

LABOMAT3: UN ENTORNO COMPLETO PARA EL APRENDIZAJE DE TÉCNICAS DE CODISEÑO UTILIZANDO UNA PLATAFORMA RECONFIGURABLE.

F.J. GOMEZ¹, G. GALEANO², H.F. RESTREPO³, J.O. HAENNI³,
C. TEUSCHER³ E. SANCHEZ³

¹Escuela Técnica Superior de Informática, Universidad Autónoma de Madrid,
E-28049 Madrid, España. E-mail: francisco.gomez@ii.uam.es

²Departamento de Informática, Escuela Politécnica, Universidad de
Extremadura,

E-10071 Cáceres, España E-mail: ggaleano@unex.es

³Logic Systems Laboratory, Swiss Federal Institute of Technology CH-1015
Lausanne, Switzerland. E-mail: <name>.<surname>@di.epfl.ch

En este trabajo se presenta una plataforma reconfigurable, Labomat3, para su uso con fines educativos. Su campo de aplicación abarca dominios tan importantes en el curriculum universitario como son: el diseño de circuitos digitales, la arquitectura de ordenadores y el codiseño. Como aplicación práctica se pide a los alumnos el desarrollo de un coprocesador que realice operaciones de calculo en coma flotante y evalúen en su diseño que tareas deben realizarse en hardware de propósito específico.

1. Introducción

El curriculum universitario en ingeniería electrónica abarca dominios tan importantes como son: el diseño de circuitos digitales, la arquitectura de ordenadores y el codiseño. Para facilitar la enseñanza de estos campos, sería conveniente disponer de una herramienta, fácil de utilizar, que permita afrontar un problema desde distintas aproximaciones, y de este modo evaluar la solución mejor adaptada para cada caso. En particular, si pensamos en técnicas de codiseño *hardware-software*, se puede optimizar el rendimiento de una tarea, dependiendo de que partes sean ejecutadas por el *software* y cuales por el *hardware*.

Las diferentes posibilidades que permite esta plataforma, para solucionar un problema son:

- 1.- Programar un algoritmo mediante un lenguaje de alto nivel (lenguaje C) para ser ejecutado en un microprocesador de propósito general.
- 2.- Implementar completamente una solución *hardware* para el algoritmo mediante la programación de circuitos reconfigurables FPGA (*Field-Programmable Gate Array*).
- 3.- La división del problema de forma que una parte se ejecute en el microprocesador y otra parte se realice por el *hardware* reconfigurable. En este caso es fundamental establecer los mecanismos adecuados para la cooperación de ambas partes.

En este artículo, se van a presentar un conjunto de programas de aplicación que permiten afrontar el desarrollo de un proyecto empleando técnicas de codiseño en un sistema reconfigurable.

2. Descripción de la plataforma Labomat3

Labomat3 ha sido desarrollada en el Laboratorio de Sistemas Lógicos (LSL) del *Ecole Polytechnique Fédérale de Lausanne* (EPFL)¹. En la figura 1 se muestra la arquitectura de la tarjeta y su comunicación con los diferentes programas que componen la plataforma completa.

La tarjeta consta básicamente de un microprocesador MC68360 conectado a dos circuitos FPGAs de tamaño medio² (10Kpuertas). Su comunicación con el exterior está garantizada mediante una conexión Ethernet 10Base-T y una conexión serie RS232. Un sistema operativo tiempo real, de dominio público, proporciona la base sobre el que se puede ejecutar directamente programas de usuario compilados para este microprocesador. También dispone del soporte TCP-IP que permita la comunicación remota basada en *sockets*.

Para crear un entorno completo se han desarrollado el siguiente conjunto de programas de aplicación para su funcionamiento en PC con sistema operativo Windows:

- Un programa iniciador **BootP**. Se trata de la implementación y adaptación a PC del servidor *Bootp*. Este programa está a la escucha en la red esperando que cualquier tarjeta que se conecte envíe un mensaje de inicio de conexión. Establecida la conexión BootP envía la imagen del sistema operativo completo para que sea cargada en la memoria de la tarjeta y comience la iniciación del sistema.
- Labomat3 solo mantiene en EPROM una versión reducida del sistema que permite únicamente solicitar una petición por la red para establecer una conexión y cargar de un fichero la imagen que contenga el sistema completo. Esta manera de operación hace que el mantenimiento del sistema sea más simple, al no ser necesario cambiar el contenido de la EPROM cada vez que se realice una actualización del sistema.
- Un **Compilador cruzado** de lenguaje C, que sirve para compilar el código fuente del sistema operativo y también los programas de usuario para ser ejecutados en el microprocesador.
- Un programa, **LaboPC**, que desde un PC conectado directamente a la tarjeta por la conexión ethernet, permite realizar todas las funciones del monitor RS232 de labomat3, incluyendo la iniciación de la tarjeta, petición y envío de ficheros de configuración de FPGA de una manera más sencilla y transparente.
- Se dispone también de aplicaciones para PC, que permiten el prototipaje rápido de sistemas lógicos (RV56200)³, o la carga de ficheros de configuración para las FPGAs empleando acceso remoto a la tarjeta mediante un protocolo de paso de mensajes sobre TCP/IP.

