

Experiment - I

Devansh Shukla
I18PH021

1 Aim:

Write a program using 8085 & verify for:

- Addition of two 8-bit numbers
- Addition of two 16-bit numbers (with carry)

2 Theory:

The 8085 is an 8-bit microprocessor produced by Intel.
It has the following configuration –

- 8-bit data bus
- 16-bit address bus, which can address upto 64KB
- A 16-bit program counter
- A 16-bit stack pointer
- Six 8-bit registers arranged in pairs: BC, DE, HL
- Requires +5V supply to operate at 3.2 MHz single phase clock

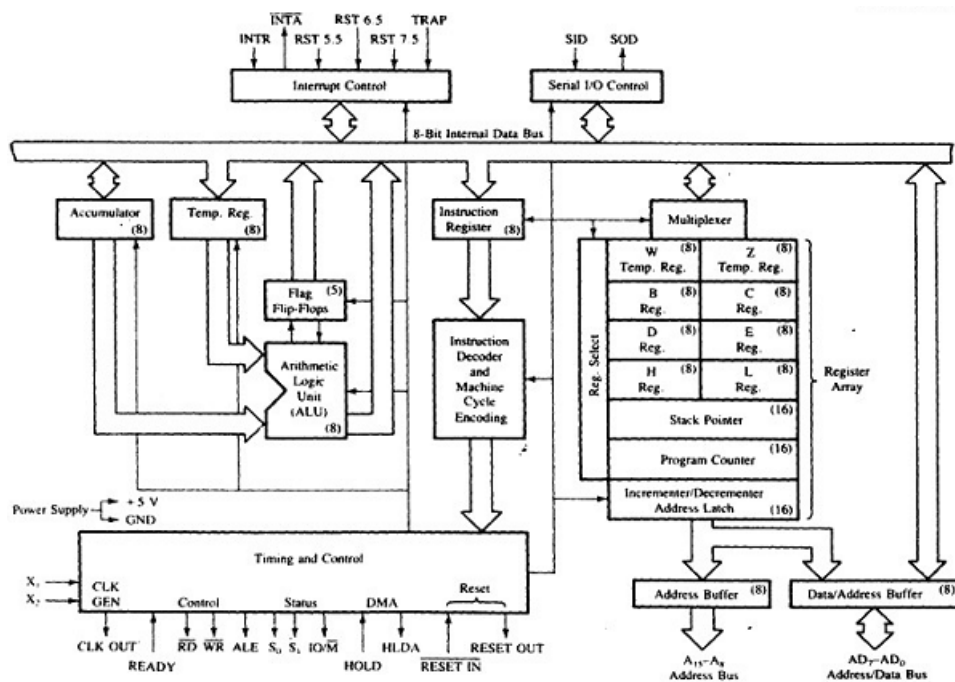


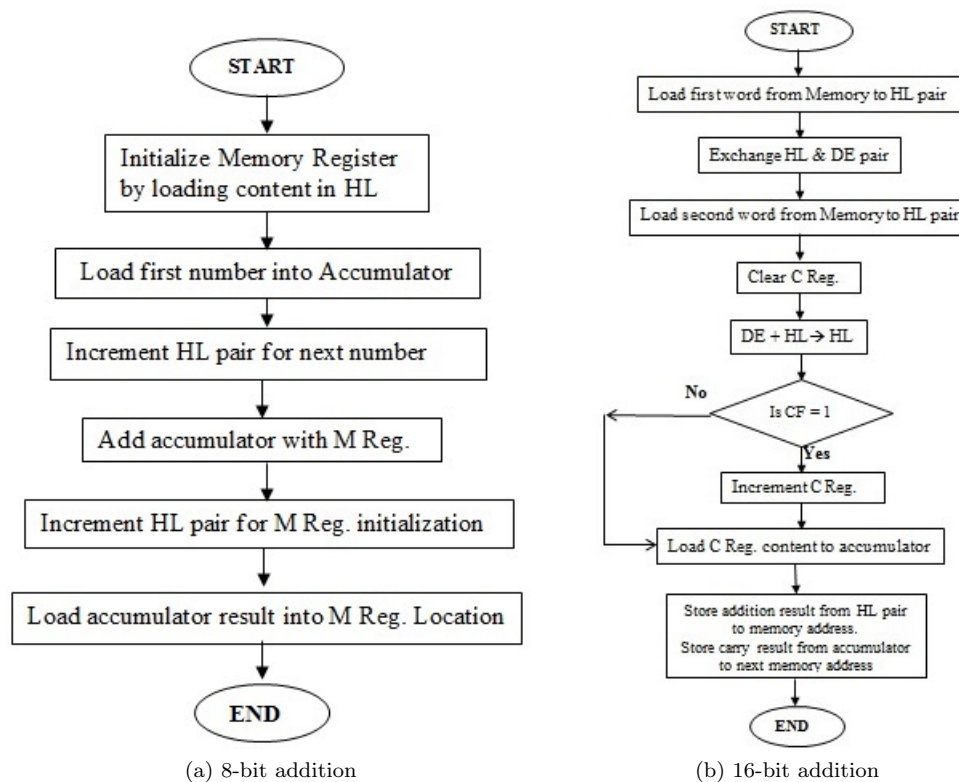
Figure 1: 8085 Architecture

2.1 Some instructions

1. **ORG** address
(Origin) directive reserves the starting address for program code or data in specified memory array,
2. **LXI H**
(Load register pair immediately) loads 16 bit data in register pair designated by operand.
3. **LHLD** addr
(Load HL pair direct) loads 16 bit data from specified address to designate in register pair.
4. **MOV A, M**
(Move M to A) copies the data byte into accumulator from the memory specified by the address in H-L pair.
5. **MVI**
(Move immediate data) moves immediate value to specified register.

6. **DAD**
(Double addition) Add specified register pair content to HL pair content and store results into HL pair.
7. **JNC addr**
(Jump if no carry) Instruction jump the execution to the specified Address if carry flag is reset.
8. **INR R**
(Increment register) increment the specified register content by 1.
9. **INX H**
(Increment register pair) increments the contents of the register pair by one.
10. **ADD M**
(Addition) adds the contents of memory to accumulator.
11. **STA addr**
(Store accumulator direct) copies the contents of the accumulator to the memory location specified in the instruction
12. **SHLD addr**
(Store HL direct) instruction store HL pair content to specified address.
13. **RST 1**
