Objective:

The objective of this lab was to build the circuit that multiples two 4-bit numbers together and displays the 8-bit result. The display showed the results with 4 least significant bits first and then the 4 most significant bits. The two inputs were positive hexadecimal numbers and two DIP switches were used to enter the two input numbers. The computation was done in less than ten clock pulses and it does not include the load of one of the numbers.

Components:

-One 74LS157 Quad 2/1 Data Selector (MUX)

-One 74LS04 Hex Inverter

-One 74LS164 8-bit Serial Shift Register

-One 74LS174 Hex D-Type Flip-Flop with Clear

-One 74LS181 Arithmetic Logic Unit/Function Generator

-One 74LS195 4-bit Parallel-Access Shift Register

-Two 74LS247 BCD to 7-Segment Decoder/Driver

-One 17TOGSD-M SPDT (On)-(On) Toggle Switch

-Two 17DIP8SS 8-Switch DIP Switch Sets

-5V Power Supply

-Four 1000 Ohm Resistors

Experimental Approach:

The main idea for this lab is to use the similar normally multiplies way in binary.

Consider how we normally multiply numbers:

```
      123
x     264
-------
      492
     7380
    24600
-------
    32472
```

If we use the same way for binary number:

```
Multiplicand      1011
Multiplier    x   0110
-------------------
                  0000
                 10110
                101100
               0000000
-------------------
Product        1000010
```

The first step of the multiply is store “0000” in a register. The second step is “1011”* “1” then shifts right and adds up two number (0000+10110). The rest of the steps are just repeating the second step till finish. The first clock cycle is to load the 8 bits input number and then use three clock cycles to repeat step 2. The main idea is “load-add and shift”.

Load:

To load the two 4 bits input numbers, we choose to use a 74LS157 Quad 2/1 Data Selector(MUX) for the multiplicand A(A0A1A2A3) input and 74LS195 4-bit Parallel-Access Shift Register for multiplier B(B0B1B2B3) input. MUX is use to choose either output the A (A0A1A2A3) or 0 and it is depends on the multiplier. For the multiplier, we use it to determine whether we add zero or a version of the multiplicand into the product. In each clock cycle of the multiplication, we check one bit (LSB), and use that to determine whether the output of the multiplicand shift register is passed into the adder, or zeroed. The shift register will shift right so we can check LSB number in every clock cycle.

Add and Shift:

After load the input in the register, the first cycle has complete and first product is A (when the LSB of multiplier is 1) or 0(when the LSB of multiplier is 0). The next step needs to store the first product in the 74LS174 Hex D-Type Flip-Flop and determine the second product. In the example “1011*0110”, the first product is 0000 and the second product is (0000+10110), so 1011 will shift to left via 74LS164 8-bit Serial Shift Register to become 10110 then add 0000 via 74LS181 Arithmetic Logic Unit/Function Generator . After we get the second product, store again in the D-Flip-Flop and repeat the load-add and shift step again.

Total Steps:

1. Clear all numbers and set Clear to 1
2. Switch the mode to load
3. Use the clock first time
4. Switch back to shift mode
5. Use the clock three times and get the final answers

Results:

The simulation was implemented successfully and the circuit was built properly without any errors thus it displayed the desired results correctly. The circuit multiplied the two 4-bit numbers together and displayed the 8-bit result. The display showed the results with 4 least significant bits first and then the 4 most significant bits. The computation was done in 4 clock pulses.

Conclusion:

To conclude, we completed the lab successfully and correctly. The circuit multiplied the two 4-bit numbers together and displayed the 8-bit result. The display showed the results with 4 least significant bits first and then the 4 most significant bits. The two inputs were positive hexadecimal numbers and two DIP switches were used to enter the two input numbers. Overall, our lab met all the requirements listed in the assignment.

Circuit Schematic: