ECEN302 Lab 3 - Functions, Procedures, and Test Benches

Joshua Benfell - 300433229

August 18, 2020

1 Objectives

2 Methodology

2.1 Introduction

2.2 Challenges

2.3 Code

2.3.1 1-1 Procedures Add two Values

```
----------------------------------------------------------------------------------
-- Module Name: add_two_values_procedure
----------------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

library UNISIM;
use UNISIM.VComponents.all;

Entity add_two_values_procedure Is Port (
    Signal ain : in STD_LOGIC_VECTOR (3 downto 0);
    Signal bin : in STD_LOGIC_VECTOR (3 downto 0);
    Signal sum : out STD_LOGIC_VECTOR (3 downto 0);
    Signal cout : out STD_LOGIC
);
end add_two_values_procedure;

Architecture behavior of add_two_values_procedure Is

    Signal total_out : STD_LOGIC_VECTOR (4 downto 0);

    procedure add_two_values (
        Signal ain_int : in STD_LOGIC_VECTOR (3 downto 0);
        Signal bin_int : in STD_LOGIC_VECTOR (3 downto 0);
        Signal total_out_int : out STD_LOGIC_VECTOR (4 downto 0)
    ) is

```

```
    begin
        total_out_int(4 downto 0) <= ('0' & ain_int) + ('0' & bin_int);
    end add_two_values;

begin
    cout <= total_out(4);
    sum <= total_out(3 downto 0);

    process (ain, bin)
    begin
          add_two_values (ain, bin, total_out);
    end process;
end behavior;
```

code/lab311.vhd

```
----------------------------------------------------
-- Module Name: add_two_values_procedure_tb
----------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

use STD.textio.all;
use IEEE.std_logic_textio.all;

library UNISIM;
use UNISIM.VComponents.all;

Entity add_two_values_procedure_tb Is
end add_two_values_procedure_tb;

Architecture behavior of add_two_values_procedure_tb Is
    Component add_two_values_procedure
    port(
        Signal ain : in STD_LOGIC_VECTOR (3 downto 0);
        Signal bin : in STD_LOGIC_VECTOR (3 downto 0);
        Signal sum : out STD_LOGIC_VECTOR (3 downto 0);
        Signal cout : out STD_LOGIC
    );
    End Component;

    Signal count : STD_LOGIC_VECTOR (2 downto 0) := "000";

    Signal ain : STD_LOGIC_VECTOR (3 downto 0) := "0000";
    Signal bin : STD_LOGIC_VECTOR (3 downto 0) := "0000";
    Signal sum : STD_LOGIC_VECTOR (3 downto 0) := "0000";
    Signal cout : STD_LOGIC := '0';

begin
    DUT:  add_two_values_procedure PORT MAP (
            ain => ain,
            bin => bin,
            sum => sum,
            cout => cout
         );

    process
```

```
45          variable k : integer := 0;
46          --variable s : line;
47
48      begin
49
50          ain <= "0110"; bin <= "1010";
51          --write (s, "ain = ");  write (s, ain); write (s, "bin = "); write
                (s, bin); write (s, "cout = "); write (s, cout); write (s, "sum =
                "); write (s, sum);
52          --writeline (output, s);
53
54          for k in 0 to 4 loop
55
56            wait for 5 ns;
57            count <= count + "1";
58            ain <= ain + count;
59            bin <= bin + count;
60            --write (s, "ain = ");  write (s, ain); write (s, "bin = "); write
                  (s, bin); write (s, "cout = "); write (s, cout); write (s, "sum
                  = "); write (s, sum);
61            --writeline (output, s);
62
63          end loop;
64
65          wait;
66
67          --write (s, "Simulation Ends");
68          --writeline (output, s);
69
70      end process;
71  end behavior;
```

code/lab311_tb.vhd

2.3.2 1-2 Calc Even Parity

```
1  ----------------------------------------------------------------------------------
2  -- Module Name: calc_even_parity_procedure
3  ----------------------------------------------------------------------------------
4
5  library IEEE;
6  use IEEE.STD_LOGIC_1164.ALL;
7  use IEEE.STD_LOGIC_ARITH.ALL;
8  use IEEE.STD_LOGIC_UNSIGNED.ALL;
9
10 library UNISIM;
11 use UNISIM.VComponents.all;
12
13 Entity calc_even_parity_procedure  Is Port (
14      Signal ain : in STD_LOGIC_VECTOR (7 downto 0);
15      Signal parity : out STD_LOGIC
16 );
17 end calc_even_parity_procedure ;
18
19
20
21 Architecture behavior of calc_even_parity_procedure  Is
22
23      procedure calc_even_parity (
24          signal input : in STD_LOGIC_VECTOR (7 downto 0);
```

```
        signal output : out std_logic
    ) is
    variable par_bit : STD_LOGIC := '0';
    begin
        for k in 0 to input'length-1 loop
            par_bit := par_bit xor ain(k);
        end loop;
        output <= par_bit;
    end calc_even_parity;

begin

    calc_even_parity(ain, parity);

end behavior;
```

