

PRÁCTICA 13
SISTEMAS SECUENCIALES IV
DISEÑO DEL INTERFAZ DE UN TECLADO MATRICIAL

1. Introducción.

En esta práctica continuaremos el aprendizaje del lenguaje VHDL para la descripción de circuitos y sistemas digitales, y del manejo básico de la herramienta “software” VIVADO de Xilinx para el diseño de sistemas digitales con FPGAs.

En concreto, trataremos los sistemas secuenciales basados en una unidad operativa y una unidad de control.

2. Objetivos.

- Aprender a diseñar sistemas secuenciales.

3. Trabajo previo.

- Repasar el diseño de sistemas secuenciales síncronos de control.
- Diseñar en VHDL los circuitos indicados en esta práctica y sus bancos de prueba.

4. Enunciado.

Realizar los siguientes pasos para cada circuito:

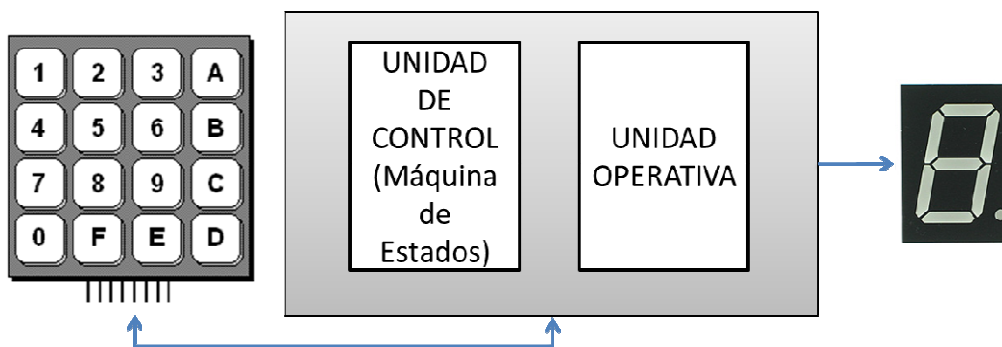
- Hacer un nuevo proyecto en VIVADO para cada circuito. Seleccionar siempre la FPGA de la placa de desarrollo.
- Describir el circuito en VHDL.
- Sintetizarlo para comprobar si la descripción es correcta y sintetizable.
- Leer el informe de síntesis (“synthesis report”).
- Simulación.
 - Realizar el banco de pruebas en VHDL.
 - Realizar la simulación funcional (“behavioral”).
 - Realizar la implementación (“fit”).
 - Leer el informe de implementación (“fitter report”).
 - Realizar la simulación temporal (“post-fit”).

- Enseñar la simulación temporal al profesor.

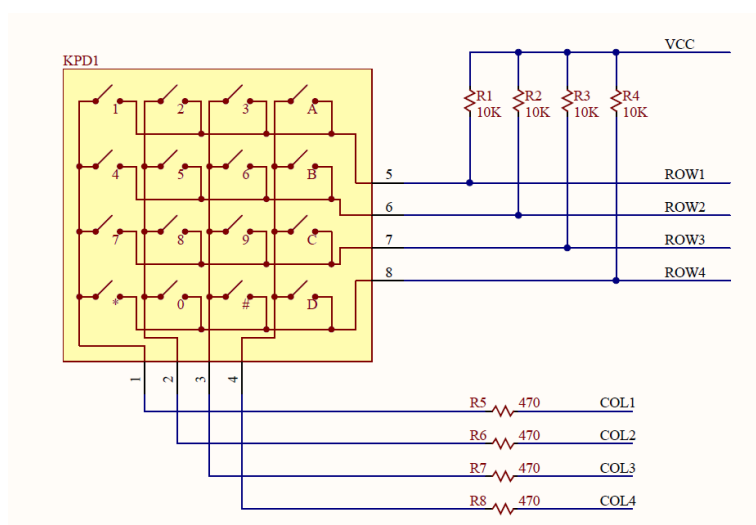
4.1. Ejercicios (10 puntos).

