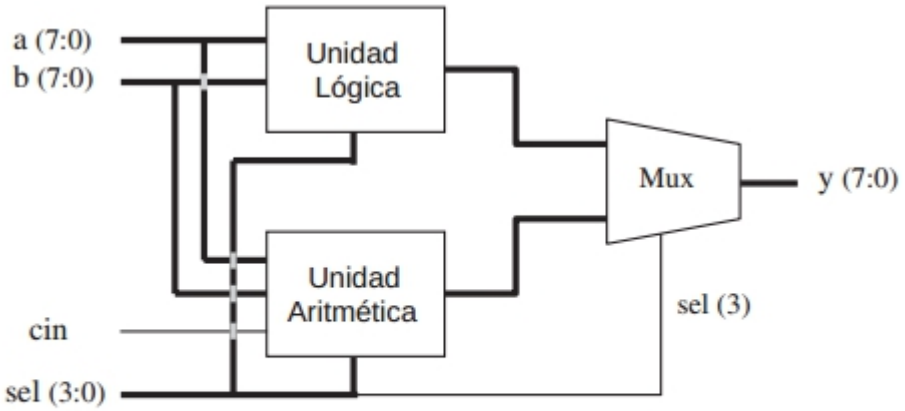
TAREA 1
MICROPROCESADORES

NOMBRE: Fernando Guiraud CÉDULA: 8-945-692

Una ALU (Unidad Lógica Aritmética) se muestra en la figura. Como su nombre lo dice, es un circuito capaz de ejecutar ambos tipos de operaciones, tanto aritméticas como lógicas. Su funcionamiento se describe en la tabla de verdad de la figura. La salida (aritmética o lógico) es seleccionado por el MSB de ***sel***, mientras que la operación específica es seleccionada por los otros tres bits de ***sel***.



sel	Operación	Función	Unidad
0000	y <= a	Transferir a	Aritmético
0001	y <= a+1	Incrementa a	
0010	y <= a-1	Decrementa a	
0011	y <= b	Transferir b	
0100	y <= b+1	Incrementa b	
0101	y <= b-1	Decrementa b	
0110	y <= a+b	Suma a y b	
0111	y <= a+b+cin	Suma a y b con acarreo	
1000	y <= NOT a	Complemento a	Lógico
1001	y <= NOT b	Complemento b	
1010	y <= a AND b	AND	
1011	y <= a OR b	OR	
1100	y <= a NAND b	NAND	
1101	y <= a NOR b	NOR	
1110	y <= a XOR b	XOR	
1111	y <= a XNOR b	XNOR	

Hacerlo por medio de sentencias concurrentes.

Entregables:

- Código
- Simulación

Código:

```
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22  use IEEE.NUMERIC_STD.ALL;
23  use IEEE.std_logic_unsigned.all;
24
25  entity opera is
26      Port ( a : in  STD_LOGIC_VECTOR (7 downto 0);
27             b : in  STD_LOGIC_VECTOR (7 downto 0);
28             sel : in  STD_LOGIC_VECTOR (3 downto 0);
29             cin : in  STD_LOGIC;
30             y : out  STD_LOGIC_VECTOR (7 downto 0));
31  end opera;
32
33  architecture Behavioral of opera is
34
35  begin
36  y <= a WHEN sel="0000" ELSE
37       a+1 WHEN sel="0001" ELSE
38       a-1 WHEN sel="0010" ELSE
39        b WHEN sel="0011" ELSE
40       b+1 WHEN sel="0100" ELSE
41       b-1 WHEN sel="0101" ELSE
42       a+b WHEN sel="0110" ELSE
43       a+b+cin WHEN sel="0111" ELSE
44       NOT a WHEN sel="1000" ELSE
45       NOT b WHEN sel="1001" ELSE
46       a AND b WHEN sel="1010" ELSE
47       a OR b WHEN sel="1011" ELSE
48       a NAND b WHEN sel="1100" ELSE
49       a NOR b WHEN sel="1101" ELSE
50       a XOR b WHEN sel="1110" ELSE
51       a XNOR b;
52  end Behavioral;
```

Código de simulación:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY opera_tb IS
END opera_tb;
ARCHITECTURE behavior OF opera_tb IS
    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT opera
    PORT(
         a : IN  std_logic_vector(7 downto 0);
         b : IN  std_logic_vector(7 downto 0);
         sel : IN  std_logic_vector(3 downto 0);
         cin : IN  std_logic;
         y : OUT  std_logic_vector(7 downto 0)
        );
    END COMPONENT;
   --Inputs
   signal a : std_logic_vector(7 downto 0) := (others => '0');
   signal b : std_logic_vector(7 downto 0) := (others => '0');
   signal sel : std_logic_vector(3 downto 0) := (others => '0');
   signal cin : std_logic := '0';

   --Outputs
   signal y : std_logic_vector(7 downto 0);

   -- Instantiate the Unit Under Test (UUT)
   uut: opera PORT MAP (
          a => a,
          b => b,
          sel => sel,
          cin => cin,
          y => y
        );
   -- Stimulus process
   stim_proc: process
   begin
      -- insert stimulus here
```

```
39  a<= "11111111";
40  b<= "10000000";
41  cin<= '1';
42
43  sel<="0000";
44  wait for 10 ns;
45
46  sel<="0001";
47  wait for 10 ns;
48
49  sel<="0010";
50  wait for 10 ns;
51
52  sel<="0011";
53  wait for 10 ns;
54
55  sel<="0100";
56  wait for 10 ns;
57
58  sel<="0101";
59  wait for 10 ns;
60
61  sel<="0110";
62  wait for 10 ns;
63
64  sel<="0111";
65  wait for 10 ns;
66
67  sel<="1000";
68  wait for 10 ns;
69
70  sel<="1001";
71  wait for 10 ns;
```

```
72
73   sel<="1010";
74   wait for 10 ns;
75
76   sel<="1011";
77   wait for 10 ns;
78
79   sel<="1100";
80   wait for 10 ns;
81
82   sel<="1101";
83   wait for 10 ns;
84
85   sel<="1110";
86   wait for 10 ns;
87
88   sel<="1111";
89   wait for 10 ns;
90
91        wait;
92     end process;
93
94   END;
```

Simulación: