

Lógica computacional

JOSE LUIS PEREA VEGA

Red Tercer Milenio

LÓGICA COMPUTACIONAL

LÓGICA COMPUTACIONAL

JOSE LUIS PEREA VEGA

RED TERCER MILENIO



AVISO LEGAL

Derechos Reservados © 2012, por RED TERCER MILENIO S.C.

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

Prohibida la reproducción parcial o total por cualquier medio, sin la autorización por escrito del titular de los derechos.

Datos para catalogación bibliográfica

José Luis Perea Vega

Lógica computacional

ISBN 978-607-733-025-7

Primera edición: 2012

Revisión editorial: Eduardo Durán Valdivieso

DIRECTORIO

José Luis García Luna Martínez
Director General

Jesús Andrés Carranza Castellanos
Director Corporativo de Administración

Rafael Campos Hernández
Director Académico Corporativo

Héctor Raúl Gutiérrez Zamora Ferreira
Director Corporativo de Finanzas

Bárbara Jean Mair Rowberry
Directora Corporativa de Operaciones

Alejandro Pérez Ruiz
Director Corporativo de Expansión y Proyectos

ÍNDICE

INTRODUCCIÓN	4
MAPA CONCEPTUAL	5
UNIDAD 1 COMPUTADORAS	6
MAPA CONCEPTUAL	7
INTRODUCCIÓN	8
1.1. ANTECEDENTES HISTÓRICOS	9
1.2. ARQUITECTURAS	10
1.3. TENDENCIAS ACTUALES	12
AUTOEVALUACIÓN	14
UNIDAD 2 MICROPROCESADORES	16
MAPA CONCEPTUAL	18
INTRODUCCIÓN	19
2.1 ESTRUCTURA INTERNA	20
2.2 PROCESAMIENTO NUMÉRICO	20
2.3 PROCESAMIENTO LÓGICO	25
2.4 SISTEMAS DIGITALES	27
2.5 CIRCUITOS COMBINACIONALES	28
2.6 PROGRAMACIÓN Y CONTROL	31
AUTOEVALUACIÓN	44
UNIDAD 3 MEMORIAS	46
MAPA CONCEPTUAL	47
INTRODUCCIÓN	48
3.1 CIRCUITOS SECUENCIALES	49
3.2 SINCRONÍA	52
3.3 TIPOS DE MEMORIAS	52

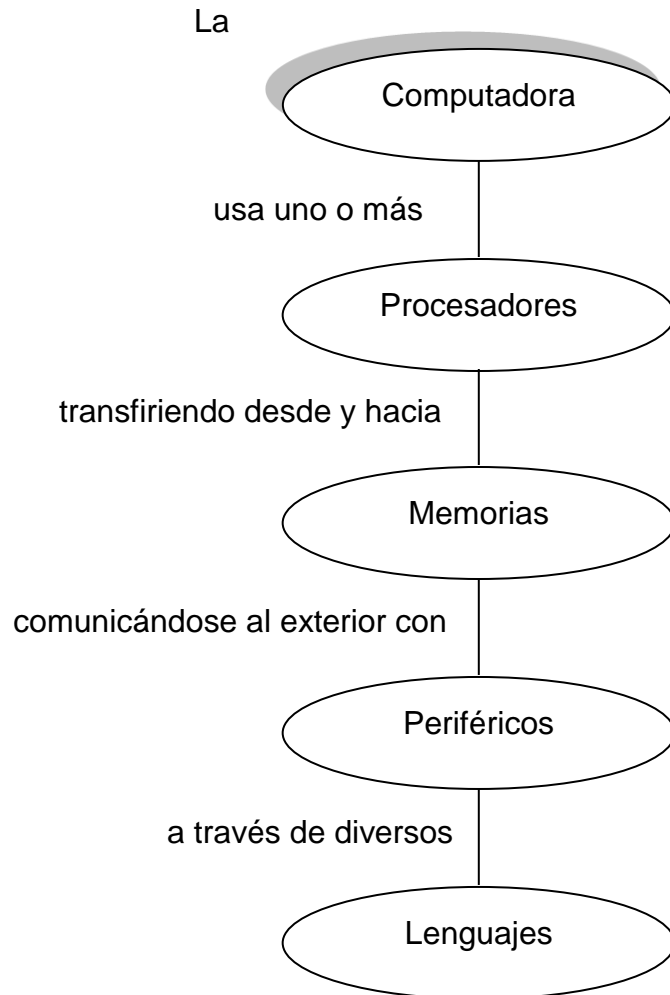
3.4	ADMINISTRACIÓN DE LA MEMORIA	54
3.5	MÁQUINAS DE ESTADO FINITO	56
	AUTOEVALUACIÓN	57
	UNIDAD 4 PERIFÉRICOS	59
	MAPA CONCEPTUAL	60
	INTRODUCCIÓN	61
4.1	SEÑALES ANALÓGICAS Y DIGITALES	62
4.2	CODIFICADORES Y DECODIFICADORES	64
4.3	MULTIPLEXORES	65
4.4	BUSES	66
	AUTOEVALUACIÓN	74
	UNIDAD 5 DISPOSITIVOS LÓGICOS PROGRAMABLES	76
	MAPA CONCEPTUAL	77
	INTRODUCCIÓN	78
5.1	CLASIFICACIÓN	79
5.2	ARQUITECTURA	81
5.3	VHDL	82
	AUTOEVALUACIÓN	107
	BIBLIOGRAFÍA	109
	GLOSARIO	110

INTRODUCCIÓN

El robot *Spirit* aterrizó en Marte el 4 de julio de 2004, contando con equipo capaz de procesar imágenes, analizar rocas y desplazarse a través de la superficie del planeta, entre otras funciones. Para poder enviar una máquina con alto grado de autonomía fuera de la Tierra se ha requerido una gran evolución científica y tecnológica desde que Aristóteles en la antigua Grecia estudiaba las formas de razonamiento y los juicios que generan, lo que forma parte de la lógica.

La producción de dispositivos electrónicos como las computadoras, cuya finalidad es la de procesar información, ha hecho posible la ampliación del conocimiento humano, mientras se sigue tratando de descubrir y replicar los procesos que lo hacen posible. Se continúa buscando la forma de dotar al ordenador de su propia lógica que le permita percibir y entender el medio ambiente que lo rodea, utilizando para tal fin toda la tecnología disponible.

MAPA CONCEPTUAL



UNIDAD 1

COMPUTADORAS

OBJETIVO:

El estudiante comprenderá la evolución y la estructura de los mecanismos de cálculo analizando los cambios tecnológicos de cada etapa.

TEMARIO

1.1 ANTECEDENTES HISTÓRICOS.

1.1.1. DISPOSITIVOS MECÁNICOS.

1.1.2. GENERACIONES DE COMPUTADORAS.

1.1.3. LEY DE MOORE.

1.2 ARQUITECTURAS

1.2.1 BABBAGE.

1.2.2 VON NEUMANN.

1.2.3 TAXONOMÍA DE FLYNN.

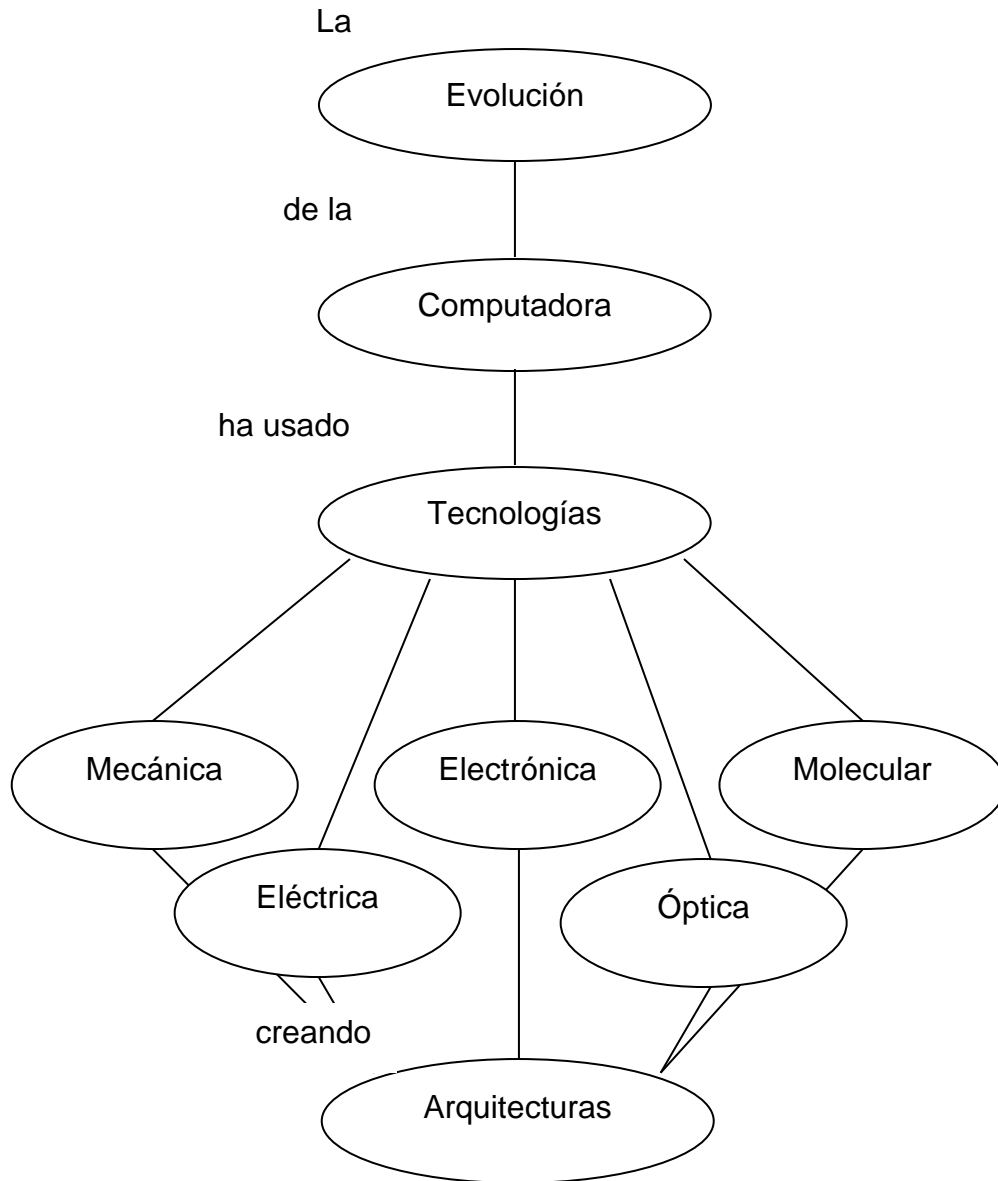
1.3 TENDENCIAS ACTUALES.

1.3.1 COMPUTACIÓN ÓPTICA.

1.3.2 COMPUTACIÓN CUÁNTICA.

1.3.3 COMPUTACIÓN MOLECULAR.

MAPA CONCEPTUAL



INTRODUCCIÓN

El desarrollo de la computadora ha involucrado una gran inversión de recursos materiales y humanos, el primer detonador fue la segunda guerra mundial y el segundo lo representó la masificación de los equipos. En esta unidad se presentan las principales aportaciones de ambas etapas.

1.1 ANTECEDENTES HISTÓRICOS

1.1.1. *Dispositivos mecánicos*

La necesidad de automatizar operaciones aritméticas motivo la creación de mecanismos que sirvieron como base para las computadoras digitales. Se han requerido múltiples contribuciones hasta llegar a los equipos con los que contamos actualmente.

Leibniz. Matemático alemán creador en 1691 de una máquina basada en engranes y cilindros capaz de sumar, restar, multiplicar, dividir y obtener raíz cuadrada. También desarrolló el sistema de numeración binario.

Babbage. Matemático inglés quien fabricó en 1833 la “máquina diferencial” para calcular logaritmos y funciones trigonométricas.

Zuse. Ingeniero alemán que construyó en 1936 la Z1 usando hojas metálicas, la cual trabajaba con sistema binario, pudiendo representar números con decimales.

1.1.2. *Generaciones de computadoras*

A finales de la segunda guerra mundial se puso el mayor énfasis en la construcción de computadoras, dando inicio a una carrera de avances tecnológicos.

Primera generación (1946-1958). Desarrollada con tubos de vacío, de gran tamaño y altos costos de construcción. Se usaron en universidades para fines científicos y militares.

Segunda generación (1959-1964). El uso del transistor marcó el inicio de esta época. Permitió la reducción de espacio y un menor precio de los equipos, lo que hizo posible su adquisición por empresas para dar soporte a sus operaciones comerciales.

Tercera generación (1965-1970). La aparición de circuitos integrados representó el punto de partida para lograr una drástica reducción de consumo de energía, aprovechamiento del espacio y costos menores, logrando que más organizaciones tuvieran acceso a una computadora.

Por haber desarrollado los transistores y circuitos integrados, sus creadores recibieron el premio nobel de física.

Cuarta generación (1971-1988). Gracias a la invención del microprocesador pudieron comercializarse equipos de cómputo personal y surgieron empresas como Apple y Microsoft.

Quinta generación (1988-). A diferencia de las etapas anteriores en las cuales los componentes electrónicos fueron evolucionando, en esta se pone énfasis en el manejo de la información y el conocimiento. Se emplea la inteligencia artificial y se desarrollan sistemas expertos.

En algunos casos se habla de una sexta generación a partir de 1990 cuya característica es el uso de múltiples procesadores trabajando al mismo tiempo.

1.1.3. Ley de Moore

Gordon Moore fundó Intel en 1968 junto con Robert Noyce. En su versión original (1965) la ley de Moore afirma que el número de transistores por pulgada dentro de los circuitos integrados se duplicaría cada año durante dos décadas.

ACTIVIDAD DE APRENDIZAJE

Elabora una tabla comparativa entre los mecanismos de cálculo creados por Napier, Pascal, Schickard y Leibniz.

1.2 ARQUITECTURAS

1.2.1 Arquitectura de Babbage

El proyecto más ambicioso de Babbage fue la creación de la “máquina analítica” para la resolución de problemas aritméticos, constaba de mecanismos que permitían almacenamiento, operaciones aritméticas, transferencias de datos, entrada y salida de los mismos. Su diseño lo inició en 1833 y no pudo ser llevado a cabo debido a las restricciones tecnológicas de la época.

A él se le conoce como el padre de la computadora, ya que los bloques funcionales que propuso se siguen usando en la actualidad.

1.2.2. Arquitectura de von Neumann

John von Neumann fue un matemático húngaro que participó en el desarrollo de la bomba atómica. Elaboró el diseño de una computadora con cinco bloques que permitían el almacenamiento de programas evitando la reconfiguración manual para cada uno, su arquitectura consta de unidades para operaciones aritméticas, control, entrada/salida, memoria y canales de comunicación.

1.2.3. Taxonomía de Flynn

En 1966 Michael Flynn establece una clasificación de computadoras de acuerdo a las características del flujo de información, ya que por los canales de comunicaciones pueden pasar datos e instrucciones.

Una instrucción, un dato (SISD). Este es el caso de la arquitectura de von Neumann la cual posee una lista de instrucciones que procesan una sola secuencia de datos.

Múltiples instrucciones, un dato (MISD). Pocas veces utilizado.

Una instrucción, múltiples datos (SIMD). La misma lista de instrucciones es utilizada por diferentes secuencias de datos de forma simultánea, a esto se le conoce como procesamiento vectorial y fue la base sobre la que se construyeron las primeras supercomputadoras como la Cray-1. Actualmente esta tecnología es poco utilizada ya que su consumo de energía es muy alto.

Múltiples instrucciones, múltiples datos (MIMD). Distintos procesadores trabajan sobre diversas listas de instrucciones con datos diferentes, lo cual es denominado procesamiento paralelo que puede hacer uso de memoria compartida o distribuida. Los *Clusters* son ejemplos de dicha arquitectura, constan de dos o más computadoras comunicadas entre sí y que cuentan con un mecanismo de control.

ACTIVIDAD DE APRENDIZAJE

Investiga cuáles son las primeras diez súper computadoras del Top 500 y qué arquitectura utilizan.

1.3 TENDENCIAS ACTUALES

1.3.1 *Computación óptica*

El primer procesador óptico fue creado por AT&T en 1990, funciona a través de láseres semiconductores para transportar la información, así como circuitos ópticos para su almacenamiento y procesamiento. El uso de la luz permite velocidades considerablemente mayores y trabajo en paralelo.

1.3.2 *Computación cuántica*

En 1982 Richard Feynman, físico estadounidense quien también participó en el Proyecto Manhattan, ideó la posibilidad de construir una computadora cuántica cuando fracasó tratando de simular fenómenos de este tipo en una computadora digital. En lugar de manejar bits se utilizan bits cuánticos o qubits que pueden tomar valores de cero, uno o ambos al mismo tiempo. Al poder tener múltiples valores simultáneamente es posible la realización de operaciones en paralelo, lo que incrementa exponencialmente la capacidad de cálculo.

1.3.3 *Computación molecular*

Se basa en la representación de la información a través de moléculas de ADN, para después hacerlas reaccionar en un tubo de ensayo. Al igual que con la computación cuántica, la estructura de la doble hélice permite el procesamiento en paralelo. Las moléculas de ADN están compuestas por nucleótidos (adenina, timina, citosina y guanina) que al combinarlos pueden representar diversos problemas, el primero en realizar operaciones matemáticas mediante esta técnica fue Leonard Adleman en 1994.

John von Neumann fue pionero en la combinación de computación con biología molecular al proponer un modelo automático de los procesos de reproducción en los seres vivos denominado autómeta celular. Dicho modelo involucra un arreglo de células con cierta configuración inicial las cuales después de una serie de etapas de cálculo pueden obtener una copia de sí mismas.

ACTIVIDAD DE APRENDIZAJE

Crea una lista de los ganadores del premio nobel de física involucrados en el desarrollo de la computadora.

AUTOEVALUACIÓN

Elige el inciso correcto.

A. En esta generación las computadoras fueron construidas con bulbos.

a) Primera b) Tercera c) Quinta d) Sexta

B. ¿Qué tipo de computadora maneja qubits en lugar de bits?

a) Cuántica b) Celular c) Molecular d) Óptica

C. Desarrolló el sistema binario de numeración.

a) Leibniz b) Babbage c) Feynman d) Pascal

D. ¿En qué categoría se encuentra la arquitectura de von Neumann?

a) SIMD b) MISD c) SISD d) MIMD

E. Es conocido como el padre de la computadora.

a) Bill Gates b) Babbage c) Steve Jobs d) Leibniz

Completa las frases.

F. Por los canales de comunicaciones pueden pasar _____.

G. Un *Cluster* está formado por _____.

H. Los bloques funcionales de una arquitectura de von Neumann son _____

_____.

I. Tanto la computadora cuántica como la molecular pueden hacer procesos en

_____.

J. La Ley de Moore asegura que _____.

RESPUESTAS DE LA AUTOEVALUACIÓN

A. a.

B. a.

C. c.

D. b.

E. b.

F. Datos e instrucciones.

G. Constan de dos o más computadoras comunicadas entre sí y que cuentan con un mecanismo de control.

H. Unidades para operaciones aritméticas, control, entrada/salida, memoria y canales de comunicación.

I. Paralelo.

J. El número de transistores por pulgada dentro de los circuitos integrados se duplicaría cada año.

UNIDAD 2

MICROPROCESADORES

OBJETIVO:

El alumno comprenderá la función de los bloques arquitectónicos de la computadora así como los métodos internos de operación del microprocesador.

TEMARIO

2.1 ESTRUCTURA INTERNA

2.2 PROCESAMIENTO NUMÉRICO.

2.2.1 SISTEMAS DE NUMERACIÓN.

2.2.1.1 CONVERSIONES.

2.2.1.2 OPERACIONES.

2.2.2 CÓDIGOS.

2.3 PROCESAMIENTO LÓGICO.

2.3.1 DEFINICIONES.

2.3.2 TABLAS DE VERDAD.

2.3.3 ALGEBRA DE BOOLE.

2.4 SISTEMAS DIGITALES.

2.4.1 ELECTRÓNICA BÁSICA.

2.5 CIRCUITOS COMBINACIONALES

2.5.1 COMPUERTAS LÓGICAS.

2.5.1.1 BÁSICAS.

2.5.1.2 UNIVERSALES.

2.5.1.3 MAPAS DE KARNAUGH.

2.6 PROGRAMACIÓN Y CONTROL.

2.6.1 REGISTROS.

2.6.2 PILA.

2.6.3 PROGRAMAS.

2.6.4 CÓDIGO MÁQUINA.

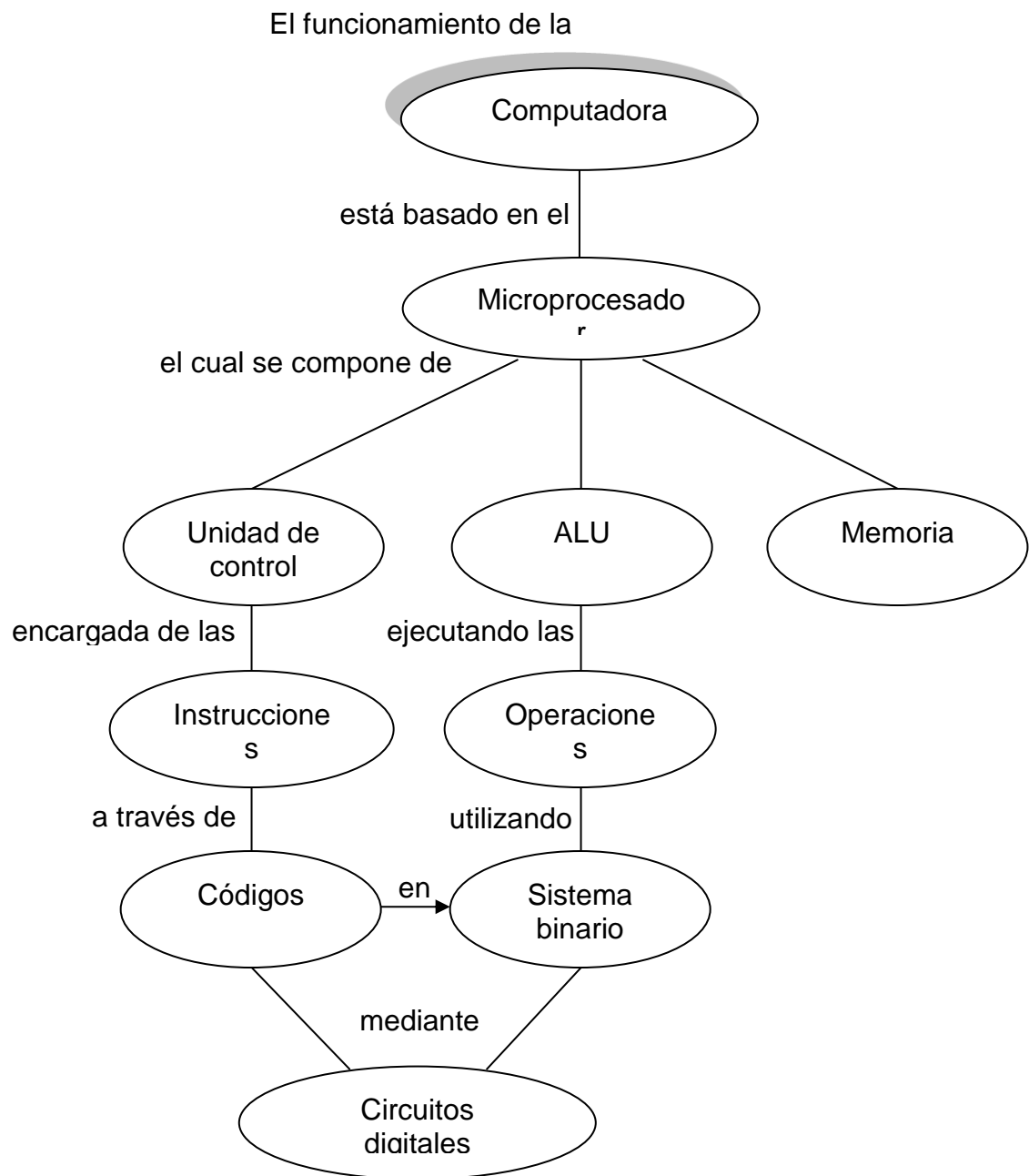
2.6.5 ENSAMBLADOR.

2.6.6 LENGUAJES DE ALTO NIVEL.

2.6.7 RISC Y CISC.

2.6.8 MÁQUINA de TURING.

MAPA CONCEPTUAL



INTRODUCCIÓN

El advenimiento del microprocesador, la reducción de su precio y el incremento del rendimiento, permitieron que las computadoras llegaran a las casas. En esta etapa, algunas de las empresas más importantes hasta hoy también surgieron en hogares y universidades, basta mencionar a MICROSOFT y APPLE. En esta unidad se expondrán los diferentes mecanismos que permiten representar la información y realizar operaciones con ella dentro del procesador.

2.1 ESTRUCTURA INTERNA

De acuerdo con la arquitectura de von Neumann existen cinco bloques funcionales dentro de la computadora.

- ✓ Unidad aritmético-lógica. Realiza las operaciones que la unidad de control le ordena.
- ✓ Unidad de control. Interpreta las instrucciones coordinando la actuación de las unidades internas y externas de la CPU.
- ✓ Subsistema de memoria.
- ✓ Subsistema de entrada-salida.
- ✓ Canales de comunicación.

Las dos primeras conforman la unidad central del proceso (CPU, por sus siglas en inglés), ésta, junto con cierto tipo de memoria, integran el microprocesador.

ACTIVIDAD DE APRENDIZAJE

Investiga cuál es la diferencia entre la arquitectura de von Neumann y la de Harvard.

2.2 PROCESAMIENTO NUMÉRICO

2.2.1 *Sistemas de numeración*

El sistema que se usa cotidianamente tiene como base el número 10 y es posicional, lo que significa que los dígitos que se encuentran a la izquierda tienen un valor más alto. Hagamos una representación del número 5867:

$$\begin{aligned} 5827 &= 5000 + 800 + 20 + 7 \\ &= 5 \times 1000 + 8 \times 100 + 2 \times 10 + 7 \times 1 \\ &= 5 \times 10^3 + 8 \times 10^2 + 2 \times 10^1 + 7 \times 10^0 \end{aligned}$$

Si se acomodan en casillas de acuerdo a los exponentes

3	2	1	0	
				10
5	8	2	7	

El subíndice de lado derecho indica la base, cuando no aparece se asume que es 10.

