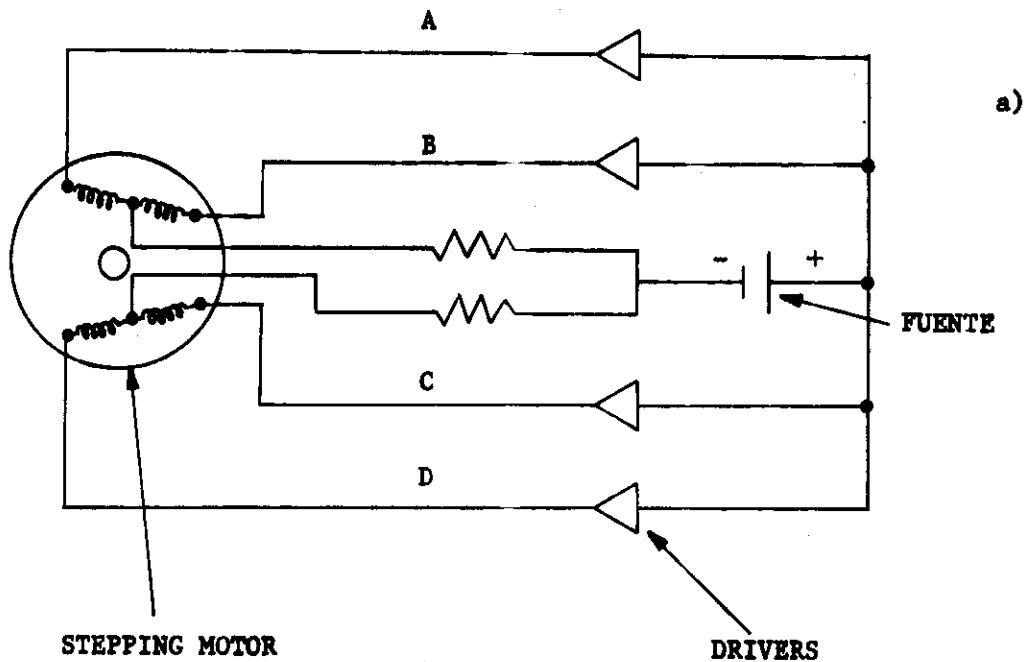
IMPLEMENTACION



8.11 EJEMPLOS DE DISEÑO SECUENCIAL

SECUENCIADOR PARA UN STEPPING MOTOR

Un Stepping Motor es un dispositivo que opera con corriente directa — conmutada. Según el fabricante, la flecha del motor avanza 200 pasos por revolución (1.8° por paso) cuando se usa una secuencia de entrada en (A, B, C y D) de 4 pasos, o 400 pasos por revolución (0.9° por paso) cuando se usa una secuencia de entrada de 8 pasos. Existen desde luego motores con diferentes pasos por revolución, sin embargo estos son los más comunes. En la fig. 8.10 aparece el símbolo de un Stepping Motor junto con sus Drivers — de conmutación o switcheo, se puede observar también las tablas para la — secuencia de entrada de 4 y 8 Pasos



| PASO | A | B | C | D |
|------|-----|-----|-----|-----|
| 1 | ON | OFF | ON | OFF |
| 2 | ON | OFF | OFF | ON |
| 3 | OFF | ON | OFF | ON |
| 4 | OFF | ON | ON | OFF |
| 1 | ON | OFF | ON | OFF |

b)

| PASO | A | B | C | D |
|------|-----|-----|-----|-----|
| 1 | ON | OFF | ON | OFF |
| 2 | ON | OFF | OFF | OFF |
| 3 | ON | OFF | OFF | ON |
| 4 | OFF | OFF | OFF | ON |
| 5 | OFF | ON | OFF | ON |
| 6 | OFF | ON | OFF | OFF |
| 7 | OFF | ON | ON | OFF |
| 8 | OFF | OFF | ON | OFF |
| 1 | ON | OFF | ON | OFF |

c)

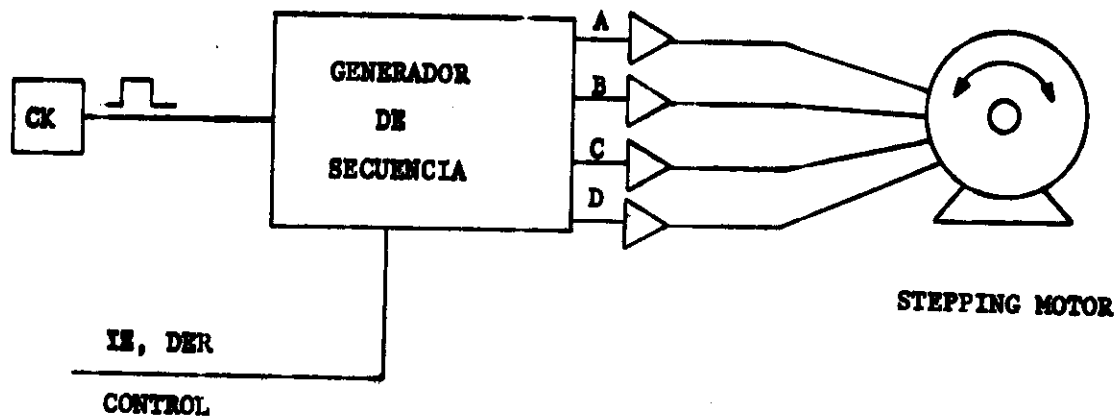
Fig.8.10 a) Símbolo de un Stepping Motor mostrando los Driver's de cada entrada, b) Secuencia de entrada de 4 pasos (200 pasos por revolución). c) Secuencia de entrada de 8 pasos (400 pasos por revolución).

Los Stepping Motors tienen 2 características que los convierten en dispositivos muy útiles. La primera es que necesitan un tiempo muy corto tanto al arrancar como al detenerse. La segunda es que no pierden ningún paso, por ejemplo a un motor de 200 pasos por revolución que se le alimenten 400 pasos completará 2 revoluciones y regresará a su estado de partida.

Las 2 características anteriores hacen que los Stepping Motors tengan una amplia variedad de aplicaciones sobre todo en instrumentos de precisión como graficadores, impresoras, instrumentos ópticos, etc.

Ejemplo 8.11

Diseñe un generador de secuencia bidireccional para un stepping motor de 4 pasos.



1.- El sistema tiene una entrada que controla el sentido de giro.

| CONTROL | GIRO |
|---------|-----------|
| 0 | DERECHA. |
| 1 | IZQUIERDA |

2.- El sistema tiene 4 pasos, por lo tanto existen solo 4 estados.

3.- El número de salidas es igual a 4, A, B, C y D.

| PASO | A | B | C | D |
|------|-----|-----|-----|-----|
| 1 | ON | OFF | ON | OFF |
| 2 | ON | OFF | OFF | ON |
| 3 | OFF | ON | OFF | ON |
| 4 | OFF | ON | ON | OFF |
| 1 | ON | OFF | ON | OFF |



| A | B | C | D | ESTADO |
|---|---|---|---|--------|
| 1 | 0 | 1 | 0 | E_0 |
| 1 | 0 | 0 | 1 | E_1 |
| 0 | 1 | 0 | 1 | E_2 |
| 0 | 1 | 1 | 0 | E_3 |
| 1 | 0 | 1 | 0 | E_0 |

De las tablas anteriores se pueda observar para los valores de A, B, C y D que B es el complemento de A al igual que D es el complemento de C.

Nótese también que pueden tomarse A y C como base para generar la secuencia 11, 10, 00 y 01 que obviamente no corresponden a una asignación de estados, en código binario natural ascendente.

Si asignamos los estados tomando en cuenta la secuencia anterior las salidas de los FF'S corresponderán a las salidas del sistema, $Q_1 = A$, $\bar{Q}_1 = B$, $Q_0 = C$ y $\bar{Q}_0 = D$.

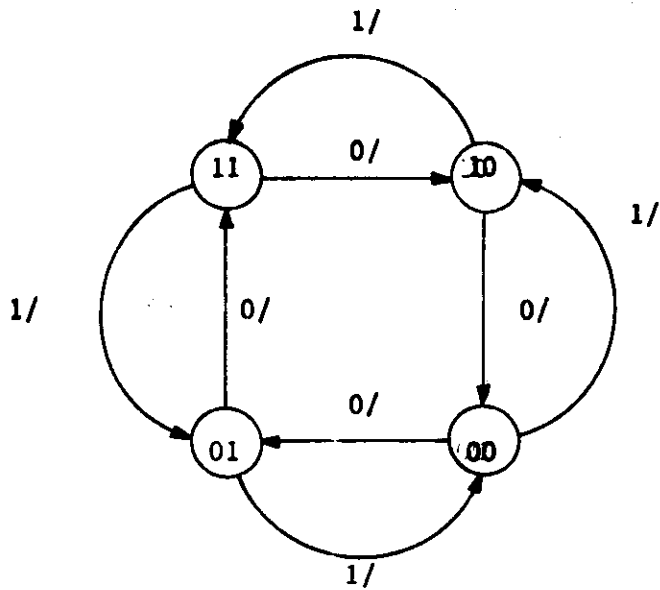


DIAGRAMA DE TRANSICION

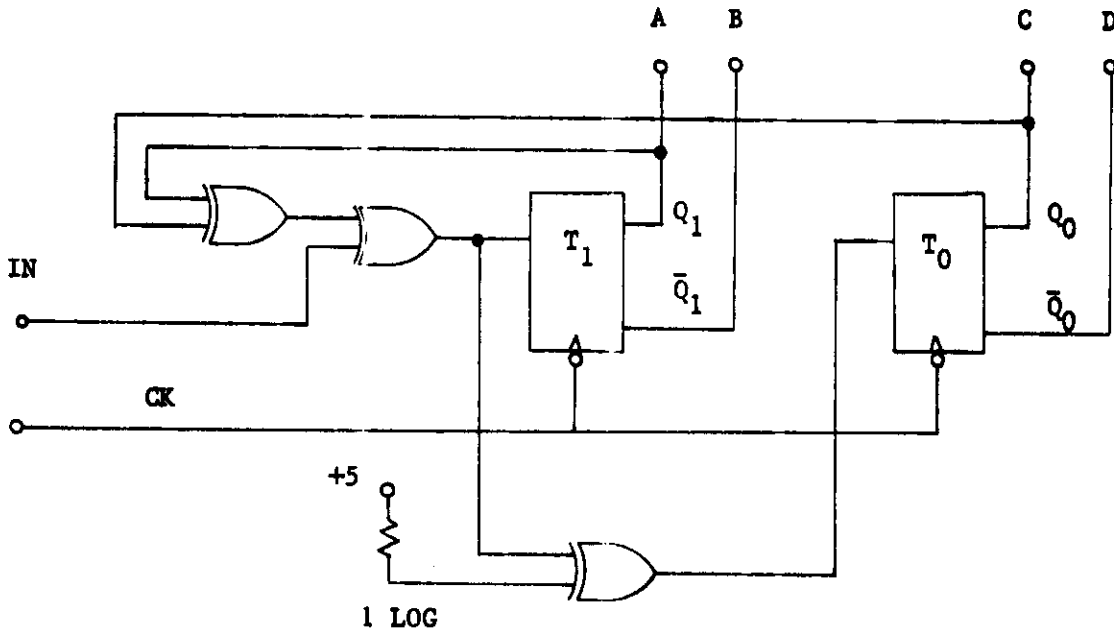
| ESTADOS PRESENTES | IN = 0 | | IN = 1 | |
|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| | ESTADOS PROXIMOS | EXCITACION A LOS FF'S | ESTADOS PROXIMOS | EXCITACION A LOS FF'S |
| Q ₁ Q ₀ | Q ₁ Q ₀ | T ₁ T ₀ | Q ₁ Q ₀ | T ₁ T ₀ |
| 0 0 | 0 1 | 0 1 | 1 0 | 1 0 |
| 0 1 | 1 1 | 1 0 | 0 0 | 0 1 |
| 1 0 | 0 0 | 1 0 | 1 1 | 0 1 |
| 1 1 | 1 0 | 0 1 | 0 1 | 1 0 |

TABLA DE ESTADOS

| | | | | | |
|-------|---|-----------|----|----|----|
| | | $Q_1 Q_0$ | | | |
| | | 00 | 01 | 11 | 10 |
| T_1 | 0 | 0 | 1 | 0 | 1 |
| | 1 | 1 | 0 | 1 | 0 |

| | | | | | |
|-------|---|-----------|----|----|----|
| | | $Q_1 Q_0$ | | | |
| | | 00 | 01 | 11 | 10 |
| T_0 | 0 | 1 | 0 | 1 | 0 |
| | 1 | 0 | 1 | 0 | 1 |

$$\begin{aligned}
 T_1 &= \bar{Q}_1 \bar{Q}_0 \text{ IN} + \bar{Q}_1 Q_0 \overline{\text{IN}} + Q_1 Q_0 \text{ IN} + Q_1 \bar{Q}_0 \overline{\text{IN}} \\
 &= \text{IN} (\bar{Q}_1 \bar{Q}_0 + Q_1 Q_0) + \overline{\text{IN}} (\bar{Q}_1 Q_0 + Q_1 \bar{Q}_0) \\
 &= \text{IN} (\overline{Q_1 \oplus Q_0}) + \overline{\text{IN}} (Q_1 \oplus Q_0) \\
 T_1 &= \text{IN} \oplus Q_1 \oplus Q_0 \\
 T_0 &= \bar{T}_1
 \end{aligned}$$



IMPLEMENTACION

El tanque de la Fig. 8.11 se alimenta por medio de 2 bombas A y B. El gasto de salida nunca será mayor al que proporcionen las dos bombas operando simultáneamente.

El tanque tiene además 2 sensores de nivel $S_2 S_1$:

Ejemplo 8.12

Diseñar un circuito que controle la siguiente secuencia de operación de las bombas.

- 1.- Partiendo de que el tanque se encuentra vacío, se inicia el llenado con ambas bombas al llegar el nivel al sensor S_2 que continúe solo la bomba A, hasta llenar el tanque y entonces desconectarla.
- 2.- Al bajar de nuevo el nivel que conecte la bomba B, si el nivel baja del sensor S_2 que actúen ambas bombas. Al recuperar de nuevo el nivel de S_2 que continúe solo la bomba B hasta llenar el tanque -- y entonces desconectarla.
- 3.- Si baja el nivel nuevamente que se repita la secuencia pero ahora con la bomba A hasta completar el ciclo, después la bomba B y así sucesivamente.
- 4.- El sistema debe ser autorestableciente, es decir, que al energizar el circuito de control automáticamente caiga en el estado que le corresponde, independientemente del nivel.
 - a) Considerar el estado inicial como estado de TANQUE VACIO al --- cual se va por medio del RESET externo o cuando S_2 y S_1 indican ausencia de líquido.

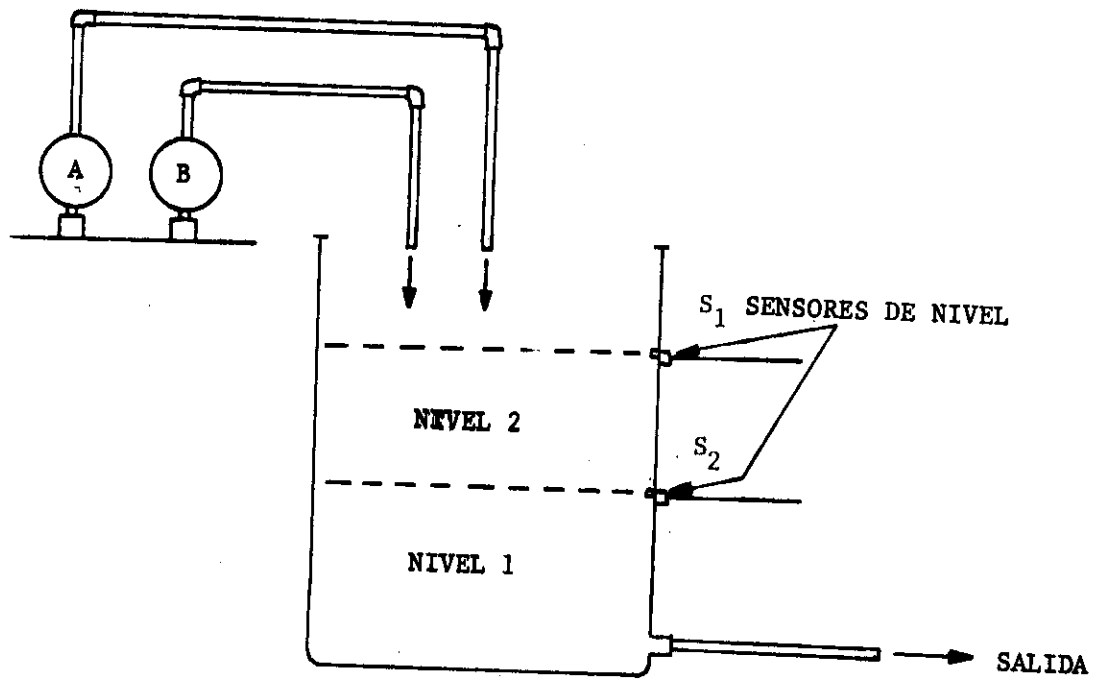


Fig. 8.11 TANQUE ALIMENTADO POR 2 BOMBAS A Y B.

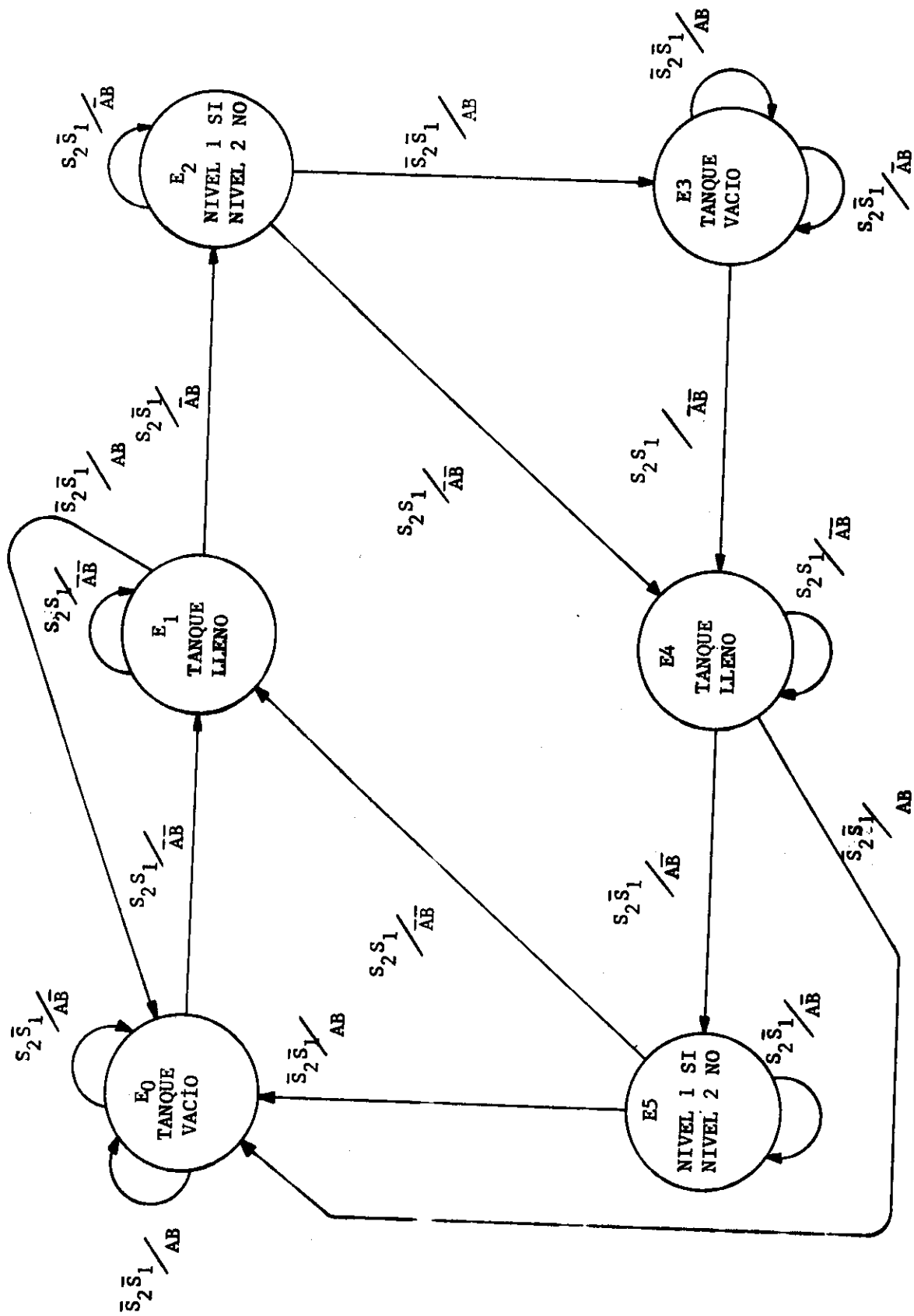


DIAGRAMA DE TRANSICION

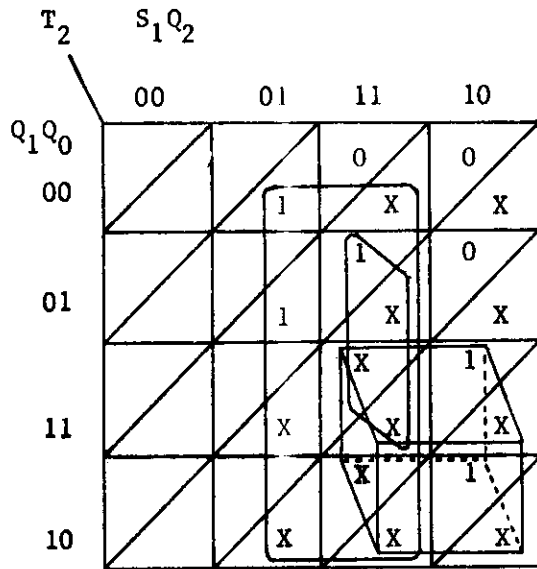
REDUCCION DE ESTADOS

| ESTADO PRESENTE | ESTADOS PROXIMOS | | | | SALIDAS | | | |
|--------------------|------------------|-----|-----|-----|-----------|-----|------------------|------------------|
| | $S_2 S_1$ | | | | $S_2 S_1$ | | | |
| | 0 0 | 0 1 | 1 0 | 1 1 | 0 0 | 0 1 | 1 0 | 1 1 |
| E0 | E0 | X | E0 | E1 | AB | X | $\bar{A}\bar{B}$ | $\bar{A}\bar{B}$ |
| E1 | E0 | X | E2 | E1 | AB | X | $\bar{A}\bar{B}$ | $\bar{A}\bar{B}$ |
| E2 | E3 | X | E2 | E4 | AB | X | $\bar{A}\bar{B}$ | $\bar{A}\bar{B}$ |
| E3 | E3 | X | E3 | E4 | AB | X | $\bar{A}\bar{B}$ | $\bar{A}\bar{B}$ |
| E4 | E0 | X | E5 | E4 | AB | X | $\bar{A}\bar{B}$ | $\bar{A}\bar{B}$ |
| E5 | E0 | X | E5 | E1 | AB | X | $\bar{A}\bar{B}$ | $\bar{A}\bar{B}$ |

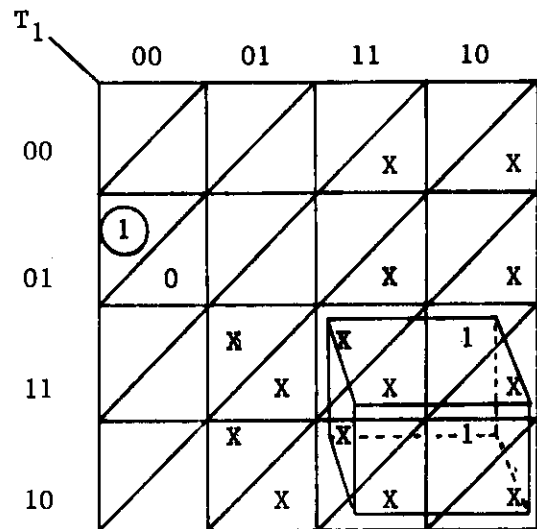
Como se puede apreciar en la tabla anterior no existen estados redundantes, por lo tanto el sistema tiene solamente seis estados.

| ESTADOS ENTRADAS PRESENTES | | ESTADOS PROXIMOS | ENTRADAS DE EXCITACION A LOS FF | SALIDAS | |
|-------------------------------|---------------|---------------------|---------------------------------------|---------|--|
| $S_2 S_1$ | $Q_2 Q_1 Q_0$ | $Q_2 Q_1 Q_0$ | $T_2 T_1 T_0$ | AB | |
| 0 0 | 0 0 0 | 0 0 0 | 0 0 0 | 1 1 | |
| | 0 0 1 | 0 0 0 | 0 0 1 | 1 1 | |
| | 0 1 0 | 0 1 1 | 0 0 1 | 1 1 | |
| | 0 1 1 | 0 1 1 | 0 0 0 | 1 1 | |
| | 1 0 0 | 0 0 0 | 1 0 0 | 1 1 | |
| | 1 0 1 | 0 0 0 | 1 0 1 | 1 1 | |
| | 1 1 0 | X X X | X X X | XX | |
| | 1 1 1 | X X X | X X X | XX | |
| | 0 1 | 0 0 0 | | X X X | |
| 0 0 1 | | | | | |
| 0 1 1 | | | | | |
| 1 0 0 | | | | | |
| 1 0 1 | | | | | |
| 1 1 0 | | | | | |
| 1 1 1 | | | X X X | | |
| 1 0 | 0 0 0 | 0 0 0 | 0 0 0 | 1 0 | |
| | 0 0 1 | 0 1 0 | 0 1 1 | 0 1 | |
| | 0 1 0 | 0 1 0 | 0 0 0 | 0 1 | |
| | 0 1 1 | 0 1 1 | 0 0 0 | 0 1 | |
| | 1 0 0 | 1 0 1 | 0 0 1 | 1 0 | |
| | 1 0 1 | 1 0 1 | 0 0 0 | 1 0 | |
| | 1 1 0 | - | X X X | XX | |
| | 1 1 1 | - | X X X | XX | |
| | 1 1 | 0 0 0 | 0 0 1 | 0 0 1 | |
| 0 0 1 | | 0 0 1 | 0 0 0 | 0 0 | |
| 0 1 0 | | 1 0 0 | 1 1 0 | 0 0 | |
| 0 1 1 | | 1 0 0 | 1 1 1 | 0 0 | |
| 1 0 0 | | 1 0 0 | 0 0 0 | 0 0 | |
| 1 0 1 | | 0 0 1 | 1 0 0 | 0 0 | |
| 1 1 0 | | - | X X X | XX | |
| 1 1 1 | | - | X X X | XX | |

