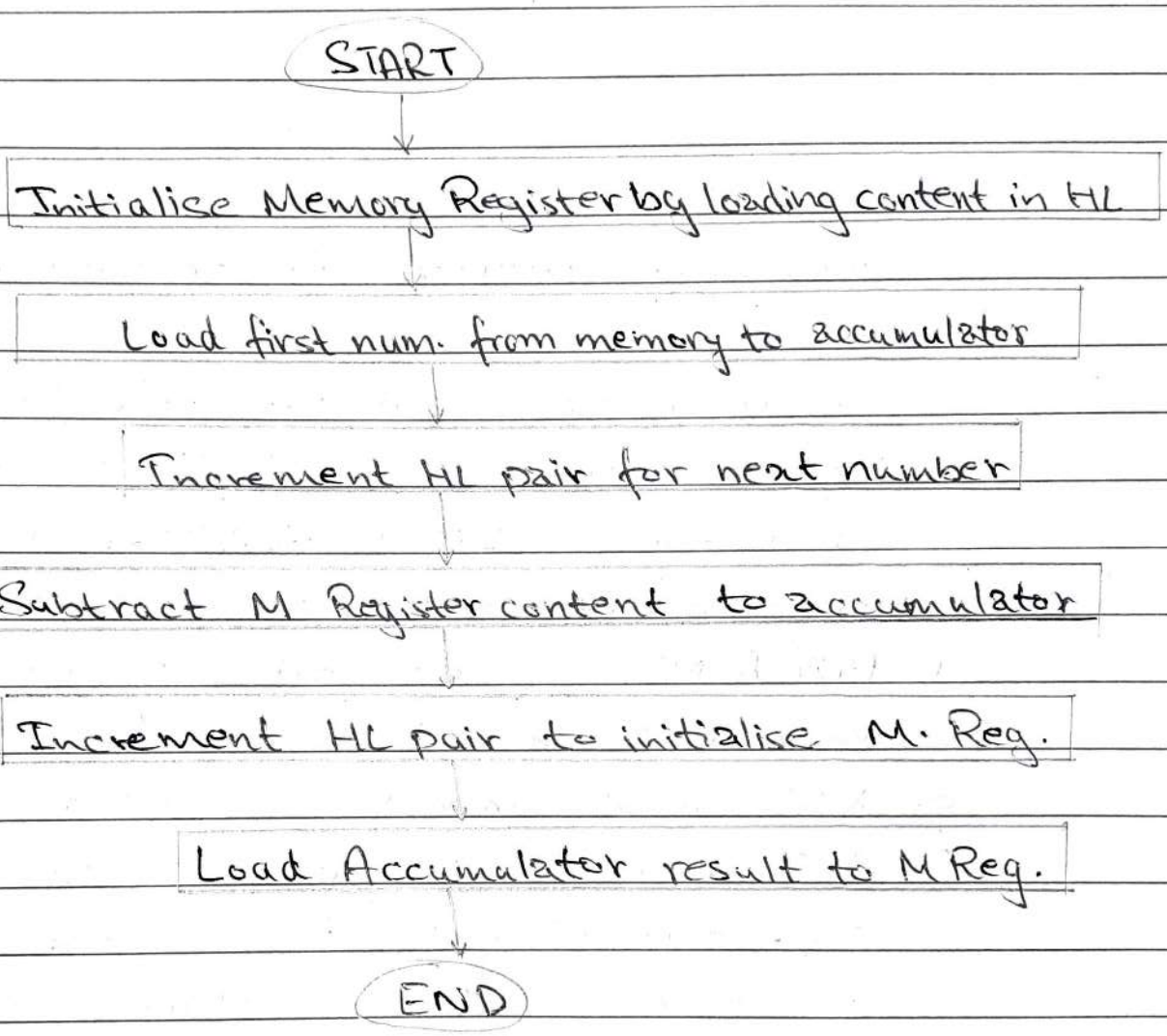
29/09/2020
Experiment-02

AIM: SUBTRACTION OF TWO 8-BIT NUMBERS AND TWO 16-BIT NUMBERS.

THEORY:

1. ORG Address Directive reserves the starting code address for Program Code or data in specified memory array.
2. LXI H loads 16 bit data in register pair designated by operand.
3. LHLD Address (LOAD HL PAIR DIRECT) loads 16 bit data from specified address to designate in register pair.
4. MOV A,M copies data byte into accumulator from the memory specified by the address in HL pair.
5. MVI moves immediate value to specified register.
6. SBB instruction subtracts specified register content and carry flag to Accumulator and stores result in the Accumulator.
7. INC Address instruction jumps the execution to the specified Address if carry flag is reset.
8. INR instruction increments specified register content by 1 value.
9. INX H increments contents of register pair by 1.
10. SUB M subtracts contents of register to accumulator.
11. STA address copies the contents of the accumulator to the memory location specified in the instruction.
12. SHLD Address instruction stores HL pair content to specified address.
13. RST 1 finishes the execution of the current instruction and stops further execution.

