

## FAMILIARIZACIÓN CON LOS DISPOSITIVOS PROGRAMABLES IN-SITU (ISP) BAJO BUS JTAG: CPLD DE XILINX

A. ROSADO; M. BATALLER; J. F. GUERRERO; A.J. SERRANO; G. CAMPS;  
J.D. MARTÍN

GPDS. Departamento de Ingeniería Electrónica. Facultad Física. Universidad de  
Valencia. Avda. Dr. Moliner, 50. 46100 Burjassot. Valencia.

email: rosado@uv.es

*Actualmente gran parte de los dispositivos lógicos programables incorporan la programación in-situ (en sistema), es decir, permiten ser programados una vez emplazados en su lugar dentro de la placa de circuito donde está destinado sin necesidad de modificación alguna del hardware. Ello se consigue mediante el bus de programación y depuración JTAG que incorporan, permitiendo modificar su programación a desde un ordenador PC a través del puerto serie o paralelo. Este trabajo pretende mostrar al alumno la dinámica de trabajo con este tipo de dispositivos así como el procedimiento a seguir. El software y los dispositivos empleados ha sido Xilinx Foundation y su familia XC9500 de CPLD.*

### 1. Introducción

Con la aparición de la tecnología de programación in-situ (isp) se han mejorado enormemente las posibilidades de empleo de las familias de dispositivos lógicos programables con grabación permanente como las CPLD, es decir, una vez programadas, su información se mantiene aunque la alimentación desaparezca, contrariamente a como ocurre en las FPGA de SRAM que dependen de una ROM externa que carga el programa de configuración cada vez que se conecta la alimentación. Anteriormente, este tipo de dispositivos únicamente permitía su reprogramación a través de un zócalo programador que obligaba a extraer el dispositivo de su emplazamiento en la placa de circuito impreso (PCB) cada vez que se deseara modificar su funcionamiento, lo que supone la imposibilidad de soldar directamente el dispositivo sobre la placa. Actualmente, a través del bus JTAG consistente en unas patillas destinadas específicamente a tal efecto, es posible conectar el dispositivo a un ordenador PC que con el software apropiado permite reconfigurar en pocos segundos el modo de funcionamiento del dispositivo lógico. Esto ofrece una gran flexibilidad a la hora de realizar el diseño ya que el modo de funcionamiento puede ajustarse tantas veces como se desee y comprobarse directamente sobre un funcionamiento real y conjunto.

La práctica propuesta se enmarca dentro de la asignatura de Diseño de Circuitos y Sistemas Electrónicos (DCSE) y su laboratorio, impartidas en 4º curso de Ingeniería Electrónica en la Universidad de Valencia. Debido a que estos estudios son de segundo ciclo exclusivamente, y dado el amplio temario a impartir, se hace un amplio recorrido por el diseño de circuitos, desde el diseño a nivel de transistor a los PLD pasando por los ASIC's, FPGA y CPLD, empleando diversas herramientas de diseño y descripción hardware, que incluye los lenguajes de alto nivel ABEL y VHDL.

El alumno debe ser capaz de completar la práctica en una única sesión de 4 horas. Para ello, en sesiones anteriores se ha introducido el lenguaje ABEL-HDL [1] con el que describirán el

diseño mediante diseños previos realizados sobre dispositivos de menor complejidad tales como dispositivos PAL sencillos. También en sesiones anteriores ha sido introducido el entorno de diseño XILINX Foundation [2] con el que deberán sintetizar el diseño realizado y programar físicamente el dispositivo.

## 2. Descripción del sistema.

El sistema a desarrollar consiste en la implementación de un sistema de giro directo-inverso programable. Se dispone de un motor DC de 12V alimentado a través de dos relés capaces de invertir el sentido de giro dependiendo del estado de las señales de control (Tabla I).

Control Motc	1
00	
0	
0	

Tabla I. Señales de control de giro del motor DC.

El giro debe realizarse cada cierto tiempo seleccionado a través de unos interruptores DIP con los que existen 8 diferentes temporizaciones (3 bits) que oscilan de 1 a 8 minutos. El tiempo transcurrido se debe medir a través de un contador que incrementa una unidad por cada pulso de reloj recibido a través de una señal de reloj externa de 1Hz; una vez realizado el giro, la cuenta debe volver a cero para comenzar un nuevo ciclo.

El recorrido que realiza el motor es semicircular (figura 1), el final de este trayecto viene marcado por dos finales de carrera que indican la llegada a sendos extremos. De este modo, para seleccionar el sentido de giro del motor es necesario tener en cuenta la posición del motor y así hacerlo girar hacia la parte opuesta.

Como elemento añadido, se incluye un display de 7 segmentos en el que se debe visualizar el número de vueltas que se realizan (de cero a nueve), y un pulsador de volteo manual para activar el giro sin necesidad de esperar el transcurso del tiempo programado.

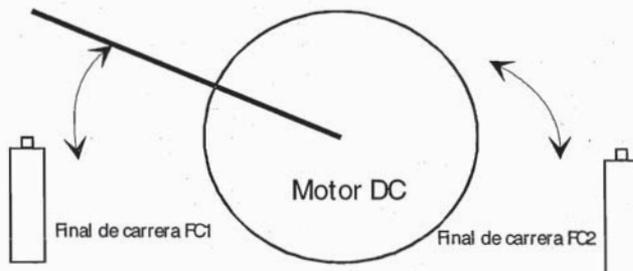


Figura 1. Esquema simplificado del sistema de giro de doble sentido.

Las placas se encuentran completamente terminadas y emplean un dispositivo CPLD XC9536 soldado directamente en placa con interface JTAG para programación in-situ.

### 3. Diseño lógico.

El diseño lógico está estructurado en tres bloques independientes donde intervienen dos partes secuenciales y una parte combinacional. Los bloques secuenciales se encargan de sendos contadores, uno de ellos para almacenar el número de segundos transcurrido en un registro interno, y otro para llevar la cuenta del número de vueltas realizado y mediante una decodificación 7 segmentos visualizarla por el display; el bloque combinacional se encarga de detectar el final de carrera presionado y en el momento que la cuenta de los segundos alcance el valor definido por el selector de tiempo (previamente calculado para cada una de las ocho posiciones), activar las salidas de control de motor en el sentido que corresponda.

Parte del código empleado es de la forma:

```
CUEN.clk = RELOJ;
CUEN.t = (CUEN.q+1) $ CUEN.q;      * contador que incrementa una unidad por cada pulso de reloj
CUEN.ar = RESET_cont;
* Se activa el giro cuando ocurre alguna de esas condiciones
  Vdchapr = (CUEN==CUEN1) & (HORA==HORA1) & FCizq & !FCdcha & !Vizq.fb & !Vdcha.fb
            # (CUEN==CUEN2) & (HORA==HORA2) & FCizq & !FCdcha & !Vizq.fb & !Vdcha.fb
            # (CUEN==CUEN3) & (HORA==HORA3) & FCizq & !FCdcha & !Vizq.fb & !Vdcha.fb
            # (CUEN==CUEN4) & (HORA==HORA4) & FCizq & !FCdcha & !Vizq.fb & !Vdcha.fb
            # (CUEN==CUEN5) & (HORA==HORA5) & FCizq & !FCdcha & !Vizq.fb & !Vdcha.fb
            # (CUEN==CUEN6) & (HORA==HORA6) & FCizq & !FCdcha & !Vizq.fb & !Vdcha.fb
            # (CUEN==CUEN7) & (HORA==HORA7) & FCizq & !FCdcha & !Vizq.fb & !Vdcha.fb
            # (CUEN==CUEN8) & (HORA==HORA8) & FCizq & !FCdcha & !Vizq.fb & !Vdcha.fb
*esta condic. sirve para el volteo manual
  # Vman & FCizq & !FCdcha & !Vizq.fb & !Vdcha.fb
*cuando no hay un final de carrera presionado, por defecto se gira hacia uno de ellos
  # !Vizq.fb & !Vdcha.fb & !FCizq & !FCdcha;
  Vdcha.ap=Vdchapr;
  Vdcha.ar=FCdcha;
  Vdcha.d=0;
  Vdcha.clk=0;
* La nueva cuenta se inicia al acabar el volteo, hasta entonces no se activa el contador
  RESET_cont = !FCizq & !FCdcha;
```

El alumno debe ser capaz de desarrollar el código y simularlo mediante vectores de test que permitan predecir un comportamiento correcto.

