

INGENIERÍA DE COMPUTADORES 3

Trabajo Práctico - Septiembre 2016

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 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 extraordinaria de 2016.

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 (3 PUNTOS)

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

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

- 1.a) (0.25 puntos) Obtenga la función lógica F a partir de la tabla de verdad. Escriba en VHDL la **entity** del circuito que implemente la función lógica. Es decir, que tenga tres entradas x , y y z , y una salida F .
- 1.b) (0.75 puntos) Escriba en VHDL la **architecture** que describa el *comportamiento* del circuito.
- 1.c) (0.25 punto) Dibuje el diagrama de un circuito que implemente esta función lógica 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) (0.75 puntos) 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 para comprobar los circuitos diseñado en los Apartados 1.b y 1.d.

EJERCICIO 2 (7 PUNTOS)

- 2.a)** (0.5 punto) Escriba en VHDL la **entity** y la **architecture** que describe el comportamiento de un codificador de prioridad 4 a 2.

El circuito codificador ha de tener una señal de entrada *r* de 4 bits y dos señales de salida, *codigo* de 2 bits y *activo* de 1 bit. El comportamiento del circuito viene descrito por la siguiente tabla de verdad:

<i>r</i> (3:0)	<i>codigo</i> (1:0)	<i>activo</i>
0000	00	'0'
0001	00	'1'
001 -	01	'1'
01 - -	10	'1'
1 - - -	11	'1'

Realice el diseño del circuito empleando una sentencia **if** para generar la señal *activo* y una sentencia **case** para generar la señal *codigo*.

- 2.b)** (0.5 puntos) Programe en VHDL un banco de pruebas que testee todas las posibles entradas al circuito *codificador* diseñado en el apartado anterior. 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.a. Incluya en la memoria el cronograma obtenido al realizar la simulación del banco de pruebas.

- 2.c)** (0.5 puntos) Escriba en VHDL la **entity** y la **architecture** que describe el comportamiento de un multiplexor 4 a 2 de 2 bits empleando una sentencia **if**.

El circuito multiplexor ha de tener las siguientes señales de entrada: *x0* de 2 bits, *x1* de 2 bits, *x2* de 2 bits, *x3* de 2 bits y una señal de selección *sel* de dos bits. Y ha de tener una señal de salida *smux* de 2 bits.

- 2.d)** (0.5 puntos) Programe en VHDL un banco de pruebas que testee todas las posibles entradas al circuito *multiplexor* diseñado en el apartado anterior. 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.c. Incluya en la memoria el cronograma obtenido al realizar la simulación del banco de pruebas.

- 2.e) (4 puntos) Escriba en VHDL la **architecture** que describe la estructura de un circuito codificador de prioridad de 16 a 4, empleando para ello únicamente codificadores de prioridad 4 a 2 iguales a los diseñados en el Apartado 2.a y el multiplexor diseñado en el Apartado 2.c. Dibuje el diagrama circuital correspondiente a dicho diseño. La tabla de verdad y la **entity** del circuito codificador de prioridad de 16 a 4 se muestran a continuación.

r(15:0)	codigo(3:0)	activo
0000000000000000	0000	'0'
0000000000000001	0000	'1'
0000000000000001 -	0001	'1'
0000000000000001 --	0010	'1'
0000000000000001 ---	0011	'1'
0000000000000001 ----	0100	'1'
0000000000000001 -----	0101	'1'
0000000000000001 -----	0110	'1'
0000000000000001 -----	0111	'1'
0000000000000001 -----	1000	'1'
0000000000000001 -----	1001	'1'
0000000000000001 -----	1010	'1'
0000000000000001 -----	1011	'1'
0000000000000001 -----	1100	'1'
0000000000000001 -----	1101	'1'
0000000000000001 -----	1110	'1'
0000000000000001 -----	1111	'1'

```
entity codificadorP16a4 is
  port ( codigo: out std_logic_vector(3 downto 0);
        activo: out std_logic;
        r : in std_logic_vector(15 downto 0) );
end entity codificadorP16a4;
```

- 2.f) (1 punto) Programe en VHDL un banco de pruebas que testee todas las posibles entradas al circuito *codificador* diseñado en el Apartado 2.e. 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.e. Incluya en la memoria el cronograma obtenido al realizar la simulación del banco de pruebas.