

INGENIERÍA DE COMPUTADORES 3

Trabajo Práctico - Septiembre 2014

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:
 - 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.
 - 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.

Plantearémos 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 2014.

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 x e y mostradas a continuación, que dependen de las tres variables A , B y C :

$$X = A'B'C' + A'BC' + A'BC + AB'C + ABC'$$

$$Y = A'BC' + AB'C' + AB'C$$

- 1.a) (0.5 puntos) Escriba en VHDL la **entity** del circuito que implemente las dos funciones lógicas. Es decir, que tenga tres entradas A , B y C , y dos salidas x e y .
- 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. 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) (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.

EJERCICIO 2

A continuación se describe un algoritmo para realizar la división entre dos enteros ($y = a/b$), donde dividendo (a), divisor (b) y cociente (y) tienen el mismo número de bits. El tamaño de dividendo, divisor y cociente es $n + 1$ bits.

Paso 1 Se declaran dos vectores a_inp y b_inp de tamaño $2 * n + 1$ bits.

Paso 2 Se realizan las dos asignaciones siguientes:

$$a_inp = 0...0a_na_{n-1}...a_1a_0$$

$$b_inp = 0...0b_nb_{n-1}...b_1b_0$$

Paso 3 Se realiza la asignación $i = n$.

Paso 4 Se rota i bits a la izquierda a b_inp y se almacena en b_inpr .

Paso 5 Se compara a_inp con b_inpr . Si $a_inp \geq b_inpr$, entonces $y_i = '1'$ y se almacena en a_inp el resultado de la operación $b_inpr - a_inp$. En caso contrario, se realiza la asignación $y_i = '0'$ y no se modifica el valor de a_inp .

Paso 7 Se actualiza el índice $i = i - 1$.

Paso 6 Si $i \geq 0$, entonces se vuelve al Paso 4. En caso contrario, se finaliza el algoritmo. El valor del cociente está almacenado en y y el valor del resto son los $n + 1$ bits menos significativos de a_inp .

En la tabla siguiente se muestra los resultados obtenidos al aplicar el algoritmo siendo $a = "1011"$ (valor 11 en decimal) y $b = "0011"$ (valor 3 en decimal), para cada valor de i . Como puede observarse, se obtiene $y = "0011"$ (valor 3 en decimal) y el resto $"0010"$ (valor 2 en decimal).

i	a_inp	b_inpr	y_i
3	"0001011"	"0011000"	0
2	"0001011"	"0001100"	0
1	"0001011"	"0000110"	1
0	"0000101" "0000010" (resto)	"0000011"	1

- 2.a)** (3 puntos) Escriba en VHDL la **architecture** que describe el comportamiento del circuito digital combinacional de un divisor de números de $n + 1$ bits siguiendo el algoritmo descrito anteriormente. El circuito ha de tener una señal llamada *err* que se pone a '1' sólo en el caso de que el divisor sea 0.

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

```
entity divider is
    generic(n: integer := 3);
    port ( y: out std_logic_vector(n downto 0);
          rest : out integer range 0 to 2**(n+1)-1;
          err: out std_logic;
          a, b: in integer range 0 to 2**(n+1)-1);
end entity divider;
```

- 2.b)** (3 puntos) Programe en VHDL un banco de pruebas que testee el circuito digital divisor. El banco de pruebas debe testear todas las posibles entradas al circuito combinacional. 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.