NAME: Kundrapu N R Sai Akash

ROLL NO: 20EC10043

DIGITAL LABORATORY EXPERIMENT 8

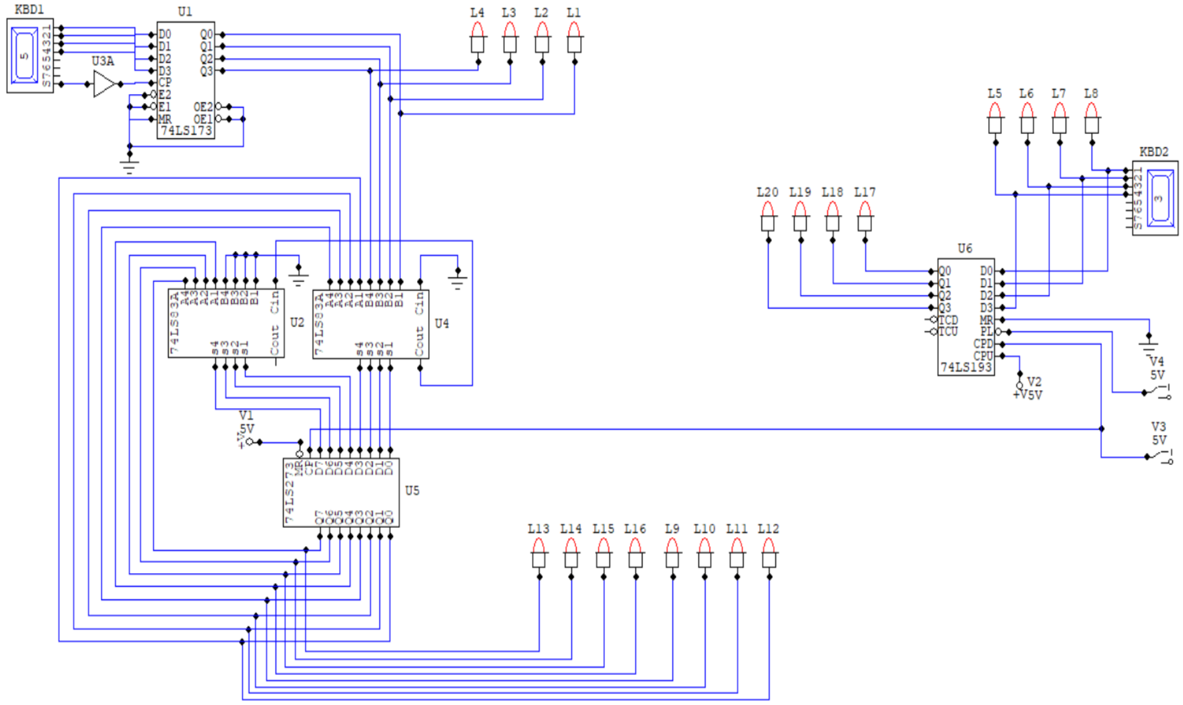
AIM:

Realizing a 4x4-bit multiplier using a down counter and product-accumulator register.

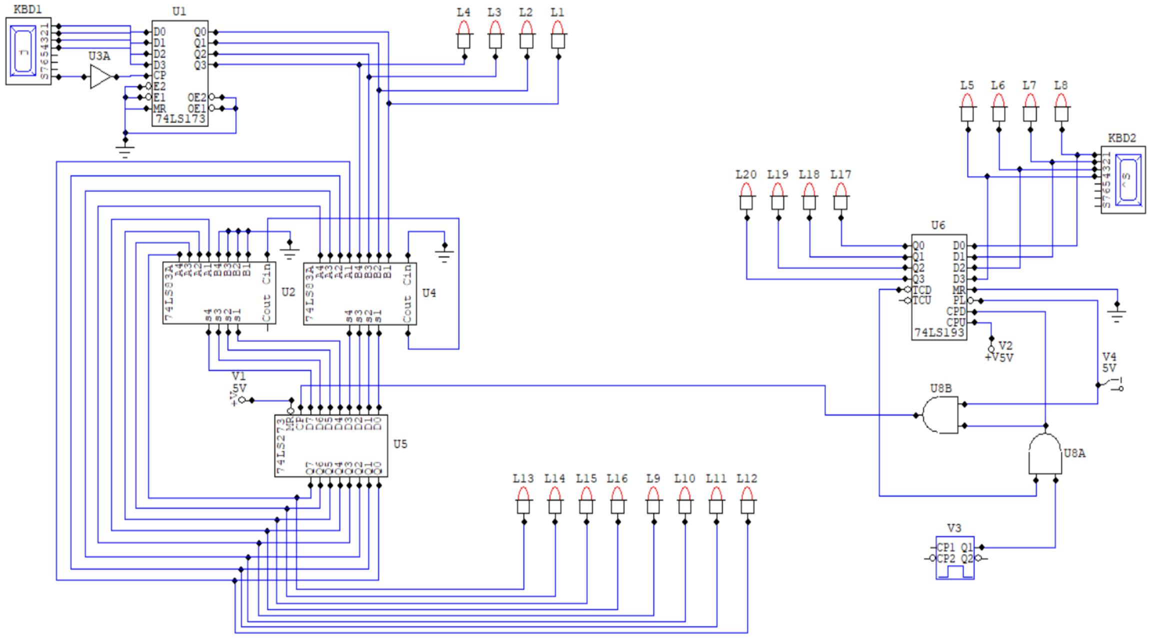
CIRCUIT DIAGRAM

For this experiment, we need 2ascii keyboards, 4-bit register, a down counter, a product-accumulator register(8-bit) and an 8-bit adder which can be realized with the help of two 4-bit full adders and a bunch of a logic displays.

