

```

1  library IEEE;
2  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);
14     type B_array is array(0 to (NUM_VALS - 1)) of std_logic_vector(15 downto 0);
15     type C_array is array(0 to (NUM_VALS - 1)) of std_logic_vector(15 downto 0);
16     type mode_array is array(0 to (NUM_VALS - 1)) of std_logic_vector(2 downto 0);
17     type Zero_array is array(0 to (NUM_VALS - 1)) of std_logic;
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
27     constant A_vals : A_array := (B"0000_0000_0001_0000",
28                                     B"1111_1111_1111_1111",
29                                     B"0000_0000_0000_0000",
30                                     B"1111_1111_1111_1111",
31                                     B"1111_1111_1111_1111");
32
33
34     constant B_vals : B_array := (B"1111_1111_1111_0100",
35                                     B"0000_0000_0000_1000",
36                                     B"0000_0000_0000_0000",
37                                     B"0000_0000_0000_0000",
38                                     B"0000_0000_0000_0011");
39
40
41
42     constant mode_vals : mode_array := (B"100",
43                                           B"100",
44                                           B"100",
45                                           B"100",
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",
55                                     B"0000_0000_0001_0000",
56                                     B"0000_0000_0000_0000",
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 => C_sig,
76                 Mode => mode_sig,
77                 Zero => Zero_sig,
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 <= A_vals(i);
85             B_sig <= B_vals(i);
86             --C_sig <= C_vals(i);
87             mode_sig <= mode_vals(i);
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
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

```