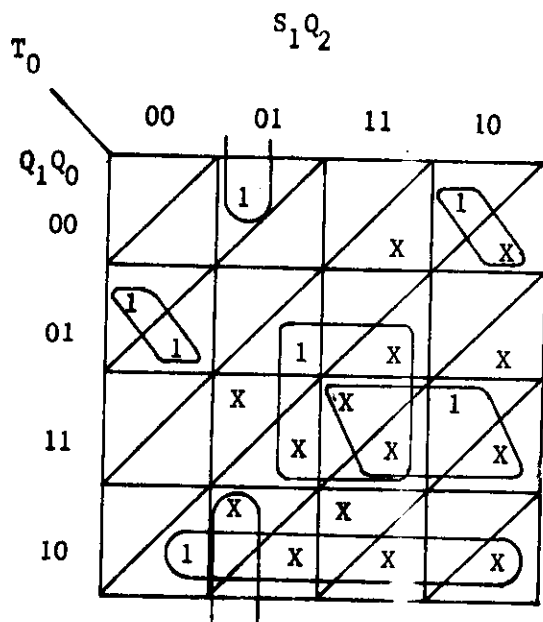
TABLA DE ESTADOS



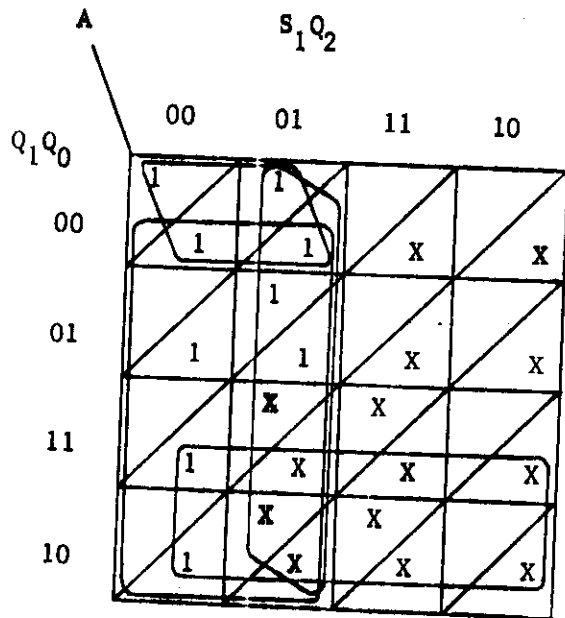
$$s_2 \text{ 1/0 } T_2 = s_1 Q_1 + \bar{s}_2 Q_2 + s_1 Q_2 Q_0$$



$$\text{1/0 } T_1 = s_1 Q_1 + s_2 \bar{s}_1 \bar{Q}_2 \bar{Q}_1 Q_0$$



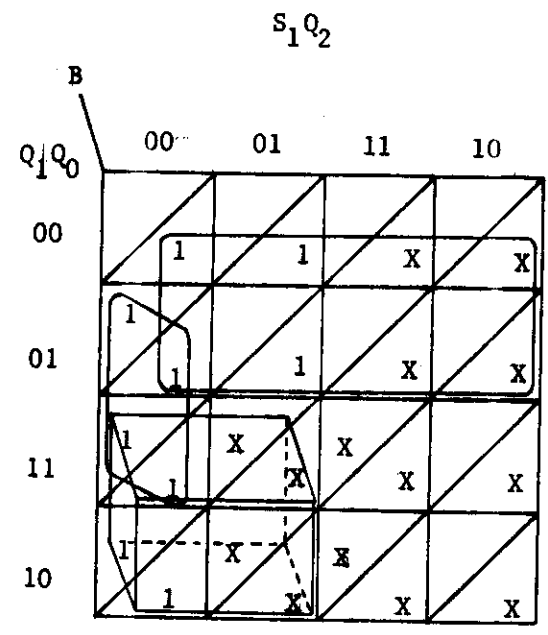
$$\text{1/0 } T_0 = \bar{s}_2 Q_2 Q_0 + \bar{s}_2 Q_1 \bar{Q}_0 + s_1 Q_1 Q_0 + s_1 \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 + \bar{s}_1 \bar{Q}_2 \bar{Q}_1 Q_0 + s_2 \bar{s}_1 Q_2 \bar{Q}_0$$



©

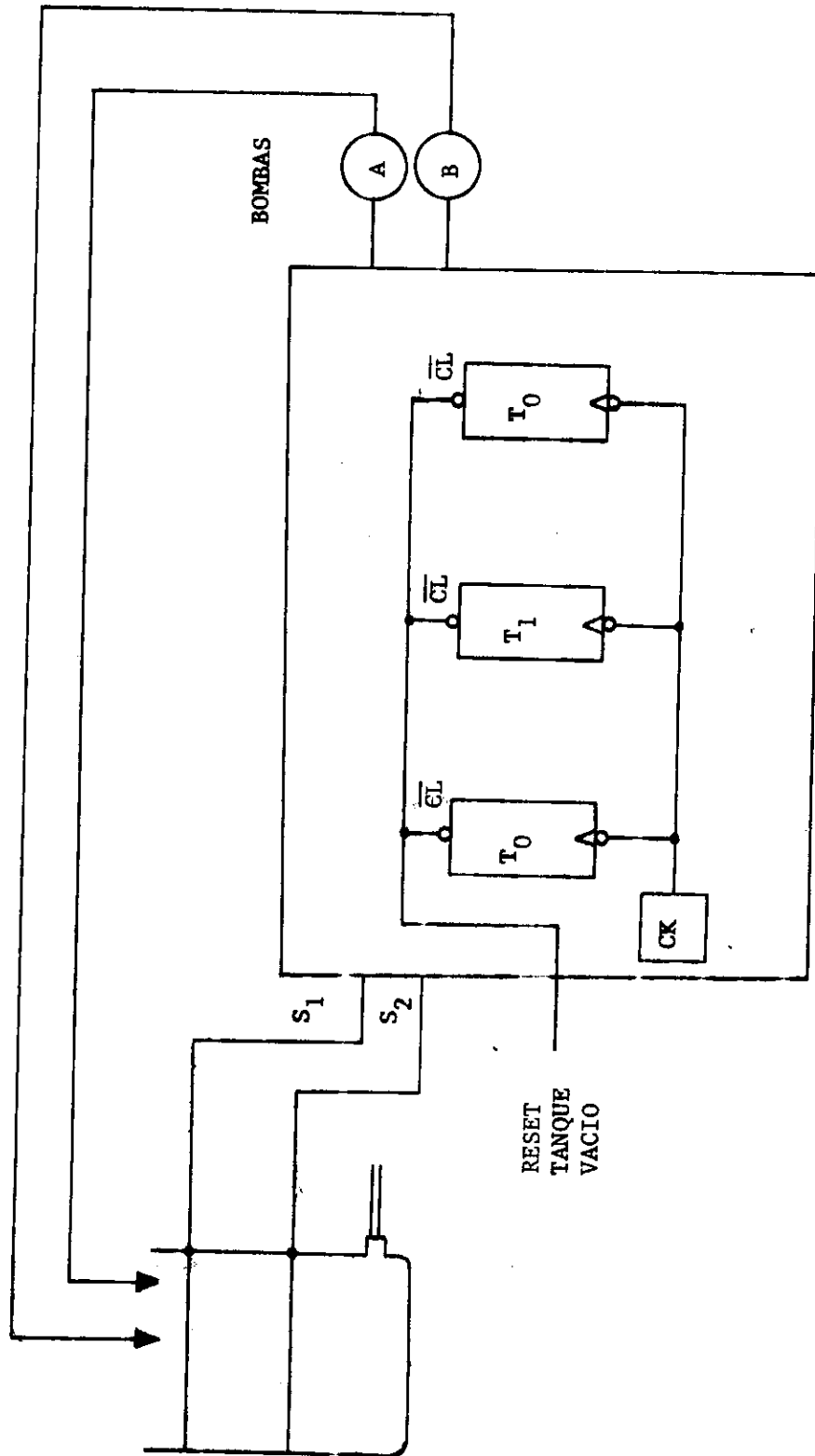
$S_2 1/0$

$$A = \bar{S}_2 Q_1 + \bar{S}_2 \bar{S}_1 + \bar{S}_1 \bar{Q}_1 \bar{Q}_0 + \bar{S}_1 Q_2$$

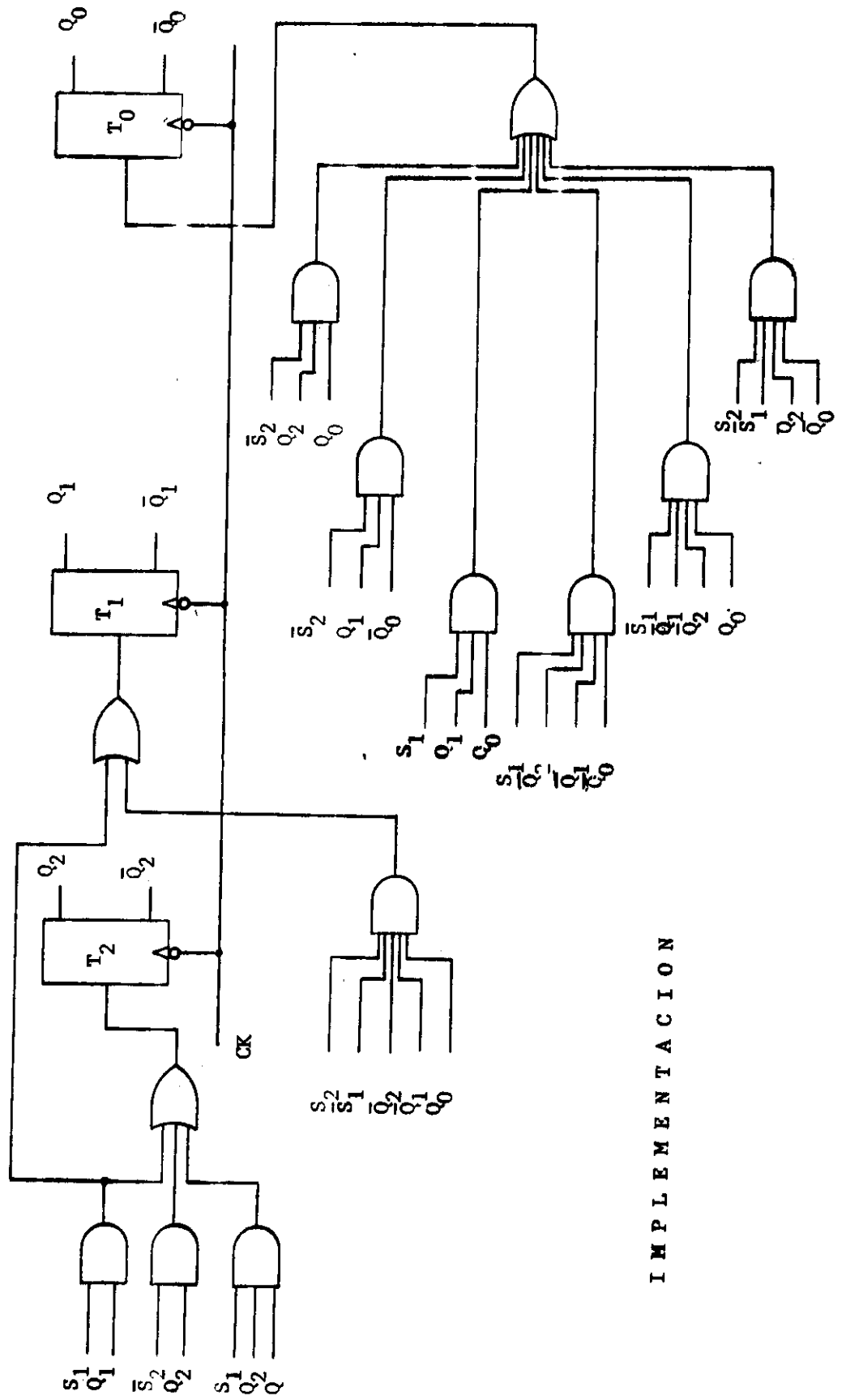
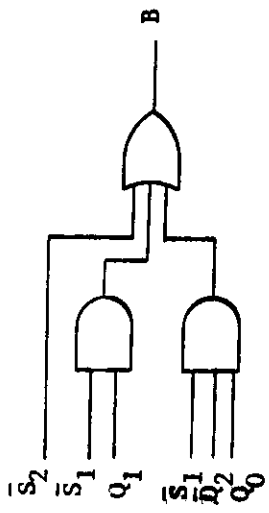
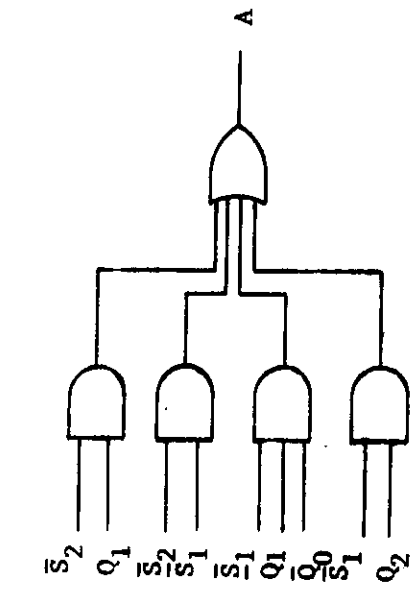


$S_2 1/0$

$$B = \bar{S}_2 + \bar{S}_1 Q_1 + \bar{S}_1 \bar{Q}_2 Q_0 + \bar{S}_2 \bar{Q}_1$$



CONEXIONES EXTERNAS



IMPLEMENTACION

8.12 CONTADORES A PARTIR DE REGISTROS DE CORRIMIENTO

Los registros de corrimiento son una alternativa para el diseño de contadores y generadores de secuencia. El sistema de control para generar una secuencia determinada suele complicarse para este tipo de contadores.

En la fig. 8.12 aparece un contador implementado por un registro de corrimiento de 3 bits. Para generar la secuencia es necesario que el registro cargue un UNO y dos CEROS o un CERO y dos UNOS. Si el registro tiene almacenados solo CEROS o UNOS no genera ninguna secuencia.

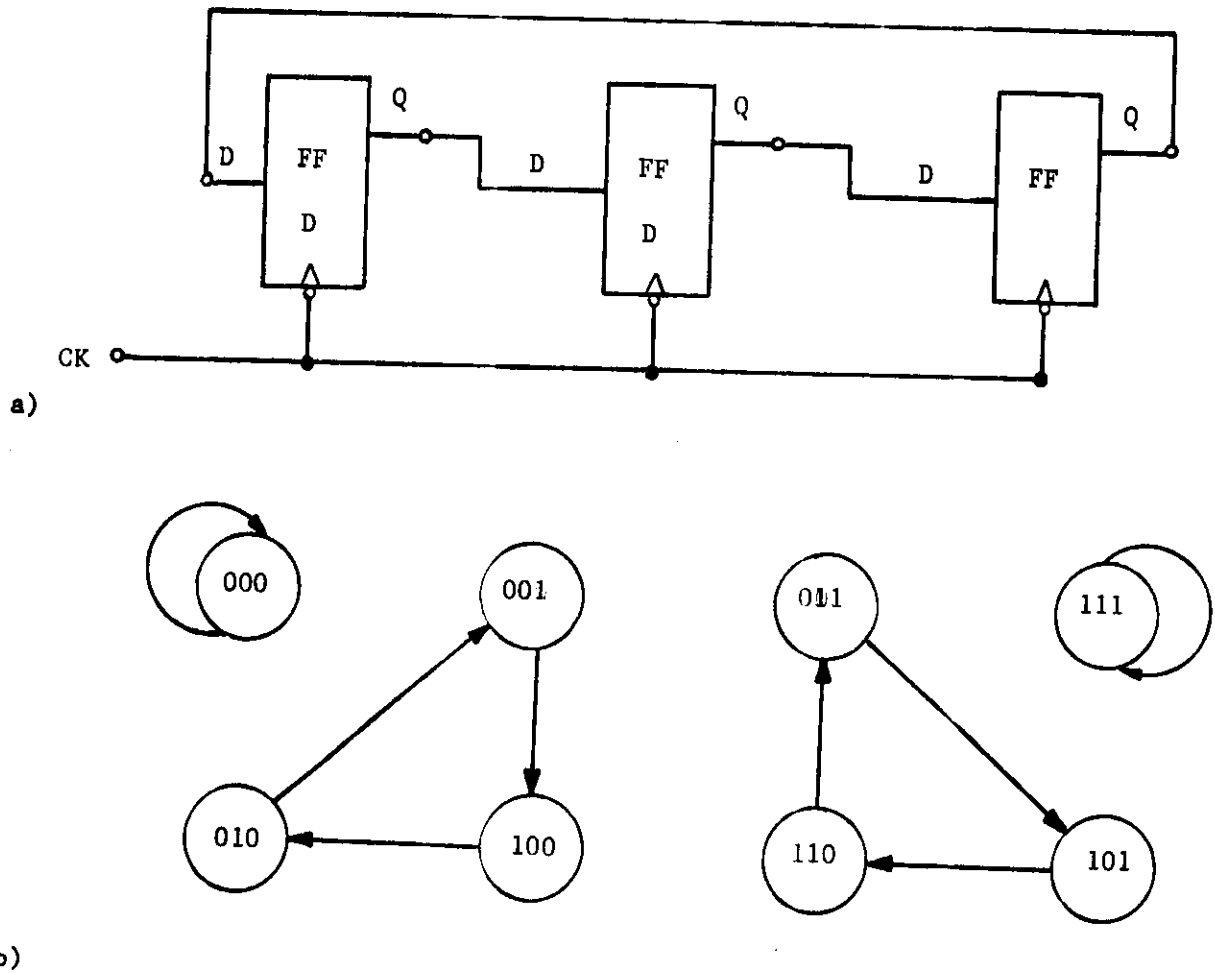


Fig. 8.12 CONTADOR IMPLEMENTADO A PARTIR DE UN REGISTRO DE CORRIMIENTO a) DIAGRAMA, b) DIAGRAMAS DE TRANSICION.

Cuando se retroalimenta la salida complementada del último FF al primero, se genera una secuencia en la que no hay necesidad de cargar datos en el registro. Al circuito se le conoce como contador Johnson y aparece en la fig. 8.13

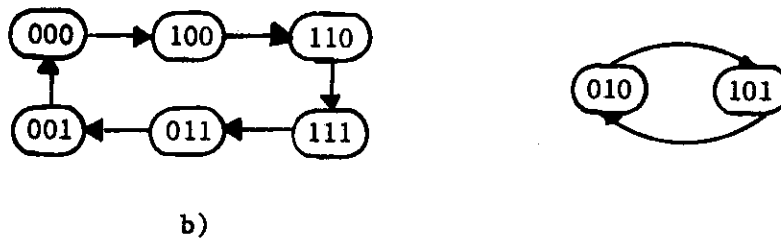
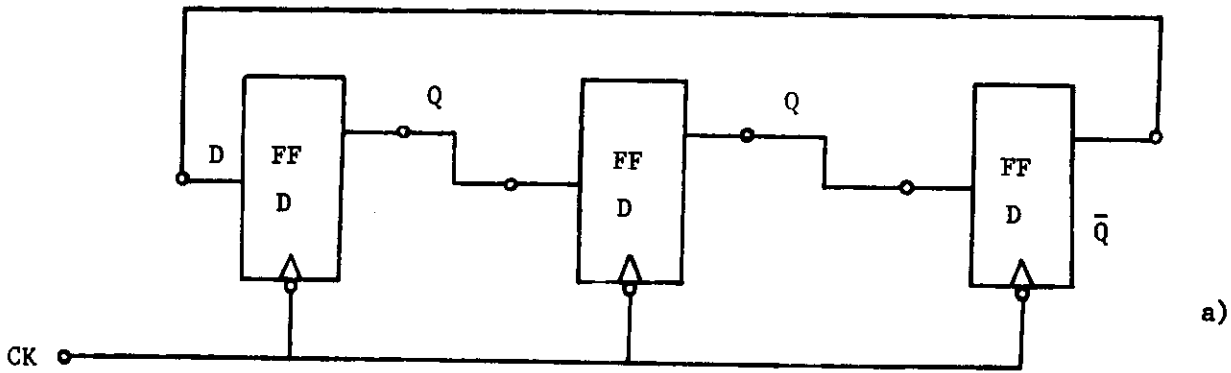


Fig. 8.13 CONTADOR JOHNSON. a) DIAGRAMA, b) DIAGRAMA DE TRANSICION.

8.13 CONTADORES ASINCRONOS

En la fig. 8.14 aparece una configuración adicional para un contador binario. El circuito está implementado con FLIP-FLOPS T. La señal de reloj se conecta a la entrada del CK del primer FF, a su vez la salida del primer FF se conecta a la entrada de reloj del segundo FF y así sucesivamente. Todos los FF-T tienen como entrada un 1 log. Como la señal de CK no es común a todos los FF el sistema es ASINCRONO.

Una desventaja que presenta el tipo de conexión de reloj de estos contadores, es que para cada FF hay un tiempo de retardo t_p , como la salida de un FF va conectada a la entrada de Ck de otro FF los t_p se acumulan, manifestándose sobre todo en contadores de muchos bits.

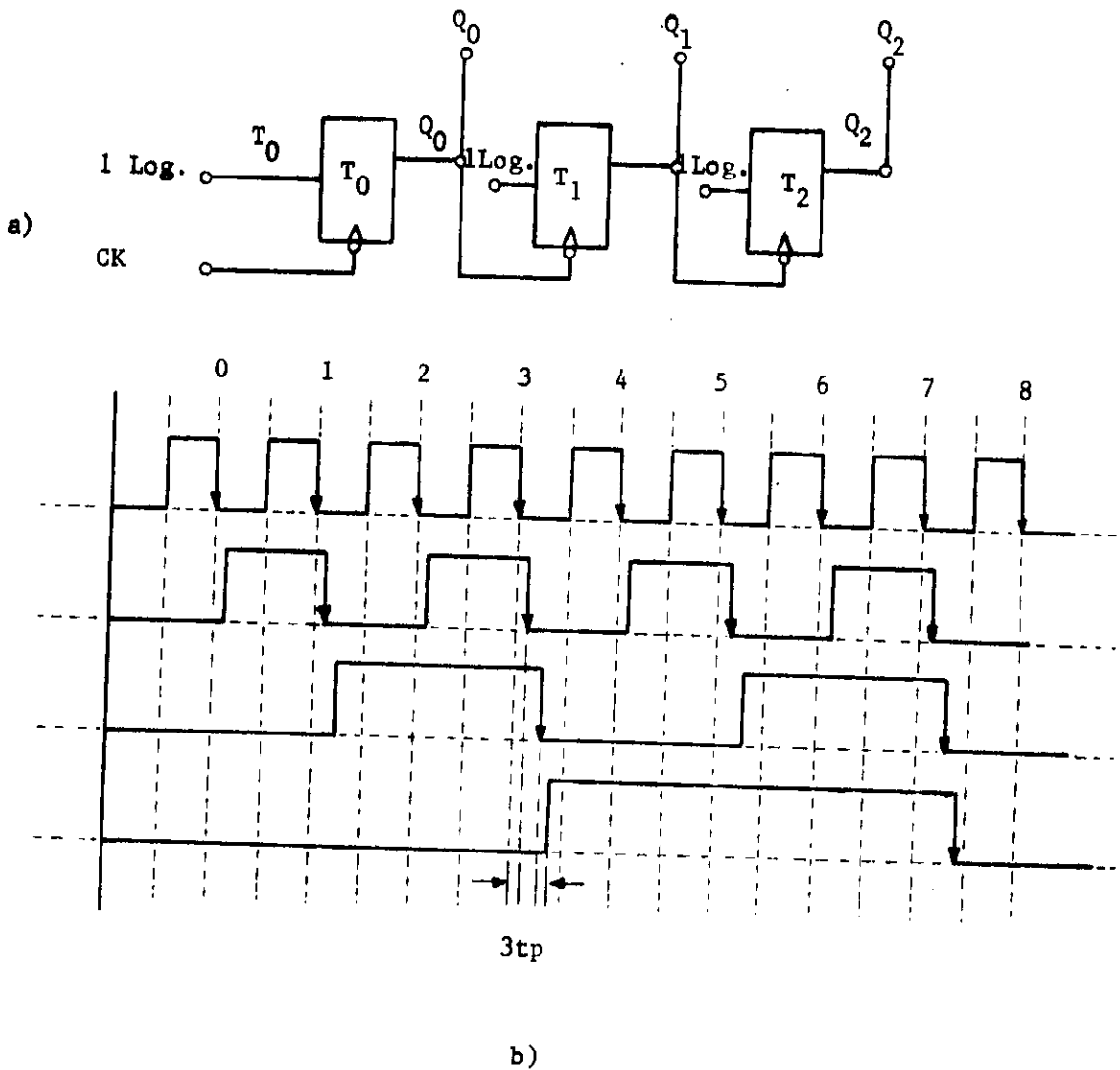


Fig. 8.14 a) CONTADOR ASINCRONO, b) DIAGRAMA DE TIEMPO.

PROBLEMAS PROPUESTOS

- 1.- Diseñar un contador de BCD, ascendente-descendente
- 2.- Diseñar un contador binario de 3 Bits, ascendente-descendente.
- 3.- Diseñar el mismo contador (Del problema anterior) de 4 Bits.
- 4.- Diseñar el mismo sistema del ejemplo 8.11 considerando la siguiente secuencia 01-00-11-10
- 5.- Diseñar un contador en código Gray, ascendente-descendente.
- 6.- Diseñar un contador en código exceso 3, ascendente-descendente.

Apendice A

Semiconductores

A.0 INTRODUCCION

Casi todos los sistemas digitales modernos utilizan circuitos integrados, que dan como resultado un aumento de confiabilidad, reducción de precio y tamaño. La tecnología digital de los circuitos integrados ha avanzado rápidamente desde integración a pequeña escala (SSI), con menos de 13 compuertas lógicas por paquete (chip) a través de mediana escala de integración (MSI), que tiene entre 13 y 99 compuertas lógicas por paquete a la gran escala de integración (LSI), la cual cubre el rango mayor a 100 compuertas lógicas por paquete. Debido al amplio uso que se le ha dado a los circuitos integrados, surge la necesidad de familiarizarse con las características de las distintas familias que existen en el mercado (TTL, ECL, MOS, C MOS), sus ventajas y desventajas. Aún cuando hay muchos fabricantes de circuitos integrados digitales, la mayoría de la nomenclatura y terminología está bastante estandarizada.

A.1 ELEMENTOS SEMICONDUCTORES

DIODOS SEMICONDUCTORES

Un diodo es un elemento semiconductor que permite el paso de la corriente en un solo sentido. A diferencia de otros elementos semiconductores, el diodo no tiene capacidad de amplificar señales de voltaje, sin embargo tiene propiedades que lo distinguen de otros elementos pasivos como son resistencias, capacitores e inductancias, cuyas características son si

métricas, es decir, los efectos que causan sobre una señal son los mismos sin importar la polaridad de la señal ni tampoco la forma en que estén conectados. La característica principal de un diodo es que es asimétrico, - presenta baja resistencia a señales de una polaridad y alta resistencia a señales de polaridad opuesta.

