Práctica 2

Diseño de una Arquitectura de Cómputo basada en el microcontrolador PicoBlaze

1. Introducción y objetivos.

El objetivo de esta práctica es el de diseñar un sistema de cómputo basado en el core de microcontrolador proporcionado gratuitamente por Xilinx denominado PicoBlaze. El sistema deberá disponer de comunicación con el usuario para la entrada/salida de datos, para lo cual puede utilizarse una interfaz RS-232 con un PC convencional.

La arquitectura del sistema de cómputo debe incluir:

- **Al menos una** instanciación del microcontrolador PicoBlaze y de su correspondiente**/s** Memoria**/s** de Programa.
- **Programa (**o **programas** cuando hay varios PicoBlaze**) de Aplicación** para el PicoBlaze, que realice una tarea específica y diferente elegida por cada grupo de prácticas. Algunos posibles ejemplos son: sistemas de codificación y decodificación (encriptación, etc.), juegos sencillos (guerra de barcos, "el ahorcado", etc.), calculadoras, etc.
- El sistema deberá incluir una **interfaz RS-232** para la comunicación con el usuario a través de un PC convencional, la cual puede realizarse "por programa" (como en el ejemplo de test del PicoBlaze "Hello world" que se realiza en prácticas) sin necesidad de utilizar una UART como periférico real.
- **Al menos un periférico de diseño propio conectado al PicoBlaze que realice un procesamiento específico**. Algunos ejemplos son: bancos de memoria adicionales (debido a su sencillez su utilización deberá ser justificada y consultada al profesor), una unidad de multiplicación rápida (por hardware), un banco de filtros, un periférico específico para realizar la comunicación RS-232 (por ejemplo la UART simplificada que se proporciona en el libro de prácticas de la asignatura), timers, contadores, manipuladores de bits (comparadores, generadores de paridad, codificadores/decodificadores…), etc.
- **Al menos una modificación de la microarquitectura** del PicoBlaze (introducción de una nueva instrucción, modificación del comportamiento de alguna instrucción, aumento de la memoria de programa, aumento del número de niveles de la pila, etc.).

2. Componentes y documentación.

A modo de guía para el alumno de cara al desarrollo del proyecto, a continuación se especifican de forma ordenada los elementos más importantes que el alumno debería estudiar, junto con la documentación (básica y avanzada) que se debe consultar en cada caso:

1. Conocer la arquitectura interna y el juego de instrucciones del microcontrolador PicoBlaze.
   - Documentación Básica: xapp387_PicoBlaze_userguide.pdf (Manual del Programador del PicoBlaze para CPLDs), xapp38_PicoBlaze_distrib.zip (Distribución con todos los archivos que componen el PicoBlaze para CPLDs).

2. Conocer las principales características y los recursos de que disponen las tarjetas de prototipado con FPGAs que se utilizarán en el laboratorio de prácticas.
   - Documentación Básica: Intro_PicoBlaze_digilent_S3Estarter.pdf (Introducción a las tarjetas S3E starter kit de Digilent Inc.).
   - Documentación Avanzada: S3EStarter_ug230.pdf (Manuales del Usuario de las tarjetas S3E starter kit de Digilent).

3. Opcional: Conocer el PicoIDE, el entorno de desarrollo para PicoBlaze (desarrollado en la UPCT) para facilitar la generación de aplicaciones con este microcontrolador. PicoIDE integra las funciones de editor de textos, ensamblador y depurador.
   - Documentación Básica: Distribución de PicoIDE PicoIDEv1.1_distrib.zip (además del programa en sí incluye ejemplos, manual del usuario, etc.).
   - Documentación Avanzada: JCRA-04_picoblaze_debugger.pdf (artículo sobre PicoBlaze y PicoIDE presentado en las Jornadas de Computación Reconfigurable y Aplicaciones del 2004.).

4. Opcional: Conocer la estructura y funcionamiento de la UART simplificada.
   - Documentación Básica: intro_pserie.ppt (Introducción a la UART simplificada),
   - Documentación Avanzada: Síntesis de Sistemas Digitales con VHDL (Proyecto 2: Diseño de una UART).

5. Opcional: Conocer la arquitectura de las FPGAs tipo Spartan3E de Xilinx.
   - Documentación Básica: ug331.pdf (Hoja de características de la Spartan 3E, información del fabricante).

