

EECS Homework Assignment #7 - ALU Design || Tyler Youk

Code:

```
//code for rc_adder_slice
module rc_adder_slice (
    input logic a,b,c_in,
    output logic s, c_out
);

    logic p, g;

    assign p = a ^ b;
    assign g = a & b;

    assign s = a^b^c_in;
    assign c_out = ((a^b)&c_in) | (a&b);

endmodule

//code for alu_slice
module alu_slice (
    input logic    a, b, c_in,
    input logic [1:0] f,
    output logic    s, c_out
);

    logic b_inv, c;

    assign b_inv = b ^ f[0];
    assign c = c_in & f[1]; // COMPLETE THIS LINE

    rc_adder_slice U1 (    // COMPLETE
        .a(a),
        .b(b_inv),
        .c_in(c),
        .s(s),
        .c_out(c_out)
    );

endmodule

// code for gen_alu.sv
module gen_alu #(
```

```

parameter N = 8
)(
  input logic [N-1:0] a, b,
  input logic  [1:0] f,
  output logic [N-1:0] s,
  output logic      co
);

logic [N:0] c;

assign c[0] = f[0];
assign co = c[N]; // COMPLETE THIS LINE

```

```

alu_slice U[N-1:0] ( // COMPLETE
  .a(a),
  .b(b),
  .f(f),
  .s(s),
  .c_out(c[N:1]), // c array output = c [8 to 1]
  .c_in(c[N-1:0]) // c array input = c [7 to 0]
);

```

Endmodule

```

// code for testbench_lab4.sv
module testbench_lab4 ();

```

```

  logic [3:0] a4, b4, s4;
  logic [7:0] a8, b8, s8;
  logic [1:0] f;
  logic      co4, co8;

```

```

  logic      clk;
  logic [25:0] tvs [15:0];
  int i = 0;

```

```

  gen_alu #(
    .N(4)
  ) UUT4 (
    .a(a4),

```

```

        .b(b4),
        .f(f),
        .s(s4),
        .co(co4)
    );

    gen_alu #( .N(8)) // COMPLETE
        UUT8 (
            .a(a8),
            .b(b8),
            .f(f),
            .s(s8),
            .co(co8)
        );

    initial begin
        clk = 1'b0;
        forever #10 clk = ~clk;
    end

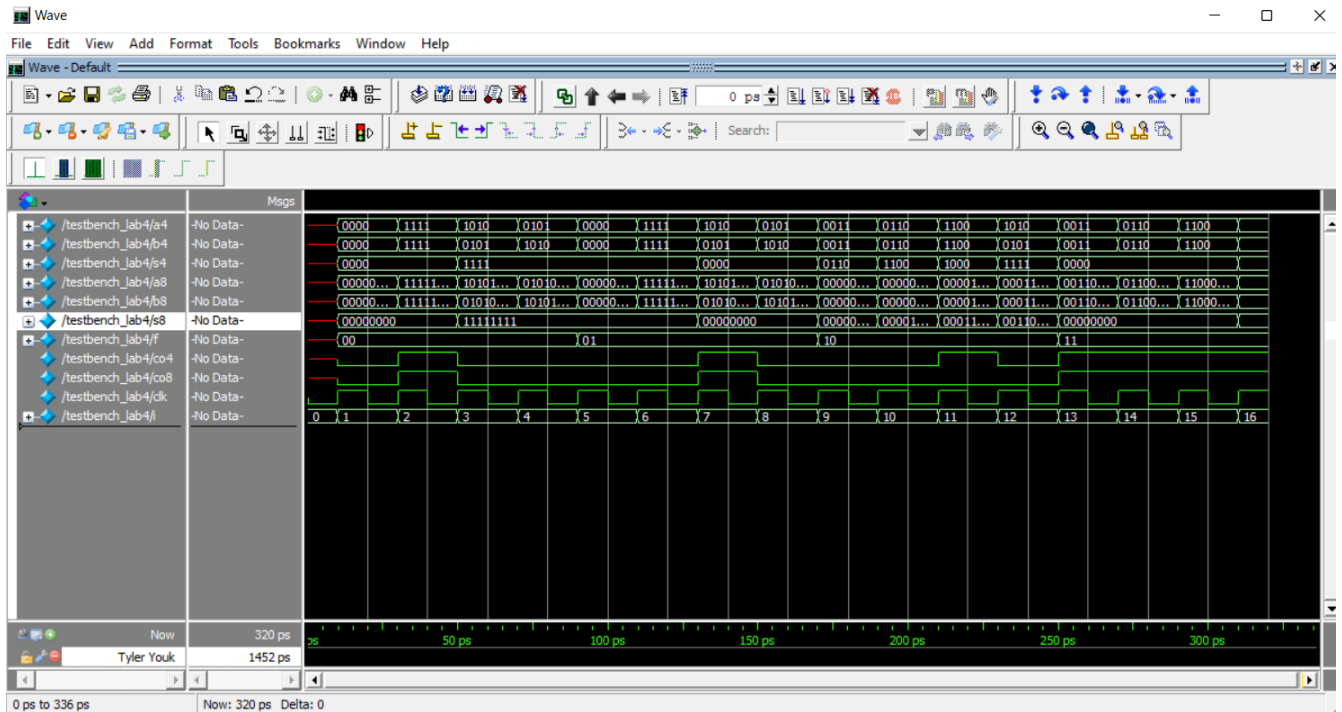
    always @(posedge clk) begin
        f = tvs[i][25:24];
        a4 = tvs[i][23:20]; // COMPLETE THIS LINE
        b4 = tvs[i][19:16]; // COMPLETE THIS LINE
        a8 = tvs[i][15:8];
        b8 = tvs[i][7:0];
        i++;
    end

    initial begin
        $readmemb("test_vectors.txt", tvs);
        $display("TIME | A4 B4 | S4 CO4 | A8 B8 | S8 CO8 |");
        $monitor(" %3d | %4b %4b | %4b %b | %8b %8b | %8b %b |", $time, a4, b4, s4, co4, a8,
b8, s8, co8);
        #320 $finish();
    end

endmodule

// Screenshots with Tyler Youk tags below

