Sistemas digitales

JOSE LUIS RAMOS GONZALEZ

Red Tercer Milenio

SISTEMAS DIGITALES

SISTEMAS DIGITALES

JOSE LUIS RAMOS GONZALEZ

RED TERCER MILENIO

AVISO LEGAL

Viveros de Asís 96, Col. Viveros de la Loma, Tlalnepantla, C.P. 54080, Estado de México.

Datos para catalogación bibliográfica

José Luis Ramos González

Sistemas digitales

ISBN 978-607-733-161-2

Primera edición: 2012

DIRECTORIO

ÍNDICE

INTRODUCCIÓN

El presente libro didáctico tiene como objetivo general, guiar al estudiante en el aprendizaje teórico y práctico del mundo digital y de los circuitos lógicos digitales. Además de servir como guía, el alumno obtendrá los fundamentos teóricos para la construcción de circuitos lógicos operacionales.

Dentro del libro didáctico el alumno tendrá que realizar investigaciones documentales además de efectuar prácticas con circuitos operacionales para establecer su enseñanza–aprendizaje.

Los sistemas digitales se enfocan en la lógica que pueden llegar a tener todos los circuitos que nos encontramos en la vida diaria y, sin saberlo, utilizamos, desde una simple calculadora hasta complicadas computadoras.

Para que el alumno comience el curso de sistemas digitales debe contar con los conocimientos básicos de operación de circuitos, vistos en electrónica básica y teoría matemática. En esas materias se forma la base fundamental de los sistemas digitales desde su concepción lógica hasta su manera de funcionar.

Este libro consta de cinco unidades donde el alumno formará el conocimiento necesario sobre los sistemas digitales, empezando por la parte teórica hasta llegar a la parte práctica, y así aplicarlo en su vida cotidiana.

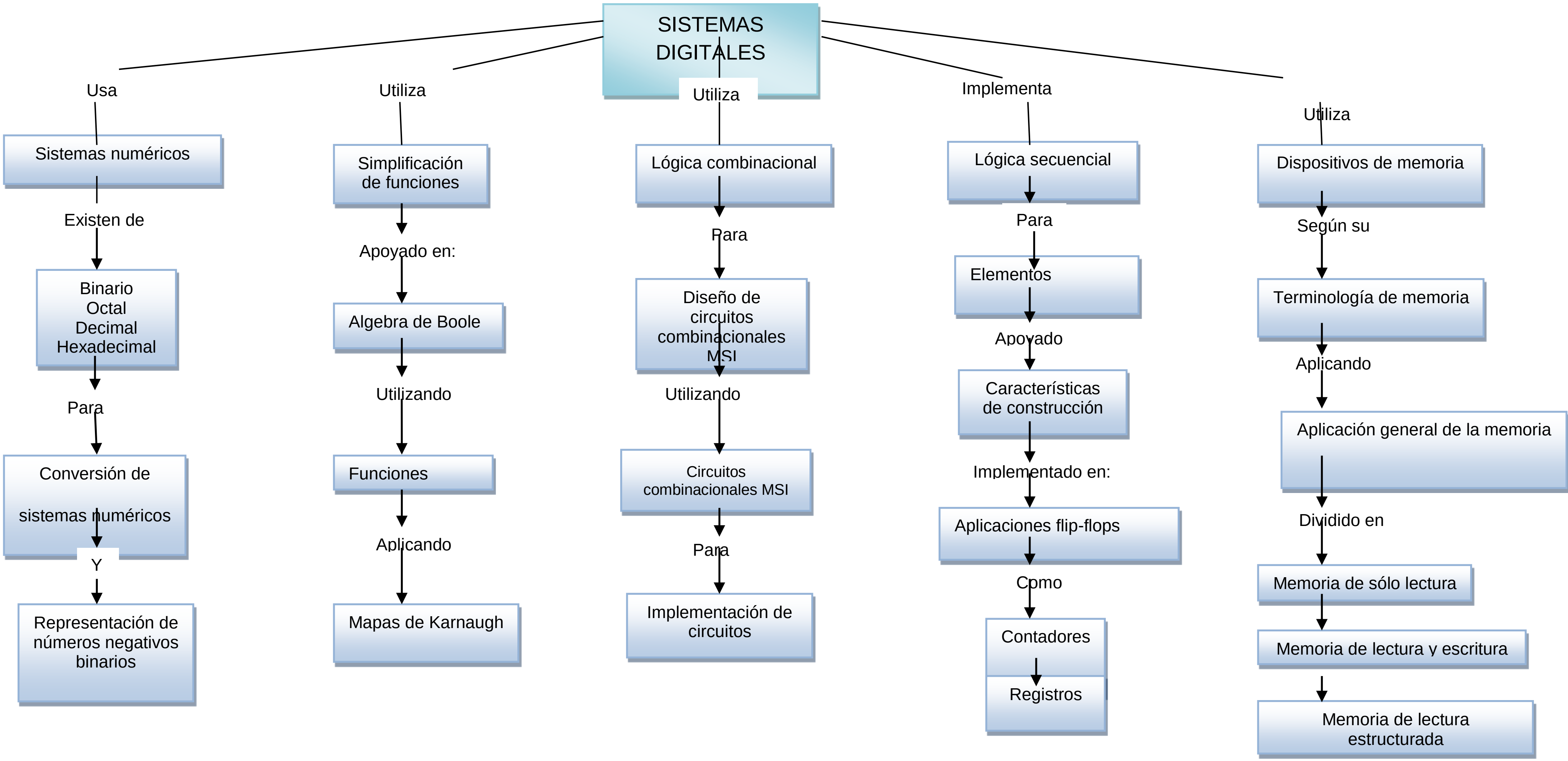
Las primeras dos unidades abarcan el concepto de operación de los sistemas digitales, su operación interna y las partes que tienen que realizar para llegar a obtener un resultado coherente dependiendo de las entradas de los mismos. Ésta es la parte lógica del los sistemas digitales. En resumen, las primeras unidades contienen desde la definición de los sistemas digitales hasta la operación interna lógica que éstos aplican para obtener sus resultados.

Las siguientes dos unidades se enfocan a la construcción interna que tienen los circuitos digitales, su manera de implementación y la lógica que debe tener cada uno de los circuitos dependiendo de las entradas y las salidas que se desean obtener. En esas unidades se incluyen prácticas para fundamentar los conocimientos de los temas abarcados.

La última unidad se refiere al concepto de memoria: amplio y confuso por los diferentes tipos que existen. En esta unidad se presentan los conceptos básicos para su comprensión, además del funcionamiento interno de cada una de las diferentes memorias.

Al finalizar el curso, el alumno tendrá los conocimientos establecidos y estará preparado para su carrera, con la finalidad de instituir juicios basados en el conocimiento adquirido.

MAPA CONCEPTUAL



UNIDAD 1

SISTEMAS NUMÉRICOS

OBJETIVO

Conocer los diferentes sistemas numéricos utilizados en los sistemas informáticos.

TEMARIO

1.1 Sistemas numéricos

1.1.1 Sistema numérico binario

1.1.2 Sistema numérico octal

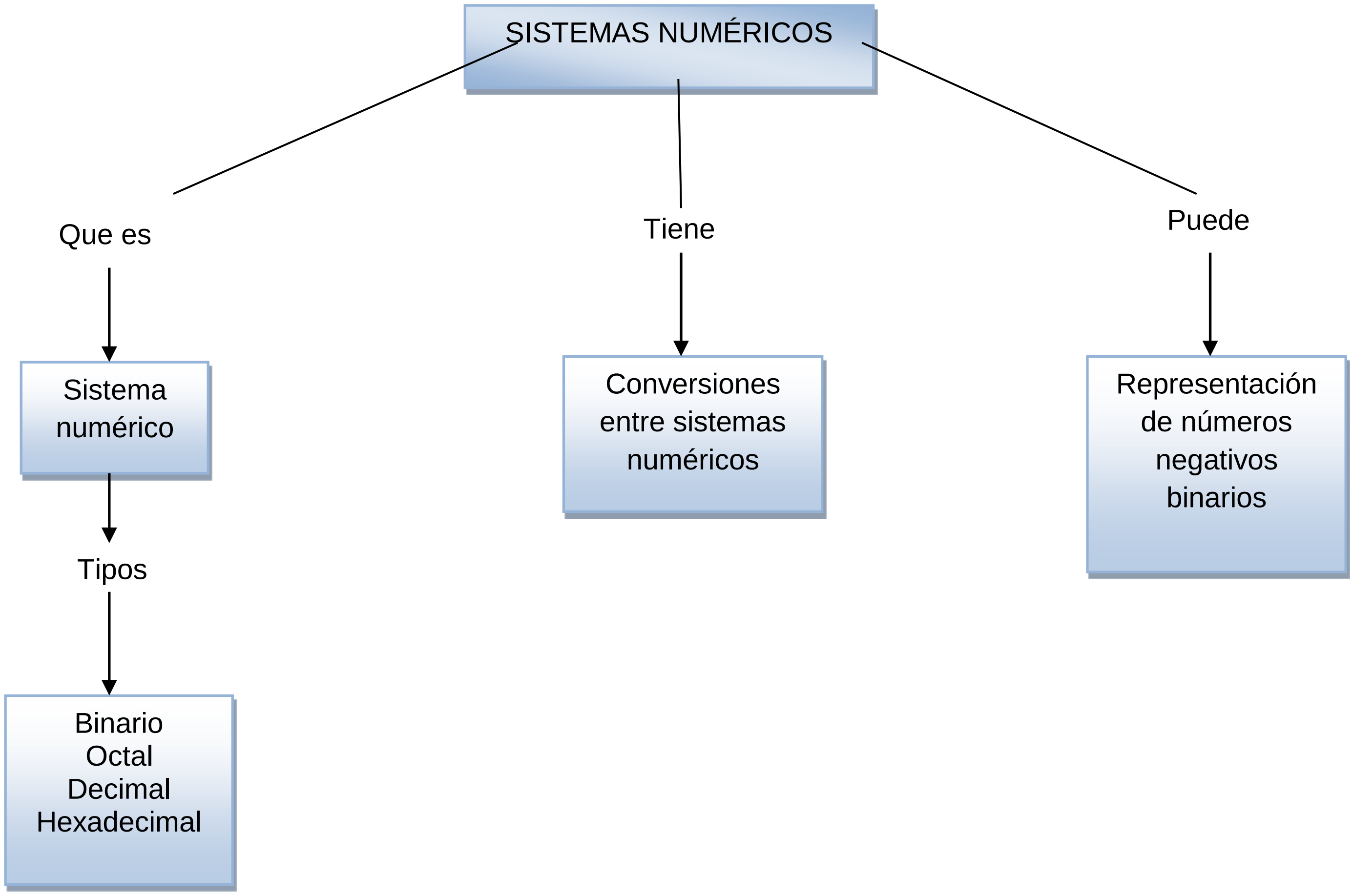
1.1.3 Sistema numérico decimal

1.1.4 Sistema numérico hexadecimal

1.2 Conversión entre sistemas numéricos

1.3 Representación de los números negativos binarios

MAPA CONCEPTUAL



INTRODUCCIÓN

En esta unidad se describe qué es un sistema numérico, los elementos que lo componen, las propiedades de cada uno de ellos, y los alcances y limitaciones que tienen cada uno.

Los sistemas numéricos tienen propiedades, dependiendo cada uno de los elementos a los que se haga referencia así como a las acciones para las cuales se hayan implementado.

En esta unidad el alumno comprenderá cada uno de los sistemas numéricos, sus propiedades y características, además de los cambios entre distintas bases numéricas y su representación negativa.

1.1 SISTEMAS NUMÉRICOS

En la vida cotidiana se manejan diferentes magnitudes que se pueden expresar en *cantidades,* que a su vez, se pueden medir con diferentes instrumentos. En cada uno de ellos se presenta algún tipo de sistema numérico. En la actualidad y con el avance en la tecnología, en la ciencia, los negocios, etcétera, dichas magnitudes se pueden medir básicamente de dos maneras distintas, tanto ANALÓGICAMENTE como DIGITALMENTE.[1]

En las representaciones analógicas, una cantidad se representa mediante un voltaje, una línea de corriente o simplemente a través de un indicador que obtiene su salida mediante una entrada de datos. Un ejemplo muy sencillo de una representación analógica es un termómetro de mercurio que funciona con la temperatura que incide en él, la cual se ve reflejada en la altura que alcanza el mercurio para indicar la temperatura del objeto aproximado al termómetro. Todas las cantidades analógicas tienen una característica peculiar: *pueden variar en un rango o escala continua de valores.*

En las representaciones digitales, contrario a las analógicas, éstas no son representadas en un rango variable sino mediante símbolos llamados dígitos. Un ejemplo muy simple y usado de manera cotidiana es el reloj digital donde la hora se expresa mediante dígitos decimales que representan las horas y minutos.

