NAME: Kundrapu N R Sai Akash

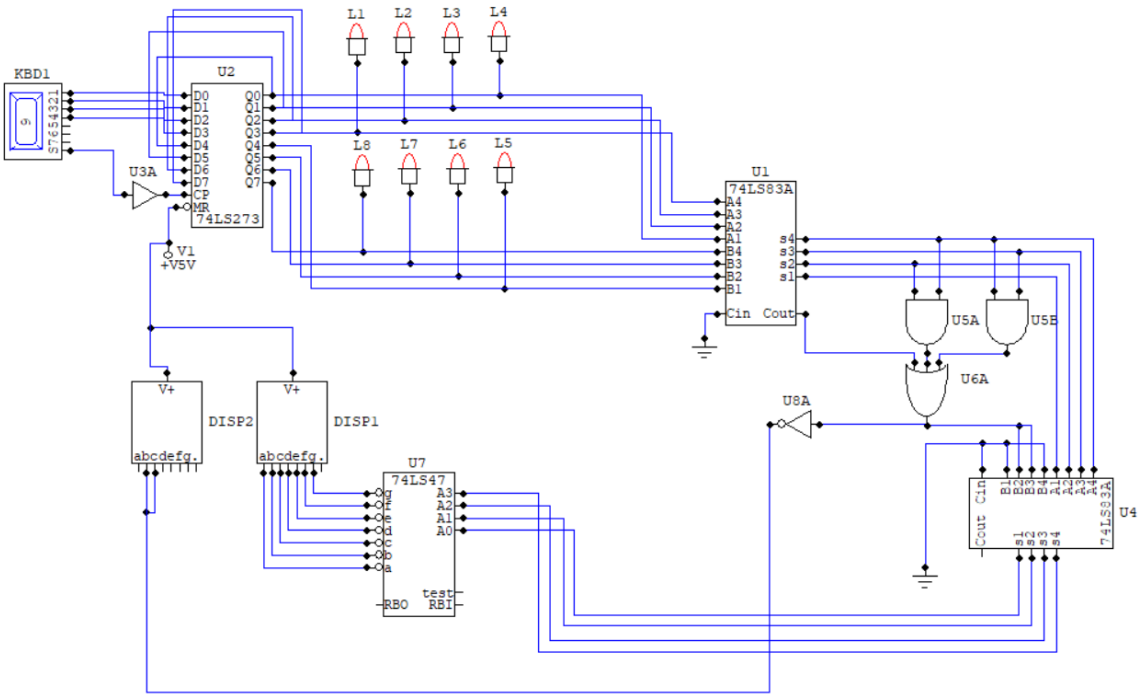
ROLL NO: 20EC10043

DIGITAL LABORATORY EXPERIMENT 5

AIM:

To add two single digit input from ASCII key and display the sum output on 7-segment displays.

CIRCUIT DIAGRAM



The devices required for this experimental setup are an ASCII keyboard, CD 5040(non-inverting buffer), Logic displays(x8), 74LS273(Octal positive edge-triggered D-type Flip Flop with reset), Power Supplies and IC 7483(x2)(4-bit Binary Full Adder), 7-segment displays(x2), IC 74LS47(4-bit decoder), and a few logic gates.

BREIF THEORY AND EXPLAINATION

We are creating a single digit adder in this experiment, so we need to take two inputs from ASCII keyboard and feed it to a 74LS273 just like the previous experiment, so the pins setup is same there, then we add those two numbers, after we do that, we get outputs from 5pins, "Cout & S1-4". This is a 5bit number, so a 7-segment display can't take this result all the time. So, we need have a logic circuit to tackle this problem.

We can achieve this by checking if the sum we got is among the BCD codes, 0-9, if yes we can display this in first display and keeping the second display blank, but when it is from 10-18, we need to be careful, as output will have flaws, to check it, we need to add 0110(6) to the sum, and display the added value in first display while the second display shows 1. So, we need a way to know when to add 6 and when not to add. The logic gate combination can be found out by using a K-map simplification with 4input pins S1-4. We don take Cout here because Cout is 1 only when the numbers are 16,17,18. So when Cout is 1, our logic output Cf must be high, that means we need to have a OR gate connection with inputs Cout and the result we get from K-map simplification.

