Universidad Autónoma de Baja California

Escuela de Ciencias de la Ingeniería y Tecnología

Sistemas Digitales con CPLDs

Manual de Prácticas

Preparado por:

- M.C. Miguel Alejandro Díaz Hernández
- Dra. Norma Alicia Barboza Tello
- Dr. Juan Miguel Colores Vargas
- Dr. Paul Medina Castro

Revisado por:

- Academia de Electrónica Analógica

Aprobado por:

- Dr. David Abdel Mejía Medina
  Subdirector de la ECITEC

[Caracterización de compuertas lógicas]

[Práctica No.1] | [Sistemas Digitales] | [Duración: 2 horas]

Introducción

Las compuertas lógicas son los bloques fundamentales de los circuitos digitales, por lo tanto, para comprender el funcionamiento de cualquier sistema digital, es necesario entender el comportamiento de las compuertas lógicas básicas.

En esta primera práctica, se comprobarán las tablas de verdad de tres compuertas lógicas básicas: AND, OR y NOT. Se armaran circuitos de prueba con cada uno de los Circuitos Integrados que contienen a estas compuertas y se probará cada una de las combinaciones de entrada posibles para así observar la salida y comparar el resultado con las tablas de verdad teóricas.

Para poder realizar esta práctica, es necesario:

- Conocer el funcionamiento y la tabla de verdad de las tres compuertas lógicas básicas.
- Saber armar circuitos utilizando las conexiones del protoboard.

Material

- Fuente de voltaje de 5V o baterías AA (si no utilizan fuente).
- Porta baterías (solo si utilizan baterías).
- Protoboard.
- Alambre de cobre para protoboard (ó cable de ethernet).
- Pinzas de punta y de corte.
- Circuito Integrado 7408 (AND).
- Circuito Integrado 7432 (OR).
- Circuito Integrado 7404 (NOT).
- 1 DIP Switch de 8 o 4 posiciones.
- 3 resistencias de 330 ohms (ó 220 ohms)
- 1 diodo emisor de luz (LED)

Actividades a realizar ANTES del día de la práctica

1.- Medir el voltaje de cada una de las baterías para determinar cuántas serán necesarias. En algunos casos (ver siguiente punto), estos circuitos integrados tienen un **voltaje mínimo recomendado** de 4.5 V y un **voltaje máximo recomendado** de 5.5 V. Normalmente, las baterías recargables tienen un voltaje alrededor de 1.2 V, por lo que se necesitarían 4 baterías (4.8 V). Las baterías alcalinas nuevas tienen 1.5 V, por lo que solo 3 serían necesarias (4.5 V).

2.- Identificar en las hojas de datos de cada una de las compuertas lógicas lo siguiente. Traer impresas las hojas con lo que se le pide marcado o señalado.

- Diagrama interno y conexiones externas.
- Valores de **voltaje de alimentación máximo y mínimo** para el fabricante de las compuertas que utilizaran. El fabricante se indica en la parte de plástico de la compuerta. Adaptar el paso número 1 para el valor obtenido.
- Valor mínimo del estado ALTO (1 lógico).
- Valor máximo del estado BAJO (0 lógico).

3.- Traer armado en el protoboard el circuito de la Figura No. 1. Se indica el diagrama eléctrico y en la Figura No. 2, a manera de ayuda, las conexiones físicas que se deben realizar. Es necesario (y requisito para entrar al laboratorio) traer ya armado el circuito, ya que la explicación se realizará en base a este.

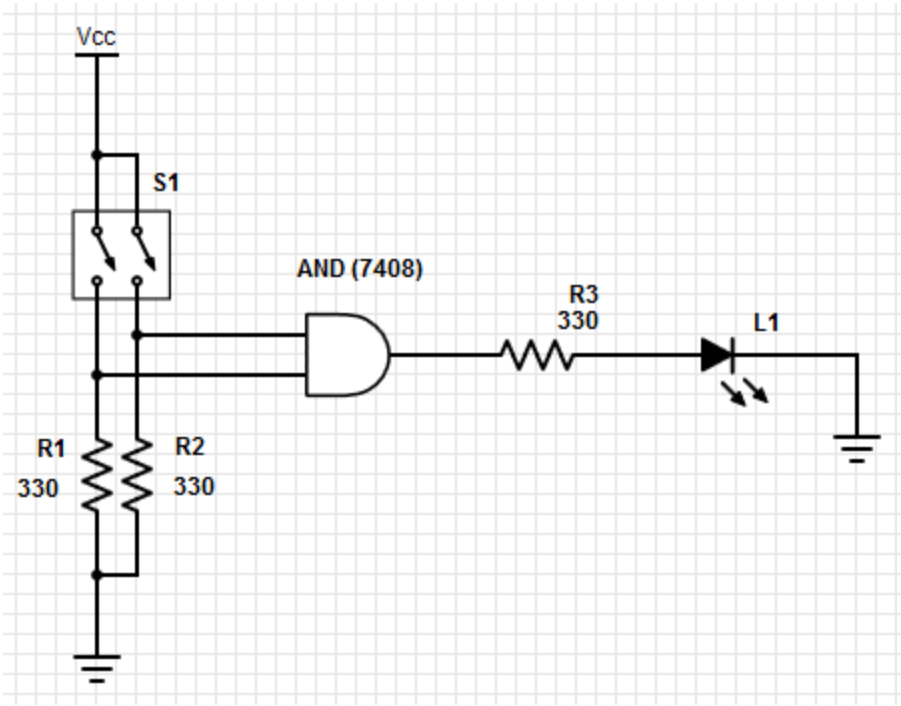


Fig.1 Diagrama eléctrico

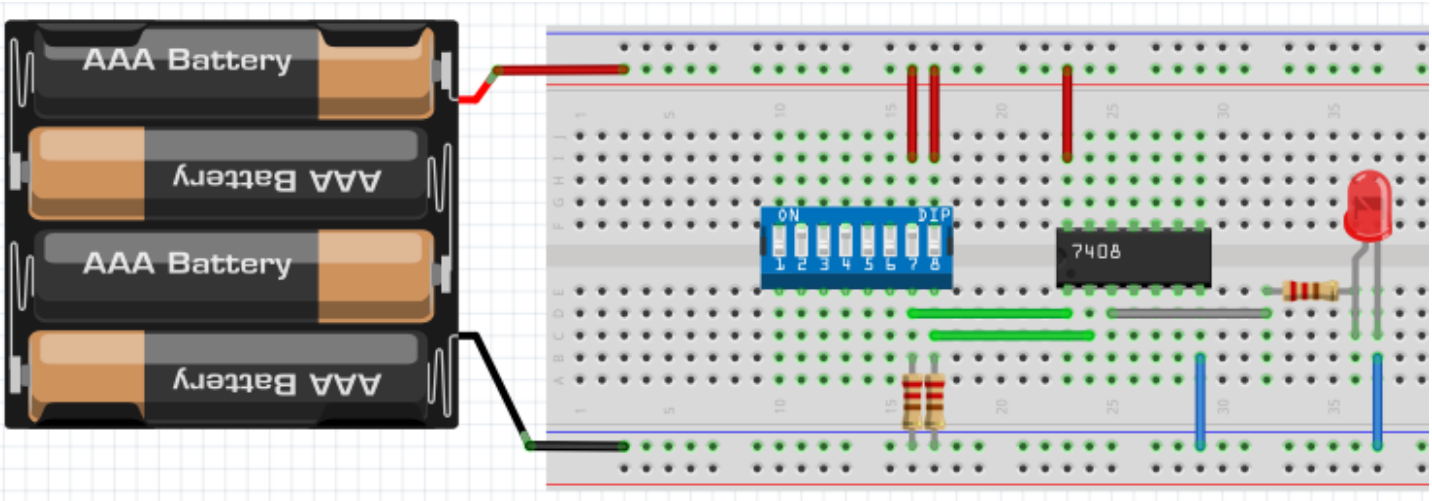


Fig. 2 Conexiones físicas

Actividades a realizar el día de la práctica

1.- Explicación.

2.- El circuito que se elaboró previamente en el protoboard es un circuito de prueba para la compuerta AND (7408). Apliqué energía al circuito y a través de las posiciones del DIP Switch introduzca todas las combinaciones de entrada de la tabla de verdad. Observe el resultado en el LED conectado a la salida. Registre lo observado.

3.- Repita el circuito básico de prueba para las otras dos compuertas (OR y NOT) y repita el experimento. Deberá revisar las hojas de especificaciones para realizar las conexiones de manera adecuada.

4.- Registre sus resultados y compárelos con las tablas de verdad vistas en clase.

5.- Una vez comprobadas las tablas de verdad de las tres compuertas. Vuelva a armar el circuito de la compuerta AND y utilizando una metodología estructurada y lógica, averigüe que comportamiento tienen las entradas de las compuertas lógicas cuando no se les conecta a 1 (5V) o a 0 (0V), es decir, cuando sus entradas se dejan desconectadas. ¿Qué sucede con la salida cuando ambas entradas están desconectadas? ¿Qué sucede cuando solo una de ellas está desconectada? Explique.

[Implementación de Circuitos Lógicos]

[Práctica No.2] | [Sistemas Digitales] | [Duración: 4 horas]

Introducción

Las compuertas lógicas son los bloques fundamentales de los circuitos digitales, por lo tanto, para comprender el funcionamiento de cualquier sistema digital, es necesario entender el comportamiento de las compuertas lógicas básicas.

En esta segunda práctica, se realizarán circuitos que contienen más de una compuerta lógica, para conocer la manera de interconectarlas entre ellas, además de obtener de manera práctica y teórica la tabla de verdad de circuitos con más de una compuerta.

Para poder realizar esta práctica, es necesario:

- Conocer el funcionamiento y la tabla de verdad de las tres compuertas lógicas básicas.
- Saber armar circuitos utilizando las conexiones del protoboard.

Material

- Fuente de voltaje de 5V o baterías AA (si no utilizan fuente).
- Porta baterías (solo si utilizan baterías).
- Protoboard.
- Alambre de cobre para protoboard (ó cable de ethernet).
- Pinzas de punta y de corte.
- Circuito Integrado 7408 (AND).
- Circuito Integrado 7432 (OR).
- Circuito Integrado 7404 (NOT).
- 1 DIP Switch de 8 o 4 posiciones.
- 6 resistencias de 330 ohms (ó 220 ohms)
- 1 diodo emisor de luz (LED)

Actividades a realizar

1.- Implementar en protoboard el circuito de la Fig. 1:

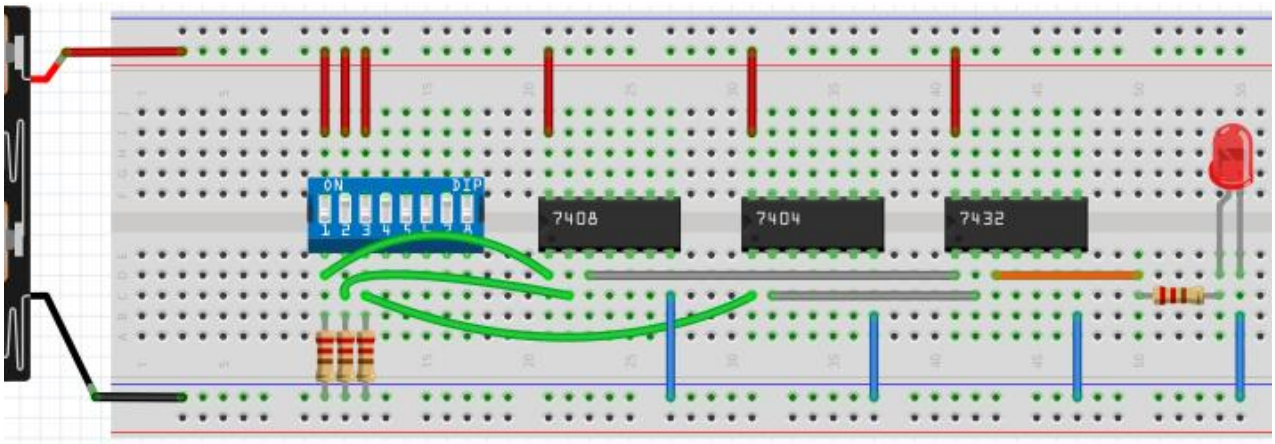


Fig.1 Circuito a implementar

2.- Obtener la tabla de verdad del circuito de la Fig.1 realizando todas las combinaciones posibles de los valores de entrada con ayuda del DIP SWITCH y observando en el LED conectado a la salida su respuesta. Escribir en el siguiente espacio la tabla de verdad obtenida de manera práctica.

A	B	C	Salida

Tabla 1 – Tabla práctica

3.- Dibujar en el siguiente espacio el diagrama lógico del circuito realizado en protobaord. Indicar claramente todas las conexiones.

