



Universidad Autónoma del Estado de México

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
CENTRO UNIVERSITARIO UAEM ATLACOMULCO

MANUAL DE PRÁCTICAS DE LABORATORIO

“SISTEMAS DIGITALES”

Programa: Ingeniería en Computación F19

Autores:

Dr. Allan Antonio Flores Fuentes

Dr. Juan Fernández García Mejía

Aprobado por los H.H. Consejos:

Consejo Académico

Consejo de Gobierno

Agosto 2022





Universidad Autónoma del Estado de México

ÍNDICE

| | Página |
|---|---------------|
| Presentación | 3 |
| Programa de Estudio | 4 |
| Práctica 1 Comprobación de compuertas lógicas | 5 |
| Práctica 2 Circuitos combinacionales | 12 |
| Práctica 3 Simulación de sistemas digitales secuenciales | 17 |
| Práctica 4 Simulación de Circuitos Integrados lineales | 20 |
| Practica 5 Decodificador BCD a 7 segmentos | 23 |
| Practica 6 Sumador y Restador completo | 28 |
| Practica 7 Simulacion de multiplexor de 2 entradas cada una de 4 bit | 36 |
| Practica 8 Comparador de magnitud | 40 |





Universidad Autónoma del Estado de México

Presentación

Las tecnologías de hardware permiten la implementación de sistemas cada vez más a la medida de las necesidades del usuario. Es una gran ventaja para el profesional de la computación conocer el diseño de los sistemas digitales; tanto combinacionales como secuenciales, que permiten la ejecución de las tareas específicas que desea implementar, ya que al saber cómo funciona el hardware, la programación se hace en función de éste, optimizando la velocidad de procesamiento y el espacio de la computadora.

Este manual de prácticas, tiene como objetivo tratar a los circuitos secuenciales y combinacionales desde el punto de vista hardware y software, es decir, llevar una integración de ambos conceptos con la finalidad de ser útil como antecedente a los sistemas digitales implementados en Lenguaje de Descripción de Hardware (HDL).

Propósito general

Conocerá el propósito de la simulación de sistemas digitales, secuenciales y combinatorios, en hardware y software, y poder proponer soluciones prácticas a problemas reales.

Competencias genéricas

Identificar los componentes y funcionamiento de una simulación de sistemas digitales; secuenciales y combinatorios.

Ámbitos de desempeño

El proceso de enseñanza-aprendizaje se realizará mediante sesiones previas de explicación en clase.





Universidad Autónoma del Estado de México

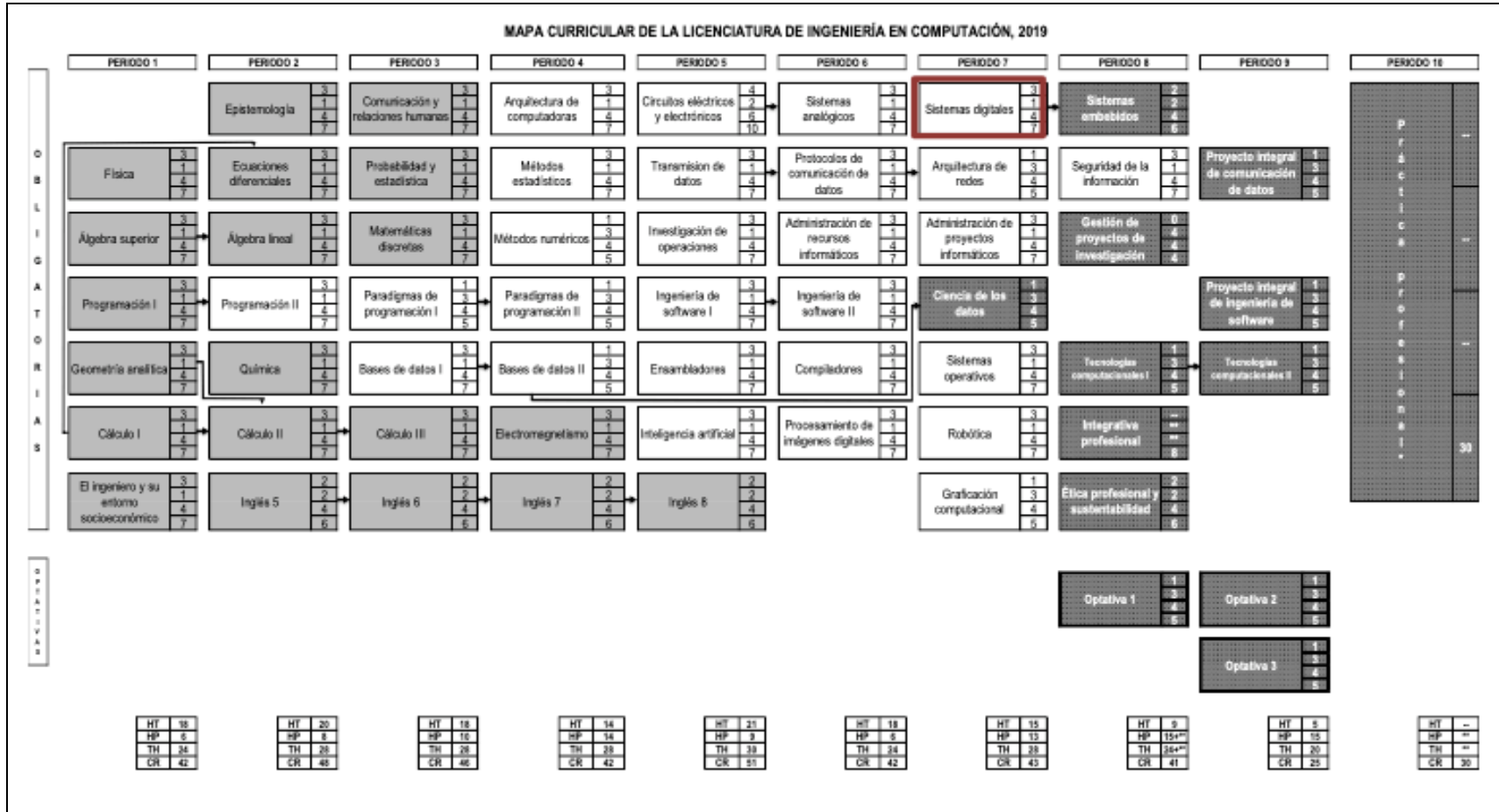


Figura A.- Programa de estudio F19, ubicación de la UA Sistemas Digitales.



Universidad Autónoma del Estado de México

PRÁCTICA 1: Comprobación de compuertas lógicas.

≡ Competencia a desarrollar

Comprobar el funcionamiento y operaciones de las compuertas lógicas, así como sus estados lógicos.

Introducción

Compuerta NOT o compuerta inversora.

La compuerta NOT como la compuerta AND y la compuerta OR es muy importante. Esta compuerta entrega en su salida el inverso (opuesto) de la entrada.

El símbolo y la tabla de verdad se muestra en la figura 1.1.



(a)

| A | X |
|---|---|
| 0 | 1 |
| 1 | 0 |

(b)

Figura 1.1.- a) Símbolo compuerta NOT, b) Tabla de verdad.

La salida de una compuerta NOT tiene el valor inverso al de su entrada. En el caso la figura 1.1 la salida $X = A$.

Esto significa que:

1. Si a la entrada tenemos un “1” lógico, a la salida hará un “0” lógico.
2. Si a la entrada tenemos un “0” lógico a la salida habrá un “1” lógico.

Nota: El apóstrofe en la siguiente expresión significa “negado”. Entonces: $X' = A$.





Universidad Autónoma del Estado de México

Las compuertas NOT se pueden conectar en cascada, logrando después de dos compuertas, la entrada original. Ver figura 1.2, donde se muestra la compuerta y la tabla de verdad.

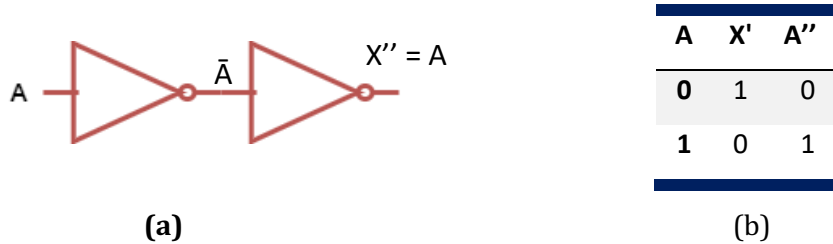


Figura 1.2.-a) Compuerta NOT en cascada, b) tabla de verdad,

Un motivo para implementar un circuito que tenga en su salida, lo mismo que tiene en su entrada, es conseguir un retraso de la señal original, esto depende de la tecnología con la que esté contruida el Circuito Integrado, pero puede estar en el orden de ~ 20 ns.

1.1. La compuerta AND o/y lógica.

Su representación es la que se muestra en las siguientes figuras.

La primera es la representación de una compuerta AND de 2 entradas y la segunda de una compuerta AND de 3 entradas, en las figuras 1.3.a y 1.3.b.



Figura 1.3 Compuertas AND, a) 2entradas, b) 3 entradas.

La compuerta AND lógica más conocida tiene dos entradas A y B, aunque puede tener muchas más (A, B, C,...n, donde n es un número entero), y tener solo una salida X.

La compuerta AND de 2 entradas tiene la siguiente tabla de verdad.





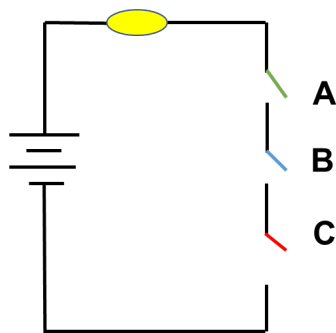
Universidad Autónoma del Estado de México

| A | B | X |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Figura 1.4. Tabla de verdad de AND 2 entradas.

Se puede ver claramente que la salida X solamente es “1” (1 lógico, nivel alto) cuando la entrada A como la entrada B están en “1”. En otras palabras; la salida X es igual a 1 cuando la entrada A y la entrada B son 1. Esta situación se representa en algebra booleana como: $X = A*B$ ó $X = AB$.

Una compuerta AND de 3 entradas se puede implementar con interruptores, como se muestra en el diagrama de la figura 1.5.



(a)

| A | B | C | Lámpara |
|---|---|---|------------------|
| A | A | A | Apagada |
| A | A | C | Apagada |
| A | C | A | Apagada |
| A | C | C | Apagada |
| C | A | A | Apagada |
| C | A | C | Apagada |
| C | C | A | Apagada |
| C | C | C | Encendida |

(b)

Figura 1.5. Compuerta AND de 3 entradas, y b) Tabla de verdad.





