PROJECT NO. - 9

AIM : To design BCD adder 4-bit and 8-bit

APPARATUS : Bi directional switches ; wires for connection ; IC 74283 ; 7 segment display; OR and AND gate

THEORY :

BCD ADDER :

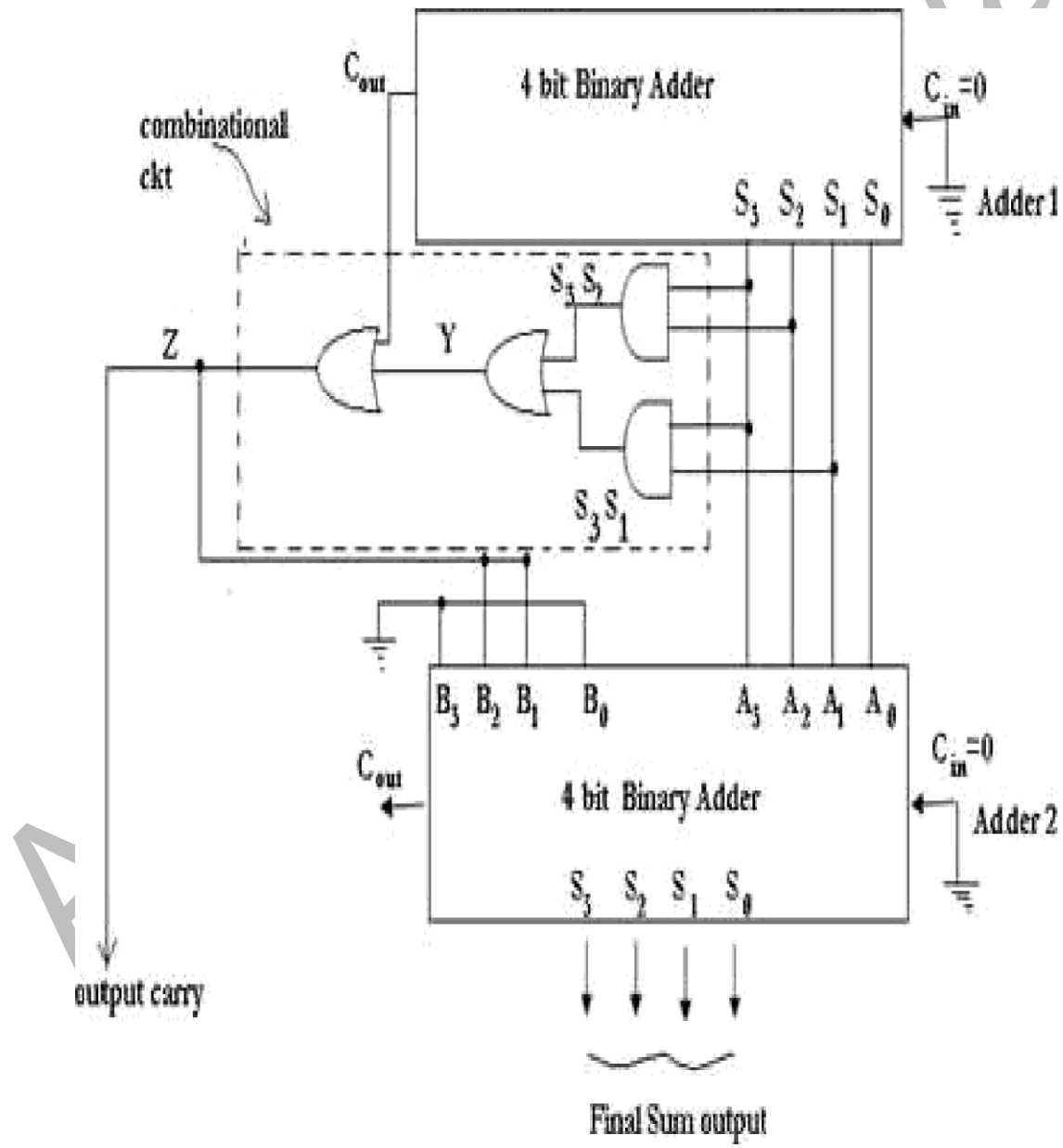
If the sum of two number is less than or equal to 9, then the value of BCD sum and binary sum will be same otherwise they will differ by 6(0110 in binary).
Now, lets move to the table and find out the logic when we are going to add “0110”.
We are adding “0110” (=6) only to the second half of the table.
The conditions are:

1. If C = 1 (Satisfies 16-19)
2. If S3.S2 = 1 (Satisfies 12-15)
3. If S3.S1 = 1 (Satisfies 10 and 11)

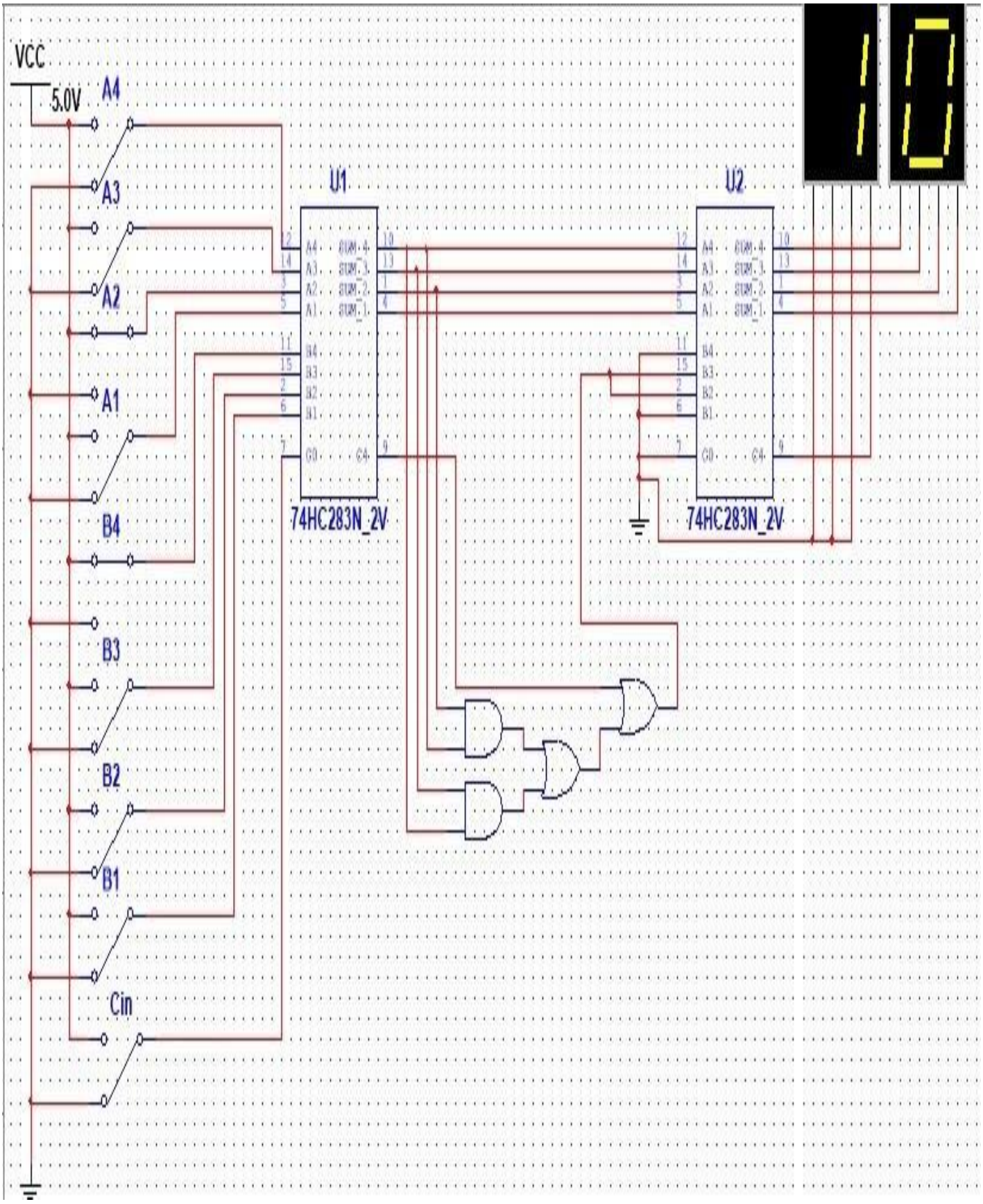
So, our logic is

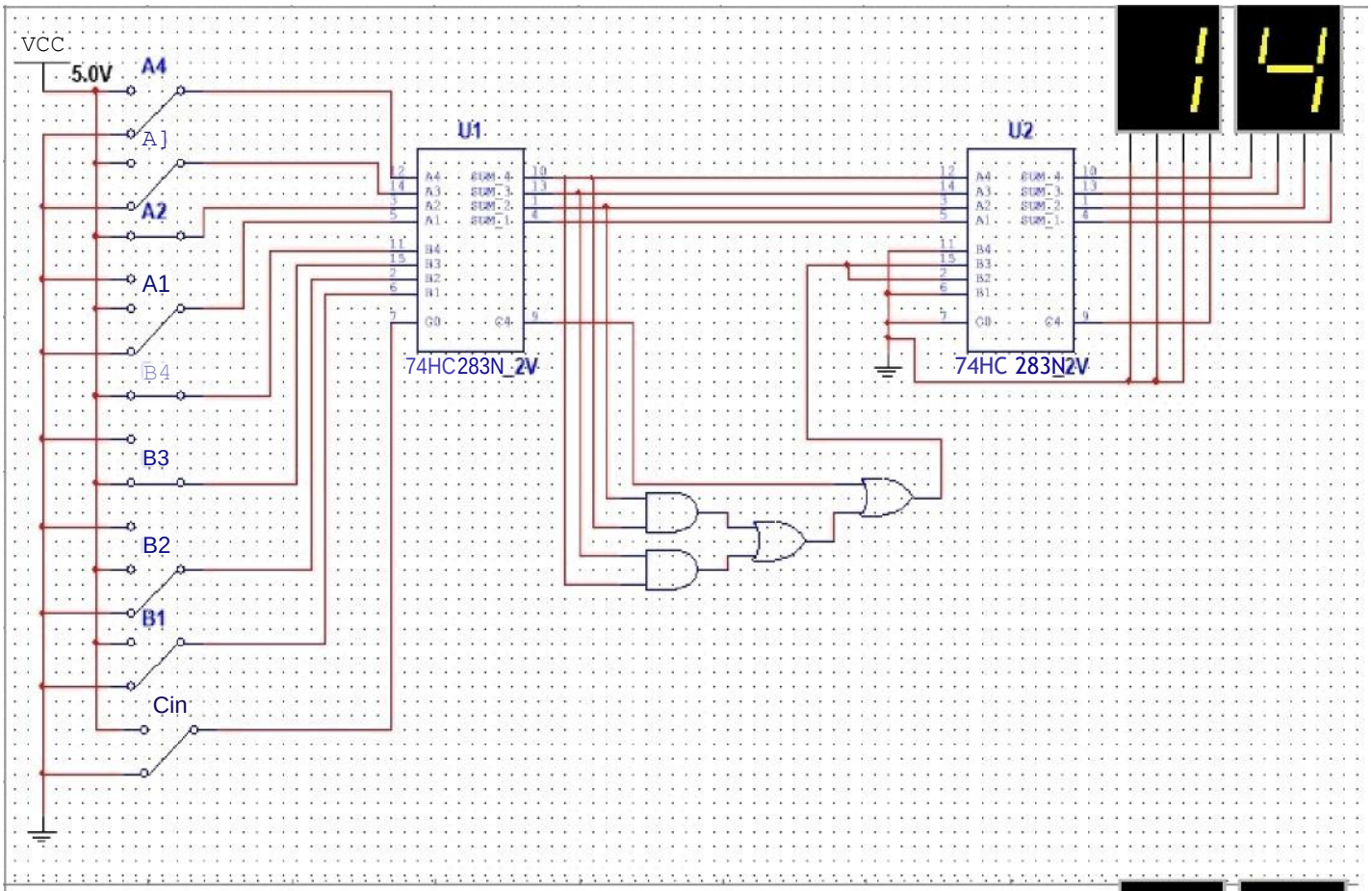
$$C + S3.S2 + S3.S1 = 1$$

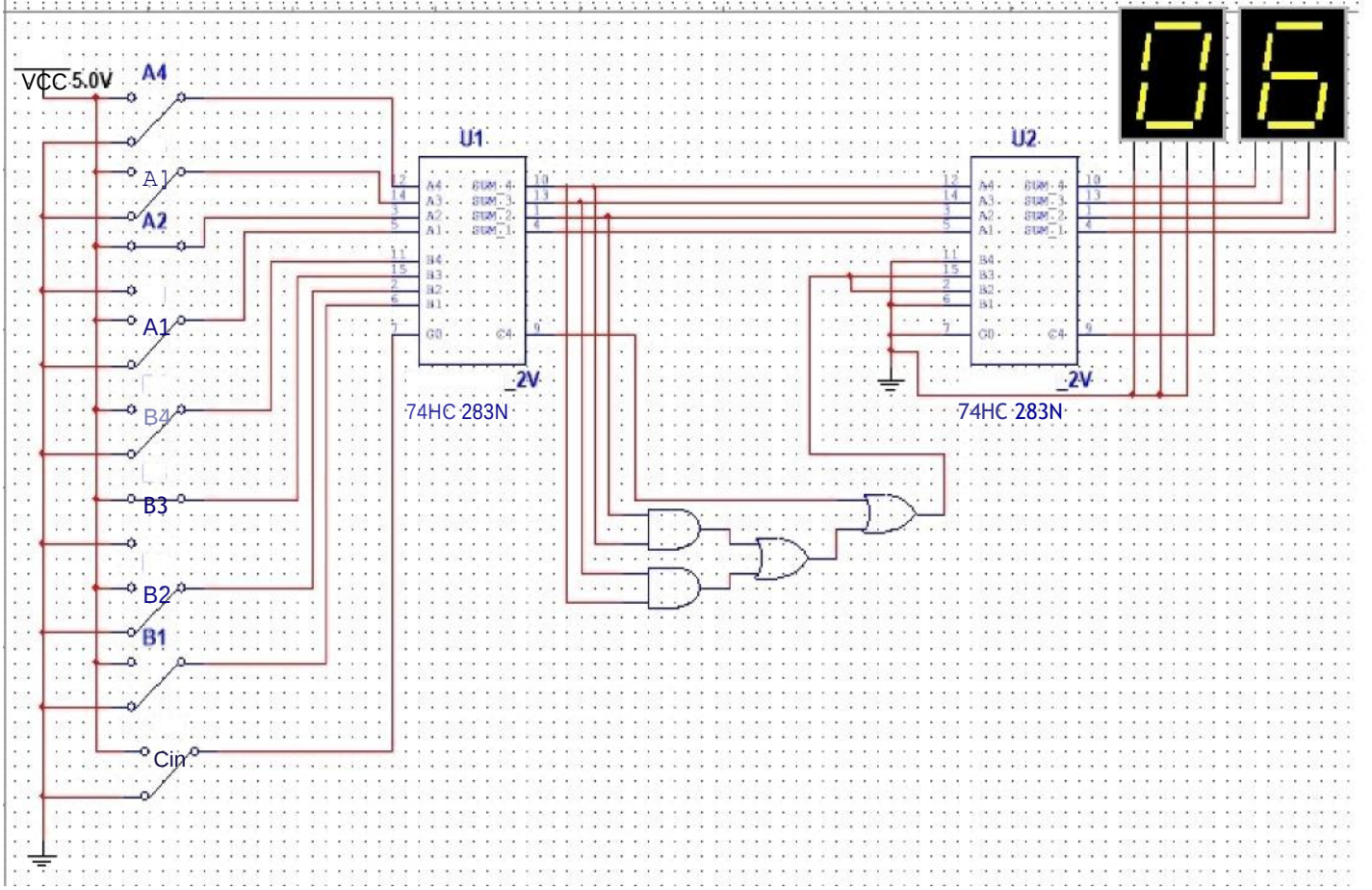
4bitBtDB4*r



Figi: BCD adder


VCC
5.0V
A4
A3
A2
A1
B4
B3
B2
B1
Cin
U1
U2
74HC283N_2V
74HC283N_2V


VCC
5.0V
A4
A3
A2
A1
B4
B3
B2
B1
Cin
U1
U2
74HC283N_2V
74HC 283N2V
1
4

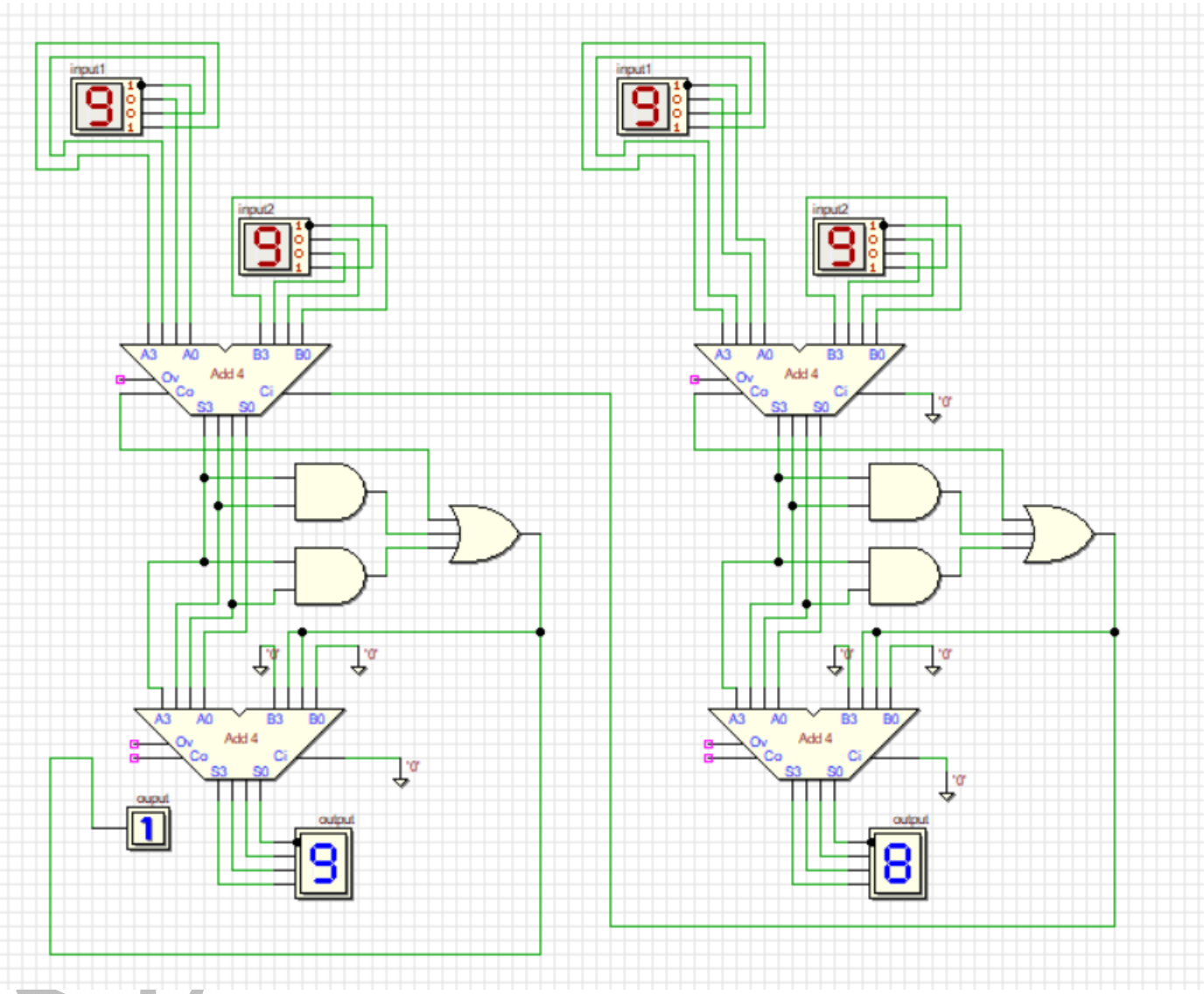

VCC 5.0V
A4
A3
A2
A1
B4
B3
B2
B1
Cin
U1
U2
_2V
74HC 283N
_2V
74HC 283N
0
6

TRUTH TABLE FORN 4 BIT ADDER :

INPUT									OUTPUT				
	A				B				Sum				Carry
Cin	A3	A2	A1	A0	B3	B2	B1	B0	S3	S2	S1	S0	Cout
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	1	0	0	1	0	0	0
0	0	0	1	1	0	0	1	1	0	1	1	0	0
0	0	1	0	0	0	1	0	0	1	0	0	0	0
0	0	1	0	1	0	1	0	1	1	0	1	0	0
0	0	1	1	0	0	1	1	0	1	1	0	0	0
0	0	1	1	1	0	1	1	1	1	1	1	0	0
0	1	0	0	0	1	0	0	0	0	0	0	0	1
0	1	0	0	1	1	0	0	1	0	0	1	0	1
0	1	0	1	0	1	0	1	0	0	1	0	0	1
0	1	0	1	1	1	0	1	1	0	1	1	0	1
0	1	1	0	0	1	1	0	0	1	0	0	0	1
0	1	1	0	1	1	1	0	1	1	0	1	0	1
0	1	1	1	0	1	1	1	0	1	1	0	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Decimal	Binary Sum					BCD Sum				
	C'	S3'	S2'	S1'	S0'	C	S3	S2	S1	S0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	1	0	0	1
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1

CIRCUIT DIAGRAM OF BCD ADDER 8 bit



Verilog Code:

BCD_Adder 4 bit

```
// Developed by: Aashi Srivastava
// Title : 4-bit BCD adder
// Date:16-10-23, 21:10 IST

module bcd_adder_4bit (
    sum,cout,in1,in2,cin, wire6
);
    input [3:0] in1, in2;
    input cin;
    output [3:0] sum;
    output cout;

    wire[3:0] wire1, wire10;
