ANTOLOGIA

ELECTRONICA II

INGENIERÍA EN SISTEMAS COMPUTACIONALES

6°CUATRIMESTRE

Marco Estratégico de Referencia

ANTECEDENTES HISTORICOS

Nuestra Universidad tiene sus antecedentes de formación en el año de 1979 con el inicio de actividades de la normal de educadoras "Edgar Robledo Santiago", que en su momento marcó un nuevo rumbo para la educación de Comitán y del estado de Chiapas. Nuestra escuela fue fundada por el Profesor de Primaria Manuel Albores Salazar con la idea de traer Educación a Comitán, ya que esto representaba una forma de apoyar a muchas familias de la región para que siguieran estudiando.

En el año 1984 inicia actividades el CBTiS Moctezuma Ilhuicamina, que fue el primer bachillerato tecnológico particular del estado de Chiapas, manteniendo con esto la visión en grande de traer Educación a nuestro municipio, esta institución fue creada para que la gente que trabajaba por la mañana tuviera la opción de estudiar por las tarde.

La Maestra Martha Ruth Alcázar Mellanes es la madre de los tres integrantes de la familia Albores Alcázar que se fueron integrando poco a poco a la escuela formada por su padre, el Profesor Manuel Albores Salazar; Víctor Manuel Albores Alcázar en septiembre de 1996 como chofer de transporte escolar, Karla Fabiola Albores Alcázar se integró como Profesora en 1998, Martha Patricia Albores Alcázar en el departamento de finanzas en 1999.

En el año 2002, Víctor Manuel Albores Alcázar formó el Grupo Educativo Albores Alcázar S.C. para darle un nuevo rumbo y sentido empresarial al negocio familiar y en el año 2004 funda la Universidad Del Sureste.

La formación de nuestra Universidad se da principalmente porque en Comitán y en toda la región no existía una verdadera oferta Educativa, por lo que se veía urgente la creación de una institución de Educación superior, pero que estuviera a la altura de las exigencias de los jóvenes que tenían intención de seguir estudiando o de los profesionistas para seguir preparándose a través de estudios de posgrado.

Nuestra Universidad inició sus actividades el 18 de agosto del 2004 en las instalaciones de la 4ª avenida oriente sur no. 24, con la licenciatura en Puericultura, contando con dos grupos de cuarenta alumnos cada uno. En el año 2005 nos trasladamos a nuestras propias instalaciones en la carretera Comitán – Tzimol km. 57 donde actualmente se encuentra el campus Comitán y el Corporativo UDS, este último, es el encargado de estandarizar y controlar todos los procesos operativos y Educativos de los diferentes Campus, Sedes y Centros de Enlace Educativo, así como de crear los diferentes planes estratégicos de expansión de la marca a nivel nacional e internacional.

Nuestra Universidad inició sus actividades el 18 de agosto del 2004 en las instalaciones de la 4ª avenida oriente sur no. 24, con la licenciatura en Puericultura, contando con dos grupos de cuarenta alumnos cada uno. En el año 2005 nos trasladamos a nuestras propias instalaciones en la carretera Comitán – Tzimol km. 57 donde actualmente se encuentra el campus Comitán y el corporativo UDS, este último, es el encargado de estandarizar y controlar todos los procesos operativos y educativos de los diferentes campus, así como de crear los diferentes planes estratégicos de expansión de la marca.

MISIÓN

Satisfacer la necesidad de Educación que promueva el espíritu emprendedor, aplicando altos estándares de calidad Académica, que propicien el desarrollo de nuestros alumnos, Profesores, colaboradores y la sociedad, a través de la incorporación de tecnologías en el proceso de enseñanza-aprendizaje.

VISIÓN

Ser la mejor oferta académica en cada región de influencia, y a través de nuestra Plataforma Virtual tener una cobertura Global, con un crecimiento sostenible y las ofertas académicas innovadoras con pertinencia para la sociedad.

VALORES

- Disciplina
- Honestidad
- Equidad
- Libertad

ESCUDO

El escudo de la UDS, está constituido por tres líneas curvas que nacen de izquierda a derecha formando los escalones al éxito. En la parte superior está situado un cuadro motivo de la abstracción de la forma de un libro abierto.

ESLOGAN

"Mi Universidad"

ALBORES

Es nuestra mascota, un Jaguar. Su piel es negra y se distingue por ser líder, trabaja en equipo y obtiene lo que desea. El ímpetu, extremo valor y fortaleza son los rasgos que distinguen.

ELECTRÓNICA II.

Objetivo de la materia:

Seleccionará y analizará los diferentes dispositivos de tiempo y digitales con la finalidad de acoplar a diferentes analogías de los circuitos electromecánicos.

Unidad 1

CIRCUITOS DE TIEMPO

1.1.- Estructuras y especificaciones de los circuitos temporizadores integrados.

1.2.- Explicación de ciclos de trabajo.

1.3.- Circuitos temporizadores.

1.4.- Circuitos Multivibradores.

1.5.- Aplicaciones.

Unidad 2

SISTEMAS NUMÉRICOS Y CÓDIGOS

2.1.- Sistemas de numéricos.

2.2.- Aritmética.

2.3.- Conversiones de base.

2.4.- Representación de números por signo.

2.5.- Códigos de computadora.

Unidad 3

MÉTODOS ALGEBRAICOS PARA EL ANÁLISIS Y SÍNTESIS DE CIRCUITOS LÓGICOS

3.1.- Fundamentos del álgebra booleana.

3.2.- Funciones de conmutación.

3.3.- Circuitos de conmutación.

3.4.- Simplificación de funciones.

3.5.- Mapas de Karnaugh.

3.6.- Método de minimización Tabular de Quine-McCluskey.

3.7.- Lógica combinatoria.

3.7.1.- Decodificadores y Codificadores.

3.7.2.- Multiplexores y Demultiplexores.

3.7.3.- Comparadores.

Unidad 4

DISPOSITIVOS SECUENCIALES

4.1.- Flip Flops.

4.2.- Registros de corrimiento.

4.3.- Contadores.

4.4.- Modelos de circuitos secuenciales sincronos.

4.5.- Análisis y síntesis de un circuito secuencial síncrono.

4.6.- Tipos de circuitos asíncronos.

4.7.- Aplicaciones.

Criterios de evaluación:

No	Concepto	Porcentaje
1	Actividades web escolar	20%
2	Actividades Áulicas	20%
3	Examen	60%
Total de Criterios de evaluación		100%

INDICE

UNIDAD I

CIRCUITOS DE TIEMPO

1.1.- Estructuras y especificaciones de los circuitos temporizadores integrados.

El temporizador IC 555 es un circuito integrado (chip) que se utiliza en la generación de temporizadores, pulsos y oscilaciones. El 555 puede ser utilizado para proporcionar retardos de tiempo, como un oscilador, y como un circuito integrado flip flop. Sus derivados proporcionan hasta cuatro circuitos de sincronización en un solo paquete.

Fue introducido en 1971 por Signetics, el 555 sigue siendo de uso generalizado debido a su facilidad de uso, precio bajo y la estabilidad. Muchas empresas los fabrican en versión de transistores bipolares y también en CMOS de baja potencia. A partir de 2003, se estimaba que mil millones de unidades se fabricaban cada año. Este circuito suele ser utilizado para trabajos sencillos como trabajos escolares, debido a su bajo costo y facilidad de trabajar con él.

1.2.- Explicación de ciclos de trabajo.

En 1970, Hans R. Camenzind, un ingeniero nacido en Suiza, quien después de terminar su educación secundaria viajó a Estados Unidos para realizar los estudios de ingeniería, se tomó un mes de vacaciones de su empleo en Signetics (ahora Philips) para escribir un libro, pero en vez de volver al final de las vacaciones, le pidió a la compañía que lo contratase como consultor durante un año, para usar los principios del oscilador controlado por tensión o VCO en el desarrollo de un circuito integrado temporizador; esta idea no era del agrado del departamento de ingeniería de Signetics, pero afortunadamente a Art Fury, el responsable de Mercadotecnia de la empresa, la idea le entusiasmó y le dio el contrato a Camenzind, quien después de seis meses, completó el diseño final (los primeros diseños no hacían uso de redes RC para la temporización y por ello preveían un circuito integrado de 14 patillas que era mucho más complejo y caro)

El 555 fue pionero en muchos aspectos, no solo fue el primer circuito integrado temporizador, también fue el primero en venderse desde su salida al mercado a bajo precio (US $0,75), cosa nunca hecha hasta entonces por ningún productor de semiconductores. Cabe acotar que por las diferencias entre Camenzind y el departamento de ingeniería de Signetics, el proyecto durmió durante un año antes de ser finalmente producido en masa por Signetics.

El temporizador fue introducido en el mercado en el año 1972 por Signetics con el nombre SE555/NE555 y fue llamado "The IC Time Machine" (La Máquina del Tiempo en Circuito Integrado). Aunque en la actualidad se emplea más su versión en tecnología CMOS desarrollada por Dave Bingham en Intersil, el SE555 se sigue usando también la versión bipolar original, especialmente en aplicaciones que requieran grandes corrientes

1.3.- Circuitos temporizadores.

El 555 es un circuito integrado cuya función principal es producir pulsos de temporización con precisión, entre sus funciones secundarias están la de oscilador, divisor de frecuencia, modulador o generador.

Este circuito integrado incorpora dentro de si, dos comparadores de voltaje, un flip flop, una etapa de salida de corriente, un divisor de voltaje por resistor y un transistor de descarga. Dependiendo de cómo se interconecten estas funciones utilizando componentes externos es posible conseguir que dicho circuito realiza un gran número de funciones tales como la del multivibrador astable y la del circuito monoestable.

1.4.- Circuitos Multivibradores.

Este tipo de funcionamiento multivibrador astable con temporizador 555 se caracteriza por una salida con forma de onda cuadrada (o rectangular) de ancho predefinido por el diseñador del circuito y que se repite en forma continua.

El esquema de conexión y las formas de onda de entrada y salida del multivibrador astable se muestran en los gráficos más adelante.

La señal de salida tiene un nivel alto por un tiempo T1 y en un nivel bajo un tiempo T2. Los tiempos de duración, tanto en nivel alto como en nivel bajo, dependen de los valores de las resistencias R1 y R2 y del condensador C.

1.5.- Aplicaciones.

El 555 tiene diversas aplicaciones, como: Control de sistemas secuenciales, divisor de frecuencias, modulación por ancho de pulso, generación de tiempos de retraso, repetición de pulsos, etc.

UNIDAD II

SISTEMAS NUMÉRICOS Y CÓDIGOS

2.1.- Sistemas de numéricos.

