

Tutorial Xilinx-ISE

Manejo básico



Susana Holgado
Escuela Politécnica Superior
UAM

Modificaciones:
Ángel de Castro (2006)
Francisco Javier Gómez Arribas (2008)
Víctor Apéstegui Palacio (2009)





1

Tutorial Xilinx-ISE

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

En este tutorial se va a utilizar concretamente:

-  **Entorno ISE:** 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 cumple con las especificaciones establecidas



2

Se describirá a continuación el funcionamiento del programa, y para ello se realizará 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



3

Haciendo la tabla de Karnaugh y las simplificaciones habituales, se obtiene:

$$Z = \overline{BD} + \overline{ABCD} + ACD\overline{D}$$

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 [Xilinx ISE](#) (Figura 1) y lo ejecutamos pulsando dos veces sobre él con el ratón usando el botón izquierdo



Fig. 1



4

1. Creación de un nuevo proyecto

Un proyecto es un conjunto de ficheros de diseño, tales como 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 conexionado, bibliotecas de componentes, vectores de test para la simulación, etc., seleccionados para un diseño específico

Nada más acceder al programa, aparecerá por pantalla una ventana como la que se muestra en la figura 2, y que da acceso al programa de diseño

Si hubiera ya algún proyecto cargado, se puede cerrar seleccionando la opción de menú **File → Close Project**



5

1. Creación de un nuevo proyecto

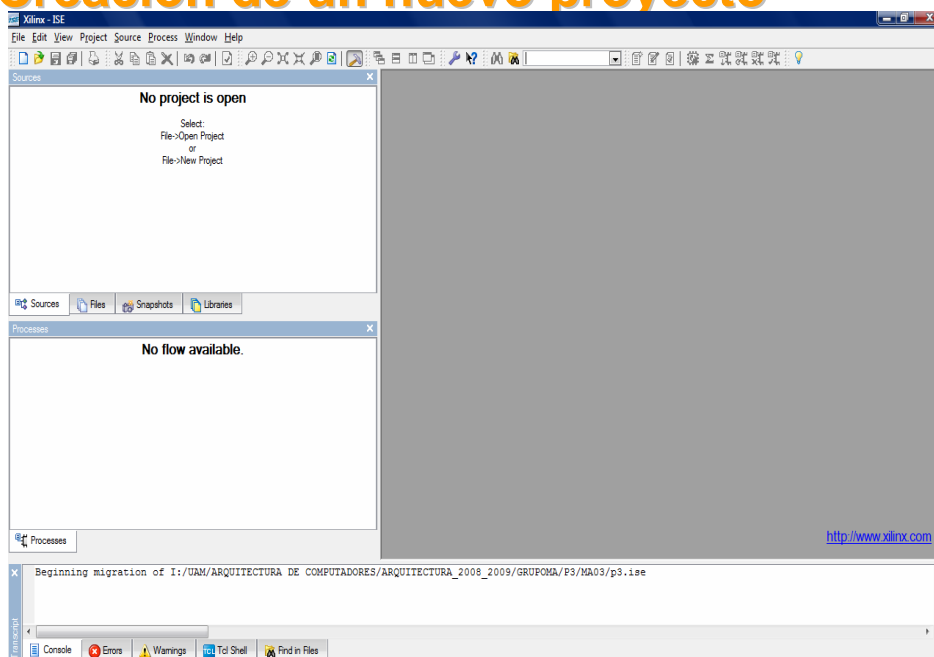


Fig. 2

6

1. Creación de un nuevo proyecto

Para crear un nuevo proyecto:

- Seleccionar **File** → **New Project**
- En la ventana de diálogo de New Project indicar el directorio de ubicación del proyecto en **Project Location**
- Añadir el nombre (p. ej. "tutorial") en **Project Name**
- En la opción **Top-Level Module Type** se elige "Schematic" si el diseño se va a realizar con esquemáticos o "HDL" si se va a utilizar un lenguaje de descripción del hardware como VHDL
- Automáticamente se crea un subdirectorio en la ruta indicada en Project Location con el nombre del proyecto (p. ej. "tutorial"), y donde se almacenará todo lo relacionado con este proyecto

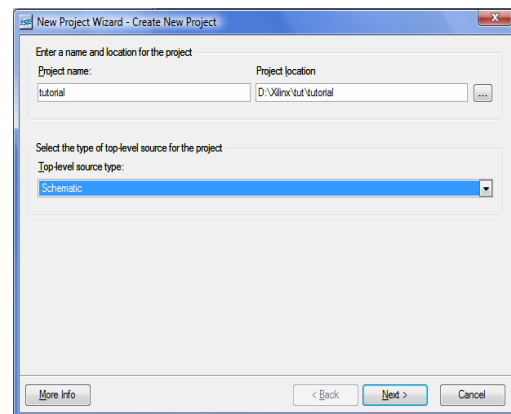


Fig. 3



7

1. Creación de un nuevo proyecto

En la segunda pantalla hay que seleccionar la FPGA sobre la que se va a sintetizar el diseño. Para este caso se seleccionará:

- **Device Family:** Spartan3 (familia)
- **Device:** xc3s200 (dispositivo dentro de la familia)
- **Package:** ft256 (encapsulado)
- **Speed grade:** -4 (indica la velocidad máxima del dispositivo)
- **Synthesis Tool:** XST (VHDL/Verilog)
- **Simulator:** Modelsim SE VHDL (herramienta de simulación)
- **Generated Simulation Language:** VHDL (lenguaje asociado para los ficheros que crea la herramienta)