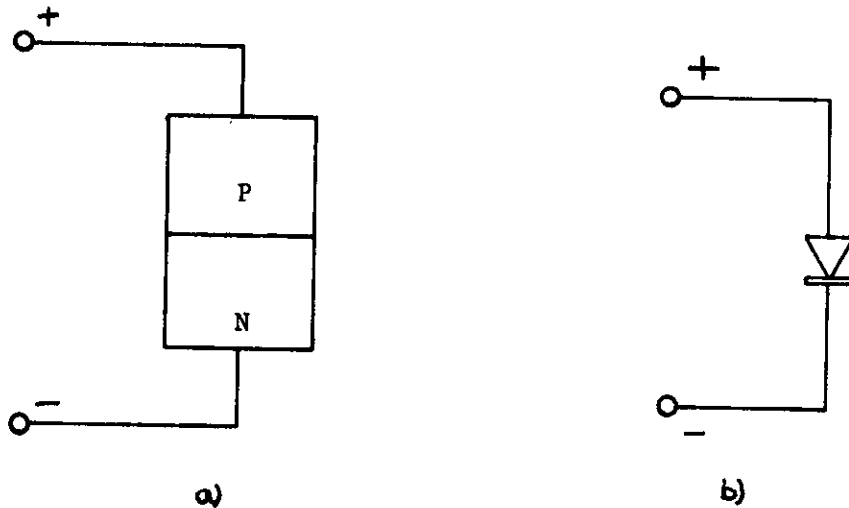


Fig. A.0 a) REGIONES P Y N DE UN DIODO BIPOLAR
b) SIMBOLO

DIODOS SCHOTTKY

Estos dispositivos, también son conocidos como diodos HOT-CARRIER o SCHOTTKY BARRIER, consiste de un contacto de metal se miconductor en el cual la corriente fluye principalmente por portadores mayoritarios. En un material tipo-p el portador mayoritario es un hueco, en un material tipo-n es el electrón. Muchos diodos SCHOTTKY están hechos de silicio de tipo-n en contacto con oro o aluminio. Cuando el metal es polarizado positivamente con respecto al semiconductor, una corriente fluye a través de la unión debido al transporte de electrones del material tipo-n. Esta corriente es portadora mayoritaria y así se diferencia de una unión p-n en la cuál la corriente se debe a portadores minoritarios, es decir los huecos se dirigen hacia la región N y los electrones hacia la región P bajo la influencia de un campo eléctrico. Ahora, cuando los campos cruzan una unión ordinaria se polarizan inversamente, los portadores minoritarios se almacenan por algún tiempo, y la conducción de corriente en la dirección del campo continúa hasta que los portadores minoritarios se hayan

desechado o recombinado. En contraste, la conducción en portadores minoritarios no existe en un diodo Schottky: los electrones que hayan cruzado la unión para entrar al metal no se distinguen de los electrones libres del mismo. Cuando el diodo es polarizado inversamente no existe el fenómeno de almacenar portadores minoritarios y la conducción cesa en un tiempo muy corto después de que se aplica la polarización inversa. Este tiempo llamado REVERSE RECOVERY de un diodo SCHOTTKY puede ser tan corto como 10 ps y el dispositivo se usa ampliamente en circuitos de switcheo de alta velocidad.

Cuando la polarización es directa, los electrones inyectados hacia el metal poseen una mayor energía que los electrones libres del metal y son llamados como HOT. De aquí el nombre de HOT CARRIER.

En la fig. A.1 aparece el símbolo del diodo Schottky y las curvas de operación para un diodo de Si , Ge y Schottky.

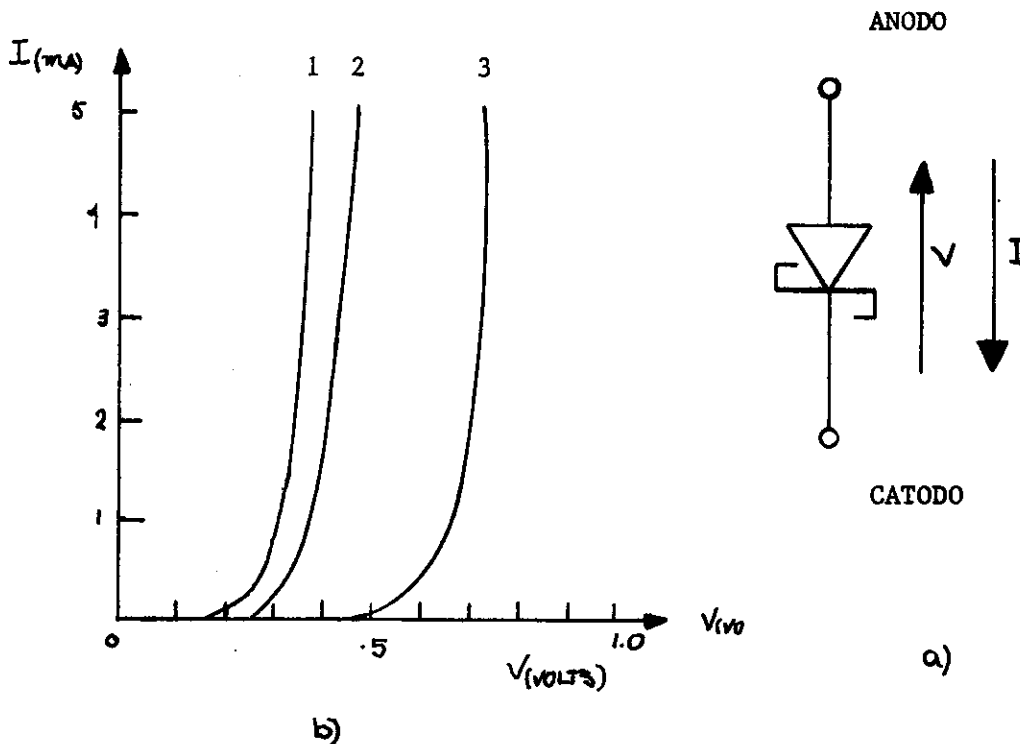


Fig. A.1 a) SIMBOLO PARA UN DIODO SCHOTTKY, b) CURVAS PARA UN DIODO
1.- Ge 2.- SCHOTTKY, 3.- Si

TRANSISTOR BIPOLAR

Un transistor bipolar, o de UNION, consiste de tres regiones semiconductoras, separadas por dos (uniones). Una estructura ideal de un transistor NPN, consiste de dos regiones N y una región P, como se muestra en la fig. A.2a). La fig. A.2b) muestra la sección transversal del mismo transistor y su símbolo en la fig. A.2c).

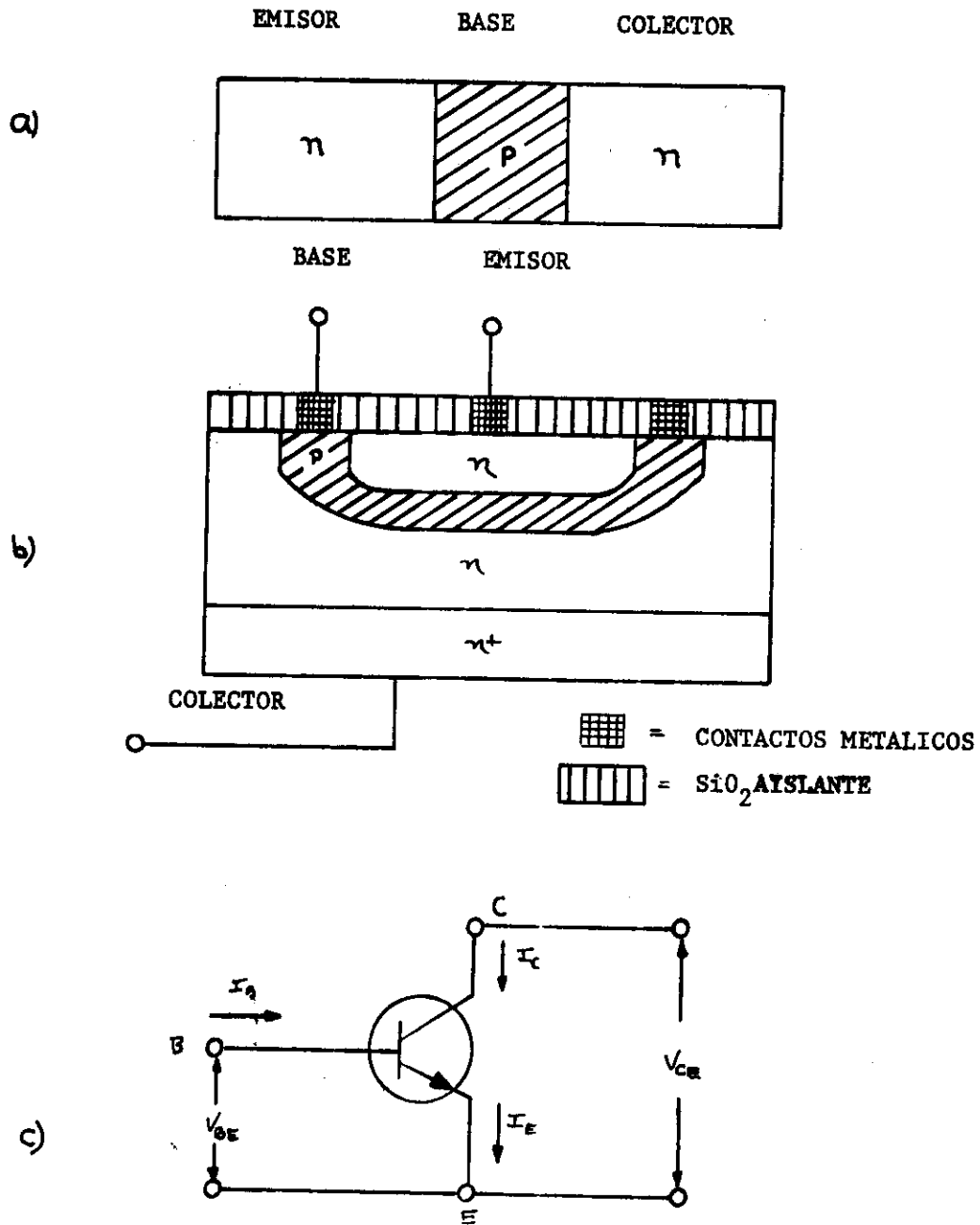


Fig. A.2a) UNION NPN. b) SECCION TRANSVERSAL DE UN TRANSISTOR NPN. c) SIMBOLO

Una u otra o ambas UNIONES pueden OPERARSE en polarización directa - o inversa, dando como resultado cuatro regiones de operación como muestra la fig. A.3. Las características típicas de transferencia de un transistor NPN se muestran en la fig. A.4.

| | | |
|------------------------------------|----------------------------------|----------------------------------|
| | POLARIZACION DIRECTA BASE-EMISOR | POLARIZACION INVERSA BASE-EMISOR |
| POLARIZACION DIRECTA BASE-COLECTOR | REGION DE SATURACION | REGION INVERSA ACTIVA |
| POLARIZACION INVERSA BASE-COLECTOR | REGION DIRECTA ACTIVA | REGION DE CORTE |

Fig. A.3 REGIONES DE OPERACION DE UN TRANSISTOR BIPOLAR

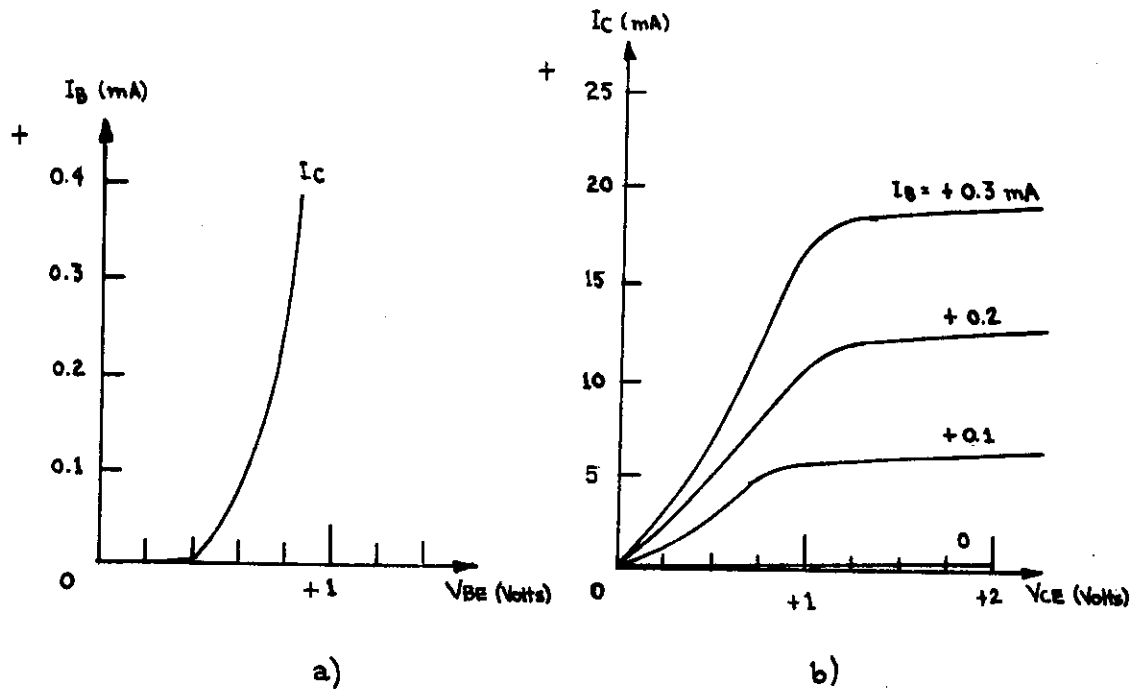


Fig. A.4 CARACTERISTICAS DE TRANSFERENCIA DE UN TRANSISTOR BIPOLAR NPN
a) CURVA DE ENTRADA, b) CURVA DE SALIDA.

En la región DIRECTA ACTIVA de operación, la unión de emisor está polarizada directamente ($V_{BE} > 0$, fig. A.2.C) y la unión de colector se polariza inversamente, $V_{CE} > V_{BE}$. Una corriente de emisor I_E , fluye -- bajo estas condiciones y consiste principalmente de electrones inyectados del emisor. La base se construye muy estrecha físicamente por lo cual la mayoría de las cargas la atraviesan y se difunden en la región del colector, por lo tanto solo una pequeña fracción se combina con los huecos en la región de base.

El rango de la corriente de colector y emisor se llama α y varía típicamente de 0.9 a 0.999. Hemos establecido previamente que la I_E , -- constituye la suma de las corrientes de colector y base.

$$I_C = \alpha I_E = \alpha (I_C + I_B) \quad A - 0$$

Cuando una corriente se aplica a la base de un transistor, la corriente de colector resultante es:

$$I_C = \frac{\alpha}{1 - \alpha} I_B = \beta I_B \quad A - 1$$

La ecuación (A-1) define la ganancia de corriente β , la cual típicamente varía en un transistor bipolar de 100 a 1000. En la discusión anterior no hemos tomado en cuenta una corriente que atraviesa la unión de colector, I_{CBO} , esto se debe a la generación térmica de cargas minoritarias. Esta corriente de corte es muy pequeña en transistores de silicio a temperatura ambiente. La región activa directa de operación se usa en circuitos lógicos ACOPLADOS POR EMISOR.

Otra región de operación es la región de corte, que se usa por la lógica acoplada por emisor y por la lógica TTL; este último también opera en la región de saturación y en la región de corte $V_{BE} \approx 0$ y $V_{CE} - V_{BE} > 0$. En la región de saturación ambas uniones están polarizadas directamente. -- Por ejemplo la fig. A.4.a) $I_B = 0.2$ mA para un $V_{BE} = + 0.7$ volts. Si $V_{CE} = + 0.1$ volts, el transistor estará en la región de saturación donde $V_{CE} - V_{BE} = - 0.6$ volts < 0 ; y solamente pocas cargas minoritarias cruzan la base, hacia la región de colector ya que el campo eléctrico que -- está ayudando a este proceso de difusión, es débil. Cuando el transistor

está apagado por un voltaje de base, estas cargas minoritarias mantienen al transistor en conducción por algún tiempo hasta que se recombinan con cargas mayoritarias o son expulsadas de la región de la base. La respuesta de un transistor saturado es por lo tanto relativamente lenta. -- Por esta razón los transistores saturados son usados principalmente en lógica TTL de mediana y baja velocidad. En un circuito TTL-SCHOTTKY, el efecto de saturación se disminuye grandemente por un diodo SCHOTTKY, que se conecta en paralelo con la unión base-colector. La polarización directa de esta unión se limita alrededor de + 0.4 volts, resultando una significativa reducción de almacenamiento de cargas minoritarias e improvisando a partir de aquí interruptores de velocidad.

La región inversa activa A-3 se caracteriza por bajos valores de β y valores muy bajos de corriente de fuga. Es muy raramente usado en circuitos digitales.

La anterior discusión, basada en un transistor NPN, se aplica en igual forma a un transistor PNP si P se reemplaza por N, N se reemplaza por P, y la polaridad de voltaje y corriente se invierten. El símbolo de un transistor PNP se muestra en la fig. A.5

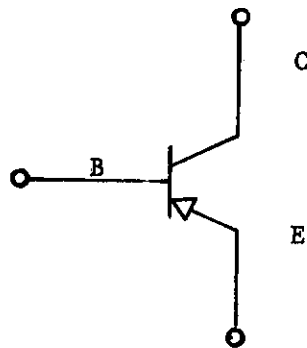


Fig. A.5 SIMBOLO PARA UN TRANSISTOR PNP

MOSFET

FET (Transistor de Efecto de Campo)

MOS (Semiconductor de Oxido Metálico)

El MOSFET consume poca potencia, debido a esto se ha utilizado ampliamente en circuitos de gran escala de integración, (LSI). Una vista de la-

sección transversal de un MOSFET canal p se muestra en la fig. A.6 (Los símbolos se muestran en la fig. A.8). El sustrato básico consiste de silicio tipo-n dentro del cual se ha difundido la fuente tipo-p y la región de drenaje. Se deposita una capa aislante de S_iO_2 entre el drenaje y la fuente, como se ve en la figura. Este aislante que tiene cerca de 1000 Å de espesor, se le denomina como compuerta de óxido. La compuerta consiste de un metal, generalmente aluminio, que se deposita en la parte superior de la compuerta. Se han desarrollado otros métodos, para la fabricación de compuertas, y se han utilizado diferentes compuestos químicos como aisladores de compuerta para lograr dispositivos con distintas características.

La operación del MOSFET se hará con referencia a un dispositivo canal tipo-p. La impedancia de entrada de la compuerta es de 10^9 a 10^{12} ohms en paralelo con una capacitancia < 1 pF. La corriente de fuente a drenaje, I_D , se encuentra en el rango de 10^{-9} a 10^{-12} Amper. con un voltaje de compuerta cero, dado que la fuente y el drenaje están aislados por el sustrato de alta resistividad. Un voltaje negativo aplicado a la compuerta induce una correspondiente carga positiva en el sustrato que está debajo de la compuerta. Cuando el voltaje de la compuerta ha alcanzado suficiente magnitud, el canal entre la fuente y el drenaje vendrá a ser de tipo p invertido, como lo muestra la fig. A.7, y conducirá corriente si se aplica un voltaje en los electrodos de la fuente y el drenaje.

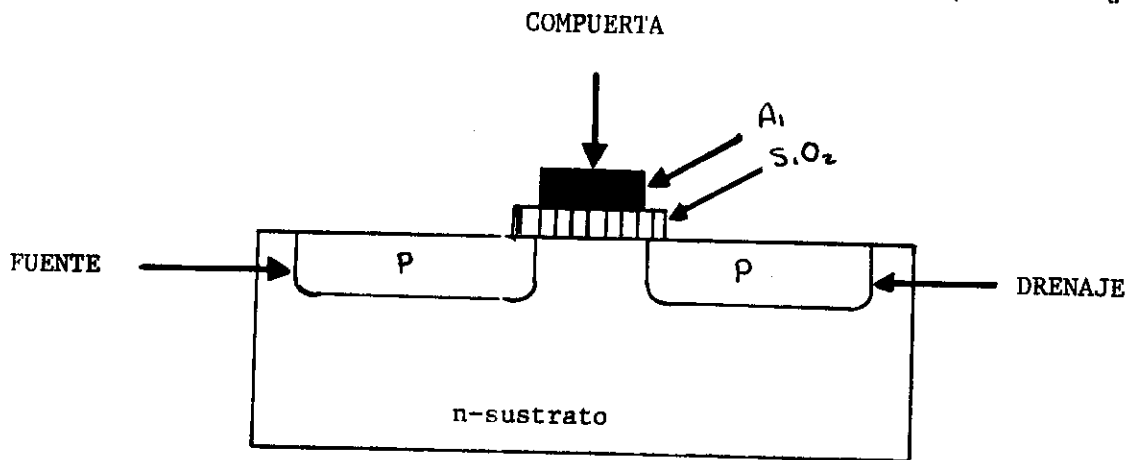


Fig. A.6 SECCION TRANSVERSAL DE UN TRANSISTOR MOSFET

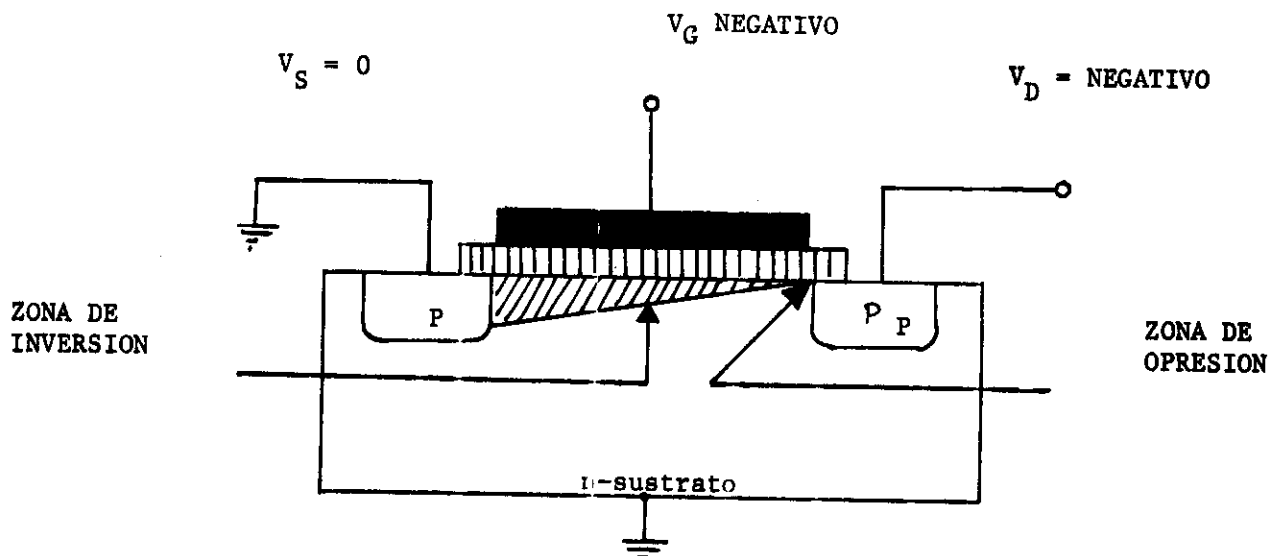


Fig. A.7 FORMACION DEL CANAL ENTRE DRENAJE Y FUENTE.

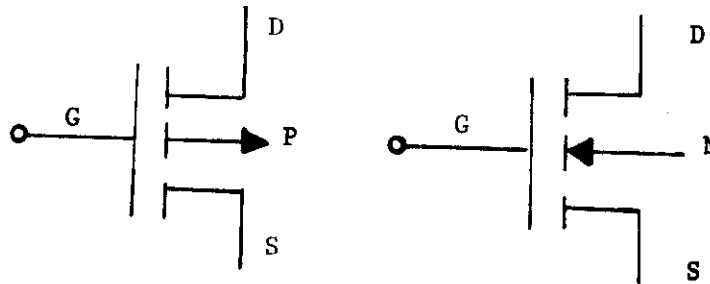


Fig. A.8 SIMBOLOS PARA UN TRANSISTOR MOSFET a) CANAL P
b) CANAL n.

La característica típica de transferencia de un MOSFET, se muestra en la fig. A.9, debe ser dividida en dos regiones de operación : del lado izquierdo de la línea punteada la corriente de drenaje, I_D , asciende rápidamente con el voltaje de fuente a drenaje, V_D , y el potencial de fuente a compuerta V_G .

$$I_D = \beta \left[(V_G - V_t) V_D - \frac{V_D^2}{2} \right] \quad A-2$$

en donde V_t es el voltaje de Threshold y β es un factor incluyendo una dependencia geométrica.

$$\beta \propto W/L$$

A-3

donde W = ancho del canal, L = largo del canal. Esta es la región de "operación de triodo" dada por:

$$\left| V_G - V_T \right| > \left| V_D \right| \quad A-4$$

Hacia la derecha de la línea punteada el MOSFET opera en su región de saturación. Como V_D se incrementa más allá de la región de triodo, el campo eléctrico que atraviesa la compuerta es insuficiente para mantener las cargas móviles en el fin del drenaje del canal. El canal es estrangulado y no es posible un incremento adicional de corriente. La corriente de drenaje en la región de saturación es:

$$I_D = \beta \frac{(V_G - V_T)^2}{2} \quad A-5$$

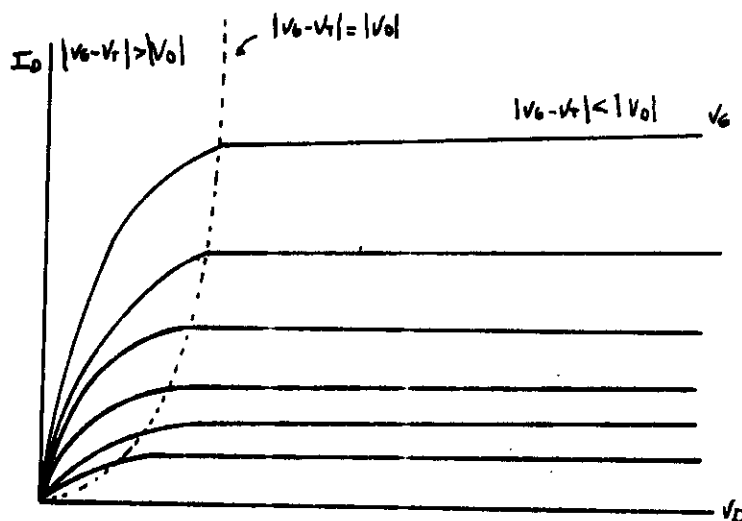


Fig. A.9 CARACTERISTICA TIPICA DE TRANSFERENCIA DE UN MOSFET CANAL-P.

y la transconductancia del dispositivo, g_m , se deriva de la igualdad -- (A-5):

$$g_m = \left. \frac{\partial I_D}{\partial V_G} \right|_{V_D} = \beta (V_G - V_T) \quad A-6$$

De la ec. (A.2) y (A.5) notamos que, en ambas regiones de operación-
 I_D se modula por el voltaje de compuerta a fuente y también depende de la
geometría del dispositivo, a través del coeficiente β .