(Reset) finishes the execution of the current instruction and stops any further execution.
14. **DB**
(Define byte) directive is defined to store values in specified memory array.

2.2 Flowchart



3 Code

3.1 A) Addition of two 8-bit numbers

```

# ORG 7000H

LXI H,7501          // Get address of 1st no. in HL pair
MOV A,M            // Move no. into accumulator
INX H              // HL points the address 7502 H
ADD M              // Add the 2nd no.
INX H              // HL points 7503 H
MOV M,A            // Store result in 7503 H
RST 1              // Terminate

# ORG 7501H          // Store input at the address
# DB 12H, 13H       // Get two 8 bit no. in successive location
  
```

3.2 B) Addition of two 16-bit numbers

```
# ORG 7000H

LHLD 7601      // Get 1st no. in HL pair from memory 7601 H
XCHG          // Exchange cont. of DE HL
LHLD 7603      // Get 2st no. in HL pair from location 7603 H
MVI C,00       // Clear reg. C.
DAD D         // Get HL+DE & store result in HL
JNC down       // If no carry move to loop/if carry then move to next step.
INR C         // Increment reg.C
MOV A,C        // Move carry from reg. C to reg. A
STA 7502       // Store carry at 7502 H
down: SHLD 7500 // Store result in 7500 H.
RST 1         // Terminate

# ORG 7601H    // Store input at the address
# DB 13,31,12,10 // Get two 16 bit no. in successive location
```

4 Observations:

4.1 A) Addition of two 8-bit numbers

The screenshot displays the 8085 Assembler-Simulator interface. The main window is divided into several sections:

- Editor/Assembler:** Shows the assembly code being processed. The address range is from 7000 to 7008. The instructions are:

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
7000		LXI H,7501	21	3	3	10
7001			01			
7002			75			
7003		MOV A,M	7E	1	2	7
7004		INX H	23	1	1	6
7005		ADD M	86	1	2	7
7006		INX H	23	1	1	6
7007		MOV M,A	77	1	2	7
7008		RST 1	CF	1	3	12
- Registers:** Shows the current state of the 8085 registers.

Register	Value	7	6	5	4	3	2	1	0
Accumulator	25	0	0	1	0	0	1	0	1
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	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	75	0	1	1	1	0	1	0	1
Register L	03	0	0	0	0	0	0	1	1
Memory(M)	25	0	0	1	0	0	1	0	1
- Flag Register:** Shows the status of the flags.

Register	Value	S	Z	*	AC	*	P	*	CY
Flag Register	00	0	0	0	0	0	0	0	0
- System Variables:** Shows the current state of various system variables.

Type	Value
Stack Pointer(SP)	FFFE
Memory Pointer (HL)	7503
Program Status Word(PSW)	2500
Program Counter(PC)	0008
Clock Cycle Counter	55
Instruction Counter	7
- Simulation Controls:** Includes buttons for "Run all At a Time" and "Step By Step".
- Converter Tool:** A section for converting between Hexadecimal, Decimal, and Binary values.

Figure 2: (a) 8-bit addition

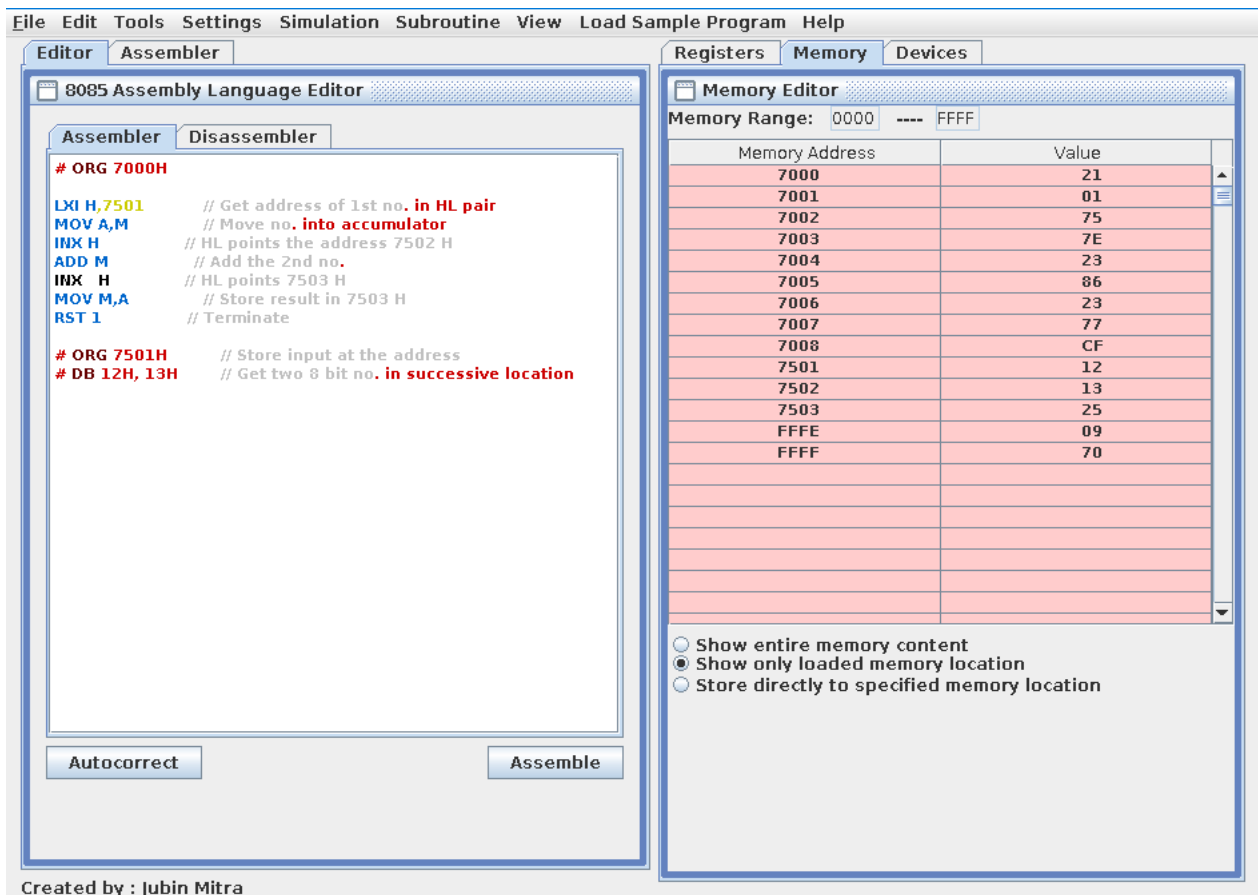


Figure 3: (b) 8-bit addition

4.2 B) Addition of two 16-bit numbers

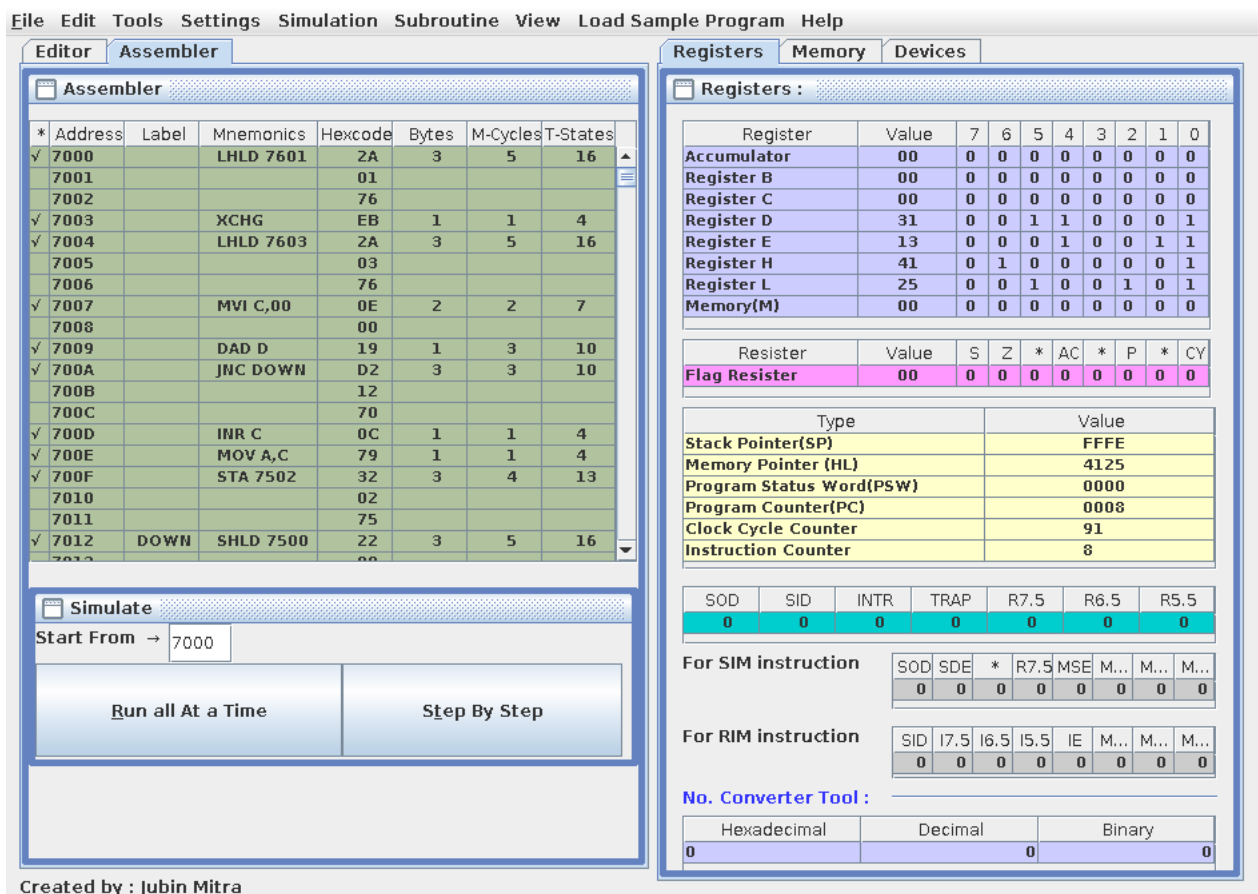


Figure 4: (a) 16-bit addition

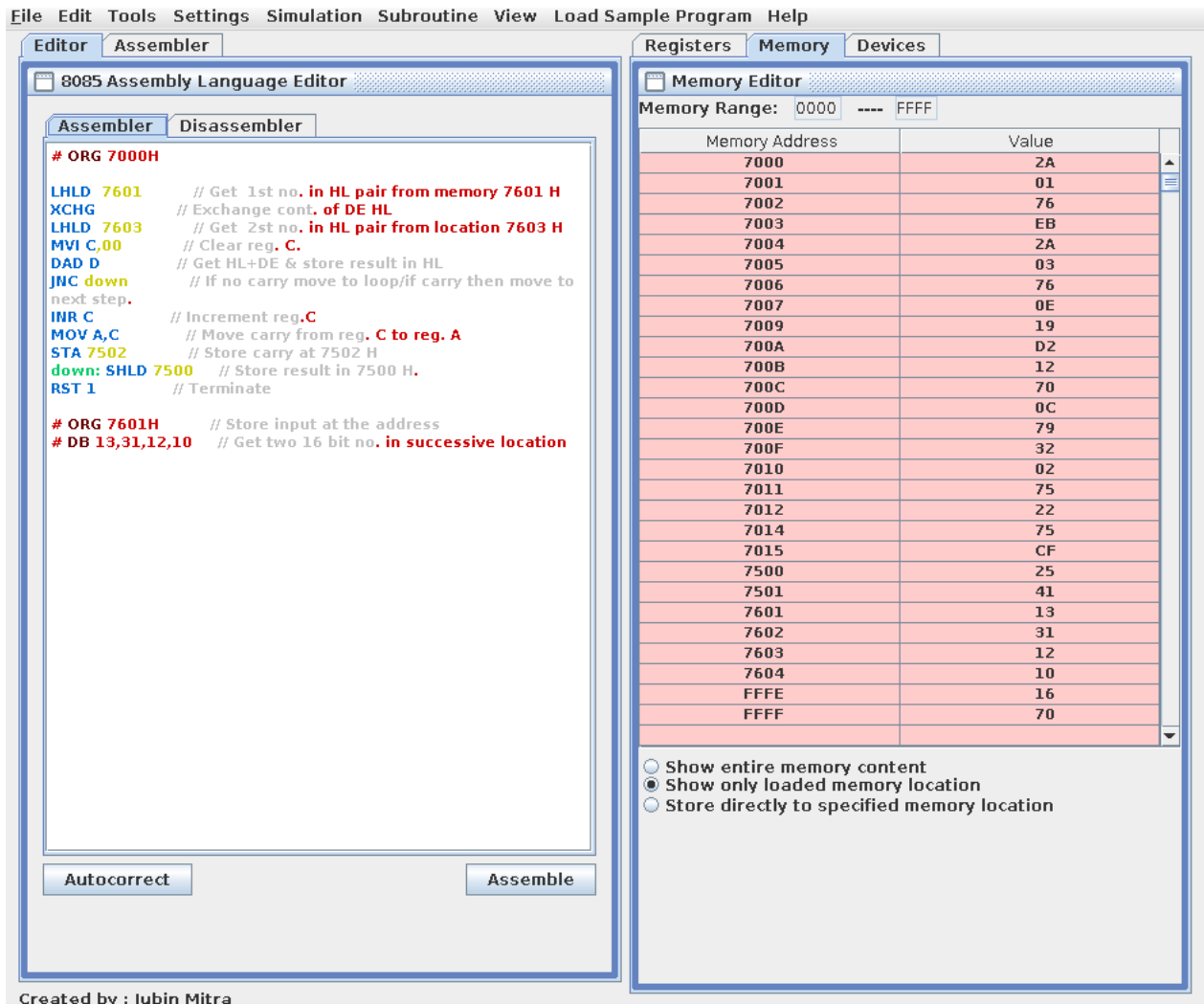


Figure 5: (b) 16-bit addition

5 Conclusion:

(A)
Input: 7501 – 13H, 7502 – 12H
Output: A – 25H, 7503 – 25H
(B)
Input: 7601 – 13H, 7602 – 31H, 7603 – 12H, 7604 – 10H
Output: 7500 – 25H, 7501 – 41H, 7502 – 00H

Hence the programs given in [section 3](#) works as expected for 8085 microprocessor.