La importancia del sistema decimal radica en que se utiliza universalmente para representar cantidades fuera de un sistema digital. Es decir que habrá situaciones en las cuales los valores decimales tengan que convenirse en valores binarios antes de que se introduzcan en sistema digital. Entonces habrá situaciones en que los valores binarios de las salidas de un circuito digital tengan que convertir a valores decimales para presentarse al mundo exterior.

Por otro lado del binario y el decimal, otros dos sistemas de numeración encuentran amplias aplicaciones en los sistemas digitales. Los sistemas octal (base y hexadecimal (base 16) se usan con el mismo fin, que es ofrecer un eficaz medio de representación de números binarios grandes. Como veremos, ambos sistemas numéricos tienen la ventaja de que pueden convenirse fácilmente al y del binario.

Definición

Para poder representar los números naturales se utilizan distintos sistemas de numeración. Cada uno de ellos está compuesto por un conjunto de símbolos y reglas.

El sistema más utilizado se denomina sistema decimal ya que utiliza diez cifras que forman la base del sistema:

Se llama cifra o dígito a cada uno de los símbolos que forman la base del sistema de numeración decimal.

Se llama base del sistema de numeración a la cantidad de elementos que se combinan.

Valor posicional y relativo de cada dígito

Esto quiere decir que dependiendo de la posición en donde se ubique cada dígito el valor que éste tendrá.

Así por ejemplo, vemos que el valor del número 2 en 3.245 no es el mismo que en el 332, esto debido a que los dígitos actúan como multiplicadores de las potencias de la base.

Así tenemos que en el número 3.245 el 2 se ubica en las centenas, por lo que su valor posicional será de 2*100, es decir 200. Sin embargo, en el número 332 su valor equivaldrá a la multiplicación de 2*1, es decir 2, ya que el 2 se encuentra en la posición de las unidades. Por otro lado, si recordamos cuál es el valor de cada base tendremos:

Unidades   1

Decenas   10

Centenas 100

Unidades de Mil   1.000

Decenas de Mil   10.000

Centenas de Mil   100.000

2.2.- Aritmética.

Actualmente las definiciones de aritmética y digital por separado, son algo común ante la sociedad, según el panorama de la misma, se conoce que la Aritmética: es aquella parte de la matemática que estudia los números y las operaciones que se realizan con ellos, por su parte el término Digital: se divaga como algo que está estrechamente vinculado con la tecnología y la informática. Sin embargo, al unir ambos términos "Aritmética Digital", resulta un nuevo concepto, que pasa por desapercibido, con el extremo desconocimiento de no comprender, que la aritmética digital forma parte de nuestra vida cotidiana. De modo que, se analizará el impacto de la aritmética digital, para con nuestro diario vivir, mediante, las causas por las que ésta surgió, los usos o aplicaciones que actualmente desempeña, y desde luego las limitaciones que implica.

Sin duda la mayoría de personas manipula diariamente al menos un dispositivo electrónico (celular, laptod, mp3, ipad……) incluyendo a los niños, no obstante, para su funcionamiento, se necesitaba de un conjunto de: procesos, operaciones, datos, funciones; para lograr establecer el uso deseado, es ahí, donde surge el concepto de aritmética digital para ejecutar y hacer posible estos procedimientos.

En el campo de la electrónica digital, la aritmética digital o aritmética binaria se define: como la unidad aritmética lógica, también conocida como ALU (arithmetic logic unit), que es un circuito digital que calcula operaciones aritméticas como (suma, resta, multiplicación y división), y operaciones lógicas (true, or, and, false, not) entre uno o dos valores de los argumentos. En 1945 John P. Eckert y John W. Mauchly le dieron vida a este concepto, y más tarde John Neuman publicó un informe, explicando la necesidad de una ALU para el uso de una computadora.

Por mucho, los circuitos electrónicos más complejos que forman parte del hardware de los diferentes dispositivos que usamos a diario, son los que están construidos dentro de los chips de microprocesadores modernos como el Intel Core Duo, en su causa, estos procesadores tienen dentro de ellos un ALU muy complejo y potente, puesto que, un microprocesador moderno puede tener múltiples núcleos, cada núcleo con múltiples unidades de ejecución y cada una de ellas con múltiples ALU. Por su parte, el cálculo aritmético digital desempeña un papel crucial en el procesamiento de información que se representan en forma binaria (bits de 0 y 1) dentro de una computadora, y demás aparatos electrónicos.

En efecto, la unidad aritmética lógica (ALU) basada en el sistema de numeración en base 2 con un número fijo de bits, posee circuitos aritméticos que hacen operaciones sobre datos de n bits, las formas de hacer operaciones aritméticas en los diferentes dispositivos especialmente en ordenadores, son las siguientes:

En hardware (mediante circuitos específicos): tradicionalmente solo para operaciones simples (suma, producto, etc.)

El software (mediante programación): tradicionalmente para operaciones complejas (trigonometría, derivación, integración, etc.)

Asimismo, se puntualizan otro tipo de operaciones que ALU como componente del CPU realiza, por ejemplo, la operación de contar, donde un contador es un circuito que almacena el número de veces que ha tenido lugar un determinado proceso o evento, de igual forma las operaciones de desplazamiento de bits, mediante esta desplazan o rotan una palabra en un número específico de bits hacia la izquierda o a la derecha. De modo que, ALU es la parte que realmente realiza las operaciones aritméticas y lógicas con los datos, el resto de los elementos como: (registros, memoria, E/S), están principalmente para suministrar datos, ante lo cual ALU, se considera como el núcleo central, de un dispositivo electrónico. Cabe señalar que en ningún circuito electrónico se suprime de realizar algún tipo de operación aritmética, incluso dentro de un reloj tendrá una ALU minúscula que se mantiene sumando 1 al tiempo actual.

Una limitación para los ingenieros, sucede que se puede diseñar una ALU para calcular cualquier operación, sin importar lo compleja que sea; el problema es que cuanto más compleja sea la operación, tanto más costosa será la ALU, más espacio usará el procesador, por ende, disipará más energía. Por otro lado, generalmente los ingenieros llaman ALU al circuito que realiza operaciones aritméticas en formatos de numero entero (complemento a dos), entonces ¿qué sucede con el formato de números flotantes?, por consiguiente para realizar esto se necesitan circuitos que calculen las operaciones en formatos más complejos, y ALU presenta esa desventaja, de tal manera que para trabajar con operaciones complejas, se ha diseñado un nuevo conjunto de aritmética llamado, Unidad de punto flotante, FPU (floating point unit) hace las mismas operaciones que ALU, pero con representación de punto flotante, no obstante, para hacer estos cálculos una FPU tiene incorporados varios circuitos complejos.

Entonces se concluye que la aritmética digital (ALU), se encuentra en todo tipo de circuitos y dispositivos electrónicos, los mismos que utilizamos a diario, cuya aplicación se destaca y sobresale para el desarrollo de nuestras actividades, incluso, es novedoso e interesante, la comparación que se establece entre una computadora y el ser humano, la cual se explica bajos los procesos de la aritmética binaria, es decir, que se examinaron algunas de las operaciones básicas de cada individuo, como son las acciones de contar, ordenar, calcular, etc., para mediante una estrategia o medio llamado "aritmética digital o binaria (ALU)" lograr que una maquina (computadora) haga lo mismo, es así, que los diferentes dispositivos electrónicos de manera indirecta suelen reemplazar nuestras actividades, sin duda, se logra observar el gran impacto que esto implica en nuestro contexto de desarrollo.

2.3.- Conversiones de base.

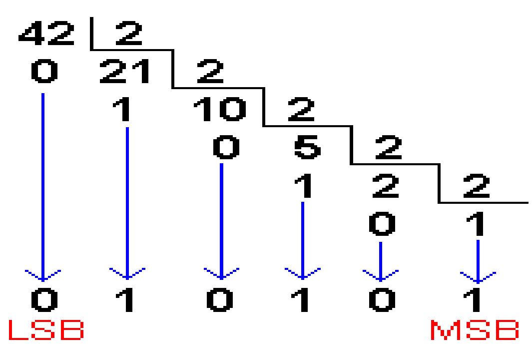
ONVERSIÓN DE UN NUMERO DECIMAL A BINARIO

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 número 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.



CONVERSIÓN DE UN NUMERO DECIMAL FRACCIONARIO A UN NUMERO BINARIO

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

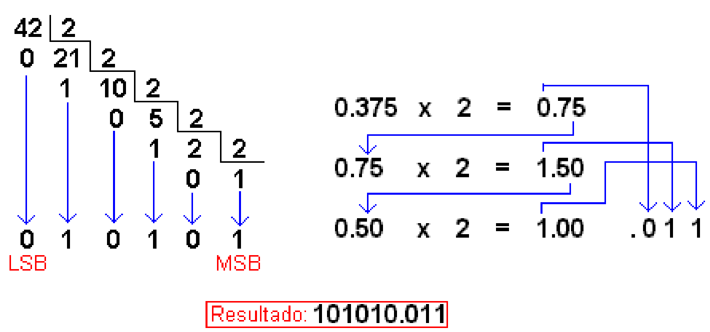
1. la parte entera se transforma de igual forma que el ejemplo anterior.

2. La parte fraccionaria de la siguiente manera:

Multiplicamos por el número 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 último .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 número binario correspondiente a el numero decimal.

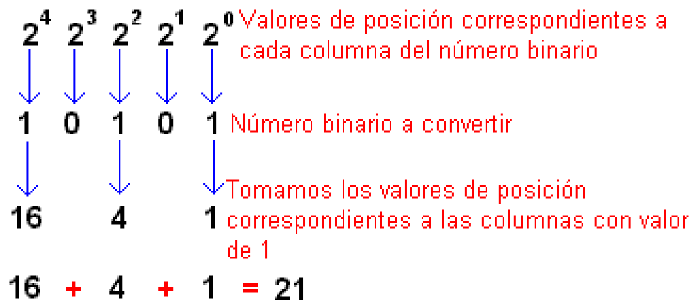


CONVERSIÓN DE UN NUMERO BINARIO A UN NUMERO DECIMAL

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

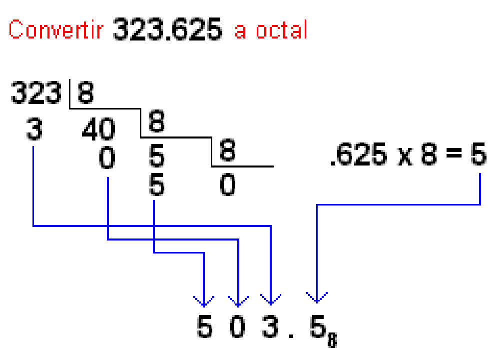
2. Sumamos los valores de posición para identificar el numero decimal equivalente



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.



CONVERSIÓN DE UN NUMERO OCTAL A BINARIO

La ventaja principal del sistema de numeración Octal es la facilidad con que 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 número octal de forma individual.