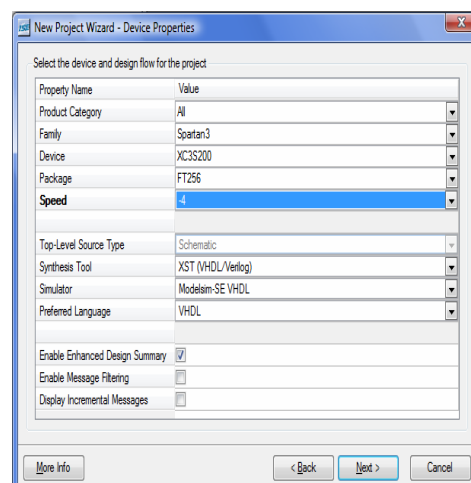


Fig. 4



8

1. Creación de un nuevo proyecto

Después se pasan las dos siguientes pantallas eligiendo **Siguiente** y al dar a **Finalizar**, ISE crea y muestra el nuevo proyecto.

Se observarán cambios con respecto al aspecto inicial de la ventana en la parte izquierda, en **Sources in Project**, donde aparece el proyecto creado, como se ve en la figura 5

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

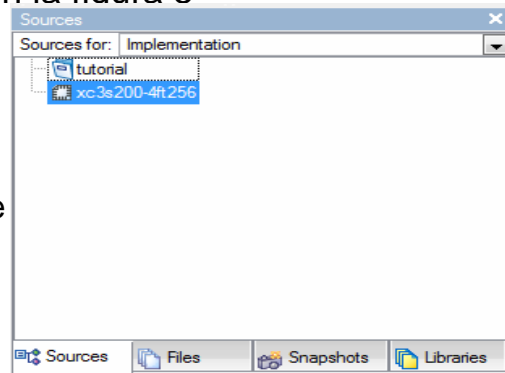


Fig. 5



9

2. Realización del esquemático del circuito

- 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**
- Se presiona **Finalizar** para concluir (Figura 6)

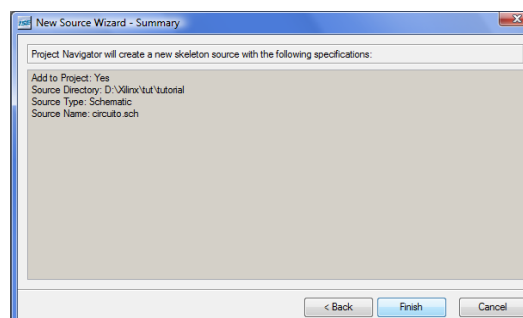
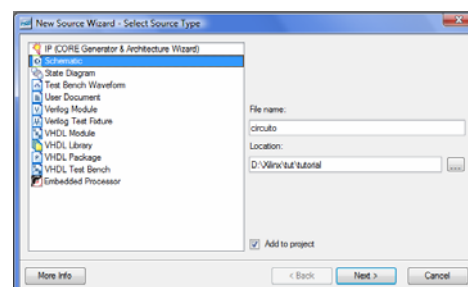


Fig. 6



10


2. Realización del esquemático del circuito

2.1. Componentes

Si se desea rotar un componente puede hacerse empleando la opción inferior del mismo menú, **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. Asimismo, 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 9

Se recomienda salvar el esquemático a menudo, para evitar posibles problemas. Ello puede hacerse desde **File** → **Save As** o pulsando el icono 



13

2. Realización del esquemático del circuito

2.1. Componentes

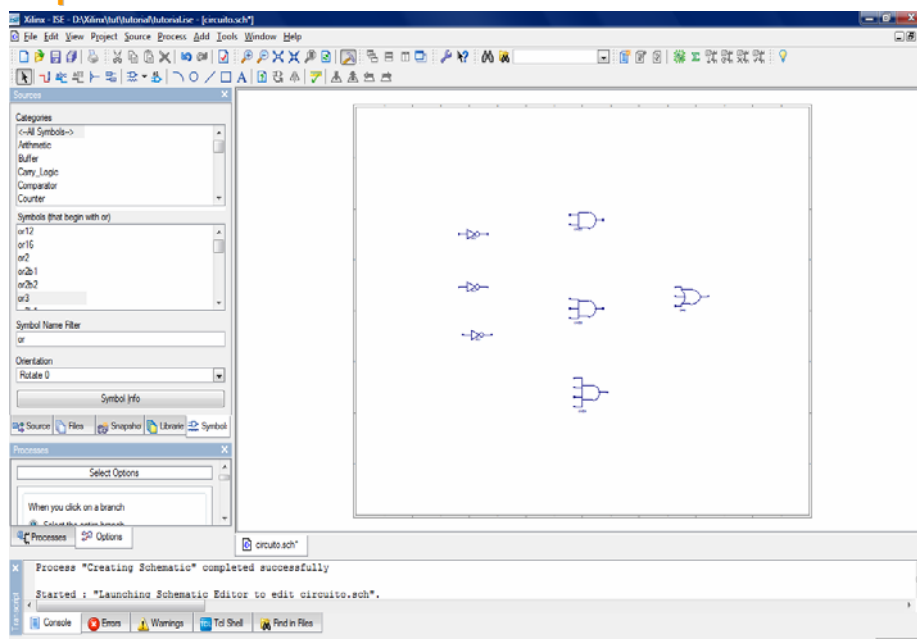



Fig. 9



14

2. Realización del esquemático del circuito

2.2. Conexionado

El siguiente paso será realizar las uniones entre las distintas puertas. Ello puede hacerse desde el menú principal del editor de esquemáticos: **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 10. Se añade, para mejor comprensión, y utilizando flechas auxiliares (que no forman parte del diseño) a qué entrada o salida corresponde cada cable