2.2.1.1 Conversiones

En el sistema decimal hay 10 símbolos para escribir los números mientras que si únicamente se usan dos, se habla de un sistema binario. Empleando nuevamente casillas y exponentes

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ \boxed{} & \boxed{} & \boxed{} & \boxed{} \end{array} _2$$

En cada casilla sólo se puede poner cero o uno. La representación del número 1011_2 sería la siguiente:

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ \boxed{1} & \boxed{0} & \boxed{1} & \boxed{1} \end{array} _2$$

Elevando la base a cada exponente y haciendo la suma

$$1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 1x8 + 0x4 + 1x2 + 1x1 = 11$$

Se ha encontrado que $1011_2 = 11_{10}$. De esta forma se convierten números en base dos a base 10. Para realizar el procedimiento inverso se llevan a cabo los siguientes pasos:

Paso 1. Se traza una línea divisoria y se escribe el número en base 10 en la parte superior izquierda.

$$\begin{array}{r|l} 11 & \end{array}$$

Paso 2. Se divide el número entre dos y se anula el decimal del cociente (si es que lo tiene).

$$\begin{array}{r|l} 11 & \\ 5\text{-}5 & \end{array}$$

Paso 3. Se repite el paso dos hasta llegar a uno.

11	
5	
2	
1	


Paso 4. Se copia el uno a la parte inferior derecha.

11		
5		
2		
1		1

Paso 5. Al lado derecho de cada número impar se escribe uno y si es par, cero.

11		1
5		1
2		0
1		1

Paso 6. Se copian los números del lado derecho de abajo hacia arriba y se acomodan de izquierda a derecha.

11		1		1011 ₂
5		1		
2		0		
1		1		

Con este procedimiento se ha convertido 11_{10} en 1011_2 .

2.2.1.2 Operaciones

Para realizar una suma en base 10 con los números nueve y uno.

1

9 Cada vez que la suma de

$$\begin{array}{r} + 1 \\ \hline 0 \end{array}$$
 dos números es 10 o más
 se dice que “llevamos uno”
 poniéndolo arriba a la
 izquierda de los sumandos.

$$\begin{array}{r} 9 \\ + 1 \\ \hline 1 0 \end{array}$$

De la misma forma, en sistema binario cada vez que la suma de dos números es igual a la base, escribimos cero debajo de los sumandos y “llevamos uno”.

$$\begin{array}{r} 1 \\ 1 \\ + 1 \\ \hline 1 0_2 \end{array}$$

Si hay tres sumandos o más, la suma se obtiene trabajando con dos a la vez. Ejemplo:

$$\begin{array}{r} 1 \\ 1 \\ + 1 \\ 1 \\ \hline 2 \end{array}$$

$$\begin{array}{r} 1 \\ 1 \\ + 1 \\ \hline 0_2 \end{array}$$

1

$$\begin{array}{r}
 + 1 \\
 \hline
 1 \ 0 \ 2 \\
 + 1 \\
 \hline
 1 \ 1 \ 2
 \end{array}$$

$$\begin{array}{r}
 1 \ 1 \\
 1 \ 1 \\
 + 1 \\
 \hline
 1 \ 0 \ 0 \ 2
 \end{array}$$

2.2.2 Códigos

Un código crea una correspondencia entre los elementos de un conjunto con otro.

El sistema binario utiliza el mínimo conjunto de símbolos para representar información. Está formado por cero y uno, a cualquiera de los dos valores lo llamaremos *bit* (*binary digit*). La agrupación de ocho bits la denominaremos *byte*, con el cual se pueden formar 256 combinaciones distintas. Si hacemos la representación usando ocho casillas, la primera puede contener dos números posibles, por cada uno la siguiente tendría otros dos y así sucesivamente.

$$\begin{array}{cccccccc}
 \square & \square & \square & \square & \square & \square & \square & \square \\
 2 & \times & 2 & \times & 2 & \times & 2 & \times & 2 & \times & 2 & \times & 2 & \times & 2 & \times & 2 & \times & = 2^8 = 256 \\
 2
 \end{array}$$

Con n bits se obtienen 2^n combinaciones.

Un código ampliamente usado es el ASCII (*american standard code for information interchange*) que representa mediante números los caracteres del alfabeto latino. Usa siete *bits* para llevar a cabo esta equivalencia.

ACTIVIDAD DE APRENDIZAJE

Investiga el funcionamiento de los códigos Gray y EBCDIC.

2.3 PROCESAMIENTO LÓGICO

2.3.1 Definiciones

La Lógica está basada en afirmaciones que pueden ser falsas o verdaderas. Entre estas oraciones podemos usar los conectores lógicos “y”, “o” y “no”, conocidos como disyunción, conjunción y negación respectivamente.

	Conjunción	Disyunción	Negación
	y (AND)	o (OR)	no (NOT)
Afirmaciones			
Hoy es lunes.	Hoy es lunes y	Hoy es lunes o	Hoy no es lunes.
Está lloviendo.	está lloviendo.	está lloviendo.	No está lloviendo.

La conjunción es verdadera si ambas afirmaciones lo son, mientras que en la disyunción es suficiente con que alguna lo sea.

En el caso de la negación, una afirmación verdadera se vuelve falsa y viceversa.

2.3.2 Tablas de verdad

De forma general podemos presentar los resultados anteriores mediante los siguientes arreglos conocidos como tablas de verdad.

Conjunción		
y	V	F
V	V	F
F	F	F

Disyunción		
o	V	F
V	V	V

F	V	F
---	---	---

Negación	
	no
F	V
V	F

2.3.3 Álgebra de Boole

George Boole matemático inglés desarrolló en 1954 un conjunto de axiomas utilizando los símbolos cero y uno junto con las operaciones \bullet (conjunción) y $+$ (disyunción).

Axioma

1	$B = 0$ si $B \neq 1$	$B = 1$ si $B \neq 0$
	Complemento (negación)	
2	$0' = 1$	$1' = 0$
	Conjunción	Disyunción
3	$0 \bullet 0 = 0$	$1 + 1 = 1$
4	$1 \bullet 1 = 1$	$0 + 0 = 0$
5	$0 \bullet 1 = 1 \bullet 0 = 0$	$1 + 0 = 0 + 1 = 1$

A partir de la tabla anterior se pueden demostrar los siguientes teoremas.

Teorema

Nombre

	Complemento (negación)		
1	$B' = B$		Involución
	Conjunción	Disyunción	
2	$B \bullet 1 = B$	$B + 0 = B$	Identidad
3	$B \bullet 0 = 0$	$B + 1 = 1$	Elemento nulo
4	$B \bullet B = B$	$B + B = B$	Idempotencia
5	$B \bullet B' = 0$	$B + B' = 1$	Complemento

B se denomina variable booleana y puede tomar los valores cero o uno.

Otro matemático inglés llamado Augustus De Morgan propuso dos teoremas importantes.

$$(B_0 \bullet B_1 \bullet B_2 \dots)' = (B_0' + B_1' + B_2' \dots)$$

$$(B_0 + B_1 + B_2 \dots)' = (B_0' \bullet B_1' \bullet B_2' \dots)$$

B_0, B_1, B_2, \dots son variables booleanas.

Para realizar las demostraciones de los teoremas se pueden usar tablas de verdad asignando todos los valores posibles a las variables booleanas.

ACTIVIDAD DE APRENDIZAJE

Investiga cómo se demuestran los teoremas de De Morgan.

2.4 SISTEMAS DIGITALES

Un sistema digital es una combinación de dispositivos diseñados para manejar valores discretos. Se construyen mediante circuitos que contienen compuertas para hacer operaciones lógicas. Claude Shannon, a quien se conoce como el padre de la teoría de la información, demostró en 1937 que existe una equivalencia entre el álgebra de Boole y los circuitos lógicos, estos se construyen a partir de una expresión algebraica que se implementa mediante interconexiones de las compuertas.

2.4.1 Electrónica básica







Antes de analizar las compuertas lógicas repasaremos algunos conceptos importantes sobre electrónica.

Al enunciado que asegura que la corriente eléctrica es directamente proporcional al voltaje e inversamente proporcional a la resistencia, se le conoce como ley de Ohm.

Magnitud	Unidad	Función
----------	--------	---------

Corriente	Voltios (V)	Representa la potencia que permite trabajar.
Resistencia	Ohmios (Ω)	Cuanta oposición al trabajo presenta.
Intensidad	Amperios (A)	Mide la potencia utilizada por unidad de tiempo.

Para realizar los sistemas se requieren varios componentes que dan soporte a las compuertas lógicas como entradas o salidas.

Componente	Función	Características	Símbolo
LED	Diodo que emite luz.	Usa 10mA a 2.5V	
Capacitor	Almacena energía temporalmente.	No se puede conectar invertido.	
Transistor	Amplifica o bloquea la corriente.		
Resistencia	Se opone al flujo de corriente.	Se puede conectar invertido.	
<i>Switch</i>	Controla el flujo de corriente.		
Fuente	Suministra energía.	Menor o igual a 12V	

ACTIVIDAD DE APRENDIZAJE

Investiga cuáles son sus características y cómo funciona un fototransistor.

2.5 CIRCUITOS COMBINACIONALES

Un sistema se puede construir a partir de compuertas lógicas, las salidas que se obtienen dependen de los valores de entrada en un momento dado, no poseen memoria. De esta forma se crean los circuitos combinacionales.

2.5.1 Compuertas lógicas

Los circuitos integrados digitales están formados por resistencias, diodos y transistores, montados en silicón u otro semiconductor conocido como sustrato.

Cada uno de los integrados contiene varias compuertas lógicas, si son menores de 12 son de escala de integración baja. Las principales tecnologías usadas son CMOS (*complementary metal-oxide semiconductor*) y TTL (*transistor-transistor logic*). Los primeros se pueden alimentar de +5V, +3.3V, 2.5V o 1.2V, mientras que los segundos requieren de +5V.

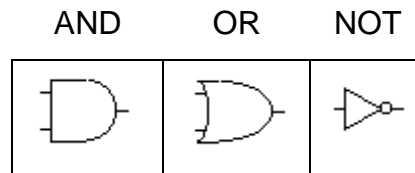
Su presentación más común se conoce como DIP (*dual-in-line package*), la cual pueden tener 14 o más patillas. En caso de que sean 14, ésta sirve para recibir la alimentación y la siete se usa como tierra.

Para representar a los bits dentro de cada compuerta se usan distintos rangos de voltajes. En el caso de TTL son los siguientes:

Nivel alto de entrada	{	5V	1	$V_{IH(max)}$
		2V		$V_{IH(min)}$
			Sin uso	
Nivel bajo de entrada	{	0.8V	0	$V_{IL(max)}$
		0V		$V_{IL(min)}$
			Sin uso	
Nivel alto de salida	{	5V	1	$V_{OH(max)}$
		2.4V		$V_{OH(min)}$
			Sin uso	
Nivel bajo de salida	{	0.4V	0	$V_{OL(max)}$
		0V		$V_{OL(min)}$

2.5.1.1 Básicas

Las tres operaciones lógicas básicas son conjunción, disyunción y negación, cuyas tablas de verdad se vieron en la sección 2.3.2. Los símbolos que se utilizan para representar a cada una son:



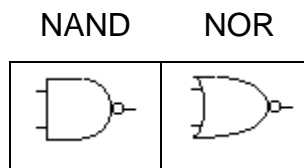
2.5.1.2 Universales

Existen dos compuertas que pueden ser configuradas para funcionar como cualquier otra, se denominan NAND y NOR. La primera produce el opuesto de una salida de AND y la segunda el opuesto de OR.

NAND	0	1
0	1	1
1	1	0

NOR	0	1
0	1	0
1	0	0

Se utiliza la siguiente simbología:



2.5.1.3 Mapas de Karnaugh

Un mapa de Karnaugh presenta todos los posibles valores de entradas y las salidas que generan en un arreglo rectangular parecido a una tabla de verdad, se usa para simplificar las expresiones booleanas.

ACTIVIDAD DE APRENDIZAJE

Investiga cuáles son los símbolos y las tablas de verdad de las compuertas XOR y XNOR.

2.6 PROGRAMACIÓN Y CONTROL

El microprocesador funciona a partir de instrucciones y datos que se guardan en la memoria y se van ejecutando en secuencia, haciendo uso de diversas áreas de almacenamiento dependiendo de la importancia de la información.

2.6.1 Registros

Están contenidos en el microprocesador junto con la CPU, son almacenes temporales de datos que se accesan de la forma más rápida.

2.6.2 Pila

Es una parte de la memoria que sirve para escribir o leer datos que necesita alguna parte de un programa.

2.6.3 Programas

Están formados por instrucciones que le indican a la computadora cómo debe hacer una tarea, se ejecutan de forma secuencial dentro de un procesador.

2.6.4 Código máquina

Es el lenguaje nativo de la computadora, ya que está formado por bytes que representan tareas fundamentales y simples que puede hacer el procesador.

2.6.5 Ensamblador

Está un nivel más arriba que el código máquina, ya que establece una equivalencia entre bytes y cierto tipo de instrucciones comúnmente de tres letras que son más fáciles de recordar y cuyo significado está abreviado, tales como ADD, JMP o MOV.

2.6.6 Lenguajes de alto nivel

Se enfocan a ser entendidos por los seres humanos a través de comandos tomados del idioma inglés, como While, Case o For. Cada uno de estos puede requerir que se ejecuten múltiples instrucciones en código máquina.

2.6.7 RISC y CISC

RISC (*reduced instruction set computer*) es un tipo de arquitectura que permite mantener un pequeño número de instrucciones con hardware menos sofisticado. En el caso de la CISC (*complex instruction set computer*) cada instrucción puede contener gran cantidad de comandos de tipo RISC demandando mayor carga de trabajo para la computadora.

2.6.8 Máquina de Turing

En 1930 el matemático Alan Turing ideó una computadora teórica con memoria infinita representada por una cinta que contenía una secuencia de símbolos de cierto alfabeto que eran procesados, y los resultados que generaban se guardaban al final de la cinta. De este modelo se desprende el término computabilidad, el cual asegura que un proceso hecho en una computadora también funcionaría en una máquina de Turing.

ACTIVIDAD DE APRENDIZAJE

Investiga cuáles son tipos de problemas que no pueden ser resueltos con una máquina de Turing.

PRÁCTICA GUIADA

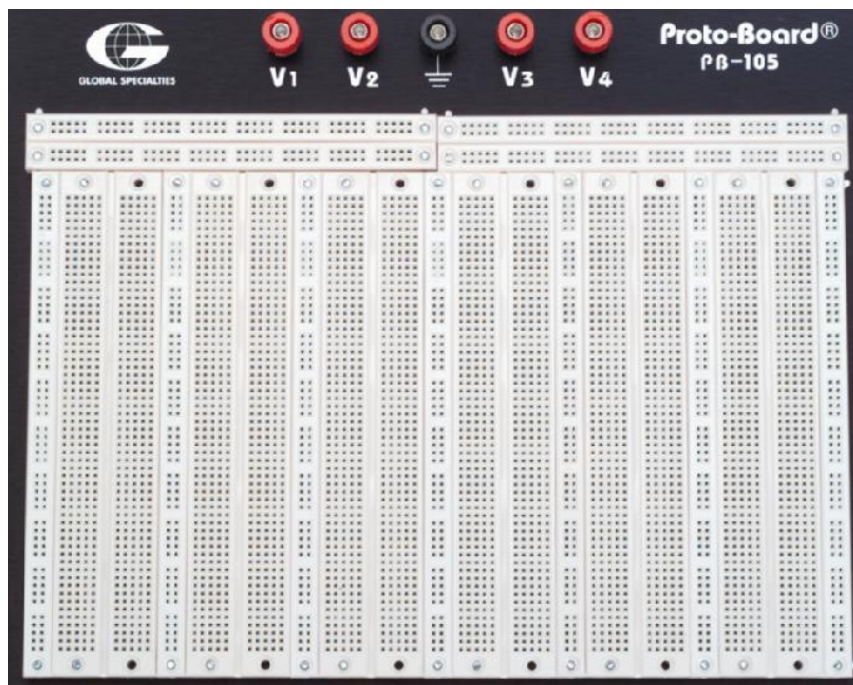
Práctica 1

Objetivo

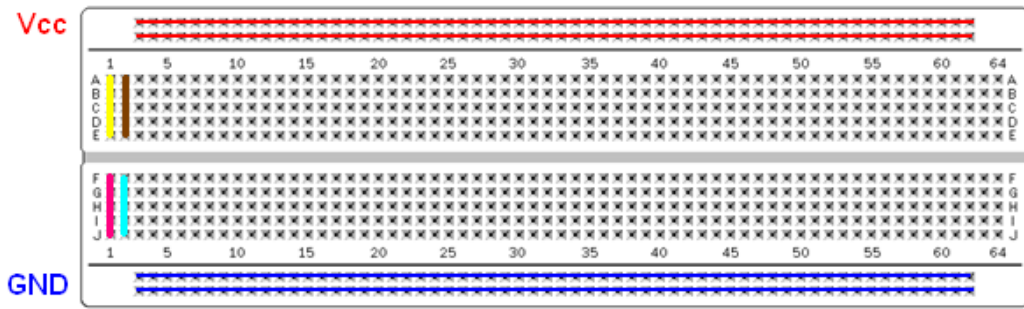
Que el alumno conozca el manejo de *software* para el diseño y pruebas de circuitos, e identifique cuales son las ventajas del mismo. A través de la utilización de un circuito que consta de un integrado de compuertas AND.

Marco teórico

Protoboard. Es una tableta que permite probar diseños de circuitos antes de soldarlos en una placa, los componentes (resistencias, capacitores, circuitos integrados) se meten a presion al igual que el cableado. Existen programas que simulan protoboards simplificando más las tareas de elaboracion y pruebas.

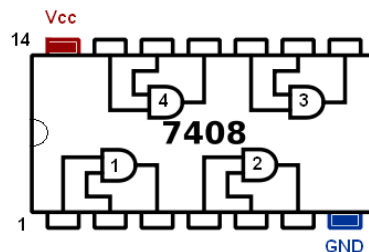


Proto-Board © de Global Specialities

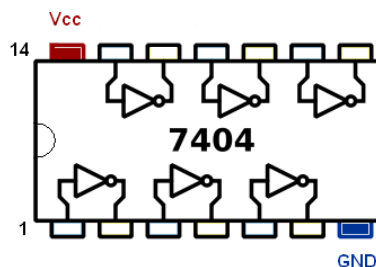


Protoboard en simulador

En la imagen anterior se marcan con diferentes colores las líneas de conexión de tal forma que el voltaje (Vcc) y la tierra (GND) se distribuyen mediante filas, mientras que las bahías para los componentes se comunican por columnas. La división horizontal corresponde a la posición destinada a los circuitos integrados DIP (*dual in-line package*) que tienen el mismo número de patillas dispuestas en forma paralela. Tal es el caso de la familia 7400, dentro de la cual se encuentra el integrado 7408 compuesto por cuatro puertas lógicas AND.



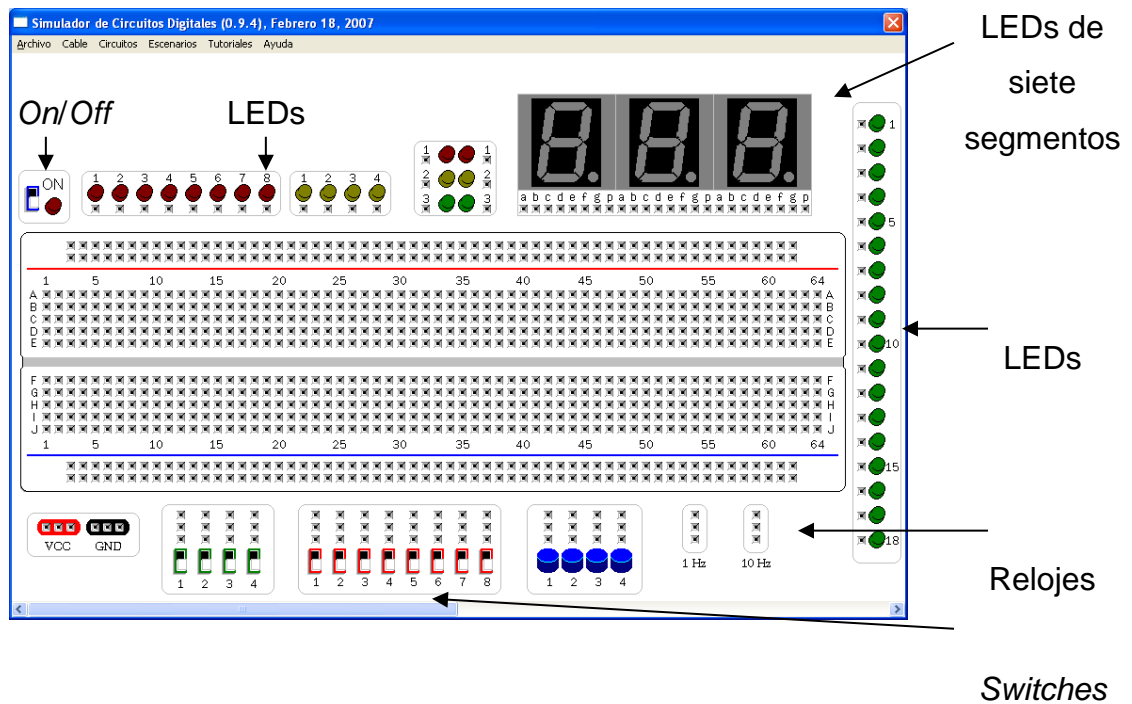
El integrado 7432 tiene cuatro compuertas OR con la misma distribución que el 7408. El circuito 7404 corresponde a seis puertas NOT.



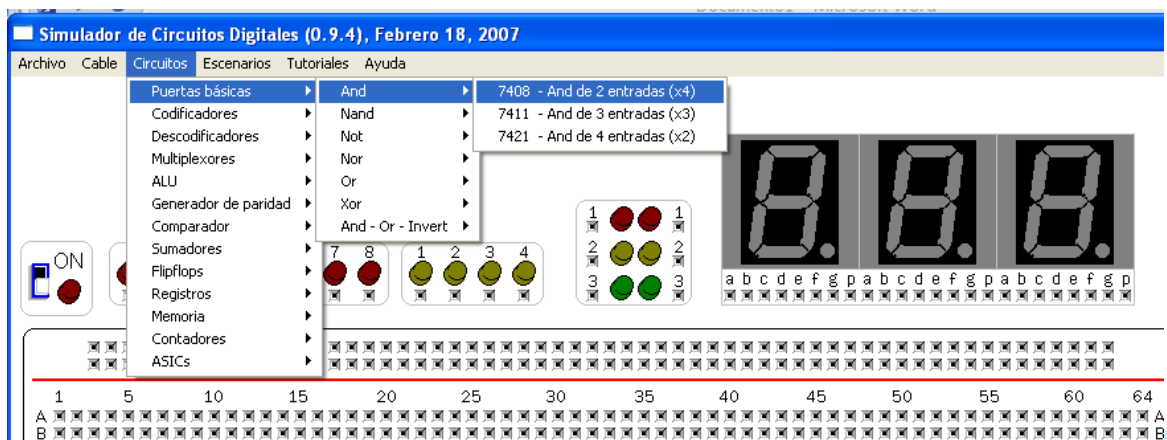
Instrucciones

Paso 1. Descargar el *software libre* Simulador de Circuitos Digitales © creado por Arturo Javier Miguel de Priego Paz Soldán.

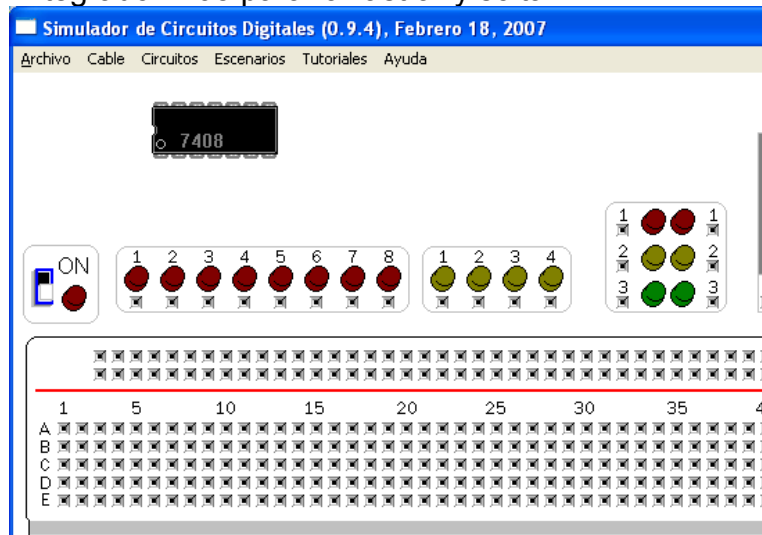
Paso 2. Descomprimir y ejecutar. La interfaz del simulador contiene:



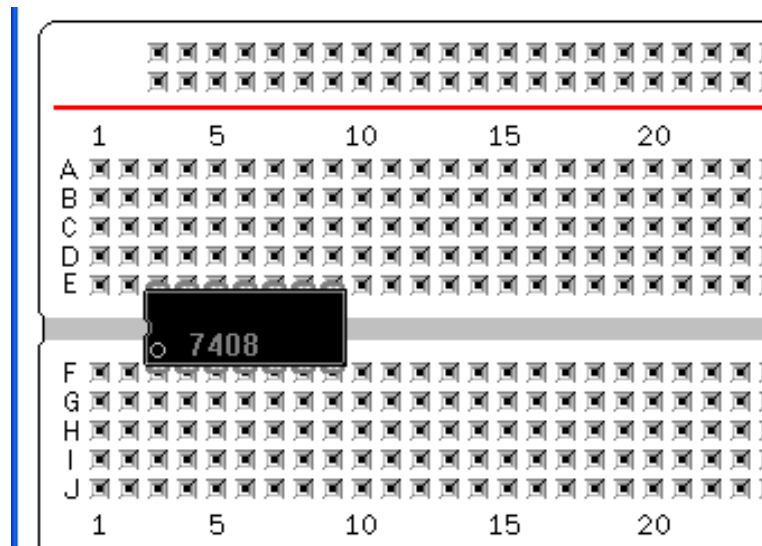
Paso 3. Entrar a Circuitos > Puertas básicas > And > 7408-And 2 entradas > dar *click* con el *mouse*.



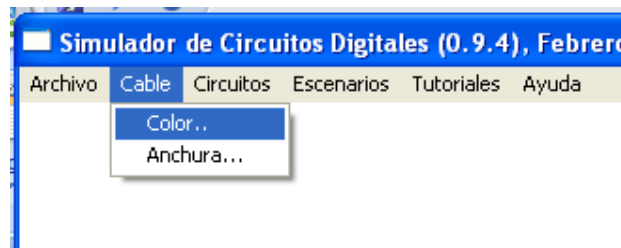
Aparecerá un integrado 7408 para “arrastrar y soltar”.

