04.03.2020

CS 223-4

DIGITAL DESIGN

LAB 2

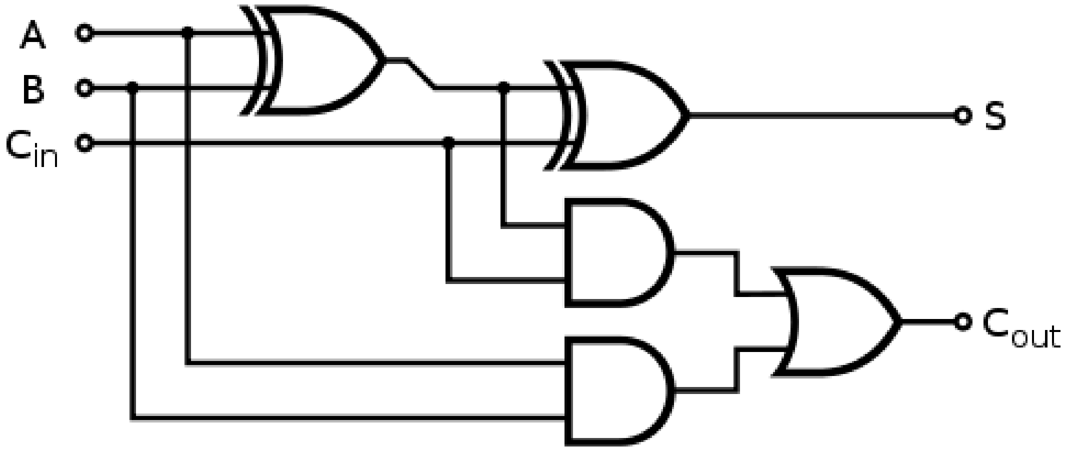
PRELIMINARY REPORT

ZÜBEYİR BODUR

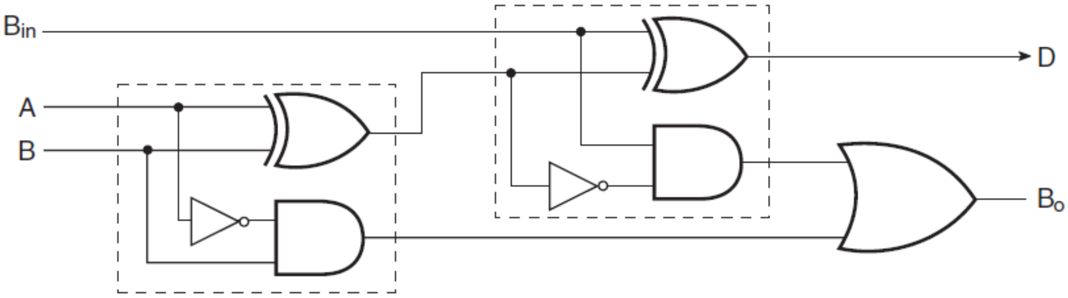
21702382

Spring 2020

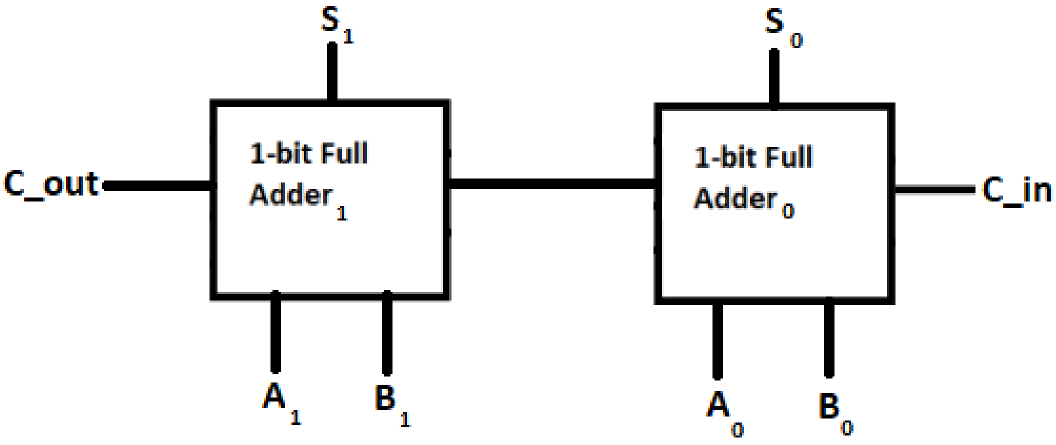
1. 1-Bit Full Adder



2. 1-Bit Full Substractor



3. 2-Bit Full Adder



4. System Verilog Modules

```
// Behavorial module for one bit full adder
//
module oneBitFullAdder( input logic a, b, c_in,
                  output logic c_out, sum
   );
   assign sum =  ~a & ~b & c_in | ~a & b & ~c_in
                           | a & ~b & ~c_in
                           | a & b & c_in;
   assign c_out = b & c_in | a & b | a & c_in;
endmodule


// Behavorial module for one bit full substractor
//
module oneBitFullSubstractor( input  logic a, b, b_in,
                       output logic b_out, dif
);
   assign dif =  ~a & ~b & b_in | ~a & b & ~b_in
                           | a & ~b & ~b_in
                           | a & b & b_in;
   assign b_out = b & b_in | ~a & b | ~a & b_in;
endmodule


// Structural module for one bit full adder
//
module fullAdder( input logic a, b, c_in,
              output logic c_out, sum
   );
    oneBitFullAdder adder(a, b, c_in, c_out, sum);
endmodule


// Structural module for one bit full substractor
//
module fullSubs( input logic a, b, b_in,
              output logic b_out, dif
   );
    oneBitFullSubstractor subs(a, b, b_in, b_out, dif);
endmodule
```

```
// Structural module for two bit full substractor
//
module TwoBitAdder( input logic a_1, a_0, b_1, b_0, c_in,
               output logic  c_out, sum_1, sum_0
   );
   logic c_out_1;
   fullAdder adder1( a_0, b_0, c_in, c_out_1, sum_0);
   fullAdder adder2( a_1, b_1, c_out_1, c_out, sum_1);
endmodule
```

5. Testbenches

```
module testAdder();
