



UNIREMINGTON®
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

ARQUITECTURA DE COMPUTADORES
INGENIERÍA DE SISTEMAS
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA

Vicerrectoría de Educación a Distancia y virtual

2016



El módulo de estudio de la asignatura ARQUITECTURA DE COMPUTADORES es propiedad de la Corporación Universitaria Remington. Las imágenes fueron tomadas de diferentes fuentes que se relacionan en los derechos de autor y las citas en la bibliografía. El contenido del módulo está protegido por las leyes de derechos de autor que rigen al país.

Este material tiene fines educativos y no puede usarse con propósitos económicos o comerciales.

AUTOR

Giovanny Alberto Flórez Osorio

Ingeniero Electrónico de la Universidad de Antioquia. Curso Virtual de Tecnologías web 2.0 para la docencia. Universidad Autónoma de Bucaramanga. Diplomado en Diseño Curricular y materiales de autoaprendizaje. Corporación Universitaria Remington. Diplomado en Metodología general de investigación. Corporación Universitaria Remington. Diplomado en Pedagogía universitaria. Corporación Universitaria Remington. Diplomado en Didáctica y currículo. Corporación Universitaria Remington. Curso Inducción a la metodología de educación a distancia. Corporación Universitaria Remington. Docente del Colegio Cooperativo Cacique Bitagüí. Docente del CENDI. Docente de la Corporación Universitaria Remington.

giovanny.florez@uniremington.edu.co

Nota: el autor certificó (de manera verbal o escrita) No haber incurrido en fraude científico, plagio o vicios de autoría; en caso contrario eximió de toda responsabilidad a la Corporación Universitaria Remington, y se declaró como el único responsable.

RESPONSABLES

Jorge Mauricio Sepúlveda Castaño

Decano de la Facultad de Ciencias Básicas e Ingeniería

jsepulveda@uniremington.edu.co

Eduardo Alfredo Castillo Builes

Vicerrector modalidad distancia y virtual

ecastillo@uniremington.edu.co

Francisco Javier Álvarez Gómez

Coordinador CUR-Virtual

falvarez@uniremington.edu.co

GRUPO DE APOYO

Personal de la Unidad CUR-Virtual
EDICIÓN Y MONTAJE

Primera versión. Febrero de 2011.
Segunda versión. Marzo de 2012
Tercera versión. noviembre de 2015
Cuarta versión 2016

Derechos Reservados



Esta obra es publicada bajo la licencia Creative Commons.
Reconocimiento-No Comercial-Compartir Igual 2.5 Colombia.

TABLA DE CONTENIDO

	Pág.
1 MAPA DE LA ASIGNATURA	6
2 unidad 1 CIRCUITOS LÓGICOS Y COMPONENTES DIGITALES	7
2.1.1 RELACIÓN DE CONCEPTOS.....	8
2.2 Tema 1 Compuertas Digitales	10
2.2.1 Ejercicios de Aprendizaje	14
2.3 Tema 2 Álgebra Booleana	22
2.4 Tema 3 Simplificación de expresiones algebraicas por medio de teoremas y de mapas de Karnaugh	24
2.4.1 Ejercicios de Aprendizaje	24
2.4.2 Ejercicio de Aprendizaje.....	28
2.4.3 Ejercicios de Aprendizaje.....	33
2.5 Tema 5 Circuitos Integrados Digitales.....	35
2.6 Tema 6 Codificador	38
2.6.1 Ejercicio de Aprendizaje.....	38
2.6.2 Ejercicios de Aprendizaje	40
2.7 Tema 8 Multiplexor	43
2.7.1 Ejercicio de Aprendizaje.....	43
2.8 Tema 9 Demultiplexor.....	45
2.8.1 Ejercicio de Aprendizaje.....	45
2.9 Tema 10 Registro	47
2.9.1 Ejercicios de Entrenamiento	47
3 UNIDAD 2 ELEMENTOS FUNCIONALES DE UN COMPUTADOR	51
3.1.1 Relación de conceptos	51

3.2	Tema 1 El Computador.....	52
3.3	Tema 2 Tipos de Computadores	52
3.4	Tema 3 Unidad Central de Procesamiento	54
3.4.1	Unidad Aritmético Lógica	55
3.4.2	Unidad de Control.....	56
3.5	Tema 4 Unidad de Memoria	57
3.6	Tema 5 Unidad de Entrada y Salida	58
3.7	Tema 6 Buses	59
3.7.1	Ejercicios de Aprendizaje.	60
3.7.2	Ejercicios de Entrenamiento	62
4	UNIDAD 3 TIPOS DE ARQUITECTURAS.....	63
4.1.1	RELACIÓN DE CONCEPTOS.....	63
4.2	Tema 1 Arquitecturas ISA.....	64
4.2.1	Arquitecturas de Pila.	65
4.2.2	Arquitecturas de Acumulador.....	65
4.2.3	Arquitecturas de Registros de Propósito General	66
4.3	Tema 2 Arquitecturas Registro - Registro	66
4.3.1	Ejercicio de Aprendizaje.....	66
a)	Arquitectura de Pila.....	67
b)	Arquitectura de Acumulador.....	67
c)	Arquitectura de Registros de Propósito General.....	67
4.3.2	Ejercicios de Entrenamiento	68
5	PISTAS DE APRENDIZAJE	70
6	GLOSARIO	71

7	BIBLIOGRAFÍA	74
---	--------------------	----

1 MAPA DE LA ASIGNATURA

ARQUITECTURA DE COMPUTADORES

PROPÓSITO GENERAL DEL MÓDULO

Busca establecer las bases teóricas y prácticas de las Arquitecturas de los computadores para desarrollar la capacidad de entender, analizar y aplicar los conceptos adquiridos.

OBJETIVO GENERAL

Aplicar los conceptos de: Circuitos lógicos y componentes digitales, elementos funcionales de un computador y tipos de arquitecturas enfocados en el funcionamiento de los computadores y su interfaz con el mundo exterior.

OBJETIVOS ESPECÍFICOS

- Aplicar los conocimientos de las compuertas lógicas y los componentes digitales en la resolución de problemas de circuitos lógicos.
- Ilustrar el funcionamiento de cada una de las unidades que componen al computador.
- Demostrar por medio de la práctica el uso de las arquitecturas del computador.

UNIDAD 1

CIRCUITOS LÓGICOS Y COMPONENTES DIGITALES

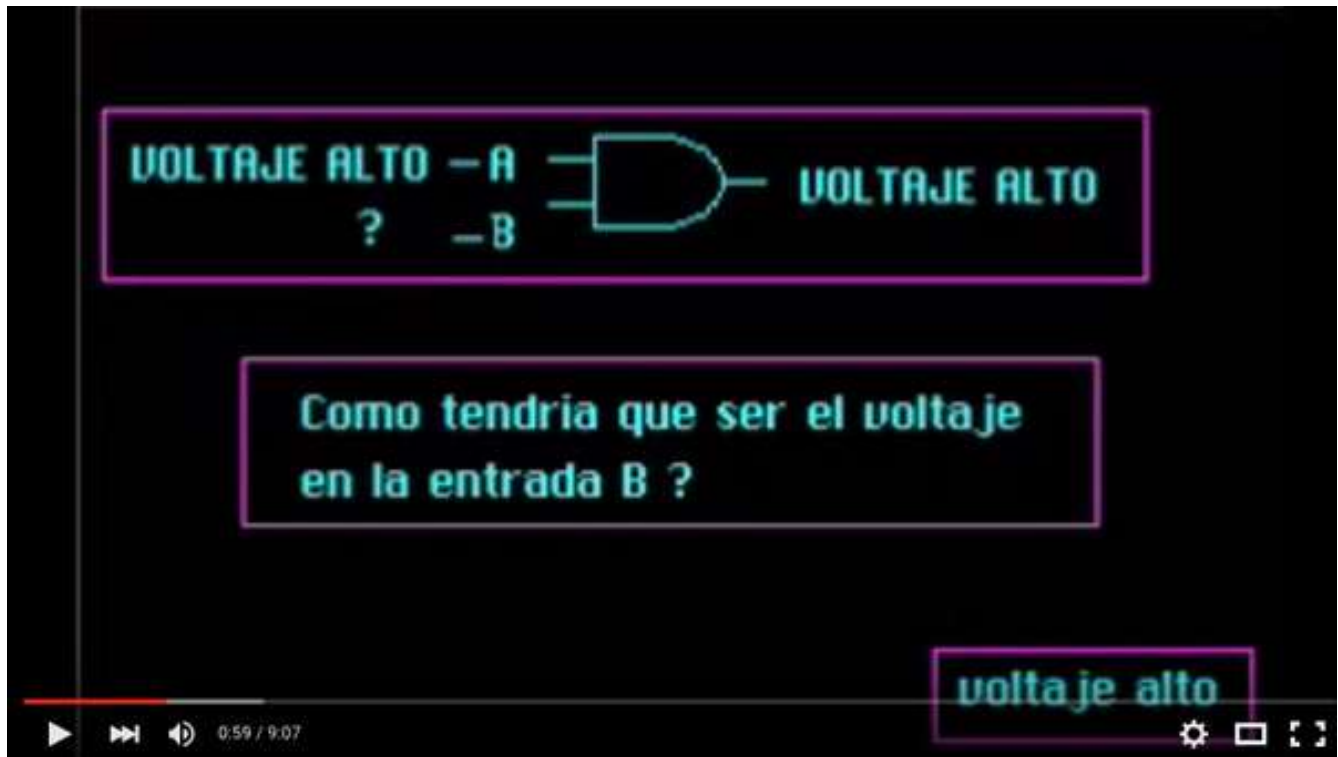
UNIDAD 2

ELEMENTOS FUNCIONALES DE UN COMPUTADOR

UNIDAD 3

TIPOS DE ARQUITECTURAS

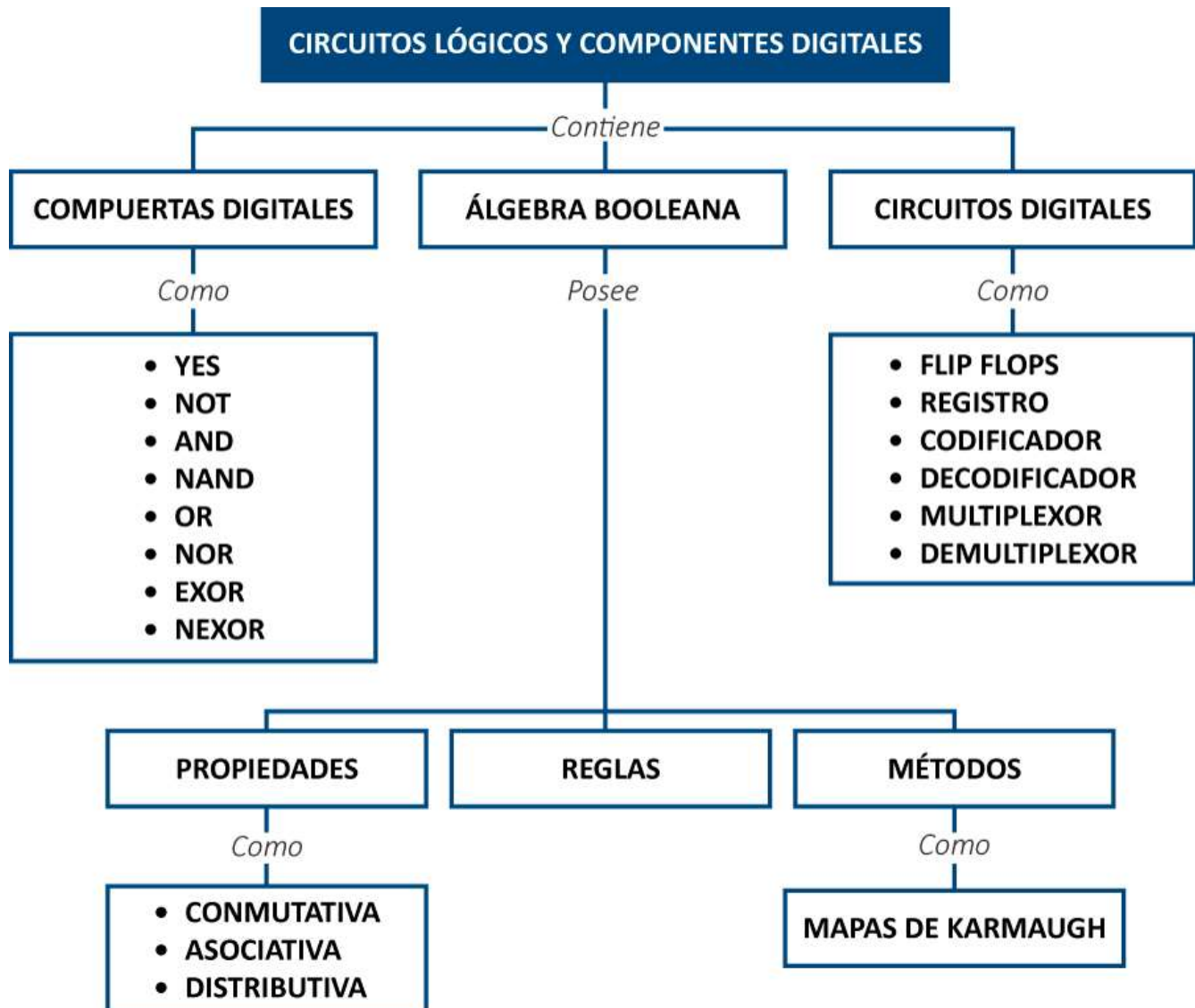
2 UNIDAD 1 CIRCUITOS LÓGICOS Y COMPONENTES DIGITALES



The screenshot shows a video player interface. The main content area has a black background with green text and a logic gate diagram. At the top, a box contains the text "VOLTAGE ALTO - A" and "? - B" on the left, an AND gate symbol in the center, and "VOLTAGE ALTO" on the right. Below this, another box asks "Como tendria que ser el voltaje en la entrada B ?". At the bottom right, a small box contains the text "voltage alto". The video player controls at the bottom show a play button, a progress bar at 0:59 / 9:07, and icons for settings, full screen, and share.