Una de las características principales de las cantidades analógicas y las digitales es que a menudo, las cantidades tomadas analógicamente están sujetas a interpretación, en comparación con las cantidades tomadas digitalmente donde no hay ambigüedades al momento de tomarlas.

En las diferentes representaciones de magnitudes analógicas o digitales se usan los llamados sistemas numéricos, o sea representaciones numéricas de las magnitudes tomadas; la más utilizada por los seres humanos es la decimal (compuesta por los números naturales 0 al 9), pero las computadoras usan el sistema numérico binario (con los dígitos 0 y 1). Con ellos se desarrollan también los sistemas octal (dígitos del 0 al 7) y el hexadecimal (dígitos el 0 al 9 y de la letra *A* a la *F*).

---

[1] Roland Tocci y Neal Widmer, *Sistemas digitales: principios y aplicaciones,* p. 4

Un sistema numérico se define como el *número máximo de dígitos que el sistema numérico puede soportar o contener*.

Las ventajas que ofrece el sistema digital sobre el analógico son cada vez mayores debido a su facilidad para construirlos e implementarlos. El almacenamiento de información es cada vez más sencillo y una de las ventajas más sobresalientes es que son más precisos que los analógicos.

Entre sus desventajas está que, como el mundo es completamente analógico, todas las magnitudes medidas se controlan mediante cálculos continuos, aproximados.

Debido al gran incremento de los sistemas digitales se cree que el mundo deberá seguir el camino de la tecnología, y que en un futuro los sistemas digitales predominarán: desde las aplicaciones sencillas hasta las muy sofisticadas.

Entre los diferentes tipos de sistemas numéricos se encuentran:

Sistema Numérico	Ejemplo
- **Binario**	1001011_2
- **Octal**	231572_8
- **Decimal**	980753_{10}
- **Hexadecimal**	$4ADF3_{16}$

En cada uno de los sistemas se manejan subíndices que representan el sistema numérico al que pertenecen. En los ejemplos anteriores se manejan los subíndices 2, 8, 10 y 16 que corresponden a los sistemas binarios, octal, decimal y hexadecimal respectivamente.

1.1.1 Sistema numérico binario

En el sistema numérico binario únicamente existen dos símbolos o posibles valores de dígitos: el 0 y el 1. En los sistemas digitales los valores representados también pueden ser dos debido a que es muy complicado manipular varios dígitos ya que la representación sólo puede hacerse mediante estos dos datos. Una de las desventajas que se pudiera presentar en este tipo de sistema es que la representación de los datos en base binaria es muy extensa, pero no imposible. Una de las ventajas que se presenta en este tipo de sistemas es que los datos se procesaron con mayor facilidad y fluidez ya que solamente manejan dos datos. El sistema numérico binario tiene los siguientes dígitos:

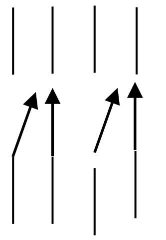
Binario= {0, 1}

Los conteos de números binarios pueden prestarse a confusión debido a que pueden ser muy complejos en su representación dada la problemática planteada. Se presenta la siguiente tabla para comprender el conteo de los números binarios.

2^3=8	**2^2=4**	**2^1=2**	**2^0=1**	**Equivalente Decimal**
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

La representación de los datos en los sistemas binarios sólo puede ser dada por dos estados. Un ejemplo de fácil representación de datos binarios suele ser el interruptor debido a que sólo tiene dos estados: abierto y cerrado.

Ejemplo: 0 1 0 1



En el ejemplo anterior se muestra un interruptor de cuatro estados representados por el dígito 0101_2, dependiente del estado del interruptor. El dígito 0 indica cuando está abierto y el dígito 1 indica cuando está cerrado.

ACTIVIDADES DE APRENDIZAJE

1. Realizar ejercicios de sistema numérico binario.

1.1.2 Sistema numérico octal

En el sistema numérico octal existen, como su nombre lo dice, únicamente 8 dígitos que se representan del 0 al 7 (en total son 8 dígitos tomando el 0 en cuenta como dígito). Una de las ventajas que se presenta en este sistema es el uso de un mayor número de dígitos que facilita el manejo de cifras muy grandes.

Este tipo de sistema numérico es utilizado en sistemas digitales porque emplea números enteros en lugar de solamente ceros y unos. Se dice que este tipo de sistema numérico fue uno de los primeros en usarse debido a su semejanza con el sistema numérico decimal. El sistema octal tiene los siguientes caracteres:

Octal= {0, 1, 2, 3, 4, 5, 6, 7}

En la siguiente tabla se representa la notación octal con su respectivo equivalente decimal:

Valor Octal	**Equivalente Decimal**
0	0
1	1
2	2
3	3
4	4

5	5
6	6
7	7
10	8
11	9
12	10
13	11
14	12
15	13
16	14
17	15

Para representar un número binario en octal se separa la cifra binaria en grupos de tres dígitos y así se convierte en dígitos octales.

Ejemplo 1:

74_{10} (decimal) = 1001010_2

Separando el número binario en grupos de tres dígitos

1 / 001 / 010 = 1 = 1

001 = 1

010 = 2

El número binario 1001010_2 en octal corresponde a 112_8

Ejemplo 2:

109_{10} (decimal) = 1101101_2

Separando el número binario en grupos de tres dígitos

1 / 110 / 110 = 1 = 1

101 = 5

101 = 5

El número binario 1101101_2 en octal corresponde a 155_8

ACTIVIDADES DE APRENDIZAJE

1. Realizar ejercicios de sistema numérico octal.

1.1.3 Sistema numérico decimal

El sistema numérico decimal es el que los seres humanos utilizamos de manera cotidiana con los números naturales que van del 0 al 9 (0, 1, 2, 3, 4, 5, 6, 7, 8 y 9): 10 dígitos en total. Se dice que el sistema numérico decimal es el punto de partida para la creación de otros sistemas numéricos debido a que éste contiene todos los dígitos posibles para la creación de diferentes cantidades e incluso cantidades infinitas. El sistema decimal usa los siguientes dígitos:

Decimal= {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Este sistema maneja los números reales del 0 al 9. Dado que su base es 10, cuenta con 10 dígitos como lo muestra la siguiente tabla:

Número de Dígito	Dígito Decimal
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9

Para la conversión de números decimales a binarios se utiliza la división entre dos, debido a que los datos que se van a convertir emplean la base 2 (número binario: dos dígitos).

Ejemplo 1:

Convertir el número 41_{10} a binario

41/2 = 20 con residuo 1

20/2 = 10 con residuo 0

10/2 = 5 con residuo 0

5/2 = 2 con residuo 1

2/2 = 1 con residuo 0

1 *(como el último número no es divisible entre dos se toma la unidad)*

El número convertido se toma desde el último hasta el primero 101001

El número 41_{10} convertido a número binario es 100101

Ejemplo 2:

Convertir el número 120_{10} a binario

120/2 = 60 con residuo 0

60/2 = 30 con residuo 0

30/2 = 15 con residuo 0

15/2 = 7 con residuo 1

7/2 = 3 con residuo 1

3/2 = 1 con residuo 1

1 *(como el último número no es divisible entre dos se toma la unidad)*

El número convertido se toma desde el último hasta el primero 1111000

El número 120_{10} convertido a número binario es 1111000

ACTIVIDADES DE APRENDIZAJE

1. Realizar ejercicios de sistema numérico decimal.

1.1.4 Sistema numérico hexadecimal

El sistema numérico hexadecimal es un sistema con una estructura de 16 dígitos, que van del 0 al 9 (10 dígitos) y de la *A* a la *F* (seis dígitos). Es uno de los más utilizados en los sistemas digitales y en la ciencia de la computación, ya que con este sistema se pueden representar posiciones de memoria, sectores del disco duro y demás. El sistema hexadecimal contiene los siguientes caracteres o dígitos:

Hexadecimal={0,1 ,2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

De donde:

A=10, B=11, C=12, D=13, E=14, F=15

Número de Dígito	Equivalente Decimal	Equivalente Decimal
1	0	0
2	1	1
3	2	2
4	3	3
5	4	4
6	5	5
7	6	6
8	7	7
9	8	8
10	9	9
11	A	10
12	B	11
13	C	12
14	D	13
15	E	14
16	F	15

Para convertir un número hexadecimal en un número binario es necesario separar los números hexadecimales y convertir por separado cada uno de ellos a su equivalente binario.

Ejemplo 1:

Convertir $34EA_{16}$ a binario

3	4	E	A
0011	0100	1110	1010

Por tanto, el número $34EA_{16}$ convertido a binario es: 0011010011101010

Ejemplo 2:

Convertir 5EAC3

5	E	A	C	3
0101	1110	1010	1100	0011

Por lo tanto, el número 5EAC3 convertido a binario es: 01011110101011000011

ACTIVIDADES DE APRENDIZAJE

1. Realizar ejercicios de sistema numérico hexadecimal.

1.2 CONVERSIÓN ENTRE SISTEMAS NUMÉRICOS

Debido a que hay diferentes tipos de bases se pueden cambiar entre sí para convertirlos a distintos tipos de sistemas numéricos.

CONVERSIÓN DE BINARIO A OCTAL

Para hacer la conversión de sistema base binaria a octal se siguen estos pasos:

1. Separar los números binarios en grupos de tres dígitos, comenzando de derecha a izquierda; si el grupo final no tiene exactamente tres dígitos se le pueden añadir ceros por el lado izquierdo.

2. Una vez realizada la agrupación, se procede a verificar la cantidad correspondiente en el sistema octal.

Ejemplo:

Convertir 1001011_2 a binario

001	001	011
1	1	3

Por lo tanto, el número 1001011_2 a binario es 113

CONVERSIÓN DE BINARIO A DECIMAL

Para hacer la conversión de sistema base binaria a decimal se siguen estos pasos:

1. Comenzar por el lado izquierdo y etiquetar los dígitos binarios comenzando por el 0.
2. Multiplicar el número binario por dos y elevarlo a la potencia de acuerdo al número etiquetado anteriormente.[2]

Ejemplo:

Convertir 10110_2 a decimal

1 = 4	$1x2^4$= 16
0 = 3	$0x2^3$= 0
1 = 2	$1x2^2$= 4
1 = 1	$1x2^1$= 2
0 = 0	$0x2^0$= 0

Dada la suma, es 22

Por lo tanto, el número 10110_2 es 22_{10}

CONVERSIÓN DE BINARIO A HEXADECIMAL

Para hacer la conversión de sistema base binaria a hexadecimal se siguen estos pasos:

[2] Roland Tocci y Neal Widmer, *op. cit.*, p. 26.

1. Se debe separar los números binarios en grupos de cuatro dígitos comenzando de derecha a izquierda, si el grupo final no tiene exactamente cuatro dígitos se le puede añadir ceros por la parte izquierda.
2. Una vez realizada la agrupación se procede a verificar la cantidad correspondiente en el sistema hexadecimal.

Ejemplo:

Convertir 110110_2 a hexadecimal

0011	0110
3	6

Por lo tanto, el número 110110_2 a hexadecimal es 36

CONVERSIÓN DE OCTAL A BINARIO

Para hacer el cálculo de conversión del sistema numérico octal a binario se realiza el cálculo directo: se separan los dígitos octales y se procede a convertirlos a binarios de acuerdo a su equivalencia. La conversión que se haga de octal a binario debe ser de tres caracteres o dígitos binarios únicamente.

Ejemplo:

Convertir 231_8 a binario

2	3	1
010	011	001

Por lo tanto, el número 231_8 convertido a binario es: 010011001

CONVERSIÓN DE OCTAL A DECIMAL

Para hacer la conversión de sistema base octal a decimal se siguen estos pasos:

1. Comenzar por el lado izquierdo y etiquetar los dígitos octales comenzando por el 0.
2. Multiplicar el número octal por ocho y elevarlo a la potencia de acuerdo al número etiquetado anteriormente.