4.- Escriba en el siguiente espacio la ecuación que representa al circuito dibujado y obtenga de manera teórica la tabla de verdad. Compárela con la tabla obtenida de manera práctica.

Ecuación:

A	B	C	Salida

Tabla 2 – Tabla teórica

5.- Implementar en protoboard el circuito descrito por la siguiente ecuación. Obtener su tabla de verdad de manera teórica y práctica.

$$X = [D + \overline{(A + B)C}].E$$

TEÓRICO						PRÁCTICO					
A	B	C	D	E	X	A	B	C	D	E	X

Tabla 3

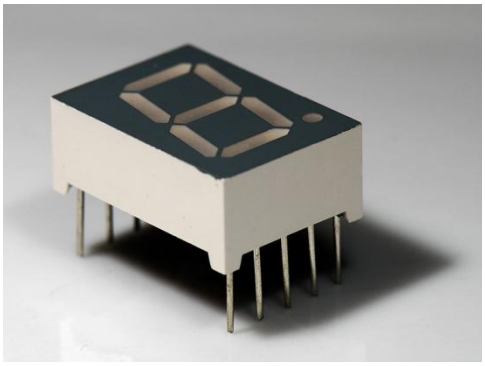
[Decodificador de 7 segmentos]

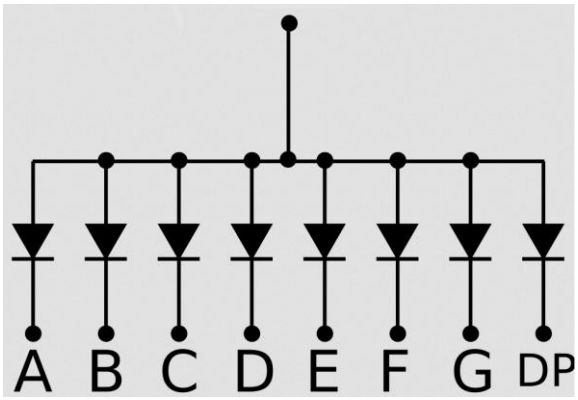
[Práctica No.3] | [Sistemas Digitales] | [Duración: 4 horas]

Introducción

Una de las maneras más sencillas y prácticas de presentar la información numérica o alfanumérica cuando esta es resultado de la implementación de algún circuito digital, es utilizando indicadores de 7 segmentos.

Los indicadores de 7 segmentos son arreglos de LEDs agrupados de tal forma que pueden formar números o letras dependiendo del LED o los LEDs que se encuentren encendidos en un instante dado.





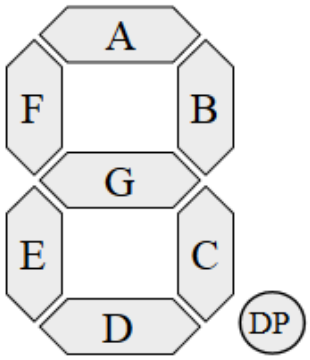


Fig.1 Ejemplo de indicador de 7 segmentos de Ánodo Común

Cada LED equivale a un segmento y para encender cada uno es necesario proporcionar la polaridad adecuada al segmento que nos interese. Existen dos tipos de indicadores de 7 segmentos:

- A aquellos en los que los ánodos (terminal positiva) de los LEDs están conectados todos unidos, les llamamos, de **"ánodo común".** Para encender un segmento individual en este tipo de indicador, necesitamos que el ánodo común esté conectado al voltaje de alimentación y proporcionar un **0 lógico** (tierra) al segmento que queremos encender.

- A aquellos en los que los cátodos (terminal negativa) de los LEDs están conectados todos unidos, les llamamos, de **"cátodo común".** Para encender un segmento individual en este tipo de indicador, necesitamos que el cátodo común esté conectado a 0V (tierra) y proporcionar un **1 lógico** (5V) al segmento que queremos encender.

Material

- Fuente de voltaje de 5V o baterías AA (si no utilizan fuente).
- Porta baterías (solo si utilizan baterías).
- Protoboard.
- Alambre de cobre para protoboard (ó cable de ethernet).
- Pinzas de punta y de corte.
- Circuitos Integrados 7408 (AND).
- Circuitos Integrados 7432 (OR).
- Circuitos Integrados 7404 (NOT).
- 1 DIP Switch de 8 o 4 posiciones.
- 12 resistencias de 330 ohms (ó 220 ohms)
- 1 indicador de 7 segmentos

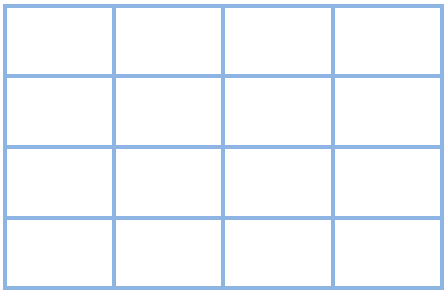
Actividades a realizar

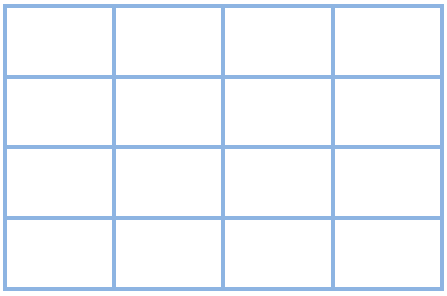
1.- El objetivo de la práctica es diseñar un circuito que tenga como entrada un número BCD (ingresado a través del DIP Switch) y que sus salidas formen el número correspondiente en un display de 7 segmentos. Como se puede ver, el circuito no tendrá una, sino 7 salidas (una para cada LED). El primer paso es realizar la tabla de verdad. Utilice la tabla que se muestra a continuación. Recuerde que es necesario saber qué tipo de display se utilizará (ánodo o cátodo común) para llenar la tabla correctamente.

A	B	C	D	a	b	c	d	e	f	g
0	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	0	0	0							
1	0	0	1							

Tabla 1 - Tabla de decodificación

2.- Para diseñar el circuito se utilizarán Mapas de Karnaugh. Será necesario realizar un mapa para cada salida. A partir de la tabla de verdad, llene y resuelva los mapas para cada una de las salidas. Dibuje el circuito resultante al lado de cada uno de los mapas.





3.- Implemente el circuito resultante en el protoboard y compruebe que los números del 0 al 9 se despliegan correctamente. Recuerde los siguientes puntos:

- El número que se desea representar en el display de 7 segmentos se introduce en BCD (código digital de 4 bits) utilizando el DIP Switch.

- Verifique los términos resultantes de los mapas de Karnaugh, no es necesario repetir términos, es decir, si la operación **and** de ***A*** y ***B*** resulta en más de un segmento, esta solo se tiene que realizar una sola vez, conectando su salida a los segmentos en donde corresponda. Esto también ayudará a simplificar un poco más el circuito.

- Para conectar correctamente el display de 7 segmentos es necesario obtener previamente su diagrama de conexión interno, es decir, en que pin se encuentra cada segmento y el cátodo o ánodo común. Si no cuenta con las hojas de datos, puede realizar este paso a prueba y error con la opción de medición de diodos del multímetro digital.

- Recuerde conectar resistencias limitadoras de corriente a cada uno de los segmentos (LEDs).

- La técnica de alambrado es de suma importancia para el éxito de esta práctica. No utilice jumpers, utilice alambre para protoboard cortado específicamente para la conexión que se este haciendo. Lleve un registro de lo que ya se alambró. No arme todo el circuito de una sola vez, alambre el circuito para cada segmento y pruébelo, ya que este funcione pase al siguiente segmento, repita hasta completar los siete.

[Simulación de Circuitos Lógicos en Quartus II]

[Práctica No.4] | [Sistemas Digitales] | [Duración: 4 horas]

Introducción

Los métodos manuales para el diseño e implementación de circuitos digitales son prácticos solamente cuando el circuito es relativamente pequeño. Para diseños más complejos, que son la mayoría sobre los que se trabaja en la actualidad, se utilizan Lenguajes de Descripción de Hardware (HDL, por sus siglas en ingles). Estos lenguajes y sus herramientas permiten la descripción, el diseño y la simulación de un circuito en software antes de su implementación o fabricación.

Un HDL se parece a un lenguaje de programación común, como C o Java, pero esta específicamente diseñado para la descripción de estructuras de hardware y el comportamiento de circuitos lógicos. En un HDL se pueden representar diagramas lógicos, tablas de verdad, expresiones Booleanas y expresiones que representan el comportamiento de un sistema digital.

En la actualidad existen dos lenguajes de descripción de hardware que se utilizan como estándares en la industria: Verilog y VHDL. Ambos lenguajes son aceptados por la IEEE, y la elección sobre cual utilizar depende de factores como son las herramientas disponibles o el lenguaje de diseños previos.

Está practica es una introducción a Verilog. Verilog utiliza palabras clave, de las cuales hay alrededor de 100. Estas palabras clave identifican a diversas construcciones del lenguaje y se escriben siempre en letras minúsculas. Algunos ejemplos de palabras clave son: module, endmodule, input, output, wire, and, or y not. En esta práctica, implementaremos un circuito digital combinacional en Verilog, utilizando el enfoque de conexión de compuertas.

Material

- Fuente de voltaje de 3.3V
- Protoboard.
- Alambre de cobre para protoboard (ó cable de ethernet).
- Pinzas de punta y de corte.
- Tarjeta de desarrollo para PLD EPM240 de Altera
- 1 DIP Switch de 8 o 4 posiciones.
- 4 resistencias de 330 ohms (ó 220 ohms)
- 6 jumpers macho-hembra
- 2 diodos emisores de luz (LED)

Actividades a realizar

1.- Abra el software **Quartus II** y de click en la opción "New Project Wizard". En la siguiente pantalla de click en "Next".

2.- En la siguiente pantalla se debe ingresar el directorio en donde se guardarán los archivos del proyecto. Este directorio debe ser creado específicamente para ello y debe estar vacío, el nombre del directorio no debe contener espacios ni caracteres especiales. Las otras dos casillas (nombre del proyecto y nombre de la entidad de mayor nivel) **deben tener el mismo nombre que el directorio.** Al terminar presione "Next".

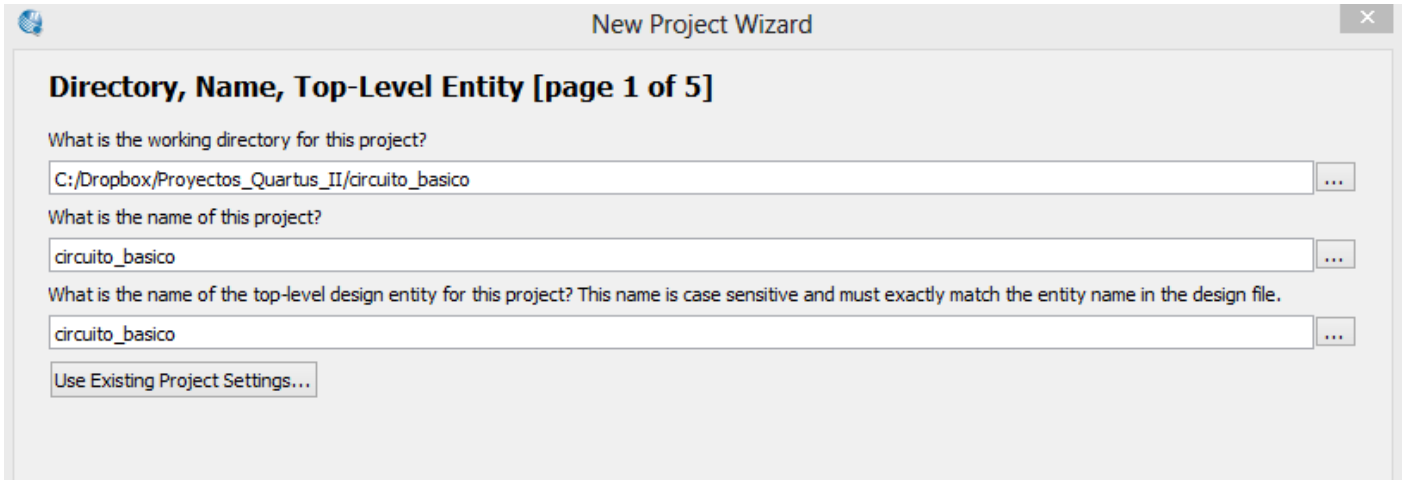


Fig.1

3.- Presione "Next" en la pantalla 2. En la pantalla 3 seleccione el dispositivo a utilizar, el dispositivo es de la familia **MAX II**, y es el **EPM240T100C5.** Selecciónelo y de click en "Finish".

