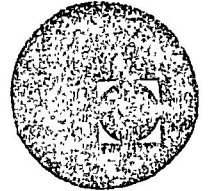




centro de educación continua  
división de estudios superiores  
facultad de ingeniería, unam



A LOS ASISTENTES A LOS CURSOS DEL CENTRO DE EDUCACION  
CONTINUA

Las autoridades de la Facultad de Ingeniería, por conducto del Jefe del Centro de Educación Continua, Dr. Pedro Martínez Pereda, otorgan una constancia de asistencia a quienes cumplan con los requisitos establecidos para cada curso. Las personas que deseen que aparezca su título profesional precediendo a su nombre en la constancia, deberán entregar copia del mismo o de su cédula a más tardar el SEGUNDO DIA de clases, en las oficinas del Centro con la señorita Barraza, encargada de inscripciones.

El control de asistencia se llevará a cabo a través de la persona encargada de entregar las notas del curso. Las inasistencias serán computadas por las autoridades del Centro, con el fin de entregarle constancia solamente a los alumnos que tengan un mínimo del 80% de asistencia.

Se recomienda a los asistentes participar activamente con sus ideas y experiencias, pues los cursos que ofrece el Centro están planeados para que los profesores expongan una tesis, pero sobre todo, para que coordinen las opiniones de todos los interesados constituyendo verdaderos seminarios.

Es muy importante que todos los asistentes llenen y entreguen su hoja de inscripción al inicio del curso. Las personas comisionadas por alguna institución deberán pasar a inscribirse en las oficinas del Centro en la misma forma que los demás asistentes entregando el oficio respectivo.

Con objeto de mejorar los servicios que el Centro de Educación Continua ofrece, al final del curso se hará una evaluación a través de un cuestionario diseñado para emitir juicios anónimos por parte de los asistentes.





# APLICACIONES DE CIRCUITOS INTEGRADOS AL DISEÑO DIGITAL

(del 12 al 23 de junio de 1978)

| FECHA       | HORARIO   | TEMA                              | PROFESOR                        |
|-------------|-----------|-----------------------------------|---------------------------------|
| 12 de junio | 17 a 19 h | INTRODUCCION                      | M. EN C. ALEJANDRO GUARDA AURAS |
|             | 19 a 21 h | FUNDAMENTOS DEL DISEÑO LOGICO     |                                 |
| 13 de junio | 17 a 19 h | FUNDAMENTOS DEL DISEÑO LOGICO     | M. EN C. ALEJANDRO GUARDA AURAS |
|             | 19 a 21 h | CIRCUITOS COMBINACIONALES         | M. EN C. ALEJANDRO GUARDA AURAS |
| 14 de junio | 17 a 19 h | CIRCUITOS COMBINACIONALES         | M. EN C. ALEJANDRO GUARDA AURAS |
|             | 19 a 21 h | FLIP - FLOPS                      | ING. ARTURO GONZALEZ HERMOSILLO |
| 15 de junio | 17 a 21 h | FLIP - FLOPS                      | ING. ARTURO GONZALEZ HERMOSILLO |
| 16 de junio | 17 a 21 h | CIRCUITOS SECUENCIALES SINCRONOS  | ING. MARIO RODRIGUEZ MANZANERA  |
| 19 de junio | 17 a 21 h | CIRCUITOS SECUENCIALES ASINCRONOS | ING. MARIO RODRIGUEZ MANZANERA  |
| 20 de junio | 17 a 21 h | FAMILIAS LOGICAS E INTERFACES     | DR. JOSE F. ALBARRAN NUÑEZ      |
| 21 de junio | 17 a 21 h | MEMORIAS RAM, ROM Y SERIE         | ING. CESAR CHAVEZ ZAPATA        |
| 22 de junio | 17 a 21 h | MEMORIAS RAM, ROM Y SERIE         | ING. CESAR CHAVEZ ZAPATA        |
| 23 de junio | 17 a 21 h | PROBLEMAS: CARRERAS E INTERFACES  | ING. CESAR CHAVEZ ZAPATA        |

pmc.

DIRECTORIO DE PROFESORES

APLICACIONES DE CIRCUITOS INTEGRADOS AL DISEÑO DIGITAL

DR. JOSE F. ALBARRAN NUÑEZ  
INSTITUTO DE INVESTIGACIONES EN  
MATEMATICAS APLICADAS Y EN SISTEMAS  
U. N. A. M.  
CIUDAD UNIVERSITARIA  
MEXICO 20, D.F.  
TEL : 550.52.15 ext. 4573

M. EN I. CESAR CHAVEZ ZAPATA  
PROFESOR DE DISEÑO DE CIRCUITOS LOGICOS  
SECCION DE ELECTRONICA  
D.E.S.F.I., UNAM  
CIUDAD UNIVERSITARIA  
MEXICO 20, D.F.  
TEL: 550.52.15 ext. 4484

ING. ARTURO GONZALEZ HERMOSILLO  
INVESTIGADOR ASOCIADO  
INSTITUTO DE INVESTIGACIONES EN  
MATEMATICAS APLICADAS Y EN SISTEMAS  
U. N. A. M.  
CIUDAD UNIVERSITARIA  
MEXICO 20, D.F.  
TEL: 550.52.15 ext. 4573 ó 548.33.60

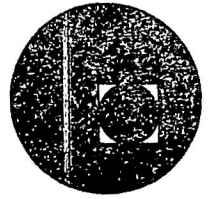
M. EN C. ALEJANDRO GUARDA AURAS  
INVESTIGADOR  
DEPARTAMENTO DE DISEÑO DE SISTEMAS DIGITALES  
I. I. M. A. S., UNAM  
CIUDAD UNIVERSITARIA  
MEXICO 20, D.F.  
TEL: 550.52.15 ext. 4573

ING. MARIO RODRIGUEZ MANZANERA  
INVESTIGADOR  
I. I. M. A. S., UNAM  
CIUDAD UNIVERSITARIA  
MEXICO 20, D.F.  
TEL: 550.52.15 ext. 4581 ó 4582

'pmc.



centro de educación continua  
división de estudios superiores  
facultad de ingeniería, unam



APLICACIONES DE CIRCUITOS INTEGRADOS AL  
DISEÑO DIGITAL

CAPITULO 1: INTRODUCCION

CAPITULO 2: CONCEPTOS BASICOS

CAPITULO 3: CIRCUITOS COMBINACIONALES

M. EN C. ALEJANDRO GUARDA AURAS

JUNIO 1978

# INDICE

## Capítulo 1: Introducción

- 1.1.- Técnicas Digitales vs. Técnicas Analógicas
- 1.2.- Exactitud e Inmunidad al Ruido
- 1.3.- Estandarización
- 1.4.- Algunas consideraciones sobre diseño de sistemas digitales.

## Capítulo 2: Conceptos Básicos de Sistemas Numéricos y Diseño Lógico

- 2.1.- Notación Posicional
- 2.2.- El Sistema Binario
- 2.3.- Sistemas Octal y Hexadecimal
- 2.4.- Representación de números negativos
- 2.5.- Algunos Códigos más usados
- 2.6.- Algebra de Boole
- 2.7.- Funciones booleanas
- 2.8.- Formas Algebraicas de Funciones Combinacionales
- 2.9.- Mapas de Karnough.

## Capítulo 3: Circuitos Combinacionales

- 3.1.- Compuertas
- 3.2.- Análisis de Circuitos Combinacionales
- 3.3.- Propiedades de las funciones NAND y NOR
- 3.4.- Análisis de Circuitos NAND y NOR
- 3.5.- Diseño con compuertas NAND y NOR
- 3.6.- Diseño con Multiplexores

## CAPITULO 1: INTRODUCCION.

### 1.1.- Técnicas Digitales versus Técnicas Analógicas:

La Electrónica puede concebirse como aquella área de la técnica que trata con el procesamiento de señales. Hasta aproximadamente diez años atrás, dicho procesamiento se hacía fundamentalmente mediante técnicas analógicas en las que "codificaba" una señal en función de su amplitud, frecuencia o fase.

Las técnicas digitales, hasta la época mencionada, se consideraban demasiado costosas, aún cuando se reconocían sus ventajas en que el tipo de codificación que empleaba, mediante niveles de voltaje, era mucho más versátil que el analógico.

Desde hace una década, con la producción de los primeros circuitos integrados digitales (compuertas RTL) se observó que los problemas de costo, tamaño, etc., podían ser superados y desde entonces, las técnicas digitales han evolucionado al punto que no hay lugar del dominio analógico donde no haya aparecido un reemplazo digital. Vemos así que en relojes [1], T.V. [2], teléfonos [3], controles industriales [4], artículos domésticos [5], etc., cada día aparecen nuevos dispositivos, sin mencionar el campo de las computadoras que le es propio.

Indudablemente, que en un universo analógico, las técnicas analógicas no van a desaparecer. Sin embargo, se observa claramente que su función se está concentrando en dispositivos para la adecuación de señales y la adquisición de datos. Es decir, hacia transformar señales analógicas en digitales para posteriormente ser procesadas digitalmente.

¿A qué se debe esta tendencia actual? A nuestro juicio, hay tres factores principales.

Exactitud  
Inmunidad al ruido  
Estandarización.

A continuación examinaremos estas cualidades de los sistemas digitales.

### 1.2.- Exactitud e Inmunidad al ruido:

Habíamos dicho que en un sistema analógico la codificación de una señal se establece a través de variaciones continuas de la amplitud, frecuencia o fase de la señal. Por lo tanto, la exactitud en un sistema analógico estará limitada por: la mínima señal detectable -nivel de ruido-, la máxima señal transferible -saturación del sistema- y la distorsión propia del sistema. En condiciones muy ideales, un amplificador de bajo ruido y con un rango dinámico de 120 db, puede procesar una señal con una exactitud de una parte por millón, lo que corresponde aproximadamente a  $2^{20}$ .

Un sistema digital, en cambio, trabaja con señales binarias (1 ó 0), las que son empleadas para codificar la información en forma numérica. Esto hace que la exactitud con que se represente la información estará limitada exclusiva-



mente por el número de dígitos binarios, o bits, empleados; es decir por la longitud de palabra\*. Como ejemplo podemos considerar la computadora CDC 6400/6600 que tiene una palabra de 64 bits y que por lo tanto, permite representar números menores o iguales a  $(2^{64}-1)$  es decir, del orden de  $1.8 \times 10^{19}$ , lo que significa contar con una exactitud de una parte en  $10^{19}$ ; es decir,  $10^{13}$  veces más exacto que el amplificador citado.

De lo anterior, podemos decir que la exactitud en un sistema digital está limitada solamente por cuestiones económicas (que también son muy válidas), ya que eventualmente podríamos trabajar con longitudes de palabras aún mayores. De hecho, si en la CDC 6400/6600 programáramos empleando doble precisión, contaríamos con una exactitud de una parte en  $10^{38}$ , lo cual, aún para trabajos astronómicos resulta suficiente. En lo que respecta al ruido, es sabido como éste afecta a una señal analógica y que al ser amplificada, también estamos amplificando ruido.

En sistemas digitales, la especificación de los niveles de 0 y 1 permite ciertas fluctuaciones en torno al nivel de operación. Esto equivale a trabajar con "bandas" de señal, como se muestra en la figura 1-1.

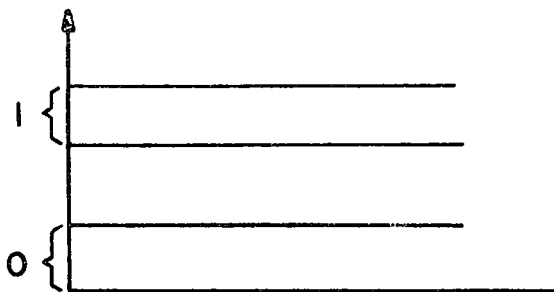


FIGURA 1-1

Esto hace que los sistemas digitales tengan cierta "tolerancia" al ruido. Es decir, si la señal llega distorsionada, pero dentro de los niveles establecidos, será reconocida como un 1 o un 0 según el caso.

Esto resulta, evidentemente, la ventaja más clara de un sistema digital con respecto a uno analógico, desde el punto de vista de operación.

### 1.3.- Estandarización:

Una característica de las técnicas digitales es que es posible (aunque no recomendable) diseñar casi cualquier sistema con el mismo tipo de compuerta y el mismo tipo de flip-flop. De hecho, en una época, esa fue la tendencia. Esto permitió la estandarización del tipo de componente empleado y como consecuencia, la producción masiva del mismo, con el consiguiente abatimiento de costos.

\* La longitud de palabra, es el nivel primario en que se agrupan los dígitos binarios (bits) en una computadora.

Como resultado de ésto, se tiene que por ejemplo, hay un sólo tipo de compuerta NAND, y la única variación está en el número de entradas (2,3,4 y 8) y en el tipo de salida (totem pole y colector abierto).

Si como comparación consideramos el amplificador operacional, veremos que existe una gama enorme de variaciones sobre el tema: bajo ruido, entrada con FET's, instrumentación.

Esto es de esperarse puesto que las señales de entrada a los sistemas digitales consisten de pulsos estándar, y en cambio el tipo de señales que se procesan analógicamente, cubre un espectro sumamente amplio. Por esta razón, en el primer caso se puede tener un mismo tipo de componente operando en sistemas cuyas funciones sean completamente diferentes, lo cual no es el caso de los sistemas analógicos.

Un ejemplo muy claro de los efectos de la estandarización y que trasciende al ámbito de la electrónica es el caso de los relojes. El reloj analógico (mecánico) está compuesto por una gran cantidad de piezas y cada una, aunque en principio sea la misma que se empleó en otro reloj, es susceptible a variaciones: por ejemplo el número de dientes en un engrane. El reloj electrónico en cambio está hecho de un circuito integrado estándar, del que se fabrican miles con el mismo proceso. Esto se refleja en los costos actuales de un reloj digital en los que el costo del circuito integrado es una fracción mínima (menos del 10%).

Otro ejemplo de estandarización es el caso de los microprocesadores. Hasta su introducción al mercado, la mayoría de los componentes más complejos, de uso específico y de fabricación masiva, eran ordenados por un cliente a una fábrica de semiconductores para su fabricación. Es lo que se llama "custom design", en la que la fábrica de circuitos desempeña la función de "sastre", diseñando un circuito adecuado a las necesidades precisas del cliente.

Con los microprocesadores se ha puesto en el mercado un dispositivo reconfigurable en su función de acuerdo a las necesidades del usuario y dicha reconfiguración es realizada por el mismo usuario. Es decir, con un circuito estándar, producido masivamente, se pueden realizar funciones completamente diferentes como las de una calculadora hasta el control de una central privada de teléfonos.

A todo lo anterior se suma la enorme ventaja de disminuir los costos de mantenimiento y servicio al requerirse un stock más reducido de componentes.

#### 1.4.- Algunas consideraciones sobre diseño de sistemas digitales.

Se ha estimado en estudios realizados que el costo de los circuitos integrados y componentes de un sistema, representa menos del 10% del costo total del sistema. Esto es muy significativo y nos hace revisar las metas de diseño que se tenían hasta hace 6 ó 7 años atrás.

Las primeras computadoras y aún las de principios de los 60, eran construídas empleando componentes discretos -diodos, transistores, resistores, capacitores, etc.- y por lo tanto, uno de los aspectos más estudiados en los centros de investigación y más enseñados en las Universidades, eran precisamente técnicas tendientes a minimizar el número de componentes, que equivale a minimizar funciones booleanas. El objetivo perseguido con estos métodos de minimización, era el de reducir el número de compuertas y flip-flop y de esta manera reducir el número de componentes. En la actualidad, los circuitos comerciales traen varias compuertas en cada circuito integrado -específicamente, 4 compuertas NAND de dos entradas, 3 compuertas NAND de 3 entradas, 2 compuertas NAND de 4 entradas, etc.- y en este contexto, el minimizar el número de compuertas pierde bastante sentido, por que si originalmente podíamos implementar una función con 8 compuertas y después de invertir cierto tiempo logramos implementarla con 5, la ganancia habrá sido cero porque en ambos casos requeriremos de dos circuitos integrados y en cambio sí hemos aumentado los costos de desarrollo, porque se invirtió el tiempo de un ingeniero en algo inútil.

Sin embargo, la cifra que dimos al principio, que el costo de todos los circuitos empleados en el sistema representa sólo el 10% de los mismos, hace aún más obsoletas las metas originales de diseño. Veamos, muy superficialmente, qué es lo que consume más dinero en un sistema y nos daremos cuenta que dichas metas deben ser modificadas drásticamente.

Para estos efectos, hemos empleado los datos proporcionados por Blakeslee [6] y que reproducimos en la Tabla 1-1. Según el autor, los datos corresponden a costos de 1974 y corresponden a una computadora de tamaño mediano y con un volumen moderado de producción.

De la tabla se observan dos cosas: que no hay un solo ítem en el que se pueda concentrar la minimización de costos y que todos los costos son casi proporcionales al número de circuitos integrados del sistema. Podría decirse que, un sistema con la mitad de los circuitos integrados, requeriría la mitad de circuitos impresos, la mitad de chasis, la mitad de la energía, la mitad de los esfuerzos de diseño, etc. De lo anterior parece evidente que la única forma real de minimizar costos es minimizar el número de circuitos integrados. Esto a su vez sugiere que si en un paquete de circuito integrado pudiéramos incluir funciones más complejas que la de una compuerta o de un flip-flop, lograríamos disminuir el total de circuitos integrados. Esto es posible hacerlo si se emplean circuitos MSI o LSI (integración a mediana y gran escala), con lo cual aunque el precio por circuito aumenta, el costo total del sistema disminuye.

En resumen, la mejor forma de abatir costos en un sistema digital, es disminuir el número de circuitos integrados mediante el uso de circuitos de integración a mediana y gran escala. Otros factores que deben ser considerados son confiabilidad y simplicidad de mantenimiento y reparación, los cuales no vamos a analizar.

TABLA 1-1

Costos proporcionales al número de circuitos integrados, en un sistema digital.

| Item   | Cost/Unit     | ICs/Unit <sup>a</sup> | Cost/IC <sup>b</sup> |
|--|---------------|-----------------------|----------------------|
| PC board   | \$20/board    | 50/board              | \$0.40               |
| PC connector   | \$5/board     | 50/board              | 0.10                 |
| Subrack (metalwork, guides, labor)                             | \$200/subrack | 1000/subrack          | 0.20                 |
| Backplane (subrack ground-power plane)                         | \$100/subrack | 1000/subrack          | 0.10                 |
| Wire wrap (automatic)  | 6¢/wire       | 1/wire                | 0.06                 |
| Power supplies (\$1/W, 70% utilized)                           | \$1.50/W      | 10/W                  | 0.15                 |
| Rack (including doors, fans, power distributor)                | \$500/rack    | 4000/rack             | 0.13                 |
| IC ordering, receiving, and inventory                          | 2¢/IC         | 1                     | 0.02                 |
| IC testing   | 8¢/IC         | 1                     | 0.08                 |
| PC board testing   | \$5/board     | 50/board              | 0.10                 |
| System checkout  | \$800/rack    | 4000/rack             | 0.20                 |
| Bypass capacitors (including insertion labor)                  | \$1/board     | 50/board              | 0.02                 |
| IC insertion and soldering                                     | 6¢/IC         | 1                     | 0.06                 |
| Interconnecting cables   | \$80/rack     | 4000/rack             | 0.02                 |
| Maintenance panel  | \$300/rack    | 4000/rack             | 0.07                 |
| System assembly  | \$200/rack    | 4000/rack             | 0.05                 |
| System packing and shipment                                    | \$200/rack    | 4000/rack             | 0.05                 |
| System design, drafting and prototype checkout (+ 100 systems) | \$2000/rack   | 4000/rack             | 0.50                 |
| Service cost for 5 years                                       | \$4000/rack   | 4000/rack             | 1.00                 |
| Total overhead cost per IC                                     |               |                       | = \$3.31             |

<sup>a</sup> Since it is impractical to fill PC boards completely, an average (rather than maximum) number of ICs per board, subrack, and so on, is assumed.

$$^b \text{Cost/IC} = \frac{(\text{cost/unit})}{(\text{ICs/unit})}$$

En los dos capítulos que siguen, estudiaremos los conceptos básicos de sistemas numéricos y álgebra de boole, en los cuales se fundamentan las técnicas digitales.

Referencias:

- |1| G.M. Walker "Choosinn Sides in digital Watch Technology", Electronics Vol. 49, No. 12, June 10, 1976. pp. 91-99.
- |2| A.A. Goldberg, "Digital Techniques promise to clarify the Television Picture", Electronics, Vol. 49 No.3, Feb. 5; 1976, pp 94-100.
- |3| R. Gunlach, "Large-Scale Integration is ready to answer the call of telecommunications", Electronics, Vol. 50, No. 9, April 28, 1977, pp 93-108.
- |4| D. M. Auslander, Y. Takahashi and M. Tonizuka, "The Next Generation of Single Loop Controllers", Journal of Dynamic Systems, Measurement and Control, Trans. ASME, Series G, Vol. 97, Sept. 1975, pp 280-282.
- |5| G.M. Walker, "LSI Controls gaining in home appliances", Electronics Vol. 50, No. 8, April 14, 1977, pp 91-99.
- |6| T. R. Blakeslee, "Digital Design with Standrad MSI and LSI", Wiley-Interscience, 1975.

## CAPITULO 2: CONCEPTOS BASICOS DE SISTEMAS NUMERICOS Y DISEÑO LOGICO.

### 2.1.- Notación Posicional:

El sistema numérico que nosotros empleamos es el sistema decimal. En dicho sistema numérico, tenemos 10 símbolos para representar una cantidad, a saber: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. A estos símbolos les llamaremos dígitos.

Un número decimal, representando una cantidad de objetos, está compuesto por dígitos ubicados en posiciones que representan las unidades, decenas, centenas, etc. Por ejemplo, el número decimal 547 nos representa una cantidad de objetos, de la cual tenemos 5 centenas, 4 decenas y 7 unidades.

Este concepto de considerar a un número formado por dígitos en distintas posiciones, nos sugiere que podríamos utilizar otro tipo de representación. Consideremos el número decimal 47821 y representémoslo de la siguiente forma:

$$47821 = 40000 + 7000 + 800 + 20 + 1$$

Esta representación es equivalente a:

$$47821 = 4 \times 10000 + 7 \times 1000 + 8 \times 100 + 2 \times 10 + 1$$

o bien, empleando potencias de 10, podemos escribir

$$47821 = 4 \times 10^4 + 7 \times 10^3 + 8 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$$

Podemos establecer entonces, que un número decimal de  $n$  dígitos, es la suma de coeficientes, cada uno con un peso asignado de acuerdo a su posición. En general, un número decimal de  $n$  dígitos se representa:

$$N_{10} = a_{n-1}(10)^{n-1} + a_{n-2}(10)^{n-2} + \dots + a_1(10)^1 + a_0(10)^0 \quad (2.1)$$

o bien,

$$N_{10} = a_{n-1} a_{n-2} \dots a_1 a_0 \quad (2.2)$$

que es la forma en que tradicionalmente escribimos un número.

En la expresión (2.2), la base 10 está sólo implicada. Pero, así como empleamos la base 10 para construir el sistema decimal, podríamos emplear cualquier otra base  $r$ . La ecuación (2.1) podemos generalizarla para un sistema base  $r$ , de la forma siguiente:

$$N_r = a_{n-1} r^{n-1} + a_{n-2} r^{n-2} + \dots + a_1 r^1 + a_0 r^0 \quad (2.3)$$

El requisito para generar números en una base  $r$  es que requerimos de  $r$  símbolos (incluyendo el 0). En la tabla siguiente se presentan los símbolos de algunos de los sistemas numéricos más comunes:

TABLA 2-1

| Base | Sistema Numérico | Símbolos                         |
|------|------------------|----------------------------------|
| 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. |

Hasta el momento, hemos considerado solamente números enteros. Para el caso de números fraccionarios, podemos extender las consideraciones anteriores. Consideremos el número decimal 0.234

$$0.234 = \frac{2}{10} + \frac{3}{100} + \frac{4}{1000}$$

o bien

$$0.234 = 2 \times 10^{-1} + 3 \times 10^{-2} + 4 \times 10^{-3}$$

o bien, en general para un número fraccionario decimal tenemos:

$$N_{10} = a_{-1} \times 10^{-1} + a_{-2} \times 10^{-2} + \dots + a_{-(m-1)} \times 10^{-(m-1)} + a_{-m} \times 10^{-m} \quad (2.4)$$

Finalmente, para un número fraccionario base  $r$

$$N_r = a_{-1} r^{-1} + a_{-2} r^{-2} + \dots + a_{-(n-1)} r^{-(n-1)} + a_{-m} \times r^{-m} \quad (2.5)$$

Combinando las ecuaciones (2.3) y (2.5), obtenemos la representación de un número mixto (que tiene parte entera y parte fraccionaria), base  $r$ .

$$N_r = a_{n-1} r^{n-1} + a_{n-2} r^{n-2} + \dots + a_1 r^1 + a_0 r^0 + a_{-1} r^{-1} + a_{-2} r^{-2} + \dots + a_{-m} r^{-m} \quad (2.6)$$

o bien

$$N_r = \sum_{i=-m}^n a_i r^i \quad (2.7)$$

A continuación explotaremos la representación posicional desarrollada para estudiar algunos sistemas numéricos de uso más frecuente. En particular, estudiaremos los sistemas binario, octal y hexadecimal.

## 2.2.- El Sistema Binario:

Este sistema es de gran importancia en sistemas digitales. Sabemos que en un sistema digital, la información se representa mediante dos niveles: 1 y 0, que pueden representar un voltaje y la ausencia de éste, o una intensidad de campo magnético y la ausencia de ésta, etc. En general, la filosofía de un sistema digital es la forma discreta y bivaluada en que se representa la información.

De lo anterior surge en forma obvia la importancia del sistema binario: el sistema binario emplea dos símbolos para representar una cantidad. Por lo tanto, debemos encontrar los mecanismos que nos permitan representar un número decimal en binario y viceversa. Debemos además establecer las reglas aritméticas de este sistema numérico. De esta forma, tendremos herramientas adecuadas para efectuar procesamientos numéricos mediante sistemas digitales.

## 2.2.- Conversión de Binario a Decimal:

El procedimiento para convertir un número binario a su equivalente decimal es tá basado en la ecuación (2.7).

Consideremos el número binario 110101101 y representémoslo en forma posicional:

$$110101101_2 = 1x2^8 + 1x2^7 + 0x2^6 + 1x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0$$

$$110101101_2 = 1x(256) + 1x(128) + 0x(64) + 1x(32) + 0x(16) + 1x(8) + 1x(4) + 0x(2) + 1x(1)$$

$$110101101 = 256 + 128 + 0 + 32 + 0 + 8 + 4 + 0 + 1$$

$$110101101 = 429_{10}$$

Si el número fuera fraccionario, procederíamos de acuerdo con la misma ecuación de la forma siguiente:

$$101101 = 1x2^{-1} + 0x2^{-2} + 1x2^{-3} + 1x2^{-4} + 0x2^{-5} + 1x2^{-6}$$

$$101101 = 1x(0.5) + 0x(0.25) + 1x(0.125) + 1x(0.0625) + 0x(0.03125) + 1x(0.015625)$$

$$101101 = 0.5 + 0.125 + 0.0625 + 0.015625$$

$$101101 = 0.703125$$

Podemos resumir entonces que:



PARA CONVERTIR UN NUMERO BINARIO A DECIMAL, REPRESENTAMOS EL NUMERO EN SU FORMA POSICIONAL Y EFECTUAMOS LAS OPERACIONES INDICADAS.

De esta regla se pueden derivar otras alternativas que pueden resultar más simples. Por ejemplo, sumar las potencias de dos correspondientes a las posiciones distintas de cero. Consideremos el caso siguiente:

$$101101_2 = ( \quad )_{10}$$

En este caso las potencias de dos correspondientes a las posiciones distintas de cero son: comenzando desde la posición menos significativa (el dígito del extremo derecho) tenemos:  $2^0, 2^2, 2^3, 2^5$ ; es decir sumamos: 1, 4, 8 y 32. El decimal equivalente será 45.

### 2.2.2.- Conversión de Decimal a Binario

Nuevamente, el procedimiento de conversión debemos buscarlo a partir de la ecuación (2.6). En este caso, el número se encuentra en decimal y queremos obtener su representación binaria. Luego, podemos escribir

$$N_{10} = a_{n-1}x2^{n-1} + a_{n-2}x2^{n-2} + \dots + a_1x2^1 + a_0x2^0 \quad (2.8)$$

Nuestro problema consiste en determinar los coeficientes  $a_{n-1}, a_{n-2}, \dots, a_1, a_0$ . Como el sistema es binario, sabemos que podrán tomar sólo los valores 1 ó 0. Es decir, debemos determinar cuales coeficientes valen 1 y cuales valen 0.

Si dividimos ambos lados de la ecuación (2.8) por 2, obtenemos un cociente  $Q_1$  que es número entero y un residuo  $a_0$ :

$$\frac{N_{10}}{2} = Q_1 + a_0 2^{-1} \quad (2.9)$$

donde

$$Q_1 = a_{n-1}x2^{n-2} + a_{n-2}x2^{n-3} + \dots + a_2x2^1 + a_1x2^0 \quad (2.10)$$

El residuo  $a_0$  obtenido de esta manera, es dígito menos significativo de la representación binaria de  $N_{10}$ . Si ahora procedemos de igual forma con  $Q_1$ , obtendremos un cociente entero  $Q_2$  y un residuo  $a_1$ :

$$\frac{Q_1}{2} = Q_2 + a_1 x 2^{-1} \quad (2.11)$$

donde

$$Q_2 = a_{n-1} x 2^{n-3} + a_{n-2} x 2^{n-4} + \dots + a_3 x 2^1 + a_2 x 2^0 \quad (2.12)$$

Si procedemos de esta forma hasta que el cociente resultante sea cero, habremos obtenido todos los coeficientes y por lo tanto, la representación binaria del número en cuestión. Por ejemplo, encontremos la representación binaria del número decimal  $429_{10}$ .

| Cociente     | Residuo | Posición |
|--------------|---------|----------|
| $429 \div 2$ | 1       | $a_0$    |
| $214 \div 2$ | 0       | $a_1$    |
| $107 \div 2$ | 1       | $a_2$    |
| $53 \div 2$  | 1       | $a_3$    |
| $26 \div 2$  | 0       | $a_4$    |
| $13 \div 2$  | 1       | $a_5$    |
| $6 \div 2$   | 0       | $a_6$    |
| $3 \div 2$   | 1       | $a_7$    |
| $1 \div 2$   | 1       | $a_8$    |
| 0            |         |          |

Luego

$$429_{10} = 110101101$$

Otra forma de proceder es buscar la potencia de dos más alta que esté contenida en el número original; por ejemplo, si el número es  $489_{10}$ , la potencia más cercana es  $2^8 = 256$ . A continuación buscamos la potencia de 2 más alta contenida en  $489 - 2^8 = 233$ , que en este caso es  $2^7 = 128$ . Luego, buscamos la potencia de 2 más alta contenida en  $233 - 2^7 = 105$ , que es  $2^6$ . Procediendo de esta forma, encontraremos las potencias de 2 cuya suma es igual al número original. En el ejemplo que estábamos analizando, dichas potencias son:

$$\begin{array}{r}
 2^8 = 256 \\
 2^7 = 128 \\
 2^6 = 64 \\
 2^5 = 32 \\
 2^3 = 8 \\
 2^0 = 1 \\
 \hline
 489
 \end{array}$$

De esta forma, obtenemos todos los coeficientes iguales a 1 de la representación binaria del número original. Todas las potencias de 2 que no aparecen en nuestra lista, serán aquellas correspondientes a los coeficientes iguales a cero.

En nuestro ejemplo, tendremos:

$$489_{10} = 111101001_2$$

El problema de convertir a binario un número decimal fraccionario, se resuelve en forma similar. Si para números enteros, los coeficientes de la representación binaria los obteníamos por medio de divisiones sucesivas, para números fraccionarios los obtenemos mediante multiplicaciones sucesivas. Esto se puede comprobar observando la ecuación (2.5)

$$N_{10} = a_{-1}x2^{-1} + a_{-2}x2^{-2} + \dots + a_{-m}x2^{-m}$$

donde  $N_{10}$  es el número decimal fraccionario y los coeficientes  $a_{-1}, a_{-2}, \dots, a_{-m}$  corresponden a la representación binaria de  $N_{10}$  y son los que debemos determinar. Multiplicando ambos lados de la ecuación, obtenemos:

$$2xN_{10} = a_{-1} + a_{-2}x2^{-1} + \dots + a_{-m}x2^{-m+1}$$

Es decir, hemos obtenido el coeficiente  $a_{-1}$ , que podrá ser 0 ó 1. Procediendo de esta forma podemos obtener los coeficientes restantes. Consideremos el siguiente ejemplo:

Convertir  $0.703125_{10}$  a binario.

$$0.703125 \times 2 = 1 + 0.406250 \quad a_{-1} = 1$$

$$0.406250 \times 2 = 0 + 0.812500 \quad a_{-2} = 0$$

$$0.812500 \times 2 = 1 + 0.625000 \quad a_{-3} = 1$$

$$0.625000 \times 2 = 1 + 0.250000 \quad a_{-4} = 1$$

$$0.250000 \times 2 = 0 + 0.500000 \quad a_{-5} = 0$$

$$0.500000 \times 2 = 1 + 0.000000 \quad a_{-6} = 1$$

$$0.000000 \times 2 = 0 + 0.000000$$

Luego

$$0.703125_{10} = 101101_2$$

En el ejemplo vemos que los coeficientes iguales a 1 son generados cuando al multiplicar el número fraccionario se obtiene una parte entera. Además se observa que el proceso termina cuando el número fraccionario se hace 0. Esto último no es siempre el caso. Consideremos el ejemplo siguiente:

Convertir  $0.59_{10}$  a binario

$$.39 \times 2 = 0 + 0.78 \quad a_{-1} = 0$$

$$.78 \times 2 = 1 + 0.56 \quad a_{-2} = 1$$

$$.56 \times 2 = 1 + 0.12 \quad a_{-3} = 1$$

$$.12 \times 2 = 0 + 0.24 \quad a_{-4} = 0$$

$$.24 \times 2 = 0 + 0.48 \quad a_{-5} = 0$$

$$.48 \times 2 = 0 + 0.96 \quad a_{-6} = 0$$

$$.96 \times 2 = 1 + 0.92 \quad a_{-7} = 1$$

$$\begin{aligned}
 & \cdot 92 \times 2 = 1 + 0.84 \quad a_{-8} = 1 \\
 & \cdot 84 \times 2 = 1 + 0.68 \quad a_{-9} = 1 \\
 & \cdot 68 \times 2 = 1 + 0.36 \quad a_{-10} = 1 \\
 & \cdot 36 \times 2 = 0 + 0.72 \quad a_{-11} = 0 \\
 & \cdot 72 \times 2 = 0 + 0.44 \quad a_{-12} = 1
 \end{aligned}$$

Como puede observarse, aún cuando sigamos con el proceso éste no terminará. Es decir, 0.39 no se puede representar en forma exacta en binario; en este caso,

$$0.39 \approx 0.011000111101_2$$

El error introducido en la representación será menor que el dígito menos representativo, es decir:

$$\epsilon < 2^{-12} = 1/4096 = 0.000244141$$

Si convertimos a decimal el número binario obtenido, tendremos:

$$0.011000111101_2 = 0.389892578$$

Es decir, el error es de 0.000107472

Este problema se traduce en un problema de exactitud de la representación y existen varias formas de atacarlo. Por lo general la limitación más grande es el número de bits o "longitud de palabra" con que se cuenta para representar un número. Por ejemplo, en la mayoría de los microprocesadores, dicha longitud de palabra es de 8 bits, en cambio, en computadoras grandes de propósito general, la longitud de palabra llega a 64 bits, como es el caso de la CDC 6400/6600.

### 2.2.3.- Aritmética Binaria:

Así como tenemos aritmética decimal, existe una aritmética binaria, cuyas reglas son idénticas a la decimal pero en este caso tenemos solo dos símbolos: 0 y 1 para representar un número.

A continuación se dan dos tablas aritméticas: suma y Multiplicación

Tabla 2.2: Suma binaria

Augendo + Adendo = Resultado + Acarreo

|   |   |   |   |     |
|---|---|---|---|-----|
| 0 | 0 | = | 0 |     |
| 0 | 1 | = | 1 |     |
| 1 | 0 | = | 1 |     |
| 1 | 1 | = | 0 | + 1 |

Tabla 2.3: Multiplicación Binaria

$$\begin{array}{r|ll}
 X & 0 & 1 \\
 \hline
 0 & 0 & 0 \\
 1 & 0 & 1
 \end{array}$$

Veamos algunos ejemplos de suma y multiplicación de números binarios

1) Sumar  $1011 + 1101$  ( $= 11 + 13$ )

$$\begin{array}{rcccc}
 (1) & (1) & (1) & (1) & \leftarrow \text{Acarreo} \\
 & 1 & 0 & 1 & 1 \\
 & 1 & 1 & 0 & 1 \\
 \hline
 1 & 1 & 0 & 0 & 0 \quad (= 34)
 \end{array}$$

2) Multiplicar  $1001 \times 1011$  ( $9 \times 11$ )

$$\begin{array}{r}
 1001 \times 1011 \\
 \hline
 1001 \\
 1001 \\
 0000 \\
 1001 \\
 \hline
 1100011 \quad (= 99)
 \end{array}$$

La substracción binaria será analizada posteriormente cuando veamos la representación de números negativos. Finalmente, la división no tiene mayor interés práctico, ya que puede ser implementada mediante restas.

### 2.3.- Sistemas Octal y Hexadecimal:

Anteriormente se hizo notar que el sistema binario es el de mayor importancia para trabajar con sistemas digitales. Sin embargo, los humanos estamos acostumbrados a trabajar con el sistema decimal. Resulta bastante difícil tener una idea de los órdenes de magnitud de cantidades representadas en binario. Por esto se ha buscado un sistema numérico lo más cercano al decimal y que a la vez sea muy cercano al decimal. Dos de estos sistemas numéricos son el octal (base 8) y el hexadecimal (base 16). El primero resulta obvio que se acerca bastante al decimal: cuenta con ocho símbolos para representar una cantidad. En el caso del sistema hexadecimal, esta ventaja ya no es tan obvia puesto que requerimos de 16 símbolos y por lo tanto hay que emplear 6 letras (A, B, C, D, E, F) para suplir los símbolos numéricos faltantes. Esto obviamente introduce un grado más de complejidad y, aparentemente, nos hace más difícil la vida. Sin embargo, estudiemos un poco estos dos sistemas nu-

méricos y veremos que es posible justificar su empleo, más aún cuando se está haciendo cada vez más empleado con el advenimiento de los microprocesadores.

La característica más importante de los sistemas octal y hexadecimal es que sus bases (8 y 16) son potencias de 2 y por lo tanto, se relacionan directamente con el sistema binario. En efecto,

$$8 = 2^3$$

$$16 = 2^4$$

Estas relaciones nos sugieren que para representar un número octal (base 8) en binario, requerimos de tres dígitos (o bits - de binary digit) y para representar uno hexadecimal, requerimos cuatro dígitos. Esto se ilustra en la tabla siguiente:

Tabla 2.4

| Hexadecimal | Octal | Binario   |
|-------------|-------|-----------|
| 0           | 0     | 0 0 0 0 0 |
| 1           | 1     | 0 0 0 0 1 |
| 2           | 2     | 0 0 0 1 0 |
| 3           | 3     | 0 0 0 1 1 |
| 4           | 4     | 0 0 1 0 0 |
| 5           | 5     | 0 0 1 0 1 |
| 6           | 6     | 0 0 1 1 0 |
| 7           | 7     | 0 0 1 1 1 |
| 8           | 10    | 0 1 0 0 0 |
| 9           | 11    | 0 1 0 0 1 |
| A           | 12    | 0 1 0 1 0 |
| B           | 13    | 0 1 0 1 1 |
| C           | 14    | 0 1 1 0 0 |
| D           | 15    | 0 1 1 0 1 |
| E           | 16    | 0 1 1 1 0 |
| F           | 17    | 0 1 1 1 1 |
| 10          | 20    | 1 0 0 0 0 |
| 11          | 21    | 1 0 0 0 1 |
| 12          | 22    | 1 0 0 1 0 |
| :           | :     |           |
| 1F          | 37    | 1 1 1 1 1 |

De la tabla vemos que los ocho símbolos octales son representados por las ocho combinaciones diferentes de tres bits. Cuando la cantidad octal requiere de más de un dígito para ser expresada, su representación en binario se obtiene simplemente reemplazando cada dígito octal por sus tres bits correspondientes a su representación en binario. Veamos esto con un ejemplo:

Representar  $712_8$  en binario:

Procedemos reemplazando cada dígito octal por su representación en binario:

$$7_8 = 111_2$$

$$1_8 = 001_2$$

$$3_8 = 011_2$$

Luego

$$713_8 = 111\ 001\ 011_2$$

Comprobemos ésto convirtiendo ambas cantidades a decimal:

$$\begin{aligned} 713_8 &= 7 \times 8^2 + 1 \times 8^1 + 3 \times 8^0 = 7 \times 64 + 8 + 3 = \\ &= 448 + 8 + 3 = 459_{10} \end{aligned}$$

$$\begin{aligned} 111001011 &= 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + \\ &+ 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 256 + 128 + 64 + 0 + 0 + 8 + 0 + 2 + 1 = \\ &= 459_{10} \end{aligned}$$

Esto mismo nos sugiere que para convertir un número binario a octal, podemos proceder agrupando (desde la derecha hacia la izquierda) los dígitos binarios en grupos de tres y luego reemplazar directamente por su equivalente en octal.

Convertir  $101011011_2$  a octal

$$\begin{array}{ccc} \underbrace{101} & \underbrace{011} & \underbrace{011} \\ 5 & 3 & 3 \end{array}$$

Luego:

$$101011011_2 = 533_8$$

¿Qué sucede cuando el último grupo tiene menos de tres bits? Muy simple: agregamos tantos ceros a la izquierda como sean necesarios. Los ceros a la izquierda no alteran la cantidad.

Convertir  $1101101111_2$  a octal

$$\begin{array}{cccc} \underbrace{001} & \underbrace{101} & \underbrace{101} & \underbrace{111} \\ 1 & 5 & 5 & 7 \end{array}$$

Luego:

$$1101101111_2 = 1557_8$$

Para el caso de números hexadecimales, los procedimientos de conversión son idénticos, excepto que ahora cada dígito hexadecimal requiere de cuatro bits y viceversa.

1) Convertir  $E70A_{16}$  a binario

$$E_{16} = 1110_2$$

$$7_{16} = 0111_2$$

$$0_{16} = 0000_2$$

$$A_{16} = 1010_2$$

Luego

$$E70A_{16} = 1110011100001010_2$$

2) Convertir  $1111101011101$  a hexadecimal

$$\begin{array}{cccc} \underbrace{0001} & \underbrace{1111} & \underbrace{0101} & \underbrace{1101} \\ 1 & F & 5 & D \end{array}$$

Luego

$$1111101011101_2 = 1F5D_{16}$$

Aún cuando existen métodos para convertir entre cualquier par de bases, y por lo tanto, existen para convertir de octal a decimal y viceversa y de hexadecimal a decimal y viceversa, éstos no serán considerados en las presentes notas. El interesado puede consultar, por ejemplo, el libro de A. Barna y D.I. Porat [1].

Sin embargo, aún cuando no es muy eficiente, si se desea convertir un número decimal a octal o hexadecimal, se puede convertir primero de decimal a binario mediante el procedimiento de la sección 2.2.2. y luego convertir de binario a octal o hexadecimal, que no es más que un paso mecánico.

Regresando a nuestra justificación del empleo de los sistemas octal y hexadecimal, vemos que cada dígito octal requiere de tres bits para su representación y en cambio cada dígito hexadecimal requiere de cuatro bits. Este hecho tan simple es la base de la preferencia por el sistema hexadecimal. Casi todos los sistemas digitales están organizados en base a registros o palabras con un número de bits que es una potencia de dos. Así vemos que muchos microprocesadores (18080, M6800, COSMAC, SC/MP, etc.) son de ocho bits; muchas minicomputadoras son de 16 o 32 bits y lo mismo sucede con varias de las computadoras grandes (IBM: 32 bits, CDC: 64 bits). Esto hace que sea muy simple agrupar bits en grupos de 4 y representarlos directamente en hexadecimal. No sucede así con el octal, en que la organización de palabras en múltiplos de 3 no se encuentran muy frecuentemente en sistemas digitales. De hecho, las instrucciones de la gran mayoría de los microprocesadores vienen especificados en hexadecimal.



#### 2.4.- Representación de Números Negativos:

Hasta el momento hemos considerado solamente el problema de representar números positivos. Sin embargo, con mucha frecuencia debemos trabajar con números negativos. Una forma de atacar el problema podría ser emplear un bit adicional y hacerlo 0 cuando la cantidad sea positiva y 1 cuando sea negativa. Aún cuando esto es perfectamente válido, tiene el inconveniente que requerimos de un bit extra que significa contar con un elemento electrónico adicional en todas aquellas partes del sistema en que debamos "recordar" el signo de la cantidad que se esté almacenando. Esto no es muy deseable desde el punto de vista económico.

Afortunadamente existe una forma de representar números negativos que no requiere de un bit extra y que además tiene grandes cualidades desde el punto de vista aritmético. Esta forma se llama complemento a la base.

El problema puede ser visto de la forma siguiente: si tenemos que restar dos números:

$$7 - 2 = 5$$

podemos invertir el problema y hacerlo de la forma siguiente:

$$7 + (10 - 2) = 7 + 8 = \cancel{15}$$

El proceso de restarle a 10 el sustraendo es lo que se conoce como el complemento a 10. Veamos otro ejemplo:

$$374 - 254 = ?$$

en primer lugar obtenemos el complemento a 10 del sustraendo 254:

$$\begin{array}{r} 1000 \\ 254 \\ \hline 746 \end{array}$$

Ahora sumamos:

$$374 + 746 = \cancel{1120}$$

el resultado lo obtenemos ignorando el acarreo a la posición más significativa.

Todo esto parece hacer más complicado (para nosotros) el simple proceso de restar dos números. Sin embargo, veremos que aún cuando para nosotros puede resultar complicado, puede implementarse en forma muy simple en un sistema digital.

Anteriormente vimos el complemento a la base decimal es decir, el complemento a 10. Veamos ahora el complemento a la base binaria, o complemento a dos. El complemento a dos lo obtenemos restando la cantidad a la potencia de dos inmediatamente superior. Obtengamos el complemento a dos de 1101:

$$\begin{array}{r}
 10000 \leftarrow 2^4 \\
 -1101 \leftarrow \text{Cantidad a complementar} \\
 \hline
 0011 \leftarrow \text{Complemento a dos}
 \end{array}$$

Todavía sigue siendo complicado. Sin embargo, se puede simplificar. Consideremos el mismo número y cambiemos los 0 por 1 y los 1 por 0 y luego sumémosle 1:

$$\begin{array}{r}
 1101 \leftarrow \text{Cantidad original} \\
 0010 \leftarrow \text{Complemento} \\
 + 1 \\
 \hline
 0011 \leftarrow \text{Complemento a dos}
 \end{array}$$

Lo que hicimos en este ejemplo fue obtener el complemento a 1 de la cantidad original y luego sumarle. El complemento a 1, formalmente se obtiene restando de  $2^n - 1$  la cantidad a complementar. Por ejemplo, obtengamos el complemento a 1 de 101101:

$$\begin{array}{r}
 111111 \leftarrow 2^6 - 1 \\
 -101101 \leftarrow \text{Cantidad a complementar} \\
 \hline
 010010 \leftarrow \text{Complemento a uno}
 \end{array}$$

Sin embargo, nos podemos dar cuenta que este procedimiento es innecesario, ya que si "negamos" cada dígito (cambiamos los 1 por 0 y viceversa) estamos logrando el mismo efecto.

Podemos concluir que: El complemento a 2 de un número binario se obtiene sumando 1 al complemento a 1 de dicho número.

Regresemos ahora al problema de la resta de dos números. Consideremos primero la resta binaria siguiente:

$$\begin{array}{r}
 \text{Minuendo} \quad 10001 \\
 \text{Sustraendo} \quad - 01011 \\
 \hline
 00110
 \end{array}$$

La misma operación la podemos realizar sumando al minuendo el complemento a dos del sustraendo:

$$\begin{array}{r}
 \text{Minuendo} \quad 10001 \\
 \text{Complemento a} \\
 \text{2 del Sustraen} \\
 \text{do.} \quad 10101 \\
 \hline
 \cancel{X}00110
 \end{array}$$

el resultado lo obtenemos ignorando el dígito de acarreo. El resultado es entonces: 00110.

Consideremos otro ejemplo:

|  |  |
|--|--|
| Minuendo: 11001<br>Sustraendo: -101<br>= 10100 | 11001 Minuendo<br>+ 11011 Complemento a dos del Sustraendo<br>X10100 |
|--|--|

Quando la magnitud del sustraendo es mayor que el minuendo, no habrá acarreo y el resultado final será el negativo del complemento a 2 del resultado de la suma. Para ver más claramente lo anterior, consideremos una versión decimal de lo anterior:

$$2 - 9 = 2 + (10-9) = 2 + 1 = 3$$

Luego, el resultado final será  $-(10-3) = 7$ . Veamos un ejemplo binario:

|   |   |
|---|---|
| 101 Minuendo<br>- 11011 Sustraendo<br>- 10110 | 101<br>+00101 Complemento a 2<br>01010 Resultado de la suma<br>-10110 Complemento a 2 del resultado |
|---|---|

Podemos concluir lo siguiente:

AL EFECTUAR UNA RESTA EMPLEANDO COMPLEMENTO A DOS, EL ACARREO A LA POSICION MAS SIGNIFICATIVA NOS DARA EL SIGNO DEL RESULTADO. SI EL ACARREO ES 1, LA RESPUESTA SERA POSITIVA (CASO EN QUE EL MINUENDO > SUSTRAENDO); SI EL ACARREO ES 0, EL RESULTADO FINAL SERA EL NEGATIVO DEL COMPLEMENTO A DOS DEL RESULTADO DE LA SUMA.

Para concluir, cabe hacer notar que la obtención del complemento a dos de un número es muy simple de implementar digitalmente y que esto nos evita el tener que efectuar restas.

## 2.5.- Algunos códigos más usados

Dado que internamente, una computadora o un sistema digital, almacena y procesa información en binario, es necesario contar con códigos que permitan traducir números y caracteres a una forma compatible con el tipo de re presentación interna de un sistema digital. En particular, nos interesan los códigos más comunes: BCD, hexadecimal y alfanumérico.

### 2.5.1.- Código BCD:

El código BCD ("Binary Coded Decimal") se emplea para representar directamente los números decimales en binarios. La representación en BCD se obtiene reemplazando cada dígito de un número decimal, por un grupo de cuatro dígitos binarios. En la tabla siguiente se muestra la representación en BCD de los nueve dígitos decimales.

| Decimal | B C D   |
|---------|---------|
| 0       | 0 0 0 0 |
| 1       | 0 0 0 1 |
| 2       | 0 0 1 0 |
| 3       | 0 0 1 1 |
| 4       | 0 1 0 0 |
| 5       | 0 1 0 1 |
| 6       | 0 1 1 0 |
| 7       | 0 1 1 1 |
| 8       | 1 0 0 0 |
| 9       | 1 0 0 1 |

Ejemplo: Representar 375 en BCD. De la tabla anterior obtenemos la representación en BCD de cada dígito decimal y lo reemplazamos directamente:

$$\underbrace{0011}_3 \underbrace{0111}_7 \underbrace{0101}_5 \text{BCD} = 375_{10}$$

En binario, el mismo número se representa como sigue:

$$10111011_2 = 375_{10}$$

Ejemplo: Convertir el número BCD 0101011110000001 a decimal.

Agrupamos de a cuatro dígitos binarios el número BCD y reemplazamos cada grupo de dígitos por su equivalente decimal.

$$\underbrace{0101}_5 \underbrace{0111}_7 \underbrace{1000}_0 \underbrace{0000}_0 \text{BCD} = 5781_{10}$$

Ejemplo: convertir el número BCD 010111000001 a decimal.

Procediendo como en el ejemplo anterior

$$\underbrace{0101}_5 \quad \underbrace{1100}_{\text{ilegal}} \quad \underbrace{0001}_1$$

En este caso, el grupo 1100 no representa ningún dígito decimal codificado en BCD y por lo tanto es un código ilegal.

### 2.5.2.- Código Hexadecimal:

El código binario puro de cuatro bits, en el que se usan todas las combinaciones posibles constituye el código hexadecimal. Las seis combinaciones no usadas en BCD se simbolizan con letras, A a F.

| DECIMAL | HEXADECIMAL | DENOMINACION |
|---------|-------------|--------------|
| 0       | 0 0 0 0     | 0            |
| :       | :           | :            |
| 9       | 1 0 0 1     | 9            |
| 10      | 1 0 1 0     | A            |
| 11      | 1 0 1 1     | B            |
| 12      | 1 1 0 0     | C            |
| 13      | 1 1 0 1     | D            |
| 14      | 1 1 1 0     | E            |
| 15      | 1 1 1 1     | F            |

Ejemplo: El número binario 1 1 0 0 0 1 0 1 corresponde a C 5 hexadecimal y a  $12 \times 16 + 5 = 197$  decimal.

### 2.5.3.- Códigos Alfanuméricos:

Este tipo de códigos permite representar en binario cualquier tipo de carácter: dígitos, letras y símbolos especiales. Los dos códigos alfanuméricos más empleados son el ASCII (American Standard Code for Information Interchange Code). El código ASCII es empleado por todos los fabricantes de minicomputadores y microcomputadoras.

Usa 7 bits (más uno de paridad, para seguridad en las comunicaciones). Además de información alfanumérica, el código contiene posiciones que son interpretadas por las terminales de comunicaciones para realizar ciertas funciones ("retorno del carro", "avance del papel", "conectar perforador de cinta", etc.)

Se adjunta una tabla del código ASCII.

### 2.6.- Algebra de Boole

Con esta sección iniciamos el estudio de otro aspecto básico de las técnicas digitales: Las herramientas matemáticas que nos permitirán describir sistemas digitales en forma exacta.

En general, un sistema digital puede caracterizarse por dos aspectos: capacidad de tomar decisiones y capacidad de almacenar información.

En esta sección estableceremos las bases para lograr la implementación física del proceso de toma de decisiones lógicas.

Para implementar un proceso de toma de decisiones, debemos en primer lugar establecer un modelo de dicho proceso. Este modelo surgió de la forma en que toma decisiones el hombre. De este estudio surgió la lógica proposicional y posteriormente, la lógica matemática.

Uno de los primeros en formular una teoría fue George Boole, quién en 1849 publicó su libro "An Investigation of the Laws of Thought, on which are Founded the Mathematical Theories of Logic and Probability", en donde presentó una formulación algebraica del proceso del pensamiento lógico y del razonamiento. Esta formulación se ha llamado Algebra de Boole y se emplea extensivamente en el estudio de circuitos combinacionales.

### 2.6.1.- Postulados de Huntington:

Como toda teoría deductiva, el Algebra de Boole está basada en un conjunto de postulados. Estos postulados fueron establecidos por Huntington en 1904.

Postulado No. 1: Existe un conjunto  $K$  de elementos, sujetos a una relación de equivalencia denotada "=", que satisface el principio de substitución.

(por substitución se entiende que si  $a = b$ , se puede substituir  $a$  por  $b$  en cualquier expresión que contenga  $a$ , sin afectar la validéz de la expresión).

Postulado No. 2: Se define una regla de combinación "+", tal que si  $A \in K$  y  $B \in K$ ,  $(A + B) \in K$ .

Postulado No. 3: Se define una regla de combinación ".", tal que  $A \cdot B \in K$  si  $A \in K$  y  $B \in K$ .

Postulado No. 4: Existe un elemento "0" tal que:  $A + 0 = A \forall A \in K$ .

Postulado No. 5: Existe un elemento "1" tal que  $A \cdot 1 = A \forall A \in K$ .

Postulado No. 6: Si  $A \in K$  y  $B \in K$ , luego: conmutación de +  $A + B = B + A$

Postulado No. 7: Si  $A \in K$  y  $B \in K$ , luego: conmutación de  $\cdot$   $A \cdot B = B \cdot A$

Postulado No. 8: Si  $A \in K$ ,  $B \in K$  y  $C \in K$ , luego : distributiva  $A + (B \cdot C) = (A + B) \cdot (A + C)$

Postulado No. 9: Si  $A \in K$ ,  $B \in K$  y  $C \in K$ , luego: distributiva  $A \cdot (B + C) = A \cdot B + A \cdot C$

Postulado No. 10: Para todo elemento  $A \in K$ , existe un elemento  $\bar{A}$  tal que:

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

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

Postulado No. 11: Existen al menos dos elementos  $x \in K$  y  $y \in K$ , tales que  $x \neq y$ .

2.6.2.- Diagramas de Venn:

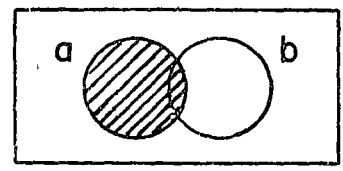
Dado que (la teoría de conjuntos) el álgebra de conjuntos es un álgebra de Boole, podemos utilizar los diagramas de Venn, empleados en teoría de conjuntos.

La correspondencia entre álgebra de conjuntos y de Boole es la siguiente

| Algebra de Boole | Algebra de conjuntos |
|------------------|----------------------|
| $\cdot$ (and)    | (intersección)       |
| $+$ (or)         | (unión)              |
| 1                | I (universo)         |
| 0                | $\emptyset$ (vacío)  |

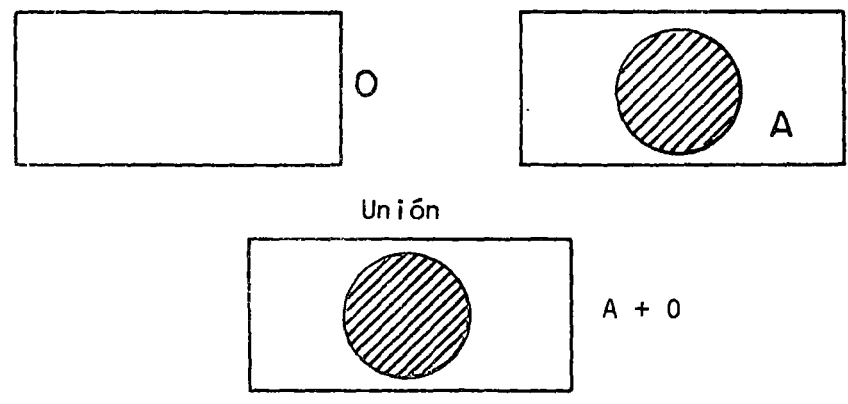
En los diagramas de Venn, los conjuntos se muestran como entornos cerrados (cuadrados, círculos, elipses, etc.)

Ejemplo:

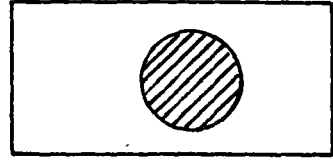
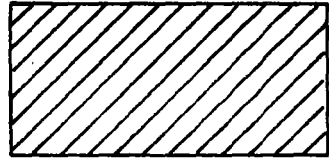


El conjunto a es el sombreado

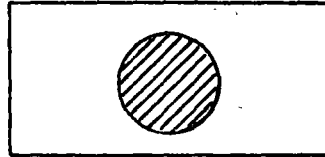
Postulado No. 4:  $A + 0 = A$



Postulado No. 5:  $A \cdot 1 = A$

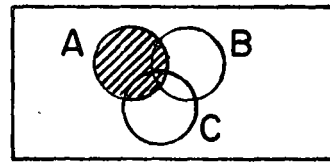
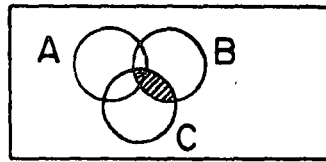


Intersección

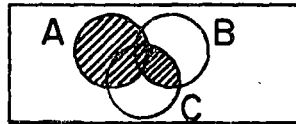


$A \cdot 1$

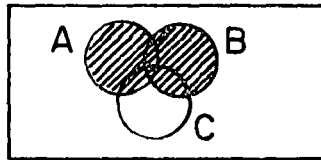
Postulado No. 8:  $A + BC = (A+B)(A+C)$



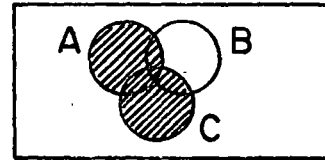
Unión



$A + BC$

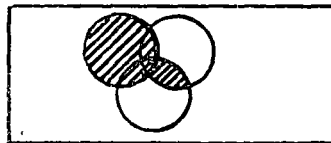


$A + B$



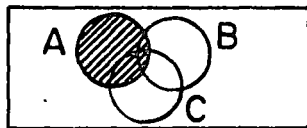
$A + C$

Intersección

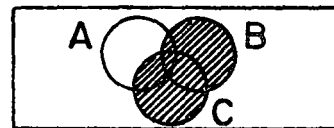


$(A+B)(A+C)$

Postulado No. 9:  $A \cdot (B+C) = AB + AC$

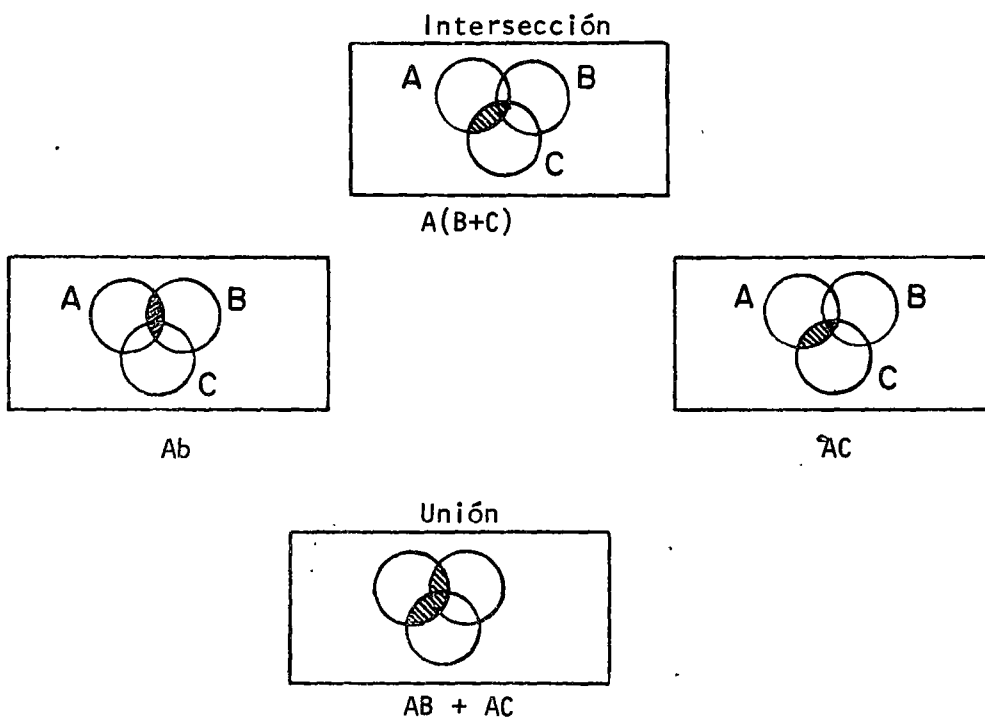


$A$



$B + C$





### 2.6.2.- Teoremas Fundamentales del Algebra de Boole:

A continuación presentamos, sin demostración, aquellos teoremas del álgebra de Boole que nos serán útiles en el trabajo subsecuente. Varios de estos teoremas pueden ser verificados empleando los diagramas de Venn.

T.1.- El 0 y el 1 de los postulados 4 y 5 son elementos únicos.

T.2.- Para todo  $a \in K$

i)  $a + a = a$

ii)  $a \cdot a = a$

T.3.- Para todo  $a \in K$

i)  $a + 1 = 1$

ii)  $a \cdot 0 = 0$

T.4.- Los elementos 1 y 0 son distintos, y  $\bar{1} = 0$ .

T.5.- Para todo par de elementos  $a$  y  $b$  en  $K$ ,

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

ii)  $a(a + b) = a$

T.6.- El elemento  $\bar{a}$  definido en el postulado 10 es único, para todo  $a \in K$ .

T.7.- Para todo  $a \in K$ ,  $\bar{\bar{a}} = a$ .

T.8.-  $a|(a+b) + c| = |(a+b) + c|a = a$

T.9.- Para todo  $a \in K$ ,  $b \in K$  y  $c \in K$ ,

$$\begin{aligned} \text{i)} & a + (b + c) = (a + b) + c \\ \text{ii)} & a \cdot (b \cdot c) = (a \cdot b) \cdot c \end{aligned}$$

T.10.- Para todo  $a \in K$  y  $b \in K$ ,

$$\begin{aligned} \text{i)} & a + \bar{a}b = a + b \\ \text{ii)} & a(\bar{a} + b) = ab \end{aligned}$$

T.11.- Leyes de De Morgan:

Para todo  $a \in K$  y  $b \in K$

$$\begin{aligned} \text{i)} & \overline{a + b} = \bar{a} \cdot \bar{b} \\ \text{ii)} & \overline{a \cdot b} = \bar{a} + \bar{b} \end{aligned}$$

T.12.- Para todo  $a \in K$ ,  $b \in K$  y  $c \in K$ ,

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

T.13.- Para todo  $a \in K$  y  $b \in K$ ,

$$\begin{aligned} \text{i)} & ab + a\bar{b} = a \\ \text{ii)} & (a + b)(a + \bar{b}) = a \end{aligned}$$

T.14.- Para todo  $a \in K$ ,  $b \in K$  y  $c \in K$ ,

$$\begin{aligned} \text{i)} & ab + a\bar{b}c = ab + ac \\ \text{ii)} & (a + b)(a + \bar{b} + c) = (a + b)(a + c) \end{aligned}$$

T.15.- Para todo  $a \in K$ ,  $b \in K$  y  $c \in K$ ,

$$\begin{aligned} \text{i)} & ab + \bar{a}c = (a + c)(\bar{a} + b) \\ \text{ii)} & (a + b)(\bar{a} + c) = ac + \bar{a}b \end{aligned}$$

### 2.6.3.- Funciones Booleanas

Los postulados y teoremas del álgebra Booleana presentados en las secciones anteriores, se dieron en términos generales y sin especificar los elementos del conjunto  $K$ , por lo cual, los resultados son válidos para cualquier álgebra de Boole. En la discusión que sigue, nos concentraremos en un álgebra donde  $K = 0, 1$ . Esta formulación es comúnmente llamada "álgebra de conmutación".

El concepto de función es conocido del álgebra ordinaria. Las funciones booleanas representan el concepto correspondiente para álgebra de Boole y se pueden definir como sigue:

Sean  $X_1, X_2, \dots, X_n$  símbolos que llamaremos variables, cada uno de las cuales puede representar un 1 o un 0 de un álgebra Booleana (se dice que 0 ó 1 son el valor de la variable).

Sea  $f(X_1, X_2, \dots, X_n)$  una función Booleana de las variables  $X_1, X_2, \dots, X_n$ .

La función  $f$  representa el valor 0 ó 1 dependiendo de los valores asignados a  $X_1, X_2, \dots, X_n$ .

Una función Booleana se puede describir mediante una expresión booleana como sigue:

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

Si  $A = 1, B = C = 0$ , entonces  $f = 1$  como se verifica a continuación:

$$\begin{aligned} f(1,0,0) &= 1 \cdot 0 + \bar{1} \cdot 0 + 1 \cdot \bar{0} \\ &= 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 = 0 + 0 + 1 \\ &= 1 \end{aligned}$$

Una alternativa en la descripción de funciones Booleanas, se puede dar mediante el uso de una tabla de verdad.

Una tabla de verdad presenta todas las combinaciones posibles de valores posibles de  $f$ .

Si evaluamos la función  $f(A,B,C) = AB + \bar{A}C + A\bar{C}$  para todas las combinaciones posibles de las variables encontramos:

$$\begin{aligned} f(0,0,0) &= 0 \\ f(0,0,1) &= 1 \\ f(0,1,0) &= 0 \\ f(0,1,1) &= 1 \\ f(1,0,0) &= 1 \\ f(1,0,1) &= 0 \\ f(1,1,0) &= 1 \\ f(1,1,1) &= 1 \end{aligned}$$

Si estos valores se listan en forma tabular, obtenemos la tabla de verdad.

| A | B | C | f(A,B,C) |
|---|---|---|----------|
| 0 | 0 | 0 | 0        |
| 0 | 0 | 1 | 1        |
| 0 | 1 | 0 | 0        |
| 0 | 1 | 1 | 1        |
| 1 | 0 | 0 | 1        |
| 1 | 0 | 1 | 0        |
| 1 | 1 | 0 | 1        |
| 1 | 1 | 1 | 1        |

En la tabla siguiente se muestra la tabla de verdad para  $f(X_1, X_2, \dots, X_n)$

| $X_1$ | $X_2$ | $\dots\dots\dots X_n$ | $f(X_1, X_2, \dots, X_n)$ |
|-------|-------|-----------------------|---------------------------|
| 0     | 0     | $\dots\dots\dots 0$   | $a_0$                     |
| 0     | 0     | $\dots\dots\dots 1$   | $a_1$                     |
|       |       | $\vdots$              | $\vdots$                  |
| 1     | 1     | $\dots\dots\dots 0$   | $a_{2^n-2}$               |
| 1     | 1     | $\dots\dots\dots 1$   | $a_{2^n-1}$               |

Como hay  $n$  variables y cada variable tiene dos valores posibles, hay  $2^n$  formas de asignar valores a las  $n$  variables; por lo tanto, la tabla de verdad tendrá  $2^n$  files.

Además, para cualquier combinación de las variables  $X_1, X_2, \dots, X_n$ , hay dos valores posibles para la función general  $f(X_1, X_2, X_3, \dots, X_n)$ ; por lo tanto podemos hacer  $2^n$  tablas de verdad para  $n$  variables, donde  $N = 2^n$ .

Es decir, para  $n$  variables hay  $2^{2^n}$  funciones Booleanas posibles.

Aún para un número pequeño de variables, el número de funciones Booleanas posibles es enorme, como lo demuestra la tabla siguiente:

| $n$      | $2^n$    | $2^{2^n}$ |
|----------|----------|-----------|
| 0        | 1        | 2         |
| 1        | 2        | 4         |
| 2        | 4        | 16        |
| 3        | 8        | 256       |
| 4        | 16       | 65.536    |
| $\vdots$ | $\vdots$ | $\vdots$  |
| $\vdots$ | $\vdots$ | $\vdots$  |

A continuación se dan las tablas para funciones de 0, 1 y 2 variables:

$$n = 0: \quad \begin{aligned} f_1 &= 0 \\ f_2 &= 1 \end{aligned}$$

| n = 1: | A | f <sub>1</sub> (A) | f <sub>2</sub> (A) | f <sub>3</sub> (A) | f <sub>4</sub> (A) |
|--------|---|--------------------|--------------------|--------------------|--------------------|
|        | 0 | 0                  | 1                  | 0                  | 1                  |
|        | 1 | 0                  | 0                  | 1                  | 1                  |

$$\begin{aligned} f_1(A) &= 0 \\ f_2(A) &= \bar{A} \\ f_3(A) &= A \\ f_4(A) &= 1 \end{aligned}$$

| A | B | f <sub>0</sub> | f <sub>1</sub> | f <sub>2</sub> | f <sub>3</sub> | f <sub>4</sub> | f <sub>5</sub> | f <sub>6</sub> | f <sub>7</sub> | f <sub>8</sub> | f <sub>9</sub> | f <sub>10</sub> | f <sub>11</sub> | f <sub>12</sub> | f <sub>13</sub> | f <sub>14</sub> | f <sub>15</sub> |
|---|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 0 | 0 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 1              | 1               | 1               | 1               | 1               | 1               | 1               |
| 0 | 1 | 0              | 0              | 0              | 0              | 1              | 1              | 1              | 1              | 0              | 0              | 0               | 0               | 1               | 1               | 1               | 1               |
| 1 | 0 | 0              | 0              | 1              | 1              | 0              | 0              | 1              | 1              | 0              | 0              | 1               | 1               | 0               | 0               | 1               | 1               |
| 1 | 1 | 0              | 1              | 0              | 1              | 0              | 1              | 0              | 1              | 0              | 1              | 0               | 1               | 0               | 1               | 0               | 1               |

|   |                 |
|---|-----------------|
| $f_0(A,B) = 0$  | = CERO          |
| $f_1(A,B) = AB$   | = AND           |
| $f_2(A,B) = \overline{AB}$  | = INHIBIDORA    |
| $f_3(A,B) = \overline{AB} + AB = A$   | = IDENTIDAD A   |
| $f_4(A,B) = \overline{AB}$  | = INHIBIDORA    |
| $f_5(A,B) = \overline{AB} + AB = B$   | = IDENTIDAD B   |
| $f_6(A,B) = \overline{AB} + \overline{AB} = A + B$                                | = OR EXCLUSIVO  |
| $f_7(A,B) = \overline{AB} + \overline{AB} + AB = A+B$                             | = OR            |
| $f_8(A,B) = \overline{AB} = A + B$  | = NOR           |
| $f_9(A,B) = \overline{AB} + AB = A \cdot B$                                       | = NOR EXCLUSIVO |
| $f_{10}(A,B) = \overline{AB} + \overline{AB} = \bar{B}$                           | = NOT B         |
| $f_{11}(A,B) = \overline{AB} + \overline{AB} + \overline{AB} = A + \bar{B}$       | = IMPLICACION   |
| $f_{12}(A,B) = \overline{AB} + \overline{AB} = \bar{A}$                           | = NOT A         |
| $f_{13}(A,B) = \overline{AB} + \overline{AB} + AB = \bar{A} + B$                  | = IMPLICACION   |
| $f_{14}(A,B) = \overline{AB} + \overline{AB} + \overline{AB} = A + B = A \cdot B$ | = NAND          |
| $f_{15}(A,B) = \overline{AB} + \overline{AB} + \overline{AB} + \overline{AB} = 1$ | = UNO           |

Anteriormente vimos como se podían evaluar funciones Booleanas usando el álgebra de Boole. Una variación de esa técnica es efectuar el álgebra directamente en la tabla de verdad.

Ejemplo:  $f(A,B, C) = AB + \bar{A}C + A\bar{C}$

La tabla de verdad será:

| A | B | C | $\bar{A}$ | $\bar{C}$ | AB | $\bar{A}C$ | $\bar{A}\bar{C}$ | $AB + \bar{A}C + \bar{A}\bar{C}$ |
|---|---|---|-----------|-----------|----|------------|------------------|----------------------------------|
| 0 | 0 | 0 | 1         | 1         | 0  | 0          | 0                | 0                                |
| 0 | 0 | 1 | 1         | 0         | 0  | 1          | 0                | 1                                |
| 0 | 1 | 0 | 1         | 1         | 0  | 0          | 0                | 0                                |
| 0 | 1 | 1 | 1         | 0         | 0  | 1          | 0                | 1                                |
| 1 | 0 | 0 | 0         | 1         | 0  | 0          | 1                | 1                                |
| 1 | 0 | 1 | 0         | 0         | 0  | 0          | 0                | 0                                |
| 1 | 1 | 0 | 0         | 1         | 1  | 0          | 1                | 1                                |
| 1 | 1 | 1 | 0         | 0         | 1  | 0          | 0                | 1                                |

Ejemplo:

Un campesino tiene un ayudante no muy inteligente' El ayudante está encargado de cuidar que una oveja no se meta a un granero cuando la puerta está abierta, ya que en el se guarda maíz. Además, hay un coyote en los alrededores, con intenciones de comerse a la oveja.

El campesino decide diseñar una caja con tres interruptores:

- uno para "puerta abierta"
- uno para "oveja cercana al granero"
- uno para "coyote a la vista"

El ayudante deberá cerrar los interruptores de acuerdo a lo que esté sucediendo en un momento determinado. Si la combinación de sucesos resulta "peligrosa", los interruptores harán sonar una alarma, con la cual el campesino tomará las medidas adecuadas.

Se supone que:

1. La situación no es peligrosa cuando el coyote y la oveja no están a la vista.
2. Al coyote no le gusta el maíz

Asignemos letras a las variables:

P = Puerta abierta  
 C = Oveja a la vista  
 S = Coyote a la vista

| P | C | S | Situaciones peligrosas |
|---|---|---|------------------------|
| 0 | 0 | 0 | 0                      |
| 0 | 0 | 1 | 0                      |
| 0 | 1 | 0 | 0                      |
| 0 | 1 | 1 | 1                      |
| 1 | 0 | 0 | 0                      |
| 1 | 0 | 1 | 0                      |
| 1 | 1 | 0 | 1                      |
| 1 | 1 | 1 | 1                      |

$$\begin{aligned}
 f &= \bar{P}CS + PC\bar{S} + PCS && \text{Post. 9} \\
 f &= \bar{P}CS + PC(\bar{S} + S) && \text{Post. 10} \\
 f &= \bar{P}CS + PC && \text{Post. 9} \\
 f &= C(P + \bar{P}S) && \text{Teor. 10} \\
 f &= CP + CS && \text{Post. 9}
 \end{aligned}$$

#### 2.6.4.- Simplificación de Funciones Booleanas:

Para cualquier función Booleana, podemos intentar la reducción o simplificación de la función para satisfacer algún criterio. Por lo general dicho criterio es el disminuir el número de componentes. Esto se traduce en la reducción del número de términos en la expresión representando una función booleana. Un término se define como cada ocurrencia de una variable a su complemento en una expresión booleana.

$$\begin{aligned}
 \text{Ejemplo: } f(X, Y, Z) &= \bar{X}Y(Z + \bar{Y}X) + \bar{Y}Z && \text{siete términos} \\
 g(A, B, C) &= \bar{A}\bar{B} + \bar{A}B + AC && \text{seis términos}
 \end{aligned}$$

La simplificación de funciones se efectúa aplicando los postulados y teoremas del álgebra de Boole.

$$\begin{aligned}
 \text{Ejemplo: } f(X, Y, Z) &= \bar{X}Y(Z + \bar{Y}X) + \bar{Y}Z \\
 &= \bar{X}YZ + \bar{X}Y\bar{Y}X + \bar{Y}Z \\
 &= \bar{X}YZ + \bar{Y}Z \\
 &= Z(\bar{Y} + \bar{Y}X) \\
 &= XYZ
 \end{aligned}$$

$$\begin{aligned}
 f(A, B, C, D) &= ABC + ABD + \bar{A}B\bar{C} + CD + B\bar{D} \\
 &= ABC + \bar{A}D\bar{C} + CD + B(\bar{D} + AD) \\
 &= ABC + \bar{A}B\bar{C} + CD + B(\bar{D} + A) \\
 &= ABC + \bar{A}B\bar{C} + CD + AB + B\bar{D} \\
 &= (\bar{A}\bar{C} + A)B + ABC + CD + B\bar{D} \\
 &= (\bar{C} + A)B + ABC + CD + B\bar{D} \\
 &= B\bar{C} + AB + ABC + CD + B\bar{D} \\
 &= AB(1 + AC) + \bar{D}C + CD + B\bar{D} \\
 &= AB + CD + B(\bar{C} + \bar{D}) \\
 &= AB + CD + B\bar{C}D \\
 &= AB + CD + B \\
 &= B(1 + A) + CD \\
 &= B + CD
 \end{aligned}$$

Comprobación mediante tabla de verdad:

| A | B | C | D | A | C | D | ABC | ABD | ABC | CD | BD | f(A,B,C,D) |       |  |
|---|---|---|---|---|---|---|-----|-----|-----|----|----|------------|-------|--|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0   | 0   | 0   | 0  | 0  | 0          |       |  |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0   | 0   | 0   | 0  | 0  | 0          |       |  |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0   | 0   | 0   | 0  | 0  | 0          |       |  |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0   | 0   | 0   | 1  | 0  | 1          | - CD  |  |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0   | 0   | 1   | 0  | 1  | 1          | } - B |  |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0   | 0   | 1   | 0  | 0  | 1          |       |  |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0   | 0   | 0   | 0  | 1  | 1          |       |  |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0   | 0   | 0   | 1  | 0  | 1          |       |  |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0   | 0   | 0   | 0  | 0  | 0          |       |  |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0   | 0   | 0   | 0  | 0  | 0          |       |  |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0   | 0   | 0   | 0  | 0  | 0          |       |  |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0   | 0   | 0   | 1  | 0  | 1          | - CD  |  |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0   | 0   | 0   | 0  | 1  | 1          | } - B |  |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0   | 1   | 0   | 0  | 0  | 1          |       |  |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1   | 0   | 0   | 0  | 1  | 1          |       |  |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1   | 1   | 0   | 1  | 0  | 1          |       |  |

## 2.8.- Formas Algebraicas de Funciones Combinacionales:

Cualquier función booleana se puede expresar en una de dos formas estándar o canónicas. Estas formas, que serán de utilidad en procesos de simplificación a discutirse posteriormente, conducen a expresiones que pueden ser implementadas mediante circuitos de dos niveles. No existe una teoría general que conduzca a una expresión minimizada para circuitos de n niveles cuando  $n > 2$ .