4 7 2

100 111 010

CONVERSIÓN DE UN NUMERO DECIMAL A UN NUMERO HEXADECIMAL

Convertir el número 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 número 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 número 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.

250 / 16 = 15 resto de 10 = A

15 / 16 = 0 resto de 15 = F

F A . 4

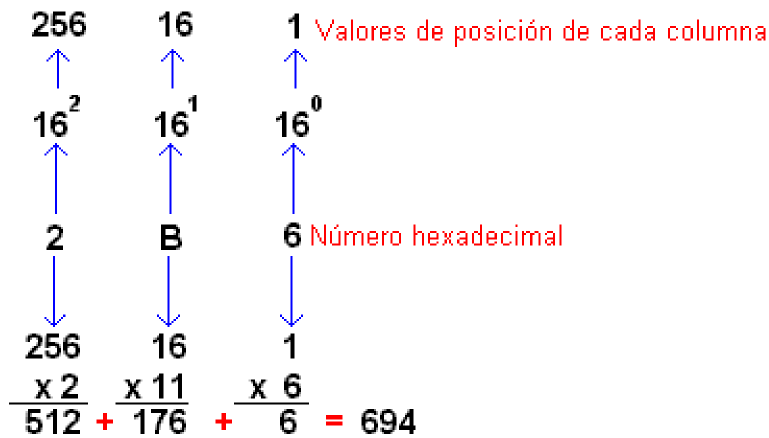
0.025 x 16 = 4.00

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.



2.4.- Representación de números por signo.

En matemáticas, los números negativos en cualquier base se representan del modo habitual, precediéndolos con un signo «−». Sin embargo, en una computadora hay varias formas de representar el signo de un número. Este artículo trata cuatro métodos de extender el sistema binario para representar números con signo: signo y magnitud, complemento a uno, complemento a dos y exceso K, donde normalmente K equivale a bn-1 - 1.

Para la mayoría de usos, las computadoras modernas utilizan típicamente la representación en complemento a dos, aunque pueden usarse otras en algunas circunstancias.

En las secciones siguientes nos referiremos exclusivamente al caso de números signados en binario (y contrastaremos con el decimal con fines didácticos). Esto no significa que lo mostrado aquí se pueda llevar en forma análoga a otras bases (hexadecimal, u octal, por ejemplo). El valor absoluto de un número es la distancia que lo separa del cero en la recta numérica; es el propio número tras prescindir de su signo. El valor absoluto se escribe entre barras: | |. Valor absoluto de 3:|3| =3. El valor absoluto de -3: |-3| =3. Los números menores que cero son por supuesto los números negativos.

El número que tiene como valor absoluto 125 y es menor que cero es -125 porque el valor absoluto solo toma en cuenta la distancia, no la dirección, razón por la cual este solo puede ser positivo o cero. |+ Para n = 8 (8 bits) en Signo y Magnitud

Un primer enfoque al problema de representar un número signado de n-bits consiste en asignar:

un bit para representar el signo. Ese bit a menudo es el bit más significativo o MSB (de sus siglas en inglés) y, por convención: un 0 denota un número positivo, y un 1 denota un número negativo;

los (n-1)-bits restantes para representar el significando que es la magnitud del número en valor absoluto.

Y se conoce como Signo y Magnitud.

Este enfoque es directamente comparable a la forma habitual de mostrar el signo (colocando "+" o "-" al lado de la magnitud del número). Algunas de las primeras computadoras binarias (la IBM 7090) utilizaron esta representación, quizás por su relación obvia con la práctica habitual.

El formato Signo y Magnitud es además el habitual para la representación del significando en números en punto flotante.

2.5.- Códigos de computadora.

El código binario es el sistema de codificación usado para la representación de textos, o procesadores de instrucciones de computadora, utilizando el sistema binario (sistema numérico de dos dígitos, o bit: el "0" y el "1"). . En informática y telecomunicaciones, el código binario se utiliza en la codificación de datos, tales como cadenas de caracteres, o cadenas de bits Por ejemplo en el caso de un CD, las señales que reflejarán el "láser" que rebotará en el CD y será recepcionado por un sensor de distinta forma indicando así, si es un cero o un uno. En un código binario de ancho fijo, cada letra, dígito, u otros símbolos, están representados por una cadena de bits de la misma longitud, como un número binario que, por lo general, aparece en las tablas en notación octal, decimal o hexadecimal.

Según Anton Glaser, en su History of Binary and other Nondecimal Numeration, comenta que los primeros códigos binarios se utilizaron en el año 1932: C.E. Wynn-Williams ("Scale of Two"), posteriormente en 1938: Atanasoff-Berry Computer, y en 1939: Stibitz ("excess three") el código en Complex Computer.

Es frecuente también ver la palabra bit referida bien a la ausencia de señal, expresada con el dígito "0", o bien referida a la existencia de la misma, expresada con el dígito "1". El byte es un grupo de 8 bits, es decir en él tenemos 256 posibles estados binarios.

UNIDAD III

MÉTODOS ALGEBRAICOS PARA EL ANÁLISIS Y SÍNTESIS DE CIRCUITOS LÓGICOS

3.1.- Fundamentos del álgebra booleana.

Postulados básicos

La descripción básica de la formulación del algebra booleana se basa en conceptos de la teoría de conjuntos, donde se define formalmente un algebra booleana como un conjunto matemático distributivo y complementado. Resumiremos aquí esta definición mediante un conjunto de postulados que sintetiza los elementos y propiedades básicos de un algebra booleana.

Postulado 1(definición)

Un algebra booleana es un sistema algebraico cerrado formado por un conjunto K de dos o mas elementos y los dos operadores • y +; de manera alternativa para cada a y b un conjunto K, a • b pertenece a K y a + b pertenece a K (+ se llama OR y • se llama AND).

El álgebra booleana es un sistema matemático deductivo centrado en los valores cero y uno (falso y verdadero). Un operador binario " ° " definido en éste juego de valores acepta un par de entradas y produce un solo valor booleano, por ejemplo, el operador booleano AND acepta dos entradas booleanas y produce una sola salida booleana. Para cualquier sistema algebraico existen una serie de postulados iniciales, de aquí se pueden deducir reglas adicionales, teoremas y otras propiedades del sistema, el álgebra booleana a menudo emplea los siguientes postulados:

Cerrado. El sistema booleano se considera cerrado con respecto a un operador binario si para cada par de valores booleanos se produce un solo resultado booleano.

Conmutativo. Se dice que un operador binario " ° " es conmutativo si A ° B = B ° A para todos los posibles valores de A y B.

Asociativo. Se dice que un operador binario " ° " es asociativo si (A ° B) ° C = A ° (B ° C) para todos los valores booleanos A, B, y C.

Distributivo. Dos operadores binarios " ° " y " % " son distributivos si A ° (B % C) = (A ° B) % (A ° C) para todos los valores booleanos A, B, y C.

Identidad. Un valor booleano I se dice que es un elemento de identidad con respecto a un operador binario " ° " si A ° I = A.

Inverso. Un valor booleano I es un elemento inverso con respecto a un operador booleano " ° " si A ° I = B, y B es diferente de A, es decir, B es el valor opuesto de A.

Los dos posibles valores en el sistema booleano son cero y uno, a menudo se llama a éstos valores respectivamente como falso y verdadero. - El símbolo · representa la operación lógica AND. Cuando se utilicen nombres de variables de una sola letra se eliminará el símbolo ·, por lo tanto, AB representa la operación lógica AND entre las variables A y B, a esto también le llamamos el producto entre A y B. - El símbolo "+" representa la operación lógica OR, decimos que A+B es la operación lógica OR entre A y B, también llamada la suma de A y B. - El complemento lógico, negación ó NOT es un operador unitario, en éste texto utilizaremos el símbolo " ' " para denotar la negación lógica, por ejemplo, A' denota la operación lógica NOT de A. - Si varios operadores diferentes aparecen en una sola expresión booleana, el resultado de la expresión depende de la procedencia de los operadores, la cual es de mayor a menor, paréntesis, operador lógico NOT, operador lógico AND y operador lógico OR. Tanto el operador lógico AND como el OR son asociativos por la izquierda. Si dos operadores con la misma procedencia están adyacentes, entonces se evalúan de izquierda a derecha. El operador lógico NOT es asociativo por la derecha.

Postulados

P1 El álgebra booleana es cerrada bajo las operaciones AND, OR y NOT

P2 El elemento de identidad con respecto a · es uno y con respecto a + es cero. No existe elemento de identidad para el operador NOT

P3 Los operadores · y + son conmutativos.

P4 · y + son distributivos uno con respecto al otro, esto es, A · (B+C) = (A ·B)+(A ·C) y A+(B ·C) = (A+B) ·(A+C).

P5 Para cada valor A existe un valor A' tal que $A \cdot A' = 0$ y $A+A' = 1$. Éste valor es el complemento lógico de A.

P6 · y + son ambos asociativos, ésto es, $(AB)\,C = A\,(BC)$ y $(A+B)+C = A+(B+C)$.

Teoremas del álgebra booleana

Es posible probar todos los teoremas del álgebra booleana utilizando éstos postulados, además es buena idea familiarizarse con algunos de los teoremas más importantes de los cuales podemos mencionar los siguientes:

Teorema 1: $A + A = A$

Teorema 2: $A \cdot A = A$

Teorema 3: $A + 0 = A$

Teorema 4: $A \cdot 1 = A$

Teorema 5: $A \cdot 0 = 0$

Teorema 6: $A + 1 = 1$

Teorema 7: $(A + B)' = A' \cdot B'$

Teorema 8: $(A \cdot B)' = A' + B'$

Teorema 9: $A + A \cdot B = A$

Teorema 10: $A \cdot (A + B) = A$

Teorema 11: $A + A'B = A + B$

Teorema 12: $A' \cdot (A + B') = A'B'$

Teorema 13: $AB + AB' = A$

Teorema 14: $(A' + B') \cdot (A' + B) = A'$

Teorema 15: $A + A' = 1$

Teorema 16: $A \cdot A' = 0$

Los teoremas siete y ocho son conocidos como Teoremas de DeMorgan en honor al matemático que los descubrió.

