

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

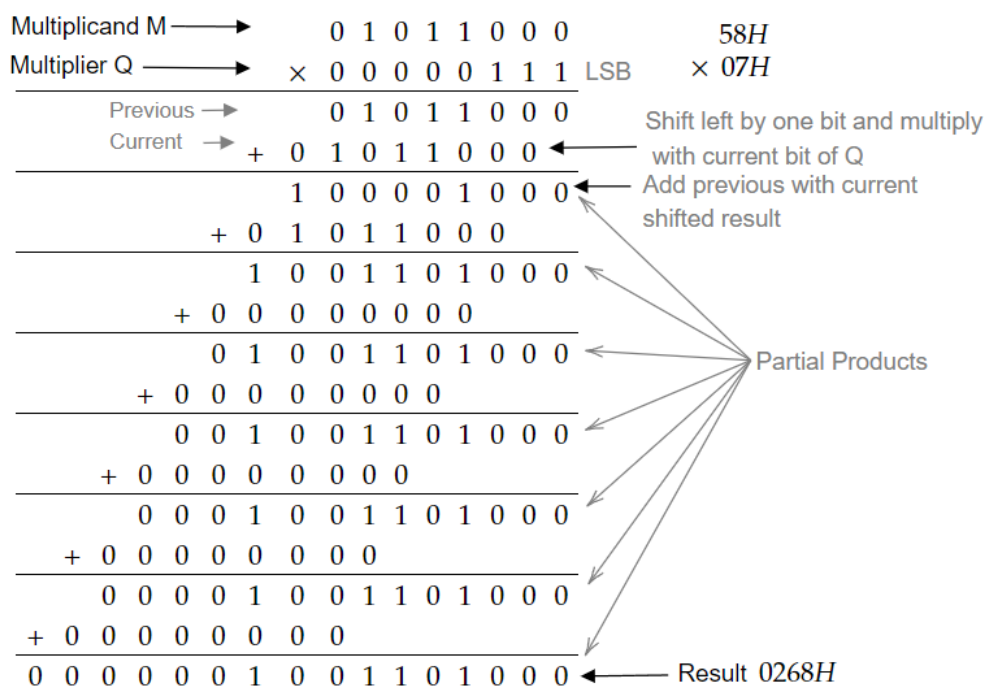
Simulator Used:

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.

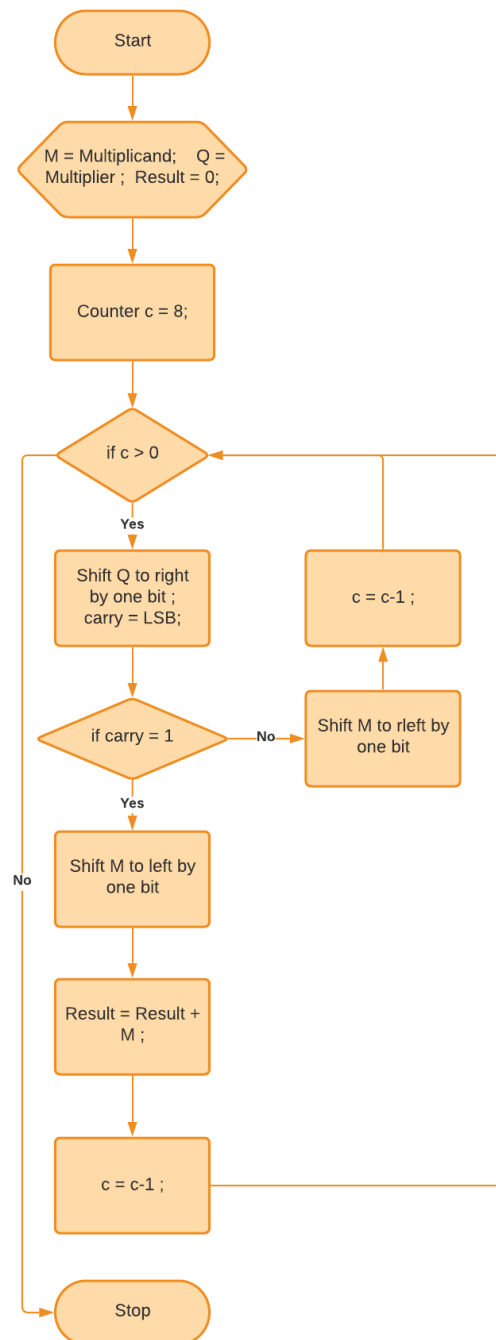
Link: [8085 Jubin Simulator](#)

Algorithm:

8-Bit Binary Multiplication using Shift/Add Method



Flow Chart:



Program:

```
1. # ORG 5500H
2.
3. // For initialize respective registers with data
4. INITIALIZE:
5. LDA 5000H // Loading the Multiplicand into Accumulator
6. MOV L,A   // Loading HL register pair with H as 00H and L as value in Accumulator
7. MVI H,00H
8. LXI D,0000H // Initializing DE register pair with 0000H
9. LDA 5001H // Load Accumulator with Multiplier
10. MVI C,08H // Loading counter C with 8 for 8-bit multiplication
11.
12. // For Multiplication
13. START:
14.     RAR           //Rotate the content of the accumulator by one bit towards right
    with carry flag having LSB
15. JNC CONTINUE // Go to Continue on no carry
16. DCR C
17. JZ STOP        // Exit loop on counter C = 0
18. PUSH H         // Store current value of HL in stack
19. DAD D          // Add contents of DE to HL
20. MOV D,H        // Store the added value in DE
21. MOV E,L
22. POP H          // loading HL with the stored stack value
23. DAD H          // Shifting the contents of HL towards left by one bit
24. JMP START
25.
26. // Shift the content of HL register pair, decrement counter and send control back to
    START
27. CONTINUE:
28.     PUSH H
29. DCR C
30. JZ STOP        // Exit loop on counter C = 0
31. JMP START
32.
33. // 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:

8085 Simulator - C:\Users\Maha Khan\OneDrive\Documents\8-BIT_MULTIPLICATION_8085.asm

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

8085 Assembly Language Editor

Assembler Disassembler