### 4. Implementación física.

Una vez simulado el diseño, a través del programa JTAG Programmer, incluido en el paquete XILINX Foundation, se procede a la programación de la placa y test del sistema. El alumno puede depurar errores no previstos gracias a la múltiple programación del dispositivo y observar el efecto directo que ello conlleva sobre el sistema global.

La figura 2a muestra una maqueta del sistema donde se puede apreciar la placa realizada, el motor DC y los finales de carrera, que constituyen los elementos principales. En la figura 2b se observa más en detalle la placa de circuito impreso con el dispositivo CPLD, las entradas para selección de tiempo, pulsador de giro manual, visualizador de 7 segmentos y relés de activación para el control de giro del motor DC.

Así, el alumno puede comprobar el funcionamiento del diseño directamente sobre la placa, y reprogramando tantas veces como sea necesario a través del bus de programación JTAG.

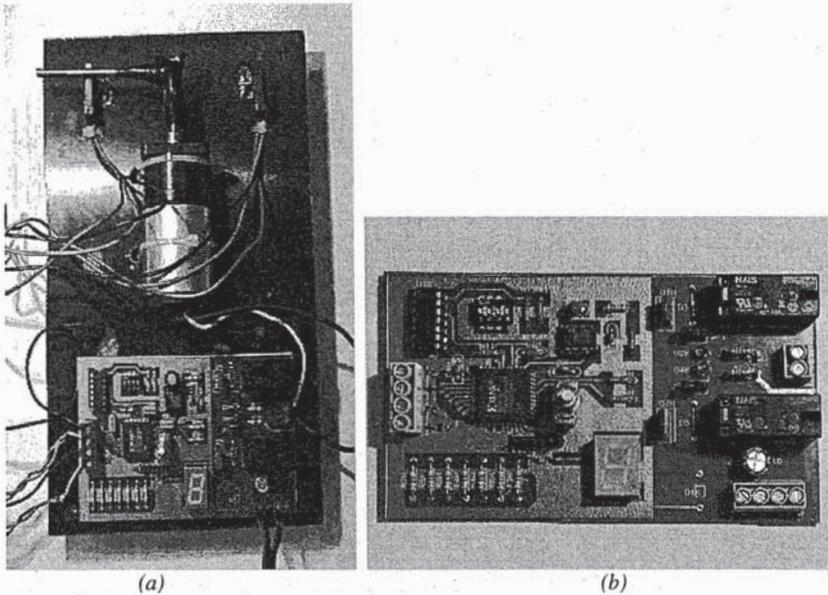


Figura 2. Maqueta con el motor DC y finales de carrera (a) y placa de circuito impreso con CPLD, relés de control del motor, display de 7 segmentos y pulsador (b).

## 5. Conclusiones.

Esta propuesta permite proporcionar al alumno una visión más general en el empleo de dispositivos lógicos programables que a veces puede parecer muy centrado en su diseño interior y se pierde el aspecto de dispositivo de ayuda al resto de elementos circundantes, pudiendo también ser capaz de controlar y realizar diversas funciones autónomas. De este modo, se comprende mejor la relación entre la descripción hardware y el funcionamiento directo del dispositivo, resultando una práctica interesante para el alumno y permitiéndoles emplear un entorno ampliamente difundido con la posibilidad de emplear la versión de estudiante de bajo coste para diseños sencillos.

## Referencias

- [1] DATA-IO: "ABEL Design Software User Manual". DATA-IO Corporation, USA 1990.
- [2] Xilinx: "Xilinx Data Book". Xilinx Development Systems. USA. 1998.