For the manual part of the operation, we use few logic switches to give signals manually.



For the part, where we automate this adder, we change a few logic switches to logic gates.



BREIF THEORY AND EXPLAINATION

In this experiment, we want to realize a 4x4 multiplier, which can be done in a simple iterative method, where we do multiplication by adding. So, we need a way to countdown the multiplier and add as many times the multiplicand with itself, so as to get the desired result.

For the process of taking the numbers as input, we use 2 ascii keyboards to take the inputs of multiplier and multiplicand, once this is done, for multiplicand side, we give the output from ascii board to a 4-bit register, with a delay using a buffer, like previous experiments. From the 4-bit register, we give the connections to 8-bit adder, where the second 8-bit number is the output, to control the addition and not continuously keep on adding, we use a 8-bit register, to add only when a signal is received by it.

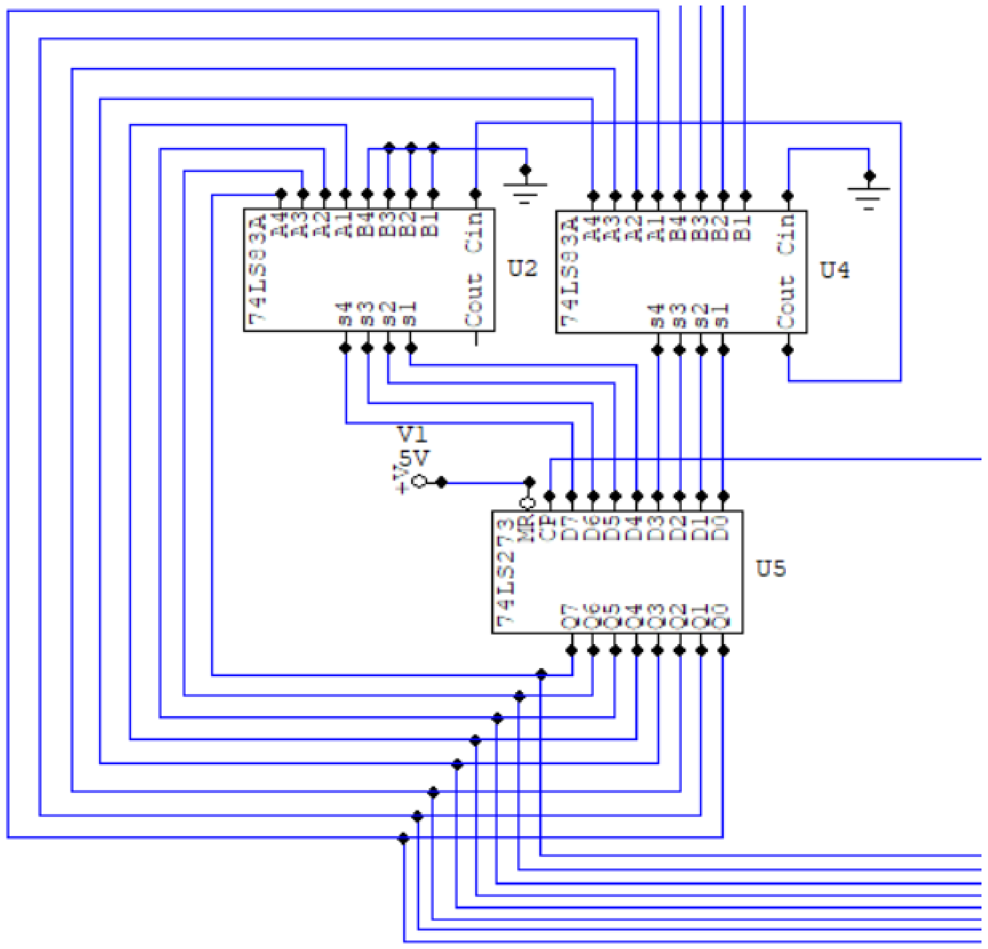
For the 8-bit adder, we achieve that by using 2 4-bit adders, where we connect them in parallel with Cout of 1st adder goes to Cin of 2nd adder. As for *the inputs of 4bit adders, we need to careful, by giving the first 8-bit number as the direct input from 4-bit register, which is our multiplicand. So, we use the lower significant nibble as the input pins for multiplicand and the higher nibble is all zeros 0000, as we are using a 4-bit number in place of 8-bit, the other number input for 8-bit adder is the output of the adder which we get after each addition, so the nibbles are in the same order of significance. Cin for 1st adder is grounded as we have no use for it. For the second one, Cout is not necessary, so we will not use it in any circuits.*

Now, for the part of adding the number iteratively every time, we make use of IC 74LS193 which is both a up and down counter. For our experiment, we use only down counter property of the IC, so we give a high signal to CPU, and a normal manual signal to CPD. We ground the Master reset as we don't need it. We have another pin in this device which is called PL (parallel load enable), which we use to start the count down in the device by sending a step signal. Then, the number we gave as input starts going down by 1 every time when it receives a signal at CPD. For the manual part, we just connect a bunch of logic switches at appropriate pins to have the intended results.