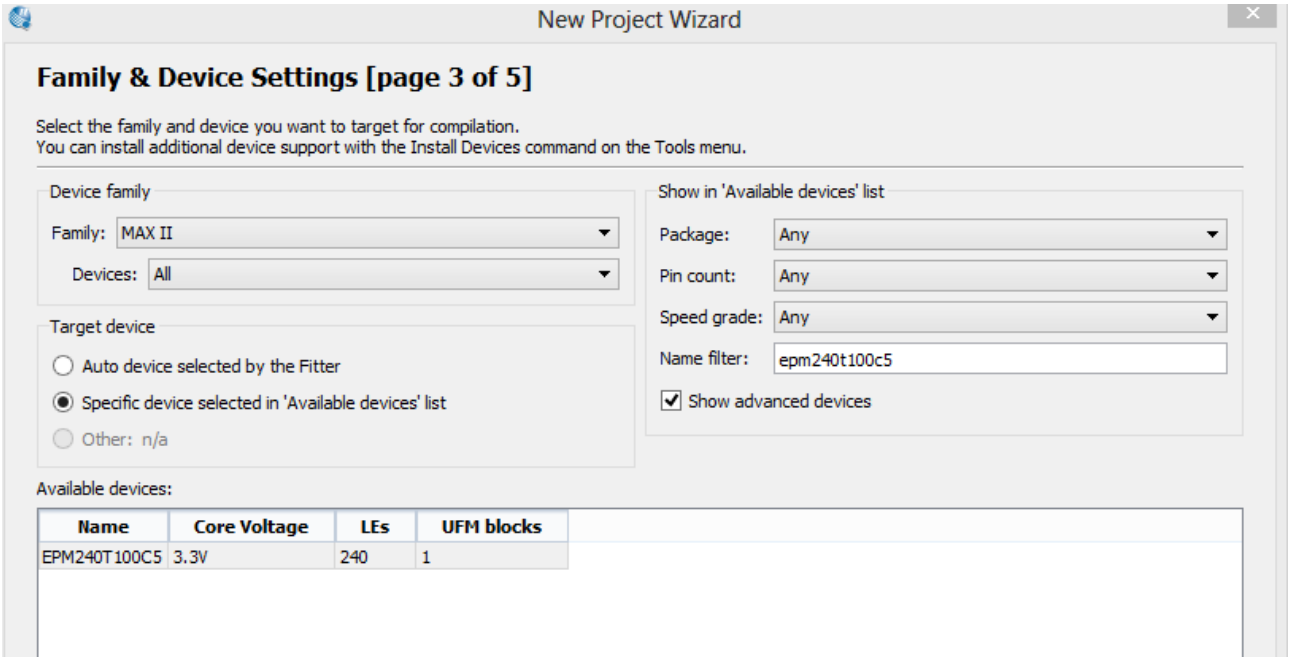


Fig. 2

4.- Seleccione la opción "File -> New -> Block Diagram/Schematic File".

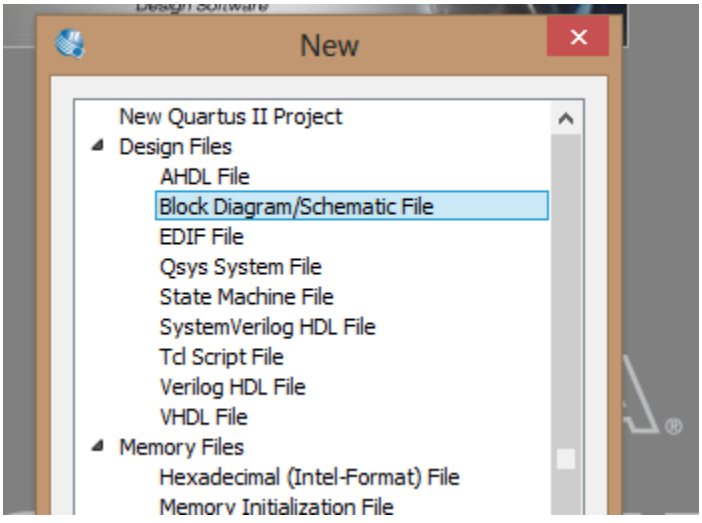


Fig. 3

5.- Con el archivo seleccionado, vaya a la opción "File -> Save As" y **guarde el archivo con el mismo nombre que el proyecto y el directorio.**

6.- Seleccione la opción "File -> New -> Verilog HDL File".

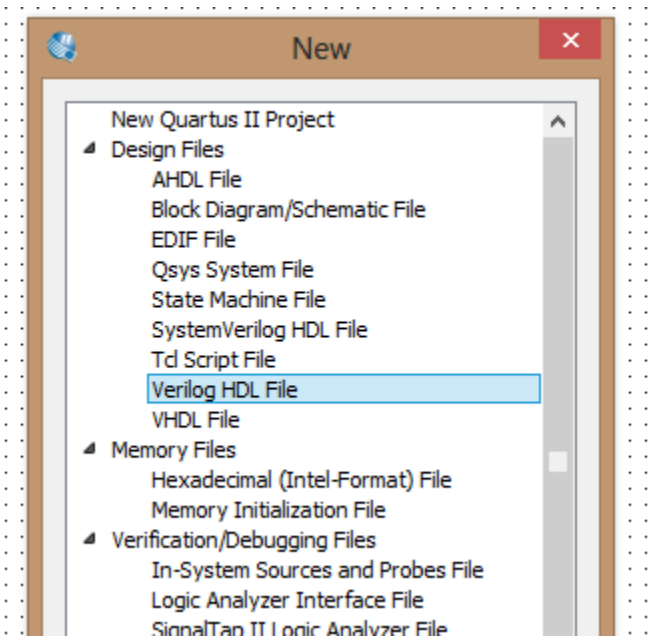


Fig. 4

7.- Con el archivo seleccionado, vaya a la opción "File -> Save As" y **guarde el archivo con un nombre descriptivo y que sea diferente al del archivo anterior.**

8.- En este momento el proyecto debe tener dos archivos, uno con extensión **bdf** (diagrama a bloques) y otro con extensión **v** (código en Verilog). Vamos a generar el circuito en Verilog.

Fig. 5

9.- El circuito a implementar es el siguiente:

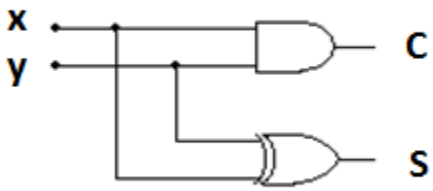


Fig. 6

10.- En Verilog, la construcción básica es el módulo. Un módulo se declara con la palabra clave **module** y al final, para cerrar la declaración, se utiliza la palabra clave **endmodule.** Cada módulo debe incluir un identificador y su lista de puertos declarada entre paréntesis y separados por comas.

```
1  module medio_sumador (S,C,x,y);
2
3
4
5
6
7  endmodule
8
```

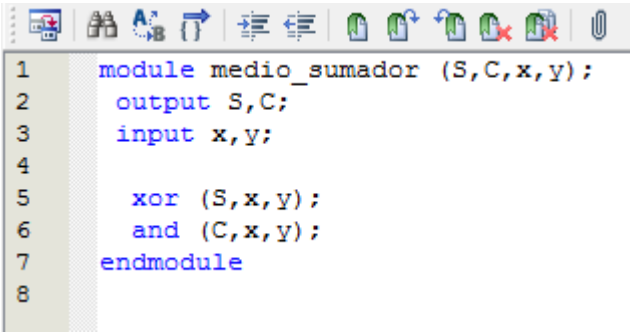
Fig. 7

11.- Ahora utilizamos las palabras claves **input** y **output** para especificar cuáles son los puertos de entrada y cuales los de salida.

```
1  module medio_sumador (S,C,x,y);
2   output S,C;
3   input x,y;
4
5
6
7  endmodule
8
```

Fig. 8

12.- El siguiente paso es definir la estructura del circuito, esto se realiza utilizando *primitivas*. Una *primitiva* es una definición básica dada por el lenguaje que el usuario no puede cambiar (a diferencia de un módulo), pero que puede utilizar como si se tratará de compuertas en un protoboard. Para implementar el medio sumador, utilizaremos las primitivas **and** y **xor.**



```
1  module medio_sumador (S,C,x,y);
2   output S,C;
3   input x,y;
4
5     xor (S,x,y);
6     and (C,x,y);
7  endmodule
8
```

Fig. 9

Las primitivas se colocan en una lista, utilizando su palabra clave para describirlas. Se deben de listar también las salidas y las entradas de cada compuerta entre paréntesis y separadas por comas, **la salida de la compuerta se debe de listar primero**, seguida de sus entradas.

13.- El siguiente paso es crear el símbolo del circuito para agregarlo al diagrama a bloques. Con el archivo **.v** seleccionado, vaya a la opción "File -> Create/Update -> Create Symbol Files for Current File".

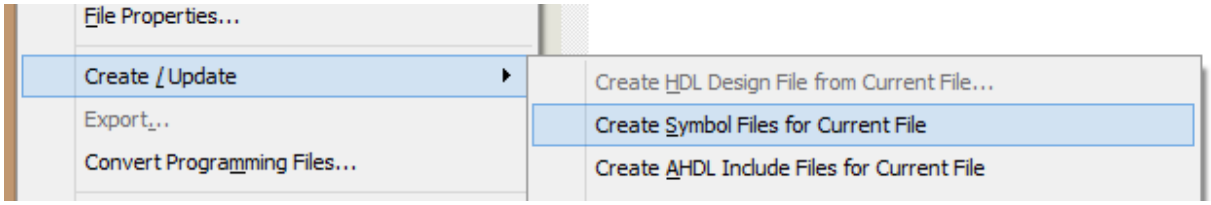


Fig. 10

14.- Un vez terminado recibirá el siguiente mensaje:

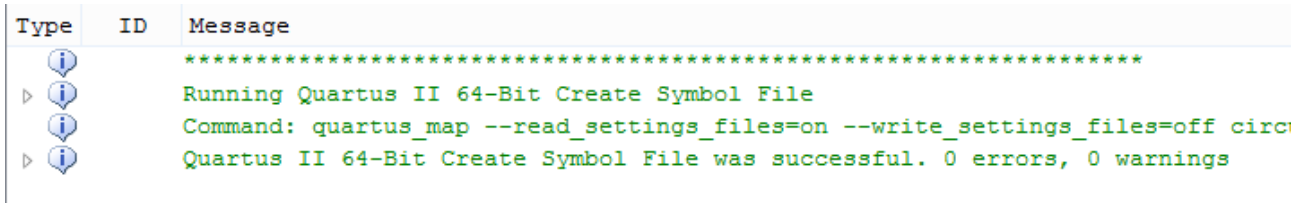


Fig. 11

15.- Con el archivo .bdf seleccionado, de click derecho en cualquier parte del diagrama y seleccione "Insert -> Symbol".

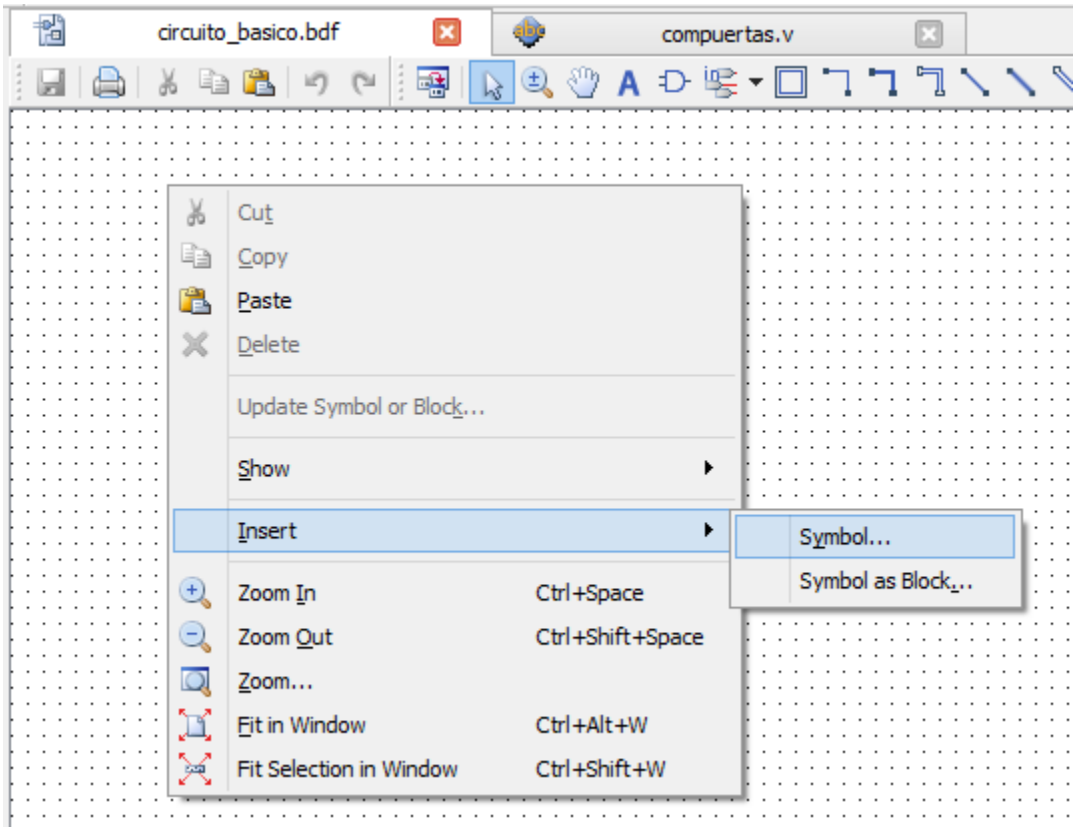


Fig. 12

16.- Inserte el símbolo que corresponde al archivo Verilog recién creado.

