

Escuela Politécnica Superior,
Grado en Informática

**Asignatura: Diseño Automático de
Sistemas**

Práctica 1

7 de febrero de 2022



UNIVERSIDAD
NEBRIJA

Índice/Tabla de contenidos

Índice/Tabla de contenidos	1
1. Introducción	2
1.1. Presentación	2
1.2. Material extra	2
1.3. Objetivo	2
1.4. Funcionamiento	2
1.5. Material provisto	2
1.6. Estilo de código	2
2. Diseño sistema	2
2.1. Organización de ficheros	2
2.2. Top_practica1.vhd File	2
2.3. Synchronizer.vhd	3
2.4. Debouncer.vhd	3
3. Verificación	4
3.1. Ficheros de pruebas (testBench)	4
3.2. Constraints	4
4. Evaluación y entrega	4
4.1. Grupos	4
4.2. Puntuación	4

1. Introducción

1.1. Presentación

En este primer hito del proyecto el objetivo es implementar un debouncer para la placa de desarrollo Nexys 4 DDR.

1.2. Material extra

En la plataforma virtual se encuentran la guía 1 que permite crear un proyecto para la placa Nexys 4 DDR y la guía 2 que es un repaso de la metodología de diseño digital y a como incluir ficheros en el proyecto.

1.3. Objetivo

Hacer lucir un LED mediante el uso de un botón, para que el funcionamiento sea correcto es necesario implementar un debouncer que nos permita utilizar los botones de la placa.

1.4. Funcionamiento

El sistema debe hacer toggle del LED (0), es decir si está apagado encenderlo y viceversa mediante el uso de un botón de pulsación (BTNC). Para poder usar el BTNC es necesario sincronizar la señal e implementar un debouncer.

1.5. Material provisto

En la carpeta correspondiente de la práctica 1 se encuentran los ficheros que contienen la plantilla del código que deberéis completar.

1.6. Estilo de código

El único estilo de código seguido ha sido utilizar el prefijo g_ para los generics y el c_ para las constantes. Los comentarios están en inglés salvo las partes donde se indica que debéis escribir el código.

2. Diseño sistema

El sistema está compuesto de tres ficheros VHDL, los tres son proporcionados y deberán ser completados de acuerdo con las especificaciones. También se proporcionan dos testbench para probar el diseño.

2.1. Organización de ficheros

En la carpeta previo tenéis dos subcarpetas RTL y Sim. Dentro de la carpeta RTL se encuentran los ficheros top_practica1.vhd, debouncer.vhd y synchronizer.vhd. En la carpeta Sim se encuentran dos ficheros: tb_debouncer.vhd y tb_top.vhd.

2.2. Top_practica1.vhd File

Contiene los generic del sistema que son:

Nombre	Tipo	Valor	Notas
g_sys_clock_freq_KHz	Integer	100e3	Frecuencia del sistema (KHz)
g_debounce_time	Integer	20	Tiempo de debounce (ms)
g_reset_value	Std_logic	0	Reset_value para el synchronizer

g_number_flip_flops	Natural	2	Número de flipflops para el synchronizer
---------------------	---------	---	--

En el fichero top se proporciona las definiciones y las instancias de los bloques del proyecto. El alumno debe completar el registro para el estado del LED de salida y la lógica pertinente para que cuando llegue el pulso de 1 ciclo que indica que la pulsación de botón es valida (Toggle_LED) el LED de la salida cambie de estado.

2.3. Synchronizer.vhd

El fichero synchronizer.vhd no requiere de ninguna modificación se trata de un sincronizador como los explicados en clase. La salida de este (BTN_sync) es una señal síncrona con el sistema. Sin embargo, dicha salida fluctuará siguiendo los rebotes del pulsador y por tanto debe ser aplicado el circuito antirrebotes.

2.4. Debouncer.vhd

El fichero debouncer.vhd es donde se va a implementar el antirrebotes. El funcionamiento del antirrebotes es el siguiente: Cuando entra la señal síncrona del botón, "sig_in", se inicia el contador que cuenta hasta el valor deseado en ms. Si transcurrido ese tiempo la señal del botón sigue a 1 la consideramos valida y generamos **un pulso de un ciclo** en la salida "debounced". Una vez la señal del botón cae iniciamos el contador otra vez para volver al estado inicial transcurrido el tiempo de antirrebote. Además, se incluye una entrada de enable "ena" que nos permite desactivar el botón y hace volver al estado inicial desde cualquier estado, reiniciando el contador.

