ARITMÉTICA Y CIRCUITOS BINARIOS

Los circuitos binarios que pueden implementar las operaciones de la aritmética binaria (suma, resta, multiplicación, división) se realizan con circuitos lógicos combinacionales (puertas lógicas conectadas).

SUMA BINARIA

Figura 1: Suma binaria

La suma o adición binaria es análoga a la de los números decimales. La diferencia radica en que en los números binarios se produce un acarreo (carry) cuando la suma excede de uno mientras en decimal se produce un acarreo cuando la suma excede de nueve(9). Del gráfico de la figura 1 podemos sacar las siguientes conclusiones:

1. Los números o sumandos se suman en paralelo o en columnas, colocando un numero encima del otro. Todos los números bajo la misma columna tienen el mismo valor posicional.
2. El orden de ubicación de los números no importa (propiedad conmutativa).

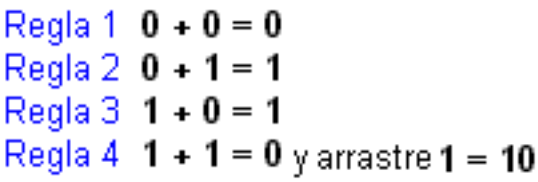


Figura 2: Reglas para la suma binaria

En la figura 2 se indican las reglas que rigen la suma binaria y en la figura 3 se muestra un circuito lógico llamado semisumador, que suma 2 bits (A y B) que genera un bit de suma y un bit de acarreo cuando este se produce. La operación de un semisumador como el anterior mostrado en la figura se puede sintetizar mediante las siguientes 2 operaciones

booleanas: Σ=A(xor)B (suma) Co=A·B (acarreo) Para realizar una suma binaria donde se tenga presente un carry de entrada se debe implementar un circuito que tenga presente esta nueva variante; como es el caso del sumador completo. El sumador completo tiene 3 entradas que se suman y son: A, B, y Cin (entrada de arrastre), y las salidas habituales Σ y Co (suma y salida de arrastre)

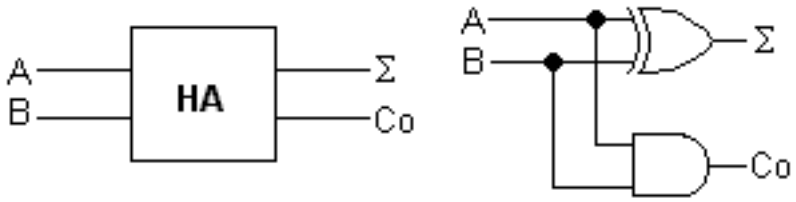


Figura 3: Semisumador

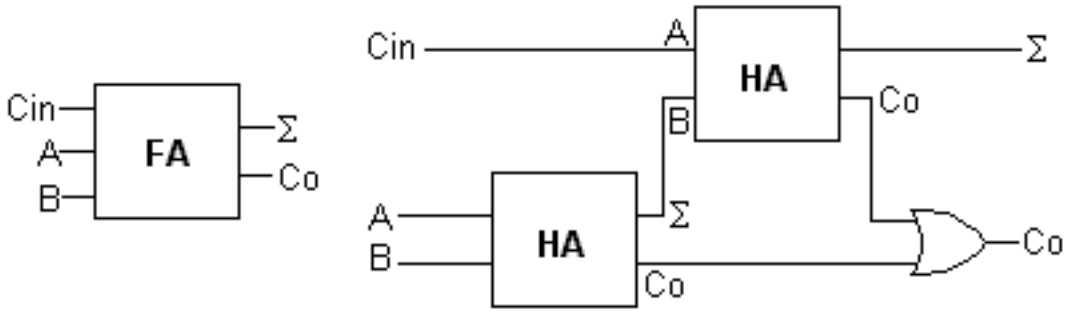


Figura 4: Sumador completo

RESTA BINARIA

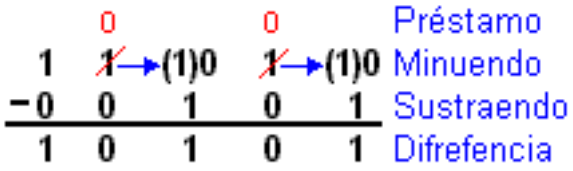


Figura 5: Resta binaria

La resta o sustracción de números binarios es similar a los números decimales. La diferencia radica en que, en binario, cuando el minuendo es menor que el sustraendo, se produce un préstamo o borrow de 2, mientras que en decimal se produce un préstamo de 10. Al igual que en la suma, el proceso de resta binaria, se inicia en la columna correspondiente a la de los dígitos menos significativos. En la figura 5 se indican las reglas que rigen la resta binaria y en la figura 6 se muestra un circuito lógico, llamado semirrestador (HS), que sustrae un B de un bit A y suministra un bit de diferencia (Di) y un bit de préstamo (Bo). La operación de un Semirrestador como el mostrado en la figura anterior se puede resumir mediante las 5 ecuaciones booleanas: Di=A·B(neg)+A(neg)·B= A(xor)B (diferencia) Bi=A(neg).B (borrow) En la figura siguiente se muestra el proceso de resta de 2 números binarios de 5 bits. El objeto de esta operación es ilustrar el manejo de los préstamos y plantear la necesidad de un restador completo de 2 bits que tenga, como entradas, el minuendo, el sustraendo, y el préstamo anterior y ofrezca como salidas, la diferencia y el préstamo, si existe. En la figura 7 se muestra el diagrama de bloques, conexión en bloques utilizando semirrestadores y una puerta OR y el diagrama lógico de un restador completo.

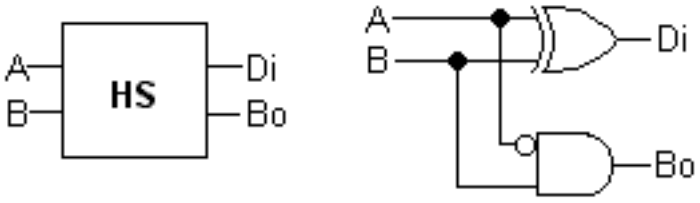


Figura 6: Semirrestador

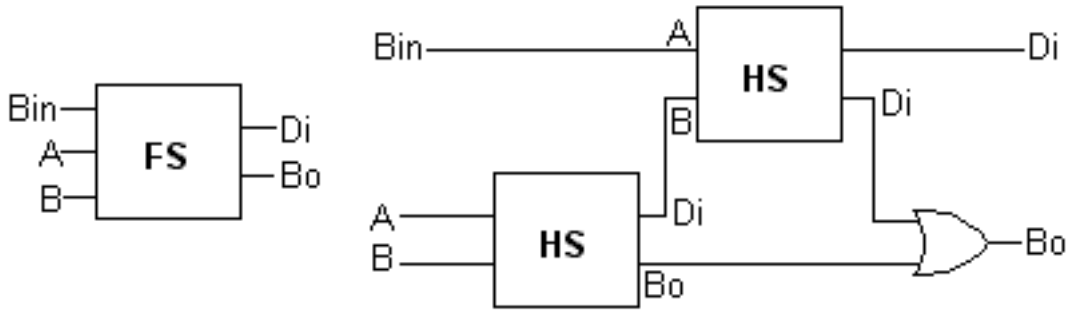


Figura 7: Restador completo ver figura pag 230 10.14

SUMADORES Y RESTADORES EN PARALELO

Los circuitos que realizan operaciones en paralelo son mas rápidos en sus respuestas, casi inmediatos para dar un resultado.

Para el caso de un sumador se toma el bit LSB de cada una de las palabras que vayan a

ser sumados y se llevan hacia las dos entradas de un semisumador (HA); donde la salida de suma puede mandarse a un visualizador el cual sería el LSB del resultado de la suma y la otra salida es la del CARRY OUT. Esta es llevada a un sumador completo (FA), el cual tiene presente 3 entradas que son : los dos bits consecutivos a los LSB de cada palabra binaria y un arrastre o acarreo de entrada que como mencionamos viene del semisumador (CARRY IN). De ahora en adelante en este ejercicio tomado como ejemplo (ver figura 9) las conexiones que se harán de la forma ya descrita (teniendo presente 3 entradas a sumar) con la única variante de que el CARRY IN ya no viene de un semisumador; sino de un sumador completo y, habrá igual numero de sumadores completos como bits menos 1 tengan las palabras binarias a sumar, debido a que el primer dispositivo a sumar es un semisumador. El CARRY OUT del ultimo sumador debe mandarse a un visualizador "en este caso" para tener presente el ultimo arrastre que se pueda generar.

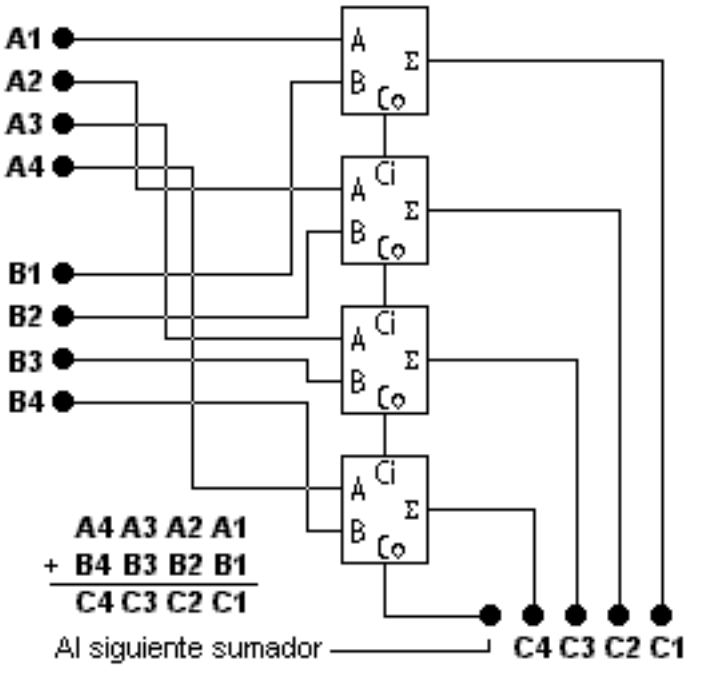


Figura 8: Sumador paralelo

Si aun te preguntas donde esta la conexión en paralelo regresa a la figura anterior y observa que los bits que son sumados (en HA y/o FA) son aquellos que tiene el mismo peso o valor por posición en cada uno de las palabras binarias. RESTADORES
La columna del 1 de la figura que se muestra al final utiliza un semirrestador (HS). Las columnas del 8,4 y 2 utilizan restadores completos (FS). Cada una de las salidas Di de los restadores esta conectada a un indicador de salida para mostrar la diferencia. Las lineas de

préstamo conectan la salida Bo de un restador a la entrada Bin del siguiente bit mas significativo. Las lineas de prestamos siguen las pista de los muchos prestamos de la resta binaria. Este tipo de restador da una respuesta casi inmediata.

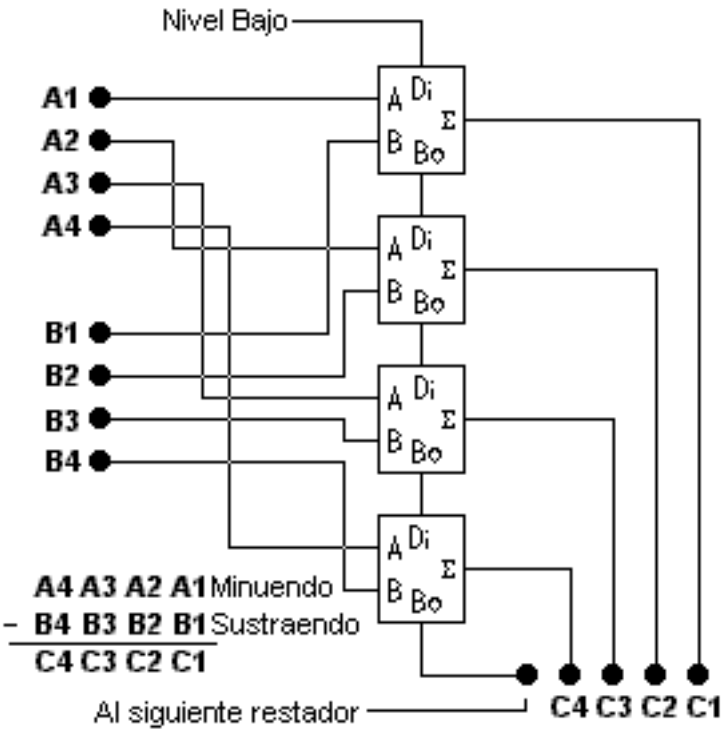


Figura 9: Restador paralelo

CIRCUITOS PRÁCTICOS

Estos circuitos no son mas que una estandarización de la circuitería empleada para el caso de los sumadores completos (FS) que el FA trabaja como HA.

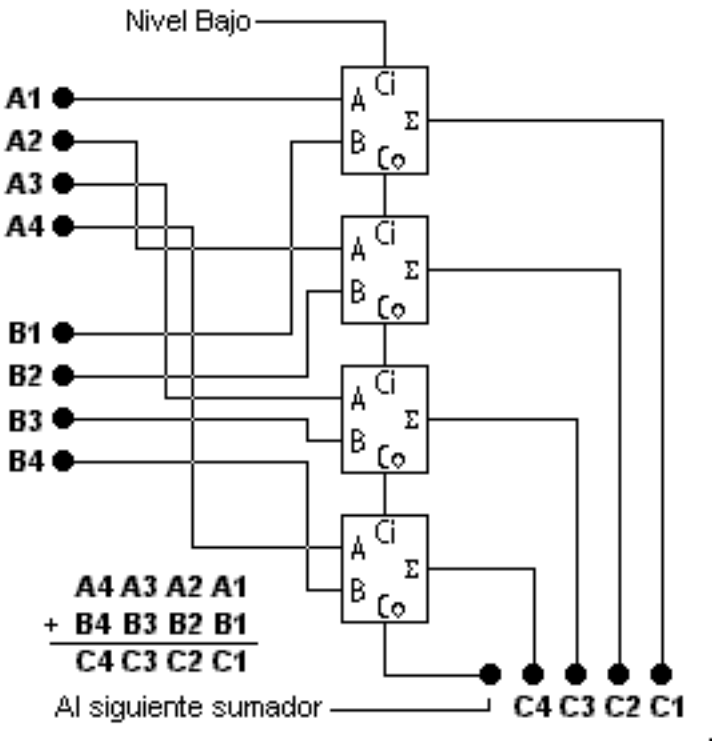


Figura 10: Sumador paralelo práctico

SUMADORES PARA LA RESTA

En una resta binaria están involucradas tres variables bien definidas: Minuendo, Sustraendo y Diferencia. Según la ley de la resta, estos parámetros se relacionan así:

```
Minuendo - Sustraendo = Diferencia
```

La resta de dos números se puede expresar también como la suma del minuendo mas el negativo del sustraendo, es decir:

```
Minuendo + (-Sustraendo) = Diferencia
```

Por ejemplo, la resta de 10 menos 5 se puede expresar como:

```
10 + (-5) = 5
```

Aplicando esta definición, es posible implementar la resta sumando el negativo del sustraendo al minuendo. Surge entonces una nueva forma en que podemos realizar la resta binaria, la cual se rige por las siguientes reglas:

1. Cambiar el sustraendo a su forma en complemento a 2.

2. Sumar el minuendo al sustraendo en complemento a 2.
3. No considerar el «overflow» (rebose). Se descarta el MSB, y los bits restantes indican la diferencia binaria.

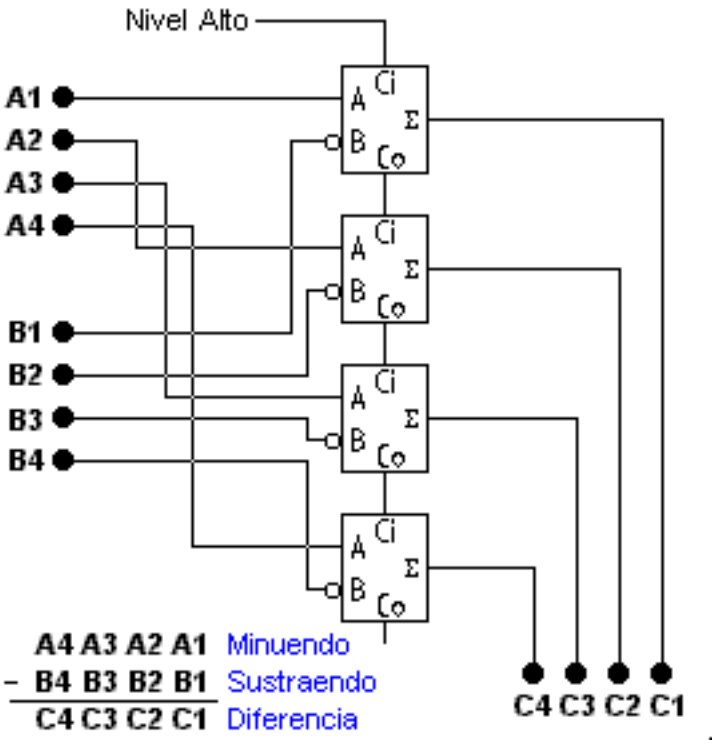


Figura 11: Restador de 4 bits utilizando sumadores completos

La razón por la cual el circuito anterior funciona como restador , se debe a que los cuatro inversores convierten el sustraendo binario a su complemento a 1 (cada 1 es cambiado a 0 y cada 0 a 1). El nivel alto de la entrada Cin en el FA del 1 es lo mismo que sumar +1 al sustraendo. El minuendo y el sustraendo en complemento a 2 se suman. El terminal Co del ultimo FA se descarta (overflow).

SUMADORES/RESTADORES

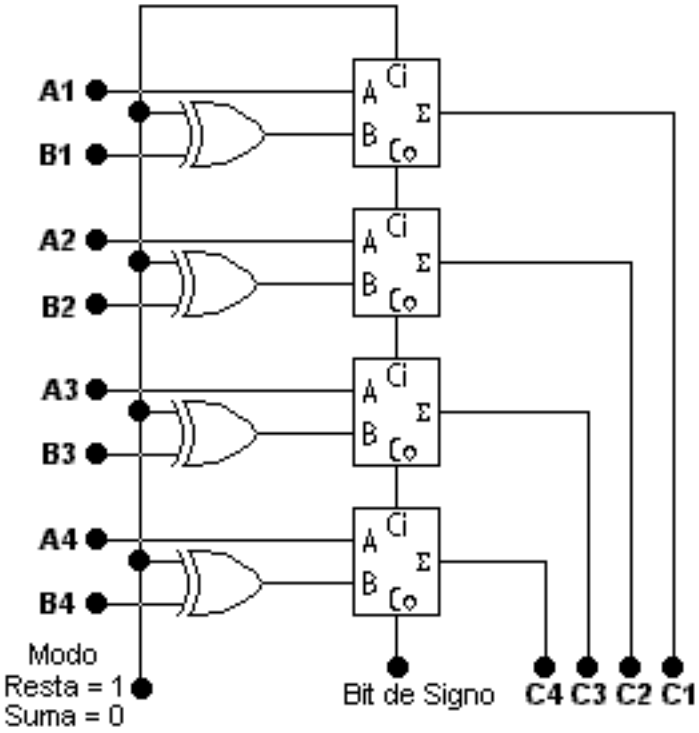


Figura 12: Sumador/restador de 4 bits

Si observamos los dos últimos gráficos podemos apreciar que estos circuitos son muy parecidos por lo que nos queda fácil implementar un circuito que realice las dos operaciones tratadas (suma y resta). El circuito Sumador/Restador mostrado en la figura 12 tiene una entrada adicional denominada MODO DE CONTROL. Si esta entrada está en un nivel bajo (0 lógico), las cuatro puertas XOR no tienen efecto en el dato de las entradas B (el dato pasa a través de las puertas XOR y no es invertido). La entrada Cin del primer FA es mantenido en un nivel BAJO, lo cual hace que este primer FA trabaje como semisumador. Cuando la entrada de Modo de Control esta en un nivel alto (1 lógico), las cuatro XOR actúan como inversores. Se invierte el sustraendo (entradas B). La entrada Cin del primer FA esta en un nivel ALTO, lo que es lo mismo que sumar +1 al sustraendo en complemento a 1. La diferencia (resultado) se puede apreciar en los visualizadores.

FAMILIAS LOGICAS DE CIRCUITOS INTEGRADOS

Una familia lógica es el conjunto de circuitos integrados (CI's) los cuales pueden ser interconectados entre si sin ningún tipo de Interface o aditamento, es decir, una salida de un CI puede conectarse directamente a la entrada de otro CI de una misma familia. Se dice entonces que son compatibles.

Las familias pueden clasificarse en bipolares y MOS. podemos mencionar algunos ejemplos. Familias bipolares: RTL, DTL, TTL, ECL, HTL, IIL. Familias MOS: PMOS, NMOS, CMOS. Las tecnologías TTL (lógica transistor- transistor) y CMOS (metal oxido-semiconductor complementario) son los mas utilizadas en la fabricación de CI's SSI (baja escala de integración) y MSI (media escala de integración).

CARACTERÍSTICAS GENERALES

NIVELES LOGICOS

Para que un CI TTL opere adecuadamente, el fabricante especifica que una entrada baja varíe de 0 a 0.8V y una alta varíe de 2 a 5V. La región que esta comprendida entre 0.8 y 2V se le denomina región prohibida o de incertidumbre y cualquier entrada en este rango daría resultados impredecibles.

Los rangos de salidas esperados varían normalmente entre 0 y 0.4V para una salida baja y de 2.4 a 5V para una salida alta.

La diferencia entre los niveles de entrada y salida (2-2.4V y 0.8-0.4V) es proporcionarle al dispositivo inmunidad al ruido que se define como la insensibilidad del circuito digital a señales eléctricas no deseadas.

Para los CI CMOS una entrada alta puede variar de 0 a 3V y una alta de 7 a 10V (dependiendo del tipo de CI CMOS). Para las salidas los CI toman valores muy cercanos a los de VCC Y GND (Alrededor de los 0.05V de diferencia).

Este amplio margen entre los niveles de entrada y salida ofrece una inmunidad al ruido mucho mayor que la de los CI TTL.

VELOCIDAD DE OPERACIÓN

Cuando se presenta un cambio de estado en la entrada de un dispositivo digital, debido a su circuitería interna, este se demora un cierto tiempo antes de dar una respuesta a la salida. A este tiempo se le denomina retardo de propagación. Este retardo puede ser distinto en la transición de alto a bajo (H-L) y de bajo a alto (L-H).

La familia TTL se caracteriza por su alta velocidad (bajo retardo de propagación) mientras que la familia CMOS es de baja velocidad, sin embargo la subfamilia de CI CMOS HC de alta velocidad reduce considerablemente los retardos de propagación.

FAN-OUT O ABANICO DE SALIDA

Al interconectar dos dispositivos TTL (un excitador que proporciona la señal de entrada a una carga) fluye una corriente convencional entre ellos.

Cuando hay una salida baja en el excitador, este absorbe la corriente de la carga y cuando hay una salida alta en el excitador, la suministra. En este caso la corriente de absorción es mucho mayor a la corriente de suministro.

Estas corrientes determinan el fan-out que se puede definir como la cantidad de entradas que se pueden conectar a una sola salida, que para los CI's TTL es de aproximadamente de 10. Los CI's CMOS poseen corrientes de absorción y de suministro muy similares y su fan-out es mucho mas amplio que la de los CI's TTL. Aproximadamente 50.

CIRCUITOS INTEGRADOS TTL

Esta familia utiliza elementos que son comparables a los transistores bipolares diodos y resistores discretos, y es probablemente la mas utilizada. A raíz de las mejoras que se han realizado a los CI TTL, se han creado subfamilias las cuales podemos clasificarlas en:

1. TTL estándar.
2. TTL de baja potencia (L).
3. TTL Schottky de baja potencia (LS).
4. TTL Schottky (S).
5. TTL Schottky avanzada de baja potencia (ALS).
6. TTL Schottky avanzada (AS).

Como sus características de voltaje son las mismas (La familia lógica TTL trabaja normalmente a +5V), analizaremos sus velocidades y consumo de potencia.

Velocidad aproximada	Subfamilia TTL
1.5 ns	Schottky avanzada
3 ns	Schottky
4 ns	Schottky avanzada de baja potencia
10 ns	Schottky de baja potencia
10 ns	estándar
33 ns	baja potencia

Tabla 1: Velocidades de las distintas subfamilias TTL

Consumo de potencia por puerta	Subfamilia TTL
1 mW	baja potencia
1 mW	Schottky avanzada de baja potencia
2 mW	Schottky de baja potencia
7 mW	Schottky avanzada
10 mW	estándar
20 mW	Schottky

Tabla 2: Consumo de potencia de las subfamilias TTL

Observemos que las subfamilias Schottky de baja potencia como la Schottky avanzada de baja potencia reúnen excelentes características de alta velocidad y bajo consumo de potencia.

Debido a su configuración interna, las salidas de los dispositivos TTL NO pueden conectarse entre si a menos que estas salidas sean de colector abierto o de tres estados.

CIRCUITOS INTEGRADOS CMOS

Estos CI’s se caracterizan por su extremadamente bajo consumo de potencia, ya que se fabrican a partir de transistores MOSFET los cuales por su alta impedancia de entrada su consumo de potencia es mínimo.

Estos CI’s se pueden clasificar en tres subfamilias:

Familia	Rango de tensión	Consumo potencia	Velocidad
estándar (4000)	3 – 15 V	10 mW	20 a 300 ns
serie 74C00	3 – 15 V	10 mW	20 a 300 ns
serie 74HC00	3 – 15 V	10 mW	8 a 12 ns

Tabla 3: Subfamilias CMOS

La serie 74HCT00 se utiliza para realizar interfaces entre TTL y la serie 74HC00.

DESCARGAS ELECTROSTÁTICAS

Los dispositivos CMOS son muy susceptibles al daño por descargas electrostáticas entre un par de pines.

Estos daños pueden prevenirse:

1. Almacenando los CI CMOS en espumas conductoras especiales.
2. Usando soldadores alimentados por batería o conectando a tierra las puntas de los soldadores alimentados por ac.
3. Desconectando la alimentación cuando se vayan a quitar CI CMOS o se cambien conexiones en un circuito.
4. Asegurando que las señales de entrada no excedan las tensiones de la fuente de alimentación.
5. Desconectando las señales de entrada antes de las de alimentación.
6. No dejar entradas en estado flotante, es decir, conectarlos a la fuente o a tierra según se requiera.

MARCAS EN UN CI

Dependiendo del fabricante, un CI puede presentar distintas demarcaciones en la parte superior del mismo, pero una marca común en un CI TTL es como la que se describe a continuación:

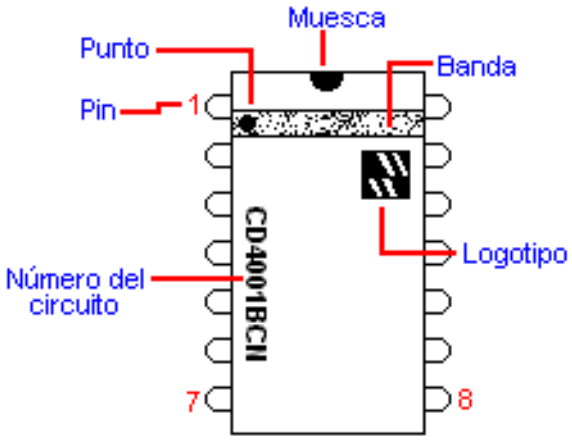


Figura 1: Marcas de un CI

El pin o patilla 1 se identifica con un punto, muesca o banda coloreada en uno de los extremos

del CI. Siempre se sitúa a la izquierda colocando el integrado con el extremo demarcado hacia arriba. El logotipo o el pequeño dibujo que identifica al fabricante puede aparecer en cualquiera de los dos extremos y el numero de circuito aparece generalmente centrado junto al costado izquierdo.

Un ejemplo de numero de circuito de un CI TTL puede ser el DM74ALS76N. Veamos como se decodifica este numero:

DM: Las primeras letras identifican al fabricante (National Semiconductor)

74: Los dos primeros números indican la serie (serie 7400)

ALS: Estas letras indican la subfamilia TTL (Schottky avanzada de baja potencia)

76: Los números siguientes especifican la función (doble flip-flop JK)

N: El sufijo N indica que es un CI encapsulado en doble linea

Para un CI CMOS las marcas son muy similares. Un ejemplo podría ser el MC74HC32N:

MC: Identifica al fabricante (Motorola)

74HC: Indica la subfamilia o serie del integrado (74HC00)

32: Especifica la función (4 puertas OR de dos entradas)

N: Este es el código de National Semiconductor para un CI DIP

INTERFACES ENTRE CI TTL Y CMOS

Ya que los requerimientos para estas dos familias son bastante diferentes, requieren para su interconexión la utilización de interfaces. A continuación hay algunos ejemplos de interfaces cuando los dispositivos trabajan con una misma fuente de voltaje y cuando trabajan con voltajes distintos. (gráficos de interfaces).

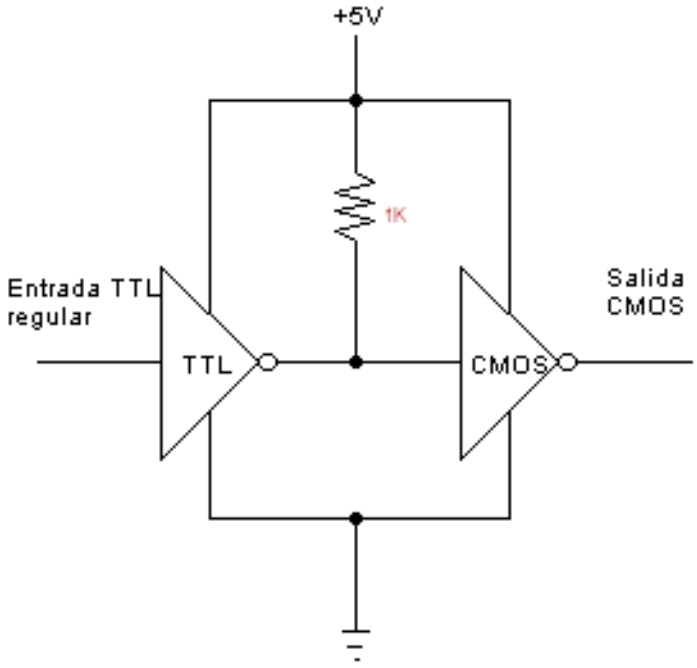


Figura 2: Interfaz estándar TTL a CMOS utilizando un resistor de "pull up"

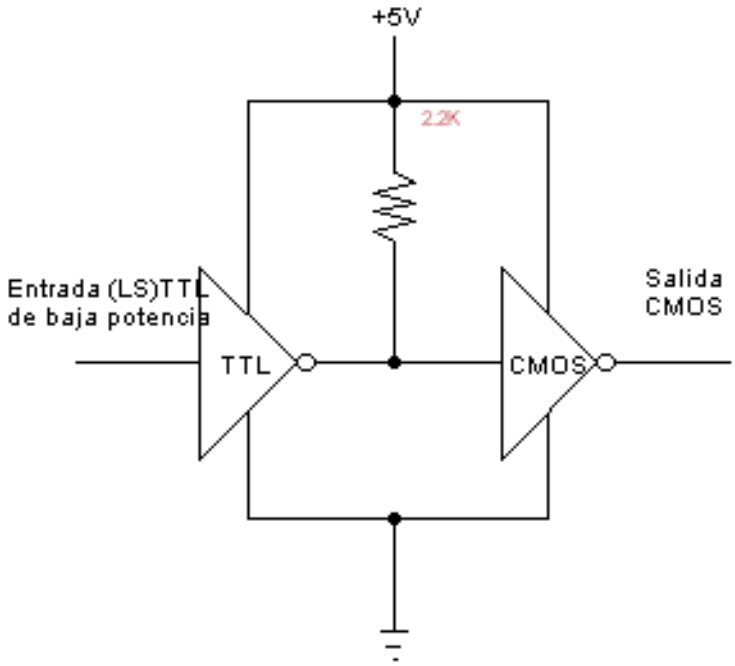


Figura 3: Interfaz Schottky TTL de baja potencia a CMOS utilizando un resistor de "pull up"

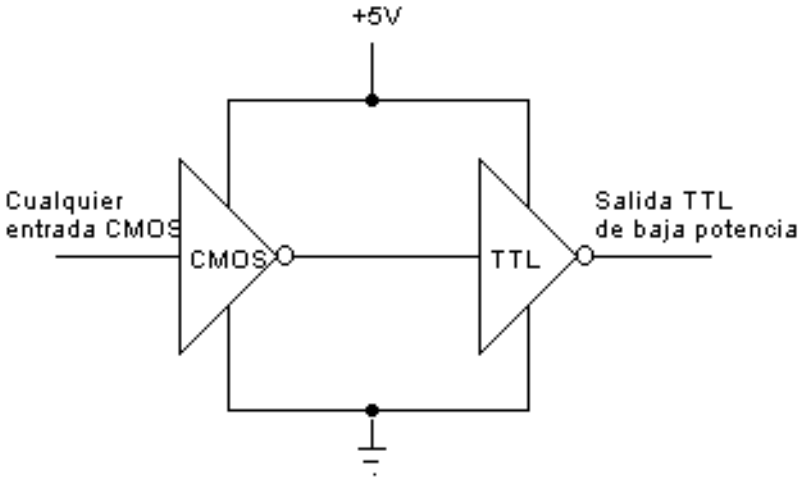


Figura 4: Interfaz CMOS a TTL Schottky de baja potencia

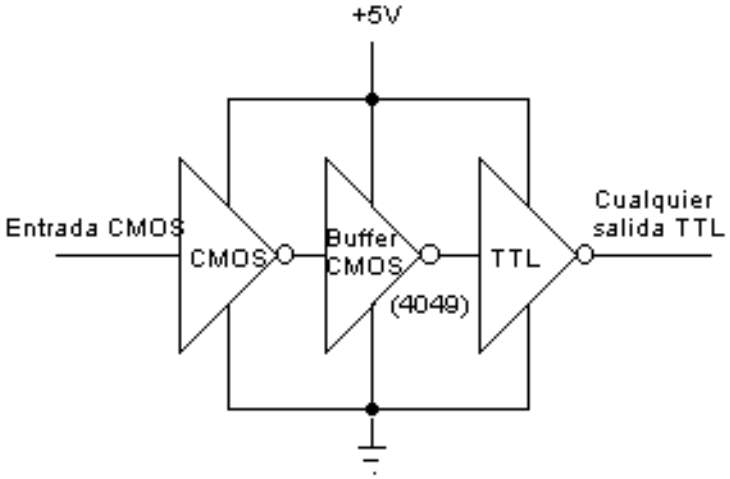


Figura 5: Interfaz CMOS a TTL estándar utilizando un buffer de CI CMOS

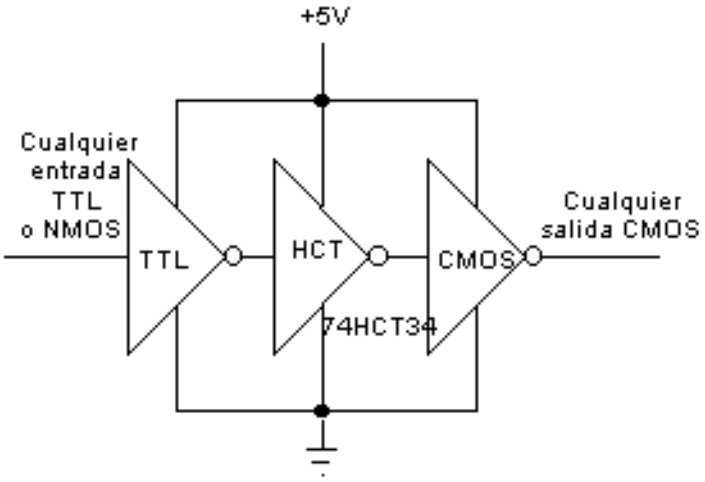


Figura 6: Interfaz TTL y CMOS usando un buffer de CI CMOS

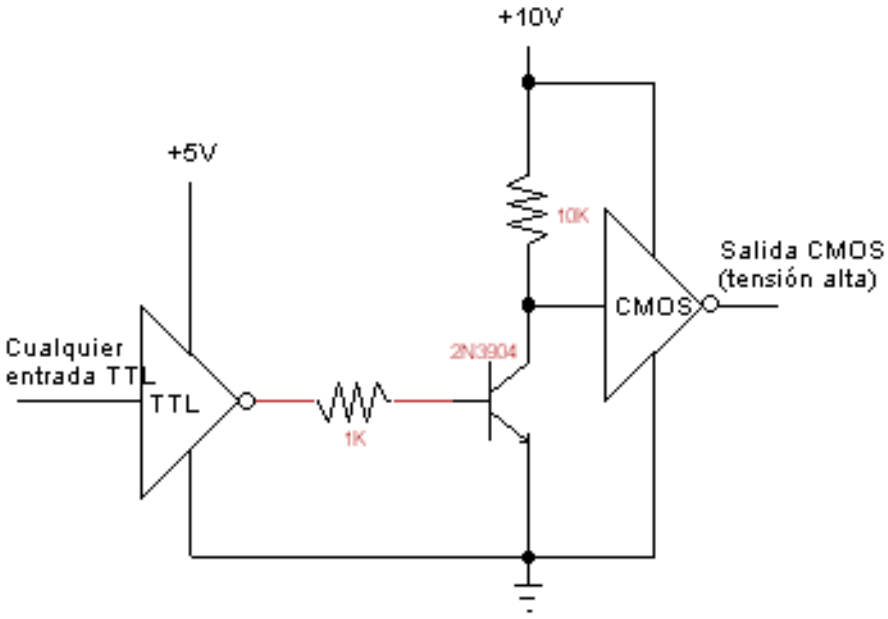


Figura 7: Interfaz TTL a CMOS utilizando un transistor

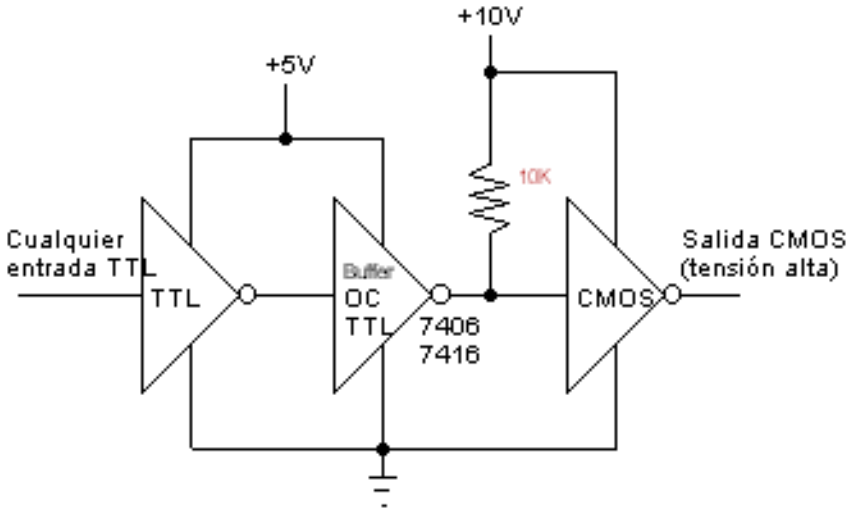


Figura 8: Interfaz TTL a CMOS utilizando un buffer TTL de colector abierto

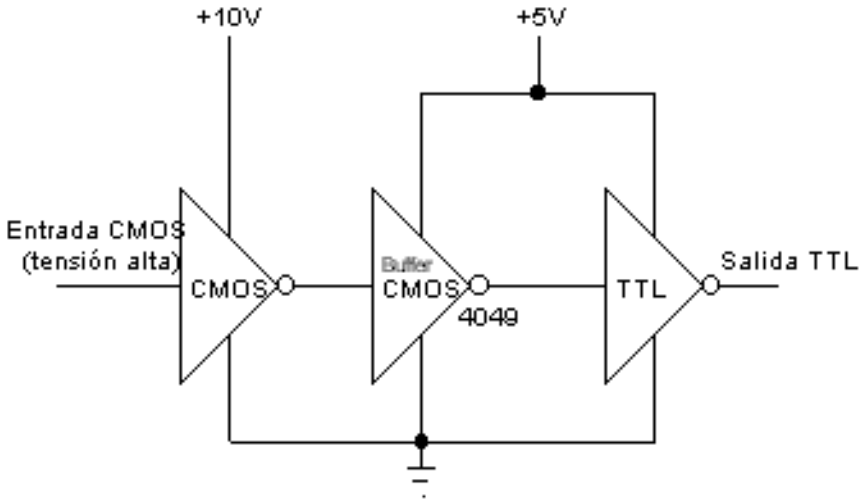


Figura 9: Interfaz CMOS a TTL utilizando un buffer de CI CMOS

Cuando las salidas de los CI’s se conectan a dispositivos distintos a puertas lógicas como por ejemplo a LED's indicadores, se pueden utilizar las interfaces siguientes:

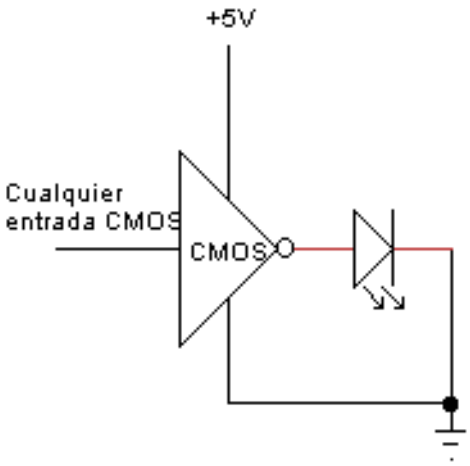


Figura 10: Interface CMOS a LED para voltaje de 5V. El led luce cuando hay salida ALTA

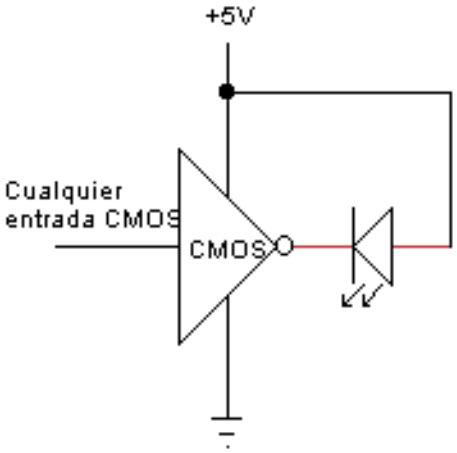


Figura 11: Interface CMOS a LED para voltaje de 5V. El led luce cuuando hay una selida BAJA

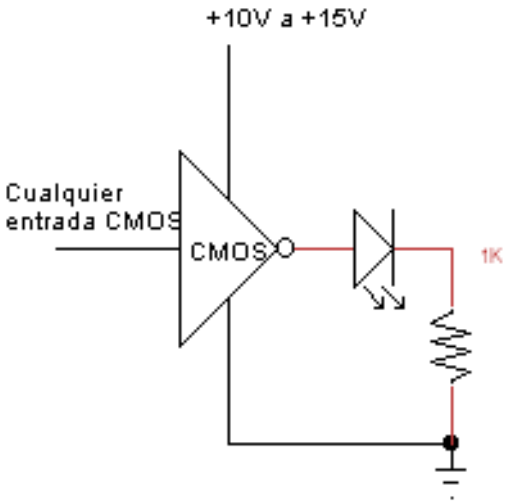


Figura 12: Interfaz CMOS a LED para un rango de tensión de 10 a 15V. El led luce cuando hay una salida ALTA

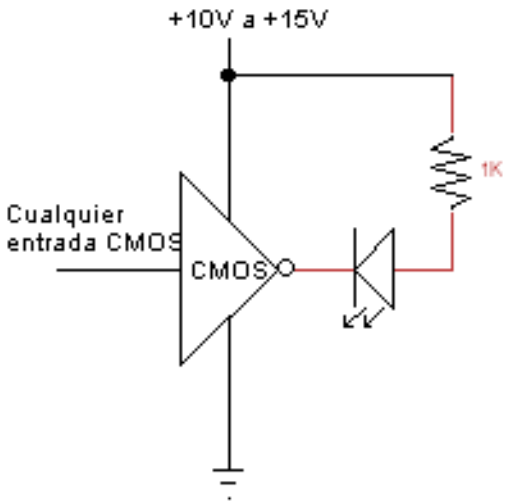


Figura 13: Interfaz CMOS a LED para un rango de tensión de 10 a 15V. El led luce cuando hay una salida BAJA

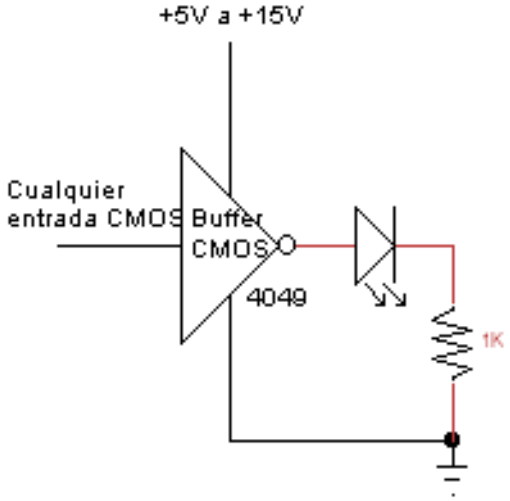


Figura 14: Interfaz buffer-inversor CMOS a LED para rango de tensión de 5V a 15V.

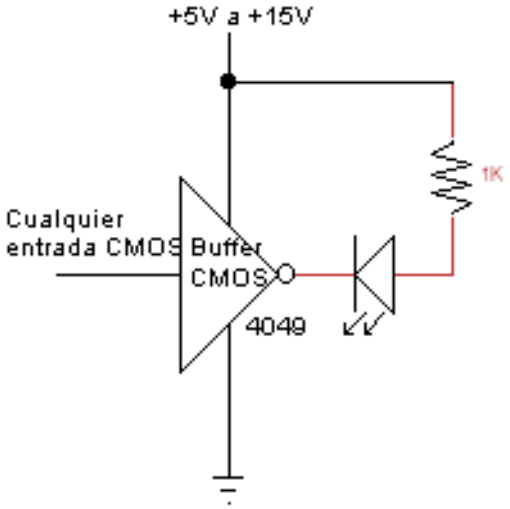


Figura 15: Interfaz buffer-no inversor CMOS a LED para un rango de tensión de 5V a 15V

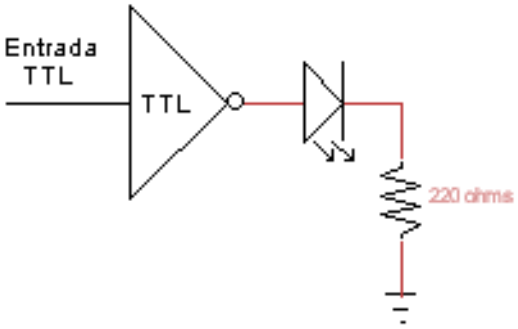


Figura 16: Interfaz TTL a LED el cual luce cuando la salida es ALTA

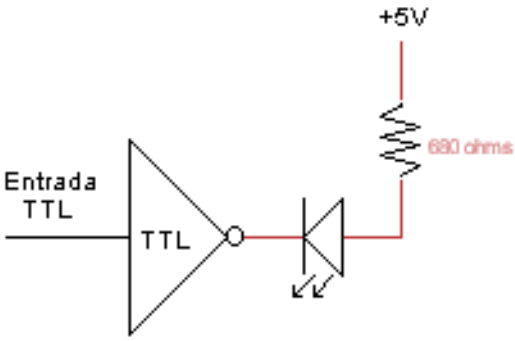


Figura 17: Interfaz TTL a LED el cual luce cuando la salida es BAJA

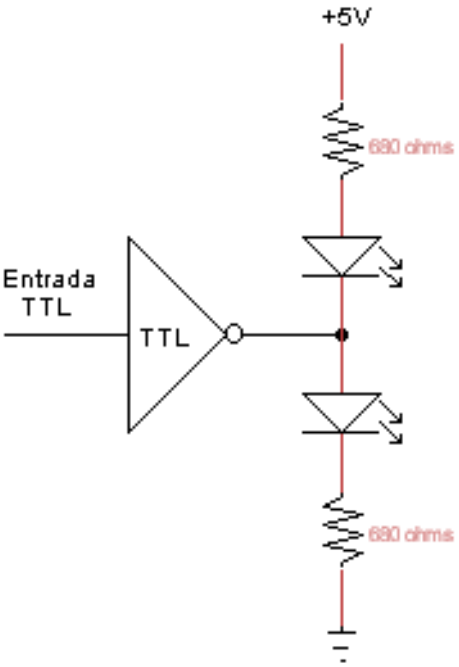


Figura 18: Interface TTL a LED con indicadores de salida ALTA y BAJA

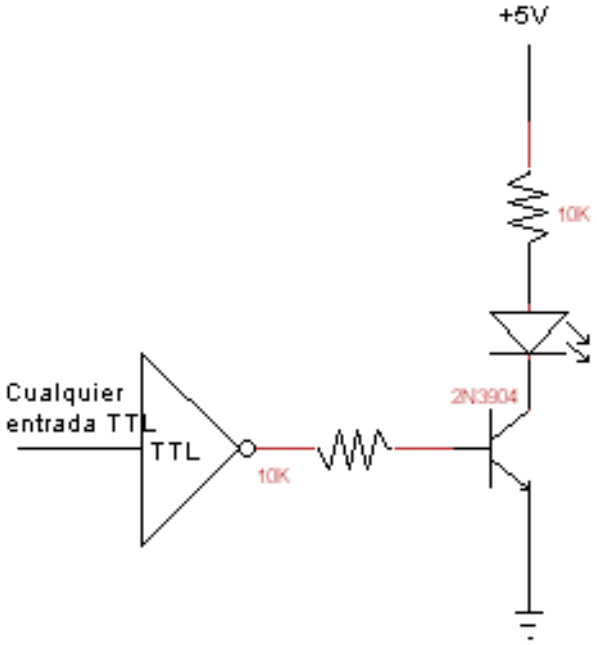


Figura 19: Interface TTL a LED utilizando un transistor

Generalmente, para introducir información a un circuito digital se utilizan los conmutadores o teclados. A continuación veremos los ejemplos clásicos de interfaces con conmutadores.

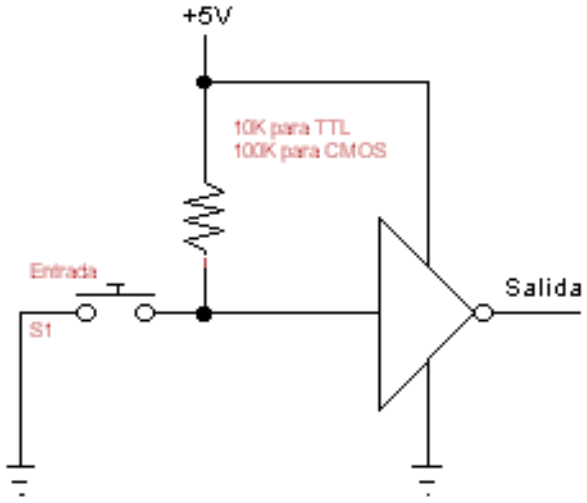


Figura 20: Interfaz de conmutador activo en BAJA

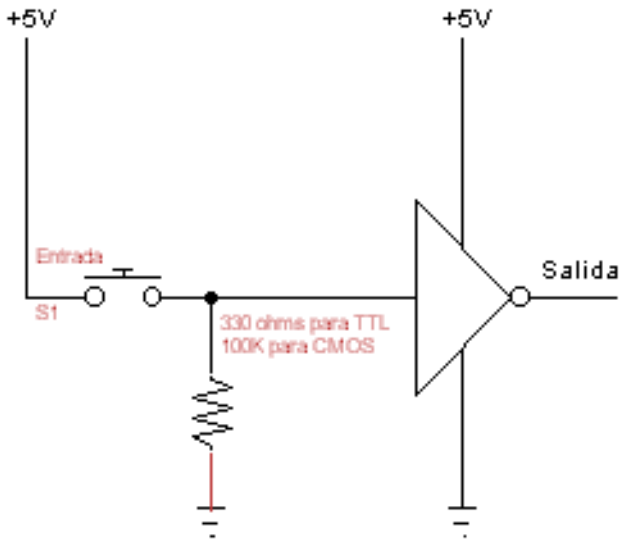


Figura 21: Interfaz de conmutador activo en ALTA

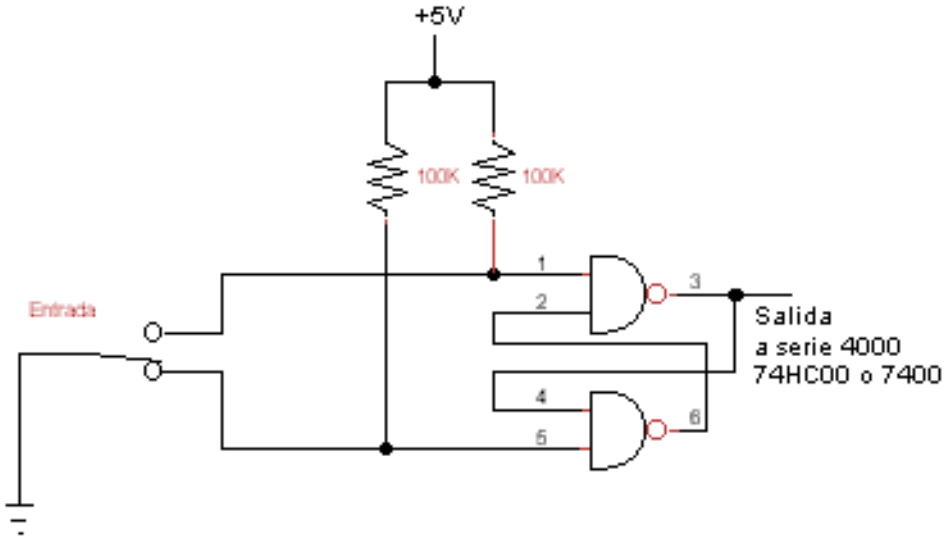


Figura 22: Circuito eliminador de rebote utilizando una compuerta NAND 74HC00 CMOS

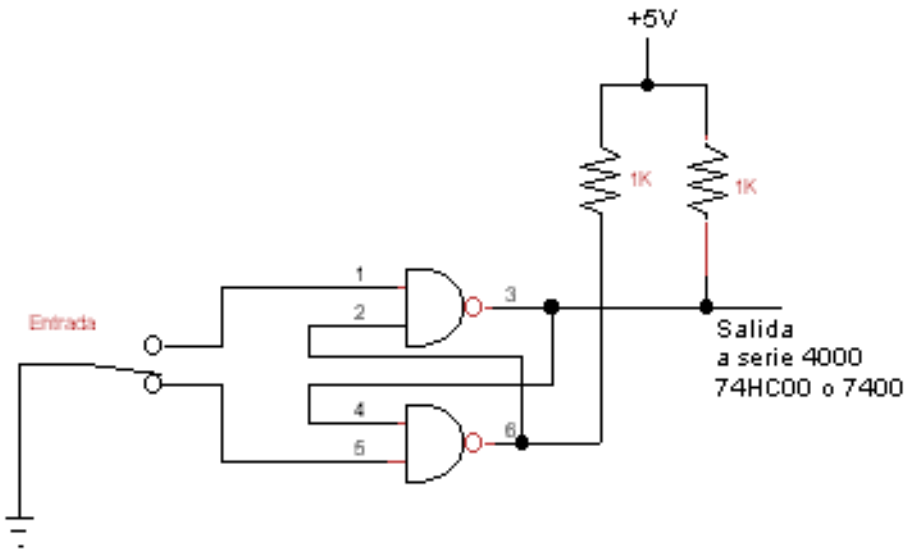


Figura 23: Circuito eliminador de rebote utilizando una compuerta 7403 TTL con colector abierto

Cuando un circuito digital debe activar dispositivos de salida (las cuales generalmente manejan una tensión mayor), se requiere el uso de las siguientes interfaces:

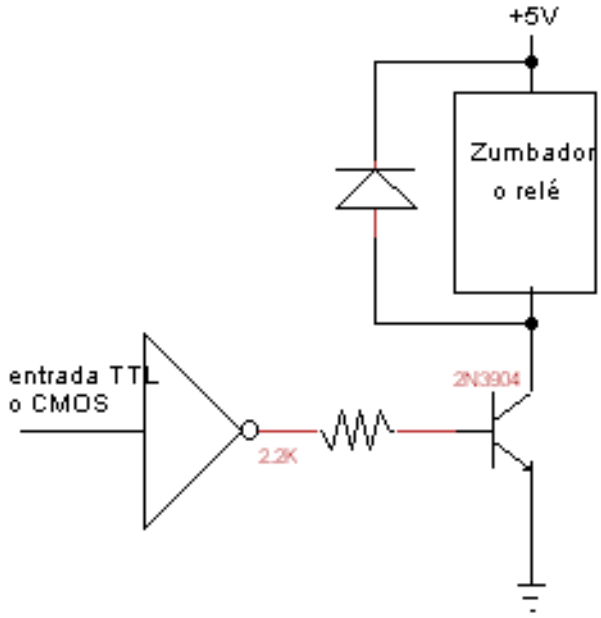


Figura 24: Interfaz con dispositivos de salida con inversor TTL o CMOS

CÓDIGOS BINARIOS

Debido a la naturaleza biestable de los circuitos de electrónica digital, estos solo procesan códigos que constan de 0 y 1 (códigos binarios) existen muchas situaciones en la electrónica digital en la que necesitamos realizar tareas especificas, por lo tanto se necesitaran utilizar una serie de códigos que también utilizan ceros (0) y unos (1), pero sus significados pueden variar. A continuación detallaremos estos tipos de códigos.

CÓDIGOS BINARIOS CON PESO

Supongamos que queramos transformar el numero decimal 89532 a su correspondiente equivalencia en binario, aplicando el método de la división sucesiva por dos, llegaremos al siguiente resultado: **1010111010110111100** pero para llegar a este resultado seguro te tomará cierto tiempo y trabajo, de igual forma si queremos diseñar un sencillo circuito digital en el que la cifra introducida en el teclado sea visualizada en la pantalla, se necesitarían una gran cantidad de compuertas lógicas para construir el circuito decodificado y codificador. Los códigos binarios con peso nos resuelven este problema pues estos códigos fueron diseñados para realizar la conversión de decimal a binario de una manera mucho mas fácil y rápida.

CÓDIGOS BCD

Los códigos BCD (Binary Coded Decimal) (Decimal Codificado en Binario) son grupos de 4 bits en el cual cada grupo de 4 bits solo puede representar a un único dígito decimal (del 0 al 9) Estos códigos son llamados códigos con peso ya que cada bit del grupo posee un peso o valor especifico. Existen por lo tanto códigos BCD's de acuerdo al valor o peso que posea cada bit. Ejemplos de estos códigos son el BCD 8421, el BCD 4221, el BCD 5421, el BCD 7421, el BCD 6311, etc. donde la parte numérica indica el peso o valor de cada bit. Así por ejemplo el código BCD 8421 nos indica que el MSB posee un valor de 8, el segundo MSB posee un valor de 4, el tercer MSB tiene un valor de 2 y el LSB tiene un valor de 1. Para el código BCD 6311 el MSB tiene un peso o valor de 6, el segundo MSB posee un peso de 3, el tercer MSB posee un valor de 1, y el LSB tiene un valor de 1. El código BCD 8421 es el código BCD mas utilizado, es común referenciarlo simplemente como código BCD, así en el transcurso del curso se entenderá el código BCD como el BCD 8421, a menos que se indique lo contrario.

CONVERSIÓN DE DECIMAL A BCD

Ya que cada grupo de 4 bits solo puede representar a un único dígito decimal, la conversión de un numero decimal a un numero BCD se lleva a cabo de la siguiente forma:

1. Separamos al dígito decimal en cada uno de sus dígitos
2. Cada dígito decimal se transforma a su equivalente BCD.
3. El número obtenido es el equivalente en BCD del número decimal.

Por ejemplo, para convertir el decimal 469 a BCD, según lo explicado anteriormente, tenemos que tomar cada dígito decimal y transformarlo a su equivalente BCD.

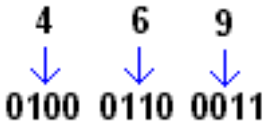


Figura 1: Conversión de decimal a BCD

De esta forma el decimal 469 equivale al BCD 010001100011

NOTA: En BCD los códigos 1010, 1011, 1100, 1101 y 1111 no tienen decimales equivalentes. Por lo tanto se les llaman códigos inválidos

CONVERSIÓN DECIMAL FRACCIONARIO A BCD

Se realiza del modo similar al anterior pero hay que tener en cuenta el punto binario, el punto del numero decimal se convertirá en el punto binario del código BCD.

Ejemplo: para convertir el decimal 74.42 a BCD:

Separamos el decimal en sus dígitos 7 4. 4 2.

Convertimos cada dígito a decimal a BCD, y colocamos el punto binario en la misma posición del punto decimal.

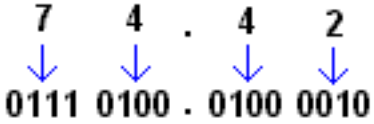


Figura 2: Conversión de decimal fraccionario a BCD

De esta forma el decimal 74.42 equivale al BCD 01100100. 010000101.

CONVERSIÓN DE BCD A DECIMAL

Ya que el código BCD son grupos de 4 bits, realizaremos lo siguiente:

1. A partir de la izquierda separamos al número BCD en grupos de 4 bits.
2. Cada grupo de 4 bits se convierte a su decimal correspondiente.
3. El número obtenido es el equivalente decimal del número BCD.

Ejemplo: Convertir el número BCD 010101000011 a decimal.

Separamos en grupos de 4 bits a partir de la izquierda 0101 0100 0011.

Transformamos cada grupo a decimal.

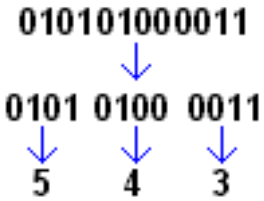


Figura 3: Conversión de BCD a decimal.

El BCD 010101000011 equivale al decimal 543

CONVERSIÓN BCD FRACCIONARIO A DECIMAL

1. A partir del punto binario separamos al número binario en grupos de 4 bits.
2. Cada grupo de 4 bits se convierte a su equivalente decimal.
3. El punto binario se convertirá en el punto decimal.

4. El número obtenido equivale en decimal al número BCD.

Ejemplo: Convertir el número BCD 01110001.0000100 a decimal.

separamos en grupo de 4 bits 0111 0001. 0000 1000.

convertimos cada grupo a decimal y colocamos el punto binario como punto decimal.

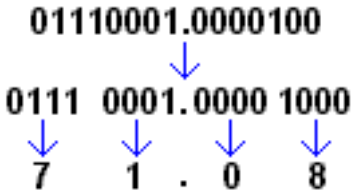


Figura 4: Conversión de BCD fraccionario a decimal.

El BCD 01110001.00001000 equivale al decimal 71.08

CONVERSIÓN BCD A BINARIO PURO

Si queremos transformar un numero BCD a su correspondiente binario llevaremos a cabo los siguientes pasos:

1. El número BCD lo transformamos a decimal.
2. Convertimos el decimal obtenido a binario mediante las técnicas ya estudiadas.
3. El binario obtenido es el equivalente en binario del número BCD.

Ejemplo: Convertir el BCD 000100000011.0101 a binario.

Convertimos 0001 0000 0011. 0101 a decimal 1 0 3. 5.

Transformamos el decimal a binario 103.5(10)=1100111.1

CONVERSIÓN DE BINARIO PURO A BCD

1. Convertimos el número binario a número decimal.
2. Cada dígito decimal se convierte a su equivalente BCD.
3. El numero obtenido es el equivalente BCD del número binario puro.

Ejemplo: convertir el binario 10001010.101 a BCD

Se convierte primero a decimal 10001010.101

```
128 + 8 + 2 + 0.5 + 0.125 = 138.625.
```

convertimos el decimal a BCD

```
138.625 = 0001 0011 1000. 0110 0010 0101
```

El binario 10001010.101 es igual al BCD 000100111000.011000100101

NOTA: Seguramente ha notado que los números en código BCD contienen mayor numero de bits que sus correspondientes números binarios, pero nuevamente recalcamos que esta desventaja es compensada por su facilidad para convertir a decimal.

CÓDIGOS BINARIOS SIN PESO

De la misma forma que existen códigos binarios con peso, también existen códigos binarios sin peso en el cual cada bit no va a poseer un valor o ponderación por posición. Aquí detallaremos dos códigos binarios sin peso: el de exceso 3 y el código Gray.

CÓDIGO DE EXCESO 3

A pesar de ser un código binario sin peso, el código de exceso 3 guarda una estrecha relación con el código BCD 8421 por el hecho de que cada grupo de 4 bits solo pueden representar a un único dígito decimal (del 0 al 9), y deriva su nombre de exceso 3 debido a que cada grupo de 4 bits equivale al número BCD 8421 mas 3.

CONVERSIÓN DE DECIMAL A EXCESO 3

1. Se separa al numero decimal en cada uno de sus dígitos.
2. Sumarle tres (3) a cada dígito decimal.
3. Convertir a BCD el número decimal obtenido.

4. El número obtenido es el equivalente en XS3 del número decimal.

Ejemplo: convertir el numero decimal 18 a su equivalente XS3.

Solución: primero le sumamos 3 a cada dígito.

```
   1       8
 + 3     + 3
-----   -----
   4      11
```

luego cada resultado se transforma a BCD

```
 4 = 0100

11 = 1001
```

Nota: En las conversiones de exceso 3 no se tiene en cuenta los códigos inválidos (1010, 1011, 1100, 1101, 1110, 1111) como vimos en el ejemplo anterior el número 11, el cual nos resultó de la suma de 8+3, se convirtió directamente al BCD 1001.

CONVERSIÓN BCD A XS3

Para convertir un número BCD a código de exceso 3 tenemos en cuenta los siguientes pasos:

1. A partir de la izquierda separamos al código BCD en grupos de 4 bits.
2. Sumamos 0011_2 a cada grupo de 4 bits.
3. El resultado es el equivalente en XS3 del código BCD.

Ejemplo: Convertir el BCD 00101001 a XS3

Separamos en grupos de bits. 0010 1001

Sumamos 0011_2 a cada grupo

```
  0010      1001
+ 0011     +0011
--------   -------
  0101      1100
```

El código XS3 01011100 equivale al BCD 00101001

CONVERSIÓN DE XS3 A DECIMAL

1. Dividimos a partir de la izquierda al número XS3 en grupos de 4 bits.
2. Convertimos a decimal cada grupo de 4 bits.
3. Restamos 3 a cada decimal.
4. El número obtenido es el equivalente decimal del número XS3.

Ejemplo : Convertir 10011010_{XS3} a decimal

Separamos en 4 bits 1001 1010
Convirtiendo a decimal 1001 1010

```
1001 = 9
1010 = 10
```

restamos 3 a cada resultado

```
  9     10
 -3     -3
----   ----
  6      7
```

el número 67_{10} equivale al XS3 10011010

CÓDIGO GRAY

Observemos lo siguiente:

El decimal 5 se representa en binario por 0101

El decimal 6 se representa en binario por 0110
¿Qué has notado?

Observa que con solo aumentar un nivel en la cuenta (del 5 al 6) dos bits cambiaron de estado (el tercer MSB y el LSB de ambos números), probablemente esto no signifique nada ni nos afectaría en lo mas mínimo sin embargo existen algunas situaciones en electrónica

digital en el cual solo necesitamos que al incrementarse la cuenta en un nivel solo cambie de estado (de 0 a 1 o viceversa) uno y únicamente un solo bit.

La solución esta en el código Gray, un código binario sin peso que no tiene ninguna relación con el código BCD.

Así para el ejemplo que hemos venido analizando:

el decimal 5 en binario es 0101 y en código Gray es 0 1 **1** 1

el decimal 6 en binario es 0110 y en código Gray es 0 1 **0** 1

el color azul indica el bit que cambió de estado.

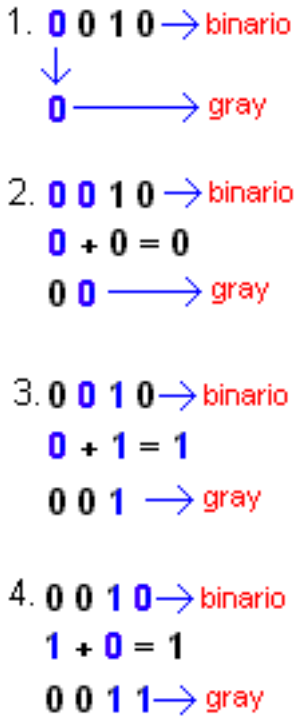
Pero, ¿cuales son los pasos que se deben llevar cabo para hacer la transformación a código Gray?

CONVERSIÓN DE NUMERO BINARIO A CÓDIGO GRAY

1. El MSB del numero binario será el mismo para el código Gray.
2. Sumar el MSB del numero binario al bit situado a su derecha inmediata y anotar el resultado del numero en código Gray que estamos formando.
3. Continuar sumando bits a los bits situados a la derecha y anotando las sumas; hasta llegar al LSB.
4. El número en código Gray tendrá el mismo número de bits que el número binario.

Ilustraremos mejor esta explicación con un ejemplo:

Ejemplo: convertir el numero binario 0010 a código Gray



Aquí finaliza la conversión dado que ya llegamos al LSB del numero binario.

Entonces el numero binario 0010 equivale al 0011 en código Gray

CONVERSIÓN DE CÓDIGO GRAY A BINARIO

1. El bit izquierdo de código Gray será el MSB del numero binario.
2. El bit obtenido es sumado al segundo bit de la izquierda del código Gray, y el resultado se anotara a la derecha del numero binario a formar.
3. Este resultado se le suma al bit situado a la derecha inmediata del ultimo bit que sumamos y el resultado será el otro bit del número binario (se ordena de izquierda a derecha).
4. Repetir el paso anterior hasta llegar al bit mas a la derecha del código Gray.
5. El número de bits del numero binario deberá coincidir con el número de bits del número en código Gray.

Ejemplo: convertir el número en código Gray 1001 a numero binario

1. 1 0 0 1 → gray

   1 ——→ binario

2. 1 0 0 1 → gray
   1 + 0 = 1
   1 1 ——→ binario

3. 1 0 0 1 → gray
   1 + 0 = 1
   1 1 1 → binario

4. 1 0 0 1 → gray
   1 + 0 = 1
   1 1 10 → binario

CÓDIGOS ALFANUMÉRICOS

Los códigos estudiados anteriormente sólo sirven para representar números, pero ; ¿y si queremos representar las letras del alfabeto o algunos símbolos? ; ¿cómo lo haríamos?.

La solución está en los códigos alfanuméricos, que no es más que un tipo de código diseñado especialmente para representar números, letras del alfabeto (mayúsculas y minúsculas), símbolos especiales, signos de puntuación y unos caracteres de control.

Un código alfanumérico muy popular y ampliamente utilizado, es el llamado código ASCII (American Standard Code for Information Interchange), que en español quiere decir: código estándar americano para el intercambio de información, el cual es un código de siete bits muy utilizado en los sistemas digitales avanzados (computadores, redes de transmisión de datos, etc.) para representar hasta 128 (2^7) piezas de información diferentes, incluyendo letras, números, signos de puntuación, instrucciones y caracteres especiales.

CONTADORES

Son circuitos digitales lógicos secuenciales de salida binaria o cuenta binaria, caracteristica de temporizacion y de memoria, por lo cual están constituidos a base de flip-flops.

CARACTERISTICAS IMPORTANTES

1. Un número máximo de cuentas (módulo del contador)
2. Cuenta ascendente o descendente.
3. Operación síncrona o asíncrona.
4. Autónomos o de autodetención.

UTILIDAD

Se utilizan para contar eventos.
Ejemplos:

1. número de pulsos de reloj.
2. medir frecuencias.
3. Se utilizan como divisores de frecuencia y para almacenar datos. Ejemplo: en un reloj digital.
4. Se utilizan para direccionamiento secuencial y algunos circuitos aritméticos.

CONTADORES DE RIZADO.

Son dispositivos contadores que tienen conectados los flip-flops en forma asincrona, es decir, que no,tienen conectadas las entradas de reloj (CLK) en paralelo, sino que tiene que esperar que el primer flip-flop, al activarse por el pulso conmute generando una salida, la cual active o coloque en modo de conmutacion al siguiente flip-flop, el cual con el siguiente pulso conmuta activando al siguiente y asi sucesivamente. El modo de conmutacion en los flip-flop se consigue colocando las entradas J y K en ALTA (1 logico).

El módulo de un contador está determinado por la cuenta máxima a la que es diseñado, es decir, si el contador es diseñado para que cuente de 0 a 15 su módulo es el 16 (contador módulo 16) y simplificado se denomina contador mod-16, si es diseñado para contar de 0 a 9 será un contador módulo 10 o mod-10, etc.

CONTADOR DE RIZADO MODULO- 16.

Este contador se encuentra constituido por flip-flop JK en modo de conmutación al mantener presente en las entradas J y K un 1 lógico y conectados entre si de forma asíncrona, es decir, que la salida del flip-flop 1 (FF1) está conectada de forma directa a la entrada de reloj del siguiente flip-flop 2 (FF2). Los indicadores de salida dan una señal binaria, donde el indicador A (QA) es el LSB (Bit Menos Significativo), el indicador D (QD) es el MSB (Bit Más Significativo).

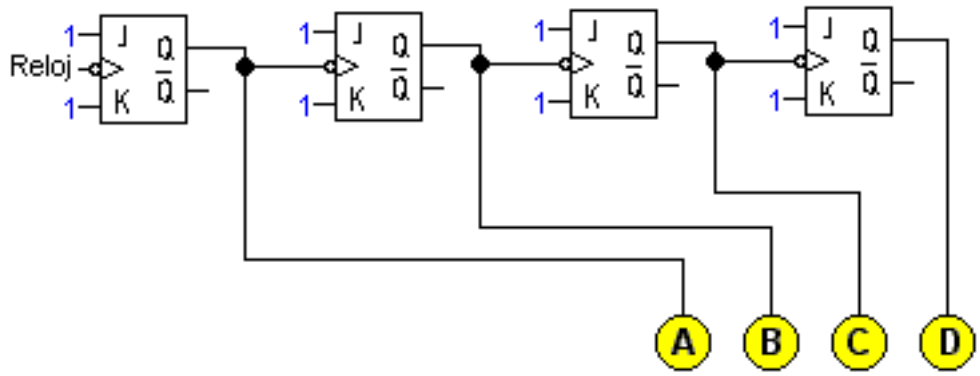


Figura 1: Contador de rizado mod-16

El circulito en la entrada de reloj (CLK) de los fip-flops, nos indica que trabajan o conmutan con lógica negativa, es decir, que se activan en la transición de ALTA a BAJA (flanco posterior) del pulso de reloj y la salida del FF1 (QA) va del nivel BAJO al ALTO dando como resultado la cuenta binaria 0001. En el pulso 2, en la transcicion del nivel ALTO a BAJO, FF1 se desactiva pasando su salida del nivel ALTO a BAJO, activando el FF2, conmutando la salida del nivel BAJO a ALTO generando la cuenta 0010, en el pulso 3 del reloj se activa FF1 generando la salida 0011, porque FF2 se encuentra en mantenimiento teniendo su salida (QB) activada, en el siguiente pulso se incrementa la cuenta a 0100, según se observa en el diagrama de tiempo de la figura 2.

Cuenta decimal	Cuenta binaria	Cuenta decimal	Cuenta binaria

0	0	0	0	0	8	1	0	0	0
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	10	1	0	1	0
3	0	0	1	1	11	1	0	1	1
4	0	1	0	0	12	1	1	0	0
5	0	1	0	1	13	1	1	0	1
6	0	1	1	0	14	1	1	1	0
7	0	1	1	1	15	1	1	1	1

Tabla 1: Secuencia de un contador mod-16

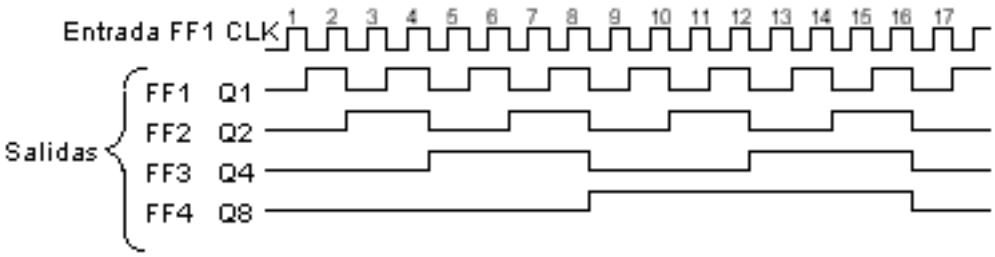


Figura 2: Diagrama de tiempos de un contador mod-16

CONTADORES PARALELOS

Con este tipo de contadores se elimina o se atenua el retardo que se presenta en los contadores asíncronos, donde se tiene que esperar que un flip-flop active al otro. Este efecto se consigue conectando el reloj directamente a las entradas de reloj (CLK) de los flip-flops, es decir, conectando los pulsos de reloj en paralelo (síncronamente) y las salidas de los flip-flops a las entradas J y K de los mismos.

CONTADOR PARALELO DE 3 BITS MOD-8.

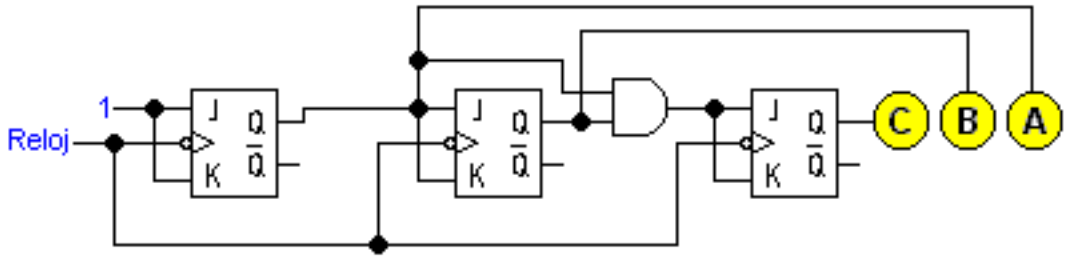


Figura 3: Contador paralelo mod-8

También está construido a base de flip-flops JK, los cuales tienen conectadas sus entradas de reloj en paralelo y sus salidas QA, va conectada a las entradas J y K del siguiente flip-flop (FF2) y asi sucesivamente por lo que estaría en modo de mantenimiento hasta que la salida del FF1 les de un 1 logico lo que los colocaría en modo de conmutacion a FF2, al estar las entradas del reloj en paralelo, la transición del primer pulso del nivel ALTO a BAJO, FF1 se activa mientras que FF2 se coloca en modo de conmutación y FF3 sigue en mantenimiento generando la cuenta 001. En el segundo pulso FF1 se desactiva y FF2 conmuta generando la salida 010, si en el tercer pulso estuviera la salida FF2 conectada directamente a las entradas J y K del FF3 se generaría la cuenta máxima 111, por que el FF2 se encuentra en estado de mantenimiento en este caso activado por el pulso anterior, teniendo en modo de conmutacion a FF3 el cual, junto con FF1 se activaría en el pulso 3. Para evitar este inconveniente se conecta la salida del FF1 y del FF2 a las entradas de una puerta AND y las salidas de la puerta AND a las entradas J y K de FF3, colocandolo en modo de conmutación solamente cuando FF1 y FF2 estén activados, es decir, en el pulso 3. Generando en el pulso 4 de reloj que se desactiven FF1 y FF2 y se active FF3 generando la cuenta 100 y en los siguientes pulsos se generarán. El resto de cuenta como se muestra en el diagrama de tiempo de la figura 4.

Cuenta Binaria	Cuenta decimal		

0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Tabla 2: Secuencia de un contador mod-8

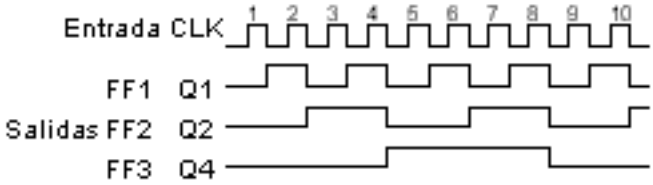


Figura 4: Diagrama de tiempos de un contador mod-8

OTROS CONTADORES.

Estos contadores no llegan a su cuenta máxima, por que se interrumpe su cuenta según el diseño o la necesidad que se tenga, por ejemplo, un contador MOD-6 o MOD-10.

CONTADOR DE RIZADO MOD-6.

Para conseguir este tipo de contador de bits, se utiliza una entrada de reset o borrado la cual se activa inmediatamente después de la cuenta más alta que se necesite, en este caso en la cuenta 110, colocando los flip-flops en 0 lógico. En la figura 5 se muestra el esquema de un contador mod-6.

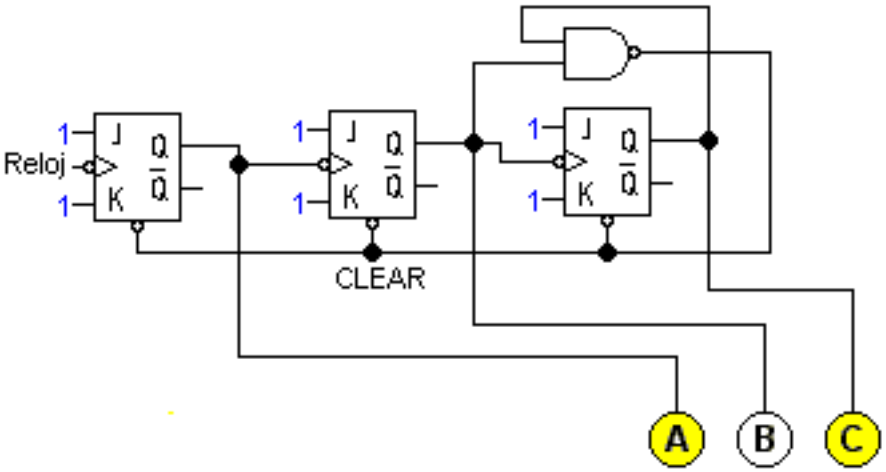


Figura 5: Esquema lógico de un contador de rizado mod-6

Este trabajo de activar las entradas de reset de cada flip-flop lo realiza una puerta NAND la cual da un 0 lógico a las entradas de reset. Al recibir en las entradas de la NAND los 1 lógicos de las salidas del FF2 y del FF3 colocándo en 0 lógico todos los flip-flops y así el contador comienza de nuevo a contar desde 000 hasta 101 o inversamente si es de cuenta descendente.

C	B	A	Cuenta decimal
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	RESET
1	1	1	

Tabla 3: Secuencia de un contador mod-6

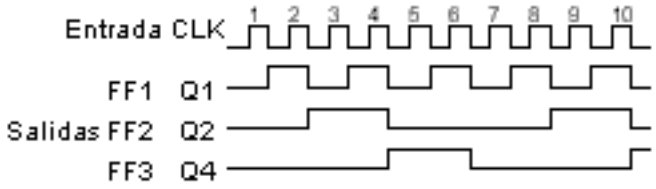


Figura 6: Diagrama de tiempos del contador mod-6

El retardo que es causado en el pulso 6 cuando va del nivel ALTO a BAJO hasta que FF2 y FF3 son puestos a 0 en el punto B del diagrama de tiempo, se le denomina tiempo de propagación y éste depende del retardo de propagación del flip-flop y de la puerta que se esté utilizando, este retardo de propagación en la familia TTL es del orden de unos 30ns (nanosegundos). En las otras familias son mayores.

CONTADOR DECADA (CUENTA DECENAS)

Es uno de los más utilizados, esta construido a base de 4 flip-flops JK y una puerta NAND la cual pone en 0 los flip-flops al llegar la cuenta máxima (1010). Como se sabe un contador de 4 bits, llega a una cuenta máxima binaria de 1111 que equivale a 16 en decimal, por lo que la puerta NAND debe activarse inmediatamente después de la cuenta 1001 0 9 en decimal para que el contador sea mod-10.

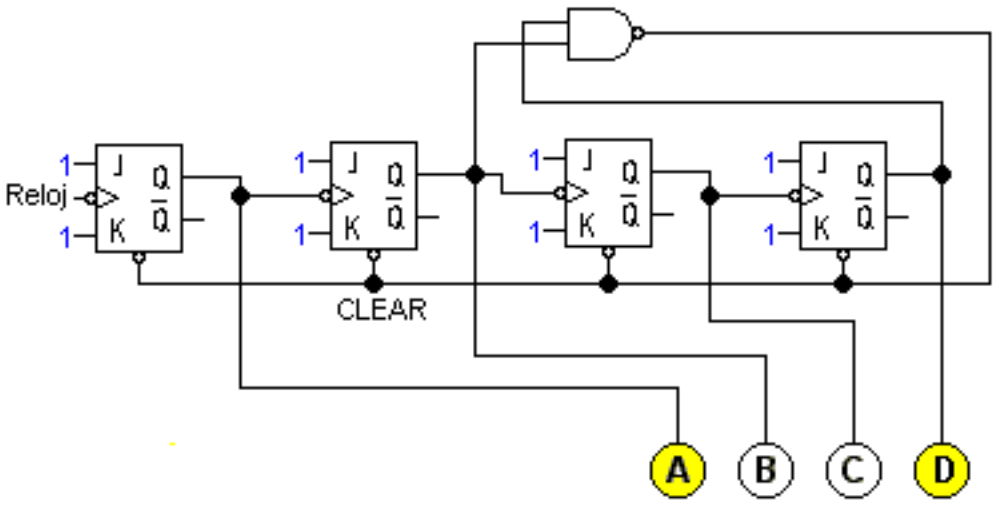


Figura 7: Diagrama lógico de un contador década rizado

Entonces, como la cuenta inmediatamente después de 1001 es 1010, entonces se conectan las entradas de la puerta NAND a las salidas de FF2 (QB) y FF4 (QD) que al mandar los unos a las entradas de la NAND, la activan enviando un pulso a las entradas de reset (borrado o CLR) de los flip-flops colocándolos en cero y reinicializando la cuenta.