Lo expuesto anteriormente del MOSFET canal-p puede aplicarse a dispo-
sitivos de canal-N si todas las P's y N's se intercambian y las polarida-
des de voltaje y corriente se invierten. En general el MOSFET canal-N --
tiene mejor característica de alta frecuencia dado que la movilidad de -
los electrones es más alta por un factor de aproximadamente 2.5 que la de
los huecos. Los MOSFET son principalmente transistores de modo de enri-
quecimiento, conducen cuando V_G negativo y no conducen cuando no se apli-
ca voltaje a V_G . En MOSFET tipo-N el drenaje es positivo con respecto --
a la fuente y tanto la fuente como el drenaje son semiconductores tipo-n.
Muchos MOSFET canal-N operan en el modo de empobrecimiento; cuando se a--
plica un voltaje cero a la compuerta, el MOSFET conduce máxima corriente-
 I_{DSS} mientras un voltaje negativo en V_G reduce la corriente, y si es de -
suficiente magnitud, puede cortarla. Canal-p en modo de vacío y el canal
-N en modo de enriquecimiento en los MOSFET se usan con frecuencia.

Apendice B

Familias Logicas

B.0 TERMINOLOGIA DE LOS CIRCUITOS INTEGRADOS

PARAMETROS DE CORRIENTE Y VOLTAJE

- V_{IH} Voltaje de entrada de NIVEL ALTO, es el nivel de voltaje requerido para un 1 lógico en una entrada. Cualquier voltaje bajo de este nivel no será aceptado como un nivel alto por el circuito lógico.
- V_{IL} Voltaje de entrada de NIVEL BAJO, es el nivel de voltaje requerido -- para un 0 lógico en una entrada. Cualquier voltaje por encima de este nivel no será aceptado como nivel bajo.
- V_{OH} Voltaje de salida de NIVEL ALTO. Es el nivel de voltaje en la salida de un circuito lógico en el estado lógico 1. El fabricante especifica generalmente el valor mínimo de V_{OH} .
- V_{OL} Voltaje de salida de NIVEL BAJO: es el nivel de voltaje a la salida de un circuito lógico en el estado lógico 0. El fabricante especifica generalmente el valor máximo de V_{OL} .
- I_{IH} Corriente de entrada de NIVEL ALTO: Es la corriente que fluye en una entrada cuando un nivel de voltaje alto especificado se aplica a esa entrada.
- I_{IL} Corriente de entrada de NIVEL BAJO: Es la corriente que fluye en una entrada cuando un nivel de voltaje bajo especificado se aplica en esa entrada.

- I_{OH} Corriente de salida de NIVEL ALTO: Es la corriente que fluye desde una salida en el estado lógico 1 bajo condiciones de carga especificadas.
- I_{OL} Corriente de salida de NIVEL BAJO: Es la corriente que fluye desde una salida en el estado lógico 0 bajo condiciones de carga especificada.

FAN-OUT

El Fan-Out (llamado también factor de carga) se define como el número máximo de entradas lógicas estándar que una salida puede manejar confiablemente. Por ejemplo, una compuerta lógica que se especifica con un Fan-Out de 10 puede manejar 10 entradas estándar. Si se excede este número los niveles de voltaje de salida no pueden garantizarse.

TIEMPO DE TRANSICION

Algunos circuitos digitales responden a niveles lógicos en sus entradas, pero otros se activan por el cambio rápido en voltajes, estos últimos requieren que las señales de entrada tengan transiciones de nivel rápidas o de lo contrario el circuito no responderá.

Los tiempos de subida " t_R " y los de caída " t_F " deben de especificarse y sus valores no son necesariamente iguales y ambos dependen de la cantidad de carga colocada en una salida lógica.

RETARDOS EN PROPAGACION

Una señal lógica siempre experimenta un retardo al pasar a través de un circuito, los tiempos de retardo en propagación son:

t_{PLH} : tiempo de retardo al ir de 0 lógico al NIVEL 1.

t_{PHL} : tiempo de retardo al ir de 1 lógico al NIVEL 0.

Los valores de t_{PLH} y t_{PHL} se usan como medida de la velocidad relativa de los circuitos lógicos.

INMUNIDAD AL RUIDO:

Los campos eléctricos y magnéticos pueden inducir voltajes en los alambres que conectan circuitos lógicos. Estas señales indeseadas, se denominan ruido y pueden ocasionar que el voltaje a la entrada de un circuito lógico caiga debajo del V_{IH} o encima de V_{IL} , lo cual produciría un error en la operación. La inmunidad al ruido de un circuito lógico se refiere a la

habilidad del circuito para tolerar voltajes de ruido en sus entradas.

REQUERIMIENTOS DE POTENCIA

La cantidad de potencia requerida por un circuito integrado es una característica importante y se especifica siempre en las hojas de datos del fabricante. A veces se da directamente como la disipación promedio de potencia P_D . A menudo se especifica indirectamente en términos del drenaje de corriente desde la fuente de alimentación para el C.I. Esta corriente se simboliza como I_{CC} . Cuando se conoce el valor de I_{CC} la potencia tomada del suministro se obtiene simplemente multiplicando I_{CC} por el voltaje de la fuente. Para algunos CI el valor de la corriente I_{CC} será diferente para los dos estados lógicos. En tales casos los dos valores de I_{CC} se especifican. I_{CCH} (o $I_{CC(1)}$) es la corriente del suministro cuando todas las salidas del paquete son altas; I_{CCL} (o $I_{CC(0)}$) es la corriente del suministro cuando todas las salidas están bajas.

CORRIENTE DE PICO

La corriente de pico se manifiesta como pequeños pulsos en la línea de la fuente, haciéndose más notorios en el cambio de "0" a "1", esto se debe a que en la región de transición los transistores Q_3 y Q_4 se encuentran en ON por un pequeño período de tiempo (ver punto B.3), lo cual causa que la fuente se aterrice a través de una pequeña resistencia. Para prevenir que los picos aparezcan como ruido, en el sistema, se debe colocar un capacitor (0.01 a 0.1 mfd) de desacoplo entre V_{CC} y tierra (GND).

CORRIENTE DE CORTO CIRCUITO

La corriente de corto circuito I_{OS} es la corriente que puede fluir hacia la salida de un dispositivo cuando ésta se conecta a tierra, puede ser interpretada como la capacidad del circuito para cargar una capacitancia externa rápidamente en el estado alto "1 log". Sin embargo I_{OS} no es indicativo de la capacidad de carga de corriente directa.

B.1 FAMILIA TTL

B.1.0 SERIES DE LA FAMILIA TTL

SERIE ESTANDARD 54/74 DE CIRCUITOS TTL

La serie estandard 54/74 de circuitos integrados ofrece la mejor combinación de velocidad y disipación de energía para la mayoría de las aplicaciones. En la figura B.0 se muestra el circuito de un bloque básico -- con su entrada a través del emisor múltiple y su salida a través del transistor activo tipo (pull-up). El transistor Q_1 con su entrada de emisor múltiple permite una gran lógica por su mínimo tamaño físico y es la base para la alta velocidad en los circuitos TTL lográndose así mismo una baja impedancia de salida por el uso del transistor activo Q_3 tipo (pull-up), el cual también contribuye a mejorar la inmunidad al ruido y la rapidez de cambio de estado. Además de los dispositivos de integración a baja escala (SSI), la Serie 54/74 ofrece registros de corrimiento, contadores, decodificadores, memorias, selectores de datos y elementos aritméticos. Estos circuitos incluyen también una gran variedad de flip-flops como: Sencillos, duales, con disparo al empezar el pulso, maestro-esclavo tipo D ó tipo J-K además, se incluyen bloques con salidas de colector abierto.

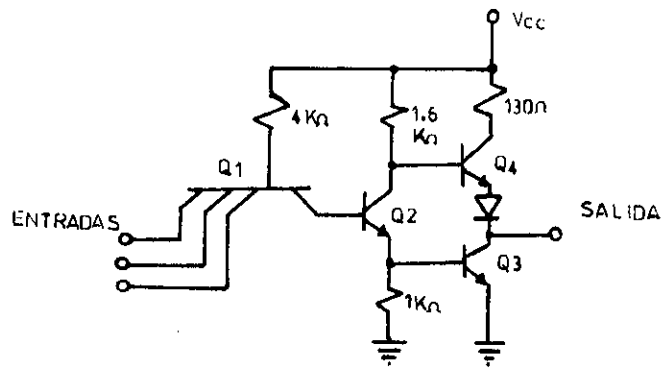


Fig. B.0 BLOQUE TTL SN 54/74

SERIE TTL DE BAJA POTENCIA 54/74

Los circuitos de baja disipación de energía asignados como Serie 54L/74L, son el producto con las mejores características de velocidad/potencia de todas las familias lógicas. En la figura B.1 se muestra un circuito básico de un bloque de baja disipación donde se observa que la configuración es la misma que los de la serie Estándar, con la diferencia que los valores de resistencia son mayores y por lo tanto, se tiene una reducción en el consumo de energía, lográndose disminuir hasta un décimo de la potencia requerida por los circuitos de la Serie Estándar. La Serie 54L/74L tiene una disipación de energía de solo 1 mW por bloques y velocidades de aproximadamente el doble de los circuitos que tienen la misma disipación de energía alcanzándose velocidades típicas de 33 ns por bloque.

La serie 54L/74L es ideal para aplicaciones donde el consumo de energía y la disipación de calor son parámetros críticos.

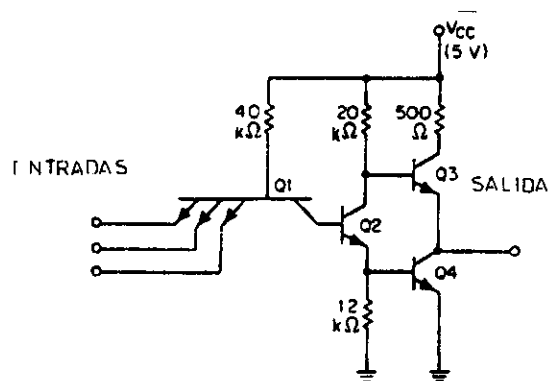


Fig. B.1 BLOQUE TTL DE BAJA POTENCIA SN 54L/74L.

SERIE TTL DE ALTA VELOCIDAD 54H/74H.

La configuración del circuito de un bloque de alta velocidad (fig. B.2) es básicamente la misma que la de un circuito estándar Serie 54/74 con la diferencia que las resistencias tienen valores más bajos y en cada uno de los emisores existe un diodo conectado a tierra el cual podría denominarse diodo sujetador (clamping diode) y cuyo propósito es el de reducir los efectos de la línea de transmisión que llegan a ser más significa

tivos al tenerse tiempos menores en las salidas y en la bajada de los pulsos.

La sección de salida consiste en un par de transistores tipo Darlington Q_3 y Q_4 que tienden a mejorar las velocidades de los bloques (6 ns -- por bloque) debido a la baja impedancia de estado estable, la cual es --- 10 ohms en el estado no saturado y 100 ohms en el estado saturado. Sin embargo, los circuitos TTL Serie 54H/74H tienen desventaja de consumir más energía que los de la Serie Estandar 54/74.

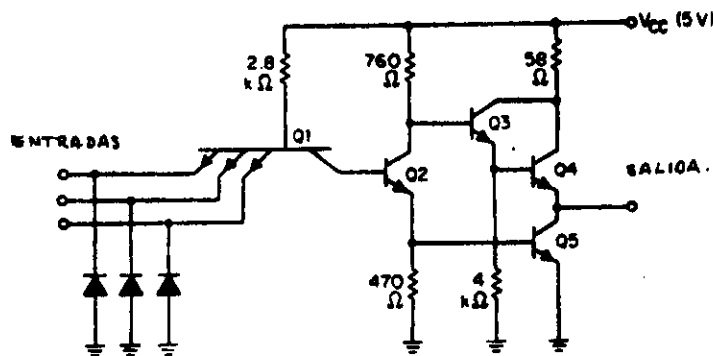


Fig. B.2 CIRCUITO TTL DE ALTA VELOCIDAD SN 54H/74H.

SERIE TTL TIPO DIODO SCHOTTKY 54S/74S.

Esta serie 54S/74S tiene la más alta velocidad de todos los elementos de la Serie 54/74 lográndose esto mediante una combinación de lógica no saturada (ECL) con el consumo de energía relativamente bajo de los circuitos TTL.

La operación de esta Serie se realiza por el uso de un diodo de barrera Schottky SBD (Schottky-Barrier-Diodo) a manera de una sujeción entre la base y el colector tal como se muestra en la figura B.3. Las propiedades del SBD que lo hacen útil son que no tienen cargas minoritarias y por lo tanto no almacena carga, además de que tiene una caída de voltaje menor que la junta p-n cuando se polarizan directamente. Cuando el SBD se-

usa como una sujeción (grapa), desvía la mayor parte del exceso de corriente de base, evitando que el transistor alcance la típica saturación.

La salida en la Serie 54S/74S ha sido modificada para dar unas características de transferencia simétricas (ver fig. B.3). Q_3 y las resistencias asociadas reemplazan la resistencia conectada a tierra que se usa en los otros tipos de la Serie 54/74.

Las características sobresalientes de la Serie 54S/74S son: 3 ns como tiempo promedio de propagación y una disipación de energía promedio de 20 mW.

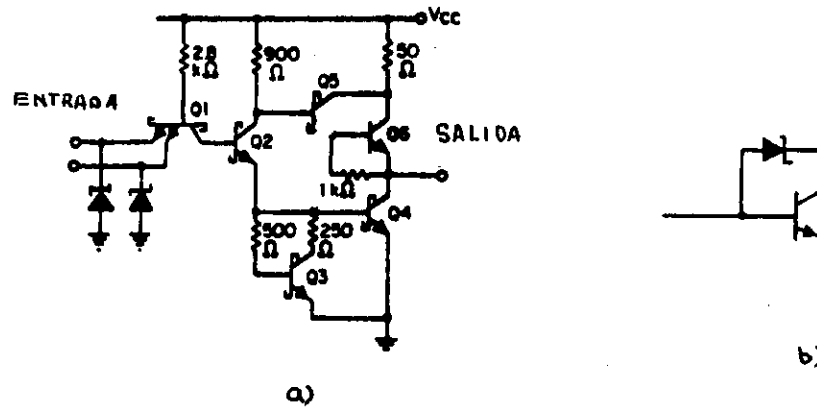


Fig. B.3 CIRCUITO TTL CON DIODO SCHOTTKY SN 54S/74S.

B.1.1 OPERACION BASICA

TRANSISTORES DE EMISOR MULTIPLE

Muchas compuertas TTL con más de una entrada usan transistores con más de un emisor. El símbolo para un transistor multiemisor se muestra en la fig. B.4

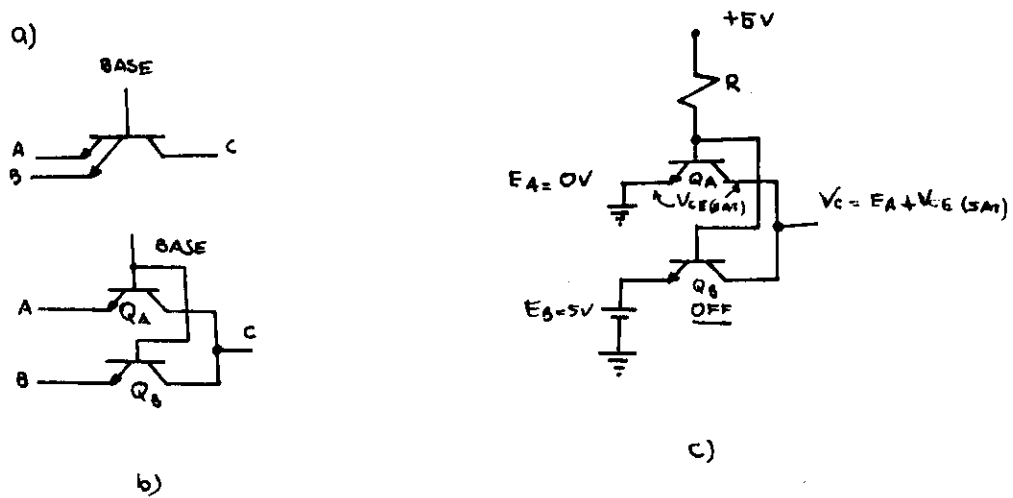


Fig. B.4 a). Símbolo de un transistor multi-emisor
 b). Circuito equivalente de un transistor multi-emisor
 c). Transistor de doble emisor con $E_A = 0$ Volts y $E_B = 5$ Volts

Un transistor con dos emisores funciona similarmente a dos transistores con sus bases conectadas como se muestra en la fig. B.4b) y opera de la siguiente manera: Suponga que la entrada $A = 0$ lógico, 0V y la entrada $B = 1$ Lógico, + 5V y la base = 5V. Entonces el V_{BE} del Q_B es inverso. El V_{BE} de Q_A es directo, por lo tanto Q_A está en ON o estado de conducción o sea, que el voltaje de colector es:

$$V_C = E_A + V_{CE(SAT)Q_A} \approx 0V + 0.1V$$

$$V_C = 0 \text{ Log}$$

Recordaremos que se está haciendo uso del sentido de corriente convencional (corriente que fluye de positivo a negativo).

COMPUERTA NAND, STANDARD TTL SERIE 54/74

El diagrama para un NAND de 2 entradas, se muestra en la figura B.5 - consiste en un transistor de entrada Q_1 de dos emisores, los transistores Q_2 , Q_3 , Q_4 , diodos D_1 , D_2 , D_3 y resistencias R_1 , R_2 , R_3 , R_4 .

El circuito puede dividirse en 3 secciones de entrada, driver y salida.

Para la entrada normal, (positiva), los diodos D_1 y D_2 están inversamente polarizados y pueden considerarse como circuitos abiertos, cualquier voltaje negativo más grande que el voltaje de disparo de los diodos será drenado a tierra a través de ellos mismos. Con esto se reducen los efectos de la línea de transmisión en alta velocidad y se protege al transistor de entrada contra voltajes negativos.

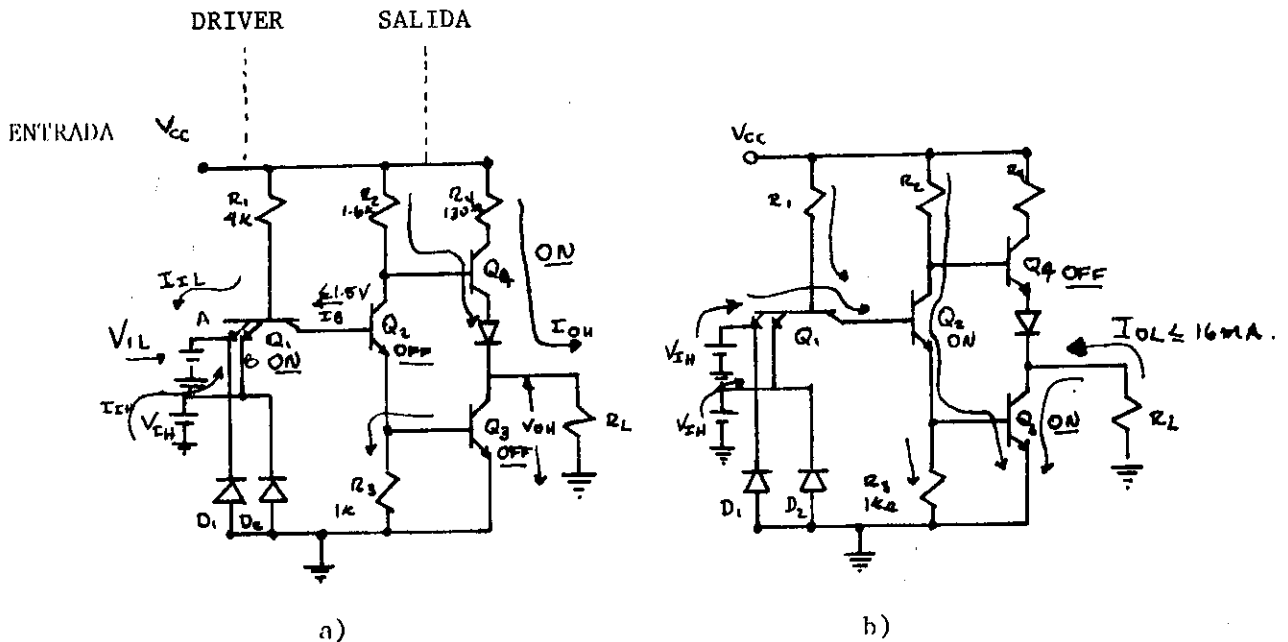


Fig. B.5 COMPUERTA NAND TTL

ENTRADAS A NIVEL BAJO, "0 LOG."

Si una o más entradas al NAND están en 0 log. la salida será un 1 log. supongamos que la entrada A está en un 0 log. y B es un 1 log. ver fig. -- B.5a).

La unión BE del transistor Q_1 está directamente polarizada debido a -- que la base de Q_1 es más positiva que el emisor A, el valor de R_1 es tal --

fluya suficiente corriente de base I_B , para saturar el transistor Q_1 y es aproximadamente igual a:

$$I_{IL} = \frac{V_{CC} - (V_{BE1} + V_{IL(\text{tip.})})}{R_1} = \frac{5V - 0.7V - 0.4V}{4K} = -0.97 \text{ mA.}$$

La unión base-emisor para la entrada "B" está inversamente polarizada ya que B está en un "1 log." con un voltaje mínimo de 2V y el V_{B1} es aproximadamente:

$$V_{B1} = V_{IL(\text{max})} + V_{BE1} \quad 0.8V + 0.7V = 1.5V$$

Según las especificaciones del fabricante la corriente de fuga $I_{IH} = -40 \mu A$ para un $V_{IH} = 2.4V$ y -1 mA para un $V_{IH} = 5.5V$.

Si Q_2 estuviera en ON (conducción) entonces Q_3 podría también conducir y el voltaje de base de Q_2 podría ser $V_{BE3} + V_{BE2} = 0.7V + 0.7V = 1.4 \text{ volts}$. El transistor Q_1 está saturado, y su voltaje de colector debe ser igual al V_{IL} Voltajes de entrada para "0 log" más el voltaje de saturación de Q_1 .

$$V_{C1} = V_{B2} = V_{IL(\text{max})} + V_{CE(\text{sat})} \quad 0.8V + 0.1V = 0.9V$$

De aquí que V_{B2} tendría que estar en 1.4V para conducir, entonces, Q_2 está en OFF.

Como Q_2 está en OFF la corriente que fluye a la base de Q_3 es nula -- y Q_3 por lo tanto no conduce. La resistencia R_2 tiene un valor tal que -- cuando Q_2 está en OFF fluya suficiente corriente a la base de Q_4 y lo haga conducir. Con Q_4 en ON, la corriente I_{OH} fluye a través de R_4 , Q_4 , D_3 y R_1 y la salida es un 1 lógico.

ENTRADAS A NIVEL ALTO "1 LOGICO"

Supongamos que ambas entradas al NAND están en un "1 Lógico", entonces ambos emisores de Q_1 están inversamente polarizadas y únicamente la corriente de fuga fluye dentro de cada emisor. La unión colector-base de Q_1 está inversamente polarizada, y la corriente que fluya a través de R_1 y el colector de Q_1 va hacia la base de Q_2 y lo cambia a ON.

$$I_{R_1} = \frac{V_{CC} - V_{BC1} - V_{BE2} - V_{BE3}}{R_1} = \frac{5V - 0.7 - 0.7 - 0.7}{4K} =$$

$$\frac{5V - 2.1V}{4K} = 0.72 \text{ mA}$$

Con Q_2 en ON, fluye corriente del V_{CC} a través de R_2 , Q_2 , R_3 y entra a Q_3 haciéndolo conducir. Con Q_3 en ON, la salida está a un nivel bajo o "0 lógico".

$$V_{ol}(\text{tip}) = V_{CE}(\text{sat}) = 0.2V$$

La máxima corriente que asegura que el voltaje de salida no sea mayor que el $V_{ol} = 0.4, V$, es $I_{OL(\text{max})} = 16 \text{ mA}$. Una corriente mayor que ésta puede llevar a Q_3 a saturación incrementando el voltaje de salida de nivel bajo V_{ol} .

Desde que Q_3 está en ON, su base está a $V_D = 0.7V$ arriba de su emisor o sea, $0.7V$. El transistor Q_2 está en ON ya que su colector está cuando menos a $0.1V$ arriba de su emisor; por esto $V_{B1} > 0.1V + 0.7V = 0.8V$. Como Q_3 está en ON, el voltaje en el emisor de Q_4 es $V_{E4} = V_{ol}(\text{min}) + V_{D3} = 0V + 0.7V$. De aquí que la unión base emisor de Q_4 está inversamente polarizada debido a que su emisor esté $0.7V$ y su base a $0.8V$, por lo tanto Q_4 está en OFF.

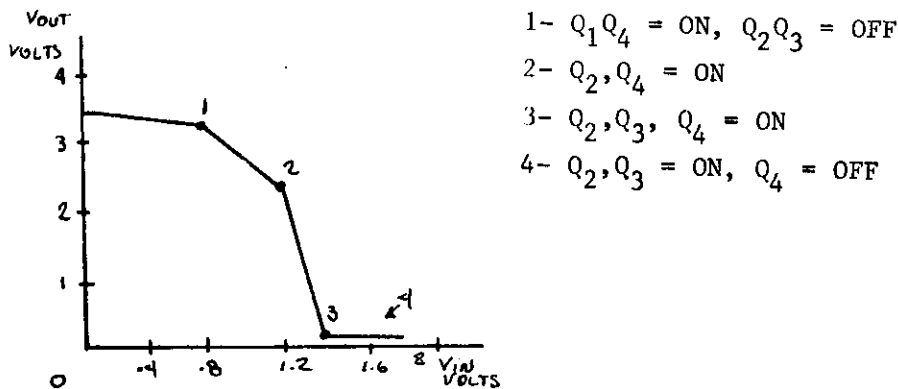
El propósito de D_3 es asegurar que Q_4 se apague por la caída de $0.7V$ que produce.

Entonces con una o más entradas en "0" la salida es un "1" y con todas las entradas en "1" la salida es un "0" y el circuito se comporta como un NAND.

B.1.2 CARACTERISTICAS DE LA FAMILIA LOGICA TTL

CURVAS DE TRANSFERENCIA DE VOLTAJE

La curva de transferencia de voltaje es una gráfica del voltaje de entrada contra el voltaje de salida.



La importancia de esta curva es que se muestran simultáneamente el V_{OH} , las regiones de transición y los voltajes de entrada. Nótese que el cambio de un nivel a otro no es instantáneo, sino que existe una zona llamada región de transición, de 1 a 2 debido a que Q_2 actúa como un amplificador lineal y de 2 a 3 por la conducción de Q_3 . La operación de un dispositivo digital en esta región puede causar que la salida oscile.

El voltaje de mantenimiento (Threshold) promedio es aquel cuando el voltaje de entrada es igual al voltaje de salida.

