CODIGO

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

entity alu is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           Sel : in STD_LOGIC_VECTOR (2 downto 0);
           Res : out STD_LOGIC_VECTOR (3 downto 0));
end alu;

architecture Behavioral of alu is
--Signals
--SIGNAL Res_temp: std_logic_vector (3 downto 0);
begin
    process(Sel)
    begin
        case (Sel) is
            when "000" => Res <= A + B;
            when "001" => Res <= A - B;
            when "010" => Res <= A - "0001";
            when "011" => Res <= A + "0001";
            when "100" => Res <= (A AND B);
            when "101" => Res <= (A OR B);
            when "110" => Res <= (NOT A);
            when "111" => Res <= (A XOR B);
            when others => null;
        end case;
    end process;
end Behavioral;
```

TEST BENCH

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

entity alu_tb is
--  Port ( );
end alu_tb;

architecture Behavioral of alu_tb is
    COMPONENT alu
        Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
               B : in STD_LOGIC_VECTOR (3 downto 0);
               Sel : in STD_LOGIC_VECTOR (2 downto 0);
               Res : out STD_LOGIC_VECTOR (3 downto 0));
    END COMPONENT;
    --Signals
    SIGNAL A:   STD_LOGIC_VECTOR (3 downto 0);
    SIGNAL B:   STD_LOGIC_VECTOR (3 downto 0);
    SIGNAL Sel: STD_LOGIC_VECTOR (2 downto 0);
    SIGNAL Res: STD_LOGIC_VECTOR (3 downto 0);
begin

    -- Instance
    DUT: alu
        PORT MAP(
            A => A,
            B => B,
            Sel => Sel,
            Res => Res
        );

     --Signals
     estimulos: process
     begin
        --A + B
        A   <= "1000";
        B   <= "0100";
        Sel <= "000";
        wait for 10 ns;
        --A - B
        A   <= "1000";
        B   <= "0100";
        Sel <= "001";
        --A - 1
        wait for 10 ns;
        A   <= "0101";
        Sel <= "010";
        wait for 10 ns;
        --A + 1
        A   <= "1100";
        Sel <= "011";
        wait for 10 ns;
        -- A AND B
        A   <= "0101";
        B   <= "0110";
        Sel <= "100";
        wait for 10 ns;
        --A OR B
        A   <= "0101";
        B   <= "0110";
        Sel <= "101";
        -- NOT A
        wait for 10 ns;
        A   <= "0000";
        Sel <= "110";
        wait for 10 ns;
        -- A XOR B
        A   <= "0101";
        B   <= "0011";
        Sel <= "111";
        wait for 10 ns;

    end process;

end Behavioral;
```