

Ejercicios VHDL

Ej. 1 — Codificar en VHDL los siguientes elementos hardware:

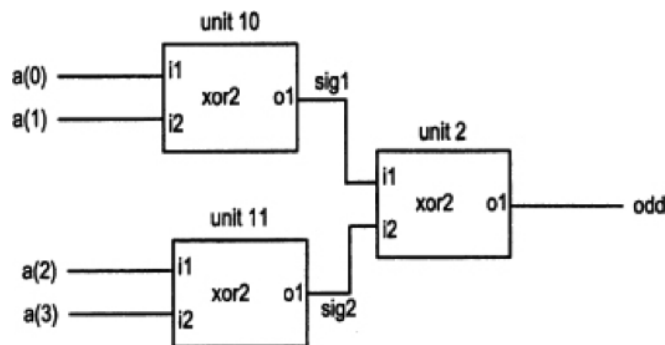
1. XOR de dos números de 4 bits.
2. Multiplexor 4 a 1.
3. Decodificador 3 a 8.
4. Codificador 8 a 3.
5. Comparador de dos números de 4 bits con salida igual a 1 cuando sean iguales.

Ej. 2 — Codificar en VHDL el sistema combinacional con entrada un número entero positivo del 0 al 15, x , y salida z tal que:

$$z = \begin{cases} 1, & \text{si } \begin{cases} x \text{ es primo} \\ x \leq 4 \text{ y par (considerar el 0 como par)} \\ x > 8 \text{ e impar} \end{cases} \\ 0, & \text{c.o.c} \end{cases} \quad (7.1)$$

Ej. 3 — Codificar en VHDL un sistema combinacional que multiplique por 3 el operando de entrada, $op1$, un número entero positivo en el rango 0 a 7. Las salidas son z , un número entero positivo en el rango 0 a 15 (4 bits), y d , puerto de ancho 1 bit que indica si hay desbordamiento.

Ej. 4 — La figura presenta el diagrama de un circuito formado por puertas XOR. Escribir, basado en el diagrama, el código VHDL que describe la arquitectura (architecture) del circuito.



Ej. 5 — El siguiente código VHDL describe de forma estructural un circuito. Dibujar el diagrama de bloques que describe el código.

```
library ieee;
use ieee.std_logic_1164.all;

entity hundred_counter is
  port (
    clk, rst      : in std_logic;
    en            : in std_logic;
```

7. EJERCICIOS VHDL

```
        q_ten, q_one: out std_logic_vector(3 downto 0);
        p_ten      : out std_logic
    );
end hundred_counter;

architecture struct of hundred_counter is
    component dec_counter
        port (
            clk, rst : in std_logic;
            en       : in std_logic;
            q        : out std_logic_vector(3 downto 0);
            pulse    : out std_logic
        );
    end component;
    signal p_one, p_ten: std_logic;

begin
    one_digit: dec_counter
        port map (clk => clk,
                  rst => reset,
                  en  => en,
                  pulse => p_one,
                  q    => q_one);

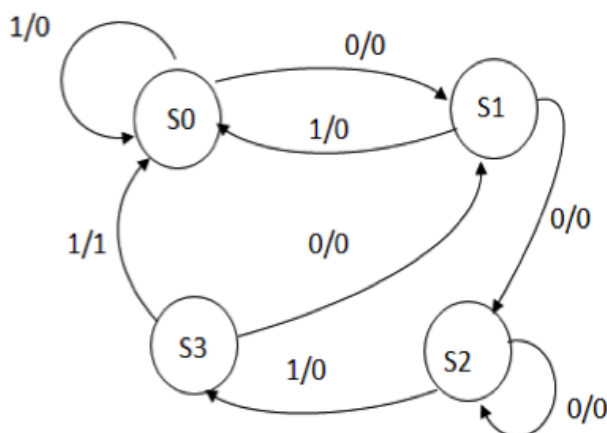
    ten_digit: dec_counter
        port map (clk => clk,
                  rst => rst,
                  en  => p_one,
                  pulse => p_ten,
                  q    => q_ten);
end struct;
```

Ej. 6 — El siguiente código VHDL calcula la paridad de una cadena de bits pero no es sintetizable: nunca podrá ejecutarse en una FPGA. Diseñar una máquina de estados finitos y codificarla en VHDL que calcule la paridad de una cadena de bits.

```
architecture beh of even_detector is
begin
    process(a)
        variable sum, r: integer;
    begin
        sum := 0;
        for i in 2 downto 0 loop
            if a(i)='1' then
                sum := sum + 1;
            end if ;
        end loop ;
        r := sum mod 2;
        if (r=0) then
            even <= '1';
        else
            even <= '0';
        end if;
    end process;
end beh;
```

Ej. 7 — Se desea implementar un registro de desplazamiento a la izquierda de 8 bits con 3 bits de control. Los bits de control indican la cantidad de bits que se tienen que desplazar. Teniendo en cuenta que el desplazamiento se tiene que realizar en un único ciclo de reloj, independientemente de la cantidad de bits a desplazar, diseñar el código VHDL describe la funcionalidad de este circuito.

Ej. 8 — Dado el diagrama de estados del sistema secuencial representado en la figura, escriba el código VHDL que lo describe.