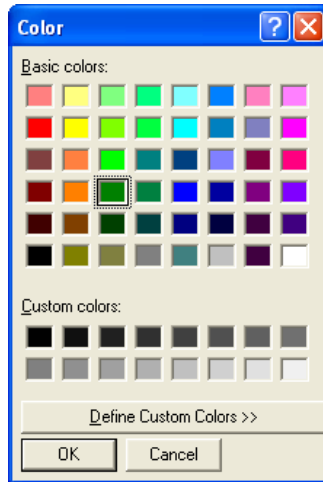


Paso 4. Acomodar la patilla 1 del integrado en la fila F columna 3.

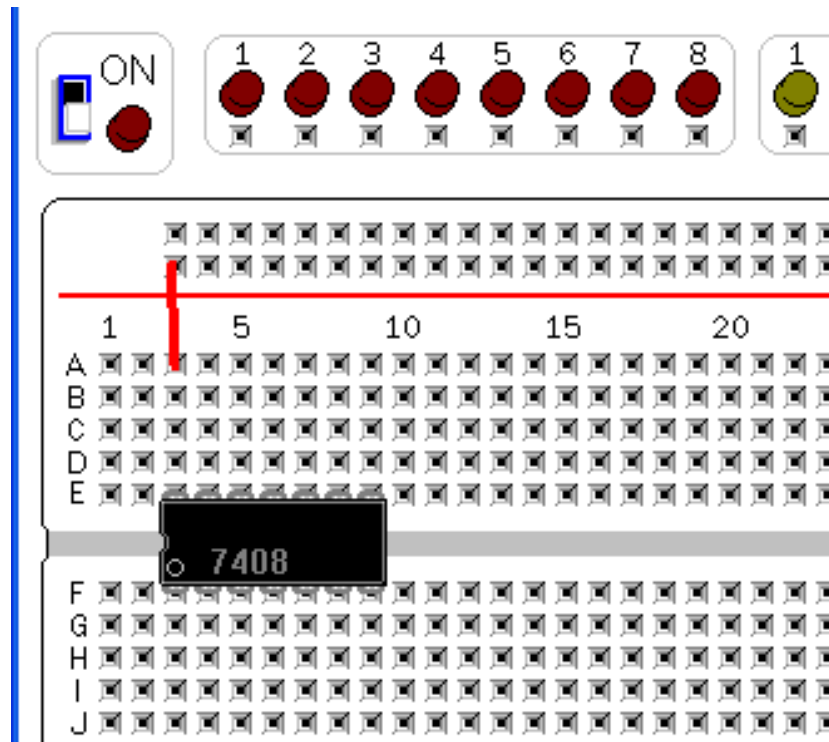


Paso 5. Cablear: ir a Cable > Color > aparece una paleta de colores para elegir el que se necesite mediante un *click* sobre él.

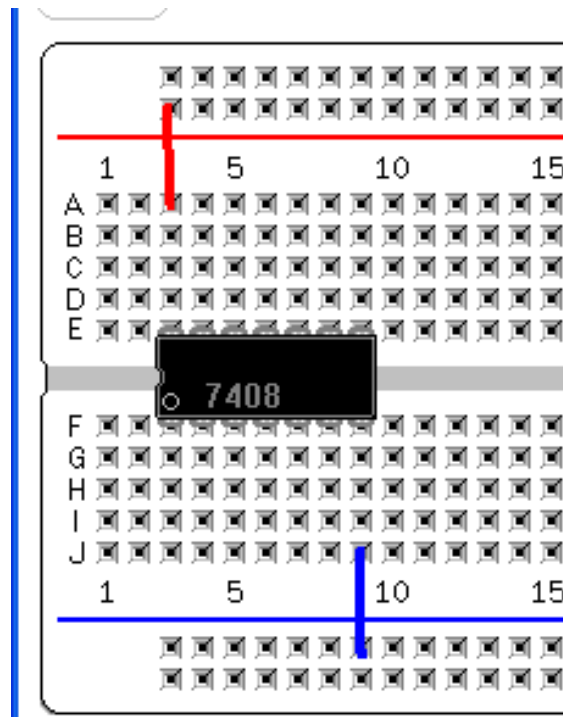




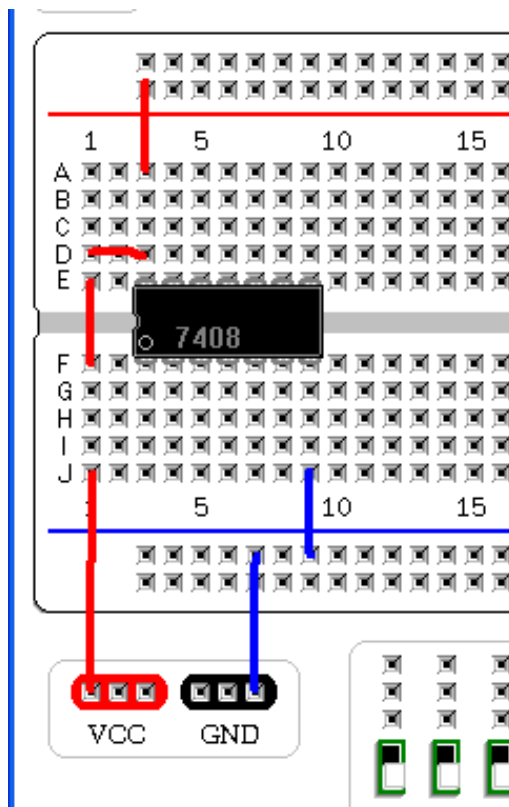
Paso 6. Alimentar: colocar los cables para suministrar corriente al integrado, se da *click* en la fila A columna 3 y sin soltar se lleva hasta la fila superior que es la que corresponde a Vcc.



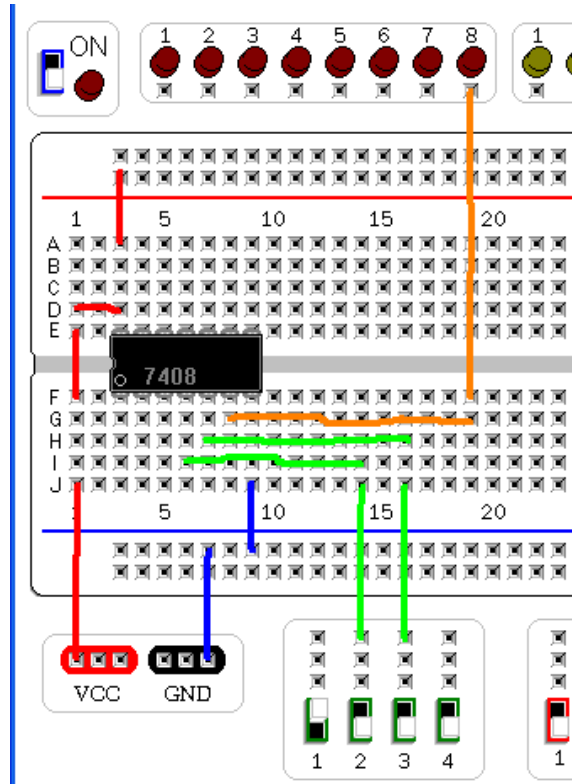
Paso 7. Aterrizar. Se procede de forma análoga al paso anterior para conectarse a GND.



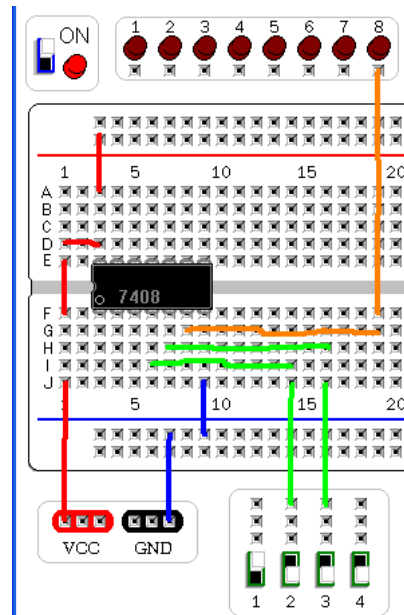
Paso 8. Hasta el momento ya se ha conectado el integrado pero falta alimentar y aterrizar el *protoboard*.



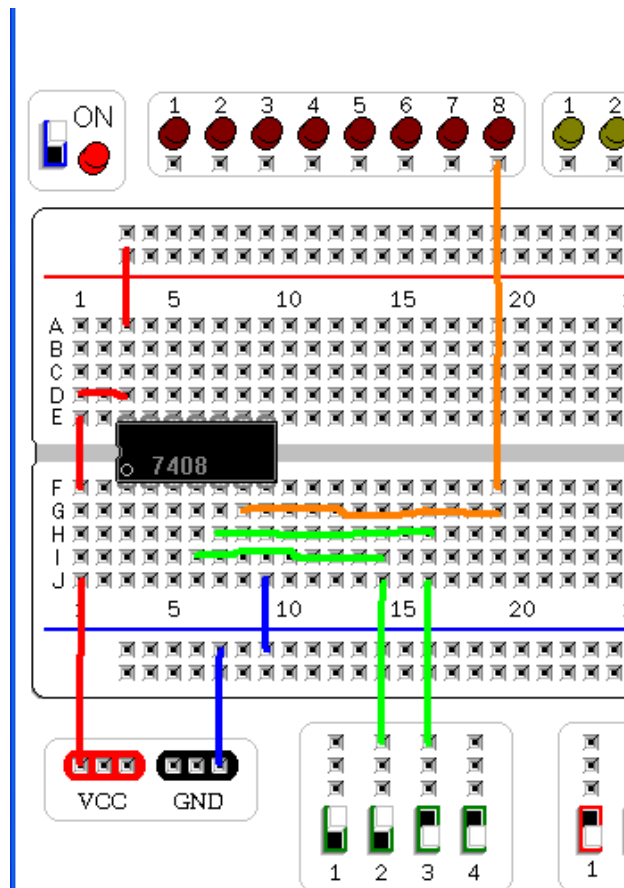
Paso 9. Hacer las conexiones de la compuerta AND, los *switches* suministrarán las entradas y un LED servirá como salida.



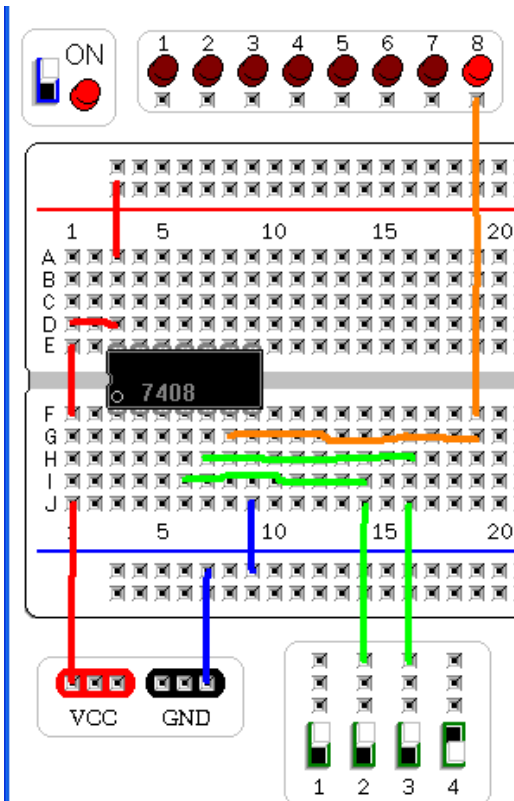
Paso 10. Encender. Poner el interruptor en *on*.



Paso 11. Probar. Cambiar el estado de los *switches* para checar que funcione bien.



Cuando los dos *switches* estén activados el LED se encenderá.

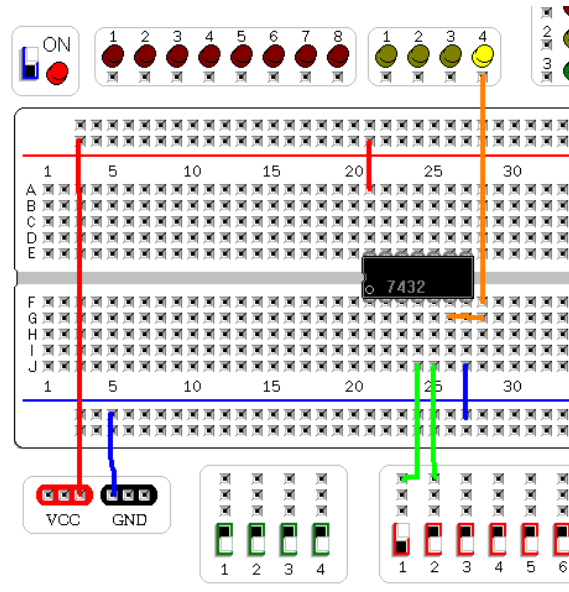


PRÁCTICAS ASIGNADAS

Práctica 1

Objetivo

El alumno construirá un circuito basado en el integrado 7432 para hacer funcionar un LED.

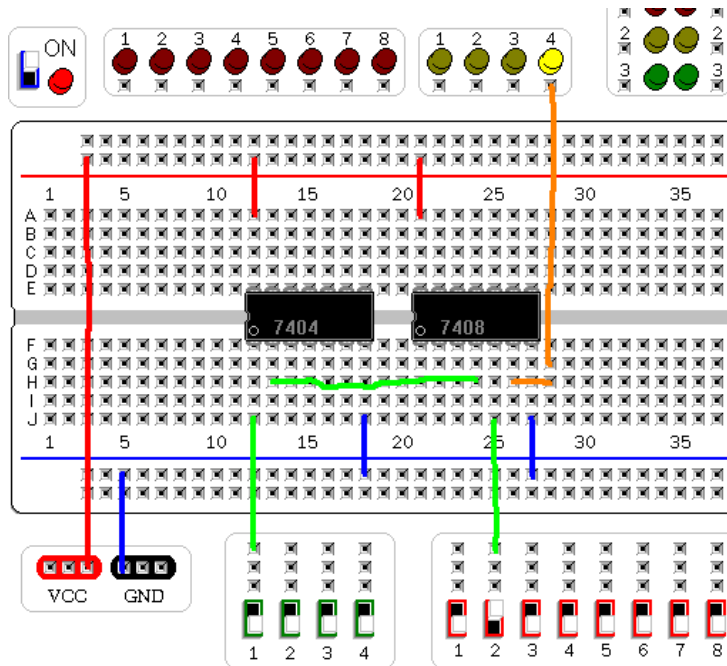


Sugerencia de montaje para el circuito.

Práctica 2

Objetivo

El alumno construirá un circuito basado en los integrados 7404 y 7408 para hacer funcionar un LED.



Sugerencia de montaje para el circuito.

AUTOEVALUACIÓN

Elige el inciso correcto.

A. Se encarga de realizar operaciones dentro del microprocesador.

a) Unidad de control b) ALU c) Entrada d) Registros

B. Es conocido como padre de la teoría de la información.

a) Babbage b) Boole c) Shannon d) De Morgan

C. Es el lenguaje nativo de la computadora.

a) Código máquina b) Ensamblador c) Alto nivel d) Programación

D. Se le considera una compuerta universal.

a) AND b) NAND c) OR d) NOT

E. Es el tipo de memoria más rápida.

a) Registros b) Cache c) RISC d) CISC

Completa las frases.

F. Los mapas de Karnaugh sirven para _____.

G. Para representar uno en TTL se ocupa el rango _____.

H. Un programa es _____

_____.

I. Los circuitos integrados están formados por _____.

J. La función de un capacitor es _____.

RESPUESTAS DE LA AUTOEVALUACIÓN

A. b.

B. c.

C. a.

D. b.

E. a.

F. Simplificar las expresiones booleanas.

G. 2V a 5V.

H. Un conjunto de instrucciones que le indican a la computadora cómo hacer una tarea.

I. Resistencias, diodos y transistores.

J. Almacenar energía temporalmente.

UNIDAD 3

MEMORIAS

OBJETIVO:

El alumno identificará los diferentes tipos de memorias, así como los mecanismos de administración de las mismas.

TEMARIO

3.1 CIRCUITOS SECUENCIALES

3.1.1 LATCHES

3.1.2 FLIP-FLOPS

3.1.3 CONTADORES

3.1.4 TEMPORIZADORES

3.2 SINCRONÍA

3.3 TIPOS DE MEMORIAS

3.3.1 ROM

3.3.2 RAM

3.3.3 SRAM

3.3.4 DRAM

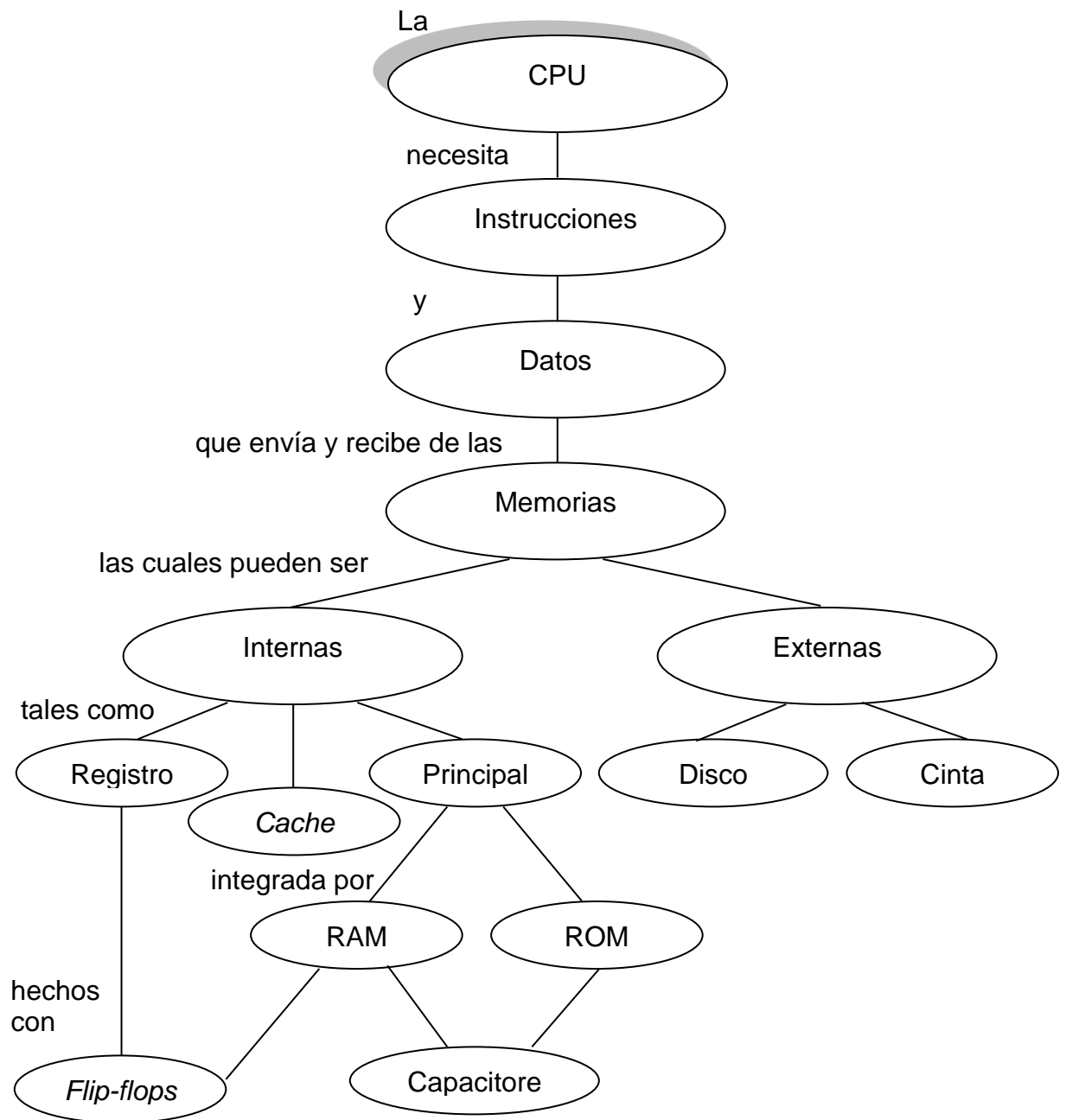
3.4 ADMINISTRACIÓN DE LA MEMORIA

3.4.1 CACHE

3.4.2 VIRTUALIZACIÓN

3.5 MÁQUINAS DE ESTADO FINITO

MAPA CONCEPTUAL



INTRODUCCIÓN

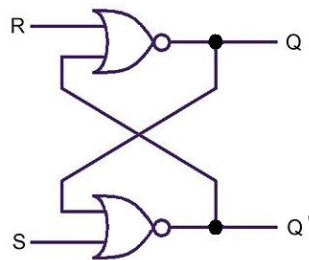
El desarrollo de nuevos componentes electrónicos ha permitido la aparición de sistemas digitales más complejos y eficientes que han hecho posible que las personas estén mejor comunicadas y generen gran cantidad de información que comparten; ante esta tendencia irreversible, se requiere de gran capacidad de almacenamiento a un costo cada vez menor.

3.1 CIRCUITOS SECUENCIALES

En los circuitos secuenciales las salidas dependen de los valores actuales y anteriores, a los cuales se les conoce como variables de estado que influyen en el comportamiento futuro del sistema. Este almacenamiento se logra a través de memorias, cuyas partes fundamentales son elementos binarios o biestables, implementados físicamente mediante semiconductores. Los *latches* y *flip-flops* son sus componentes principales.

3.1.1 Latches

Son dispositivos biestables de memoria temporal. Pueden construirse mediante compuertas lógicas que forman ciclos, a éstos se les conoce como retroalimentación, que convierten las salidas en entradas. La salida se ve reflejada inmediatamente después que se modifican las entradas, sin la necesidad de un mecanismo coordinador, conocido como reloj, el cual indica en qué momento se deben refrescar las variables de estado. Al sistema que no posee señal de reloj se le denominará *asíncrono*. Existen diversos tipos de *latches* entre ellos el SR integrado por dos compuertas NOR, tiene dos entradas: R (*reset*) y S (*set*), así como dos salidas: Q y Q' (complemento de Q).

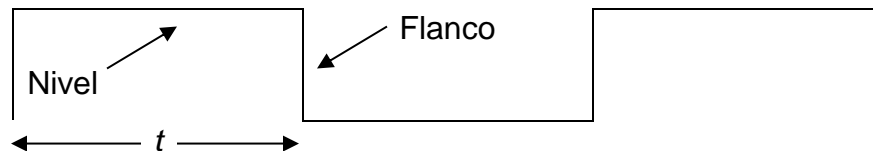


S R Estado

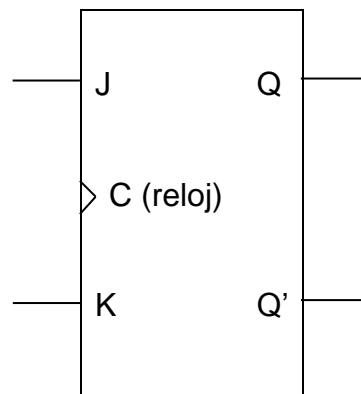
0	0	No cambia
0	1	Q=0
1	0	Q=1
1	1	Indefinido

3.1.2 Flip-flops

Los elementos constitutivos de un *flip-flop* son *latches* junto con una señal de reloj que le proporciona sincronía, esto significa que en ciertos intervalos de tiempo t se actualiza su estado. Existen dos tipos de sincronía: por nivel y por flanco.



Si es por nivel, el sistema actualiza las entradas mientras se encuentre en un nivel activo (bajo o alto). Si se hace por flanco, se toma en cuenta el cambio hacia arriba o de bajada. A los *flip-flops* también se les conoce como multivibradores biestables. Hay diversas clases de *flip-flops*, como ejemplo se puede mencionar el JK cuyo símbolo es:

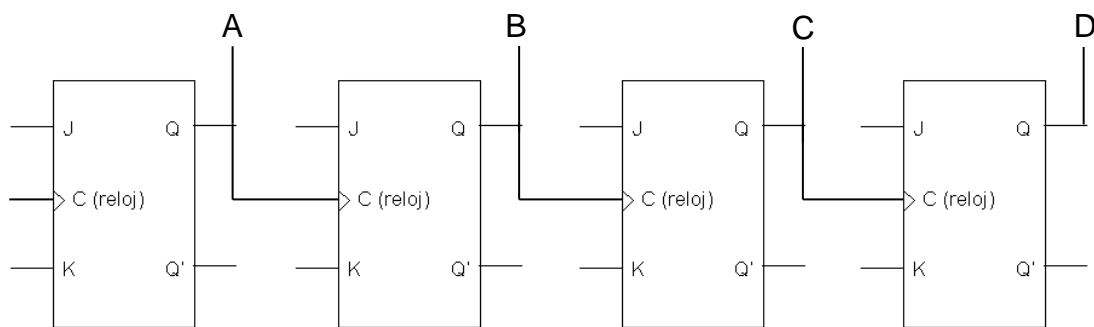


Con la siguiente tabla de estados

J	K	$Q(t+1)$	Estado
0	0	$Q(t)$	Sin cambio
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complemento

3.1.3 Contadores

Los *flip-flops* pueden conectarse entre sí para realizar funciones de conteo, la cantidad usada y su forma de conexión determinan el número de estados (módulo). Para realizar un contador binario se debe conectar la salida Q de cada *flip-flop* con la entrada de reloj del siguiente. A su vez cada salida representa un bit, si tenemos un arreglo de cuatro multivibradores podemos representar del cero al 15 es decir, 0000_2 a 1111_2 , con cada ciclo de reloj que cambia entre cero y uno se produce un incremento unitario.



3.1.4 Temporizadores

El circuito temporizador o de retardo permite la activación o desactivación de algún dispositivo en función del tiempo. El integrado 555 es ampliamente usado, fue desarrollado por Camenzind en 1970, está conformado por aproximadamente 20 transistores, dos diodos y 15 resistencias, encapsulados en un chip de ocho patillas.

El 555 tiene tres modos de configuración: *monostable*, *astable* y *bistable*. En el primero funciona como interruptor, en el segundo tiene el comportamiento de un oscilador, y en el último tiene la funcionalidad de un *flip-flop*.

ACTIVIDAD DE APRENDIZAJE

Busca la hoja de datos del circuito integrado 555 e identifica cómo se establecen los distintos modos de configuración.

3.2 SINCRONÍA

En un circuito síncrono ideal, los cambios se producen simultáneamente de acuerdo con las señales de reloj. En la práctica existe un problema conocido como *race condition*, el cual se presenta cuando dos o más señales tienen que llegar simultáneamente para activar dispositivos y en algún momento provocan fallas, esto se puede solucionar en la fase de diseño de los circuitos.

ACTIVIDAD DE APRENDIZAJE

Investiga cómo se emplea la sincronía trabajando con múltiples procesadores.

3.3 TIPOS DE MEMORIAS

Todos los datos que son procesados por la computadora necesitan ser almacenados en diversos tipos de memorias. Estas se pueden construir usando *flip-flops* como en el caso de los registros, los cuales son los dispositivos de mayor velocidad ya que se encuentran dentro del microprocesador. Los registros requieren de sincronía por nivel o por flanco.