Ejemplo:

Convertir 564_8 a decimal

5 = 2 $5x8^2= 320$

6 = 1 $6x8^1= 48$

4 = 0 $4x8^0= 4$

Por lo tanto la suma es 370

El número 564_8 corresponde a 370_{10} decimal

CONVERSIÓN DE OCTAL A HEXADECIMAL

Para convertir un número octal a hexadecimal se necesitan dos pasos, ya que no existe un método directo para hacer la conversión.

1. Convertir el número octal a binario.
2. Convertir el número binario a hexadecimal.

Ejemplo:

Convertir 347_8 a hexadecimal

3	4	7
011	100	111

Por tanto, 347_8 en binario es 011100111_2

Convertir 011100111_2 a hexadecimal

1110	0111
E	7

Por lo tanto, 347_8 corresponde a $E7_{16}$ hexadecimal

1.3 REPRESENTACIÓN DE LOS NÚMEROS NEGATIVOS BINARIOS

En los sistemas numéricos existen tanto números positivos como negativos, de tal manera que pueden diferenciarse entre sí; aunque también pueden hacerse diferencias entre dos cifras. En el sistema numérico binario se presentan los números negativos; como en todas las representaciones se manejan sólo los mismos caracteres 0 y 1, se dice que es un sistema algo confuso.

En las operaciones de resta de números negativos binarios ocurre un error muy común: al restar, los números binarios aparecen como positivos, por lo que al hacer la operación erróneamente se suman.

Para hacer una resta binaria se necesita que uno de los números sea negativo, para que al momento de hacer la operación realmente se reste. Dentro de la representación de números negativos, según los datos se realiza una resta, pero realmente se está realizando una suma ya que uno de los dos datos es negativo.

La representación de números negativos dentro del sistema numérico binario depende de una técnica llamada *complemento*. Existen dos complementos diferentes.

Complemento a 1.

Es aquel que cambia los elementos 0 por los 1 y los 1 por los 0 del número a convertir a negativo, ejemplo:

24_{10} corresponde a 11000_2

El complemento a 1 de 11000_2 es:

00111_2

El complemento a 2 es aquel que a la cifra a modificar a negativo se le añade o suma un bit más previamente convertido a complemento a 1, ejemplo:

El complemento a 2 de 00111_2 es:

$$\begin{array}{r} 00111_2 \\ \underline{\quad 1_1} \\ 01000_2 \end{array}$$

Por lo tanto el número 24_{10} convertido a binario negativo es: 01000_2

Para saber si al momento de hacer la operación de resta un número es negativo o positivo, se verifica éste mediante un bit adicional que se recorre al principio del resultado de la operación. A este bit se le conoce como *bit de acarreo* y es aquel que nos indica si el número es negativo o positivo dependiendo del bit acarreado. Si el bit acarreado es 1, el resultado es positivo; si el bit acarreado es 0, el resultado es negativo.

Ejemplo:

30_{10} a binario es 11110_2

24_{10} a binario negativo es 01000_2

La resta de los dos dígitos es:

$$
\begin{array}{r}
11110 \\
\underline{-01000} \\
\underline{1}00110
\end{array}
$$

El bit de acarreo dio como resultado 1 que significa que el número obtenido es positivo.

Ejemplo 2:

Restar 24_{10}-14_{10}

24_{10} a binario es 11000_2

14_{10} a binario es 1110_2

Primero hay que convertir el 14_{10} a binario negativo. Para lograrlo debemos implementar los complementos. Debido a que el número binario positivo es de cinco caracteres, necesitamos agregar un bit más al número binario negativo. Esto se logra agregando un cero a la izquierda.

Complemento a 1: 01110_2 (cambiar los 0 por 1 y los 1 por 0)

10001_2

Complemento a 2: 10001_2 (sumar un bit más al resultado del complemento a 1)

$$
\begin{array}{r}
\underline{\quad\quad 1_2} \\
10010_2
\end{array}
$$

14_{10} a binario negativo es 10010_2

Procedemos a realizar la resta:

$$\begin{array}{r} 11000_2 \\ -\ \underline{10010_2} \\ 101010 \end{array}$$ (El bit de acarreo es 1 por lo tanto el resultado es positivo)

El número 1010_2 convertido a decimal es: 10_{10} por lo tanto 24_{10}-14_{10} es 10_{10} el resultado es correcto.

AUTOEVALUACIÓN

Responde los siguientes cuestionamientos:

1. ¿Qué es un sistema numérico?
2. ¿Qué elementos forman el sistema numérico decimal?
3. ¿Qué elementos forman el sistema numérico octal?

Representa las siguientes cifras en base hexadecimal:

1. 9287_{10}
2. 237_{10}
3. 3434_{10}
4. 342_{10}

Realiza las siguientes conversiones:

1. 1101_{2} a decimal.
2. 175_{16} a decimal.
3. 110_{10} a binario.
4. 1032_{10} a binario.

RESPUESTAS

1. Son las representaciones de medidas en escalas establecidas. Estas medidas se pueden dar de dos diferentes tipos, tanto analógicas como digitales.

2. Los elementos que conforman al sistema numérico decimal son los dígitos 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9, también conocidos como naturales.

3. Los elementos que conforman al sistema numérico octal son los dígitos 0, 1, 2, 3, 4, 5, 6 y 7.

4. 269F

5. ED

6. D6A

7. 156

8. 13

9. 373

10. 1101110

11. 10000001000

UNIDAD 2

SIMPLIFICACIÓN DE FUNCIONES

OBJETIVO

Resolver funciones lógicas digitales a través de lógica booleana.

TEMARIO

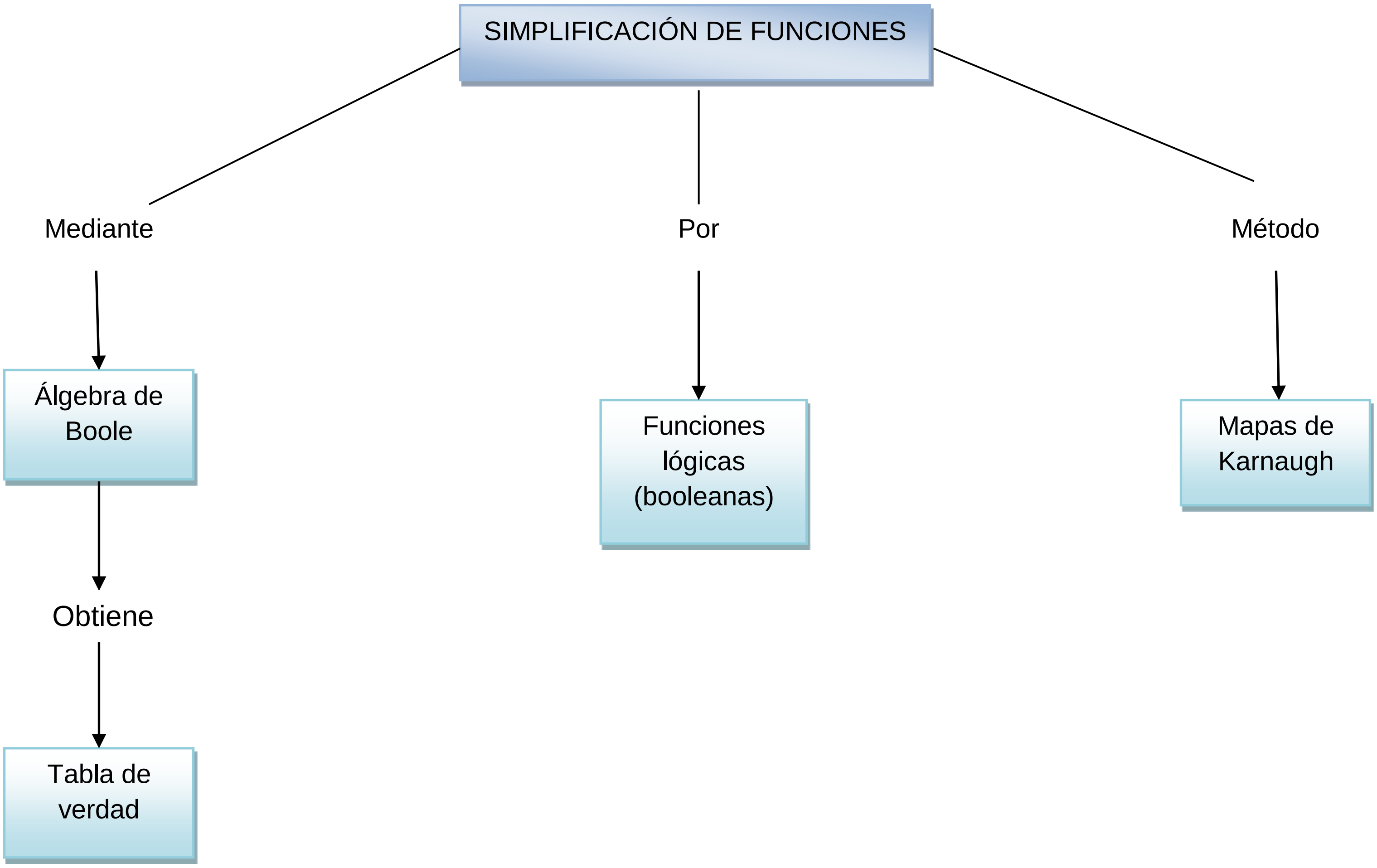
2.1 ÁLGEBRA DE BOOLE

2.1.1 Tabla de verdad

2.2 FUNCIONES LÓGICAS (BOOLEANAS)

2.3 MAPAS DE KARNAUGH

MAPA CONCEPTUAL



INTRODUCCIÓN

En esta unidad se abordará el tema de simplificación de funciones mediante los métodos de álgebra de Boole así como de los mapas de Karnaugh.

En cada uno de los temas, se comenzará con su definición para luego desarrollar ejercicios para la aplicación de los mismos y se tomarán las funciones lógicas para implementar los circuitos lógicos en un ambiente práctico con la implementación de circuitos.

2.1 ÁLGEBRA DE BOOLE

El álgebra booleana es la base fundamental del funcionamiento de los circuitos lógicos digitales, debido a que está presente en los estados que pueden llegar a tener los distintos circuitos digitales. En la actualidad todos los circuitos digitales actúan mediante “estados”. También se conocen como activado o desactivado, según la entrada a la que se haga referencia. Por ejemplo, un interruptor simple únicamente puede llegar a tener dos estados, que son: estado abierto (cuando no hay flujo de corriente) y cerrado (cuando hay flujo de corriente). En sistemas digitales los circuitos manejan los mismos estados, pero a diferencia de lo anterior, ellos sólo manejan estado activado (representado por el 1) o desactivado (representado por el 0).

A todo esto el álgebra booleana representa los estados que pueden llegar a tener los circuitos, dado que el álgebra booleana únicamente maneja dos posibles resultados: el 0 y el 1, y a esto se le conoce como NIVEL LÓGICO. A continuación se presenta una tabla con las posibles aplicaciones de los valores lógicos booleanos.

0 Lógico	1 Lógico
Falso	Verdadero
Bajo	Alto
Desactivado	Activado
Abierto	Cerrado
No	Sí

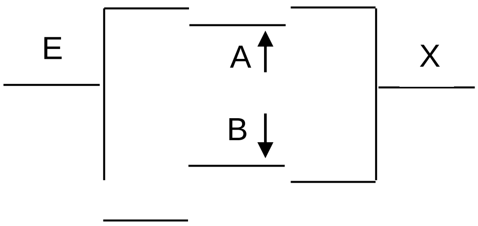
El álgebra booleana se utiliza principalmente para definir los posibles estados que pueden llegar a presentarse en los circuitos digitales así como también las posibles salidas que pueden llegar a tener estos mismos circuitos. La manera de operar los circuitos digitales es mediante entradas denominadas variables lógicas y cuyos estados o niveles determinarán los estados o niveles de salida.

Una de las aplicaciones que llega a tener el álgebra booleana es que es muy sencilla de manejar debido a que sólo controla dos posibles resultados: el 0 y el 1, Otra de las ventajas que tiene es que no se manejan fracciones ni decimales entre muchas otras más relativas al álgebra común.