### 2.8.1.- Formas Suma de productos y Productos de sumas:

Las funciones en forma de Suma de Productos SP se construyen mediante la disyunción de términos productos y los términos productos se obtienen mediante la conjunción de variables complementadas y no complementadas.

Ejemplos:

a) Términos productos:  $\frac{A\bar{B}C}{B\bar{D}}$

b) Función en forma SP

$$f = A\bar{B}C + \bar{B}\bar{D} + \bar{A}C\bar{D}$$

Las funciones Productos de Sumas PS se construyen mediante la conjunción de términos sumas y los términos sumas se obtienen mediante la disyunción de variables complementadas y sin complementar.



$$\begin{array}{ll}
 f = \bar{P}CS + PC\bar{S} + PCS & \\
 f = \bar{P}CS + PC(\bar{S} + S) & \text{Post. 9} \\
 f = \bar{P}CS + PC & \text{Post. 10} \\
 f = C(P + \bar{P}S) & \text{Post. 9} \\
 f = C(P + S) & \text{Teor. 10} \\
 f = CP + CS & \text{Post. 9}
 \end{array}$$

#### 2.6.4.- Simplificación de Funciones Booleanas:

Para cualquier función Booleana, podemos intentar la reducción o simplificación de la función para satisfacer algún criterio. Por lo general dicho criterio es el disminuir el número de componentes. Esto se traduce en la reducción del número de términos en la expresión representando una función booleana. Un término se define como cada ocurrencia de una variable a su complemento en una expresión booleana.

$$\begin{array}{ll}
 \text{Ejemplo: } f(X, Y, Z) = \bar{X}Y(Z + \bar{Y}X) + \bar{Y}Z & \text{siete términos} \\
 g(A, B, C) = \bar{A}\bar{B} + \bar{A}B + AC & \text{seis términos}
 \end{array}$$

La simplificación de funciones se efectúa aplicando los postulados y teoremas del álgebra de Boole.

$$\begin{array}{l}
 \text{Ejemplo: } f(X, Y, Z) = \bar{X}Y(Z + \bar{Y}X) + \bar{Y}Z \\
 = \bar{X}YZ + \bar{X}Y\bar{Y}X + \bar{Y}Z \\
 = \bar{X}YZ + \bar{Y}Z \\
 = Z(\bar{Y} + \bar{Y}X) \\
 = \bar{X}YZ
 \end{array}$$

$$\begin{array}{l}
 f(A, B, C, D) = ABC + ABD + \bar{A}\bar{B}\bar{C} + CD + B\bar{D} \\
 = ABC + \bar{A}\bar{D}\bar{C} + CD + B(\bar{D} + AD) \\
 = ABC + \bar{A}\bar{B}\bar{C} + CD + B(\bar{D} + A) \\
 = ABC + \bar{A}\bar{B}\bar{C} + CD + AB + B\bar{D} \\
 = (\bar{A}\bar{C} + A)B + ABC + CD + B\bar{D} \\
 = (\bar{C} + A)B + ABC + CD + B\bar{D} \\
 = B\bar{C} + AB + ABC + CD + B\bar{D} \\
 = AB(1 + AC) + \bar{D}\bar{C} + CD + B\bar{D} \\
 = AB + CD + B(\bar{C} + \bar{D}) \\
 = AB + CD + B\bar{C}\bar{D} \\
 = AB + CD + B \\
 = B(1 + A) + CD \\
 = B + CD
 \end{array}$$

Comprobación mediante tabla de verdad:

$$\begin{aligned} f(A,B,C) &= m_2 + m_6 + m_3 + m_7 \\ &= m_2 + m_3 + m_6 + m_7 \end{aligned}$$

Un paso más en la simplificación, es escribir la función en la forma de lista de mintérminos

$$f(A,B,C) = \sum m(2, 3, 6, 7)$$

El orden en que se escriban las variables en la notación funcional es muy importante, ya que dicho orden se emplea para de codificar los mintérminos:

$$\begin{aligned} \text{Ejemplo: } f(BCA) &= \sum m(2,3,6,7) \\ &= m_2 + m_3 + m_6 + m_7 \\ &\quad 010 \quad 011 \quad 110 \quad 111 \\ &= BC\bar{A} + \bar{B}CA + BC\bar{A} + BCA \\ &= \bar{A}\bar{B}C + A\bar{B}C + \bar{A}BC + ABC \end{aligned}$$

Obsérvese que esta ecuación no es igual a la del ejemplo anterior, aún cuando las listas de mintérminos son iguales.

Consideremos la función:

$$\begin{aligned} f(A,B,C) &= \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC \\ &\quad 001 \quad 011 \quad 101 \quad 111 \\ &= m_1 + m_3 + m_5 + m_7 \\ &= \sum m(1,3,5,7) \end{aligned}$$

construyamos su tabla de verdad:

| Fila No. | A B C | m                 |                |                |                | f(A,B,C) |
|----------|-------|-------------------|----------------|----------------|----------------|----------|
|          |       | m <sub>1</sub>    | m <sub>3</sub> | m <sub>5</sub> | m <sub>7</sub> |          |
|          |       | $\bar{A}\bar{B}C$ | $\bar{A}BC$    | $A\bar{B}C$    | $ABC$          |          |
| 0        | 0 0 0 | 0                 | 0              | 0              | 0              | 0        |
| 1        | 0 0 1 | 1                 | 0              | 0              | 0              | 1        |
| 2        | 0 1 0 | 0                 | 0              | 0              | 0              | 0        |
| 3        | 0 1 1 | 0                 | 1              | 0              | 0              | 1        |
| 4        | 1 0 0 | 0                 | 0              | 0              | 0              | 0        |
| 5        | 1 0 1 | 0                 | 0              | 1              | 0              | 1        |
| 6        | 1 1 0 | 0                 | 0              | 0              | 0              | 0        |
| 7        | 1 1 1 | 0                 | 0              | 0              | 1              | 1        |

Obsérvese en la tabla que cada fila está numerada de acuerdo

al código decimal, y que los únicos 1 que aparecen en la tabla son aquellos en la fila  $i$ , producidos por el mintermino  $m$ .

Por lo tanto, podemos eliminar todos los pasos intermedios y escribir la tabla de verdad directamente de la lista de minterminos.

Ejemplo:

| Fila No. | A B C | $f(A,B,C)$ |
|----------|-------|------------|
| 0        | 0 0 0 | 0          |
| 1        | 0 0 1 | 0          |
| 2        | 0 1 0 | 1          |
| 3        | 0 1 1 | 1          |
| 4        | 1 0 0 | 0          |
| 5        | 1 0 1 | 0          |
| 6        | 1 1 0 | 1          |
| 7        | 1 1 1 | 1          |

$f(A,B,C) = \sum m(2,3,6,7)$

Otro aspecto importante es observar la tabla de verdad para  $\bar{f}(A,B,C)$

| Fila No. | A B C | $f(A,B,C)$ | $\bar{f}(A,B,C)$ |
|----------|-------|------------|------------------|
| 0        | 0 0 0 | 0          | 1                |
| 1        | 0 0 1 | 0          | 1                |
| 2        | 0 1 0 | 1          | 0                |
| 3        | 0 1 1 | 1          | 0                |
| 4        | 1 0 0 | 0          | 1                |
| 5        | 1 0 1 | 0          | 1                |
| 6        | 1 1 0 | 1          | 0                |
| 7        | 1 1 1 | 1          | 0                |

$\bar{f}(A,B,C) = \sum m(0,1,4,5)$

La tabla indica que  $\bar{f}(A,B,C)$  tiene 1 en las filas 0,1,4, y 5, y por lo tanto:

$$\bar{f}(A,B,C) = \sum m(0,1,4,5)$$

y

$$f(A,B,C) = \sum m(2,3,6,7)$$

Obsérvese que todos los minterminos compuestos de tres variables (8 en total) están contenidos en una de las dos listas de

mintérminos:  $f(A,B,C)$  o  $\bar{f}(A,B,C)$ .

En general, los  $2^n$  mintérminos de  $n$  variables aparecerán siempre en la forma canónica de SP para  $f(X_1, X_2, X_3, \dots, X_n)$  o  $\bar{f}(X_1, X_2, \dots, X_n)$ .

Por ejemplo, si:

$$f(A,B,C,D) = \sum m(0,1,6,7)$$

el complemento tendrá  $2^4 - 4 = 12$

$$\begin{aligned} \bar{f}(A,B,C,D) &= \sum m(2,3,4,5,8,9,10,11,12,13,14,15) \\ &= \sum m(2-5, 8-15) \end{aligned}$$

Finalmente, del álgebra de Boole

$$f(X_1 X_2 \dots X_n) + \bar{f}(X_1 X_2 \dots X_n) = 1$$

Pero como

$$f(X_1 X_2 \dots X_n) + \bar{f}(X_1 X_2 \dots X_n) = \sum_{i=0}^{2^n-1} m_i$$

$$\text{luego: } \sum_{i=0}^{2^n-1} m_i = 1$$

En otras palabras la disyunción de todos los mintérminos de  $n$  variables es igual a 1.

Maxtérminos: Si un término suma de una función de  $n$  variables contine las  $n$  variables en forma complementada o sin complementar, dicho término-suma se llama maxtérmino.

Si una función está compuesta de productos de términos-suma, siendo cada uno de los cuales un maxtérmino, se dice que la función está en forma canónica de suma de producto.

$$\text{Ejemplo: } f_1(A_1 B_1 C) = (A+B+C)(A+B+\bar{C})(\bar{A}+B+C)(A+B+\bar{C})$$

$f_1$  es una función en forma canónica con tres variables y cuatro maxtérminos.

Tal como para el caso de los mintérminos, existe una notación especial para maxtérminos.

Variables sin complementar: 0  
 Variables complementadas: 1

Con este código, los maxtérminos de la función  $f_1$  del problema anterior quedan:

|                         | Código<br>Maxtérmino | Lista |
|-------------------------|----------------------|-------|
| $A + B + C$             | 0 0 0                | $M_0$ |
| $A + B + \bar{C}$       | 0 0 1                | $M_1$ |
| $\bar{A} + B + C$       | 1 0 0                | $M_4$ |
| $\bar{A} + B + \bar{C}$ | 1 0 1                | $M_5$ |

Los maxtérminos se abrevian como  $M_i$ , donde  $i$  es el decimal entero correspondiente al código binario del maxtérmino. Luego, la función  $f_1$  del ejemplo quedaría:

$$f_1(A, B, C) = M_0 M_1 M_4 M_5$$

O bien, escribiéndola en forma de lista de maxtérminos:

$$f_1(A_1 B_1 C) = \overline{11} M(0, 1, 4, 5)$$

Las dos últimas expresiones para  $f_1$  están en forma canónica de suma de productos.

Tal como en el caso de mintérminos, el orden de las variables en la notación funcional es muy importante.

La tabla de verdad para la función  $f_1$  anterior es:

| Fila No. | A+B+C | $M_0$         | $M_1$         | $M_4$               | $M_5$ | $f_1(A_1 B_1 C)$ |
|----------|-------|---------------|---------------|---------------------|-------|------------------|
|          |       | $A+B+\bar{C}$ | $\bar{A}+B+C$ | $\bar{A}+B+\bar{C}$ | A+B+C |                  |
| 0        | 0 0 0 | 0             | 1             | 1                   | 1     | 0                |
| 1        | 0 0 1 | 1             | 0             | 1                   | 1     | 0                |
| 2        | 0 1 0 | 1             | 1             | 1                   | 1     | 1                |
| 3        | 0 1 1 | 1             | 1             | 1                   | 1     | 1                |
| 4        | 1 0 0 | 1             | 1             | 0                   | 1     | 0                |
| 5        | 1 0 1 | 1             | 1             | 1                   | 0     | 0                |
| 6        | 1 1 0 | 1             | 1             | 1                   | 1     | 1                |
| 7        | 1 1 1 | 1             | 1             | 1                   | 1     | 1                |

Obsérvese que los únicos ceros que aparecen en la tabla, están en la fila  $i$  y son producidos por el maxtérmino  $M_i$ . Por lo tanto, tal como en el caso de los mintérminos, la tabla de verdad puede ser generada por inspección de la lista de maxtérminos.

Examinemos la función siguiente:

$$f(A,B,C) = \underbrace{(A+B+\bar{C})}_{001} \underbrace{(A+\bar{B}+\bar{C})}_{011} \underbrace{(\bar{A}+B+\bar{C})}_{101} \underbrace{(\bar{A}+\bar{B}+C)}_{111}$$

$$f(A,B,C) = M_1 M_3 M_5 M_7$$

$$f(A,B,C) = \Pi M(1,3,5,7)$$

Los maxtérminos de la función ubican los ceros en las filas 1, 3, 5 y 7 de la tabla de verdad.

| Fila | A | B | C | F(A,B,C) |
|------|---|---|---|----------|
| 0    | 0 | 0 | 0 | 1        |
| 1    | 0 | 0 | 1 | 0        |
| 2    | 0 | 1 | 0 | 1        |
| 3    | 0 | 1 | 1 | 0        |
| 4    | 1 | 0 | 0 | 1        |
| 5    | 1 | 0 | 1 | 0        |
| 6    | 1 | 1 | 0 | 1        |
| 7    | 1 | 1 | 1 | 0        |

Observando la tabla de verdad, vemos que  $f(A,B,C) = \sum m(0,2,4,6)$ .

Luego

$$\begin{aligned} \bar{f}(A,B,C) &= \sum m(1,3,5,7) \\ &= m_1 + m_3 + m_5 + m_7 \\ &\quad 001 \quad 011 \quad 101 \quad 111 \\ &= \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC \end{aligned}$$

Consecuentemente,

$$\begin{aligned} f(A,B,C) &= \overline{\bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC} \\ &= \overline{\bar{A}\bar{B}C} \cdot \overline{\bar{A}BC} \cdot \overline{A\bar{B}C} \cdot \overline{ABC} \\ &= \underbrace{(A+B+\bar{C})}_{001} \underbrace{(A+\bar{B}+\bar{C})}_{011} \underbrace{(\bar{A}+B+\bar{C})}_{101} \underbrace{(\bar{A}+\bar{B}+C)}_{111} \\ &= M_1 M_3 M_5 M_7 \\ &= \Pi M(1,3,5,7) \end{aligned}$$

Es decir, hemos demostrado que:

$$f(A,B,C) = \Pi M(1,3,5,7) = \sum m(0,2,4,6)$$

lo cual es evidente al inspeccionar la tabla de verdad.

En las manipulaciones algebraicas del ejemplo anterior, resultaron aparentes ciertas relaciones entre mintérminos y maxtérminos

$$\overline{m_1} = \overline{\overline{A} \overline{B} C} = \underbrace{A + B + \overline{C}}_{0 \ 0 \ 1} = M_1$$

$$\overline{m_3} = \overline{\overline{A} B C} = \underbrace{A + \overline{B} + \overline{C}}_{0 \ 1 \ 1} = M_3$$

En general:  $\overline{m_i} = M_i$

$$\overline{M_i} = m_i$$

y por lo tanto los mintérminos y maxtérminos: son complementos el uno del otro.

Observemos la tabla de verdad del complemento de la función del ejemplo anterior:

$$f(A,B,C) = \Pi M(1,3,5,7)$$

| Fila | A | B | C | f(A,B,C) | $\overline{f}(A,B,C)$ |
|------|---|---|---|----------|-----------------------|
| 0    | 0 | 0 | 0 | 1        | 0                     |
| 1    | 0 | 0 | 1 | 0        | 1                     |
| 2    | 0 | 1 | 0 | 1        | 0                     |
| 3    | 0 | 1 | 1 | 0        | 1                     |
| 4    | 1 | 0 | 0 | 1        | 0                     |
| 5    | 1 | 0 | 1 | 0        | 1                     |
| 6    | 1 | 1 | 0 | 1        | 0                     |
| 7    | 1 | 1 | 1 | 0        | 1                     |

de la tabla vemos que los ceros de  $\overline{f}(A,B,C)$  están en las filas 0, 2, 4 y 6. Luego:

$$\overline{f}(A,B,C) = \Pi M(0,2,4,6)$$

y

$$f(A,B,C) = \Pi M(1,3,5,7)$$

es decir todos los maxtérminos generados por tres variables, aparecen en  $f(A,B,C)$  o en  $\bar{f}(A, B, C)$ . Además, del álgebra de Boole, tenemos que:

$$f(A,B,C) \cdot \bar{f}(A,B,C) = 0$$

luego:

$$(M_0 M_2 M_4 M_6)(M_1 M_3 M_5 M_7) = 0$$

o bien

$$\prod_{i=0}^{2^3-1} M_i = 0$$

En general, para  $n$  variables tenemos:

$$\prod_{i=0}^{2^n-1} M_i = 0$$

Resumen:

Resumiremos los resultados obtenidos mediante la función - -  
 $f(A,B,C) = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$$f(A,B,C) = \sum m(2,3,6,7) = \Pi M(0,1,4,5)$$

$$\bar{f}(A,B,C) = \sum m(0,1,4,5) = \Pi M(2,3,6,7)$$

El significado de lo anterior, se ilustra mediante la función siguiente:

$$f(W,X,Y,Z) = \Pi M(0,4 - 8, 10, 12, 15)$$

La función es de cuatro variables,  $W$ ,  $X$ ,  $Y$  y  $Z$  y está especificada en forma canónica de productos de sumas o en forma de lista de maxtérminos.

Con esta información, podemos escribir la función en forma de lista de mintérminos

$$\begin{aligned} f(W,X,Y,Z) &= \Pi M(0,4 - 8, 10, 12, 15) \\ &= \sum m(1,2,3,9,11,13,14) \end{aligned}$$



$$\begin{aligned}\bar{f}(W, X, Y, Z) &= \prod M(1, 2, 3, 9, 11, 13, 14) \\ &= \sum m(0, 4 - 8, 10, 12, 15)\end{aligned}$$

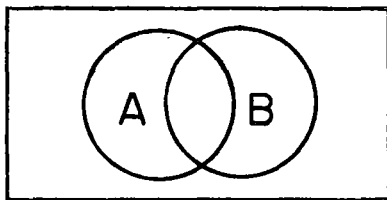
## 2.9.- Mapas de Karnough:

Anteriormente vimos que es posible simplificar funciones booleanas ya sea empleando los teoremas del álgebra de Boole, o a partir de la tabla de verdad de la función. El empleo de la tabla de verdad, es un procedimiento gráfico que puede resultar poco eficiente. Sin embargo, si la tabla de verdad es modificada un tanto, puede obtenerse una representación gráfica de la función que resulta muy útil para efectos de minimización. Esta forma es el mapa de Karnough.

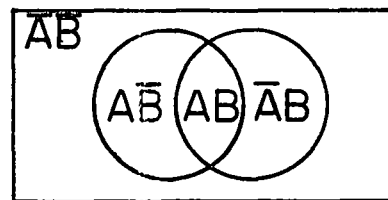
El mapa de Karnough no es más que una extensión de los conceptos de: tablas de verdad, diagramas de Venn y minterminos.

Para hacer explícita esta extensión, transformemos un diagrama de Venn en un mapa de Karnough.

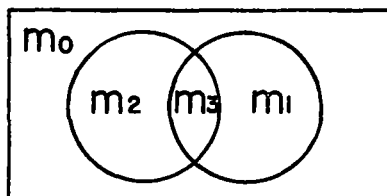
Consideremos el diagrama de Venn de la figura (a) siguiente. Las dos variables A y B se representan mediante subdivisiones del conjunto universal. En la figura (b) se ilustra el hecho que cada subdivisión disjunta del diagrama de Venn está formada por las intersecciones  $AB$ ,  $\bar{A}\bar{B}$ ,  $A\bar{B}$ ,  $\bar{A}B$ , las que son los minterminos de dos variables y por lo tanto podemos redesignar las



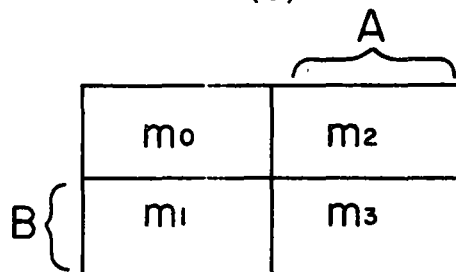
(a)



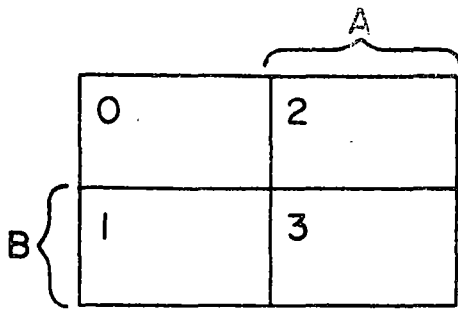
(b)



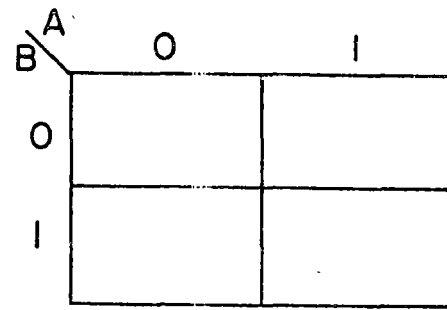
(c)



(d)



(e)



(f)

las subdivisiones del diagrama de Venn como los mintérminos  $m_0$ ,  $m_1$ ,  $m_2$  y  $m_3$ , lo cual se ilustra en la figura (c). En la figura (d) se ha redibujado el diagrama de Venn de la figura (c), con las áreas de cada subdivisión iguales. Obsérvese que las áreas adyacentes del diagrama de Venn son también áreas adyacentes en la figura (d). Sin embargo, en la figura (d), una mitad del diagrama representa a la variable A y otra mitad representa a la variable B. Como la notación de mintérminos se identifica con cada cuadrado del diagrama, podemos omitir la letra m y dejar solamente el subíndice, como se muestra en la figura (e). Esta figura muestra una forma del mapa de Karnaugh. En la figura (f) se muestra una segunda forma del mapa, que es la más usada.

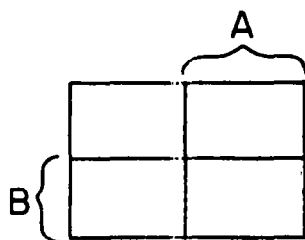
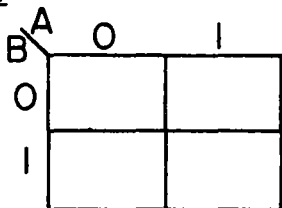
Es importante observar que el mapa de Karnaugh es una representación gráfica de una tabla de verdad y existe por lo tanto una correspondencia uno a uno entre ambas. La tabla de verdad tiene una fila para cada mintérmino, mientras el mapa de Karnaugh tiene un cuadrado para cada mintérmino.

Los requisitos que debe satisfacer un mapa de Karnaugh son los siguientes: 1) debe haber un cuadrado para cada combinación de variables; es decir deber haber  $2^n$  cuadrados para n variables 2) Los cuadrados deben estar organizados de tal forma que cualquier par de cuadrados inmediatamente adyacentes el uno del otro (horizontal y verticalmente) deben corresponder a condiciones de combinaciones de variables lógicamente adyacentes: es decir, que difieran en una sola variable.

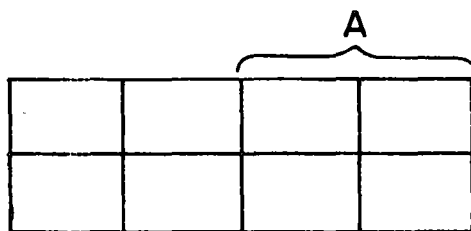
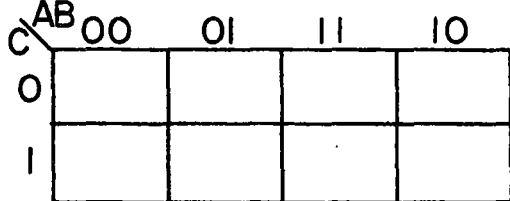
Observemos el paso de la figura (c) a la (d); consideremos el mintérmino  $m_0$  en la figura (c); dicho mintérmino es adyacente a  $m_1$ ,  $m_2$  y  $m_4$ , lo cual satisface el requisito 2) mencionado anteriormente. Sin embargo, en la figura (d),  $m_0$  no está físicamente adyacente a  $m_4$ . Para conciliar esta inconsistencia con el requisito 2) se considera que los extremos izquierdo y derecho del mapa son adyacentes. Es decir, el mapa-K viene a ser un cilindro.

En la serie de figuras siguientes, se encuentran los mapas de Karnough para 2, 3, 4 y 5 variables.

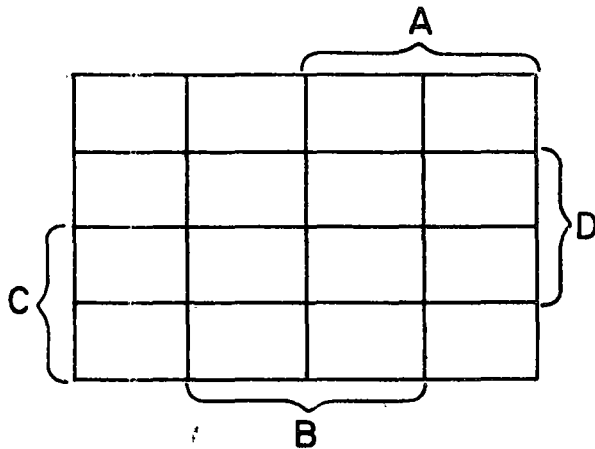
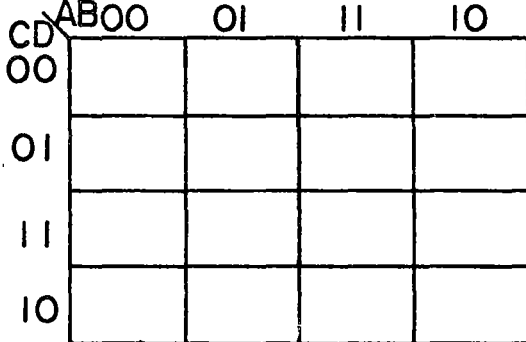
1)  $n = 2$



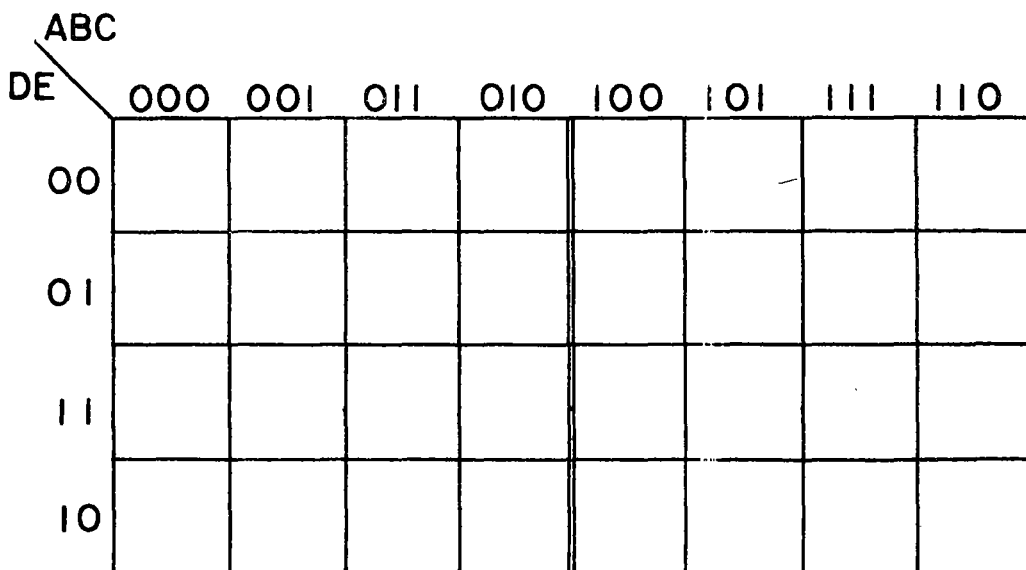
2)  $n = 3$

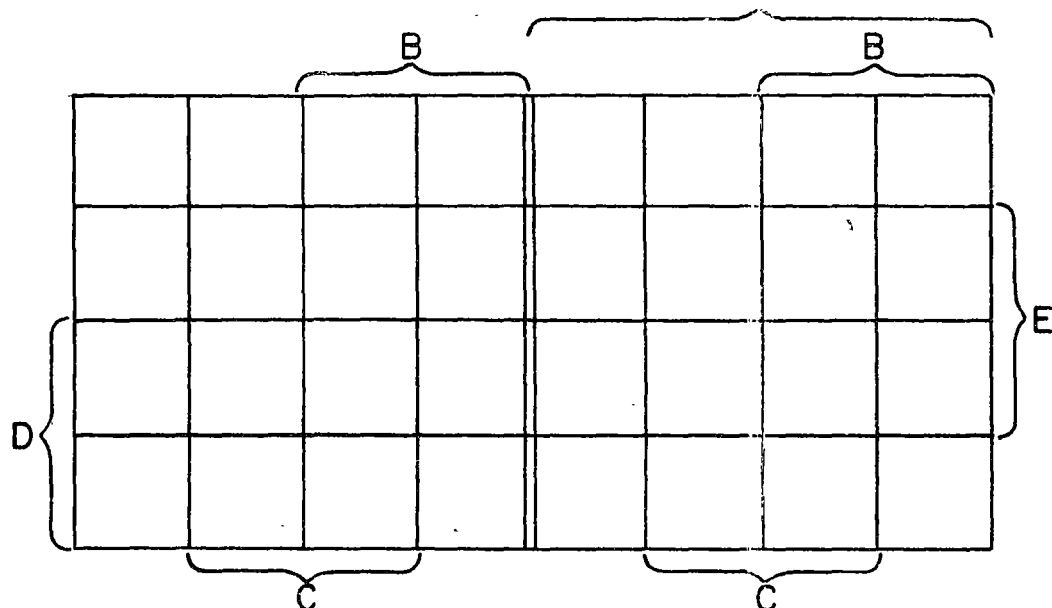


3)  $n = 4$



4)  $n = 5$





### 2.9.1.- Representación de Funciones en Mapas - K:

Una función booleana puede ser representada fácilmente en un mapa-K si se encuentra en una de sus formas canónicas.

- 1) Si la función está representada en forma de lista de minterminos, se ponen 1 en los cuadrados correspondientes a cada mintermino de la lista.
- 2) Si la función está representada en forma de lista de maxtérminos, se ponen 0 en los cuadrados correspondientes a cada maxtérmino de la lista o 1 en los cuadrados correspondientes a maxtérminos que no estén en la lista.

Ejemplos:

- 1) Representar en un mapa-K la función

$$f(A,B,C,D) = \sum m(0,3,5,7,10 - 15)$$

|          |    |    |    |    |
|----------|----|----|----|----|
| AB<br>CD | 00 | 01 | 11 | 10 |
| 00       | 1  |    | 1  |    |
| 01       |    | 1  | 1  |    |
| 11       | 1  | 1  | 1  | 1  |
| 10       |    |    | 1  | 1  |

- 2) Representar en un mapa - K la función

$$f(A,B,C) = \Pi M(0,1,3,6,7)$$

|         |    |    |    |    |
|---------|----|----|----|----|
| AB<br>C | 00 | 01 | 11 | 10 |
| 0       | 0  |    | 0  |    |
| 1       | 0  | 0  | 0  |    |

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

Para el caso de funciones que no estén en forma de lista de minterminos o maxtérminos, se presentan dos casos:

- 1) La función está en forma de suma de productos. En este caso, lo más conveniente es representar directamente la función en un mapa de Karnough; sin expandirla.

Ejemplo:

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

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

- 2) La función está en forma de productos de suma. En este caso, lo más conveniente es pasar a suma de productos y luego representarla en el mapa-K.

### 2.9.2.- Simplificación de Funciones:

La simplificación de una función debe estar orientada hacia la minimización de algún objetivo de diseño. Estos objetivos de diseño pueden ser varios, pero en general hay dos que tienen gran importancia: costo y velocidad.

Estos objetivos de diseño obligan a establecer un compromiso entre ambos ya que invariablemente existen las relaciones:

$$\begin{aligned} \text{Alta velocidad} &\Rightarrow \text{Alto costo} \\ \text{Bajo costo} &\Rightarrow \text{Baja velocidad} \end{aligned}$$

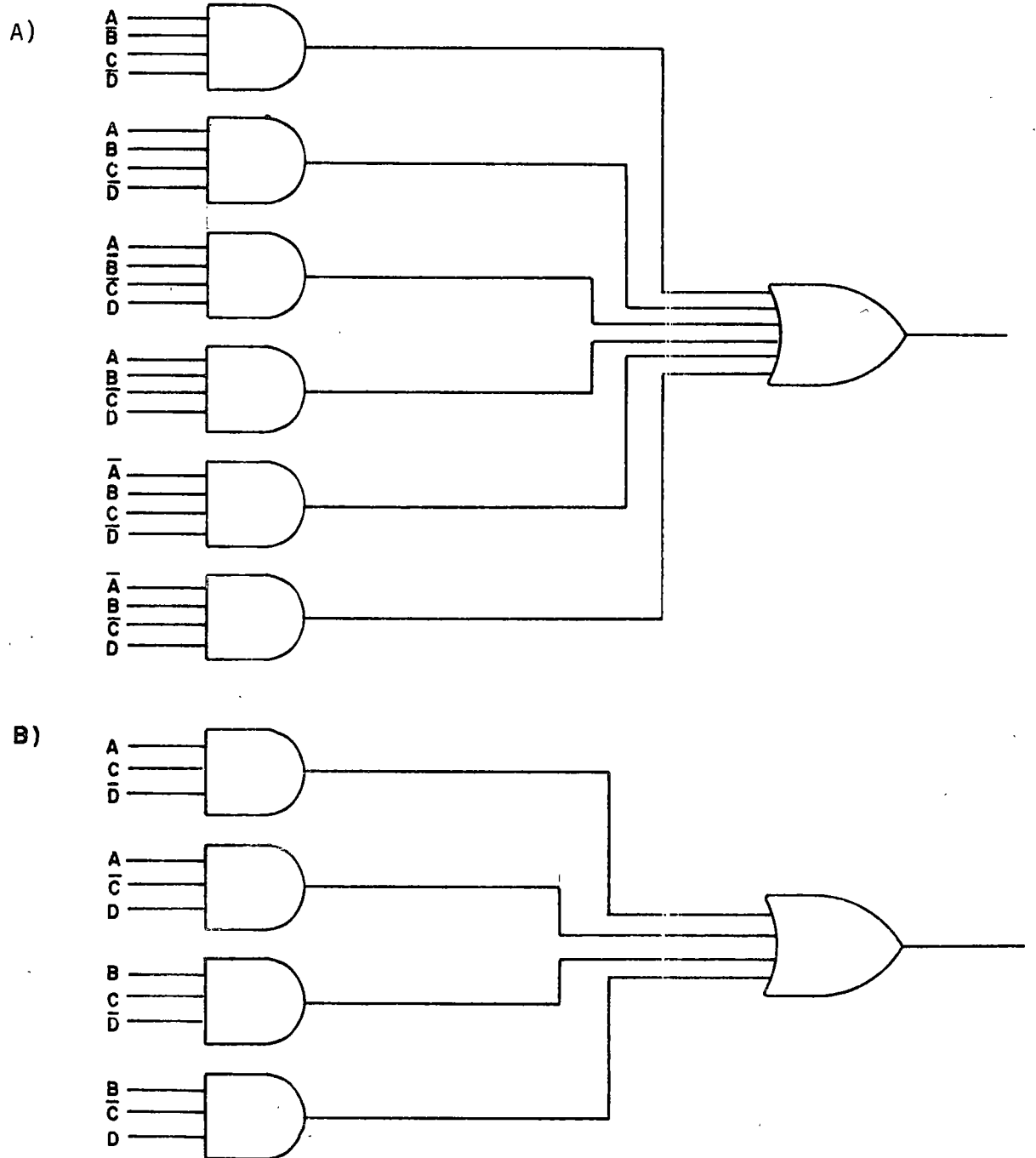
Consideremos el caso de la función:

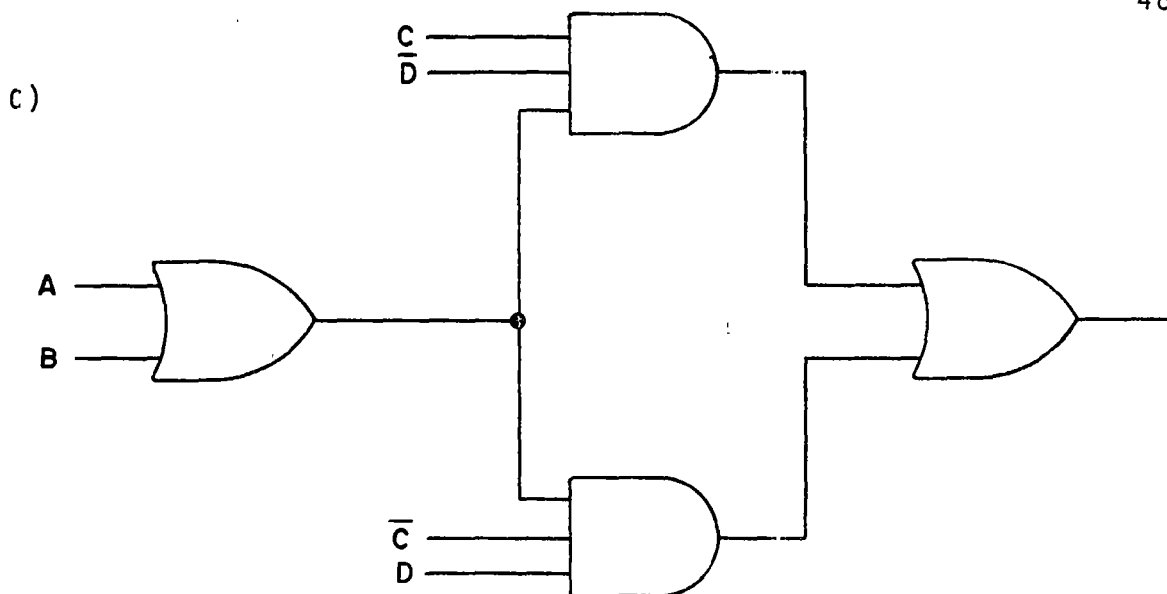
$$f = (A, B, C, D) = \sum m(5, 6, 9, 10, 13, 14) \quad (A)$$

$$= AC\bar{D} + A\bar{C}D + BC\bar{D} + B\bar{C}D \quad (B)$$

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

La realización de cada una de las formas de  $f$  se muestra en la siguiente figura.





Obviamente el circuito A es el más complejo: requiere de 7 compuertas. Comparando los circuitos B y C vemos que el más simple es el C, ya que requiere de 4 compuertas contra 5 del B. Sin embargo, una señal de entrada al circuito C experimentará 3 retardos de tiempo antes de salir, ya que hay 3 niveles de compuertas; en cambio en el B hay sólo dos niveles. Aquí represente el compromiso entre costo y velocidad.

Nuestro criterio de minimización será el de optimizar velocidad. Es decir, trataremos de obtener circuitos de dos niveles de compuerta.

La simplificación mediante mapas-K se basa en el hecho que aquellos conjuntos de minterminos que se puedan combinar en términos-producto más simples, deben ser adyacentes o aparecer en patrones simétricos en el mapa-K.

Los mapas K están ordenados de tal forma que los minterminos en cuadrados adyacentes son idénticos excepto por una variable. Dicha variable aparece en forma sin complementar en un mintermino y complementada en el otro y por lo tanto, el valor de la función será independiente del valor de dicha variable.

Cualquier par de minterminos de  $n$  variables, adyacentes en un mapa-K se puede combinar en un término-producto de  $(n-1)$  variables.

Ejemplo:

$$f(A,B,C,D) = \sum m(2, 3, 7, 10, 11, 12, 13, 15)$$

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

$$f = AB\bar{C} + CD + \bar{B}C$$

Las adyacencias deben ser de filas o de columnas, pero no de diagonales.

Como los conjuntos de dos minterminos se combinan para eliminar una variable y los de cuatro minterminos, para eliminar dos, los de ocho se combinarán para eliminar tres.

Ejemplos:

$$1) f = \sum m(0, 2, 8, 11, 15, 18, 20, 21, 27, 28, 29, 31)$$

|    |   |     |     |     |     |     |     |     |     |
|----|---|-----|-----|-----|-----|-----|-----|-----|-----|
|    |   | ABC |     |     |     |     |     |     |     |
|    |   | DE  |     |     |     |     |     |     |     |
|    |   | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 |
| 00 | 1 |     |     | 1   |     | 1   | 1   |     |     |
| 01 |   |     |     |     |     | 1   | 1   |     |     |
| 11 |   |     |     | 1   | 1   |     |     | 1   | 1   |
| 10 | 1 |     |     |     | 1   |     |     |     |     |

$$2) f(A, B, C, D) = \sum m(0, 1, 3, 8, 9, 11, 13, 14)$$



|          |   |    |    |    |    |
|----------|---|----|----|----|----|
| AB<br>CD |   | 00 | 01 | 11 | 10 |
|          |   | 00 | 01 | 11 | 10 |
| 00       | 1 |    |    | 1  |    |
| 01       | 1 |    | 1  | 1  |    |
| 11       | 1 |    |    | 1  |    |
| 10       |   |    | 1  |    |    |

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

$$3) f = \sum m(1, 2, 3, 6)$$

|         |   |    |    |    |    |
|---------|---|----|----|----|----|
| AB<br>C |   | 00 | 01 | 11 | 10 |
|         |   | 0  | 01 | 11 | 10 |
| 0       |   | 1  | 1  |    |    |
| 1       | 1 | 1  |    |    |    |

$$f = B\bar{C} + \bar{A}C$$

$$4) f = \sum m(0, 1, 2, 7, 8, 9, 10, 15)$$

|          |   |    |    |    |    |
|----------|---|----|----|----|----|
| AB<br>CD |   | 00 | 01 | 11 | 10 |
|          |   | 00 | 01 | 11 | 10 |
| 00       | 1 |    |    | 1  |    |
| 01       | 1 |    |    | 1  |    |
| 11       |   | 1  | 1  |    |    |
| 10       | 1 |    |    | 1  |    |

$$f = \bar{B}\bar{C} + \bar{B}\bar{D} + BCD$$

$$5) f(A, B, C, D) = \sum m(3, 4, 6, 8, 9, 12, 14)$$

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

$$f = \bar{A}\bar{B}CD + A\bar{B}\bar{C} + B\bar{D}$$

- 6) Se requiere producir una salida siempre que ocurra un número entre 4 y 11

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

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

- 7)  $f(A, B, C, D, E, F) = \sum m(2, 3, 6, 7, 10, 14, 18, 19, 22, 23, 27, 37, 42, 43, 45, 46)$

| BCD |     | AEF |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|     |     | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 |
| AEF | 000 |     |     |     |     |     |     |     |     |
|     | 001 |     |     |     |     |     |     |     |     |
|     | 011 | 1   | 1   |     |     | 1   | 1   |     | 1   |
|     | 010 | 1   | 1   | 1   | 1   | 1   | 1   |     |     |
|     | 100 |     |     |     |     |     |     |     |     |
|     | 101 |     | 1   | 1   |     |     |     |     |     |
|     | 111 |     |     |     | 1   |     |     |     |     |
|     | 110 |     |     | 1   | 1   |     |     |     |     |

$$f = \bar{A}\bar{B}\bar{D}FE + A\bar{B}\bar{D}\bar{E}F + A\bar{B}C\bar{D}E + \bar{B}CE\bar{F}$$

### 2.9.3.- Simplificación de Funciones Productos de Suma:

La simplificación de funciones en forma de productos de suma se puede realizar en forma muy simple de la siguiente forma:

1. Representar la función (PS o SP) en el mapa-K.
2. Complementar el mapa. Ahora se tiene  $\bar{f}$ .
3. Minimizar  $\bar{f}$  en forma de SP.
4. Complementar  $\bar{f}$  y obtenemos  $f$  en forma de PS.

Ejemplos:

$$1) f(A,B,C,D) = \Pi M(0,1,2,3,6,9,14)$$

|          |   |    |    |    |    |
|----------|---|----|----|----|----|
| AB<br>CD |   | 00 | 01 | 11 | 10 |
|          |   | 00 | 01 | 11 | 10 |
| 00       | 0 |    |    |    |    |
| 01       | 0 |    |    | 0  |    |
| 11       | 0 |    |    |    |    |
| 10       | 0 | 0  | 0  |    |    |

$$\bar{f} = BC\bar{D} + \bar{B}\bar{C}D + \bar{A}\bar{B}$$

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

$$2) f(A, B, C, D) = \sum m(1, 3, 4, 6, 9, 11, 12, 14)$$

|          |   |    |    |    |    |
|----------|---|----|----|----|----|
| AB<br>CD |   | 00 | 01 | 11 | 10 |
|          |   | 00 | 01 | 11 | 10 |
| 00       | 0 | 1  | 1  | 0  |    |
| 01       | 1 | 0  | 0  | 1  |    |
| 11       | 1 | 0  | 0  | 1  |    |
| 10       | 0 | 1  | 1  | 0  |    |

$$f = (B + D)(\bar{B} + \bar{D})$$

#### 2.9.4.- Funciones Incompletamente Especificadas:

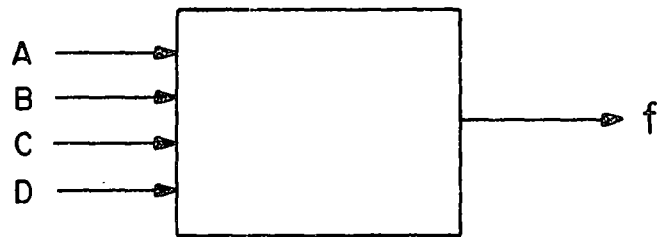
En el diseño de circuitos digitales es frecuente encontrar casos en que la función booleana no está completamente especificada. Es decir, puede darse el caso de una función de la que se requieran ciertos minterminos, se omitan otros y los restantes sean opcionales. Esto puede ser el caso de un circuito que sea una parte de un sistema más grande en el que ciertas entradas ocurrirán bajo circunstancias tales que la salida del circuito no tendrá influencia sobre el sistema en global. También puede ser el caso de que cierta combinación de entradas no se producirá nunca debido a ciertas restricciones externas.

Cuando la salida no tenga efecto sobre el resto del sistema, obviamente no nos importará si ésta es un 0 o un 1. En tal condición, decimos que la salida no está especificada y lo indicamos mediante una X en la tabla de verdad.

Ejemplo:

La entrada de un circuito es la representación en BCD de los números decimales 0-9. El circuito se usa como detector de redondeo, es decir, producirá una salida cuando la entrada sea 5, 6, 7, 8 ó 9.

| ABCD | f | m        |
|------|---|----------|
| 0000 | 0 | $m_0$    |
| 0001 | 0 | $m_1$    |
| 0010 | 0 | $m_2$    |
| 0011 | 0 | $m_3$    |
| 0100 | 0 | $m_4$    |
| 0101 | 1 | $m_5$    |
| 0110 | 1 | $m_6$    |
| 0111 | 1 | $m_7$    |
| 1000 | 1 | $m_8$    |
| 1001 | 1 | $m_9$    |
| 1010 | X | $m_{10}$ |
| 1011 | X | $m_{11}$ |
| 1100 | X | $m_{12}$ |
| 1101 | X | $m_{13}$ |
| 1110 | X | $m_{14}$ |
| 1111 | X | $m_{15}$ |



| AB \ CD | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      |    |    | X  | 1  |
| 01      |    | 1  | X  | 1  |
| 11      |    | 1  | X  | X  |
| 10      |    | 1  | X  | X  |

$$f = A + BC + BD$$

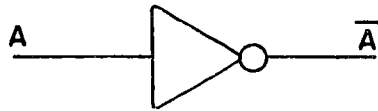
## CAPITULO 3: CIRCUITOS COMBINACIONALES

### 3.1.- Compuertas Lógicas

En este capítulo estudiaremos algunos aspectos de análisis y diseño de circuitos combinacionales. En primer lugar, estudiaremos los dispositivos más comunes en el diseño lógico: las compuertas.

#### 3.1.1.- Inversor:

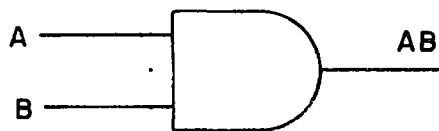
El inversor es un circuito de una entrada; a la salida produce el complemento del estado lógico de la entrada. Por ejemplo, si la entrada es 0, la salida es 1 y viceversa.



| A | $\bar{A}$ |
|---|-----------|
| 0 | 1         |
| 1 | 0         |

#### 3.1.2.- AND:

El circuito AND tiene dos o más entradas. en la figura siguiente se muestra su símbolo y su tabla de verdad.

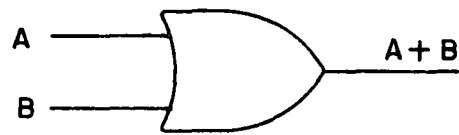


| A | B | A·B |
|---|---|-----|
| 0 | 0 | 0   |
| 0 | 1 | 0   |
| 1 | 0 | 0   |
| 1 | 1 | 1   |

De la tabla de verdad vemos que la salida del circuito será 1 sólo cuando ambas entradas, A y B, sean 1. Cualquier otra combinación a la entrada producirá un 0 a la salida.

#### 3.1.3.- OR

El circuito OR tiene dos o más entradas. En la figura se muestra su símbolo y su tabla de verdad.



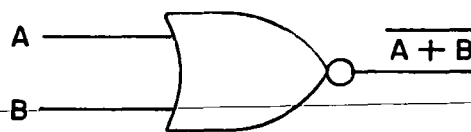
| A | B | A + B |
|---|---|-------|
| 0 | 0 | 0     |
| 0 | 1 | 1     |
| 1 | 0 | 1     |
| 1 | 1 | 1     |

De la tabla de verdad vemos que la salida del circuito será 1 cuando cualquiera (o todas) las entradas sean 1, y será 0 sólo cuando todas las entradas sean 0.

Los circuitos AND, OR e Inversor son los básicos de un sistema lógico y de hecho con estos tres es posible (aunque no es aconsejable) diseñar una computadora. Estos circuitos se combinan para producir funciones más elaboradas, algunas de las cuales veremos a continuación.

#### 3.1.4.- NOR:

El circuito NOR es una combinación del circuito OR y del Inversor. En la figura se muestra su símbolo y su tabla de verdad.

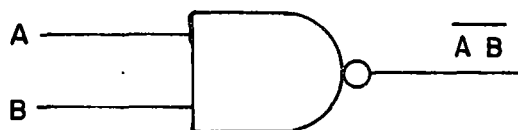


| A | B | A + B |
|---|---|-------|
| 0 | 0 | 1     |
| 0 | 1 | 0     |
| 1 | 0 | 0     |
| 1 | 1 | 0     |

De la tabla de verdad vemos que la salida será 1 sólo cuando todas las entradas sean 0 y será 0 cuando cualquiera de las entradas sea 1.

#### 3.1.5.- NAND:

El circuito NAND es una combinación del circuito AND y el Inversor. En la figura se muestra su símbolo y su tabla de verdad.

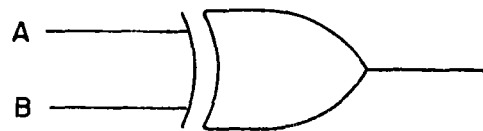


| A | B | A · B |
|---|---|-------|
| 0 | 0 | 1     |
| 0 | 1 | 1     |
| 1 | 0 | 1     |
| 1 | 1 | 0     |

La salida será 0 cuando todas las entradas sean 1 y será 1 cuando cualquiera de ellas sea 0.

### 3.1.6.- OR Exclusivo

El circuito OR Exclusivo realiza la función OR con la excepción de que cuando ambas entradas son 1, la salida es 0.



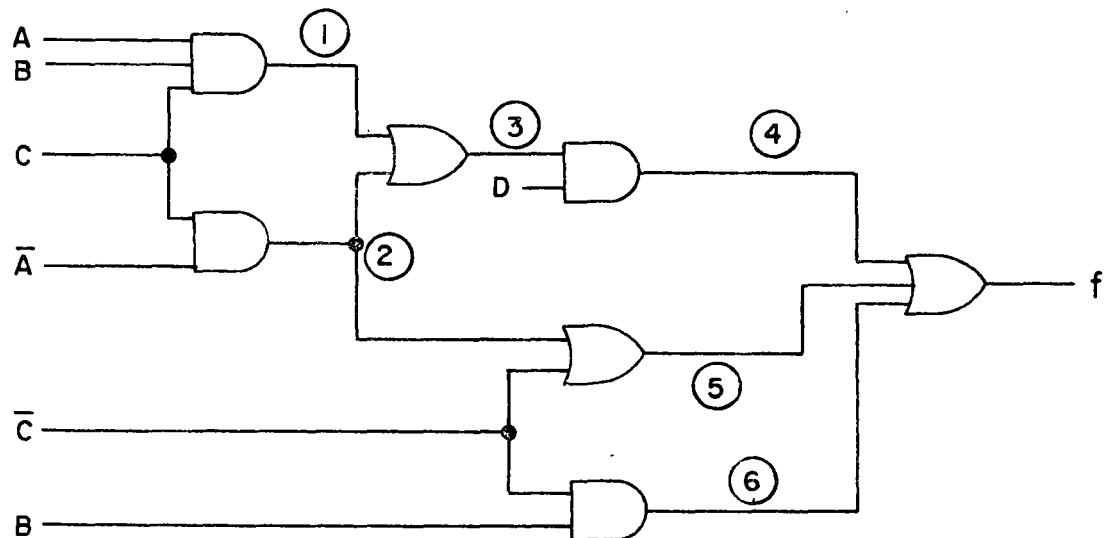
| A | B | $A \oplus B$ |
|---|---|--------------|
| 0 | 0 | 0            |
| 0 | 1 | 1            |
| 1 | 0 | 1            |
| 1 | 1 | 0            |

### 3.2.- Análisis de Circuitos Combinacionales:

El análisis de circuitos combinacionales se refiere a circuitos cuya salida depende exclusivamente de sus entradas y están formados por interconexiones de compuertas.

El análisis de tales circuitos requiere obtener la ecuación booleana que lo representa y luego la caracterización completa de la función resultante, para todas las combinaciones posibles de entrada:

Ejemplos:





$$\begin{aligned} f &= 4 + 5 + 6 \\ 4 &= D \cdot 3 \\ 5 &= \bar{C} + 2 \\ 6 &= B \cdot \bar{C} \end{aligned}$$

$$\begin{aligned} 3 &= 1 + 2 \\ 2 &= \bar{A} \cdot C \end{aligned}$$

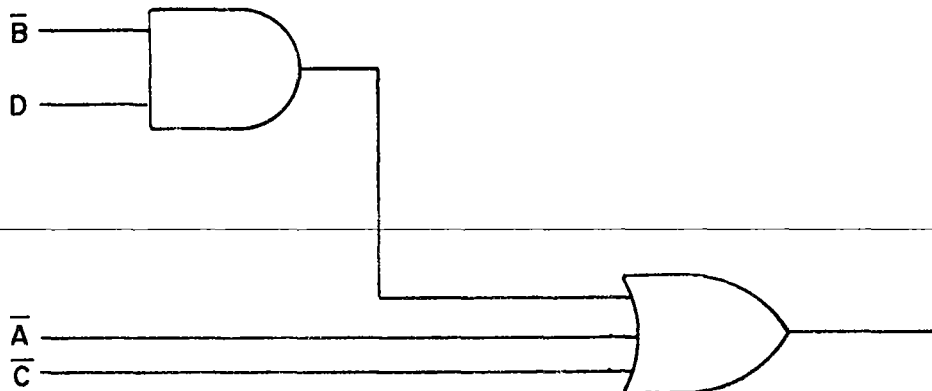
$$1 = A \cdot \bar{B} \cdot C$$

Luego:

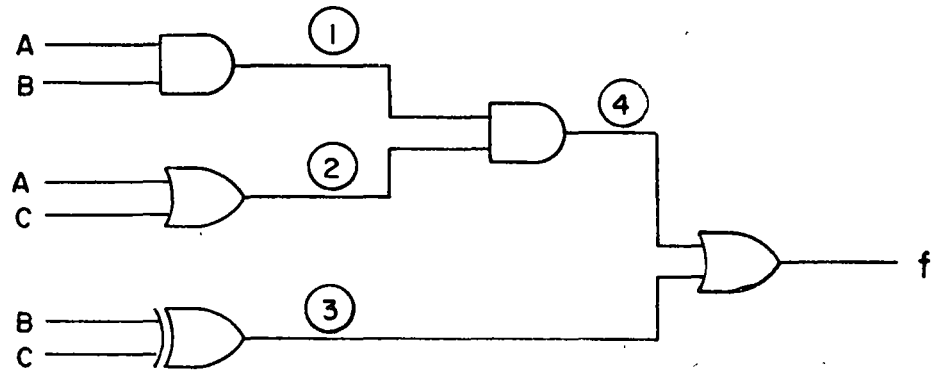
$$\begin{aligned} f &= D \cdot 3 + \bar{C} + 2 + B\bar{C} \\ f &= D(1 + 2) + \bar{C} + \bar{A}C + B\bar{C} \\ f &= D(\bar{A}BC + \bar{A}c) + \bar{C} + \bar{A}C + B\bar{C} \end{aligned}$$

Pero esta función podemos simplificarla como sigue:

$$\begin{aligned} f &= \bar{A}\bar{B}CD + \bar{A}\bar{C}D + \bar{A}C + B\bar{C} + \bar{C} \\ f &= CD(\bar{A} + \bar{A}\bar{B}) + \bar{A}C + \bar{C}(B + 1) \\ f &= CD(\bar{A} + \bar{B}) + \bar{A}C + \bar{C} \\ f &= \bar{A}CD + \bar{B}CD + \bar{A}C + \bar{C} \\ f &= \bar{A}C(D + 1) + \bar{B}DC + \bar{C} \\ f &= \bar{A}C + \bar{B}D + \bar{C} \\ f &= \bar{A}C + \bar{C} + \bar{B}D \\ f &= \bar{A} + \bar{C} + \bar{B}D \end{aligned}$$



2)



$$f = \overline{3 + 4}$$

$$4 = \overline{1 \cdot 2}$$

$$3 = B \oplus C$$

$$\textcircled{1} = \overline{AB}$$

$$\textcircled{2} = \overline{A + C}$$

$$f = \overline{(B \oplus C) + \textcircled{1} \cdot \textcircled{2}}$$

$$f = \overline{(B \oplus C + \overline{AB} (\overline{A + C}))}$$

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

Luego:

$$f = \overline{(\overline{B}C + BC) + \overline{AB} (\overline{A + C})}$$

$$\overline{f} = (\overline{B}C + BC) + \overline{AB} (\overline{A + C})$$

$$\overline{f} = \overline{B}C + BC + (\overline{A} + \overline{B}) A\overline{C}$$

$$\overline{f} = \overline{B}C + BC + A\overline{B}\overline{C}$$

$$\overline{f} = \overline{B}C(1 + A) + BC$$

$$\overline{f} = \overline{B}C + BC$$

$$\overline{f} = B \odot C \quad \text{COINCIDENCIA}$$

$$f = B \oplus C$$

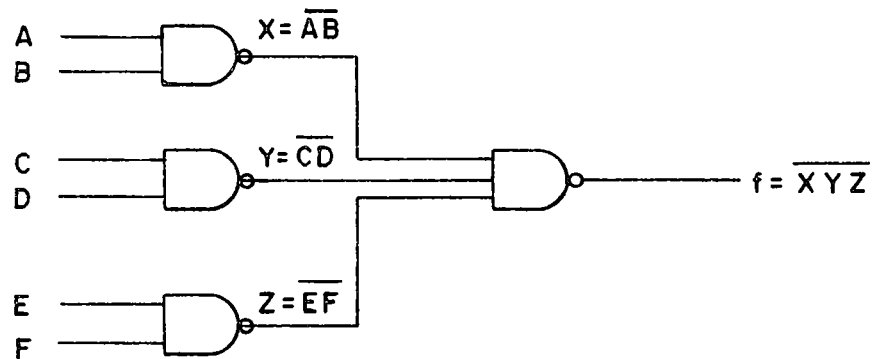
En los ejemplos anteriores vimos como se podía obtener un conjunto de ecuaciones para una estructura dada. En dichas ecuaciones se describían el tipo de compuerta y sus interconexiones. Por lo tanto, se pueden utilizar conjuntos de ecuaciones en vez de diagramas lógicos.

### 3.3.- Propiedades de las funciones NAND y NOR

Al revisar las dieciseis funciones definidas por las combinaciones de dos variables, observamos que todas ellas se pueden implementar mediante las funciones AND, OR y NOT. En general, cualquier función lógica puede ser implementada mediante las tres compuertas mencionadas.

Sin embargo, cabe preguntarse si no es posible implementar cualquier función con menos de tres tipos diferentes de compuertas.

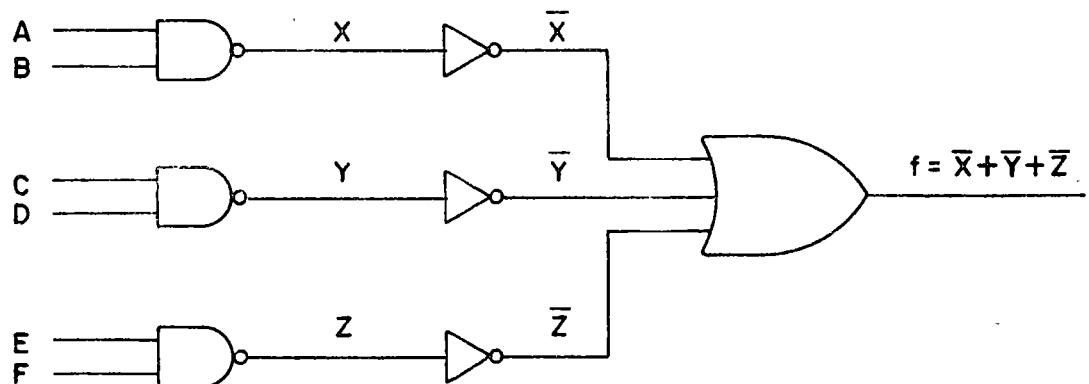
Consideremos el circuito de la figura, que consiste de tres compuertas NAND, cuyas salidas están conectadas a la entrada de otra compuerta NAND.



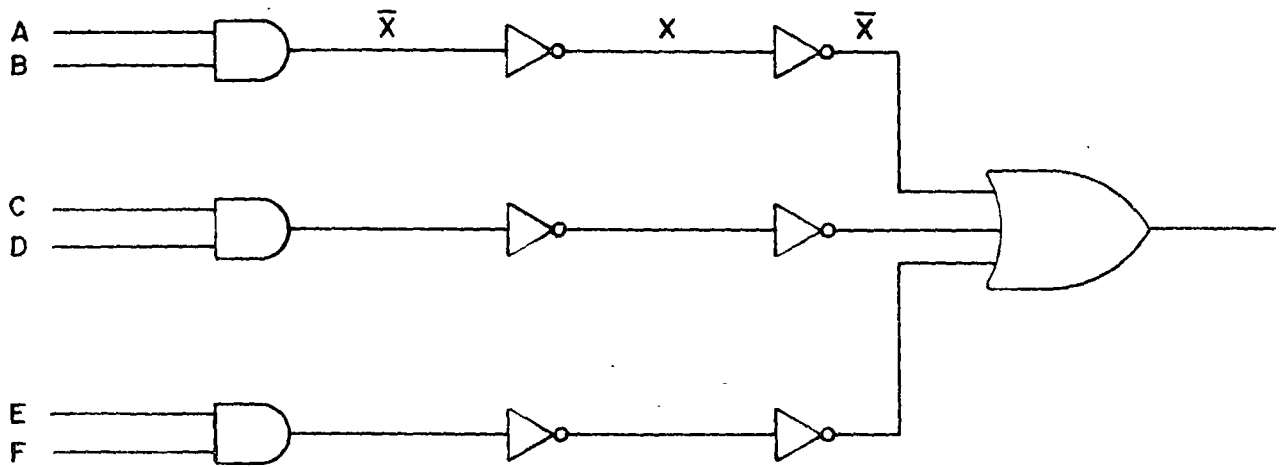
De las leyes de Morgan tenemos que:

$$f = \overline{XYZ} = \overline{X} + \overline{Y} + \overline{Z}$$

Dicha función también puede ser implementada mediante el siguiente circuito:



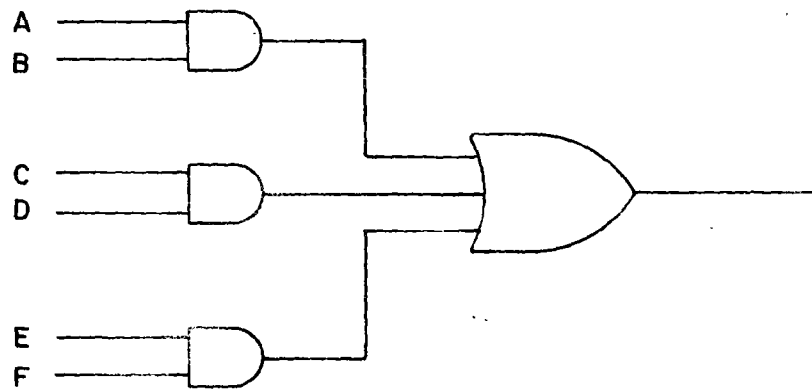
o bien, podemos además hacer lo siguiente



Pero tener dos inversores en cascada, equivale algebraicamente a:

$$\overline{\bar{X}} = X$$

Luego:



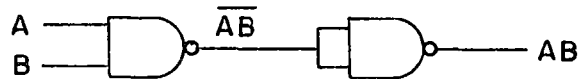
Es decir, utilizando sólo compuertas NAND, se implementó un circuito que utilizaba compuertas AND y OR. Esto sugiere que podemos implementar dichas

funciones (AND y OR) mediante compuertas NAND, y como además, ésta tiene implícita la función NOT, tendremos una compuerta mediante la cual se puede implementar cualquier función:

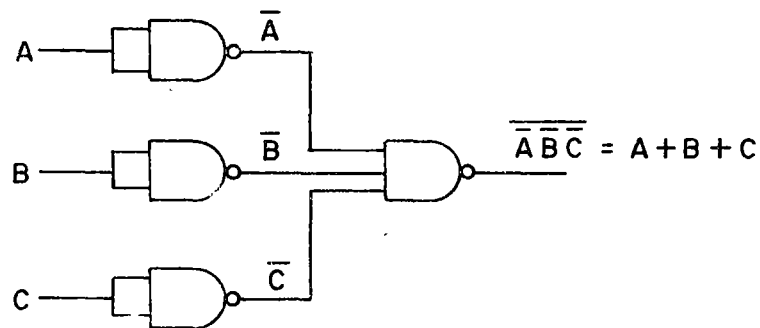
NOT



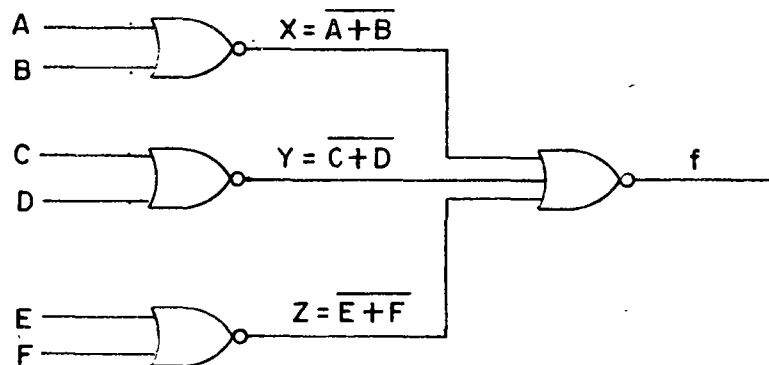
AND:



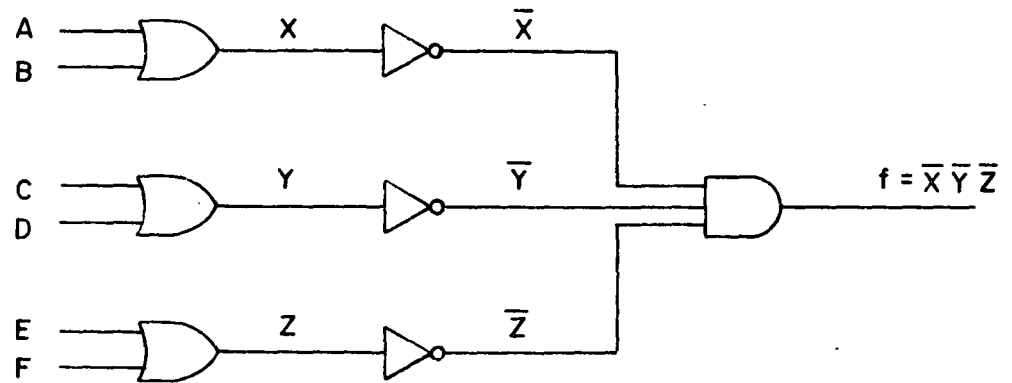
OR:



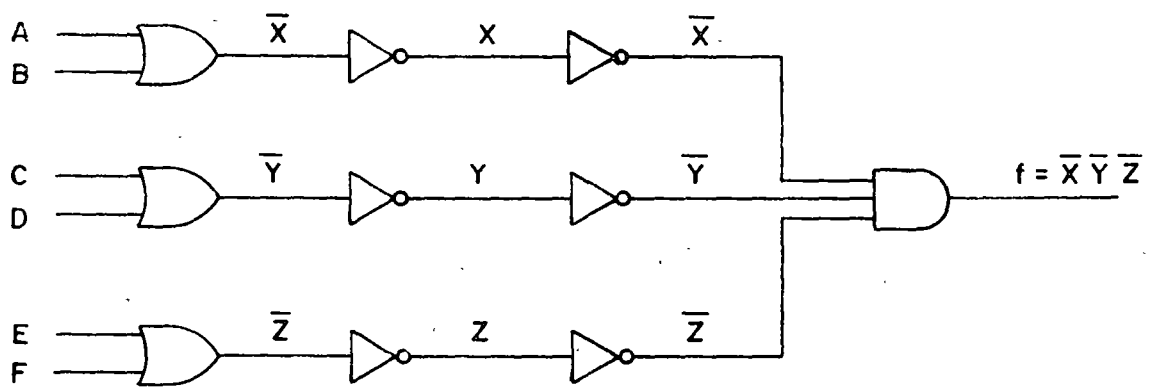
Lo anterior también se cumple para circuitos NOR como veremos a continuación: Consideremos nuevamente un circuito de segundo orden, pero a base de compuertas NOR:



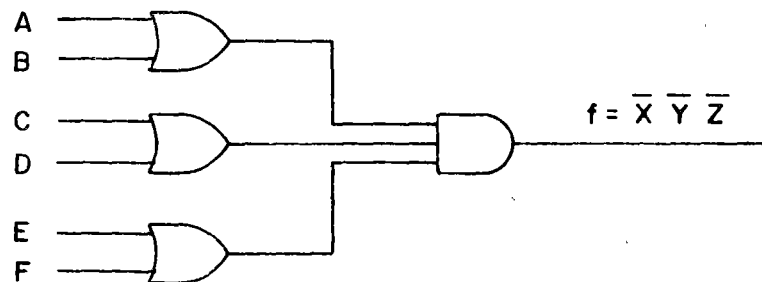
Dicha función se puede también implementar de la siguiente forma:



o bien



o bien

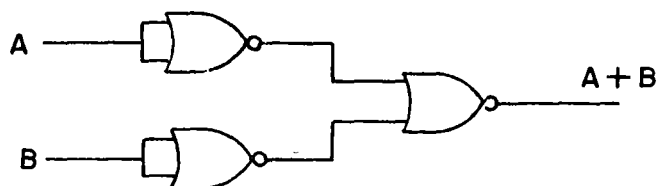


Las funciones NOT, AND y OR implementadas mediante compuertas NOR, toman las siguientes formas:

NOT:



AND:



OR:

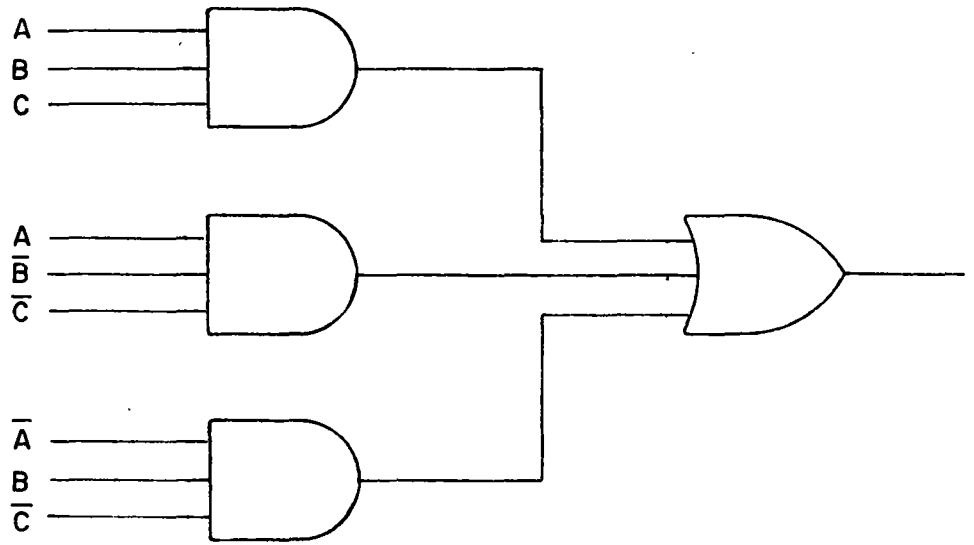


### 3.4.- Análisis de Circuitos NAND y NOR.

En la sección anterior vimos que es posible implementar cualquier función Booleana empleando solamente compuertas NAND o compuertas NOR. Vimos que cada una de las funciones básicas NOT, AND y OR tienen un circuito equivalente NAND y otro NOR. Una posibilidad sería entonces utilizar dichos circuitos equivalentes para convertir un circuito AND-OR a NAND o un circuito OR-AND a NOR o viceversa. Obviamente, aún cuando es posible hacerlo, la solución es impráctica debido a que aumenta considerablemente el número de compuertas usadas. A continuación veremos que el procedimiento es bastante más simple:

Consideremos la siguiente función:

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

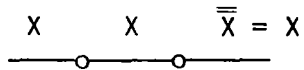


Dicha función está en forma de suma de productos y su implementación se realiza mediante compuertas AND/OR como se muestra en la figura.

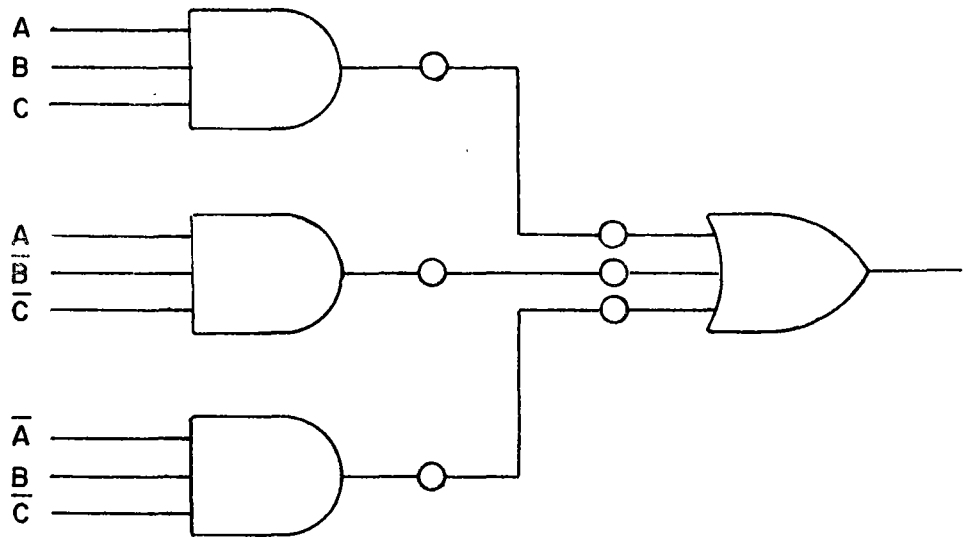
Del álgebra de boole sabemos que;

$$\overline{\overline{X}} = X$$

usaremos círculos pequeños para indicar la complementación, por ejemplo, en una línea tenemos:

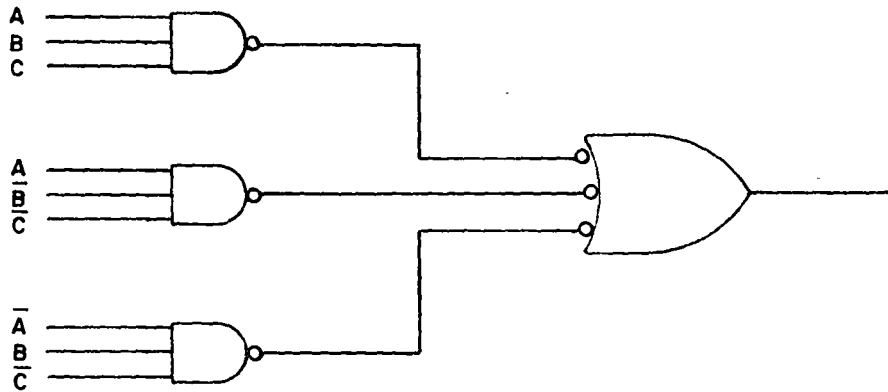


Con esta conversión, podemos modificar el circuito anterior sin alterarlo, de la siguiente forma:





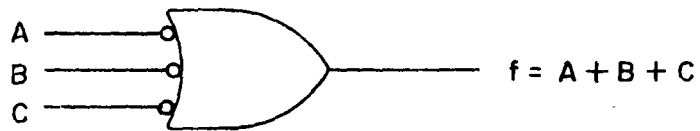
En el circuito anterior tenemos una compuerta AND seguida de un inversor, lo que equivale a tener una compuerta NAND, por lo tanto el circuito se modifica de la siguiente forma:



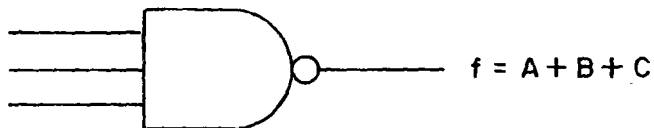
De esta forma hemos obtenido un circuito compuesto de tres compuertas NAND y una compuerta que invierte y ORea las señales a la entrada. Esta última compuerta no es familiar; sin embargo, como veremos es otra compuerta NAND, lo cual se deduce de las leyes de De Morgan:

Consideremos la compuerta NOT-OR Y LA NAND.