Curso de Electrónica Digital: Compuertas Lógicas [Enlace](#)

2.1.1 RELACIÓN DE CONCEPTOS



Álgebra booleana: Son las Matemáticas de los Sistemas digitales. Utiliza variables lógicas (Toman valores de 0 ó 1) y operadores lógicos (Yes, Not, And, Or, Nand, Nor, Exor, Nexor).

Circuito lógico (o digital): Es aquel que maneja la información utilizando el sistema binario (0 y 1).

Codificador: Es un circuito combinacional. Se obtiene en la salida el código binario correspondiente a la entrada que se activa. (Colocada en uno).

Componentes digitales: Son aquellos dispositivos que forman parte de un circuito electrónico digital.

Compuertas digitales: Son funciones básicas de los sistemas electrónicos digitales; trabajan con números binarios.

Compuerta digital AND: Es aquella que realiza la operación denominada multiplicación lógica.

Compuerta digital EXOR: Es aquella que realiza la operación de comparación (Dos entradas). Se obtiene un nivel alto en la salida cuando las entradas son diferentes.

Compuerta digital NAND: Es aquella que realiza la operación inversa de la compuerta AND. Se puede utilizar como compuerta universal ya que se pueden combinar varias de éstas compuertas para implementar el resto de las compuertas lógicas básicas (Por ejemplo: NOT, AND, OR, EXOR, etc.)

Compuerta digital NEXOR: Es aquella que realiza la operación de comparación (Dos entradas). Se obtiene un nivel alto en la salida cuando las entradas son iguales.

Compuerta digital NOR: Es aquella que realiza la operación inversa de la compuerta OR.

Compuerta digital NOT: Es aquella que invierte el valor de la entrada (Operación de negación o complemento)

Compuerta digital OR: Es aquella que realiza la operación denominada suma lógica.

Compuerta digital YES: Es aquella que amplifica la señal. Se obtiene en la salida el mismo bit de entrada.

Decodificador: Es un circuito combinacional. Dado el código binario en la entrada se activa (Coloca en uno) la salida correspondiente a dicho código.

Demultiplexor: Es un circuito combinacional que permite seleccionar a cuales de las salidas (mediante selectores) pasará la entrada.

Flip Flop: Es un circuito secuencial que está conformado por compuertas lógicas. Se conoce como un dispositivo de memoria.

Mapas de Karnaugh: Método sistemático de simplificación de expresiones lógicas.

Método: Modo ordenado y sistemático de proceder para llegar a un resultado o fin determinado.

Multiplexor: Es un circuito combinacional que permite seleccionar de las entradas (mediante selectores) cuál de ellas pasará a la salida.

Propiedades: Son reglas que se obtienen a partir de los axiomas y deben ser demostradas mediante estos.

Propiedad asociativa: Enuncia que el resultado en la salida es el mismo, independiente de la forma en que se agrupen las variables de entrada.

Propiedad conmutativa: Enuncia que al cambiar el orden de las variables de entrada, permanece invariable el resultado que se obtiene en la salida.

Propiedad distributiva: Enuncia la relación entre la multiplicación lógica y la suma lógica.

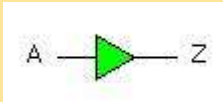
Registro: Es un dispositivo electrónico semiconductor creado a partir de implementaciones con flip flops los cuales permiten el manejo de datos usando circuitos combinatorios y secuenciales específicos.

Reglas: Son los métodos para llevar a cabo una operación (En este caso, operación lógica).

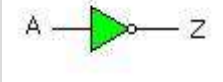
2.2 TEMA 1 COMPUERTAS DIGITALES

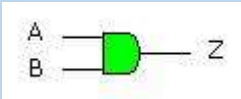
Las **compuertas digitales** **son funciones básicas** de los **sistemas electrónicos digitales**. Estos **componentes** trabajan con **números binarios**.

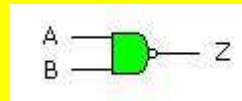
Las **compuertas digitales básicas** son:

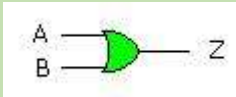
Compuerta YES (Si)														
Número de entradas	Funcionamiento	Símbolo	Tabla de estados	Expresión lógica										
Una	La salida es igual a la entrada.	 Autoría propia	<table><tr><th>ENTRADA</th><th>SALIDA</th></tr><tr><th>A</th><th>Z</th></tr><tr><td> </td><td> </td></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	ENTRADA	SALIDA	A	Z			0	0	1	1	$Z = A$
ENTRADA	SALIDA													
A	Z													
0	0													
1	1													

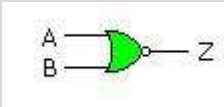
Compuerta NOT (Inversora, negación)						
Número de entradas	Funcionamiento	Símbolo	Tabla de estados	Expresión lógica		
			<table><tr><th>ENTRADA</th><th>SALIDA</th></tr></table>	ENTRADA	SALIDA	
ENTRADA	SALIDA					

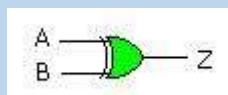
Una	La salida es la negación de la entrada .	 Autoría propia	A	Z	$Z = \bar{A}$
			0	1	
			1	0	

Compuerta AND (Y, Producto lógico)																									
Número de entradas	Funcionamiento	Símbolo	Tabla de estados	Expresión lógica																					
Mínimo dos	La salida es 1 cuando todas las entradas sean 1 .	Para dos entradas se tiene:  Autoría propia	<table><tr><th colspan="2">ENTRADAS</th><th>SALIDA</th></tr><tr><th>A</th><th>B</th><th>Z</th></tr><tr><td></td><td></td><td></td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	ENTRADAS		SALIDA	A	B	Z				0	0	0	0	1	0	1	0	0	1	1	1	$Z = AB$
ENTRADAS		SALIDA																							
A	B	Z																							
0	0	0																							
0	1	0																							
1	0	0																							
1	1	1																							

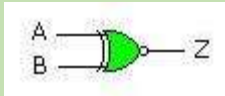
Compuerta NAND (Negación de la AND)																									
Número de entradas	Funcionamiento	Símbolo	Tabla de estados	Expresión lógica																					
Mínimo dos	La salida es 1 cuando al menos una de las entradas sea 0 .	Para dos entradas se tiene:  Autoría propia	<table><tr><th colspan="2">ENTRADAS</th><th>SALIDA</th></tr><tr><th>A</th><th>B</th><th>Z</th></tr><tr><td></td><td></td><td></td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	ENTRADAS		SALIDA	A	B	Z				0	0	1	0	1	1	1	0	1	1	1	0	$Z = \overline{AB}$
ENTRADAS		SALIDA																							
A	B	Z																							
0	0	1																							
0	1	1																							
1	0	1																							
1	1	0																							

Compuerta OR (O, Suma lógica)																						
Número de entradas	Funcionamiento	Símbolo	Tabla de estados	Expresión lógica																		
Mínimo dos	La salida es 1 cuando al menos una de las entradas sea 1 .	Para dos entradas se tiene:  Autoría propia	<table><tr><th colspan="2">ENTRADAS</th><th>SALIDA</th></tr><tr><th>A</th><th>B</th><th>Z</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	ENTRADAS		SALIDA	A	B	Z	0	0	0	0	1	1	1	0	1	1	1	1	$Z = A + B$
ENTRADAS		SALIDA																				
A	B	Z																				
0	0	0																				
0	1	1																				
1	0	1																				
1	1	1																				

Compuerta NOR (Negación de la OR)																						
Número de entradas	Funcionamiento	Símbolo	Tabla de estados	Expresión lógica																		
Mínimo dos	La salida es 1 cuando todas las entradas sean 0 .	Para dos entradas se tiene:  Autoría propia	<table><tr><th colspan="2">ENTRADAS</th><th>SALIDA</th></tr><tr><th>A</th><th>B</th><th>Z</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	ENTRADAS		SALIDA	A	B	Z	0	0	1	0	1	0	1	0	0	1	1	0	$Z = \overline{A + B}$
ENTRADAS		SALIDA																				
A	B	Z																				
0	0	1																				
0	1	0																				
1	0	0																				
1	1	0																				

Compuerta EXOR (Or Exclusiva)													
Número de entradas	Funcionamiento	Símbolo	Tabla de estados	Expresión lógica									
Solo dos	La salida es 1 cuando las entradas sean diferentes.		<table><tr><th colspan="2">ENTRADAS</th><th>SALIDA</th></tr><tr><th>A</th><th>B</th><th>Z</th></tr><tr><td>0</td><td>0</td><td>0</td></tr></table>	ENTRADAS		SALIDA	A	B	Z	0	0	0	$Z = A \oplus B$
ENTRADAS		SALIDA											
A	B	Z											
0	0	0											

		Autoría propia	<table><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	0	1	1	1	0	1	1	1	0	
0	1	1											
1	0	1											
1	1	0											

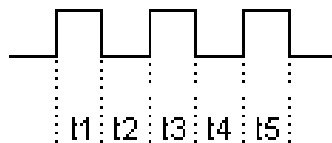
Compuerta NEXOR (Negación de la Or Exclusiva)																						
Número de entradas	Funcionamiento	Símbolo	Tabla de estados	Expresión lógica																		
Solo dos	La salida es 1 cuando las entradas sean iguales.	 Autoría propia	<table><tr><th colspan="2">ENTRADAS</th><th>SALIDA</th></tr><tr><th>A</th><th>B</th><th>Z</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	ENTRADAS		SALIDA	A	B	Z	0	0	1	0	1	0	1	0	0	1	1	1	$Z = \overline{A \oplus B}$
ENTRADAS		SALIDA																				
A	B	Z																				
0	0	1																				
0	1	0																				
1	0	0																				
1	1	1																				

Señal digital

Se **compone** de un **tren de impulsos**. Puede ser:

Periódica.

También se llama **onda cuadrada**. Ésta **señal** se **repite** a **intervalos de tiempo fijo (Período)**.



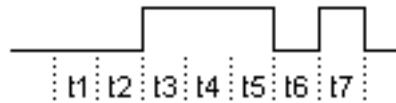
Autoría propia

La **duración** del **bit** se representa como t_b

No periódica.

La **señal** **no** se **repite** a **intervalos de tiempos fijos**. Además el **ancho** de los **impulsos es distinto**.

Por ejemplo:



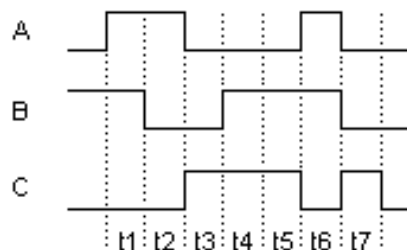
Autoría propia

Se pueden **diseñar y construir** muchos tipos de **circuitos** con el solo hecho de **combinar** las **compuertas lógicas básicas**, entre ellos se tiene:

- **Amplificación** de la **señal digital** con compuerta **YES**.
- **Oscilador** con compuertas **NOT** y **capacitor**.
- **Alarma** con compuertas **NAND**.
- **Circuito** que genera el complemento a **1** utilizando compuertas **NOT**.
- Compuerta **AND** como un dispositivo de **activación / inhibición**.
- **Circuito detector de intrusos** con compuerta **OR**.

2.2.1 EJERCICIOS DE APRENDIZAJE

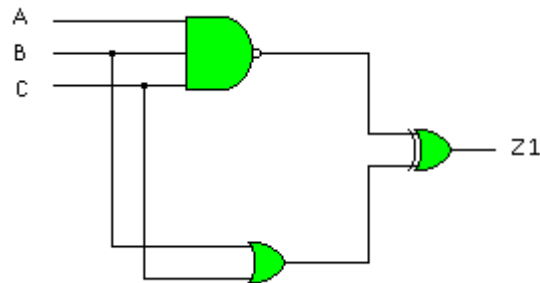
Dadas las **señales de entrada**



Autoría propia

Hallar la **señal** y la **expresión lógica de salida** para:

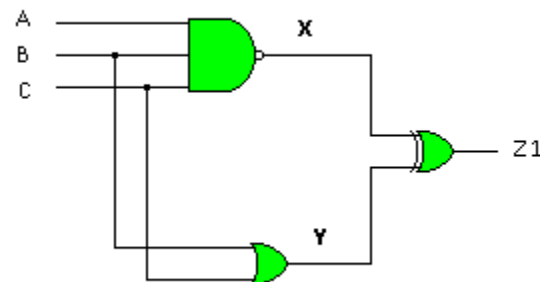
a)



Autoría propia

Solución.

