

PRÁCTICA 4  
**MANEJO DE UNA TABLA DE VERDAD DENTRO DE UN PROCESS**

OBJETIVO:

El alumno aprenderá el manejo de una tabla de verdad dentro de la instrucción process y verificará que la lista de sensibilidad está dada por las entradas en todo sistema combinacional.

ACTIVIDADES:

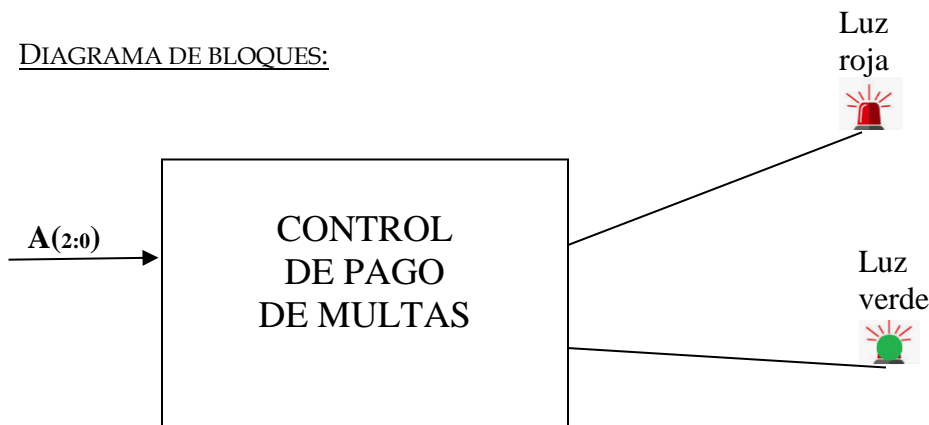
Diseñar el control de pago de multas que maneje dos luces (roja y verde) utilizando la instrucción process en lenguaje VHDL.

ESPECIFICACIONES:

Si la luz roja se prende el infractor paga la multa y si la luz verde se prende no la paga, las luces no se pueden prender simultáneamente. Este sistema tiene por entrada la decisión de un jurado compuesto por tres personas ( $A_2, A_1, A_0$ ) cuyas opiniones valen:  $A_2=35\%$ ,  $A_1=30\%$ ,  $A_0=15\%$ . Un uno lógico a la entrada, significa que ese juez desea que el culpable pague la multa.

La multa se paga cuando el jurado quiere que sea pagada por un 50% o más. La figura 2.1 muestra el diagrama de bloques.

DIAGRAMA DE BLOQUES:



2.1 Diagrama de Bloques del sistema de pago de multas

TABLA DE VERDAD:

ENTRADAS	SALIDAS
A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	L <sub>V</sub> L <sub>R</sub>
000	10
001	10
010	10
011	10
100	10
101	01
110	01
111	01

El código en VHDL de este sistema se observa en la figura 2.2 tomando la entrada como un vector A de tres bits (A<sub>2</sub>, A<sub>1</sub>, A<sub>0</sub>) y la figura 2.3 muestra su simulación.

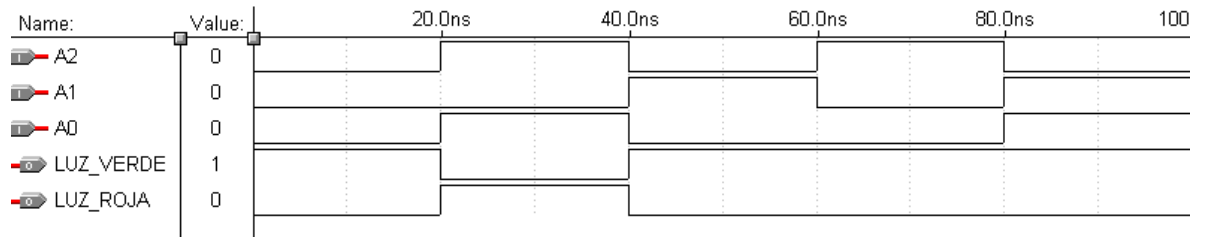
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Pago is
  Port (A : in  STD_LOGIC_VECTOR (2 downto 0);
        LUZ_ROJA : out STD_LOGIC;
        LUZ_VERDE : out STD_LOGIC);
end Pago;

architecture Behavioral of Pago is
begin

P1: Process (A)
begin
  case A is
    when "000" => Luz_Roja <= '0'; Luz_Verde <= '1';
    when "001" => Luz_Roja <= '0'; Luz_Verde <= '1';
    when "010" => Luz_Roja <= '0'; Luz_Verde <= '1';
    when "011" => Luz_Roja <= '0'; Luz_Verde <= '1';
    when "100" => Luz_Roja <= '0'; Luz_Verde <= '1';
    when "101" => Luz_Roja <= '1'; Luz_Verde <= '0';
    when "110" => Luz_Roja <= '1'; Luz_Verde <= '0';
    when others => Luz_Roja <= '1'; Luz_Verde <= '0';
  end case;  end process;  end Behavioral;
```

Código VHDL del sistema de control de pago de multas.



2.3. Simulación del código VHDL del sistema de control de pago de multas.