We get, Cf = Cout + S4.S2 + S4.S3

We can realize this by having two AND gates(2In) and an OR gate(3In). With this we get the sum in correct format. Now we feed this binary value to 4-bit decoder which shows the display in ones place. To show display on the second screen, it is either 1 or blank, we can realize that by setting pins b,c to low when we need 1 and to high when we need a blank. That can be achieved by adding a inverted to Cf and feeding it to b,c pins of second display. In this way, we get the complete circuit for the experiment.

SNAPSHOTS OF THE SIMULATION:

To show the results of the simulations, I have taken 12 examples of binary additions. All the images are in the order given below,

Example01: 2+2=04

Example02: 4+5=09

Example03: 3+4=07

Example04: 1+7=08

Example05: 2+4=06

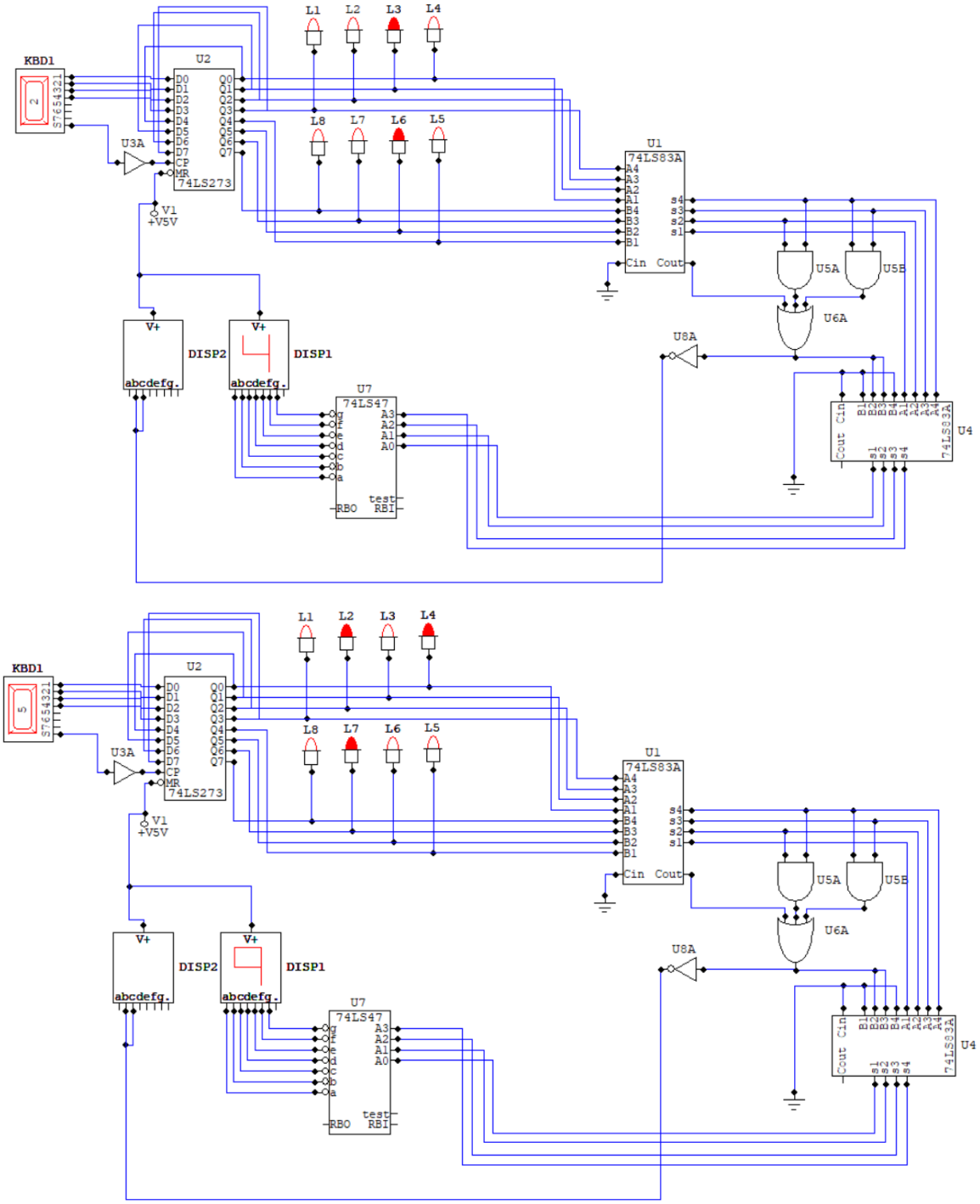
Example06: 5+6=11

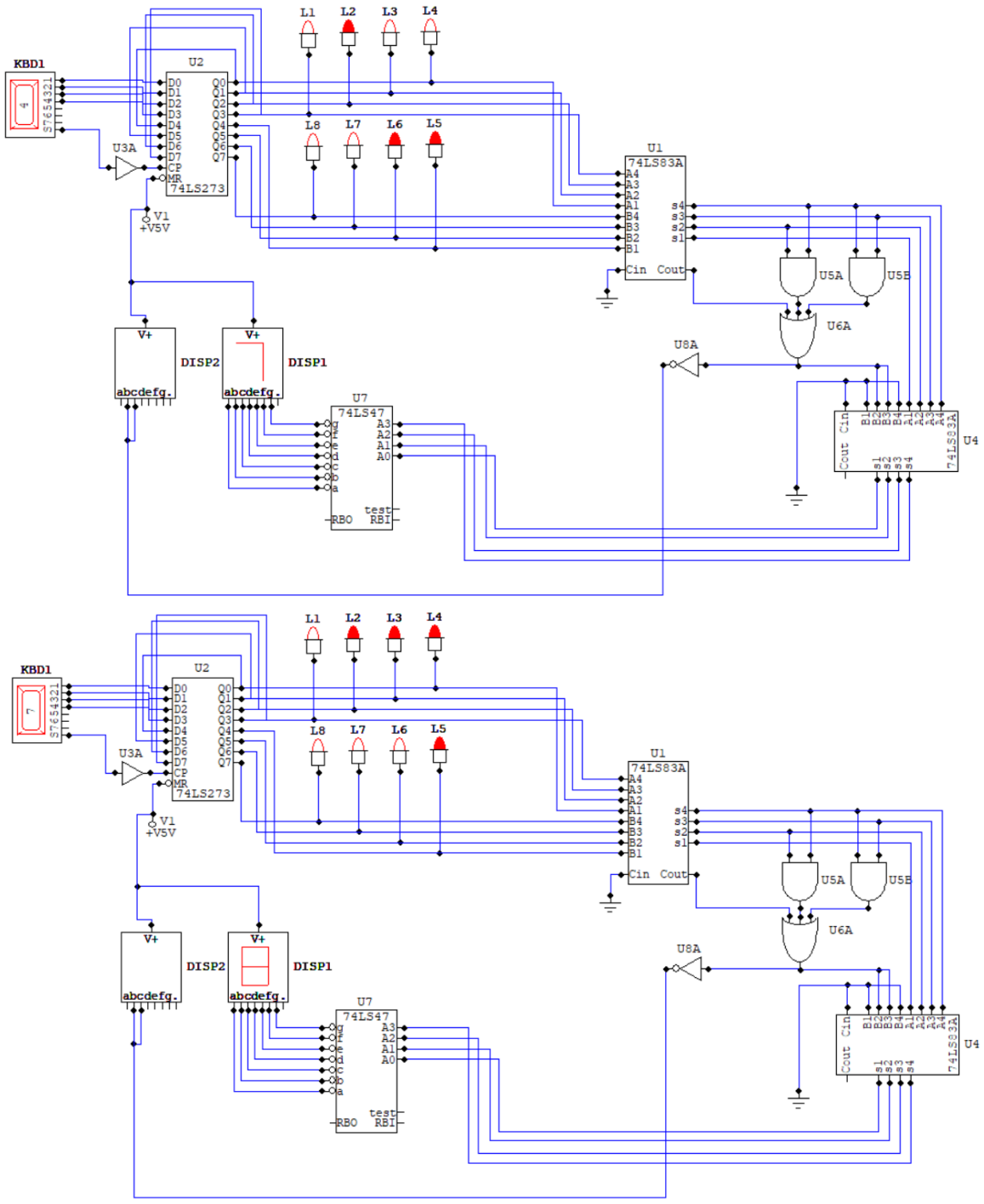
Example07: 4+7=13

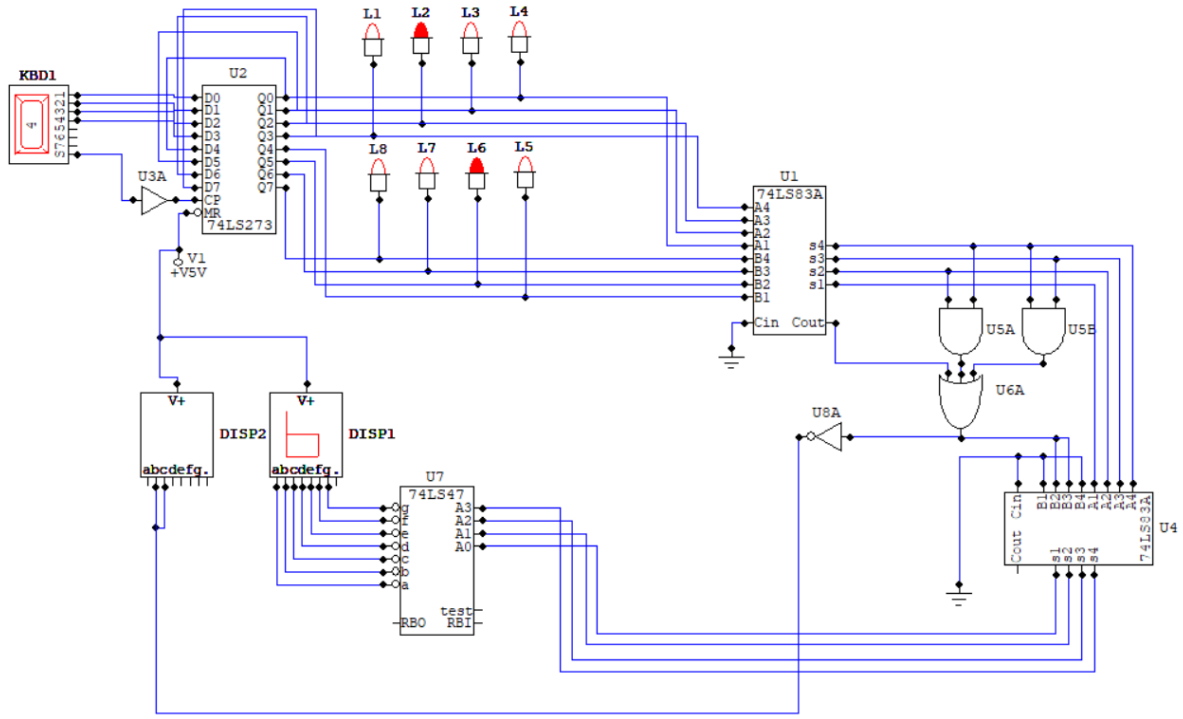
Example08: 6+8=14

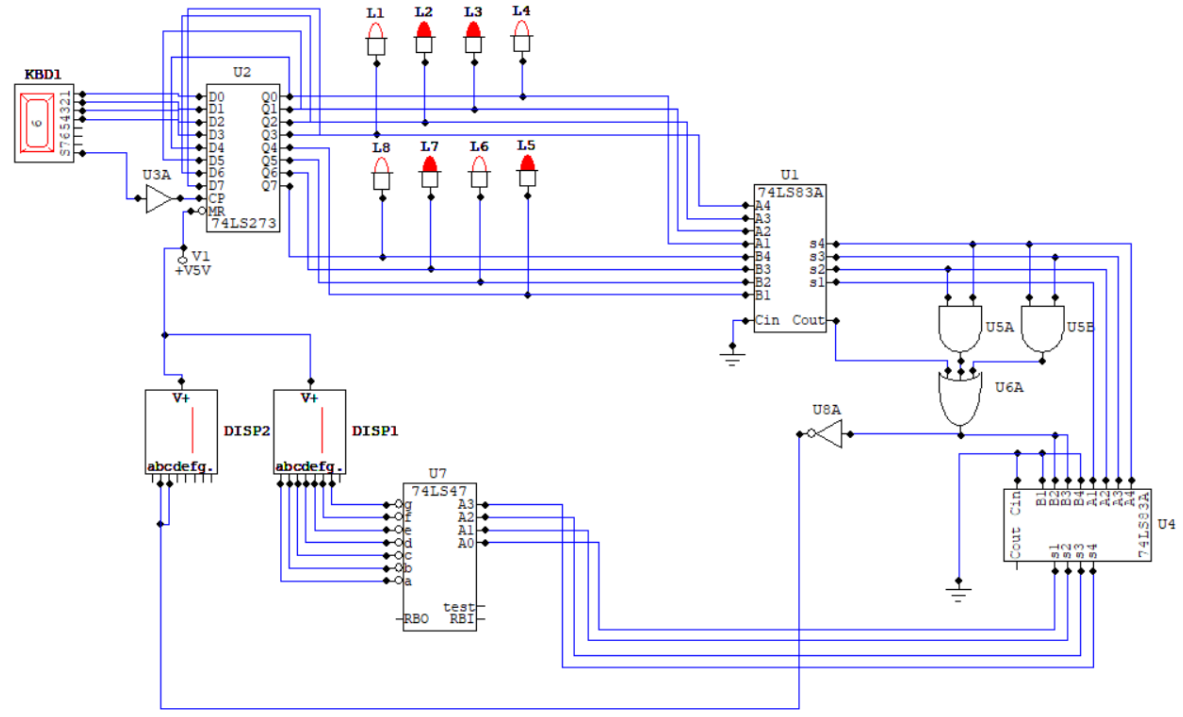
