

LABO-MICRO: ENTORNO DE TEST PARA LA VERIFICACION DE MICROPROCESADORES EXPERIMENTALES SOBRE CIRCUITOS FPGA

I. GONZÁLEZ, C. J. VENEGAS, S. LÓPEZ-BUEDO, F. J. GÓMEZ,
J. MARTÍNEZ Y J. GARRIDO

Departamento de Ingeniería Informática. E.T.S. de Informática.

Universidad Autónoma de Madrid. Cantoblanco, E-28049 Madrid, España.

En este documento se presenta un entorno de verificación que permite comprobar el funcionamiento de un microprocesador diseñado en VHDL para ser implementado en un circuito reconfigurable del tipo FPGA. Se ha desarrollado un conjunto de herramientas que permiten a) volcar el diseño en la FPGA, b) utilizar un microcontrolador externo para cargar el programa de prueba y c) observar los resultados de su ejecución a través de una memoria de doble puerto situada entre la FPGA y el microcontrolador.

1. Introducción

La utilización de lenguajes de descripción de hardware hace viable la realización de un procesador elemental como práctica para los alumnos de la asignatura de tercer curso “Arquitectura e Ingeniería de Computadores”. El trabajo básico consiste en desarrollarlo en VHDL y simular su funcionamiento. Un reto que conlleva un esfuerzo adicional consiste en la síntesis del diseño para ser implementado en hardware, utilizando por ejemplo, FPGAs. Aunque la verificación del funcionamiento de la implementación física del procesador es posible con herramientas de simulación post-layout resulta más gratificante y atractivo verificar el correcto funcionamiento del diseño utilizando una plataforma de test, que permita comprobar al alumno su funcionamiento de una manera interactiva. En este artículo se emplea una plataforma reconfigurable que incluye un microcontrolador, circuitos FPGA, y memoria para realizar el test del prototipo de procesador. El procesador experimental se implementa en los recursos reconfigurables (FPGA) y la memoria de instrucciones y datos conectada al mismo es directamente la *Dual-Port SRAM* presente en la plataforma. Las herramientas de test se basan en la utilización del microcontrolador MC68360 para comprobar el funcionamiento del diseño implementado en los recursos reconfigurables y se aprovecha que se dispone sobre la plataforma de un sistema operativo con soporte de comunicaciones TCP/IP para permitir que las herramientas puedan funcionar como aplicaciones Web.

2. Descripción del microprocesador: Arquitectura y repertorio de instrucciones

El objetivo de las prácticas es que el alumno desarrolle en VHDL un sencillo microprocesador RISC, con las conexiones al exterior esquematizadas en la figura 1 y con las siguientes características:

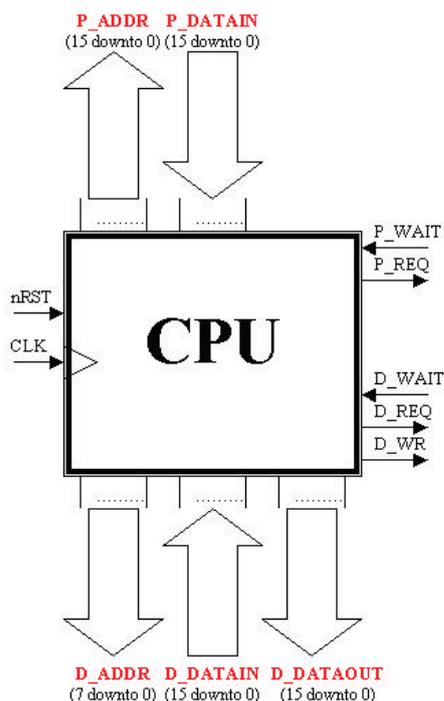


Figura 1: Arquitectura del microprocesador

- Arquitectura de 16 bits, tipo LOAD-STORE
- Arquitectura Harvard, con memorias separadas para datos y programa.
- Cuatro registros internos de propósito general.
- Un bit de flag para saltos condicionales.
- 8 instrucciones básicas (LOAD, STORE, MOVE, ADD, SUB, comparación, salto condicional e incondicional).