For the automated part of the experiment, we need to stop giving signals manually, so we need a pulser to do the same stuff, but for PL, we still use a switch so as to start the count down and each iterative adding at the same time. As, we do this, we will face a problem where the adding never stops, which is caused due to the countdown going from say(x), x-1, ..., 1, 0 and will still go to -1, -2 and so on. This is something which we don't want to happen in a multiplication operation. So, we need a way to stop this, which we need to do exactly when number goes to 0 as we countdown. So, at that moment, TCD goes to 0. So, we give an AND gate connection to CDU with the normal pulser clock input and TCD output, so the input to CDU goes to zero when we reach 0, and thus stopping the counter to go further low. We give the same connection to the clock input of the 8-bit register, as we need to add exactly as many times. I have added another AND gate to this output signal and a logic switch so as the start the addition operation and countdown operation at the exact same time. By doing so, we can achieve the desired output.

SNAPSHOTS FROM THE EXPERIMENT

In this experiment, one main component is the 8-bit adder, which is important for the correct working of the circuit. The circuit for the 8-bit adder is given below,



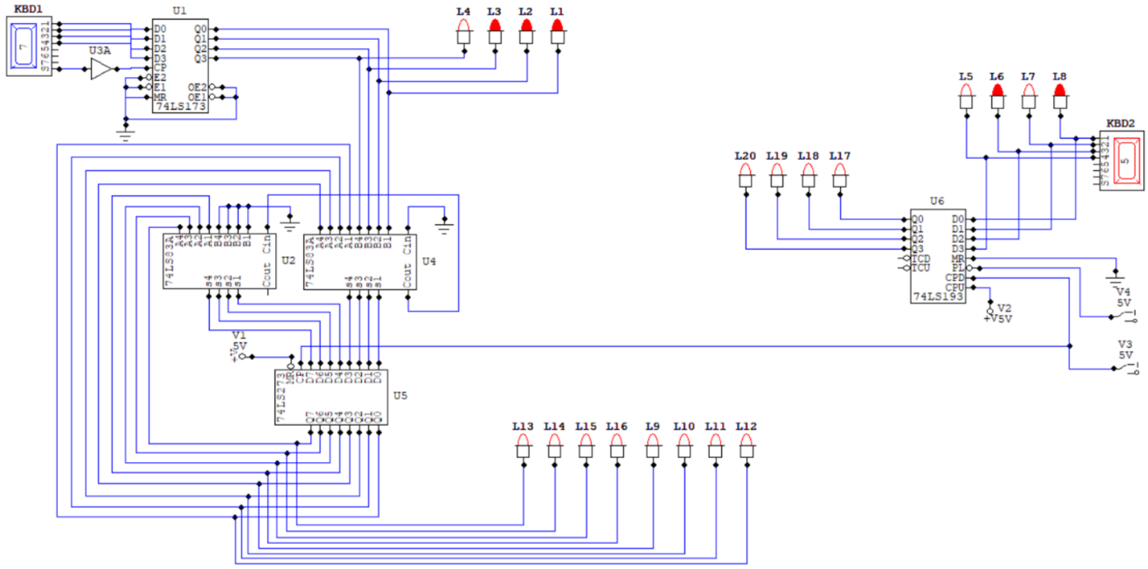
Here, in U4, B4-1 are the multiplicand input.

In U2, B4-1 are grounded, both together makes the 8-bit number in the form 0000ABCD, which is the the 4-bit multiplicand.

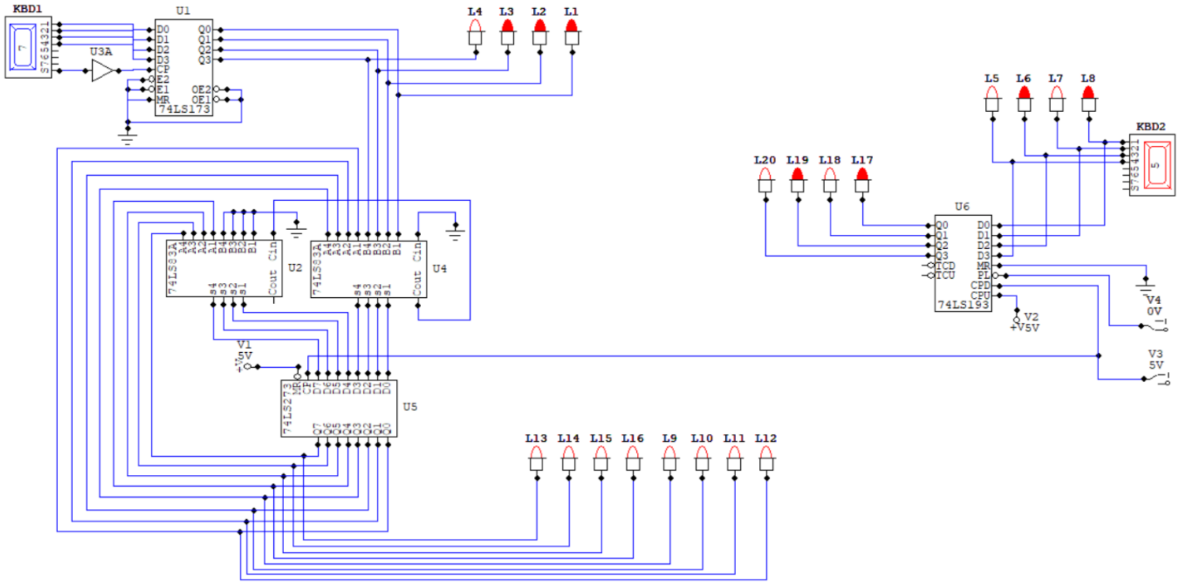
The A4-1 nibble of U4 unit is connected to Q0-3 of register, which is the lower significant nibble, where are the upper nibble is connected to A1-4 of U2.