Se utilizan **variables auxiliares** (X y Y) para **obtener las salidas** de las **compuertas Nand y Or**.

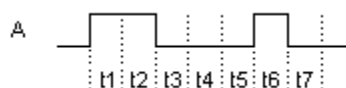


Autoría propia

Para **obtener las salidas de las compuertas Nand y Or** se **trabaja** con los **códigos binarios** correspondientes a las **señales de entrada (A, B y C)**.

De esta forma se tiene:

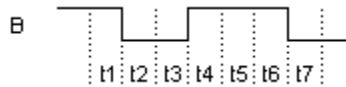
Para la **señal de entrada A**



Autoría propia

Su código binario es **1100010**

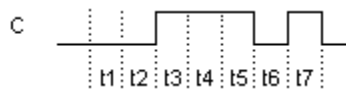
Para la señal de entrada B



Autoría propia

Su código binario es **1001110**

Para la señal de entrada C



Autoría propia

Su código binario es **0011101**

Para hallar la salida de la compuerta Nand se tiene en cuenta que la salida es **1** cuando al menos una de las entradas sea 0.

A **1100010**

B **1001110**

C **0011101**

X **1111111** (Salida de la compuerta Nand)

De igual forma se procede a determinar la salida de la compuerta Or

Para hallar la salida de la compuerta Or se tiene en cuenta que la salida es **1** cuando al menos una de las entradas sea 1.

B **1001110**

C **0011101**

Y **1011111** (Salida de la compuerta Or)

Para determinar **Z1 (Salida de la compuerta Exor)** se utilizan las **respuestas obtenidas para X y Y**.

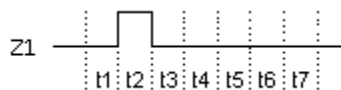
Para **hallar la salida de la compuerta Exor** se tiene en cuenta que la salida es **1 cuando las entradas sean diferentes**.

X **1111111** (Salida de la compuerta Nand)

Y **1011111** (Salida de la compuerta Or)

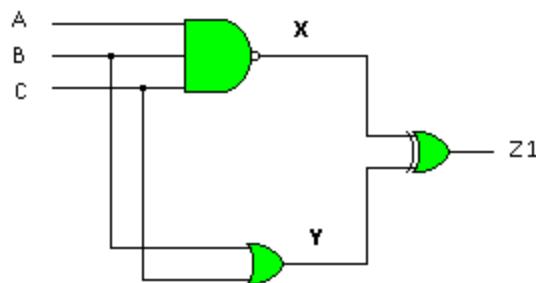
Z1 **0100000** (Salida de la compuerta Exor)

Así la **señal de salida** es



Autoría propia

Para determinar la **expresión lógica** se procede:



Autoría propia

Z1 (Salida de la compuerta Exor) se escribe como:

$$Z1 = X \oplus Y$$

Se deben encontrar las **expresiones lógicas para X y Y**

X (Salida de la compuerta Nand) se escribe como:

$$X = \overline{ABC}$$

Y (Salida de la compuerta Or) se escribe como:

$$Y = B + C$$

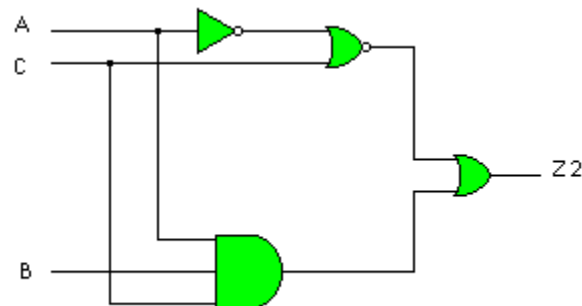
Reemplazando en Z1

$$Z1 = X \oplus Y$$

las expresiones lógicas de X y Y se obtiene:

$$Z1 = \overline{ABC} \oplus (B + C)$$

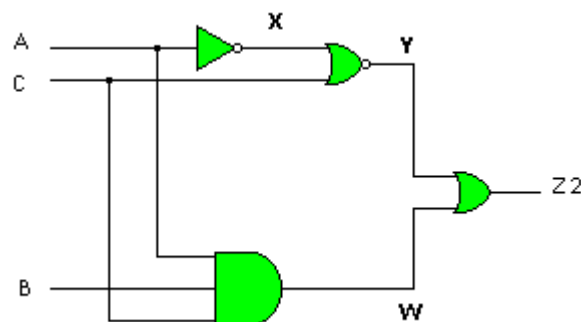
b)



Autoría propia

Solución.

Se utilizan **variables auxiliares** (X, Y y W) para **obtener las salidas** de las **compuertas Not, Nor y And**.

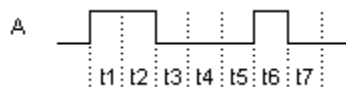


Autoría propia

Para **obtener** las **salidas** de las **compuertas Not, Nor y And** se trabajará con los **códigos binarios** correspondientes a las **señales de entrada (A, B y C)**.

De esta forma se tiene:

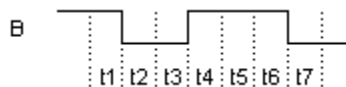
Para la **señal de entrada A**



Autoría propia

Su **código binario** es **1100010**

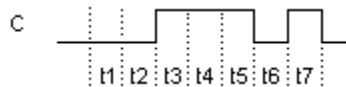
Para la **señal de entrada B**



Autoría propia

Su **código binario** es **1001110**

Para la **señal de entrada C**



Autoría propia

Su **código binario** es **0011101**

Para hallar la **salida de la compuerta Not** se tiene en cuenta que la **salida** es la **negación de la entrada**.

A **1100010**

X **0011101** (Salida de la compuerta Not)

De igual forma se procede a determinar la **salida de la compuerta And**.

Para hallar la **salida de la compuerta And** se tiene en cuenta que la salida es **1 cuando todas las entradas sean 1**.

A **1100010**

B **1001110**

C **0011101**

W **0000000** (Salida de la compuerta And)

De igual forma se procede a determinar la **salida de la compuerta Nor**.

Para hallar la **salida de la compuerta Nor** se tiene en cuenta que la salida es **1 cuando todas las entradas sean 0**.

X **0011101**

C **0011101**

Y **1100010** (Salida de la compuerta Nor)

Para determinar **Z2 (Salida de la compuerta Or)** se utilizan las **respuestas** obtenidas para **Y y W**.

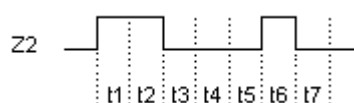
Para hallar la **salida de la compuerta Or** se tiene en cuenta que la salida es **1 cuando al menos una de las entradas sea 1**.

Y **1100010** (Salida de la compuerta Nor)

W **0000000** (Salida de la compuerta And)

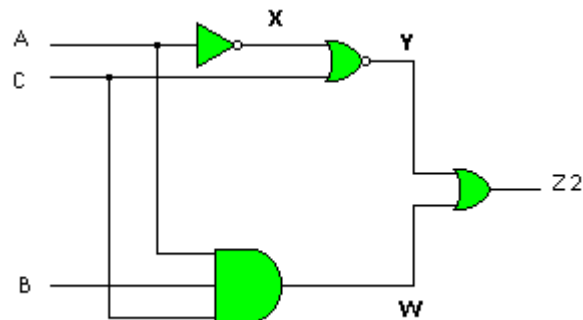
Z2 **1100010** (Salida de la compuerta Or)

Así la **señal de salida** es



Autoría propia

Para determinar la **expresión lógica** se procede:



Autoría propia

Z2 (Salida de la compuerta Or) se escribe como:

$$Z2 = Y + W$$

Se deben encontrar las **expresiones lógicas para X, Y y W.**

X (Salida de la compuerta Not) se escribe como:

$$X = \bar{A}$$

Y (Salida de la compuerta Nor) se escribe como:

$$Y = \overline{X + C}$$

W (Salida de la compuerta And) se escribe como:

$$W = ABC$$

Reemplazando en Y

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

la **expresión lógica de X** se obtiene:

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

Reemplazando en Z2

$$Z2 = Y + W$$

las **expresiones lógicas de Y y W** se obtiene:

$$Z2 = \overline{\overline{A} + \overline{C}} + ABC$$

2.3 TEMA 2 ÁLGEBRA BOOLEANA

Son las **Matemáticas** de los **Sistemas digitales**.

Al igual que en otras áreas de las **Matemáticas**, existen en el **Algebra de Boole** una serie de **leyes y reglas**.

Propiedades de las Operaciones Booleanas.

Las **operaciones booleanas** están se enfocan en **tres propiedades**:

 **Conmutativa.**

 **Asociativa.**

 **Distributiva.**

Propiedad conmutativa de la suma.

$$A + B = B + A$$

El **orden** de las **entradas** en una **compuerta OR** **no modifica la salida**.

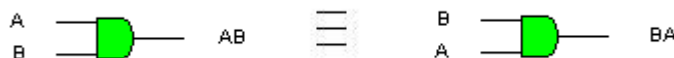


Autoría propia

Propiedad conmutativa del producto.

$$A \cdot B = B \cdot A$$

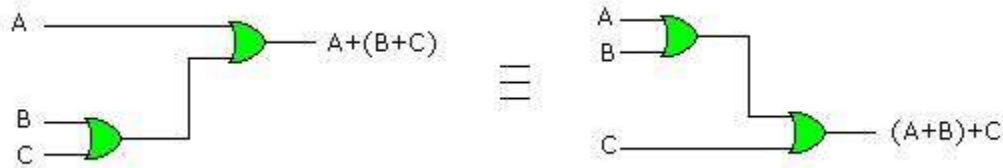
El **orden** de las **entradas** en una **compuerta AND** **no modifica la salida**.



Autoría propia

Propiedad asociativa de la suma.

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



Autoría propia

Propiedad asociativa del producto.

$$A (B \cdot C) = (A \cdot B) \cdot C$$

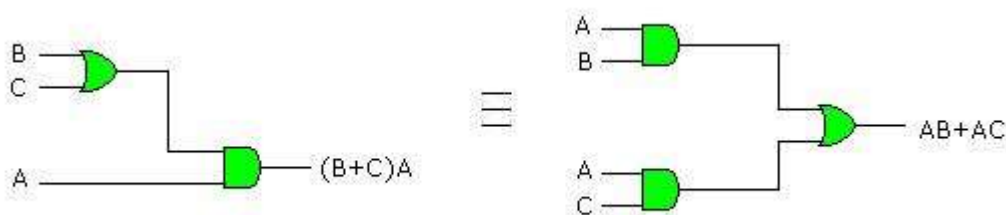


Autoría propia

Propiedad distributiva.

Se **distribuye** la(s) **variable(s)** que **multiplica(n)** a los **elementos** que hay **dentro del paréntesis**.

$$A (B + C) = AB + AC$$



Autoría propia

2.4 TEMA 3 SIMPLIFICACIÓN DE EXPRESIONES ALGEBRAICAS POR MEDIO DE TEOREMAS Y DE MAPAS DE KARNAUGH

Teoremas Booleanos.

Los **teoremas booleanos** son **reglas** que permiten simplificar **funciones lógicas complejas**.

Algunos son:

$$A + 0 = A$$

$$A + 1 = 1$$

$$A + A = A$$

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

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A \cdot A = A$$

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

$$\bar{\bar{A}} = A$$

$$A + AB = A$$

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

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

2.4.1 EJERCICIOS DE APRENDIZAJE

Simplificar utilizando el **Álgebra de Boole** las siguientes expresiones lógicas:

a) $Z = A(\bar{A} + AB)$

Solución.

Sea $Z = A(\bar{A} + AB)$

Paso 1

De acuerdo a la **propiedad distributiva**

$$A (B + C) = AB + AC$$

Se **aplica** en la **expresión inicial**

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

$$Z = A.\bar{A} + A.A.B$$

Paso 2

De acuerdo a los **siguientes teoremas**:

$$A . A = A$$

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

Se **aplican** en la **expresión** obtenida en el **paso 1**

$$Z = A.\bar{A} + A.A.B$$

$$Z = 0 + A.B$$

Paso 3

De acuerdo al siguiente **teorema**:

$$A + 0 = A$$

Se **aplica** en la **expresión** obtenida en el **paso 2** para **obtener** la **expresión simplificada**

$$Z = 0 + A.B$$

$$Z = AB$$

$$\text{b) } Z = (A + \bar{A})(A.B + A.B.\bar{C})$$

Solución.

$$\text{Sea } Z = (A + \bar{A})(A.B + A.B.\bar{C})$$

Paso 1

De acuerdo al siguiente **teorema**:

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

Se **aplica** en la **expresión inicial**

$$Z = (A + \bar{A})(A.B + A.B.\bar{C})$$

$$Z = 1(A.B + A.B.\bar{C})$$

Paso 2

De acuerdo al siguiente **teorema**:

$$A.1 = A$$

Se **aplica** en la **expresión** obtenida en el **paso 1**

$$Z = 1(A.B + A.B.\bar{C})$$

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

Paso 3

De acuerdo al siguiente teorema:

$$A + AB = A$$

Se **aplica** en la **expresión** obtenida en el **paso 2** para **obtener** la **expresión simplificada**

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

$$Z = A.B$$

Teoremas de DeMorgan.

Son **dos teoremas** propuestos por el matemático **Augustus De Morgan**.

a) El **complemento** de un **producto de variables** es igual a la **suma de los complementos de las variables**.