Inicialmente se pide sólo simular el correcto funcionamiento del microprocesador a nivel de descripción VHDL comportamental. Sin embargo, se alienta a todos los alumnos a que realicen una implementación sobre una FPGA que ejecute alguna sencilla aplicación. Esto presenta una mayor dificultad, ya que es necesario realizar una correcta codificación RTL para poder pasar la síntesis sin problemas. Pero lo que puede resultar menos gratificante para el alumno es realizar el test de su sistema, pues la única herramienta de la que se dispone en principio es de la simulación post-layout, que es un método lento y poco adecuado para la depuración del software.

3. Plataforma de verificación: herramientas de test

El tipo de plataforma que se requiere para este tipo de test debe incluir una FPGA donde poder implementar el microprocesador diseñado, un microcontrolador que pueda comunicarse con la FPGA y que se encarga del control de flujo de datos, y al menos una memoria que pueda ser accedida tanto por el microcontrolador como por la FPGA, la cual va a contener el programa a ejecutar y los datos del mismo. En la actualidad, existen muchas plataformas que cumplen estas características: Labomat 3 [1], RC1000 PP de Celoxica [2], XESS [3], etc.

Nosotros hemos empleado la tarjeta Labomat3, desarrollada en el Laboratorio de Sistemas Lógicos (LSL) del École Polytechnique Fédérale (EPFL) de Lausanne en Suiza. Consta básicamente de un microprocesador de Motorola MC68360 [4], dos FPGAs de Xilinx, una XC4013E y una XC6216, y una serie de memorias adicionales que son accesibles por las FPGAs y el micro. De las FPGAs se ha escogido la XC4013E por ser la de mayor tamaño. Como memoria se emplea una DPSRAM de 4kB que está estrechamente vinculada a la FPGA seleccionada y es accesible desde el MC68360. Otra de las características importantes de esta plataforma son sus posibilidades de comunicación, ya que dispone de conexión Ethernet con la posibilidad de utilización en remoto a través de Internet.

La plataforma es independiente ya que dispone de su propio sistema operativo RTEMS y se ha desarrollado un conjunto de aplicaciones para utilizar todos sus recursos a través de Web[5]. En el caso que nos ocupa las acciones que se necesitan son:

- Cargar la configuración en la FPGA XC4013E con el diseño del microprocesador experimental bajo test

- Ejecutar una función de usuario codificada en lenguaje C para MC68360 y que debe realizar las tareas de monitorización y test. Esta función se encarga de escribir y leer en la zona de memoria usada por el microprocesador así como de la inicialización del sistema y el envío de pulsos de reloj a la FPGA para conseguir la ejecución del programa de prueba escrito en memoria.

4. Adaptación a la plataforma de test.

El microprocesador propuesto en las prácticas presenta una arquitectura Harvard, con memorias de programa y de datos separadas. Para adaptar los recursos de Labomat a esta arquitectura, lo que se hace es dividir la memoria de doble puerto en dos mitades: los primeros 2 KB se emplean como memoria de código, y la segunda mitad como memoria de datos. Tener una memoria de doble puerto simplifica en gran medida el test, pues se puede utilizar el MC68360 para cargar el programa que ejecutará el microprocesador, o para ver el estado de la memoria de datos, todo ello sin tener que interrumpir el microprocesador o añadir lógica adicional en la FPGA. Lo único que es necesario es tener un controlador de memoria que traduzca desde la sencilla interfaz propuesta (Fig. 1) hacia la memoria de doble puerto. Adicionalmente, el MC68360 debe saber la posición del contador de programa para poder visualizar el punto actual de ejecución del programa en el modo paso a paso. Esta información se obtiene directamente del bus de direcciones de la memoria de programa, y se hace visible al MC68360 a través de un puerto, implementado en el mismo controlador antes comentado. Este mismo método se podría usar en futuras ampliaciones del sistema para poder visualizar por ejemplo el estado de los registros internos.

