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

Interfaz PS/2 maestro y esclavo, descripción de hardware

Tropea, S. E.[(i)]; Galluzzi, D.[(i)]

[(i)]INTI-Electrónica e Informática

Introducción

En este trabajo presentamos el desarrollo de dos periféricos que implementan la interfaz de comunicación PS/2[1][2]. Estos periféricos pueden ser usados por un microprocesador utilizando el estándar de interconexión WISHBONE[3].

El *bus* PS/2 es el puerto que utilizan gran cantidad de *mouses* y teclados para comunicarse con computadoras personales. Por esta razón existen en el mercado una gran cantidad de dispositivos de entrada que pueden conectarse al mismo. Al mismo tiempo un dispositivo que implemente este protocolo puede conectarse a cualquier PC moderna.

El objetivo de este trabajo consistió en implementar dos periféricos sintetizables en cualquier *FPGA* y que permitieran implementar la modalidad de maestro y de esclavo del *bus* PS/2. Pudiendo así crearse dispositivos basados en *FPGAs* que aprovecharan teclados y/o punteros tipo *mouse* o bien que reemplazaran a los mismos.

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* de *Spartan* 3[4] de *Xilinx*. Aun 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[5][6][7] impulsado por nuestro laboratorio. Para este desarrollo se utilizaron estaciones de trabajo que corren Debian[8] GNU[9]/Linux.

Se estudió el funcionamiento del *bus* PS/2. El mismo es un protocolo de comunicación serie, sincrónico y *half-duplex*. Esto quiere decir que la comunicación se realiza en un único sentido debiendo detenerse para invertir el sentido de la misma. Debido a estas características el *bus* necesita de sólo tres cables: datos, reloj y referencia (tierra). El *bus* cuenta con información de paridad para la detección de errores. El reloj utilizado en la comunicación es generado por el esclavo.

La interfaz de comunicación con el microcontrolador se implementó de acuerdo con el estándar de interconexión WISHBONE. Tanto el periférico maestro como el esclavo poseen dos puertos, uno de datos y otro de estado. El módulo esclavo posee una salida de interrupciones para indicar que un nuevo dato ha sido recibido.

Luego se procedió a la escritura de un banco de pruebas. Para verificar el funcionamiento del conjunto se realizó un simulador de teclado que genera los códigos de teclas presionadas y los envía utilizando el módulo esclavo. Por otro lado el banco de pruebas recibe y verifica los códigos de las teclas utilizando el módulo maestro *(ver Fig. 1)*. También se incluyó la simulación de las luces indicadoras de estado de teclado numérico y similares, esto precisa de una comunicación bidireccional entre maestro y esclavo.

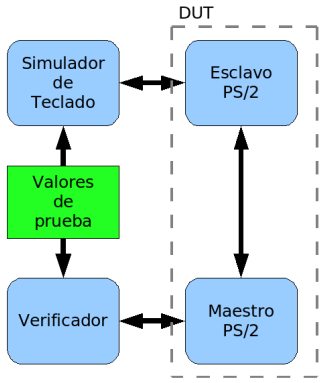

Fig. 1: Esquema del banco de pruebas simulado.

Para la verificación en hardware se creó una pequeña descripción de hardware que toma los datos provenientes de un teclado, utilizando el módulo maestro, y que los ofrece a una computadora, utilizando el módulo esclavo *(ver Fig. 2)*. De esta manera ambos módulos fueron probados simultáneamente. La misma se llevó acabo utilizando una *FPGA Spartan* 3 200 (XC3S200-FT256).

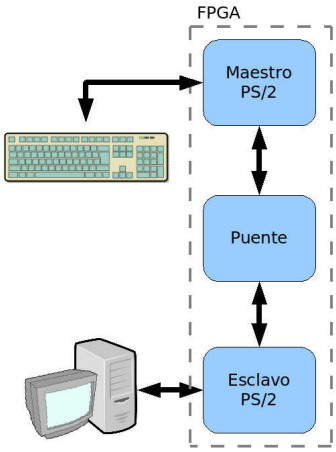

Fig. 2: Esquema de la descripción de hardware verificada.

Resultados

Se obtuvieron dos periféricos compatibles con el *bus* de interconexión WISHBONE. El maestro es capaz de comunicarse con dispositivos del tipo teclado o *mouse*. Por otro lado el esclavo permite la implementación de periféricos que reemplacen un teclado y/o *mouse* y que puedan conectarse a cualquier computadora personal.

El área ocupada por el periférico maestro fue de 127 *slices* lo que corresponde al 6,61 % de la *FPGA* antes mencionada. Por otro lado el esclavo

ocupó 123 *slices* lo que corresponde al 6,41 % de la misma *FPGA*. Estos valores corresponden a aproximadamente el 0,88 % de una *Spartan* 3 1500.

Conclusiones

Se obtuvieron dos periféricos de tamaño razonable y por lo tanto que permiten su uso aun en *FPGAs* de tamaño reducido como la *Spartan* 3 200.

La posibilidad de utilizar dispositivos estándares y económicos, como lo son los teclados y punteros de PC, en aplicaciones embebidas permite reducir los costos y el tiempo de desarrollo. En particular si se tiene en cuenta que en aplicaciones embebidas es común utilizar dispositivos especialmente diseñados y donde la lectura de las teclas se realiza por barrido de una matriz.

El periférico esclavo abre la posibilidad al desarrollo de dispositivos de entrada para PCs basados en *FPGAs*. Debido a que las *FPGAs* se caracterizan por sus altas prestaciones, en términos de velocidad, es posible pensar en dispositivos de entrada innovadores como podría ser un puntero controlado por el movimiento de la cabeza de un discapacitado.

La utilización del estándar de interconexión WISHBONE abre la posibilidad a usar este desarrollo en conjunto con otros desarrollos realizados por nuestro laboratorio y/o disponibles en internet, tal es el caso de los *IP cores* disponibles en el proyecto OpenCores[10].

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

Referencias

[1] A. Chapweske, "The PS/2 Mouse/Keyboard Protocol", http://www.Computer-Engineering.org, 1993.
[2] C. Peacock, "Interfacing the AT keyboard", http://www.beyondlogic.org/keyboard/keybrd.htm, 2005.
[3] 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
[4] *Xilinx Spartan* 3 *FPGA*, http://www.xilinx.com/products/silicon_solutions/fpgas/spartan_series/spartan3_fpgas/
[5] 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.
[6] 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.
[7] Proyecto *FPGA* Libre, http://fpgalibre.sourceforge.net/
[8] Debian project, http://www.debian.org/
[9] GNU project, http://www.gnu.org/
[10] OpenCores.Org, http://www.opencores.org/

Para mayor información contactarse con:

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