Microprocessor, Assembly Language & Computer Interfacing Sessional

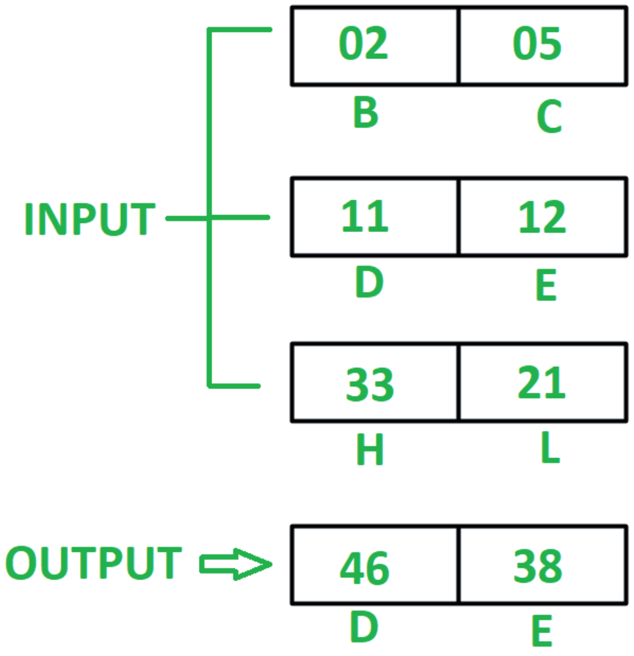
EEE-3212

Fareha Akter Juha
Adjunct Faculty, Dep. of CSE,
Leading University, Sylhet
Email: farehaakter02@gmail.com

Lab-7: 1’s and 2’s complement using the 8085 microprocessor.

8085 program to add three 16 bit numbers stored in registers

Problem – Write an assembly language program to add three 16 bit numbers stored in register HL, DE, BC and store the result in DE with minimum number of instructions.



INPUT

02 05
B C

11 12
D E

33 21
H L

OUTPUT

46 38
D E

Program

using memory location

```
LHLD  2000
MOV  C,L
MOV B,H
LHLD 2002
XCHG
LHLD 2004
DAD D
MOV D,B
MOV E,C
DAD D
XCHG
HLT
```

Without using memory location

```
MVI B,02
MVI C,05
MVI D,11
MVI E,12
MVI H,33
MVI L,21
DAD D
MOV D,B
MOV E,C
DAD D
XCHG
HLT
```

ALGORITHM

1. Assumptions –

1.Number to be added are already stored in register HL, DE, BC

2.Numbers stored in the register are such that final result should not be greater than

FFFF **DAD D** performs the following task:

H <- H + D

L <- L + E

DAD instruction takes one argument and that argument can be

register B, D, H, or SP **XCHG** instruction exchanges the content of register D with H