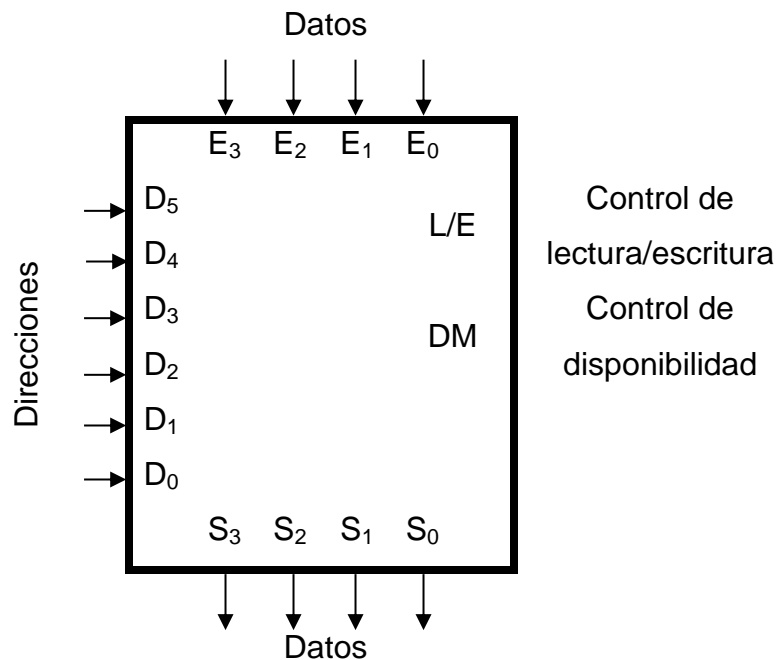
Los datos también se pueden almacenar en capacitores mediante cargas, los cuales son fabricados con semiconductores, proporcionando rapidez y bajo consumo de voltaje. Cada elemento de la memoria que contiene un bit es llamado celda. Las celdas se agrupan en *words* que varían en tamaño.

Después de los registros en orden descendente de acuerdo con su velocidad, se encuentran: *cache*, RAM y ROM. Estas dos últimas conforman la memoria principal.

Existen dispositivos de memoria externa o auxiliar que tienen que pasar por el subsistema de entrada/salida, en esta categoría se encuentran: cintas, CD, DVD, etc.

La comunicación entre la memoria y el microprocesador se logra a través de *buses* formados por cables, estos canales transmiten: direcciones, datos y señales de control.

Las direcciones son números que identifican cierta ubicación en la memoria. Si se toma como ejemplo una memoria que almacene 64 celdas de cuatro bits, se requeriría tener ocho canales para los datos de entrada como de salida, dos asignados al control y seis conteniendo direcciones.



3.3.1 ROM

En la memoria sólo de lectura (ROM, por sus siglas en inglés) la información almacenada tiene la característica de ser permanente o semipermanente. Se graba cuando se fabrica y después puede ser leída sin restricción. Existen diversos tipos como: MROM, PROM, EPROM y EEPROM, éstas últimas

pueden ser borradas y reprogramadas, unas mediante luz ultravioleta y otras a través de electricidad.

3.3.2 RAM

En las primeras computadoras se usaron anillos ferro magnéticos (*cores*) como memorias de acceso aleatorio (RAM, por sus siglas en inglés). Estas mantienen su contenido mientras cuentan con suministro eléctrico, además el tiempo de acceso a cualquier dirección no cambia.

3.3.2 SRAM

La RAM estática (SRAM, por sus siglas en inglés) es un dispositivo digital que se conforma de *flip-flops*. Se usa para memoria *cache*.

3.3.3 DRAM

Una RAM dinámica (DRAM, por sus siglas en inglés) está construida con celdas de capacitores, si están cargados equivalen a uno, en caso contrario se toman como cero.

Ya que los capacitores tienden a perder su carga, requieren ser refrescados en intervalos regulares de tiempo. Se considera como un dispositivo analógico de menor costo y tamaño que la correspondiente SRAM. Por sus características es empleada para la memoria principal de la computadora.

ACTIVIDAD DE APRENDIZAJE

Investiga cuáles son las características de la memoria flash.

3.4 ADMINISTRACIÓN DE LA MEMORIA

Los diferentes métodos de acceso a la memoria son los siguientes: secuencial, directo, aleatorio, asociativo.

- ✓ Secuencial. La memoria está dividida en registros, se revisa uno tras otro de forma lineal hasta encontrar la información buscada.
- ✓ Directo. Se encuentra conformado por bloques que se pueden acceder mediante cierto tipo de coordenadas con un mecanismo de ubicación compartido.
- ✓ Aleatorio. Cada área direccionable de memoria tiene su propia ruta física de acceso, lo que se traduce en un tiempo constante de ubicación de cada dato.
- ✓ Asociativo. Los datos se buscan de acuerdo con su contenido en lugar de especificar su ubicación. Se usa en la memoria *cache*.

3.4.1 Cache

La memoria *cache* tiene el propósito de almacenar la información que está siendo usada con mayor frecuencia por el microprocesador, cada vez que éste requiere algún dato lo busca primero dentro de la cache, si lo encuentra lo transfiere al procesador, si no lo tiene, localiza su ubicación dentro de la memoria principal y copia a la *cache* el bloque completo donde se encuentra, esperando que se requiera algún otro dato dentro de la misma zona, con la finalidad de mejorar el rendimiento.

3.4.2 Virtualización

Debido a que la memoria principal es limitada, se requiere disponer de espacio adicional para proteger la integridad de la información, la parte extra se toma del disco duro que se divide en páginas o segmentos, haciendo el acceso a través de direcciones virtuales, a este proceso se le conoce como mapeo de memoria. La virtualización se usa en DRAMs y en discos duros.

ACTIVIDAD DE APRENDIZAJE

Investiga cuáles son distintos tipos de *cache*.

3.5 MÁQUINAS DE ESTADO FINITO

Los circuitos secuenciales síncronos pueden ser representados a través de autómatas finitos, estos fueron desarrollados por Moore y Mealy en la década de 1950.

Una máquina de estado finito tiene M entradas, N salidas y k bits de estados, haciendo un total de 2^k estados posibles.

También posee dos dispositivos de lógica combinacional y un registro. Existen dos tipos de autómatas: el creado por Moore que toma en cuenta el registro de estado actual, y el Mealy que además considera las entradas actuales.

ACTIVIDAD DE APRENDIZAJE

Investiga cómo se hace la representación gráfica de un autómata finito.

AUTOEVALUACIÓN

Elige el inciso correcto.

A. Es un mecanismo que funciona como coordinador.

a) *Flip-flop* b) Compuertas c) *Latches* d) Reloj

B. Es el método de acceso a memoria más lento.

a) Asociativo b) Secuencial c) Directo d) Aleatorio

C. La luz ultravioleta sirve para borrar un tipo de.

a) ROM b) RAM c) *Cache* d) Sistema

D. Una DRAM está constituida de.

a) *Flip-flops* b) Capacitores c) *Latches* d) Compuertas

E. La cache es una memoria de tipo.

a) Asociativo b) Secuencial c) Directo d) Aleatorio

Completa las frases.

F. La retroalimentación convierte _____.

G. Un contador se puede implementar interconectando _____.

H. Las direcciones son _____.

I. Se copian bloques completos a la *cache* para _____.

J. Los autómatas finitos pueden representar _____.

RESPUESTAS DE LA AUTOEVALUACIÓN

A. d.

B. b.

C. a.

D. b.

E. a.

F. Las salidas en entradas.

G. *Flip-flops*.

H. Números que identifican cierta ubicación en la memoria.

I. Mejorar el rendimiento.

J. Circuitos secuenciales síncronos.

UNIDAD 4

PERIFÉRICOS

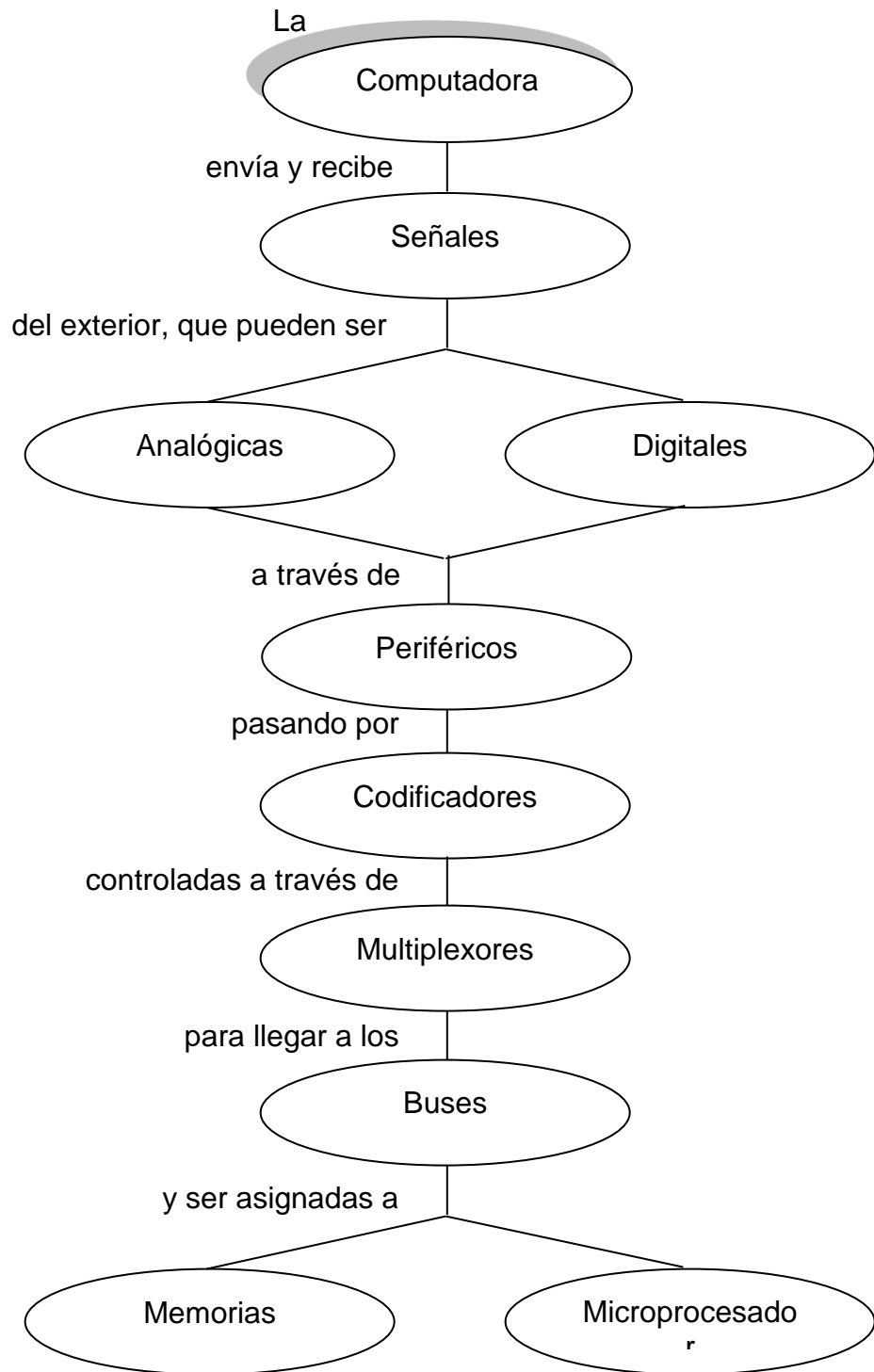
OBJETIVO:

El alumno conocerá a detalle las etapas de que permiten la comunicación de información entre la computadora y su medio ambiente.

TEMARIO

- 4.1 SEÑALES ANALÓGICAS Y DIGITALES
- 4.2 CODIFICADORES Y DECODIFICADORES
- 4.3 MULTIPLEXORES
- 4.4 BUSES

MAPA CONCEPTUAL



INTRODUCCIÓN

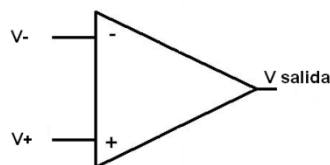
Un periférico es un dispositivo externo conectado a la computadora a través del cual interactúa con su medio ambiente. Uno de los principales problemas al realizar las transferencias de información entre uno y otra, es la diferencia de velocidades para: procesar, convertir y transmitir los datos. Computadoras más rápidas requieren de nuevas tecnologías de interconexión para dar soporte a una sociedad que se acostumbra cada día más a la comunicación global.

4.1 SEÑALES ANALÓGICAS Y DIGITALES

La comunicación entre dos entidades se logra a través de señales, que pueden ser discretas o continuas. Las primeras toman un número limitado de valores y reciben el nombre de digitales mientras que las segundas tienen un comportamiento de onda y se conocen como analógicas. Como ejemplo se puede comparar un mensaje en código Morse contra su versión hablada.

Existen dispositivos electrónicos que permiten la conversión entre señales, éste proceso es necesario si se requiere enviar información desde o hacia el exterior de una computadora digital.

- ✓ Sensor (Transductor). Convierte variables físicas (calor, luz, fuerza) en señales eléctricas analógicas. Tal es el caso de un micrófono.
- ✓ Actuador. Hace la función inversa del transductor.
- ✓ Conversor analógico-digital (ADC, por sus siglas en inglés). Transforma voltaje analógico a digital.
- ✓ Conversor digital-analógico (DAC, por sus siglas en inglés). Realiza el proceso contrario que un ADC, tal como sucede al reproducir un disco compacto.
- ✓ Amplificador operacional (op-amp). Es un integrado que posee dos entradas y una salida. Ésta última se genera a partir de la diferencia de las entradas, multiplicada por un factor G (ganancia). Por esta característica recibe el nombre de amplificador, puede tener distintas configuraciones: inversor, comparador, integrador, etc. Se puede usar en el proceso de conversión entre señales, tiene el símbolo:

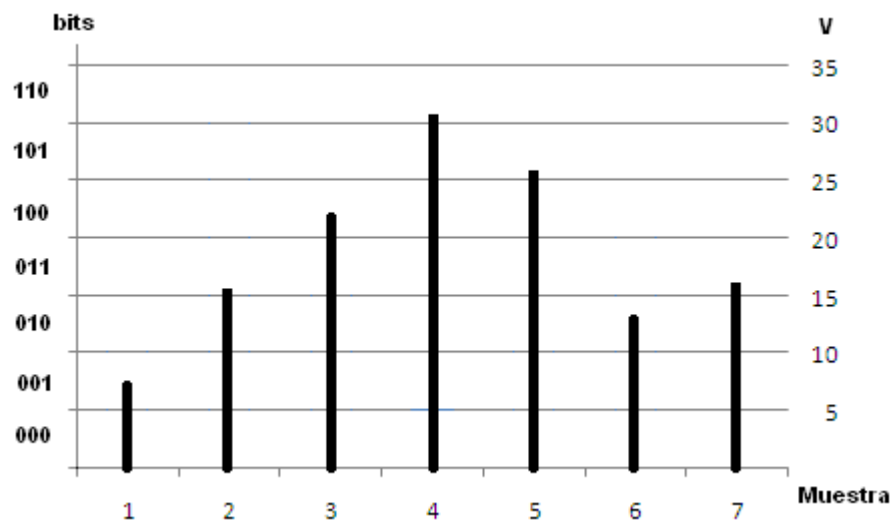


Al proceso que selecciona un número de valores discretos a partir de una señal continua por segundo.

El ingeniero sueco Niquist conjeturó en 1928 que la frecuencia de muestreo debe ser superior al doble de la máxima frecuencia a muestrear.

Ejemplo. Para digitalizar una canción en CD se utiliza una tasa de 44100 muestras por segundo (44.1 MHz), ya que la máxima frecuencia perceptible para el oído humano está cercana a 20KHz.

Después de realizar el muestreo se requiere de un procedimiento conocido como cuantificación, que se encarga de asignar a cada valor dentro de cierto rango, su equivalente en código binario, tomando en cuenta que el número de *bits* define los límites de los rangos.



ACTIVIDAD DE APRENDIZAJE

Investiga cuáles integrados se pueden emplear para DAC y ADC, así como sus hojas de datos.

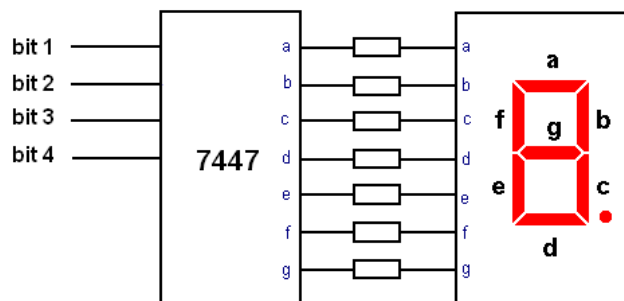
4.2 CODIFICADORES Y DECODIFICADORES

Un decodificador tiene n entradas de bits que generan 2^n combinaciones distintas, las cuales son convertidas en algún tipo de código que es utilizado por otro dispositivo.

Ejemplo. En un *display* con siete segmentos de LEDs cada uno debe ser activado para mostrar números del cero al nueve de acuerdo a la tabla:

Segmentos							
#	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

Para poder enviarle al *display* las señales necesarias, se requiere de un circuito integrado (7447), que realiza la decodificación de cuatro bits para el encendido de cada segmento, ya que $2^4 = 16$.



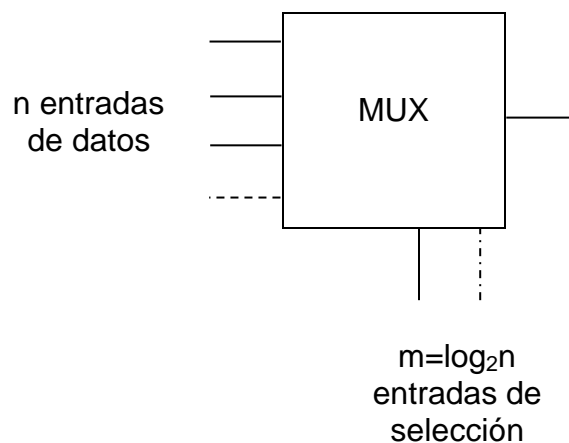
Los codificadores hacen la función inversa que los decodificadores.

ACTIVIDAD DE APRENDIZAJE

Busca la hoja de datos del circuito integrado 7447 e identifica para qué se usa cada patilla.

4.3 MULTIPLEXORES

Un multiplexor (MUX) posee n entradas de datos y m entradas de selección que, según la combinación, elige la información a ser enviada por la única salida disponible. Por esta razón se le denomina selector de datos.



Los demultiplexores tienen la funcionalidad opuesta al MUX.

ACTIVIDAD DE APRENDIZAJE

Investiga cómo se realiza un MUX de cuatro entradas a una salida, utilizando puertas lógicas.

4.4 BUSES

El subsistema de entrada/salida es el encargado de administrar las comunicaciones entre la computadora y los periféricos, para lograrlo cuenta con los siguientes elementos:

- ✓ Bloques de memoria asignados a las funciones de entrada/salida (E/S).
- ✓ *Buses* para la comunicación de datos.
- ✓ Módulos de control de E/S.
- ✓ Interfaces.
- ✓ Medios de comunicación como cables o señales de radio.

A las vías que permiten el enlace lógico se les conoce como puertos, mientras que las conexiones físicas mediante placas y/o cables reciben el nombre de *buses*, los cuales se pueden dividir en dos tipos:

- ✓ Internos. Transmiten direcciones, datos y señales de control.
- ✓ Externos. Se conectan a través de puertos e interfaces en serie y paralelo, una transmite un *bit* a la vez, y la otra un *byte*.

Módulos de E/S. Están encargados de transferir datos entre el dispositivo y la memoria principal o alguno de los registros, a través de una serie de reglas conocidas como interfaces. Las funciones que desarrollan son:

- ✓ Temporización y control.
- ✓ Comunicación con el procesador.
- ✓ Comunicación con los dispositivos.
- ✓ *Buffer* de datos.
- ✓ Detección de errores.

Las operaciones de E/S requieren de las siguientes fases:

- ✓ Direccionamiento. Se identifica al dispositivo con el que se tendrá comunicación.
- ✓ Sincronización (*Handshake*). Se decide cuándo se iniciará la transferencia.
- ✓ Transferencia.

A continuación se describen los métodos de control de E/S:

- ✓ Programado (*Polled*). La CPU le pregunta al dispositivo si necesita atención.
- ✓ Interrupciones. El dispositivo manda un mensaje pidiendo ser atendido.
- ✓ DMA (*direct memory access*). Se hace la transferencia de grandes volúmenes de datos entre los periféricos y la memoria principal, liberando al procesador de esta tarea.
- ✓ Canales E/S. Transfiere un *byte* a la vez.

ACTIVIDAD DE APRENDIZAJE

Investiga las características de los diferentes estándares de buses: Multibus I, Multibus II y VMEbus.

EJEMPLOS DE APLICACIÓN

Debido a que el mundo físico lo percibimos mediante señales analógicas, es necesario realizar procesos de conversión antes de poderlas introducir a la computadora. Dos sentidos que nos permiten interactuar con los demás son la vista y la audición, cuando los procesamos en el cerebro, nos permiten dar una respuesta mediante la voz o de alguna otra forma que constituya un lenguaje. A continuación se presentan algunos dispositivos electrónicos encargados de hacer éstas tareas a través de ordenadores.

MODEM

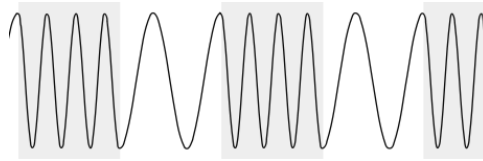
El modem (modulador-demodulador) fue desarrollado con fines militares durante la década de 1950 para la comunicación de datos. La existencia de una red telefónica para la transmisión de voz que podría ser aprovechada, requirió que se diseñara un dispositivo capaz de convertir bits en señales audibles, que pudieran viajar a través de las líneas existentes.



© Facultat d'Informàtica de Barcelona

AT&T lanzó al mercado un modem con una tasa de transmisión de 300 bits por segundo en el año 1962 usando FSK. En 1996, el doctor Townshed desarrolló uno que podía enviar 56Kb.

Al proceso de transportar datos a través de ondas variando su frecuencia (*pitch*), se le conoce como modulación. Una de estas técnicas se denomina *Frecuency-shift keying* (FSK), su versión más simple es la binaria o BFSK, la cual utiliza dos tonos: marca, que representa uno; y espacio, equivalente a cero.



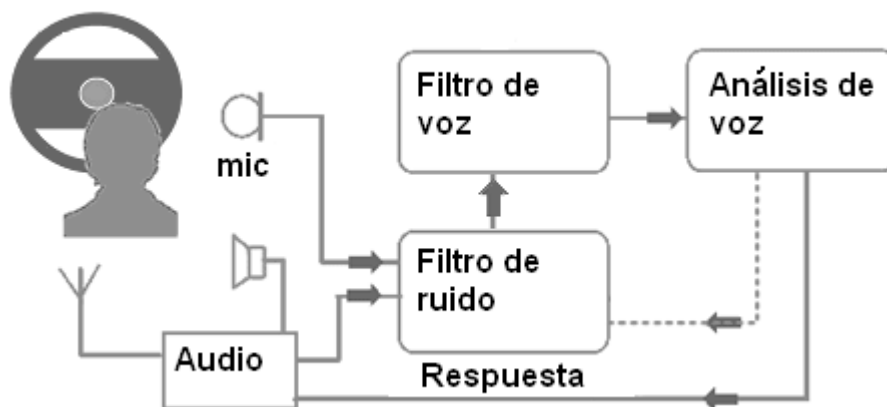
Una vez que la señal llega a su destino se utiliza el proceso inverso conocido como demodulación.

NAVEGACIÓN MEDIANTE VOZ

Los sistemas de navegación que permiten interactuar a través de la voz, necesitan reducir las interferencias provocadas por la gran cantidad de sonidos provenientes del medio ambiente.

La voz del conductor es recibida por un micrófono y filtrada para quitar el eco y eliminar ruidos, también se debe distinguir entre solicitudes del usuario y respuestas generadas por el sistema para no producir ciclos, como quien ve a su sombra de forma que fuera otra persona.

A continuación, las ondas son recibidas por el sistema de análisis de voz, capaz de reconocer fonemas y construir frases, que serán digitalizadas y procesadas para producir respuestas en forma de audio, que puedan ser usadas por el conductor.



CCD

El Premio Nobel de Física en 2009, fue entregado a tres científicos, dos de ellos inventaron el sensor CCD (*charge-decoupled device*).

En 1969, Williard Boyle y George Smith, ambos de los Laboratorios Bell, idearon un dispositivo construido con silicón del tamaño de un timbre postal, formado de millones de fotoceldas sensibles a la luz, dispuestas de forma rectangular. Cada una es revisada y su contenido se guarda linealmente, de tal forma que si se tuviera un arreglo de 10x10 sensores, se convertirían en una cadena de 100 elementos, en donde cada señal eléctrica es convertida a bits.

Cada fotocelda recibe el nombre de *pixel* cuando su contenido ya forma una imagen que se puede ver en la computadora. La capacidad del sensor se obtiene multiplicando sus dimensiones, por ejemplo: un CCD de 1280x1024 pixeles tiene una capacidad de 1.3 megapixeles.

La imagen en el CCD es recibida en blanco y negro, para poder formar colores se requiere del uso de filtros: verde, rojo y azul. En 1972, la empresa Fairchild construyó el primer sensor de 100x100 pixeles.