Características

Un álgebra de Boole es un conjunto en el que destacan las siguientes características: 1- Se han definido dos funciones binarias (que necesitan dos parámetros) que llamaremos aditiva (que representaremos por x + y) y multiplicativa (que representaremos por xy) y una función monaria (de un solo parámetro) que representaremos por x'. 2- Se han definido dos elementos (que designaremos por 0 y 1) Y 3- Tiene las siguientes propiedades:

Conmutativa respecto a la primera función: x + y = y + x

Conmutativa respecto a la segunda función: xy = yx Asociativa respecto a la primera función: (x + y) + z = x + (y +z) Asociativa respecto a la segunda función: (xy)z = x(yz) Distributiva respecto a la primera función: (x +y)z = xz + yz Distributiva respecto a la segunda función: (xy) + z = (x + z)(y + z) Identidad respecto a la primera función: x + 0 = x Identidad respecto a la segunda función: x1 = x Complemento respecto a la primera función: x + x' = 1 Complemento respecto a la segunda función: xx' = 0

Propiedades del Álgebra De Boole

Idempotente respecto a la primera función: x + x = x Idempotente respecto a la segunda función: xx = x Maximalidad del 1: x + 1 = 1 Minimalidad del 0: x0 = 0 Involución: x = x Inmersión respecto a la primera función: x + (xy) = x Inmersión respecto a la segunda función: x(x + y) = x Ley de Morgan respecto a la primera función: (x + y)' = x'y' Ley de Morgan respecto a la segunda función: (xy)' = x' + y'

Función Booleana

Una función booleana es una de A x A x A x....A en A, siendo A un conjunto cuyos elementos son 0 y 1 y tiene estructura de álgebra de Boole. Supongamos que cuatro amigos deciden ir al cine si lo quiere la mayoría. Cada uno puede votar si o no. Representemos el voto de cada uno por xi. La función devolverá sí (1) cuando el numero de votos afirmativos sea 3 y en caso contrario devolverá 0. Si x1 vota 1, x2 vota 0, x3 vota 0 y x4 vota 1 la función booleana devolverá 0. Producto mínimo (es el número posible de casos) es un producto en el que aparecen todas las variables o sus negaciones. El número posible de casos es 2n.

Las funciones booleanas se pueden representar como la suma de productos mínimos (minterms) iguales a 1.

Álgebra Booleana y circuitos electrónicos

La relación que existe entre la lógica booleana y los sistemas de cómputo es fuerte, de hecho se da una relación uno a uno entre las funciones booleanas y los circuitos electrónicos de compuertas digitales. Para cada función booleana es posible diseñar un circuito electrónico y viceversa, como las funciones booleanas solo requieren de los operadores AND, OR y NOT podemos construir nuestros circuitos utilizando exclusivamente éstos operadores utilizando las compuertas lógicas homónimas Un hecho interesante es que es posible implementar cualquier circuito electrónico utilizando una sola compuerta, ésta es la compuerta NAND Para probar que podemos construir cualquier función booleana utilizando sólo compuertas NAND, necesitamos demostrar cómo construir un inversor (NOT), una compuerta AND y una compuerta OR a partir de una compuerta NAND, ya que como se dijo, es posible implementar cualquier función booleana utilizando sólo los operadores booleanos AND, OR y NOT. Para construir un inversor simplemente conectamos juntas las dos entradas de una compuerta NAND. Una vez que tenemos un inversor, construir una compuerta AND es fácil, sólo invertimos la salida de una compuerta NAND, después de todo, NOT (NOT (A AND B)) es equivalente a A AND B. Por supuesto, se requieren dos compuertas NAND para construir una sola compuerta AND, nadie ha dicho que los circuitos implementados sólo utilizando compuertas NAND sean lo óptimo, solo se ha dicho que es posible hacerlo. La otra compuerta que necesitamos sintetizar es la compuerta lógica OR, ésto es sencillo si utilizamos los teoremas de DeMorgan, que en síntesis se logra en tres pasos, primero se reemplazan todos los " ·" por "+" después se invierte cada literal y por último se niega la totalidad de la expresión: A OR B A AND B.......................Primer paso para aplicar el teorema de DeMorgan A' AND B'.....................Segundo paso para aplicar el teorema de DeMorgan (A' AND B')'..................Tercer paso para aplicar el teorema de DeMorgan (A' AND B')' = A' NAND B'.....Definición de OR utilizando NAND Si se tiene la necesidad de construir diferentes compuertas de la manera descrita, bien hay dos buenas razones, la primera es que las compuertas NAND son las más económicas y en segundo lugar es preferible construir circuitos complejos utilizando los mismos bloques básicos. Observe que es posible construir cualquier circuito lógico utilizando sólo compuertas de tipo NOR (NOR = NOT(A OR B)). La correspondencia entre la lógica NAND y la NOR es ortogonal entre la correspondencia de sus formas canónicas. Mientras que la lógica NOR es útil en muchos circuitos, la mayoría de los diseñadores utilizan lógica NAND.

Teoremas Básicos Del Algebra Booleana

Los Teoremas Básicos del álgebra Booleana son: TEOREMA 1 Ley Distributiva A (B+C) = AB+AC

TEOREMA 2 A+A = A AA = A

TEOREMA 3 Redundancia A+AB = A

TEOREMA 4 0+A = A Equivalente a una compuerta OR con una de sus terminales conectada a tierra

Equivalente a una compuerta AND con una de sus terminales conectada a 1

3.2.- Funciones de conmutación.

Una aplicación importante del álgebra booleana es el álgebra de circuitos de conmutación. Un conmutador es un dispositivo con dos estados que son cerrado y abierto y que se denotarán respectivamente 1 y 0.

En esta forma, un álgebra de circuitos de conmutación no es más que un álgebra booleana con dos elementos a saber: 0 y 1.

Notación. Se designará un conmutador con una sola letra: a, b, c, x, y etcétera.

Si dos conmutadores operan en tal forma que se abren y se cierran simultáneamente, se designarán con la misma letra. Si operan en tal forma que cuando uno está abierto el otro está cerrado, y viceversa entonces se designará uno de ellos con una letra y el otro por su complemento.

Un circuito consistente de los conmutadores x e y conectados en paralelo, se designará por x + y, si los conmutadores están conectados en serie se designarán por xy. Para cada circuito serie paralelo corresponderá una expresión algebraica y viceversa, tales expresiones involucran las operaciones (+), (.), (´).

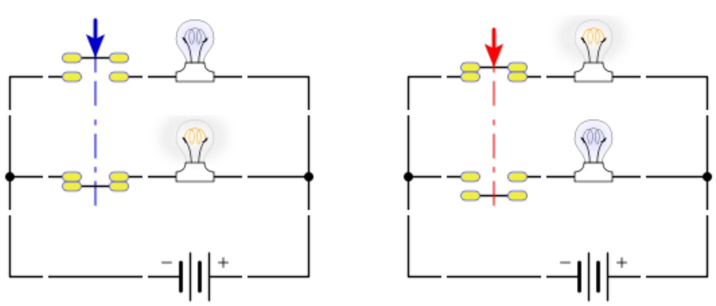
Se dice que dos circuitos de conmutadores son equivalentes si para cualquier posición de los conmutadores de cada circuito o pasa la corriente a través de ambos circuitos (circuitos cerrados) o por ninguno pasa (circuitos abiertos).

Dos expresiones booleanas serán iguales sí sólo sí representan circuitos equivalentes.

Se tendrán en cuenta sólo los factores que determinan si un circuito está abierto o cerrado. Se desecharán problemas referentes a corriente, voltaje, resistencia, etc.

3.3.- Circuitos de conmutación.

En electricidad y electrónica, las leyes del álgebra de Boole y de la lógica binaria, pueden estudiarse mediante circuitos de conmutación. Un circuito de conmutación estará compuesto por una serie de contactos que representarán las variables lógicas de entrada y una o varias cargas que representarán las variables lógicas o funciones de salida.



Los contactos pueden ser normalmente abiertos (NA) o normalmente cerrados (NC). Los primeros permanecerán abiertos mientras no se actúe sobre ellos (por ejemplo, al pulsar sobre interruptor, saturar un transistor, etc.). Los contactos NC funcionarán justamente, al contrario. Esto significa que si se actúa sobre un contacto NA se cerrará y si se hace sobre uno NC se abrirá.

Los circuitos de conmutación se basan en interruptores que permiten o no la circulación de una corriente eléctrica, estos interruptores pueden ser manuales si se actúan directamente, como un interruptor de la luz, por ejemplo; eléctricos: relés o contactores, si su actuación es electro-mecánica, o electrónicos, transistores o puertas lógicas, si se basan en la tecnología electrónica.

Por sencillez, representaremos un interruptor o conmutador por sus contactos eléctricos, si un interruptor conecta dos puntos a y b, diremos que está abierto si no permite la circulación eléctrica entre esos dos puntos: a y b. Diremos que está cerrado si permite la circulación eléctrica entre esos dos puntos.

Un interruptor diremos que esta normalmente abierto (NA) si cuando no se actúa sobre él está abierto, a la posición normal también se le denomina posición de reposo, que el interruptor tendrá normalmente por la actuación de un muelle o resorte que lo lleva a esa posición.

Cuando se actúa sobre un interruptor normalmente abierto (NA), el interruptor se cierra, permitiendo la circulación eléctrica a su través.

Venciendo la fuerza ejercida por el muelle o resorte, y dando lugar al contacto eléctrico entre sus terminales.

Si entre dos puntos a y b, colocamos un interruptor normalmente cerrado (NC), que cuando no se actúa sobre él está cerrado, en este caso, la relajación del muelle o resorte da lugar a poner en contacto los terminales eléctricos del interruptor, permitiendo la circulación eléctrica a su través, el interruptor está cerrado. Si actuamos sobre él venciendo la acción del muelle, separando los contactos, el interruptor se abre, no permitiendo la circulación eléctrica. En estos interruptores el resultado es el contrario de la acción, si actuamos sobre el interruptor el interruptor se abre, cortando el paso de la corriente eléctrica, si no actuamos sobre el, se cierra permitiendo la circulación eléctrica.

Como se ha visto, los interruptores pueden ser actuados manualmente, o mecánicamente mediante fines de carrera, presostatos u otros elementos que partiendo de una acción exterior den lugar a una conexión o desconexión eléctrica.

Pero un circuito puede actuar sobre otro circuito, mediante relés o contactores, de modo que podemos disponer de un circuito de conmutación, cuyo resultado es la actuación sobre otro circuito, en estos casos la presencia o no de una corriente eléctrica da lugar a la modificación del estado de un interruptor, que pasara de su posición de reposo a la de actuado.

En un circuito de conmutación se realiza un análisis de la lógica del circuito, haciendo abstracción de los detalles de funcionamiento de los mecanismos que intervienen, así como del dimensionado de los aparatos y resto del circuito para las intensidades de corriente y diferencia de potencial con los que trabaja, prestando atención prioritaria a la lógica de la conmutación, por ello no son necesarios, algunos de los detalles eléctricos, propios de los circuitos eléctricos, y si es necesario determinar un convenio de representación de los circuitos que impida errores en su interpretación, teniendo en cuenta lo siguiente:

Circuito de conmutación, es un esquema de funcionamiento y no un plano de construcción, por lo tanto la situación de los aparatos se hará según esa lógica.

En un circuito de conmutación no se señalan detalles eléctricos, como intensidades o tensiones eléctricas.

Los aparatos se representan siempre en su posición de reposo, aunque estén conectados directamente a una fuente de energía.