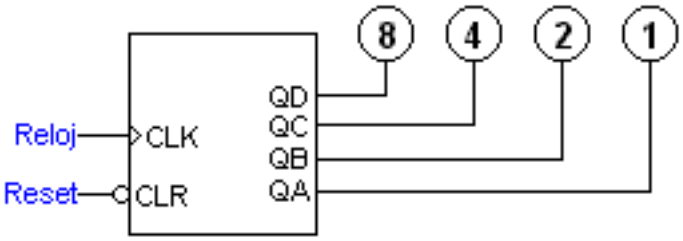


Figura 8: Símbolo lógico simplificado de un contador década

CONTADORES DESCENDENTES

Son los contadores en los cuales su cuenta va en sentido inverso a la normal, es decir, de 16 a 0 o en binario de 1111 a 0000. (si es de mod-16)

CONTADOR DE RIZADO DESCENDENTE DE 3 BITS

Esta diseñado similarmente a los demás contadores, con la diferencia que este trae en los flip-flops una salida negada (1), la cual da el pulso contrario a la salida normal (Q), es decir, cuando Q es positivo, 1 es negativo. Esta salida 1 es la que va a ir conectada a la entrada de reloj (CLR) de los otros flip-flops, de resto todas las conexiones son iguales como se muestra en la figura 9.

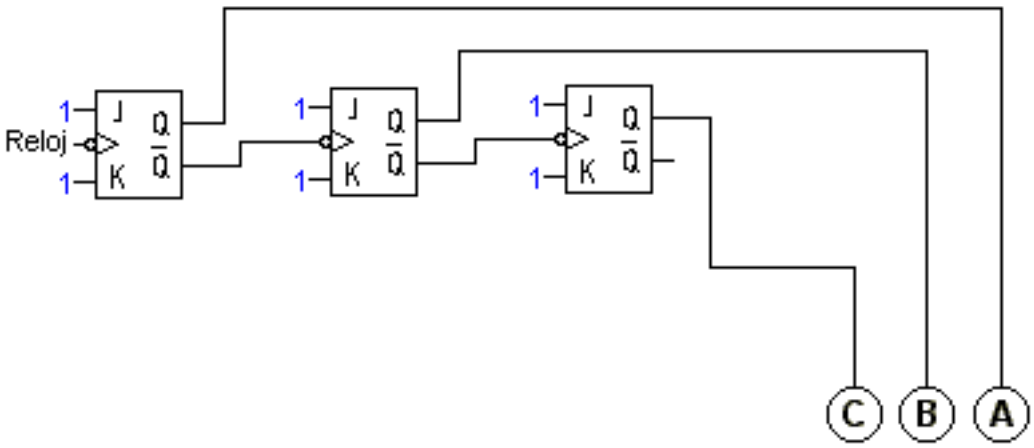


Figura 9: Contador de rizado descendente de 3 bits

El funcionamiento es el siguiente: al tener los 3 flip-flops sus entradas J y K en estado de conmutación (ambas entradas en ALTO) y sus salidas Q activadas o en estado de SET en los flip-flops, al llegar el primer pulso en la transición de ALTO a BAJO, el FF1 conmuta, con lo cual Q va del nivel ALTO a BAJO y 1 va del nivel BAJO al ALTO y la cuenta pasa de 111 a 110 (de 7 a 6 en decimal), en el pulso 2 en la transición de ALTO a BAJO, FF1 comuta con lo cual la salida Q va del nivel BAJO al ALTO y la salida 1 va del nivel BAJO al ALTO y se genera la cuenta 101 (5 en decimal) y así hasta llegar a la cuenta máxima, que en este caso es 0000 como se muestra en el diagrama de tiempo,

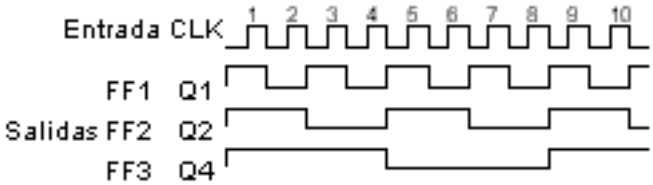


Figura 10: Diagrama de tiempos de un contador descendente de 3 bits

CONTADORES CON CI TTL

Son circuitos integrados donde vienen incluidos los flip-flops conectados según el tipo de contador y las puertas. Estos contadores se pueden llamar de propósito general. El CI 74192 es un contador reversible BCD síncrono TTL, es decir, módulo-10. Tiene doble entrada de reloj, una para cuenta ascendente y una para cuenta descendente que conmutan en la transición del nivel BAJO al nivel ALTO del pulso. La entrada de borrado síncrono se activa en nivel ALTO colocándo las salidas en nivel BAJO (0000) y se inicializa en cualquier número que se cargue en las entradas de datos en forma binaria y se

transfieren asíncronamente a la salida BCD (A=QA, B=QB, C=QC, D=QD). La salida de arrastre se utiliza para conectar en cascada serie varios contadores.

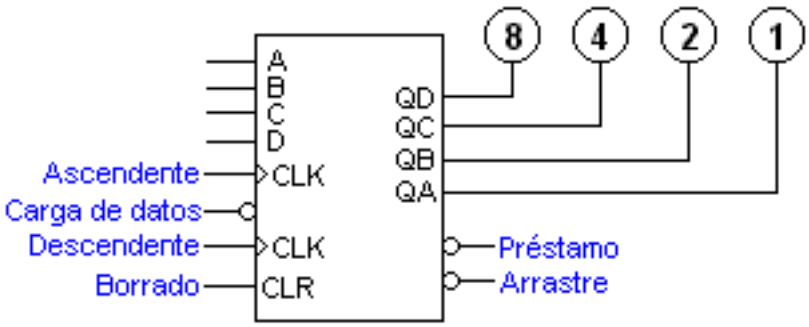


Figura 11: Símbolo del contador 74192

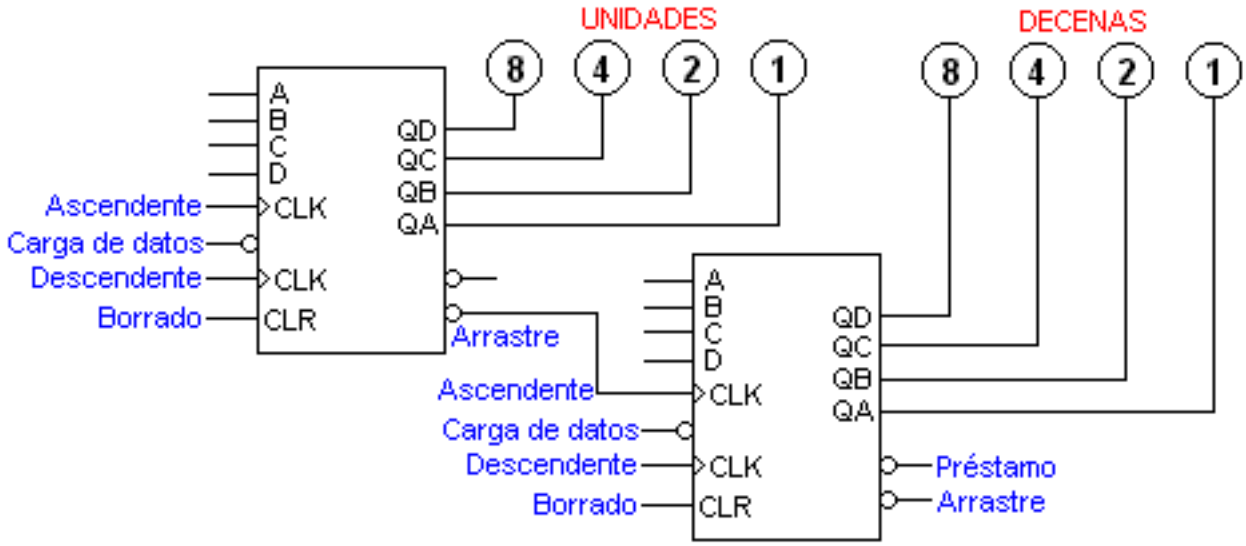


Figura 12: Conexión en cascada de dos contadores de 0 a 99

CONTADOR BINARIO DE 4 BITS TTL 7493.

El contador 7493 utilizan 4 flip-flops JK en modo de conmutación, con entradas de reloj ÇP0 y ÇP1 en donde ÇP1 es la entrada de reloj del segundo flip-flop por lo que para formar un contador de 4 bits mod-16 hay que conectar la salida del primer flip-flop de manera externa (puente) con la entrada ÇP1, quedando ÇP0 como la entrada de reloj del contador. También tiene dos entradas de reset (MR1 y MR2) las cuales no se deben dejar desconectadas (flotando) porque, como estas se activan en ALTA, al estar flotando toman un nivel ALTO lo que mantendría en reset al contador.

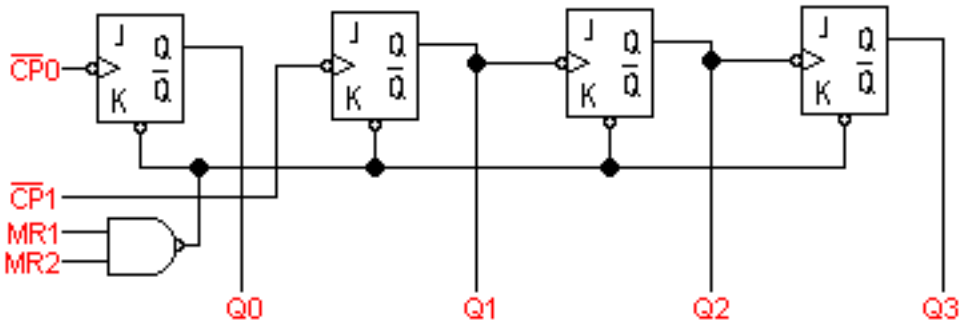


Figura 13: Contador 7493

CONTADORES CON CI CMOS.

CONTADOR CMOS 74HC393

El CI 74HC393 es un doble contador binario de 4 bits. Esta construido a base del flip-flop T. Las entradas de reloj (1ÇP y 2ÇP) son activadas por flanco posterior, o sea, en la transicion de ALTO a BAJO del pulso de reloj. Las entradas de reset (1MR y 2 MR) del maestro en el contador se activan en nivel ALTO, las salidas se etiquetan desde Q0 a Q3, siendo Q0 el LSB y, Q3 el MSB del numero binario de 4 bits. Requiere una fuente de alimentacion de 5V DC y viene en un CI DIP de14 patillas.

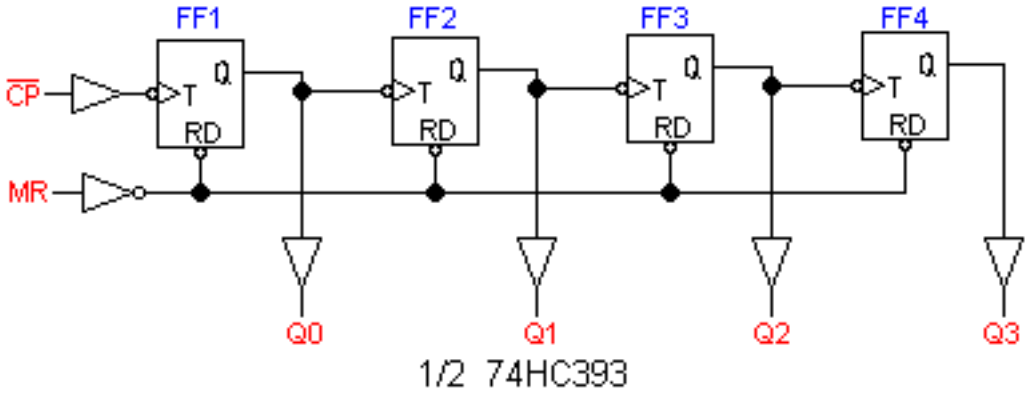


Figura 14: Diagrama lógico del contador CMOS 74HC393

CONTADOR CMOS CI 74HC193

El CI 74HC193 es un contador reversible síncrono de 4 bits preinicializable como lo muestra la hoja de datos.

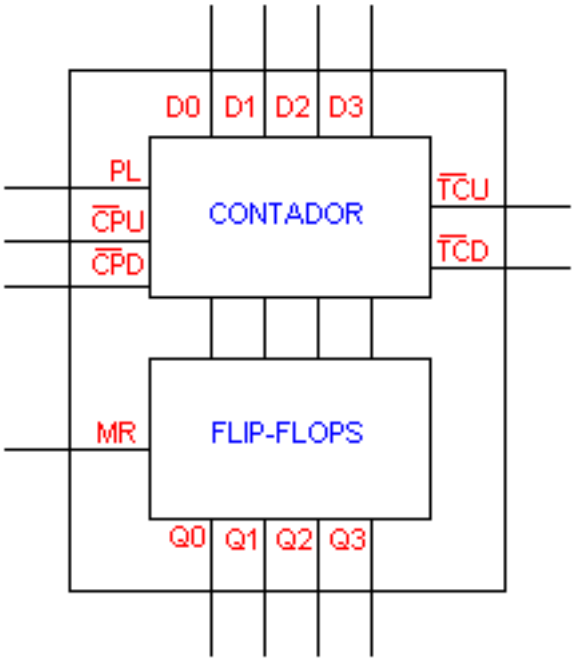


Figura 16: Contador CMOS 74HC193

Tiene 2 entradas de reloj (CPU y CPD), que se activan en la transición del nivel BAJO al ALTO del pulso de reloj, la entrada CPU es para la cuenta ascendente (UP) y la entrada CPD es para la cuenta descendente (D), por lo que dependiendo si el contador que se necesite se conecta al nivel alto o +5V. Los modos de operacion del contador CMOS 74HC193 se muestran en la tabla de verdad 5. El modo de reset borra asíncronamente las salidas (Q0 a Q3) al binario 0000 activándose en ALTO el cual puede ser un pulso de corta duración. Las entradas de carga de datos en paralelo (D0 a D3) se utilizan para programar un número en binario desde donde se quiere que empieze a contar de nuevo al activar la entrada de carga en paralelo (P)) con un nivel BAJO y los datos son transferidos asíncronamente a las salidas (Q0 a Q3). Las salidas de arrastre TÇ5 y TÇĐ generan un pulso negativo, para la conexión en cascada de contadores, ya sea en forma ascendente o en forma descendente la cuenta de estos. El contador 74HC193 viene en un DIP de 16 patillas y opera con una tensión de alimentacion de +5V DC.

DIVISION DE FRECUENCIA: EL RELOJ DIGITAL.

En un contador digital de salida binaria el retraso que se forma al activarse cada flip-flop a determinado pulso de reloj, en realidad es una division de frecuencia, por ejemplo, en un contador de 4 bits la salida QA divide la frecuencia en 2 porque necesita un pulso para activarse y otro para desactivarse, la salida QB divide en 4 la frecuencia del reloj de entrada

porque tiene que esperar que pasen los 2 pulsos en la salida QA para poder activarse y otros 2 pulsos para desactivarse, la salida QC es una salida que divide por 8 y la salida QD divide por 16.

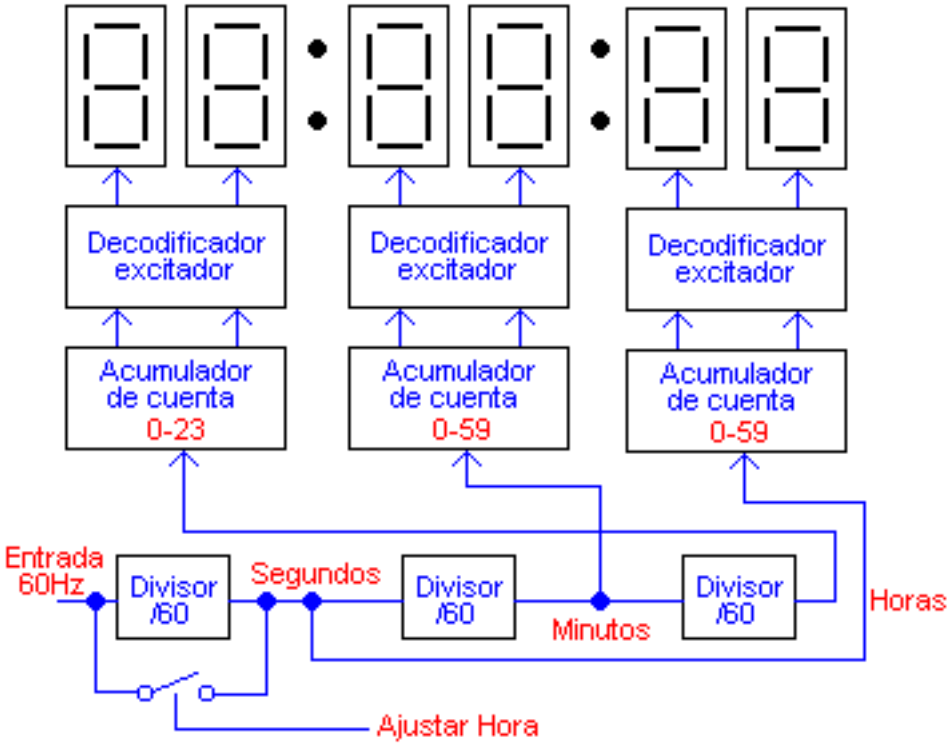


Figura 20: Diagrama de bloques de un reloj digital.

El reloj digital utiliza los contadores como divisores de frecuencia y acomuladores de cuenta. La función del contador como acumulador de cuentas es contar los pulsos de entrada y sirve como memoria temporalmente mientras muestra la hora actual que es decodificada y pasada a los visualizadores de hora. Los contadores como divisores de frecuencia tienen en su entrada una onda cuadrada de 60 Hz, el bloque divisor por 60, es construido por un contador divisor por 6, conectado a un contador divisor por 10.

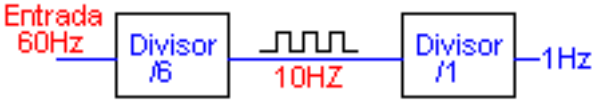


Figura 21: Contador divisor por 60

El contador divisor por 6 transforma los 60 Hz en 10 hz y el contador divisor por 10 transforma los 10Hz en 1Hz o 1 pulso por segundo. El contador divisor por 10 es construido con un CI 7493 por lo que la primera conexion que se debe hacer es un puente entre Q0 y ÇP1 para convertirlo en un contador de 4 bits, en segundo lugar el CI debe convertirse en un

contador decadal (mod-10) como se explico anteriormente, conectando Q3 y Q1 a las 2 entradas de reset. El contador divisor por 6 es hecho con un CI 7493 conectando la entrada de reloj a ÇP1, es decir, que el primer flip-flop (entrada ÇP0) no se utiliza. Los acumuladores de cuenta de 0 a 59 son 2 contadores en donde uno es un contador mod-10 para acumular las unidades (0 al 9) de los segundos y el otro es un contador mod-6 que recibe el pulso de arrastre del mod-10 para contar las decenas de los segundos. Los decodificadores/excitadores sirven para decodificar la salida BCD al visualizador de 7 segmentos.

CONVERSION ANALOGA/DIGITAL

Una cantidad digital tiene un valor que se especifica por una de dos posibilidades, mientras que una cantidad análoga puede tener posibilidades infinitas. Las cantidades digitales tienen la ventaja sobre las análogas de que se pueden modificar fácilmente sin perder exactitud, pero en el mundo real practicamente todas las señales son de carácter análogo, por eso existen los dispositivos convertidores A/D y D/A, que se encargan de tomar señales análogas del mundo exterior, convertirlas a digitales para poder tratarlas con exactitud, y finalmente volverlas a convertir en análogas ya modificadas y corregidas.

CONVERSIÓN DIGITAL ANÁLOGA

Recordemos que una señal digital es aquella que tiene solamente 2 niveles discretos de tensión, y una señal analógica es aquella que varia continuamente desde un valor mínimo hasta un valor máximo de tensión ó corriente.
Con frecuencia los equipos digitales deben conectarse (mediante una interfaz) con equipos analógicos. Esta interfaz o codificador especial que hace posible esta conexión es lo que llamamos conversor digital analógico (D/A).
Como hemos notado la tarea de este conversor es tomar una señal digital y transformarla en una señal analógica equivalente, estos dispositivos son mas sencillos que el conversor análogo digital (conversor A/D) que se estudiara mas adelante.

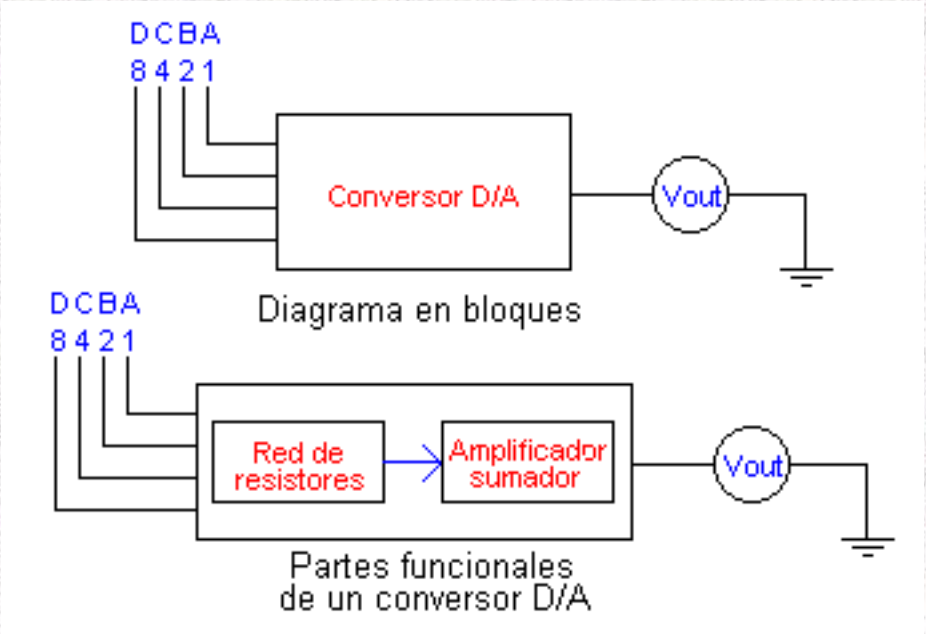


figura 1: Conversor D/A

Hay que observar que las entradas binarias del conversor D/A tiene asignado un peso de la siguiente forma:

La entrada A tiene un peso de 1.
La entrada B tiene un peso de 2.
La entrada C tiene un peso de 4.
La entrada D tiene un peso de 8.

Un conversor D/A esta dividido en 2 circuitos o partes funcionales:

1. Una red de resistencias.
2. Un amplificador sumador.

El incremento en la tensión de salida del conversor D/A se presenta por la acción que tienen las resistencias de las entradas (red de resistores) sobre la resistencia de realimentación del circuito amplificador. Esto lo detallaremos mas adelante.
La tarea de la red de resistores es asignar adecuadamente pesos a la entrada del conversor D/A. Es común que encontremos un amplificador operacional (am-op) tipo CI, conectado como amplificador sumador; la función de esta parte del conversor D/A es graduar o ajustar la tensión analógica de salida de acuerdo con la tabla de verdad, teniendo en cuenta obviamente el peso de las entradas binarias

AMPLIFICADOR OPERACIONAL

Las características mas importantes de un amp-op son :

1. Alta impedancia de entrada
2. Baja impedancia de salida.
3. Ganancia de tensión (Av) variable (depende del valor de las resistencia externas).

El símbolo esquemático de forma triangular para un amp-op se muestra en la figura 2. Las 2 entradas están etiquetadas con un (+) y con un (-); la entrada (-) se denomina la entrada inversora y la (+) se denomina la entrada no inversora; la salida se muestra en la parte derecha del símbolo. El amp-op requiere de 2 fuentes de alimentación de CC y se ubican en la parte superior e inferior del símbolo. La ganancia de tensión del amp-op (Av) puede. determinarse o fijarse por el valor de las resistencias externas

Rin (resistor input) resistencia de entrada
Rf (resistor feedback) resistencia de realimentación

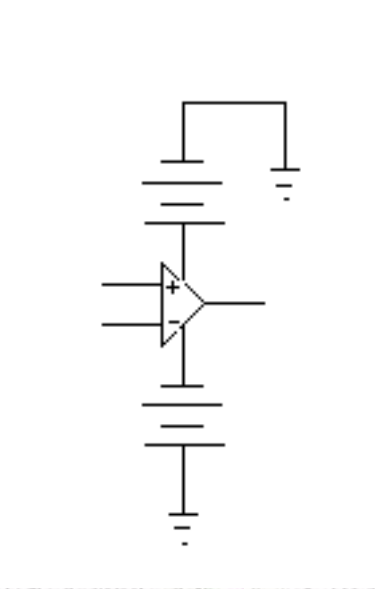


Figura 2: Símbolo esquemático de un amplificador operacional

Los valores de la resistencias (Rf y Rin) determinan la ganancia de tensión Av del circuito amplificador. La ganancia de tensión se calcula utilizando la formula:

```
Av = Rf/Rin
```

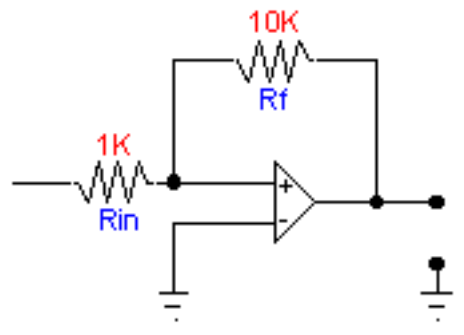


Figura 3: Amplificador Sumador

La ganancia de tensión se calcula simplemente dividiendo el valor de la resistencia de realimentación entre el valor de la resistencia de entrada. Los valores de las 2 resistencia mostrada en la figura 3 son:

```
Rf  = 10K
Rin = 1K
```

utilizando la formula de la ganancia de tensión tendremos:

```
      Rf    10000
Av = --- = ----- = 10
      Rin    1000
```

La ganancia de tensión en este caso fue 10. Con una ganancia de tensión de 10 v si aplicamos 1 v a la entrada obtenemos 10 v a la salida. Es decir

```
AV = 10 ----> Si Vin = 1v, entonces Vout = 10v.
```

la ganancia de tensión de un amp-op puede obtenerse teniendo en cuenta los voltajes de entrada y salida de acuerdo con la siguiente formula :

```
Av = Vout/Vin
```

Suponiendo una tensión de entrada de 1v y una tensión de salida de 10v, la ganancia de tensión será :

```
      Vout   10
Av = ---- = -- = 10
      Vin    1
```

La ganancia de tensión puede cambiarse fácilmente, cambiando la relación entre los valores de la resistencia de entrada y la resistencia de realimentación

CONVERSOR D/A BÁSICO

Un conversor digital-analogico básico aparece en la figura 4 como habíamos anotado el conversor D/A esta dividido en 2 circuitos:
La red de resistencia y el amplificador sumador. La tensión de entrada (Vin) se aplica a través de los conmutadores de entrada (D, C, B, A), en la parte superior del esquema. La tensión de salida analógica (Vout) se mide con un voltímetro. a la derecha. La tensión de entrada Vin es de 3V, y la salida varia de acuerdo a la tabla de verdad 1. Observar los valores de las resistencias en la red de resistencias. La resistencia que corresponde al MSB (o sea R4) es la resistencia de valor mas bajo. La resistencia R3, o sea, la resistencia con peso 4 es dos veces la resistencia R4. También la resistencia R2, es decir la entrada cuyo peso es 2, es 2 veces la resistencia de R3 y así sucesivamente. Para que un conversor D/A se preciso hay que tener en cuenta lo siguiente:

1. Los valores de resistencia deben ser bastantes precisos.
2. La tensión de alimentación también debe ser precisa.

Línea	D	C	B	A	Vout

1	0	0	0	0	0
2	0	0	0	1	0.4
3	0	0	1	0	0.8
4	0	0	1	1	1.2
5	0	1	0	0	1.6
6	0	1	0	1	2.0
7	0	1	1	0	2.4
8	0	1	1	1	2.8
9	1	0	0	0	3.2
10	1	0	0	1	3.6
11	1	0	1	0	4.0
12	1	0	1	1	4.4
13	1	1	0	0	4.8
14	1	1	0	1	5.2
15	1	1	1	0	5.6
16	1	1	1	1	6.0

Tabla 1: Tabla de verdad de un conversor D/A

Suponer que un conversor D/A opera de acuerdo con la tabla de verdad 1. Observar que la tensión analógica (Vout) aumenta gradualmente de 0 a 6 V. Cada aumento en la cuenta binaria incrementa la tensión analógica en 0.4 V por ejemplo, cuando la cuenta binaria pasa del 0001 al 0010, la tensión de salida analógica aumenta de 0.4 a 0.8 V. Considerar la situación de la figura1,donde la entrada binaria es 0000 (linea 1 en la tabla de verdad de la figura 1. Todos los conmutadores están a tierra (GND), VIN = 0 Y por tanto Vout = 0v. Ahora considerar activado solamente el conmutador de la entrada A.

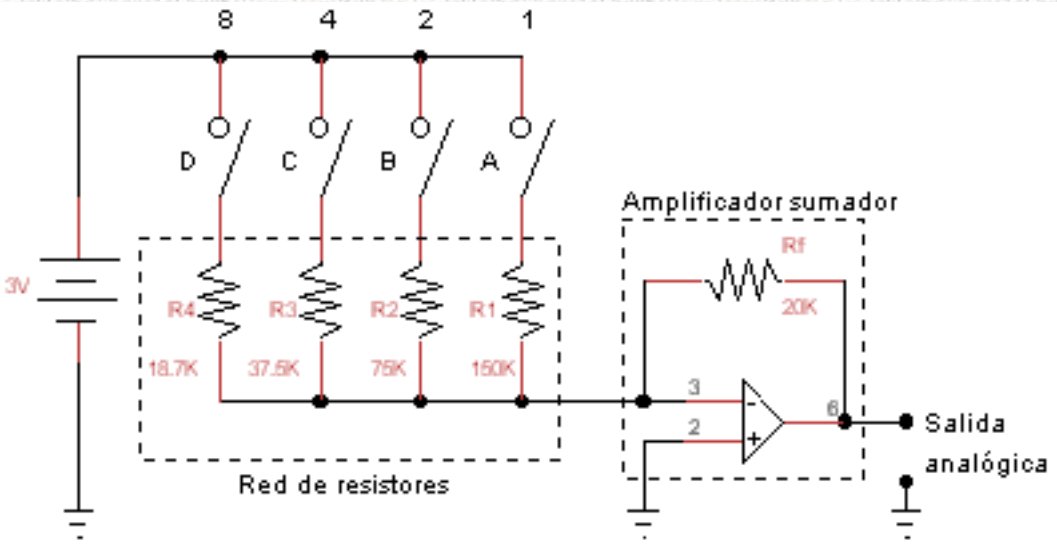


Figura 4: Diagrama esquemático del circuito conversor D/A

Esto corresponde a la linea 2 de la tabla de verdad 1. Primero calcular la ganancia del amp-op como sigue, observando que el valor de la resistencia de realimentación Rf es de 20 K y el de la resistencia de entrada R1 es de 150K

$$Av = \frac{Rf}{Ri} = \frac{20000}{15000} = 0.133$$

La ganancia de tensión del amp-op es de 0.1333 cuando se activa el conmutador ahora se calcula la tensión de salida (Vout)

$$Vout = Vin \cdot Av = 3 \cdot 0.133 = 0.4\ V$$

La tensión calculada para la salida (Vout) del amp-op del conversor D/A de la figura 4 cuando solamente esta activada el conmutador A es de 0.4V esto satisface los requerimientos de la tabla de verdad (linea 2) de la figura 4. A continuación considerar activado solamente l conmutador B (entrada binaria 0010 figura 4) esto corresponde a la linea 3 de la tabla de verdad. Primero debe calcularse la ganancia de tensión del amp-op

$$Av = \frac{Rf}{Ri} = \frac{20K}{75K} = 0.276$$

La ganancia del amp-op es de 0.276 con una resistencia de entrada Rin=75K y Rf =20K

A continuación se calcula la salida de tensión del conversor D/A (Vout).

$$Vout = Vin \cdot Av = 3 \cdot 0.276 = 0.8V$$

Suponer que solamente se activa el conmutador C (entrada binaria 0100) figura 1. la ganancia de tensión del amp-op se calcula.

$$Av = \frac{Rf}{Rin} = \frac{20000}{37500} = 0.533$$

La ganancia del amp-op es de 0.533 cuando Rf= 20K y Rin =37.5K A continuación se calcula la tensión de salida Vout del amp-op.

$$Vout = Vin \cdot Av = 3 \cdot 0.533 = 1.6V$$

Esto satisface las especificaciones de la tabla de la verdad de la linea 5 de la figura 1 Observar la linea 7 de la tabla de verdad para el conversor D/A figura 1. La entrada binaria es 0110. Se activan 2 conmutadores de entrada C y B Colocando a R3 y a R2 en paralelo formando la resistencia de entrada (Rin) esta debe calcularse de acuerdo con la formula que ya conocemos.

$$Rt = \frac{R3 \cdot R2}{R3+R2} = \frac{37.5K \cdot 75K}{37.5K+75K} = 25K$$

Se calcula la ganancia del amp-op

$$Av = \frac{Rf}{Rin} = \frac{20K}{25K} = 0.8$$

La tensión analógica de salida Vout del conversor D/A se calcula como

$$Vout = Vin \cdot Av = 3 \cdot 0.8 = 2.4V$$

Esto satisface los requerimientos de la linea 7 de la tabla de verdad de la figura 1

Considerar la linea 16 de la tabla de verdad de la tabla de verdad de la figura xxx para el conversor D/A. La entrada binaria es 1111 todos los conmutadores están activos poniendo en paralelo la resistencia R4, R3, R2 y R1. Se calcula el valor de Rin utilizando la formula de la resistencia en paralelo:

$$Rin = \frac{1}{\frac{1}{\quad} = \frac{1}{\quad}}$$

$$1/R4 + 1/R3 + 1/R2 + 1/R1 \quad 1/18.7K + 1/37.5K + 1/75K + 1/150K$$

$$= \frac{1}{0.00005 + 0.00003 + 0.00001 + 0.00006} = \frac{1}{0.0001} = 10.000$$

El valor de Rin es por lo tanto 10K. La ganancia de tensión Av del amp-op puede calcularse como:

$$Av = \frac{Rf}{Rin} = \frac{20K}{10K} = 2K$$

Ahora se calcula la tensión de salida del amp-op como:

$$Vout = Vin \cdot Av = 3 \cdot 2 = 6V$$

Esto satisface los requerimientos de la tabla de la verdad de la figura xxx.

Para cambiar el escalamiento en la salida solo basta con cambiar el valor de la resistencia de realimentación; por ejemplo, si en este caso cambiamos el valor de Rf de 20K a 10K nos dará incrementos mas finos de tensión de salida.

CONVERSOR D/A TIPO ESCALERA

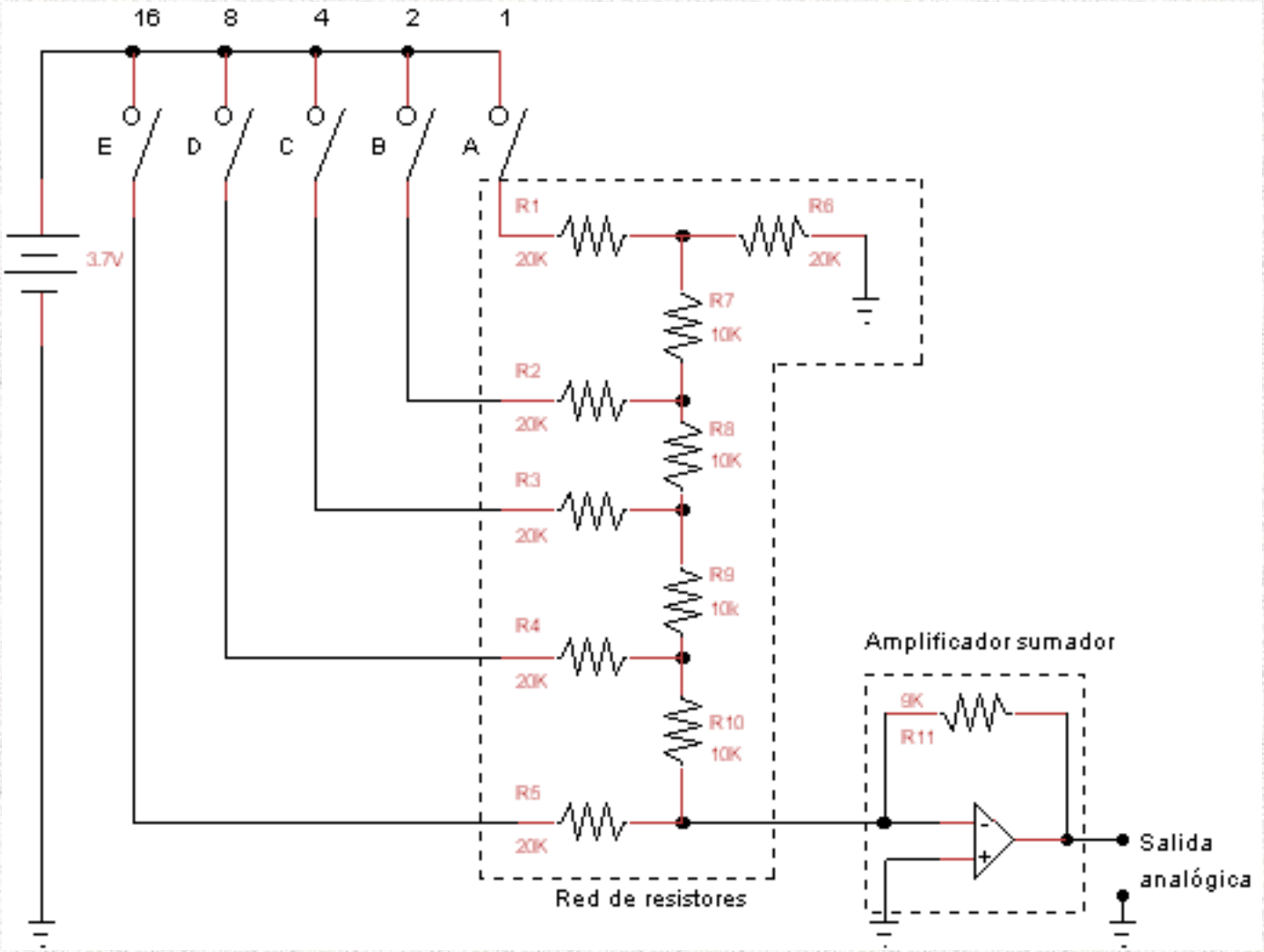


Figura 5: Conversor D/A de 5 bits con red de resistencias R-2R.

E	D	C	B	A	Vout
0	0	0	0	0	0
0	0	0	0	1	0.1
0	0	0	1	0	0.2
0	0	0	1	1	0.3
0	0	1	0	0	0.4
0	0	1	0	1	0.5
0	0	1	1	0	0.6
0	0	1	1	1	0.7
0	1	0	0	0	0.8
0	1	0	0	1	0.9
0	1	0	1	0	1.0
0	1	0	1	1	1.1
0	1	1	0	0	1.2
0	1	1	0	1	1.3
0	1	1	1	0	1.4
0	1	1	1	1	1.5

1	0	0	0	0	1.6
1	0	0	0	1	1.7
1	0	0	1	0	1.8
1	0	0	1	1	1.9
1	0	1	0	0	2.0
1	0	1	0	1	2.1
1	0	1	1	0	2.2
1	0	1	1	1	2.3
1	1	0	0	0	2.4
1	1	0	0	1	2.5
1	1	0	1	0	2.6
1	1	0	1	1	2.7
1	1	1	0	0	2.8
1	1	1	0	1	2.9
1	1	1	1	0	3.0
1	1	1	1	1	3.1

Tabla 2: Tabla de verdad para un conversor D/A de 5 bits

Este conversor consta de un amplificador sumador y una red de resistencias diferente al utilizado por el conversor anterior que se denomina R-2R ya que solo necesita 2 valores de resistencias, uno doble (2R) del otro (R) de aquí su nombre; su función es la misma que la red de resistencia anterior. En este tipo de conversores la precisión depende en gran medida de la precisión de la fuente de alimentación. Para este propósito los fabricantes disponen de referencia especiales de tensiones de precisión. El conversor que tomaremos como ejemplo es un conversor D/A de 5 bits, posee una tensión de entrada de 3.7V. figura 5. El resistor de realimentación R11 del amplificador sumador es de 9K. Este valor se seleccionó y/o calculo para producir una tensión de salida a escala completa (es decir, con todos los conmutadores cerrados a + 3.7V) de 3.1V. La tabla de verdad para este circuito se da en la tabla 2. Observar que cada incremento en la cuenta binaria hará incrementar la tensión de salida en 0.1V. El conversor D/A de nuestro ejemplo se dice que tiene una resolución de 5 bits, esto significa que tiene 32 posibilidades de salida (2^5=32). En el conversor D/A anterior la resolución era de 4 bits; en la mayoría de los casos los usuarios prefieren conversores con mas resolución para obtener incrementos mas finos en la tensión de salida. La resolución de un conversor D/A es una característica importante, ella viene dada por el numero de entradas o por el porcentaje a escala completa. Por ejemplo el conversor D/A de 4 bits tendrá su incremento de salida mas pequeño igual a 1 parte de 16. Al utilizar la formula, la resolución en tanto por ciento puede calcularse como:

$$\text{porcentaje de resolución} = \frac{1}{n} \cdot 100 = \frac{1}{16-1} \cdot 100 = \frac{1}{15} \cdot 100 = 1.7\%$$

en este caso 2^n = 2^4 = 16, donde n es el numero de bits de entradas. El resultado significa que, para cada aumento en la cuenta binaria, la tensión de salida (Vout) del conversor D/A cambia el 6.7 por ciento de la máxima tensión de salida.

$$\text{Así: } V_{out(max)} = 6.0\ V$$

$$\text{incremento} = \frac{6.0 \cdot 6.7}{100} = 0.4$$

que es en efecto el incremento que teníamos.