En el álgebra booleana solamente se manejan tres operaciones básicas: AND, OR y NOT, que reciben el nombre de OPERACIONES LÓGICAS o FUNCIONES LÓGICAS.

Dentro del álgebra de Boole se puede asociar las funciones lógicas en interruptores, por ejemplo:

La función lógica OR es una función lógica de suma donde indica que si cualquiera de las entradas es positiva (estado 1), la salida es positiva (estado 1).



Donde:

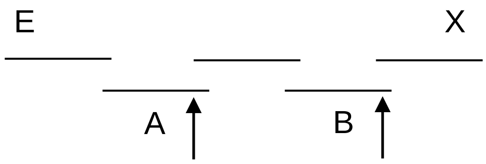
E = entrada

A = Interruptor A

B = Interruptor B

X = Salida

La función lógica AND es una función lógica de multiplicación que se representa por medio de interruptores simples. La función lógica AND indica que la salida será activada (estado 1) siempre y cuando las dos entradas estén activadas (estado 1), en caso contrario no se activará la salida.



Donde:

E = entrada

A = Interruptor A

B = Interruptor B

X = Salida

La función lógica OR es una función lógica booleana que representa una negación simple. Si el estado está activado (estado 1), la salida será desactivada (estado 0) y en caso contrario, si la entrada es desactivada (estado 0), la salida será activada (estado 1)

Donde:

E = entrada

A = Interruptor A

X = Salida

2.1.1 Tabla de verdad

La tabla de verdad es un instrumento que sirve como referencia para el uso de las funciones lógicas debido a que en ella, se hace referencia a los distintos estados de entrada que pueden llegar a tener los circuitos digitales que a su vez, dan la salida del circuito o resultado.

Las tablas de verdad se construyen dependiendo del número de entradas presentes en los circuitos. Éstos a su vez, dan el resultado lógico de salida que va a obtener el circuito. Para la construcción de las tablas de verdad se debe tener en cuenta el número de entradas ya que en ellas se desarrollan todas las demás combinaciones que llegaría a tener el circuito. La construcción debe ser hecha a partir de dos entradas debido a que si el circuito llegara a tener una entrada los posibles resultados únicamente serían dos (0 para falso y 1 para verdadero).

El número de combinaciones que puede llegar a tener una tabla de verdad es igual al número de entradas que puede llegar a tener. En concreto, para llegar a saber el número exacto de combinaciones basta con realizar la siguiente expresión: 2^n donde n es el número de entradas.

Las posibles combinaciones que se llegan tener son igual al conteo binario debido a que se sigue una lógica secuencial con números binarios. A continuación se dan a conocer tablas de verdad de dos, tres y cuatro entradas.

Tabla de verdad de dos entradas:

Entrada A	Entrada B	Salida
0	0	1
0	1	0
1	0	1
1	1	0

Tabla de verdad de tres entradas:

Entrada A	Entrada B	Entrada C	Salida
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Tabla de verdad de cuatro entradas:

Entrada A	Entrada B	Entrada C	Entrada D	Salida
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1

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

ACTIVIDADES DE APRENDIZAJE

1. Resolver ejercicios de tabla de verdad.

2.2 FUNCIONES LÓGICAS (BOOLEANAS)

Las funciones lógicas booleanas son tres y son las únicas tres operaciones que se puede llegar a realizar. Cada una de ellas tiene un tipo de codificación implícita y es diferente para cada una de las tres. Es como si cada una de las funciones lógicas tuviera presente una tabla de verdad impresa en ella. Estas mismas operaciones se pueden llegar a mezclar para crear otras funciones lógicas.[3]

Las funciones lógicas son tres: OR, AND y NOT.

La función OR es la primera operación booleana básica. El único caso especial en su tabla de verdad es cuando los dos estados de entrada son falsos (dos estados 0), la salida va a ser falsa. Para los demás casos, cuando esté presente un estado verdadero (estado en 1) la salida será verdadera. La expresión booleana OR de operación es:

$$X = A + B$$

Donde:

X = Salida

A = Entrada A

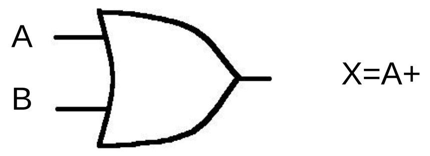
B = Entrada B

Por lo tanto, la tabla de verdad para la función lógica OR quedaría de la siguiente manera:

[3] *Ibidem.*, pp. 58-62.

A	B	X=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Representación gráfica de la función OR



La función AND es la segunda operación booleana básica. El único caso especial en su tabla de verdad es cuando los dos estados de entrada son verdaderos (dos estados 1), la salida va a ser verdadera. Para los demás casos, cuando esté presente un estado falso (estado en 0), la salida será falsa y es prácticamente lo contrario de la compuerta lógica OR. La expresión booleana AND de operación es:

$$X = A * B$$

Donde:

X = Salida

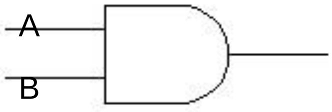
A = Entrada A

B = Entrada B

Por lo tanto, la tabla de verdad para la función lógica OR quedaría de la siguiente manera:

A	B	X=A*B
0	0	0
0	1	0
1	0	0
1	1	1

Representación gráfica de la función AND



X=A+

La función NOT es completamente distinta a las otras operaciones booleanas debido a que ésta únicamente puede ser controlada por una variable de entrada ya que sólo maneja un dato. Si una variable A es sometida a la operación NOT queda de la siguiente manera:

$$X = \overline{A}$$

Donde:

X = Salida

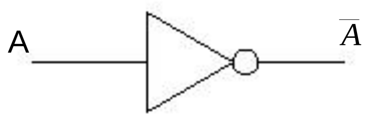
A = Entrada A

En la operación booleana NOT se sobrepone una barra que significa la negación de la salida. Ejemplo:

$\overline{1}$ = 0 Debido a que NOT 1 es 0

$\overline{0}$ = 1 Debido a que NOT 0 es 1

Representación gráfica de la función NOT



ACTIVIDADES DE APRENDIZAJE

1. Investigar las funciones lógicas booleanas.

2.3 MAPAS DE KARNAUGH

Los mapas de Karnaugh son un método gráfico que se utiliza para simplificar funciones booleanas. Sirven para demostrar las posibles reducciones que puede sufrir un circuito lógico digital dependiendo de los estados en los cuales esté activado (estado 1) y dependiendo también de su tabla de verdad.

El mapa de Karnaugh utiliza la tabla de verdad del circuito lógico que se va a reducir como base fundamental. En esta tabla de verdad se escogen los estados en los que se va a reducir el circuito. Dado que se pueden escoger los estados activados de los circuitos, también se pueden escoger los estados desactivados.

Dentro de la tabla de verdad se obtienen los datos de salida y su combinación de datos de entrada. A cada uno de los estados de la tabla de verdad le corresponde un dato en el mapa de Karnaugh.[4]

En pocas palabras, los mapas de Karnaugh se utilizan para poder simplificar las funciones lógicas booleanas, partiendo de su tabla de verdad y escogiendo el estado en el cual el circuito debe realizar una función en específico. Al hablar de función en específico, nos referimos a cuando el sistema se activa (estado 1).

Ejemplo 1: Tabla de verdad de dos entradas y una salida, se desea simplificar la función mediante mapa de Karnaugh cuando la salida sea positiva (estado 1).

Entrada A	Entrada B	Salida X
0	0	1
0	1	0
1	0	0
1	1	1

El primer paso es identificar las salidas activadas (estado 1) de la tabla de verdad.

Primer estado: 1 → $\bar{A}\ \bar{B}$

[4] *Ibidem.,* p. 122.

Segundo estado: 1 → AB

Se realiza una tabla adicional que se llama mapa de Karnaugh donde se colocan los datos obtenidos de la tabla de verdad.

	$\bar{B}$	B
$\bar{A}$	1	0
A	0	1

Por último, se crea la ecuación que satisface al mapa de Karnaugh donde sólo seleccionará los estados positivos o activados (estado 1).

$$\{ X = \bar{A}\,\bar{B} + A\,B \}$$

Ejemplo 2: Tabla de verdad de tres entradas y una salida; se desea simplificar la función mediante mapa de Karnaugh cuando la salida sea positiva (estado 1).

Entrada A	Entrada B	Entrada C	Salida X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

El primer paso es identificar las salidas activadas (estado 1) de la tabla de verdad.

Primer estado: 1 → $\bar{A}\,\bar{B}\,\bar{C}$

Segundo estado: 1 → $\bar{A}\,\bar{B}\,C$

Tercer estado: 1 → $\bar{A}\,B\,\bar{C}$

Cuarto estado: 1 → $AB\bar{C}$

Se realiza una tabla adicional que se llama mapa de Karnaugh donde se colocan los datos obtenidos de la tabla de verdad.

	$\overline{B}$	B
$\overline{A}\,\overline{B}$	1	1
$\overline{A}B$	1	0
AB	1	0
$A\overline{B}$	0	0

Por último se crea la ecuación que satisface al mapa de Karnaugh donde sólo seleccionará los estados positivos o activados (estado 1).

$$\{\, X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + AB\overline{C} \,\}$$

Ejemplo 3: Tabla de verdad de cuatro entradas y una salida; se desea simplificar la función mediante mapa de Karnaugh cuando la salida sea positiva (estado 1).

Entrada A	Entrada B	Entrada C	Entrada D	Salida X
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

El primer paso es identificar las salidas activadas (estado 1) de la tabla de verdad.

Primer estado: 1 → $\overline{A}\,\overline{B}\,\overline{C}D$

Segundo estado: 1 → $\overline{A}B\overline{C}D$

Tercer estado: 1 → $AB\overline{C}D$

Cuarto estado: 1 → ABCD

Se realiza una tabla adicional que se llama mapa de Karnaugh donde se colocan los datos obtenidos de la tabla de verdad.

	$\bar{C}\,\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\,\bar{B}$	0	1	0	0
$\bar{A}B$	0	1	0	0
AB	0	1	1	0
$A\bar{B}$	0	0	0	0

Al final se elabora la ecuación que satisface al mapa de Karnaugh donde sólo seleccionará los estados positivos o activados (estado 1).

$$\{ X = \bar{A}\,\bar{B}\,\bar{C}D + \bar{A}B\bar{C}D + AB\bar{C}D + ABCD \}$$

ACTIVIDADES DE APRENDIZAJE

1. Definir qué es un mapa de Karnaugh y sus propiedades.

2. Realizar ejemplos de mapas de Karnaugh con sus respectivas tablas de verdad.

AUTOEVALUACIÓN

Responde los siguientes cuestionamientos:

1. ¿Qué entiendes por álgebra booleana?

2. ¿Qué entiendes por nivel lógico?

3. ¿Qué es una función lógica booleana?

4. ¿Cuáles son las funciones lógicas booleanas?

5. Realiza el mapa de Karnaugh de la siguiente tabla de verdad.

Entrada A	Entrada B	Salida X
0	0	0
0	1	1
0	0	0
1	1	1

RESPUESTAS

1. Es la base fundamental del funcionamiento de los circuitos lógicos digitales debido a que esta álgebra se ve presente en los estados que pueden llegar a tener los distintos circuitos digitales.

2. Son los estados que pueden llegar a tener los circuitos lógicos digitales que son el estado en 1 (activado) y 0 (desactivado)

3. Es la que se utiliza principalmente para definir los posibles estados que pueden llegar a presentarse en los circuitos digitales así como también las posibles salidas.