La actuación de los interruptores es siempre de arriba hacia abajo, la posición de reposo es la más alta y la actuada la más baja.

Componentes para un circuito de interruptores.

Un contacto NA representa una variable conmutable

Un contacto NC representa una variable lógica negada (A').

Un circuito cerrado se considera un uno lógico (1).

Un circuito abierto se considera un cero lógico (0).

Si no se actúa sobre un contacto se considera que la variable que representa es 0.

Si se actúa sobre un contacto se considera que la variable que representa es 1.

Si la carga no se excita la función se considera 0 (por ejemplo una lámpara apagada).

Si la carga se excita la función se considera 1 (lámpara encendida).

Interruptor múltiple

Un interruptor múltiple, es el que con solo un mando mueve varios contactos simultáneamente, este tipo de interruptor, no tan sencillo, se emplea para conmutar varios circuitos al mismo tiempo, electivamente separados.

El relé

Un relé o Contactor, es un interruptor automático controlado eléctricamente, de este modo una señal eléctrica da lugar a nuevos contactos que, a su vez, alimentan o dejan de alimentar otros circuitos.

Oscilador electromecánico

La construcción de un Oscilador, con medios exclusivamente electromecánicos, se hace sencillamente, conectando la bobina de un relé a uno de sus contactos normalmente conectados (NC), cuando el relé se excita, el contacto (NC) se desconecta, desconectando la bobina, que da lugar a que el contacto (NC) entre en contacto de nuevo.

Este es el mecanismo en el que se basa el timbre eléctrico clásico.

3.4.- Simplificación de funciones.

Simplificación de funciones a partir del uso de teoremas

Cuando una expresión de Boole es pasada a compuertas, cada una de las literales en la expresión corresponde a una entrada para el circuito, lo que se traduce en componentes. Debido a que el costo es generalmente un factor importante en la construcción de un circuito, es deseable que el circuito a construir tenga el menor número de compuertas lógicas. Para cumplir con este objetivo, se utilizan los teoremas del álgebra booleana para simplificar expresiones booleanas.

Al simplificar una expresión booleana se puede llegar a distintos resultados, lo ideal es llegar a la expresión mínima.

Ejemplo. Simplifique la siguiente expresión booleana: $XY(Z+Y'X)+YZ$

$$
\begin{aligned}
XY(Z+Y'X)+YZ &= XY'Z + XYYZ + YZ && (8)\\
&= XY'Z + 0 + YZ && (5D)\\
&= Z(XY + Y') && (8)\\
&= Z(X' + Y') && (11D)\\
&= Z(XY)' && (12)
\end{aligned}
$$

3.5.- Mapas de Karnaugh.

Un mapa de Karnaugh (también conocido como tabla de Karnaugh o diagrama de Veitch, abreviado como Mapa-K o Mapa-KV) es un diagrama utilizado para la simplificación de funciones algebraicas Booleanas. El mapa de Karnaugh fue inventado en 1953 por Maurice Karnaugh, un físico y matemático de los laboratorios Bell.

Los mapas de Karnaugh reducen la necesidad de hacer cálculos extensos para la simplificación de expresiones booleanas, aprovechando la capacidad del cerebro humano para el reconocimiento de patrones y otras formas de expresión analítica, permitiendo así identificar y eliminar condiciones muy inmensas.

El mapa de Karnaugh consiste en una representación bidimensional de la tabla de verdad de la función a simplificar. Puesto que la tabla de verdad de una función de N variables posee 2N filas, el mapa K correspondiente debe poseer también 2N cuadrados. Las variables de la expresión son ordenadas en función de su peso y siguiendo el código Gray, de manera que sólo una de las variables varía entre celdas adyacentes. La transferencia de los términos de la tabla de verdad al mapa de Karnaugh se realiza de forma directa, albergando un 0 ó un 1, dependiendo del valor que toma la función en cada fila. Las tablas de Karnaugh se pueden fácilmente realizar a mano con funciones de hasta 6 variables, para funciones de mayor cantidad de variables es más eficiente el uso de software especializado.

3.6.- Método de minimización Tabular de Quine-McCluskey.

El Algoritmo Quine–McCluskey Es un método de simplificación de funciones booleanas desarrollado por Willard Van Orman Quine y Edward J. McCluskey. Es funcionalmente idéntico a la utilización del mapa de Karnaugh, pero su forma tabular lo hace más eficiente para su implementación en lenguajes computacionales, y provee un método determinista de conseguir la mínima expresión de una función booleana.

Pasos

El método consta de dos pasos:

Encontrar todos los implicantes primos de la función.

Usar esos implicantes en una tabla de implicantes primos para encontrar los implicantes primos esenciales, los cuales son necesarios y suficientes para generar la función.

Complejidad

Aunque es más práctico que el mapa de Karnaugh, cuando se trata de trabajar con más de cuatro variables, el tiempo de resolución del algoritmo Quine-McCluskey crece de forma exponencial con el aumento del número de variables. Se puede demostrar que para una función de n variables el límite superior del número de implicantes primos es 3n/n. Si n = 32 habrá más de 6.5 * 1015 implicantes primos. Funciones con un número grande de variables tienen que ser minimizadas con otros métodos heurísticos.

3.7.- Lógica combinatoria.

Es una de las direcciones de la lógica matemática; se ocupa del análisis de los conceptos que, en el marco de la lógica matemática clásica, se tomen sin ulterior estudio. Al número de tales conceptos pertenecen los de variable, función, regla de la sustitución, &c. En la lógica matemática clásica, se utilizan reglas de dos clases. Las primeras se enuncian con sencillez y se aplican sin limitaciones de ninguna clase. Tal es, por ejemplo, la regla del modus ponens. Se formula del modo siguiente: si se han inferido las proposiciones "Si A, entonces 'B' y 'A'", entonces se infiere la proposición 'B'. Esta regla se aplica al cumplimiento automático de un solo acto. Las reglas de la segunda clase (por ejemplo, la regla de la sustitución) se formulan de manera muy compleja e implican varias limitaciones y salvedades (sin éstas, aquéllas no pueden utilizarse de manera puramente formal). Uno de los objetivos de la lógica combinatoria estriba en crear sistemas formales en los que no se encuentren reglas análogas a la regla de la sustitución. Los comienzos de la lógica combinatoria se deben a los trabajos del matemático soviético M. I. Scheinfinkel (sus principales resultados se publicaron en 1924). Independientemente de él, Alonzo Church estructuró el cálculo de la conversión lambda relacionándolo estrechamente con la lógica combinatoria. Importantes resultados han obtenido también el lógico americano H. B. Curry. Investigan los problemas de la lógica combinatoria, Bertrand Russell, W. Craig y Robert Feys entre otros.

La lógica combinatoria es la lógica última y como tal puede ser un modelo simplificado del cómputo, usado en la teoría de la computabilidad (el estudio de qué puede ser computado) y la teoría de la prueba (el estudio de qué se puede probar matemáticamente).

La teoría, a causa de su simplicidad, captura las características esenciales de la naturaleza del cómputo. La lógica combinatoria (LC) es el fundamento del cálculo lambda, al eliminar el último tipo de variable de éste: la variable lambda. En LC las expresiones lambda (usadas para permitir la abstracción funcional) son substituidas por un sistema limitado de combinadores, las funciones primitivas que no contienen ninguna variable libre (ni ligada). Es fácil transformar expresiones lambda en expresiones combinatorias, y puesto que la reducción de un combinador es más simple que la reducción lambda, LC se ha utilizado como la base para la puesta en práctica de algunos lenguajes de programación funcionales no-estrictos en software y hardware.

Sumario del cálculo lambda

El cálculo lambda se refiere a objetos llamados lambda-términos, que son cadenas de símbolos de una de las formas siguientes:

- *v*
- *λv.E1*
- *(E1 E2)*

donde v es un nombre de variable tomado de un conjunto infinito predefinido de nombres de variables, y E1 y E2 son lambda-términos. Los términos de la forma λv.E1 son llamadas abstracciones. La variable v se llama el parámetro formal de la abstracción, y E1 es el cuerpo de la abstracción.

El término λv. E1 representa la función que, si es aplicada a un argumento, liga el parámetro formal v al argumento y entonces computa el valor resultante de E1--- esto es, retorna E1, con cada ocurrencia de v substituido por el argumento.

Los términos de la forma (E1 E2) son llamados aplicaciones. Las aplicaciones modelan la invocación o ejecución de una función: La función representada por E1 es invocada, con E2 como su argumento, y se computa el resultado. Si E1 (a veces llamado el aplicando) es una abstracción, el término puede ser reducido: E2, el argumento, se puede substituir en el cuerpo de E1 en lugar del parámetro formal de E1, y el resultado es un nuevo término lambda que es equivalente al antiguo. Si un término lambda no contiene ningún subtérmino de la forma (λv.E1 E2) entonces no puede ser reducido, y se dice que está en forma normal.

3.7.1.- Decodificadores y Codificadores.

Codificadores

Un codificador es un circuito combinacional con un conjunto de entradas (2N) y un número de salidas N cuyo propósito es mostrar en la salida el código binario correspondiente a la entrada activada.

Por ejemplo, un codificador de 4 entradas X0, X1, X2, X3 y 2 salidas S0, S1. Si se activa la entrada X0 mediante la introducción de un 1, el código mostrado a la salida será S0S1=00. Y así para el resto de las entradas: X1 activará una salida 01, X2 activará una salida 10 y X3 activará una salida 11. Obsérvese que el valor en binario de la salida en su conjunto 00, 01,10, 11 es igual al número decimal de la entrada activada 0,1,2,3 que acompaña a la letra 'X'.

Las funciones algebraicas de un codificador se pueden deducir a partir de su funcionamiento. Por ejemplo, en el caso de que se disponga de un codificador de 4 entradas (y dos salidas) éstas serán:

S0 = X1 + X3

S1 = X0 + X2

Para ello, se ha tenido en cuenta que la salida S0 sólo vale 1 para los valores 1 y 3 (en decimal) o 01 y 11 (en binario). La salida S1 sólo vale 1 para los valores 2 y 3 (en decimal) o 10 y 11 (en binario).

Para obtener estas funciones se ha considerado que nunca va a producirse una combinación a la entrada que tenga más de un 1 y por lo tanto, no importa el valor que produce a la salida esa situación. Todo ello lo podríamos resumir en la siguiente tabla de verdad resumida en la que sólo se han puesto 4 combinaciones de las 24=16 posibles. Las celdas sombreadas en azul son las salidas.

X3	X2	X1	X0	S1	S0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Si en algún momento se activarán más de una entrada, la salida no estaría definida, es decir, no se puede saber a priori qué valor se obtendría. Dependiendo de cómo se haya diseñado puede haber variaciones entre unos dispositivos y otros.