```
# 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
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
STOP: XCHG // Exchanging value at DE register pair with HL
SHLD 5003H // Loading content of the HL Pair at memory
location as, L at 5003H and H at 5004H
HLT

// Loading 5000H,5001H Memory location with Multiplicand and Multiplier
# ORG 5000H
# DB 58H,07H
```

Registers Memory Devices

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
5000	58
5001	07
5500	3A
5502	50
5503	6F
5504	26
5506	11
5509	3A
550A	01
550B	50
550C	0E
550D	08
550E	1F
550F	D2
5510	1F
5511	55
5512	0D
5513	CA
5514	27
5515	55
5516	E5
5517	19
5518	54
5519	5D
551A	E1
551B	29
551C	C3
551D	0E
551E	55
551F	E5
5520	0D
5521	CA
5522	27
5523	55
5524	C3
5525	0E
5526	55
5527	EB
5528	22
5529	03
552A	50
552B	76

8085 Simulator - C:\Users\Maha Khan\OneDrive\Documents\8-BIT_MULTIPLICATION_8085.asm

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 550C		MVI C,08	0E	2	2	7
✓ 550E	START	RAR	1F	1	1	4
✓ 550F		JNC CONTIN...	D2	3	3	10
✓ 5510			1F			
✓ 5511			55			
✓ 5512		DCR C	0D	1	1	4
✓ 5513		JZ STOP	CA	3	3	10
✓ 5514			27			
✓ 5515			55			
✓ 5516		PUSH H	E5	1	3	12
✓ 5517		DAD D	19	1	3	10
✓ 5518		MOV D,H	54	1	1	4
✓ 5519		MOV E,L	5D	1	1	4
✓ 551A		POP H	E1	1	3	10
✓ 551B		DAD H	29	1	3	10
✓ 551C		JMP START	C3	3	3	10
✓ 551D			0E			
✓ 551E			55			

Go to Continue on no carry

Simulate

Start From → 5500

Backward Stop Forward

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	03	0	0	0	0	0	0	1	1
Register B	00	0	0	0	0	0	0	0	0
Register C	08	0	0	0	0	1	0	0	0
Register D	00	0	0	0	0	0	0	0	0
Register E	00	0	0	0	0	0	0	0	0
Register H	00	0	0	0	0	0	0	0	0
Register L	58	0	1	0	1	1	0	0	0
Memory(H)	00	0	0	0	0	0	0	0	0

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	01	0	0	0	0	0	0	0	1

Type	Value
Stack Pointer(SP)	0000
Memory Pointer (HL)	0058
Program Status Word(PSW)	0301
Program Counter(PC)	5512
Clock Cycle Counter	65
Instruction Counter	8

SOD	SID	INTR	TRAP	R7.5	R6.5	R5.5
0	0	0	0	0	0	0

For SIM instruction

SOD	SDE	*	R7.5	MSE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

For RIM instruction

SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
0	0	0	0	0	0	0	0

No. Converter Tool :

Hexadecimal	Decimal	Binary
0196	406	0000000110010110

Result:

The screenshot shows the 8085 Simulator interface. The main window is divided into three panes: Assembler, Memory Editor, and Simulator. The Assembler pane displays a list of assembly instructions with their addresses, labels, mnemonics, hexcodes, bytes, M-Cycles, and T-States. The Memory Editor pane shows the memory range from 0000 to FFFF, with a table of memory addresses and values. The Simulator pane shows the start address 5500 and buttons for 'Run all at a Time' and 'Step By Step'.

Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
5500	INITIA...	LDA 5000	3A	3	4	13
5501			00			
5502			50			
5503		MOV L,A	6F	1	1	4
5504		MVI H,00	26	2	2	7
5505			00			
5506		LXI D,0000	11	3	3	10
5507			00			
5508			00			
5509		LDA 5001	3A	3	4	13
550A			01			
550B			50			
550C		MVI C,08	0E	2	2	7
550D			08			
550E	START	RAR	1F	1	1	4
550F		JNC CONTIN...	D2	3	3	10
5510			1F			
5511			55			
5512		DCR C	0D	1	1	4

Memory Address	Value
5000	58
5001	07
5003	68
5004	02
5500	3A
5502	50
5503	6F
5504	26
5506	11
5509	3A
550A	01
550B	50
550C	0E
550D	08
550E	1F
550F	D2
5510	1F
5511	55
5512	0D
5513	CA
5514	27
5515	55
5516	E5
5517	19
5518	54
5519	5D
551A	E1
551B	29
551C	C3
551D	0E
551E	55
551F	E5
5520	0D
5521	CA
5522	27
5523	55
5524	C3
5525	0E
5526	55
5527	EB
5528	22
5529	03

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.