4. Las funciones lógicas booleanas son AND, OR y NOT.

5. $X = \bar{A}B + AB$

UNIDAD 3

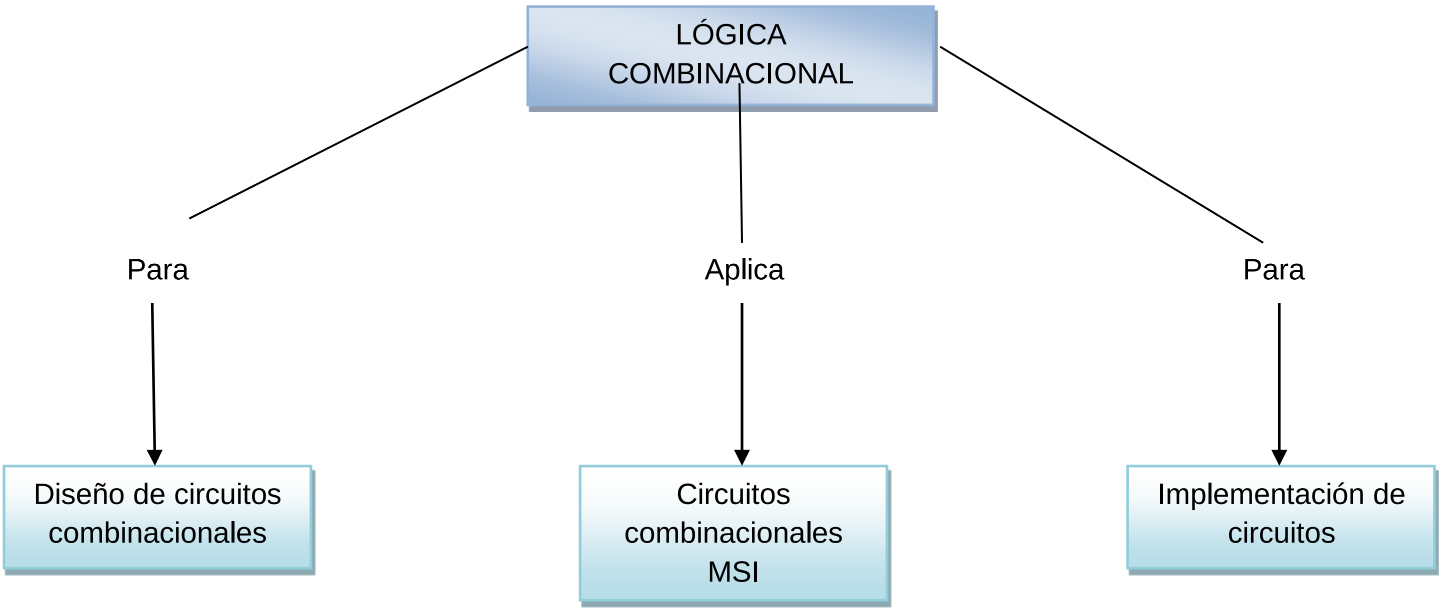
LÓGICA COMBINACIONAL

OBJETIVO

Analizar los circuitos lógicos combinacionales así como sus aplicaciones.

TEMARIO

MAPA CONCEPTUAL



INTRODUCCIÓN

La lógica secuencial se aplica en los sistemas digitales para la implementación de circuitos. Es el funcionamiento que se desea transmitir al circuito dado que en él se encuentra toda la lógica aplicada.

En esta unidad se dedicará una práctica para revisar los conocimientos adquiridos: será el punto clave en la unidad ya que se verán los temas abarcados.

3.1 DISEÑO DE CIRCUITOS COMBINACIONALES

En el diseño, el punto a analizar consiste en determinar qué circuito cumple con determinadas especificaciones de funcionalidad y de trabajo. Esto se puede resumir en dos puntos esenciales:

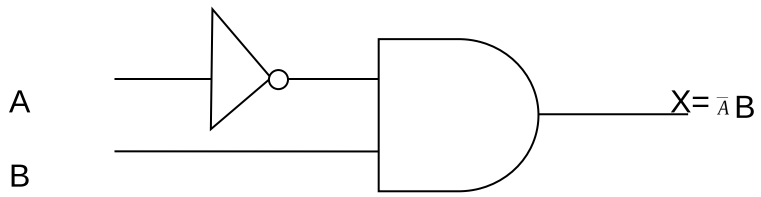
- Obtención de la función que cumpla la funcionalidad deseada.
- Obtención de los circuitos que cumplan con la función deseada.

En la siguiente tabla de verdad se muestran dos entradas y una salida que obtienen su estado en alto sólo cuando A=0 y B=1 la salida será X=1.

Entrada A	Entrada B	Salida X
0	0	0
0	1	1
1	0	0
1	1	0

Según la tabla de verdad, el resultado de salida sería $X=\bar{A}B$ donde únicamente se puede llegar a ese valor cuando la entrada A sea falsa y la entrada B sea verdadera.

De manera gráfica quedaría así:



El procedimiento para el diseño de circuitos combinacionales depende de los elementos de salida de la tabla de verdad, debido a que ésta da la pauta de lo que nosotros debemos hacer y construir.

Una vez identificado el tipo de sistema de funciones lógicas que se va a construir, se procede a implementar las compuertas lógicas con su respectivo funcionamiento. Para ello, se deben implementar las compuertas lógicas OR, AND y NOT.[5]

Ejemplo 1: Se desea construir un circuito lógico operacional a partir de la siguiente tabla de verdad:

---

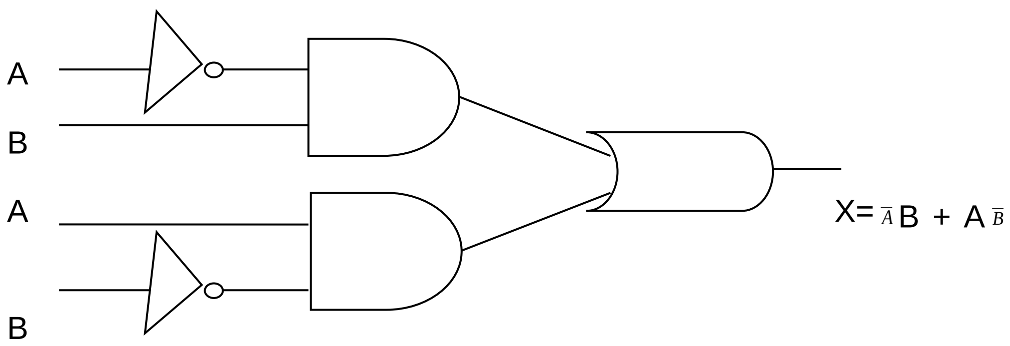
5 *Ibidem*., p. 115.

Entrada A	Entrada B	Salida X
0	0	0
0	1	1
1	0	1
1	1	0

Primero se identifican las salidas en alto para entonces partir a la construcción de los circuitos con las compuertas lógicas. Los puntos en alto son:

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

Una vez teniendo el resultado que arroja la tabla de verdad se procede a la construcción del circuito lógico que se va a desarrollar.



Ejemplo 2:

Se necesita diseñar un circuito que tenga tres entradas y una salida; la salida va a ser en alto cuando la mayoría de las entradas estén en alto. Esto significa que el circuito va ser activado cuando al menos dos de las entradas estén en alto.

Primero, se establece la tabla de verdad:

Entrada A	Entrada B	Entrada C	Salida X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Se establece la salida según la tabla de verdad:

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

Como el resultado de la salida es amplio, el diseño del circuito sería muy grande. Para evitar hacer un trabajo muy extenso, se procede a reducir la expresión.

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

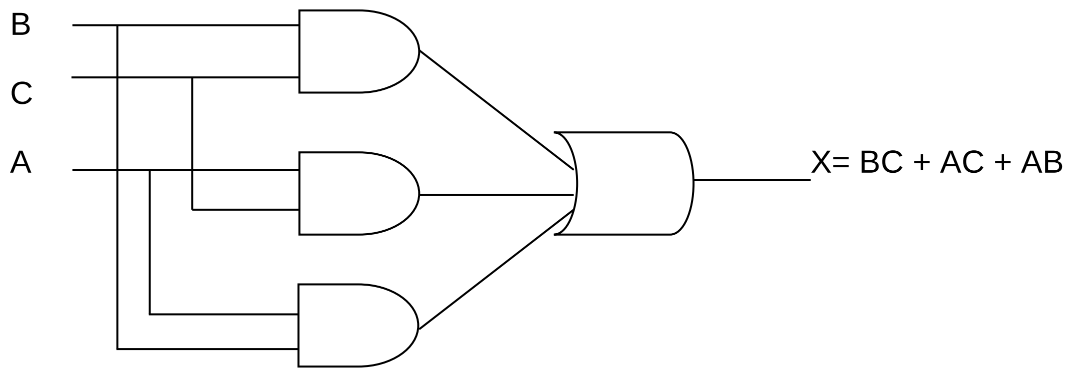
Factorizando los términos tenemos:

$X = BC(\bar{A} + A) + AC(\bar{B} + B) + AB(\bar{C} + C)$

Como cada término entre parentesis es igual a 1, tenemos:

$X = BC + AC + AB$

Como se tiene el resultado de la tabla de verdad que cumple con el objetivo, se procederá a diseñar el circuito logico combinacional.



ACTIVIDADES DE APRENDIZAJE

1. Investigar los diferentes tipos de circuitos lógicos combinacionales con sus aplicaciones.

3.2 Circuitos combinacionales MSI

Los circuitos integrados han tenido una evolución enorme; los que se manejan actualmente han llegado a reducir su tamaño además de alcanzar una capacidad para contener gran cantidad de infomación en ellos.

Los circuitos combinacionales abarcan diversos elementos como se muestra a continuación:

- Escala baja de integración (SSI) de 2 a 12 compuertas lógicas.

- Escala media de integración (MSI), hasta 100 compuertas lógicas.
- Escala alta y muy alta de integración (LSI y VLSI), más de 100 compuertas lógicas.

Entre las ventajas que ofrecen los circuitos lógicos combinacionales MSI se encuentran:

- Soluciones compactas, debido a que en ellos se encuentran todos los métodos exactos para las tareas especiales.
- Se hacen menos conexiones debido a que todas las integraciones ya están hechas internamente.
- Las posibles salidas del circuito se hacen antes debido a que ya están configurados.
- Son escalares, se pueden colocar varios circuitos en serie para aumentar su funcionabilidad.

La gama de circuitos lógicos operacionales comerciales se divide en familias:

- Circuitos aritméticos (sumadores, comparadores).
- Generadores de paridad.
- Multiplexores y demultiplexores.
- Codificadores y decodificadores.

Los circuitos aritméticos, como su nombre lo indica, son circuitos que tienen operaciones aritméticas codificadas internamente debido a que los circuitos programados para esas funciones se encuentran dentro de ellos. Todas las operaciones que estos circuitos tengan que hacer se llevan a cabo mediante lógica combinacional a través de codificación binaria.

Las operaciones se realizan mediante código binario, debido a que es el único lenguaje que entienden para su funcionalidad.

Los generadores de paridad son los circuitos lógicos MSI que detectan una entrada, debido a que están programados para seguir una secuencia. Estos circuitos están fabricados para seguir una secuencia lógica: si la entrada es par, generan una entrada 0 y si la entrada es impar, generan una entrada en 1 para seleccionar la instrucción que deben seguir.

Los detectores conocidos como de paridad son aquellos que toman parte de las entradas del circuito, convirtiéndolo en otro bit de datos dentro del circuito lógico. Su finalidad es la de detectar algún error dentro del circuito implementado.

Los circuitos multiplexores y demultiplexores son aquellos que utilizan un mismo canal de comunicaciones para generar más salidas. La función principal de estos circuitos es la de incrementar su rendimiento eliminando canales de comunicacióninnecesarios .

En los circuitos codificadores y decodificadores, el circuito codificador es aquel que tiene como objetivo transmitir información, aunque ya se encuentre en otro formato distinto al original.[6]

El circuito decodificador es aquel que tiene como finalidad ejecutar un trabajo específico, según la entrada a la que se le haga referencia. También se utiliza para convertir equivalentes numéricos, pasar de binarios a octales o decimales a hexadecimales, por ejemplo.

ACTIVIDADES DE APRENDIZAJE

1. Realizar una investigación sobre los circuitos combinacionales MSI y elaborar un resumen.

2. Formular una conclusión personal acerca de los circuitos lógicos combinacionales MSI.

3.3 IMPLEMENTACIÓN DE CIRCUITOS

OBJETIVO

Implementar un circuito integrado MSI para verificar el funcionamiento de cada una de sus terminales.

La implementación de circuitos se refiere a la ejecución física de éstos pero tomando en cuenta sus características de construcción, con la finalidad de obtener un buen cumplimiento del circuito msi.

El siguiente es un circuito MSI biestal para su implementación.

