

PRÁCTICA 5

Diseño de una unidad aritmética lógica (ALU)

OBJETIVO:

El alumno aprenderá el diseño y construcción de una unidad aritmética lógica, que realiza operaciones aritméticas (suma, multiplicación) y operaciones lógicas (and, or).

ACTIVIDADES:

Diseñar una unidad aritmética, lógica.

ESPECIFICACIONES:

Se requiere el diseño y construcción de un sistema digital en el que se visualice en 8 leds, las operaciones aritméticas (suma y multiplicación binarias) y las operaciones lógicas (AND y OR). Tendrá por entradas de datos dos números (A, B) cada uno de cuatro bits, y una entrada de control (C) de 2 bits para selección de 4 distintas operaciones.

DIAGRAMA DE BLOQUES:



TABLA DE VERDAD:

Entradas de control		salida	operación realizada
C_1	C_0	R	
0	0	$A * B$	multiplicación binaria
0	1	$A + B$	suma binaria
1	0	$A \text{ AND } B$	multiplicación lógica
1	1	$A \text{ OR } B$	suma lógica

El código en VHDL de este sistema se observa en la figura 2.2 tomando la entrada como un vector A de cuatro bits (A_3, A_2, A_1, A_0), un vector B de cuatro bits (B_3, B_2, B_1, B_0) y un vector C (C_1, C_0). 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 ALU is
  Port (A, B : in std_logic_vector (3 downto 0);
        C : in std_logic_vector (1 downto 0);
        R : out std_logic_vector (7 downto 0));
end ALU;
architecture Behavioral of ALU is
begin
  with C select
    R <=  A * B when "00",
    ("0000" & A) + ("0000" & B) when "01",
    ("0000" & A) and ("0000" & B) when "10",
    ("0000" & A) or ("0000" & B) when others;
end Behavioral;

```



<https://www.youtube.com/watch?v=ms98GJ0hCZQ>