

TECNOLOGÍA DE COMPUTADORES. LABORATORIO

Curso 2007-2008

¿Por dónde debo comenzar a trabajar?

En primer lugar debes obtener el enunciado de la práctica disponible en la web de la asignatura dentro del Campus Virtual de la UMA. Una vez explicado el enunciado por el profesor en el laboratorio los primeros pasos a realizar serán con papel y lápiz, y consistirán en el diseño de nuestro procesador, comenzando por el formato de instrucción.

Al final, este diseño en papel habrá que montarlo y probarlo (simulándolo) usando la herramienta *Xilinx Foundation Series*. Dispones de un manual, el texto *Prácticas de Tecnología de Computadores*¹, dónde puedes encontrar una descripción de la herramienta a utilizar (*Xilinx*). El profesor introducirá en el laboratorio las nociones básicas del manejo de la herramienta para poder simular nuestros circuitos y diseños.

Además el profesor de tu grupo dará las indicaciones pertinentes en el laboratorio para de registrarte en el sistema de entrega de prácticas (GUAC). Con el registro obtendrás un identificador de equipo, que es indispensable para la realización y entrega del trabajo.

Instrucciones y FAQ para XILINX

¿Cómo consigo el proyecto inicial (prototipo) para empezar a montar mi procesador en Xilinx?

El proyecto prototipo para el curso 2007/08 está disponible en la página *web* de la asignatura en el Campus Virtual. En la sección de prácticas encontraras el fichero TG00PROC.ZIP. Ése es el que necesitas.

No sé como usar Xilinx para dibujar o simular mi procesador ¿qué puedo hacer?

Dispones de un manual, el texto *Prácticas de Tecnología de Computadores*, dónde puedes encontrar una descripción de la herramienta a utilizar (*Xilinx*). Lee este manual como

¹ Manual nº64 del Servicio de Publicaciones de la Universidad de Málaga (SPICUM)

introducción a *Xilinx* y úsalo como referencia en tu trabajo con la herramienta. Búscalo en la biblioteca o en alguna librería (QPROQUO).

¿Cómo abro un proyecto que está en un archivo ZIP?

¿Cómo me llevo un proyecto después de haber trabajado en él?

¿Cómo genero los archivos ZIP que hay que dejar en el servidor?

Puesto que cada proyecto Xilinx se compone de un buen número de ficheros y directorios, la propia herramienta de desarrollo permite crear y recuperar de forma compacta un proyecto en forma de archivo comprimido .ZIP.

Estos archivos .ZIP, para descargar y dejar en el servidor, deben ser generados ó recuperados dentro de la propia herramienta Xilinx Foundation, con la opción de menú:

File →Archive_Project / File →Restore_Project.

NO debe utilizarse en ningún caso herramientas como WinZIP para generar ó recuperar estos archivos. Ver en sección E1.7 del manual *Prácticas de Tecnología de Computadores*.

¿Cómo realizo una copia de un proyecto con otro nombre?

Toda copia de un proyecto debe realizarse exclusivamente con **File→Copy_Project**. Ver sección E1.5. Por ejemplo, para pasar del proyecto prototipo TG00PROC al proyecto propio del equipo 74 de sistemas D, correctamente nombrado, es necesario copiarlo con el nuevo nombre SD74PROC). Esta operación **deben realizarla todos los equipos de trabajo antes de empezar a trabajar** con su proyecto.

No utilices en ningún caso una copia directa con el explorador de archivos o en línea de comandos ya que no funcionará. El proceso de copia de un proyecto implica más acciones además de las de meramente copiar los ficheros y hay que hacerlo como se indica (**File→Copy_Project** dentro del *Project Manager*).

¿Qué diferencia hay entre Archivar/Recuperar un proyecto y Abrir?

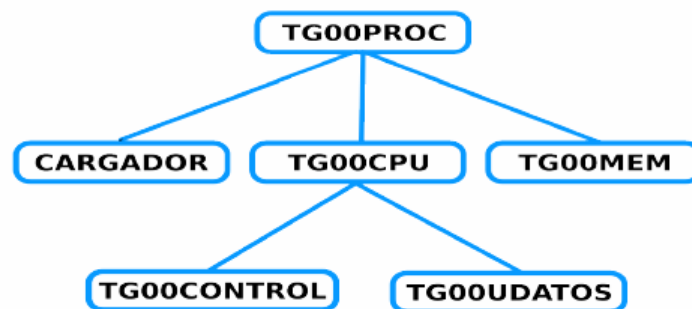
No confundas las opciones **File→Archive_Project**, **Restore_Project**, para almacenar y restaurar respectivamente el proyecto en un fichero comprimido .ZIP (para subirlo al

servidor o llevarlo de un sitio a otro) con **File Open** que abre un proyecto ya descomprimido en su directorio correspondiente.