[6] María José Gómez Caño y Thomas L. Floyd, *Fundamentos de sistemas digitales*, p. 276

Materiales:

Circuito integrado NE555

2 Resistencias de 10K

2 Capacitores de 16MF

1 Potenciómetro de 100K

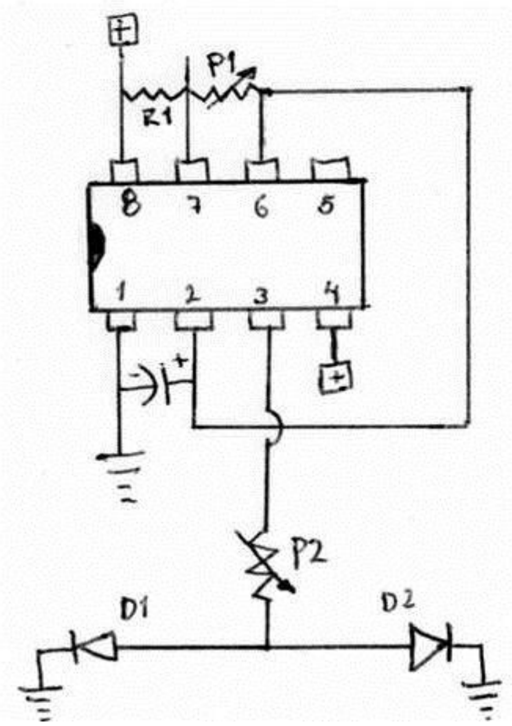
1 Potenciómetro de 5K

5 Led's

1 Protoboard

Fuente de alimentación de 5v

1m de cable para redes CAT 5



En la práctica se presenta un circuito MSI biestal al que se le puede regular la entrada y también la salida. El circuito es la implementación de un circuito MSI de forma física.

ACTIVIDADES DE APRENDIZAJE

1. Ejemplificar físicamente un circuito digital mediante sus componentes físicos y diagrama.

AUTOEVALUACIÓN

1. Realizar el circuito lógico combinacional de la siguiente tabla de verdad:

Entrada A	Entrada B	Salida X
0	0	0
0	1	0
1	0	1
1	1	0

El sistema combinacional únicamente debe accionarse cuando la salida *X* esté en alto (estado en 1).

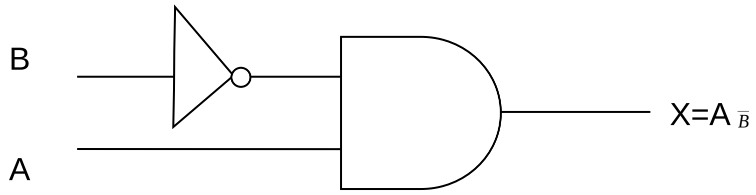
Responde a los siguientes cuestionamientos:

2. ¿Qué es un circuito MSI?

3. ¿Cuáles son las ventajas de los circuitos MSI?

4. ¿Qué número de compuertas lógicas soporta un circuito MSI?

RESPUESTAS

1. 

2. Por sus siglas en inglés, MSI (escala media de integración) son los circuitos lógicos combinacionales de escala media, conformados por una gran cantidad de elementos que llegan hasta las 100 compuertas lógicas.

3. Entre sus ventajas están:
    - Soluciones compactas, debido a que en ellos se encuentran todos los métodos exactos para realizar las tareas especiales.
    - Se hacen menos conexiones debido a que todas las integraciones ya están hechas de manera interna.
    - Las posibles salidas contenidas en el circuito se hacen previamente debido a que ya están configurados.
    - Son escalares, se pueden colocar varios circuitos en serie para aumentar su funcionabilidad.

4. Los circuitos lógicos operacionales llegan a soportar hasta 100 compuertas lógicas en su configuración.

UNIDAD 4

LÓGICA SECUENCIAL

OBJETIVO:

Dar a conocer el funcionamiento lógico de los elementos biestales y sus características de construcción.

TEMARIO

4.1 ELEMENTOS BIESTALES

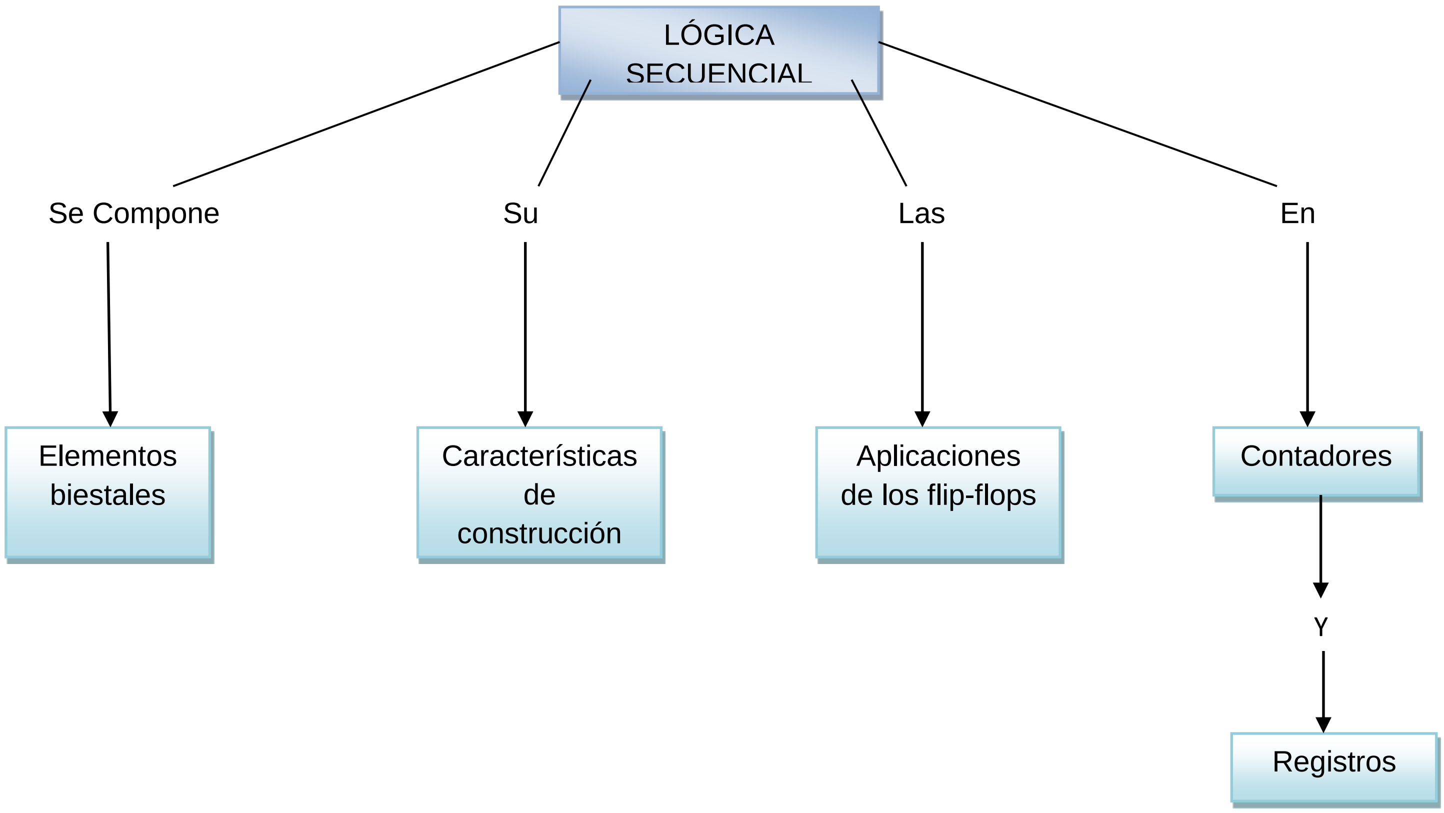
4.2 CARACTERÍSTICAS DE CONSTRUCCIÓN

4.3 APLICACIONES FLIP-FLOPS

4.4 CONTADORES

4.5 REGISTROS

MAPA CONCEPTUAL



INTRODUCCIÓN

En esta unidad se abordan los elementos digitales cuya característica principal es que pueden actuar de dos maneras distintas, ya que son capaces de tener internamente un elemento más, al que se le conoce como elemento biestal.

Además de definir los elementos biestales, se explican las características que deben contener, las aplicaciones que tienen y sus cualidades en el manejo de los estados; asimismo, nos enfocaremos en otros elementos que tienen esta misma característica, como los contadores y registros.

4.1 ELEMENTOS BIESTALES

Los circuitos biestales son circuitos digitales, capaces de almacenar un bit de información durante un tiempo indeterminado. De ahí la utilidad que se les da como elementos de memoria en un circuito electrónico. Se les suele denominar flip-flop ya que poseen dos estados en los que su comportamiento es estable: el estado de nivel alto y de nivel bajo. A estos estados, en la terminología electrónica en el campo práctico, se les denomina como 1 y 0 lógicos.

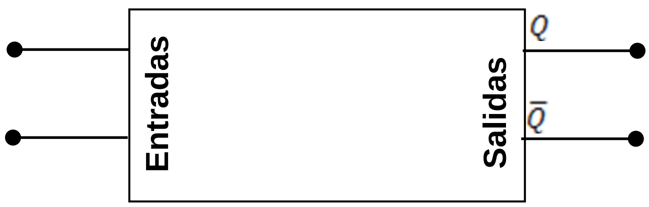
El funcionamiento de los circuitos biestales es a través de pulsos de reloj. Se considera que los circuitos biestales son el fundamento de los circuitos secuenciales.

En las ciencias de la electrónica y computación, los circuitos digitales que componen el hardware utilizados en la vida cotidiana, corren datos en forma binaria. Cada circuito digital del hardware está diseñado en función de ciertas necesidades, por las que es capaz de realizar conteos, sumas y restas de datos binarios que circulan por él; aunque no sería capaz de ejecutar todo esto si no fuera gracias al empleo de compuertas lógicas combinadas.

Ocurre que en los circuitos digitales, tanto las entradas como en las salidas del circuito, cambian de acuerdo con los pulsos de reloj que se generen en él. Suponga un circuito digital diseñado para realizar la suma de dos números en binario, pero el usuario debe sumar tres números. Si nuestro sistema está diseñado para la suma de dos números, necesita el resultado almacenado. La pregunta es dónde almacenar dicho resultado para cuando al sistema se le introduzca el resultado, Aquí es donde los circuitos biestales son de utilidad, ya que son capaces de almacenar datos por un tiempo indeterminado, hasta el momento en que el proceso lo requiera.

4.2 CARACTERÍSTICAS DE CONSTRUCCIÓN

En su expresión más sencilla, un flip-flop se representa de forma gráfica por el siguiente esquema:



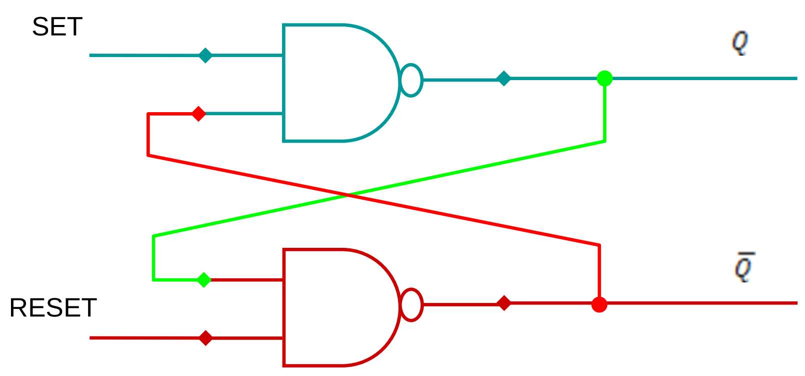
Donde Q es la salida normal del circuito, $\bar{Q}$ es la salida invertida o salida negada del circuito.

Debido al diseño de los flip-flops, éstos pueden tener varias entradas pero únicamente dos salidas. Dichas salidas son contrarias a las entradas. La siguiente tabla muestra el comportamiento de las salidas en un flip-flop.

Salida	**Símbolo**	**Estado 1 SET**	**Estado 2 RESET**
Normal	Q	1	0
Invertida	$\bar{Q}$	0	1

La variación de las entradas en un flip-flop da paso a que éstos se categoricen en flip-flops asíncronos, los cuales sólo tienen entradas de control y flip-flops síncronos con una entrada extra proveniente de un reloj -además de contar con sus entradas-..