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

b) El **complemento** de una **suma de variables** es igual al **producto de los complementos de las variables**.

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

Al **aplicar** los **teoremas de DeMorgan** se tiene en cuenta el **número de negaciones** tanto para las **variables** como para los **conectivos lógicos** así:

Si el número de negaciones es par	Si el número de negaciones es impar
- La variable no cambia. - El conectivo lógico no cambia.	- La variable se escribe negada. - El conectivo lógico Y (And) cambia por el conectivo lógico O (Or) y el conectivo lógico O (Or) cambia por el conectivo lógico Y (And).

Ejercicios de Aprendizaje.

Aplicar los **teoremas de DeMorgan** a las siguientes expresiones:

a) $Z = \overline{(A + B + C)D}$

Solución.

Sea $Z = \overline{(A + B + C)D}$

Se observa que la **expresión** completa se **complementa una vez**.

Con ello se tiene en cuenta:

Si el número de **negaciones** es **impar**:

- La variable se **escribe negada**
- El **conectivo lógico Y (And)** cambia por el **conectivo lógico O (Or)** y el **conectivo lógico O (Or)** cambia por el **conectivo lógico Y (And)**.

Se **aplica** lo antes descrito en la **expresión original** y se **obtiene**:

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

b) $Z = \overline{AB + CD + E(F + G)}$

Solución.

Sea $Z = \overline{AB} + CD + \overline{E}(F + \overline{G})$

Se observa que en la **expresión** hay elementos que **se complementan una y dos veces**.

Con ello se tiene en cuenta:

Si el número de **negaciones es par**:

- La variable **no cambia**.
- El **conectivo lógico** no cambia.

Si el número de **negaciones es impar**:

- La variable se **escribe negada**.
- El conectivo lógico Y (**And**) cambia por el conectivo lógico O (**Or**) y el conectivo lógico O (**Or**) cambia por el conectivo lógico Y (**And**).

Se **aplica** lo antes descrito en la **expresión original** y se **obtiene**:

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

Mapas de Karnaugh

Un **mapa de Karnaugh** es la **representación gráfica** de una **función lógica** a partir de la **tabla de verdad**. El número de **casillas** del **mapa** es igual al **número de combinaciones** que se pueden obtener con las **variables de entrada**. Los mapas se pueden utilizar para **2, 3, 4 y 5 variables**.

Si se aplica adecuadamente genera **expresiones simples**.

2.4.2 EJERCICIO DE APRENDIZAJE.

Dada la siguiente **tabla de estados**; hallar:

- a) **Expresión lógica**.
- b) **Simplificar** la expresión lógica utilizando un **Mapa de Karnaugh**.

ENTRADAS			SALIDA
A	B	C	Z
0	0	0	1

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	1

Solución.

a) Para pasar de la **tabla de estados** a la **función lógica** primero se **ubican** las **combinaciones** en las cuales la **salida es 1**. Luego se procede a **decodificar** cada una de las **combinaciones** así:

■ Si la **variable** en la **tabla de verdad** es **1** se escribe la **variable**.

■ Si la **variable** en la **tabla de verdad** es **0** se escribe la **variable negada**.

Al final **todos** los términos se **unen** por medio de **Or (Suma lógica)**.

Teniendo en cuenta lo anterior, las **combinaciones** en las cuales la **salida es 1** son:

ENTRADAS			SALIDA
A	B	C	Z
0	0	0	1
0	0	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Decodificando se **obtiene** la siguiente **expresión lógica**:

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

b) Ahora pasamos a **implementar** el **mapa de Karnaugh** correspondiente a **tres variables**.

El **mapa** tiene **2ⁿ** casillas; es decir **2³ = 8**.

AB \ C	0	1
00	0 1	1 1
01	2	3
11	6 1	7 1
10	4 1	5 1

Autoría propia

- En el **mapa de Karnaugh** se coloca **1** en las **casillas** que corresponden a las **combinaciones** en las cuales **la función es 1** en la **tabla de estados**.
- Luego se procede a **generar grupos de "1"s**. Cada **agrupación** debe contener **1, 2, 4, 8 ó 16 unos**.
- Dos celdas son **adyacentes**
- Los **unos** que se **agrupan** deben ser **adyacentes (no en diagonal)**.
- La **fila superior** y la **fila inferior** son **adyacentes**.
- Las **columnas** de los **extremos** son **adyacentes**.
- Entre más **unos** se **agrupen** más se **simplifica la función original**.

En el **ejemplo** se ve la posibilidad de **generar dos agrupaciones**. Se puede **compartir unos** en más de una **agrupación** ya que son **agrupaciones** diferentes (**No hay unos redundantes**).

AB \ C	0	1
00	0 1	1 1
01	2	3
11	6 1	7 1
10	4 1	5 1

Autoría propia

La **expresión simplificada** se determina de la siguiente forma:

- De cada **agrupación** sale un **término** el cual está conformado por las **variables** que **no cambian** para todos los **unos de la agrupación**.
- Las **variables** que cambian se **desechan**.
- La(s) variable(s) que **no cambian** se escriben **como son** (0 negada y 1 variable).
- Finalmente se **unen** los términos **por medio de Or (Suma lógica)**.

Para el ejemplo **se obtiene**:

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

Tema 4 Flip - Flops

Un flip flop es un **circuito secuencial** que está conformado por **compuertas lógicas**. Es un **dispositivo de memoria**.

Son elementos **biestables síncronos**, ya que la **salida** varía de **estado** únicamente cuando la **señal de control** llamada **reloj (Clock, CLK)** lo indica.



Autoría propia

Flanco.

En la **señal de reloj** se denomina **flanco** a la **transición** del **nivel bajo al alto** (**flanco de subida -- transición de 0 a 1**) o del **nivel alto al bajo** (**flanco de bajada --transición de 1 a 0**). En esos instantes es que los **flip flops** trabajan.

Flanco de subida:



Autoría propia

Flanco de bajada:



Autoría propia

Existen **cuatro tipos** de flip flops:

- Flip flop **SR**.
- Flip flop **JK**.
- Flip flop **D**.
- Flip flop **T**.

- Flip flop SR.

El **flip flop SR** es la unidad de **memoria** más simple, y en base a él, se diseñan **flip flops más avanzados**. El **circuito** tiene **dos entradas**, Set (S) y Reset (R), y **dos salidas**, llamadas Q y $\bar{Q}$

La siguiente **tabla** muestra el funcionamiento de un **flip flop SR** activo en **flanco de subida**. (Para **flanco de bajada** el **comportamiento** es similar)

ENTRADAS			SALIDA
CLK	S	R	Q
?	0	0	Permanece
?	0	1	0
?	1	0	1
?	1	1	Prohibido

- Flip flop JK.

El **flip flop JK** es la **mejora del Flip flop SR**. La **diferencia** se da cuando las **señales de entrada J=1 y K=1** (**Prohibido, en el caso del flip flop SR**) pues la **salida** es la **negación**.

La siguiente **tabla** muestra el funcionamiento de un **flip flop JK** activo en **flanco de subida**. (Para **flanco de bajada** el comportamiento es similar)

ENTRADAS			SALIDA
CLK	J	K	Q
?	0	0	Permanece
?	0	1	0
?	1	0	1
?	1	1	Invierte

- Flip flop D.

El **flip flop D** tiene **una entrada, Data (D)** y **dos salidas**, llamadas **Q** y $\bar{Q}$.

La siguiente **tabla** muestra el funcionamiento de un **flip flop D** activo en **flanco de subida**. (Para **flanco de bajada** el comportamiento es similar)

ENTRADAS		SALIDA
CLK	D	Q
?	0	0
?	1	1

- **Flip flop T.**

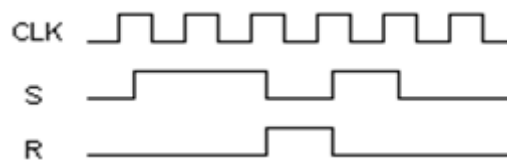
El **flip flop T** tiene **una entrada, Toggle (T)** y **dos salidas**, llamadas **Q** y $\bar{Q}$.

La siguiente **tabla** muestra el funcionamiento de un **flip flop T** activo en **flanco de subida**. (Para **flanco de bajada** el comportamiento es similar)

ENTRADAS		SALIDA
CLK	T	Q
?	0	Permanece
?	1	Invierte

2.4.3 EJERCICIOS DE APRENDIZAJE.

1) Dadas las **señales de entrada** determine la **señal de salida** (Flip flop activo en flanco de bajada)



Autoría propia

Solución.

Se **señalan** los **flancos** en los cuales **trabaja** el **flip flop**.

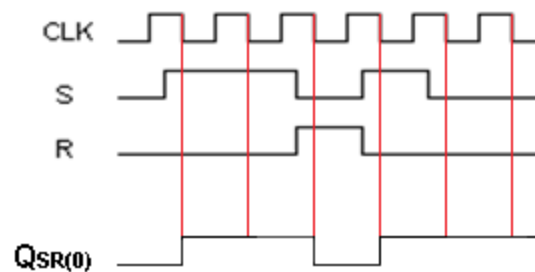
La **señal de salida** se **inicia** en **0** o en **1**

Para éste **ejercicio** se **inicia** en **0**

Se utiliza la **tabla de funcionamiento** del **flip flop SR**

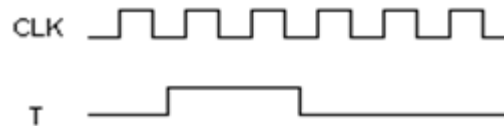
ENTRADAS			SALIDA
CLK	S	R	Q
?	0	0	Permanece
?	0	1	0
?	1	0	1
?	1	1	Prohibido

La **señal de salida** que se obtiene es:



Autoría propia

2) Dada la **señal de entrada** determine la **señal de salida** (Flip flop activo en flanco de subida)



Autoría propia

Solución.

Se **señalan** los **flancos** en los cuales **trabaja** el **flip flop**.

La **señal de salida** se **inicia** en **0** o en **1**

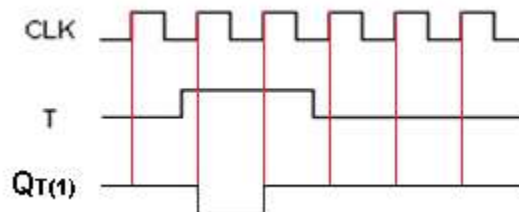
Para éste **ejercicio** se **inicia** en **1**

Se utiliza la **tabla de funcionamiento** del **flip flop T**

ENTRADAS		SALIDA
CLK	T	Q

0	0	Permanece
1	1	Invierte

La **señal de salida** que se obtiene es:



Autoría propia

2.5 TEMA 5 CIRCUITOS INTEGRADOS DIGITALES

Los **circuitos integrados (chips)** son dispositivos **semiconductores electrónicos** que contiene diferentes **elementos electrónicos** como: **Resistores, diodos, transistores** entre otros.

El crédito al diseño del **primer circuito integrado** se debe al ingeniero **Yack Kilby (1958)** por haber logrado integrar **seis transistores** en un mismo **circuito integrado**, con el propósito de hacer un **oscilador de rotación de fase**.

Los **circuitos integrados** facilitaron la **miniaturización** de los **aparatos electrónicos** debido a su **incremento** en el desarrollo e invención de **nuevas tecnologías** como:

- ✓ Computadores,
- ✓ celulares,
- ✓ informática,
- ✓ comunicaciones,
- ✓ manufactura,
- ✓ transporte,
- ✓ internet, entre otros.

Según la **naturaleza de la señal** los **circuitos integrados** se clasifican en dos grupos:

Circuitos integrados analógicos.	Pueden ser simples transistores encapsulados juntos, sin unión, hasta dispositivos completos como amplificadores, osciladores reguladores o receptores de radio completos.
Circuitos integrados digitales.	Pueden ser puertas básicas lógicas (Y, O, NO) hasta los más complejos como microprocesadores o micro controladores.

Circuitos integrados digitales

Los **circuitos integrados digitales** se **clasifican** de acuerdo a:

- Familia Lógica.
- Complejidad (Integración).

Familia Lógica.

- TTL (**Lógica transistor transistor**).

Es la **más utilizada**.

- ECL (**Lógica de emisor acoplado**).

Se utiliza en **sistemas de alta velocidad**.

- MOS (**Semiconductor Metal Oxido**).

Maneja gran **densidad de componentes**.

- CMOS (**Semiconductor Metal Oxido Complementario**).

Se utiliza en **sistemas** que requieren **bajo consumo de potencia**.

Complejidad (Integración).

- SSI (**Small Scale Integration**) nivel pequeño: desde **10** a **100** transistores.
- MSI (**Medium Scale Integration**) nivel medio: De **101** a **1000** transistores.
- LSI (**Large Scale Integration**) nivel grande: De **1001** a **10000** transistores.

- **VLSI (Very Large Scale Integration)** nivel muy grande: De **10001** a **100000** transistores.
- **ULSI (Ultra Large Scale Integration)** nivel ultra grande: De **100001** a **1000000** transistores.
- **GLSI (Giga Large Scale Integration)** nivel giga grande: Mucho más de un **millón** de transistores.