   logic a, b, c_in, c_out, sum;
   oneBitFullAdder test( a, b, c_in, c_out, sum);
   initial begin
      a = 0; b = 0; c_in = 0; #10;
      repeat(2) begin
         repeat(2) begin
            repeat(2) begin
               c_in = ~c_in; #10;
            end
            b = ~b; #10;
          end
          a = ~a; #10;
      end
   $stop;
   end
endmodule

module testFullAdder();
   logic a, b, c_in, c_out, sum;
   fullAdder test( a, b, c_in, c_out, sum);
   initial begin
      a = 0; b = 0; c_in = 0; #10;
      repeat(2) begin
         repeat(2) begin
            repeat(2) begin
               c_in = ~c_in; #10;
            end
            b = ~b; #10;
          end
          a = ~a; #10;
      end
   $stop;
   end
endmodule
```

```
module testFullSubstractor();
   logic a, b, b_in, dif, b_out;
   fullSubs test( a, b, b_in, b_out, dif);
   initial begin
      a = 0; b = 0; b_in = 0; #10;
      repeat(2) begin
         repeat(2) begin
            repeat(2) begin
               b_in = ~b_in; #10;
            end
            b = ~b; #10;
          end
          a = ~a; #10;
      end
   $stop;
   end
endmodule

module testTwoBitAdder();
   logic a_1, a_0, b_1, b_0, c_in,  c_out, sum_1, sum_0;
   TwoBitAdder test( a_1, a_0, b_1, b_0, c_in, c_out, sum_1, sum_0 );
   initial begin
      a_1 = 0; a_0 = 0 ; b_1 = 0; b_0 = 0; c_in = 0; #10;
      repeat(2) begin
         repeat(2) begin
            repeat(2) begin
               repeat(2) begin
                  repeat(2) begin
                     b_0 = ~b_0; #10;
                  end
                  b_1 = ~b_1; #10;
               end
               a_0 = ~a_0; #10;
             end
             a_1 = ~a_1; #10;
          end
          c_in = ~c_in; #10;
      end
   $stop;
   end
endmodule
```