Cout of U4 is connected to Cin of U2 which provides to carry between 4th and 5th bit of the number.

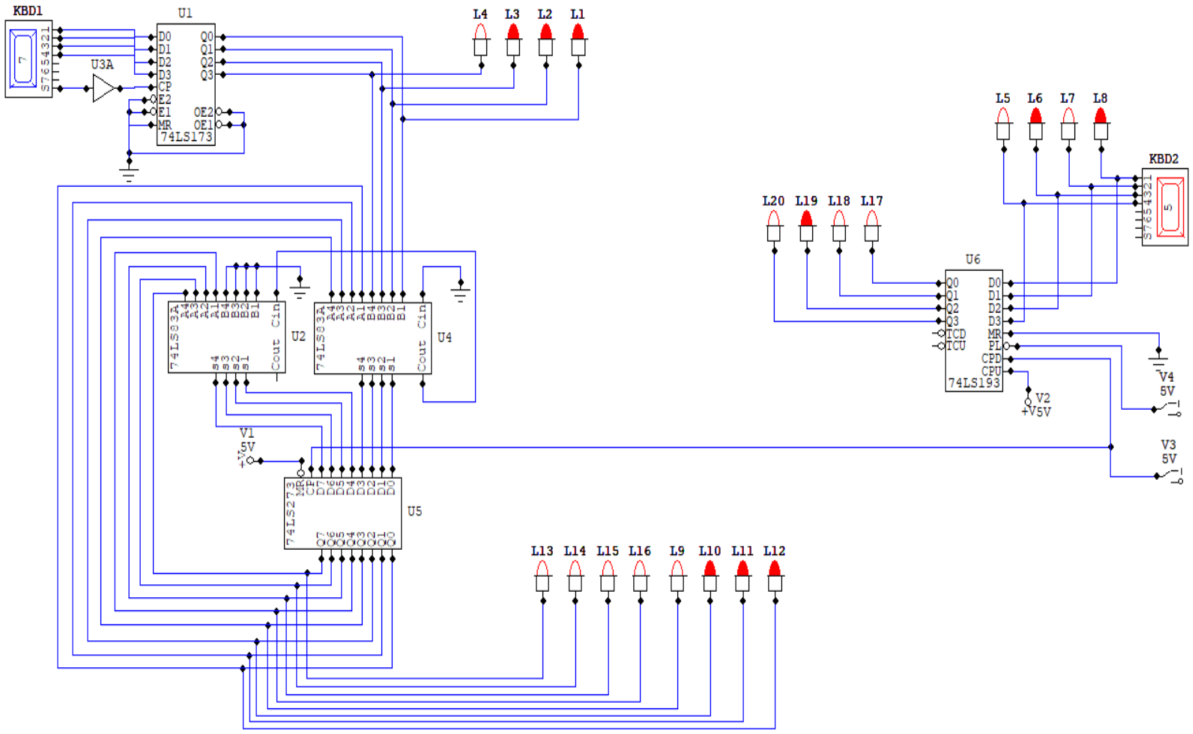
For the purpose of demonstrating the working of manual multiplication, I have taken multiplicand as 7 and multiplier as 5(a small number so as to make it simple). Snapshots of this experiment are given below,

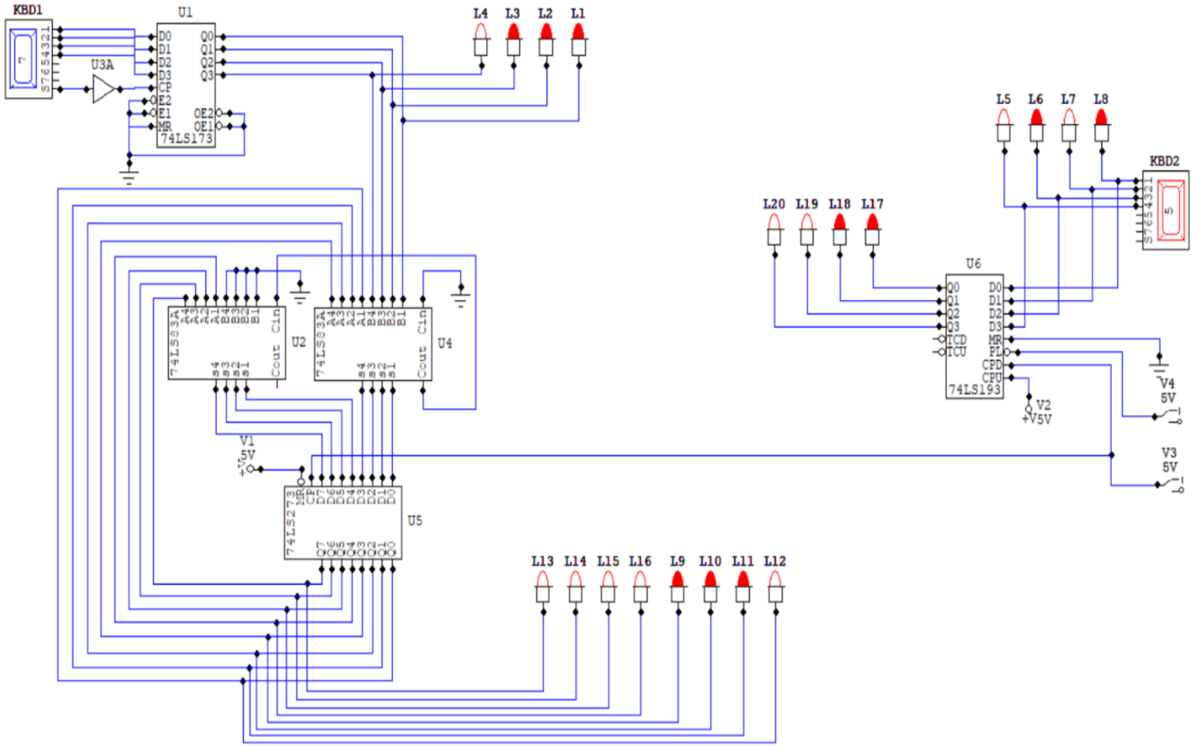


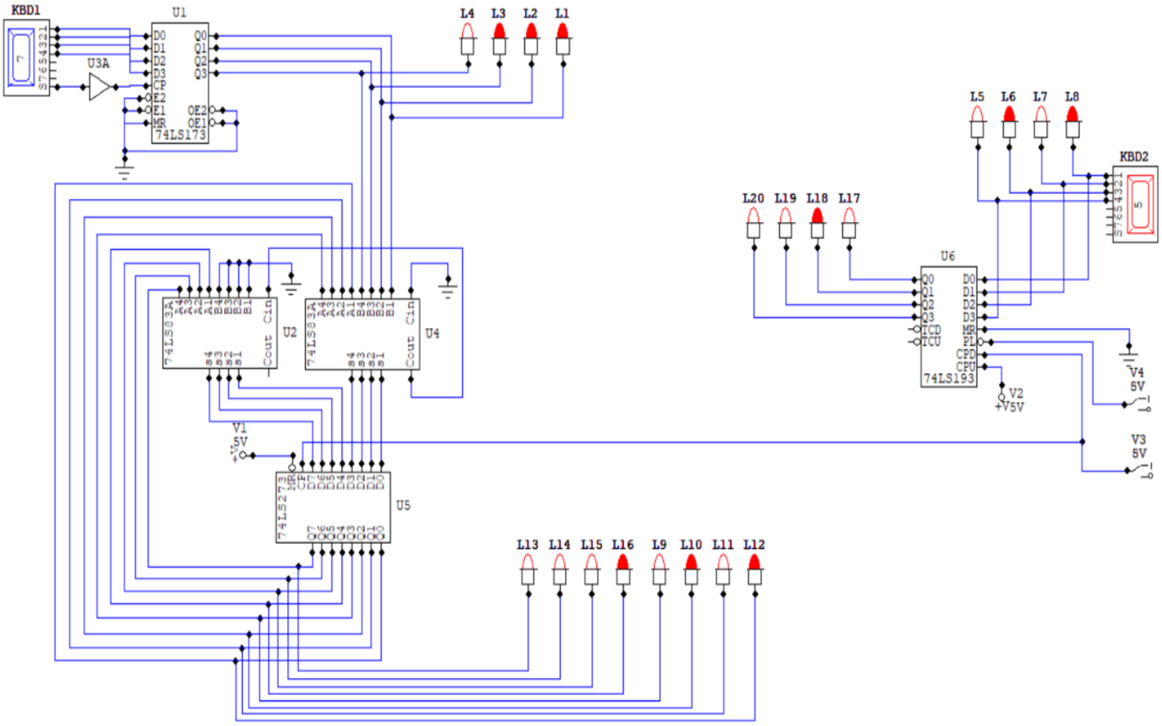
Given the inputs at their specific keyboards, then,