Example09: 9+9=18

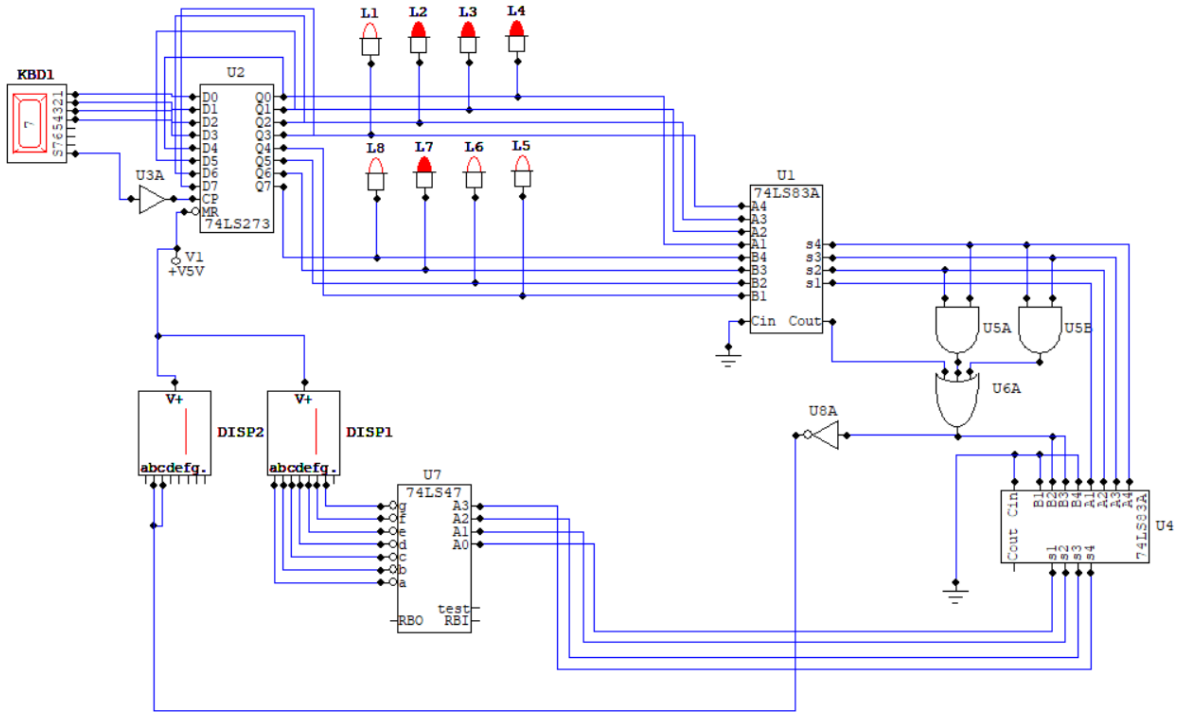
Example10: 9+4=13

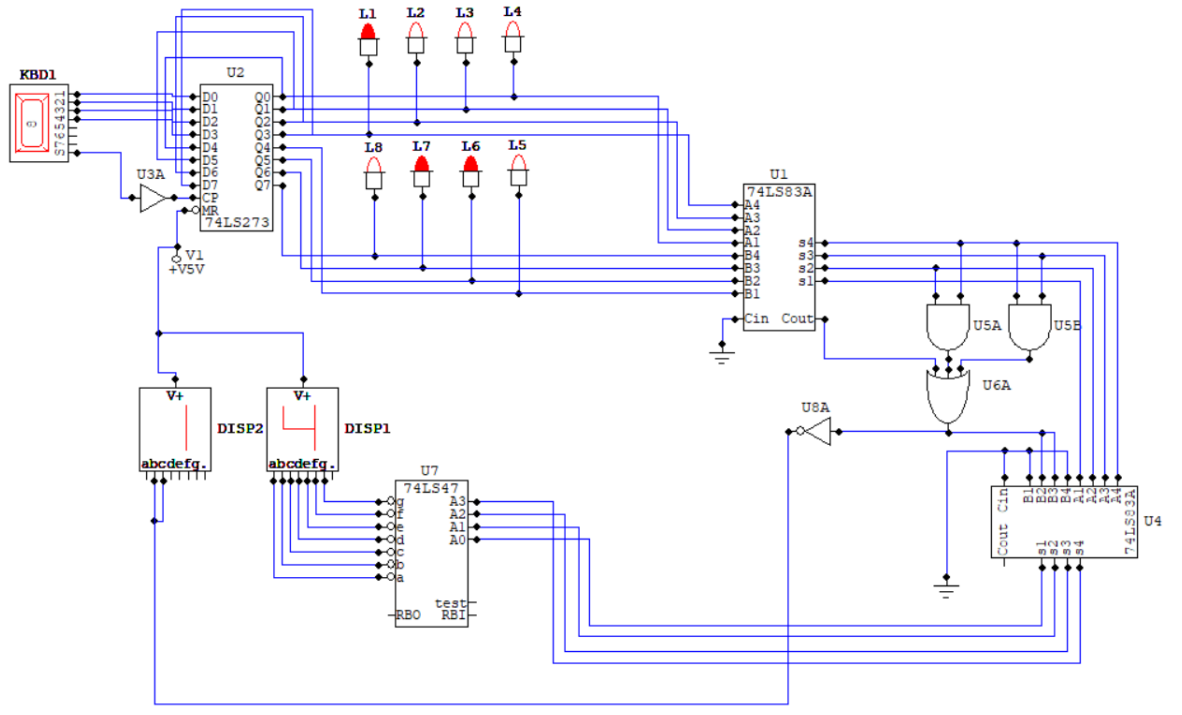

KBD1
U2
74LS273
U3A
V1
+V5V
L1
L2
L3
L4
L8
L7
L6
L5
U1
74LS83A
U5A
U5B
U6A
U8A
U4
V+
DISP2
DISP1
abcdefg.
U7
74LS47
test
RBO
RBI

