

# INGENIERÍA DE COMPUTADORES 3

## Trabajo Práctico - Junio 2017

### 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:
  - 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 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.

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 ordinaria de 2017.

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'	'0'
'0'	'0'	'1'	'0'	'1'
'0'	'1'	'0'	'1'	'0'
'0'	'1'	'1'	'0'	'0'
'1'	'0'	'0'	'0'	'0'
'1'	'0'	'1'	'1'	'1'
'1'	'1'	'0'	'0'	'0'
'1'	'1'	'1'	'0'	'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) (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 Apartado 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 pretende diseñar un circuito incrementador del código de Gray. Este circuito combinacional tiene como entrada una palabra de código de Gray de  $N$  bits y su salida es la siguiente palabra del código de Gray de  $N$  bits. El código de Gray de 4 bits ( $g_3g_2g_1g_0$ ), su correspondiente código binario ( $b_3b_2b_1b_0$ ) y el código de Gray incrementado ( $gInc_3gInc_2gInc_1gInc_0$ ) se muestran en la siguiente tabla.

Código binario $b(3:0)$	Código Gray $g(3:0)$	Código Gray incrementado $gInc(3:0)$
0000	0000	0001
0001	0001	0011
0010	0011	0010
0011	0010	0110
0100	0110	0111
0101	0111	0101
0110	0101	0100
0111	0100	1100
1000	1100	1101
1001	1101	1111
1010	1111	1110
1011	1110	1010
1100	1010	1011
1101	1011	1001
1110	1001	1000
1111	1000	0000

- 2.a)** (1 punto) Escriba en VHDL la **entity** y la **architecture** que describa el comportamiento del incrementador de código Gray de 4 bits empleando para ello una asignación concurrente de selección (sentencia **with - select**).
- 2.b)** (3 punto) Escriba en VHDL la **entity** y la **architecture** que describa el comportamiento del incrementador de código Gray de  $N$  bits. El número de bits  $N$  ha de ser una constante de tipo **generic**.

El comportamiento del circuito ha de ser descrito por el siguiente algoritmo:

**Paso 1** Convertir la palabra de código de Gray ( $g_{N-1} \dots g_0$ ) a la correspondiente palabra binaria ( $b_{N-1} \dots b_0$ ). Para ello aplica la fórmula siguiente:

$$b_i = g_i \oplus b_{i+1},$$

siendo  $i = 0, \dots, N - 1$  y  $b_N = 0$ .

**Paso 2** Incrementa en una unidad la palabra binaria.

**Paso 3** Convierte la palabra binaria resultante ( $b_{N-1} \dots b_0$ ) al código de Gray correspondiente ( $g_{N-1} \dots g_0$ ). Para ello aplica la fórmula siguiente:

$$g_i = b_i \oplus b_{i+1},$$

siendo  $i = 0, \dots, N - 1$  y  $b_N = 0$ .

- 2.c)** (2 puntos) Programe en VHDL un banco de pruebas que testee todas las posibles entradas al circuito *incrementador* diseñado en el apartado 2.b cuando la constante de tipo **generic**  $N$  vale 4. 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.

Emplee este banco de pruebas para comprobar el diseño realizado al contestar el Apartado 2.b cuando  $N$  vale 4.

Incluya en la memoria el cronograma obtenido al realizar la simulación del banco de pruebas.