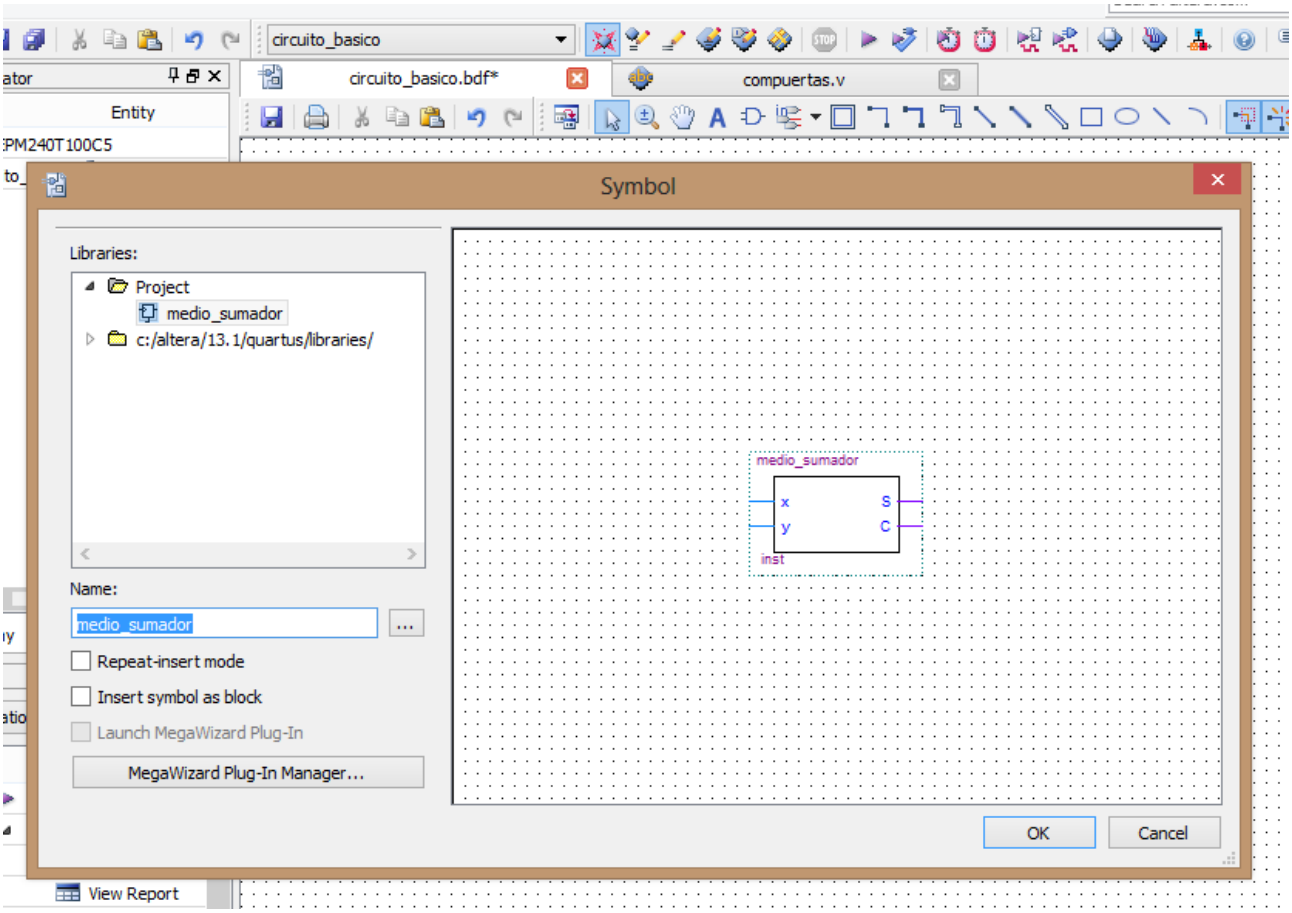


Fig. 13

17.- De click derecho sobre el símbolo recién insertado y seleccione la opción "Generate Pins for Symbol Ports"

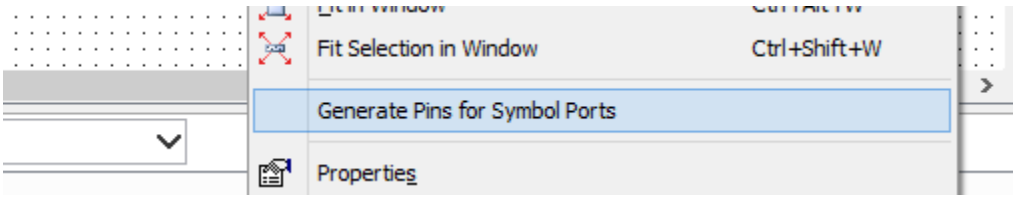
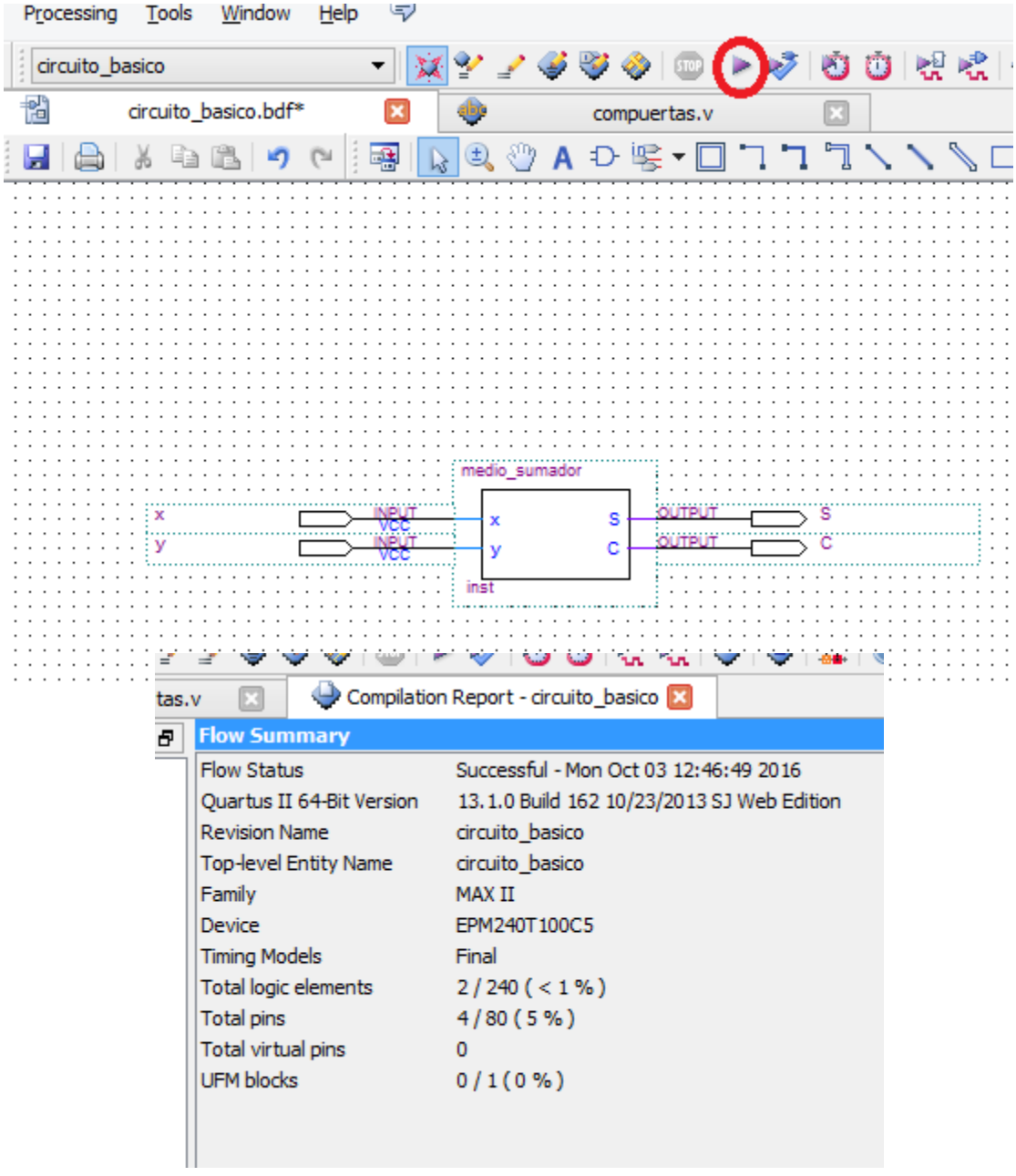


Fig. 14

18.- Compile su circuito seleccionando el botón indicado o la opción "Processing -> Start Compilation."



La información que se nos presenta al finalizar la compilación es importante, ya que nos indica posibles errores, y además, nos da un reporte con la cantidad de recursos utilizados, siendo el más importante el que nos indica el número de **elementos lógicos.**

19.- Una vez concluido exitosamente el proceso de compilación, es necesario simular el circuito. Para esto, vamos a la opción "File -> New -> University Program VWF".

20.- Se abrirá la pantalla del "Simulator Waveform Editor", que nos permitirá indicar formas de onda para las entradas del circuito. Antes que cualquier cosa, es necesario agregar las entradas y salidas que queremos simular, para esto, vaya a la opción "Edit -> Insert -> Insert Node or Bus -> Node Finder -> List". Una vez ahí, agregué todas las señales a la pantalla *Selected Nodes* utilzando las flechas **>>.** Presione dos veces seguidas OK y las señales deberán aparecer en la pantalla.

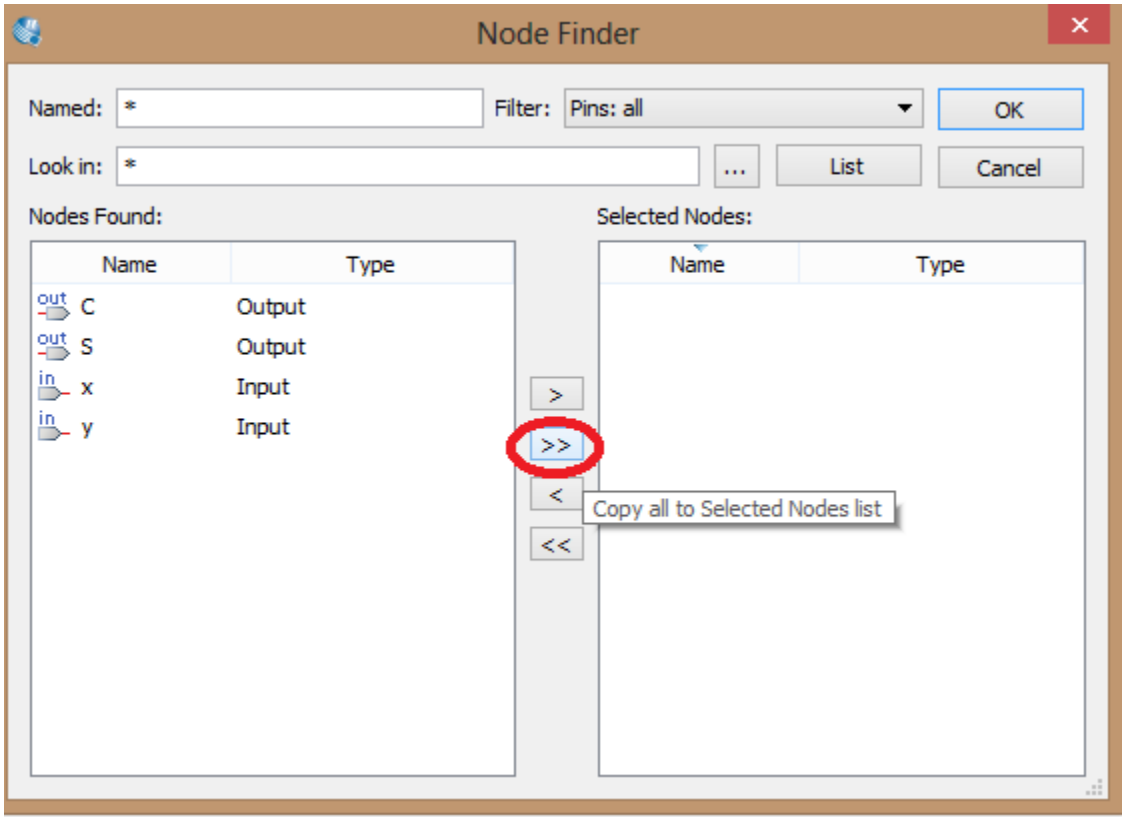


Fig. 15

Una vez que se agregaron todas las señales, están se pueden ordenar seleccionándolas dando click en su nombre y moviéndolas. Póngalas en el siguiente orden de arriba abajo: **x, y, S, C.**

21.- Una vez ordenadas, tenemos que asignarles a las entradas los valores que queremos sean simulados. Cada señal puede ser manipulada individualmente, seleccionando el rango de tiempo que queremos modificar y utilizando los botones de la barra de herramientas. En la siguiente imagen se asignó un "1" al rango seleccionado.

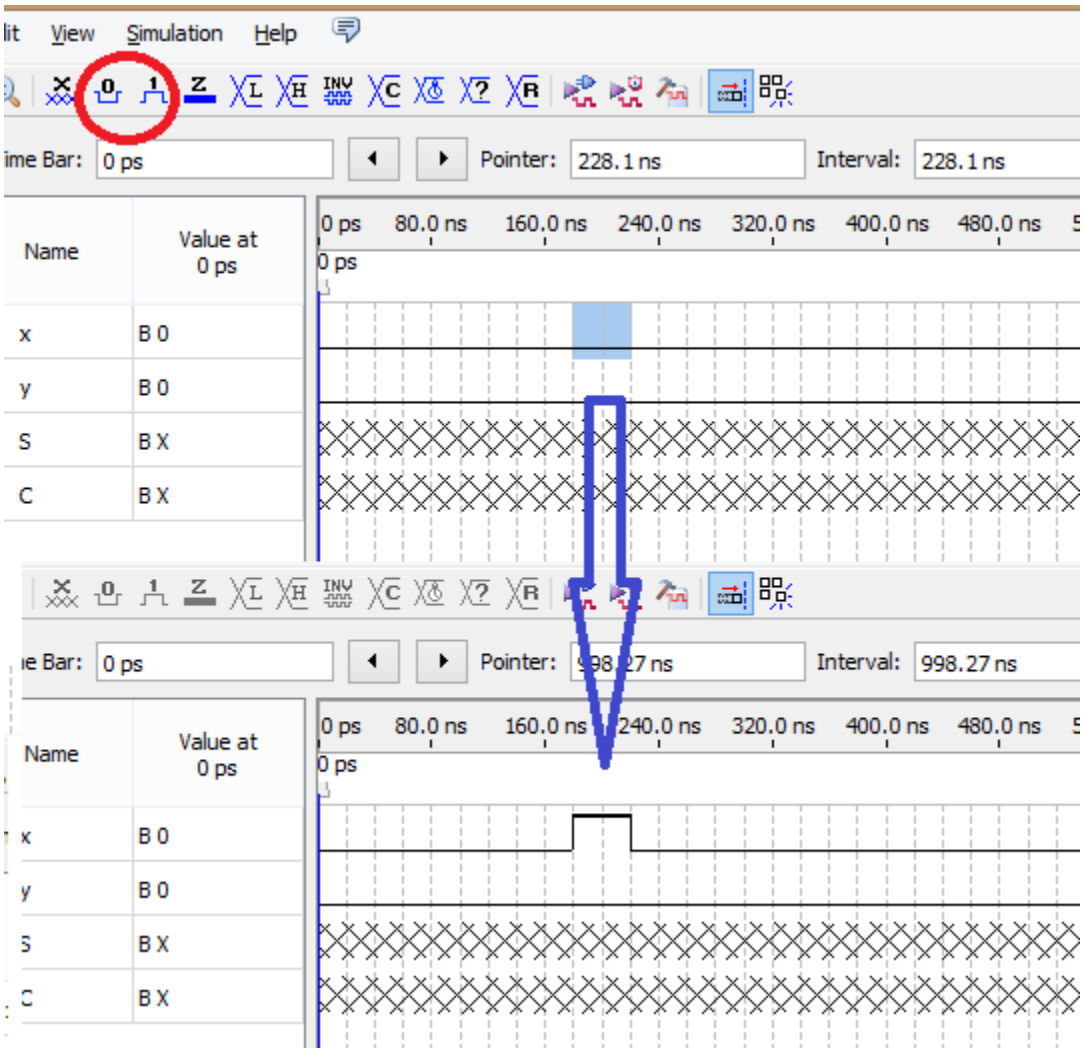


Fig. 16

22.- También es posible automatizar un poco este proceso. Para asignar los valores de entrada, la opción más lógica es poner todas las posibilidades, para así tener completamente caracterizado el circuito. Esto se puede lograr de la siguiente manera:

- Se agrupan las señales de entrada manteniendo presionado "Ctrl" en el teclado y dando click en ellas. Después se da click derecho en ellas y se selecciona "Grouping -> Group…" En la pantalla siguiente se les da un nombre y se deja la opción "Radix" en "Binary".

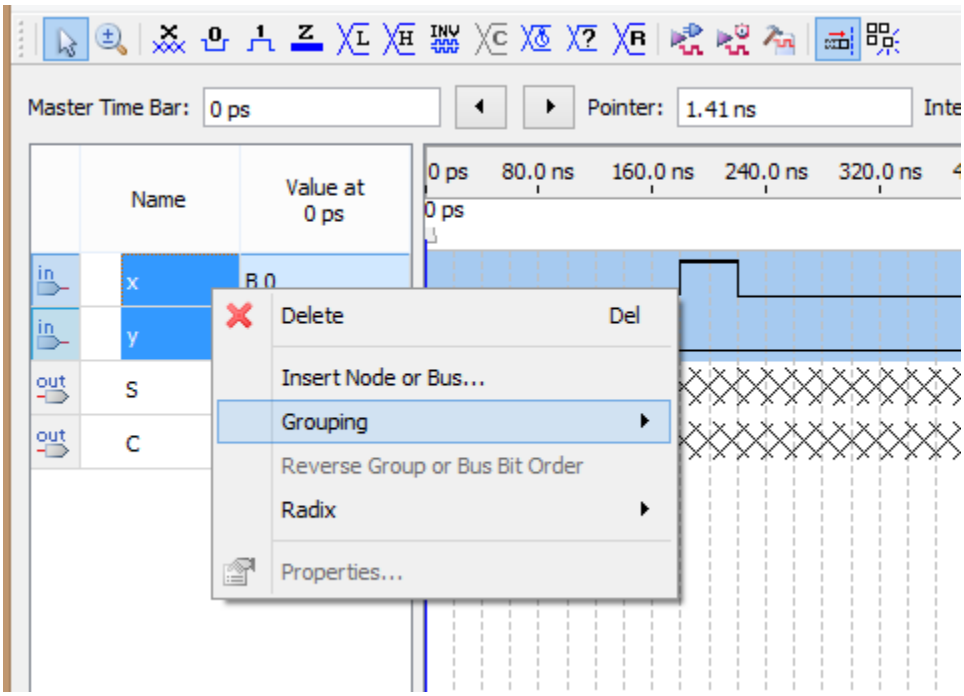


Fig. 17

- Con el nombre del grupo seleccionado se selecciona la opción "Count Value".

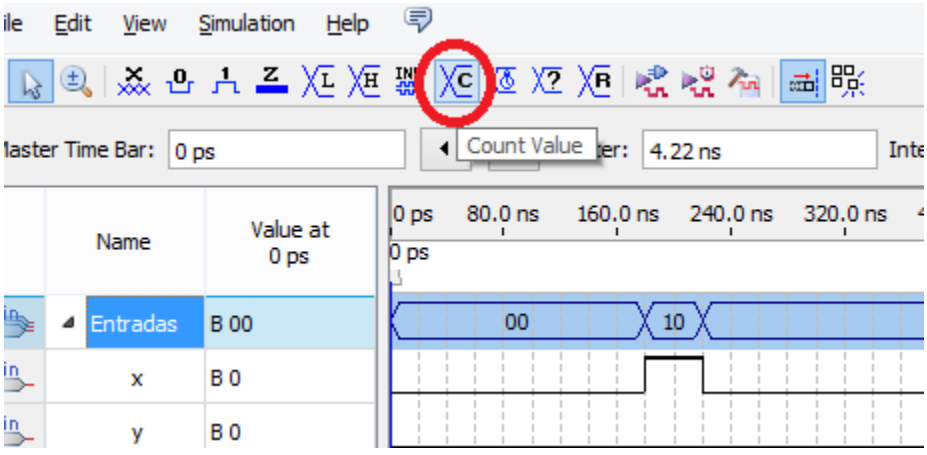


Fig. 18

- Cómo se puede observar en el eje de tiempo, la simulación tiene una duración de 1000 nS. Con dos entradas tenemos cuatro posibles combinaciones, por lo que dividimos este espacio de 1000 nS entre cuatro, cada combinación nos da un total de 250 nS.

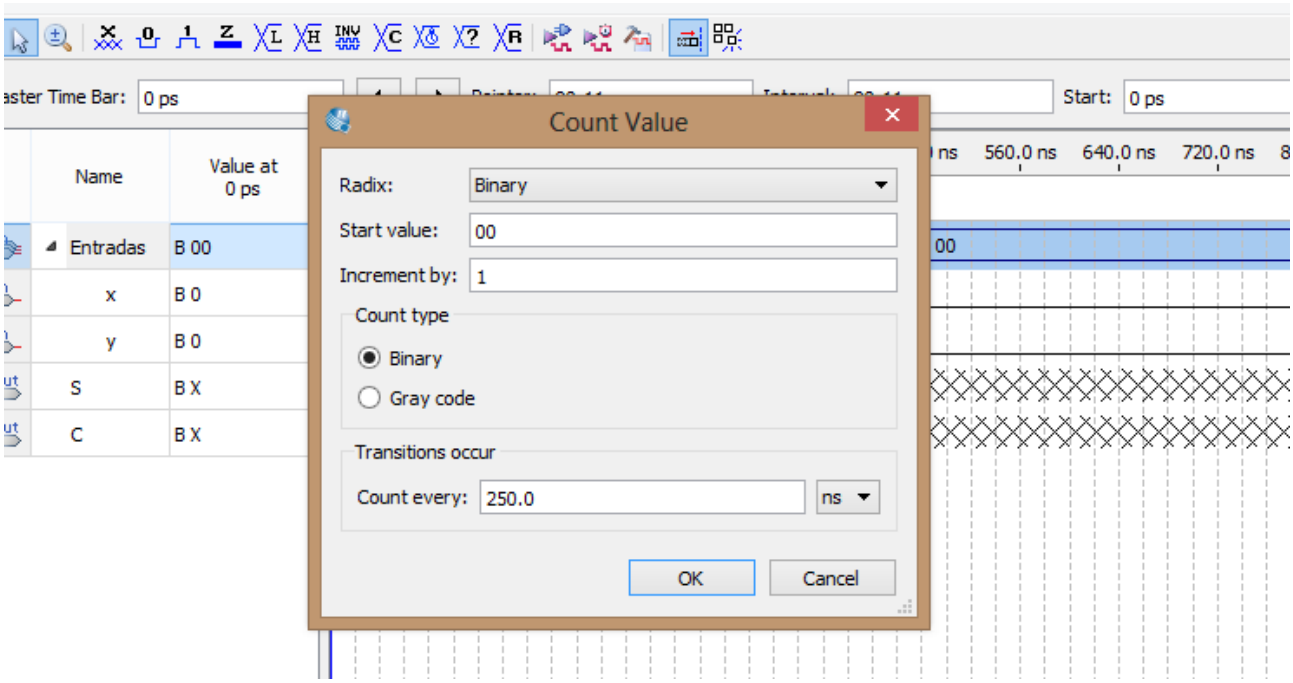


Fig. 19

- Finalmente, las señales de entrada quedan como sigue, y tenemos las 4 combinaciones posibles: **00, 01,10, 11.**

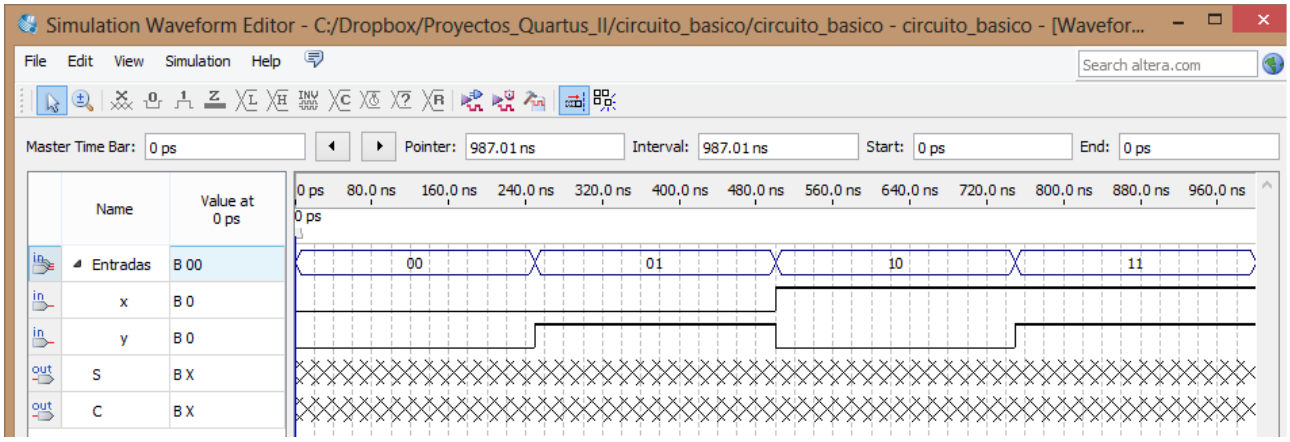


Fig. 20

23.- La simulación se realiza seleccionando la opción "Simulation -> Run Functional Simulation".