15

2. Realización del esquemático del circuito

2.2. Conexionado

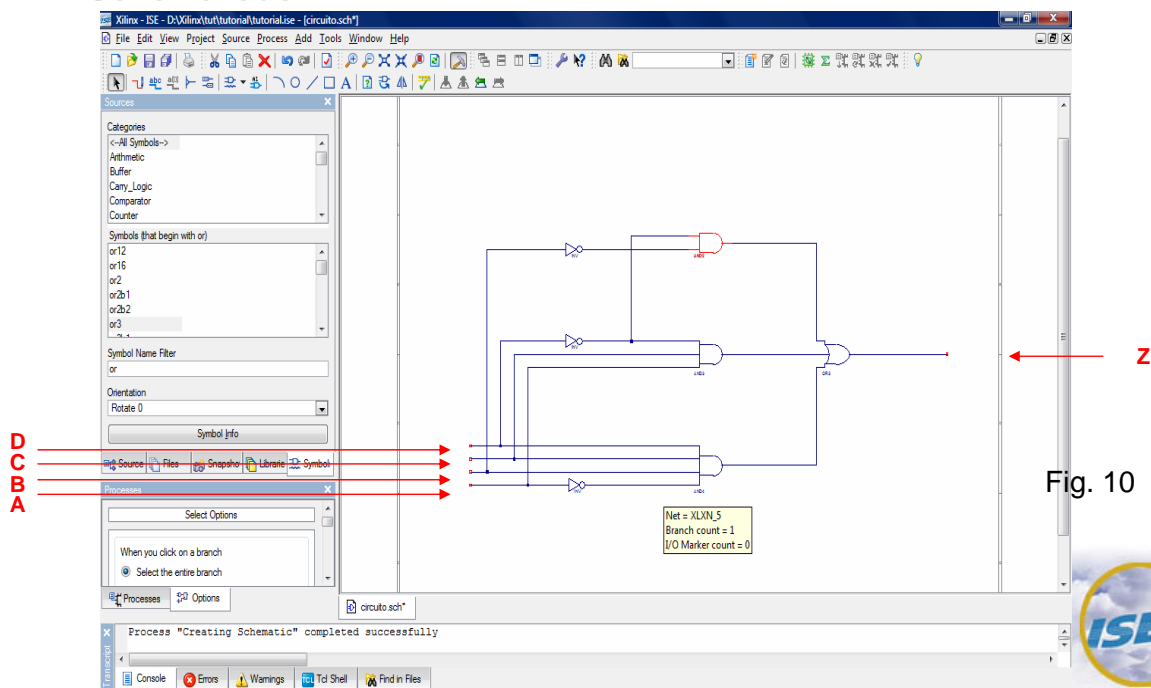


Fig. 10




16

2. Realización del esquemático del circuito

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 inferior izquierdo habrá que añadir el nombre del cable en el hueco de **Name**. Comenzaremos por "A" (ver figura 11a), y se seguirá con "B", "C", "D" y "Z"

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

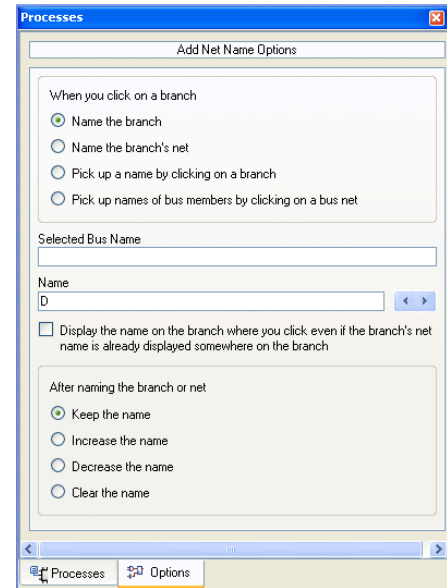


Fig. 11 a




17

2. Realización del esquemático del circuito

2.4. Terminales de entrada/salida

El siguiente paso será 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 inferior izquierdo (fig 11b) 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:

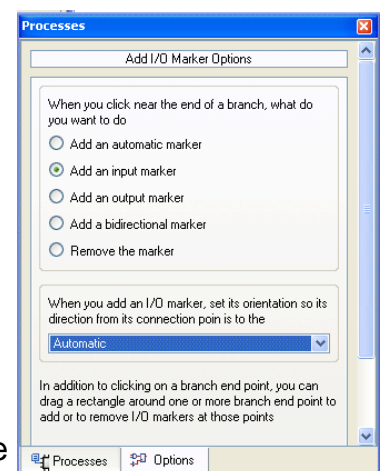


Fig. 11 b



18

2. Realización del esquemático del circuito

2.4. Terminales de entrada/salida

Tras repetir la operación para el resto de entradas y para la salida (seleccionando previamente **Output** en el mismo menú) el aspecto del esquemático del circuito se muestra en la figura 12