and E with L

ALGORITHM

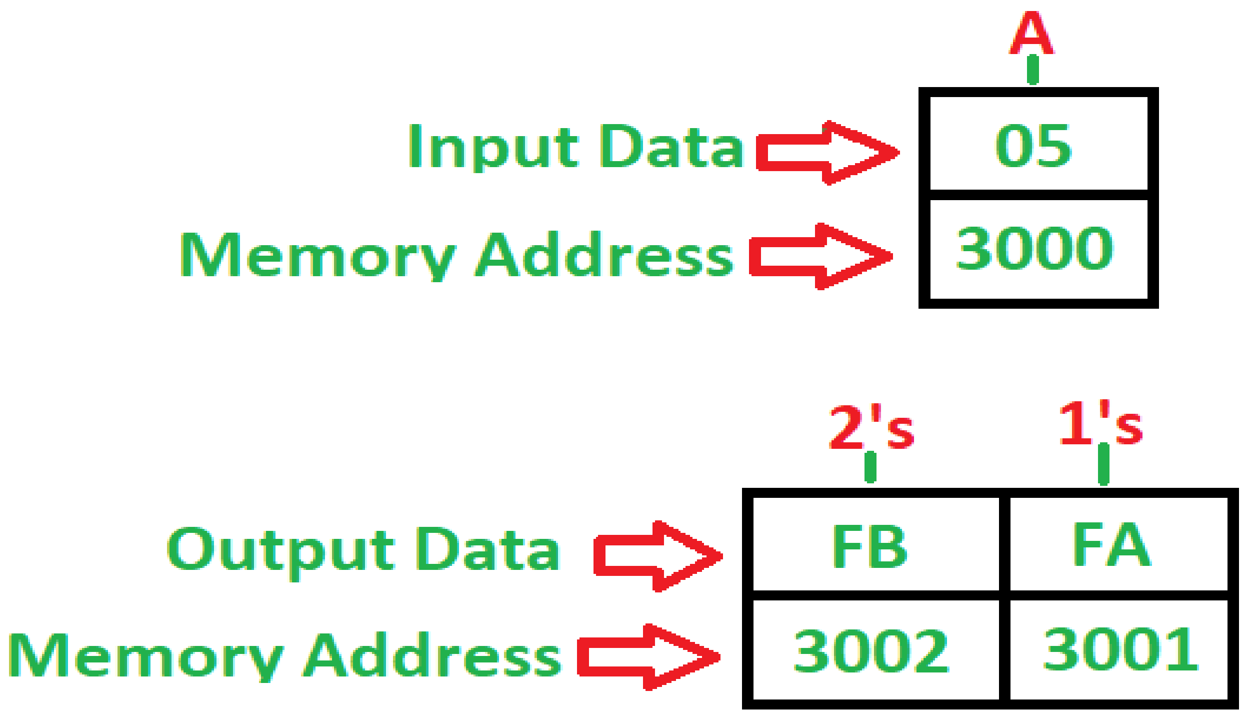
- Add the content of DE register in HL and store the result in HL by help of DAD instruction
- Move the content of register B in D and C in E
- Repeat step 1
- Use XCHG instruction to swap the content of DE with HL. We will get the result in DE

EXPLANATION –

- DAD D – adds the content of register D in H and register E in L and store the result in HL
- MOV D, B – moves the value of register B in register D
- MOV E, C moves the value of register C in register E
- Same as step 1
- XCHG – exchange the content of register H with register D and L with E.
- HLT – stops executing the program and halts any further execution

8085 program to find 1's and 2's complement of 8-bit number

Problem–Write a program to find 1's and 2's complement of 8-bit number where starting address is **2000** and the number is stored at **3000** memory address and store result into **3001** and **3002** memory address.



A

Input Data → 05

Memory Address → 3000

2's 1's

Output Data → FB FA

Memory Address → 3002 3001

ALGORITHM

- Load the data from memory 3000 into A (accumulator)
- Complement content of accumulator
- Store content of accumulator in memory 3001 (1’s complement)
- Add 01 to Accumulator content
- Store content of accumulator in memory 3002 (2’s complement)
- Stop