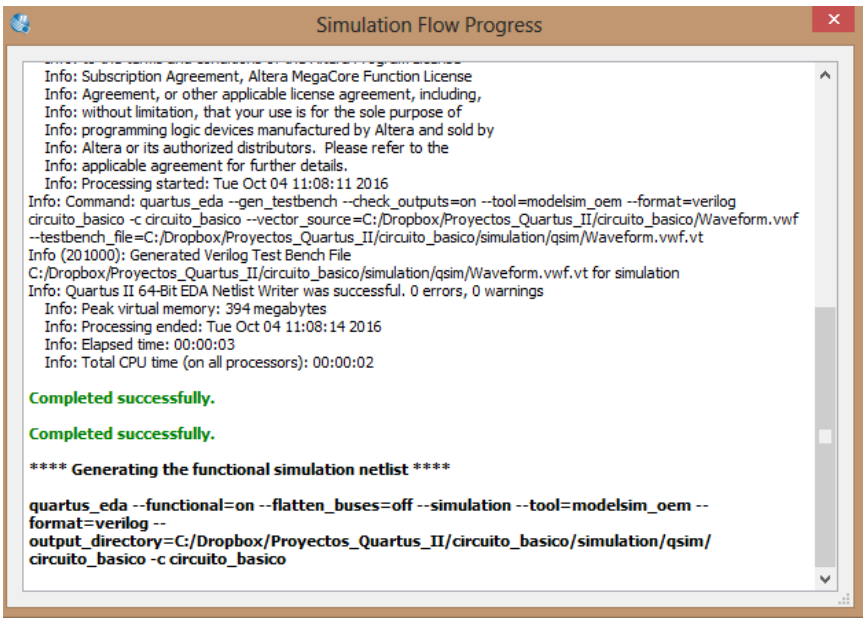


Fig. 20

24.- Finalmente, queda como sigue:

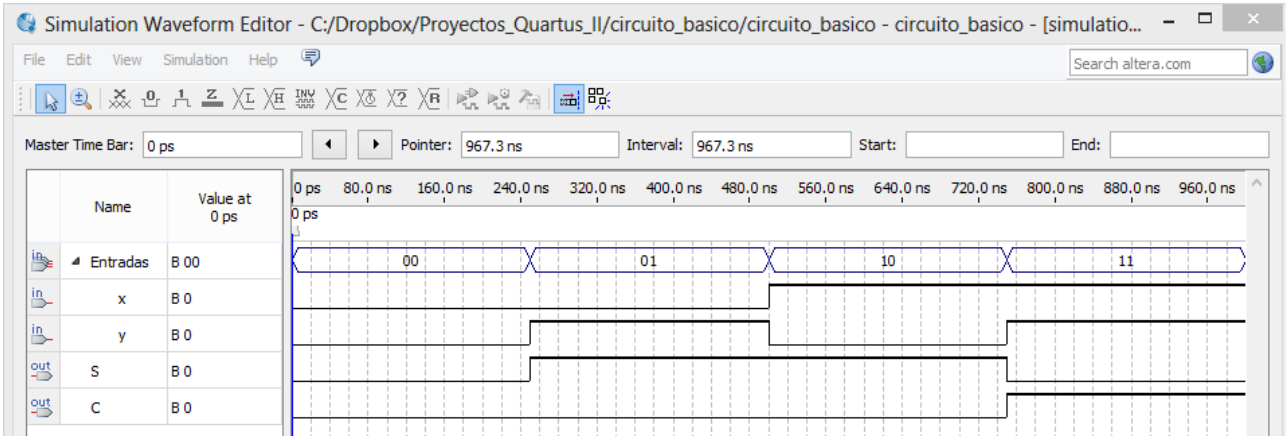


Fig. 21

25.- A partir de las señales obtenidas en la simulación, realice la tabla de verdad y compruébela de manera teórica.

X	Y	S	C

a) Simulación

X	Y	S	C

b) Teórica

Tabla. 1

26.- El siguiente es un circuito lógico implementado en Verilog, dibuje el diagrama esquemático que representa interpretando correctamente las declaraciones de módulos, la implementación de las primitivas y su interconexión (la palabra ***wire*** se utiliza para indicar conexiones internas entre compuertas, es decir, que no son ni entradas ni salidas del circuito). Realice la simulación del circuito para obtener su tabla de verdad y compárela con la obtenida de manera teórica. Reporte sus resultados.

```
1  module circuito_basico (A,B,C,D,E);
2   output D,E;
3   input A,B,C;
4   wire w1;
5
6   and G1 (w1,A,B);
7   not (E,C);
8   or (D,w1,E);
9  endmodule
```

Fig. 22

[Circuitos lógicos combinacionales en Verilog HDL]

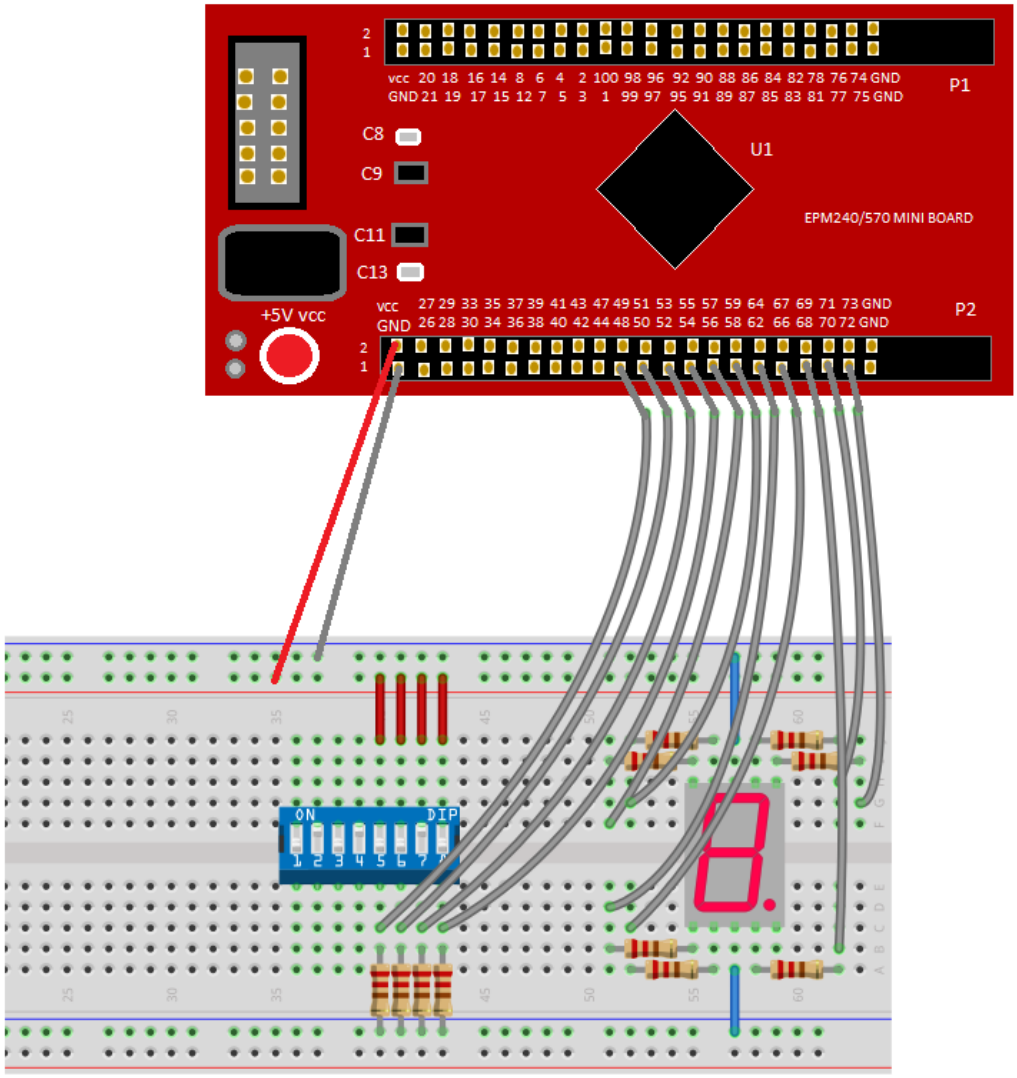
[Práctica No.5] | [Sistemas Digitales] | [Duración: 4 horas]

Material

- Fuente de voltaje de 5V o baterías AA (si no utilizan fuente).
- Porta baterías (solo si utilizan baterías).
- Protoboard.
- Alambre de cobre para protoboard (ó cable de ethernet).
- Pinzas de punta y de corte.
- Tarjeta de desarrollo para PLD EPM240 de Altera
- 1 DIP Switch de 8 posiciones.
- 12 resistencias de 330 ohms (ó 220 ohms)
- 13 jumpers macho-hembra
- 1 indicador de 7 segmentos

Actividades a realizar

1.-Implemente el siguiente circuito en protoboard utilizando el Dip Switch y el Display de 7 Segmentos. Antes de aplicar alimentación al circuito atienda la explicación del profesor sobre este particular.



2.- Utilizando el lenguaje **Verilog** y el software Quartus II, implementar y simular el circuito decodificador de BCD a 7 segmentos. Llene y utilice la tabla que se muestra a continuación. Recuerde que es necesario saber qué tipo de display se utilizará (ánodo o cátodo común) para llenar la tabla correctamente. Utilizando mapas de Karnaugh, obtenga las ecuaciones lógicas para cada segmento y realice la captura del circuito en Verilog.

A	B	C	D	a	b	c	d	e	f	g
0	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	0	0	0							
1	0	0	1							

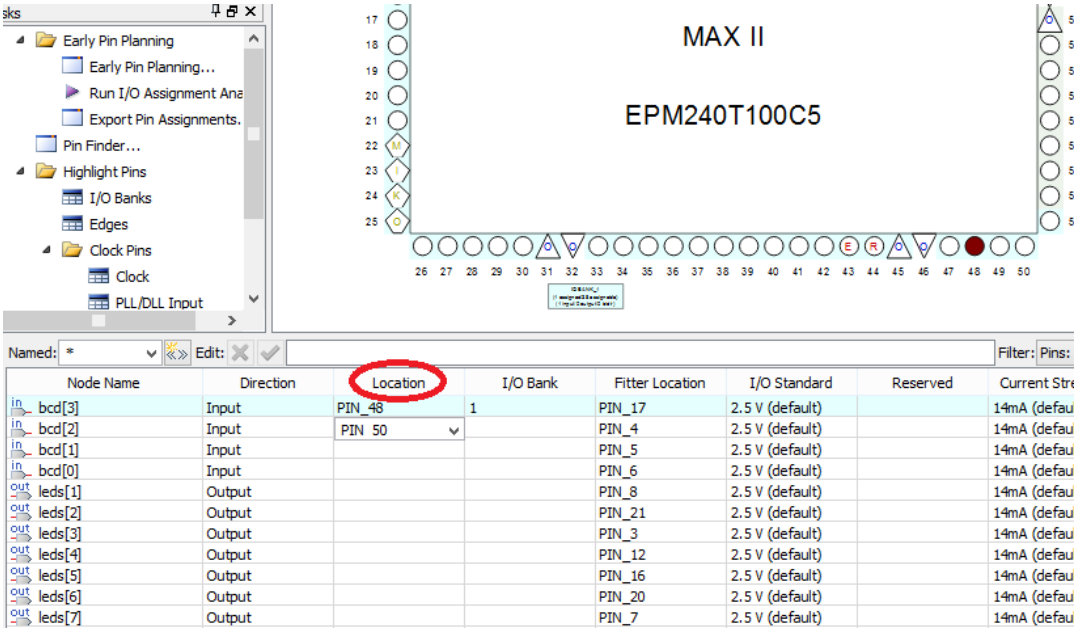
Tabla 1 - Tabla de decodificación

3.- Realice la compilación y la simulación del circuito utilizando las herramientas aprendidas en la práctica anterior.

