

## INGENIERÍA DE COMPUTADORES 3

### Trabajo Práctico - Convocatoria Ordinaria 2012

#### INSTRUCCIONES

- El trabajo práctico debe realizarse de manera individual. No puede realizarse en grupo. Por ello, se penalizará cualquier uso compartido de las soluciones propuestas y de los códigos programados.
- El trabajo debe entregarse a través del curso virtual de la asignatura en la plataforma Alf.
- La fecha límite de entrega es el día 16 de abril.
- El alumno debe entregar un fichero comprimido, en formato zip o tar, que contenga:
  - Una memoria en la cual explique la solución a los ejercicios, incluyendo los listados documentados del código VHDL desarrollado. Este documento deberá estar en formato pdf.
  - Los ficheros del código VHDL solución a los ejercicios.

El nombre del fichero comprimido debe ser la concatenación del nombre y apellidos del alumno. Por ejemplo, LuisaGomezMartin.zip

## CRITERIOS DE EVALUACIÓN DEL TRABAJO

- Para que el trabajo pueda ser corregido, es imprescindible que el alumno entregue dentro del plazo establecido un fichero comprimido que contenga tanto la memoria en formato pdf, como el código VHDL de los ejercicios que haya realizado.
- El trabajo se compone de 2 ejercicios con varios apartados. En el enunciado se indica la puntuación de cada apartado.
- Para aprobar el trabajo es necesario que la suma de la nota obtenida en los dos ejercicios sea igual o mayor que 5.
- Si el código VHDL solución de un apartado tiene uno o varios errores de compilación, o no tiene la funcionalidad pedida, dicho apartado se valorará con cero puntos.
- Si el código solución de un apartado compila sin errores y tiene la funcionalidad pedida, la puntuación en dicho apartado será al menos el 80 % de la nota del apartado.
- Se valorará positivamente la eficiencia y la adecuada documentación del código, así como la presentación y calidad de las explicaciones proporcionadas en la memoria.

## PROCEDIMIENTO DE EVALUACIÓN EN ESTA ASIGNATURA

Para aprobar la asignatura es necesario aprobar el trabajo y aprobar el examen.

Plantearémos un trabajo para la convocatoria ordinaria (junio) y otro diferente para la convocatoria extraordinaria (septiembre). Este trabajo que está leyendo corresponde a la convocatoria ordinaria de 2012.

La nota obtenida en la convocatoria ordinaria en el trabajo y en el examen se guarda para la convocatoria extraordinaria. Es decir:

- Si un alumno aprueba el trabajo de la convocatoria ordinaria y no aprueba el examen, se le guarda la nota del trabajo para la convocatoria extraordinaria. Es decir, no debe hacer el trabajo de la convocatoria extraordinaria.
- Si un alumno no entrega o suspende el trabajo en convocatoria ordinaria, pero sí aprueba el examen en convocatoria ordinaria, entonces se le guarda la nota del examen para la convocatoria extraordinaria, debiendo aprobar el trabajo de la convocatoria extraordinaria para superar la asignatura.

La nota del trabajo y del examen no se guarda de un curso para otro.

**EJERCICIO 1**

A continuación se muestran dos funciones lógicas  $F$  y  $G$ , que dependen de las cuatro variables  $A$ ,  $B$ ,  $C$  y  $D$  de la forma mostrada a continuación:

$$F = A B C + C D$$

$$G = A B C'$$

- 1.a) (0.5 puntos) Escriba en VHDL la **entity** de un circuito que implemente las dos funciones lógicas. Es decir, que tenga cuatro entradas  $A$ ,  $B$ ,  $C$  y  $D$ , y dos salidas  $F$  y  $G$ .
- 1.b) (1 punto) Escriba en VHDL la **architecture** que describa el *comportamiento* de un circuito que implemente las dos funciones lógicas.
- 1.c) (0.5 puntos) Dibuje el diagrama al nivel de puertas lógicas de un circuito que implemente estas dos funciones. Emplee para ello puertas lógicas AND y OR de dos entradas, y puerta NOT. A continuación, escriba en VHDL la **entity** y la **architecture** de cada una de las puertas lógicas que componen el diagrama que acaba de dibujar.
- 1.d) (1 punto) Escriba en VHDL una **architecture** que describa la *estructura* del circuito que ha dibujado, instanciando y conectando las puertas lógicas que ha diseñado anteriormente.
- 1.e) (1 punto) Escriba en VHDL un banco de pruebas que permita visualizar, para todos los posibles valores de las entradas, las salidas de los circuitos diseñados en los Apartados 1.b y 1.d. Compruebe mediante inspección visual que los dos diseños funcionan correctamente.

