



2018-2019 FALL SEMESTER
CS 223 – DIGITAL DESIGN

LAB 2 – 22.10.2018

SECTION: 1

NAME: MUNIB EMRE

SURNAME: SEVILGEN

STUDENT ID: 21602416

TRAINER PACK: 09

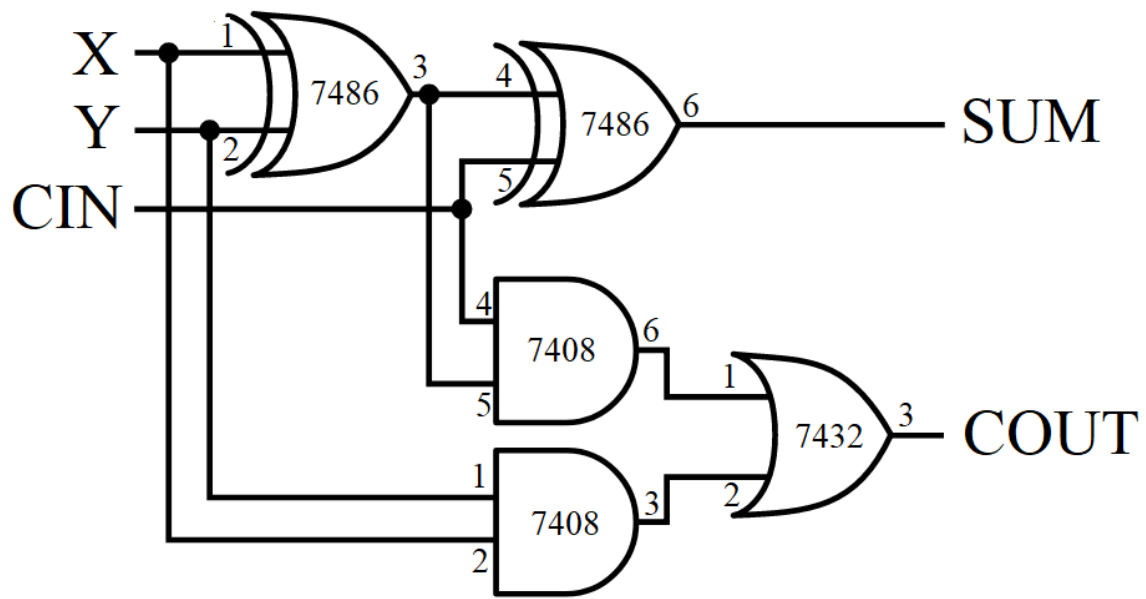


Figure 1 - Schematic for a 1-bit fulladder

IC List:

- One 7486 Quad 2-input XOR gate
- One 7408 Quad 2-input AND gate
- One 7432 Quad 2-input OR gate

