

OBJETIVOS

Los objetivos que el estudiante debe alcanzar al final de esta práctica se resumen en los siguientes puntos:

1. Conocer el funcionamiento del simulador de circuitos digitales.
2. Ser capaz de usar el simulador de circuitos digitales para implementar (construir) circuitos combinatoriales sencillos.

MATERIAL

Para la realización de la práctica se dispondrá de un ordenador PC, con los sistemas operativos LINUX y/o Windows, y el entorno de diseño Xilinx ISE, que es la herramienta de diseño y síntesis basados en FPGAs de Xilinx. La práctica se debe realizar arrancando el ordenador en el sistema operativo Ms Windows.

INICIO DE LA PRÁCTICA

- A) Arrancar la máquina en modo Windows. Para ello al arrancar el ordenador cuando aparezca la palabra LILO teclear windows y pulsar ENTER. Utilizar el login "*usuario*" y el password "*practicas*".
- B) Hacer click sobre el icono "Project Navigator" del Escritorio para el entorno.

I. ENTORNO XILINX-ISE

La herramienta Xilinx-ISE (Integrated Software Environment) es una herramienta de diseño de circuitos profesional que nos va a permitir, entre otras funciones, la realización de esquemáticos y su posterior simulación

La herramienta consta de dos partes:

- **Project Navigator:** donde se realizará el diseño del circuito, bien mediante un esquemático o utilizando un lenguaje específico de diseño.
- **ModelSim:** donde podrá realizarse la simulación del funcionamiento del circuito y de este modo comprobar si funciona según las especificaciones establecidas

Se describirá a continuación el funcionamiento del programa, y para ello se realizará como ejemplo el diseño de un circuito lógico que cumpla la siguiente tabla de verdad:

NUM	A	B	C	D	Z
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	0

Simplificando obtenemos la expresión $Z = \bar{B} \cdot \bar{D} + A \cdot C \cdot \bar{D} + \bar{A} \cdot B \cdot C \cdot D$ para tener más detalles de la simplificación revidad el apéndice IV.

Para implementar este circuito, por tanto, necesitaremos:

- 1 puerta OR de 3 entradas,
- 1 puerta AND de 2 entradas,
- 1 puerta AND de 3 entradas,
- 1 puerta AND de 4 entradas,
- 3 inversores para generar las líneas negadas de A, B y D.

Este programa de diseño de circuitos, trabaja bajo entorno WINDOWS, por lo que primero habrá que entrar en este sistema. Una vez iniciado Windows buscamos el icono correspondiente al programa Project Navigator (Figura 1-a) y lo ejecutamos pulsando dos veces sobre él con el ratón usando el botón izquierdo.

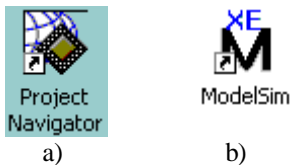


Figura 1. Iconos de las aplicaciones del entorno Xilinx ISE.

1. Creación de un nuevo proyecto

Un proyecto es un conjunto de ficheros de diseño, tales como esquemas de circuitos (a veces llamados esquemáticos), líneas de código de programas (si se ha realizado el diseño utilizando un lenguaje de programación específico de diseño de circuitos HDL), listas de conexiones, librerías de componentes, vectores de prueba para la simulación, etc., seleccionados para un diseño específico.

Nada más acceder al programa, aparecerá por pantalla una ventana como la que se muestra a la derecha, y que da acceso al programa de diseño.

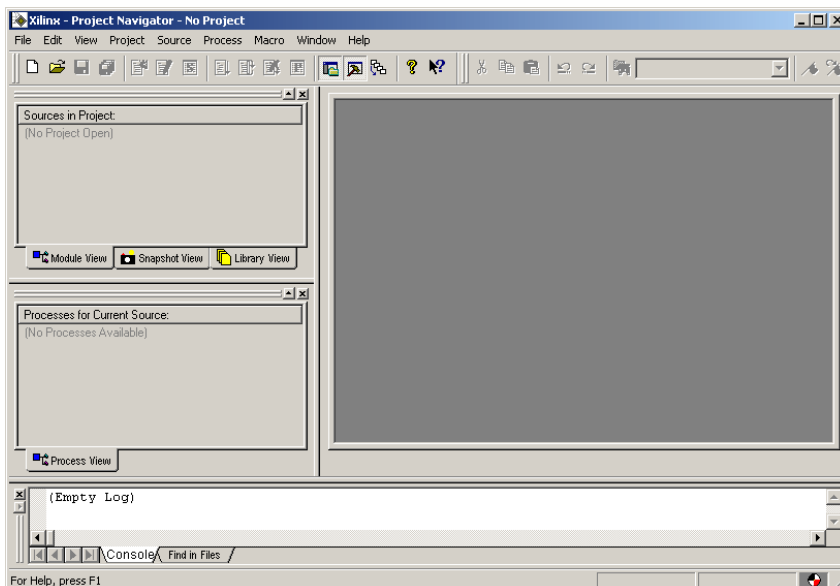


Figura 2.

Procedemos a crear un nuevo proyecto:

- Seleccionamos File ? New Project
- En la ventana de diálogo de New Project indicaremos el directorio de ubicación del proyecto en Project Location. Dado que a la versión que vamos a utilizar no le gustan los nombres largos usaremos una carpeta con un nombre corto justo en la principal del disco. (ej: c:\is09proy)
- Añadir el nombre en Project Name, “tutorial” en nuestro ejemplo.
- Automáticamente se crea un subdirectorio en la ruta indicada en Project Location con el nombre del proyecto, en este caso “tutorial”, y donde se almacenará todo lo relacionado a este proyecto
- Usar las flechas de desplazamiento para añadir el valor adecuado en los campos correspondientes a Project Device Options (recordar que el entorno sirve para implementar los circuitos en FPGA y por tanto debemos elegir el dispositivo final):

- Device name: Virtex
- Device: XCV50- 6bg256
- Design Flow: XST VHDL

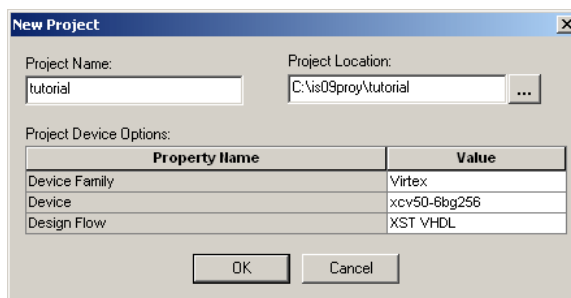


Figura 3

Al presionar el botón “OK” ISE crea y muestra el nuevo proyecto en el Project Navigator. Se observarán cambios con respecto al aspecto inicial de la ventana en la parte izquierda, en Sources in Project, donde aparece el proyecto creado.

El paso siguiente será el diseño del circuito que responda a la tabla de verdad indicada al comienzo de la sección.

2. Realización del esquema del circuito.

Los pasos son los siguientes

- Seleccionar Project ? New Source
- Seleccionar Schematic como tipo de fuente
- En File Name le damos el nombre al diseño, por ejemplo “circuito”
- Se presiona Siguiente y luego Se presiona Finalizar para concluir (Figura 4)

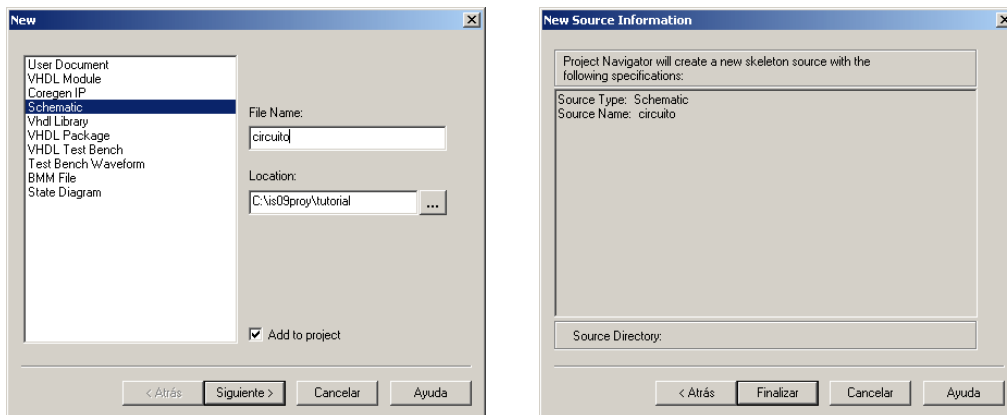


Figura 4.