```



```

M Transcript
File Edit View Bookmarks Window Help

# Loading work.testbench_lab4
# Loading work.gen_alu
# Loading work.alu_slice
# Loading work.rc_adder_slice
add wave -position insertpoint sim:/testbench_lab4/*
VSIM 19> run -all
# TIME | A4 | B4 | S4 | CD4 | A8 | B8 | S8 | CO8 |
# 0 | XXXX XXXX | XXXX X | XXXXXXXX XXXXXXXX | XXXXXXXX X |
# 10 | 0000 0000 | 0000 0 | 00000000 00000000 | 00000000 0 |
# 30 | 1111 1111 | 0000 1 | 11111111 11111111 | 00000000 1 |
# 50 | 1010 0101 | 1111 0 | 10101010 01010101 | 11111111 0 |
# 70 | 0101 1010 | 1111 0 | 01010101 10101010 | 11111111 0 |
# 90 | 0000 0000 | 1111 0 | 00000000 00000000 | 11111111 0 |
# 110 | 1111 1111 | 1111 0 | 11111111 11111111 | 11111111 0 |
# 130 | 1010 0101 | 0000 1 | 10101010 01010101 | 00000000 1 |
# 150 | 0101 1010 | 0000 0 | 01010101 10101010 | 00000000 0 |
# 170 | 0011 0011 | 0110 0 | 00000011 00000011 | 00000110 0 |
# 190 | 0110 0110 | 1100 0 | 00000110 00000110 | 00001100 0 |
# 210 | 1100 1100 | 1000 1 | 00001100 00001100 | 00011000 0 |
# 230 | 1010 0101 | 1111 0 | 00011000 00011000 | 00110000 0 |
# 250 | 0011 0011 | 0000 1 | 00110000 00110000 | 00000000 1 |
# 270 | 0110 0110 | 0000 1 | 01100000 01100000 | 00000000 1 |
# 290 | 1100 1100 | 0000 1 | 11000000 11000000 | 00000000 1 |
# 310 | 1010 0101 | 0101 1 | 11111111 10101010 | 01010101 1 |
# ** Note: $finish : C:/intelFPGA/HW7Lab/testbench_lab4.sv(49)
# Time: 320 ps Iteration: 0 Instance: /testbench_lab4
# 1
# Break in Module testbench_lab4 at C:/intelFPGA/HW7Lab/testbench_lab4.sv line 49
VSIM 20> Tyler Youk
Tyler alu_slice screenshot.png
Tyler alu_slice.sv
Tyler alu_slice.sv.bak
Tyler Gen alu code screenshot.png

