CS 223 Digital Design

Section 01

Laboratory Assignment 2

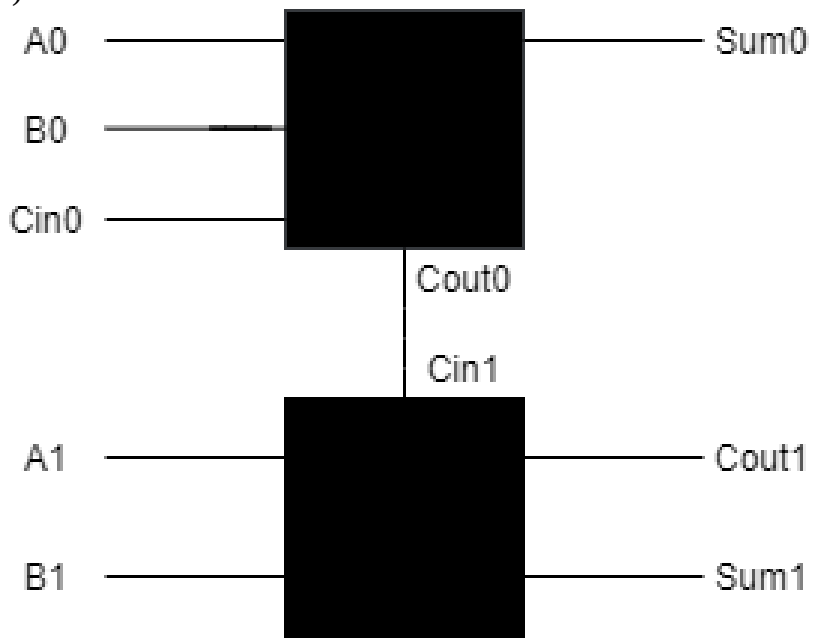
Full adder, full subtractor and 2-bit adder on FPGA

Arda Önal

21903350

17.10.2020

(b) Circuit schematic for a 2-bit adder made from two full adders



(c) Behavioral SystemVerilog module for the full adder.

```
module full_adder_behavioral(input logic a, b, cin, output logic s, cout );
    assign s = (a ^ b)^ cin;
    assign cout = ((a ^ b)& cin) | ( a & b);
endmodule
```

(d) Structural SystemVerilog module for the full adder and a testbench for it.

```
module full_adder_structural(input logic a, b, cin,output logic s, cout);
    logic n1;
    logic n2;
    logic n3;

    xor (n1,a,b);
    xor(s,n1,cin);
    and(n2,n1,cin);
```

```
    and(n3,a,b);
    or(cout,n2,n3);

endmodule

module testbench_full_adder();
    logic a,b,cin;
    logic s,cout;

    full_adder_structural dut(a,b,cin,s,cout);

    initial begin
        a = 0; b = 0; cin = 0; #10;
        if (s != 0 || cout != 0) $display("000 failed");
        cin = 1; #10;
        if (s != 1 || cout != 0) $display("001 failed");
        b = 1; cin = 0; #10;
        if (s != 1 || cout != 0) $display("010 failed");
        cin = 1; #10;
        if (s != 0 || cout != 1) $display("011 failed");
        a = 1; b = 0; cin =0; #10;
        if (s != 1 || cout != 0) $display("100 failed");
        cin = 1;#10;
        if (s != 0 || cout != 1) $display("101 failed");
        b = 1; cin = 0;#10;
        if (s != 0 || cout != 1) $display("110 failed");
        cin = 1;#10;
        if (s != 1 || cout != 1) $display("111 failed");
    end
endmodule
```

