

Linux embebido en un soft-processor Microblaze

Claudio Aciti y Nelson Acosta

INTIA/INCA - Fac. de Ciencias Exactas - Universidad Nacional del Centro de la Prov. de Bs As
Pinto 399 - Tel/Fax : (02293) 43-9680 - Tandil(7000) - Buenos Aires - Argentina
{caciti, nacosta}@exa.unicen.edu.ar

1. Introducción

Los sistemas embebidos son sistemas dedicados a una única tarea, es por ello que se los llama "sistemas de propósitos específicos". En la actualidad, los sistemas embebidos han logrado un gran auge gracias a sus diferentes campos de aplicaciones y sus bajos costos comparados con sistemas tradicionales [4, 12]. Es muy común el uso cotidiano de sistemas embebidos, ya sea en electrónica de consumo (lavarropas, heladeras, microondas, relojes, consolas de juegos, control remoto, cámaras de video, fax, CD, DVD, GPS, televisión digital), en sistemas de comunicación (sistemas de telefonía, contestadores, celulares, beepers, PDAs, routers), en automóviles (inyección electrónica, frenos, elevadores de vidrios, control de asientos, instrumentación, seguridad), en la industria (instrumentación, monitoreo, control, robótica, control de tráfico, manejo de códigos de barras, ascensores), en medicina (monitores cardíacos, renales y de apnea, marcapasos, máquina de diálisis), entre otros [7].

Por esto, hoy en día, se ha adoptado un nuevo paradigma de diseño de bajo costo el cual ha mostrado gran eficiencia al ser dedicado a una tarea específica, dado que solo se diseñan e implementan los módulos que se van a utilizar y por tanto se usa el hardware estrictamente necesario. Además puede ser optimizado en cualquier momento ya que en la mayoría de las ocasiones, es implementado sobre dispositivos que pueden ser reprogramados. Es así como los sistemas embebidos son la primera opción en el campo de la ingeniería para la solución de problemas específicos. Si un sistema embebido ofrece grandes ventajas, estas serán mayores si se cuenta con un sistema operativo que le brinde al usuario una mayor facilidad a la hora de trabajar y crear un sistema especializado mucho más robusto. Esto se ha logrado gracias a que se cuenta con el sistema operativo Linux, uno de los más estables y difundidos, que puede ser portado a este tipo de hardware a través de diferentes distribuciones (VxWorks, BlueCat, MontaVista linux, ThreadX, uC/OS-II o uClinux, entre otros) a una arquitectura Microblaze de Xilinx en FPGA [1,3,4].

2. Arquitectura Microblaze

Microblaze es un procesador de 32 bits (Figura 1) que ha sido desarrollado con unos requisitos muy rígidos de ocupación y prestaciones, debido a la limitación impuesta por los recursos disponibles en una FPGA, sin embargo está catalogado como uno de los soft-core más rápidos del mundo. Microblaze cuenta con un set de instrucciones reducido (RISC) optimizado para la implementación en una FPGA [6, 11, 13]. Este microprocesador es un IP core que debe instanciarse en el proyecto de hardware, sólo puede estudiarse a partir de la documentación del fabricante, de los documentos de resultados experimentales publicados en la red o de la experiencia propia del usuario. Aún así, es una herramienta muy potente para desarrollar proyectos relacionados con arquitecturas en paralelo, diseño, control y en general, cualquier investigación sobre software, ya que permite una comparación inmediata con otras arquitecturas y metodologías de desarrollo.

MicroBlaze puede ejecutar diferentes sistemas operativos como [10] Nucleus, ThreadX, uC/OS-II y uClinux .

