6° Jornadas de Desarrollo e Innovación Tecnológica | 2007
ELECTRÓNICA
Desarrollo e Innovación
Instituto Nacional de Tecnología Industrial
INTI 50

Controlador de video en modo texto compatible con VGA, descripción de hardware

Tropea, S. E.[(i)]

[(i)]INTI-Electrónica e Informátic

Introducción

En este trabajo presentamos el desarrollo de un periférico controlador de video en modo texto que puede conectarse a un monitor compatible con VGA. Dicho periférico es compatible con el estándar WISHBONE[1] para la interconexión de periféricos.

Este periférico fue diseñado para poder ser utilizado como accesorio de un microcontrolador desarrollado en nuestro laboratorio[2][3]. Los microcontroladores son similares a los microprocesadores o *CPUs* utilizados en computadoras. Las principales diferencias radican en su capacidad, mucho más modesta, su consumo, mucho menor, y el hecho de que prácticamente todos sus componentes periféricos se encuentran incluidos en el mismo circuito integrado, por ejemplo: memoria de programas. No es común encontrar en el mercado microcontroladores económicos que incluyan salida de video como periférico, estas prestaciones sólo se observan en dispositivos más grandes.

El objetivo principal de este periférico fue el de poder agregar la posibilidad de representar información en un monitor de PC estándar utilizando nuestro microcontrolador. Tanto el microcontrolador como el controlador de video debían ser sintetizables en una misma *FPGA*. Adicionalmente era deseable evitar el uso de memorias externas a los fines de simplificar el diseño. También se requería una interfaz con el monitor que fuera lo más simple posible.

Metodología

Las *FPGAs* son circuitos integrados reconfigurables y son los miembros más avanzados de la familia de circuitos lógicos programables. Una *FPGA* está compuesta por lógica combinacional, registros y mecanismos de interconexión para unir estos dos últimos. Todos estos recursos son reconfigurables de manera tal que se puede modificar el funcionamiento para que se ajuste a nuestras necesidades. Estos elementos se complementan con celdas de entrada y salida que permiten conectar nuestro circuito con el exterior.

Las *FPGAs* presentan una alternativa muy interesante cuando es necesario desarrollar un circuito integrado a medida. Los circuitos integrados realizados a medida poseen costos muy elevados y sólo se justifican cuando el volumen de producción supera las decenas de miles. Esto es válido aún para los *ASIC* (*Application Specific Integrated Circuit* = Circuito Integrado de Aplicación Específica) que utilizan técnicas de fabricación de las conocidas como *semi-custom* en las que sólo una parte del proceso es específica del cliente y el resto son estándares. Cuando los volúmenes de producción son pequeños las *FPGAs* aparecen como una excelente alternativa. Las mismas no poseen grandes gastos iniciales o del tipo *NRE* (*Non-Recurring Engineering*) como en el caso de los *ASICs*. Por otra parte si se detectara un error en el diseño, o fuera necesario introducir otro tipo de cambio o mejora, las *FPGAs* pueden reconfigurarse sin ser necesario reemplazar el circuito integrado. Como contrapartida el costo por unidad de las *FPGAs* es superior, su velocidad es inferior y el consumo de energía es mayor cuando se las compara con los *ASICs*.

Dentro de una *FPGA* se puede incluir la

funcionalidad de varios circuitos integrados. Esta funcionalidad puede ser desarrollada por el mismo equipo de trabajo o adquirida a través de un tercero. Debido a que estas funcionalidades son como componentes electrónicos, pero sin su parte física, se los suele llamar componentes virtuales. En la industria se los conoce como bloques de propiedad intelectual o *IP cores*.

El desarrollo de un *IP core* para una *FPGA* es muy similar al de un *ASIC*. En el caso de la *FPGA* el proceso se encuentra simplificado gracias a que nuestro circuito utilizará recursos ya probados y algo sobredimensionados. Este diseño se realiza utilizando lo que se conoce como lenguajes de descripción de hardware. Los mismos tienen cierto parecido con los lenguajes de programación de computadoras pero poseen diferencias conceptuales muy importantes. No se trata de programar un dispositivo sino de describir el comportamiento del mismo. Luego la descripción se convierte en una configuración para la *FPGA* utilizando herramientas de síntesis.