Se abre entonces la ventana correspondiente al editor de esquemáticos, Xilinx ECS (Engineering Schematic Capture) (Figura 5), donde aparece, en la parte superior izquierda y entre paréntesis el nombre que hemos dado al esquema: “circuito”.

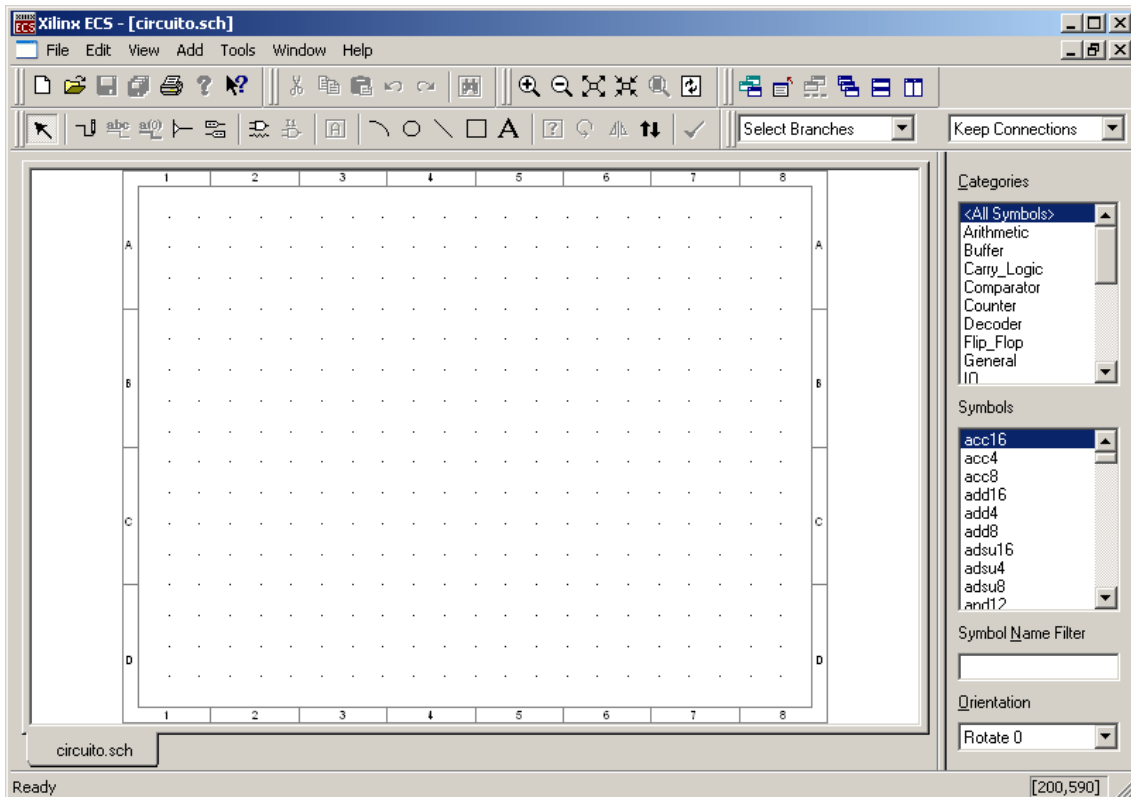


Figura 5.

2.1. Componentes y su Colocación


Lo primero será fijar las puertas necesarias para la realización del circuito. En la parte derecha del editor de esquemáticos (Figura 5) puede verse una ventana llamada Categories, donde se muestra una clasificación de las distintas categorías de componentes. Seleccionando cualquiera de ellas puede verse en la ventana inmediatamente inferior, Symbols, las puertas y componentes que forman la categoría seleccionada. Si se selecciona <All Symbols> podrá tenerse un listado de todos los componentes pertenecientes a la librería de componentes.

Para poder obtener las puertas lógicas necesarias, habrá que seleccionar <All Symbols> o Logic en la ventana de Categories, y una vez en Symbols, coger:

- and2: puerta AND de 2 entradas
- and3: puerta AND de 3 entradas
- and4: puerta AND de 3 entradas
- or3: puerta OR de 3 entradas
- inv: inversor (se necesitarán 3)

Si se desea **rotar** un componente puede hacerse empleando la ventana inferior derecha del editor de esquemáticos, Orientation, y utilizando el menú desplegable elegir la orientación deseada. Para **eliminar** un componente del editor de esquemáticos simplemente hay que seleccionarlo y pulsar la tecla Supr. Así mismo, para **mover** un componente bastará con seleccionarlo y arrastrarlo hasta el lugar deseado mientras se mantiene pulsado el botón izquierdo del ratón.

Tras la elección de las puertas necesarias, el editor de esquemáticos muestra la apariencia de la Figura 6.

Se recomienda salvar el esquemático a menudo, para evitar posibles problemas. Ello puede hacerse desde File ? Save As o pulsando el icono del disquete () en la barra de herramientas.

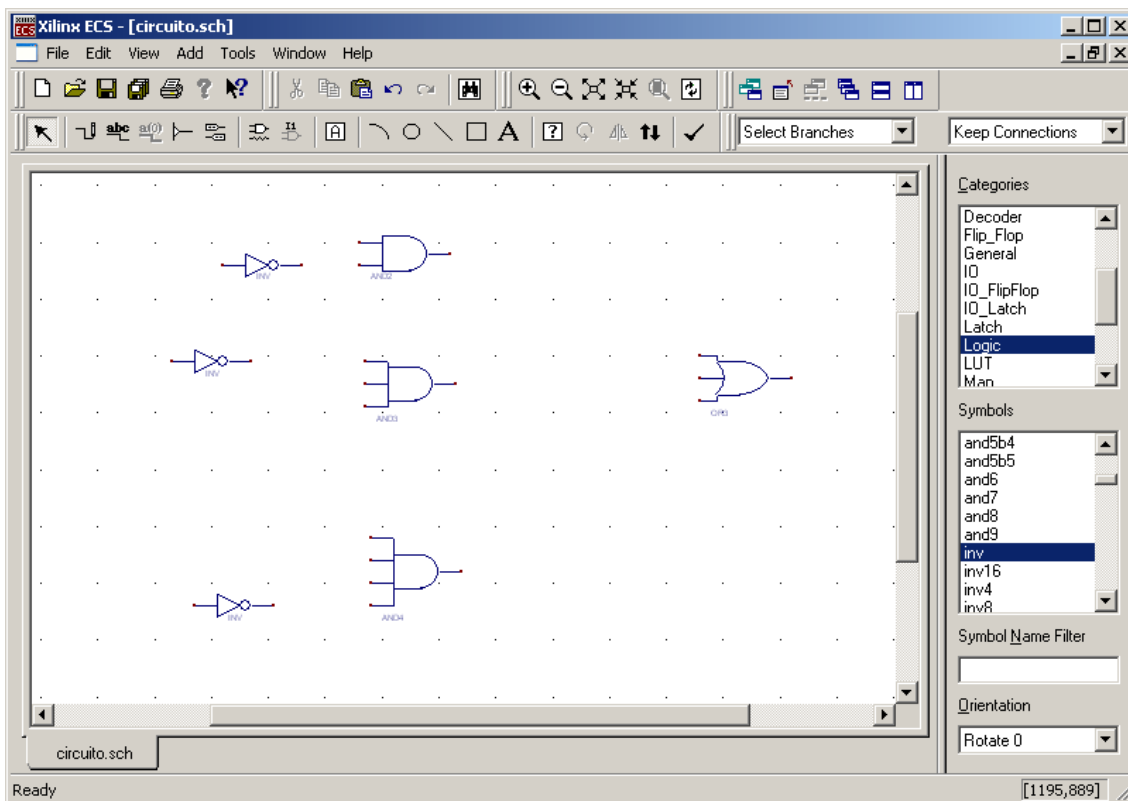



Figura 6.