4.- El siguiente paso antes de poder programar la tarjeta e implementar el circuito es asignar los pines que se utilizarán.

- Se asignará un pin para cada uno de las entradas y salidas que se declararon en el modulo.
- Dentro del del menú ***Assignments*** utilice la opción ***Pin Planner*** para asignar pines físicos a cada entrada y salida lógica. Se captura el número de PIN simplemente escribiendo el número en la casilla *location*, automaticamente se llena la parte de "PIN" con el número que se introdujo. Presione *Enter* y capture el siguiente Pin. Al terminar, cierre la ventana del ***Pin Planner.***

  * No se utilicen los pins 64, 44, 37, 39, 88, 90 y 59.



- Compile de nuevo.

5.- Utilice el programador para implementar el circuito en la tarjeta de desarrollo, esto se realiza desde la opción "Tools -> Programmer". Realice la programación y verifique el correcto funcionamiento del circuito.

[Programación Modular en Quartus II]

[Práctica No.6] | [Sistemas Digitales] | [Duración: 4 horas]

Introducción

Los PLDs o Dispositivos Lógicos Programables permiten la implementación de circuitos lógicos complicados de una manera más sencilla. Como ya se ha visto, nos pueden ahorrar mucho trabajo de alambrado y en algunos casos, en el diseño de circuitos y su simplificación. Es esta práctica utilizaremos la herramienta de diagrama a bloques de Quartus II para implementar circuitos digitales de manera modular, lo que nos dará la posibilidad de reutilizar código e implementar diseños de manera más eficiente.

Material

- Tarjeta de desarrollo para PLD EPM240 de Altera
- Fuente de voltaje de 5V o 3.3V.
- Protoboard.
- Jumpers macho-hembra
- Alambre de cobre para protoboard (ó cable de ethernet).
- Pinzas de punta y de corte.
- 1 DIP Switch de 8 posiciones.
- 13 resistencias de 330 ohms (ó 220 ohms)
- 5 LEDs

Actividades a realizar

1. El circuito a implementar es el siguiente.

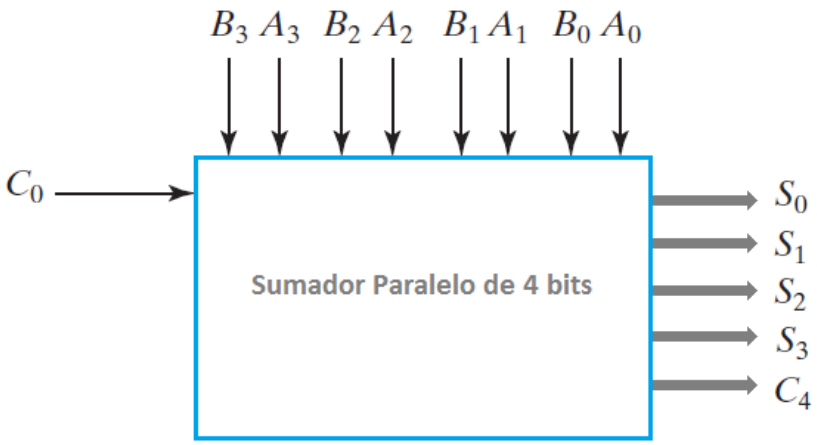


Fig. 1 Sumador Paralelo de 4 bits

2. Este circuito se compone de 4 sumadores completos interconectados.

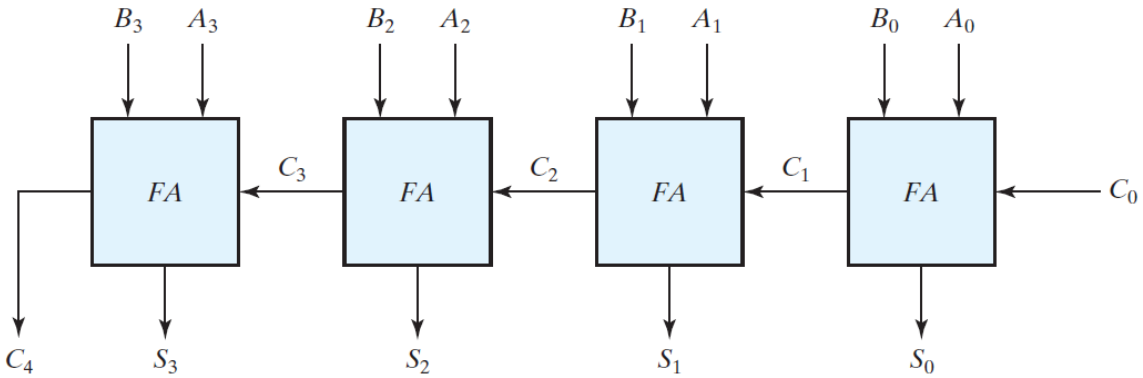


Fig. 2 Sumador Paralelo de 4 bits utilizando sumadores completos (Morris-Mano)

3. Y cada sumador completo se puede expresar en términos de medios sumadores de la siguiente manera.

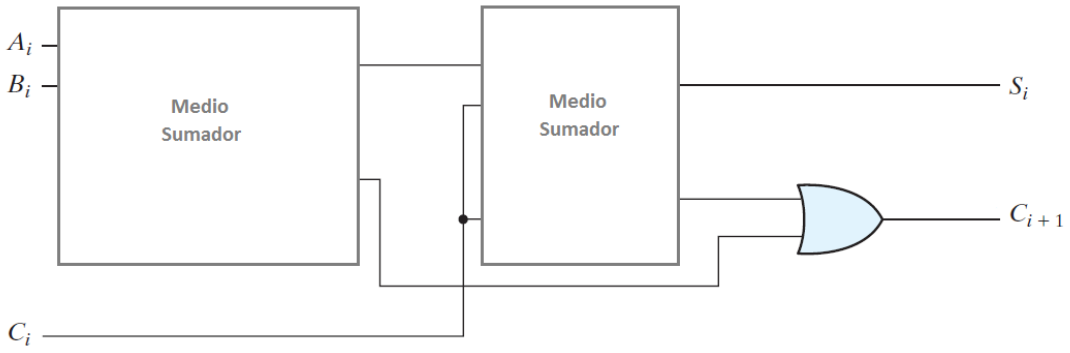


Fig. 3 Sumador Completo utilizando medios sumadores (Morris-Mano)

4. Antes de iniciar, ejecute el software Quartus II y cree un proyecto nuevo.

5. Agregué un archivo ***.bdf*** y nómbrelo de la misma manera que el proyecto.

6. Agregue un archivo Verilog HDL y nómbrelo ***medio_sumador.v*** Escriba en este archivo el código para el medio sumador utilizando compuertas. El nombre del módulo debe ser también ***medio_sumador***. Observe la correspondencia entre el código y el diagrama lógico del medio sumador (en sus apuntes). A partir de este código, cree el símbolo correspondiente. No lo inserte en el diagrama a bloques.

```
1  module medio_sumador (S,C,x,y);
2   output S,C;
3   input x,y;
4
5   xor (S,x,y);
6   and (C,x,y);
7
8  endmodule
9
```

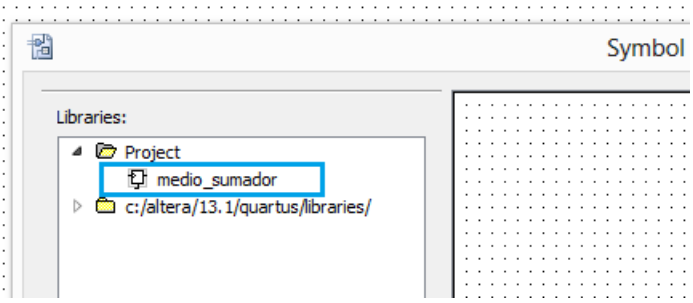


Fig. 4 Medio Sumador en Verilog y símbolo creado correctamente

7. Agregué otro archivo Verilog y nómbrelo ***sumador_completo.v*** Escriba en este archivo el código para el sumador completo que se muestra abajo. Observe la correspondencia entre el módulo y el diagrama a bloques de la figura 3. Ponga especial atención a la manera en que se encuentran interconectados los dos medios sumadores (note que en el código, a cada medio sumador se le da un nombre diferente) y en la manera en que interactúan las variables de cada módulo. A partir de este módulo cree el símbolo correspondiente. No lo inserte en el diagrama a bloques.

```
1   module sumador_completo (S,C,x,y,z);
2    output S,C;
3    input x,y,z;
4    wire S1,C1,C2;
5
6    medio_sumador ms1 (S1,C1,x,y);
7    medio_sumador ms2 (S,C2,S1,z);
8
9    or (C,C1,C2);
10
11  endmodule
12
```

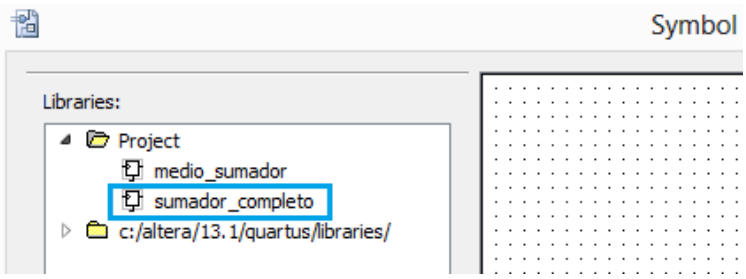


Fig. 5 Sumador Completo en Verilog y símbolo creado correctamente

8. El siguiente paso consiste en agregar otro archivo Verilog y crear el módulo ***sumador_paralelo.*** El diagrama a bloques de este módulo se encuentra en la figura 2. Como se puede apreciar es un sumador paralelo de 4 bits formado partir de la interconexión de **4 sumadores completos**. Repitiendo el procedimiento de los últimos dos pasos, escriba el código Verilog correspondiente al sumador paralelo, cree el símbolo correspondiente y agréguelo a su diagrama a bloques.

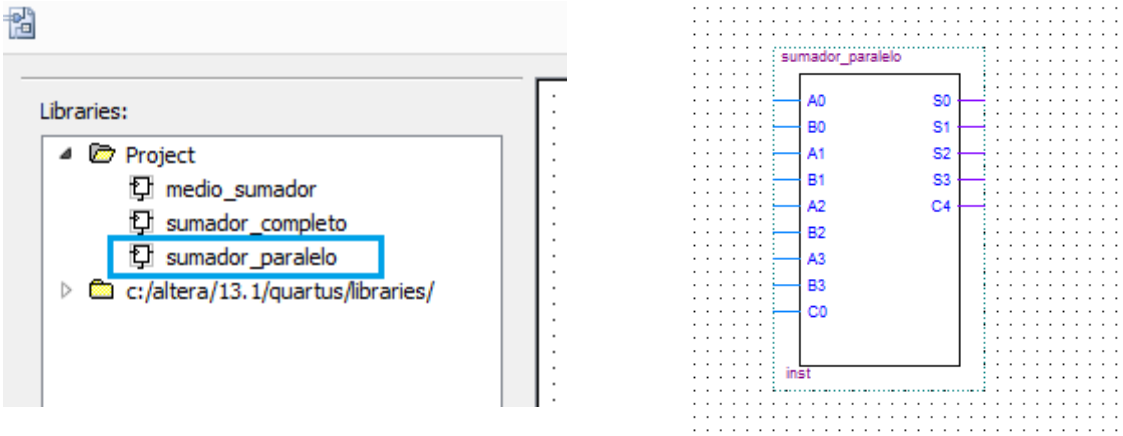


Fig. 6 Símbolo del Sumador Paralelo

9. Genere los pines de entrada y salida.

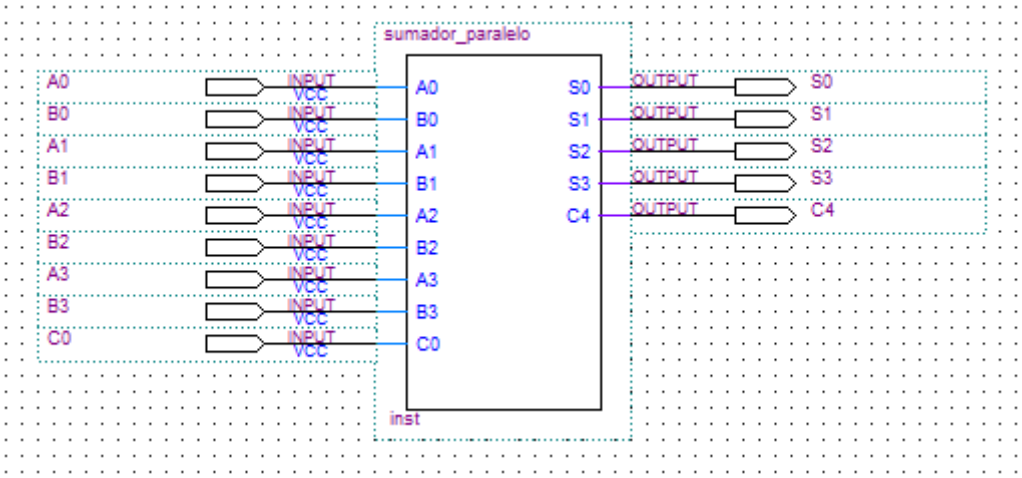


Fig. 7 Símbolo del Sumador Paralelo con pines de entrada y salida

10. Compile su proyecto. Dentro del menú ***Assignments*** utilice el ***Pin Planner*** para asignar pines físicos a cada entrada y salida lógica.