Codificador con prioridad

En la explicación anterior se ha supuesto que únicamente una de las entradas X0, X1, X2, X3 puede estar activa (con un 1) a la vez. Esto es cierto para algunas aplicaciones. Sin embargo, hay otras aplicaciones para las cuales es posible que estén activadas (con un 1) más de una entrada a la vez.

En este caso el circuito debe estar diseñado para establecer una prioridad o precedencia entre las entradas para determinar en cada caso cual es la que realmente se indica a la salida.

Podemos escribir la siguiente tabla de verdad en la que hacemos uso de condiciones don't care (marcadas con 'x'). Las celdas sombreadas en azul son las salidas.

X3	X2	X1	X0	S1	S0
0	0	0	1	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1

La interpretación de esta tabla es la siguiente: si la entrada X3 está a 1 no importa cuál sea el valor del resto de entradas ya que el código 11 a la salida (el correspondiente a 3, valor decimal de X3). Esto significa que es la más prioritaria frente a las demás.

Para que se active la salida 10 (correspondiente a 2, valor decimal de X2), es necesario que X2=1 pero también que X3=0 (o sea, que no esté activada). Dado que X2 es prioritaria frente a X1 y X0, se pone un valor 'x' en esas entradas.

Finalmente, se puede deducir que la X0 es la menos prioritaria de todas porque para que se active su código a la salida, es necesario, además de que esté a 1 (o sea, activada), que todas las demás estén a 0 (o sea, desactivadas).

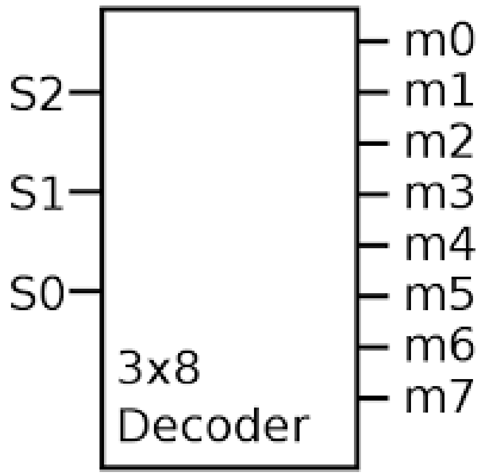
En este caso, las funciones resultantes, obtenidas mediante la aplicación de mapas de Karnaugh, serán:

$S0 = X1 \cdot X2' + X3$

$S1 = X2 + X3$

Decodificadores

Los decodificadores efectúan la operación inversa de los codificadores. Disponen de un conjunto N de entradas y un conjunto 2N de salidas. Cuando aparece un código binario a la entrada, se activa (tiene un 1) la salida identificada con el número decimal equivalente.



En el siguiente ejemplo, se plantea un codificador de 2 a 4, que tiene la siguiente tabla de verdad (las celdas sombreadas en azul son las salidas):

E1	E0	Z3	Z2	Z1	Z0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

En esta tabla se disponen dos entradas E1, E0 y 4 salidas Z3, Z2, Z1 y Z0. Se activará un 1 en la salida correspondiente al código introducido en la entrada.

Las funciones de salida son bastante sencillas de obtener a partir de esta tabla de verdad:

$Z3 = E1 \cdot E0$

$Z2 = E1 \cdot E0'$

$Z1 = E1' \cdot E0$

$Z0 = E1' \cdot E0'$

Se puede considerar que la función de estos dispositivos es la de generar los 2N minitérminos de las "N" variables de entrada. Esta visión es muy interesante porque pueden ser utilizados para la implementación de cualquier función algebraica del mismo número de variables.

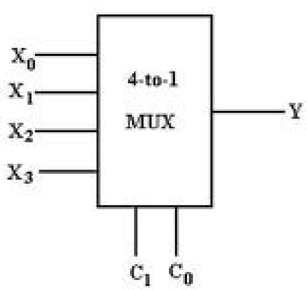
3.7.2.- Multiplexores y Demultiplexores.

Multiplexores y demultiplexores

En el diseño de circuitos digitales es habitual encontrarse de forma reiterada con las mismas necesidades que requieren las mismas soluciones. Esto da lugar a un conjunto de subcircuitos que aparecen con mucha asiduidad a la hora de proporcionar una solución a un problema. Estos elementos se estudian en profundidad ya que por una parte ahorran tiempo de diseño (no debe "reinventarse la rueda") y por otra parte están disponibles con circuitos integrados comerciales o en forma de bloques en los programas de diseño electrónico.

Multiplexores

Este el primero de estos elementos o subcircuitos comunes a muchos diseños. Un multiplexor es un circuito combinacional de varias entradas de señal y una única salida. Dispone además de unas entradas de selección que permiten redirigir cualquiera de las entradas elegidas a la salida. En sentido estricto, las entradas de selección se deben considerar exactamente igual que las otras entradas de señal, pero debido a su particular propósito ("selección") suelen dibujarse separadas. En la figura siguiente se muestra un multiplexor de 4 a 1:



Multiplexor 4 a 1

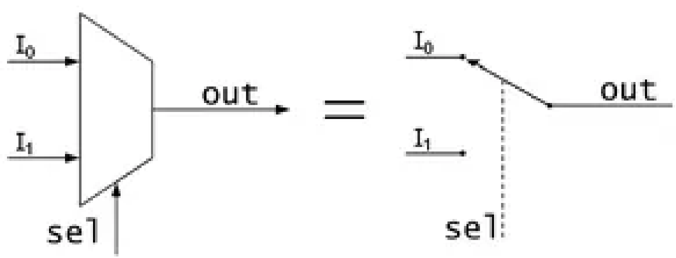
Se suele indicar 4 a 1 para proporcionar el número de entradas de señal que puede manejar (X0, X1, X2 y X3). Como se ha comentado antes, las entradas de selección (C1 y C0) aunque son propiamente entradas del circuito, se dibujan en un lateral (en la parte inferior en este caso) con el objeto de clarificar su función.

Por otra parte, se puede observar que una vez fijado el número de entradas, queda unívocamente fijado el número de entradas de selección ya que si dispone de N entradas de selección se podrán seleccionar 2^N entradas de señal. La función lógica de este circuito es muy sencilla de obtener (incluyendo únicamente los términos que dan lugar a 1) y es la siguiente :

$$Y = X0 \cdot C1' \cdot C0' + X1 \cdot C1' \cdot C0 + X2 \cdot C1 \cdot C0' + X3 \cdot C1 \cdot C0$$

Por ejemplo, el primer término refleja que la salida será 1 si X0 = 1 y la combinación C1C0 = 00, que corresponde a la selección de la entrada X0. y así para los otros 3 términos de la suma de productos.

Utilizando este método, se puede obtener la función algebraica de cualquier multiplexor. En la figura siguiente se muestra el símbolo que normalmente se utiliza para representar multiplexores, en este caso, particularizado para un multiplexor 2 a 1.



Multiplexor 2 a 1 y circuito eléctrico equivalente
(Extraído de la Wikipedia)

En la parte derecha de la figura se observa el equivalente eléctrico del multiplexor. Se puede observar que tiene forma de trapecio girado 90°. El circuito de selección actúa como un conmutador que conecta la entrada requerida a la salida para que pueda fluir la señal.

Demultiplexores

Los demultiplexores efectúan la operación inversa de los multiplexores y se utilizan para distribuir una señal que llega por una única entrada a un conjunto de salidas. De la misma forma, las entradas de selección permiten elegir a cuál de las salidas se redirige la señal de entrada.

En este caso, se cumple igualmente que N entradas de selección va a permitir distribuir la señal en 2^N salidas. Si consideramos una única entrada X y un conjunto de salidas Y0, Y1, Y2, Y3, ya se puede deducir que tendrá dos entradas de selección,por ejemplo, S0 y S1.

Las funciones de cada una de las salidas se pueden expresar de forma independiente y son las siguientes:

Y0 = X ·S1' ·S0′

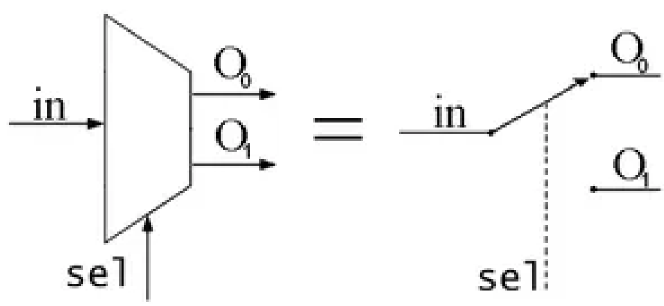
Y1 = X ·S1' ·S0

Y2 = X ·S1 ·S0′

Y3 = X ·S1 ·S0

En la que por ejemplo, la salida Y0 sólo tiene un 1 cuando la entrada X=1 y la combinación S1S0 = 00 indica que es la salida seleccionada. Al igual que en el caso anterior, el patrón de cada función algebraica permite desarrollar las funciones de salida para cualquier demultiplexor.

El símbolo genérico de este elemento digital se muestra en la siguiente figura particularizado para un demultiplexor 1 a 2. Se puede observar que tiene también la forma de un trapecio (girado 90º).



En la parte derecha se muestra el equivalente eléctrico del demultiplexor. Funciona como un conmutador que permite conectar la entrada con la salida deseada para que fluya la señal.

3.7.3.- Comparadores.

Los comparadores son circuitos combinacionales capaces de comparar dos combinaciones presentes en sus entradas indicando si son iguales o diferentes; en caso de ser diferentes, indican cuál de las dos es mayor. Tienen tres salidas que indican el resultado de la comparación: A=B, A<B y A>B.

El procedimiento para comparar dos datos binarios consiste primero en comparar el bit más significativo de cada uno de ellos, si éstos son iguales, se compara el siguiente bit más significativo y así sucesivamente hasta encontrar una desigualdad que indica cuál de los datos es mayor o menor. Si se comparan todos los bits de ambos datos y no hay desigualdad entre ellos, entonces evidentemente son iguales.

Los circuitos integrados más utilizados son:

CIRCUITO INTEGRADO TTL 7485 COMPARADOR DE 4 BITS.
Este circuito integrado contiene un comparador de dos datos de 4 bits cada uno.

La relación de pines de este integrado es la siguiente:

A>B, A<B, y A=B: entradas de comparación en cascada activas a nivel alto (1)

A>B, A<B, y A=B: salidas de comparación activas a nivel alto (1).

A0, A1, A2, A3: entradas del dato A.

B0, B1, B2, B3: entradas del dato B.

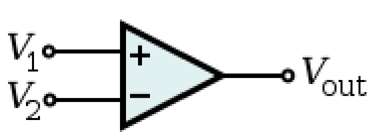
Los comparadores, son circuitos no lineales que, sirven para:

Comparar dos señales (una de las cuales generalmente es una tensión de referencia)

Determinar cuál de ellas es mayor o menor.