code/lab312.vhd

```
----------------------------------------------------
-- Module Name: calc_even_parity_procedure_tb
----------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

use STD.textio.all;
use IEEE.std_logic_textio.all;

library UNISIM;
use UNISIM.VComponents.all;

Entity calc_even_parity_procedure_tb Is
end calc_even_parity_procedure_tb;

Architecture behavior of calc_even_parity_procedure_tb Is
    Component calc_even_parity_procedure
    port(
        Signal ain : in STD_LOGIC_VECTOR (7 downto 0);
        Signal parity : out STD_LOGIC
    );
    End Component;

    Signal count : STD_LOGIC_VECTOR (2 downto 0) := "000";

    Signal ain : STD_LOGIC_VECTOR (7 downto 0);
    Signal parity : STD_LOGIC;

begin
    DUT:  calc_even_parity_procedure PORT MAP (
            ain => ain,
            parity => parity
         );

    process
        variable k : integer := 0;
        --variable s : line;

    begin
```

```
43
44          ain <= "10101000";
45          --write (s, "ain = ");  write (s, ain); write (s, "parity = ");
                write (s, parity);
46          --writeline (output, s);
47
48          for k in 0 to 4 loop
49
50            wait for 5 ns;
51            count <= count + "1";
52            ain <= ain + count;
53            --write (s, "ain = ");  write (s, ain); write (s, "parity = ");
                  write (s, parity);
54            --writeline (output, s);
55
56          end loop;
57
58          wait;
59
60          --write (s, "Simulation Ends");
61          --writeline (output, s);
62
63      end process;
64  end behavior;
```

code/lab312_tb.vhd

2.3.3 2-1 Functions Add Two Values

```
1  ----------------------------------------------------------------------------------
2  -- Module Name: add_two_values_function
3  ----------------------------------------------------------------------------------
4
5  library IEEE;
6  use IEEE.STD_LOGIC_1164.ALL;
7  use IEEE.STD_LOGIC_ARITH.ALL;
8  use IEEE.STD_LOGIC_UNSIGNED.ALL;
9
10 library UNISIM;
11 use UNISIM.VComponents.all;
12
13 Entity add_two_values_function  Is Port (
14      Signal ain : in STD_LOGIC_VECTOR (3 downto 0);
15      Signal bin : in STD_LOGIC_VECTOR (3 downto 0);
16      Signal sum : out STD_LOGIC_VECTOR (4 downto 0)
17 );
18 end add_two_values_function ;
19
20 Architecture behavior of add_two_values_function  Is
21
22      signal ain_int : STD_LOGIC_VECTOR(4 downto 0);
23      signal bin_int : STD_LOGIC_VECTOR(4 downto 0);
24
25      function add_two_values  (signal ain, bin : in STD_LOGIC_VECTOR)
26      return std_logic_vector is
27          variable result : STD_LOGIC_VECTOR(4 downto 0);
28      begin
29          result := ain + bin;
30          return result;
31      end add_two_values;
```

```
32
33 begin
34     ain_int <= "0" & ain;
35     bin_int <= "0" & bin;
36
37     process (ain_int, bin_int) begin
38         sum <= add_two_values(ain_int, bin_int);
39     end process;
40 end behavior;
```