En nuestro desarrollo utilizamos el lenguaje de descripción de hardware *VHDL* (*Very high speed integrated circuit Hardware Description Language* = Lenguaje de Descripción de Hardware para Circuitos Integrados de Muy Alta Velocidad). Se trata de un lenguaje muy utilizado por agencias gubernamentales y en el área aeroespacial. El mismo fue originalmente desarrollado por el Departamento de Defensa Norteamericano y hoy se encuentra estandarizado por la IEEE (normas 1076, 1164 y accesorias).

El desarrollo fue verificado utilizando *FPGAs Spartan* II[4] y *Spartan* 3[5] de *Xilinx*. Aún así el mismo fue realizado de manera tal que pudiera ser sintetizado con cualquier *FPGA*.

Las herramientas de desarrollo utilizadas fueron las recomendadas por el proyecto FPGALibre[6][7][8] impulsado por nuestro laboratorio. Para este desarrollo se utilizaron estaciones de trabajo que corren Debian[9] GNU[10]/Linux.

Debido a que se pretendía evitar la necesidad de utilizar memoria externa y a que este desarrollo debía poder sintetizarse en *FPGAs* tan pequeñas como la *Spartan* II 100, evitando el consumo de todos sus recursos, se decidió limitar las prestaciones del periférico. Con esto en mente se seleccionó un modo de 40 columnas y 25 líneas, utilizando 8 colores diferentes y sólo 64 caracteres diferentes.

Se seleccionó una frecuencia de sincronismo horizontal de 31,25 kHz y una vertical de aproximadamente 70 Hz, compatibles con los estándares VGA y VESA[11].

Para la verificación del periférico se desarrolló un banco de pruebas escrito en *VHDL* cuya salida es interpretada por un programa escrito en C++. Este último verifica las frecuencias de sincronismo y realiza una representación gráfica de la imagen que se observará en el monitor[12] *(ver Fig. 1)*.

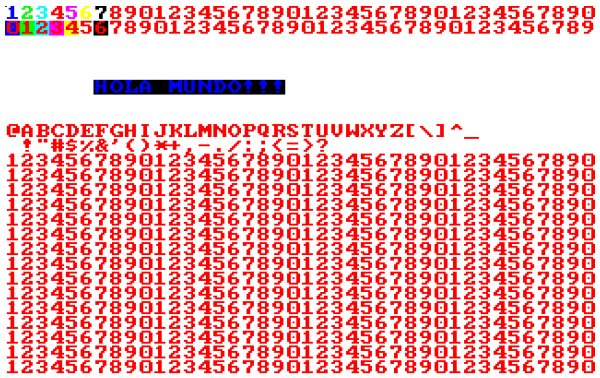

Fig. 1: Imagen generada por el banco de pruebas.

Resultados

Se obtuvo la descripción de un periférico capaz de generar una señal de video compatible con los estándares del mercado, escrito en *VHDL* estándar, y por lo tanto útil para ser utilizado con cualquier *FPGA*. En este caso el único requisito es que la *FPGA* contenga al menos 12.000 bits de memoria SRAM.

Gracias a la correcta selección de sus características el periférico sólo insumió el 50 % de la memoria disponible en una *FPGA Spartan* II 100.

En lo que se refiere al área el mismo insumió aproximadamente 75 *slices*, esto es 6,25 % de una *Spartan* II 100 y sólo 0,52 % de una *Spartan* 3 1500.

El circuito de conexión entre la *FPGA* y el monitor resultó muy simple *(ver Fig. 2)*.

Conclusiones

El periférico realizado logró cumplir con los

requisitos de ser compacto, compatible con los estándares de monitores modernos y de muy fácil conexión. La utilización de sólo ocho colores simplificó el circuito de conexión con el monitor, el mismo consta de cinco resistores.

Aún cuando el periférico necesita del uso de bloques de memoria (BRAM) de la *FPGA* no fue necesario utilizar elementos del lenguaje *VHDL* que no fueran estándares o que dependieran del fabricante de la *FPGA*.