Universidad Autónoma del Estado de México

En la tabla de verdad se muestra en la figura 1.5, a la derecha A = Abierto y C = Cerrado. Una compuerta AND puede tener muchas entradas. Una compuerta AND de múltiples entradas puede ser creada conectando compuertas simples en serie. El problema de poner compuertas en cascada es que el tiempo de propagación de la señal desde la entrada hasta la salida, aumenta. Si se necesita una compuerta AND de 3 entradas y no una hay disponible, es fácil crearla con dos compuertas AND de 2 entradas en serie cascada como se muestra en la figura 1.6.

Se observa que la tabla de verdad correspondiente es similar a la mostrada anteriormente, donde se utilizan interruptores. Se puede deducir que el tiempo de propagación de la señal de la entrada C es menor que los de las entradas A y B (Estas últimas deben de propagarse por dos compuertas mientras que la entrada C se propaga solo por una compuerta).

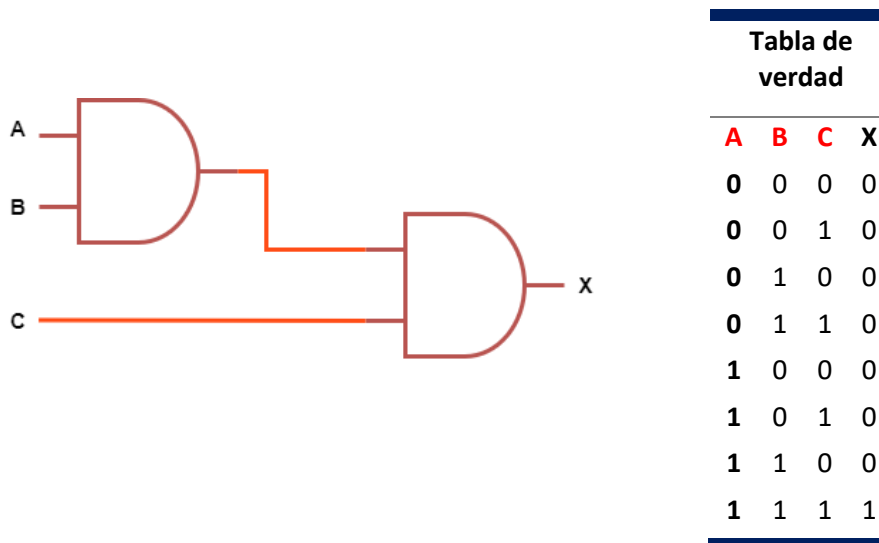


Figura 1.6. a) compuertas AND de 2 entradas en serie cascada, b) Tabla de verdad.





1.2 Compuerta O lógica o compuerta OR

La salida X de la compuerta OR será “1” cuando la entrada “A” o la entrada “B” estén en “1”. Expresándolo en otras palabras; en una compuerta OR, la salida será “1”, cuando en cualquiera de sus estradas haya un “1”. La compuerta OR se representa con la siguiente función booleana: $X = A+B$ ó $X = B+A$.

La representación de la compuerta “OR” de 2 entradas y su tabla de verdad se muestran en la figura 1.7.

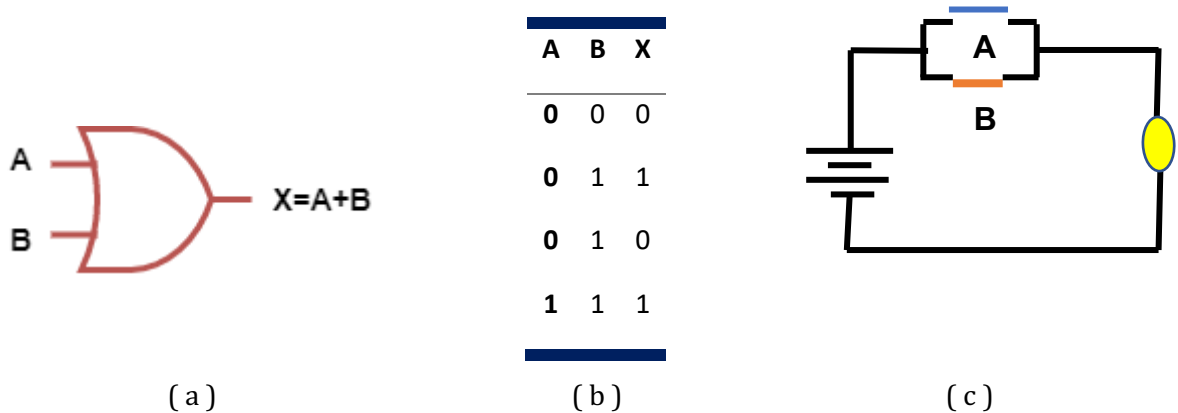


Figura 1.7. a) Símbolo compuerta OR, b) Tabla de verdad, y c) Representación en un circuito.

La compuerta OR también se puede implementar con interruptores como se muestra en la figura 1.7.c, en donde se puede ver que: cerrando el interruptor A “0” el interruptor B se encenderá la luz. “1” = cerrado, “0” = abierto, “1” = luz encendida

1.3. Práctica a realizar.



MATERIAL Y EQUIPO A UTILIZAR:

- Compuertas lógicas: AND, OR Y NOT
- Resistencias: 220 ohms
- Led's
- Cable de teléfono
- Dip switch
- Protoboard
- Pinzas





Universidad Autónoma del Estado de México

Metodología

1. Se pelaron los cables y se obtuvieron alambres más pequeños.
2. Los cables se cortaron un poco mas pequeños a la medida que eran necesarias.
3. Se insertaron bajo la especificación de tierra y de corriente para que esta fuera insertada en el voltímetro, ya que esta proporcionaría energía para la tabla (Protoboard)
4. Se insertaron los cables y se iniciaron conexiones de acuerdo a los nodos y al mismo tiempo eran insertados en el Protoboard para iniciar con el ensamblado de los led's y que estos se interconectarán al dip switch, este tenía la función del interruptor, la cual hacía que funcionaran los led's y de acuerdo a la compuerta lógica que era insertada, los led's encendían, esto lo podemos observar con las tablas de verdad que estas manejaban como lo es la AND, OR y NOT.
5. Al momento de hacer el cambio con alguna compuerta lógica los cables que eran insertados a los nodos estos tendían a aparecer a alguna tabla lógica y la única que era diferente era el NOT.

Sugerencias Didácticas

El profesor:

1. Establecerá los límites de la investigación.
2. Corroborará que los alumnos investiguen en la bibliografía proporcionada los conceptos relacionados con las secciones del laboratorio.
3. Planificar la forma de trabajo, tiempo de recorrido, número de integrantes por equipo y formas de evaluación.
4. Asesoría acerca del funcionamiento y operación del equipo y reglas de seguridad.
5. En debate se analizarán los resultados obtenidos, así como los problemas que se presenten antes y después de la práctica.





Universidad Autónoma del Estado de México

Conclusiones

Al concluir, el alumno será capaz reconocer cada uno de los instrumentos para desarrollar cada una de las prácticas posteriores que se presentan, conociendo los fundamentos básicos de las compuertas lógicas. Asimismo, pondrá en práctica el cuidado del equipo con el objetivo de mantener en buenas condiciones el laboratorio.





PRÁCTICA 2: Circuitos combinacionales

≡ Competencia a desarrollar

Comprender los conceptos básicos de los sistemas digitales de tipo secuencial.

INTRODUCCIÓN:

2.1 COMPUERTA LOGICA AND

La operación AND se ejecuta exactamente igual que la multiplicación ordinaria de unos y ceros. Una salida igual a 1 ocurre solo en el único caso donde todas las entradas son 1. La salida es cero en cualquier caso donde una o más entradas son 0.

El símbolo de la compuerta AND se muestra en la figura adjunta, en este caso una compuerta Y de 2 entradas. La salida de la compuerta AND es igual al producto Y de las entradas lógicas; es decir:

$$X = A \cdot B \quad (2.1)$$

En otras palabras, la compuerta AND es un circuito que opera en forma tal que su salida es ALTA, solo cuando todas sus entradas son ALTAS. En todos los otros casos se salida es BAJA.

2.2 COMPUERTA LOGICA OR

La operación lógica OR produce un resultado 1, cuando cualquiera de las variables de entrada es 1. La operación OR, genera un resultado de 0 solo cuando todas las variables de entrada son 0. En la adición, $1+1=1$, $1+1+1=1$, etc.

La compuerta OR es un circuito que tiene 2 o mas entradas y cuya salida es igual a la suma OR de entradas. La figura adjunta, muestra el símbolo correspondiente a una compuerta 0 de 2 entradas. Las entradas A y B son niveles de voltaje lógico y la salida (o resultado) X es un nivel de voltaje lógico, cuyo valor es el resultado de la adición OR de A y B; esto es:

$$X = A + B \quad (2.2)$$





Universidad Autónoma del Estado de México

En otras palabras, la compuerta OR opera de tal forma que su salida es ALTA si las entradas A, B o ambas están en un nivel lógico 1. La salida de la compuerta OR será BAJA, si las entradas están en un nivel lógico 0. A continuación se presenta la tabla de verdad de la compuerta OR:

2.3 COMPUERTA LOGICA NOT

La operación NOT difiere de las operaciones Y y O en que esta puede efectuarse con una sola variable de entrada. Por ejemplo, si la variable A se somete a la operación NOT, el resultado X se puede expresar como: $X=A'=/A$, donde el apóstrofe y la diagonal representan la operación NOT (también se usa una barra sobrepuesta). La operación NOT se conoce asimismo como inversor o complemento y estos términos se pueden usar como sinónimos.

El símbolo de la compuerta NOT se muestra en la figura adjunta, el cual se conoce comúnmente como INVERSOR (invertir en inglés). Este circuito siempre tiene una sola entrada y su nivel lógico de salida siempre es contrario al nivel lógico de esta entrada; es decir:

$$X = A' = /A \quad (2.3)$$



MATERIAL Y EQUIPO A UTILIZAR:

1. Protoboard
2. Compuertas lógicas NOT (74LS04), AND (74LS08), OR (74LS32)
3. Led's miniatura
4. Resistencia de 220 ohms
5. 1 Dip switch de 8 interruptores
6. Cable de red
7. Voltímetro
8. Fuente de voltaje



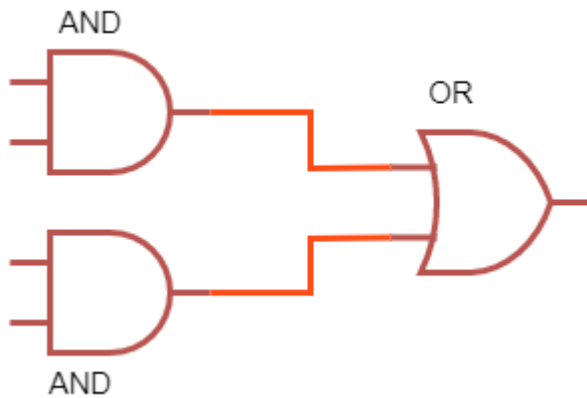


Universidad Autónoma del Estado de México

DESARROLLO Y METODOLOGÍA.

En este apartado se pretende situar el transistor en zona activa para la instalación de la máquina virtual.