code/lab321.vhd

```
1  ----------------------------------------------
2  -- Module Name: add_two_values_function_tb
3  ----------------------------------------------
4
5  library IEEE;
6  use IEEE.STD_LOGIC_1164.ALL;
7  use IEEE.STD_LOGIC_ARITH.ALL;
8  use IEEE.STD_LOGIC_UNSIGNED.ALL;
9
10 use STD.textio.all;
11 use IEEE.std_logic_textio.all;
12
13 library UNISIM;
14 use UNISIM.VComponents.all;
15
16 Entity add_two_values_function_tb Is
17 end add_two_values_function_tb;
18
19 Architecture behavior of add_two_values_function_tb Is
20     Component add_two_values_function
21     port(
22         Signal ain : in STD_LOGIC_VECTOR (3 downto 0);
23         Signal bin : in STD_LOGIC_VECTOR (3 downto 0);
24         Signal sum : out STD_LOGIC_VECTOR (4 downto 0)
25     );
26     End Component;
27
28     Signal count : STD_LOGIC_VECTOR (2 downto 0) := "000";
29
30     Signal ain : STD_LOGIC_VECTOR (3 downto 0);
31     Signal bin : STD_LOGIC_VECTOR (3 downto 0);
32     Signal sum : STD_LOGIC_VECTOR (4 downto 0);
33
34 begin
35     DUT:  add_two_values_function PORT MAP (
36               ain => ain,
37               bin => bin,
38               sum => sum
39           );
40
41     process
42         variable k : integer := 0;
43         --variable s : line;
44
45     begin
46
47         ain <= "0110"; bin <= "1010";
```

```
48          --write (s, "ain = ");  write (s, ain); write (s, "bin = "); write
                (s, bin); write (s, "sum = "); write (s, sum);
49          --writeline (output, s);
50
51          for k in 0 to 4 loop
52
53            wait for 5 ns;
54            count <= count + "1";
55            ain <= ain + count;
56            bin <= bin + count;
57            --write (s, "ain = ");  write (s, ain); write (s, "bin = "); write
                 (s, bin); write (s, "sum = "); write (s, sum);
58            --writeline (output, s);
59
60          end loop;
61
62          wait;
63
64          --write (s, "Simulation Ends");
65          --writeline (output, s);
66
67      end process;
68  end behavior;
```

code/lab321_tb.vhd

2.3.4 2-2 Calc Ones

```
1  ----------------------------------------------------------------------------------
2  -- Module Name: calc_ones_function
3  ----------------------------------------------------------------------------------
4
5  library IEEE;
6  use IEEE.STD_LOGIC_1164.ALL;
7  use IEEE.STD_LOGIC_ARITH.ALL;
8  use IEEE.STD_LOGIC_UNSIGNED.ALL;
9
10 library UNISIM;
11 use UNISIM.VComponents.all;
12
13 Entity calc_ones_function  Is Port (
14     Signal ain : in STD_LOGIC_VECTOR (7 downto 0);
15     Signal number_of_ones : out STD_LOGIC_VECTOR (2 downto 0)
16 );
17 end calc_ones_function ;
18
19 Architecture behavior of calc_ones_function  Is
20
21     function count_ones(
22         signal input : std_logic_vector (7 downto 0)
23     )
24     return std_logic_vector is
25        variable count : std_logic_vector (2 downto 0) := "000";
26     begin
27        for k in 0 to input'length-1 loop
28            if (input(k) = '1') then
29               count := count + 1;
30            end if;
31        end loop;
32     return count;
```

```
    end count_ones;


begin
    number_of_ones <= count_ones(ain);


end behavior;
```

code/lab322.vhd

```
--------------------------------------------------
-- Module Name: calc_ones_function_tb
--------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

use STD.textio.all;
use IEEE.std_logic_textio.all;

library UNISIM;
use UNISIM.VComponents.all;

Entity calc_ones_function_tb Is
end calc_ones_function_tb;

Architecture behavior of calc_ones_function_tb Is
    Component calc_ones_function
    port(
        Signal ain : in STD_LOGIC_VECTOR (7 downto 0);
        Signal number_of_ones : out STD_LOGIC_VECTOR (2 downto 0)
    );
    End Component;

    Signal count : STD_LOGIC_VECTOR (2 downto 0) := "000";

    Signal ain : STD_LOGIC_VECTOR (7 downto 0);
    Signal number_of_ones : STD_LOGIC_VECTOR (2 downto 0);

begin
    DUT:  calc_ones_function PORT MAP (
            ain => ain,
            number_of_ones => number_of_ones
         );

    process
        variable k : integer := 0;
        --variable s : line;

    begin

        ain <= "01001010";
        wait for 5 ns;
        --write (s, "ain = ");  write (s, ain); write (s, "number of ones =
            "); write (s, number_of_ones);
        --writeline (output, s);

