Course: CS-223 Digital Design
Section 01
Lab 02

Miray Ayerdem
21901987
25/02/2021

b) **Schematic for a 1-bit fulladder**

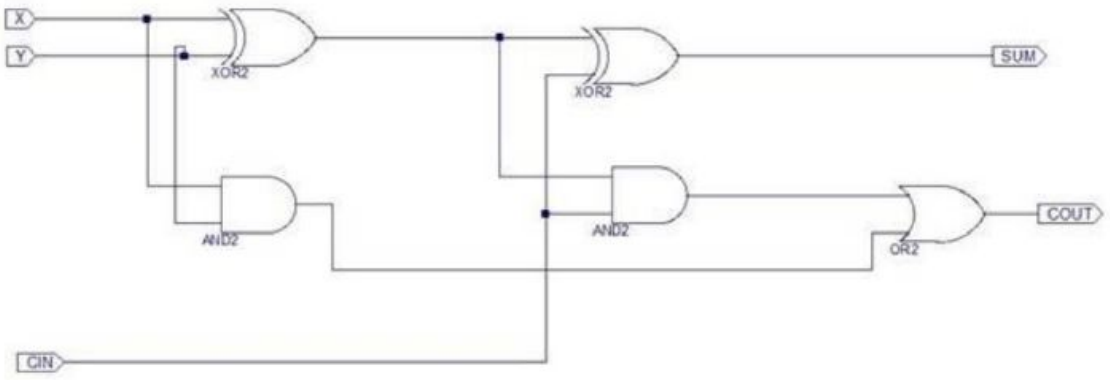
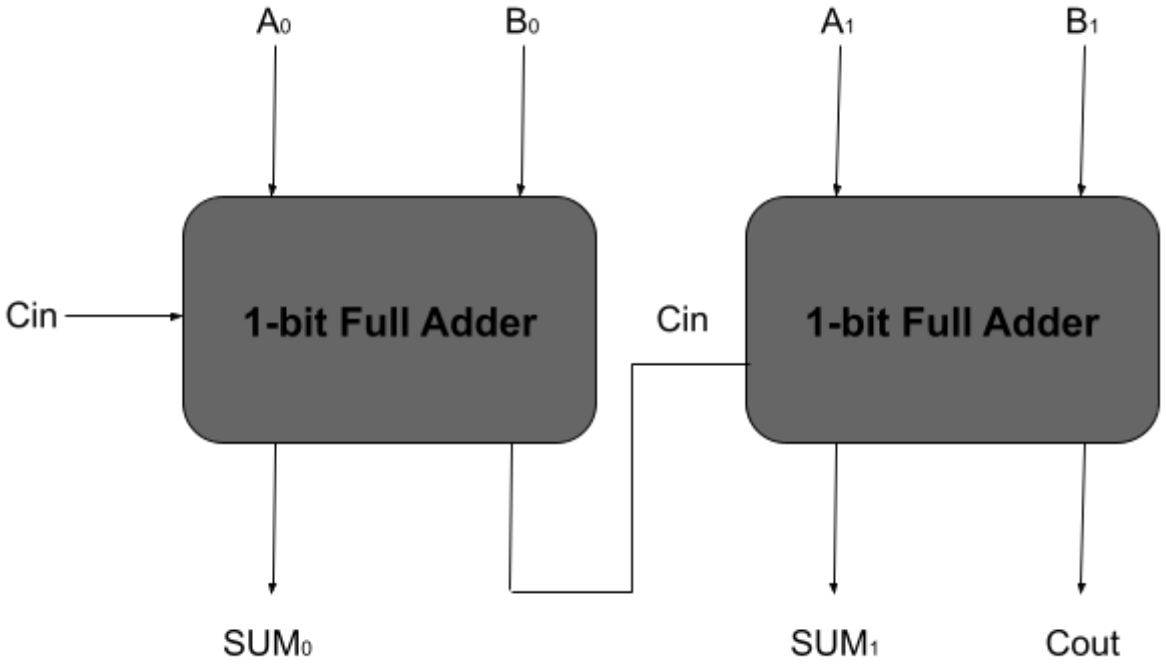


Fig 3: Full adder

c) **Schematic for a 2-bit adder made from two 1-bit full adders**



d) **Dataflow Systemverilog module for the 1-bit fulladder**

```
module fulladder(input logic x, y, z,
                 output logic sum, cout);
        assign sum = (x ^ y) ^ z;
        assign cout = (x & y) | ((x ^ y) & z);
endmodule
```

e) **Structural Systemverilog module for the 1-bit fulladder**

```
module xor2( input logic x, y,
                 output logic z);
        assign z = x ^ y;
endmodule

module and2(input logic x, y,
                 output logic z);
        assign z = x & y;
endmodule

module or2(input logic x, y,
                 output logic z);
        assign z = x | y;
endmodule

module fulladder2( input logic a, b, cin,
                 output logic d, cout );
        logic n1, n2, n3;
        xor2 gatexor1(a, b, n1);
        xor2 gatexor2(n1, cin, d);
        and2 gateand1(a, b, n2);
        and2 gateand2(n1, cin, n3);
        or2 gateor(n2, n3, cout);
endmodule
```