Fig. 1.1.6 Se muestra el diagrama del cual nos vamos a guiar para hacer el circuito el cual demuestra el procedimiento que debe de llevar.



| A | B | X | Y | Z |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | * |
| 0 | 0 | 0 | 1 | * |
| 0 | 0 | 1 | 0 | * |
| 0 | 0 | 1 | 1 | * |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | * |
| 1 | 1 | 0 | 1 | * |
| 1 | 1 | 1 | 0 | * |
| 1 | 1 | 1 | 1 | * |

(a)

(b)

Figura 2.1. a) Circuito combinatorio representado por compuertas lógicas, b) tabla de verdad.





Universidad Autónoma del Estado de México

A continuación se muestran los mapas de Karnaugh para el ejercicio anterior.

MAPAS DE KARNAUGH

| | $\overline{Y_0} \overline{Y_1}$ | $\overline{Y_0} Y_1$ | $Y_0 Y_1$ | $Y_0 \overline{Y_1}^1$ | | $\overline{Y_0} \overline{Y_1}$ | $\overline{Y_0} Y_1$ | $Y_0 Y_1$ | $Y_0 \overline{Y_1}^1$ |
|---------------------------------|---------------------------------|----------------------|-----------|------------------------|---------------------------------|---------------------------------|----------------------|-----------|------------------------|
| $\overline{X_0} \overline{X_1}$ | 1 | 2 | 4 | 3 | $\overline{X_0} \overline{X_1}$ | 1 | 2 | 4 | 3 |
| | * | * | * | * | | 1 | 1 | 1 | 1 |
| $\overline{X_0} X_1$ | 5 | 6 | 8 | 7 | $\overline{X_0} X_1$ | 5 | 6 | 8 | 7 |
| | 1 | 1 | 1 | 1 | | 1 | 1 | 1 | 1 |
| $X_0 X_1$ | 13 | 14 | 16 | 15 | $X_0 X_1$ | 13 | 14 | 16 | 15 |
| | * | * | * | * | | 1 | 1 | 1 | 1] |
| $X_0 \overline{X_1}$ | 9 | 10 | 12 | 11 | $X_0 \overline{X_1}$ | 9 | 10 | 12 | 11 |
| | 1 | 1 | 1 | 1 | | 0 | 0 | 0 | 1 |
| | (a) | | | | | (b) | | | |

| | $\overline{Y_0} \overline{Y_1}$ | $\overline{Y_0} Y_1$ | $Y_0 Y_1$ | $Y_0 \overline{Y_1}^1$ |
|---------------------------------|---------------------------------|----------------------|-----------|------------------------|
| $\overline{X_0} \overline{X_1}$ | 1 | 2 | 4 | 3 |
| | * | * | * | * |
| $\overline{X_0} X_1$ | 5 | 6 | 8 | 7 |
| | 0 | 0 | 1 | 1 |
| $X_0 X_1$ | 13 | 14 | 16 | 15 |
| | * | * | * | * |
| $X_0 \overline{X_1}$ | 9 | 10 | 12 | 11 |
| | * | 1 | 1 | * |

(c)





Universidad Autónoma del Estado de México



Conclusiones

Al final el alumno conocerá operaciones lógicas básicas, tales como suma o producto. Además, conocerá el concepto de tabla de verdad. El alumno será evaluado con una serie de pregunta que permite conocer el conocimiento de los alumnos en el tema.





Universidad Autónoma del Estado de México

PRACTICA 3: Simulación de sistemas digitales secuenciales.

≡ Competencia a desarrollar

En esta práctica, se comprobará el funcionamiento de los registros de corrimiento, usándolos primero como contador digital tipo anillo y posteriormente, ver cómo funciona un registro de corrimiento de entrada y salida paralelo.



MATERIAL Y EQUIPO A UTILIZAR:

- Osciloscopio
- Multímetro.
- Fuente de voltaje.
- Protoboard
- 8 leds miniatura
- 8 resistencias de 2220 ohms
- 2 C.I J-K

Descripción de la práctica

Un contador de anillo es un dispositivo secuencial que tiene un flip-flop como registro de corrimiento por cada estado del contador. La salida serial del registro de corrimiento se retroalimenta al pin de entrada serial del registro. En el circuito resultante circula un patrón de bits alrededor del registro. Si inicializamos el registro de modo que tenga un único 1 lógico en su primer flip-flop y ceros lógicos en todos los demás, en el dispositivo circulará el 1 lógico en su primer flip-flop y ceros lógicos en todos los demás, en el dispositivo circulará el 1 lógico en su ciclo de flip-flops. Sea n el número de flip-flops y, por tanto, el número de estados del contador. Los flip-flops del registro de corrimiento se rotulan X_1, X_2, \dots, X_n . El funcionamiento del contador se inicia con un pulso en la línea de control Inicialice. Esto hace que X_1 sea alta y $X_2, X_3, \dots, X_{n-1}, X_n$ sean bajas. En este momento, sólo hay un 1 lógico en el flip-flop X_1 . En la siguiente transición negativa de la señal de entrada Clock, el 1 lógico se transfiere del flip-flop X_1 al flip-flop X_2 .





Universidad Autónoma del Estado de México

El proceso continúa hasta que el 1 lógico llega al final del registro de corrimiento, el flip-flop X_n . En la transición negativa del siguiente pulso de reloj, el 1 lógico se transfiere mediante la línea de retroalimentación al primer flip-flop en el registro de corrimiento, X_1 . Después, el proceso se repite. En otras palabras, el 1 lógico recorre un ciclo a través del registro de corrimiento cada n pulsos de reloj. Así, el contador de anillo tiene un único estado para cada flip-flop.

Metodología.

Alumno:

1. El alumno deberá conocer los conceptos de compuertas lógicas.
2. Asimismo, deberá conocer el concepto operaciones lógicas y tablas de verdad.

Desarrollo de la práctica.

Observar el estado en el que se ponen las salidas QA-QD con la primera bajada del reloj. Cambiar ahora la posición del interruptor a el punto 3 y subir la frecuencia del generador de señales a aproximadamente 1 KHz. Con la ayuda del OSCILOSCOPIO dibujar las formas de onda de entrada RELOJ y SHIFT, así como de la salida (QA, QB, QC y QD).

Los contadores habituales son secuenciales que cuentan en binario (hasta la capacidad que le permitan su número de bits de salida) los impulsos que reciben por su entrada. La señal de entrada suele proceder de algún tipo de reloj, aunque puede ser asíncrona.

Mediante diagramas de estados se pueden diseñar contadores tipo anillo donde cada estado es una cuenta.

El problema práctico que encontraremos con esta filosofía es que si el reloj es rápido (frecuencias del orden de MHz), las cantidades de pulsos que deberemos contar para generar relojes o señales lentas (con frecuencias en rangos de pocos Hz a kHz) son enormes, y los diagramas de estados resultantes son de miles o millones de estados. En esta situación, el código que se ha propuesto para sintetizar diagramas de estados, basado en dos procesos y determinadas instrucciones, no resulta cómodo. La figura 3.1 muestra un contador de anillo, el cual debe elaborarse con Flip-Flop tipo D.





Universidad Autónoma del Estado de México

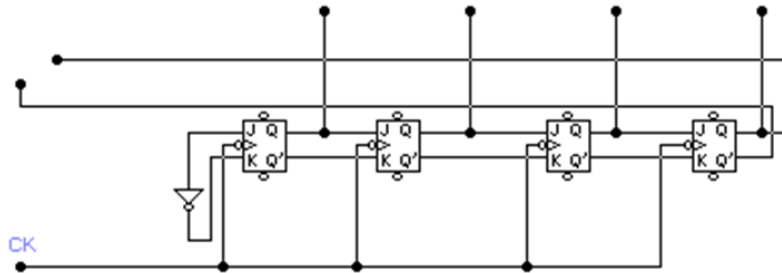


Figura 3.1. Esquema de un contador tipo anillo.

❓ Conclusiones

A partir de los conceptos fundamentales de Flip-Flops, se desarrolla un circuito contador, el cual debe ser primeramente simulado en Software Proteus®, y posteriormente elaborado en laboratorio. El funcionamiento de un contador de anillo, refiere a un sistema de tipo secuencial, el cual opera con una señal de reloj. El alumno podrá observar a través de un osciloscopio el efecto en el incremento en la frecuencia de la señal de reloj.





Universidad Autónoma del Estado de México

Practica 4: Simulación de Circuitos Integrados lineales.

≡ Competencia a desarrollar

Comprobar el funcionamiento y operaciones de las compuertas lógicas, así como sus estados lógicos.



MATERIAL Y EQUIPO A UTILIZAR:

- Circuito integrado NE555
- Resistencias: 10 K ohms
- Potenciómetro de 100 k ohms
- Microfaradios 0.01
- Microfaradios 0.001
- Led's
- Cable de teléfono
- Protoboard
- Pinzas

Metodología

1. Se realiza la comprobación de un circuito integrado de NE555 que es una señal de reloj.

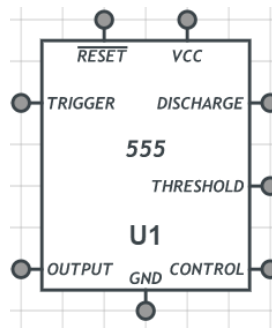


Figura 4.1 Circuito integrado (Fuente, <https://www.ti.com/lit/ds/symlink/ne555.pdf>)

2. Forme al cambio del potenciómetro la velocidad de la señal de reloj puede ser más lenta o más rápida.





Universidad Autónoma del Estado de México

3. Se enciende, pero con un proceso lento ya que no se puede visualizar de forma detallada.

MARCO TEÓRICO.

GND (normalmente la 1): es el polo negativo de la alimentación, generalmente tierra.

Disparo (normalmente la 2): Es en esta patilla, donde se establece el inicio del tiempo de retardo, si el 555 es configurado como monoestable. Este proceso de disparo ocurre cuando este pin va por debajo del nivel de $1/3$ del voltaje de alimentación. Este pulso debe ser de corta duración, pues si se mantiene bajo por mucho tiempo la salida se quedará en alto hasta que la entrada de disparo pase a alto otra vez.

Salida (normalmente la 3): Aquí veremos el resultado de la operación del temporizador, ya sea que esté conectado como monoestable, astable u otro. Cuando la salida es alta, el voltaje será el voltaje de alimentación (V_{cc}) menos 1.7 Voltios. Esta salida se puede obligar a estar en casi 0 voltios con la ayuda de la patilla de reset (normalmente la 4).

Reset (normalmente la 4): Si se pone a un nivel por debajo de 0.7 Voltios, pone la patilla de salida a nivel bajo. Si por algún motivo esta patilla no se utiliza hay que conectarla a V_{cc} para evitar que el 555 se "resete".

