Embedded System Lab

(ELC3930)

Experiment No.: 02

Object:

Write a program using 8085 simulator for Multiplication of Two 8-bit numbers (using Shift/Add method)

G. No: GL3136

S. No: A3EL-02

F. No: 19ELB056

Name: Maha Zakir Khan

Date of performing experiment: 17|01|2022

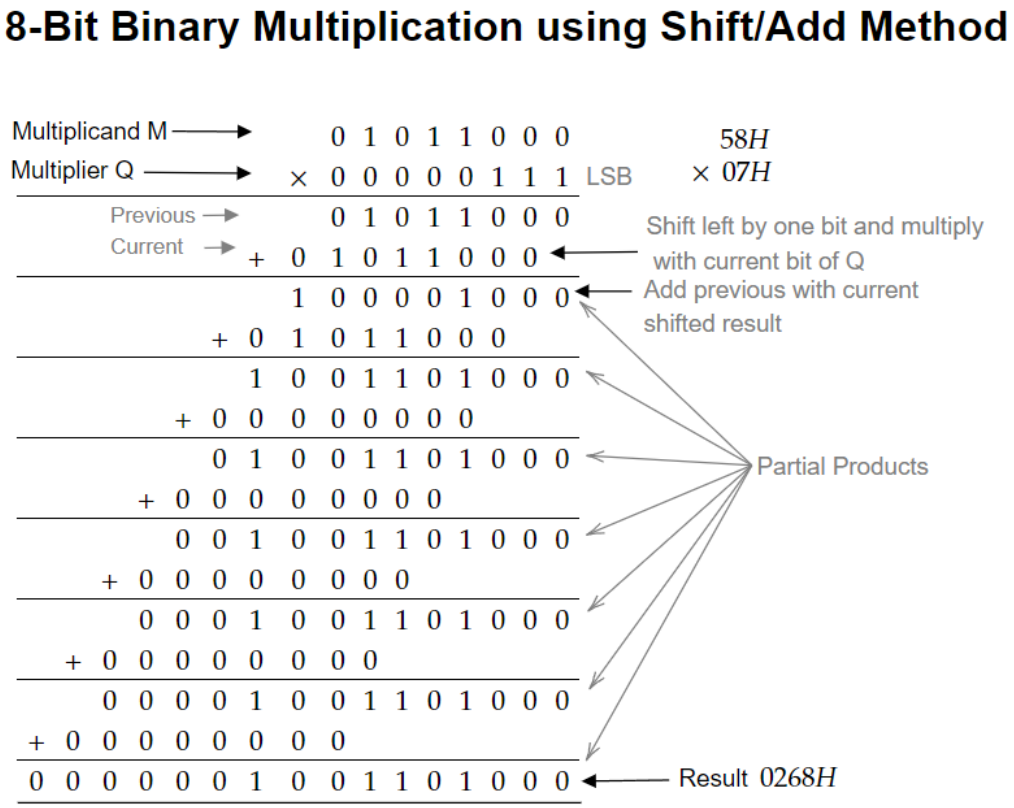
Date of report submission: 23|01|2022

Simulator Used:

8085 Simulator by Jubin Mitra. It helps in get started easily with example codes, and to learn the architecture playfully. This tool is an integrated software environment for teaching microprocessor concepts. The software is shared under opensource GNU license.