2.2. Conexión de los elementos.

El siguiente paso será realizar las uniones entre las distintas puertas. Ello puede hacerse desde el menú principal del editor de esquemáticos con Add ? Wire, o pulsando el símbolo . El cursor se transformará en una cruz. Tendremos que pulsar con el botón izquierdo del ratón los dos extremos para que queden unidos por un cable. Si queremos unir una entrada o salida de un componente y dejar el otro lado del

cable “al aire” tendremos que hacer un doble click con el botón izquierdo del ratón para indicar que el extremo final va suelto.

El aspecto que ofrece el circuito en este momento, y tras hacer un zoom para tener una mejor visión, es el mostrado en la Figura 7. Se añade, para mejor comprensión, y utilizando letras auxiliares (que no forman parte del diseño) a qué entrada o salida corresponde cada cable.

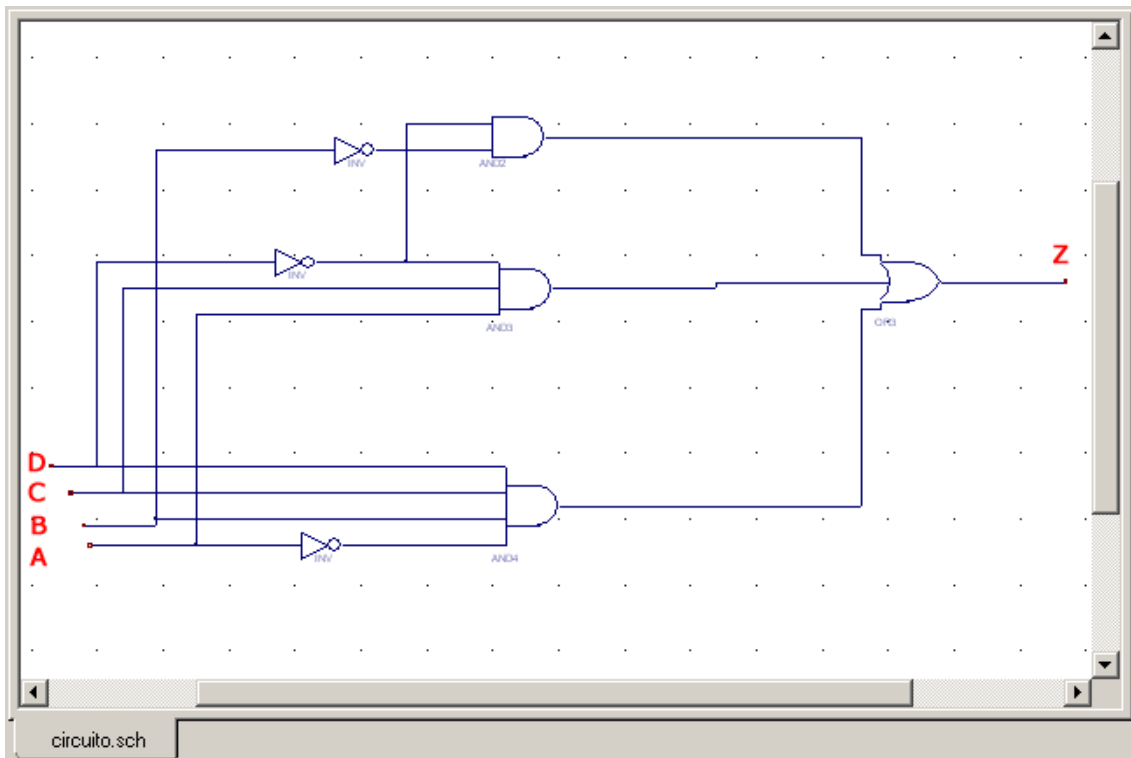


Figura 7.

2.3. Añadir nombres a los Cables.

Llega el momento de indicar el nombre de cada uno de los cables, para saber así qué señal están llevando u obteniendo de las puertas. Esto puede realizarse desde el menú principal del editor de esquemáticos: Add ? Net Name, o pulsando el símbolo . El cursor se transformará en una cruz.

En el menú principal habrá que añadir el nombre del cable en la ventana a la derecha de Name Branch. Comenzaremos por “A”:



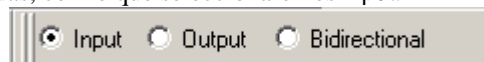
En el cursor, que se había transformado en una cruz, se añade ahora la letra A. Habrá que arrastrar el cursor hasta el punto final del cable correspondiente a la señal A y que habíamos dejado libre y pinchar con el botón izquierdo en ese punto para añadir el nombre.

A continuación se repite la operación para las otras señales B, C, D y Z.

2.4. Terminales de entrada/salida.

El siguiente paso sería identificar la dirección de cada señal, esto es, si corresponde a una entrada o a una salida. Para añadir terminales de entrada/salida: Add ? I/O Marker o bien pulsar el icono .

En el menú principal hay que indicar si es un terminal de entrada, de salida o bidireccional. Comenzaremos con las entradas, con lo que seleccionaremos input:



El cursor se transforma en un símbolo de terminal de entrada que habrá que arrastrar hasta el punto final de alguna de las entradas, que hemos nombrado previamente, y pinchar con el botón izquierdo del ratón. Veremos que el símbolo de entrada engloba a la letra correspondiente, fijándola como terminal de entrada:



Tras repetir la operación para el resto de entradas y para la salida (seleccionando previamente Output en el menú principal) el esquema ya está terminado. El aspecto es el que aparece en la Figura 8.

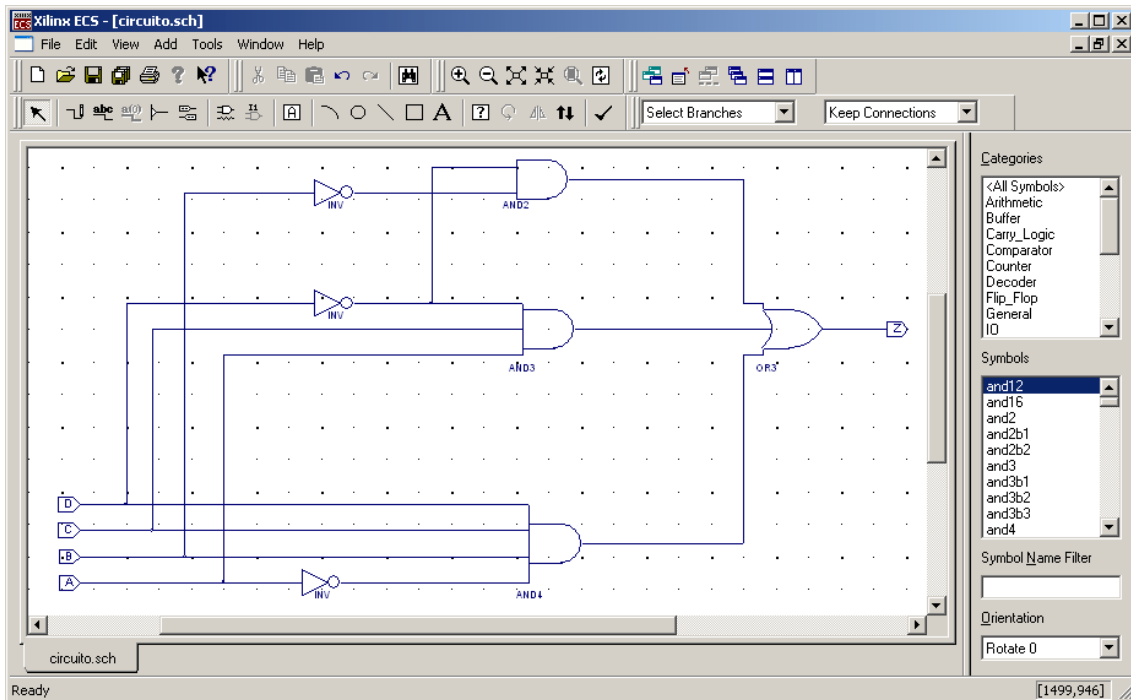



Figura 8.