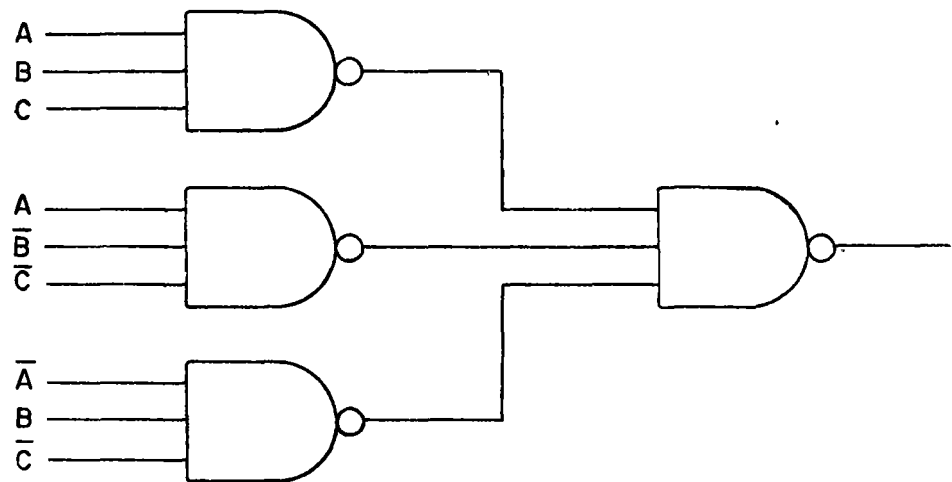
NOT-OR



NAND



es decir, ambas compuertas son idénticas, luego el circuito queda finalmente:



Es decir, para obtener un circuito con compuertas NAND a partir de AND-OR, simplemente reemplazamos las compuertas por NAND. Aunque lo anterior es cierto para la función anterior y en general para todas las funciones en forma de suma de productos, el procedimiento no es totalmente general y hay que observar ciertas reglas.

En general, podemos resumir lo siguiente:

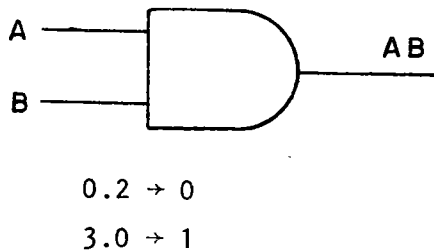
- 1.- UNA EXPRESION LOGICA EN FORMA DE SUMA DE PRODUCTOS SE PUEDE REALIZAR DE DOS FORMAS MEDIANTE DOS NIVELES DE COMPUERTAS.
  - i) UN NIVEL DE AND SEGUIDO POR UN NIVEL DE OR.
  - ii) DOS NIVELES DE NAND
- 2.- UNA EXPRESION LOGICA EN FORMA DE PRODUCTO DE SUMAS se puede realizar de dos formas mediante dos niveles de computar.
  - i) UN NIVEL DE OR SEGUIDO POR UN NIVEL DE AND.
  - ii) DOS NIVELES DE NOR.

En este punto conviene definir un nuevo concepto: el de lógica positiva y lógica negativa, que nos permitirá aclarar la dualidad de las compuertas del ejemplo anterior.

### 3.4.1.- Lógica Positiva y Negativa

Al usar una tabla de verdad para describir operaciones lógicas, no hacemos referencia a los niveles de voltaje con respecto a tierra. Sin embargo, al asignar el "1" o el "0" lógico a un nivel, cambiará la función lógica de la compuerta, según la forma en que se asignen los valores lógicos a los niveles de voltaje.

Consideremos el caso siguiente: típicamente los voltajes de trabajo de las compuertas TTL so de 0.2V y 3.0V. Supongamos que una compuerta AND la hemos definido asignando a 0.2V el "0" lógico y a 3.0V el "1" lógico. Tenemos entonces las siguientes tablas de verdad.



| A   | B   | A·B |
|-----|-----|-----|
| 0.2 | 0.2 | 0.2 |
| 0.2 | 3.0 | 0.2 |
| 3.0 | 0.2 | 0.2 |
| 3.0 | 3.0 | 3.0 |

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

Si ahora para la misma compuerta cambiamos nuestra conversión y hacemos que 0.2V le corresponda el "1" lógico y que a 3.0V le corresponda el "0" lógico, la tabla de verdad queda:

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

Veremos qué ahora obtenemos una función diferente referida al caso anterior.

En el primer caso, cuando  $0.2V \rightarrow "0"$  y  $3.0 \rightarrow "1"$  se tiene Lógica positiva y en el segundo caso cuando  $0.2V \rightarrow "1"$  y  $3.0 \rightarrow "0"$  se tiene Lógica negativa.

En los cuadros siguientes se muestran las tablas de verdad de las funciones más comunes, en lógica positiva y lógica negativa.

#### LOGICA POSITIVA

| Entradas |   | Salidas |    |      |     |
|----------|---|---------|----|------|-----|
| A        | B | AND     | OR | NAND | NOR |
| 0        | 0 | 0       | 0  | 1    | 1   |
| 0        | 1 | 0       | 1  | 1    | 0   |
| 1        | 0 | 0       | 1  | 1    | 0   |
| 1        | 1 | 1       | 1  | 0    | 0   |

## LOGICA NEGATIVA

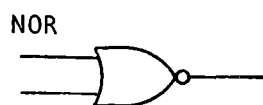
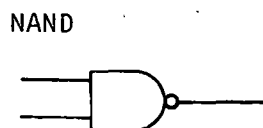
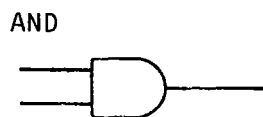
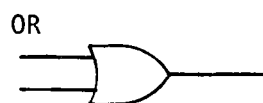
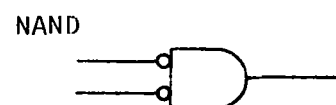
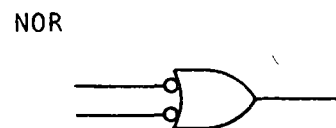
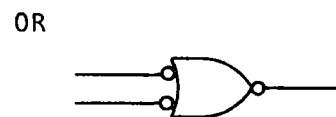
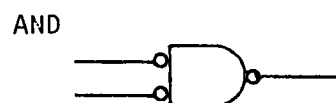
| Entradas |   | Salidas |    |      |     |
|----------|---|---------|----|------|-----|
| A        | B | AND     | OR | NAND | NOR |
| 0        | 0 | 0       | 0  | 1    | 1   |
| 0        | 1 | 1       | 0  | 0    | 1   |
| 1        | 0 | 1       | 0  | 0    | 1   |
| 1        | 1 | 1       | 1  | 0    | 0   |

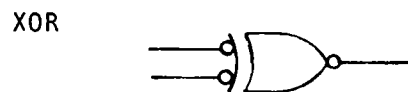
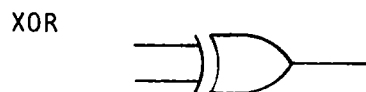
Comparando ambos cuadros vemos que existen las siguientes relaciones:

LOGICA POSITIVALOGICA NEGATIVA

AND  $\xrightarrow{\text{equivale}}$  OR  
 OR  $\xrightarrow{\text{equivale}}$  AND  
 NAND  $\xrightarrow{\text{equivale}}$  NOR  
 NOR  $\xrightarrow{\text{equivale}}$  NAND

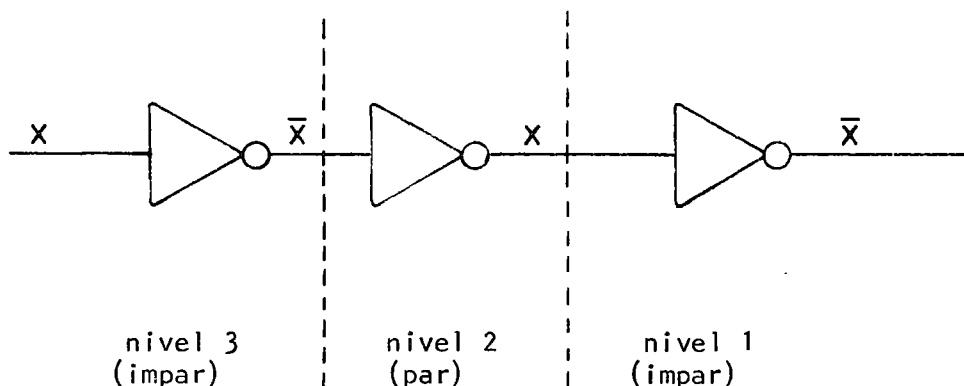
Estos resultados tienen implicaciones prácticas muy importantes, ya que mejoran substancialmente la flexibilidad de diseño. En lo que sigue veremos como utilizar estas propiedades.

SímbolosLOGICA POSITIVALOGICA NEGATIVA



### 3.4.2.- Consideraciones para el análisis de circuitos combinacionales con NAND/NOR.

Consideremos el circuito de la figura, que consiste de una cadena de inversores:



Observemos en dicho circuito que la señal de entrada  $X$  se obtiene en forma complementada en ciertos niveles y sin complementar en otros.

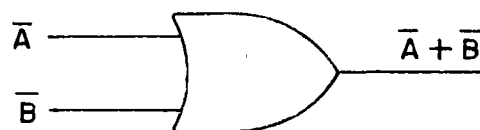
En general:

SI LA VARIABLE PASA A TRAVES DE LOS NUMEROS PAR DE NIVELES DE INVERSION, APARECERA A LA SALIDA EN LA MISMA FORMA QUE A LA ENTRADA, SI EN CAMBIO PASA POR UN NUMERO IMPAR DE NIVELES DE INVERSION APARECERA A LA SALIDA COMO EL COMPLEMENTO DE LA ENTRADA.

Para aplicar esta regla, comenzamos a numerar los niveles desde la salida sin complementar y las entradas en la forma correspondiente.

La regla anterior nos será de mucha utilidad para analizar y sintetizar circuitos con compuertas NAND o NOR.

Analicemos algunas características de la compuerta NAND. Anteriormente vimos que una compuerta NAND se comportaba como una compuerta OR con las entradas complementadas.



Esto lo podemos interpretar en términos de niveles de inversión y de lógica positiva y negativa, de la siguiente forma:

Para una compuerta en un nivel impar las variables de entrada a la compuerta están complementadas. En términos de lógica positiva, están en su nivel bajo. Si usamos lógica mezclada (positiva y negativa) podemos usar lógica negativa en los niveles impares de inversión. Como la compuerta es de lógica positiva, tendremos lo siguiente:

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

| A | B | "OR" |
|---|---|------|
| 1 | 1 | 1    |
| 1 | 0 | 1    |
| 0 | 1 | 1    |
| 0 | 0 | 0    |

} Lógica negativa a la entrada.

y vemos que la compuerta NAND se comporta como una compuerta OR cuando asignamos lógica negativa a las entradas, lo que hacemos en los niveles impares de inversión.

Un argumento similar se puede aplicar al caso de compuertas NOR y veremos que ésta se comporta como una compuerta AND en niveles impares de inversión.

Similarmente, para una compuerta NAND en un nivel par, sus entradas estarán sin complementar y su salida estará complementada; si consideramos la salida como lógica negativa y las entradas como lógica positiva tendremos:

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

| A | B | "AND" |
|---|---|-------|
| 0 | 0 | 0     |
| 0 | 1 | 0     |
| 1 | 0 | 0     |
| 1 | 1 | 1     |

} Lógica negativa a la salida.

De donde vemos que bajo dichas condiciones la compuerta NAND se comporta como una AND en niveles pares de inversión.

Resumiendo:

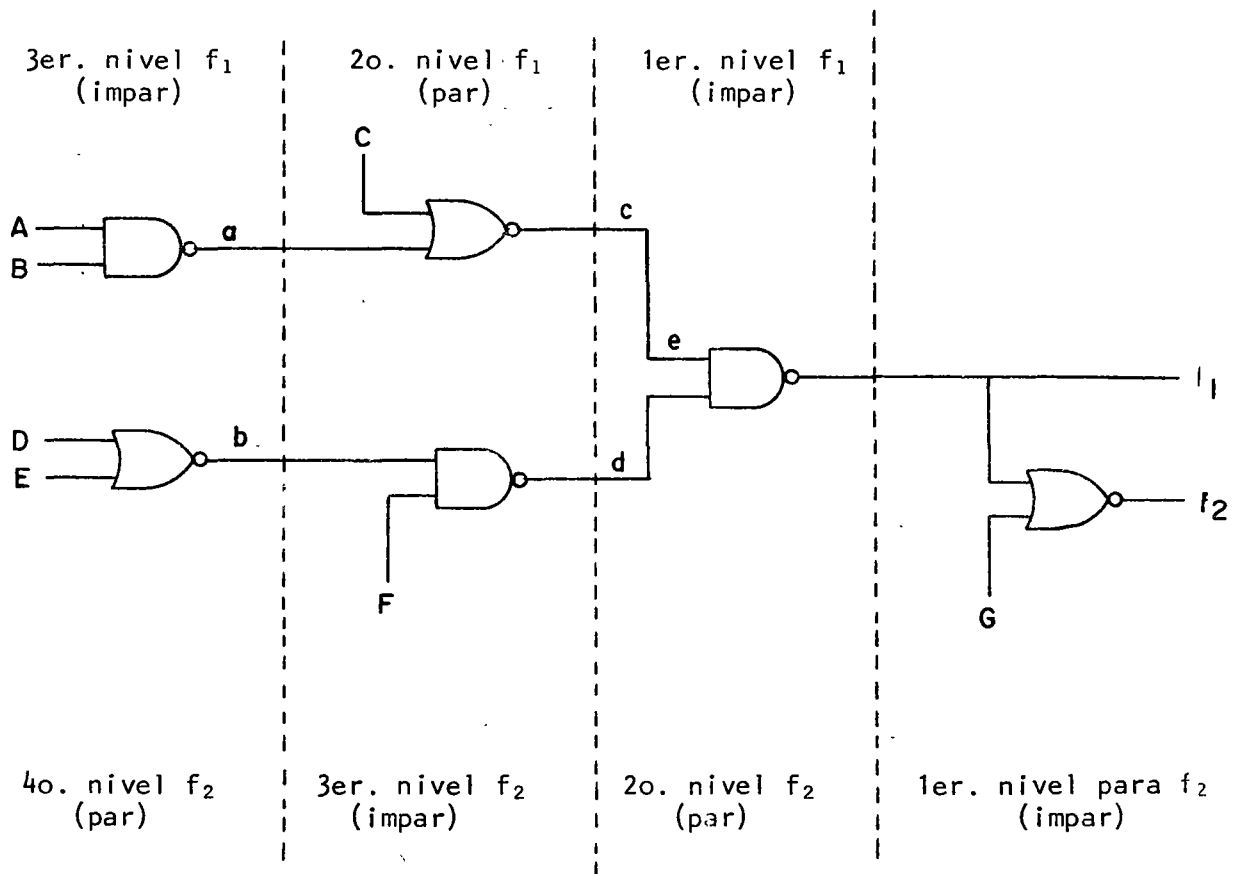
|      | Comportamiento |             |
|------|----------------|-------------|
|      | Nivel par      | Nivel impar |
| NAND | AND            | OR          |
| NOR  | OR             | AND         |

Utilizando estos resultados podemos elaborar el siguiente conjunto de reglas para obtener la expresión de salida de un circuito que utilice compuertas inversoras:

- 1.- CONSIDERAR LA COMPUERTA DESDE LA CUAL SE OBTENDRA LA SEÑAL DE SALIDA COMO EL PRIMER NIVEL DE INVERSION (IMPARE); LA COMPUERTA PRECEDENTE COMO SEGUNDO NIVEL, ETC.
- 2.- CONSIDERAR TODAS LAS COMPUERTAS NAND EN NIVELES IMPARES REALIZANDO LA FUNCION OR.
- 3.- CONSIDERAR TODAS LAS COMPUERTAS NAND EN NIVELES PARES, REALIZANDO LA FUNCION AND.
- 4.- CONSIDERAR TODAS LAS COMPUERTAS NOR EN NIVELES IMPARES REALIZANDO LA FUNCION AND.
- 5.- CONSIDERAR TODAS LAS COMPUERTAS NOR EN NIVELES PARES REALIZANDO LA FUNCION OR.
- 6.- TODAS LAS VARIABLES DE ENTRADA A COMPUERTAS EN NIVELES IMPARES DEBEN APARECER EN FORMA COMPLEMENTADA EN LA EXPRESION DE LA SEÑAL DE SALIDA.
- 7.- TODAS LAS VARIABLES DE ENTRADA A COMPUERTAS EN NIVEL PAR DEBEN APARECER SIN COMPLEMENTAR EN LA EXPRESION DE LA SEÑAL DE SALIDA.

Observación: Estas reglas se utilizan para pasar de una configuración AND-OR (OR-AND) a NAND (NOR) o para derivar la expresión de la señal de salida de un circuito de compuertas NAND-NOR en términos de compuertas AND-OR.

Ejemplo: Derivar las expresiones para  $f_1$  y  $f_2$



Para  $f_1$ :

- La compuerta NAND # 5 está en el primer nivel (impar) y se debe considerar como realizando la función OR.
- La compuerta NAND # 4 está en el segundo nivel (par) y se debe considerar como realizando la función AND; además la variable F debe aparecer sin complementar en la expresión para  $f_1$ .
- La compuerta NOR # 3 está en el segundo nivel (par) y se debe considerar como realizando la función OR; la variable C debe aparecer sin complementar en la expresión para  $f_1$ .
- NOR # 2 está en el tercer nivel para  $f_1$  (impar), se debe considerar como realizando la función AND. Las variables D y E deben aparecer complementadas en la expresión para  $f_1$ .
- NAND # 1 está en el tercer nivel para  $f_1$  (impar), se debe considerar como realizando la función OR. Las variables A y B deben aparecer complementadas en la expresión para  $f_1$ .

Luego:

1er. nivel NAND # 5

$$f_1 = \underbrace{\overline{\overline{A + B + C}}}_{\substack{\text{3er.n.} \\ \text{NAND1}}} + \underbrace{\overline{\overline{D E F}}}_{\substack{\text{2o.n.} \\ \text{NAND4}}}$$

De otra manera habríamos tenido

$$\begin{aligned} f_1 &= \overline{c d} & c &= \overline{c + a} & a &= \overline{AB} \\ & & d &= \overline{Fb} & b &= \overline{D + E} \\ f_2 &= \overline{\overline{(c + a)} \overline{(Fb)}} = \overline{\overline{(c + \overline{AB})} \overline{(F(\overline{D + E}))}} \\ f_2 &= \overline{\overline{(c + \overline{AB})} + \overline{(F(\overline{D + E}))}} \\ f_2 &= (c + \overline{A} + \overline{B}) + F\overline{D}\overline{E} \end{aligned}$$

Para  $f_2$ :

- NOR # 6: nivel impar; considerar como AND; la variable B debe aparecer complementada en expresión para  $f_2$ .
- NAND # 5: nivel par; considerar como AND
- NAND # 4: nivel impar; considerar como OR; la variable F debe aparecer complementada en la expresión para  $f_2$ .



- d) NOR # 3: nivel impar; considerar como AND; la variable C debe aparecer complementada en  $f_2$ .
- e) NOR # 2: nivel par; considerar como OR; las variables D y E deben aparecer sin complementar en  $f_2$ .
- f) NAND # 1: nivel par; considerar como AND, las variables A y B deben aparecer sin complementar en  $f_2$ .

Luego:

$$f_2 = \bar{G} \left[ (\bar{F} + (D + E)) (\bar{C}(A \cdot B)) \right] = ABC\bar{G} (D + E + F)$$

De otra manera habríamos tenido que proceder como sigue:

$$f_2 = \overline{e + G} \quad e = \overline{cd} \quad c = \overline{C + a} \quad a = \overline{AB}$$

$$d = \overline{Fb} \quad b = \overline{D + E}$$

$$f_2 = \overline{\overline{cd} + G}$$

$$= \overline{(\overline{C + a}) \overline{FB} + G}$$

$$= \overline{(C + \overline{AB}) F (\overline{D + E}) + G}$$

$$= \bar{G} \left[ (C + \overline{AB}) (F(\overline{D + E})) \right]$$

$$= \bar{G} (\bar{C}AB) (\bar{F} + D + E)$$

$$= ABC\bar{G} (D + E + \bar{F})$$

o bien:

$$f_2 = \overline{\bar{f}_1 + G} = \bar{f}_1 \bar{G}$$

$$\text{pero } f_1 = (\bar{A} + \bar{B} + C) + \bar{D}\bar{E}F$$

$$\text{luego } \bar{f}_1 = ABC (D + E + \bar{F})$$

$$\text{y } f_2 = ABC\bar{G} (D + E + \bar{F})$$

En el ejemplo anterior vimos una aplicación de las reglas establecidas anteriormente. En este caso se trató de derivar las expresiones de las funciones de salida de un circuito NAND-NOR, refiriendo las compuertas a NAND y OR según el caso.

A continuación veremos síntesis de circuitos NAND y NOR y extenderemos la aplicación de las reglas para emplearlas en la síntesis de circuitos NAND y NOR.

### 3.5.- Diseño con compuertas NAND y NOR.

De lo que se ha visto hasta este punto, se puede intuir un proceso de síntesis de circuitos implementados con NAND y NOR. A continuación veremos dos métodos de síntesis:

3.5.1.- Método de Doble complemento: El método más simple para realizar una función booleana con compuertas NAND es mediante la aplicación del teorema de DeMorgan a una función en forma de suma de productos. La expresión resultante se puede realizar directamente mediante compuertas NAND.

#### Ejemplo:

$$f = A\bar{C} + BCD + \bar{A}D$$

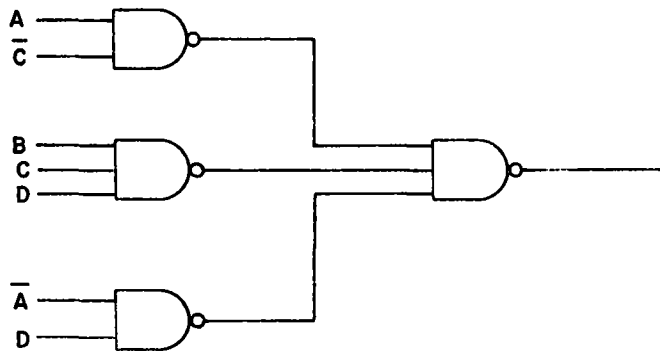
Si complementamos dos veces la función, obtenemos nuevamente la función:

$$f = \overline{\overline{A\bar{C} + BCD + \bar{A}D}}$$

Usando una de las leyes de DeMorgan, la función queda:

$$f = \overline{\overline{A\bar{C}} \overline{BCD} \overline{\bar{A}D}} = \overline{X_1 X_2 X_3}$$

En esta expresión reconocemos inmediatamente que puede ser realizada mediante compuertas NAND.



### 3.5.2.- Método de Transformación:

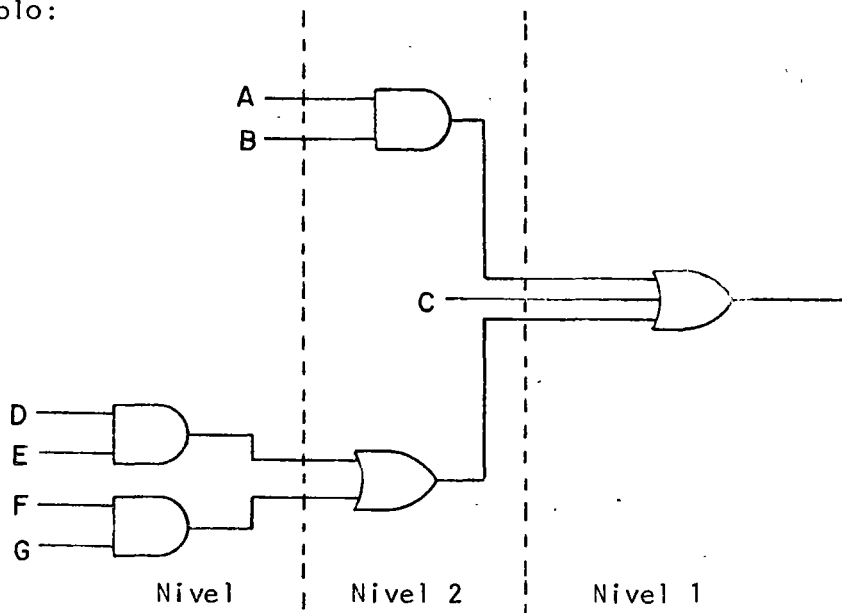
Este método consiste en la aplicación de las reglas desarrolladas en la sección anterior, con el objeto de transformar un circuito AND-OR en NAND. El procedimiento se resume en:

#### Regla:

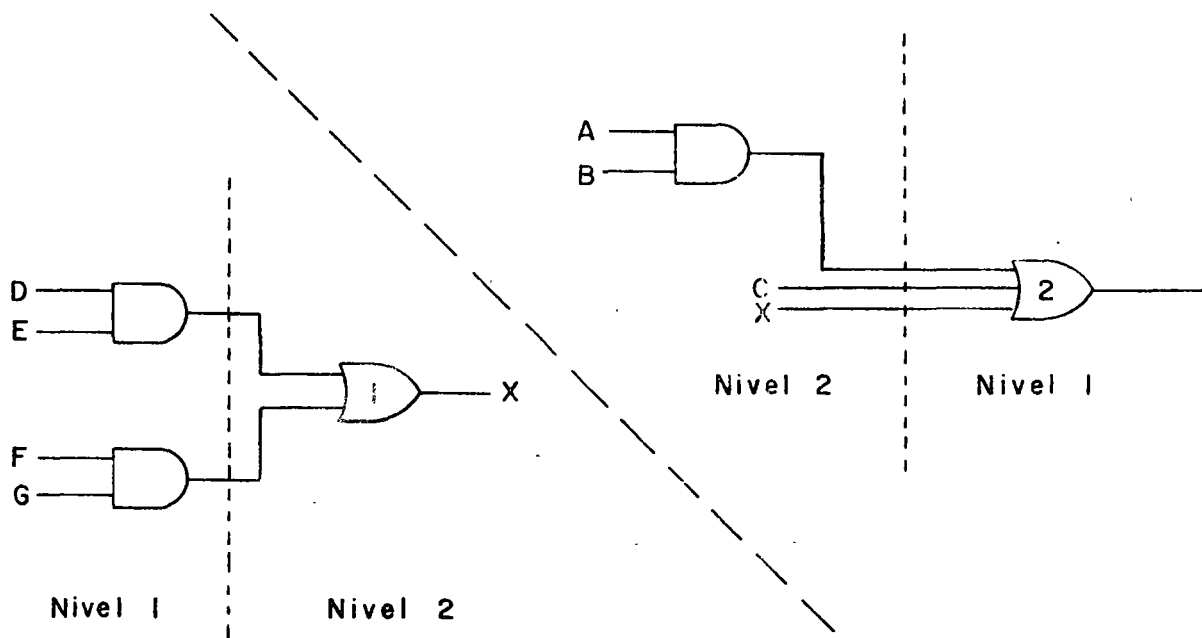
Arreglar el circuito en forma de niveles alternados de compuertas AND y OR,

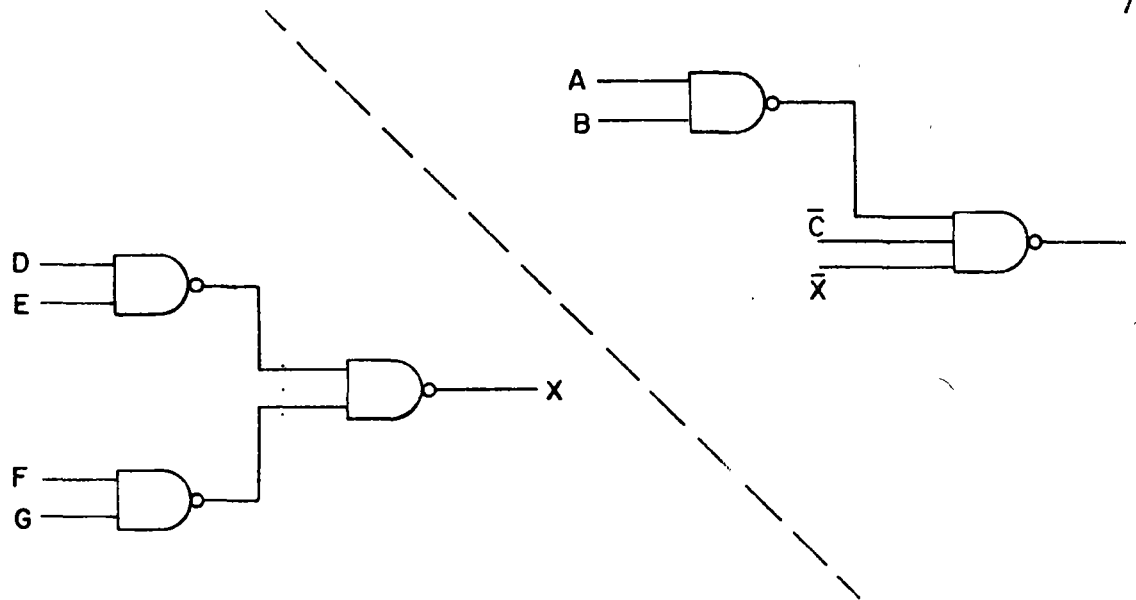
con una salida OR (AND). Reemplazar todas las compuertas AND y OR por compuertas NAND (NOR): Complementar todas las entradas a compuertas en niveles impares.

Ejemplo:

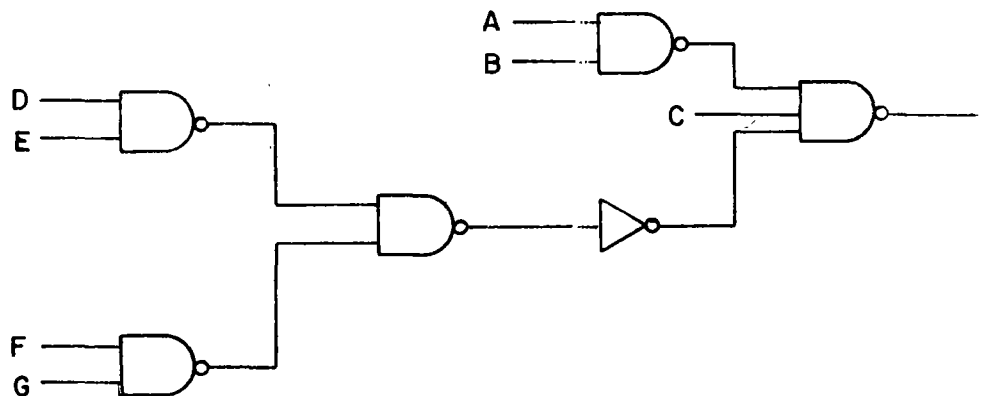


Este circuito se puede arreglar en la forma contemplada por la Regla anterior debido a los dos niveles sucesivos de compuertas OR, por lo tanto, dividimos el circuito en dos subcircuitos de dos niveles AND-OR.





Obsérvese que la salida del OR-1 debe entrar complementada al OR 2, luego necesitamos poner un inversor.



Como hemos visto en los dos métodos anteriores, ambos implican la minimización utilizando mapas de Karnaugh para llegar a una forma de Suma de Productos o de Productos de Suma. Posteriormente se debe analizar si es posible obtener una forma más simplificada aún al factorizar la función.

### 3.5.- Circuitos NAND de tres niveles:

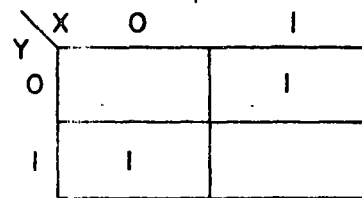
Hasta el momento, en nuestros métodos de minimización se han orientado hacia la obtención de circuitos con el mínimo número de compuertas y con el mínimo número total de entradas a compuertas, suponiéndose que se cuenta

con las variables de entrada en sus formas complementada y sin complementar. Hemos visto que se pueden obtener circuitos NAND en forma de suma de productos directamente del mapa de Karnaugh. Aún cuando estos circuitos son interesantes, su utilidad está basada en una suposición falsa: que se dispone de todas las entradas en su forma complementada y sin complementar, lo cual es un caso muy raro en la práctica. Si se usan soluciones en forma de suma de productos, nos encontraremos con que se requerirán varios inversores y nuestra solución dejará de ser mínima.

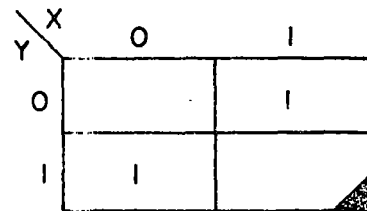
El problema de inversores se puede eliminar en gran parte usando circuitos de tres niveles. El procedimiento para obtener circuitos de tres niveles emplea el mapa de Karnaugh como herramienta básica. El procedimiento es similar al requerido para una solución en forma de suma de productos, con algunas modificaciones que se ilustran en el ejemplo desarrollado a continuación.

Ejemplo: Implementar la función  $f = X\bar{Y} + \bar{X}Y$ . No se dispone de entradas complementadas.

1.- Dibujar en mapa-K de la función deseada.

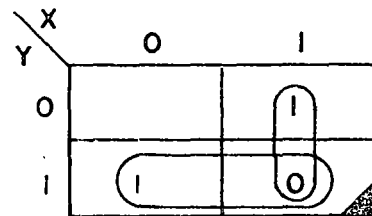


2.- Indicar en el mapa el o los cuadrados correspondientes a las entradas disponibles.



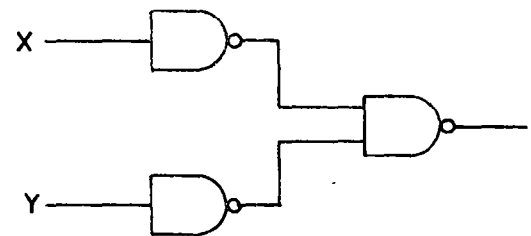
Sólo  
y Y e  
tán d s  
ponib s  
es.

3.- Todos los loops dibujados en el mapa deben incluir al cuadrado marcado. Dibujar loops implicante en el mapa, para cubrir todos los 1. Asegurarse que cada loop cubra un cuadrado marcado aún cuando encierre algunos 0s.

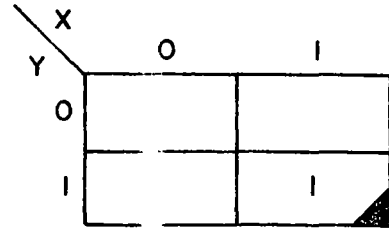


$$f = X + y$$

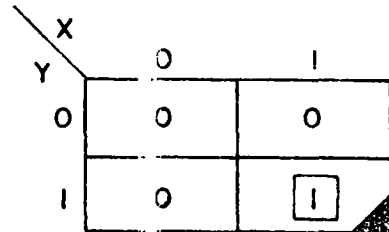
4.- Los loops se tratan temporalmente como un conjunto normal de implicantes y se construye un circuito. Se requiere un NAND por cada lazo más un NAND adicional para la salida. Las líneas de entrada a las NANDS se obtienen refiriéndose a lazos en el mapa.



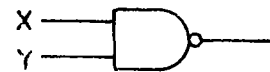
- 5.- En este punto, el circuito es correcto excepto por los ceros que se han encerrado en los lazos. Dichos ceros deben ser inhibidos. Esto se logra considerando dichos ceros como un nuevo problema y mapeándolos como 1 en un nuevo mapa.



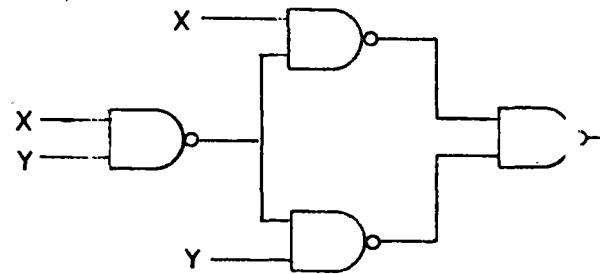
- 6.- Nuevamente, dibujar loops implicantes que cubran todos los 1 del mapa. Asegurarse que cada loop cubra un cuadrado marcado. En este punto ya no es necesario encerrar ceros.



- 7.- Se asigna una compuerta NAND a cada uno de los lazos. Las entradas a dichas compuertas se toman de los lazos en el mapa.



- 8.- Las salidas de estas compuertas se pueden considerar como términos inhibidores. Cada NAND se relaciona con un 0 específico o con un conjunto de ceros que se agruparon previamente en un lazo. Las salidas de estos bloques se conectan selectivamente a las NAND's previamente conectadas. Asegurarse que cada cero en cada lazo es inhibido.

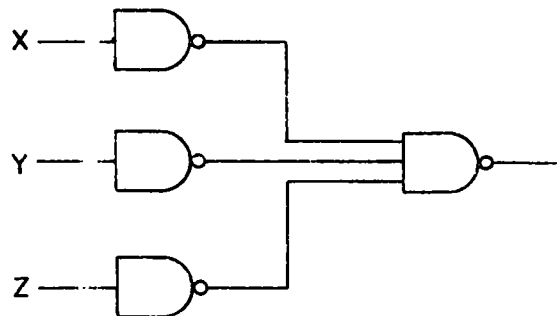


Si un cero es encerrado por más de un loop se le debe inhibir en cada uno de los loops que lo encerraba.

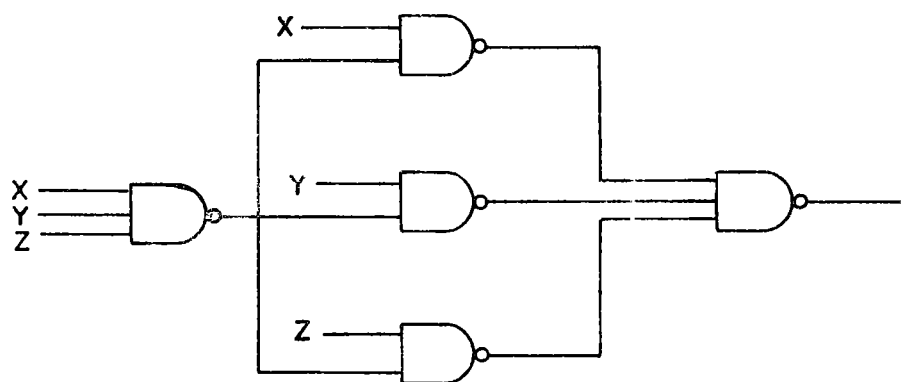
Ejemplo:

$$f = \bar{X}\bar{Y}Z + \bar{X}YZ + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + X\bar{Y}Z + XY\bar{Z}$$

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

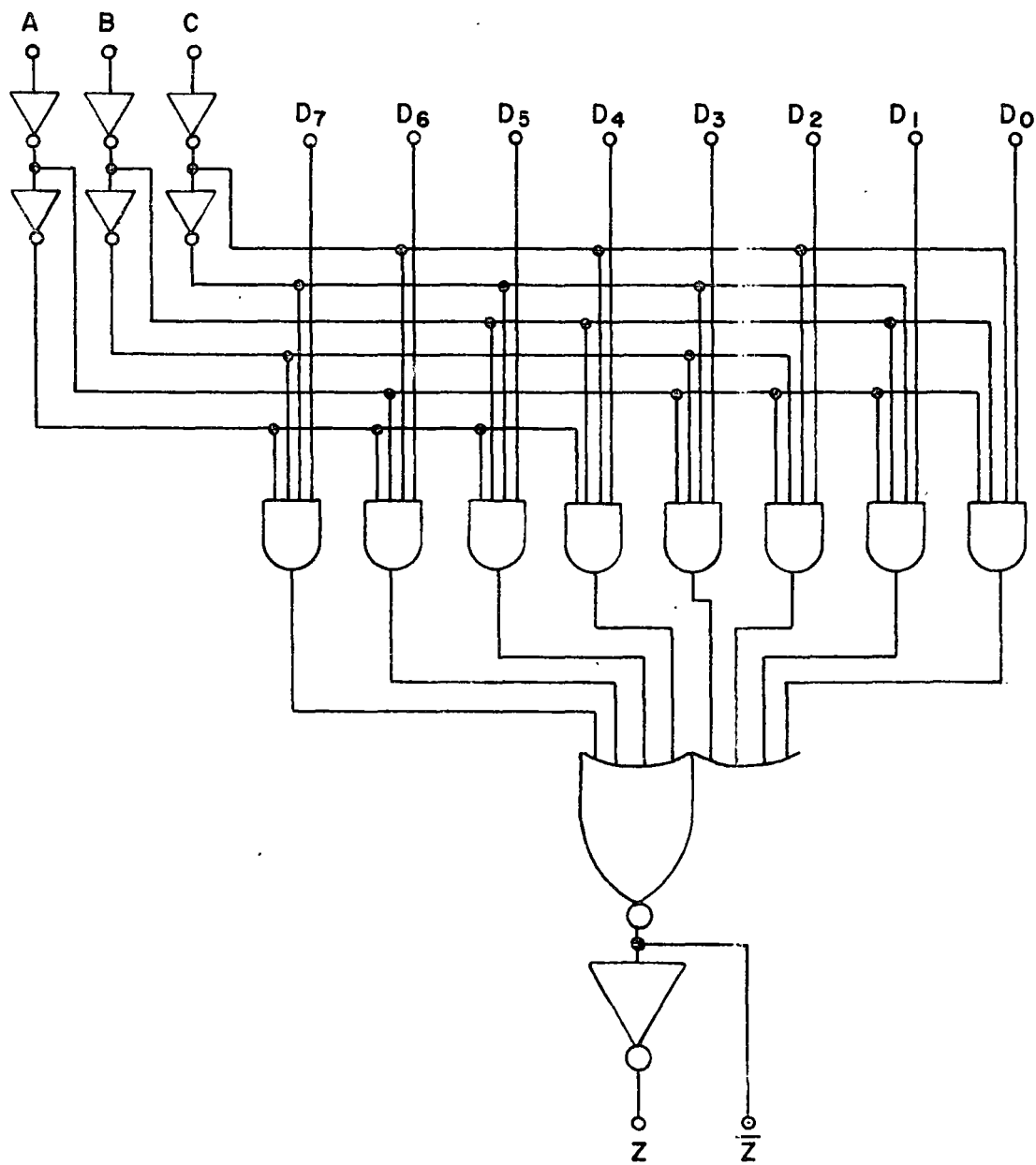
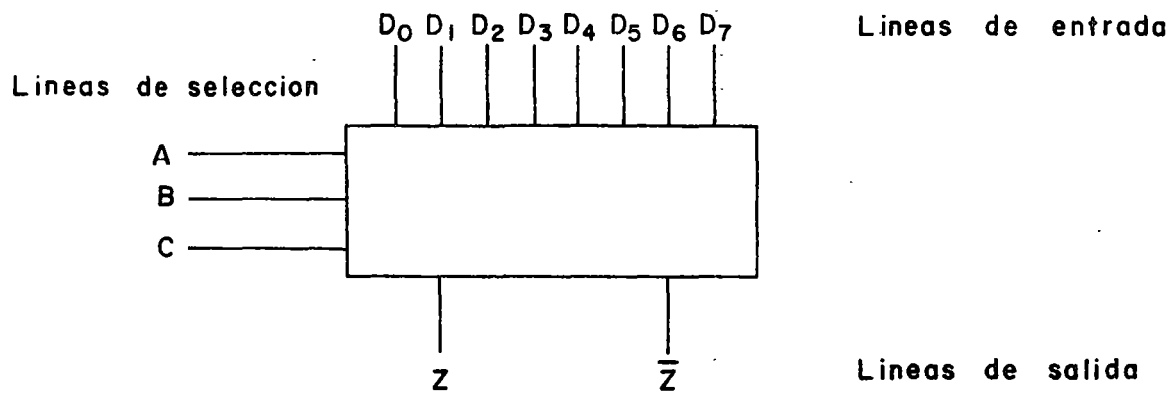


| XY |   | Z  |    |    |    |
|----|---|----|----|----|----|
|    |   | 00 | 01 | 11 | 10 |
| Z  | 0 | 0  | 0  | 0  | 0  |
|    | 1 | 0  | 0  | 1  | 0  |









La ecuación de salida del multiplexor de 8 entradas es la siguiente:

$$Z = D_0 \bar{A} \bar{B} \bar{C} + D_1 \bar{A} \bar{B} C + D_2 \bar{A} B \bar{C} + D_3 \bar{A} B C + D_4 A \bar{B} \bar{C} + D_5 A \bar{B} C + D_6 A B \bar{C} + D_7 A B C$$

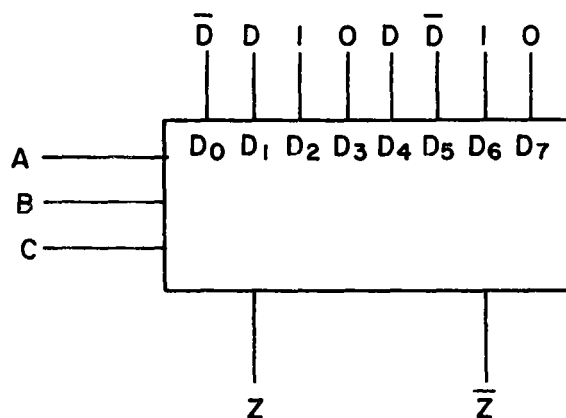
Analizando esta ecuación y observando al dispositivo, vemos que contamos con una entrada para cada una de las 8 combinaciones de las líneas de selección A, B y C. Por lo tanto, para una función de n variables, podemos "factorizar" A, B y C de la función y obtener ocho funciones separadas, cada una de n-3 variables, que deberán conectarse a la entrada correspondiente del multiplexor.

Por ejemplo consideremos la función:

$$f(A,B,C,D) = \bar{A} \bar{B} \bar{C} \bar{D} + \bar{A} \bar{B} C \bar{D} + \bar{A} B \bar{C} \bar{D} + \bar{A} B C \bar{D} + A \bar{B} \bar{C} D + A \bar{B} C D + A B \bar{C} D + A B C D$$

Hagamos la siguiente tabla:

| Entrada        | Dirección                 | Otras Variables |           |
|----------------|---------------------------|-----------------|-----------|
| D <sub>0</sub> | $\bar{A} \bar{B} \bar{C}$ | $\bar{D}$       | $\bar{D}$ |
| D <sub>1</sub> | $\bar{A} \bar{B} C$       | D               | D         |
| D <sub>2</sub> | $\bar{A} B \bar{C}$       | D + $\bar{D}$   | 1         |
| D <sub>3</sub> | $\bar{A} B C$             |                 | 0         |
| D <sub>4</sub> | A $\bar{B} \bar{C}$       | D               | D         |
| D <sub>5</sub> | A $\bar{B} C$             | $\bar{D}$       | D         |
| D <sub>6</sub> | A B $\bar{C}$             | $\bar{D} + D$   | 1         |
| D <sub>7</sub> | A B C                     |                 | 0         |



$$Z = \bar{A} \bar{B} \bar{C} \bar{D} + \bar{A} \bar{B} C \bar{D} + \bar{A} B \bar{C} \bar{D} + \bar{A} B C \bar{D} + A \bar{B} \bar{C} D + A \bar{B} C D + A B \bar{C} D + A B C D$$

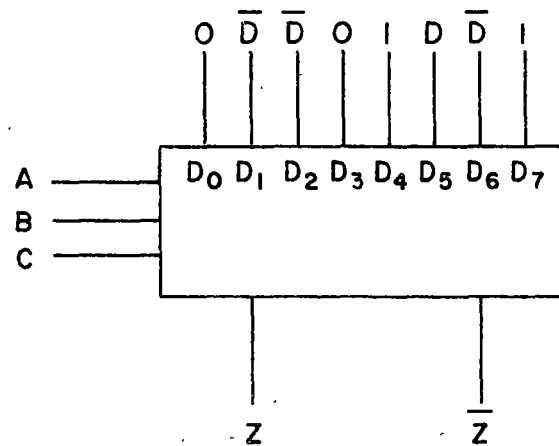
Consideremos otro ejemplo y veamos las distintas alternativas disponibles para su implementación.

$$f(A,B,C,D) = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD$$

1a. Alternativa: Emplear un multiplexor de 8 entradas

$$f = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD$$

| Entrada        | Dirección                 | Otras Variables |           |
|----------------|---------------------------|-----------------|-----------|
| D <sub>0</sub> | $\bar{A} \bar{B} \bar{C}$ | 0               | 0         |
| D <sub>1</sub> | $\bar{A} \bar{B} C$       | $\bar{D}$       | $\bar{D}$ |
| D <sub>2</sub> | $\bar{A} B \bar{C}$       | D               | D         |
| D <sub>3</sub> | $\bar{A} B C$             | 0               | 0         |
| D <sub>4</sub> | $A \bar{B} \bar{C}$       | $\bar{D} + D$   | 1         |
| D <sub>5</sub> | $A \bar{B} C$             | D               | D         |
| D <sub>6</sub> | $A B \bar{C}$             | $\bar{D}$       | $\bar{D}$ |
| D <sub>7</sub> | $A B C$                   | $\bar{D} + D$   | 1         |

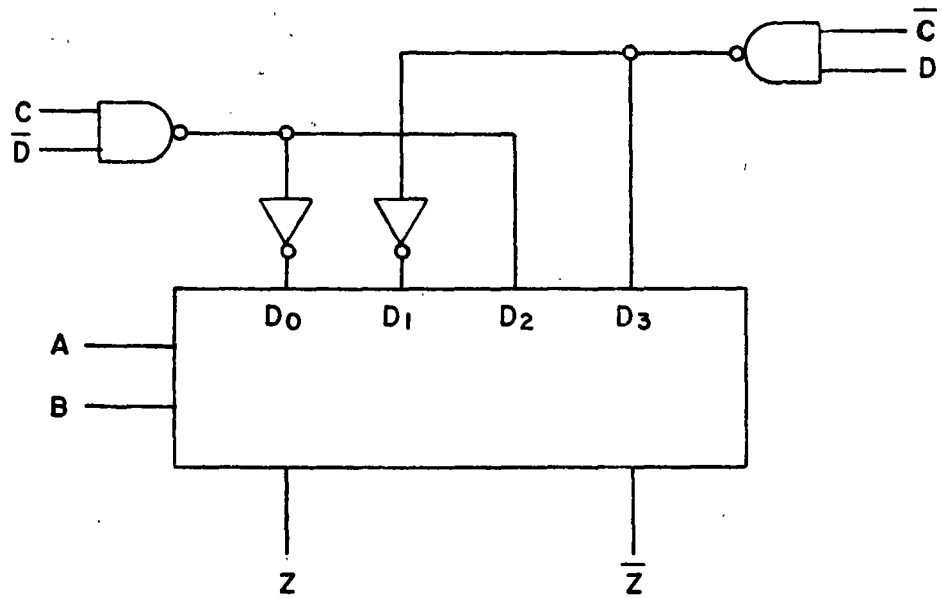


$$Z = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD$$

2a. Alternativa: Un multiplexor de 4 entradas y lógica discreta

| Entrada        | Dirección | Otras Variables |                 |
|----------------|-----------|-----------------|-----------------|
| D <sub>0</sub> | A B       | C D             | C D             |
| D <sub>1</sub> | A B       | C D             | C D             |
| D <sub>2</sub> | A B       | C D + C D + C D | C + C D = C + D |
| D <sub>3</sub> | A B       | C D + C D + C D | C + C D = C + D |

minimizar



3a. Alternativa: Emplear 4 mux de 2 entradas y 1 de 4 entradas

Para esto, generaremos cuatro tablas de verdad, cada una de las cuales servirá para definir las entradas del primer nivel de multiplexores.

| A = D; B = 0 |   |                | A = 0; B = 1 |   |                | A = 1; B = 0 |   |                | A = 1; B = 1 |   |                |
|--------------|---|----------------|--------------|---|----------------|--------------|---|----------------|--------------|---|----------------|
| C            | D | f <sub>1</sub> | C            | D | f <sub>2</sub> | C            | D | f <sub>3</sub> | C            | D | f <sub>4</sub> |
| 0            | 0 | 0              | 0            | 0 | 0              | 0            | 0 | 1              | 0            | 0 | 1              |
| 0            | 1 | 0              | 0            | 1 | 1              | 0            | 1 | 1              | 0            | 1 | 0              |
| 1            | 0 | 1              | 1            | 0 | 0              | 1            | 0 | 0              | 1            | 0 | 1              |
| 1            | 1 | 0              | 1            | 1 | 0              | 1            | 1 | 1              | 1            | 1 | 1              |

A partir de estas cuatro sub-funciones, construimos las tablas de entrada a los multiplexores:

| $f_1:$ | Entrada | Dirección | Otras Variables |                    |
|--------|---------|-----------|-----------------|--------------------|
|        | $D_0$   | $\bar{C}$ | 0               | Salida: $C\bar{D}$ |
|        | $D_1$   | C         | $\bar{D}$       |                    |

| $f_2:$ | Entrada | Dirección | Otras Variables |            |
|--------|---------|-----------|-----------------|------------|
|        | $D_0$   | $\bar{C}$ | D               | Salida: CD |
|        | $D_1$   | C         | 0               |            |

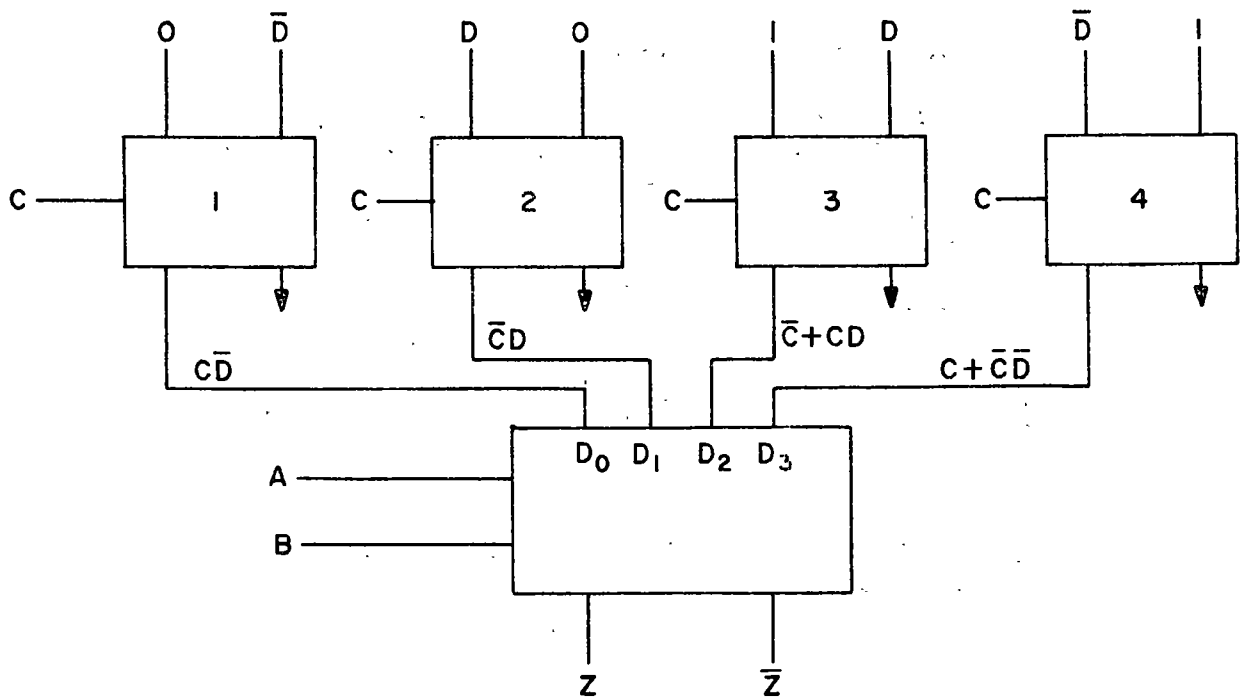
  

| $f_3:$ | Entrada | Dirección | Otras Variables |                  |
|--------|---------|-----------|-----------------|------------------|
|        | $D_0$   | $\bar{C}$ | 1               | Salida: $C + CD$ |
|        | $D_1$   | C         | D               |                  |

| $f_4:$ | Entrada | Dirección | Otras Variables |                              |
|--------|---------|-----------|-----------------|------------------------------|
|        | $D_0$   | $\bar{C}$ | $\bar{D}$       | Salida: $C + \bar{C}\bar{D}$ |
|        | $D_1$   | C         | 1               |                              |

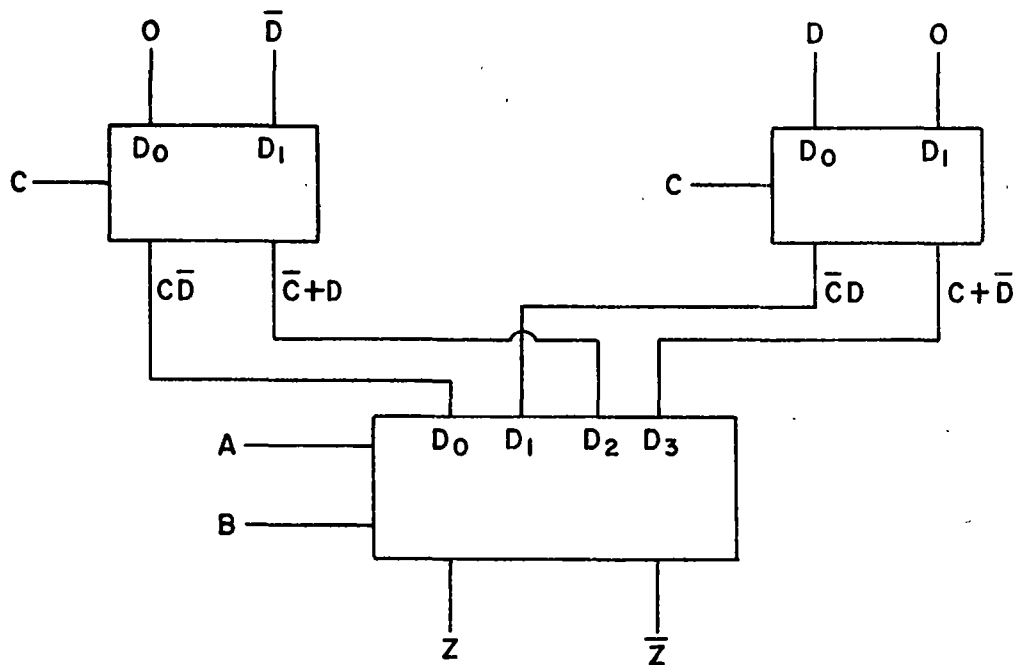
El circuito quedará como se muestra en la figura siguiente:



Sin embargo, si analizamos el complemento de las salidas de los multiplexores 1 y 2 vemos que:

$$\begin{aligned}\overline{CD} &= \overline{C} + D = \overline{C} + CD \\ \overline{CD} &= C + \overline{D} = C + \overline{CD}\end{aligned}$$

Luego, el circuito puede simplificarse aún más:



De los ejemplos anteriores podemos establecer la siguiente regla para el diseño de circuitos de dos niveles empleando multiplexores:

Regla:

- 1.- DIVIDIR LA FUNCION EN DOS PARTES ( $g_1 = g_1(A,B,C)$  Y  $g_2 = g_2(C,D,E)$ ) REDUCIENDO EL PROBLEMA DE 6 VARIABLES A 2 DE 3 VARIABLES.
- 2.- AGRUPAR LOS TERMINOS DE  $g_1(A,B,C)$  CORRESPONDIENTES A CADA COMBINACION DE LAS VARIABLES D, E, F.
- 3.- IMPLEMENTAR CADA TERMINO DE  $g_1(A,B,C)$  CON MUX DE 4 ENTRADAS.
- 4.- ALIMENTAR LAS 8 ENTRADAS DEL MUX DE SALIDA CON LAS SALIDAS DE LOS MUX DEL PRIMER NIVEL.

Veamos a continuación un último ejemplo, más complejo, que ilustre el uso de la regla anterior. El sistema está definido por la siguiente tabla de verdad.

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

En primer lugar tenemos que dividir la función original en dos funciones parciales  $g_1$  y  $g_2$ . Este es el problema más complejo ya que dependiendo de la forma en que se agrupen las variables para formar  $g_1$  y  $g_2$ , se puede obtener un circuito más simple o más complejo. Optemos ahora por la agrupación más obvia:

$$g_1(A,B,C) \text{ y } g_2(D,E,F).$$

El segundo paso es agrupar los términos de  $g_1(A,B,C)$  correspondientes a cada combinación de D, E y F.

Revisando la tabla de verdad, vemos que la función es 1 en las 8 combinaciones posibles de D, E y F. Por lo tanto, tendremos ocho grupos de términos para  $g_1(A,B,C)$ .

En la tabla siguiente se muestra lo anterior:

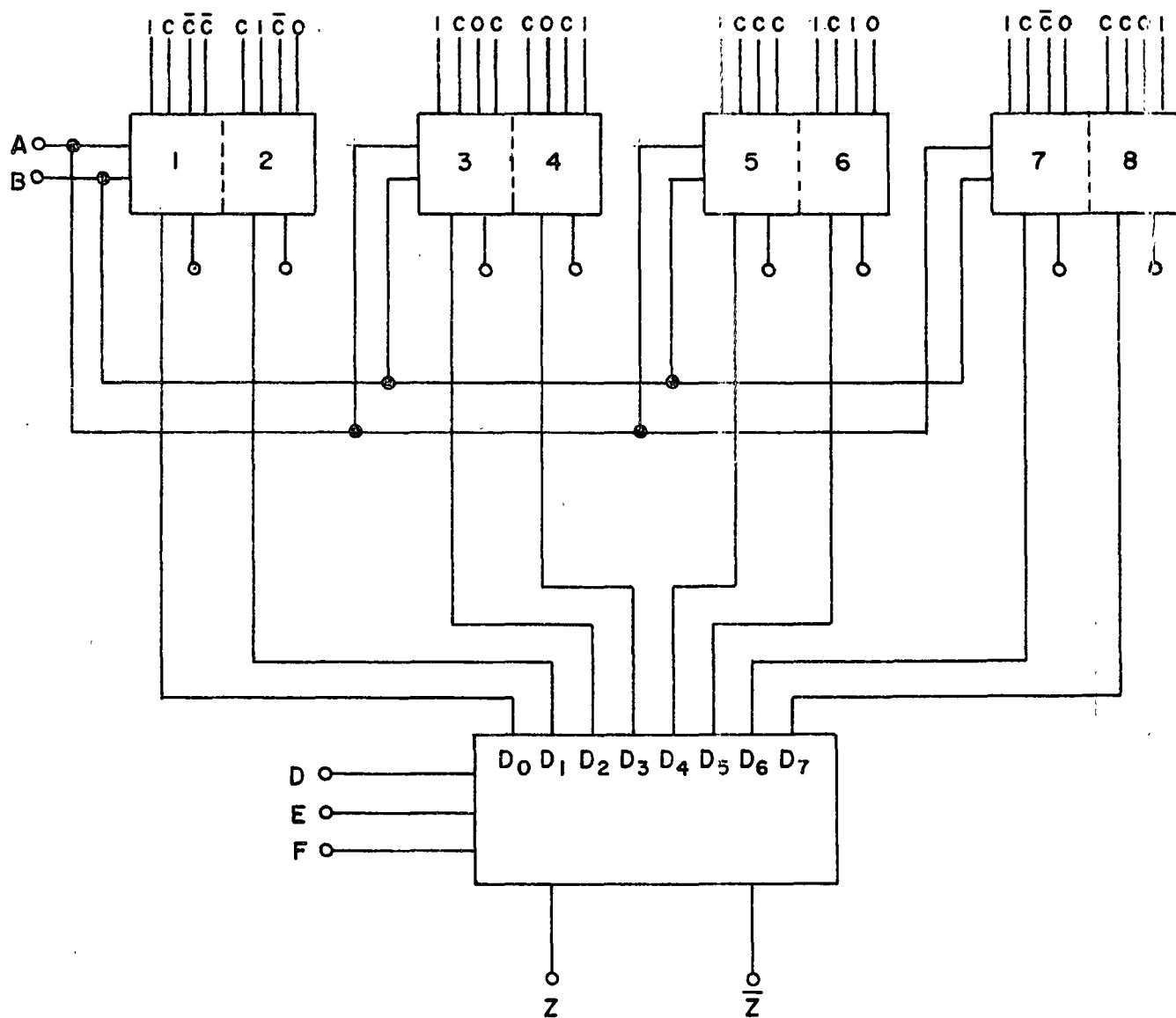
|                |   | A     | B | C | D | E | F | Direcc. |                |    |           |
|----------------|---|-------|---|---|---|---|---|---------|----------------|----|-----------|
|                |   |       |   |   |   |   |   | AB      |                |    |           |
| D <sub>0</sub> | { | 0     | 0 | 0 | 0 | 0 | 0 | Mux 1:  | D <sub>0</sub> | 00 | 1         |
|                |   | 0     | 0 | 1 | 0 | 0 | 0 |         | D <sub>1</sub> | 01 | C         |
|                |   | 0     | 1 | 1 | 0 | 0 | 0 |         | D <sub>2</sub> | 10 | $\bar{C}$ |
|                |   | 1     | 0 | 0 | 0 | 0 | 0 |         | D <sub>3</sub> | 11 | $\bar{C}$ |
|                |   | ----- |   |   |   |   |   | -----   |                |    |           |
| D <sub>1</sub> | { | 0     | 0 | 1 | 0 | 0 | 1 | Mux 2:  | D <sub>0</sub> | 00 | C         |
|                |   | 0     | 1 | 0 | 0 | 0 | 1 |         | D <sub>1</sub> | 01 | 1         |
|                |   | 0     | 1 | 1 | 0 | 0 | 1 |         | D <sub>2</sub> | 10 | $\bar{C}$ |
|                |   | 1     | 0 | 0 | 0 | 0 | 1 |         | D <sub>3</sub> | 11 | 0         |
|                |   | ----- |   |   |   |   |   | -----   |                |    |           |
| D <sub>2</sub> | { | 0     | 0 | 0 | 0 | 1 | 0 | Mux 3:  | D <sub>0</sub> | 00 | 1         |
|                |   | 0     | 0 | 1 | 0 | 1 | 0 |         | D <sub>1</sub> | 01 | C         |
|                |   | 0     | 1 | 1 | 0 | 1 | 0 |         | D <sub>2</sub> | 10 | 0         |
|                |   | 1     | 1 | 1 | 0 | 1 | 0 |         | D <sub>3</sub> | 11 | C         |
|                |   | ----- |   |   |   |   |   | -----   |                |    |           |
| D <sub>3</sub> | { | 0     | 0 | 1 | 0 | 1 | 1 | MUX 4:  | D <sub>0</sub> | 00 | C         |
|                |   | 1     | 0 | 1 | 0 | 1 | 1 |         | D <sub>1</sub> | 01 | 0         |
|                |   | 1     | 1 | 0 | 0 | 1 | 1 |         | D <sub>2</sub> | 10 | C         |
|                |   | 1     | 1 | 1 | 0 | 1 | 1 |         | D <sub>3</sub> | 11 | 1         |
|                |   | ----- |   |   |   |   |   | -----   |                |    |           |
| D <sub>4</sub> | { | 0     | 0 | 0 | 1 | 0 | 0 | MUX 5:  | D <sub>0</sub> | 00 | 1         |
|                |   | 0     | 0 | 1 | 1 | 0 | 0 |         | D <sub>1</sub> | 01 | C         |
|                |   | 1     | 0 | 1 | 1 | 0 | 0 |         | D <sub>2</sub> | 10 | C         |
|                |   | 1     | 1 | 1 | 1 | 0 | 0 |         | D <sub>3</sub> | 11 | C         |
|                |   | ----- |   |   |   |   |   | -----   |                |    |           |
| D <sub>5</sub> | { | 0     | 0 | 0 | 1 | 0 | 1 | MUX 6:  | D <sub>0</sub> | 00 | 1         |
|                |   | 0     | 0 | 1 | 1 | 0 | 1 |         | D <sub>1</sub> | 01 | C         |
|                |   | 0     | 1 | 1 | 1 | 0 | 1 |         | D <sub>2</sub> | 10 | 1         |
|                |   | 1     | 0 | 0 | 1 | 0 | 1 |         | D <sub>3</sub> | 11 | 0         |
|                |   | ----- |   |   |   |   |   | -----   |                |    |           |
| D <sub>6</sub> | { | 0     | 0 | 0 | 1 | 1 | 0 | MUX 7:  | D <sub>0</sub> | 00 | 1         |
|                |   | 0     | 0 | 1 | 1 | 1 | 0 |         | D <sub>1</sub> | 01 | C         |
|                |   | 0     | 1 | 1 | 1 | 1 | 0 |         | D <sub>2</sub> | 10 | C         |
|                |   | 1     | 0 | 0 | 1 | 1 | 0 |         | D <sub>3</sub> | 11 | 0         |
|                |   | ----- |   |   |   |   |   | -----   |                |    |           |
| D <sub>7</sub> | { | 0     | 0 | 1 | 1 | 1 | 1 | MUX 8:  | D <sub>0</sub> | 00 | C         |
|                |   | 0     | 0 | 1 | 1 | 1 | 1 |         | D <sub>1</sub> | 01 | C         |
|                |   | 1     | 1 | 0 | 1 | 1 | 1 |         | D <sub>2</sub> | 10 | 0         |
|                |   | 1     | 1 | 1 | 1 | 1 | 1 |         | D <sub>3</sub> | 11 | 1         |

Entradas al mux de salida.



Debemos realizar la primer columna mediante mux de 4 entradas. Necesitaremos un mux por cada entrada del mux de salida, ya que de la tabla vemos que no hay agrupaciones iguales que pudieran ser generadas con un sólo mux, ni tampoco hay agrupaciones que sean el complemento de la otra.

Como se decía, el problema más complejo, estriba en las agrupaciones que se hagan de las variables con el objeto de obtener las funciones  $g_1$  y  $g_2$ . De esta agrupación dependerá si el número de multiplexores empleados es mínimo.



Como ejemplo de esto, repetamos el ejemplo con otra agrupación:

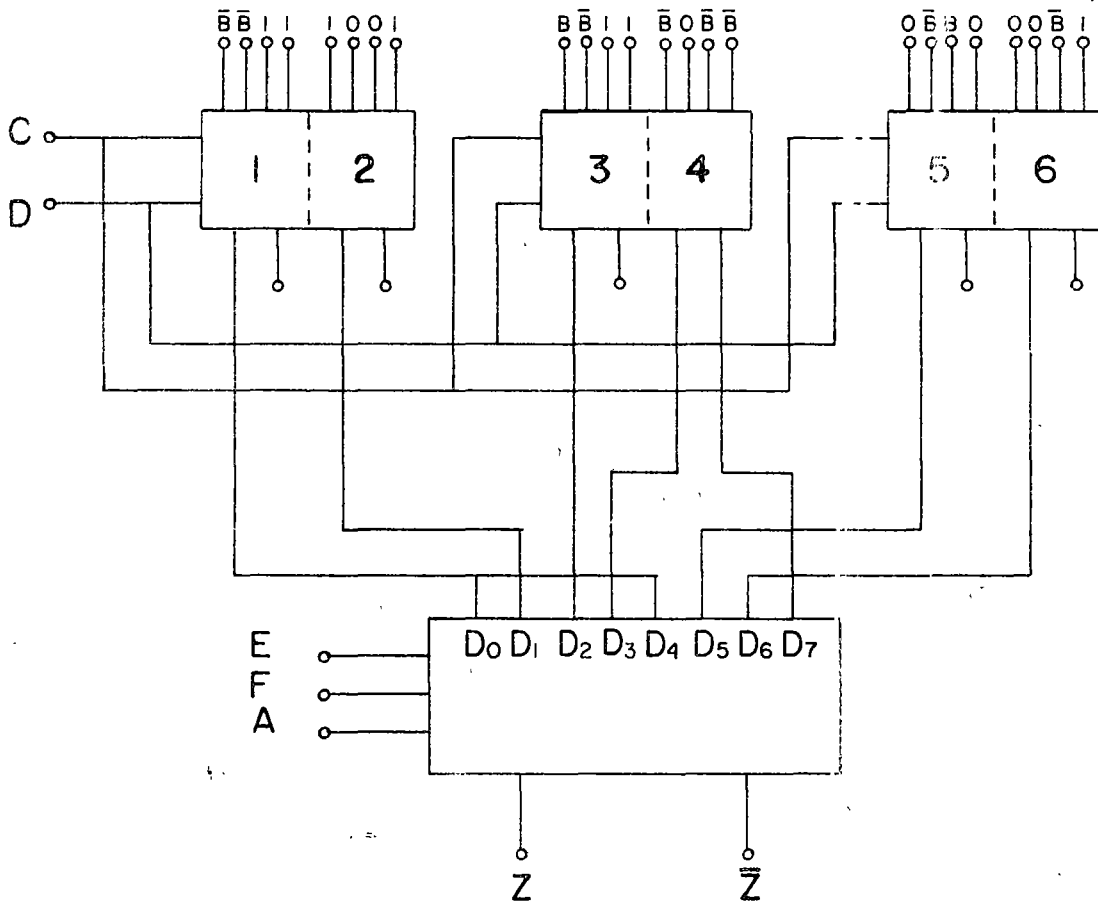
|                | C | D | B | E | F | A |                |    |   |               |
|----------------|---|---|---|---|---|---|----------------|----|---|---------------|
| D <sub>0</sub> | 0 | 0 | 0 | 0 | 0 | 0 | D <sub>0</sub> | 00 | B | MUX 1         |
|                | 1 | 0 | 0 | 0 | 0 | 0 | D <sub>1</sub> | 01 | B |               |
|                | 0 | 1 | 0 | 0 | 0 | 0 | D <sub>2</sub> | 10 | 1 |               |
|                | 1 | 0 | 1 | 0 | 0 | 0 | D <sub>3</sub> | 11 | 1 |               |
|                | 1 | 1 | 1 | 0 | 0 | 0 |                |    |   |               |
| D <sub>1</sub> | 0 | 0 | 0 | 0 | 0 | 1 | D <sub>0</sub> | 00 | 1 | MUX 2         |
|                | 1 | 1 | 0 | 0 | 0 | 1 | D <sub>1</sub> | 01 | 0 |               |
|                | 0 | 0 | 1 | 0 | 0 | 1 | D <sub>2</sub> | 10 | 0 |               |
|                | 1 | 1 | 1 | 0 | 0 | 1 | D <sub>3</sub> | 11 | 1 |               |
|                | 1 | 1 | 1 | 0 | 0 | 1 |                |    |   |               |
| D <sub>2</sub> | 1 | 0 | 0 | 0 | 1 | 0 | D <sub>0</sub> | 00 | B | MUX 3         |
|                | 0 | 1 | 0 | 0 | 1 | 0 | D <sub>1</sub> | 01 | B |               |
|                | 1 | 1 | 0 | 0 | 1 | 0 | D <sub>2</sub> | 10 | 1 |               |
|                | 0 | 0 | 1 | 0 | 1 | 0 | D <sub>3</sub> | 11 | 1 |               |
|                | 1 | 0 | 1 | 0 | 1 | 0 |                |    |   |               |
| D <sub>3</sub> | 0 | 0 | 0 | 0 | 1 | 1 | D <sub>0</sub> | 00 | B | MUX 4         |
|                | 0 | 1 | 0 | 0 | 1 | 1 | D <sub>1</sub> | 01 | B |               |
|                | 1 | 1 | 0 | 0 | 1 | 1 | D <sub>2</sub> | 10 | B |               |
|                | 1 | 1 | 0 | 0 | 1 | 1 | D <sub>3</sub> | 11 | B |               |
| D <sub>4</sub> | 0 | 0 | 0 | 1 | 0 | 0 | D <sub>0</sub> | 00 | B | MUX 1 = MUX 5 |
|                | 1 | 0 | 0 | 1 | 0 | 0 | D <sub>1</sub> | 01 | B |               |
|                | 0 | 1 | 0 | 1 | 0 | 0 | D <sub>2</sub> | 10 | 1 |               |
|                | 1 | 0 | 1 | 1 | 0 | 0 | D <sub>3</sub> | 11 | 1 |               |
|                | 1 | 1 | 1 | 1 | 0 | 0 |                |    |   |               |
| D <sub>5</sub> | 0 | 1 | 0 | 1 | 0 | 1 | D <sub>0</sub> | 00 | 0 | MUX 6         |
|                | 1 | 0 | 1 | 1 | 0 | 1 | D <sub>1</sub> | 01 | 0 |               |
|                | 1 | 0 | 1 | 1 | 0 | 1 | D <sub>2</sub> | 10 | 0 |               |
| D <sub>6</sub> | 1 | 1 | 0 | 1 | 1 | 0 | D <sub>0</sub> | 00 | 0 | MUX 7         |
|                | 1 | 1 | 0 | 1 | 1 | 0 | D <sub>1</sub> | 01 | 0 |               |
|                | 1 | 1 | 1 | 1 | 1 | 0 | D <sub>2</sub> | 10 | 0 |               |
|                | 1 | 1 | 1 | 1 | 1 | 0 | D <sub>3</sub> | 11 | 1 |               |
| D <sub>7</sub> | 1 | 0 | 0 | 1 | 1 | 1 | D <sub>0</sub> | 00 | B | MUX 4 = MUX 8 |
|                | 0 | 0 | 1 | 1 | 1 | 1 | D <sub>1</sub> | 01 | B |               |
|                | 1 | 0 | 1 | 1 | 1 | 1 | D <sub>2</sub> | 10 | 1 |               |
|                | 0 | 1 | 1 | 1 | 1 | 1 | D <sub>3</sub> | 11 | B |               |

Vemos entonces que con esta agrupación obtenemos un ahorro de 2 mux, ya que:

Mux 1 = Mux 5

Mux 8 = Mux 4

El circuito queda:



Criterios:

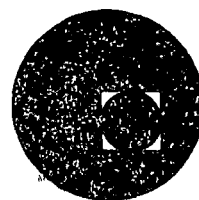
- 1) Utilizar señales multiplexadas que se empleen en más de una de las entradas del Mux de salida.
- 2) Seleccionar combinaciones que maximicen pares de funciones multiplexadas que sean complementos una de la otra.

Por último cabe mencionar que la misma función se podría realizar con 4 mux de 8 entradas en el primer nivel y uno de 4 en el 2o. nivel. Sin embargo, en el primer nivel conviene emplear mux más baratos y por lo tanto, para el ejemplo, los más convenientes son los de 4 entradas ( vienen 2/ circuito integrado)

---



centro de educación continua  
división de estudios superiores  
facultad de ingeniería, unam



APLICACIONES DE CIRCUITOS INTEGRADOS AL  
DISEÑO DIGITAL

CAPITULO 4: CIRCUITOS BIESTABLES, MONOESTABLES Y  
ASTABLES

ING. ARTURO GONZALEZ HERMOSILLO  
JUNIO 1978

## CAPITULO 4: BIESTABLES Y REGISTROS DE CORRIMIENTO.

**4.0 Introducción.** Dentro de la electrónica digital, se utilizan diferentes tipos de elementos, de los más importantes en la división de los circuitos secuenciales, son los biestables. Un biestable es un elemento con capacidad de almacenar información; dicho de otra forma es un "elemento de memoria", no siendo estos los únicos con esta capacidad; entre otros, se pueden enunciar las cintas magnéticas, cintas de papel, tambores magnéticos, etc. pero por los requerimientos en el proceso de información, el grado de avance de la tecnología y sobre todo el estado actual en la microintegración de circuitos, los biestables tienen un especial interés. En la actualidad, existen circuitos integrados de no más de  $2 \text{ cm}^2$  en cuyo interior se encuentra un arreglo de memoria con más de 4000 biestables; una computadora puede tener un porcentaje alto de biestables dentro de los elementos electrónicos que utiliza; el funcionamiento de un reloj electrónico de pulsera es en base de biestables. En general dentro de la electrónica digital, es difícil encontrar un sistema secuencial en donde su memoria no esté implementada con biestables.

