



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Apuntes complementarios
para la materia de Diseño
Digital**

MATERIAL DIDÁCTICO

Que para obtener el título de

Ingeniero Eléctrico-Electrónico

P R E S E N T A

Arlette Paola Olvera Fuentes

ASESOR DE MATERIAL DIDÁCTICO

M.I Vicente Flores Olvera



Ciudad Universitaria, Cd. Mx., 2022

A través de este tiempo he llegado a una importante conclusión, para mí los circuitos se parecen a los humanos, si hacemos el símil encontraremos muchas similitudes.....

Por sus líneas de transmisión circula la energía que los mueve, presentan capacidad de memoria, corrigen sus errores al tener retroalimentación, cada vez los encontramos en muchos lugares más. Al igual que nosotros su corazón pulsa con cada ciclo de reloj, presentan estado de ánimo altos o bajos..... e incluso he llegado a pensar que tienen alma, porque.....

Cuando un led se prende algo en mí se ilumina...

Cuando un led está intermitente sé que dentro del circuito algo nuevo está por ocurrir

Cuando un led se paga, sé que ha llegado la hora de descansar y dormir



AGRADECIMIENTOS:

¡A ti Dios! te dedico este y todos mis logros pasados y futuros.

De todo corazón te agradezco por estar conmigo siempre y en todo momento, siendo Padre y amigo fiel en el transcurso de mi vida personal y profesional, por cuidarme con amor y cariño en la enfermedad y la salud siendo la más hermosa y grata compañía.

A María Santísima por escucharme en mis oraciones y consolarme en mis aflicciones volviéndolas inmensas alegrías.

Pero no he labrado sola este camino; las lágrimas que mi familia ha derramado por mí; han dado las más bellas flores de cariño incondicional que solo puede brindar la unión familiar y la bendición del todopoderoso.

Por ello dedico este trabajo y en especial mi esfuerzo a lo largo de estos años con gratitud e inmenso amor a:

Mi amadísima madre Juanita Fuentes Pérez, quien me ha inspirado cada día con sus frases y amor incondicional a superar mis propios límites, impulsándome a ponerme de pie cuando no podía; cuando mi alma agotada y exhausta se hubiera rendido, ella con sus palabras, me ayudo a recobrar la paz, la estabilidad y la fe en un mundo mejor, un mundo donde alcanzar mis sueños era posible si estudiaba con disciplina. Gracias por sus horas de trabajo con ello conseguí los libros que necesitaba para mi estudio y poder decirte hoy: ¡lo logramos madre de mi vida y de mi alma! ¡Te amo con todo mi corazón, nunca me sueltes de tú mano mamita!

Mi preciosa, amadísima y tierna hermanita Normita Elizabeth Olvera Fuentes una persona especial, que me ha acompañado en las situaciones más difíciles sin dar marcha atrás. Ella, hermana, amiga, consejera, maestra; cuyas virtudes me han cubierto para darme abrigo e iluminar mi camino por el sendero de la verdad y el conocimiento buscando siempre mi superación personal. Gracias por tu amor y esfuerzo incansable y tus horas de desvelo y constantes preocupaciones hoy te puedo decir ¡Lo logramos hermanita, gracias por confiar en mí! ¡Te amo gracias por ser luz y guía en mi vida hermanita! Sin ti mi vida no sería la

misma, sin ti no sería quien soy ahora. Dios te bendiga y te guarde siempre.

Mi sabio y amadísimo padre Armando Olvera, por sus innumerables consejos y apoyo incondicional. Sus palabras me han ayudado a abrirme camino en la vida y la escuela. Gracias por darme una infancia muy feliz. Gracias por ser mi padre y darme tantas cosas bellas en la vida como tu gran amor y cariño.

¡Dios te bendiga! ¡Te amo papá gracias por ser el regalo de Dios que me dio para actuar con prudencia y caridad!

Para mi querida y amadísima madrina Lupita Fuentes Pérez, muchas gracias por tu cariño, tu paciencia y todos tus consejos y tu apoyo constante en momentos difíciles. Para una alma bella y noble. ¡Te quiero mucha madrina Dios te bendiga! Eres la segunda madre que la vida me dio.

Para mi amadísima y queridísima tía Rafita, te agradezco todos los momentos que jugaste conmigo y por todas las bellas historias que me contaste de niña y aún de grande, por tu poesía y palabras que me hacen sentir mejor y muy feliz, gracias por ser una parte tan importante en mi vida. Gracias por crecer a mi lado y conmigo. ¡Te quiero mucho Rafita con un cariño que traspasa las barreras de la edad y el pensamiento! Gracias por hacerme tan feliz. Contigo aprendí que el tiempo no tiene edad.

A mi querida y gran hermana Verónica Pérez Avalos a quien quiero mucho y considero parte importante de mi vida, gracias por escucharme y apoyarme en momentos difíciles. Si bien no llevamos la misma sangre, compartimos un lazo de amor y cariño familiar. Gracias por ser mi guía y apoyo en quien puedo confiar. Amada Hermanita que Dios te llene de amor y bendiciones.

Para mi querida tía Vicenta Benavides muchas gracias por todo tu amor y cariño, por todos tus consejos.

Con mucho cariño para mi queridísima madrina Lolita Villaseñor por todo su amor, sus palabras y su hermosa compañía son un valioso y grato tesoro en las páginas de mi vida. Dios la bendiga. Te quiero mucha madrina gracias por ser como eres.

Con mucho cariño para mi querida amiga María Enriqueta Villaseñor y su hermano José Enrique González muchas gracias por cuidarme y estar conmigo, su amistad es un gran tesoro.

A mi tío Paulino gracias por compartir tu tiempo conmigo y enseñarme a jugar damas chinas y españolas.

M.I Miguel Figueroa Bustos muchas gracias ingeniero, porque sus consejos me han abierto el camino hacia rumbos mejores y más hermosos, considero que conocerlo ha sido una bendición porque gracias a su apoyo he podido realizar muchos sueños uno de ellos mi titulación. Dios lo bendiga siempre y en todo momento. Le pido a Dios lo cuide en todos sus caminos y siempre le brinde lo mejor de la vida. ¡Gracias por creer en mí y ser un amigo y padre académico que me cuida durante mi estancia en la Facultad! Le tengo gran afecto.

M.I José Gonzalo Guerrero Zepeda muchas gracias ingeniero, por todo el apoyo y la confianza depositada, permítame expresarle mi más profundo agradecimiento por creer siempre en mí, e impulsarme a seguir adelante con la mejor de las sonrisas y una actitud positiva, alegre y jovial. Pido a Dios lo llene de salud y éxito en todos los sentidos. ¡Gracias por ser un amigo y padre académico que me cuida durante mi estancia en la Facultad! Le tengo gran afecto.

Con profunda gratitud y estima a M.I Vicente Flores y su esposa María del Socorro por sus enseñanzas a través de este tiempo, por su dedicación, paciencia, apoyo y comprensión.

Con gran afecto, estima y cariño a la Lic. Angélica Gutiérrez Vázquez muchas gracias por todo su apoyo y su valioso tiempo para ayudarme con mis trámites durante mi trayectoria académica. Porque sin su apoyo y su esfuerzo mi titulación no sería posible.

Ing. Salvador Zamora muchas gracias por todas sus palabras, por su tiempo para platicarle algunas decepciones en mi camino, pero sobre todo nunca olvidare el día que gracias a usted y a su instrucción funciono los cambios de velocidad del carrito inteligente, unos de los días más felices de mi vida.

Sra. Lupita Marroquín y su hija Verito Marroquín, gracias por tratarme como un miembro más de su familia y por ayudarme cuando más lo necesitaba. He disfrutado mucho todas las pláticas que hemos tenido con ustedes.

A mi amigo Gerardo Gómez Soto por su apoyo en momentos de debilidad y soledad.

Con cariño para mi amigo Fabián Dolores quien ha estado conmigo y mi familia en momentos difíciles, gracias por ser apoyo, consuelo y una franca y sincera amistad. Dios te bendiga.

Jairo Zubeldía Gama gracias amigo por tu apoyo y por escucharme siempre, sé que eres hombre de pocas palabras, pero grandes hechos, por estar cuando te necesito, por tu paciencia, compañía, por los momentos que platicamos, tu buen humor y por todo aquello que me regalas que solo puede verse con el corazón y el alma. ¡Te quiero mucho porque me impulsas a seguir creciendo y mirar con firmeza hacia adelante!

Con mucho cariño para mis queridas amigas Yamilett, Rosita, Alexandra e Isabella Loya a quienes considero parte de importante y especial de mi vida y familia. Dios las bendiga y las cuide mucho.

Con gran afecto para mi amigo él M.I. Juan Carlos Cedeño Vázquez; gracias por escucharme, son muchas las cosas que de ti he aprendido, gracias por ser mi amigo. Gracias por estar caminado conmigo a través de estos años; tu amistad me dio la fuerza muchas veces para seguir adelante, eres el amigo; que me vio llorar, reír, fracasar y triunfar. Gracias por estos años de apoyo incondicional, me siento afortunada de conocerte, tu amistad es uno de los regalos más bellos que Dios me ha brindado, tu amistad es luz que ilumina mi vida.

A mi querida amiga Belén Gutiérrez Ortega muchas gracias por tu cariño y comprensión nos conocemos poco, pero ha sido el tiempo suficiente para darme cuenta de que eres una gran persona, amiga y confidente, Dios te bendiga, muchas gracias por la canción que me compartiste y todos tus consejos. Los llevo con cariño en el equipaje de mi vida.

Mi infinito agradecimiento a mi amiga la señora Rebeca Ortega Horta y Liliana Gutiérrez Ortega por estar conmigo en uno de los momentos más difíciles de mi vida, les estoy profundamente agradecida gracias por su apoyo, Dios las bendiga las quiero mucho.

Agradezco infinitamente al Dr. Carlos Gay García y Corinne Otto por su apoyo que me brindaron a través de mi hermana. Para ustedes con mucho cariño: mi agradecimiento y este esfuerzo y trabajo, que Dios los colme de bendiciones infinitas. Mil gracias.

A la Sritas. Rubí y Fernanda con mucho cariño, muchas gracias por su apoyo para llevar acabo mis trámites académicos les agradezco su valioso tiempo y comprensión.

Con mucho cariño para mi amiga Ángeles, gracias por tu amistad, tus consejos y tu apoyo, los atesoro enormemente, hemos crecido y llorado juntas, muchas gracias por todo, que Dios te bendiga a ti y a tu familia.

Con cariño y un sincero afecto para mis queridos y queridas amistades por sus frases y consejos; por su grata compañía, así como su ayuda invaluable apoyo, porque gracias a eso he llegado a ser la mejor persona que puedo ser y que aún sigue creciendo y cambiando para bien, porque sin ellos mi vida no sería la misma y con ellos se vuelve una hermosa experiencia el vivir, su amistad es tu tesoro invaluable.

M.I Miguel Figueroa Bustos

M.I José Gonzalo Guerrero Zepeda

Ing. José Salvador Zamora Alarcón

Dr. José Ismael Martínez

Lic. Bárbara Byer Clark

Ing. Pablo Pérez Alcaraz

Ing. Alfonso Ibarra Castillo

Lic. Angélica Gutiérrez Vázquez

Antonia Rubí Mendoza

M.I María del Socorro Guevara Rodríguez

M.I Flores Olvera Vicente

M.I Juan Carlos Cedeño Vázquez

Alejandro Carretero Ortega

Padre Rodiber Izquierdo

Monseñor J. Guadalupe Aguilera Oviedo

Padre José de Jesús León Reyes

Padre Marcelino Monroy Tolentino

Padre. Cristian

Rosi Álvarez Del Castillo Alcántara

Señor Mauro Carmona Velázquez

Señora María Camargo Pérez

Ing. Rogelio Bolio García

Ing. Jesús María Francisco Hernández Morales

Dra. Anahí Olvera Carrillo

Dr. Arturo Linares

Blanca Ramos Anastasio

Dra. Rodríguez

Dra. Sara Piña
Dr. Fernando
Dr. Fernando Ceja
Dr. Daniel
Lic. María de la Paz Adelia Peña Clavel
Edú Alberto Cruz Lecona
Dra. María del Pilar Cuenca Verde
Dr. González
Dr. Marcos Armela
Dr. Carlos Gay García
Sra. Dolores Villaseñor
Ing. José Villaseñor
Lic. Quetita Villaseñor
Melanie Slone
Margaret Bott
Octavio García Marín
Sra. Lupita Juárez
Sra. Mary Huesca
Lenina
Josefina
Dr. Fermín Matute

Para todas las personas que he encontrado en mi vida y me han tomado de la mano para volar cada vez más alto, cada vez más lejos.

Para mí; porque este es el principio de un sueño que tenía de niña y que hoy se ve realizado gracias a un esfuerzo constante, pero sobre todo a la lealtad conmigo misma y mis principios de rectitud y de justicia.

Agradezco de manera especial a:

- ❖ M.I MIGUEL FIGUEROA BUSTOS**
- ❖ M.I VICENTE FLORES OLVERA**
- ❖ M.C EDGAR BALDEMAR AGUADO CRUZ**
- ❖ M.I MARIA DEL SOCORRO GUEVARA RODRIGUEZ**
- ❖ DR. SAUL DE LA ROSA NIEVES**

Por compartir este momento tan especial y tan esperado para mi familia, amigos y para mi: “Mi titulación”.

Sé que este día no pasara a la historia como el día que el hombre alcanzo la luna, pero si pasara a mi historia como el día que alcance mis sueños, que descubrí que todo es posible como me dijeron mis padres desde pequeña, que todo es bello como me dijo mi hermana y que todo es constancia, empeño y disciplina como me enseñaron ustedes mis profesores y ahora jurados de uno de los días que nunca olvidare, que recordare con mucho cariño, con mucha satisfacción, porque si bien a veces fallé también triunfe y un examen profesional es más que un examen de conocimientos es un examen de vida, de actitud, de valor que solo se supera con amor al conocimiento, con pasión a la carrera y con la virtud que engendra la ética en la ciencia y sobre todo gracias a su apoyo.

Deseo expresar mi gratitud al M.I Flores Olvera Vicente por su valioso tiempo y sobre todo su dedicación para dirigirme y orientarme en la creación de los APUNTES COMPLEMENTARIOS PARA LA MATERIA DE DISEÑO DIGITAL.

Prefacio

Estimados profesores y alumnos, es un gusto para nosotros presentarles los presentes Apuntes complementarios para la materia de Diseño Digital; para que todo aquel que se sienta motivado por estudiar la materia, encuentre en este recurso los conocimientos necesarios para desarrollarse con éxito en este campo de la electrónica.

Como sabemos el mundo de los circuitos integrados y el conocimiento sobre el cual estos descansan; está siempre en constante cambio y con continuas mejoras, no obstante, las bases son siempre imprescindibles para poder hacer posible el avance de conocimiento.

El presente material; dirigido por el M.I Flores Olvera Vicente y elaborado por la alumna Olvera Fuentes Arlette Paola tiene como finalidad; brindar a los alumnos de la carrera de Ingeniería Eléctrica Electrónica una herramienta para apoyar la adquisición y aplicación de los conocimientos, obtenidos en materias como Diseño Digital, en la cual se manejan temas necesarios para comprender las aplicaciones de la electrónica en los diversos campos de conocimiento.

Sin duda alguna; el contenido de este manual será de gran ayuda.

Les agradeceremos sus comentarios y sugerencias.

Atentamente los autores.

M.I Flores Olvera Vicente

Arlette Paola Olvera Fuentes alumnafi@yahoo.com.mx

Contenido

1. INTRODUCCIÓN GENERAL.....	23
1.1. OBJETIVO.....	24
1.2. DEFINICIÓN DEL PROBLEMA.....	24
1.3. METODOLOGÍA.....	25
1.4. ANTECEDENTES.....	25
1.5. DESCRIPCIÓN DEL MATERIAL DIDÁCTICO QUE SE REALIZARÁ.....	25
1.6. ALCANCES.....	25
2. SISTEMAS DIGITALES.....	26
2.1. APLICACIONES DE LOS SISTEMAS DIGITALES.....	27
2.2. ¿Cómo se aplican los sistemas digitales en los aparatos anteriores?.....	33
3. SISTEMAS NUMÉRICOS Y CÓDIGOS.....	35
3.1. SISTEMAS NUMÉRICOS.....	35
4. CODIGOS.....	60
4.1 CODIGO GRAY.....	60
5. OPERACIONES ARITMETICAS Y BINARIAS.....	61
5.1 OPERACIONES ARITMÉTICAS.....	61
5.2 OPERACIONES BINARIAS.....	63
6. COMPUERTAS LÓGICAS.....	70
6.1 EL INVERSOR.....	70
6.2 LA COMPUERTA AND.....	72
6.3 LA COMPUERTA OR.....	78
6.4 LA COMPUERTA NAND.....	82
6.5 LA COMPUERTA NOR.....	84
6.6 LA COMPUERTA OR EXCLUSIVA.....	89
6.7 LA COMPUERTA NOR EXCLUSIVA.....	89
7. ALGEBRA BOOLEANA (EXPRESIONES ALGEBRAICAS).....	90
7.1 LEYES Y PROPIEDADES EN EL ALGEBRA BOOLENA.....	95
8. Universalidad de las compuertas NAND y NOR.....	104
8.1 LA COMPUERTA NAND.....	105

8.2 LA COMPUERTA NOR	108
8.3 EJEMPLOS DE APLICACIÓN	113
9. Mapas de Karnaugh	117
9.1 EXPRESIONES SOP Y POS	134
10. FORMAS CANONICAS Y ESTANDÁR	137
10.1 Método de Quine-McCluskey	170
11. SUMA DE PRODUCTOS (SOP) Y PRODUCTO DE SUMAS (POS).....	176
11.1 SUMA DE PRODUCTOS (SOP).....	176
12. Circuitos combinacionales.....	182
12.1 Análisis de circuitos combinacionales.....	182
13. Escalas de Integración MSI.....	184
14. Codificadores, Decodificadores, Multiplexores, Demultiplexores, Comparadores y Generadores de paridad.....	191
14.1 CODIFICADORES	191
14.2 Simulación de un codificador de 4 a 2	192
14.4 APLICACIÓN DE LOS CODIFICADORES	199
14.5 DECODIFICADORES	199
14.6 BCD DECIMAL CODIFICADO EN BINARIO.....	206
14.7 Simulación de un decodificador 2 a 4 usando VHDL.....	210
14.8 Multiplexores/Selectores de Datos.....	213
14.9 DISEÑO DE UN MULTIPLEXOR	214
14.10 Ejemplo de un multiplexor empleando VHDL.....	220
14.11 DEMULTIPLEXORES	224
14.12 Simulación de demultiplexor	226
14.13 COMO SE EXPANDE UN COMPARADOR.....	231
14.14 DETECTORES DE PARIDAD	232
15. SUMADOR COMPLETO Y RESTADOR COMPLETO	235
15.1 SUMADOR COMPLETO	235
15.2 MEDIO SUMADOR.....	236
15.3 SUMADOR COMPLETO	237
15.4 RESTADOR MEDIO.....	239
15.5 RESTADOR COMPLETO.....	240

16.	Memorias RAM y ROM.....	241
16.1	OPERACIONES BÁSICAS DE LA MEMORIA	242
16.3	LAS MEMORIAS ROM	249
17.	Circuitos secuenciales (Circuitos secuenciales latch SR,D, JK y T y flip-flops SR,T,JK) 254	
17.1	INTRODUCCIÓN	254
17.2	LOS LATCH.....	255
17.3	Registros y contadores	269
17.4	CONTADORES.....	274
17.5	CONTADORES SÍNCRONOS	274
17.6	CONTADORES ASÍNCRONOS	279
18.	Máquina de estado.....	281
18.1	INTRODUCCIÓN	281
18.2	ESTRUCTURA PARA CONSTRUIR MÁQUINAS MEALY Y MOORE	283
18.3	MÁQUINAS DE ESTADO.....	287
19.	Quartus	292
20.	BIBLIOGRAFÍA.....	345
21.	BIBLIOGRAFÍA DE INTERNET	346

Índice de Tablas

Tabla 1 Números decimales	38
Tabla 2 Números 3 en decimal y binario	40
Tabla 3 Número del 0-7 en decimal y binario	40
Tabla 4 Números del 0-15 en hexadecimal, decimal y binario	52
Tabla 5 Estados de una compuerta not	71
Tabla 6 Tabla de verdad de la compuerta AND de 2 entradas	73
Tabla 7 Tabla de verdad de la compuerta AND para tres variables	77
Tabla 8 Tabla de verdad para la compuerta OR de dos entradas	78
Tabla 9 Tabla de verdad para la compuerta OR de tres entradas	81
Tabla 10 Tabla de verdad para la compuerta NAND para dos variables	83
Tabla 11 Tabla de verdad para la compuerta NAND de tres variables	84
Tabla 12 Tabla de verdad para la compuerta NOR de 2 variables	85
Tabla 13 Tabla de verdad para la compuerta NOR de tres variables	85
Tabla 14 Tabla de verdad para la compuerta NAND de dos entradas	86
Tabla 15 Tabla de verdad de la compuerta OR de 2 entradas	86
Tabla 16 Tabla de verdad de la compuerta OR negada para dos entradas	86
Tabla 17 Tabla de verdad para la compuerta NOR de dos entradas	86
Tabla 18 Tabla de verdad para la compuerta AND de dos entradas	87
Tabla 19 Tabla de verdad para la compuerta AND negada	87
Tabla 20 Tabla de verdad para la compuerta XOR de dos entradas	89
Tabla 21 Tabla de verdad para la compuerta XNOR de dos entradas	90
Tabla 22 Tabla de verdad de la compuerta OR para dos variables de entrada	91
Tabla 23 Tabla de verdad para la ecuación propuesta	92
Tabla 24 Tabla de verdad de la ecuación vista	93
Tabla 25 Tabla de verdad con la respuesta del ejemplo anterior	93
Tabla 26 Tabla de verdad para la ecuación vista	94
Tabla 27 Tabla de verdad del ejemplo mencionado	100
Tabla 28 Tabla de verdad para las compuertas de la figura 22	101
Tabla 29 Tabla de verdad para la función lógica planteada	101
Tabla 30 Tabla de verdad para la función plateada	102
Tabla 31 Tabla de verdad para el ejercicio de la fábrica de pan	113
Tabla 32 Tabla de verdad del siguiente proceso de la fábrica de pan	114
Tabla 33 Tabla de verdad de la aplicación de la compuerta NAND	115
Tabla 34 Tabla de verdad de un sistema de seguridad de una central camionera	115
Tabla 35 Valores de entrada del mapa de Karnaugh	125
Tabla 36 Número del 0 al 7 en decimal y binario	128
Tabla 37 Números del 0 al 15 en decimal y binario	129
Tabla 38 Tabla de mintérminos (0,3,5,6)	133
Tabla 39 Agrupación de términos en base al número de mintérminos que tienen	133
Tabla 40 Ejemplo de tabla de verdad donde la función vale 1	137

Tabla 41	Tabla donde se muestra la colocación del guion en el término que difieren	173
Tabla 42	Tabla para obtener la mínima expresión de la función IF	173
Tabla 43	Tabla de los grupos que contienen uno, dos, tres y cinco unos	175
Tabla 44	Tabla de los términos esenciales o implicantes primos	175
Tabla 45	Tabla de verdad para la evaluación de la función.	182
Tabla 46	Evolución d la capacidad de integración	191
Tabla 47	Tabla de verdad del decodificador de 10 a 4 líneas	195
Tabla 48	Tabla de verdad de un decodificador de 4 a 16 líneas	204
Tabla 49	Tabla de verdad para el multiplexor del ejemplo	215
Tabla 50	Tabla de verdad de la función implementada para un decodificador de 4 a 1 líneas	218
Tabla 51	Tabla de verdad para un demultiplexor de 1 a 8 líneas	225
Tabla 52	Tabla para el análisis de paridad par e impar	232
Tabla 53	Tabla de compuerta XOR	232
Tabla 54	Tabla de paridad par e impar	233
Tabla 55	Tabla de compuerta XOR	233
Tabla 56	Tabla de paridad par e impar con compuertas XOR	234
Tabla 57	Tabla de verdad para un medio sumador de dos bits	236
Tabla 58	Tabla de verdad para un sumador completo	237
Tabla 59	Tabla de verdad para un medio restador	239
Tabla 60	Tabla de verdad de un restador completo	240
Tabla 61	tabla de verdad del latch SR	256
Tabla 62	Tabla característica de latch SR	256
Tabla 63	Tabla de verdad de latch SR con compuertas NAND	257
Tabla 64	Tabla cacterística de lacth SR con compuertas NAND	257
Tabla 65	Tabla de verdad de FF SR	260
Tabla 66	Tabla de verdad de la ecuación característica de un FF SR	261
Tabla 67	Tabla de verdad de FF D	262
Tabla 68	Característica de FF D	262
Tabla 69	Mapa de Karnaugh de Función reducida del FF D	263
Tabla 70	Tabla de verdad de FF T	265
Tabla 71	Tabla característica de FF T	266
Tabla 72	Registro de corrimiento	274
Tabla 73	Contador de tres bits síncrono	275

Índice de Figuras

<i>Figura 1 Dispositivos médicos para obtener señales en forma analógica y digital</i>	26
<i>Figura 2 Termómetro analógico</i>	27
<i>Figura 3 Termómetro digital</i>	27
<i>Figura 4 Aplicaciones en el área Industrial</i>	28
<i>Figura 5 Aplicaciones en el área química</i>	28
<i>Figura 6 Aplicaciones en el área de sistemas</i>	29
<i>Figura 7 Oxímetro de pulso</i>	30
<i>Figura 8 Baumanómetro</i>	31
<i>Figura 9 glucómetro</i>	32
<i>Figura 10 Monitor Doppler fetal</i>	32
<i>Figura 11 circuito interno de un oxímetro de pulso</i>	34
<i>Figura 12 Símbolo lógico para la compuerta not o negada</i>	70
<i>Figura 13 Representación de la compuerta not con el estado encendido y apagado de un foco</i>	71
<i>Figura 14 Circuito físico de la compuerta NOT en estado apagado o bajo</i>	72
<i>Figura 15 Circuito físico de la compuerta NOT en estado encendido o alto</i>	72
<i>Figura 16 Símbolo lógico para una compuerta AND de dos entradas</i>	73
<i>Figura 17 Circuito físico de la compuerta AND cuando los dos estados son altos</i>	74
<i>Figura 18 Circuito físico de la compuerta AND cuando los dos estados están en bajo</i>	74
<i>Figura 19 Circuito físico de la compuerta AND cuando un estado es bajo y el otro alto</i>	74
<i>Figura 20 Circuito físico de la compuerta AND cuando un estado es alto y otro bajo</i>	75
<i>Figura 21 Símbolo lógico para una compuerta OR de dos entradas</i>	78
<i>Figura 22 Circuito físico de la compuerta OR cuando los dos estados están en alto</i>	79
<i>Figura 23 Circuito físico de la compuerta OR cuando un estado está en bajo y otro en alto</i>	79
<i>Figura 24 Circuito físico de la compuerta OR cuando una de las entradas esta en alto y la otra en bajo</i>	79
<i>Figura 25 Circuito físico de la compuerta OR cuando las dos entradas están en bajo.</i>	79
<i>Figura 26 Símbolo lógico para una compuerta NAND de dos entradas</i>	83
<i>Figura 27 Compuerta NOR con dos entradas</i>	85
<i>Figura 28 Símbolo lógico para la compuerta AND negada con dos entradas respectivamente</i>	87
<i>Figura 29 compuerta XOR</i>	89
<i>Figura 30 Símbolo lógico de la compuerta XNOR con dos entradas</i>	89
<i>Figura 31 Implementación de la expresión lógica $C+(A+B)$ con compuertas lógicas</i>	100
<i>Figura 32 Implementación de la expresión lógica ABC con compuertas lógicas</i>	100
<i>Figura 33 Implementación de la expresión lógica</i>	101
<i>Figura 34 Implementación de la función lógica</i>	102
<i>Figura 35 Símbolo lógico de una compuerta NAND</i>	105
<i>Figura 36 Símbolo lógico de una compuerta NAND implementada como inversor</i>	105
<i>Figura 37 Diagrama de una compuerta NAND implementada para que funcione como una compuerta AND</i>	105
<i>Figura 38 Diagrama de una compuerta NAND implementada como una compuerta OR</i>	106
<i>Figura 39 Diagrama de la aplicación de compuertas NAND y algebra booleana para obtener la función de salida del circuito</i>	106

Figura 40 Diagrama de la aplicación de compuertas NAND y algebra booleana para obtener la función de salida del circuito	107
Figura 41 Aplicación de la compuerta NOR como una compuerta NOT	109
Figura 42 Aplicación de la compuerta NOR como una compuerta OR	109
Figura 43 Diagrama de la aplicación de la compuerta NOR	110
Figura 44 Diagrama de compuertas NAND para habilitar la expresión	112
Figura 45 Diagrama de compuertas NAND para habilitar la expresión	112
Figura 46 Diagrama de Venn de la operación AND	118
Figura 47 Diagrama de Venn de la operación OR	119
Figura 48 Formación de grupos de 1 del ejemplo 1F	172
Figura 49 Tabla de los minterminos de la función del ejercicio 2F	174
Figura 50 Implementación de la función lógica con compuertas	183
Figura 51 Implementación de la función lógica con compuertas	183
Figura 52 Implementación de la función lógica con compuertas	183
Figura 53 Imagen de un chip	185
Figura 54 Circuitos Integrados	186
Figura 55 circuito integrado 74LS148	187
Figura 56 Tabla de verdad del circuito 74LS48 cátodo común	189
Figura 57 Tabla de verdad del circuito 74LS47 ánodo común	189
Figura 58 Representación gráfica de la evolución de la capacidad de integración.	190
Figura 59 Diagrama de Bloques de la simulación	193
Figura 60 Codificador de 10 a 4 líneas	195
Figura 61 Implementación de un codificador de decimal a binario con compuertas diagrama tomado del libro de Floyd página 332	197
Figura 62 Codificador 74147	198
Figura 63 Construcción de un decodificador utilizando compuertas AND y NOT	202
Figura 64 Implementación de un circuito con compuertas AND y NOT que activan una salida baja	202
Figura 65 Tabla de verdad del circuito 74LS48 cátodo común	203
Figura 66 Tabla de verdad del circuito 74LS47 ánodo común	203
Figura 67 Imagen del circuito integrado 74154	204
Figura 68 Imagen de un decodificador de 1 a 16 líneas	205
Figura 69 Codificador 7447	206
Figura 70 Codificador 7448	206
Figura 71 Implementación de un Decodificador de 4 a 16 líneas para la función dada por la tabla de verdad	209
Figura 72 Simulación de decodificador de 2 a 4	211
Figura 73 Simulación de decodificador 2 a 4	211
Figura 74 Simulación de decodificador 2 a 4	212
Figura 75 Simulación de decodificador 2 a 4	212
Figura 76 Diagrama de elementos de un multiplexor	213
Figura 77 Selector de datos	214
Figura 78 Implementación de un multiplexor con compuertas	215
Figura 79 Diagrama de la construcción de un multiplexor de 16 canales	217
Figura 80 Expansión de multiplexor 4 a 1	217
Figura 81 Diagrama de un selector de datos	218
Figura 82 Expansión de un multiplexor 4:1 para obtener un mux de 16:1 imagen tomada de https://docplayer.es/251676-Tema-5-sistemas-combinacionales-msi.html	220

Figura 83 simulación de multiplexor _____	223
Figura 84 Simulación de multiplexor _____	223
Figura 85 Diagrama de elementos de un multiplexor _____	224
Figura 86 Simulación de demultiplexor _____	227
Figura 87 Simulación de demultiplexor _____	227
Figura 88 Expansión de un codificador para comparar dos números de 16 bits _____	231
Figura 89 Diagrama de medio sumador _____	236
Figura 90 Implementación de un sumador completo de 3 bits _____	238
Figura 91 Diagrama de un medio restador para dos bits _____	240
Figura 92 Mapa de Karnaugh para la función de salida resta _____	241
Figura 93 Mapa de Karnaugh para la función préstamo _____	241
Figura 94 Análisis de un entorno _____	241
Figura 95 Diagrama interno de una memoria tomado del libro de Floyd página 586 _____	244
Figura 96 Arreglo de 64 celdas _____	246
Figura 97 Cuadro sinóptico de las memorias RAM. Cuadro tomado de la página 602 del libro Tomas Floyd _	248
Figura 98 Cuadro sinóptico de las memorias ROM. Cuadro tomado de la página 588 del libro de Tomas Floyd _	250
Figura 99 Imagen autora estudiando Arlette Paola Olvera F _____	254
Figura 100 Latch SR _____	255
Figura 101 Latch con compuertas NAND _____	257
Figura 102 Diagrama de bloques de un FF _____	259
Figura 103 Diagrama de un FF SR _____	260
Figura 104 Imagen de FF D _____	262
Figura 105 FF JK _____	263
Figura 106 Tabla d verdad de FF JK _____	264
Figura 107 Tabla característica de FF JK _____	264
Figura 108 Mapa de Karnaugh de FF JK _____	264
Figura 109 FF T _____	265
Figura 110 FF SR con flanco negativo _____	269
Figura 111 FF D conectado para formar un registro _____	270
Figura 112 Dos FF SR conectados para formar un registro _____	270
Figura 113 Dos FF JK conectados para formar un registro _____	271
Figura 114 FF JK conectados en paralelo para formar un registro _____	272
Figura 115 Imagen de un mensaje en pantalla LCD _____	273
Figura 116 FF D conectado en serie _____	273
Figura 117 Contador de 3 bits síncrono _____	275
Figura 118 Implementación de contador ascendente del 0-15 _____	279
Figura 119 FF de 3 bits asíncrono _____	280
Figura 120 Diagrama de estado de la máquina expendedora de alimentos _____	284
Figura 121 Implementación del circuito de la máquina expendedora _____	286
Figura 122 diagramas de una carta ASM _____	288
Figura 123 Icono de Quartus _____	292
Figura 124 Presentación del paso 3 y 4 _____	293
Figura 125 Presentación del paso 5 _____	293
Figura 126 Presentación del paso 6 _____	294
Figura 127 Presentación del paso 7 _____	294
Figura 128 Presentación del paso 8 _____	295

<i>Figura 129</i> Presentación del paso 9	295
<i>Figura 130</i> Presentación del paso 10	296
<i>Figura 131</i> Presentación del paso 11	296
<i>Figura 132</i> SElección de Max II	297
<i>Figura 133</i> Selección de la tarjeta	297
<i>Figura 134</i> Cuadros después del proceso	298
<i>Figura 135</i> Paso 11	298
<i>Figura 136</i> Paso 12	299
<i>Figura 137</i> Paso 13	299
<i>Figura 138</i> Paso 14	300
<i>Figura 139</i> Representación del paso 15	301
<i>Figura 140</i> Paso 16	301
<i>Figura 141</i> Representación paso 17	302
<i>Figura 142</i> Paso 18	302
<i>Figura 143</i> Selección de componentes	303
<i>Figura 144</i> Ejemplo de selección de componentes	303
<i>Figura 145</i> Paso 20	304
<i>Figura 146</i> Paso 21	304
<i>Figura 147</i> Paso 22	305
<i>Figura 148</i> Paso 22 simplificado	305
<i>Figura 149</i> Paso 23	306
<i>Figura 150</i> Imagen del simbolo para las entradas y salidas	307
<i>Figura 151</i> Imagen del simbolo para las entradas y salidas	307
<i>Figura 152</i> Imagen del simbolo para las entradas y las salidas	307
<i>Figura 153</i> Imagen del símbolo para las entradas y salidas	307
<i>Figura 154</i> Como se cambia el nombre de los pines	308
<i>Figura 155</i> Cambio de nombre de pin	308
<i>Figura 156</i> Proceso de compilación	309
<i>Figura 157</i> Circuito	309
<i>Figura 158</i> Después del proceso de compilación	310
<i>Figura 159</i> Imagen del paso 2 del proceso de simulación	311
<i>Figura 160</i> Proceso de simulación	311
<i>Figura 161</i> Proceso de simulación	312
<i>Figura 162</i> Proceso de simulación	313
<i>Figura 163</i> Proceso de simulación	313
<i>Figura 164</i> Proceso de simulación	314
<i>Figura 165</i> Proceso de simulación para el paso 4	314
<i>Figura 166</i> Proceso de simulación	315
<i>Figura 167</i> Proceso de simulación paso 5	315
<i>Figura 168</i> Proceso de simulación paso 6	316
<i>Figura 169</i> Proceso de simulación	316
<i>Figura 170</i> Proceso de simulación del paso 7	317
<i>Figura 171</i> Proceso de simulación	317
<i>Figura 172</i> Proceso de simulación	318
<i>Figura 173</i> Proceso de simulación	319
<i>Figura 174</i> Proceso de simulación	319

<i>Figura 175</i> Proceso de simulación	320
<i>Figura 176</i> Proceso de simulación	320
<i>Figura 177</i> Proceso de simulación	321
<i>Figura 178</i> Proceso de simulación	321
<i>Figura 179</i> Proceso de simulación	322
<i>Figura 180</i> Proceso de simulación	322
<i>Figura 181</i> Proceso de simulación	323
<i>Figura 182</i> Proceso de simulación	324
<i>Figura 183</i> Proceso de simulación	324
<i>Figura 184</i> Proceso de simulación	325
<i>Figura 185</i> Proceso de simulación	325
<i>Figura 186</i> Proceso de simulación	326
<i>Figura 187</i> Proceso de simulación	326
<i>Figura 188</i> Proceso de simulación	327
<i>Figura 189</i> Proceso de simulación	327
<i>Figura 190</i> Proceso de simulación	328
<i>Figura 191</i> Proceso de simulación	328
<i>Figura 192</i> Proceso de simulación	329
<i>Figura 193</i> Proceso de simulación	329
<i>Figura 194</i> Proceso de simulación	330
<i>Figura 195</i> Proceso de simulación	330
<i>Figura 196</i> Proceso de simulación	331
<i>Figura 197</i> Proceso de simulación	332
<i>Figura 198</i> Proceso de simulación	332
<i>Figura 199</i> Proceso de simulación	333
<i>Figura 200</i> Proceso de simulación	333
<i>Figura 201</i> Proceso de simulación	334
<i>Figura 202</i> Proceso de simulación	334
<i>Figura 203</i> Imagen del ejemplo 1	335
<i>Figura 204</i> Resultado de la simulación del ejemplo 1	335
<i>Figura 205</i> Diagrama del ejemplo 2	336
<i>Figura 206</i> Simulación del resultado del ejemplo	338
<i>Figura 207</i> Resultado de la simulación	338
<i>Figura 208</i> Diagrama del ejemplo	339
<i>Figura 209</i> Resultado de la simulación	339
<i>Figura 210</i> Diagrama de ejemplo	340
<i>Figura 211</i> Simulación del ejemplo propuesto	342

1. INTRODUCCIÓN GENERAL

El conocimiento de la materia de Diseño Digital es una herramienta indispensable para que los alumnos se desenvuelvan en las distintas áreas en las cuales se aplican los conocimientos de forma directa o indirecta.

Es necesario que el alumno no solo presente disciplina en la adquisición de sus conocimientos si no también que cuente con los elementos necesarios para el desarrollo de los mismos.

Es por ello que el presente material cuenta con teoría y ejercicios de amplio interés para que los profesores y los alumnos tengan una base confiable que les aporte de una manera didáctica y amable el conocimiento.

En la actualidad es más evidente el uso de aparatos que tengan cada vez mayor precisión, en la medición o en la aplicación de algún procedimiento es por ello que día a día el papel del ingeniero es indispensable para el diseño y la construcción de los mismos.

Diseño Digital es una materia cuyos principios son básicos para la comprensión de una parte de los circuitos electrónicos eléctricos, de aquí es necesario conocer el alma de estos y una parte de estos lo forma dicho conocimiento.

Gracias a la comprensión del Diseño digital podemos llegar a comprender el funcionamiento de los circuitos para poder manejarlos con facilidad y restarles complejidad a los procesos.

En este manual encontrará los temas básicos del diseño digital para iniciarse en este campo de conocimiento con seguridad, por lo cual la explicación de los temas se desarrolla de manera lo más clara y concisa posible.

1.1. OBJETIVO

El objetivo del presente material; es que, a través de los apuntes, se proporcione material; que refuerce el aprendizaje y paralelamente se desarrolle la capacidad de análisis para aplicar la teoría en cuestiones prácticas de uso cotidiano que puedan ayudar a realizar diseños con mayor nivel de complejidad para lograr grandes avances en la sociedad.

1.2. DEFINICIÓN DEL PROBLEMA

Diseño Digital es una de las materias que representa un gran reto para los alumnos que desean incursionar en esta parte de la electrónica.

Por lo cual se hace presente la necesidad de que los alumnos cuenten con apoyos didácticos como Apuntes para la materia de Diseño Digital; que ayuden a los alumnos a adquirir y asimilar los conocimientos relacionados con dicha materia.

1.3. METODOLOGÍA

Se lleva a cabo una explicación a detalle de la teoría para que los lectores comprendan de una manera lo más fácil posible la teoría.

1.4. ANTECEDENTES

La materia de Diseño Digital requiere de conocimientos sólidos y un buen entendimiento de los mismos, por lo cual es indispensable que los profesores y los alumnos cuenten con un material de apoyo para la consulta de la teoría y ejercicios que aporten y puedan generar la construcción de bases firmes para el desarrollo de la materia.

1.5. DESCRIPCIÓN DEL MATERIAL DIDÁCTICO QUE SE REALIZARÁ

Se realizarán apuntes para la materia de Diseño digital, los cuales están orientados a complementar la teoría que se proporciona en la clase, para que el alumno adquiera la habilidad para que pueda implementar su conocimiento de una manera óptima y eficiente en este campo de interés.

1.6. ALCANCES

Lograr que el presente material para la Materia de Diseño Digital forme parte de los materiales de apoyo que ofrece la

Facultad de Ingeniería a sus alumnos para el estudio de la asignatura.

Que los alumnos encuentren en los apuntes complementarios para la materia de Diseño Digital, un apoyo invaluable para la adquisición y el desarrollo de sus conocimientos.

2. SISTEMAS DIGITALES

Partimos primeramente de la definición de dispositivo:

Un dispositivo es un elemento o conjunto de elementos cuya función es formar parte de un sistema simple o complejo que permita cuantificar variables en su forma digital o analógica.



Figura 1 Dispositivos médicos para obtener señales en forma analógica y digital

Un sistema digital; por ejemplo, son los dispositivos destinados a la obtención de valores del medio para que podamos establecer un rango o patrón de un fenómeno sujeto a estudio que pueda ser trabajado de manera digital es decir con los dos estados posibles 0 o 1, alto o bajo, encendido o apagado.

Por otra parte, un sistema digital es aquel que asigna un valor determinado que podría presentarse dentro de un rango.

Por ejemplo, tomemos el termómetro analógico cuando tratamos de asignar una temperatura la lectura es ambigua porque se encuentra entre dos marcas, no obstante, cuando le asignamos un valor, como podría ser 36° estamos digitalizando el sistema y como un ejemplo directo el termómetro digital.



Figura 2 Termómetro analógico



Figura 3 Termómetro digital

2.1. APLICACIONES DE LOS SISTEMAS DIGITALES

Sin duda alguna el Diseño Digital está presente en muchas áreas de gran auge en la actualidad a continuación se mencionan algunas:

Industrial

El área Industrial el Diseño lógico para que los sistemas funcionen en las mejores condiciones es de gran importancia. Por ejemplo, el conocimiento de esta materia permite diseñar los circuitos necesarios para llevar acabo el conteo de los productos, manufacturados, embolsados, procesados,

pintados, armados, en fin, todo aquello que precise de llevar a cabo una cuenta y la memoria necesaria para guardar esta información.



Figura 4 Aplicaciones en el área Industrial

Química

Es muy importante la medición de las cantidades necesarias para la elaboración de químicos, fármacos, alimentos, por lo cual llevar a cabo una cuidadosa medición es de gran importancia, ya que un mlg de más o menos, altera toda la composición química de la sustancia. Para ello es necesario contar con dispositivos construidos con la más alta tecnología que permitan llevar a cabo estas mediciones.



Figura 5 Aplicaciones en el área química

Sistemas

La elaboración de dispositivos e interfaces capaces de permitir la interacción entre humano-máquina cada vez se torna más común.

Es por ello que mejorar el diseño de los circuitos para que esto sea posible se vuelve relevante.

Esto solo se puede lograr si se comprende cómo opera el lenguaje máquina y de que forma el humano ingresa sus instrucciones para que sean llevadas a cabo por la máquina.

Ciertamente para manipular dichos sistemas lo anterior no es indispensable, pero si requiere mejorar la arquitectura y la capacidad de respuesta del dispositivo, es necesarios adentrarse en el comportamiento lógico del sistema.



Figura 6 Aplicaciones en el área de sistemas

Es de suma importancia los conocimientos relativos a esta materia para el desarrollo de equipo médico; que se tenga una capacidad mayor de resolver problemas como son el diagnóstico de enfermedades que aquejan a la sociedad en general gracias al uso de dispositivos médicos.

Área médica

A continuación, se muestran unos ejemplos de equipo médico con los cuales se puede apreciar la presencia del Diseño Digital en la tecnología de nueva era.

A continuación, algunos de los aparatos con mayor uso en los diagnósticos.

Oxímetro de pulso

Los pulmones son los encargados de llevar a cabo el proceso de respiración, es en estos donde hay un intercambio entre gases; entrando oxígeno y expulsando bióxido de carbono, este último es un estimulante para el funcionamiento de los pulmones. Gracias a estos el oxígeno pasa a la sangre, gran parte de este elemento se une a la hemoglobina, esta es una proteína ubicada en el interior de la célula de la sangre.

Esta sangre oxigenada circula por los tejidos, el oxímetro de pulso es un dispositivo foto electrónico, donde la diferencia entre el color de la hemoglobina saturada con oxígeno, la cual es de un rojo intenso y la insaturada que presenta un color más oscuro, midiendo así la intensidad de ambos tonos esta acción la realiza paralelamente con las pulsaciones del corazón.

Es decir, el dispositivo realiza la detección del pulso; posteriormente resta la intensidad de color cuando el pulso está ausente.

La intensidad de color restante es la que corresponde a la sangre oxigenada.

Así pues, en la pantalla aparece como un porcentaje de la saturación de oxígeno en la sangre



Figura 7 Oxímetro de pulso

Baumanómetro

El baumanómetro o esfigmomanómetro se utiliza para la medición de la presión arterial de una manera indirecta y no invasiva.

Existen principalmente dos tipos de baumanómetros:

El que funciona con una columna de mercurio y el digital.

El digital

A continuación, explicaremos los digitales.

Una manga inflable, la cual lleva en su interior sensores los cuales detectan la oscilación producida por la circulación de la sangre cuando viaja por una arteria y en donde se ejerce presión lo cual permite conocer el intervalo entre presión sistólica y diastólica.

Una pantalla LCD en esta se muestran los resultados de la presión sistólica y diastólica

Una microcomputadora esta contiene un reloj y una memoria



Figura 8 Baumanómetro

Glucómetro

Este dispositivo es utilizado para medir el nivel de glucosa en la sangre. Para realizar la medición se utiliza una gota de sangre del paciente que se coloca en una tira reactiva, la cual

contienen glucosa oxidasa, cuando se da la reacción, en esta se provoca un cambio de color el cual varía dependiendo de la cantidad de glucosa que hay en la sangre por ejemplo si el color es muy oscuro indicara que el nivel de glucosa es alto.



Figura 9 glucómetro

Monitor Doppler fetal

Este dispositivo es un transductor de ultrasonido, el cual fue inventado en 1958 por Edward Hon.

Como su nombre lo indica se basa en el efecto Doppler para detectar los latidos del corazón.



Figura 10 Monitor Doppler fetal

2.2. ¿Cómo se aplican los sistemas digitales en los aparatos anteriores?

Ejemplos:

La mayoría de los equipos están dotados de un teclado para que exista una interacción entre la máquina y el usuario, el teclado numérico nos ayuda para introducir un número en forma digital el cual mediante el proceso adecuado será transformado en un lenguaje que la máquina pueda entender y así llevar a cabo los procesos necesarios para tener un mayor grado de precisión.

La forma en la que interviene el diseño en esta parte es muy importante, en páginas posteriores por ejemplo se verá que se tienen circuitos que funcionan de manera síncrona y asíncrona. En base al estudio de este material se podrá llegar a la conclusión de que para la construcción de un dispositivo siempre serán mejor los sistemas que funcionen con pulsos de reloj para tener un mayor control en la respuesta, es decir más precisión, ya que el resultado dependerá de los niveles altos o bajos o de los flancos de subida o bajada.

Otra forma en la que se puede observar como el diseño digital funciona para mejorar la precisión es en la elección de los FF, como se estudiará más adelante se verá que el FF JK es el que se ajusta mejor a la mayoría de las situaciones ya que ninguna de las entradas puede provocar un estado indefinido, solo

se menciona el concepto, en el correspondiente apartado podrá realizar el análisis respectivo.

Estos aparatos también cuentan con circuitos integrados como memorias RAM y ROM para poder almacenar información, según lo requiera la aplicación

Otra aplicación de los sistemas digitales en estos aparatos es cuando los decodificadores que contienen, sirven para controlar los dispositivos de salida a los cuales están conectados como pantallas, impresoras, bases de datos, teclados, etc.

Dentro de los aparatos citados anteriormente se llevan a cabo gran cantidad de operaciones binarias para poder codificar la información que lleva acabo una cadena de bits.

Estos aparatos están contruidos con circuitos integrados que pertenecen a las distintas familias como son TTL o CMOS clasificadas según el número de componentes que empleen en su construcción, sin duda alguna estos aparatos han sido contruidos con los nuevos avances tecnológicos para tener confiabilidad en los resultados para mejorar los procesos de diagnóstico y de ser de utilidad en la toma de decisiones de los médicos.

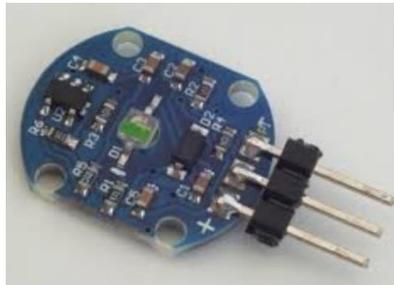


Figura 11 circuito interno de un oximetro de pulso

3. SISTEMAS NUMÉRICOS Y CÓDIGOS

3.1. SISTEMAS NUMÉRICOS

La electrónica digital; como la conocemos hoy día, no sería posible si no se tuviera un sistema numérico que permitiera la comunicación entre la máquina y las personas.

Este sistema lo conocemos como **SISTEMA NUMÉRICO BINARIO.**

El cual se basa en dos números 0 y 1 los cuales representan dos pulsos bajo y alto o dos estados apagado y encendido, respectivamente.

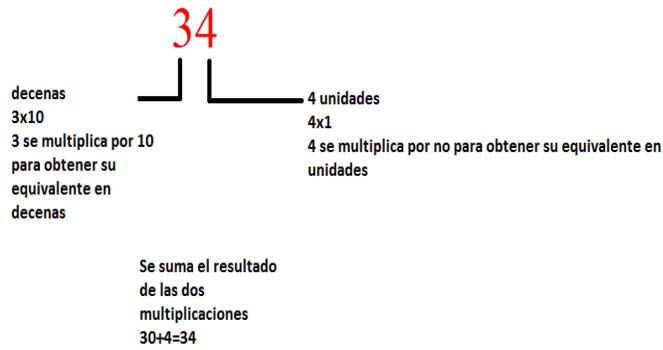
Para poder comprender como funciona el sistema numérico binario se dará primeramente una introducción al sistema numérico decimal.

El sistema numérico decimal se basa en 10 dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Es decir, en base a estos dígitos y tomando en consideración que los sistemas numéricos son posicionales podemos construir la representación de cualquier cantidad.

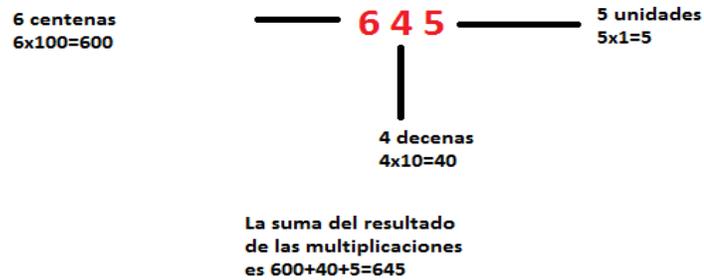
Ejemplo:

Sea el siguiente número:

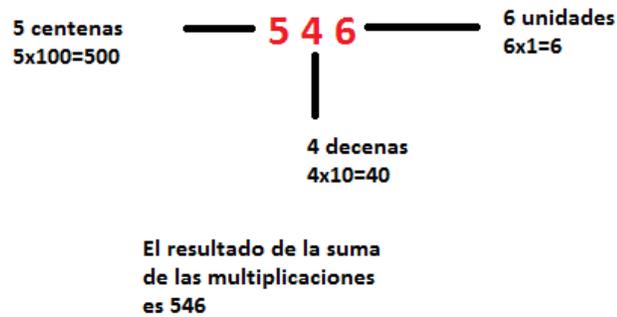


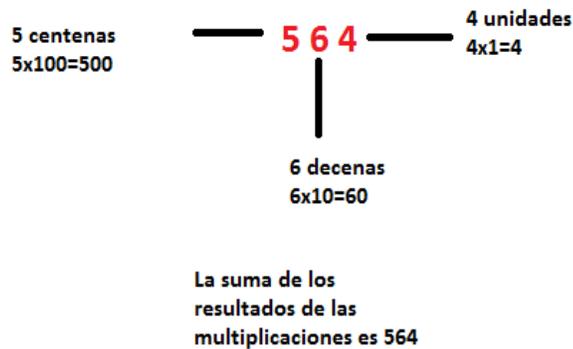
Analice los siguientes ejercicios y encuentre el patrón de comportamiento.

Ejemplo 1



Ejemplo 2



Ejemplo 3

Análisis del proceso.

La base del sistema es 10 por dos razones principalmente, la primera de ellas es que podemos representar cualquier número utilizando 0,1,2,3,4,5,6,7,8,9 como podrás haber comprobado.

Y la segunda para obtener el valor equivalente utilizamos potencias de 10

Para las unidades es $10^0 = 1$

Para las decenas es $10^1 = 10$

Para las centenas es $10^2 = 100$

Y así sucesivamente

Podemos llegar a la siguiente ecuación 10^n , donde n representa un número natural entero que asigna un peso al valor correspondiente de la cantidad.

.1	Decimas
.01	Centésimas
.001	Milésimas
.0001	diez milésimas
.00001	mil milésimas
.000001	un micro

Tabla 1 Números decimales

En base al siguiente ejemplo:

21

$$(2x10^1) + (1x10^0) = (20) + (1) = 21$$

Desarrolle los siguientes ejercicios

- a) 778
- b) 5699

Solución a los ejemplos

$$(7x10^2) + (7x10^1) + (8x10^0) = (7x100) + 70 + 8 = 700 + 70 + 8 = 778$$

$$\begin{aligned} \text{a) } (5x10^3) + (6x10^2) + (9x10^1) + (9x10^0) &= \\ (5x1000) + (6x100) + (9x10) + (9x1) &= \\ 5000 + 600 + 90 + 9 &= 5699 \end{aligned}$$

Sabemos que los números decimales se construyen multiplicando un dígito por el número 10 elevado a una potencia la cual depende del número de dígitos que tenga la cantidad.

En caso de que vayamos a utilizar números decimales se multiplica por el número 10 elevado a una potencia negativa.

$$\begin{array}{l} .33 \quad \text{---} \quad 3x10^{-2} \\ | \\ 3x10^{-1} \end{array}$$

$$(3x10^{-1}) + (3x10^{-2}) =$$

$$\begin{aligned}(3 \times 0.1) + (3 \times 0.01) &= \\ .3 + 0.03 &= \\ &= .33\end{aligned}$$

Nota: Entre más a la derecha se encuentre un número decimal más pequeño es su valor.

Revise el siguiente ejemplo:

54.18

Desarrollo

$$\begin{aligned}(5 \times 10^1) + (4 \times 10^0) + (1 \times 10^{-1}) + (8 \times 10^{-2}) &= \\ (5 \times 10) + (4 \times 1) + (1 \times 0.1) + (8 \times 0.01) &= 50 + 4 + 0.1 + 0.08 \\ &= 54.18\end{aligned}$$

EL SISTEMA BINARIO

El sistema binario maneja dos dígitos que son el 0 y 1, con esos dos números es suficiente para hacer que los dispositivos como las computadoras, teléfonos en fin unas gamas de sistemas electrónicos digitales funcionen.

En dicho sistema hay que tener presente la siguiente regla:

$$2^n - 1$$

Que indicara el número hasta el cual podremos contar donde n representa el número de bits que emplearemos en la cantidad.

Por ejemplo, para 2 bits tendremos:

$$2^2 - 1 = 4 - 1 = 3$$

Podremos representar hasta el número 3 en decimal.

Por el momento no se preocupe si no queda claro el cuadro.

Decimal	Binario
0	00
1	01
2	10
3	11

Tabla 2 Números 3 en decimal y binario

Con tres bits

$$2^n - 1 = 2^3 - 1 = 7$$

Decimal	Binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Tabla 3 Número del 0-7 en decimal y binario

CONVIERTIENDO FRACCIONES DECIMALES A BINARIO.

Como podrá observar el número decimal se obtiene como resultado de realizar las fracciones que aparecen debajo de estas. Por ejemplo 0.5 se obtiene de realizar la fracción $1/2$, 0.25 se obtiene realizar la fracción $1/4$.

$$\begin{array}{cccc}
 \mathbf{0.5} & \mathbf{0.25} & \mathbf{0.125} & \mathbf{0.0625} \\
 \mathbf{\frac{1}{2}} & \mathbf{\frac{1}{4}} & \mathbf{\frac{1}{8}} & \mathbf{\frac{1}{16}}
 \end{array}$$

Verifique el siguiente patrón

$$\begin{array}{ccccc}
 \mathbf{\frac{0.5}{2}} & \mathbf{\frac{0.25}{4}} & \mathbf{\frac{0.125}{8}} & \mathbf{\frac{0.0625}{16}} & \mathbf{\frac{0.03125}{32}}
 \end{array}$$

Analice la siguiente sucesión de números.

0.5 0.25 0.125 0.0625 0.03125

Ahora obtenga el resultado de las siguientes fracciones

$$\frac{1}{2} \quad \frac{1}{4} \quad \frac{1}{8} \quad \frac{1}{16} \quad \frac{1}{32}$$

A continuación, se propone una ecuación que muestra cómo se obtendría la siguiente fracción de la serie....

Solución:

$$\frac{1}{2^n}, \text{ Donde } n \text{ pertenece a los naturales, por lo tanto } n=1,2,3,4,5,N$$

Como podrá apreciar esta serie cumple con el patrón de números que se necesita en la serie, un pequeño análisis muestra que el numerador siempre es uno y que el denominador siempre es dos elevado a la n (dado que esta parte nos dará 2, 4, 8, 16, etc.) y al llevar acabo la división se obtendrá el respectivo número en la serie.

Convierta los siguientes números binarios a decimales:

$$\begin{aligned} .111 &= \left(1x\frac{1}{2}\right) + \left(1x\frac{1}{4}\right) + \left(1x\frac{1}{8}\right) \\ &= (1x0.5) + (1x0.25) + (1x0.125) = \end{aligned}$$

$$0.5 + 0.25 + 0.125 = 0.875$$

$$.0011 = \left(0x\frac{1}{2}\right) + \left(0x\frac{1}{4}\right) + \left(1x\frac{1}{8}\right) + \left(1x\frac{1}{16}\right) =$$

$$0 + 0 + 0.125 + 0.0625 = 0.1875$$

$$0.0110 = \left(0x\frac{1}{2}\right) + \left(1x\frac{1}{4}\right) + \left(1x\frac{1}{8}\right) + \left(0x\frac{1}{16}\right) =$$

$$0 + 0.25 + 0.125 + 0 = 0.375$$

Ahora como convertir un número fraccionario decimal a binario.

Se multiplica la cantidad sucesivamente por dos; hasta que la parte fraccionaria, sea solo de ceros o hasta que se alcance el número buscado de lugares decimales.

Ejercicio: Convertir el número binario .11010 en decimal

$$.11010 = \left(1x\frac{1}{2}\right) + \left(1x\frac{1}{4}\right) + \left(0x\frac{1}{8}\right) + \left(1x\frac{1}{16}\right) + \left(0x\frac{1}{32}\right) =$$

$$= 0.5 + 0.25 + 0.0625 = 0.8125$$

Ejercicio: convertir el número decimal .8125 en binario

Desarrollo:

0.8125

Primero se multiplica la parte fraccionaria por dos

$$0.8125 \times 2 = 1.6250$$

Del resultado se toma la parte fraccionaria y se multiplica por dos

$$0.6250 \times 2 = 1.2500$$

Del resultado se toma la parte fraccionaria y se multiplica por dos

$$.2500 \times 2 = 0.5000$$

Del resultado se toma la parte fraccionaria y se multiplica por dos

$$.5000 \times 2 = 1.0000$$

Cuando se obtiene ceros en la parte decimal se deja de efectuar las multiplicaciones

La parte entera de este:

Representa al bit MSB)

Respuesta el número sería: $0.8125 = .1101$

Ejercicio. Si el número binario es $.0101$ obtenga el número decimal

$$\begin{aligned} .0101 &= \left(0x\frac{1}{2}\right) + \left(1x\frac{1}{4}\right) + \left(0x\frac{1}{8}\right) + \left(1x\frac{1}{16}\right) = \\ &= 0.25 + 0.0625 = 0.3125 \end{aligned}$$

Comprobación: Dado el número anterior obtenga el correspondiente número binario:

Desarrollo:

$$\begin{array}{ll} 0.3125 & 0.3125 \times 2 = 0.6250 \\ & 0.6250 \times 2 = 1.2504 \\ & 0.2504 \times 2 = 0.5008 \\ & 0.5008 \times 2 = 1.0016 \end{array}$$

Tomando la parte entera de los resultados y tomando en consideración que la parte entera del último número es el MSB se obtiene:

$$0.3125 = 0.0101$$

Ejemplo: Dado el número binario $.11101$ obtenga el número decimal.

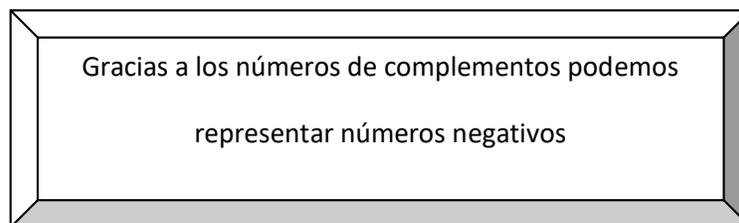
$$\begin{aligned} .11101 & \quad \left(1x\frac{1}{2}\right) + \left(1x\frac{1}{4}\right) + \left(1x\frac{1}{8}\right) + \left(0x\frac{1}{16}\right) + \left(1x\frac{1}{32}\right) = \\ & \quad (0.5) + (0.25) + (0.125) + (0.03125) = 0.90625 \end{aligned}$$

Comprobación: Obtener el número binario del número decimal 0.90625

Desarrollo: $0.90625 \times 2 = 1.81250$
 $.8125 \times 2 = 1.6250$
 $.6250 \times 2 = 1.2500$
 $.2500 \times 2 = 0.5000$
 $.5000 \times 2 = 1$

Nota: Como podemos ver el proceso termina. Cuando todos son ceros.

COMPLEMENTOS DE 1'S Y 2' DE NÚMEROS BINARIOS



COMPLEMENTO A 1's

Para obtener el complemento a 1's de un número binario se cambian los 0 por 1 y los 1 por 0.

Ejemplo: Obtenga el complemento 1's de los siguientes números:

Sea el número	10101101
El complemento del número es:	01010010
Sea el número	111000101
El complemento del número es:	000111010
Sea el número	110010110
El complemento del número es	001101001

A continuación, en los siguientes ejercicios obtenga el complemento a 2's de los números binarios.

Para obtener el complemento 2's de un número; primero se obtiene el complemento 1's y luego se suma un 1 al bit menos significativo LSB.

Ejemplo:

$$\begin{array}{r}
 1010101 \\
 0101010 \\
 + \quad \quad 1 \\
 \hline
 0101011
 \end{array}$$

Ejemplo:

$$\begin{array}{r}
 10101011 \\
 01010100 \\
 + \quad \quad 1 \\
 \hline
 01010101
 \end{array}$$

Ejemplo:

$$\begin{array}{r}
 10100100 \\
 01011011 \\
 + \quad \quad 1 \\
 \hline
 01011100
 \end{array}$$

Ejemplo:

$$\begin{array}{r}
 1010100 \\
 0101011 \\
 + \quad \quad 1 \\
 \hline
 0101100
 \end{array}$$

COMO SE REPRESENTAN LOS NÚMEROS AFECTADOS POR SIGNO

Cuando trabajas con las computadoras, habrás notado que puedes manejar números negativos, a continuación, descubrirá como se hacen estas operaciones.

El signo es aquel que indica si un número es positivo o negativo y la magnitud indica el valor en base a al sistema posicional.

En un número con signo el bit del extremo izquierdo denominado como MSB, indica sí es positivo, si es 0, en el caso de ser 1 es lo contrario, los demás bits indican la magnitud.

NÚMEROS OCTALES

El sistema numérico octal se forma con los siguientes 8 dígitos:

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

Si necesitamos seguir la sucesión será:

11, 12, 13, 14, 15, 16, 17

La siguiente sucesión será:

20, 21, 22, 23, 24, 25, 26, 27

Los subíndices indican el valor de la raíz con la cual se trabaja es decir la base, por ejemplo, para el sistema octal el subíndice es un 8 y un 10 como subíndice para los números decimales como se muestra a continuación.

12_8 Para los números octales.

12_{10} Para los números decimales.

¿Cómo convertir un número octal a decimal?

La base del sistema octal son 8 dígitos.

Los cuales son:

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

Por lo tanto, se emplearán las potencias de 8 para llegar a convertir un número octal a decimal.

A continuación, analice los siguientes ejemplos:

$$\begin{aligned} 321_8 &= (3 \times 8^2) + (2 \times 8^1) + (1 \times 8^0) \\ &= (3 \times 64) + (16) + (1) \\ &= 192 + 16 + 1 \\ &= 209_{10} \end{aligned}$$

Para que verifique los resultados convierta el número decimal a octal con divisiones sucesivas. Es decir, tome el número decimal y divídalo sucesivamente entre 8.

$$\frac{209}{8} = 26 \quad \text{Residuo } 1$$

$$\frac{26}{8} = 3 \quad \text{Residuo } 2$$

$$\frac{3}{8} = 0 \quad \text{Residuo } 3 \text{ este último dígito es el MSB}$$

Finalmente se obtiene el número 321

Convierta el número 674 octal a decimal.

$$674_8 \rightarrow 384_{10}$$

Nota: Analice con cuidado el ejercicio.

$$(6 \times 8^2) + (7 \times 8^1) + (4 \times 8^0) =$$

$$(6 \times 64) + (56) + (4) = 384_{10}$$

Verifique su resultado convirtiendo el número decimal 384 en octal.

$$\frac{384}{8} = 48 \quad \text{Residuo } 0$$

$$\frac{48}{8} = 6 \quad \text{Residuo } 0$$

$$\frac{6}{8} = 0.75 \quad \text{Residuo } 0$$

Nota: Como podrá ver el número 384 no tiene representación en el sistema octal, ya que como podrá recordar los dígitos con los cuales trabaja dicho sistema son:

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

Siguiente ejercicio.

$$663_8 \rightarrow 435_{10}$$

$$(6 \times 8^2) + (6 \times 8^1) + (3 \times 8^0) =$$

$$(6 \times 64) + (48) + (3) =$$

$$384 + 48 + 3 = 435$$

Verifique su resultado.

Solución

El resultado lo puede verificar de la siguiente manera.

$\frac{435}{8} = 54$	Residuo 3
$\frac{54}{8} = 6$	6
$\frac{6}{8} = 0$	6 Donde este es el MSB

Finalmente se obtiene el resultado (leyendo de abajo hacia arriba los datos)

$$663_8$$

¿Cómo convertir un número octal a binario?

Analice los siguientes ejemplos:

$$24_8$$

Se toma cada dígito por separado y el número se convierte al equivalente en binario:

2	4
010	100

Entonces el número binario será:

$$010100$$

A continuación, analice los siguientes ejercicios:

$$36_8$$

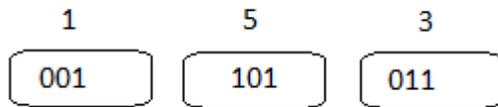
3	6
011	110

El número binario equivalente será:

$$011110$$

Siguiente ejercicio

$$153_8$$



El número binario equivalente será:

001101011

Sea el siguiente ejercicio.

4310_8



El número binario equivalente será:

100011001000

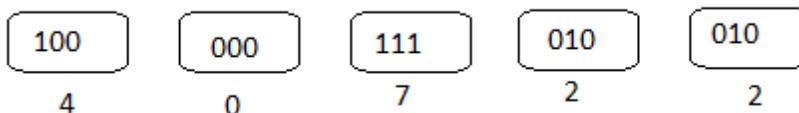
CONVERSION DE NÚMEROS BINARIOS A OCTALES

Se toma el número binario y se van haciendo grupos de 3 bits de derecha a izquierda; si hacen falta dígitos para a completar el grupo de 3 se agregan ceros a la izquierda.

Analice los siguientes ejemplos:

100 000 111 010 010

Realizando los grupos de 3



Como podrás notar se formará el siguiente número octal:

40722_8

Sea el siguiente número binario:

000 110 110 111

Realizando los grupos de tres:



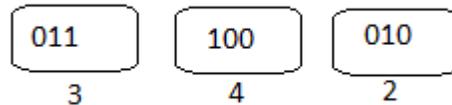
Se formará el siguiente número octal:

664_8

Sea el siguiente número binario convertir a octal.

011 100 010

Se realizan los grupos de tres.



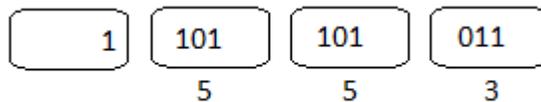
Por lo tanto, se formará el siguiente número octal:

342_8

Sea el siguiente número binario convertir a octal:

1 101 101 011

Primero se realizan los grupos de tres:



Como podrás notar hay un grupo que hace falta a completar con ceros:



Como resultado se formará el siguiente número octal:

1553_8

NÚMEROS HEXADECIMALES

El sistema numérico hexadecimal se basa en 10 dígitos y 6 letras, que son:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D,

E, F

Analice la siguiente tabla.

Hexadecimal	Decimal	Binario
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Tabla 4 Números del 0-15 en hexadecimal, decimal y binario

La tabla muestra que cada dígito hexadecimal se representa con un número binario de cuatro bits.