Control de voltaje (normalmente la 5): Cuando el temporizador se utiliza en el modo de controlador de voltaje, el voltaje en esta patilla puede variar casi desde V_{cc} (en la práctica como $V_{cc} - 1$ voltio) hasta casi 0 V (aprox. 2 Voltios). Así es posible modificar los tiempos en que la salida está en alto o en bajo independiente del diseño (establecido por los resistores y condensadores conectados externamente al 555). El voltaje aplicado a la patilla de control de voltaje puede variar entre un 45 y un 90 % de V_{cc} en la configuración monoestable. Cuando se utiliza la configuración astable, el voltaje puede variar desde 1.7 voltios hasta V_{cc} . Modificando el voltaje en esta patilla en la configuración astable causará la frecuencia original del astable sea modulada en frecuencia (FM). Si esta patilla no se utiliza, se recomienda ponerle un condensador de $0.01\mu F$ para evitar las interferencias.





Universidad Autónoma del Estado de México

Umbral (normalmente la 6): Es una entrada a un comparador interno que tiene el 555 y se utiliza para poner la salida a nivel bajo.

Descarga (normalmente la 7): Utilizado para descargar con efectividad el condensador externo utilizado por el temporizador para su funcionamiento.

V+ (normalmente la 8): También llamado Vcc, alimentación, es el pin donde se conecta el voltaje de alimentación que va de 4.5 voltios hasta 18 voltios (máximo). Hay versiones militares de este integrado que llegan hasta 18 Voltios.

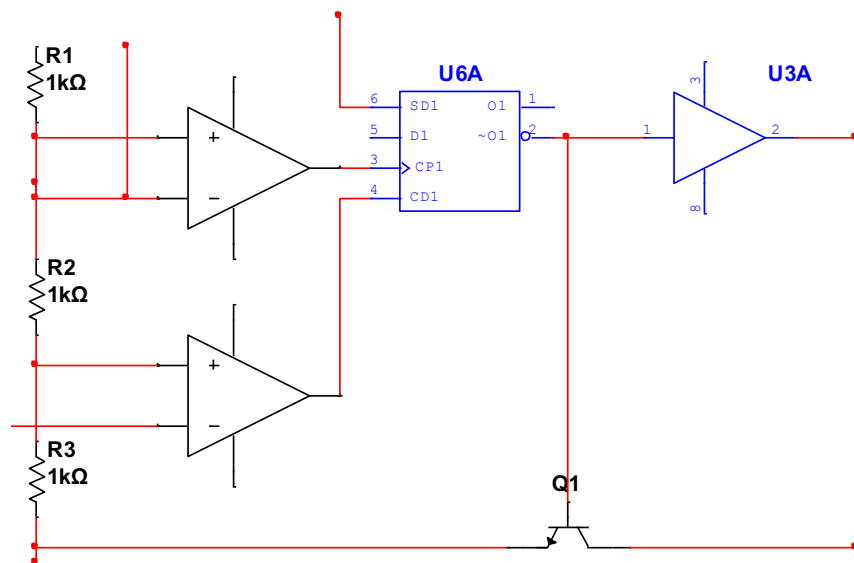


Figura 4.2. Circuito electrónico interno que compone a un NE555. Simulador Proteus® versión de prueba.

Conclusiones

El alumno aprenderá los principios básicos para la puesta en operación de un oscilador NE555 para su uso en los sistemas secuenciales, como una alternativa para la sincronización de los diseños realizados.





Universidad Autónoma del Estado de México

PRÁCTICA 5: Decodificador BCD a 7 segmentos

≡ Competencia a desarrollar

Comprender los conceptos básicos de los sistemas digitales de tipo secuencial.



INTRODUCCIÓN:

En la presente practica se desarrolla un decodificador BCD a 7 segmentos haciendo uso de compuertas lógicas y reducción por mapas de Karnaugh y algebra booleana. Este decodificador puede tener salidas normales o salidas negadas, donde el display de 7 segmentos que puede ser catado común o bien ánodo común en la que desea visualizar los números del 0 al 9 en un display de 7 segmentos. Tiene como función detectar la presencia de una determinada combinación de bits en los cuales indica una salida determinada basándose en datos de entrada, característicos en la que su función operacional se basa en la instrucción a sus entradas de un numero en código binario correspondiente a su equivalente en decimal para mostrar en los siete pines de salidas.

5.1 Decodificador

El decodificador es un elemento digital el cual funciona a base de estados lógicos, con los cuales indica salida determinada basándose en un dato de entrada característico, la función operacional se basa en la introducción a sus entradas de un numero en código binario correspondiente a su equivalente en decimal para que así se puedan mostrar en los siete pines de salida que están establecidos para el integrado, una serie de estados lógicos los cuales están diseñados para que se conecten a un elemento alfanumérico en el que se visualizara el numero introducido en las entradas del decodificador. El decodificador está formado internamente por compuertas lógicas y sus conexiones internas son un sistema predefinido por el diseñador para que su función operacional sea un acople perfecto con el display.

5.2. Display





Universidad Autónoma del Estado de México

Es un dispositivo alfanumérico que está formado por diodos de luz, en la que están posicionados de forma en la que forme 8 números, a cada uno de ellos se les denomina segmentos, en la cual se encuentran encendidos algunos y los otros apagados para formar los diferentes números por medios de las combinaciones que existen entre ellos. Cada segmento esta designado con una letra, el display se encuentra en una representación de encapsulado en la que se le asigna la letra correspondiente del segmento. El display cuenta con 8 leds que se encuentra colocados en forma de dígitos con punto decimal, cada led tiene dos extremos ánodos y cátodo. Como en total son 8 led, debería tener 16 extremos es decir 8 catados y 8 ánodos, sin embargo, el encapsulamiento solo cuenta con 10 esto sucede para reducir el tamaño del encapsulamiento.



MATERIAL Y EQUIPO A UTILIZAR:

El material a utilizar para esta practica fue el simulador Proteus medio por donde se realizo el circuito.



DESARROLLO Y METODOLOGÍA.

1. Dentro de esta práctica se desarrolló un circuito con la finalidad de simular un decodificador BCD a 7 segmentos.
2. Se realizan los mapas de karnaugh, del (a) al (g)
3. A continuación se presenta en la figura 5.1 el circuito realizado en Proteus® versión de prueba.

| | DC | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|
| BA | | | | | |
| 00 | 1 | 0 | 1 | 1 | |
| 01 | | 0 | 1 | 1 | 0 |
| 11 | X | X | X | X | |
| 10 | 1 | 1 | X | X | |

(a) Mapa de Karnaugh para "a"

| | DC | 00 | 01 | 11 | 10 |
|----|----|----|----|----|----|
| BA | | | | | |
| 00 | 1 | 1 | 1 | 1 | |
| 01 | 1 | 0 | 1 | 0 | |
| 11 | X | X | X | X | |
| 10 | 1 | 1 | X | X | |

(b) Mapa de Karnaugh para "b"





Universidad Autónoma del Estado de México

| DC \ BA | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 1 | 1 | 1 | 0 |
| 01 | 1 | 1 | 1 | 1 |
| 11 | X | X | X | X |
| 10 | 1 | 1 | X | X |

c) Mapa de Karnaugh para "c"

| DC \ BA | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 1 | 0 | 1 | 1 |
| 01 | 0 | 1 | 0 | 1 |
| 11 | X | X | X | X |
| 10 | 1 | 0 | X | X |

d) Mapa de Karnaugh para "d"

| DC \ BA | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 1 | 0 | 0 | 1 |
| 01 | 0 | 0 | 0 | 1 |
| 11 | X | X | X | X |
| 10 | 1 | 0 | X | X |

e) Mapa de Karnaugh para "e"

| DC \ BA | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 1 | 0 | 0 | 0 |
| 01 | 1 | 1 | 0 | 1 |
| 11 | X | X | X | X |
| 10 | 1 | 1 | X | X |

f) Mapa de Karnaugh para "f"

| DC \ BA | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | 0 | 1 | 1 |
| 01 | 1 | 1 | 0 | 1 |
| 11 | X | X | X | X |
| 10 | 1 | 1 | X | X |

g) Mapa de Karnaugh para "g"





Universidad Autónoma del Estado de México

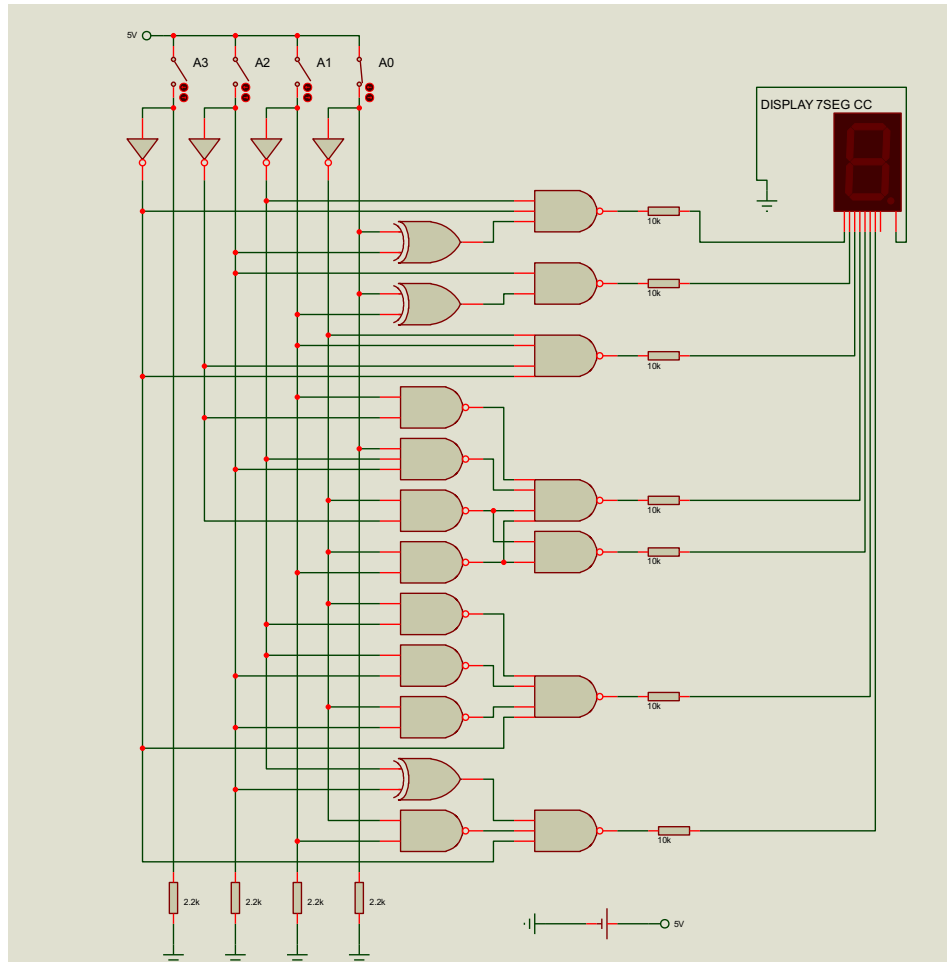


Figura 5.1 Circuito de 7 segmentos, elaborado en Proteus®

Resultados obtenidos

En la figura 5.2 se muestra una imagen en donde se muestra el funcionamiento del circuito mostrado al principio en donde mostrara un numero del 0 al 9 al azar