Program

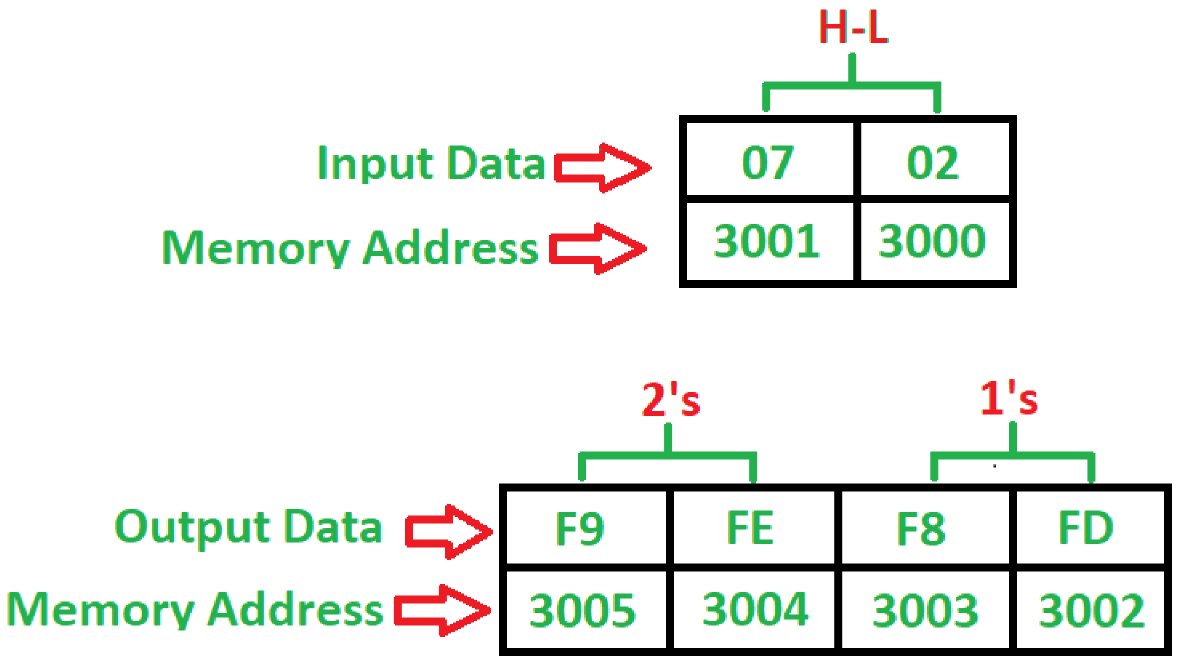
Memory	Mnemonics	Operands	Comment
2000	LDA	[3000]	[A] <- [3000]
2003	CMA		[A] <- [A^]
2004	STA	[3001]	1’s complement
2007	ADI	01	[A] <- [A] + 01
2009	STA	[3002]	2’s complement
200C	HLT		Stop

EXPLANATION –

1. A is an 8-bit accumulator which is used to load and store the data directly
2. LDA is used to load accumulator direct using 16-bit address (3 Byte instruction)
3. CMA is used to complement content of accumulator (1 Byte instruction)
4. STA is used to store accumulator direct using 16-bit address (3 Byte instruction)
5. ADI is used to add data into accumulator immediately (2 Byte instruction)
6. HLT is used to halt the program

8085 program to find 1’s and 2’s complement of 16-bit number

Problem – Write a program to find 1’s and 2’s complement of 16-bit number where starting address is 2000 and the number is stored at 3000 memory address and store result into 3002 and 3004 memory address.



Input Data	07	02
Memory Address	3001	3000

Output Data	F9	FE	F8	FD
Memory Address	3005	3004	3003	3002

ALGORITHM

- Load a 16-bit number from memory 3000 into a register pair (H-L)
- Move content of register L to accumulator
- Complement content of accumulator
- Move content of accumulator to register L
- Move content of register H to accumulator
- Complement content of accumulator
- Move content of accumulator to register H
- Store content of register pair in memory 3002 (1’s complement)
- Increment content of register pair by 1
- Store content of register pair in memory 3004 (2’s complement)
- Stop

Program

Memory	Mnemonics	Operands	Comment
2000	LHLD	[3000]	[H-L] <- [3000]
2003	MOV	A, L	[A] <- [L]
2004	CMA		[A] <- [A^]
2005	MOV	L, A	[L] <- [A]
2006	MOV	A, H	[A] <- [H]
2007	CMA		[A] <- [A^]
2008	MOV	H, A	[H] <- [A]
2009	SHLD	[3002]	1’s complement
200C	INX	H	[H-L] <- [H-L] + 1
200D	SHLD	[3004]	2’s complement
2010	HLT		Stop

EXPLANATION –

- A is an 8-bit accumulator which is used to load and store the data
- LHLD is used to load register pair H-L direct using 16-bit address (3 Byte instruction)
- MOV is used to transfer the data from accumulator to register(any) or register(any) to accumulator (1 Byte)
- CMA is used to complement content of accumulator (1 Byte instruction)
- SHLD is used to store data from register pair H-L into memory direct using 16-bit address (3 Byte instruction)
- INX is used to increase H-L register pair by 1 (1 Byte instruction)
- HLT is used to halt the program

Thank You