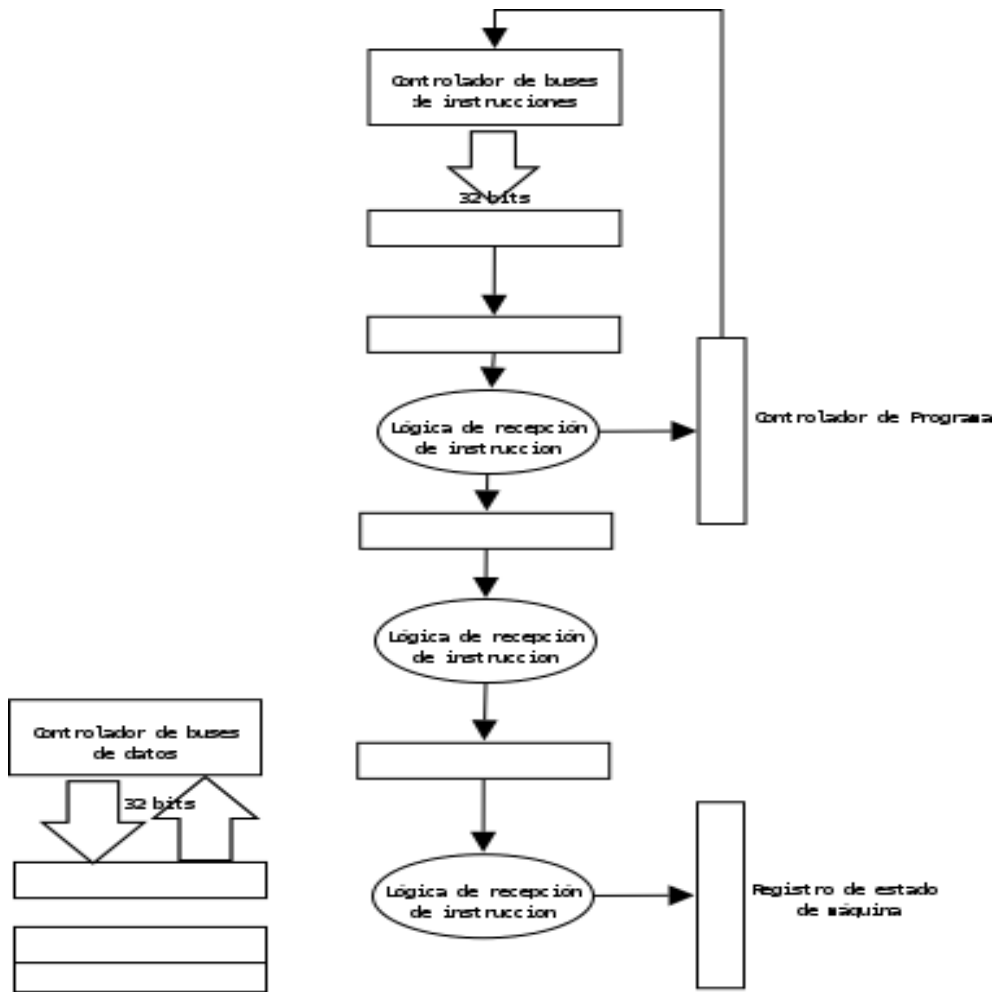


Figura 1. Esquema de bloques de Microblaze

3. Linux embebido

En el área de los sistemas embebidos, Linux es ampliamente usado. Su fiabilidad es consecuencia directa de la filosofía que lleva implícita la aportación altruista de miles de programadores de todo el mundo observando el código, mejorándolo, cambiándolo y probándolo en miles de configuraciones posibles del sistema. Linux para los dispositivos embebidos comenzó con el soporte del kernel y el compilador para los microprocesadores más populares de 32 bits: x86, ARM, PowerPC, MIPS y SH. Y luego continuó con la aparición de diferentes distribuciones de Linux con soporte para características específicas de los sistemas embebidos.

Sucesivas distribuciones, de licencias propietarias, adaptaron el kernel de Linux para usarlo en diferentes dispositivos. Es así que Montavista Linux es usado en arquitecturas PowerPC y actualmente fue portado a teléfonos móviles. BlueCat Linux fue portado a PowerPC y Microblaze. ThreadX, VxWorks y uC/OsII también han sido instalados en dispositivos como PALM o routers.