4.1.1. Introducción.

El objetivo de esta práctica es diseñar un sistema de interfaz de un teclado matricial que siga el esquema representado en la figura, basado en una unidad de control y una unidad operativa.



Se utilizará un teclado PMOD-KYPD de Digilent cuyo esquema se representa en la siguiente figura.

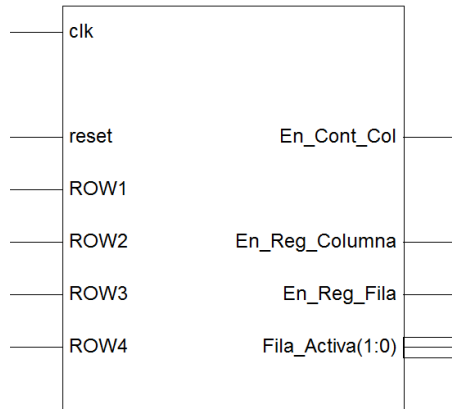


Para explorar el teclado, las columnas (COL1, COL2, COL3 y COL4) se deben activar a nivel bajo de forma secuencial (barrido de columnas). Mediante la lectura de las filas es posible detectar si alguna de las teclas está pulsada (nivel bajo en la fila correspondiente ROW1, ROW2, ROW3 o ROW4).

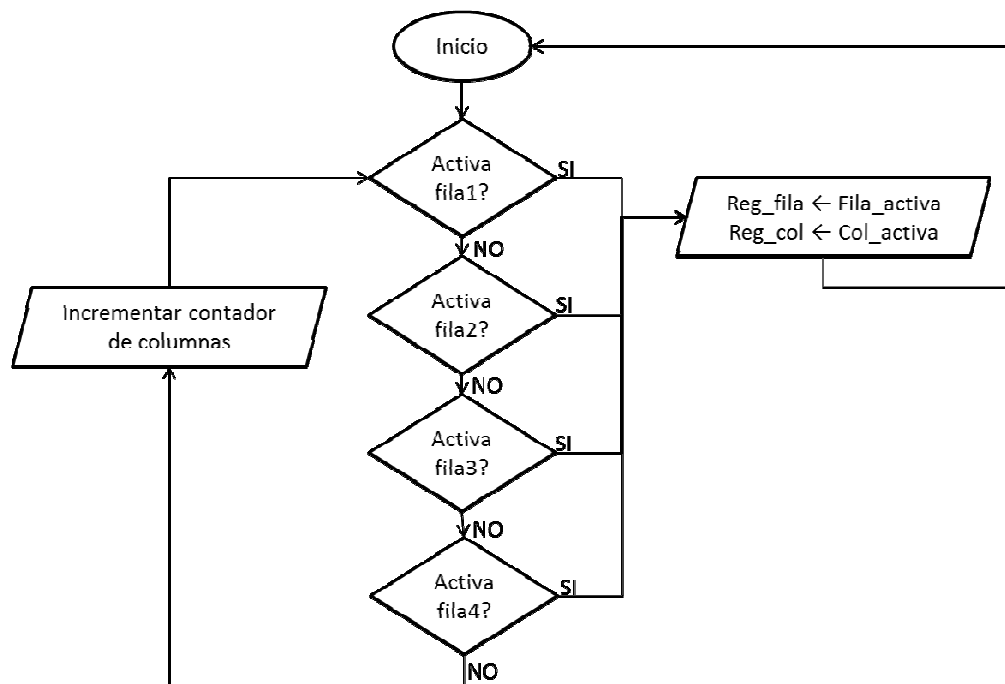
4.1.2. Máquina de estados de la unidad de control (3 puntos).

La unidad operativa es controlada por la unidad de control que debe generar las señales adecuadas para el funcionamiento correcto del sistema. Las señales implicadas en este ejemplo se indican en la figura siguiente.

Maq_Estados



Se debe diseñar en VHDL una máquina de estados que realice la secuencia de operaciones indicada en la figura siguiente. Se deben utilizar 3 procesos para el diseño en VHDL de la máquina de estados.

