```
----------------------------------------------------------------------------------
-- Company:
-- Engineer: Brandon Frazier
--
-- Create Date: 09/19/2024 02:33:08 PM
-- Design Name:
-- Module Name: ALU - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.numeric_std.all;

entity ALU is
    generic(N : integer := 16);
    Port ( A: in std_logic_vector(N-1 downto 0);
           B: in std_logic_vector(N-1 downto 0);
           -- 000: +
           -- 001: -
           -- 010: negate (-A)
           -- 011: SLL
           -- 100: AND
           -- 101: |
           -- 110: XOR
           -- 111: not A
           Mode: in std_logic_vector(0 to 2);
           OE: in std_logic := '0';
           Zero: out std_logic := '0';
           Cout: out std_logic := '0';
           C: out std_logic_vector(N-1 downto 0)
);
end ALU;

architecture behavioral of ALU is

begin

    process(A, B, Mode)
        variable sumAns: signed(N downto 0) := (others => '0');
        variable sumCalc1,sumCalc2 : signed(N downto 0);
        variable ans : signed(N-1 downto 0);

        begin
                if(OE = '1') then
                case Mode is
                    -- addition
                    when "000" =>
                      sumCalc1 := resize(signed(A),N+1);                    --
                      resize to N+1 long
                      sumCalc2 := resize(signed(B),N+1);                    --
                      resize to N+1 long
                      sumAns := sumCalc1 + sumCalc2;

                      C <= std_logic_vector(sumAns(N-1 downto 0));
                      Cout <= sumAns(N);

                    -- Subtraction
```

```
68                      when "001" =>
69                        sumCalc1 := resize(signed(A),N+1);                    --
                          resize to N+1 long
70                       sumCalc2 := resize(signed(B),N+1);                    --
                         resize to N+1 long
71                        sumAns := sumCalc1 - sumCalc2;
72                        C <= std_logic_vector(sumAns(N-1 downto 0));
73                        Cout <= sumAns(N);
74
75                      -- -A
76                       when "010" =>
77                        ans := NOT signed(A);
78                        ans := ans + 1;
79                        Cout <= '0';
80                        C <= std_logic_vector(ans(N-1 downto 0));
81
82                      -- sll
83                       when "011" =>
84                        ans := signed(A) SLL 1;
85                        Cout <= ans(N-1) XOR A(N-1);
86                        C  <= std_logic_vector(ans(N-1 downto 0));
87
88                      -- &
89                       when "100" =>
90                        ans := signed(A) AND signed(B);
91                        Cout <= ans(0);
92                        C <= std_logic_vector(ans);
93
94                      -- |
95                       when "101" =>
96                        ans := signed(A) OR signed(B);
97                        Cout <= ans(0);
98                        C <= std_logic_vector(ans);
99
100                     -- XOR
101                      when "110" =>
102                       ans := signed(A) XOR signed(B);
103                       Cout <= ans(0);
104                       C <= std_logic_vector(ans);
105
106
107                     -- NOT(A)
108                     when "111" =>
109                       ans := NOT signed(A);
110
111                       Cout <= ans(0);
112                       C <= std_logic_vector(ans);
113                     when others =>
114
115                 end case;
116                   if(ans = 0) then
117                         Zero <= '1';
118                       else
119                         Zero <= '0';
120                       end if;
121                   else
122                     C <= "ZZZZZZZZZZZZZZZ";
123                 end if;
124         end process;
125
126 end behavioral;
127
```