

INSTRUCCIONES:

1. Resuelva este ejercicio en las mismas condiciones en que realizará el examen: dos horas de tiempo y sin emplear ningún material.
2. Revise sus contestaciones, empleando para ello el texto y el simulador que esté usando para estudiar la asignatura.
3. Compare sus respuestas revisadas con la solución.

Pregunta 1 (3 puntos)

Escriba en VHDL la **architecture** que describe el comportamiento de un contador síncrono ascendente módulo 4 en términos de una máquina de Moore. La salida del contador toma cíclicamente los valores “00”, “01”, “10”, “11”. Si la cuenta está habilitada, se obtiene un nuevo valor de la salida en cada flanco de subida de la señal de reloj. La entrada `c` habilita la cuenta cuando está a '1' y la deshabilita cuando está a '0'. La entrada `reset` es una señal asíncrona activa a nivel alto que pone la cuenta a “00”. La **entity** del contador se muestra a continuación.

```
entity contador_Mod_4 is port(  
    state          : out std_logic_vector(1 downto 0);  
    clock, reset, c : in std_logic);  
end entity contador_Mod_4;
```

Pregunta 2 (3 puntos)

Programa en VHDL el banco de pruebas del contador que ha diseñado al resolver la Pregunta 1. El banco de pruebas debe comprobar el funcionamiento del circuito de forma exhaustiva. El banco de pruebas debe generar un conjunto de vectores de test, comprobar si la salida de la UUT es correcta, mostrar un mensaje cada vez que la salida de la UUT no sea correcta y mostrar un mensaje al finalizar el test indicando el número total de salidas incorrectas.

Pregunta 3 (2 puntos)

Escriba en VHDL la **architecture** que describa el comportamiento de un circuito combinacional que realiza operaciones de desplazamiento sobre una entrada de 4 bits. La entrada de selección de operación del circuito es *op*. Las señales de entrada y salida se llaman, respectivamente, *entrada* y *salida*. La tabla de operaciones y la **entity** del circuito son:

<i>op</i>	Operación
0 0	Desplaza 1 bit a la izquierda rellenando con '0'
0 1	Desplaza 1 bit a la derecha rellenando con '1'
1 0	Rota 1 bit a la izquierda
1 1	Rota 1 bit a la derecha

```
entity desplazador is
    port( salida : out std_logic_vector(3 downto 0);
          op      : in  std_logic_vector(1 downto 0);
          entrada : in  std_logic_vector(3 downto 0));
end entity desplazador;
```

Pregunta 4 (2 puntos)

Escriba en VHDL la **architecture** del flip-flop D cuyo comportamiento se describe a continuación, empleando para ello una sentencia **if**.

Además de la entrada de reloj (*Clock*), el flip-flop tiene otras cuatro señales de entrada: *D*, *Enable*, *Set* y *Clear*. El flip-flop tiene una única señal de salida, cuyo valor coincide en todo momento con el valor del estado del circuito (*Q*).

```
entity FFD is
    port( Q           : out std_logic;
          Clock, Enable, Set, Clear, D : in  std_logic);
end entity FFD;
```

Cuando la entrada asíncrona *Set* es puesta al valor '1', el estado del flip-flop cambia inmediatamente al valor '1'. Por el contrario, cuando la entrada asíncrona *Clear* es puesta al valor '1', el estado del flip-flop cambia inmediatamente al valor '0'. La entrada *Set* ha de tener prioridad sobre la entrada *Clear*.

Si la señal *Enable* vale '1', entonces en el flanco de subida de la señal de reloj se asigna el valor de la entrada *D* al estado del flip-flop (*Q*). Por el contrario, si la señal *Enable* vale '0', la carga del flip-flop desde la entrada *D* está deshabilitada.