En un circuito electrónico, se llama comparador a un amplificador operacional en lazo abierto (sin realimentación entre su salida y su entrada) y suele usarse para comparar una tensión variable con otra tensión fija que se utiliza como referencia.

Funcionamiento del comparador



En este circuito, se alimenta el amplificador operacional con dos tensiones +Vcc = 15V y -Vcc = -15 V. Se conecta la patilla V+ del amplificador a masa (tierra) para que sirva como tensión de referencia, en este caso 0 V. A la entrada V- del amplificador se conecta una fuente de tensión (Vi) variable en el tiempo, en este caso es una tensión sinusoidal.

Hay que hacer notar que la tensión de referencia no tiene por qué estar en la entrada V+, también puede conectarse a la patilla V-, en este caso, se conectaría la tensión que queremos comparar con respecto a la tensión de referencia, a la entrada V+ del amplificador operacional.

A la salida (Vo) del amplificador operacional puede haber únicamente dos niveles de tensión que son en este caso 15 o -15 V (considerando el AO como ideal, si fuese real las tensiones de salida serían algo menores).

Cuando la tensión sinusoidal Vi toma valores positivos, el amplificador operacional se satura a negativo; esto significa que como la tensión es mayor en la entrada V- que en la entrada V+, el amplificador entrega a su salida una tensión negativa de -15 V.

UNIDAD IV

DISPOSITIVOS SECUENCIALES

4.1.- Flip Flops.

Los circuitos secuenciales son aquellos en los cuales su salida depende de la entrada presente y pasada. Dentro de estos circuitos se tienen a los Flip-Flops.

Los Flip-Flops son los dispositivos con memoria más comúnmente utilizados. Sus características principales son:

Asumen solamente uno de dos posibles estados de salida.

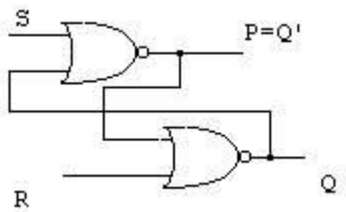
Tienen un par de salidas que son complemento una de la otra.

Tienen una o más entradas que pueden causar que el estado del Flip-Flop cambie.

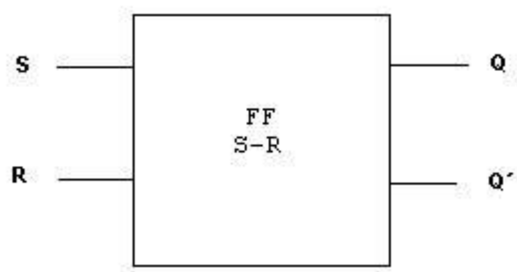
A continuación, se describirán 4 tipos de Flip-Flops.

Flip-Flop S-R (Set-Reset)

La siguiente figura muestra una forma posible de implementar un Flip-Flop S-R. Utiliza dos compuertas NOR. S y R son las entradas, mientras que Q y Q' son las salidas (Q es generalmente la salida que se busca manipular.)



Como existen varias formas de implementar un Flip-Flop S-R (y en general cualquier tipo de Flip-Flop) se utilizan diagramas de bloque que representen al Flip-Flop. El siguiente diagrama de bloque representa un FF S-R. Nótese que ahora, por convención, Q se encuentra en la parte superior y Q' en la inferior.



Para describir el funcionamiento de un FF se utilizan las llamadas Tablas de Estado y las Ecuaciones Características. La siguiente tabla muestra la tabla de estado para un FF S-R.

S	R	Q	Q+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

Como encabezado de las columnas tenemos las entradas S y R, y una de las salidas Q. La salida Q es la salida que en un tiempo t se puede detectar en el FF, es decir, es la salida en el tiempo actual. Q+ es la salida en el tiempo, una vez que se ha propagado la señal en el circuito (recuerde que los FF tienen un componente de retroalimentación.) Por lo tanto, es decir, es la salida que tendrá Q en el futuro – una vez que se haya realizado la propagación.

Si analizamos la tabla de estado, vemos que para si S = 0, R = 0 y Q = 0 ó 1, la salida futura de Q (Q+) será siempre lo que se tenía antes de la propagación. A este estado (S = 0, R = 0) se le conoce por tanto como estado de memoria.

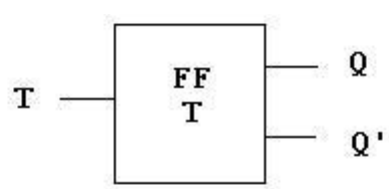
Viendo ahora el caso S = 0, R = 1, se aprecia que siempre Q+ = 0 sin importar el valor de Q antes de la propagación, es decir, se hace un reset de Q. Si, por el contrario, se tiene S = 1, R = 0, entonces Q+ = 1 en ambos casos, por tanto, se hace un set de Q.

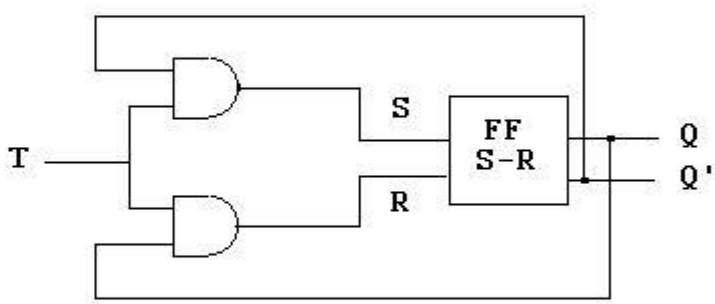
Finalmente, nótese que la combinación S = 1, R = 1 no es válida en el FF S-R. La razón es que dicho estado vuelve inestable al circuito y, como una de las características de todo FF es que el estado es estable, al usar dicha combinación se está violando este principio de los FF.

Ahora, si se mapea la información de la tabla de estado del FF S-R en un mapa de Karnaugh, se obtiene la siguiente ecuación característica: Esta ecuación describe también el funcionamiento. Nos dice que Q+ será 1 siempre y cuando se haga un set del FF o el reset no está activado y la salida tiene un 1 en ese momento.

Flip-Flop T

El Flip-flop T cambia de estado en cada pulso de T. El pulso es un ciclo completo de cero a 1. Las siguientes dos figuras muestran el diagrama de bloque y una implementación del FF T mediante un FF S-R y compuertas adicionales.





Nótese que en la implementación del FF T, las dos entradas del FF S-R están conectadas a compuertas AND, ambas conectadas a su vez a la entrada T. Además, la entrada Q esta conectada a R y Q’ a S. Esta conexión es así para permitir que el FF S-R cambié de estado cada que se le mande un dato a T. Por ejemplo, si Q = 1 en el tiempo actual, eso significa que Q’ = 0, por lo tanto, al recibir T el valor de 1, se pasaran los valores de R = 1 y S = 0 al FF S-R, realizando un reset de Q.

La siguiente tabla muestra el comportamiento del FF T y del FF S-R en cada pulso de T

T		S	R		Q	Q´
0		0	0		0	1
1		1	0		1	0
0		0	0		1	0
1		0	1		0	1
0		0	0		0	1
1		1	0		1	0

La tabla de estado para el FF T se presenta a continuación. Es muy sencilla: cuando T = 0 el estado de Q no cambia, es decir Q = Q+ (estado de memoria), cuando T = 1, Q es complementada y, por lo tanto, Q+ = Q’.

Tabla de estado para el FF T

T	Q	Q+
0	0	0
0	1	1
1	0	1
1	1	0

De la tabla de estado anterior, se obtiene la siguiente ecuación característica para el FF T

Q+ = T ’Q + TQ´ = T Å Q

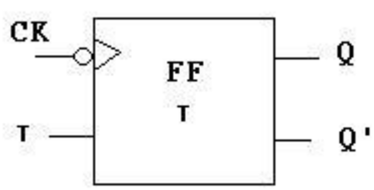
Ahora bien, analicemos un poco más el comportamiento del FF T y tratemos de responder la siguiente pregunta: ¿Qué pasa si T=1 por mucho tiempo?

Los valores de S y R cambiarían constantemente de la siguiente manera:

S = 0-> 1 -> 0 -> 1

R= 1-> 0 ->1 -> 0

es decir, el FF empezaría a oscilar y por tanto no mantendría el estado (inestable.) Por lo tanto, la mayoría de los FF utilizan un reloj para determinar en que momento se tomará en cuenta el valor que se encuentre en la entrada del FF. La siguiente figura muestra un FF T con reloj (CK)



Nótese que la entrada marcada como CK tiene un círculo. Este círculo indica que el FF tomará en cuenta la entrada del FF cuando el pulso del reloj sea cero (0). Si es uno (1), la entrada no será tomada en cuenta.

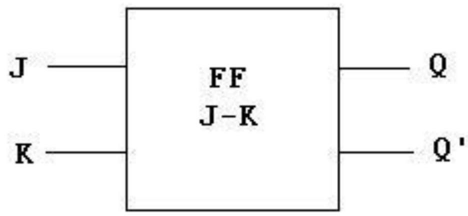
Flip-Flop J-K

El flip-flop J-K es una mezcla entre el flip-flop S-R y el flip-flop T. Esto ocurre de la siguiente manera:

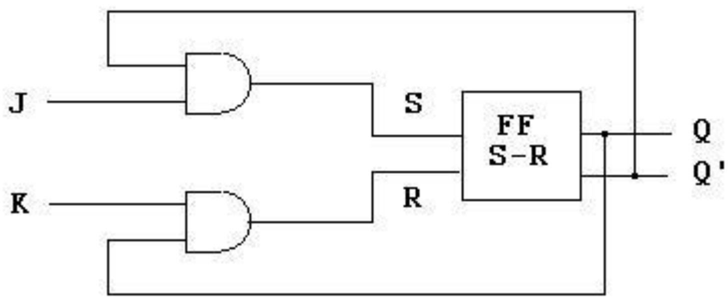
En J=1, K=1 actúa como Flip-flop T

De otra forma, actúa como flip-flop S-R

El siguiente diagrama de bloque es el perteneciente el FF J-K



Una implementación tentativa de un FF J-K a partir de un FF S-R sin reloj es la siguiente:

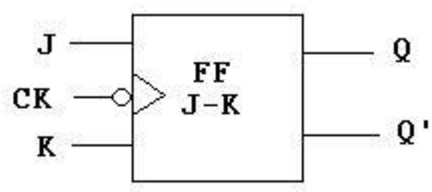


La tabla de estado aparece a continuación. Note que es muy parecida a la del FF S-R solo que ahora los estados de J=1 y K=1 sí son válidos.

