

Práctica 7: Introducción al diseño con lógica programable

Objetivo de la práctica:

El objetivo de la práctica es implementar en lógica programable un controlador del motor de un secador de pelo de tres velocidades, similar al que ya se ha implementado en prácticas anteriores utilizando componentes de lógica discreta.

Realizaremos la práctica en tres sesiones, a lo largo de las cuales iremos alcanzando diferentes objetivos parciales:

1ª sesión:

- Familiarizarse con las herramientas de diseño Xilinx ISE.
- Primera toma de contacto con la tarjeta de desarrollo de CPLD (Complex Programmable Logic Device) "Coolrunner-II CPLD Starter Board", implementando un diseño sencillo en la tarjeta.
- Planteamiento del diseño hardware del controlador del secador de pelo.

2ª sesión:

- Descripción hardware, utilizando el lenguaje VHDL, de los distintos módulos del controlador: divisor de frecuencia (*divisor.vhd*), máquina de estados (*fsm.vhd*) y módulo de jerarquía superior top-level (*top.vhd*).

3ª sesión:

- Simulación del diseño.
- Asignación de pines (*top.ucf*).
- Implementación y pruebas sobre la tarjeta.

Evaluación de la práctica:

La nota final de la práctica será calculada de la siguiente forma:

- Por asistencia y terminar el código VHDL sin errores de síntesis: 5 puntos.
- Si la síntesis del código no produce ningún *warning*: 1 punto.
- Por funcionamiento correcto en simulación: 2 puntos.
- Por funcionamiento correcto en la tarjeta: 2 puntos.

Electrónica Digital

Conocimientos previos:

Para la realización de la práctica el alumno debe conocer:

- Los conceptos de VHDL impartidos en las clases de teoría. En particular:
 - Secciones de un fichero VHDL (library, entity, architecture).
 - Tipos de datos "std_logic" e "integer" (con rango).
 - Asignaciones y operaciones lógicas.
 - Procesos y sentencias IF... ELSIF ... END IF.
 - Descripción de un proceso combinacional (lógica).
 - Descripción de un proceso síncrono (biestables).
 - Declaración e instanciación de componentes.

Material necesario:

Para esta práctica es necesario que cada alumno lleve su ordenador al aula en la que se imparta.

- Ordenador Portátil (debe llevarlo cada alumno).
- Software Xilinx ISE (debe instalarlo cada alumno en su ordenador antes de la práctica: en la web de la asignatura se encuentra disponible una guía de instalación).
- Software Digilent Adept (debe instalarlo cada alumno en su ordenador antes de la práctica: el software se encuentra disponible en la web de la asignatura).
- Kit de desarrollo CoolRunner-II CPLD Starter Board (lo llevará el profesor al aula de prácticas)

Especificaciones de la funcionalidad a implementar:

Se desea diseñar el controlador del motor de un secador de pelo de tres velocidades (parado, lento y rápido)

Las velocidades del motor se regulan mediante dos pulsadores, P1 y P2, *activos a nivel bajo*. Con el pulsador P1 se aumenta la velocidad del motor, mientras que con el pulsador P2 se reduce la velocidad del mismo. No se especifica el comportamiento si ambos pulsadores se activan simultáneamente.

El motor se controla mediante dos salidas digitales, M1, y M0, que el circuito tendrá que fijar, en función de la velocidad que se desee dar al motor, según la siguiente tabla:

Velocidad deseada	M0	M1
Parado	'1'	'1'
Lento	'0'	'1'
Rápido	'0'	'0'

Independientemente de la velocidad a la que el usuario utilice los pulsadores, las velocidades del motor sólo deben cambiar una vez por segundo.

El diseño debe tener un reset asíncrono global que lleve a todos los elementos de memoria del circuito a un estado conocido.