Para la generación de los ficheros de configuración de las FPGA se debe emplear el software que distribuye el fabricante. En caso de que se utilicen lenguajes de descripción de

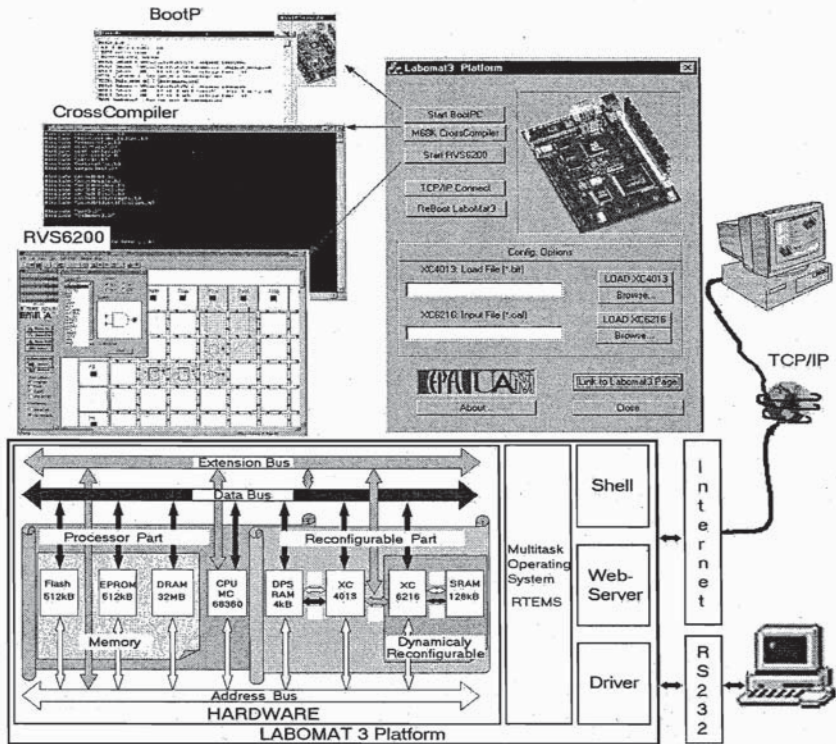


Figura 1: Arquitectura de Labomat3 y esquema de comunicaciones con los programas que forman la plataforma.

hardware se debe disponer de un sintetizador. Versiones de evaluación de libre distribución para estudiantes son distribuidas por el fabricante Xilinx Inc. en el paquete de programas FOUNDATION v1.5.

3. Un ejemplo de aplicación.

Una aplicación para introducir al alumno en el ámbito del codiseño, consiste en la realización de un coprocesador de coma flotante (FPU, *Floating Point Unit*), y comparar su rendimiento con un algoritmo que realice el mismo cálculo en el microprocesador MC68360, que no dispone de instrucciones para manejo de números en coma flotante. Los

estudiantes pueden diseñar un multiplicador de coma flotante bien sea a nivel de esquema de puertas lógicas, o bien emplear un lenguaje de descripción de hardware, y además de simular su funcionamiento deben comprobarlo realmente sobre la tarjeta Labomat3. Una parte importante del trabajo consistirá, en definir una interfaz adecuada con el procesador. El programa que ejecute el procesador debe ser el encargado de comprobar la validez de los operandos introducidos y además ser el que controle y compruebe el funcionamiento del coprocesador.

El procesador puede acceder a la FPGA como si fuera un periférico, utilizando directamente los buses de direcciones y de datos. La interfaz de comunicaciones a implementar en la FPGA se debe ocupar solo de la decodificación de la dirección y registrar los datos que llegan por el bus. Sólo se necesita generar señales de control para el asentimiento de la recepción de operandos y para indicar cuando la FPU ha terminado el cálculo. Estas señales ya estarán sincronizadas con la frecuencia de reloj utilizada. El protocolo de comunicación comienza con la escritura por parte del procesador de los operandos en direcciones específicas de la FPGA. La FPGA debe asentir la recepción de operandos. El procesador queda en estado de espera hasta que la FPGA termine el cálculo. Esta espera puede controlarse mediante lectura continua de un registro de estado de la FPGA, que indique el estado del proceso de cálculo, o bien mediante interrupciones. Finalizado el cálculo en la FPU, el procesador puede leer el resultado en una dirección prefijada. Gracias a los programas ya disponibles para la tarjeta Labomat3, el usuario no debe preocuparse de la inicialización del procesador ni de otros problemas de bajo nivel.

4. Tutoriales y material de ayuda

Con el fin de hacer más fácil la toma de contacto inicial con la plataforma Labomat3 se dispone de un enlace a una página Web con dirección <http://islwww.epfl.ch/labomat> donde se puede obtener toda la documentación técnica de la tarjeta y seguir un tutorial de los primeros pasos a seguir para comenzar.

5. Conclusiones

Labomat 3 puede emplearse para desarrollar y validar herramientas automáticas de codiseño, y como su arquitectura es conceptualmente simple, el diseñador puede concentrar su esfuerzo en el particionado y la síntesis sin tener que estar pendiente de la complejidad de la plataforma de desarrollo.

Referencias

- [1] C. Teuscher, J.O. Haenni, F.J. Gomez, H.F. Restrepo, E. Sanchez. A Reconfigurable platform for academic purposes. Proceedings of the IEEE Symposium FCCM '99, Napa USA, pp. 282-283, April 15-17, 1999.
- [2] C. Teuscher, J.O. Haenni, F.J. Gomez, H.F. Restrepo, E. Sanchez. A Tool for Teaching and Research on Computer Architecture and Reconfigurable Systems. Proceedings of the IEEE EUROMICRO 99 Conference. Milan, Italy, pp. 343-350, September 8-10, 1999.
- [3] F.J. Gomez and E. Sanchez. Teaching Digital Systems Using Dynamic Reconfiguration. Proceedings of the DCIS'99. Palma de Mallorca, pp. 75-80, November 16-19, 1999