```

//additional screenshots of code

```
C:/intelFPGA/HW7Lab/gen_alu.sv (/testbench_lab4/UUT8) - Default
Ln# 3546 ps
1 module gen_alu #(
2     parameter N = 8
3 ) (
4     input logic [N-1:0] a, b,
5     input logic [1:0] f,
6     output logic [N-1:0] s,
7     output logic co
8 );
9
10 logic [N:0] c;
11
12 assign c[0] = f[0];
13 assign co = c[N]; // COMPLETE THIS LINE
14
15
16 alu_slice U[N-1:0] ( // COMPLETE
17     .a(a),
18     .b(b),
19     .f(f),
20     .s(s),
21     .c_out(c[N:1]), // c array output = c [8 to 1]
22     .c_in(c[N-1:0]) // c array input = c [7 to 0]
23 );
24
25
26
27 endmodule
```

```
C:/intelFPGA/HW7Lab/rc_adder_slice.sv (/testbench_lab4/UUT8/U[0]/U1) - Default
Ln# 3546 ps
1 module rc_adder_slice (
2     input logic a,b,c_in,
3     output logic s, c_out
4 );
5
6     logic p, g;
7
8     assign p = a ^ b;
9     assign g = a & b;
10
11     assign s = a^b^c_in;
12     assign c_out = ((a^b)&c_in) | (a&b);
13
14 endmodule
```

```
C:/intelFPGA/HW7Lab/testbench_lab4.sv (/testbench_lab4) - Default
Ln# 3546 ps
1  module testbench_lab4 ();
2
3      logic [3:0] a4, b4, s4;
4      logic [7:0] a8, b8, s8;
5      logic [1:0] f;
6      logic      co4, co8;
7
8      logic      clk;
9      logic [25:0] tvs [15:0];
10     int i = 0;
11
12     gen_alu #(
13         .N(4)
14     ) UUT4 (
15         .a(a4),
16         .b(b4),
17         .f(f),
18         .s(s4),
19         .co(co4)
20     );
21
22     gen_alu #( .N(8) ) // COMPLETE
23     UUT8 (
24         .a(a8),
25         .b(b8),
26         .f(f),
27         .s(s8),
28         .co(co8)
29     );
30
31     initial begin
32         clk = 1'b0;
33         forever #10 clk = ~clk;
34     end
```

gen_alu.sv testbench_lab4.sv rc_adder_slice.sv alu_slice.sv

```
C:/intelFPGA/HW7Lab/testbench_lab4.sv (/testbench_lab4) - Default
Ln# 3546 ps
20 );
21
22 gen_alu #( .N(8)) // COMPLETE
23 UUT8 (
24     .a(a8),
25     .b(b8),
26     .f(f),
27     .s(s8),
28     .co(co8)
29 );
30
31 initial begin
32     clk = 1'b0;
33     forever #10 clk = ~clk;
34 end
35
36 always @(posedge clk) begin
37     f = tvs[i][25:24];
38     a4 = tvs[i][23:20]; // COMPLETE THIS LINE
39     b4 = tvs[i][19:16]; // COMPLETE THIS LINE
40     a8 = tvs[i][15:8];
41     b8 = tvs[i][7:0];
42     i++;
43 end
44
45 initial begin
46     $readmemb("test_vectors.txt", tvs);
47     $display("TIME | A4 B4 | S4 CO4 | A8 B8 | S8 CO8 |");
48     $monitor(" %3d | %4b %4b | %4b %b | %8b %8b | %8b %b |", $time, a4, b4, s4, co4, a8, b8, s8, co8);
49     #320 $finish();
50 end
51
52 endmodule

gen_alu.sv testbench_lab4.sv rc_adder_slice.sv alu_slice.sv
```

```
C:/intelFPGA/HW7Lab/alu_slice.sv (/testbench_lab4/UUT8/U[0]) - Default
Ln# 3546 ps
1 module alu_slice (
2     input logic a, b, c_in,
3     input logic [1:0] f,
4     output logic s, c_out
5 );
6
7     logic b_inv, c;
8
9     assign b_inv = b ^ f[0];
10    assign c = c_in & f[1]; // COMPLETE THIS LINE
11
12    rc_adder_slice U1 ( // COMPLETE
13        .a(a),
14        .b(b_inv),
15        .c_in(c),
16        .s(s),
17        .c_out(c_out)
18    );
19
20 endmodule

gen_alu.sv testbench_lab4.sv rc_adder_slice.sv alu_slice.sv
```