Hay diferentes tipos de biestables; por su comportamiento en función de sus entradas de control reciben un nombre en especial. La forma de entender el comportamiento o el tipo de biestable, es a través de su tabla de transición. Esta tabla de transición, nos determina el estado sucesor del biestable a un vector entrada.

**4.1 Biestable Básico.** La forma más sencilla de pensar en la construcción de un biestable, es con la realimentación lógica de dos inversores, suponiendo que el primer inversor se encuentra en un estado lógico  $\bar{A}$ ; esto forzará a la salida del segundo a dar la entrada negada  $A$ , realimentándose al primero y así el arreglo permanecerá en un estado estable (figura 4-1).

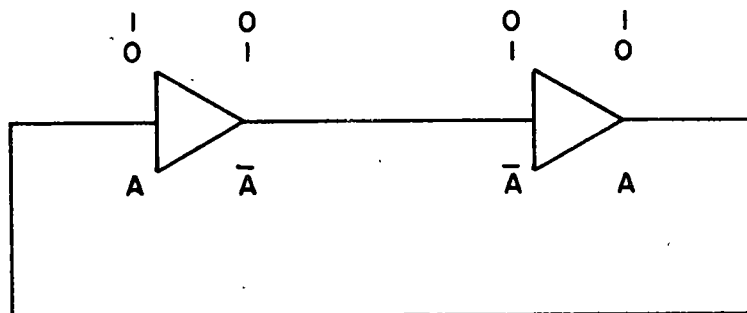


Figura (4-1)

Los inversores en su versión más sencilla es como lo muestra la Figura 4-2 de tal forma que un biestable básico queda implementado como se muestra en la figura 4-3.

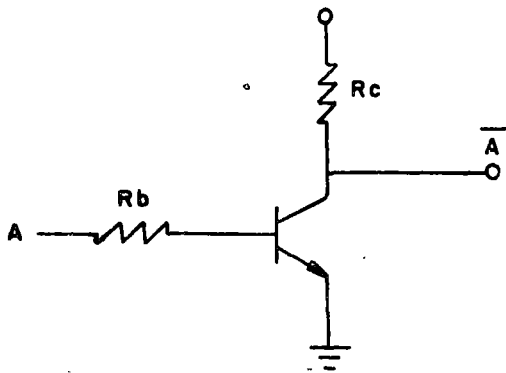


Figura 4-2.

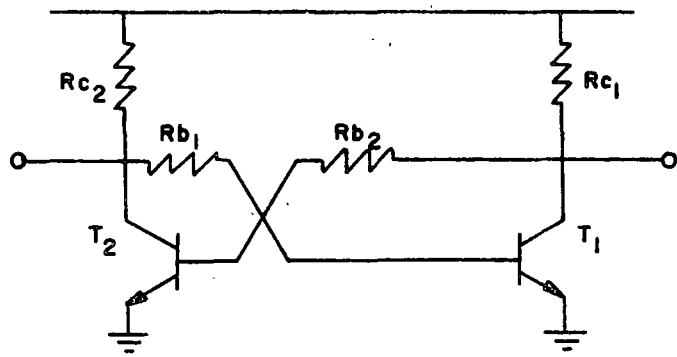


Figura 4-3.

Como es sabido, la forma de sensar el estado del biestable es a través de niveles de voltaje. Para este caso, los valores de resistencia con relación a la beta mínima del transistor, deben de ser tales, que cuando un transistor se encuentre en corte, produzca la saturación del otro; entonces se debe asegurar que la corriente de base alimentada al transistor en saturación, sea mayor que la corriente requerida para su saturación.

Estando un transistor en saturación, produce el corte en el otro, de tal forma que las salidas pueden ser directamente los colectores de los transistores, y si se encuentran trabajando en forma complementaria al llamar a un colector salida  $Q$ , el colector del otro transistor será  $\bar{Q}$ .

Las entradas de control para el circuito son como las mostradas en la figura 4-4, la acción que producen estas entradas con un voltaje alto en "S" o "R", es la saturación del transistor correspondiente.

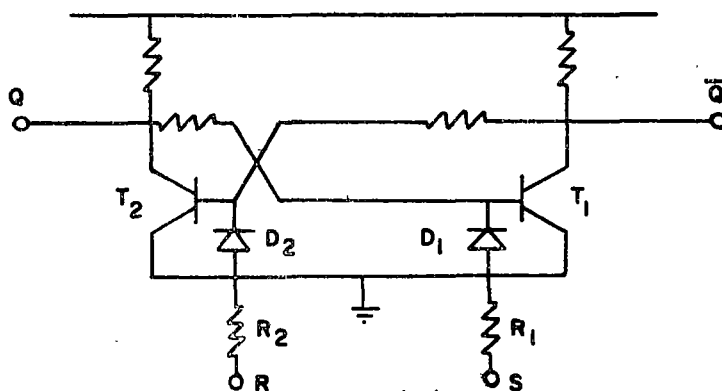


Figura 4-4.

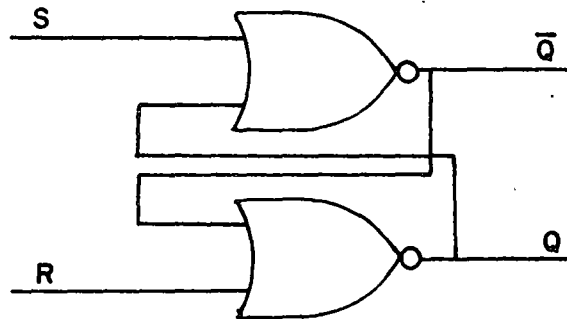
Si S es un voltaje alto, ( $V_H$ ),  $D_1$  conduce y satura  $T_1$ ; al saturar  $T_1$  corta a  $T_2$  y la salida  $Q$  será alta mientras  $\bar{Q}$  es baja, entonces al tener alta la entrada S produce la acción de poner al biestable alto ( $Q=V_H$ ) y se le llama entrada "SET". Por simetría del circuito, la acción de R, para cuando se le aplica un voltaje alto será saturar a  $T_2$  quedando  $Q$  baja y  $\bar{Q}$  alta por lo que a esta entrada se le denomina "RESET".

Es obvio que no se podrán poner ambos transistores en saturación simultáneamente; esto implica que no podrán ser ambas entradas SET y RESET altas al mismo tiempo. De lo anterior se puede inferir la tabla de transición de este biestable.

| S | R | $Q_{t+1}$ |
|---|---|-----------|
| 0 | 0 | $Q_t$     |
| 0 | 1 | 0         |
| 1 | 0 | 1         |
| 1 | 1 | *         |

\* No definido

4.2 Latch. El Latch es un arreglo de dos compuertas realimentadas, que produce estados estables. De aquí que con la ayuda de algunas compuertas más sea utilizado como biestable.



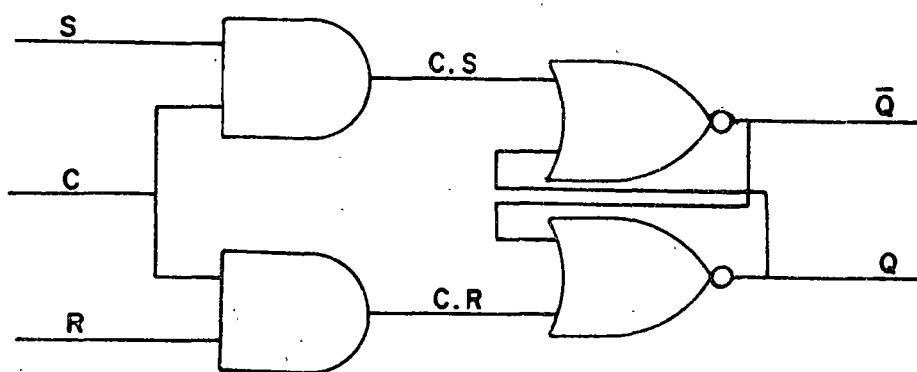
Circuito Latch.  
Figura 4-5.

El análisis del funcionamiento es como sigue: Asumimos que la salida  $Q=0$  y  $S=R=0$ , entonces  $\overline{Q}$  la compuerta superior tendrá entradas  $Q=S=0$ ; su salida será entonces  $0+0 = 1$  y le llamaremos  $\overline{Q}$ . La compuerta inferior, tiene entradas  $Q=1$  y  $R=0$ ; su salida será  $0+1 = 0$  que es el estado donde habíamos empezado.

Suponiendo que  $R$  cambia rápidamente a 1. La compuerta inferior tiene entonces entradas  $\overline{Q}=R=1$  y su salida será  $Q = \overline{Q}+R = 1+1 = 0$  y permanece sin cambio, que es la condición de RESET. Sin embargo si  $S$  es la que cambia a 1, la salida de la compuerta superior será  $\overline{Q} = \overline{Q}+S = 0+1 = 0$  y  $\overline{Q}$  cambiará a cero. Como  $\overline{Q}$  ha cambiado a cero, las entradas de la compuerta inferior son afectadas como sigue:  $\overline{Q} = 0$ ,  $R=0$ , la salida  $Q$  será  $Q = \overline{Q}+R = 0+0 = 1$  cambiará a 1, condición de SET.

Biestable SR. A un latch, al agregársele dos compuertas AND como lo muestra la figura 6, se podrá tener una señal de sincronía para el cambio del latch, quedando un biestable S.R.





Biestado S.R.  
Figura 4-6.

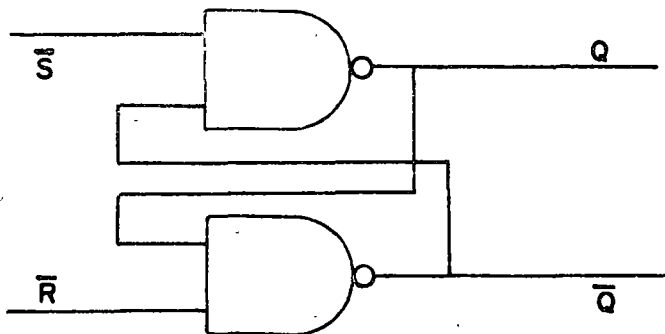
Para cuando  $C=0$  no importa el estado de  $S$  y  $R$ ,  $C.S.=C.R.=0$  y el latch no cambia su estado; para cuando  $C=1$  el valor de  $C.S.$  y  $C.R.$  está en función de  $S$  y  $R$ ; el estado del latch será entonces dependiente de sus entradas. A la entrada  $C$  se le llama control de reloj;  $Q_{t+1}$  se entiende como el estado del biestado después de un pulso de reloj en  $C$ . La tabla de transición de este biestado será:

| S | R | $Q_{t+1}$ |
|---|---|-----------|
| 0 | 0 | $Q_t$     |
| 0 | 1 | 0         |
| 1 | 0 | 1         |
| 1 | 1 | *         |

$$Q^{t+1} = S^t + \bar{R}^t Q^t$$

\* No determinado

El mismo análisis de latch, se puede hacer con compuertas NAND quedando el arreglo y tabla de transición de la forma:



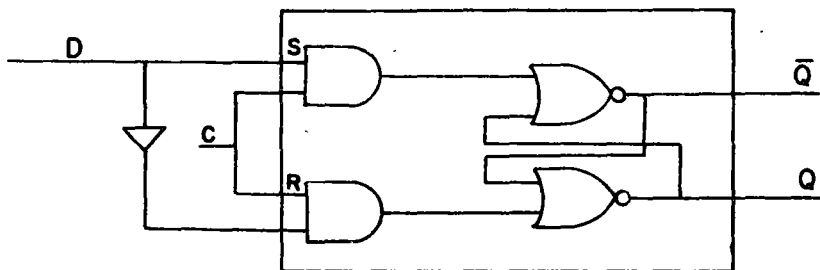
| S | R | $Q_{t+1}$ |               |
|---|---|-----------|---------------|
| 0 | 0 | *         | * No definido |
| 0 | 1 | 1         |               |
| 1 | 0 | 0         |               |
| 1 | 1 | $Q_t$     |               |

Figura 4-7

4.3. Biestable D. El nombre de biestable D es tomado de "Delay", la lógica de este biestable es que la entrada que le sea presentada, será el sucesor de la salida Q del biestable; su tabla de transición será:

| D | $Q_{t+1}$ |
|---|-----------|
| 0 | 0         |
| 1 | 1         |

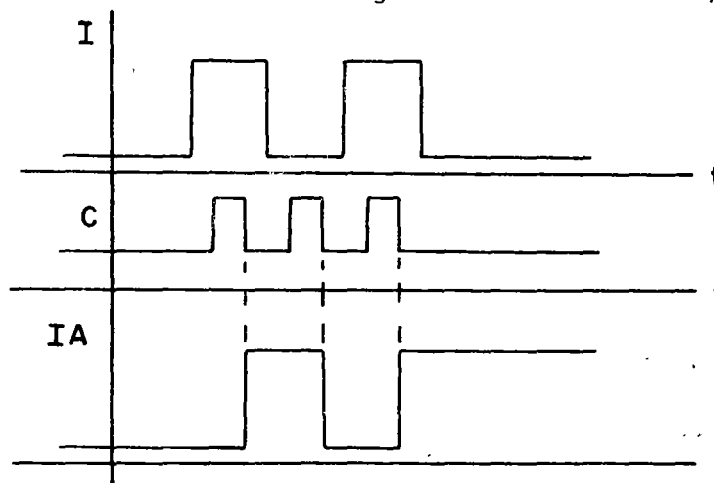
Observando la tabla de transición de un biestable SR, si trabajamos únicamente para cuando  $S=\bar{R}$  podremos construir un biestable D.



Biestable D,  
Figura 4-8.

De tal forma que si  $D=0$  implica que  $S=0$ ;  $R=1$ ; la salida  $Q=0$ . Si  $D=1$  implica que  $S=1$ ,  $R=0$  y la salida será  $Q=1$ . Luego entonces cumple con la tabla de transición propuesta.

4.4. Principio de Maestro Esclavo. En ocasiones el conectar biestables en "cadena" como los que hasta ahora hemos visto, nos produce salidas no deseables. Un ejemplo de un efecto no deseado es el siguiente: Supóngase que se quiere construir un registro en donde se desea almacenar información binaria de 3 bits, se requiere que la forma de almacenarla sea en serie como lo muestra la siguiente carta de tiempos:



I es una señal digital que se encuentra variando en el tiempo; se sincroniza un reloj C que determina el tiempo donde I es estable y puede ser almacenada; IA es la información almacenada en los biestables. Si

se construye un arreglo de biestables de los que se han estudiado como lo muestra la figura 4-9 para almacenar esta información, al momento de ser cierto C los tres biestables tomarán la información; esto se debe a que la salida de los biestables no está desacoplada de sus entradas.

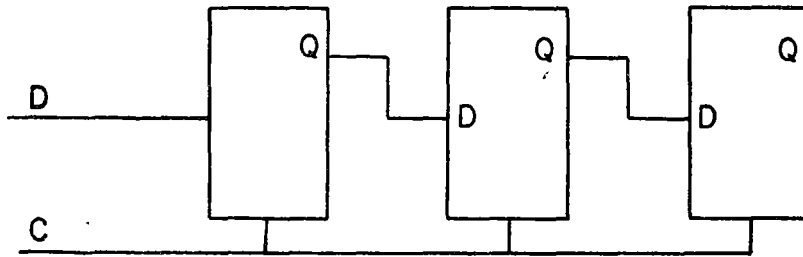


Figura 4-9.

Si se hace un desacoplamiento de las entradas de los biestables con sus salidas, de tal forma que las entradas censan la información a la subida del reloj y la pasen a la salida de los mismos en la bajada del reloj, tendremos:

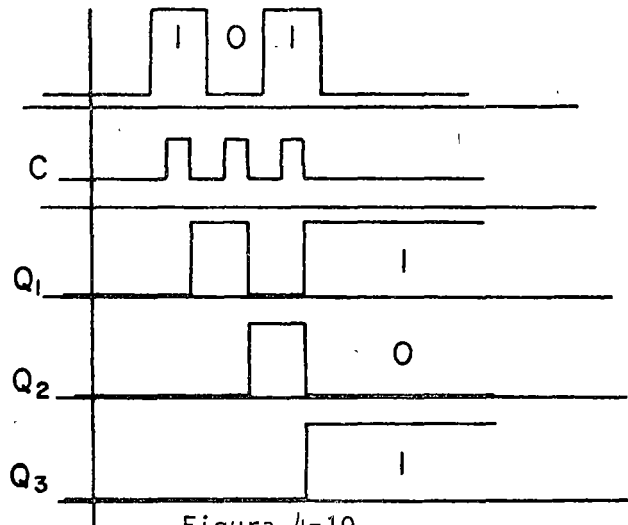


Figura 4-10.

La información queda entonces almacenada en las salidas Q de los biestables.

La siguiente forma de desacoplar las salidas de las entradas se llama Maestro-esclavo y su funcionamiento es a través de dos latches como lo muestra la figura 4-11; el primero forma la sección maestro, y el segundo la sección esclavo.

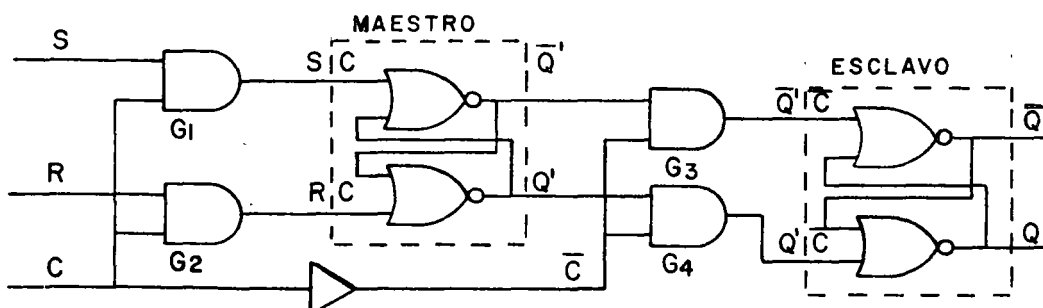


Figura 4-11.

Para cuando  $C=0$  las compuertas  $G_1$  y  $G_2$  se encuentran "abiertas"; S.C. y R.C. son ambas iguales a cero, haciendo que maestro no cambia de estado;  $G_3$  y  $G_4$  estarán "cerradas" y esclavo tomará la información de Maestro. En la transición de  $C=1$ ,  $G_3$  y  $G_4$  se abren y  $\bar{C}=0$ ;  $\bar{Q}$   $\bar{C}=Q$   $\bar{C}=0$   $\bar{Q}$  y esclavo almacena la información;  $C=1$ , S.C=S, R.C=R y maestro cambiará en función de S y R, al tiempo de que C cambia de uno a cero, maestro es "bloqueado" y "habilitado" esclavo; hasta este tiempo Q será función de S y R. El arreglo de la figura 10.a nos representa un biestable "SET RESET MAESTRO ESCLAVO".

4.5. Biastable Toggle. El comportamiento del biestable Toggle (T) es que en la transición del reloj, ya sea de 0 1 ó de 1 0, dependiendo de su implementación, la salida Q del biestable sea el negado del estado anterior; el funcionamiento es mostrado en el diagrama siguiente.

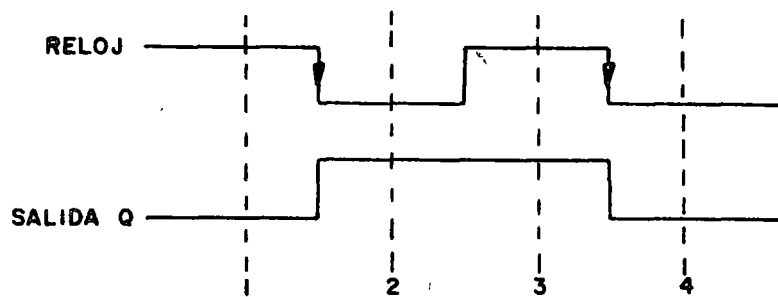


Figura 11.a.

Para nuestro análisis tomaremos al biestable que cambia en la bajada del reloj. El análisis del circuito se hace con los métodos convencionales. Primero del comportamiento del biestable construimos una tabla de estados; esta tabla nos da la información de los estados sucesores y los valores de Q.

| $\bar{C}$ | C | Q   |
|-----------|---|-----|
| 1         | 2 | 1 0 |
| 2         | 2 | 3 1 |
| 3         | 4 | 3 1 |
| 4         | 4 | 1 0 |

Haciendo la asignación de estados llegamos a la tabla de excitación del biestable.

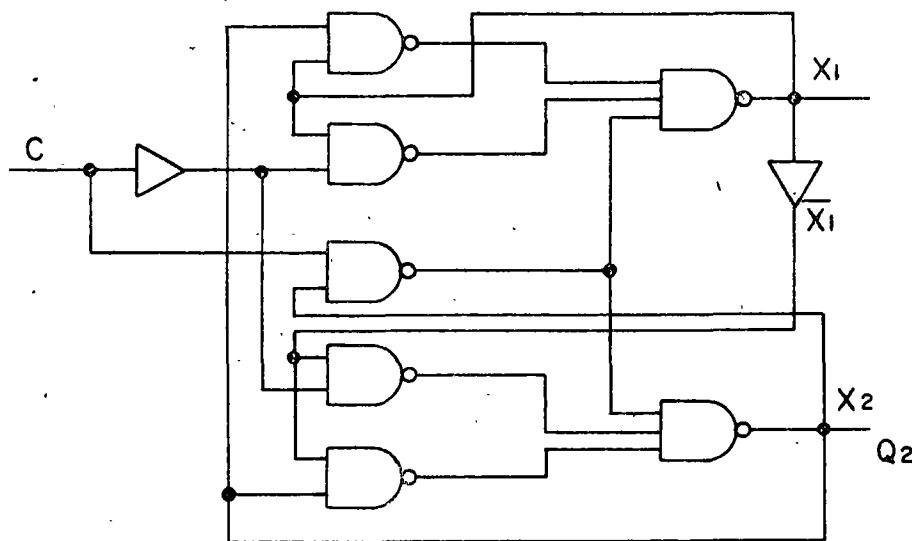
| $X_1 X_2$ | C=0 | C=1 | Q |
|-----------|-----|-----|---|
| 0 0       | 01  | 00  | 0 |
| 0 1       | 01  | 11  | 1 |
| 1 1       | 10  | 11  | 1 |
| 1 0       | 10  | 00  | 0 |

Resolviendo por Karnaugh.

$$X_1 = \bar{C}X_1 + CX_2 + X_1X_2$$

$$X_2 = \bar{C}\bar{X}_1 + CX_2 + \bar{X}_1X_2$$

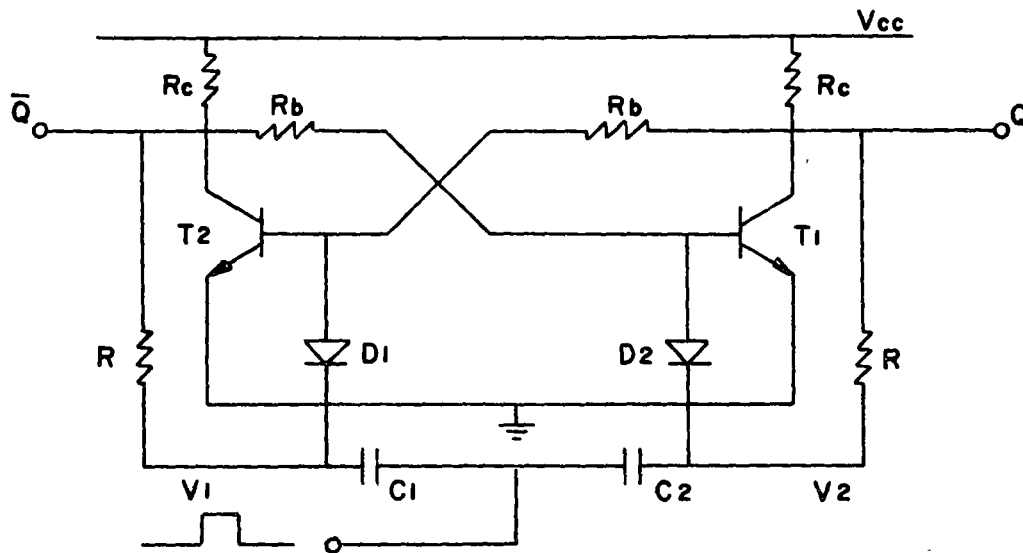
Y el circuito correspondiente a las ecuaciones es mostrado en la figura 10.



Biestable Toggle.  
Figura 4-12.

Otra forma de implementar un biestable Toggle, es a partir de una celda básica; el circuito de disparo opera este caso, trabaja con la diferenciación de la señal de sincronía o reloj.

Los pulsos obtenidos de esta diferenciación son negativos y actúan directamente sobre las bases de los transistores de la celda; en cada ciclo de reloj, se produce el pulso que corta al transistor en saturación, saturándose el otro; al siguiente ciclo, el transistor que se encuentra en saturación, ahora es cortado y así sucesivamente, para cada ciclo de reloj.



Biastable Toggle.  
Figura 4-13.

La figura 4-13 muestra una celda básica, con un circuito de disparo formado por  $D_1$ ,  $D_2$ ,  $C_1$ ,  $C_2$ ,  $R_1$  y  $R_2$ , con este circuito la celda básica actúa como un biestable Toggle. La interacción de la celda y el circuito de disparo es del modo siguiente: Suponiendo que  $T_1$  se encuentra en saturación, para el frente de onda positivo del reloj se tendrán pulsos positivos en  $V_1$  y  $V_2$ , los diodos  $D_1$  y  $D_2$  se encontrarán entonces en circuito abierto y no habrá cambio en las condiciones del biestable; antes de producirse el frente de onda negativo del reloj, si  $T_1$  está en saturación, el capacitor  $C_1$  se encuentra cargado en forma positiva hacia la entrada del reloj, ya que por su otro extremo tendremos un voltaje casi igual a cero, que es el voltaje de colector de saturación de  $T_1$ ; al momento de la transición o frente de onda negativo, por la carga de  $C_1$  se tendrá un pulso negativo en  $V_2$  que hace que conduzca  $D_2$ , y fuerza a la base de  $T_1$  a un voltaje negativo, cortándose este transistor. Para el siguiente ciclo de reloj, y por simetría del circuito en forma recíproca pasará de corte a saturación  $T_1$ . Esta acción de  $T_1$  se muestra gráficamente en la figura 4-14.

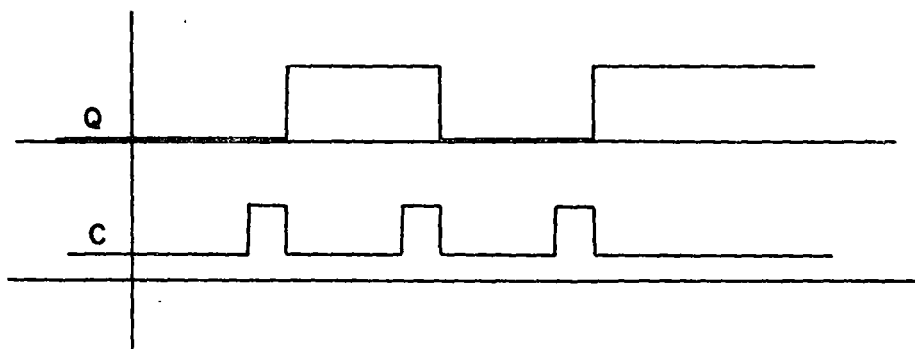


Figura 4-14.

4.6. Biastable J-K. Los biestables JK, posiblemente sean los más usados dentro de la familia de los biestables, es debido a que en sus dos entradas, J y K es posible conectarlas ambas a uno lógico, rompiéndose

la ambigüedad para cuando se conectaban SET y RESET altas al mismo tiempo.

Para cuando  $J=K=1$ , la transición del biestable es el negado de su estado anterior, la figura 4-15, presenta su tabla de transición.

| J | K | $Q_{t+1}$   |
|---|---|-------------|
| 0 | 0 | $Q_t$       |
| 0 | 1 | 0           |
| 1 | 0 | 1           |
| 1 | 1 | $\bar{Q}_t$ |

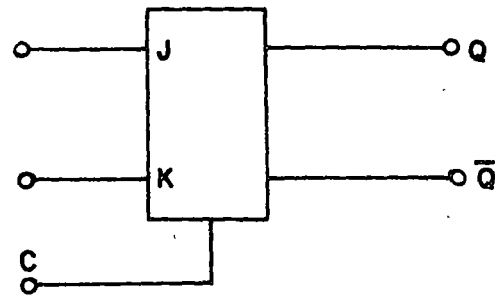


Figura 4-15.

La construcción de este biestable, es partiendo de un biestable "SET" "RESET" maestro esclavo y permitiendo con lógica implementada, la combinación de ambas entradas altas, esto es mostrado en la figura 4-16.

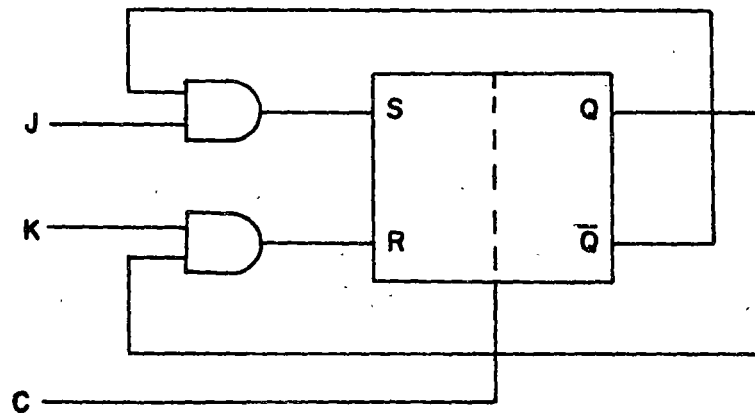


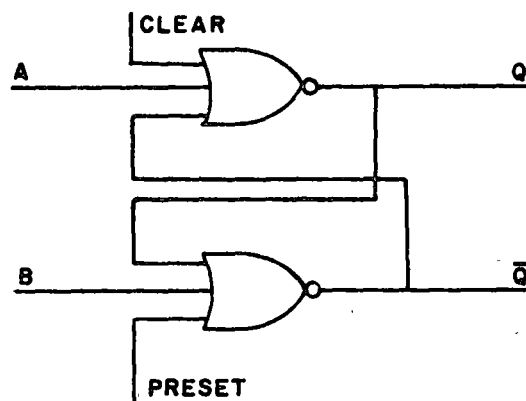
Figura 4-16.

Se observa que  $S=J\bar{Q}$ ;  $R=KQ$  de tal forma que si se quiere que  $Q=Q_n$  implica que  $S=0$  y  $R=0$  y si  $J=0$  y  $K=0$  entonces  $S=R=0$  cumpliendo; si se quiere que  $Q=0$  se debe de tener  $J=0$ ,  $K=1$ , de donde no importando  $Q$ ,  $S=0$ , y si  $Q=1$ ,  $R=1$  haciendo que  $Q=0$  y si  $Q=0$ ,  $R=0$  y queda que  $Q_{n+1}=Q_n=0$  que también cumple; para hacer  $Q=1$ , las entradas deben de ser  $J=1$ ,  $K=0$  implica que  $R=0$  y  $S=\bar{Q}_n$  de tal forma que si  $Q_n=1$ ,  $S=0$  y  $Q_{n+1}=Q_n=1$ , y si  $Q_n=0$ ,  $S=1$  y  $Q_{n+1}=1$  cumpliendo para ambos casos; por último si  $J=K=1$  tendremos que si  $Q_n=0$ , implica que  $R=0$ ,  $S=1$  y  $Q_{n+1}=1=Q_n$ ; si  $Q_n=1$  implica que  $R=1$ ,  $S=0$  y  $Q_{n+1}=0=Q_n$  cumpliendo el último caso.

4.6.1. Clear y Preset. En ocasiones, los biestables tienen una o dos entradas llamadas Clear y Preset.

La acción del Clear es fijar la salida Q del biestable a un cero lógico; la acción del Preset es fijar la salida Q al uno lógico, ambas entradas funcionan en modo asíncrono, esto es que son independientes del reloj. Por la acción que tienen estas entradas, es obvio que no podrán ser activadas simultáneamente.

La implementación del Clear y el Preset, se hace directamente sobre el latch de salida del biestable (Figura 4-17) de tal forma que es independiente del tipo de biestable que se trate.



Para cuando Clear y Preset son ceros, el funcionamiento del latch no se ve afectado y la salida Q será función de A y B como ya se analizó; para cuando  $A = B = 0$  y Clear es alto, se fuerza  $Q = \overline{A+Q+C} = \overline{0+Q+1} = 0$  y  $\overline{Q} = \overline{Q+B+P} = \overline{0+0+0} = 1$  siendo el efecto esperado del Clear; para cuando  $A = B = 0$  y el Preset es alto, se tiene que  $\overline{Q} = \overline{B+Q+P} = \overline{0+Q+1} = 0$  y  $Q = \overline{C+B+Q} = \overline{0+0+0} = 1$  siendo el efecto esperado del Preset. Como ya se dijo, A y B tienen que ser ceros; así que la implementación del biestable es tal que, el latch de salida se debe de encontrar cerrado para cuando el reloj está bajo.

4.7 Registros de Corrimiento. Un registro de corrimiento es un arreglo de biestables conectados en cascada; la salida de los biestables, es conectada directamente a la entrada del siguiente. Suponiendo que se tiene un registro de n biestables, el biestable  $i < n$ ; tendrá a su entrada la  $Q_{i-1}$  y a su salida la entrada  $i+1$ . Por características de la tecnología en circuitos integrados, los registros de corrimiento integrados tienen limitaciones en cuanto al número de puertos de acceso a la lógica interna del arreglo de biestables, esto es que el número de pines de un circuito integrado está limitado; por lo anterior, los registros en donde se desea acceder las salidas Q de todos los biestables, no podrán ser arreglos mayores de 8 biestables y en el caso donde se desea acceder a  $Q_i$  y  $\overline{Q}_i$  no podrá ser mayor a 4 biestables.

Para la implementación de registros de corrimiento en base a biestables integrados, el número de biestables conectados en cascada, prácticamente no tienen límite. Para este tipo de registros lo común es utilizar biestables J.K., conectados como lo muestra la figura 4-17.



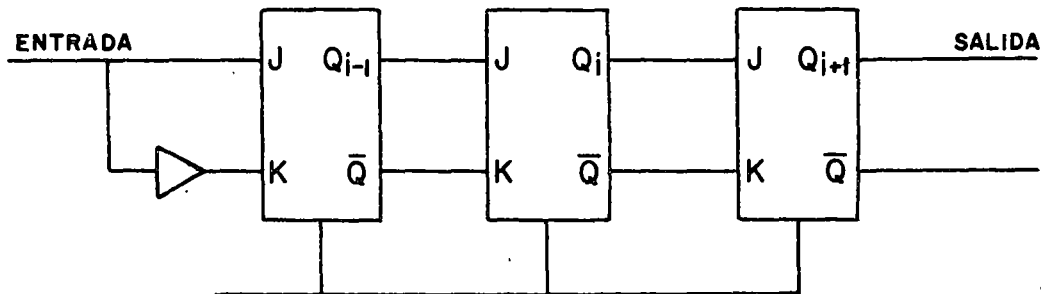
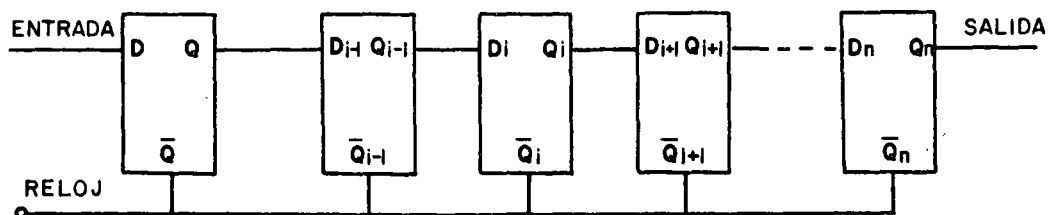


Figura 4-17

Como se observa de la figura, las entradas de los biestables están conectadas de tal forma que  $J=K$ . Esto de la tabla de transición de los biestables implica que si  $J=1$ ,  $Q_{n+1} = 1$  y si  $J = 0$ ,  $Q_{n+1} = 0$ ; entonces  $J=Q_{n+1}$ .

El funcionamiento es de la forma siguiente: Al tener los biestables el reloj común, cuando el reloj hace la transición de cero a uno lógico, las secciones maestro de los biestables se abren y censan la información del biestable anterior; al estar el reloj alto, la sección esclavo de los biestables es cerrada y los biestables no producen cambio en sus salidas; al hacer el reloj la transición del uno al cero lógico, las secciones maestro de cada uno de los biestables se cierran y capturan el dato para cuando el reloj se encontraba alto; en este mismo tiempo, la sección esclavo se abre y toma el dato capturado por maestro, produciéndose que el dato que se encontraba en la salida  $Q_{i-1}$  del biestable  $i-1$ , fue censado y capturado por el biestable  $i$ . También se pueden implementar registros de corrimiento, con biestable D y S.R. El funcionamiento es análogo para el caso cuando son implementados con biestables J.K., de la tabla de transición del biestable D, se sabe que  $Q_{n+1} = D$  de tal forma que con biestables D ahora queda el circuito implementado como en la figura 4-18.



Para la implementación con biestables S y R, la figura 4-19 muestra la interconexión haciendo trabajar al biestable  $i$  con  $S = \bar{R}$ , y si  $S = 1$ ;  $R = 0$  implica que  $Q_{n+1} = 1$ , y si  $S = 0$   $R = 1$  implica que  $Q_{n+1} = 0$  concluyendo que  $Q_{n+1} = S$ .

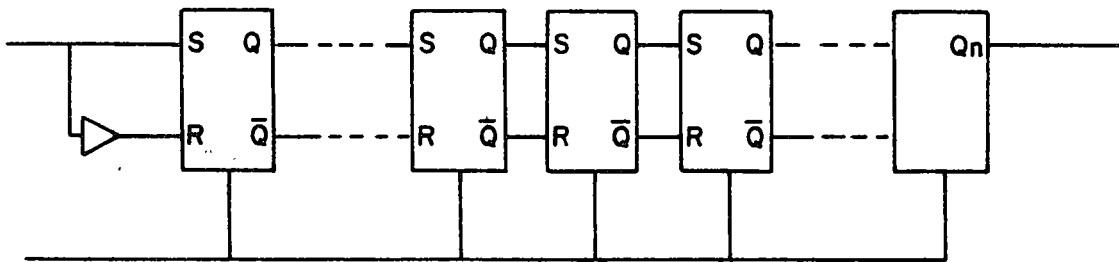


Figura 4-19

Un registro de corrimiento puede ser cargado en paralelo y posteriormente hacerse el corrimiento. Las señales por medio de las cuales se hace la carga en paralelo, son el "Clear" y el "Preset". Estas señales fijan un cero y un uno lógico respectivamente en la salida del biestable. La figura 4-20 muestra un registro con carga en paralelo.

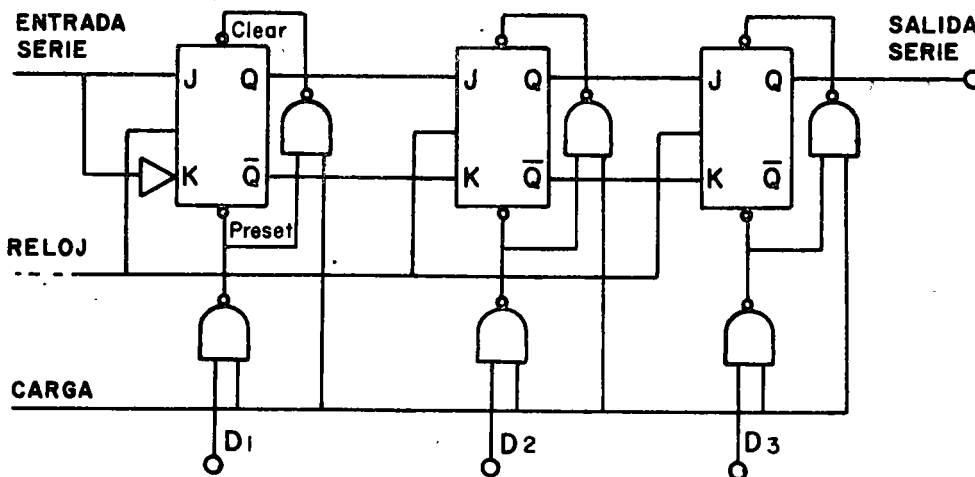


Figura 4-20

Las funciones lógicas para el Clear y Preset son diseñadas en la siguiente forma: Se requiere que para cuando carga sea igual a cero, el Clear y Preset sean altos para que el biestable opere en modo síncrono y hacer el corrimiento; para cuando carga es alto, el dato debe ser cargado en el biestable correspondiente, activándose el Clear o el Preset del mismo. La lógica anterior se interpreta de la siguiente tabla de verdad.

| Carga | Dato | Clear | Preset |
|-------|------|-------|--------|
| 0     | 0    | 1     | 1      |
| 0     | 1    | 1     | 1      |
| 1     | 0    | 0     | 1      |
| 1     | 1    | 1     | 0      |

de la tabla de verdad se observa que para el Preset.

$$\overline{P} = C \text{ o } D$$

$$P = \overline{C \text{ o } D}$$

para el Clear,

$$\overline{C} = \overline{CD}$$

$$C = \overline{\overline{CD}}$$

$$C = \overline{C \text{ o } (C+D)}$$

$$C = C \text{ o } CD$$

Un arreglo de biestables junto con lógica combinacional, puede dar la implementación de un registro de corrimiento en donde el corrimiento sea a la derecha o izquierda, a elección de una señal de modo de operación (figura 4-21).

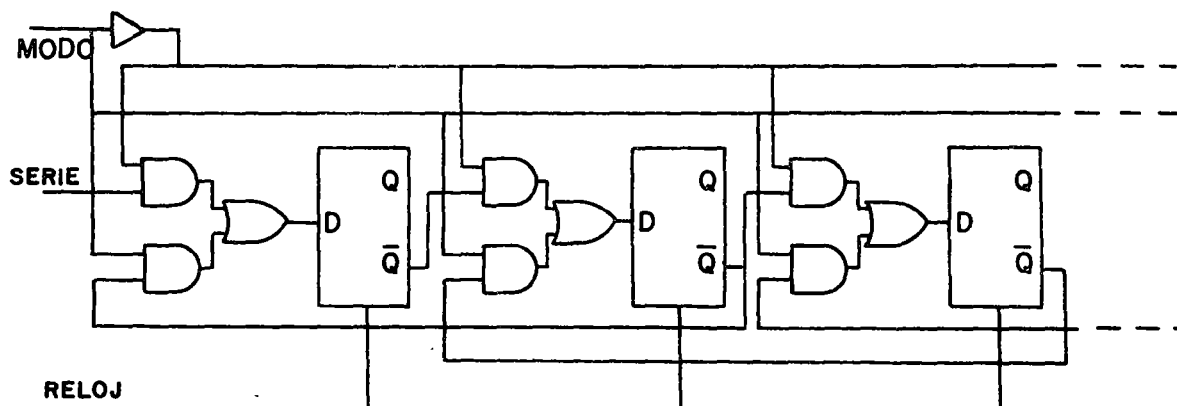


Figura 4-21  
Registro de corrimiento  
a la izquierda y derecha.

El diseño de la función lógica para la entrada  $D_n$  es: Si el modo es alto, el corrimiento se hace hacia la izquierda,<sup>n</sup> y si el modo es bajo el corrimiento se hace a la derecha. La forma de operación de este Registro se representa en la siguiente tabla.

| Modo | $Q_{n-1}$ | $Q_{n-1}$ | $Q_n$ |
|------|-----------|-----------|-------|
| 0    | 0         | 0         | 0     |
| 0    | 0         | 1         | 0     |
| 0    | 1         | 0         | 1     |
| 0    | 1         | 1         | 1     |
| 1    | 0         | 0         | 0     |
| 1    | 0         | 1         | 1     |
| 1    | 1         | 0         | 0     |
| 1    | 1         | 1         | 1     |

de la tabla se obtiene la implementación de  $D_n$  con mapas de Karnaught.

$$Q_{n+1} \quad Q_{n-1}$$

Modo

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

$$D = M Q_{n+1} + \overline{M} Q_{n-1}$$

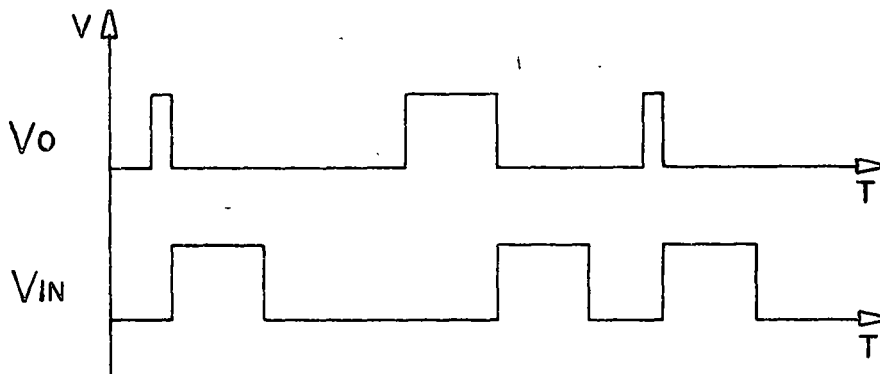
$$D = M Q_{n+1} + \overline{M} Q_{n-1}$$

#### 4.3. Monoestable-(One-Shot).

##### Descripción.

El monoestable es una compuerta lógica regenerativa, tiene como característica que su salida es un pulso de duración fija según la constante de tiempo de un capacitor.

Idealmente el funcionamiento del monoestable se ilustra en la figura 4.22; en donde se observa que a cada bajada en la señal de entrada se sucede un pulso de salida.



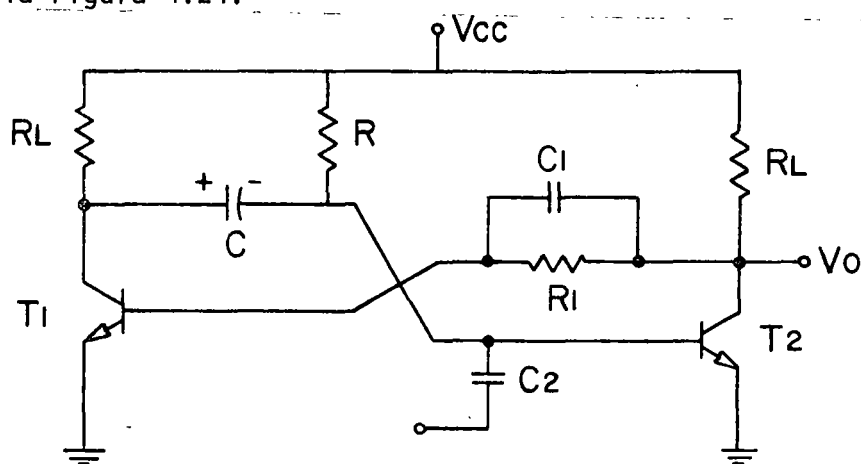
4,22

##### Funcionamiento.

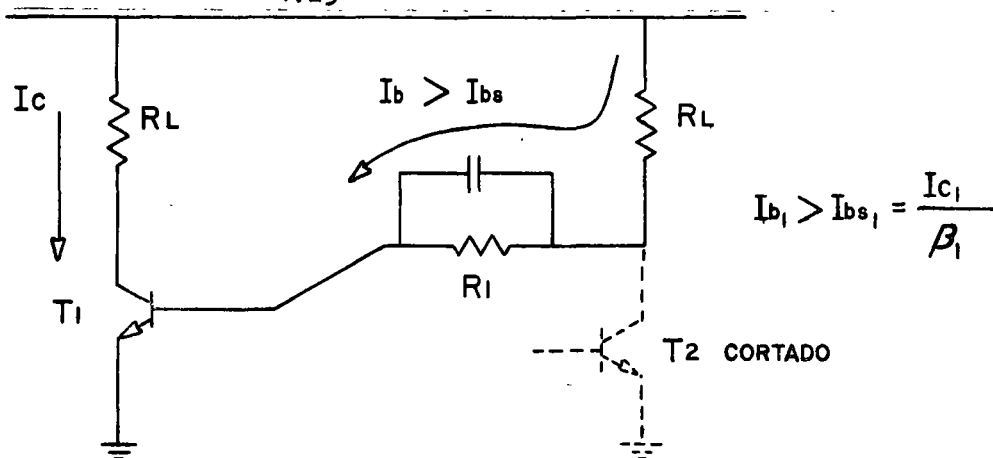
La idea fundamental en el funcionamiento del monoestable es utilizar a un transistor en la junta base emisor como comparador y tener el voltaje en la base por la carga de un capacitor. Los monoestables se encuentran en circuitos como son el NE555 y el 9602, o bien, ser implementados con

elementos discretos.

En la figura 4.23 se muestra un monoestable con elementos discretos. Los valores de las resistencias  $R_L$  y  $R_1$  deben de ser tales que cuando el transistor  $T_2$  corte la corriente a través de las resistencias  $R_L$  y  $R_1$  sea lo suficiente para saturar a  $T_1$ . El circuito en este caso para D.C. se muestra en la Figura 4.24.



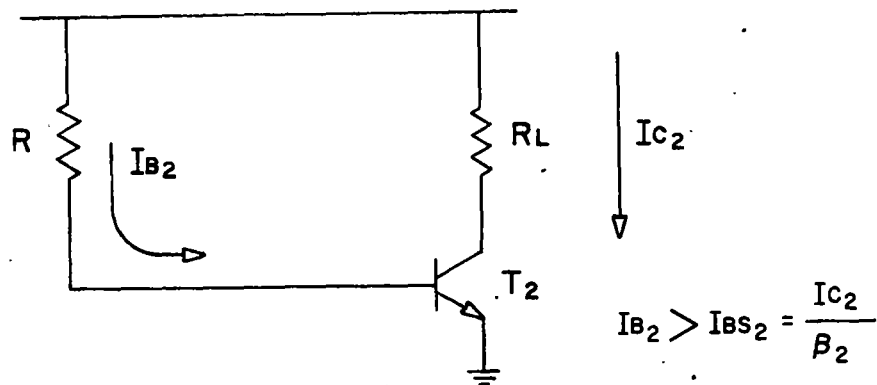
4.23



4.24

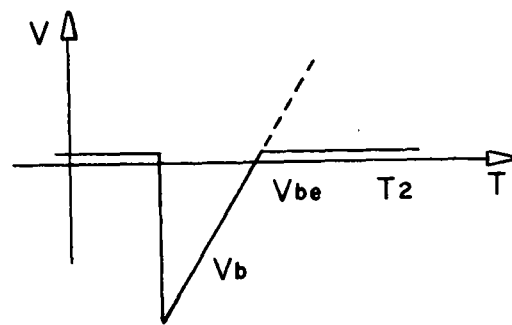
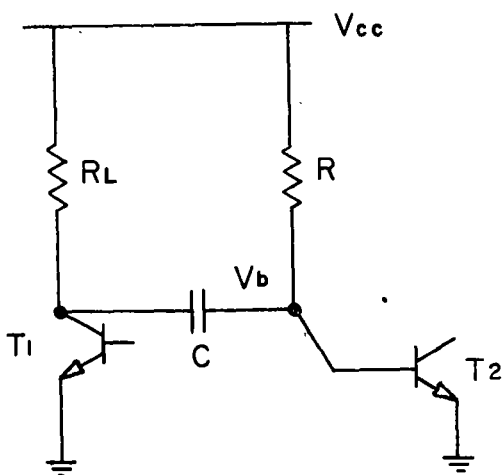
La resistencia  $R$  se calcula para mantener el estado estable, es decir el transistor  $T_2$  en saturación y el  $T_1$  en corte. El circuito equivalente

se muestra en la figura 4.25.



4.25

Teniéndose los parámetros calculados de la forma anterior, el funcionamiento del circuito es como sigue: Al ser conectado el circuito a  $V_{cc}$  y pasar un tiempo para que  $C$  se cargue, el transistor  $T_2$  satura y corta  $T_1$ , el capacitor  $C$  se encuentra cargado prácticamente a  $V_{cc}$  en el sentido que lo indica la figura 4.23, si se aplica un pulso negativo a través de  $C_2$  a la base de  $T_2$  para cortarlo, por diseño  $T_1$  satura, haciendo que el voltaje en el colector de  $T_1$  sea aproximadamente cero volts, debido a que el capacitor  $C$  se encontraba cargado en el sentido mostrado por la figura 4.23, en la base de  $T_2$  se tiene un voltaje negativo y este voltaje tenderá a irse a  $V_{cc}$  como lo muestra la figura 4.26.



$$V_b = -V_{cc} + 2V_{cc}(1 - e^{-t/\tau})$$

$$\tau = RC$$

4.26

Siempre que el voltaje  $V_b$  sea menor que el voltaje de conducción en la base del transistor  $T_2$ , éste se encontrará cortado, obteniéndose en la salida un voltaje alto; resolviendo la ecuación para  $V_b$ , se llega a que el tiempo alto de salida es  $t = .7RC$ .

#### 4.9. Astable.

##### Descripción.

El astable es un circuito oscilador de forma de onda cuadrada y cuya frecuencia de oscilación y ciclo de trabajo es función de parámetros enfermos.

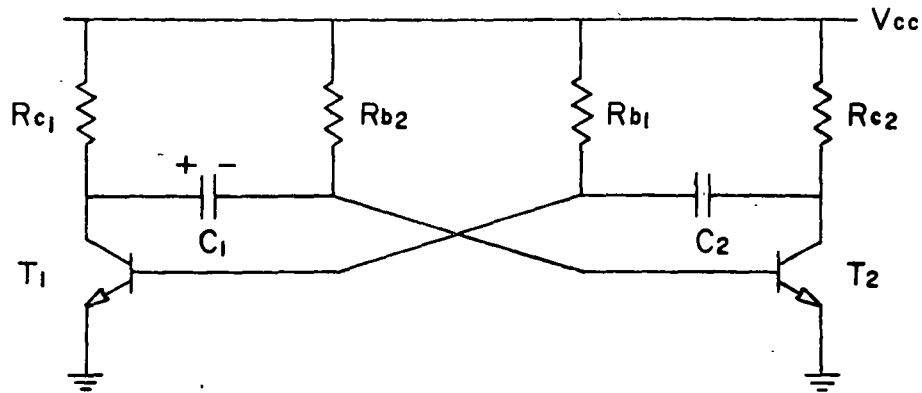
##### Funcionamiento.

El astable es un circuito oscilador de forma de onda cuadrada y cuya frecuencia de oscilación y ciclo de trabajo es función de parámetros enfermos.

##### Funcionamiento.

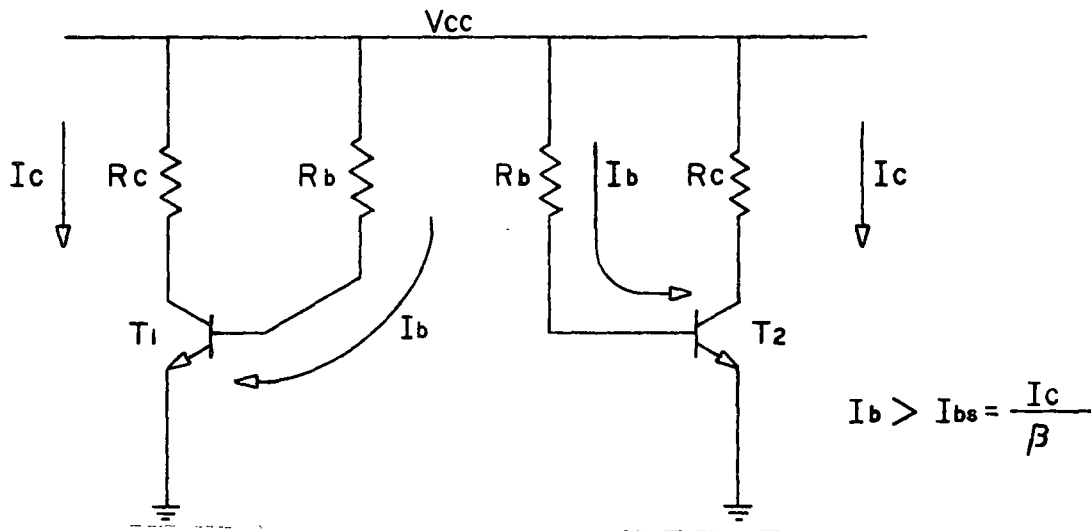
Al igual que el monoestable, este circuito puede ser encontrado en circuitos integrados y puede realizarse con elementos discretos. La figura 4.27 muestra al astable acoplado por colector con elementos discretos.





4.27

en la figura se observa que un análisis de D.C. nos lleva a dos arreglos, cada uno de ellos de un transistor y dos resistencias (figura 4.28).

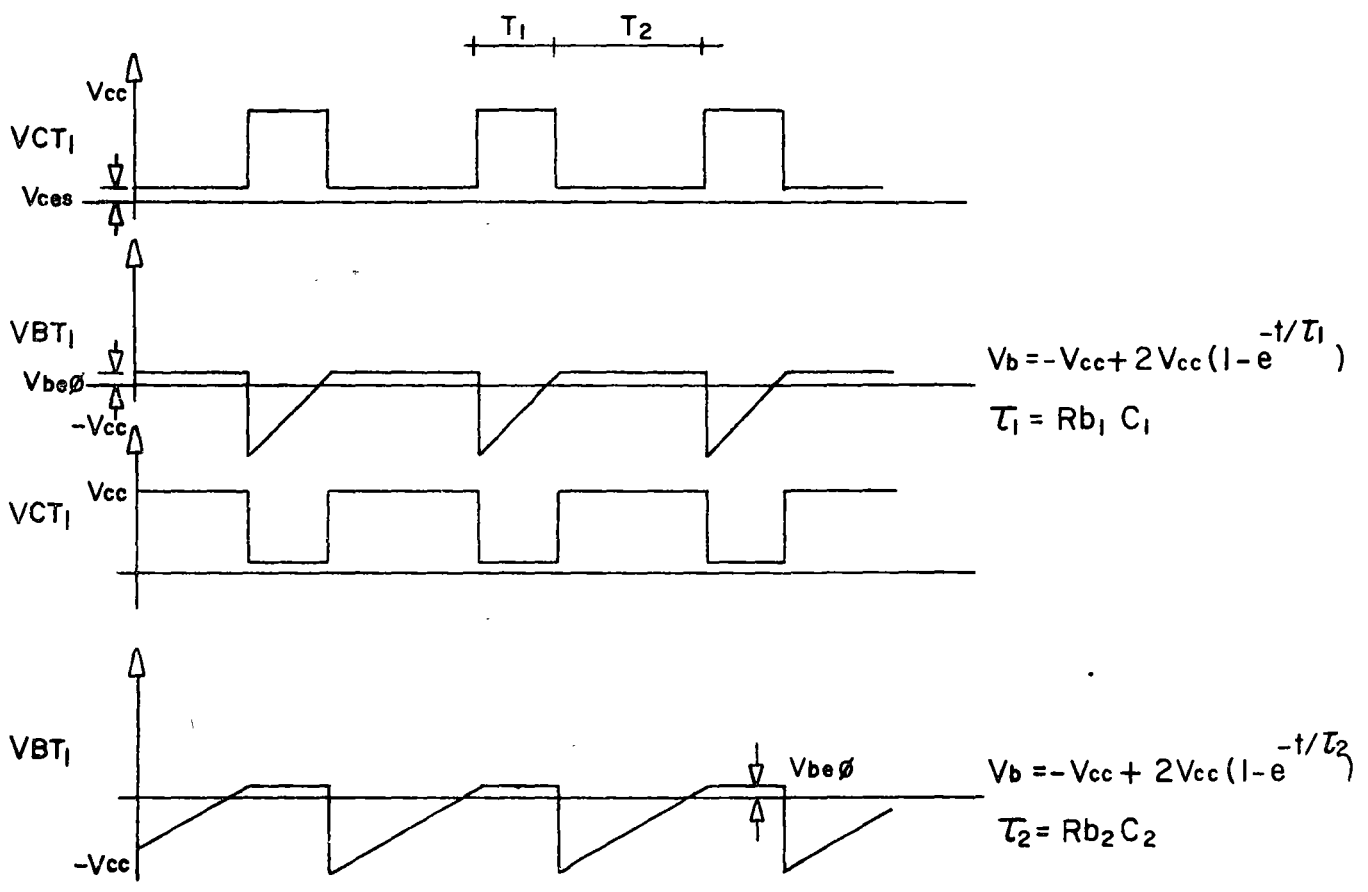


4.28

Por diseño se hace la resistencia  $R_b$  lo suficientemente pequeña para que los transistores, en D.C. se encuentren en saturación; el análisis de A.C. se hace con los capacitores de acoplamiento (figura 4.27) y es como sigue:

Al ser conectado a  $V_{cc}$  el circuito de la figura 4.28, por diseño ambos transistores tenderán a saturar; por diferencia en la fabricación de los transistores, alguno de ellos llegará primero a saturación, suponiendo que este sea  $T_2$  y al saber que al tiempo de ser conectado el circuito a  $V_{cc}$ , se tienen los capacitores descargados, la saturación de  $T_2$  producirá el corte de  $T_1$ , seguido de la carga del capacitor  $C_1$  en el sentido que lo muestra la figura 4.28 al mismo tiempo se tendrá que el capacitor  $C_2$  se tratará de cargar a  $V_{cc}$  por la resistencia  $R_{b1}$ , al momento de que el voltaje en el capacitor sea igual a  $0.7V$  el transistor  $T_1$  conduce y por diseño satura; la saturación de  $T_1$  produce el corte de  $T_2$  y el análisis es semejante en modo simétrico para el caso cuando  $T_2$  saturaba.

Las formas de onda en los colectores y bases de los transistores se muestran en la figura 4.29, resolviendo las ecuaciones para los voltajes en las bases de los transistores se calculan los tiempos en que los transistores se encuentran cortados.

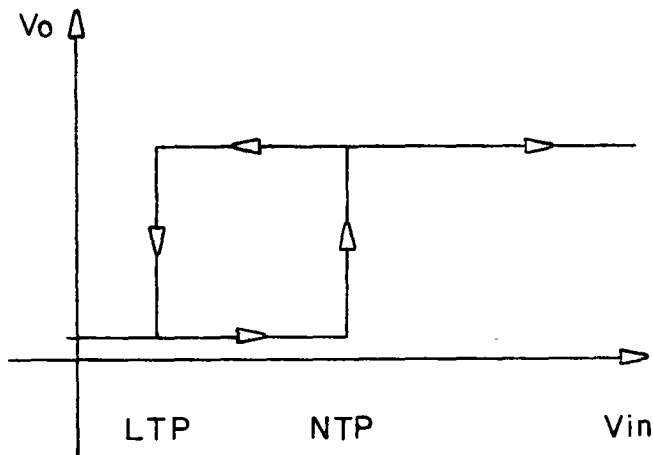


4.29

#### 4.10. Schmitt Trigger.

##### Descripción.

El schmitt trigger es un comparador con histerisis; los niveles de comparación están en función de parámetros de diseño, la curva de transferencia se muestra en la figura 4.30.

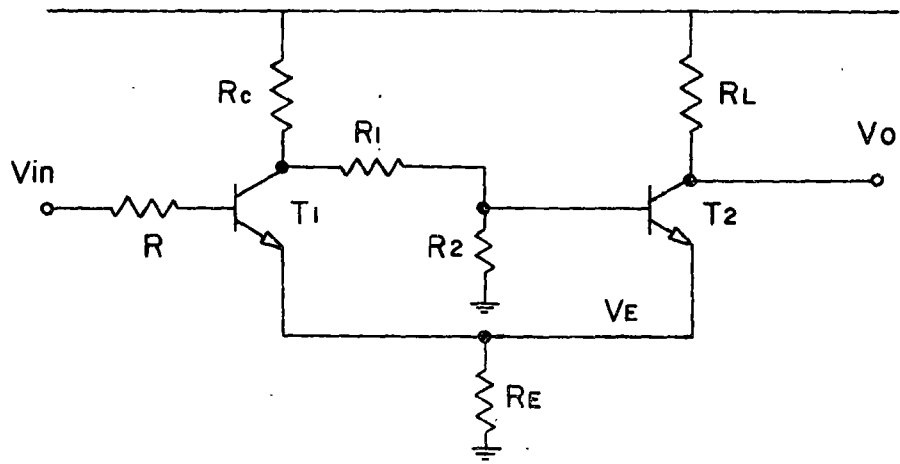


4.30

##### Funcionamiento.

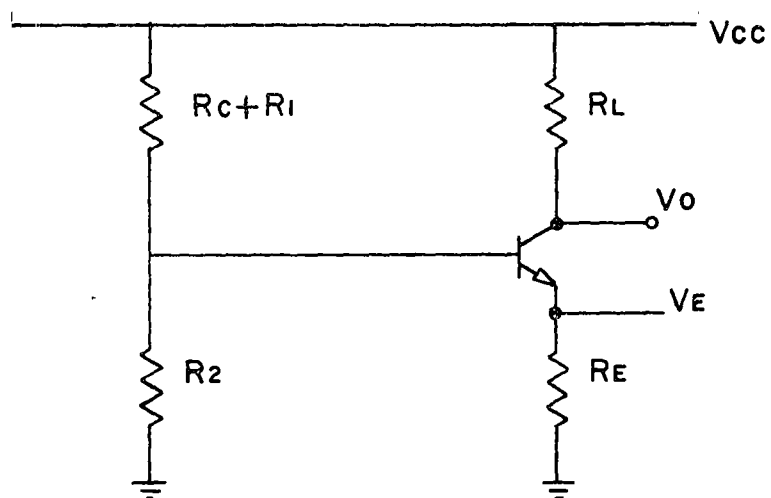
La figura 4.31 muestra el circuito del schmitt trigger trabajando los transistores  $T_1$  y  $T_2$  como amplificadores con realimentación positiva.

Por diseño se hace que cuando  $T_1$  corta, el transistor  $T_2$  sature mediante el divisor resistivo  $R_c$ ,  $R_1$ ,  $R_2$ . (Figura 4.31).



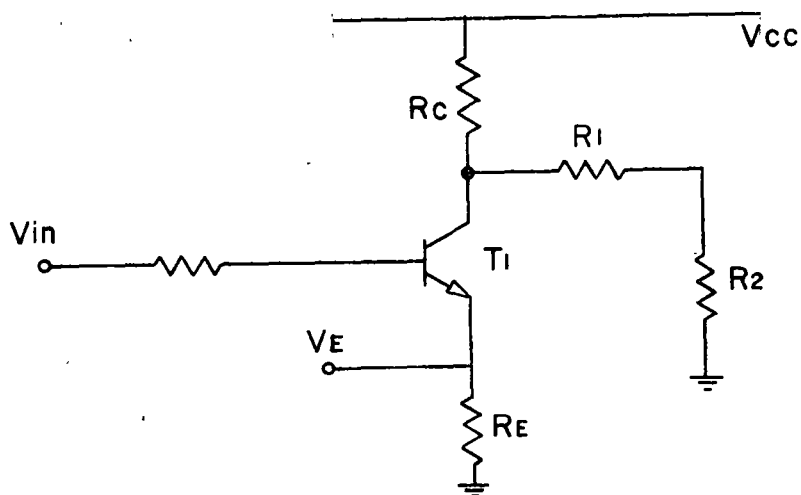
4.31

Si el voltaje de entrada ( $V_{in}$ ) es menor que el voltaje en los emisores  $V_E$ ,  $T_1$  se encontrará en corte y el transistor  $T_2$  por diseño en saturación (figura 4.32).



4.32

Si suponemos que ahora el voltaje de entrada ( $V_{in}$ ) se incrementa, para cuando  $V_{in}$  es igual a  $V_E$ , la junta base-emisor del transistor  $T_1$  empieza a ponerse en directo; para cuando:  $V_{in} = V_e' + V_{ce0}$  el transistor  $T_1$  conduce, haciendo que  $T_2$  salga de saturación y disminuya el voltaje  $V_E$  aumentando la diferencia entre  $V_{in}$  y  $V_e$ , haciendo que el transistor  $T_1$  conduzca más hasta cortar a  $T_2$ . Ahora las condiciones del circuito se muestran en la figura 4.33.



4.33

Por diseño en los valores de las componentes se hace que  $V_e' < V_E$  con las condiciones de la figura 4.31  $V_{in}$  deberá ser menor que  $V_e' + 0.7$  para cortar a  $T_1$  y saturar  $T_2$ . La salida es el colector de  $T_2$ .

#### 4.11. NE 555.

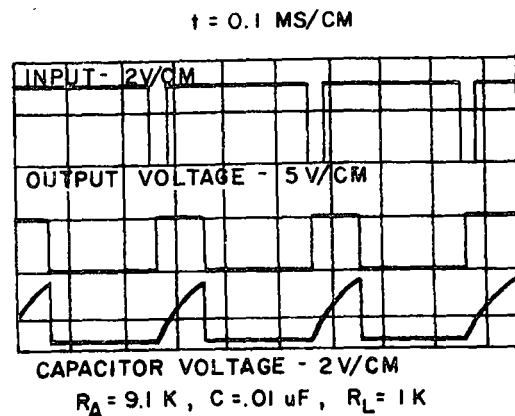
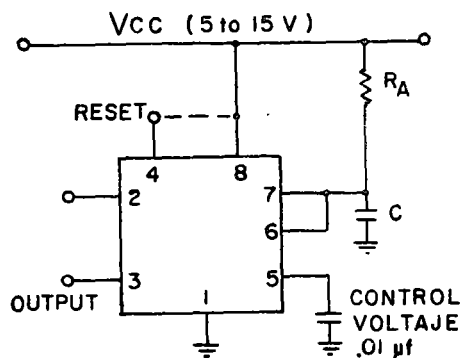
##### Operación como monoestable.

En este modo de operación el timer 555 da a su salida un pulso seguido

del frente de onda negativo de la entrada. Referido a la figura 4.34 el capacitor  $C$  es inicialmente descargado por un transistor interno del circuito. Al tenerse el frente negativo en la entrada (pata 2) se fija un biestable que corte al transistor para liberar el corto en el capacitor externo. El voltaje en el capacitor se incrementa exponencialmente con la constante de carga  $= R_A C$ . Cuando el voltaje en el capacitor sea  $2/3 V_{CC}$ , un comparador limpia al biestable dando un voltaje bajo a la salida, al tiempo que el capacitor es descargado.

El pulso de entrada es sentido al ser menor de  $1/3 V_{CC}$ . Una vez disparado, la salida es alta hasta que se alcanza el tiempo de carga del capacitor, este tiempo es  $t = 1.1 R_A C$  y es calculado de la figura 4.34.

El circuito tiene un "Reset" (Pata 4), que al tener un nivel bajo, fuerza la descarga del capacitor y deja al circuito en posibilidad de no volver a ser disparado.



### Ejemplos de tiempos en biestables.

Los biestables al estar implementados con compuertas, las señales de control y datos, sufren de retardos dentro del mismo. Cada biestable en particular tendrá sus propios retardos atendiendo a su arquitectura tecnología y fabricación.

Cada fabricante según la tecnología, arquitectura y fabricación, garantiza tiempos mínimos y máximos en el funcionamiento de los biestables. No existe una nomenclatura de acoplación general, pero la filosofía de las pruebas y la información proporcionada es suficiente y cambia poco de fabricante a fabricante. Algunos ejemplos se dan en los cuadros 2, 3, 4, 5 y 6.



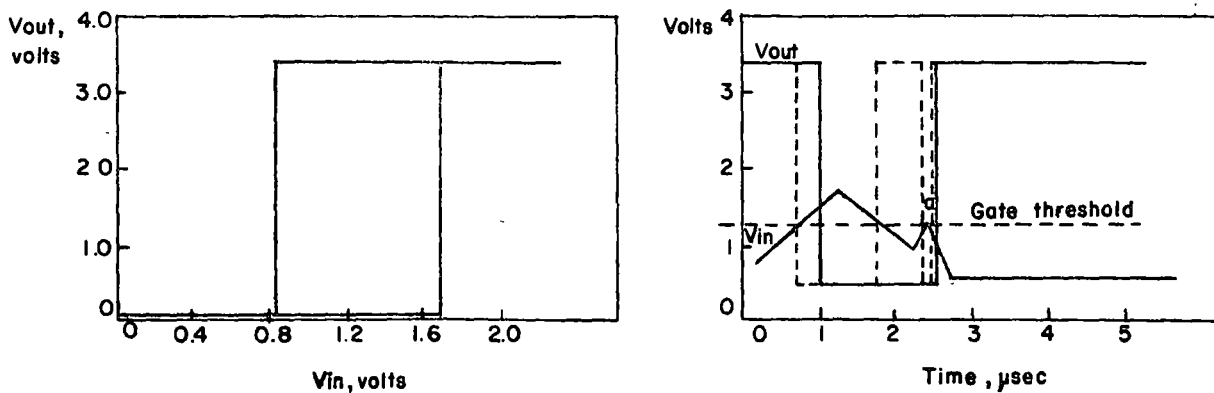
La figura 4.35 muestra las formas de onda en la salida y en el capacitor, de ellas se obtienen los tiempos para el cálculo de la frecuencia y el ciclo de trabajo.

$$f = \frac{1}{T} = \frac{1.44}{(R_A + 2R_B)C}$$

$$D = \frac{R_B}{R_A + 2R_B}$$

#### 4.12. SN 7413-14.

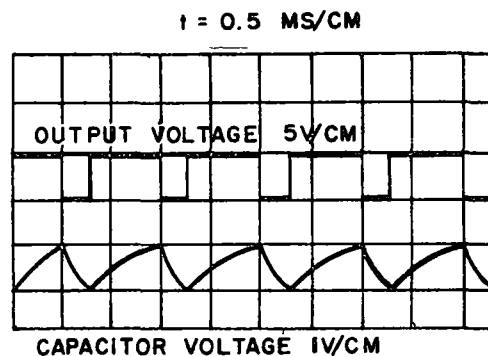
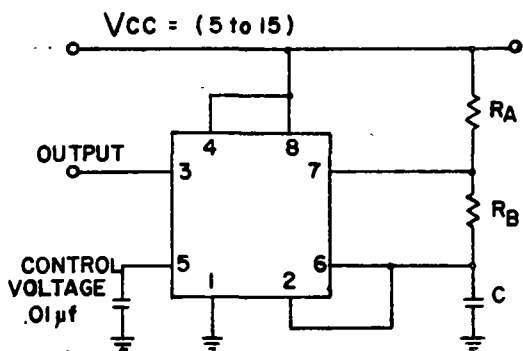
Debido a las necesidades de uso de los Schmitt trigger, para señales lentas o aplicaciones en donde se requiere un gran tiempo de subida y bajada, han sido diseñados los circuitos SN7413 y SN7414, ambos tienen entradas por comparadores con histerisis. Estos circuitos son completamente compatibles con TTL o DTL y requiere sólo una fuente de 5 volts. La figura 4.36 muestra las características de entrada/salida. El cuadro 1 muestra las características de estas compuertas.



### Operación como astable.

Si el NE 555 es conectado como se ilustra en la figura 4.35 (Pata 2 y 6 conectadas) será disparado por si mismo y se obtiene un "corrimiento libre" como astable. El capacitor externo se carga a través de  $R_A$  y  $R_B$ , y se descarga solamente sobre  $R_B$ . El ciclo de trabajo es determinado por la relación de  $R_A$  y  $R_B$ .

En este modo de operación, el capacitor se carga y descarga entre  $1/3$  y  $2/3$   $V_{CC}$ . Al igual que en el monoestable, los tiempos de carga y descarga son independientes de  $V_{CC}$ , haciendo que la frecuencia de oscilación también sea independiente de  $V_{CC}$ .