La distribución uClinux, de licencia libre, está basada en el kernel de Linux, del cual mantiene sus principales ventajas: estabilidad, capacidad superior de red y excelente soporte del sistema de archivos. El kernel de uClinux ha sido reescrito para reducirlo y para que sea más eficiente y, la principal ventaja es que ha sido desarrollado especialmente para microprocesadores que carecen de Unidad de Manejo de Memoria. uClinux además, incluye un conjunto amplio de aplicaciones típicas

de Linux, en las que se destacan: servidores web, clientes de red y versiones reducidas de interpretes de perl y python.

El proyecto para portar uClinux a MicroBlaze comenzó en Enero de 2003 en la Universidad de Queensland (Australia), de la mano de John Williams como parte de su trabajo de investigación. Este proyecto, denominado Petalinux, es "open source" y por lo tanto abierto a la colaboración. A la fecha, fue actualizado al kernel 2.6.20 estable.

4. Portando Linux a una FPGA

El proyecto se divide en tres partes: la descripción en Hardware del Microblaze; instalar un sistema operativo Linux con control de tiempo real en el procesador soft-core Microblaze; Y descargar la imagen creada en una FPGA.

4.1 Creación del Proyecto de Hardware

El diseño de la descripción en hardware del Microblaze en VHDL o Verilog se hace a través de una herramienta específica que proporciona el fabricante: EDK (Embedded Development Kit), que permite la configuración del hardware y el desarrollo de los periféricos, y el ISE (Integrated Software Environment) Foundation que proporciona las herramientas necesarias para llevar a cabo las tareas de síntesis, implementación y programación de la FPGA. Ambas distribuciones son propias de Xilinx, y tienen licencia propietaria.

Al hacer una configuración de hardware, se crea un archivo .mhs (microprocessor hardware specification), en el que se encuentran las especificaciones de los puertos del sistema, la memoria o memorias externas a la FPGA, los buses, los recursos del hardware, las instancias y configuraciones de los periféricos del sistema, tamaños de caché y habilitación de las mismas, así como el rango de memoria en el cual está el mapeado de los periféricos. En este archivo, está la especificación del software necesario para que el hardware funcione correctamente. Aquí se instancian los drivers necesarios para cada uno de los periféricos que se determinaron en el archivo de descripción del hardware, así como las opciones de compilación.

Después de generar el proyecto de hardware. Es necesario cambiar el sistema operativo por el de Petalinux [8], y se debe adicionar un proyecto de software nuevo, el cual viene en la distribución y es llamado FS-Boot (First Stage Bootloader) [2]. Este permite cargar un sistema más grande llamado U-Boot, que en principio no cabe en la caché del microprocesador, por lo tanto, debe existir una manera de cargarlo a una memoria más grande de la tarjeta de desarrollo. De esto y otros cambios se encarga el FS-Boot.

4.2 Preparación del Kernel

Ya configurado el Microblaze, se continúa con la compilación del kernel. Antes hay que tener en cuenta las siguientes consideraciones:

1. Tener las fuentes del kernel
2. Conocer cuales dispositivos de hardware se han creado sobre la tarjeta de desarrollo. Esto con el fin de compilar un kernel que se adecue perfectamente a las necesidades que se tengan.

El Sistema Operativo que se compilará tendrá la ventaja de utilizar un archivo de texto plano que contiene todas las definiciones de hardware que se hicieron sobre el microprocesador. Este archivo permite disminuir los errores de compilación. La máquina sobre la que se compila el kernel, deberá

tener SO basado en Linux, aunque es posible compilar el kernel desde windows utilizando las herramientas adecuadas[4].

Primero se indican todos los dispositivos que se van a usar, el nombre de la tarjeta para la cual se esta compilando el kernel, la forma en que la tarjeta se va a comunicar con el exterior y en que lugar del sistema se guardarán las imágenes del kernel. Una vez configurado el kernel, se construyen todas las dependencias del proyecto. Si el proceso es exitoso, las imágenes del kernel quedará compilada y lista para ser descargada.

