

FRANCISCO J. CORBERA PEÑA
MARIO A. GONZÁLEZ PEÑALVER
ELADIO D. GUTIÉRREZ CARRASCO
JULIÁN RAMOS CÓZAR
SERGIO ROMERO MONTIEL
MARÍA ANTONIA TRENAS CASTRO

PRÁCTICAS DE
TECNOLOGÍA DE COMPUTADORES

UNIVERSIDAD DE MÁLAGA
©2005 Los autores



Parte I

Entorno de desarrollo

E1 | Descripción general



E1.1. Los proyectos

Con la finalidad de organizar nuestro trabajo, el entorno de desarrollo *XILINX Foundation Series* agrupa a todos los archivos implicados en nuestro diseño en lo que se denomina un **proyecto**. Entre otros, estos proyectos estarán compuestos por:

- **Ficheros fuente de nuestro diseño:** entre los que debemos destacar los esquemáticos (ficheros con extensión **.sch**, que serán los utilizados en este curso), los ficheros fuente descritos mediante HDL y los ficheros con diagramas de estado.
- **Ficheros intermedios y ficheros de salida:** generados durante el proceso de diseño, como pueden ser los ficheros de estímulos o ficheros de informes (*reports* y *log files*) .
- **Ficheros de configuración.**

Cada uno de los proyectos que vayamos implementando se almacenará en un directorio independiente, al que denominaremos **directorio de trabajo** de nuestro proyecto. Este directorio se crea con el mismo nombre que se haya dado al proyecto, debiéndose tener la precaución de que no exceda de 8 caracteres.

Para cada uno de los proyectos que vayamos creando, el **gestor de proyectos** (en inglés *Project Manager*) generará e irá actualizando un fichero de

descripción de nuestro proyecto. Aunque su extensión sea *.PDF (de *Project Description File*), no debe ser confundido con los ficheros .pdf de Acrobat. Este fichero .pdf nos describirá en todo momento el contenido y el estado en el que se encuentra el proyecto, almacenándose en nuestro directorio de trabajo con el mismo nombre que asignamos al proyecto.

Cuando trabajamos con un proyecto se genera un fichero denominado **Current.pdf**. Este fichero describe el proyecto con el que en ese momento estamos trabajando en el *Project Manager*. Al abrir un proyecto, el *Project Manager* copia el fichero PDF asociado a dicho proyecto en el **Current.pdf**. Toda la información acerca del estado y la configuración del proyecto que esté abierto es salvada al fichero **Current.pdf** en lugar del fichero pdf original del proyecto, actualizándose este último solamente en determinados casos:

- cuando abrimos otro proyecto,
- cuando creamos una copia del proyecto que esté abierto,
- cuando salimos del *Project Manager*.

El gestor de proyectos, cada vez que nos encontremos en alguna de las situaciones anteriores, actualiza el fichero original simplemente sobrescribiendo con el **Current.pdf**.

E1.2. El gestor de proyectos o *Project Manager*

En la figura E1.1 se muestra el aspecto general que presenta el gestor de proyectos de *XILINX Foundation*.

Puede observarse que la ventana del gestor de proyectos se divide en tres zonas principales:

- **Explorador de la jerarquía** (*Hierarchy Browser*): en esta zona podemos observar que existen dos pestañas, *Files* y *Versions*. Si pinchamos sobre la pestaña *Files*, veremos una estructura jerárquica en árbol con todos los documentos implicados en nuestro proyecto (diseños, librerías del proyecto, documentos, ficheros externos añadidos a nuestro proyecto, etc...). Si por el contrario, pinchamos sobre la pestaña *Versions*, aparecerán las distintas implementaciones que hayamos realizado de nuestro diseño. La pestaña que más nos interesará este curso, es la marcada con *Files*, pues en ella navegando o pulsando sobre las diferentes ramas del árbol de la jerarquía, podemos realizar múltiples operaciones como añadir, borrar y ordenar los documentos y librerías de nuestro proyecto, u obtener información de cualquiera de esos elementos.

Figura E1.1: Gestor de proyectos (*Project Manager*).

- **Diagrama de flujo del proyecto** (*Project Flowchart*): esta zona del gestor de proyectos representa de una manera gráfica el conjunto de pasos a dar para realizar una implementación completa de un diseño. Esto incluye el llegar a la implementación física del diseño sobre una FPGA. En este curso nos limitaremos a utilizar la herramienta de introducción de esquemáticos (*Design Entry*) y la posterior simulación de los mismos (*Simulation*) para comprobar que nuestro diseño funciona correctamente.
- **Ventana de mensajes** (*Message Window*): todos los comandos ejecutados en el gestor de proyectos, así como los mensajes originados por los mismos, se almacenan en un fichero log asociado al proyecto.

El contenido de este fichero log se nos muestra en la ventana de mensajes, ubicada en la parte inferior del gestor de proyectos. En esta ventana

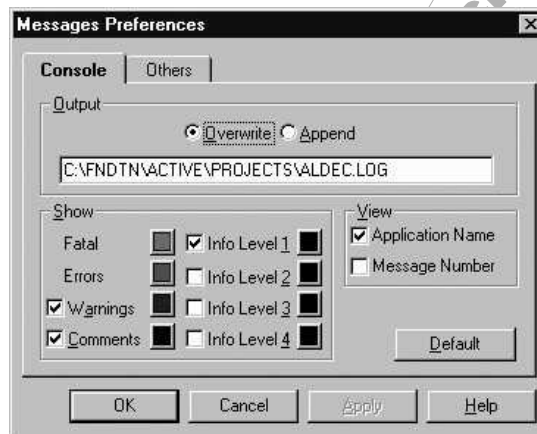








Figura E1.2: Preferencias de la ventana de mensajes.

también aparecerán los posibles errores y advertencias (*warnings*) presentes en nuestro diseño.

Accediendo a **File**→ **Preferences**→ **Messages** podemos personalizar la ventana de mensajes. Para ello, se abrirá una ventana como la mostrada en la figura E1.2, con la que podremos seleccionar entre las diferentes opciones.

Para la gestión y la ejecución de determinados comandos sobre los proyectos, el gestor de proyectos (figura E1.1) dispone de una barra de herramientas (*Toolbar*). Los botones más destacables de esta barra de herramientas realizan las siguientes acciones:  crea un nuevo proyecto,  abre un proyecto existente,  permite añadir librerías a nuestro proyecto,  nos da información del objeto seleccionado en ese instante dentro del árbol jerárquico (*Hierarchy Browser*) de nuestro proyecto,  para encontrar algún símbolo o algún cable dentro de nuestro diseño y el botón  ofrece una ayuda sobre el elemento que se seleccione.

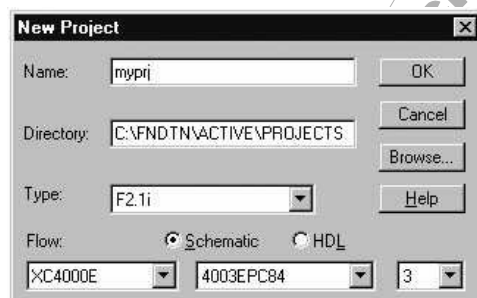




Figura E1.3: Ventana de nuevo proyecto.

Por último, en la parte más baja del gestor de proyectos tenemos la barra de estado (*Status Bar*), que nos da información referente a las operaciones que se están desarrollando en ese instante en el gestor de proyectos.

E1.3. Creando un nuevo proyecto

Para crear un nuevo proyecto debemos seleccionar **File**→ **New Project** o pulsar sobre el botón  de la barra de tareas. Como resultado se nos abrirá una ventana como la mostrada en la figura E1.3, en la que deberemos asignar un nombre a nuestro proyecto, teniendo la precaución de no elegir un nombre coincidente con el de alguna de las librerías de Xilinx o alguno de los componentes de dichas librerías.

En los siguientes apartados, especificaremos el directorio donde se almacenará y el tipo de proyecto que implementaremos, esquemático o VHDL. En este curso todos los proyectos serán implementados mediante esquemáticos, por lo que seleccionaremos *Schematic*. Finalmente, podemos elegir el tipo de dispositivo que utilizaremos (familia, tipo y velocidad), lo que en este curso se hará tal y como aparece en la figura E1.3. Una vez hayamos terminado de completar todos los apartados de este cuadro de diálogo, pulsaremos sobre el botón OK, con lo que el nombre del proyecto y del dispositivo utilizado aparecerán en el *Hierarchy Browser* del gestor de proyectos.

A partir de este momento podremos utilizar las herramientas de diseño, ubicadas en el *Design Entry* del *Project Manager*, para añadir ficheros a nuestro proyecto. Durante este curso únicamente añadiremos esquemáticos, que generaremos con el editor de esquemáticos (botón ). El modo de hacerlo se describirá en el siguiente capítulo.

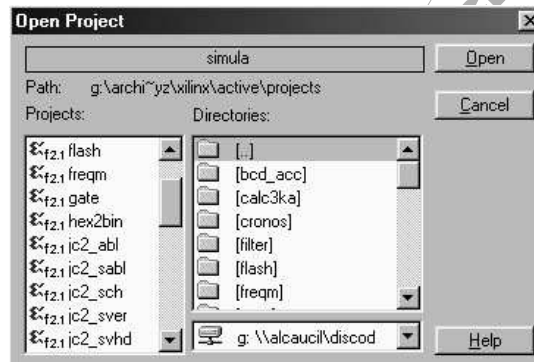



Figura E1.4: Abrir un proyecto.

E1.4. Abriendo un proyecto existente

Para abrir un proyecto existente, es decir, que haya sido creado con anterioridad, simplemente seleccionaremos **File**→ **Open Project** o pulsaremos sobre el botón . Como resultado se nos abrirá un cuadro de diálogo similar al que muestra la figura E1.4, en el que podremos seleccionar el proyecto que deseamos abrir. Si por algún motivo, el proyecto que deseamos abrir está fuera del directorio de proyectos o de trabajo, podremos cambiarnos de directorio seleccionándolo en la ventana *Directories*. Una vez que hayamos seleccionado el proyecto, y que éste haya sido abierto, su nombre y el dispositivo que utiliza aparecerán en la barra superior de título del *Project Manager*.

A la hora de abrir un proyecto hay que tener en cuenta una serie de aspectos:

- El *Project Manager* únicamente permite trabajar con un solo proyecto, por lo que al abrir uno automáticamente se cerrará el proyecto que estuviera abierto.
- Al iniciar el *Project Manager*, se abre por defecto el proyecto con el que se estuvo trabajando en la última sesión. Para ello utiliza el fichero **Current.pdf** que se mencionó con anterioridad.
- La primera vez que ejecutemos el *Project Manager*, el fichero **Current.pdf** no existirá, por lo que en ese momento se nos preguntará mediante un cuadro de diálogo si deseamos crear uno nuevo o abrir uno existente.

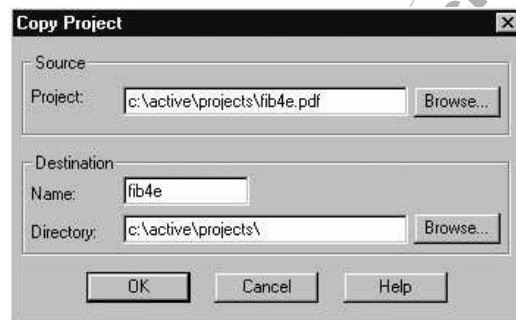


Figura E1.5: Copia de proyectos.

E1.5. Copiando un proyecto

En cualquier momento podemos generar una copia de nuestro proyecto, lo que se traduce en la realización automática de las siguientes operaciones:

- Copia del fichero PDF asociado al proyecto.
- Copia de todos los ficheros almacenados en el directorio de trabajo del proyecto.
- Copia de las librerías utilizadas en el proyecto.

Para crear una copia de un proyecto, debemos seleccionar **File**→ **Copy Project**. Como resultado se nos mostrará el cuadro de diálogo de la figura E1.5. En el campo señalado con *Source* debemos seleccionar el fichero PDF del proyecto que queremos copiar. Por defecto, en este campo se muestra el fichero PDF del proyecto que actualmente esté abierto. Para seleccionar otro proyecto, podemos usar el botón *Browse* para buscarlo o directamente escribir el nombre del fichero. En el campo *Destination* asignaremos un nombre a la copia del proyecto y en el campo *directory* seleccionaremos el directorio en el que se almacenará. Por defecto, el *Project Manager* sugiere un directorio, pero se puede elegir libremente otro directorio mediante el botón *Browse* o insertándolo manualmente. Finalmente pulsaremos OK para finalizar la copia. Recordemos que si estamos haciendo una copia del proyecto abierto en ese momento, antes de comenzar la operación de copia se actualizará su fichero PDF.

El método para realizar una copia que hemos descrito en los párrafos anteriores es el único recomendable si no queremos sufrir una pérdida de datos.

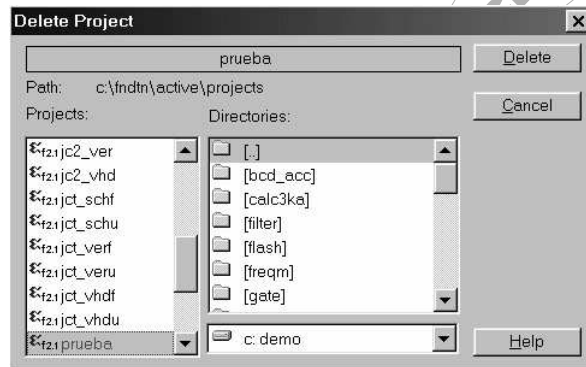


Figura E1.6: Borrado de proyectos.

Será el que se utilice en el laboratorio cuando se indique que un proyecto debe ser copiado.

E1.6. Borrando un proyecto

Para borrar de forma permanente un proyecto, seleccionaremos **File**→**Delete Project**, con lo que aparecerá un cuadro de dialogo como el de la figura E1.6. El cuadro de diálogo muestra el directorio y la ruta completa (*path*) para el proyecto actualmente abierto. En la parte izquierda del cuadro de diálogo aparecen todos los proyectos ubicados en el directorio indicado: para borrar cualquiera de ellos simplemente lo seleccionaremos y pulsaremos *Delete* para completar el borrado. Si el proyecto se encuentra en otro directorio, sólo habrá que seleccionarlo del árbol de directorios mostrado en la ventana *Directories*.

E1.7. Almacenando y recuperando un proyecto

El gestor de proyectos nos permite almacenar nuestro proyecto en un fichero comprimido con la extensión ZIP, seleccionando **File**→ **Archive Project**. Un asistente (*wizard*) nos irá guiando a lo largo del proceso mediante una serie de ventanas.