Universidad Autónoma del Estado de México

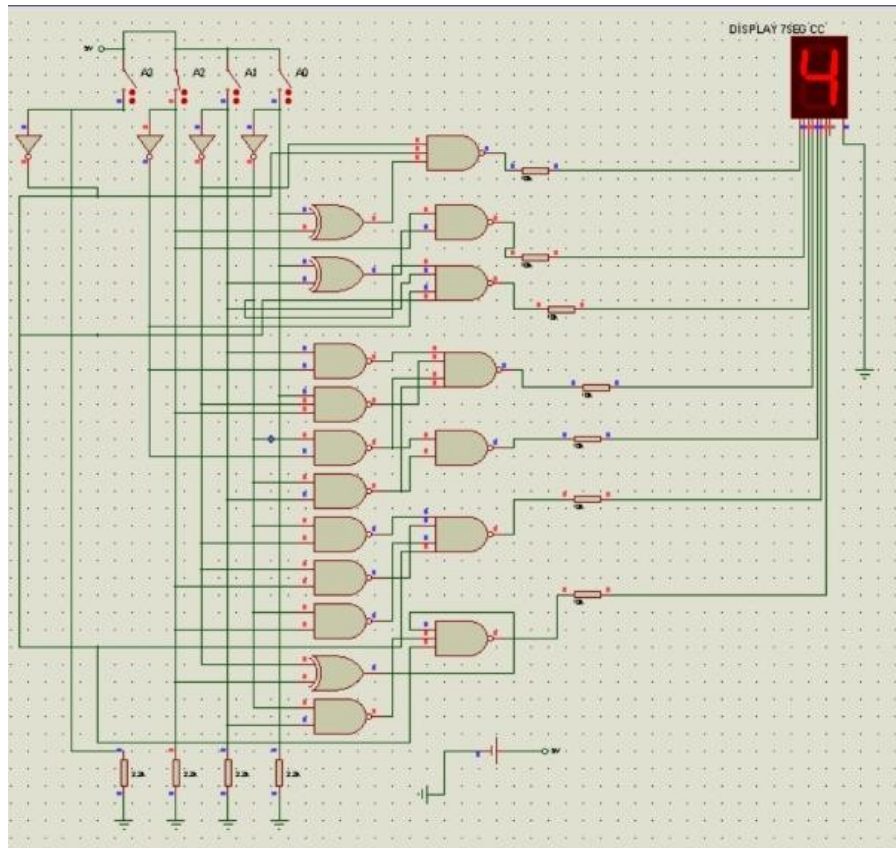


Figura 5.2. Prueba de funcionamiento en software Proteus®.

- ① El alumno pondrá en práctica el diseño y reducción de circuitos combinatoriales por medio de Mapas de Karnaugh. De esta manera, será capaz de hacer uso de esta herramienta de manera eficiente para el diseño de circuitos combinatoriales.





Universidad Autónoma del Estado de México

PRÁCTICA 6: Sumador y Restador Completo

≡ Competencia a desarrollar

Comprender los conceptos básicos de los sistemas digitales de tipo secuencial.



INTRODUCCIÓN:

En la presente practica se simula por medio del software “Proteus” dos circuitos lógicos para un sumador y un restador de dos números de dos bits, para ello se diseñan con la finalidad de hacer uso de tablas de verdad, mapas K con suma y resta en la lógica combinacional, y en base a ello se crean los circuitos a simular. Es importante mencionar que para ello se deben conocer los conceptos que se muestran a continuación.

6.1. Sumador

El sumador completo supone un circuito un poco más complejo a diferencia de un semi—sumador, nos permite incluir en la suma de dos bits y acarreo de una suma anterior a través de una entrada que se llama C_{in} con lo que se puede realizar la suma tendremos tres variables de entrada, a las que llamaremos A , B y la propia C_{in} y dos de salida: S , la suma y C_{cout} , el acarreo.

6.2. Semi-sumador

un semi-sumador admite dos dígitos binarios en sus entradas y generar dos dígitos binarios en sus salidas: un bit de suma y un bit de acarreo

La lógica del semi-sumador consiste en que, a partir de las expresiones correspondientes a la suma y al acarreo de salida se pueden obtener como funciones de las entradas.

6.3. Sumador completo

Es un circuito combinacional que realiza la suma de dos bits y un acarreo de entrada, obteniendo en su salida la suma y un acarreo de salida. La diferencia principal entre un sumador completo y un semi-sumador es que el sumador completo acepta un acarreo de entrada.





Universidad Autónoma del Estado de México

La lógica del sumador completo consiste en sumar dos bits de entrada y un acarreo de entrada. Del semi-sumador en la cual se sabe que la suma de los bits de entrada A Y B es la operación *OR* exclusiva de esas dos variables $A \oplus B$ para sumar el acarreo de entrada C_{in} a los bits de entrada, hay que aplicar de nuevo las operaciones *OR* exclusiva, obteniéndose la siguiente ecuación para la salida de suma del sumador completo.

6.4. Restador completo

Es un circuito combinatorial que lleva a cabo una sustracción entre 2 bits, Este tipo de circuitos tiene 3 entradas y 2 salidas, en donde, las 3 entradas x, y, z denotan el minuendo, sustraendo y a la toma previa respectivamente, las 2 salidas D, B representan la diferencia y la salida tomada, respectivamente.

Los ocho renglones bajo las variables de entrada designan todas las combinaciones posibles de 1 y 0 que pueden tomarlas variables binarias.

Los 1 y 0 para las variables de salida están determinados por la sustracción de $x - y - z$. Las combinaciones que tienen salida de toma $z = 0$ se reducen a las mismas cuatro condiciones del medio sumador.



MATERIAL Y EQUIPO A UTILIZAR:

El material a utilizar para esta practica fue el simulador Proteus medio por donde se realizo el circuito



DESARROLLO Y METODOLOGÍA.

Dentro de esta práctica se desarrollan dos circuitos lógicos con la finalidad de crear un sumador y un restador de dos números de dos bits, para ello se hace uso de la herramienta de simulación “Proteus”. Como se desarrolló, se realiza el diseño (tablas de verdad, mapas K, obtención de función) y con ello la creación del circuito, esto tanto para el sumador como para el restador.





Universidad Autónoma del Estado de México

Es importante mencionar que; para crear un sumador de dos bits debemos considerar que está compuesto por dos sumadores completos, para ello el diseño se emplea para un sumador completo, ya para crear el circuito se contemplan dos sumadores completos.

Para iniciar el diseño, se crea la tabla de verdad; para el desarrollo de esta, se establecen los dos números de entrada como "X y "Y", y se agregaron todas las combinaciones de estas, considerando también un acarreo de entrada (Cin), posterior a ello también se consideran las salidas (s) y un acarreo de salida (Cout), considerando la suma de "X+Y", como se muestra en la siguiente tabla.

Tabla 6.1. Tabla de verdad para sumador completo

| Y | X | Cin | S | Cout |
|---|---|-----|---|------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Posterior a ello y dadas las salidas de la tabla de verdad se desarrollan dos mapas de Karnaugh, el primero para las salidas en general o el resultado de la suma y el otro mapa para el acarreo de salida, a continuación, se muestran estos mapas K con su respectiva función:

| Y/Cin | 00 | 01 | 10 | 11 |
|-------|----|----|----|----|
| X | | | | |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

Figura 6.1 Mapa de Karnaugh para salidas (S) (resultado de la suma)





Universidad Autónoma del Estado de México

Una vez analizado el mapa de Karnaugh y considerados los grupos generados, se obtiene la siguiente función.

$$s = X \oplus Y \oplus Cin \tag{6.1}$$

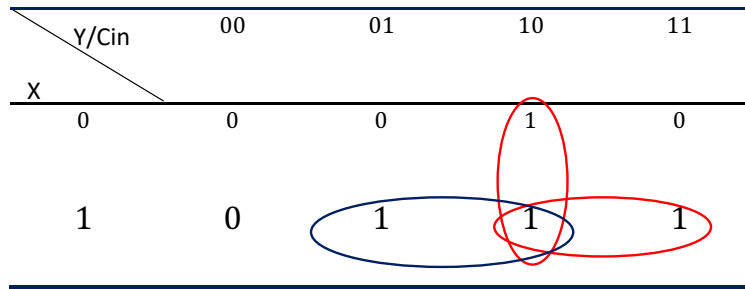


Figura 6.2 Mapa de Karnaugh para el Acarreo de salida Cout.

Una vez analizado el mapa de Karnaugh y considerados los grupos generados, se obtiene la siguiente función.

$$Cout = YCin + XCin + XY \tag{6.2}$$

Dadas las funciones, se puede plantear el circuito aplicando dos sumadores completos, como se muestra en la siguiente imagen, donde también se observa los componentes que se usaron.

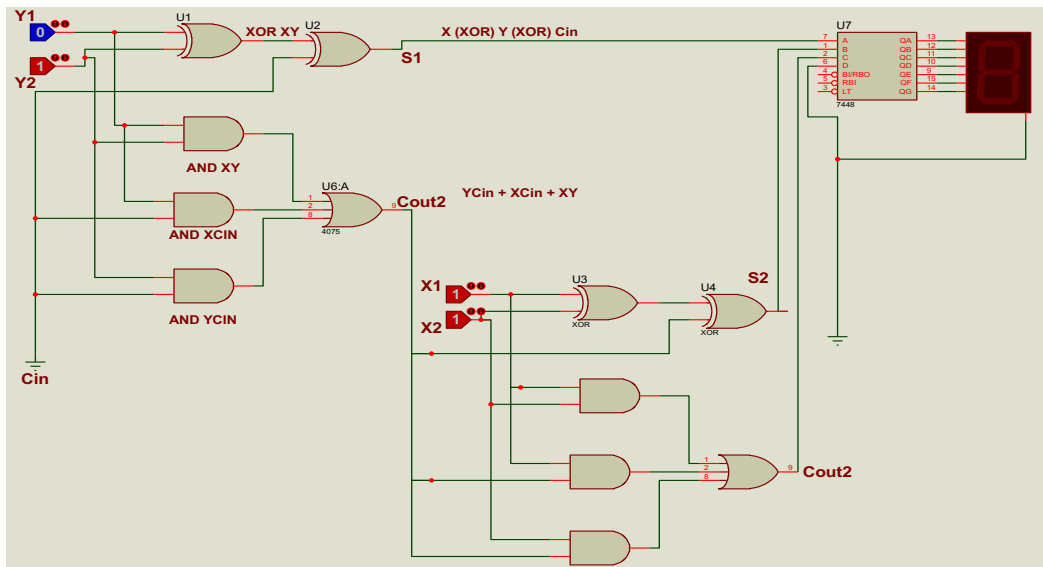


Figura 6.3.- Construcción del circuito, elaborado en Proteus®





Universidad Autónoma del Estado de México

6.5. Restador

Al igual que el sumador es importante mencionar que; para crear un restador de dos bits debemos considerar que está compuesto por dos restadores completos, para ello el diseño se emplea para un restador completo, ya para crear el circuito se contemplan dos restadores completos.