La **fabricación** de los **circuitos integrados** es compleja por tener una alta **integración de componentes** en un **espacio muy reducido**, de manera que llegan a ser **microscópicos**. Permiten grandes **simplificaciones** con relación a los **antiguos circuitos**, y el **montaje** se realiza de **forma más rápida**.

Los **circuitos integrados digitales** son **dispositivos de dos estados**, un estado está cercano a **0 v o tierra (bajo = 0)** y el otro está cercano al **voltaje de alimentación del circuito integrado (alto = 1)**. Además son capaces de procesar **dígitos binarios**.

Lógica TTL (Lógica Transistor Transistor): Utiliza **transistores** para su funcionamiento y una alimentación de **tensión de 5 voltios DC**.

Series:

- **74LSXX**
- **74SXX**
- **74HCXX**

Según su **número de serie** se diferencian como **compuertas**.

Por **ejemplo**:

COMPUERTAS				
Nand (Dos entradas)	Not	And (Dos entradas)	Or (Dos entradas)	Exor
7400	7404	7408	7432	7486

Circuito combinacional.

Es aquel que está formado por **compuertas lógicas elementales (AND, OR, NAND, NOR, entre otros)**, que tiene un determinado número de **entradas y salidas**, dependiendo los valores que toman **las salidas** exclusivamente de los que toman las **entradas** en ese instante.

Ejemplo de este **tipo de circuitos** son: **los codificadores, decodificadores, multiplexores, demultiplexores**, entre otros.

Circuito secuencial.

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**.

Ejemplo de este **tipo de circuitos** son: **Flip Flops, contadores, etc.**

2.6 TEMA 6 CODIFICADOR

Un **codificador** (es un **circuito combinacional**, cuyo funcionamiento es el siguiente:

- Se obtiene en la **salida el código binario** correspondiente a la **entrada que se activa (Colocada en uno)**.
- Posee **2ⁿ** entradas y **n** salidas.
- Existen **dos** tipos de **codificadores**: **codificadores sin prioridad** y **codificadores con prioridad**.
- Los **codificadores con prioridad** generan el **código binario** de la entrada **activa más alta (Valor decimal)**
- Un ejemplo de aplicación se da en el **teclado del computador** que posee un **circuito codificador**.

2.6.1 EJERCICIO DE APRENDIZAJE.

Representar un **codificador** que cumpla con las siguientes **características**:

- **n=3**
- **Entrada A₅=1 y el resto en 0**

Solución.

Dado que $n=3$; esto indica el **número de salidas**.

Con el dato del **número de salidas** se procede a determinar el **número de entradas** utilizando 2^n

Reemplazando n se tiene

$$2^n$$

$$2^3 = 8$$

El **codificador** posee **8** entradas.

Como se da el **valor de las entradas** y se referencian con **A**; las entradas se **numeran** desde **A₀** hasta **A₇**

Como no se da la referencia de las **salidas** se representan utilizando **cualquier variable**. Para este caso se utilizará la **Y**. Las **salidas** se numeran desde **Y₀** hasta **Y₂**

Dado que:

Entrada A₅=1 y el resto en 0

Esto indica que en la **salida** se obtiene el **código binario** correspondiente a la **entrada que se activa (Colocada en uno)**.

Con ello se **obtiene** en la **salida** el **código binario del número 5 (101)**

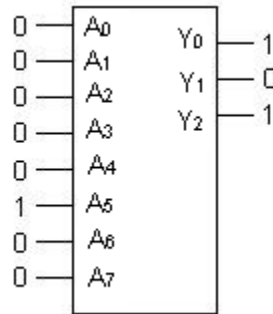
$$Y_0 = 1$$

$$Y_1 = 0$$

$$Y_2 = 1$$

Nota: Tener en cuenta que Y₀ es el bit menos significativo y que Y₂ es el bit más significativo.

Así el **codificador requerido** es:



Autoría propia

Tema 7 Decodificador

Un **decodificador** es un **circuito combinacional**, cuyo **funcionamiento** es el siguiente:

- Dado el **código binario** en la **entrada**, se **activa** (Coloca en uno) la **salida correspondiente a dicho código**.
- Posee **n entradas** y **2ⁿ salidas**.
- Un ejemplo de aplicación se da con el manejo de los diferentes **periféricos del computador**.

2.6.2 EJERCICIOS DE APRENDIZAJE

- a) Elaborar la **tabla de estados** para un **decodificador de dos entradas**.
- b) Implementar el **decodificador con compuertas lógicas** propuesto en el numeral anterior.

Solución.

- a) Dado que **n=2**; esto indica el **número de entradas**.

Con el dato del **número de entradas** se procede a determinar el **número de salidas** utilizando **2ⁿ**

Reemplazando **n** se tiene

$$2^n$$

$$2^2 = 4$$

El **decodificador** posee **4 salidas**.

Como no se da la **referencia** de las **entradas ni salidas** se representan utilizando **cualquier variable**.

Para este caso se utilizará la **A** para representar las **entradas** y la **D** para representar las **salidas**.

Las **entradas** se numeran como **A₀** y **A₁**

Las **salidas** se numeran desde **D₀** hasta **D₃**

Nota:

Tener en cuenta que **A₀** es el **bit menos significativo** y que **A₁** es el **bit más significativo**.
Tener en cuenta que **D₀** es el **bit menos significativo** y que **D₃** es el **bit más significativo**.

Con ello la **tabla** que muestra el funcionamiento del **decodificador** es:

ENTRADAS		SALIDAS			
A₁	A₀	D₃	D₂	D₁	D₀
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 la **tabla** se observa que dependiendo del **código de entrada** se activa (Coloca en 1) la **salida** correspondiente a dicho **código**.

b) Para implementar el **circuito** con **compuertas lógicas** se tiene en cuenta la **tabla de estados** elaborada en el numeral **a**.

Se procede a determinar la **relación** de cada una de las **salidas** con las **entradas**.

Para ello si la **variable** en la **tabla de verdad** es **1** se coloca **normal** pero si es **0** se coloca la **variable negada**.

Para **D₃**

ENTRADAS		SALIDA
A₁	A₀	D₃
0	0	0
0	1	0
1	0	0
1	1	1

Se observa que **D₃** es **1** cuando **A₀** y **A₁** son **1**

Así

$$D_3 = A_1 A_0$$

Para D_2

ENTRADAS		SALIDA
A_1	A_0	D_2
0	0	0
0	1	0
1	0	1
1	1	0

Se observa que D_2 es 1 cuando $A_0 = 0$ y $A_1 = 1$

Así

$$D_2 = A_1 \bar{A}_0$$

Para D_1

ENTRADAS		SALIDA
A_1	A_0	D_1
0	0	0
0	1	1
1	0	0
1	1	0

Se observa que D_1 es 1 cuando $A_0 = 1$ y $A_1 = 0$

Así

$$D_1 = \bar{A}_1 A_0$$

Para D_0

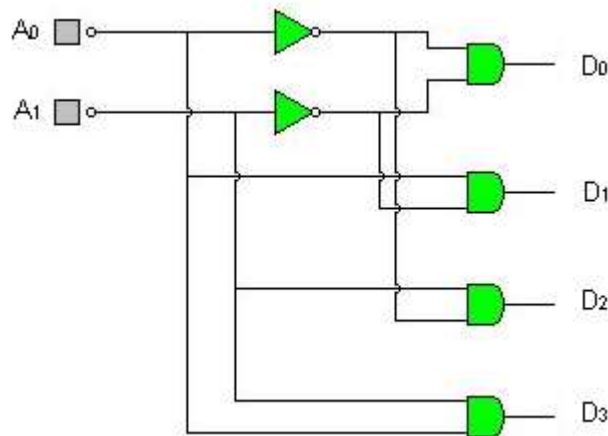
ENTRADAS		SALIDA
A_1	A_0	D_0
0	0	1
0	1	0
1	0	0
1	1	0

Se observa que D_0 es 1 cuando A_0 y A_1 son 0

Así

$$D_0 = \overline{A_1} \overline{A_0}$$

El **decodificador** con **compuertas lógicas** es:



Autoría propia

2.7 TEMA 8 MULTIPLEXOR

El **multiplexor (MUX)** es un **circuito combinacional** que permite seleccionar de las **entradas** (mediante **selectores**) cuál de ellas pasará a la **salida**.

- Al **multiplexor** también se le llama **selector de datos**.
- Posee 2^n entradas y **una salida**.
- La cantidad de **selectores** está dada por **n**.
- Número de entradas = **2^n** .

2.7.1 EJERCICIO DE APRENDIZAJE

Representar un **multiplexor** que cumpla con las siguientes características:

- $n=2$
- Entradas A_2 y A_0 en 1 y el resto en 0

 Selectores $S_1=1$ $S_0=1$

Solución.

Dado que $n=2$; esto indica el **número de selectores**.

Con el dato del **número de selectores** se procede a determinar el **número de entradas** utilizando 2^n
Reemplazando n se tiene

2^n

$2^2 = 4$

El **multiplexor** posee **4 entradas**.

Nota: Tener en cuenta que todo **multiplexor** posee una salida.

Como se da el valor de las **entradas** y se referencian con **A**; las **entradas** se numeran desde **A₀** hasta **A₃**

Como se dan los valores de los **selectores** y se referencian con **S**; los **selectores** se numeran como **S₀** y **S₁**

Como no se da la **referencia** de la **salida** se representa utilizando **cualquier variable**. Para este caso se utilizará la **Y**.

“

Nota:

Tener en cuenta que **A₀** es el **bit menos significativo** y que **A₃** es el **bit más significativo**.
Tener en cuenta que **S₀** es el **bit menos significativo** y que **S₁** es el **bit más significativo**.

”

Dado que:

Entradas A₂ y A₀ en 1 y el resto en 0

Con ello se tiene:

A₀ = 1

A₁ = 0

A₂ = 1

$A_3 = 0$

Además se indica los **valores de los selectores**:

$S_0 = 1$

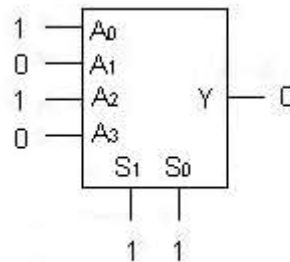
$S_1 = 1$

Esto indica que en la **salida se obtiene el bit** correspondiente a la **entrada seleccionada por el código 11**

El **código 11** corresponde a la **entrada A_3**

Como $A_3 = 0$ en la **salida Y se obtiene 0**

Así el **multiplexor** requerido es:



Autoría propia

2.8 TEMA 9 DEMULTIPLEXOR

Un **demultiplexor (DEMUX)** es un **circuito combinacional**.

- Permite **seleccionar** a cuales de las **salidas** (mediante selectores) pasará la **entrada**.
- Posee **una entrada y 2^n salidas**.
- Al **demultiplexor** también se le llama **distribuidor de datos**.
- La cantidad de **selectores** está dada por **n** .
- **Número de salidas = 2^n** .

2.8.1 EJERCICIO DE APRENDIZAJE

Representar un **demultiplexor** que cumpla con las siguientes características:

- **$n=2$**
- **Entrada A en 1**

 Selectores $S_1=0$ $S_0=1$

Solución.

Dado que $n=2$; esto indica el **número de selectores**.

Con el dato del **número de selectores** se procede a determinar el **número de salidas** utilizando 2^n

Reemplazando n se tiene

$$2^n$$

$$2^2 = 4$$

El **demultiplexor** posee **4 salidas**.

Nota: Tener en cuenta que todo **demultiplexor** posee **una entrada**.

Se da el **valor de la entrada** y se referencia con **A**

Como se dan los **valores de los selectores** y se referencian con **S**; los **selectores** se numeran como S_0 y S_1

Como no se da la **referencia de las salidas** se representan utilizando **cualquier variable**. Para este caso se utilizará la **Y**. Las **salidas** se numeran desde Y_0 hasta Y_3

“

Nota:

Tener en cuenta que Y_0 es el **bit menos significativo** y que Y_3 es el **bit más significativo**.

Tener en cuenta que S_0 es el **bit menos significativo** y que S_1 es el **bit más significativo**.

”

Dado que:

Entrada A = 1

Además se indica los **valores de los selectores**:

$$S_0=1$$

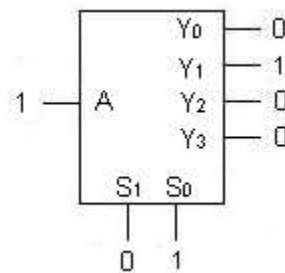
$$S_1=0$$

Esto indica que el valor de la **entrada** pasará a la **salida seleccionada** por el **código 01**

El **código 01** corresponde a la **salida Y_1**

Como **$A = 1$** en la **salida Y_1** se obtiene **1**

Así el **demultiplexor** requerido es:



Autoría propia

2.9 TEMA 10 REGISTRO

Es un **dispositivo electrónico semiconductor** creado a partir de implementaciones con **flip flops** los cuales permiten el manejo de **datos** usando **circuitos combinatorios** y **secuenciales específicos**.

El **registro de n-bits** tiene un grupo de **n flip flops** y **almacena** cualquier información **binaria** que contenga **n bits**. Los **flip flops** guardan la **información binaria** y las **compuestas lógicas** controlan cuando y como se **transfiere información**.

Los **registros** se representan por **letras mayúsculas** las cuales pueden estar acompañadas por **números**.

La representación típica de un **registro** es el de un **rectángulo** con el **nombre del en la parte superior**.

