Course Code: CS223

Course Name: Digital Design

Section: 1

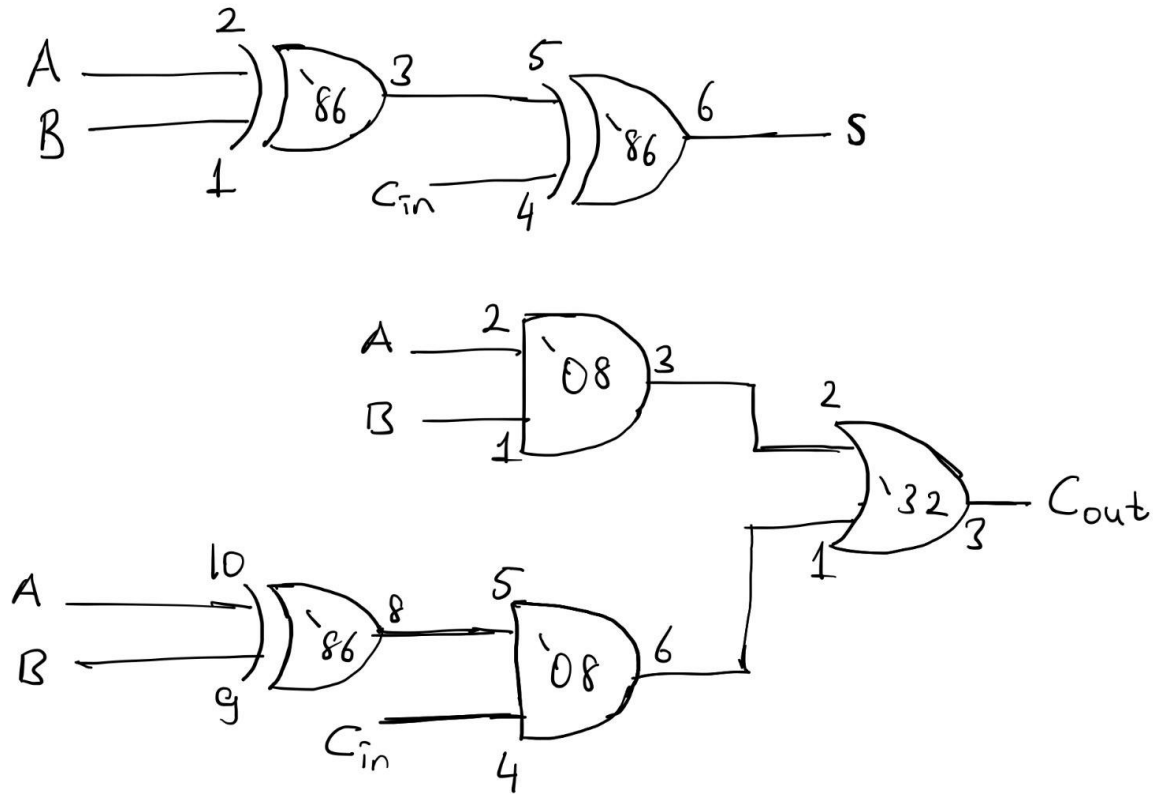
Lab 2

Mustafa Mert Gülhan

22201895

Date: 6.03.2024

Full Adder Circuit Schematic



IC List:

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

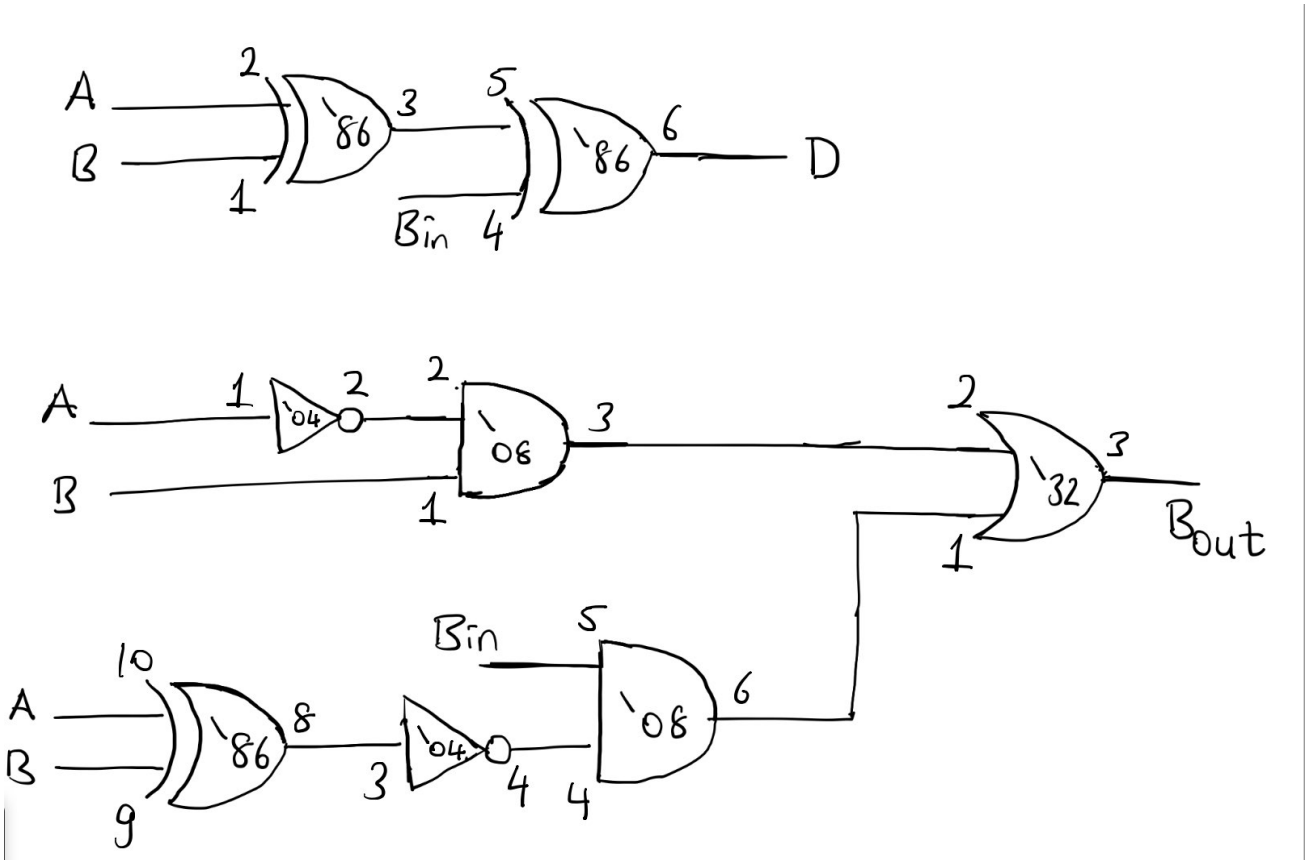
Pin Numbers for GND and Vcc:

7486 -> GND-7 Vcc-14

7408 -> GND-7 Vcc-14

7432 -> GND-7 Vcc-14

Full Substractor Circuit Schematic



IC List:

- One 7486 Quad 2-input XOR gate
- One 7408 Quad 2-input AND gate
- One 7432 Quad 2-input OR gate
- One 7404 Hex Inverter Gate

Pin Numbers For GND and Vcc:

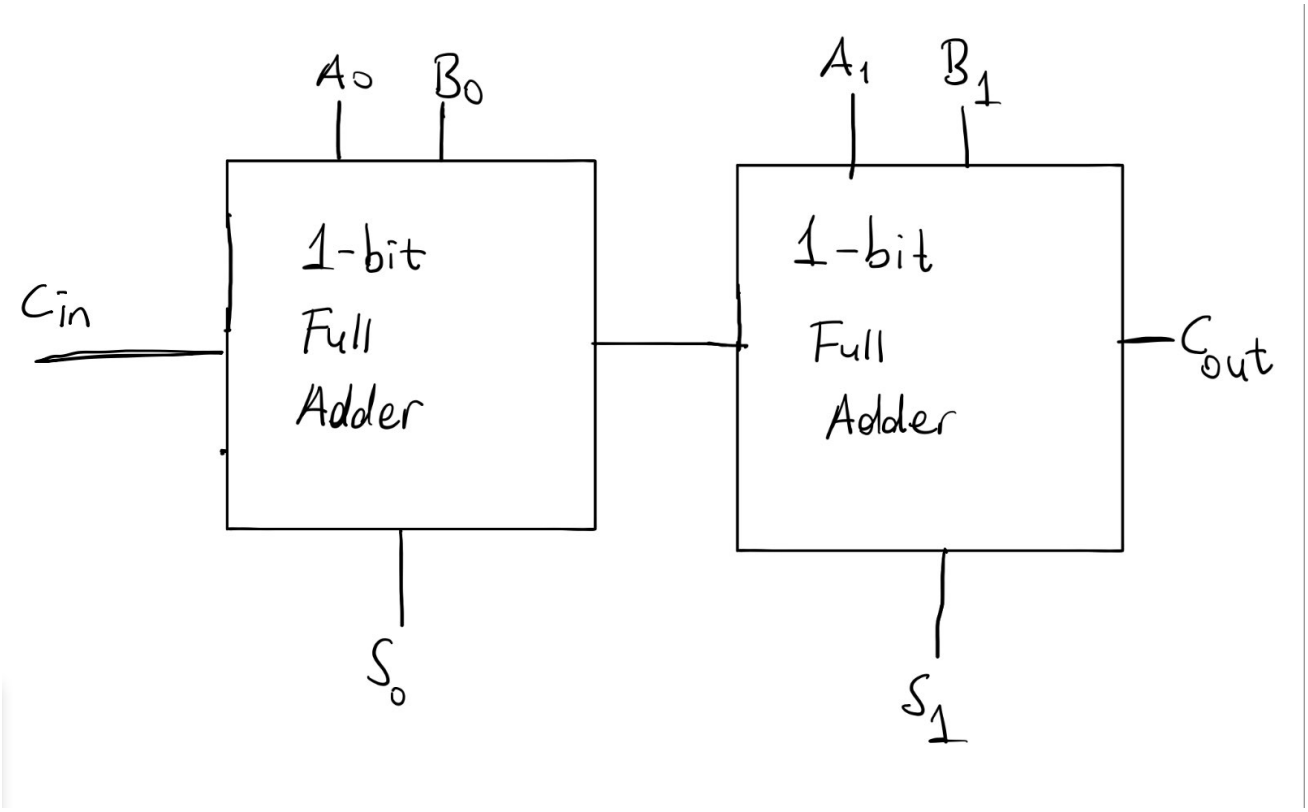
7486 -> GND-7 Vcc-14

7408 -> GND-7 Vcc-14

7432 -> GND-7 Vcc-14

7404 -> GND-7 Vcc-14

2-Bit Adder Circuit Schematic



IC List:

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

Pin Numbers for GND and Vcc:

7486 -> GND-7 Vcc-14

7408 -> GND-7 Vcc-14

7432 -> GND-7 Vcc-14

Behavioral Full Adder Module:

```
module fulladder_behavioral(input logic a,b,cin,
output logic sum, cout);
    assign sum = (a ^ b) ^ cin;
    assign cout = (a & b) | ((a ^ b) & cin);
endmodule
```

Behavioral Full Adder Testbench:

```
module fulladder_behavioral_tb();
    logic a,b,cin,s,cout;
    fulladder_behavioral dut(a,b,cin,s,cout);
    initial begin
        a=0; b=0; cin=0; #20;
        a=0; b=0; cin=1; #20;
        a=1; b=0; cin=0; #20;
        a=1; b=0; cin=1; #20;
        a=0; b=1; cin=0; #20;
        a=0; b=1; cin=1; #20;
        a=1; b=1; cin=0; #20;
        a=1; b=1; cin=1; #20;
    end

endmodule
```

Structural Full Adder Module:

```
module fulladder_structural(input logic a,b,cin,
output logic sum,cout);
    logic w1,w2,w3;
    xor(w1,a,b);
    xor(sum,w1,cin);

    and(w2,a,b);
    and(w3,w1,cin);
    or(cout,w2,w3);
endmodule
```

Structural Full Adder Testbench:

```
module fulladderstructural_tb();
logic a,b,cin,s,cout;
    fulladder_structural dut(a,b,cin,s,cout);
    initial begin
        a=0; b=0; cin=0; #20;
        a=0; b=0; cin=1; #20;
        a=1; b=0; cin=0; #20;
        a=1; b=0; cin=1; #20;
        a=0; b=1; cin=0; #20;
```

```
        a=0; b=1; cin=1; #20;
        a=1; b=1; cin=0; #20;
        a=1; b=1; cin=1; #20;
    end
endmodule
```

Behavioral Full Substractor Module:

```
module fullsubstractor_behavioral(input logic
    a,b,bin, output logic d,bout);
    assign d = (a ^ b) ^ bin;
    assign bout = ((~a) & b ) | ((~(a ^ b)) &
    bin);
endmodule
```