La resolución para el conversor D/A de 5 bits se calcula entonces

$$\text{porcentaje de resolución} = \frac{1}{2^n - 1} \cdot 100 = \frac{1}{2^n - 1} \cdot 100 = \frac{1}{31} \cdot 100 = 3.2\ \%$$

El conversor D/A de 5 bits tiene una resolución de 3.2%. El porcentaje es inferior hace que el conversor de 5 bits sea mejor para la mayoría de los trabajos que el conversor D/A de 4 bits. El conversor de nuestro ejemplo, puede cambiarse para que tenga una mejor resolución añadiendo otro conmutador de entrada F, una resistencia vertical de 10K y una resistencia horizontal de 20K debajo de R5. La conexión al amp-op vendría del extremo derecho, de la parte inferior izquierda, de la resistencia de 20K en la escalera R-2R. Otros factores a considerar a la hora de comprar conversores D/A son la precisión y velocidad de operación, o tiempo de respuesta

CONVERSIÓN ANALOGICA/DIGITAL

La tecnología digital tiene muchas ventajas sobre la tecnología analógica, ya que los sistemas digitales son más fáciles de diseñar, tienen mayor exactitud y precisión, alta inmunidad al ruido, entre otras, pero, sin embargo cuando se emplean técnicas digitales existe, en realidad solo una limitante: El mundo real es fundamentalmente analógico. La mayor parte de las cantidades físicas son de naturaleza analógica, y a menudo estas cantidades son las entradas y salidas de un sistema que las monitorea, que efectúa operaciones con ellas y que las controla. Algunos ejemplos son la temperatura, la presión, la posición, la velocidad, el nivel de un líquido, y muchas mas. Cuando se tienen entradas y salidas analógicas, deben seguirse tres pasos para aprovechar las técnicas digitales:

1. Convertir las entradas analógicas del "mundo real" a la forma digital.
2. Procesar (realizar operaciones con) la información digital.
3. Convertir de nuevo las salidas digitales a la forma analógica del mundo real.

Un método para convertir una señal análoga a digital es mediante el llamado ADC (Analog Digital Converter) de rampa digital el cual es uno de los métodos más sencillos de conversión que emplea un contador binario como registro y permite que el reloj incremente el estado del contador un paso a la vez que Vax Š Va. Este tipo de convertidor recibe el nombre de ADC de rampa digital debido a que la forma de onda en Vax (salida del conversor D/A) es una rampa (en realidad una escalera) como la que se muestra en la figura xx.B. Este tipo de conversión también se conoce con el nombre de ADC tipo contador. La figura xx.A es el diagrama de un ADC de rampa digital. Como se observa, este contiene un contador, un DAC, un comparador analógico y una compuerta AND de control. La salida del comparador también proporciona la señal de fin de conversión activa en BAJO, FDC. Si se supone que Va, el voltaje analógico de entrada al convertidor, es positivo, la operación del mismo es la siguiente:

1. Se aplica el pulso INICIO para poner el contador en cero. El estado ALTO de INICIO también inhibe el paso de los pulsos de reloj por la compuerta AND y de aquí hacia el contador.
2. Cuando las entradas del DAC son todas cero (0), la salida de este es Vax=0v
3. Dado que Va > Vax, la salida del comparador, FDC es ALTO.
4. Cuando INICIO regresa al estado BAJO, se habilita la compuerta AND y los pulsos de reloj entonces pasan hacia el contador.
5. A medida que cambia de estado el contador, la salida del DAC, Vax, aumenta un paso a la vez, como lo muestra la figura xx.B
6. Este proceso continua hasta Vax alcanza un paso que excede a Va por una cantidad igual o mayor que Vt (por lo general de 10 a 100µV). En ese momento FDC cambia hacia el estado BAJO e inhibe el flujo de pulsos hacia el contador, motivo por el cual este deja de contar.
7. El proceso de conversión está terminado, lo que es señalado por la transición de ALTO hacia BAJO de la señal FDC; el contenido del contador es la representación digital de Va.
8. El contador retiene el valor digital hasta que el siguiente pulso INICIO da comienzo otra vez al proceso de conversión.

Ejemplo: Supóngase que el ADC de la figura xx.A tiene las siguientes características: .frecuencia de reloj = 1 MHz. .Vt = 0.1mV .La salida del DAC a escala completa es de 10.23 V. .Una entrada de 10 bits. Determinar lo siguiente:

1. El equivalente digital obtenido para Va = 3.728V
2. El tiempo real de conversión.
3. La resolución del convertidor.

solución:

1. El DAC tiene una entrada de 10 bits y una salida fs de 10,23V. Por lo tanto, el numero de posibles pasos totales es de 2^10-1=1023, de manera que el tamaño de paso es: 10,23V ------ = 10 mV 1023 Esto significa que Vax crece en pasos de 10 mV cuando el contador cuenta hacia arriba desde cero (0). Ya que Va = 3.728V y Vt = 0.1mV, entonces Vax tiene que llegar a 3.7821 o mas antes que el comparador cambie a BAJO. Esto requerirá 3.7821V ------- = 372.81 = 373 pasos 10 mV Al termino de la conversión, por lo tanto, el contador contendrá el equivalente binario de 373, que es 0101110101. Este es el equivalente digital deseado de Va = 3.728V, como lo produce este convertidor A/D.
2. Se necesitaron 373 pasos para completar la conversión en consecuencia, ocurrieron 373 pulsos de reloj a razón de uno por microsegundo. Esto hace un tiempo real de conversión de 373 µS.
3. La resolución de este convertidor es igual al tamaño de paso del convertidor D/A que es 10 mV. En porcentaje es 1/1023 * 100 por ciento 0.1 %

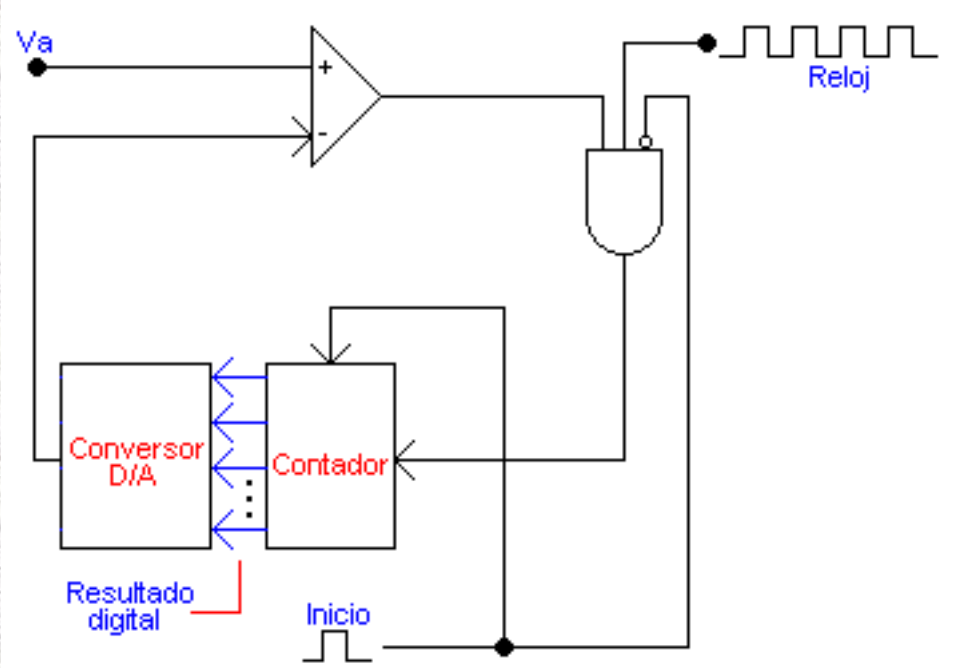


Figura 6: Conversión A/D de rampa digital

CONVERSIÓN DE CÓDIGOS

Los conversores de códigos son una aplicación de las puertas lógicas en los sistemas digitales. Los códigos mas utilizados son el binario BCD 8421, octal, hexadecimal y el decimal. Los dispositivos digitales pueden procesar solamente los bits "1" y "0" . Estas largas cadenas de 1 y 0 son difíciles de comprender por las personas. Por esta razón se necesitan los conversores de códigos para traducir el lenguaje de la gente al lenguaje de la maquina.

Un ejemplo de conversor de código es una sencilla calculadora manual, la cual esta constituida por un dispositivo de entrada llamado teclado. Entre el teclado y la unidad central de tratamiento "CPU" hay un codificador, que traduce el numero decimal pulsado en el teclado a código binario. La "CPU" realiza su operación en binario y produce un resultado en código binario. El decodificador traduce el código binario de la CPU a un código especial que hacen que luzcan los segmentos adecuados en el visualizador de siete segmentos.

Los conversores de códigos se dividen en dos tipos:

- Codificador
- Decodificador

CODIFICADORES

Un codificador es considerado como un traductor del lenguaje de la gente al lenguaje de la maquina, es decir, traduce una entrada decimal a un numero BCD 8421.

El diagrama lógico, en forma simplificada, de un codificador decimal a BCD se muestra en la figura 1

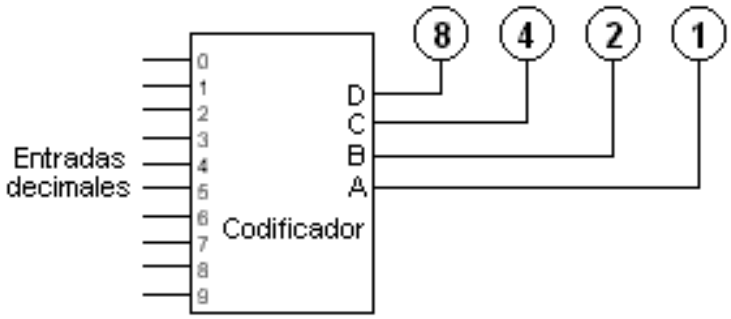


Figura 1: Codificador BCD

El codificador tiene diez entradas a la izquierda y cuatro salidas a la derecha, además puede tener una entrada activa, que produce una única salida.

Una característica poco habitual del codificador es que no hay entrada 0. Una entrada cero significa una salida 1111 (en D, C, B y A) que es verdadera cuando todas las entradas del 1-9

están desconectadas. Cuando las entradas no están conectadas, se dice que están flotando.

En la figura se presenta el diagrama de bloques y la tabla de verdad de un codificador comercial denominado de prioridad de 10-4 líneas.

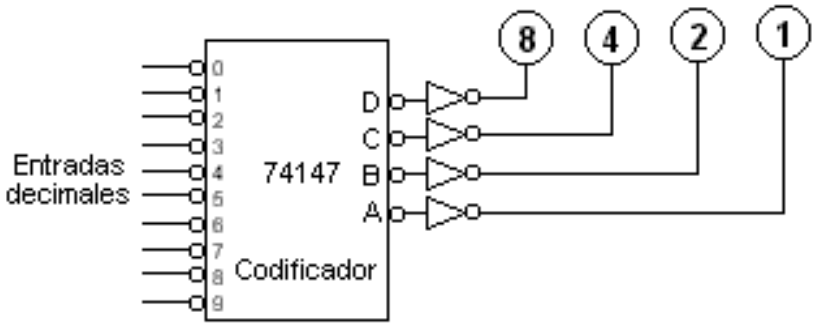


Figura 2: Símbolo lógico del codificador 74147

1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

Tabla 1: Tabla de verdad del codificador 74147

La primera línea de la tabla de verdad indica que no hay entrada. Cuando todas las entradas flotan el alto, las salidas flotan en alto, lo cual es interpretado como 0000 por los indicadores de salidas. La segunda línea de la tabla muestra la entrada decimal 9 activada por un nivel bajo, lo que produce LHHL en la salida. Esta salida la invierten los cuatro inversores y en los indicadores BCD se lee 1001. En la misma línea se muestra las entradas del 1 al 8 marcadas con X (irrelevante). Una entrada irrelevante puede estar alta o baja. Este codificador tiene una característica de prioridad, que activa él número mayor que tenga una entrada en baja. Si por ejemplo tenemos un nivel bajo en el 3 y en 8, el codificador dará una salida en binario correspondiente al numero mayor, en este caso el 8.

En la figura se presenta el diagrama lógico del codificador 74147, donde se presentan las 30 puertas lógicas que lo conforman.

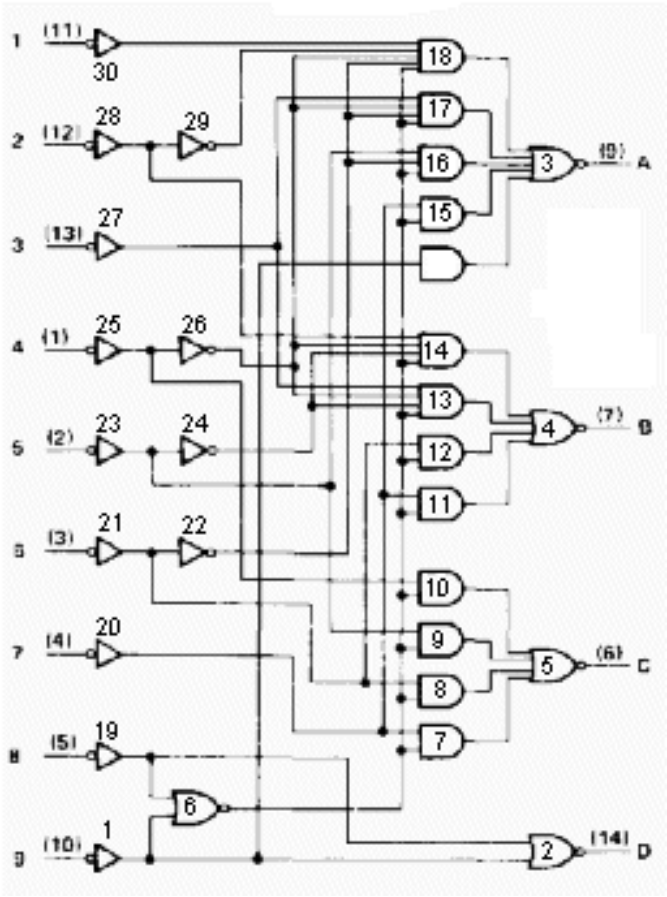


Figura 3: Diagrama lógico del codificoador 74147

Si queremos activar el 9 decimal le colocamos un nivel bajo. Esta entrada a 0 la invierte el inversor 1, y se aplica a las puertas NOR 2 y 3, que se activan entonces, dando una salida en baja. Las puertas NOR 4 y 5 se desactivan por la presencia de 0 en las entradas de las puertas AND (de la 7-18) desactivadas. Estas puertas AND están desactivadas por los 0 de sus entradas inferiores, producidos por la puerta NOR 6. Las puertas AND de la (7-18) aseguran que tenga prioridad sobre las demás, la entrada decimal correspondiente al numero mayor.

También se dispone de codificadores con tecnología CMOS, de donde se destaca el codificador de prioridad de 10-4 líneas 74HC147.

DECODIFICADOR BCD A DECIMAL

Un decodificador es considerado como el proceso inverso de un codificador, es decir, un traductor de lenguaje de las maquina al lenguaje de la gente.

El diagrama de bloque del decodificador se muestra en la figura 4.

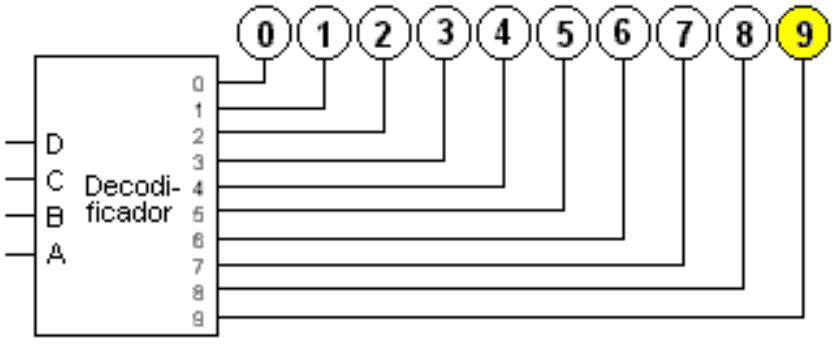


Figura 4: Símbolo lógico de un decodificador BCD a decimal

El decodificador tiene 4 entradas a la izquierda que están conformadas por el código BCD 8421, y tiene a la derecha diez líneas de salidas. En la figura se muestra el decodificador comercial BCD a decimal, TTL 7442 y su correspondiente tabla de verdad.

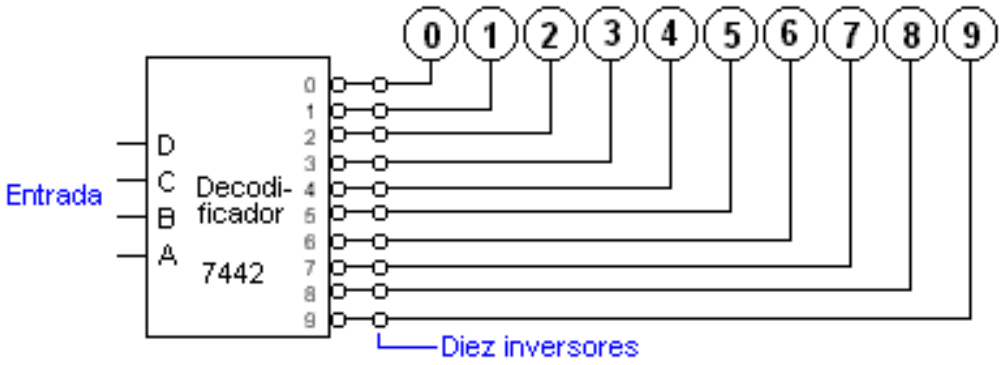


Figura 5: Símbolo lógico del decodificador/excitador BCD a decimal 7442

Línea	Nº	Entradas BCD													
		D	**C**	**B**	**A**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Línea 1	0	L	L	L	L	L	H	H	H	H	H	H	H	H	H
Línea 2	1	L	L	L	H	H	L	H	H	H	H	H	H	H	H
Línea 3	2	L	L	H	L	H	H	L	H	H	H	H	H	H	H
Línea 4	3	L	L	H	H	H	H	H	L	H	H	H	H	H	H
Línea 5	4	L	H	L	L	H	H	H	H	L	H	H	H	H	H
Línea 6	5	L	H	L	H	H	H	H	H	H	L	H	H	H	H
Línea 7	6	L	H	H	L	H	H	H	H	H	H	L	H	H	H
Línea 8	7	L	H	H	H	H	H	H	H	H	H	H	L	H	H
Línea 9	8	H	L	L	L	H	H	H	H	H	H	H	H	L	H
Línea 10	9	H	L	L	H	H	H	H	H	H	H	H	H	H	L
Líneas 11-16	Inválido			H	H	H	H	H	H	H	H	H	H	H	H

Tabla 2: Tabla de verdad del decodificador 7442

A la izquierda se encuentran las 4 entradas BCD etiquetadas con D, C, B y A. Estas entradas se activan con el uno lógico, o nivel alto. A la derecha se encuentran las 10 salidas del decodificador, cada una con un circulito que indican que las salidas son activas en baja, es decir, que normalmente están en alta. Los inversores que se encuentran a la salida se añaden por conveniencia para controlar las luces de los indicadores decimales, es decir, que una salida activa se invierte a uno lógico en los indicadores de salidas.

En la primera línea de la tabla de verdad se muestran todas las entrada en el nivel bajo (L). Con una entrada LLLL se activa la salida del cero decimal al estado bajo. El inversor inferior complementa esta salida al nivel alto, lo que hace que luzca el indicador de la salida decimal cero, no permitiendo que ninguno de los demás luzcan. De igual forma, la quinta línea muestra la entrada BCD LHLL, lo que hace que se active la salida cuatro en el nivel bajo. Esta salida es invertida haciendo que luzca el indicador de la salida decimal 4.

La línea 11 muestra la entrada HLHL, que normalmente representa el decimal 10. Como el código BCD no contiene este número, esta entrada es invalida y no lucirá ninguna lampara de

salida. Igualmente para las 5 últimas líneas de la tabla del diagrama lógico del decodificador 7442, BCD a decimal, se muestran las figuras.

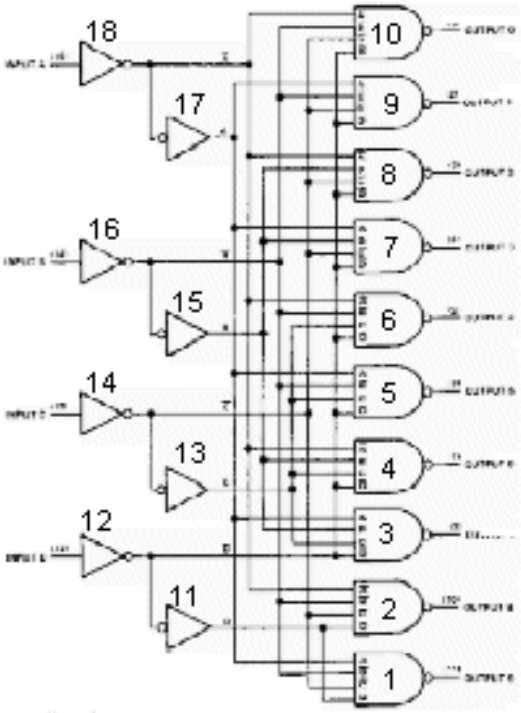


Figura 6: Diagrama lógico del decodificador BCD a decimal 7442

La entrada A3 es el BIT más significativo (MSB), o la entrada del 8 y la entrada A0 es el BIT menos significativo (LSB), o la entrada del 1. La salidas están etiquetadas con números decimales. La salidas que se encuentran activas en baja aparecen con barras sobres las salidas decimales (9, 8 etc...)

Supongamos que la entrada BCD es LLLL (0000). Si seguimos cuidadosamente el camino de las cuatro entradas a través de los inversores 12,14,16 y 18, se observa que a la puerta NAND 1 se aplican cuatro 1 lógicos, que la activan produciendo entonces un cero lógico. Todas las demás puertas NAND quedan inhabilitadas por la presencia de un cero en algunas de sus entradas.

En la tecnología CMOS también encontramos diversos tipos de decodificadores BCD a decimal, dentro de los cales los más representativos son el 4028, 74C42 y 74HC42.

DECODIFICACION BCD A CÓDIGO DE 7 SEGMENTOS

Un dispositivo de salida muy utilizado para visualizar números decimales es el visualizado de 7 segmentos.

Los 7 segmentos se marcan con las letras de la a a la g.

Existes varios tipos de visualizadores dentro de los cuales encontramos, el denominado incandescente, que es similar a una lampara común, el de tubo de descarga de gas, que opera a tensiones altas y produce una iluminación anaranjada, el de tubo fluorescente, que da una iluminación verdosa cuando luce y opera con tensiones bajas, el mas moderno que es el de cristal liquido (LCD), este crea números negros sobre fondos plateados, y por último el visualizador común de diodos emisores de luz (LED) que produce un brillo rojo cuando luce. Existen visualizadores LED que cuando lucen emiten colores distintos del rojo.

Como el visualizador LED es el mas fácil de utilizar y el mas común por eso se tratará con mas detalles.

En la figura se muestra la forma de operación de un visualizador de 7 segmentos.

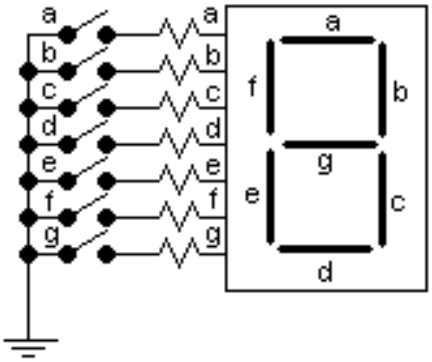


Figura 7: Operación de un visualizador de 7 segmentos

Cada segmento (de a a g) contiene un LED. Como la corriente típica de un LED es de 20 mA, se colocan resistores de 150 (ohmios) con el fin de limitar dicha corriente. Sin este resistor, el LED podría quemarse debido a que un LED puede soportar solo 1.7V a través de sus terminales.

Existen dos tipos de visualizadores LED, el de ánodo común y el de cátodo común.

Cátodo común: cuando todos los cátodos están unidos entre sí y van directo a tierra.

Anodo común: cuando todos los ánodos están conectados entre sí y van a la fuente de alimentación como el caso del ejemplo del cual estamos hablando.

Si, por ejemplo, se desea que aparezca el número decimal 7 en el visualizador de la figura deben cerrarse los conmutadores a, b y c para que luzcan los segmentos a, b y c del LED. Observar que una tensión de tierra (baja) activa a los segmentos de este visualizador LED.

En la figura se muestra el dispositivo TTL denominado decodificador excitador 7447A BCD a 7 segmentos, con su respectiva tabla de verdad.

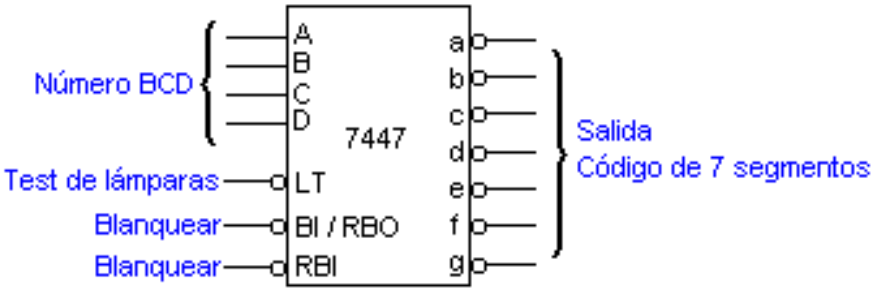


Figura 8: Símbolo lógico del decodificador 7447

DECIMAL OR FUNCTION	INPUTS						BI/RBO*	OUTPUTS							NOTE
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	ON	ON	ON	ON	ON	ON	OFF	1
1	H	X	L	L	L	H	H	OFF	ON	ON	OFF	OFF	OFF	OFF	
2	H	X	L	L	H	L	H	ON	ON	OFF	ON	ON	OFF	ON	
3	H	X	L	L	H	H	H	ON	ON	ON	ON	OFF	OFF	ON	
4	H	X	L	H	L	L	H	OFF	ON	ON	OFF	OFF	ON	ON	
5	H	X	L	H	L	H	H	ON	OFF	ON	ON	OFF	ON	ON	
6	H	X	L	H	H	L	H	OFF	OFF	ON	ON	ON	ON	ON	
7	H	X	L	H	H	H	H	ON	ON	ON	OFF	OFF	OFF	OFF	
8	H	X	H	L	L	L	H	ON	ON	ON	ON	ON	ON	ON	
9	H	X	H	L	L	H	H	ON	ON	ON	OFF	OFF	ON	ON	
10	H	X	H	L	H	L	H	OFF	OFF	OFF	ON	ON	OFF	ON	
11	H	X	H	L	H	H	H	OFF	OFF	ON	ON	OFF	OFF	ON	
12	H	X	H	H	L	L	H	OFF	ON	OFF	OFF	OFF	ON	ON	
13	H	X	H	H	L	H	H	ON	OFF	OFF	ON	OFF	ON	ON	
14	H	X	H	H	H	L	H	OFF	OFF	OFF	ON	ON	ON	ON	
15	H	X	H	H	H	H	H	OFF	OFF	OFF	OFF	OFF	OFF	OFF	
BI	X	X	X	X	X	X	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2
RBI	H	L	L	L	L	L	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3
LT	L	X	X	X	X	X	H	ON	ON	ON	ON	ON	ON	ON	4

1. La entrada de borrado (BI) debe estar abierta o mantenida en el nivel lógico ALTO cuando se desean las funciones de salida 0 a 15. La entrada de borrado de rizado (RBI) debe estar abierta o en ALTA sino se desea borrar ningún cero decimal.
2. Cuando se aplica el nivel lógico BAJO directamente a la entrada de borrado (BI), todas las salidas de los segmentos están en OFF sin tener en cuenta el nivel de las otras entradas.
3. Cuando la entrada de borrado de rizado (RBI) y las entradas A, B, C y D están en nivel BAJO, con la entrada de test de lámparas en ALTO, todos los segmentos de salida están en OFF y la salida de borrado de rizado (RBO) va a nivel bajo (condición de respuesta).
4. Cuando la entrada BI/RBO se mantiene en ALTA, y se aplica un nivel bajo a LT, todos los segmentos de salida están en ON.

Tabla 3: Tabla de verdad del decodificador 7447

La entrada es un número BCD de 4 BITS, el número BCD se transforma en un código de 7 segmentos que ilumina los segmentos del visualizador LED. También se muestran 3 entradas extras en el símbolo lógico. La entrada de test de lamparas hará lucir todos los segmentos adecuados para ver si son operativos. Las estradas de borrado que son las que desconectan todos los elementos activados. Las entradas de borrado y test de lamparas son activadas por niveles de tensión bajo y las entradas BCD son activadas por 1 lógicos.

Observar la línea 1 de la tabla de verdad. Para que aparezca el 0 decimal en el visualizador, las entradas BCD deben ser LLLL. Esto activará los segmentos a, b, c, d, e y f para formar el cero decimal.

Las entradas BCD inválidas (decimal 10, 11, 12, 13, 14 y 15) no son números BCD; sin embargo, generan una única salida. Para la línea decimal 10, entradas HLHL, la columna de salida indica que se activan la salida d, e, y g. Formando una pequeña c.

En la familia CMOS existen muchos decodificadores para visualizadores dentro de los cuales se destacan el 74C48 que no necesita circuitería extra para la mayoría de los visualizadores LED, el 4511 y el 74HC4511.

VISUALIZADOR DE CRISTAL LIQUIDO

Están hechos de vidrio y son muy frágiles. Las principales ventajas de los LCD son su extremadamente bajo consumo de energía y su larga vida. La principal desventaja de los LCD es su lento tiempo de conmutación, que pude ser desde 40 hasta 100 ms. Una segunda desventaja es la necesidad de luz ambiental debido a que el LCD refleja luz pero no emite como los LED.

En la figura se muestra una sección de un LCD de efecto de campo típico

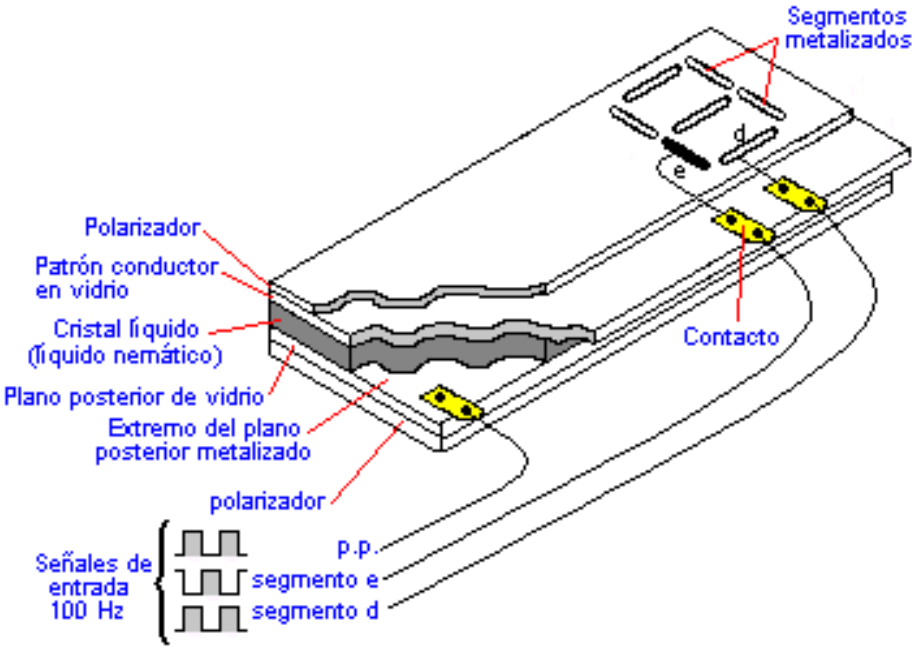


Figura 9: LCD de efecto de campo

Cuando se aplica una tensión entre los segmentos metalizados del vidrio superior y del plano posterior, el segmento cambia a negro sobre un fondo plateado. Esto se debe a que el cristal liquido o fluido "nemático" emparedado entre las partes frontal y posterior del vidrio transmite luz de forma diferente cuando esta activado. Este LCD efecto de campo usa filtro polarizado en las parte superior e inferior de la pantalla. Cada segmento y el plano posterior están conectados internamente a contactos en el flanco del empaquetamiento del LCD .

Los LCD están controlados por señales en forma de onda cuadrada (30 a 200 Hz) de baja frecuencia con un ciclo de trabajo del 50% (50% de tiempo esta en alta). En resumen, las señales en fase no activan el visualizador, mientras que las señales desfasadas 180 grados activan un segmento del LCD.

En la figura se muestra un LCD típico que se encuentra en un encapsulamiento de 40 patillas.

Este LCD esta construido con fluido nemático emparedado entre placas de cristal y polarizadores en los extremos superior e inferior. Cabeceras de plástico que aseguran las placas de vidrio del LCD en las patillas

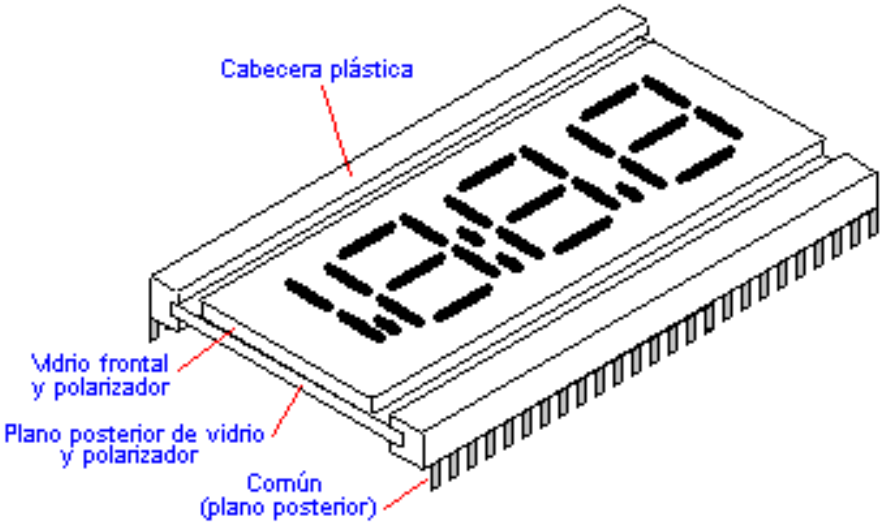


Figura 10: LCD comercial de 3 1/2 dígitos

Las señales de control de los LCD deben ser generadas por CI CMOS, ya que estos consumen muy poca energía y sus señales no tiene un desplazamiento de tensión DC como el que se presenta cuando se utilizan CI TTL. Un desplazamiento de tensión DC aplicando a través del fluido nemático destruirá el LCD después de cierto tiempo.

CONTROLADORES DE LCD

En la figura se muestra un diagrama de bloques de un sencillo circuito de codificador / controlador LCD.

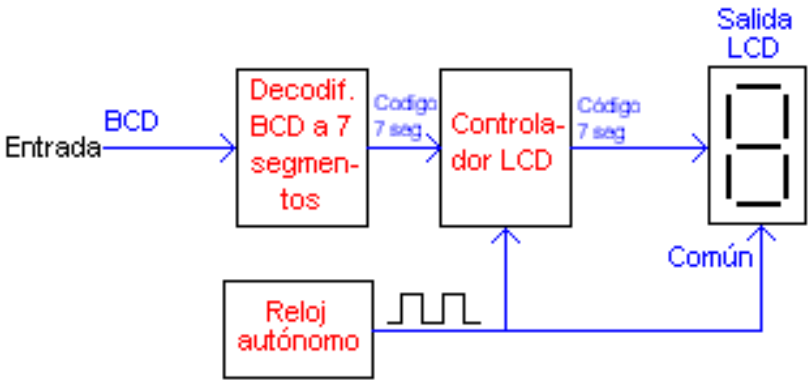


Figura 11: Diagrama de bloques de un decodificador/controlador LCD de 7 segmentos

Este decodificador convierte el código BCD de entrada a código de siete segmento. A continuación, la unidad controladora LCD tomaría la señal de onda cuadrada de 100 Hz del reloj autónomo y envía señales invertidas (desfasadas 180º) solamente a los segmentos LCD que se van a activar. El reloj autónomo es un multivibrador estable que continuamente genera una cadena de pulsos de onda cuadrada con un ciclo de trabajo del 50%.

En la figura se muestra un diagrama más detallado del controlador/ decodificador LCD.

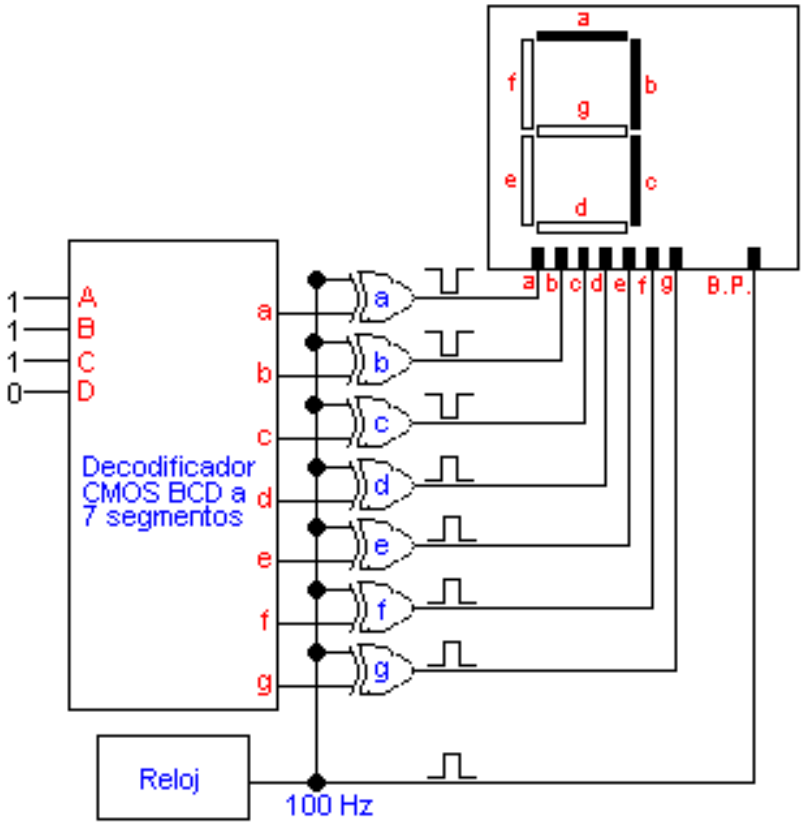


Figura 12: Diagrama de conexiones de un decodificador/controlador LCD

Observe que la entrada BCD al decodificador es 0111. El decodificador traduce la entrada y activa las salidas a, b y c al nivel ALTO, que es el código de siete segmentos adecuado para visualizar el decimal 7. Las demás salidas (d, e, f, y g) permanecen en el nivel bajo.

La sección controladora del LCD contiene siete puertas XOR CMOS de dos entradas. La señal de 100 Hz controla la entrada superior de cada puerta XOR y la entrada inferior esta conectada directamente al decodificador. Si la entrada inferior esta en nivel BAJO, la señal pasa a través de la puerta sin cambiar (en fase con la señal del reloj). Pero si por el contrario la entrada esta al nivel ALTO, la señal se invierte y pasa a través de la puerta (se desfasa 180º con respecto a la señal del reloj).

Existen dos CI CMOS comerciales, que realizan la tarea del decodificador / controlador LCD. Estos son los CI 4543 y 74HC4543, descritos por el fabricante como un cerrojo / decodificador/ controlador BCD a siete segmentos para LCD. En la figura se presenta un diagrama de bloques del decodificador / controlador que utiliza el CI 74HC4543.

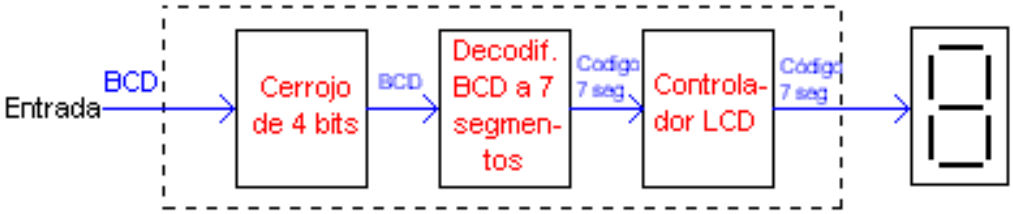


Figura 13: Diagrama de bloques del controlador LCD 74HC4543

Este chip contiene una sección decodificadora BCD a siete segmentos, una sección controladora del LCD y una sección de cerrojos de 4 bits para bloquear la entrada BCD en un instante dado. Se considera el cerrojo como una unidad de memoria que almacena los 4 bits de entrada en la entrada de la sección decodificadora durante un cierto tiempo.

Para un mejor entendimiento en la figura se presenta un diagrama de conexiones del circuito decodificador / controlador que utiliza el CI 74HC4543.

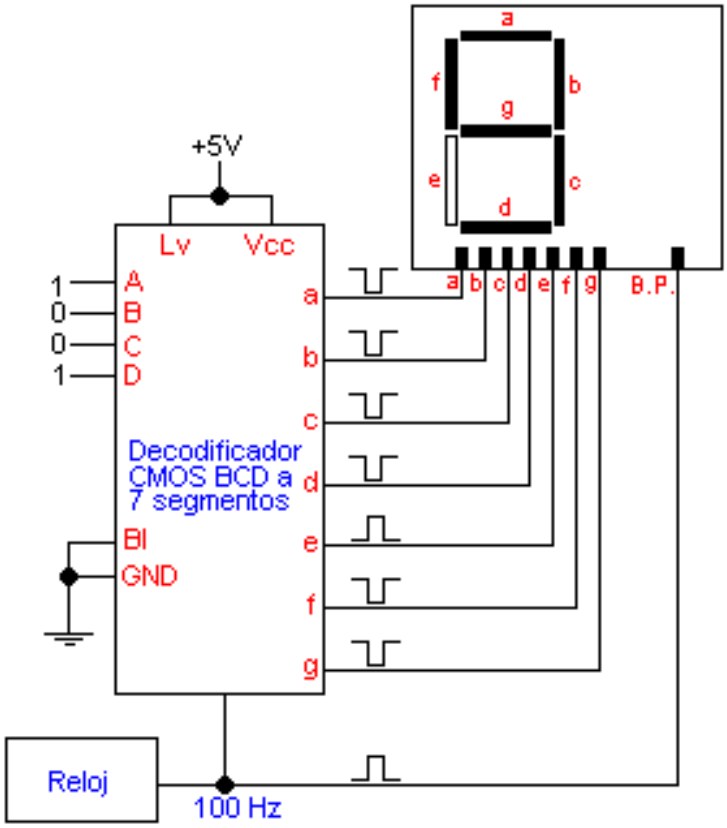


Figura 14: Diagrama de conexiones del contador 74HC4543.