2.9.1 EJERCICIOS DE ENTRENAMIENTO

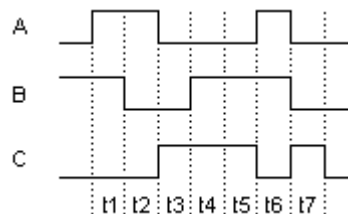
Pautas para desarrollar los siguientes ejercicios:

De acuerdo a los conceptos estudiados y a investigación previa desarrollar los siguientes ítems:

1) Consultar los niveles lógicos de la familia TTL.

- 2) Consultar las referencias de circuitos integrados de las compuertas lógicas básicas.
- 3) Determinar aplicabilidad de las compuertas lógicas en diferentes sectores del medio.
- 4) Establecer ¿por qué es importante la simplificación de expresiones lógicas para la implementación de circuitos digitales?
- 5) Enunciar tres ejemplos de circuitos combinacionales y tres ejemplos de circuitos secuenciales.

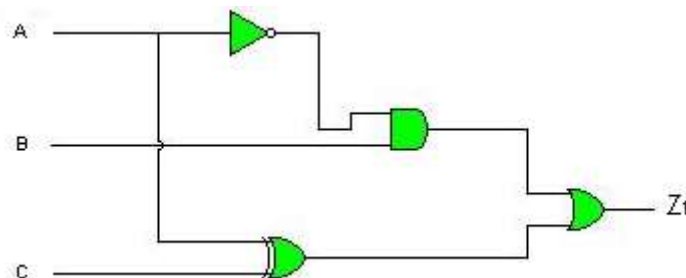
Dadas las señales de entrada



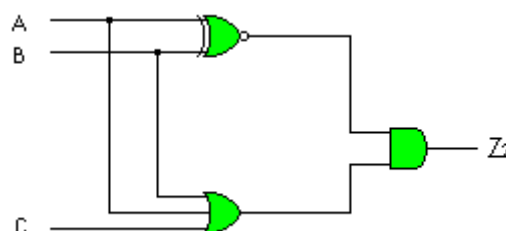
Autoría propia

Hallar la señal y la expresión lógica de salida para:

1)



2)



Mediante las técnicas del Álgebra de Boole, simplificar:

- a) $Z = (W + XY)(\bar{X}Y + W)(W + F)$
- b) $Z = [AB(C + \bar{B} + \bar{D}) + \bar{A} + \bar{B}]CD$
- c) $Z = (\bar{P} + P) + (PQ + P\bar{H}Q)$

Aplicar los teoremas de DeMorgan a:

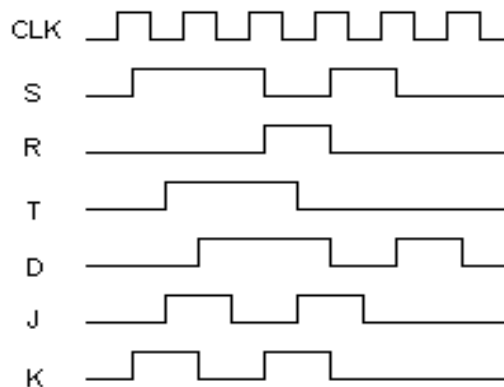
- a) $Z = \overline{F + E\bar{C}} + H(\overline{G + I})$
- b) $Z = \overline{\bar{A}N(B + \bar{D})}$
- c) $Z = \overline{(\bar{A} + \bar{D})(N + \bar{X})}$

Mediante Mapas de Karnaugh simplificar.

- a) $Z = AB\bar{C} + A\bar{B}C + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC$
- b) $Z = \bar{A}\bar{B}\bar{C}D + \bar{A}BC\bar{D} + ABCD + \bar{A}\bar{B}\bar{C}D + \bar{A}BC\bar{D} + A\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}BCD + \bar{A}\bar{B}CD + \bar{A}BCD$

Dadas las señales de entrada determine las señales de salida.

(Flip flops activos en flanco de subida)



Decodificador

- a) Elaborar la tabla de estados para un decodificador de tres entradas.
- b) Implementar el decodificador con compuertas lógicas propuesto en el numeral anterior.

Laboratorio

Utilizando el protoboard comprobar la tabla de estados de las siguientes

compuertas lógicas:

■ 7404

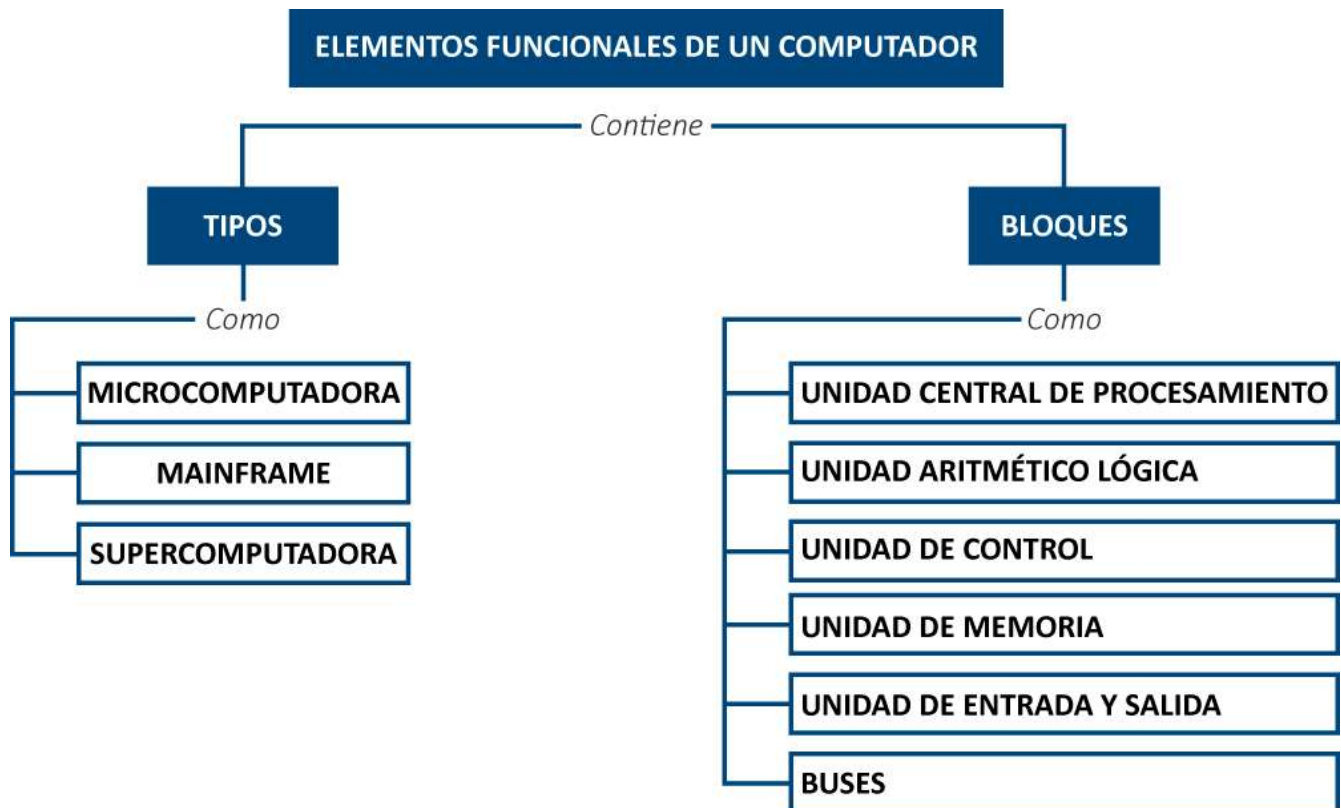
■ 7408

■ 7432

■ 7486

3 UNIDAD 2 ELEMENTOS FUNCIONALES DE UN COMPUTADOR

3.1.1 RELACIÓN DE CONCEPTOS



Buses: Son circuitos que conectan a la unidad central de procesamiento con el resto de los componentes del computador.

Computador: Es una máquina digital electrónica que recibe y procesa información automáticamente a través de programas informáticos.

Elementos funcionales: Son aquellos elementos que forman parte de una estructura de orden superior (En éste caso el computador)

Mainframe: Conocida como computadora central. Es un gran computador capaz de realizar el procesamiento de datos complejos. Se utilizan como sistemas centrales de grandes organizaciones o empresas.

Microcomputadora: Es un tipo de computadora que utiliza un microprocesador como unidad central de procesamiento. Generalmente son computadoras que ocupan espacios físicos pequeños.

Supercomputadora: Son computadoras de alto desempeño. Extremadamente potentes y capaces de realizar tareas de cálculo a una velocidad que equivale a cientos de veces la velocidad de una computadora convencional.

Unidad Aritmético Lógica: Es un circuito digital que se encarga de realizar las operaciones aritméticas y lógicas que hacen parte del conjunto de operadores que tiene disponibles un computador.

Unidad Central de Procesamiento: Es un circuito integrado que se encarga del control y procesamiento de la información en el sistema digital.

Unidad de Control: Es un circuito digital que se encarga de controlar todas las operaciones del computador.

Unidad de Entrada y Salida: Conjunto de interfaces que usan los dispositivos funcionales de un sistema de procesamiento de datos para comunicarse con los demás medios de información, usando señales recibidas o enviadas.

Unidad de Memoria: Es un circuito digital diseñado para almacenar datos en forma binaria durante un intervalo de tiempo, usando estructuras y arreglos de registros con flip flops.

3.2 TEMA 1 EL COMPUTADOR

- La **computación** se centra en **actividades** que requieren del **computador** para lograr los **objetivos** propuestos.
- El **computador** es una **máquina digital electrónica** que **recibe y procesa información** automáticamente a través de **programas informáticos**.
- También se nombra al **computador** como **computadora u ordenador**.
- Los **computadores** se usan en diferentes **áreas** como: **Medicina, Matemáticas, Ingeniería, Educación**.
- La ventaja del **computador** radica en que por medio de un **conjunto de instrucciones** llamado **programa**, el **usuario** puede modificarlo de acuerdo a su requerimiento específico. Así la **máquina** puede realizar simultáneamente diferentes **tipos de tareas** al **procesar información** de una amplia variedad.
- Las **funciones básicas** de un **computador** son: **Procesar, almacenar y transferir información**.

3.3 TEMA 2 TIPOS DE COMPUTADORES

Supercomputadora.

- Es aquella **máquina** diseñada para realizar **cálculos** que requieren de **altas velocidades** para el procesamiento de la información.
- Este tipo de **computadoras** son utilizadas para **fines muy específicos**.
- Poseen un gran **número de procesadores** que trabajan en **paralelo**.
- Realizan **billones de operaciones** por **segundo**.



Tomado de: <http://www.muylinux.com/2015/07/14/tianhe-2-supercomputadora>

Mainframe.

- Es aquella **máquina** diseñada para **procesar información** de **grandes empresas**.
- La **potencia de cálculo** es **menor** a la de una **supercomputadora**.
- Realizan **millones de operaciones** por **segundo**.



Tomado de: <http://blog.syncsort.com/2015/07/mainframe/breathing-new-life-into-the-mainframe/>

Microcomputadoras.

- Es aquella **máquina** diseñada para **propósitos generales**.
- El **componente principal** es un **microprocesador**.
- Pueden ser: **Computadora de escritorio, portátil**.



Tomado de: <http://informatica1-17.blogspot.com.co/2015/01/las-microcomputadoras-o-pc.html>

3.4 TEMA 3 UNIDAD CENTRAL DE PROCESAMIENTO

La **Unidad Central de Procesamiento** o **CPU** es un **circuito integrado** que se encarga del **control y procesamiento de la información** en el **sistema digital**.

La **CPU** toma la **información (Instrucciones)** almacenada en la **memoria** y le realiza los siguientes pasos: **Examina, interpreta y ejecuta**.

La **Unidad Central de Procesamiento** está compuesta por: **Registros, unidad aritmético lógica y unidad de control**.

3.4.1 UNIDAD ARITMÉTICO LÓGICA

La **unidad Aritmético Lógica** o **ALU** es un **circuito digital** que se encarga de realizar las **operaciones aritméticas y lógicas** que hacen parte del **conjunto de operadores** que tiene disponibles un **computador**.

Está **formada** por:

- Operadores aritméticos.
- Operadores lógicos.
- Operadores de desplazamiento.
- Registros para almacenar datos temporales.
- Registro acumulador.
- Registro de estado.
- Registro contador de programa.
- Registro de direcciones de interrupción.

Operador aritmético

Es un **circuito electrónico** encargado de realizar una o varias **operaciones aritméticas** tales como: **Suma, resta, multiplicación y división**.

Operador lógico

Es un **circuito electrónico** encargado de realizar una o varias **operaciones lógicas** tales como: **Not, And, Or, Exor**.

Operador de desplazamiento

Es un **circuito electrónico** encargado de realizar el **corrimiento de los bits** de **una palabra, dato o registro** hacia la **derecha** o hacia la **izquierda**. Normalmente el **operador de desplazamiento** está formado por **compuertas lógicas**.

Registros para almacenar datos temporales

Son aquellos **registros** que tienen como propósito **almacenar** temporalmente **datos y resultados intermedios**.

Registro acumulador

Es aquel que se utiliza para **almacenar los resultados del operador**.

