①

22.02.2021

CS223

-Digital Design-

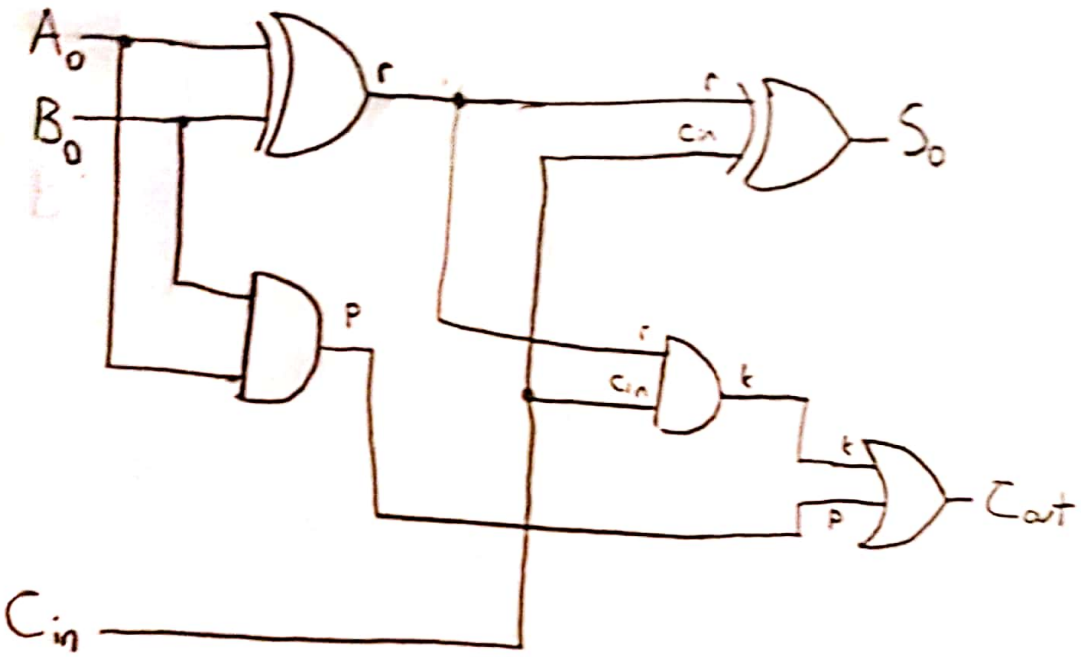
Section: 2

Name : Barkın Saday

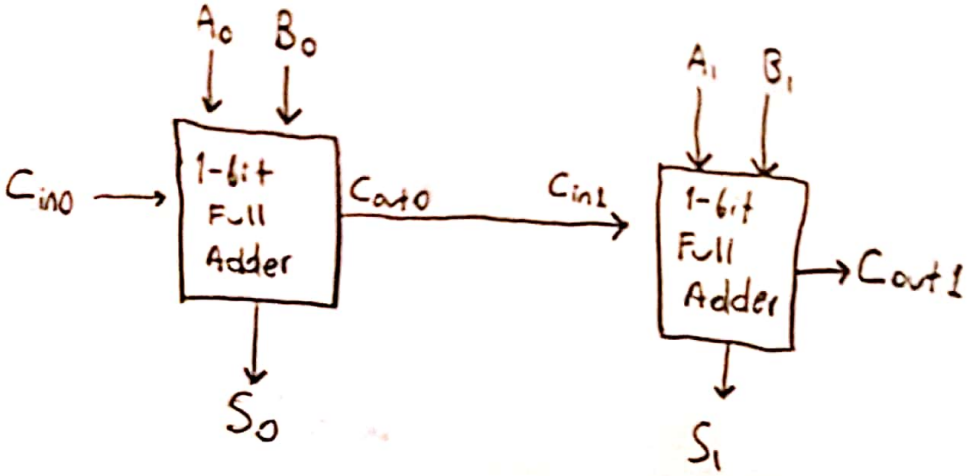
ID: 21902967

Lab: 2

Schematic for 1-bit full adder:



Schematic for 2-bit full adder: (using 1-bit full adders)