Tabla de estado del FF J-K

J	K	Q	Q+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

De la tabla anterior se obtiene la siguiente ecuación característica mediante mapas de Karnaugh: Este flip-flop es uno de los más comunes con reloj. El siguiente diagrama lo muestra con entrada para reloj:

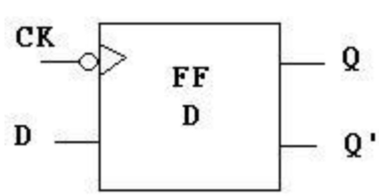


Flip-Flop D (Delay)

El flip-flop D es uno de los FF más sencillos. Su función es dejar pasar lo que entra por D, a la salida Q, después de un pulso del reloj. Es, junto con el FF J-K, uno de los flip-flops mas comunes con reloj. Su tabla de estado se muestra a continuación:

D	Q	Q+
0	0	0
0	1	0
1	0	1
1	1	1

De la tabla se infiere que la ecuación característica para el FF D es: Q+= D. El siguiente diagrama de bloques representa este flip-flop.



Inicialización de Flip-Flops

 Cuando se están utilizando flip-flops en la construcción de circuitos, es necesario poder controlar el momento en el que un FF empieza a funcionar y el valor con el que inicia su secuencia. Para esto, los flip-flops cuentan con dos entradas que le permiten al diseñador seleccionar los valores iniciales del FF y el momento en el que empieza a funcionar. Estas entradas son llamadas en Inglés: Clear y Preset.

Clear - inicializa Q en cero sin importar entradas o reloj ( ).

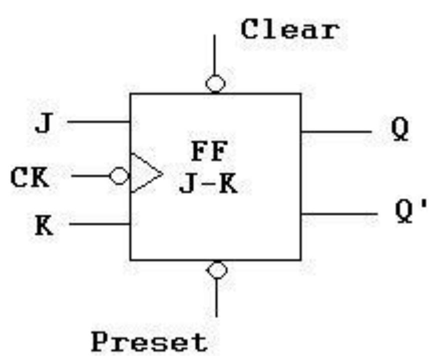
Preset - inicializa Q en 1 sin importar entradas o reloj ( ).

Para ambas entradas, si reciben el valor de:

0 : inicializan el FF en el valor correspondiente.

1: el flip-flop opera normalmente

La siguiente figura muestra un FF J-K con entradas de inicialización. Note que tanto la entrada Clear, como la entrada Preset, tienen un círculo. Esto significa que la entrada funciona con un 0.



4.2.- Registros de corrimiento.

Un registro de corrimiento es un circuito secuencial síncrono capaz de contractar varios bits de información. El formato de esta información puede ser de dos tipos:

Serie: los bits se transfieren uno a continuación del otro por una misma línea.

Paralelo: se intercambian todos los bits al mismo tiempo, utilizando un número de líneas de transferencia igual al número de bits.

Contadores de registro de corrimiento

En los contadores de registro de corrimiento se utiliza retroalimentación, lo cual significa que la salida del último flip-flop del registro se conecta en retroceso con el primer flip-flop en alguna forma.

Contador de anillo

El contador de registro de corrimiento mas simple es un registro de corrimiento circulante conectado que el ultimo ff desplace su valor al primer ff. Los ff se conectan de modo que la información se corra de izquierda a derecha de Q-0 a Q-3. En muchos casos solo hay un 1 el registro y se hace que circule alrededor del registro en tanto se apliquen los pulsos del reloj. Por esta razón se le denomina contador de anillo.

4.3.- Contadores.

Contador electrónico. Un contador es un circuito secuencial construido a partir de biestables y puertas lógicas (flip-flops) conectados en cascada, cuyo número varia en dependencia de la escala de conteo que se necesita. Cuando un contador tiene n flip-flops y avanza por todos los estados posibles antes de regresar a su estado inicial, su módulo es 2 n y decimos que su base es 2n o que es un contador de n bits.

Contadores sincrónicos

Todos los flip-flops cambian simultáneamente con cada pulso del reloj (de acuerdo con el estado de sus entradas de control).

Contadores asincrónicos

Todos los flip-flops no cambian simultáneamente con cada pulso del reloj.

Contadores asincrónicos cuya base no es potencia de dos

Este tipo de contador puede ser construido realimentando convenientemente las salidas a algunas de las entradas, incluyendo las entradas directas, para eliminar estados de un contador 2n superior.

Conteo Programable

En algunas aplicaciones es importante poder programar diferentes bases de conteo en un mismo contador por medio de conmutadores o de datos en las entradas de preset.

Tiempo de acarreo en contadores

El acarreo en un contador es el tiempo requerido por el mismo para complementar la respuesta a un pulso de entrada. El tiempo de acarreo para un contador, es el tiempo máximo que toma la respuesta del mismo al pulso de entrada.

Contadores monolíticos

Contadores construidos a base de integrados con distintas bases de conteo para ser usados en los sistemas digitales, por ser más confiables, mas económicos y mas pequeños. La familia TTL es la más utilizada.

Aplicaciones

Relojes y temporizadores

Divisores de frecuencia

Frecuencímetros.

4.4.- Modelos de circuitos secuenciales sincronos.

Los circuitos considerados hasta aquí, tienen la característica de que su salida depende solamente de la combinación presente de valores de las entradas, es decir, a una misma combinación de entrada responden siempre con la misma salida. Debido a esto, estos circuitos se denominan combinacionales.

Los circuitos combinacionales tienen muchas limitantes debido a que no son capaces de reconocer el orden en que se van presentando las combinaciones de entradas con respecto al tiempo, es decir, no pueden reconocer una secuencia de combinaciones, ya que no poseen una manera de almacenar información pasada, es decir no poseen memoria.

Un circuito cuya salida depende no solo de la combinación de entrada, sino también de la historia de las entradas anteriores se denomina Circuito Secuencial. La historia de las entradas anteriores en un momento dado se encuentra resumida en el estado del circuito, el cual se expresa en un conjunto de variables de estado.

El circuito secuencial debe ser capaz de mantener su estado durante algún tiempo, para ello se hace necesario el uso de dispositivos de memoria. Los dispositivos de memoria utilizados en circuitos secuenciales pueden ser tan sencillos como un simple retardador (inclusive, se puede usar el retardo natural asociado a las compuertas lógicas) o tan complejos como un circuito completo de memoria denominado multivibrador biestable o Flip Flop.

Circuito secuencial asíncrono

En un circuito secuencial asíncrono, los cambios de estado ocurren al ritmo natural marcado por los retardos asociados a las compuertas lógicas utilizadas en su implementación, es decir, estos circuitos no usan elementos especiales de memoria, pues se sirven de los retardos propios (tiempos de propagación) de las compuertas lógicas usados en ellos. Esta manera de operar puede ocasionar algunos problemas de funcionamiento, ya que estos retardos naturales no están bajo el control del diseñador y además no son idénticos en cada compuerta lógica.

Circuito secuencial síncrono

Los circuitos secuenciales síncronos, sólo permiten un cambio de estado en los instantes marcados por una señal de sincronismo de tipo oscilatorio denominada reloj. Con ésto se pueden evitar los problemas que tienen los circuitos asíncronos originados por cambios de estado no uniformes en todo el circuito.

4.5.- Análisis y síntesis de un circuito secuencial síncrono.

A diferencia de los sistemas combinacionales, en los sistemas secuenciales, los valores de las salidas, en un momento dado, no dependen exclusivamente de los valores de las entradas en dicho momento, sino también dependen del estado anterior o estado interno.1 El sistema secuencial más simple es el biestable, de los cuales, el de tipo D (o cerrojo) es el más utilizado actualmente.

El sistema secuencial requiere de la utilización de un dispositivo de memoria que pueda almacenar la historia pasada de sus entradas (denominadas variables de estado) y le permita mantener su estado durante algún tiempo, estos dispositivos de memoria pueden ser sencillos como un simple retardador o celdas de memoria de tipo DRAM, SRAM2 o multivibradores biestables también conocido como Flip-Flop1 entre otros.

4.6.- Tipos de circuitos asíncronos.

Circuitos asíncronos

Un circuito asíncrono es el que registra el orden en el que cambian sus variables de entrada, y envía una salida que depende del resultado. Este tipo de circuito también debe ser capaz de cambiar sus variables de entrada en cualquier momento. Hay también un tipo específico de circuito asíncrono, denominado circuito asíncrono tipo compuerta. Los circuitos tipo compuerta son circuitos combinacionales esencialmente (es decir, que se basan únicamente en la entrada actual) con un camino de realimentación. El camino de realimentación significa que la información de la salida puede ser realimentada en la entrada. Debido a la retroalimentación, estos tipos de circuitos pueden ser inestables, por lo que no se usan comúnmente.

Un sistema secuencial dispone de elementos de memoria cuyo contenido puede cambiar a lo largo del tiempo.

El estado de un sistema secuencial viene dado por el contenido de sus elementos de memoria. Es frecuente que en los sistemas secuenciales exista una señal que inicia los elementos de memoria con un valor determinado: señal de inicio (reset). La señal de inicio determina el estado del sistema en el momento del arranque (normalmente pone toda la memoria a cero).

La salida en un instante concreto viene dada por la entrada y por el estado anterior del sistema. El estado actual del sistema, junto con la entrada, determinará el estado en el instante siguiente realimentación.

4.7.- Aplicaciones.

El circuito secuencial debe ser capaz de mantener su estado durante algún tiempo, para ello se hace necesario el uso de dispositivos de memoria. Los dispositivos de memoria utilizados en circuitos secuenciales pueden ser tan sencillos como un simple retardador (inclusive, se puede usar el retardo natural asociado a las compuertas lógicas) o tan complejos como un circuito completo de memoria denominado multivibrador biestable o Flip Flop.

La salida del elemento de retraso es una copia de la señal de entrada retraso un determinado tiempo; mientras que la salida del elemento de memoria copia los valores de la entrada cuando la señal de control tiene una transición de subida, por lo que la copia no es exacta, sino que sólo copia lo que interesa. Por lo tanto, el modelo clásico de un sistema secuencial consta de un bloque combinacional, que generará la función lógica que queramos realizar, y un grupo de elementos de memoria con una serie de señales realimentadas.

Como ya hemos comentado, los sistemas secuenciales forman un conjunto de circuitos muy importantes en la vida cotidiana. En cualquier elemento que sea necesario almacenar algún parámetro, es necesario un sistema secuencial. Así, cualquier elemento de programación (o lo que es lo mismo, con más de una función) necesita un sistema secuencial.

Bibliografía

Laboratorio de circuitos electrónicos II	Nicolás Reyes Ayala, Raymundo Barrales Guadarrama y Ernesto Rodríguez Vázquez Cerón	Ediciones de la U - 9789587626568
ELECTRONICA DE POTENCIA: CIRCUITOS, DISPOSITIVOS Y APLICACIONES 3ª ED.	MUHAMMAD H. RASHID	Prentice Hall Mexico - 9789702605324
CIRCUITOS BASICOS DE CONTADORES Y TEMPORIZADORES.	Vicent Lladonosa	Marcombo