Observar que toman como ejemplo el 9 decimal, es decir, que la entrada BCD es 1001. Esta entrada es decodificada en el código de siete segmentos. La señal de reloj de 100 Hz se conecta tanto a la parte común (plano posterior) del LCD como la entrada ph (fase) del CI 74HC4543. Observa que la sección controladora invierte la señal a los segmentos que se van activar. Este ejemplo se activan los segmentos a, b, c, d, f y g, visualizando el decimal que en el LCD. La únicas señales que pasan al LCD son las de segmentos inactivos. En el ejemplo del segmento e.

VISUALIZADORES FLUORESCENTES DE VACÍO

Este tipo de visualizador es un pariente del antiguo tubo triodo de vacío. Por esta razón se hace necesario un pequeño recuento del tubo triado de vacío. Se divide en tres partes, la placa (p), rejilla de control (g) y el cátodo (k). La placa a veces se denomina ánodo, mientras que el cátodo puede denominarse filamento o calentador. El cátodo es un hilo fino que cuando se reviste con un material tal como óxido de bario emite electrones cuando se calienta. La rejilla de control es una pantalla colocada entre cátodo y placa.

El fenómeno de emisión de electrones al vacío por parte del cátodo cuando se calienta, a

veces se denomina emisión termoiónica. Si la rejilla y placas son positivas los electrones cargados negativamente serán atraídos y fluirán a través de la rejilla hacia la placa. El triodo esta conduciendo corriente del cátodo al ánodo.

Para que el diodo deje de conducir se pueden emplear dos métodos. Primero se pude colocar una carga negativa en la rejilla de control. Esto repelaran los electrones y dejaran de pasar a través la rejilla hacia la placa. Segundo, llevar la tensión la rejilla a 0 voltios. Sin tensión en la placa, esta no emitirá electrones y el tubo triodo no conducirá.

En la figura se muestra el diagrama esquemático del visualizador fluorescente de vacío.

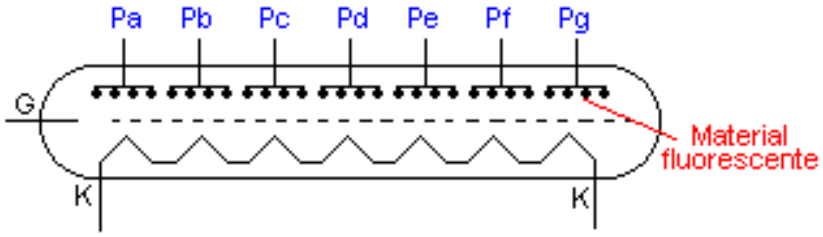


Figura 15: Diagrama esquemático de un visualizador fluorescente VF

Este esquema representa un único dígito de siete segmentos que tiene siete placas cada una revestida como un material fluorescente de oxido de cinc. Este visualizador tiene una rejilla que controla el visualizador completo, un único cátodo / filamento (k) y la unidad entera que esta encerrada en vidrio en el que se ha hecho el vacío.

La operación de un solo dígito de un visualizador VF se ilustra en la figura.

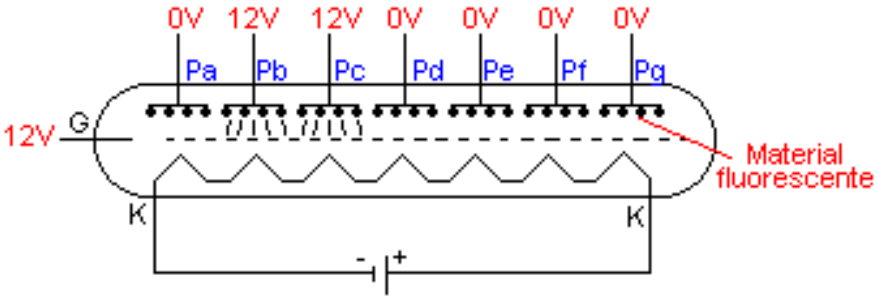


Figura 16: Operación de un sólo dígito de un visualizador VF

El filamento se calienta usando una tensión dc. La rejilla de control tiene aplicados + 12 voltios, que activan el visualizador completo. En este ejemplo solo los segmentos b y c son los que se activan, por esta razón solo las placas pb y pc están activadas con + 12 voltios. Los electrones fluyen solamente desde el cátodo hasta las placas pb y pc del visualizador vf. En conclusión una tensión de placas de 12 v ilumina un segmento, mientras que 0 v en una placa significa que el segmento no lucirá.

Los visualizadores fluorescentes de vacío se utilizan especialmente en los equipos electrónicos de los automóviles. Estos visualizadores tienen una vida extremadamente larga, respuesta rápida, opera a bajas tensiones (12 v), consume poca potencia, tiene buena fiabilidad y es barato. Los visualizadores de vf son compatibles con la familia CMOS de CI.

CONTROL DE VISUALIZADORES VF CON CMOS

La figura muestra el decodificador / controlador y el circuito visualizador VF utilizanod un CI 4511.

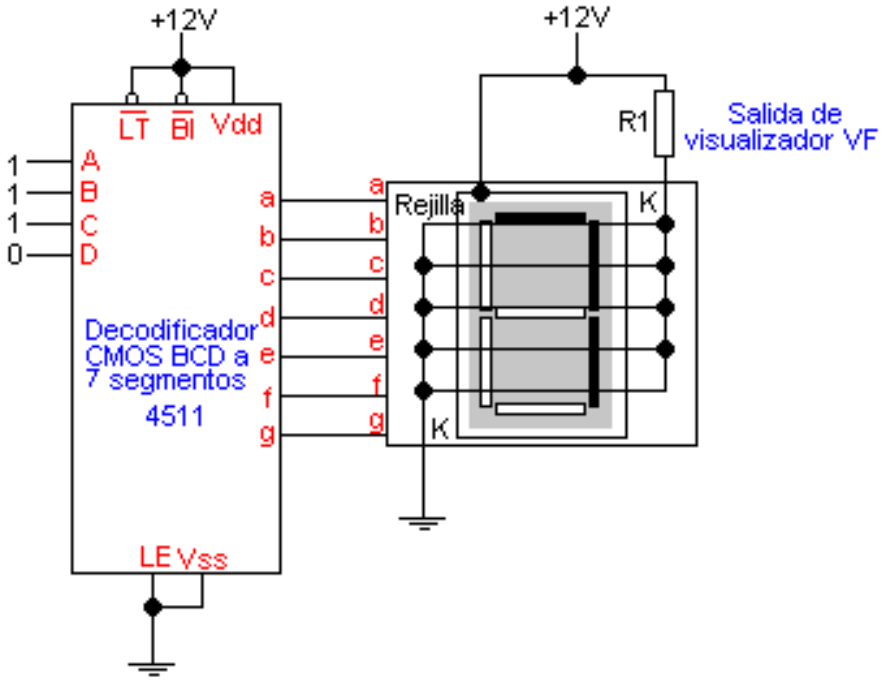


Figura 17: Cerrojo/decodificador/controlador BCD a 7 segmentos 4511

En este ejemplo, la entrada BCD es 0111. Esta entrada es decodificada por el CI cerrojo / decodificador / controlador 4511, y el visualizador VF hace visible el decimal 7.

Solo las salidas a, b, y c se activan (nivel ALTO) en el 4511. Estos tres niveles ALTOS controlan las placas de los segmentos a, b y c del visualizador VF a +12 V. La rejilla se conecta directamente al terminal positivo de la fuente de alimentación de +12 V y el cátodo (k) se conecta en serie con un resistor limitador (R_1) para calentar el filamento. El resistor limita la corriente a través del filamento a un nivel seguro.

La sección de cerrojos (entrada LE) del CI 4511 se inhabilita, manteniéndolo en el nivel BAJO. Con el cerrojo inhabilitado, los datos de la entrada BCD pasan a través de la sección decodificadora del CI 4511.

Se usa una fuente de alimentación de +12 V tanto para el visualizador fluorescente de vacío, como para el chip CMOS 4511.

La sección controladora del CI 4511 tiene conectadas sus salidas directamente a las placas (ánodos) de visualizador VF. Un nivel ALTO en la salida del controlador activa el segmento en el visualizador de siete segmentos VF, siempre y cuando esté activada la rejilla de control del visualizador. Un nivel BAJO en la salida del controlador desactiva el segmento del visualizador VF, y no luce. Cuando la entrada LT (test de luz) se activa con un nivel BAJO todas las salidas de CI 4511 alcanzan el nivel ALTO y cuando BI (entrada de bloqueo) se activa con un nivel BAJO, todas las salidas alcanzan el nivel BAJO y todos los segmentos del visualizador conectados se ponen en blanco.

FLIP-FLOPS

Los circuitos lógicos se clasifican en dos categorías. Los grupos de puertas descritos hasta ahora, y los que se denominan circuitos lógicos secuenciales. Los bloques básicos para construir los circuitos lógicos secuenciales son los flip-flops. La importancia de los circuitos lógicos se debe a su característica de memoria. Los flip-flops también se denominan "cerrojos", "multivibradores biestables" o "binarios".

FLIP-FLOPS RS

Este es el flip-flop básico, su símbolo es el siguiente:

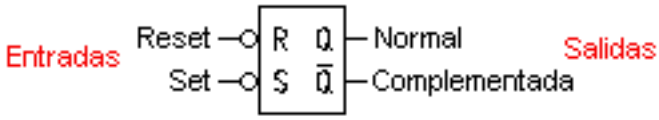


Figura 1: Símbolo lógico de un flip-flop SR

El flip-flop tiene dos entradas R (reset) y S (set), se encuentran a la izquierda del símbolo. Este flip-flop tiene activas las entradas en el nivel BAJO, lo cual se indica por los circulitos de las entradas R y S. Los flip-flop tienen dos salidas complementarias, que se denominan Q y $\overline{1}$, la salida Q es la salida normal y $\overline{1}$ = 0. El flip-flop RS se puede construir a partir de puertas lógicas. A continuación mostraremos un flip-flop construido a partir de dos puertas NAND, y al lado veremos su tabla de verdad correspondiente.

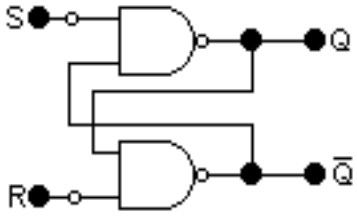


Figura 2: Circuito equivalente de un flip-flop SR

Modo de operación	Entradas S	R	Salidas Q	$\overline{Q}$
Prohibido	0	0	1	1
Set	0	1	1	0
Reset	1	0	0	1
Mantenimiento	1	1	No cambia	

Tabla 1: Tabla de verdad del flip-flop SR

Observar la realimentación característica de una puerta NAND a la entrada de la otra. En la tabla de la verdad se define la operación del flip-flop. Primero encontramos el estado "prohibido" en donde ambas salidas están a 1, o nivel ALTO. Luego encontramos la condición "set" del flip-flop. Aquí un nivel BAJO, o cero lógico, activa la entrada de set(S). Esta pone la salida normal Q al nivel alto, o 1. Seguidamente encontramos la condición "reset". El nivel BAJO, o 0, activa la entrada de reset, borrando (o poniendo en reset) la salida normal Q. La cuarta línea muestra la condición de "inhabilitación" o "mantenimiento", del flip-flop RS. Las salidas permanecen como estaban antes de que existiese esta condición, es decir, no hay cambio en las salidas de sus estados anteriores. Indicar la salida de set, significa poner la salida Q a 1, de igual forma, la condición reset pone la salida Q a 0. La salida complementaria nos muestra lo opuesto. Estos flip-flop se pueden conseguir a través de circuitos integrados.

FLIP-FLOPS RS SINCRONO

El flip-flop RS es un dispositivo asíncrono. No opera en conjunción con un reloj o dispositivo de temporización. El flip-flop RS síncrono opera en conjunción con un reloj, en otras palabras opera sincronizadamente. Su símbolo lógico se muestra a continuación. Es igual a un flip-flop RS añadiéndole una entrada de reloj.

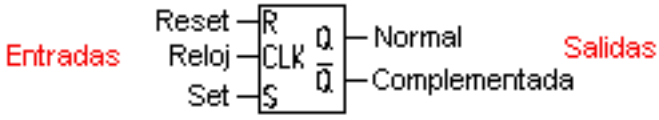


Figura 3: Símbolo de un flip-flop SR síncrono

El flip-flop RS síncrono puede implementarse con puertas NAND. En las siguientes ilustraciones vemos primero como se añaden dos puertas NAND al flip-flop RS para construir un flip-flop RS síncrono. Las puertas NAND 3 y 4 añaden la característica de sincronismo al cerrojo RS. La tabla de la verdad nos muestra la operación del flip-flop RS síncrono. El modo de mantenimiento se describe en la primera línea de la tabla de la verdad. Cuando un pulso de reloj llega a la entrada CLK (con 0 en las entradas R y S), las salidas no cambian, permanecen igual que antes de la llegada del pulso de reloj. Este modo también puede llamarse de "inhabilitación" del FF. La línea 2 es el modo de reset. La salida normal Q se borrará cuando un nivel ALTO active la entrada R y un pulso de reloj active la entrada de reloj CLK. Si R=1 y S=0, el FF no se pone a 0 inmediatamente, esperará hasta que el pulso del reloj pase del nivel BAJO al ALTO, y entonces se pone a 0. La línea 3 de la tabla describe el modo set del flip-flop. Un nivel ALTO activa la entrada S (con R=0 y un pulso de reloj en el nivel ALTO), poniendo la salida Q a 1. La línea 4 de la tabla de verdad es una combinación "prohibida" todas las entradas están en 1, no se utiliza porque activa ambas salidas en el nivel ALTO.

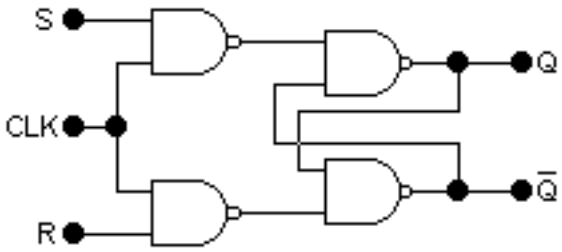


Figura 4: Circuito eléctrico equivalente de un flip-flop SR síncrono

Modo de operación	ENTRADAS			SALIDAS	
	CLK	S	R	Q	$\underline{Q}$
Mantenimiento	⎍	0	0	No cambia	
Reset	⎍	0	1	1	1
Set	⎍	1	0	1	0
Prohibido	⎍	1	1	1	1

Tabla 2: Tabla de verdad de un flip-flop SR síncrono

Las formas de ondas, o diagramas de tiempo, se emplean mucho y son bastante útiles para trabajar con flip-flop y circuitos lógicos secuenciales. A continuación mostraremos un

diagrama de tiempo del flip-flop RS síncrono. Las 3 líneas superiores representan las señales binarias de reloj, set y reset. Una sola salida Q se muestra en la parte inferior. Comenzando por la izquierda, llega el pulso de reloj 1, pero no tiene efecto en Q porque las entradas R y S están en el modo de mantenimiento, por tanto, la salida Q permanece a 0. En el punto a del diagrama del tiempo, la entrada de set se activa en el nivel ALTO. Después de cierto tiempo en el punto b, la salida se pone a 1. Mirar que el flip-flop ha esperado a que el pulso 2 pase del nivel BAJO a ALTO antes de activar la salida Q a 1. El pulso está presente cuando las entradas R y S están en modo de mantenimiento, y por lo tanto la salida no cambia. En el punto C la entrada de reset se activa con un nivel ALTO. Un instante posterior en el punto d la salida Q se borra ó se pone a 0, lo cual ocurre durante la transición del nivel BAJO a ALTO del pulso del reloj. En el punto e está activada la entrada de set, por ello se pone a 1 la salida Q en el punto f del diagrama de tiempos. La entrada S se desactiva y la R se activa antes del pulso 6, lo cual hace que la salida Q vaya al nivel BAJO o a la condición de reset. El pulso 7 muestra que la salida Q sigue a las entradas R Y S todo el tiempo que el reloj está en ALTA. En el punto g del diagrama de tiempos, la entrada de set (S) va a nivel ALTO y la salida Q alcanza también el nivel ALTO. Después la entrada S va a nivel BAJO. A continuación en el punto h, la entrada de reset (R) se activa por un nivel ALTO. Eso hace que la salida Q vaya al estado de reset, o nivel BAJO. La entrada R entonces vuelve al nivel BAJO, y finalmente el pulso de reloj finaliza con la transición del nivel ALTO al BAJO. Durante el pulso de reloj 7, la salida estuvo en el nivel ALTO y después en el BAJO. Observar que entre los pulsos 5 y 6 ambas entradas R y S están a 1. La condición de ambas entradas R y S en el nivel ALTO, normalmente, se considera un estado prohibido para el flip-flop. En este caso es aceptable que R y S estén en el nivel ALTO, porque el pulso de reloj está en el nivel BAJO y el flip-flop no está activado.

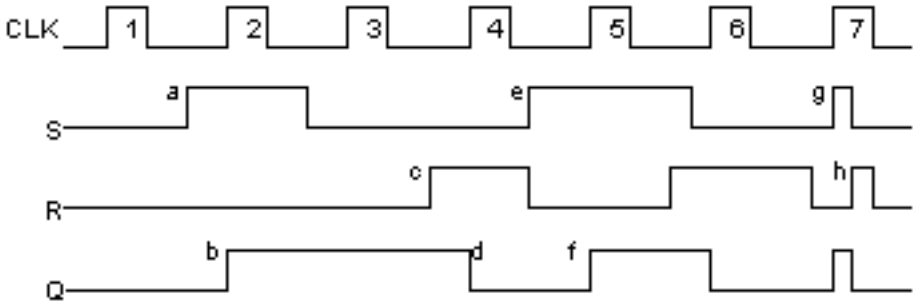


Figura 5: Diagrama de pulsos

FLIP-FLOP D

El símbolo lógico para un flip-flop D es el siguiente:

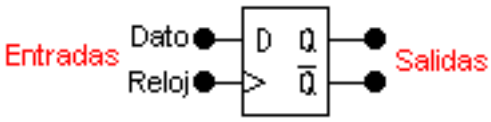


Figura 6: Símbolo lógico de un flip-flop D

Tiene solamente una entrada de datos (D), y una entrada de reloj (CLK). Las salidas Q Y 1. También se denomina " flip-flop de retardo ". Cualquiera que sea el dato en la entrada (D), éste aparece en la salida normal retardado un pulso de reloj. El dato se transfiere durante la transición del nivel BAJO al ALTO del pulso del reloj.

FLIP-FLOP JK

El símbolo lógico para un flip-flop JK es el siguiente:

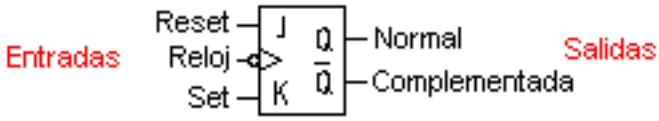


Figura 7: Símbolo lógico de un flip-flop JK

Este flip-flop se denomina como "universal" ya que los demás tipos se pueden construir a partir de él. En el símbolo anterior hay tres entradas síncronas (J, K y CLK). Las entradas J y K son entradas de datos, y la entrada de reloj transfiere el dato de las entradas a las salidas.

A continuación veremos la tabla de la verdad del flip-flop JK:

Modo de operación	**ENTRADAS** **CLK** **S** **R**			**SALIDAS** **Q** **Q**	
Mantenimiento	⎍	0	0	No cambia	
Reset	⎍	0	1	0	1

Set	⎍	1	0	1	0
Conmutación	⎍	1	1	Estado opuesto	

Tabla 3: Tabla de verdad para un flip-flop JK

Observamos los modos de operación en la parte izquierda y la tabla de la verdad hacia la derecha. La línea 1 muestra la condición de "mantenimiento", o inhabilitación. La condición de "reset" del flip-flop se muestra en la línea 2 de la tabla de verdad. Cuando J=0 y K=1 y llega un pulso de reloj a la entrada CLK, el flip-flop cambia a 0(Q=0). La línea 3 muestra la condición de "set" del flip-flop JK. Cuando J=1 y K=0 y se presenta un pulso de reloj, la salida Q cambia a 1. La línea 4 muestra una condición muy difícil para el flip-flop JK que se denomina de conmutación.

DISPARO DE LOS FLIP-FLOPS

La mayor parte de los complicados equipos digitales operan como un sistema secuencial síncrono, lo que sugiere que un reloj maestro envíe las señales a todas las partes del sistema para la operación del mismo. Un tren de pulsos de reloj, típico, se muestra en la siguiente figura.

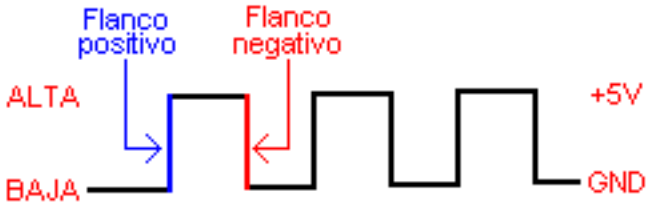


Figura 8: Disparo de los flip-flops

La distancia horizontal en la onda es el tiempo y las distancia vertical es la tensión.

En la figura 8 la tensión está primero en el nivel BAJO, o GND (tierra), también denominado 0 lógico.

El pulso de a muestra el "flanco anterior" o "flanco positivo" de la forma de onda, que va de la tensión GND a +5 V. Este flanco también se denomina de BAJA a ALTA (L a H). En la

parte derecha del pulso a, la onda cae de +5 V a GND. Este flanco también se denomina de ALTA a BAJA (H a L) del pulso de reloj, aunque también se conoce como " flanco negativo " o "flanco posterior " del pulso de reloj

OTROS MULTIVIBRADORES

MULTIVIBRADORES ASTABLES: RELOJES

Un multivibrador (MV) es un circuito generador de pulsos que produce una salida de onda rectangular, se clasifican en: astables, biestables o monoestables.

Los MV astables también se denominan " multivibradores autónomos ", el MV astable genera un flujo de pulsos continuos como lo vemos a continuación.

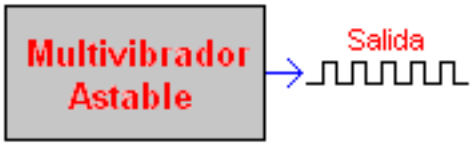


Figura 9: Multivibrador Astable

MULTIVIBRADORES BIESTABLES

Los MV biestables también se pueden llamar " flip-flops ". El MV biestable está siempre en uno de dos estados estables (set o reset). La idea básica de un MV biestable es que el pulso de entrada produzca en la salida un cambio de nivel BAJO al ALTO, como lo vemos a continuación.

Figura 10: Multivibrador biestable

MULTIVIBRADORES MONOESTABLES

Los MV mono estables también se denominan "multivibradores de un disparo". Cuando se dispara el monoestable, este produce un pulso de corta duración, como lo vemos a continuación.

Figura 11: Multivibrador monoestable

MEMORIAS DE LAS MICROCOMPUTADORAS

Las memorias de las microcomputadoras, son un ejemplo de la aplicación de los dispositivos de almacenamientos de datos llamados memorias.

El sistema de los MC esta compuesto por dispositivos de entradas como son los teclados, escáners, etc. ; y dispositivos de salida como son el monitor y la impresora. La unidad central de procesamiento CPU controla la operación de la MC y procesa los datos.

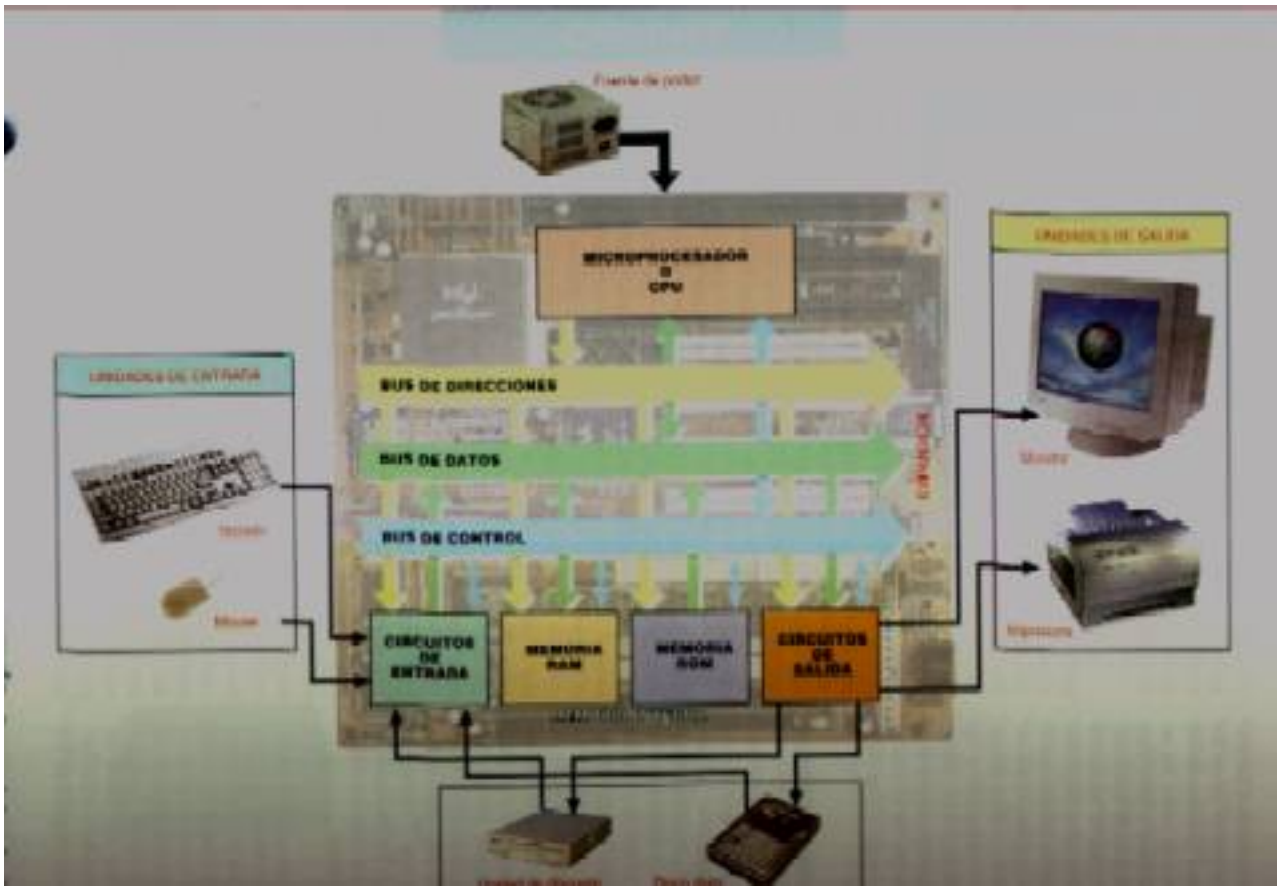


Figura 1: Microcomputadora

La memoria interna de una MC esta constituida por tres tipos de memoria semiconductoras. La memoria no volátil es llamada ROM (memoria de solo lectura) y la memoria volátil es llamada RAM (memoria de acceso aleatorio) .

Los dispositivos de almacenamiento RAM y ROM vienen en forma de CI y están moteados en tarjetas de circuitos impresos.

La mayoría de los datos son almacenados normalmente en dispositivos magnéticos de almacenamiento masivos, denominado disco flexible o disco duro .

RANDOM ACCESS MEMORY (RAM)

La memoria RAM es una memoria volátil muy utilizadas en los MC para almacenar los datos

temporalmente, y tiene características de volátil debido a que pierde los datos almacenados en ella cuando se desconecta de la alimentación. La RAM se denomina memoria de lectoescritura. La operación de ubicar un dato y visualizarlo se denomina lectura. Almacenar los datos se denomina escritura.

Figura 2: Memorias RAM

La siguiente tabla es una representación gráfica del interior de una memoria de 32 bits. Las 32 casilla están organizadas en ocho grupos de cuatro bits y cada grupo de cuatro bits es una palabra.

Dirección	Bit D	Bit C	Bit B	Bit A	Dirección	Bit D	Bit C	Bit B	Bit A
Palabra 0					Palabra 4				
Palabra 1					Palabra 5	1	1	E	1
Palabra 2					Palabra 6				
Palabra 3					Palabra 7				

Tabla 1: Representación de una memoria RAM de 8X4

Consideremos la memoria de figura como una RAM si la RAM estuviese en el modo de escribir, el dato (por ejemplo, 11 E 1) puede grabarse en la memoria en posición 5 si la RAM estuviese en el modo de leer , el dato puede ser leído en la posición indicada, la RAM también es llamada SCRATCH - PAD , esta memoria es llamada de acceso aleatorio debido a que puedes saltar de una palabra a otra en un solo paso.

La siguiente figura muestra el diagrama lógico de un sencillo CI RAM 74F189 TTL de 64 bits, este CI RAM esta construido con tecnología Schottky TTL más moderna, FAST, una subfamilia que muestra una combinación de rendimiento y eficiencia no alcanzada por otras familias TTL.

Figura 3: Diagrama lógico del CI 74F189

Uno de los modos de operación del 74F189 es el modo de escritura. Durante esta operación los 4 bits ubicados en la entrada de datos (D3 , D2 , D1 , D0) se escriben en la posición de la memoria especificada por las entradas de dirección. Por ejemplo, para escribir 11 E 1 en la posición de la palabra 5 las entradas de datos deben ser D3 = 1, D = E y D0 = 1 y las entradas de dirección deben ser A3 = E , A2 = 1 , A1 = AE = 1. Igualmente la entrada de habilitación de escritura WE debe estar en un nivel bajo y la entrada de selección de pastilla CS debe estar en BAJO.

Otro modo de operación es el modo de lectura para la RAM 74F189. Las entradas de control C5 deben estar en un nivel BAJO y WE en ALTO.

A continuación el contenido de la posición direccionada aparecerá en la salida de datos (O3, O2, O ,Oo) . debe entenderse que la operación de lectura uno destruye el dato almacenado , sino que saca una copia invertida de ese dato.

También encontramos el modo de almacenamiento , (store) o de inhibición.

Existiendo tipos de RAM básicos , el estático y el dinámico. Un ejemplo de RAM estático es la CI 74F189 esta RAM estática pueden fabricarse utilizando tecnología bipolar o MOS. La RAM estática utiliza un flip-flop, celda de memoria, y conserva la información siempre que la alimentación este conectada al integrado.

La RAM dinámica o DRAM son utilizados como unidades de gran capacidad de memoria, una celda de RAM dinámica esta basada en un dispositivo MOS que al almacenar un carga como lo haría un capacitor. Un inconveniente es que todas las celdas deben ser recargadas cada pocos milisegundos para que no pierda los datos.

Los sistemas basados en microprocesador como las microcomputadoras, convenientemente adecuadas almacena y transfiere los datos en grupos de ocho bits llamadas bits.

Una SRAM muy popular es la 2114 fabricada con tecnología MOS almacena 4.096 bits organizados en 1.024 palabras de 24 bits cada una.

Con la conexión de dos RAM 2114 pueden conformar una memoria de 1.024 palabras de ocho bits por palabra, a esto se le domina 1 KB de memoria.

Comúnmente las RAM son valoradas mencionando algunas características distintivas como son:

1. Tamaño (en bits) y organización (palabra x bits por palabra ; por ejemplo ; la RAM 2114 seria de 4096 bits, o 1024 x 4
2. Tecnología utilizadas para la fabricación del clip por ejemplo; NMOS para la RAAM 2114
3. Tipo de salida ; esta ser ambas RAM para 2114, o como en otras que tienen

salida de 3 estados

4. Velocidad (tiempo de acceso de la memoria) para la RAM 2114 es de unos 50 a 450ns
5. Tipo de memoria (SRAM ó DRAM)

READ ONLY MEMORY (ROM)

Generalmente las microcomputadoras debe almacenar informático permanentemente en forma de programas, en una memoria de solo lectura ROM.

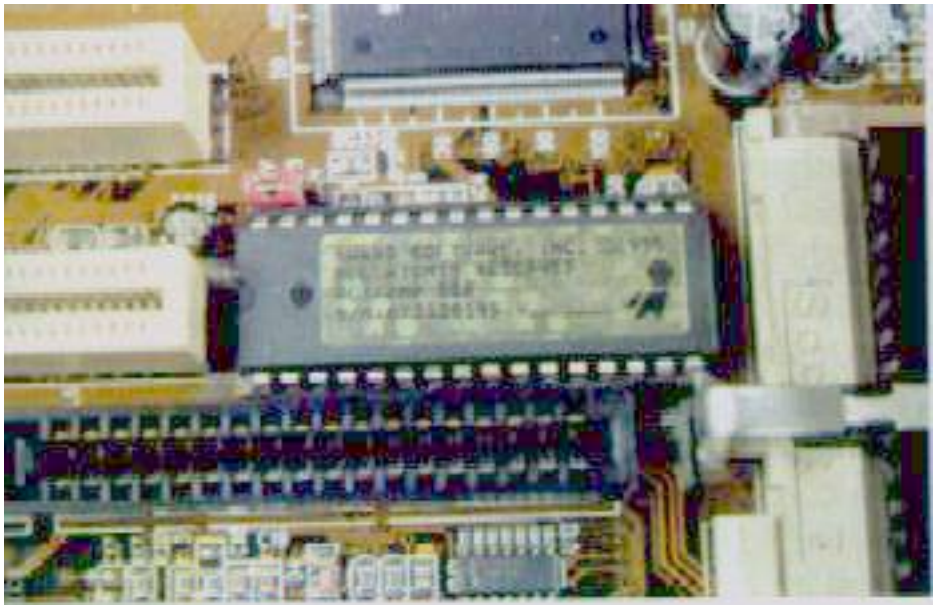


Figura 4: Memoria ROM de una microcomputadora

Las ROM son memorias no volátiles porque no pierden sus datos cuando es desconectadas de la alimentación . las ROM son utilizadas en aplicaciones de alto volumen de almacenamiento. Para aplicaciones de bajo volumen que utilizan diversas memorias de solo lectura programables (PROM).

Anteriormente eran utilizadas las ROM fabricada con diodos, pero estas tenían muchos inconvenientes debido a que sus niveles lógicos eran marginales y tenían una capacidad de conexión muy limitada pues no tenían buffers de entrada y de salidas, que son necesarios para trabajar con buces de datos y direcciones.

Actualmente las ROM pueden variar desde muy pequeñas unidades hasta ROM de gran capacidad. Las ROM se fabrican utilizando tecnologías TTL, CMOS, NMOS, PMOS y GaAs (Arsenuro de calcio).

La tecnología GaAs . consigue CI muy rápidos, actualmente las ROM que utilizan tecnología CMOS y NMOS son las mas populares podemos citar como ejemplo la ROM NMOS 512 x 8 82HM141C de Harris con un tiempo de acceso de menor 70ns. Una similar fabricado con GaAs es la 146M048 de Tri Quint semiconductor, con velocidad de 1.5ns. una ROM muy popular es TMS47256 es una NMOS 262 de 144 bit organizadas como 32.768 bytes. Desde un punto de vista practico se denomina como ROM de 32 kbytes.

MEMORIA PROGRAMA DE SOLO LECTURA (PROM)

Se disponen de PROM que acortan los tiempos de desarrollo y de costos mas bajos. En estas es mucho mas fácil de corregir errores de programa y actualizar los productos debido a que pueden ser reprogramados por el usuario .

Existe una variedad de PROM entre los cuales se pueden mencionar los siguientes :

1. PROM borrables (E PROM). Está esta dotada de una ventana de cuarzo especial en la parte superior del encapsulado ; la pastilla es borrada exponiendo el CI a los rayos ultravioletas (UV) dejando las celdas de memoria a 1 lógico, para luego ser reprogramado , es de tipo de EPROM es conocida como PROM borrables UV.
2. PROM electrónicamente borrables (EEPROM ó E2PROM). Debido a que este tipo de PROM son borrables eléctricamente , es posible borrarlas y reprogramarlas muestran permanecen en el circuito . también cabe notar que en estas PROM se borra solo un byte a la vez.
3. EPROM flash, este tipo de PROM también puede ser borrada estando en el circuito impreso, pero una diferencia es que la EPROM flash se borra por completo y luego se reprograma , y tiene una ventaja y es que debido a la EPROM flash es mas moderna , la unidad de almacenamiento es mas sencilla y por eso puede almacenar una información en una unidad mas pequeña .

Una popular familia de EPROM es la 27XX ; fabricadas por compañías como Intel, Advanced Micro Devices y Fujitsu Micro Electronics, Inc. Algunas de las más importantes de la serie 27XX son :

MEMORIA	CAPACIDAD
2716	16Kbits (8 x 2KB)
2732	32Kbits (8 x 4KB)
2764	64Kbits (8 x 8KB)
27128	128Kbits (8 x 16KB)
27256	256Kbits (8 x 32KB)

Tabla 2: Memorias UV-EPROM de la serie 27XX

Un ejemplo de CI de la serie 27XX de la familia EPROM es la PROM borrables - UV de·32K (4K x 8) 2732A . La 2732A tiene 12 pastillas de dirección (Ao - A11) que pueden acceder a las 4096 bytes de memoria. Tiene ocho pastillas de salida etiquetadas como Oo - O_7 .

Frente a las ROM las RAM tienen una gran desventaja , el ser volátiles. para resolver este problema , se han desarrollado las RAM no volátiles.

Actualmente las RAM no volátiles se implementa de dos formas :

1. usando una SRAM CMOS con una batería de seguridad , esto se puede hacer debido a que la tecnología CMOS tienen un bajo consumo de potencia. Normalmente se usa una batería de larga vida como una batería de litio. El sistema es activado por medio de un comparador, que cuando la alimentación normal de SRAM falla, este activa la conexión con la batería que se encuentra en un modo de Stand By.
2. Usando una RAM estática no volátil (NVS RAM). Este es un producto mas moderno , que tiene las capacidades de lectura / escritura y su diseño no requiere de una batería. ejemplo típico de NVS RAM es la CMOS STK10C68 producida por Simtek, esta organizada como una memoria de 8 KB x 8 para acudir a 8192 . La NV SRAM STK10C68 usa líneas de dirección ($A_{o\ a}\ A_{12}$) para acudir 8192 palabras de bits, el tiempo de acceso que maneja es de mas 25ns

MEMORIA MASIVAS DE LAS MICROCOMPUTADORAS

Muchas veces los programas y los datos almacenados en la microcomputadora se clasifican en internos y / o Externos.

En una MC , de los dispositivos de almacenamiento interno son las RAM , ROM (ó EPROM) semiconductoras y diversos registros .

Actualmente la forma común de almacenamiento externo son los discos magnéticos ; subsidiados en discos duros y flotantes

Los datos se almacenan en los discos flotantes, de la misma forma que en las cintas magnéticas ; esto es como grabar y escuchar una cinta. Pero hay una cinta, pues el disco en dispositivo de acceso aleatorio , en cambio, la cinta es de acceso secuencial lo que hace el acceso mucho mas lento, en comparación a la velocidad con que accede en un disco.

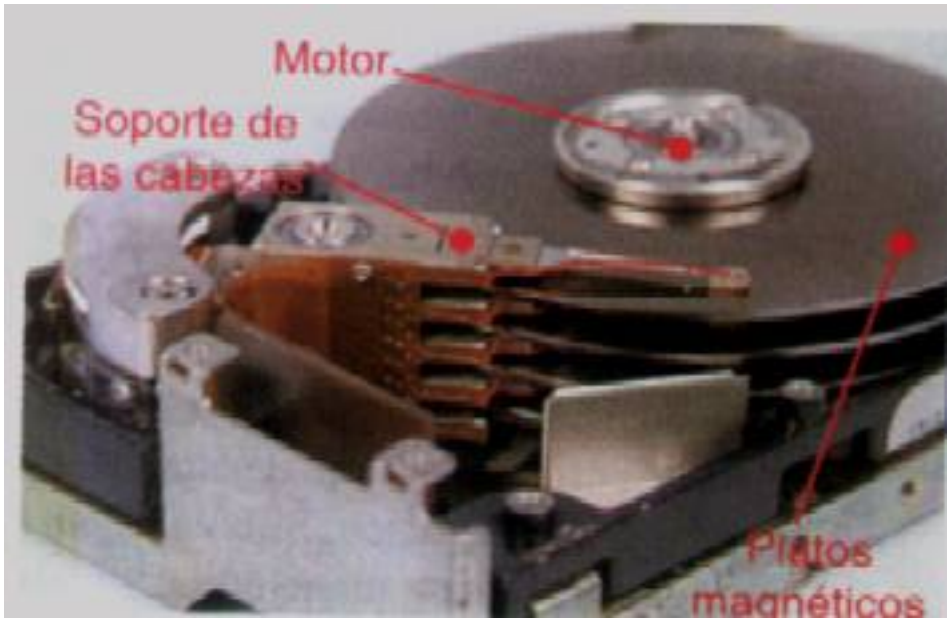


Figura 5: Disco duro

SELECTORES DE DATOS/MULTIPLEXORES.

Es la versión electrónica de un conmutador rotatorio en un solo sentido, se puede comparar con un selector mecánico en una sola dirección. También se puede definir como un proceso de selección de una entrada entre varias y la transmisión de los datos seleccionados hacia un solo canal de salida.

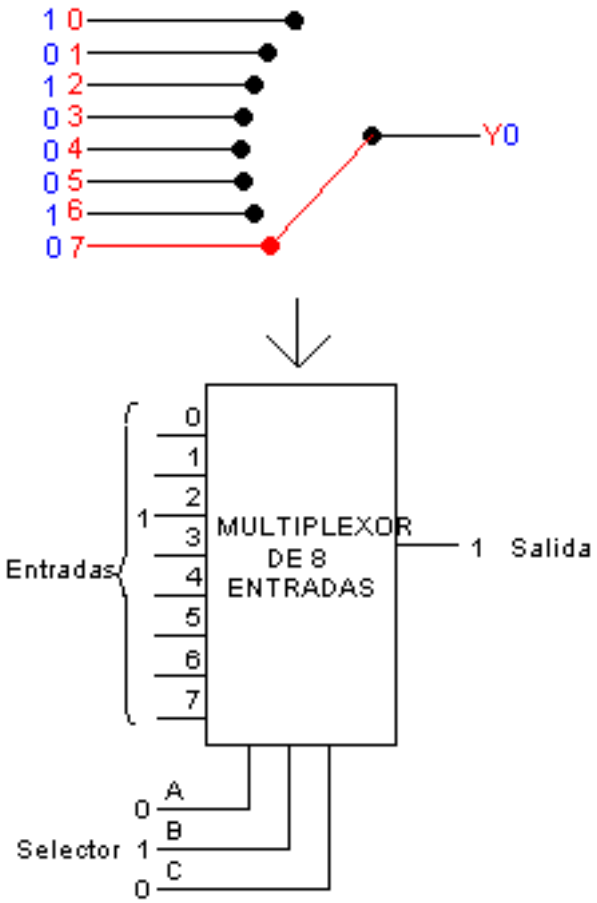


Figura 1: Selector de datos

En la figura 1, se compara un selector mecánico de datos y un selector electrónico de datos. En el primer caso la selección del dato se logra girando mecánicamente el rotor del conmutador, y en el selector electrónico de datos multiplexor se selecciona el dato colocando el numero binario adecuado en las entradas de selección de datos A, B, C.

