CS 223 DIGITAL DESIGN

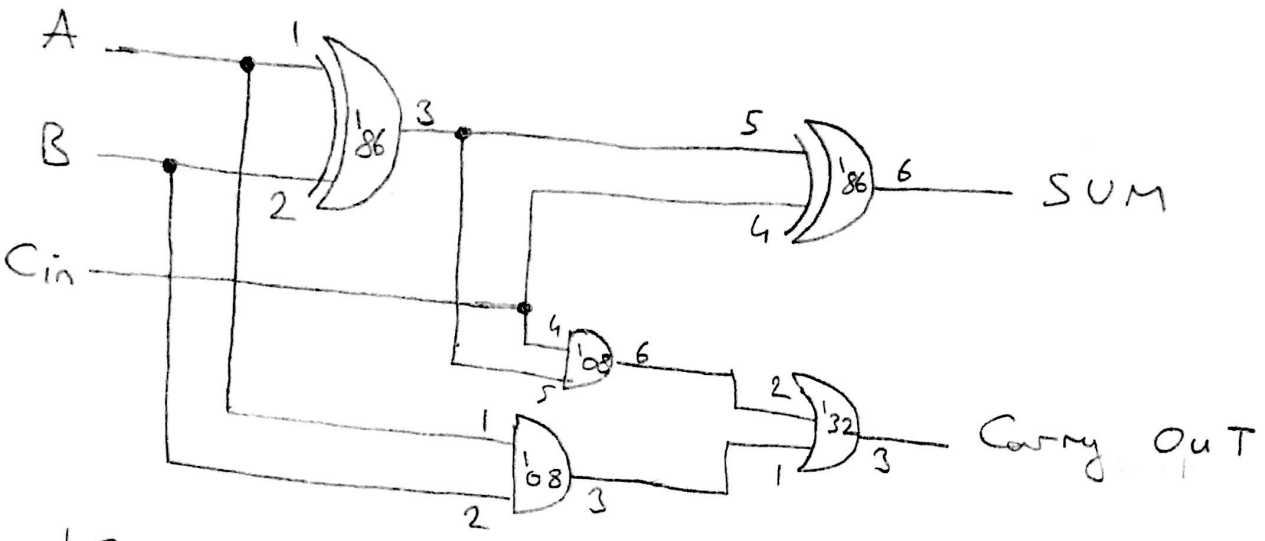
SECTION 5 LAB 2

ÖMER OKTAY GULTEKIN

21901413

18.10.2021

* FULL-ADDER CIRCUIT SCHEMATICS :



IC LIST

(1) One IC-7408 QUAD 2-input AND Gate

(2) One IC-7432 QUAD 2-input OR Gate

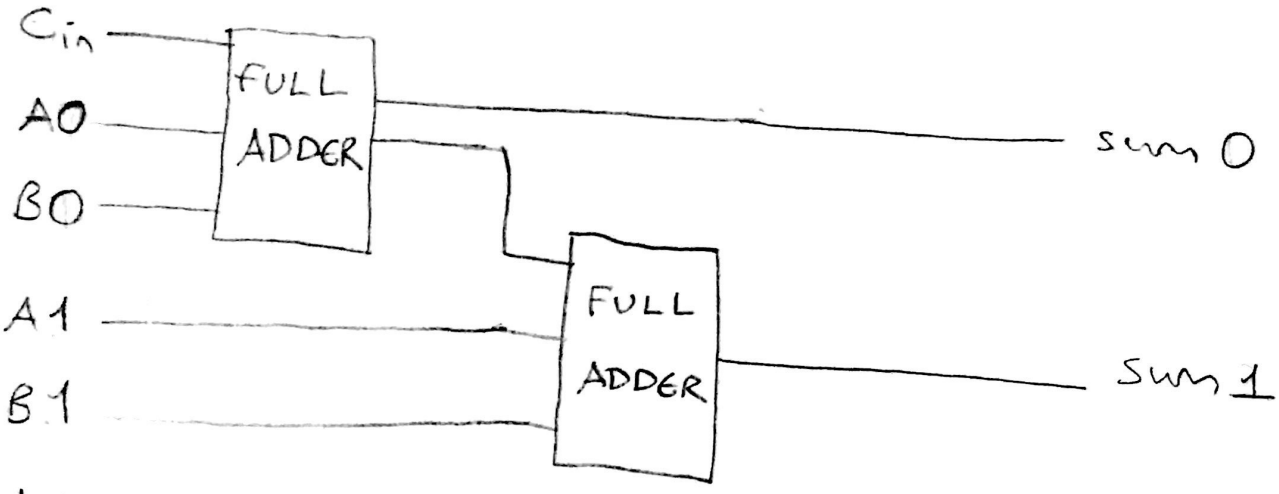
(3) One IC-7486 QUAD 2-input X-OR Gate

* 74HC86
GND-7
Vcc-14

* 74HC08
GND-7
Vcc-14

* 74HC32
GND-7
Vcc-14

* TWO-BIT ADDER CIRCUIT SCHEMATICS:



IC LIST

① One IC-7408 AND Gate
② One IC-7432 OR Gate
③ One IC-7486 X-OR Gate

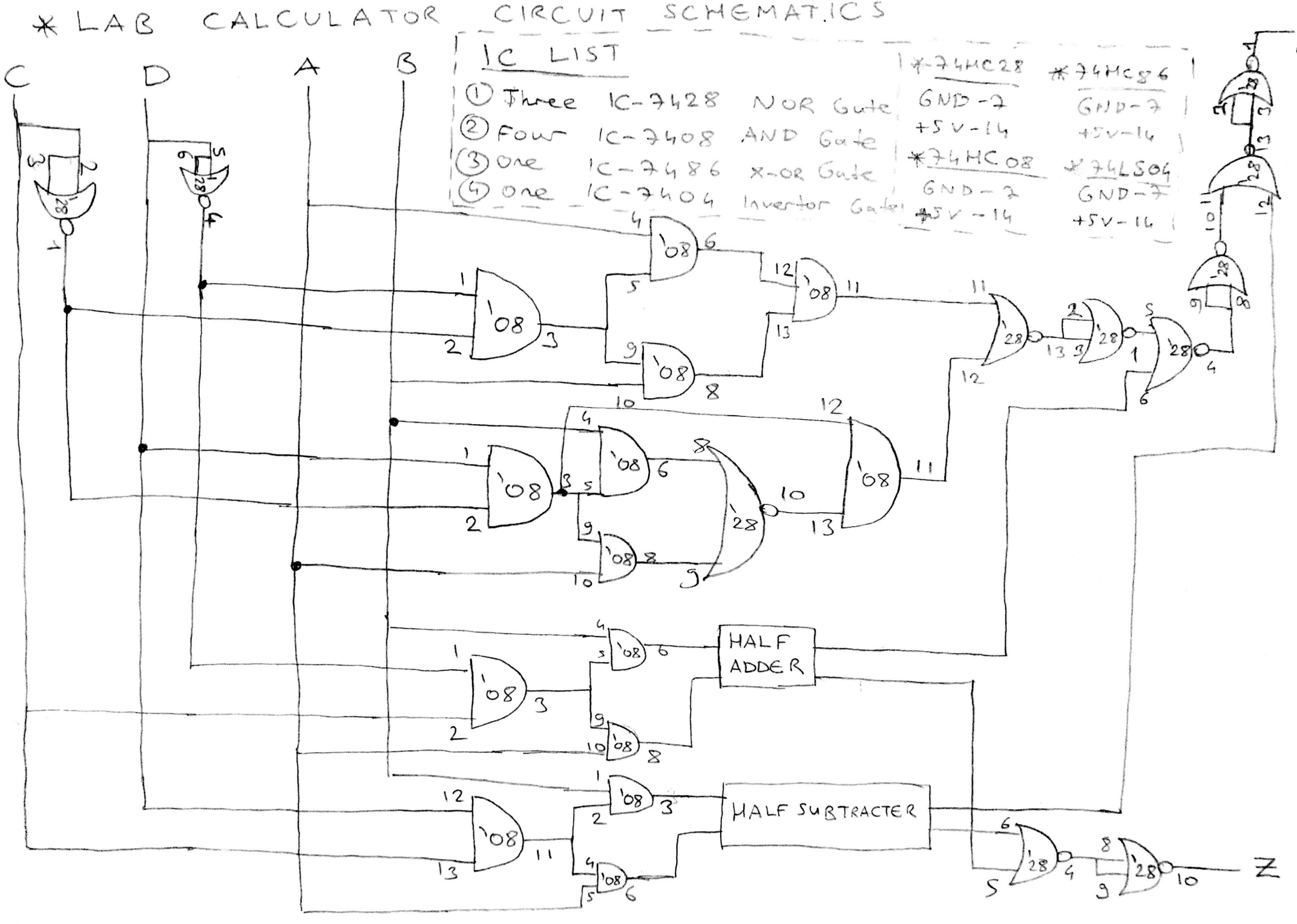
} Each have 4 corresponding gates

* 74HC86
GND(0V) - 7
V_{cc} (+5V) - 14

* 74HC08
GND (0V) - 7
V_{cc} (+5V) - 14

* 74HC32
GND (0V) - 7
V_{cc} (+5V) - 14

* LAB CALCULATOR CIRCUIT SCHEMATICS



* Behavioral SystemVerilog for the full adder and a testbench for it

```
`timescale 1ns/1ps
module fulladder(
    input logic a, b, cin,
    output logic s, cout
    );

    logic p, g;
    assign p = a ^ b;
    assign g = a & b;
    assign s = p ^ cin;
    assign cout = g | (p & cin);
endmodule
```

