Experiment No

Objective:

Write an assembly language program for **8 bit addition, subtraction**.

Prerequisite:

TASM assembler

<u>Algorithm for 8 bit addition</u>:

1) Start
2) Initialize data segment through AX register in the DS register.
3) Display the message as "Enter the first number"
4) Read first digit in AL register through keyboard (e.g. AL=31h)
5) Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number.AL=01h
6) Move contents of AL register to a BL. (BL← AL so BL=01h)
7) Rotate the contents of BL register by 4 positions at left side. (BL=10h)
8) Read a second digit in AL register through keyboard AL=35h
9) Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number. AL=05h
10) Add the contents of BL and AL store the result in BL (BL←BL+AL so BL=15h)
11) Display the message as "Enter the second number"
12) Read first digit in AL register through keyboard AL=32h
13) Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number.AL=02h
14) Move contents of AL register to a CL. (CL← AL so CL=02h)
15) Rotate the contents of CL register by 4 positions at left side. (CL=20h)
16) Read a second digit in AL register through keyboard (AL=33h)

17) Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number. AL=03h

18) Add the contents of CL and AL store the result in CL (CL←CL+AL so CL=23h)

(Now both numbers are accepted as 15h and 23h)

19) Add the contents of BL and CL and result gets stored in BL (E.g ADD BL,CL so BL=38h)

20) Preserve the result of addition in some temporary variable say temp from BL.

21) Mask the first nibble by AND operation with number F0h (AND BL,F0h so BL=30h)

22) Call Output procedure with BL register to make a digit back in ASCII hexadecimal range (BL=33h)

23) Move the contents of BL to DL and display it on the screen

24) Move result from temporary variable to BL again (So BL=38h)

25) Mask the second nibble by AND operation with number 0Fh (AND BL,0Fh so BL=08h)

26) Call Output procedure with BL register to make a digit back in ASCII hexadecimal range (BL=38h)

27) Move the contents of BL to DL and display it on the screen

28) Stop

Algorithm for Input procedure:(To accept input from 0 to F)

1) Compare the contents of AL with 41h.
2) Jump to step no 4 if carry flag is set(digit is in the range of 0 to 9 so add only 30h)
3) Sub 07h to AL register(If digit is in the range from A to F then add 30h and 7h both)
4) Sub 30h to AL register
5) Return

Algorithm for Output procedure:

1) Compare the contents of BL with 0Ah
2) Jump to step no 4 if carry flag is set(digit is in the range of 0 to 9 so add only 30h)
3) Add 07h to BL register(If digit is in the range from A to F then add 30h and 7h both)
4) Add 30h to BL register
5) Return

Note:

While masking F or f is not case sensitive. But in input procedure 41h number is considered for comparison because 41h is ASCII hex value for 'A'. In output procedure '0A' is considered not 'a' is considered as small case a has 61h ASCII hex value. So this input and output procedure are applicable for only capital 'A' to 'F'

<u>Algorithm for 8 bit subtraction</u>:

1. Start
2. Initialize data segment through AX register in the DS register.
3. Display the message as "Enter the first number"
4. Read first digit in AL register through keyboard (e.g. AL=31h)
5. Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number.AL=01h
6. Move contents of AL register to a BL. (BL← AL so BL=01h)
7. Rotate the contents of BL register by 4 positions at left side. (BL=10h)
8. Read a second digit in AL register through keyboard AL=35h
9. Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number. AL=05h
10. Add the contents of BL and AL store the result in BL (BL←BL+AL so BL=15h)
11. Display the message as "Enter the second number"

12. Read first digit in AL register through keyboard AL=32h

13. Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number.AL=02h

14. Move contents of AL register to a CL. (CL← AL so CL=02h)

15. Rotate the contents of CL register by 4 positions at left side. (CL=20h)

16. Read a second digit in AL register through keyboard (AL=33h)

17. Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number. AL=03h

18. Add the contents of CL and AL store the result in CL (CL←CL+AL so CL=23h) (Now both numbers are accepted as 15h and 23h)

19. Subtract the contents of CL from BL and result gets stored in BL (E.g SUB BL,CL so BL=F2h)

20. Preserve the result in some temporary variable say temp from BL.

21. Mask the first nibble by AND operation with number F0h (AND BL,F0h so BL=30h)

22. Call Output procedure with BL register to make a digit back in ASCII hexadecimal range (BL=33h)

23. Move the contents of BL to DL and display it on the screen

24. Move result from temporary variable to BL again (So BL=38h)

25. Mask the second nibble by AND operation with number 0Fh (AND BL,0Fh so BL=08h)

26. Call Output procedure with BL register to make a digit back in ASCII hexadecimal range (BL=38h)

27. Move the contents of BL to DL and display it on the screen

28. Stop

Algorithm for Input procedure:(To accept input from 0 to f)

1. Compare the contents of AL with 41h

2. Jump to step no 4 if carry flag is set
3. Sub 07h to AL register
4. Sub 30h to AL register
5. Return

Algorithm for Output procedure:

1. Compare the contents of BL with 0Ah
2. Jump to step no 4 if carry flag is set
3. Add 07h to BL register
4. Add 30h to BL register
5. Return

Note:

While masking F or f is not case sensitive. But in input procedure 41h number is considered for comparison because 41h is ASCII hex value for ‘A’. In output procedure ‘0A’ is considered not ‘a’ is considered as small case a has 61h ASCII hex value.So this input and output procedure are applicable for only capital ‘A’ to ‘F’.