DIGITAL DESIGN 2nd LAB

PRELIMINARY REPORT

Course Code: CS 223

Course Name: Digital Design

Section: 1

Name: Arda

Surname: İynem

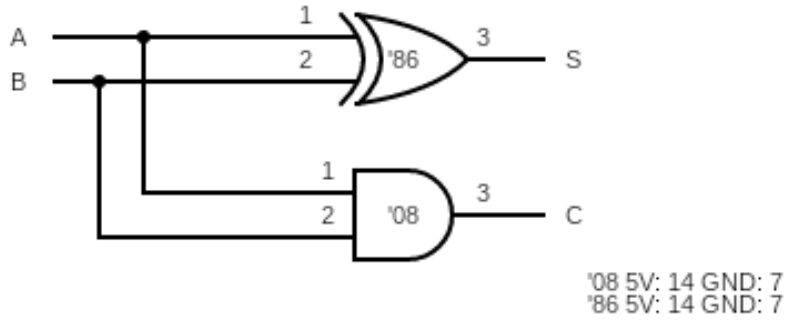
Student ID: 22002717

Date: 27.02.2022

HALF ADDER

CIRCUIT SCHEMATIC
BEHAVIORAL MODULE
TESTBENCH

(b) Circuit Schematic for Half Adder



(g) Behavioral SystemVerilog Module for Half Adder

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

(g) Testbench for Half Adder

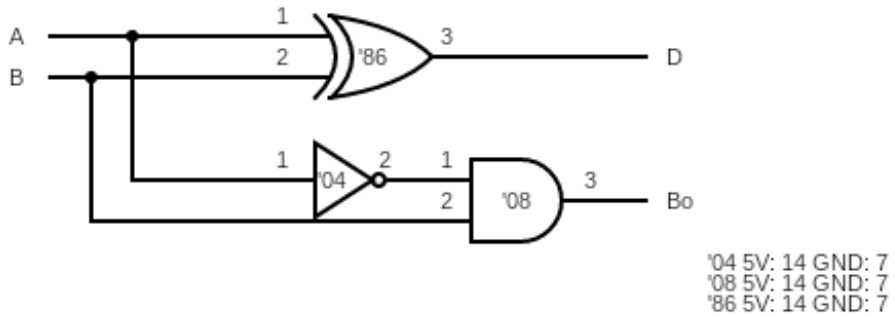
```
module testbench_half_adder();
 logic a,b;
 logic s,c;

 half_adder dut(a,b,s,c);
 initial begin
  a = 0; b = 0; #10;
  b = 1; #10;
  a = 1; b = 0; #10;
  b = 1; #10;
 end
endmodule
```

HALF SUBTRACTOR

CIRCUIT SCHEMATIC
BEHAVIORAL MODULE
TESTBENCH

(c) Circuit Schematic for Half Subtractor



(h) Behavioral SystemVerilog Module for Half Subtractor

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

(h) Testbench for Half Subtractor

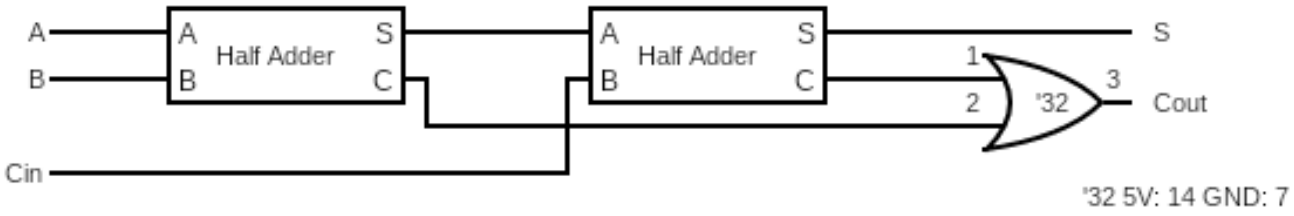
```
module testbench_half_subtractor();
 logic a,b;
 logic d,bout;

 half_subtractor dut(a,b,d,bout);
 initial begin
  a = 0; b = 0; #10;
  b = 1; #10;
  a = 1; b = 0; #10;
  b = 1; #10;
 end
endmodule
```

FULL ADDER (USING HALF ADDERS)

CIRCUIT SCHEMATIC
STRUCTURAL MODULE
TESTBENCH

(d) Circuit Schematic for Full Adder (Using Half Adders)