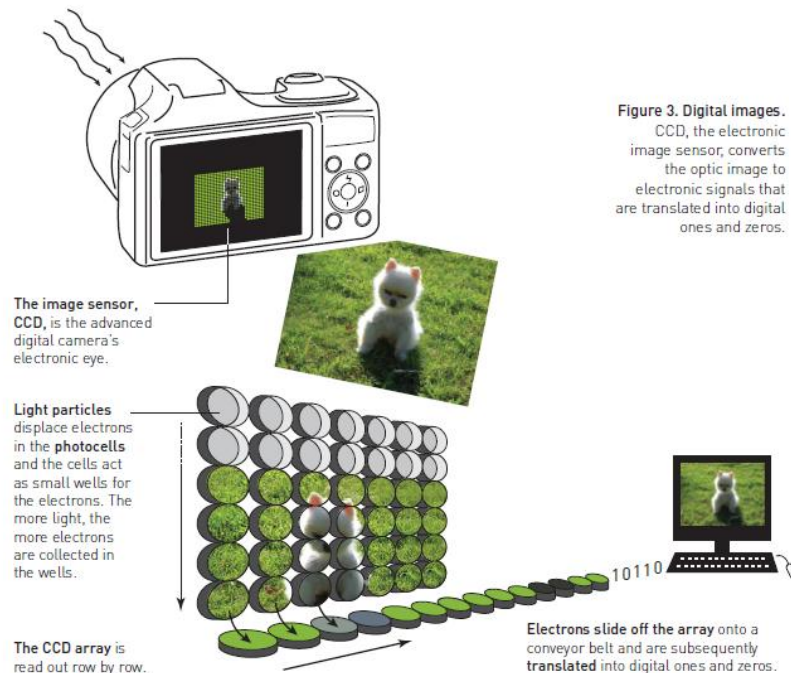


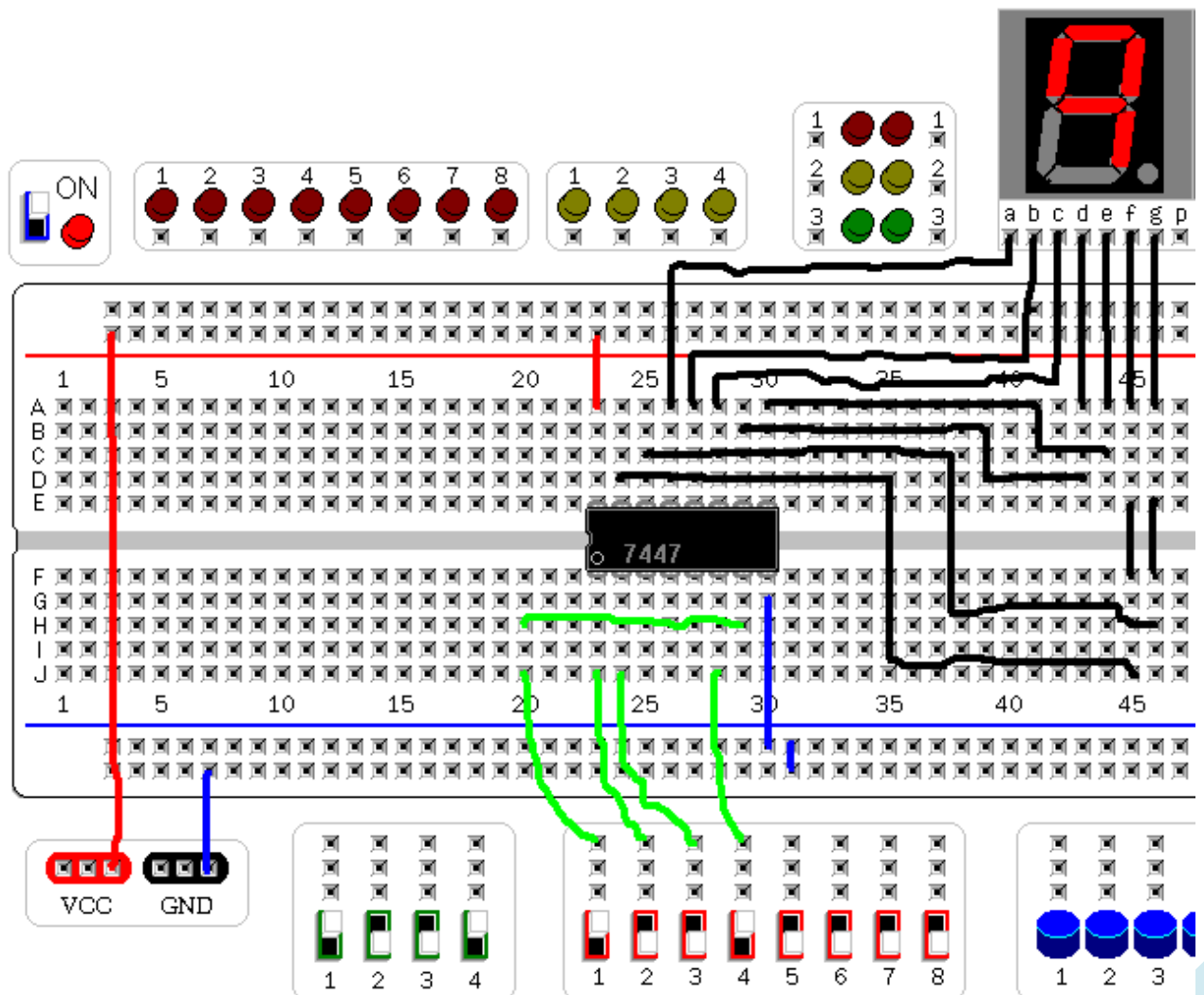
Figure 3. Digital images. CCD, the electronic image sensor, converts the optic image to electronic signals that are translated into digital ones and zeros.

PRÁCTICAS ASIGNADAS

Práctica 1

Objetivo

El alumno construirá un circuito basado en el integrado 7447 para hacer funcionar un *display* de siete segmentos de LEDs, usando cuatro *switches* para hacer la representación binaria de números del cero al nueve.



Sugerencia de montaje para el circuito.

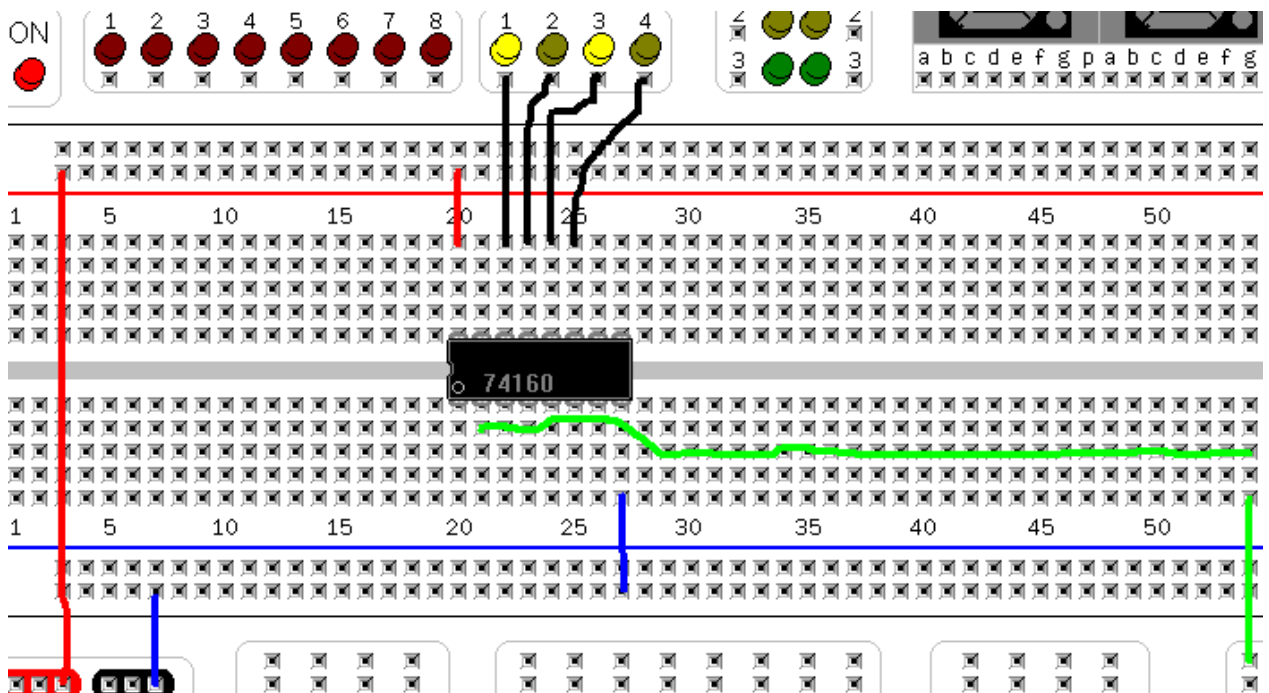
Práctica 2

Objetivo

El alumno construirá un circuito basado en el integrado 74160 para hacer funcionar un grupo de cuatro LEDs para hacer la representación binaria de números del cero al nueve.

Instrucciones

- ✓ Busca la hoja de datos del circuito integrado 74160 y coteja para qué se usa cada patilla.
- ✓ Utiliza una señal de reloj del simulador para sincronizar.

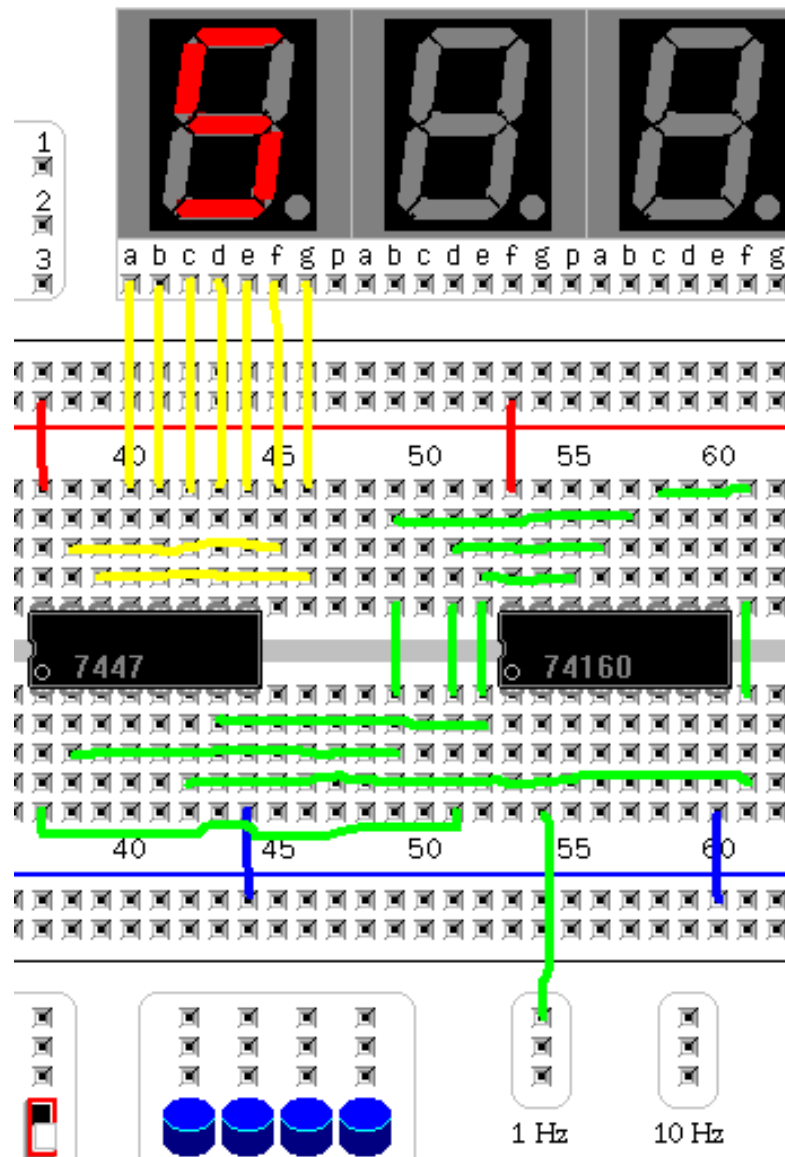


Sugerencia de montaje para el circuito.

Práctica 3

Objetivo

El alumno construirá un circuito basado en los integrados 74160 y 7447, para hacer funcionar un *display* de cuatro segmentos de LEDs de forma automática.



Sugerencia de montaje para el circuito.

AUTOEVALUACIÓN

Elige el inciso correcto.

A. Convierte señales físicas en variables eléctricas.

- a) Transductor b) Actuador c) ADC d) DAC

B. El código Morse es un ejemplo de señal.

- a) Continua b) Discreta c) Analógica d) Abierta

C. Puede tener distintas configuraciones: inversor, comparador, integrador, etc.

- a) Op-amp b) Transductor c) DAC d) ADC

D. El integrado 7447 hace la función de:

- a) DAC b) Decodificar c) ADC d) Multiplexar

E. Método que libera al procesador de transferir datos a la memoria.

- a) ADC b) DAC c) Buffer d) DMA

Completa las frases.

F. Un periférico es _____.

G. Las señales continuas tienen un comportamiento de _____.

H. Los _____ transmiten direcciones, datos y señales de control.

I. La tasa o frecuencia de muestreo es _____.

J. Niquist conjeturó que _____

_____.

RESPUESTAS DE LA AUTOEVALUACIÓN

A. a.

B. b.

C. a.

D. b.

E. d.

F. Un dispositivo externo conectado a la computadora.

G. Onda.

H. Buses internos.

I. La cantidad de mediciones que se hacen por unidad de tiempo.

J. La frecuencia de muestreo debe ser superior al doble de la máxima frecuencia a muestrear.

UNIDAD 5

DISPOSITIVOS LÓGICOS PROGRAMABLES

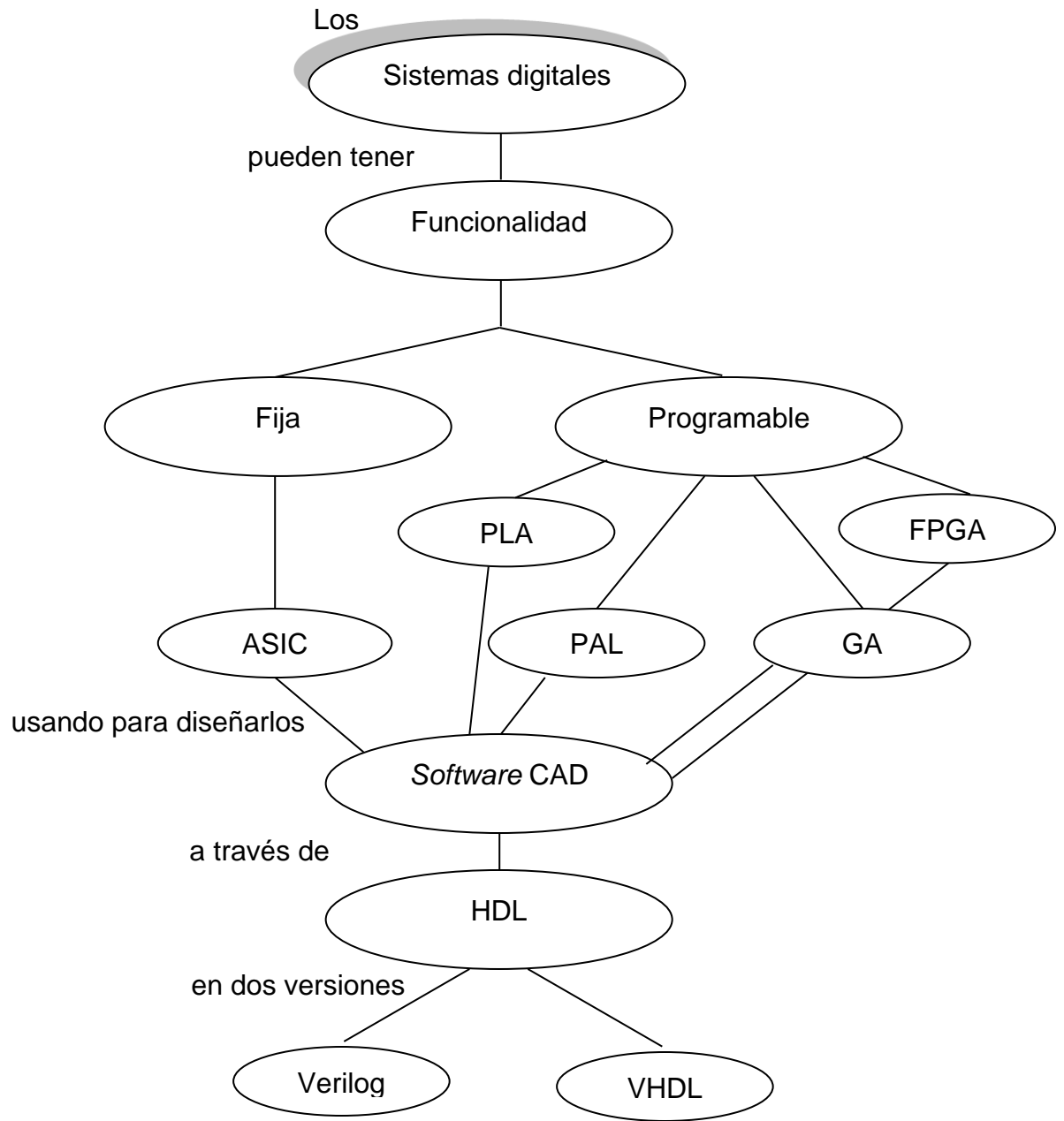
OBJETIVO:

El alumno comparará las diferentes tecnologías involucradas en la fabricación y funcionamiento de dispositivos lógicos. También conocerá el proceso de diseño de los mismos, basándose en una descripción escrita de sus características.

TEMARIO

- 5.1 CLASIFICACIÓN
- 5.2 ARQUITECTURA
- 5.3 VHDL

MAPA CONCEPTUAL



INTRODUCCIÓN

Los sistemas electrónicos actuales están compuestos por microprocesadores, memorias y dispositivos lógicos, los cuales son de varios tipos: interfaces, temporización, control y despliegue de datos. Las estructuras lógicas que establecen las funciones de cada uno pueden ser fijas o adaptarse a los requerimientos cambiantes del usuario, mediante el uso de dispositivos lógicos programables (PLDs, por sus siglas en inglés). La producción de dispositivos genéricos más flexibles ha permitido la reducción de costos, de tal forma que podemos encontrar estos sistemas en autos, electrodomésticos, teléfonos celulares, etcétera.

5.1 CLASIFICACIÓN

Las compuertas se pueden interconectar de dos formas: fija, cuando su función se conoce desde la fabricación, no siendo posible hacerle cambios; y programable, permitiendo al usuario adaptarlas a sus necesidades.

Los circuitos integrados para aplicaciones específicas (ASIC, por sus siglas en inglés), están hechos a medida en las etapas finales de su elaboración, cumpliendo con el diseño proporcionado por el usuario. Se pueden encontrar en teléfonos celulares, consolas de video juego, pantallas, etcétera.

Los dispositivos lógicos programables (PLDs, por sus siglas en inglés) constan de compuertas interconectadas formando arreglos cuya función se puede establecer una o más veces. Puede tener las mismas aplicaciones que los ASICs pero a un costo menor ya que son de tipo genérico al salir de la fábrica, correspondiendo al usuario la personalización de sus funciones.

Algunos tipos de PLDs son:

- ✓ Memorias de solo lectura (ROM, por sus siglas en inglés).
- ✓ *Programmable logic array* (PLA).
- ✓ *Programmable array logic* (PAL).
- ✓ *Gate array* (GA).

PAL y GA se conocen como PLDs simples (SPLD, por sus siglas en inglés), las cuales al interconectarse forman PLDs complejas (CPLD, por sus siglas en inglés). Éstas últimas aparecen debido al aumento en la complejidad de las aplicaciones de nuevos dispositivos.

ROM, PLA y PAL están formadas por dos niveles: un arreglo de compuertas AND y otro de OR. Es importante recordar que cualquier función de lógica combinacional puede ser expresada en forma de suma (OR) de productos (AND), conocida como forma SOP, por sus siglas en inglés. Ésta puede a su vez, ser escrita como un producto de sumas (POS, por sus siglas en inglés) usando los teoremas de De Morgan.

PROM (*programmable* ROM). La ROM programable fue introducida en 1970. Con n bits de entrada, requiere de 2^n puertas AND fijas y un arreglo de ORs programables compuestos por fusibles, los cuales permiten que sólo se configure una vez. El uso de transistores hizo posible la reconfiguración de la memoria, usando rayos ultravioleta (EPROM) o eléctricamente (EEPROM).

Existen dos versiones de PROM:

- ✓ *Mask-programmable*. Su funcionalidad la fija el fabricante.
- ✓ *Field-programmable*. Puede ser cambiada por el usuario.

PLA. Los arreglos AND y OR son programables, aparecieron en 1975 y usan la forma SOP. Su rendimiento no es óptimo.

PAL. El arreglo AND es programable y el OR fijo, con la finalidad de aumentar su rendimiento. También usa la forma SOP a través de los ANDs constituidos por flip-flops, siendo posible retroalimentarlos de las salidas que proporcionan los ORs. A éstas se pueden conectar circuitos adicionales conocidos como Macroceldas.

GA. Es similar a PAL, aunque a diferencia de éste que usa fusibles, puede ser borrada eléctricamente. Cuenta con canales tanto de entrada/salida como de comunicación interna, las compuertas se agrupan en celdas funcionales tales como multiplexores, contadores, sumadores, etc. Su proceso de fabricación es lento, sin embargo, se apoya en el uso de *software* CAD (*computer aided design*), el cual permite la sustitución del papel por un lienzo virtual para hacer diseños permitiendo flexibilidad y trabajo colaborativo a nivel global, ya que las especificaciones de un circuito pueden ser elaboradas por varios equipos en un continente y ser fabricado en otro.

Para solucionar las desventajas presentadas por los GA en cuanto a su estructura interna rígida, aparecieron los FPGAs (*field programmable gate array*) que en lugar de tener arreglos de ANDs y ORs, poseen bloques cuya funcionalidad lógica es adaptable. Pueden implementar circuitos

combinacionales y secuenciales, así como funciones lógicas multinivel, en lugar de solamente dos.

Existen dos tecnologías en las que se basan los FPGAs:

- ✓ SRAM. Fabricadas por XILINX, ALTERA y AT&T.
- ✓ *Antifuse*. Suministradas por ACTEL, QUICKLOGIC, CYPRESS y XILINX.

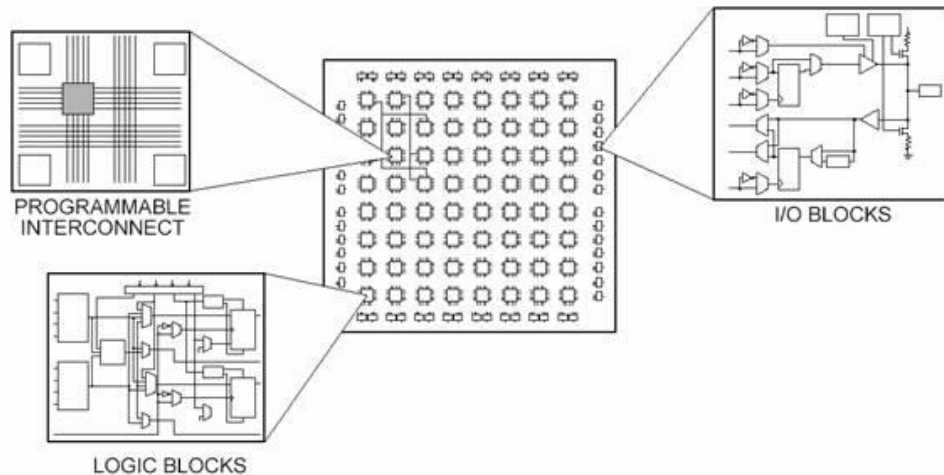
ACTIVIDAD DE APRENDIZAJE

Busca al menos tres ejemplos de *software* CAD para diseño de sistemas digitales.

5.2 ARQUITECTURA

A la distribución interna de los componentes de un dispositivo electrónico así como a los canales de interconexión entre estos y hacia el exterior; se le denomina arquitectura. En el caso de los PLDs su arquitectura es rígida, los FPGAs no tienen esta limitación.

Los FPGAs están contruidos con bloques lógicos configurables (CLB, por sus siglas en inglés). Cada CLB puede ser configurado para tener funciones combinacionales o secuenciales, asimismo, cuentan con bloques de entrada/salida (IOBs, por sus siglas en inglés) para comunicarse con dispositivos externos. Los CLBs pueden conectarse a otros CLBs y/o con IOBs mediante canales de rutas programables. También tienen tablas de búsqueda (LUT, por sus siglas en inglés), multiplexores configurables y registros. Cada LUT es un arreglo de n *bits* que sirven como entrada de un CLB y puede generar 2^n salidas de un *bit*.

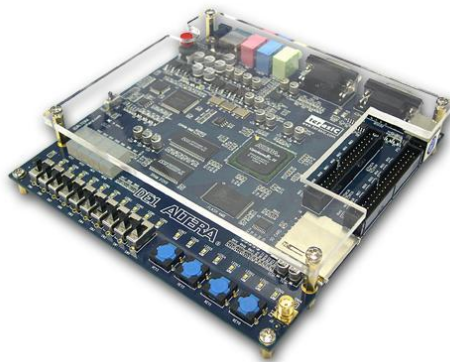


Arquitectura de un FPGA

© zone.ni.com

ACTIVIDAD DE APRENDIZAJE

Investiga las características del FPGA Cyclone II de ©ALTERA, así como de la tarjeta para pruebas DE1 Development and Educational Board del mismo fabricante.



© ALTERA

5.3 VHDL

Una descripción nos proporciona las características de un objeto mediante lenguaje hablado o simbólico.

Los HDLs (*hardware description language*) son lenguajes que permiten expresar diseños de circuitos lógicos programables a través de texto. Existen

dos versiones principales: la de Verilog; y VHDL, por sus siglas en inglés. Ambas tienen funcionalidad equivalente, pero diferente escritura.

VHDL fue desarrollado en 1981 por el Departamento de Defensa de Estados Unidos en conjunto con IBM, TEXAS INSTRUMENTS e INTERMETRICS. Está basado en el lenguaje de programación Ada.

Existen tres formas descriptivas para los objetos:

- ✓ Por su comportamiento (*behavioral*), estableciendo qué hace.
- ✓ Flujo de datos. A través de instrucciones de tipo booleano.
- ✓ Estructural. Se especifican las funciones lógicas y la forma de conectarse entre sí.

Se conocen como “palabras reservadas” a las instrucciones que se usan dentro de un lenguaje y se escribirán en negritas para diferenciarlas. La descripción se organiza a través de tres bloques de instrucciones:

- ✓ Librerías (**library**). Reutiliza especificaciones que ya se encuentran elaboradas.
- ✓ Entidad (**entity**). Describe funciones lógicas, así como entradas y salidas a través de puertos (**port**).
- ✓ Arquitectura (**architecture**). Establece la operación interna de la función lógica, la cual se puede predefinir para reutilizarla como un componente (**component**) con sus respectivas conexiones (**port map**).

ACTIVIDAD DE APRENDIZAJE

Investiga cuáles son todas las instrucciones que acepta VHDL.

PRÁCTICAS GUIADAS

Práctica 1

Objetivo

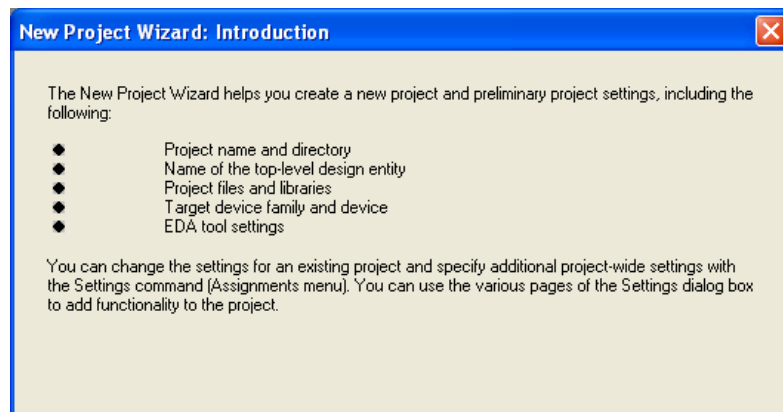
Que el alumno conozca el manejo de *software* CAD para el diseño de circuitos, e identifique cuales son las ventajas del mismo. A través de la construcción y simulación de un circuito que consta de una compuerta AND.

