CS223

Digital Design

Section 2

Lab 2 – Preliminary Report

Seçkin Alp Kargı

22001942

CS

24.04.2023

Behavioral System Verilog Module for the Half Adder

```
module halfAdd
(
   input x,
   input y,
   output carry,
   output sum
);
assign carry = x &  y;
assign sum = x  ^ y;
endmodule
```

Test Bench for Half Adder

```
module testHalfAdd;
reg a;
reg b;
wire s, c;
half_adder halfAdd
(
   .a(a),
   .b(b),
   .s(s),
   .c(c)
);
Initial
begin
   a = 0;  b = 0;
    #10 $display("Case 1 - a=%b, b=%b, sum=%b, carry=%b", a, b, s, c);
   a = 0; b = 1;
   #10 $display("Case 2 - a=%b, b=%b, sum=%b, carry=%b", a, b, s, c);
   a = 1; b = 0;
   #10 $display("Case 3 - a=%b, b=%b, sum=%b, carry=%b", a, b, s, c);
   a = 1; b = 1;
   #10 $display("Case 4 - a=%b, b=%b, sum=%b, carry=%b", a, b, s, c);
   $finish;
end
endmodule
```

Behavioral System Verilog Module for the Half Subtractor

```
module halfSubs
(
   output a,
   output out,
   input x,
   input y
);
  assign out = ~x & y;
  assign a = x ^ y;
endmodule
```

Test Bench for Half Substractor

```
module test_half_subs;
reg X;
reg Y;
wire s;
wire c;
half_substractor halfSubs(
   .a(X),
   .b(Y),
   .sum(s),
   .carry(c)
);
  assign s = halfSubs.sum;
  assign c = halfSubs.carry;
initial
begin
X = 0; Y = 0; #10;
$display("Case 1 - X=%b, Y=%b, sum=%b, carry=%b", X, Y, s, c);
X = 0; Y = 1; #10;
$display("Case 2 - X=%b, Y=%b, sum=%b, carry=%b", X, Y, s, c);
X = 1; Y = 0; #10;
$display("Case 3 - X=%b, Y=%b, sum=%b, carry=%b", X, Y, s, c);
X = 1; Y = 1;  #10;
$display("Case 4 - X=%b, Y=%b, sum=%b, carry=%b", X, Y, s, c);
$finish;
end
endmodule
```

Structural System Verilog Module for the Full Adder

```
module addFull
(
   input a,
   input b,
   input carryfirst,
   output sum,
   output carrylast
);
wire halfAddSum1;
wire halfAddCar1;
wire halfAddSum2;
wire halfAddCar2;
half_adder half1
 (
   .a(a),
   .b(b),
   .sum(halfAddSum1),
   .carry(halfAddCar1)
);
half_adder half2
(
   .a(halfAddSum1),
   .b(carryfirst),
   .sum(halfAddSum2),
   .carry(halfAddCar2)
);
assign sum = halfAddSum2;
assign carrylast = halfAddCar1 | halfAddCar2;
endmodule
```

Test Bench for Full adder

```
module test_fullAdd;
wire sum;
wire carryLast;
reg x, y, carryFirst;
full_adder fadd
(
   .a(x),
   .b(y),
   .carry_in(carryFirst),
   .sum(sum),
   .carry_out(carryLast) );
initial begin
   x = 0; y = 0; carryFirst = 0; #10;
   $display("Case 1 - x=%b, y=%b, carryFirst=%b, sum=%b, carryLast=%b", x, y, carryFirst, sum, carryLast);
   x = 0; y = 1; carryFirst = 0; #10;
   $display("Case 2 - x=%b, y=%b, carryFirst=%b, sum=%b, carryLast=%b", x, y, carryFirst, sum, carryLast);
   x = 1; y = 0; carryFirst = 0; #10;
   $display("Case 3 - x=%b, y=%b, carryFirst=%b, sum=%b, carryLast=%b", x, y, carryFirst, sum, carryLast);
   x = 1; y = 1; carryFirst = 0; #10;
   $display("Case 4 - x=%b, y=%b, carryFirst=%b, sum=%b, carryLast=%b", x, y, carryFirst, sum, carryLast);
   x = 0; y = 0; carryFirst = 1; #10;
   $display("Case 5 - x=%b, y=%b, carryFirst=%b, sum=%b, carryLast=%b", x, y, carryFirst, sum, carryLast);
   x = 0; y = 1; carryFirst = 1; #10;
   $display("Case 6 - x=%b, y=%b, carryFirst=%b, sum=%b, carryLast=%b", x, y, carryFirst, sum, carryLast);
   x = 1; y = 0; carryFirst = 1; #10;
   $display("Case 7 - x=%b, y=%b, carryFirst=%b, sum=%b, carryLast=%b", x, y, carryFirst, sum, carryLast);
   x = 1; y = 1; carryFirst = 1; #10;
   $display("Case 8- x=%b, y=%b, carryFirst=%b, sum=%b, carryLast=%b", x, y, carryFirst, sum, carryLast);end
endmodule
```

Structural System Verilog Module for the Full Subtractor

```
module fullSub
 (
   output Y,
   output out,
   input A,
   input B,
   input in,
);
wire out1;
wire out2;
wire Y1;
wire Y2;
half_subtractor halfSub1
(
   .A(A),
   .B(B),
   .Y(Y1),
   .out(out1)
);
half_subtractor halfSub2
(
   .A(Y1),
   .B(in),
   .Y(Y),
   .out(out2)
);
assign out = out1 | out2;
endmodule
```

Test Bench for Full Subtractor

```
module testFullSub;
wire Y;
wire out;
reg in;
reg A;
reg B;
full_subtractor fullsub
(   .A(A),
    .B(B),
    .in(in),
    .Y(Y),
    .out(out) );
initial begin
    A = 0; B = 0; in = 0; #10;
    $display("Case 1 - A=%b, B=%b, in=%b, Y=%b, out=%b", A, B, in, Y, out);
    A = 0; B = 0; in = 1; #10;
    $display("Case 2 - A=%b, B=%b, in=%b, Y=%b, out=%b", A, B, in, Y, out);
    A = 0; B = 1; in = 0; #10;
    $display("Case 3 - A=%b, B=%b, in=%b, Y=%b, out= %b", A, B, in, Y, out);
    A = 0; B = 1; in = 1; #10;
    $display("Case 4 - A=%b, B=%b, in=%b, Y=%b, out=%b", A, B, in, Y, out);
    A = 1; B = 0; in = 0; #10;
    $display("Case 5 – A = %b, B=%b, in=%b, Y=%b, out=%b", A, B, in, Y, out);
    A = 1; B = 0; in = 1; #10;
    $display("Case 6 - A=%b, B=%b, in=%b, Y=%b, out=%b", A, B, in, Y, out);
    A = 1; B = 1; in = 0; #10;
    $display("Case 7 - A=%b, B=%b, in=%b, Y=%b, out=%b", A, B, in, Y, out);
    A = 1; B = 1; in = 1; #10;
    $display("Case 8 - A=%b, B=%b, in=%b, Y=%b, out=%b", A, B, in, Y, out);
    $finish;
end
endmodule
```