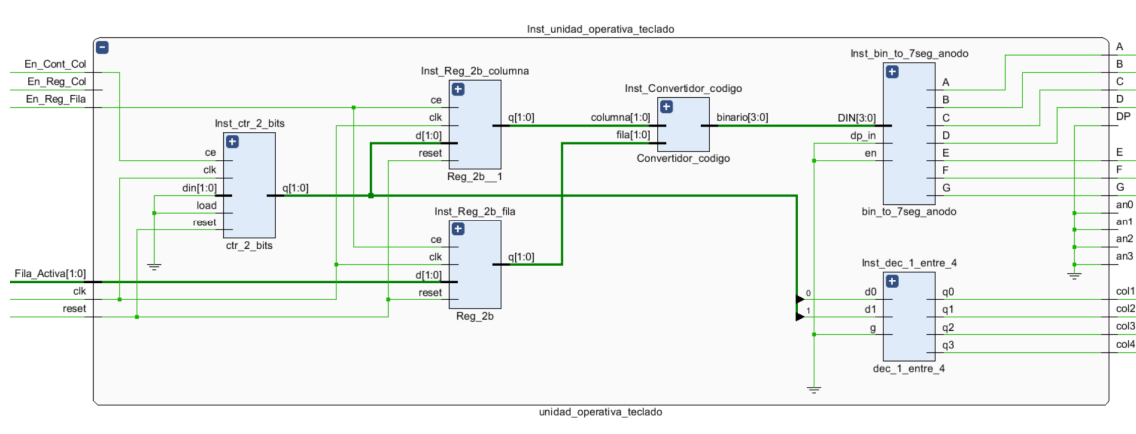


Realizar todas las tareas indicadas en el enunciado, excepto la prueba en la placa.

4.1.3. Unidad operativa (3 puntos).

Práctica 13 Electrónica Digital. Diseño del interfaz de un teclado matricial

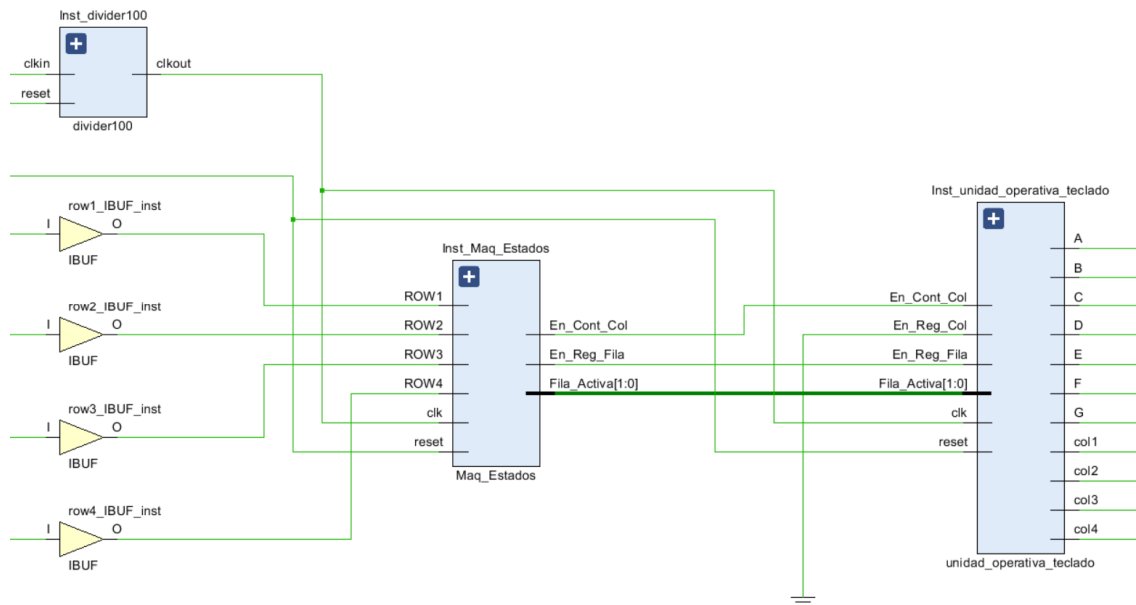
Diseñar en VHDL todos los bloques de la unidad operativa basándose en los circuitos realizados en prácticas anteriores. Diseñar un nuevo fichero VHDL denominado “unidad_operativa.vhd” que conecte todos los bloques, de acuerdo al esquema de la figura siguiente. Realizar todas las tareas indicadas en el enunciado, excepto la prueba en la placa.



