

# Problemas propuestos

1.1. Realizar un descripción VHDL de las siguientes funciones lógicas, o grupos de funciones lógicas

$$a) F(A, B, C) = AB + A \bar{B} \bar{C}$$

$$b) F(A, B, C, D) = \bar{C} (A + \bar{D}) (\bar{A} + \bar{B} + D)$$

$$c) F(A, B, C) = \overline{A + \bar{B}} (A \oplus \bar{C}) + \bar{B} C + A \bar{C}$$

$$d) F1(A, B, C) = A \bar{B} C + \bar{B} \bar{C} + \bar{A} B C + B \bar{C}$$

$$F2(A, B, C) = A \bar{B} C + B \bar{C}$$

$$F3(A, B, C) = \bar{A} B C + \bar{B} \bar{C}$$

1.2. Una bombilla (B) en un panel de control se enciende si: el sistema (S) está ON y, el modo (M) de funcionamiento es automático, ó bien el modo de funcionamiento es manual y el control (C) está en situación de espera.

Representar este enunciado por una función lógica y su descripción VHDL.

2.1. Describir en lenguaje VHDL los siguientes conjuntos de funciones lógicas:

a)  $F(A, B, C) = \sum(0, 5, 6, 7)$

b)  $F(A, B, C) = \prod(1, 2, 4, 5, 6, 7)$

c)  $F1(A, B, C) = \sum(0, 1, 2, 3, 5)$

$F2(A, B, C) = \sum(2, 3, 5, 6)$

$F3(A, B, C) = \sum(0, 1, 6)$

d)  $F1(A, B, C) = \sum(0, 2, 3, 4, 5) + \sum\emptyset(6)$

$F2(A, B, C) = \sum(5, 6) + \sum\emptyset(1, 2)$

$F3(A, B, C) = \sum(3, 4) + \sum\emptyset(0, 1)$

3.1. Se desea diseñar un circuito lógico para determinar el vencedor de un combate entre dos contendientes X e Y mediante las siguientes especificaciones:

- El combate será a tres toques. El vencedor se declara cuando uno o los dos contendientes llegue a tres toques (se permite la posibilidad de toque simultáneo), o se llegue al llegar al final del tiempo de combate. El número de toques (de 0 a 3) realizado por cada contendiente se almacena en binario en dos variables lógicas para X ( $x_1x_0$ ) y dos para Y ( $y_1y_0$ ).

- Al finalizar el combate se declara vencedor al contendiente que haya realizado más toques. En caso de empate el combate se dilucida por la decisión de un árbitro (variable lógica A) que declara vencedor a X (A a valor 1) ó a Y (A a valor 0). El árbitro no puede declarar el combate empatado.

Realizar una descripción VHDL que permita obtener el ganador del combate. Se deben considerar las siguientes entradas: T que determina si se ha llegado o no al final del tiempo de combate,  $n_X$  y  $n_Y$  que indica el número de toques realizado por cada contendiente y A que contiene la decisión del árbitro. Se deben utilizar dos salidas  $G_x$  y  $G_y$  que indican el vencedor,  $G_x$  y  $G_y$  están a 0 hasta que se determine si ha vencido X ( $G_x$  a 1) ó Y ( $G_y$  a 1) en función del valor de las entradas.