Ejemplo: Construya la sucesión de números del 0 hasta el:

FF_{16}

Solución:

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,10,11,12,13,14,15,16,17,18,19,1A,1B,1C,1D,1E,1F,20,21,22,23,24,25,26,27,28,29,2A,2B,2C,2D,2E,2F,30,31,32,33,34,35,36,37,38,39,3A,3B,3C,3D,3E,3F,40,41,42,43,44,45,46,47,48,49,4A,4B,4C,4D,4E,4F,50,51,52,53,54,55,56,57,58,59,5A,5B,5C,5D,5E,5F,60,61,62,63,64,65,66,67,68,69,6A,6B,6C,6D,6E,6F,70,71,72,73,74,75,76,77,78,79,7A,7B,7C,7D,7E,7F,80,81,82,83,84,85,86,87,88,89,8A,8B,8C,8D,8E,8F,90,91,92,93,94,95,96,97,98,99,9A,9B,9C,9D,9E,9F,A0,A1,A2,A3,A4,A5,A6,A7,A8,A9,AA,AB,AC,AD,AE,AF,B0,B1,B2,B3,B4,B5,B6,B7,B8,B9,BA,BB,BC,BD,BE,BF,C0,C1,C2,C3,C4,C5,C6,C7,C8,C9,CA,CB,CC,CD,CE,CF,D0,D1,D2,D3,D4,D5,D6,D7,D8,D9,DA,DB,DC,DD,DE,DF,E0,E1,E2,E3,E4,E5,E6,E7,E8,E9,EA,EB,EC,ED,EE,EF,F0,F1,F2,F3,F4,F5,F6,F7,F8,F9,FA,FB,FC,FD,FE,FF.

¿A qué número es equivalente el número FF en decimal?

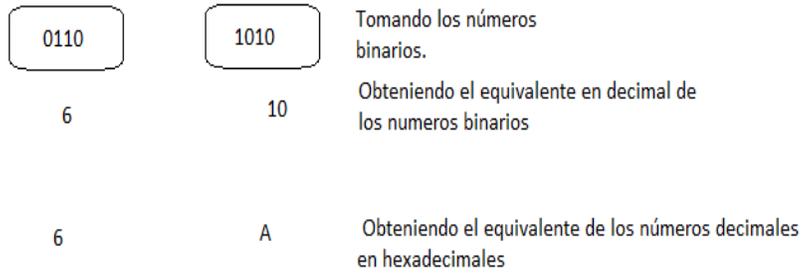
$$\begin{aligned}
 FF_{16} &= (15 \times 16^1) + (15 \times 16^0) \\
 &= (15 \times 16) + (15 \times 1) \\
 &= 240 + 15 \\
 &= 255_{10}
 \end{aligned}$$

¿CÓMO CONVERTIR UN NÚMERO BINARIO A HEXADECIMAL?

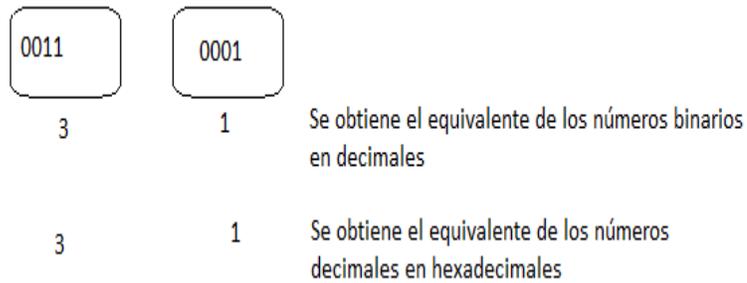
Se divide el número binario en grupos de 4 bits. Comienza en el extremo del lado derecho y después en base al valor que tengan se les asigna un número o una letra en función del formato hexadecimal

Ejercicio. Analice los siguientes ejemplos:

Sea el número binario 01101010 convertirlo a hexadecimal

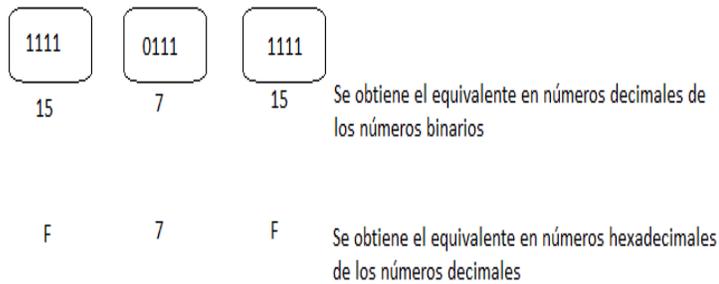


Sea el siguiente número binario 00110001 convertirlo a hexadecimal.



Finalmente el número equivalente en hexadecimal del número binario dado es: 31_{16}

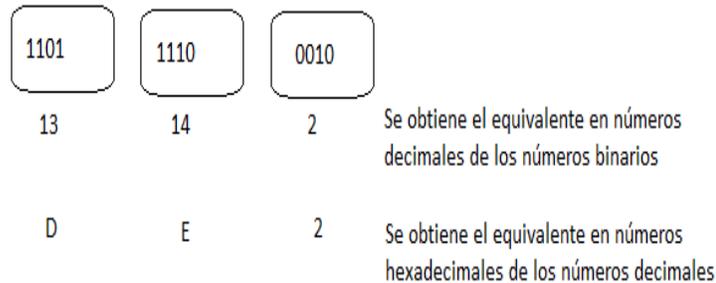
Sea el siguiente número binario 111101111111 convertirlo a hexadecimal.



Finalmente, el número hexadecimal equivalente será:

$$F7F_{16}$$

Sea el siguiente número binario 110111100010 convertirlo a hexadecimal.



Finalmente, el número equivalente en el sistema hexadecimal del número binario es:

$DE2_{16}$

¿CÓMO SE CONVIERTE UN NÚMERO HEXADECIMAL A BINARIO?

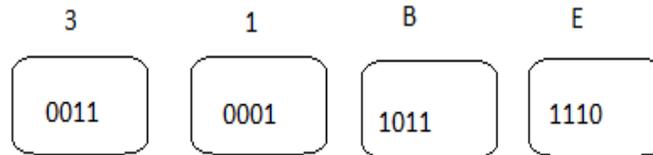
Cada letra o número de los que forman la cantidad hexadecimal se convierte a un número binario de 4 dígitos, al final se juntan todos los dígitos que hay para formar una sola cantidad.

A continuación, analice los siguientes ejemplos:

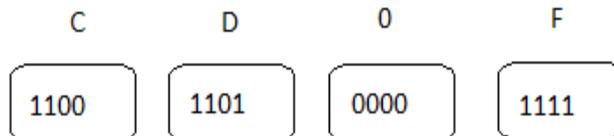


La cifra final a la que se
llega es

0010010011111010

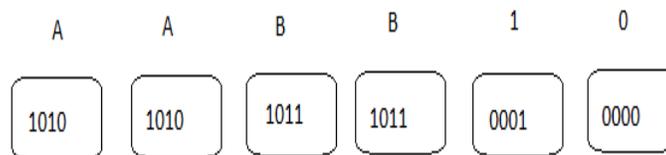


Finalmente, el número será: 0011000110111110



Finalmente, el número al cual se llega es:

1100110100001111



Finalmente, al número al que se llegaría sería:

1010101011101100010000

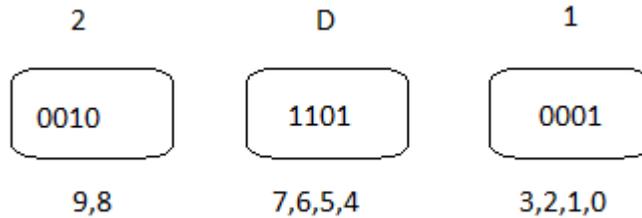


Finalmente, el número al que se llega es:

1100110111111010

¿CÓMO SE CONVIERTE UN NÚMERO HEXADECIMAL A DECIMAL?

Para ello se convierte primero el número hexadecimal a binario y del número binario se obtiene el equivalente a decimal, tomando en cuenta la i -ésima posición.



Donde tenemos:

$$2^9, 2^7, 2^6, 2^4, 2^0$$

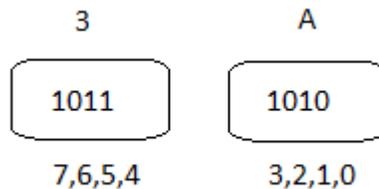
Si obtenemos la suma de estas potencias:

$$512 + (128+64+16) + 1 = 512 + 208 + 1 = 721$$

Respuesta el número $2D1_{16}$, equivale a 721_{10}

Convierta el siguiente hexadecimal a decimal.

$$3A_{16}$$



$$2^7, 2^5, 2^4, 2^3, 2^1$$

Desarrollando las sumas de las potencias anteriores

$$(128+32+16+8+2) =$$

$$176+10=$$

$$186$$

Respuesta el número $3A_{16}$ equivale a 186_{10}

Existe una segunda forma de realizar la conversión y es la siguiente:

Se toma cada letra y cada dígito del número hexadecimal, se obtiene el equivalente en número decimal y se multiplica por la base 16 elevada a una potencia que va de acuerdo a la posición.

Analice los siguientes ejercicios se recomienda que primero intente hacerlos por su cuenta y revise el resultado con el del manual.

Ejercicio sea el número 3B convertir el número a su equivalente en decimal.

Desarrollo:

$$3B_{16}$$

$$(3x16^1) + (11x16^0) =$$

$$(3x16) + (11x1) = 48+11=59$$

Ejercicio sea el número C4 en hexadecimal obtener su equivalente en números decimales

$$C4_{16}$$

$$(12x16^1) + (4x16^0) =$$

$$(12x16) + (4x1) = 192+4=196$$

Ejercicio sea el número 5F9 en hexadecimal obtener su equivalente en números decimales

$$5F9_{16}$$

$$(5x16^2) + (15x16^1) + (9x16^0) =$$

$$(5x256) + (240) + (9) =$$

$$1280+240+9=1529$$

Ejercicio sea el número FFA0₁₆ en hexadecimal obtener su equivalente en números decimales

$$FFA0_{16}$$

$$(15x16^3) + (15x16^2) + (10x16^1) + (0x16^0) =$$

$$61440+3840+160+0=65440$$

Resultado el número decimal equivalente es 65440₁₀

Ejercicio sea el número $DE51_{16}$ en hexadecimal obtener su equivalente en números decimales

$$(13 \times 16^3) + (14 \times 16^2) + (5 \times 16^1) + (5 \times 16^0) =$$

$$53248 + 3584 + 80 + 5 = 56917$$

¿Cómo convertir un número decimal a hexadecimal?

Se lleva a cabo una serie de divisiones sucesivas y se va tomando el residuo como el número hexadecimal, tomando en cuenta que el primero de ellos es el LSB.

$$\frac{436}{16} = 27.25$$

$$\frac{27}{16} = 1.6875$$

$$\frac{1}{16} = 0.0625$$

1 B 4

Verificación

$$\begin{array}{rcccc} & 1 & B & 4 & \\ & 1 & 11 & 4 & \\ (1 \times 16^2) & + & (11 \times 16^1) & + & (4 \times 16^0) = \\ & 256 & + & 176 & + & 4 & = \end{array}$$

436_{10}

4. CODIGOS

4.1 CODIGO GRAY

La característica principal de este código es que cuando se tienen dos números sucesivos, estos se diferencian entre sí por que se presenta un cambio de valor en un bit.

Analice el siguiente ejemplo:

Convierta de binario a código Gray

10001 número binario

11001 números Gray

Metodología

10001

Se toma el bit MSB (más significativo), del número binario, ese será el primer número del número en código gray.

Después sume el bit MSB y el adyacente a este, del número binario este será el número siguiente del código Gray.

Para los casos en los que se presente un acarreo, no tome en cuenta este último y solo tome el valor de la suma, por ejemplo, si tiene $1+1=0$ tendrá un acarreo de 1, el bit del código Gray, será cero y el 1 no se tomará en cuenta.

O también es equivalente decir que se está aplicando la operación XOR entre dígitos.

Después sume el siguiente bit con el que le es adyacente, continúe sucesivamente hasta haber sumado todos los bits del número binario por pares.

Nota: El número de bits del número binario debe coincidir con el número de bits del número en código Gray.

CONVERSIÓN DE CÓDIGO GRAY A BINARIO.

Analice el siguiente ejemplo para convertir un número de binario a Gray.

Convierta el siguiente número en código Gray a número binario.

110101 números en Gray

100110 número binario

Para convertir un número de código Gray a Binario, se toma el bit MSB del código Gray el cuál será el bit MSB del número binario a continuación se realiza una suma de bits en diagonal, es decir la primera suma del bit MSB del código binario con el bit adyacente del bit MSB del código Gray.

5. OPERACIONES ARITMETICAS Y BINARIAS

5.1 OPERACIONES ARITMÉTICAS

Las operaciones aritméticas son con las que ha trabajado en niveles académicos anteriores, en donde empleo la suma, la resta, la multiplicación y la división y en las cuales para llevar a cabo dichas operaciones solamente empleo el sistema decimal.

En base al cuadro anterior se muestran unos ejemplos para

que pueda visualizar los conceptos.

El empleo de la suma es las operaciones a las cuales se recurre con más frecuencia.

Las operaciones aritméticas se basan en la raíz 10, es decir, en los 10 números que constituyen esta base que son: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Las operaciones aritméticas son la suma, la resta, la multiplicación y la división.

ADICIÓN ARITMETICA

$$8+2=10$$

$$1+1=2$$

$$1+0=1$$

$$0+0=0$$

$$3+5=8$$

SUSTRACCIÓN ARITMETICA

$$8-3=5$$

$$1-1=0$$

$$2-1=1$$

$$3-2=1$$

$$10-8=2$$

MULTIPLICACIÓN ARITMETICA

$$3 \times 2 = 6$$

$$7 \times 4 = 28$$

$$11 \times 3 = 33$$

$$5 \times 7 = 35$$

DIVISIÓN ARITMETICA

$$\frac{10}{2} = 5$$

$$\frac{20}{3} = 6.66$$

$$\frac{25}{4} = 6.25$$

5.2 OPERACIONES BINARIAS

Para poder comprender como funcionan los sistemas digitales, es indispensable saber las operaciones básicas que se llevan a cabo, las cuales son: Suma, Resta, Multiplicación y División

ADICIÓN BINARIA

Los principios básicos para la adición de dígitos binarios son:

$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

$$1+1=10 \quad \text{suma de cero}$$

con acarreo de un 1

Sumar los números binarios

Nota: Para que pueda utilizar los resultados es necesario que convierta los números binarios a decimales.

a)

$$\begin{array}{r} 10010 \\ + 11001 \\ \hline 101011 \end{array}$$

$$\begin{array}{r} 18 \\ + 25 \\ \hline 43 \end{array}$$

b)

$$\begin{array}{r} 101011 \\ + 001101 \\ \hline 111000 \end{array}$$

$$\begin{array}{r} 32+8+3= 43 \\ 8+4+1= 13 \\ \hline 56 \end{array}$$

La transformación a base decimal se lleva acabo obteniendo la equivalencia del número correspondiente, estas equivalencias se obtienen con la posición que ocupan los cero y unos en el número.

Por ejemplo, en el caso anterior:

Se tiene:

101011

Tomando en cuenta que se tiene la siguiente sucesión en base 2 elevado a un exponente para obtener el equivalente en número decimal. En este caso solo se ha elevado hasta la potencia 5 ya que solo se necesitan 6 posiciones para obtener el equivalente en número decimal.

$$2^5 2^4 2^3 2^2 2^1 2^0$$

De la sucesión anterior se desprende que:

$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	0	1	1

Se analiza en donde se encuentran los números unos y se obtiene su equivalente en número decimal por ejemplo los unos se encuentran en:

$$2^5 \quad 2^3 \quad 2^1 \quad 2^0$$

Para obtener el equivalente en número decimal se suma el resultado de elevar el dos a las potencias indicadas.

De donde se obtiene:

$$32+8+2+1=43$$

En número 43 es el número equivalente en número decimal

c)

$$\begin{array}{r}
 1011011 \\
 + \quad 1011 \\
 \hline
 1100110
 \end{array}
 \qquad
 \begin{array}{r}
 64+16+8+2+1=91 \\
 \hline
 8+3=11 \\
 \hline
 =102
 \end{array}$$

Comprobación
 $64+32+4+2=102$

SUSTRACCIÓN BINARIA

Los principios básicos para restar los dígitos binarios son:

$$0-0=0$$

$$1-1=0$$

$$1-0=1$$

$$10-1=1 \qquad 0-1 \text{ y tiene}$$

un préstamo 1

Este préstamo se representa con uno en resultado como se muestra a continuación:

$$\begin{array}{r}
 \text{1 préstamo} \\
 \underline{10} \\
 - \quad 1 \\
 \hline
 \mathbf{01}
 \end{array}$$

a)

$$\begin{array}{r}
 1001 \quad 9 \\
 - \quad 101 \quad - 5 \\
 \hline
 0100 \quad 4
 \end{array}$$

b)

$$\begin{array}{r}
 100110 \quad 38 \\
 - \quad 1011 \quad -11 \\
 \hline
 101101 \quad 27
 \end{array}$$

$$\begin{array}{r}
 101010 \quad 42 \\
 - \quad 1110 \quad -14 \\
 \hline
 011100 \quad 28
 \end{array}$$

d)

$$\begin{array}{r}
 101101 \quad 45 \\
 - \quad 1110 \quad -14 \\
 \hline
 011111 \quad 31
 \end{array}$$

e)

$$\begin{array}{r}
 10110110 \quad 182 \\
 - \quad 10101 \quad -21 \\
 \hline
 10100001 \quad 161
 \end{array}$$

MULTIPLICACIÓN BINARIA

Las reglas para la multiplicación binaria se muestran a continuación:

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

La multiplicación binaria es un procedimiento, muy similar al de la multiplicación decimal.

La diferencia de trabajar con una operación lógica y una aritmética se basa en la raíz, es decir las operaciones aritméticas trabajan con la base 10, y las operaciones lógicas trabajan solo con dos números 1 y 0.

A continuación, se muestra el siguiente ejercicio.

a)

$$\begin{array}{r} 110 \\ \times 11 \\ \hline 110 \\ 110 \\ \hline 10010 \end{array}$$

$$6 \times 3 = 18$$

Nota: si convertimos el resultado e valor obtenido es 18

b)

$$\begin{array}{r} 10110 \\ \times 110 \\ \hline 00000 \\ 10110 \\ 10110 \\ \hline 10000100 \end{array}$$

$$22 \times 6 = 132$$

Nota: el resultado que obtenemos en binario es el número 132 en decimal

c)

$$\begin{array}{r} 11011 \\ \times 111 \\ \hline 11011 \\ 11011 \\ 11011 \\ \hline 10111101 \end{array} \qquad \begin{array}{r} 27 \\ \times 7 \\ \hline 189 \end{array}$$

DIVISIÓN BINARIA

Analice los siguientes ejercicios:

$$111 \div 10$$

$$\begin{array}{r}
 11.1 \\
 10 \overline{) 111} \\
 \underline{011} \\
 010 \\
 \underline{0}
 \end{array}$$

$$11.1 = \left(1x\frac{1}{2}\right) + (1x2^0) + (1x2^1)$$

La igualdad anterior se desprende de lo siguiente:

El número uno de la parte decimal se multiplica por $\frac{1}{2}$, como recordará $\frac{1}{2}$ es equivalente a

$$1/2^0 = \frac{1}{2}$$

Posteriormente se toma la parte entera y se toma en cuenta el valor que toma cada uno en base a la posición que tiene, por ejemplo al primero uno se multiplica por 2^0 ; el segundo uno se multiplica por 2^1 obteniendo con ello la igualdad que se presenta.

$$\begin{aligned}
 & (0.5) + (1) + (2) \\
 & = 0.5 + 3 \\
 & = 3.5
 \end{aligned}$$

Como podrá notar; la división Binaria es similar a como se realiza en el sistema Decimal, revise los siguientes ejemplos y podrá comprobarlo.

$$11101 \div 100 \quad 29 \div 4 = 7.25$$

$$\begin{array}{r}
 111.01 \\
 100 \overline{) 11101} \\
 \underline{0110} \\
 01001 \\
 \underline{00100} \\
 000
 \end{array}$$

Verificación

$$\begin{aligned}
 111.01 &= (1x2^0) + (1x2^1) + (1x2^2) + \left(0x\frac{1}{2}\right) + \left(1x\frac{1}{4}\right) = \\
 &= (1x1) + (2) + (4) + 0 + (0.25) \\
 &= 1+2+4+0.25 \\
 &= 7.25
 \end{aligned}$$

El sistema binario es un ejemplo de sistema ponderado, es decir a cada número le corresponde un peso determinado en base a la posición que ocupa. El sistema binario es de base dos, porque solo emplea dos valores 0 o 1. La posición que ocupe el 0 el 1 le corresponde un determinado valor que se relaciona con la base elevada a una potencia, esta potencia se determina elevando el dos a la primera potencia, segunda potencia o tercera potencia según corresponda como se muestra en la siguiente ecuación.

$$2^n \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$$

Ejemplo:

Sea el siguiente número binario:

$$1 \quad 0 \quad 1 \quad 0$$

Empleando la teoría vista anteriormente el equivalente en decimal de este número será:

$$(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) = 10$$

Otro ejemplo de sistema posicional es el sistema decimal, esta emplea 10 dígitos como base estos son: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Cada dígito tiene un valor en base a su posición.

Por ejemplo, el número 22 representa al número veintidós que es dos decenas y dos unidades.

Es decir, para representar un dígito se recurre a las unidades, para representar dos dígitos se recurre a las decenas, para representar tres dígitos se recurre a las centenas y así sucesivamente.

Con lo anterior se aprecia por que el sistema decimal es posicional porque cada dígito en un número tiene un peso, de unidad, decena, centena o el correspondiente.

Ejercicio

$$11110 \div 101$$

$$\begin{array}{r} 110 \\ 101 \overline{) 11110} \\ \underline{0101} \\ 0000 \\ \underline{0} \end{array}$$

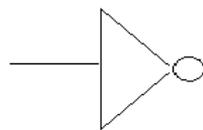
Verificación:

$30 \div 5 = 6$ En el resultado se obtiene 110 lo que equivale al número 6 en decimal

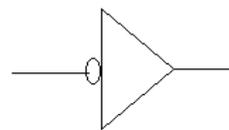
6. COMPUERTAS LÓGICAS

6.1 EL INVERSOR

El inversor se puede representar mediante dos símbolos lógicos estándar.



a) Símbolo 1



b) Símbolo 2

Figura 12 Símbolo lógico para la compuerta not o negada

La función del inversor es obtener el nivel opuesto al de la entrada. Como podrá verse en los símbolos lógicos solamente se tiene una entrada y una salida.

Por ejemplo, si tenemos una entrada en alto, la salida será cero

Si tenemos una entrada en bajo la salida será alto

Si viéramos lo anterior con un foco tendríamos:



Figura 13 Representación de la compuerta not con el estado encendido y apagado de un foco

Es muy importante hacer notar que la operación del inversor, no cambia debido a la colocación de la polaridad o indicador de negación el cual se representa con un círculo.

A continuación, se muestra la tabla de verdad del inversor.

Entrada	Salida
Alto(1)	Bajo(0)
Bajo(0)	Alto(1)

Tabla 5 Estados de una compuerta not

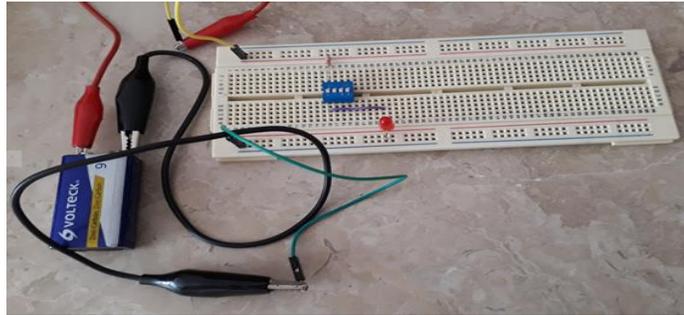


Figura 14 Circuito físico de la compuerta NOT en estado apagado o bajo

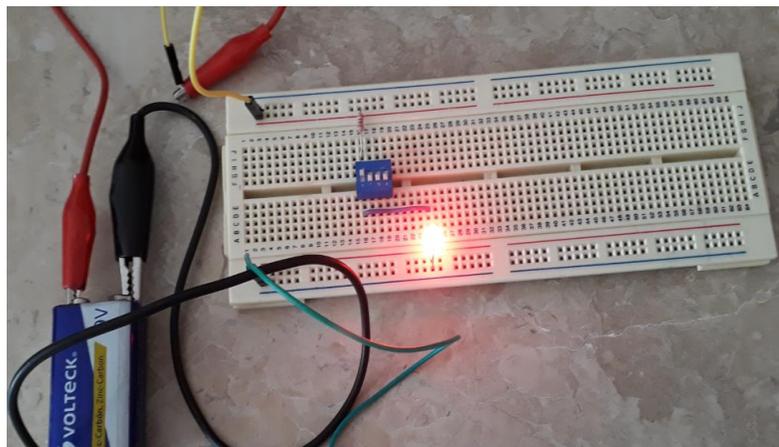


Figura 15 Circuito físico de la compuerta NOT en estado encendido o alto

6.2 LA COMPUERTA AND

La función de la compuerta AND es llevar a cabo la multiplicación lógica. La compuerta AND puede tener dos o más entradas y tiene solamente una salida.

Como habremos notado las compuertas llevan a cabo operaciones lógicas básicas.

SIMBOLOS PARA LA COMPUERTA AND

Figura 16 Símbolo lógico para una compuerta AND de dos entradas

Como podemos apreciar la compuerta AND puede tener más de dos entradas, en este caso se muestran compuertas AND con dos entradas.

A continuación, se muestra la tabla de verdad de la compuerta AND de 2 entradas.

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

Tabla 6 Tabla de verdad de la compuerta AND de 2 entradas

Como podrás observar en la tabla anterior las salidas son ceros cuando una o ambas entradas son ceros y la salida es 1 cuando todas las entradas son 1.

Por lo tanto, una regla matemática es que aplique la multiplicación recordando que cualquier número multiplicado por cero es cero.

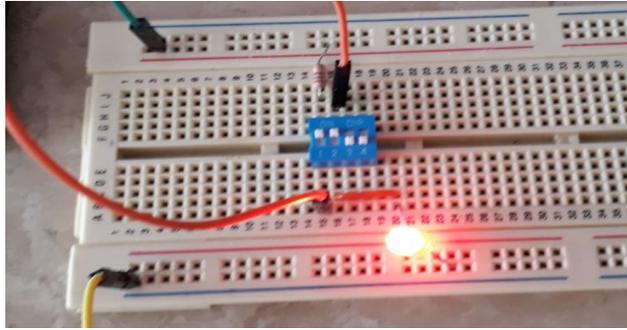


Figura 17 Circuito físico de la compuerta AND cuando los dos estados son altos

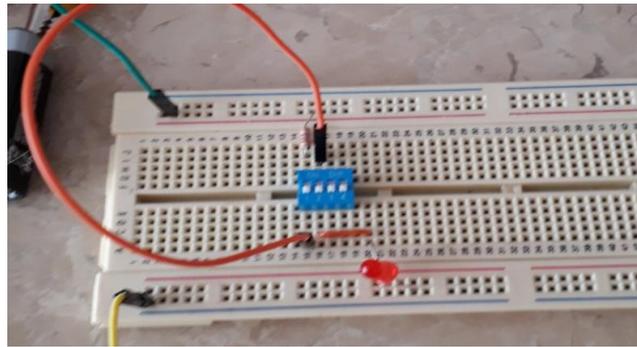


Figura 18 Circuito físico de la compuerta AND cuando los dos estados están en bajo

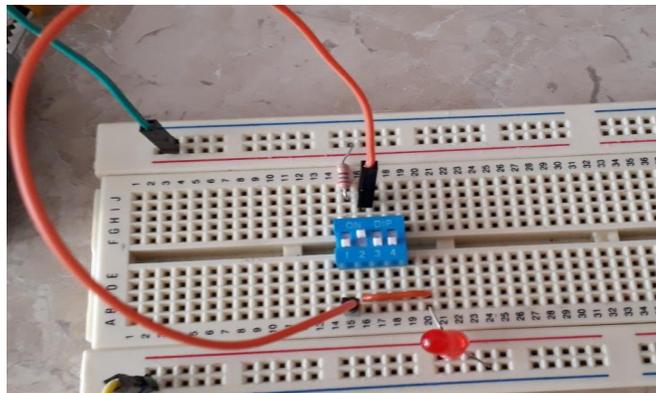


Figura 19 Circuito físico de la compuerta AND cuando un estado es bajo y el otro alto

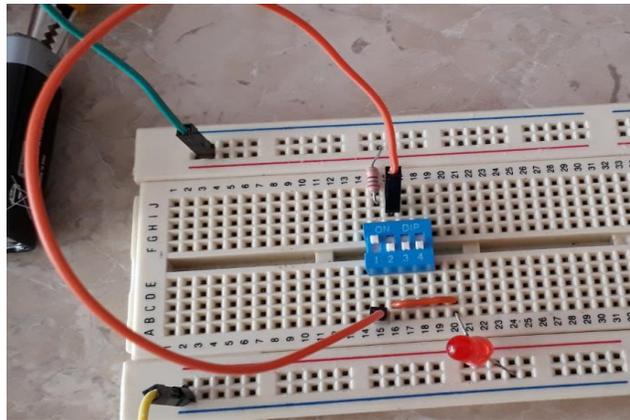
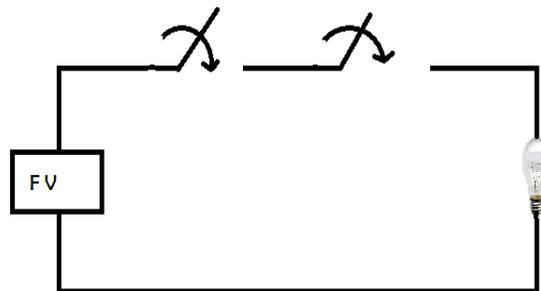


Figura 20 Circuito físico de la compuerta AND cuando un estado es alto y otro bajo

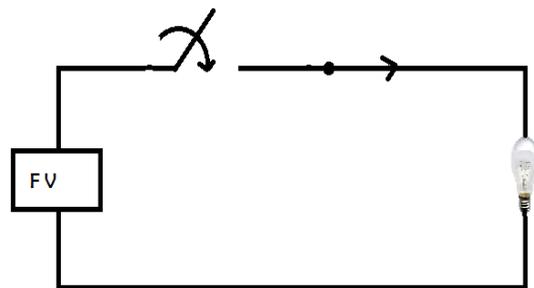
Diagrama de circuito lógico de la compuerta AND

La compuerta AND se puede ver como un circuito en serie.

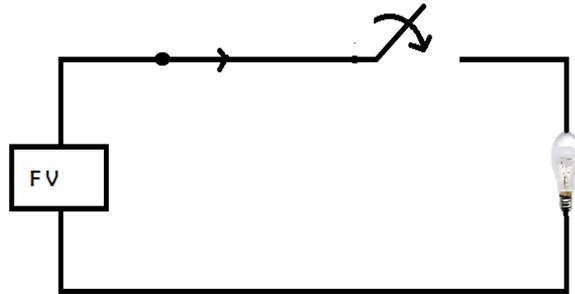
Cuando se tiene a las entradas dos estados bajos o apagados el foco no se enciende lo que equivale a tener los interruptores abiertos.



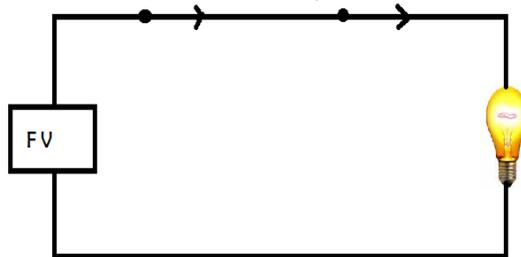
Cuando una entrada es cero y la otra uno o visto de otra manera un bajo o un alto, equivale a tener un switch abierto y otro cerrado



Cuando se tiene un 1 en la entrada y un cero o visto como un alto y un bajo lo que equivale a verlo como un switch cerrado y otro abierto el foco del circuito está pagado



Cuando se tienen dos unos a la entrada o dos altos, se puede ver de manera equivalente con los dos switch cerrados lo cual da como resultado que el foco se encienda.



Con la teoría anterior se puede construir tablas de verdad, con el número de entradas que se desee (matemáticamente) y obtener el resultado de forma confiable y rápida, como se podrá imaginar para el mundo físico; el número de entradas es limitado.

Para encontrar el número total de combinaciones posibles en base al número de variables de entrada aplicamos la siguiente fórmula.

$$N = 2^n$$

Donde N es el número de combinaciones posibles y n es el número de variables de entrada.

Ejemplo: Para 3 variables construir la tabla de verdad de la compuerta AND

$$N = 2^3 = 8$$

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

Tabla 7 Tabla de verdad de la compuerta AND para tres variables

Otra forma de emplear la compuerta AND es aplicándole formas de onda a la entrada.

Para el álgebra booleana la notación para la compuerta AND que tiene dos entradas se usa la siguiente notación:

$$X=AB$$

En caso de que sean tres la variable de entrada se tiene:

$$X=ABC$$

Es decir, solo se escriben las variables de forma adyacente para indicar la función AND.

6.3 LA COMPUERTA OR

Al igual que en la compuerta AND, en la compuerta OR se pueden tener dos o más variables en la entrada, teniendo un solo resultado a la salida.

La salida OR lleva a cabo la adición lógica.

EL SIMBOLO DE FORMA DISTINTIVA PARA LA COMPUERTA OR ES:



Figura 21 Símbolo lógico para una compuerta OR de dos entradas

A continuación, se muestra la tabla de verdad de la compuerta OR de dos entradas.

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

Tabla 8 Tabla de verdad para la compuerta OR de dos entradas

Como podrá observar la salida es baja cuando todas las entradas son bajas.

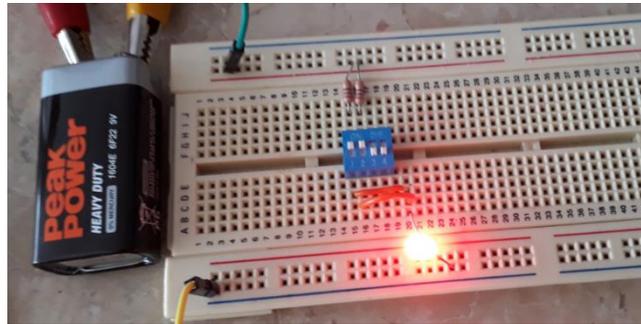


Figura 22 Circuito físico de la compuerta OR cuando los dos estados están en alto

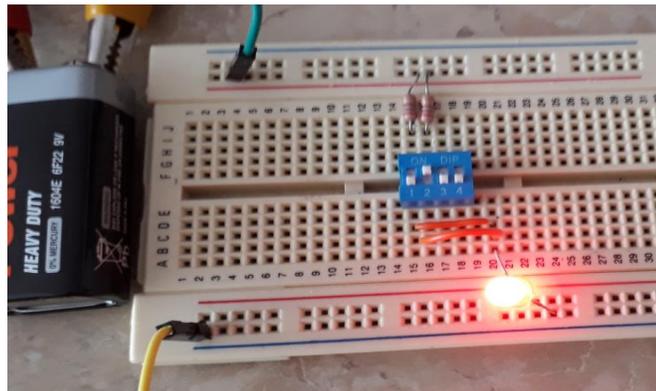


Figura 23 Circuito físico de la compuerta OR cuando un estado está en bajo y otro en alto

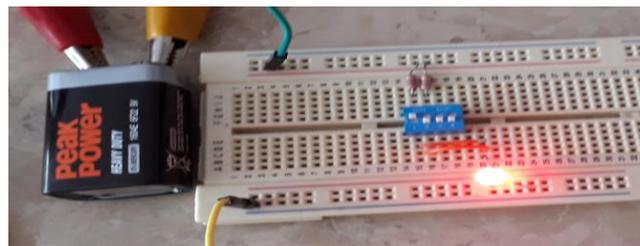


Figura 24 Circuito físico de la compuerta OR cuando una de las entradas esta en alto y la otra en bajo

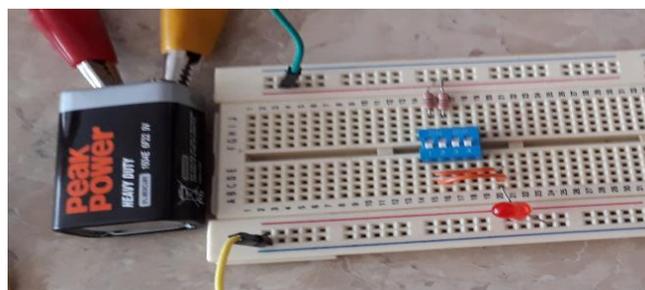
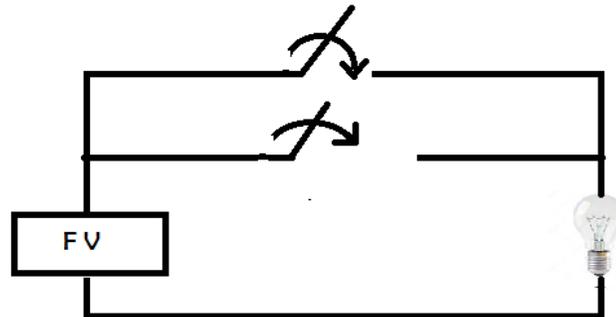


Figura 25 Circuito físico de la compuerta OR cuando las dos entradas están en bajo.

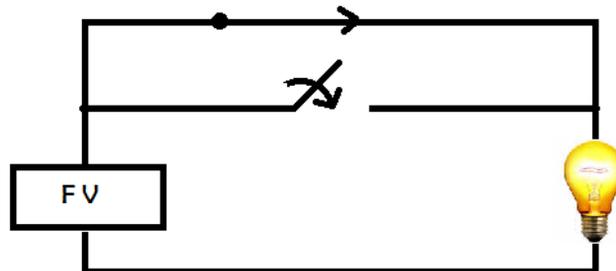
Diagrama de circuito OR con dos entradas

La compuerta OR se puede alambrear como un circuito en paralelo, el análisis se realizará a continuación.

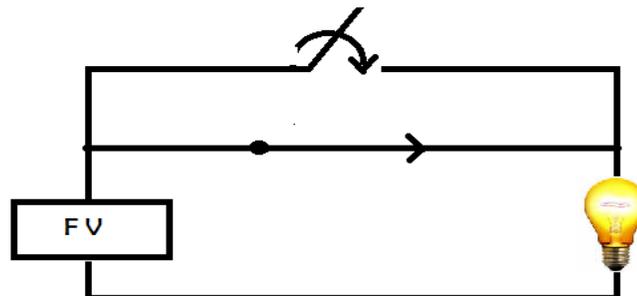
Si consideramos que la primera entrada son dos ceros, visto también como dos estados bajos esto sería equivalente a ver dos switch abiertos.



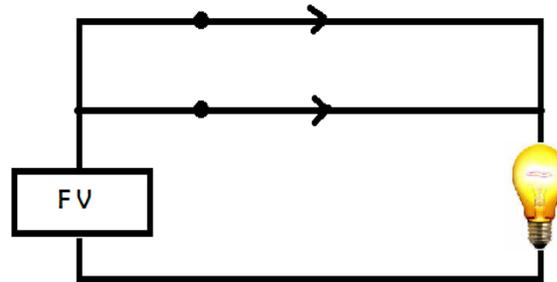
Se tiene en la entrada un uno y un cero lo que se puede ver como como un alto o un bajo, lo que es equivalente que se tenga un switch cerrado y un abierto lo que da como resultado que el foco se prenda.



Si se tiene a la entrada un cero y un uno que es lo mismo que tener un estado bajo y un alto esto equivale a tener un switch abierto y uno cerrado en el circuito provocando que el foco se encienda.



Si en la entrada se tienen dos unos o lo que es lo mismo dos estados altos equivale en el circuito a tener los dos switch cerrados lo que da por resultado que el foco se encienda.



A continuación, se analiza un ejemplo en donde se tienen 3 entradas para ver cómo es la tabla de verdad

Para 3 entradas la tabla es la siguiente:

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

Tabla 9 Tabla de verdad para la compuerta OR de tres entradas

Para una función lógica de dos variables con la compuerta OR se representa de la siguiente manera.

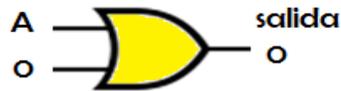
$$X = A + B$$

Para tres variables será:

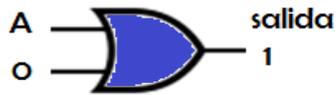
$$X = A + B + C$$

Donde por cada variable se agrega un símbolo + perteneciente a la suma lógica.

NOTA: Análisis de A+0



Si $A=0$, la salida será cero



Si $A=1$ la salida es igual a 1

Si una variable de entrada es alta en una compuerta OR y las otras son ceros las salidas serán siempre 1. Si todas las variables de entrada de una compuerta OR son bajas o iguales a cero la salida será siempre cero

Dentro del algebra siguen 4 reglas que se muestran a continuación:

$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

$$1+1=1$$

Es muy importante destacar que la adición booleana difiere de la adición binaria en el siguiente caso:

$$1+1=10$$

Para la adición Booleana (lógica)

Para la adición binaria

$$1+1=1$$

$1+1=0$ y llevamos un 1 de acarreo

6.4 LA COMPUERTA NAND

La importancia de la compuerta NAND radica en que podemos realizar cualquier operación con ella.

Es decir, podemos utilizarla como una compuerta AND, OR, NOT.

A continuación, se muestra el símbolo lógico estándar de una compuerta NAND.



Figura 26 Símbolo lógico para una compuerta NAND de dos entradas

Como puede notar en la imagen anterior se muestra la compuerta tiene 2 entradas respectivamente.

Cabe recalcar que la compuerta NAND es una combinación de la compuerta AND y la compuerta NOT, en base a lo anterior podemos llegar a la conclusión de que un camino para llegar a los resultados de aplicar dicha compuerta es primero realizar la operación AND y negar el resultado.

Ejercicio:

Verifique el resultado de las siguientes tablas de verdad.

Para una función de dos variables.

Entrada	Entrada	Salida	Salida	Salida
A	B	AND	NOT	NAND
0	0	0	1	1
0	1	0	1	1
1	0	0	1	1
1	1	1	0	0

Tabla 10 Tabla de verdad para la compuerta NAND para dos variables

Para una función de 3 variables.

Aplicamos la fórmula:

$$N = 2^3 = 8$$

Se tienen 8 posibles combinaciones

Entrada	Entrada	Entrada	Salida	Salida	Salida
A	B	C	AND	NOT	NAND
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	0	1	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	0	0

Tabla 11 Tabla de verdad para la compuerta NAND de tres variables

En base a las dos ondas de entrada A y B aplique la compuerta NAND y obtenga la función de salida.

La expresión de salida para la compuerta NAND es:

$$X = \overline{AB}$$

Para el caso de tres variables la expresión sería:

$$X = \overline{ABC}$$

6.5 LA COMPUERTA NOR

La importancia de la compuerta OR radica en lo siguiente:

Por ejemplo, cuando se desea armar un circuito para representar la función correspondiente, el cual consta de llevar operaciones que realizarían las compuertas NOT, OR y AND, estas compuertas pueden ser sustituidas por compuertas NOR realizando las conexiones pertinentes.

El símbolo estándar de la compuerta NOR se muestra a continuación, en donde también se puede apreciar que puede tener 2 entradas.



Figura 27 Compuerta NOR con dos entradas

El funcionamiento de la compuerta puede verlo como la aplicación de la compuerta OR seguida de la compuerta NOT.

Tabla de verdad para una función NOR de 2 variables.

Entrada	Entrada	Salida	Salida	Salida
A	B	OR	NOT	X(NOR)
0	0	0	1	1
0	1	1	0	0
1	0	1	0	0
1	1	1	0	0

Tabla 12 Tabla de verdad para la compuerta NOR de 2 variables

Entrada	Entrada	Entrada	Salida	Salida	Salida
A	B	C	OR	NOT	X(NOR)
0	0	0	0	1	1
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	1	0	0

Tabla 13 Tabla de verdad para la compuerta NOR de tres variables

La compuerta NAND operando como un OR con entradas negadas

Analícemos las tablas de verdad para ambas compuertas.

TABLA A

Entrada	Entrada	Salida
A	B	X(NAND)
0	0	1
0	1	1
1	0	1
1	1	0

Tabla 14 Tabla de verdad para la compuerta NAND de dos entradas

Entrada	Entrada	Salida
A	B	X(OR)
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 15 Tabla de verdad de la compuerta OR de 2 entradas

TABLA (OR NEGADA)

\overline{A}	\overline{B}	X(OR NEGADA)
1	1	1
1	0	1
0	1	1
0	0	0

Tabla 16 Tabla de verdad de la compuerta OR negada para dos entradas

Entrada	Entrada	Salida
A	B	X(NOR)
0	0	1
0	1	0
1	0	0
1	1	0

Tabla 17 Tabla de verdad para la compuerta NOR de dos entradas

Entrada	Entrada	Salida
A	B	X(AND)
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 18 Tabla de verdad para la compuerta AND de dos entradas

Entrada	Entrada	Entrada	Entrada	Salida
A	B	\overline{A}	\overline{B}	X(AND negativa)
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Tabla 19 Tabla de verdad para la compuerta AND negada

Como podemos ver la tabla A y la tabla C tienen resultados equivalentes por lo tanto podemos decir que la compuerta NOR es equivalente a la compuerta AND con entradas negadas

El símbolo estándar para la compuerta AND negativa es la que se muestra en las siguientes figuras donde varía el número de entradas en la compuerta es decir se muestra con 2 entradas.



Figura 28 Símbolo lógico para la compuerta AND negada con dos entradas respectivamente

La expresión lógica para indicar una compuerta NOR de dos variables es:

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

Para el caso de tres variables

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

Donde por cada variable que se agregue se acompaña del símbolo + y se extiende la barra (o testa)

NOTA: Observe lo siguiente....

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

Una compuerta NOR es equivalente a una compuerta AND con entradas negadas, en este caso el número de entradas es dos.

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

Una compuerta NAND es equivalente a una compuerta OR con entradas negadas, en este caso el número de entradas es dos. Es muy importante mencionar que en los casos anteriores se aplicó el Teorema De Morgan para poder llegar a la equivalencia.

6.6 LA COMPUERTA OR EXCLUSIVA

Esta compuerta generalmente se conoce con el nombre de XOR.

El símbolo de la compuerta es:



Figura 29 compuerta XOR

La tabla de verdad de la función XOR es:

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

Tabla 20 Tabla de verdad para la compuerta XOR de dos entradas

La compuerta XOR obedece a las siguientes reglas.

Las salidas de la compuerta XOR serán altas solo cuando las entradas sean opuestas, si son iguales la salida será baja.

6.7 LA COMPUERTA NOR EXCLUSIVA

Cabe destacar un punto importante es la compuerta XOR y XNOR solo tienen dos entradas.

Su símbolo es el siguiente:



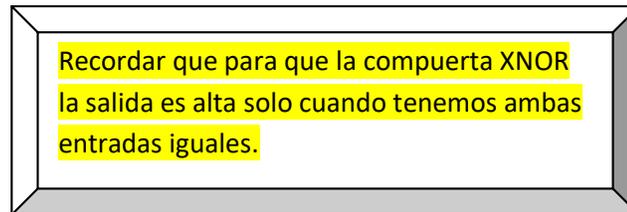
Figura 30 Símbolo lógico de la compuerta XNOR con dos entradas

Como podrá apreciar la burbuja indica que su salida es contraria a la de la compuerta XOR

La tabla de verdad para la compuerta XNOR es la siguiente:

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

Tabla 21 Tabla de verdad para la compuerta XNOR de dos entradas



7. ALGEBRA BOOLEANA (EXPRESIONES ALGEBRAICAS)

En el álgebra booleana las variables se designan por una letra.

Por regla general las variables suelen designarse con las últimas letras del abecedario estos son:

X, Y, Z

En el caso de un inversor el resultado se muestra como el complemento de la variable.

Ejemplo; si la variable es Y después de aplicar la operación del inversor se obtiene: \overline{Y}

La operación de un inversor se conoce como (circuito NOT).

El álgebra booleana es una parte fundamental de las matemáticas en la cual se basan los circuitos digitales.

En el álgebra booleana tenemos lo que se conoce como variables. Por ser álgebra booleana cada variable solo puede tener dos valores 0 o 1.

Cuando hacemos referencia al complemento de una función estamos indicando que es el valor contrario al cual tiene la variable.

En este caso para indicar el complemento se utiliza una barra sobre la variable.

En una forma más general podemos decir que una literal es un variable complementada o sin complementar.

Un ejemplo de lo anterior se muestra a continuación:

$$X = 0$$

$$\overline{X} = 1$$

Las reglas básicas de la adición Booleana son:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

Efectuar una suma en el álgebra booleana es como aplicar la compuerta OR a las variables

Recordemos la tabla de verdad de la compuerta OR.

Entrada X	Entrada Y	Salida(OR) Z
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 22 Tabla de verdad de la compuerta OR para dos variables de entrada

Y ahora veremos algunos ejemplos

$$\overline{X+Y} = 0$$

Si $X=1, Y=1$

$$X+Y+\overline{Z} = 1$$

$$2^3 = 8$$

X	Y	Z	\overline{Z}	S
0	0	0	1	1
0	0	1	0	0
0	1	0	1	1
0	1	1	0	1
1	0	0	1	1
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

Tabla 23 Tabla de verdad para la ecuación propuesta

Como se puede apreciar en la tabla pasada el resultado indica un bajo cuando:

$$X=0, Y=0 \text{ y } Z=0$$

Para los demás casos la salida es un alto

Como ejercicio se deja que sustituya el valor de las variables de la tabla de verdad en la ecuación y compruebe el resultado.

Ejercicio: Se tiene la siguiente ecuación

$$X + \overline{Y} + \overline{Z} = 1$$

En base a ello determine que valores adoptan las variables X normal y (Y, Z) negadas para obtener una alta a la salida.

$$X + \overline{Y} + \overline{Z} = 1$$

X	Y	Z	\overline{Y}	\overline{Z}	X	\overline{Y}	\overline{Z}	S
0	0	0	1	1	0	1	1	1
0	0	1	1	0	0	1	0	1
0	1	0	0	1	0	0	1	1
0	1	1	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1
1	0	1	1	0	1	1	0	1
1	1	0	0	1	1	0	1	1
1	1	1	0	0	1	0	0	1

Tabla 24 Tabla de verdad de la ecuación vista

Respuesta.

X	Y	Z	\overline{Y}	\overline{Z}	X	\overline{Y}	\overline{Z}	S
0	0	0	1	1	0	1	1	1
0	0	1	1	0	0	1	0	1
0	1	0	0	1	0	0	1	1
1	0	0	1	1	1	1	1	1
1	0	1	1	0	1	1	0	1
1	1	0	0	1	1	0	1	1
1	1	1	0	0	1	0	0	1

Tabla 25 Tabla de verdad con la respuesta del ejemplo anterior

En páginas anteriores se había dicho que las reglas básicas para la multiplicación binaria son:

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

Y que la operación de estas era el equivalente de aplicar la compuerta AND a las entradas.

Una multiplicación booleana podemos expresarla con dos literales juntas.

Ejemplo de ello son las siguientes expresiones:

$$XY$$

$$\bar{Y}\bar{Y}Z$$

$$\bar{X}Y\bar{Z}$$

Sea la siguiente ecuación:

$$\bar{X}Y\bar{Z} = 1$$

Determine para que valores de las variables X, Y, Z la salida es un alto.

$$\bar{X}Y\bar{Z} = 1$$

X	Y	Z	\bar{X}	\bar{Z}	Y	S
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	1	0	1	1	1	1
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	1	0	0	0	0
1	1	0	0	1	1	0
1	1	1	0	0	1	0

Tabla 26 Tabla de verdad para la ecuación vista

Respuesta: Los valores que hacen que las salidas sean un alto son:

$$X=0 \quad Y=1 \quad \text{y} \quad Z=0$$

Sustituya el valor de las variables obtenidas en la tabla de verdad y compruebe los resultados

7.1 LEYES Y PROPIEDADES EN EL ALGEBRA BOOLENA

A continuación, se presentan leyes y reglas básicas para llevar a cabo las operaciones.

Tenemos tres leyes principales:

LEY CONMUTATIVA

$$X + Y = Y + X$$

$$XY = YX$$

LEY ASOCIATIVA

Para tres variables la ley asociativa será:

$$X + (Y + Z) = (X + Y) + Z$$

$$X (YZ) = (XY) Z$$

LEY DISTRIBUTIVA

Se tienen tres variables; observe como se aplica la ley distributiva:

$$X (Y + Z) = XY + XZ$$

Reglas para llevar a cabo operaciones en el álgebra booleana

$$A + 0 = A$$

$$A + 1 = 1$$

$$A \times 0 = 0$$

$$A \times 1 = A$$

$$A + A = A$$

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

$$A \times A = A$$

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

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

$$A + AB = A$$

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

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

Probando la ecuación 11

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

Desarrollando el lado izquierdo

Aplicando la propiedad 10

La Ley asociativa

La propiedad 6

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

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

$$A + (1)B =$$

$$A + B$$

Demostrando la regla 12

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

$$AA+AC+AB+BC=$$

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

$$A+AC+BC=$$

$$(1+C)A+BC=$$

$$1(A)+BC=$$

$$A+BC$$

Se aplicó la Ley Asociativa

Se aplicó la propiedad 10

Se aplicó la propiedad 4

7.2 TEOREMAS DE MORGAN

Morgan establece dos teoremas fundamentales para el álgebra booleana.

Los dos teoremas son:

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

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

Estas dos leyes se derivan de manera eléctrica

La primera Ley se puede visualizar como la aplicación de una compuerta NAND

La segunda Ley se puede visualizar como la aplicación de una compuerta NOR

NOTA: Los teoremas De Morgan pueden aplicarse a más de dos variables.

Analice los siguientes ejercicios donde se aplican los teoremas De Morgan.

$$\overline{X+Y+Z}$$

Solución:

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

Ejercicio

$$\overline{X Y Z}$$

Solución

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

Ejercicio:

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

Simplificando la ecuación quedaría como:

$$X + Y + Z$$

Se recomienda que siempre realice los ejercicios por su cuenta y los compare con los del manual, para que vaya adquiriendo la habilidad.

Ejercicio:

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

Desarrollo:

$$[\overline{\overline{A} \overline{B} C}] [\overline{Z}] =$$

Se aplicó Teorema de Morgan

$$[\bar{A} \bar{B} C] [\bar{Z}] = \text{Se aplicó Teorema de Morgan}$$

$$\bar{A} \bar{B} C \bar{Z} \text{ Se empleó la multiplicación}$$

Ejercicio:

$$\overline{A B C + \bar{Z} + D Q}$$

Desarrollo:

$$\overline{A B C} [\bar{Z}] [\bar{D} \bar{Q}] = \text{Se aplicó Teorema de}$$

Morgan

$$[A B C][Z][\bar{D} + \bar{Q}] = \text{Se aplicó la regla 9 y Teorema}$$

de Morgan respectivamente

$$[A B C Z] [\bar{D} + \bar{Q}]$$

Ejercicio:

$$\overline{\bar{A} \bar{B} \bar{C}} [\bar{Z} + \bar{D}]$$

Desarrollo

$$\overline{A + B} [\bar{C}] [(\bar{Z})(\bar{D})] \text{ Se aplicó Teorema de}$$

Morgan

$$[\bar{A} + \bar{B}] [C][(\bar{Z})(D)] = \text{Se aplicó regla 9}$$

$$[\bar{A} + \bar{B}][C][\bar{Z} D]$$

Expresiones booleanas

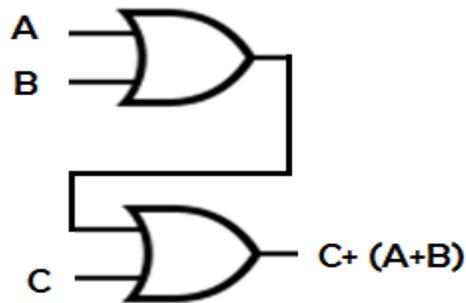


Figura 31 Implementación de la expresión lógica $C+(A+B)$ con compuertas lógicas

Se tienen 3 variables A, B y C por lo tanto se tendrá:

$$2^3 = 8$$

Su tabla de verdad es la siguiente:

A	B	C	S (OR)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Tabla 27 Tabla de verdad del ejemplo mencionado

Sea el siguiente ejemplo:

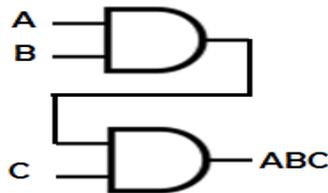


Figura 32 Implementación de la expresión lógica ABC con compuertas lógicas

Construyendo la tabla de verdad

A	B	C	S(AND)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Tabla 28 Tabla de verdad para las compuertas de la figura 22

Combinación de compuertas

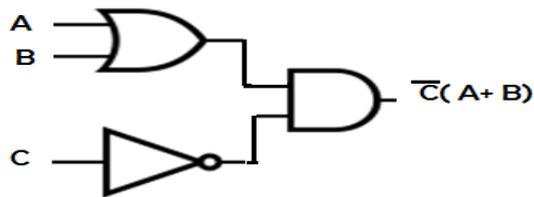


Figura 33 Implementación de la expresión lógica

Como podrá apreciar anteriormente hay tres casos para los cuales la salida es alta:

$$A=0 \quad B=1 \quad C=0 \quad S=1$$

$$A=1 \quad B=0 \quad C=0 \quad S=1$$

$$A=1 \quad B=1 \quad C=0 \quad S=1$$

Sea la tabla de verdad

A	B	C	\overline{C}	$\overline{C}(A+B)$
0	0	0	1	0
0	0	1	0	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	0	0

Tabla 29 Tabla de verdad para la función lógica planteada

Sea el siguiente circuito en el que se implementa la expresión lógica $D[\bar{C} + (\bar{A} + \bar{B})]$

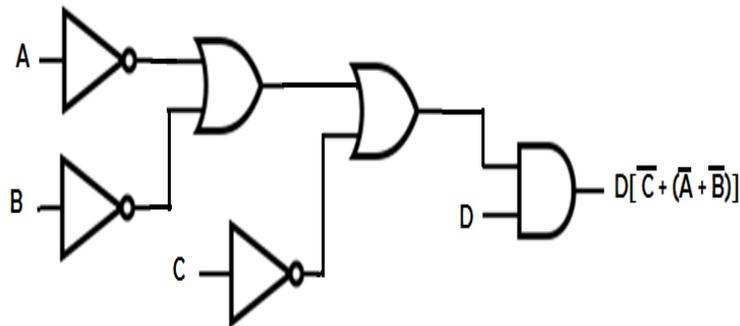
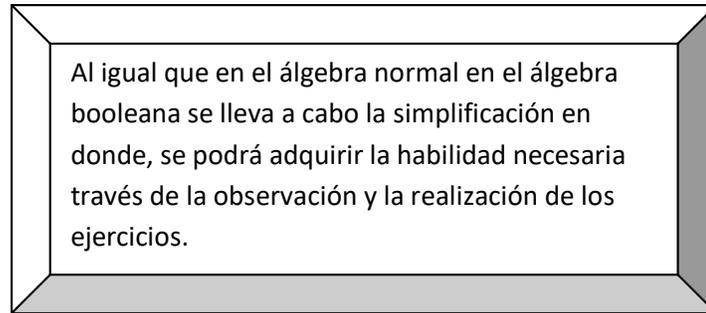


Figura 34 Implementación de la función lógica

A	B	C	D	\bar{A}	\bar{B}	\bar{C}	$D[\bar{C} + (\bar{A} + \bar{B})]$
0	0	0	0	1	1	1	0
0	0	0	1	1	1	1	1
0	0	1	0	1	1	0	0
0	0	1	1	1	1	0	0
0	1	0	0	1	0	1	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	0	0
0	1	1	1	1	0	0	1
1	0	0	0	0	1	1	0
1	0	0	1	0	1	1	1
1	0	1	0	0	1	0	0
1	0	1	1	0	1	0	1
1	1	0	0	0	0	1	0
1	1	0	1	0	0	1	1
1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	0

Tabla 30 Tabla de verdad para la función planteada

Nota: cómo podemos apreciar en la tabla anterior hay seis salidas cuyo valor es igual a uno.



Ejercicio: Lleve a cabo la simplificación de las siguientes expresiones booleanas.

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

Desarrollo:

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

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

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

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

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

$$\overline{\overline{A}B}[\overline{\overline{A}C\overline{D}}] + D$$

Desarrollo

$$[\overline{\overline{A}} + \overline{\overline{B}}][\overline{\overline{A}} + \overline{\overline{C}} + \overline{\overline{D}}] + D =$$

$$[A + B][A + C + D] + D =$$

$$\cancel{AA} + AC + AD + AB + BC + BD + D =$$

$$AC + AD + AB + BC + D$$

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

c)

$$AB(\bar{C} + \bar{D}) + \bar{D}(D + AB + \bar{C}) + AB\bar{D}$$

Desarrollo:

$$AB\bar{C} + AB\bar{D} + \bar{D}D + AB\bar{D} + \bar{D}\bar{C} + AB\bar{D} =$$

$$AB\bar{C} + AB\bar{D} + 0 =$$

$$AB(\bar{C} + \bar{D})$$

8. Universalidad de las compuertas NAND y NOR

Como introducción a este tema plantearemos el siguiente planteamiento:

Sabemos que un circuito combinacional puede implementarse con las compuertas AND, compuertas OR e inversores.

En esta sección aprenderemos que una compuerta NAND podemos emplearla como un inversor; por otra parte, es posible emplear combinaciones de la compuerta NAND para realizar operaciones AND y OR.

Un caso parecido lo podemos ver con la compuerta NOR la cual podemos emplear operaciones AND y OR y no solo eso sino también NAND.

8.1 LA COMPUERTA NAND

¿QUÉ SIGNIFICA QUE SEA UNA COMPUERTA SEA UNIVERSAL?

Se dice que una compuerta es universal cuando con ella podemos sustituir a las compuertas básicas como la NOT, AND y la OR.

Para este caso en particular la compuerta NAND es una compuerta universal.

Diagrama de una compuerta NAND



Figura 35 Símbolo lógico de una compuerta NAND

Implementación de una función NAND para que funcione como inversor.



Figura 36 Símbolo lógico de una compuerta NAND implementada como inversor

Se aplica la misma función negada

¿Cómo implementarías la compuerta NAND para que funcione como compuerta AND?

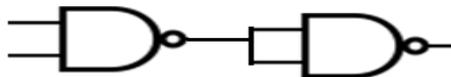


Figura 37 Diagrama de una compuerta NAND implementada para que funcione como una compuerta AND

Como se puede ver se aplica una compuerta NAND a la compuerta vista en la figura 26 y se obtiene como resultado lo siguiente.

$$\overline{\overline{AB}} = AB$$

¿Cómo implementarías las compuertas NAND para ser usadas como una compuerta OR?

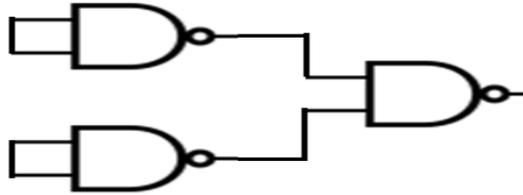


Figura 38 Diagrama de una compuerta NAND implementada como una compuerta OR

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

Analice el siguiente ejercicio:

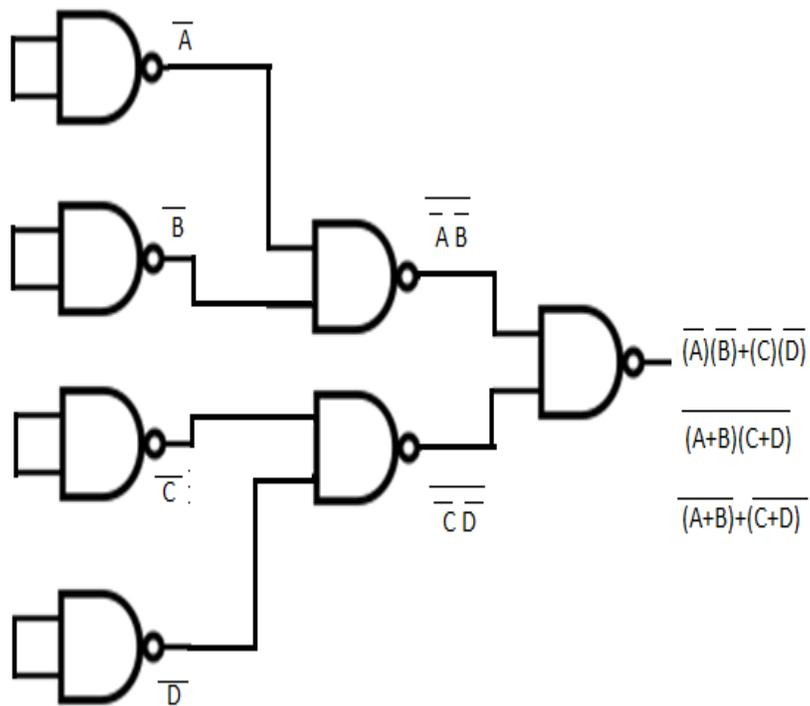


Figura 39 Diagrama de la aplicación de compuertas NAND y algebra booleana para obtener la función de salida del circuito

Como se puede apreciar se aplicaron los dos Teoremas de Morgan a la salida.

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

Al aplicar el Teorema De Morgan

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

Al aplicar nuevamente el Teorema De Morgan

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

En esta parte se aplica una doble inversión

Analice el siguiente ejemplo:

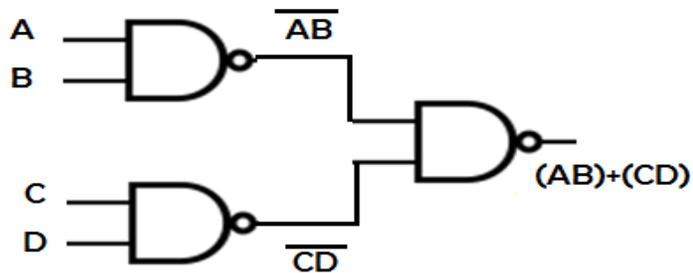


Figura 40 Diagrama de la aplicación de compuertas NAND y algebra booleana para obtener la función de salida del circuito

***Desarrollo

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

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

$$\overline{(\bar{A} \bar{B}) + (\bar{C} \bar{D})}$$

$$\overline{(\bar{A} \bar{B}) + (\bar{C} \bar{D})}$$

$$\overline{(\bar{A} \bar{B})} \overline{(\bar{C} \bar{D})}$$

$$\overline{\overline{(\bar{A} + B)}} \overline{\overline{(\bar{C} + D)}}$$

$$(A+B)(C+D)$$

Como podrá observar los circuitos lógicos combinacionales anteriores se han construido solamente con compuertas NAND y la función que realizan es equivalente a que, si se hubieran empleado compuertas AND, OR e inversores.

8.2 LA COMPUERTA NOR

La compuerta NOR se considera como una compuerta universal porque se pueden alambrear de tal forma que se pueden sustituir las compuertas básicas como la NOT, la NAND y la OR para la implementación de la función correspondiente.

En la sección anterior se dio una definición de lo que era una compuerta universal para el caso de la compuerta NAND la compuerta NOR también es un ejemplo de una compuerta universal.

La diferencia entre ambas es que una es una multiplicación negada y la otra es una suma negada, en donde

si aplicamos el Teorema de De Morgan a cada una de las partes resultantes encontraremos en el primer caso que la NAND da una suma de términos negados individualmente y la compuerta NOR después de aplicar el Teorema de De Morgan al resultado se obtiene una multiplicación de términos negados individualmente. Como los términos están negados individualmente no podemos decir que una compuerta sea la inversa de la otra más bien es un complemento

Bien como habíamos visto anteriormente el hecho de que se diga que una compuerta es universal implica el hecho de que puede implementarse para habilitar funciones como NOT, AND, OR y para el caso de la compuerta NOR poder implementar la función NAND.

Veamos algunos ejemplos:

Podemos tener una compuerta NOT, de la siguiente manera:

Puenteamos las entradas de la compuerta NOR



Figura 41 Aplicación de la compuerta NOR como una compuerta NOT

Vea el siguiente ejemplo en el que la compuerta NOR se aplica como una compuerta OR

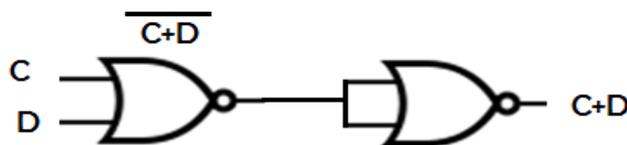


Figura 42 Aplicación de la compuerta NOR como una compuerta OR

Desarrollo:

$$\begin{aligned} & \overline{C+D} \\ & \overline{(C)(D)} \\ & \overline{\overline{(C)(D)}} \\ & \overline{\overline{C+D}} \\ & C+D \end{aligned}$$

Como se puede observar esta segunda negación se obtiene de la segunda compuerta.