(e) Structural SystemVerilog module for the full subtractor and a testbench for it

```
module full_subtractor_structural(input logic a, b, bin,output logic d, bo);

    logic n1,n2,n3,n4;

    xor(n1,a,b);
    xor(d,bin,n1);
    and(n2,~a,b);
    and(n3,bin,~n1);
    or(bo,n2,n3);
endmodule

module testbench_full_subtractor();
    logic a,b,bin;
    logic d,bo;

    full_subtractor_structural dut(a,b,bin,d,bo);

    initial begin
        a = 0; b = 0; bin = 0; #10;
        if (d != 0 || bo != 0) $display("000 failed");
        bin = 1; #10;
        if (d != 1 || bo != 1) $display("001 failed");
        b = 1; bin = 0; #10;
        if (d != 1 || bo != 1) $display("010 failed");
        bin = 1; #10;
        if (d != 0 || bo != 1) $display("011 failed");
        a = 1; b = 0; bin =0; #10;
```

```
        if (d != 1 || bo != 0) $display("100 failed");
        bin = 1;#10;
        if (d != 0 || bo != 0) $display("101 failed");
        b = 1; bin = 0;#10;
        if (d != 0 || bo != 0) $display("110 failed");
        bin = 1;#10;
        if (d != 1 || bo != 1) $display("111 failed");
    end
endmodule
```

(f) Structural SystemVerilog module for the 2-bit adder and a testbench

```
module two_bit_full_adder ( input logic a0, b0, cin0, a1, b1, output logic s0, s1,cout1);

    logic cout0;

    full_adder_behavioral adder0( a0, b0, cin0, s0, cout0);
    full_adder_behavioral adder1( a1, b1, cout0, s1, cout1);
endmodule

module testbench_two_bit_adder();

    logic a0,b0,a1,b1,cin0;

    logic sum0,sum1,cout1;

    two_bit_full_adder dut(a0,b0,cin0,a1,b1,sum0,sum1,cout1);

    initial begin
        a0 = 0; b0 = 0; a1 = 0; b1 = 0; cin0 = 0; #10;
```

```
if ( sum0 != 0 || sum1 != 0 || cout1 != 0) $display("00000 failed");
b0 = 1; #10;
if ( sum0 != 1 || sum1 != 0 || cout1 != 0) $display("01000 failed");
b0 = 0; b1 = 1; #10;
if ( sum0 != 0 || sum1 != 1 || cout1 != 0) $display("00001 failed");
b0 = 1; #10;
if ( sum0 != 1 || sum1 != 1 || cout1 != 0) $display("01001 failed");
b0 = 0; b1 = 0; a0 = 1; #10;
if ( sum0 != 1 || sum1 != 0 || cout1 != 0) $display("10000 failed");
b0 = 1; #10;
if ( sum0 != 0 || sum1 != 1 || cout1 != 0) $display("11000 failed");
b0 = 0; b1 = 1; #10;
if ( sum0 != 1 || sum1 != 1 || cout1 != 0) $display("10001 failed");
b0 = 1; #10;
if ( sum0 != 0 || sum1 != 0 || cout1 != 1) $display("11001 failed");
b0 = 0; b1 = 0; a0 = 0; a1 = 1; #10;
if ( sum0 != 0 || sum1 != 1 || cout1 != 0) $display("00010 failed");
b0 = 1; #10;
if ( sum0 != 1 || sum1 != 1 || cout1 != 0) $display("01010 failed");
b0 = 0; b1 = 1; #10;
if ( sum0 != 0 || sum1 != 0 || cout1 != 1) $display("00011 failed");
b0 = 1; #10;
if ( sum0 != 1 || sum1 != 0 || cout1 != 1) $display("01011 failed");
b0 = 0; b1 = 0; a0 = 1; #10;
if ( sum0 != 1 || sum1 != 1 || cout1 != 0) $display("10010 failed");
b0 = 1; #10;
if ( sum0 != 0 || sum1 != 0 || cout1 != 1) $display("11010 failed");
b0 = 0; b1 = 1; #10;
if ( sum0 != 1 || sum1 != 0 || cout1 != 1) $display("10011 failed");
b0 = 1; #10;
```

```
        if ( sum0 != 0 || sum1 != 1 || cout1 != 1) $display("11011 failed");
    end
endmodule
```