El registro básico NAND está compuesto por dos compuertas NAND o NOR con dos entradas, conectadas de forma que una entrada de cada compuerta quede libre, la otra entrada queda conectada a la salida de la compuerta compañera como se ilustra en la siguiente figura.[7]



[7] *Ibidem*., p. 502.

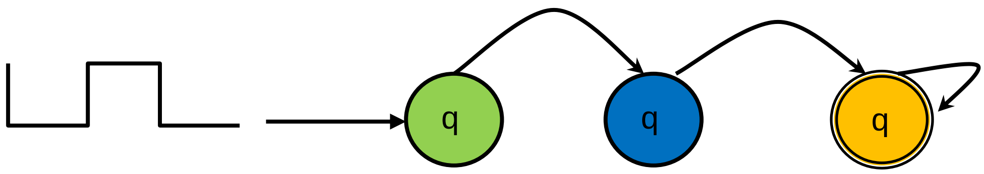
ACTIVIDADES DE APRENDIZAJE

1. Investigar sobre el funcionamiento fundamental de un circuito biestal y elaborar una síntesis.

4.3 Aplicaciones de los flip-flops

Una vez aclarado el término biestal o flip-flop como aquel circuito digital que contiene dos estados binarios estables, es necesario hacer mención de los tipos de biestales RS (Set Reset) asíncrono, RS síncrono (Set Reset), biestal D (Delay), biestal T (Toggle), biestal JK (Jump Keep) los cuales tienen diferentes aplicaciones en circuitos digitales.

Los flip-flops se usan en diferentes circuitos electrónicos pero la aplicación más común es para diseñar máquinas de estados finitos. Las máquinas de estados finitos o autómata finito son aquellas que realizan sumas de forma automática sobre una entrada para producir resultados en una salida. La mayor parte de entradas son generadas de otras salidas como se ilustra en el siguiente ejemplo.



El biestal T se aplica para hacer conteos, ya que una señal repetitiva de reloj hace que el biestal cambie de estado por cada transición de la señal digital, cuando ésta cambia de estado; es decir, de 1 a 0 a 1 a 0…

4.4 Contadores

La unidad lógica aritmética mejor conocida como ALU, es el corazón de cualquier procesador de las computadoras personales de estos días. Es un circuito digital que se encarga de realizar las operaciones aritméticas como la suma, resta, multiplicación y comparación lógica de las entradas que recibe. Esto es en su esencia más simple ya que en la actualidad, los procesadores modernos contienen una ALU capaz de realizar varias operaciones a la vez.

Para comprender cómo se lleva a cabo el conteo, retomemos los principios de la suma binaria la cual denota que:

$$0 + 0 = \mathbf{0}$$

$$0 + 1 = \mathbf{1}$$

$$1 + 0 = \mathbf{1}$$

$$1 + 1 = 1\mathbf{0}$$

El biestal T puede ser empleado para diseñar lo que se denomina semisumador que tiene como requerimiento poder recibir en sus entradas dos dígitos binarios y en sus salidas tiene que generar dos dígitos binarios. De estos dos dígitos, uno es el dígito del producto de la suma y el otro es el dígito de acarreo.

Suponga el diseño de un circuito digital semisumador capaz de realizar la operación aritmética para los números binarios 101 y 110, realizando esta operación como cualquier suma se tiene:

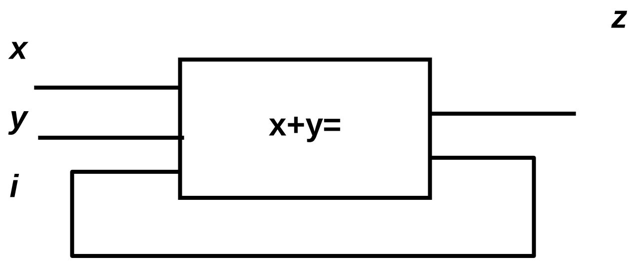
$$\begin{array}{r} 10\mathbf{1} \\ +\,11\mathbf{0} \\ \hline 101\mathbf{1} \end{array}$$

En determinado momento, de la suma se aplicó el bit de acarreo, es decir:

$$\begin{array}{lr} \overrightarrow{bit\ de\ acarreo} & \\ \square & 1 \\ \square & 010\mathbf{1} \\ \square & +\,011\mathbf{0} \\ \square & \\ \hline & 101\mathbf{1} \end{array}$$

Como se sabe, el cero a la izquierda no tiene valor, se agrega al ejemplo gráfico para entender cómo el bit de acarreo que es uno, baja hasta el resultado, es decir la suma de 1+0 =1.

Gráficamente, en una máquina sumadora suponga que el número binario 101 está representado por x y la posición de sus elementos por el lugar que ocupan éstos, es decir, x_0, x_1, x_2, para el número binario 110 se representa con y, la posición de sus elementos por y_0,y_1, y_2, el resultado de la operación como z, y la letra i representa el bit de acarreo.



En la primera interación las variables de entrada se inicializan como x=1, y=0, i=0 por no haber bit de acarreo, dando como resultado z=1; en la segunda iteración x=0, y=1, i=0 por no haber bit de acarreo, z=1; la tercera iteración es igual a x=1, y=1, i=1 por haber bit de acarreo, z=0; es necesaria una cuarta iteración donde x=0, y=0 y z=1 que equivale al valor que estaba almacenado en i, dando como resultado z=101.

4.5 Registros

Los registros en lógica combinacional son circuitos digitales que están conectados entre sí por medio de flip-flops creando un circuito secuencial en cascada.

Los registros se aplican en circuitos digitales que requieren transmitir y recibir datos, ya sea en serie o paralelo, también en la conversión de los datos para ser enviados en serie o paralelo, y al igual que los demás biestales, tienen la utilidad de almacenar información.

Los registros en serie o paralelo se clasifican de acuerdo con su función; por ejemplo: los registros de desplazamiento de tipo entrada serie y salida serie son los más sencillos, ya que como se reciben los datos en serie y salen en serie, lo único a considerar es el retardo que tenga la salida por la forma en que se encuentren interconectados los flip-flops.

Por su tipo de topología es fácil entender que en los registros de entrad, serie, salida en paralelo tienen que haber ciertos retardos en el circuito digital para poder almacenar los bits que produzca una misma salida a la vez para que se dé el caso de una salida en paralelo.

Los registros de entrada paralelo y salida serie tienen que forzar a los bits de entrada a generar retardos para que los datos binarios vayan saliendo uno a uno como es la transmisión en serie.

ACTIVIDADES DE APRENDIZAJE

1. Realizar un resumen sobre los registros de contadores, así como su funcionamiento y sus aplicaciones.

AUTOEVALUACIÓN

Conteste las siguientes preguntas:

1. ¿Qué es un elemento biestal?

2. ¿Cuáles son sus características?

3. ¿Por qué se les denomina flip-flops?

4. ¿Por qué se les denomina flip-flops asíncronos?

5. ¿Por qué se les denomina flip-flops síncronos?

6. Dibuje el esquema básico de un flip-flop.

7. ¿Qué es un contador?

8. ¿Qué es un registro?

RESPUESTAS

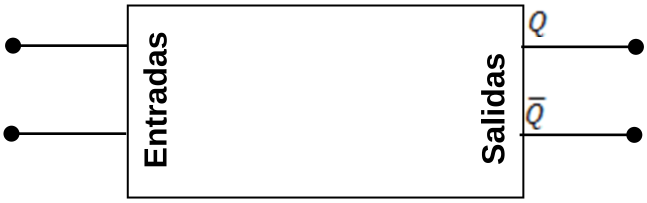
1. Los circuitos biestales son circuitos digitales capaces de almacenar un bit de información durante un tiempo indeterminado. De ahí la utilidad que se les da como elementos de memoria en un circuito electrónico.

2. Dentro de sus características, estos circuitos pueden ser utilizados como circuitos de memoria debido a su característica principal de almacenar un bit de datos sin que éste se pierda en un determinado tiempo.

3. Se les denomina flip-flop ya que poseen dos estados en los que su comportamiento es estable, el estado de nivel alto y de nivel bajo. A estos estados, en la terminología electrónica en el campo práctico, se les denomina como 1 y 0 lógicos.

4. Se les denomina flip-flops asíncronos a aquellos que sólo tienen entradas de control.

5. Se les denomina flip-flops síncronos a aquellos que además de tener entradas tienen una entrada extra que proviene de un reloj.



6. Es un dispositivo que lleva el control de las operaciones ejecutadas por la unidad aritmética y lógica; su función principal es la de contabilizar las operaciones realizadas en un tiempo dado.

7. Son circuitos digitales que se encuentran conectados entre sí, por medio de flip-flops creando un circuito secuencial en cascada.

UNIDAD 5

DISPOSITIVOS DE MEMORIA

OBJETIVO

Analizar los conceptos de la memoria aplicables a los circuitos digitales y su funcionamiento interno.

TEMARIO

5.1 TERMINOLOGÍA DE MEMORIA

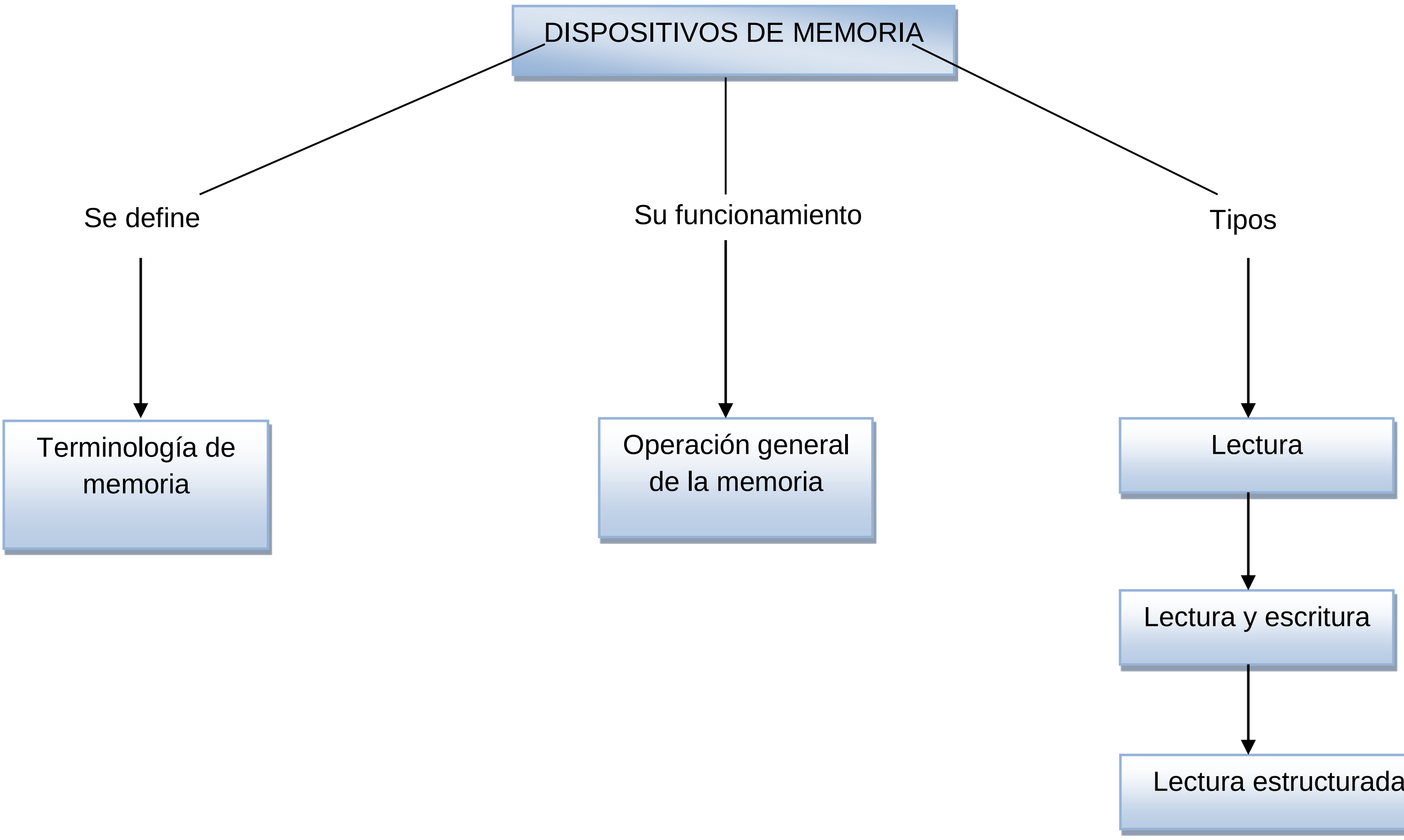
5.2 OPERACIÓN GENERAL DE LA MEMORIA

5.3 MEMORIA DE SÓLO LECTURA

5.4 MEMORIA DE LECTURA Y ESCRITURA

5.5 MEMORIA DE LECTURA ESTRUCTURADA

MAPA CONCEPTUAL



INTRODUCCIÓN

En esta unidad se abordará el tema de memoria. Se definirá la memoria de manera general, además de enfocar los elementos que la componen y sus diferentes aplicaciones en el mundo digital.