El aspecto de la primera de estas ventanas se muestra en la parte superior izquierda de la figura E1.7. Tendremos que proporcionar un nombre para el fichero ZIP y cuál será su ubicación (*path*); añadir notas o comentarios,

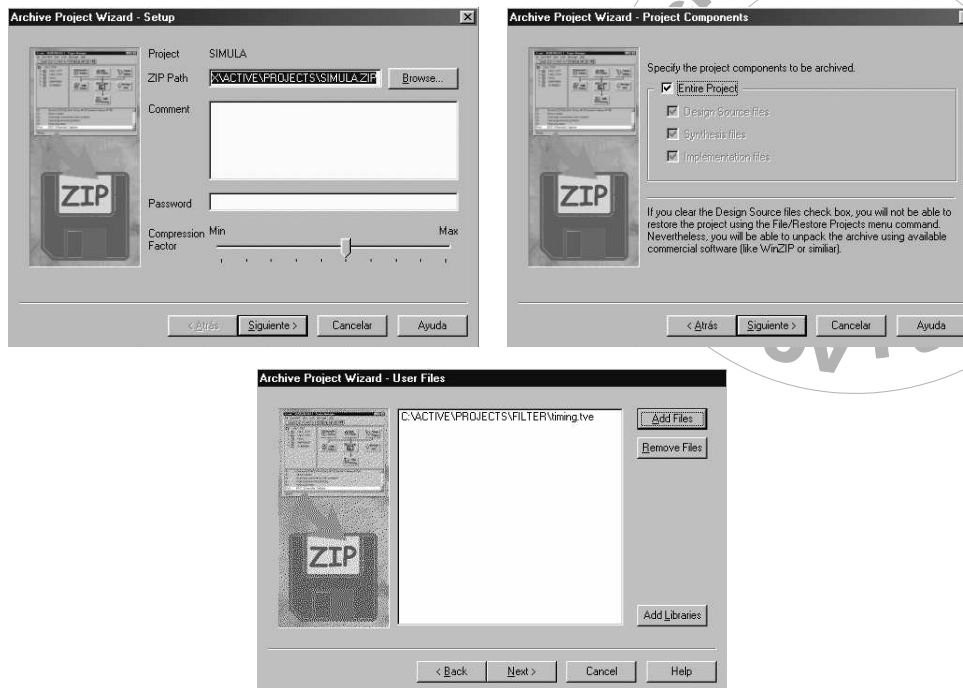


Figura E1.7: Almacenamiento de proyectos.

si lo deseamos; también opcionalmente, introducir una *password* o clave para proteger nuestro fichero, y finalmente, seleccionar el nivel de compresión requerido.

Una vez completando los distintos campos, pulsando **Next** aparecerá la siguiente ventana, mostrada en la parte superior derecha de la figura E1.7. Nos preguntará qué tipo de información deseamos incluir en el fichero ZIP, es decir, si deseamos almacenar el proyecto completo o solamente determinadas partes del mismo. Una vez marcadas las casillas adecuadas pulsaremos **Next** y llegaremos a la última ventana, mostrada en la parte inferior de la figura E1.7.

Esta ventana nos permite agregar los ficheros que deseemos almacenar en nuestro ZIP, simplemente pulsando sobre el botón **Add Files**, que nos permitirá seleccionar los ficheros a añadir. También podemos almacenar librerías adicionales mediante el botón **Add Libraries**. Por último, esta ventana nos permite borrar ficheros añadidos con anterioridad a nuestro fichero ZIP. Para finalizar, pulsaremos sobre el botón **Start**, con lo que el proceso de almacena-

miento comenzará, mostrándose con una barra su progreso. Una vez completado el proceso, pulsaremos sobre el botón **Close** para cerrar la ventana.

Un problema que puede surgir cuando se almacene un proyecto, es el del tamaño del fichero ZIP generado: es muy probable que exceda los 1.4 Mb de límite que tiene un disquete (formato muy utilizado para almacenar el trabajo realizado en el laboratorio). La solución de este problema es muy sencilla: si indicamos que queremos que se guarde el fichero ZIP en la unidad **a:** (junto con el nombre del fichero), el programa se encargará de irnos pidiendo los sucesivos discos a medida que vayan haciendo falta.

El archivo ZIP incluye la siguiente información:

- el directorio de trabajo con todo su contenido y un fichero **Readme.txt** que contiene los comentarios que hayamos introducido durante la generación del ZIP,
- el fichero PDF de descripción de nuestro proyecto,
- los ficheros de configuración INI y el log del proyecto,
- ficheros de configuración del sistema (**Autoexec.bat** y **Config.sys**) y ficheros añadidos por el usuario al proyecto.

El proyecto almacenado en el fichero ZIP puede ser recuperado en su totalidad, para continuar trabajando con él, mediante el comando **File→ Restore Project**. A continuación elegiremos cuál es el fichero ZIP del proyecto a recuperar, introduciendo la *password* si se le añadió alguna, y pulsaremos sobre el botón **Next**. En ese momento comenzará el proceso de recuperación de nuestro proyecto. Si el *Project Manager*, encuentra que algunos de los ficheros ya existen, nos preguntará que queremos hacer con ellos: si queremos sobrescribir un fichero, todos los ficheros, no sobrescribir, o simplemente abortar la operación.

E1.8. Gestionando los documentos de un proyecto

Los documentos de un proyecto son todos los ficheros que contienen una descripción de nuestro diseño. En nuestro caso utilizaremos sólo esquemáticos (ficheros con la extensión *.sch), pero también podría describirse mediante ficheros HDL o con diagramas de estados. Los documentos se clasifican como:

- **Documentos Top Level:** aquellos que se encuentran en lo más alto de la jerarquía de un proyecto.

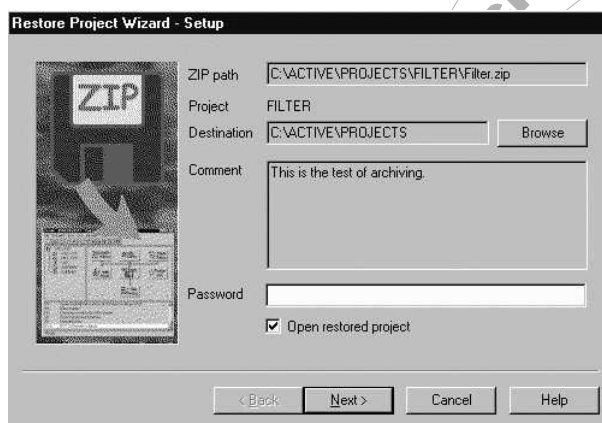


Figura E1.8: Recuperando proyectos.

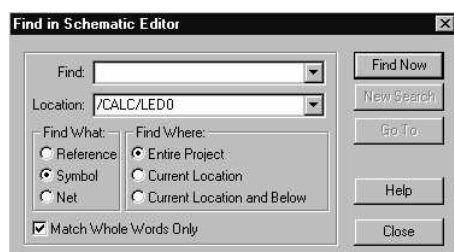


Figura E1.9: Búsqueda en proyectos.

- **Macros:** los que ocupan un nivel más bajo dentro de la jerarquía.

Para que cualquier documento forme parte de un proyecto debemos añadirselo, ejecutando para ello el comando **Document** → **Add**.

Resulta útil la posibilidad de buscar un componente o cable de dentro del árbol jerárquico del proyecto, para lo cual podemos ejecutar el comando **Document** → **Find Object**. En un cuadro de diálogo como el mostrado en la figura E1.9, podremos escribir el nombre del componente que queremos localizar, qué clase de componente es (*Reference*, *Symbol* o *Net*) y el lugar donde deseamos buscarlo (en todo el proyecto o sólo en parte).

Una vez que se haya realizado la búsqueda, y si los resultados de dicha búsqueda con los parámetros introducidos fueron positivos, pulsando con un doble clic sobre ellos podremos acceder al elemento en el editor de esquemáticos.

E1.9. Configurando el *Project Manager*

Algunos aspectos del funcionamiento del *Project Manager* pueden ser configurados mediante la opción **File** → **Preferences** → **Configuration**. Tras la ejecución de dicho comando, nos aparecerá un cuadro de diálogo como el mostrado en la figura E1.10, en el que podremos elegir entre las diferentes opciones que se presentan. Una vez elegidas las opciones, y confirmados los cambios, esta información se almacenará en el fichero de configuración `ALDEC.ini`. En algunos casos, un mensaje nos indicará que es necesario resetear *Project Manager* para que los cambios tengan efecto.

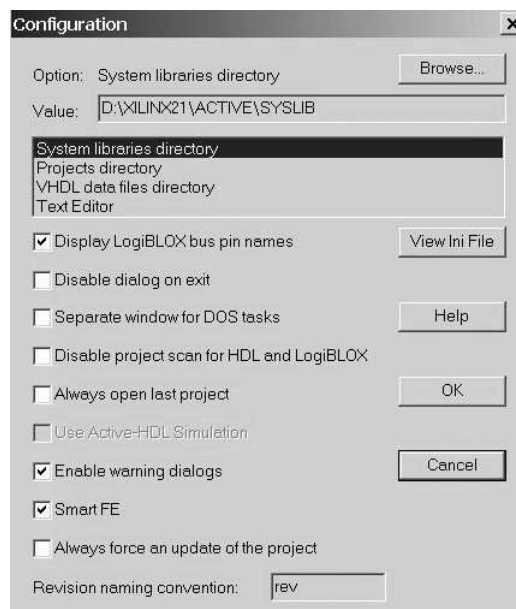


Figura E1.10: Opciones del *Project Manager*.

E1.10. Gestionando las librerías del proyecto


Una librería es, básicamente, el lugar en el que se almacenan todos los componentes utilizados en un proyecto. Con la herramienta *XILINX Foundation* se utilizan dos tipos de librerías:

- **librerías del sistema**, que son las librerías que trae consigo la herramienta y que están disponibles para todos los proyectos, y

- **librerías del usuario**, definidas por él dentro de su proyecto, conteniendo componentes de diseño propio (denominados **macros**).

Denominaremos librerías de un proyecto a la suma de los dos tipos de librerías descritos anteriormente. Las librerías de sistema se añaden a los proyectos por defecto y no pueden ser borradas de los mismos. En cambio, las definidas por el usuario, siempre podrán ser eliminadas de un proyecto, pero siempre con precaución, puesto que su eliminación puede dar lugar a errores a la hora de implementarlo.

En cualquier momento podemos añadir manualmente nuevas librerías procedentes de otros proyectos al nuestro. La gestión de las librerías y de las operaciones que se pueden hacer con ellas, se realiza de una manera óptima con el **gestor de librerías**, que forma parte de la herramienta *XILINX Foundation*. Este tema no será tratado en profundidad en este manual. Sólo destacaremos algunas operaciones interesantes, como la forma de adjuntar una librería a un proyecto o eliminarla.

Si abrimos el gestor de librerías, ya sea seleccionando **File→ Project libraries** o pulsando sobre el botón , se nos abrirá un cuadro de dialogo como el representado en la figura E1.11. Este cuadro nos permite añadir (*add*) librerías a nuestro proyecto de entre las librerías disponibles, borrarlas (*Remove*), obtener información de las librerías (*Info*) o acceder al gestor de librerías (*Lib Manager*).

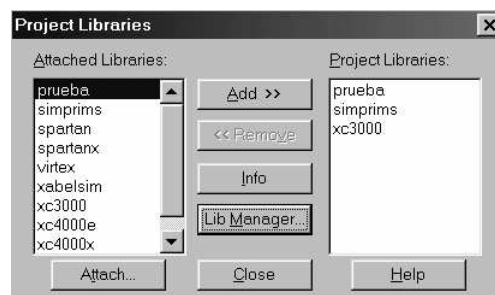


Figura E1.11: Librerías de proyectos.





La forma de introducir esquemáticos (archivos .SCH) en un proyecto Xilinx es mediante la herramienta “Editor de esquemáticos” (**Schematic Editor**). Una vez abierto el proyecto sobre el que queremos trabajar, para acceder a dicha herramienta bastará con acceder al menú **Tools→Design Entry→Schematic Editor** o bien pulsando sobre el botón que representa el editor de esquemáticos en la ventana del **Project Flowchart**, como se puede ver en la figura E2.1.

Tras abrir el editor de esquemáticos nos encontramos con una clásica ventana windows, con una barra de menú en la parte superior con las opciones habituales, una “barra de herramientas” en la parte superior, una “barra de herramientas esquemáticas” en la parte izquierda y en la parte inferior una “barra de estado”. La mayor parte de la pantalla está ocupada por el área de trabajo para crear esquemáticos, y es ahí donde se hará el diseño de nuestros circuitos. En la figura E2.2 se muestra el aspecto de la ventana de edición de esquemático en la que se han destacado las funcionalidades más significativas de las barras de botones.

A continuación se presentan los aspectos fundamentales del diseño de circuitos digitales basados en esquemáticos. Como este documento no pretende ser un manual exhaustivo de utilización de editor de esquemáticos, sino una pequeña guía para empezar a crear nuestros circuitos, se recomienda consultar la extensa ayuda de la propia aplicación para obtener información sobre conceptos más avanzados o para ampliar información sobre los presentados aquí.

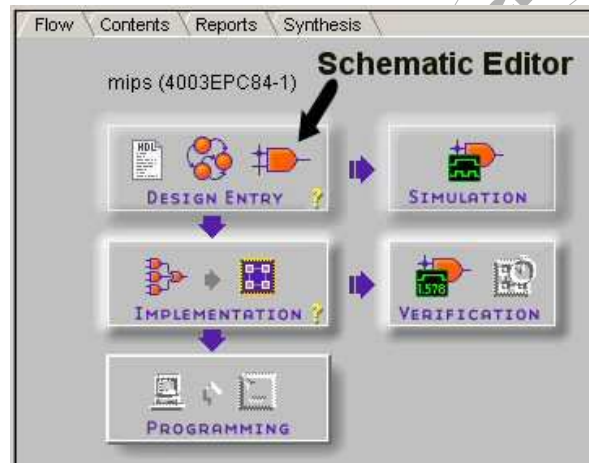


Figura E2.1: *Schematic Editor* en la ventana del *Project Flowchart*.


E2.1. Diseño básico

En el menú **File** nos encontramos con las opciones típicas de cualquier aplicación windows. Entre ellas podemos destacar:

- **New Sheet** que se utilizará para crear un esquemático nuevo en blanco.
- **Open** y **Close** abren y cierran respectivamente un esquemático que ya existe.
- **Save** y **Save all** los utilizaremos para guardar el esquemático actual o todos los que tengamos abiertos.

Desde este menú también podremos imprimir nuestros diseños y por supuesto salir del editor de esquemáticos.