A continuación se ilustra el multiplexor comercial TTL 74150 que tiene las siguientes

características:

1. Consta de 16 entradas de datos.
2. Tiene una única salida invertida w (pin 10).
3. Posee cuatro entradas selectoras de datos de A a D (pin 15 al 11).
4. Tiene una entrada de habilitación denominada STROBE que se considera como un conmutador ON-OFF.

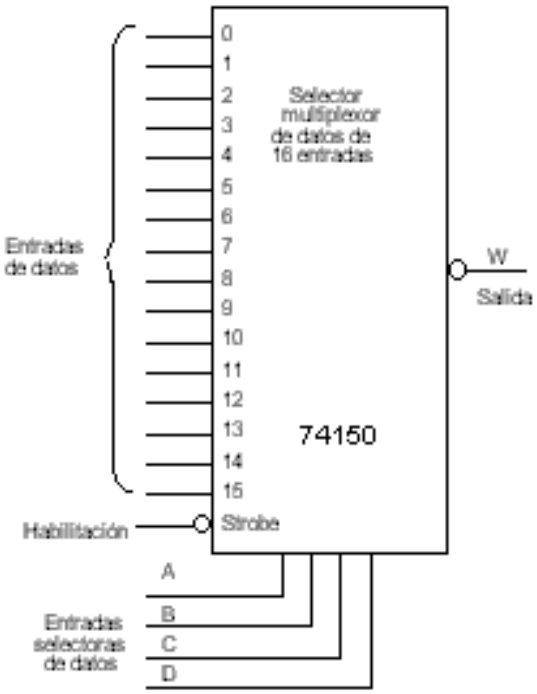


Figura 2: Selector de datos 74150

La tabla de verdad del selector de datos 74150 nos muestra en su primera línea la entrada de habilitación (STROBE) en alto lo cual no habilita ningún dato, sea cualquiera la entrada de selección, como resultado obtendremos en la salida una tensión alta. En la segunda línea tenemos las entradas de habilitación en bajo lo cual habilita las entradas selectoras de datos que en este caso están en bajo por lo cual en la salida obtendremos la entrada E.

D	C	B	A	Strobe	W
X	X	X	X	H	H
L	L	L	L	L	E0
L	L	L	H	L	E1

L	L	H	L	L	E2
L	L	H	H	L	E3
L	H	L	L	L	E4
L	H	L	H	L	E5
L	H	H	L	L	E6
L	H	H	H	L	E7
H	L	L	L	L	E8
H	L	L	H	L	E9
H	L	H	L	L	E10
H	L	H	H	L	E11
H	H	L	L	L	E12
H	H	L	H	L	E13
H	H	H	L	L	E14
H	H	H	H	L	E15

Tabla 1: Tabla deverdad del 74150

En la tercera línea además de tener la entrada de STROBE activado en BAJO tenemos en las entradas selectoras de datos LLLH lo cual nos da una salida de E1 y así sucesivamente hasta llegar en las entradas selectoras de datos HHHH que corresponde en la salida a E15.

Este CI tiene muchas aplicaciones como la solución de problemas lógicos difíciles de simplificar. Como ejemplo puede mostrar la figura a continuación donde necesitaríamos muchos CI de lógica combinacional para implementar este circuito.

Línea	Entradas				Salida
	D	C	B	A	Y
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	1	0	0
4	0	0	1	1	1
5	1	1	0	0	0
6	1	1	0	1	0
7	1	1	1	0	1
8	1	1	1	1	0
9	0	0	0	0	0
10	0	0	0	1	1
11	0	0	1	0	1
12	0	0	1	1	0
13	1	1	0	0	1
14	1	1	0	1	0
15	1	1	1	0	0
16	1	1	1	1	1

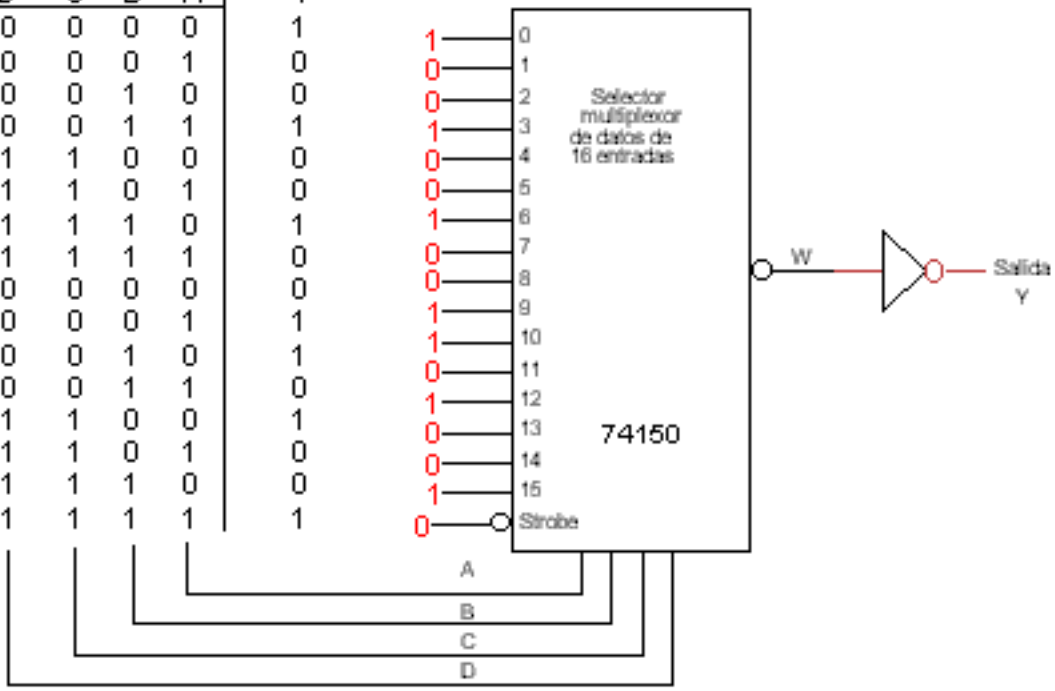


Figura 3: Solución de problemas lógicos con multiplexores

Además de todo lo anterior el CI 74150 se puede utilizar para transmitir una palabra paralela de 16 bits en forma serie esto se realiza conectando un contador a las entradas de selección de datos y se cuenta desde 0000 hasta 1111, esta puede ser una palabra paralela de 16 bits en las entradas de datos de 0 a 15. Finalmente esta se transmite a la salida en forma serie o sea de dato por vez.

Visualizar la multiplexación.

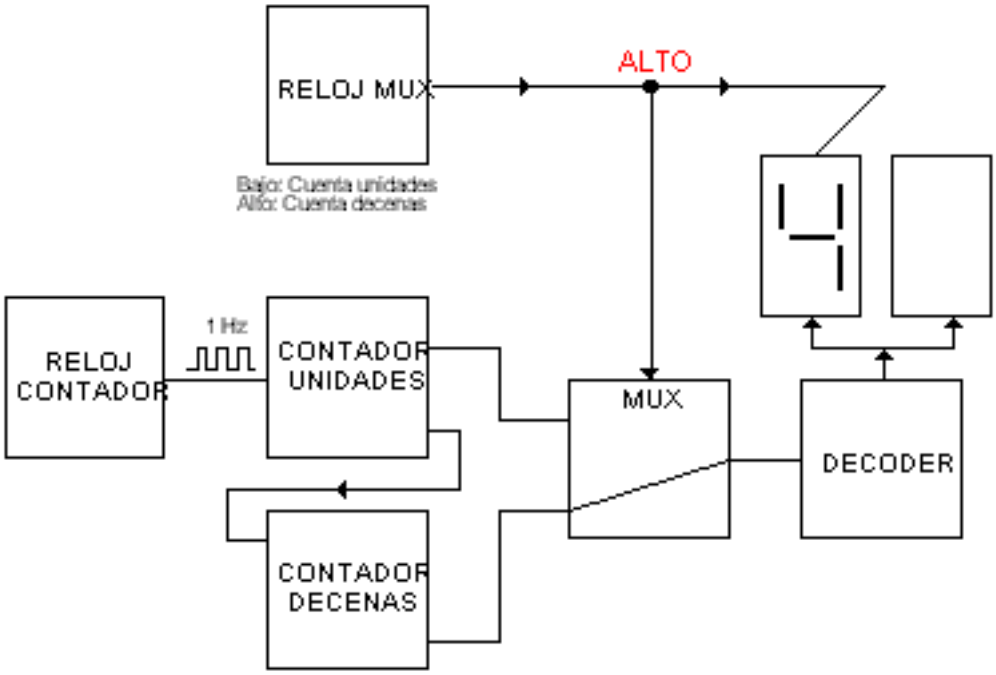
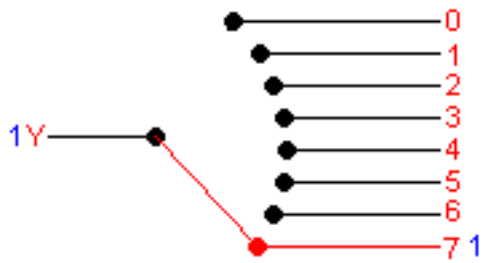


Figura 4: Aplicación de la multiplexación

En este ejemplo, el multiplexaje reduce el consumo de potencia de los visualizadores y elimina la necesidad de un decodificador extra. El multiplexor trabaja a una frecuencia de 100 Hz que activa alternativamente el conteo de las unidades o de las decenas.

DEMULTIPLEXORES.

El demultiplexor (DEMUX) invierte la operación del multiplexor, el DEMUX tiene una sola entrada de datos que en la salida puede ser distribuida a cualquier canal.



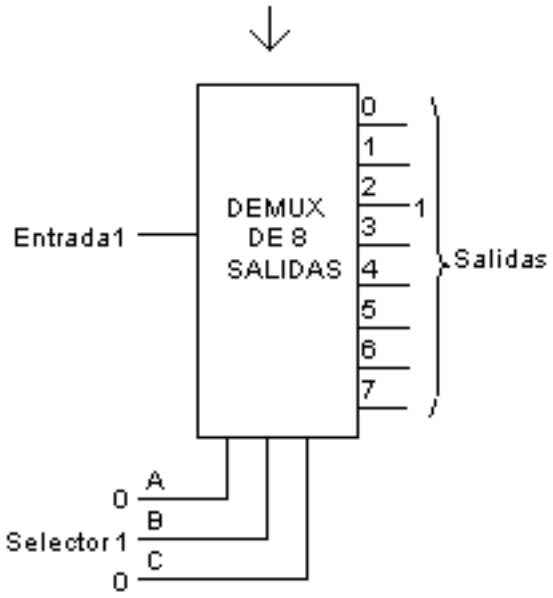


Figura 5: Demultiplexor

El DEMUX también se denomina decodificador y a veces distribuidor de datos, el DEMUX solo permite que los datos fluyan de la entrada a las salidas y no en ambas direcciones.

Los DEMUX están disponibles en versiones TTL y CMOS de una entrada y cuatro salidas, una entrada y ocho salidas, una entrada y diez salidas y una entrada y dieciséis salidas.

El CI decodificador/demultiplexor de 4 a 16 TTL 74LS154 tiene dos entradas de datos G1 y G2 que activan a una única entrada en el nivel BAJO.

La figura 6 muestra el DEMUX 74LS154 que tiene 16 salidas de 0 a 15 con 4 entradas de datos (D a A) sus salidas son activas en bajo por lo que normalmente están en alto y cuando se activan están en bajo, además como se había dicho antes tiene dos entradas de datos G1 y G2 negados que realizan la operación NOR para generar la única entrada de datos lo que quiere decir que para poder activar un dato deben estar los dos en bajo.

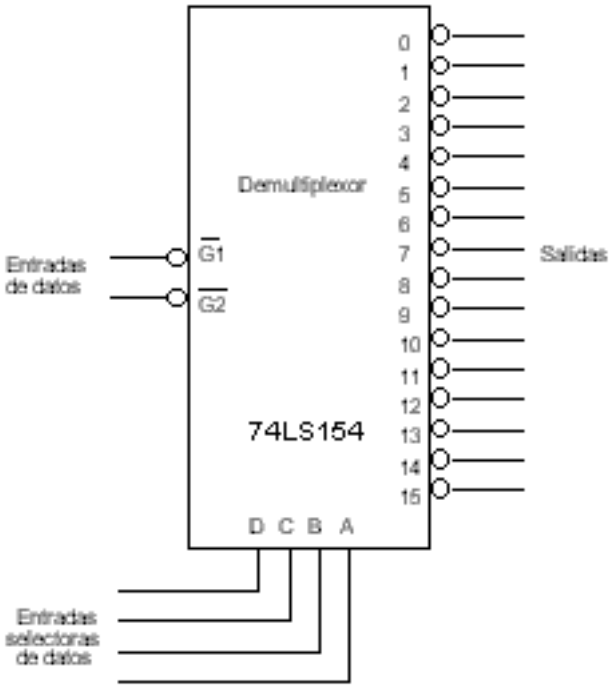


Figura 5: Demultiplexor 74154

El DEMUX 74LS154 pertenece a la familia TTL schottky de baja potencia y mayor velocidad que tiene un retardo de propagación de 30 ns.

G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L

L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

Tabla 2: Tabla de verdad del demultiplexor 74154

CERROJOS Y BUFFERS DE TRES ESTADOS.

En los sistemas digitales a veces se hace necesario tener memorizado o detenido un dato por algún tiempo que sea necesario. Analicemos la figura a continuación.

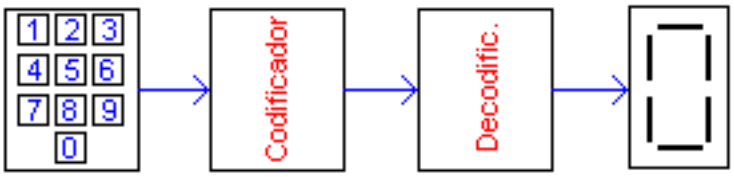


Figura 6: Circuito sin cerrojos

En este ejemplo si se deja de presionar el dígito este se borra del visualizador, este problema que se presenta en este sistema digital se puede solucionar anexando un cerrojo o también llamado memoria antes del decodificador.

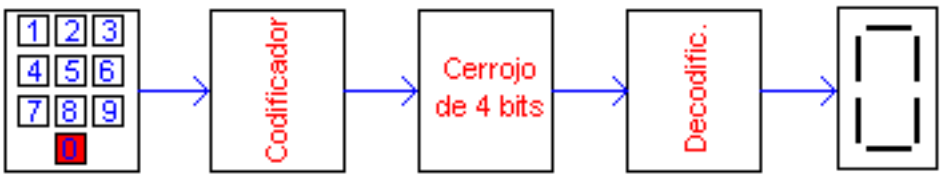


Figura 7: Circuito con cerrojos

En la figura 8 se detalla un sencillo cerrojo fabricado en forma de CI cerrojo transparente de cuatro bits TTL 7475 este diagrama lógico nos muestra que CI 7475 tienen cuatro entradas que aceptan datos en paralelo, los datos Do a D3 pasan a través del 7475 a sus salidas normal y complementaria, cuando las entradas de habilitación de datos están en alto y se dice que el cerrojo es transparente, ya que cualquier cambio en los datos de entrada se transmite de inmediato a la salidas.

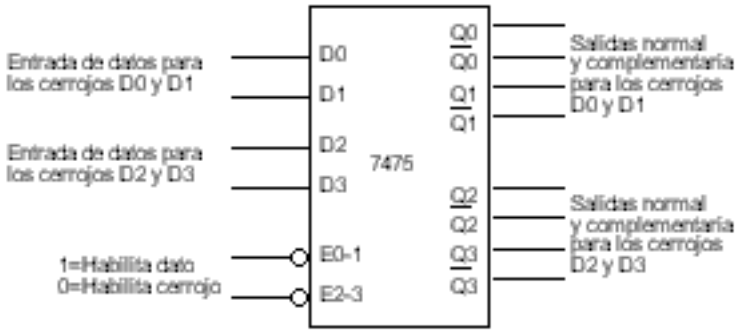


Figura 8: Cerrojo 7475

Cuando las entradas de habilitación están en bajo el dato esta encerrado o mantenido en las salidas por lo que las variaciones en las entradas no afectan las salidas. Este cerrojo se considera un registro de entrada paralela/salida paralela.

Modo de operación	Entradas		Salidas	
	E	**D**	**Q**	**1**
Datos habilitados	1	0	0	1
	0	1	1	0
Datos encerrados	0	X	No cambia	

Tabla 3: Tabla de verdad del cerrojo 7475

Para los sistemas basados en microprocesador (Microcomputadoras) se utiliza un bus de datos bidireccional para transferir los datos entre los dispositivos.

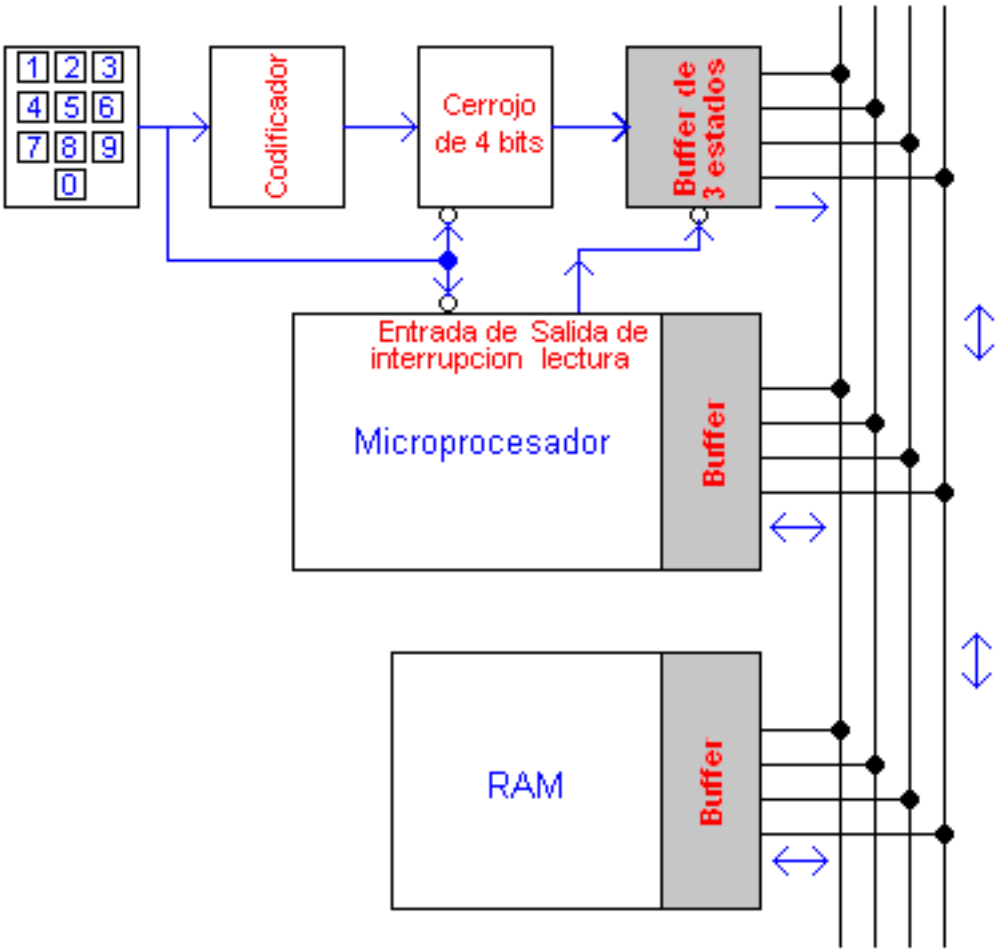


Figura 9:Buffers utilizados para aislar dispositivos de un bus de datos

En la figura 9 se ilustra un sencillo sistema basado en un microprocesador que utiliza un bus de datos bidireccional de 4 bits. Para que el bus de datos pueda funcionar correctamente cada dispositivo debe estar aislado del bus, utilizando un buffer de tres estados, se ilustra un teclado de entrada familiar con un buffer de tres estados para desconectar del bus de datos el dato encerrado, exceptuando el corto intervalo de tiempo durante el cual el microprocesador envía una señal de nivel de bajo de lectura.

Cuando se activa la entrada de control del buffer c, el dato encerrado activa las líneas del bus de datos del nivel alto al nivel bajo dependiendo el dato presente. Después el microprocesador retira ese dato del bus de datos y desactiva el buffer (el control vuelve al nivel alto).

El buffer de tres estados mostrado en forma de bloques en la figura anterior puede implementarse utilizando el CI TTL 74125, cuádruple buffer de tres estados.

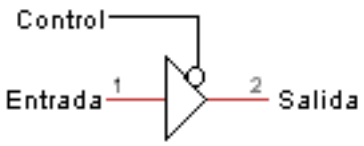


Figura 10: Símbolo de un buffer de tres estados

En la electrónica digital existen dos valores 0 y 1 el buffer de tres estados logra la creación de un nuevo estado, es el estado de alta impedancia donde la salida no va tener tensiones altas ni bajas, sino un aislamiento de protección para los elementos de un circuito, que normalmente se utiliza para realizar interconexiones con los buses en las microcomputadoras como parte de microprocesador y de RAM. Actualmente están disponibles muchos dispositivos denominados adaptadores de interfaces de periféricos (PIA) que contienen cerrojos, buffers, registros y líneas de control, están disponibles para cada microprocesador y cuidan las necesidades de entrada y salida del sistema.

Los cerrojos se encuentran en las familias TTL y CMOS, vienen normalmente en versiones de flip-flops D de 4 u 8 bits, algunos tienen salidas de tres estados. Los cerrojos que hemos estudiado sólo permiten que los datos fluyan de la entrada a la salida pero el transceptor de bus es la diferencia, ya que este dispositivo permite que los datos fluyan en ambas direcciones, así como los que se presentan en la figura.

TRANSMISIÓN DIGITAL DE DATOS

Este es el proceso de enviar información de un lugar a otro del sistema que pueden estar próximos o separados, se puede hacer de dos formas: paralela o serie.

- **PARALELA:** Se utiliza mucho en los sistemas basados en microprocesador como los microprocesadores donde números enteros de bits se transmiten al mismo tiempo, pero se necesitan muchos registros, cerrojos y conductores. Este se utiliza cuando la velocidad o el tiempo es muy importante.

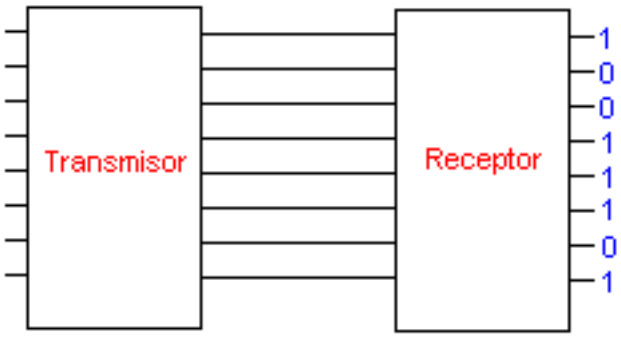


Figura 11: Transmisión de datos paralelo

- **SERIE:** Este sistema sólo utiliza una línea de transmisión y se utiliza cuando se transmiten datos a largas distancias, se transmite primero un bit de arranque en el nivel 0 luego se transmiten los siguientes 7 bits de datos, un bit de paridad para la detección de errores y finalmente 2 bits de parada en el nivel alto.

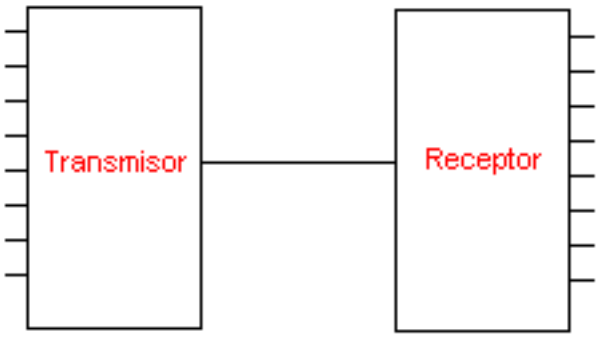


Figura 12: Transmisión de datos en serie

ARRAYS LOGICOS PROGRAMABLES (PLA)

Son circuitos integrados cuya característica principal consiste en brindarnos muchas entradas y salidas en un solo dispositivo; generalmente son utilizados en lógica convencional.

La forma de programación de estos dispositivos es muy sencilla. Contienen un sistema de fusibles que en el momento de ser suministrados por el fabricante se encuentran intactos, para luego realizar la programación quemando los fusibles pertenecientes a la línea que no se van a utilizar, dejando conectadas las líneas que se van utilizar, dependiendo a la expresión booleana; así que si tenemos las líneas A, Å; B y B, conectadas a una puerta

AND al quemar los fusibles de las entradas Å y B, solo quedarían activadas las entradas A y B, produciendo una salida en la puerta AND de A·B.

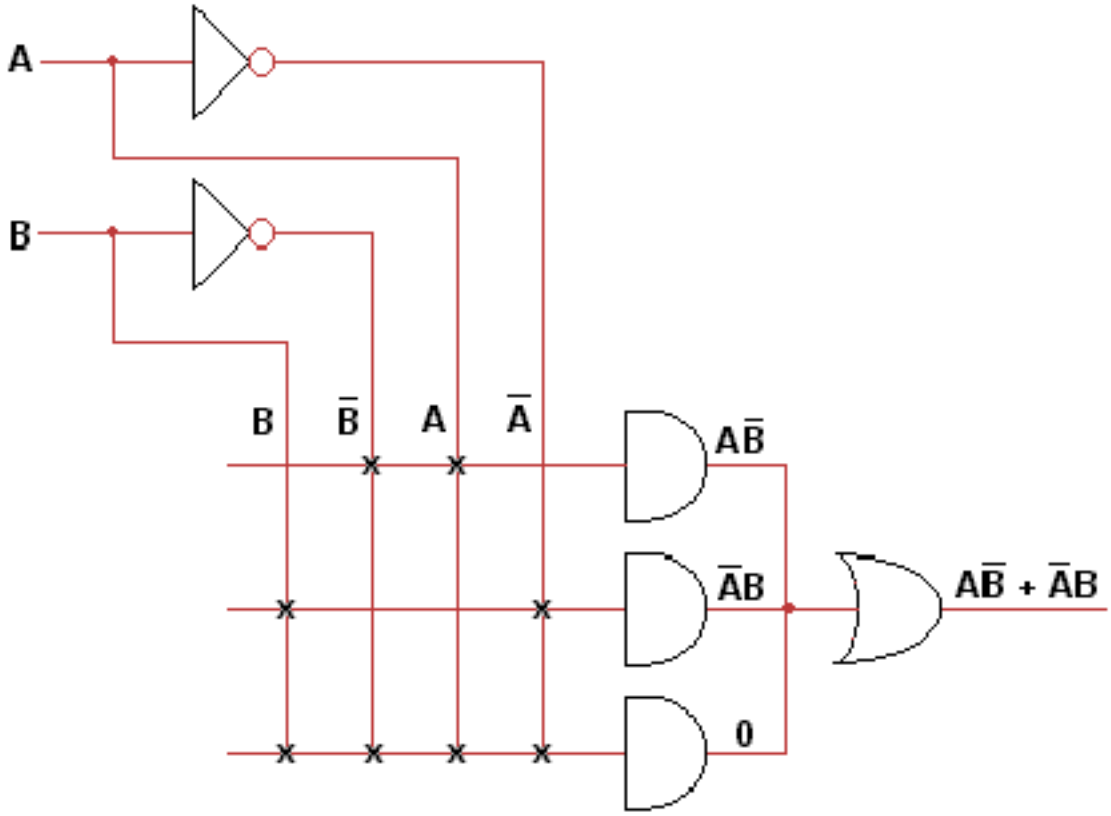


Figura 13: Notación abreviada de los PLA

COMPARADORES DE MAGNITUD

Son dispositivos que compraran dos números binarios produciendo una salida de comparación como A = B, A > B ó A < B, un circuito integrado comparador característico es el 74HC85 , que es un comparador de 4 bits. Estos dispositivos se pueden utilizar en aplicaciones como comparar magnitudes de variables como la de la temperatura de un horno, y compararla con una temperatura referencial, luego deducir si mantiene una temperatura adecuada, poder variarla o ajustarla.

DISPOSITIVOS DISPARADORES SCHMITT

Son dispositivos utilizados para convertir las ondas seno en ondas cuadradas, ya que estas ultimas tienen tiempos de subida y de bajada bastante rápidos.

Estos dispositivos también ofrecen una mayor histéresis por lo que se incrementa una mayor inmunidad al ruido debido a la diferencia que hay entre la tensión umbral de subida y la tensión umbral de bajada.

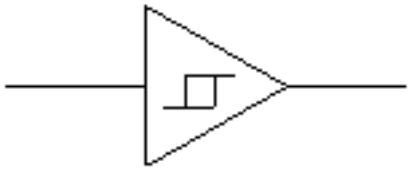


Figura 14: Símbolo de un disparador Schmitt

La tensión umbral es aquella tensión de entrada en cuya salida cambia de un estado alto a bajo ó bajo a alto.

PUERTAS LOGICAS

La puerta lógica es el bloque de construcción básico de los sistemas digitales. Las puertas lógicas operan con números binarios. Por tanto las puertas lógicas se denominan puertas lógicas binarias.

En los circuitos digitales todos los voltajes, a excepción de los voltajes de las fuentes de potencia, se agrupan en dos posibles categorías: voltaje altos y voltajes bajos. No quiere decir esto que solo se encuentren dos voltajes, si no que cierto rango de voltajes se define como alto y otro cierto rango como bajos. Entre estos dos rangos de voltajes existen existe una denominada zona prohibida o de incertidumbre que los separa.

Una tensión alta significa un 1 binario y una tensión baja significa un cero binario.

Todos los sistemas digitales se construyen utilizando tres puertas lógicas básicas. Estas son las puertas AND, la puerta OR y la puerta NOT.

LA PUERTA AND.

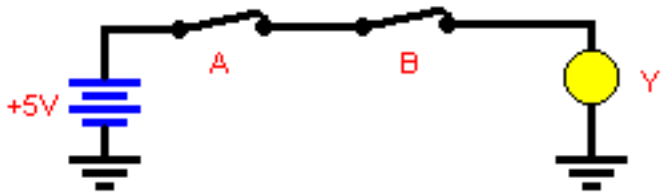


Figura 1: Circuito equivalente de una puerta AND

La puerta AND es denominada la puerta de << Todo o Nada >> . observar el esquema de la figura 1, la cual muestra la idea de la puerta AND. Examinando de cerca el circuito, notamos que la lampara encenderá solo si ambos interruptores se cierran o se activan simultáneamente. Si uno de los de los interruptores esta abierto, el circuito se interrumpe y la lampara no se enciende. Todas las posibles combinaciones para los interruptores A y B se muestran en la tabla 1 . La tabla de esta figura que la salida (y) esta habilitada (encendida) solamente cuando ambas entradas están cerradas.

Interruptores de entrada		**Luz de salida**
A	**B**	**Y**

Abierto	Abierto	Apagado
Abierto	Cerrado	Apagado
Cerrado	Abierto	Apagado
Cerrado	Cerrado	Encendido

Tabla 1: Combinaciones posibles de la compuerta AND

Con el ánimo de presentar en forma mas compacta la tabla, anterior, convengamos en que la condición de interruptor cerrado la representamos con un 1, y la de interruptor abierto con un 0. De manera similar, el encendido de la lampara la representamos con un 1. Y su apagado con un 0 (cero). Con estas convenciones, la tabla 1 nos quedaría como en la tabla 2.

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 2: Tabla 1 simplificada

LOS SÍMBOLOS DE LAS COMPUERTAS

Son una representación gráfica de la función que ayuda a visualizar las relaciones lógicas existente en un diseño o circuito. En la figura 2 se muestra el símbolo de la compuerta AND con lo que se quiere significar que esta compuerta AND es un dispositivo que posee dos entradas A y B y una salida Y.

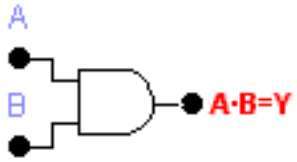


Figura 2: Símbolo de una compuerta AND

El álgebra booleana es una forma de lógica simbólica que muestra como operan las compuertas lógicas. Una expresión booleana es un método << taquígrafo >> de mostrar que ocurre en un circuito lógico. La expresión booleana para el circuito de la figura 3 es.

A · B = Y

Figura 3: Expresión booleana de la compuerta AND

La expresión booleana se lee A AND B igual a la salida Y. El punto (·) significa la función lógica AND en álgebra booleana, y no la operación de multiplicar como en el álgebra regular.

Con frecuencia un circuito lógico tiene tres variables. La fig. 4 muestra una expresión booleana para una puerta AND de tres entradas. El símbolo lógico para esta expresión AND de tres entradas esta dibujada en la fig. 5. La tabla de verdad 3 muestra las 8 posibles combinaciones de la variables a, b y c observar que solo cuando todas las entradas están en 1 y la salida de la puerta AND se habilita a 1.

A · B · C = Y

Figura 4: Expresión booleana para una compuerta AND de tres entradas

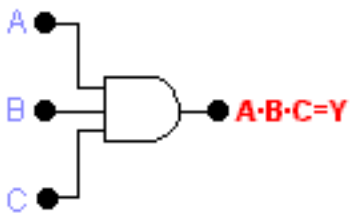


Figura 5: Compuerta AND de tres entradas

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Tabla 3: Tabla de verdad de una compuerta AND de tres entradas

LA PUERTA OR

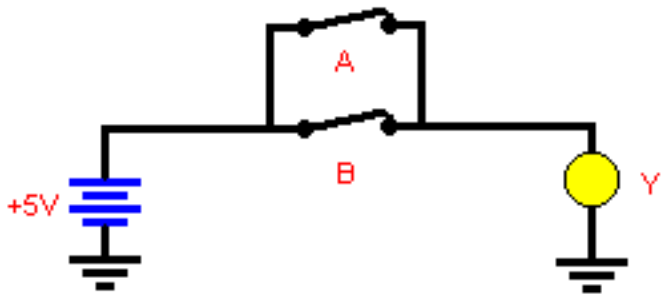


Figura 6: CIrcuito equivalente de una compuerta OR

La puerta OR se denomina y la puerta de << cualquiera o todo >>. El esquema de la figura 6 nos muestra la idea de la puerta OR, en el cual los interruptores han sido conectados en paralelo. El encendido de la lampara se producirá si se cierra cualquiera de los dos interruptores o ambos. Todas las posibles combinaciones de los interruptores se muestran en la tabla 4. La tabla de verdad detalla la función OR del circuito de interruptores y lampara.

Interruptores de entrada		Luz de salida
A	**B**	**Y**
Abierto	Abierto	Apagado
Abierto	Cerrado	Encendido
Cerrado	Abierto	Encendido
Cerrado	Cerrado	Encendido

Tabla 4: Combinaciones posibles de la compuerta OR

La tabla de la 4 describe el funcionamiento del circuito. Observamos, que de las 4 posibles combinaciones de cierre y apertura de los interruptores, 3 de ellas producen el encendido de la lampara , y de nuevo utilizando la convención de representar la condición cerrado o encendido por un 1 y la de abierto o apagado por un 0, se obtiene la tabla de verdad de la tabla 5.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 5: Tabla de verdad de una compuerta OR de dos entradas

El símbolo lógico estándar para la puerta OR esta dibujado en la fig. 7. observar la forma diferente de la puerta OR. La expresión booleana abreviada para esta función OR es A + B = Y observar que símbolo + significa OR en álgebra booleana . la expresión (A+ B = Y) se lee A OR B igual a salida Y .

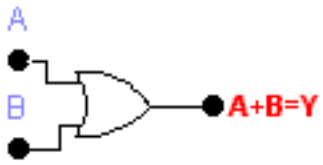


Figura 7: Símbolo de una compuerta OR

La expresión booleana , símbolo y tabla de verdad de una puerta OR de tres entradas o variables están dibujadas en las figuras 8, 9, y en tabla 6.

A + B + C = Y

Figura 8: Expresión booleana para una compuerta OR de tres entradas

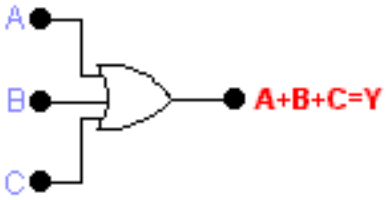


Figura 9: Compuerta OR de tres entradas

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Tabla 6: Tabla de verdad de una compuerta OR de tres entradas

LA PUERTA NOT

Las dos compuertas descritas anteriormente poseen cada una dos entradas y una salida. La compuerta NOT o inversora, posee una entrada y una salida como se muestra en la fig. 10. Su función es producir una salida inversa o contraria a su entrada es decir convertir unos a ceros y ceros a unos . la tabla de verdad 7 resume el funcionamiento de esta compuerta .

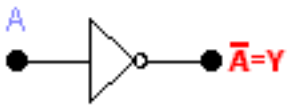


Figura 10: Símbolo de una compuerta NOT

A	Y
0	1
1	0

Tabla 7: Tabla de verdad de una compuerta NOT

La expresión booleana para la inversión es Å = A. La expresión Å = A indica que A es igual a la salida no A. Un símbolo alternativo para la puerta NOT o inversor , se muestra a continuación .

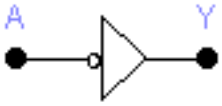


Figura 11: Símbolo alternativo de una compuerta NOT

El circulo inversor puede estar en la parte de entrada o de salida del símbolo triangular. cuando el circulo inversor aparece en la parte de la entrada del símbolo NOT, el diseñador

habitualmente intenta sugerir que esta una es una señal activa en baja . una señal activa en baja requiere que una tensión baja active alguna función en circuito lógico .

LA PUERTA NAND

Una compuerta NAND es un dispositivo lógico que opera en forma exactamente contraria a, una compuerta, AND, entregando una salida baja cuando todas sus entradas son altas y una salida alta mientras exista por lo menos un bajo a cualquiera de ellas .

Considerar el diagrama de los símbolos lógicos de la fig. 12, una puerta AND esta conectada a un inversor. Las entradas A y B realizan la función AND y forma la expresión booleana A · B la puerta NOT invierte A · B a la derecha del inversor se añade la barra de complementaron a la expresión booleana obteniéndose $\overline{A \cdot B} = Y$ a este circuito se denomina NOT-AND o NAND.

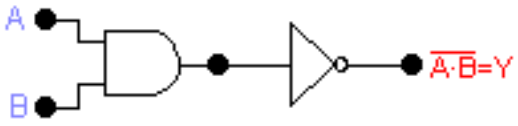


Figura 12: Circuito equivalente de una compuerta NAND

El símbolo lógico convencional para la puerta se muestra en el diagrama de la fig. 13 observar que el símbolo NAND es símbolo AND con un pequeño circulo a la salida. El circulo a veces se denomina circulo inversor. Esta es una forma simplificada de representar la puerta NOT . la tabla de verdad describe la operación exacta de la puerta lógica . la tabla de la verdad para la puerta NAND se ilustra en la tabla 8, observe como sus salida son las inversas de las salidas de la puerta AND .

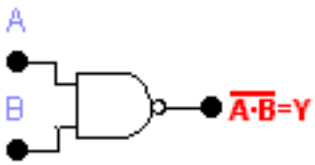


Figura 13: Símbolo lógico de una compuerta NAND

A	B	NAND	AND
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

Tabla 8: Tabla de verdad de una compuerta NAND de dos entradas

La operación de una puerta NAND es análoga a la del circuito eléctrico mostrado en la fig. 14 los interruptores A y B representan las entradas de la puerta y la lampara (Y) su salida .

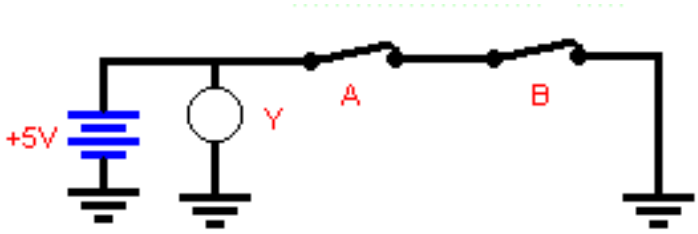


Figura 14: Circuito eléctrico equivalente de una compuerta NAND

Debido a que los interruptores A y B están en serie entre si y en paralelo con la lampara (Y) , esta ultima solo se apaga cuando ambos interruptores están cerrados y permanece encendida mientras cualquiera de ellos este abierto.

LA PUERTA NOR

Considerar el diagrama lógico de la fig. 15 . se ha conectado un inversor a la salida de una puerta OR . la expresión booleana en la entrada de un inversor es A + B . el inversor complementa la salida de la puerta OR , lo que se indica colocando una barra encima de la expresión booleana . obteniéndose $\overline{A+B} = Y$. Esta es una función NOT-OR. La función NOT-OR puede representarse por un símbolo lógico llamado puerta NOR que se ilustra en el diagrama de la fig. 16. Observar que se ha añadido un pequeño circulo inversor al símbolo OR para formar el símbolo NOR .

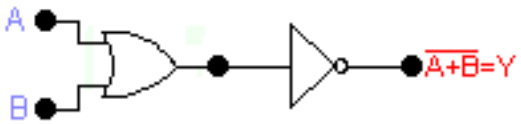


Figura 15: Circuito equivalente de una compuerta NOR

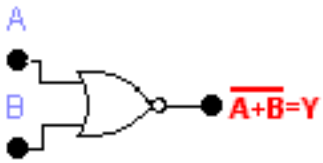


Figura 16: Símbolo lógico de una compuerta NOR

Podemos decir que este dispositivo lógico opera en forma exactamente opuesta a una puerta OR , entregando una salida alta cuando todas sus entradas son bajas y una salida baja cuando existe por lo menos un alto en cualquiera de ellas .