Observe el siguiente ejemplo:

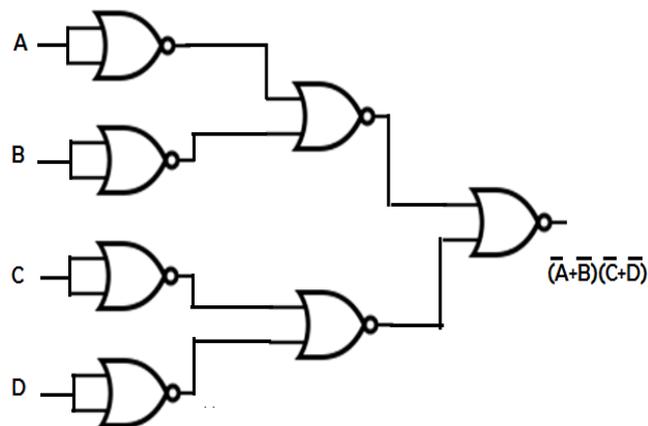


Figura 43 Diagrama de la aplicación de la compuerta NOR

Desarrollo explicación de pasos para obtener las respectivas salidas

De las dos primeras entradas A y B a la compuerta NOR se obtiene:

$$\overline{A} \text{ y } \overline{B} \quad \text{Respectivamente}$$

A la salida de la siguiente compuerta NOR se obtiene:

$$\begin{aligned}\overline{\overline{A + B}} &= \\ &= \overline{\overline{A}}\overline{\overline{B}} \\ &= (AB)\end{aligned}$$

De las dos siguientes entradas C y D a la compuerta NOR se obtiene:

$$\overline{C} \text{ y } \overline{D} \quad \text{Respectivamente}$$

A la salida de la siguiente compuerta NOR se obtiene:

Aplicando el Teorema de De Morgan

$$\begin{aligned}\overline{\overline{C + D}} &= \\ &= \overline{\overline{C}}\overline{\overline{D}} \\ &= (CD)\end{aligned}$$

La función de salida total será:

Aplicando el Teorema de De Morgan

$$\begin{aligned}\overline{AB + CD} &= \\ &= \overline{\overline{AB}}\overline{\overline{CD}} \\ &= (\overline{A + B})(\overline{C + D})\end{aligned}$$

Ejemplo:

Emplee compuertas NAND para habilitar las siguientes expresiones:

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

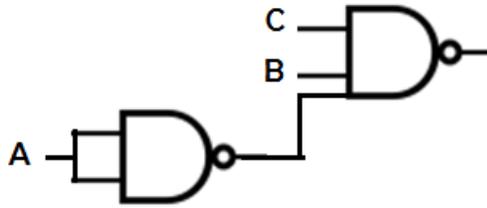


Figura 44 Diagrama de compuertas NAND para habilitar la expresión

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

Desarrollo:

$$\overline{\overline{A} B C}$$

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

A la entrada de la última compuerta NAND se tendrá:

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

Sea la siguiente expresión:

$$X = \overline{A} B C$$

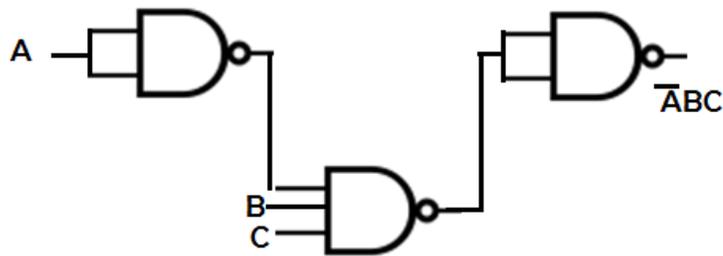


Figura 45 Diagrama de compuertas NAND para habilitar la expresión

$$X = \overline{A B C}$$

Desarrollo:

$$\begin{aligned} \overline{A B C} &= \\ &= \overline{\overline{A + B + C}} \end{aligned}$$

Introduciendo la expresión anterior en la última compuerta:

$$\begin{aligned} &\overline{\overline{A + B + C}} \\ &\overline{(\overline{A})(\overline{B})(\overline{C})} \\ &\overline{ABC} \end{aligned}$$

8.3 EJEMPLOS DE APLICACIÓN

Por ejemplo, para el empleo de la compuerta AND

La tabla de verdad es:

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

Tabla 31 Tabla de verdad para el ejercicio de la fábrica de pan

Se tiene una fábrica de pan

En donde ocurren los casos siguientes:

Si no hay masa y si no está preparado el horno para hornearla la banda transportadora no funciona

Si no hay masa, pero el horno está listo no camina la banda transportadora.

Si hay masa, pero el horno no está preparado para hornearla la banda transportadora no camina

Si la masa esta lista y el horno está preparado la banda transportadora camina y se activa el siguiente procedimiento.

SIGUIENTE PROCESO:

El proceso que le sigue los describiremos con una compuerta AND y una compuerta NOT

A	B	Salida(AND)	Salida(NOT)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Tabla 32 Tabla de verdad del siguiente proceso de la fábrica de pan

Si no hay pan en la charola y la banda transportadora no camina, suena una alarma para indicarle al vendedor que verifique el sistema.

Si no hay pan en la charola y se mueve la banda transportadora suena una alarma para que el vendedor verifique el funcionamiento del sistema.

Si hay pan y no se mueve la banda transportadora suena una alarma para que el vendedor verifique el funcionamiento del sistema.

Si hay pan y se mueve la banda transportadora la salida marca una música que le indica al vendedor que el sistema está funcionando muy bien.

Cuando se presenta este último estado el pan pasa a la charola de exhibición una vez allí con un sensor de presencia se activa una alarma y el vendedor va a acomodar la mercancía.

APLICACIÓN DE LA COMPUERTA NAND

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

Tabla 33 Tabla de verdad de la aplicación de la compuerta NAND

Análisis de las variables.

Para la variable A

A es 1 si es de día

A es 0 si es de noche

Para la variable B

B es 0 si no hay personas

B es 1 si hay personas

Análisis de los casos

Si es de noche y no hay personas se activa la lámpara que ilumina la calle y un video-cámara.

Si es de noche y hay personas se ilumina la lámpara y se prende la video-cámara

Si es de día y no hay personas se ilumina la lámpara y no se prende la video-cámara

Si es de día y hay personas no se ilumina la lámpara.

APLICACIÓN DE LA COMPUERTA NOR

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

Tabla 34 Tabla de verdad de un sistema de seguridad de una central camionera

Se tienen 2 detectores de seguridad en una central camionera, un sistema que revisa a las personas A y el segundo sistema B que revisa el equipaje.

Si el sensor no detecta ningún objeto metálico en la persona o el sensor B no detecta ningún objeto metálico en el equipaje una luz verde se prende para indicar que puede pasar.

Si el sensor A no detecta ningún objeto metálico o el sensor B detecta un objeto metálico la luz verde no se enciende.

Si el sensor A detecta que la persona tiene un objeto metálico o el sensor B detecta que el equipaje B tiene un objeto metálico la luz verde no se prende

Si el sensor A detecta que la persona tiene un objeto metálico y el sensor B detecta que el equipaje tiene un objeto metálico la luz verde no se enciende.

9. Mapas de Karnaugh

Los mapas de Karnaugh son un método práctico y eficiente para poder llevar a cabo simplificaciones algebraicas de funciones lógicas.

Un Mapa de Karnaugh está formado por cuadros o celdas, cada cuadro o celda representa un mintermino, como se aprecia en el siguiente ejemplo:

Se tiene un mapa de dos variables, X y Y por lo tanto se tendrán cuatro celdas donde se tendrán los minterminos de la siguiente manera:

		0	1
X	Y		
0		m0 $x'y'$	m1 $x'y$
1		m2 xy'	m3 xy

Como se puede ver en el cuadro el mintermino cero ocupa la primera casilla- primer renglón.

El mintermino uno ocupa la segunda casilla, primer renglón.

El mintermino dos ocupa la primera casilla el segundo renglón.

El mintermino tres ocupa el segundo renglón-segunda casilla.

EXPLICACIÓN DE LOS MINTÉRMINOS Y LOS MAXTÉRMINOS.

Sea el siguiente diagrama de Venn

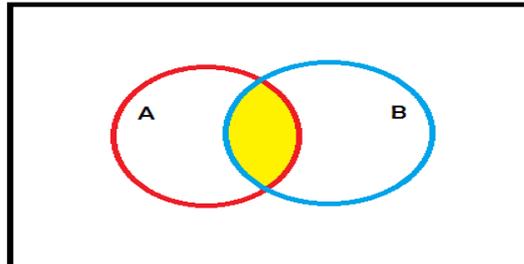


Figura 46 Diagrama de Venn de la operación AND

La intersección del conjunto A con el conjunto B representa los elementos que resultan de llevar la operación AND dando un número N de entradas al circuito donde $N \geq 2$.

A estos elementos se les conoce como mintérminos, los mintérminos se representan con una letra m minúscula y un subíndice el cual indica el número decimal del cual se está obteniendo el mintérmino.

Es decir, el mintérmino es el número binario en la tabla de verdad que representa las entradas del sistema.

Como se muestra en el siguiente ejemplo:

Se tiene un problema donde intervienen tres variables de entrada, los cuales representan tres situaciones distintas en el sistema, estas tres situaciones tienen 8 posibles combinaciones en las que pueden ocurrir, esto se ve en la tabla de verdad siguiente:

X	Y	Z	Mintérmino (expresado con variables)	mintérmino
0	0	0	$x'y'z'$	m0
0	0	1	$x'y'z$	m1
0	1	0	$x y z'$	m2
0	1	1	$x'y z$	m3
1	0	0	$x y' z'$	m4
1	0	1	$x y' z$	m5
1	1	0	$x y z'$	m6
1	1	1	Xyz	m7

MAXTÉRMINOS

Un análisis similar se utiliza para explicar que son los maxtérminos.

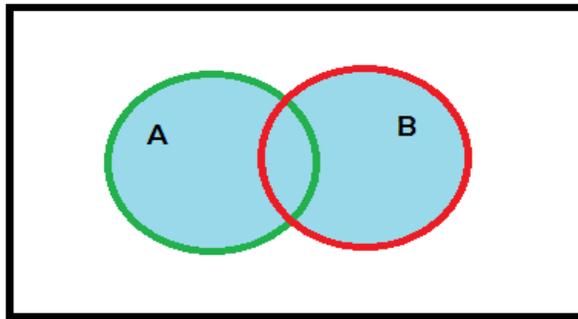


Figura 47 Diagrama de Venn de la operación OR

Para el caso de los maxtérminos se hace uso de la compuerta OR. Es decir, los maxtérminos pertenecen al grupo de elementos que se encuentran en el conjunto A o el conjunto B o ambos, es decir la compuerta OR representa la operación disyunción o unión.

Los maxtérminos son el complemento de los mintérminos y se caracteriza por que los elementos se disyuntas (se suman).

Se les designa con la letra M (mayúscula) y con un subíndice numérico, cabe destacar que $N \geq 2$.

El subíndice numérico indica el equivalente en decimal del número binario de la tabla de verdad.

Un ejemplo de ello se muestra en la siguiente tabla.

Se tienen tres variables de entrada los cuales representan tres posibles situaciones, el número de combinaciones en que se pueden presentar estas tres combinaciones es ocho, como se muestra en la tabla de verdad.

X	Y	Z	Maxtérmino (en términos de las variables)	Maxtérmino
0	0	0	$x+y+z$	M0
0	0	1	$x+y+z'$	M1
0	1	0	$x+y'+z$	M2
0	1	1	$x+y'+z'$	M3
1	0	0	$x'+y+z$	M4
1	0	1	$x'+y'+z$	M5
1	1	0	$x'+y'+z$	M6
1	1	1	$x'+y'+z'$	M7

Nota: Los mintérminos se indican con un 1 y representan una función SOP

Los maxtérminos se indican con un cero y representan una función POS

Más adelante se explicará que es una función SOP y POS

Como se mencionó un mapa de Karnaugh está formado por celdas donde cada una representa un mintérmino de cual también se puede obtener el correspondiente Maxtérmino. Analizando más el Mapa de Karnaugh al exterior de las celdas se colocan las variables y el valor binario de las entradas. Es decir, los números representan las posibles combinaciones que puede tomar el valor por ejemplo Si se tiene una variable

esta solo puede tener dos posibles valores 1 y 0, estos son los que se colocan en la tabla. Cuando se representa la entrada de dos variables éstas pueden tomar los valores 00, 01, 11 y 10 y son estos valores los que se ponen en la tabla. Estos valores no son arbitrarios siguen el principio del Código Gray, es decir cambian en un bit para asegurar la adyacencia y con ello permitir la continuidad de la función, como recordará de sus cursos de cálculo una función debe cumplir con ser continua en todo su dominio.

El número de celdas se obtiene de la siguiente expresión:

$$2^n$$

Donde n es el número de variables.

Dependiendo del número de variables será el número de cuadros de la siguiente manera:

Si se tienen dos variables se tendrá 4 cuadros dado que $2^2 = 4$, si se tienen tres variables se emplearán 8 cuadros dado que $2^3 = 8$

El número de cuadros o celdas que se empleen dependerá de la manera en que se lleven a cabo las agrupaciones.

Por ejemplo:

Para 2 variables:

$$2^2 = 4$$

Se tendrán 4 celdas

Para 3 variables

$$2^3 = 8$$

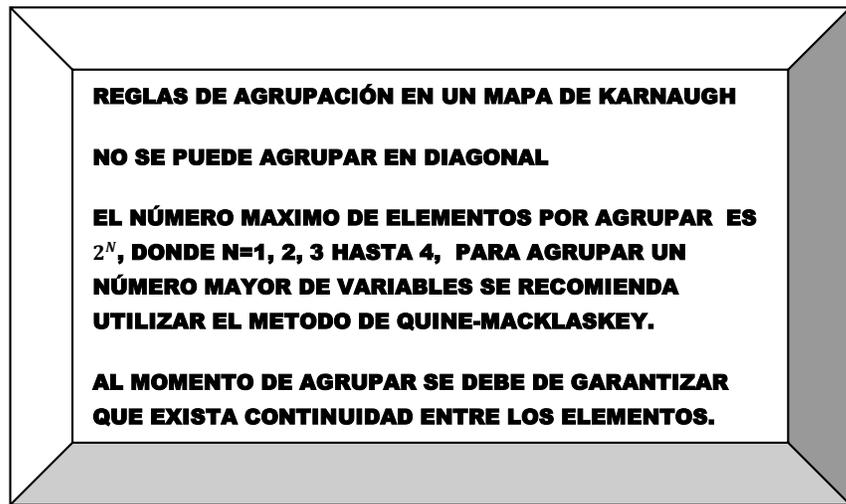
Se tendrán 8 celdas

Para 4 variables

$$2^4 = 16$$

Se tendrán 16 celdas

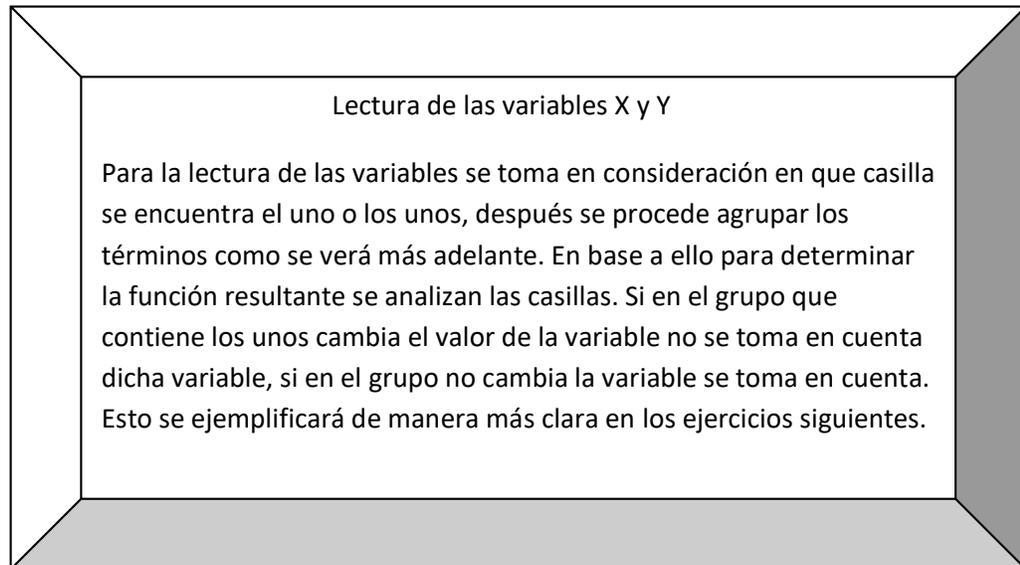
En caso de que se tenga un número igual a 5 o más variables emplearemos otro método conocido con el nombre de Quine-McCluskey el cual expondremos en la sección siguiente.



El mapa de Karnaugh para dos variables es como se muestra a continuación.

	$x \backslash y$	0	1
0		00	01
1		10	11

Mapa de Karnaugh 1 Agrupación de términos



Características importantes del Mapa de Karnaugh

Eliminación de variables en el grupo

Una de las propiedades importantes del Mapa de Karnaugh es la adyacencia. Es decir que entre dos casillas que son adyacentes solo cambia un bit.

A este hecho le podemos aplicar una de las propiedades del algebra booleana y demostrar el resultado de la simplificación de las casillas adyacentes.

Por ejemplo:

Los siguientes mintérminos son adyacentes.

$$m_3 \text{ con } m_2$$

Como se muestra en el siguiente mapa.

		BC			
A		00	01	11	10
0				m3	m2
1					

Realizando un análisis con las propiedades del algebra booleana.

$$m_3 + m_2 = \bar{A}BC + \bar{A}B\bar{C}$$

Si aplicamos las propiedades: asociativa y el postulado $x + \bar{x} = 1$

Se tendrá:

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

Por otra parte, el Mapa de Karnaugh puede verse como un arreglo de matrices que presentan continuidad.

Si tomamos el mapa de Karnaugh en la hoja en la cual lo hemos escrito y doblamos esta hoja de la siguiente manera:

Si doblamos uniendo los lados laterales veremos que se presenta una continuidad entre un lado y el otro, dándose la propiedad de adyacencia es decir solo cambiara un bit entre las casillas que se unieron.

Si volvemos a realizar un procedimiento semejante pero ahora juntamos el lado superior con el lado inferior vemos nuevamente que se presenta la continuidad entre las celdas, apareciendo nuevamente la propiedad de adyacencia en donde solamente varía un bit.

Ejemplo

Sea la siguiente tabla de verdad expresar la función de salida en un mapa de Karnaugh

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

Tabla 35 Valores de entrada del mapa de Karnaugh

Mapeando los valores de salida dados en alto en el Mapa de Karnaugh

		B	
		0	1
A	0	0	1
	1	1	0

Mapa de Karnaugh 2 Agrupación de términos

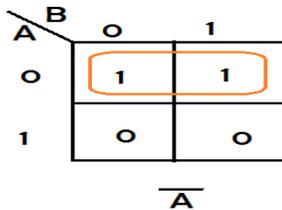
La expresión resultante será para el mapa de Karnaugh 2 es:

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

Para llevar a cabo la lectura se realizan los siguientes pasos:

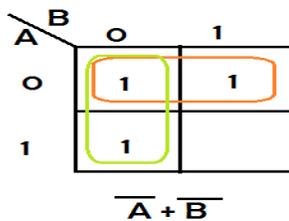
Después de que se han agrupado los unos se procede a obtener los términos de la función. Para el caso del primer uno que se encuentra en la primera columna y el segundo renglón se lee qué valor tiene la variable A, a la cual le corresponde un valor de 1, después se procede a leer el valor de la segunda variable el cual es 0 por lo tanto es B'. Para la lectura del segundo uno que se encuentra en la segunda columna primer-renglón se lee el valor de la variable A el cual es 0 por lo tanto es A' y luego se lee el valor de B el cual le corresponde el valor de 1 correspondiéndole la letra B. El orden en que se leen las variables depende de cual representa al bit más significativo en este caso A es MSB y B es LSB.

FORMAS DE AGRUPACIÓN EN UN MAPA CON DOS VARIABLES

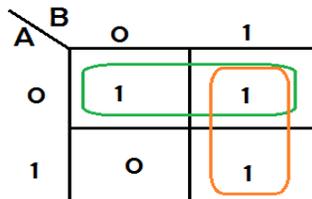


Mapa de Karnaugh 3 Agrupación de términos

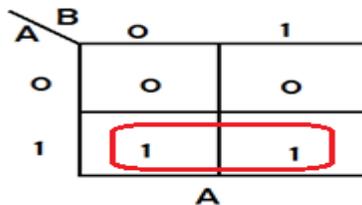
Es A negada porque A es constante, la variable B cambia, clásicos casos son variables que se tienen que eliminar.



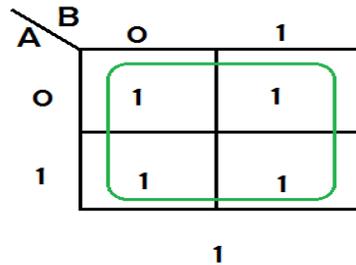
Mapa de Karnaugh 4 Agrupación de términos



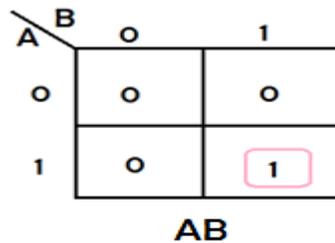
Mapa de Karnaugh 5 Agrupación de términos



Mapa de Karnaugh 6 Agrupación de términos



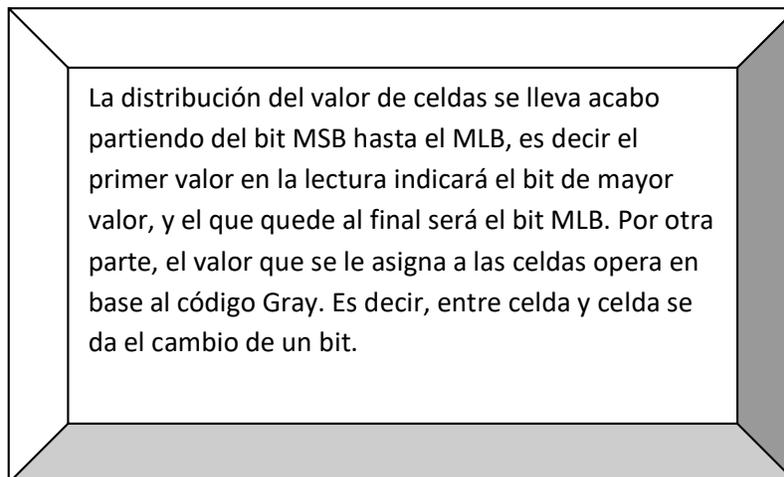
Mapa de Karnaugh 7 Agrupación de términos



Mapa de Karnaugh 8 Agrupación de términos

A continuación, se muestra el Mapa de Karnaugh para tres variables

Para tres variables XYZ



		C	
		0	1
AB	00	000	001
	01	010	011
	11	110	111
	10	100	101

Mapa de Karnaugh 9 Agrupación de términos

Por la adyacencia de las celdas se utilizó código gray para garantizar la continuidad entre celdas adyacentes

Número Decimal	Número Binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Tabla 36 Número del 0 al 7 en decimal y binario

Si analizamos la tabla 1 y el cuadro podremos ver que la tabla representa a los números del 0 al 7 en binario y si vemos el cuadro 1 notamos que cada celda representa uno de estos números binarios.

Trata de construir por ti mismo la tabla y el cuadro para el caso de cuatro variables.

Desarrollo:

Primero planteamos la ecuación:

$$2^n$$

En este caso $n=4$

$$2^4 = 16$$

Las 16 combinaciones posibles.

Ahora se procede a construir la tabla con los 16 valores numéricos para ver las entradas.

Número decimal	WXYZ / número binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Tabla 37 Números del 0 al 15 en decimal y binario

La codificación superior y laterales son importantes para obtener la reducción total de la función. Se necesita tomar en cuenta los MSB y los LSB, así como los números que se emplean en código Gray, porque esto nos garantiza que la función es continua y cíclica.

		YZ			
	WX	00	01	11	10
00		0000	0001	0011	0010
01		0100	0101	0111	0110
11		1100	1101	1111	1110
10		1000	1001	1011	1010

Mapa de Karnaugh 10 Colocación de valores binarios en las celdas

WZ corresponde a los bits más significativos y YZ corresponden a los bits menos significativos

Un punto que es importante destacar dentro de los mapas de Karnaugh es la adyacencia de celdas ¿Qué significa lo anterior?

Cuando los cuadros están uno al lado del otro, pero nunca en diagonal, porque no garantizaría continuidad.

O también puede darse por ejemplo cuando imaginamos que el mapa de karnaugh está sobre una hoja la cual podemos doblar para formar un cilindro de dos formas distintas ya sea que doblemos la hoja de abajo hacia arriba o de izquierda a derecha.

Podemos ver que hay cuadros o celdas que han quedado adyacentes después de formar el cilindro.

Muy bien ya hemos visto cómo se construye un Mapa de Karnaugh a continuación veremos cómo mapear una expresión SOP o suma de productos.

Mapear es colocar cada termino SOP en la celda que le corresponde mediante un 1

Ejemplos:

Mapee la siguiente expresión SOP en el mapa de Karnaugh siguiente y agrupando los términos

$$\bar{X}\bar{Y}\bar{Z} + XY\bar{Z} + \bar{X}YZ + XYZ$$

		Z	
		0	1
XY	00	1	
	01		1
	11	1	1
	10		

Mapa de Karnaugh 11 Agrupación de término

La función reducida del mapa de Karnaugh 11 será:

$$\bar{X}\bar{Y}\bar{Z} + YZ + XY$$

Mapeando la siguiente función en el Mapa de Karnaugh

$$XY\bar{Z} + \bar{X}\bar{Y}Z + X\bar{Y}Z$$

		Z	
		0	1
XY	00		1
	01		
	11	1	
	10		1

Mapa de Karnaugh 12 Agrupación de los términos

La función reducida del mapa de Karnaugh 12 será:

$$\bar{X}\bar{Y}Z + XY\bar{Z} + X\bar{Y}Z$$

Mapee la siguiente expresión SOP en el mapa de Karnaugh:

$$\bar{X}\bar{Y}\bar{Z} + XY\bar{Z} + \bar{X}YZ + X\bar{Y}Z$$

		z	
		0	1
xy	00	1	
	01		1
	11	1	
	10		1

Mapa de Karnaugh 13 Agrupación de términos

La función reducida del mapa de Karnaugh 13 quedaría como:

$$\bar{x}\bar{y}\bar{z} + xy\bar{z} + \bar{x}yz + x\bar{y}z$$

Si se reduce la función mediante algebra quedaría como:

$$\bar{x}\bar{y}\bar{z} + x y \bar{z} + \bar{x}yz + x\bar{y}z$$

Aplicando la propiedad asociativa

$$\bar{y}(\bar{x}\bar{z} + xz) + y(x\bar{z} + \bar{x}z)$$

Aplicando la propiedad $A+A'=1$

Se tiene

$$\bar{y}(1) + y(1) = 1$$

Reducción por el método de Quine-Macluskey

De la función se obtienen los mintérminos

$$f = \sum (0,3,5,6)$$

Se ubican en una tabla como la siguiente, los mintérminos

Mintérmino	X	Y	Z
0	0	0	0
1	-	-	-
2	-	-	-
3	0	1	1
4	-	-	-
5	1	0	1
6	1	1	0
7	-	-	-

Tabla 38 Tabla de mintérminos (0,3,5,6)

Posteriormente se realiza una segunda tabla en donde se agrupan los términos en base al número de unos que tienen por cada mintérmino o si no tiene unos.

La tabla se muestra a continuación.

Mintérmino	X	Y	Z
0	0	0	0
3	0	1	1
5	1	0	1
6	1	1	0

Tabla 39 Agrupación de términos en base al número de mintérminos que tienen

A continuación, se realizan las combinaciones.

Las combinaciones se realizan comparando los renglones entre sí buscando que solo difieran en un bit, si esto no ocurre no es posible la combinación.

Si comparamos el primer renglón con los demás veremos que difieren en dos bits por lo cual no es posible la combinación y por lo tanto la reducción con el método de Quine McCluskey.

9.1 EXPRESIONES SOP Y POS

Mapee la siguiente expresión SOP en el mapa de Karnaugh

$$\bar{W}\bar{X}\bar{Y}\bar{Z} + \bar{W}X\bar{Y}\bar{Z} + W\bar{X}YZ + W\bar{X}Y\bar{Z}$$

		YZ			
	wx	00	01	11	10
00		1			
01		1			
11					
10				1	1

Mapa de Karnaugh 14 Agrupación de términos en el mapa

Reduciendo la expresión anterior en el Mapa

		YZ			
	wx	00	01	11	10
00		1			
01		1			
11					
10				1	1

Mapa de Karnaugh 15 Agrupación de términos en el mapa

La expresión reducida del mapa de Karnaugh 15 quedaría como:

$$\bar{W}\bar{Y}\bar{Z} + W\bar{X}Y$$

En el ejemplo siguiente se reduce mediante algebra la ecuación anterior.

$$\bar{W}\bar{X}\bar{Y}\bar{Z} + \bar{W}X\bar{Y}\bar{Z} + W\bar{X}YZ + W\bar{X}Y\bar{Z}$$

Reduciendo con algebra

$$\bar{W}\bar{X}\bar{Y}\bar{Z} + \bar{W}X\bar{Y}\bar{Z} + W\bar{X}Y(Z + \bar{Z})$$

Aplicando la propiedad $A+A'=1$

$$\bar{W}\bar{X}\bar{Y}\bar{Z} + \bar{W}X\bar{Y}\bar{Z} + W\bar{X}Y$$

$$\bar{W}\bar{Y}\bar{Z}(\bar{X} + X) + W\bar{X}Y$$

Aplicando la propiedad $A+A'=1$

$$\bar{W}\bar{Y}\bar{Z} + W\bar{X}Y$$

Mapee la siguiente expresión SOP en el mapa de Karnaugh:

$$\bar{W}\bar{X}\bar{Y}\bar{Z} + W\bar{X}\bar{Y}\bar{Z} + W\bar{X}YZ + WXY\bar{Z}$$

Agrupando los términos en el Mapa y reduciendo

		YZ			
		00	01	11	10
WX	00				
	01	1			
	11				1
	10		1	1	

Mapa de Karnaugh 16 Agrupación de los términos en el mapa

La mínima expresión sería con base en el Mapa de Karnaugh número 16:

$$W\bar{X}Z + \bar{W}XYZ + WXY\bar{Z}$$

Reduciendo la expresión inicial con algebra.

$$\bar{W}X\bar{Y}\bar{Z} + W\bar{X}\bar{Y}Z + W\bar{X}YZ + WXY\bar{Z}$$

Aplicando la propiedad de $A+A'=1$

$$\bar{W}X\bar{Y}\bar{Z} + W\bar{X}Z(Y + \bar{Y}) + WXY\bar{Z}$$

$$\bar{W}X\bar{Y}\bar{Z} + WXY\bar{Z} + W\bar{X}Z$$

$$X\bar{Z}(\bar{W}\bar{Y} + WY) + W\bar{X}Z$$

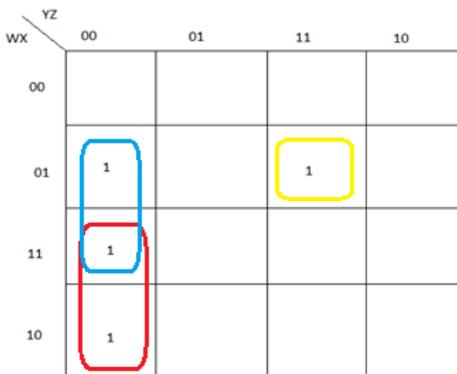
Aplicando la propiedad de $A+A'=1$

La minimización con algebra quedaría como:

$$X\bar{Z} + W\bar{X}Z$$

Mapee la siguiente expresión SOP en el mapa de Karnaugh:

$$\bar{W}X\bar{Y}\bar{Z} + W\bar{X}\bar{Y}\bar{Z} + WXY\bar{Z} + \bar{W}XYZ$$



Mapa de Karnaugh 17 Agrupación de términos en el mapa

En base al mapa la mínima expresión del mapa de Karnaugh 17 quedaría como:

$$X\bar{Y}\bar{Z} + W\bar{Y}\bar{Z} + \bar{W}XYZ$$

Reduciendo la expresión siguiente con algebra

$$\bar{W}X\bar{Y}\bar{Z} + W\bar{X}\bar{Y}\bar{Z} + WX\bar{Y}\bar{Z} + \bar{W}XY\bar{Z}$$

$$X\bar{Y}\bar{Z}(W + \bar{W}) + W(\bar{X}\bar{Y}\bar{Z}) + \bar{W}XY\bar{Z}$$

Aplicando la propiedad $A + A' = 1$

$$X\bar{Y}\bar{Z} + W\bar{X}\bar{Y}\bar{Z} + \bar{W}XY\bar{Z}$$

$$X\bar{Y}\bar{Z} + \bar{Z}(W\bar{X}\bar{Y} + XY\bar{W})$$

Aplicando la propiedad $A + A' = 1$

Se obtiene la expresión

$$X\bar{Y}\bar{Z} + \bar{Z}$$

10. FORMAS CANONICAS Y ESTANDÁR

Cuando se expresan los mintérminos o maxtérminos que intervienen en una función tomados de una tabla de verdad se dice que se está empleando una forma canónica.

Por ejemplo:

Se tienen la siguiente tabla de verdad y se expresa para que valores la función es 1.

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Tabla 40 Ejemplo de tabla de verdad donde la función vale 1

En base a la tabla de verdad la función expresada en su forma canónica utilizando mintérminos sería:

$$\bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$$

Otra forma es la función expresada en su forma canónica utilizando maxtérminos será:

$$(A + B + \bar{C})(\bar{A} + B + C)(\bar{A} + \bar{B} + \bar{C})$$

Como se puede observar la función se puede expresar como una función SOP (suma de productos) como el primer caso o una función POS (producto de sumas) como en el segundo caso.

Ambas están en forma canónica porque se tomaron de la tabla de verdad, y para ello en cada uno de los términos aparecen las variables de entrada ya se en su forma complementada o sin complementar.

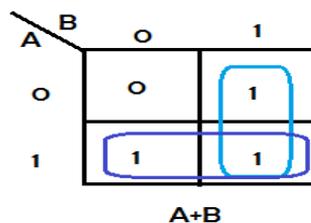
FORMA ESTANDAR

Otra forma de expresar una función es en su forma estándar.

La forma estándar se caracteriza por encontrarse reducida, es decir contiene el mínimo número de variables de la función.

Por ejemplo:

El resultado de mapear una función en un Mapa de Karnaugh es el siguiente:



La función ahora se encuentra en su forma estándar porque la expresión contiene las variables simplificadas.

En los ejemplos anteriores todas las funciones SOP se encontraban en su forma canónica.

Otro nombre que recibe la función SOP es suma de productos, dado que las variables que proceden de los mintérminos se encuentran multiplicándose y en conjunto cada uno de estos términos se suma.

Una función en su forma canónica implica que no se encuentra reducida, es decir que los mintérminos contienen todas las variables de entrada ya sea que estén complementadas o no.

Si la función SOP se toma de la tabla de verdad, entonces la función está en su forma canónica. Si la función SOP se toma de un Mapa de Karnaugh es decir en forma reducida, entonces la función está en forma estándar.

Nota: Cuando una función SOP no se encuentra en su forma estándar no es posible mapearla, para lo cual es necesario expandir la función

Primero se debe expandir para dejar en su forma canónica para poder mapearla en el mapa de Karnaugh

Ejemplo:

Sea la siguiente función de tres variables XYZ

$$XY + XY\bar{Z} + Z$$

Como podemos apreciar la función no se encuentra en su forma canónica, es necesario completar los términos de la siguiente manera:

El primero XY

Se completa con z complementada y sin complementar

$$XYZ + XY\bar{Z}$$

El término

$$XY\bar{Z}$$

Ya está completo puesto que tiene las tres variables.

El siguiente término es Y

Para complementar el término hacen falta las variables X y Z sin complementar y complementadas por lo cual el término quedaría como:

$$YX + Y\bar{X}$$

Agregando la tercera variable Z complementada y sin complementar

$$YXZ + YX\bar{Z} + Y\bar{X}Z + Y\bar{X}\bar{Z}$$

La expresión completa quedaría como:

$$XYZ + XY\bar{Z} + X\bar{Y}Z + X\bar{Y}\bar{Z} + XY\bar{Z} + X\bar{Y}Z + \bar{X}YZ + \bar{X}\bar{Y}\bar{Z}$$

Quitando los términos que se repiten la expresión final quedaría como:

Agrupando los términos en el Mapa de Karnaugh

		z	
		0	1
xy			
00			
01		1	1
11		1	1
10			

Mapa de Karnaugh 18 agrupando términos

La expresión reducida del mapa de Karnaugh 18 quedaría como: Y

Reduciendo la expresión con algebra.

$$XYZ + XY\bar{Z} + \bar{X}YZ + \bar{X}Y\bar{Z}$$

$$YZ(X + \bar{X}) + XY\bar{Z} + \bar{X}Y\bar{Z}$$

Aplicando la propiedad $A + A' = 1$

$$YZ + XY\bar{Z} + \bar{X}Y\bar{Z}$$

$$YZ + Y\bar{Z}(X + \bar{X})$$

Aplicando la propiedad $A + A' = 1$

$$YZ + Y\bar{Z}$$

$$Y(Z + \bar{Z})$$

Aplicando la propiedad $A + A' = 1$

$$Y$$

Sea la siguiente función booleana mapéela en un mapa de Karnaugh

$$XYZ + XY\bar{Z} + ZY + Z$$

Análisis

Como podemos observar la función se conforma por tres variables y a también podemos decir por la forma en que se encuentran las variables que es una función de tipo SOP.

Los dos primeros términos están en su forma canónica, los otros dos no.

Se procede a completar los otros dos términos:

Se tiene el término ZY

Se procede a completar con X complementada y sin complementar:

$$ZXY + ZY\bar{X}$$

Al último término le hacen falta las variables XY

Por lo tanto, procedemos a completar:

Si la función se compone de tres variables: X, Y y Z y solo tenemos la variable Z procedemos a completar con X y Y sin complementar y complementadas respectivamente:

$$ZX + Z\bar{X}$$

Con Y

$$ZXY + ZX\bar{Y} + Z\bar{X}Y + Z\bar{X}\bar{Y}$$

Finalmente, la función que se obtiene es:

$$XYZ + XY\bar{Z} + X\bar{Y}Z + \bar{X}YZ + XYZ + X\bar{Y}Z + \bar{X}YZ + \bar{X}\bar{Y}Z$$

Quitamos los términos que se repiten

$$XYZ + XY\bar{Z} + \bar{X}YZ + X\bar{Y}Z + \bar{X}YZ + \bar{X}\bar{Y}Z$$

Ordenando los términos

$$XYZ + XY\bar{Z} + \bar{X}YZ + X\bar{Y}Z + \bar{X}\bar{Y}Z$$

Mapeando la expresión:

$$XYZ + XY\bar{Z} + \bar{X}YZ + X\bar{Y}Z + \bar{X}\bar{Y}Z$$

Reduciendo la expresión en un Mapa de Karnaugh

		Z	
XY		0	1
00			1
01			1
11	1	1	1
10			1

Mapa de Karnaugh 19 Agrupando términos

En base al Mapa de Karnaugh número 19 la expresión reducida quedaría como:

$$Z + XY$$

Reduciendo la expresión con algebra

$$XYZ + XY\bar{Z} + \bar{X}YZ + X\bar{Y}Z + \bar{X}\bar{Y}Z$$

$$XY(Z + \bar{Z}) + \bar{X}YZ + X\bar{Y}Z + \bar{X}\bar{Y}Z$$

$$XY + \bar{X}YZ + X\bar{Y}Z + \bar{X}\bar{Y}Z$$

$$XY + \bar{X}YZ + \bar{Y}\bar{Z}(X + \bar{X})$$

$$XY + \bar{X}YZ + \bar{Y}\bar{Z}$$

Sea la siguiente función booleana mapearla en un mapa de Karnaugh.

$$WXY\bar{Z} + W\bar{X}\bar{Y}Z + WX + Z$$

Análisis:

Como podemos apreciar la función es de 4 variables donde los dos primeros términos están en su forma estándar y las otras no.

Por lo tanto, para mapear la expresión, procedemos a completar los términos:

Se tienen las variables WXYZ

El tercer término solo contiene a las variables WX por lo cual procedemos a completar con las variables YZ en su forma complementada y sin complementar.

Desarrollo:

$$WXY + W\bar{X}\bar{Y}$$

A completando los términos anteriores con la variable z

$$WXYZ + WXY\bar{Z} + W\bar{X}\bar{Y}Z + W\bar{X}\bar{Y}\bar{Z}$$

El cuarto término de la expresión principal solo contiene a la variable Z procedemos a completar con las variables W, X y Y en su forma complementada y sin complementar.

Desarrollo:

Z

$$ZW + Z\bar{W}$$

$$ZWX + ZW\bar{X} + Z\bar{W}X + Z\bar{W}\bar{X}$$

$$ZWXY + ZWXY\bar{Z} + ZW\bar{X}Y + ZW\bar{X}\bar{Y} + Z\bar{W}XY + Z\bar{W}X\bar{Y} + Z\bar{W}\bar{X}Y + Z\bar{W}\bar{X}\bar{Y}$$

Ordenando las variables:

$$WXYZ + WXY\bar{Z} + W\bar{X}YZ + W\bar{X}\bar{Y}Z + \bar{W}XYZ + \bar{W}X\bar{Y}Z + \bar{W}\bar{X}YZ + \bar{W}\bar{X}\bar{Y}Z$$

Mapeando la función resultante en un Mapa de Karnaugh

$$WXYZ + WXY\bar{Z} + W\bar{X}YZ + W\bar{X}\bar{Y}Z + \bar{W}XYZ + \bar{W}X\bar{Y}Z + \bar{W}\bar{X}YZ + \bar{W}\bar{X}\bar{Y}Z$$

Agrupando los términos en el Mapa de Karnaugh

		YZ			
WX		00	01	11	10
00			1		1
01			1		1
11			1		1
10			1		1

Mapa de Karnaugh 20 Agrupando términos en el mapa

La función reducida en base al Mapa número 20 será:

$$\bar{Y}Z + Y\bar{Z}$$

Reduciendo la expresión con algebra.

$$WXYZ + WX\bar{Y}Z + W\bar{X}YZ + W\bar{X}\bar{Y}Z + \bar{W}XYZ + \bar{W}X\bar{Y}Z + \bar{W}\bar{X}YZ + \bar{W}\bar{X}\bar{Y}Z$$

Agrupando términos y utilizando la propiedad $A+A'=1$

$$Z\bar{W}\bar{X}(Y + \bar{Y}) + XYZ(W + \bar{W}) + X\bar{Y}Z(W + \bar{W}) + W\bar{X}Z(Y + \bar{Y})$$

$$Z\bar{W}\bar{X} + XYZ + X\bar{Y}Z + W\bar{X}Z$$

Agrupando términos y utilizando la propiedad $A+A'=1$

$$Z\bar{X}(W + \bar{W}) + XZ(Y + \bar{Y})$$

$$\bar{X}Z + XZ$$

$$Z(X + \bar{X})$$

Z

Utilidad de los Mapas de Karnaugh

Como se había mencionado anteriormente los mapas de Karnaugh ayudan a simplificar expresiones SOP.

A continuación, se verá como:

Sea la siguiente función booleana mapearla en un Mapa de Karnaugh

$$\overline{X}\overline{Y}\overline{Z} + \overline{X}Y\overline{Z} + \overline{X}\overline{Y}Z + \overline{X}YZ$$

Agrupando los términos en el Mapa de Karnaugh

		z	
		0	1
xy	00	1	1
	01	1	1
	11		
	10		

Mapa de Karnaugh 21 Agrupando términos en el mapa

En base al Mapa de Karnaugh número 21 la función reducida será: \overline{X}

Reduciendo la expresión con algebra.

$$\overline{X}\overline{Y}\overline{Z} + \overline{X}Y\overline{Z} + \overline{X}\overline{Y}Z + \overline{X}YZ$$

$$\overline{X}\overline{Y}\overline{Z} + \overline{X}Y\overline{Z} + \overline{X}Z(\overline{Y} + Y)$$

Aplicando la propiedad $A+A'=1$

$$\overline{X}\overline{Y}\overline{Z} + \overline{X}Y\overline{Z} + \overline{X}Z$$

$$\overline{X}\overline{Z}(\overline{Y} + Y) + \overline{X}Z$$

$$\overline{X}\overline{Z} + \overline{X}Z$$

$$\overline{X}(\overline{Z} + Z)$$

Aplicando la propiedad $A+A'=1$

La función reducida quedaría como:

$$\bar{X}$$

PROCEDIMIENTO DE AGRUPACIÓN DE LAS VARIABLES

Para una función de tres variables la forma de agrupar es la siguiente:

XY \ Z	0	1
00	1	1
01		
11		
10		

Mapa de Karnaugh 22 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 22 es

$$\bar{X}\bar{Y}$$

La agrupación mínima para el mapa de Karnaugh 23 es

$$\bar{Z}$$

		Z	
			0 1
XY	00	1	
	01	1	
	11	1	
	10	1	

Mapa de Karnaugh 23 Agrupación de términos

Sea la ecuación

$$\bar{X}\bar{Y}Z + XY\bar{Z}$$

		Z	
			0 1
XY	00		1
	01		
	11	1	
	10		

Mapa de Karnaugh 24 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 24 es

$$XY\bar{Z} + \bar{X}\bar{Y}Z$$

		Z	
			0 1
XY	00		1
	01		1
	11		1
	10		1

Mapa de Karnaugh 25 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh de la figura 25 es

$$Z$$

		z	
			0 1
xy			
	00		
	01	1 1	
	11		
	10		

Mapa de Karnaugh 26 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 26 es

XY

		z	
			0 1
xy			
	00		
	01		
	11	1 1	
	10		

Mapa de Karnaugh 27 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 27 es

XY

		z	
			0 1
xy			
	00		
	01	1 1	
	11	1 1	
	10		

Mapa de Karnaugh 28 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 28 es

Y

		0	1
XY	Z		
	00		
	01		
	11		
	10	1	1

Mapa de Karnaugh 29 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 29 es

$$X\bar{Y}$$

Que la función resultante es un 1, es decir que la salida siempre estará conectada a un uno lógico o su equivalente en este caso Vcc

		0	1
XY	Z		
	00	1	1
	01	1	1
	11	1	1
	10	1	1

Mapa de Karnaugh 30 Agrupación de términos

¿Cómo se agrupan para funciones de 4 variables?

La agrupación mínima para el mapa de Karnaugh número 30 es :1

		00	01	11	10
WX	YZ				
	00				
	01	1	1		
	11	1	1		
	10				

Mapa de Karnaugh 31 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 31 es:

$$XZ$$

		YZ			
	WX	00	01	11	10
00					1
01					1
11					1
10					1

Mapa de Karnaugh 32 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 32 es:

$$\overline{Y}Z$$

		YZ			
	WX	00	01	11	10
00		1	1	1	1
01					
11					
10					

Mapa de Karnaugh 33 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 33 es

$$\overline{W}\overline{X}$$

		YZ			
	WX	00	01	11	10
00		1			
01		1			
11		1			
10		1			

Mapa de Karnaugh 34 Agrupación de términos

El resultado de simplificar la función del mapa de Karnaugh 34 es:

$$\bar{Y}\bar{Z}$$

Mapeando la siguiente función en un Mapa de Karnaugh

$$\bar{W}\bar{X}\bar{Y}Z + \bar{W}\bar{X}Y\bar{Z} + W\bar{X}\bar{Y}\bar{Z}$$

		YZ			
	WX	00	01	11	10
00			1		1
01					
11		1			
10					

Mapa de Karnaugh 35 Agrupación de términos

Recuerde lo que se explicó sobre los Mapas de Karnaugh cuando lo podía tomar como una hoja de papel y formar dos cilindros y hacer si es que se da el caso que los unos coincidan.

Analice las siguientes formas de agrupación que siguen el principio anterior.

La agrupación mínima para el mapa de Karnaugh número 35 es:

$$\overline{W}\overline{X}\overline{Y}Z + \overline{W}\overline{X}Y\overline{Z} + W\overline{X}\overline{Y}\overline{Z}$$

		YZ			
		00	01	11	10
WX	00	1			1
	01	1			1
	11	1			1
	10	1			1

Mapa de Karnaugh 36 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 36 es

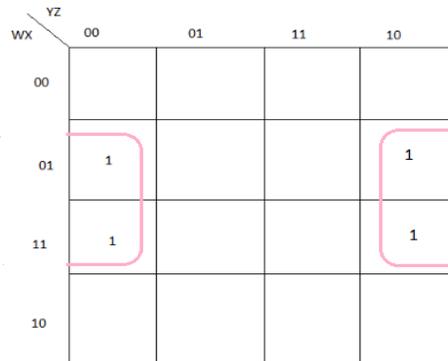
$$\overline{Z}$$

		YZ			
		00	01	11	10
WX	00	1	1	1	1
	01				
	11				
	10	1	1	1	1

Mapa de Karnaugh 37 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 37 es

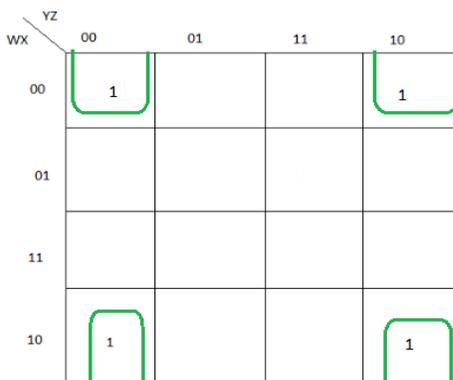
$$\overline{X}$$



Mapa de Karnaugh 38 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 38 es:

$$\overline{Z} X$$



Mapa de Karnaugh 39 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 39 es :

$$\overline{X} Z$$

Obteniendo la mínima expresión del siguiente Mapa de Karnaugh

		YZ			
wx		00	01	11	10
00			1	1	
01					
11					
10			1	1	

Mapa de Karnaugh 40 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 41 es

$$\bar{X}Z$$

Ejemplo sea el Mapa de Karnaugh

		YZ			
wx		00	01	11	10
00			1	1	
01			1	1	
11					
10					1

Mapa de Karnaugh 41 Agrupación de términos

La función en su mínima expresión del Mapa de Karnaugh número 42 quedaría como:

$$\bar{W}Z + W\bar{X}Y\bar{Z}$$

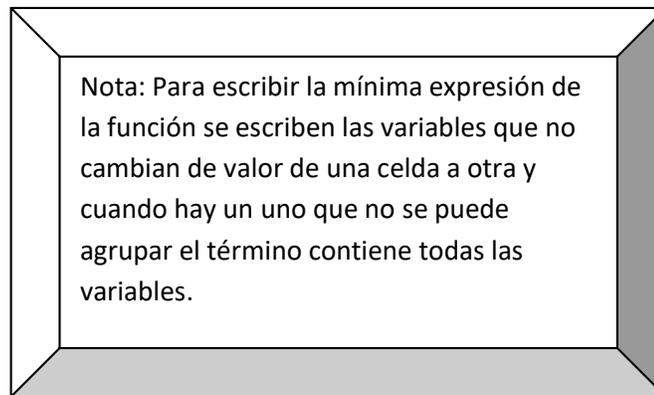
Obteniendo la forma reducida de la función en base al siguiente Mapa de Karnaugh

		YZ			
		00	01	11	10
wx	00	1			1
	01	1			1
	11				
	10		1		

Mapa de Karnaugh 42 Agrupación de términos

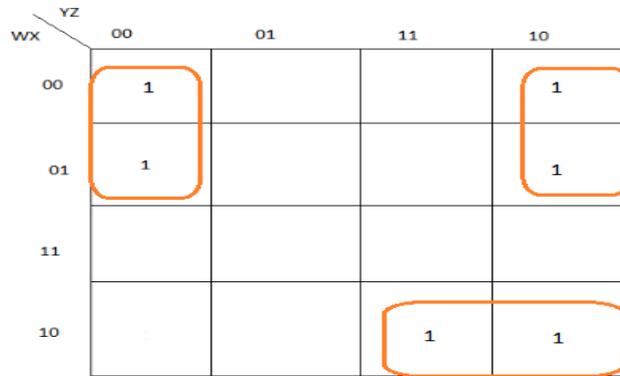
La función en su mínima expresión del Mapa de Karnaugh número 43 será:

$$\overline{W}\overline{Z} + W\overline{X}\overline{Y}Z$$



Ejemplo: Sea la función SOP siguiente mapeando los términos y obteniendo la mínima expresión

$$\overline{W}\overline{X}\overline{Y}\overline{Z} + \overline{W}X\overline{Y}\overline{Z} + \overline{W}\overline{X}Y\overline{Z} + \overline{W}X\overline{Y}Z + W\overline{X}Y\overline{Z} + W\overline{X}\overline{Y}Z$$



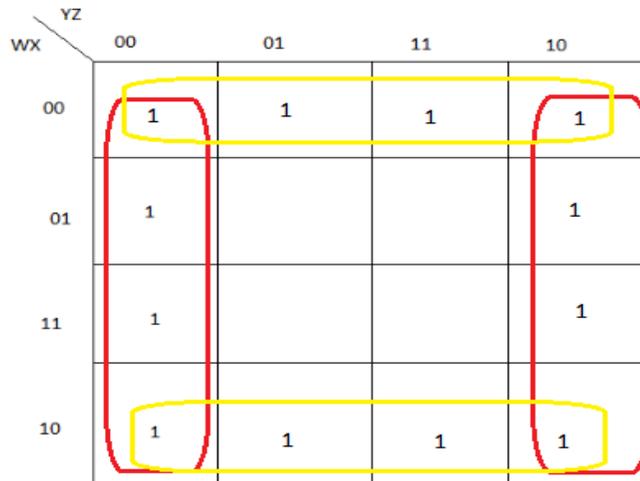
Mapa de Karnaugh 43 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh número 44 es

$$\overline{W} \overline{Z} + W \overline{X} Y$$

Mapee los términos siguientes de la función SOP y obteniendo la mínima expresión:

$$\overline{W} \overline{X} \overline{Y} \overline{Z} + \overline{W} X \overline{Y} \overline{Z} + W X \overline{Y} \overline{Z} + W \overline{X} \overline{Y} \overline{Z} + W \overline{X} \overline{Y} Z + \overline{W} \overline{X} Y Z + W \overline{X} Y Z + \overline{W} \overline{X} Y Z + \overline{W} X Y \overline{Z} + W X Y \overline{Z} + W \overline{X} Y \overline{Z}$$



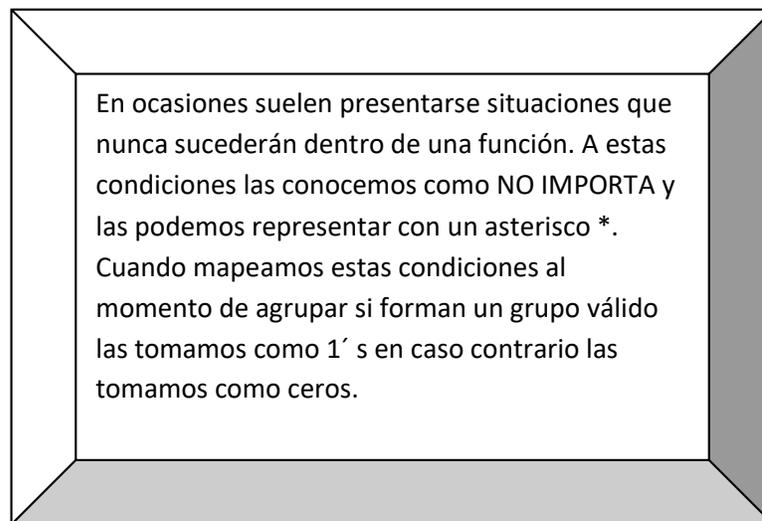
Mapa de Karnaugh 44 Agrupación de términos

Nota: En el ejemplo anterior la agrupación se realizó empleando el símil con la hoja de papel que se mencionó anteriormente.

La mínima expresión para el Mapa de Karnaugh 45 será:

$$\overline{X} + \overline{Z}$$

CONDICIONES NO IMPORTA



Cuando se trabaja con problemas en diseño digital al construir la tabla de verdad podría pensarse que la función de salida solo puede tener dos posibles valores 0 o 1; es decir un estado bajo o alto, activo o inactivo.

No obstante, puede ser que a la función no se le pueda asignar ningún de las dos entradas antes citadas por que no cumple con ninguna de esas condiciones, es decir no está en alto o bajo, es decir puede ser que las condiciones de entrada arrojen un resultado que no se utiliza nunca.

Por ejemplo:

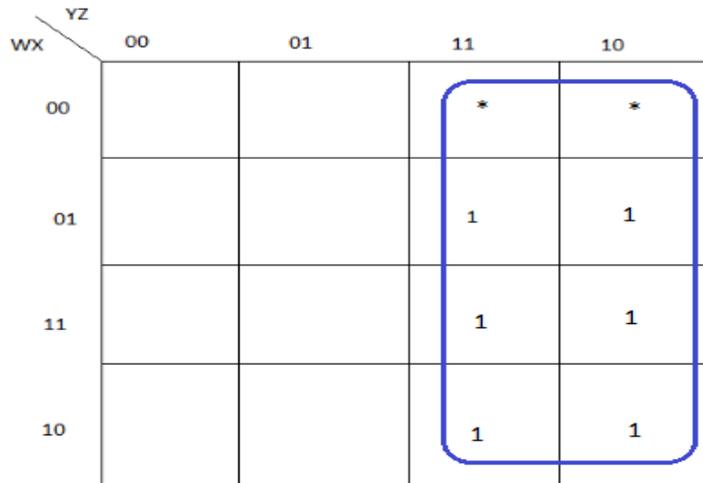
Se tiene un decodificador BCD a 7 segmentos

De las 16 posibles combinaciones de entrada solo se utilizarán 10 las otras 6 nunca ocurren ya que tales números no pueden representarse en el display 7 segmentos. Algo que podría usarse, pero sería por convención sería que los 6 dígitos representados se representaran mediante el sistema hexadecimal es decir con letras. Pero para este caso diremos que solo se puede representar los números del 0 al 9.

La importancia de utilizar don't care o condiciones no importa radica en el hecho de que permiten que la simplificación sea más fácil y con un mayor número de términos reducidos.

A	B	C	D	S
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	*
1	0	1	1	*
1	1	0	0	*
1	1	0	1	*
1	1	1	0	*
1	1	1	1	*

Observe los siguientes ejemplos:

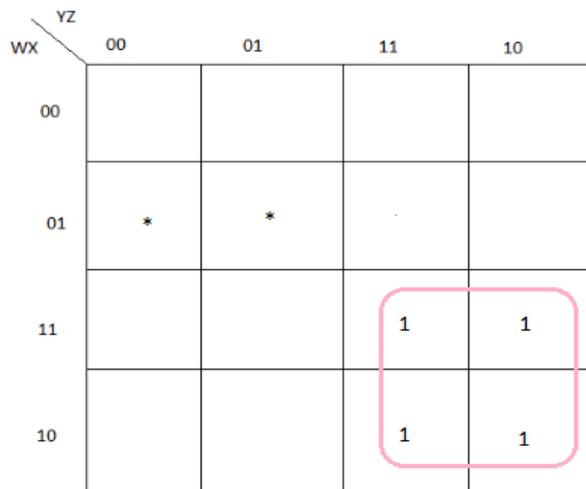


Mapa de Karnaugh 45 agrupación de términos

En este caso conviene tomar los * como 1's

Así la mínima expresión quedaría como: Y

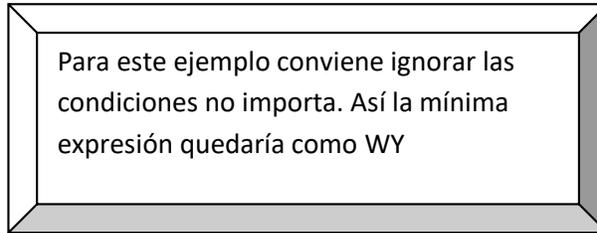
Ejemplo dos de agrupamiento de condiciones no importa.



Mapa de Karnaugh 46 Agrupación de términos

La agrupación mínima para el mapa de Karnaugh de la figura 46 es

WY



Como vimos en un principio los mapas de Karnaugh ayudan a simplificar funciones booleanas, vimos como emplearlos para funciones SOP a continuación veremos cómo usarlos para funciones POS.

Analice los siguientes ejercicios:

Nota:

La metodología para la lectura de los ceros es:

Expresar la variable en términos suma de productos donde cada variable se escribirá en el estado contrario a como se realiza la lectura en el mapa de Karnaugh por ejemplo si la lectura indica que la variable X esta en alto se expresaría como \bar{X} en la expresión SOP.

Como se muestra en el siguiente ejercicio.

Ejercicio 1D: Sea la siguiente función de tres variables:

$$(X + Y + Z)(X + \bar{Y} + \bar{Z})(\bar{X} + \bar{Y} + \bar{Z})$$

Obtenemos los ceros de la función:

$$(0\ 0\ 0)\ (0\ 1\ 1)\ (1\ 1\ 1)$$

Y los mapeamos a continuación:

		z
xy	0	1
00	0	
01		0
11		0
10		

Mapa de Karnaugh 47 Agrupación de términos

Función POS

Los ceros representan maxterminos.

Si se toman en cuenta los ceros se está obteniendo la inversa de la función solo que en la forma POS (suma de productos)

Por ejemplo, en el caso anterior la lectura de los ceros para el mintermino 0, se tomaría como altos que se suman así por ejemplo se obtendría $X+Y+Z$

Y para la siguiente reducción se obtendría también una suma de productos con los valores contrarios a como aparecen en la tabla, es decir en lugar de tomar YZ que sería un producto; se toman los valores contrarios $\bar{Y} + \bar{Z}$.

La mínima expresión será:

$$(X + Y + Z)(\bar{Y} + \bar{Z})$$

Cuando la función POS se obtiene de la tabla de verdad, es decir sí que haya ninguna reducción se dice que está en forma canónica. Si la Función en cambio se toma de un mapa de Karnaugh, es decir que ha pasado por un proceso de reducción se dice que está en su forma estándar.

Ejemplo: En base a la siguiente función mapee los términos y obtenga la mínima expresión:

$$(\overline{W} + X + Y + \overline{Z})(W + \overline{X} + \overline{Y} + Z)(\overline{W} + \overline{X} + \overline{Y} + Z)$$

Obteniendo los ceros de la función:

$$(1\ 0\ 0\ 1)\ (0\ 1\ 1\ 0)\ (1\ 1\ 1\ 0)$$

wx \ yz	00	01	11	10
00				
01				0
11				0
10		0		

Mapa de Karnaugh 4x4 Agrupación de términos

La mínima expresión será:

$$(\overline{X} + \overline{Y} + Z)(\overline{W} + X + Y + \overline{Z})$$

Cuando un término POS no está es su forma estándar se procede a expandirlo.

Analice el siguiente ejemplo:

Sea la siguiente función:

$$(W + \bar{X})(W + X + \bar{Y} + \bar{Z})(\bar{Y} + \bar{Z} + X)$$

Tenemos que los términos primero y tercero no están en su forma estándar ya que no contienen todas las variables así que procedemos a expandirlos como:

$$(W + \bar{X})$$

$$(W + \bar{X} + Y)(W + \bar{X} + \bar{Y})$$

$$(W + \bar{X} + Y + Z)(W + \bar{X} + Y + \bar{Z})(W + \bar{X} + \bar{Y} + Z)(W + \bar{X} + \bar{Y} + \bar{Z})$$

Para el tercer término:

$$(\bar{Y} + \bar{Z} + X)$$

Expandimos el término y nos quedaría:

$$(\bar{Y} + \bar{Z} + X + W)(\bar{Y} + \bar{Z} + X + \bar{W})$$

Finalmente, la función SOP expandida quedaría como:

$$(W + \bar{X} + Y + Z)(W + \bar{X} + Y + \bar{Z})(W + \bar{X} + \bar{Y} + Z)(W + \bar{X} + \bar{Y} + \bar{Z})(W + X + \bar{Y} + \bar{Z})(\bar{W} + X + \bar{Y} + \bar{Z})$$

A continuación, obtenga los ceros de la función y mapéela.

$$(0100) (0101) (0110) (0111) (0011) (1011)$$

Se deja como ejercicio mapear la función.

Obtenga la mínima expresión en base al siguiente mapa

		YZ			
WX		00	01	11	10
00					
01		0	0	0	0
11		0	0		
10					

Mapa de Karnaugh 49 Agrupación de términos

Los grupos se pueden agrupar debido a que se cumple la propiedad de continuidad, es decir que entre celdas adyacentes solo cambia un bit.

La mínima expresión de la función quedaría como:

$$(W + \bar{X})(\bar{X} + Y)$$

La agrupación de color rojo corresponde a la primera expresión

La agrupación de color anaranjado corresponde a la segunda expresión

¿Cómo obtener una función SOP de una función POS y viceversa?

Primeramente, se obtienen los ceros de la función POS, las celdas que no contienen 0's contienen 1's y estas se agrupan para obtener la función SOP.

Analice el siguiente ejemplo:

Sea la función POS siguiente. Mapearla y posteriormente obtener la función SOP.

$$(W + X + \bar{Y} + \bar{Z})(\bar{W} + \bar{X} + Y + Z)(W + X + Y + Z)(\bar{W} + \bar{X} + \bar{Y} + Z)$$

Primero se obtienen los ceros de la función:

$$(0011) (1100) (0000) (1110)$$

Mapeando los términos:

		YZ			
		00	01	11	10
wx	00	0		0	
	01				
	11	0			0
	10				

Mapa de Karnaugh 50 Agrupación de términos

La mínima expresión POS será:

$$(W + X + Y + Z)(W + X + \bar{Y} + \bar{Z})(\bar{W} + \bar{X} + Y + Z)(\bar{W} + \bar{X} + \bar{Y} + Z)$$

Utilizando algebra para reducir la función se tendrá:

$$(W + X + Y + Z)(W + X + \bar{Y} + \bar{Z})(\bar{W} + \bar{X} + Y + Z)(\bar{W} + \bar{X} + \bar{Y} + Z)$$

Realizando la multiplicación de los primeros dos paréntesis:

$$(W + X + Y + Z)(W + X + \bar{Y} + Z)$$

$$\begin{aligned}
 & WW + WX + W\bar{Y} + W\bar{Z} + XW + XX + X\bar{Y} + X\bar{Z} + YW + YX \\
 & + YY + Y\bar{Z} + ZW + ZX + Z\bar{Y} + ZZ
 \end{aligned}$$

Aplicando propiedades y reduciendo términos:

$$\begin{aligned}
 & W + WX + X + W\bar{Y} + W\bar{Z} + XW + X\bar{Y} + X\bar{Z} + YW + YX + Y\bar{Z} \\
 & + ZW + ZX + Z\bar{Y}
 \end{aligned}$$

$$\begin{aligned}
 & W(1 + X) + X(1 + W) + W(Y + \bar{Y}) + W(Z + \bar{Z}) + X\bar{Y} + X\bar{Z} \\
 & + YX + Y\bar{Z} + ZX + Z\bar{Y}
 \end{aligned}$$

$$W + X + W + W + X\bar{Y} + X\bar{Z} + YX + Y\bar{Z} + ZX + Z\bar{Y}$$

$$W + X(1 + \bar{Y}) + X\bar{Z} + YX + Y\bar{Z} + ZX + Z\bar{Y}$$

$$W + X + X\bar{Z} + YX + Y\bar{Z} + ZX + Z\bar{Y}$$

$$W + X(1 + \bar{Z}) + YX + Y\bar{Z} + ZX + Z\bar{Y}$$

$$W + X + YX + Y\bar{Z} + ZX + Z\bar{Y}$$

$$W + X(1 + Y) + Y\bar{Z} + ZX + Z\bar{Y}$$

$$W + X + Y\bar{Z} + ZX + Z\bar{Y}$$

$$W + X(1 + Z) + Y\bar{Z} + Z\bar{Y}$$

$$W + X + Y\bar{Z} + Z\bar{Y}$$

El primer resultado será:

$$W + X + Y\bar{Z} + Z\bar{Y}$$

Realizando la segunda parte de la multiplicación.

$$(\bar{W} + \bar{X} + Y + Z)(\bar{W} + \bar{X} + \bar{Y} + Z)$$

$$\begin{aligned}
 & \bar{W}\bar{W} + \bar{W}\bar{X} + \bar{W}\bar{Y} + \bar{W}Z + \bar{W}\bar{X} + \bar{X}\bar{X} + \bar{X}\bar{Y} + \bar{X}Z + Y\bar{W} + Y\bar{X} \\
 & + Y\bar{Y} + YZ + Z\bar{W} + Z\bar{X} + Z\bar{Y} + ZZ
 \end{aligned}$$

$$\begin{aligned}
 & \bar{W} + \bar{W}\bar{X} + \bar{W}\bar{Y} + \bar{W}Z + \bar{W}\bar{X} + \bar{X} + \bar{X}\bar{Y} + \bar{X}Z + Y\bar{W} + Y\bar{X} + YZ \\
 & + Z\bar{W} + Z\bar{X} + Z\bar{Y} + Z
 \end{aligned}$$

$$\begin{aligned}
 & \bar{W}(1 + \bar{X}) + \bar{W}(\bar{Y} + Y) + \bar{X}(Y + \bar{Y}) + Z(\bar{X} + 1) + Z(Y + \bar{Y}) \\
 & + \bar{X}(\bar{W} + 1) + WZ
 \end{aligned}$$

$$\bar{W} + \bar{W} + \bar{X} + Z + Z + \bar{X} + \bar{W}Z$$

El segundo resultado será:

$$\bar{W} + \bar{X} + Z$$

Multiplicando los resultados:

$$(W + X + Z\bar{Y} + Y\bar{Z})(\bar{W} + \bar{X} + Z)$$

$$W\bar{W} + W\bar{X} + WZ + X\bar{W} + X\bar{X} + XZ + Z\bar{Y}\bar{W} + Z\bar{Y}X + Z\bar{Y} \\ + Y\bar{Z}\bar{W} + Y\bar{Z}\bar{X} + Y\bar{Z}Z$$

REDUCIENDO

$$W\bar{X} + WZ + X\bar{W} + XZ + Z\bar{Y}\bar{W} + Z\bar{Y}X + Z\bar{Y} + Y\bar{Z}\bar{W} + Y\bar{Z}\bar{X}$$

Reduciendo términos

Diviendo en dos la reducción de términos

$$Z\bar{Y} + Z\bar{Y}X = Z\bar{Y}(1 + X)$$

$$Z\bar{Y}(\bar{W} + 1)$$

$$WZ + \bar{W}\bar{Z}Y$$

$$\bar{W} + WZ$$

Segunda reducción

$$WZ + Z\bar{Y}\bar{W} + Y\bar{Z}\bar{W}$$

$$Z(W + \bar{W}\bar{Y}) + Y\bar{Z}\bar{W}$$

$$WZ + Z\bar{Y}\bar{W} + Y\bar{Z}\bar{W}$$

$$Z(W + \bar{W}\bar{Y}) + Y\bar{Z}\bar{W}$$

$$ZW + Z\bar{Y} + Y\bar{Z}\bar{W}$$

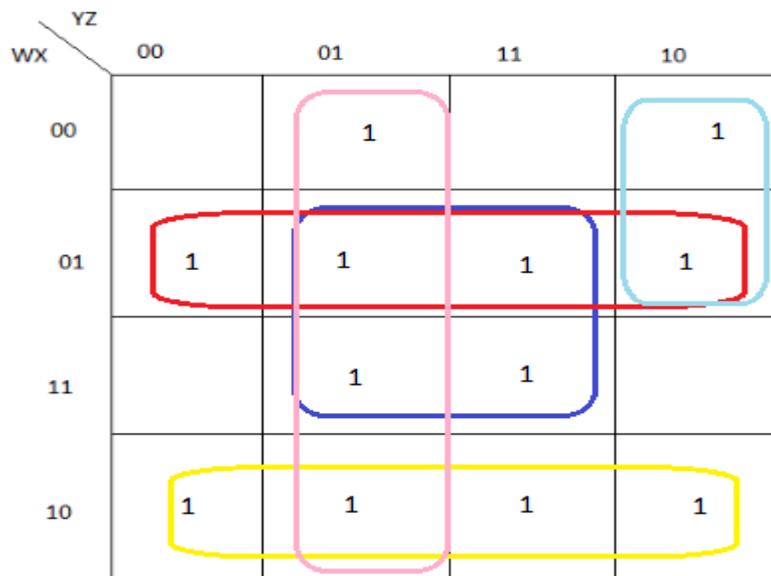
$$Y(ZW + \bar{Z}\bar{W})$$

El término final de la reducción será:

$$W\bar{X} + X\bar{W} + XZ + Z\bar{Y} + Y\bar{Z}\bar{X}$$

Nota: Es importante recordar que si se agrupan los unos en las casillas faltantes donde no hay ceros se obtiene una función SOP equivalente a la función POS de la cual se partió

Obteniendo la función SOP en el mapa de karnaugh.