E2.1.1. Colocando elementos en nuestro circuito

Para colocar elementos en nuestro diseño utilizaremos el acceso directo  de la barra de herramientas, la entrada de menú **Mode**→ **Symbols** o la tecla **F3**. Esto nos abrirá una pequeña ventana **SC Symbols** en la cual aparecen listados todos los elementos (puertas lógicas, biestables, etc..) que existen en las librerías del proyecto y que son los únicos que podemos utilizar. Dicha ventana se muestra en la figura E2.3. Bastará con seleccionar el que

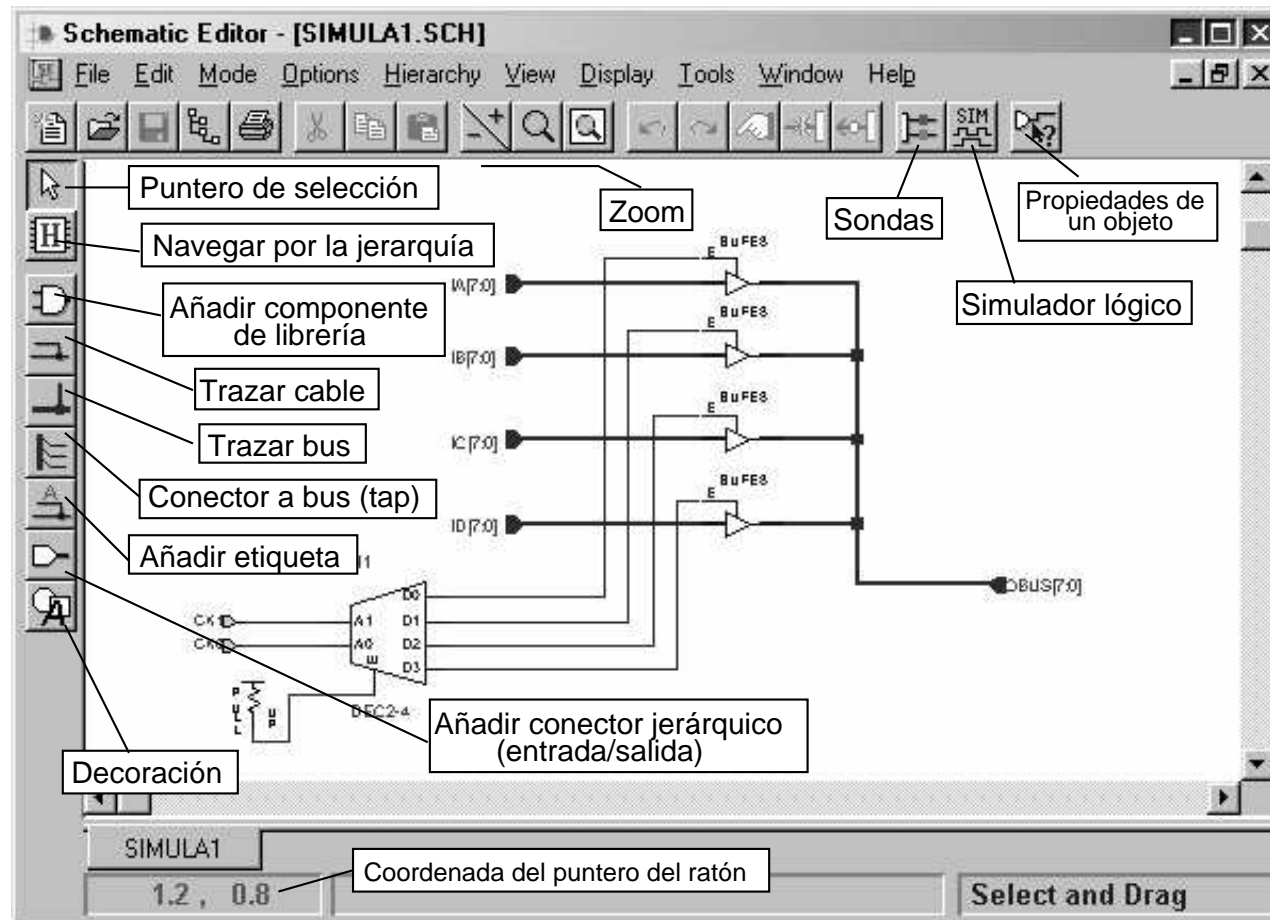


Figura E2.2: La ventana de edición de esquemáticos.

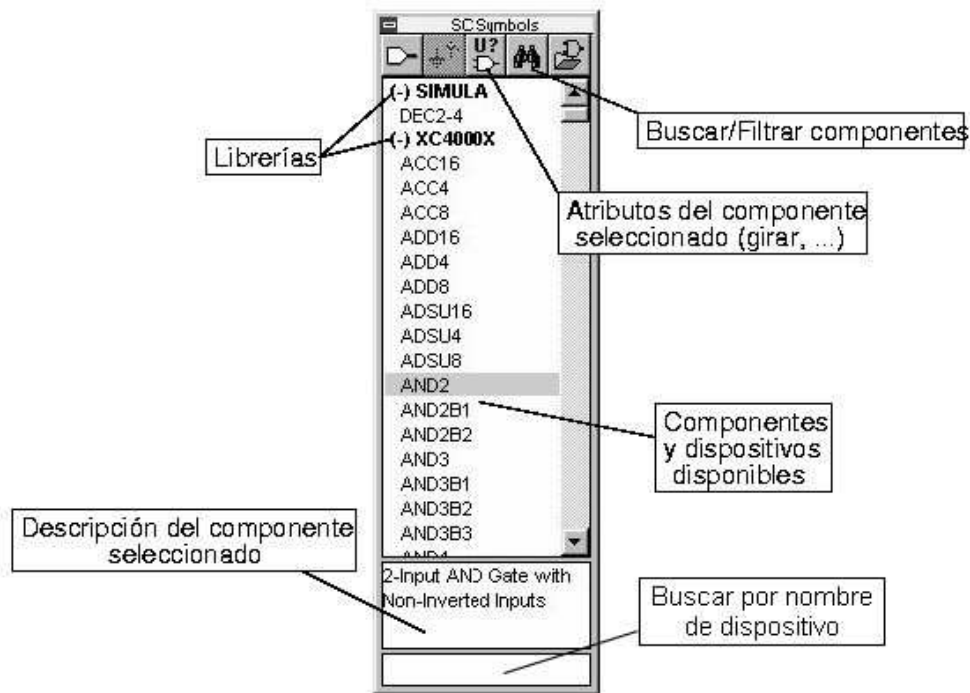



Figura E2.3: Selección de componentes de librería.

queramos y al pasar el cursor al área de trabajo aparecerá el símbolo del elemento seleccionado que podremos situar donde queramos. En la ventana de símbolos aparecen todos los disponibles en las librerías, por lo que buscar uno en concreto si no se conoce su nombre exacto, puede ser complicado.

Por este motivo, existe un **symbol filter**  en la ventana de símbolos que nos permite seleccionar el tipo de elementos a mostrar (puertas, biestables, registros, memorias, etc.) en la ventana de selección. El panel abierto con esta acción es el mostrado en la figura E2.4, y básicamente lo que haremos es activar aquellas casillas que corresponden con la categoría en la que estamos interesados. Observemos que en el panel se pueden desactivar (**Clear**) o activar (**Default**) todas las categorías. Podemos también buscar por nombre de dispositivo.

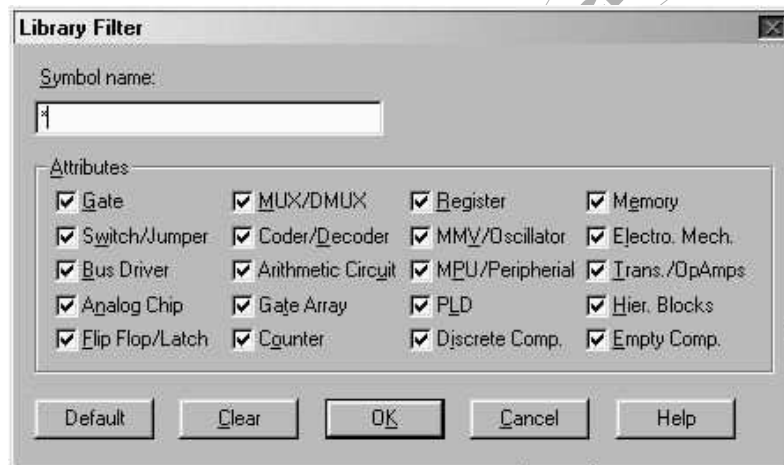


Figura E2.4: Filtrando los componentes que vemos en la ventana de selección de símbolos de librería.

E2.1.2. Conectando los elementos del circuito

Una vez colocados los elementos que componen nuestro circuito habrá que conectarlos entre sí mediante cables (señales simples) o buses (conjunto de señales).

		Señal simple	Bus
Dentro de la misma hoja	Cables		
	Etiquetas		
Conexión jerárquica			



Para dibujar un cable entre dos nodos basta con usar el botón , ó la entrada de menú **Mode**→ **Draw Wires** ó la tecla **F4**. Basta con seleccionarlo y pulsar con el botón izquierdo del ratón allí donde queremos empezar a dibujar el cable y hacer lo mismo sobre el punto donde lo queremos terminar. Automáticamente se hará un trazado para que el cable no cruce por encima los elementos del circuito. Si no nos agrada el trazado automático, podemos ir creándolo nosotros, simplemente pulsando con el botón izquierdo del ratón allí donde queramos que el cable tenga un vértice.




Figura E2.5: Poniendo etiqueta a una señal.

Si se quiere llevar una misma señal a varios puntos (por ejemplo la salida de una puerta lógica a las entradas de otras tantas), basta con dibujar el primer cable y los demás comenzarlos en cualquier punto del cable anterior y llevándolo hasta el nuevo destino, apareciendo automáticamente un símbolo de conexión entre los cables.

Otra manera de llevar a cabo esa conexión múltiple o simplemente hacer una conexión sin tener que dibujar por completo el cable, es utilizando nombres (etiquetas) para los cables o **nets**. Ello permite dibujar un esquemático más despejado, en caso de existir un buen número de conexiones. Si se pulsa dos veces con el botón izquierdo sobre un cable o bien con el botón derecho se selecciona **net properties**, se abre una ventana en la cual se le puede dar un nombre a dicho cable o net (**net name**). Todos los cables que nombremos con el mismo *net name* es como si estuvieran todos conectados entre si.

También es posible darle nombre a los cables sin tener que editar sus propiedades, sino con el botón . Aunque abre la misma ventana que con el método anterior (**net name**), una vez puesto el nombre del net, nos permite pulsar sobre un botón repetir (**Repeat**) que es muy útil a la hora de nombrar cables con identificadores de bits consecutivos de un bus. Por ejemplo, si pulsamos dicho botón y ponemos como nombre **A0**, al darle a **Repeat**, desaparecerá la ventana y aparecerá un recuadro en el cursor. Situaremos dicho recuadro sobre el cable a nombrar con **A0** y una vez hecho, seguirá el recuadro en el cursor para que podamos renombrar más cables, pero ahora el siguiente se nombrará con **A1** y así sucesivamente.

Es posible hacer que un cable se pueda comunicar con el exterior del circuito (entrada/salida). Para ello hay que conectarlo con un elemento denominado **Hierarchy connector** que se introducen utilizando el botón . Para definirlo bastará con darle un nombre y un tipo (entrada o salida). También existe el botón **Repeat** que tiene exactamente la misma función que hemos

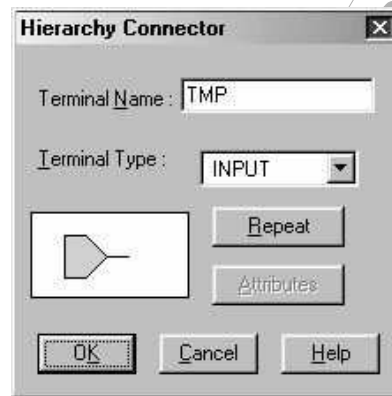


Figura E2.6: Conector jerárquico para una señal individual.

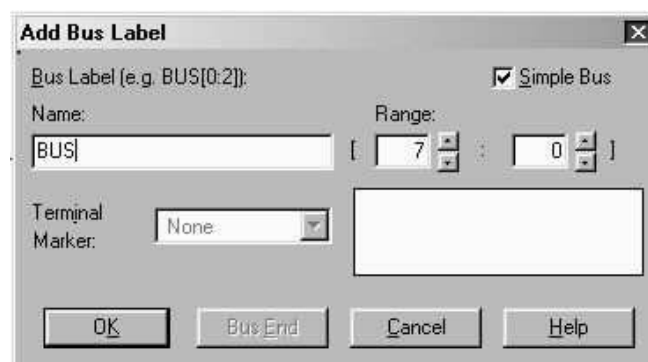



Figura E2.7: Poniendo etiqueta a un bus.

explicado para los **net name**, es decir, crear entradas/salidas con el mismo nombre e índices consecutivos.

El trazado de **buses** es básicamente igual pero utilizando el botón  o utilizando la entrada de menú **Mode**→ **Draw Buses (F5)**. Una diferencia importante con los cables es que al pulsar dos veces sobre el bus o con el botón derecho seleccionando **edit bus**, se accede a una ventana en la que no solo se le puede cambiar el nombre al bus (**Name**) como se hacía con los cables, sino que además se puede cambiar el rango (**Range**) del mismo (índice del bit más y menos significativo).

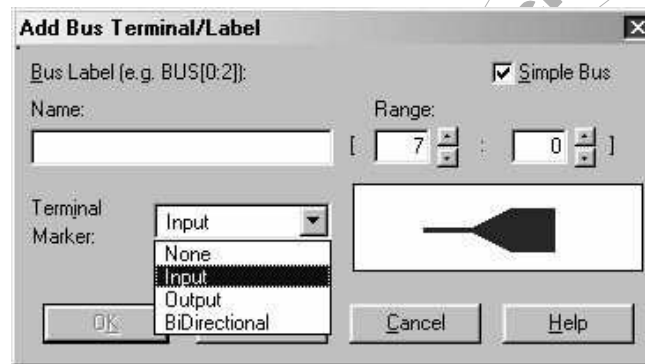



Figura E2.8: Conector jerárquico para un bus.


Si utilizamos un bus para conectarnos con el exterior, finalizaremos el trazado del mismo con un doble clic con el botón izquierdo allá donde queramos que termine. En ese momento se nos volverá a abrir la ventana de las propiedades del bus, donde podremos establecer el terminador del bus (**Terminal Marker**) que sirve para indicar si es un bus de entrada (**INPUT**), salida (**OUTPUT**) o simplemente un bus interno (**NONE**).

Hay un tipo de conexión no comentada hasta el momento, que consiste en conectar un cable a un bit concreto de un bus. Este tipo de conexiones se pueden hacer de varias formas.

- Una es utilizando los nombres, es decir, si un cable se quiere conectar con el bit 5 de un bus llamado **A**, bastaría con dibujar el cable y ponerle de nombre **A5**, es decir el nombre del bus seguido del número del bit al que se quiere conectar.
- Otra forma es utilizando los **Bus Taps** que se introducen con el botón  (entrada de menú **Mode** → **Draw Bus Taps** o **F6**). Una vez seleccionada esta opción, lo primero que hay que hacer es seleccionar el bus en cuestión. A continuación cambiará la forma del cursor indicando que ya podemos ir seleccionando los cables (nets) a los que lo queremos unir (dichos cables deberán tocar el bus para poder ser seleccionados por el Bus Tap). Hay que tener en cuenta que este método empezará conectando el bit más significativo del bus y continuará hacia el menos significativo sucesivamente.


NOTA: No es posible conectar dos *nets* con distinto nombre, por lo tanto no se puede conectar por ejemplo el bit 5 de un bus **A** con un cable llamado **B**.

Cuando se quiera dar dos nombres a una misma señal/cable, habrá que poner un elemento entre medias (por ejemplo un buffer (BUF)).

También es posible “adornar” el diseño del circuito añadiendo texto y gráficos que nos ayuden a mejorar la presencia visual del mismo. Para ello utilizaremos la **Graphics Toolbox**  que nos abrirá una pequeña ventana que nos permitirá la inserción de texto y el trazado de líneas, cuadros, círculos, etc.


E2.1.3. Modificando el circuito

Todos los elementos introducidos en nuestro diseño pueden ser modificados fácilmente.

Así se pueden **mover** simplemente arrastrándolos de la forma estándar en windows (**drag and drop**), manteniéndose en todo momento las conexiones que tengan. Además se pueden hacer las típicas acciones de **cortar**, **copiar** y **pegar** con los botones  o simplemente **eliminados** seleccionándolos primero y luego pulsando la tecla suprimir.

Se puede acceder a sus **propiedades** pulsando sobre ellos dos veces con el botón izquierdo o una vez con el derecho y seleccionando la opción correspondiente en el menú desplegable.

Podemos cambiar la salida por pantalla del circuito haciendo zoom sobre el mismo para una visualización más amplia o por el contrario con más detalle. Para ello utilizaremos los siguientes botones de la barra superior:

 . El primero disminuye(-)/aumenta(+) el tamaño de los objetos a intervalos regulares, el segundo nos permite marcar un área que ocupará todo el área de trabajo y el tercero nos muestra toda la hoja de trabajo.