2.5. Comprobación del circuito.

Una vez terminado el diseño del circuito puede hacerse una simple comprobación sobre el mismo, para asegurarnos de que todas las conexiones están hechas correctamente y que no quedan cables sueltos o puertas sin conectar. Para ello se ejecuta, desde el menú principal Tools? Check Schematic, o bien mediante el icono . Aparecerá una ventana que nos informará sobre si hay algún error en el circuito. Si todo está correcto la ventana será la mostrada en la Figura 9.

Una vez comprobado que no hay errores guardamos de nuevo el circuito y podemos cerrar la ventana del editor de esquemas (Xilinx ECS) y volver a la del navegador de proyectos (Project Navigator)

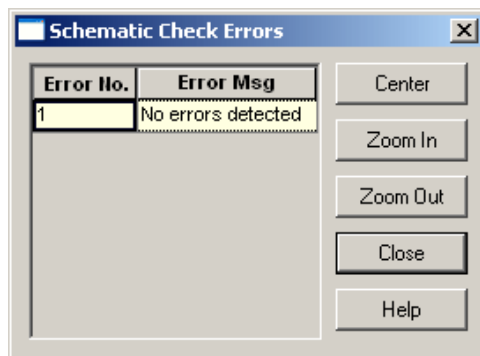


Figura 9.

Si queremos reutilizar este circuito como parte de otros mayores se puede crear un nuevo símbolo que lo englobe, con sus entradas y salidas. Para ver como hacer esto lee el apéndice 2.

3. Simulación.

Una vez que se tiene el circuito listo, revisado y libre de errores, será necesario comprobar su correcto funcionamiento, y para ello habrá que realizar la simulación del circuito.

3.1. Creación de un banco de pruebas.

Lo primero será crear un “banco de pruebas”, donde introduciremos los valores que se asignarán a las entradas del circuito y en función de los cuales se obtendrá una salida que habrá que ver si coincide con la esperada.

Los pasos a seguir son los siguientes:

Seleccionar el circuito (circuito.sch) en la ventana Sources in Project

- Seleccionar Project? New Source
- En la ventana de diálogo que aparece (Figura 10), seleccionar Test Bench Waveform
- Asignarle un nombre, por ejemplo “circuito_tbw” (tbw son las iniciales de Test Bench Waveform, para recordar qué es lo que se está haciendo)
- Presionar Siguiente, Siguiente y Finalizar

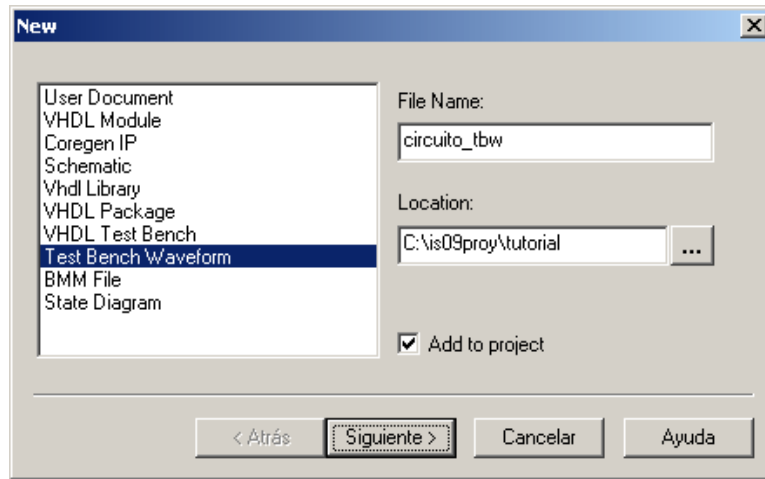


Figura 10.

Realizados los pasos anteriores, se inicia el banco de pruebas y el programa queda listo para introducir los datos y condiciones de la prueba.

En la ventana Initialize Timing hay que especificar los parámetros de tiempos que se usarán en la simulación (Figura 11)

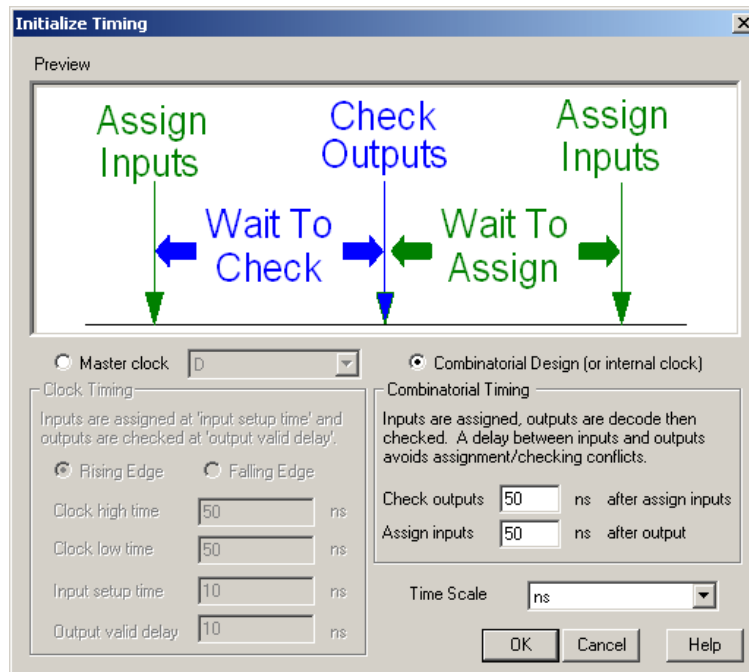


Figura 11.

Con el reloj (clock) a nivel bajo (cero lógico) y a nivel alto (uno lógico) se define el periodo de operación de cada señal. La entrada Input setup time define cuándo son válidas las entradas. La salida Output valid delay, define el tiempo después de la activación del reloj, tras el que las salidas son válidas.

