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


entity ALU is
    Port ( A : in       STD_LOGIC_VECTOR (7 downto 0);
           B : in       STD_LOGIC_VECTOR (7 downto 0);
           F : out      STD_LOGIC_VECTOR (7 downto 0);
           Cin : in     STD_LOGIC;
           OPCODE : in  STD_LOGIC_VECTOR (3 downto 0);
           Cout : out   STD_LOGIC);
end ALU;

architecture Behavioral of ALU is

    -- signals
    signal A_add_Cin:   STD_LOGIC_VECTOR (7 downto 0);
    signal A_add_B:     STD_LOGIC_VECTOR (7 downto 0);
    signal Sub:         STD_LOGIC_VECTOR (7 downto 0);
    signal R_SHIFT:     STD_LOGIC_VECTOR (7 downto 0);
    signal L_SHIFT:     STD_LOGIC_VECTOR (7 downto 0);

    signal Carry:       STD_LOGIC_VECTOR (2 downto 0);

    COMPONENT ADD8
    PORT(
        A : IN      std_logic_vector(7 downto 0);
        B : IN      std_logic_vector(7 downto 0);
        Cin : IN    std_logic;
        Sum : OUT   std_logic_vector(7 downto 0);
        Cout : OUT  std_logic
        );
    END COMPONENT;
begin

    -- A + Cin
    Inst_ADD8_0: ADD8 PORT MAP(
        A => A,
        B => "00000000",
        Cin => Cin,
        Sum => A_add_Cin,
        Cout => Carry(0)
    );

    -- A + B + Cin
    Inst_ADD8_1: ADD8 PORT MAP(
        A => A,
        B => B,
        Cin => Cin,
        Sum => A_add_B,
        Cout => Carry(1)
    );

    -- A + (not B) + Cin
    Inst_ADD8_2: ADD8 PORT MAP(
        A => A,
        B => (not B),
        Cin => Cin,
        Sum => Sub,
        Cout => Carry(2)
    );

    -- shift to right
    R: for i in 0 to 6 generate
        R_SHIFT(i) <= A(i+1);
    end generate;
        R_SHIFT(7) <= '0';

    -- shift to left
    L: for i in 1 to 7 generate
        L_SHIFT(i) <= A(i-1);
    end generate;
        L_SHIFT(0) <= '0';


    with OPCODE (3 downto 0) select
        F <=    A_add_Cin   when "0000",
                A_add_B     when "0001",
                Sub         when "0010",
                A-1       when "0011",
                A and B     when "0100",
                A or B      when "0101",
                A xor B     when "0110",
                not A       when "0111",
```

```vhdl
                R_SHIFT     when "1000",
                R_SHIFT     when "1001",
                R_SHIFT     when "1010",
                R_SHIFT     when "1011",
                L_SHIFT     when "1100",
                L_SHIFT     when "1101",
                L_SHIFT     when "1110",
                L_SHIFT     when "1111",
                "ZZZZZZZZ"  when others;

    with OPCODE (3 downto 0) select
        Cout <= Carry(0) when "0000",
                Carry(1) when "0001",
                Carry(2) when "0010",
                '0'      when others;

end Behavioral;
```