

# INGENIERÍA DE COMPUTADORES 3

## Trabajo Práctico - Septiembre 2020

### 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 2020.

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  $F1$  y  $F2$  cuya tabla de verdad se muestra a continuación, que dependen de las tres variables  $x$ ,  $y$  y  $z$ :

x	y	z	F1	F2
'0'	'0'	'0'	'0'	'0'
'0'	'0'	'1'	'0'	'1'
'0'	'1'	'0'	'0'	'1'
'0'	'1'	'1'	'1'	'0'
'1'	'0'	'0'	'0'	'1'
'1'	'0'	'1'	'1'	'0'
'1'	'1'	'0'	'1'	'0'
'1'	'1'	'1'	'1'	'1'

- 1.a) (0.5 puntos) Obtenga las funciones lógicas  $F1$  y  $F2$  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  $F1$  y  $F2$ .
- 1.b) (1 punto) Escriba en VHDL la **architecture** que describa el *comportamiento* del circuito.
- 1.c) (1 punto) 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) (0.5 puntos) Escriba en VHDL un banco de pruebas que permita visualizar, para todos los posibles valores de las entradas, la salida del circuito cuya **entity** ha especificado en el Apartado 1.a. Emplee dicho banco de pruebas para comprobar mediante inspección visual que los dos diseños de los Apartados 1.b y 1.d funcionan correctamente. Incluya en la memoria los dos cronogramas obtenidos al realizar la simulación del banco de pruebas usando en un caso como circuito de test el circuito de Apartado 1.b y en el otro caso el circuito del Apartado 1.d.

## EJERCICIO 2

Se quiere programar en VHDL una alarma de seguridad de un museo. La alarma está compuesta por dos circuitos, el circuito A y el circuito B. El circuito B controla el funcionamiento de 3 displays de 7 segmentos, que nos permiten visualizar si la alarma está habilitada o deshabilitada (mostrando las palabras on u off). En la Figura 1.1 se muestra el circuito de alarma, compuesto por los circuitos A y B y los 3 displays.

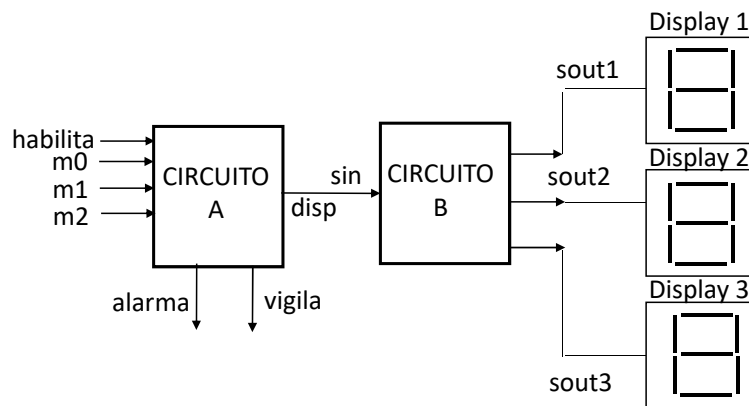


Figura 1.1: Alarma.

El circuito A tiene como entrada la señal de un bit *habilita* y 3 señales de entrada de un bit llamadas *m0*, *m1* y *m2*. Tiene como salida la señal de un bit *alarma*, la señal de dos bits *disp* y la señal de un bit *vigila*. La señal *disp* va a ser la entrada del circuito B.

El museo tiene 3 salas con un sensor de movimiento en cada una de ellas (señales *m0*, *m1* y *m2*). Mientras en la sala  $i$  ( $i = 1, 2, 3$ ) se detecta movimiento, la señal  $m_i$  tiene valor '1'. Por el contrario, si no se detecta movimiento en la sala  $i$ , la señal  $m_i$  tiene valor '0'. En el museo hay un único guarda de seguridad.

Si se detecta movimiento en dos o más habitaciones y la alarma está habilitada, se pone a '1' la señal *alarma*. En caso contrario, la señal *alarma* está a '0'. Para habilitar/deshabilitar la alarma se da a la señal *habilita* el valor '1'/'0', respectivamente.