Para iniciar el diseño, se crea la tabla de verdad; para el desarrollo de esta, se establecen los dos números de entrada como "X y "Y", y se agregaron todas las combinaciones de estas, considerando también un acarreo de entrada (Cin), posterior a ello también se consideran las salidas (s) y un acarreo de salida (Cout), como se muestra en la siguiente tabla.

Tabla 6.2. Salidas y acarreo de salidas

| Y | X | Cin | S | Cout |
|---|---|-----|---|------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Posterior a ello y dadas las salidas de la tabla de verdad se desarrollan dos mapas de Karnaugh, el primero para las salidas en general o el resultado de la resta y el otro mapa para el acarreo de salida, a continuación, se muestran estos mapas K con su respectiva función:

| | | | | | |
|---|-------|----|----|----|----|
| | Y/Cin | 00 | 01 | 10 | 11 |
| X | 0 | 0 | 1 | 0 | 1 |
| | 1 | 1 | 0 | 1 | 0 |

Figura 6.4. Mapa de Karnaugh para salidas (S) (resultado de la Resta





Universidad Autónoma del Estado de México

Una vez analizado el mapa de Karnaugh y considerados los grupos generados, se obtiene la siguiente función.

$$s = X \oplus Y \oplus Cin \tag{6.3}$$

| Y/Cin | 00 | 01 | 10 | 11 |
|-------|----|----|----|----|
| X | | | | |
| 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |

Figura 6.5.-Mapa de Karnaugh para el Acarreo de salida Cout

Una vez analizado el mapa de Karnaugh y considerados los grupos generados, se obtiene la siguiente función.

$$Cout = \overline{X}Y + \overline{X}(Cin) + YCin \tag{6.4}$$

Dadas las funciones, se puede plantear el circuito aplicando dos restadores completos, como se muestra en la siguiente imagen, donde también se observa los componentes que se usaron.

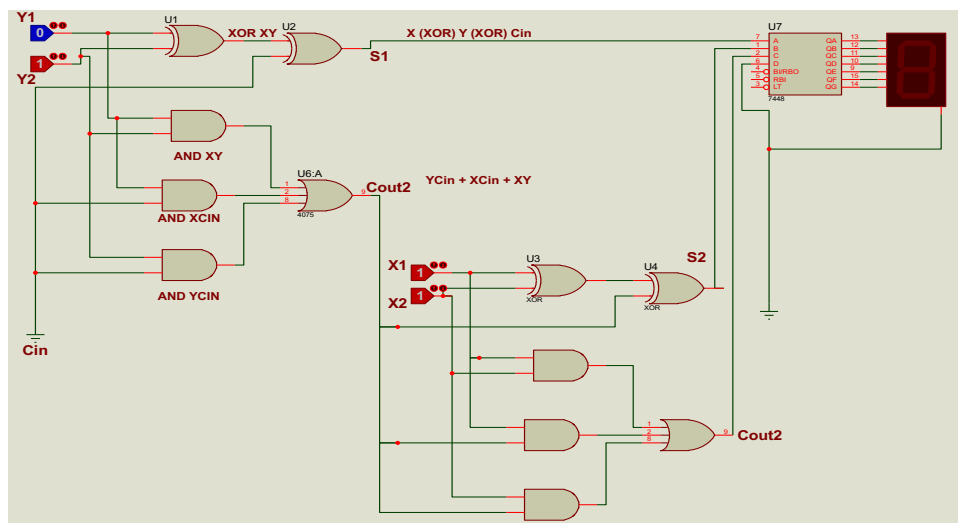


Figura 6.6 Circuito de Sumador y restador completo, elaborado en Proteus®

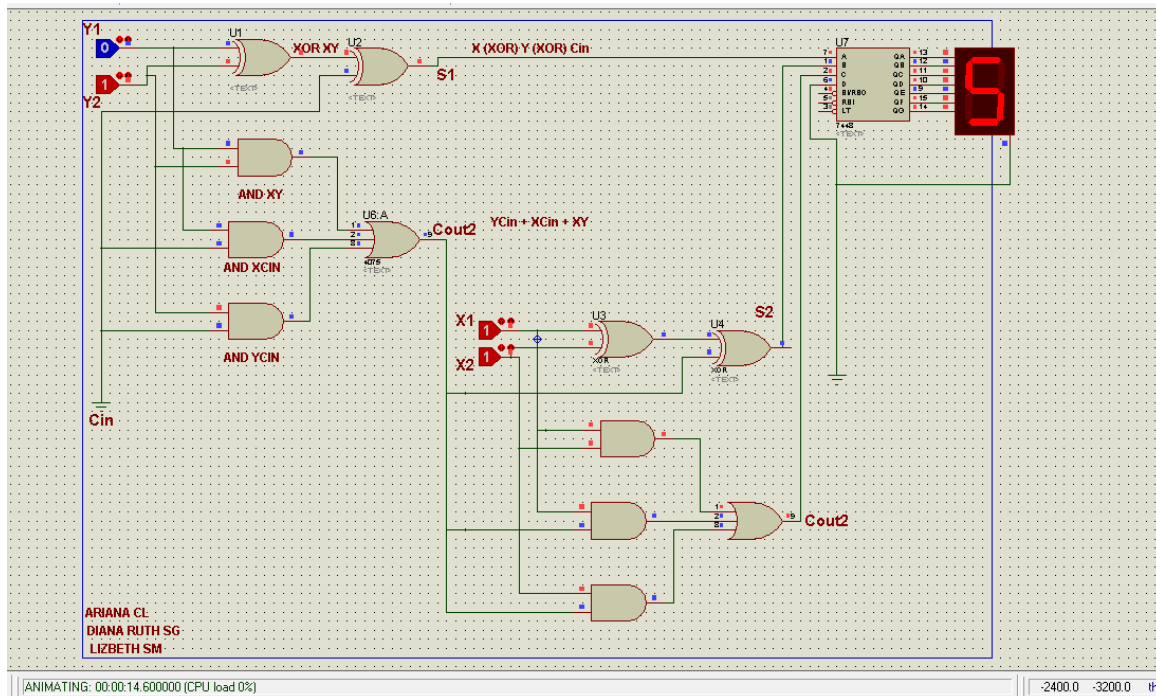




Resultados obtenidos

1. Sumador

Para comprobar el funcionamiento de la simulación, se ejecutó una simple suma; $2+3$ a lo que el resultado debe ser 5, para ello en la siguiente imagen se puede observar que ingresan los datos 2 y 3 en binario ($2_{b10}=10_{b2}$; $3_{b10}=11_{b2}$) en los estados lógicos, y en el *display* se observa el número 5 de manera correcta.



Figuras 6.7.- Suma correcta elaborado en Proteus®

2. Restador

Para comprobar el funcionamiento de la simulación, se ejecutó una simple resta; $3-2$ a lo que el resultado debe ser 1, para ello en la siguiente imagen se puede observar que ingresan los datos 3 y 1 en binario ($2_{b10}=10_{b2}$; $3_{b10}=11_{b2}$) en los estados lógicos, y en el *display* se observa el número 1 de manera correcta.





Universidad Autónoma del Estado de México

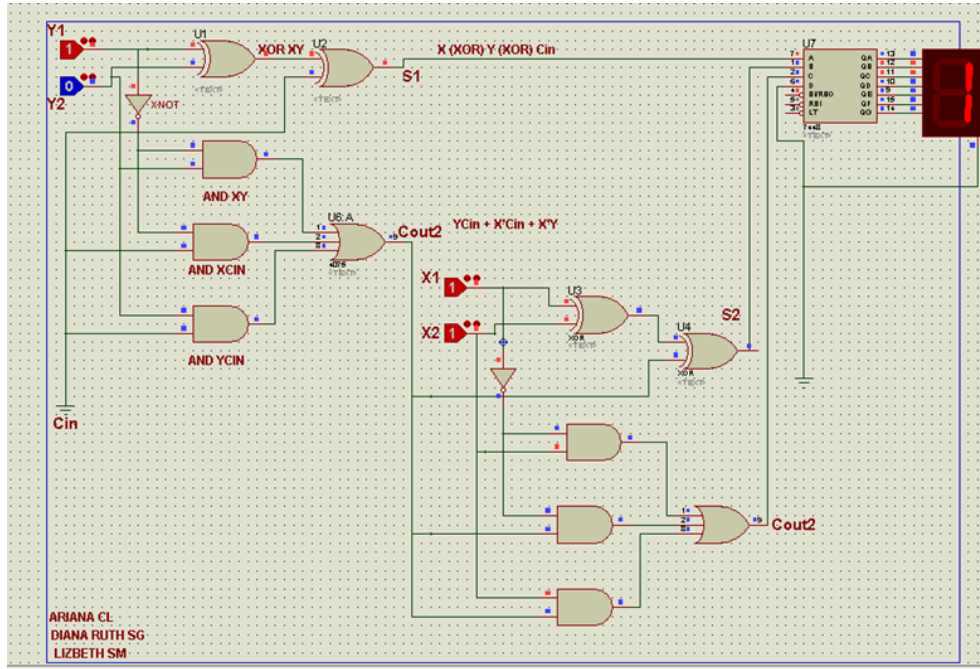


Figura 6.8- Resta correcta de los binario, elaborado en Proteus®

❓ Conclusion.

El alumno llevará a cabo las operaciones binarias de sumador y restador, para su posterior uso o aplicación en algún tipo de aritmética lógica.





Universidad Autónoma del Estado de México

PRÁCTICA 7: Simulación de multiplexor de 2 entrada cada una de 4 bit

≡ Competencia a desarrollar

Realizar el diseño del circuito de multiplexor de dos estradas de 4 bits, haciendo uso de lógica combinacional, aplicando los conceptos vistos en clase.

🔧 INTRODUCCIÓN:

En la presente práctica se simula un multiplexor de dos entradas en la que cada entrada deberá estar compuesta por cuatro bits y una salida de 4 bits, para ellos se hace uso de las tablas de verdad para la elaboración de la practica por lo que es importante mencionar algunos conceptos que se muestran a continuación:

Multiplexor

Un multiplexor es un selector de datos equivalente a un conmutador de m entradas y una salida, por lo que también recibe el nombre de selector de datos o conmutador electrónico. La selección de la entrada se controla mediante unas entradas de selección o control. Cuando sólo tenemos una entrada de control (2 entradas), también se le llama entrada de habilitación.

La multiplexación es el término genérico utilizado para describir la operación de enviar una o más señales analógicas o digitales a través de una línea de transmisión común en diferentes momentos o velocidades y, como tal, el dispositivo que utilizamos para hacer precisamente eso se llama un multiplexor.