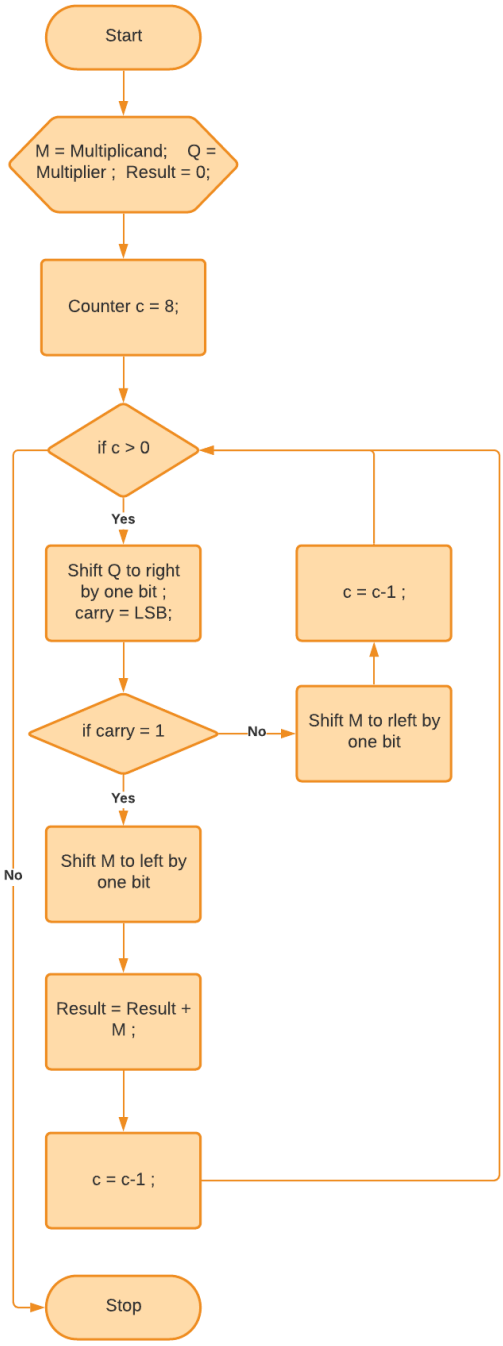
Link: [8085 Jubin Simulator]

Algorithm:



For 8-bit Binary Multiplication we start with Multiplicand M (the number to be multiplied) and Multiplier Q (the number that multiplies the Multiplicand). We take partial products of M with all bits of Q starting with LSB, shift each product towards left by one bit (except first partial product) and add them consecutively as shown in the diagram

Flow Chart:

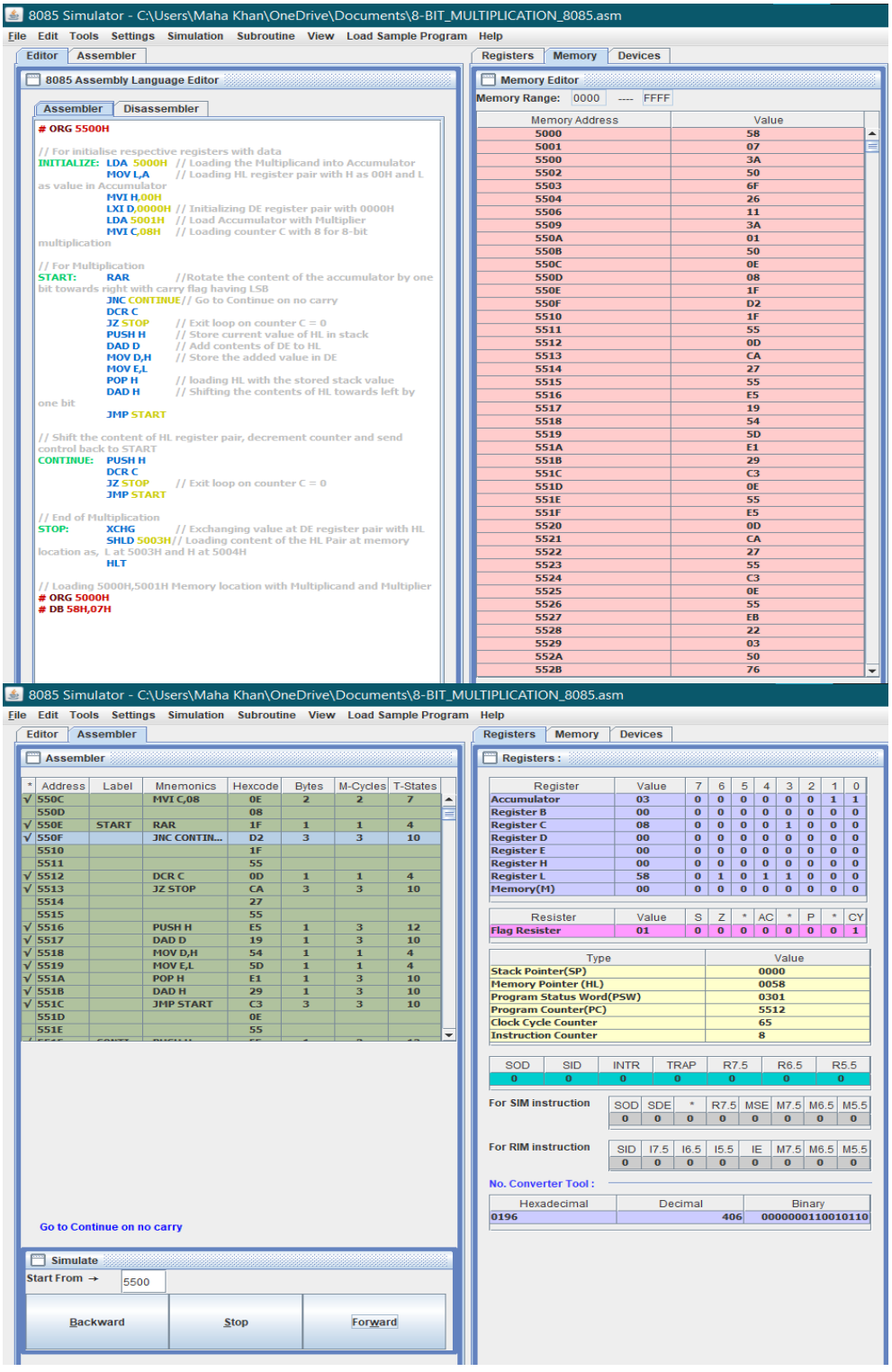

Start
M = Multiplicand; Q = Multiplier ; Result = 0;
Counter c = 8;
if c > 0
Yes
No
Shift Q to right by one bit ; carry = LSB;
if carry = 1
No
Yes
Shift M to rleft by one bit
c = c-1 ;
Shift M to left by one bit
Result = Result + M ;
c = c-1 ;
Stop

Program:

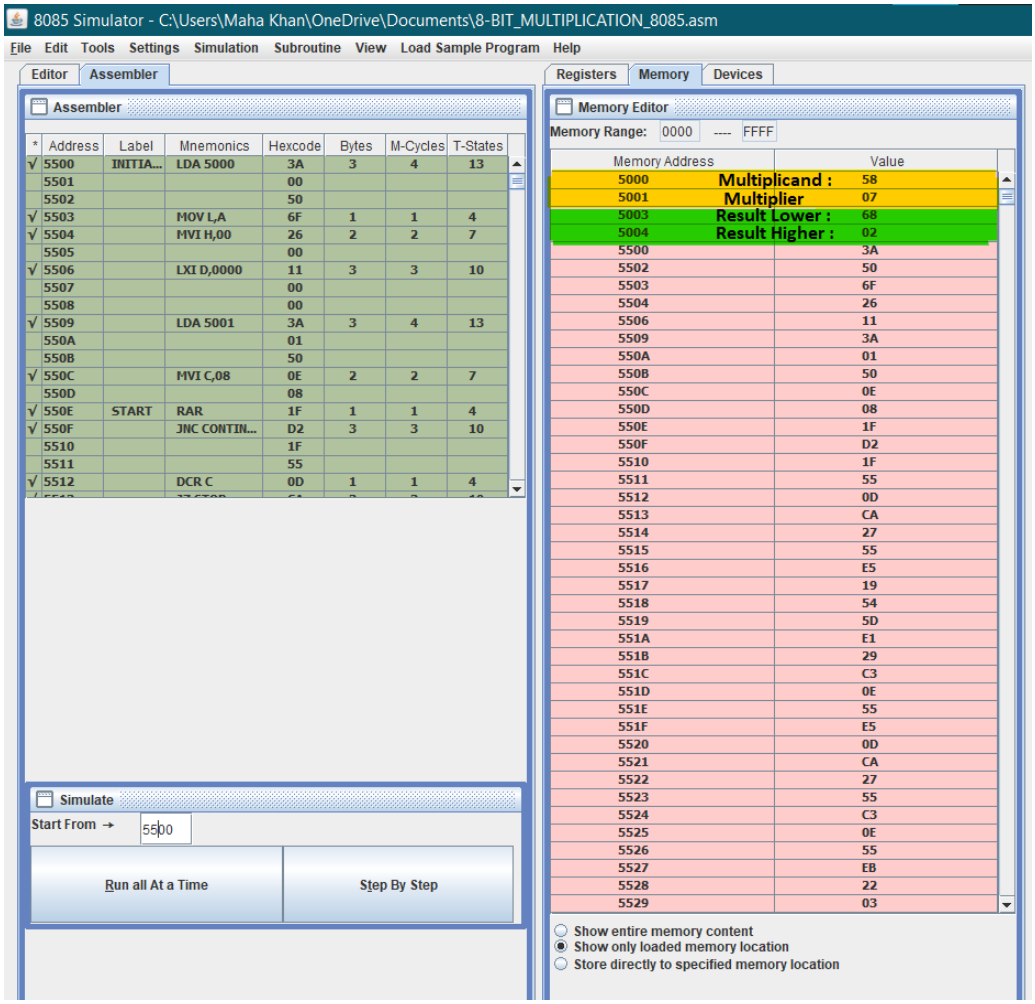
```
# ORG 5500H

// For initialize respective registers with data
INITIALIZE:
LDA  5000H  // Loading the Multiplicand into Accumulator
MOV L,A     // Loading HL register pair with H as 00H and L as value in Accumulator
MVI H,00H
LXI D,0000H // Initializing DE register pair with 0000H
LDA 5001H   // Load Accumulator with Multiplier
MVI C,08H   // Loading counter C with 8 for 8-bit multiplication

// For Multiplication
START:
        RAR             //Rotate the content of the accumulator by one bit towards right with carry flag having LSB
JNC CONTINUE // Go to Continue on no carry
DCR C
JZ STOP         // Exit loop on counter C = 0
PUSH H          // Store current value of HL in stack
DAD D           // Add contents of DE to HL
MOV D,H         // Store the added value in DE
MOV E,L
POP H           // loading HL with the stored stack value
DAD H           // Shifting the contents of HL towards left by one bit
JMP START

// Shift the content of HL register pair, decrement counter and send control back to START
CONTINUE:
        PUSH H
DCR C
JZ STOP         // Exit loop on counter C = 0
JMP START

// End of Multiplication
```

```
34. STOP:
35.         XCHG          // Exchanging value at DE register pair with HL
36. SHLD  5003H     // Loading content of the HL Pair at memory location as, L at
    5003H and H at 5004H
37. HLT
38.
39. // Loading 5000H,5001H Memory location with Multiplicand and Multiplier
40. # ORG 5000H
41. # DB 58H,07H
```

Screen-grab of Simulator:



Result:



Discussion:

In this experiment, I tested few techniques to multiply two 8-bit binary numbers. I used DAD and XCHG instruction specially to run the code more efficiently, and RAR instruction to rotate the multiplier with carry containing LSB towards right by 1 bit, and shifted current result (Shift and Add) left. Another approach to multiply is to use RAL instruction to shift multiplicand to left with carry containing MSB and the current result is shifted towards right till LSB of multiplicand is evaluated.