- 4.1. Realizar la descripción VHDL de quiere realizar un circuito de 4 entradas (I4-I1) que muestre como resultado 4 salidas (O4-O1), tal que la salida muestra la entrada pero eliminando todos los unos menos el más significativo. Por ejemplo, si I = "0110" => O = "0100"; si I = "0001" => O = "0001", etc. Si todos los bits de la entrada son 0, los de la salida también: I = "0000" => O = "0000".
- 4.2. En una competición hay cuatro jueces que puntúan con valores enteros entre 0 y 3. Se quiere determinar cuando el promedio de la puntuación es mayor que 1.5. Realizar el código VHDL para la descripción de un circuito digital que resuelva este problema.
- 4.3. Realizar la descripción VHDL de un circuito que realice la suma dos operandos A y B de 5 bits en notación con bits de signo: 4 bits de módulo y un bit de signo (a 0 positivo, a 1 negativo). El resultado Z debe ser de 6 bits: 5 bits de módulo y un bit de signo. Una forma de hacer esto es comprobando si los signos de A y B son iguales, en ese caso el signo de Z es el signo de los operandos y su módulo la suma de los dos módulos. Si los signos son distintos, el signo de Z será el signo del operando de mayor módulo, y el módulo de Z la resta del mayor módulo menos el menor módulo.

- 5.1. Realizar el código VHDL que permita convertir una entrada X de 5 bits que representa un número binario sin signo en dos salidas que representen el número en notación decimal: D de 2 bits que representa las decenas del número y U de 4 bits que representa las unidades del número. Hay que usar sentencias estándar VHDL (case, if-else, etc) sin recurrir a funciones de librerías, aunque sí se pueden usar operadores.
- 5.2. Desarrollar el código VHDL de un circuito que comprueba si en una entrada A de N bits (por defecto N es 4) hay más unos, más ceros o igual número de unos y ceros.
- 5.3. Desarrollar el código VHDL (sin desarrollar la tabla de verdad) de un circuito lógico que permite realizar en su salida Z de 1 bit las operaciones de comparación mayor ( $A > B$ ), igual ( $A = B$ ) o menor ( $A < B$ ) sobre dos entradas de datos A y B de N bits (por defecto N es 4). La operación se selecciona mediante dos entradas de control S1 y S0 con valores 00 (igual), 01 (mayor), 10 (menor), respectivamente, teniendo en cuenta que, cuando S1 y S0 son 1 simultáneamente, la salida Z se debe fijar a 0.

6.1. Un sistema activa una luz artificial L en función de las medidas de cuatro sensores de luz natural que activan o desactivan unas señales lógicas A, B, C y D según el nivel de luz sea menor o mayor de un valor umbral dado, y una señal de control horario H que permite calibrar de dos maneras distintas la influencia de los sensores. En función de los sensores se obtiene un nivel final de luz dado por

$$N = PA \cdot A + PB \cdot B + PC \cdot C + PD \cdot D,$$

donde PA, PB, PC y PD es el peso de cada sensor.

En función de la señal H los pesos PA, PB, PC y PD son respectivamente 3, 5, 7, 2 cuando se usa el primer control horario (H = 0), y son 6, 4, 1, 4 cuando se usa el segundo (H = 1).

L se activa cuando N toma los valores 2, 4, 5, 7, 11 y 17, y se desactiva cuando N toma los valores 3, 6, 8, 9, 12, 13 y 15. Se desea obtener un circuito que implemente  $L = F(H, A, B, C, D)$ .

Realizar una descripción VHDL según el enunciado del problema.

7.1. Dada la siguiente descripción VHDL de una función lógica, encontrar la tabla de verdad en notación decimal para  $Z = F(A,B,C,D_1,D_0)$ . Se utiliza el tipo `std_logic` para poder trabajar con “don't cares”.

```
library ieee;
use ieee.std_logic_1164.all;
entity pr3 is
port (A,B,C: in std_logic;
      D: in std_logic_vector(1 downto 0);
      Z: out std_logic);
end pr3;

architecture descr of pr3 is
begin
process (A,B,C,D)
begin
if ( (A = D(1) and B /= D(0)) or (C /= D(1) and C = D(0) and A = B ) ) then
    Z <= '-';
else
    case D is
        when "00" => Z <= (A and B and C) or ((not A) and (not B) and (not C));
        when "10" => Z <= A or (not C);
        when "11" => if (A = '0') then Z <= not B;
                       else Z <= C;
                       end if;
        when others => Z <= C;
    end case;
end if;
end process;
end descr;
```