La selección de cada línea de entrada en un multiplexor es controlada por un conjunto adicional de entradas de llamadas de control de líneas y de acuerdo con la condición binaria de estas entradas de control, ya sea alto o LOW la entrada de datos apropiado está conectado directamente a la salida. Normalmente, un multiplexor tiene un número par de líneas de entrada de datos $2N$ y un número de entradas de control que se corresponden con el número de entradas de datos. Permiten seleccionar una de las entradas para realizar la transmisión de datos desde dicha entrada a la salida, que es única.





Universidad Autónoma del Estado de México



MATERIAL Y EQUIPO A UTILIZAR:

El material a utilizar para esta practica fue el simulador proteus medio por donde se realizo el circuito



DESARROLLO Y METODOLOGÍA.

entro de esta práctica se desarrollan un circuito lógico con la finalidad de crear un multiplexor de dos entradas de 4 bits, para ello se hace uso de la herramienta de simulación “Proteus”. Como se desarrolló, se realiza el diseño y con ello la creación del circuito. Dada la siguiente tabla de verdad del multiplexor a crear.

Tabla 7.1.- tabla de verdad del multiplexor

| X | B | Y | S |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |





Universidad Autónoma del Estado de México

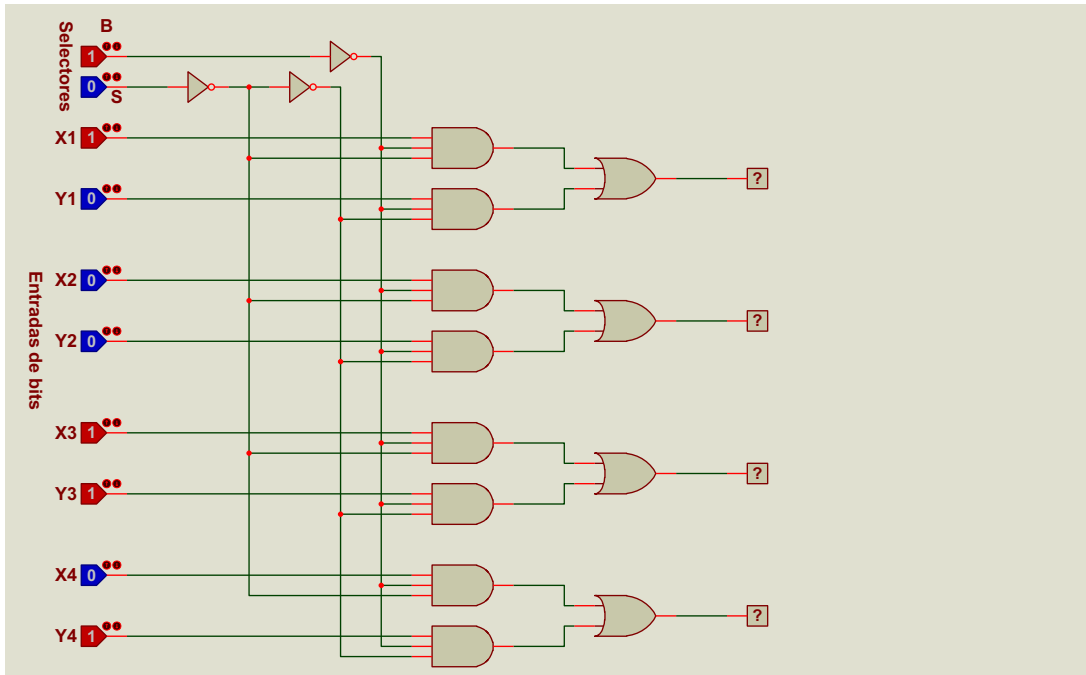


Figura 7.1.- Circuito obtenido por medio del simulador, elaborado en Proteus®

Resultados obtenidos

Una vez diseñado y creados el circuito, se simulan o ejecutan, como se observa en las siguientes figuras.





Universidad Autónoma del Estado de México

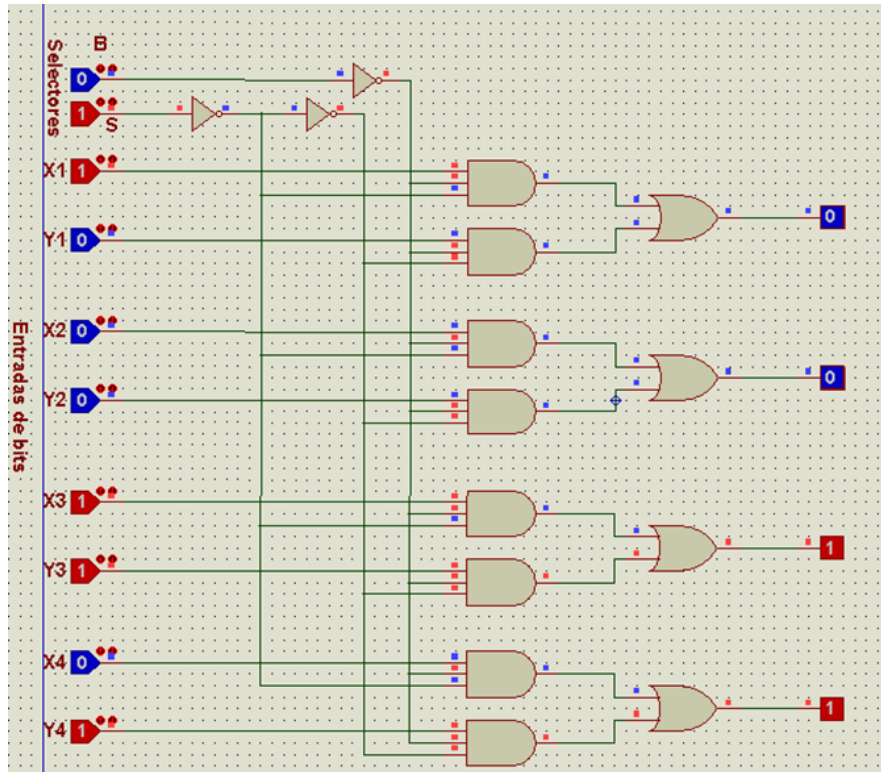


Figura 7.2.- Circuito lógico combinacional para su prueba, prueba 1, elaborado en Proteus®





Universidad Autónoma del Estado de México

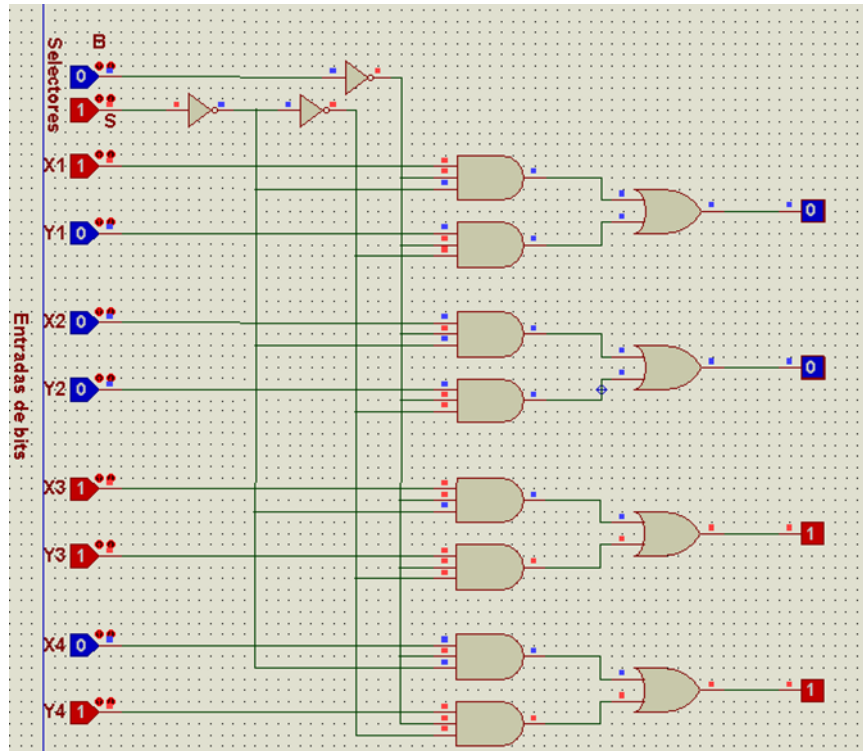


Figura 7.3 Circuito lógico combinacional para su prueba, prueba 2, elaborado en Proteus®

❓ Conclusion

El alumno comprobará de manera práctica y en simulador el funcionamiento de un circuito multiplexor de dos entradas.





Universidad Autónoma del Estado de México

PRÁCTICA 8: Comparador de magnitud

≡ Competencia a desarrollar

Realizar el diseño del circuito de dos números binarios de 2 bits cada uno, haciendo uso de lógica combinacional, aplicando los conceptos vistos en clase.



INTRODUCCIÓN:

La presente practica desarrolla el diseño de un circuito para simular un comparador de magnitud de dos números binarios de 2 bits cada uno, con la finalidad de conocer los componentes de un comparador haciendo uso de lógica secuencial y combinacional, a partir de la tabla de verdad, diseñando los mapas K para obtener funciones y emplearlas para la construcción del circuito en el simulador “Proteus” donde se observa de manera correcta el funcionamiento de un comparador de magnitud de dos entradas de dos bits. Comparadores Un circuito digital comparador realiza la comparación de dos palabras A y de B de N bits tomadas como un numero entero sin signo e indica si son iguales a si una es mayor que otra en tres salidas $A = B$, $A > B$ y $A < B$.

Comparador de magnitud Es un circuito combinacional que compara dos números A y B y determina sus magnitudes relativas. La salida de la comparación se especifica por tres variables binarias que indica si $A > B$, $A = B$ o $A < B$. para determinar si A es mayor o menor que B, se inspeccionan las magnitudes relativas de pares de dígitos significativos principiando desde la posición más significativa. Si los dos dígitos son iguales, el par de dígitos de la siguiente posición significativa más baja se comparan. Esta comparación continúa hasta que se alcanza un par de dígitos desiguales. Si el dígito correspondiente de A es 0 y B es 1, se tiene que $A < B$.

Comparador de dos bits Las comparaciones de dos bits se pueden realizar por medio de una puerta XOR o una XNOR. Las salidas del circuito es 1 si sus dos bits de entrada son diferentes y 0 si son iguales. Dentro de esta práctica se emplea este concepto considerando dos entradas de dos bits.





MATERIAL Y EQUIPO A UTILIZAR:

El material a utilizar para esta practica fue el simulador Proteus medio por donde se realizo el circuito.



DESARROLLO Y METODOLOGÍA.