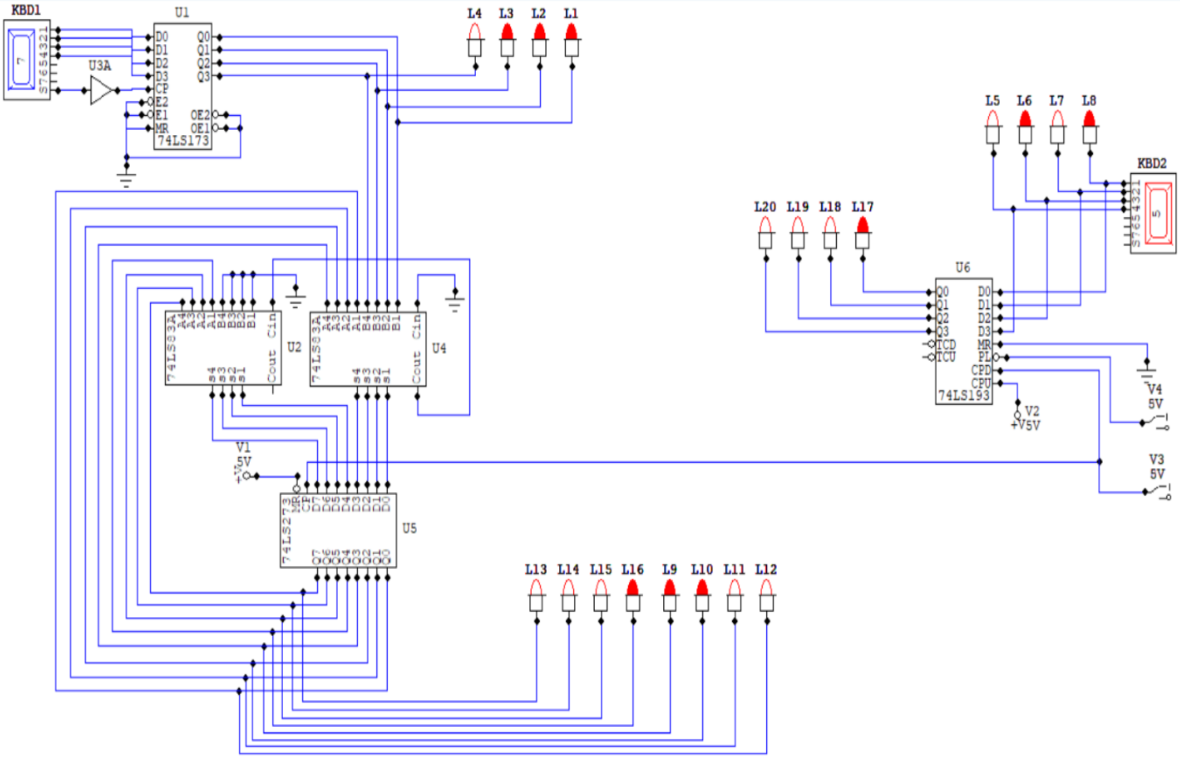


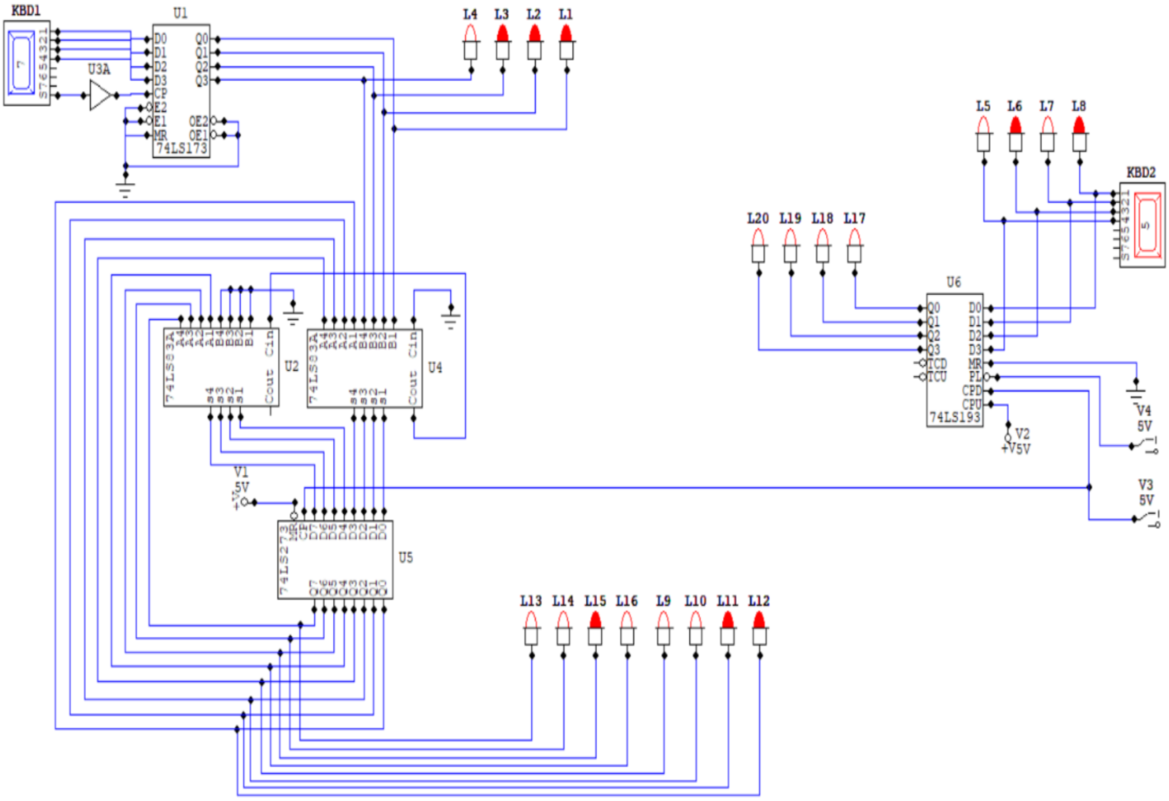
Taking PL to 0, so as to initiate the counter, as we see the number 5 on the output side as well right now. Now, we will take PL back to HIGH and switch on and off the second logic switch(V3) which connects CP of register and CPD of counter.