```

```
        for k in 0 to 4 loop

          wait for 5 ns;
          count <= count + "1";
          ain <= ain + count;
          --write (s, "ain = ");  write (s, ain); write (s, "number of ones
             = "); write (s, number_of_ones);
          --writeline (output, s);

        end loop;

        wait;

        --write (s, "Simulation Ends");
        --writeline (output, s);

    end process;
end behavior;
```

code/lab322_tb.vhd

2.3.5 3-1 RCA Test Bench

```
--------------------------------------------------
-- Module Name: rca_dataflow_tb
--------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

use STD.textio.all;
use IEEE.std_logic_textio.all;

library UNISIM;
use UNISIM.VComponents.all;

Entity rca_dataflow_tb Is
end rca_dataflow_tb;

Architecture behavior of rca_dataflow_tb Is
    Component rca_dataflow
    port(
        a    : in STD_LOGIC_VECTOR (3 downto 0);
        b    : in STD_LOGIC_VECTOR (3 downto 0);
        cin  : in STD_LOGIC;
        s    : out STD_LOGIC_VECTOR (3 downto 0);
        cout : out STD_LOGIC
    );
    End Component;

    Signal count : STD_LOGIC_VECTOR (2 downto 0) := "000";
    Signal cout_expected: STD_LOGIC;

    Signal a : STD_LOGIC_VECTOR (3 downto 0);
    Signal b : STD_LOGIC_VECTOR (3 downto 0);
    Signal c : STD_LOGIC;
    Signal test_failed : STD_LOGIC;
    Signal cout_compare1 : STD_LOGIC;
```

```
38      Signal s_compare1 : STD_LOGIC;
39      Signal s_compare2 : STD_LOGIC;
40      Signal s_int : STD_LOGIC_VECTOR (3 downto 0);
41      Signal cout : STD_LOGIC;
42
43      procedure add_two_values (
44          a_in : in std_logic_vector(3 downto 0);
45          b_in : in std_logic_vector(3 downto 0);
46          c_in : in std_logic;
47
48          sum : out std_logic_vector(4 downto 0)
49      ) is
50
51      begin
52          sum := ("0" & a_in) + ("0" & b_in) + ("0" & c_in);
53      end add_two_values;
54
55  begin
56      DUT:  rca_dataflow PORT MAP (
57              a => a,
58              b => b,
59              cin => c,
60              s => s_int,
61              cout => cout
62           );
63
64      process
65          variable k : integer := 0;
66          variable s : line;
67          variable sum_out : STD_LOGIC_VECTOR (4 downto 0);
68
69      begin
70
71          a <= "0100"; b <= "1010"; c <= '0'; count <= "000"; test_failed <=
                '0';
72          wait for 10 ns;
73
74          add_two_values(a, b, c, sum_out);
75          cout_expected <= sum_out(4);
76
77          wait for 10 ns;
78          write (s, string'("a = "));
79          write (s, a);
80          write (s,  string'("b = "));
81          write (s, b);
82          write (s, string'(" cin = "));
83          write (s, c);
84          write (s, string'(" sum expected = "));
85          write (s, sum_out);
86          write (s, string'(" cout expected = "));
87          write (s, cout_expected);
88          write (s, string'(" actual sum = "));
89          write (s, s_int);
90          write (s, string'(" actual cout = "));
91          write (s, cout);
92          writeline (output, s);
93
94          cout_compare1 <= cout_expected XOR cout;
95          s_compare2 <= (sum_out(3) XOR s_int(3)) OR (sum_out(2) XOR s_int(2))
```

```
        OR (sum_out(1) XOR s_int(1)) OR (sum_out(0) XOR s_int(0)) ;
96
97      if (cout_compare1 = '1') or (s_compare2 = '1') then
98          test_failed <= '1';
99      end if;
100
101     for k in 1 to 4 loop
102       count <= count + "1";
103
104       wait for 10 ns;
105       a <= a + count; b <=  b - count;
106       add_two_values(a, b, c, sum_out);
107       cout_expected <= sum_out(4);
108
109       wait for 10 ns;
110         write (s, string'("a = "));
111         write (s, a);
112         write (s,  string'("b = "));
113         write (s, b);
114         write (s, string'(" cin = "));
115         write (s, c);
116         write (s, string'(" sum expected = "));
117         write (s, sum_out);
118         write (s, string'(" cout expected = "));
119         write (s, cout_expected);
120         write (s, string'(" actual sum = "));
121         write (s, s_int);
122         write (s, string'(" actual cout = "));
123         write (s, cout);
124         writeline (output, s);
125