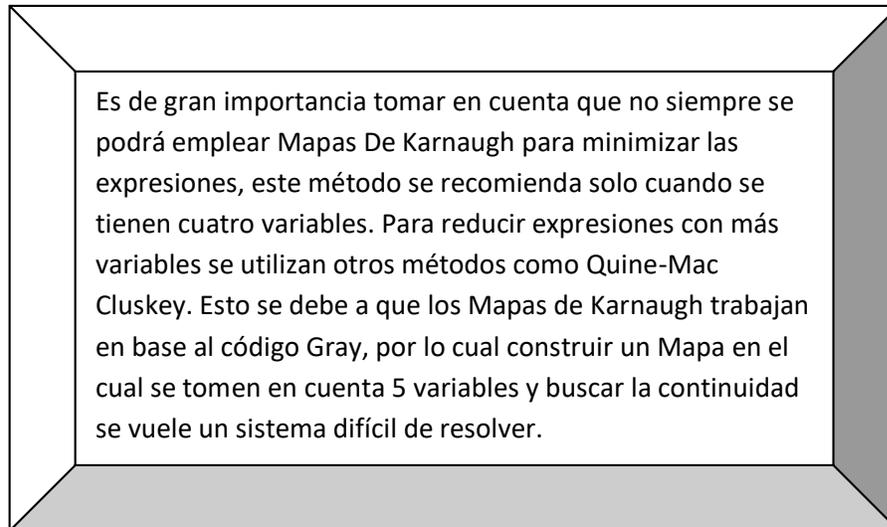


Mapa de Karnaugh 51 Agrupación de término

La función SOP en su mínima expresión quedaría como:

$$XZ + W\bar{X} + \bar{W}X + \bar{Y}Z + \bar{W}Y\bar{Z}$$

Al comparar el resultado por reducción algebraica y por mapa de Karnaugh se puede ver que se llega al mismo resultado.



10.1 Método de Quine-McCluskey

Ejemplo:

Sea la siguiente función:

$$ABC\bar{D}E + A\bar{B}CDE + A\bar{B}\bar{C}DE + \bar{A}BC\bar{D}\bar{E} + \bar{A}BCD\bar{E} + \bar{A}BCDE + \\ A\bar{B}\bar{C}DE + \bar{A}\bar{B}\bar{C}\bar{D}\bar{E} + ABCD\bar{E}$$

Primero obtenemos los min términos:

	MINTÉRMINO
$A B C \bar{D} E$ 1 1 1 0 1	29
$A \bar{B} C D E$ 1 0 1 1 1	23
$A \bar{B} \bar{C} D E$ 1 0 0 1 1	19
$\bar{A} B C \bar{D} \bar{E}$ 0 1 1 0 0	12
$\bar{A} B C D \bar{E}$ 0 1 1 1 0	14
$\bar{A} B C D E$ 0 1 1 1 1	15
$A B \bar{C} D E$ 1 1 0 1 1	27
$\bar{A} \bar{B} \bar{C} \bar{D} \bar{E}$ 0 0 0 0 0	0
$A B C D \bar{E}$ 1 1 1 1 0	30

Los mintérminos serán:

0,12,14,15,19,23,27,29,30

En base a los mintérminos obtenidos hacemos grupos estos serán grupos en los que no haya ningún 1, en los que haya un uno, en los que haya dos unos, en los que haya tres unos, en los que haya cuatro unos.

0's	Grupo 1	0	A B C D E 0 0 0 0 0
un uno	Grupo 2	-	-----
dos unos	Grupo 3	12	0 1 1 0 0
tres unos	Grupo 4	14	0 1 1 1 0
		19	1 0 0 1 1
cuatro unos	Grupo 5	15	0 1 1 1 1
		23	1 0 1 1 1
		27	1 1 0 1 1
		29	1 1 1 0 1
		30	1 1 1 1 0

Figura 48 Formación de grupos de 1 del ejemplo 1F

Agrupando los términos que difieren en una variable conocidos también como los esenciales primos.

Y posteriormente se coloca un guion en el término en el cual difieren como muestra la siguiente tabla.

12 14	A B C D E 0 1 1 0 0 0 1 1 1 0	011-0
14 15	0 1 1 1 0 0 1 1 1 1	0111-
14 30	0 1 1 1 0 1 1 1 1 0	-1110
19 23	1 0 0 1 1 1 0 1 1 1	10-11
19 27	1 0 0 1 1 1 1 0 1 1	1-011

Tabla 41 Tabla donde se muestra la colocación del guion en el término que difieren

A continuación, se construye la siguiente tabla:

	12	14	15	30	19	23	27
(12,14)	*	*					
(14,15)		*	*				
(14,30)		*		*			
(19,23)					*	*	
(19,27)					*		*

Tabla 42 Tabla para obtener la mínima expresión de la función 1F

La función en su mínima expresión quedaría como:

$$\bar{A} B C \bar{E} + \bar{A} B C D + B C D \bar{E} + A \bar{B} D E + A \bar{C} D E$$

Ejemplo 2F:

Sea la función:

$$\bar{A}\bar{B}\bar{C}DE + \bar{A}\bar{B}C\bar{D}\bar{E} + \bar{A}B\bar{C}D\bar{E} + \bar{A}BCD\bar{E} + \bar{A}BCDE + ABCDE + \bar{A}\bar{B}\bar{C}\bar{D}\bar{E}$$

Obtenemos los min términos de la función:

$\bar{A}\bar{B}\bar{C}DE$ 0 0 0 1 1	MINTÉRMINO 3	A B C D E 0 0 0 1 1
$\bar{A}\bar{B}C\bar{D}\bar{E}$ 0 0 1 0 0	4	0 0 1 0 0
$A\bar{B}\bar{C}D\bar{E}$ 1 0 0 1 0	18	1 0 0 1 0
$\bar{A}B C D \bar{E}$ 0 1 1 1 0	14	0 1 1 1 0
$\bar{A}\bar{B}CDE$ 0 0 1 1 1	7	0 0 1 1 1
$ABCDE$ 1 1 1 1 1	31	1 1 1 1 1
$\bar{A}\bar{B}\bar{C}\bar{D}\bar{E}$	0	0 0 0 0 0

Figura 49 Tabla de los mintérminos de la función del ejercicio 2F

Formamos grupos que no contengan ningún 1, que contengan un uno, dos unos, tres unos, cuatro unos y cinco unos.

		MINTERMINO	A B C D E
0 1's	Grupo 1	0	0 0 0 0 0
1 1's	Grupo 2	4	0 0 1 0 0
2 1's	Grupo 3	3 18	0 0 0 1 1 1 0 0 1 0
3 1's	Grupo 4	7 14	0 0 1 1 1 0 1 1 1 0
4 1's	Grupo 5	-	-----
5 1's	Grupo 6	31	1 1 1 1 1

Tabla 43 Tabla de los grupos que contienen uno, dos, tres y cinco unos

Formando pares que difieran en un bit, es decir obteniendo los términos esenciales o implicantes primos:

	A B C D E	MINTERMINO	A B C D E
0 4	0 0 0 0 0 0 0 1 0 0	(0,4)	0 0 - 0 0
3 7	0 0 0 1 1 0 0 1 1 1	(3,7)	0 0 - 1 1

Tabla 44 Tabla de los términos esenciales o implicantes primos

La mínima función será:

$$F(A, B, C, D, E) =$$

$$\overline{A} \overline{B} \overline{D} \overline{E} + \overline{A} \overline{B} D E + A \overline{B} \overline{C} D \overline{E} + \overline{A} B C D \overline{E} + A B C D E$$

11. SUMA DE PRODUCTOS (SOP) Y PRODUCTO DE SUMAS (POS)

Como vera en el desarrollo de este tema podrá ver que todas las expresiones booleanas, sin importar su forma pueden convertirse en la forma suma de productos o producto de semas.

11.1 SUMA DE PRODUCTOS (SOP)

La forma suma de productos es la misma de literales que se están multiplicando; a continuación, unos ejemplos:

$$X Y + Z \overline{X}$$

$$Y X Z + A B C$$

$$\overline{X} \overline{Z} + X Y Z + Z X + \overline{Y}$$

\overline{Y} NOTA: En la última expresión

Se está multiplicando por 1.

Para llegar a cualquiera de las funciones es necesario aplicar previamente los teoremas De Morgan

Analice los siguientes ejercicios y vea como se obtienen las funciones SOP después de aplicar el álgebra booleana a las expresiones originales.

$$1) \overline{A} \overline{B} (C + D) + \overline{Z} (\overline{C} \overline{D}) + D$$

Desarrollo

Aplicando Teorema de Morgan

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

Aplicando la propiedad distributiva

$$\bar{A}C + \bar{A}D + \bar{B}C + \bar{B}D + \bar{Z}\bar{C} + \bar{Z}\bar{D} + D$$

$$\bar{A}C + \bar{A}D + \bar{B}C + \bar{B}D + \bar{Z}\bar{C} + \bar{Z}\bar{D} + D$$

Tomando en consideración:

$$D(\bar{B} + 1) = D$$

$$\bar{A}C + \bar{A}D + \bar{B}C + \bar{Z}\bar{C} + \bar{Z}\bar{D} + D$$

Tomando en consideración que:

$$D(\bar{A} + 1) = D$$

$$\bar{A}C + \bar{B}C + \bar{Z}\bar{C} + \bar{Z}\bar{D} + D$$

EJEMPLO:

$$2) A(1 + CD) + \bar{A}(A + B) + \bar{D}\bar{F}$$

Aplicando la propiedad distributiva y la regla 8

$$A + ACD + \bar{A}A + \bar{A}B + \bar{D} + \bar{F}$$

Aplicando la regla 11

$$A + \bar{A}B + ACD + \bar{D} + \bar{F}$$

$$A + B + ACD + \bar{D} + \bar{F}$$

$$3) C(\bar{D}\bar{E} + F) + F(A + \bar{B})$$

Aplicando Teorema de Morgan

$$C(\bar{D} + \bar{E} + F) + F(A + \bar{B})$$

Aplicando la propiedad distributiva

$$C\bar{D} + C\bar{E} + CF + AF + F\bar{B}$$

$$4) A(ZD) + B(\bar{F} + \bar{A})$$

Aplicando teorema de Morgan

$$AZD + B(\bar{F}\bar{A})$$

$$A Z D + B \bar{F} \bar{A}$$

$$5) Z + D + F (A + \bar{A}) + X (\bar{Y} \bar{Z})$$

$$Z + D + A F + \bar{A} F + X (\bar{Y} + \bar{Z})$$

Aplicando la propiedad distributiva

$$Z + D + A F + \bar{A} F + X \bar{Y} + X \bar{Z}$$

Aplicando propiedad asociativa

$$Z + D + F (A + \bar{A}) + X \bar{Y} + X \bar{Z}$$

Aplicando la regla 6

$$Z + D + F + X \bar{Y} + X \bar{Z}$$

Esta es la forma más reducida para evitar que se eleven los costos.

Dentro de las expresiones SOP; esta lo que se conoce como expresión SOP estándar.

En una expresión SOP, cada término contiene todas las variables de la expresión se conoce como forma canónica de la función.

Analice los siguientes ejercicios:

$$A B + C D$$

El dominio de una función se refiere a todas las variables que contiene la expresión. Por lo tanto, el dominio de la función será A, B, C, D

$$(A + \bar{A}) \quad \text{Utilizamos la regla}$$

Que equivale a multiplicar por un 1 la expresión.

Es así como al analizar la expresión vemos que el primer término no contiene a la variable C y D

Por lo tanto, procedemos a realizar el desarrollo:

Ejemplos:

$$1) \quad A B (C + \overline{C}) = A B C + A B \overline{C}$$

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

Aplicando la propiedad distributiva

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

$$2) \quad C D (A + \overline{A}) = C D A + C D \overline{A}$$

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

Aplicando la propiedad distributiva

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

La expresión SOP estándar será:

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

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

11.2 LA FORMA POS

Estas expresiones consisten en realizar la multiplicación de la suma de variables.

Se muestra a continuación unos ejemplos:

$$(B + D + F) (A + \overline{B} + \overline{F})$$

$$(A + F) (B + D)$$

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

$$\overline{F} (D + A) \overline{D}$$

NOTA: Para llegar al resultado final es necesario primero aplicar los Teoremas De Morgan y el álgebra booleana.

Para convertir una función POS a una expresión POS estándar se toma en consideración la siguiente regla:

$$(A \bar{A}) = 0$$

Que es el equivalente de sumar un cero a la expresión:

Revise los siguientes ejercicios y analice como se aplica la teoría anterior.

$$1) (\bar{A} + B + \bar{D} + \bar{F})(C + \bar{B} + F)$$

El dominio de la función es:

A, B, C, D, F

En el primer término hace falta C

Por lo tanto:

$$\bar{A} + B + \bar{D} + \bar{F} + C \bar{C} =$$

$$(\bar{A} + B + \bar{D} + \bar{F} + C)(\bar{A} + B + \bar{D} + \bar{F} + \bar{C})$$

El siguiente término no contiene a las variables A y D

Procedemos a agregarlos

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

$$(C + \bar{B} + F + A)(C + \bar{B} + F + \bar{A})$$

Procediendo a agregar la variable D a cada término:

$$C + B + F + A + D \bar{D} =$$

$$(C + B + F + A + D)(C + B + F + A + \bar{D})$$

$$C + \bar{B} + F + \bar{A} + D \bar{D} =$$

$$(C + \bar{B} + F + \bar{A} + D)(C + \bar{B} + F + \bar{A} + \bar{D})$$

La forma POS de la expresión:

$$(\bar{A} + B + \bar{D} + \bar{F})(C + \bar{B} + F)$$

Será:

$$(\bar{A} + B + \bar{D} + \bar{F} + C)(\bar{A} + B + \bar{D} + \bar{F} + \bar{C})(C + B + F + A + D)(C + B + F + A + \bar{D})(C + \bar{B} + F + \bar{A} + D)(C + \bar{B} + F + \bar{A} + \bar{D})$$

Conversión de una expresión SOP a POS.

Analice el siguiente ejemplo:

$$A\bar{B}\bar{C} + \bar{B}C + \bar{A}BC$$

NOTA: La función que se presenta anteriormente no se encuentra en la forma estándar, para ello cada uno de los términos debe contener todas las variables de la función.

Primeramente el dominio de la función es ABC como podemos apreciar el segundo término es el no contiene la variable A.

Así que procedemos a agregárselo.

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

La expresión SOP canónica quedaría como:

$$A\bar{B}\bar{C} + A\bar{B}C + \bar{A}\bar{B}C + \bar{A}BC$$

Obteniendo los valores binarios de los términos:

$$110 + 101 + 001 + 011$$

Cuando se tienen 3 variables tenemos 8 posibles combinaciones

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Tabla 45 Tabla de verdad para la evaluación de la función.

Los minterminos correspondientes a esta tabla son:

$$m_1, m_3, m_5, m_6$$

Obteniendo la expresión POS

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

Obteniendo la función SOP

$$(A + B + \bar{C})(A + \bar{B} + \bar{C})(\bar{A} + B + \bar{C})(\bar{A} + \bar{B} + C)$$

12. Circuitos combinacionales

12.1 Análisis de circuitos combinacionales

Trabajando con lógica AND-OR

Analice los siguientes diagramas:

Ejemplo A:

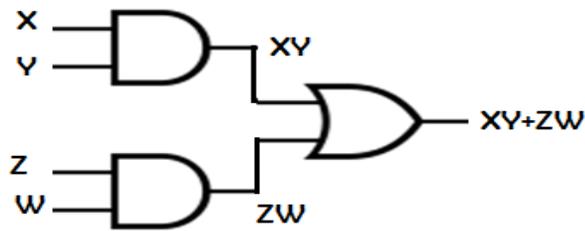


Figura 50 Implementación de la función lógica con compuertas

Ejemplo B:

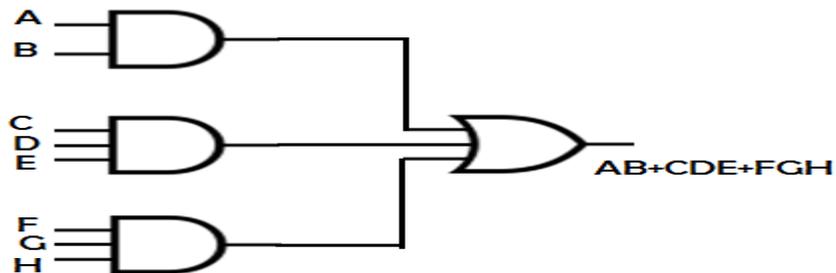


Figura 51 Implementación de la función lógica con compuertas

Ejemplo C:

Sea la función $ABC + DEFG + HIJKLM$

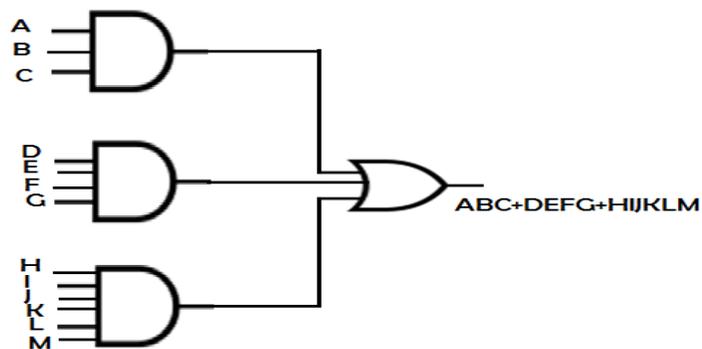


Figura 52 Implementación de la función lógica con compuertas

Como podrá haberse dado cuenta un circuito AND-OR está formado por N números de compuertas AND y una compuerta OR. Y cada compuerta AND se conforma por N número de entradas.

Una función SOP se implementa mediante un circuito AND-OR.

Para los ejemplos anteriores A, B y C diga que necesita cumplirse para que las salidas sean altas.

Ejemplo A:

Que X y Y sean altas

Que Z y W sean altas

Ó que todas sean altas

Ejemplo B

Que A y B sean altas

Que C, D y E sean altas

Que F, G y H sean altas

Ó que todas sean altas

Ejemplo C

Que A, B y C sean altas

Que D, E, F y G sean altas

Que H, I, J, K, L y M sean altas

Ó que todas sean altas

13. Escalas de Integración MSI

El avance en el desarrollo de la tecnología ha hecho posible que el tamaño de las compuertas lógicas haya disminuido a tal grado que hoy es posible montar una gran cantidad de ellas en un área muy pequeña.

Antes de continuar veremos la definición de chip:

El chip es un circuito electrónico de material semiconductor especialmente sencillo en forma de cubo

minúsculo que combinado con otros componentes forma un sistema integrado más complejo y realiza una función electrónica específica.

El término MSI significa Mediana Escala de Integración y es utilizado para referirse a los circuitos integrados o chips que tienen una capacidad de contener entre 10 y 500 transistores.



Figura 53 Imagen de un chip

Muchas han sido las ventajas de los nuevos circuitos que han salido al mercado.

Entre estas se encuentran:

La disminución en los costes de producción

La reducción de espacio, es decir cada vez se pueden colocar más componentes en una placa por cm cuadrado

La confiabilidad en las operaciones ha mejorado

Se ha dado una mejora en los tiempos de respuesta

Se ha dado una mejora en la portabilidad de los circuitos

Es importante tener presente el número de compuertas que se montan en un chip ya que en base a esto podrá clasificar el chip en un tipo de escala de integración.

Los circuitos integrados se clasifican principalmente en cuatro familias:

SSI o pequeña escala de integración.

En esta familia se encuentran los circuitos de funciones lógicas básicas.

El número de componentes en un circuito es de 100 y puede contener como máximo hasta 10 compuertas lógicas, los componentes son transistores en base a la cantidad de estos se dice es mediana o gran escala.

MSI o Mediana escala de integración

En esta categoría se encuentran los codificadores, multiplexores, contadores, registros, flip-flops.

El número de componentes por chip está en un rango de 100 y 1000.

El número máximo de compuertas es de 100



Figura 54 Circuitos Integrados

Los circuitos llamados MSI se introdujeron en la década de los 60.

Los ordenadores del año de 1970 implementan estos integrados.

El trabajo en el ensamblaje era menos costoso y difícil.

Son un poco más costosos que los SSI.

Como se mencionaba anteriormente ejemplos de estos circuitos son los: codificadores, multiplexores, contadores, registros y FF.

A continuación, se mencionan algunos.

Circuitos de mediana escala

El integrado 74LS151TTL el cual es un multiplexor de 8 entradas y selector de datos.

Descripción.

El SN74LS151N es un selector de datos o multiplexor de 8 a 1 línea con decodificación binaria en el chip para seleccionar la fuente de datos deseada.

El LS151 selecciona una de las ocho fuentes de datos.

Un nivel alto a la salida W alta y a la salida Y (según corresponda) a bajas.

Sus aplicaciones son en la industria, comunicaciones y en redes.



Figura 55 circuito integrado 74LS148

El circuito integrado SN74LS148N es un codificador de prioridad de 8 a 3 líneas cuya principal característica de descodificación es garantizar que solo se codifique la línea de datos de mayor orden.

El circuito integrado SN74S138 es un Decodificador/Demultiplexor de 3-8 líneas. Estos circuitos son empleados en memorias de alta rapidez y sistemas cuya capacidad de respuesta sea rápida. El uso de estos circuitos reduce el diseño y hace que se requieran un menor número de compuertas.

El circuito SN54150 es un Selector de datos/Multiplexor.

El circuito SN54190 y el SN54LS191, el primero es un contador BCD y el segundo es un contador binario de cuatro bits.

Las operaciones de estos circuitos están sincronizadas por FF los cuales contienen reloj así que las salidas tienen cambios coincidentes, evitando con ello el uso o empleo de respuestas asociadas a contadores asíncronos.

Otro circuito que podemos encontrar es el LS147 el cual codifica nueve líneas de datos para código BCD de cuatro líneas (8-4-2-1). La principal condición implícita de cero decimales la cual no requiere ninguna condición de entrada puesto que el cero está codificado cuando las nueve líneas de datos están en un nivel lógico alto.

Ejemplos más comunes de decodificador es el BCD a 7 segmentos.

Se tienen dos circuitos en el mercado son:

El 74LS48 el cual es cátodo común

Sea el circuito 74LS48 su tabla de verdad es:

N	Entradas	a	B	C	D	E	f	G
0	0000	1	1	1	1	1	1	0
1	0001	0	1	1	0	0	0	0
2	0010	1	1	0	1	1	0	1
3	0011	1	1	1	1	0	0	1
4	0100	0	1	1	0	0	1	1
5	0101	1	0	1	1	0	1	1
6	0110	0	0	1	1	1	1	1
7	0111	1	1	1	0	0	0	0
8	1000	1	1	1	1	1	1	1
9	1001	1	1	1	0	0	1	1

Figura 56 Tabla de verdad del circuito 74LS48 cátodo común

El 74LS47 el cual es ánodo común

La tabla de verdad será:

D	C	B	A	a	B	C	D	E	F	G
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	0	0
1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0

Figura 57 Tabla de verdad del circuito 74LS47 ánodo común

a) LSI o gran escala de Integración

Estos circuitos se caracterizan por llevar acabo funciones lógicas muy complejas.

Dentro de esta categoría podemos encontrar los dispositivos de lógica programable, como memorias, microprocesadores.

Para este el número de componentes está en el rango entre 1000 y 10000. En donde el circuito puede contener como máximo 1000 compuertas lógicas.

VLSI o Muy gran escala de Integración

A esta categoría pertenecen los circuitos que tienen más de 10 000 000 de componente.

Y puede tener 1000 millones de transistores.

GRÁFICA

REPRESENTACIÓN GRÁFICA DE LA EVOLUCIÓN DE LA CAPACIDAD DE INTEGRACIÓN.

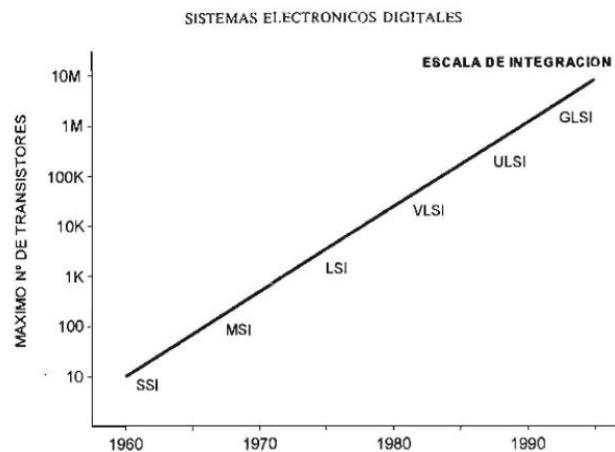


Figura 58 Representación gráfica de la evolución de la capacidad de integración.

EVOLUCIÓN DE LA CAPACIDAD DE INTEGRACIÓN.

Nivel de integración	Nº de componentes	Nº de puertas
Pequeña escala de integración (<u>SSI</u>)	10 a 100	1 a 12
Mediana escala de integración (<u>MSI</u>)	100 a 1000	12 a 100
Gran escala de integración (<u>LSI</u>)	1.000 a 10.000	100 a 1.000
Muy gran escala de integración (<u>VLSI</u>)	10.000 a 100.000	1.000 a 10.000
Ultra gran escala de integración (<u>ULSI</u>)	100.000 a 1.000.000	10.000 a 100.000
Giga gran escala de integración (<u>GLSI</u>)	Más de 1.000.000	Más de 100.000

Tabla 46 Evolución d la capacidad de integración

Observaciones

A mayor escala de Integración se obtienen sistemas funcionales cada vez más complejos, que permiten un desarrollo creciente en la tecnología, en donde se mejoran muchos aspectos esenciales como son el diseño, la confiabilidad, la respuesta en el tiempo, por mencionar algunas.

14. Codificadores, Decodificadores, Multiplexores, Demultiplexores, Comparadores y Generadores de paridad

14.1 CODIFICADORES

Dentro de la gama de circuitos combinacionales entra el codificador. El codificador se caracteriza por ser un circuito que acepta en su entrada un dígito en forma digital u octal, es decir acepta un dígito que puede ser: 0,1,2,3,4,5,6,7,8,9 y del

0 al 7 si es un sistema octal y a la salida expresa dicha cantidad en forma binaria.

Un codificador tiene 2^n entradas o puede ser menos líneas de entrada y cuenta con n líneas de salida.

De las líneas de salida se genera el código binario para las 2^n variables de entrada.

La tarea de un codificador es detectar un número en su forma decimal y expresarlo en forma binaria en las salidas, solo puede detectar una línea a la vez.

No obstante, se tienen codificadores con prioridad es decir se puede elegir varias líneas de entrada a la vez, por ejemplo 3, 5, 1 y el 7 de todas estas líneas de entrada en este tipo de circuitos solo seleccionara el de mayor prioridad que en este caso es el que tiene mayor peso es decir será el número 7, como se explicará más adelante.

Tenemos codificadores de:

4x2

8x3

16x4

32x5

Uno de los ejemplos más sencillos de estos circuitos es el codificador de octal a binario este acepta 8 líneas de entrada y produce un código de salida de 3 bits el cual corresponde a la entrada que se activó.

14.2 Simulación de un codificador de 4 a 2

```
library ieee;
use ieee.std_logic_1164.all;

entity codificador_4_2 is

port (a: in std_logic_vector (3 downto 0);
      b: out std_logic_vector (1 downto 0);
      c: out std_logic
```

```
);
```

```
end entity codificador_4_2;
```

```
architecture whenelse of codificador_4_2 is begin
```

```
b<= "11" when a (3) = '1' else
```

```
  "10" when a (2) = '1' else
```

```
  "01" when a (1) = '1' else
```

```
  "00";
```

```
c<=a (3) or a (2) or a (1) or a (0);
```

```
end architecture whenelse;
```

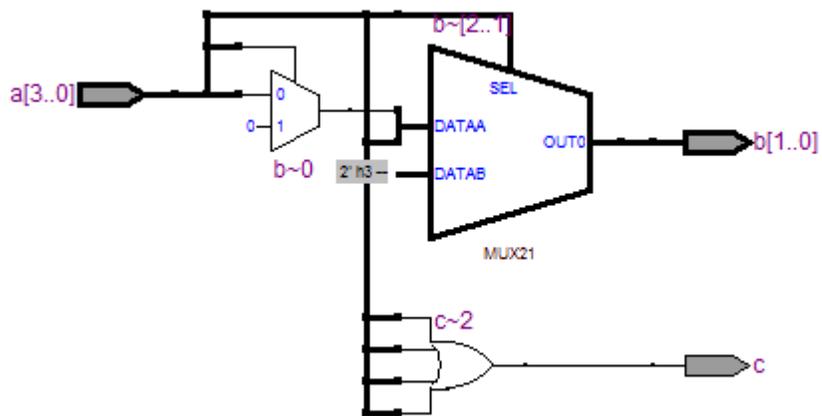
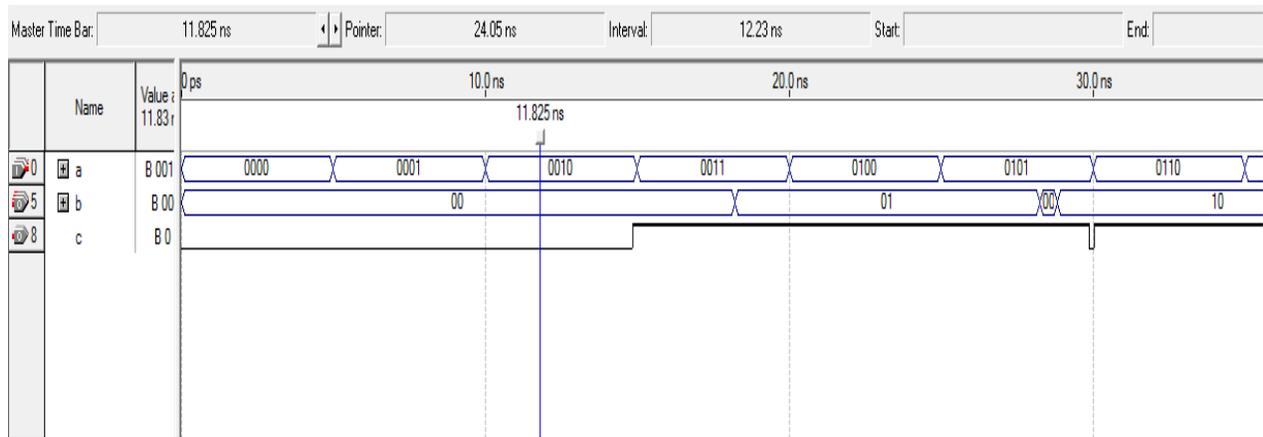
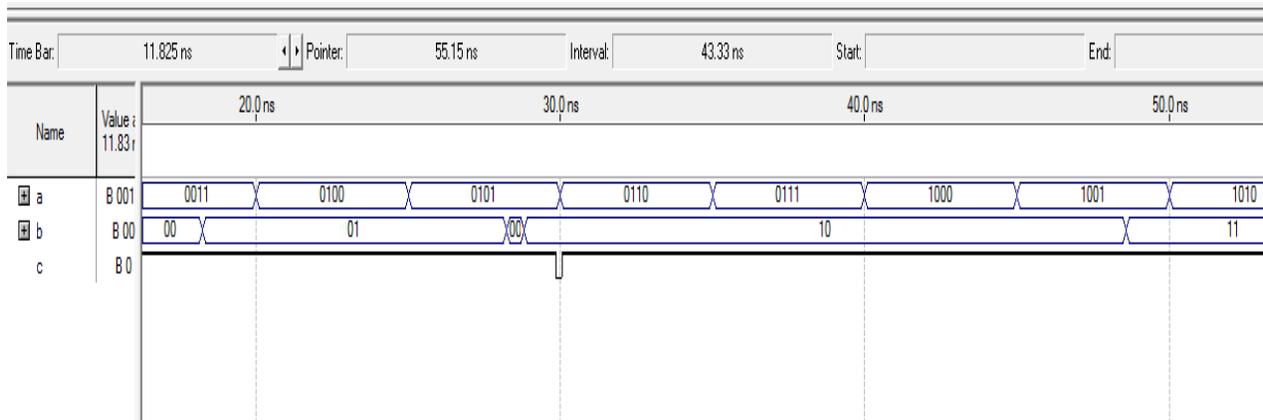


Figura 59 Diagrama de Bloques de la simulación



En base a la teoría que se vio anteriormente se observa que en la salida efectivamente se obtienen los valores esperados es decir 0, 1, 2 y 3 que vistos en el equivalente binario serían 00, 01, 10 y 11



En base a la teoría que se vio anteriormente se observa que en la salida efectivamente se obtienen los valores esperados es decir 0, 1, 2 y 3 que vistos en el equivalente binario serían 00, 01, 10 y 11

14.3 CODIFICADORES CON PRIORIDAD

Estos se diferencian de los codificadores simples por la siguiente característica principal:

Los codificadores con prioridad cumplen con la tarea de que cuando se activen dos o más entradas, el código de salida siempre muestre la entrada de mayor valor.

Por ejemplo:

Si se tiene en sus entradas los números 000, 001 y 100 en base al criterio de prioridad a la salida del codificador aparecerá el número 100 el cual es de mayor valor.

Dentro de los codificadores con prioridad se tiene el codificador decimal a BCD.

El diagrama se muestra a continuación.

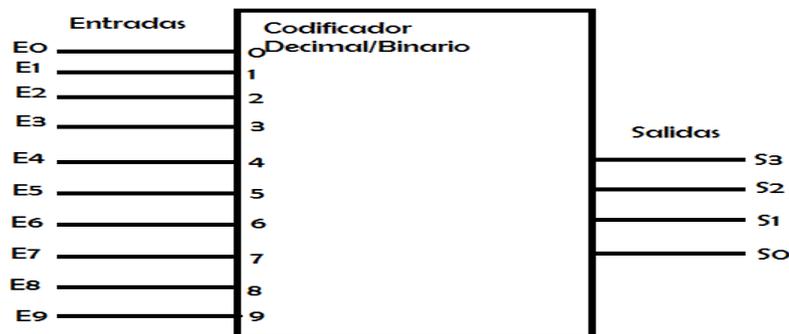


Figura 60 Codificador de 10 a 4 líneas

La tabla de verdad se muestra a continuación.

Entrada decimal	S3	S2	S1	S0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Tabla 47 Tabla de verdad del decodificador de 10 a 4 líneas

En el codificador que no es de prioridad solo una línea se encuentra activa en un codificador con prioridad se pueden encontrar varias líneas a la entrada.

Las aplicaciones de los codificadores se explicarán más adelante

A continuación, se analizan los bits MSB; es decir en qué casos se tiene un 1 (un alto)

Comenzando por S3

S3 tiene el valor de 1 cuando se tiene un 8 o un 9 lo cual puede expresarse con una compuerta OR

$$S3 = 8 + 9$$

Siguiendo con este orden el siguiente MSB será S2 los números para los cuales es un alto serán:

$$S2 = 4 + 5 + 6 + 7$$

El MSB siguiente es S1

$$S1 = 2 + 3 + 6 + 7$$

El MSB siguiente es S0

$$S0 = 1 + 3 + 5 + 7 + 9$$

En base a lo anterior se puede construir el codificador con compuertas como se muestra a continuación.

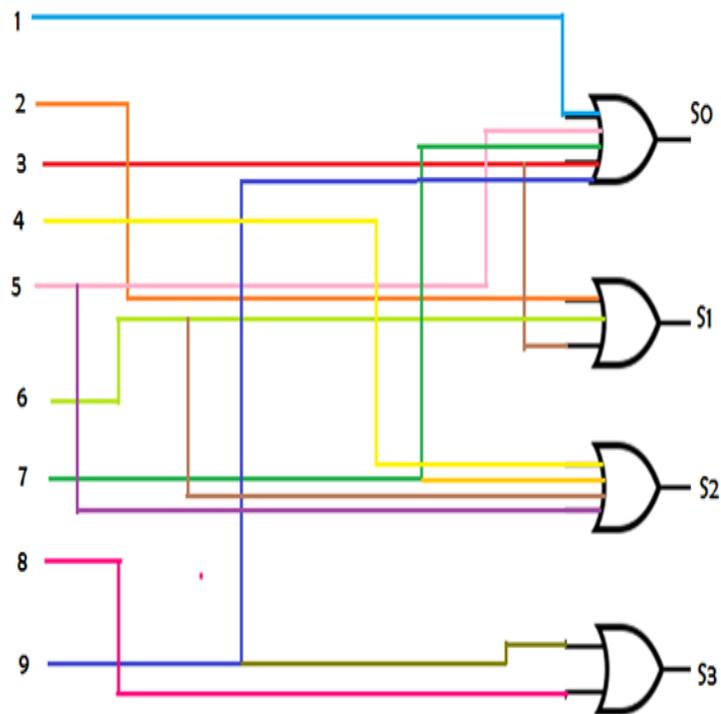


Figura 61 Implementación de un codificador de decimal a binario con compuertas diagrama tomado del libro de Floyd página 332

El codificador de prioridad como su nombre lo indica selecciona el de mayor prioridad esto es elige al de mayor valor, por ejemplo, si se tienen dos entradas 4 y 7 el codificador mostrará el valor 7.

Un ejemplo de codificador con prioridad es codificador decimal a binario 74147 cuyo diagrama lógico se muestra a continuación.

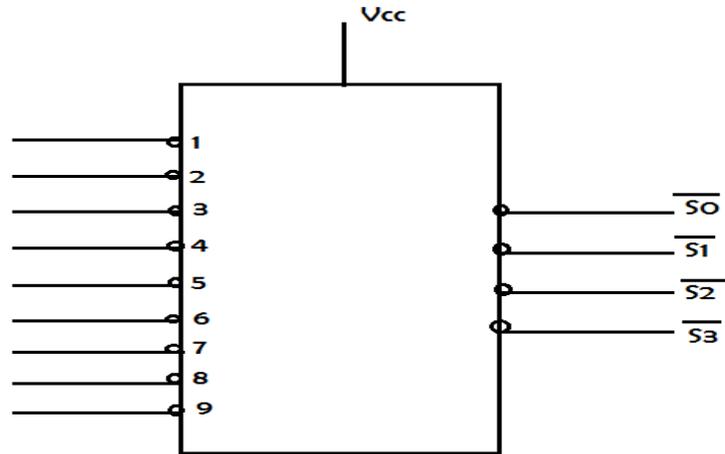


Figura 62 Codificador 74147

Como se puede ver en el diagrama los circuitos representan una negación, por lo tanto, tendremos entradas activas en bajo y salidas activas en cero es decir en bajo.

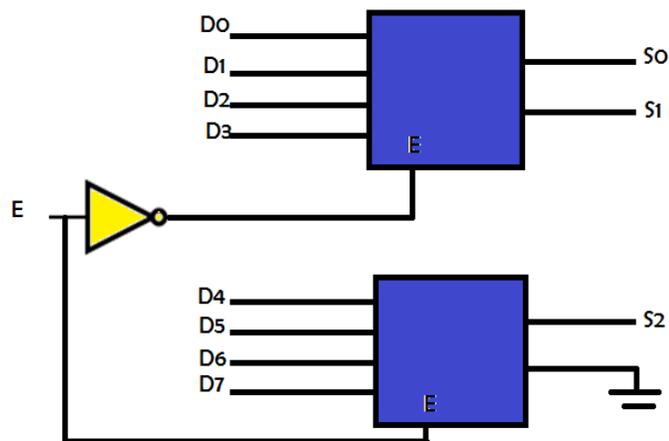
Cabe hacerse una pregunta:

¿Cómo se obtiene el número cero en el codificador?

Para obtener el número 0 todas las entradas deben actuar en 1.

Expansión de los codificadores

Implemente un codificador de 8:3 utilizando codificadores de 4:2



14.4 APLICACIÓN DE LOS CODIFICADORES

Un claro ejemplo de un codificador se aprecia en los teclados que permiten el acceso a áreas restringidas, por ejemplo, en los bancos.

En el teclado se indica el número en valor decimal, después este valor es codificado a un número en BCD (es decir a un número codificado por unos y ceros)

Y manda la salida correspondiente.

Otra aplicación típica de los codificadores es en el teclado de las computadoras.

Nota:

Los codificadores tienen la capacidad de convertir un número del código decimal u octal en un número binario, por otra parte, también pueden convertir caracteres alfanuméricos a un número binario.

Para este ejemplo en particular se mencionará el circuito 74147 el cual es un codificador que convierte un dígito decimal a binario.

Cada vez que se oprime un dígito en el teclado la información es procesada por la circuitería luego se pasa al codificador quien finalmente arroja como resultado el correspondiente número binario equivalente a la combinación del número introducido.

14.5 DECODIFICADORES

Si observa a su alrededor mucha información que se encuentra en forma discreta se representa en forma digital empleando para ello circuitos combinacionales.

Como usted sabe si se tiene n bits podemos ser capaces de representar hasta 2^n elementos distintos de información.

La tarea principal de los decodificadores es poder detectar información a la entrada del circuito dada en forma de código empleando un nivel de salida.

De forma general el decodificador tiene n líneas de entrada para manejar un número de bits igual a n y a la salida se tiene 2^n líneas cuya función es indicar las combinaciones

Podemos definir al decodificador como un circuito combinacional que convierte información binaria tomando para ello n líneas de entrada obteniendo un máximo de 2^n líneas de salida

Hay que tomar en consideración que si se presentan condiciones no usadas o no importa en la salida del decodificador encontraremos menos de 2^n salidas.

Los decodificadores pueden emplearse para la implementación de cualquier circuito combinacional.

La base de construcción de los decodificadores básicos son las compuertas AND; si se quieren salidas activas altas y compuertas NAND si se requieren salidas bajas activas además de emplear compuertas inversoras para la correcta funcionalidad del diseño.

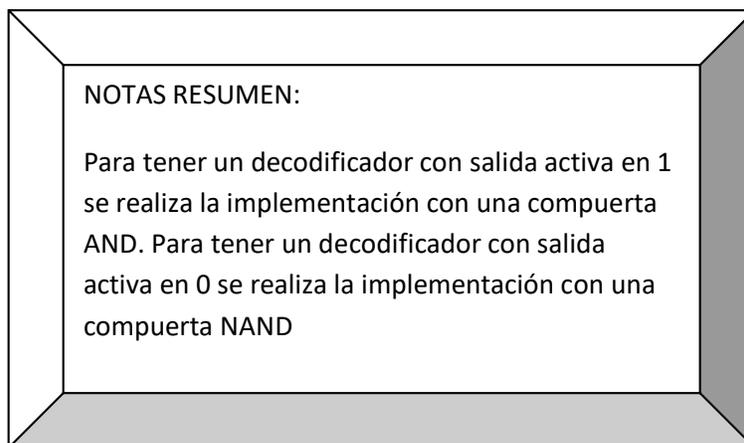
Analice los siguientes ejercicios:

Construya un decodificador básico para el código binario 1100 en salidas activas altas.

Si queremos que sean salidas altas activas inmediatamente nos viene a la mente emplear compuertas AND e inversoras

Con respecto a los decodificadores en alto y en bajo, los que tienen salida en bajo solo existen en la familia ttl, es decir son negados. Los decodificadores con salida en alto solo se utilizan para ilustrar la lógica positiva.

Si queremos que el mismo número 1100 se construya utilizando una salida activa baja emplearemos compuertas NAND e inversores como se muestra a continuación:



Ejemplo

Se tiene el número 00110

Si desea que la salida sea alta cuando aparezca este número construya un decodificador utilizando

Compuertas AND y NOT

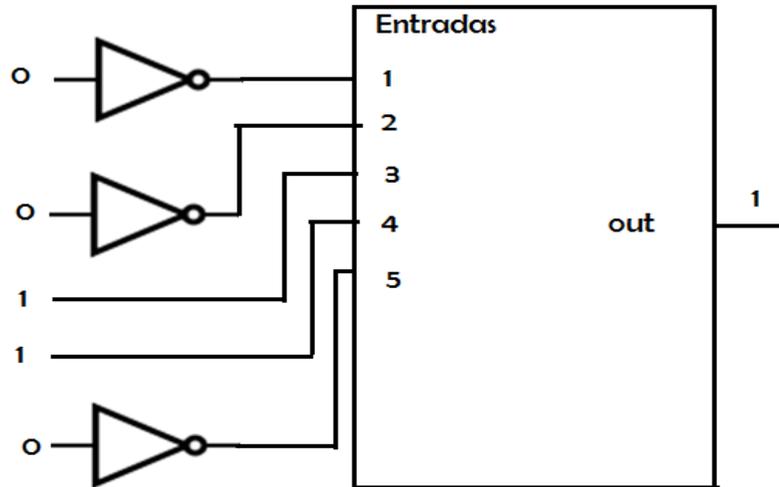


Figura 63 Construcción de un decodificador utilizando compuertas AND y NOT

¿Qué pasaría si ahora se desea que el circuito active una salida baja cuando identifique la entrada?

Se Trabajaría con una compuerta NAND

Por otra parte, podemos trabajar con una compuerta AND y compuertas NOT de la siguiente manera:

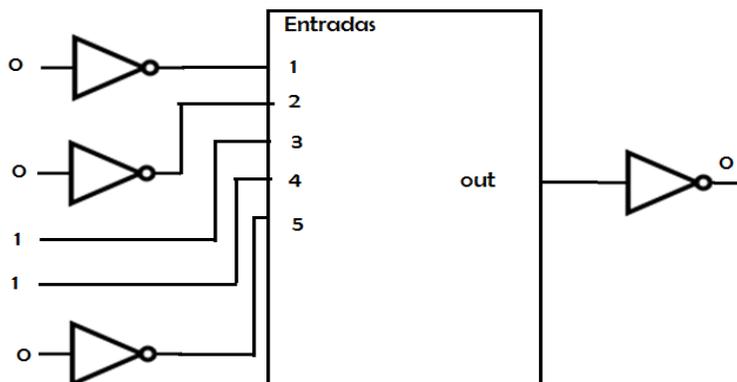


Figura 64 Implementación de un circuito con compuertas AND y NOT que activan una salida baja

Uno de los ejemplos más comunes de decodificador es el BCD a 7 segmentos.

Se tienen dos circuitos en el mercado son:

El 74LS48 el cual es cátodo común

Sea el circuito 74LS48 su tabla de verdad es:

N	Entradas	a	B	C	D	E	f	G
0	0000	1	1	1	1	1	1	0
1	0001	0	1	1	0	0	0	0
2	0010	1	1	0	1	1	0	1
3	0011	1	1	1	1	0	0	1
4	0100	0	1	1	0	0	1	1
5	0101	1	0	1	1	0	1	1
6	0110	0	0	1	1	1	1	1
7	0111	1	1	1	0	0	0	0
8	1000	1	1	1	1	1	1	1
9	1001	1	1	1	0	0	1	1

Figura 65 Tabla de verdad del circuito 74LS48 cátodo común

El 74LS47 el cual es ánodo común

La tabla de verdad será:

D	C	B	A	a	B	C	D	E	F	G
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	0	0
1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0

Figura 66 Tabla de verdad del circuito 74LS47 ánodo común

Decodificador de 4 a 16 líneas.

Dentro de la industria podemos encontrar un decodificador de 4 a 16 líneas con la matricula 74154, la tabla de verdad se muestra a continuación con las salidas activas en cero.

Digito Decimal	Equivalente Binario				Salidas															
0	0	0	0	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
4	0	1	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
5	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
6	0	1	1	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
7	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
8	1	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
10	1	0	1	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
11	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
12	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
13	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
14	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Tabla 48 Tabla de verdad de un decodificador de 4 a 16 líneas



Figura 67 Imagen del circuito integrado 74154

Aplicación de los decodificadores.

La aplicación de los decodificadores es muy amplia y versátil, un ejemplo muy claro y de mucha concurrencia es cuando se integran en una computadora para la manipulación y correcto direccionamiento de la información que viaja del CPU a cada uno de los distintos periféricos o dispositivos de E/S de las computadoras como es la pantalla, el teclado, el ratón.

Para ejemplificar lo anterior, imagine por un momento las casas de una localidad, cada una de las casas, tiene una dirección única para poder localizarla.

Lo mismo pasa con los periféricos cada uno de ellos tiene una dirección asignada.

La CPU emite la dirección o cadena de números binarios o código binario específico, el decodificador lleva a cabo el proceso y activa la señal adecuada a la que se desea activar.

Por ejemplo, suponga que el teclado tiene la dirección 1110.

Vea el siguiente diagrama.

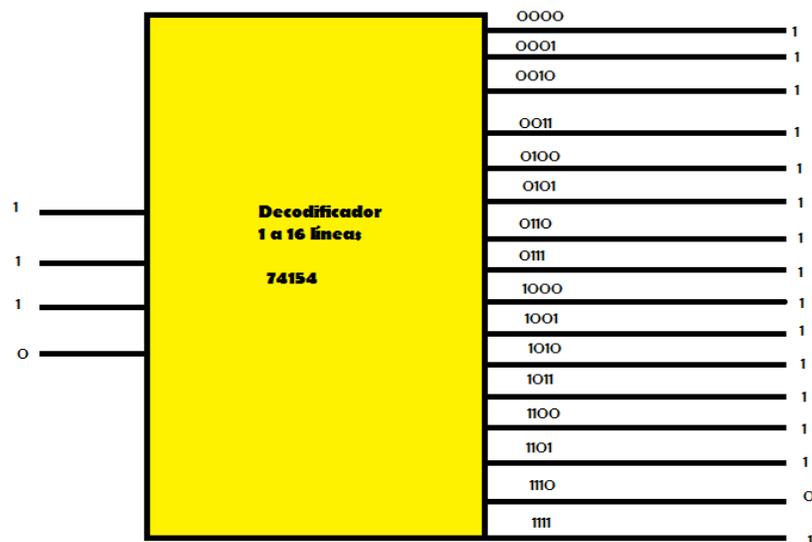


Figura 68 Imagen de un decodificador de 1 a 16 líneas

El diagrama anterior muestra cómo podría emplearse un decodificador en una computadora, es decir las líneas de entrada reciben la información, la procesan y activan la salida que se espera por ejemplo como se había mencionado anteriormente si la entrada es un 1110 perteneciente a la

dirección del teclado, el decodificador buscara esta dirección y activará la salida para activar el teclado en este caso se utilizó un cero para la activación en alto, en caso contrario de que la activación se indicará con un uno para indicar un alto se todas las demás salidas estarían en cero.

14.6 BCD DECIMAL CODIFICADO EN BINARIO.

Nota: Con el código BCD se pueden representar números decimales de los 10 dígitos que forman el sistema de base 10.

Ejemplos típicos de estos decodificadores son: 7447 y 7448.

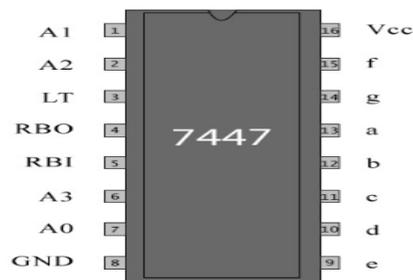


Figura 69 Codificador 7447

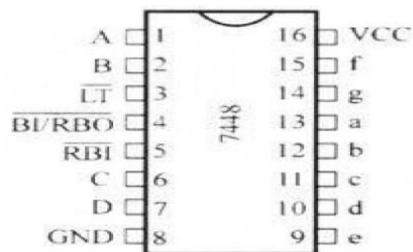


Figura 70 Codificador 7448

Ambos decodificadores mencionados anteriormente (7447 y 7448) tienen la característica de que un número entre el 0 y el 9 lo manen en BCD a sus entradas para arrojar como resultado el número elegido en forma digital en un display de 7 segmentos.

Cabe hacer mención que el circuito 7448 es cátodo común y el 7447 es ánodo común.

Un demultiplexor tiene una línea de entrada y n líneas de salida.

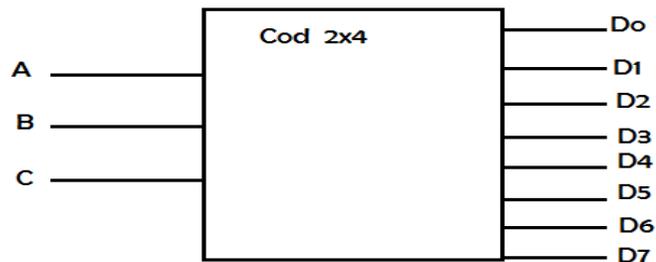
Como se había visto anteriormente tenemos el ejemplo del decodificador de 1 a 16 líneas es decir de las 16 salidas que tiene solo se activará una.

Por lo tanto, en base al ejemplo anterior se puede ver al demultiplexor como un caso de un decodificador.

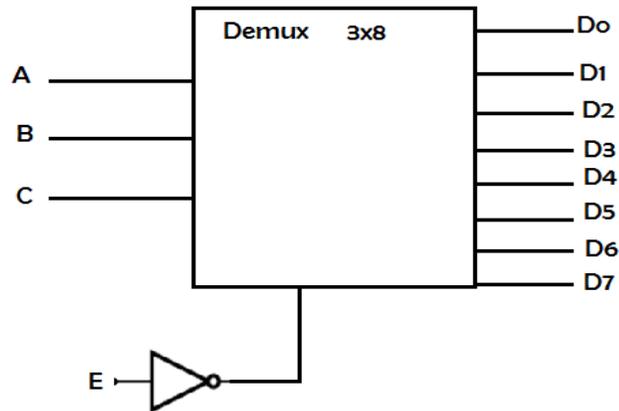
El enunciado anterior parte de lo siguiente:

Analice el siguiente ejemplo:

Se tiene un codificador de 3 entradas por 8 salidas como se muestra en el siguiente diagrama:



Un demultiplexor es como un codificador solo que le primero tiene una entrada de habilitación o data que junto con las líneas de seleccionan determina cuáles serán los datos que se transmitan

**NOTAS:****CAPACIDAD DE UN DECODIFICADOR**

Como se ha estudiado previamente ; los decodificadores tienen n líneas de entrada y 2^n líneas de salida, partiendo de un punto de vista matemático n puede tener un valor que cumpla con la siguiente restricción $n \geq 0$, en donde en base a este valor se obtiene la salida.

Desde el punto de vista físico los decodificadores comerciales tienen 2, 3 o 4 entradas, en base a ello y sustituyendo en la formula se tiene 4,8 y 16 salidas respectivamente

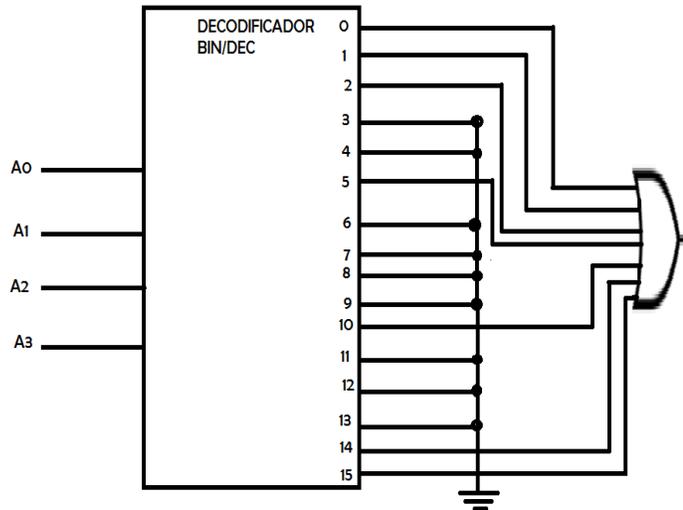


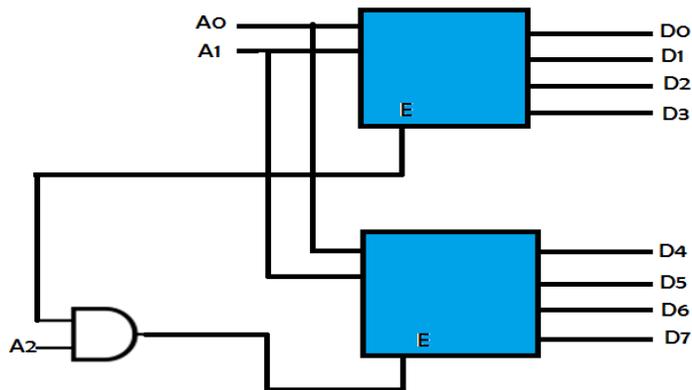
Figura 71 Implementación de un Decodificador de 4 a 16 líneas para la función dada por la tabla de verdad

La función de salida estará dada por la siguiente ecuación:

$$F(S) = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} + ABC\bar{D} + ABCD$$

Expansión de decodificadores:

Implemente un decodificador de 3:8 utilizando 2 decodificadores 2:4



14.7 Simulación de un decodificador 2 a 4 usando VHDL

```
library ieee;
use ieee.std_logic_1164.all;

entity ejemplo_deco_2_4 is

port (A:in std_logic_vector (1 downto 0);
      E: in std_logic;
      O:out std_logic_vector (3 downto 0));

end ejemplo_deco_2_4;

architecture DEC of ejemplo_deco_2_4 is
begin
process (A,E)
begin

if E ='0' then

O<= "0000";
else

case A is

when "00" => O <="0001";
when "01" => O <= "0010";
when "10" => O <= "0100";
when "11"=> O <= "0000";

end case;
end if;
end process;

end DEC;
```

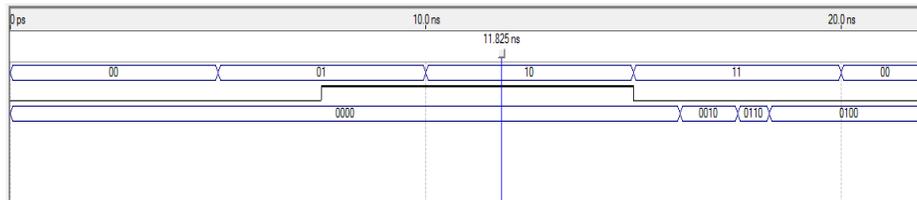
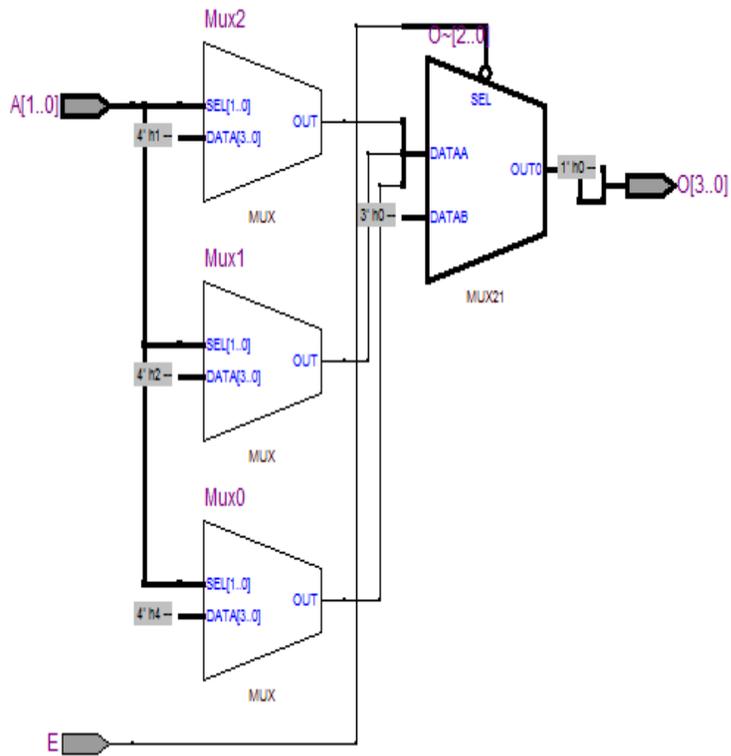


Figura 72 Simulación de decodificador de 2 a 4

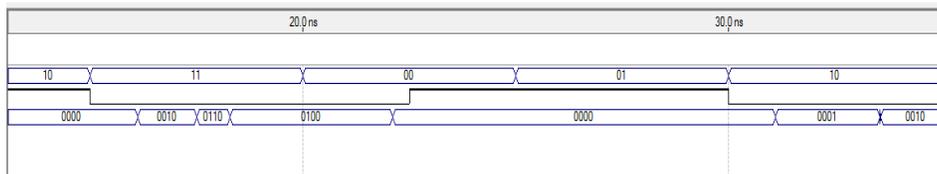


Figura 73 Simulación de decodificador 2 a 4

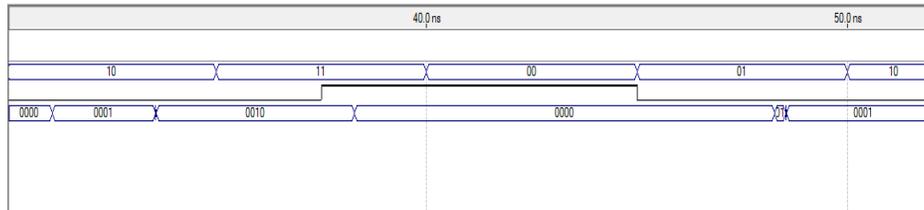


Figura 74 Simulación de decodificador 2 a 4

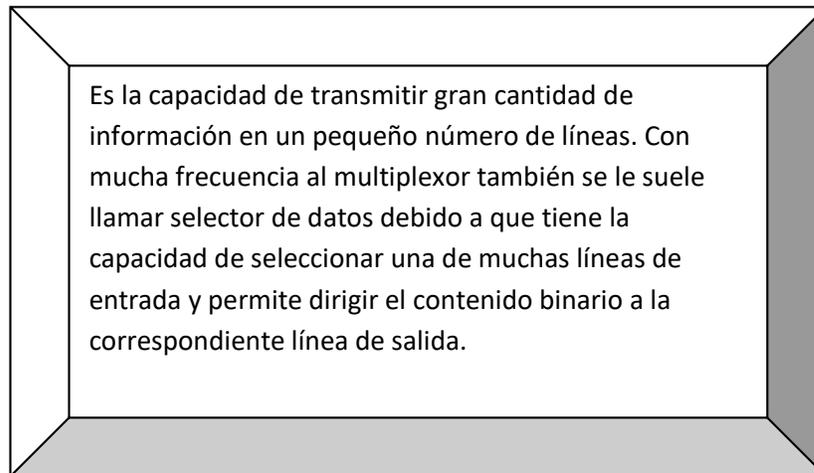


Figura 75 Simulación de decodificador 2 a 4

En base a las figuras que se muestran los resultados son los esperados es decir cuando se da un cambio entre un número y otro se da también el cambio en el número activo en la línea de selección.

14.8 Multiplexores/Selectores de Datos

La multiplexión es:



Imagine por un momento que para ir de paseo te

presentan las siguientes opciones:

Un lago, un río, la playa o el bosque

De todas estas solamente puedes elegir una.

Esto es lo que hace el multiplexor.

De varias opciones solo escoge una y la manda a la salida.

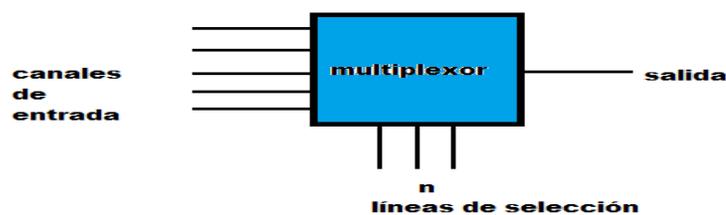


Figura 76 Diagrama de elementos de un multiplexor

14.9 DISEÑO DE UN MULTIPLEXOR

Para ello se necesita el número de líneas de selección n

Después para obtener el número de entradas se empleará la fórmula:

$$2^n$$

Finalmente, todos los multiplexores tienen 1 salida, por lo tanto, en base al número que se elija en el selector de datos, este será el que aparecerá en la línea de salida del multiplexor.

Por ejemplo

Si se tiene un multiplexor con dos líneas de selección, esto le permitirá elegir una de las cuatro posibles entradas, la cual será la que se muestre a la salida.

A continuación, se muestra el diagrama lógico para un multiplexor de 1 a 4 es decir una línea de salida y cuatro de entrada.

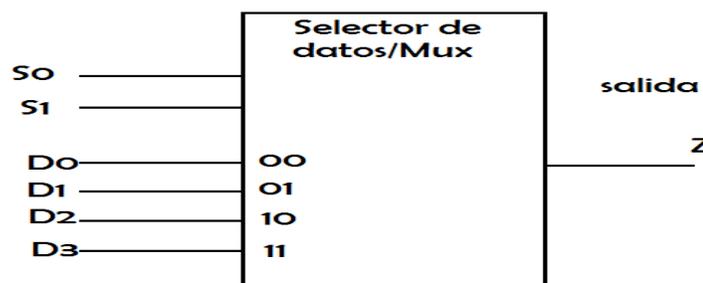


Figura 77 Selector de datos

Tabla de verdad para este multiplexor

D3	D2	D1	D0	Selector S1	Selector S0	Salida
0	0	0	1	0	0	D0
0	0	1	0	0	1	D1
0	1	0	0	0	0	D2
1	0	0	0	1	1	D3

Tabla 49 Tabla de verdad para el multiplexor del ejemplo

Esto se lleva acabo de la siguiente manera: en las líneas de selección se introduce el número binario que se desea introducir, en base a este número se selecciona un canal de entrada el cual se transmitirá directamente a la salida.

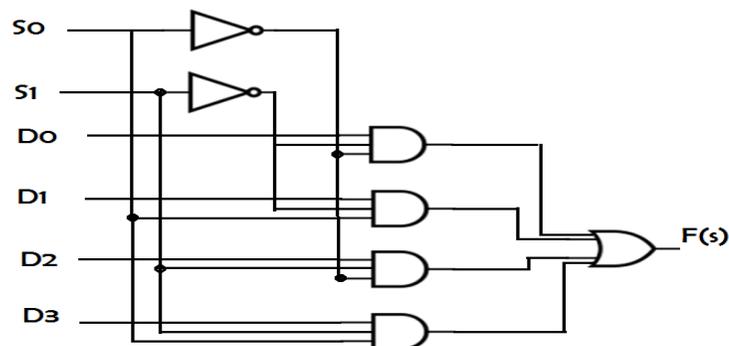


Figura 78 Implementación de un multiplexor con compuertas

La función de salida del multiplexor quedaría de la siguiente manera:

$$F(s) = D0 \bar{S1} \bar{S0} + D1 \bar{S1} S0 + D2 S1 \bar{S0} + D3 S1 S0$$

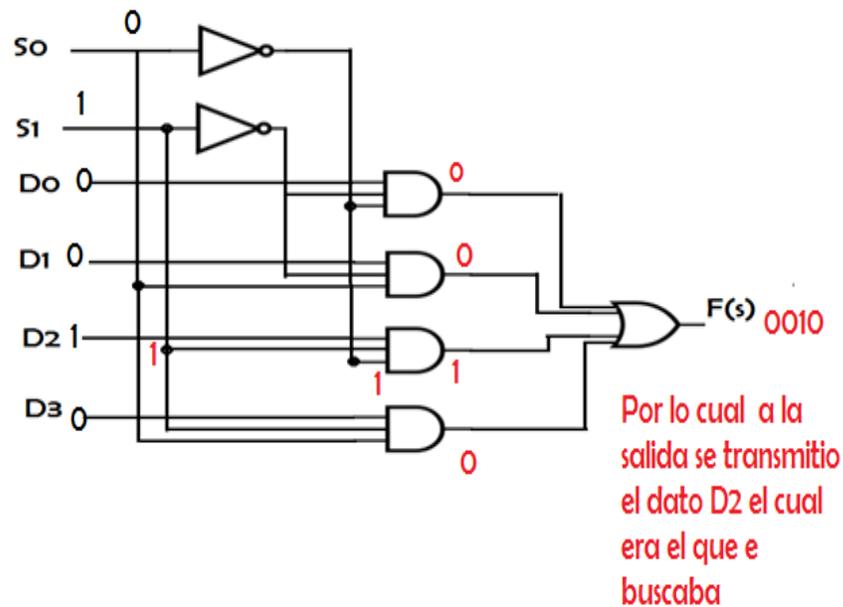
El diagrama puede entenderse de la siguiente forma:

Se tienen las dos líneas de selección, en base a ello se elegirá un canal de entrada el cual se transmitirá a la salida.

Analice el siguiente ejemplo:

Se tiene $S_0 = 0$ y $S_1 = 1$ a la salida del circuito se tendrá D2.

Analizando el diagrama entra un cero a S_0 siguiendo las conexiones entran 3 unos a la tercera compuerta AND que le corresponde a D2 esta entrada se transmite a la compuerta OR y sale el uno es decir D2.



EXPANSIÓN DE UN MULTIPLEXOR

Ejercicio: Construya un multiplexor de 16 canales si solo cuenta con mux 8:1

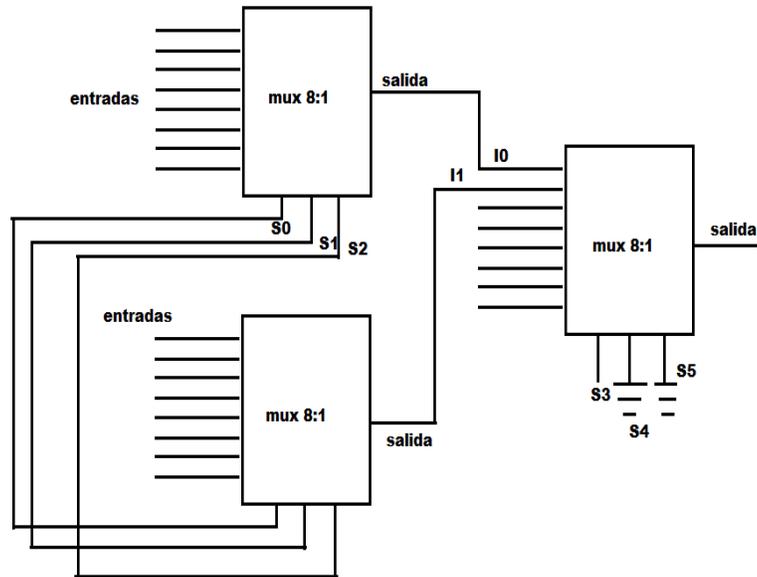


Figura 79 Diagrama de la construcción de un multiplexor de 16 canales

La fig.79 muestra la forma en la que se utilizan tres multiplexores para poder expandir el número de entradas. Como se aprecia en el ejemplo comercialmente se cuenta con multiplexor de 8 entradas a un canal de salida con tres líneas de selección cada uno.

En el caso presentado se necesitan 16 canales de entrada y una salida, es por ello que se arman los multiplexores para que utilizando tres con las características que se mencionan s pueda obtener el resultado que se desea es decir contar con 16 canales de entrada y una salida.

