

# INGENIERÍA DE COMPUTADORES 3

## Trabajo Práctico - Septiembre 2022

### 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 2022.

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**

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

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

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

- 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) Simplifique las funciones lógicas y dibuje el diagrama de un circuito que implemente estas dos funciones lógicas ya simplificadas al nivel de puertas lógicas. Para ello, se ha de emplear únicamente puertas AND de dos entradas, puertas NOT y puertas XOR de dos entradas. A continuación, escriba en VHDL la **entity** y la **architecture** de cada una de las puertas lógicas AND de dos entradas, NOT y XOR de dos entradas que ha empleado en el diseño del anterior circuito.
- 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

Se quiere diseñar un circuito secuencial síncrono que opera en el flanco de subida de la señal de reloj cuyo propósito es el control del sistema de alarma de un coche que enciende o apaga una sirena.

El circuito tiene las cinco señales de entrada de un bit siguientes:

- Señal  $M$  que permite controlar la alarma según su valor sea '1' o '0'.
- Señal  $D$ , que indica si la puerta del coche está abierta o cerrada. Si  $D$  vale '1' la puerta está abierta, y si vale '0' está cerrada.
- Señal  $V$ , que indica si existe una vibración en el coche. Si  $V$  vale '1' existen vibraciones, y si vale '0' no existen vibraciones.
- Señal de reloj llamada  $clk$ .
- Señal de reset asíncrona activa a nivel alto llamada  $reset$ .

El circuito tiene una única señal de salida llamada  $S$  que indica si la sirena del coche está encendida o apagada. Si la señal  $S$  vale '1' la sirena está encendida, y si vale '0' está apagada.

El circuito se ha de diseñar como una máquina de estados de Moore con los dos estados siguientes:

- Estado ON. En este estado la sirena está encendida. Una vez el circuito entra en este estado ON, el circuito permanece en el estado ON mientras la señal  $M$  tenga el valor '1', con independencia del valor de las señales  $D$  y  $V$ . El circuito pasa del estado ON al estado OFF cuando la señal  $M$  tiene el valor '0'.
- Estado OFF. En este estado la sirena está apagada. El circuito pasa del estado OFF al estado ON solo si la señal  $M$  tiene el valor '1' y, además, la señal  $D$  tiene el valor '1' o la señal  $V$  tiene el valor '1' o ambas tienen el valor '1'.

Cuando la señal  $reset$  tiene valor '1' el circuito pasa asíncronamente al estado OFF. El resto de transiciones entre estados son síncronas, teniendo siempre lugar en el flanco de subida de la señal de reloj.

- 2.a** (3 puntos) Proporcione el código VHDL de la **architecture** del circuito secuencial anterior describiendo su comportamiento como una máquina de estados de Moore. En la memoria ha de mostrar, además de todo el código VHDL diseñado, el diagrama de estados del circuito. El diagrama de estados debe mostrar los dos estados y todas las posibles transiciones entre estados para cada uno de los posibles valores de las señales M, D y V.

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

```
entity alarma is
  port ( S      : out std_logic;
        M, D, V : in  std_logic;
        clk     : in  std_logic;
        reset   : in  std_logic);
end entity alarma;
```

- 2.b** (3 puntos) Programe en VHDL un banco de pruebas que testee el circuito que ha diseñado en el Apartado 2.a. El banco de pruebas debe comparar la salida de la UUT con la salida esperada, mostrando el correspondiente mensaje de error en caso de que la salida obtenida de la UUT no corresponda con la esperada. En el programa de test todos los arcos de las transiciones entre estados han de ser recorridos por los menos una vez. Incluya en la memoria el cronograma obtenido al realizar la simulación del banco de pruebas del circuito diseñado en el Apartados 2.a y el diagrama de estados del circuito. En el diagrama de estados se debe mostrar junto a cada flecha de la transición de estado un número que refleje el orden en que se ha recorrido dicha transición de estado.