

INGENIERÍA DE COMPUTADORES 3

Trabajo Práctico - Convocatoria extraordinaria 2025

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.
- 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 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 los ficheros fuente del código VHDL de los ejercicios que haya realizado.
- La memoria ha de incluir el código VHDL de los ejercicios que haya realizado así como capturas de pantalla de las simulaciones realizadas.
- 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 2025.

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 cuatro variables A, B, C y D:

$$X = A'B'D' + A'BC' + A'BD + AB'D$$

$$Y = A'BD' + AB'D + A'BD$$

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

Se quiere diseñar un circuito secuencial conversor serie-a-paralelo de 8 bits que opera en el flanco de subida de la señal de reloj y tiene una entrada de reset. El circuito tiene la siguiente **entity**:

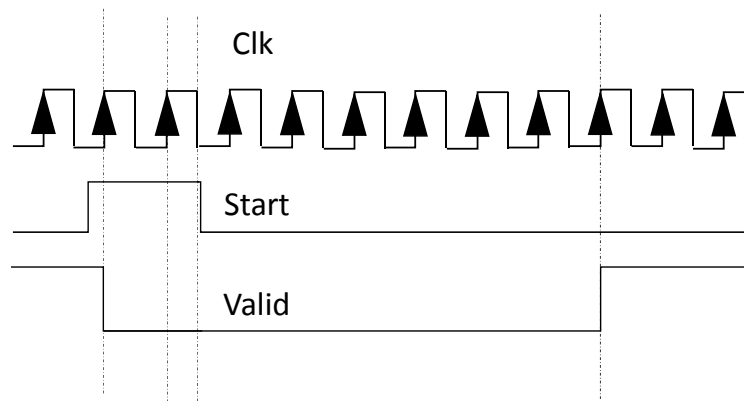
```
entity conversor is
    port( DataOut:  out std_logic_vector( 7 downto 0);
          Valid: out std_logic;
          Clk: in std_logic;
          Reset: in std_logic;
          Start: in std_logic;
          DataIn: in std_logic );
end entity conversor;
```

A continuación se describe el funcionamiento del circuito conversor.

Los bits recibidos por la entrada serie `DataIn` se almacenan en un registro interno del circuito llamado `Content`, tal como se describe a continuación.

En el flanco de subida de la señal de reloj, si la señal `Valid` vale '0' y la señal `Reset` vale '0', se introduce el valor de la señal de entrada serie `DataIn` en la posición más significativa del registro `Content` desplazando el contenido del registro a la derecha. El contenido del registro `Content` es reseteado al valor "00000000" en el flanco de subida de la señal de reloj cuando la señal `Reset` vale '1'. Si la señal `Reset` vale '0' y la señal `Valid` vale '1', se mantiene el valor del registro `Content`.

La señal de salida `Valid` se pone a '0' cuando la señal de entrada `Start` vale '1' en el flanco de subida de la señal de reloj. Si a partir de ese momento se producen 7 flancos consecutivos de subida de la señal de reloj en los que la señal `Start` valga '0', la señal `Valid` pasa a tomar el valor '1' en el séptimo flanco de subida de la señal de reloj. El valor de la señal `Valid` se mantiene a '1' siempre que la señal `Start` valga '0' en los flancos de subida de la señal de reloj. En el siguiente cronograma pueden verse los valores que toma la señal de salida `Valid` en función de los valores que tienen la señal de reloj `Clk` y la señal de entrada `Start`.



El contenido del registro `Content` se carga en la salida paralelo `DataOut` siempre que la señal `Valid` tenga el valor '1' y la señal `Reset` valga '0'. Si la señal `Valid` tiene el valor '0' y la señal `Reset` vale '0', la señal `DataOut` mantiene su valor. La salida `DataOut` se resetea al valor "00000000" cuando la señal `Reset` vale '1'.

- 2.a) (3 puntos) Diseñe en VHDL la **architecture** que describe el comportamiento del circuito secuencial descrito anteriormente.
- 2.b) (3 puntos) Programe en VHDL el banco de pruebas del circuito que ha diseñado en el anterior apartado. La señal de reloj (`Clk`) debe tener un periodo de 20 ns e inicialmente valer '0'. El primer flanco de subida de la señal de reloj se ha de producir en el instante 10 ns. El programa de test debe realizar consecutivamente las acciones siguientes:
 1. *Reset*. La señal de reset ha de tener el valor '1' durante los primeros 15 ns.
 2. *Cargar en la señal de salida paralelo DataOut el valor "11111100"*. La señal `Start` ha de tener el valor '1' únicamente entre los instantes 40 ns y 60 ns, debiendo tener el valor '0' el resto del tiempo. La señal de entrada `DataIn` ha de valer inicialmente '0' y cambiar al valor '1' en el instante 80 ns. En consecuencia, la señal `DataIn` toma en ciclos de reloj consecutivos los valores '0', '0', '1', '1', '1', '1', '1' y '1'. Comprobar que las señales de salida `DataOut` y `Valid` toman los valores correctos. En caso contrario, el banco de pruebas debe mostrar un mensaje indicándolo.