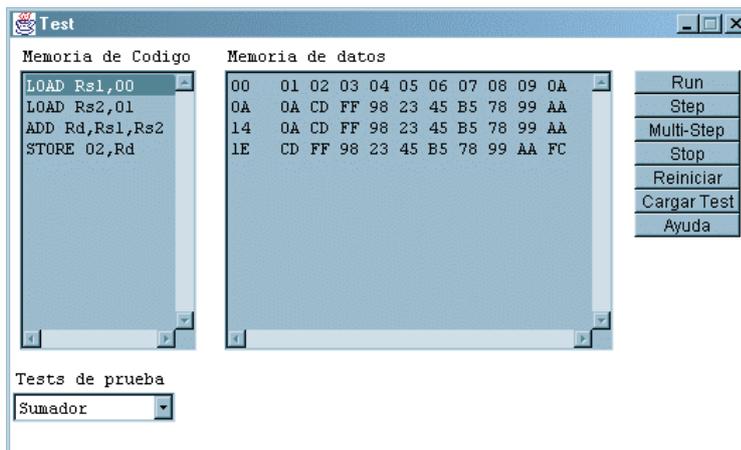
Actualmente se proporciona a los alumnos ya resuelto el módulo de interfaz entre FPGA, DPSRAM, y MC68360, y se detalla la zona de memoria a utilizar como datos y como instrucciones así como el modo de acceso a los registros para el control de la verificación, que básicamente controlan la señal de reset y el envío de pulsos de reloj. La tarea del alumno es realizar un programa en C que permita realizar un test de funcionamiento de su diseño. Los pasos que debe seguir son escribir en la memoria de instrucciones la codificación de un programa. Inicializar el módulo de interfaz para comenzar la ejecución y enviar pulsos de reloj. Los valores que puede verificar son los contenidos del bus de direcciones y de datos así como los contenidos de las posiciones de memoria.

La experiencia actual es que la escritura de este código en C resulta demasiado complicada para el alumno y por esta razón se está desarrollando nuevas herramientas de test y depuración que oculten toda esta parte. En el punto siguiente se describen las aplicaciones en curso de desarrollo.

5. Desarrollo de aplicaciones Web y herramientas de depuración:

Preferiblemente la aplicación debería presentar un interfaz muy similar a la que se puede encontrar en cualquier herramienta de depuración empleada en la programación de microprocesadores. Se está desarrollando un entorno de ventanas, donde se visualice el código que se está ejecutando y se muestre el contenido de la memoria de datos, muy adecuado para la descripción de una arquitectura Harvard. Además, se permite interactuar con el microprocesador mediante de una serie de botones que se emplean para cargar un nuevo

programa, permitir la ejecución del código paso a paso o de forma continua, así como la posibilidad de reiniciar el microprocesador.



El objetivo es que desde este entorno el alumno puede probar distintos programas sobre su microprocesador sin necesidad de conocer el funcionamiento interno del programa de test. Solamente se necesita cargar un programa de prueba y comprobar que funciona correctamente, observando el resultado en la ventana de la memoria de datos, que muestra en todo momento el contenido actualizado de la misma.

Figura 2: Entorno de ventanas

6. Conclusiones

Se ha presentado una aplicación cuya finalidad docente es facilitar al alumno la tarea de comprobar el correcto funcionamiento del diseño de un microprocesador experimental implementado sobre un circuito FPGA. El mayor aliciente de este tipo de herramienta es la posibilidad de realizar una verificación real del funcionamiento mismo. La funcionalidad básica permite cargar y ejecutar programas en la memoria conectada al prototipo de microprocesador, y permite comprobar el correcto funcionamiento de los valores generados en la memoria de datos. La base de la herramienta de tests es un plataforma reconfigurable para la que se han desarrollado una serie de aplicaciones accesibles vía Web que configuran los recursos reconfigurables, y permiten la utilización de las memorias y del microcontrolador presente en la plataforma.

Referencias

- [1] C.Teuscher, J.O. Haenni, F.J. Gomez, H.F. Restrepo, E. Sanchez."Labomat 3: A Re-configurable Platform for Academic Purposes". Proc. FCCM '99. Napa USA. April 1999.
- [2] <http://www.celoxica.com>
- [3] <http://www.xess.com>
- [4] Motorola MC68360. "Quad Integrated Communication Controller". User's Manual. 1993.
- [5] I.Gonzalez, F.J.Gomez, J. Martinez, LabomatWeb: Recursos reconfigurables Remotos via Word Wide Web , Jornadas sobre computación Reconfigurable y aplicaciones pp 102-109. Alicante 2001