E2.1.4. Comprobando el circuito

Durante la fase de creación de un circuito es probable que cometamos errores de diseño. Con el editor de esquemáticos es posible detectar fácilmente dichos errores.

Una manera de hacerlo es generar la **netlist** del proyecto. La netlist es una descripción de las conexiones entre los componentes del esquemático, que se genera a partir del mismo y se almacena en un fichero binario con extensión ALB. Aunque la netlist se genera automáticamente cuando se ejecuta una simulación funcional (ver siguiente capítulo), se puede crear manualmente. Esto permite verificar si el circuito es correcto en cualquier momento, ya que si existen conexiones erróneas se detectarán durante la generación de la netlist.

Para generar la netlist hay que seleccionar la entrada de menú **Options**→**Create Netlist (SHIFT F2)**. Los errores que se detecten aparecerán en la ventana de mensajes del **Project Manager**.

Sin embargo la netlist tan solo detectará errores graves de conexionado. Para realizar un chequeo más complejo del circuito se puede realizar un test de integridad o **integrity test**. Este test realiza un completo análisis de la netlist del proyecto. No crea la netlist pero detecta todos los errores de diseño e inconsistencias. De nuevo, todos los errores y ambigüedades aparecerán en la ventana de mensajes del **Project Manager**. Para ejecutar el test bastará seleccionar **Options**→**Integrity test (CTRL F2)**. Se pueden cambiar las opciones del test de integridad, seleccionando **Options**→**Integrity test Options**. Para ver el significado de cada opción, se recomienda consultar la ayuda del editor de esquemáticos.

E2.2. Diseño Jerárquico

El diseño jerárquico nos será de gran utilidad para el desarrollo de circuitos complejos en los que el problema de diseño completo del circuito se aborda de una manera más cómoda y limpia si el diseño es descompuesto en una serie de circuitos más simples, que una vez integrados, tendrán la misma funcionalidad exigida al circuito original.

Por tanto vemos que el diseño jerárquico consistirá en la realización de circuitos simples, que una vez terminados y testeados serán la base para la construcción de circuitos más complejos. Dichos circuitos más simples serán incorporados a la librería del proyecto y serán utilizados como si de primitivas normales se tratara (ver sec. E2.1.1).

E2.2.1. Creando un diseño jerárquico

Como ya se ha comentado, para realizar un diseño jerárquico, una vez que se haya finalizado el diseño de un circuito (.SCH), habrá que introducirlo en la librería del proyecto donde lo queramos insertar. Para ello antes tenemos que crear lo que se denomina una **Macro**. Una macro es un trozo de diseño esquemático que se almacena en la librería con su propio símbolo. El símbolo de la macro puede ser insertado en otros esquemáticos, múltiples veces, como cualquier otro elemento de librería.


¿Cómo podemos crear una macro en un proyecto, a partir de un esquemático .SCH que hemos construido previamente?

Lo primero que tendremos que hacer es incorporar el esquemático al proyecto, si aún no pertenece al mismo. Una vez abierto el proyecto en cuestión, abriremos el editor de esquemáticos y seleccionamos la opción de menú **Hierarchy** → **Add Sheet to Project**, buscando el fichero .SCH del esquemático que queremos convertir en macro.

Una vez tengamos el esquemático cargado en el editor tenemos que crear la macro a partir de él. Para ello utilizaremos el menú **Hierarchy** → **Create Macro Symbol from Current Sheet**. Se abrirá una ventana en la que se nos preguntará por el nombre del símbolo (**Symbol Name**), que es el nombre con el que aparecerá en la librería del proyecto. Además se nos mostrarán los terminales (entradas y salidas) de la macro. Hay que destacar que la macro se conecta con el nivel superior de la jerarquía a través de sus terminales y que cada terminal del esquemático de la macro se corresponderá con su correspondiente pin en el símbolo de la macro. A la hora de construirla se creará la netlist del esquemático, lo que nos servirá para comprobar de nuevo que el esquemático no contiene errores.

El símbolo asociado a la macro se crea automáticamente, siendo un rectángulo con los pines que representan entradas en el lado izquierdo y los que representan salidas en el derecho. La disposición de estos pines podrá ser variada posteriormente.

¿Cómo se utiliza una macro previamente creada?

Para insertar una macro en cualquier esquemático del proyecto al cual se ha añadido dicha macro, bastará con insertarla como se hacía para cualquier elemento de librería común. Pulsaremos sobre  y buscaremos el nombre que le hemos dado a la macro, y colocaremos tantas instancias de la misma como queramos.

También se puede insertar nuestra nueva macro en sustitución de alguna que ya exista en nuestro proyecto. Por ejemplo, supongamos que tenemos construido un proyecto que implementa un banco de registros basándose en una macro **REG1** que implementa un solo registro. Si quisiéramos sustituir el registro base **REG1** por otra implementación nueva (macro **REG2**), no haría falta borrar todas las instancias de la macro **REG1** y añadir en su lugar las de **REG2**, sino que podemos utilizar la opción **Replace Symbol** para cambiarlas todas a la vez. Para realizar esta acción, basta con pulsar con el botón secundario del ratón sobre la macro a sustituir (**REG1** en nuestro ejemplo), y seleccionar la

opción **Replace Symbol**. En la ventana que nos aparece, como la mostrada en la figura E2.9, se nos pregunta si queremos reemplazar todas las apariciones de dicha macro (**all**) o sólo la que hemos seleccionado (**selected**). Además aparece el nombre de la macro a sustituir (REG1) y en una lista desplegable se tendrá que seleccionar por que macro de las librerías del proyecto queremos sustituirla (por tanto REG2 deberá haber sido incorporada como macro a la librería previamente). Una vez seleccionadas estas opciones y pulsando sobre el botón de **OK**, las instancias de la macro seleccionada/s serán sustituidas por otras tantas de la nueva macro elegida.

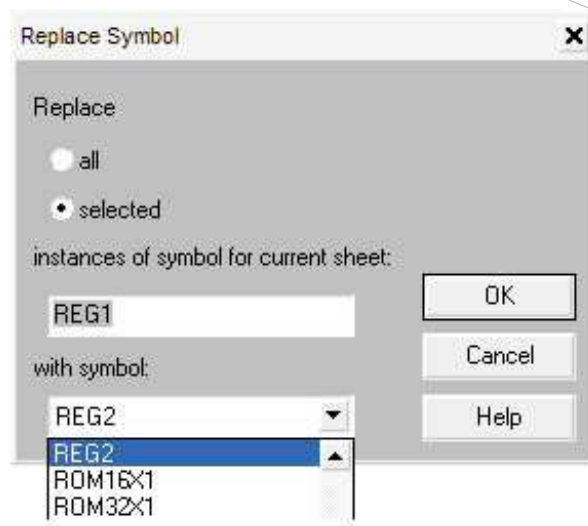



Figura E2.9: Replace Symbol.

Es posible modificar la forma del símbolo de una macro, simplemente seleccionándola, pulsando con el botón derecho del ratón y abriendo el editor de símbolos (**Symbol Editor**). Desde este se podrá cambiar la forma del símbolo y mover de sitio los pines de entrada y salida.

Una vez insertadas las macros en nuestro diseño es posible entrar en ellas directamente desde el esquemático que las contiene. Basta con pulsar el botón  y a continuación hacer doble clic con el botón izquierdo sobre la macro que queramos abrir. Esta operación abrirá el esquemático correspondiente a dicha macro. Para volver al esquemático padre, bastará con pulsar con el botón derecho sobre el fondo del área de trabajo y seleccionar **Hierarchy Pop**.

E2.3. Creando nuevos elementos con la herramienta LogiBLOX

La herramienta LogiBLOX sirve para crear elementos de una manera rápida y sencilla. Esta herramienta nos va a ser de gran utilidad para crear las memorias ROMs que necesitemos en nuestros diseños, pudiendo cambiar su contenido fácilmente.

Para acceder a esta herramienta basta con seleccionar, en el **Project Manager**, la entrada de menú **Tools**→ **Design Entry**→ **LogiBLOX Module Generator** o bien directamente desde el editor de esquemáticos en el menú **Tools**→ **LogiBLOX Module Generator**. Se abrirá una nueva ventana similar a la mostrada en la figura E2.10.

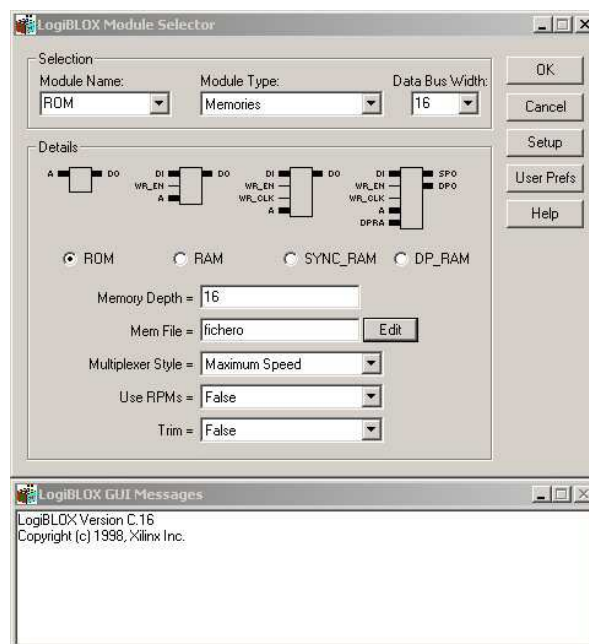


Figura E2.10: LogiBLOX Module Generator.

En realidad aparecen dos ventanas, una en la que definiremos las propiedades del nuevo elemento que queremos crear (**LogiBLOX Module Selector** en la parte superior) y otra en la que aparecerán los mensajes informándonos de la correcta o incorrecta creación del módulo (**LogiBLOX GUI Messages** en la parte inferior).

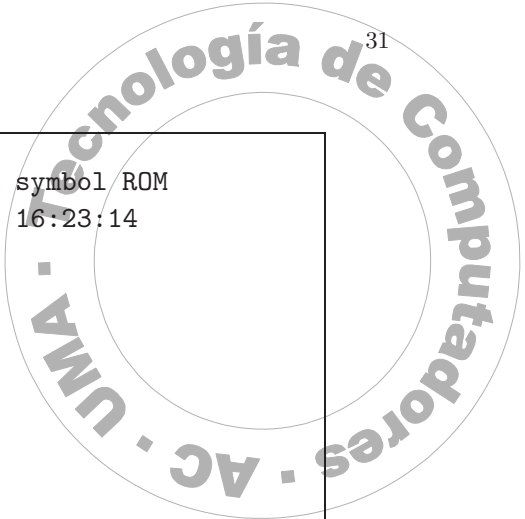
Lo primero que deberemos hacer es seleccionar el tipo de elemento que queremos crear. Para ello utilizaremos el campo **Module Type**, pudiendo seleccionar entre contadores, registros, sumadores, multiplexores, puertas lógicas, memorias, etc. Una vez determinado el tipo de elemento, la parte inferior de la ventana **LogiBLOX Module Selector** cambiará para mostrar los detalles del tipo de elemento que hayamos seleccionado (en la figura E2.10 se ha seleccionado un elemento de memoria, **Memories**). Tenemos que darle un nombre al elemento que estamos creando en **Module Name** y definir el ancho del bus de datos que va a poseer en **Data Bus Width**.

En el ejemplo de la figura E2.10 se ha querido construir un módulo de memoria ROM de 16 posiciones de 16 bits cada una. Se le ha puesto como nombre **ROM** y hemos puesto un ancho de bus de datos de 16 bits, como se quería.

En la parte inferior de la ventana (**Details**) hemos seleccionado los detalles particulares de la memoria que queremos crear. En concreto de todos los posible tipos de memorias que permite crear (ROM, RAM, RAM síncrona, etc.) hemos seleccionado **ROM** que es el que queríamos.

En el campo **Memory Depth** hemos puesto la profundidad de la memoria, es decir, el número de palabras que contiene, en nuestro ejemplo 16. De los demás campos, tan solo nos interesa el denominado **Mem File** en el cual ponemos el nombre del fichero asociado con la memoria y que almacenará el contenido de la misma. En nuestro ejemplo lo hemos llamado **fichero**. Una vez introducido el nombre del fichero asociado a la memoria, si pulsamos el botón **Edit**, se abrirá una nueva ventana, que básicamente es un editor de texto, con el contenido de dicho fichero. La extensión de los ficheros asociados a las memorias es **.MEM**. Por tanto el nombre del fichero editado en nuestro ejemplo sería **FICHERO.MEM**. Si hemos puesto el nombre un fichero que no existe, la herramienta creará uno automáticamente con las secciones adecuadas, preparado para que introduzcamos los datos. Por ejemplo, para la memoria de la figura E2.10, el contenido creado automáticamente para el fichero es el mostrado en la figura E2.11.

Básicamente podemos diferenciar dos secciones, la de cabecera **Header Section** y la de los datos **Data Section** (las líneas que comienzan por “;” son comentarios). En la cabecera se definen las características del módulo de memoria asociado al fichero. Con la palabra clave **RADIX** se define la base de los números que aparecerán a continuación. Así en el ejemplo **RADIX 10** indica que todos los números que aparecen a continuación (hasta el siguiente RADIX) están en base 10. A continuación define el número de palabras de la memoria (**DEPTH 16**) y el número de bits por palabra (**WIDTH 16**). Con



```
;
; memfile fichero.mem for LogiBLOX symbol ROM
; Created on Friday, June 27, 2003 16:23:14
;
; Header Section
RADIX 10
DEPTH 16
WIDTH 16
DEFAULT 0
;
; Data Section
; Specifies data to be stored in different addresses
; e.g., DATA 0:A, 1:0
RADIX 16
DATA
; end of LogiBLOX memfile
```

Figura E2.11: Fichero para la generación de una ROM con LogiBLOX.

DEFAULT 0 establecemos que por defecto, el contenido de las posiciones de memoria es 0.

En la sección de los datos, encabezada con la palabra clave **DATA**, se especificará el contenido de las posiciones de memoria que se deseen (no tienen por que especificarse todas, las que no se especifiquen tendrán el valor establecido en **DEFAULT**). Justo antes de **DATA** vemos que se ha puesto un **RADIX 16**, estableciendo que todas los números que aparezcan a continuación estarán en base 16. Esto se hace normalmente ya que pasar el contenido binario de una memoria a base 16 es muy fácil.

Para especificar el contenido de posiciones de memoria consecutivas basta con escribir el contenido (en hexadecimal en este caso) de cada posición, separado por comas (ya sea en la misma línea o en distintas). También se puede especificar el contenido de posiciones de memoria no consecutivas. Para ello bastará con indicar la posición de memoria, seguida de dos puntos “:” y a continuación el contenido de dicha dirección.

Por ejemplo, supongamos que queremos especificar el contenido de algunas posiciones de la memoria del ejemplo de la figura E2.10 (ROM16x16) según la siguiente tabla:

Pos. memoria	Contenido binario	Pos. memoria	Contenido binario
0 ₁₆	0010011010100101	6 ₁₆	1111001010000010
1 ₁₆	0010010011110011	9 ₁₆	1101010100100111
2 ₁₆	0001001011111100	F ₁₆	0000111111110000

en este caso el contenido de la sección **DATA** de nuestro fichero quedaría como:

```
; Data Section
; Specifies data to be stored in different addresses
; e.g., DATA 0:A, 1:0
RADIX 16
DATA
26A5,
24F3,
12FC,
6:F282,
9:D527,
F:0FF0
; end of LogiBLOX memfile
```

También se podría expresar todo ese contenido en una única línea:

```
DATA 26A5, 24F3, 12FC, 6:F282, 9:D527, F:0FF0
```

Una vez terminada la especificación del contenido de la memoria y grabado en el fichero, cerraríamos la ventana del editor de texto y volveríamos la del **LogiBLOX Module Selector**. Para indicar a la herramienta que hemos finalizado la especificación de nuestra memoria, pulsaríamos el botón **OK**.

