Digital Electronics Circuits lab
Lab Experiment:9

Name:Harshavardhan Alimi

Roll No:18EC10021

AIM:To design a circuit to implement a 4x4 bit multiplier using registers and down counter.

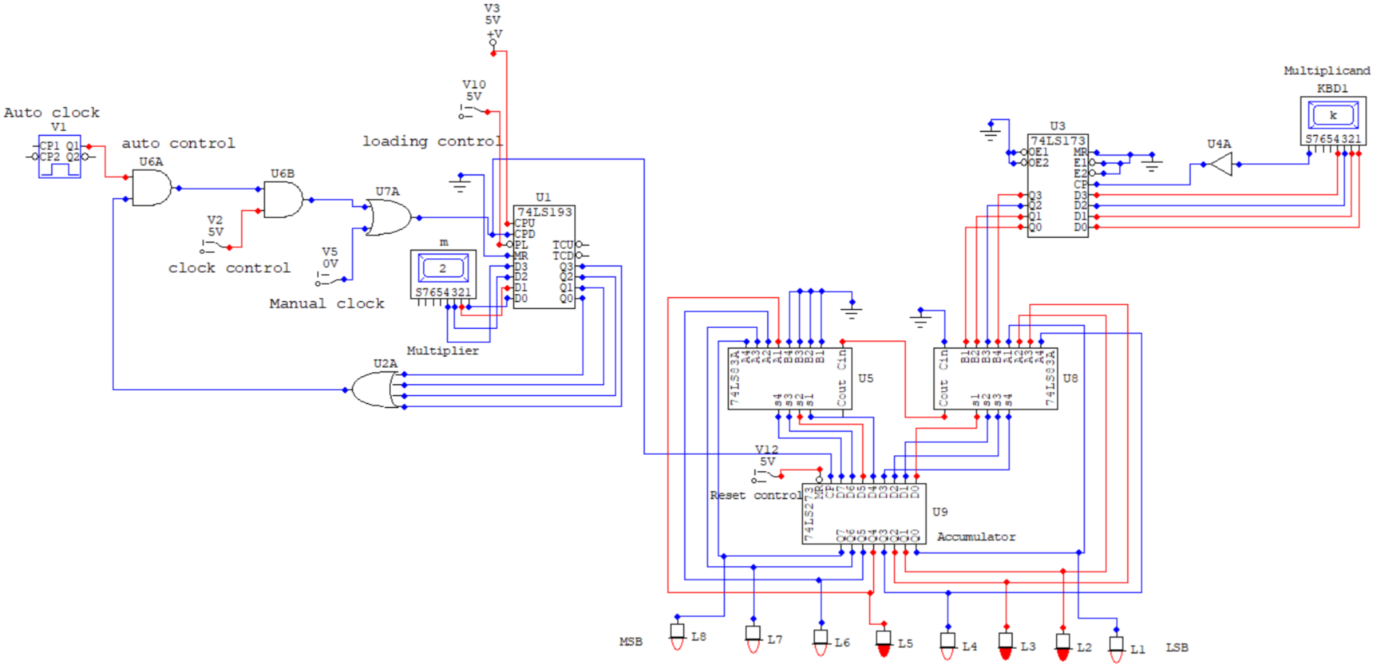
ICs used:-

- IC74LS193:-1 quantity
- IC 74LS173:-1 quantity
- IC 74LS273:-1 quantity
- IC 74LS83:-2 quantity

Buffer/Driver:-CD 4050:-1quantity

Components used:- Logic Gates(and gate;or gate),ASCII Keypad(2),logic switches(4),logic display(8).

Circuit:Digital circuit to display the product of the multiplier and multiplicand at the 8 logic displays.



product=multiplicand **x** multiplier

Working and Discussion of circuit:

- We have seen the working of ASCII Keypad in the previous experiment.
  - We will give the values for multiplier and multiplicand from the ASCII keypad.
- IC 74LS173:Quad D-type flip-flops with positive-edge trigger and 3-state outputs.
  - 16-pin DIP: 4D's,4Q's,OE_1',OE_2',IE_1',IE_2',CP,MR,V_{cc},GND.
  - When IE_1',IE_2'=0,data on D inputs loaded into the register with (0->1 edge) transition on CP input.
  - MR:asynchronous active high clear input.
  - IE_1',IE_2':always enabled('0').
  - OE_1',OE_2':active low output enable.if either one of them(or both) is high ,data at Q outputs is tristated.only if both of them are low,register data is available at Q outputs.
  - CP:it is connected to strobe output of ASCII keypad through non-inverting buffer CD4050,to provide necessary delay of clock pulse after data are presented at Dinputs.
- IC 74LS273:octal D-type flip-flops with positive edge trigger.
  - Inputs and outputs of 8 flip-flops are 8D's and 8Q's respectively.
  - MR:when it is low then the data at the all 8 outputs are erased,so to control this we made a reset control.
  - CP:for each positive edge trigger at the CP,it will store the data present at the inputs(8D's) and set the output values(8Q's).
  - We use logic displays at the 8 outputs of this IC to sense the values of IC74273.
- IC74LS83: 4-bit input adder.
  - We require 2 4-bit adders as our output value will be a 8-bit output.
  - In 1 adder we connect the 4-bit multiplicand and the lower nibble from the IC74273 as inputs and we will be getting a lower nibble of the output product.
  - In 2nd adder we connect the upper nibble from the IC74273 as 1 input and other 4inputs are made '0' and we will be getting the higher nibble of the output product.

- IC74LS193: synchronous 4-bit counter with dual clocks.
  - We use down counting in our experiment,so we keep CPU (clk for up counting) inactive i.e.,low and feed clock pulses in CPD(clk for down counting).
  - While feeding multiplier value (from the ascii keypad) into the down counter,place the value in parallel inputs(D_3….D_0) and make PL low(0) for some time by a logic switch then make PL high(1) to enable progressive counting in either direction, for this we made a loading control.
  - We keep MR inactive(low) for all times.
  - Based on the value of the multiplier, multiplicand gets repeatedly added to the accumulator(register IC74273). At the beginning of the experiment we make MR to be low and again make MR high by using reset control.
  - During down counting ,the TCL output which becomes low for a small amount of time may be used to inhibit(disable) the clk to 74193(as well as 74273) to freeze the contents of the product register. An alternative way to generate clock enable(auto control) is Q_3+Q_2+Q_1+Q_0 this will become 0 when all Q's are '0' i.e., 0000.
  - This will make the adder add the **multiplicand multiplier** number of times.
- We use manual clk for smaller values of multipliers and we use pulser for higher values of multiplier to control this we use clock control.

Result:

- We gave multiplicand as k(11) from ascii then it get stored at the outputs of 4Q's of(74173).
- We gave multiplier as 2 from ascii then we made PL from low to high so as to get values stored at the 4Q's of 74193.
- Accumulator values:(for each positive trigger from the clk)
  - 00001011 :-11
  - 00010111 :-22 (11+11)
  - And it freezes here as the clk is inhibited going further,this gives the output product as 22 in binary format.