## EJERCICIO 2

- 2.a) (1.5 puntos) Diseñe en VHDL la **architecture** de un codificador 8 a 3 con prioridad, describiendo el comportamiento del circuito. La **entity** del circuito y la tabla de la verdad se muestran a continuación.

```
entity codif_8a3 is
  port ( codigo : out std_logic_vector(2 downto 0);
        activo  : out std_logic;
        x       : in  std_logic_vector(7 downto 0) );
end entity codif_8a3;
```

x	codigo	activo
1 - - - - -	1 1 1	1
0 1 - - - - -	1 1 0	1
0 0 1 - - - - -	1 0 1	1
0 0 0 1 - - - - -	1 0 0	1
0 0 0 0 1 - - - -	0 1 1	1
0 0 0 0 0 1 - - -	0 1 0	1
0 0 0 0 0 0 1 -	0 0 1	1
0 0 0 0 0 0 0 1	0 0 0	1
0 0 0 0 0 0 0 0	0 0 0	0

- 2.b) (1.5 puntos) Programe en VHDL un banco de pruebas que testee todas las posibles entradas al circuito que ha diseñado en el Apartado 2.a. El banco de pruebas debe comparar las salidas de la UUT con las salidas esperadas, mostrando el correspondiente mensaje de error en caso de que las salidas obtenidas de la UUT no correspondan con las esperadas.
- 2.c) (1.5 puntos) Existen algunas aplicaciones en las cuales se precisa poder modificar la prioridad de las señales de entrada  $x(7)$ , ...,  $x(0)$ . Los codificadores que soportan esta funcionalidad se denominan *codificadores con prioridad programable*. Diseñe en VHDL la **architecture** de un codificador 8 a 3 con prioridad programable. La **entity** del circuito es:

```
entity codif_8a3_prog is
  port ( codigo : out std_logic_vector(2 downto 0);
        activo  : out std_logic;
        x       : in  std_logic_vector(7 downto 0);
        c       : in  std_logic_vector(2 downto 0) );
end entity codif_8a3_prog;
```

El valor de la palabra binaria de tres bits  $c(2), c(1), c(0)$  especifica cuál de las señales de entrada  $x(7), \dots, x(0)$  tiene mayor prioridad. Por ejemplo, si  $c$  vale "011", entonces la prioridad es:  $x(3), x(2), x(1), x(0), x(7), \dots, x(4)$ .

En la siguiente tabla se indica la prioridad de las señales  $x(7), \dots, x(0)$  en función del valor de las señales de control  $c(2), c(1), c(0)$ .

c	Prioridad							
	<- Mayor				Menor ->			
1 1 1	x(7)	x(6)	x(5)	x(4)	x(3)	x(2)	x(1)	x(0)
1 1 0	x(6)	x(5)	x(4)	x(3)	x(2)	x(1)	x(0)	x(7)
1 0 1	x(5)	x(4)	x(3)	x(2)	x(1)	x(0)	x(7)	x(6)
1 0 0	x(4)	x(3)	x(2)	x(1)	x(0)	x(7)	x(6)	x(5)
0 1 1	x(3)	x(2)	x(1)	x(0)	x(7)	x(6)	x(5)	x(4)
0 1 0	x(2)	x(1)	x(0)	x(7)	x(6)	x(5)	x(4)	x(3)
0 0 1	x(1)	x(0)	x(7)	x(6)	x(5)	x(4)	x(3)	x(2)
0 0 0	x(0)	x(7)	x(6)	x(5)	x(4)	x(3)	x(2)	x(1)

- 2.d)** (1.5 puntos) Programe en VHDL un banco de pruebas que testee todas las posibles entradas al circuito que ha diseñado en el Apartado 2.c. El banco de pruebas debe comparar las salidas de la UUT con las salidas esperadas, mostrando el correspondiente mensaje de error en caso de que las salidas obtenidas de la UUT no correspondan con las esperadas.