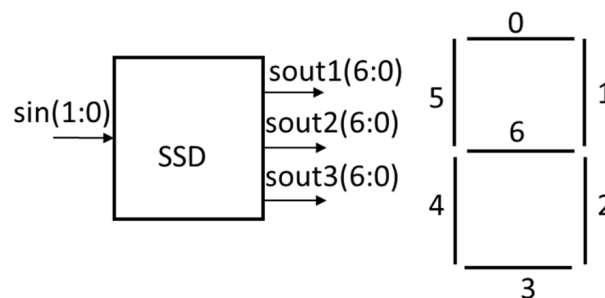
Si en el museo no se detecta movimiento en ninguna de las salas, se supone que el vigilante no está patrullando. Entonces, se pone la señal *vigila* a '1'. En cualquier otro caso, el valor de la señal *vigila* ha de ser '0'.

La señal *disp* toma los siguientes valores:

- La señal `disp` toma el valor "00" si hay que poner la palabra on en los displays. Esto ocurre cuando la señal habilita está a '1'.
- La señal `disp` toma valor "01" si hay que poner la palabra off en los displays. Esto ocurre cuando la señal habilita está a '0'.
- La señal `disp` toma los valores "10" o "11" si los tres displays se han de mantener apagados. Esto no sucede en el comportamiento normal de la alarma.

En la siguiente figura se muestra el display de 7 segmentos, donde se ha numerado cada segmento con números de 0 a 6. El circuito B se emplea para controlar el funcionamiento de los tres displays de siete segmentos. Este circuito tiene una señal de entrada de dos bits llamada `sin` y tres señales de salida de 7 bits llamadas `sout1`, `sout2` y `sout3`.

Para encender un determinado segmento del display  $i$  ( $i = 1,2,3$ ) hay que poner a '1' el componente correspondiente de la señal `souti` ( $i = 1,2,3$ ). Los componentes 6 a 0 de la señal `souti` se corresponden con los segmentos 6 a 0 del display  $i$ . Es decir, un valor de la señal `souti` igual a "0000001" indica que está iluminado el segmento 0 del display  $i$  y que el resto de los segmentos de dicho display están apagados.



**Figura 1.2:** Circuito B y display de siete segmentos.

Los valores que han de tener las señales de salida `sout1`, `sout2` y `sout3` dependen de los valores de la señal de entrada `sin`:

- Si la señal de entrada `sin` tiene el valor "00", el display 1 tiene que estar apagado, en el display 2 se ha de mostrar la o y en el display 3 se ha de mostrar la n.

- Si la señal de entrada `sin` tiene el valor "01", el display 1 tiene que mostrar la o, el display 2 tiene que mostrar la f y el display 3 tiene que mostrar la f.
  - Para cualquier otro valor de la señal `sin`, los tres displays tienen que estar apagados.
- 2.a)** (2 puntos) Obtenga para el circuito A las funciones lógicas de las señales de salida de dicho circuito en función de sus señales de entrada. Dibuje el diagrama de un circuito que implemente las funciones lógicas obtenidas 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. Escriba en VHDL la **entity** y la **architecture** que describe el circuito que ha dibujado anteriormente. Los nombres de los puertos de la **entity** deber ser los mismos que se han especificado para las señales de entrada y salida del circuito.
- 2.b)** (1.5 puntos) Escriba en VHDL la **entity** y la **architecture** que describe el funcionamiento del circuito B empleando un bloque **process** y sentencias **case**. Los nombres de los puertos de la **entity** deber ser los mismos que se han especificado para las señales de entrada y salida del circuito.
- 2.c)** (0.5 puntos) Escriba en VHDL la **entity** y **architecture** que describa el circuito completo. El circuito completo se tiene que describir instanciando y conectando adecuadamente los circuitos A y B diseñados en los apartados 2.a y 2.b, respectivamente. Los nombres de los puertos de la **entity** deber ser los mismos que se han especificado para las señales de entrada y salida del circuito.
- 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 Apartados 2.c.