Para este ejemplo no cambiaremos ninguna de las especificaciones por defecto. Estas especificaciones son:

- Clock high time: 50 ns
- Clock low time: 50 ns
- Input setup time: 10 ns
- Output valid delay: 10 ns

En la ventana Initialize Timing presionaremos OK para aceptar las especificaciones por defecto. Aparecerá entonces la ventana correspondiente al banco de pruebas que se utilizará para la simulación (Figura 12)

Para los circuitos secuenciales en los que necesitaremos una señal de reloj deberemos seleccionar la señal que actuará como tal en Master Clock y fijar las condiciones en la parte izquierda de la ventana (la que ahora, con los combinacionales, está desactivada)

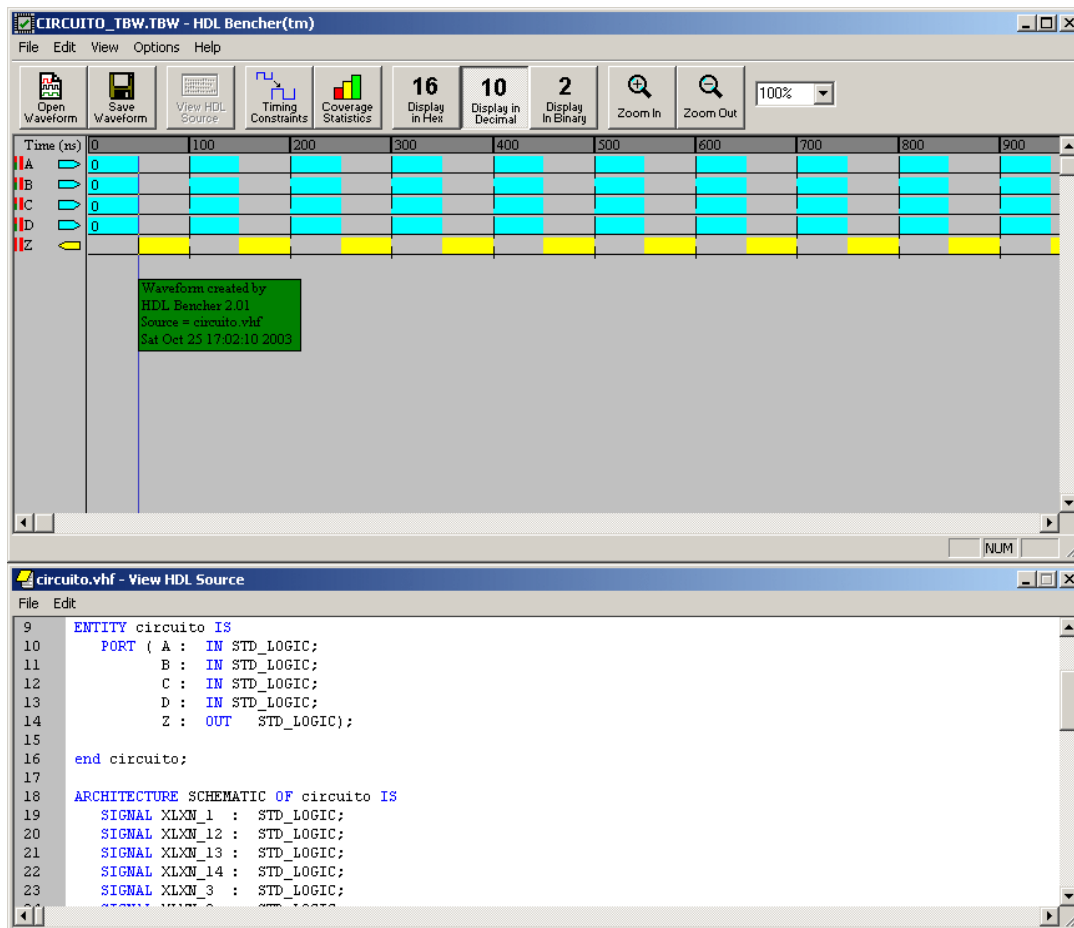


Figura 12.

En la parte superior de la ventana del banco de pruebas (Figura 12) aparece el banco de pruebas como tal, donde habrá que introducir los estímulos (ceros o unos) correspondientes a las entradas para poder reproducir la tabla de verdad del circuito diseñado. También aparecen, en la parte izquierda, las señales de las cuatro entradas y la salida. Inicialmente, todas las entradas se encuentran a cero.

En la parte inferior de la ventana aparece la transcripción, en lenguaje HDL, del circuito creado en el editor de esquemáticos.

En la ventana de formas de onda, en el banco de pruebas, es necesario inicializar las entradas con los valores adecuados para poder comprobar el correcto funcionamiento del circuito. Los estímulos (ceros o unos) se introducirán en la zona azul de cada celda.

Según la tabla de verdad, la entrada A permanece 8 estados a “0” y otros 8 a “1”, esto es, permanece 8 ciclos de reloj a nivel bajo y 8 a nivel alto. La entrada B cambia de 0 a 1 cada 4 ciclos de reloj, la entrada C realiza el cambio cada 2 ciclos de reloj y, por último, la entrada D cambia continuamente de 0 a 1. En la ventana del banco de pruebas (Figura 12) iremos introduciendo los valores (0 o 1) para reproducir la tabla de verdad. Los cambios se reflejarán para cada intervalo de tiempo en la parte superior de la ventana (Time (ns), parte gris).

Los valores (0 o 1) se introducen en los rectángulos azules (pinchando con el botón izquierdo del ratón) que aparecen para cada intervalo de tiempo y en cada una de las variables:

- Señal A: 0 en 0 ns; 1 en 800 ns
- Señal B: 0 en 0 ns; 1 en 400 ns; 0 en 800 ns; 1 en 1200 ns
- Señal C: 0 en 0 ns; 1 en 200 ns; 0 en 400 ns; 1 en 600 ns; 0 en 800 ns ...
- Señal D: 0 en 0 ns; 1 en 100 ns; 0 en 200 ns; 1 en 300 ns; 0 en 400 ns...

La salida se completará en 1500 ns. Pueden utilizarse los botones Zoom In, Zoom Out o seleccionar en el botón desplegable el porcentaje de ventana mostrado para poder ver todas las señales introducidas

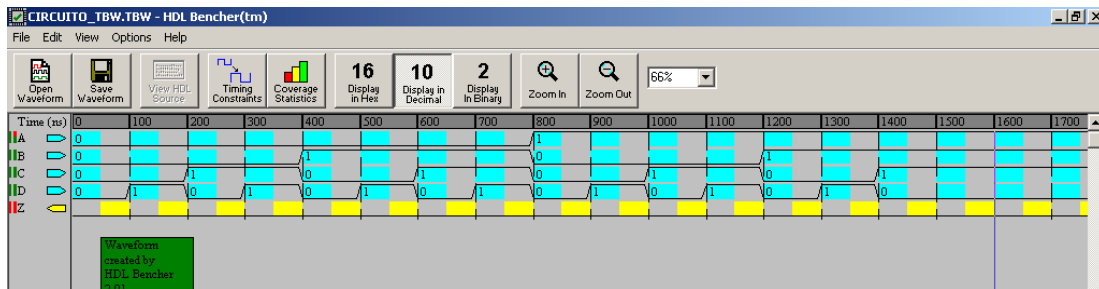


Figura 13.

El banco de pruebas así creado se salva desde el menú principal: File ? Save Waveform o pinchando en el icono del disquete. Al realizar la operación de guardado, aparece un cuadro de diálogo donde el banco de pruebas nos permite establecer los ciclos de reloj que quieren simularse. En este caso se necesitan 8 ciclos de reloj para visualizar el comportamiento completo del circuito, así que se introducirá el valor “8”:

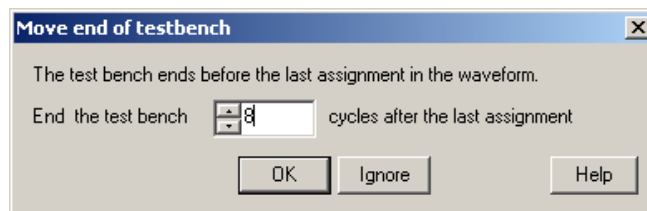


Figura 14.

Se pulsa OK al terminar, y después se puede salir del banco de pruebas: File ? Exit

3.2. Generación de los Resultados.

Los pasos a seguir son los siguientes:

- Seleccionar circuito_tbw.tbw en la ventana Sources in Project del Project Navigator
- En la ventana Processes for Current Source, pinchar en el símbolo + junto a ModelSim Simulator para expandir la jerarquía (Figura 15)
- Doble click en Generate Expected Simulation Results

Tras realizar los pasos anteriores, aparece la ventana del banco de pruebas con el resultado de la simulación. Puede verse los valores que va tomando la salida Z en función de las entradas (Figura 16)

Tras comparar los resultados con la tabla inicial puede salirse del HDL Bencher sin salvar las formas de onda.