Behavioral Full Substractor Testbench:

```
module fullsubstractor_behavioral_tb();
    logic a,b,bin,d,bout;
    fullsubstractor_behavioral dut(a,b,bin,d,bout);
    initial begin
        a=0; b=0; bin=0; #20;
        bin=1; #20;
        b=1; bin=0; #20;
        bin=1; #20;
        a=1; b=0; bin=0; #20;
```

```
        bin=1; #20;
        b=1; bin=0; #20;
        bin=1; #20;
    end
endmodule
```

Structural Full Substractor Module:

```
module fullsubstractor_structural(input logic
a,b,bin, output logic d,bout);
    logic w1,w2,w3,w4,w5;
    xor(w1,a,b);
    xor(d,bin,w1);

    not(w2,a);
    and(w3,w2,b);
    not(w4,w1);
    and(w5,w4,bin);
    or(bout,w5,w3);
endmodule
```

Structural Full Substractor Testbench:

```
module fullsubstractor_structural_tb();
    logic a,b,bin,d,bout;
```

```
    fullsubstractor_structural dut(a,b,bin,d,bout);
    initial begin
        a=0; b=0; bin=0; #20;
        bin=1; #20;
        b=1; bin=0; #20;
        bin=1; #20;
        a=1; b=0; bin=0; #20;
        bin=1; #20;
        b=1; bin=0; #20;
        bin=1; #20;
    end
endmodule
```

Structural 2-bit Adder Module:

```
module twobitadder_structural(input logic
a0,b0,a1,b1,cin, output logic s0,s1,cout);
    logic w1;
    fulladder_structural first(a0,b0,cin,s0,w1);
    fulladder_structural second(a1,b1,w1,s1,cout);
endmodule
```

Structural 2-bit Adder Testbench:

```
module twobitadder_structural_tb();
```

```
    logic a0,b0,a1,b1,cin,s0,s1,cout;
    twobitadder_structural
dut(a0,b0,a1,b1,cin,s0,s1,cout);
    initial begin
        a0=0; b0=0; a1=0; b1=0; cin=0; #20;
        a0=0; b0=0; a1=0; b1=0; cin=1; #20;
        a0=1; b0=0; a1=0; b1=0; cin=0; #20;
        a0=1; b0=0; a1=0; b1=0; cin=1; #20;
        a0=1; b0=0; a1=1; b1=0; cin=0; #20;
        a0=1; b0=0; a1=1; b1=0; cin=1; #20;
        a0=0; b0=0; a1=1; b1=0; cin=0; #20;
        a0=0; b0=0; a1=1; b1=0; cin=1; #20;
        a0=0; b0=1; a1=0; b1=0; cin=0; #20;
        a0=0; b0=1; a1=0; b1=0; cin=1; #20;
        a0=0; b0=0; a1=0; b1=1; cin=0; #20;
        a0=0; b0=0; a1=0; b1=1; cin=1; #20;
        a0=0; b0=1; a1=0; b1=1; cin=0; #20;
        a0=0; b0=1; a1=0; b1=1; cin=1; #20;
        a0=1; b0=1; a1=0; b1=0; cin=0; #20;
        a0=1; b0=1; a1=0; b1=0; cin=1; #20;
        a0=1; b0=0; a1=0; b1=1; cin=0; #20;
        a0=1; b0=0; a1=0; b1=1; cin=1; #20;
        a0=0; b0=1; a1=1; b1=0; cin=0; #20;
        a0=0; b0=1; a1=1; b1=0; cin=1; #20;
        a0=0; b0=0; a1=1; b1=1; cin=0; #20;
```

```
        a0=0; b0=0; a1=1; b1=1; cin=1; #20;
        a0=1; b0=1; a1=1; b1=0; cin=0; #20;
        a0=1; b0=1; a1=1; b1=0; cin=1; #20;
        a0=1; b0=0; a1=1; b1=1; cin=0; #20;
        a0=1; b0=0; a1=1; b1=1; cin=1; #20;
        a0=1; b0=1; a1=1; b1=1; cin=0; #20;
        a0=1; b0=1; a1=1; b1=1; cin=1; #20;
        a0=1; b0=1; a1=0; b1=1; cin=0; #20;
        a0=1; b0=1; a1=0; b1=1; cin=1; #20;
        a0=0; b0=1; a1=1; b1=1; cin=0; #20;
        a0=0; b0=1; a1=1; b1=1; cin=1; #20;
    end
endmodule
```