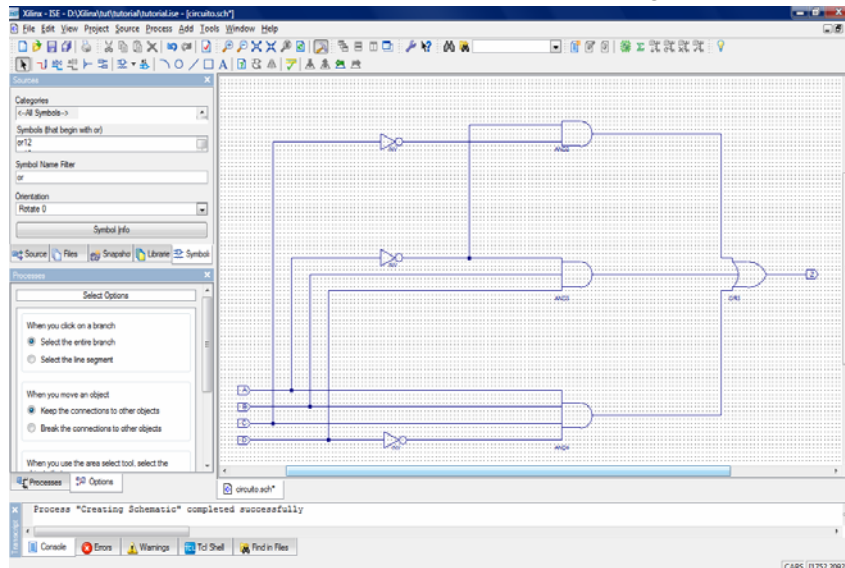


Fig. 12

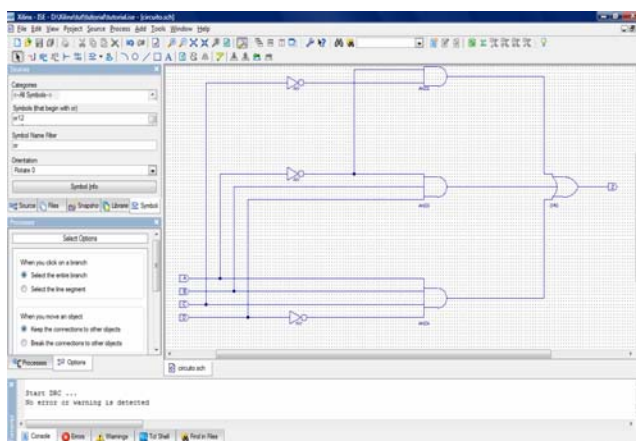



19

2. Realización del esquemático del circuito

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, desde el menú principal: **Tools**→**Check Schematic**, o bien mediante el icono 

En la ventana inferior se informa si hay algún error en el circuito.

Si todo está correcto la ventana será la mostrada en la figura 13

Fig. 13



20

3. Creación de un símbolo

Con el esquema realizado se puede hacer un nuevo componente o símbolo 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

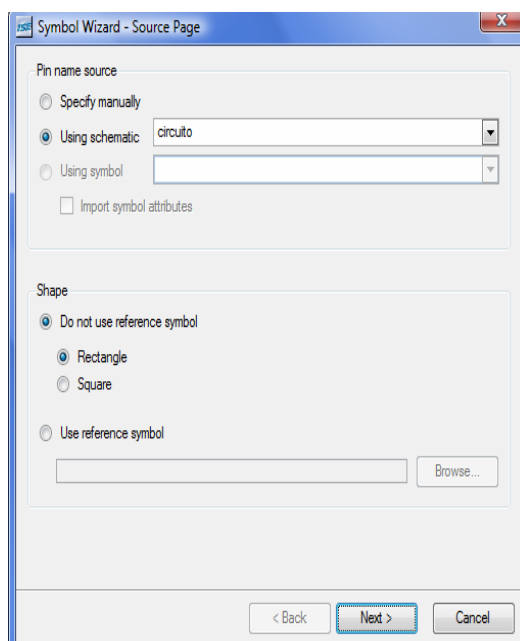
Este símbolo que se crea pasa a formar parte de la biblioteca de componentes del proyecto, y podrá utilizarse como un componente más

Para crearlo: **Tools**→**Symbol Wizard** desde el menú principal. Aparece entonces la ventana de la figura 14, donde hay que indicar que el símbolo se crea a partir del esquemático ya realizado (opción **Using Schematic** con el esquemático “circuito”)



21

3. Creación de un símbolo



Pulsando Siguiente tres veces y después Finalizar (aceptando las opciones por defecto a menos que se quiera variar el aspecto final del símbolo) el programa genera el símbolo y lo muestra (figura 15)