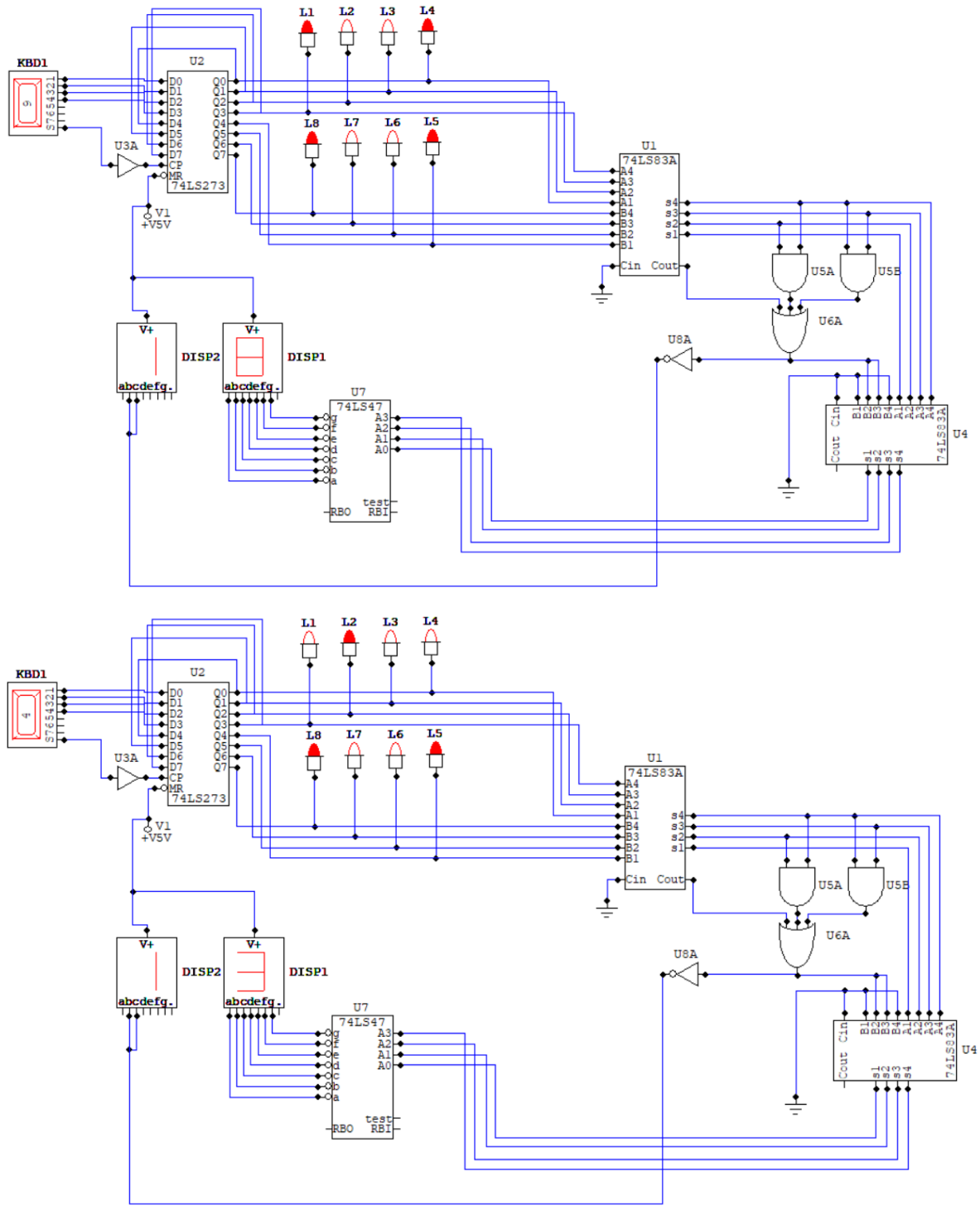

KBD1
U2
74LS273
U3A
V1
+V5V
L1
L2
L3
L4
L8
L7
L6
L5
U1
74LS83A
U5A
U5B
U6A
U8A
U4
74LS83A
DISP2
DISP1
abcdefg.
U7
74LS47
RBO
test
RBI


KBD1
U2
74LS273
U3A
V1
+V5V
L1
L2
L3
L4
L8
L7
L6
L5
U1
74LS83A
U5A
U5B
U6A
U8A
U4
74LS83A
DISP2
DISP1
V+
abcdefg.
U7
74LS47
test
RBO
RBI


KBD1
U2
74LS273
U3A
V1
+V5V
L1
L2
L3
L4
L8
L7
L6
L5
U1
74LS83A
U5A
U5B
U6A
U8A
U4
74LS83A
DISP2
DISP1
V+
abcdefg.
U7
74LS47
test
RBO
RBI