    wire wire2, wire3,wire4,wire5;

    output wire6;

    parallel_adder p1(wire1, wire2, in1, in2, cin);
    and a1(wire3, wire1[3], wire1[2]);
    and a2(wire4, wire1[3], wire1[1]);
    or o1(wire5, wire3, wire4);
    or o2(wire6, wire5, wire2);

    assign wire10[3]=1'b0;
    assign wire10[2]= wire6;
    assign wire10[1]= wire6;
    assign wire10[0]=1'b0;

    parallel_adder p2(sum, cout, wire1, wire10, cin);

endmodule
module parallel_adder (
    sum,carry_out,in1,in2,carry_in
);
    input [3:0] in1, in2;
    output [3:0]sum;
    output carry_out;
    input carry_in;
    wire [2:0] wire1;

    full_adder_CA f1(sum[0],wire1[0],in1[0],in2[0],carry_in);

    full_adder_CA f2(sum[1],wire1[1],in1[1],in2[1],wire1[0]);
```

```
    full_adder_CA f3(sum[2],wire1[2],in1[2],in2[2],wire1[1]);

    full_adder_CA f4(sum[3],carry_out,in1[3],in2[3],wire1[2]);

endmodule

module full_adder_CA(
    sum,carry_out,in1,in2,carry_in
);
input in1,in2, carry_in;
output sum, carry_out;
wire wire1, wire2, wire3;
assign sum=(in1^in2)^carry_in;
assign carry_out=(in1 & in2)|(in2 & carry_in)|(carry_in & in1);
endmodule
```