11. Compile de nuevo. Utilice el programador para implementar el circuito en la tarjeta de desarrollo y conecte el Dip Switch y los LEDs de manera que pueda comprobar el correcto funcionamiento del sumador completo.

[Multiplicador Combinacional]

[Práctica No.7] | [Sistemas Digitales] | [Duración: 6 horas]

Introducción

Los multiplicadores combinacionales son circuitos realizados a partir de compuertas lógicas que se encargan de realizar la operación de multiplicación. Para realizar un multiplicación, el multiplicando se tiene que multiplicar por cada uno de los bits del multiplicador, empezando por el bit menos significativo. Cada multiplicación produce un producto parcial, que después debe recorrerse una posición a la izquierda. Para obtener el resultado, se realiza la suma de todos los productos parciales.

En la siguiente figura se muestra una multiplicación de un multiplicando de 4 bits y un multiplicador de 3 bits (4 x 3). Se indican los productos parciales recorridos adecuadamente a la izquierda, y el resultado que se obtiene de la suma de dichos productos.

			B3	B2	B1	B0
		X		A2	A1	A0
			A0.B3	A0.B2	A0.B1	A0.B0
		A1.B3	A1.B2	A1.B1	A1.B0	
	A2.B3	A2.B2	A2.B1	A2.B0		
C6	C5	C4	C3	C2	C1	C0

Fig. 1 Multiplicación de 4 x 3 bits

Las multiplicaciones de bits individuales se pueden realizar utilizando compuertas AND, ya que ambas operaciones son idénticas. La suma de los productos parciales se realiza utilizando sumadores. En el caso del multiplicador de 4 x 3, el circuito queda como sigue:

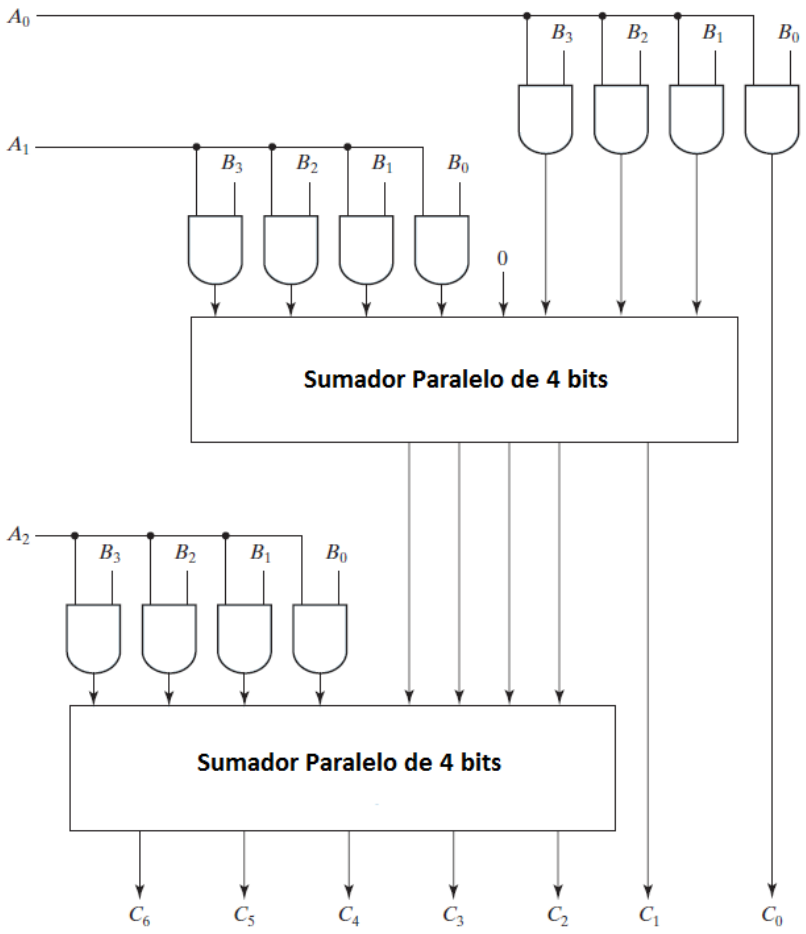


Fig. 2 Circuito multiplicador de 4 x 3 bits (Morris-Mano)

Se puede construir un multiplicador combinacional de mayor número de bits utilizando este procedimiento. Si el multiplicando tiene ***K*** bits y el multiplicador ***J*** bits, necesitaremos ***(J x K) compuertas AND, (J -1) sumadores paralelos de K bits*** y el resultado tendrá ***(J + K) bits.***

Material

- Tarjeta de desarrollo para PLD EPM240 de Altera
- Fuente de voltaje de 5V o 3.3V.
- Protoboard.
- Jumpers macho-hembra
- Alambre de cobre para protoboard (ó cable de ethernet).
- Pinzas de punta y de corte.
- 2 DIP Switch de 8 posiciones.
- Resistencias de 330 ohms (ó 220 ohms)
- Displays de 7 segmentos

Actividades a realizar

12. El primer paso es determinar el número de compuertas, sumadores y bits de salida del multiplicador a realizar.

- Bits del multiplicador x bits del multiplicando: _____
- Número de compuertas AND: _____
- Número de sumadores paralelos: _____
- Número de bits de cada sumador paralelo: _____
- Número de bits del resultado: _____
- Número de displays de 7 segmentos necesarios: _____

13. El orden y la organización es muy importante para realizar esta práctica. El resultado de la multiplicación debe ser desplegado en displays de 7 segmentos por lo que se necesitará conectar un ***convertidor de binario a BCD*** entre el resultado de la multiplicación y los displays de 7 segmentos. Este convertidor se realizó en la práctica #6, pero se deberá de adaptar al número de bits del resultado del multiplicador.

14. Necesitará también tener a la mano el modulo ***convertidor de BCD a 7 segmentos,*** este funcionará tal y cual se realizó en la práctica # 3.

15. Deberá realizar también un ***sumador paralelo*** del número de bits necesario para el multiplicador que se le asignó.

16. El primer paso para realizar el multiplicador es realizar la multiplicación indicando cada uno de los productos parciales, como se muestra en el ejemplo en la **figura 1**. Realice la multiplicación para su multiplicador en el siguiente espacio:

Fig. 3 Multiplicación de ***J x K*** bits

17. Basándose en la **figura 3**, realice en un módulo Verilog las interconexiones necesarias entre las entradas, las compuertas AND y los sumadores paralelos. Este módulo será su **multiplicador**. Antes de continuar conectando el ***convertidor de binario a BCD*** y el ***convertidor de BCD a 7 segmentos***, realice la simulación del módulo del **multiplicador** solamente y asegúrese que funciona correctamente.

18. Una vez hecho esto, realice la interconexión de todos los módulos en el diagrama a bloques. La **figura 4** muestra como ejemplo el diagrama a bloques para un multiplicador de 8 x 8.

19. Implemente el circuito en su tarjeta y asegúrese de que funciona correctamente.

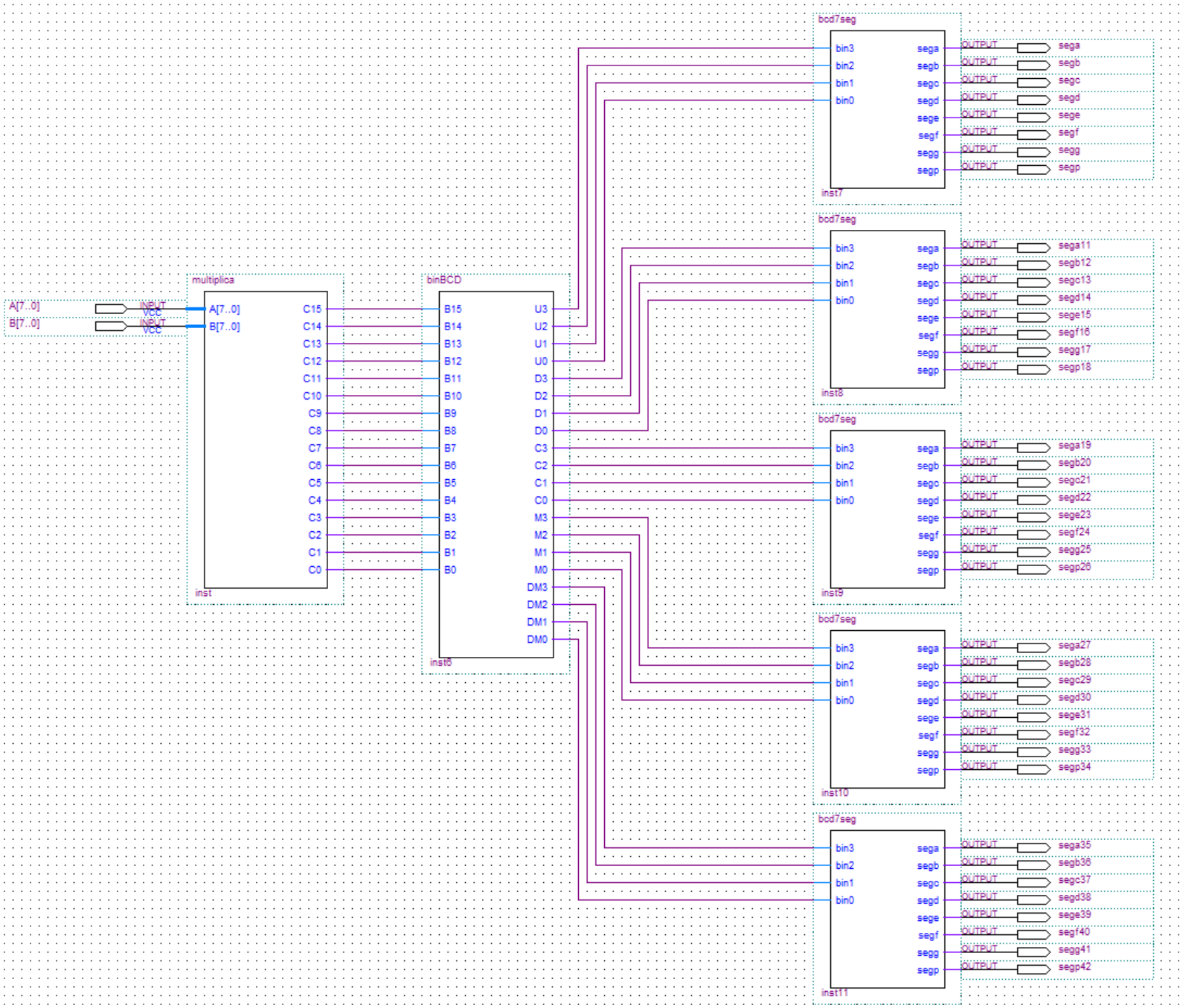


Fig. 4 Multiplicador de 8 x 8 bits

Bibliografía

- Tocci, Ronald J; Widmer, Neal S. (2007). Sistemas digitales : principios y aplicaciones. México: Pearson educación.

  http://catalogocimarron.uabc.mx/cgi-bin/koha/opac-detail.pl?biblionumber=149039

- Wakerly, John F. (2001). Diseño digital : principios y prácticas. México: Pearson educación.

  http://catalogocimarron.uabc.mx/cgi-bin/koha/opac-detail.pl?biblionumber=149421

- Mano, M. Morris; Ciletti, Michael D. (2013). Diseño digital : con una introducción a Verilog HDL. México: Pearson Educación de México.

  http://catalogocimarron.uabc.mx/cgi-bin/koha/opacdetail.pl?biblionumber=187491&query_desc=kw%2Cwrdl%3A%20digital%20design