126       cout_compare1 <= cout_expected XOR cout;
127       s_compare1 <= (sum_out(3) XOR s_int(3)) OR (sum_out(2) XOR
              s_int(2)) OR (sum_out(1) XOR s_int(1)) OR (sum_out(0) XOR
              s_int(0));
128
129       if (cout_compare1 = '1') or (s_compare2 = '1') then
130         test_failed <= '1';
131       end if;
132     end loop;
133
134
135     a <= "1000"; b <= "0011"; c <= '1'; count <= "000";
136     wait for 10 ns;
137
138     add_two_values(a, b, c, sum_out);
139     cout_expected <= sum_out(4);
140
141     wait for 10 ns;
142     write (s, string'("a = "));
143     write (s, a);
144     write (s,  string'("b = "));
145     write (s, b);
146     write (s, string'(" cin = "));
147     write (s, c);
148     write (s, string'(" sum expected = "));
149     write (s, sum_out);
150     write (s, string'(" cout expected = "));
151     write (s, cout_expected);
```

```
152         write (s, string'(" actual sum = "));
153         write (s, s_int);
154         write (s, string'(" actual cout = "));
155         write (s, cout);
156         writeline (output, s);
157 
158         cout_compare1 <= cout_expected XOR cout;
159         s_compare2 <= (sum_out(3) XOR s_int(3)) OR (sum_out(2) XOR s_int(2))
               OR (sum_out(1) XOR s_int(1)) OR (sum_out(0) XOR s_int(0)) ;
160 
161         if (cout_compare1 = '1') or (s_compare2 = '1') then
162             test_failed <= '1';
163         end if;
164 
165         for k in 2 to 4 loop
166           count <= count + '1';
167 
168           wait for 10 ns;
169           a <= a - count; b <=  b + count;
170           add_two_values(a, b, c, sum_out);
171           cout_expected <= sum_out(4);
172 
173           wait for 10 ns;
174             write (s, string'("a = "));
175             write (s, a);
176             write (s,  string'("b = "));
177             write (s, b);
178             write (s, string'(" cin = "));
179             write (s, c);
180             write (s, string'(" sum expected = "));
181             write (s, sum_out);
182             write (s, string'(" cout expected = "));
183             write (s, cout_expected);
184             write (s, string'(" actual sum = "));
185             write (s, s_int);
186             write (s, string'(" actual cout = "));
187             write (s, cout);
188             writeline (output, s);
189 
190           cout_compare1 <= cout_expected XOR cout;
191           s_compare1 <= (sum_out(3) XOR s_int(3)) OR (sum_out(2) XOR
                 s_int(2)) OR (sum_out(1) XOR s_int(1)) OR (sum_out(0) XOR
                 s_int(0));
192 
193           if (cout_compare1 = '1') or (s_compare1 = '1') then
194             test_failed <= '1';
195           end if;
196         end loop;
197 
198         wait for 10 ns;
199 
200         if (test_failed = '1') then
201             write (s, string'("Simulation DONE"));
202             write (s, string'(" Test Failed"));
203             writeline (output, s);
204         else
205             write (s, string'("Simulation DONE"));
206             write (s, string'(" Test Passed"));
207             writeline (output, s);
```

```
208         end if;
209
210         wait;
211     end process;
212 end behavior;
```

code/lab331_tb.vhd

2.3.6 3-2 Waveform Generator Testbench

```
1  ----------------------------------------------------------------------------------
2  -- Module Name: waveform_generation_tb
3  ----------------------------------------------------------------------------------
4
5  library IEEE;
6  use IEEE.STD_LOGIC_1164.ALL;
7
8  library UNISIM;
9  use UNISIM.VComponents.all;
10
11 Entity waveform_generation_tb Is
12 end waveform_generation_tb;
13
14 Architecture behavior of waveform_generation_tb Is
15     Signal a : STD_LOGIC := '0';
16     Signal g1 : STD_LOGIC := '0';
17     Signal g2 : STD_LOGIC := '1';
18
19 begin
20
21     process
22     begin
23         wait for 40ns;
24         a <= '1';
25         wait for 20ns;
26         g1 <= '1';
27         wait for 20ns;
28         g2 <= '0';
29         wait for 20ns;
30         a <= '0';
31         wait for 20ns;
32         g1 <= '0';
33         wait for 20ns;
34         g2 <= '1';
35     end process;
36
37
38 end behavior;
```

code/lab332_tb.vhd

3 Questions