(i) Structural SystemVerilog Module for Full Adder (Using Half Adders)

```
module full_adder(input logic a,b,cin,
                  output logic s,cout);
 logic s0, c0, c1;

 half_adder ha(a,b,s0,c0);
 half_adder ha2(s0,cin,s,c1);

 assign cout = c0 | c1;
endmodule
```

(i) Testbench for Full Adder

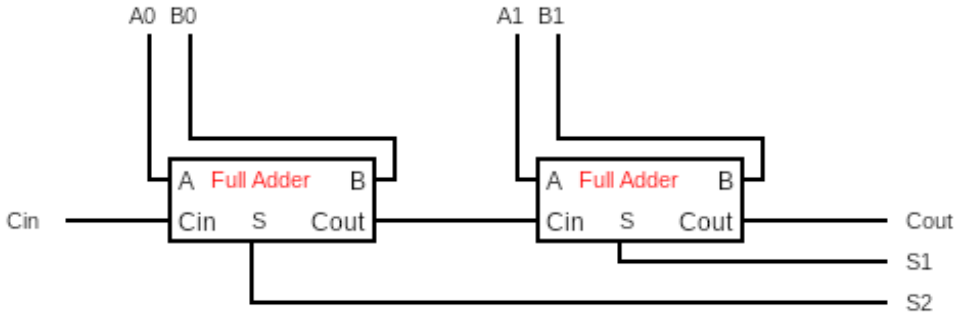
```
module testbench_full_adder();
 logic a,b,cin;
 logic s,cout;

 full_adder dut(a,b,cin,s,cout);
 initial begin
  a = 0; b = 0; cin = 0; #10;
  cin = 1; #10;
  b = 1; cin = 0; #10;
  cin = 1; #10;
  a = 1; b = 0; cin = 0; #10;
  cin = 1; #10;
  b = 1; cin = 0; #10;
  cin = 1; #10;
 end
endmodule
```

2-BIT ADDER (USING FULL ADDERS)

CIRCUIT SCHEMATIC
STRUCTURAL MODULE
TESTBENCH

(e) Circuit Schematic for 2-Bit Adder (Using Full Adders)



(j) Structural SystemVerilog Module for 2-Bit Adder (Using Full Adders)

```
module two_bit_adder(input logic a0,b0,a1,b1,
                     output logic s0,s1,cout);
 logic cout0;

 full_adder fa(a0,b0,0,s0,cout0);
 full_adder fa2(a1,b1,cout0,s1,cout);
endmodule
```

(j) Testbench for 2-Bit Adder

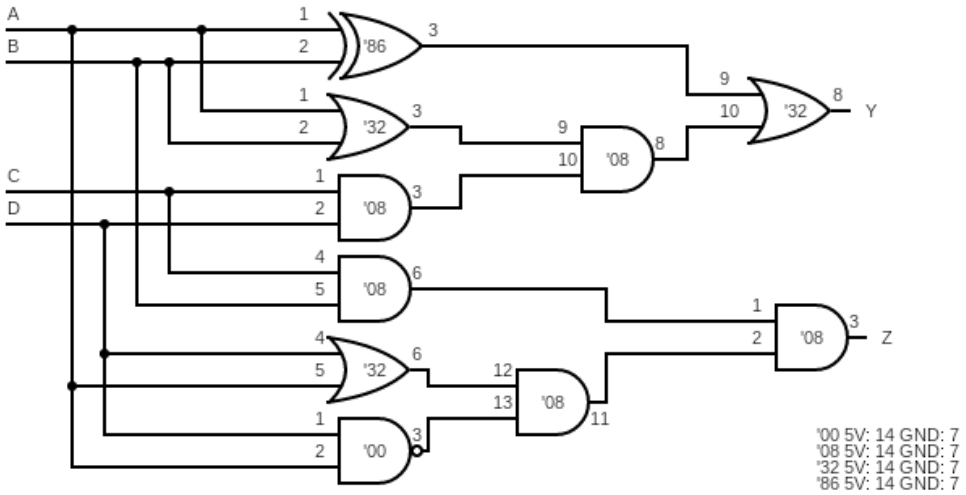
```
module testbench_two_bit_adder();
 logic a0,b0,a1,b1;
 logic s0,s1,cout;

 two_bit_adder dut(a0,b0,a1,b1,s0,s1,cout);
 initial begin
  a0 = 0; b0 = 0; a1 = 0; b1 = 0; #10;
  b1 = 1; #10;
  a1 = 1; b1 = 0; #10;
  b1 = 1; #10;
  b0 = 1; a1 = 0; b1 = 0; #10;
  b1 = 1; #10;
  a1 = 1; b1 = 0; #10;
  b1 = 1; #10;
  a0 = 1; b0 = 0; a1 = 0; b1 = 0; #10;
  b1 = 1; #10;
  a1 = 1; b1 = 0; #10;
  b1 = 1; #10;
  b0 = 1; a1 = 0; b1 = 0; #10
  b1 = 1; #10;
  a1 = 1; b1 = 0; #10;
  b1 = 1; #10;
 end
endmodule
```