Ejemplo: Implementar un multiplexor 8:1 empleando para ello multiplexores de 4:1

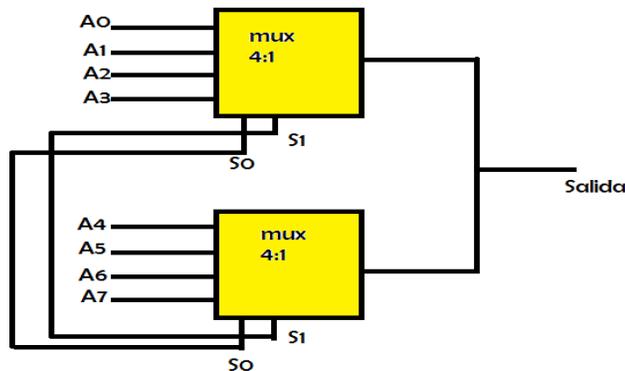


Figura 80 Expansión de multiplexor 4 a 1

El Multiplexor como Generador de funciones

El multiplexor puede ser empleado como generador de funciones lógicas, expresando el resultado como una suma de productos o producto de sumas.

Por ejemplo, si tenemos un multiplexor 8 a 1 (8 entradas una salida), cuya tabla de verdad se muestra más adelante y que puede implementarse mediante el circuito lógico 74151, podemos representar lo anterior mediante el siguiente diagrama lógico, que se muestra a continuación.

Selector de Datos S2	Selector de datos S1	Selector de datos S0	Salida F(s)	Salida activa D
0	0	0	1	D0
0	0	1	0	D1
0	1	0	0	D2
0	1	1	1	D3
1	0	0	0	D4
1	0	1	1	D5
1	1	0	0	D6
1	1	1	1	D7

Tabla 50 Tabla de verdad de la función implementada para un decodificador de 4 a 1 líneas

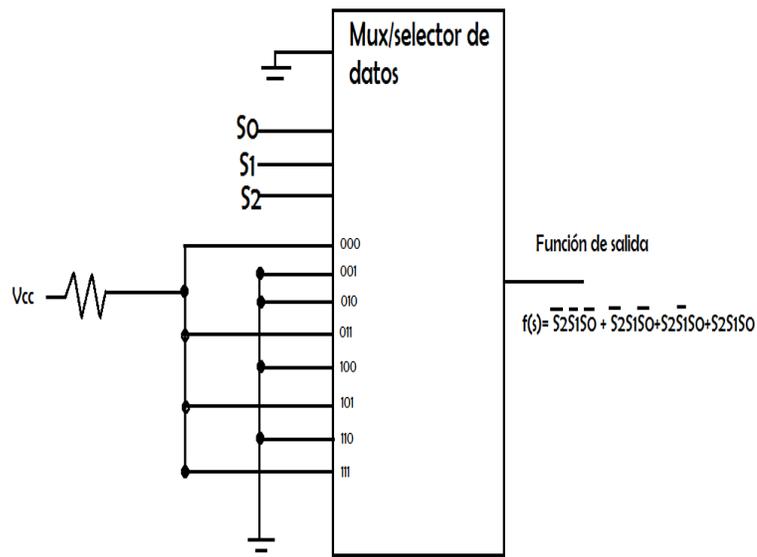


Figura 81 Diagrama de un selector de datos

Como se vio previamente el uso de un multiplexor para la implementación de una función, es mucho más rápida y en términos económicos mucho más barata ya que ahorra la implementación de muchas compuertas, lo cual asimismo se traduce en ahorro de espacio y en una mejora del tiempo de respuesta.

Nota: Solo el decodificador salida en baja se puede utilizar como demultiplexor

Expansión de multiplexor.

Utilizar dos multiplexores de 4:1 para construir uno de 16:1

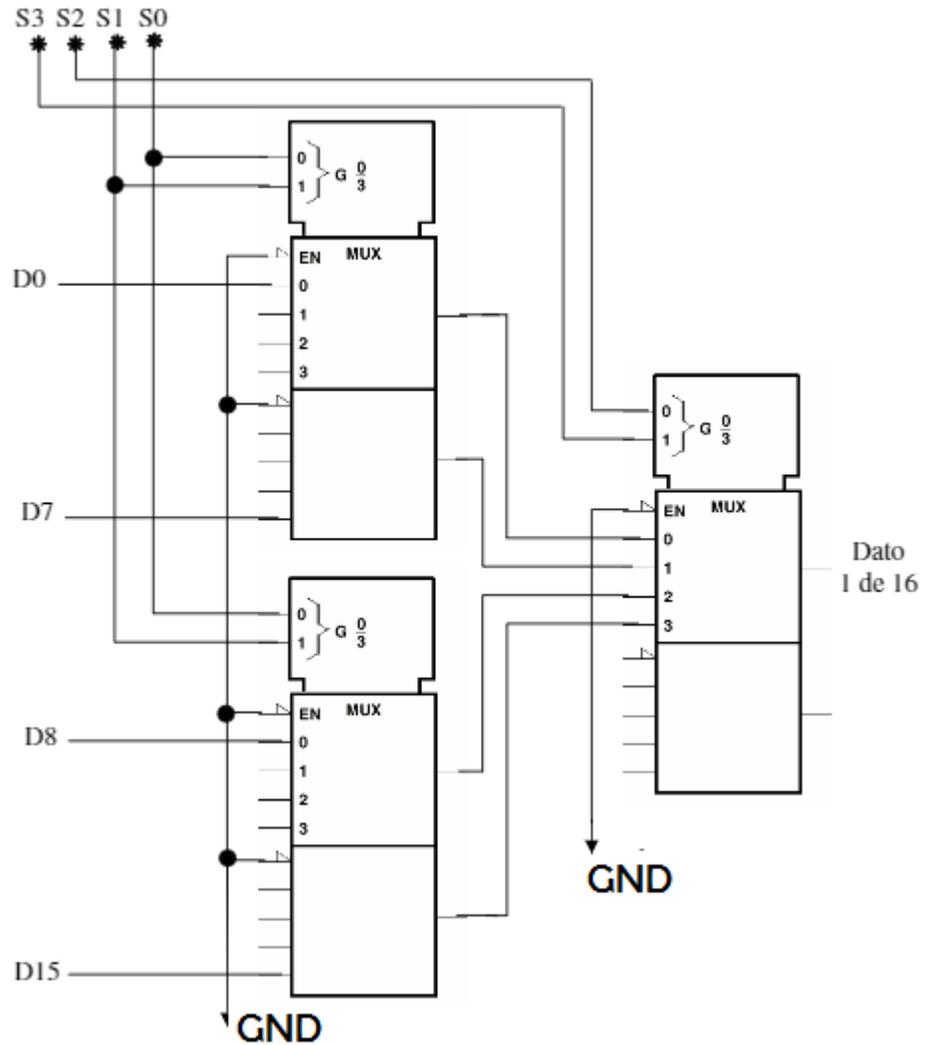


Figura 82 Expansión de un multiplexor 4:1 para obtener un mux de 16:1 imagen tomada de <https://docplayer.es/251676-Tema-5-sistemas-combinacionales-msi.html>

14.10 Ejemplo de un multiplexor empleando VHDL

```

library ieee;
use ieee.std_logic_1164.all;

entity programa_multiplexor_1 is
Port (a: in std_logic_vector (3 downto 0);
      b: in std_logic_vector (3 downto 0);

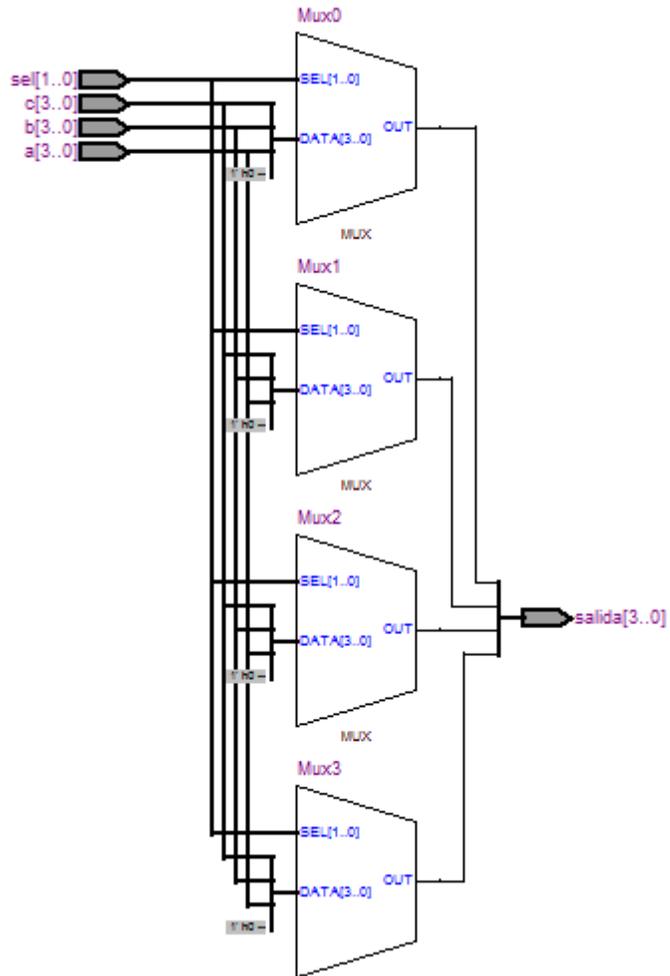
```

```
c: in std_logic_vector (3 downto 0);
sel: in std_logic_vector (1 downto 0);
salida: out std_logic_vector (3 downto 0));

end programa_multiplexor_1;

architecture synth of programa_multiplexor_1 is
begin
process (sel, a,b,c) is
begin

case sel is
when "00"=> salida <=(others => '0');
when "01"=> salida <=a;
when "10" => salida <=b;
when "11" => salida <=c;
when others =>salida <=(others =>'0');
end case;
end process;
end synth;
```



Resultado de la simulación

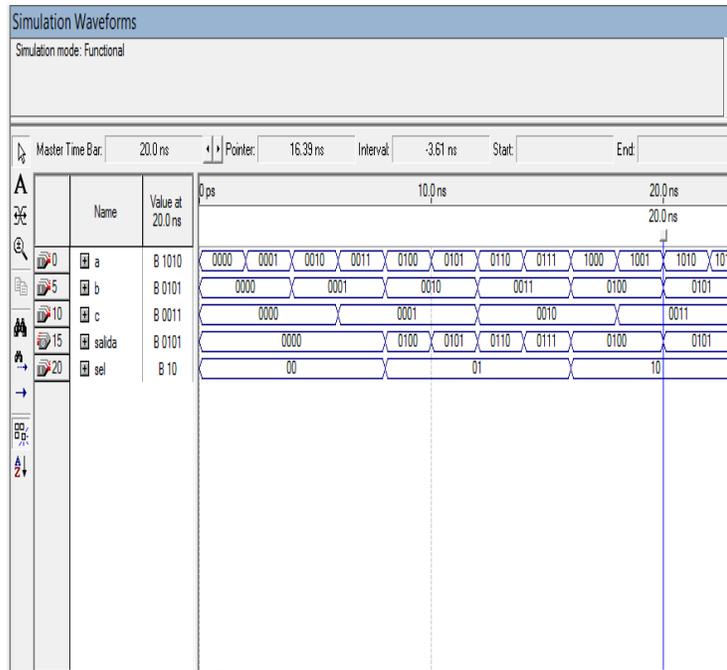


Figura 83 simulación de multiplexor

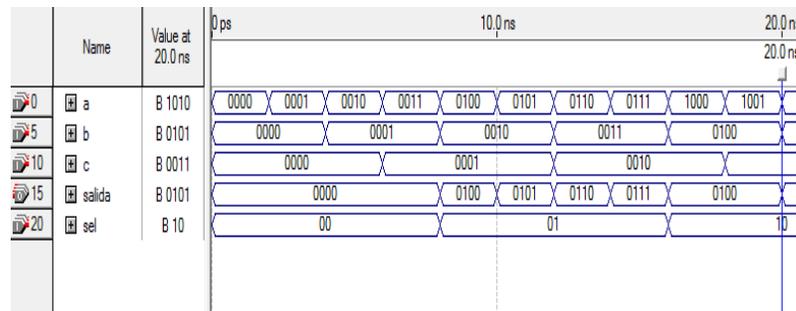


Figura 84 Simulación de multiplexor

En base a las figuras de la simulación de multiplexor se aprecia que la salida es la que se espera es decir se tienen varios canales de entrada y una sola salida es la que se selecciona en base a la línea de selección.

14.11 DEMULTIPLEXORES

La característica principal de un demultiplexor es que tiene una línea de entrada y 2^n ; líneas de salida y muy importante tiene n líneas de control también conocidas como líneas de selección.

Otro nombre con el cual se conoce a los Demultiplexores es como distribuidores de datos.

Existen casos de CI en donde el decodificador puede emplearse como Demultiplexor gracias a las entradas de habilitación con la que cuentan.

El diagrama para un demultiplexor es:

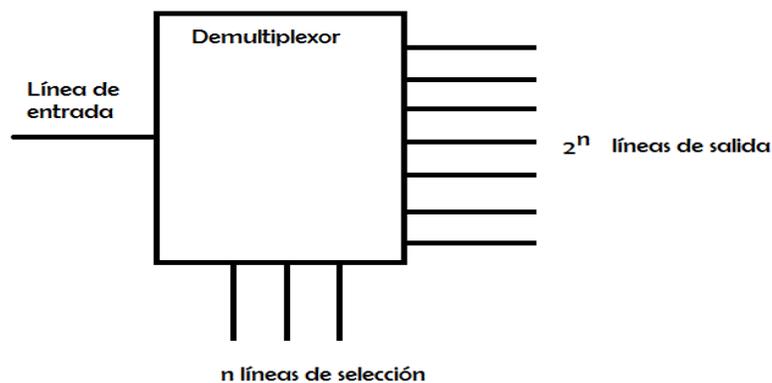


Figura 85 Diagrama de elementos de un multiplexor

Un caso especial del decodificador es como un demultiplexor

NOTA:

Si compara el diseño de un demultiplexor de 1 a 8 líneas con el decodificador de 3 a 8 líneas podrá notar que son muy

similares, la única diferencia es en el pin de entrada correspondiente a la habilitación.

Un ejemplo de lo anterior es cuando el decodificador cuenta con un pin o entrada de habilitación, en este caso se puede ver al decodificador como un demultiplexor, ejemplo de este circuito integrado es el que tienen la matricula 74138.

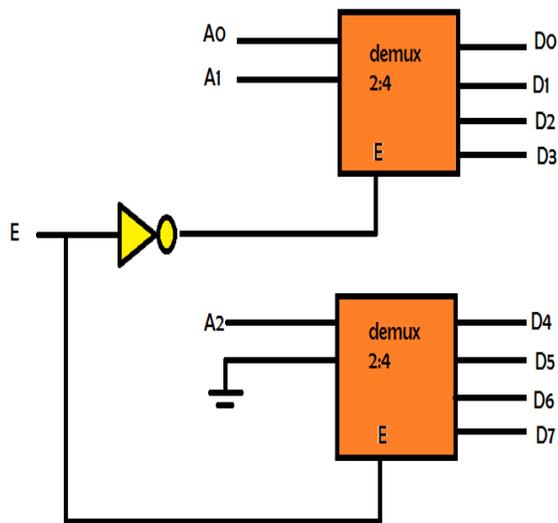
**TABLA DE VERDAD PARA EL DEMULTIPLEXOR
DE 1 A 8 LÍNEAS**

Entrada De Selección	Entrada De selección	Entrada De selección	Sal							
S2	S1	S0	Out0	Out1	Out2	Out3	Out4	Out5	Out6	Out7
0	0	0	E	0	0	0	0	0	0	0
0	0	1	0	E	0	0	0	0	0	0
0	1	0	0	0	E	0	0	0	0	0
0	1	1	0	0	0	E	0	0	0	0
1	0	0	0	0	0	0	E	0	0	0
1	0	1	0	0	0	0	0	E	0	0
1	1	0	0	0	0	0	0	0	E	0
1	1	1	0	0	0	0	0	0	0	E

Tabla 51 Tabla de verdad para un demultiplexor de 1 a 8 líneas

Expansión del demultiplexor

Ejemplo emplear dos demultiplexores de 2:4 para obtener un demultiplexor de 3:8



14.12 Simulación de demultiplexor

```

library ieee;
use ieee.std_logic_1164.all;

entity Demux is

port (

X: in std_logic;
S:in std_logic;
Y:out std_logic_vector(1 downto 0)

);

end Demux;

architecture simple of Demux is

```

```

begin

Y(0) <=X when (S='0') else '0';
Y(1) <=X when (S='1') else '0';

end simple;

```

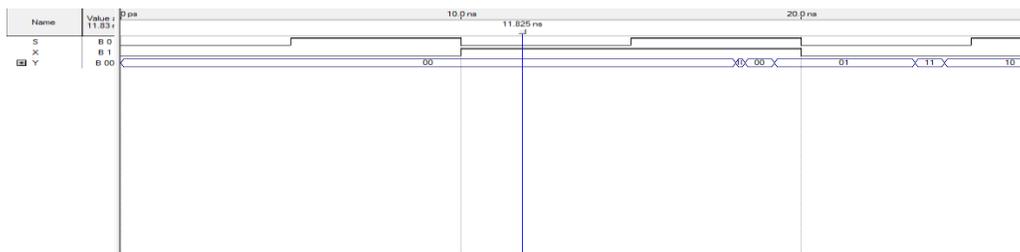
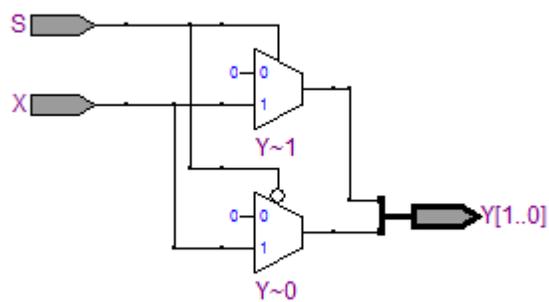


Figura 86 Simulación de demultiplexor

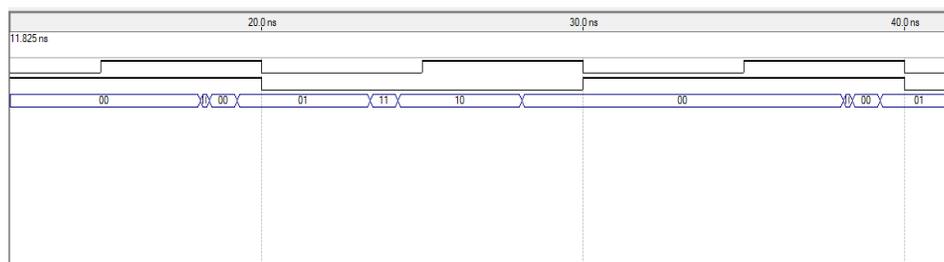
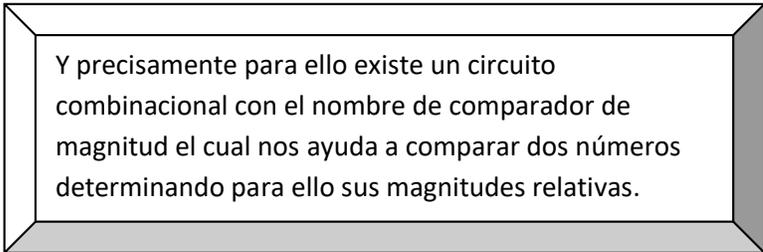


Figura 87 Simulación de demultiplexor

Como se puede apreciar en las simulaciones el resultado es el que se esperaba ya que un demultiplexor tiene una sola entrada y se activan la señal de salida correspondiente.

COMPARADORES DE MAGNITUD

Siempre cuando se trabaja con dos cantidades una de las características que nos interesa conocer entre ellas es la relación que existe si una es mayor, menor o igual con respecto a la otra.



Y precisamente para ello existe un circuito combinacional con el nombre de comparador de magnitud el cual nos ayuda a comparar dos números determinando para ello sus magnitudes relativas.

Se utilizan para ello 3 variables binarias para indicar:

$A < B$, $A > B$, $A = B$

Donde A y B representan dos números.

Como ejemplo se mostrará un procedimiento para comparar dos números de 5 bits de una forma generalizada.

Supongamos que tenemos dos números.

$$A = A_1A_3A_2A_1A_0$$

$$B = B_4B_3B_2B_1B_0$$

Lo primero que hacemos es comparar cada uno de los bits de la siguiente manera para verificar si son iguales:

$$A_4 = B_4 \quad A_3 = B_3 \quad A_2 = B_2 \quad A_1 = B_1 \quad A_0 = B_0$$

Si resulta que todos los bits coinciden, podemos expresar la siguiente relación para cada par de números:

$$X_i = A_iB_i + \bar{A}_i\bar{B}_i \quad i = 0, 1, 2, 3$$

Tenemos que $X_i = 1$, si y solo si el par de números son iguales ya sea a 1 o un 0

Se representa $A=B$ a la salida cuando, la salida=1, si los números A y B son iguales y es un 0 en caso contrario.

En este caso emplearemos la compuerta AND en donde todas las variables X_i , deben ser iguales a 1

$$(A = B) = X_4X_3X_2X_1X_0$$

En caso de que los números no sean iguales nos preguntamos cual es mayor que o menor que.

Entonces nos ubicamos en la posición más significativa. Si los dos dígitos son iguales seguimos a la siguiente posición hasta encontrar una diferencia.

Cuando ocurre la primera diferencia preguntamos lo siguiente:

A_i Es 1 y el de $B_i = 0$

Entonces concluimos que $A > B$

En caso contrario cuando

A_i es 0 y es de $B_i = 1$

Concluimos que el número B es mayor por lo tanto $A < B$

Esta comparación secuencial la podemos expresar de la siguiente manera:

$$1) \quad (A > B) = A_4B'_4 + x_4A_3B'_3 + x_4x_3A_2B'_2 + x_4x_3x_2A_1B'_1 + x_4x_3x_2x_1A_0B'_0$$

$$2) \quad (A < B) = A'_4B_4 + x_4A'_3B_3 + x_4x_3A'_2B_2 + x_4x_3x_2A'_1B_1 + x_4x_3x_2x_1A'_0B_0$$

Las variables binarias $A > B$ y $A < B$ son iguales a 1 cuando se cumplen las condiciones:

14.13 COMO SE EXPANDE UN COMPARADOR

A continuación, se muestra el diagrama donde se expande dos comparadores para comparar dos números de 16 bits.

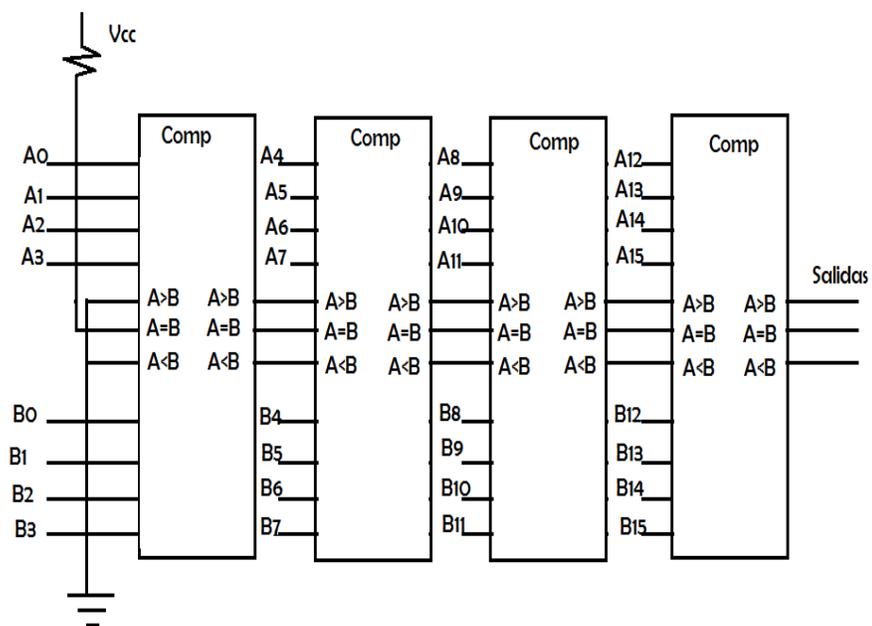


Figura 88 Expansión de un codificador para comparar dos números de 16 bits

Las entradas de expansión son $A > 0$, $A = B$, $A < B$

Como sabe el circuito 7485 es un comparador que permite llevar a cabo la comparación de dos números de 4 bits.

Para este caso en particular se comparan 2 números de 16 bits empleando el circuito integrado que se muestra en la figura 88.

14.14 DETECTORES DE PARIDAD

Suponga que se tiene la siguiente tabla:

A	B	
0	0	Par
0	1	Impar
1	0	Impar
1	1	Par

Tabla 52 Tabla para el análisis de paridad par e impar

Cuando los valores de A y B son todos iguales tenemos paridad par y cuando uno es diferente tenemos paridad impar.

Si quisiera implementar lo anterior

Así es una XOR

Recordemos su tabla

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 53 Tabla de compuerta XOR

¿Cómo revisar la paridad cuando se tienen más de dos bits?

En este caso se emplea una adición como podrá ver en los siguientes ejemplos:

$$\begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array}$$

$$\begin{array}{r} 1 \\ +1 \\ \hline 11 \end{array}$$

$$\begin{array}{r} 1 \\ 1 \\ + 1 \\ \hline 100 \end{array}$$

Supongamos que se tiene la siguiente tabla:

A	B	
0	0	Par
0	1	Impar
1	0	Impar
1	1	Par

Tabla 54 Tabla de paridad par e impar

Recuerde la tabla:

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 55 Tabla de compuerta XOR

¿Cómo se revisa la paridad cuando se tienen más de dos bits?

En este caso se emplea una adición.

Observa los siguientes ejemplos:

$$\begin{array}{r}
 1 \\
 +1 \\
 \hline
 10
 \end{array}
 \qquad
 \begin{array}{r}
 1 \\
 + 1 \\
 \hline
 1 \\
 11
 \end{array}
 \qquad
 \begin{array}{r}
 1 \\
 1 \\
 + 1 \\
 \hline
 1 \\
 100
 \end{array}$$

Como se habrá dado cuenta en la adición se emplea un número par de unos, se obtiene como resultado cero sin tomar en cuenta los acarreo y cuando se adicionan un número impar de unos se obtiene como resultado un uno sin tomar en cuenta nuevamente los acarreo

Implementación con compuertas XOR

A	B	C	Paridad impar	Paridad par	
0	0	0	1	0	
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	0	1	
1	0	0	0	1	1
1	0	1	0	1	
1	1	0	0	1	
1	1	1	0	1	1

Tabla 56 Tabla de paridad par e impar con compuertas XOR

Nota: se agrega un uno para que tener paridad par, es decir para que el número de unos que se tenga sea un número par.

15. SUMADOR COMPLETO Y RESTADOR COMPLETO

INTRODUCCIÓN

15.1 SUMADOR COMPLETO

Los sumadores forman parte importante del diseño perteneciente a la categoría de lógica combinacional. Es decir, las salidas dependen del estado presente del sistema.

Se puede observar la aplicación de los sumadores en las computadoras, por ejemplo.

La adición se lleva a cabo mediante cuatro operaciones fundamentales:

$$0+0=0$$

$$0+1 =1$$

$$1+0=1$$

$$1+1=10$$

Se puede apreciar que en el inciso d se obtuvo como resultado d la suma de dos bits, el 1 que aparece en este resultado es conocido como bit de acarreo. Para llevar a cabo estas operaciones se emplea lo que se conoce como medio sumador.

15.2 MEDIO SUMADOR

El medio sumador se emplea para llevar a cabo la adición de dos bits.

Cuando se requiere sumar dos bits se emplea el adicador completo.

La tabla de verdad para un medio adicador es la siguiente

A	B	C	Out
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Tabla 57 Tabla de verdad para un medio sumador de dos bits

Al analizar la tabla se aprecia que las dos funciones de salida C que es el acarreo y Out que es la salida se pueden implementar con una compuerta XOR y AND respectivamente, como se muestra a continuación.

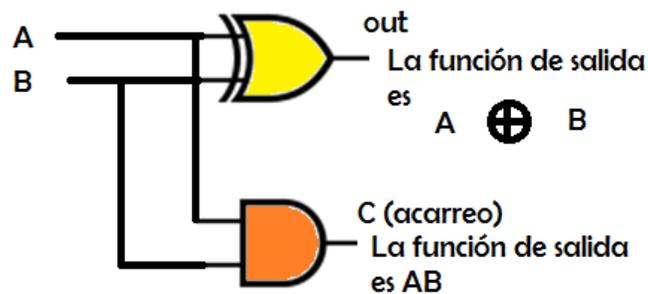


Figura 89 Diagrama de medio sumador

15.3 SUMADOR COMPLETO

El sumador completo se utiliza para llevar a cabo sumas de más de dos bits, por ejemplo, a continuación, se verá la implementación del sumador completo para sumar tres bits. Como podrá verse en el ejercicio la implementación consiste de dos medios sumadores.

A continuación, se muestra la tabla y el respectivo diagrama implementado con compuertas.

E	F	G	C	Out
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Tabla 58 Tabla de verdad para un sumador completo

Para el análisis de la tabla será necesario que vaya realizando las sumas renglón por renglón y de izquierda a derecha por ejemplo para el primer renglón la suma de $E + F = 0$ a este valor le suma el valor de G el resultado de la suma será nuevamente cero en este caso no llevo ningún acarreo por lo tanto $C=0$ y Out será igual a cero.

Realizando la suma del siguiente renglón el resultado será el siguiente $E=0$ y $F=0$ la suma será igual a cero si se suma el valor de G , donde $G=1$ se obtendrá como resultado $Out=1$ y $C=0$ puesto que no habrá ningún acarreo.

Las demás sumas se efectúan de manera parecida. Los resultados los puede comprobar con los de la tabla.

Como se mencionó anteriormente la implementación del sumador completo es con dos medios adicionadores y una compuerta OR para implementar la función de acarreo.

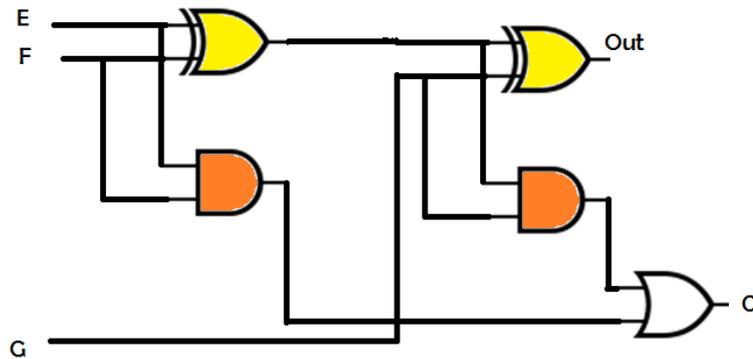
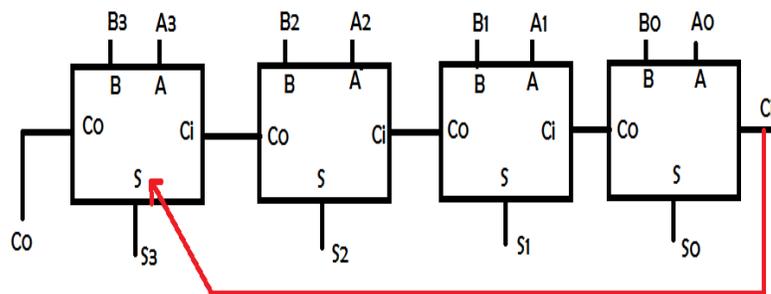


Figura 90 Implementación de un sumador completo de 3 bits

EXPANSIÓN DE SUMADORES

Para llevar a cabo la expansión de sumadores, estos se conectan en serie como se muestra a continuación.



15.4 RESTADOR MEDIO

Otra de las operaciones básicas que llevan a cabo las computadoras es la sustracción las operaciones que se realizan son las siguientes:

$$0-0=0$$

$$1-0=1$$

$$1-1=0$$

$$10-1=1$$

Para la situación anterior se toma prestado un 1 lo cual será equivalente a tomar un dos por lo cual la operación que se estaría realizando sería $2-1=1$.

A continuación, se muestra la tabla de verdad para un medio restador.

A	B	P(préstamo)	R(resta)
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Tabla 59 Tabla de verdad para un medio restador

Como se puede apreciar de la tabla se puede implementar la función de salida para la función resta con una compuerta XOR, como fue en el caso del medio sumador y para la función de salida de la función préstamo implementar con una función AND.

El diagrama se muestra a continuación:

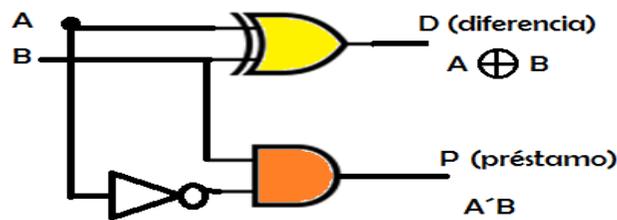


Figura 91 Diagrama de un medio restador para dos bits

15.5 RESTADOR COMPLETO

Para analizar el comportamiento del restador completo se empleará su respectiva tabla de verdad.

A	B	C	P	R
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Tabla 60 Tabla de verdad de un restador completo

Analizando el cuadro anterior llevando la diferencia, $A-B=0$ a este valor se le resta el valor de C en base a lo anterior se tienen que $P=0$ y la resta da como resultado $R=0$.

Tomando el renglón 2 $A-B=0$ y a este valor se le resta C $0-1=1$ donde $R=1$ con un préstamo de 1.

El análisis de toda la tabla se hace de manera similar.

Se pueden utilizar mapas de karnaugh para minimizar las funciones.

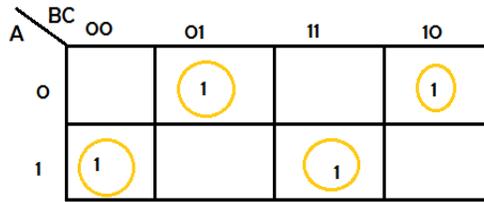


Figura 92 Mapa de Karnaugh para la función de salida resta

La función será: $A'B'C + A'BC' + AB'C' + ABC$

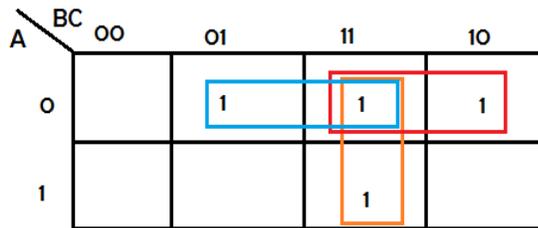


Figura 93 Mapa de Karnaugh para la función préstamo

La función de salida será: $A'B + B'C + BC$

La expansión de un restador se hace de manera similar que en el caso del sumador.

16. Memorias RAM y ROM

Deténgase por un momento y observe a su alrededor.



Figura 94 Análisis de un entorno

A continuación, analice su entorno y describa en que parte se está llevando a cabo una serie de procesos en los cuales se recabe información en forma analógica o digital y como se guarda está.

Como se habrá dado cuenta hay muchos sistemas que llevan a cabo dichos procesos.

Y es que actualmente estamos inmersos en un mundo en donde la adquisición y evaluación de datos es imprescindible y esto solo se puede hacer gracias al almacenamiento de los mismos.

Todos los días grandes cantidades de información son almacenadas en dispositivos de memoria durante lapsos de tiempo que varía según la aplicación de los datos.

16.1 OPERACIONES BÁSICAS DE LA MEMORIA

Los datos almacenados deben estar disponibles para poder llevar acabo más análisis con ellos o para ser sustituidos por otros.

Son 3 las operaciones básicas que se llevan a cabo en la memoria:

Direccionamiento

Escritura

Lectura

Gracias a estas tres operaciones la información entra y sale de la memoria.

A continuación, se describirá cada una de estas operaciones:

DIRECCIONAMIENTO

Para conocer la ubicación de una celda para una palabra es necesario dar su dirección, es decir

Proporcionar la columna o la fila en la que se encuentra.

Precisamente el direccionamiento consiste en seleccionar la dirección apropiada de la localidad de memoria.

La operación de direccionamiento cuenta con un elemento muy importante la barra de datos, esto permite que los datos entren o salgan de la memoria cumpliendo así con las características de ser bidireccional.

Dentro del proceso de direccionamiento para que el decodificador reconozca el código es necesario contar con una barra de dirección es precisamente en esta barra donde se ingresa el código que identifica la dirección de la localidad de memoria.

En base a esto la capacidad de memoria ayuda a determinar el número de líneas en la barra de dirección.

Por ejemplo, si se cuentan 3 bits se tienen:

$2^3 = 8$; Se tendrán 8 líneas.

OPERACIÓN ESCRITURA

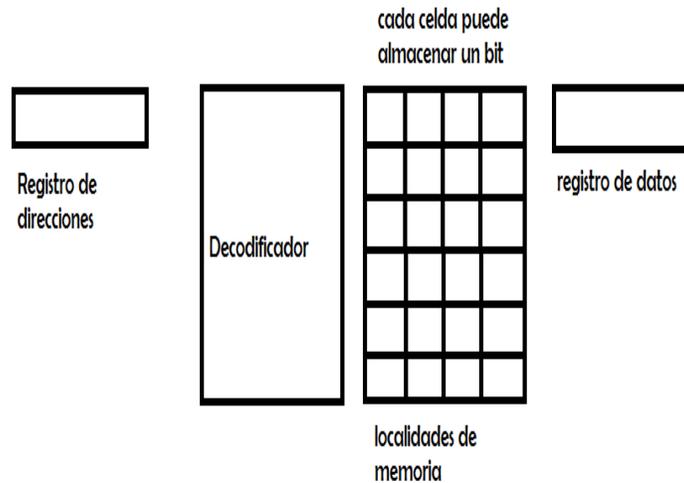


Figura 95 Diagrama interno de una memoria tomado del libro de Floyd página 586

Se ingresa la palabra en el registro de direcciones posteriormente esta es seleccionada y decodificada, el decodificador activa el código en la salida indicada y finalmente el código aparece en el registro de datos.

Es importante recalcar que cada vez que se lleva a cabo este proceso se anula o se elimina el código que estaba antes escrito en la localidad de memoria, otro nombre con el que se conoce a la memoria es registro.

OPERACIÓN DE LECTURA

El proceso de lectura es muy similar al que se lleva a cabo en el registro de direcciones. El código entra en el registro de direcciones, después pasa por la barra de direcciones para posteriormente entrar al decodificador, el cual decodifica el código y selecciona la celda de la memoria una vez hecho esto el código se va a registro de datos.

En el proceso de lectura el código que se ha leído permanece en esa dirección temporalmente.

Hay principalmente dos tipos de memorias las semiconductoras y las magnéticas.

Como una regla general las unidades de memoria tienen capacidad para uno u 8 bits. Donde el bit es la unidad más pequeña de información.

LA MEMORIA DE 8 BITS

Una característica fundamental de las memorias es que las unidades en las que se almacenan los datos pueden guardar de uno a 8 bits, cuando se tiene un grupo de 8 bits se les conoce como byte.

Un bit o un grupo de bits, constituye lo que se conoce como palabra, cada una de estas palabras constituye una unidad de información.

Las memorias de 8 bits se conocen por ser los elementos de memoria básicos.

Las memorias están divididas en celdas cada una de estas puede tomar dos valores 0 o 1.

A continuación, se muestra un arreglo de 64 celdas en las cuales se tienen 8 filas y 8 columnas.

1								
2								
3								
4								
5								
6								
7								
8								
	1	2	3	4	5	6	7	8

Figura 96 Arreglo de 64 celdas

Capacidad de una memoria

La capacidad de una memoria se determina por el número de bits que puede almacenar.

Por ejemplo, la capacidad de una memoria de 8x8 es de 64 bits

Lo anterior se puede expresar como:

Capacidad de una memoria = Número de bits

Dirección o Localidad de memoria

Como se había visto anteriormente una unidad de memoria puede contener un bit o más de uno.

Cuando se desea conocer la dirección o localidad de un bit se proporciona la columna y el renglón en el que se encuentra.

Cuando se tiene más de un bit solo se proporciona el renglón.

De manera general, los datos son introducidos a la memoria a través de la operación de escritura y son extraídos con la operación de lectura.

Los dos tipos de memoria semiconductora de gran importancia son la RAM y la ROM.

Veamos en que radica la diferencia entre ellos:

16.2 LAS MEMORIA RAM

Las memorias RAM conocidas también como memoria de acceso aleatorio son conocidas como memorias volátiles ya que, si bien es cierto que estas memorias permiten leer y escribir, la información se pierde cuando se va la energía eléctrica.

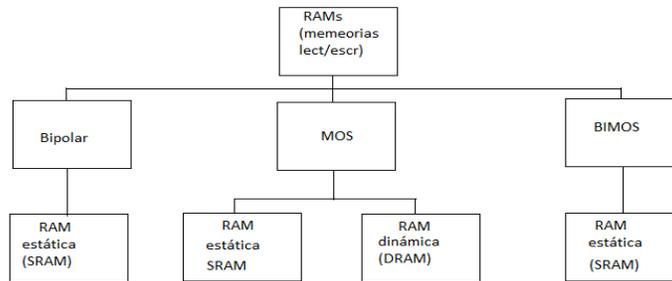


Figura 97 Cuadro sinóptico de las memorias RAM. Cuadro tomado de la página 602 del libro Tomas Floyd

Las memorias RAM son fabricadas con tecnología bipolar o MOS.

Cuando las memorias RAM se elaboran con elementos bipolares todas ellas son de tipo estático.

En contraste cuando las memorias RAM son elaboradas con tecnología MOS podemos encontrar tanto de tipo estático o de tipo dinámico.

COMO SE PROGRAMAN

Los CI están compuestos de transistores y capacitores y la memoria RAM no es la excepción, los transistores son aquellos elementos eléctricos que regulan la corriente en los circuitos.

El capacitor es un elemento eléctrico que sirve para almacenar energía.

Si se combina un transistor con un capacitor se crea una celda de memoria, las cuales como se vio anteriormente cada una puede almacenar un 0 o un 1.

Para obtener el valor de 0 o 1 hacemos referencia a la capacidad del capacitor.

Si se encuentra en un 50% o más se tendrá un 1.

Si el capacitor está a menos del 50% se tendrá un cero.

Para programar la memoria RAM se selecciona la celda o un grupo de celdas activando con un 1 (altas), las entradas en fila y columna.

El pin que está dedicado a lectura/escritura se habilita de la siguiente manera, si entra un 0 se lleva a cabo el proceso de lectura, el conjunto de datos que se encontraba guardado el cual es enviado a la línea de salida, si se activa un 1 el bit que se encuentre en la entrada de datos queda escrito en la celda.

16.3 LAS MEMORIAS ROM

Estas son conocidas como memorias no volátiles y aunque la energía eléctrica se vaya se sigue conservando la información permanentemente o semipermanentemente, en este tipo de memoria solo se puede escribir.

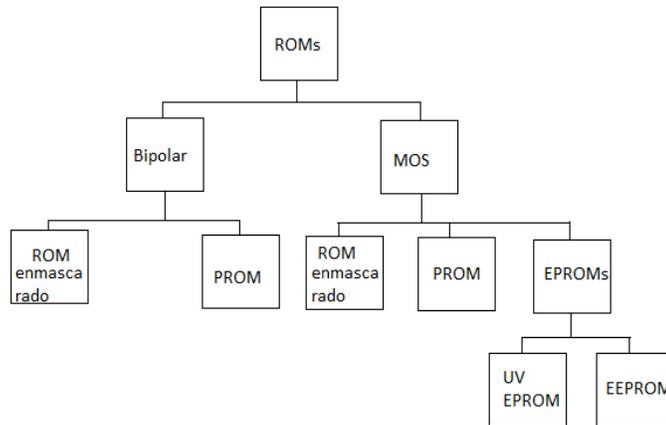


Figura 98 Cuadro sinóptico de las memorias ROM. Cuadro tomado de la página 588 del libro de Tomas Floyd

La memoria ROM de enmascaramiento es aquella donde los datos se almacenan de forma permanente.

La memoria PROM o ROM programable es aquella que con ayuda de equipo especializado se puede realizar la programación.

Cuando se pidió que observara a su alrededor uno de los sistemas con mayor aplicación para la adquisición y transmisión de datos fue la computadora.

En efecto las computadoras es un claro ejemplo donde se utilizan los dos tipos de memoria RAM y ROM

La memoria ROM se emplea generalmente para tratar funciones fundamentales de supervisión y soporte para la computadora.

APLICACIONES DE LA MEMORIA ROM

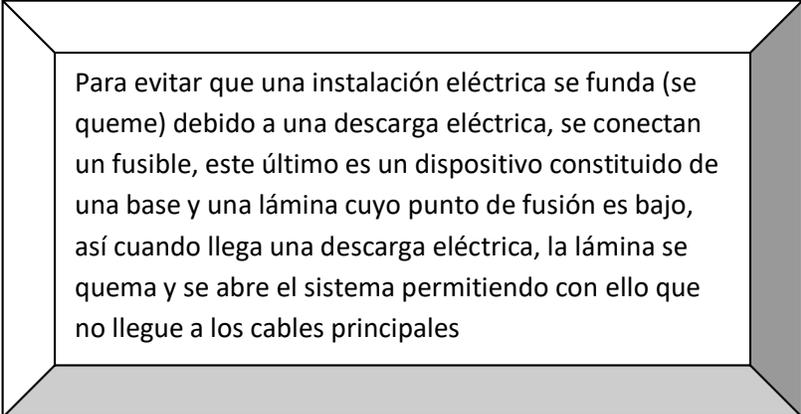
El campo de aplicaciones de la memoria ROM es muy amplio y variado, tal memoria la podemos encontrar en computadoras, celulares, reproductores de audio, tablets y mucho más.

La memoria ROM contiene la información necesaria para llevar a cabo las funciones de arranque es decir la forma imprescindible de las entradas y salida del dispositivo para que operen correctamente.

Las memorias ROM se dividen en dos grupos principales EPROM y EEPROM las cuales han llegado a reemplazar a la memoria ROM.

Memoria PROM

Para programar las memorias PROM se utiliza un instrumento especial conocido como Programador PROM. La característica principal de las memorias PROM es que cuentan con fusibles que pueden ser utilizados solo una vez ya que durante el proceso de grabado de la información se queman y no pueden volver a utilizarse.



Para evitar que una instalación eléctrica se funda (se queme) debido a una descarga eléctrica, se conectan un fusible, este último es un dispositivo constituido de una base y una lámina cuyo punto de fusión es bajo, así cuando llega una descarga eléctrica, la lámina se quema y se abre el sistema permitiendo con ello que no llegue a los cables principales

Para la programación de la PROM se toma en cuenta que todos los valores inician en 1, posteriormente se manda el código deseado mediante las líneas de salida, a las líneas de salida que tienen un cero, se les manda un pulso mediante el programador de EPROM que funde el fusible, obteniendo el código deseado.

EPROMS

La característica de estas memorias es que pueden volver a programarse se borra primero el programa almacenado en dicha memoria.

Se tienen principalmente dos tipos de memoria programables de este tipo, las que se borran con luz ultravioleta y las que se borran eléctricamente a las primeras se les conoce como UV EPROM y a las segundas EEPROM

UV PROMS

La característica de esta memoria es que cuenta con una membrana de cuarzo transparente en la parte superior. Dentro se encuentra flotando la compuerta de un FET. Cuando se está programando la memoria, los electrones se remueven de la compuerta flotante.

Cuando se expone a la ultravioleta la parte de la memoria que contiene el cuarzo se neutraliza la carga positiva.

EEPROM

Para programar o borrar información de estas memorias solo se necesitan emplear pulsos eléctricos.

Debido a que los cambios pueden producirse por el proceso antes descrito, otro nombre que reciben estas memorias son EAPROM que significa memorias alterables eléctricamente.

FLASH. Estas memorias son utilizadas para el almacenamiento y traslado de datos. Este tipo de memorias se utilizan en una gran variedad de dispositivos como USB, cámaras digitales, reproductores de MP3 por mencionar algunos.

17. Circuitos secuenciales (Circuitos secuenciales latch SR,D, JK y T y flip-flops SR,T,JK)

17.1 INTRODUCCIÓN

Imagine por un momento que se encuentra estudiando.

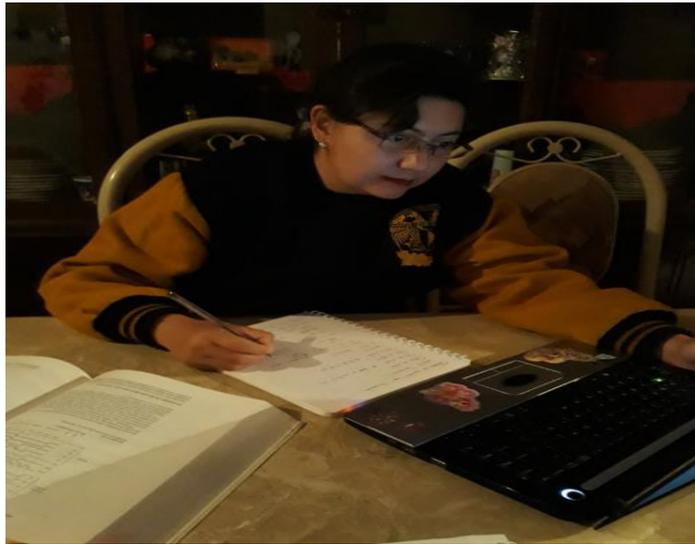


Figura 99 Imagen autora estudiando Arlette Paola Olvera F

Para llevar a cabo el proceso de estudio se emplea la memoria, es decir esa capacidad que poseen los seres humanos para recordar todos los procesos, ideas, hechos, lugares, por mencionar solo algunos procedimientos que se experimentan diariamente.

Algo similar ocurre con algunos circuitos, se emplean ciertos elementos que permiten tener memoria.

En este apartado se estudiarán los latch y los FF los cuales son circuitos con capacidad de memoria.

17.2 LOS LATCH.

Los Latch son elementos de memoria asíncronos, es decir que no cuentan con un pulso de reloj que los coordine.

Se tienen diversos tipos de Latch los cuales se presentan a continuación:

Latch SR

Este se puede construir con dos compuertas NAND O NOR.

Construcción de un Latch con compuertas NOR.

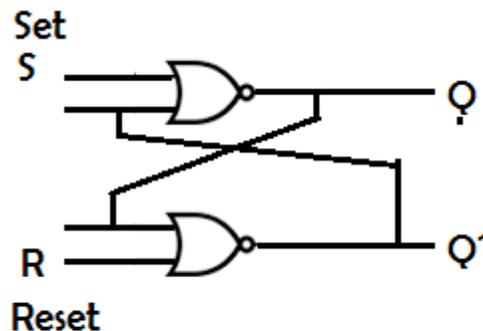


Figura 100 Latch SR

En este diagrama podemos apreciar las dos salidas donde una salida es la inversa de la otra como:

Q y \bar{Q}

Así se tiene:

$$Q = 1 \quad \bar{Q} = 0$$

$$Q = 0 \quad \bar{Q} = 1$$

Como se nota en el diagrama se tienen dos entradas nombradas como S y R las cuales se nombran respectivamente Set y Reset o ajustar y borrar respectivamente.

Se dice que se encuentra en el estado Set cuando esta entrada se encuentra en alto y Reset en estado bajo.

La tabla de verdad se muestra a continuación:

S	R	Q	Q'
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

Tabla 61 tabla de verdad del latch SR

Tabla característica

R	S	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	*
1	1	1	*

Tabla 62 Tabla característica de latch SR

Ecuación característica

$$Q_{t+1} = S + \bar{R}Q_t$$

$$RS = 0$$

CONSTRUCCIÓN DE UN LATCH CON COMPUERTAS NAND

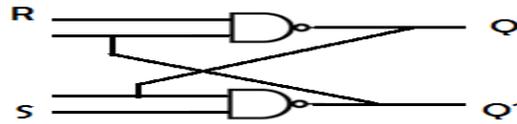


Figura 101 Latch con compuertas NAND

Tabla de verdad

S	R	Q	Q'
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
1	0	1	1

Tabla 63 Tabla de verdad de latch SR con compuertas NAND

Tabla característica

R	S	Q_t	Q_{t+1}
0	0	0	*
0	0	1	*
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Tabla 64 Tabla cacteristica de latch SR con compuertas NAND

Reduciendo el circuito con un Mapa de Karnaugh

		SQt			
R		00	01	11	10
	0	*	*	0	0
	1	1	1	0	0
		$Q_{t+1}=S'$			

Ecuación característica

$$Q_{t+1} = S'$$

FLIP-FLOPS

En la práctica de la electrónica para que muchos dispositivos funcionen se hace necesario el empleo de la lógica secuencial síncrona o temporizada, es decir cuyos elementos dentro del circuito se encuentren controlados por pulsos de reloj, la base de estos circuitos son los elementos de memoria conocido como FF. La base de construcción de los FF son las compuertas. La capacidad de almacenamiento de los FF es de un bit. Una característica principal de los FF es que tienen dos salidas una para el valor en la salida y otra para el complemento.

Este elemento de memoria se encuentra combinado por un conjunto de compuertas lógicas, la forma en la cual se conecten se lo que permita que funcionen como elementos de memoria.

El diagrama de bloques para un FF es el siguiente:

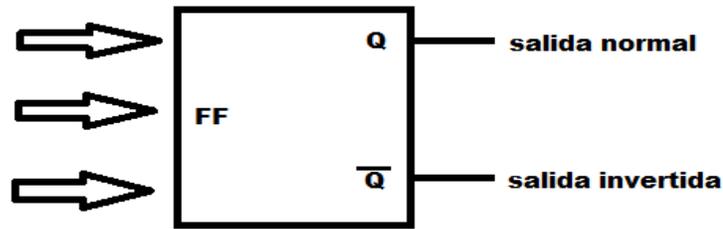


Figura 102 Diagrama de bloques de un FF

Como podrá observar existe una retroalimentación entre las salidas y las entradas.

En el diagrama 1 cuando $Q=1$ se tiene una entrada 1 en la compuerta B de 1 y la salida será $\bar{Q} = 0$, esta salida será la entrada de A y entonces la salida será $Q=1$

En el siguiente circuito se tiene $Q=0$ esta salida será la entrada de la compuerta B' la salida será $\bar{Q} = 1$, esta salida será la entrada de la compuerta A y a la salida se tendrá $Q=0$

Es muy importante destacar que existen 4 posibles estados en los cuales se puede encontrar el FF de compuertas NAND

Cuando $SET=1$ y $RESET=1$

Como esta condición es prácticamente de reposo los valores que tendrán Q y \bar{Q} serán las que hayan tenido antes de estas entradas.

$SET=0$, $RESET=1$

Con estas entradas siempre se mantendrá el valor $Q=1$ a lo anterior se le conoce como establecer.

SET=1 RESET=0

Con estas entradas siempre se mantendrá el valor $Q=0$, a lo anterior se le conoce como reestablecer el FF

SET=0 RESET=0

Con estas condiciones de entradas se produce como salidas que $Q=1$ y $\bar{Q} = 1$, con lo cual el resultado que se obtenga se vuelve incierto, esto se debe a que no se está cumpliendo con hecho de que las salidas deben ser inversas o una el complemento de la otra.

DIAGRAMA DE FLIP-FLOP SR

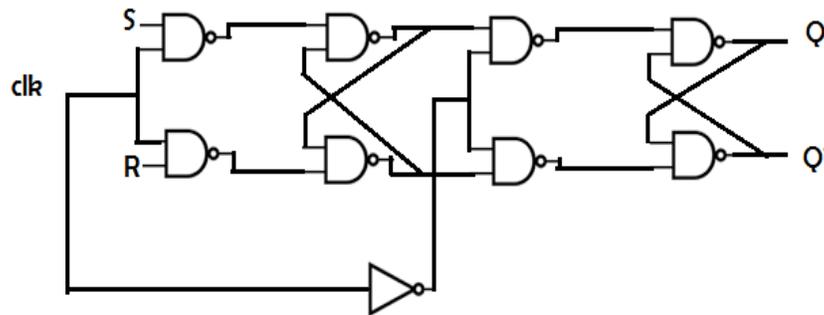


Figura 103 Diagrama de un FF SR

TABLA DE VERDAD

S	R	Q	Q'
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0

Tabla 65 Tabla de verdad de FF SR

TABLA CARACTERÍSTICA

R	S	Q_t	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	*
1	1	1	*

Tabla 66 Tabla de verdad de la ecuación característica de un FF SR

Reduciendo la función en un Mapa de Karnaugh para obtener la ecuación característica

		SQ_t			
		00	01	11	10
R	0	0	1	1	1
	1	0	0	*	*

Ecuación característica

$$Q_{t+1} = S + \bar{R}Q_t$$

$$SR = 0$$

OBTENIENDO LA FUNCIÓN EN SU FORMA REDUCIDA

	D	0	1
Q	0		1
1			1

Tabla 69 Mapa de Karnaugh de Función reducida del FF D

ECUACIÓN CARACTERÍSTICA $Q_{T+1} = D$

FLIP-FLOP JK

La característica principal de este FF (JK) es que no presentan estados indeterminados.

Es decir, cuando se presentan entradas simultáneas, la salida cambia de valor a estado complementario.

Cuando los dos valores de entrada son iguales a 1 la señal de reloj emplea solamente una compuerta NAND para transmitirse.

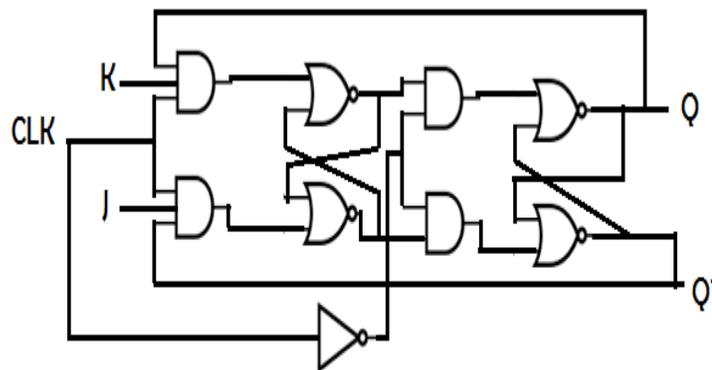


Figura 105 FF JK

TABLA DE VERDAD

J	K	Q	Q'
0	1	0	1
0	0	1	0
1	0	1	0
1	1	1	0

*Figura 106 Tabla d verdad de FF JK***TABLA CARACTERISTICA**

Q	J	K	Q _{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

*Figura 107 Tabla característica de FF JK***REDUCIENDO LA FUNCIÓN EN UN MAPA DE KARNAUGH**

JK	00	01	11	10
Q	0	0	1	1
1	1	0	0	1

Figura 108 Mapa de Karnaugh de FF JK

ECUACIÓN CARACTERÍSTICA

$$Q_{t+1} = JQ' + K'Q$$

FLIP-FLOP T

Para la construcción del FF T se toma un FF JK se unen las entradas y esta entrada única será la del FF T. Se conoce como FF T por T de Toggle cuyo significado es que tiene la capacidad de conmutar, es decir puede cambiar de un estado a otro cuando la entrada se encuentra en 1.

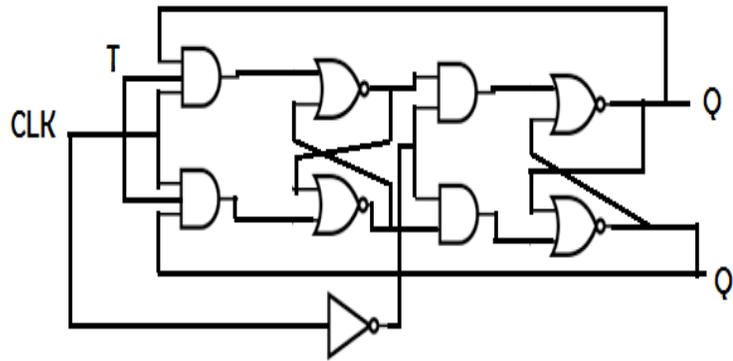


Figura 109 FF T

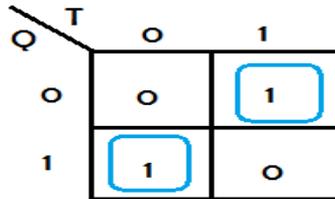
TABLA DE VERDAD

T	Q	Q'
0	0	0
1	1	1

Tabla 70 Tabla de verdad de FF T

TABLA CARACTERISTICA

Q	T	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

*Tabla 71 Tabla característica de FF T***OBTENIENDO LA FUNCIÓN REDUCIDA EN UN MAPA DE KARNAUGH***Mapa de Karnaugh 52 De FF T***OBTENIENDO LA ECUACIÓN CARACTERISTICA**

$$Q_{t+1} = QT + Q'T$$

NOTAS:

Nota: Un FF es un arreglo de 2 latch

Los latch se habilitan con un nivel alto o bajo

Los FF se activan con una serie de pulsos es decir con un nivel alto o bajo; si la tecnología es TTL con un 1 o con un cero.

Si el circuito es asíncrono se tiene un latch, es decir podemos tener uno tipo SR, JK o T, en cambio si el circuito tiene un reloj se tiene un FF (Flip-Flop)

SISTEMAS SINCRONOS Y ASINCRONOS

Imagine que va a un parque a correr y solo quieres tomar el aire y disfrutar del ejercicio y no lleva reloj.

No obstante, ahora imagine que estas en una pista de carrera con otros competidores ¿Cuál será la herramienta básica para tomar tu tiempo?



Fig. 122 Imagen de corredor con reloj en mano

La respuesta es un reloj



Fig.123 Imagen de reloj

Algo similar ocurre con los circuitos cuyas entradas no están temporizadas se conocen como asíncronos. Y aquellos cuyas entradas están controladas por un reloj se conocen como síncronos

En electrónica este reloj es una señal que se da como pulsos de onda cuadrada.

Nota: Cuando un circuito cuenta con reloj formalmente se maneja que se trata de un FF, en caso de que el circuito no cuente con reloj se dice que se tiene un latch.

Donde FS es el flanco de subida, es importante mencionar que también se pueden activar por un flanco de bajada o FB.

Cuando se tiene que las entradas de control tienen un valor de 0 la salida $Q=0$

Nota: Recordar que todos los cambios programados con una señal de reloj pueden ser por flancos positivos o flancos negativos.

Cuando se dan los cambios con flancos negativos el cambio de nivel es de 1 a cero.

El diagrama y el cuadro se presentan a continuación:

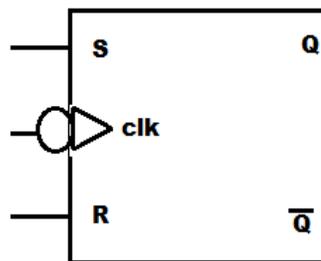


Figura 110 FF SR con flanco negativo

17.3 Registros y contadores

Los FF tienen muchas aplicaciones una de ellas es que cuando se agrupan pueden funcionar como registros.

Gracias a los registros podemos llevar un sin número de operaciones en las máquinas una de ellas aplicaciones típicas como la transmisión de datos.

Clasificación de los registros:

Los registros se clasifican en base a la forma en cómo se introducen y se envían los datos tanto a la entrada como a la salida del registro respectivamente.

Nombre	Entrada	Salida
PIPO	Paralelo	Paralelo
SISO	Serie	Serie
PISO	Paralelo	Serie
SIPO	Serie	Paralelo

Ejemplos de registros.

Primer ejemplo:

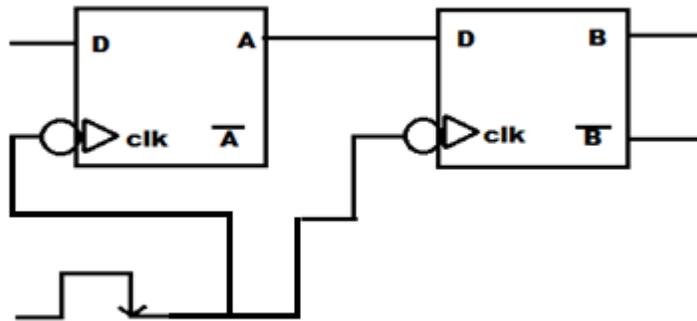


Figura 111 FF D conectado para formar un registro

Estos datos se transfieren mediante un NGT es decir un pulso de entrada negativo

Segundo ejemplo:

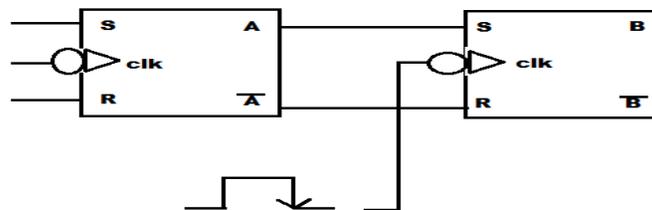


Figura 112 Dos FF SR conectados para formar un registro

Estos datos se transfieren mediante un NGT es decir un pulso de entrada negativo

Tercer ejemplo:

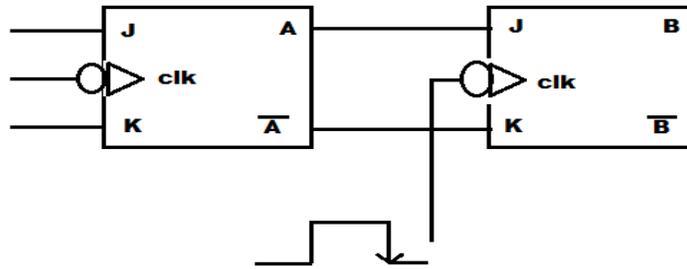


Figura 113 Dos FF JK conectados para formar un registro

Estos datos se transfieren mediante un NGT es decir un pulso de entrada negativo

La salida que se encuentre en B será la misma que tenga A.

Como podrás notar en los ejemplos anteriores el pulso NGT se encontraba en el clk por lo tanto en este caso tenemos una transferencia síncrona.

Como transferir datos en paralelo

Como un ejemplo de transferencia de datos en paralelo se ha seleccionado un conjunto de FF JK para hacer los registros, se han seleccionado de este tipo debido a que estos no presentan ambigüedades cuando $J=K=1$ y con el uso de reloj forzamos al sistema a que se a síncrono y tenga una mejor en la funcionalidad, es decir que pueda mejorar los tiempos de respuesta ya que si no se presentan ambigüedades el sistema presenta un grado más alto de confiabilidad en la respuesta.

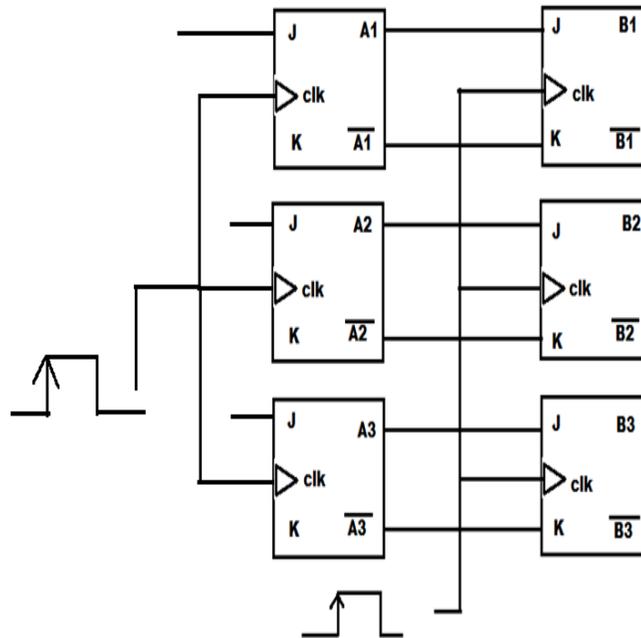
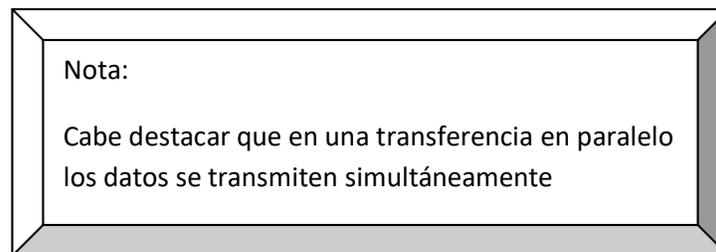


Figura 114 FF JK conectados en paralelo para formar un registro



La siguiente pregunta que muy seguramente está pensando es:

¿Y cómo se lleva a cabo la transmisión de datos en serie?

Imagine por un momento que está muy contento debido a que se encuentra trabajando en un proyecto muy importante y emplearas un LCD y un teclado.



Figura 115 Imagen de un mensaje en pantalla LCD

Cuando se programa que los dígitos aparezcan en la pantalla LCD conforme se van tecleando los dígitos, estos van apareciendo uno por uno en la pantalla y aparece una barra en espera del siguiente dígito, si nota que hay un desplazamiento en los lugares.

De igual manera sucede con la transferencia de datos en serie. De un conjunto de bits.

Como ejemplo se muestra un FF D para mostrar la conexión.

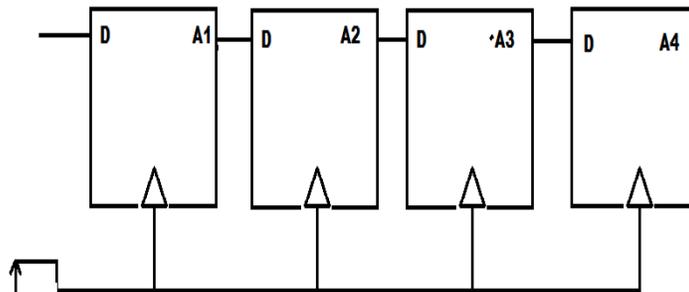


Figura 116 FF D conectado en serie

El registro anterior es de 2 bits

A continuación, se muestra la tabla de un registro de corrimiento.

X1	X0	Y1	Y0
1	1	0	0
0	1	1	1
1	1	1	0
0	0	1	1

Tabla 72 Registro de corrimiento

17.4 CONTADORES

Cuando se conectan dos o más FF se puede formar un circuito conocido con el nombre de contador.

Los contadores se dividen en 2 grupos: los síncronos y los asíncronos, los primeros se caracterizan por que a la entrada de CP entran el pulso de reloj, t los asíncronos funcionan con las transiciones del pulso.

17.5 CONTADORES SÍNCRONOS

Como se había mencionado anteriormente un contador se conforma por la conexión de FF. Los contadores síncronos se caracterizan por que todos los FF se encuentran controlados por el mismo pulso de reloj.

A continuación, se muestra un diagrama de un contador de 3 bits.

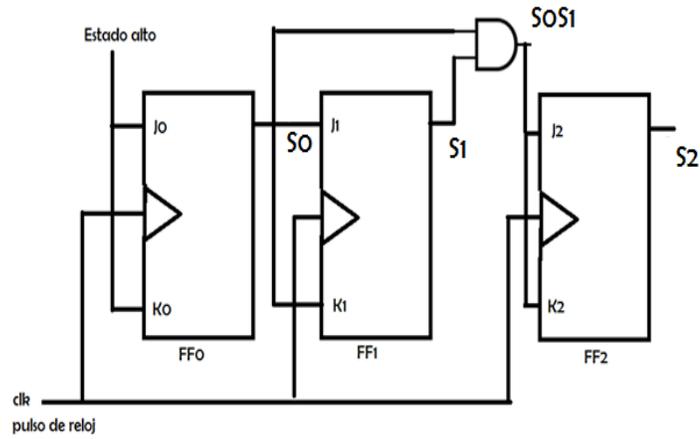


Figura 117 Contador de 3 bits síncrono

La tabla se muestra a continuación

Pulso de reloj	Salida S2	Salida S1	Salida S0
Estado inicial	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

Tabla 73 Contador de tres bits síncrono

La forma de operar de este contador es la siguiente:

Q0 mantendrá un estado oscilante en base a los pulsos de reloj que reciba.