Fig. 14



22

3. Creación de un símbolo

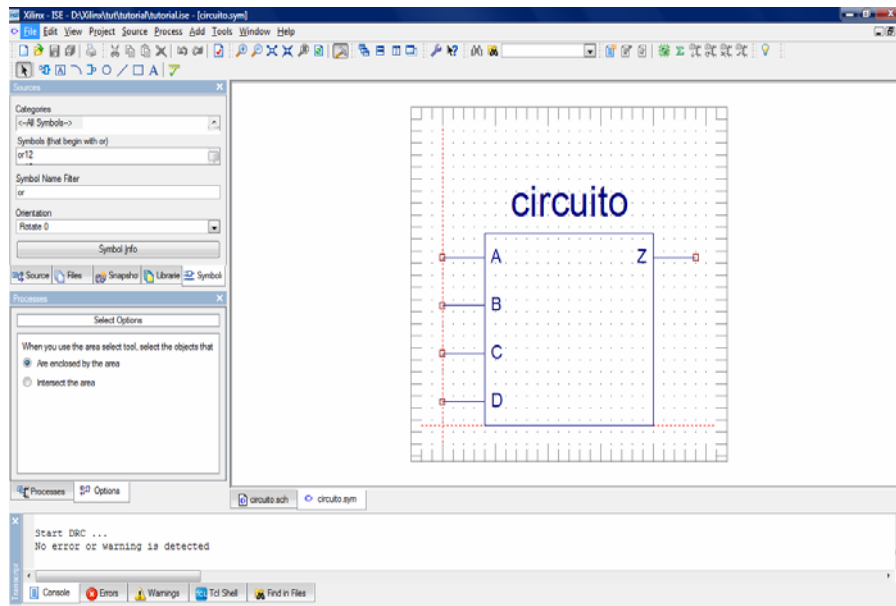


Fig. 15



23

3. Creación de un símbolo

Si nos fijamos en la parte inferior izquierda de la ventana del editor de esquemáticos, podemos ver que hay dos pestañas, [circuito.sch](#) y [circuito.sym](#), correspondientes a las dos hojas que tenemos abiertas en el editor de esquemáticos. Pulsando sobre [circuito.sch](#) 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, [<D:/Xilinx/tut/>](#), que se corresponde a la biblioteca 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



24

4. 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

4.1. 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 **Project**→**New Source**
- En la ventana de diálogo que aparece (figura 16), 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**
- Presionar **Siguiente**
- Presionar **Finalizar**



25

4. Simulación

Realizados los pasos siguientes, se inicia el banco de pruebas y el programa queda listo para introducir los requisitos necesarios

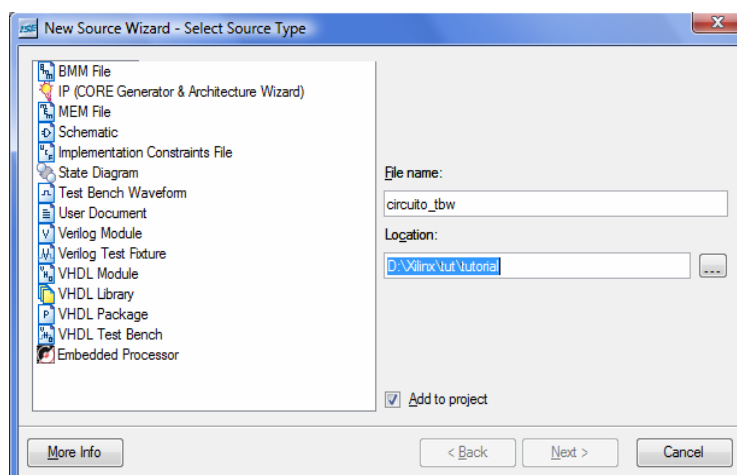


Fig. 16



26

4. Simulación

En la ventana **Initialize Timing** hay que especificar los parámetros de tiempos que se usarán en la simulación (figura 17), especificando que no hay reloj, sino que es un circuito combinacional (opción **Combinatorial (or internal clock)**)

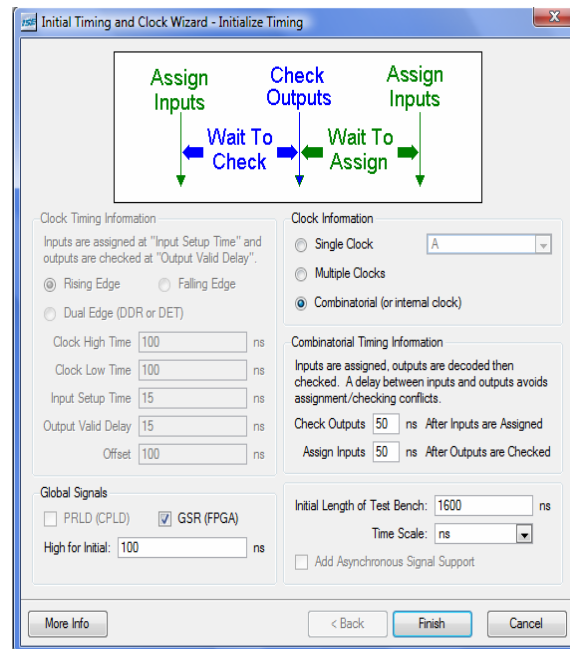


Fig. 17



27

4. Simulación

La opción **Check Outputs** define cuándo deberían estar ya en su valor correcto las salidas tras un cambio en las entradas. La opción **Assign Inputs** define el tiempo que pasa desde que se comprueban las salidas hasta que se asigna un nuevo valor a las entradas

Para este tutorial no cambiaremos ninguna de las especificaciones por defecto salvo la de **Initial Length of Test Bench**, que pondremos a 1600 ns (cada paso lleva 100 ns y así podremos probar las 16 opciones de la tabla de verdad inicial). Por tanto, las especificaciones quedarán finalmente:

- **Check Outputs:** 50 ns
- **Assign Inputs:** 50 ns
- **Initial Length of Test Bench:** 1600 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 18)



28

4. Simulación

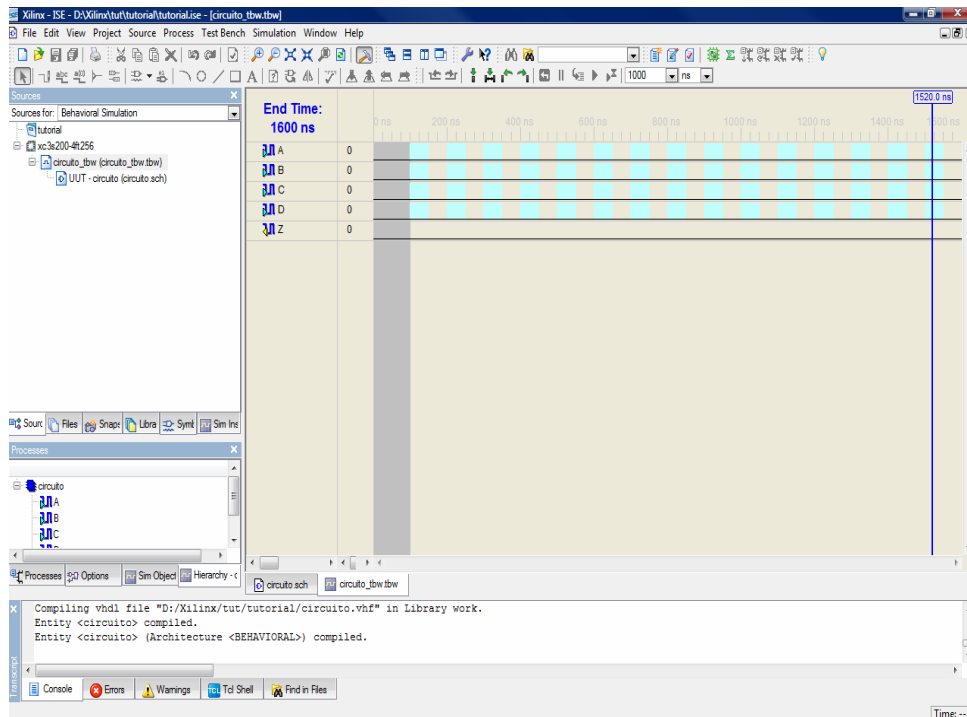


Fig. 18



29

4. Simulación

4.2. Inicialización de las entradas del circuito

En la nueva ventana del banco de pruebas (figura 18) aparece un cronograma donde habrá que introducir los estímulos (ceros o unos) correspondientes a las entradas para poder reproducir la tabla de la verdad del circuito diseñado

Aparecen, en la parte izquierda, las señales de las cuatro entradas y la salida. Inicialmente, todas las entradas se encuentran a cero

Es necesario inicializar las entradas con los valores adecuados para poder comprobar el correcto funcionamiento del circuito. Los estímulos (ceros o unos) se introducirán pinchando en la zona azul de cada celda

Para reproducir la tabla de verdad en el mismo orden, la entrada A permanece 8 ciclos a "0" y otros 8 a "1". 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

Los cambios se irán reflejando según se vayan introduciendo en el cronograma



30

4. Simulación

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 1600 ns

Pueden utilizarse los botones **Zoom In**, **Zoom Out** para ver más claramente los valores introducidos

