

INGENIERÍA DE COMPUTADORES 3

Trabajo Práctico - Junio 2015

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 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 2015.

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

Se desea diseñar un circuito digital que implemente las funciones F y G cuya tabla de verdad se muestra a continuación, que dependen de las tres variables x, y y z:

x	y	z	F	G
'0'	'0'	'0'	'0'	'1'
'0'	'0'	'1'	'1'	'1'
'0'	'1'	'0'	'0'	'0'
'0'	'1'	'1'	'1'	'1'
'1'	'0'	'0'	'0'	'0'
'1'	'0'	'1'	'0'	'0'
'1'	'1'	'0'	'1'	'0'
'1'	'1'	'1'	'1'	'0'

- 1.a) (0.5 puntos) Obtenga las funciones lógicas F y G a partir de la tabla de verdad. Escriba en VHDL la **entity** del circuito que implemente las dos funciones lógicas. Es decir, que tenga tres entradas x, y y z, y dos salidas F y G.
- 1.b) (1 punto) Escriba en VHDL la **architecture** que describa el *comportamiento* del circuito.
- 1.c) (0.5 puntos) Dibuje el diagrama de un circuito que implemente estas dos funciones lógicas al nivel de puertas lógicas. No es necesario que el circuito esté simplificado. A continuación, escriba en VHDL la **entity** y la **architecture** de cada una de las puertas lógicas que componen el circuito 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. Incluya en la memoria el cronograma obtenido al realizar la simulación del banco de pruebas.

EJERCICIO 2

Se desea diseñar un circuito digital combinacional con una señal de entrada de 8 bits llamada `data`, una señal de salida de 4 bits llamada `total` y una señal de salida de 1 bit llamada `error`.

El circuito ha de tener el siguiente comportamiento:

- Si no existe una secuencia de ceros en la señal de entrada (es decir, si `data` tiene el valor “1111111”), la señal `error` ha de valer ‘0’ y la señal `total` ha de valer “0000”.
- Si existe una única secuencia de ceros en la señal de entrada, la señal de salida `error` ha de valer ‘0’ y la señal `total` ha de tener como valor el número de ceros en la secuencia de ceros. Una secuencia de ceros es uno o más bits ‘0’ en posiciones consecutivas. Por ejemplo:
 - El valor de la señal de entrada `data` “11000001” da como resultado un valor de la señal `error` de ‘0’ y un valor de la señal `total` de “0101”.
 - El valor de la señal de entrada `data` “11011111” da como resultado un valor de la señal `error` de ‘0’ y un valor de la señal `total` de “0001”.
- Si existen dos o más secuencias de ceros, el valor de la señal `error` es ‘1’ y el valor de la señal `total` es “0000”. Por ejemplo, el valor de la señal de entrada `data` “00111110” da como resultado un valor de la señal `error` de ‘1’ y un valor de la señal `total` de “0000”.

2.a) (3 puntos) Escriba en VHDL la **architecture** que describe el comportamiento del circuito digital combinacional descrito. Puede emplear en el diseño del circuito la sentencia **exit**.

La **entity** del circuito se muestra a continuación.

```
entity contador is
  port ( error: out std_logic;
        total: out  std_logic_vector( 3 downto 0);
        data: in  std_logic_vector( 7 downto 0) );
end entity contador;
```

2.b) (3 puntos) Programe en VHDL un banco de pruebas que testee el circuito digital. El banco de pruebas debe generar los 10 vectores de test siguientes:

- Valor de la señal de entrada `data` “00000000”.
- Valor de la señal de entrada `data` “11111111”.
- Valor de la señal de entrada `data` “00011000”.
- Valor de la señal de entrada `data` “11100000”.
- Valor de la señal de entrada `data` “00000111”.
- Valor de la señal de entrada `data` “01111100”.
- Valor de la señal de entrada `data` “01111110”.
- Valor de la señal de entrada `data` “01001101”.
- Valor de la señal de entrada `data` “10000001”.
- Valor de la señal de entrada `data` “11110001”.

Defina en el banco de pruebas los tres tipos vectoriales siguientes:

```
type vector_test is array (0 to 9) of std_logic_vector(7 downto 0);
type error_array is array (0 to 9) of std_logic;
type total_array is array (0 to 9) of std_logic_vector(3 downto 0);
```

Los valores de los vectores de test y los valores esperados de las señales `error` y `total` se han de almacenar en constantes de estos tipos vectoriales. La asignación de valor a la señal de entrada del circuito bajo test se ha de realizar dentro de un bucle **for**.

El banco de pruebas ha de mostrar un mensaje de error en caso de que las salidas obtenidas de la UUT no correspondan con las esperadas, indicando la posición del vector de test (0 a 9) que no ha producido la salida correcta. Emplee este banco de pruebas para comprobar el diseño realizado al contestar el Apartado 2.a. Incluya en la memoria el cronograma obtenido al realizar la simulación del banco de pruebas.