¿Qué diferencias existen entre hojas de esquemáticos y macros?

Debes tener muy claro los conceptos del diseño jerárquico así como la diferencia entre hojas de esquemáticos y macros de librería (capítulo E.1).

Recuerda que nuestro proyecto consta de UNA sola hoja (TG00PROC.SCH) y que las prácticas consisten en la realización de las diferentes macros necesarias para completar el procesador (CPU, unidad de control, unidad de datos, memoria) que finalmente quedarán embutidas en la librería del proyecto. La jerarquía inicial de macros es la siguiente:



¿Cómo genero una macro a partir de un esquemático?

Los esquemáticos creados de acuerdo con las especificaciones deben ser agregados a la librería del proyecto, utilizando para ello la generación de la macro correspondiente (opción **Hierarchy→Create Macro from Current Sheet**, ver sección de diseño jerárquico E.2.2).

No olvides que el nombre de las macros seguirá el convenio fijado, según se describe en el enunciado de las prácticas (ver fin de epígrafe “3.- Diseño del ciclo de instrucción y de la unidad de datos”).

¿Cómo sustituyo en el prototipo una instancia de librería (caja vacía) por una macro que ya he diseñado?

Cada macro creada deberá sustituir a la TG00.... correspondiente del proyecto prototipo. Para ello debe utilizarse la opción **Replace Symbol** (Ver sección E2.2.1).

¿Cómo se simula una macro individualmente?

Las macros individuales se simularán y probarán por separado OBLIGATORIAMENTE, utilizando para ello la opción **Tools→Simulate_Current_Macro** (simular macro actual), del editor de esquemáticos. Ver nota al pie al comienzo de la sección E3.1. No obstante, una vez completadas todas las macros del proyecto, se podrá simular el proyecto, es decir el procesador, en su totalidad.

INSTRUCCIONES ADICIONALES

Las siguientes instrucciones son de obligado cumplimiento. Se darán por no válidas aquellas prácticas que no se ajusten a estas normas.

De cara a agilizar la verificación y corrección de las prácticas, se va a exigir que se proporcione una versión de las mismas en el formato VHDL, quedando incluidos los ficheros VHDL en el proyecto Xilinx a entregar. La generación de los ficheros en este formato se realiza desde la propia herramienta Xilinx Foundation Series, y el alumno simplemente tendrá que exportar su práctica a dicho formato. A continuación se ofrecen unas sencillas reglas de cómo hacer esto:

¿Qué es VHDL?

Del acrónimo *VLSI² Hardware Description Language*, VHDL es un lenguaje de descripción de hardware orientado a la descripción en alto nivel de sistemas digitales.

¿Cuándo he de exportar mi práctica a VHDL?

Antes de realizar la entrega de la práctica en el servidor, una vez finalizada la práctica en cuestión y comprobado que funciona correctamente.

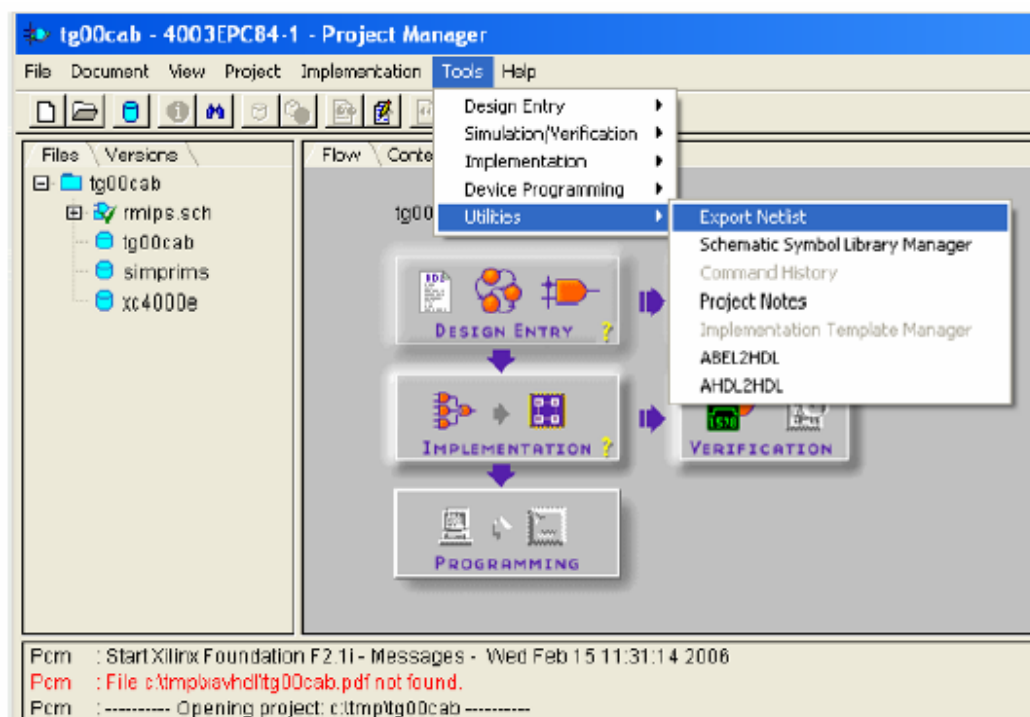
Los ficheros VHDL generados, quedarán en el directorio del proyecto y por tanto irán incluidos en el archivo ZIP obtenido mediante la opción **File → Archive Project**.