En ese momento empezará la generación del elemento (en nuestro ejemplo la memoria ROM). Si hay algún problema aparecerá un mensaje en la ventana de mensajes indicándolo y tendremos que solucionarlo. Si no hay ningún problema, se generará el módulo correctamente y se insertará en la librería del proyecto. A partir de ese momento podremos usarlo como cualquier otro elemento de librería.

Los propiedades de un elemento creado por medio de esta herramienta se pueden cambiar en cualquier momento. Bastará con pulsar con el botón derecho del ratón sobre dicho elemento (una vez insertado en un esquemático) y seleccionar la opción **Edit LogiBLOX Symbol**. Se abrirá de nuevo la ventana mostrada en la figura E2.10, desde la cual, por ejemplo, podremos

editar el contenido de las posiciones de nuestra memoria ROM (editando el fichero .MEM).

NOTA: Siempre que queramos cambiar el contenido de una memoria creada con la herramienta **LogiBLOX**, tendremos que editar el contenido del fichero .MEM asociado a dicho elemento de memoria pero siempre desde dentro de la ventana **LogiBLOX Module Selector** (fig. E2.10) y **nunca editaremos directamente el fichero .MEM**. Esto es así ya que el contenido de dicho fichero se asocia a la memoria en el momento que pulsamos **OK** en el **LogiBLOX Module Selector**, creándose e introduciéndose de nuevo el elemento en la librería.


E2.4. Ejemplo

A modo de recapitulación de todo lo visto hasta ahora en el manejo de la herramienta del laboratorio, vamos a desarrollar el proceso completo necesario, desde que se introduce el esquemático de un circuito, previamente diseñado, hasta que se simula para comprobar que no existen errores (en nuestro caso además nos servirá para conocer más a fondo el funcionamiento de los elementos estudiados en teoría). En este ejemplo se creará el esquemático, postergando su simulación para el ejemplo del próximo capítulo. El diseño que se plantea es un sistema que contiene 4 buses con información, del que queremos volcar uno de ellos a un bus de salida. Se usará un decodificador para seleccionar cuál de ellos. En la figura E2.12 se puede ver el esquemático completo. Para revisar el diseño jerárquico, crearemos como elemento nuevo de librería un decodificador 2 a 4 con entrada de habilitación.

E2.4.1. Creación del proyecto e introducción del esquemático

Al abrir la aplicación deberemos indicar que queremos crear un proyecto nuevo (el decodificador). Le daremos el nombre de *ejemplo* y dejamos las demás casillas al valor que tienen por defecto.

A continuación abrimos el editor de esquemáticos con **Tools**→ **Design Entry**→ **Schematic Editor** (o pulsando sobre el botón que representa el editor de esquemáticos en la ventana del **Project Flowchart**) para la introducción del decodificador. El diagrama del mismo aparece representado en la figura E2.13.

Comencemos por colocar las puertas. Al seleccionar el acceso directo de la barra de herramientas  o el menú **Mode**→**Symbols (F3)** se nos desplegará la ventana con todos los elementos disponibles. En nuestro caso

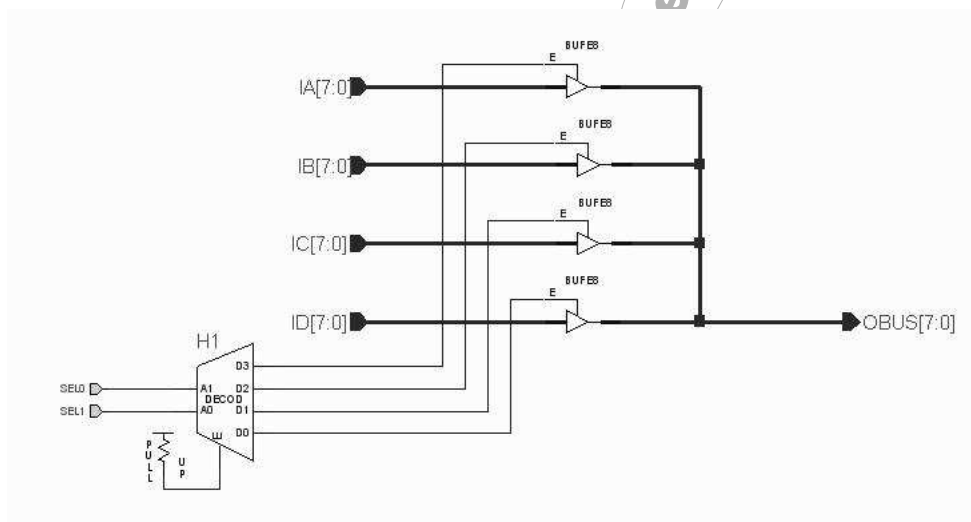


Figura E2.12: Circuito que se implementará en el ejemplo.

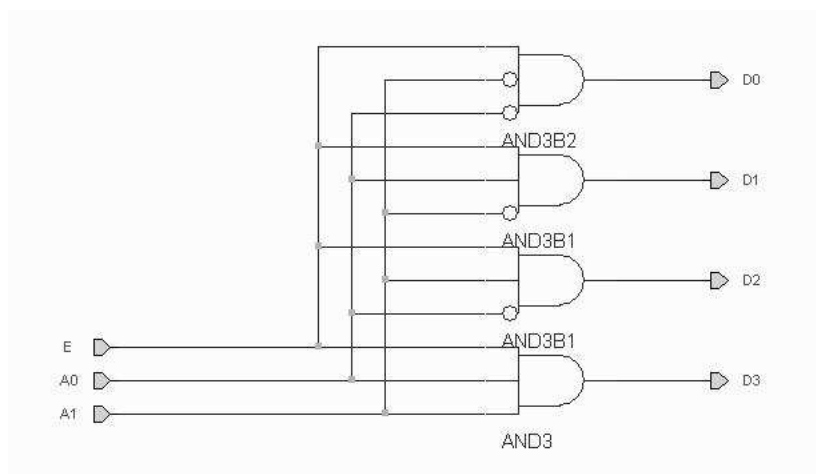


Figura E2.13: Esquemático del decodificador 2 a 4 implementado en el ejemplo.

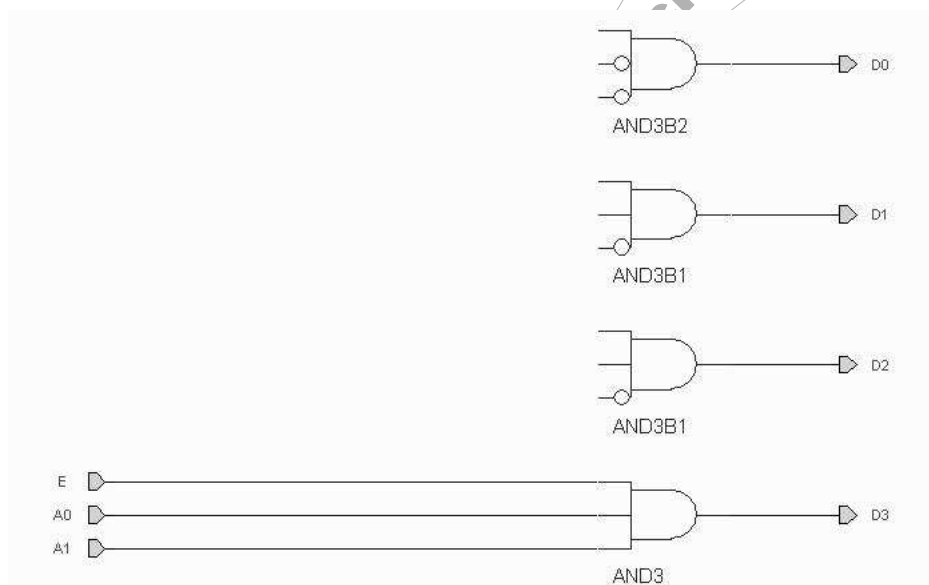




Figura E2.14: Aspecto del esquemático una vez colocados las puertas y los pines.

debemos usar: una AND3 (AND de 3 entradas), dos AND3B1 (AND de 3 entradas con una negada), y una AND3B2 (AND de 3 entradas con dos negadas).

Una vez tengamos dispuestas las puertas según la figura, procederemos a incluir los pines de entrada (E, A0 y A1) y de salida (D0-3) del decodificador. El botón de **Hierarchy Connector** () nos permite ir añadiéndolos, tras indicar su nombre y tipo (entrada o salida). Para las señales A0-1 y D0-3 podemos valernos de la opción **Repeat** para simplificar el trabajo (debemos de dar como nombre inicial A0 y D0 respectivamente para que empiece a numerar por 0). Cuando terminemos de colocarlos pulsaremos ESC. Automáticamente se pasará al modo de colocación de cables, por lo que podemos ir pinchando con el ratón sucesivamente en los pines y en las patillas a los que van unidos directamente, para que se trace una línea de cable entre ellos. Si no nos gusta como ha quedado la disposición, podemos pasar al modo seleccionar y arrastrar (**Mode→Select and Drag (F2)**) para reubicar los elementos sobre la hoja del esquemático. A estas alturas el esquemático debería tener un aspecto similar al de la figura E2.14.

El resto de conexiones se harán pasando al modo de colocación de ca-

bles  (**Mode**→**Draw Wires (F4)**). Para ello pincharemos sobre los dos puntos a unir (patilla y cable), y el programa añadirá automáticamente los puntos de soldadura necesarios. Con esto habremos concluido la introducción del esquemático del decodificador. Para finalizar, lo grabaremos con el nombre `decod2-4.sch`

E2.4.2. Creación de un elemento de librería

El siguiente paso es convertirlo en un elemento más de librería (macro), para poderlo incluir posteriormente en nuestros diseños (que serán, por tanto, jerárquicos). Para ello, teniendo abierto el esquemático, hacemos **Hierarchy**→**Create Macro Symbol from Current Sheet**. Cuando se abra el cuadro de diálogo correspondiente le daremos el nombre `DECOD2-4`, que es con el que aparecerá en la librería. Al construir la macro se crea el *netlist* del esquemático, lo que nos va a permitir asegurarnos de que no tiene errores. Una vez listo, nos preguntará si queremos editarlo, por si pensamos hacer alguna modificación.

A partir de este momento, el nuevo elemento aparecerá en la lista de dispositivos que pertenecen a la librería del proyecto, pudiendo ser usado como uno más. Creemos una nueva hoja de esquemático. Al abrir la ventana de **SC Symbols (F3)**, si desplegamos la librería **EJEMPLO** podremos localizar nuestro nuevo componente, `DECOD2-4`. Seleccionarlo y colocarlo sobre la nueva hoja.

Vamos a cambiarle el aspecto, para que su símbolo se parezca más al empleado para un decodificador. Pulsando sobre el símbolo actual el botón derecho del ratón, abrimos el menú contextual y seleccionamos **Symbol Editor**. Se abre una nueva ventana, donde podemos modificar entre otras cosas su aspecto. En la figura E2.15 se aprecia la apariencia por defecto del símbolo.

Utilizando los botones de dibujo que aparecen en la barra de herramientas o las opciones bajo el menú **Symbol** le vamos a dar un nuevo aspecto. Para facilitar el dibujo comenzaremos por activar la rejilla guía (**View**→**Grid**). Ampliamos la imagen el máximo que nos permita el zoom con **View**→**Zoom In (Ctrl +)**. Redimensionamos la caja para que se ajusten mejor los pines a la nueva forma, reduciendo su tamaño horizontalmente en 3 cuadrículas y en una vertical. Con **Symbol**→**Line** dibujamos el trapecio típico de estas puertas y con la herramienta **Symbol**→**Select** reubicamos los pines en las posiciones que aparecen en la figura E2.16. Solo nos queda eliminar la antigua caja, deseleccionando la opción **Symbol**→**Autobox** y colocarle el texto `DECOD`, con **Symbol**→**Text**, teniendo la precaución de hacer el tamaño de la letra pequeño (pulsando el botón de **Settings** y seleccionando **Small** en **Font Size**).

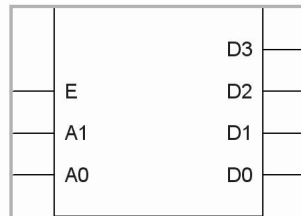


Figura E2.15: Aspecto inicial del decodificador, como el de un elemento más de librería.

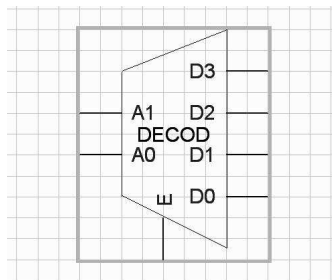



Figura E2.16: Aspecto final del decodificador, más parecido al usado frecuentemente.

E2.4.3. Completando el circuito

Ya solo nos resta terminar de dibujar el circuito de más alto nivel en la jerarquía. Para movernos por los diferentes niveles de la jerarquía, bastará con activar la opción **Hierarchy**→**Hierarchy Push (Ctrl+H)** ó pulsar el botón y hacer doble clic con el ratón sobre la macro en la que queremos entrar. Para salir bastará con hacer lo contrario, **Hierarchy**→**Hierarchy Pop (Ctrl+I)**.

Los elementos que van a funcionar como buffers triestado de 8 bits se denominan BUFE8. Colocaremos cuatro. Necesitamos además una resistencia de *pull-up*, llamada PULLUP, para conectarla a la entrada de habilitación del decodificador. Como la librería contiene gran cantidad de elementos, puede ser más cómodo usar el cuadro de texto inferior de la ventana **SC Symbols**, de modo que al ir escribiendo el nombre nos desplazaremos más rápidamente por los componentes que si usamos las barras de desplazamiento. Cuando tengamos

estos elementos los uniremos en el modo cable y añadiremos los conectores de jerarquía de entrada SEL0-1.

La forma de proceder para añadir los buses es similar a la de los cables, pero trabajando en **Mode→Draw Buses (F5)** (pulsando el botón ). La principal diferencia radica en que cuando terminamos de dibujar un bus, con un doble clic del ratón, nos pedirá información del nombre, líneas que incluye (de la 0 a la 7 en nuestro caso) y tipo de bus (entrada para IA, IB, IC e ID y salida para OBUS). De modo análogo al trazado de cables, al pinchar con un bus sobre otro cuando estamos dibujando, nos añade automáticamente el punto de soldadura. Con esto habremos terminado el esquemático, por lo que salvaremos el trabajo para poderlo simular en el próximo capítulo.



Durante el proceso de dibujo, puede ser que queden rastros de elementos inexistentes (por que los borramos o no se llegaron a dibujar). La forma de eliminarlos es mediante la opción de redibujado, **Display→Redraw (F10)**.

E3

La simulación del diseño




E3.1. Arrancando el simulador lógico

Una vez diseñado el circuito, testeado y generado el *netlist* podemos pasar a la simulación del mismo, para lo cual debemos arrancar el simulador lógico. Esto se puede hacer desde el menú del editor de esquemáticos con **Tools** → **Simulator**¹ o haciendo clic en el botón . También podemos abrir el simulador lógico desde el gestor de proyectos con el botón .

