CS223 DIGITAL DESIGN

SECTION 3

Laboratory Assignment 2:

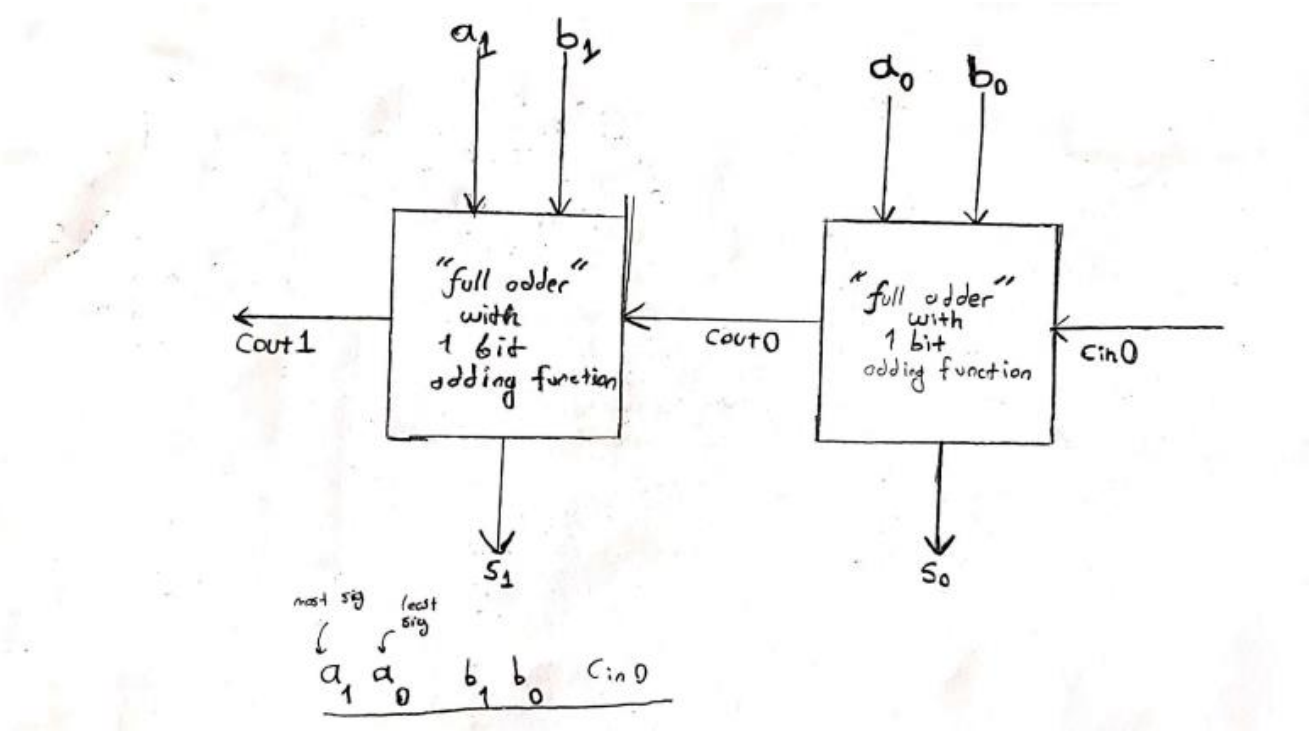
Full adder, full subtractor and 2-bit adder on FPGA

Nurettin Onur Vural 21902330

18.10.2020

PRELIMINARY REPORT

(b) Circuit schematic for a 2-bit adder made from two full adders. (use full adders as black-boxes, you don't need to draw the logic diagram of full adder again).



(c) Behavioral SystemVerilog module for the full adder.

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

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

```
module halfAdder(  input logic a, b, output logic sum, cout );
  assign sum =  a ^ b;
  assign cout =  a & b;
endmodule

module orGate( input logic a, b, output logic sum );
  assign sum =  a | b;
endmodule
```

```
module fullAdder_Structural( input logic a, b, cin, output logic sum, cout );
   logic sum1, cout1, cout2;

   halfAdder ha1(a , b, sum1, cout1);
   halfAdder ha2(sum1, cin, sum, cout2);
   orGate oG(cout1, cout2, cout);
endmodule
//TESTBENCH FOR FULL ADDER--------------------------------------------------------------------------------
module fullAdder_Structural_tb( );
   logic a, b, cin;
   logic sum, cout;

 // instantiate device under test
 fullAdder_Structural dut(a, b, cin, sum, cout);

 // apply inputs one at a time
 initial begin
 a = 0; b = 0; cin = 0; #10;
 cin = 1; #10;
 b = 1; cin = 0; #10;
 cin = 1; #10;
 a = 1; b = 0; cin = 0; #10;
 cin = 1; #10;
 b = 1; cin = 0; #10;
 cin = 1; #10;
 end
endmodule
```

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

```
module halfSubstractor( input logic a, b, output diff, bout);
  assign diff =  a ^ b;
  assign bout =  ~a & b;
endmodule

module orGate( input logic a, b, output logic sum);
  assign sum =  a | b;
endmodule

module fullSubstractor( input logic a, b, bin, output logic diff, bout);
  logic d1,b1,b2;

  halfSubstractor hs1(a, b, d1, b1);
  halfSubstractor hs2(d1, bin, diff, b2);
  orGate og(b1,b2,bout);
endmodule
//TESTBENCH FOR FULL SUBSTRACTOR----------------------------------------------------------------------------
module fullSubstractor_tenchbench();
 logic a, b, bin;
 logic diff, bout;

 // instantiate device under test
 fullSubstractor dut(a, b, bin, diff, bout);

 // apply inputs one at a time
 initial begin
 a = 0; b = 0; bin = 0; #10;
 bin = 1; #10;
 b = 1; bin = 0; #10;
 bin = 1; #10;
```

```
 a = 1; b = 0; bin = 0; #10;
 bin = 1; #10;
 b = 1; bin = 0; #10;
 bin = 1; #10;
 end
endmodule
```

(f) Structural SystemVerilog module for the 2-bit adder and a testbench for it. Use the full adder module you wrote in part (c).

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

module twoBitAdder( input logic a0, b0, a1, b1, cin0, output logic s0, s1, cout1);
   logic cout0;

  fullAdder fa1( a0, b0, cin0, s0, cout0);
  fullAdder fa2( a1, b1, cout0, s1, cout1);
endmodule
//TESTBENCH FOR TWO BIT ADDER----------------------------------------------------------------------------
module twoBitAdder_testbench();
 logic a0, b0, a1, b1, cin0;
 logic s0, s1, cout1;

 // instantiate device under test
 twoBitAdder dut( a0, b0, a1, b1, cin0, s0, s1, cout1);

 // apply inputs one at a time
 initial begin
 a1 = 0; a0 = 0; b1 = 0; b0 = 0; cin0 = 0; #10;
 cin0 = 1; #10;
 cin0 = 0; b0 = 1; #10;
 b0 = 0; a0 = 1; #10;
 a0 = 0; b0 = 1; cin0 = 1; #10;
 b0 = 0; a0 = 1; #10;
 b0 = 1; cin0 = 0; #10;
 cin0 = 1; #10;
```

```
a0 = 0; b1 = 1; b0 = 0; cin0 = 0; #10;
cin0 = 1; #10;
b0 = 1; cin0 = 0; #10;
a0 = 1; b0 = 0; #10;
a0 = 0; b0 = 1; cin0 = 1; #10;
a0 = 1; b0 = 0; #10;
b0 = 1; cin0 = 0; #10;
cin0 = 1; #10;
a1 = 1; a0 = 0; b1 = 0; b0 = 0; cin0 = 0; #10;
cin0 = 1; #10;
b0 = 1; cin0 = 0; #10;
a0 = 1; b0 = 0; #10;
a0 = 0; b0 = 1; cin0 = 1; #10;
a0 = 1; b0 = 0; #10;
b0 = 1; cin0 = 0; #10;
cin0 = 1; #10;
a0 = 0; b1 = 1; b0 = 0; cin0 = 0; #10;
cin0 = 1; #10;
b0 = 1; cin0 = 0; #10;
a0 = 1; b0 = 0; #10;
a0 = 0; b0 = 1; cin0 = 1; #10;
a0 = 1; b0 = 0; #10;
b0 = 1; cin0 = 0 ; #10;
cin0 = 1; #10;
end
endmodule
```