La operación de una puerta NOR es análoga a la del circuito eléctrico mostrado en la fig. 17 los interruptores A y B representan las entradas de la puerta y la lampara (Y) su salida.

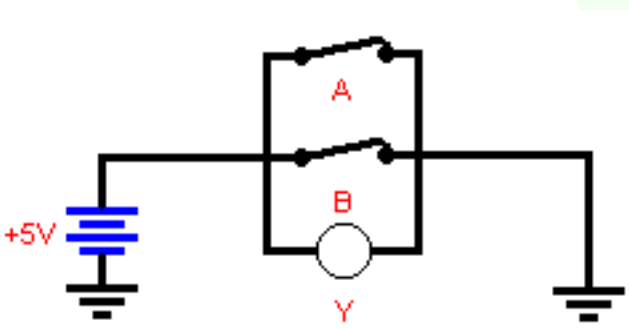


Figura 17: Circuito eléctrico equivalente a una compuerta NOR

Debido a que los interruptores A y B están en paralelo entre si y con la lampara (Y) esta ultima solo enciende cuando ambos interruptores están abiertos y permanece apagada mientras cualquiera de ellos , o ambos , estén cerrados.

La tabla de verdad 9 detalla la operación de la puerta NOR. Es complemento (ha sido invertida) de la columna OR en otras palabras , la puerta NOR pone un 0 donde la puerta OR produce un 1

A	B	NOR	OR
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

Tabla 9: Tabla de verdad de una compuerta NOR de dos entradas

LA PUERTA OR EXCLUSIVA O XOR

La OR - exclusiva se denomina la puerta de << algunos pero no todos >>. El termino OR - exclusiva con frecuencia se sustituye por XOR. La tabla de verdad para la función XOR se muestra en la tabla 10 . un cuidadoso examen muestra que esta tabla de verdad es similar a la tabla de verdad OR, excepto que cuando ambas entradas son 1 la puerta XOR genera un 0.

A	B	OR	XOR
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0

Tabla 10: Tabla de verdad de una compuerta XOR de dos entradas

La operación de una puerta XOR es análoga a la del circuito eléctrico mostrado en la fig. 18. los interruptores A y B simulan las entradas y la lampara (Y) la salida .

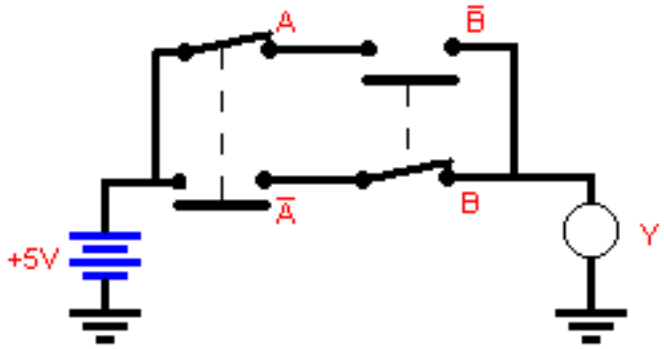


Figura 18: Circuito eléctrico equivalente de una compuerta XOR

Los interruptores A y B están acoplados mecánicamente a los interruptores A y B de modo que cuando A se cierra entonces A se abre y viceversa . lo mismo puede decirse del interruptor B con respecto al B.

Cuando los interruptores A y B están ambos cerrados o ambos abiertos la lampara no enciende. En cambio , cuando uno de ellos , por ejemplo el A , esta abierto y el otro, B, esta cerrado , entonces la lampara se enciende.

Una booleana para la puerta XOR puede obtenerse de la tabla de verdad la fig. 19 la expresión es A ·B + Å · B = Y a partir de esta expresión booleana puede construirse un circuito lógico utilizando puertas AND, puertas OR e inversores dicho circuito aparece en la fig. 19 a este circuito lógico realiza la función lógica XOR.

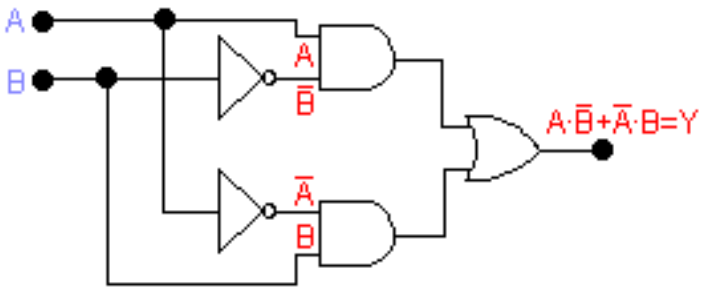


Figura 19: Circuito lógico que realiza la función XOR

El símbolo lógico convencional para la puerta XOR se muestra en la fig. 20 la expresión booleana A ⊕ B, es una expresión XOR simplificada . el símbolo ⊕ significa la función XOR en álgebra booleana. Se dice que las entradas A y B de la fig. 20 realiza la función OR - exclusiva.

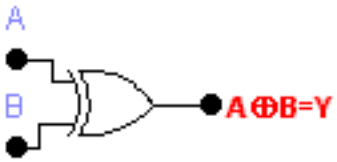


Figura 20: Símbolo lógico de una compuerta XOR

LA PUERTA NOR EXCLUSIVA O XNOR

Una compuerta NOR - exclusiva o XNOR opera en forma exactamente opuesta a una compuerta XOR, entregando una salida baja cuando una de sus entradas es baja y la otra es alta y una salida alta cuando sus entradas son ambas altas o ambas bajas.

Es decir que una compuerta XNOR indica, mediante un lógico que su salida, cuando las dos entradas tienen el mismo estado.

Esta característica la hace ideal para su utilización como verificador de igual en comparadores y otros circuitos aritméticos ..

En la figura 21 se muestra el símbolo lógico, y en la tabla 11 el funcionamiento de una compuerta XNOR. La expresión Y = A ⊕ B pude leerse como Y = A o B exclusivamente negada .

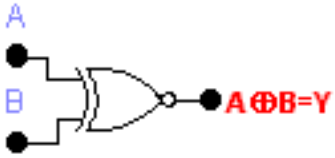


Figura 21: Símbolo lógico de una compuerta XNOR

A	B	Y

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

Tabla 11: Tabla de verdad de una compuerta XNOR de dos entradas

Para efectos prácticos una compuerta XNOR es igual una compuerta XOR seguida de un inversor. En la fig. 22 se indica esta equivalencia y se muestra un circuito lógico de compuertas AND , OR y NOT que opera exactamente como una compuerta X NOR.

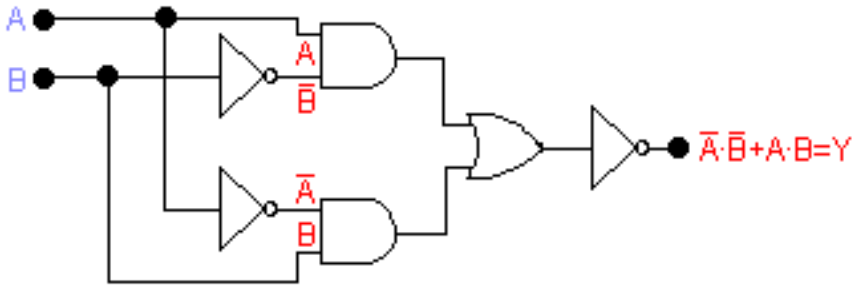


Figura 22: Circuito lógico que realiza la función XNOR

La operación de una compuerta XNOR es análoga a la del circuito eléctrico mostrado en la figura 23 los interruptores A y B están acoplados de la misma forma que el circuito XOR. Cuando los interruptores A y B están ambos cerrados o ambos abiertos , la lampara se enciende . en cambio cuando uno de ellos por ejemplo el A esta abierto y el B esta cerrado , entonces la lámpara no se enciende.

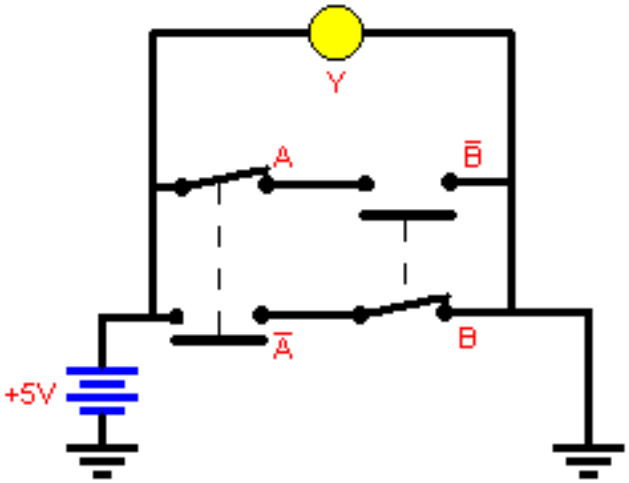


Figura 23: Circuito eléctrico equivalente de una compuerta XNOR

REGISTROS DE DESPLAZAMIENTO

Es un circuito digital que acepta datos binarios de una fuente de entrada y luego los desplaza, un bit a la vez, a través de una cadena de flip-flops.

Este sistema secuencial es muy utilizado en los sistemas digitales. Un ejemplo de esto se ve en las calculadoras comunes, donde al escribir una cifra de varios números, se nota que el primer número pulsado le cede espacio a los demás corriéndose a la izquierda, donde además se nota que hay características de memoria porque se mantienen visualizados los números pulsados.

Los registros de desplazamiento son construidos a partir de flip-flops. Además de tener características de memoria y la función de desplazar datos, también se utilizan para convertir datos serie a paralelo y paralelo a serie.

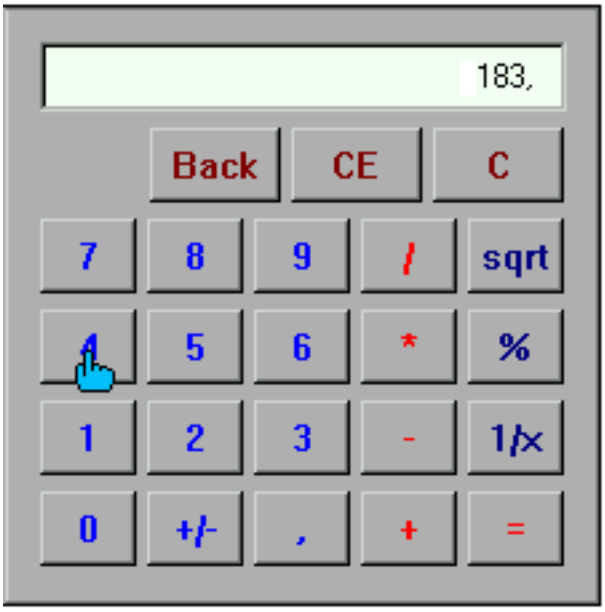
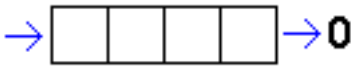


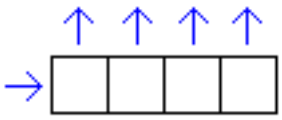
Figura 1 Ejemplo de registro de desplazamiento

Un método de identificar los registros de desplazamiento es por la forma en que se introducen y leen los datos en la unidad de almacenamiento.

Existen cuatro categorias de registro de desplazamiento.

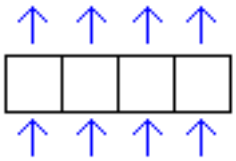


Entrada serie/Salida serie



Entrada serie/Salida paralelo

Entrada paralelo/Salida serie



Entrada paralelo/Salida paralelo
Figura 2 Tipos de registros de desplazamiento

REGISTRO DE DESPLAZAMIENTO DE CARGA SERIE.

Estos registros se denominan de carga serie porque los datos entran secuencialmente a traves del primer flip-flop.

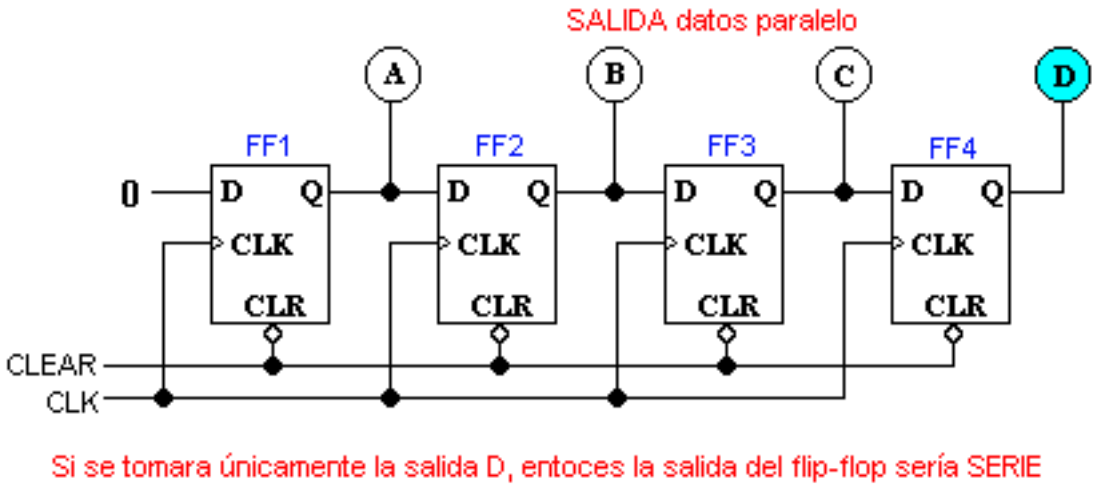


Figura 3: Registro de desplazamiento carga serie

La figura 3 ilustra un sencillo registro de desplazamiento serie de 4 bits que tiene las siguientes características:

1. Esta construido a partir de flip-flops tipo D
2. La entrada de datos se hace en FF1 (entrada serie).
3. La entrada de borrado (CLR) esta conectada en paralelo a todos los flip-flops y se activa en el nivel BAJO.
4. La entrada de reloj (CLK) esta conectada en paralelo a los flip-flops que se accionan con el flanco positivo del tren de pulso.
5. Cada salida de los flip-flops tiene indicadores de salida (salida paralelo).

Funcionamiento.

El diagrama de tiempo ilustra claramente su funcionamiento.

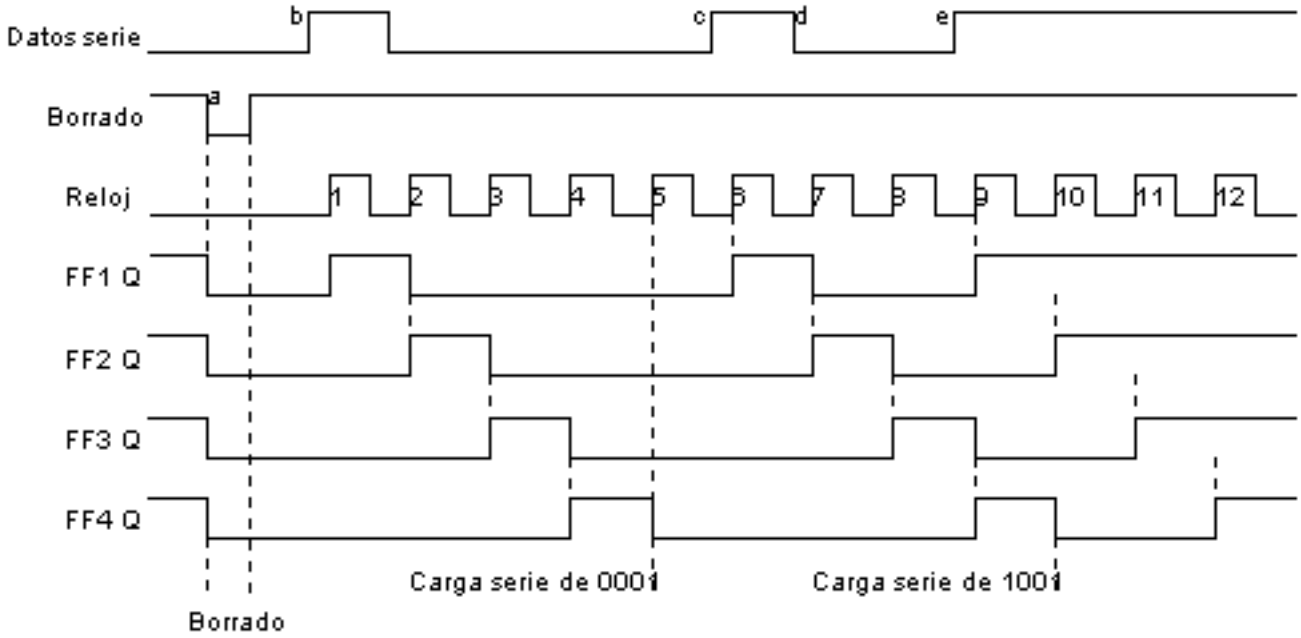


Figura 4: Diagrama de tiempo para un registro de desplazamiento a la derecha, de carga serie de 4 bits.

1. Primero colocamos la entrada de borrado a 1 y colocamos un 1 en la entrada de datos.
2. Con el pulso de reloj 1 (flanco positivo) la entrada de 1 del FF1 se transfiere a la salida de este FF. En la salida se lee entonces 1000 (A = 1, B = 0, C = 0, D = 0).
3. Colocamos un cero en la entrada de datos de FF1 y pulsamos la entrada de reloj una segunda vez. La salida será 0100 (A = 0, B = 1, C = 0, D = 0).
4. Seguimos haciendo el mismo procedimiento y comprobamos que en el pulso de reloj 5 el dato se pierde fuera del registro.
5. De los pulsos del 6 al 9 repetimos el mismo procedimiento, pero antes del pulso 9 introducimos un 1 en la entrada de datos y con el pulso de reloj 9 se visualizará 1001.
6. En los pulsos de reloj del 10 al 12 mantenemos la entrada de FF1 activada y comprobamos que en el pulso de reloj 12 la salida será 1111.

REGISTRO DE DESPLAZAMIENTO DE CARGA PARALELO.

Estos registros se denominan de carga paralelo porque cada flip-flop tiene una entrada preset (PS) en paralelo que es por donde se introducen los datos. También tienen las entradas de reloj borrado y las salidas Q y 1.

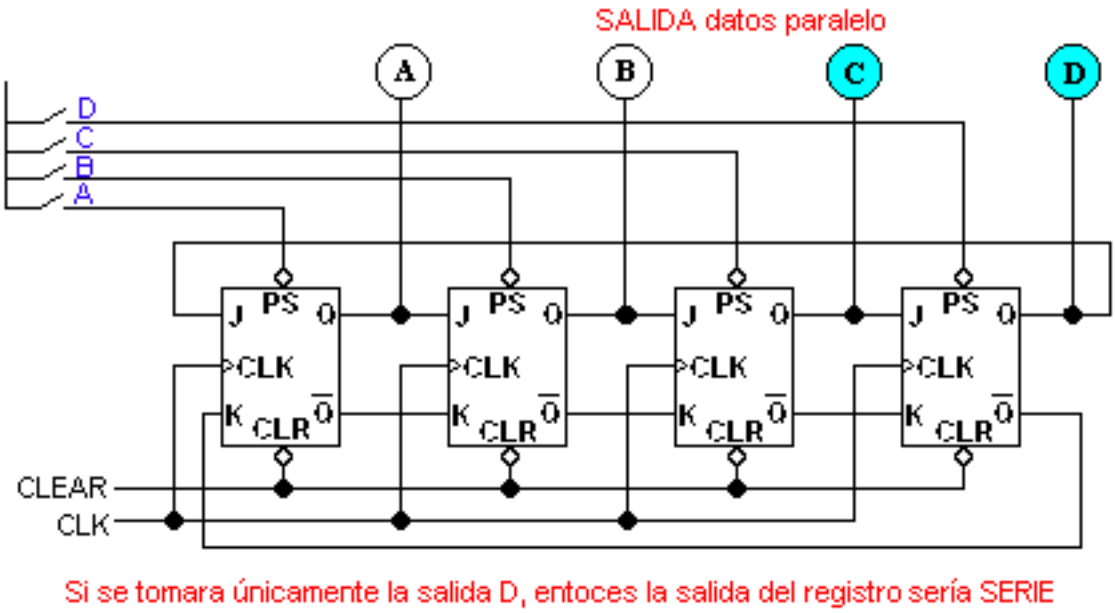


Figura 5: registro de desplazamiento carga paralelo

La figura 5 ilustra un diagrama lógico de un registro de desplazamiento a la derecha, recirculante de carga paralelo de 4 bits.

Este diagrama se caracteriza porque tiene una realimentación que va de la salida Q de FF4 a la entrada J de FF1 y de 1 de FF4 a la entrada K de FF1 para evitar que los datos se pierdan por el extremo derecho de este registro.

El diagrama de tiempo muestra su funcionamiento:

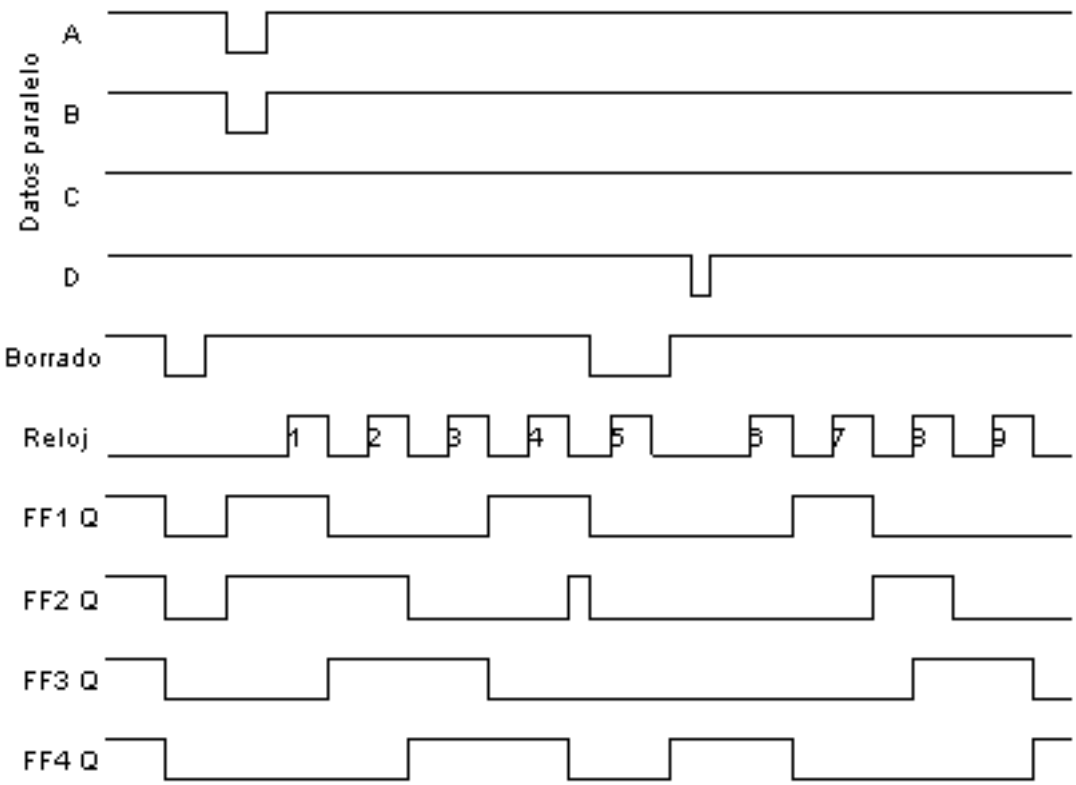


Figura 6: Diagrama de tiempos de un registro de desplazamiento carga paralelo

1. Primero accionamos la entrada de borrado para poner las salidas a 0000 (punto a).
2. Se activan las entradas A y B de datos en paralelo. Como son entradas asíncronas van inmediatamente al nivel ALTO. En el punto C desactivamos estas entradas. El registro de salida será 1100.
3. En el flanco posterior del pulso de reloj 1 los datos se desplazan a la derecha dando como resultado (0110).
4. En el pulso de reloj 3 la salida pasa de (0011) a (1001) debido a la realimentación que existe de Q de FF4 a J de FF1.
5. En el pulso de reloj 4 el registro de salida será igual al del comienzo (1100). Entonces se necesitara de 4 pulsos de reloj para que recircule el a su posición original.
6. En el punto e se acciona la entrada D que genera una salida en FF4 (0001). Después del pulso 6 el dato recircula a FF1.
7. Después de 4 pulsos (6 a 9), el dato es el mismo que el original (0001).

REGISTRO DE DESPLAZAMIENTO TTL

Los fabricantes de circuitos integrados ofrecen muchos registros de desplazamiento. El que estudiaremos a continuación es un registro de desplazamiento universal. El símbolo lógico de bloques para el registro de desplazamiento /universal de 4 bits, TTL 74194 se muestra en la figura 5. Este registro tiene 10 entradas y 4 salidas; estas ultimas están conectadas a la salidas normales (Q) de cada flip flop en el circuito integrado.

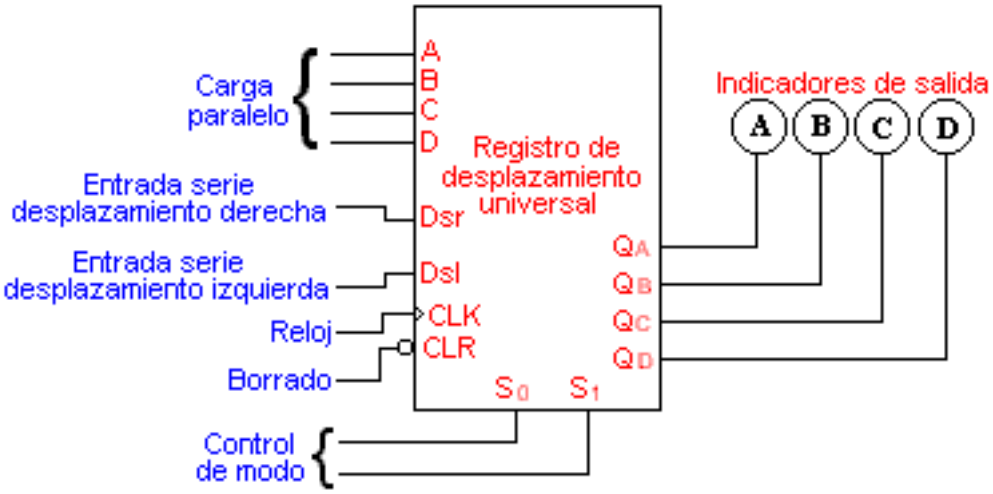


Figura 7: Registro de desplazamiento TTL 74194

Observar que las 4 entradas del registro 74194 (A, B, C, D) son las entradas de carga en paralelo las 2 entradas siguientes introducen los datos en el registro en forma serie (o sea, cada vez un bit), estas son: entrada serie de desplazamiento a la derecha (DCR). esta introduce los bits por la posición A (QA) (es decir, el visualizador A) de esta forma el registro se ha desplazado hacia la derecha. La entrada serie de desplazamiento a la izquierda (DCL) introduce los bits por la posición D (QD) (es decir visualizador D) y así el registro se desplaza hacia la izquierda Las entradas del reloj (CLK) dispara los 4 flip-flops durante las transición L a H (bajo a alto) del pulso de reloj. Cuando la entrada de borrado (CLR) la activamos con un nivel BAJO automáticamente ponemos todos los flip-flops a cero. Los controles de modo a través de una red de puertas le indican al registro que desplace a la izquierda, a la derecha, que cargue en paralelo, o no haga nada (mantenimiento). Como todos los CI's TTL el 74194 tiene sus conexiones de alimentación +5V y GND, pero habitualmente esta no se indican en el símbolo lógico. Los modos de operación del registro de desplazamiento son: reset, mantenimiento, desplazamiento a la izquierda, desplazamiento a la derecha, y carga en paralelo. En los registros de desplazamiento la forma de identificar las entradas y las salidas varían de un fabricante a otro.

REGISTRO DE DESPLAZAMIENTO CMOS

Los fabricantes de circuitos integrados disponen de gran variedad de registros de desplazamiento CMOS. "El que estudiaremos a continuación es el CI 74HC164 es un registro de desplazamiento de 8 bits entrada serie salida paralelo. El diagrama en bloques para el registro de desplazamiento CMOS 74HC164 se muestra en la figura xx, este CI viene encapsulado en forma de DIP de 14 patillas, opera con una fuente de alimentación de

+5V DC y opera disparado con flanco y solo permite la entrada de datos serie.

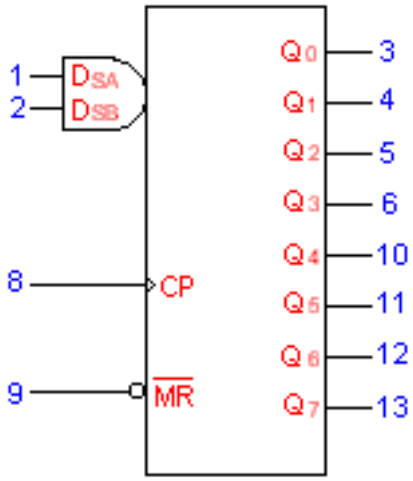


Figura 8: Registro de desplazamiento CMOS 74HC164

El CI CMOS 74HC164 es un registro de desplazamiento disparado por flanco, que solo permite la entrada de datos en serie. Que proceden de cada uno de los 8 flip-flops internos y, por cada flip-flop hay disponible una salida (Q0 a Q7). Los datos se introducen bit a bit (serie) a través de cada una de las 2 entradas de datos (Dsa y Dsb) Estas 2 entradas pueden realizar la operación AND. Esto significa que una entrada puede utilizarse como entrada de habilitación de datos activa en un nivel ALTO, mientras que el dato serie se introduce por la segunda entrada de dato. Si no se necesita la habilitación de entrada de datos, ambas entradas de datos (DSa y DSb) se unen y se utilizan como única entrada de datos serie. La entrada del reloj (CP) desplaza una posición a la derecha desde (Q0 a Q7) en la transición de nivel L a H (BAJO a ALTO). La entrada de reset maestro (MR) en el 74CH164 es una entrada activa en nivel BAJO que reinicializa los 8 flip-flops y pone las salidas a cero, esta es una entrada asíncrona, que elimina las demás entradas. Los fabricantes producen diversos registros de desplazamiento CMOS. Si se conectan registros de desplazamiento que contengan flip-flop D, se pueden utilizar los CI 4076 y 40174. El CI 4014 es un registro de desplazamiento estático de 8 etapas es un dispositivo de entrada serie salida paralelo. El 4031 es un registro de desplazamiento estático de 64 etapas. El registro de desplazamiento de 4 bits 4035 es una unidad de desplazamiento entrada serie salida paralelo. El registro de desplazamiento estático de 8 bits 4034 es una unidad universal de entrada/salida serie/paralelo bidireccional de 3 estados, con la que se puede entrar y salir a las lineas del bus. También hay disponibles otros muchos registros de desplazamiento en las series 74H y 74HCT de CI CMOS.

SIMPLIFICACION DE CIRCUITOS LOGICOS :

Una vez que se obtiene la expresión booleana para un circuito lógico, podemos reducirla a una forma más simple que contenga menos términos, la nueva expresión puede utilizarse para implantar un circuito que sea equivalente al original pero que contenga menos compuertas y conexiones.

SIMPLIFICACION ALGEBRAICA.

El álgebra booleana (Algebra de los circuitos lógicos tiene muchas leyes o teoremas muy útiles tales como :

1. Ley de Morgan :

    1. 1. $\overline{A} + \overline{B} = \overline{A \cdot B}$
       2. $\overline{A \cdot B} = \overline{A} + \overline{B}$

2. Ley Distributiva :

    1. 3. $A+(B \cdot C) = (A+B) \cdot (A+C)$
       4. $A \cdot (B+C) = A \cdot B + A \cdot C$

Ademas de las leyes formales para las funciones AND y OR :

1. 5. $A \cdot 0 = 0$; $A+0 = A$
   6. $A \cdot 1 = A$; $A+1 = 1$
   7. $A \cdot A = A$; $A+A = A$
   8. $A \cdot \overline{A} = 0$; $A+\overline{A} = 1$

y la Ley de la Involución:

1. 9. $\overline{A}$(negada) = A

Considerar la expresión booleana $A \cdot \overline{B} + \overline{A} \cdot B + A \cdot B = Y$, un diagrama lógico de ésta

expresión aparece en la Figura 1. Observar que deben utilizarse seis puertas para implementar este circuito lógico, que realiza la lógica detallada en la tabla de verdad (Tabla 1)

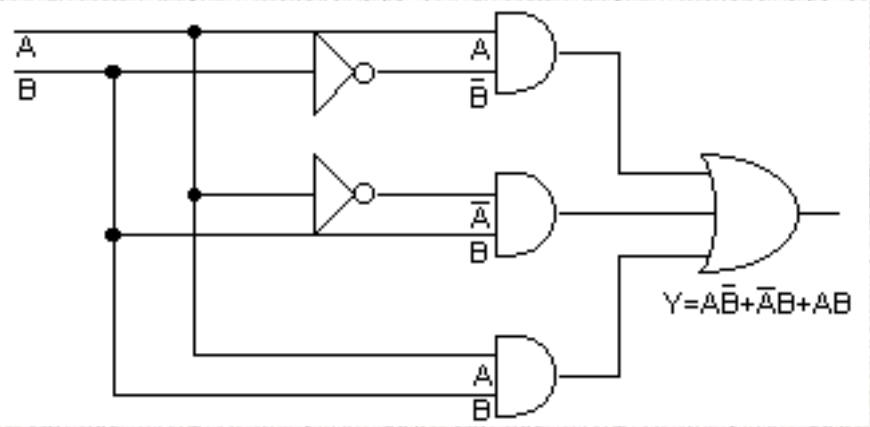


Figura 1: Circuito lógico no simplificado

ENTRADAS		SALIDA
B	**A**	**Y**
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 1: Tabla de verdad de la función OR

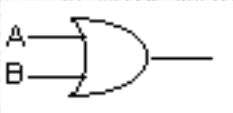


Figura 2: Circuito lógico simplificado

Aplicando el álgebra booleana :

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

RAZONES

$= A \cdot \overline{B} + (\overline{A} \cdot B + A \cdot B)$	,	Propiedad asociativa
$= A \cdot \overline{B} + B \cdot (\overline{A} + A)$	,	4. [A·(B + C) = A·B + A·C]
$= A \cdot \overline{B} + B \cdot 1$	,	8. [A + $\overline{A}$ = 1]
$= A \cdot B + B$	,	6. [B·1 = B]
$= B + A \cdot \overline{B}$	,	Propiedad conmutativa
$= (B + A) \cdot (B + \overline{B})$	,	3. [A + (B·C) = (A + B)·(A + C)]
$= (B + A) \cdot 1$	,	8. [A + $\overline{A}$ = 1]
$= B + A$	,	6. [A * 1 = A]

Concluimos entonces que una sola puerta OR de dos entradas realiza la misma función (¡ De hecho la tabla 1 corresponde a la función OR !)

EXPRESIONES BOOLENAS EN FORMA DE MINTERMS (SUMA DE PRODUCTOS).

Cuando se comienza un problema de diseño lógico, lo normal es construir primero una tabla de verdad, que detalle la operación exacta del circuito digital. Considerar la tabla de verdad 2, que contiene las variables C, B y y A. Observar que sólo dos combinaciones de variables generan una salida 1. Estas combinaciones se muestran en la lineas octava y segunda (sombreadas) de la tabla de verdad. La linea 2 se lee « una entrada no C Y (AND) una entrada no B Y (AND) una entrada A generan una salida I ». Esto se muestra en la parte derecha de la linea 2 con la expresión booleana $\overline{C} \cdot \overline{B} \cdot A$. La otra combinación de variables que genera un 1 se muestra en la linea 8 de la tabla de verdad. La linea 8 se lee «una entrada C Y (AND) una entrada B Y (AND) una entrada A generan una salida 1». La expresión booleana de la linea 8 aparece a la derecha y es $C \cdot B \cdot A$. Estas dos posible combinaciones se relacionan mediante el operador OR para formar la expresión booleana

completa de la tabla de verdad, que se muestra en la tabla 2, como $C \cdot B \cdot A + \underline{C} \cdot \underline{B} \cdot A = Y$. Esta expresión, a veces, se denomina forma en *suma de productos* de la expresión booleana. Los ingenieros también llaman a esta forma, *forma de minterms*.

Esta expresión puede traducirse al patrón AND-OR de puertas lógicas. El diagrama lógico de la Figura 5.3.c realiza la lógica descrita por la expresión booleana C · B · A + C ·B· A = Y , y genera la tabla de verdad 2.

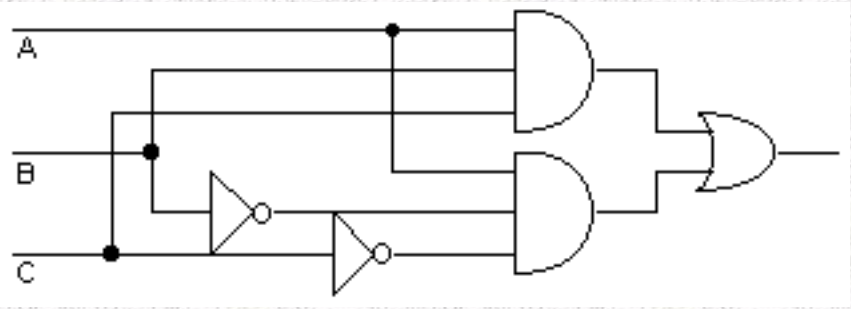


Figura 3: Circuito lógico equivalente AND-OR

ENTRADAS			**SALIDAS**	
C	**B**	**A**	**Y**	
0	0	0	0	
0	0	1	1	C·B·A
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	1	C·B·A
C·B·A + C·B·A = Y				

Tabla 2: Expresión booleana

El procedimiento típico que se sigue en el trabajo de diseño lógico consiste en construir *primero* una tabla de verdad. *A continuación*, determinar una expresión booleana en forma de minterms a partir de la tabla de verdad. *Finalmente*, dibujar el circuito lógico AND-OR a partir de la expresión booleana en minterms.

EXPRESIONES BOOLENAS EN FORMA DE MAXTERMS (PRODUCTO DE SUMAS).

Considerar la tabla de verdad 3. La expresión booleana para esta tabla de verdad puede escribirse de dos formas, cómo se observó en la sección introductoria. La expresión booleana en minterms se obtiene de las salidas que son 1 en la tabla de verdad. Cada 1 en la columna de salida se convierte en un termino, que se relaciona con los demás, mediante el operador OR, en la expresión en forma de minterms. La expresión en minterms para esta tabla de verdad se da en la tabla 3, como :

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

(a) Expresión booleana en forma de maxterms : B + A = Y

TABLA DE VERDAD OR		
ENTRADAS		**SALIDA**
B	**A**	**Y**
0	0	0
0	1	1 -> $B \cdot \overline{A}$
1	0	1 -> $\overline{B} \cdot A$
1	1	1 -> $B \cdot A$
Expresión: $B \cdot A + B \cdot \overline{A} + \overline{B} \cdot A = Y$		

Tabla 3: Expresión booleana en forma de maxterms

La tabla de verdad 3 también puede describirse utilizando una expresión booleana en *forma de maxterms*. Este tipo de expresión se desarrolla a partir de los 0 de la columna de salida de la tabla de verdad. Por cada 0 de la columna de salida se realiza una operación OR. Observar que las *variables de entrada se invierten y después se realiza la operación OR*. La expresión booleana en maxterms de esta tabla de verdad aparece en la tabla 3. La expresión en maxterms para la tabla de verdad OR es *B + A = Y*. Esto significa lo mismo que la familiar expresión OR: *A + B = Y*. Para la tabla de verdad 3, la expresión booleana en

maxterms es la más simple, aunque ambas formas describen con precisión la lógica de dicha tabla de verdad.

ENTRADAS			SALIDA	
C	B	A	Y	
0	0	0	1	
0	0	1	1	
1	1	0	1	
0	1	1	1	
1	0	0	0	$\underline{C}+B+A$
1	0	1	1	
1	1	0	1	
1	1	1	0	$\underline{C}+\underline{B}+\underline{A}$
$(\underline{C}+B+A)\cdot(\underline{C}+\underline{B}+\underline{A})= Y$				

Tabla 4: Expresión booleana en Maxterms.

Considerar la tabla de verdad 4. La expresión en minterms para esta tabla es demasiado larga. La expresión booleana en *maxterms* se obtiene a partir de las variables de las lineas 5 y 8. Cada una de estas lineas tiene un 0 en la columna de salida. Las variables se invierten y se relacionan con operadores OR. Los términos así obtenidos se ponen entre paréntesis y se relacionan con operadores AND. La expresión booleana completa, en forma de maxterms, se da en la tabla 4, y también se la denomina *forma de producto de sumas* de la expresión booleana. El termino producto de sumas viene de la organización de los símbolos de suma (+) y producto (·).

Una expresión booleana en maxterms se implementa utilizando el patrón OR-AND de puertas lógicas según indica la figura 4. Observar que las salidas de las dos puertas OR están alimentando una puerta AND. La expresión en maxterms *($\underline{C}$ + $\underline{B}$ + $\underline{A}$) * ($\underline{C}$ + B + A) = Y*

, se implementa utilizando el patrón OR-AND de puertas lógicas de la Figura 4.