3. Fases de desarrollo.

A continuación se detalla el flujo de trabajo típico cuando se aborda el desarrollo un sistema de cómputo basado en PicoBlaze sobre una FPGA:

1. Crear un **nuevo proyecto** con la herramienta ISE de Xilinx.
2. **Incluir en este proyecto todos los archivos .vhd correspondientes al PicoBlaze** (comenzando por el picoblaze.vhd que es el principal. Cuando se añaden los sucesivos archivos, verificar que los diferentes módulos o *entities* (contador de programa, banco de registros, etc.) se van colocando automáticamente en su lugar dentro de la jerarquía de componentes del PicoBlaze. También se pueden añadir todos los archivos de una vez.
3. **Definir una nueva *entity*, que será nuestra unidad principal de diseño** (la de mayor nivel en la jerarquía de componentes), y donde instanciaremos como componentes el PicoBlaze y, más adelante, su memoria de programa y cualquier otro periférico o hardware adicional que necesitemos incluir en el sistema.
4. **Si es necesario, alterar la estructura interna o el comportamiento del PicoBlaze en la forma deseada** modificando según convenga los archivos .vhd donde se definen los módulos que lo componen. Igualmente, y en caso de ser necesario, definir el código correspondiente a los periféricos o módulos adicionales que deseemos crear y añadirlos a la jerarquía de componentes del proyecto.
5. Utilizar el entorno PicoIDE para **definir el programa que ejecutará el PicoBlaze, ensamblarlo y depurarlo** (ejecutándolo paso a paso si es necesario) hasta verificar que funcione correctamente. En lugar PicoIDE, también puede hacerse utilizando un editor de texto plano cualquiera (como el NotePad) y ejecutando el programa ensamblador (asm.exe) desde la línea de comandos. Tener en cuenta que, si en el paso anterior se elige modificar el juego de instrucciones del PicoBlaze, habrá que modificar también el programa ensamblador (asm.exe) y, opcionalmente, el depurador (PicoIDE.java) para que puedan ensamblar y depurar respectivamente las nuevas instrucciones.
6. **Incluir en el Proyecto del ISE la nueva entidad conteniendo la memoria de programa** generada por el programa ensamblador en el paso anterior.
7. **Añadir al proyecto del ISE un archivo .ucf con las restricciones (condiciones) de usuario para implementación el circuito**, que son fundamentalmente: la colocación de los puertos de E/S del circuito en los pines adecuados para el caso de la FPGA y la tarjeta que utilizamos en prácticas, así como la definición de la velocidad del reloj a la que deberá funcionar el sistema (que en nuestro caso es de 50MHz).
8. **Sintetizar el proyecto** del ISE para comprobar que no posee errores y que el circuito completo cabe dentro de la FPGA seleccionada.
9. **Verificar que no se producen errores** y que se tuvieron en cuenta las restricciones (de pines y velocidades) definidas en el paso anterior. Para ello consultar los diferentes informes (*reports*) que genera la herramienta.
10. **Definir un banco de pruebas (*testbench*) y simular** nuestro sistema completo, para verificar que su funcionamiento es correcto. Si estamos seguros de que la comunicación RS-232 funciona (las rutinas de comunicación RS-232 proporcionadas por el profesor ya están verificadas), es interesante **simular únicamente la parte del programa correspondiente al algoritmo de procesamiento implementado, eliminando del mismo la parte de entrada/salida de datos por el puerto serie** (se pueden utilizar datos constantes para las pruebas),

con el fin de simplificar y sobre todo acortar enormemente el tiempo de simulación (la transmisión serie consume miles de ciclos de reloj!!!).

11. **Modificar las opciones de ISE para la generación del archivo de programación** de la FPGA según corresponda para adecuarlas a la tarjeta de prototipado y al método de programación disponible en el laboratorio. **Generar el archivo .bit de programación de la FPGA** con estas opciones.
12. **Conectar la tarjeta de FPGA al PC utilizando el cable de programación**. **Alimentar la tarjeta y utilizar el programa iMPACT del entorno ISE para programar la FPGA** con nuestro sistema volcando el archivo .bit generado en el apartado anterior.
13. **Realizar todas las conexiones adicionales necesarias para verificar el funcionamiento** de nuestra aplicación: conexión RS-232 con PC, o cualquier otra que se haya utilizado.