```
library IEEE;
     use IEEE.std logic 1164.all;
 3
     use IEEE.numeric std.all;
 4
 5
     entity andTB is
 6
     end entity;
 7
8
     architecture behavior of andTB is
9
       constant TIME DELAY : time := 20 ns;
10
       constant NUM VALS : integer := 5;
11
12
13
       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);
14
       type C array is array(0 to (NUM VALS - 1)) of std logic vector(15 downto 0);
15
16
       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;
17
18
       type OE array is array(0 to (NUM VALS - 1)) of std logic;
19
       type Cout array is array(0 to (NUM VALS - 1)) of std logic;
20
21
       -- positive and negative
22
       -- negative and positive
23
       -- two full zeros
24
       -- one that results in a zero
25
       -- result is odd
26
       constant A_vals : A_array := (B"0000 0000 0001 0000",
27
28
                                      B"1111_1111_1111_1111",
29
                                      B"0000_0000_0000_0000",
                                      B"1111_1111_1111_1111",
30
                                      B"1111_1111_1111, 11111");
31
32
33
       constant B vals : B array := (B"1111 1111 1111 0100",
34
                                      B"0000 0000 0000 1000",
35
                                      B"0000 0000 0000 0000",
36
                                      B"0000 0000 0000 0000",
37
38
                                      B"0000 0000 0000 0011");
39
40
41
42
       constant mode vals : mode array := (B"100",
                                            B"100",
43
                                            B"100",
44
                                            B"100",
45
46
                                            B"100");
47
48
       constant Zero vals : Zero array := ('0','0','1','1','0');
49
50
       constant OE_vals : OE_array := ('1','1','1','1','1');
51
52
       constant Cout vals : Cout array := ('0','0','0','0','1');
53
54
       constant C_vals : C_array := (B"0000 0000 0001 0000",
                                      B"0000 0000 0001 0000",
55
                                      B"0000 0000 0000 0000",
56
57
                                      B"0000 0000 0000 0000",
58
                                      B"0000 0000 0000 0011");
59
60
61
62
       signal A sig : std logic vector(15 downto 0);
63
       signal B sig : std logic vector(15 downto 0);
64
       signal C sig : std logic vector(15 downto 0);
65
       signal mode sig : std logic vector(2 downto 0);
66
       signal Zero sig : std logic;
67
       signal OE_sig : std_logic;
68
       signal Cout sig : std logic;
69
```

```
70
     begin
 71
 72
        DUT : entity work.ALU (behavioral)
 73
          port map(A => A sig,
 74
                   B => B_sig,
 75
                    C \Rightarrow C_{sig}
 76
                   Mode => mode sig,
                    Zero => Zero_sig,
 77
 78
                    OE => OE sig,
 79
                    Cout => Cout sig);
 80
 81
        stimulus : process
 82
        begin
 83
          for i in 0 to (NUM VALS - 1) loop
 84
            A sig \leq A vals(\overline{i});
 85
            B_sig <= B_vals(i);</pre>
 86
            --C_sig <= C_vals(i);
 87
            mode sig <= mode vals(i);</pre>
 88
            OE sig <= OE vals(i);
 89
            wait for TIME DELAY;
 90
          end loop;
 91
          wait;
 92
        end process stimulus;
 93
 94
        monitor : process
 95
          variable i : integer := 0;
 96
        begin
 97
          wait for TIME_DELAY/4;
 98
          while (i < NUM_VALS) loop</pre>
 99
            assert C sig = C vals(i)
100
              report "C value is incorrect."
101
              severity error;
102
103
            assert Zero sig = Zero vals(i)
104
              report "Zero value is incorrect."
105
              severity error;
106
107
            wait for TIME DELAY/2;
108
109
            assert Cout sig = Cout vals(i)
110
              report "Cout value is incorrect."
111
              severity error;
112
113
            i := i + 1;
114
            wait for TIME DELAY/2;
115
          end loop;
116
          wait;
117
        end process monitor;
118
119
     end behavior;
120
```