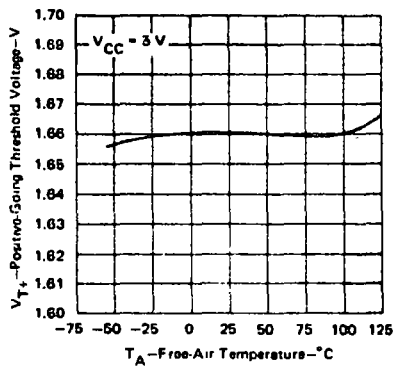


4.35

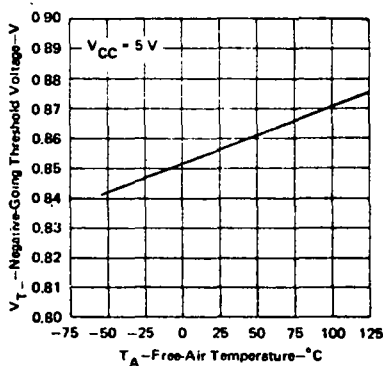
# CIRCUIT TYPES SN5413, SN7413 DUAL NAND SCHMITT TRIGGERS

## TYPICAL CHARACTERISTICS

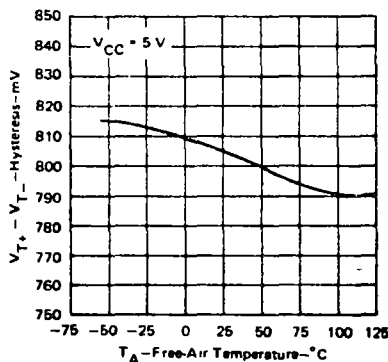
POSITIVE-GOING THRESHOLD VOLTAGE  
vs  
FREE-AIR TEMPERATURE



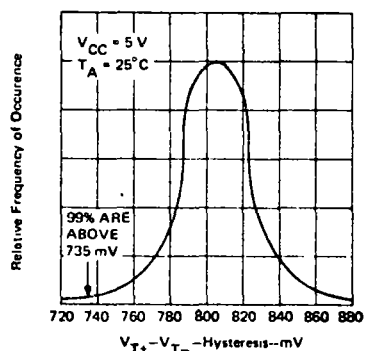
NEGATIVE-GOING THRESHOLD VOLTAGE  
vs  
FREE-AIR TEMPERATURE



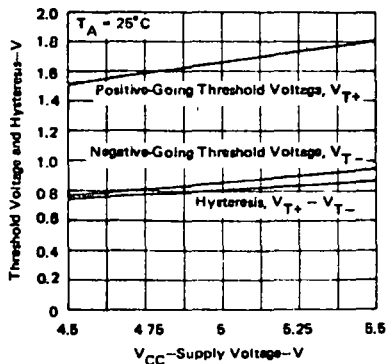
HYSTERESIS  
vs  
FREE-AIR TEMPERATURE



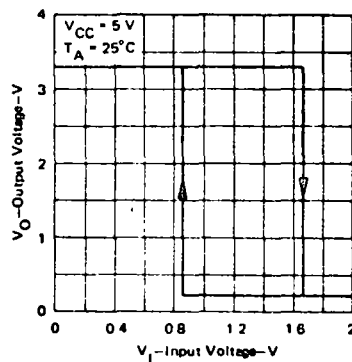
DISTRIBUTION OF UNITS  
FOR HYSTERESIS



THRESHOLD VOLTAGES AND HYSTERESIS  
vs  
SUPPLY VOLTAGE



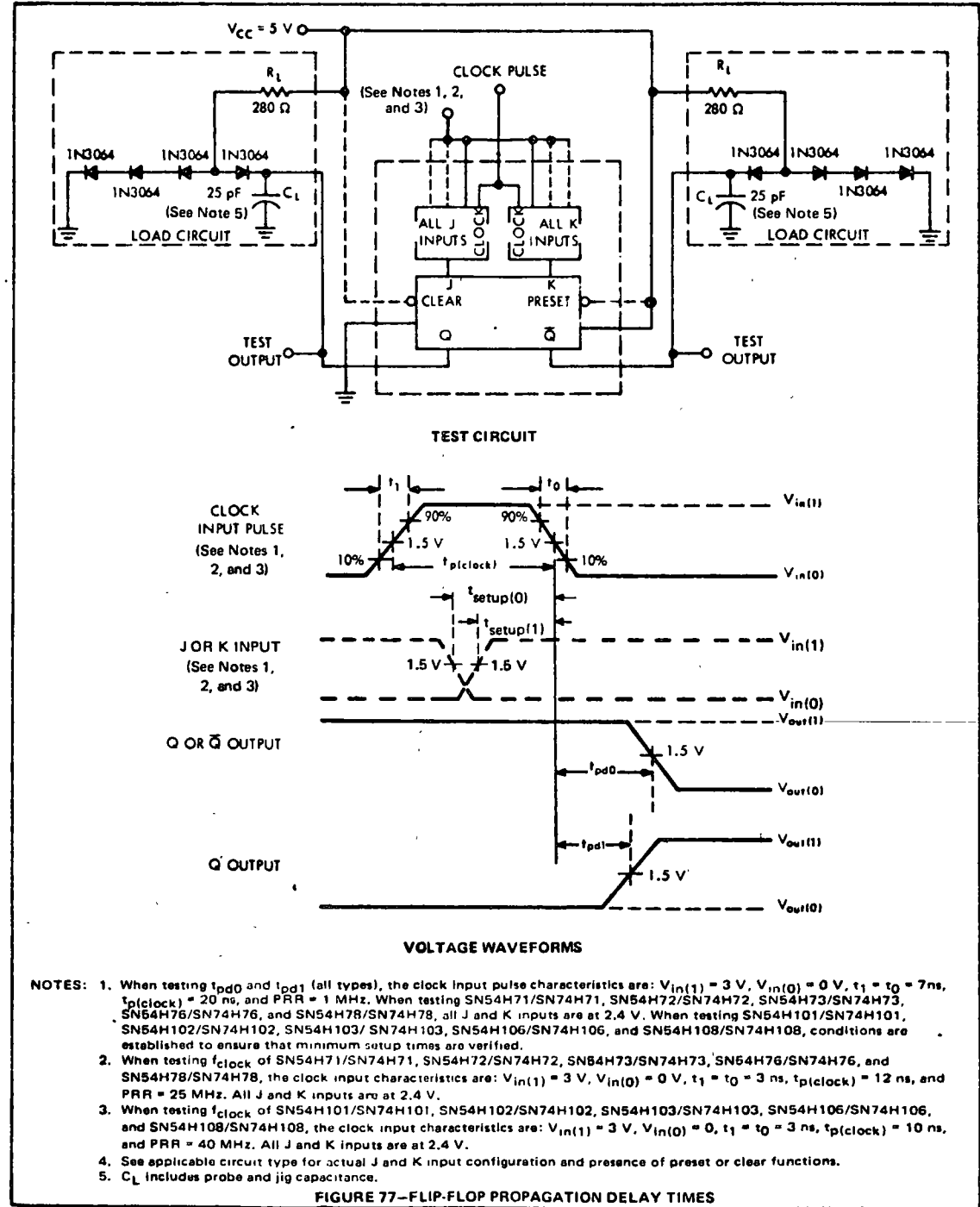
OUTPUT VOLTAGE  
vs  
INPUT VOLTAGE



## SERIES 54H, 74H HIGH-SPEED TRANSISTOR-TRANSISTOR LOGIC

### PARAMETER MEASUREMENT INFORMATION

switching characteristics (continued)

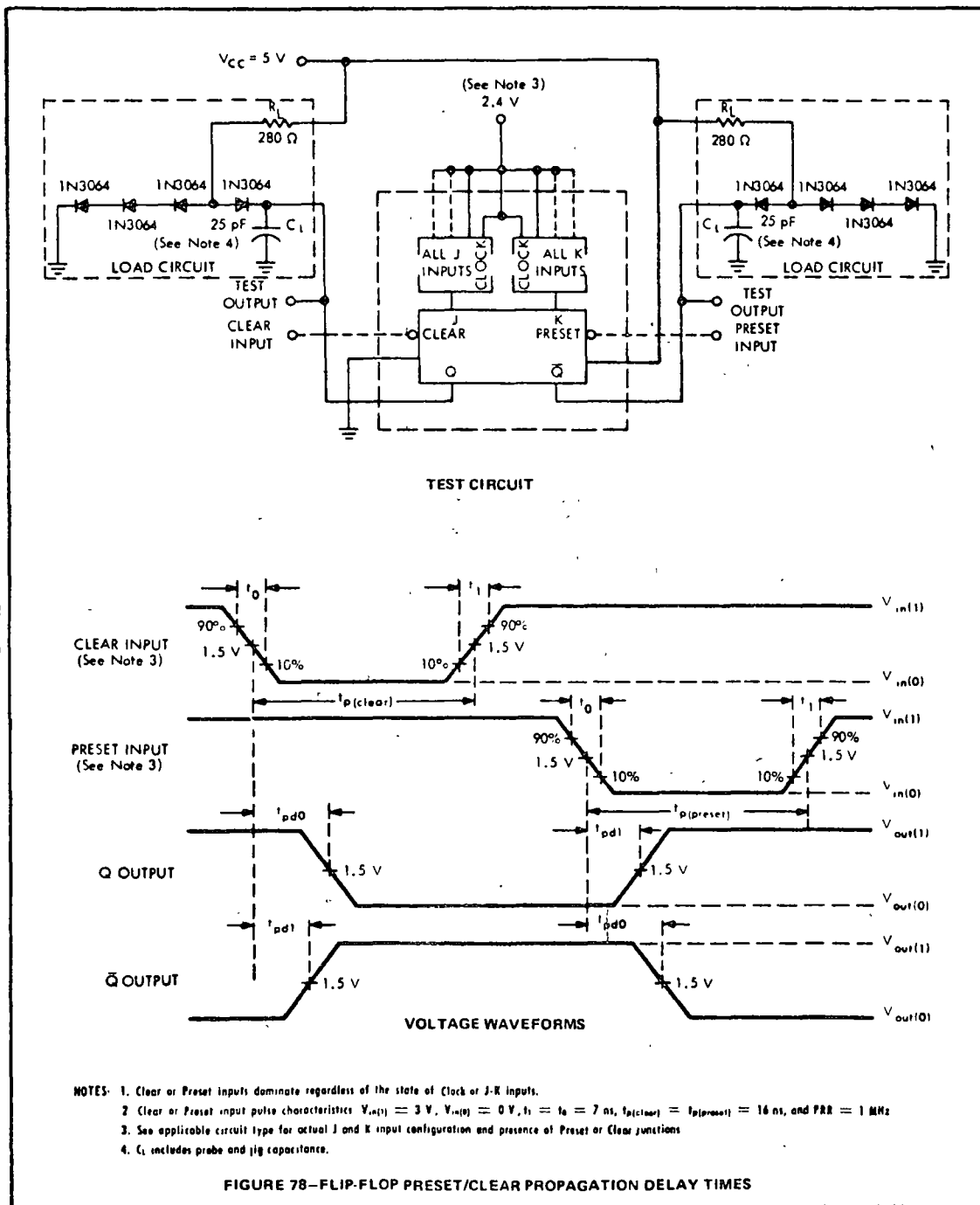


CUADRO 2.

**SERIES 54H, 74H**  
**HIGH-SPEED TRANSISTOR-TRANSISTOR LOGIC**

**PARAMETER MEASUREMENT INFORMATION**

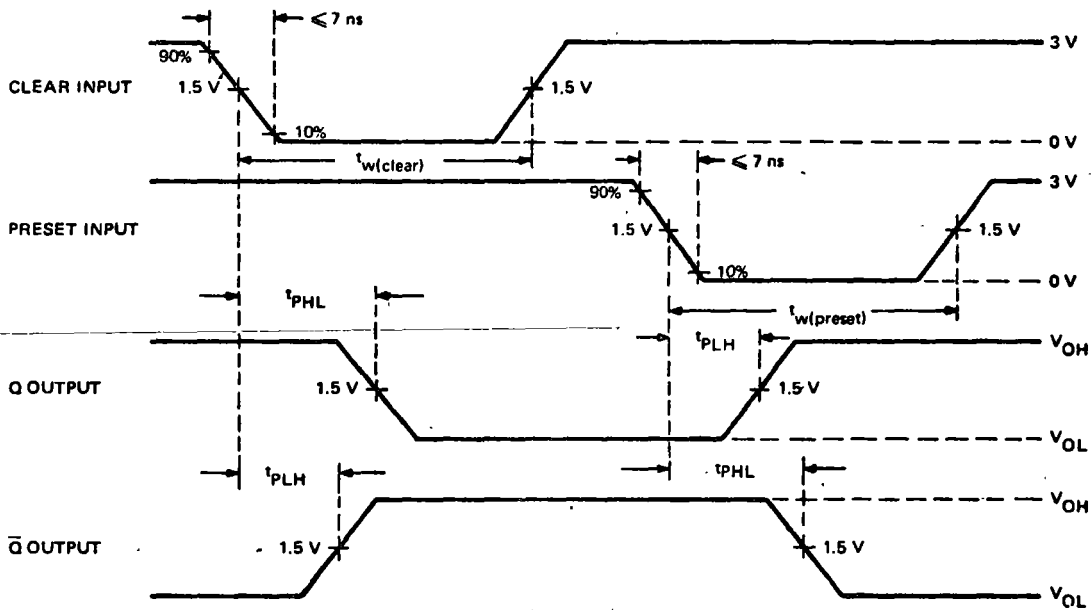
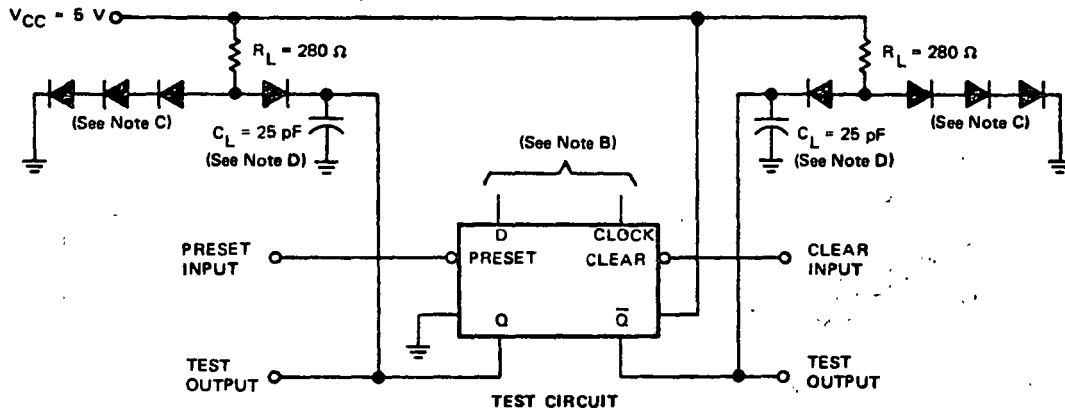
switching characteristics (continued)



# SERIES 54H, 74H HIGH-SPEED TRANSISTOR-TRANSISTOR LOGIC

## PARAMETER MEASUREMENT INFORMATION

### switching characteristics



VOLTAGE WAVEFORMS

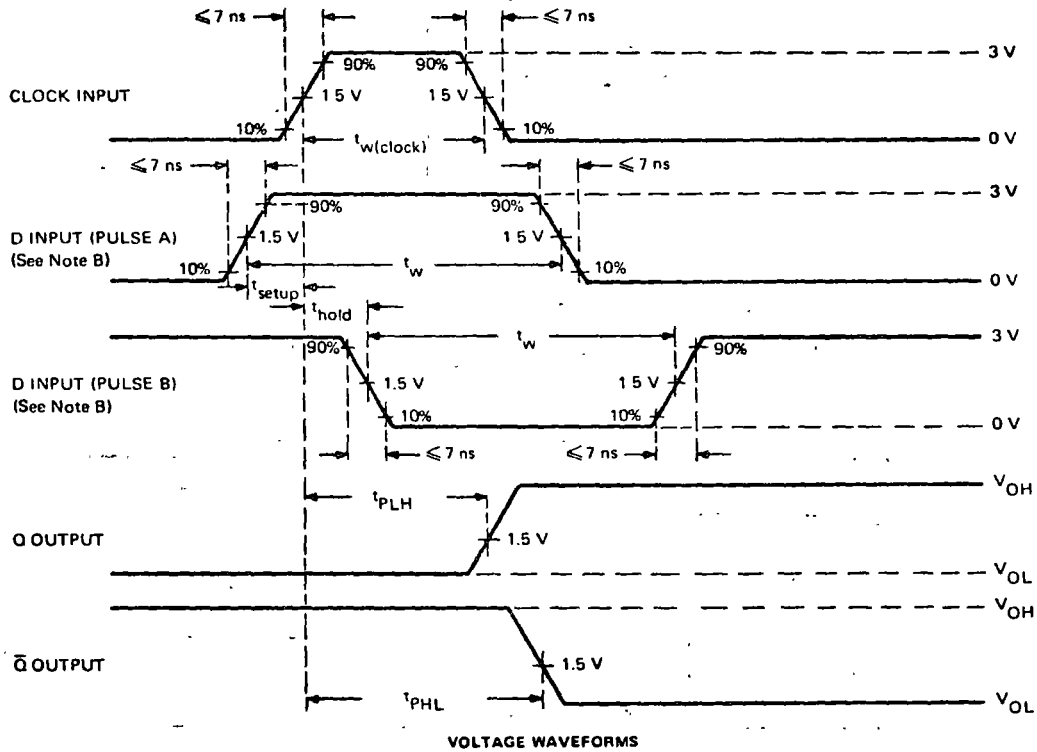
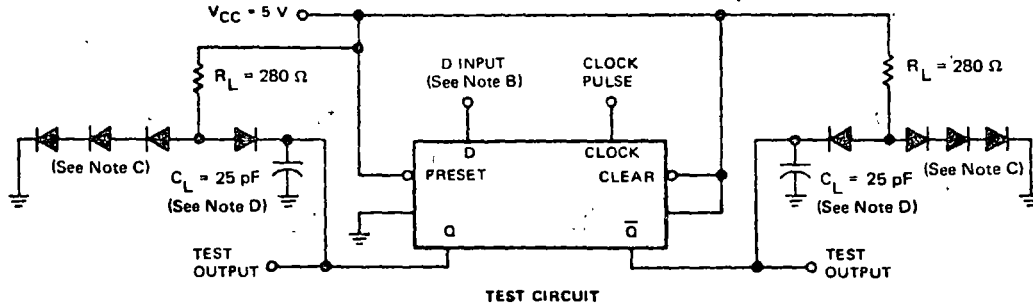
- NOTES. A. Clear or Preset Input pulse characteristics:  $t_w(\text{clear}) = t_w(\text{preset}) = 25 \text{ ns}$ , PRR = 1 MHz.  
 B. Clear and Preset inputs dominate regardless of the state of Clock or D inputs.  
 C. All diodes are 1N3064.  
 D. C<sub>L</sub> includes probe and jig capacitance.

FIGURE 86—ASYNCHRONOUS INPUTS SWITCHING CHARACTERISTICS

**SERIES 54H, 74H  
HIGH-SPEED TRANSISTOR-TRANSISTOR LOGIC**

**PARAMETER MEASUREMENT INFORMATION**

switching characteristics (continued)



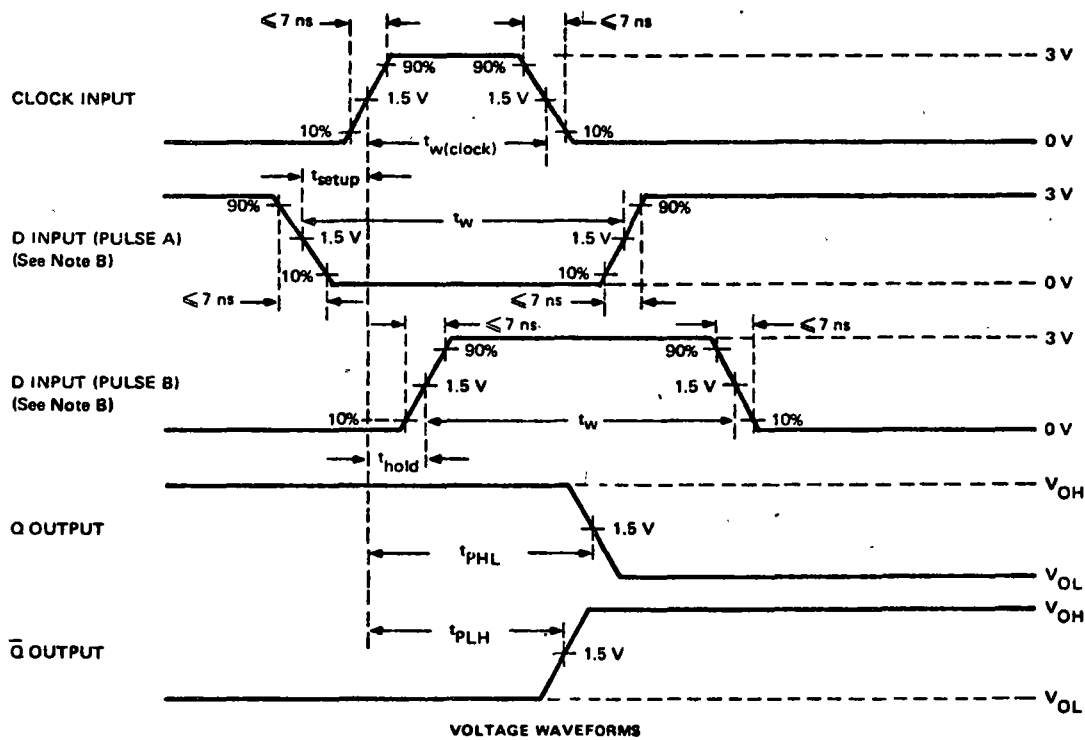
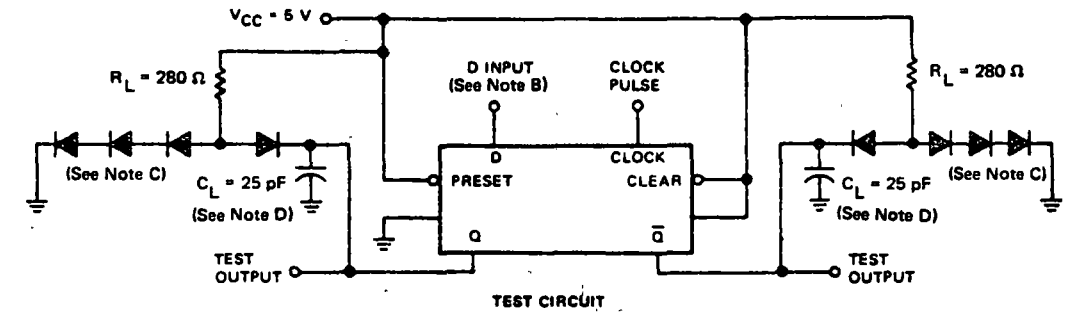
- NOTES:**
- A. Clock input pulse has the following characteristics:  $t_w(\text{clock}) = 20 \text{ ns}$  and  $\text{PRR} = 1 \text{ MHz}$ . When testing  $f_{\text{clock}}$ , vary PRR.
  - B. D input (pulse A) has the following characteristics.  $t_{\text{setup}} = 10 \text{ ns}$ ,  $t_w = 60 \text{ ns}$ , and PRR is 50% of the clock PRR. D input (pulse B) has the following characteristics.  $t_{\text{hold}} = 0 \text{ ns}$ ,  $t_w = 80 \text{ ns}$ , and PRR is 50% of the clock PRR.
  - C. All diodes are 1N3064
  - D.  $C_L$  includes probe and jig capacitance.

**FIGURE 87—SWITCHING CHARACTERISTICS, CLOCK AND SYNCHRONOUS INPUTS (HIGH-LEVEL DATA)**

# SERIES 54H, 74H HIGH-SPEED TRANSISTOR-TRANSISTOR LOGIC

## PARAMETER MEASUREMENT INFORMATION

switching characteristics (continued)



- NOTES:
- A. Clock input pulse has the following characteristics:  $t_w = 20 \text{ ns}$  and  $\text{PRR} = 1 \text{ MHz}$ . When testing  $f_{\text{clock}}$ , vary PRR.
  - B. D input (pulse A) has the following characteristics:  $t_{\text{setup}} = 15 \text{ ns}$ ,  $t_w = 60 \text{ ns}$ , and PRR is 50% of the clock PRR. D input (pulse B) has the following characteristics:  $t_{\text{hold}} = 0 \text{ ns}$ ,  $t_w = 60 \text{ ns}$ , and PRR is 50% of the clock PRR.
  - C. All diodes are 1N3064.
  - D.  $C_L$  includes probe and jig capacitance.

FIGURE 88—SWITCHING CHARACTERISTICS, CLOCK AND SYNCHRONOUS INPUTS (LOW-LEVEL DATA)





centro de educación continua  
división de estudios superiores  
facultad de ingeniería, unam



APLICACIONES DE CIRCUITOS INTEGRADOS AL  
DISEÑO DIGITAL

CAPITULO 5: CIRCUITOS SECUENCIALES SINCRONOS

CAPITULO 6: CIRCUITOS SECUENCIALES ASINCRONOS

ING. MARIO RODRIGUEZ MANZANERA

JUNIO 1978

## CAPITULO 5: DISEÑO DE CIRCUITOS SECUENCIALES.

5.1 Introducción: Una de las partes de mayor importancia en el diseño lógico es la correspondiente a la realización de circuitos secuenciales.

5.1.1. Definición: Cuando en un circuito el valor presente en sus salidas (z), no es dependiente únicamente del valor presente en sus entradas (X) sino que lo es también de la historia del circuito (Q), este será denominado 'secuencial'.

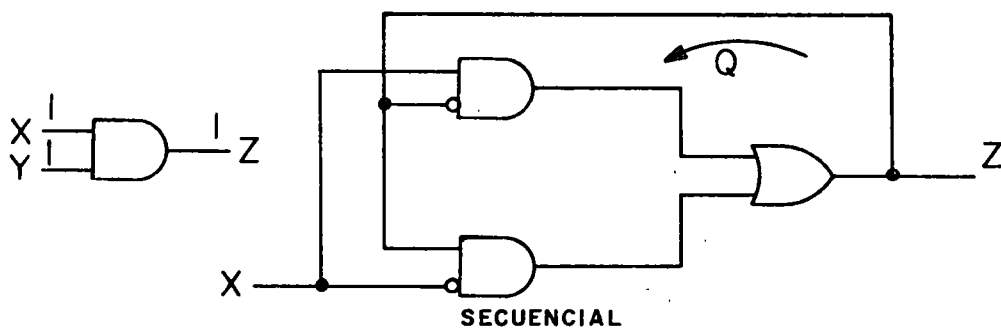


Figura 5.1

5.1.2 Estado: Es una sencilla representación de la historia de un circuito, la cual puede considerarse como una secuencia de estados. Un estado es una propiedad inherente al circuito.

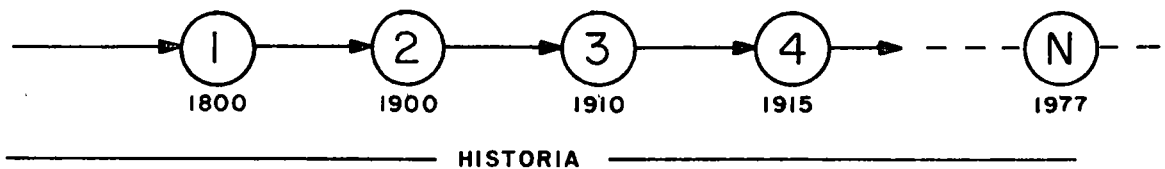


Figura 5.1.1.

### 5.1.3. Circuitos secuenciales síncronos y asíncronos.

Un circuito secuencial es síncrono cuando sus cambios de estado pueden llevarse a cabo únicamente en coincidencia con una señal de reloj es decir en sincronía con éste.

Un circuito secuencial asíncrono es aquel que no requiere de una señal de reloj fija para cambiar de estado.

## 5.2. Circuitos secuenciales síncronos.

### 5.2.1. Representación de circuitos secuenciales.

Para plantear un problema el diseño lógico se auxilia de la teoría de autómatas y de la teoría de gráficas con el fin de facilitar su solución.

#### 5.2.1.1. Representación gráfica o diagrama de flujo.

En la representación gráfica se utilizarán los elementos de teoría de gráficas siendo estos: círculos denominados nodos, flechas denominadas ramas o trayectorias.

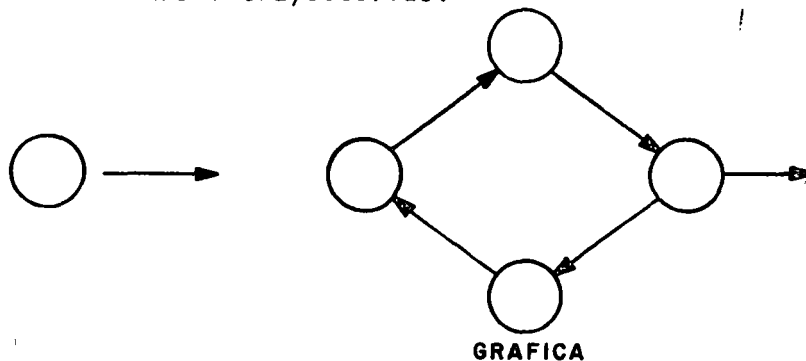


Figura 5.3.

Cada nodo representará un estado y será etiquetado con el nombre de este, por ejemplo los estados A, B, C



Cada rama será etiquetada con el nombre de la entrada al circuito que afecta a un cierto estado sobre la salida del circuito al realizarse el paso de un nodo a otro, es decir:

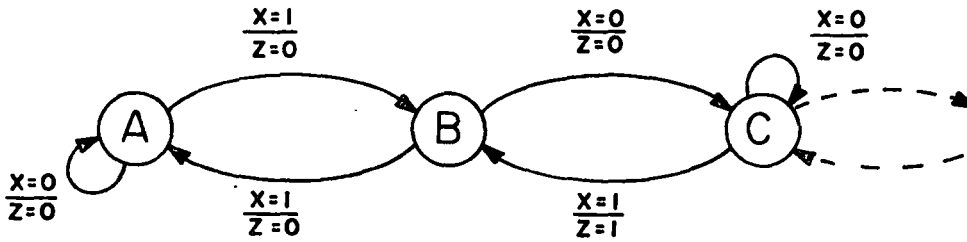


Figura 5.4.

En la figura 5.4 se tienen todos los elementos para definir un autómata esto es, se tiene:

Un conjunto de estados  $\{A, B, C, \dots\} = \{Q\} = \{q_1, q_2, \dots\}$

Un conjunto de entradas  $\{X_1, \dots\} = \{X\} = \{x_1, x_2, \dots\}$

Un conjunto de salidas  $\{Z_1, \dots\} = \{Z\} = \{z_1, z_2, \dots\}$

y dos funciones, una llamada  $\delta$  y otra  $\lambda$ .

$\delta$  es la función de transición y se define como:

$$\delta(q, x) = q^{t=1}$$

y  $\lambda$  la función de salida:

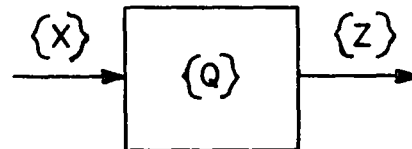
$$\lambda(q, x) = Z$$

Así en la figura 1.4.

$$Q = A, B, C$$

$$X = X$$

$$Z = Z$$



$$\begin{array}{ll}
 \delta(A, x=0) = A & \lambda(A, x=0) = 0 \\
 \delta(A, x=1) = B & \lambda(A, x=1) = 0 \\
 \delta(B, x=1) = A & \lambda(B, x=0) = 0 \\
 \delta(B, x=1) = C & \lambda(B, x=0) = 0 \\
 \delta(C, x=0) = C & \lambda(C, x=0) = 0 \\
 \delta(C, x=1) = B & \lambda(C, x=1) = 1
 \end{array}$$

Entonces el siguiente estado de B al tener una entrada  $X=0$  será el estado C y el circuito produce una salida  $Z=0$ .

El empleo de esta representación facilitará grandemente el planteamiento de cualquier problema más no así su solución. Con este fin, se introducen las tablas de transición.

5.2.1.1. Tablas de Transición. Otra forma de representación de un autómata son las tablas de transición en donde cada uno de los elementos anteriormente introducidos se incluyen.

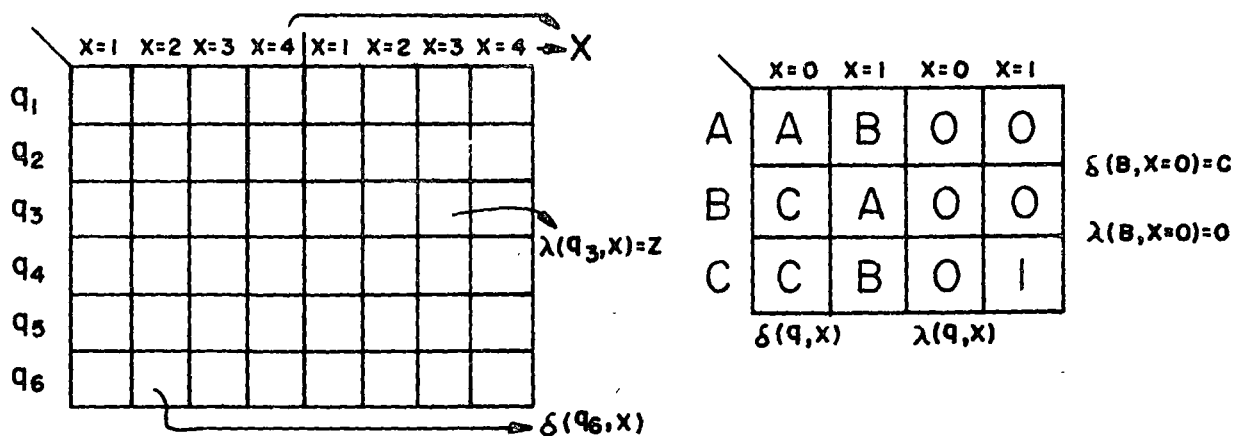


Figura 5.5.

A través del uso de estas representaciones el diseñador puede en forma sencilla plantear los problemas de diseño en circuitos secuenciales.

5.3. Ejemplos de Diseño: A continuación se incluyen diversos diseños para circuitos necesarios en múltiples aplicaciones.

5.3.1. Contadores. El número de estados de un contador está limitado por la base de este; así, para un contador base 5 serán necesarios 5 estados. En la Figura 5.6 se presenta el diagrama de flujo para este contador y su respectiva tabla de transición.

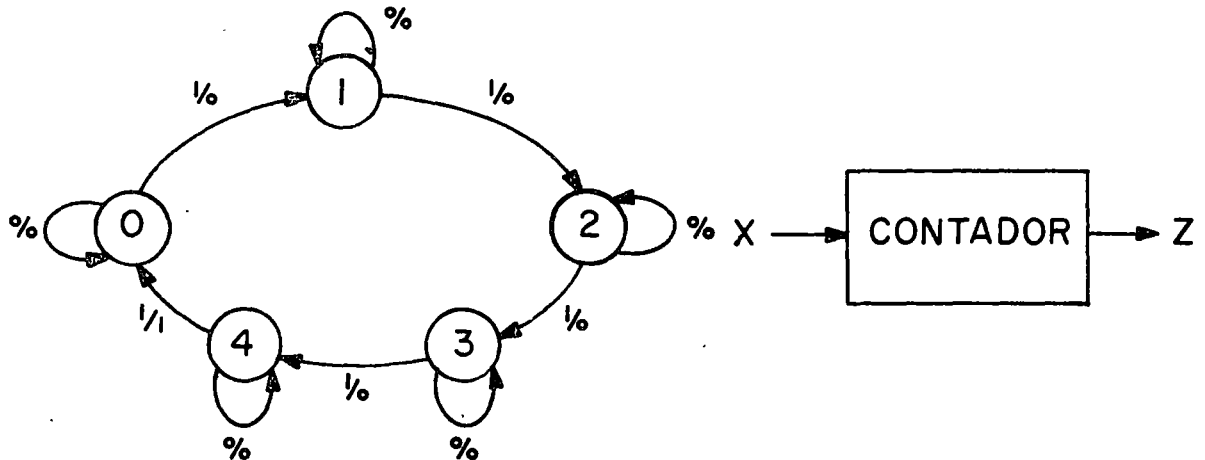


Figura 5.6.

ENUNCIADO: Realizar un circuito el cual reciba pulsos por una entrada X y produzca una salida Z cada 5 pulsos.

| Q         | X=0 | X=1 | X=0 | X=1 |
|-----------|-----|-----|-----|-----|
| $q_0 = 0$ | 0   | 1   | 0   | 0   |
| $q_1 = 1$ | 1   | 2   | 0   | 0   |
| $q_2 = 2$ | 2   | 3   | 0   | 0   |
| $q_3 = 3$ | 3   | 4   | 0   | 0   |
| $q_4 = 4$ | 4   | 0   | 0   | 1   |

δ
λ

Así en la figura 5.4.

$Q = A, B, C$

$X = X$

$Z = Z$

5.3.1.2. Asignación de estados: El problema de la asignación de estados corresponde a etiquetar cada estado con un número en Binario.

| ABC | X=0 | X=1 | X=0 | X=1 |
|-----|-----|-----|-----|-----|
| 000 | 000 | 001 | 0   | 0   |
| 001 | 001 | 010 | 0   | 0   |
| 010 | 010 | 011 | 0   | 0   |
| 011 | 011 | 100 | 0   | 0   |
| 100 | 100 | 000 | 0   | 1   |

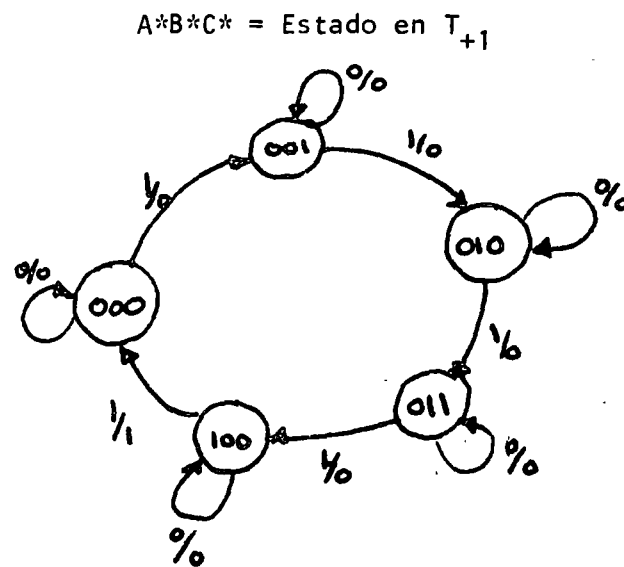


Figura 5.7

Habiendo asignado un valor a cada estado en la tabla de transición se ha dado un nombre a cada columna teniéndose C, B, A.

5.3.1.3. Tabla de verdad. A partir de la tabla de transiciones será posible generar la tabla de verdad de las variables A, B y C.

Procedimiento:

Coloque las entradas como columnas (X) a continuación las variables (ABC) y sus combinaciones como se muestra en la Figura 5.8; coloque una línea vertical y vierta las funciones  $\delta$  y  $\lambda$ .

| XABC              | A*B*C* | Z |
|-------------------|--------|---|
| 0 000             | 000    | 0 |
| 0 001             | 001    | 0 |
| 0 010             | 010    | 0 |
| 0 011             | 011    | 0 |
| 0 100             | 100    | 0 |
| 0 101             | ***    | * |
| No definido 0 110 | ***    | * |
| 0 111             | ***    | * |
| 1 000             | 001    | 0 |
| 1 001             | 010    | 0 |
| 1 010             | 011    | 0 |
| 1 011             | 100    | 0 |
| 1 100             | 000    | 1 |

Figura 5.8.

5.3.1.4. Reducción: Una vez planteada la tabla de verdad el problema se reduce a obtener las funciones estrella o asterisco  $A^*$ ,  $B^*$  y  $C^*$ , para lo cual utilizamos los mapas de Karnaugh, (Figura 5.9).

|    |    | BC |    |    |    |
|----|----|----|----|----|----|
|    |    | 00 | 01 | 11 | 10 |
| XA | 00 | 0  | 0  | 0  | 0  |
|    | 01 | 1  | *  | *  | *  |
|    | 11 | 0  | *  | *  | *  |
|    | 10 | 0  | 0  | 1  | 0  |

$$A^* = \bar{X}A + XBC = A(\bar{X}) + XBC$$

Figura 5.9.



|    |    |    |    |    |
|----|----|----|----|----|
|    | BC |    |    |    |
| XA | 00 | 01 | 11 | 10 |
| 00 | 0  | 0  | 1  | 1  |
| 01 | 0  | *  | *  | *  |
| 11 | 0  | *  | *  | *  |
| 10 | 0  | 1  | 0  | 1  |

$$B^* = X \bar{B} C + \bar{X} B + B\bar{C} = B(\bar{X} + \bar{C}) + \bar{B}(XC)$$

|    |    |    |    |    |
|----|----|----|----|----|
|    | BC |    |    |    |
| XA | 00 | 01 | 11 | 10 |
| 00 | 0  | 1  | 1  | 0  |
| 01 | 0  | *  | *  | *  |
| 11 | 0  | *  | *  | *  |
| 10 | 1  | 0  | 0  | 1  |

$$C^* = \bar{X}C + X\bar{A}C = C\bar{X} + \bar{C}X\bar{A}$$

5.3.1.5. Funciones. Una vez encontradas las funciones estrella tenemos:

$$A^* = A(\bar{X}) + XBC$$

$$B^* = B(X+C) + B(XC)$$

$$C^* = C(\bar{X}) + \bar{C}(X\bar{A})$$

y podemos observar que en las funciones  $B^*$  y  $C^*$  se tiene el valor real y el negado de la variable sin estrella; para lograr que  $A^*$  tenga la misma forma utilizamos el siguiente artificio:

$$A^* = A(\bar{X}) + XBC (A + \bar{A})$$

De donde:

$$A^* = A(X+XBC) + \bar{A} (XBC)$$

Y así:

$$A^* = A(\bar{X} + BC) + \bar{A} (XBC)$$

Quedando la función con su real y se negado resumiendo:

$$A^* = A(\bar{X}+BC) + \bar{A} (XBC)$$

$$B^* = B(\bar{X}+\bar{C}) + \bar{B} (XC)$$

$$C^* = C(\bar{X}) + \bar{C} (XA)$$

5.3.1.6. Implementación. Al observar la forma de las funciones encontradas recordamos la clásica representación de un Flip-Flop que en función de Q nos queda:

$$Q^* = g_1 Q + g_2 \bar{Q}$$

y para el Flip-Flop JK:

$$Q^* = J\bar{Q} + \bar{K}Q$$

De donde para la variable A\* se tiene:

$$Q^* = \bar{Q} J + Q\bar{K}$$

$$A^* = \bar{A} (XBC) + A (\bar{X}+BC)$$

De donde:

$$J_a = XBC \quad ; \quad K_a = \overline{\bar{X}+BC}$$

y para B y C:

$$J_b = XC \quad ; \quad K_b = \overline{\bar{X}+\bar{C}}$$

$$J_c = X\bar{A} ; K_c = \bar{X}$$

Obteniéndose las ecuaciones de diseño:

$$J_a = XBC \quad K_a = X(\bar{B} + \bar{C})$$

$$J_b = XC \quad K_b = XC \quad \therefore J_b = K_b = XC$$

$$J_c = X\bar{A} \quad K_c = X$$

De aquí es posible realizar el diagrama lógico:

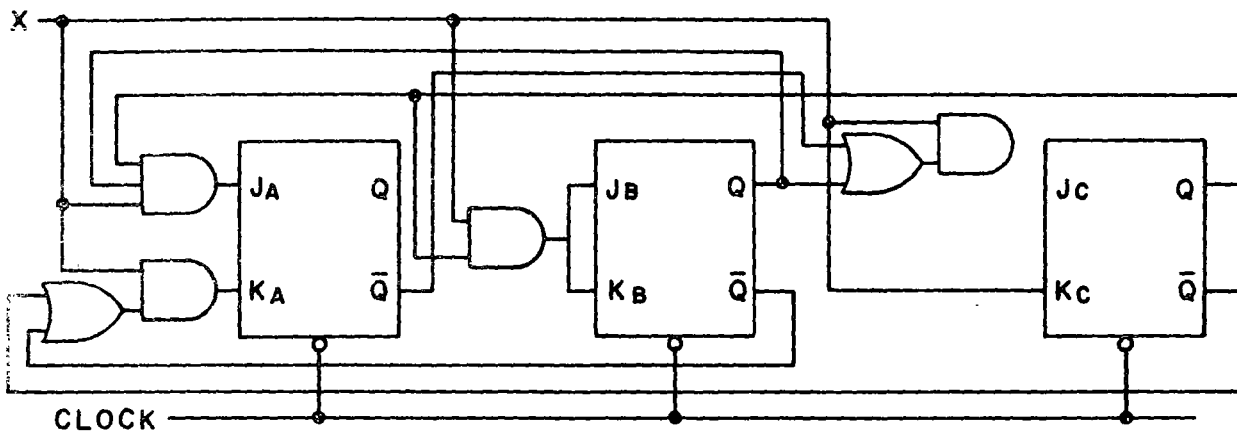


Figura 5.10

5.3.1.7. Función de Salida. Resta por el momento obtener la función de salida Z, utilizando el mismo procedimiento tenemos:

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

$$Z = XA$$

esto es:

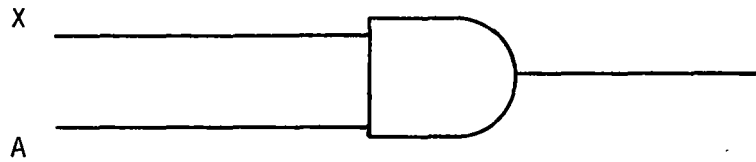


Figura 5.11.

Rearreglando el circuito y agragando la salida tenemos:

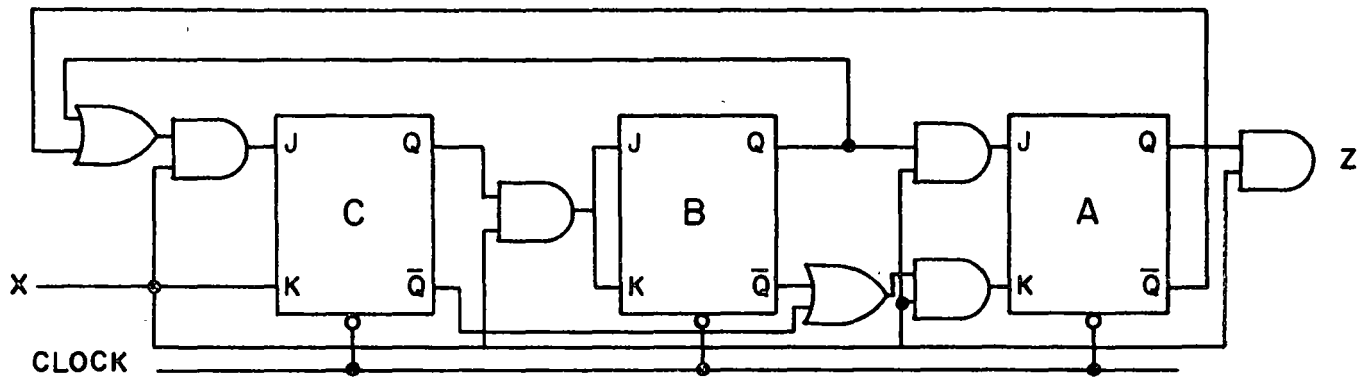


Figura 5.12

5.3.1.8. Posibles Reducciones. Cuando en el diseño no interesa que debe suceder al circuito cuando la entrada  $X$  es cero, es posible realizar una reducción en el diseño anterior.

es decir,

| ABC | $X=1$ | $X=1$ |
|-----|-------|-------|
| 000 | 001   | 0     |
| 001 | 010   | 0     |
| 010 | 011   | 0     |
| 011 | 100   | 0     |
| 100 | 000   | 1     |

Figura 5.13

de donde:

| X ABC | A*B*C* |
|-------|--------|
| 0 000 | X X X  |
| 0 001 | X X X  |
| 0 010 | X X X  |
| 0 011 | X X X  |
| 0 100 | X X X  |
| 0 101 | X X X  |
| 0 110 | X X X  |
| 0 111 | X X X  |
| 1 000 | 0 0 1  |
| 1 001 | 0 1 0  |
| 1 010 | 0 1 1  |
| 1 011 | 1 0 0  |
| 1 100 | 0 0 0  |
| 1 101 | X X X  |
| 1 110 | X X X  |
| 1 111 | X X X  |

BC

XA

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

A\* = BCX

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

$$B = (\overline{BC} + BC)X$$

BC

XA

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

$$C = \overline{C}AX$$

En la misma forma:

$$\begin{aligned}
 A^* &= BCX(A+\bar{A}) = A(BCX) + \bar{A}(BCX) \quad \therefore J_a = BCX \quad K_a = \overline{BCX} = \bar{B} + \bar{C} \\
 B^* &= \text{---} B(C\bar{X}) + (\bar{B}CX) \quad \therefore J_b = CX \quad K_b = \overline{CX} = \bar{C} + \bar{X} \\
 C^* &= \bar{C}(\bar{A}X) = 0 + C(\bar{A}X) \quad \therefore J_c = \bar{A}X \quad K_c = 0 = 1
 \end{aligned}$$

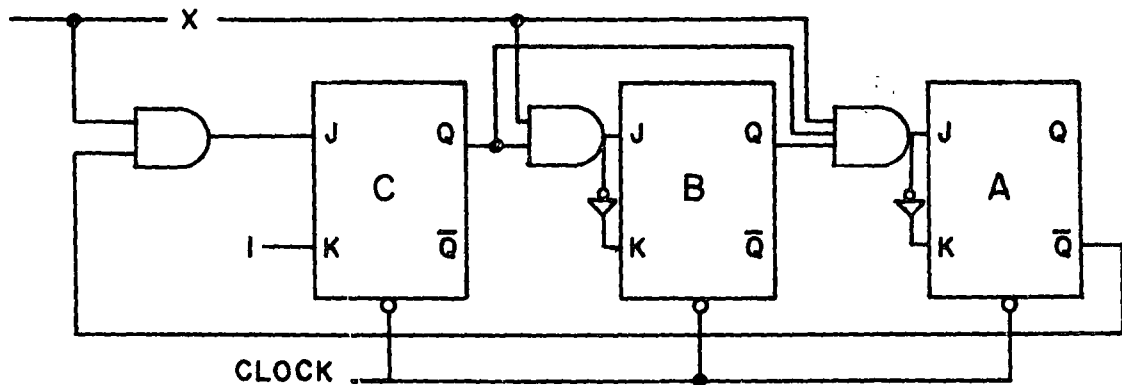


Figura 5.14

Si se desea tener un contador dependiente únicamente de la señal de reloj con  $X=1$  siempre tenemos:

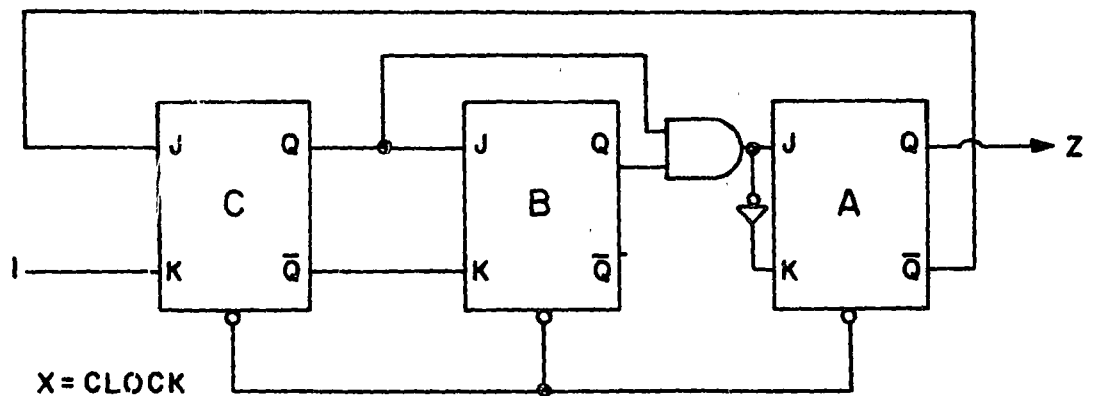
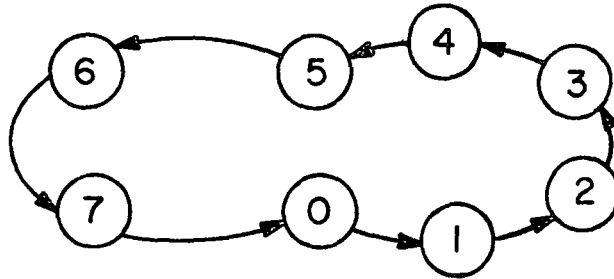


Figura 5.15

5.3.2: Ejemplo. Diseñar un contador base 8 utilizando FFT, JK.

Diagrama:



| X ABC | A* B* C* | Z |
|-------|----------|---|
| 0 000 | 0 0 0    | 0 |
| 0 001 | 0 0 1    | 0 |
| 0 010 | 0 1 0    | 0 |
| 0 011 | 0 1 1    | 0 |
| 0 100 | 1 0 0    | 0 |
| 0 101 | 1 0 1    | 0 |
| 0 110 | 1 1 0    | 0 |
| 0 111 | 1 1 1    | 0 |
| 1 000 | 0 0 1    | 0 |
| 1 001 | 0 1 0    | 0 |
| 1 010 | 0 1 1    | 0 |
| 1 011 | 1 0 0    | 0 |
| 1 100 | 1 0 1    | 0 |
| 1 101 | 1 1 0    | 0 |
| 1 110 | 1 1 1    | 0 |
| 1 111 | 0 0 0    | 1 |

XA

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

$$A^* = \bar{X}A + A\bar{B} + A\bar{C} + X\bar{A}BC$$

$$A^* = A(\bar{X} + \bar{B} + \bar{C}) + \bar{A}(XBC)$$

XA

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

$$B^* = B\bar{C} + B\bar{X} + \bar{B}CX$$

$$B^* = B(\bar{C} + \bar{X}) + \bar{B}(CX)$$

Figura 5.16.

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

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

$Z = XABC$

$C^* = XC + XC$

De las ecuaciones:  $Q^* = J\bar{Q} + \bar{K}Q$

|             |  |                             |
|-------------|--|-----------------------------|
| $J_a = XBC$ | $K_a = \overline{(\bar{x} + \bar{B} + \bar{C})} = XBC$ | $J_a = K_a \rightarrow T_a$ |
| $J_b = XC$  | $K_b = \overline{(\bar{C} + \bar{X})} = XC$            | $J_b = K_b \rightarrow T_b$ |
| $J_c = X$   | $K_c = \bar{X} = X$                                    | $J_c = K_c \rightarrow T_c$ |

Solucion:

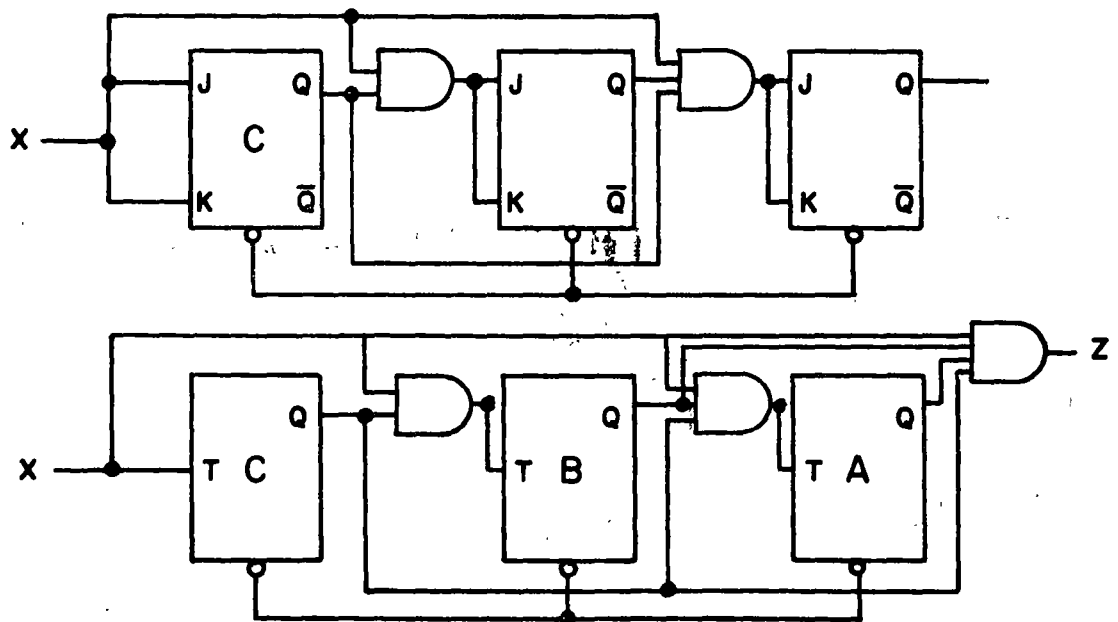


Figura 5.17.



Si ahora sólo tomamos los estados iniciales como no importantes tenemos:

$$A^* = A(\bar{B} + \bar{C}) + \bar{A}BC \quad \therefore J_a = K_a = T_a = B_c$$

$$B^* = BC + \bar{B}\bar{C} \quad \therefore J_b = K_b = T_b = C$$

$$C^* = C\bar{x} + \bar{C}x \quad \therefore J_c = K_c = T_c = x$$

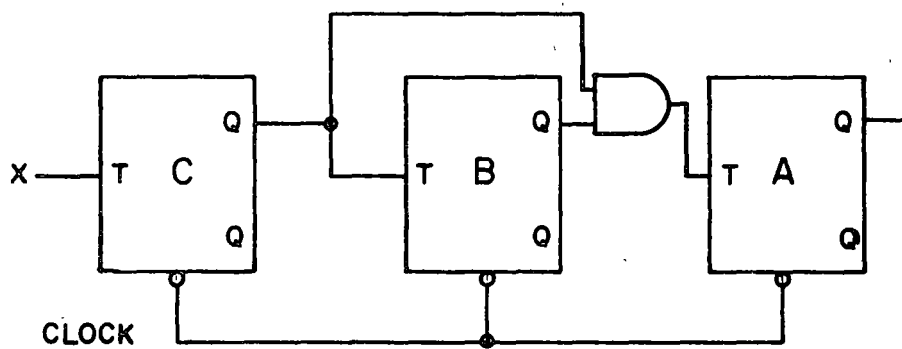


Figura 5.18.

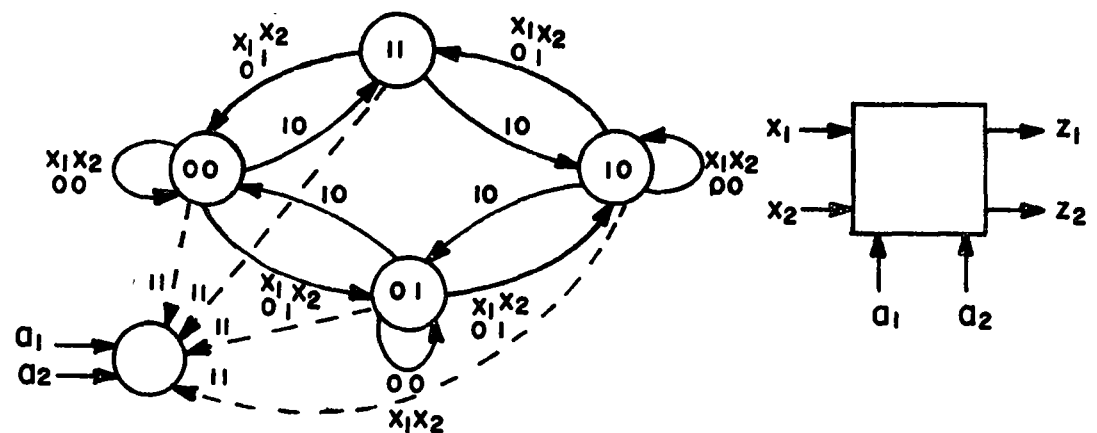
#### 5.4. Contadores ascendentes descendentes con cualquier secuencia y programables.

5.4.1. Contadores ascendentes y descendentes. Hasta ahora el diseño ha tomado únicamente en consideración contadores que incrementan su secuencia es decir:  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 0$

Para construir un contador de secuencia descendente bastará con tomar de un diseño ya construido las salidas  $\bar{Q}$ . En donde se encuentra el complemento del diseño es decir:  $15 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 11 \rightarrow 10$  respectivamente.

5.4.2. En la práctica muchas veces es indispensable fijar el número desde el cual se requiere efectuar una cuenta ascendente o descendente. Para esto se utiliza un contador programable en donde bajo ciertas señales de control, se decide cuando se está programando, cuando sube o cuando baja la cuenta.

5.4.3. Ejemplo: Diseño de un contador programable base 4 que cuente hacia arriba y hacia abajo.



|    | $x_1x_2$ | $x_1x_2$ | $x_1x_2$ | $x_1x_2$ |
|----|----------|----------|----------|----------|
| AB | 0 0      | 0 1      | 1 0      | 1 1      |
| 00 | 0 0      | 0 1      | 1 1      | $A_1A_2$ |
| 01 | 0 1      | 1 0      | 0 0      | $A_1A_2$ |
| 10 | 1 0      | 1 1      | 0 1      | $A_1A_2$ |
| 11 | 1 1      | 0 0      | 1 0      | $A_1A_2$ |

Programa

Cuenta abajo

Cuenta arriba

Permanece

Figura 5.19

| $X_1$ | $X_2$ | A | B | $A*B*$    |
|-------|-------|---|---|-----------|
| 0     | 0     | 0 | 0 | 0 0       |
| 0     | 0     | 0 | 1 | 0 1       |
| 0     | 0     | 1 | 0 | 1 0       |
| 0     | 0     | 1 | 1 | 1 1       |
| 0     | 1     | 0 | 0 | 0 1       |
| 0     | 1     | 0 | 1 | 1 0       |
| 0     | 1     | 1 | 0 | 1 1       |
| 0     | 1     | 1 | 1 | 0 0       |
| 1     | 0     | 0 | 0 | 1 1       |
| 1     | 0     | 0 | 1 | 0 0       |
| 1     | 0     | 1 | 0 | 0 1       |
| 1     | 0     | 1 | 1 | 1 0       |
| 1     | 1     | 0 | 0 | $A_1 A_2$ |
| 1     | 1     | 0 | 1 | $A_1 A_2$ |
| 1     | 1     | 1 | 0 | $A_1 A_2$ |
| 1     | 1     | 1 | 1 | $A_1 A_2$ |

|           |       | AB    |       |       |    |
|-----------|-------|-------|-------|-------|----|
| $X_1 X_2$ |       | 00    | 01    | 11    | 10 |
|           | 00    | 0     | 0     | 1     | 1  |
| 01        | 0     | 1     | 0     | 1     |    |
| 11        | $A_1$ | $A_1$ | $A_1$ | $A_1$ |    |
| 10        | 1     | 0     | 1     | 0     |    |

$$A^* = A(\bar{X}_1 \bar{B} + \bar{X}_2 B) + \bar{A}(\bar{X}_1 X_2 B + X_1 \bar{X}_2 \bar{B}) + X_1 X_2 A_1$$

|           |       | AB    |       |       |    |
|-----------|-------|-------|-------|-------|----|
| $X_1 X_2$ |       | 00    | 01    | 11    | 10 |
|           | 00    | 0     | 1     | 1     | 0  |
| 01        | 1     | 0     | 0     | 1     |    |
| 11        | $A_2$ | $A_2$ | $A_2$ | $A_2$ |    |
| 10        | 1     | 0     | 0     | 1     |    |

$$B^* = B(\bar{X}_1 \bar{X}_2) + \bar{B}(\bar{X}_1 X_2 + X_1 \bar{X}_2) + X_1 X_2 A_2$$

$$A^* = A(\bar{X}_1 \bar{B} + \bar{X}_2 B + X_1 X_2 A_1) + \bar{A}(\bar{X}_1 X_2 B + X_1 \bar{X}_2 \bar{B} + X_1 X_2 A_1)$$

$$B^* = B(X_1 X_2 + X_2 A_2) + \bar{B}(X_1 X_2 + X_1 \bar{X}_2 + X_1 X_2 A_2)$$

Figura 5.20

Así:

$$\bar{J}_a = \bar{X}_1 X_2 B + X_1 \bar{X}_2 \bar{B} + X_1 X_2 A_1$$

$$K_a = \bar{X}_1 \bar{B} + \bar{X}_2 B + X_1 X_2 A_1$$

$$J_b = \bar{X}_1 X_2 + X_1 \bar{X}_2 + X_1 X_2 + X_1 X_2 A_2 = X_1 + X_2 X_1 X_2 A_2$$

$$K_b = \bar{X}_1 \bar{X}_2 + X_1 X A$$

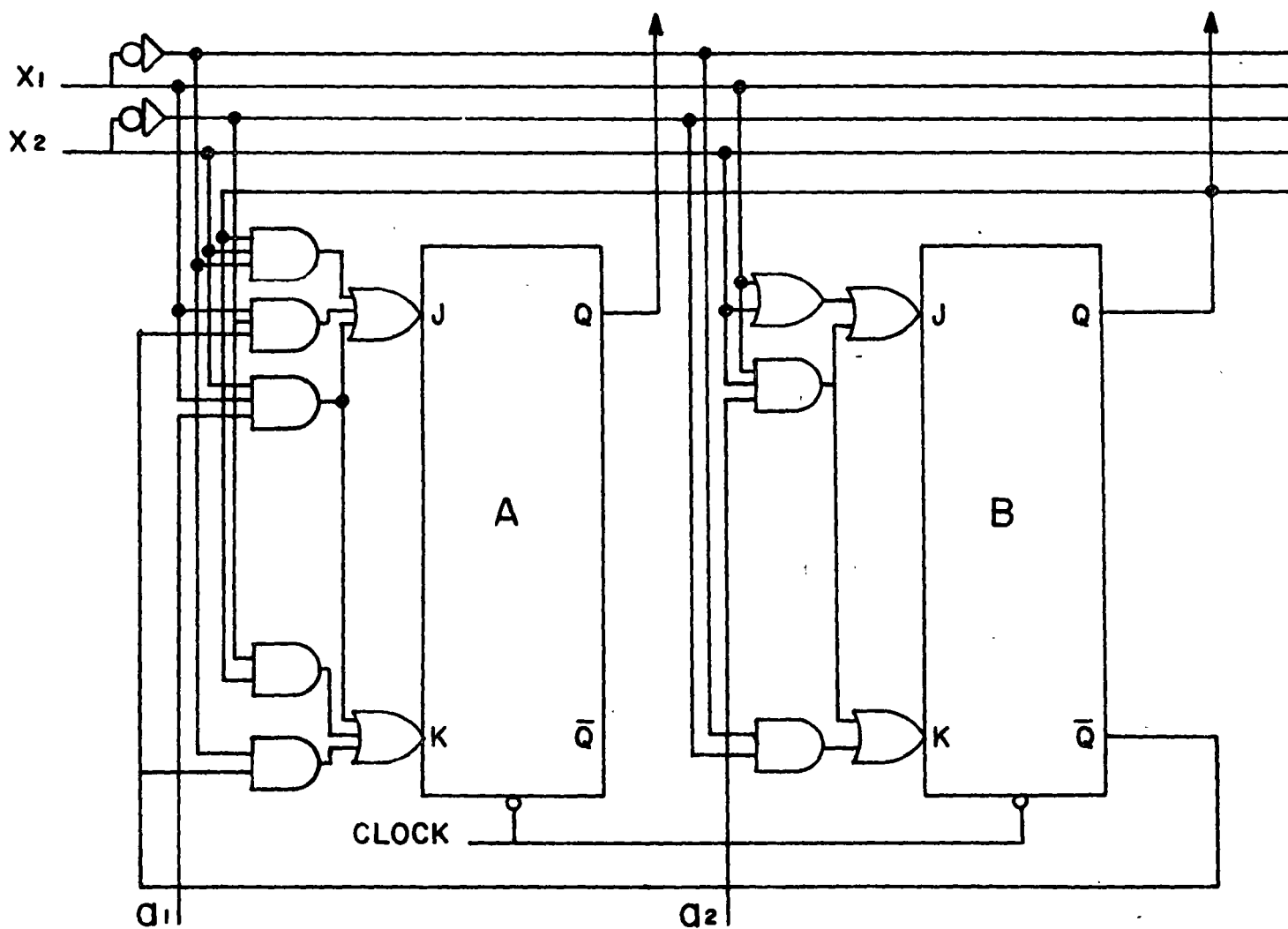
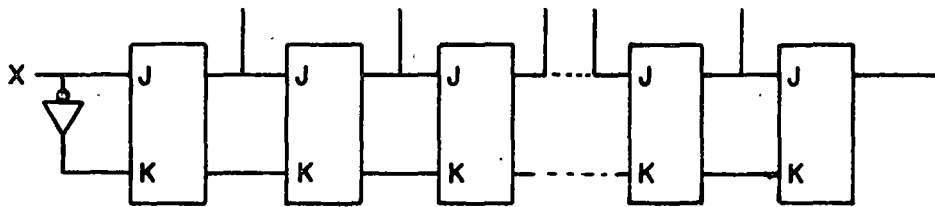


Figura 5.21

## 5.5 Registros de Corrimiento.

5.5.1. Los registros de corrimiento son otros de los elementos más utilizados en el diseño lógico y su máxima aplicación se encuentra en los sistemas de cómputo los cuales manejan paquetes o conjuntos de bits de información.

Un registro de corrimiento es fácilmente diseñado a través de las técnicas presentadas en el capítulo 1. Supongamos por ejemplo una realización con FFKJ como se muestra en la figura 1.22.



Este constituye uno de los casos más simples de corrimiento: "corrimiento a la derecha"; en la práctica existen registros de corrimiento a la derecha e izquierda al igual que en forma circular.

## 5.6 Conclusión:

El método de diseño presentado en este capítulo puede ser aplicado fácilmente en forma manual hasta para 6 variables. Sin embargo, al hacerlo con mayor número de variables, se requerirá utilizar un programa de computadora.

Se aconseja al estudiante interesado en este tema acuda a la bibliografía de autómatas en lo referente a método de particiones y reducción de

estados al igual que realización de autómatas en paralelo, serie o feed back.

## CAPITULO 6: CIRCUITOS SECUENCIALES ASINCRONOS.

### 6.1. Introducción.

Dentro del diseño de circuitos secuenciales es bastante común el no contar con una señal de sincronía o de reloj lo que da base a introducirnos en el diseño de circuitos secuenciales asíncronos, es decir, independientes de señales de sincronía. La conveniencia en la aplicación de tal diseño consiste en evitar la limitante de velocidad que tiene un circuito al depender de una señal de reloj, excepto en los casos en que la frecuencia de este sea igual o mayor que la velocidad de conmutación de dicho circuito.

En la práctica se cuenta con circuitos cuya velocidad de switcheo o conmutación es del orden de  $2 \times 10^9$  segundos: Por ejemplo, en una compuerta NAND en tecnología ECL (Emitter Coupled Logic) si observamos la frecuencia de un reloj capaz de controlar óptimamente a este tipo de dispositivos, tendríamos que es de varios millones de Hertz en nuestro caso: 500 MHz; como es lógico suponer, tales frecuencias son difíciles de obtener y trabajar con ellas implica un diseño sujeto al cálculo de capacitancias y a un cuidado riguroso en la separación entre los elementos del circuito, ya que físicamente la velocidad de propagación de una señal eléctrica a estas frecuencias puede fácilmente alterar el funcionamiento del circuito desincronizándolo. Cuando este problema se presente será

conveniente aplicar las técnicas y métodos de síntesis de circuitos secuenciales asíncronos.

## 6.2. Circuitos en modo fundamental.

La representación de un circuito asíncrono (Fig. 1) tiene como característica principal el trabajar con niveles y no con pulsos, pudiendo tenerse cambios en las entradas  $\{X_1, X_2, \dots, X_n\}$  en cualquier instante.

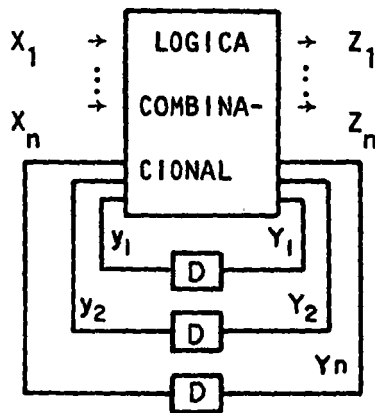


Figura 6.1.

Como dispositivos de memoria se utilizan elementos de retardo o delays.

### 6.2.1. Estado interno, estado externo, variables de excitación, estado estable, estado inestable.

Las variables de estado  $\{y_1, y_2, \dots, y_n\}$  definen el estado interno del circuito.

Las variables de entrada  $\{X_1, X_2, \dots, X_n\}$  definen el estado externo.

Las variables de excitación forman el conjunto  $\{Y_1, Y_2, \dots, Y_n\}$ .

Es estado estable se define cuando  $y_i = Y_i$  para toda  $i$ .



Al efectuarse un cambio en cualquier entrada, la lógica combinacional producirá un nuevo conjunto de valores para las variables de excitación así el circuito tenderá a estabilizarse cuando las variables de estado asuman el mismo valor de las variables de excitación. Obviamente, después de un cierto delay. Aquellos estados o conjuntos de estados transitorios, hasta lograrse la estabilización, serán los estados inestables.

6.2.2. Cuando en un circuito las variables de entrada cambian solamente después de un cierto tiempo al suficiente para que este se estabilice, diremos que la operación se encuentra en modo fundamental.

6.2.3. Restricciones de diseño.

- a) Solamente una de las entradas puede variar a un mismo tiempo.
- b) El tiempo de cambio entre dos entradas debe ser mayor o igual al tiempo mínimo de conmutación de los elementos que constituyen al circuito.

Este tipo de restricciones se deben a la imposibilidad de asegurar que las señales de entrada coincidan exactamente en su arribo al circuito, de hecho los elementos de delay o memoria pueden no existir y los retardos deberse únicamente a los tiempos de switcheo de los elementos combinacionales del circuito.

Sin embargo, se incluyen en la figura para clarificar la exposición.

### 6.3. Síntesis.

#### 6.3.1. Diagrama de tiempos.

Una forma sencilla de verificar visualmente el funcionamiento de un circuito es plantear su diagrama de tiempos en el cual se representa el cambio de las variables de entrada y la correspondiente salida en función del tiempo como se muestra en la siguiente figura:

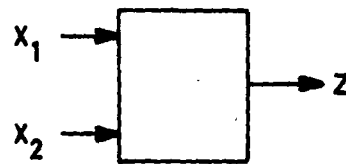
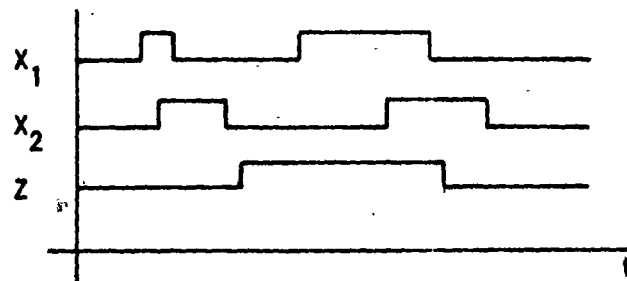


Figura 6.2.



Esta representación será la base del diseño y para cada problema deberemos dibujar el diagrama de tiempos correspondiente como primer paso.

#### 6.3.2. Tabla de flujo.

Al igual que en el diseño de circuitos síncronos es posible representar el funcionamiento del circuito en una tabla de flujo.

### 6.3.2.1. Construcción de la tabla de flujo.

Consideremos el siguiente ejemplo: Se tiene un circuito con 2 entradas  $X_1$ ,  $X_2$  y una salida  $Z$ . Consideremos que el estado inicial se presenta cuando  $X_1 = X_2 = 0$ . El circuito responde con un "1" a su salida cuando  $X_1 = 1$   $X_2 = 0$  y la entrada precedente es  $X_1 = 1$ ,  $X_2 = 1$ .

El primer paso como indicamos será realizar el diagrama de tiempos del circuito.

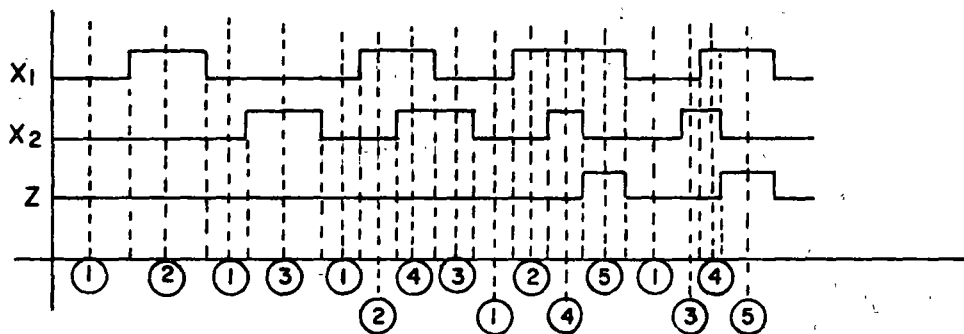


Figura 6.3.

Para construir la tabla de flujo empezamos por colocar una etiqueta en cada combinación de las entradas. A continuación de acuerdo con la secuencia de entrada/salida determinamos los estados estables encerrando en un círculo el número correspondiente como se muestra en la figura siguiente:

| $X_1$   | $X_2$ | $X_1$ | $X_2$ | $X_1$ | $X_2$ | $X_1$ | $X_2$ | $X_1$   | $X_2$ | $X_1$ | $X_2$ | $X_1$ | $X_2$ | $X_1$ | $X_2$ |
|---------|-------|-------|-------|-------|-------|-------|-------|---------|-------|-------|-------|-------|-------|-------|-------|
| 0       | 0     | 0     | 1     | 1     | 0     | 1     | 1     | 0       | 0     | 0     | 1     | 1     | 0     | 1     | 1     |
| ①       |       |       |       |       |       |       |       | 0       |       |       |       |       |       |       |       |
|         |       |       |       | ②     |       |       |       |         |       |       | 0     |       |       |       |       |
|         |       | ③     |       |       |       |       |       |         | 0     |       |       |       |       |       |       |
|         |       |       |       |       |       | ④     |       |         |       |       |       |       | 0     |       |       |
|         |       |       |       | ⑤     |       |       |       |         |       |       |       | 1     |       |       |       |
| ESTADOS |       |       |       |       |       |       |       | SALIDAS |       |       |       |       |       |       |       |

Tabla de Flujo Parcial.

Figura 6.4.

Note usted que en esta tabla se han representado únicamente los estados estables.

A continuación determinaremos para cada trayectoria señalada el estado por el que forzosamente pasa el circuito de un estado estable a otro. Por ejemplo, para pasar del estado 1 al 3 hay un estado inestable al que llamaremos con el mismo nombre que el estado al que tiende, es decir 3. De la misma forma 1 3 lo llamaremos 2, etc....; la salida para estos estados inestables no está especificada, sin embargo puede por motivos prácticos tomarse como idéntica a la del estado destino, con esto se asegura que el cambio en la salida sea tan rápido como sea posible cuando exista. Haciendo referencia a las restricciones de diseño ciertas transiciones están prohibidas cuando existen dos o más

cambios en las variables de entrada; es decir, cuando  $X_1X_2 = 00$  y cambian a  $X_1X_2 = 11$  o bien cuando  $X_1X_2 = 01$  y cambian a 10 ó de 01 a 10. Tomando en consideración lo anterior procedemos a plantear la tabla de flujo primitiva.

| $X_1 X_2$ | $X_1 X_2$ | $X_1 X_2$ | $X_1 X_2$ | $X_1 X_2$ | $X_1 X_2$ | $X_1 X_2$ | $X_1 X_2$ |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0 0       | 0 1       | 1 0       | 1 1       | 0 0       | 0 1       | 1 0       | 1 1       |
| ①         | 3         | 2         | —         | 0         | 0         | 0         | —         |
| 1         | —         | ②         | 4         | 0         | —         | 0         | 0         |
| 1         | ③         | —         | 4         | 0         | 0         | —         | 0         |
| —         | 3         | 5         | ④         | —         | 0         | 1         | 0         |
| 1         | —         | ⑤         | 4         | 0         | —         | 1         | 0         |

Tabla de flujo primitiva.

Figura 6.5.

En la tabla anterior consideramos las salidas de los estados inestables de tal manera que la transición sea inmediata. Note que cada renglon contiene solamente un estado estable.

#### 6.3.2.2. Reducción de la tabla de flujo.

Como puede observarse la tabla presenta estados y salidas incompletamente especificados, lo que era natural al tomar en consideración las restricciones de diseño.

Al igual que en circuitos secuenciales síncronos es conveniente minimizar lo más posible la tabla de flujo de tal manera que el circuito resultado sea a su vez mínimo.

### 6.3.2.3. Tabla completa.

La tabla planteada quedará completa al efectuar una analogía con el diseño de secuenciales síncronos e introducir el concepto de estado actual y estado siguiente:

| Estado Actual | 00 | 01 | 10 | 11 | 00 | 01 | 10 | 11 |
|---------------|----|----|----|----|----|----|----|----|
| ①             | ①  | 3  | 2  | -  | 0  | 0  | 0  | -  |
| ②             | 1  | -  | ②  | 4  | 0  | -  | 0  | 0  |
| ③             | 1  | ③  | -  | 4  | 0  | 0  | -  | 0  |
| ④             | -  | 3  | 5  | ④  | -  | 0  | 1  | 0  |
| ⑤             | 1  | -  | ⑤  | 4  | 0  | -  | 1  | 0  |

Tabla completa.

Figura 6.6.

### 6.4. Autómatas incompletamente especificados.

Como observamos anteriormente el diseño en modo fundamental nos conduce a un tipo de autómatas que se encuentra incompletamente especificado. Sin entrar a la teoría de autómatas o tratar de fundamentar la validez de estos autómatas plantearemos el método a aplicar en su reducción.

El lector interesado puede acudir a la bibliografía de referencia.

Un autómata incompletamente especificado es aquel en que ciertos estados y/o salidas o bien no se conocen, no interesan, o, como en este caso no pueden ocurrir.

#### 6.4.1. Tabla de compatibilidad.

La tabla de compatibilidad se forma de la siguiente manera: Realice una gráfica en forma de escalera con  $N-1$  peldaños en donde  $N$  es el número de estados, es decir para nuestro ejemplo, en donde  $N=5$ .

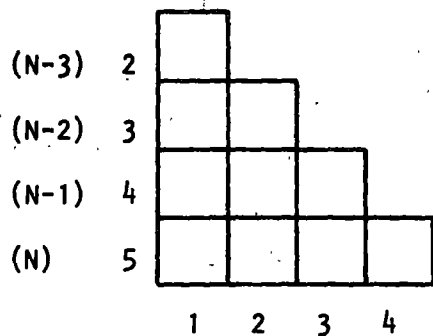


Figura 6.7.

(N-4) (N-3) (N-2) (N-1)

Etiquetando las columnas y los renglones como se muestra en la figura tenemos todas las posibles conjunciones entre estados

Las casillas de la gráfica se llenan con una ( $\checkmark$ ) en aquellos casos en que la intersección de dos estados: sus salidas sean iguales para todas sus combinaciones de entrada (o bien no especificadas) y con una ( $\times$ ) en aquellos que sus salidas difieran:

Para nuestro ejemplo:

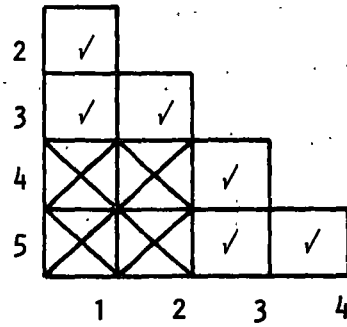


Figura 6.8.

6.4.2, Estados de máxima compatibilidad procediendo de derecha a izquierda enumere los estados compatibles:

(45), (35), (34), (23), (1,3), (1,2)

con estos trataremos de formar ternas, cuartetos etc.... viendo que exista compatibilidad entre todos los elementos:

|                |             |
|----------------|-------------|
| 4              | (4,5)       |
| 3              | (453)       |
| 2              | (453) (23)  |
| 1              | (453) (231) |
| CLASES →       | (453) (231) |
| MAXIMO         |             |
| COMPATIBILIDAD |             |

Figura 6.9.



En este caso las clases de compatibilidad máxima cubren directamente a todos los estados del autómata y son necesarias para su implementación en caso de encontrar un gran número de clases deberá escogerse de entre ellas las que contengan el mayor número de estados y en la selección final asegurar que las clases escogidas cubran todos los estados del autómata original con el menor número de clases.

(4 5 3) (2 3 1)  
 NECESARIO ↑      ↑ NECESARIO PARA CUBRIR  
 PARA CUBRIR      EL ESTADO 1 Y EL 2.  
 EL ESTADO 4  
 Y EL ESTADO  
 5.

Haciendo  $A = (453)$  y  $B = (231)$  podemos plantear la siguiente tabla:

Tome cualquiera de los estados que integran una clase y refiérase a la tabla completa Figura 6. Revise en donde está el siguiente estado si en A o en B y substituyalo junto con su salida en la tabla siguiente:

|   | 00 | 01  | 10 | 11 | 00 | 01 | 10 | 11 |
|---|----|-----|----|----|----|----|----|----|
| A | B  | A/B | A  | A  | 0  | 0  | 1  | 0  |
| B | B  | A/B | B  | A  | 0  | 0  | 0  | 0  |

Figura 6.10.

En la tabla hay dos transiciones que el diseñador puede escoger (A/B)

Aquí a su gusto puede colocar A o B sin afectar el diseño.

6.5. Implementación.

Utilizando las técnicas del capítulo anterior podemos llevar el diseño hasta sus últimas consecuencias:

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

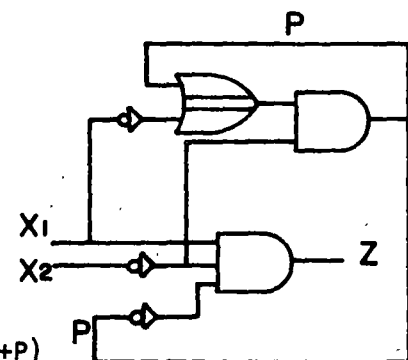
Asignando  $A = 0$  y  $B = 1$ .

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

| $X_1 X_2 P$ | $P^*$ | Z |
|-------------|-------|---|
| 0 0 0       | 1     | 0 |
| 0 0 1       | 1     | 0 |
| 0 1 0       | 0     | 0 |
| 0 1 1       | 0     | 0 |
| 1 0 0       | 0     | 1 |
| 1 0 1       | 1     | 0 |
| 1 1 0       | 0     | 0 |
| 1 1 1       | 0     | 0 |

En donde  $P^* = \bar{X}_1 \bar{X}_2 + \bar{X}_2 P$

|       |         |    |    |    |    |
|-------|---------|----|----|----|----|
|       | $X_2 P$ | 00 | 01 | 11 | 10 |
| $X_1$ | 0       | 1  | 1  | 0  | 0  |
|       | 1       | 0  | 1  | 0  | 0  |



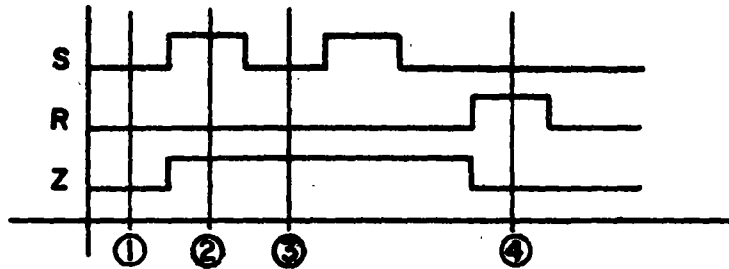
$P^* = \bar{X}_2 (\bar{X}_1 + P)$

y  $Z = X_1 \bar{X}_2 \bar{P}$

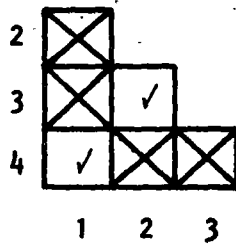
Ejemplo:

Diseñe el siguiente circuito suponiendo que S y R no pueden aparecer

junatas = 1.



|   | 00 | 01 | 11 | 10 | 00 | 01 | 11 | 10 |
|---|----|----|----|----|----|----|----|----|
| ① | ①  | 4  | -  | 2  | 0  | 0  | -  | 1  |
| ② | 3  | -  | -  | ②  | 1  | -  | -  | 1  |
| ③ | ③  | -  | -  | 2  | 1  | 0  | 0  | 1  |
| ④ | 1  | ④  | -  | -  | 0  | 0  | -  | -  |



CLASES: (23) (1,4) si (1,4) = A y (2,3) = B

| SR | 00 | 01 | 11 | 10 | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|----|----|----|
| A  | A  | A  | -  | B  | 0  | 0  | -  | 1  |
| B  | B  | -  | -  | B  | 1  | -  | -  | 1  |

Asignando A = 0 y B = 1

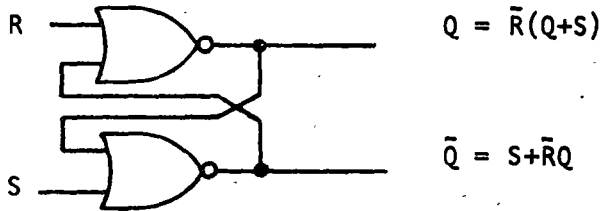
|   |  |    |    |    |    |
|---|--|----|----|----|----|
|   |  | SR |    |    |    |
| Q |  | 00 | 01 | 11 | 10 |
| 0 |  | 0  | 0  | -  | 1  |
| 1 |  | 1  | -  | -  | 1  |

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

$$Q^* = \bar{R}Q + RS$$

$$Q^* = \bar{R}(Q+S) \quad Z = \bar{R}(Q+S)$$

Realización con NORS.

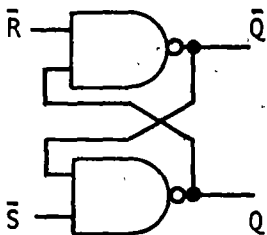


Ya que:  $Q = \bar{R}(Q+S)$  y  $\overline{\bar{R}(Q+S)} = R + \bar{Q}\bar{S}$

y además  $\overline{\bar{Q}\bar{S}} = Q+S$

Realización con NANDS.

En lugar de utilizar  $\bar{R}(Q+S)$  tomemos como punto de partida el negado  $R + \bar{Q}\bar{S}$ , de donde



Ya que:  $\bar{Q} = R + \bar{Q}\bar{S}$  y

$$R + \bar{Q}\bar{S} = \bar{R}(Q+S)$$

Y además  $\overline{\bar{Q}\bar{S}} = \bar{Q}\bar{S}$

Por otro lado  $Q = S + \bar{R}Q$

El circuito anterior en cualquiera de sus implementaciones es el equivalente de un flip flop elemento de memoria utilizado en el diseño secuencial síncrono, pero con el fin de distinguirlo se le denomina latch que en español significa cerrojo.

### 6.6. Asignación de estados en circuitos asncronos.

Hasta ahora en los ejemplos realizados hemos hecho una asignación arbitraria, sin embargo, en los circuitos asncronos la asignación de estados es muy importante y del buen análisis hecho por el diseñador dependerá el funcionamiento o riesgo de falla del circuito realizado.

Por ello la asignación de variables secundarias en las filas de una tabla reducida debe ser tal que el circuito funcione correctamente aún si se asocian diferentes retardos a los elementos secundarios.

### 6.7. Carreras y ciclos.

Consideremos la siguiente tabla y la asignación dada

para  $y_2 y_1$

| $y_2 y_1$ | $x_1 x_2$ |    |    |    |
|-----------|-----------|----|----|----|
|           | 00        | 01 | 11 | 10 |
| 00        | ①         | 2  |    |    |
| 01        |           | 3  | ②  |    |
| 11        | ③         | 4  |    |    |
| 10        |           | ④  |    |    |

| $y_1 y_2$ | $x_1 x_2$ |    |    |    |
|-----------|-----------|----|----|----|
|           | 00        | 01 | 11 | 10 |
| 00        | ①①        | 01 |    |    |
| 01        |           | 11 | ①① |    |
| 11        | ①①        | 10 |    |    |
| 10        |           | ①① |    |    |

En este tipo de circuitos es común encontrar que la transición entre estados estables toque más de un estado inestable hasta realizarse la estabilización. Si para un estado estable inicial y una transición de entradas la secuencia de estados inestables es única, ésta será denominada un ciclo. Por ejemplo, de la tabla mostrada, el circuito ciclará a través de tres estados inestables en la transición de 1 a 4. Note que el siguiente estado para  $y_2y_1x_2x_1 = 0001$  no es  $y_2y_1 = 10$ . En lugar de esto, la máquina se conduce de  $00 \rightarrow 01 \rightarrow 11 \rightarrow 10$ . A este ciclo se puede entrar desde los estados 2 y 3. Cada transición envuelve sólo un cambio en un secundario  $y_1y_2$ .

Cuando más de un cambio en un secundario es especificado a esta situación se le denomina carrera.

La carrera ilustrada en la siguiente figura se considera como no crítica.

| $Y_1Y_2$ |   | $X_1X_2$ |    |    |    |
|----------|---|----------|----|----|----|
|          |   | 00       | 01 | 11 | 10 |
| 00       | ① | 2        |    | 0  |    |
| 01       |   | 2        |    |    |    |
| 11       |   | ②        |    |    |    |
| 10       |   | 2        |    |    |    |

| $Y_1Y_2$ |    | $X_1X_2$ |    |    |    |
|----------|----|----------|----|----|----|
|          |    | 00       | 01 | 11 | 10 |
| 00       | ①① | 11       |    |    |    |
| 01       |    | 11       |    |    |    |
| 11       |    | ①①       |    |    |    |
| 10       |    | 11       |    |    |    |

Si en ésta ya cambia primero, entonces la transición será

$$\textcircled{00} \rightarrow 01 \rightarrow \textcircled{11}$$

pero si  $y_1$  cambia primero

$$\textcircled{00} \rightarrow 10 \rightarrow \textcircled{11}$$

Un ejemplo de carrera "crítica" se muestra en la tabla reducida de la siguiente figura:

| $y_1 y_2 \backslash x_1 x_2$ | 00 | 01 | 11 | 10 |
|------------------------------|----|----|----|----|
| 00                           |    | 00 | 10 |    |
| 01                           |    | 00 | 11 |    |
| 11                           |    | 00 | 10 |    |
| 10                           |    | 10 | 10 |    |

Considere por un momento que  $y_1 y_2 = 11$  y  $x_1 x_2 = 01$ . La transición requerida debe concluir en el estado estable 00. Ahora si  $y_1$  cambia primero, es decir  $y_1 y_2 = 01$  la transición será correcta y arribaremos a 00 pero si  $y_2$  cambia primero, es decir  $y_1 y_2 = 10$ , entonces el circuito arribará al estado 10 permaneciendo en este debido a que es estable por sí mismo.

Las carreras pueden evitarse usando estados inestables intermedios, siendo el caso de la transición para  $y_1 y_2 x_1 x_2 = 0111$  en donde usando como pilote el estado inestable 11 se logra el paso del estado 01 a 10 sin caer en carreras críticas.

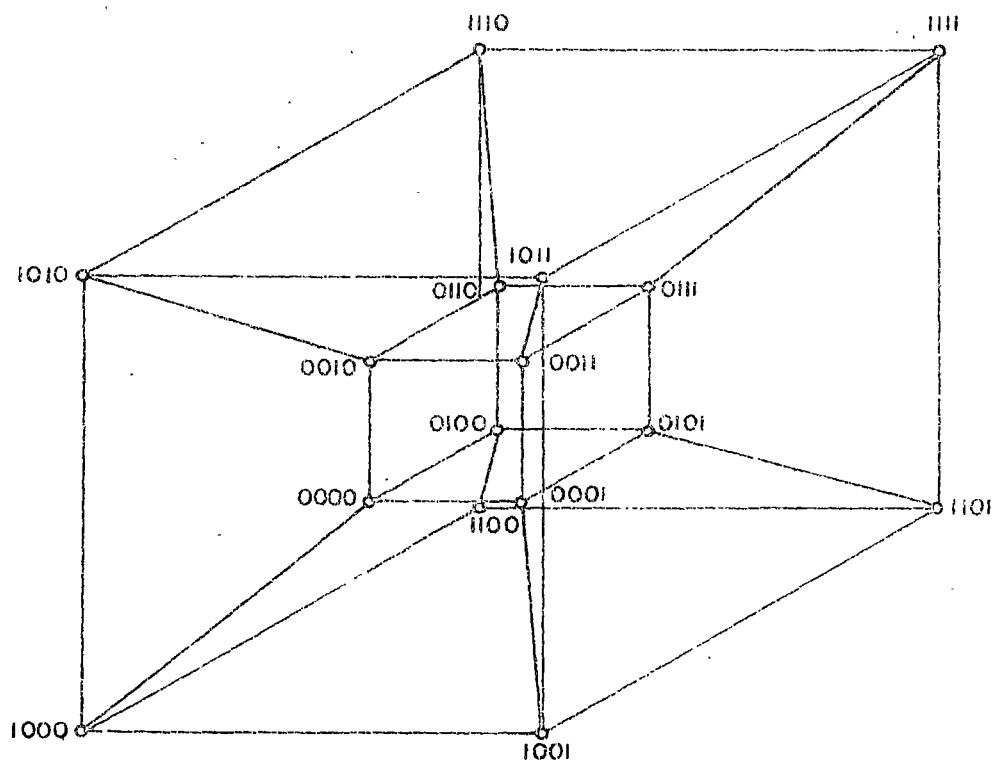
Cuando el circuito pasa a través de una secuencia única de estados inestables se dice realiza un ciclo. Si un ciclo no arriba a un estado estable, se dice oscilará hasta que exista un cambio en las entradas del circuito.

6.8. Métodos de asignación.

Un método sencillo de asignación consiste en definir aquellos estados que difieren en un solo valor en las variables de estado como estados adyacentes.

Para indicar la adyacencia podemos incluir un hipercubo o un mapa de Karnaugh o bien un código de Hamming.

Para ejemplificar mostramos el hipercubo:



En esta figura es fácil analizar por ejemplo: que 0000 es adyacente a 0001, 0010, 0100, 1000 y 1100 y 1111 es adyacente a 0111, 1101, 1110, 1011.



Consideremos ahora la siguiente tabla:

|   | 00 | 01 | 11 | 10 |
|---|----|----|----|----|
| a | ①  | 3  | ④  | ⑥  |
| b | 1  | 3  | ⑤  | ⑦  |
| c | ②  | ③  | 5  | 6  |

Examinando rápidamente esta tabla concluimos que los renglones a y b deben ser adyacentes ya que involucran un cambio solamente en sus variables de estado. Además de esto debemos concluir que:

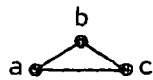
Para  $X_1 X_2 = 00$  : b es adyacente a a

01 : a y b debe ser adyacente a c

11 : c debe ser adyacente a b

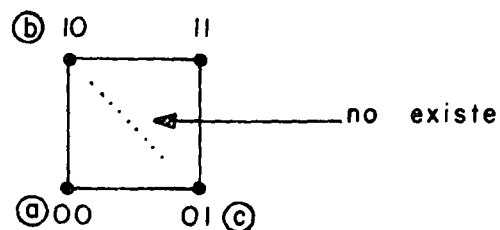
10 : 0 debe ser adyacentes a a

es decir



a b y c deben ser adyacentes.

Sin embargo, para 3 estados se requieren dos variables:  $(y_1, y_2)$  y son posibles asignaciones 00, 01, 10 y 11 haciendo referencia al hiper-cubo:

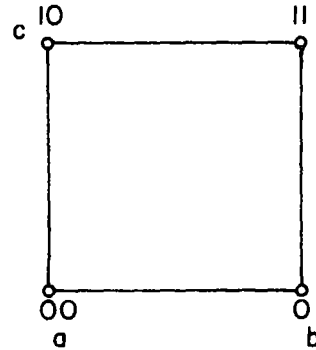


Notamos que para cumplir las adyacencias requeridas es necesario:

- 1) Utilizar más variables de estado.
- 2) Utilizar combinaciones no especificadas como pivote.

En general la alternativa 2 es la más económica así la asignación sería:

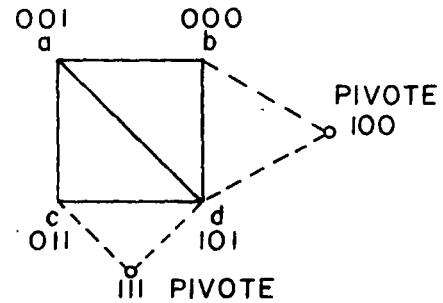
|      |     |    |     |     |
|------|-----|----|-----|-----|
| a—00 | ⊙00 | 10 | ⊙00 | ⊙00 |
| b—01 | 00  | 10 | ⊙01 | ⊙01 |
| c—10 | ⊙10 | 10 | 11  | 00  |
| 11   | -   | -  | 01  | -   |



Como puede observarse utilizamos el estado  $y_1 y_2 = 11$  como pivote para pasar de 01 a 10.

Ejemplo:

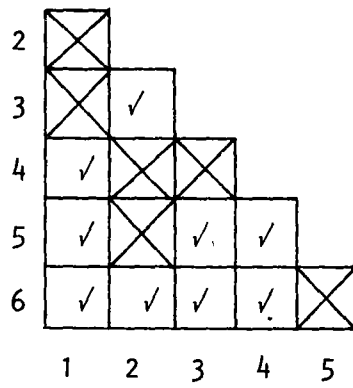
|   |    |    |    |    |
|---|----|----|----|----|
|   | 00 | 01 | 11 | 10 |
| a | ⊙1 | ⊙2 | 4  | ⊙6 |
| b | 1  | 3  | ⊙4 | ⊙7 |
| c | 1  | 2  | ⊙5 | ⊙8 |
| d | 1  | ⊙3 | 5  | 6  |



Recordando que para 4 estados se requieren 2 variables las adyacencias se tienen que violar para este caso y agregando una tercer variable.

|   |    |    |    |    |
|---|----|----|----|----|
|   | 00 | 01 | 11 | 10 |
| 0 | b  | a  | c  |    |
| 1 | *  | d  | *  |    |

↑      ↑  
pivotes



Classes

(145) (236)

|     | 00 | 01 | 11 | 10 | 00 | 01 | 11 | 10 |
|-----|----|----|----|----|----|----|----|----|
| 145 | 1  | 4  | 5  | 2  | 0  | 0  | 0  | 1  |
| 236 | 3  | 4  | 6  | 2  | 1  | 0  | 1  | 1  |

JK

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

| S | R | Q* |
|---|---|----|
| 2 | 0 | Q  |
| 0 | 1 | 0  |
| 1 | 0 | 1  |
| 1 | 1 | -  |

De donde la solución será:

|     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|
| a → | 001 | 001 | 001 | 000 | 001 |
| b → | 000 | 001 | 100 | 000 | 000 |
| c → | 011 | 001 | 001 | 011 | 011 |
| d → | 101 | 001 | 101 | 111 | 001 |
| * → | 100 | -   | 101 | -   | -   |
| * → | 111 | -   |     | 011 | -   |

### 6.9. Diseño utilizando Latch SR.

En el diseño asíncrono no podemos utilizar flip flops debido a la característica de estos de depender de una señal de reloj o sincronía, en su lugar utilizaremos latches.

|   |   | JK |    |    |    |    |    |    |    |
|---|---|----|----|----|----|----|----|----|----|
|   |   | 00 | 01 | 11 | 10 | 00 | 01 | 11 | 10 |
| 1 | 1 | 1  | 4  | -  | 2  | 0  | 0  | -  | 1  |
| 2 | 3 | 3  | -  | 6  | 2  | 1  | -  | 0  | 1  |
| 3 | 3 | 3  | 4  | -  | 2  | 1  | 0  | -  | 1  |
| 4 | 1 | 1  | 4  | 5  | -  | 0  | 0  | 1  | -  |
| 5 | - | -  | 4  | 5  | 2  | -  | 0  | 1  | 1  |
| 6 | - | -  | 4  | 6  | 2  | -  | 0  | 0  | 1  |

| JKy | *y | Y | SR | S <sub>1</sub> R <sub>1</sub> |
|-----|----|---|----|-------------------------------|
| 000 | 0  | 0 | 0* | 0 *                           |
| 001 | 1  | 1 | *0 | * 0                           |
| 010 | 0  | 0 | 0* | 0 *                           |
| 011 | 0  | 0 | 01 | 0 1                           |
| 100 | 1  | 1 | 10 | 1 0                           |
| 101 | 1  | 1 | *0 | * 0                           |
| 110 | 1  | 0 | 10 | 0 *                           |
| 111 | 0  | 1 | 01 | * 0                           |

KQ

| S <sub>1</sub> | 00 | 01 | 11 | 10 |
|----------------|----|----|----|----|
| 0              | 0  | X  | 0  | 0  |
| 1              | 1  | X  | X  | 0  |

$$S_1 = \overline{JK}$$

KQ

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

KQ

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

S=JK

$$S = J\overline{Q}$$

$$R = KQ$$

$$R_1 = \overline{JK}$$

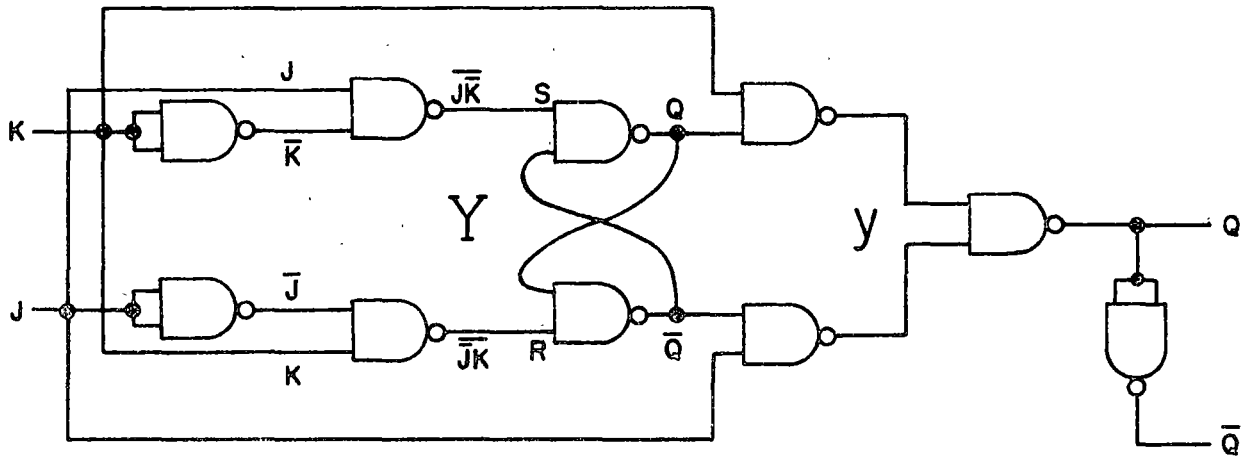
$$S_1 = \overline{JK}$$

$$R_1 = \overline{JK}$$

KQ

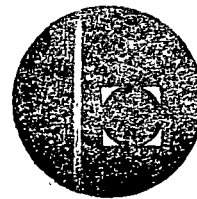
| J | 00 | 01 | 11 | 10 |
|---|----|----|----|----|
| 0 | *  | 0  | 1  | *  |
| 1 | 0  | 0  | 1  | 0  |

R=KQ





centro de educación continua  
división de estudios superiores  
facultad de ingeniería, unam



APLICACIONES DE CIRCUITOS INTEGRADOS AL  
DISEÑO DIGITAL

CAPITULO 7: FAMILIAS LOGICAS E INTERFACES

DR. JOSE FRANCISCO ALBARRAN N.  
JUNIO 1978

## CAPITULO 7 FAMILIAS LOGICAS

### 7.0 INTRODUCCION.

7.0.1 Una clasificación.- El procesamiento digital moderno está basado en forma absoluta en la electrónica de estado sólido desde la década de los 50's con transistores, pasando por los 60's y los primeros circuitos integrados de pequeña-mediana escala, a la década actual en la que los circuitos integrados de gran escala han hecho realidad el microprocesador. En la actualidad existen tres formas principales de distinguir los circuitos integrados digitales: en base a la tecnología de fabricación, en base al funcionamiento interno y en base al grado de integración. La primera distinción tiene dos grandes grupos: Bipolar y MOS (Metal-óxido-semiconductor). La segunda en cierto modo combina en parte el proceso de fabricación y el mecanismo de acción interna del circuito, distinguiendo en la actualidad varias "familias" de las cuales las principales son: DTL, TTL, STTL, I<sup>2</sup>L, ECL, NMOS, PMOS, CMOS y CCD. Finalmente, el grado de integración, aunque no está muy bien definido puede ser: pequeño (SSI), con menos de 100 elementos por circuito, mediano (MSI) con menos de 1000 elementos por circuito y grande (LSI).

En el resto de esta sección se describirán someramente algunas de las características de estas familias lógicas.

Por el momento, la tabla 7.1 muestra la interrelación entre las diferentes clasificaciones de familias lógicas.

|     | BIPOLAR      |     |                      | MOS  |       |       |     |
|-----|--------------|-----|----------------------|------|-------|-------|-----|
|     | TTL/DTL/STTL | ECL | I <sup>2</sup> L/MTL | CMOS | N-MOS | P-MOS | CCD |
| SSI | ✓            | ✓   | NO                   | ✓    | NO    | ✓     | NO  |
| MSI | ✓            | ✓   | ✓                    | ✓    | ✓     | ✓     | NO  |
| LSI | NO           | NO  | ✓                    | ✓    | ✓     | ✓     | ✓   |

En dicha tabla se muestra como algunas familias no son empleadas en ciertos grados de integración, ya sea por imposibilidad tecnológica o por inconveniencia comercial.

7.0.2 Características de Transferencia y Margen de Ruido.- La característica de transferencia ideal de una compuerta lógica (por ejemplo una compuerta NAND con todas sus entradas conectadas entre sí), debe hacer una clara distinción entre dos niveles de voltaje, los cuales representan un 1 o un 0. Por ejemplo, la figura 7.1 muestra la característica de transferencia ideal para una compuerta NAND de dos entradas (conectadas entre sí).



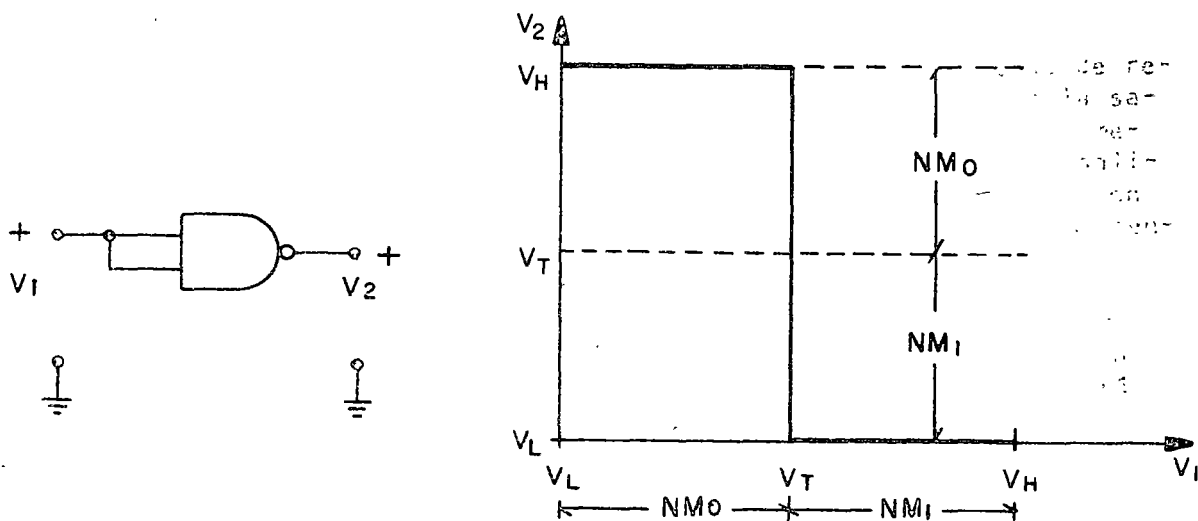


Figura 7.1

Los voltajes están referidos a un punto común (tierra). Mientras la entrada tenga un voltaje cercano a cero, la salida tiene un voltaje alto ( $V_H$ ). Cuando el voltaje de entrada sobrepase el voltaje de umbral ( $V_T$ ), la salida cambiará a un voltaje ( $V_L$ ). Como se supone que esta compuerta está siendo excitada por otra compuerta idéntica, el voltaje de entrada tiene como límites los voltajes máximo ( $V_H$ ) y mínimo ( $V_L$ ) de salida.

Se define como Margen de Ruido (NM) a la diferencia entre el voltaje máximo (mínimo) obtenible a la salida de la compuerta excitadora y el mínimo (máximo) tolerable antes de que el nivel lógico sea irreconocible. En este caso ideal, el margen de ruido para un CERO a la salida ( $NM_0$ ) es ( $V_H - V_T$ ), mientras que el margen de ruido para un nivel UNO a la salida ( $NM_1$ ) es ( $V_T - V_L$ ). También en este caso ideal, para que ambos márgenes de ruido sean máximos se debe tener que  $V_T = \frac{1}{2} (V_H - V_L)$ .

En la práctica, no es posible tener el caso ideal, aunque algunos circuitos (como los CMOS) se aproximan bastante.

**7.0.3 Tiempo de conmutación y potencia.**- La conmutación de un CERO a un UNO, o a la inversa, en una compuerta lógica no sucede en forma instantánea. El tiempo que tarda en conmutar depende del diseño del circuito y de la carga que tenga conectada a la salida. Las causas que producen este fenómeno, así como el análisis del mismo, se salen del propósito de estas notas, por lo que se limitarán a describir la forma que este parámetro se especifica.

En la figura 7.2 se muestra un diagrama en el tiempo de las señales de entrada y salida de una compuerta inversora (NAND o NOR)

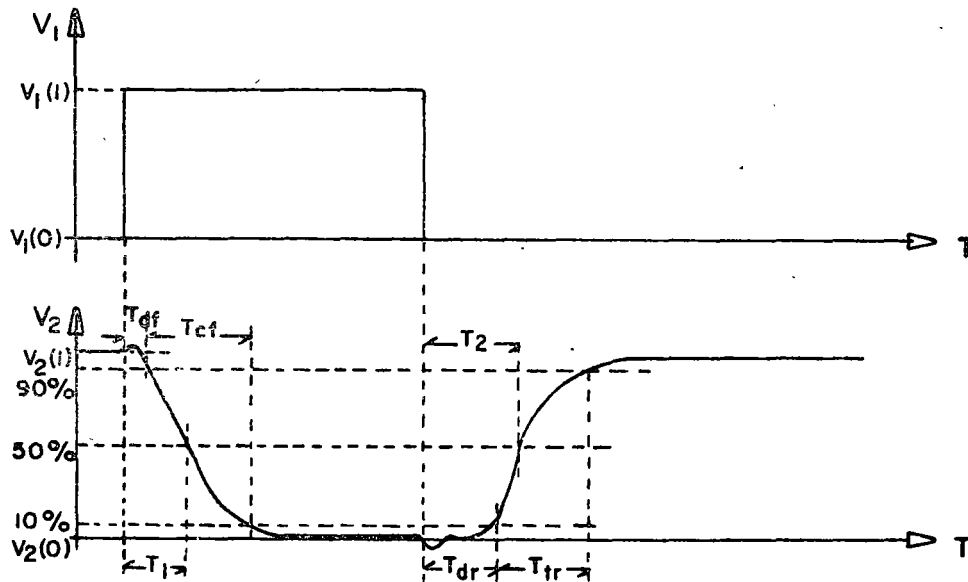


Figura 7.2

El tiempo de conmutación de un estado a otro se considera compuesto de dos etapas: tiempo de retardo inicial ( $t_0$ ) y tiempo de transición. ( $t_t$ ). El primero representa el tiempo que tarda el circuito en empezar a cambiar, mientras que el segundo marca el tiempo que tarda en llegar de un 10% del valor final a un 90% de dicho valor.

Otra forma de medir el retardo de la señal es medir el tiempo que tarda en llegar al 50% de la transición total, desde un CERO ( $t_1$ ) y desde un UNO ( $t_2$ ), dándose como "tiempo de propagación" el promedio de ambas  $t_p = \frac{1}{2} (t_1 + t_2)$ .

La especificación de estos tiempos de propagación, retardo, etc., se hacen considerando al circuito en ciertas condiciones de carga; normalmente se especifican con una carga equivalente a 10 compuertas similares, aunque esto no es siempre así.

La potencia disipada en un circuito depende en mucho de su diseño y de la tecnología de fabricación. Igualmente depende del valor de la fuente de poder que emplee y de la carga capacitiva que esté conmutando.

La potencia "estática" es aquella que disipa la compuerta en estado de reposo (o sea sin conmutar), ya sea estando en un CERO o en un UNO a la salida. La potencia dinámica es aquella que disipa la compuerta en promedio, conmutando a su máxima de velocidad con una onda cuadrada a la salida (es decir con un "ciclo de trabajo" del 50%). Estas potencias no son iguales y aunque a veces son parecidas, la potencia dinámica es normalmente mayor.

Existe también una relación entre la velocidad de conmutación de un circuito y la potencia dinámica que disipa. Normalmente, para una familia dada, el producto Potencia Dinámica x Retardo (PXD) es constante y está dado por:

$$P \times D = \frac{1}{2} C V_{CC} AV_L$$

En donde: C es la carga capacitiva a la salida del circuito,  $V_{CC}$  es el valor de la fuente de alimentación y  $AV_L$  es la excursión de voltaje entre UNO y el CERO Lógicos del circuito.

## 7.1 FAMILIAS LOGICAS BIPOLARES.

Las familias lógicas bipolares con las que el diseñador actual se encontrará en la práctica son, principalmente: TTL, STTL, ECL, DTL. Aunque I<sup>2</sup>L está teniendo un gran auge, su presencia en un circuito es "transparente" al usuario, ya que siempre aparece con interfaces apropiados para interconectarse con TTL.

7.1.1 DTL (Diode-transistor-Logic).- Este tipo de circuito lógico apareció primeramente en forma discreta (es decir no-integrada) en los albores del procesamiento digital moderno. Junto con RTL (resistor-transistor-Logic), formaban las familias lógicas más usadas, y aún en la actualidad, en ocasiones se hace necesario usar algunos de estos circuitos en problemas sencillos. El uso de circuitos discretos tiene ciertas ventajas, como flexibilidad de operación, a cambio de una gran cantidad de desventajas: costo, dificultad de diseño (para el que no conoce los rudimentos del diseño electrónico), espacio ocupado, baja velocidad. Por lo anterior, sólo se usa en contadas ocasiones.

El principio de funcionamiento de la compuerta DTL se ilustrará por medio del circuito que se muestra en la figura 7.3 y se describe a continuación.

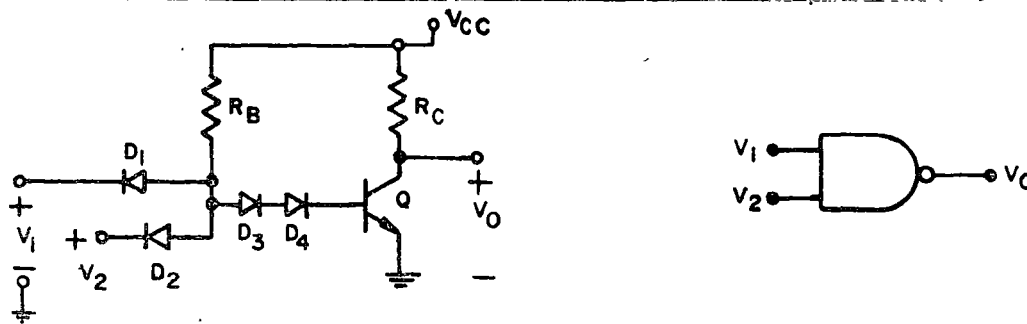


Figura 7.3

Primeramente, en forma cualitativa, se puede apreciar que la función lógica de este circuito es la de una compuerta NAND (considerado al nivel 1 como un voltaje alto positivo, y al nivel 0 como un voltaje cercano a cero). Cuando los dos voltajes de entrada ( $V_1$  y  $V_2$ ) son altos (por ejemplo  $V_1 = V_2 \cong V_{CC}$ ), los diodos  $D_1$  y  $D_2$  se encuentran en inversa, y por lo tanto el transistor  $Q$  queda encendido a través de  $R_B$ ,  $D_3$  y  $D_4$ . En cambio si cualquiera de las entradas es un CERO (por ejemplo  $V_1=0$ ), el potencial en el nodo B es demasiado pequeño para que  $Q$  pudiera encenderse, quedando éste apagado. Cuando el transistor  $Q$  está encendido, la corriente que circula a través de  $R_C$  fuerza al voltaje de salida ( $V_0$ ) a ser bajo (o sea un CERO). Por el contrario, al estar  $Q$  apagado, la corriente a través de  $R_C$  es cero y el voltaje de salida es  $V_0 = V_{CC}$  (o sea un uno). Por lo tanto, en este circuito se tendrá un UNO a la salida siempre que una de las entradas sea un CERO, y sólo se tendrá un CERO a la salida cuando ambas entradas sean UNO; es decir es una compuerta NAND.

Para mostrar el margen de ruido de esta compuerta es necesario añadir la excitación y la carga a que será sujeta. Ambas son normalmente compuertas del mismo tipo, así que se empleará el circuito de la figura 7.4, en la que se ha añadido como carga al circuito de entrada de otra compuerta DTL, y se han propuesto valores para las resistencias. En añadidura se requiere conocer los parámetros del transistor.

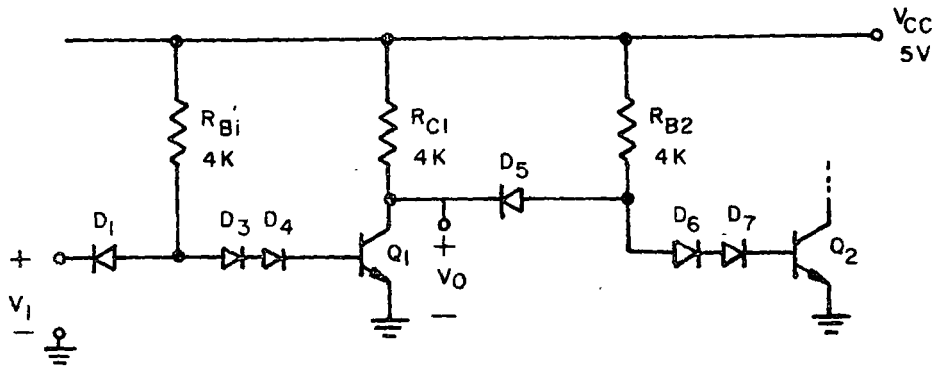


Figura 7.4

En este caso, para simplificar al problema se considerarán las siguientes características para el transistor:  $V_{CE}(\text{sat}) = 0.2\text{V}$ ,  $V_{BE}(\text{encendido}) = 0.6\text{V}$  y  $\beta(\text{min}) = 40$ .

Para los diodos se considerará que:  $V_D(\text{encendido}) = 0.6\text{V}$  y que en reversa llevan una corriente de fuga  $I_R = 10\mu\text{A}$ .

Con estos datos se puede proceder a analizar el circuito. Primero se considerará el caso en que  $V_1$  es alto ( $V_1 = V_{CC}$ ). En esta condición el circuito se puede representar por la figura 7.5, en la que se han eliminado los diodos y transistores que quedan apagados.

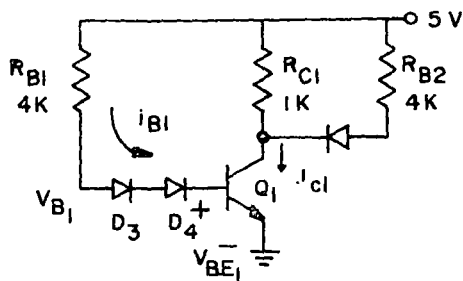


Figura 7.5

De la figura se puede obtener que:

$$i_{B1} = \frac{V_{CC} - V_{D3} - V_{D4} - V_{BE1}}{R_{B1}} = \frac{3.2V}{4K} = 0.8mA$$

y, suponiendo que  $Q_1$  queda saturado (suposición que se comprobará posteriormente).

$$i_{C1} = \frac{V_{CC} - V_{ce1}(\text{sat})}{R_{C1}} + \frac{V_{CC} - V_{DS} - V_{ce1}(\text{sat})}{R_{B2}}$$

$$= \frac{4.8V}{1K} + \frac{4.2V}{4K} = 0.585 \text{ mA}$$

Obviamente,  $\beta i_{B1} > i_{C1}$ , con lo que se comprueba que  $Q_1$  está saturado.

La solución de punto de operación anterior dá el primer punto mostrado en figura 7.6.

Ahora se propone disminuir el voltaje  $V_1$  hasta llegar al punto en el que  $Q_1$  sale de saturación y se encuentra en la región activa. Este punto será aquel en el que:  $V_1 \cong (V_{BE1} + V_{D3} + V_{D4}) - V_{D1} = 1.2V$  (ver punto 2 en la figura 7.6) De este punto, la ganancia del inversor determinará a qué voltaje se apaga el transistor. Este voltaje es típicamente del orden de 1.1V a 1.0V.

Cuando  $Q_1$  ha sido apagado (punto 3 en la figura 7.6), la única corriente que circula por  $R_{C1}$  es la corriente de fuga en los diodos, con lo que queda:

$$V_O(1) = V_{CC} - R_{C1} I_R = 4.99 \text{ V}$$

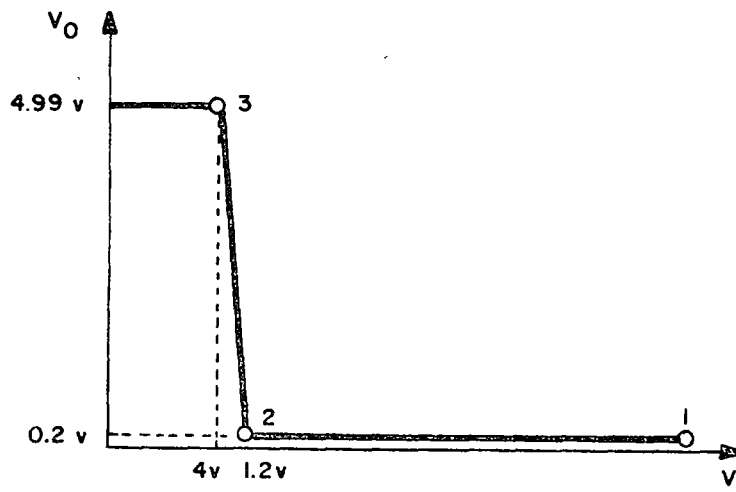


Figura 7.6

Aunque el análisis anterior ha ido simplificando algunos problemas del circuito, representa el método usual de análisis manual de las características estáticas de la compuerta. Existen variaciones en los parámetros del circuito que cambian la característica de transferencia, tales como: temperatura, número de compuertas conectadas a la salida y variaciones propias de las componentes del circuito (tolerancias, envejecimiento, etc.). Algunas de estas variaciones se especifican normalmente en las compuertas comerciales, como se mostrará más adelante.

El margen de ruido de este circuito, para las condiciones "ideales" en lo que ha sido presentado es:

$$NM_1 = (4.99 - 1.2) \cong 3.8 \text{ V.}$$

$$NM_0 = (1.0 - 0.2) = 0.8 \text{ V.}$$

Sin embargo, estos márgenes se verán afectados notablemente por las variaciones mencionadas en los párrafos anteriores.

Los tiempos de retardo y propagación no se pueden definir aquí y dependerán del circuito empleado. La potencia estática es máxima para el cero a la salida y vale alrededor de 6.5MW.

**7.1.2 TTL (transistor - Transistor Logic.** Sin duda, esta es la compuerta más empleada actualmente a nivel de SSI y MSI. Tiene varios años

de desarrollo y ofrece al diseñador una amplia gama de funciones lógicas como: multiplexores, demultiplexores, contadores, compuertas, flip-flops, etc.

Desde el punto de vista de circuito, es una compuerta DTL un poco modificada. El circuito de la figura 7.7 es prácticamente equivalente a DTL de la figura 7.3, con las siguientes modificaciones: primero, los diodos  $D_1$  y  $D_2$  son reemplazados por dos diodos formados por uniones base-emisor (este procedimiento es válido solo para circuitos integrados), el diodo  $D_3$  es reemplazado por el diodo formado por la unión base-colector del transistor de entrada ( $Q_1$ ); el diodo  $D_4$  es reemplazado por un transistor, aumentando la  $\beta$  efectiva del transistor de salida. La adición del Resistor  $R_1$  permite una ruta de descarga a la base de  $Q_3$ , con lo que mejora la velocidad de conmutación; esta resistencia está normalmente presente también en un DTL práctico.

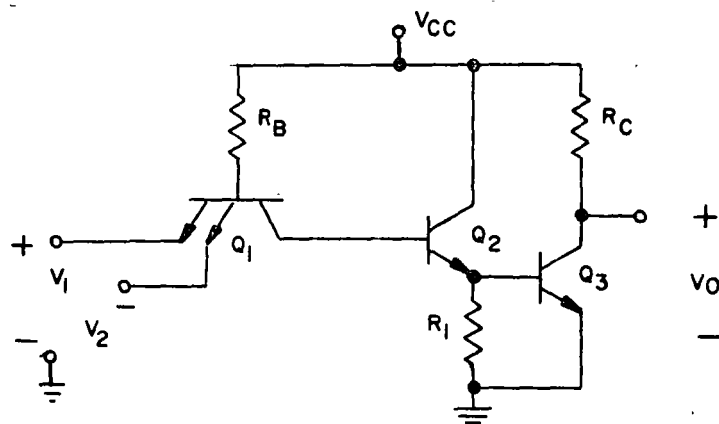


Figura 7.7

Aunque esta compuerta TTL es común, aunque por lo general no incluya a  $R_C$  (por lo que se le llama circuito de "colector abierto"), el circuito TTL por excelencia emplea una etapa del tipo pugh-pull, llamada "Totem pole". Este circuito, ya clásico, se muestra en la figura 7.8. El transistor de entrada  $Q_1$  integra los diodos que forman la esencia de la compuerta DTL. Los diodos  $D_2$  y  $D_3$  sirven para fijar excursiones negativas en el voltaje de entrada, y evitar la destrucción de la compuerta. El transistor  $Q_2$  actúa en parte como un amplificador, proporcionando a las bases de  $Q_3$  y  $Q_4$  señal en contrafase (es decir, que cuando el



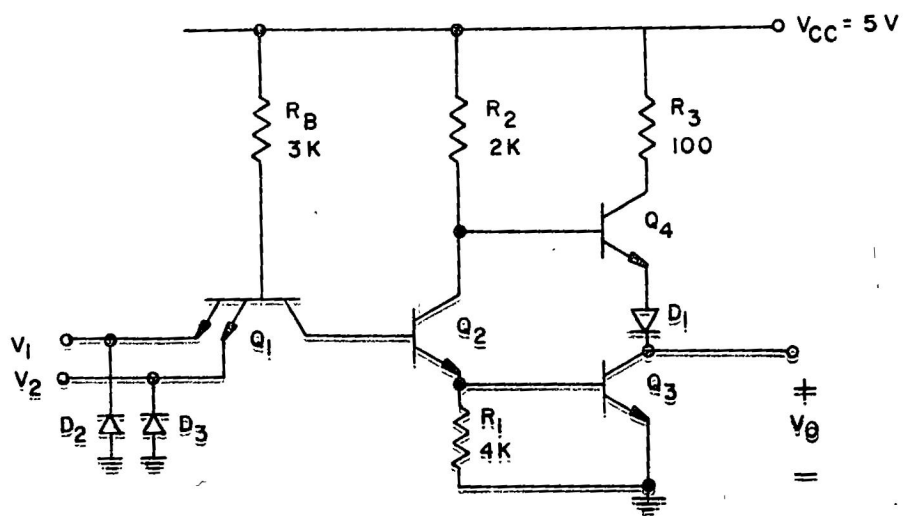


Figura 7.8

voltaje en la base de  $Q_3$  aumenta, el de la base de  $Q_4$  disminuye, y viceversa). Los transistores de salida,  $Q_3$  y  $Q_4$ , proporcionan la corriente necesaria para accionar las demás compuertas a que está conectada ésta.

El diodo  $D_1$  tiene por objeto evitar que  $Q_4$  se encienda mientras  $Q_3$  está encendido en estado estable.

El principal propósito del circuito Totem-pole es tener la capacidad de proporcionar mucha corriente al circuito de carga, con el propósito de aumentar la velocidad de propagación del circuito.

A continuación se describe este circuito, con algunos valores típicos empleados en la conocida serie 7400 de compuertas TTL (ver figura 7.8). Los parámetros de los transistores, por ilustración serán iguales a los especificados para la compuerta TTL. Al igual que para la DTL, se "cargará" a la compuerta, sólo que esta vez se utilizará el equivalente a diez compuertas TTL, para ilustrar el efecto de la carga. La carga se simulará con el circuito de la figura 7.9, que es también lo recomendado por los fabricantes para esta prueba. El diodo DA representa la unión base-emisor de entrada; los diodos DB, DC y DD representan los voltajes de la fijación de:

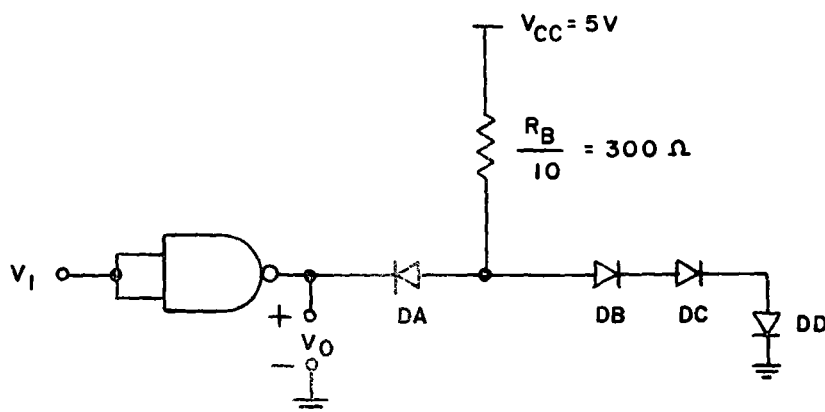


Figura 7.9

base colector de  $Q_1$  y base emisor de  $Q_2$  y  $Q_3$ . La resistencia se ha igualado a la resistencia  $R_b/10$ , para simular la carga de diez compuertas en paralelo.

Para analizar el circuito se supondrá primeramente que el voltaje de en-

trada es alto ( $V_1 = V_2 = V_{cc}$ ).

Con esto, al eliminar el circuito a los elementos que quedan cortados o apagados, queda el circuito de la figura 7.10

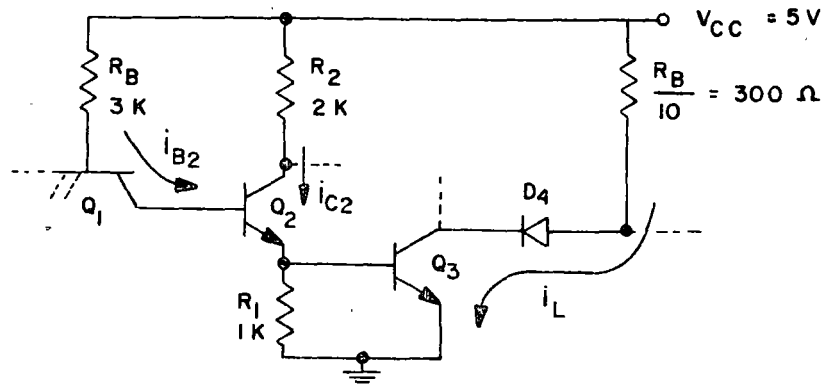


Figura 7.10

Obviamente,  $Q_2$  y  $Q_3$  están encendidos, y por lo tanto:  $V_{B3} = 0.6V$  y  $V_{B2} = 1.2V$ . Con esto se conoce que  $i_{B2} = \frac{V_{cc} - V_{BC1} - 1.2V}{R_B} \cong 1mA$

Con esta corriente de base,  $Q_2$  queda saturado, por lo que el voltaje en su colector es aproximadamente de  $0.8V$ . Por lo anterior se puede encontrar que  $i_{C2} = \frac{4.2V}{2K} = 2.1mA$ . Como la corriente a través de  $R_1$  es  $0.6V/R_1 = .6mA$ , la corriente a la base  $Q_3$  será:

$$i_{B3} = i_{C2} + i_{B2} - \frac{0.6V}{R_1} \cong 2.5mA$$

La corriente de carga está dada por:

$$I_L = \frac{V_{cc} - V_{D1} - V_{CE3}(\text{sat})}{\frac{R_B}{10}} = \frac{4.2V}{.30K} \cong 14mA$$

ya que  $\beta i_{B3} > i_L$ , se comprueba que  $Q_3$  está saturado. Con esto, el voltaje a través de la base-emisor de  $Q_4$  y  $D_1$  es  $V_{C2} = 0.6V$ , lo que asegura que  $Q_4$  está apagado.

Para encender a la unión base-emisor de  $Q_1$ , se requiere que  $V_1 = V_2 \approx V_{BE3} + V_{BE2} + V_{BC1} - V_{BE1}$ , o sea del orden de 1.2V. A partir de entonces, el circuito tiende a apagarse en función de la ganancia de  $Q_3$ . Cuando  $Q_3$  se apaga, la salida depende de  $Q_4 - D_1$ , los cuales a su vez dependen del voltaje  $V_{C2}$ . Este voltaje, tiene un valor conocido una vez que se apaga  $Q_3$ , ya que al no llevar éste corriente de base, la corriente en el colector de  $Q_2$  es casi igual a la de su emisor, lo que a su vez está dada por :

$$i_{C2} \approx \frac{V_{B3}}{R_1}$$

Con esto se tiene que:

$$V_{C2} \approx V_{CC} - V_{B3} \frac{R_2}{R_1}$$

Al acabarse de apagar  $Q_3$ , su voltaje base emisor es muy poco inferior a 0.6V, por lo que en ese punto  $V_{C2} \approx 5 - 0.6 \frac{2}{1} = 3.8V$ . En este punto,

el voltaje de salida vale:  $V_0 = 3.8V - V_{BE4} - V_{D1} = 2.6V$ . Este es el punto 2 de la figura 8.11.

A partir de este momento, la ganancia del circuito está dada por el cociente  $R_2/R_1$ , hasta que  $Q_2$  y el diodo base-colector de  $Q_1$  se apagan.

Esto sucede cuando el voltaje de entrada es un poco menor que:  $V_1 = V_2 \approx V_{BC1} + V_{BE2} - V_{BE1} \approx 0.6V$ . Esto representa el punto 3 en la gráfica de la figura 7.11.

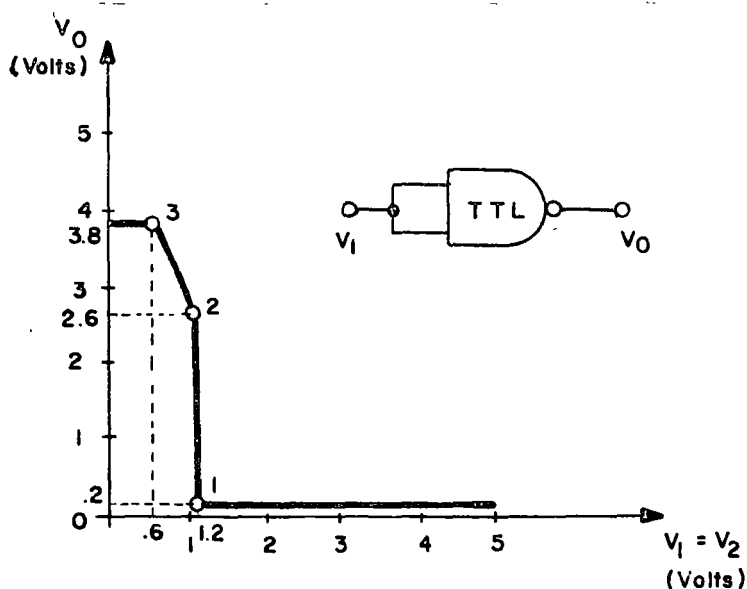


Figura 7.11

Para tensiones menores a ésta, el voltaje de salida permanece aproximadamente en  $V_0(1) = V_{cc} - V_{BE4} - V_{D1} = 3.8V$ .

Conociendo los voltajes límites de la compuerta se puede calcular el margen de ruido. Aquí existe, sin embargo, la disyuntiva de escoger el punto 2 ó 3 de la gráfica como el 1 lógico de la compuerta. Si se escoge el 2, se tendrá que:  $NM(1) \approx 0.9V$  y  $NM(0) \approx 1.5V$ ; en cambio si se escoge 3, se tiene que  $NM(1) = .54$  y  $NM(0) = 2.6V$ .

Nuevamente se hace notar que el ejemplo anterior deja de lado consideraciones de segundo orden que son de máxima importancia para el diseñador del circuito. Algunas de estas consideraciones son: variaciones en los parámetros de los transistores, efecto de la temperatura y aspectos referentes al tiempo de conmutación y disipación de potencia del circuito.

7.1.3 STTL (Schottky TTL). Esta compuerta emplea en añadidura un elemento llamado diodo Schottky, cuyo símbolo se muestra en la figura 7.12a. Este diodo tiene como características fundamentales las siguientes: es fabricable en forma integrada junto con transistores bipolares, su tiempo de propagación es muy inferior al de un diodo de unión PN, y su voltaje de encendido es del orden de 0.4V. Con estas características, el principal uso de este diodo es el de fijador de voltaje para evitar la saturación de los transistores. Al evitar esta saturación, aumenta considerablemente la velocidad de propagación de la señal en estas compuertas.

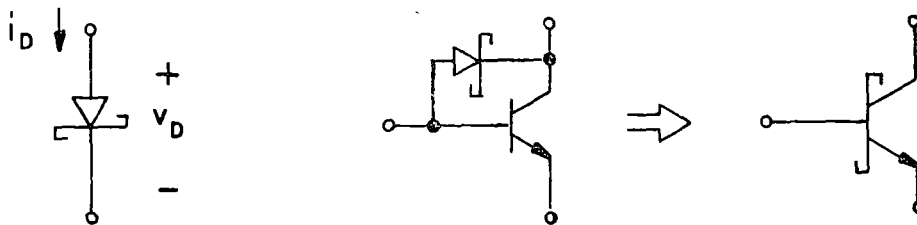


Figura 7.12.

El uso extensivo del transistor con fijador Schottky se ha hecho que se utilice el símbolo que se muestra en la figura 7.12c. Empleando esta simbología la figura 7.13 muestra una compuerta STTL de las más recientes.

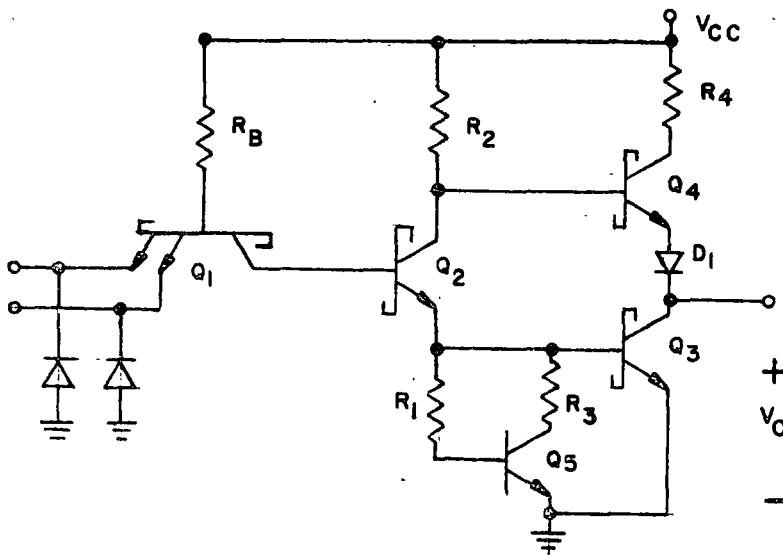


Figura 7.13

Las principales diferencias entre este circuito y el TTL típico son:

- El uso extensivo de fijadores Schottky, con el objeto de reducir el tiempo de transición del circuito.
- Distintos valores de resistencias con el propósito de reducir la potencia disipada en el circuito, así como presentar una impedancia más apropiada a las líneas de  $50\Omega$ .
- La carga "activa"  $Q5-R1-R3$ , que aumenta la ganancia de  $Q2$ , haciendo más empinada la función de transferencia (entre los puntos 2 y 3 de la gráfica de la figura 7.11), aumentando el margen de ruido.

Al ir aumentando su volumen de ventas, la familia STTL ha ido reduciendo sus precios y aumentando la diversidad de sus funciones, estimando que en un futuro próximo STTL dominará a TTL en su empleo dada sus mejores características de conmutación.

El producto  $P \times D$  del STTL es del orden de  $110 \text{ pJoules}$ , mientras que su retardo típico es de  $5 \text{ nseg}$ .

## 7.2. Familia Lógica CMOS (Complementary Metal-Oxide-Semiconductor).

La única familia lógica con transistores MOS que se puede usar en forma MSI y SSI es la familia CMOS. Los circuitos de MOS canal simple se emplean principalmente en circuitos de gran escala (LSI), y su funcionamiento interno es "transparente" al usuario, es decir que la entrada y la salida se parecen a las de las familias de uso común (particularmente TTL).

La familia CMOS en cambio, tiene características particulares que la hacen materia de estudio en este curso. Por un lado son accesibles comer-

cialmente a través de varios fabricantes, y por otro lado ofrecen una alternativa muy significativa a las familias bipolares (particularmente TTL), dando una mayor flexibilidad al arte de diseñar circuitos digitales.

Evitando entrar a detalles de fabricación y diseño, en los párrafos siguientes se describe a un inversor CMOS como el de la figura 7.14.

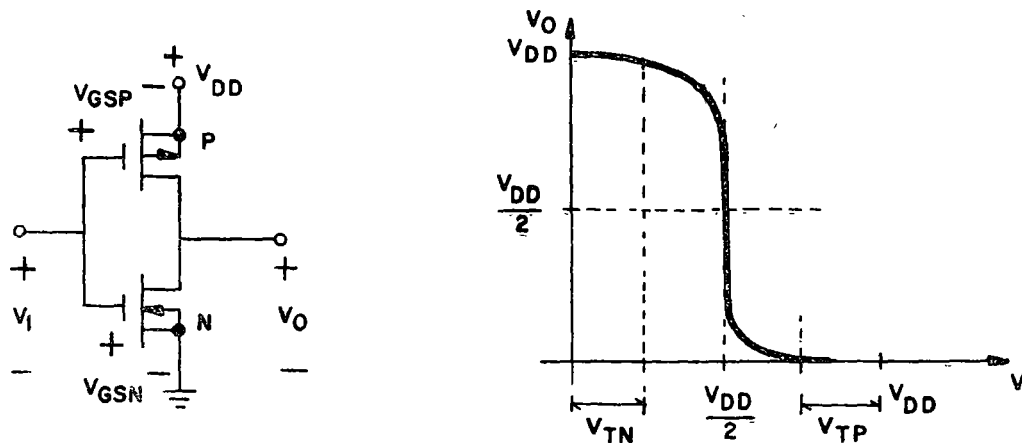


Figura 7.14

Cuando el voltaje de entrada es bajo ( $V_I = 0$ ), el MOS canal N está apagado y el canal P está encendido ( $V_{GSP} = V_{DD}$ ). En estas condiciones, el voltaje de salida es  $V_O = V_{DD}$ , siempre y cuando la "carga" al circuito sea otro circuito CMOS (en cuyo caso la impedancia de carga es muy alta).

Cuando el voltaje de entrada es igual al voltaje de umbral del MOS canal N ( $V_{TN}$ ), dicho transistor empieza a encenderse, "cargando" al MOS canal P (ver punto (2) en la figura 7.14B).

Al seguir subiendo el potencial a la entrada, el voltaje de salida disminuye: primero en forma ligera, y muy abruptamente cuando  $V_I \approx V_{DD}/2$  (ver puntos (3) y (4) en la figura 7.14b). Para  $V_I > V_{DD}/2$ , el voltaje de salida disminuye paulatinamente hasta llegar a cero cuando  $V_I \approx V_{DD} - V_{TP}$ , en cuyo caso el MOS canal P se apaga.

Varias características interesantes son deducibles de la operación del inversor CMOS. Primero, que su margen de ruido es amplio, ya que se puede considerar cuando menos del orden de  $V_{TP} \approx V_{TN}$ , que para la mayoría de los dispositivos comerciales es del orden de 1.5 volts; en todo caso es posible considerar que el margen de ruido sea parecido a  $V_{DD}/2$  (tanto para el 0 como para el 1), siempre y cuando se sea cuidadoso en

el diseño del sistema. Segundo, que en los extremos lógicos la corriente que se drena de la fuente de alimentación es mínima, ya que será solo la corriente de fuga de los dispositivos, la cual es del orden de las nanoamperes; esto significa una disipación de energía sumamente bajo en estado estático (no confundir con la energía que se gasta en el caso dinámico). Tercero que la impedancia de entrada del circuito es sumamente alta (característica inherente al dispositivo MOS), lo que permite un "Fan-out" ilimitado.

Otra importante característica de este inversor en su alta flexibilidad en cuanto al valor de la fuente de alimentación ( $V_{DD}$  puede variar desde 3 volts hasta 15 volts en la mayoría de los dispositivos comerciales).

La disipación dinámica del dispositivo depende de la carga capacitiva del mismo ( $C_L$ ), de la frecuencia a que opera ( $f_0$ ) y de la fuente de alimentación ( $V_{DD}$ ) en la siguiente relación:

$$P(\text{dinámica}) = \frac{1}{2} V_{DD}^2 f_0 C_L$$

Para la familia CMOS, el producto  $P \times D$  es del orden de 1 nJ para  $C_L \approx 20\text{pF}$  y  $V_{DD} = 10\text{V}$  o 250 pJ para  $V_{DD} = 5\text{V}$  y  $C_L = 20\text{pF}$ . La velocidad de propagación de las compuertas CMOS es típicamente de 20 a 50 nseg para las condiciones de carga y alimentación señaladas antes.

Así es que en general, la flexibilidad de la familia CMOS se ve compensada por su baja velocidad, además de ser relativamente frágil, ya que la carga estática del cuerpo humano puede destruir a un circuito integrado MOS, si no se tiene la precaución debida.

**7.2.1 Compuertas CMOS.** Basadas en el funcionamiento del inversor CMOS se pueden generar compuertas NAND y NOR, así como interruptores (también llamadas compuertas de transmisión). Una breve descripción de estas compuertas siguen en las secciones posteriores.

**7.2.1.1 Interruptor o compuerta de transmisión.** Una característica importante del dispositivo MOS es su bilateralidad. A diferencia del bipolar, el circuito MOS, una vez que está encendido, puede conducir en ambas direcciones, es decir que su drenaje y su compuerta son intercambiables. Esta propiedad se muestra en la figura 7.15, en la que aparece la característica voltaje-corriente de un MOS canal N.

por esto es que solo cuando  $V_{GS} = V_{DS}$



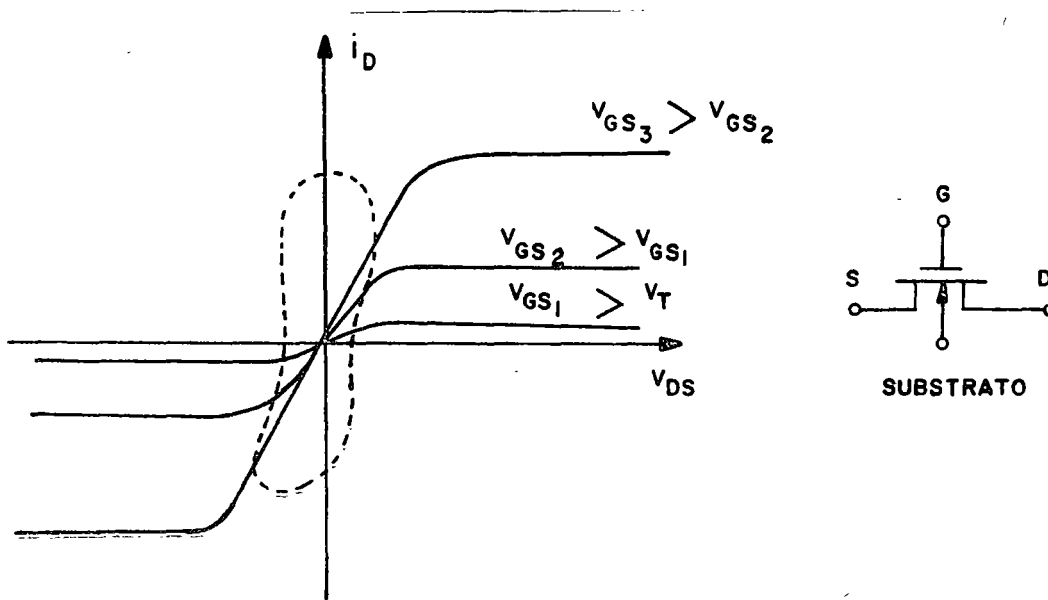


Figura 7.15

Mientras al potencial en la compuerta sea mayor que el potencial en el drenaje y en la fuente, el dispositivo estará encendido. Además, para bajos voltajes entre drenaje y fuente, el MOS se comporta como una resistencia (ver zona punteada en la figura 7.15).

Una compuerta de transmisión CMOS se realiza en la forma mostrada en la figura 7.16a.

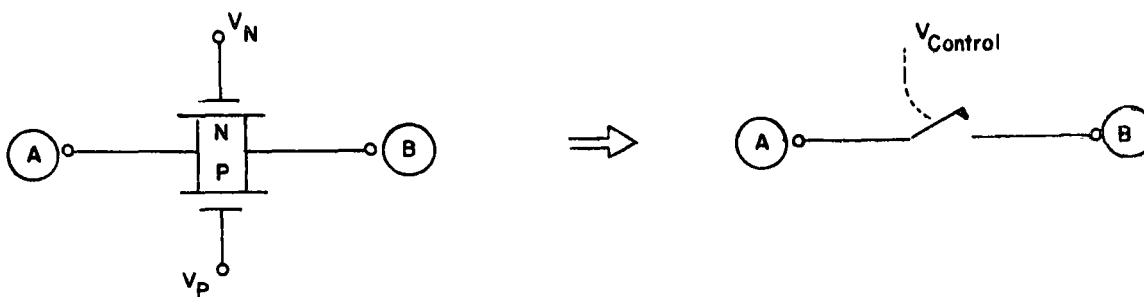


Figura 7.16

Cuando  $V_N = V_{DD}$  y  $V_P = 0$ , la compuerta está encendida y puede transmitir (o sea que equivale a un interruptor cerrado). Cuando  $V_N = 0$  y  $V_P = V_{DD}$ , la compuerta está inhibida (o sea el interruptor está abierto). En la figura 7.16b se muestra el modelo equivalente, o sea un interruptor que es controlado por un voltaje  $V_C$ .

En la familia CMOS es muy conocido el circuito 4016 que consiste de cuatro interruptores con su respectivo control. Cada interruptor tiene el circuito mostrado en la figura 7.17. El inversor CMOS adicional se emplea para asegurarse de que  $V_P$  y  $V_N$  están siempre en estados lógicos opuestos.

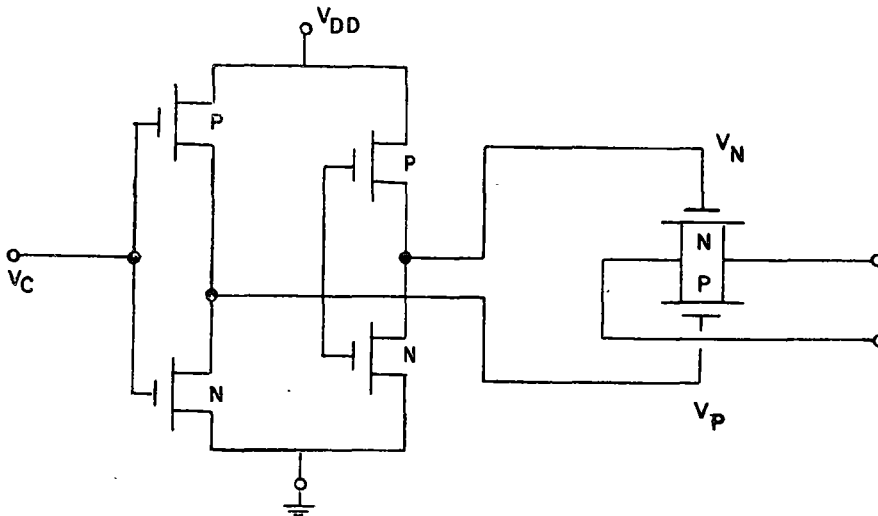


Figura 7.17

**7.2.1.2 Compuerta NAND.** La compuerta NAND se muestra en la figura 7.18. Su funcionamiento estático es como sigue. Cuando  $V_1 = V_2 = 0$ , M1 y M2 están apagados, mientras que M3 y M4 están encendidos, con lo que  $V_0 = V_{DD}$ . En virtud de que M1 y M2 están conectadas en serie, la única forma en la que la salida puede ser 0 es si ambas están encendidas; por esto es que solo cuando  $V_1 = V_2 = V_{DD}$ , se tiene que  $V_0 = 0$ .

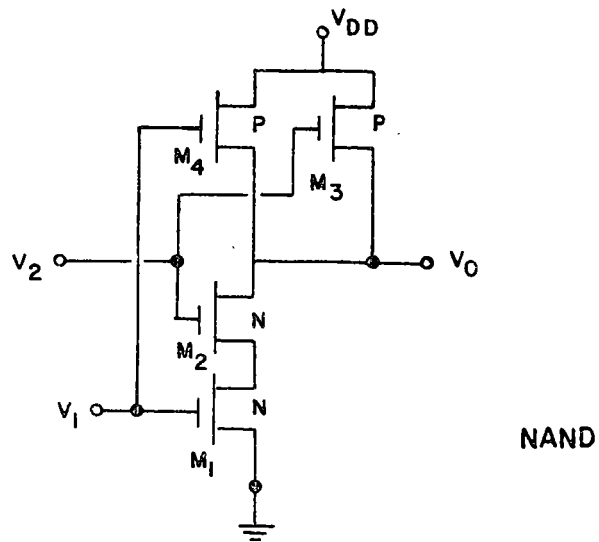


Figura 7.18

7.2.1.3. Compuerta NOR. Esta compuerta es el dual de la anterior y se muestra en la figura 7.19. En ésta, los dispositivos del canal N ( $M_1$  y  $M_2$ ) están en paralelo, por lo que su salida será baja siempre que alguna de las entradas sea alta. La salida será alta solamente cuando  $V_1 = V_2 = 0$ .

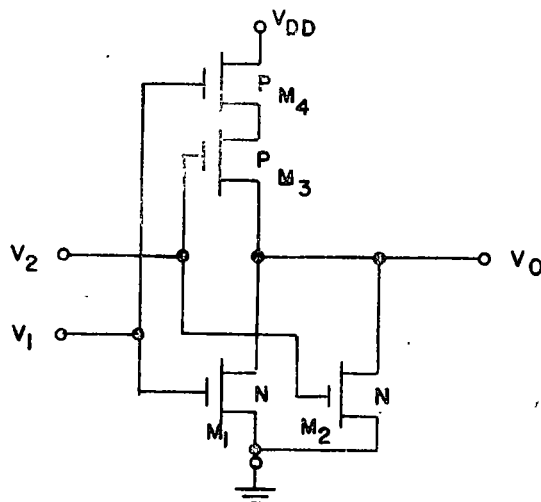


Figura 7.19

## 7.4. INTERCONEXION.

7.4.1. El concepto de BUS. En muchas ocasiones se desea minimizar el cableado de interconexión en un sistema digital. En estas ocasiones es común usar un cable al que se conectan las salidas de varias compuertas y las entradas de otras compuertas. El objetivo es usar este cable para transmitir información de una sola salida a una o más entradas. Por supuesto, durante el tiempo en que una transmisión dada sea efectuada, las demás compuertas no deberán interferir; por lo tanto es necesario inhibir la acción de las compuertas que no deben estar enviando información y de aquellas que no deban recibir dicha información.

En la figura 7.20 se muestra un arreglo en el que tres transmisores se comunican con dos receptoras.

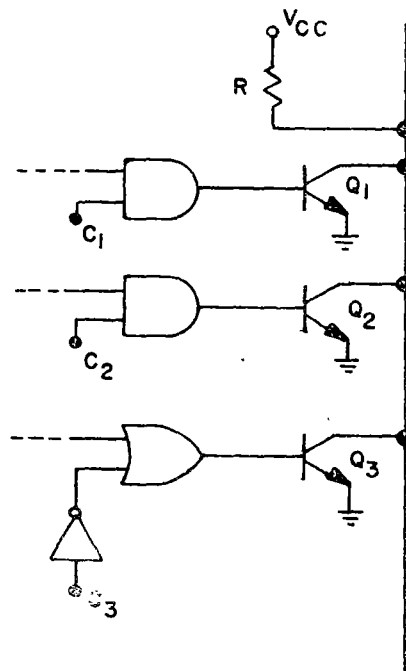


Figura 7.20

La acción transmisora del elemento  $T_i$  se inhibe con la señal de Control  $C_i$ , mientras que la acción receptora del elemento  $R_j$  se inhibe con el control  $E_j$ .

Supóngase que un cero en la señal de control inhibe, y que un UNO habilita. Entonces para comunicar a  $T_2$  con  $R_1$ , se deben tener las señales de control:  $C_1 = 0$ ,  $C_2 = 1$ ,  $C_3 = 0$ ,  $E_1 = 1$  y  $E_2 = 0$ . Si por el contrario, se desea que  $T_3$  transmita tanto a  $R_1$  como a  $R_2$ , se deberá tener:  $C_1 = 0$ ,  $C_2 = 0$ ,  $C_3 = 1$ ,  $E_1 = 1$  y  $E_2 = 1$ .