El elemento fundamental del simulador es la ventana de formas de onda o cronogramas (*waveform window*), que se visualiza por defecto cuando abrimos el simulador. Si la ventana de forma de ondas no apareciera, o bien se hubiera cerrado, se puede abrir una desde el menú del simulador con **Window** → **Waveform Viewer**.

Cuando estamos realizando modificaciones sobre el circuito no se nos debe olvidar ir actualizando la simulación desde el editor de esquemáticos mediante **Tools** → **Update Simulation**, ello supone el chequeo del esquemático así como la generación del *netlist*.

¹Si el diseño se ha construido jerárquicamente se puede simular un bloque jerárquico (macro) individualmente. Para ello navegaremos en el editor de esquemáticos y entraremos en el bloque (macro) a simular (usar botón ). Una vez abierta la macro arrancaremos el simulador con **Tools** → **Simulate Current Macro**. Pasaremos entonces a simular exclusivamente el bloque jerárquico en cuestión, y no el resto del circuito en el cual se halle. Esta misma operación se puede realizar desde el simulador mediante **File** → **Simulate Single Component**.

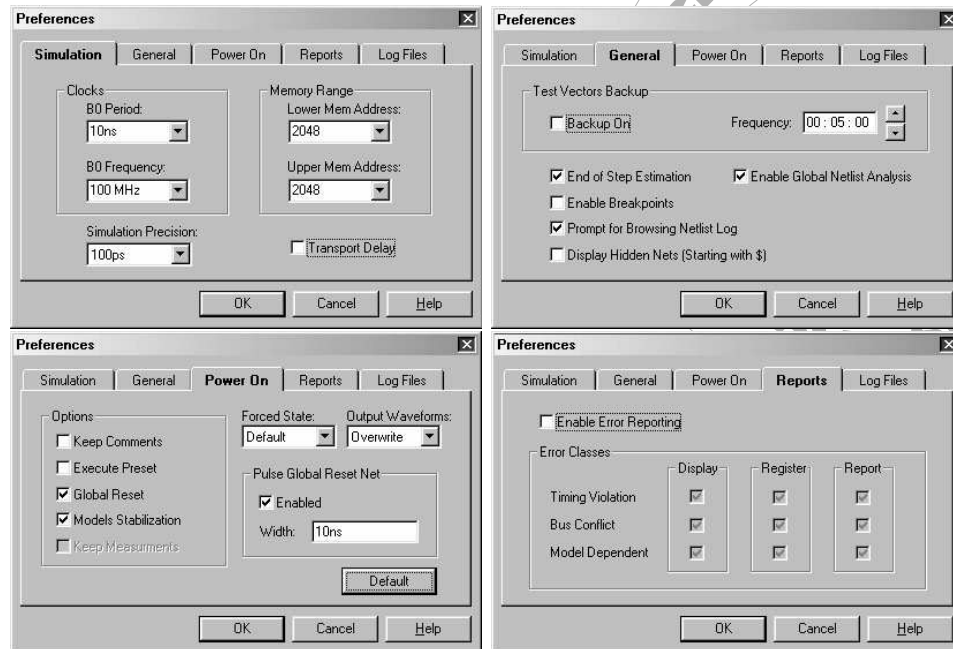



Figura E3.1: Configuración del simulador.

E3.2. Configurando la simulación

Existen varios aspectos configurables de los que dependen el transcurso de la simulación. Los parámetros de configuración por defecto se pueden modificar accediendo al panel de configuración que se abre desde el menú del simulador con **Options** → **Preferences**, o bien presionando .

En la figura E3.1 se muestran los parámetros configurables desde dicho panel y los valores que se recomienda fijar de cara a las prácticas que se han de realizar. Mencionamos a continuación el valor a asignar a algunos de los parámetros:

- Pestaña *Simulation*. Dejaremos el valor de la frecuencia del reloj y la precisión de la simulación (cuanto de simulación) por defecto. Observemos que mientras menor es el cuanto de simulación, mayor precisión tendrá la simulación realizada. Desactivaremos el retardo de las líneas de transmisión (*transport delay*).
- Pestaña *General*. Desactivaremos la escritura automática de las formas

de onda (*Test Vector Backup*). Cuando esta opción está activa, se fija cada cuánto tiempo se guarda periódicamente los resultados de la simulación. Si está activa podría sobrescribir sin darnos cuenta resultados obtenidos que ya hubiéramos salvado en disco.

- Pestaña *Power on*. Dejaremos la configuración por defecto (*Default*).
- Pestaña *Reports*. Desactivaremos la casilla que habilita el informe de errores durante la simulación (*Enable Error Reports*). Cuando se trabaja con diseños de cierta complejidad, tener activa esta opción da lugar a numerosos *pop-ups* mostrando avisos de potenciales errores que resultan molestos. No obstante podría ser útil activar el informe de errores durante la simulación si se sospecha de que algo no está marchando bien.

E3.3. Visión general de la simulación

En la figura E3.2 se muestra una visión general del simulador en la que se ha indicado sucintamente la funcionalidad de los principales botones y opciones, tanto del simulador como del visualizador de formas de onda. Observemos que cuando dejamos quieto durante unos segundos el puntero del ratón sobre cada botón aparece un globo indicando la funcionalidad de dicho botón.

La simulación se realiza paso a paso, para lo cual deberemos:

- Elegir el tipo de simulación. Existen 4 modos de simulación posibles: *functional*, *timing*, *glitch* y *unit delay*, según los tiempos de propagación considerados en la simulación (ver figura E3.3). En la práctica usaremos los modos *functional* o *unit delay*.

El primer modo (*functional*) corresponde a una simulación funcional en la que se considera que los elementos de los circuitos tienen retardo nulo (cero). Lo seleccionaremos cuando queramos chequear que la lógica del circuito funciona correctamente. Este modo puede dar lugar a oscilaciones en caso de existir realimentaciones, no siendo recomendable en estos casos.

El modo *timing*, por el contrario, tiene en cuenta los retardos asociados a los dispositivos del circuito (proporcionados por el fabricante). Por tanto, este modo permite analizar el comportamiento del diseño de una forma más realista, permitiendo observar los transitorios que se producirían en el circuito real².

²El simulador permite modificar a mano los retardos asociados a los componentes de circuito desde la entrada de menú **Device** → **Edit Timing Specification**.

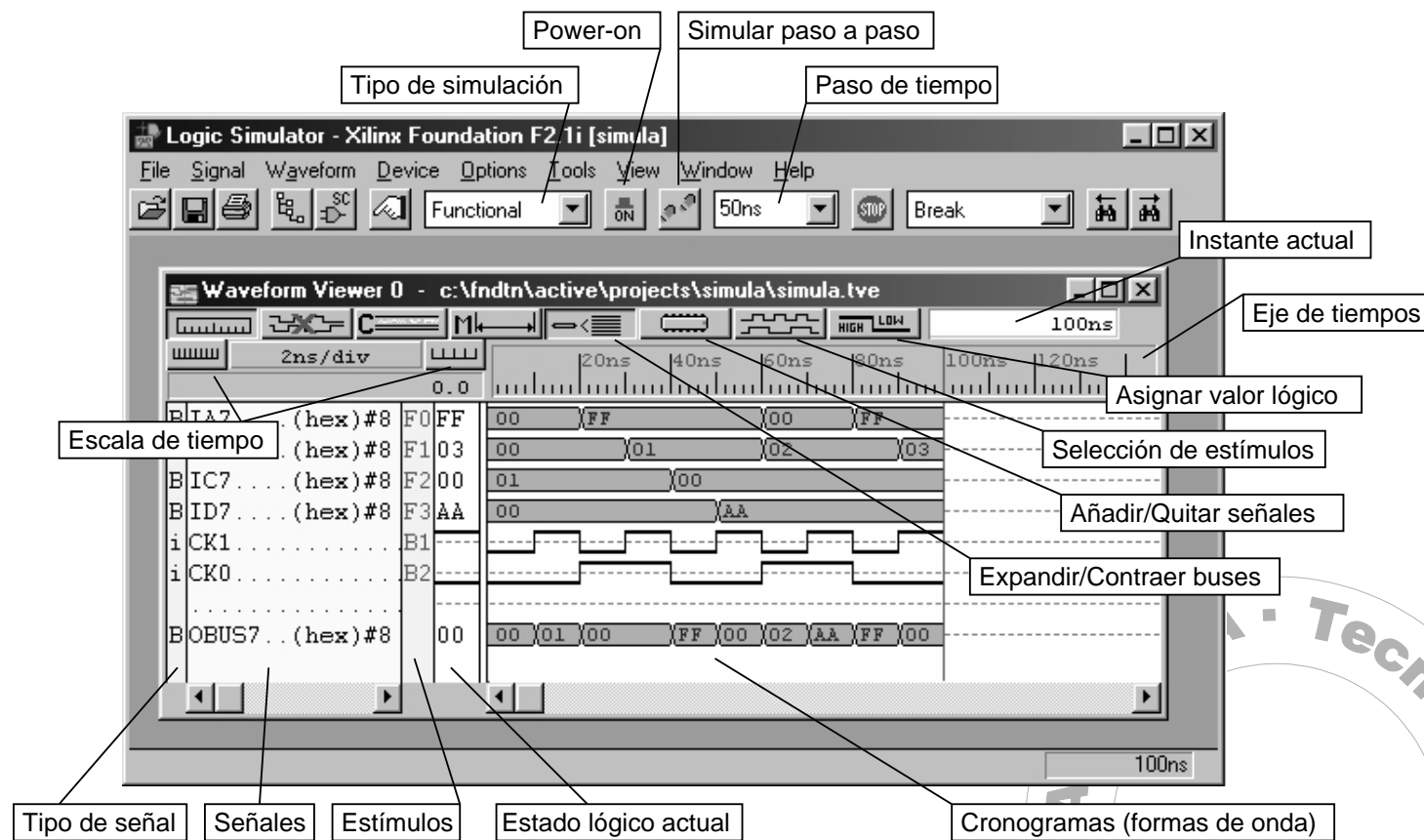


Figura E3.2: Visión general del simulador y del visualizador de formas de onda.

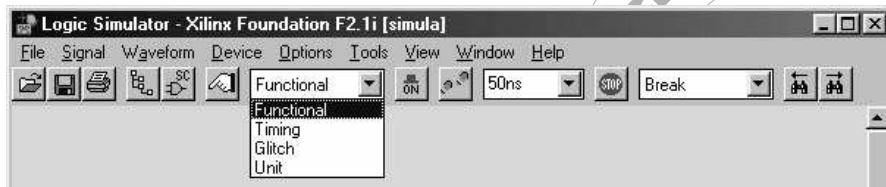




Figura E3.3: Modos de simulación.

El modo *unit delay* asigna un retardo unitario a todos los componentes del circuito, permitiendo ver también el efecto del tiempo de propagación. Este retardo unitario asignado es igual al cuanto de simulación asignado en la configuración.

- Inicializar la simulación al origen de tiempos (botón *Power-on*: ).
- Seleccionar el valor del paso de tiempo. Por ejemplo, en la figura E3.2 se ha seleccionado un paso de 50ns.
- Hacer avanzar la simulación paso a paso (botón .

Tal como se observa en la figura E3.2 distinguimos tres zonas diferenciadas en la ventana de cronogramas, que son:

Señales. En esta zona se muestra el subconjunto de señales del circuito que queremos visualizar. Corresponden con etiquetas que se han definido en el esquemático (entradas, salidas, nodos internos, ...) y pueden ser buses o señales individuales.


Estímulos (estimuladores). Indica cuál es el estímulo que se está aplicando a la señal, esto es, cuáles son los valores que va a tomar en el tiempo. Sólo tiene sentido en nuestro caso para las señales de entrada del circuito. Los indicadores de estímulos se explican en la sección E3.5.



Cronograma (formas de onda). Muestra la evolución en el tiempo de los valores de las señales. En caso de buses podremos ver su valor en binario, octal, decimal y hexadecimal, como se describe más adelante.

En cada una de estas tres zonas podemos abrir un menú contextual haciendo clic con el botón derecho del ratón.

Además, para cada señal se proporciona información adicional como:

- Indicador del tipo de señal: bus (B), individual (i).
- Estado lógico actual. Para cada señal se muestra su valor lógico en el instante actual de la simulación, cuando ya ha transcurrido algún paso de tiempo.

En la zona de cronograma es posible visualizar una regla graduada que representa el eje de tiempos. Ello puede ser muy útil por ejemplo en el caso de que se desee realizar mediciones entre eventos que se muestren en la simulación, así como para cuantificar el efecto de los retardos. El eje de tiempos se puede mostrar/ocultar con un clic sobre el botón .

Así mismo podemos escalar el eje de tiempos, lo que permite acercarnos o alejarnos en el cronograma (zoom) visualizando las señales con más o menos detalle. Esto se realiza haciendo clic sobre los botones alejar o acercar ( ). Observamos que la escala correspondiente al zoom actual se muestra en unidades de tiempo por división del eje de abscisas (por ejemplo 2ns/div).

En los siguientes apartados se explica cómo trabajar con cada uno de los componentes que intervienen en la simulación: señales, estímulos y formas de onda.

E3.4. Selección de señales

Denominaremos señales a las líneas lógicas del circuito a las cuales haremos mención mediante las etiquetas que se hayan empleado en el diseño del esquemático. Las señales incluyen las entradas y salidas del circuito, los nodos internos que se hayan etiquetado, así como los pines de los componentes.

El menú relacionado con las señales se despliega bien con la entrada **Signal** del simulador, o bien con el menú contextual en la zona de señales de la ventana de visualización (haciendo clic botón derecho del ratón en dicha zona). En la figura E3.4 se muestra las dos formas de acceder al menú de señales.

E3.4.1. Incorporando al cronograma las señales a visualizar

En primer lugar veamos la forma en que se incorporan al cronograma las señales de nuestro circuito que queremos visualizar. En el simulador ello se realiza con la entrada añadir señales del menú (**Signal** → **Add Signal**) o

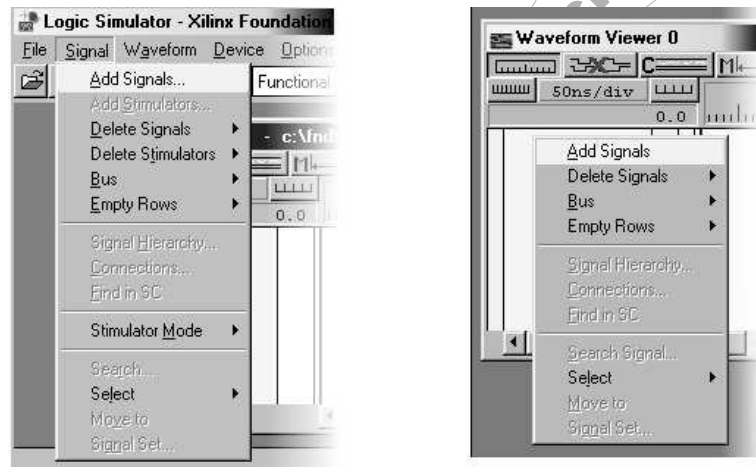


Figura E3.4: Trabajando con señales, desde la barra del simulador y menú contextual en la zona de señales (clic en botón derecho).