- Contador de columnas: su contenido indica en binario la columna seleccionada en cada momento. La unidad de control debe incrementarlo para ir seleccionando una por una todas las columnas del teclado de forma cíclica.
- Decodificador de columnas: convierte el valor binario almacenado en el contador de columnas para activar una de las 4 salidas que se conectan a las columnas del teclado. La activación es a nivel bajo.
- Registro de columna: cuando la unidad de control detecta que se ha pulsado una tecla, debe almacenar en este registro el valor de la columna actual.
- Registro de fila: cuando la unidad de control detecta que se ha pulsado una tecla, debe almacenar en este registro el valor de la fila actual.
- Convertidor de código: a partir del número de la fila y la columna de la tecla pulsada, debe generar el código binario correspondiente a la tecla.
- Decodificador de binario a 7 segmentos: a partir del código binario correspondiente a la tecla pulsada debe generar el código de 7 segmentos necesario para su visualización.

4.1.4. Sistema completo de interfaz de un teclado matricial de 16 teclas PMOD-KYPD conectado a la placa de desarrollo (2 puntos).

Conectar la unidad de control y la unidad operativa en un nuevo fichero VHDL denominado “top_teclado.vhd”, según el diagrama de bloques siguiente.



El circuito “divider1000” será suministrado por el profesor.

Para la prueba de este circuito en la placa, el teclado PMOD-KYPD de Digilent se debe conectar al conector JA de expansión de la placa de desarrollo. Recuerda que el fichero de restricciones distingue entre mayúsculas y minúsculas, y los nombres de las señales en el fichero de restricciones tienen que coincidir exactamente con los puertos del fichero principal en VHDL.

Señal del circuito	Elemento de la placa Basys 3	Señal del circuito	Elemento de la placa Basys 3
clk	clk	an0	an0
RESET	BTNC	A	A
COL4	JA-1	B	B
COL3	JA-2	C	C
COL2	JA-3	D	D
COL1	JA-4	E	E
ROW4	JA-7	F	F
ROW3	JA-8	G	G
ROW2	JA-9	DP	DP

Señal del circuito	Elemento de la placa Basys 3	Señal del circuito	Elemento de la placa Basys 3
ROW1	JA-10		

4.1.5. Informe de implementación del circuito de interfaz de un teclado matricial de 16 teclas PMOD-KYPD (2 puntos).

Se debe rellenar un informe de los resultados de diseño e implementación del ejercicio 3, siguiendo el modelo de informe en Word propuesto por el profesor.

5. Resultados.

- En el laboratorio, el día de las prácticas, se debe enseñar al profesor la simulación temporal de cada uno de los circuitos, una vez implementados sin errores, con el informe de implementación (“fitting”) abierto.
- Asimismo, se deberá enseñar el funcionamiento en la placa de desarrollo de cada uno de los circuitos en los que se pida esta tarea.
- Se debe rellenar un informe de los resultados de diseño e implementación, siguiendo el modelo de informe en Word propuesto por el profesor.
- Antes de la fecha límite de cada práctica, deberán subirse al apartado correspondiente a la práctica, de la carpeta “Ejercicios” de FaiTIC, la siguiente documentación.
 - Informe de los resultados de diseño e implementación de los circuitos realizados. El nombre del archivo debe ser: informe_practica_13.doc (o .docx)”
 - Proyectos Vivado comprimidos correspondientes a cada uno de los circuitos diseñados. El nombre de los archivos debe ser el siguiente:
 - practica_13_ejercicio_1.zip
 - practica_13_ejercicio_2.zip
 - practica_13_ejercicio_3.zip
 - Informe_teclado.doc