7486: GND → 7, +5V → 14

7408: GND → 7, +5V → 14

7432: GND → 7, +5V → 14

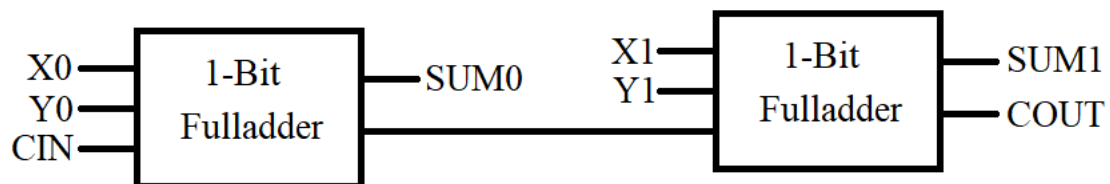


Figure 2 - Schematic for a 2-bit adder

// Dataflow Systemverilog module for a 1-bit fulladder

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

// Testbench module for the dataflow 1-bit fulladder module

```
module testbench1();
    logic x, y, cin;
    logic sum, cout;
    one_bit_full_adder_dataflow dut(x, y, cin, sum, cout);
    initial begin
        x = 0; y = 0; cin = 0; #10;
        if ( sum !== 0 & cout !== 0) &display ("000 failed");
        cin = 1; #10;
        if ( sum !== 1 & cout !== 0) &display ("001 failed");
        y = 1; cin = 0; #10;
        if ( sum !== 1 & cout !== 0) &display ("010 failed");
        cin = 1; #10;
        if ( sum !== 0 & cout !== 1) &display ("011 failed");
        x = 1; y = 0; cin = 0; #10;
        if ( sum !== 1 & cout !== 0) &display ("100 failed");
        cin = 1; #10;
        if ( sum !== 1 & cout !== 0) &display ("101 failed");
        y = 1; cin = 0; #10;
        if ( sum !== 1 & cout !== 0) &display ("110 failed");
        cin = 1; #10;
        if ( sum !== 1 & cout !== 1) &display ("111 failed");
    end
endmodule
```

// Structural Systemverilog module for a 1-bit fulladder

```
module xor2(input logic a, b, output logic c);
    assign c = a ^ b;
endmodule
```

```
module and2(input logic a, b, output logic c);
    assign c = a & b;
endmodule
```

```
module or2(input logic a, b, output logic c);
    assign c = a | b;
endmodule
```

```

module one_bit_full_adder_structural (input logic x, y, cin, output logic sum, cout);
    logic l1, l2, l3;
    xor2 xor_gate (x, y, l1);
    xor2 xor_gate2(l1, cin, sum);
    and2 and_gate(l1, cin, l2);
    and2 and_gate2(x, y, l3);
    or2 or_gate(l2, l3, cout);
endmodule

```

// Testbench module for the structural 1-bit fulladder module

```

module testbench2();
    logic x, y, cin;
    logic sum, cout;
    one_bit_full_adder_structural dut(x, y, cin, sum, cout);
    initial begin
        x = 0; y = 0; cin = 0; #10;
        if ( sum !== 0 & cout !== 0) &display ("000 failed");
        cin = 1; #10;
        if ( sum !== 1 & cout !== 0) &display ("001 failed");
        y = 1; cin = 0; #10;
        if ( sum !== 1 & cout !== 0) &display ("010 failed");
        cin = 1; #10;
        if ( sum !== 0 & cout !== 1) &display ("011 failed");
        x = 1; y = 0; cin = 0; #10;
        if ( sum !== 1 & cout !== 0) &display ("100 failed");
        cin = 1; #10;
        if ( sum !== 1 & cout !== 0) &display ("101 failed");
        y = 1; cin = 0; #10;
        if ( sum !== 1 & cout !== 0) &display ("110 failed");
        cin = 1; #10;
        if ( sum !== 1 & cout !== 1) &display ("111 failed");
    end
endmodule

```

// Structural Systemverilog module for a 2-bit adder

```

module two_bit_adder (input logic x0, x1, y0, y1, cin, output logic sum0, sum1, cout);
    logic a;
    one_bit_full_adder_structural full_adder(x0, y0, cin, sum0, a);
    one_bit_full_adder_structural full_adder2(x1, y1, a, sum1, cout);
endmodule

