

## **IMPLEMENTACIÓN DE LA LÓGICA DE CONTROL DE UN SISTEMA DE ADQUISICIÓN DE DATOS MEDIANTE UN CPLD**

F. J. Quiles, J. I. Benavides, R. Vargas Y M. A. Ortiz  
*Departamento de Electrotecnia y Electrónica. Escuela Politécnica Superior.  
Universidad de Córdoba. 14004-Córdoba. España.*

*Se describe brevemente el diseño del módulo de control de una placa de adquisición de datos mediante un CPLD. Se analizará las características del módulo de control y su implementación. Se justifica la tecnología y el tipo de CPLD empleado y se describe brevemente la herramienta software utilizada. Se finaliza indicando los resultados obtenidos. Esta se ha desarrollado como aplicación docente en un Proyecto Fin de Carrera, y se usará en las prácticas de la asignatura de Interfaces y Periféricos*

### **1. Introducción**

Una de las líneas de nuestro grupo tiene como objetivo el diseño y desarrollo de sistemas de adquisición de datos (SAD) y programas de aplicación para implementar instrumentos de medida virtuales, que puedan ser aplicados en la docencia. Esta aplicación tiene dos vertientes. Por una, disponer de herramientas que permitan automatizar o facilitar las prácticas de electrónica, tanto analógica como digital. Por otra, plantear sistemas electrónicos de cierta complejidad, y acordes a la tecnología actual en los Proyectos Fin de Carrera.

Dentro de esta línea se han desarrollado tres SAD para el bus AT, denominadas ADQLOW, ADQ y ADQLOW2. La placa ADQLOW es un sistema de muy bajo coste, en la que el proceso de conversión está controlado completamente por el microprocesador. ADQ es un sistema de altas prestaciones, que ofrece varios modos de funcionamiento automáticos para la conversión analógica a digital y digital a analógica. Como el hardware de control es bastante complejo se implementó mediante varios PLD Simples (PALCE22V10), y bloques funcionales MSI. ADQLOW2 es el último SAD que se ha desarrollado, y el objetivo que se planteó fue obtener una excelente relación prestaciones-coste. Por ello, se diseñó el bloque de control mediante un CPLD. Este ha sido el objetivo de un Proyecto Fin de Carrera correspondiente a un alumno de la titulación de Ingeniero Técnico en Informática de Sistemas. Además de la placa de adquisición de datos, incluye una librería de funciones para facilitar el desarrollo de los programas de aplicación.

En esta comunicación se describe brevemente el diseño del módulo de control de la placa ADQLOW2 mediante un Dispositivo Lógico Programable Complejo (CPLD). Comenzaremos analizando las características de la placa para indicar el diagrama de bloques del módulo de control. A continuación se describirá su implementación y finalmente se indicarán los resultados obtenidos.

## 2. Características de la placa ADQLow2

Se diseñó con el objetivo de disponer de un Sistema [1] con una excelente relación prestaciones-coste, que permitiese automatizar por hardware el proceso de conversión A/D. Para no extendernos demasiado, indicaremos solamente las características más importantes:

- Resolución de 12 bits tanto para la conversión A/D como D/A
- Conversión A/D automática de varios canales con cambio de canal. Modo ráfaga o normal
- Frecuencia de conversión A/D y número de muestras programable por software
- Ocho ganancias diferentes programables por software para cada canal de entrada
- Fifo de hasta 4Km de memoria buffer para la conversión A/D
- Transferencia de datos mediante programa, interrupción o DMA
- Varias peticiones de interrupción configurables por software
- Dos canales de salida analógicos. Conversión D/A controlada por software
- Tres puertos digitales de ocho bits y dos timers de 16 bits disponibles para el usuario

El modo ráfaga permite adquirir varios canales sucesivamente a la máxima frecuencia del convertidor. Esto permite realizar un muestreo más real si la frecuencia de conversión es baja.

## 3. Diagrama de bloques del módulo de control

El diagrama de bloques de la Lógica de Control se muestra en la figura 1. El componente decodificador genera los Chip Select de los distintos periféricos de la placa, entre los que cabe destacar el timer 82C54, el PPI 82C55, el convertidor D/A y la lectura de la fifo.

El control del interface al bus AT genera las señales que controlan los transceivers de datos. Los registros de escritura y lectura implementan los señales de configuración de los modos de funcionamiento de la conversión A/D y D/A y otros periféricos, y los registros que almacenan la ganancia de cada canal.

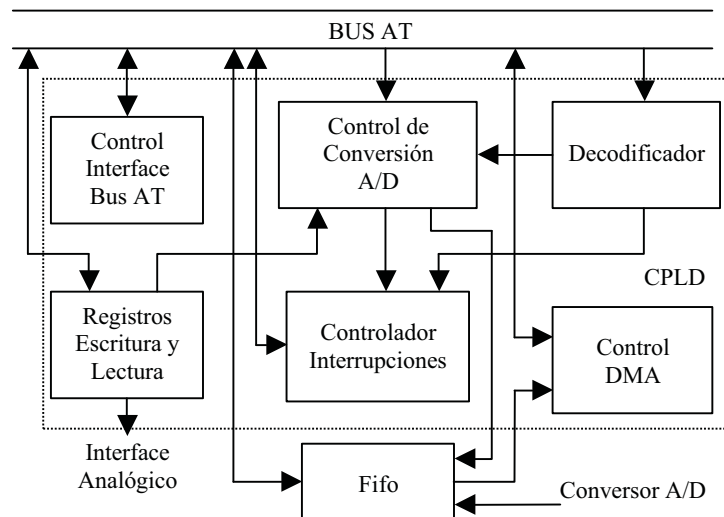


Figura 1. Bloque de control del Sistema ADQLow2

El control de la conversión A/D genera la secuencia de señales para que el proceso de conversión se realice según el modo de funcionamiento configurado, entre los que cabe destacar, el muestreo de varios canales en modo ráfaga o normal. En éstos, de una manera automática se cambia de canal, selecciona su ganancia y se genera el inicio de conversión. Por otra parte, implementa un timer de 16 bits para programar la frecuencia de conversión.

El controlador de interrupciones genera una petición de interrupción al bus AT a partir de varias causas, entre las que cabe destacar, fin de proceso de conversión A/D, overflow en la fifo y fin de cuenta de los timers. El control de DMA activa una de las peticiones de DMA de

16 bits si la FIFO no está vacía, y controla el acceso de lectura. Se puede seleccionar por software el canal de DMA y una petición de interrupción entre 4 posibles.

#### **4. Implementación de la lógica de control**

Dado que el bloque de control tiene cierta complejidad, si se implementa con bloques funcionales MSI y puertas lógicas sería necesario un número elevado de circuitos integrados. Por ello, se decidió usar Dispositivos Lógicos Programables (PLD). En la actualidad la oferta de estos dispositivos es muy amplia, variando desde los PLD Simples a las FPGA, que corresponden a los que tienen el menor y mayor número de puertas integradas, respectivamente. En la parte intermedia se encuentran los CPLD, que tienen la ventaja de que integran un número medio de puertas, por lo que su coste es inferior respecto a las FPGA y a su vez pueden funcionar a una frecuencia superior. Por tanto, éste es el dispositivo idóneo para la implementación de la lógica de control.

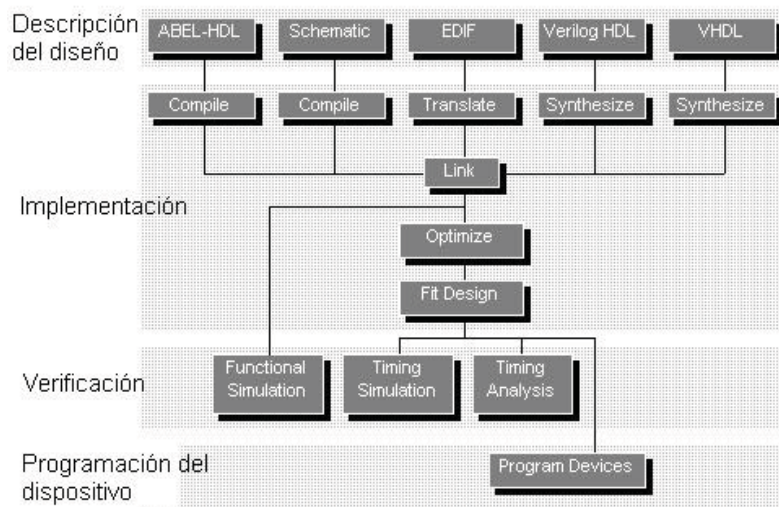
Cuando se selecciona un CPLD, al igual que cualquier otro PLD, lo primero que se debe tener en cuenta, es la disponibilidad de la herramienta software necesaria para su desarrollo. En nuestro caso se ha optado por la familia de CPLD de Lattice Semiconductor, ya que éste distribuye gratuitamente su herramienta IspDesignExpert, que permite automatizar el proceso de diseño de todos sus CPLD hasta 600 macrocélulas. El diseño se puede describir mediante esquemas eléctricos o ficheros fuente ABEL-HDL, VHDL o Verilog. Se ha seleccionado el lenguaje ABEL[2], ya que permite realizar la descripción de una manera más sencilla y exacta, por lo que se puede aprovechar mejor las características de los CPLD.