Diseño de la funcionalidad a implementar:

Para implementar el diseño que se nos pide, necesitaremos realizar una máquina de estados que controle el motor (M0, M1) según las pulsaciones que se vayan haciendo en P1 y P2:

fsm.vhd

- Entradas: p1, p2, enable, rst, clk. Tipo std_logic;
- Salidas: m0, m1. Tipo std_logic;

Adicionalmente, ya que el kit de desarrollo dispone de un reloj global a 8 MHz, tendremos que realizar un divisor de frecuencia que proporcione un pulso de habilitación (*enable*) para la máquina de estados una vez por segundo. La máquina de estados debe evolucionar utilizando el reloj global, pero sólo debe ir a un estado diferente en los ciclos en los que esté habilitada:

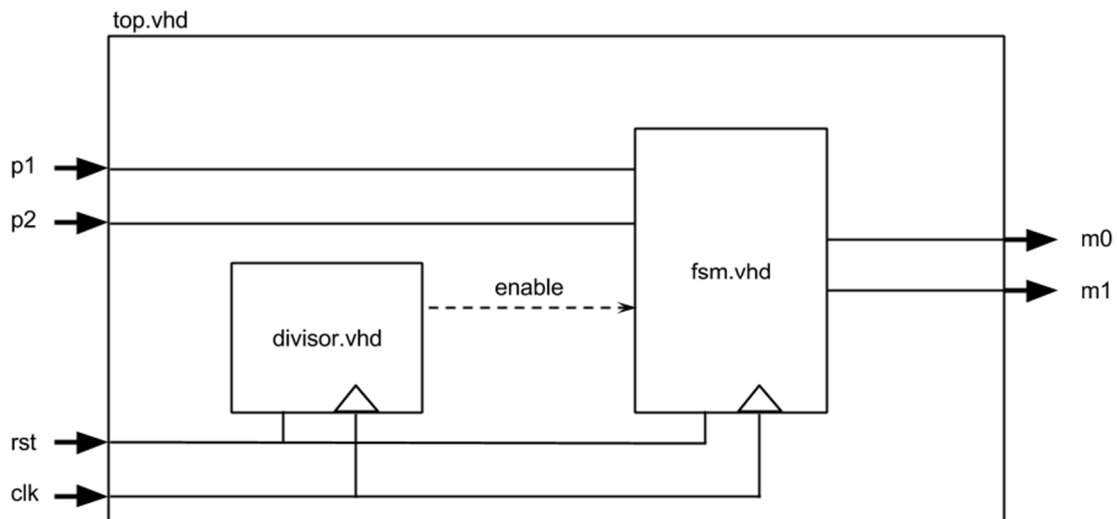
divisor.vhd

- Entradas: rst, clk. Tipo std_logic;
- Salidas: enable. Tipo std_logic;

Finalmente, tendremos que realizar un bloque de jerarquía superior que contenga los componentes anteriores y los conecte:

top.vhd

- Entradas: p1, p2, rst, clk. Tipo std_logic;
- Salidas: m0, m1. Tipo std_logic;



Asignación de pines:

Para hacer las pruebas en la tarjeta, utilizaremos los pulsadores BTN0 y BTN1 para P1 y P2 respectivamente, y el switch SW0 para el reset. Visualizaremos las salidas M0 y M1 en los leds LD0 y LD1 respectivamente. Podemos encontrar a qué pines de la CPLD están conectados los pulsadores y los leds en la serigrafía de la tarjeta. El reloj a 8 MHz está conectado al pin "P38" de la CPLD.

Con esta información crearemos el fichero *top.ucf* que añadiremos al proyecto de Xilinx.

Los ficheros *.ucf* (*User Constraints File*) son ficheros de texto plano en los que el usuario puede especificar restricciones de implementación a la herramienta. En particular, nosotros queremos especificar a qué pines de la CPLD queremos que vayan las diferentes entradas y salidas del diseño. Si no los especificamos, nuestro diseño no funcionará.

Ejemplo de formato del fichero *ucf* (el carácter # indica que se trata de un comentario):

```
# Constrain I/O pins LOcation here
NET "clk" LOC = "P38";
NET "rst" LOC = "39";
[etc ...]
```

Donde el campo NET indica la entrada o salida de la entidad top-level, y el campo LOC indica el identificador del pin al que se asignará esa entrada/salida.