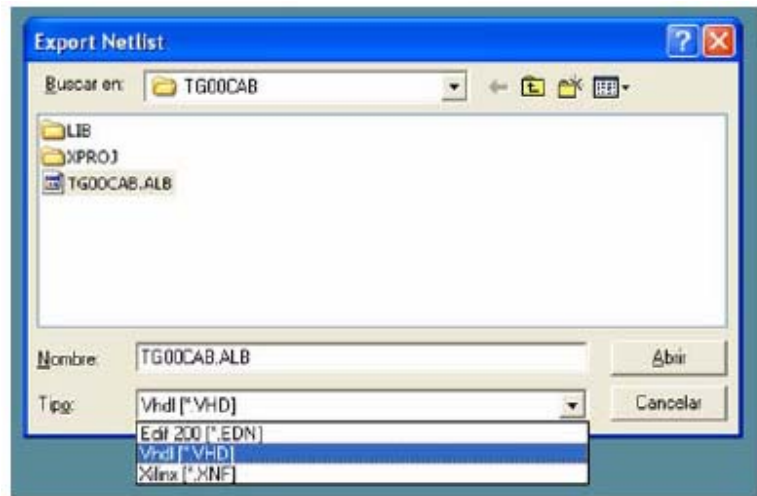
² *VLSI = Very Large Scale Integration*

¿Cómo exporto una práctica finalizada a VHDL?

Una vez finalizada la práctica, y verificado su funcionamiento, podemos exportar a VHDL directamente desde el *Project Manager*. Seleccionaremos la opción del menú **Tools** → **Utilities** → **Export Netlist**.

Tras esta acción en un panel de selección de ficheros, elegiremos el *netlist* a exportar (nombre de la macro con extensión .ALB), y seleccionaremos como tipo VHDL. Las figuras que siguen ilustran estas acciones.





¿Debo tener especial cuidado con las macros LogiBLOX?

Sí. Las macros LogiBLOX son una excepción en el proceso de exportación ya que su descripción VHDL no se crea durante la exportación, sino que se genera automáticamente durante la creación de la propia macro.

En realidad, durante la creación de una macro LogiBLOX se generan diversos ficheros auxiliares, entre los cuales está la descripción VHDL. Debemos procurar no borrar ninguno de ellos.

En caso de que por error borráramos el VHDL de una macro LogiBLOX, no habrá otro remedio que editar dicha macro y regenerarla de nuevo (desde el editor de esquemáticos con **Tools** → **LogiBLOX module generator**).

¿Debo tener en cuenta alguna consideración en la elaboración de los esquemáticos?

Para que los VHDL generados sean de "buena calidad", es necesario evitar interconexiones en los esquemáticos que puedan dar lugar a ambigüedades o malinterpretaciones sintácticas durante el procesamiento del código VHDL. A continuación se comenta alguna regla a seguir para evitar estas situaciones. El no seguir estas reglas puede dar lugar a errores de formato cuando la práctica entregada sea evaluada.

Estas son las reglas:

- NO definir nunca un bus con el mismo nombre de una señal escalar.



- Un conector jerárquico (salidas/entradas de una macro) NO debe ser un subrango de un bus ya definido. Utiliza *buffers*, si quieres conectar una porción de un bus a un conector jerárquico de entrada o salida.

