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

entity xorTB is
end entity;

architecture behavior of xorTB is
  constant TIME_DELAY : time := 20 ns;
  constant NUM_VALS : integer := 5;


  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;

  -- positive and negative
  -- negative and positive
  -- two full zeros
  -- result is zero
  -- result is odd

  constant A_vals : A_array := (B"0000_0000_0001_0000",
                                B"1111_1111_1111_1111",
                                B"0000_0000_0000_0000",
                                B"1111_1111_1111_1111",
                                B"1111_1111_1111_1111");


  constant B_vals : B_array := (B"1111_1111_1111_0100",
                                B"0000_0000_0000_1000",
                                B"0000_0000_0000_0000",
                                B"1111_1111_1111_1111",
                                B"1111_1111_1111_1110");



  constant mode_vals : mode_array := (B"110",
                                      B"110",
                                      B"110",
                                      B"110",
                                      B"110");

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

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

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

  constant C_vals : C_array := (B"1111_1111_1110_0100",
                                B"1111_1111_1111_0111",
                                B"0000_0000_0000_0000",
                                B"0000_0000_0000_0000",
                                B"0000_0000_0000_0001");



  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;
```

```vhdl
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
```