Dentro de esta práctica se diseña, desarrolla y simula un circuito lógico con la finalidad de crear un comparador de magnitud de dos números de dos bits, para ello se hace uso de la herramienta de simulación “Proteus” para la simulación y como diseño se crea la tabla de verdad y los mapas de Karnaugh para obtener las funciones y así construir el circuito como se muestra a continuación. Se muestra la tabla de verdad correspondiente al comparador de dos bits (Tala 8.1), dado esto se tienen 4 entradas (A0, A1, B0 Y B1) y se consideran 3 salidas (M, N, P), tomando un valor lógico 1 según las siguientes características:a) A=B para M, b) A>B para N, c) A<B para P

Tabla 8.1.- Tabla de verdad del comparador de magnitud

| A1 | A0 | B1 | B0 | P | N | M |
|----|----|----|----|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 |





Mapas de Karnaugh para obtener funciones

Posterior a la tabla de verdad, y dadas las salidas se realizan los mapas de Karnaugh para obtener las respectivas funciones con las que se creara el circuito.

Para la salida P (A<B)

Dadas las salidas para el segmento "P" en tabla de verdad, se desarrolla el mapa de Karnaugh a 4 variables, así mismo se establecen las agrupaciones según las salidas a considerar como se muestra a continuación:

| A1/A0 | 00 | 01 | 11 | 10 |
|-------|----|----|----|----|
| B1/B0 | | | | |
| 00 | 0 | 1 | 1 | 1 |
| 01 | 0 | 0 | 1 | 1 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 1 | 0 |

Figura 8.1.- Mapa de karnaugh, P (A<B)

A partir del anterior mapa se establece la función y esta es aún más reducida haciendo uso de algebra de Boole, se observa a continuación.

$$P = A1B1' + (A0A1)B0'B1 + A0B0'B1'$$

Para la salida N(A>B)

Dadas las salidas para el segmento "N" en tabla de verdad, se desarrolla el mapa de Karnaugh a 4 variables, así mismo se establecen las agrupaciones según las salidas a considerar como se muestra a continuación:





Universidad Autónoma del Estado de México

| A1/A0 | 00 | 01 | 11 | 10 |
|-------|----|----|----|----|
| B1/B0 | | | | |
| 00 | 0 | 0 | 0 | 0 |
| 01 | 1 | 0 | 0 | 0 |
| 11 | 1 | 1 | 0 | 1 |
| 10 | 1 | 1 | 0 | 0 |

Figura 8.2.- Mapa de karnaugh, salida N(A>B)

A partir del anterior mapa se establece la función y esta es aún más reducida haciendo uso de algebra de Boole, se observaa continuación.

$$N = A0'B1 + (B0A0')A1' + A1A0'B1B0$$

Para la salida M (A=B). Dadas las salidas para el segmento "M" en tabla de verdad, se desarrolla el mapa de Karnaugh a 4 variables, así mismo se establecen las agrupaciones según las salidas a considerar como se muestra a continuación:

| A1/A0 | 00 | 01 | 11 | 10 |
|-------|----|----|----|----|
| B1/B0 | | | | |
| 00 | 1 | 0 | 0 | 0 |
| 01 | 0 | 1 | 0 | 0 |
| 11 | 0 | 0 | 1 | 0 |
| 10 | 0 | 0 | 0 | 1 |

Figura 8.3.- Mapa de karnaugh, salida M (A=B)





Universidad Autónoma del Estado de México

A partir del anterior mapa se establece la función y esta es aún más reducida haciendo uso de algebra de Boole, se observaa continuación.

$$M = (A0 \oplus B0)'(A1 \oplus B1)'$$

8.3. Construcción de circuito

Una vez obtenidas las funciones, con estas se construye analizando que contiene las 4 entradas (A0, A1, B0 Y B1) y lassalidas correspondientes:

$$A=B \text{ para } M = (A0 \oplus B0)'(A1 \oplus B1)'$$

$$A > B \text{ para } N = A1B1' + (B0B1)'A0 + A1A0B0'$$

$$A < B \text{ para } P = A1'B1 + (A0A1)'B0 + A0'B0B1$$

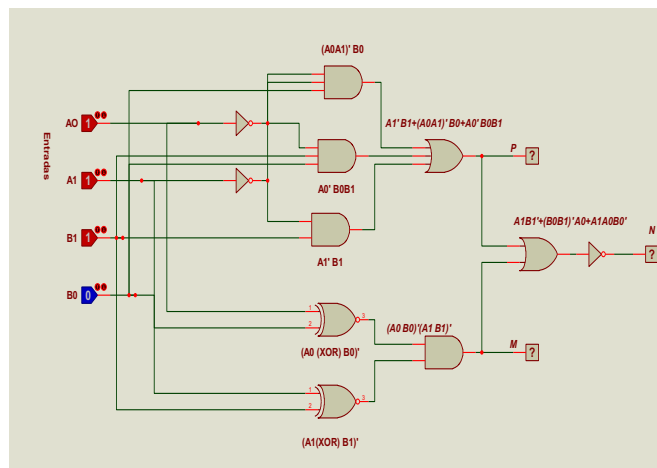


Figura 8.4.- Circuito de comparador de magnitud, elaborado en Proteus®

Resultados obtenidos

Una vez diseñado y creado el circuito, se simulan o ejecutan, como se observa en las siguientes imágenes, donde se comprueba con las entradas (A partir de 1 lógico) el funcionamiento para cada una de las salidas (P, N, M).



Universidad Autónoma del Estado de México

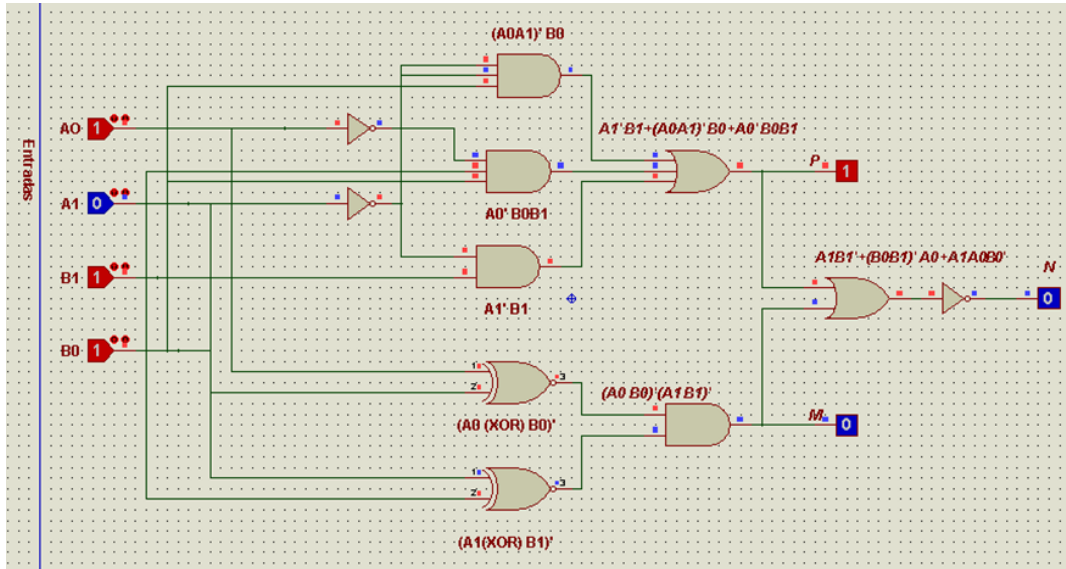


Figura 8.5.- Simulación correcta, se agregan los valores de entrada A0, A1(10) y B0, B1 (11) donde $A < B$ para el funcionamiento de P (En decimal $A=2$ $B=3$ $A < B$), elaborado en Proteus®

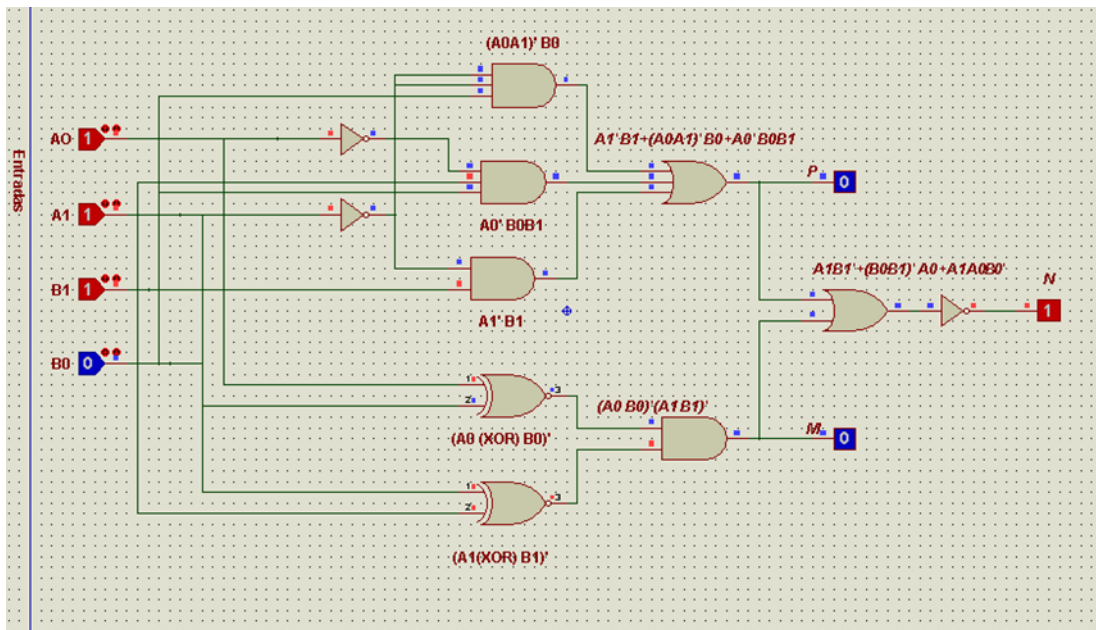


Figura 8.6.- Simulación correcta, se agregan los valores de entrada A0, A1(11) y B0, B1 (10) donde $A > B$ para el funcionamiento de N (En decimal $A=3$ $B=2$, elaborado en Proteus®



Conclusiones Generales

Este manual de prácticas tiene como objetivo proporcionar los conceptos básicos de la lógica combinatoria y secuencial, por medio del uso de hardware y software. Los Circuitos Integrados utilizados, son de tipo comercial por lo que el alumno puede adquirirlo de manera rápida y sencilla. El software utilizado, en ocasiones se cuenta disponible con licencia, dependiendo del lugar donde se trabaje, si no fuera así, existe la posibilidad de utilizar software de tipo de prueba.

El alumno a través de los conceptos impartidos en su clase, y reforzados en la literatura propuesta puede comprender de manera rápida los conceptos aquí planteados, además de los problemas, en este sentido, este manual de prácticas tiene sólo como objetivo ser un complemento para que el alumno pueda realizar y cotejar sus resultados con base en cada una de las ocho prácticas planteadas.

La información agregada a manera introductoria es fundamental, por lo tanto, el alumno podrá realizar la búsqueda en la literatura que más le agrade y contenga los contenidos del temario.

El software utilizado puede cambiar dependiendo de las características de licencia que se tenga en el espacio académico.