Cs223-03

Lab02

Osman Buğra Aydın

21704100

18 October 2020

A0
B0
Cin

Full Adder 1-bit

Cout

Q0

A1
B1
Cin

Full Adder 1-bit

Q1
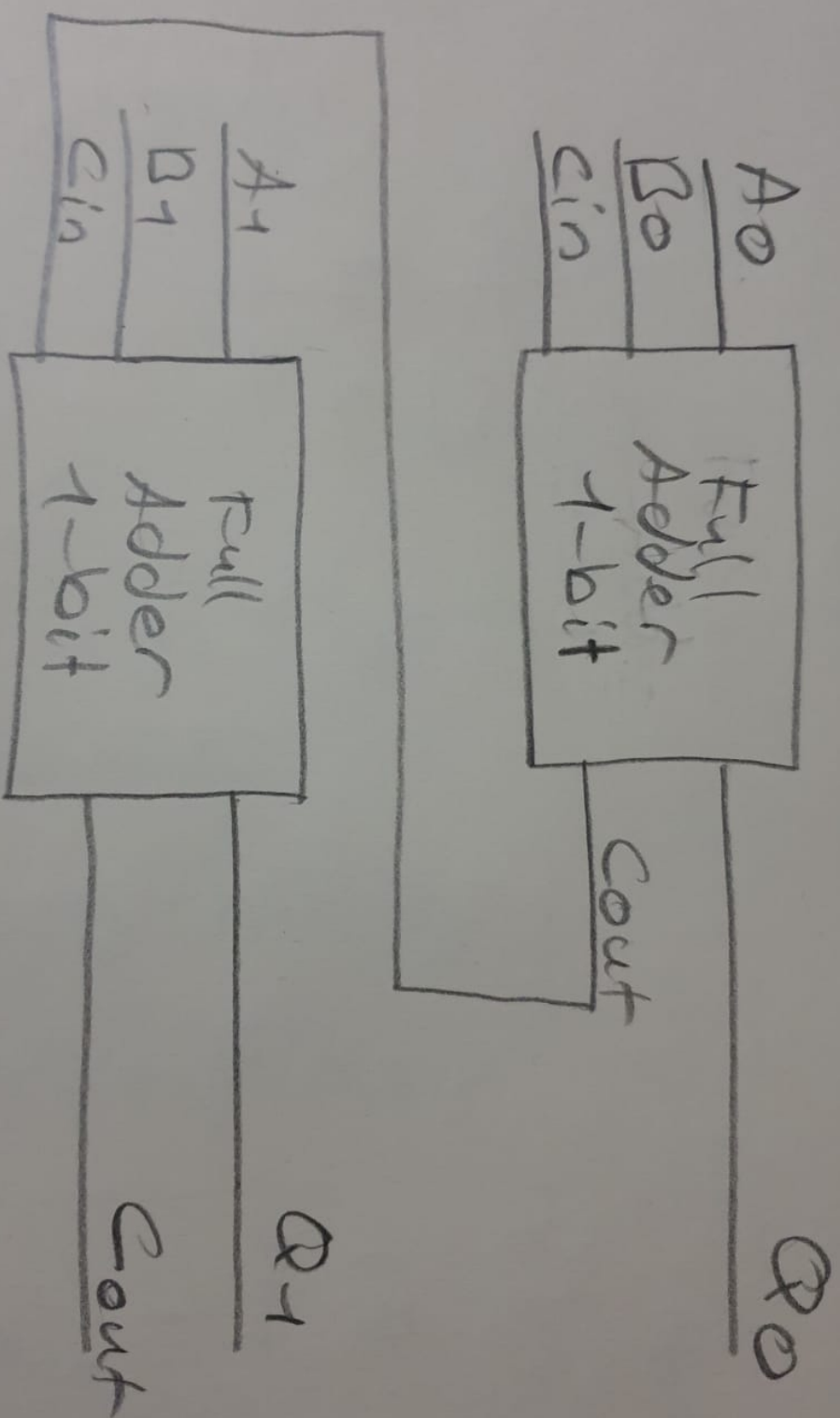
Cout

# c-) Behavioral Full Adder SystemVerilog

```
`timescale 1ns / 1ps

module BehavioralFa( input a, b, cin, output sum, cout );

    assign sum = ( a ^ b ) ^ cin ;
    assign cout = ( a & b ) | ( a ^ b ) & cin ;

endmodule
```

# d-) Structural Full Adder SystemVerilog

```
`timescale 1ns / 1ps

module Structural_FA( input a, b, cin, output sum, cout );

    wire w1, w2, w3;

    xor xor1(w1, a, b);
    xor xor2(sum, w1, cin);

    and and1(w2, a, b);
    and and2(w3, w1, cin);

    or or1(cout, w2, w3);

endmodule
```

# Testbench

```
`timescale 1ns / 1ps

module Structural_FA_tb;

   reg a, b, cin;
   wire sum, cout;

   Structural_FA uut(.a(a), .b(b), .cin(cin), .sum(sum), .cout(cout));

      initial begin
        #300;
        #50 a=0; b=0; cin=0;
        #50 a=0; b=0; cin=1;
        #50 a=0; b=1; cin=0;
        #50 a=0; b=1; cin=1;
        #50 a=1; b=0; cin=0;
        #50 a=1; b=0; cin=1;
        #50 a=1; b=1; cin=0;
        #50 a=1; b=1; cin=1;
      end

      initial $monitor($time,,"a=%d,b=%d,cin=%d, sum=%d, cout=%d",
a,b,cin,sum,cout);

endmodule
```