De esta forma, a expensas de tener que compartir una ruta de transmisión, se puede reducir la interconexión en muchos circuitos. Una de las principales desventajas de este sistema es su lentitud, ya que mientras ocurre una transmisión, las demás deben esperar. Por supuesto, debe existir algún procesador que controle el uso de BUS, es decir, que opere las señales de control de transmisión y recepción.

Desde el punto de vista de los circuitos, existe un importante problema para la implementación de un BUS. Si una compuerta digital solo puede cambiar entre un CERO y un UNO, al conectar varias compuertas a la salida se puede tener un estado indeterminado. Esto pasaría muy claramente si se conectasen directamente a la salida dos compuertas TTL con totem-pole o bien dos compuertas CMOS del tipo descrito anteriormente.

La solución común es de dos tipos:

- a) usar compuertas del tipo "colector abierto"
- b) usar compuertas especiales denominadas "TRI-STATE". A continuación se describe el principio de funcionamiento de estas compuertas.

7.4.2. Interconexión a través de "colector abierto". En la figura 7.21 se muestran tres circuitos con "colector abierto", conectados a la misma línea de BUS.

Cuando las señales de control son un 0 lógico, los transistores se encuentran apagados y por lo tanto, su colector-emisor equivale a un circuito abierto. En este estado, la línea se encuentra "en reposo" con un estado alto (o sea un 1 lógico). La señal se transmite por la línea de BUS, sin que los otros dos intervengan. Por supuesto, no se puede permitir que dos señales de control sean 1 simultáneamente, ya que en ese caso la señal en el BUS es el "AND" de ambas señales.

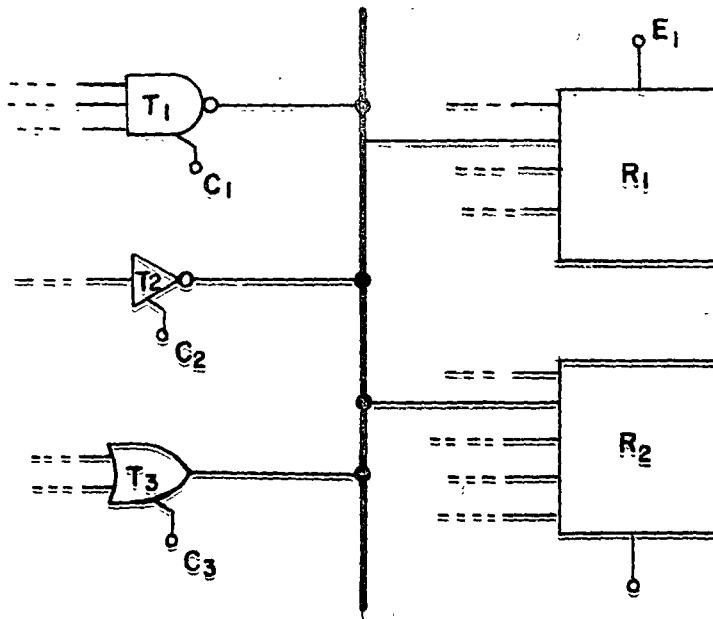


Figura 7.21

7.4.3. TRI-STATE. El sistema a base de "colector abierto" tiene algunos problemas en lo referente a potencia, retardo y número de receptores a que puede dar servicio. Ciertas modificaciones a las compuertas TTL (totem-pole) y CMOS, permite realizar la interconexión al BUS. Estas modificaciones convierten a estas compuertas en las llamadas TRI-STATE debido a que tienen tres estados: los dos lógicos (1y0) y un estado que equivale un circuito abierto y que es llamado "de alta impedancia"

7.4.3.1. TTL/TRI-STATE. La figura 7.22 muestra la modificación más comúnmente usada por los fabricantes de TTL para convertir compuertas al tipo TRI-STATE.

Cuando la señal de control es un 1, el transistor Q7 se halla encendido, con lo que tanto Q3 como Q4-D $\bar{1}$  se encuentran apagados. Al estar encendido Q7, equivale a tener un cero en la compuerta NAND/TTL, lo que produce que Q3 se apague. En añadidura, a través del diodo D2, Q4-D1 son apagados. En este estado la salida del totem-pole equivale a un circuito abierto (excepción hecha de la corriente de fuga de Q3, que es normalmente inferior a 40 $\mu$ A), es decir que queda en el estado de alta impedancia.

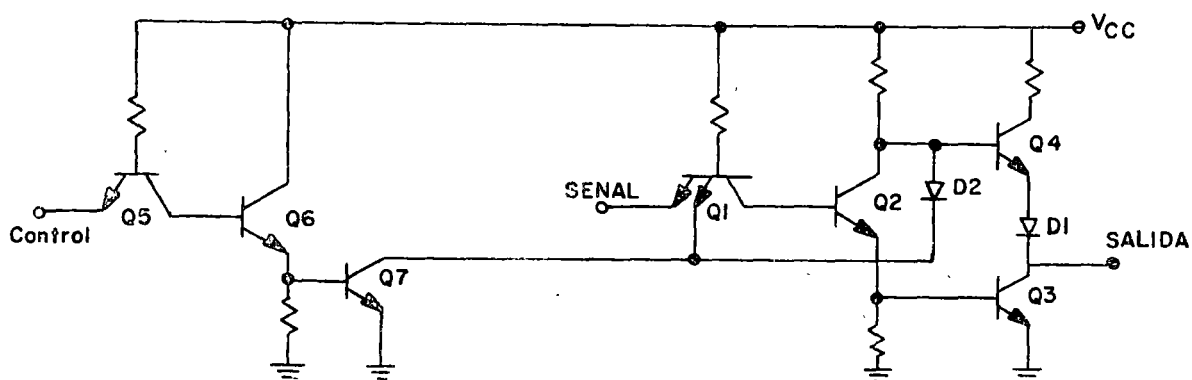


Figura 7.22

Cuando el control está en un nivel CERO, el transistor Q7 está apagado y la señal se transmite normalmente a la salida.

7.4.3.2. CMOS/TRI-STATE. La forma más común de usar un CMOS en TRI-STATE es añadiendo una compuerta de transmisión a la salida de las compuertas lógicas comunes. Cuando la compuerta de transmisión se abre se tiene el estado de alta impedancia y cuando ésta se cierra; se puede tener alguno de los estados lógicos. La figura 7.23 muestra este circuito para un inversor.

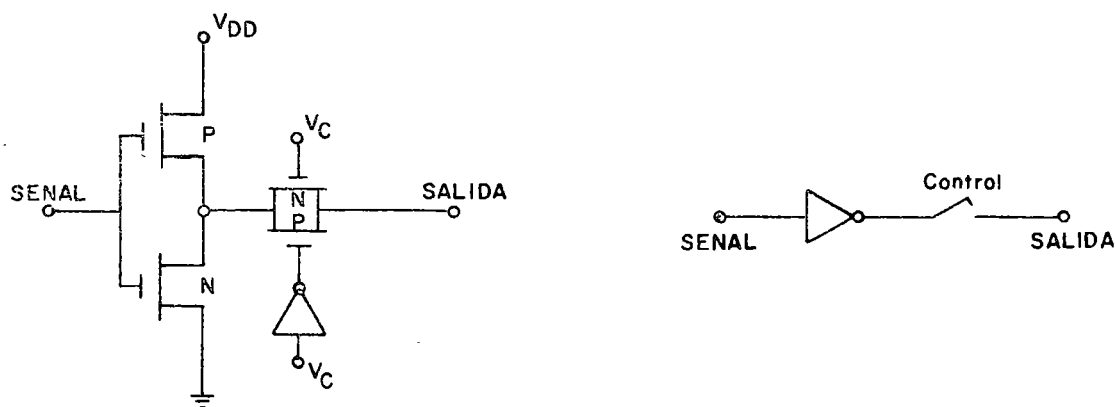


Figura 7.23

Una alternativa consiste en añadir elementos en serie con los transistores MOS de salida, como se muestra en la figura 7.24. Cuando los dispositivos en serie se encuentran apagados ( $C=0$ ), el circuito se encuentra en el estado de alta impedancia.

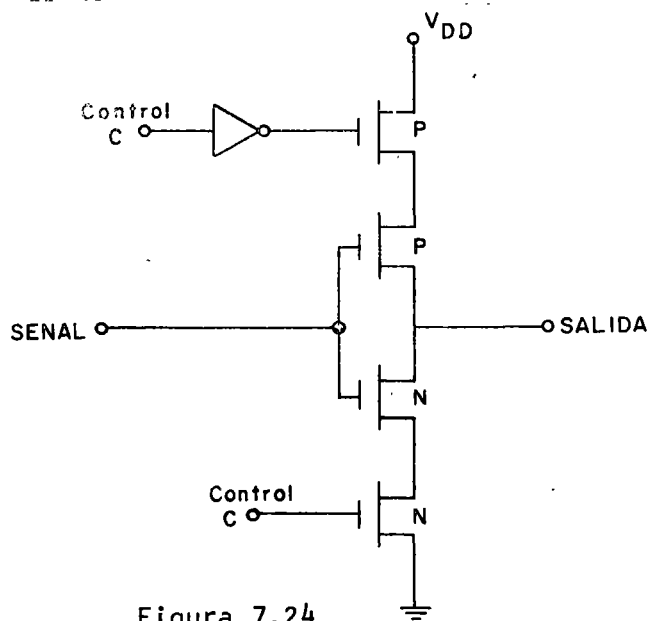
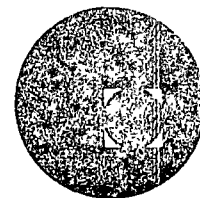


Figura 7.24



centro de educación continua  
división de estudios superiores  
facultad de ingeniería, unam



APLICACIONES DE CIRCUITOS INTEGRADOS AL  
DISEÑO DIGITAL

CAPITULO 8: MEMORIAS RAM, ROM Y SERIE

CAPITULO 9: PROBLEMAS DE INTERCONEXION

M. EN I. CESAR CHAVEZ ZAPATA  
JUNIO 1978



## MEMORIAS

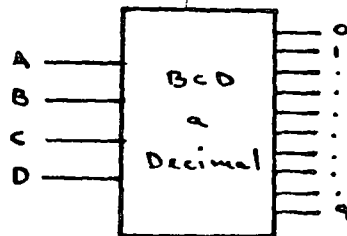
- I. Introducción
- II. Almacenamiento de Información Digital
- III. La Función de un Bloque de Memoria como parte de un Sistema Digital
- IV. RAM
- V. ROM
- VI. MEMORIAS SERIE
- VII. Características Tecnológicas

## MEMORIAS

## I. INTRODUCCION:

Desde el punto de vista de la naturaleza del procesamiento de información en sí, los sistemas digitales pueden dividirse en dos grupos básicos: sistemas combinacionales y sistemas secuenciales.

Los sistemas combinacionales son aquellos en los cuales la resultante del proceso depende directa y únicamente del estado actual de las variables a la entrada del sistema. Considérese como ejemplo la operación de un decodificador BCD-Decimal:



en este sistema, una de las líneas de salida (0 - 9) estará activa dependiendo del estado de las variables de entrada (A - D) de tal forma que si en un instante dado el estado de las variables de entrada se modifica, la situación a la salida se ajusta a la nueva situación de entrada independientemente de cual haya sido el o los estados anteriores.

En los sistemas secuenciales en cambio, la resultante a la salida dependerá tanto del estado actual de las variables a la entrada, como de la historia pasada del sistema. Considérese como ejemplo la operación de un candado de combinación en el cual, la resultante a la salida será que el candado abra. Esto se logra mediante la combinación de la última de las situaciones a la entrada (situación actual) y de las situaciones previas suministradas en el orden determinado por la clave.

La función del sistema secuencial consiste pues en "recordar" las situaciones previas a la entrada y cotejarlas con la situación actual para, de esta forma, tomar decisiones.

Un sistema secuencial requiere de elementos que le permitan almacenar (memoria) las diferentes situaciones que tienen lugar a la entrada y de elementos combinatoriales que le permitan procesar la información para obtener resultados.

La capacidad de tomar decisiones es una característica que hace de cualquier sistema una herramienta poderosa en el procesamiento de información, y cuando la naturaleza del proceso establece que las variables involucradas no concurren simultáneamente, el almacenamiento de información se convierte en una necesidad, de ahí la enorme importancia que se asigna a memorias en el área de sistemas digitales.

## II. ALMACENAMIENTO DE INFORMACION DIGITAL:

Almacenamiento de información digital es el mecanismo mediante el cual la información digital proveniente de alguna fuente, es almacenada físicamente en medios cuya principal característica es la de retener dicha información durante un intervalo de tiempo que proporcione al usuario la libertad de operar confiablemente el sistema para el cual está destinada.

Los principales medios de almacenamiento de información digital son:

- 1) Memorias magnéticas
- 2) Memorias semiconductoras.

### Memorias Magnéticas:

El almacenamiento de información en este tipo de memorias se produce en áreas específicas dentro de una película delgada de material magnético, la cual está ubicada a su vez, sobre una superficie no magnética que le sirve de soporte. La principal característica del material magnético es la de ser capaz de mantener un flujo magnético definido en ausencia de campos externos aplicados. La inserción (escritura) de información o la ob

tención (lectura) de la misma se realiza por medio de una cabeza de lectura/escritura una vez que el área de almacenamiento ha sido localizada. La localización del área específica se realiza ya sea por el movimiento del material magnético solamente (cinta) o bien por el movimiento conjunto del material magnético y de la cabeza (disco). La inserción de información se realiza por medio de la creación de un flujo magnético en la cabeza de escritura el cual a su vez crea un patrón de magnetización en el medio, el cual permanece una vez que éste se aleja de la cabeza. La lectura de información se realiza reconociendo dichos patrones de magnetización los cuales son censados como una corriente inducida en la cabeza de lectura.

La forma más simple de almacenar información digital en medios magnéticos es quizá la grabadora de audio - cassette. Un cassette o cartucho es capaz de almacenar hasta  $10^7$  bits con tiempos de acceso de hasta 100 seg dependiendo de la ubicación del bloque específico dentro de la cinta y de la calidad del sistema. La velocidad de la cinta está limitada por desgaste y calor debidos a que la cabeza está en contacto físico con la cinta.

Esta desventaja se elimina grandemente en sistemas más sofisticados basados tanto en cintas magnética como discos y tambores, empleando un colchón de aire entre la cabeza y el medio magnético. Esto permite incrementar sensiblemente la velocidad del medio magnético y eleva el grado de confiabilidad del sistema de almacenamiento. Las memorias de disco de mayor capacidad almacenan de  $10^9$  a  $10^{10}$  bits en uno o más discos montados en un solo eje accionado por una solo motor. El tiempo de acceso en sistemas típicos es de aproximadamente 20 milisegundos.

El acceso a información en este tipo de memorias es por bloques, el cual depende del movimiento mecánico de sus piezas. Una vez que el bloque deseado ha sido localizado, la velocidad de transferencia depende de la velocidad de la superficie magnética y de la densidad de almacenamiento en el medio.

La característica de la transferencia de información es de  $10^6$  a  $10^7$  bits por segundo típicamente.

Quizá la característica más importante de estos sistemas de almacenamiento es la de ser "no volátiles". Esto implica que la información permanece en el medio magnético aun cuando la energía haya sido removida.

Su principal aplicación es la de "archivos" capaces de almacenar grandes cantidades de información empleada en sistemas de computación.

#### Memorias Semiconductoras:

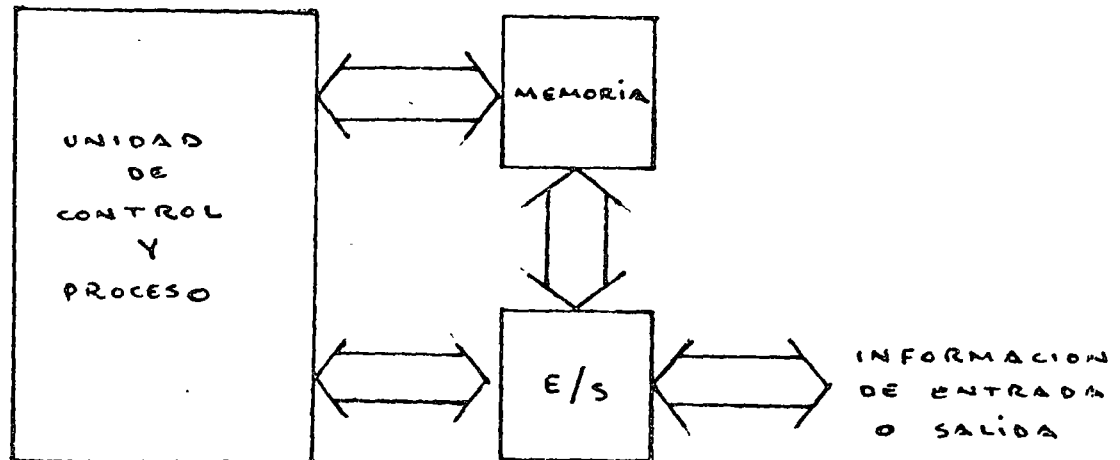
El avance de la tecnología de semiconductores ha hecho posible la realización de elementos con una alta densidad de componentes electrónicos integrados en superficies inferiores a  $1 \text{ cm}^2$ , y uno de los mayores esfuerzos se ha enfocado en la elaboración de memorias.

Este tipo de memorias poseen características que las han situado en un tiempo muy corto, como los elementos que mayor aplicación encuentran en aquellos sistemas digitales que requieren elementos de almacenamiento rápidos, versátiles, y de bajo costo.

Su rapidez estriba en el hecho de que tanto el almacenamiento como la lectura de información se realizan por medios puramente electrónicos. La versatilidad que brinda el ser compatibles eléctricamente con los demás elementos que conforman el sistema, además de poder ser accésados aleatoria o secuencialmente, presenta al diseñador una gran flexibilidad. Otro factor importante en cuanto a versatilidad se refiere, es el de poder utilizar elementos volátiles o no volátiles dependiendo de la aplicación específica. En cuanto a bajo costo, puede establecerse que comparado con elementos como núcleos de ferrita (cada vez más en desuso), tanto en costos de operación, especialmente en consumo de potencia, como en costo de sistemas, los elementos semiconductores ofrecen indiscutiblemente las mejores características. Los diferentes tipos y características de memorias semiconductoras se tratarán con mayor detalle a lo largo de este trabajo.

### III. LA FUNCION DE UN BLOQUE DE MEMORIA COMO PARTE DE UN SISTEMA DIGITAL:

La estructura típica para un sistema digital de procesamiento de información puede establecerse como se muestra en la siguiente figura:



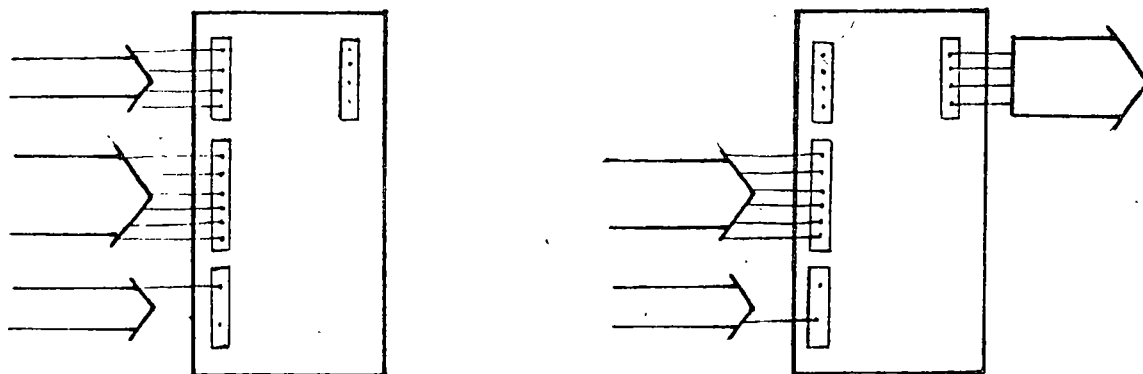
La unidad de control y procesamiento es el cerebro del sistema. Ella determina la secuencia a seguir, efectúa operaciones aritméticas y lógicas sobre variables que provienen tanto del mundo exterior (a través del bloque E/S), como del bloque de memoria. Dicha unidad puede constituirse ya sea por una unidad de procesamiento de propósito general (CPU) o bien por un bloque de lógica de propósito específico.

El bloque de E/S (Entrada/Salida), se encarga de sincronizar y/o de adecuar eléctricamente los parámetros que entran y salen del sistema. La operación del bloque se realiza bajo el control de la unidad de control y procesamiento.

El bloque de memoria contiene información que de una u otra forma es utilizada por el sistema. Desde el punto de vista de su ingerencia en el procesamiento, la información contenida en memoria puede ser de tres tipos diferentes: Programa, constantes de procesamiento y resultados parciales. El primer tipo de

información es el que indica a la unidad de control y procesamiento la secuencia a seguir en la ejecución del proceso. Esto se logra por medio de instrucciones que son leídas y traducidas (Decodificadas) por el procesador, y cuyo contenido indica a éste la operación a realizar. Las constantes de procesamiento son herramientas empleadas por el procesador para operar en situaciones específicas como toma de decisiones por comparación, operaciones aritméticas y lógicas con constantes, etc. Como su nombre lo indica la información contenida en memoria en la forma de Resultados Parciales es aquella que resulta de una etapa del procesamiento y que será requerida en etapas posteriores del mismo.

Aunque existen memorias semiconductoras de otro tipo, las cuales se analizarán posteriormente, la característica principal de las memorias descritas anteriormente es la de ser direccionables aleatoriamente, lo cual quiere decir que cada localidad de almacenamiento tiene una "dirección" única la cual puede ser accesada directamente. De aquí que los elementos involucrados en la operación de una memoria se clasifiquen como: Líneas de Datos, Líneas de Dirección y Líneas de Control. La siguiente figura representa tanto el almacenamiento como la obtención de datos de una memoria.



Las líneas de datos contienen un conjunto de niveles lógicos binarios (unos o ceros) el cual contiene la información a almacenar.

Las líneas de dirección contienen un conjunto de niveles lógicos binarios, el cual presenta a la memoria la localidad (dirección) en la cual se ha de almacenar el contenido de las líneas de datos, o bien, la dirección a partir de la cual se pretende obtener su contenido. Las líneas de control determinan la naturaleza de la operación que se desea realizar (lectura o escritura).

Si bien, el esquema presentado anteriormente puede considerarse general, existen dos tipos principales de memorias de acceso aleatorio:

- 1) RAM (Random Access Memory)
- 2) ROM (Read Only Memory)

Ram: Este tipo de memorias son conocidas también como "memorias de lectura y escritura" y el esquema anterior aplica directamente. Una característica importante en RAM es la de ser "volátiles" esto implica que al desenergizar el sistema, pierden la información almacenada.

Rom: Como su nombre lo indica (Memorias sólo de lectura), son elementos pre-programados generalmente en medios ajenos a aquellos en los cuales van a operar, de tal forma que la unidad de control y procesamiento es capaz solamente de leer su contenido, más no de alterarlo.

#### IV. RAM:

Como se estableció anteriormente, este tipo de memorias poseen las siguientes características:

- a) Son memorias de acceso aleatorio, esto es cada localidad de almacenamiento tiene una única dirección la cual puede ser accesada directamente.
- b) Son memorias de lectura y escritura, esto es, su contenido puede ser leído y alterado por el mismo sistema en operación normal.
- c) Sin elementos volátiles.



### Estructura Básica:

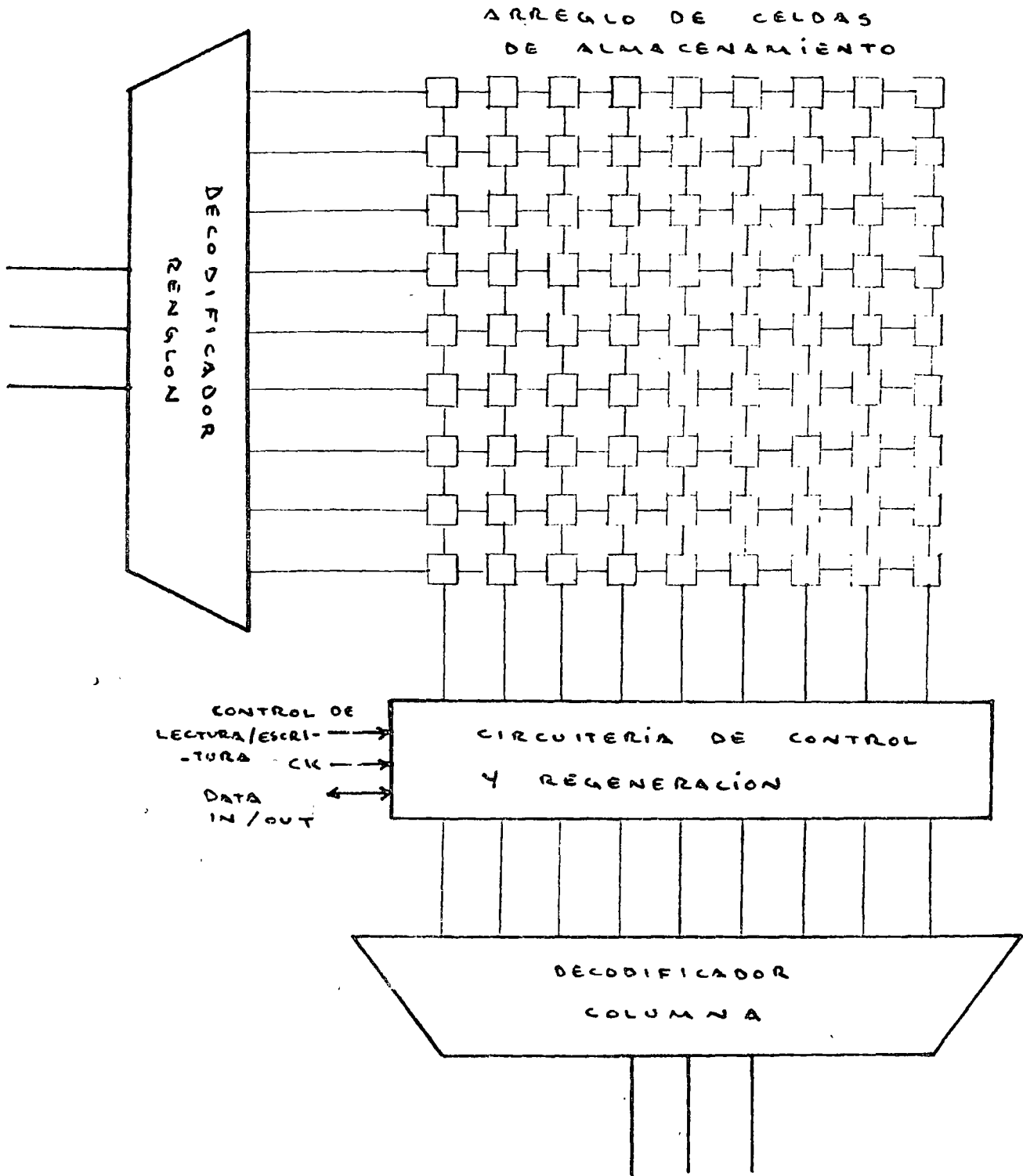
Las memorias de lectura/escritura se fabrican en la forma de un elemento (chip) con un alto nivel de integración de componentes electrónicos capaces de almacenar miles de bits\* en un área menor de medio centímetro por lado. El chip contiene un número definido de circuitos individuales, cada uno de los cuales almacena un bit. Dichos circuitos están organizados en un arreglo rectangular. El acceso a la localidad de uno de estos bits se logra proporcionando al chip una "dirección" codificada en binario la cual es procesada por los decodificadores de dirección (que son parte integral del chip), los cuales a su vez seleccionan una columna y un renglón en el arreglo de memoria. Solamente el elemento de almacenamiento ubicado en la intersección de la columna y el renglón seleccionados, estará en comunicación con el mundo exterior ya sea para leer su contenido o bien para alterarlo por medio de una operación de escritura. La operación de Lectura o Escritura se define por el estado de una línea de control, la cual es proporcionada al chip en forma independiente a las líneas de dirección. El arreglo de memoria puede ser diseñado ya sea con una sola línea de entrada/salida para la transferencia de datos, o bien, con varias líneas paralelas para transferir simultáneamente bloques (palabras) de 4, 8, o más bits. La siguiente figura muestra un arreglo de memoria de 8 columnas por 8 renglones para el almacenamiento de 64 bits; con una sola línea para transferencia de datos.

### Ram Dinámica:

Los elementos de almacenamiento en el arreglo rectangular, reciben el nombre genérico de "celdas". Dichas celdas se estructuran de muy diversas formas y se catalogan por el número de transistores (bipolares o MOS) que emplean. El concepto de Memoria Dinámica estrba en el hecho de que estos dispositivos retienen la información almacenada por un período

---

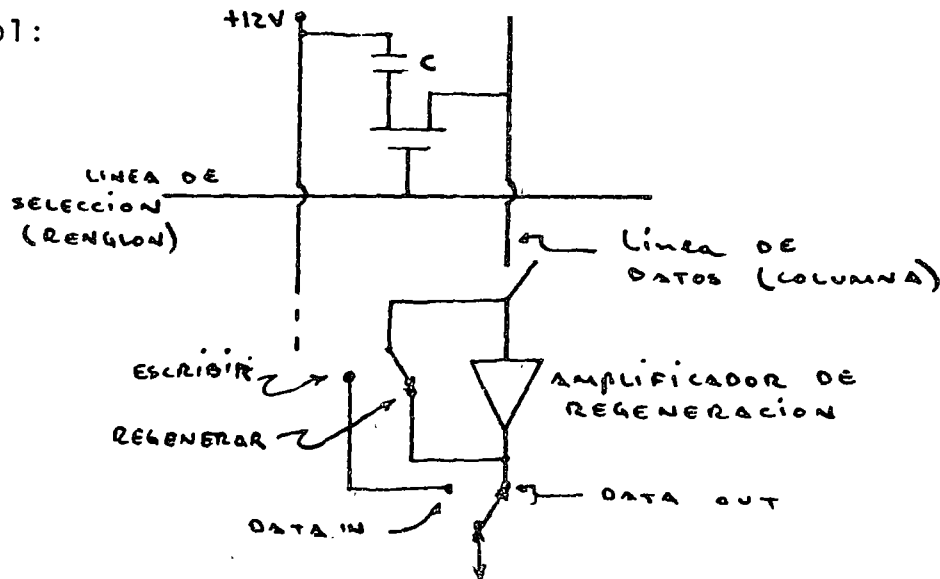
\* Bit: Dígito binario que puede adquirir uno de dos valores posibles (1 ó 0).



de tiempo muy corto, es por esto que se requiere continuamente

"recordar" a la memoria su contenido por medio de un ciclo conocido como refrescamiento. Esta inconveniencia se vé compensada por la simplicidad de fabricación la cual se realiza en base a celdas cuya estructura es relativamente sencilla todo lo cual permite alcanzar densidades de almacenamiento de información elevadas.

Las memorias dinámicas más económicas son construidas en base a celdas de un solo transistor (MOS). La siguiente figura muestra un esquema simplificado de una celda y de sus elementos de control:



La información se almacena como una carga eléctrica en el capacitor "C". Para la representación de un dígito binario, se requieren dos niveles de carga definidos los cuales podrán ser: ausencia de carga para un estado, y una cantidad de carga almacenada para el opuesto.

El transistor en la celda actúa como switch "on-off" para conectar el capacitor a la línea de datos. Dicha línea es compartida por muchas celdas idénticas, pero solo una de ellas estará activa a la vez. La línea de datos corresponde a una columna en el arreglo rectangular de memoria. Solamente una de las líneas de selección, correspondientes en el arreglo, estará activa a la vez, encendiendo todas los transistores conectados a ella, pero solo uno de ellos estará en comunicación con la línea de datos.

El capacitor de almacenamiento puede perder información de dos maneras:

- 1) Fugas inherentes al capacitor.
- 2) En cada operación de lectura, la carga almacenada en la celda se comparte con la capacitancia de la línea de datos.

Por fuga, la carga almacenada puede perderse en un tiempo del orden de mili segundos. Por otro lado, en cada operación de lectura, el nivel de carga almacenada se atenúa típicamente por un factor de 10 ó 20 dependiendo de la capacitancia en la línea de datos.

La información almacenada, entonces, deberá ser regenerada:

- 1) Inmediatamente después de cada ciclo de lectura.
- 2) Periódicamente para compensar fugas, típicamente cada dos milisegundos.

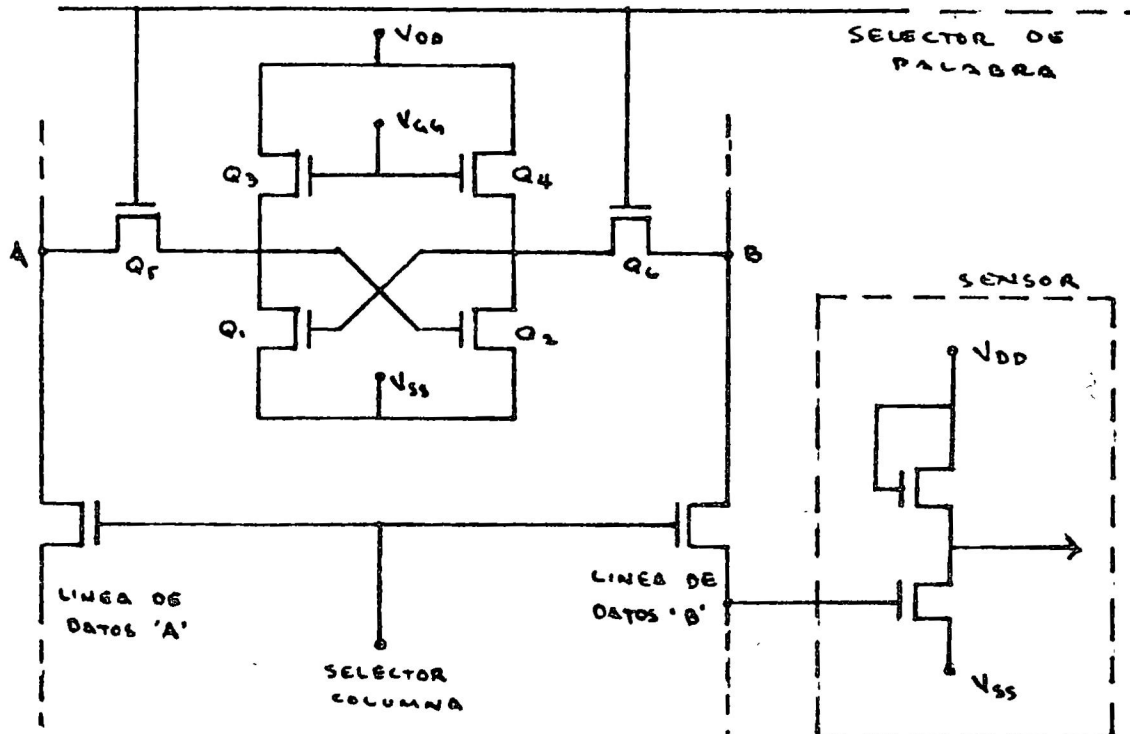
El elemento regenerador es un amplificador de umbral el cual compara el nivel de voltaje en la celda contra un nivel fijo situado al valor intermedio de los voltajes nominales para el cero y el uno, regenerando, por realimentación, el valor nominal en la celda. Existe un amplificador de umbral por cada línea de datos (columna), el cual es compartido en tiempo por todas las celdas que componen dicha columna.

Ram Estática: Aquellas memorias que no requieren refrescamiento se conocen como memorias estáticas. Son empleadas frecuentemente en sistemas que requieren relativamente poca capacidad de memoria. Esto se debe a que no requieren la circuitería externa que las memorias dinámicas emplean en el control de refrescamiento.

El concepto de memoria estática radica en el empleo de elementos biestables (flip-flops) como celdas de almacenamiento. Esto implica una mayor complejidad de fabricación aunada a una menor capacidad de almacenamiento por chip. Por lo tanto, tanto costo como capacidad de almacenamiento por chip son las cuotas que se pagan al emplear este tipo de memorias que son operativamente más simples.

En vista de que el elemento biestable puede obtenerse tanto con tecnología bipolar como con tecnología MOS, existe una gran diversidad de tipos de RAM estática dependiendo de la tecnología y de las diferentes estructuras de las celdas en cada caso. Como ejemplo, la siguiente figura muestra la estructura de una celda de seis transistores construida en tecnología MOS.

$Q_1$  y  $Q_2$  se emplean como los elementos de almacenamiento de la celda,  $Q_3$  y  $Q_4$  son utilizados como resistencias de carga. Los transistores  $Q_5$  y  $Q_6$  conectan o aíslan la celda de las líneas de datos (A y B) siendo controladas



a su vez por la línea de selección de palabra. En el modo "almacenamiento", la celda mantiene uno de sus dos estados estables y está aislada de las líneas de datos mientras la línea de selección de palabra mantenga apagados a  $Q_5$  y  $Q_6$ . Para modificar la información almacenada (escribir), una vez se selecciona la columna, la línea de selección de palabra encenderá a  $Q_5$  y  $Q_6$  conectando la celda a las líneas de datos. Dichas líneas de datos son a su vez comandadas por la circuitería de escritura para adquirir uno de dos posibles estados complementarios:  $A=1, B=0$  ó bien  $A=0, B=1$ . Suponiendo el primero de los dos casos,  $Q_1$  se apagará mientras que  $Q_2$  encenderá independientemente de cual haya sido el estado anterior de la celda. El ciclo de escritura termina cuando la línea de selección de palabra apaga a  $Q_5$  y  $Q_6$  quedando la celda en su nuevo estado hasta que otro ciclo de escritura lo modifique. Para obtener el contenido de la celda (leer), por lo menos una de las dos líneas de datos deberá contar con un elemento sensor capaz de determinar el estado de la celda. En este caso consiste de un inversor MOS el cual sensa el contenido de la celda cuando la línea de selección de palabra activa a  $Q_5$  y  $Q_6$ .

El hecho de poder ser fabricadas en diferentes tecnologías, presenta al diseñador una interesante perspectiva. Un diseño específico puede caracterizarse ya sea por requerir una alta velocidad de operación, o bien sea necesario consumir la menor potencia posible, o en otro caso implementar el sistema al menor costo. El hecho es que en base a lo que sería la principal demanda del diseño, siempre será factible encontrar el elemento que la satisfaga, sin embargo, dicha selección irá en detrimento de las demás características. Por ejemplo, si el diseño requiere alta velocidad, la tecnología bipolar es la que ofrece las características de mayor rapidez, sin embargo, presenta el mayor costo y consumo de potencia. Si por otro lado se requiere bajo consumo de potencia, las memorias CMOS serían la mejor opción a costa de una relativa baja velocidad a un costo relativamente alto. En cuanto a mínimo costo, si la densidad de almacenamiento lo justifica, RAM dinámica sería la opción más adecuada a costa de mediana velocidad y mediano consumo de potencia.

Independientemente de las consideraciones anteriores, las principales aplicaciones para RAM pueden considerarse dentro de sistemas de computación como:

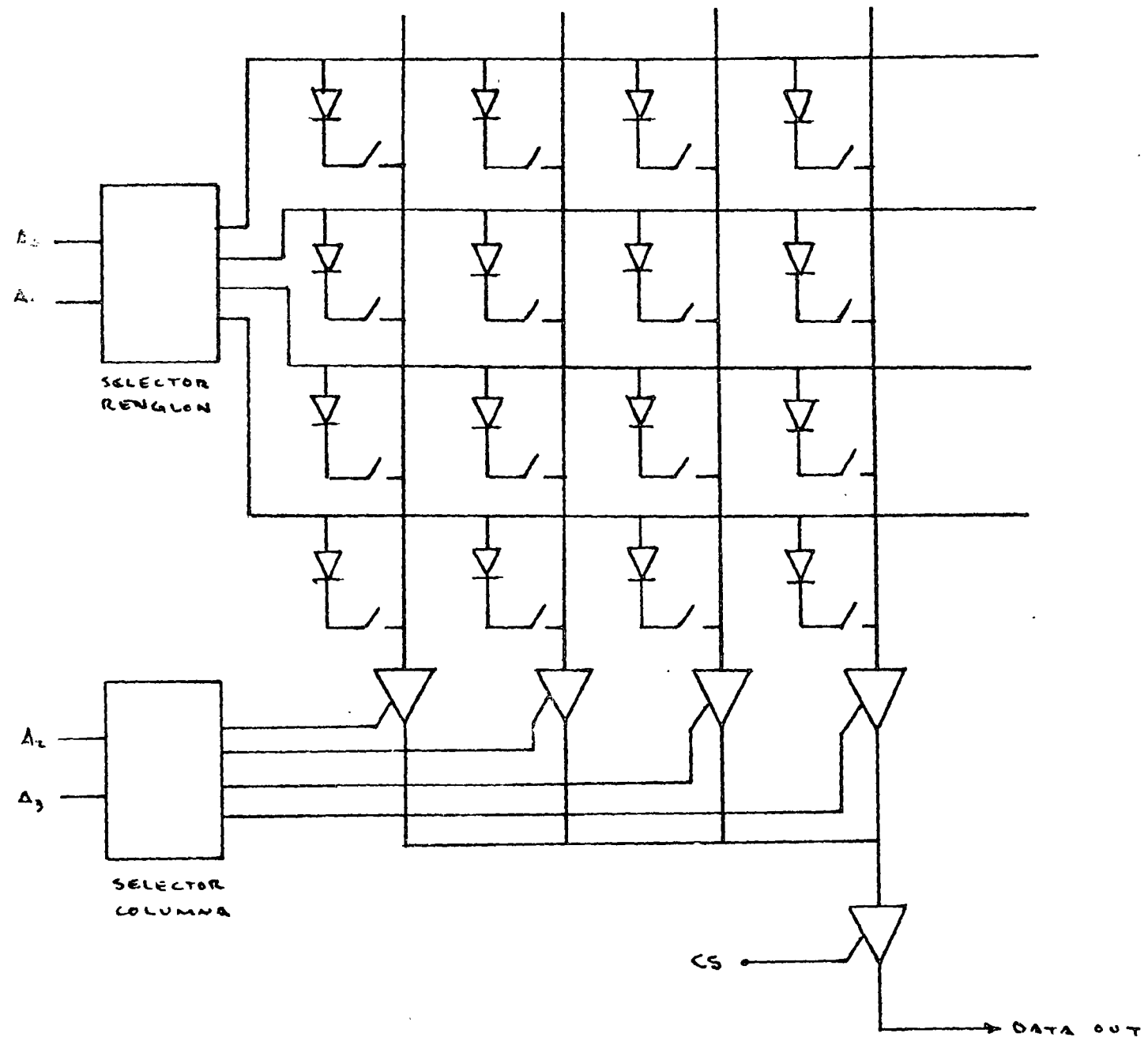
- 1) Almacén de variables del proceso: las cuales pueden ser resultados parciales que se requerirán posteriormente, datos provenientes del mundo exterior, status del procesador, stocks, etc.
- 2) Memoria de programa: las Memorias RAM se emplean como memoria de programa en los siguientes casos:
  - a) Cuando la etapa de desarrollo de un proyecto tiene lugar. RAM es un elemento esencial para desarrollar el software (programación), el cual una vez completo y probado podrá pasar a elementos de memoria de otro tipo más adecuado.
  - b) Cuando algunos programas especiales como editores o ensambladores, que por sus características residen normalmente en medios magnéticos (disco, cinta, etc.), requieren ser "cargados" al sistema de cómputo para su empleo.

## V. ROM

Algunas aplicaciones requieren memorias de acceso aleatorio cuya información almacenada sea permanente o bien raramente alterada. Es obvio que emplear RAM con estos fines implica el desperdicio de su capacidad para alterar su

contenido. Las Memorias solo de lectura (ROM) presentan la opción más adecuada para este tipo de aplicaciones. La diferencia fundamental entre ROM's y RAM's estriba en la celda de almacenamiento. Aunque la información almacenada en algunos tipos de ROM's puede ser alterada, su contenido se considera fijo durante operación manual.

La siguiente figura muestra el concepto básico de una memoria ROM:

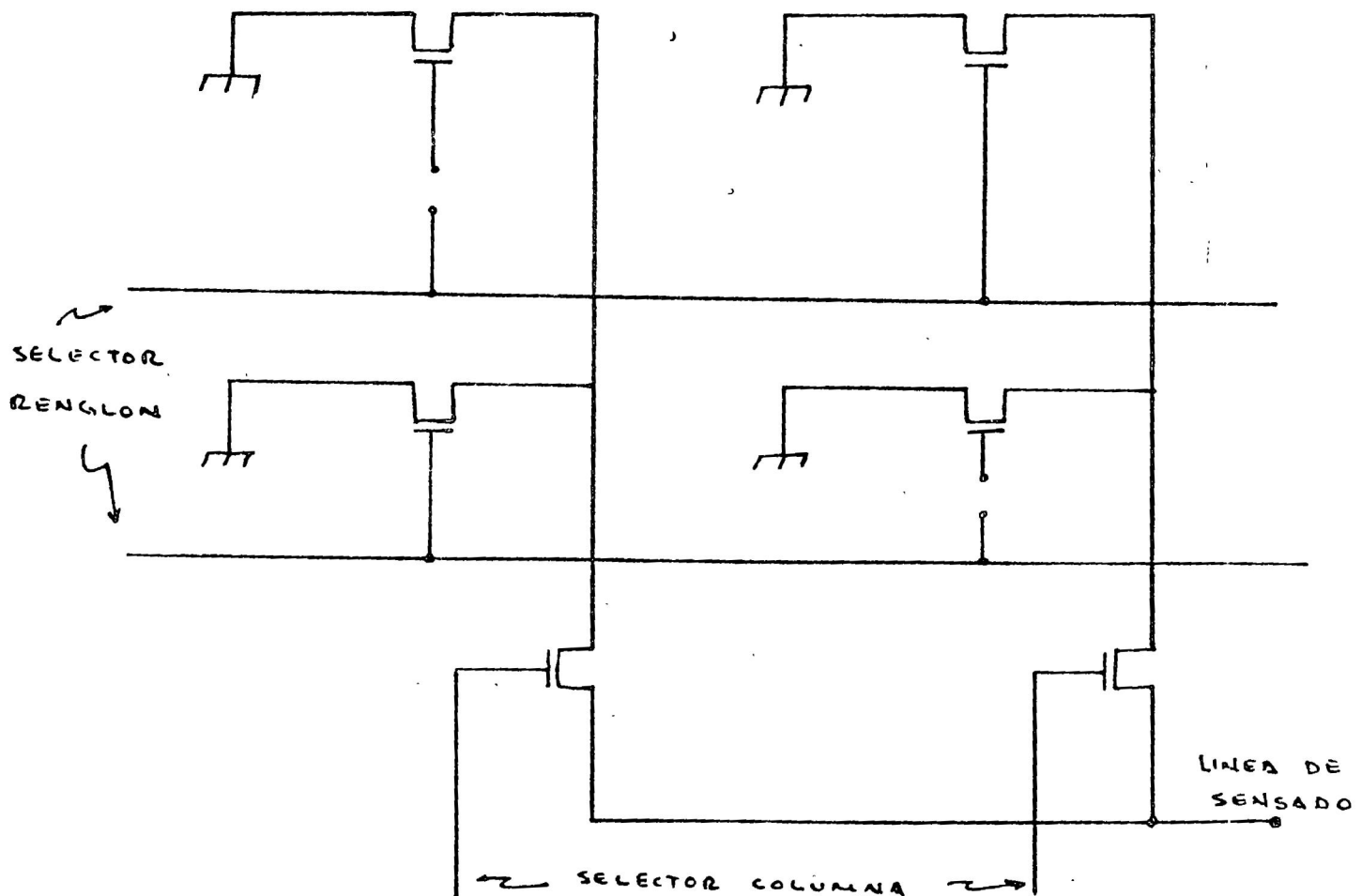


Una memoria ROM puede visualizarse como un arreglo de contactos unidireccionales selectivamente abiertos y cerrados. Las líneas de dirección  $A_0$  y  $A_1$  activan uno de cuatro renglones;  $A_2$  y  $A_3$  a su vez activan uno de cuatro amplificadores de columna de tal forma que si es CS es válido, la información de la celda seleccionada es transferida a la salida.

Atendiendo a las características de la celda, las memorias tipo ROM pueden catalogarse en:

1) ROM programable por Máscara:

El patrón de interconexiones o programa contenido por este tipo de ROM's se asigna desde el momento de su fabricación. Dicho patrón se realiza en la etapa final de la misma; esto es, al momento de asignar los contactos de aluminio a los diferentes dispositivos que componen el chip, dejando las líneas que determinan el estado de la celda conectadas o desconectadas dependiendo del patrón requerido. Obviamente, este proceso es irreversible, es decir, una vez que un patrón ha sido asignado, éste es inamovible. La siguiente figura muestra una porción de una memoria ROM programada por máscara empleando transistores MOS.



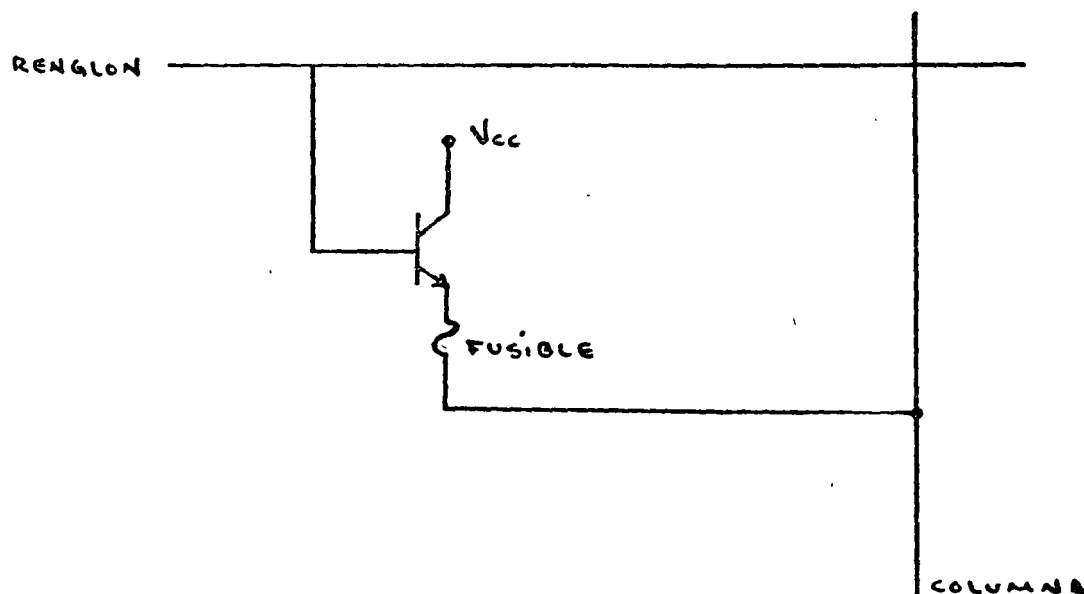


En este caso la presencia de la conexión física a la compuerta de los transistores MOS determina un "cero" o nivel bajo almacenado, y la ausencia de tal conexión determina un "uno" o nivel alto. La simplicidad de la celda (un solo transistor) permite almacenar grandes cantidades de información en forma estática por chip. Es lógico suponer que este medio de almacenamiento es económico, sin embargo hay que considerar que la producción de este tipo de memorias tiene que ser enfocada a las necesidades del usuario, por tanto, la máscara que determina el patrón requerido ha de fabricarse exprofeso para cada usuario, lo cual es costoso. De aquí que solo cuando la cantidad de memorias requeridas (con el mismo patrón) sea la suficientemente grande para justificar el costo de la máscara, esta opción podrá considerarse económica.

## 2) PROM (Memorias sólo de lectura, programables)

Existen muchas aplicaciones para ROM's en las cuales, el volumen requerido no justifica el costo de elaboración de una máscara. En estos casos, es factible operar con elementos estándares programables por el usuario. Este tipo de memorias son fabricadas en tecnología bipolar. Todas las celdas se fabrican con un elemento de material fusible, de tal forma que durante el proceso de programación (por el usuario), estos fusibles pueden ser abiertos mediante la aplicación de una corriente suficientemente alta. El patrón a almacenar queda determinado por la presencia o la ausencia de dichos elementos fusibles.

La siguiente figura muestra una celda típica de este tipo de memoria.



La programación consiste en suministrar una corriente alta a través de la juntura base-emisor en tal forma que el fusible se abra confiablemente, y por un tiempo tal que no se ocasione daño al transistor. Una vez que un patrón ha sido programado, aquellas celdas cuyo fusible ha sido abierto permanecerán en su estado definitivamente.

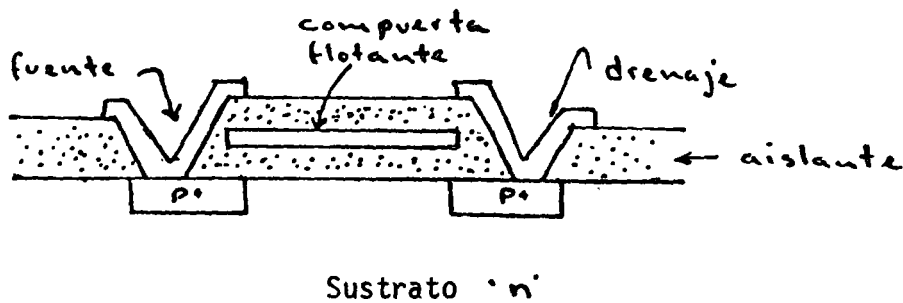
Este tipo de dispositivos son fabricados esencialmente con transistores bipolares, ya que la filosofía de operación hace imposible el empleo de tecnología MOS (las altas corrientes requeridas para fundir un fusible son incompatibles con las altas impedancias MOS). Esto implica que, como se estableció anteriormente, este tipo de memorias sean relativamente costosas y en general de menor capacidad de almacenamiento que las que emplean tecnología MOS.

### 3) EPROM (Memorias solo de lectura Reprogramables)

Los tipos de ROM's analizados previamente presentan características de definitividad en su contenido una vez que han sido programados. En algunas aplicaciones, sobre todo en etapas de desarrollo de proyectos, es de gran utilidad contar con elementos reprogramables. Es en sí esta característica la que ha hecho que los EPROM's sean quizá los elementos más populares en el campo de las memorias solo de lectura. La reprogramación se lleva a cabo "borrando" el contenido de la memoria exponiéndola a radiación de luz ultravioleta la cual sitúa la totalidad de sus celdas en el estado original, siendo posible de esta manera, introducir un nuevo patrón de información.

Este tipo de dispositivos se fabrican empleando tecnología MOS. Cada celda en el arreglo cuenta con un transistor MOS cuya característica es la de tener su compuerta flotante, esto es, completamente rodeada de dióxido de silicio (aislador) lo cual implica que esta no está conectada físicamente a nada. El estado lógico de la celda se determina por el estado de este transistor: encendido o apagado. El encendido del transistor se logra almacenando suficiente carga en la compuerta flotante, lo cual se logra aplicando un voltaje suficientemente elevado entre el sustrato y el drenaje del transistor. Esto ocasiona que algunos electrones adquieran suficiente energía para cruzar la barrera aislante, almacenándose en la compuerta flotante. Al desaparecer la excitación, dichos electrones quedan atrapados en la región de la com-

puerta ocasionando que el transistor quede encendido. Cuando no existe carga en la compuerta flotante, el transistor está apagado. La siguiente figura muestra un esquema simplificado de este tipo de transistores MOS:



Otra alternativa en memorias sólo de lectura reprogramables son las memorias del tipo EAROM (Memorias solo de lectura Electricamente Alterables). Sin estructura es muy similar a la estructura de los EPROM's, difiriendo básicamente en el empleo de una compuerta adicional, la cual en combinación con potenciales apropiados en las terminales del transistor, permite cargar y descargar la compuerta flotante empleando exclusivamente medios electrónicos.

Las principales aplicaciones para ROM's pueden establecerse en:

1) Generación de caracteres:

En el manejo de terminales del tipo CRT (tubo de rayos catódicos), este tipo de dispositivos son la mejor opción para traducir el código de transmisión de información digital en señales que manejen la deflexión del rayo en el CRT, de tal forma de reproducirlos en la forma de caracteres en la pantalla.

2) Ejecución de funciones Lógicas:

Estos elementos son empleados en la ejecución de lógica cada vez más frecuentemente ya que resulta más económico emplear un chip de alto nivel de integración (LSI) que muchos de bajo nivel (SSI). Se utilizan como la implementación directa de cualquier tabla de verdad, proveyendo resultados sin detalles lógicos intermedios.

3) Microprocesadores.

En el campo de los microprocesadores las memorias tipo

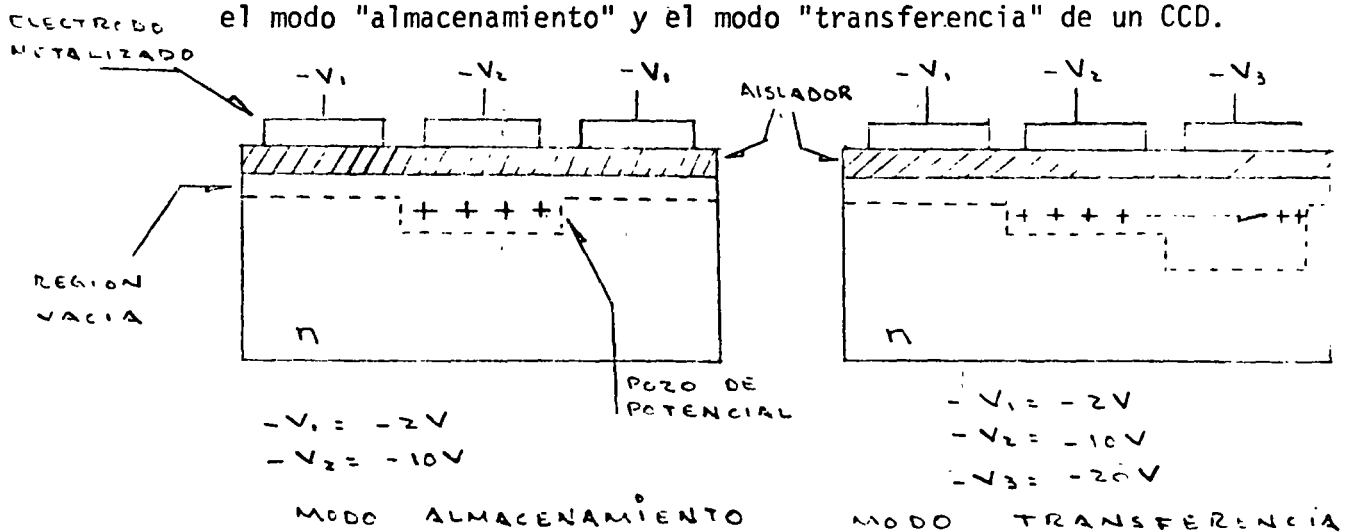
ROM son empleadas principalmente como almacen de programas fijos tales como Monitores, subrutinas de propósito general, programas de propósito específico, etc. También son empleados como tablas de referencia (Look-up tables) conteniendo la solución de algunas funciones cuya obtención por medio del procesamiento no se justifica debido ya sea a su complejidad o bien al tiempo involucrado en su ejecución.

## VI. MEMORIAS SERIE:

Hasta este punto se han descrito memorias de acceso aleatorio, esto es, en las que el tiempo requerido para una operación de lectura o escritura es independiente de la localidad física dentro del arreglo. Cuando la aplicación no demande acceso aleatorio, es factible obtener importantes ahorros en el diseño si se emplean memorias de acceso en serie en las que los bits en el arreglo circulan a través de 64 o más localidades de almacenamiento, lo cual determina el intervalo de tiempo comprendido, desde el momento en que la información se almacena hasta que está disponible para ser leída. Las tecnologías más empleadas en la elaboración de memorias serie son CCD y Burbujas Magnéticas.

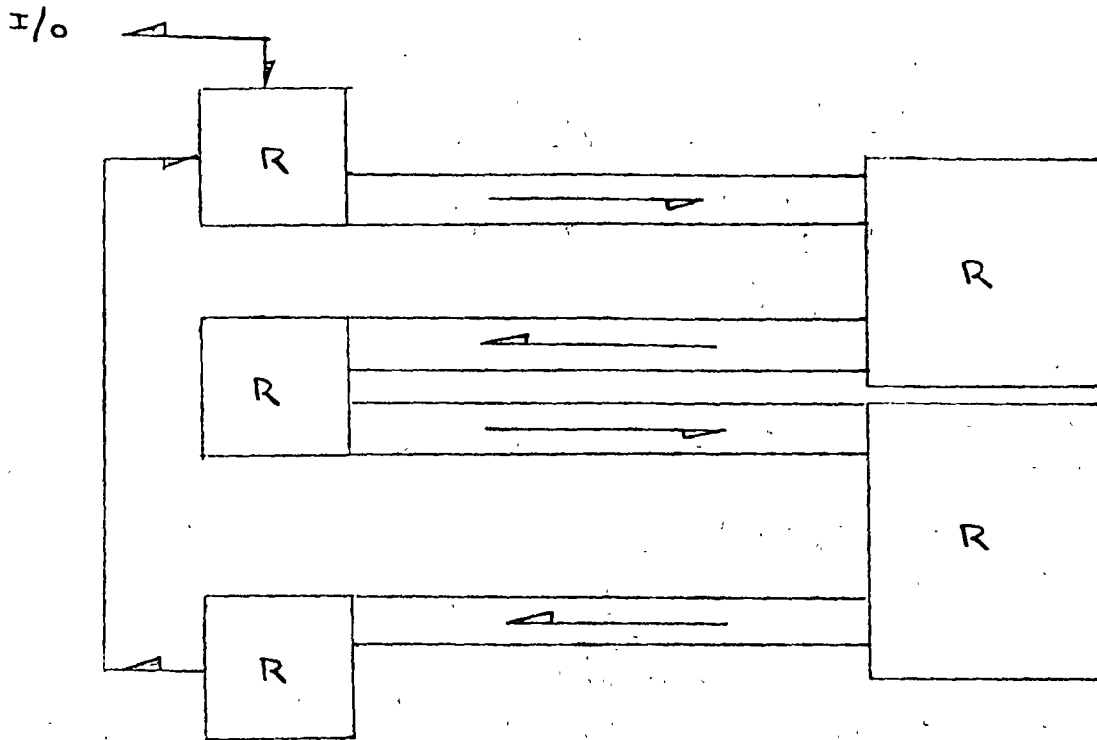
### 1) CCD (Charge Coupled Devices)

Este tipo de dispositivos operan en base a almacenamiento y transferencia de carga entre "pozos" de potencial ubicados en la superficie del silicio. Dichos "pozos" de potencial se forman en base a una serie de capacitores MOS ubicados en sucesión. La siguiente figura representa el modo "almacenamiento" y el modo "transferencia" de un CCD.



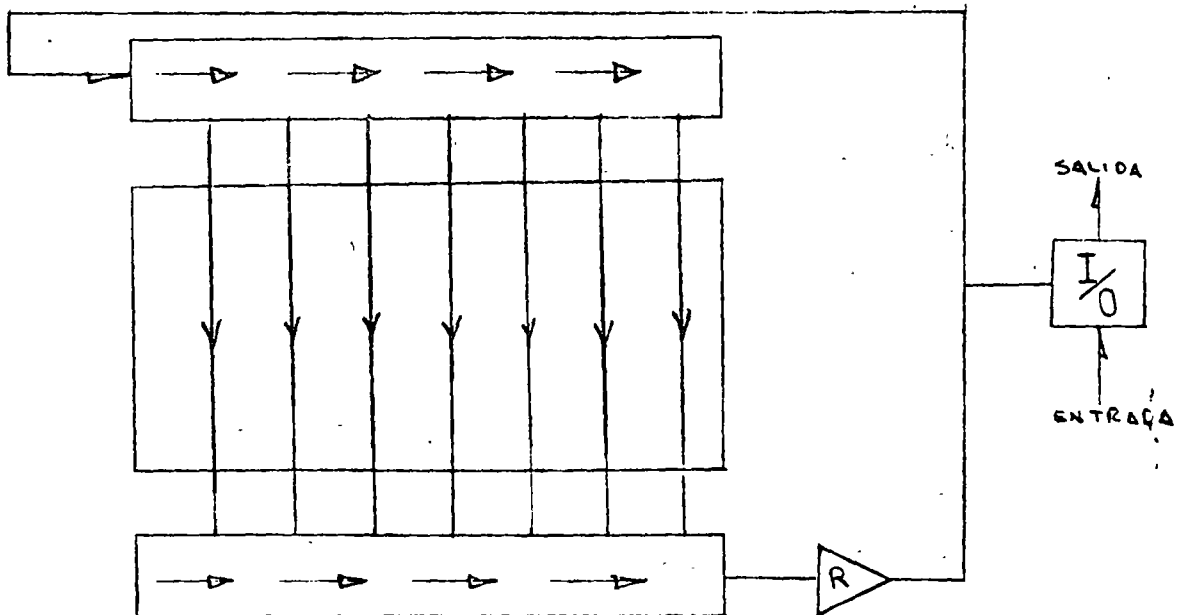
Dentro de las diferentes estructuras de memoria CCD, pueden citarse:

1) Estructura serpentina:



en esta estructura, la señal pasa a través de segmentos cortos del "circuito" total, al final de cada cual es regenerada e insertada de nuevo al siguiente segmento.

2) Estructura Serie - Paralelo - Serie



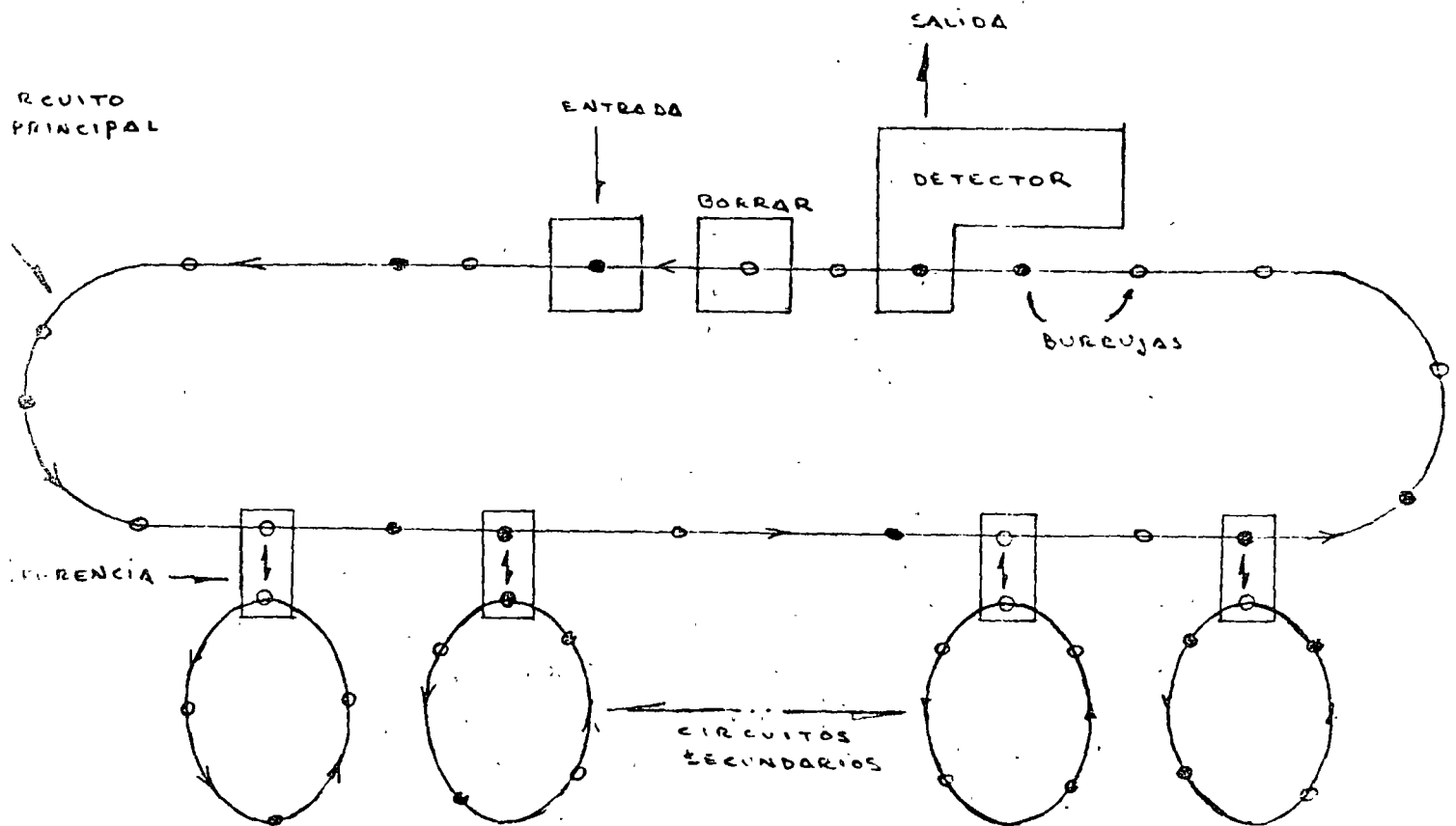
Esta estructura posee  $M$  registros paralelos, las cuales almacenan  $N$  bits cada uno, y dos registros serie, uno de entrada y otro de salida. La frecuencia aplicada a los registros serie de entrada/salida es alta en comparación a la aplicada a los  $M$  registros paralelos. La información es transferida en paralelo del registro serie de entrada a la primera etapa de los canales verticales. A la salida el registro serie es cargado en paralelo proporcionando la información a través de un circuito regenerador.

Existen muchas aplicaciones donde el acceso en serie es útil. Por ejemplo, una memoria empleada para refrescar la información presente en una terminal de video, la cual es barrida punto por punto, no requiere una memoria de acceso aleatorio. Las principales ventajas de los dispositivos CCD son: su alto nivel de integración, la complejidad en la decodificación de direcciones es mínima, los sistemas de regeneración de información son simples en vista de que la eficiencia en la transferencia de carga es muy alta. Todo esto ocasiona que el área total de silicio para elementos completos de memoria sea de dos a tres veces menor en CCD que en RAM.

## 2) Burbujas Magnéticas:

Este tipo de memorias de acceso en serie consisten en pequeñísimos dominios de polarización magnética ubicados en una película delgada de material magnético los cuales pueden moverse aplicando campos magnéticos en ángulos rectos al campo principal. Si uno de estos pequeños dominios con polarización opuesta a la de la película se designa como "uno" binario, la ausencia de tal dominio se considerará "cero" binario. La siguiente figura muestra una representación de una memoria típica de este tipo.

La principal ventaja de estos elementos (No son memorias semiconductoras), estriba en el hecho de ser no volátiles, de ahí su aplicación potencial más importante sea el reemplazo de memorias magnéticas como cintas y discos y cuya capacidad oscila entre 1 millón y 10 millones de bits.

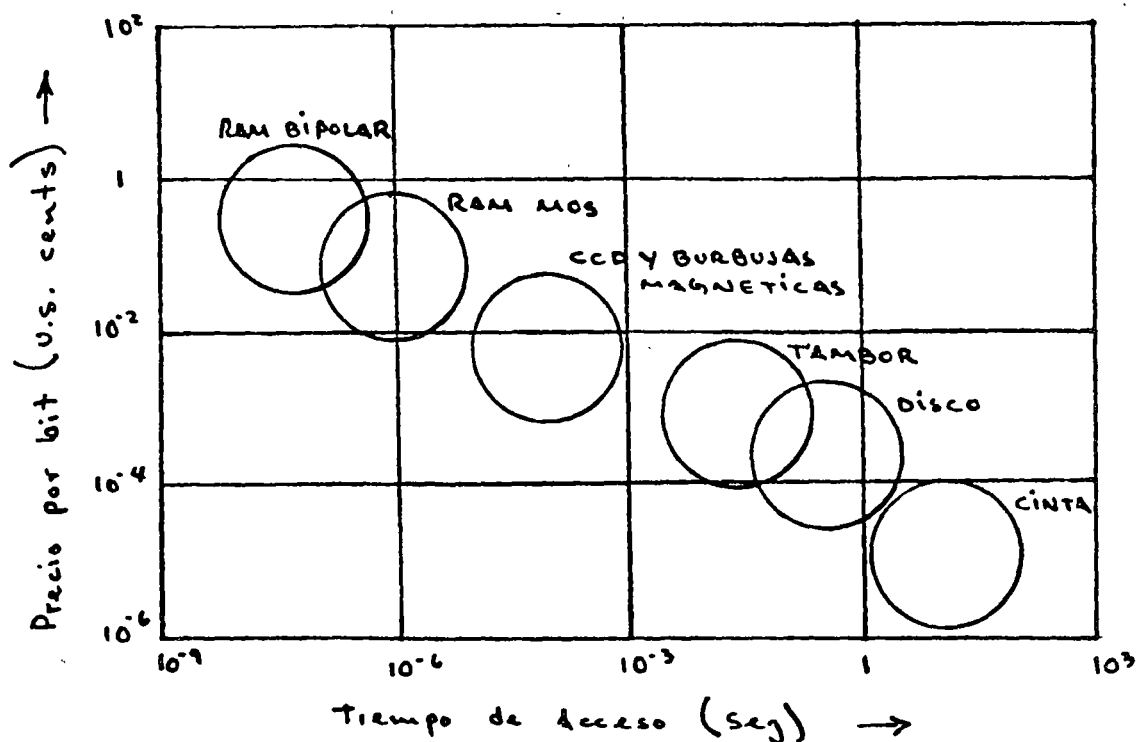


## VII. CARACTERÍSTICAS TECNOLÓGICAS.

Dentro de los factores más importantes a considerar en la selección de una tecnología específica están: velocidad (tiempo de acceso) y costo.

En vista de las diferentes capacidades de almacenamiento de los elementos actualmente en el mercado, hablar de costo por bit es quizá el medio más efectivo para obtener una relación generalizada en el aspecto de costos.

La velocidad de operación de una memoria se mide en términos de su tiempo de acceso, el cual es simplemente el tiempo involucrado en leer o escribir en una determinada localidad del arreglo. La siguiente tabla muestra la relación que existe actualmente entre tiempo de acceso y precio por bit. En general, puede observarse que CCD'S y burbujas magnéticas ocupan un espacio importante entre las memorias semiconductoras (acceso aleatorio) y las lentas memorias magnéticas. Otro detalle apreciable lo representa la



relación de incrementos en tiempo de acceso contra decremento en costo; en general, implica incrementar el tiempo de acceso en dos órdenes de magnitud para obtener un orden de magnitud de ahorro.

En muchos casos, el consumo de potencia será un factor importante a considerar. En estos casos, el compromiso se establece entre potencia versus velocidad. En general, como se ha dejado entrever a lo largo del tema, aquellos diseños que requieren altas velocidades de operación acarrearán como consecuencia un mayor consumo de potencia. Por otra parte, cuando las características del diseño lo permiten, los dispositivos que ofrecen las mayores densidades de almacenamiento por chip ofrecen mayores ventajas desde el punto de vista de ahorro de espacio y simplificación del diseño. Se estima que durante la próxima década, se obtendrán importantes mejoras en los medios de almacenamiento tanto magnéticos como semiconductores, to do esto aunado a reducciones en costo. La gran versatilidad de las memorias semiconductoras las hace fuertes favoritas a seguir dominando el mer cado de las memorias semiconductoras. El desarrollo de nuevas técnicas



en la fabricación de microcircuitos electrónicos, tales como rayos x y haces de electrones, posiblemente ocasionará que las densidades de almacenamiento se incrementan por un factor de 100, con incrementos mínimos en costo. Puede entonces establecerse que durante los próximos 10 años se producirá una caída de más de un orden de magnitud en el precio por bit en todos los tipos de memorias digitales.

## BIBLIOGRAFIA

1. Hodges D.A.: "Microelectronic Memories"; Scientific American, Sept. 1977.
2. Hnatck E.R.; "A User's Handbook of Semiconductor Memories"; Wiley Interscience, 1977.
3. Luecke G., Mize J.P., Carr W.N.; "Semiconductor Memory Design and Application"; Mc Graw-Hill, 1973.

## PROBLEMAS DE INTERCONEXION

- I. Introducción
- II. Lógica síncrona y asíncrona
- III. Condiciones de carrera; Glitches
- IV. Defasamiento de Reloj (Clock skew)
- V. Reflexiones de voltaje.

## PROBLEMAS DE INTERCONEXION

### I. INTRODUCCION

El diseño lógico tanto combinacional como secuencial se basa en una serie de establecimientos y procedimientos cuyo origen es puramente matemático. Dicha mecanización se enfoca única y exclusivamente sobre el tratamiento de las variables que serán procesadas por el sistema, sin considerar que en la implementación física entran en juego las características operativas de los elementos que lo componen. Esto quiere decir que el mecanismo formal de diseño considera inherentemente una serie de situaciones ideales entre los elementos y señales utilizadas, tales como:

- a) Niveles lógicos perfectamente definidos.
- b) Respuesta instantánea de los elementos.
- c) Capacidad ilimitada de manejo entre elementos.

Es por esto que el diseñador debe estar familiarizado con una serie de conceptos que definen el comportamiento real de los diferentes elementos que formarán parte del sistema. Dichos conceptos pueden separarse en dos grupos principales atendiendo a sus características:

- 1) Compatibilidad eléctrica
- 2) Tiempos de respuesta

#### 1) Compatibilidad eléctrica:

El concepto de compatibilidad eléctrica está directamente relacionado con las diferentes familias lógicas que se emplean. Es este quizá el concepto más claro en virtud de la cantidad de información existente y de lo determinístico de sus características. Es siempre recomendable emplear elementos de una sola familia lógica lo cual brinda al diseñador la simplicidad de interconexionado directo, sin embargo, en algunos casos, la característica del diseño requiere del empleo de elementos de diferentes familias lógicas. En tales casos, el interconexionado rara vez será directo, y el diseñador deberá conocer las características de entrada y salida de dichos elementos de tal forma que el interfaceo sea el adecuado.

## 2) Tiempos de Respuesta.

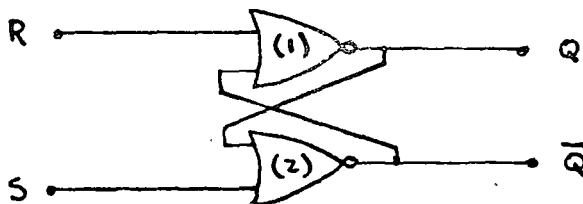
El tiempo involucrado desde el momento de presentarse la excitación hasta que el elemento responde, es llamado tiempo de respuesta o de propagación. Las variaciones de este parámetro en los diferentes elementos que componen un sistema digital son causantes de infinidad de problemas en los casos en que el diseño no es adecuado. La mayor parte de estos problemas provienen de una operación marginal de los elementos, lo cual es típicamente el resultado de un diseño sin la adecuada consideración en el aspecto de tiempos.

Lo realmente delicado de este tipo de problemas estriba en el hecho de que algunas veces no se detectan en el prototipo inicial, y debido a las diferencias (normales) que existen en las características de los componentes, el problema puede presentarse una vez que el producto se encuentra en producción, o bien cuando el producto se somete a condiciones de operación diferentes como variaciones de temperatura, etc.

## II. LOGICA SINCRONA Y ASINCRONA.

En el diseño lógico, el empleo de elementos de almacenamiento (flip-flops) es de gran utilidad, sin embargo, sus características de operación las hacen aparecer como los elementos que mayor número de problemas acarrearán si no se toman las debidas precauciones en el diseño inicial.