KBD1
U1
74LS173
U3A
L4 L3 L2 L1
L5 L6 L7 L8
KBD2
L20 L19 L18 L17
U6
74LS193
74LS83A
U2
U4
V1 5V
V2 +V5V
V3 5V
V4 5V
74LS273
U5
L13 L14 L15 L16 L9 L10 L11 L12


KBD1
U1
74LS173
U3A
L4 L3 L2 L1
L5 L6 L7 L8
KBD2
L20 L19 L18 L17
U6
74LS193
74LS83A
U2
U4
V1 5V
V2 +V5V
V3 5V
V4 5V
74LS273
U5
L13 L14 L15 L16 L9 L10 L11 L12

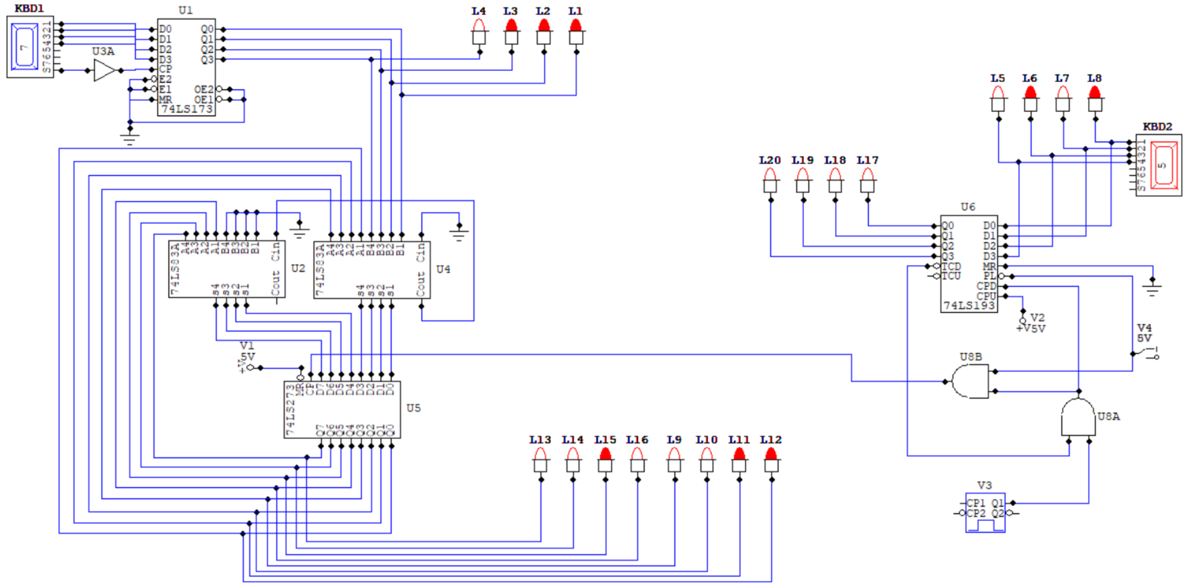

KBD1
U1
U3A
74LS173
D0
D1
D2
D3
CP
E2
E1
MR
Q0
Q1
Q2
Q3
OE2
OE1
L4
L3
L2
L1
L5
L6
L7
L8
KBD2
L20
L19
L18
L17
U6
74LS193
TCD
TCU
MR
PL
CPD
CPU
V2
+V5V
V4
5V
V3
5V
74LS83A
U2
U4
A4
A3
A2
A1
B4
B3
B2
B1
Cin
Cout
S4
S3
S2
S1
V1
5V
U5
74LS273
L13
L14
L15
L16
L9
L10
L11
L12


KBD1
U1
U3A
74LS173
L4
L3
L2
L1
L5
L6
L7
L8
KBD2
L20
L19
L18
L17
U6
74LS193
V2
+V5V
V4
5V
V3
5V
74LS83A
U2
U4
V1
5V
U5
74LS273
L13
L14
L15
L16
L9
L10
L11
L12

