

INGENIERÍA DE COMPUTADORES 3

Trabajo Práctico - Septiembre 2021

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 10 de septiembre.
- El alumno debe entregar un fichero comprimido, en formato zip o tar, que contenga:
 - o 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.
 - o Los ficheros del código VHDL solución a los ejercicios.

El nombre del fichero comprimido debe ser la concatenación de los apellidos y nombre del alumno. Por ejemplo, GomezMartinLuisa.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.

Plantaremos un trabajo para la convocatoria ordinaria (junio) y otro diferente para la convocatoria extraordinaria (septiembre). Este trabajo que está leyendo corresponde a la convocatoria extraordinaria de 2021.

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

Dada las dos funciones lógicas (F y G) mostradas a continuación, que dependen de 3 variables (x, y y z):

$$F = \text{not } x \text{ or } (y \text{ and } z)$$

$$G = x \text{ and } y \text{ and } z$$

- 1.a) (0.5 puntos) Escriba en VHDL la **entity** del circuito.
- 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. 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 los dos cronogramas obtenidos al realizar la simulación del banco de pruebas del circuito diseñado en los Apartados 1.b y 1.d.

EJERCICIO 2

2.a (1 punto) Diseñe en VHDL la **architecture** de un decodificador 2 a 4 con entrada enable activa a nivel alto, describiendo la estructura del circuito usando para ello solo puertas AND de tres entradas y puertas NOT. Debe proporcionar el código VHDL de la **entity** y **architecture** de la puerta AND de tres entradas y la puerta NOT. El diseño estructural del decodificador se ha de realizar instanciando y conectando las puertas lógicas AND y NOT diseñadas. En la memoria ha de mostrar, además de todo el código VHDL diseñado, el diagrama circuital del diseño realizado.

La **entity** del circuito y la tabla de la verdad se muestran a continuación.

```
entity decodificador2a4 is
  port ( y0, y1, y2, y3 : out std_logic;
         w0, w1       : in  std_logic;
         En           : in  std_logic);
end entity decodificador2a4;
```

En	w1	w0	y0	y1	y2	y3
'1'	'0'	'0'	'1'	'0'	'0'	'0'
'1'	'0'	'1'	'0'	'1'	'0'	'0'
'1'	'1'	'0'	'0'	'0'	'1'	'0'
'1'	'1'	'1'	'0'	'0'	'0'	'1'
'0'	'X'	'X'	'0'	'0'	'0'	'0'

Donde el valor 'X' en la tabla indica que se produce esa salida tanto para el valor '0' como el valor '1' de dicho bit.

2.b (1 punto) 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. Incluya en la memoria el cronograma obtenido al realizar la simulación del banco de pruebas del circuito diseñado en el Apartados 2.a.

2.c (2 puntos) Diseñe un decodificador 3 a 8 con entrada enable activa a nivel alto, describiendo su estructura empleando para ello únicamente dos decodificadores 2 a 4 como el diseñado en el Apartado 2.a, puertas NOT y puertas AND de dos entradas. El comportamiento de este decodificador es

análogo al descrito en el Apartado 2.a para el decodificador 2 a 4, pero con diferente número de entradas y salidas. Debe proporcionar el código VHDL de la **entity** y **architecture** de la puerta AND de dos entradas. El diseño estructural del decodificador se ha de realizar instanciando y conectando los decodificadores 2 a 4, las puertas lógicas AND y NOT diseñadas. En la memoria ha de mostrar, además de todo el código VHDL diseñado, el diagrama circuital del diseño realizado.

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

```
entity decodificador3a8 is
  port (y0, y1, y2, y3, y4, y5, y6, y7 : out std_logic;
        w0, w1, w2           : in std_logic;
        En                   : in  std_logic);
end entity decodificador3a8;
```

- 2.d)** (2 puntos) Programe en VHDL un banco de pruebas que testee el circuito diseñado en el Apartado 2.c para todos los posibles valores de las entradas. 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.

Incluya en la memoria el cronograma obtenido al realizar la simulación del banco de pruebas del circuito diseñado en el Apartado 2.c.