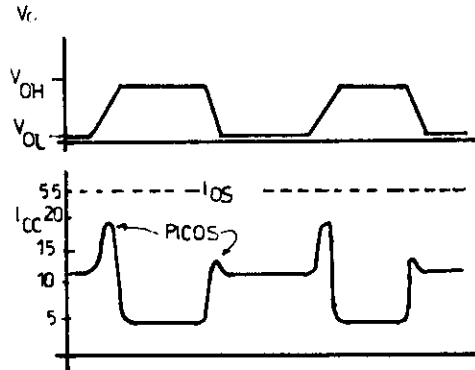
CORRIENTE DE CORTO CIRCUITO

La corriente de corto circuito I_{OS} es la corriente que puede fluir hacia la salida de un dispositivo cuando ésta se conecta a tierra. Puede ser interpretada como la capacidad del circuito para cargar una capacitancia externa rápidamente en el estado alto "1 log". Sin embargo la I_{OS} no es indicativo de la capacidad de carga de corriente directa.

CORRIENTE DE PICO

La corriente de pico se manifiesta como pequeños pulsos en la línea de la fuente de alimentación. En los cambios de estado haciéndose más notorios en el cambio de "0" a "1", esto se debe a que en la región de transición los transistores Q_3 y Q_4 se encuentran en ON por un pequeño período de tiempo, lo cual causa que la fuente se aterrice a través de una

pequeña resistencia, para prevenir que los picos aparezcan como ruido en el sistema, se debe colocar un capacitor (0.01 a 0.1 μ f) de desacoplo entre V_{cc} y tierra (GND).



IMPEDANCIA

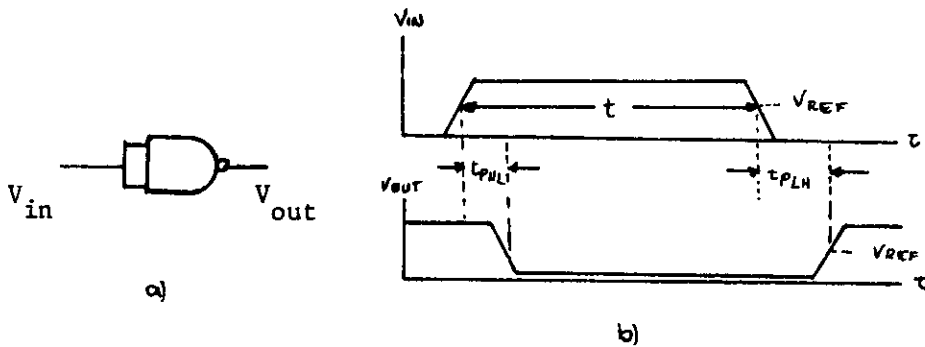
La impedancia de salida Z_0 de una compuerta TTL es bastante baja debido a la configuración totem-pole de los transistores de salida Q_3 y Q_4 . En el estado alto Q_4 actúa como un seguidor emisor y en el estado bajo está saturado, en ambos casos la impedancia de salida es muy baja, $Z_{OL} = 10$ ohms en el estado bajo y $Z_{OH} = 70$ ohms en el estado alto. La impedancia de entrada es de $Z_{IH} = 400$ K y $Z_{IL} = 4$ Kohms.

La ventaja de la alta impedancia de entrada es que se necesita menos corriente para activar las entradas y la ventaja de una baja impedancia de salida es incrementar la velocidad de switcheo cuando se manejan cargas capacitivas.

TIEMPO DE PROPAGACION

El tiempo de propagación para un dispositivo digital es el intervalo de tiempo requerido para que la salida responda a una entrada, medido respecto al nivel de referencia ($V_{REF} = 1.5V$).

Cuando la entrada cambia a un nivel alto la salida cambia a un nivel bajo después de un tiempo llamado turn-on delay time = T_{PHL} (Tiempo de Encendido).



RETARDOS DE PROPAGACION a) DIAGRAMA LOGICO, b) VOLTAJE DE ENTRADA Y SALIDA MOSTRANDO EL TPLH Y TPHL.

El TPHL Típico para una compuerta NAND 7400 es de 7 ns. Cuando la entrada cambia a un nivel bajo la salida cambia a un nivel alto después de un tiempo llamado turn-off delay time T_{PLH} . El TPLH típico para una compuerta NAND 7400 es de 11 nseg.

El tiempo promedio de retardo es de $\frac{7 \text{ ns} + 11 \text{ ns}}{2} = 9 \text{ ns}$

El tiempo de retardo típico para la serie TTL es usualmente de 10 ns, y se ve afectado por:

- 1) Cargas capacitivas
- 2) Fuente de poder
- 3) Temperatura

La máxima frecuencia de operación para una compuerta es aproximadamente:

$$f_{\text{max}} = \frac{1}{T_{PLH} + T_{PHL}} = \frac{1}{(11 + 7)} = 55 \text{ MHz.}$$

COMPUERTAS Y ENTRADAS NO USADAS

Las entradas abiertas o libres de una compuerta son llamadas entrada-flotantes y tienen el mismo efecto en la operación del circuito que un nivel alto l log. Las entradas libres permiten que entren voltajes de rui-

do por las terminales del circuito integrado, las cuales actúan como antenas. Unos cuantos milivolts negativos de ruido serían suficientes para cambiar una entrada flotante a un estado bajo, 0 log, por tal motivo es aconsejable conectar todas las entradas libres ya sea a un nivel alto o un nivel bajo. Las entradas libres de un AND o NAND (con entradas multi-emisor) pueden conectarse a otras entradas que si se estén usando, si esto puede representar problemas de carga deben alambrarse permanentemente a un nivel alto, por ejemplo:

1. Al Vcc si este no excede 5.5V. (Vin max).
2. A otra fuente de 2.4V a 5.5 V
3. A una compuerta no usada cuya salida esté a un nivel alto.
4. Al Vcc a través de una resistencia limitadora de corriente de 1K.

Lo mismo se recomienda para un OR y NOR, con la variante que aquí deben conectarse a tierra. Se recomienda también forzar las salidas de las compuertas no usadas a un nivel alto, con lo cual se reduce la disipación de potencia y se previene la oscilación de las compuertas.

FAN OUT

Es el máximo número de entradas que pueden conectarse a la salida de una compuerta ya sea en nivel alto o bajo, sin que sus niveles de voltaje se alteren y es un promedio de 10.

B.1.3 OTROS DISPOSITIVOS TTL

DISPOSITIVOS DE COLECTOR ABIERTO (OPEN COLLECTOR)

A menudo es necesario conectar las salidas de diversos dispositivos como compuertas lógicas para:

- 1) Incrementar el fan-out
- 2) Desarrollar alguna función lógica
- 3) Conectar varios dispositivos a una línea común (bus).

Supongamos que las salidas de dos compuertas del tipo totem-pole fueran conectadas. Si una de las salidas estuviera en un nivel alto y la otra en un nivel bajo, la compuerta en nivel alto drenaría hacia la del nivel bajo una corriente de corto circuito $I_{OS(max)} = 55 \text{ mA}$, lo cual dañaría al transistor cuya $I_{OL} = 16 \text{ mA}$.

Las compuertas de colector abierto se diseñan especialmente para poder conectar sus salidas al mismo punto como se muestra en la fig. B.6.

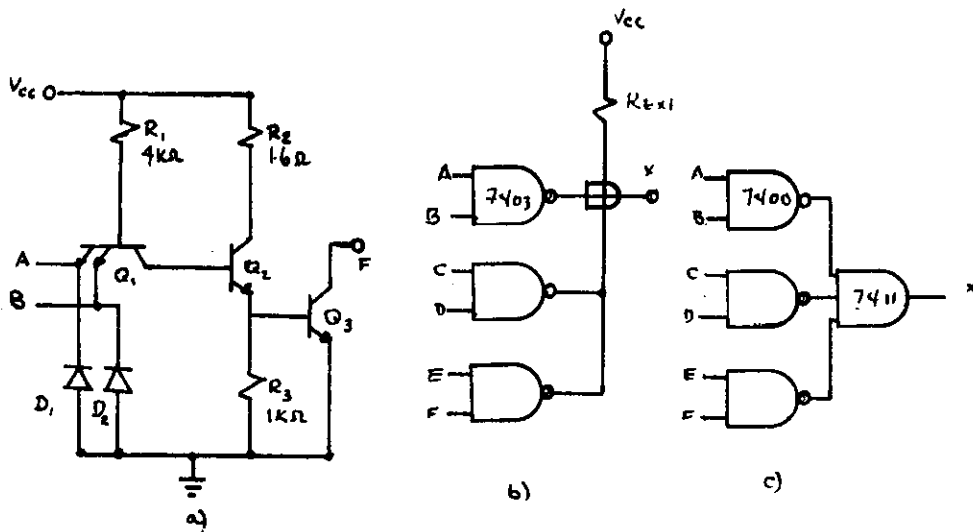


Fig. B.6 COMPUERTAS NAND DE COLECTOR ABIERTO, a) CONFIGURACION INTERNA b) DIAGRAMA LOGICO PARA UN WIRE AND, AND (alambrado), c) IMPLEMENTACION DE b) USANDO COMPUERTAS ESTANDARD.

El circuito del inciso b requiere únicamente 3 compuertas y un solo - circuito integrado, con lo cual se ahorra una compuerta y un circuito integrado en comparación con el inciso c).

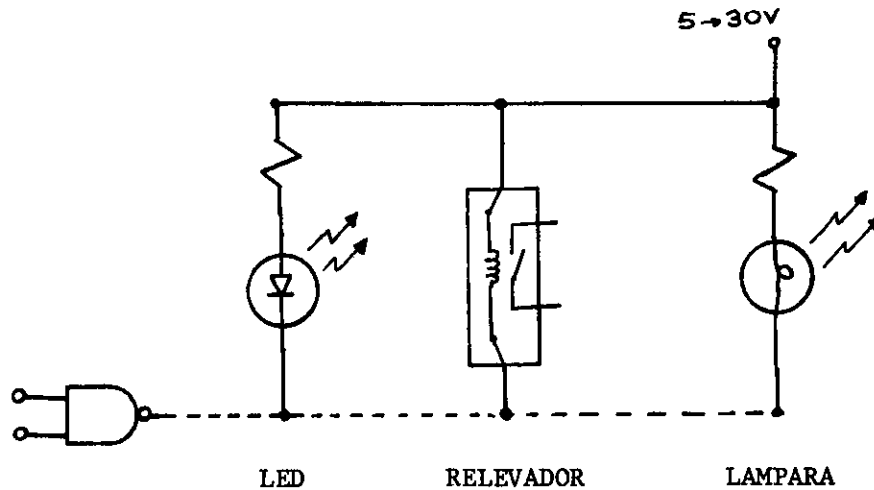
Ventajas: Se pueden ahorrar uno o más niveles de compuertas con los - respectivos tiempos de propagación y disipación de potencia.

Desventajas:

- 1) Se requiere una resistencia externa
- 2) Es más lento
- 3) Menor inmunidad al ruido
- 4) Menor capacidad para manejar cargas capacitivas.

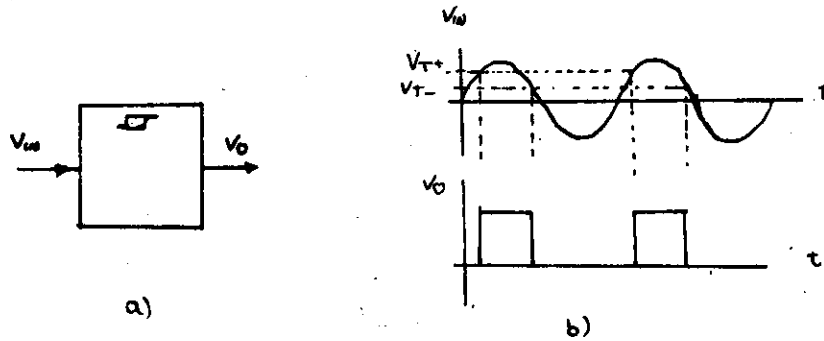
COMPUERTAS BUFFER/DRIVER

En una compuerta con una corriente y/o voltaje mayor en su salida. - Pueden manejar cargas a un voltaje de 30 V y 40 mA de corriente.



SCHMITT TRIGGER

El Schmitt trigger es un dispositivo cuya salida cambia a 1 log. cuando un voltaje de entrada ascendente excede el voltaje threshold positivo, - la salida permanecerá en este nivel hasta que un voltaje descendente alcance el voltaje Threshold negativo V_{T-} , en este momento la salida cambiará -- a un 0 Log. Los voltajes Threshold de un Schmitt Trigger generalmente se establecen por el fabricante y el usuario no puede cambiarlos.



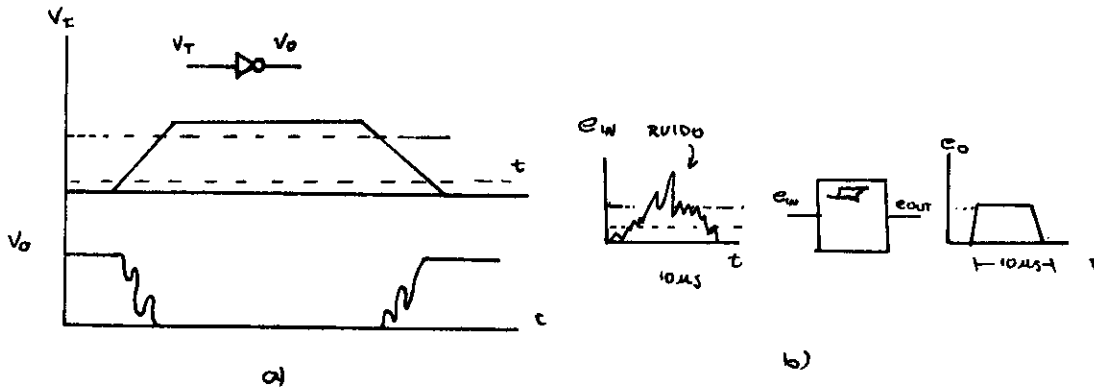
a) BLOQUE SCHMITT TRIGGER

b) VOLTAJES DE ENTRADA Y SALIDA.

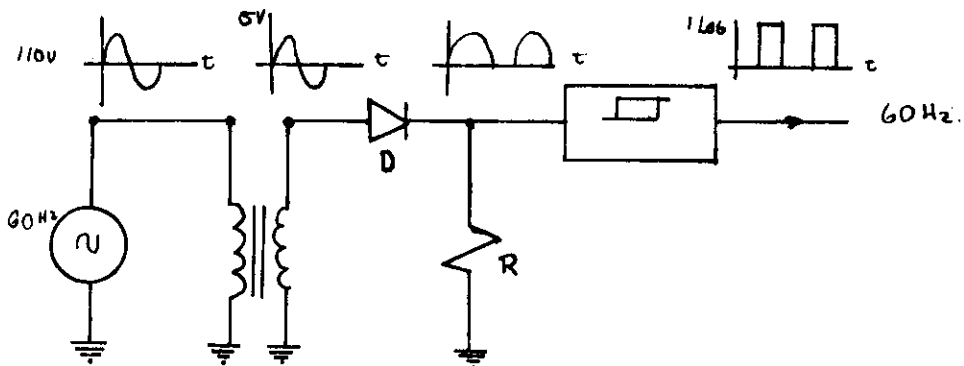
La principal aplicación del schmitt trigger es convertir voltajes cuyos tiempos de transición son muy cortos, por ejemplo, en la regeneración de señales de una línea de transmisión, al alimentar una señal lenta a un dispositivo lógico TTL ocurren problemas tales como: inestabilidad -- (salidas oscilatorias) y tiempos de propagación impredecibles.

A continuación se ilustra la señal alimentada a un NOT TTL con tiempos de transición de 10 seg. En el momento en que el voltaje de entrada está entre el $V_{IL(max)}$ y $V_{IH(min)}$ la compuerta NOT se encuentra en la región de transición y su salida será oscilatoria si permanece su entrada en este nivel por un tiempo mayor al tiempo de propagación.

Para prevenir problemas de inestabilidad en circuitos lógicos TTL se recomiendan tiempos de retardo máximos de 50 ns.



OSCILACIONES EN LA SALIDA DE UNA COMPUERTA NOT DEBIDO A UNA ENTRADA CON TRANSICIONES LENTAS.



GENERACION DE UNA SEÑAL RECTANGULAR (ONDA CUADRADA POSITIVA) DE 60 HZ. USANDO UN SCHMITT TRIGGER.

DISPOSITIVO THREE STATE

En sistemas digitales como computadoras y sistemas de comunicación de datos la información es transferida a lo largo de una línea común llamada bus de datos, la cual conecta dos o mas salidas a una o más entradas en la misma línea como se ilustra en la figura B.7c).

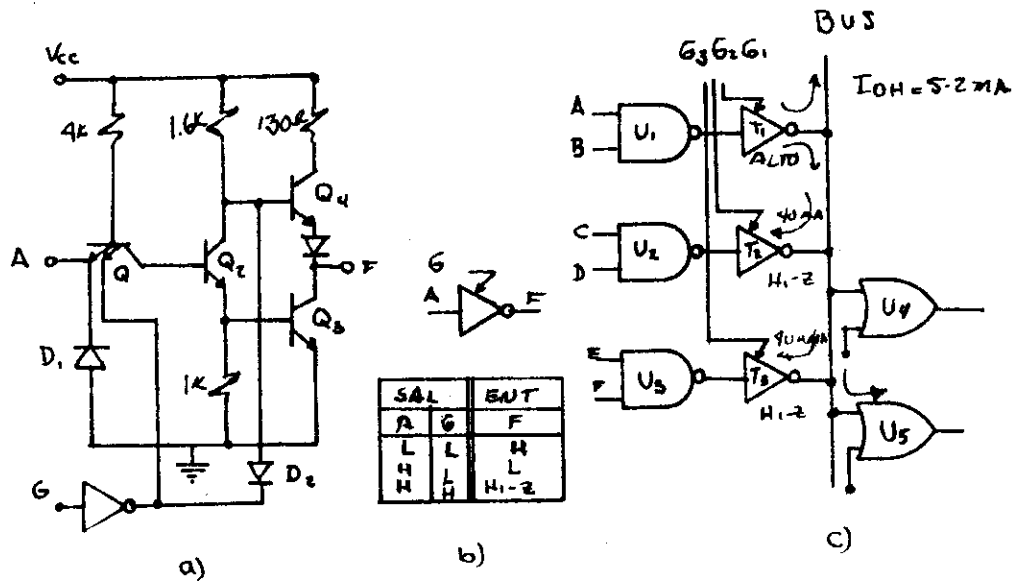


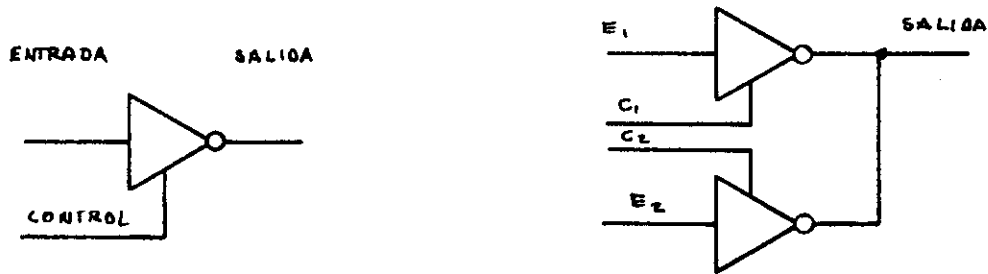
Fig. B.7 COMPUERTA THREE STATE. a) CIRCUITO DE UN NOT, b) SIMBOLO Y TABLA c) COMPUERTAS CONECTADAS A UN BUS.

El (Three-state) es un dispositivo especial que tiene tres estados. -- Los estados normales de baja impedancia, nivel bajo 0 log y nivel alto. -- 1 log. y un estado de alta impedancia. En la figura B.7 se muestra un inversor three-state. Cuando la entrada de control $G = 1$ log el voltaje de base para Q_4 es un nivel bajo y Q_4 está en OFF, también el colector de Q_2 está a un nivel bajo y por lo tanto Q_2 está en OFF.

La impedancia de salida es alta debido a que Q_3 y Q_4 están en OFF, -- cuando la señal de control $G = 0$ log el circuito opera como un inversor -- normal.

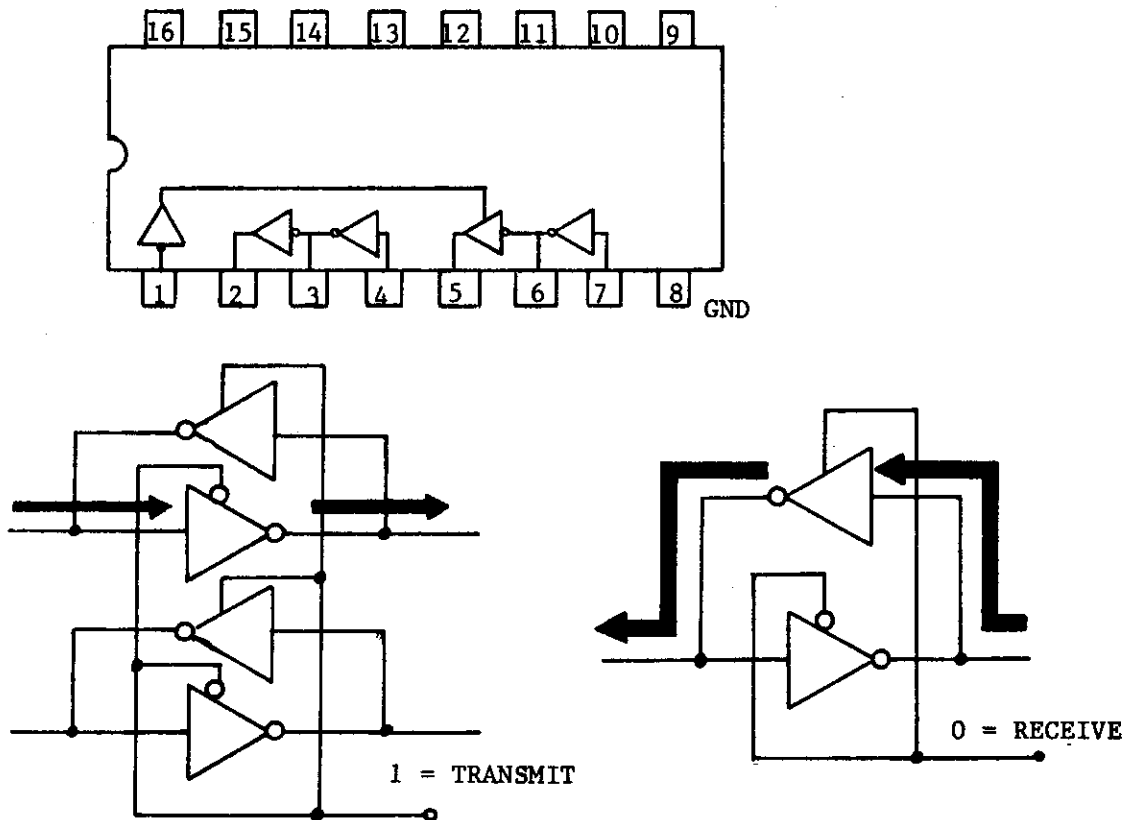
El símbolo y su tabla de verdad se muestran en la fig. B.7 b), en la fig. B.7c) se muestra un sistema con 3 compuertas con salida totem-pole - conectada al bus a través de 3 buffers de three-state. Supongamos que deseamos conectar N_1 al bus, entonces, $G_1 = 0$; $G_2 = 1$, $G_3 = 1$. En el estado de alta impedancia pueden drenar únicamente $40 \mu A$ de corriente de fuga de aquí que puedan conectarse 50 dispositivos al bus.

Existe también la posibilidad de transmitir información por una línea en ambas direcciones lo que se conoce como bus bidireccional, desde luego aquí también una sola compuerta debe habilitarse a la vez.

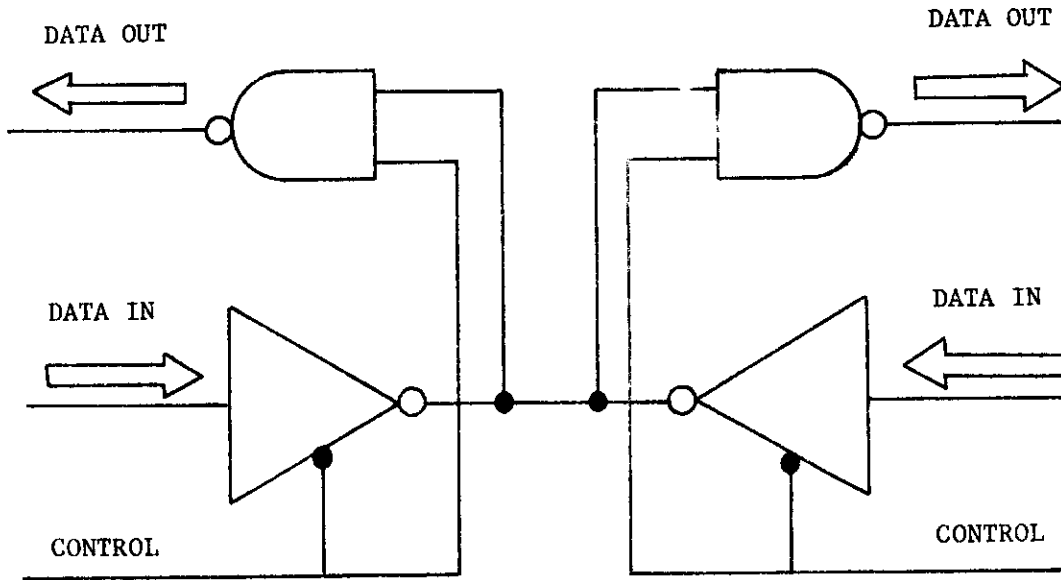


EJEMPLO B.0

Diseñar un Bus bidireccional con un Quad Bus Driver/Receiver.



BUS DE DATOS BIDIRECCIONAL

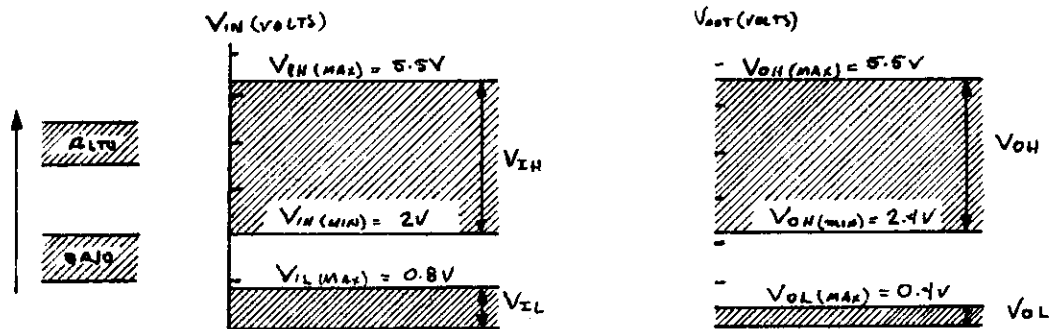


0 = TRANSMIT
1 = RECEIVE

0 = TRANSMIT
1 = RECEIVE

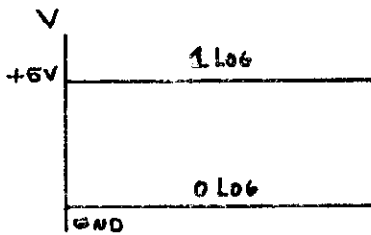
B.1.4 NIVELES DE VOLTAJE

Habíamos asumido hasta ahora que los voltajes altos V_H y bajos V_L , que representan los datos lógicos son: $V_H = 5V$ y $V_L = 0V$, desde luego -- los parámetros de un dispositivo lógico TTL varían con la temperatura, -- ruido electromagnético, etc. Los voltajes de operación tienen el siguiente rango:

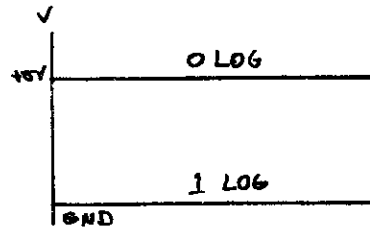


NIVELES DE VOLTAJE a) CASO GENERAL, b) RANGO DE ENTRADA, c) RANGO DE SALIDAS.