El resultado puede verse en la figura 19



31

4. Simulación

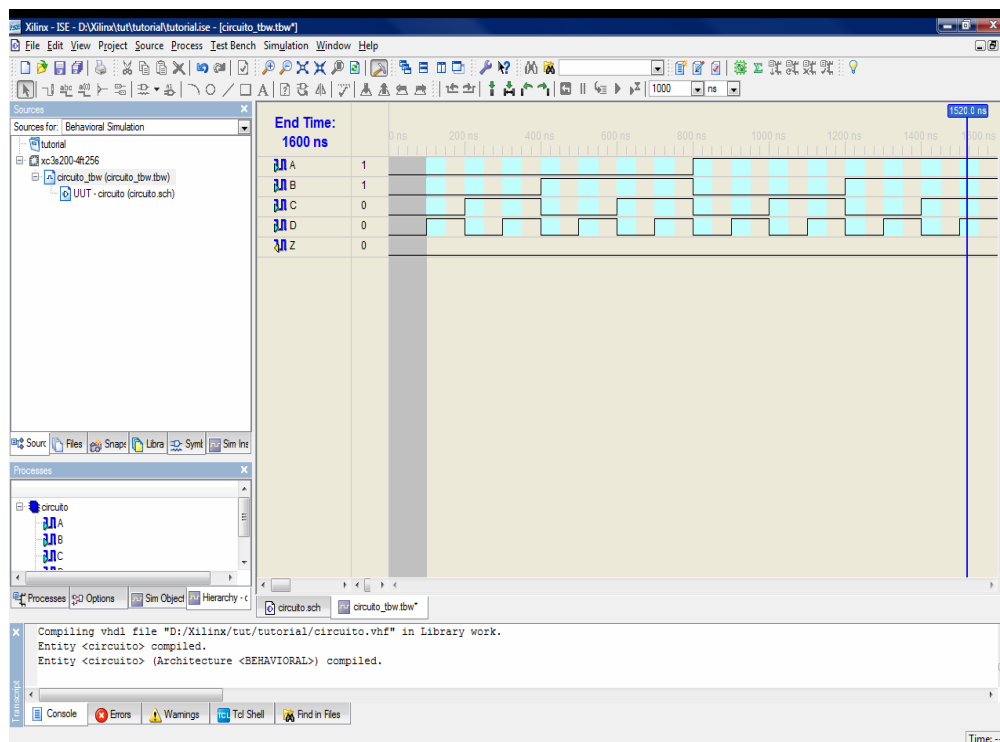


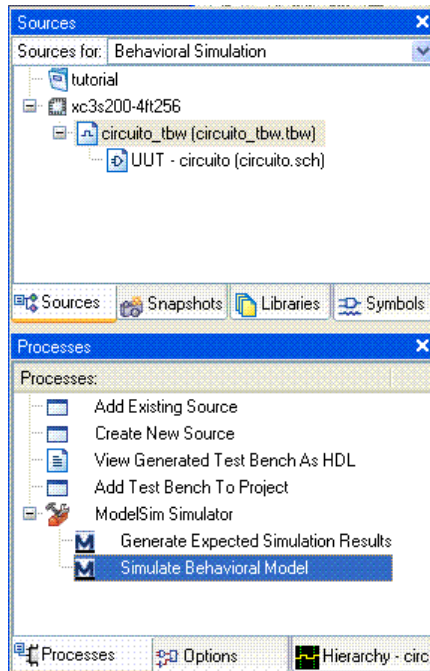
Fig. 19



32

4. Simulación

4.3. Simulación del comportamiento del circuito



Para realizar una **simulación del comportamiento** del circuito (behavioral), se utiliza la herramienta **ModelSim**

Pasos a seguir:

- En **Sources**, seleccionar primero **Sources for: Behavioral Simulation** desplegar el menú de jerarquía con **+** y seleccionar el fichero **circuito_tbw.tbw**
- En la ventana **Processes**, desplegar el menú de jerarquía con **+** y hacer doble click en **Simulate Behavioral Model** (figura 20). En ese momento se carga el programa **ModelSim**



Fig. 20

33

4. Simulación

Al arrancar el programa ModelSim aparece la ventana ModelSim SE, se ejecuta la simulación y aparecen una serie de ventanas embebidas en la principal (ver figura 21)

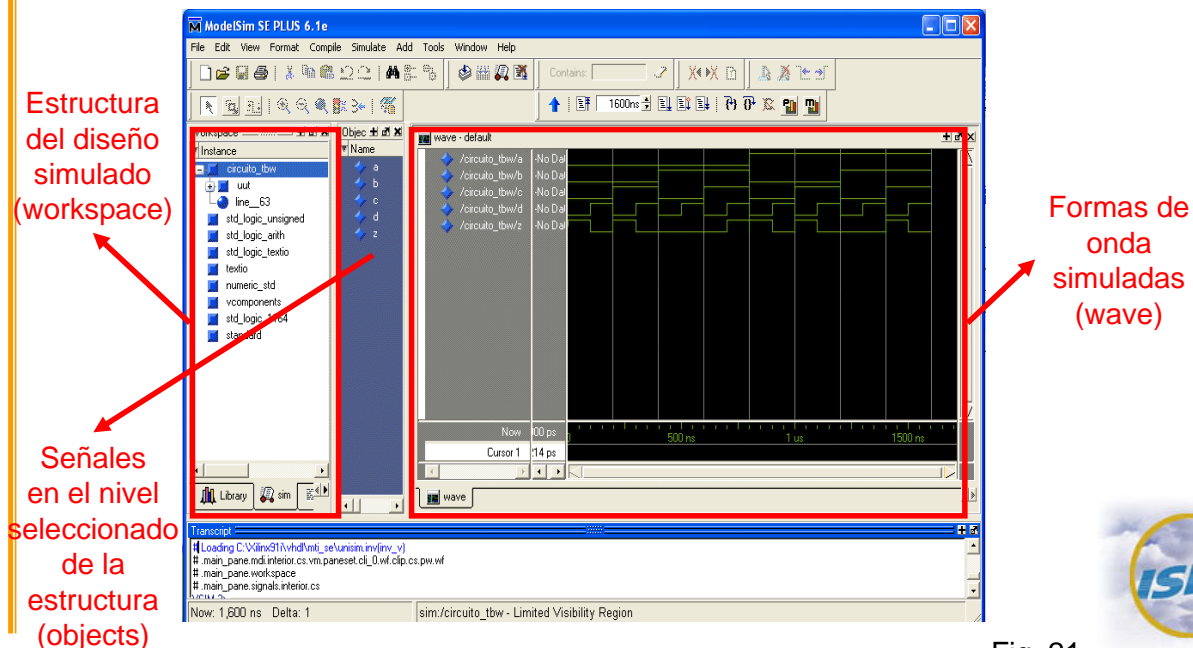


Fig. 21

34

4. Simulación

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 bibliotecas 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](#))
- Corre un tiempo igual a la duración del Test Bench Waveform generado previamente



35

4. Simulación

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](#). El resultado de la simulación debería ser como el mostrado en la figura 22