4.3 Descarga de la imagen del kernel.

Una vez completados los pasos anteriores, se puede iniciar la descarga de la imagen del kernel. Para lograrlo se debe conectar la tarjeta de desarrollo al puerto serial del computador, y lanzar un programa que permita comunicación por este puerto. El programa se configura de la misma manera que el hardware, bits por segundo, bits de datos, paridad, bit de parada, control de flujo, etc.

Cuando la configuración esté completa, se descarga el microblaze desde el EDK. Con lo cual se completa el proceso y se tiene un kernel completamente funcional sobre una FPGA.

5. Estado actual del proyecto

Hasta el momento se están estudiando alternativas de sistemas operativos para montar en una fpga. También se está estudiando el manejo de las herramientas EDK y ISE de Xilinx Inc., y las descripciones de Hardware genéricas existentes.

El equipo de trabajo está compuesto por un Director, un doctorando y un alumno de grado. Para estos últimos, este proyecto forma parte de su tesis de doctorado y de grado, respectivamente.

Se cuenta con dos placas de desarrollo, una Virtex II Pro Developer System con una FPGA Virtex II Pro modelo XC2VP30 (Figura 2), y una Spartan 3 Starter Board con una FPGA Spartan 3 XE35200 (Figura 3). Además, de las herramientas del fabricante: EDK (Embedded Development Kit) y el ISE (Integrated Software Environment) Foundation (Figura 3). Una computadora con un sistema operativo GNU/Linux Debian Etch con un kernel 2.6.18-4 estable.



Figura 2. Virtex II Pro Developer System con una FPGA Virtex II Pro.



Figura 3. Spartan 3 Starter Board con una FPGA Spartan 3.

6. Bibliografía

- [1] E. Aguayo, I. Gonzáles. "Tutorial Xilinx Microblaze-uClinux". JCRA 2004. Madrid. España. 2004
- [2] First Stage Bootloader. <http://developer.petalogix.com/wiki/UserGuide/Bootloaders/FSboot>
- [3] I. Gonzalez, E. Aguayo, S. Lopez-Buedo, "Self-Reconfigurable Embedded Systems on Low-Cost FPGAs", IEEE Micro 2007 (Vol. 27, No. 4) pp. 49-57. 2007
- [4] J. A. Jaramillo Villegas, L. M. Perez Perez, H. J. Osorio Ríos. "Linux sobre una FPGA". Ciencia et technica. Año XIII. Número 37. ISSN 0122-1701. 2007
- [5] R. H. Klenke. "Experiences Using the Xilinx Microblaze Softcore Processor and uClinux in Computer Engineering Capstone Senior Design Projects". IEEE International Conference on Microelectronic Systems Education (MSE'07) pp. 123-124. Virginia Commonwealth University, USA. 2007
- [6] J. Lazanyi. "Instruction set extension using Microblaze processor". 15th International Conference on Field Programmable Logic and Applications, IEEE, pp. 729-730, Finlandia. 2005
- [7] J. Osio, F. Salguero, J. Rapallini y A. Quijano. " Análisis de Modelos Computacionales para Sistemas Embebidos". XII Iberchip, IWS06. Costa Rica. 2006
- [8] Petalinux. <http://developer.petalogix.com/>
- [9] C.C.W. Robson, A. Boussetham, C. Bohm, "An FPGA based general purpose data acquisition controller," 14th IEEE-NPSS Real Time Conference, p. 49. 2005
- [10] O. D. Sánchez, C. I. Camargo. "Linux Embebido como Herramienta para Realizar Reconfiguración Parcial". XII Iberchip, IWS06. Costa Rica. 2006
- [11] D. Sheldon, R. Kumar, F. Vahid, D. Tullsen and R. Lysecky, "Conjoining Soft-Core FPGA Processors," 2006 IEEE/ACM International Conference on Computer Aided Design, ICCAD, pp. 694-701. 2006
- [12] Valvano, J. "Introducción a los sistemas de microcomputadores". Thomson Learning Ibero. ISBN 9706863168. 2003
- [13] Xilinx Inc., "MicroBlaze Processor Reference Guide", UG081 (V 6.3). 2006. www.xilinx.com/ise/embedded/mb_ref_guide.pdf