La respuesta de Q1 dependerá de lo que este en Q0.

Si $Q0=0$ $Q1=1$

Si $Q0=1$ $Q1=0$

Estos resultados afectarán por ende la entrada en Q2.

Es importante observar que Q2 se mantiene en el estado de RESET o bajo hasta que $Q0=1$ y $Q1=1$. Solo cuando se presenta la condición el FF2 oscilara es decir cambiara su valor.

Los contadores pueden ser ascendentes o descendientes, otra característica es que se dice son líneas, esto significa que se conectan en serie.

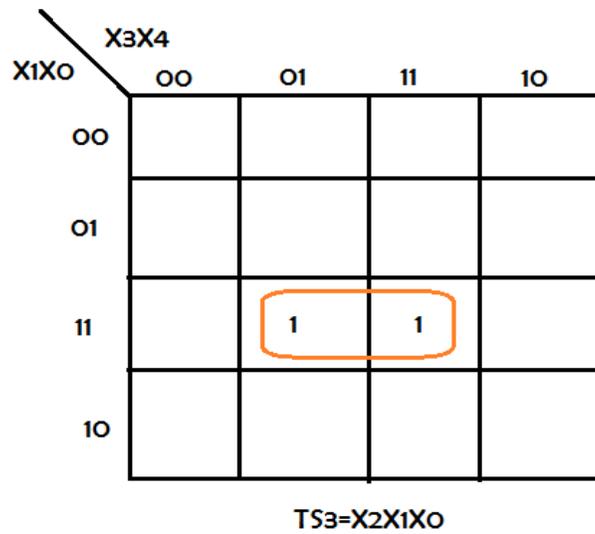
A continuación, se muestra un ejemplo de contador ascendente del número del 0 al 15 de tipo síncrono.

Estado presente

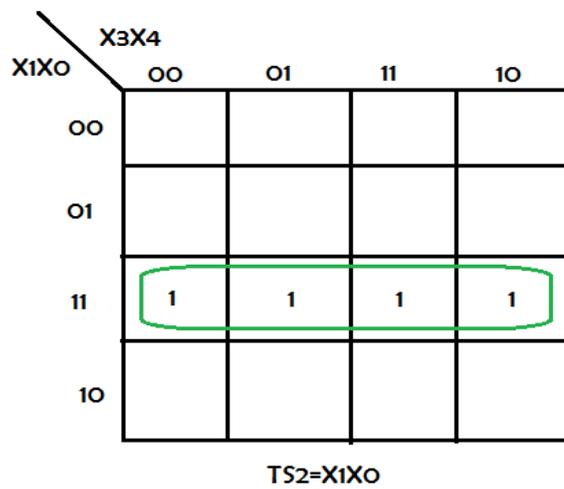
Estado siguiente

X3	X2	X1	X0	TS3	TS2	TS1	TS0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	1
0	0	1	0	0	0	0	1
0	0	1	1	0	1	1	1
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	1
0	1	1	0	0	0	0	1
0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1
1	0	0	1	0	0	1	1
1	0	1	0	0	0	0	1
1	0	1	1	0	1	1	1
1	1	0	0	0	0	0	1
1	1	0	1	0	0	1	1
1	1	1	0	0	0	0	1
1	1	1	1	1	1	1	1

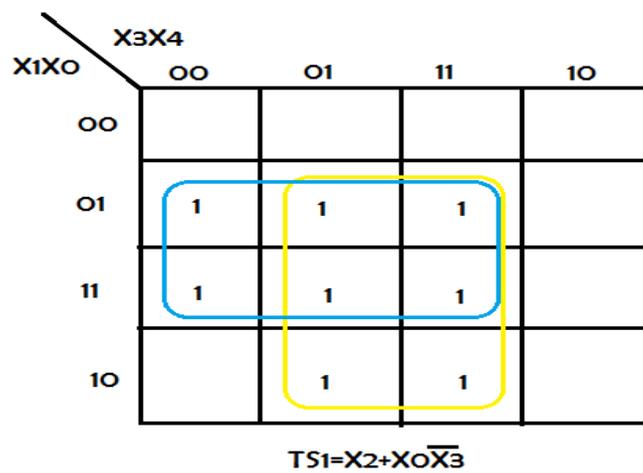
Obteniendo los Mapas de Karnaugh correspondientes



Para TS2



Para TS1



Para TS0

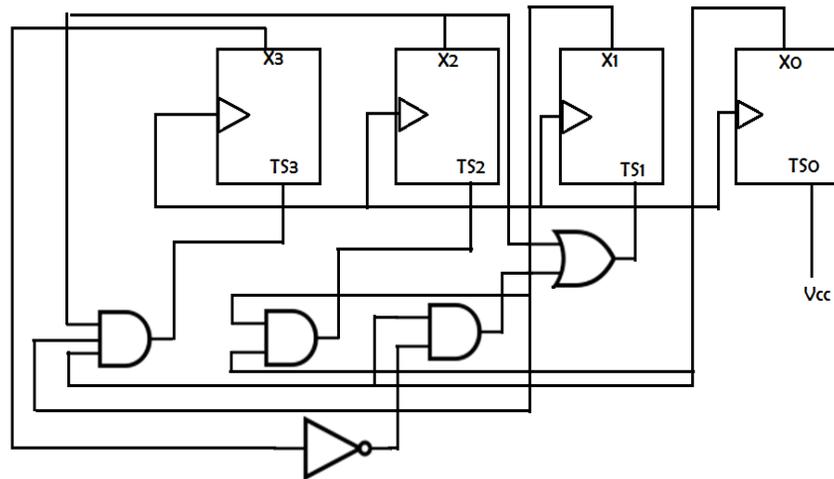
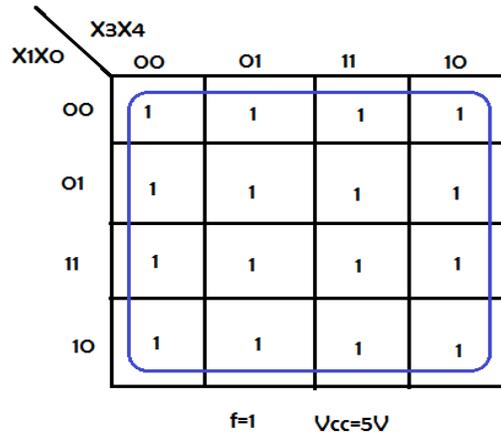


Figura 118 Implementación de contador ascendente del 0-15

17.6 CONTADORES ASINCRONOS

La característica de estos contadores es que solamente cuentan con una entrada de reloj en uno de los FF debido a esto el tiempo de respuesta es mayor para llegar a los otros FF lo que hace que el sistema sea muy inestable o fluctuante.

A continuación, se presenta un contador asíncrono de 3 bits y la tabla respectiva.

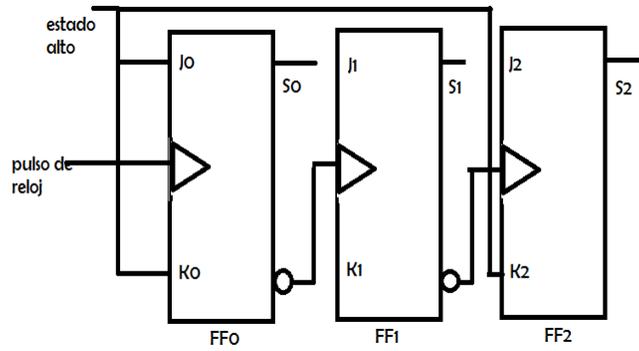


Figura 119 FF de 3 bits asíncrono

En un estado inicial todos comienzan en RESET es decir en cero después del primer pulso de reloj Q0=1, Q1=0 y Q2=0

Pulso de reloj	S2	S1	S0
Estado inicial	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

18. Máquina de estado

18.1 INTRODUCCIÓN

Antes de tratar el tema expliquemos brevemente que es un circuito lógico secuencial.

Los botones de avance y retroceso en el control de una televisión son un ejemplo.

El estado o momento de la película dependerá de la selección del momento en el cual se desea ver.

Definición de estado tomado del libro Herbert Hellerman. Digital Computer Systems Principles (Mc Graw Hill, 1967)

“El estado de un circuito secuencial es una colección de variables de estados cuyos valores en cualquier tiempo contienen toda la información acerca del pasado necesario para explicar el comportamiento futuro del circuito”

En base a la definición anterior cabe hacerse la siguiente pregunta:

¿Cuántos estados posibles pueden tener un sistema?

Cuando se tienen n variables se aplica la fórmula:

$$2^n$$

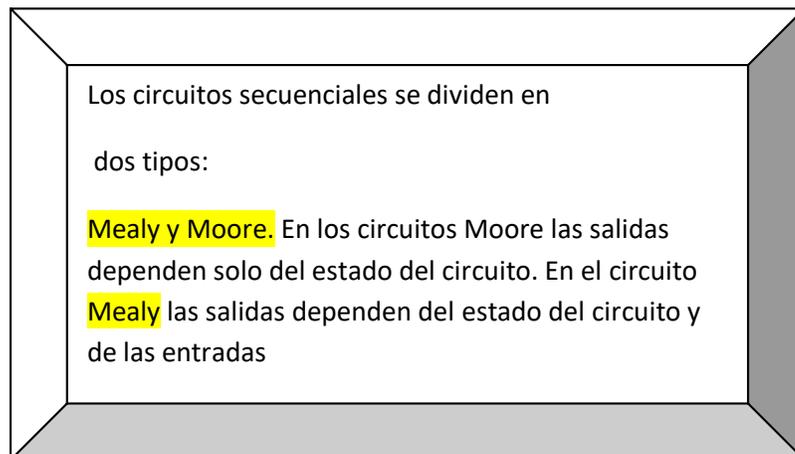
Este número puede ser muy grande pero su característica principal es que es finito, es decir que tiene como característica principal que sus estados tienen un número finito de resultados, por lo tanto, es posible trabajar con ellos, esto es necesario porque si el sistema contara con una serie de estados finitos el ultimo estado no tendría retorno por lo cual se volvería indeterminado.

Otro nombre para designar a los circuitos secuenciales cuya característica principal es ser finito es máquinas de estado finito.

Como se había visto anteriormente la salida de los circuitos secuenciales depende de las entradas que se tenga.

En cambio, en los circuitos secuenciales las salidas no dependen solamente de las entradas si no del estado del circuito en el estado pasado y presente.

Los elementos que componen a este tipo de circuitos (secuenciales) son circuitos combinacionales y ff (flip-flops)



18.2 ESTRUCTURA PARA CONSTRUIR MÁQUINAS MEALY Y MOORE

Crear el algoritmo

Crear el diagrama de estado

Asignar las etiquetas al diagrama de estado

Codificar las etiquetas del diagrama de estado

Obtener la tabla de estados

Seleccionar los elementos de memoria

Elaborar los respectivos mapas de Karnaugh

Obtener las ecuaciones de estado.

NOTA: UN DIAGRAMA DE ESTADO

Para representar el comportamiento de un sistema secuencial se puede emplear el diseño conocido diagrama de estado, este consta de nodos, transiciones, entradas, con las cuales se puede describir el flujo del sistema.

Ejemplo de la construcción de un diagrama de estado.

Se tiene una máquina que entrega un refresco, una hamburguesa, unas papas y el postre. Cuando ha entregado todo vuelve a iniciar el ciclo, si falta uno de estos elementos en la entrega la maquina no avanza.

A Despecha refresco

B Despacha hamburguesa

C Despacha papas

D Despacha postre

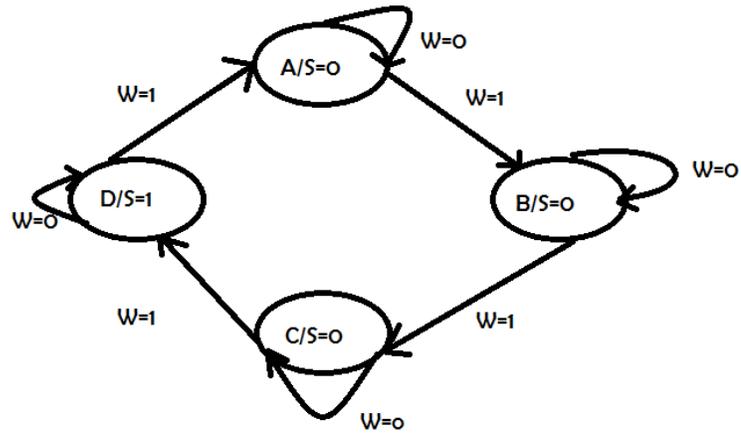


Figura 120 Diagrama de estado de la máquina expendedora de alimentos

Estado presente	Estado siguiente W=0	Estado siguiente W=1	Salida
x2x1	X2Y1	X2Y1	
A 00	A	B	0
B 01	B	C	0
C 10	C	D	0
D 11	D	A	1

Obteniendo los Mapas de Karnaugh de salida

	x_2x_1	00	01	11	10
W	0			1	1
1			1		1

$$X_2 = \bar{w}x_2 + x_2\bar{x}_1 + wx_2x_1$$

Mapa de Karnaugh 53 Mapa de Karnaugh de máquina expendedora

	x_2x_1				
w		1	1		
	1	1			1

$$X_1 = x_1 + w\bar{x}_1$$

Mapa de Karnaugh 54 Correspondiente a máquina expendedora

	x_1	0	1
x_2	0		
1			1

$$S = x_2x_1$$

Mapa de Karnaugh 55 Correspondiente a Máquina expendedora

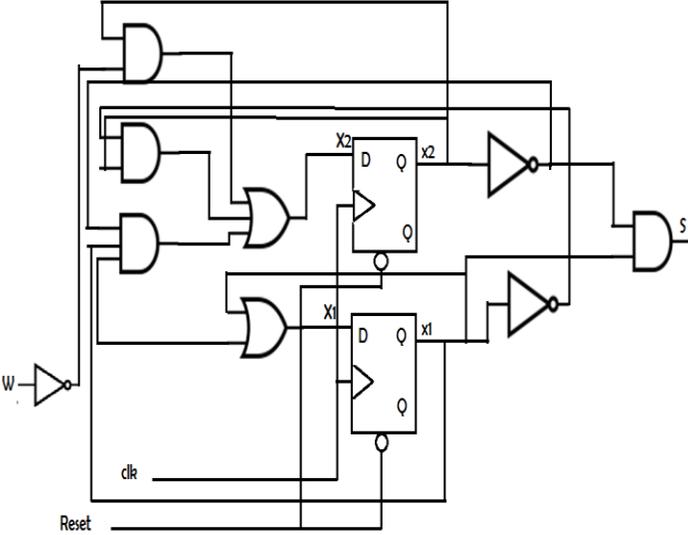


Figura 121 Implementación del circuito de la máquina expendedora

18.3 MÁQUINAS DE ESTADO

Las máquinas de estados representan el comportamiento de un circuito lógico secuencial controlado por distintas señales.

Lo más importante de estos es un pulso de reloj, aunque podrían también seguir un comportamiento asíncrono.

Por lo general esta secuencia de pasos o algoritmo es conocido como ASM (algoritmos de máquinas de estados) ayuda a la descripción de la secuencia que seguirá el circuito.

Estas representaciones esquemáticas ayudan a una mejor comprensión del sistema.

Debido a las variables de entrada y salida que entran en juego en el análisis del circuito el algoritmo de la máquina de estado ASM hace más fácil su control ya que en cada uno de los cuadros se coloca el valor de los estados, el nombre, la dirección del flujo de información, así como las decisiones que se toman.

Una característica principal del Algoritmo de máquina de estado es que debe ser finito es decir que el número de estados debe ser un número concreto.

Otra característica de vital importancia es que se deben especificar todos los procedimientos que rigen el comportamiento de las acciones del circuito desde las entradas y las salidas hasta las decisiones.

Las máquinas de estado se clasifican principalmente en dos en modelo Mealy y Moore

Nota: El contador es una máquina de estados.

“En el estado Mealy el estado siguiente depende del estado anterior”

Por lo general la estructura de ASM tipo Mealy tiene salidas condicionadas es decir las señales dependen tanto de la entrada actual como del estado en el que se encuentra.

Ejemplo:

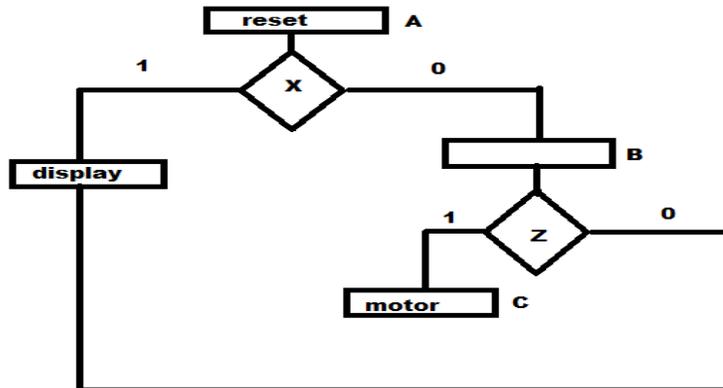


Figura 122 diagramas de una carta ASM

Entrada presente	Entrada X	Entrada Z	Estado futuro	Salidas
A	1	*	D	Display
A	0	*	B	
B	*	1	C	Motor
B	*	0	D	Display

Tabla 151. Tabla de máquina de estados del ejemplo anterior

El estado presente es el estado actual en el que se encuentra el sistema.

Las entradas corresponden a los respectivos cuadros de decisión

El estado futuro corresponde al estado siguiente del estado presente

Para este caso se tienen definida una serie de procesos que el brazo robot debe cumplir.

Solo hasta que se cumpla correctamente la tarea, el robot hará la siguiente función, cuando se han cumplido todas las tareas es decir que la cirugía ha sido exitosa el robot termina su operación.

Realicemos su diagrama de estado.

Cartas ASM

DIAGRAMA DE FLUJO

Podemos definir un algoritmo como una secuencia de pasos a seguir para llevar a cabo una tarea. En los sistemas digitales, existe lo que se conoce como algoritmo de hardware.

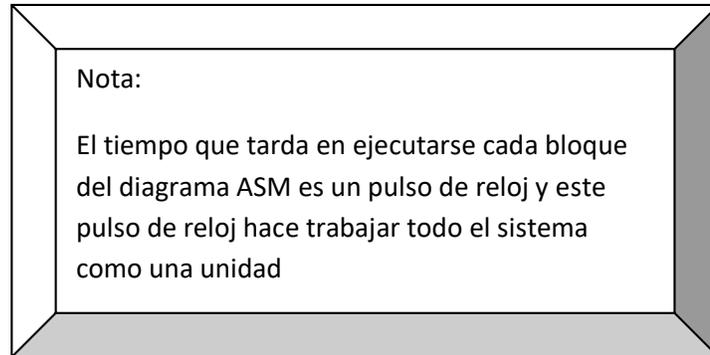
El algoritmo consiste en determinar en qué forma se llevarán a cabo el procesamiento de datos y las secuencias de control.

La característica de estos algoritmos es que son finitos, para llevar a cabo un control detallado de los procesos seguir se implementan los diagramas de flujo, en este diagrama se plasma el algoritmo elaborado en palabras de manera esquemática utilizando cierta simbología para la representación de las partes principales.

Estos diagramas se conocen como diagramas ASM, una característica importante de los diagramas ASM es que toman en cuenta la característica del sistema con mayor detalle.

Se puede pasar de un diagrama de estados a una carta ASM o de una carta ASM a un diagrama de estado la principal diferencia es que en las cartas ASM se muestran las

características del sistema con mayor detalle, permitiendo llevar un control mejor.



El diagrama ASM consta de tres elementos fundamentales:

Cuadros de estado

Cuadros de decisión

Cuadros de condición

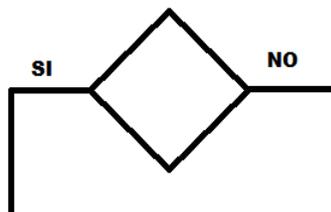
Su representación se muestra a continuación:

Cuadro de estado



Es un rectángulo en el cual se llevan a cabo las operaciones, como pueden ser las operaciones de registros. (Tipo Moore)

Cuadro de decisión



En este cuadro se evalúa si la condición es falsa o verdadera por lo cual presenta dos caminos posibles a seguir. Este cuadro recibe una entrada proveniente del estado y solo puede tener dos salidas

Cuadro de condición



Aquí la entrada del cuadro de condición es la salida del cuadro de decisión. Es una caja de salida condicional (Tipo Mealy)

19. Quartus

INTRODUCCIÓN A QUARTUS

Como realizar un diseño y una simulación

Si tu respuesta es afirmativa puedes omitir este capítulo; en caso contrario aquí te guiaremos en un breve recorrido con captura de pantallas para que puedas ver como se emplea dicho programa.

PASO 1

Si no cuentas con el programa de Quartus lo puedes descargar de la página oficial de Altera

PASO 2

Una vez que cuentes con el programa de Quartus aparecerá el siguiente símbolo en tu pantalla.



Figura 123 Icono de Quartus

PASO 3

Da doble click sobre el icono y aparecerá la siguiente pantalla

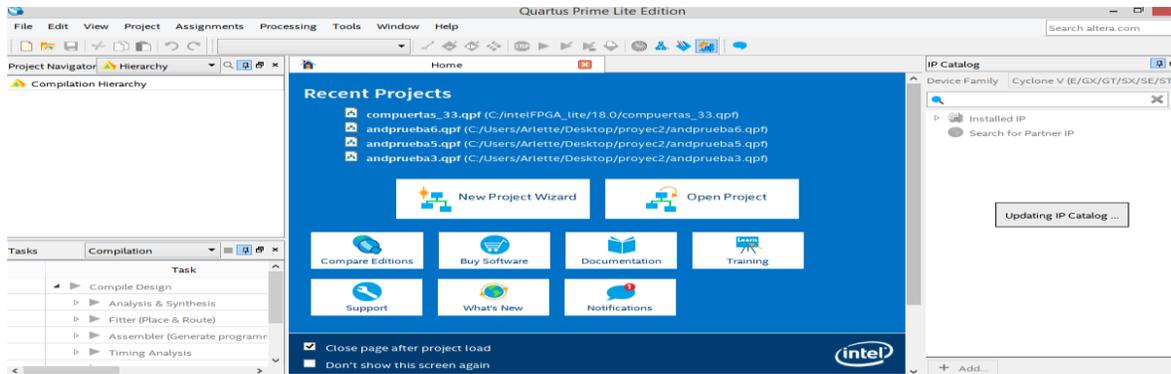


Figura 124 Presentación del paso 3 y 4

PASO 4

Selecciona la opción New Project Wizard

PASO 5

Elige la opción Next en el siguiente recuadro

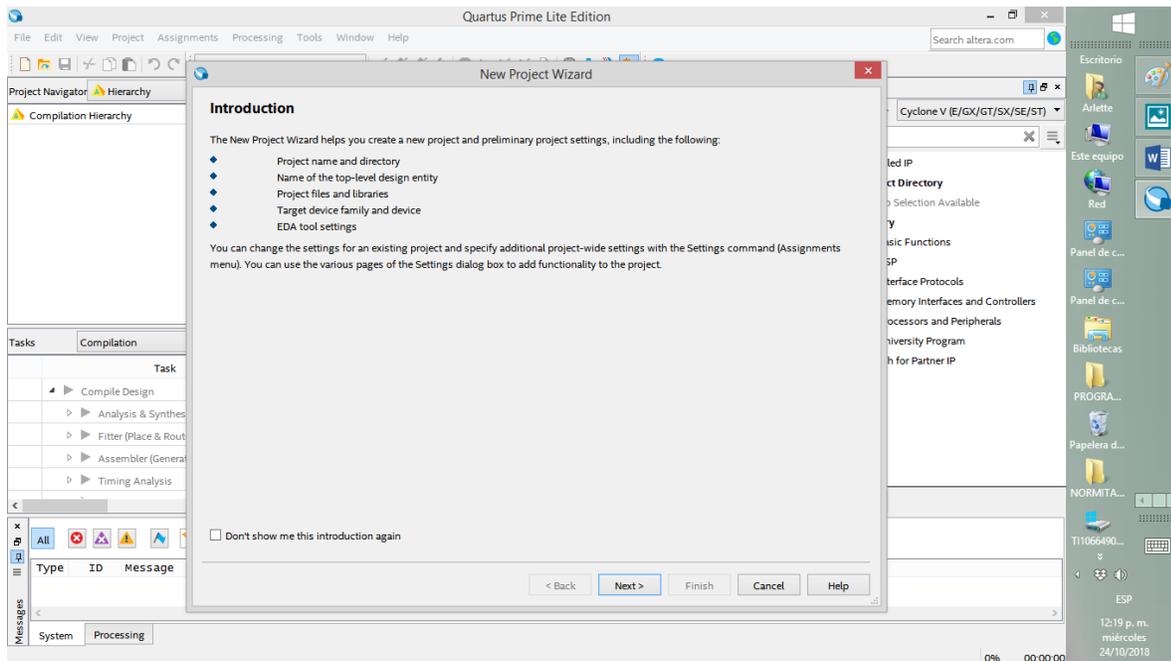


Figura 125 Presentación del paso 5

PASO 6

El primer recuadro indica la dirección en la cual se aguardarán tus proyectos. Para terminar esta etapa es necesario llenar el siguiente recuadro con el nombre que

llevar a cabo el proyecto es importante no utilizar palabras reservadas, no dejar espacios y no utilizar caracteres especiales, para evitar errores en la compilación.

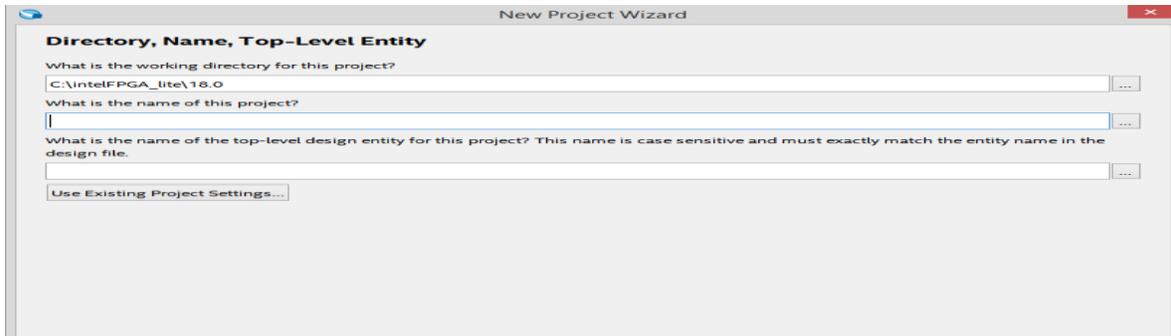


Figura 126 Presentación del paso 6

PASO 7

Una vez que escribiste el nombre del proyecto da un click en next

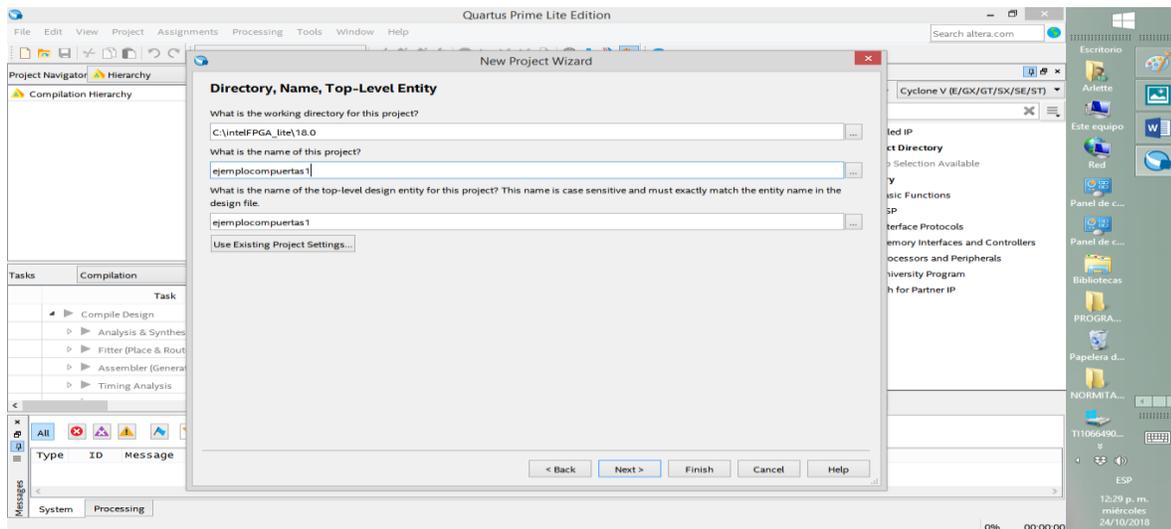


Figura 127 Presentación del paso 7

PASO 8 (OPCIONAL)

En caso de que te aparezca un mensaje como el siguiente selecciona la opción No

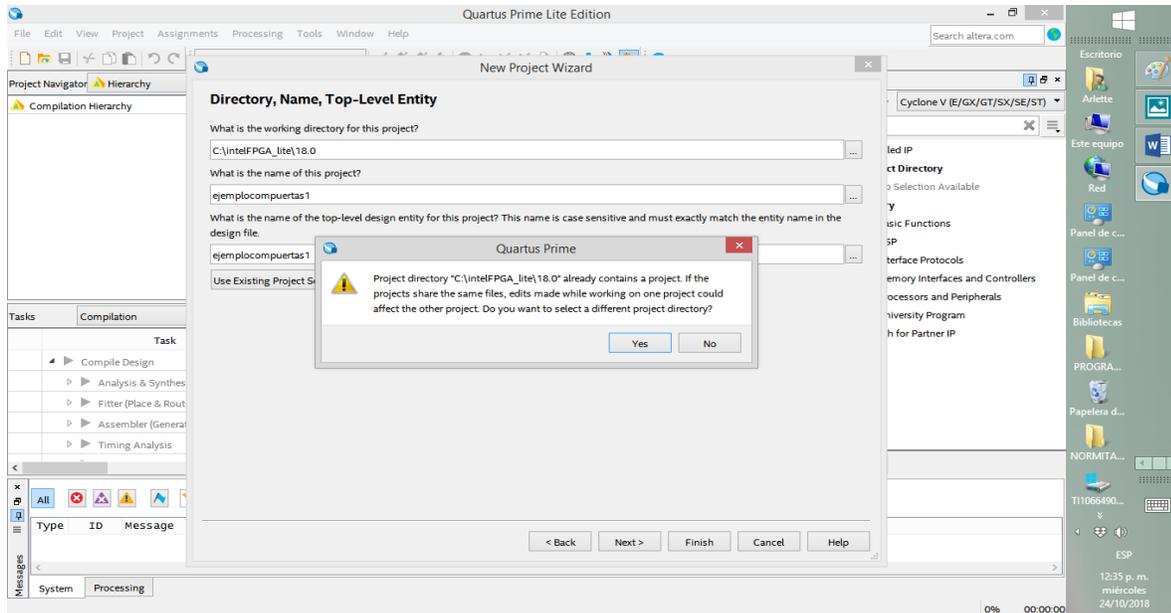


Figura 128 Presentación del paso 8

PASO 9

En el siguiente recuadro la opción empty project viene dada por default así que solo selecciona la opción Next

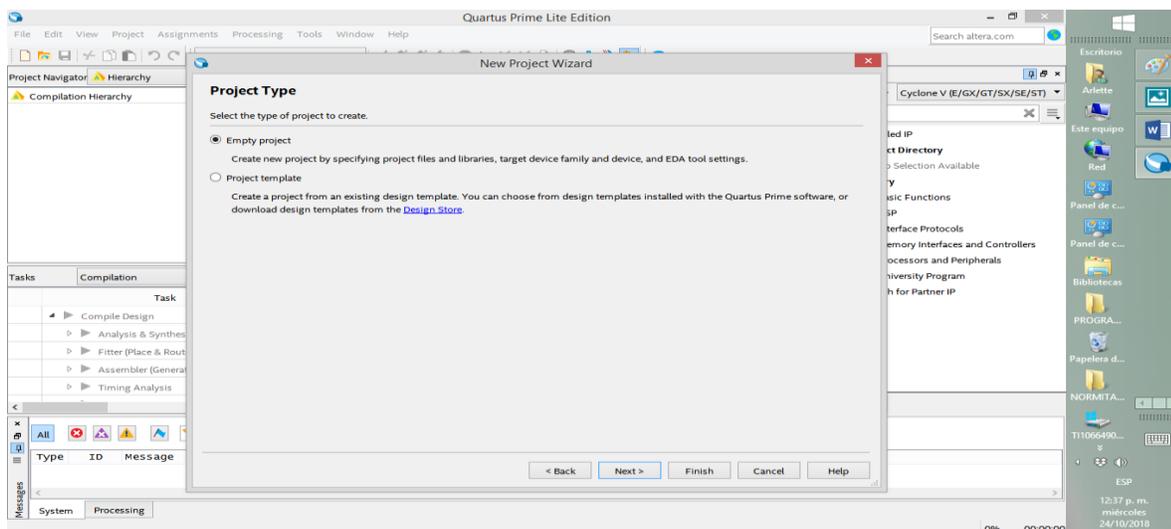


Figura 129 Presentación del paso 9

PASO 10

En el siguiente cuadro selecciona Next

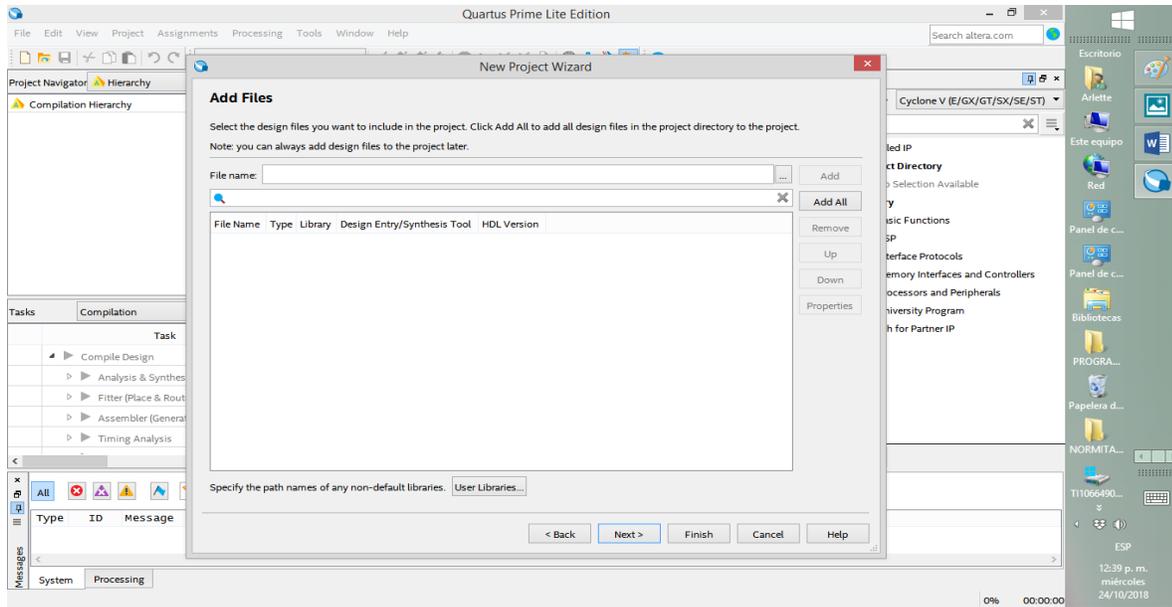


Figura 130 Presentación del paso 10

PASO 11

En el siguiente recuadro aparecen los siguientes datos iniciales aquí solo cambiaremos dos datos.

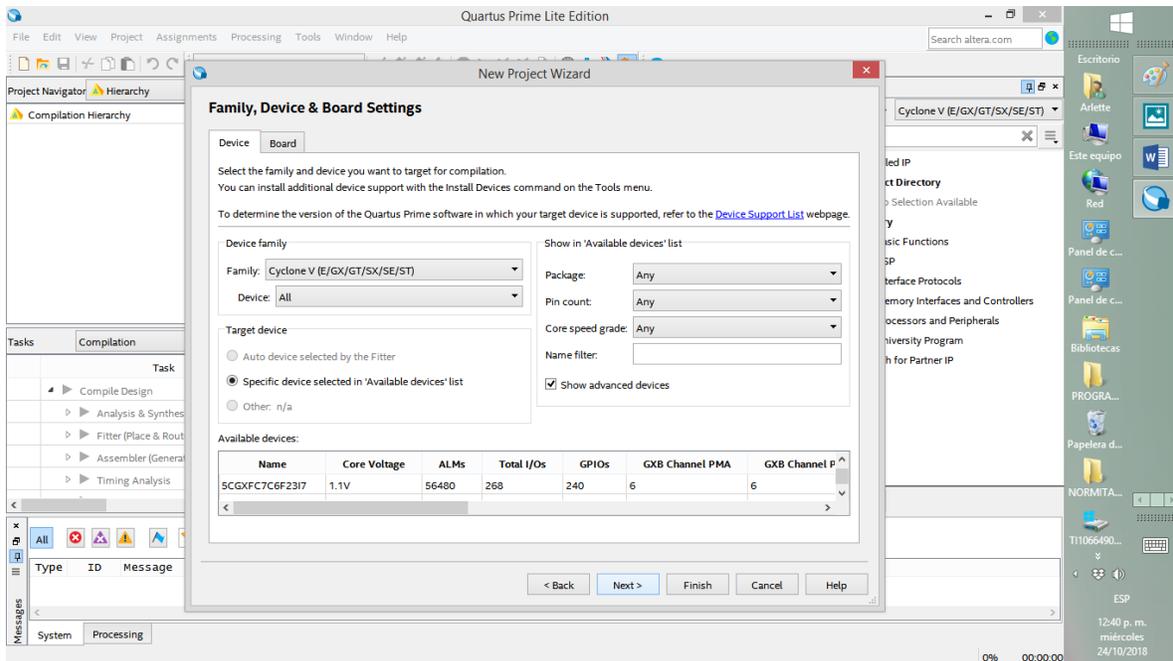


Figura 131 Presentación del paso 11

En familia selecciona la opción Max II

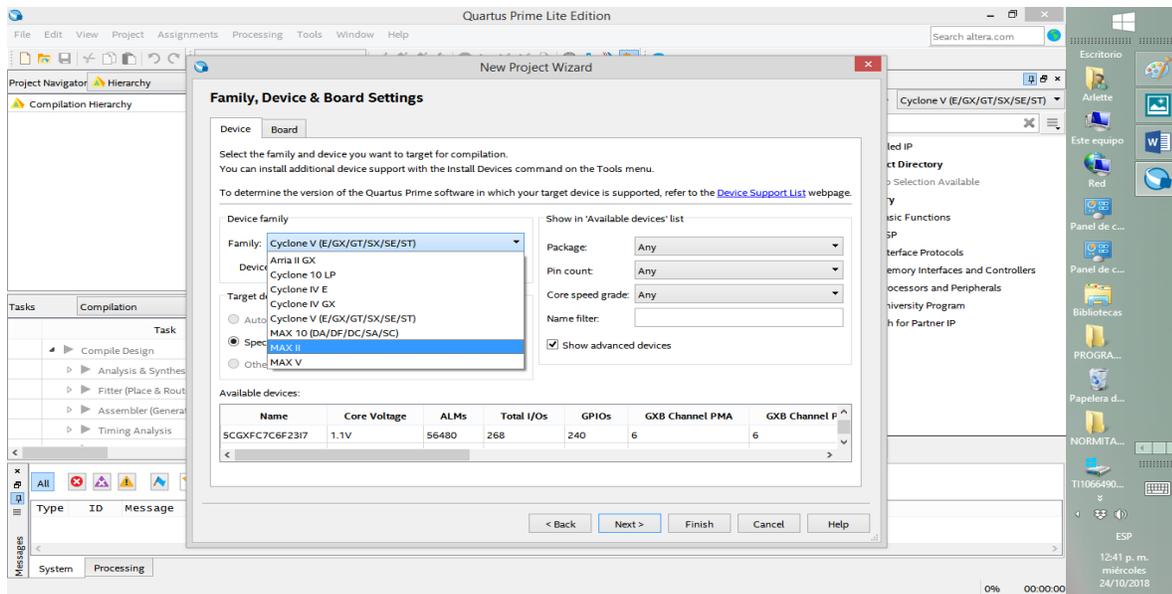


Figura 132 Selección de Max II

En la opción Available device selecciona EMP240TIC100C5

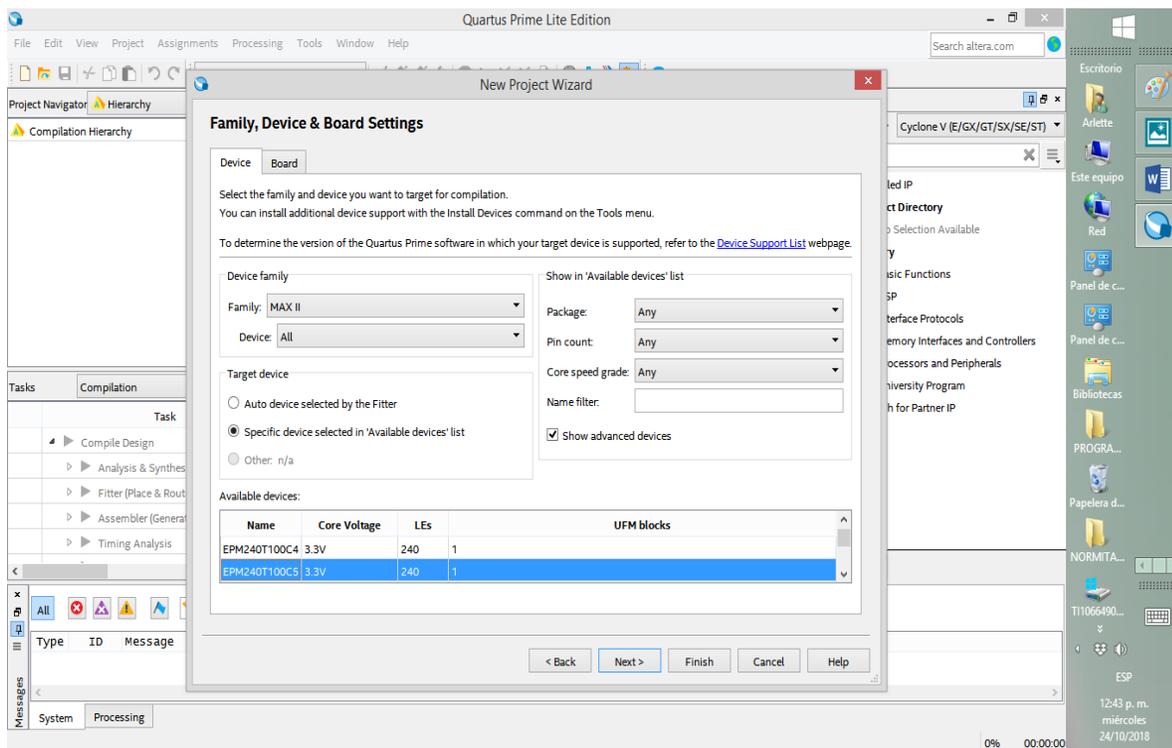


Figura 133 Selección de la tarjeta

Finalmente, los dos recuadros quedarían así

Device family

Family: MAX II

Device: All

Target device

Auto device selected by the Fitter

Specific device selected in 'Available devices' list

Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Core speed grade: Any

Name filter:

Show advanced devices

Available devices:

Name	Core Voltage	LEs	UFM blocks
EPM240T100C4	3.3V	240	1
EPM240T100C5	3.3V	240	1

Figura 134 Cuadros después del proceso

PASO 11

Una vez completados los recuadros selecciona la opción next

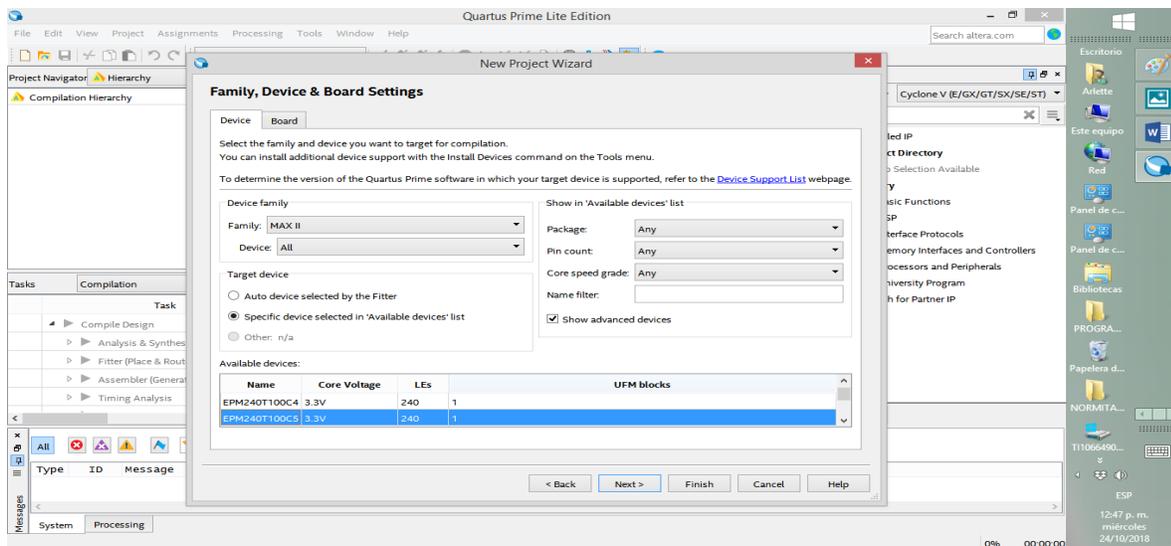


Figura 135 Paso 11

PASO 12

En el siguiente recuadro no es necesario cambiar nada, así que solo selecciona la opción Next

EDA Tool Settings

Specify the other EDA tools used with the Quartus Prime software to develop your project.

EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synth...	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	<None>	<None>	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

< Back Next > Finish Cancel Help

Figura 136 Paso 12

PASO 13

Enseguida aparecerá el siguiente recuadro solo selecciona opción Finish

Summary

When you click Finish, the project will be created with the following settings:

Project directory: C:\intelFPGA_lite\18.0
 Project name: ejemplocompuertas1
 Top-level design entity: ejemplocompuertas1
 Number of files added: 0
 Number of user libraries added: 0

Device assignments:

 Design template: n/a
 Family name: MAX II
 Device: EPM240T100C5
 Board: n/a

EDA tools:

 Design entry/synthesis: <None> (<None>)
 Simulation: <None> (<None>)
 Timing analysis: 0

Operating conditions:

 VCCINT voltage: 3.3V
 Junction temperature range: 0-85 °C

< Back Next > Finish Cancel Help

Figura 137 Paso 13

PASO 14

A continuación, aparece el siguiente cuadro que nos indica esperar un momento después aparecerá el siguiente cuadro

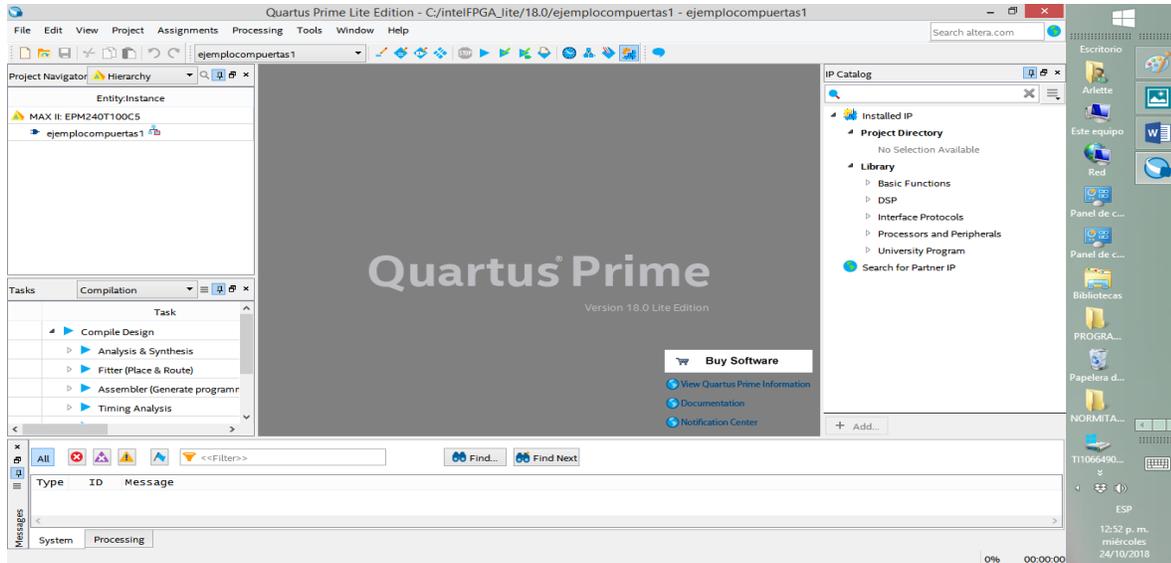


Figura 138 Paso 14

PASO 15

Seleccionamos la opción New (la que tiene una hoja en

blanco)



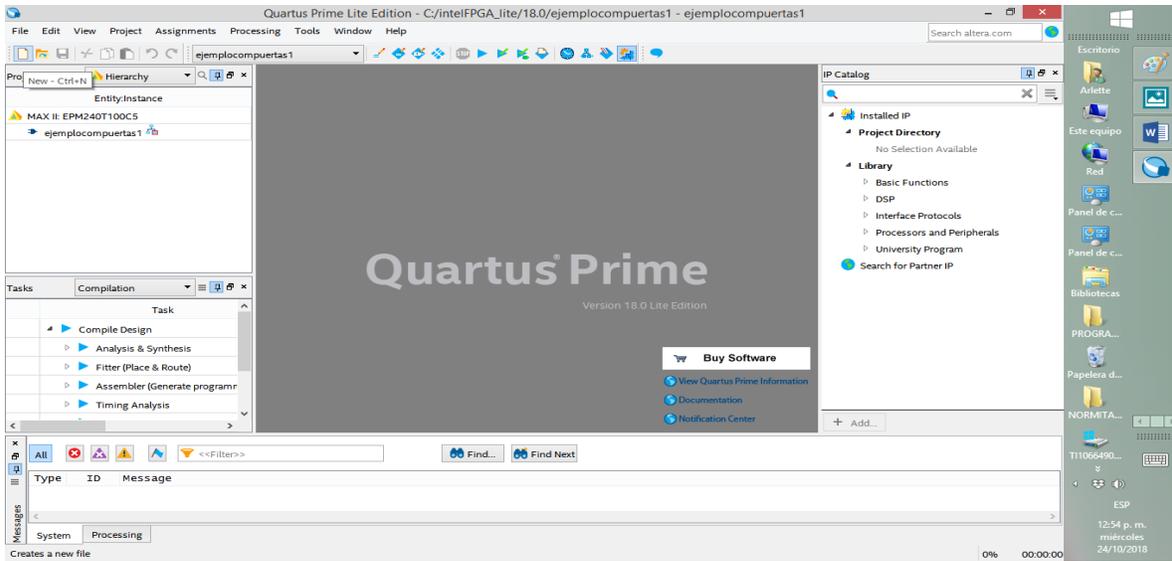


Figura 139 Representación del paso 15

PASO 16

Después seleccionamos la opción

Block Diagram/Schematic File

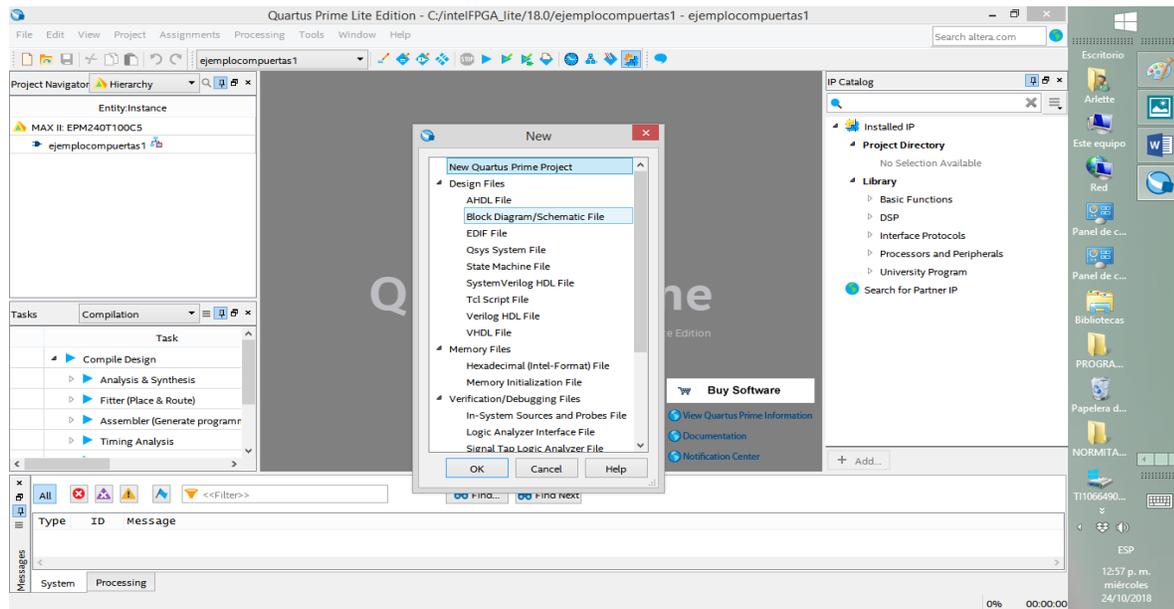


Figura 140 Paso 16

PASO 17

Se selecciona la opción Ok y aparece la siguiente pantalla

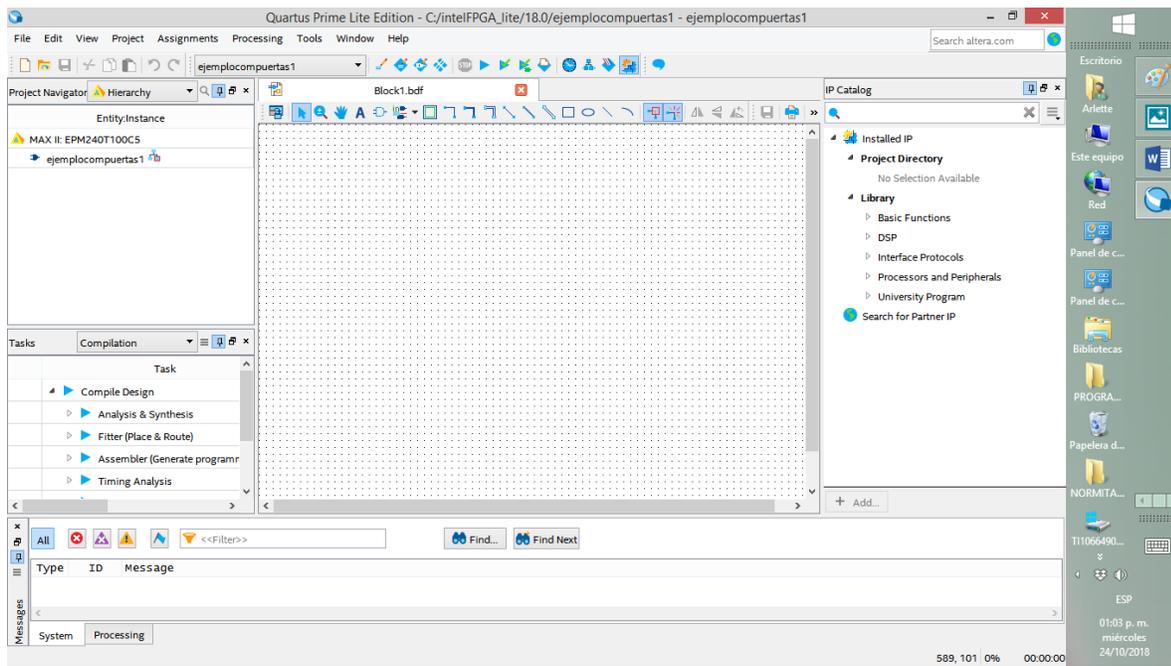


Figura 141 Representación paso 17

En esta pantalla es donde colocaremos los componentes, los uniremos y posteriormente los simularemos

PASO 18

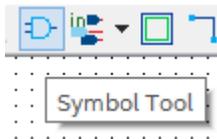


Figura 142 Paso 18

Los componentes se colocarán con la opción Symbol Tool.

PASO 19

Una vez seleccionada la opción Symbol Tool se despliega un recuadro en el cual podremos seleccionar las compuertas y especificaremos también el número de entradas compuertas y especificaremos también el número de entradas

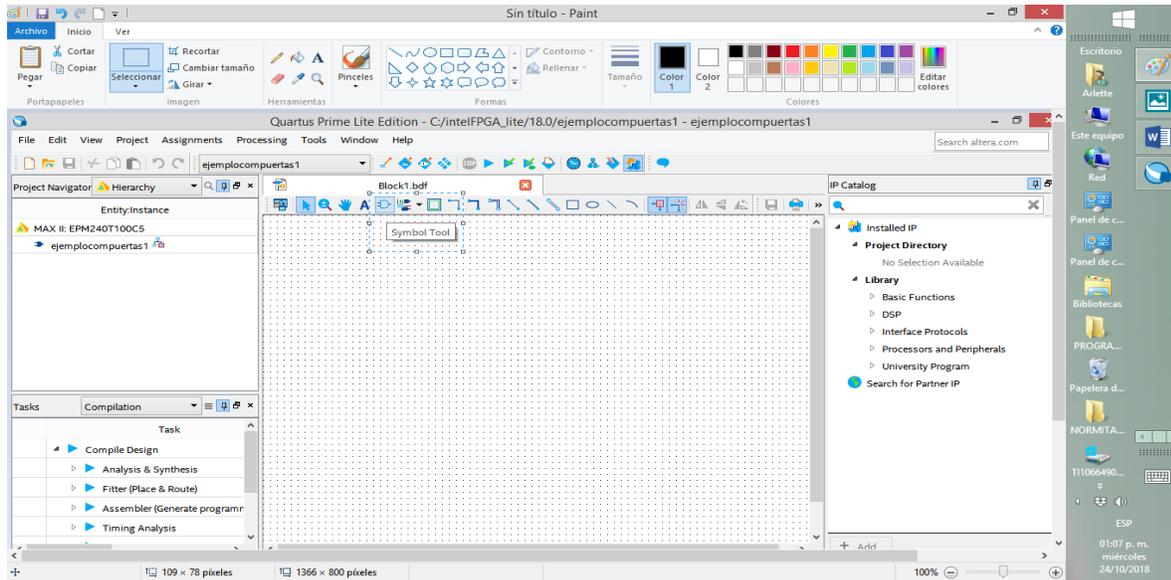


Figura 143 Selección de componentes

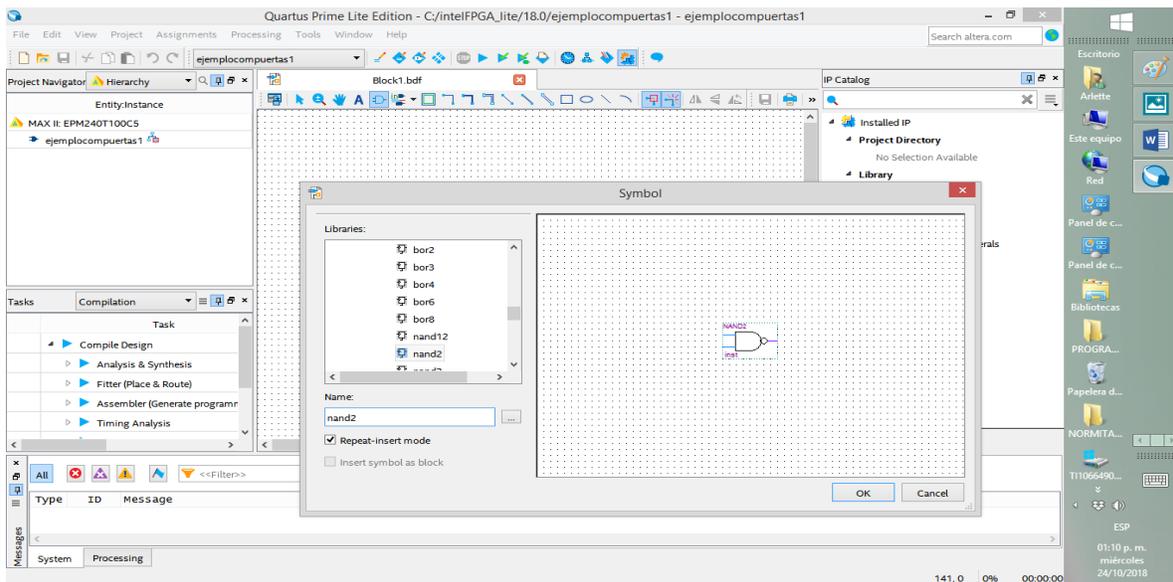


Figura 144 Ejemplo de selección de componentes

Ejemplo

nand2

Indica una compuerta nand 2 de dos entradas

PASO 20

Colocando la siguiente compuerta

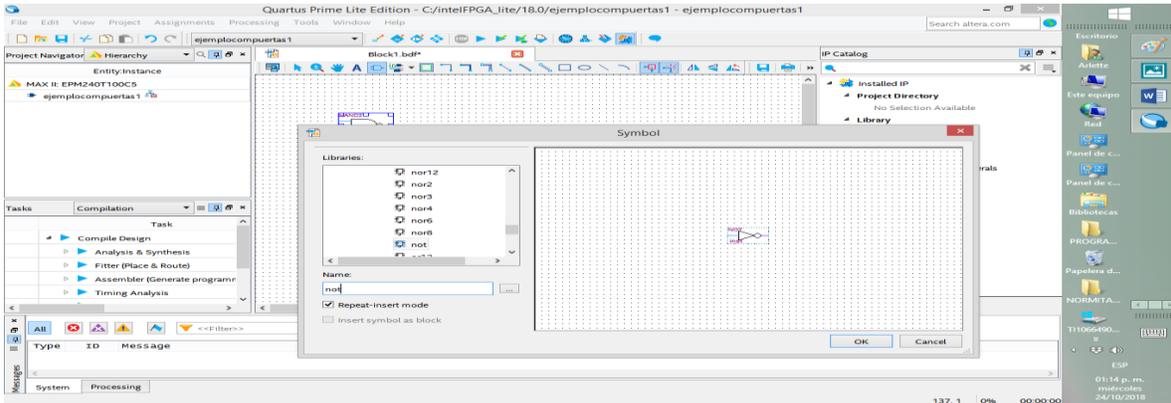


Figura 145 Paso 20

PASO 21

Se muestra la pantalla donde todos los componentes necesarios han sido colocados

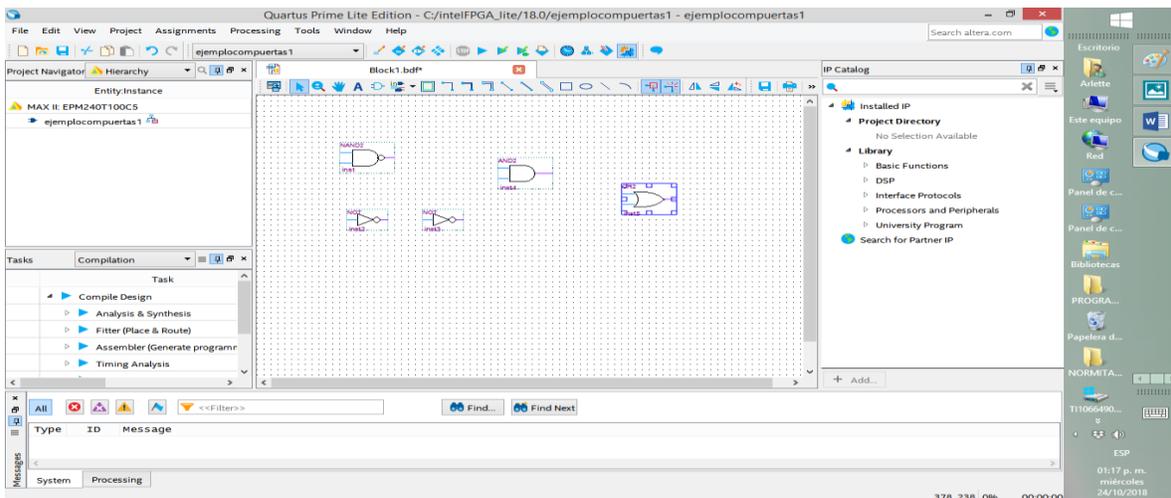


Figura 146 Paso 21

PASO 22

Se procede a la unión de los componentes

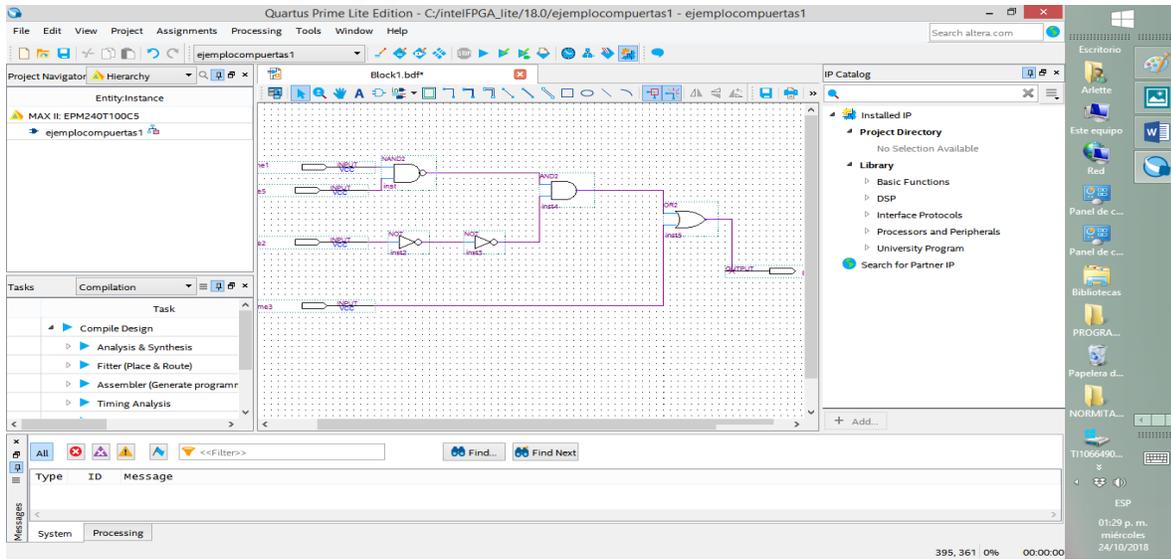


Figura 147 Paso 22

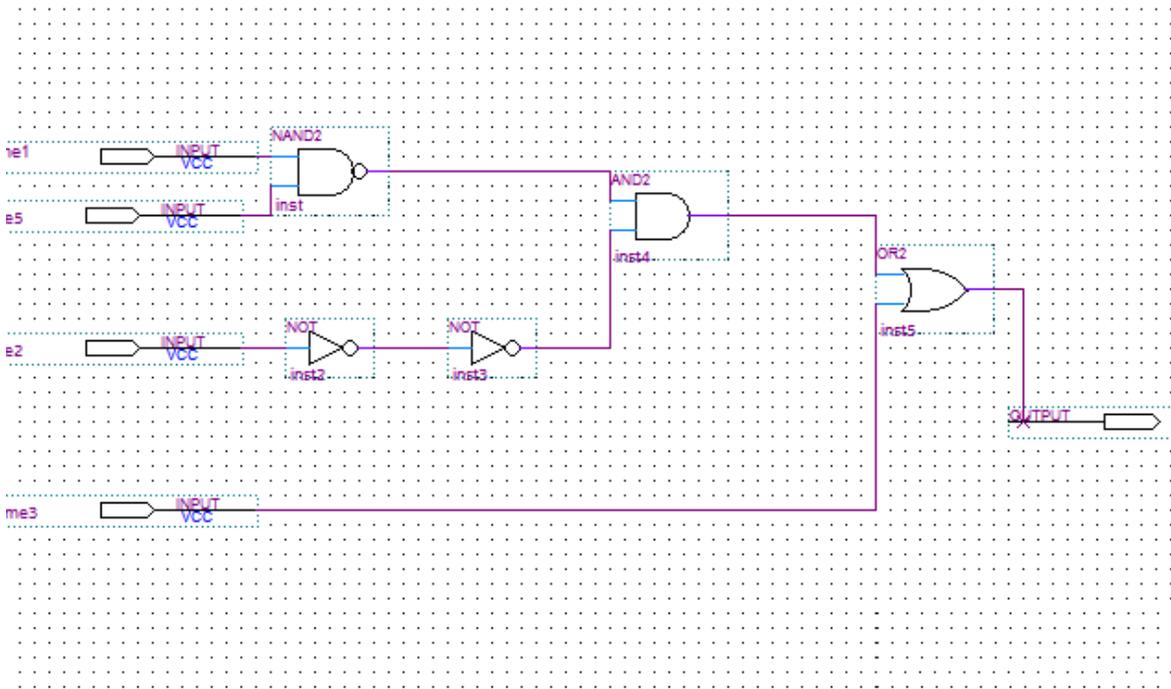


Figura 148 Paso 22 simplificado

Se muestran los resultados de la siguiente función para que los pueda verificar con los de la simulación que se presentará posteriormente.

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

Tabla 152. Resultados de función para simular como ejercicio

PASO 23

De la opción de Symbol Tools también se obtienen las entradas y las salidas como se muestra a continuación.