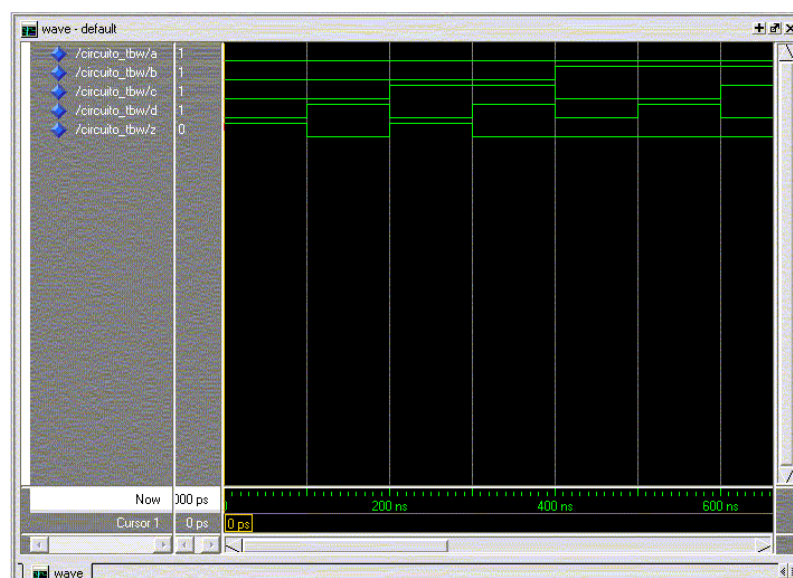


Fig. 22



36

4. Simulación

Para una comprobación más fácil del correcto funcionamiento del circuito y puesto que las entradas recorren todos los posibles estados (0, 1, 2... 15) 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 **wave-default** pulsando el botón izquierdo del ratón y las teclas shift o ctrl, con lo que aparecerán en el interior de un recuadro blanco:

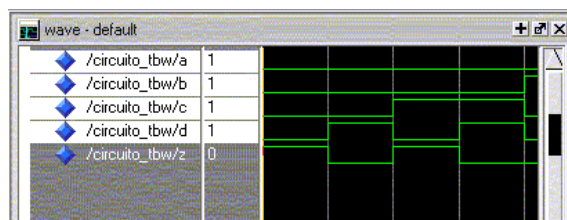


Fig. 23



37

4. Simulación

Con las señales que se vayan a combinar ya seleccionadas, se utiliza la opción de menú **Wave → Combine Signals...**:

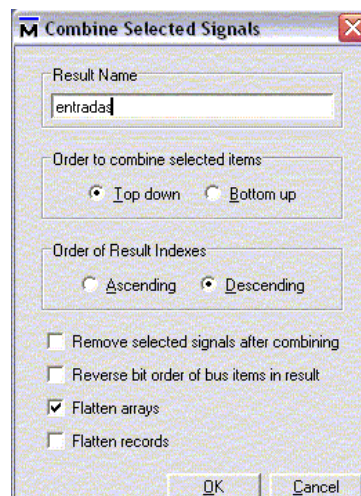


Fig. 24

En el cuadro de diálogo que aparece hay que dar un nombre a la nueva señal combinada, por ejemplo “**entradas**”



38

4. Simulación

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 como número en vez de en binario, habrá que seleccionar la entrada combinada, pulsar el botón derecho del ratón y en el menú de opciones elegir `Radix→Unsigned`. 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 25



39

4. Simulación

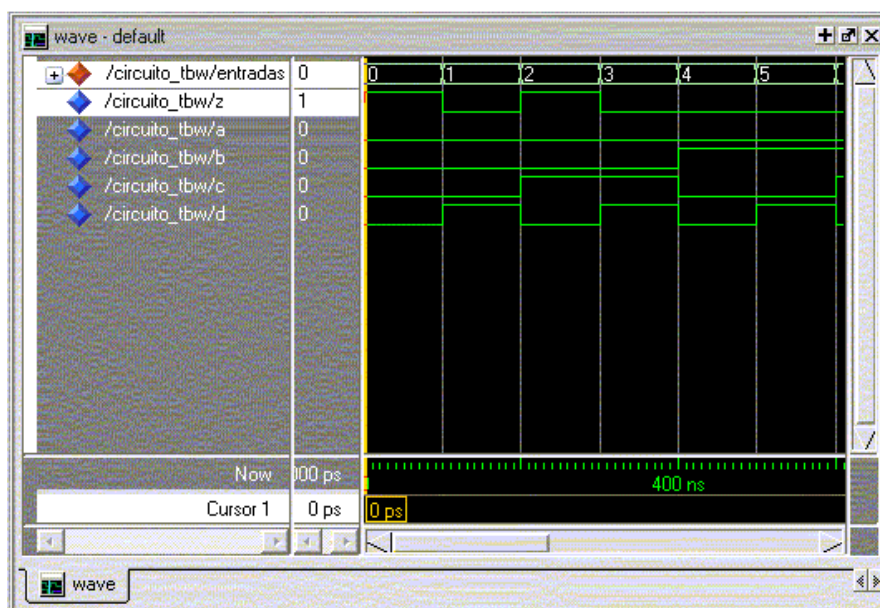


Fig. 25

Se sale de ModelSim cerrando la ventana `ModelSim SE`



40