Testbench for the 1-bit fulladder

```
module testbench();
   logic x, y, z;
   logic sum, cout;
   fulladder2 dut(x, y, z, sum, cout);
   initial begin
   x = 0; y = 0; z = 0; #10;
   x = 0; y = 0; z = 1; #10;
   x = 0; y = 1; z = 0; #10;
```

```
    x = 0; y = 1; z = 1; #10;
    x = 1; y = 1; z = 0; #10;
    x = 1; y = 0; z = 0; #10;
    x = 1; y = 0; z = 1; #10;
    x = 1; y = 1; z = 1; #10;
    end
endmodule
```

f) **Structural Systemverilog module for the 2-bit adder**

```
module fulladder2bit(input logic a0, b0, a1, b1, cin,
                     output logic sum0, sum1, cout);
          logic n1;
          fulladder2 fulladder1(a0, b0, cin, sum0, n1);
          fulladder2 fulladder2(a1, b1, n1, sum1, cout);
endmodule
```

Testbench for the 2-bit adder

```
module testbench2();
    logic x0, y0, x1, y1, cin, sum0, sum1, cout;
    fulladder2bit dut(x0, y0, x1, y1, cin, sum0, sum1, cout);
    initial begin
        x0 = 0; y0 = 0; cin = 0; x1 = 0; y1 = 0; #10;
        x0 = 0; y0 = 0; cin = 0; x1 = 0; y1 = 1; #10;
        x0 = 0; y0 = 0; cin = 0; x1 = 1; y1 = 0; #10;
        x0 = 0; y0 = 0; cin = 0; x1 = 1; y1 = 1; #10;
        x0 = 0; y0 = 0; cin = 1; x1 = 0; y1 = 0; #10;
        x0 = 0; y0 = 0; cin = 1; x1 = 0; y1 = 1; #10;
        x0 = 0; y0 = 0; cin = 1; x1 = 1; y1 = 0; #10;
        x0 = 0; y0 = 0; cin = 1; x1 = 1; y1 = 1; #10;
        x0 = 0; y0 = 1; cin = 0; x1 = 0; y1 = 0; #10;
        x0 = 0; y0 = 1; cin = 0; x1 = 0; y1 = 1; #10;
        x0 = 0; y0 = 1; cin = 0; x1 = 1; y1 = 0; #10;
        x0 = 0; y0 = 1; cin = 0; x1 = 1; y1 = 1; #10;
        x0 = 0; y0 = 1; cin = 1; x1 = 0; y1 = 0; #10;
        x0 = 0; y0 = 1; cin = 1; x1 = 0; y1 = 1; #10;
        x0 = 0; y0 = 1; cin = 1; x1 = 1; y1 = 0; #10;
        x0 = 0; y0 = 1; cin = 1; x1 = 1; y1 = 1; #10;
        x0 = 1; y0 = 0; cin = 0; x1 = 0; y1 = 0; #10;
        x0 = 1; y0 = 0; cin = 0; x1 = 0; y1 = 1; #10;
        x0 = 1; y0 = 0; cin = 0; x1 = 1; y1 = 0; #10;
        x0 = 1; y0 = 0; cin = 0; x1 = 1; y1 = 1; #10;
        x0 = 1; y0 = 0; cin = 1; x1 = 0; y1 = 0; #10;
        x0 = 1; y0 = 0; cin = 1; x1 = 0; y1 = 1; #10;
        x0 = 1; y0 = 0; cin = 1; x1 = 1; y1 = 0; #10;
        x0 = 1; y0 = 0; cin = 1; x1 = 1; y1 = 1; #10;
        x0 = 1; y0 = 1; cin = 0; x1 = 0; y1 = 0; #10;
```

```
        x0 = 1; y0 = 1; cin = 0; x1 = 0; y1 = 1; #10;
        x0 = 1; y0 = 1; cin = 0; x1 = 1; y1 = 0; #10;
        x0 = 1; y0 = 1; cin = 0; x1 = 1; y1 = 1; #10;
        x0 = 1; y0 = 1; cin = 1; x1 = 0; y1 = 0; #10;
        x0 = 1; y0 = 1; cin = 1; x1 = 0; y1 = 1; #10;
        x0 = 1; y0 = 1; cin = 1; x1 = 1; y1 = 0; #10;
        x0 = 1; y0 = 1; cin = 1; x1 = 1; y1 = 1; #10;
    end
endmodule
```