Considérese al flip-flop R-S mostrado en la siguiente figura:



Supóngase que  $Q = 1$  y  $\bar{Q} = 0$ , y que las entradas R y S estaban situadas en cero. Si en un momento dado, la entrada R adquiere el valor de uno, el flip-flop alcanzará un estado de "reset" ( $Q = 0$ ,  $\bar{Q} = 1$ ). Sin embargo, la transición del estado original ( $Q = 1$ ,  $\bar{Q} = 0$ ) al estado de "reset" se efectúa en dos etapas:

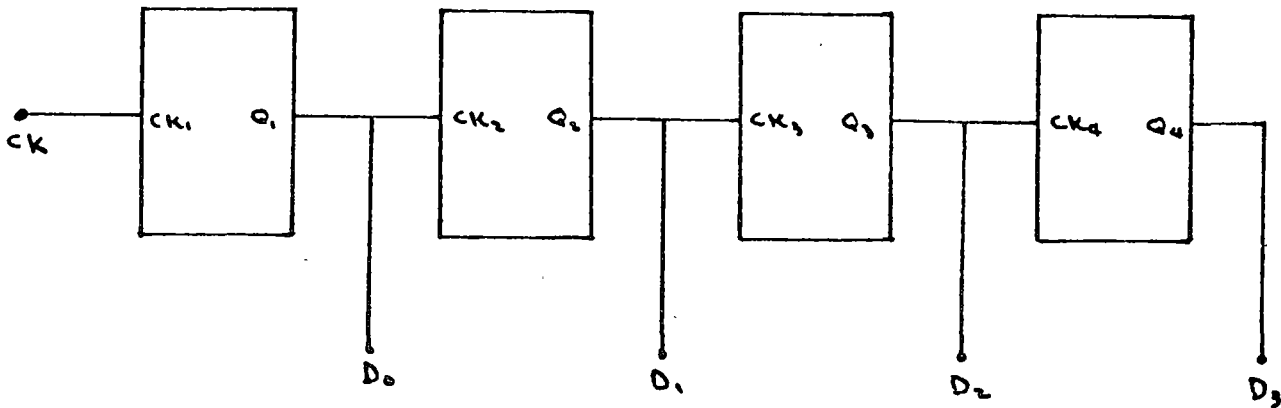
- a) En el momento en que  $R=1$ , la compuerta (1) comienza a reaccionar hasta que después de su tiempo de respuesta característico, sitúa su salida  $Q = 0$ .
- b) Una vez que  $Q$  se sitúa en cero, la compuerta (2) comienza a reaccionar hasta que después de su tiempo de respuesta característico, sitúa su salida  $\bar{Q} = 1$ .

Esto quiere decir que si el estado original fue  $Q = 1$ ,  $\bar{Q} = 0$ , para llegar al estado final donde  $Q = 0$ ,  $\bar{Q} = 1$ , es necesario pasar por un estado intermedio donde  $Q = 0$ ,  $\bar{Q} = 0$  el cual tiene una duración igual al tiempo de retraso de la compuerta (2). Es obvio que dicho estado intermedio es un estado indeseable el cual, dependiendo de las funciones asignadas a  $Q$  y  $\bar{Q}$ , puede ocasionar serios trastornos al diseño.

Considerando que la característica de respuesta descrita anteriormente se aplica a todos los tipos de flip-flops, se ha desarrollado el concepto de lógica síncrona el cual elimina el peligro potencial producido por los cambios de estado de los mismos.

La presencia de una entrada de reloj permite al flip-flop cambiar de estado solo cuando la señal de reloj es válida. En esta situación, el flip flop estará en condición de muestrear el estado de sus líneas de entrada sólo cuando el pulso de reloj está presente, produciéndose en ese momento el cambio de estado. Ahora bien, si dicho pulso de reloj sincroniza a todos los flip-flops del sistema, es posible eliminar los estados intermedios si el intervalo de tiempo entre pulsos de reloj es lo suficientemente grande para permitir que los flip-flops se establezcan.

Extendiendo aun más el concepto descrito anteriormente, considérese la operación de un contador asíncrono de 4 bits el cual se muestra en la siguiente figura:



supóngase que el estado del contador es  $D_0 = 1$ ,  $D_1 = 1$ ,  $D_2 = 1$ ,  $D_3 = 0$ , y se presenta un pulso en CK. La tabla de estados ideal mostraría:

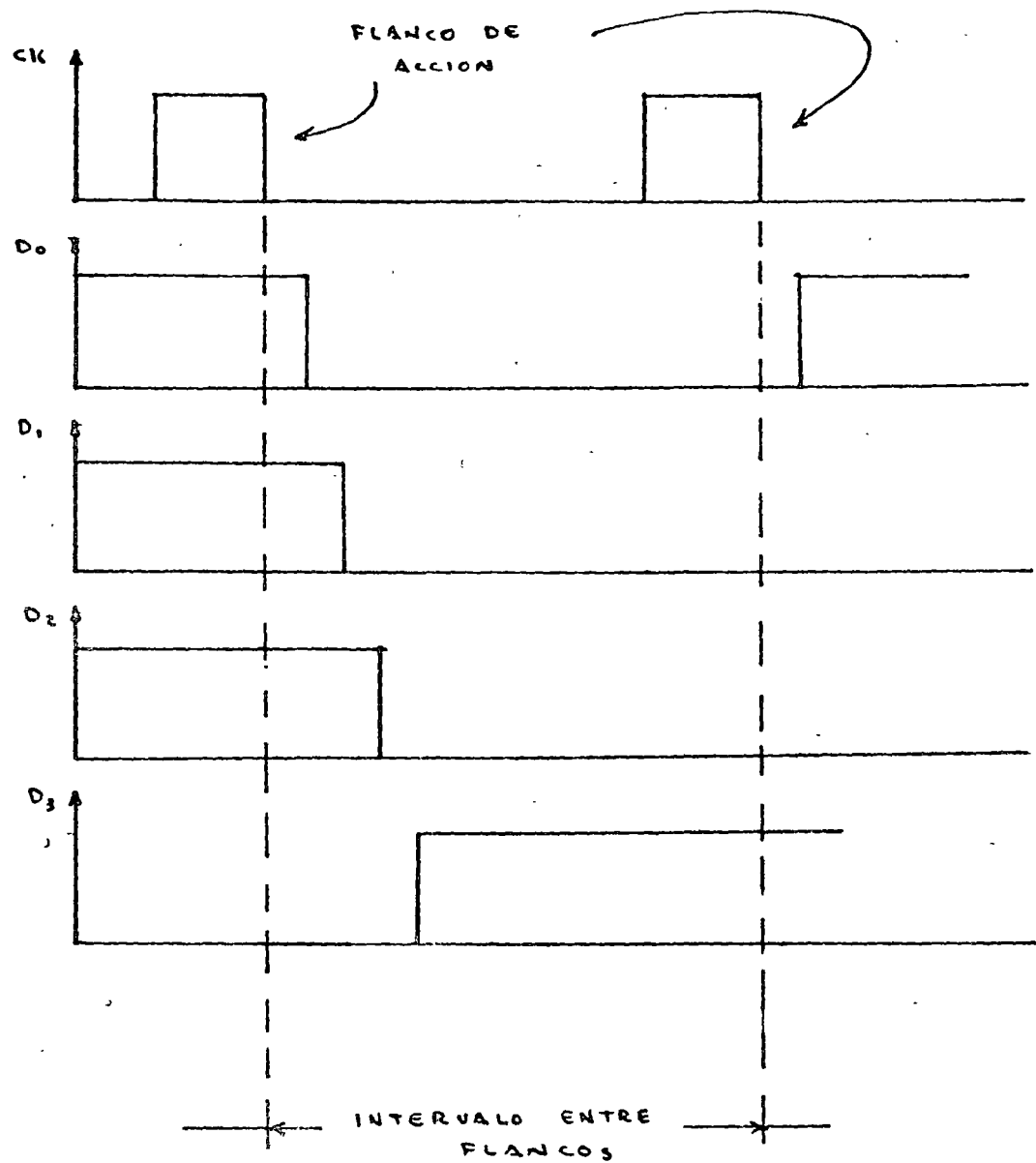
| <u><math>D_3</math></u> | <u><math>D_2</math></u> | <u><math>D_1</math></u> | <u><math>D_0</math></u> |                  |
|-------------------------|-------------------------|-------------------------|-------------------------|------------------|
| 0                       | 1                       | 1                       | 1                       | ← estado inicial |
| 1                       | 0                       | 0                       | 0                       | ← estado final   |

Sin embargo, el mecanismo de cambio se origina en el flip-flop menos significativo hasta llegar al más significativo de la siguiente manera:

| <u><math>D_3</math></u> | <u><math>D_2</math></u> | <u><math>D_1</math></u> | <u><math>D_0</math></u> |                  |
|-------------------------|-------------------------|-------------------------|-------------------------|------------------|
| 0                       | 1                       | 1                       | 1                       | ← estado inicial |
| 0                       | 1                       | 1                       | 0                       | } transición     |
| 0                       | 1                       | 0                       | 0                       |                  |
| 0                       | 0                       | 0                       | 0                       |                  |
| 1                       | 0                       | 0                       | 0                       | ← estado final   |

donde cada transición tiene una duración igual al tiempo de respuesta característico del flip-flop en turno.

Un diagrama de tiempos del sistema mostraría:



Si las líneas  $D_0 - D_3$  se emplean para manejar de alguna forma a otros elementos biestables, y suponiendo que la totalidad de los

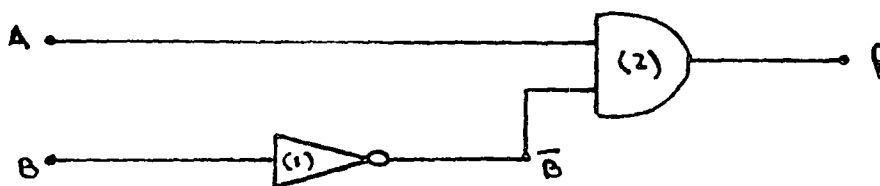


flip-flops en el sistema están sincronizados con el flanco de bajada de la línea "CK", la presencia de dicho flanco será el único instante válido durante el cual los estados de las variables de entrada de los flip-flops serán muestreadas, y no será sino hasta el siguiente flanco de bajada cuando se produzcan el siguiente muestreo. Esto significa que durante el intervalo de tiempo comprendido entre flancos de bajada, el estado de las líneas de salida de los flip-flops no afecta el funcionamiento del sistema, dando tiempo a que dichas líneas de salida alcancen un estado estable válido, el cual será muestreado en el siguiente flanco de bajada.

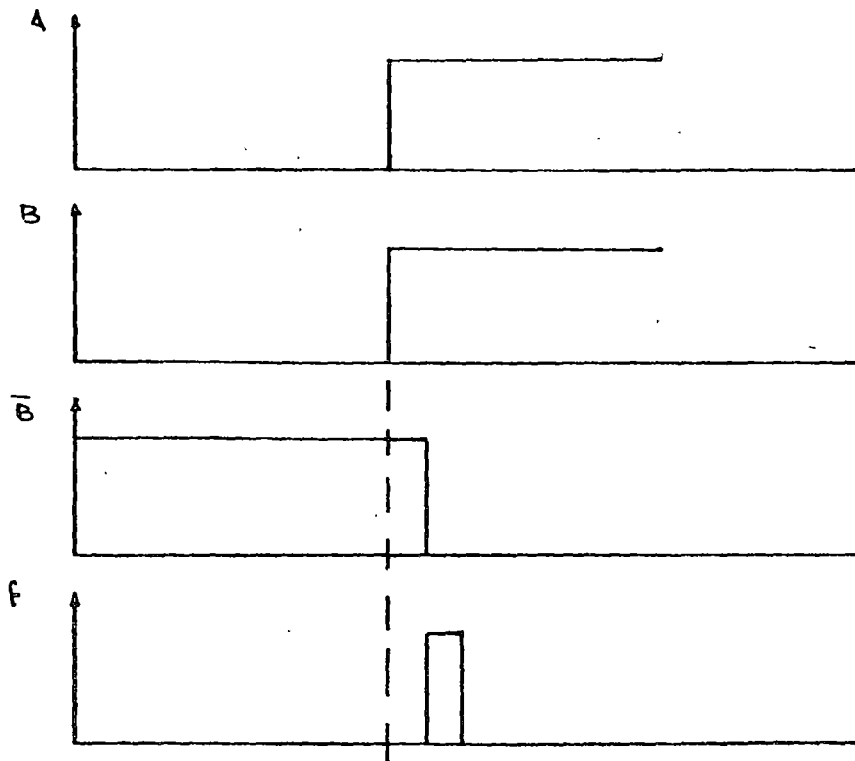
El empleo de lógica síncrona permite, además, inmunizar al sistema contra cambios momentáneos en el estado de las líneas producidos por agentes externos al funcionamiento normal del mismo, tales como ruido.

### III. CONDICIONES DE CARRERA; GLITCHES

Un "glitch" se produce cuando el estado de una línea cambia durante un intervalo de tiempo muy corto, volviendo finalmente a su estado inicial. Dichos glitches o pequeños pulsos son generalmente elementos indeseables dentro de cualquier sistema, ya que su presencia se debe a problemas de tiempos de respuesta ocasionados por el interconexión de elementos en la implementación física de un diseño. Considérese, por ejemplo, que se desea implementar la función lógica  $f = A \cdot \bar{B}$ . Una realización simple se muestra en la siguiente figura:

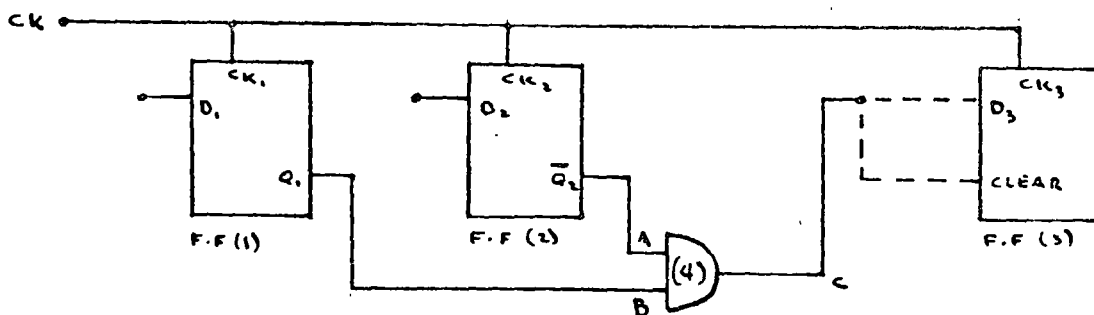


El circuito mostrado realiza efectivamente la función deseada, sin embargo, considérese que las variables de entrada presentan en un determinado momento la característica mostrada en el siguiente diagrama:

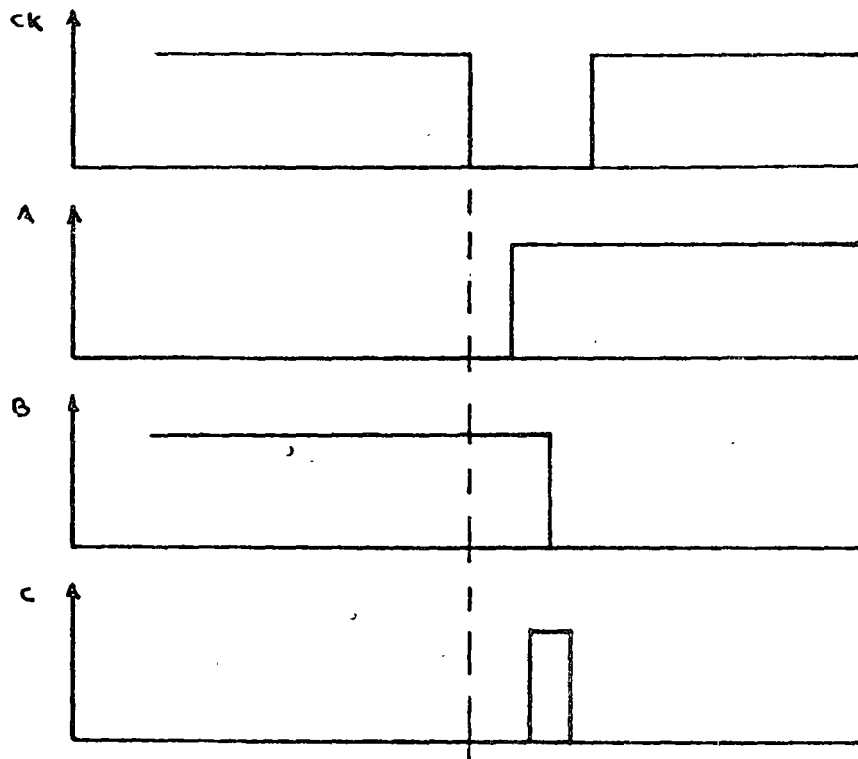


En el instante en que A y B adquieren el valor de "uno", las entradas a la compuerta (2) son verdaderas, ya que  $\bar{B}$  no cambia sino hasta que la compuerta (1) reacciona. Esta situación ocasiona que la compuerta (2) comience a reaccionar tratando de producir una salida igual a "uno". Si el problema se agudiza considerando que la compuerta (1) es más lenta que la compuerta (2), la salida de (2) valdrá "uno" antes de que  $\bar{B}$  valga "cero". Cuando  $\bar{B} = 0$ , la compuerta (2) comenzará a reaccionar hasta que después de su tiempo de respuesta su salida vuelva a "cero". Todo esto produce un pequeño pulso o "glitch" el cual es una condición inválida dentro del diseño.

Considérese el siguiente circuito:



Supóngase que se desea obtener una señal en "C" válida solo cuando A y B sean válidas ( $C = A \cdot B$ ). Considérese que los flip flops (1) y (2) están situados en el estado  $Q_1 = 0$ ,  $\bar{Q}_2 = 1$ , además tanto  $D_1$  como  $D_2$  están en "uno". El siguiente diagrama muestra el efecto producido por la llegada de un pulso de reloj en esas condiciones:



En este caso la presencia del "glitch" en C se debe a la diferencia en tiempo que toma a las salidas  $Q_1$  y  $\bar{Q}_2$  establecerse. Si la línea "C" se conecta a la entrada  $D_3$  de F·F (3), la presencia de dicho glitch no afectaría en nada la operación del circuito ya que esto se produce entre los flancos de bajada de los pulsos de reloj. Sin embargo, la mayoría de los flip-flops constan de entradas asíncronas tales como "preset" y "clear" las cuales son de gran utilidad en el diseño. Considérese que la línea "C"

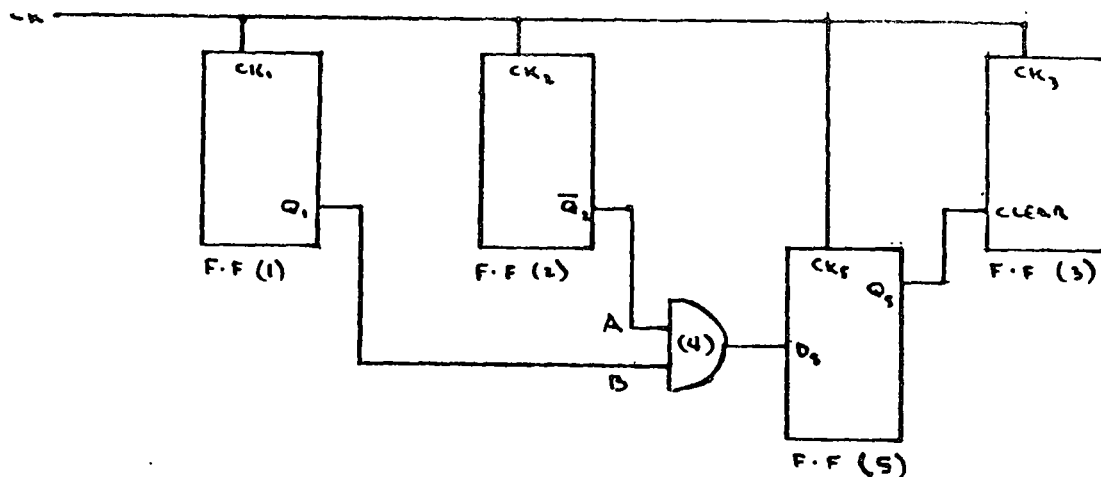
del circuito anterior se conecta a la entrada "clear" del F·F (3). En este caso, la presencia del "glitch" en "C" es de extremo peligro ya que eventualmente es capaz de resetear al F·F (3) en una condición evidentemente falsa.

La situación descrita anteriormente establece una condición de carrera. La cual se denomina de esta manera en virtud de que presenta una "competencia" entre la duración del "glitch" y la velocidad de respuesta del elemento afectado. El glitch puede no ser lo suficientemente grande en tiempo como para resetear a F·F (3), sin embargo, esto podría suceder si el F·F (3) se cambia por otro (del mismo tipo) cuya velocidad de respuesta, aun dentro de especificación, sea mayor, o bien, el efecto de reseteo puede presentarse al variar la temperatura de operación, etc.

Es un hecho entonces que aún cuando este tipo de entradas asín cronas (clear y Preset) son de gran utilidad en el diseño, su utilización sin la debida precaución puede acarrear problemas sustanciales al producto.

En vista de lo anterior es siempre recomendable empeler lógica síncrona en todas las etapas del sistema, sin embargo, situaciones como la descrita anteriormente pueden presentarse. En estos casos, tales situaciones de "peligro" deberán identificarse desde el diseño original mediante un análisis concienzudo del circuito inicial propuesto. Una vez detectada la situación problemática, es factible de resolver ya sea empleando elementos de retardo situados en lugares estratégicos, o bien empleando lógica adicional que sincronice la operación de tales elementos. La siguiente figura muestra un método posible para eliminar el problema presentado en el circuito anterior.

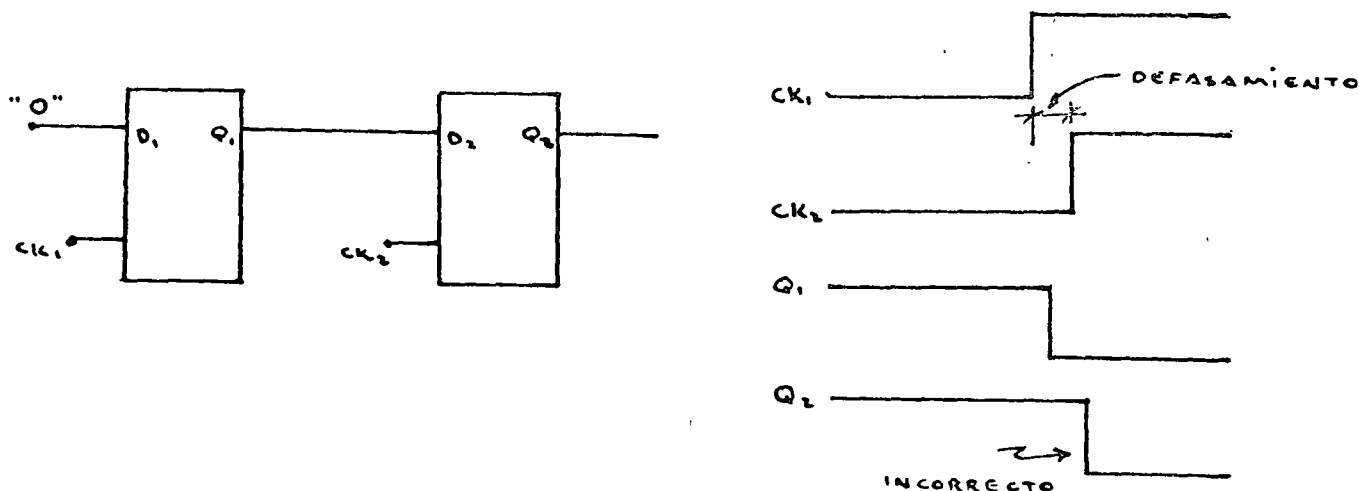
La presencia de F·F (5) permite eliminar el peligro potencial presentado por el "glitch" ya que cuando éste se presente F·F (5) no estará activado por la línea de reloj y F·F (3) tendrá que esperar hasta el siguiente pulso de reloj para "enterarse" del estado de F·F(1) y F·F(2).



#### IV. DEFASAMIENTO DE RELOJ (CLOCK SKEW)

El principio básico en lógica síncrona es que el siguiente estado de todos los elementos de almacenamiento se determina en el mismo instante, sin embargo, en algunos casos no es posible alcanzar dicho objetivo.

En sistemas grandes, es posible que se requieran varios "drivers" para manejar todas las líneas de reloj, y si cada uno de estos dispositivos tiene diferente tiempo de respuesta, los pulsos de reloj llegarán a diferentes tiempos a los dispositivos. Este error de sincronización es llamado "Clock skew" y en los casos en que sea considerable puede ocasionar trastornos al sistema. Considérese la siguiente figura:



El circuito es un registro de desplazamiento cuyo estado inicial es "11" y se aplica un "cero" a la entrada  $D_1$ , lo cual debe llevar al sistema al estado "01", sin embargo, debido al defasamiento del reloj, el sistema va directamente al estado "00".

Observar que si el defasamiento de reloj fuese en el sentido inverso, es decir, que  $CK_2$  llegara antes que  $CK_1$ , el circuito operaría sin ningún problema, sin embargo en muchos casos es menester transferir información en ambas direcciones, en cuyo caso el problema seguiría presente. La solución más práctica sería reducir al máximo el defasamiento de reloj, ya sea empleando un solo driver de suficiente capacidad o bien empleando diferentes drivers muy rápidos y con tiempos de respuesta similares.

Otra situación que propicia defasamientos de reloj es la de operar circuitos MSI (mediano nivel de integración) con circuitería TTL, ya que mientras la mayoría de los circuitos MSI se sincronizan con el flanco de subida de "CK", la mayoría de los flip-flops J-K se sincronizan con el flanco de bajada del mismo (master-slave). Esta situación dificulta grandemente la utilización de un solo "driver" para la línea de reloj, y es necesario realizar inversiones lógicas lo cual acarrea defasamientos producidos por las etapas inversoras. Una solución es la de emplear solamente flip-flops rápidos y elementos de inversión cuyo tiempo de propagación sea significativamente menor que el de los flip-flops.

#### V. REFLEXIONES DE VOLTAJE.

Como quedó establecido anteriormente, el mecanismo formal de diseño proporciona resultados idealizados en comparación con lo que sería el comportamiento real del sistema, debido a que es incapaz de considerar las limitaciones operativas de los elementos empleados. Aunado a este problema se presenta la característica real de los alambres de interconexión de elementos. En vez de comportarse como conductores ideales, los alambres de interconexión agregan reflexiones a la señal y ruido, lo cual pueda llegar a ocasionar una operación errática del sistema.

En general, el problema de reflexiones está relacionado con la longitud de los cables empleados, así como con la rapidez de los elementos lógicos que conforman el sistema.

Anexo al presente trabajo se encuentra el artículo "Reflexiones de voltaje en la Interconexión de Elementos Digitales Rápidos", el cual contiene la información relativa tanto al análisis algebraico como al análisis gráfico del problema.

## BIBLIOGRAFIA

1. Blakeslee T.R.; "Digital Design with Standard MSI and LSI"; Wiley-Interscience, 1975
2. True K.M.; "The Interface Handbook, Line Drivers and Receivers"; Fairchild Semiconductor, 1975.



# Reflexiones de voltaje en la interconexión de elementos digitales rápidos

Jesús Alejandro Guinea Trigo\*  
Pedro Joselevich\*

## INTRODUCCION

En la actualidad, la técnica digital tiene gran importancia en los campos de la electrónica. Los circuitos digitales son parte esencial en los sistemas de control, comunicaciones, instrumentación y otros.

La actividad de los circuitos electrónicos digitales se basa en el manejo de variables eléctricas de carácter discreto, principalmente binario, y opera con funciones lógicas. Estas variables se representan por un intervalo de voltaje o corriente bien definido para cada uno de los valores discretos (fig 1). En el caso binario se asignan un par de intervalos para los estados de la variable digital conocidos como el estado alto (1) y el bajo (0). La variable se presenta en un estado a la vez y puede cambiarlo efectuando una conmutación.

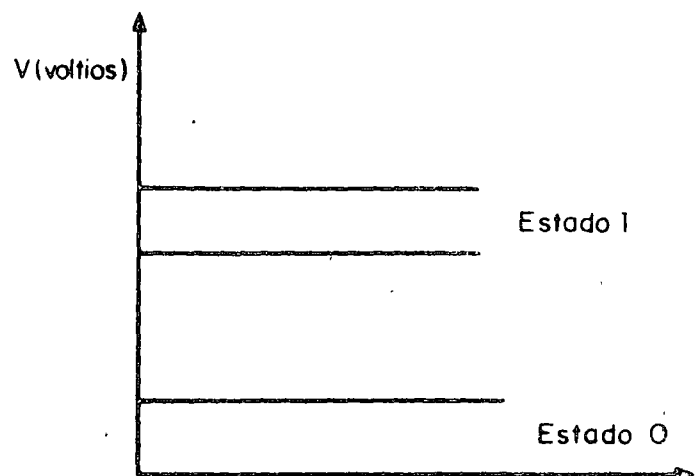


Fig 1 Los estados binarios en la electrónica digital

La electrónica digital está representada por varias familias de circuitos o familias lógicas que son autosuficientes y están a su vez formadas por celdas básicas llamadas compuertas (fig 2).

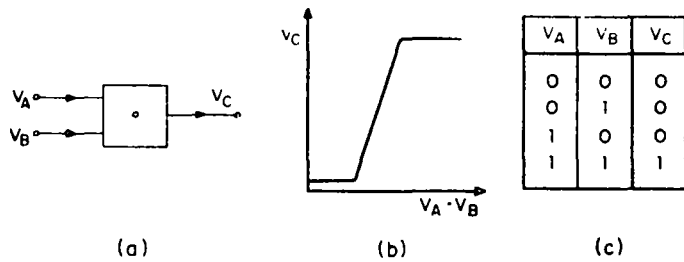


Fig 2 Símbolo, curva de transferencia y tabla de verdad de una compuerta lógica Y

La calidad o el mérito de la familia puede cuantificarse en función de los parámetros de su funcionamiento entre los que se encuentran los siguientes:

- Rapidez de conmutación
- Potencia de trabajo

Los diferentes desarrollos tecnológicos tienden a mejorar estos parámetros, siendo esta la causa de las variadas familias de circuitos digitales existentes.

La principal ventaja de la electrónica digital es la sencillez que presenta en el agrupamiento de componentes o celdas básicas para crear sistemas complejos.

Debido a la individualidad y aislamiento de cada celda, el funcionamiento del sistema depende del trabajo de cada una de sus partes; sin embargo, en la práctica no se puede considerar cierto esto debido a causas externas que actúan sobre las interconexiones, la alimentación, etc. Estas causas afectan la variable binaria, efectuando en ella cambios y degradaciones que representan errores en el funcionamiento del sistema. Los principales problemas se pueden resumir como sigue:

- Acoplamiento electromagnético entre líneas
- Generación de señales eléctricas espurias en los lazos de tierra
- Efectos de reflexiones de voltaje y corriente

Los primeros pueden ser resueltos por medio de un cuidadoso diseño de las rutas de interconexión, de alimentación y tierra. Sin embargo, el último problema se presenta debido al efecto de una línea mal terminada y al fenómeno de conducción eléctrica, y

depende de factores que difícilmente se pueden alterar en el diseño, como son la rapidez de conmutación y la longitud de la interconexión.

La reflexión se lleva a cabo cuando un voltaje y una corriente que viajan por una línea llegan a un punto discontinuo donde no pueden avanzar y se ven reflejados. El fenómeno es de carácter transitorio, es indeseable para nuestro sistema, y debe ser cuantificado por el diseñador de sistemas y eliminado cuando su efecto sea considerable.

## 2. LA INTERCONEXION COMO LINEA DE TRANSMISION

En este caso, la señal de voltaje y corriente se presentará en dos estados y conmutará entre ellos, como se muestra en la fig 3, en un tiempo  $t_s$  o  $t_b$  (tiempo de subida y tiempo de bajada). El flanco de señal viaja por la línea a una velocidad dada por las características de la misma, como sigue (fig 5):

$$vd = \frac{k}{\sqrt{u_0 \epsilon_0}} \quad (1)$$

$u_0 \epsilon_0$  = constantes físicas del vacío

$k$  = constante relativa del conductor

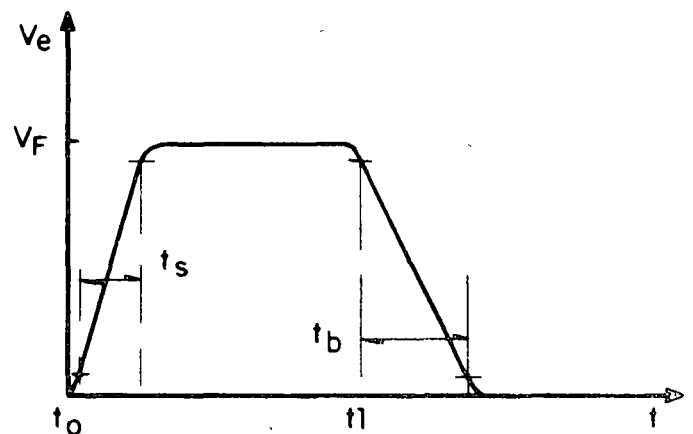


Fig 3 Forma de onda de la señal digital en un pulso de conmutación. Conmutación positiva en  $t = t_0$ . Conmutación negativa en  $t = t_1$

De esta manera, se define como tiempo de propagación el lapso que transcurre desde que sale la señal en  $x = 0$ , hasta que llega a la carga en  $x = d$  (fig 4)

$$\tau d = \frac{d}{vd} \quad (2)$$



Fig 4 Interconexión de elementos digitales

En la línea mal terminada y en el caso de que se trate con señales del tipo trapezoidal (fig 3), las reflexiones tienen un efecto considerable sólo en el caso en que

$$t_s < \tau d \quad \text{y} \quad t_b < \tau d \quad (3)$$

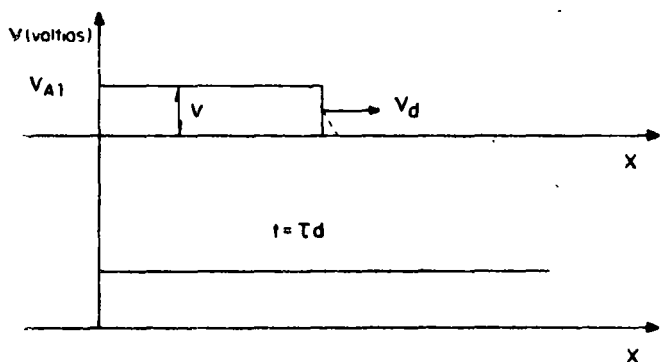


Fig 5 Propagación del pulso a lo largo de la línea. La onda incidente viaja de  $x=0$  a  $x=d$  a una velocidad de propagación  $v_d$

El flanco de la señal o frente de señal que se desplaza, formado por el voltaje y la corriente, llamados  $V, I$  (las flechas indican la dirección del movimiento) va generando, por cada punto que pasa, unos  $V, I$ , relacionados entre sí por las características puntuales de la línea (impedancia característica), que se modelan, para una línea sin pérdidas, como se ve en la fig 6. Se expresa ahora la impedancia característica como

$$z_o = \sqrt{\frac{L}{C}} \quad (4)$$

y siendo el sentido de la corriente el mostrado en la fig 6 se tendrá

$$z_o = \frac{\vec{V}}{\vec{I}} \quad (5)$$

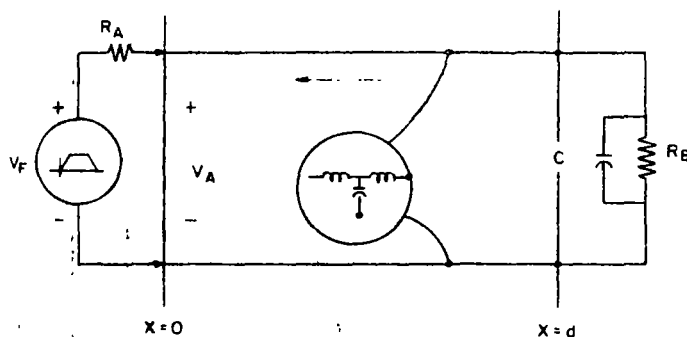


Fig 6 Modelo de la interconexión entre compuertas. En caso de que el efecto del capacitor sea despreciable, se tendrá el modelo simplificado

Desde el punto de vista del circuito, el transitorio de reflexiones lo carga considerablemente, como se puede deducir de la ec 5.

### 3. ANALISIS DE LAS REFLEXIONES EN LA CONMUTACION

Considerando que se está en presencia de la condición dada en la ec 3, se analizará el fenómeno en la conmutación positiva, transición del estado alto al bajo, y en la negativa, transición del estado bajo al alto, en un circuito como el de la fig 6. La conmutación positiva se realiza en el tiempo  $t = 0$  y la negativa en  $t' = 0$ .

Se proponen las variables del presente análisis definidas como sigue:

- $V_F$       voltaje de la fuente en el estado alto
- $V_{A_i}$     voltaje del punto A, para el intervalo  $2(i-1)\tau d \leq t \leq 2i\tau d$
- $V_{B_i}$     voltaje del punto B, para el intervalo  $(2i-1)\tau d \leq t \leq 2(i+1)\tau d$
- $I_{A_i}$     corriente tomada de la fuente en el intervalo igual al de  $V_{A_i}$
- $I_{B_i}$     corriente tomada por la carga en el intervalo igual al de  $V_{B_i}$
- $\vec{V}_i$     incremento de voltaje que viaja entre los puntos A y B (en el sentido de la flecha)

$I_i$  incremento de corriente que viaja entre los puntos **A** y **B** (en el sentido de la flecha)

$V_{re}, I_{re}, V_{00}, I_{00}$  valores estacionarios de voltaje y corriente alcanzados al terminar el efecto de las reflexiones en las conmutaciones positiva y negativa, respectivamente

### 3.1 Conmutación positiva

Considérese el circuito de la fig 6 inicialmente descargado, es decir, que el voltaje y la corriente en la línea son nulos

$$V_{A0} = V_{B0} = 0 \text{ para } t \leq 0 \quad (6)$$

$$I_{A0} = I_{B0} = 0 \text{ para } t \leq 0 \quad (7)$$

Se tienen las siguientes condiciones

$$\text{para } t \leq 0 \quad V = 0$$

$$0 \leq t \leq td \quad V = \frac{V_F}{ts} t \quad (8)$$

$$t \geq td \quad V = V_F \quad (9)$$

para  $t = 0_+$ , la fuente de voltaje ve la carga como una sola impedancia  $z_0$ , por lo que el voltaje en el punto **A** ( $x = 0$ ),  $V_{A1}$ , estará dado por

$$V_{A1} = V_F \frac{z_0}{R_A + z_0} \quad (10)$$

Nota: el subíndice indica el primer valor tomado por el voltaje en el punto **A** (fig 8a)

La corriente tomada de la fuente será

$$I_{A1} = - \frac{V_F}{R_A + z_0} \quad (11)$$

Por la definición del frente de señal que viaja,  $\vec{V}, \vec{I}$ , como incrementos de voltaje y corriente, se tiene:

$$\vec{V}_1 = V_{A1} - V_{B0} = V_{A1} \quad 0 < t < td \quad (12)$$

$$\vec{I}_1 = \frac{V_F}{R_A + z_0} = I_{A1} \quad (13)$$

Entonces de la ec 5:  $V_{A1} = -z_0 I_{A1}$  (13a)

En las ecs 10, 11 y 12, 13 se observa que el frente de señal para el primer "evento"  $0 < t < td$  está dado por los valores de  $V_{A1}, I_{A1}$ . El frente llega al punto **B**, en  $x = d$ , lugar donde está la discontinuidad ( $z_0 \neq R_B$ ), en un tiempo dado por  $t = td$ , ahí los valores,  $\vec{V}, \vec{I}$  no son congruentes, por la ley de Ohm, con el valor de la resistencia  $R_B$ , por lo que en ese punto el voltaje tomará el valor  $V_{B1} \neq \vec{V}$  y circulará una corriente  $I_{B1} \neq \vec{I}$ , de tal modo que se cumpla que

$$\frac{V_{B1}}{I_{B1}} = R_B \quad (14)$$

y que las diferencias de voltaje y corriente

$$V_{B1} - \vec{V} = \vec{V} \quad (15)$$

$$\vec{I} - I_{B1} = \vec{I} \quad (16)$$

cumplan con

$$\frac{\vec{V}}{\vec{I}} = z_0 \quad (17)$$

Los valores  $\vec{V}, \vec{I}$  son el voltaje y la corriente reflejados en el punto **B** y están relacionados con  $V, I$  por los coeficientes de reflexión. Atendiendo a las reflexiones de voltaje se puede expresar el coeficiente de reflexión de voltaje como

$$\vec{V}_1 / \vec{V} = \rho_B \quad (18a)$$

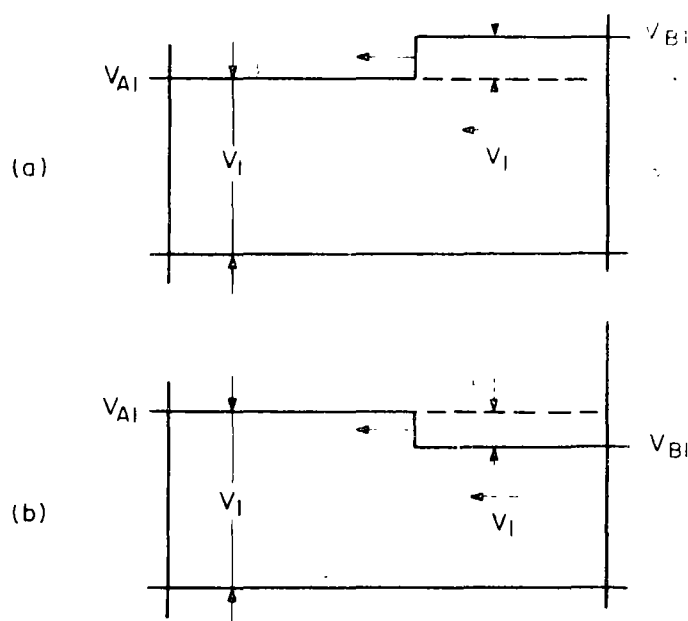


Fig 7 Onda reflejada en el punto **B** ( $x=d$ ) para un tiempo  $td \leq t \leq 2td$  a)  $R_B > Z_0$ , b)  $R_B < Z_0$

El valor de  $\vec{V}_1$  para  $td < t < 2$  se puede expresar como (fig 7)

$$\vec{V}_1 = V_{B1} - V_{A1} \quad (18b)$$

De las ecs 5 y 16 se obtiene

$$\rho_B = \frac{R_B - z_0}{R_B + z_0} \quad \text{donde } -1 \leq \rho_B \leq 1 \quad (19)$$

Usando  $\rho_B$  y despejando  $V_{B1}$ , en la ec 15 se obtiene

$$V_{B1} = V_{B0} + \vec{V}_1 (\rho_B + 1) = V_{A1} 2 \frac{R_B}{R_B + z_0} \quad (20)$$

Desde este momento,  $t = t_d +$ , el valor  $\vec{V}_1$  viaja hacia el punto  $A$  experimentando los mismos fenómenos que el voltaje  $V_{A1} = \vec{V}_1$ , de modo que al llegar a  $A$  se reflejará según el coeficiente  $\rho_A$  generándose  $\vec{V}_2$ .

$$\frac{\vec{V}_2}{\vec{V}_1} = \rho_A \quad (21)$$

$$\rho_A = \frac{R_A - z_0}{R_A + z_0} \quad -1 \leq \rho_A \leq 1 \quad (22)$$

igualmente que en la ec 20.

$$V_{A2} = V_{A1} + \vec{V}_1 (\rho_A + 1) = \quad (23)$$

Observando que  $\vec{V}_2$  se comporta análogamente a  $\vec{V}_1$ , se pueden generalizar las ecuaciones anteriores como sigue

$$\vec{V}_{n+1} = V_{A_{n+1}} - V_{B_n} \quad \text{para } 2ntd < t < (2n+1)td \quad (24)$$

$$n = 0, 1, \dots$$

$$\vec{V}_{n+1} = V_{B_{n+1}} - V_{A_{n+1}} \quad \text{para } (2n+1)td < t < (2n+2)td \quad (25)$$

De la misma manera, las ecs 18 a 21 quedarán.

$$\rho_B = \frac{\vec{V}_{n+1}}{\vec{V}_n} \quad (26)$$

$$\rho_A = \frac{\vec{V}_{n+2}}{\vec{V}_{n+1}} \quad (27)$$

Y de las ecs 20 y 23 se obtiene:

$$V_{B_{n+1}} = V_{A_{n+1}} (1 + \rho_B) - V_{B_n} \rho_B \quad (28)$$

$$V_{A_{n+2}} = V_{B_{n+1}} (1 + \rho_A) - V_{A_{n+1}} \rho_A \quad (29)$$

El valor de  $V_{A1}$ , está dado en la ec 10.

El efecto transitorio de las reflexiones se puede obtener observando las ecs 19 y 22. El valor de las reflexiones va disminuyendo, llegándose prácticamente después de varios tiempos,  $td$ , a un estado estacionario,  $V_{ee}, I_{ee}$ , donde la línea sin pérdidas se comporta como un nodo y se tiene:

$$V_{ee} = V_A = V_B = V_F \frac{R_B}{R_A + R_B} \quad (30)$$

$$I_{ee} = -V_{ee}/R_B \quad (31)$$

### 3.2 Conmutación negativa

Sean ahora las condiciones iniciales de la línea las de una línea cargada, y conmutese de 1 a 0 (fig 3)

$$t' \leq 0 \quad V = V_F$$

$$0 \leq t' \leq tb \quad V = -\frac{V_F}{tb} t \quad (32)$$

$$t' \geq tb \quad V = 0$$

Se tiene que  $V_A = V_B = V_{ee}$  y la corriente como en la ec 31 (fig 8). Luego de la conmutación de 1 a 0, la línea cargada a  $V_{ee}$  no se podrá descargar inmediatamente, el punto  $A$  tomará un voltaje  $V_{A1}$  diferente de  $V_{ee}$ , por lo que se tendrá un frente  $\vec{V}_1$  que estará dado por

$$\vec{V}_1 = V_{ee} - V_{A1} \quad (33)$$

$$\text{igualmente la corriente } \vec{I}_1 = I_{A1} - I_{ee} \quad (34)$$

El valor de  $V_{A1}$ , se puede obtener de la ec 34 como sigue

$$\vec{V}_1 = V_{A1} \frac{z_0}{R_A} - z_0 I_{ee} \quad (35)$$

Sustituyendo en la ec 33

$$V_{ee} = V_{A1} (1 + z_0/R_A) - z_0 I_{ee} \quad (35a)$$

De donde, finalmente, se obtiene

$$V_{A1} = \frac{R_A}{R_A + z_0} (V_{ee} + I_{ee} z_0) \quad (36)$$

Por la ec 33 se sabe cuál será el frente  $\vec{V}_1$  y por el procedimiento idéntico al de las ecs 9 en adelante se obtienen los valores  $V_{B1}$ ,  $V_{A2}$ , etc. Obsérvese que los valores  $V_{An}$ ,  $V_{Bn}$  cambian cada  $2ntd$  y  $(2n + 1)td$ , respectivamente (fig 8).

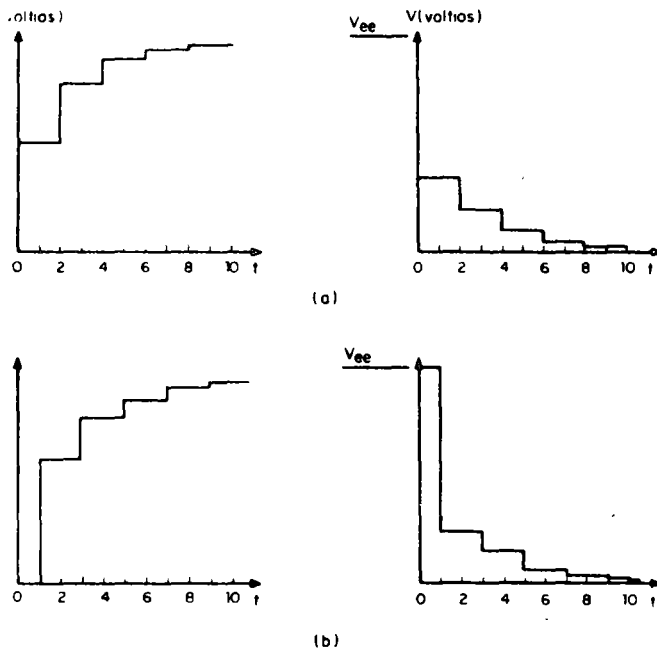


Fig 8 Efectos de las reflexiones observados en los puntos A y B (figs a y b, respectivamente) para las conmutaciones positiva y negativa

Las ecuaciones generales para este caso son idénticas a las tratadas en la conmutación positiva y numeradas del 24 al 29. En este caso el valor estacionario está dado por

$$\begin{aligned} V_{00} = V_A = V_B = 0 \\ I_{00} = 0 \end{aligned} \quad (37)$$

Hasta ahora el análisis se ha basado en un circuito como el de la fig 5, donde las resistencias  $R_A$  y  $R_B$  son lineales; de modo que en estas ecuaciones los coeficientes de reflexión son lineales respecto al voltaje. Por lo contrario, en los circuitos digitales dichos coeficientes son funciones no lineales del voltaje ( $\rho_{(V)}$ ); sin embargo las fórmulas siguen siendo válidas aunque su resolución se complica.

Las ecs 28 y 29 son adaptadas a programas de diseño de circuitos donde se puede manejar la no linealidad de los coeficientes de reflexión.

#### 4. METODO GRAFICO

El diseñador de sistemas digitales puede encontrar una solución en el método analítico y en los programas de computadora; sin embargo, existe la posibilidad de usar un método gráfico rápido y sencillo, el cual se describe a continuación.

Dicho método consiste en trabajar con las curvas características, voltaje-corriente ( $V-I$ ) de los dispositivos que entran en juego en el "circuito modelo" de la fig 6, mismas que se encuentran demostradas en la fig 9

En el circuito se observa que se presenta un problema de divisores de voltaje que se tratará a continuación.

##### 4.1 La conmutación positiva

El tratamiento gráfico para los divisores de voltaje consiste en encontrar el primer valor,  $V_{A1}$ , que es generado por un voltaje  $V_F$  a través de la resistencia  $R_A$  (resistencia interna de la fuente), y la resistencia de carga ( $z_0$  en la fig 10a), actuando estas como un divisor de tensión.

En la fig 10a se muestra el circuito de cuyo análisis se obtienen las siguientes ecuaciones.

$$\begin{aligned} V &= -I z_0 \\ V &= V_F + I_1 R_A \end{aligned} \quad (38)$$

Observando el sentido de la corriente  $I$ , la solución gráfica de la ec 38 ( $V_{A1}$ ,  $I_{A1}$ ) estará dada por la intersección de las curvas relativas a las mismas;  $I = f_1(V)$ ,  $I = f_2(V)$ , que se muestran en las figs 9a y 9c, o sea

$$I = -\frac{1}{z_0} V \quad (39)$$

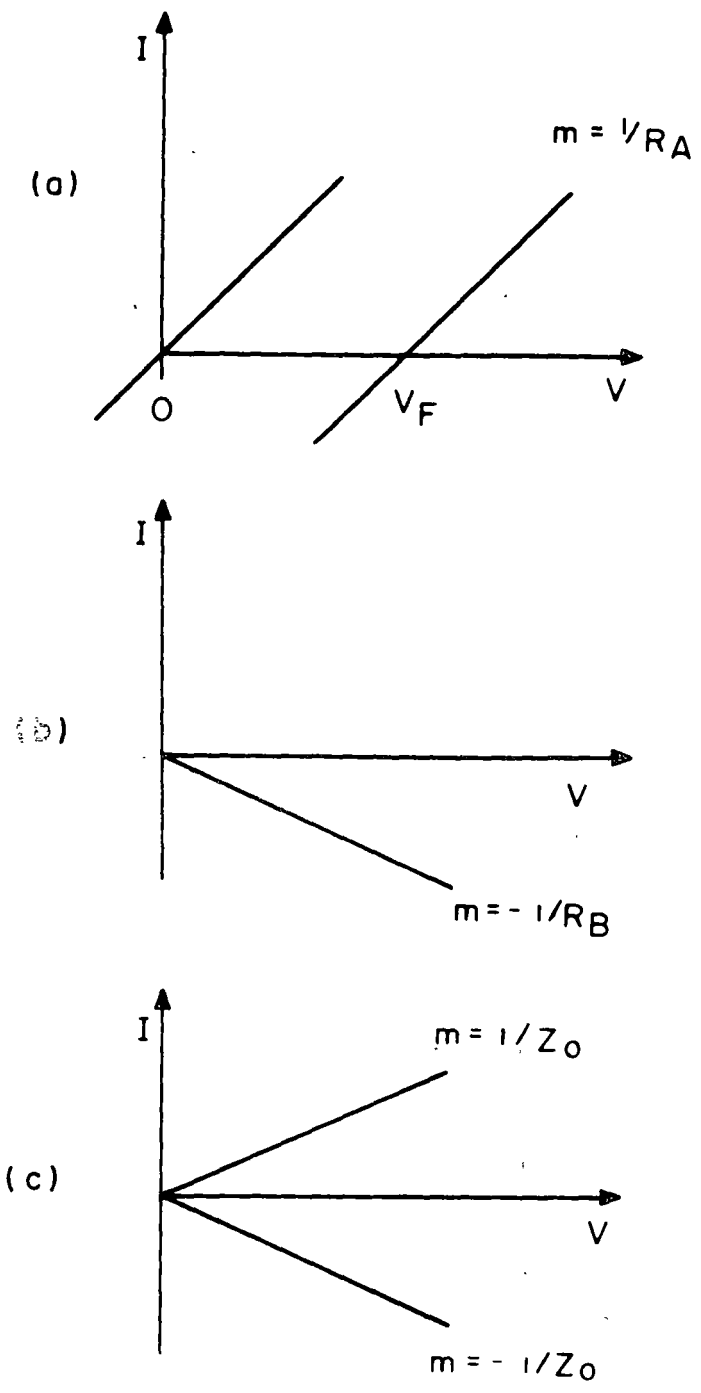


Fig 9 Rectas características, voltaje-corriente de: a) fuente no excitada y excitada; b) carga; c) línea de interconexión actuando como comando y como carga

$$I = \frac{V - V_F}{R_A} \quad (39a)$$

donde  $f_1$  es una recta con pendiente negativa ( $-1/z_0$ ) y  $f_2$  es una recta con pendiente ( $1/R_A$ ) que corta al eje de voltaje en el valor  $V_F$ .

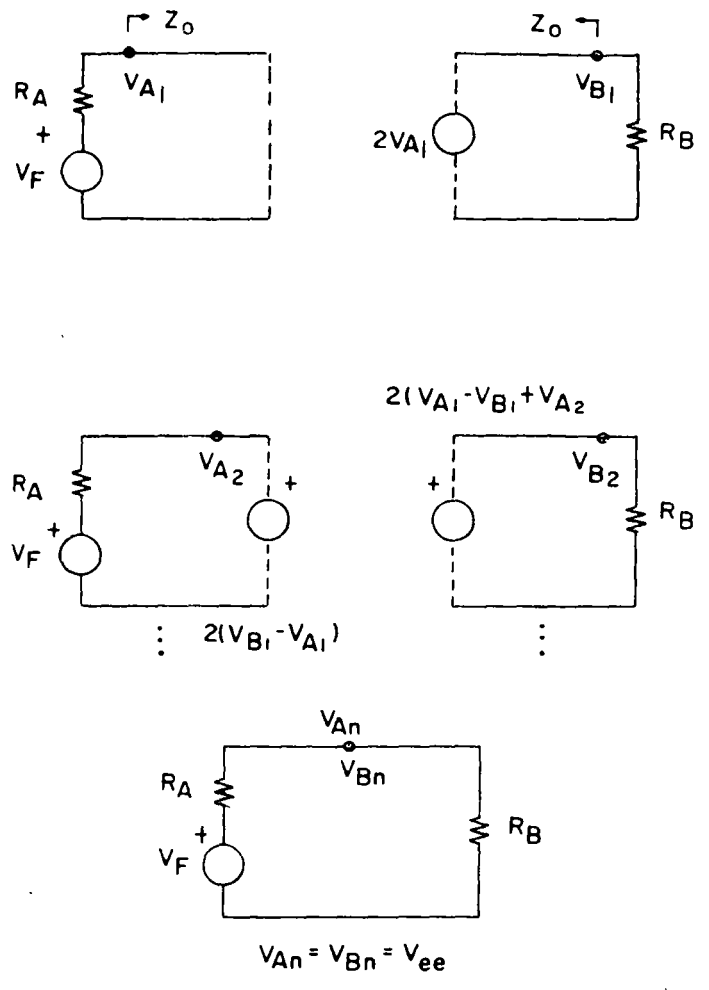


Fig 10 Divisores de voltaje usados en la solución por método gráfico

La solución de las ecs 39 para el voltaje es

$$V_{A1} = V_F \frac{z_0}{R_A + z_0} \quad (39b)$$

Una vez que se ha encontrado  $V_{A1}$  (fig 11a), se procede a obtener  $V_{B1}$ , el cual está dado por la ec 20 hallada anteriormente

$$V_{B1} = 2 V_{A1} \frac{R_B}{R_B + Z_0} \quad (39)$$

O sea, que el valor del voltaje en el punto B es función del voltaje del punto A obtenido con anterioridad. La ec 39 representa un divisor de voltaje cuya expresión gráfica está dada por las siguientes ecuaciones (figs 9 y 10a)

$$I = -\frac{-1}{R_B} V \quad (40)$$

$$I = \frac{V - 2V_{A1}}{Z_0}$$

La primera ecuación es la de la recta de carga, y la segunda, la de la recta de comando (fig 11a), como se puede observar, la recta correspondiente a la impedancia de la línea de interconexión parte siempre del valor de la intersección anterior y se dirige a intersecar en forma alternada las rectas correspondientes a la fuente de voltaje y a la resistencia de carga. De este modo se van obteniendo los voltajes  $V_{A2}, V_{B2}, \dots$  hasta que el procedimiento converge al valor estacionario  $V_{ee}, I_{ee}$ .

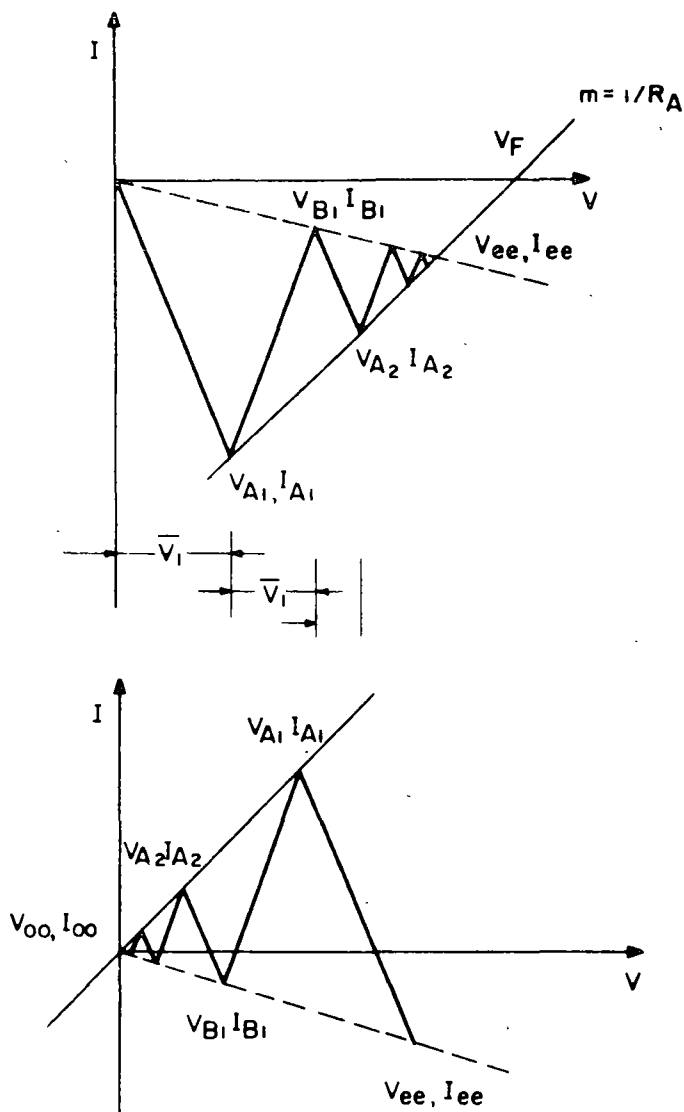


Fig 11 Método gráfico para obtener los voltajes en la reflexión: a) conmutación positiva, b) conmutación negativa

## 4.2 Conmutación negativa

De la misma forma que el estudio anterior, en la conmutación negativa el primer valor de voltaje,  $V_{A1}$ , después de la conmutación, se obtiene por la intersección de la característica de la fuente con  $V_F = 0$  y pendiente  $1/R_A$  y la recta de carga  $-1/Z_0$  trazada desde los valores estacionarios a los que estaba cargada la línea ( $V_{ee}, I_{ee}$ ) (fig 11b). Continuando el procedimiento igual al de la conmutación positiva se encuentran  $V_{B1}, V_{A2}, \dots$  finalizando con el estado estacionario ( $V_{oo}, I_{oo}$ ) donde en el presente circuito se tendrá

$$V_{oo} = 0 \quad (44)$$

$$I_{oo} = 0 \quad (45)$$

## 4.3 Aplicación del método gráfico al análisis en circuitos digitales

El método gráfico requiere del conocimiento de las características  $V-I$ , tanto del elemento de comando en sus dos estados como de la carga. Las curvas pueden obtenerse del manual del fabricante o lograrse de manera experimental. Las curvas para *TTL* y *ECL* se muestran en la fig 12, donde se puede observar que las características representan resistencias no lineales y son diferentes de una familia lógica a otra. Sin embargo, el método gráfico sigue siendo válido para cualquier tipo de características.

En las curvas se indican los puntos de estado estacionario, como  $Q_{ee}$  y  $Q_{oo}$ .

## 5. LOS PROBLEMAS DE LAS REFLEXIONES Y SU SOLUCION

### 5.1 Problemas principales que se presentan en un sistema lógico cuando hay reflexiones

- Retraso en la transmisión. La conmutación de la fuente no se realiza de inmediato, sino que va cambiando por pasos (fig 8). Esto provoca algunas veces que el umbral de conmutación de la compuerta receptora no sea alcanzado hasta después de varias reflexiones.
- Transición en la región prohibida. Se da el caso en que el valor del voltaje en la entrada del receptor se mantiene en la región prohibida (fig 1) por un tiempo, dando una salida indeterminada.



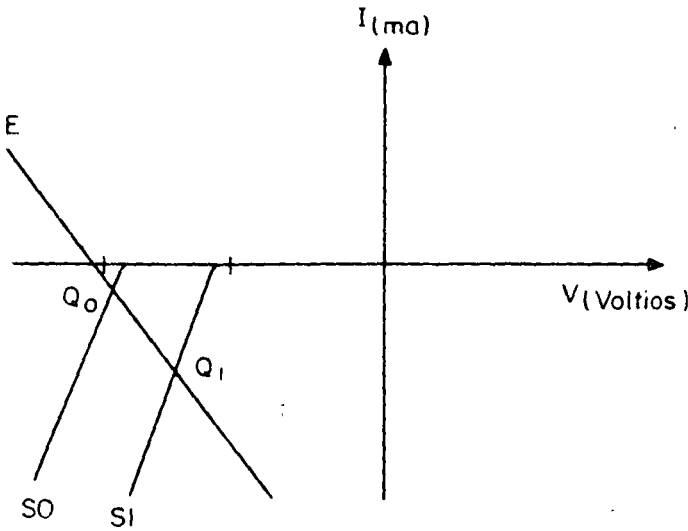
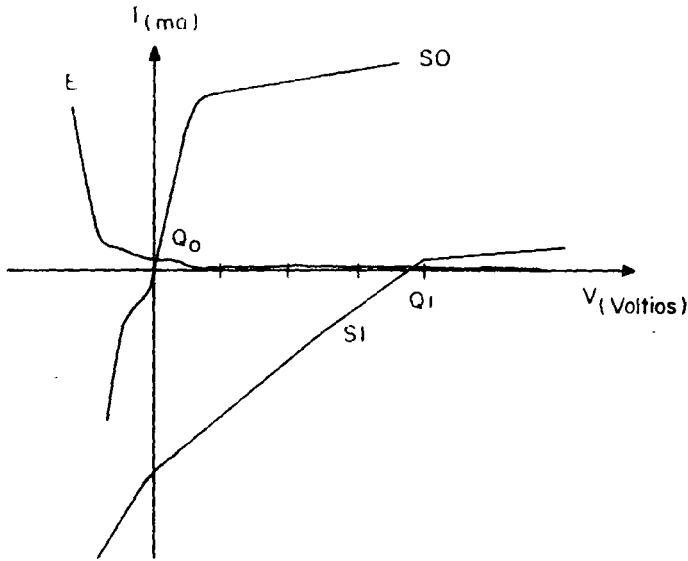


Fig 12 Curvas características, voltaje - corriente, de los circuitos digitales; a) familia TTL; b) familia ECL, donde E=entrada; SI=salida del estado 1; SO = salida en el estado 0

- c) Efectos en la salida; compuerta de comando. La compuerta que comanda presenta en su salida también el efecto de las reflexiones, por lo que cualquier otra compuerta conectada en la salida (punto A) puede tener los mismos problemas que aquella al final de la línea.
- d) Corriente de línea. En la conmutación la fuente debe cargar una línea de baja impedancia ( $Z_0$ ), por lo cual el consumo de corriente es grande y puede ser que la compuerta no esté capacitada para dar esa corriente y no podrá manejar la línea.

e) Si el comando es un elemento biestable, la onda reflejada puede hacerlo cambiar de estado erróneamente.

**5.2 SOLUCION**

El problema de las reflexiones puede resolverse siguiendo estas recomendaciones:

- a) Buena terminación de la línea. Se puede compensar la terminación de la línea cargando con elementos resistivos en serie o en paralelo (fig 13). El procedimiento serie provoca una disminución en el nivel de voltaje de la señal que se trasmite, la cual hay que considerar, y el segundo carga el circuito con una corriente de drenaje que limita aún más la capacidad de comando.
- b) Uso de elementos lógicos que actúen como transmisores o receptores. En caso de no poder comandar la línea con elementos normales, se pueden usar interfases de más capacidad que pueden conseguirse en el mercado.
- c) Abstenerse de manejar líneas con elementos biestables, manteniendo a estos propiamente aislados mediante compuertas.

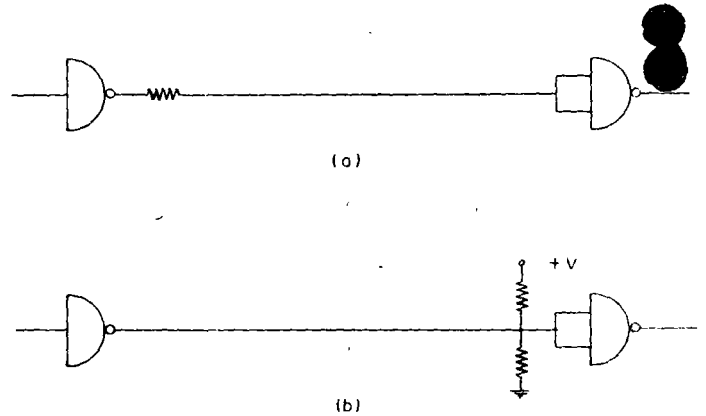


Fig 13 Conexiones usuales para eliminar las reflexiones: a) conexión serie, b) conexión paralelo

**6. LA SIMULACION VIA PROCESADOR DIGITAL**

Debido a que el análisis de las reflexiones presentó una solución recursiva (ecs 28 y 29), se supone que el problema puede ser fácilmente simulado en una computadora digital.

No es propósito de este trabajo presentar el program, codificado en ningún lenguaje, por lo que se pod

usar cualquier máquina, desde una calculadora programable, un microprocesador y minicomputador hasta las máquinas de gran capacidad.

Es interesante plantear el diagrama de flujo, donde se indica claramente el procedimiento a seguir para resolver el problema (diagrama 1).

Como se puede observar, el programa propuesto trabaja sobre los datos tanto de características de la línea (velocidad,  $L$ ,  $C$ ) como de las cargas terminales, los voltajes de conmutación, el sentido de conmutación y la rapidez de la fuente de conmutación

El programa obtiene los siguientes resultados, 1) El voltaje y la corriente estable del circuito al final de las reflexiones transitorias, 2) Los valores discretos que toma el voltaje en el comando y en la carga mientras dura el efecto de las reflexiones, 3) Los coeficientes de reflexión al inicio, punto A, y al final de la línea, punto B, y 4) El tiempo de propagación de la señal en la línea

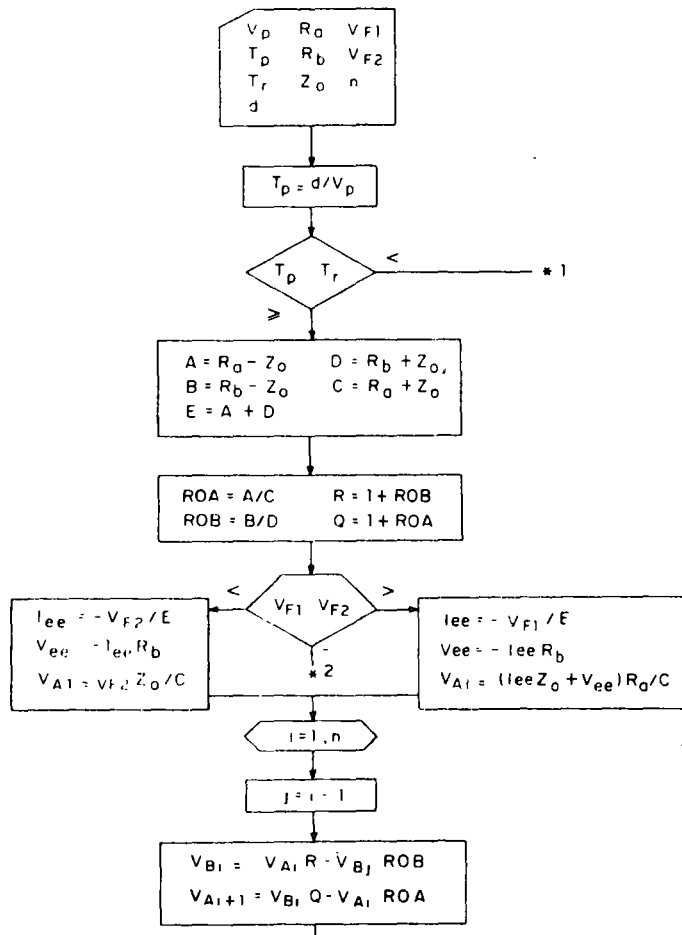


Diagrama 1 Diagrama de flujo del programa para obtener las reflexiones.

A partir de los datos obtenidos en el punto 2, se puede hacer que la máquina despliegue la información en forma gráfica (fig 8), empleando un formato propio

Para usar el programa en el estudio de sistemas electrónicos como el TTL, las características  $V.I$  de entrada y salida se pueden linealizar por segmentos. Esto implicaría el uso de instrucciones de salto condicionales en el programa que tomaría una recta diferente con su propia pendiente para cada intervalo de voltaje.

Del diagrama de flujo  $\left\{ \begin{array}{l} * \text{ Com 1 : esta simulación está fuera de lo tratado por este programa} \\ * \text{ Com 2 : no hay conmutación} \end{array} \right.$

## 7. EFECTOS REALES

### 7.1 Efecto de las pérdidas en la línea

Las pérdidas en la línea de interconexión se toman en cuenta solo para longitudes muy grandes (7 300 m). El efecto de estas en el presente análisis consiste únicamente en que las reflexiones son de menor amplitud y el transitorio se anula más rápidamente. En el estado estacionario, se tendrá una caída de voltaje en la línea en función de la longitud,  $V_{Lee(x)}$ , y valdrá la siguiente expresión:

$$V_{Aee} = V_{Lee(x)} V_{Bee} \quad (46)$$

el voltaje en B estará degradado por  $V_{Lee(x)}$

### 7.2 Efecto de capacitancia de carga

→ Si se considera ahora el caso en que la carga está formada por una resistencia en paralelo con una capacitancia ( $C \neq 0$ ) (fig 5) cuyo valor es considerable para las velocidades de conmutación, el programa se altera, pues ahora los coeficientes de reflexión se vuelven complejos y la respuesta de la señal se comporta como se observa en la fig 14.

En la ref 2 se cita un programa desarrollado por Motorola, donde se generaliza la carga a una impedancia  $RC$ .

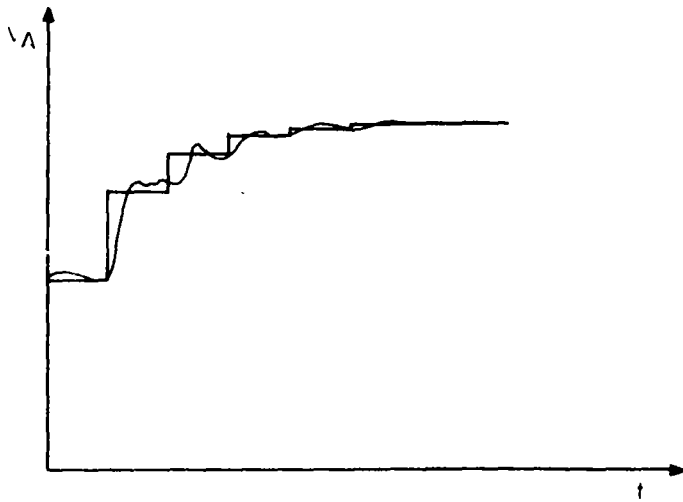


Fig 11 Efectos de la carga capacitiva en el fenómeno de las reflexiones

## 8. CONCLUSIONES

Se considera necesario que todo diseñador de sistemas digitales tenga presente estos fenómenos prácticos en su diseño y analice todos los parámetros de sus interconexiones, que son tan importantes como las propias compuertas.

Los métodos que se muestran aquí pueden ayudar al diseñador o al estudiante de circuitos digitales a comprender y resolver el problema de las reflexiones en los elementos lógicos rápidos.

## 9. REFERENCIAS

1. Fields Waves in Communication Electronics Ramo-Whinnery, **John Wiley**
2. Computer Design, CAD Program to MS Logic Interconnections (may 1975)
3. Handbook for Radio Engineers, Terman, **McGraw-Hill**
4. Electromagnetic Waves and Radiating Systems Jordan/Balmain, **Prentice Hall**, 2nd edition

Directorio de Alumnos del Curso: APLICACIONES DE CIRCUITOS INTEGRADOS AL DISEÑO DIGITAL del 12 al 23 de Junio de 1978.

1. SR. JAVIER JESUS BUSTAMANTE PARTIDA  
Automatización Industrial, S.A. de C.V.  
José A. Torres No. 520  
Col. Vista Alegre  
México 8, D.F.  
Tel. 519-32-09  
Diego Arenas Guzmán No. 198-2  
Col. Villa de Cortés  
México 13, D.F.  
Tel. 590-61-98
2. SR. J. MANUEL ESPINOSA ARANDA  
Instituto de Ingeniería  
Ciudad Universitaria  
México 20, D.F.  
Tel. 548-54-18  
Villa Olímpica Edif, 10-104  
Col. Tlalpan  
México 22, D.F.  
Tel. 568-39-71
3. ING. JORGE GARCIA CAMACHO  
Centro de Cálculo de la Facultad de  
Ingeniería - UNAM  
Ciudad Universitaria  
México 20, D.F.  
Tel. 550-52-15 ext. 4150  
Dr. Balmis No. 24 Int. 14  
Col. Doctores  
México 7, D.F.
4. SR. FERNANDO GARCIA NUÑEZ CANO  
Escuela Nacional de Estudios Profesionales C.  
Dom. Conocido  
Cuautitlán Izcalli, Edo. de México  
Esther No. 10  
Col. Industrial  
México 14, D.F.  
Tel. 517-76-21 - 517-23-19
5. SR. LUIS CARLOS GONZALEZ DURAZO  
F. IOS CER  
Av. Revolución No. 1570  
Col. San Angel  
México 20, D.F.  
Tel. 550-72-55  
Angel Urraza No. 718 - 102  
Col. Del Valle  
México 12, D.F.  
Tel. 575-11-47
6. SR. CLAUDIO GONZALEZ HERMOSILLO  
Honeywell, S.A.  
Av. Constituyentes No. 900  
Col. Lomas Altas  
México 10, D.F.  
Tel. 570-20-33 ext. 172  
Retorno 204 No. 2  
Col. Unidad Modelo  
México 13, D.F.  
Tel. 582-95-39

7. SR. JOSE GPE. MALDONADO HERAS  
Plaza De Comonfort No. 7-14  
Centró  
México 1, D.F.
8. SR. ISMAEL MARTINES ALVA  
Sub-Secretaría de Mejoramiento del  
Ambiente  
Col. Roma  
México 7, D.F.  
Tel. 528-74-56
9. SR. JAVIER MARTINES NAVARRO  
SARH Comisión del Plan Nacional  
Hidráulico  
Col. Roma  
México 7, D.F.  
Tel. 574-10-73
10. SR. PEDRO MENDEZ ROJAS  
Crouse Hinds Domex, S.A. de C.V.  
Javier Rojo Gómez No. 277  
Col. Iztapalapa  
México 12, D.F.  
Tel. 582-33-00
11. SR. JOSE FRANCISCO PEREZ GARCIA  
F. Angeles y 5 de Mayo  
Col. Lomas de la Selva  
Cuernavaca, Mor.  
Tel. 3-01-18
12. SR. DANIEL RIOS ZERTUCHE ORTUÑO  
Facultad de Ingeniería  
Ciudad Universitaria  
México 20, D.F.
13. SR. HECTOR RIVERA MARTINEZ  
Centro de Cálculo  
Facultad de Ingeniería  
Ciudad Universitaria  
México 20, D.F.  
Tel. 550-52-15 ext. 4150
- José T. Cuellar No. 311  
Col. Vista Alegre  
México 8, D.F.  
Tel. 538-37-88 - 528-74-56
- Valle de Juárez No. 42-4  
Col. Valle de Aragón  
Edo. de México
- Dr. Barragán No. 738 Depto. 407  
Col. Narvarte  
México 12, D.F.  
Tel. 590-24-61
- Puebla No. 207  
Col. San Angel  
México 20, D.F.  
Tel. 548-42-69
- Campo Encantada No. 39  
Fracc. San Antonio  
México 16, D.F.  
Tel. 352-25-51

14. SR. ANTONIO SALVA CALLEJA  
Facultad de Ingeniería  
Ciudad Universitaria  
México 20, D.F.  
Tel. 550-52-15 ext. 3746
- Sur 73 No. 138  
Col. Prado  
México 13, D.F.  
Tel. 539-08-79
15. SR. VICTOR MANUEL SEVILLA ALCARAZ  
Facultad de Ingeniería  
Ciudad Universitaria  
México 20, D.F.
- Petén Sur No. 46-22  
Col. Narvarte  
México 20, D.F.
16. SR. ANASTASIO VARGAS HERNANDEZ  
Facultad de Ingeniería  
Ciudad Universitaria  
México 20, D.F.
- Av. 1° de Mayo No. 14  
Col. Melchor Ocampo  
Edo. de México.

