



## Periférico controlador de *display* alfanumérico, descripción de hardware

Melo, R. A.<sup>(1)</sup>; Tropea, S. E.<sup>(1)</sup>

<sup>(1)</sup>INTI-Electrónica e Informática

### Introducción

En este trabajo presentamos el desarrollo de un periférico que implementa la interfaz de comunicación con un *display* alfanumérico basado en un controlador HD44780<sup>[1]</sup>. Dicho controlador se ha convertido en un estándar de industria cuyas especificaciones funcionales son imitadas por la mayoría de los fabricantes<sup>[2]</sup>. Este periférico puede ser usado por un microprocesador utilizando el estándar de interconexión WISHBONE<sup>[3]</sup>.

El *display* alfanumérico es utilizado en innumerable cantidad de equipos electrónicos. Es de común utilización en pequeños y medianos desarrollos que necesiten interactuar con el ser humano, entregando información inteligible por el mismo.

El objetivo de este trabajo consistió en implementar un periférico, sintetizable en cualquier *FPGA* (*Field Programmable Gate Arrays* = Arreglo de Compuertas Programable en Campo), que permitiera simplificar la utilización de un *display*. Esto es, que el periférico se encargue del manejo del protocolo de comunicación con el *display*. Se deseaba que el mismo, opcionalmente, ofreciera una memoria tipo *FIFO* (*First In First Out* = Primero que Entra, Primero que Sale) para los datos a procesar. Esto permitiría disminuir el tiempo de procesamiento del controlador principal.

### 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 combinatorial, 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 específico 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*<sup>[4]</sup> 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*<sup>[5][6][7]</sup> impulsado por nuestro laboratorio. Para este desarrollo se utilizaron estaciones de trabajo que corren *Debian*<sup>[8]</sup> *GNU*<sup>[9]</sup>/*Linux*.

Tras el análisis de los usos más frecuentes se optó por la configuración que necesitaba de la menor cantidad de cableado posible, esto es cuatro líneas de datos, dos de control y dos de alimentación. No se utilizó la línea de control de escritura/lectura, que sólo tiene utilidad en aplicaciones muy particulares que escapaban a los intereses de este desarrollo.

La interfaz de comunicación con el microcontrolador se implementó de acuerdo con el estándar de interconexión *WISHBONE*. Se implementaron tres registros, dos de escritura, para datos y comandos, y uno de lectura para el estado.

Se dotó al periférico de una salida de interrupciones, que indica cuando ha concluido la transferencia de un dato, y de una salida para indicar si el periférico se encuentra en condiciones para recibir un nuevo dato.

Se agregaron dos características opcionales. Por un lado el uso de una *FIFO*, para el tratamiento de múltiples datos de entrada de manera sencilla, y por el otro la inclusión del proceso de inicialización del *display*, evitando así realizarlo en el dispositivo que lo controla.

Luego se procedió a la escritura de un banco de pruebas. Para verificar el funcionamiento del conjunto se realizó un generador de datos de entrada y un simulador de *display*, para procesar los datos de salida (ver Fig. 1).

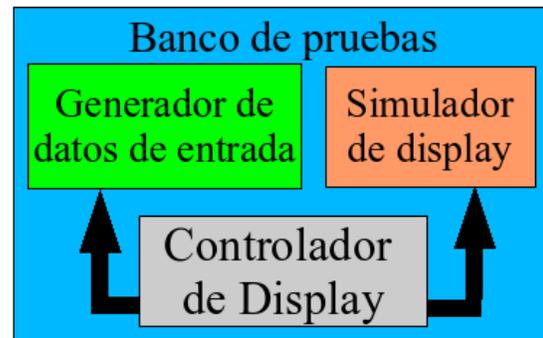


Fig. 1: Esquema del banco de pruebas simulado.

La verificación en hardware utilizó dos fuentes de datos: una constaba del generador de datos utilizado en la verificación del banco de pruebas y la otra permitió controlar el *display* utilizando una PC. Esto último se logró utilizando un puente entre el puerto paralelo de la PC, en modo EPP, y el *bus WISHBONE*<sup>[10]</sup>. De esta manera el periférico fue probado con diversas fuentes de datos. Las pruebas se llevaron a cabo utilizando varias de las configuraciones opcionales, con y sin *FIFO* y con o sin inicializador. La verificación se llevó a cabo utilizando una *FPGA Spartan II* (XC2S100-5-PQ208) y un *display* modelo 161A<sup>[11]</sup> (ver Fig. 2).



Fig. 2: *display* modelo 161A.

## Resultados

Se obtuvo un periférico compatible con el estándar de interconexión *WISHBONE*, de fácil utilización y suficiente control para la mayoría de los desarrollos.

El área ocupada por el periférico varía entre 36 y 100 *slices*, dependiendo que se incluyan o no la *FIFO* y el inicializador, lo que corresponde a entre el 2,08% y el 5,79% de la *FPGA* antes mencionada. Esto corresponde aproximadamente a entre el 0,25% y el 0,69% de una *Spartan 3*<sup>[12]</sup> 1500.

## Conclusiones

Se obtuvo un periférico de tamaño reducido para lo que es una *FPGA* económica y de bajas prestaciones.

La posibilidad de utilizar dispositivos estándares y

---

económicos, como lo son este tipo de *displays*, permiten reducir los costos y el tiempo de desarrollo.

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<sup>[13]</sup>.

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

## Referencias

- [1] Hitachi HD44780, <http://web.mit.edu/6.115/www/datasheets/44780.pdf>
- [2] "Wikipedia, la enciclopedia libre", <http://es.wikipedia.org/wiki/LCD>
- [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](http://prdownloads.sourceforge.net/fpgalibre/wbspec_b3-2.pdf?download)
- [4] *Xilinx Spartan II FPGA*, [http://www.xilinx.com/products/silicon\\_solutions/fpgas/spartan\\_series/spartan2\\_fpgas/index.htm](http://www.xilinx.com/products/silicon_solutions/fpgas/spartan_series/spartan2_fpgas/index.htm)
- [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] A. Trapanotto, D. J. Brengi, S. E. Tropea, "Descripción de hardware para interfase con puerto paralelo tipo IEEE 1284", 5° Jornadas de Desarrollo e Innovación Tecnológica, Instituto Nacional de Tecnología Industrial, Argentina, 2004.
- [11] DisplayTech Ltd 161A Series, <http://www.displaytech.com.hk/pdf/char/161a%20series-v11.PDF>
- [12] *Xilinx Spartan 3 FPGA*, [http://www.xilinx.com/products/silicon\\_solutions/fpgas/spartan\\_series/spartan3\\_fpgas/](http://www.xilinx.com/products/silicon_solutions/fpgas/spartan_series/spartan3_fpgas/)
- [13] OpenCores.Org, <http://www.opencores.org/>

Para mayor información contactarse con:

Ing. Salvador E. Tropea - [salvador@inti.gov.ar](mailto:salvador@inti.gov.ar)