```vhdl
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY ALU IS
    PORT (
    ALUOP  :  IN STD_LOGIC_VECTOR(2  DOWNTO 0);-- := "111";
    DATA1  :  IN STD_LOGIC_VECTOR(31 DOWNTO 0);-- := "10000011100000000111000000000011";
    DATA2  :  IN STD_LOGIC_VECTOR(31 DOWNTO 0);-- := "00000001100100000001100000000001";
    RESULT : OUT STD_LOGIC_VECTOR(31 DOWNTO 0) := "00000000000000000000000000000000";
    ZERO   : OUT STD_LOGIC := '0';
    CARRY  : OUT STD_LOGIC := '0';
    OVER   : OUT STD_LOGIC := '0');
END ALU;

ARCHITECTURE STRUCTURAL OF ALU IS
    SIGNAL AND_BUFFER  : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL SRL_BUFFER  : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL SRA_BUFFER  : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL SLL_BUFFER  : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL OOR_BUFFER  : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL ADD_BUFFER  : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL SUB_BUFFER  : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL DATA2_1COMP : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL CLK         : STD_LOGIC := '0';
    SIGNAL CARRY_ADD   : STD_LOGIC := '0';
    SIGNAL CARRY_SUB   : STD_LOGIC := '0';
    SIGNAL ZERO_ADD    : STD_LOGIC := '0';
    SIGNAL ZERO_SUB    : STD_LOGIC := '0';
    SIGNAL OVER_SUB    : STD_LOGIC := '0';
    SIGNAL OVER_ADD    : STD_LOGIC := '0';
BEGIN

  DATA2_1COMP <= NOT DATA2;
 --
  SHIFT_LEFT_LOGICAL :
    ENTITY WORK.GEN_SLEFT(BEHAVIORAL)
    PORT MAP(DATA1_IN => DATA1, RESULT => SLL_BUFFER);
 --
  SHIFT_RIGHT_ARITHMATIC :
    ENTITY WORK.GEN_ARIGHT(BEHAVIORAL)
    PORT MAP(DATA1_IN => DATA1, RESULT => SRA_BUFFER);
 --
  SHIFT_RIGHT_LOGICAL :
    ENTITY WORK.GEN_RLEFT(BEHAVIORAL)
    PORT MAP(DATA1_IN => DATA1, RESULT => SRL_BUFFER);
 --
  ANDDER :
    ENTITY WORK.GEN_AND(BEHAVIORAL)
    PORT MAP(DATA1_IN => DATA1, DATA2_IN => DATA2, RESULT => AND_BUFFER);
 --
  ORRER :
    ENTITY WORK.GEN_OR(BEHAVIORAL)
    PORT MAP(DATA1_IN => DATA1, DATA2_IN => DATA2, RESULT => OOR_BUFFER);
 --
  ADDER :
    ENTITY WORK.GEN_ADD_SUB(BEHAVIORAL)
    PORT MAP(DATA1_IN => DATA1, DATA2_IN => DATA2, RESULT => ADD_BUFFER,
             C => CARRY_ADD, OP => '0', Z => ZERO_ADD, V => OVER_ADD);
 --
  SUBBER :
    ENTITY WORK.GEN_ADD_SUB(BEHAVIORAL)
    PORT MAP(DATA1_IN => DATA1, DATA2_IN => DATA2_1COMP, RESULT => SUB_BUFFER,
             C => CARRY_SUB, OP => '1', Z => ZERO_SUB, V => OVER_SUB);
 --
  ALUMUX :
    ENTITY WORK.ALUMUX(STRUCTURAL)
    PORT MAP (
    ALUOP_IN       => ALUOP,
    AND_BUFFER_IN  => AND_BUFFER,
```

```
70        SRL_BUFFER_IN  => SRL_BUFFER,
71        SRA_BUFFER_IN  => SRA_BUFFER,
72        SLL_BUFFER_IN  => SLL_BUFFER,
73        OOR_BUFFER_IN  => OOR_BUFFER,
74        ADD_BUFFER_IN  => ADD_BUFFER,
75        SUB_BUFFER_IN  => SUB_BUFFER,
76        CARRY_ADD_IN   => CARRY_ADD,
77        CARRY_SUB_IN   => CARRY_SUB,
78        ZERO_ADD_IN    => ZERO_ADD,
79        ZERO_SUB_IN    => ZERO_SUB,
80        OVER_SUB_IN    => OVER_SUB,
81        OVER_ADD_IN    => OVER_ADD,
82        CLOCK_IN       => CLK,
83        RESULT_MUX     => RESULT,
84        ZERO_MUX       => ZERO,
85        CARRY_MUX      => CARRY,
86        OVER_MUX       => OVER);
87     --
88        CLOCK:
89          PROCESS
90            BEGIN
91              WHILE 1 = 1 LOOP
92                CLK <= NOT CLK;
93                WAIT FOR 0.5 NS;
94              END LOOP;
95        END PROCESS CLOCK;
96    END STRUCTURAL;
```