Note: $C_{out0} = C_{in1}$

One-Bit Full Adder Dataflow:

```
module one_bit_full_adder (
   input logic A, B, Cin,
   output logic S,Cout
   );

   //local logic variables
   logic r, p, k;
   assign r = A ^ B;
   assign p = A & B;
   assign k = r & Cin;

   //assiging output logics
   assign S = r ^ Cin;
   assign Cout = k | p;
Endmodule
```

Structural One-Bit Full Adder:

```
module st_one_bit_full_adder(
   input A,
   input B,
   input Cin,
   output S,
   output Cout
   );

   wire r, p, k;

   xor u1(r, A, B);
   and u2(p, A, B);
   and u3(k, r, Cin);
```

```
    xor u4(S, r, Cin);
    or u5(Cout, k, p);
Endmodule
```

Test Bench One-Bit Full Adder:

```
module tb_one_bit_full_adder();
    logic A, B, Cin, S, Cout;

    st_one_bit_full_adder uut(
        .A(A),
        .B(B),
        .Cin(Cin),
        .S(S),
        .Cout(Cout)
    );

    initial
     begin

     A = 0;
     B = 0;
     Cin = 0;
     #10;//1

     A = 1;
     B = 0;
     Cin = 0;
     #10;//2

     A = 0;
     B = 1;
```

```
    Cin = 0;
    #10;//3

    A = 1;
    B = 1;
    Cin = 0;
    #10;//4

    A = 0;
    B = 0;
    Cin = 1;
    #10;//5

    A = 1;
    B = 0;
    Cin = 1;
    #10;//6

    A = 0;
    B = 1;
    Cin = 1;
    #10;//7

    A = 1;
    B = 1;
    Cin = 1;
    #10;//8
    end
endmodule
```

Structural Two-Bit Adder:

```
module st_two_bit_adder(
   input A0,
   input A1,
   input B0,
   input B1,
   input Cin,
   output S0,
   output S1,
   output Cout
   );

   // c is Cout of first adder and Cin of second adder
   wire c;
   st_one_bit_full_adder g1(A0, B0, Cin, S0, c);//found S0 and c

   st_one_bit_full_adder g2(A1, B1, c, S1, Cout );//found S1 and Cout
Endmodule
```

TestBench Two-Bit Adder:

```
module tb_two_bit_adder();

   logic A0, A1, B0, B1, Cin, S0, S1, Cout;

   st_two_bit_adder f(
   .A(A), .B(B), .Cin(Cin),
   .S0(S0), .S1(S1), .Cout(Cout)
    );
```

```
initial
 begin

 A0 = 0;
 B0 = 0;
 A1 = 0;
 B1 = 0;
 #10;

 A0 = 1;
 B0 = 0;
 A1 = 0;
 B1 = 0;
 #10;

 A0 = 0;
 B0 = 1;
 A1 = 0;
 B1 = 0;
 #10;

 A0 = 1;
 B0 = 1;
 A1 = 0;
 B1 = 0;
 #10;

 A0 = 0;
 B0 = 0;
 A1 = 1;
```

```
B1 = 0;
#10;

A0 = 1;
B0 = 0;
A1 = 1;
B1 = 0;
#10;

A0 = 0;
B0 = 1;
A1 = 1;
B1 = 0;
#10;

A0 = 1;
B0 = 1;
A1 = 1;
B1 = 0;
#10;

A0 = 0;
B0 = 0;
A1 = 0;
B1 = 1;
#10;

A0 = 1;
B0 = 0;
A1 = 0;
B1 = 1;
```

```
#10;

A0 = 0;
B0 = 1;
A1 = 0;
B1 = 1;
#10;

A0 = 1;
B0 = 1;
A1 = 0;
B1 = 1;
#10;

A0 = 0;
B0 = 0;
A1 = 1;
B1 = 1;
#10;

A0 = 1;
B0 = 0;
A1 = 1;
B1 = 1;
#10;

A0 = 0;
B0 = 1;
A1 = 1;
B1 = 1;
#10;
```

```
      A0 = 1;
      B0 = 1;
      A1 = 1;
      B1 = 1;
      #10;
      end

     always
      begin
        Cin = 0;#5;
        Cin = 1;#5;
      end
endmodule
```