# e-) Structural Full Substractor SystemVerilog

```
`timescale 1ns / 1ps

module Structural_FS( input a, b, bin, output dif, bout );

   wire w1;
   wire w2;
   wire w3;
```

```verilog
    wire inversedA;
    wire inversedXor1;

    xor xor1( w1, a, b );
    xor xor2( dif, w1, bin );

    not not1( inversedA, a );
    not not2( inversedXor1, w1 );

    and and1( w2, inversedA, b );
    and and2( w3, inversedXor1, bin );

    or or1(bout, w2, w3);

endmodule
```

## Testbench

```verilog
`timescale 1ns / 1ps


module Structural_FS_tb;

    reg a, b, bin;
    wire dif, bout;

    Structural_FS uut(.a(a), .b(b), .bin(bin), .dif(dif), .bout(bout));

        initial begin
          #300;
          #50 a=0; b=0; bin=0;
          #50 a=0; b=0; bin=1;
          #50 a=0; b=1; bin=0;
          #50 a=0; b=1; bin=1;
          #50 a=1; b=0; bin=0;
          #50 a=1; b=0; bin=1;
          #50 a=1; b=1; bin=0;
```

```
        #50 a=1; b=1; bin=1;
    end

    initial $monitor($time,,"a=%d,b=%d,bin=%d, dif=%d, bout=%d",
a,b,bin,dif,bout);

endmodule
```

# F-) Structural 2-bit Full Adder SystemVerilog

```
`timescale 1ns / 1ps

module FA_TwoBit( input a0, a1, b0, b1, cin, output sum0, sum1, cout );

    wire w1;

    Structural_FA fullAdder1( a0, b0, cin, sum0, w1 );
    Structural_FA fullAdder2( a1, b1, w1, sum1, cout );

endmodule
```

## Testbench

```
`timescale 1ns / 1ps

module FA_TwoBit_tb;

    reg a0, a1, b0, b1, cin;
    wire sum0, sum1, cout;

    FA_TwoBit
uut( .a0(a0), .a1(a1), .b0(b0), .b1(b1), .cin(cin), .sum0(sum0), .sum1(sum1
), .cout(cout));

    initial begin
```

```verilog
        #100;
        #10 a0 = 0;  a1=0; b0=0;  b1=0; cin=0;
        #10 a0=0;  a1=0; b0=0;  b1=1; cin=0;
        #10 a0=0;  a1=0; b0=1;  b1=0; cin=0;
        #10 a0=0;  a1=0; b0=1;  b1=1; cin=0;
        #10 a0=0;  a1=1; b0=0;  b1=0; cin=0;
        #10 a0=0;  a1=1; b0=0;  b1=1; cin=0;
        #10 a0=0;  a1=1; b0=1;  b1=0; cin=0;
        #10 a0=0;  a1=1; b0=1;  b1=1; cin=0;
        #10 a0=1;  a1=1; b0=0;  b1=0; cin=0;
        #10 a0=1;  a1=0; b0=0;  b1=1; cin=0;
        #10 a0=1;  a1=0; b0=1;  b1=0; cin=0;
        #10 a0=1;  a1=0; b0=1;  b1=1; cin=0;
        #10 a0=1;  a1=1; b0=0;  b1=0; cin=0;
        #10 a0=1;  a1=1; b0=0;  b1=1; cin=0;
        #10 a0=1;  a1=1; b0=1;  b1=0; cin=0;
        #10 a0=1;  a1=1; b0=1;  b1=1; cin=0;
        #10 a0=0;  a1=0; b0=0;  b1=0; cin=1;
        #10 a0=0;  a1=0; b0=0;  b1=1; cin=1;
        #10 a0=0;  a1=0; b0=1;  b1=0; cin=1;
        #10 a0=0;  a1=0; b0=1;  b1=1; cin=1;
        #10 a0=0;  a1=1; b0=0;  b1=0; cin=1;
        #10 a0=0;  a1=1; b0=0;  b1=1; cin=1;
        #10 a0=0;  a1=1; b0=1;  b1=0; cin=1;
        #10 a0=0;  a1=1; b0=1;  b1=1; cin=1;
        #10 a0=1;  a1=1; b0=0;  b1=0; cin=1;
        #10 a0=1;  a1=0; b0=0;  b1=1; cin=1;
        #10 a0=1;  a1=0; b0=1;  b1=0; cin=1;
        #10 a0=1;  a1=0; b0=1;  b1=1; cin=1;
        #10 a0=1;  a1=1; b0=0;  b1=0; cin=1;
        #10 a0=1;  a1=1; b0=0;  b1=1; cin=1;
        #10 a0=1;  a1=1; b0=1;  b1=0; cin=1;
        #10 a0=1;  a1=1; b0=1;  b1=1; cin=1;
    end

        initial $monitor($time,,"a0=%b, a1=%b, b0=%b, b1=%b, cin=%b,
sum0=%b, sum1=%b, cout=%b", a0,a1,b0,b1,cin,sum0,sum1,cout);

endmodule
```