Registro de estado

También se conoce como **flags (Banderas)**. Se utiliza para indicar el **estado del procesador** y de los **resultados de ciertas operaciones**.

Los **campos** más comunes de éste **registro** son:

- **Signo:** Contiene el **bit de signo** del **resultado** de la última **operación aritmética**.
- **Cero:** Se coloca en **1** cuando el **resultado** es **0**.
- **Acarreo:** Se coloca en **1** si una **operación** genera **acarreo (suma) o préstamo (resta)** del **bit** más significativo.
- **Igual:** Se coloca en **1** si el resultado de una **comparación lógica** es la **igualdad**.
- **Desbordamiento:** Usado para indicar un **desbordamiento aritmético**.
- **Supervisor:** Indica si la **Unidad Central de Procesamiento** funciona en **modo de supervisor o usuario**. Únicamente en **modo supervisor** se pueden ejecutar ciertas **instrucciones privilegiadas** y se puede acceder a ciertas **áreas de memoria**.

Registro contador de programa.

Es aquel que **almacena** la **dirección** de la siguiente **instrucción** a ejecutar.

Registro de direcciones de interrupción.

Es aquel que **almacena** la **dirección** de atención a la **interrupción**.

3.4.2 UNIDAD DE CONTROL

- La **Unidad de Control** se encarga de controlar todas las **operaciones del computador**. De esta forma **obtiene e interpreta** las **instrucciones** que se encuentran **almacenadas en la memoria**. De acuerdo a su **interpretación** le envía la **información** al **componente adecuado** para su **manipulación**.
- Determina las **operaciones a realizar** y el **orden de ejecución**.

Está **formada** por:

- Registro de instrucción.
- Controlador y decodificador.
- Secuenciador.
- Registro contador de programa.

 Reloj.

Registro de instrucción

Éste **registro** contiene el **código de la instrucción** con la que la **Unidad de Control** determina el **conjunto de microoperaciones** a realizar durante el **ciclo de ejecución**.

Controlador y decodificador

El **controlador** se encarga de interpretar la **instrucción** para su posterior **proceso**.

El **decodificador** se encarga de extraer el **código de operación** de la **instrucción en curso**.

Secuenciador

Se encarga de generar **órdenes** necesarias para ejecutar la **instrucción**.

Registro contador de programa.

Es aquel que **almacena la dirección** de la siguiente **instrucción** a **ejecutar**.

Reloj

Es el encargado de **generar pulsos periódicos** para ejecutar una o varias **microoperaciones** en cada **pulso de reloj**.

3.5 TEMA 4 UNIDAD DE MEMORIA

La **memoria** es un **circuito integrado** diseñado para **almacenar datos** en forma **binaria** durante un **intervalo de tiempo**, usando **estructuras y arreglos** de **registros con flip flops**.

Celda de memoria

Circuito electrónico que se usa para almacenar **bits de datos** en forma de **0 o 1**, a través de una celda o **flip-flop**. Posee **palabra de memoria**, como un grupo de **bits o celdas** que representan **instrucciones o datos** de información de algún tipo.

Las **memorias** pueden estar diseñadas de **flip flops tipo D**, cada espacio de **memoria o celda** posee una **dirección hexadecimal** que corresponde exactamente a la **posición de la celda** donde se encuentra o se va a **almacenar un dato**.

Fundamentos de las memorias

Los **elementos** básicos de las **memorias** son:

- El **medio o soporte** donde se **almacenan** los **datos** como **unos o ceros**.
- El **transductor** que ubique en el soporte el **dato requerido** o que detecte su **valor actual**.
- El **medio de direccionamiento**, que permita **leer, fijar o grabar** la **información** en **lugar y tiempos** requeridos para lo cual hay **tres condiciones**:
- Debe presentar **dos estados estables**, como una **magnitud física**.
- Cambio de un **estado** a otro cuando se aplique una **señal externa**.
- Saber el **estado** en el cual se **encuentra** en todo **momento**.

Los **medios** pueden ser **discretos** o sea un **dispositivo físico** individual que almacena cada **bit**, y **continuos** donde se almacenan unos **bits** a continuación de otros, en los **medios discretos** el acceso a los **datos** es más **simple y rápido**.

Según el tiempo que los datos permanecen grabados se puede clasificar la memoria en:

Clasificación de la memoria				
No volátil	Volátil	Con refresco	Lectura destructiva	Solo lectura o permanente
Permanecen los datos , hasta que se realice una operación de escritura .	La información desaparece cuando deja de suministrarse energía a la memoria .	Aun cuando la memoria este alimentada, los datos se van degradando presentando instantes en que no se pueden leer , para lo cual, deben refrescarse periódica y continuamente.	La lectura conlleva el borrado de la información e implica grabar el dato nuevamente	Permanece la información y no se puede borrar , algunas necesitan un proceso electrónico especial para ser borradas .

3.6 TEMA 5 UNIDAD DE ENTRADA Y SALIDA

Conjunto de interfaces que usan los **dispositivos funcionales** de un **sistema de procesamiento** de **datos** para **comunicarse** con los demás **medios de información**, usando **señales recibidas o enviadas**.

Cuando el **computador** recibe una **señal** de otra **interfaz** se le llama **señal de entrada** y cuando envía una **señal** a otro **dispositivo de datos** se le llama **señal de salida**.

Algunos **elementos de entrada** son:

-  **Mouse.**
-  **Teclado.**
-  **Scanner.**

Algunos **elementos de salida** son:

-  **Impresora.**
-  **Monitor.**
-  **Modem.**
-  **Red.**

3.7 TEMA 6 BUSES

Son **circuitos** que conectan a la **Unidad Central de Procesamiento** con el resto de los **componentes del computador**. A través de estos llega a la **CPU** la **información** y los requerimientos de **procesamiento** desde el **exterior** y permiten **transportar** hacia afuera los **resultados del procesamiento** realizado.

Bus principal

Es el encargado de llevar la **información** administrándola internamente entre todos los **dispositivos de entrada-salida** interconectado con la **memoria RAM**.

Se **subdivide** en:

-  **Bus de datos.**
-  **Bus de direcciones.**
-  **Bus de control.**

Bus de datos.

Es un **bus bidireccional** encargado de llevar **datos e instrucciones** desde el resto de **unidades** hacia la **Unidad Central de Procesamiento** o viceversa.

Bus de direcciones.

Contiene la **información digital** que envía la **CPU** a la **memoria** y demás **dispositivos direccionales** del sistema para seleccionar una posición de **memoria**, una unidad de **entrada/salida** o un **registro** particular de la misma.

Bus de control.

Se utiliza para **coordinar, sincronizar y comunicarse** con los **dispositivos externos**.

3.7.1 EJERCICIOS DE APRENDIZAJE.

1) Diseñar la **etapa lógica** de una **ALU** que permita realizar las siguientes operaciones:

☐ Not

☐ And

☐ Or

☐ Exor

Solución.

Para la **implementación** de la **etapa lógica** de la **ALU** se utilizan las **cuatro compuertas** correspondientes a las operaciones requeridas (**Not, And, Or, Exor**).

Para la **selección** de la **operación lógica** a realizar se utiliza un **multiplexor** que posea **cuatro entradas** (Cada una correspondiente a una **operación lógica**).

Como el **número de entradas es 4**

Se tiene

$$2^n = 4$$

Para ello **n** debe ser **2**

$$2^2 = 4$$

El **multiplexor** posee **2 selectores**.

Los **selectores** se referencian con **S**. Estos se numeran como **S₀** y **S₁**

La **salida** se referencia con **Y**.

Se utiliza **A** y **B** para los valores de **entrada** que van a ser operados.

“

Nota:

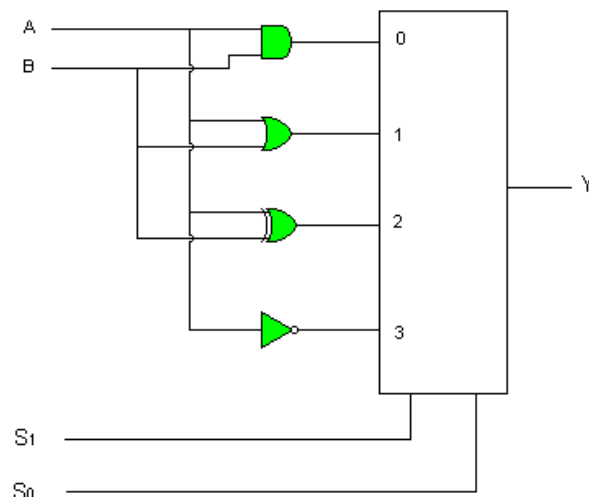
Tener en cuenta que **S0** es el **bit menos significativo** y que **S1** es el **bit más significativo**.

”

La **tabla de estados** que muestra el **funcionamiento** es:

S ₁	S ₀	Y	Operación
0	0	$Z = AB$	And
0	1	$Z = A + B$	Or
1	0	$Z = A \oplus B$	Exor
1	1	$Z = \bar{A}$	Not

Así la **etapa lógica** de la **ALU** requerida es:



Autoría propia

2) Enunciar la **secuencia** para **complementar** el contenido del **registro acumulador**.

Solución.

La **secuencia** es la siguiente:

- a) **Mover** el contenido del **acumulador** al **bus de datos**.
- b) **Mover** el contenido del **bus de datos** al **complementador**.
- c) **Activar** la lógica del **complementador**.
- d) **Mover** el contenido del **complementador** al **bus de datos**.
- e) **Mover** el contenido del **bus de datos** al **acumulador**.

3.7.2 EJERCICIOS DE ENTRENAMIENTO

Pautas para desarrollar los siguientes ejercicios:

De acuerdo a los conceptos estudiados y a investigación previa desarrollar los siguientes ítems:

- 1) Consultar las características de dos (2) de los procesadores de última generación.
- 2) Elaborar un mapa conceptual de un computador de última generación.
- 3) Diseñar una ALU que permita realizar las siguientes operaciones:

☐ Suma

☐ Resta

☐ And

☐ Or

☐ Exor

- 4) Elaborar un cuadro comparativo entre periféricos de entrada y periféricos de salida.
- 5) Investigar ¿Cuál es el ranking de las supercomputadoras?

Laboratorio

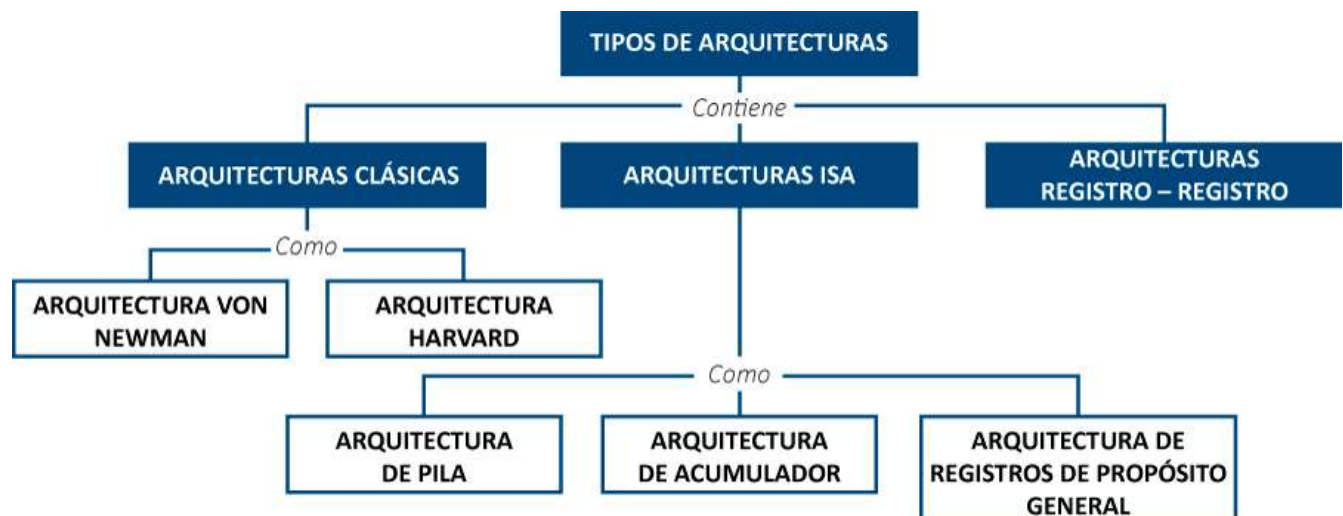
Utilizando el protoboard comprobar la tabla de funcionamiento de la ALU 74181

4 UNIDAD 3 TIPOS DE ARQUITECTURAS



Arquitectura de un ordenador [Enlace](#)

4.1.1 RELACIÓN DE CONCEPTOS



Arquitecturas clásicas: Son aquellas arquitecturas que se desarrollaron en las primeras computadoras electromecánicas y de tubos de vacío. Aun son usadas en procesadores empujados de gama baja y son la base de la mayoría de las arquitecturas modernas.

Arquitectura de Acumulador: Es aquella en la cual un operando esta implícitamente en el acumulador siempre leyendo e ingresando dato.

Arquitectura de Computador: Es la estructura funcional de un computador.

Arquitectura de Pila: Es aquella en la cual no es necesario nombrar a los operandos ya que estos se encuentran en el tope de la pila.