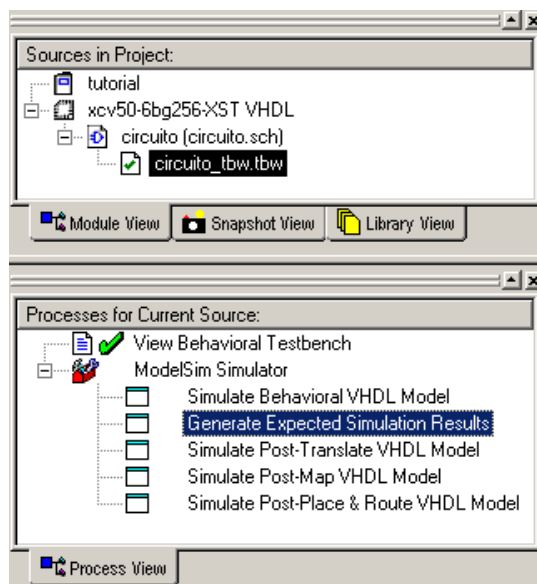


Figura 15.

La simulación realizada es una **simulación funcional**, donde simplemente se ven los valores de la salida en función de las entradas, pero sin tener en cuenta que el circuito está realizado con componentes reales que sufren retrasos que pueden afectar a las entradas.

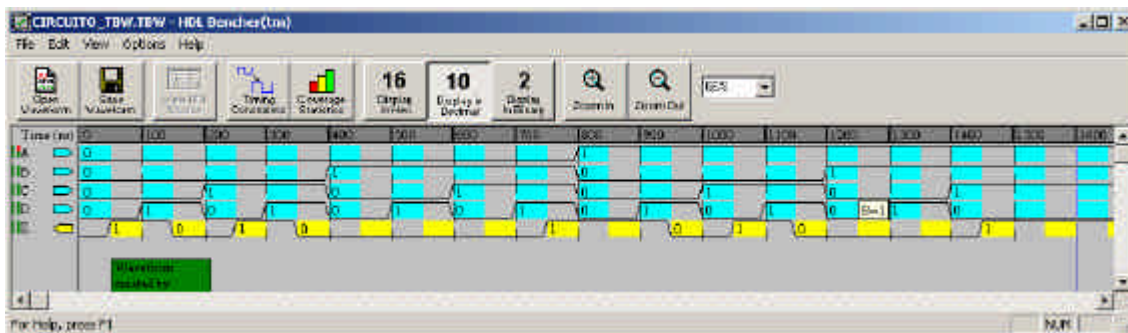


Figura 16.

También se puede realizar una **simulación del comportamiento** del circuito (behavioral), utilizando la herramienta **ModelSim** para ver una salida real que produce el circuito diseñado.

3.3. Simulación del comportamiento del Circuito.

Los pasos a seguir son muy parecidos al caso anterior pero cambiando Generate Expected Simulation Results por Simulate Behavioral VHDL Model:

- En Project Navigator, seleccionar circuito_tbw.tbw en la ventana Sources in Project
- En la ventana Processes for Current Source, desplegar el menú de jerarquía con + y hacer doble click en Simulate Behavioral VHDL Model. En ese momento se carga el programa ModelSim

Al arrancar el programa ModelSim aparece la ventana ModelSim XE Starter:

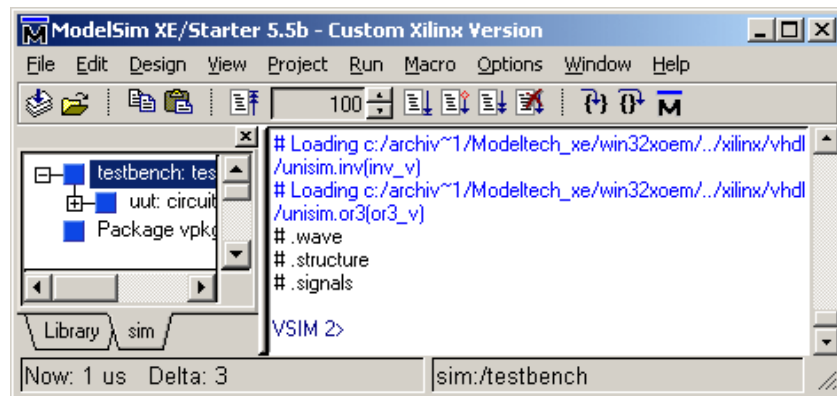
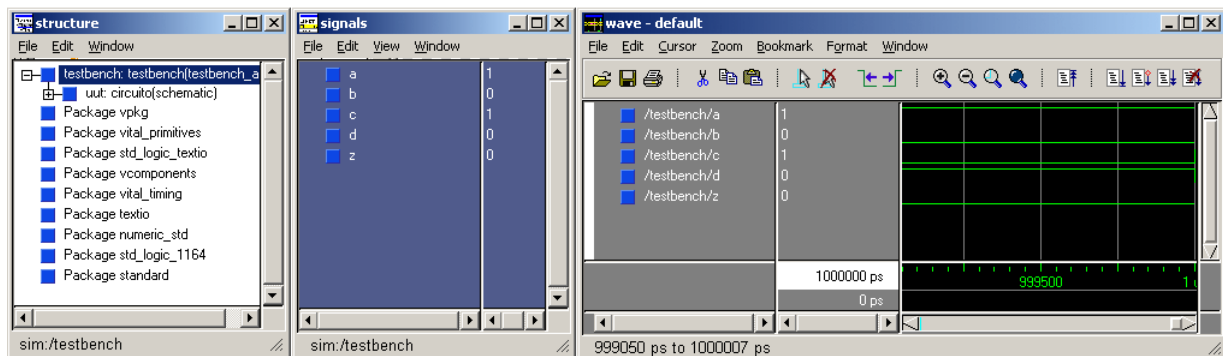


Figura 17.

Y tras ella, al ejecutarse la simulación del comportamiento aparecen las siguientes:



V. Structure

V. Signals

Ventana Wave

Figura 18.

Los resultados de la simulación del comportamiento del circuito se mostrarán en la ventana de ondas (Wave).

Nota: ISE automatiza el proceso de simulación creando y ejecutando un fichero macro de simulación (fichero .fdo), que no es visible para el usuario pero que realiza las siguientes funciones:

- Crea las librerías de diseño
- Compila el diseño y los ficheros del banco de pruebas
- Llama al simulador
- Abre todas las ventanas relacionadas con la simulación
- Añade todas las señales a la ventana de ondas (Wave window)
- Añade todas las señales a la ventana de señales (Signal window)
- Corre la simulación para el tiempo especificado en Run Time Property (por defecto, 1000 ns). Para cambiarlo deberemos hacerlo en la ventana de Project Navigator, con el archivo de pruebas seleccionado, en el menú Process → Properties → Pestaña Simulation Properties cambiaremos el valor de Simulation Run Time al valor adecuado (en nuestro caso 1600ns)

En la ventana Wave-default, habrá que hacer un zoom para poder visualizar la salida del circuito. Presionar Zoom ? Zoom Full y Zoom ? Zoom In.

La forma de ondas del circuito será de la forma mostrada en la Figura 19.