En general el "1 lógico" o nivel alto se define como el potencial -- más grande. Ya sea positivo o negativo, con respecto a tierra. Y un "0-lógico" como lo contrario.



LOGICA POSITIVA



LOGICA NEGATIVA

B.1.5 LOGICA POSITIVA Y NEGATIVA

Una tabla de verdad puede usarse para describir operaciones lógicas -- con referencia a un nivel de voltaje real respecto a un potencial de tierra.

Asumiendo que el mayor potencial H, (con respecto a tierra) de una compuerta lógica representa la lógica "verdadera", en tanto que el potencial bajo, L, representa la lógica "falsa". Con este convenio, el cual es llamado LOGICA POSITIVA, obtenemos una tabla de verdad como se muestra en la fig. B.8.

Tomando el convenio anterior invertido, una L representa la lógica "verdadera" y H representa la lógica "falsa". La tabla para esta LOGICA-NEGATIVA, se muestra en la fig. B.9. Comprobando las dos tablas observamos lo siguiente:

1. El AND positivo es igual al OR negativo
2. El OR positivo es igual al AND negativo
3. El NAND positivo es igual al NOR negativo
4. El NOR positivo es igual al NAND negativo
5. El EX-OR y el coincidence no se afectan por la representación con venida de la lógica "verdadera" o "falsa".

| ENTRADAS | | SALIDAS | | | | | |
|----------|---|---------|----|------|-----|-------|-------------|
| A | B | AND | OR | NAND | NOR | EX-OR | COINCIDENCE |
| L | L | L | L | H | H | L | H |
| L | H | L | H | H | L | H | L |
| H | L | L | H | H | L | H | L |
| H | H | H | H | L | L | L | H |

Fig. B.8 TABLA DE VERDAD PARA LA LOGICA POSITIVA

| ENTRADAS | | SALIDAS | | | | | |
|----------|---|---------|----|------|-----|-------|-------------|
| A | B | AND | OR | NAND | NOR | EX-OR | COINCIDENCE |
| L | L | L | L | H | H | H | L |
| L | H | H | L | L | H | L | H |
| H | L | H | L | L | H | L | H |
| H | H | H | H | L | L | H | L |

Fig. B.9 TABLA DE VERDAD PARA LA LOGICA NEGATIVA

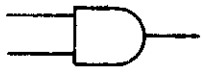











Los términos de lógica POSITIVA y NEGATIVA no deben tomarse para implicar potenciales positivos o negativos con respecto a tierra. Por ejemplo, los valores reales en algunas compuertas lógicas (RTL, DTL, o TTL) - se representan por 0 volts y aproximadamente + 3 volts, mientras que para la lógica acoplada por emisor los valores respectivos son de -0.75 volts- y -1.5 volts. Usaremos el término LOGICA POSITIVA cuando el más positivo de los datos potenciales, represente en 1 lógico.

La LOGICA NEGATIVA, usará cuando el más negativo de los potenciales- (el mas bajo) representa en 1 lógico, un circuito dado puede ser usado -- en una u otra convención lógica, sin embargo, su operación lógica debe -- redefinirse.

Dado que la misma compuerta físicamente puede ser usada para cumplir dos funciones lógicas, en la mayoría de los casos es conveniente representar esto con un símbolo gráfico un pequeño círculo en la entrada (s) o salida(s) de la compuerta indicará que el potencial más bajo de los dos está presente en el punto respectivo.

Con este convenio y dado la dualidad de las funciones de la compuerta (fig. B.8 y B.9) podemos dibujar símbolos gráficos para la lógica positiva y negativa como se muestra en la Fig. B.10. Note que cada hilera muestra el mismo dispositivo físico y la función lógica que ejecuta. La dualidad de funciones lógicas de las 4 hileras de arriba pueden demostrarse fácilmente con el teorema de D'Morgan.

Distintos fabricantes de circuitos integrados especifican la dualidad de las funciones designándolos como AND/OR o NAND/NOR, etc., o por el uso del término "lógica positiva". La mayoría de las especificaciones y algunas de las discusiones siguientes ignoran el aspecto complementario y presentan a las compuertas como AND, OR, NAND, NOR, etc.

| NOMBRE DE LA COMPUERTA EN LOGICA POSITIVA | SIMBOLO GRAFICO | SIMBOLO GRAFICO | NOMBRE DE LA COMPUERTA EN LOGICA NEGATIVA |
|---|---|--|---|
| AND |  |  | OR |
| OR |  |  | AND |
| NAND |  |  | NOR |
| NOR |  |  | NAND |
| EX-OR |  |  | EX-OR |
| COINCIDENCE |  |  | COINCIDENCE |

282

Fig. B.10 SIMBOLOS GRAFICOS PARA LA LOGICA POSITIVA Y NEGATIVA.

283

B.2 FAMILIA LOGICA MOS

Las principales ventajas del MOSFET son de que es relativamente simple y barato de fabricar, es de tamaño pequeño y consume muy poca potencia. La fabricación de los circuitos integrados MOS es aproximadamente la tercera parte de compleja que la fabricación de los circuitos integrados bipolares (TTL, ECL, etc.) adicionalmente, los dispositivos MOS ocupan mucho menos -- espacio en un paquete que los transistores bipolares; típicamente, un MOSFET requiere 1 milésimo de pulgada cuadrada de área de el paquete mientras que un transistor bipolar requiere alrededor de 50 milésimas de pulgada cuadrada. Más importante, los circuitos integrados digitales MOS normalmente no usan elementos resistivos, los cuales toman tanto el área del paquete en -- los circuitos integrados bipolares.

Todo esto significa que los circuitos integrados MOS pueden acomodar -- un mayor número de elementos circuitales en un solo paquete que los circuitos integrales bipolares. Esta ventaja se evidencia por el hecho que los -- circuitos integrados MOS están sobrepasando a los circuitos integrales bipolares en el área de integración en gran escala, (LSI) la gran densidad de -- empaque de los circuitos integrales MOS resulta en una mayor confiabilidad del sistema debido a la reducción en el número de conexiones externas necesarias, la principal desventaja de los circuitos integrales MOS es su relativa velocidad baja de operación cuando se compara con las familias de circuitos integrales bipolares. En muchas aplicaciones esto no es una consideración primaria, así que la lógica MOS ofrece una alternativa a menudo superior a la lógica bipolar.

B.2.1 CARACTERISTICAS DE LA LOGICA MOS

VELOCIDAD E IMPEDANCIA

Comparadas con las familias lógicas (TTL, ECL) las familias lógicas -- MOS son típicamente 10 veces más lentas en velocidad de operación, requieren mucha menos potencia, tienen mejor margen de ruido y un fan-out más alto y, como se mencionó antes, requieren mucho menos espacio (área del paquete).

Una compuerta nand P-MOS típica tiene un retardo de tiempo de propagación de 100 ns. Esto se debe a dos factores: La resistencia de salida relativamente alta (100 ohms) en el estado alto y la carga capacitiva presenta-

da por las entradas de los circuitos lógicos que se están manejando. Las entradas lógicas MOS tienen una resistencia de entrada muy alta ($>10^{12}$ - ohms) pero poseen una capacitancia de compuerta razonablemente alta (capacitor MOS), típicamente 2-5 picofaradios. Esta combinación de R grande y C carga alta sirve para aumentar el tiempo de conmutación.

MARGEN DE RUIDO

Típicamente los márgenes de ruido MOS son alrededor de 2V, lo cual es sustancialmente mayor que para TTL o ECL.

FAN-OUT

Debido a la extremadamente alta resistencia de entrada en cada entrada MOSFET uno esperaría que las capacidades de FAN-OUT de lógica MOS serían ilimitadas. Esto es esencialmente correcto para operación DC o en baja frecuencia. Sin embargo, para frecuencias mayores de 100 KHz, las capacitancias de entrada de las compuertas causan un deterioro en el tiempo de conmutación el cual aumenta en proporción al número de cargas que se están manejando. Aún así, la lógica MOS puede operar fácilmente con un FAN-OUT DE 50, lo cual es algo mejor que para las familias bipolares.

DISIPACION DE POTENCIA

Los circuitos lógicos MOS toman pequeñas cantidades de potencia debido a las resistencias relativamente altas que se usan. La Baja Disipación de potencia de la lógica MOS la hace apropiada para LSI, donde muchas compuertas, FF, etc. pueden estar en un mismo paquete sin ocasionar recalentamiento.

COMPLEJIDAD DE PROCESO

La familia lógica MOS es la más simple para fabricar puesto que usa solo un elemento básico, un transistor P-MOS (o N-MOS). No requieren otros elementos tales como resistencia, diodos, etc.

La velocidad de operación y los niveles de voltajes de p-MOS y n-MOS no son compatibles con TTL, así que muy poco se ha hecho con ellas en aplicaciones de SSI o MSI.

B.2.2 CONFIGURACION DE BLOQUES LOGICOS

INVERSOR

El transistor de carga Q_2 (equivalente a una resistencia de carga de un transistor bipolar) normalmente está conduciendo debido al V_{GG} , y el transistor driver Q_1 está conduciendo dependiendo del voltaje de entrada V_{in} .

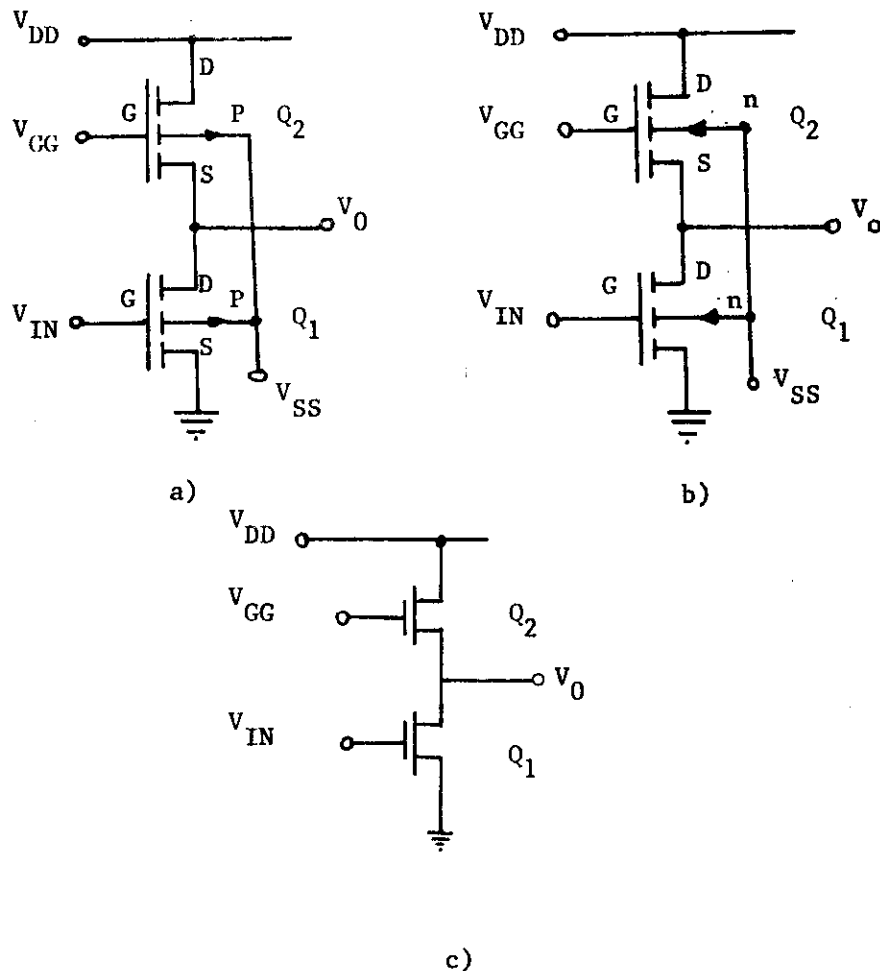


Fig. B.11 INVERSOR MOS: (a) CANAL P, (b) CANAL N, (c) SIMBOLO DEL INVERSOR CON POLARIDAD (P o N). D = drenaje, S = FUENTE, G = COMPUERTA, V_{DD} = VOLTAJE DE ALIMENTACION DE DRENAJE, V_{SS} = VOLTAJE DE ALIMENTACION DE SUSTRATO.

Una COMPUERTA NOR puede implementarse conectando otro transistor en paralelo al transistor Q_2 del inductor básico.

La figura B.12 muestra una compuerta NOR de dos entradas, su símbolo y la tabla de verdad. Si A o B son niveles altos, Q_2 o Q_3 estarán en ON, y el voltaje de salida será un nivel bajo (0 volts).

La función equivalente del NAND ocurre cuando ambas entradas A y B son niveles bajos. Esto hace que los transistores Q_2 y Q_3 dejen de conducir resultando un nivel alto en la salida.

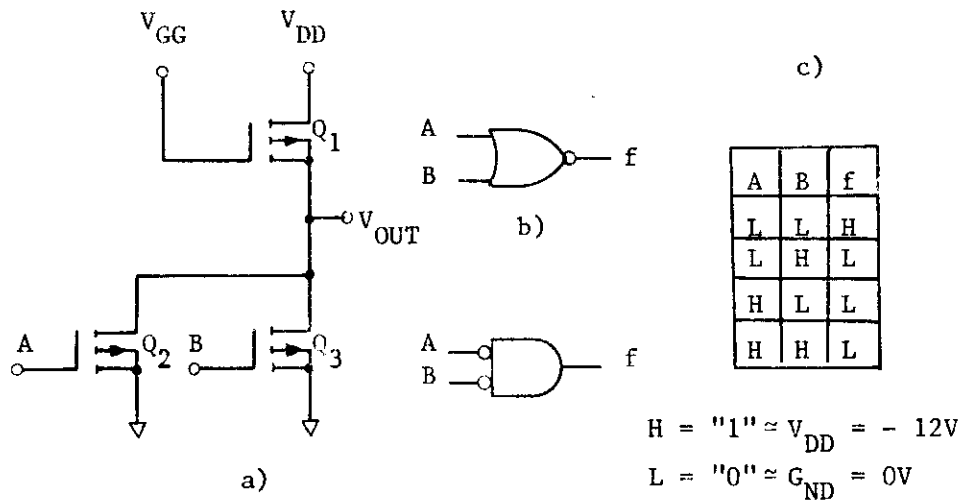


Fig. B.12 COMPUERTAS NOR DE DOS ENTRADAS: (a) DIAGRAMA ESQUEMATICO, (b) SIMBOLO, (c) TABLA DE VERDAD.

COMPUERTA NAND

El circuito del inversor básico puede modificarse para formar una compuerta NAND, colocando un transistor adicional Q_3 en serie con el transistor Q_2 como se muestra en la figura B.13.

Si las entradas A y B están en un nivel alto los transistores Q_2 y Q_3 conducen y el voltaje de salida es un nivel bajo (0 volts). La función equivalente NOR ocurre cuando las entradas A o B están en un nivel bajo por lo tanto Q_2 o Q_3 no conducirán la corriente resultante en este caso es cero y la salida deberá ser un nivel alto.

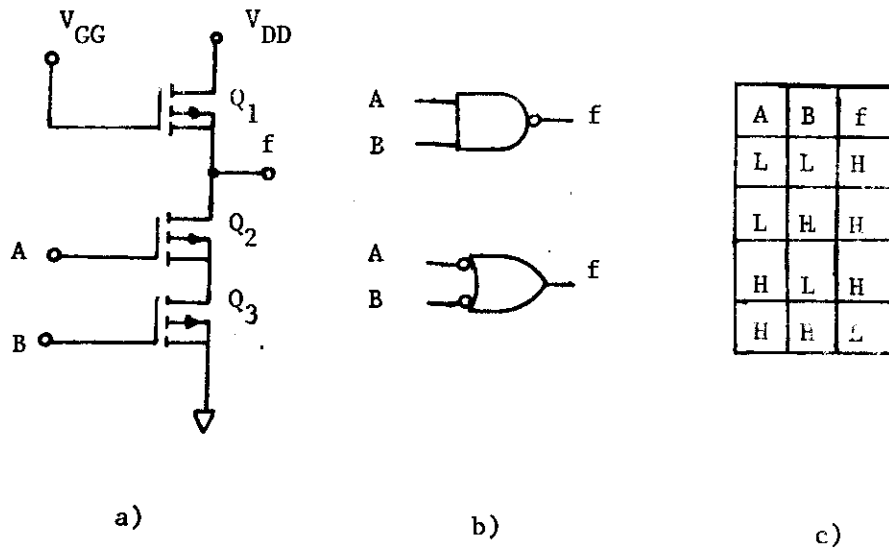


Fig. B.13 COMPUERTA NAND DE DOS ENTRADAS: (a) DIAGRAMA ESQUEMATICO, (b) SIMBOLO, (c) TABLA DE VERDAD.

COMPUERTA EX-OR

La implementación de la función OR exclusivo empleando compuerta estática MOS se muestra en la siguiente figura B.14.

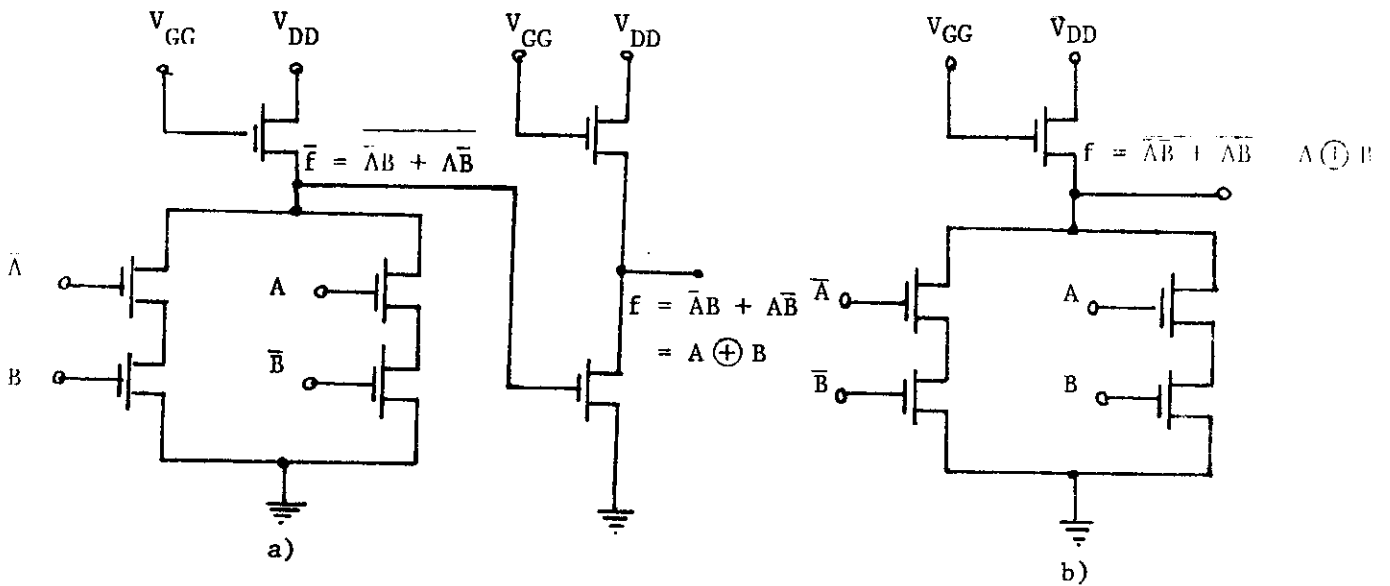


Fig. B.14 (a) IMPLEMENTACION DE UNA FUNCION OR-EXCLUSIVO.

B.3 FAMILIA LOGICA C-MOS

La familia CMOSL (complementary MOS logic) tiene baja disipación de potencia de tan solo 6nW por compuerta y un gran margen de ruido de dc. Los dispositivos CMOS consisten de pares de transistores de canal-N y canal-P en su modo de enriquecimiento.

El CMOS es utilizado en aplicaciones que requieren bajo consumo de energía (relojes de pulsera, equipo portátil, etc.) y en ambientes ruidosos (plantas industriales). La baja disipación de potencia se debe a que, bajo condiciones estáticas, cuando un transistor en cada par está en OFF, el otro estará en ON.

La gran mayoría de los LSI (large Scale Integration) como las memorias (Flip-Flop), usan transistores N-MOS y P-MOS porque éstos son más pequeños en tamaño que los transistores C-MOS.

Los voltajes típicos de alimentación V_{DD} son de + 3V a + 18V, con la alimentación V_{SS} aterrizada. Los voltajes de entrada y de salida dependen del voltaje de alimentación:

$$V_H \approx V_{DD} \quad (V_{SS} = 0)$$

$$V_L \approx 0V$$

B.3.1 CIRCUITOS BASICOS

La figura B.15(a) muestra el circuito para una compuerta "NOT". Esta consiste en un transistor de un canal P y un canal N en modo de enriquecimiento. El sustrato y la fuente del canal P se conectan directamente al voltaje positivo de alimentación V_{DD} . El drenaje del canal P se conecta al drenaje del canal N del transistor. El sustrato y la fuente del canal N se conecta a tierra. Las compuertas de los dos transistores se conectan unidas y forman la entrada de la compuerta "NOT". La salida se toma de las terminales comunes del drenaje.

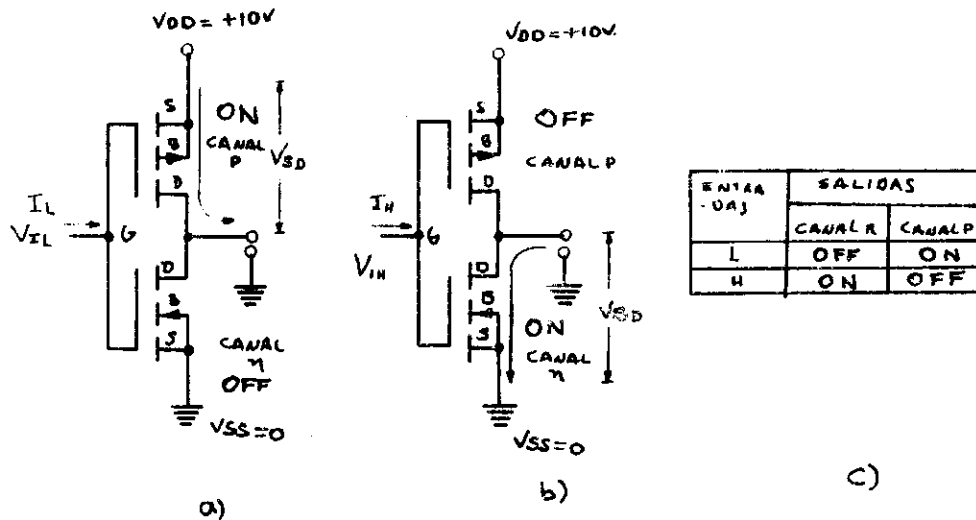


Fig. B.15 COMPUERTA NOT C MOS: (a) ENTRADA BAJA; (b) ENTRADA ALTA, (c) TABLA DE SWITCHEO PARA TRANSISTORES N-MOS Y P-MOS.

Si aplicamos un nivel bajo (0V) a la entrada de la compuerta "NOT" - con un $V_{DD} = +10V$ y $V_{SS} = 0V$. El canal N del transistor está en "OFF" - porque V_{GS} es menor que V_T . El canal P del transistor se pone en "ON" debido a que su compuerta es más negativa que V_T . Por lo tanto la salida se conecta directamente a V_{DD} a través del canal P del transistor. Cuando el transistor MOS está en "ON" tiene una caída de voltaje de 10 mV entre su fuente y el drenaje, ($V_{SD} \approx 10 \text{ mV}$). La salida de la compuerta "NOT" en la figura B.15(a) es alta, cuando $V_{OH} = V_{DD} - V_{SD} = 10V - 10 \text{ mV} \approx 10V$. La salida de la fuente de salida se llama I_{OH} .

Cuando la entrada es alta (+10V), el transistor de canal N está en "ON" porque $V_{GS} > V_T$ como se muestra en la fig. B.15(b). El transistor de canal P se pone en OFF porque $V_{GS} < V_T$. Por lo tanto la salida se conecta a V_{SS} (tierra), así es que la salida es baja, cuando $V_{OL} = V_{SS} - V_{SD} = 0 - 10\text{mV} \approx 0V$. La disipación de la corriente de salida es I_{OL} . Por lo tanto en el circuito de la fig. B.15 se efectúa la operación lógica "NOT". En la fig. B.15(c) se resumen las condiciones para el switcheo del transistor de canal N y canal P.

En la figura B.16(a) se muestra un esquema para compuerta NOR C MOS -

14 001. Esta consiste de dos transistores de canal P en serie y dos --- transistores de canal N en paralelo. Cada entrada se conecta una al canal N y la otra al canal P.

Si la entrada A es alta el transistor de canal P se pone en OFF, -- el cual desconecta V_{DD} de la salida. El transistor del canal N para "A" se pone en "ON", así la salida está conectada a 0V (bajo). Cuando ambas entradas son bajas, ambos transistores de canal P se ponen en "ON" y ambos transistores de canal N se ponen en OFF.

En éste caso el V_{DD} se conecta a la salida, así la salida "Y" es alta. Por lo tanto este circuito ejecuta la operación lógica NOR.