En la Figura 1 se muestra un diagrama orientativo de una posible implementación mediante una máquina de estados de Moore.

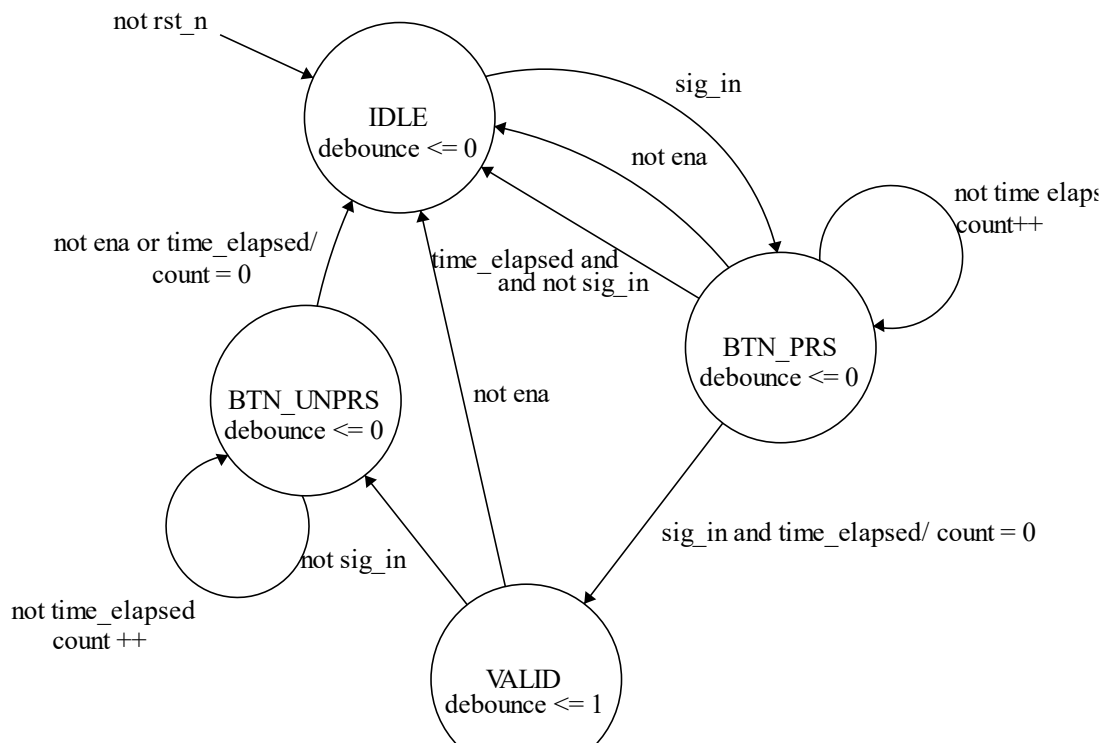


Figura 1. Posible implementación mediante una máquina de Moore.

En el fichero se proporciona una constante “c_cycles” que deberá ser convertida a unsigned mediante el uso de to_unsigned para poder ser comparada con el valor del contador.

3. Verificación

3.1. Ficheros de pruebas (testBench)

Se han proporcionado dos testbench que permiten comprobar el funcionamiento del sistema. Uno sólo para el debouncer y otro para el sistema completo, Se recomienda para las simulaciones reducir el valor del tiempo de antirrebotes para acelerar el proceso.

Los ficheros de testbench pueden ser modificados por el alumno como deseen.

3.2. Constraints

El fichero de constraints debe ser modificado para que funcione el sistema. Se debe descomentar el reloj y aquellas entradas y salidas que sea necesario.

4. Evaluación y entrega

4.1. Grupos

Se formarán grupos de 2 personas y mediante un formulario que se habilitará se comunicará el nombre de las dos parejas y la dirección del repositorio de git.

4.2. Puntuación

Este hito se corresponde con un **20%** del valor del proyecto final. La entrega se realizará tras el último día de prácticas de la asignatura donde se evaluará el proyecto completo, para esta entrega final será necesario adjuntar un pdf documentando el proyecto. El código será revisado por el profesor en la siguiente sesión de prácticas. Para la revisión es necesario llevar el proyecto funcionando y con una visualización de las señales de interés clara que permita ver el funcionamiento del sistema.