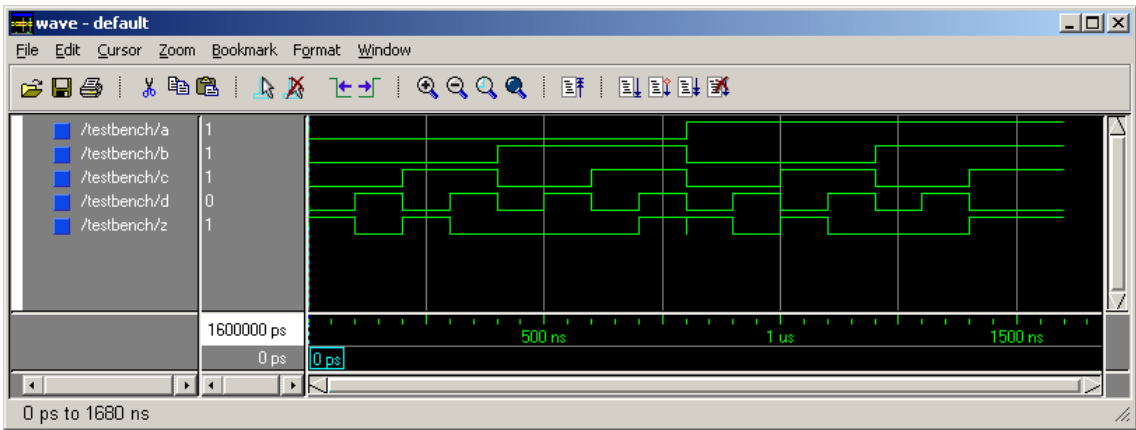


Figura 19.

Como puede apreciarse, aparece una inestabilidad en la salida para el instante de transición en 800ns. Esto es lo que nos ofrece una simulación del comportamiento que la primera no nos ofrecía.

Para una comprobación más fácil del correcto funcionamiento del circuito y puesto que las entradas recorren todos los posibles estados (0, 1, 2... 7) sería bueno poder reunir las entradas individuales en una, bajo la forma de un vector y que así nos mostrara los estados por los que van pasando las combinaciones de las señales a las que representa. Para ello, seleccionamos las entradas del circuito en la ventana wavedefault pulsando el botón izquierdo del ratón mientras mantenemos pulsada la tecla de mayúsculas o de control de igual forma que se seleccionan los archivos en el explorador de Ms Windows, con lo que aparecerán en el interior de un recuadro blanco.

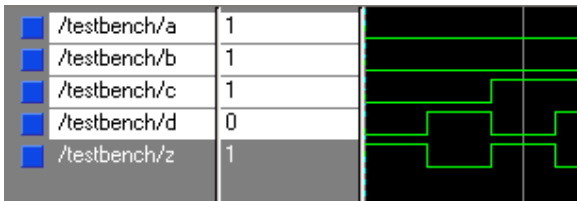


Figura 20.

Sobre las entradas seleccionadas pulsamos el botón derecho del ratón y accedemos a Combine. En el cuadro de diálogo que aparece hay que dar un nombre a la nueva señal combinada, por ejemplo “entradas”.

Pulsamos OK y vemos que en la ventana wave-default aparece una nueva señal /testbench/entradas. Si pulsamos en el símbolo + podemos ver la jerarquía, esto es, las señales que tiene por debajo y a las que representa.

Para visualizar los valores de esta señal combinada en decimal en vez de en binario como aparecen, habrá que seleccionar la entrada combinada, pulsar el botón derecho del ratón y en el menú de opciones elegir Radix? Decimal (realmente Unsigned es una mejor opción, pero se puede experimentar). Para mayor facilidad, podemos desplazar la señal de salida z inmediatamente debajo de la señal de entrada combinada, pinchando en la señal de salida y arrastrando hacia arriba la señal con el botón izquierdo del ratón pulsado. El aspecto de la simulación se muestra en la Figura 21.

Una vez comprobados los resultados se puede cerrar el simulador desde la ventana ModelSim XE Starter.

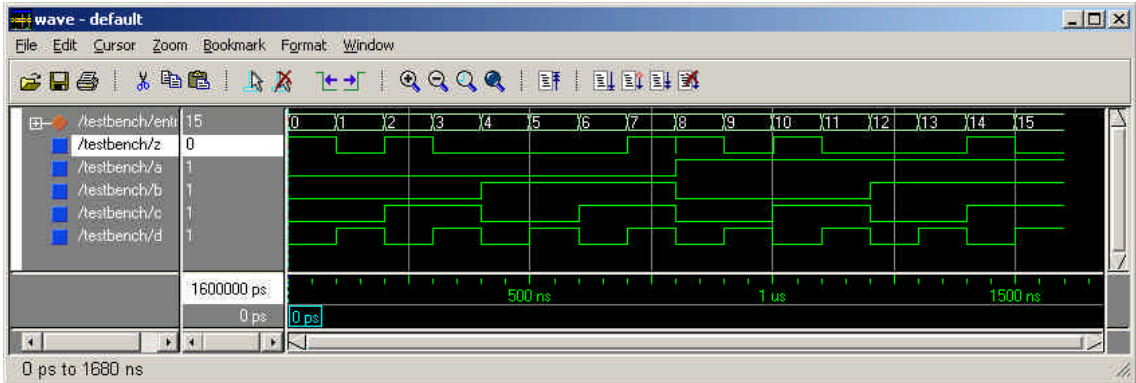
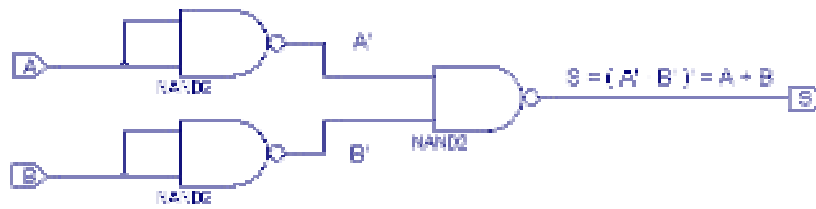


Figura 21.

II. EJERCICIOS DE IMPLEMENTACIÓN DE CIRCUITOS COMBINACIONALES.

EJERCICIO 1.- Implementa en el simulador el siguiente circuito y, siguiendo su funcionamiento, obtén la tabla de verdad correspondiente.

OR CON PUERTAS NAND



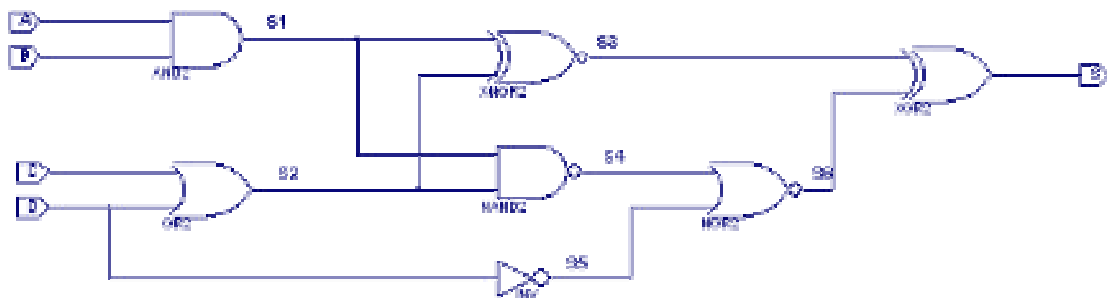
EJERCICIO 2.- Implementa en el simulador el siguiente circuito y, siguiendo su funcionamiento, obtén la tabla de verdad correspondiente.

AND CON PUERTAS NAND



EJERCICIO 3.- Implementa el ejercicio anterior utilizando únicamente puertas NOR (es decir, implementar ambas puertas NAND mediante puertas NOR).

EJERCICIO 4.- Implementa en el simulador el siguiente circuito y, siguiendo su funcionamiento, obtén la tabla de verdad correspondiente.



EJERCICIO 5.- Dada la siguiente tabla de verdad implementa un circuito que tenga dicha tabla utilizando tan sólo puertas de dos entradas.