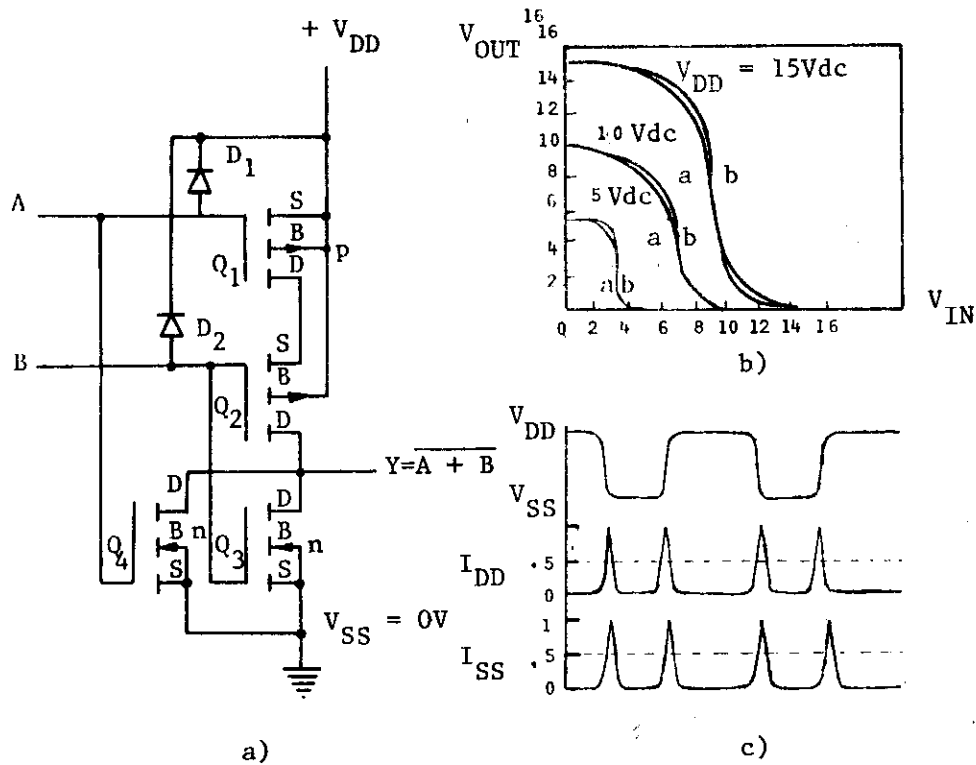


Fig. B.16 COMPUERTA NOR C MOS L: (a) ESQUEMA; (b) CURVAS DE VOLTAJE TÍPICAS; (c) CORRIENTES DE PICO EN LAS LINEAS DE ALIMENTACION (I_{DD}) Y TIERRA I_{SS} .

Los diodos D_1 y D_2 en la entrada de la compuerta NOR en la fig. B.16 (a), la cual ilustra un método para proteger el transistor contra los altos campos eléctricos. Estos diodos pueden operar en la región de polarización directa con corrientes abajo de 10 mA. Su voltaje de pico inverso es de 30V.

Las ventajas de los dispositivos C MOS sobre los dispositivos MOS -- los cuales usan solo transistores de canal N o transistores de canal P, -- es su mayor velocidad, baja disipación de potencia, gran inmunidad al ruido, gran fan-out y además no necesitan una perfecta regulación de la fuente de poder.

B.3.2 CARACTERÍSTICAS DE LA SERIE CMOS 4000A

La serie 4000A es la línea más ampliamente usada de C.I. digitales C MOS. Contienen unas pero no todas las funciones disponibles en la serie TTL 7400 y está en expansión constante.

DISIPACION DE POTENCIA

La disipación de potencia en el estado estático (dc) de los circuitos lógicos C MOS es extremadamente baja. Para estos circuitos en cualquier estado de salida, no hay nunca un cambio de baja resistencia desde V_{DD} hasta tierra; esto es, para cualquier condición de entrada siempre -- hay un MOSFET apagado en el camino de la corriente. Este hecho resulta en disposiciones de potencia típicas en dc de 12 n watts por compuerta -- usando $V_{DD} = 10$ volts.

NIVELES DE VOLTAJE

Los niveles lógicos de voltaje C MOS son 0 volts para "0" lógico y $+V_{DD}$ para "1" lógico.

El suministro $+V_{DD}$ puede estar en el rango de 3 a 18 volts para la serie 4000A. Cuando se usa C MOS o TTL, el voltaje de la fuente se hace 5 volts así que los niveles de voltaje de las familias sean los mismos.

Los niveles de entrada requeridos para C MOS dependen de V_{DD} como sigue:

$$V_{IL} = 30\% \times V_{DD}$$

$$V_{IH} = 70\% \times V_{DD}$$

VELOCIDAD DE CONMUTACION

C MOS, como P-MOS y N-MOS sufre de las grandes capacidades de carga causadas por las entradas MOS que se manejan, cada entrada C MOS es típicamente una carga de 5 pF.

La velocidad de la familia C MOS 4000 A varía con el voltaje de la fuente. Un valor alto V_{DD} produce valores menores de resistencia de encendido por lo cual produce una conmutación más rápida, debido a la carga más rápida de las capacidades. Esto significa que para aplicaciones en frecuencias mayores es mejor usar un valor alto de V_{DD} (hasta 15V para la serie 4000A).

EFFECTO DE LA FRECUENCIA EN P_D

Cuando los circuitos lógicos C MOS están en un estado estable por largos períodos de tiempo o conmutando a bajas frecuencias, entonces la disipación de potencia será extremadamente baja. A medida que la frecuencia de conmutación aumenta, la disipación promedio de potencia aumentará proporcionalmente, esto es, porque cada vez que la salida C MOS conmuta a ALTO, una corriente transitoria de carga debe de ser suministrada a cualquier capacitancia de carga. Estos pulsos momentáneos de corriente vienen de la fuente V_{DD} . A medida que aumenta la frecuencia, la corriente promedio y por consiguiente la potencia promedio P_D tomada de la fuente V_{DD} también aumentará.

MARGEN DE RUIDO

La serie C MOS 4000A tiene el mismo margen de ruido tanto en el estado alto como en el bajo. Los valores de V_{NL} y V_{NH} están garantizados de ser el 30% del voltaje V_{DD} de la fuente. En la práctica un margen de ruido típico mayor del 30% de V_{DD} (probablemente del 45% de V_{DD}), pero 30% es lo que el fabricante garantiza en los peores casos.

FAN-OUT

La capacitancia de entrada de C MOS llega a ser un factor limitante cuando la capacitancia total de carga se hace lo suficientemente alta para limitar la velocidad de conmutación del circuito debajo de la requerida para la aplicación. Así el fan-out de C MOS está limitado por las capacitancias de las entradas.

ENTRADAS NO USADAS

Todas las entradas C MOS deben estar conectadas a un nivel de voltaje preferiblemente a tierra o V_{DD} . Las entradas no usadas no pueden dejarse desconectadas, porque serían susceptibles al ruido el cual podría polarizar los canales de los MOSFET P y N al estado conductor, resultando una excesiva disipación de potencia. Las entradas no usadas pueden también conectarse a unas de las entradas usadas, siempre y cuando no exceda al FAN-OUT de la fuente de señal. Esto es altamente improbable debido al alto fan-out de C MOS.

B.4 FAMILIA LOGICA ECL

La familia lógica TTL (con la excepción de Schottky TTL) usa transistores que operan en el modo saturado. Como resultado, su velocidad de conmutación está limitada por el tiempo de retardo de almacenamiento de carga asociado con un transistor que se conduce a saturación. Otra familia lógica bipolar se ha desarrollado de tal forma que previene la saturación de transistores, aumentando por consiguiente la velocidad total de conmutación. Esta familia lógica se llama LOGICA ACOPLADA POR EMISOR (ECL) y opera basada en el principio de conmutación de corriente por el cual una corriente fija de polarización menor que $I_c(\text{sat})$ se conmuta del colector de un transistor a otro. Debido a esta operación en modo de corriente, esta forma lógica se denomina como lógica en modo de corriente (CML).

B.4.1 CARACTERISTICAS DE LA FAMILIA LOGICA ECL

1. Los transistores nunca se saturan de tal modo que la velocidad de conmutación es muy alta. Tiempo típico de retardo en propagación de 2ns, lo cual hace a ECL un poco más rápida que la serie Schottky TTL (serie 54S/74S). Aún cuando la serie 54S/74S es casi tan rápida como ECL, requiere un proceso de fabricación algo más complejo, así que es un poco más costosa.
2. Los niveles lógicos son nominalmente de -0.8V y - 1.70V para el "1" lógico y "0" lógico respectivamente.
3. Los márgenes de ruido para ECL en el peor caso son aproximadamente de 250 mv. Estos márgenes de ruido bajos hacen la serie ECL - algo poco confiable para uso en ambientes industriales pesados.
4. Un bloque lógico ECL produce generalmente una salida y su complemento. Esto elimina la necesidad de inversores.
5. Los Fan-outs son típicamente alrededor de 25, debido a las salidas de baja impedancia de los seguidores de emisor.
6. Disipaciones típicas de potencia para una compuerta básica ECL --- de 25 mW, justo algo mayor que Schottky TTL.
7. El flujo total de corriente en un circuito ECL permanece relativamente constante sin importar su estado lógico. Esto ayuda a mantener un drenaje invariable de corriente de alimentación aún durante transiciones conmutadas. Así, no se generan picos de ruido inter-

namente como los producidos por circuitos TTL en poste totémico.

B.4.2 CIRCUITO BASICO ECL

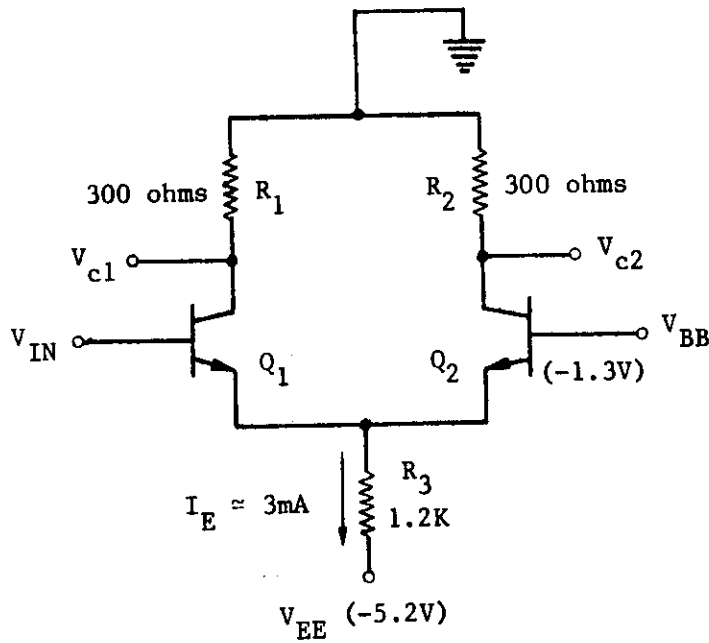
El circuito básico para lógica acoplada por emisor (ECL) es esencialmente la configuración de amplificador diferencial de la fig. B.16a) el suministro V_{EE} produce una corriente esencialmente fija I_E que permanece alrededor de 3mA durante operación normal. Se permite que ésta corriente fluya a través de Q_1 o Q_2 dependiendo del nivel de voltaje en V_{in} . En otras palabras, esta corriente conmutará entre el colector de Q_1 y el colector de Q_2 a medida que V_{IN} conmuta entre sus dos niveles lógicos de -1.7V (0 lógico para ECL) y -0.8V (1 lógico para ECL). La tabla en la fig. B.16 muestra los voltajes resultantes de salida para estas dos condiciones en V_{IN} . Deberían notarse dos puntos importantes: (1) V_{C1} y V_{C2} son complementarios el uno con respecto al otro y (2) los niveles de voltaje de salida no son los mismos que los niveles lógicos a la entrada.

Del segundo punto descrito arriba se hace un arreglo conectando V_{C1} y V_{C2} a etapas seguidoras de emisor (Q_3 y Q_4), como se muestra en la fig. B.16b). Los seguidores de emisor cumplen dos funciones: (1) sustraen aproximadamente 0.8V de V_{C1} y V_{C2} para desplazar los niveles de salida a los niveles lógicos correctos ECL y (2) proveen muy baja impedancia de salida (típicamente 7 ohms) lo cual provee un fan-out grande y carga rápida de capacitancia de carga. Este circuito produce dos salidas complementarias: V_{out1} el cual es igual a V_{in} y V_{out2} el cual es igual a V_{in} .

B.4.3 COMPUERTAS LOGICAS ECL

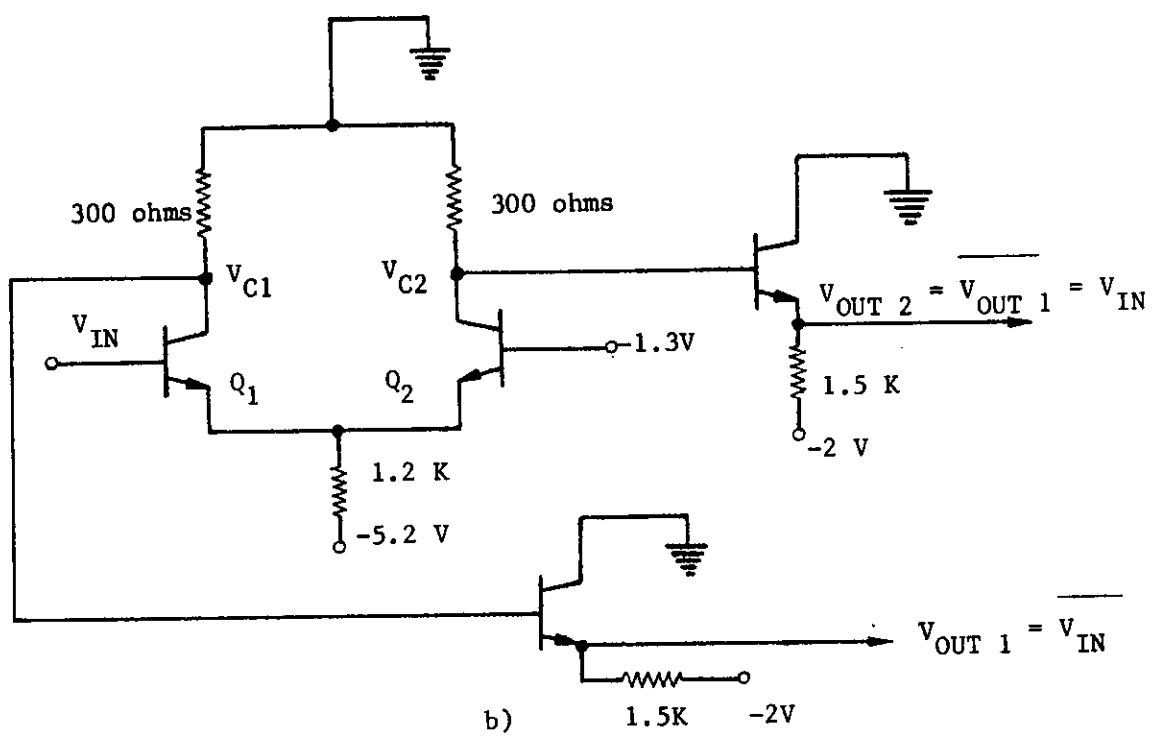
El circuito básico ECL de la fig. B.16b) puede usarse como un inversor si la salida se toma en V_{OUT1} . Este circuito puede expandirse a más de una entrada colocando otros transistores en paralelo a Q_1 como se muestra en la fig. B.17a). Aquí Q_1 o Q_3 pueden causar la conmutación de la corriente fuera de Q_2 resultando 2 salidas V_{OUT1} y V_{OUT2} que forman las operaciones lógicas NOR y OR respectivamente. El símbolo para la compuerta OR/NOR aparece en la fig. B.17b).

ESTADOS DE OPERACION



| V_{IN} | SALIDAS | |
|------------------|-----------------------------------|---------------|
| -1.7V (0 Log) | $V_{C1} = 0V$ $V_{C2} = -0.8V$ | Q_2 conduce |
| -0.8V (1 Log) | $V_{C1} = -0.9V$ $V_{C2} = 0V$ | Q_1 conduce |

a)



b)

Fig. B.16 a) CIRCUITO ECL BASICO, b) ECL CON UN SEGUIDOR EMISOR

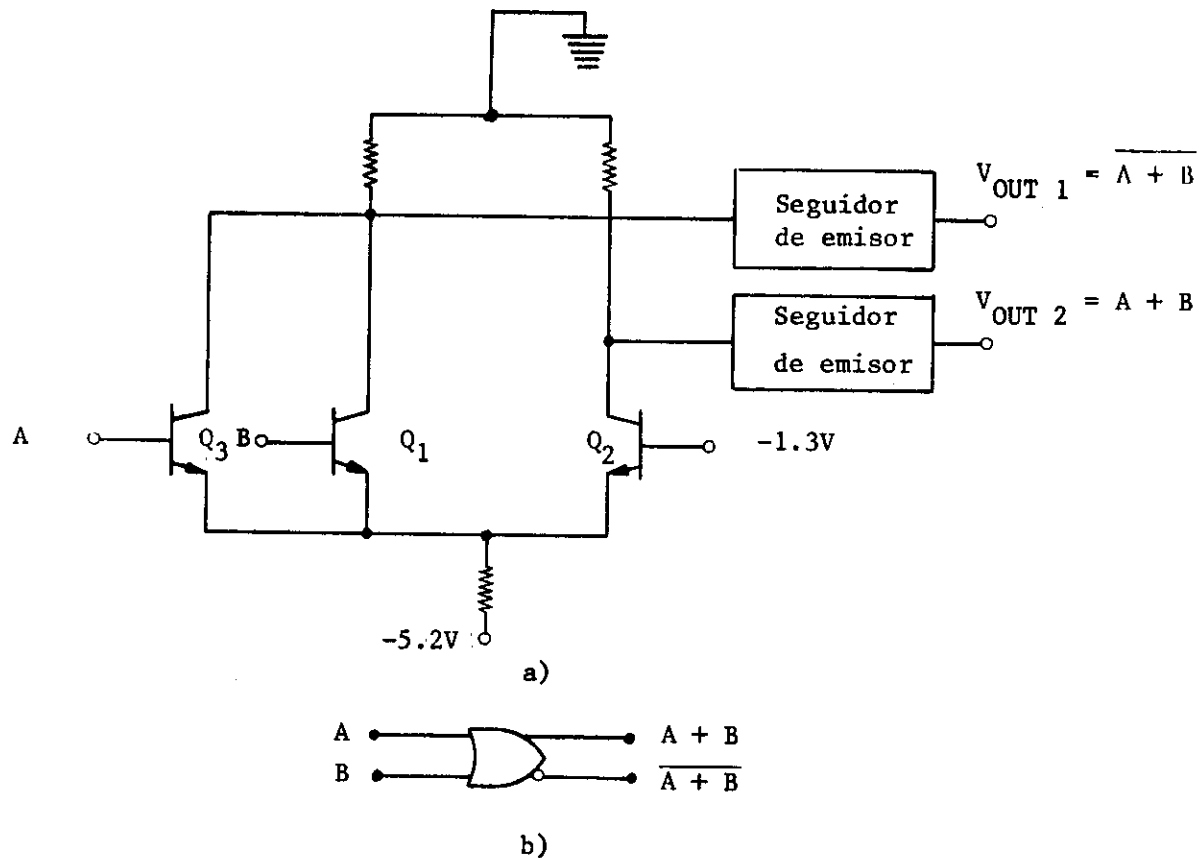


Fig. B.17 a) CIRCUITO PARA UN OR/NOR ECL, b) SIMBOLO

B.5 FAMILIA LOGICA RTL

La familia l3gica RESISTOR-TRANSISTOR (RTL) fu3 uno de los primeros - circuitos integrados usados ampliamente. Esto solamente consiste en resis- tencias y transistores de ah3 el nombre de RTL. Aunque 3sta ha sido des- plazada por otras familias.

La compuerta RTL que ejecuta la funci3n NOR, se muestra en la fig. - B.18.

El s3mbolo se interpreta como "si A es un nivel ALTO, o B es un ni- vel ALTO, o ambas son un nivel ALTO, la salida debe de ser nivel BAJO.

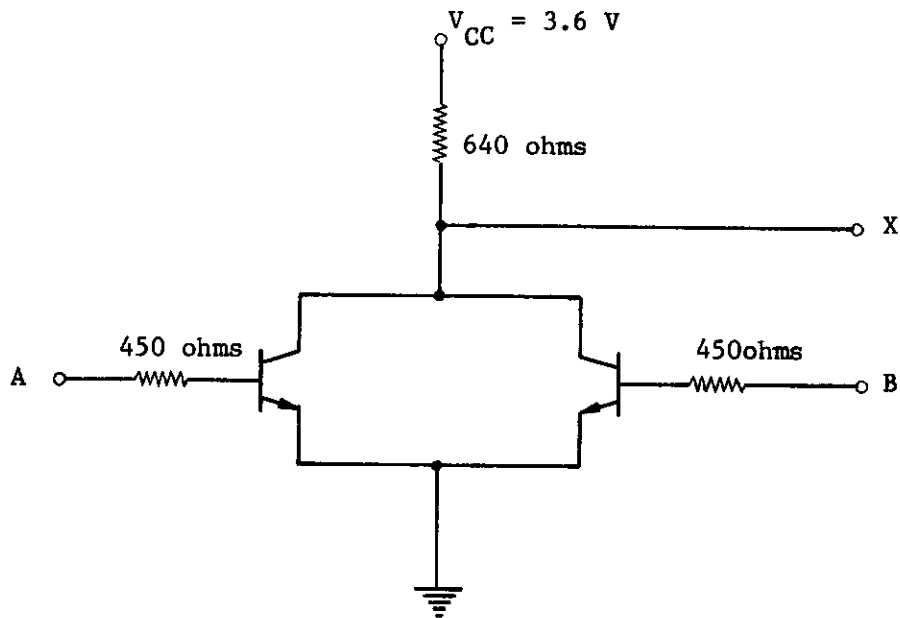


Fig. B.18 CIRCUITO BASICO PARA UNA COMPUERTA RTL

El circuito básico para la compuerta RTL se muestra en la fig. B.18.- Los valores de la resistencia ha sido estandarizado en todas las indus---trias. El voltaje especificado que alimenta a esta compuerta (V_{CC}) es de 3.6V. Este circuito es un ejemplo de "current sourcing" porque requiere - una corriente de entrada para encender otros transistores. Si uno u otro transistor está en "ON" o ambos transistores están en "ON", la salida de - voltaje debe de ser un nivel BAJO. El nivel BAJO se define como menor de 0.4 volts. Sin embargo, éste voltaje está usualmente alrededor de 0.2 --- volts, el cuál es el voltaje de saturación de un transistor en conducción. Si ambos transistores están en "OFF", la salida de voltaje debe ser un nivel ALTO y variará entre 1 y 3.6 volts aproximadamente dependiendo de la - cantidad de corriente. Un 1 lógico es definido como un nivel de voltaje - ALTO de aproximadamente un volt. Un 0 lógico se define como un nivel BAJO y está especificado menor que 0.4 volts.

La cantidad de corriente depende del número de entradas a las compuer- tas que se manejan por una salida en la compuerta; este número se llama -

"Fan-Out". Como el número de compuertas aumenta, la corriente en la salida aumenta, causando un decremento de nivel de voltaje de la salida. La fig. B.19 muestra las características de voltaje de transferencia cuando la compuerta maneja una entrada (fan-out = 1).

La fig. B.20 muestra las características del voltaje de transferencia cuando la compuerta maneja 5 entradas (fan-out = 5). Note que el nivel baja aproximadamente a 1 volts con un fan-out para las compuertas RTL es 5. El número de entradas puede ser incrementado agregando más transistores, incluyendo resistencias, en paralelo con los transistores existentes.

La fig. B.21 muestra el dual RTL 9915 con compuerta NOR de tres entradas. El valor de los componentes seleccionados están basados en el siguiente criterio: (1) disipación de potencia, (2) velocidad y (3) Factor de complejidad en la interconexión de los circuitos. En este orden se puede mantener un switcheo rápido, el valor de resistencias seleccionadas deben de ser bastante bajas. El valor bajo de la resistencia de base tiende a abastecer una trayectoria de alta conductancia para almacenar cargas en la región de base. En este orden la corriente se mantiene y la disipación de potencia es mínima, un bajo potencial de voltaje ($V_{CC} = 3.6 \text{ V.}$) es usado. Solamente compuertas NOR existen en la familia RTL.

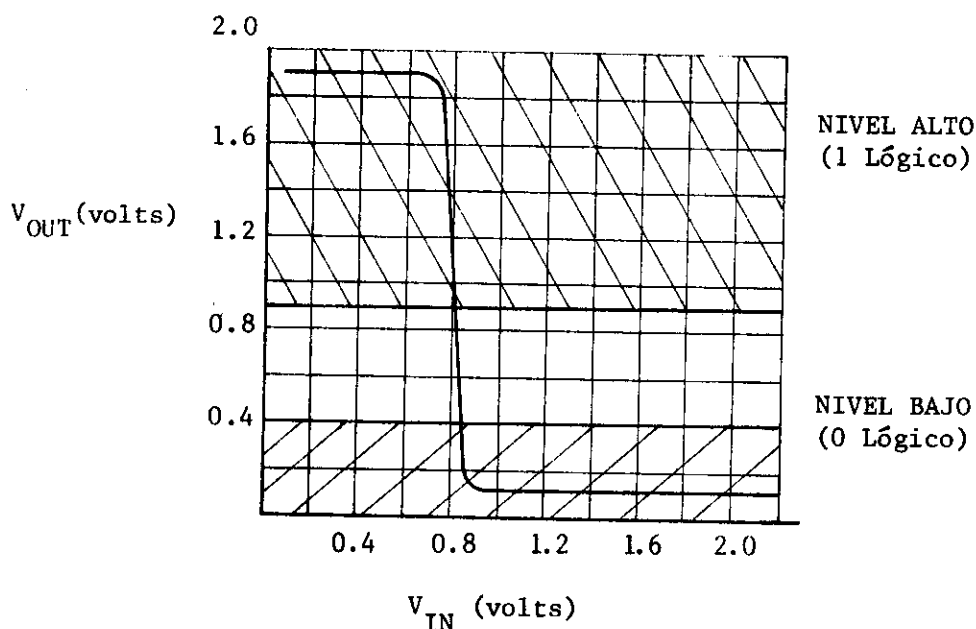


Fig. B.19 CARACTERISTICA DE TRANSFERENCIA RTL PARA UN FAN-OUT DE 1

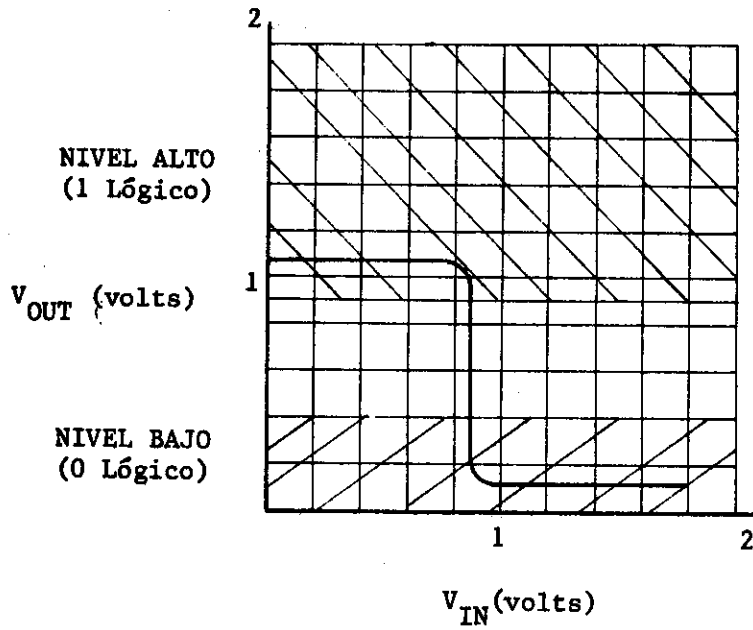


Fig. B.20 CARACTERISTICA DE TRANSFERENCIA RTL PARA UN FAN-OUT DE 5

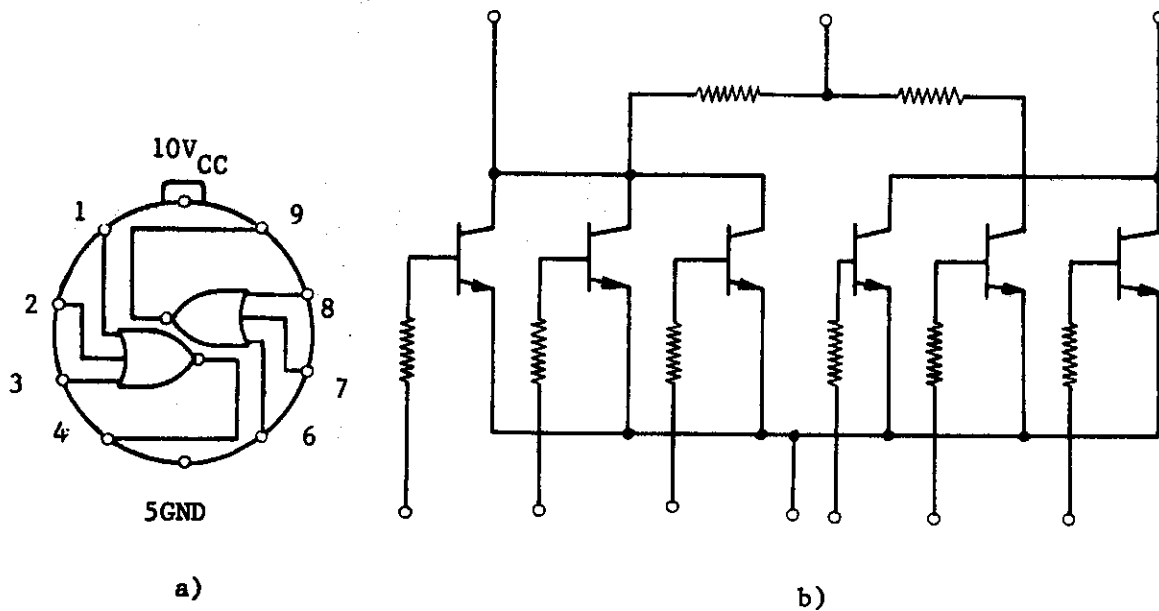


Fig. B.21 a) COMPUERTA NOR RTL, b) DIAGRAMA INTERNO

B.6 FAMILIA LOGICA DTL

Otra familia l3gica usada en circuitos integrados emplea diodos y -- transistores; se llama DIODO-TRANSISTOR-LOGIC (DTL). La compuerta DTL cumple la funci3n l3gica NAND, su equivalente D'Morgan y su tabla.