Se determinó que, aún cuando este tipo de periféricos sólo se encuentran disponibles en microcontroladores de alta gama, la utilización de una *FPGA* permite utilizarlos con microcontroladores pequeños. Si bien existen casos documentados en los cuales microcontroladores de este tipo han sido utilizados para generar señales de video por software[13] las ventajas de disponer de un periférico que lo haga por hardware son apreciables. Uno de los detalles de mayor importancia es que la generación de la señal por hardware permite obtener imágenes más complejas y no consumir capacidad de procesamiento del microcontrolador.

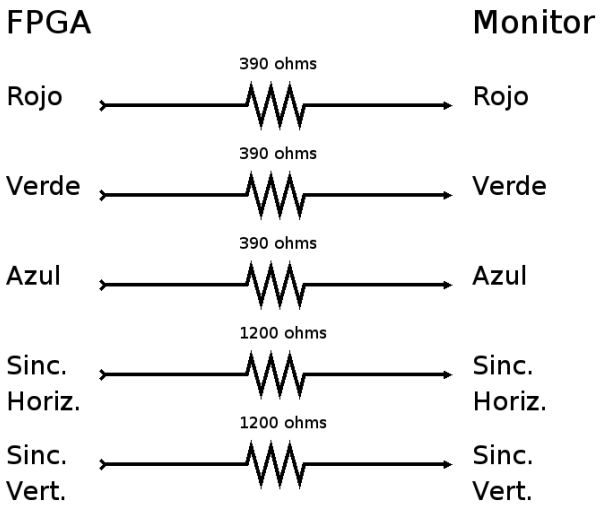

Fig. 2: Circuito de conexión entre la *FPGA* y el monitor.

La utilización de las herramientas propuestas por el proyecto FPGALibre mostró ser adecuada para este desarrollo.

Referencias

[1] Silicore and OpenCores.Org, "WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores", http://prdownloads.sourceforge.net/fpgalibre/wbspec_b3-2.pdf?download

[2] S. E. Tropea, J. P. D. Borgna, "Microcontrolador compatible con PIC16C84, bus Wishbone y video", *FPGA* Based Systems, ISBN 84-609-8998-4, 2006.

[3] S. E. Tropea, "Microntrolador compatible con *PIC* 16C84, descripción de hardware", 6° Jornadas de Desarrollo e Innovación Tecnológica, Instituto Nacional de Tecnología Industrial, Argentina, 2007.

[4] *Xilinx Spartan* II *FPGA*, http://www.xilinx.com/products/silicon_solutions/fpgas/spartan_series/spartan2_fpgas/index.htm

[5] *Xilinx Spartan* 3 *FPGA*, http://www.xilinx.com/products/silicon_solutions/fpgas/spartan_series/spartan3_fpgas/

[6] S. E. Tropea, D. J. Brengi, J. P. D. Borgna, "FPGALibre: Herramientas de Software Libre para diseño con *FPGAs*", *FPGA* Based Systems, ISBN 84-609-8998-4, pp 173-180, 2006.

[7] S. E. Tropea, D. J. Brengi, J. P. D. Borgna, S. N. Gwirc, "FPGALibre: Herramientas de Software Libre para diseño con *FPGAs*", 6° Jornadas de Desarrollo e Innovación Tecnológica, Instituto Nacional de Tecnología Industrial, Argentina, 2007.

[8] Proyecto *FPGA* Libre, http://fpgalibre.sourceforge.net/

[9] Debian project, http://www.debian.org/

[10] GNU project, http://www.gnu.org/

[11] Video Electronics Standards Association, http://www.vesa.org/

[12] S. E. Tropea, J. P. D. Borgna, "Creación de bancos de prueba complejos usando Software Libre", 1st Southern Conference on Programmable Logic - SPL 2005, Mar del Plata, 2005 (http://utic.inti.gov.ar/publicaciones/Bancos_de_prueba_usando_SL.pdf)

[13] R. Gunee, "How to generate video signals in software using PIC", 1998, http://www.rickard.gunee.com/projects/video/pic/howto.php

Para mayor información contactarse con:

Ing. Salvador E. Tropea - salvador@inti.gov.ar