Arquitectura de Registros de Propósito General: Es aquella en la cual los registros se utilizan para almacenar datos o direcciones de forma flexible.

Arquitectura de Registro - Registro: Es aquella en la cual se trabaja solo con los datos de los registros con el fin de evitar el acceso frecuente a memoria, lo cual ayuda a mejorar la velocidad, ya que, la memoria es más lenta.

Arquitectura Harvard: Es aquella en donde el programa se almacena como un código numérico en la memoria, pero no en el mismo espacio de memoria ni en el mismo formato que los datos.

Arquitectura ISA: Es aquella que detalla las instrucciones que una unidad central de procesamiento puede entender y ejecutar, o el conjunto de todos los comandos implementados por un diseño particular de una CPU.

Arquitectura Von Newman: Es aquella en donde la unidad central de procesamiento se comunica a través de un solo bus con la memoria en la cual se almacenan tanto los códigos de instrucción del programa, como los datos que serán procesados por este.

4.2 TEMA 1 ARQUITECTURAS ISA

Se denomina **Arquitectura del conjunto de instrucciones** (Instruction Set Architecture, ISA)

Características.

■ La **interfaz** entre el **hardware y el software** de bajo nivel.

■ El **conjunto de atributos** es visible a:

- a) El programador en lenguaje de máquina.
- b) El sistema operativo.
- c) El compilador.

■ Posee:

- a) Conjunto de registros visibles al programador.
- b) Conjunto de instrucciones de ensamblador.
- c) Tipos básicos de datos soportados por las instrucciones.
- d) Modos de direccionamiento.
- e) Mecanismos de Entrada / Salida.

■ Se diferencian por el tipo de **almacenamiento interno** que usa el **procesador**, definiendo a la vez su tipo de **operando**.

4.2.1 ARQUITECTURAS DE PILA.

Características.

- Poseen **operandos implícitos** en el tope de la **pila**.
- El **registro TOS (Top of stack)** apunta al primer **operando** de **entrada**.
- El segundo **operando** de **entrada** es el que está una **posición** más **abajo**.
- El primer **operando** se descarta de la **pila**, el resultado ocupa el lugar del segundo y el **registro TOS** se actualiza para que apunte al **resultado**.
- Se emplean **instrucciones push y pop** para escribir **datos** en la **pila** y sacarlos de ella, respectivamente.

4.2.2 ARQUITECTURAS DE ACUMULADOR

Características.

- Posee un **operando implícito** en el **acumulador**.
- El **acumulador** es el primer **operando de entrada**.
- El segundo **operando** es una **posición de memoria**.
- El **resultado** se escribe en el **acumulador**.

4.2.3 ARQUITECTURAS DE REGISTROS DE PROPÓSITO GENERAL

Características.

- Un **operando** de entrada está en un **registro**.
- El segundo **operando** está en **memoria**.
- El **resultado** se escribe en un **registro**.

4.3 TEMA 2 ARQUITECTURAS REGISTRO - REGISTRO

Se denomina Arquitectura **Registro - Registro (Load / Store)**

Características.

- **Registros** como medio de **almacenamiento interno del procesador (General Purpose Registers, GPRs)**
- **Acceso** más rápido que a **memoria**.
- Eficientes para el **compilador**.
- **Almacenamiento** de **variables**.
- Menor **tráfico de memoria**.
- Mejor **rendimiento**.
- **Instrucciones** más compactas.
- Para cargar un **registro** con un valor desde **memoria** se emplean **instrucciones load**.
- Para transferir el **resultado** contenido en un **registro** a **memoria** se usan **instrucciones store**.

4.3.1 EJERCICIO DE APRENDIZAJE.

Escribir por medio de **instrucciones** la secuencia de código para **C= A+B**

Para:

- A) ARQUITECTURA DE PILA.
- B) ARQUITECTURA DE ACUMULADOR.
- C) ARQUITECTURA DE REGISTROS DE PROPÓSITO GENERAL.

Solución.

A, B y C son variables que se almacenan en memoria.

a) Arquitectura de Pila

PUSH A	/ Guarda el dato A en la pila
PUSH B	/ Guarda el dato B en la pila
ADD	/ Realiza la suma de A con B
POP C	/ Recupera de la pila el resultado de la suma y se almacena / en C

b) Arquitectura de Acumulador

LOAD A	/ Se carga el registro acumulador con el dato A
ADD B	/ Se almacena en el registro acumulador el resultado de / la suma de lo que contiene el registro acumulador / con B
STORE C	/ Copia lo que contiene el registro acumulador a C

c) Arquitectura de Registros de Propósito General

LOAD R1, A	/ Se carga el registro R1 con el dato A
LOAD R2, B	/ Se carga el registro R2 con el dato B
ADD R3, R1, R2	/ Se almacena en el registro R3 el resultado de la suma / de R2 con R3
STORE R3, C	/ Copia R3 a C

4.3.2 EJERCICIOS DE ENTRENAMIENTO

Pautas para desarrollar los siguientes ejercicios:

De acuerdo a los conceptos estudiados y a investigación previa desarrollar los siguientes ítems:

- 1) Realizar un mapa cognitivo de las arquitecturas Von Newman y Harvard.
- 2) Elaborar un cuadro comparativo entre las arquitecturas: de pila, de acumulador, de registros de propósito general y de registro – registro.
- 3) Escribir por medio de instrucciones la secuencia de código para $D = A + B + C$

Para:

- a) **Arquitectura de Pila.**
 - b) **Arquitectura de Acumulador.**
 - c) **Arquitectura de Registros de Propósito General.**
- 4) ¿Qué es una arquitectura superescalar?
 - 5) Investigar los modos de direccionamiento. Dar un ejemplo por cada modo.
 - 6) Investigar características sobre las siguientes arquitecturas:



■ Motorola

■ HP

■ INTEL

5 PISTAS DE APRENDIZAJE

Traer a la memoria Las compuertas lógicas Exor y Nexor sólo poseen dos entradas.

Traer a la memoria En los mapas de Karnaugh al agrupar la mayor cantidad de unos posibles se obtiene la expresión lógica más simplificada.

No olvide La conexión de flip flops permite construir contadores digitales.

Traer a la memoria La tabla correspondiente a cada compuerta lógica resulta teniendo en cuenta su operador lógico.

Tenga Presente Los circuitos secuenciales digitales tienen memoria.

Traer a la memoria Las instrucciones toman valores que equivalen a la función que debe realizar el microprocesador.

Tenga presente La memoria es un dispositivo que almacena datos en forma binaria.

No olvide La Unidad Central de Procesamiento está formada por la unidad de control y la unidad aritmético lógica.

6 GLOSARIO

Álgebra booleana: Son las Matemáticas de los Sistemas digitales. Utiliza variables lógicas (Toman valores de 0 ó 1) y operadores lógicos (Yes, Not, And, Or, Nand, Nor, Exor, Nexor).

Arquitecturas clásicas: Son aquellas arquitecturas que se desarrollaron en las primeras computadoras electromecánicas y de tubos de vacío. Aun son usadas en procesadores empotrados de gama baja y son la base de la mayoría de las arquitecturas modernas.

Arquitectura de Acumulador: Es aquella en la cual un operando esta implícitamente en el acumulador siempre leyendo e ingresando dato.

Arquitectura de Computador: Es la estructura funcional de un computador.

Arquitectura de Pila: Es aquella en la cual no es necesario nombrar a los operandos ya que estos se encuentran en el tope de la pila.

Arquitectura de Registros de Propósito General: Es aquella en la cual los registros se utilizan para almacenar datos o direcciones de forma flexible.

Arquitectura de Registro - Registro: Es aquella en la cual se trabaja solo con los datos de los registros con el fin de evitar el acceso frecuente a memoria, lo cual ayuda a mejorar la velocidad, ya que, la memoria es más lenta.

Arquitectura Harvard: Es aquella en donde el programa se almacena como un código numérico en la memoria, pero no en el mismo espacio de memoria ni en el mismo formato que los datos.

Arquitectura ISA: Es aquella que detalla las instrucciones que una unidad central de procesamiento puede entender y ejecutar, o el conjunto de todos los comandos implementados por un diseño particular de una CPU.

Arquitectura Von Newman: Es aquella en donde la unidad central de procesamiento se comunica a través de un solo bus con la memoria en la cual se almacenan tanto los códigos de instrucción del programa, como los datos que serán procesados por este.

Buses: Son circuitos que conectan a la unidad central de procesamiento con el resto de los componentes del computador.

Circuito lógico (o digital): Es aquel que maneja la información utilizando el sistema binario (0 y 1).

Codificador: Es un circuito combinacional. Se obtiene en la salida el código binario correspondiente a la entrada que se activa. (Colocada en uno).

Componentes digitales: Son aquellos dispositivos que forman parte de un circuito electrónico digital.

Compuertas digitales: Son funciones básicas de los sistemas electrónicos digitales; trabajan con números binarios.

Compuerta digital AND: Es aquella que realiza la operación denominada multiplicación lógica.

Compuerta digital EXOR: Es aquella que realiza la operación de comparación (Dos entradas). Se obtiene un nivel alto en la salida cuando las entradas son diferentes.

Compuerta digital NAND: Es aquella que realiza la operación inversa de la compuerta AND. Se puede utilizar como compuerta universal ya que se pueden combinar varias de éstas compuertas para implementar el resto de las compuertas lógicas básicas (Por ejemplo: NOT, AND, OR, EXOR, etc.)

Compuerta digital NEXOR: Es aquella que realiza la operación de comparación (Dos entradas). Se obtiene un nivel alto en la salida cuando las entradas son iguales.

Compuerta digital NOR: Es aquella que realiza la operación inversa de la compuerta OR.

Compuerta digital NOT: Es aquella que invierte el valor de la entrada (Operación de negación o complemento)

Compuerta digital OR: Es aquella que realiza la operación denominada suma lógica.

Compuerta digital YES: Es aquella que amplifica la señal. Se obtiene en la salida el mismo bit de entrada.

Computador: Es una máquina digital electrónica que recibe y procesa información automáticamente a través de programas informáticos.

Decodificador: Es un circuito combinacional. Dado el código binario en la entrada se activa (Coloca en uno) la salida correspondiente a dicho código.

Demultiplexor: Es un circuito combinacional que permite seleccionar a cuales de las salidas (mediante selectores) pasará la entrada.

Elementos funcionales: Son aquellos elementos que forman parte de una estructura de orden superior (En éste caso el computador)

Flip Flop: Es un circuito secuencial que está conformado por compuertas lógicas. Se conoce como un dispositivo de memoria.

Mainframe: Conocida como computadora central. Es un gran computador capaz de realizar el procesamiento de datos complejos. Se utilizan como sistemas centrales de grandes organizaciones o empresas.

Mapas de Karnaugh: Método sistemático de simplificación de expresiones lógicas.

Método: Modo ordenado y sistemático de proceder para llegar a un resultado o fin determinado.

Microcomputadora: Es un tipo de computadora que utiliza un microprocesador como unidad central de procesamiento. Generalmente son computadoras que ocupan espacios físicos pequeños

Multiplexor: Es un circuito combinacional que permite seleccionar de las entradas (mediante selectores) cuál de ellas pasará a la salida.

Propiedades: Son reglas que se obtienen a partir de los axiomas y deben ser demostradas mediante estos.

Propiedad asociativa: Enuncia que el resultado en la salida es el mismo, independiente de la forma en que se agrupen las variables de entrada.

Propiedad conmutativa: Enuncia que al cambiar el orden de las variables de entrada, permanece invariable el resultado que se obtiene en la salida.

Propiedad distributiva: Enuncia la relación entre la multiplicación lógica y la suma lógica.

Registro: Es un dispositivo electrónico semiconductor creado a partir de implementaciones con flip flops los cuales permiten el manejo de datos usando circuitos combinatorios y secuenciales específicos.

Reglas: Son los métodos para llevar a cabo una operación (En este caso, operación lógica).

Supercomputadora: Son computadoras de alto desempeño. Extremadamente potentes y capaces de realizar tareas de cálculo a una velocidad que equivale a cientos de veces la velocidad de una computadora convencional.

Unidad Aritmético Lógica: Es un circuito digital que se encarga de realizar las operaciones aritméticas y lógicas que hacen parte del conjunto de operadores que tiene disponibles un computador.

Unidad Central de Procesamiento: Es un circuito integrado que se encarga del control y procesamiento de la información en el sistema digital.

Unidad de Control: Es un circuito digital que se encarga de controlar todas las operaciones del computador.

Unidad de Entrada y Salida: Conjunto de interfaces que usan los dispositivos funcionales de un sistema de procesamiento de datos para comunicarse con los demás medios de información, usando señales recibidas o enviadas.

Unidad de Memoria: Es un circuito digital diseñado para almacenar datos en forma binaria durante un intervalo de tiempo, usando estructuras y arreglos de registros con flip flops.

7 BIBLIOGRAFÍA

- Floyd, T. (2006). Fundamentos de Sistemas Digitales. (9 Ed). Pearson Educación, España.
- Quiroga, P. (2010). Arquitectura de Computadoras. (1 Ed). Alfaomega.
- Stallings, W. (2006). Organización y Arquitectura de Computadores. (7 Ed). Pearson Educación, España.
- Tocci, R, Widmer, N & Moss,G. (2007). Sistemas Digitales. Principios y aplicaciones. (10 Ed). Pearson Educación, México.