X	Y	Z	Función
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

APÉNDICE 1. GUÍA RÁPIDA DE CREACIÓN Y SIMULACIÓN DE PROYECTOS

1. Creación del Proyecto. File → New Project, Fijar el nombre, la localización y el dispositivo FPGA.
 2. Introducir la descripción de los circuitos: Creación o Importación del esquema o descripción HDL
Introducir la nueva fuente (new source) en el proyecto. Dibujar o describir según sea necesario.
Para dibujar el esquema seguiremos los siguientes pasos:
 1. Colocar los símbolos de los elementos a utilizar.
 2. Cablear el circuito.
 3. Darle nombre a los extremos de los cables.
 4. Introducir los símbolos de entradas/salidas.
 5. Comprobar que no hay fallos.
- a) Crear los vectores de pruebas.
Fijar los tipos de las señales y temporización, los valores de las entradas y el límite de ciclos (el periodo de simulación)
- b) Simulación
Seleccionar el archivo de pruebas a utilizar en Sources in Project
- Simulación funcional: ejecutar ModelSim Simulator → Generate Expected Simulation Results.
 - Simulación de comportamiento: revisar el límite de tiempo y después ejecutar ModelSim Simulator → Simulate Behavioral VHDL Model.

APÉNDICE 2. CREACIÓN DE SÍMBOLOS A PARTIR DE ESQUEMAS PREVIOS.

Con el esquema realizado se puede hacer un nuevo componente consistente en una especie de “caja negra”, donde el usuario sólo verá un bloque con el mismo número de entradas y de salidas que el esquema original, que englobará a éste y realizará su misma función. Este símbolo que se crea pasa a formar parte de la librería de componentes del proyecto, y podrá utilizarse como un componente más.

Para crearlo: Tools? Create Symbol desde el menú principal. Aparece entonces la ventana de la Figura 22, donde indica el nombre que se le dará al símbolo, y que por defecto es el nombre del esquemático (puede cambiarse), y las entradas y salidas del mismo.

Pulsando OK el programa genera el símbolo y lo muestra (Figura 23).

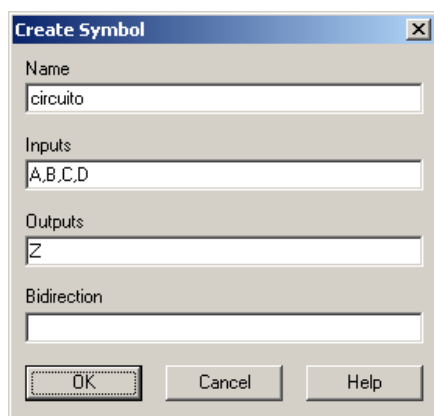


Figura 22.

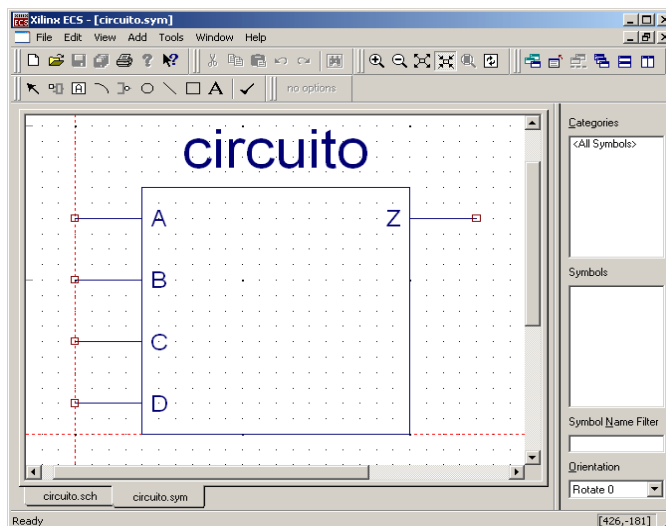


Figura 23.

Si nos fijamos en la parte inferior izquierda de la ventana de editor de esquemáticos, podemos ver que hay dos pestañas, circuito y circuito.sym, correspondientes a las dos hojas que tenemos abiertas en el editor de esquemáticos. Pulsando sobre circuito se accede al esquema original y sobre circuito.sym al símbolo que hemos creado del circuito.

En el esquema del circuito original, si ahora nos fijamos en la ventana de la parte derecha, Categories, vemos que hay una nueva categoría, <c:/is09proy/tutorial>, que corresponde a la librería del proyecto que hemos abierto, y donde ha almacenado el componente creado. De hecho, si seleccionamos esta categoría, podemos ver en la ventana inferior, Symbols, que ahí está el componente circuito creado, y que ahora puede utilizarse como un componente más a la hora de diseñar.

APÉNDICE 3. RESUMEN DEL ALGEBRA DE BOOLE

Un álgebra de Boole es un conjunto en el que:

- 1- Se han definido dos funciones binarias (que necesitan dos parámetros) que llamaremos aditiva (que representaremos por $x + y$) y multiplicativa (que representaremos por $x \cdot y$) y una función monaria (de un solo parámetro) que representaremos por x' .
- 2- Se han definido dos elementos (que designaremos por 0 y 1)
- 3- Tiene las siguientes propiedades:

Conmutativa, función '+'	$x + y = y + x$
Conmutativa, Función '·'	$x \cdot y = y \cdot x$
Asociativa, función '+'	$(x + y) + z = x + (y + z)$
Asociativa, Función '·'	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$
Distributiva, función '+'	$(x + y)z = x \cdot z + y \cdot z$
Distributiva, Función '·'	$(x \cdot y) + z = (x + z)(y + z)$
Identidad, función '+'	$x + 0 = x$
Identidad, Función '·'	$x \cdot 1 = x$
Complemento, función '+'	$x + x' = 1$
Complemento, Función '·'	$x \cdot x' = 0$

Propiedades del álgebra de Boole

Idempotente, función '+'	$x + x = x$
Idempotente, Función '·'	$x \cdot x = x$
Maximalidad del 1	$x + 1 = 1$
Minimalidad del 0	$x \cdot 0 = 0$
Involución	$x'' = x$
Inmersión, función '+'	$x + (x \cdot y) = x$
Inmersión, función '·'	$x \cdot (x + y) = x$
Ley de Morgan, función '+'	$(x + y)' = x' \cdot y'$
Ley de Morgan, función '·'	$(x \cdot y)' = x' + y'$

APÉNDICE 4. SIMPLIFICACIÓN. METODO DEL MAPA DE KARNAUGH.

Los mapas de Karnaugh constituyen un método sencillo y apropiado para la minimización de funciones lógicas. El tamaño del mapa depende del número de variables, y el método de minimización es efectivo para expresiones de hasta 6 variables.

Un mapa de Karnaugh es una representación gráfica de una tabla de verdad, y por lo tanto existe una asociación unívoca entre ambas. El proceso de minimización se basa en la forma en como se acomodan las celdas del mapa que representan cada una un minitérmino.

Nos ceñimos al caso planteado como ejemplo. Dada la tabla de verdad de la que queremos obtener la función simplificada planteamos una tabla con la estructura que sigue:

Tras agrupar siguiendo los conjuntos de la figura 1, vemos que del conjunto rojo podemos simplificar A y C (puesto que ocurren tanto en su forma normal como negada), que del azul podemos simplificar B y, finalmente, nos queda el término solitario rodeado por el círculo negro.

CD\AB	00	01	11	10		
00	1	0	0	1	C̄	D̄
01	0	0	0	0		
11	0	1	0	0	C	D
10	1	0	1	1		
	Ā		A			
	B̄		B		B̄	

Figura 24.

En conclusión así es como tenemos el resultado presentado en la página 1.

Es decir, que tenemos la expresión $Z = \bar{B} \cdot \bar{D} + A \cdot C \cdot \bar{D} + \bar{A} \cdot B \cdot C \cdot D$