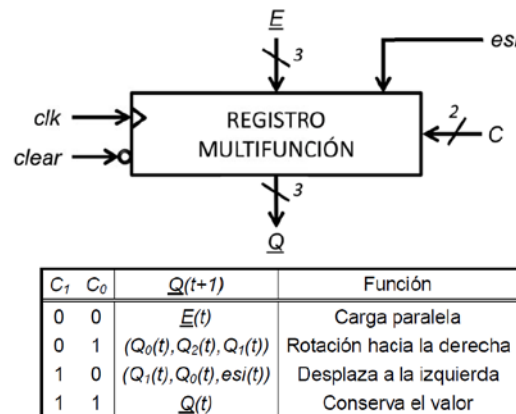




PRÁCTICA 4: DISEÑO CON VHDL Y SIMULACIÓN MEDIANTE MODELSIM DE UN REGISTRO MULTIFUNCIÓN

El objetivo de esta práctica es diseñar en VHDL y simular con Modelsim un registro multifunción como el que se muestra en la figura, capaz de realizar las funciones descritas en la tabla.



El circuito tendrá los puertos siguientes:

- Una salida paralela de datos de 3 bits, \underline{Q} , que muestra el contenido del registro.
- Una entrada paralela de datos de 3 bits, \underline{E} , por la que introducir el valor a cargar.
- Una entrada serie de datos, esi, por la que introducir el bit necesario para el desplazamiento a la izquierda.
- Una entrada de 2 bits, \underline{C} , para seleccionar el modo de funcionamiento.
- Una entrada, clk , de reloj.
- Una entrada, clear, de inicialización asíncrona a (000) activa a baja.

El tipo `bit_vector`.

En esta práctica se va a utilizar el tipo `bit_vector` para describir arrays de bits, como los que se dan, por ejemplo, en los buses de datos o de direcciones de un procesador, o vectores de bits necesarios para codificar números. A continuación vemos algún ejemplo de su uso:

- Declaración de un vector de 8 bits, ordenados en sentido descendente:
 - Signal Bus: `Bit_vector(7 downto 0);`
- asignación del valor '1' a la posición 5 del vector de bits:
 - `Bus(5) <= '1';`
- asignación del valor "01110000" al vector mediante la concatenación de los valores '0' y "1110000"

- `Bus<='0' & "111000";`
- Asignación al bus de los valores "01111111" utilizando la expresión `others`
 - `Bus <= ('0', others => '1');`
- Rotación izquierda de los valores del vector:
 - `Bus <= Bus(6 downto 0) & Bus(7);`
- Asignación a una señal de tipo bit del elemento 3 del vector
 - `A<=Bus(3)`

Vamos a ver un ejemplo de cómo generar las señales de simulación de una señal del tipo `bit_vector(2 downto 0)`. En concreto queremos que el valor inicial sea un "000", a los 50ns tome el valor "001" y a los 150ns tome el valor "010":
`force e 3'b000, 3'b001 50, 3'b 010 150 ns`

Desarrollo de la práctica

A los alumnos se les proporcionan 4 archivos:

- Biestable.vhd
- Multiplexor4a1.vhs
- Registro_mf_plantilla.vhd
- Input_signals.do

Los dos primeros son versiones ligeramente modificadas de los diseños de un biestable y de un multiplexor.

El tercero es una plantilla que deben completar los alumnos con el diseño propuesto en esta práctica. Como se puede observar, a diferencia de lo que se ha hecho en otras prácticas, en ésta los diseños de las componentes que se utilizan en el registro multifunción no están incluidos dentro de la plantilla.

Para realizar la compilación, cuando se cree el proyecto se deben incluir en el mismo los tres ficheros. Para que la compilación del registro multifunción se realice correctamente se deben compilar primero los archivos del biestable y del multiplexor.

El alumno deberá completar el archivo `input_signals.do` de manera que se realicen en orden las siguientes operaciones:

1. Se carga el valor 010 en paralelo
2. Se realiza un desplazamiento a la izquierda poniendo `esi='1'`
3. Se realiza una rotación a la derecha
4. Conserva el valor
5. Se carga el valor en paralelo 011
6. Se realiza una rotación derecha
7. Se realiza un desplazamiento a la izquierda poniendo `esi='0'`
8. Se hace un reset

Cada operación debe durar solo un ciclo de reloj. El resultado de la salida debe ser el que se muestra a continuación:

+ /registro_mf/q	-No Data-	3'b000	3'b010	3'b101	3'b110	3'b011	3'b101	3'b010	3'b000
------------------	-----------	--------	--------	--------	--------	--------	--------	--------	--------