KBD1
U2
74LS273
U3A
V1
+V5V
L1
L2
L3
L4
L8
L7
L6
L5
U1
74LS83A
U5A
U5B
U6A
U8A
U4
74LS83A
DISP2
DISP1
V+
abcdefg.
U7
74LS47
RBO
RBI
test


KBD1
U2
74LS273
U3A
V1
+V5V
L1
L2
L3
L4
L8
L7
L6
L5
U1
74LS83A
U5A
U5B
U6A
U8A
U4
74LS83A
DISP2
DISP1
V+
abcdefg.
U7
74LS47
RBO
RBI
test

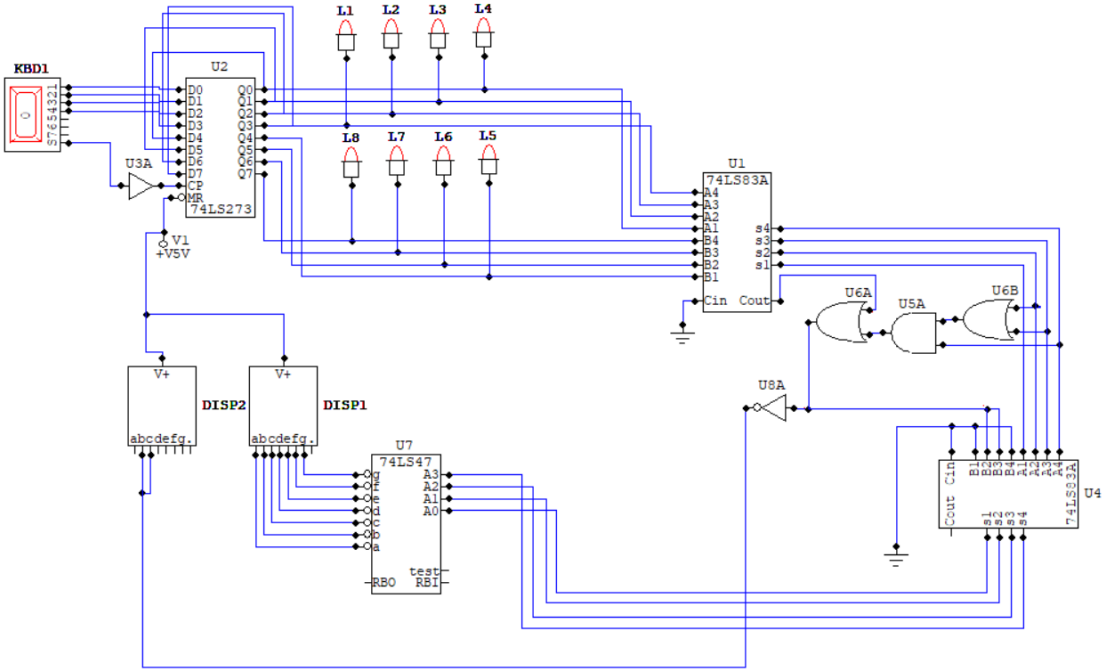

KBD1
U2
74LS273
U3A
V1
+V5V
L1
L2
L3
L4
L8
L7
L6
L5
U1
74LS83A
U5A
U5B
U6A
U8A
U4
DISP2
DISP1
U7
74LS47
test
RBO
RBI

OBSERVATION AND CONCLUSION

From the observations, it is clearly working. We can modify the logic gates parts as we like, but it is preferred to have a circuit which costs less along with being hazard free. Our experiment doesn't contain any hazards, so we can go ahead with the minimized SOP expression.

We got the Boolean expression as Cf = Cout +S4.S2 + S4.S3

We implemented the 2-level logic system, where the number of gates are 3, and the number of gate inputs are 7. So the total cost is 10. When we use the same equation like Cf = Cout + S4.(S2+S3). We need two OR gates(2In) and 1 AND gate(2In). This is 3-level implementation where there are 3 gates and 6 gate inputs, which results in the total cost being 9. So this is more economical that the previous setup, for this the circuit will be like this,



I have attached the ckt file for this logic setup with less cost for checking. This brings to the end of this experiment.