- Test Bench:

```
`timescale 1ns / 1ps

module testbench1();
    logic a, b, cin;
    logic s, cout;
    fulladder adder(a, b, cin, s, cout);
    initial begin
        a=0; b=0; cin=0; #10;
        if (s !==0 & cout !==0) $display("000 failed");
        cin = 1; #10;
        if (s !== 1 & cout !=0) $display("001 failed");
        b=1; #10;
        cin=0; #10;
        a=1; #10;
        b=0; #10;
        cin=1; #10;
        b=1; #10;
    end
endmodule
```

→ I will omit if's

* Structural Systemverilog Module For The FULL Adder And Testbench for It

```
`timescale 1ns/1ps
Module structuralfulladder(
    input logic a, b, cin,
    output logic S, Cout
    );
    logic n1, n2, n3;
    halfadder ha1(a, b, n1, n2);
    halfadder ha2(n1, cin, s, n3);
    Or2 or1(n2, n3, cout);
endmodule
Module halfadder(input logic a, b, output logic sum, cout);
    and2 and1(a, b, cout);
    xor2 xor1(a, b, sum);
endmodule
module and2(input logic a, b, output logic c);
    assign c = a & b;
endmodule
Module xor2(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
```

* Testbench

```
Module testbench2();
    logic a, b, cin;
    logic s, cout;
    structubor-fulladder adder(a, b, cin, s, cout);
    initial begin
        a=0; b=0; cin=0; #10;
        cin=1; #10;
        b=1; #10;
        cin=0; #10;
        a=1; #10;
        b=0; #10;
        cin=1; #10;
        b=1; #10;
endmodule
```

* Structural systemverilog module for the 2-bit adder and testbench for it using the full adder module

```
`timescale 1ns / 1ps

module adder2bit(
    input logic a0, a1, b0, b1, cin,
    output logic s0, s1, cout
    );

    logic n1;

    structuralfulladder fulladder1(a0, b0, cin, s0, n1);
    structuralfulladder fulladder2(a1, b1, n1, s1, cout);
endmodule
```

• Testbench

```
`timescale 1ns / 1ps

module testbench3();
    logic a0, b0, a1, b1, cin;
    logic s0, s1, cout;
    adder2bit adder(a0, a1, b0, b1, cin, s0, s1, cout);
    initial begin
        a=0; a1=0; b0=0; b1=0; cin=0; #10;
f   { cin=1; #10
      b1=1; #10
      cin=0; #10
      b0=1; #10
      cin=1; #10
      b1=0; #10
      cin=0; #10
    a1=1; b=0; #10;
      f → same as the above code block
    a0=1; b0=0; #10
      f
    a1=0; b0=0; #10
      f
    end
endmodule
```

*Structural Systemverilog Module For The Lab Calculator And a Test Bench for It

```
`timescale 1ns / 1ps
module labcalculator(
    input logic a, b, c, d,
    output logic y, z);
    logic n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14,
    n15, n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27;

    nor2 NOR1(c, c, n1);
    nor2 NOR2(d, d, n2);
    and2 AND1(n1, n2, n3);
    and2 AND2(n3, a, n4);
    and2 AND3(n3, b, n5);
    and2 AND4(n4, n5, n6);
    and2 AND5(n1, d, n7);
    and2 AND6(n7, a, n8);
    and2 AND7(n7, b, n9);
    nor2 NOR3(n8, n9, n10);
    and2 AND8(n7, n10, n11);
    nor2 NOR4(n6, n11, n12);
    nor2 NOR5(n12, n12, n18);
    and2 AND9(n2, c, n13);
    and2 AND10(n13, a, n14);
    and2 AND11(n13, b, n15);
    halfadder adder(n14, n15, n16, n17);
    nor2 NOR6(n16, n18, n19);
    nor2 NOR7(n19, n19, n20);
    and2 AND12(c, d, n21);
    and2 AND13(n21, a, n22);
    and2 AND14(n21, b, n23);
    halfsubt subt(n22, n23, n24, n25);
    nor2 NOR8(n20, n24, n26);
    nor2 NOR9(n26, n26, y);
    nor2 NOR10(n17, n25, n27);
endmodule
module nor2(input logic a, b, output logic c);
    assign c = ~(a|b);
endmodule
module halfsubt(input logic a, b, output logic d, bout);
    assign d = a ^ b;
    assign bout = ~a & b;
endmodule
```

• TESTBENCH

```
`timescale 1ns / 1ps

module testbench4();
    logic a, b, c, d;
    logic y, z;
    lab calculator calculator(a, b, c, d, y, z);
    initial begin
        a = 0; b = 0; c = 0; d = 0; #10;
        d = 1; #10;
        c = 1; #10;
        d = 0; #10;
        b = 1; c = 0; #10;
        d = 1; #1;
        c = 1; #1;
        d = 0; #1;
        a = 1; c = 0; #10
        d = 1; #10;
        c = 1; #10;
        d = 0; #10;
        b = 0; c = 0; #10;
        d = 1; #10;
        c = 1; #10;
        d = 0; #10;
    end
endmodule
```