Instrucciones:

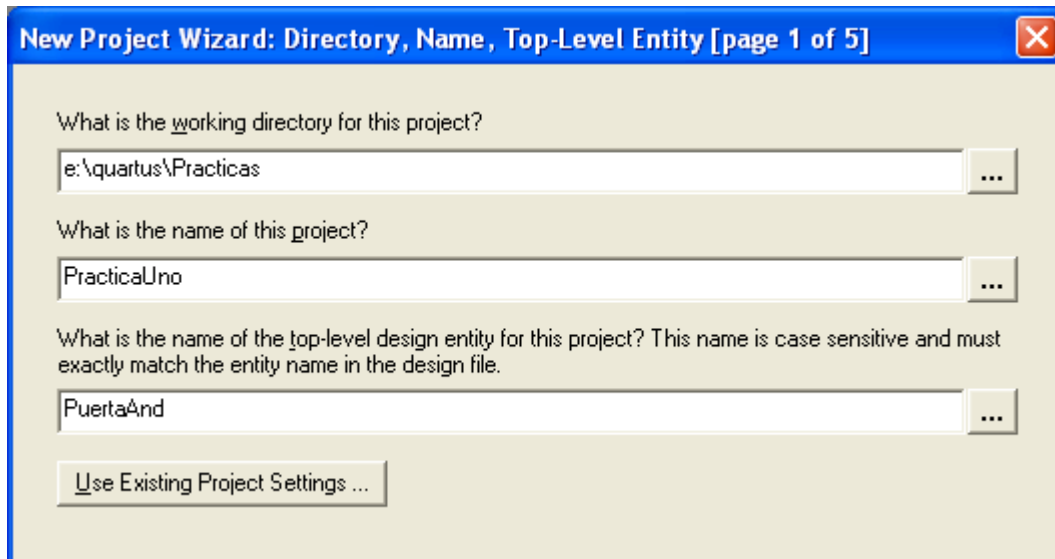
Paso 1. Descargar el *software libre* Quartus II 9.1 Web Edition de Altera[®] e instalar.

Paso 2. Ejecutar el programa: Inicio > programas > Altera > Quartus II 9.1 sp 2 Web Edition > Quartus II 9.1 sp 2 Web Edition ©.

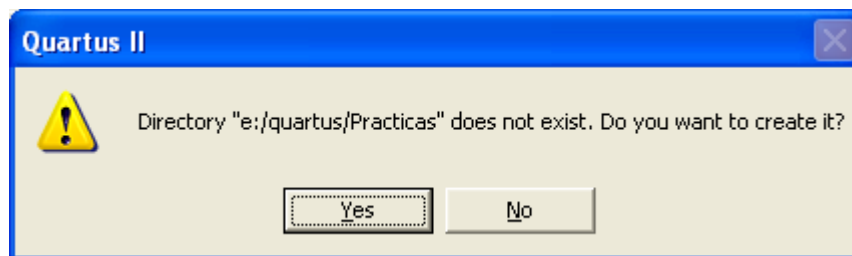
Paso 3. Crear un nuevo proyecto: File > New Project Wizard >



Next >

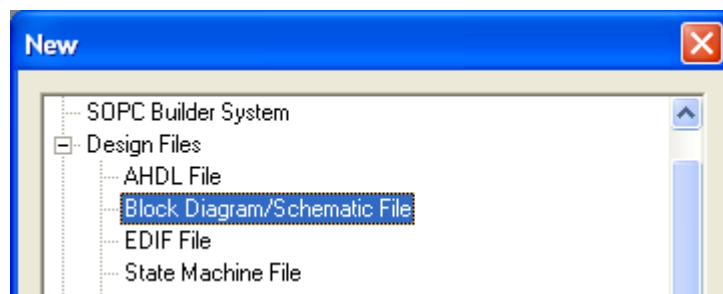


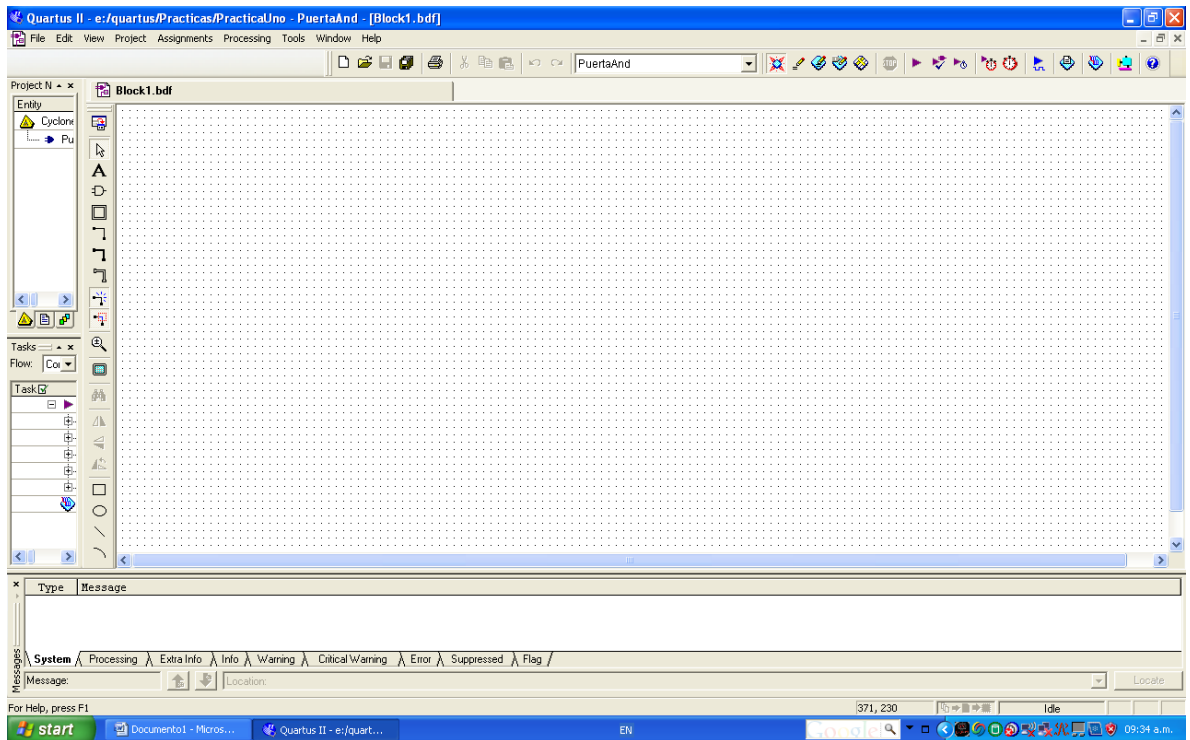
Especificar la carpeta de trabajo y los nombres de proyecto y entidad (eliminar acentos para evitar posibles problemas de compatibilidad) > Next >



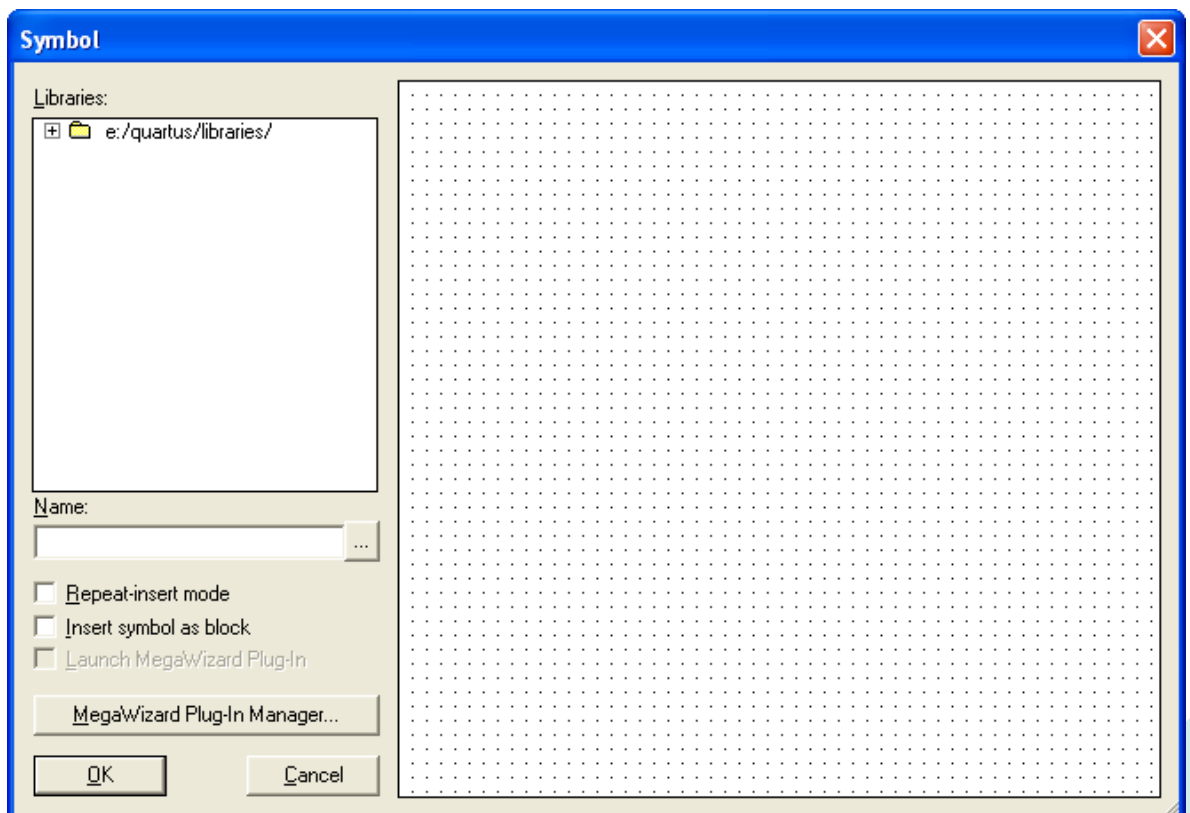
Yes > Finish.

Paso 4. Editar gráficamente: File > New > dentro de Design Files elegir Block Diagram/ Schematic File > OK.

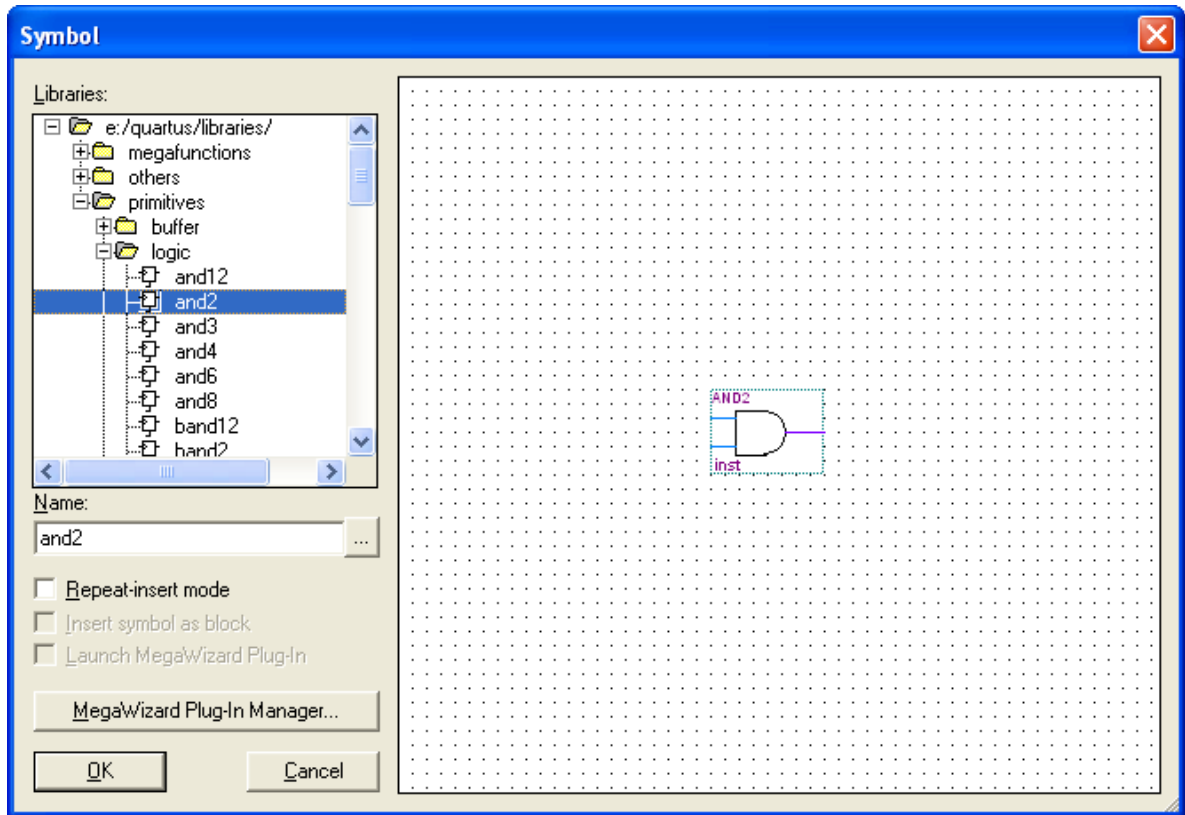




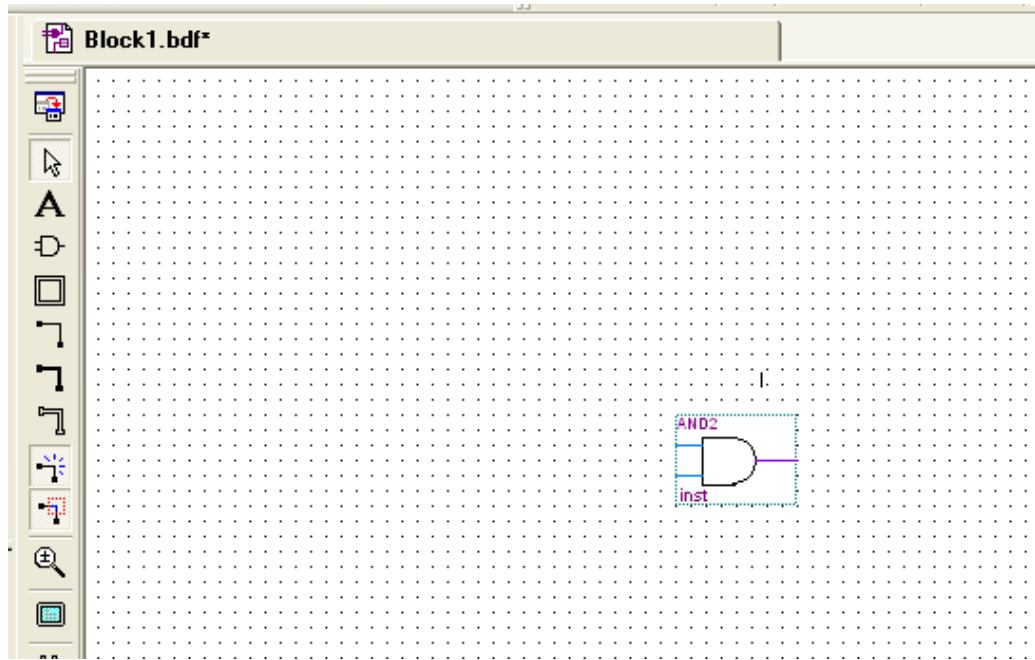
Paso 5. Agregar símbolos: Edit > Insert Symbol >



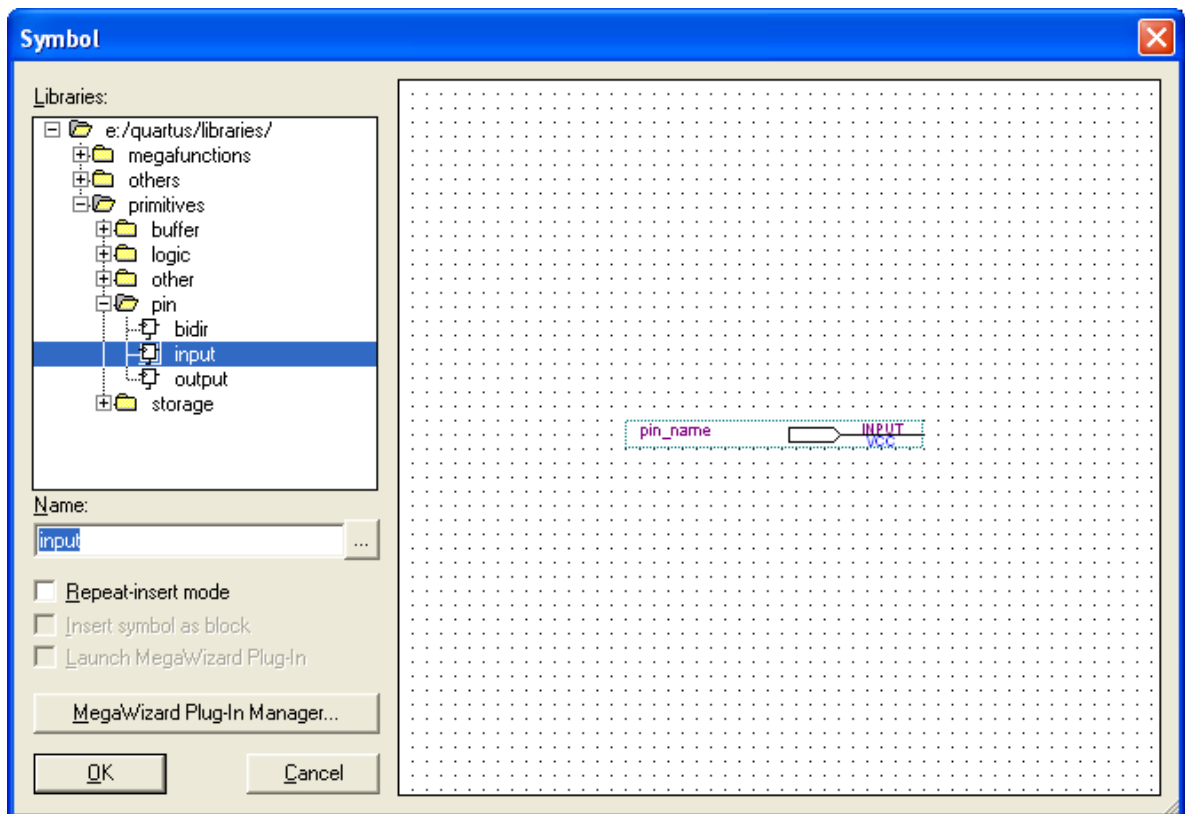
Abrir las opciones de libraries, dentro de primitives, abrir la carpeta logic y elegir and2 > OK.

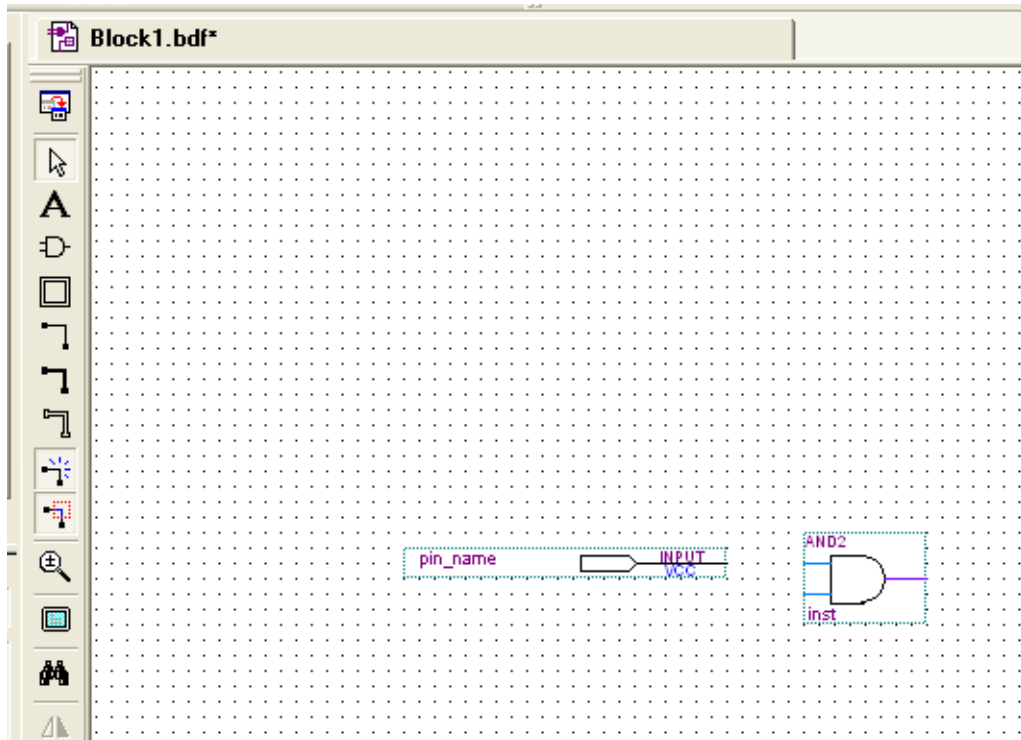


Aparece un símbolo AND flotante, dar *click* izquierdo en donde se quiera dejar.

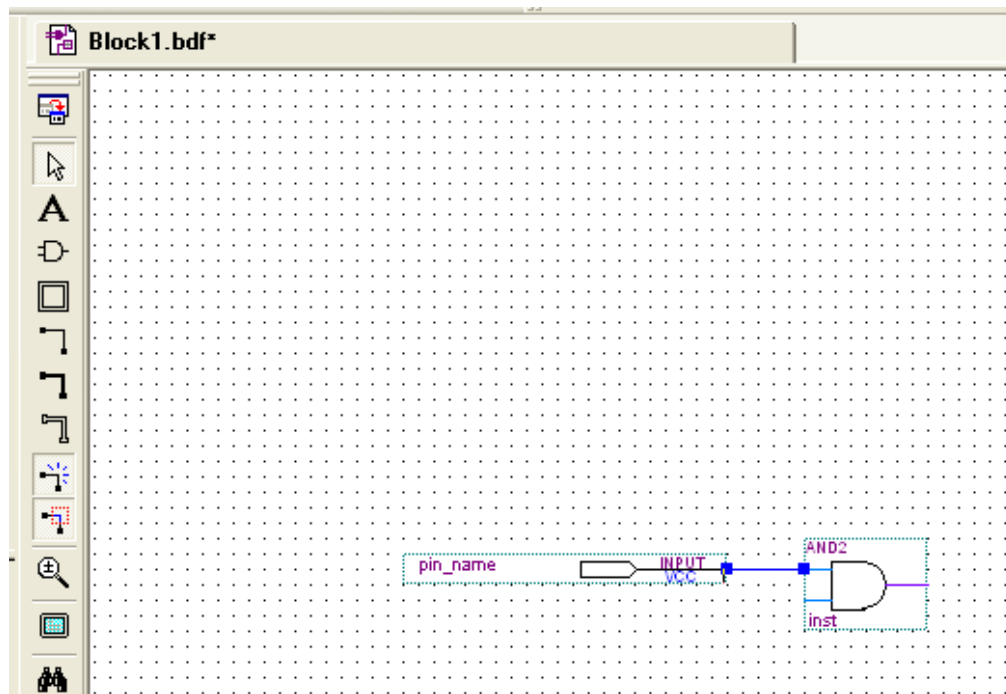


Repetir el proceso anterior eligiendo input en la carpeta pin > OK.

