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

entity shiftTB is
end entity;

architecture behavior of shiftTB is
  constant TIME_DELAY : time := 20 ns;
  constant NUM_VALS : integer := 4;


  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 shift
  -- negative shift
  -- sign has changed

  constant A_vals : A_array := (B"0000_0000_0000_0000",
                                B"0000_0000_0001_0000",
                                B"1111_1111_1111_0000",
                                B"1000_0000_0000_0000");


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



  constant mode_vals : mode_array := (B"011",
                                      B"011",
                                      B"011",
                                      B"011");

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

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

  constant Cout_vals : Cout_array := ('0','0','0','1' );

  constant C_vals : C_array := (B"0000_0000_0000_0000",
                                B"0000_0000_0010_0000",
                                B"1111_1111_1110_0000",
                                B"0000_0000_0000_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,
```

```
70                B => B_sig,
71                C => C_sig,
72                Mode => mode_sig,
73                Zero => Zero_sig,
74                OE => OE_sig,
75                Cout => Cout_sig);
76
77    stimulus : process
78    begin
79      for i in 0 to (NUM_VALS - 1) loop
80        A_sig <= A_vals(i);
81        B_sig <= B_vals(i);
82        C_sig <= C_vals(i);
83        mode_sig <= mode_vals(i);
84        OE_sig <= OE_vals(i);
85        wait for TIME_DELAY;
86      end loop;
87      wait;
88    end process stimulus;
89
90    monitor : process
91      variable i : integer := 0;
92    begin
93      wait for TIME_DELAY/4;
94      while (i < NUM_VALS) loop
95        assert C_sig = C_vals(i)
96          report "C value is incorrect."
97          severity error;
98
99        assert Zero_sig = Zero_vals(i)
100         report "Zero value is incorrect."
101         severity error;
102
103       wait for TIME_DELAY/2;
104
105       assert Cout_sig = Cout_vals(i)
106         report "Cout value is incorrect."
107         severity error;
108
109       i := i + 1;
110       wait for TIME_DELAY/2;
111     end loop;
112     wait;
113   end process monitor;
114
115 end behavior;
116
```