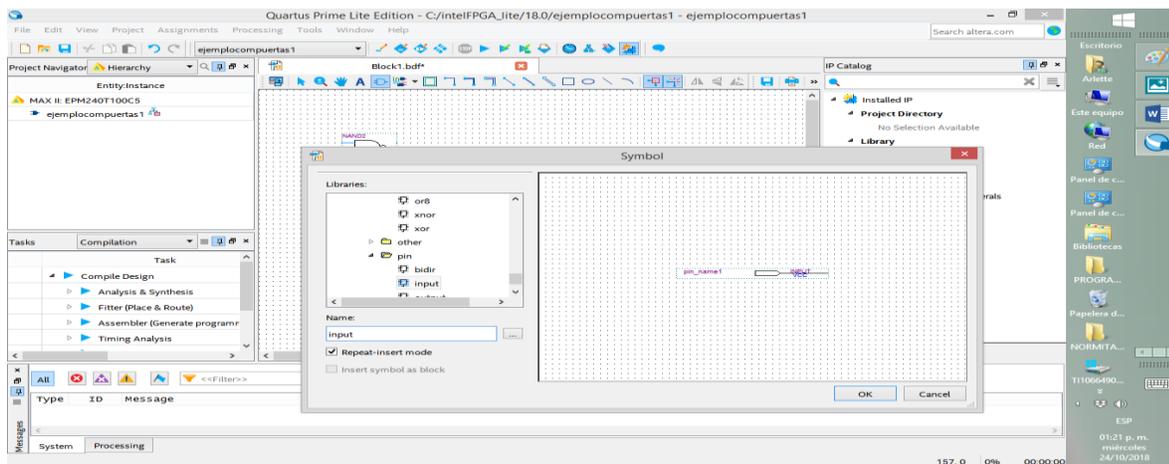


Figura 149 Paso 23

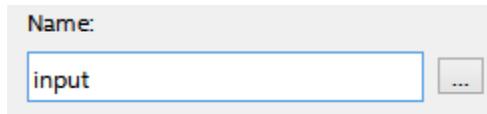


Figura 150 Imagen del simbolo para las entradas y salidas



Figura 151 Imagen del simbolo para las entradas y salidas

Y las salidas como

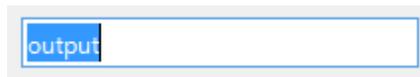


Figura 152 Imagen del simbolo para las entradas y las salidas

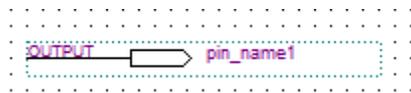


Figura 153 Imagen del simbolo para las entradas y salidas

Como cambiar el nombre a los pines de salida y entrada:

Colocarse sobre el pin que se desea cambiar nombre dar click derecho seleccionar propiedades

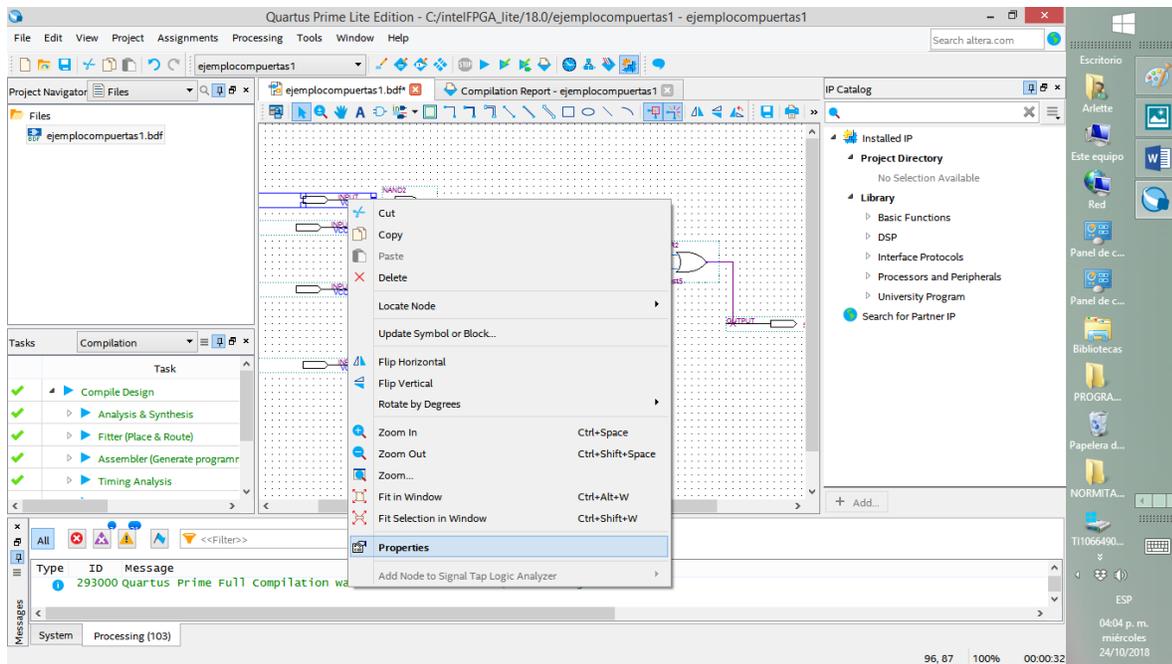


Figura 154 Como se cambia el nombre de los pines

En el recuadro que tiene la opción pin name colocar el nombre que se desea y dar ok

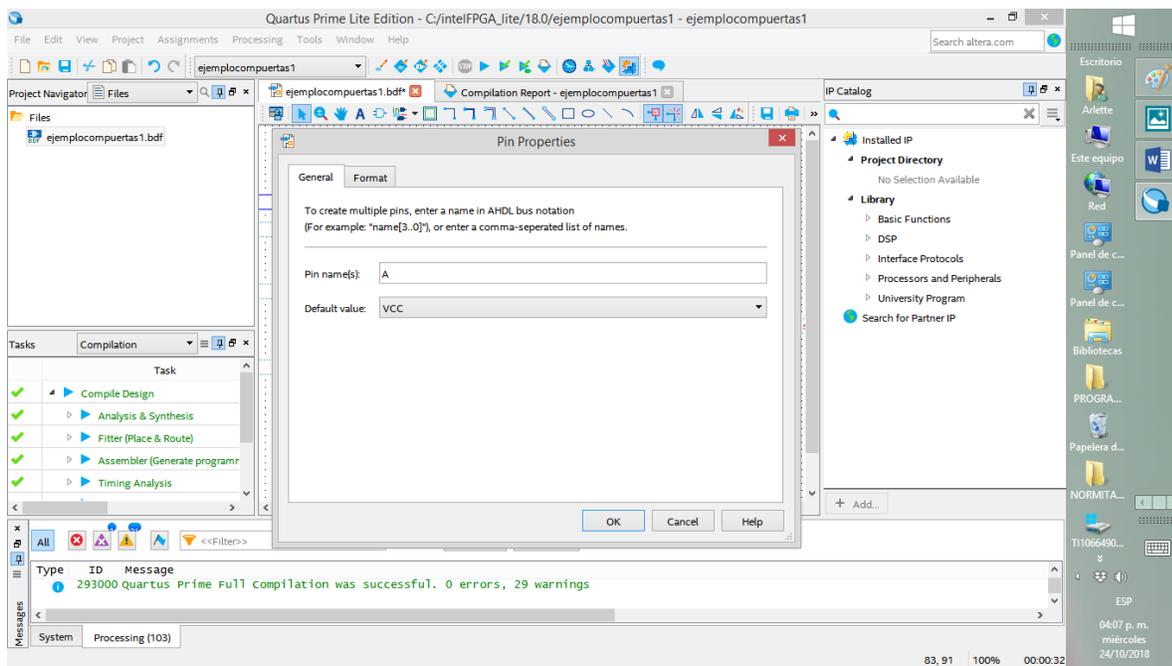


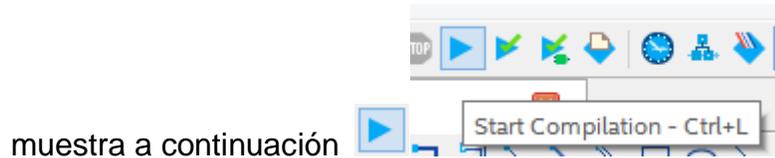
Figura 155 Cambio de nombre de pin

SIMULACIÓN

PASO 1

Una vez que realizaste tu diagrama procederemos a realizar la simulación.

Procedemos al proceso de compilación con el icono que se



muestra a continuación

Figura 156 Proceso de compilación

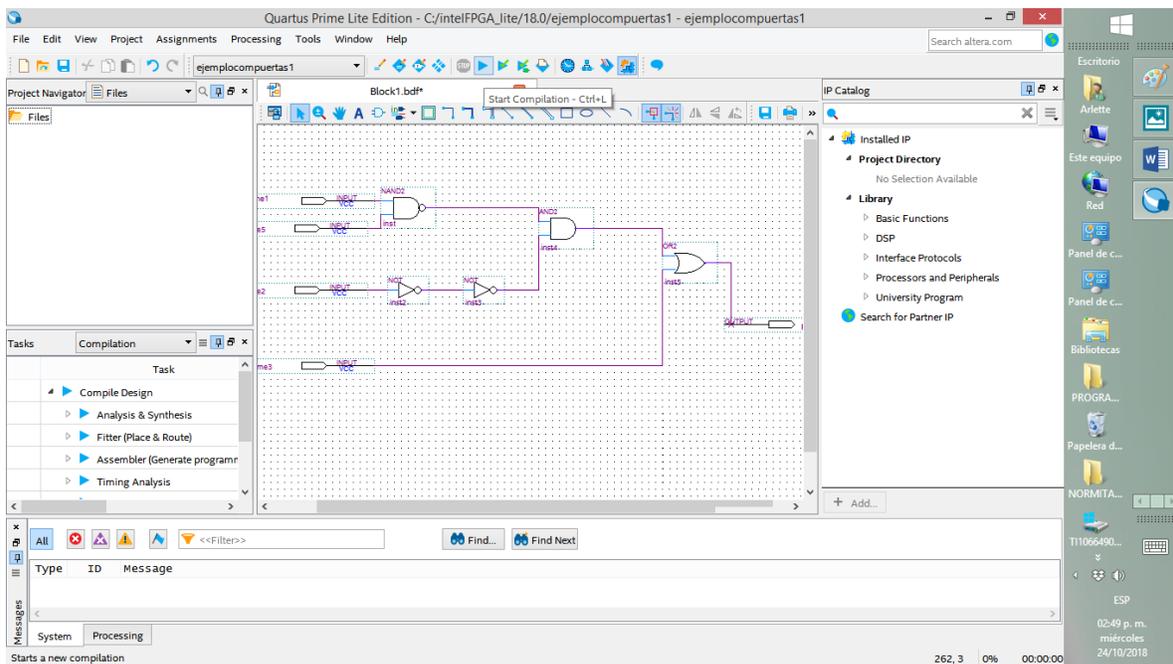


Figura 157 Circuito

Después del proceso de compilación aparece la siguiente pantalla

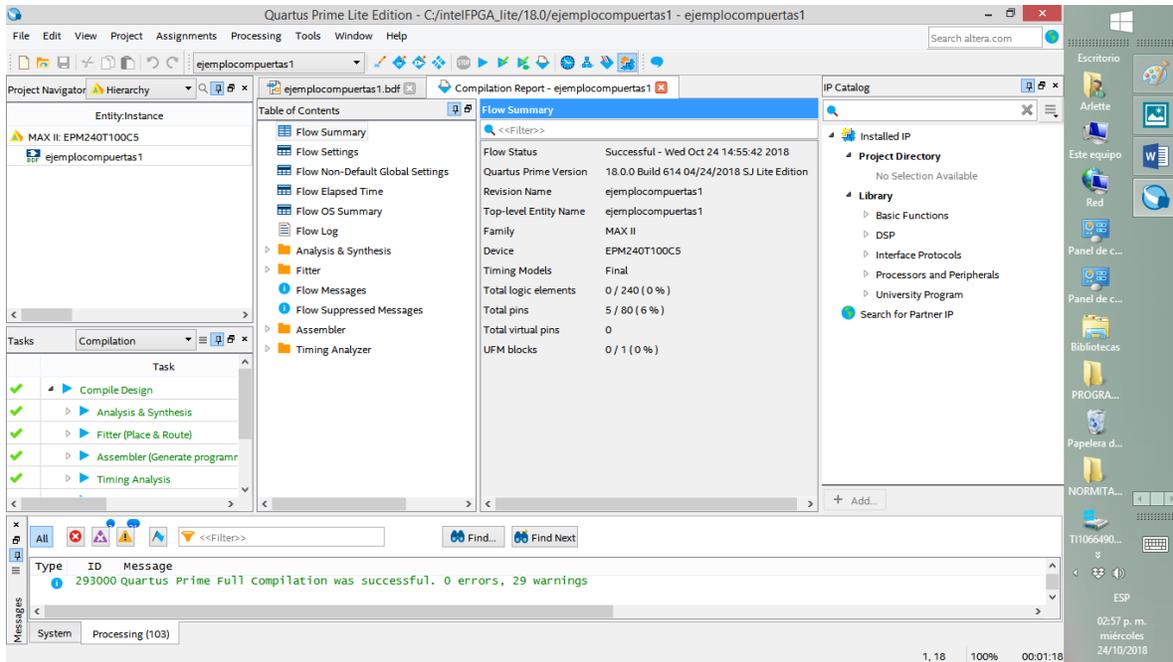


Figura 158 Después del proceso de compilación

Donde lo importante es la pantalla de abajo es que aparezca un letrero que diga cero errores.

Type	ID	Message
	293000	Quartus Prime Full compilation was successful. 0 errors, 29 warnings

PASO 2

Después de la compilación se debe elegir la jerarquía para que nuestro proyecto actual sea el que se simule

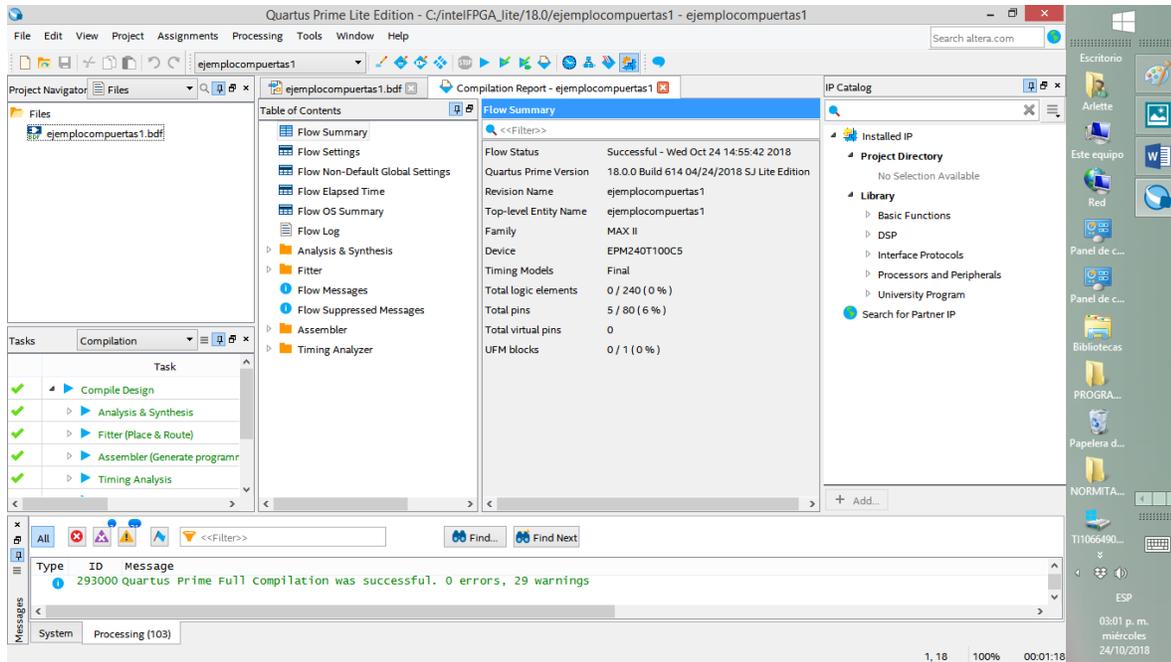


Figura 159 Imagen del paso 2 del proceso de simulación

Elegimos la opción

File como se muestra en la imagen

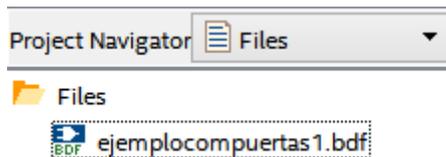


Figura 160 Proceso de simulación

Damos click derecho sobre la opción ejemplocompuestas1.bdf y elegimos la opción set Top-Level Entity

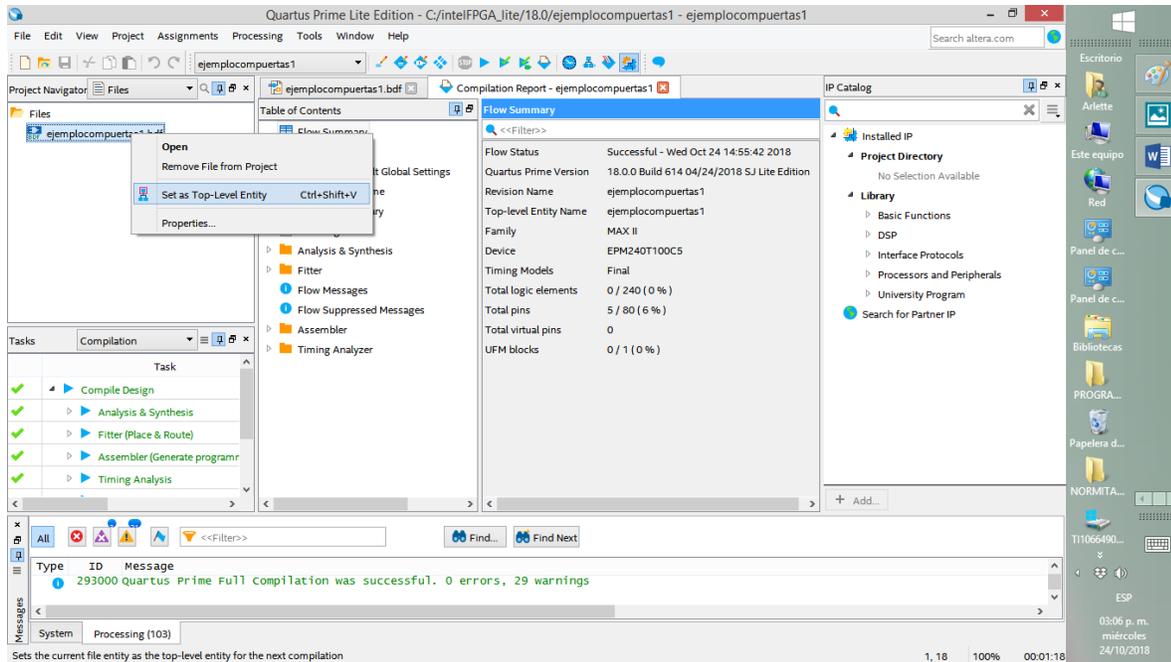
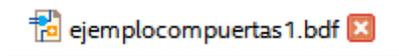


Figura 161 Proceso de simulación

PASO 3

Posteriormente volvemos a compilar, luego nos vamos al cuadro de dialogo que se muestra a continuación.



Una vez que estamos en esa opción se escoge la opción New



Y aparecerá un cuadro de dialogo como el siguiente

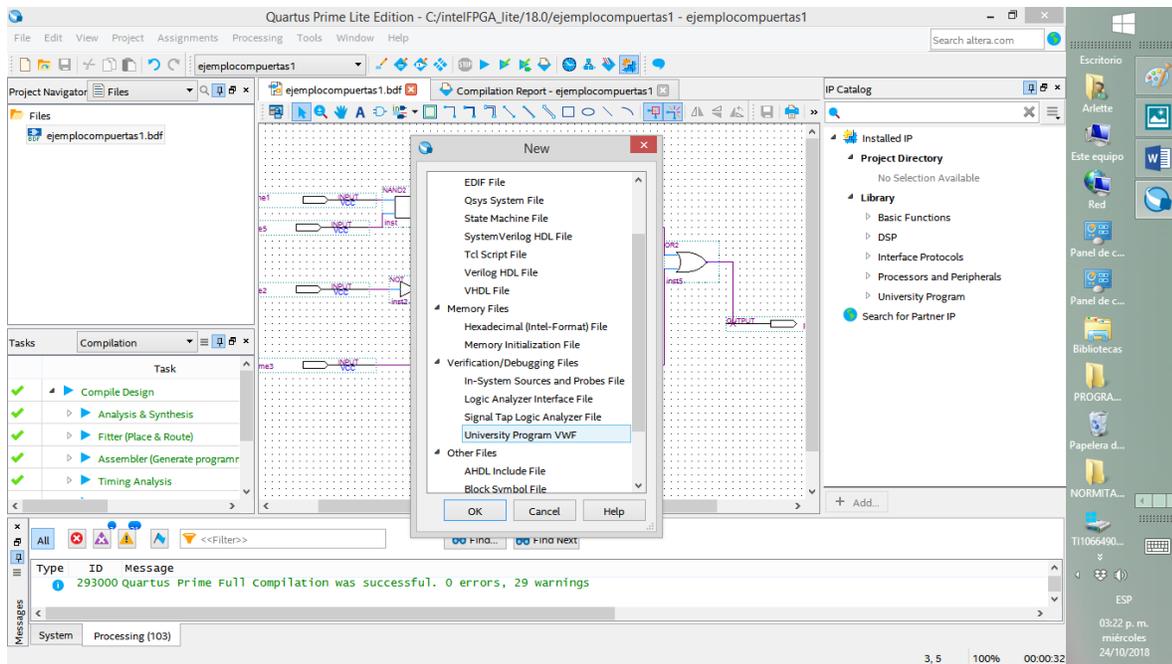


Figura 162 Proceso de simulación

Seleccione la opción

University Program VWF

Y seleccione la opción Ok

Se desplegará una pantalla como la siguiente

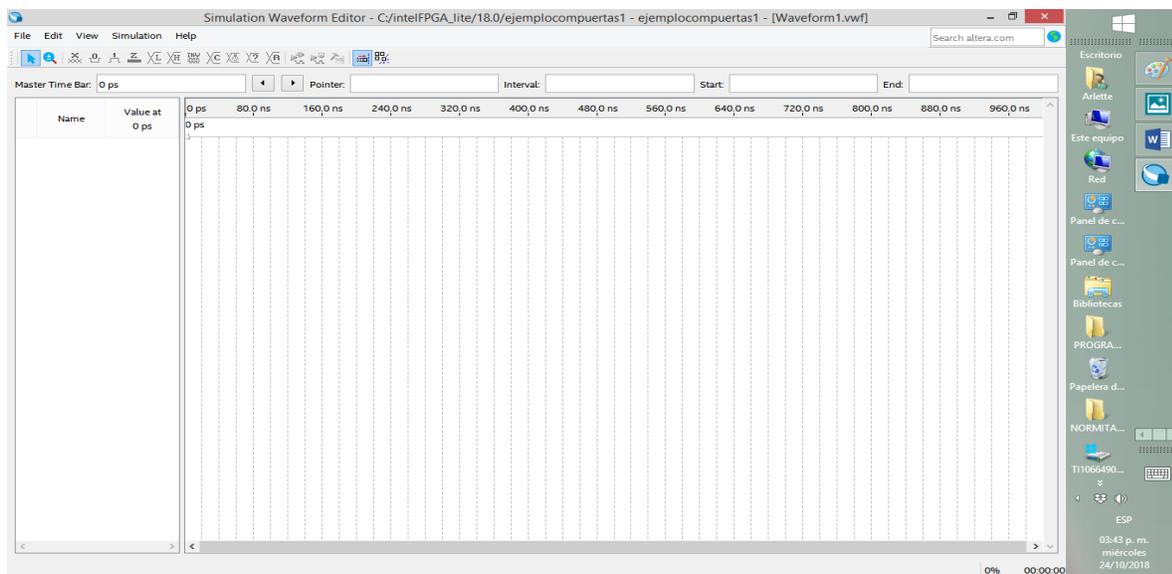


Figura 163 Proceso de simulación

PASO 3

Se selecciona la opción Edit y se va a la opción Set End Time

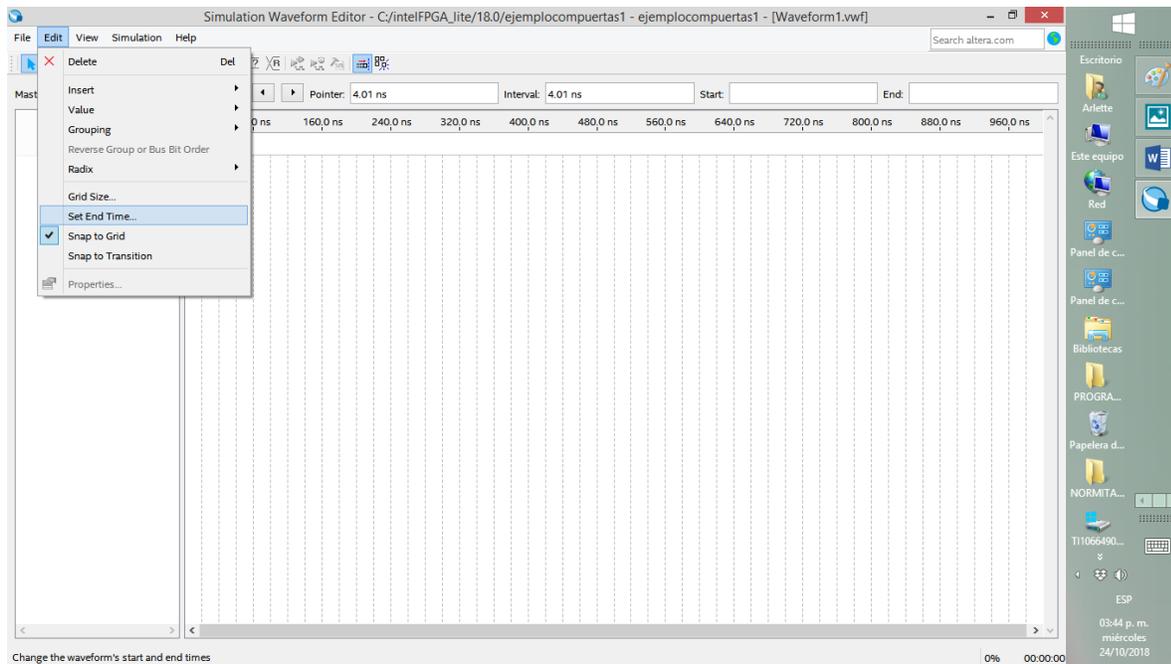


Figura 164 Proceso de simulación

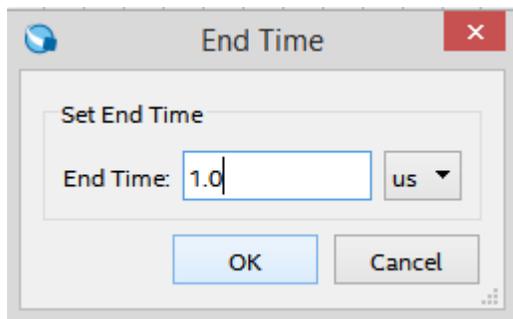


Figura 165 Proceso de simulación para el paso 4

PASO 4

Se va a la opción Edit y se escoge un periodo de 20 micro segundos

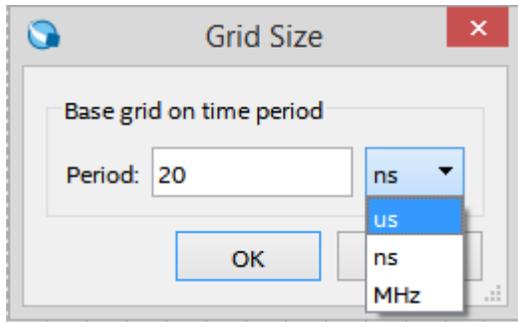


Figura 166 Proceso de simulación

PASO 5

Después de realizados los procedimientos anteriores aparece la siguiente pantalla.

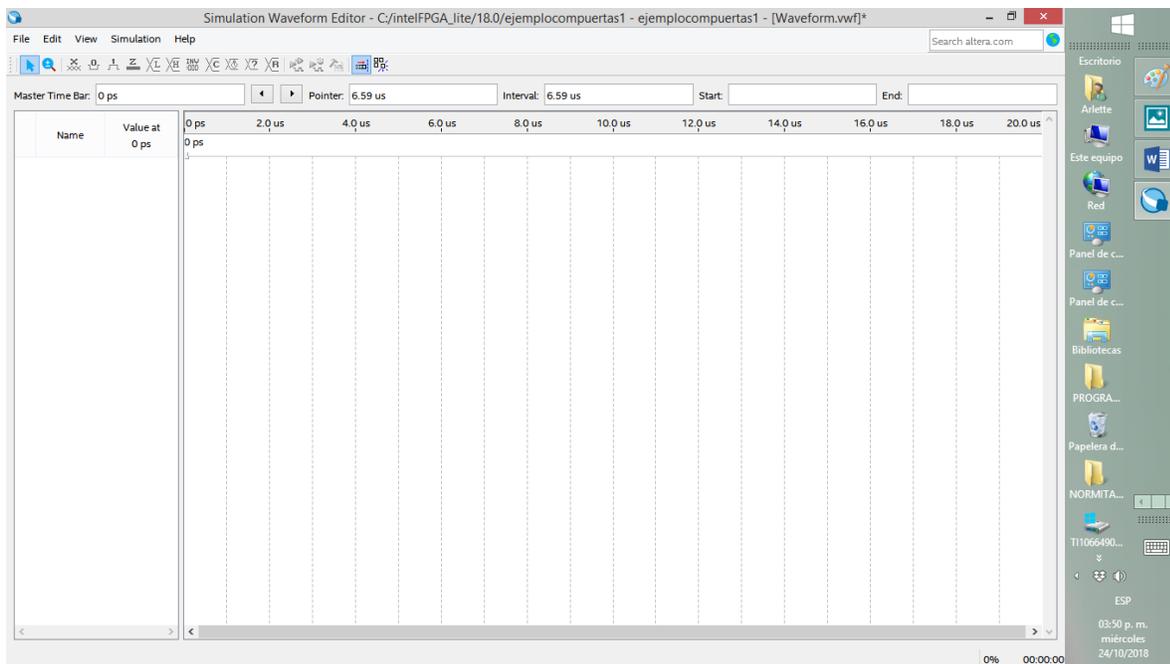


Figura 167 Proceso de simulación paso 5

PASO 6

Colocarse en el cuadro name value 0 y con el botón derecho llevar a cabo los pasos siguientes:

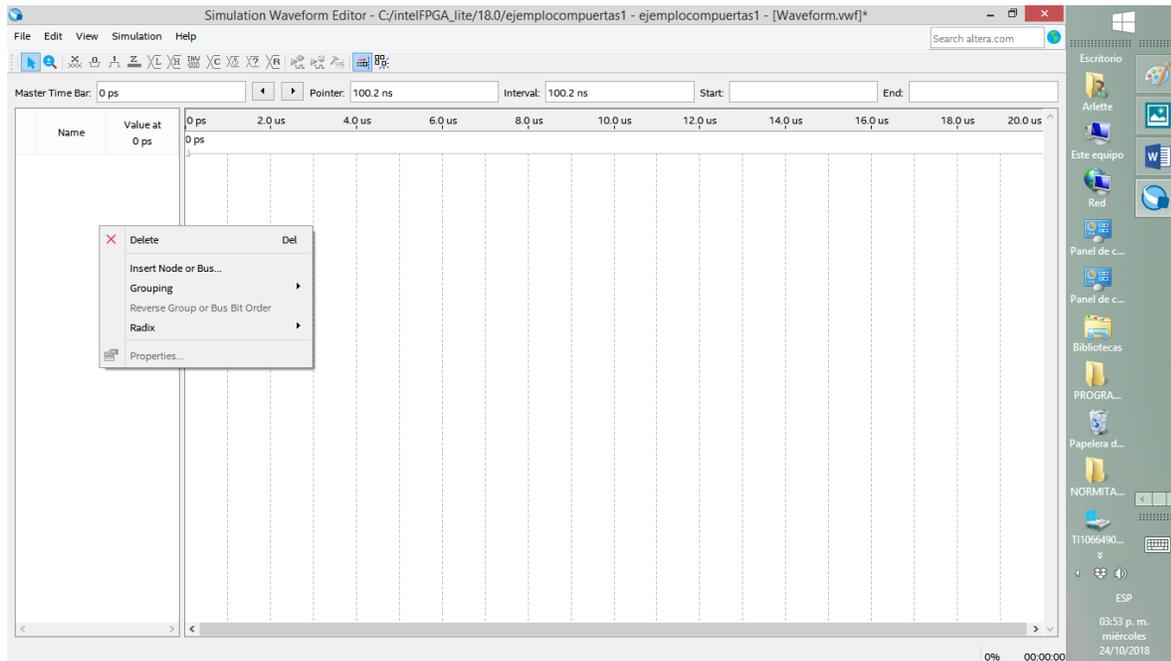


Figura 168 Proceso de simulación paso 6

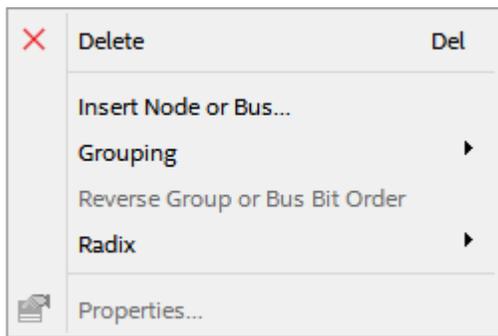


Figura 169 Proceso de simulación

PASO 7

Dar click derecho en Insert Node or Bus

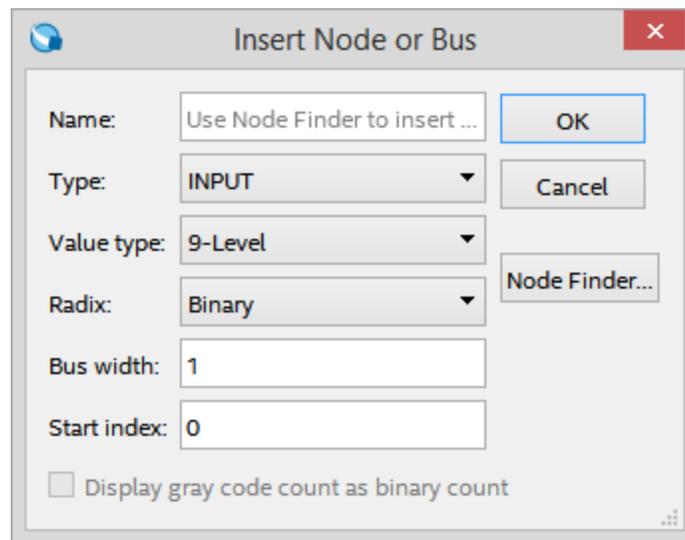


Figura 170 Proceso de simulación del paso 7

PASO 8

Dar click en Node Finder

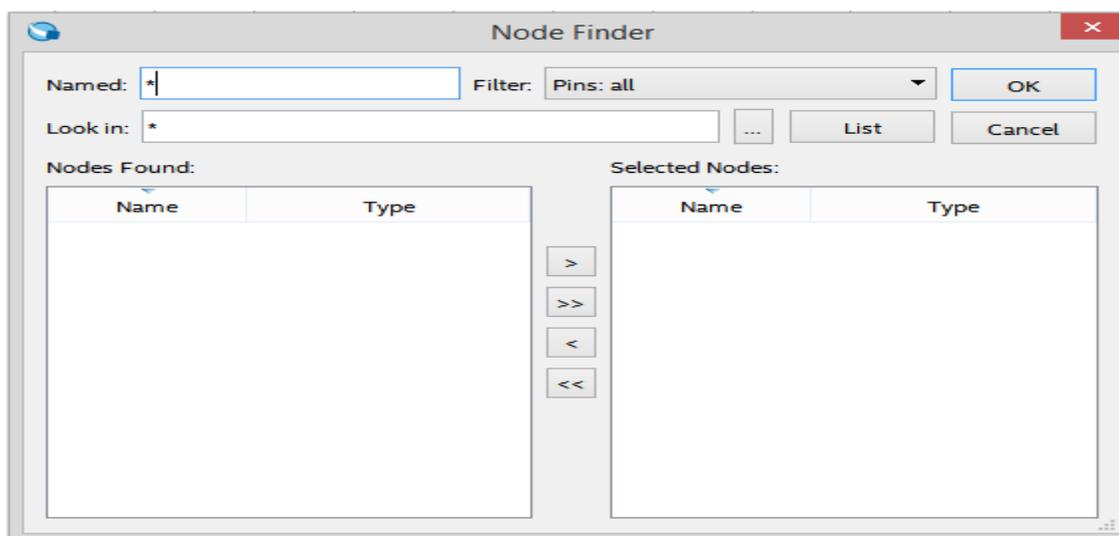


Figura 171 Proceso de simulación

Seleccionar List y aparecerá en forma de lista los elementos del proyecto es decir las entradas, las salidas y las compuertas.

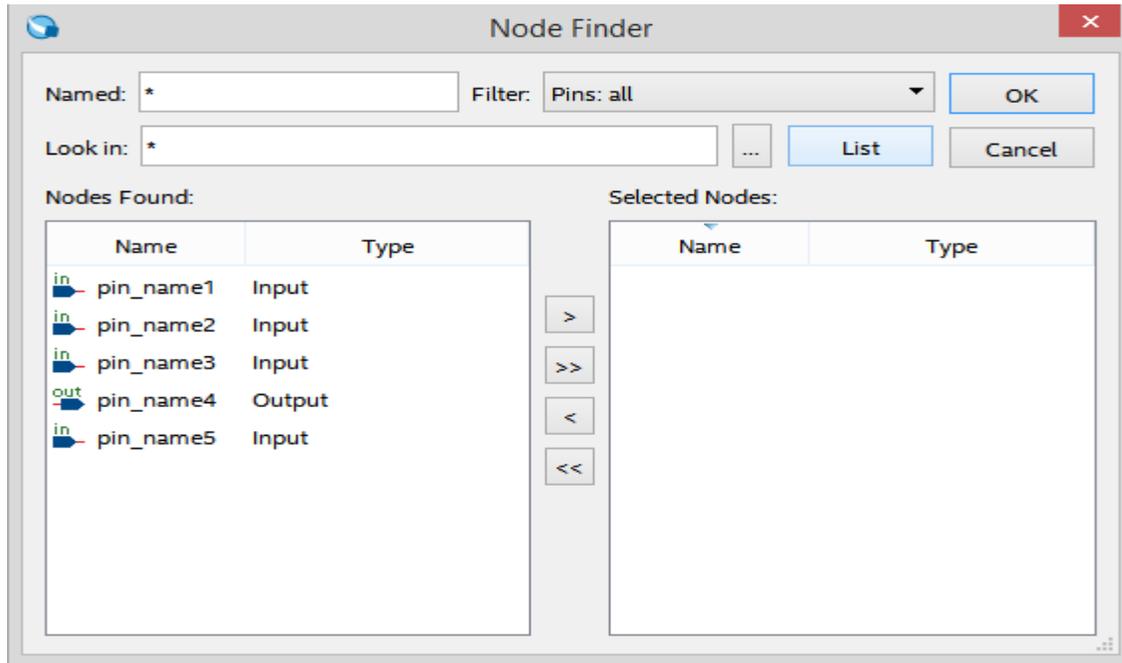


Figura 172 Proceso de simulación

Después seleccionar con el icono y aparecerá la pantalla

siguiente 

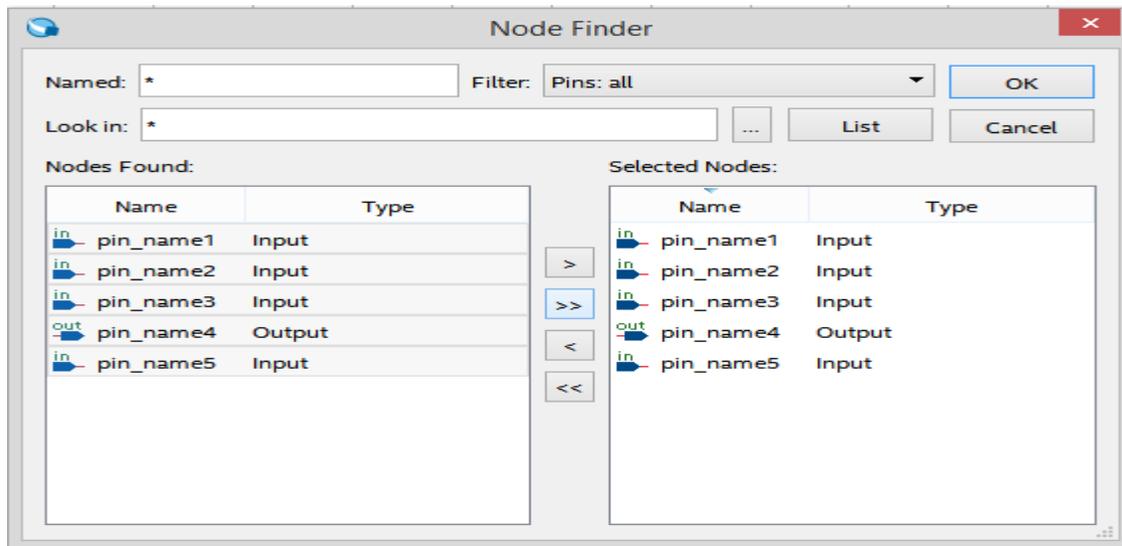


Figura 173 Proceso de simulación

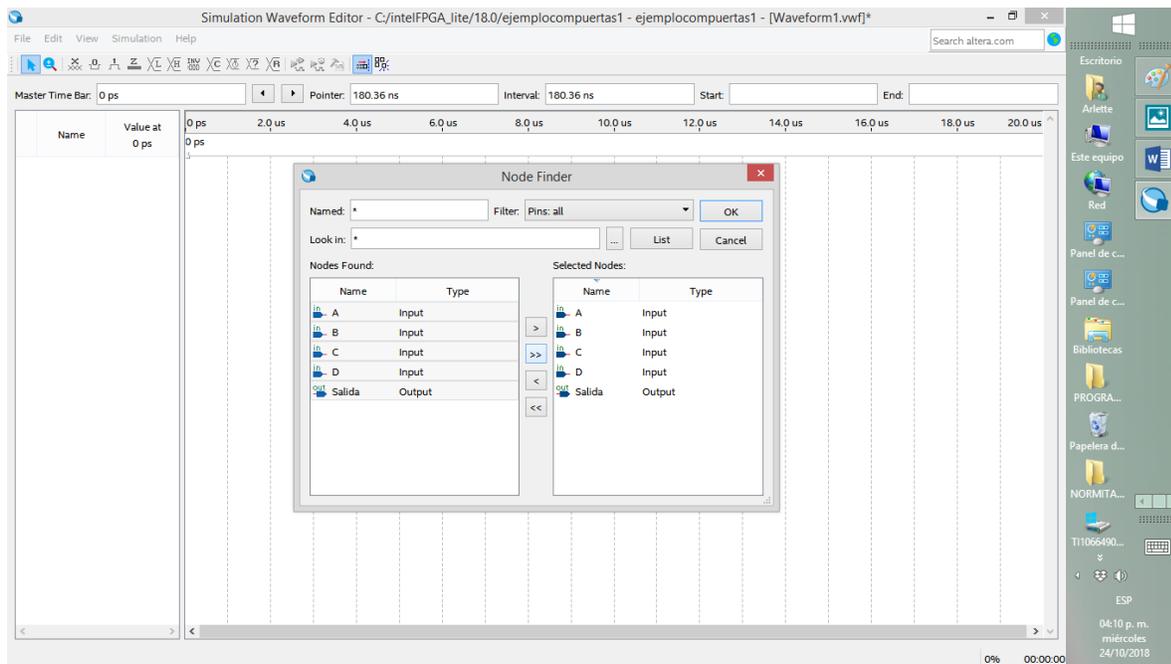


Figura 174 Proceso de simulación

Seleccionamos la opción ok

En caso de que se haya dado nombre a las variables de entrada y de salida estas aparecerán en el recuadro

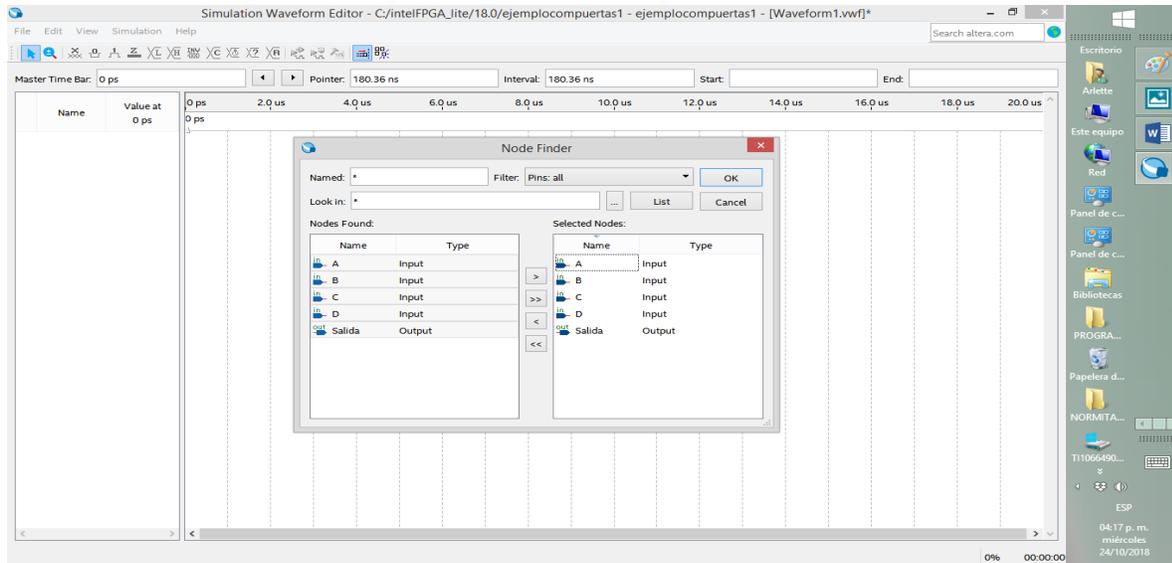


Figura 175 Proceso de simulación

Damos ok al recuadro aparecerá el siguiente recuadro y le damos ok

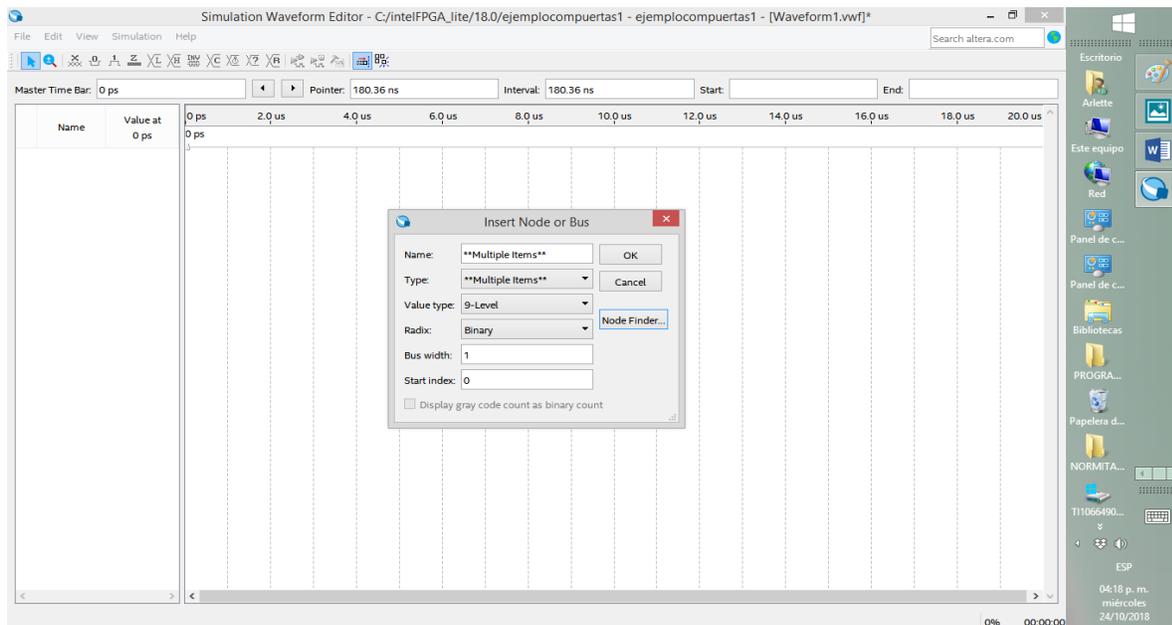


Figura 176 Proceso de simulación

Y se muestra el siguiente recuadro

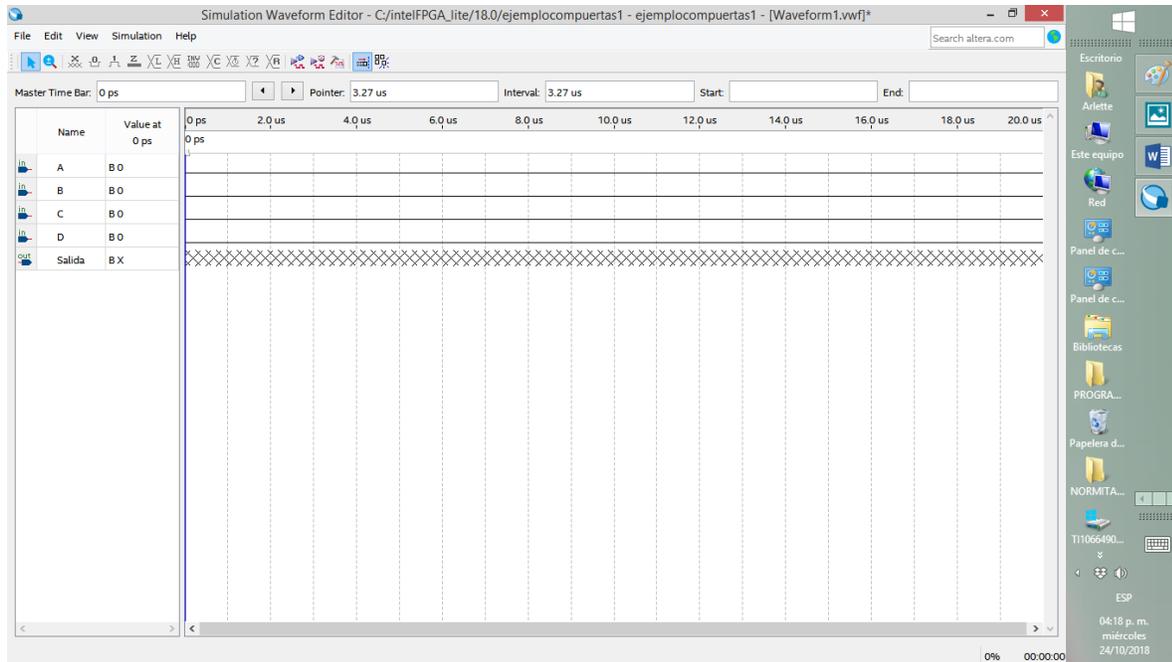


Figura 177 Proceso de simulación

El símbolo en salida es normal dado que el programa no ha simulado la función.

PASO 7

Se seleccionará una variable como se muestra a continuación

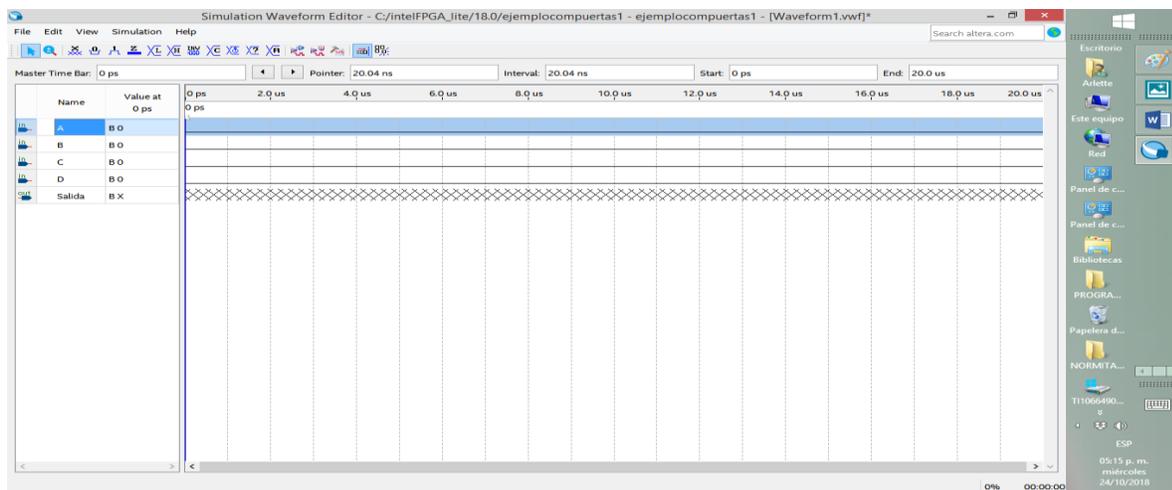


Figura 178 Proceso de simulación

Posteriormente seleccionamos la opción **overwrite clock**  y en recuadro que aparece a continuación vamos a la opción **periodo** y damos un valor en este caso **2 us**

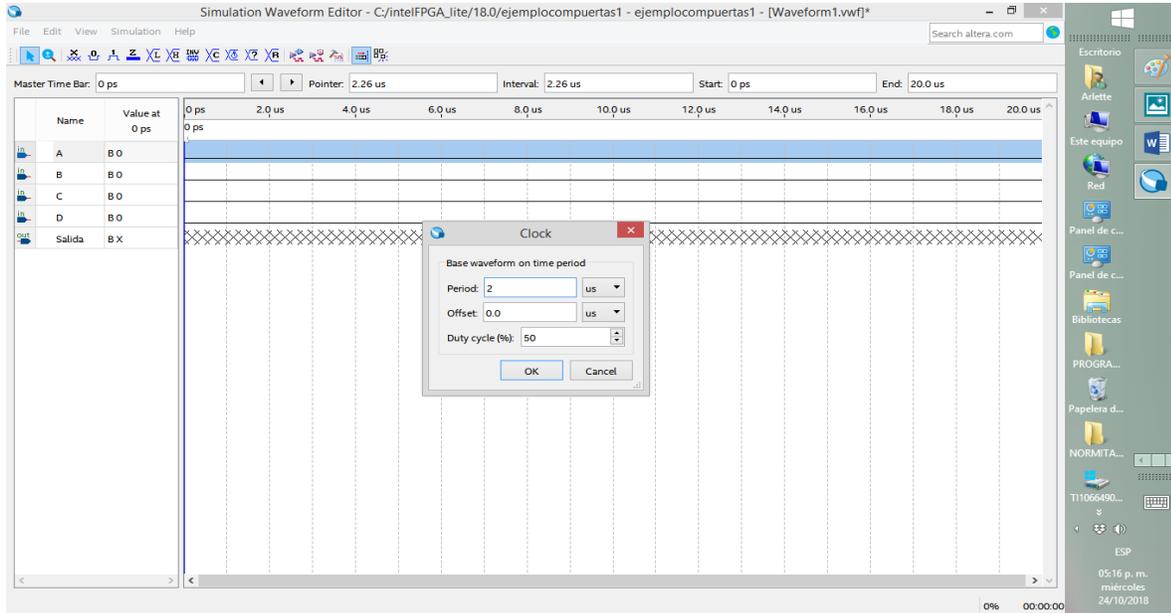


Figura 179 Proceso de simulación

Seleccionamos la opción **Ok** y se desplegara la siguiente pantalla

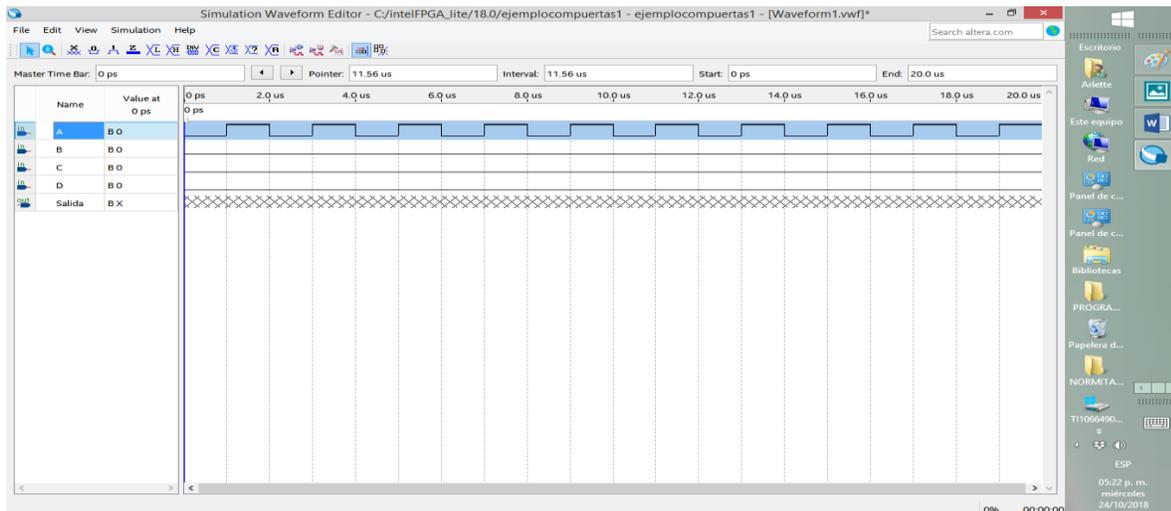


Figura 180 Proceso de simulación

Escogemos un valor distinto de periodo para cada una de las variables faltantes, se muestran los resultados a continuación.

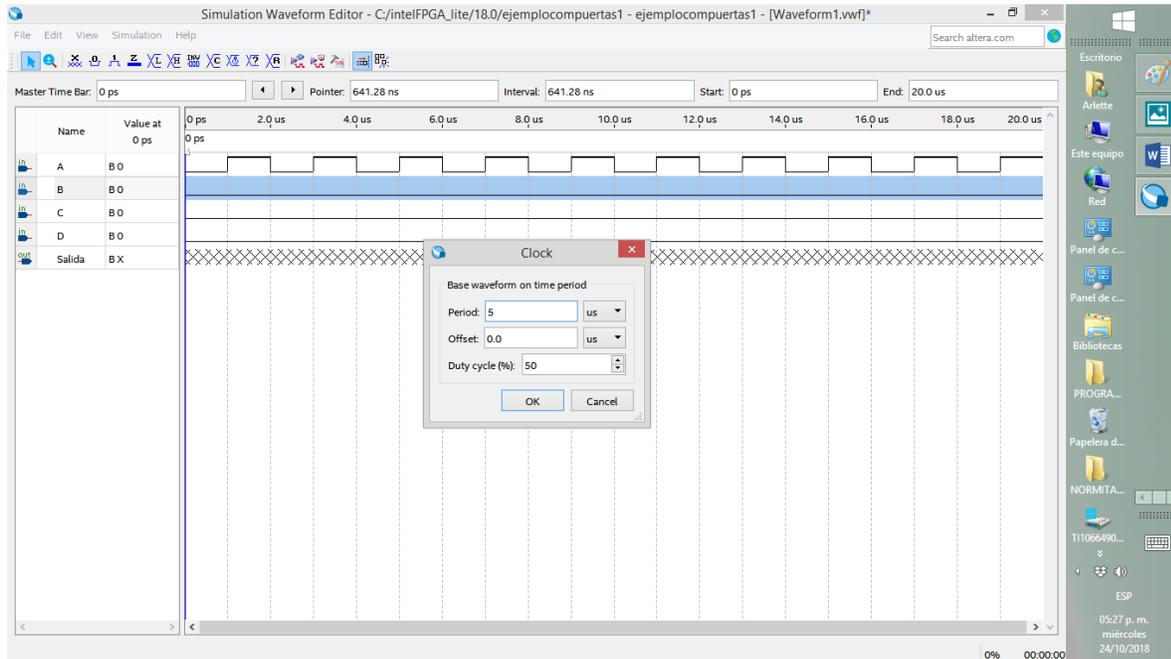


Figura 181 Proceso de simulación

Resultado de aplicar el periodo de 5us

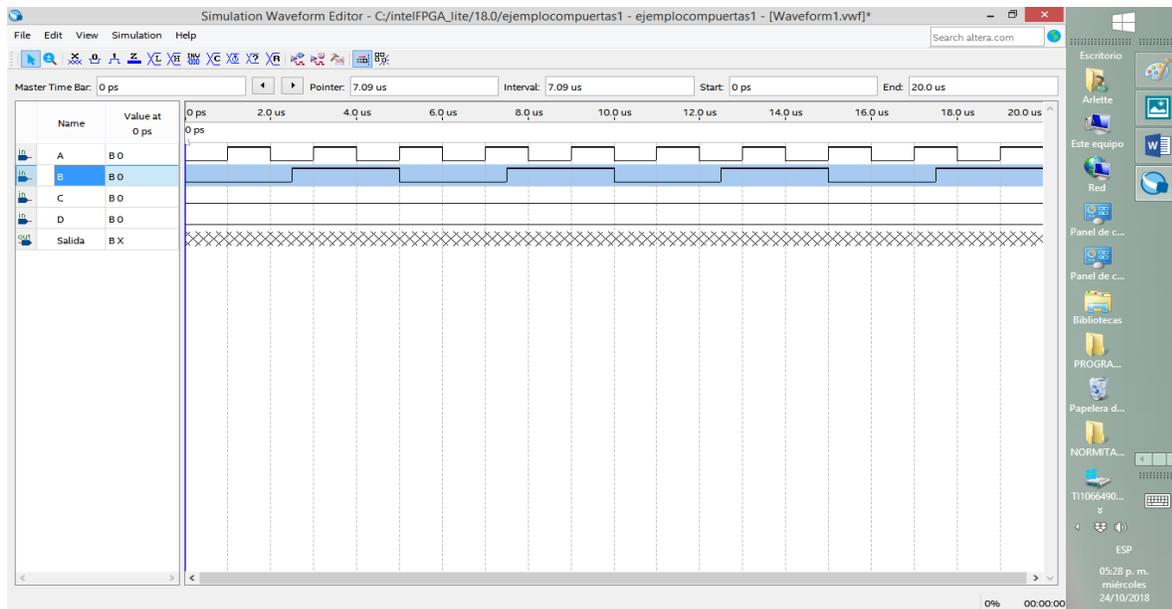


Figura 182 Proceso de simulación

Para la variable C
Aplicando un periodo de 7us

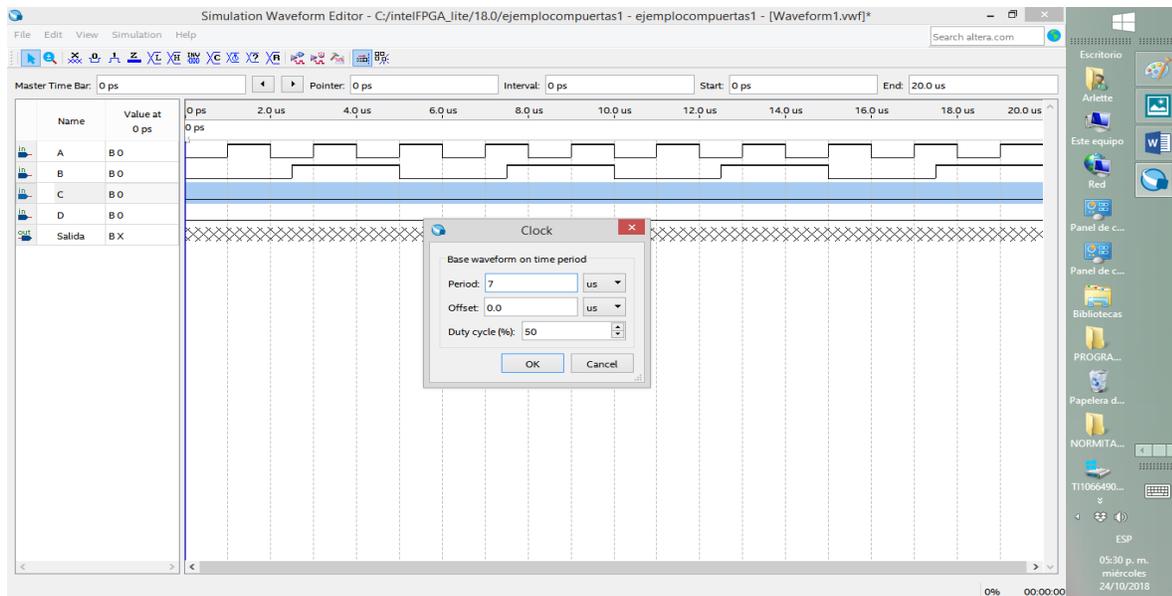


Figura 183 Proceso de simulación

Resultado

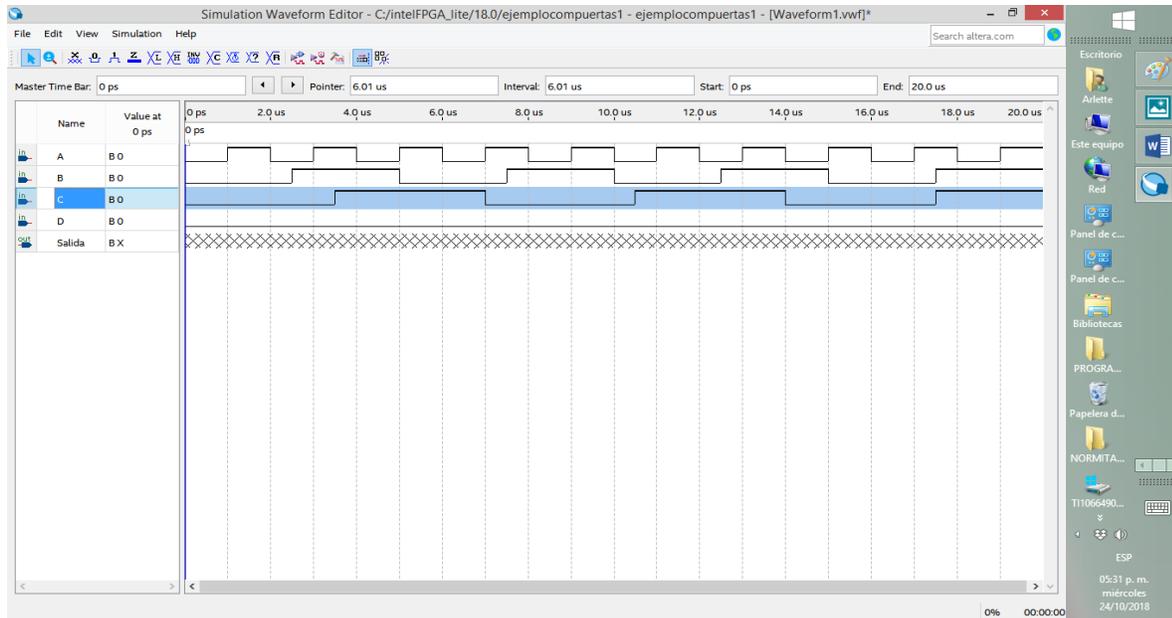


Figura 184 Proceso de simulación

Para la variable D

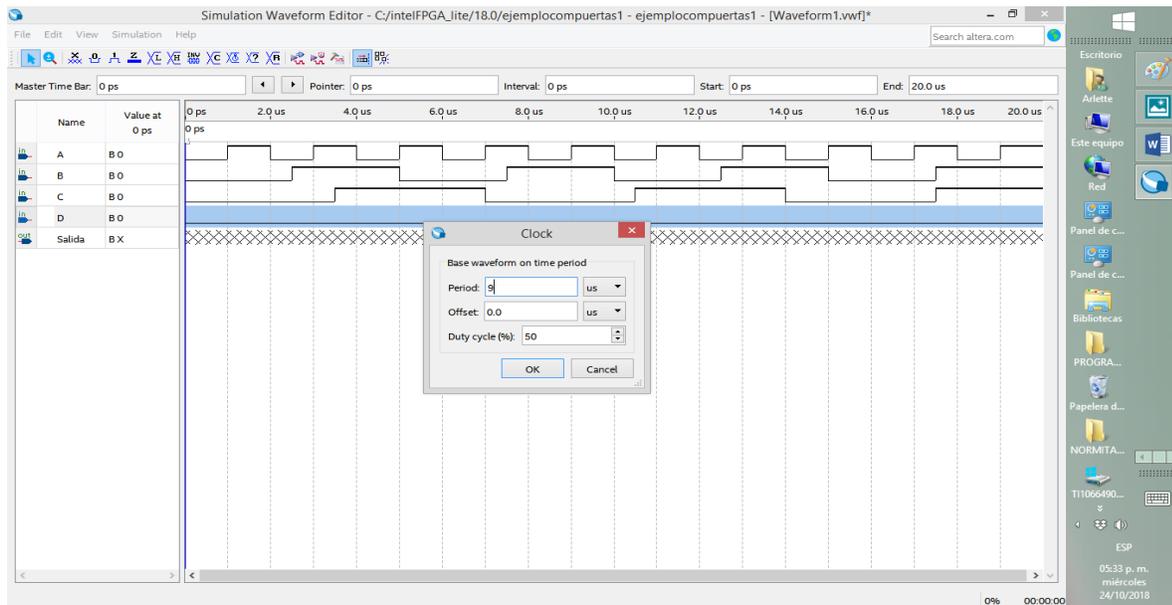


Figura 185 Proceso de simulación

Resultado de aplicar un periodo de 9us

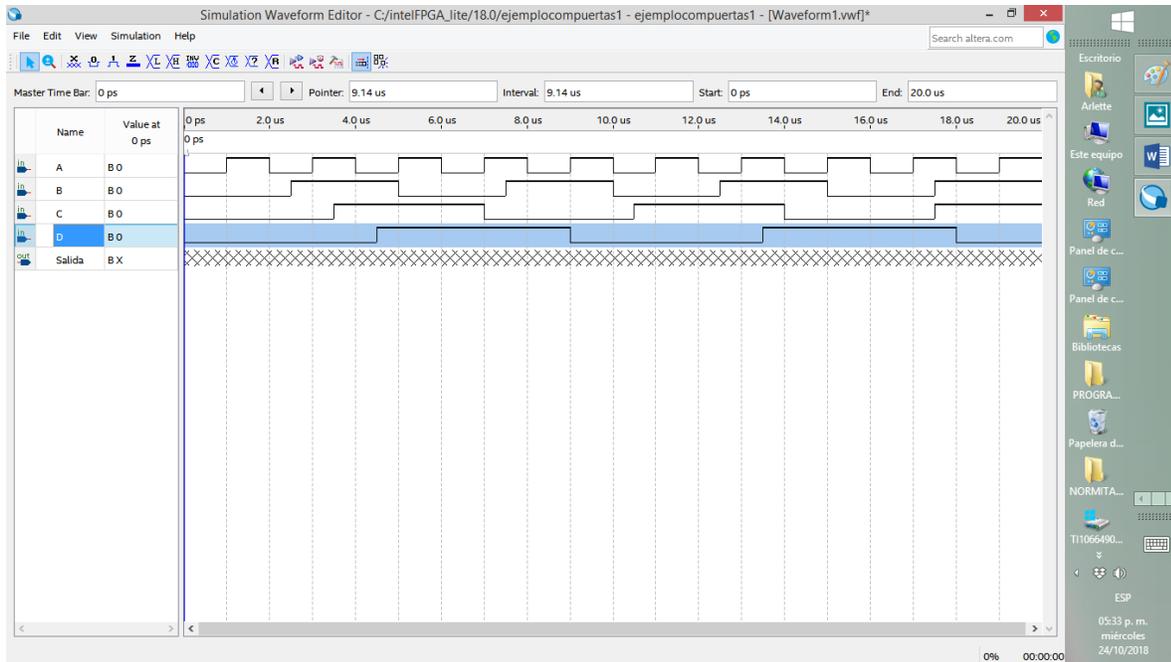


Figura 186 Proceso de simulación

Seleccionamos todas las variables de entrada

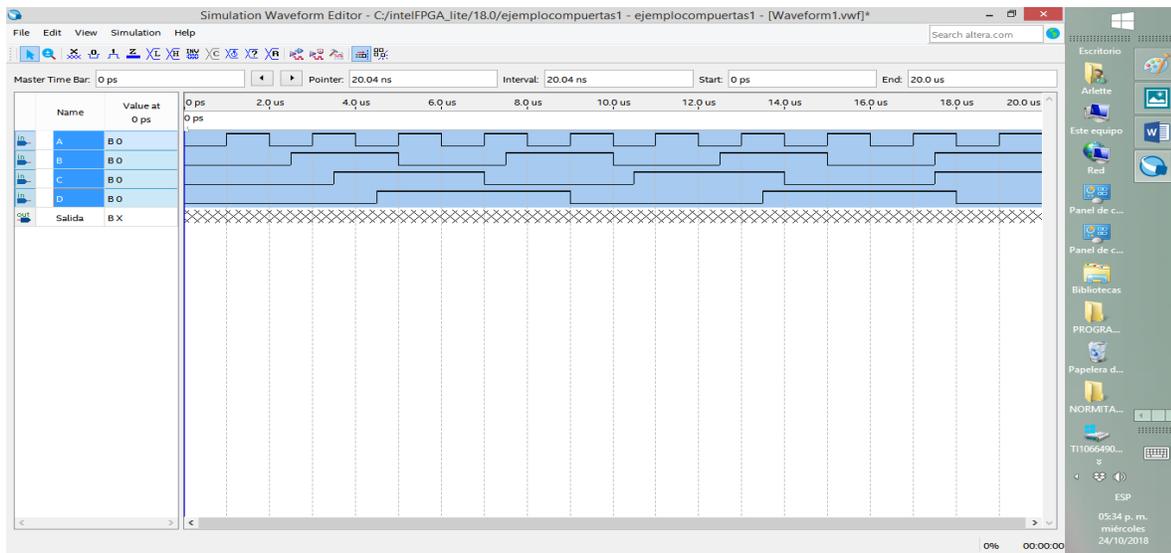


Figura 187 Proceso de simulación

Damos click derecho y seleccionamos la opción agrupar

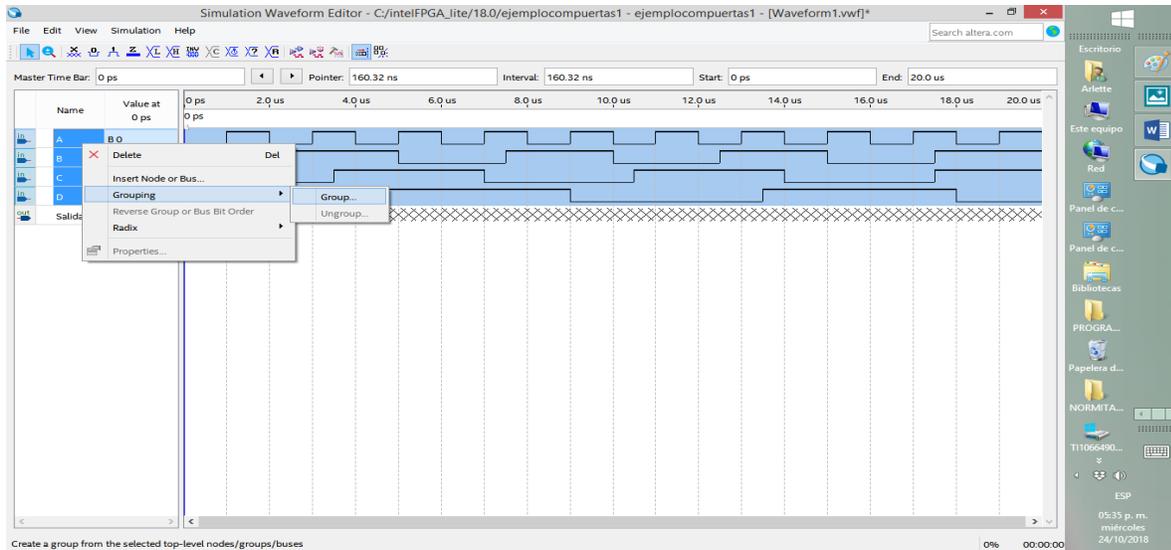


Figura 188 Proceso de simulación

Damos un nombre al grupo

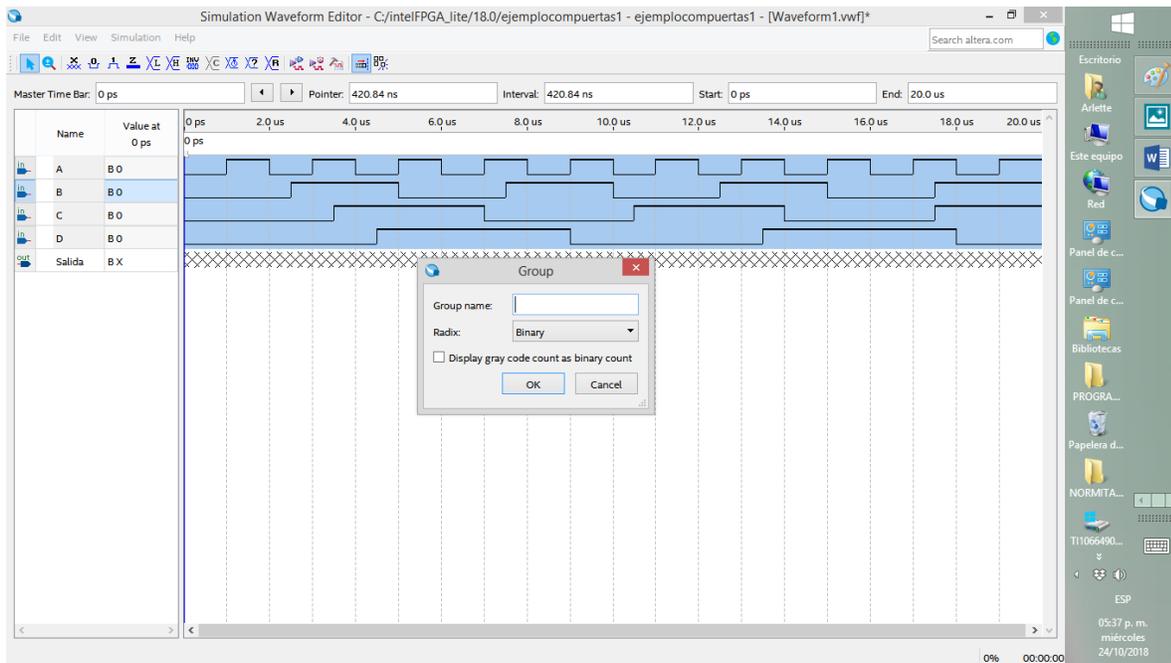


Figura 189 Proceso de simulación

Para este caso ejemplo 1

Y damos ok

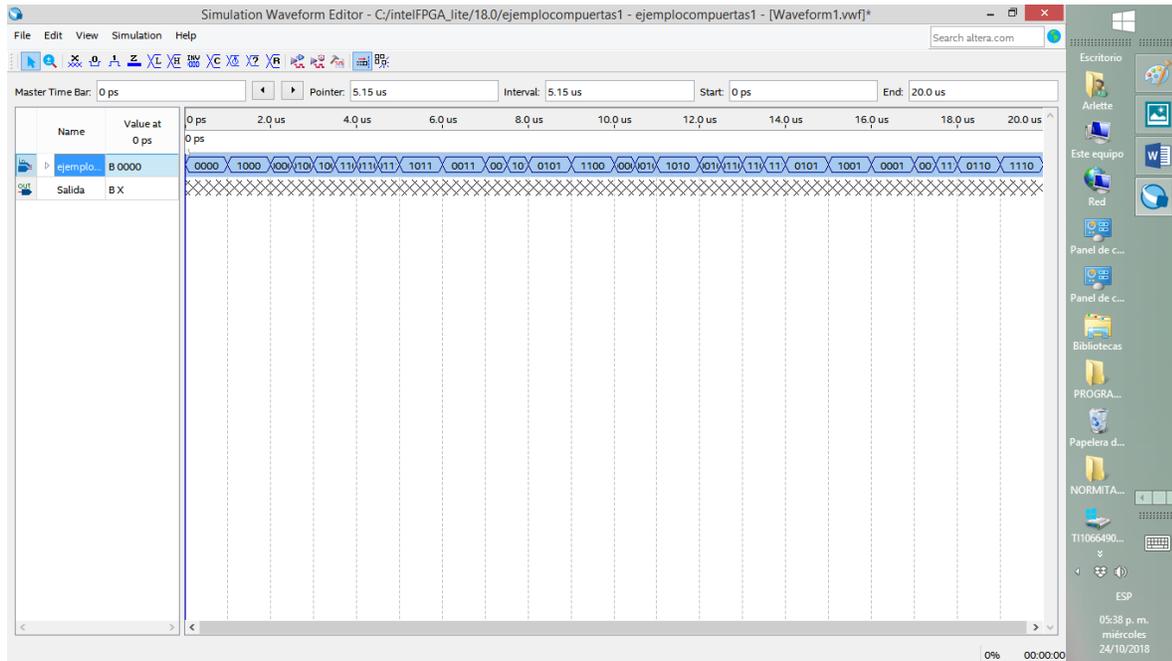


Figura 190 Proceso de simulación

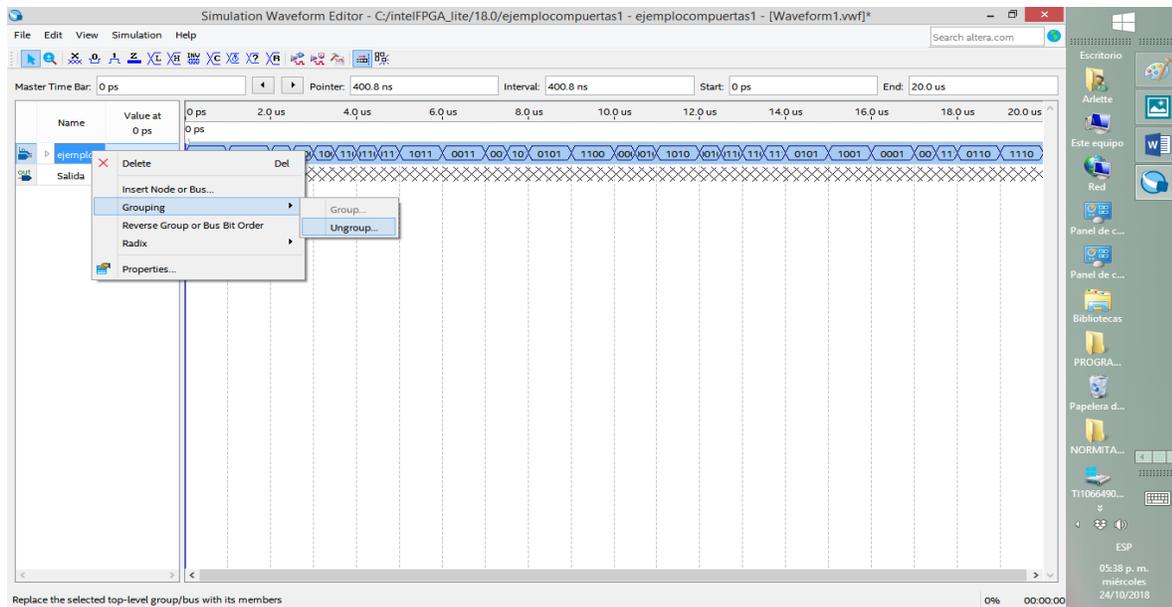


Figura 191 Proceso de simulación

Algo muy importante que hay que tomar en cuenta es que el sistema es posicional

Así que cambiamos el orden en el que están las variables al siguiente

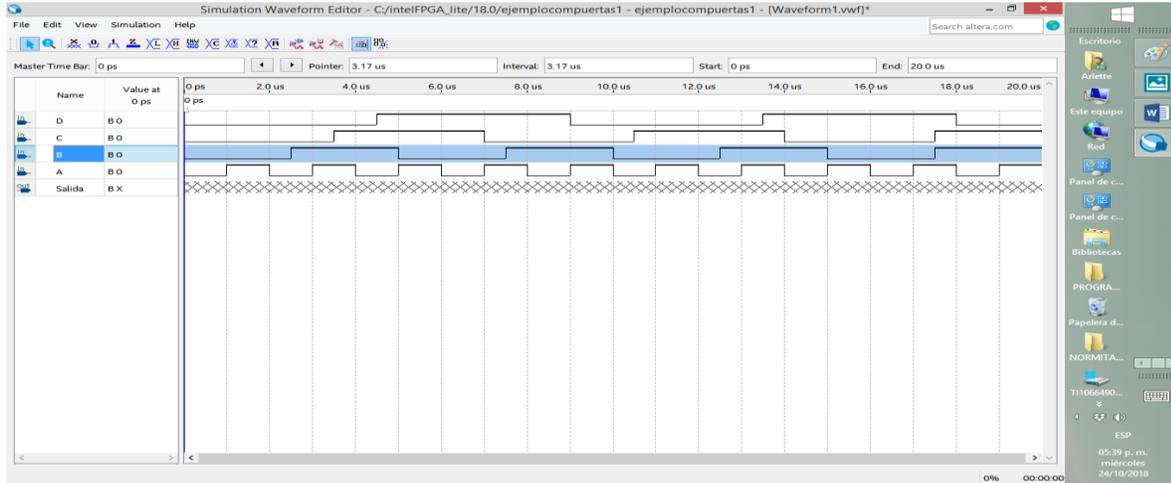


Figura 192 Proceso de simulación

Nota: Para cambiar el lugar de una variable la seleccionamos y la arrastramos hasta que ocupe la posición que deseamos
Volvemos a seleccionar las variables ahora con el nuevo orden

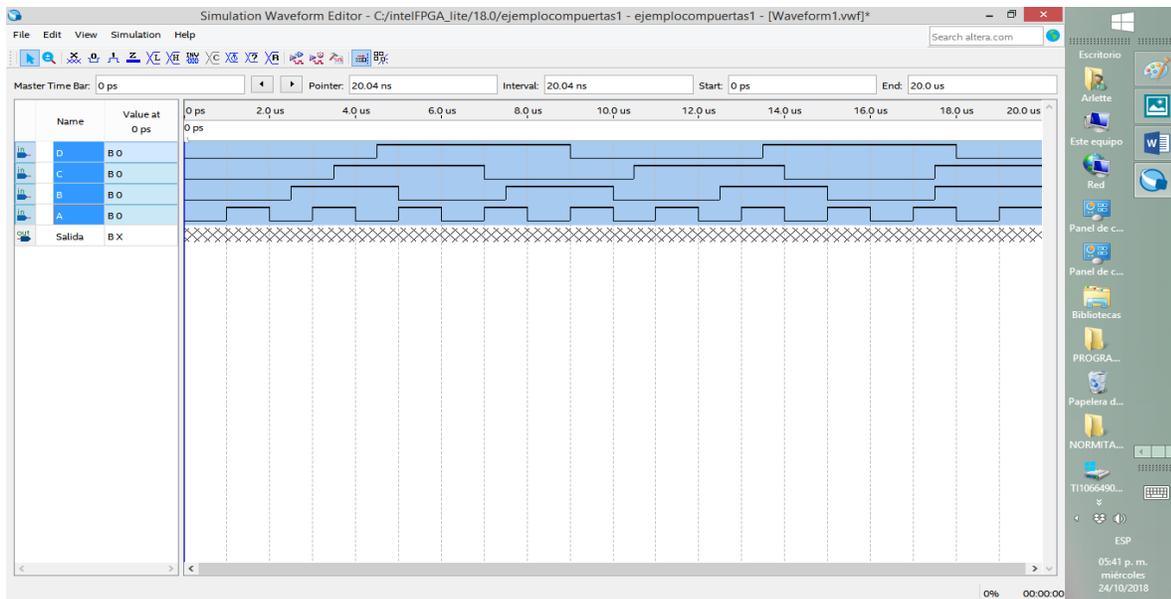


Figura 193 Proceso de simulación

Para tener un orden en los números

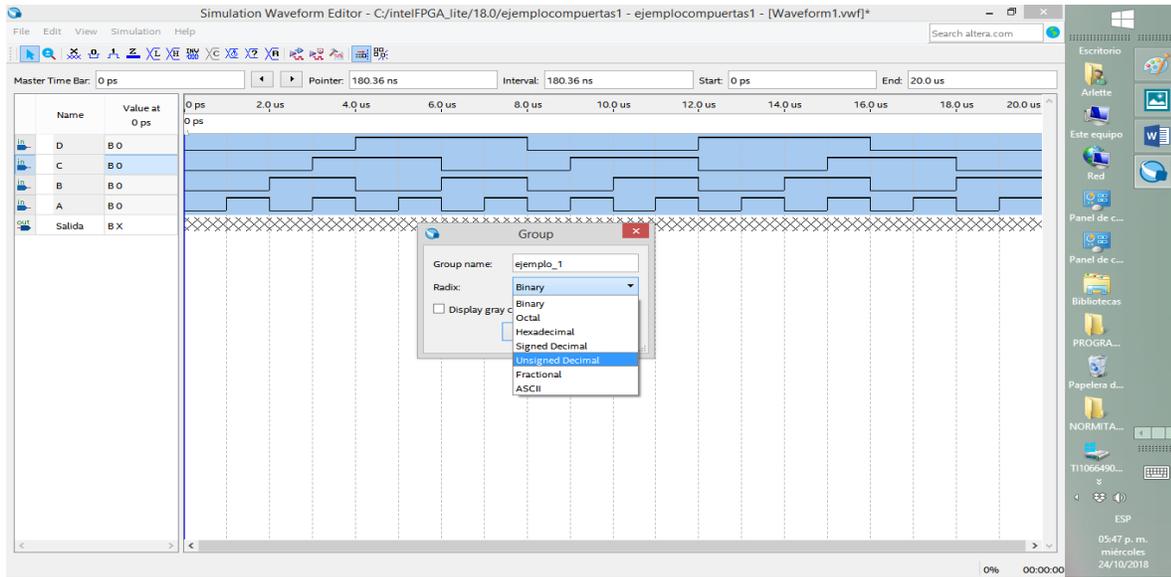


Figura 194 Proceso de simulación

Se tomó el mismo ejercicio solo que ahora los periodos se cambiaron a 1, 2, 4, 6 para A, B, C, D respectivamente

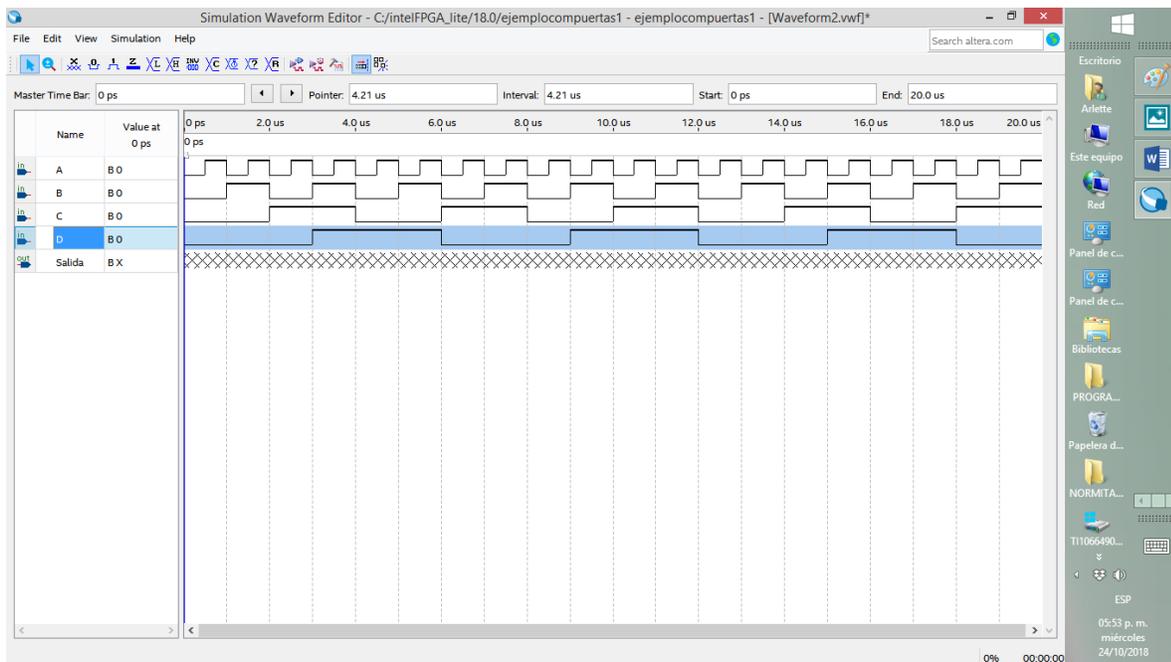


Figura 195 Proceso de simulación

Si ordenamos los números tendremos la siguiente pantalla

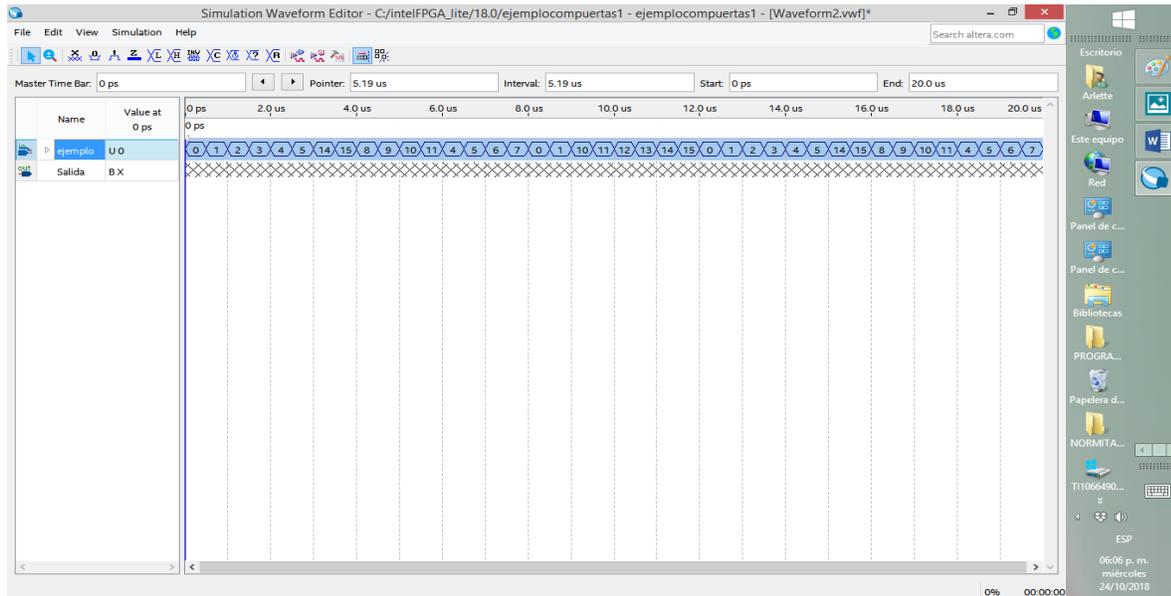
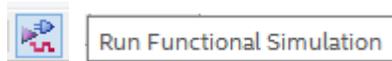


Figura 196 Proceso de simulación

PASO

Para llevar a cabo la simulación utilizamos el siguiente icono



Y aparece la siguiente pantalla

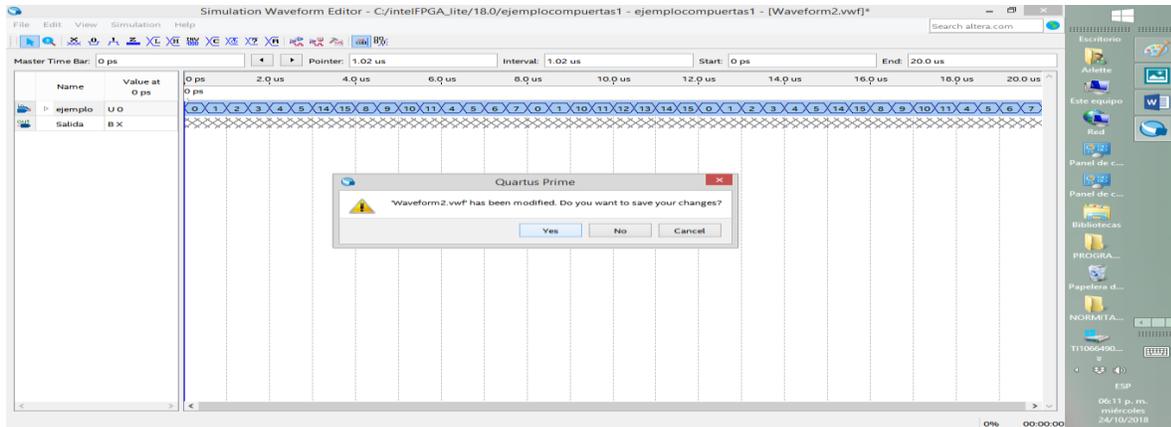


Figura 197 Proceso de simulación

Le indicamos Yes
Luego aparece la siguiente pantalla

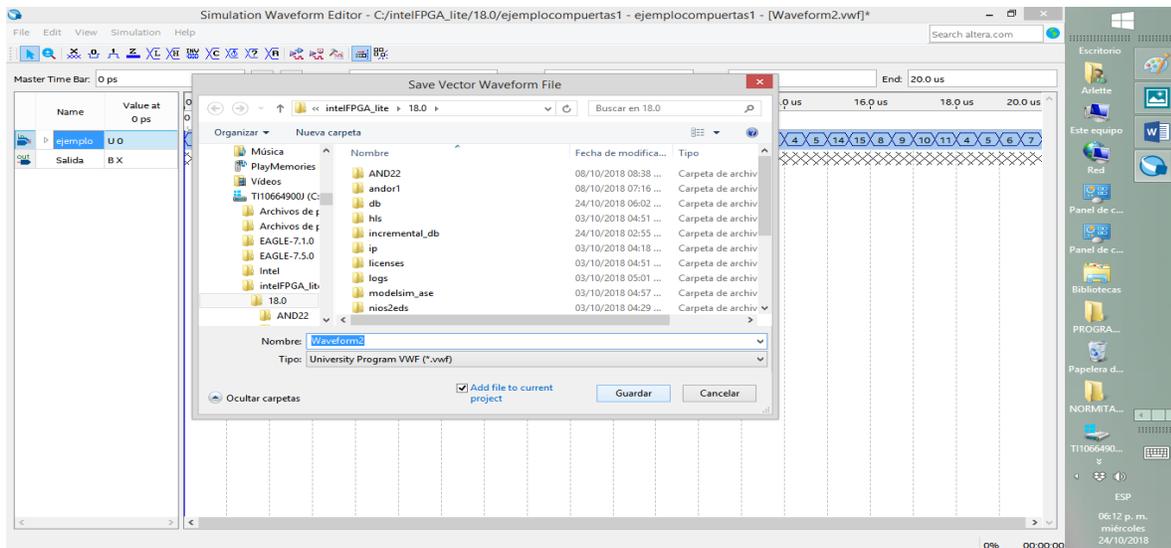


Figura 198 Proceso de simulación

Indicamos simplemente guardar
La simulación comienza a correr y si ha sido exitosa nos va mostrando un cuadro como este

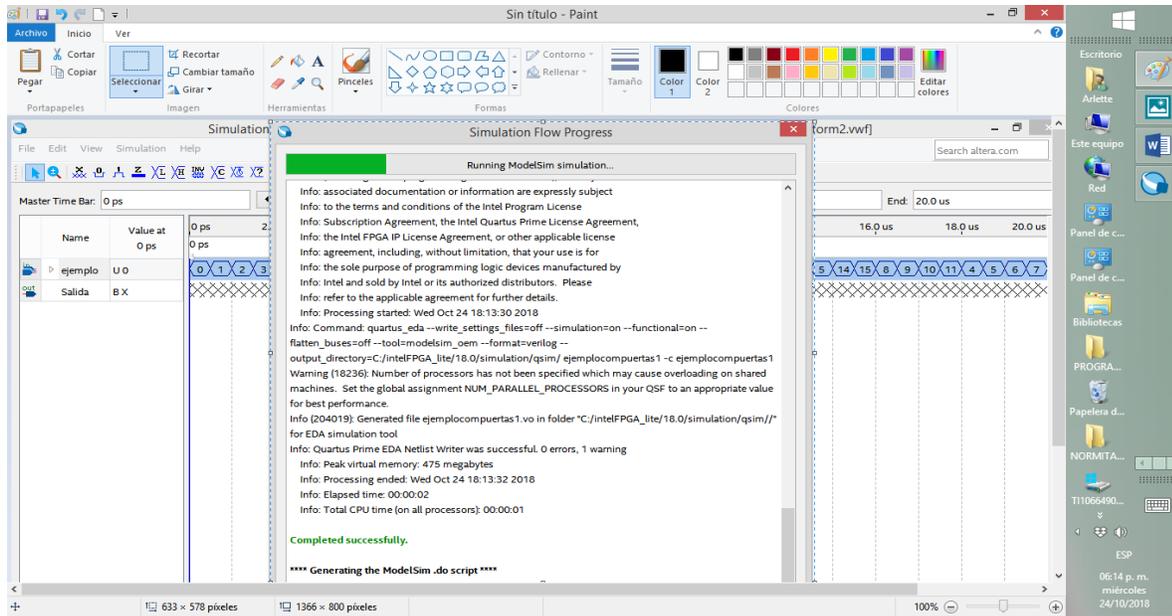


Figura 199 Proceso de simulación

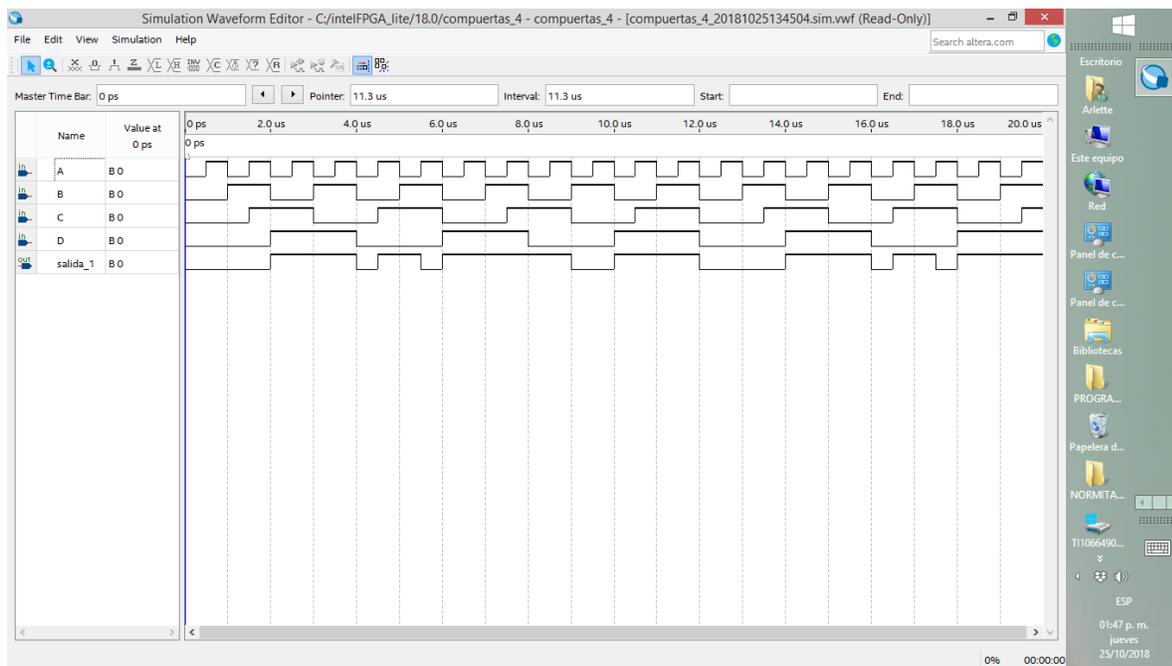


Figura 200 Proceso de simulación

Después de agrupar las entradas como se mostró anteriormente se tiene la siguiente imagen

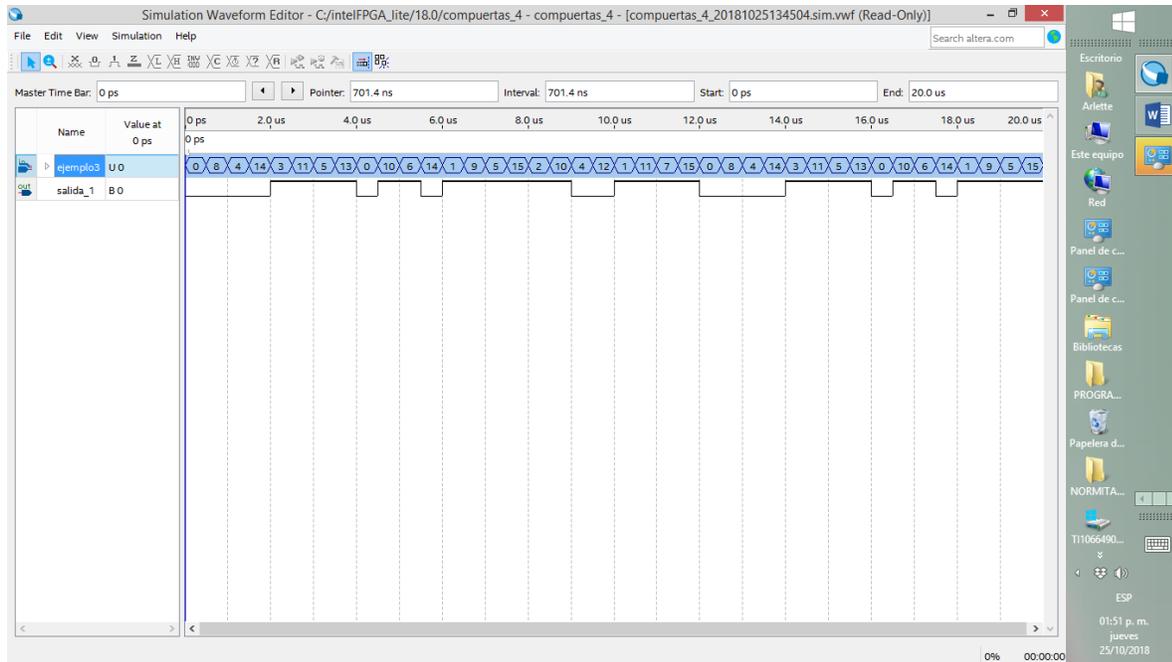


Figura 201 Proceso de simulación

Se muestra el resultado de la función en el siguiente cuadro dando los resultados con el correspondiente número en decimal.

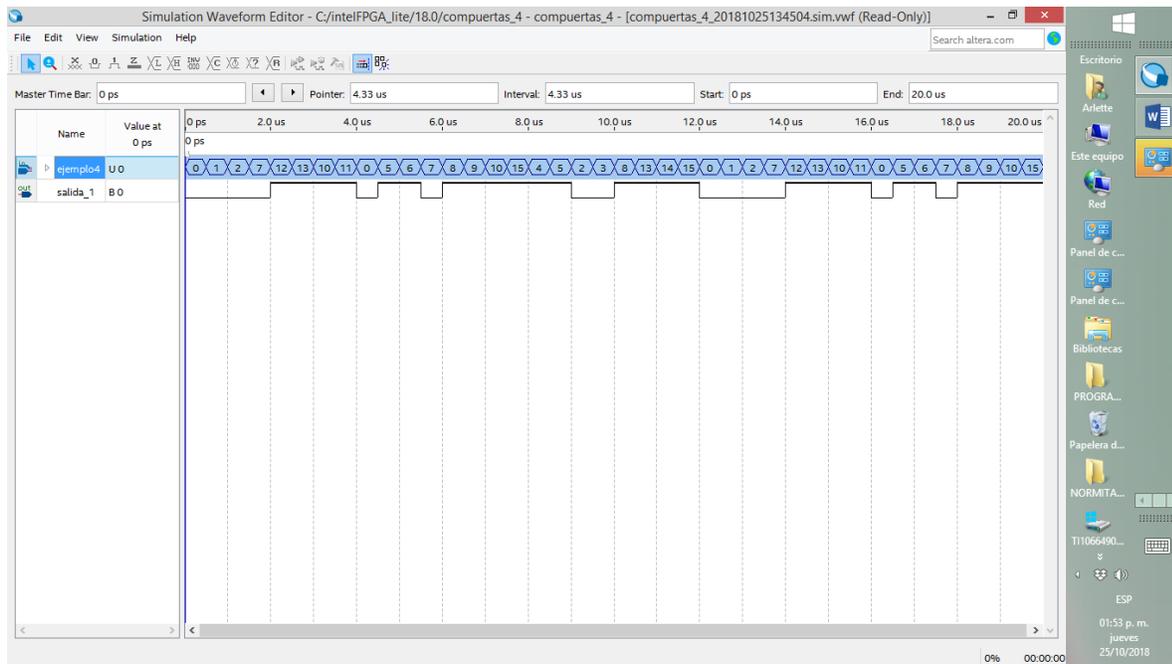


Figura 202 Proceso de simulación

RESULTADOS DE LOS EJERCICIOS PROPUESTOS

Ejemplo

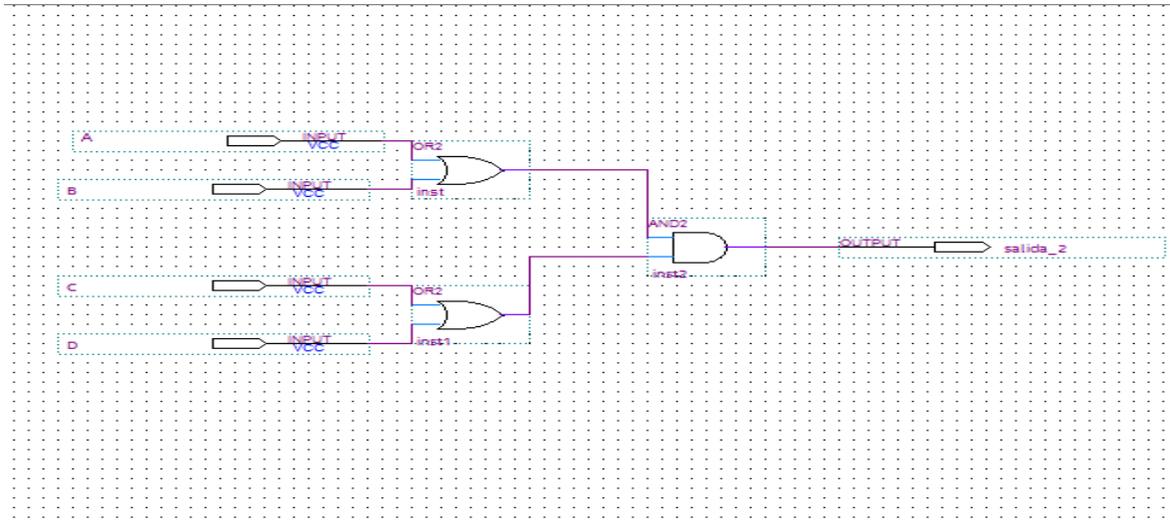


Figura 203 Imagen del ejemplo 1

RESULTADO DE LA SIMULACIÓN

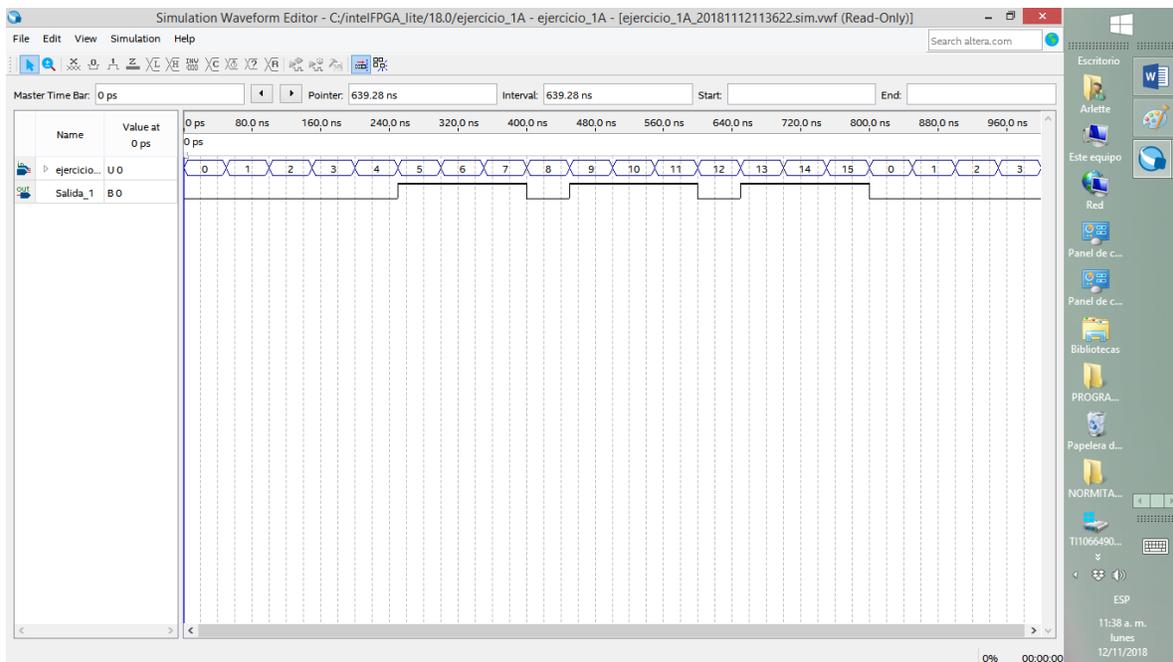


Figura 204 Resultado de la simulación del ejemplo 1

Se proporciona la tabla con los resultados de la función

Se proporciona la tabla con los resultados de la función

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

Tabla 153. Resultados de la función

Ejemplo 2

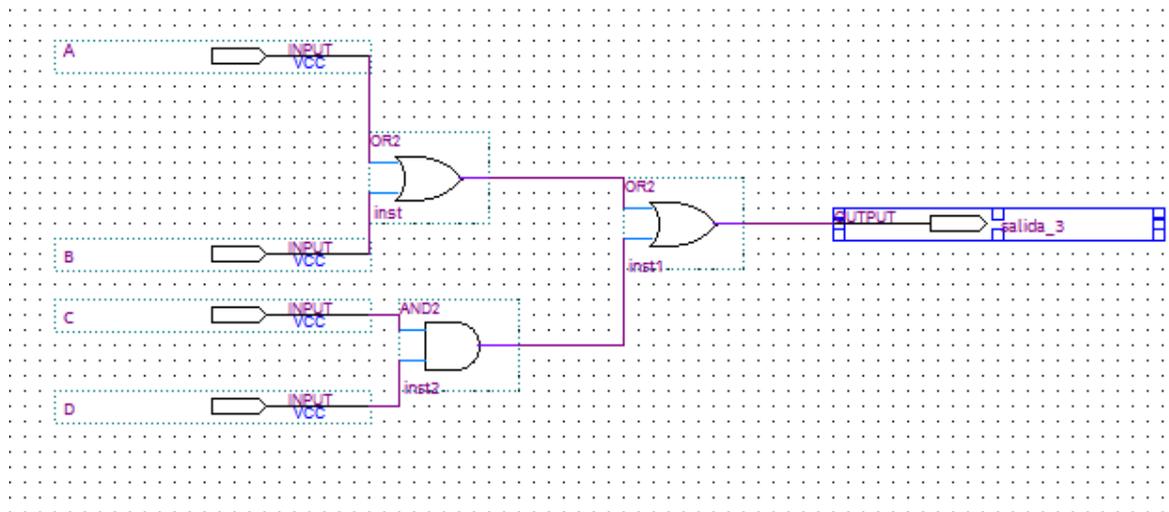


Figura 205 Diagrama del ejemplo 2

Tabla de resultados de los valores de salida de la función

A	B	C	D	S
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1
1	1	1	1	1

Tabla 154. Tabla de valores de salida de la función

Resultados de la simulación del ejercicio

Antes de la simulación procedemos a ordenar los números en una secuencia creciente, para la mejor visualización de los resultados, recordar que como solo hemos trabajado con 4 bits el número máximo posible que podemos obtener es el 15 decimal.

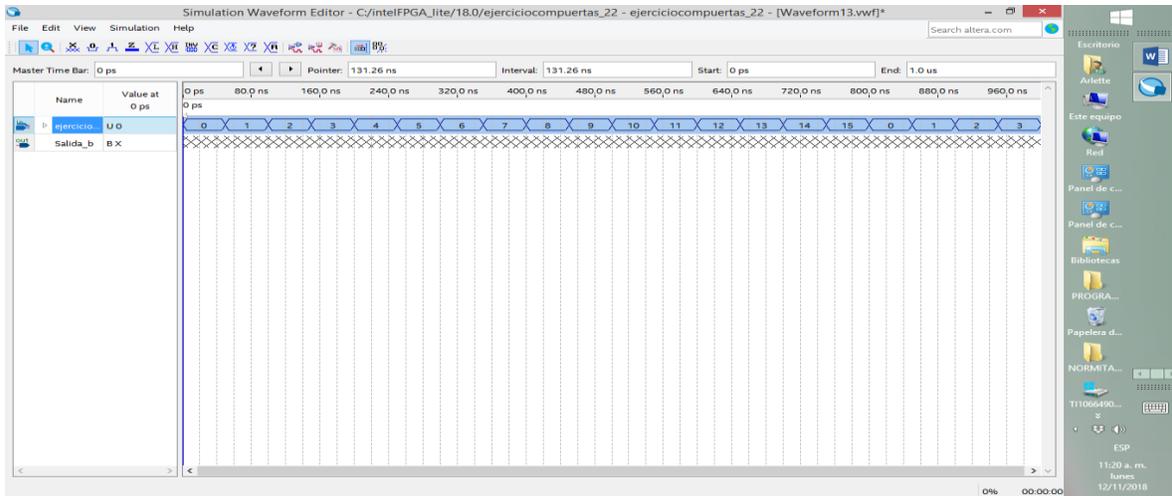


Figura 206 Simulación del resultado del ejemplo

Resultado después de proceder con la simulación

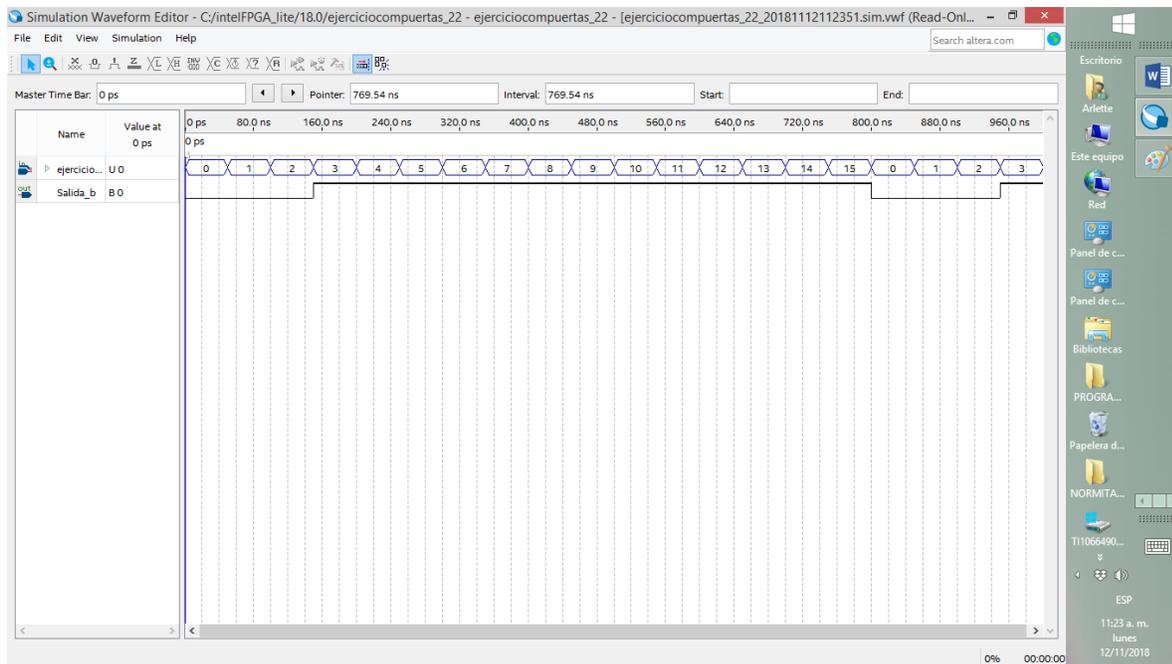


Figura 207 Resultado de la simulación

Desarrolla los siguientes ejercicios obteniendo el resultado de las funciones a mano y comprobando el resultado con la correspondiente simulación en Quartus.

Ejercicio 1

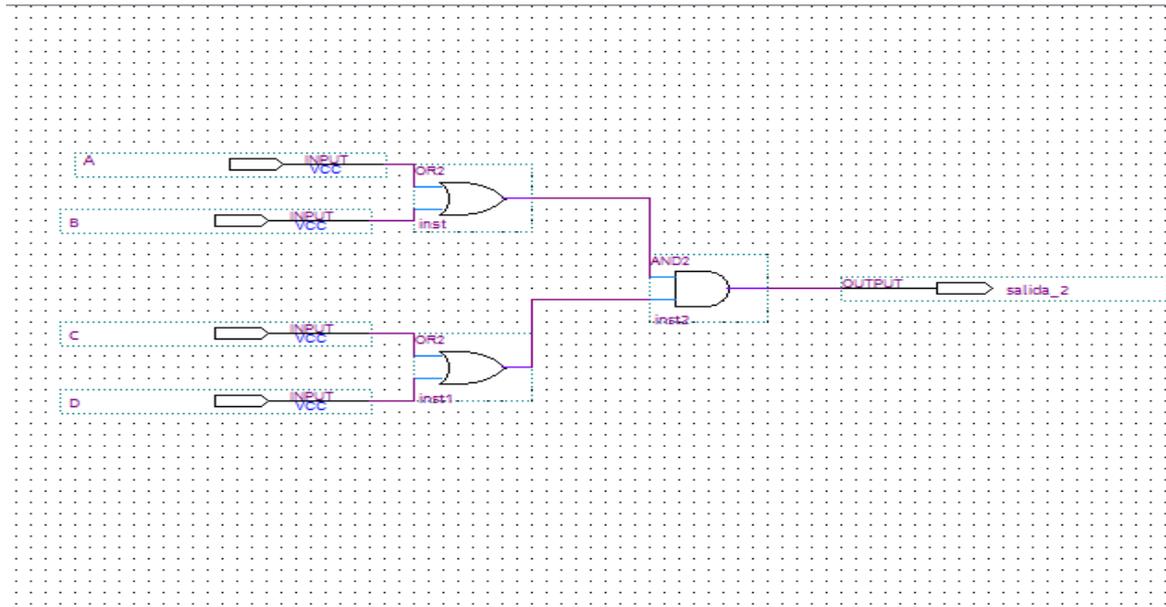


Figura 208 Diagrama del ejemplo

RESULTADO DE LA SIMULACIÓN

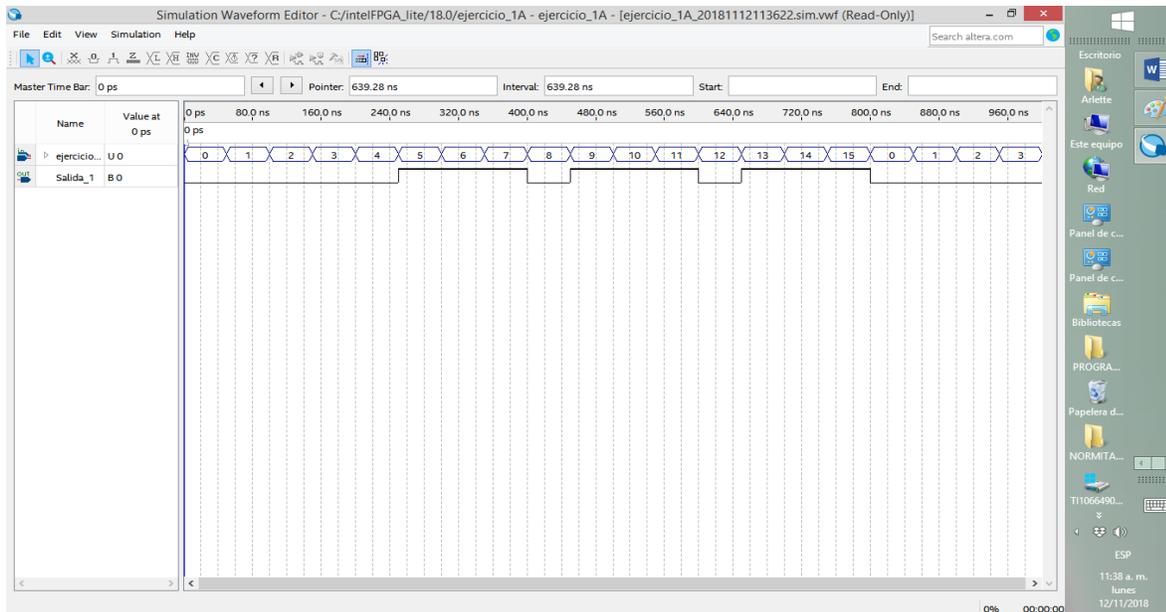


Figura 209 Resultado de la simulación

Se proporciona la tabla con los resultados de la función

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

Tabla 155. Tabla resultados de la función

Ejercicio 2

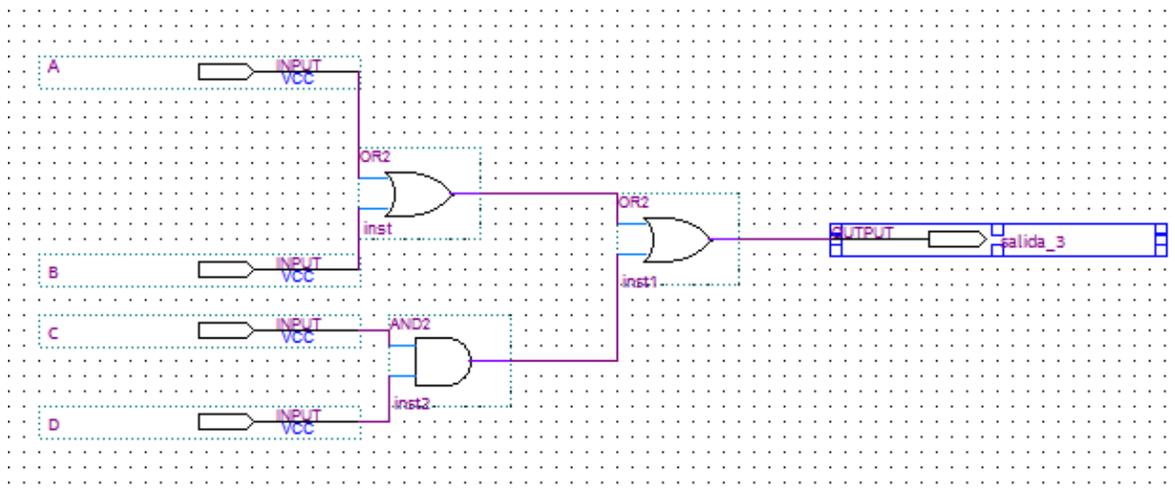


Figura 210 Diagrama de ejemplo

Tabla de resultados de los valores de salida de la función

A	B	C	D	S
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Tabla 156. Tabla de valores de salida de la función del ejercicio propuesto

RESULTADO DE LA SIMULACIÓN

Comprobé que el programa es eficiente.

Resultados de la simulación del ejercicio

Antes de la simulación procedemos a ordenar los números en una secuencia creciente, para la mejor visualización de los resultados, recordar que como solo hemos trabajado con 4 bits el número máximo posible que podemos obtener es el 15 decimal.

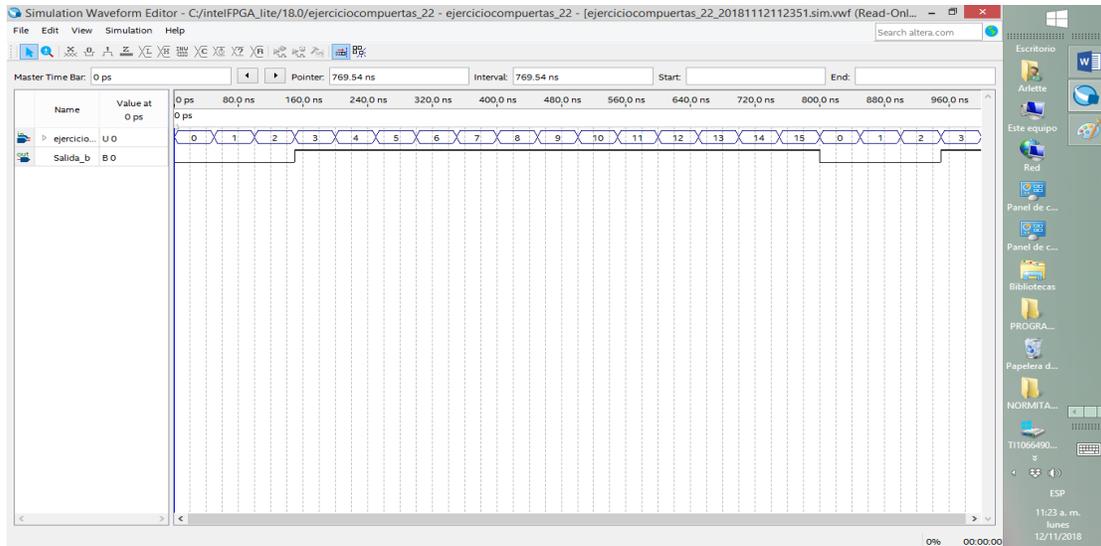


Figura 211 Simulación del ejemplo propuesto

CONCLUSIONES

El Manual de Apuntes Complementarios para la materia de Diseño Digital me permitió

✓ Mejorar y ampliar mi entendimiento de la materia.

✓ Comprender como funcionan eléctricamente muchos de los circuitos que se encuentran inmersos en la tecnología presente

✓ Comprender el principio de operación de los sistemas electrónicos con los cuales interactuamos diariamente como son: los cajeros automáticos, los cajeros para realizar los pagos de CFE, los cajeros de estacionamiento, el alumbrado público elaborados a base de sensores de luz, las lavadoras, los hornos de microondas, las computadoras por citar algunos ejemplos.

✓ Puedo identificar aplicaciones en las cuales puedo contar con un diseño a base de circuitos que establecen la lógica primaria del sistema.

➤ Considero que la materia de Diseño Digital es esencial para comprender los nuevos avances tecnológicos desde la creación de los circuitos hasta su implementación en sistemas físicos.

➤ Considero que elaborar el Manual me ayudo a desarrollar y mejorar múltiples habilidades como son

- La claridad y calidad de la redacción
- El análisis cuantitativo y cualitativo de los ejemplos

- Evaluación de la complejidad de los temas vistos

➤ La elaboración del Manual me llevó a una interpretación de la realidad más cercana a mis conocimientos, teniendo empatía por los dispositivos al comprender la lógica con la cual operan.

❖ Diseño Digital no solo es una materia que permite diseñar circuitos, si no que el hecho de incursionar en esta materia también proporciona las habilidades necesarias para llevar acabo diseño de sistemas complejos como puede ser el diseño de un entorno social, factible de variables, es por ello la

importancia de aprender está y todas materias de la carrera ya que brindan la posibilidad de abrir nuestros horizontes para aplicar nuestros conocimientos en cualquier área.

Hoy día la tecnología avanza rápidamente, cada vez los dispositivos tienen una mayor rapidez de respuesta, esto debido en gran parte a que cada vez albergan una mayor cantidad de componentes más sofisticados, sin duda alguna el diseño juega un papel muy importante en este punto pues gracias a esta materia se pueden comprender las bases que sustentan a los circuitos y mejorar el diseño.

El presente material es de gran valor tanto para alumnos como para profesores ya que ambos pueden enriquecerse al analizar las formas en las cuales se conduce al alumno para que adquiera los conocimientos de una manera reflexiva, precisa e innovadora.

20. BIBLIOGRAFÍA

Tocci Ronald J, Neal S. Widner, Gregory L. Moss. (2007).
Sistemas Digitales Principios y Aplicaciones. Ed. Decima. México.
Pearson Educación.

Floyd T. (2007). Fundamentos de Electrónica Digital.
Primera Edición. México. Limusa

M. Mano. (2003). Diseño Digital. Tercera Edición. México.
Pearson Educación

21. BIBLIOGRAFÍA DE INTERNET

Introducción

<http://www.elhospital.com/temas/Diseno-de-dispositivos-medicos+8090832>

Aparatos médicos digitales

https://medical.adanisystems.com/upload/uf/51d/ADANI_medical_full_brochure_spa_150715.pdf

Oxímetro de pulso

<https://oximetro.com.mx/blog/noticias/que-es-la-oximetria/>

Baumanómetro o esfigmomanómetro

<https://www.actuamed.com.mx/informacion-pacientes/baumanometro-digital>

Glucómetro

<https://www.diabetesjuntosxti.mx/noticias/como-funciona-el-glucometro/2010/03/>

Monitor Doppler fetal

https://es.wikipedia.org/wiki/Monitor_fetal_Doppler

Lógica secuencial

<https://docplayer.es/251676-Tema-5-sistemas-combinacionales-msi.html>

Expansión de sumadores

https://personales.unican.es/manzanom/Planantiguo/EDigital/Tema_V.pdf

Decodificador 2 a 4 programa

https://personales.unican.es/manzanom/planantiguo/edigital/d ecg9_09.pdf

Codificador programa

<https://susycursos.com/tag/codificador-de-4-a-2/>

Demultiplexor programa

http://www.itq.edu.mx/carreras/IngElectronica/archivos_contenido/Apuntes%20de%20materias/Apuntes_VHDL_2016.pdf

REFERENCIA DE FIGURAS

Fig. 1 Dispositivos médicos para obtener señales en forma analógica o digital

https://www.google.com/search?rlz=1C1AVNA_enMX574MX587&biw=1381&bih=774&tbm=isch&sa=1&ei=Y2I8XeSbFITosAX00JKwAg&q=imagenes+de+dispositvos+medicos&oq=imagenes+de+dispositvos+medicos&gs_l=img.3...431288.440181..440838...4.0..0.121.3909.43j3.....0....1..gws-wiz-img.....0j0i67.CybiO52wZXw&ved=0ahUKEwik2aCiqXjAhUENKwKHXS0BCYQ4dUDCAY&uact=5#imgrc=RbpyRdnwAVvuVM:

Fig. 2 termómetro analógico

https://www.google.com/search?q=IMAGEN+DE+TERMOMETRO+ANALOGICO&tbm=isch&source=iu&ictx=1&fir=00k3FHPXscsnQM%253A%252CLMKP6cAXRVcnaM%252C_&vet=1&usg=AI4_-kSOMeuRv-PEI1MwGtq8ak48kDRiXw&sa=X&ved=2ahUKEwiqxrq9qdXjAhVHMqwkHQcHBHYQ9QEwAXoEAcQBg#imgrc=00k3FHPXscsnQM:

Fig.3 termómetro digital

https://www.google.com/search?rlz=1C1AVNA_enMX574MX587&biw=1381&bih=774&tbm=isch&sa=1&ei=VWI8Xeb1CibatAXwq4G4AQ&q=imagen+termometro+digital&oq=imagen+termometro+digital&gs_l=img.3..0.9413.11861..13108...0.0..0.117.1387.11j4.....0....1..gws-wiz-img.....0i67j0i7i30j0i7i5i30j0i8i7i30.Eb41nnQqkLk&ved=0ahUKEwim9L6bqtXjAhUGLa0KHfBVABcQ4dUDCAY&uact=5#imgrc=RuK4_xGye32V0M:

Fig. Imagen de aplicación industrial

https://www.google.com/search?q=imagen+de+ingenieria+industrial&rlz=1C1AVNA_enMX574MX587&tbm=isch&source=iu&ictx=1&fir=jzwDlBt5uBcuxM%253A%252CpBc5XzW15VBMrM%252C_&vet=1&usg=AI4_-kQ2XTKLkOoU0Jmo9xv8I5HeJEhT6g&sa=X&ved=2ahUKEwie-

[9vijaPpAhWsHjQIHUibDGkQ9QEwCXoECAoQLg&biw=1382&bih=774#imgrc=KnEfsrOZKYyu1M](https://www.google.com/search?q=imagen+de+industria+quimica&rlz=1C1AVNA_enMX574MX587&tbm=isch&source=iu&ictx=1&fir=z8szhPVHT4VHjM%253A%252CPf753qMax5GhqM%252C_&vet=1&usg=AI4_-kRJAYD0qj3_OrI48DUINZTtiz4zw&sa=X&ved=2ahUKEwjv783MtcXpAhVaVs0KHRYRaDnYQ9QEwB3oECAoQKg&biw=1382&bih=774#imgrc=KnEfsrOZKYyu1M)

fig imagen de aplicación en la industria química

https://www.google.com/search?q=imagen+de+ingenieria+de+sistemas&rlz=1C1AVNA_enMX574MX587&tbm=isch&source=iu&ictx=1&fir=7nBGFcUZxn6qAM%253A%252C8O7oE3grLxhOzM%252C_&vet=1&usg=AI4_-kQfiojAfVvbuph8IXSvnt-0vBaoMw&sa=X&ved=2ahUKEwjknmit8XpAhUOQq0KHfEuB8wQ9QEwBnoECAoQKA&biw=1382&bih=774#imgrc=7nBGFcUZxn6qAM:

Fig imagen de aplicación en sistemas

https://www.google.com/search?q=imagen+de+ingenieria+de+sistemas&rlz=1C1AVNA_enMX574MX587&tbm=isch&source=iu&ictx=1&fir=7nBGFcUZxn6qAM%253A%252C8O7oE3grLxhOzM%252C_&vet=1&usg=AI4_-kQfiojAfVvbuph8IXSvnt-0vBaoMw&sa=X&ved=2ahUKEwjknmit8XpAhUOQq0KHfEuB8wQ9QEwBnoECAoQKA&biw=1382&bih=774#imgrc=7nBGFcUZxn6qAM:

Aparatos médicos digitales

Fig. 4 Oxímetro de pulso

<https://oximetro.com.mx/blog/noticias/como-funciona-el-oximetro-de-pulso/>

Fig. 5 Baumanómetro

https://www.google.com/search?q=baumanometro+digital+imagen&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjopa7W3NjjAhVOKawKHZOpD8kQ_AUIESgB&biw=1381&bih=774#imgrc=YSGmiYihJqtooM:

Fig.6 Glucómetro

https://www.google.com/search?q=glucometro+imagen&rlz=1C1AVNA_enMX574MX587&tbm=isch&source=iu&ictx=1&fir=zZHT-27jzZG8GM%253A%252Cw60qOJLtJy9dVM%252C_&vet=1&usg=AI4_-kTG2F2ZSr44YMC_f2h67HwdEMkmWw&sa=X&ved=2ahUKEwiPj6aT5djjAhUJD60KHVr6AOMQ9QEwA3oECAkQDw#imgrc=zZHT-27jzZG8GM:

Fig. 7 Monitor Doppler fetal

https://www.google.com/search?q=imagen+monitor+doppler+fetal&rlz=1C1AVNA_enMX574MX587&tbm=isch&source=iu&ictx=1&fir=bXP5Tup5_LmD_M%253A%252C6AuEvZW9P0vHQ M%252C_&vet=1&usg=AI4_-kQNcr41N_PLU45n5oxISckB46vx-A&sa=X&ved=2ahUKEwjM7sPpgNnjAhUDP6wKHSRkA_gQ9QEwAHoECAkQBg#imgrc=bXP5Tup5_LmD_M:

Fig. 8 Circuito interno del oxímetro de pulso

https://www.google.com/search?q=imagen+de+los+circuitos+de+oxímetro+de+pulso&rlz=1C1AVNA_enMX574MX587&tbm=isch&source=iu&ictx=1&fir=GouRo6QZZCJkEM%253A%252C CVy1S7sS5rjgvM%252C_&vet=1&usg=AI4_-kQvCF45Hb0nQv-K9pNp-1CGgRilVg&sa=X&ved=2ahUKEwj715-8_tjjAhUDD60KHbNRCAQQ9QEwAnoECAkQDA#imgrc=GouRo6QZZCJkEM:&vet=1

Fig 85 Figura de chip

https://www.google.com/search?q=imagen+de+chip+electrónico&rlz=1C1AVNA_enMX574MX587&tbm=isch&source=iu&ictx=1&fir=3Vu4IyIT4QCnaM%253A%252C3JjMnuTKUeWkOM%252C_&vet=1&usg=AI4_-kS55LWXiahqXzQNht6PMlh-OmJ6_Q&sa=X&ved=2ahUKEwja54DmjPzjAhUJ7qwKHWtXBvQQ9QEwAXoECAkQCQ#imgrc=3Vu4IyIT4QCnaM:&vet=1

Fig. 86 Imagen de circuitos integrados

https://www.google.com/search?rlz=1C1AVNA_enMX574MX587&q=imagen+de+circuitos+integrados&tbm=isch&source=univ&sa=X&ved=2ahUKEwit9bGanvzjAhUHSa0KHbsSBYoQ7Al6BAgHEBs&biw=1381&bih=774#imgrc=ZGBBJnXSp6X2MM:

Fig. 87 Imagen de circuito integrado 74LS148

<https://www.carrod.mx/products/ci-ttl-codificador-prioritario-de-8-a-3-lineas-74ls148>

Fig. 113 Análisis de un entorno

<https://gruposolutia.com/aplicando-el-concepto-byod-al-entorno-educativo/>

Fig. 122 Imagen de corredor con reloj en mano

<https://escueladerunning.com/entrenamiento-intervalos/>

Fig. 123 Imagen de reloj

<https://es.aliexpress.com/item/32922867051.html>

Fig. 124 Imagen de brazo robótico

<https://www.amazon.es/aluminio-AS-6DOF-rob%C3%B3tico-incluyendo-el%C3%A9ctrico/dp/B00NBPRQXK>

Fig. Imagen de circuito integrado 74154

<https://articulo.mercadolibre.com.ve/MLV-521707779-nte-74154-ci-ttl-24pin-sn74154n-nte74154- JM?quantity=1>

Fig. Imagen de circuito integrado 7447

<http://laboratorioelectronica.blogspot.com/2013/08/decodificador-bcd.html>

Fig. Imagen de circuito integrado 7448

https://www.google.com/search?q=codificador+7448&tbm=isch&ved=2ahUKEwj_mKzGIJjoAhVSYKwKHXhgAP8Q2-cCegQIABAA&oq=codificador+7448&gs_l=img.3...399602.401154..401941...0.0..0.137.239.0j2.....0....1..gws-wiz-img.....0i67j0i10i24.wKacIH49k9k&ei=rdtrXr-OOtLAsQX4wIH4Dw&rlz=1C1AVNA_enMX574MX587#imgrc=RIVttBqrQx9zVM&imgdii=UvByTPaSoTl2FM