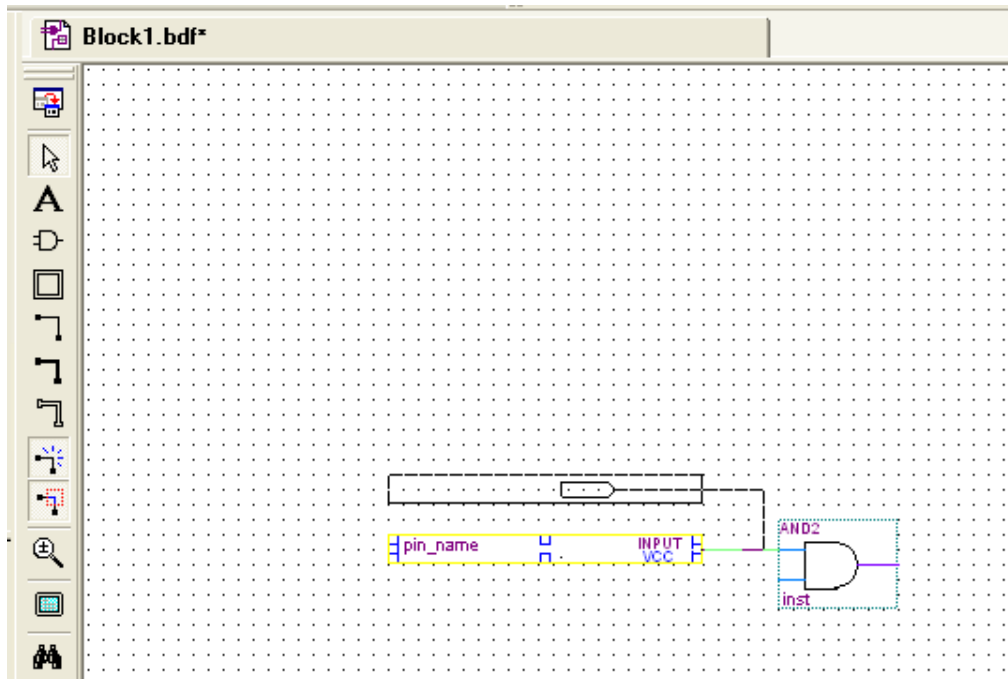




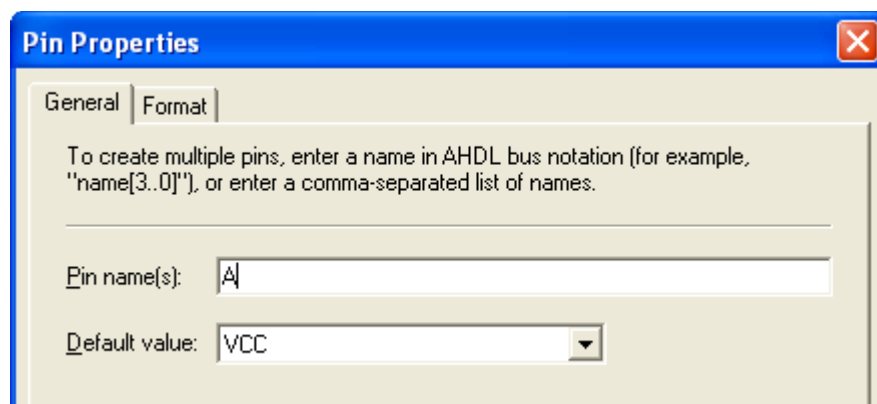
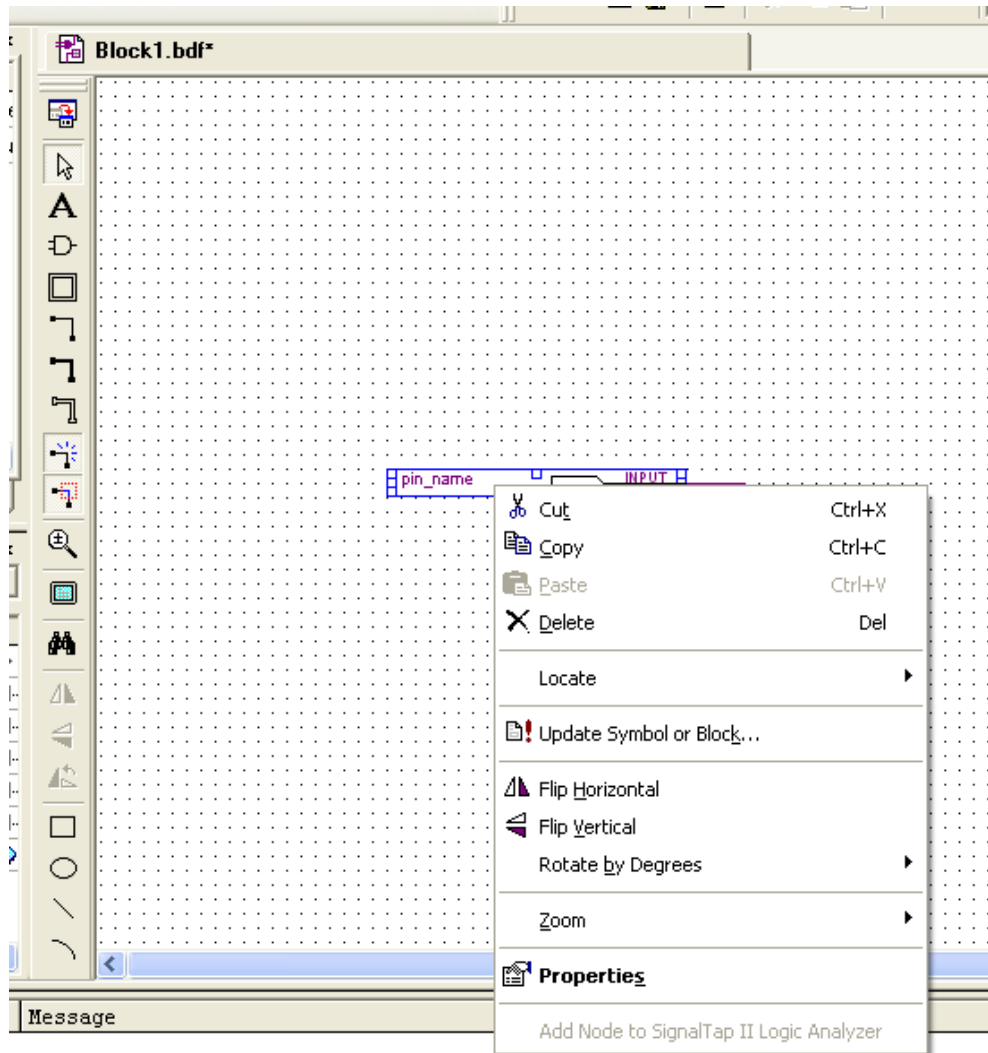
Paso 6. Alargar la línea para establecer comunicación.



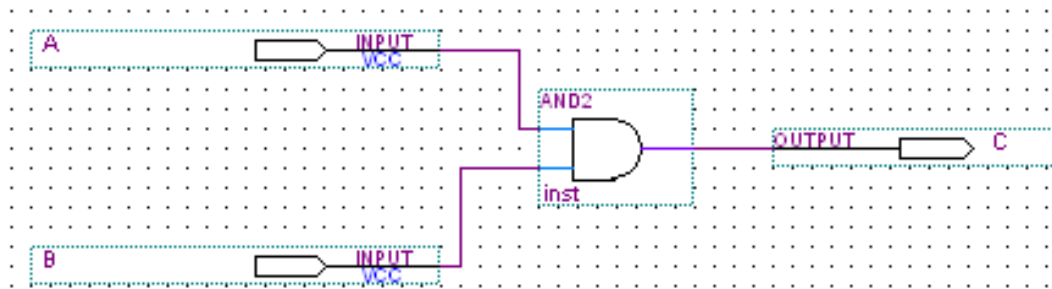
Paso 7. Desplazamiento: Poner el puntero del mouse sobre el pin > dar *click* sin soltar y mover hacia arriba > soltar.



Paso 8. Cambiar propiedades: Presionar el botón derecho del mouse > Properties > OK.



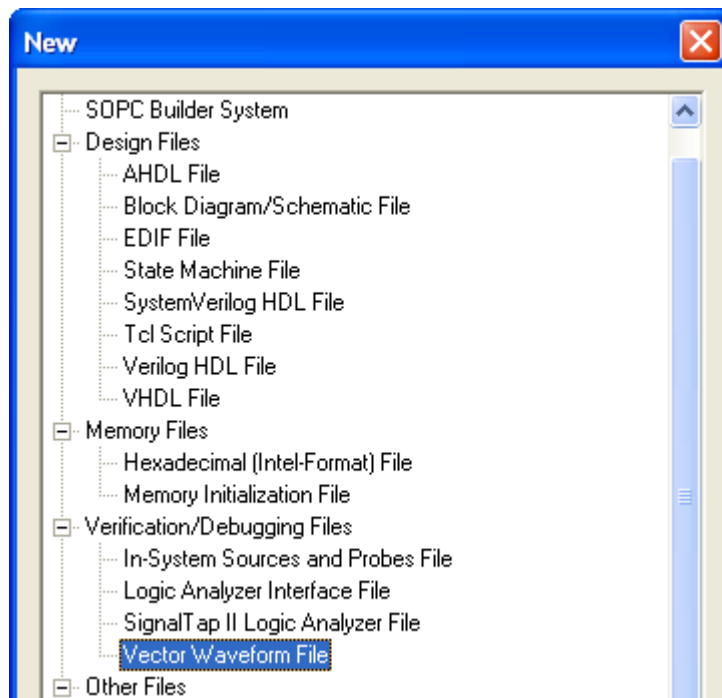
Repetir los pasos necesarios hasta lograr la siguiente distribución:

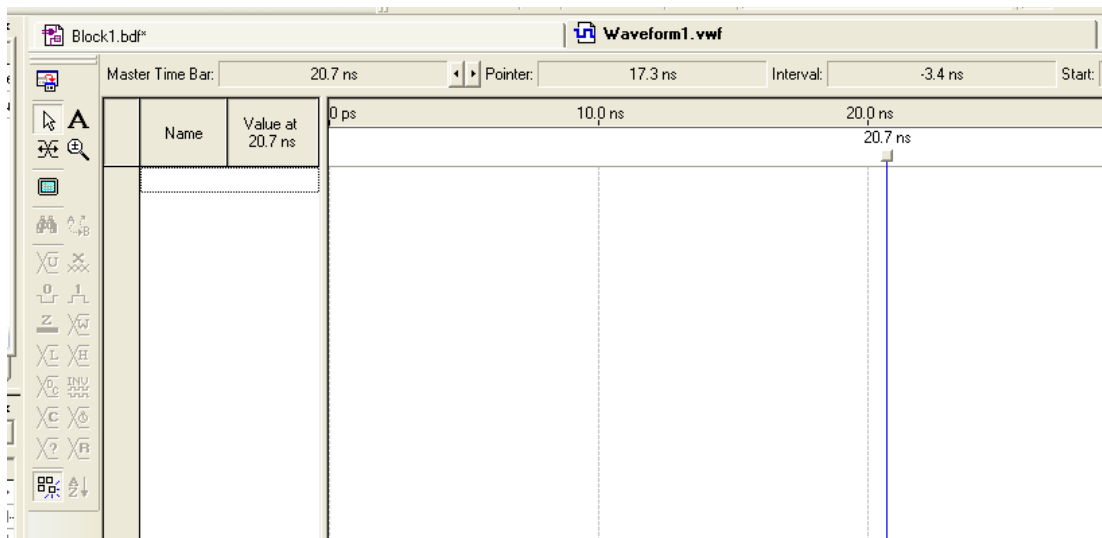
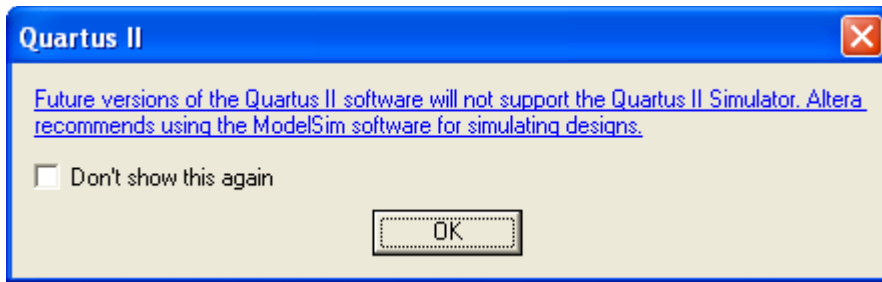


Paso 9. Guardar: File > Save > Save.

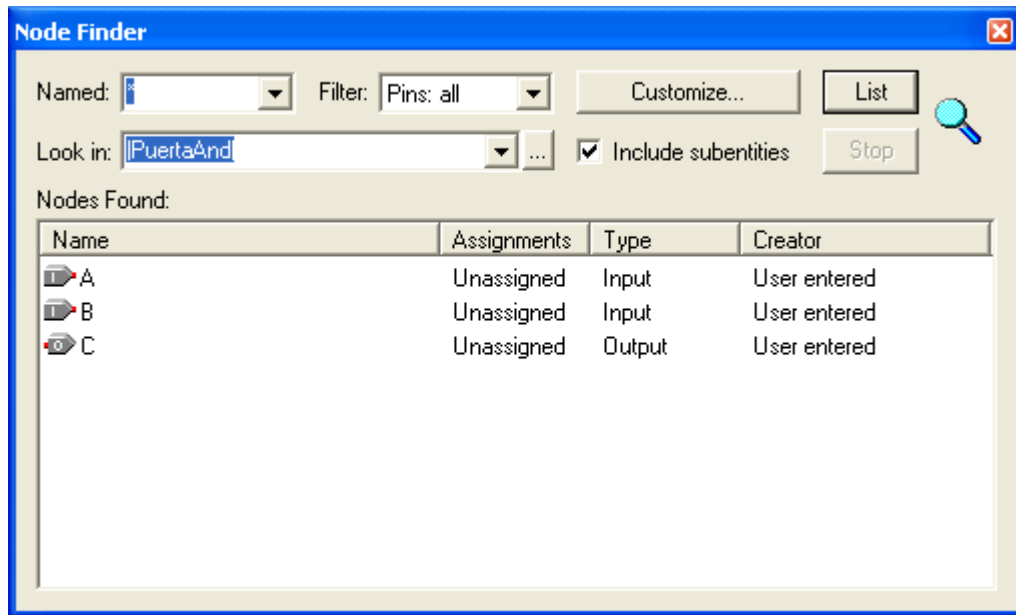
Paso 10. Compilar: Processing > Start Compilation (al terminar la compilación puede marcar *warnings*, pero no debe reportar errores).

Paso 11. Preparar simulación: File > New > > Vector Waveform File > OK > OK.

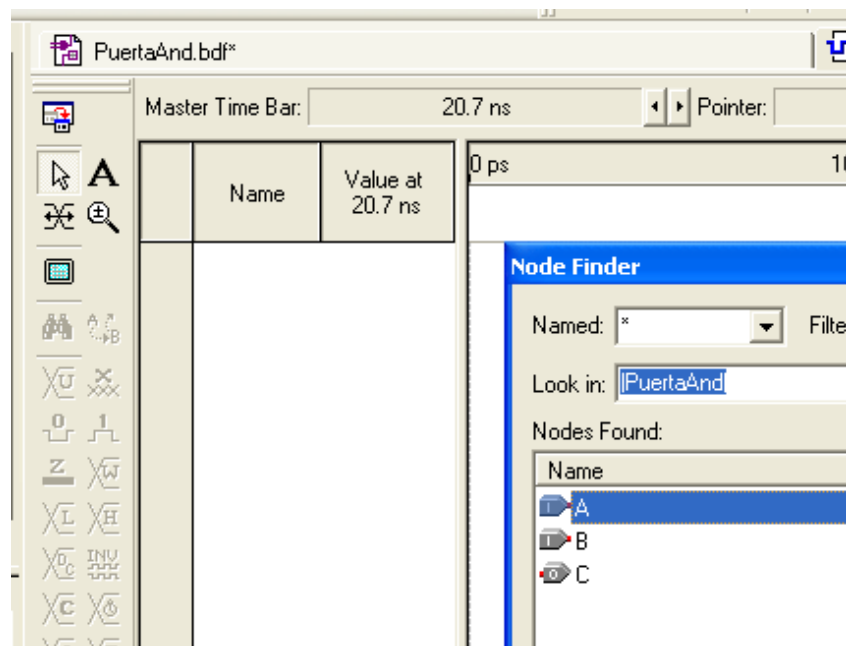


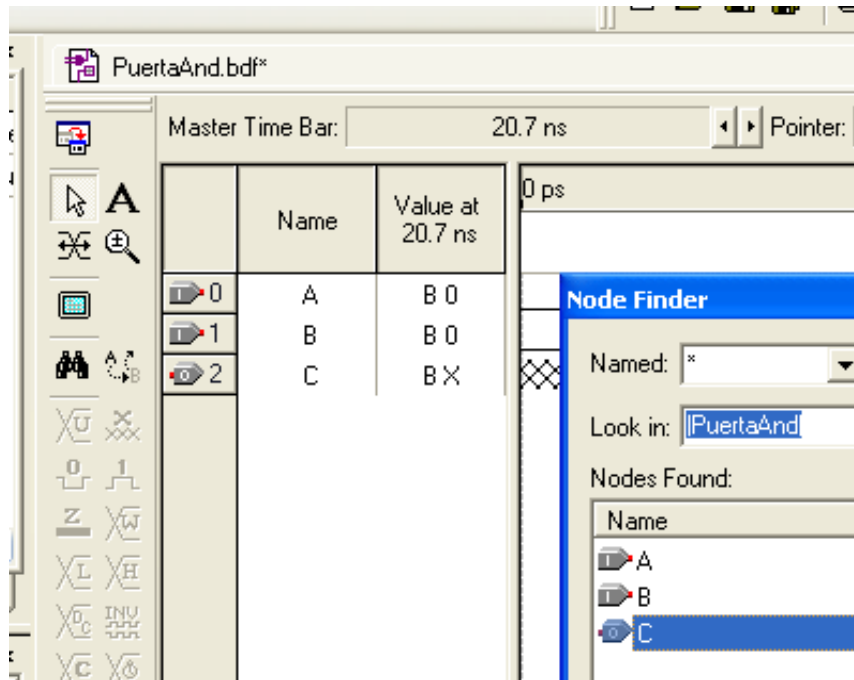


Paso 12. Buscar variables: View > Utility Windows > Node Finder > dentro de la opción Filter, elegir Pins: all > List.

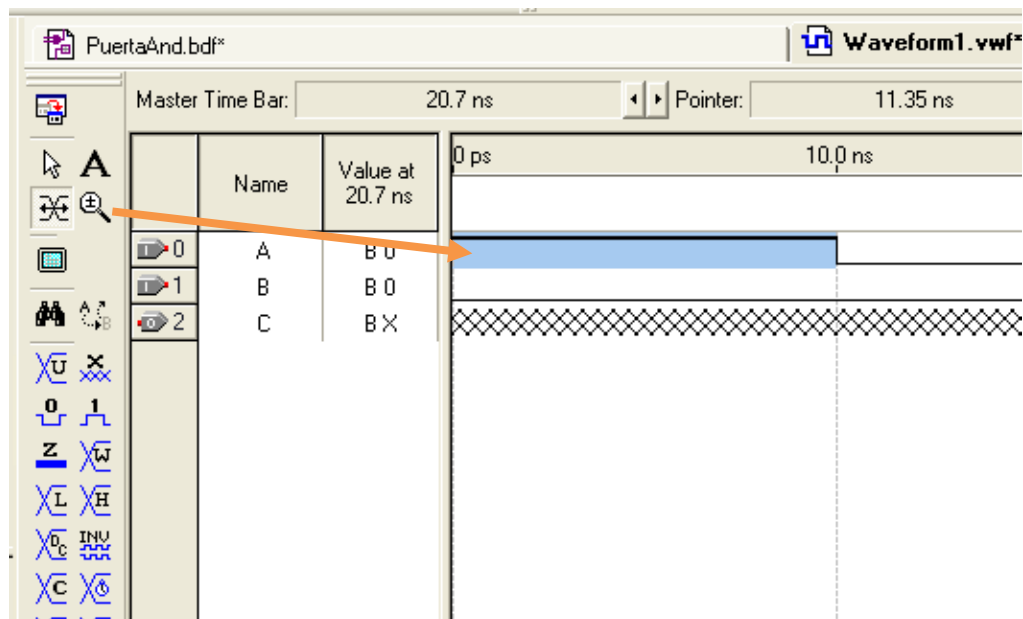


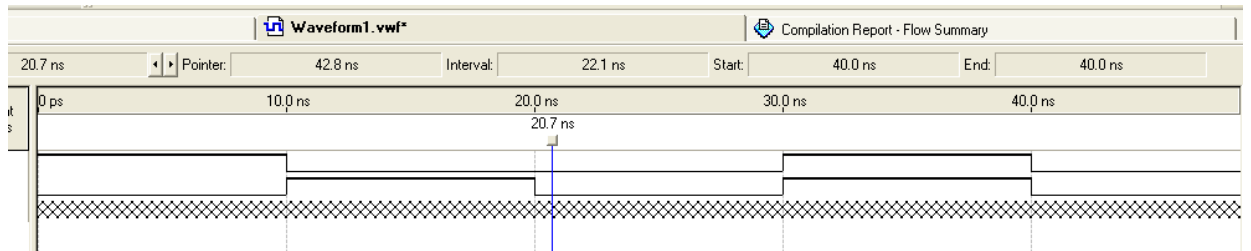
Paso 13. Agregar variables: Arrastrar y soltar cada variable, cerrar la ventana Node Finder.



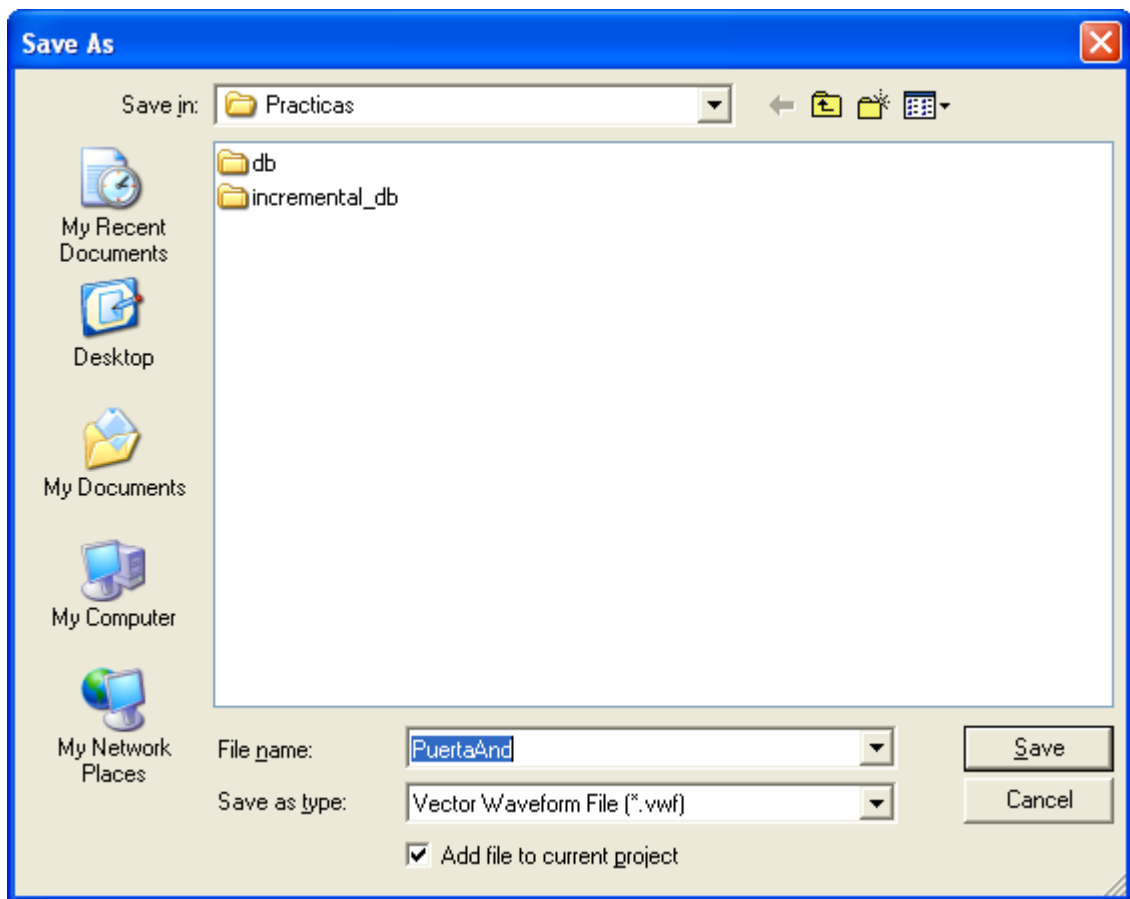


Paso 14. Establecer valores de simulación: En la barra de herramientas de lado izquierdo, elegir Waveform Editing Tool y establecer el valor alto (1) o bajo (0) dentro de cada lapso de tiempo, en las variables de entrada.

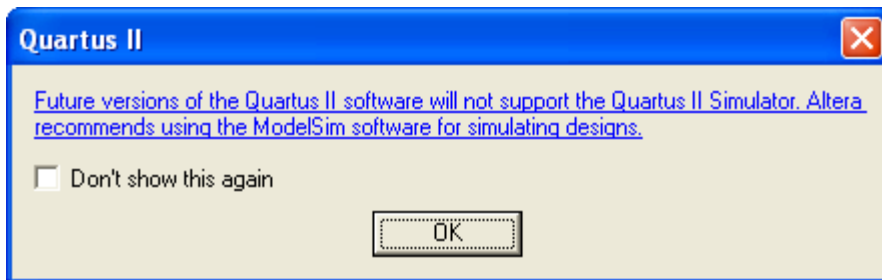
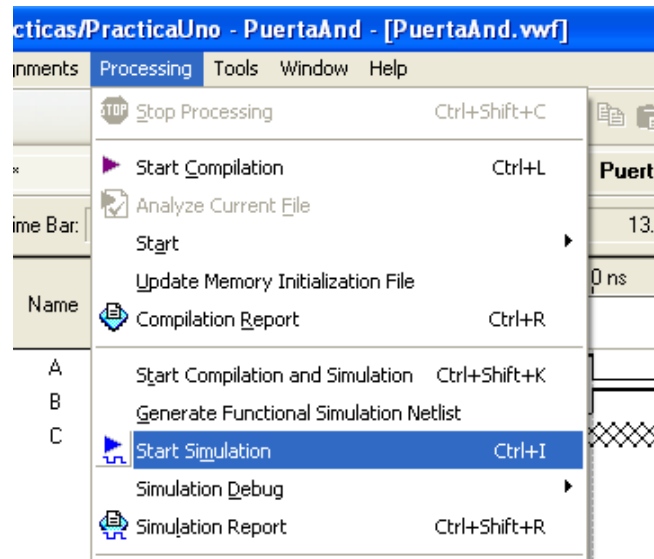


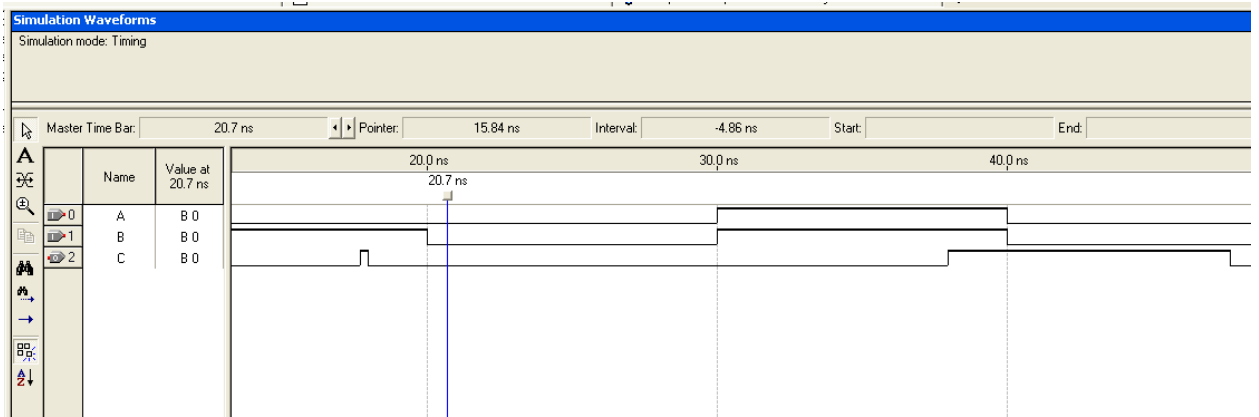


Paso 15. Guardar el archivo Vector Waveform.



Paso 16. Iniciar simulación: Processing > Start Simulation OK.





Paso 17. Cerrar proyecto: File > Close Project.