Un circuito b3sico para la compuerta NAND DTL aparece en la fig. B.22 Este circuito pertenece a la categori3a "current sinking" porque R_1 limita la corriente necesaria en la base para encender al transistor. Si las -- entradas A y B se conectan a un nivel ALTO, el D_1 y D_2 se les debe de polarizar inversamente durante un per3odo corto y el D_3 y D_4 polarizados directamente, Q_1 est3 encendido, resultando a la salida un nivel BAJO. Esta -- condici3n describe la funci3n de la compuerta NAND "Si A y B son nivel ALTO entonces la salida debe ser nivel BAJO.

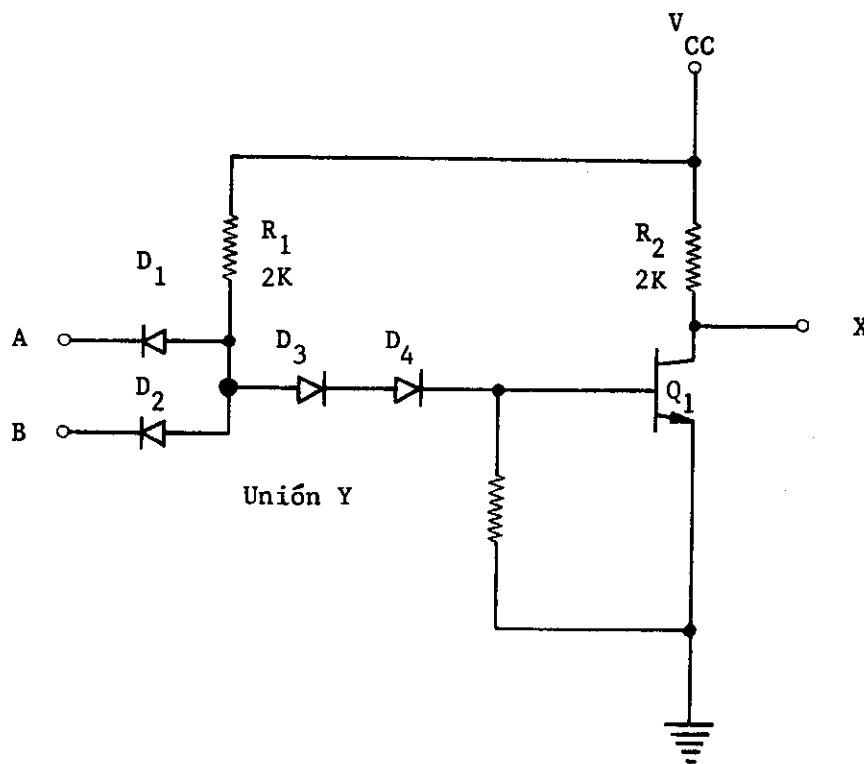


Fig. B.22 CIRCUITO BASICO PARA UNA COMPUERTA NAND DTL

El equivalente D'Morgan, función NOR ocurre cuando A es nivel BAJO, o B es nivel BAJO o ambos son niveles BAJOS. Refiriéndonos a la figura anterior, note que cuando A o B, son BAJOS (0.4 volts o menos), el potencial de la unión "Y" debe de ser aproximadamente 1 volt. Este potencial es la suma del voltaje de entrada y la caída en la unión (0.6 V) a través de D_1 y D_2 . Sin embargo un potencial en la unión "Y" con un mínimo de 1.8 volts es necesario para mantener el Q_1 en conducción (0.6V para cada diodo D_3 y D_4 y 0.6 V para la unión base-emisor de Q_1). Por lo tanto, uno debe de apagarse cuando A o B o ambas son niveles BAJOS y la función NOR se realiza "si A o B son niveles BAJOS, entonces la salida debe de ser nivel ALTO.

El circuito en la fig. B.23 es el diagrama de la compuerta NAND, DTL, el cual se ha estandarizado por muchos fabricantes. El rasgo más sobresaliente de este circuito es que uno de los diodos (D_3) ha sido reemplazado por un transistor cuyo colector se conecta entre R_1 y R_4 . Esto incrementa la ganancia de corriente del circuito, y por consiguiente la impedancia de entrada. En la fig. B.24 aparecen las curvas características del DTL básico.

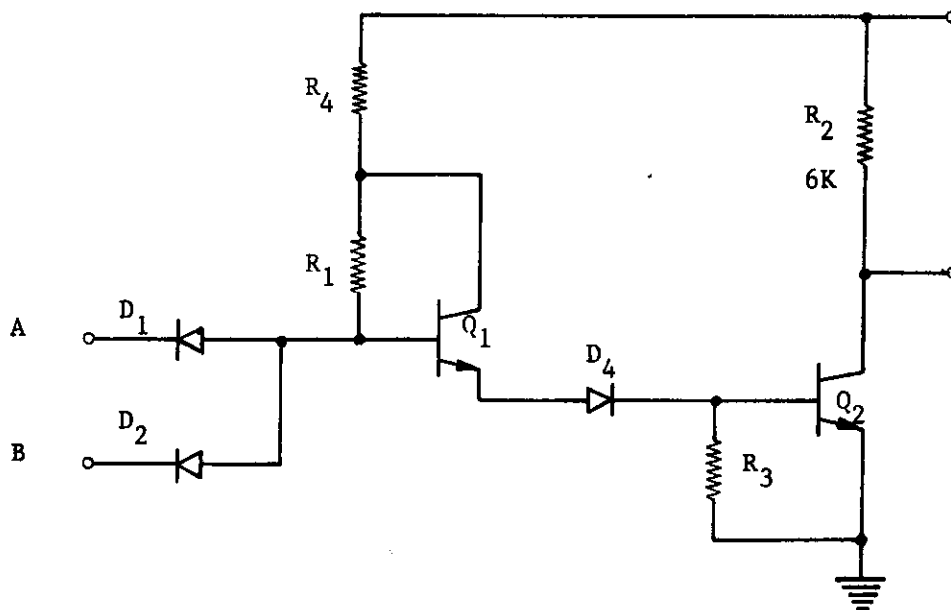


Fig. B.23 COMPUERTA NAND DTL

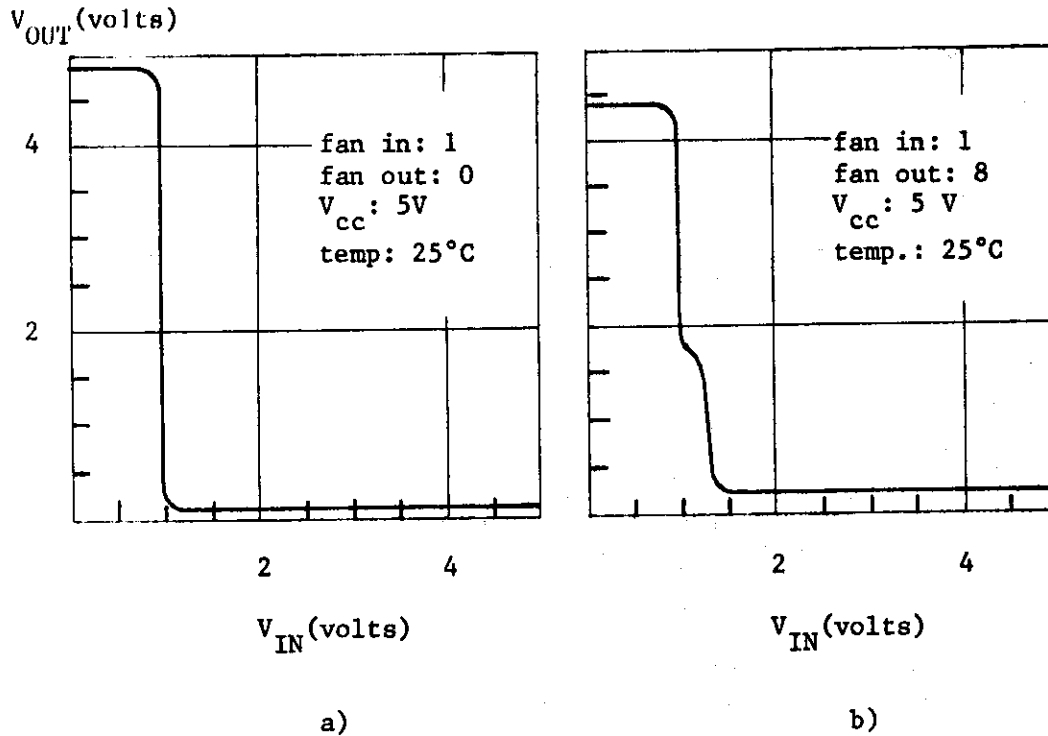


Fig. B.24 CURVAS CARACTERISTICAS DE UNA COMPUERTA DTL. a) CON FAN-OUT DE 0. b) CON FAN-OUT DE 8.

B.7 FAMILIA LOGICA HTL

La familia lógica TTL, que es la más ampliamente utilizada en la actualidad, presenta el inconveniente de que tiene una pequeña inmunidad al ruido.

Para superar este problema se puede usar la lógica HTL que es una tecnología bipolar de alta inmunidad al ruido. Esta familia es pues adecuada para aplicaciones industriales sencillas donde las condiciones de ruido sean elevadas.

En la Fig. B.25 se muestra una compuerta NAND HTL de 2 entradas. La operación es básicamente igual a la familia lógica DTL con la excepción de que el diodo D_4 se sustituye por el D_2 que es un diodo Zener. Con un V_Z aproximadamente de 5 volts.

Una vez saturado el transistor Q_1 si un ruido llegara a presentarse tendría que ser superior al V_Z para poder cambiar el estado de la salida.

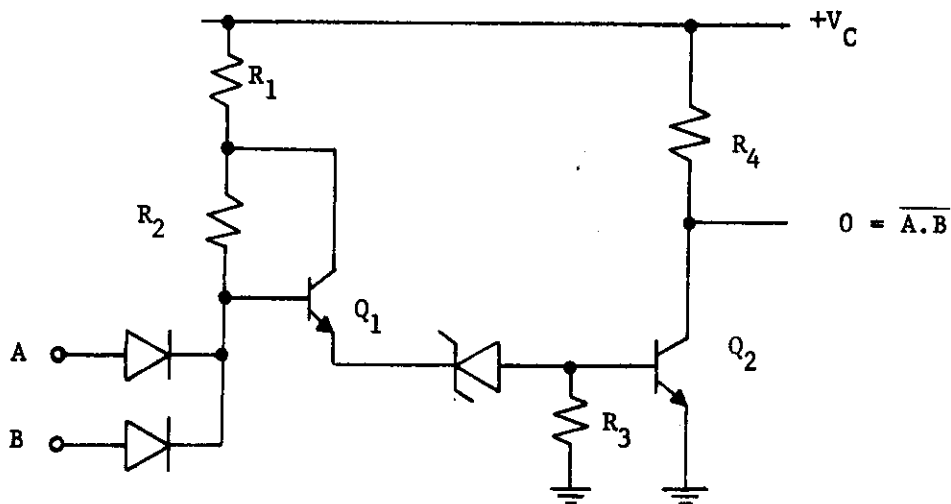


Fig. B.25 COMPUERTA NAND DE 2 ENTRADAS, HTL

B.7.1 CARACTERISTICAS DE LA FAMILIA LOGICA HTL

La inmunidad al ruido anda alrededor de 5 volts, el voltaje de alimentación es de 15 volts de C.D. La frecuencia de operación es de hasta 500 Hz.

En la actualidad presenta una desventaja el uso de la tecnología - HTL, no existe en el mercado una amplia variedad de circuitos lógicos de SSI y mucho menos de MSI, por lo tanto ha sido desplazada por la familia lógica CMOS.

B.8 COMPARACION DE LAS FAMILIAS LOGICAS

La tabla de la fig. B.26 es una comparación de las características -- principales de las familias lógicas que han sido discutidas en éste apéndice. Los números usados en ésta tabla son valores típicos o promedios para la compuerta lógica de cada familia y no debería usarse como una indicación exacta de las ventajas relativas de cada familia.

| | TTL | ECL | MOS | CMOS |
|---------------------------|--|--------------------------------------|--|------------------------------------|
| Voltaje de la Fuente | $V_{CC} = 5$ VOLTS | $V_{EE} = -5.2$ VOLTS | $V_{DD} = -12$ VOLTS -2/V. para P-MOS | $V_{DD} = 3-18$ VOLTS |
| Niveles Lógicos Nominales | "0" = 0 volts "1" = 5 volts | "0" = -1.7 volts "1" = -0.8 volts | "0" = 0 volts "1" = V_{DD} | "0" = 0 volts "1" = V_{DD} |
| P_D promedio | Serie 7400, 10mW Serie 74100, 1mW Serie 74S00, 23mW | 25 mWATTS | 2mWATTS | 4 nW usando $V_{DD} = 5$ volts |
| tpd promedio | Serie 7400, 9nseg Serie 74100, 33nseg. Serie 74S00, 3nseg. | 2 n seg. | 100 n seg. | 30 n seg. |
| Fan-Out | Típico 10 a veces 20 | 25 | 50 | 50 100 |
| Márgenes de Ruido | En el peor caso 400 mV | 250 mVOLTS | 2 Volts | HASTA UN 45% DE V_{DD} |
| Densidad de empaque | Baja | Baja | La mayor-la más apro- piada para LSI. | Alta pero no tan alta como MOS. |

Fig. B.26 TABLA COMPARATIVA DE FAMILIAS LOGICAS

B I B L I O G R A F I A

CAPITULO 2 - SISTEMAS NUMERICOS

INTEGRATED CIRCUITS IN DIGITAL ELECTRONICS

ARPAD, BARNA - JOHN WILEY

DIGITAL COMPUTER FUNDAMENTALS

BRATEE, THOMAS C. - MC. GRAW - HILL

DIGITAL COMPUTER DESIGN FUNDAMENTALS

CHU, YA OHAN - MC. GRAW - HILL

DIGITAL CIRCUITS AND DEVICES

K OHONEN, T. - PRENTICE-HALL

DIGITAL COMPUTERS CIRCUITS AND CONCEPTS

DEEM, MUCHOW - PRENTICE-HALL

CAPITULO 3 - ALGEBRA BOOLEANA

CAPITULO 5 - MINIMIZACION DE FUNCIONES BOOLEANAS

CAPITULO 6 - DISEÑO COMBINACIONAL

INTEGRATED CIRCUITS IN DIGITAL ELECTRONICS

ARPAD, BARNA - JOHN WILEY

FUNDAMENTALS OF DIGITAL SYSTEMS DESIGN

RHYNE - PRENTICE HALL

SWITCHING AND FINITE AUTOMATA THEORY

KOHAVI, Z - MC GRAW - HILL

AN INTRODUCTION TO SWITCHING SYSTEM DESIGN

MARCOVITZ - JOHN WILEY

INTRODUCTION TO THE THEORY OF SWITCHING CIRCUITS

MC. CLUSKEY MC GRAW - HILL

CAPITULO 4 - CODIGOS Y REPRESENTACION DE INFORMACION

ERROR CORRECTING CODES

PETERSON - JOHN WILEY

DIGITAL NET WORKS AND COMPUTER SYSTEMS

BOOTH, TAYLOR L. JOHN WILEY

CAPITULO 7 - FLIP-FLOPS

CAPITULO 8 - DISEÑO SECUENCIAL

SWITCHING CIRCUITS THEORY AND LOGIC DESIGN
TORNG - ADDISON - WESLEY

INTEGRATED CIRCUITS IN DIGITAL ELECTRONICS
ARPAD, BARNA - JOHN WILEY

LOGIC DESIGN OF DIGITAL SYSTEMS
DIETMEYER - ALLYN AND BACON

INTRODUCTION TO SWITCHING THEORY AND LOGICAL DESIGN
HILL, PETERSON - JOHN WILEY

APENDICE A y B

DESIGN WITH TTL INTEGRATED CIRCUITS
MORRIS, MILLER - MC GRAW - HILL

INTRODUCTION TO SWITCHING THEORY AND LOGICAL DESIGN
HILL, FREDERICK, GERALD - JOHN WILEY

ELECTRONIC CIRCUITS: DISCRETE AND INTEGRATED
SCHILLING AND BELOVE - MC GRAW - HILL

INDICE ALFABETICO

| | |
|--|------------|
| ACARREO | 15 |
| ALGEBRA BOOLEANA | 46 |
| ALFANUMERICO, CODIGO | 93 |
| ANALOGICO | 6 |
| ANALOGICO A DIGITAL, CONVERSION | 10 |
| AND | 39 |
| AND, OR, NOT | 39, 41, 42 |
| A.O.N. FORMA | 61 |
| ARITMETICA BINARIA, DECIMAL, HEXADECIMAL Y OCTAL | 30 |
| ASCII, CODIGO | 94 |
| ASINCRONO, SISTEMA SECUENCIAL | 244 |
| ASOCIATIVA, LEY | 59 |
| | |
| BASE | 249 |
| BAUDOT, CODIGO | 96 |
| BCD, CODIGO | 88 |
| BIESTABLE, MULTIVIBRADOR | 189 |
| BINARIA, ARITMETICA | 31 |
| RESTA, 1 y 2 COMPLEMENTO | 34 |
| SUMA | 31 |
| BINARIO A DECIMAL, CONVERSION | 21 |
| BINARIO, CONTADOR | 218 |
| BINARIO, SISTEMA NUMERICO | 20 |
| BIPOLAR, TRANSISTOR | 249 |
| BIT | 20 |
| BOOLEANA, ALGEBRA | 46 |
| BYTE | 83 |
| | |
| CANONICA, FORMA | 72 |
| CARACTER | 83 |

| | |
|---------------------------|-----|
| CARACTERISTICAS, TABLAS | 178 |
| FF D | 181 |
| FF JK | 178 |
| FF RS | 179 |
| FF T | 180 |
| CK RELOJ | 170 |
| CLEAR | 182 |
| CMOS | 288 |
| CODIGOS | 82 |
| ALFANUMERICO | 93 |
| ASCII | 94 |
| BAUDOT | 96 |
| BCD | 88 |
| BINARIO | 82 |
| 2-4-2-1 | 89 |
| EBCDIC | 93 |
| EXCESO 3 | 89 |
| GRAY | 89 |
| 8-4-2-1 | 88 |
| PESADOS | 84 |
| COLECTOR | 249 |
| COLECTOR ABIERTO | 272 |
| COMBINACIONALES, SISTEMAS | 132 |
| APLICACIONES | 135 |
| DISEÑO | 133 |
| COMPARADOR | 152 |
| COMPLEMENTO, 1 Y 2 | 35 |
| COMPUERTAS | |
| AND | 47 |
| COINCIDENCE | 54 |
| EXOR | 51 |
| NAND | 52 |
| NOR | 53 |
| NOT | 50 |
| OR | 49 |

| | |
|-------------------------------|-----|
| CONMUTATIVA, LEY | 59 |
| CONTADORES | 217 |
| ASCENDENTE/DESCENDENTE | 225 |
| BCD | 221 |
| BINARIO | 218 |
| CON REGISTROS DE CORRIMIENTOS | 242 |
| DISEÑO | 217 |
| CONVERSION | |
| BINARIO/GRAY, GRAY/BINARIO | 91 |
| BINARIO - DECIMAL | 21 |
| BINARIO - OCTAL - HEXADECIMAL | 27 |
| COSTO UNITARIO | 107 |
| CHIP | 66 |
| DATO | 10 |
| DECIMAL, ARITMETICA | 30 |
| DECIMAL, SISTEMA NUMERICO | 15 |
| DECODIFICADORES | 148 |
| BCD A 7 SEGMENTOS | 145 |
| N a 2 ^N | 148 |
| DE MORGAN, TEOREMA | 60 |
| DIAGRAMA DE TIEMPO | 163 |
| DIAGRAMA DE TRANSICION | 198 |
| DIGITAL, CONCEPTO | 9 |
| DIGITALIZACION | 9 |
| DIGITO | 15 |
| DIODO | 246 |
| DIODO SCHOTTKY | 247 |
| DISTRIBUTIVA, LEY | 59 |
| DIVISION | 38 |
| DON'T CARE | 141 |
| DRENAJE | 253 |

| | |
|-----------------------|----------|
| DTL | 301 |
| ECL | 294 |
| EMISOR | 249 |
| EMITTER COUPLED LOGIC | 294 |
| ENCODIFICADOR | 89 |
| ENTRADAS NO USADAS | 271, 293 |
| EQUIVALENTE, ESTADO | 215 |
| ESTADO, TABLA DE | 197 |
| EXCESO 3 | 89 |
| EXCITACION, TABLA DE | 177 |
| FF D | 181 |
| FF JK | 178 |
| FF RS | 179 |
| FF T | 180 |
| EXOR | 51 |
| FAMILIAR LOGICAS | 283 |
| FAN OUT | 259 |
| FET | 252 |
| FLIP - FLOP | |
| D | 180 |
| JK | 172 |
| MAESTRO/ESCLAVO | 175 |
| RS | 169, 171 |
| T | 180 |
| GERMANIO | 248 |
| GRAY, CODIGO | 89 |
| HEXADECIMAL | 25 |
| IC | 66 |
| IMPEDANCIA | 270 |
| IMPLEMENTACION | 133 |

| | |
|-------------------------------------|---------|
| INVERSOR | 50, 285 |
| KARNAUGH, MAPA DE | 110 |
| LEY | |
| ASOCIATIVA | 59 |
| CONMUTATIVA | 59 |
| DISTRIBUTIVA | 59 |
| LOGICA | |
| NEGATIVA | 279 |
| POSITIVA | 279 |
| MAESTRO-ESCLAVO, FF | 175 |
| MAXITERMINO | 73 |
| MEMORIA | 162 |
| MINIMIZACION | 106 |
| MINITERMINO | 67 |
| MOS | 252 |
| MSB | 20 |
| MSD | 17 |
| MSI | 246 |
| MULTIEMISOR, TRANSISTOR CON ENTRADA | 265 |
| MULTIPLE, FUNCION | 158 |
| MULTIPLEXER | 135 |
| MULTIPLICACION | 38 |
| MULTIVIBRADORES | 189 |
| ASTABLE | 193 |
| MONOESTABLE | 189 |
| ONE SHOT | 189 |
| REDISPARABLE | 191 |
| NAND | 52 |
| NEGACION | 50 |
| NEGATIVA, LOGICA | 279 |

| | |
|------------------------------|-----|
| NOR | 53 |
| NOT | 50 |
| NOTACION POSICIONAL | 15 |
| NPN | 249 |
| NUMERO | 17 |
| NUMERO NEGATIVO | 99 |
| OCTAL | 24 |
| ONE SHOT | 189 |
| OPEN COLLECTOR | 272 |
| PARIDAD | 98 |
| PNP | 252 |
| POSICION | 16 |
| POSITIVA, LOGICA | 279 |
| PRESET | 182 |
| PRODUCTO DE SUMATORIAS | 73 |
| PROPAGACION, TIEMPO, RETARDO | 270 |
| REDUCCION DE ESTADOS | 215 |
| REDUNDANCIA | 215 |
| REGISTROS | |
| PI - PO | 186 |
| PI - SO | 186 |
| SI - PO | 185 |
| SI - SO | 185 |
| RESET | 182 |
| RESIDUOS, METODO | 22 |
| RESOLUCION | 8 |
| ROMANOS, NUMEROS | 14 |
| RTL | 297 |
| SATURACION, REGION | 250 |
| SCHMITT TRIGGER | 274 |

| | |
|--------------------------------|--------------------|
| SCHOTTKY, DIODO | 247 |
| SECUENCIAL, CIRCUITO, SISTEMAS | 196 |
| SECUENCIAL, SISTEMA | 162 |
| SEMICONDUCTOR | 246 |
| SIGNO | 99 |
| SIMBOLOS | 49, 50, 51, 52, 55 |
| SISTEMA ANALOGICO | 6 |
| SISTEMA DIGITAL | 9 |
| SISTEMA NUMERICO | |
| BINARIO | 20 |
| DECIMAL | 15 |
| HEXADECIMAL | 25 |
| NOTACION POSICIONAL | 15 |
| OCTAL | 24 |
| SSI | 246 |
| SUMA | 30, 31, 32, 33 |
| SUMATORIA DE PRODUCTOS | 72 |
| | |
| TABLAS | |
| CARACTERISTICAS | 177 |
| DE ESTADO | 177 |
| DE EXCITACION | 177 |
| DE VERDAD | 47 |
| THRESHOLD, VOLTAJE | 274 |
| TIEMPO, DIAGRAMAS DE | 163 |
| TOTEM POLE | 261 |
| TRANSICION, DIAGRAMAS DE | 198 |
| TRANSISTOR BIPOLAR | 249 |
| TRI - STATE, LOGICA DE | 276 |
| TTL | 261 |
| | |
| WIRE AND | 273 |