```

// Testbench module for 2-bit adder module

```
module testbench3();
    logic x0, x1, y0, y1, cin;
    logic sum0, sum1, cout;
    two_bit_adder dut(x0, x1, y0, y1, cin, sum0, sum1, cout);
    initial begin
        x0 = 0; x1 = 0; y0 = 0; y1 = 0; cin = 0; #10;
        if ( sum0 !== 0 & sum1 !== 0 & cout !== 0) &display ("00000 failed");
        cin = 1; #10;
        if ( sum0 !== 1 & sum1 !== 0 & cout !== 0) &display ("00001 failed");
        y1 = 1; cin = 0; #10;
        if ( sum0 !== 0 & sum1 !== 1 & cout !== 0) &display ("00010 failed");
        cin = 1; #10;
        if ( sum0 !== 1 & sum1 !== 1 & cout !== 0) &display ("00011 failed");
        y0 = 1; y1 = 0; cin = 0; #10;
        if ( sum0 !== 1 & sum1 !== 0 & cout !== 0) &display ("00100 failed");
        cin = 1; #10;
        if ( sum0 !== 0 & sum1 !== 1 & cout !== 0) &display ("00101 failed");
        y1 = 1; cin = 0; #10;
        if ( sum0 !== 1 & sum1 !== 1 & cout !== 0) &display ("00110 failed");
        cin = 1; #10;
        if ( sum0 !== 0 & sum1 !== 0 & cout !== 1) &display ("00111 failed");
        x1 = 1; y0 = 0; y1 = 0; cin = 0; #10;
        if ( sum0 !== 0 & sum1 !== 1 & cout !== 0) &display ("01000 failed");
        cin = 1; #10;
        if ( sum0 !== 1 & sum1 !== 1 & cout !== 0) &display ("01001 failed");
        y1 = 1; cin = 0; #10;
        if ( sum0 !== 0 & sum1 !== 0 & cout !== 1) &display ("01010 failed");
        cin = 1; #10;
        if ( sum0 !== 1 & sum1 !== 0 & cout !== 1) &display ("01011 failed");
        y0 = 1; y1 = 0; cin = 0; #10;
        if ( sum0 !== 1 & sum1 !== 1 & cout !== 0) &display ("01100 failed");
        cin = 1; #10;
        if ( sum0 !== 0 & sum1 !== 0 & cout !== 1) &display ("01101 failed");
        y1 = 1; cin = 0; #10;
        if ( sum0 !== 1 & sum1 !== 0 & cout !== 1) &display ("01110 failed");
        cin = 1; #10;
        if ( sum0 !== 0 & sum1 !== 0 & cout !== 1) &display ("01111 failed");
        x0 = 1; x1 = 0; y0 = 0; y1 = 0; cin = 0; #10;
        if ( sum0 !== 1 & sum1 !== 0 & cout !== 0) &display ("10000 failed");
        cin = 1; #10;
        if ( sum0 !== 0 & sum1 !== 1 & cout !== 0) &display ("10001 failed");
        y1 = 1; cin = 0; #10;
        if ( sum0 !== 1 & sum1 !== 1 & cout !== 0) &display ("10010 failed");
```

```

    cin = 1; #10;
    if ( sum0 != 0 & sum1 != 0 & cout != 1) &display ("10011 failed");
    y0 = 1; y1 = 0; cin = 0; #10;
    if ( sum0 != 0 & sum1 != 1 & cout != 0) &display ("10100 failed");
    cin = 1; #10;
    if ( sum0 != 1 & sum1 != 1 & cout != 0) &display ("10101 failed");
    y1 = 1; cin = 0; #10;
    if ( sum0 != 0 & sum1 != 0 & cout != 1) &display ("10110 failed");
    cin = 1; #10;
    if ( sum0 != 1 & sum1 != 0 & cout != 1) &display ("10111 failed");
    x1 = 1; y0 = 0; y1 = 0; cin = 0; #10;
    if ( sum0 != 1 & sum1 != 1 & cout != 0) &display ("11000 failed");
    cin = 1; #10;
    if ( sum0 != 0 & sum1 != 0 & cout != 1) &display ("11001 failed");
    y1 = 1; cin = 0; #10;
    if ( sum0 != 1 & sum1 != 0 & cout != 1) &display ("11010 failed");
    cin = 1; #10;
    if ( sum0 != 0 & sum1 != 1 & cout != 1) &display ("11011 failed");
    y0 = 1; y1 = 0; cin = 0; #10;
    if ( sum0 != 0 & sum1 != 0 & cout != 1) &display ("11100 failed");
    cin = 1; #10;
    if ( sum0 != 1 & sum1 != 0 & cout != 1) &display ("11101 failed");
    y1 = 1; cin = 0; #10;
    if ( sum0 != 0 & sum1 != 0 & cout != 1) &display ("11110 failed");
    cin = 1; #10;
    if ( sum0 != 1 & sum1 != 1 & cout != 1) &display ("11111 failed");

end
endmodule

```