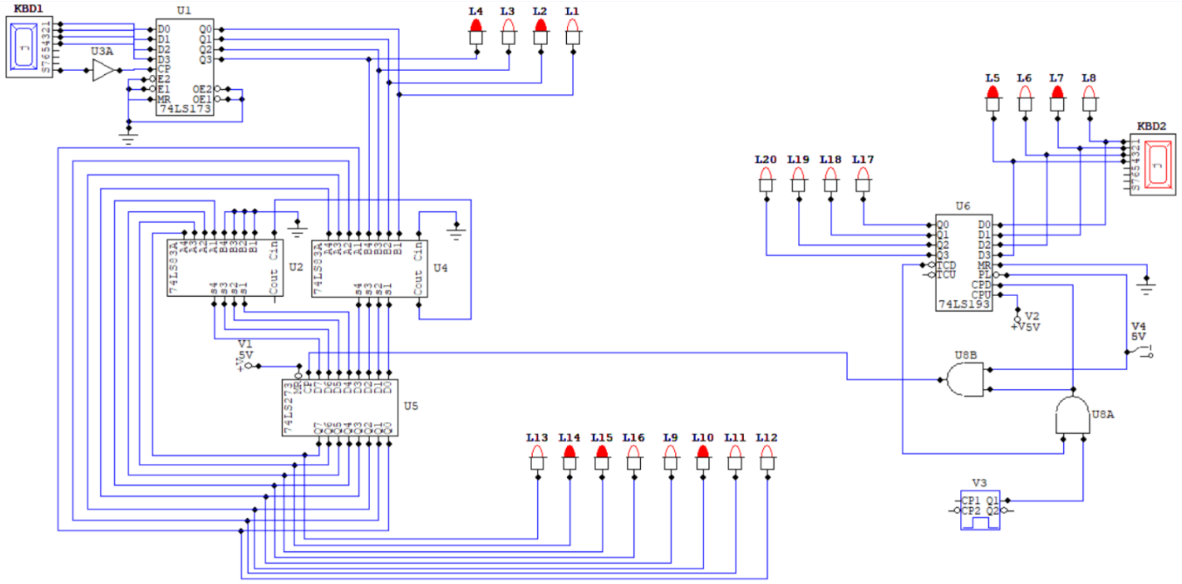


*As we expect to see, after the counter goes to 0 as we go from 5-4-3-2-1-0, we see that the sum is going from 0-7-14-21-28-35, which shows the process of recursive adding for each single down count. So, we get the final result as 35 which is indeed 5*7.*

*For the automated part of the experiment, we cannot show the results in intermediate stages in form of snapshots. So, I have attached the final result of 2 examples, the same 7*5 and bigger multiplication with 10*10.*



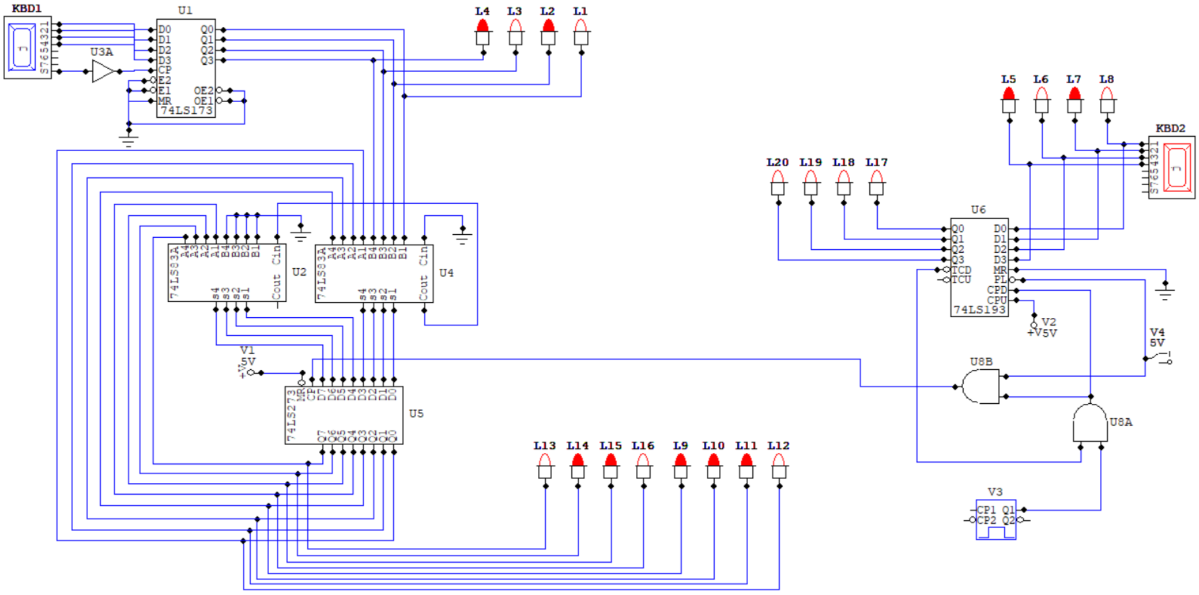
*5*7 = 35*



*10*10 = 100*

OBSERVATION AND CONCLUSION

*Regarding the observation part, the manual design works absolutely fine every time, whereas the automated part sometimes shows skewed results. This occurs due to the time delay added by the AND gates used in the logic circuit. When this happens the addition starts earlier that the countdown which results in form of answer being wrong and have another multiplicand added to it. To make it accurate we need to check if the first countdown and first addition takes place at the same time or not, for this, we can reduce the simulation speed to clearly check it. If the both change at same time, answer will be correct, else an additional iteration takes place which results in wrong answer. The snapshot given below shows the skewed result for the same example taken above which is 10*10, but we get 110 as an additional iteration has taken place due to time delay.*



For the purpose of checking, I will attach both the .ckt files along with this experiment. This takes us to the end of this experiment.