

# **Tecnología de Computadores**

## **Práctica 4**



**Universidad  
Rey Juan Carlos**

## 1. Objetivo

En esta cuarta sesión de prácticas se pretende que el alumno se familiarice con dos de los componentes combinacionales más utilizados: el multiplexor y el decodificador.

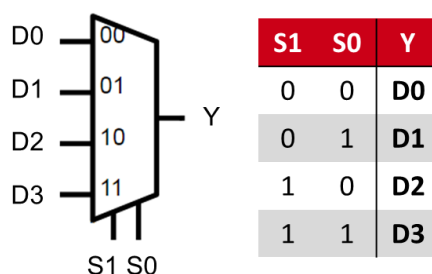
Una vez terminada la práctica el alumno será capaz de:

- Diseñar circuitos combinacionales basados en multiplexores y decodificadores en VHDL
- Implementar diseños con multiplexores y decodificadores en una FPGA

## 2. Desarrollo de la práctica

### El multiplexor.

Como se ha visto en las clases de teoría, el multiplexor es un módulo combinacional que permite seleccionar entre varias entradas. Así, mediante el multiplexado de la señal, podemos conectar circuitos que tienen varias salidas con circuitos que tienen una única entrada. A modo de recordatorio se presenta a continuación la tabla de verdad de un multiplexor de 4 a 1:

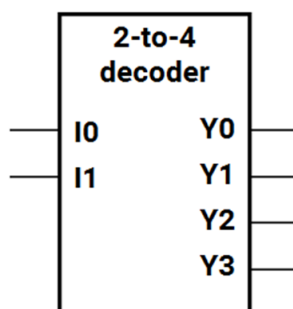


Cree un nuevo proyecto en Vivado y escriba el código VHDL del multiplexor de 4 a 1 de la imagen anterior. Añada a su proyecto el fichero de constraints "mux\_constraints.xdc", genere el bitstream y cargue el diseño en la FPGA. Analice el comportamiento del circuito manipulando los interruptores de la placa hasta que entienda su funcionamiento. Rellene la siguiente tabla con el valor de la salida del circuito:

Interruptores de entrada						LED de salida
J15	L16	M13	R15	U11	V10	H17
0	0	1	1	0	0	
1	0	1	0	1	0	
1	0	1	0	0	1	
0	0	0	1	1	1	

### El decodificador.

Como se ha visto en teoría, un decodificador es otro módulo combinacional básico utilizado ampliamente en circuitos electrónicos digitales para transformar la información. Un ejemplo clásico de uso es el decodificador de binario a decimal como el presentado en la siguiente figura:



Implemente en VHDL el decodificador de 2 a 4 de la figura anterior. Añada a su proyecto el fichero de constraints “*dec\_constraints.xdc*” proporcionado por el profesor, genere el bitstream y cargue el diseño en la FPGA. Analice el comportamiento del circuito manipulando los interruptores de la placa hasta que entienda su funcionamiento. Rellene la siguiente tabla con el valor de la salida del circuito:

Interruptor de entrada		LED de salida			
J15	L16	H17	K15	J13	N14
0	0				
1	0				
0	1				
1	1				

### 3. Ejercicios propuestos

1. Además de como elemento de transmisión de información, un multiplexor se puede utilizar para implementar una función lógica. Esto permite crear un diseño versátil, ya que con el mismo multiplexor se pueden implementar infinidad de funciones lógicas únicamente cambiando los valores de las entradas del multiplexor de las señales de selección. Por ejemplo, dada la siguiente función lógica:

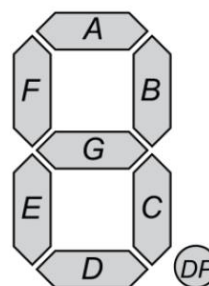
$$Y = D0 \cdot D1 + \overline{D1} \cdot D2 + D1 \cdot \overline{D3} + D0$$

- Implemente la ecuación lógica directamente en VHDL utilizando una arquitectura *Dataflow*.
- Implemente ahora la función lógica en VHDL utilizando el multiplexor de 4 a 1 que diseñó al principio de la práctica. Elija primero las señales de selección y realice un dibujo del circuito que va a diseñar. Luego, utilizando el código VHDL del multiplexor, cree un diseño modular en un nuevo fichero VHDL.
- Compare ambos diseños en términos de utilización de recursos (LUTs e IOBs) y de consumo dinámico.

**Nota:** use el fichero de constraints “*function\_constraints.xdc*” proporcionado por el profesor en ambos diseños.

2. En la práctica se ha visto un ejemplo sencillo de un decodificador de binario a decimal. Existen otros decodificadores como el decodificador de binario a 7 segmentos. Este decodificador permite representar un número binario de entrada en un display de 7 segmentos como el mostrado en la siguiente imagen:

B3 B2 B1 B0	A B C D E F G
0 0 0 0	0 0 0 0 0 1
0 0 0 1	1 0 0 1 1 1 1
0 0 1 0	0 0 1 0 0 1 0
0 0 1 1	0 0 0 0 1 1 0
0 1 0 0	1 0 0 1 1 0 0
0 1 0 1	0 1 0 0 1 0 0
0 1 1 0	0 1 0 0 0 0 0
0 1 1 1	0 0 0 1 1 1 1
1 0 0 0	0 0 0 0 0 0 0
1 0 0 1	0 0 0 0 1 0 0



Utilice la tabla anterior como referencia para diseñar este decodificador en VHDL. Añada a su proyecto el fichero de constraints “*dec7seg\_constraints.xdc*” proporcionado por el profesor, genere el bitstream y cargue el diseño en la FPGA. Analice el comportamiento del circuito manipulando los interruptores de la placa hasta que entienda su funcionamiento. ¿Por qué en la tabla no aparecen más combinaciones de entrada después de la “1001”? ¿Qué comportamiento se podría diseñar para gestionar esas combinaciones?