Se definirán los diferentes tipos de memorias que se han creado a partir de las lógicas digitales, además se definirá el tipo de almacenamiento que éstas utilizan y también el tipo de funcionamiento que emplean para realizar su funcionamiento.

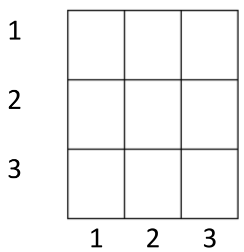
5.1 TERMINOLOGÍA DE MEMORIA

La memoria es la parte de un sistema que almacena datos binarios en grandes cantidades. La regla general de almacenamiento de bits de una memoria está dada por aquella que tiene la capacidad de almacenar datos que se encuentren en unidades que van desde un bit hasta 8 bits. El lenguaje nativo de las computadoras es el sistema binario que contiene como unidad mínima al bit que equivale a un pulso de reloj en el valor de 0 o 1.

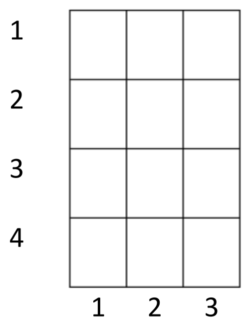
Los datos binarios se almacenan en grupos de ocho bits que forman un byte. Dicho byte se puede subdividir en dos grupos, es decir, cada grupo contiene cuatro bits al que se le denomina nibbles. Para transmisión de datos algunas memorias almacenan grupos de nueve bits por la necesidad de colocar el bit de paridad. Como es sabido, este bit de paridad lo emplean los códigos de transmisión para encriptar, desencriptar y reestructurar el mensaje que se envía bajo códigos como el Hamming o CRC.

Cada elemento de almacenamiento que conforma una memoria es capaz de almacenar un bit con el valor de 1 o 0, al elemento que funge como almacén en la memoria se le denomina celda.

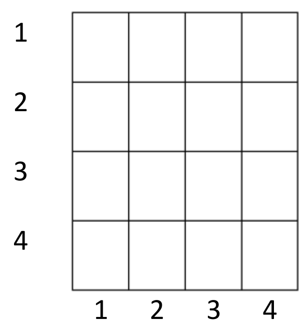
Un conjunto de celdas agrupadas forman una matriz de celdas y dicha matriz es la que conforma la memoria. La matriz que compone la memoria, al igual que una matriz de álgebra vectorial, está compuesta por filas y columnas. Entre más amplia sea la matriz de la memoria ofrecerá mayor capacidad de almacenamiento. En programación, el estudiante está familiarizado con el concepto de array y vector que tiene la misma representación gráfica que una matriz de memoria como la que se ilustra a continuación.



Matriz de 3 x 3



Matriz de 4 x 3



Matriz de 4 x 4

El producto de las filas por columnas que conforman una matriz da como resultado el número de celdas que contiene la memoria. Para el primer caso, nuestra matriz es de tres filas por tres columnas, el número de celdas es equivalente a 9, la siguiente matriz contiene 12 y la tercera 16 celdas.

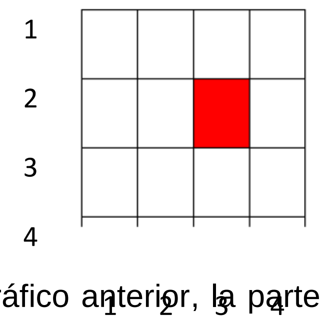
Estos ejemplos gráficos almacenan 9, 12 y 16 bits. Dimensionemos estos ejemplos a las memorias RAM de 16, 32, 168, 128, 256 y 512 megabytes, o sea menos unidades de capacidad de memoria de las que maneja la computadora, ya que ella puede manipular millones de bytes de almacenamiento en sus operaciones cuando interactúa con las personas.[8]

ACTIVIDADES DE APRENDIZAJE

1. Realizar una investigación sobre la terminología de memoria y elaborar una síntesis.

5.2 OPERACIÓN GENERAL DE LA MEMORIA

La matriz de memoria compuesta por las filas y columnas da paso a las celdas, que contienen o pueden contener 1 bit almacenado ya sea 1 o 0, pero es importante que se conozca el lugar donde se almacena dicho bit. A este término se le conoce como dirección de memoria.



Tomando de referencia el gráfico anterior, la parte sombreada ocupa la dirección de memoria (2,3) la cual contiene un bit almacenado.

El número de bits que puede almacenar una memoria se le denomina capacidad de almacenamiento.

Para llevar a cabo el proceso de almacenamiento, las memorias tienen que tener la capacidad de realizar dos operaciones sencillas que son

[8] *Ibidem.*, p. 733.

la de escritura para colocar los bits en una dirección de memoria y para saber el valor que se encuentra almacenado, se procede a la operación de lectura.

5.3 MEMORIA DE SÓLO LECTURA.

En los circuitos digitales las memorias de sólo lectura ROM (Read Only Memory), son aquellas memorias que no son del tipo volátil. Es decir que al carecer de una fuente de alimentación eléctrica no pierden sus datos, por consiguiente puede seguir siendo utilizada la información que en ellas se contiene cuantas veces se desee, siempre y cuando el circuito digital tenga una fuente de alimentación eléctrica.

En la aplicación de las memorias ROM en las computadoras mencionaremos como ejemplo la memoria EPROM es aquella que contiene el programa de arranque de la tarjeta madre denominado SETUP y las instrucciones de configuración de encendido de todo el hardware al que se le denomina BIOS.

Aunque el término de memoria ROM significa que sólo se pude obtener la lectura de los datos escritos en la memoria, cabe señalar que a esta memoria se le tuvo que escribir dichos datos, pero este proceso de escritura es un poco lento y requiere de otros dispositivos electrónicos para poder realizar el proceso de grabación, además de una fuente alta de energía eléctrica.[9]

ACTIVIDADES DE APRENDIZAJE

1. Realizar una investigación sobre el tema de la memoria de sólo lectura y elaborar un resumen.

5.4 MEMORIA DE LECTURA Y ESCRITURA

En circuitos digitales, a las memorias de lectura y escritura se les denomina con el término de RAM (Random Access Memory), memoria de acceso aleatorio, que es una memoria de lectura y escritura de datos binarios que se pueden leer y escribir en cualquier orden.

---

[9] *Ibidem.*, p. 757.

Una vez almacenados los datos en la memoria RAM, quedan guardados ahí hasta que son requeridos por el procedimiento. A este proceso se le conoce como escritura.

El proceso de lectura es aquel en el cual los datos son leídos y borrados para liberar el espacio de memoria y que dicha dirección quede libre en caso de que para otro proceso, sea necesario ocupar ese espacio para almacenar otros datos.

Se dice que las memorias RAM son memorias de tipo volátil, a diferencia de la memoria ROM que al carecer de una fuente de energía, mantiene los datos almacenados en ellas. Al carecer la memoria RAM de la fuente de energía, los datos que en ella se encontraban son eliminados por no tener la capacidad de almacenamiento permanente.

Las memorias RAM utilizan flip-flops como elementos de almacenamiento. Mientras cuenten con una fuente de energía continua y estable, dichos datos pueden permanecer en la memoria RAM por tiempo indefinido.

5.5 MEMORIA DE LECTURA ESTRUCTURADA

La distinción entre las memorias SRAM y DRAM es la capacidad y velocidad de almacenamiento de los datos binarios. La memoria DRAM tiene gran capacidad de almacenamiento a diferencia de la memoria SRAM.

Las memorias SRAM se clasifican en memorias SRAM asíncronas y síncronas. Las memorias SRAM asíncronas son aquellas que dentro de los sistemas digitales no se encuentran sincronizadas con el reloj del sistema digital. La diferencia de la memoria SRAM síncrona con respecto a la SRAM asíncrona es que esta memoria sí se encuentra sincronizada con el reloj del sistema digital.

Las memorias DRAM se emplean principalmente en los circuitos de las computadoras personales por tener una mayor capacidad de almacenamiento a diferencia de la memoria SRAM. La estructura digital de una memoria DRAM cuenta además con flip-flops, transistores y condensadores haciéndola un tipo de memoria más versátil.

AUTOEVALUACIÓN

Conteste los siguientes cuestionamientos:

1. ¿Qué es una memoria?

2. ¿Cómo funciona una memoria?

3. ¿Cómo almacena datos la memoria internamente?

4. ¿Qué tipos de memoria son de acceso aleatorio?

5. ¿Qué diferencias hay entre memoria de lectura y la de lectura y escritura?

6. Menciona al menos tres ejemplos de memorias que conozcas.

RESPUESTAS

1. La memoria es la parte de un sistema que almacena datos binarios en grandes cantidades.

2. El funcionamiento de la memoria se rige por una regla general para el almacenamiento de bits dada por la capacidad de almacenar datos que se encuentren en unidades que van desde un bit hasta ocho bits.

3. Los datos a almacenar los agrupa en datos binarios y se almacenan en grupos de ocho bits que forman un byte. Dicho byte se puede subdividir en dos grupos, es decir, cada grupo contiene cuatro bits al que se le denomina nibbles.

4. Las memorias RAM se consideran de acceso aleatorio debido a que los procesos cargados en ellas se van colocando como se van creando y ejecutando. A estas memorias se les conoce como acceso aleatorio.

5. Las memorias de sólo lectura, como su nombre lo afirma, son memorias que sólo pueden ser leídas y no modificadas, y las de lectura y escritura son aquellas que pueden ser leídas y a su vez pueden ser modificadas.

6. Memorias RAM, memorias ROM, memorias EPROM, Memorias SRAM y DRAM.

BIBLIOGRAFÍA

Boylestad, Robert y Nachelsky, Louis, *Electrónica: teoría de circuitos y dispositivos electrónicos,* México, Pearson Educación, 2003

Gómez Caño, María José y Floyd, Thomas L., *Fundamentos de sistemas digitales,* España, Prentice Hall, 2000

Tocci, Roland y Widmer, Neal, *Sistemas digitales: principios y aplicaciones*, México, Pearson Educación, 2003

Bibliografía complementaria

Katsuhiko, Ogata, *Ingeniería de control moderna,* México, Pearson Educación, 2003

Proakis, John G. y Manolakis, Dimitris G., *Tratamiento digital de señales,* España, Prentice Hall, 1999

Rashid, Muhammad, *Electrónica de potencia*, 3ra edición, México, Pearson Educación, 2004

GLOSARIO[10]

Bit = Un bit es una señal electrónica que puede estar encendida (1) o apagada (0). Es la unidad más pequeña de información que utiliza un ordenador. Son necesarios ocho bits para crear un byte.

Byte = Un byte es la unidad fundamental de datos en los ordenadores personales, un byte son ocho bits contiguos. El byte es también la unidad de medida básica para memoria, almacenando el equivalente a un carácter.

Circuito Integrado = Un circuito integrado es una pastilla (o "chip") muy delgada en la que se encuentran miles o millones de dispositivos electrónicos interconectados.

Matriz = Es una tabla bidimensional de números consistente en cantidades abstractas que pueden sumarse y multiplicarse.

[10] Boylestad, Robert y Nachelsky, Louis, *Electrónica: teoría de circuitos y dispositivos electrónicos,* México, Pearson Educación, 2003, Tocci, Roland y Widmer, Neal, *Sistemas digitales: principios y aplicaciones*, México, Pearson Educación, 2003.