Práctica 2

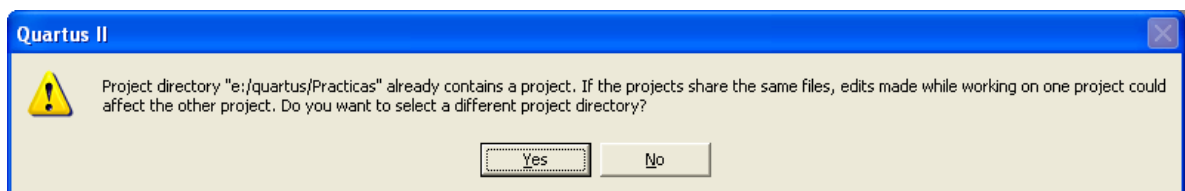
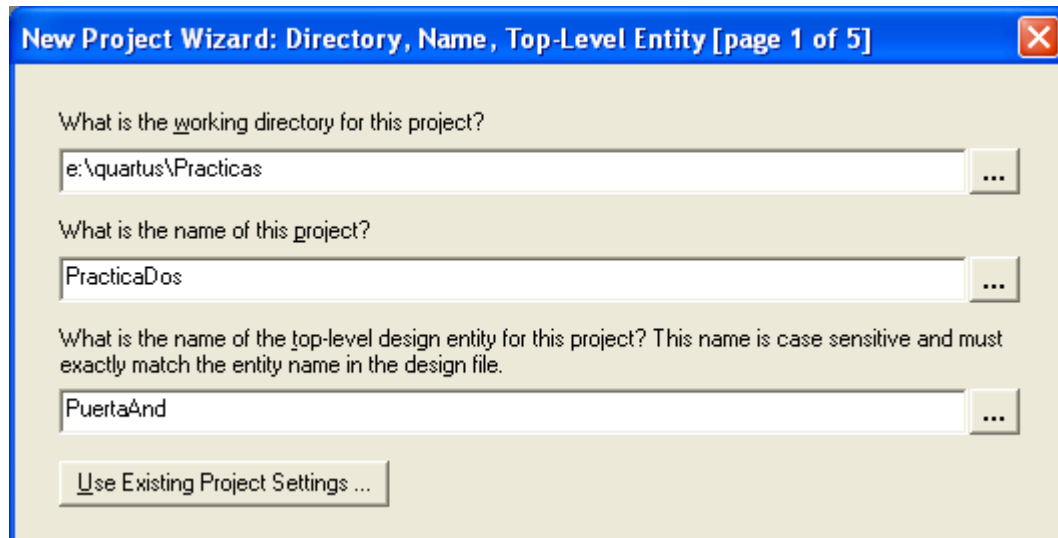
Objetivo

Que el alumno especifique un circuito usando VHDL y realice su simulación, mediante el uso de una compuerta AND para la elaboración del circuito.

Instrucciones:

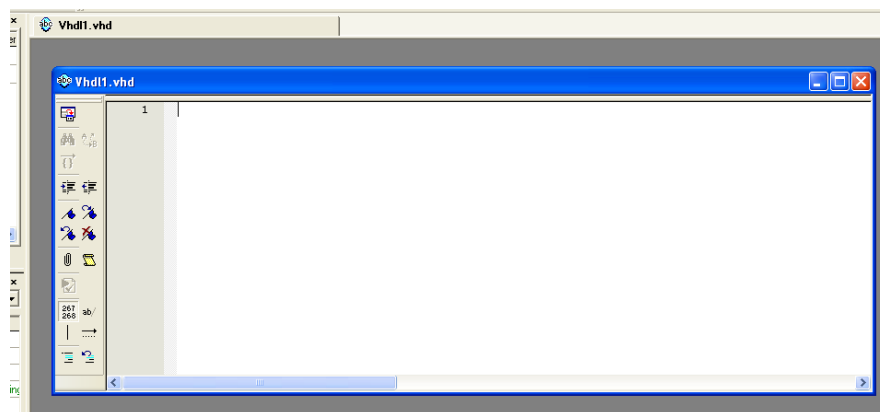
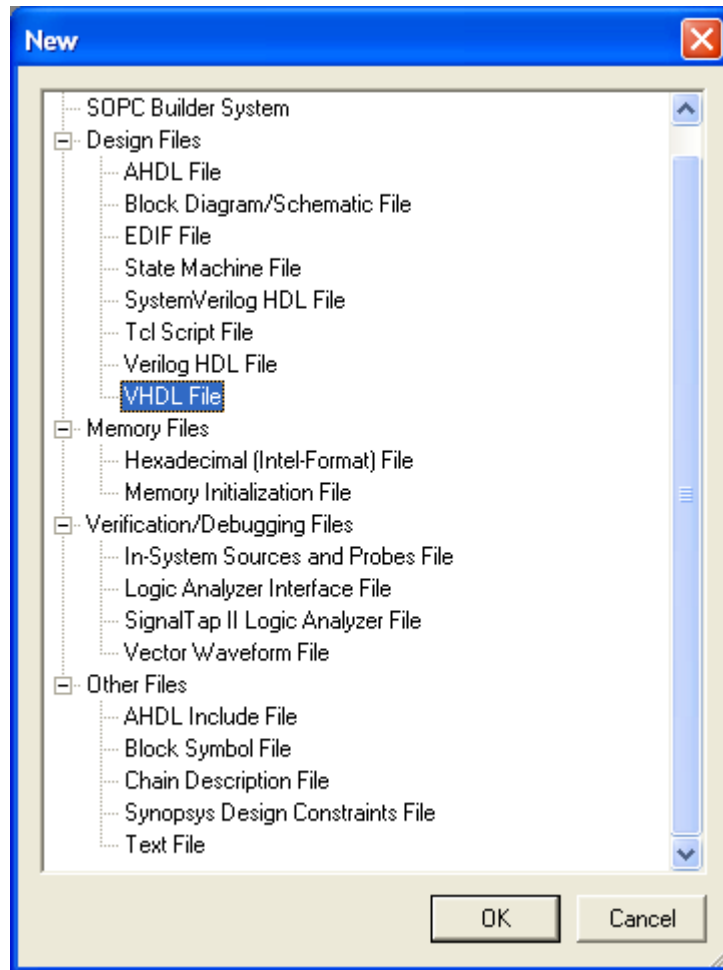
Paso 1. Ejecutar el programa.

Paso 2. Crear un nuevo proyecto.

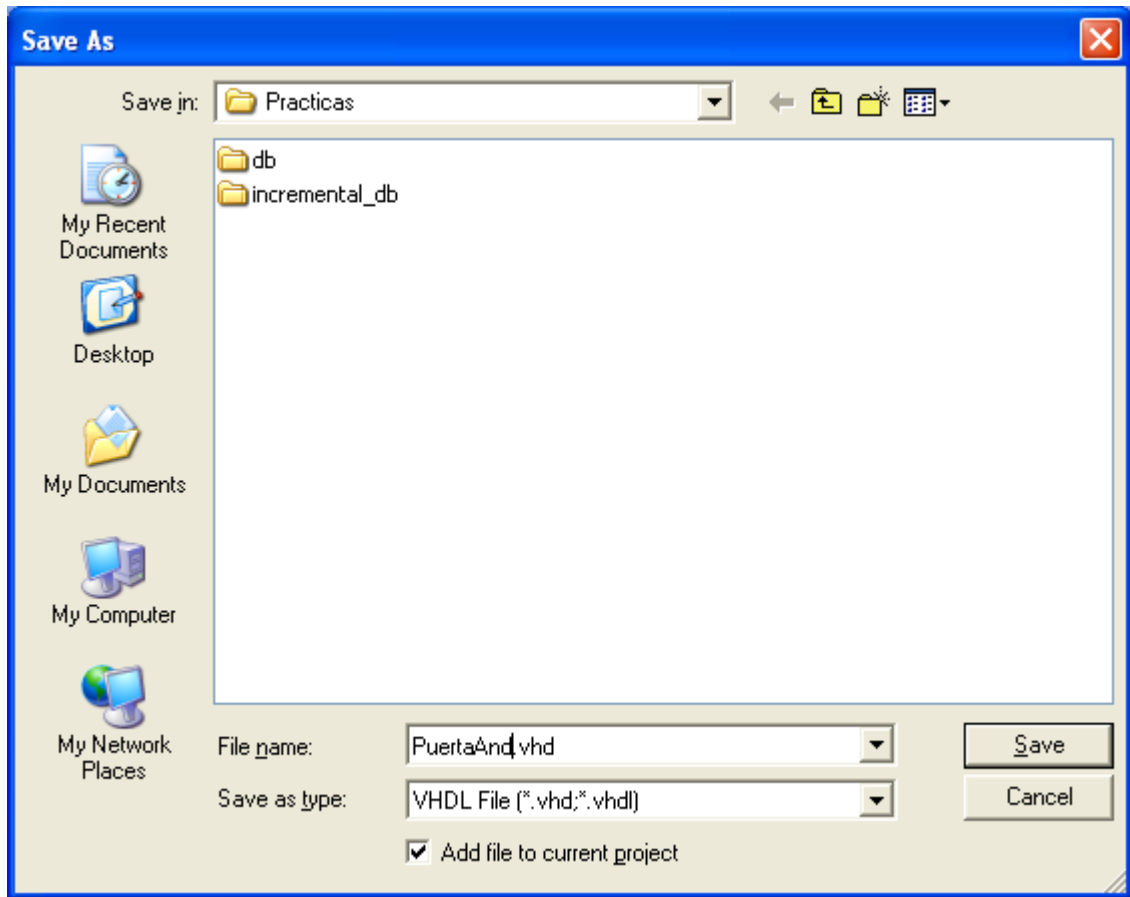


Elegir NO > Finish.

Paso 3. Crear archivo: File > New > VHDL File > OK.

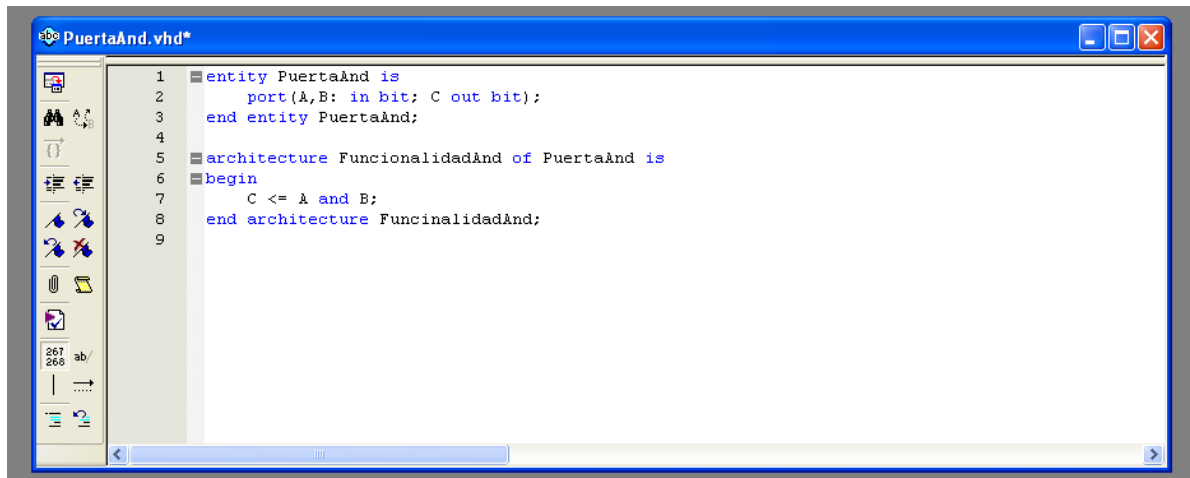


Paso 4. Cambiar de nombre: File > Save As (los nombres de archivo y entidad deben ser iguales).



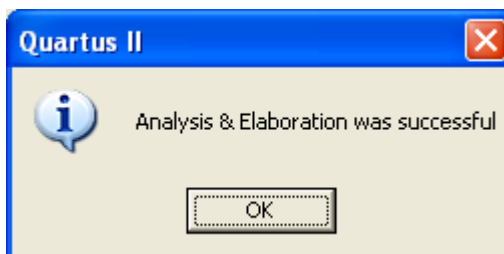
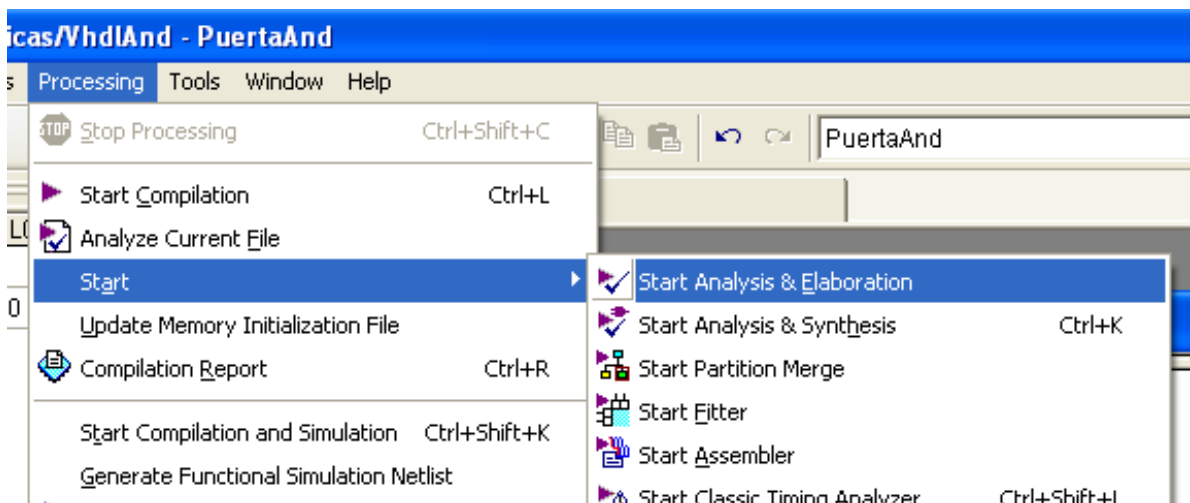
Paso 5. Escribir código:

Instrucción	Significado
entity PuertaAnd is	Nombre de la entidad
port (A,B: in bit ; C: out bit);	A,B son entradas; C es salida
end entity PuertaAnd;	Termina la especificación de la entidad
architecture FuncionalidadAnd of PuertaAnd is	
Begin	Establece las operaciones que se deben hacer con las variables
C <= A and B;	
end architecture FuncionalidadAnd;	

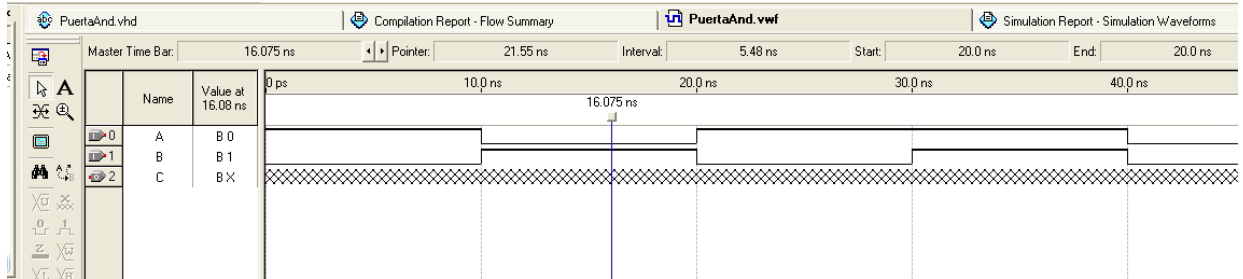


```
1 entity PuertaAnd is
2     port(A,B: in bit; C out bit);
3 end entity PuertaAnd;
4
5 architecture FuncionalidadAnd of PuertaAnd is
6 begin
7     C <= A and B;
8 end architecture FuncionalidadAnd;
9
```

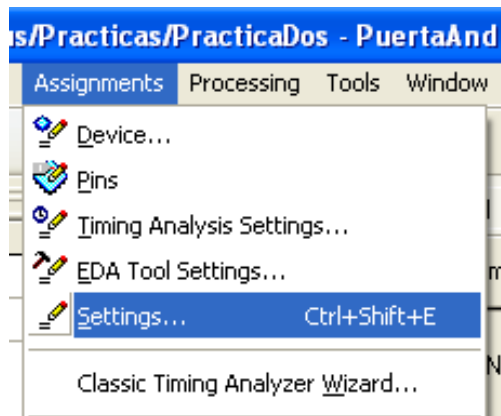
Paso 6. Compilar parcialmente: Processing > Start > Start Analysis & Colaboration.

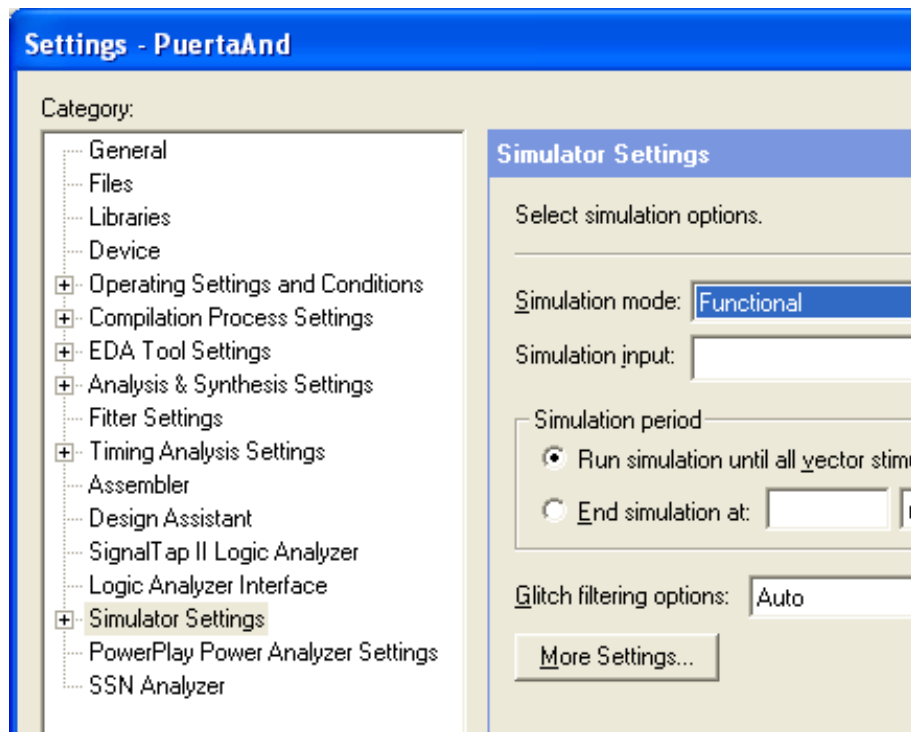


Paso 7. Repetir los pasos 11 a 15 de la práctica uno.

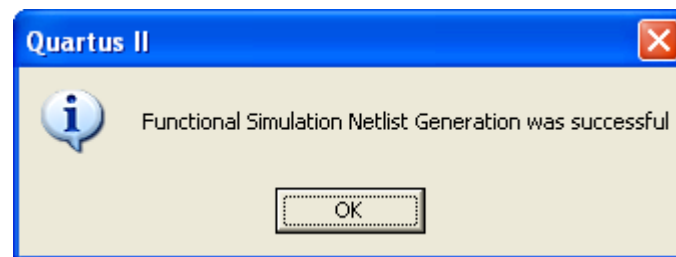
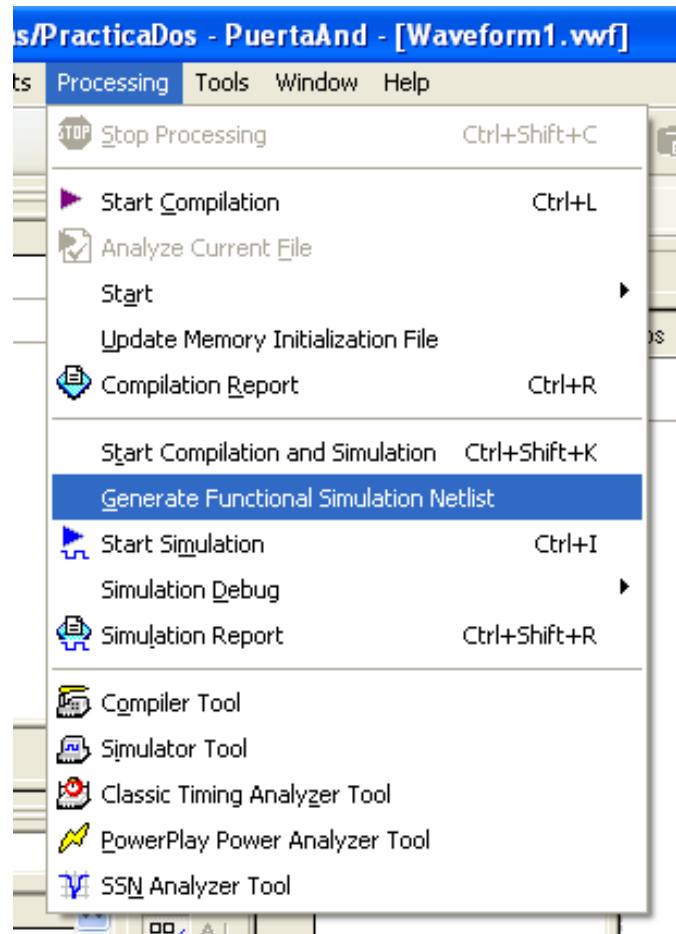


Paso 8. Cambiar el modo de simulación: Assignments > Settings > dentro de las opciones de Simulator Settings, elegir Functional en Simulation mode > OK.



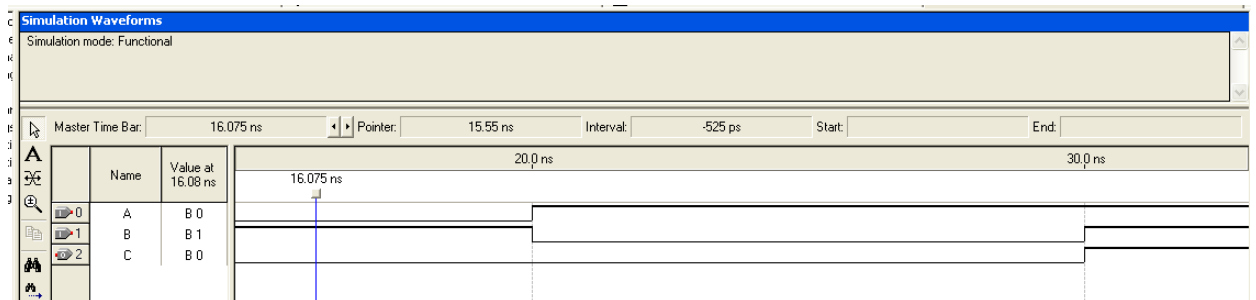
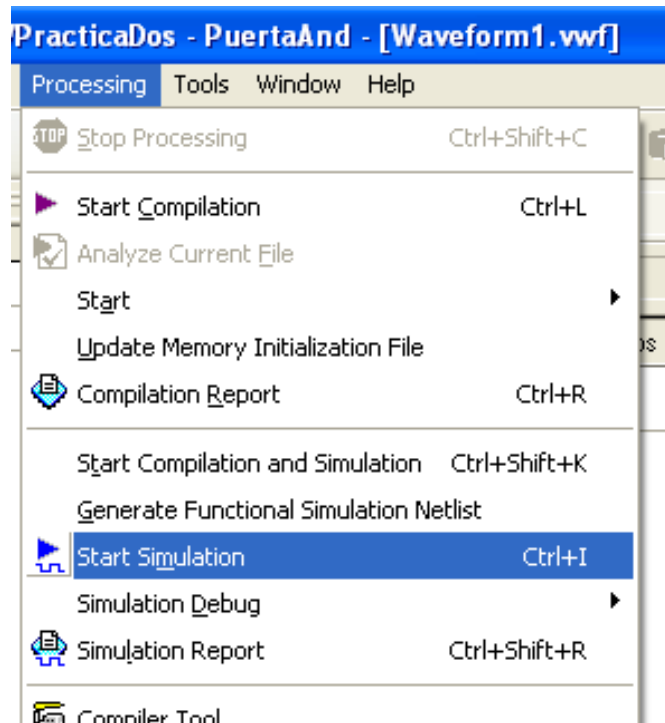


Paso 9. Generación de parámetros: Processing > Generate Functional Simulation Netlist.



Paso 10. Guardar.

Paso 11. Iniciar simulación.



Paso 12. Cerrar proyecto: File > Close Project.

AUTOEVALUACIÓN

Elige el inciso correcto.

A. No es un dispositivo programable por el usuario.

- a) PLA b) ASIC c) PAL d) GA

B. Son tablas de búsqueda.

- a) IOT b) PAL c) LUT d) CLB

C. Es usa electricidad para borrarla.

- a) ROM b) RAM c) EPROM d) EEPROM

D. Ésta forma establece qué hace un objeto en VHDL.

- a) Comportamiento b) Datos c) Estructural d) Lenguaje

E. Reutiliza especificaciones que ya se encuentran elaboradas.

- a) **library** b) **entity** c) **port** d) **component**

Completa las frases.

F. La _____ se borra con rayos UV.

G. Los _____ pueden implementar funciones lógicas multinivel.

H. Los HDLs permiten

.

I. Cualquier función de lógica combinacional, puede se expresada en

_____.

J. El proceso de diseño de sistemas digitales se hace más eficiente con la ayuda de _____.

RESPUESTAS DE LA AUTOEVALUACIÓN

A. b.

B. c.

C. d.

D. a.

E. a.

F. EPROM.

G. FPGAs.

H. Expresar diseños a través de texto.

I. Forma de suma (OR) de productos (AND).

J. *Software* CAD.

BIBLIOGRAFÍA

- Brown, Stephen, *Fundamentos de lógica digital con diseño VHDL*, México, 2a. Ed., Mc Graw-Hill, 2006.
- Cavanagh, Joseph, *Verilog VHDL: digital design and modeling*, CRC Pres, 2007.
- Feynman, Richard, *Conferencias sobre computación*, España, Editorial Crítica, 2003.
- Floyd, Thomas, *Digital Fundamentals*, USA, Pearson 9th ed., 2006.
- Maini, Anil, *Digital Electronics: principles, devices & applications*, Great Britain, John Wiley & Sons Ltd, 2007.
- Mano, Morris, *Fundamentos de diseño lógico y de computadoras*, 3a. Ed., Pearson Educación, España, 2005.
- Stallings, William, *Organización y arquitectura de computadores*, México, Pearson Educación de México, 2006.
- Tanenbaum, Andrew, *Organización de computadoras un enfoque estructurado*, México, Prentice Hall, 2000.
- Tocci, Ronald, *Digital Systems: principles & applications*, USA, Pearson 10th ed. 2007.
- Wakerly, John, *Diseño digital: principios y prácticas*, México, Pearson, 2001.

GLOSARIO

Arquitectura. Es la distribución interna de los componentes de un dispositivo electrónico, los canales de interconexión entre estos y hacia el exterior

Bit. Acrónimo de *binary digit*, representa el valor de cero o uno.

Circuito integrado. Conjunto de componentes eléctricos y electrónicos montados sobre una superficie semiconductor.

Compuerta. Componente electrónico que realiza operaciones lógicas.

Computadora. Dispositivo electrónico que recibe datos, los procesa e informa los resultados.

Encapsulado. Forma física de un circuito integrado.

Oscilador. Circuito que produce señales continuas de forma sinusoidal o cuadrada.

Puerta. Sinónimo de compuerta.

Semiconductor. Material que puede conducir carga eléctrica o dejar de hacerlo, dependiendo de su temperatura.

Señal. Fenómeno físico que se usa para transmitir información.

Sistema. Elementos que trabajan juntos para lograr un fin común.