Para seleccionar un CPLD se debe analizar varias características, como son, la arquitectura interna y número de macrocélula, que determina la capacidad para implementar lógica digital, el número de patillas de E/S, el coste y el encapsulado. En base a lo anterior, se ha optado por el M4-128N/64[3], ya que dispone de 128 macrocélulas, que son suficientes para implementar la lógica de control, y tiene un encapsulado PLCC de 84 patillas. Este encapsulado tiene las patillas necesarias para generar todas las señales externas de control, y se puede usar un zócalo PLCC-PGA para insertarlo en la placa, por lo que no se debe emplear un sistema de soldadura SMD.

La metodología de diseño [4] consta de varias fases, que pueden implicar uno o varios procesos, tal y como se indica en la figura 2. Una vez que se ha determinado las especificaciones del diseño, se describe mediante el lenguaje ABEL-HDL. Debido a la complejidad del bloque de control se ha descrito como un sistema jerárquico de dos niveles, utilizando un fichero fuente para cada componente del nivel inferior, y un fichero fuente raíz para describir su estructura, es decir unir entre sí los componentes del nivel inferior. Los componentes corresponden a los bloques de la figura 1 encerrados en el cuadro de línea discontinua. Una vez descrito el diseño se compila. La herramienta IspDesignExpert abre la ventana .log en la que indica el proceso que se está ejecutando y los errores o warning que se generan. El compilador comprueba la sintaxis de los ficheros fuente, y si no se generan errores convierte el diseño a ecuaciones lógicas, que es el formato que utilizarán los procesos siguientes. De lo contrario el proceso se detiene y habrá que corregirlos y volver a ejecutar el compilador.

A continuación se realiza una simulación funcional utilizando los vectores de test descritos en un fichero .abv, que indican el valor de las entradas en cada instante de simulación. Si el resultado de la simulación es correcto se ejecuta a continuación el proceso fit, que fija el diseño en el CPLD. Este consta esencialmente de 4 fases, de las que la más importante es la denominada Place and Route. El proceso Place asigna las ecuaciones lógicas a los recursos del bloque físico, tales como, patillas de E/S, registros y clusters de términos producto. El proceso Route interconecta la matriz de conmutación según las ecuaciones lógicas.

Antes de programar el CPLD debe realizarse una simulación temporal. Esta simulación a diferencia de la funcional, tiene en cuenta los retardos de propagación para las señales de salida y los tiempos de establecimiento y mantenimiento para las entradas. Si el diseño funciona a la frecuencia especificada y las señales de salida tienen unos retardos de propagación adecuados se genera el fichero JEDEC. De lo contrario, habrá que modificar el diseño o seleccionar otro CPLD con un retardo ( $t_{pd}$ ) menor. Finalmente se carga el fichero JEDEC en el equipo de programación mediante el cual se programa el CPLD.



**Figura 2.** Flujo del diseño con IspDesignExpert

## 5. Conclusiones

El alumno ha comprobado todas las ventajas que se han expuesto en clase de teoría sobre la utilización de los PLD en el diseño de los Sistemas Digitales, como son, la reducción del número de circuitos integrados, tamaño de la placa y por tanto su costo, y sobre todo, facilitar la puesta a punto del sistema. Esta última ha sido esencial, ya que al integrar todo el diseño lógico en el CPLD, se han solucionado todos los errores relativos al control, simplemente modificando el diseño y programándolo de nuevo. De lo contrario, hubiese sido necesario diseñar una o varias PCI. Por otra parte, el alumno ha comprendido que la implementación de toda la lógica de control en un CPLD ha optimizado las prestaciones del diseño respecto a los anteriores. En concreto, se ha mejorado las características de la conversión automática de la placa ADQ al incluirse el modo ráfaga, y la relación señal/ruido frente a las placas anteriores. Esto último ha sido posible al disminuir el número de señales digitales, ya que la mayoría son internas al CPLD, y por reducirse el número de componentes, ya que se ha separado claramente la parte analógica de la digital en la placa de circuito impreso.

## Referencias

- [1] R. Vargas Caballero. *Diseño e Implementación de un SAD de Bajo Coste*. (2000)
- [2] Lattice Semiconductor Corporation. *ABEL-HDL Reference Manual*. Versión 7.0. (1998)
- [3] Lattice Semiconductor Corporation. *MACH4 CPLD Family*. (2000)
- [4] Lattice Semiconductor Corporation. *IspDesignExpert User Manual*. Versión 8.0. (1999)