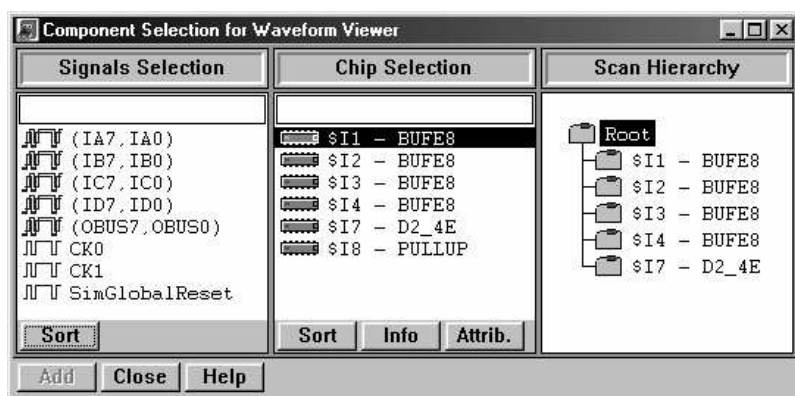




Figura E3.5: Panel de selección de señales.

presionando  en la ventana de simulación. Aparecerá entonces el panel de selección de señales cuyo aspecto es el mostrado en la figura E3.5.

En dicho panel podemos seleccionar/deseleccionar qué señales del circuito queremos visualizar. Las señales se seleccionarán o deseleccionarán haciendo doble clic con el botón izquierdo del ratón. Observemos que las señales pueden ser individuales o buses. Cuando una señal está seleccionada aparece indicado con . También se puede seleccionar haciendo clic sobre la señal y

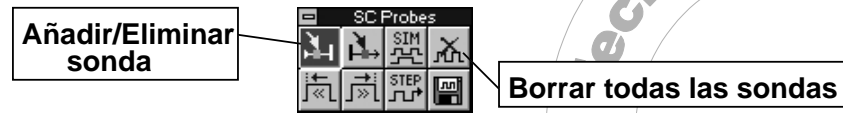





Figura E3.6: Sondeando señales en el esquemático.

pulsando el botón **Add**. Al seleccionar una señal ésta aparece visible en la ventana de simulación.

Otra manera de seleccionar las señales a visualizar es pinchando sondas (*probes*) sobre las señales en el editor de esquemáticos. Esta forma puede ser muy cómoda si se ha localizado la señal a analizar en el esquemático. La herramienta de sondas (*probes tool*) se lanza desde el editor de esquemáticos presionando en . Con ello abrimos la ventana *SC Probes* cuya descripción se muestra en la figura E3.6.

En la herramienta *SC Probes* presionando  pasaremos al modo denominado *test point mode*, que nos permite seleccionar señales. El puntero del ratón cambia de aspecto, siendo ahora , para indicar que podemos sondear sobre el esquemático. Es posible sondear cables y buses siempre que tengan asociados una etiqueta. Esto es tan fácil como pinchar sobre la etiqueta que queremos sondear. Observamos que aparece un pequeño display, al lado de la etiqueta sondeada, donde se mostrará el valor de la señal en el instante actual de la simulación. Podemos eliminar la sonda de una señal ya sondeada sin más que pinchar sobre ese pequeño display, desde el modo sondeo (*test point mode*).

Selección de señales en diseños jerárquicos


En caso de que el diseño sea jerárquico en el panel de selección de señales (figura E3.5) aparecerá un árbol representando la jerarquía en la subventana derecha, etiquetada *Scan Hierarchy* (si no aparece esta subventana presionar el botón **Hierarchy**). El nodo denominado *Root* corresponde a la hoja padre del diseño, de la cual colgarán las hojas jerárquicamente (los chips que forman el circuito también aparecen como hojas de este árbol). Haciendo doble clic en los bloques de la jerarquía irán apareciendo las diferentes señales que la componen.

Observamos, además, que en el panel de selección existe la posibilidad de seleccionar las patillas (pines) de los chips presentes en el circuito. Ello se hace seleccionando, con un doble clic, el chip en la lista etiquetada *Chip Selection*

y seleccionando posteriormente los pines deseados.

En caso de diseño jerárquico, los bloques de la jerarquía (macros) aparecen como chips del circuito, y sus entradas y salidas aparecen como pines de dichas pastillas.

En lugar de usar el panel de selección podemos navegar en el editor de esquemáticos por la jerarquía y utilizar sondas (*probes*) sobre las señales deseadas, tal como se describió anteriormente. En ocasiones puede resultar una forma cómoda de chequear señales pertenecientes a una macro.

También es posible simular un bloque jerárquico (macro) de forma independiente, lo cual puede resultar útil para comprobar la funcionalidad del bloque. Para ello navegaremos por el circuito en el editor de esquemáticos y entraremos en el bloque (macro) a simular (usar botón ). Una vez abierta la macro arrancaremos el simulador con **Tools** → **Simulate Current Macro**, simulándose dicha macro de forma exclusiva, independientemente del resto del circuito.

E3.4.2. Eliminando señales de la visualización

Para eliminar una o varias señales de las que se están visualizando en el cronograma las seleccionaremos³ y desde el menú de señales ó contextual elegiremos **Delete Signals** → **Selected**. También se pueden eliminar todas las que están visualizadas, en un momento dado, eligiendo la entrada de menú **Delete Signals** → **All**. No se debe confundir eliminar las señales que se visualizan con eliminar formas de ondas.

E3.4.3. Buses

Desde la entrada de menú **Signal** → **Bus**, o bien desde el menú contextual de señales, podemos realizar ciertas operaciones en relación al tratamiento de los buses (ver figura E3.7):

- Combinar un grupo de señales individuales en un bus. Esto se hace seleccionando las señales a combinar y combinarlas, una vez seleccionadas, con **Signal** → **Bus** → **Combine**. La operación contraria (o sea, separar las señales) corresponde a **Signal** → **Bus** → **Flatten**.

³La selección de señales sigue la misma filosofía de selección de objetos de windows: botón derecho de ratón selecciona una, Shift+botón izquierdo permite seleccionar un grupo contiguo, y Ctrl+botón izquierdo permite seleccionar un conjunto de señales una a una. Las señales seleccionadas se resaltan en un color diferente

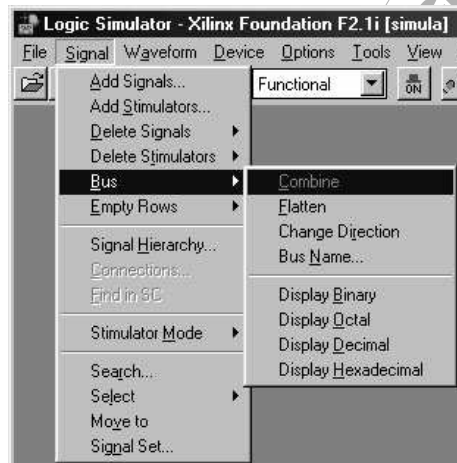



Figura E3.7: Operaciones sobre buses.

- Mostrar los buses definidos en el esquemático como buses ó expandidos como un conjunto de señales individuales. Expandir ó contraer las señales de un bus se realiza haciendo clic en .
- Elegir la base de la representación del valor del bus: binario, octal, decimal o hexadecimal, para lo cual seleccionaremos la opción correspondiente en el menú **Signal** → **Bus** una vez seleccionado el bus sobre el que aplicar esta propiedad.

E3.5. Edición de estímulos y formas de onda

Una vez que hemos elegido las señales a visualizar tenemos que indicar qué valores tomarán en el tiempo las entradas del circuito. Si no asignamos estímulo alguno a una señal, su valor en principio es desconocido (X). Existen varias posibilidades a la hora de establecer la excitación de las entradas de las que mencionamos las siguientes:

1. Uso de **estimuladores** (RECOMENDADO). Los estimuladores proporcionan una serie de estímulos predefinidos y configurables que podemos asociar a cada señal de entrada. La elección de un estimulador para una señal se hará seleccionando la señal y posteriormente haciendo clic sobre el estímulo deseado en el panel de selección de estimuladores.

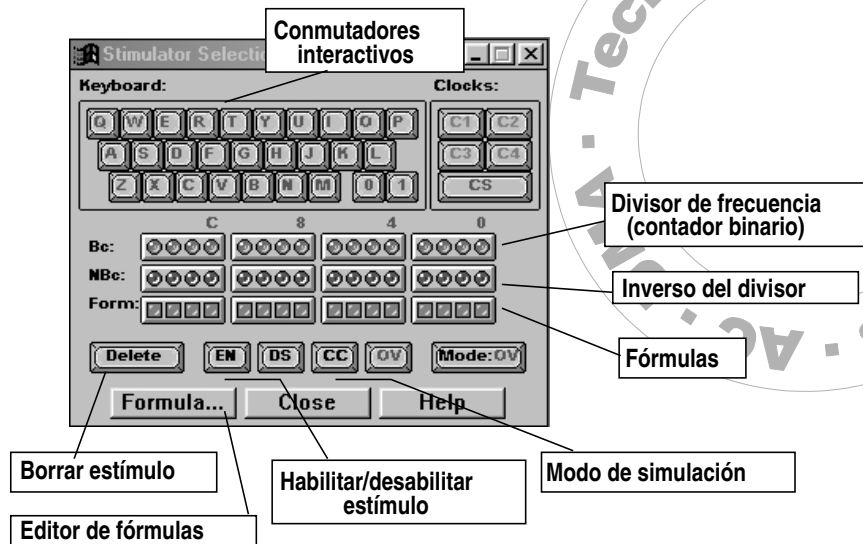



Figura E3.8: Panel de selección de estimuladores.

Dicho panel se abre, desde la ventana de simulación, presionando el botón , o bien, en la entrada de menú **Signal** → **Add Stimulators**. En la figura E3.8 se comenta la funcionalidad de dicho panel.

Podemos asociar a una señal varios tipos de estimuladores:

- ★ **Fórmulas** de usuario (RECOMENDADO). Es la manera más flexible de generar estímulos y asignarlos a las señales, ya sean individuales (1 bit sólo) o buses. Podemos definir hasta 16 fórmulas abriendo para ello el editor de fórmulas haciendo clic sobre **Formula...**. El editor de fórmulas se muestra en la figura E3.9.

Para editar una fórmula la seleccionaremos haciendo doble clic de ratón. Las fórmulas están numeradas desde F0 hasta FF. En la casilla denominada *Edit formula* teclearemos la fórmula que deseemos y la aceptaremos bien pulsando retorno o bien pulsando el botón **Accept**. Podemos editar una por una las dieciséis fórmulas disponibles, modificándolas si fuera necesario.

La validación de los cambios realizados se realiza al cerrar el editor de fórmulas haciendo clic sobre el botón **Close**. Si cerráramos la ventana directamente, ¡los cambios no se guardarán!

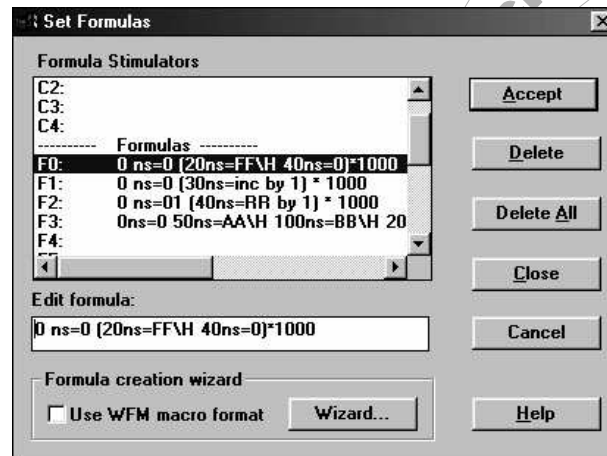


Figura E3.9: Editor de expresiones (fórmulas) lógicas.

En la tabla E3.1 se resume la sintaxis que hemos de seguir a la hora de definir un estímulo mediante una fórmula. Básicamente consiste en especificar el tiempo transcurrido desde la transición anterior y el valor lógico a tomar. En la tabla E3.2 se muestran algunos ejemplos de estímulos y su forma de onda correspondiente.

Podemos observar que en la ventana de cronograma, una vez asignada una fórmula, se indica el número de dicha fórmula a la derecha de la señal correspondiente (F0, F1, F2, etc.).

- ★ Bits de un **divisor de frecuencia**. Están disponibles los bits de un contador binario de 16 bits que actúa como divisor de frecuencia. Estos bits se pueden asignar a señales individuales. La frecuencia de actualización del contador es la elegida durante la configuración en el menú *Preferences* (ver sección E3.2). También está disponible como estímulo el negado de los bits del contador binario. La principal utilidad de este tipo de estímulos es la generación de señales de reloj.

Podemos observar que en la ventana de cronograma, una vez asignada un bit del divisor de frecuencias se indica el número de bit correspondiente (B0, B1, B2, ... hasta BF para los bits afirmados ó N0, N1, N2, ... hasta NF para los negados, siendo B0 y N0 los menos significativos respectivamente.).

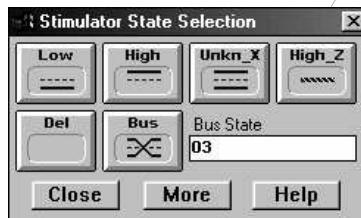




Figura E3.10: Panel de selección de estado de una señal para la simulación interactiva.

★ **Conmutadores** interactivos. Estos estímulos están orientados a una simulación interactiva. Asociamos una tecla del teclado (ver figura E3.8) a una señal individual. Pulsando dicha tecla conmutamos el valor lógico de la señal (0 ó 1) en el instante de simulación actual, eligiendo por tanto su valor para el siguiente paso de tiempo de la simulación.

Se observa que si asociamos un conmutador interactivo a una señal, se indicará en la ventana de cronograma la tecla correspondiente.

2. Asignación directa de **fórmulas**. Para ello, seleccionar una señal haciendo clic con el botón izquierdo del ratón, activar la entrada de menú **Waveform** → **Insert Formula** y teclear la fórmula que queremos asociar a la señal. Así habría que editar las señales una a una. El indicativo Cs se muestra en este caso en la subventana de estímulos.
3. **Selección directa** de estado en el instante actual. Orientado a una simulación interactiva, permite poner una señal en un estado lógico. Para ello seleccionamos la señal de la que queremos cambiar el estado y a continuación presionamos , con lo que abriremos el panel de selección de estado (figura E3.10). En este panel seleccionamos el valor lógico deseado (en caso de un bus teclearemos el valor en hexadecimal y pulsaremos el botón ). Tras asignar un valor el indicativo @ aparece junto a la señal.
4. **Edición** de la forma de onda. Podemos modificar una forma de onda indicando en cada instante futuro que valor va a tomar. Para ello seleccionar en el menú del simulador **Waverform** → **Edit**, aparecien-

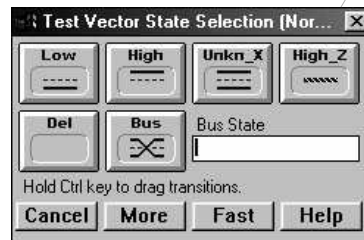


Figura E3.11: Panel de edición de formas de onda.

do un panel de edición de forma de onda (ver figura E3.11). Sobre la señal ó bus que queramos editar: con el ratón, seleccionamos un trozo de cronograma y asignamos el valor deseado en el panel. Mientras se está realizando la edición de una onda se pueden realizar operaciones de copiar/pegar/cortar con el menú contextual. Para indicar que una forma de onda ha sido manipulada manualmente aparecerá el indicativo Cs junto a la señal.

He aquí algunas observaciones adicionales a tener en cuenta a la hora de establecer las entradas de nuestra simulación:

- Podemos inicializar la simulación borrando las formas de onda (para ver el cronograma “limpio”) con la entrada de menú **Waveform → Delete → All Waveform with Power On** (también se pueden borrar sólo señales seleccionadas con la opción **Waveform → Delete**).

¡Atención!: En este caso ¡los estímulos editados a mano se borran!, y sólo los estimuladores permanecen. Por ello, en general, recomendamos asignar valores mediante el método 1 (estimuladores) y dejar las otras posibilidades (2, 3 y 4) para circunstancias en que queramos depurar a mano algún detalle.

- Si deseamos modificar la excitación de una señal a la que ya le hemos asignado valores, se recomienda borrar el estímulo (entrada de menú **Signal → Delete Stimulators**) antes de asignar una excitación nueva.
- Mencionamos que los estímulos pueden actuar de dos modos diferentes denominados *override* (OV) y *chip controlled* (CC) (ver figura E3.8). En principio, usaremos en las simulaciones el modo existente por defecto.

En el modo OV, cuando asignamos un estímulo, éste tiene preferencia y se impone sobre el posible estado de la señal asociada. En este modo

podríamos, por ejemplo, asignar un estímulo a la salida de un chip, y el valor del estímulo se impondría sobre el de la salida. Esto podría servir para chequear una parte concreta del circuito sin desconectarla del resto.

En el modo CC, son los chips los que según su funcionalidad deciden el valor de la señales. En este modo sólo tiene sentido asignar estímulos a entradas del circuito.

E3.6. Guardando y recuperando formas de onda (vectores de test)

Con la entrada del menú **File** → **Save Waveform...** podemos guardar el contenido de la simulación que estamos realizando en un fichero con extensión **.tve**. En una ventana de selección de archivo (figura E3.12) se elige el nombre del fichero destino, que por defecto coincide con el nombre del proyecto.

El fichero **.tve**, denominado fichero de vectores de test⁴, contiene fundamentalmente los valores lógicos de las señales en los instantes de tiempos simulados. Junto con las formas de onda, además, en el fichero se almacena información adicional como son: las fórmulas y estimuladores definidos, la agrupación en buses que hayamos realizado en la visualización e información sobre la jerarquía.

Podemos cargar una simulación almacenada en un fichero de vectores de test mediante la entrada de menú **File** → **Load Waveform...**. Debemos tener en cuenta que no sólo recuperaremos las formas de onda, sino que cargaremos los estimuladores y fórmulas, sobrescribiendo, por tanto, las que estén definidas en ese momento.

Los ficheros **.tve** son binarios. Es interesante la posibilidad de guardar la simulación en formato ASCII. Esto se puede hacer seleccionando el tipo de archivo ASCII a la hora de grabar (ver figura E3.12). Este formato puede ser en ocasiones especialmente útil, ya que los vectores de test almacenados en formato ASCII (con extensión **.asc**) son legibles desde un editor de texto y fácilmente modificables a mano siguiendo la sintaxis adecuada.

No olvidemos que si en las preferencias generales está activo el *backup* automático, cada cierto tiempo el archivo de vectores de test se sobrescribirá automáticamente. Por ello se recomendaba en la sección E3.2 desactivar dicha opción.

⁴Por vector de test se entiende un vector que para un instante de tiempo especifica los valores lógicos de un conjunto de señales

Expresión	Significado
0 1 X Z	Valores lógicos de señales individuales. El símbolo X indica desconocido (<i>unknown</i>) y Z alta impedancia.
10001011\B 139\D 8B\H	Valores lógicos de buses. Los sufijos \B, \D, \H indican la base (binario, decimal y hexadecimal respectivamente).
ps ns us	Unidades de tiempo: picosegundos ($1\text{ps}=10^{-12}\text{s.}$), nanosegundos ($1\text{ns}=10^{-9}\text{s.}$), microsegundos ($1\text{us}=10^{-6}\text{s.}$).
$T\text{ns}=V$	Transcurridos T nanosegundos desde la transición anterior, asignar a la señal ó bus el valor lógico V . Podemos generar una secuencia de transiciones con asignaciones de este tipo separadas por espacios.
$(S)*N$	Repetir una secuencia S un número de veces N .
$(T\text{ns}=Oper\text{ by }m)*N$	Repetir N veces la operación $Oper$ en una cantidad m , con periodicidad T nanosegundos. Entre otras, la operación ($Oper\text{ by }m$) puede ser: (inc by m): incrementar la señal en m , (dec by m): decrementar la señal en m , (sl by m): desplazamiento lógico a la izquierda de m bits, (sr by m): desplazamiento lógico a la derecha de m bits.

Cuadro E3.1: Sintaxis WFM para la edición de fórmulas lógicas.

Ejemplo de fórmula	Cronograma
<p>$0ns=0$ $25ns=1A_{(16)}$ $50ns=32_{(10)}$ $100ns=01001000_{(2)}$</p> <p>Descripción: Se parte inicialmente del valor lógico 0. Transcurridos 25ns la señal toma el valor $1A_{(16)}$, y 50ns después de esta última transición su valor es $32_{(10)}$. 100ns más tarde del último cambio el valor que tomará es $01001000_{(2)}$, permaneciendo así a partir de ese momento.</p>	
<p>$0ns=0$ $100ns=1$ $(50ns=0 \ 50ns=1)*1000$</p> <p>Descripción: Partiendo de valor lógico 0, la señal conmuta a 1 a los 100ns. Después empieza un bucle que se repite 1000 veces en el que transcurridos 50ns conmuta a 0 y 50ns más tarde conmuta a 1.</p>	
<p>$0ns=AA_{(16)}$ $50ns=FF_{(16)}$ $(25ns=0 \ 25ns=FF_{(16)})*2$ $(25ns=11_{(16)} \ 25ns=EF_{(16)})*2$</p> <p>Descripción: Partiendo de $AA_{(16)}$, la señal conmuta a $FF_{(16)}$ a los 50ns. Después repite dos veces alternar entre 0 y $FF_{(16)}$ cada 25ns y posteriormente otras dos veces alterna los valores $11_{(16)}$ y $EF_{(16)}$ cada 25ns quedándose en este último valor.</p>	
<p>$0ns=0$ $(50ns=inc \ by \ 1)*1000$</p> <p>Descripción: Partiendo de 0 el bus se autoincrementa en una unidad cada 50ns mil veces.</p>	
<p>$0ns=FF_{(16)}$ $(25ns=sr \ by \ 1)*8$</p> <p>Descripción: Partiendo de $FF_{(16)}$ se realiza un desplazamiento lógico de un bit a la derecha cada 25ns hasta un total de 8 veces.</p>	

Cuadro E3.2: Ejemplos de fórmulas y sus cronogramas correspondientes.

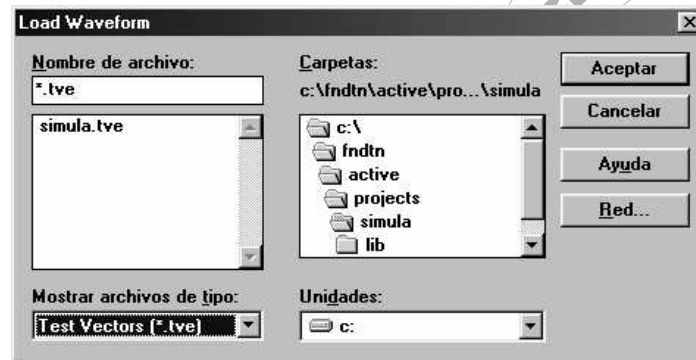






Figura E3.12: Guardando la simulación.

E3.7. Ejemplo

En esta sección terminaremos el ejemplo que se comenzó en el capítulo anterior. Anteriormente creamos el esquemático, usando las ventajas del diseño jerárquico. En este apartado procederemos a la simulación de su comportamiento.

Comenzaremos por recuperar el proyecto que habíamos creado y salvado anteriormente. Una vez tengamos nuestro trabajo cargado en memoria, simulación se puede arrancar desde la ventana de introducción de esquemáticos con  (también desde el menú **Tools**→**Simulator**) o desde el gestor de proyectos con el botón .

Las señales que queramos visualizar deberán añadirse al cronograma con  (menú **Signal**→**Add Signals...**). En nuestro caso seleccionaremos todas menos **SimGlobalReset** y cerraremos la ventana. Podemos hacer más cómoda la lectura de las señales reordenándolas mediante arrastrar y soltar. Incluso si tenemos seleccionada alguna señal podemos incluir una línea en blanco entre ésta y la anterior con **Signal**→**Empty Rows**→**Insert**. En nuestro caso las organizaremos del modo siguiente: **IA**, **IB**, **IC** e **ID**, línea en blanco, **SELO**, **SEL1**, línea en blanco, y **OBUS**. Además, visualizaremos **IA** en binario (**Signal**→**Bus**→**Display Binary**) y para **ID** veremos el bus bit a bit (**Signal**→**Bus**→**Flatten**).

El siguiente paso es definir los estímulos. Usaremos fórmulas, la señal de reloj y modificación de señales mediante el teclado. En primer lugar introduciremos las fórmulas siguientes pulsando sobre  (**Signal**→**Add**

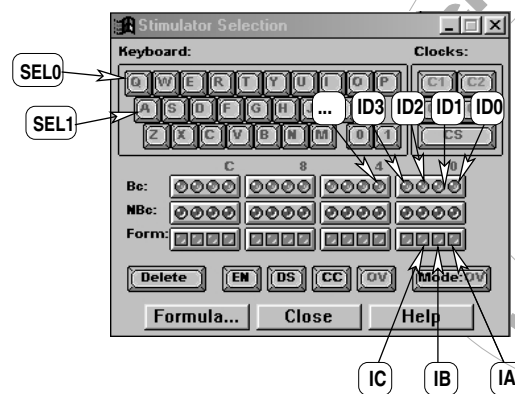


Figura E3.13: Botones usados para asignar estimuladores en el ejemplo.


Stimulators...→Formula...):

F0: 0ns=0 (5ns=01010101\B 5ns=00\H)*100

F1: 0ns=FF\H (15ns=dec by 1)*100


F2: 0ns=FF\H (25ns=sr by 1)*100

Una vez introducidas cerramos (**Close**) la ventana de edición de fórmulas y realizamos la asignación de estímulos a señales. Para ello debemos seleccionar en primer lugar la señal (queda sobresaltada en azul) y posteriormente pulsamos sobre el botón correspondiente al estímulo que le queremos asignar. Si todo ha ido bien en la columna situada a la derecha del nombre de la señal en el cronograma aparecerá en rojo un texto indicativo del estímulo seleccionado. A los buses **IA**, **IB** e **IC** haremos corresponder las fórmulas **F0**, **F1** y **F2** respectivamente (línea de botones **Form** en el panel **Stimulator Selection**). A los bits del bus **ID** les haremos corresponder los bits del contador (primeros 8 botones por la derecha de la línea **Bc** del panel). Las señales **SEL0** y **SEL1** las modificaremos de forma manual con las teclas **q** y **a** respectivamente (dibujadas dentro del recuadro **keyboard** sobre el panel). Todo esto aparece resumido en la figura E3.13.

Solo nos queda por configurar los parámetros de las escalas de tiempo. Elegiremos un paso de 10ns y una escala de 200ps/div. A partir de este momento, cada vez que pulsemos el botón  o la tecla **F8** la simulación avanzará 10ns. Pulsando las teclas **q** y **a** (estímulos asignados a las señales **SEL0** y **SEL1** respec-

tivamente) podemos observar el efecto sobre la salida de modificar las señales de selección del decodificador.

En cualquier momento podemos conocer el valor de una señal en un instante de tiempo ya simulado solo pinchar con el ratón sobre el cronograma en el momento y señal deseado. El valor lógico en el instante seleccionado se muestra en el indicador del estado lógico actual.

Si en algún momento deseamos volver a empezar, podemos pulsar el botón  (**Device → Power On/Reset**). Si además queremos que se borre la simulación anterior (no nos interesa compararlas) podemos hacer **Waveform → Delete → All Waveforms with Power On**.

En la figura E3.14 se puede ver el estado de la simulación del ejemplo en un momento dado de la misma.

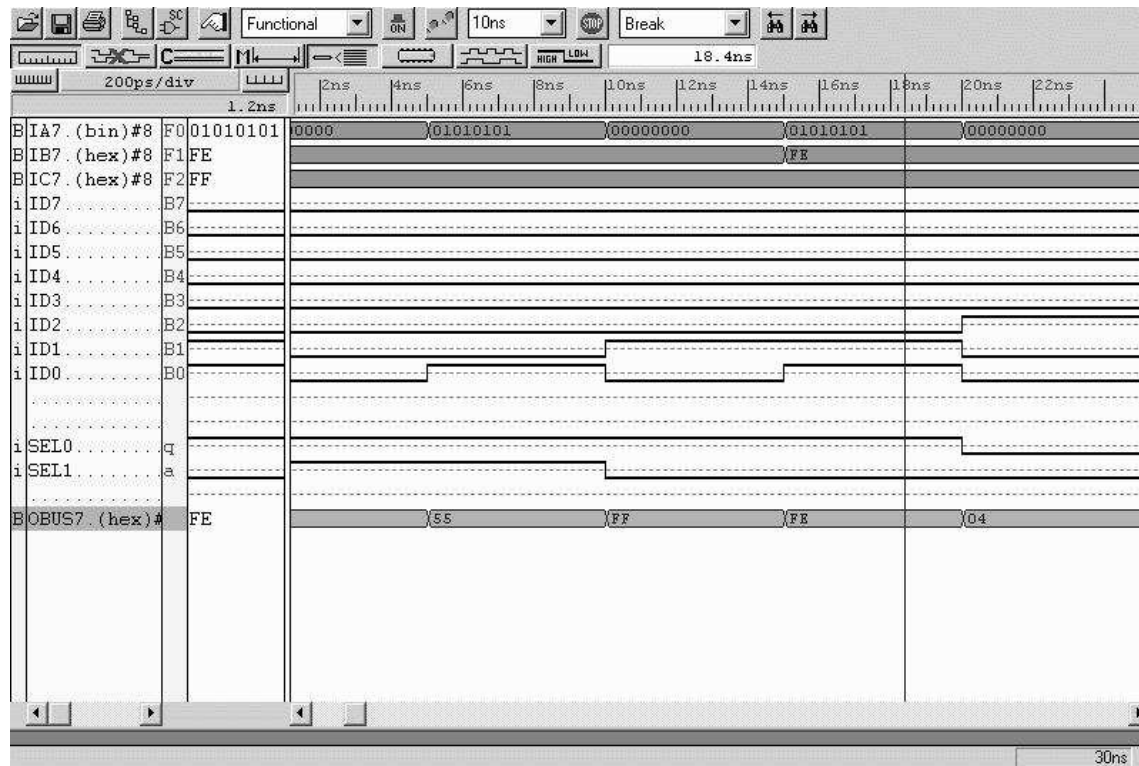


Figura E3.14: Visión general del simulador y del visualizador de formas de onda en la simulación del ejemplo.

Bibliografía



- [1] G. Bandera, F. Corbera, M.A. González, M. González, E. Gutiérrez, A. Rodríguez, M. Sánchez, M.A. Trenas: *Tecnología de Computadores*, Universidad de Málaga/Manuales, 2000.
- [2] F. Corbera, M.A. González, M. González, E. Gutiérrez, A. Rodríguez, S. Romero, M.A. Trenas: *Ejercicios de Tecnología de Computadores*, Universidad de Málaga/Manuales, 2002.
- [3] David A. Patterson, John L. Hennessy: *Organización y Diseño de Computadores*, Ed. Reverté. 2000.
- [4] *Xilinx University Program* en <http://www.xilinx.com/univ>, con información relativa a la herramienta *Xilinx Design Series*.
- [5] P. de Miguel Anasagasti: *Fundamentos de los Computadores*, Ed. Thomson Paraninfo, 9ª edición, 2004.