Test-Bench:

```
module bcd_adder_4bit_tb (

);
    reg [3:0]in1,in2;
    wire [3:0] sum;
    wire carry_out;
    reg carry_in;
    wire wire6;

    bcd_adder_4bit p(
    sum,carry_out,in1,in2,carry_in,wire6
);

initial begin
    $dumpfile("bcd_adder_4bit.vcd");
    $dumpvars(0,bcd_adder_4bit_tb);
    $monitor(" sum=%0d%0d in1=%0d in2=%0d ",wire6,sum,in1,in2);
        #40 $finish;
    end
initial begin
    in1=4'b1001;
    in2=4'b0011;
    carry_in=1'b0;
end
endmodule
```

Verilog Code:

BCD_Adder 8 bit

```
// Developed by: Aashi Srivastava
// Title : 8-bit BCD adder
// Date:16-10-23, 21:10 IST

module bcd_adder_8bit (
    S0,S1,cout,A1,A0,B1,B0,cin,wire6
);
    input [3:0] A1, A0, B1, B0;
    input cin;
    output [3:0] S0, S1;
    output cout;

    wire[3:0] wire1, wire10;
    wire wire2, wire3,wire4,wire5;
    wire[3:0] wire20, wire30;
    wire wire7, wire8,wire9, wire12,wire11;

    output wire6;

    parallel_adder p1(wire1, wire2, A0,B0, cin);
    and a1(wire3, wire1[3], wire1[2]);
    and a2(wire4, wire1[3], wire1[1]);
    or o1(wire5, wire3, wire4);
    or o2(wire11, wire5, wire2);

    assign wire10[3]=1'b0;
    assign wire10[2]= wire11;
    assign wire10[1]= wire11;
    assign wire10[0]=1'b0;

    parallel_adder p2(S0, cout, wire1, wire10, cin);

    parallel_adder p3(wire20, wire7,A1,B1, wire11);
    and a4(wire8, wire20[3], wire20[2]);
    and a5(wire9, wire20[3], wire20[1]);
    or o3(wire12, wire8, wire9);
    or o4(wire6, wire12, wire7);

    assign wire30[3]=1'b0;
    assign wire30[2]= wire6;
    assign wire30[1]= wire6;
    assign wire30[0]=1'b0;

    parallel_adder p4(S1, cout, wire20, wire30, cin);
```

```
endmodule
module parallel_adder (
    sum,carry_out,in1,in2,carry_in
);
    input [3:0] in1, in2;
    output [3:0]sum;
    output carry_out;
    input carry_in;
    wire [2:0] wire1;

    full_adder_CA f1(sum[0],wire1[0],in1[0],in2[0],carry_in);

    full_adder_CA f2(sum[1],wire1[1],in1[1],in2[1],wire1[0]);

    full_adder_CA f3(sum[2],wire1[2],in1[2],in2[2],wire1[1]);

    full_adder_CA f4(sum[3],carry_out,in1[3],in2[3],wire1[2]);

endmodule

module full_adder_CA(
    sum,carry_out,in1,in2,carry_in
);
input in1,in2, carry_in;
output sum, carry_out;
wire wire1, wire2, wire3;
assign sum=(in1^in2)^carry_in;
assign carry_out=(in1 & in2)|(in2 & carry_in)|(carry_in & in1);
endmodule
```

Test-Bench:

```
// Developed by: Aashi Srivastava
// Title : 8-bit BCD adder test bench
// Date:16-10-23, 21:10 IST
module bcd_adder_8bit_tb (

);

    reg [3:0] A1, A0, B1, B0;
    reg cin;
    wire  [3:0] S0, S1;
    wire cout;
    wire wire6;


    bcd_adder_8bit p(
     S0,S1,cout,A1,A0,B1,B0,cin,wire6
);

initial begin
    $dumpfile("bcd_adder_8bit.vcd");
    $dumpvars(0,bcd_adder_8bit_tb);
    $monitor(" sum=%0d%0d%0d in1=%0d%0d in2=%0d%0d ",wire6,S1,S0,A1,A0,B1,B0);
        #40 $finish;
    end
initial begin
    A0=4'b1001;
    B0=4'b1001;
    A1=4'b1001;
    B1=4'b1001;
    cin=1'b0;
end
endmodule
```