Ej. 9 — Diseñar en VHDL un sistema secuencial para controlar el funcionamiento de un lavaplatos. Las especificaciones del sistemas son:

- Spec 1. La señal de reloj es `clk`.
- Spec 2. La señal de reset, `rst_n`, es activa a nivel bajo.
- Spec 3. El sistema tiene dos entradas: `start/stop` (1 bit), y `ciclo_rapido` (1 bit).
- Spec 4. La entrada `start/stop` sirve para arrancar y parar el lavavajillas.
- Spec 5. La entrada `ciclo_rapido` sirve para indicar el modo de lavado: ciclo rápido o ciclo lento.
- Spec 6. El sistema tiene cinco salidas: `entrar_agua`, `calentar_agua`, `mover_aspas`, `secar` y `abrir_cajetin_detergente`.
- Spec 7. Todas las salidas valen 0 en el estado inicial.
- Spec 8. Desde cualquier estado se va inmediatamente al estado inicial siempre que la tecla `start/stop` vale 0, y allí se permanece hasta que `start/stop` vale 1, cuando comienza a funcionar desde el principio.
- Spec 9. Durante el funcionamiento el aparato pasa por 3 etapas: lavado (2 ó 4 ciclos dependiendo del valor de la entrada `ciclo_rápido`), aclarado (1 ó 2 ciclos dependiendo del valor de la entrada `ciclo_rapido`) y secado (1 ciclo). Después del secado se pasa siempre al estado inicial.
- Spec 10. Durante el lavado entra agua durante el primer ciclo, y durante el mismo se calienta. En el segundo ciclo se abre el cajetín del detergente.
- Spec 11. Las aspas se mueven durante todos los ciclos del lavado.
- Spec 12. Durante el secado se activa la salida `secar`.

Ej. 10 — Diseñar en VHDL un sistema secuencial para controlar el funcionamiento de un túnel de lavado de coches. Las especificaciones del sistemas son:

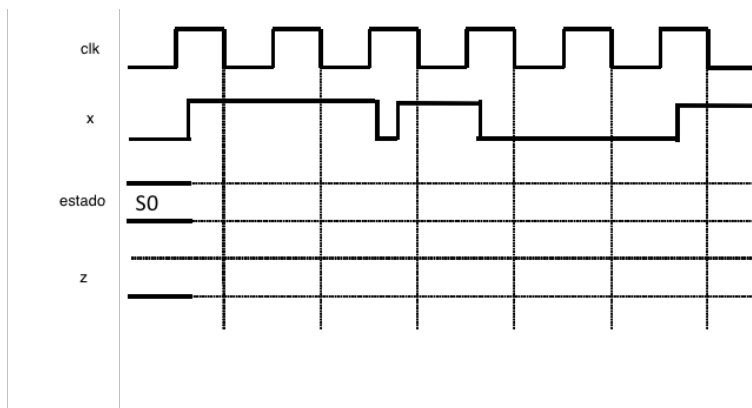
- Spec 1. La señal de reloj es `clk`.
- Spec 2. La señal de reset, `rst_n`, es activa a nivel bajo.
- Spec 3. El sistema tiene dos entradas: `start/stop` (1 bit), y `dar_cera` (1 bit).
- Spec 4. La entrada `start/stop` sirve para arrancar y parar el tunel de lavado.

- Spec 5. La entrada `dar_cera` permite la opción de encerrar el coche.
- Spec 6. El sistema tiene cinco salidas: `dar_agua`, `aire`, `mover_rodillo`, `jabon` y `encerrar`.
- Spec 7. En el estado inicial, todas las salidas valen 0.
- Spec 8. Desde cualquier estado se va inmediatamente al estado inicial siempre que la tecla `start/stop` vale 0, y allí se permanece hasta que `start/stop` vale 1, cuando comienza a funcionar desde el principio.
- Spec 9. Durante el funcionamiento el aparato pasa por 4 etapas: durante 1 ciclo rocía el coche con jabón, luego activa los rodillos durante 2 ciclos, después activa el agua para aclarar (1 ciclo) y el aire para secar (1 ciclo). Si la tecla `dar_cera` está activada, antes de acabar estará 2 ciclos dando cera. En caso contrario vuelve al estado inicial.

Ej. 11 — Diseñar como máquina de Mealy un sistema secuencial con entrada x (1 bit) y salida z (1 bit) tal que:

$$z(n) = \begin{cases} 1, & \text{si } (x(n), x(n-1), x(n-2)) = \begin{cases} 111 \\ 000 \end{cases} \\ 0, & \text{c.o.c} \end{cases} \quad (7.2)$$

Completar el siguiente cronograma y crear un fichero de simulación que se base en dicho cronograma y comparar resultados. Suponer que el primer flanco de reloj ocurre a 100 ns y el periodo de reloj es de 100 ns.



Ej. 12 — El diagrama de estados de la figura representa un reconocedor de patrón. ¿Qué patrón reconoce? Implementar dicho reconocedor en VHDL y completar el cronograma. Crear un fichero de test para simulación a partir de dicho cronograma, suponiendo que el primer flanco ocurre a 100ns y que la frecuencia de reloj es de 5 MHz.