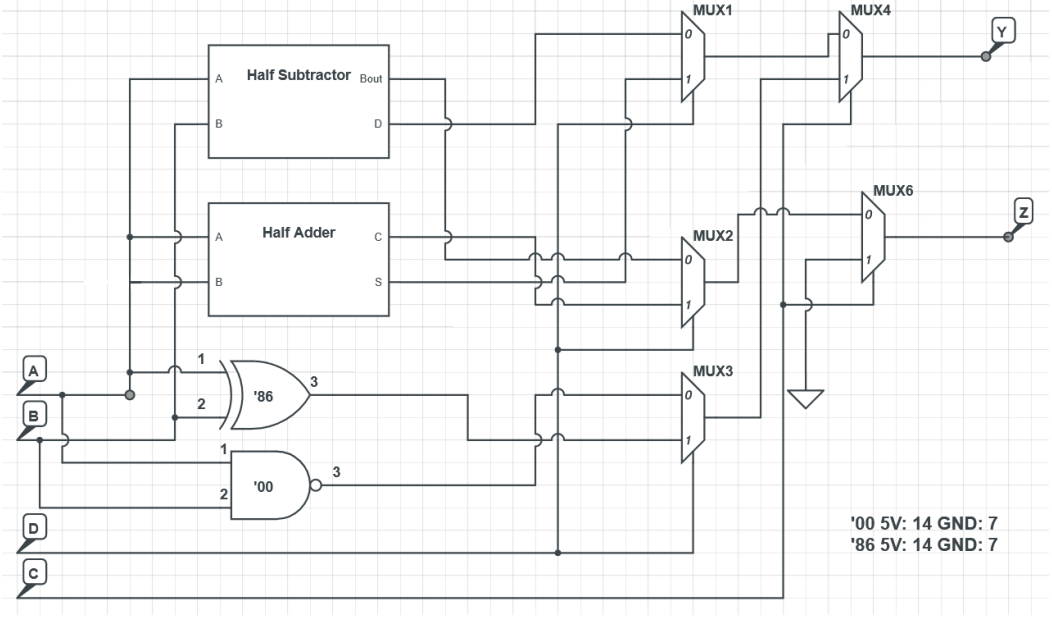
LAB CALCULATOR

CIRCUIT SCHEMATIC

(f) Circuit Schematic for Lab Calculator (Using Gates Only)



(f) Circuit Schematic for Lab Calculator (Using Half Adder/Subtractor and MUX)



LAB CALCULATOR

STRUCTURAL MODULE TESTBENCH

(k) Structural SystemVerilog Module for Lab Calculator (Using Half Adders and Half Subtractors)

```
module lab_calculator(input logic a,b,c,d,
                      output logic y,z);
 logic tmp_s, tmp_cout, tmp_d, tmp_bout;

 half_adder ha(a, b, tmp_s, tmp_cout);
 half_subtractor hs(a, b, tmp_d, tmp_bout);

 assign y = c ? (d ? tmp_d : tmp_s)
              : (d ? ~(a & b) : (a ^ b));
 assign z = c ? (d ? tmp_bout : tmp_cout)
              : (d ? 0 : 0);
endmodule
```

(k) Testbench for Lab Calculator

```
module testbench_lab_calculator();
 logic a,b,c,d;
 logic y,z;

 lab_calculator dut(a,b,c,d,y,z);
 initial begin
  c = 0; d = 0; a = 0; b = 0; #10;
  b = 1; #10;
  a = 1; b = 0; #10;
  b = 1; #10;
  d = 1; a = 0; b = 0; #10;
  b = 1; #10;
  a = 1; b = 0; #10;
  b = 1; #10;
  c = 1; d = 0; a = 0; b = 0; #10;
  b = 1; #10;
  a = 1; b = 0; #10;
  b = 1; #10;
  d = 1; a = 0; b = 0; #10
  b = 1; #10;
  a = 1; b = 0; #10;
  b = 1; #10;
 end
endmodule
```