```vhdl
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity negationTB is
end entity;

architecture behavior of negationTB is
  constant TIME_DELAY : time := 20 ns;
  constant NUM_VALS : integer := 3;


  type A_array is array(0 to (NUM_VALS - 1)) of std_logic_vector(15 downto 0);
  type B_array is array(0 to (NUM_VALS - 1)) of std_logic_vector(15 downto 0);
  type C_array is array(0 to (NUM_VALS - 1)) of std_logic_vector(15 downto 0);
  type mode_array is array(0 to (NUM_VALS - 1)) of std_logic_vector(2 downto 0);
  type Zero_array is array(0 to (NUM_VALS - 1)) of std_logic;
  type OE_array is array(0 to (NUM_VALS - 1)) of std_logic;
  type Cout_array is array(0 to (NUM_VALS - 1)) of std_logic;

  -- Expected input and output data.
  -- full zeros
  -- positive negation
  -- negative negation

  constant A_vals : A_array := (B"0000_0000_0000_0000",
                                B"0000_0000_0001_0000",
                                B"1111_1111_1111_0000");

  constant B_vals : B_array := (B"0000_0000_0000_0000",
                                B"0000_0001_0001_0000",
                                B"1000_0101_0000_0000");



  constant mode_vals : mode_array := (B"010",
                                      B"010",
                                      B"010");

  constant Zero_vals : Zero_array := ('1','0','0');

  constant OE_vals : OE_array := ('1','1','1');

  constant Cout_vals : Cout_array := ('0','0','0');

  constant C_vals : C_array := (B"0000_0000_0000_0000",
                                B"1111_1111_1111_0000",
                                B"0000_0000_0001_0000");


  signal A_sig : std_logic_vector(15 downto 0);
  signal B_sig : std_logic_vector(15 downto 0);
  signal C_sig : std_logic_vector(15 downto 0);
  signal mode_sig : std_logic_vector(2 downto 0);
  signal Zero_sig : std_logic;
  signal OE_sig : std_logic;
  signal Cout_sig : std_logic;

begin

  DUT : entity work.ALU(behavioral)
    port map(A => A_sig,
             B => B_sig,
             C => C_sig,
             Mode => mode_sig,
             Zero => Zero_sig,
             OE => OE_sig,
             Cout => Cout_sig);

```

```
70      stimulus : process
71      begin
72        for i in 0 to (NUM_VALS - 1) loop
73          A_sig <= A_vals(i);
74          B_sig <= B_vals(i);
75          C_sig <= C_vals(i);
76          mode_sig <= mode_vals(i);
77          OE_sig <= OE_vals(i);
78          wait for TIME_DELAY;
79        end loop;
80        wait;
81      end process stimulus;
82
83      monitor : process
84        variable i : integer := 0;
85      begin
86        wait for TIME_DELAY/4;
87        while (i < NUM_VALS) loop
88          assert C_sig = C_vals(i)
89            report "C value is incorrect."
90            severity error;
91
92          assert Zero_sig = Zero_vals(i)
93            report "Zero value is incorrect."
94            severity error;
95
96          wait for TIME_DELAY/2;
97
98          assert Cout_sig = Cout_vals(i)
99            report "Cout value is incorrect."
100           severity error;
101
102         i := i + 1;
103         wait for TIME_DELAY/2;
104       end loop;
105       wait;
106     end process monitor;
107
108   end behavior;
109
```