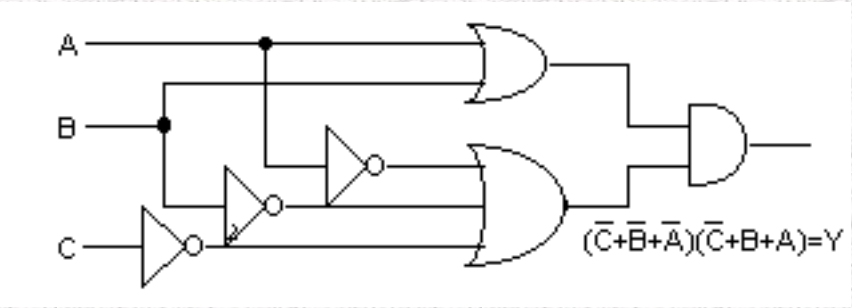


Figura 4:Expresión en forma de maxterms

Aplicando el álgebra booleana podemos pasar expresiones en forma de minterms a maxterms y viceversa. Ejemplo: Pasar la expresión booleana en forma de maxterms,

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

a su correspondiente en forma de mimterms, $Y = \underline{C}\cdot\underline{B}\cdot A + C\cdot B\cdot A$

tenemos :

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

$$= [(C + B + A)\cdot(C + \underline{B} + A)]\cdot[(C + \underline{B} + \underline{A})\cdot(\underline{C} + B + \underline{A})]\cdot[(\underline{C} + B + A)\cdot(\underline{C} + \underline{B} + A)],\ \text{Propiedad asociativa y conmutativa}$$

$$= \{[(C + A) + B]\cdot[(C + A) + \underline{B}]\}\cdot\{[(C + \underline{B}) + \underline{A}]\cdot[(\underline{C} + B) + \underline{A}]\}\cdot\{[(\underline{C} + A) + B]\cdot[(\underline{C} + A) + \underline{B}]\},\ \text{Propiedad asociativa y conmutativa.}$$

$$= [(C + A) + B\cdot\underline{B}]\cdot[(C + \underline{B})\cdot(\underline{C} + B) + \underline{A}]\cdot[(\underline{C} + A) + B\cdot\underline{B}] - - - - ,\ [A + (B\cdot C) = (A + B)\cdot(A + C)]$$

$$(C + A)\cdot[(C + \underline{B})\cdot(\underline{C} + B) + \underline{A}]\cdot(\underline{C} + A) - - - ,\quad [A\cdot\underline{A} = 0]\ \text{y}\ [A + 0 =$$

A]

(C + A)·(C̲ + A)·[(C + B̲)·(C̲ + B) + A̲] - - - , Propiedad conmutativa

(C·C̲ + A)·[(C + B̲)·C̲ + (C + B̲)·B + A̲] , [A + (B·C) = (A + B)(A + C)], [A·(B+C) = A·B + A·C]

A·[C̲·C + C̲·B̲ + C·B + B·B̲ + A̲], [A·A̲ = 0],[A·(B + C) = A·B + A·C] y [A + 0 = A]

A·[C̲·B̲ + C·B + A̲] - - - - - - - - - - - ,[A·A̲ = 0] y [A + 0 = A]

A·C̲·B̲ + A·C·B + A·A̲] - - - - - , [A·(B + C) = A·B + A·C]

A·C̲·B̲ + A·C·B - - - - - - - - - - , [A·A̲ = 0] y [A + 0 = A]

C̲·B̲·A + C·B·A - - - - - - - - , Propiedad conmutativa

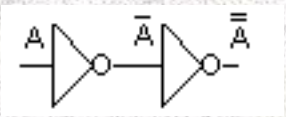
Otra forma de pasar una expresión booleana en forma de minterms a maxterms y viceversa es utilizando únicamente el teorema de D'Morgan. El ejemplo anterior quedaría :

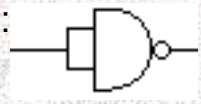
Y = (C + B + A)·(C + B̲ + A)·(C + B̲ + A̲)·(C̲ + B + A)·(C̲ + B + A̲)·(C̲ + B̲ + A)

= (C̲·B̲·A̲)·(C̲·B·A̲)·(C̲·B̲·A)·(C̲·B·A)·(C·B̲·A̲)·(C·B̲·A),

= C̲·B̲·A̲ + C̲·B·A̲ + C̲·B·A + C·B̲·A̲ + C·B̲·A + C·B·A,

UTILIZACION DE LA LOGICA NAND Y NOR.

La lógica NAND y NOR se utiliza para simplificar circuitos compuestos, por puertos AND, OR y NOT, en circuitos compuestos únicamente por puertas NAND o únicamente por puertas NOR. Esta lógica se fundamenta en la ley de la Involución (A̲ = A), la cual puede representarse por : 

, teniendo en cuenta que una puerta NOT es equivalente a : 

la lógica NAND se utiliza para simplificar circuitos AND-OR como se ilustra en el siguiente ejemplo :

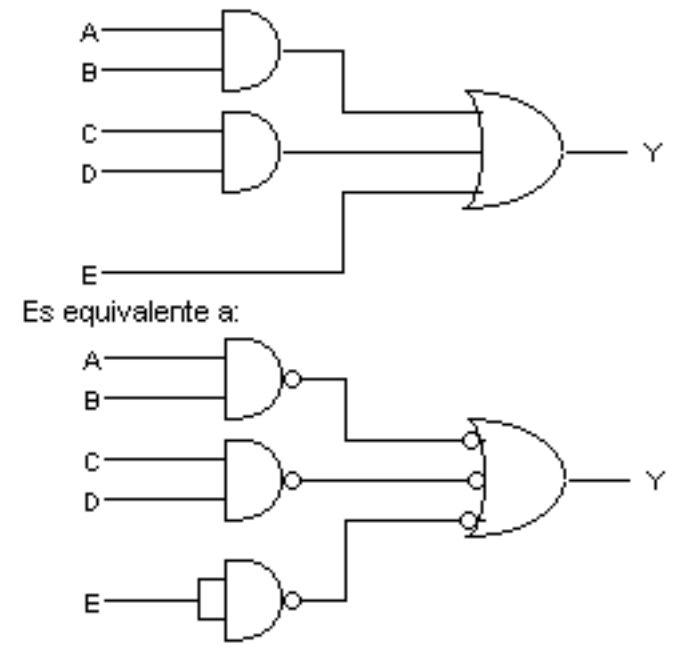


Figura 5: Circuito lógico NAND

Observar que negamos las entradas de la puerta OR, al igual que las salidas de las puertas AND (1 y 2). Dado que la linea E solo se negó una sola vez (A la entrada de la puerta OR), la negamos otra vez con una puerta NOT, para que el circuito no se altere, y teniendo en cuenta la ley de la Involución; es decir E = E.

De manera similar la lógica NOR se utiliza para simplificar circuitos OR-AND como se ilustra en el siguiente ejemplo :

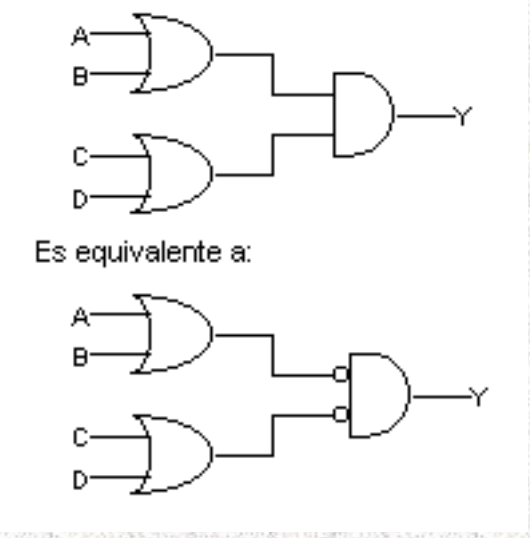
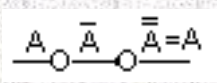


Figura 6: Circuito lógico NOR

Observar que tanto para la utilización de la lógica NAND como para la NOR, sobre cualquier linea se niega dos veces : 

,lo cual es consistente con la ley de la Involución.

DIAGRAMAS DE KARNAUGH

Es un metodo grafico que se utiliza para simplificar circuitos logicos en un proceso simple y ordenado. Es metodo que se basa en los teoremas booleanos estudiados anteriormente y su utilidad practica se limita a 5 variables. Las reglas a seguir son las siguientes:

1. A partir de la tabla de verdad sacar las expresiones booleanas en forma de minterns o maxterms.
2. Colocar los 1 corespondientes en el diagrama por cada grupo de variables operadas por AND si es en forma de minterns u operadas por OR si es en forma de maxterms.
3. Agrupar los 1 adyacentes (las agrupaciones se realizan en grupos de 2, 4, 8 1)
4. Eliminar las variables que aparezcan con su complemento.
5. Enlazamos con OR los resultados obtenidos (si es en forma de minterns) o con AND (si es en forma de maxterms).

Tomemos la tabla de verdad 5. Lo primero que debemos hacer es sacar las expresiones booleanas correspondientes:

A	B	Q	
0	0	0	
0	0	1	$\underline{A}\cdot B$
1	0	1	$A\cdot\underline{B}$
1	1	1	$A\cdot B$
$Q=(\underline{A}\cdot B)+(A\cdot\underline{B})+(A\cdot B)$			

Tabla 5

Luego procedemos a colocar cada 1 correspondiente en el diagrama por cada grupo de variables operadas con AND (para nuestro ejemplo). Los diagramas de Karnaugh pueden presentarse de dos maneras diferentes: la americana y la alemana, demos un vistazo a dichas presentaciones:

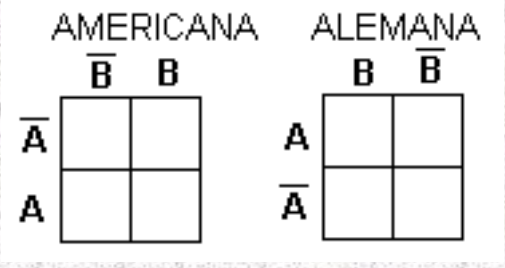


Figura 7: Diagramas de Karnaugh para 2 variables

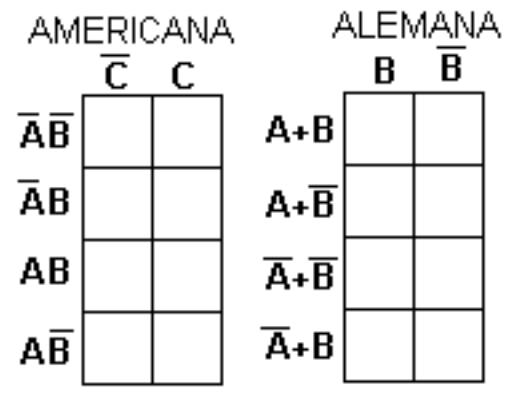


Figura 8: Diagramas de Karnaugh para 3 variables

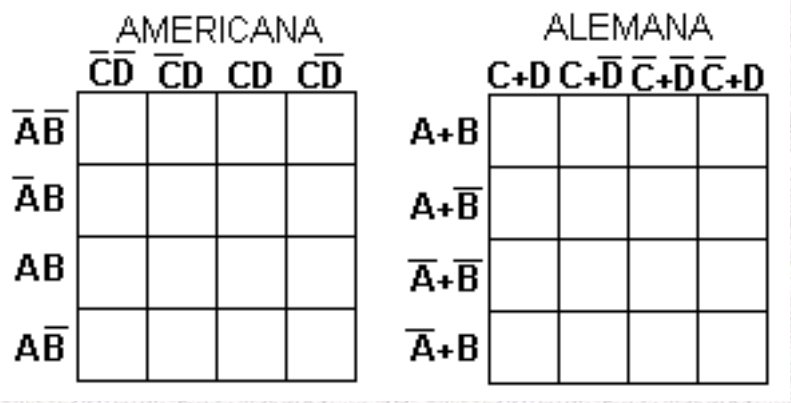


Figura 9: Diagramas de Karnaugh para 4 variables

Ahora que conocemos las maneras en que se pueden presentar las diagramas procedemos a colocar los 1 correspondientes por cada grupo de variables operadas con AND (en nuestro ejemplo)

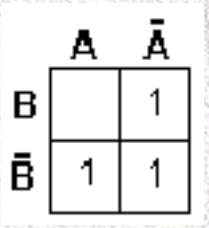


Figura 10: Colocación de los unos en el mapa de Karnaugh

Luego procedemos a agrupar los 1 adyacentes que se encuentren en el diagrama, estas agrupaciones se realizan en grupos de 2, 4, o de 8 "1" . Debemos tratar en lo posible de no realizar tantas agrupaciones.

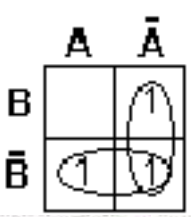


Figura 11: Agrupación de términos

Despues de realizar las agrupaciones eliminanos por cada grupo las variables que aparezcan con su complemento. En el agrupamiento de 2 "1" se elimina una variable; en el agrupamiento de 4 "1" se eliminan 2 variables y en el agrupamiento de 8 "1" se eliminan 3 variables.

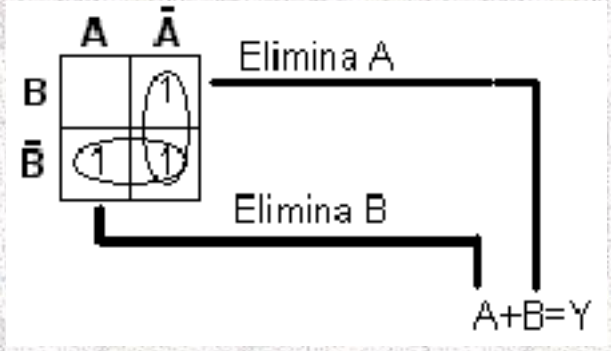


Figura 12: Eliminación de términos

Por ultimo enlazamos con OR (ya que nuestro ejemplo es en forma de minterns) los resultados que obtuvimos de la eliminacion de variables.

$$Q = A + B$$

De esta manera la ecuacion logica $Q=(\underline{A}\cdot B)+(A\cdot B)+(A\cdot \underline{B})$ nos quedaría reducida a una puerta OR

DIAGRAMAS DE KARNAUGH CON 5 VARIABLES

Para realizar simplificaciones con 5 variables se utilizan los llamados diagramas bidimensionales, en donde un plano nos indica la quinta variable y el otro plano su complemento, veamos:

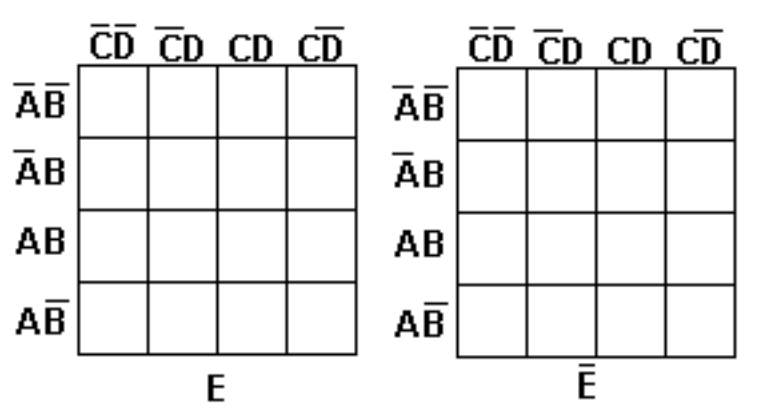


Figura 13: Diagrama de Karnaugh para 5 variables

Realicemos un ejercicio para asimilar la simplificacion con 5 variables. Tomemos la siguiente tabla de verdad:

A	B	C	D	E	Q
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0

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

Tabla 6: Tabla de verdad de cinco variables

Luego procedemos a sacar la ecuacion no simplificada

$$Q = \overline{A}\overline{B}\overline{C}\overline{D}\overline{E} + \overline{A}\overline{B}\overline{C}\overline{D}E + \overline{A}\overline{B}\overline{C}D\overline{E} + \overline{A}\overline{B}\overline{C}DE + \overline{A}\overline{B}C\overline{D}\overline{E} + \overline{A}\overline{B}C\overline{D}E + \overline{A}\overline{B}CD\overline{E}$$

Despues que obtenemos la ecuacion no simplificada pasamos los 1 correspondientes al diagrama y realizamos las agrupaciones. Si existen agrupaciones que ocupan el mismo lugar en ambos planos, los reflejamos para obtener una ecuación más simplificada. El proceso de simplificación es el mismo que utilizamos anteriormente.

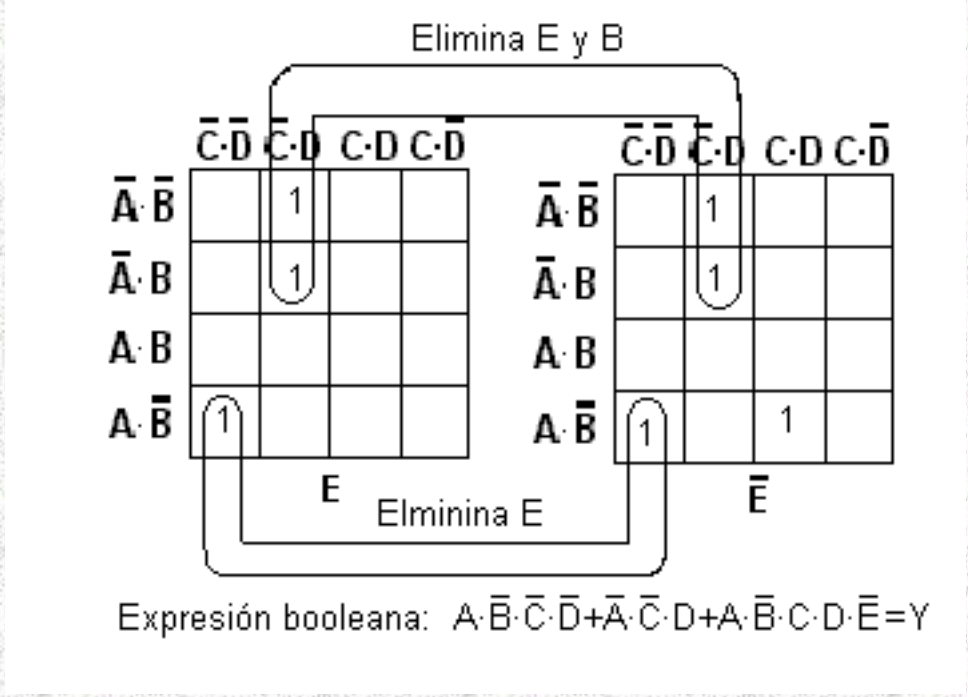


Figura 14: Simplificación de diagramas de Karnaugh de 5 variables

De esta manera obtenemos la siguiente ecuación:

$$Q = ABCD + ACD + ABCDE$$

CONDICIONES NO IMPORTA

En muchos circuitos logicos hay condiciones de entrada para las que no se especifican los niveles de salida,en la mayoria de los casos es por que estas condiciones nunca se presentaran o simplemente el nivel logico de la salida es irrelevante.

A	B	C	Q	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	X	

1	0	0	X	
1	0	1	1	A·B·C
1	1	0	1	A·B·C
1	1	1	1	A·B·C

Tabla 7

En la tabla de verdad no se especifica el nivel de salida para las condiciones "0,1,1" y "1,0,0". En su lugar se coloca una x que representa la condicion no importa. La persona que este realizando la simplificacion tiene la libertad de determinar el nivel logico para la salida de la condicion "no importa", con el fin de producir la expresion mas simple. Realicemos la simplificacion:

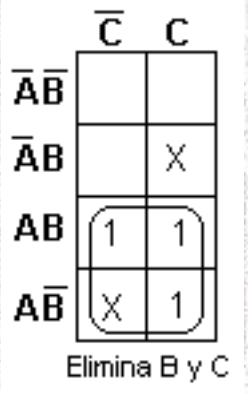


Figura 15: Simplificación de diagramas de Karnaugh con condiciones "no importa"

de esta manera obtenemos que: Q = A.

En muchos casos se trabaja con el código BCD, sabemos que en este codigo existen 6 cobinaciones que son prohibidas (1010,1011,1101, 1110,1111), estas condiciones tambien son llamadas condiciones no importa.

8	4	2	1	Q

0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Tabla 8: Términos irrelevantes en los números BCD

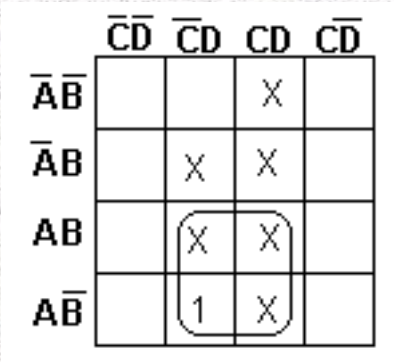


Figura 16: Simplificación

NÚMEROS UTILIZADOS EN ELECTRÓNICA DIGITAL

Los sistemas de numeración utilizados en electrónica digital son los siguientes: sistema decimal, sistema binario, sistema octal y sistema hexadecimal

SISTEMA DECIMAL

Este sistema consta de diez símbolos que van desde el numero 0 hasta el numero 9, los cuales le dan la característica principal a este sistema conocido por todo el mundo. Estos símbolos numéricos también forman unidades numéricas compuestas, al tomarlos como exponentes de un número que se encargará de regular el procedimiento, este número es llamado base. El numero base va a ser 10, por tal motivo también es conocido como "sistema de numeración en base 10".

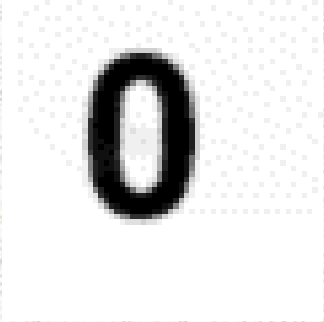


Figura 1: Sistema decimal

SISTEMAS DE NÚMEROS BINARIOS

Figura 2: Sistema de numeros binarios

Este es el sistema numérico que utilizan los sistemas digitales para contar y es el código al que traduce todas las informaciones que recibe. Se dice "Binario" a todo aquello que tiene dos partes, dos aspectos, etc. Muchas cosas en los sistemas digitales son binarias: Los

impulsos eléctricos que circulan en los circuitos son de baja o de alta tensión, los interruptores biestables están encendidos o apagados, abiertos o cerrados, etc. A diferencia del sistema decimal al que estamos habituados, y que utiliza diez cifras, del 0 al 9, el sistema numérico binario utiliza solo dos cifras, el 0 y el 1. En el sistema binario las columnas no representan la unidad, la decena, la centena, como en el sistema decimal, sino la unidad (2^0), el doble (2^1), el doble (2^2), etc. De modo que al sumar en la misma columna 1 y 1, dará como resultado 0, llevándonos 1 a la columna inmediatamente a la izquierda. Para los sistemas digitales es fácil, hasta el punto que reduce todas las operaciones a sumas y restas de números binarios.

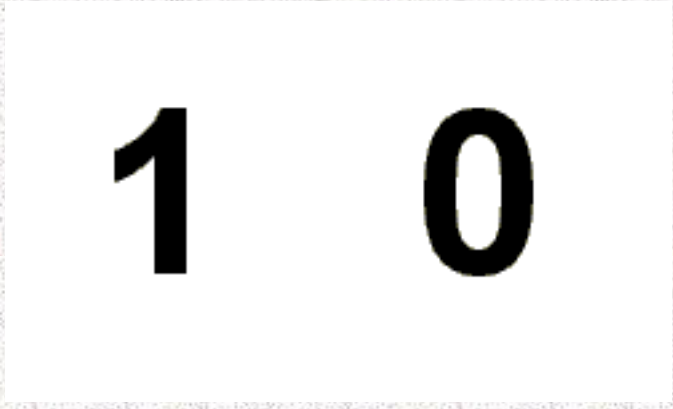


Figura 3: Sistema binario

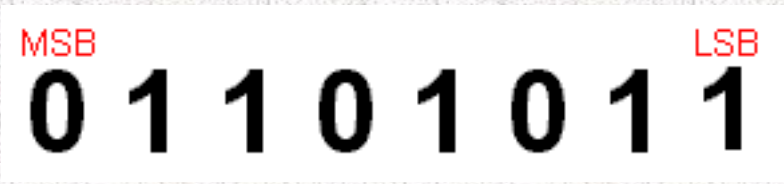


Figura 4: Números binarios

También las palabras, los números y los dibujos se traducen en el ordenador en secuencias de 1 y 0. De hecho toda letra, cifra o símbolo gráfico es codificado en una secuencia de 0 y 1. Si, por ejemplo, nuestro nombre tiene cinco letras, la representación para el ordenador constara de cinco bytes. La palabra bit deriva de las dos palabras inglesas "binary digit" cifra binaria, y designa a las dos cifras 0 y 1, que se utilizan en el sistema binario. Un bit es también, la porción más pequeña de información representable mediante un número, e indica si una cosa es verdadera o falsa, alta o baja, negra o blanca, etc.
Un byte es generalmente una secuencia de 8 bits. Ocho ceros y unos se pueden ordenar de 256 maneras diferentes ya que cada bit tiene un valor de posición diferente, donde el bit

numero 1 le corresponderá un valor de posición de 2^0(1), el siguiente bit tendrá un valor de 2^1(2), el siguiente 2^2(4), el siguiente 2^3(8), el siguiente 2^4(16), el siguiente un valor de 2^5(32), y así sucesivamente hasta llegar la ultima posición, o ultimo bit, en este caso el numero 8, que también es llamado el MSB (Bit Mas Significativo) y el LSB (Bit Menos Significativo) correspondiente a la primera posición o bit numero 1. Ejemplo:

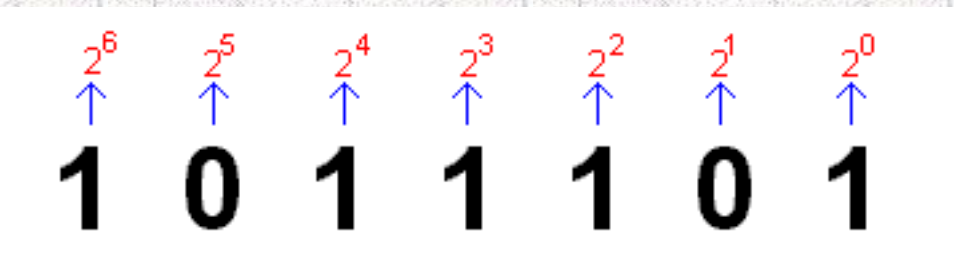


Figura 5: Valores de las posiciones de los números binarios

SISTEMA DE NUMERACIÓN OCTAL

Figura 6: Sistema de numeración octal

Este sistema consta de 8 símbolos desde el 0 hasta el 7, es muy poco utilizado en los computadores. La facilidad con que se pueden convertir entre el sistema Octal y el binario hace que el sistema Octal sea atractivo como un medio "taquigráfico" de expresión de números binarios grandes. Cuando trabajamos con una gran cantidad de números binarios de muchos bits, es mas adecuado y eficaz escribirlos en octal y no en binarios. sin embargo, recordemos los circuitos y sistemas digitales trabajan eléctricamente en binario, usamos el sistema Octal solo por conveniencia con los operadores del sistema

SISTEMA DE NUMERACIÓN HEXADECIMAL

Este sistema consta de 16 símbolos donde desde el 0 hasta el 9 son números y del 10 hasta el 15 son letras, las cuales se encuentran distribuidas en la siguiente forma:

Hexadecimal	Decimal	Hexadecimal	Decimal
0	0	8	8
1	1	9	9
2	2	A	10
3	3	B	11
4	4	C	12
5	5	D	13
6	6	E	14
7	7	F	15

Tabla 1: Símbolos utilizados en el sistema de numeración hexadecimal

La ventaja principal de este sistema de numeración es que se utiliza para convertir directamente números binarios de 4 bits. En donde un solo dígito hexadecimal puede representar 4 números binarios o 4 bits.

CONVERSIONES DE SISTEMAS DE NUMERACIÓN

CONVERSIÓN DE UN NUMERO DECIMAL A BINARIO

Para esta transformación es necesario tener en cuenta los pasos que mostraremos en el siguiente ejemplo: Transformemos el numero 42 a numero binario

1. Dividimos el numero 42 entre 2
2. Dividimos el cociente obtenido por 2 y repetimos el mismo procedimiento hasta que el cociente sea 1.
3. El numero binario lo formamos tomando el primer dígito el ultimo cociente, seguidos por los residuos obtenidos en cada división, seleccionándolos de derecha a izquierda, como se muestra en el siguiente esquema.

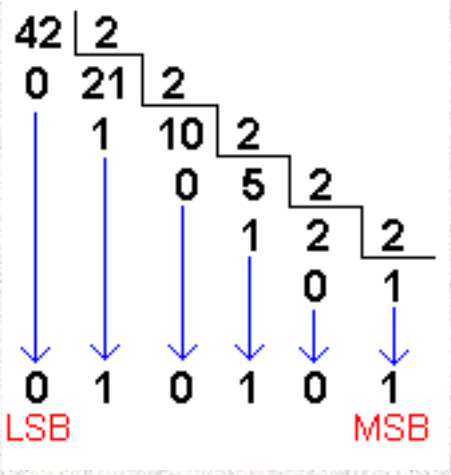


Figura 7: Conversión de decimal a binario

CONVERSIÓN DE UN NUMERO DECIMAL FRACCIONARIO A UN NUMERO BINARIO

Para transformar un número decimal fraccionario a un numero binario debemos seguir los pasos que mostramos en el siguiente ejemplo: transformemos el numero 42,375.

1. la parte entera se transforma de igual forma que el ejemplo anterior.
2. La parte fraccionaria de la siguiente manera:

    - Multiplicamos por el numero 2 y tomamos la parte entera del producto que ira formando el numero binario correspondiente
    - Tomamos nuevamente la parte entera del producto, y la parte fraccionaria la multiplicamos sucesivamente por 2 hasta llegar a 0
    - Tomamos nuevamente la parte entera , y como la parte fraccionaria es 0, indica que se ha terminado el proceso .El numero binario correspondiente a la parte decimal será la unión de todas las partes enteras, tomadas de las multiplicaciones sucesivas realizadas durante el transcurso del proceso , en donde el primer dígito binario corresponde a la primera parte entera , el segundo dígito a la segunda parte entera , y así sucesivamente hasta llegar al ultimo .Luego tomamos el numero binario , correspondiente a la parte entera , y el numero binario , correspondiente a la parte fraccionaria y lo unimos en un solo numero binario correspondiente a el numero decimal.

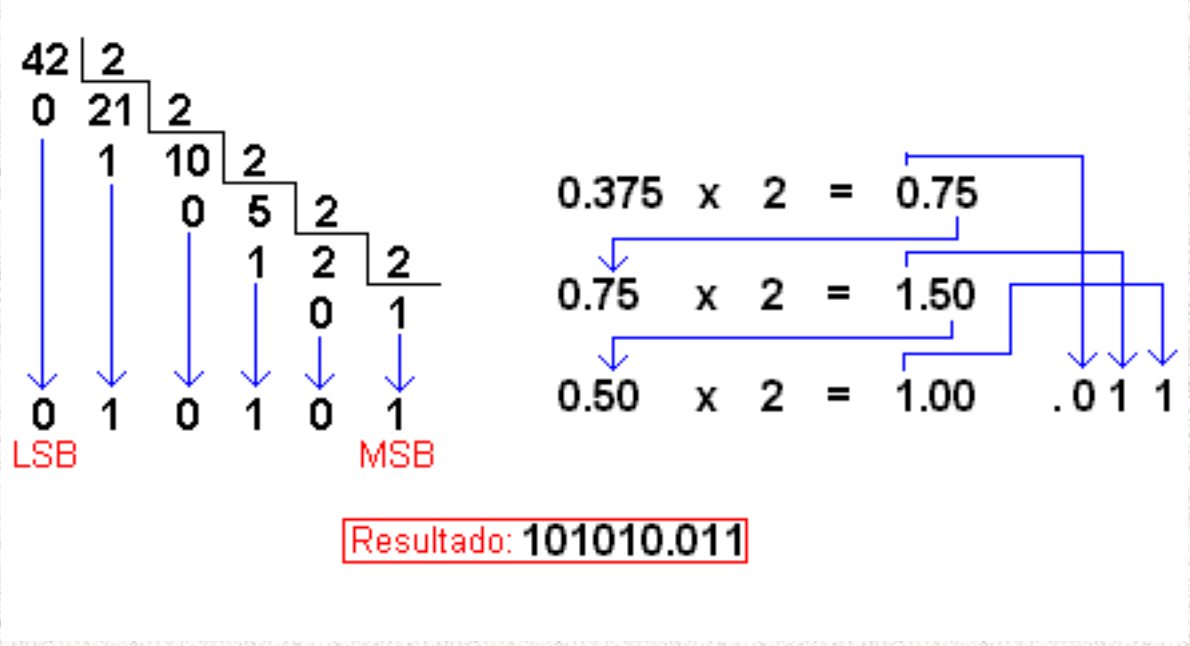


Figura 8: Conversión de decimal fraccionario a binario

CONVERSIÓN DE UN NUMERO BINARIO A UN NUMERO DECIMAL

Para convertir un número binario a decimal, realizamos los siguientes pasos:

1. Tomamos los valores de posición correspondiente a las columnas donde aparezcan únicamente unos
2. Sumamos los valores de posición para identificar el numero decimal equivalente

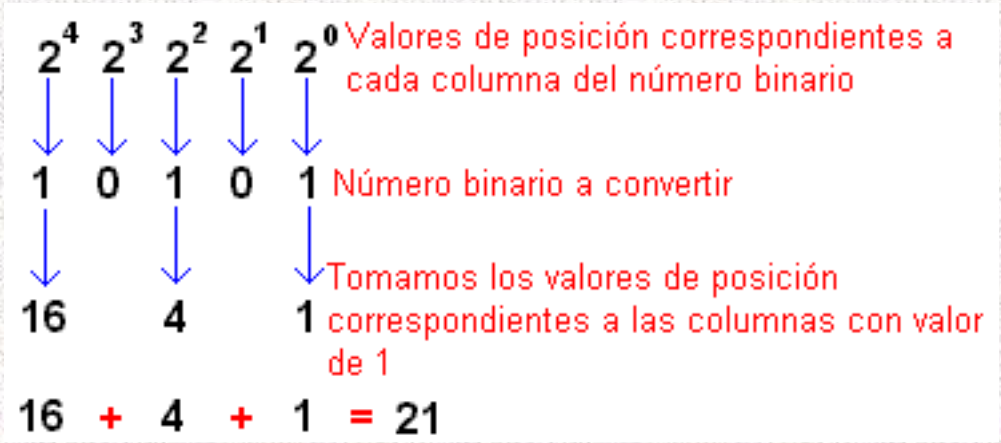


Figura 9: Conversión de binario a decimal

CONVERSIÓN DE UN NUMERO DECIMAL A OCTAL

Para convertir un numero en el sistema decimal al sistema de numeración Octal, debemos seguir los pasos que mostraremos en el siguiente ejemplo Convertir el numero decimal 323.625 a el sistema de numeración Octal

1. Se toma el numero entero y se divide entre 8 repetidamente hasta que el dividendo sea menor que el divisor, para colocar entonces el numero 0 y pasar el dividendo a formar el primer dígito del numero equivalente en decimal
2. Se toma la parte fraccionaria del numero decimal y la multiplicamos por 8 sucesivamente hasta que el producto no tenga números fraccionarios
3. Pasamos la parte entera del producto a formar el dígito correspondiente
4. Al igual que los demás sistemas , el numero equivalente en el sistema decimal , esta formado por la unión del numero entero equivalente y el numero fraccionario equivalente.

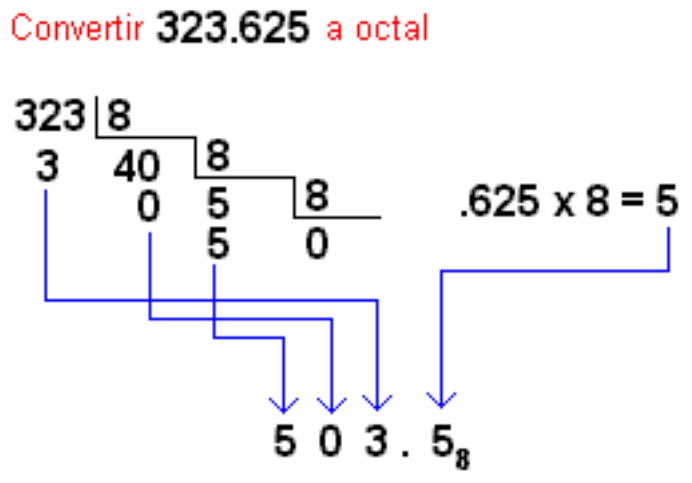


Figura 10: Conversión de decimal a octal

CONVERSIÓN DE UN NUMERO OCTAL A BINARIO

La ventaja principal del sistema de numeración Octal es la facilidad conque pueden realizarse la conversión entre un numero binario y octal. A continuación mostraremos un ejercicio que ilustrará la teoría. Por medio de este tipo de conversiones, cualquier numero Octal se convierte a binario de manera individual. En este ejemplo, mostramos claramente el equivalente **100 111 010** en binario de cada numero octal de forma individual.

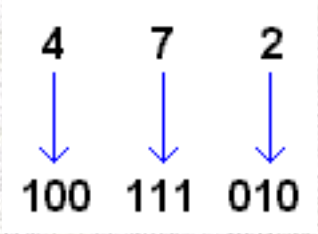


Figura 11: Conversión de octal a binario

CONVERSIÓN DE UN NUMERO DECIMAL A UN NUMERO HEXADECIMAL

Convertir el numero 250.25 a Hexadecimal

1. Se toma la parte entera y se divide sucesivamente por el numero decimal 16 (base) hasta que el cociente sea 0
2. Los números enteros resultantes de los cocientes, pasarán a conformar el numero hexadecimal correspondiente, teniendo en cuenta que el sistema de numeración hexadecimal posee solo 16 símbolos, donde los números del 10 hasta el 15 tienen símbolos alfabéticos que ya hemos explicado
3. La parte fraccionaria del numero a convertir se multiplica por 16 (Base) sucesivamente hasta que el producto resultante no tenga parte fraccionaria
4. Al igual que en los sistemas anteriores, el numero equivalente se forma, de la unión de los dos números equivalentes, tanto entero como fraccionario, separados por un punto que establece la diferencia entre ellos.

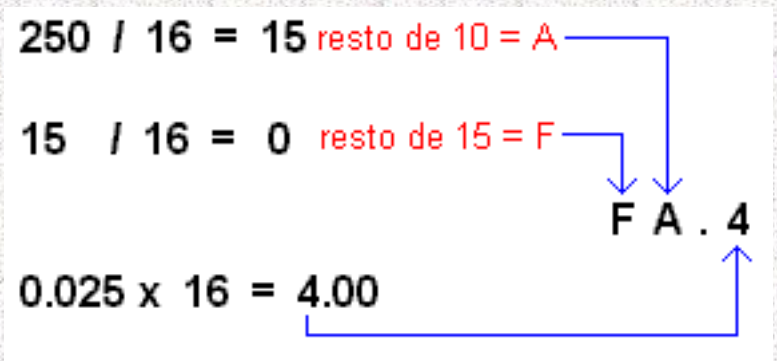


Figura 12: Conversión de decimal a hexadecimal

CONVERSIÓN DE UN NUMERO HEXADECIMAL A UN NUMERO DECIMAL

Como en los ejemplos anteriores este también nos ayudará a entender mejor este procedimiento: Convertir el numero hexadecimal 2B6 a su equivalente decimal.

1. Multiplicamos el valor de posición de cada columna por el dígito hexadecimal

correspondiente.

2. El resultado del número decimal equivalente se obtiene, sumando todos los productos obtenidos en el paso anterior.

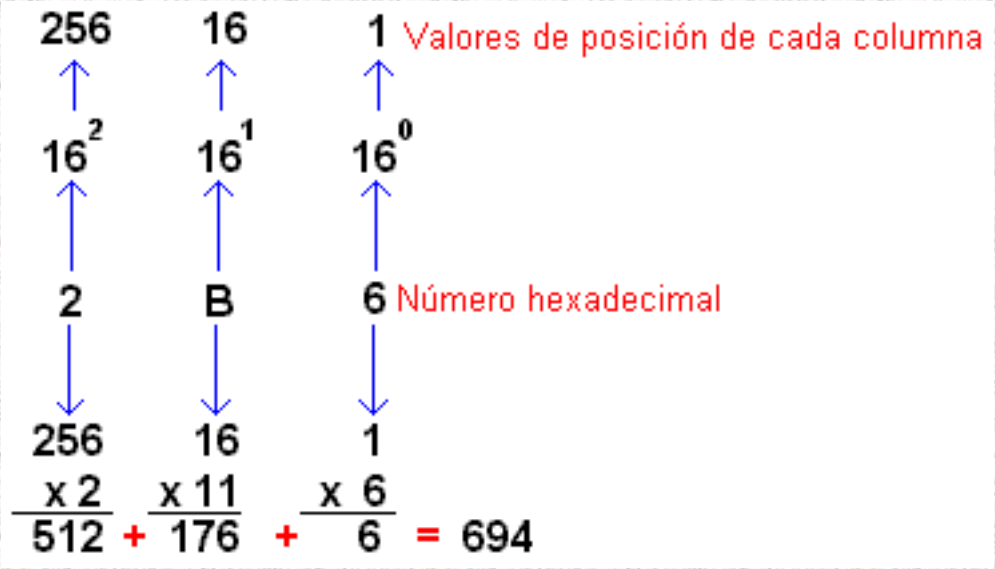


Figura 13: Conversión de hexadecimal a decimal

SISTEMA DE NÚMEROS EN COMPLEMENTO A 2

Este es un sistema que nos permite representar números binarios de forma negativa, en donde el MSB (Bit mas Significativo) es el bit del signo. Si este bit es 0 entonces el numero binario es positivo (+), si el bit del signo es 1, entonces el numero es negativo(-) los siete bits restantes del registro representan la magnitud del numero **1010110**, para complementar mejor la explicación tendremos que dedicarle mucha atención a la explicación de conversiones donde interviene este tipo de numeración, que es bastante utilizado en los microprocesadores, ya que estos manejan tanto números positivos como números negativos.

Para comprender mejor la conversión de sistema de numeración de este sistema de numeración, hay que tener en cuenta las siguientes definiciones

FORMA COMPLEMENTO A 1

El complemento a 1 de un numero binario se obtiene cambiando cada 0 por 1 y viceversa. En otras palabras, se cambia cada bit del numero por su complemento.

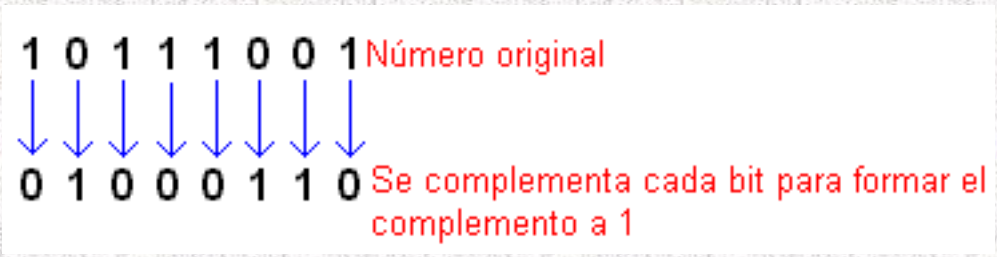


Figura 14: Complemento a uno

FORMA COMPLEMENTO A 2

El complemento a 2 de un numero binario se obtiene tomando el complemento a 1, y sumándole 1 al bit menos significativo. A continuación se ilustra este proceso para el numero 1001 = 9

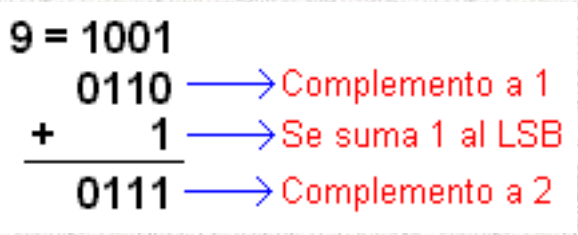


Figura 15: Complemento a 2

Cuando se agrega el bit de signo 1 al MSB, el numero complemento a 2 con signo se convierte en **10111** y es el numero equivalente al - 9.