FLOWCHART:



PROGRAM (Subtraction of two 8-bit numbers)

Address	Mnemonics	Comment	Hexcode.
	#ORG 7000H		
7000	LXI H, 7501	// Get address of 1st no. in HL pair	21
7001			01
7002			75
7003	MOV A, M	// Move no. into accumulator	7E
7004	INX H	// HL points to 7502 H	23
7005	SBB M	// Subtract 2nd no from 1st no.	9E
7006	INX H	// HL points to 7503H	23
7007	MOV M, A	// Move contents of acc. to memory	77

7008	RST 1	// Terminate	CF
	#ORG 7501H	// Store no. at address	
	#DB 20, 10	// Get two 8 bit no. at successive locations	

PROGRAM (Subtraction of two 16- bit numbers)

Address	Mnemonics	Comments	Hexcodes
	#ORG 7000H		
7000	LHLD 7501	// Get 1st 16 bit no. in HL Pair	2A
7001			01
7002			75
7003	XCHG	// Exchange HL pair with DE	EB
7004	LHLD 7503	// Get 2nd 16 bit no. in HL pair.	2A
7005			03
7006			75
7007	MOV A, E	// Get lower byte of 1st number	7B
7008	SUB L	// Subtract lower byte of 2nd number.	95
7009	MOV L, A	// Store the result in reg. L.	6F
700A	MOV A, D	// Get higher byte of 1st number.	7A
700B	SBB H	// Subtract higher byte of 2nd no. with borrow	9C
700C	MOV H, A	// Mov from acc. to H	67
700D	SHLD 7505	// Store 16 bit result at 7505H & 7506H	22
700E			05
700F			75
7010	RST 1	// Terminate	CF
	#ORG 7501H	// Stores inputs at the address	
	#DB 30,40,10,20	// Get two 16 bit nos. from successive locations.	

__/__/____

RESULT:

for subtraction of two ~~bit~~ 8-bit numbers,

INPUT - 7501 - 20H ; 7502 - 10H

OUTPUT - 7503 - 10H

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor | Assembler

8085 Assembly Language Editor

Assembler | Disassembler

```
# ORG 7000H
        LXI H,7501
        MOV A,M
        INX H
        SBB M
        INX H
        MOV M,A
        RST 1
# ORG 7501H
# DB 20, 10
```

Autocorrect | Assemble

Registers | Memory | Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	10	0	0	0	1	0	0	0	0
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)	10	0	0	0	1	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	FFFE
Memory Pointer (HL)	7503
Program Status Word(PSW)	1000
Program Counter(PC)	0008
Clock Cycle Counter	55
Instruction Counter	7

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
0	0	0

Created by : Jubin Mitra

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor | Assembler

Assembler

*	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		SBB M	9E	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

Simulate

Start From → 7000

Run all At a Time | Step By Step

Created by : Jubin Mitra

Registers | Memory | Devices

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
7000	21
7001	01
7002	75
7003	7E
7004	23
7005	9E
7006	23
7007	77
7008	CF
7501	20
7502	10
7503	10
FFFE	09
FFFF	70

Show entire memory content
Show only loaded memory location
Store directly to specified memory location

__/__/____

for subtraction of two - 16-bit numbers.

INPUT - 7501 - 30H ; 7502 - 40H

7503 - 10H ; 7504 - 20H

OUTPUT - 7505 - 20H

7506 - 20H

for subtraction of two - 16-bit numbers.

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor | Assembler

8085 Assembly Language Editor

Assembler | Disassembler

```
# ORG 7000H
        LHLD 7501
        XCHG
        LHLD 7503
        MOV A,E
        SUB L
        MOV L,A
        MOV A,D
        SBB H
        MOV H,A
        SHLD 7505
        RST 1
# ORG 7501H
# DB 30, 40, 10, 20
```

Autocorrect | Assemble

Created by : Jubin Mitra

Registers | Memory | Devices

Registers :

Register	Value	7	6	5	4	3	2	1	0
Accumulator	20	0	0	1	0	0	0	0	0
Register B	00	0	0	0	0	0	0	0	0
Register C	00	0	0	0	0	0	0	0	0
Register D	40	0	1	0	0	0	0	0	0
Register E	30	0	0	1	1	0	0	0	0
Register H	20	0	0	1	0	0	0	0	0
Register L	20	0	0	1	0	0	0	0	0
Memory(M)	00	0	0	0	0	0	0	0	0

Resister	Value	S	Z	*	AC	*	P	*	CY
Flag Resister	00	0	0	0	0	0	0	0	0

Type	Value
Stack Pointer(SP)	FFFE
Memory Pointer (HL)	2020
Program Status Word(PSW)	2000
Program Counter(PC)	0008
Clock Cycle Counter	88
Instruction Counter	11

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
0	0	0

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor | Assembler

Registers | Memory | Devices

Assembler

*	Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
√	7000		LHLD 7501	2A	3	5	16
	7001			01			
	7002			75			
√	7003		XCHG	EB	1	1	4
√	7004		LHLD 7503	2A	3	5	16
	7005			03			
	7006			75			
√	7007		MOV A,E	7B	1	1	4
√	7008		SUB L	95	1	1	4
√	7009		MOV L,A	6F	1	1	4
√	700A		MOV A,D	7A	1	1	4
√	700B		SBB H	9C	1	1	4
√	700C		MOV H,A	67	1	1	4
√	700D		SHLD 7505	22	3	5	16
	700E			05			
	700F			75			
√	7010		RST 1	CF	1	3	12

Simulate

Start From → 7000

Run all At a Time | Step By Step

Created by : Jubin Mitra

Memory Editor

Memory Range: 0000 ---- FFFF

Memory Address	Value
7000	2A
7001	01
7002	75
7003	EB
7004	2A
7005	03
7006	75
7007	7B
7008	95
7009	6F
700A	7A
700B	9C
700C	67
700D	22
700E	05
700F	75
7010	CF
7501	30
7502	40
7503	10
7504	20
7505	20
7506	20
FFFE	11
FFFF	70

- ○ Show entire memory content
- ◉ Show only loaded memory location
- ○ Store directly to specified memory location