Experiment No

Objective:

Write an assembly language program to add/subtract/multiply/divide two 16-bit numbers.

Prerequisite:

TASM assembler

Algorithm

1. Start
2. Initialize data segment through AX register in the DS register.
3. Display the 3 text message as "1. 16 bit addition 2 .16 bit subtraction 3. Exit Enter your choice"
4. Compare accepted choice with 03h.
5. If zero flag is set then goto step no.6 otherwise goto step no 7
6. Exit the program
7. Display the message as "Enter first 16 bit number"
8. Read first digit in AL register through keyboard (e.g. AL=32h)
9. Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number.AL=02h
10. Move contents of AH with 00h. (AH← 00h so AX=0002h)
11. Rotate AX contents in left directions by 12 bits. (AX=2000h)
12. Move the contents of AX to BX(BX←AX so BX=2000h)
13. Read a second digit in AL register through keyboard AL=35h
14. Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number. AL=05h
15. Move contents of AH with 00h. (AH← 00h so AX=0005h)
16. Rotate AX contents in left directions by 8 bits. (AX=0500h)
17. Add the contents of AX and BX (BX=BX+AX so BX=2500h)
18. Read a third digit in AL register through keyboard AL=31h

19. Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number. AL=01h

20. Move contents of AH with 00h. (AH← 00h so AX=0001h)

21. Rotate AX contents in left directions by 4 bits. (AX=0010h)

22. Add the contents of AX and BX (BX=BX+AX so BX=2510h)

23. Read a fourth digit in AL register through keyboard AL=30h

24. Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number. AL=00h

25. Move contents of AH with 00h. (AH← 00h so AX=0000h)

26. Add the contents of AX and BX (BX=BX+AX so BX=2510h)

27. Display the message as "Enter second 16 bit number"

28. Read first digit in AL register through keyboard (e.g. AL=37h)

29. Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number.AL=07h

30. Move contents of AH with 00h. (AH← 00h so AX=0007h)

31. Rotate AX contents in left directions by 12 bits. (AX=7000h)

32. Move the contents of AX to CX(CX←AX so CX=7000h)

33. Read a second digit in AL register through keyboard AL=35h

34. Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number. AL=05h

35. Move contents of AH with 00h. (AH← 00h so AX=0005h)

36. Rotate AX contents in left directions by 8 bits. (AX=0500h)

37. Add the contents of AX and CX (CX=CX+AX so CX=7500h)

38. Read a third digit in AL register through keyboard AL=31h

39. Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number. AL=01h

40. Move contents of AH with 00h. (AH← 00h so AX=0001h)

41. Rotate AX contents in left directions by 4 bits. (AX=0010h)

42. Add the contents of AX and CX (CX=CX+AX so CX=7510h)

43. Read a fourth digit in AL register through keyboard AL=34h

44. Call Input procedure to make a number from ASCII hexadecimal to a normal hexadecimal number. AL=04h

45. Move contents of AH with 00h. (AH← 00h so AX=0004h)

46. Add the contents of AX and BX (CX=CX+AX so CX=7514h)

47. Compare accepted choice from AL with 02h

48. If zero flag is set then goto step no 69 otherwise goto step no 49

49. Add the contents of BX and CX(BX=BX+CX so BX=9A24h)

50. Preserve the result in temporary variable as t of 16 bit (so t=9A24h)

51. Mask the first nibble by AND operation with number F000h (AND BX,f000h so BX=9000h)

52. Rotate the BX contents right by 12(in decimal so BX=0009h)

53. Call Output procedure with BL register to make a digit back in ASCII hexadecimal range (BX=0039h)

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

55. Move result from temporary variable t to BX again (Now BX=9A24h)

56. Mask the second nibble by AND operation with number 0F00h (AND BX,0F00h so BX=0A00h)

57. Rotate the contents of BX to right by 8(in decimal)

58. Call Output procedure with BL register to make a digit back in ASCII hexadecimal range (BX=0041h(ASCII hex value of 'A'))

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

60. Move result from temporary variable to BX again (Now AX=9A24h)

61. Mask the third nibble by AND operation with number 00F0h (AND BX,00F0h so BX=0020h)

62. Rotate the contents of BX to right by 4(in decimal)

63. Call Output procedure with BL register to make a digit back in ASCII hexadecimal range (BX=00032h)

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

65. Move result back from temporary variable to BX again (Now BX=9A24h)

66. Mask the fourth nibble by AND operation with number 000Fh (AND BX,000fh so BX=0004h)

67. Call Output procedure with BL register to make a digit back in ASCII hexadecimal range (BX=0004h)

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

69. Subtract the contents of CX from BX(BX←BX-CX so BX=AFFCh)

70. Preserve the result in temporary variable as t of 16 bit (so t=AFFCh)

71. Mask the first nibble by AND operation with number F000h (AND BX,F000h so BX=A000h)

72. Rotate the BX contents right by 12(in decimal so BX=000Ah)

73. Call Output procedure with BL register to make a digit back in ASCII hexadecimal range (BX=0041h (i.e. ASCII hex value for 'A'))

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

75. Move result from temporary variable t to BX again (Now BX=AFFCh)

76. Mask the second nibble by AND operation with number 0f00h (AND BX,0F00h so BX=0F00h)

77. Rotate the contents of BX to right by 8(in decimal)

78. Call Output procedure with BL register to make a digit back in ASCII hexadecimal range (BX=0046h(i.e ASCII hex value of 'F'))

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

80. Move result from temporary variable to BX again (Now BX=AFFCh)

81. Mask the third nibble by AND operation with number 00F0h (AND BX,00F0h so BX=00F0h)

82. Rotate the contents of BX to right by 4(in decimal)

83. Call Output procedure with BL register to make a digit back in ASCII hexadecimal range (BX=0046h (i.e ASCII hex value of 'F'))

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

85. Move result back from temporary variable to BX again (Now BX=AFFCh)

86. Mask the fourth nibble by AND operation with number 000fh (AND AX,000Fh so AX=000Ch)

87. Call Output procedure with BL register to make a digit back in ASCII hexadecimal range (BX=0043h(i.e ASCII hex value of 'C))

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

89. Stop

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

1. Compare the contents of AL with 41h(Small case)
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 AL register
4. Add 30h to AL 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’

Similarly, the procedures for 8-bit subtraction, multiplication and division can be extended to 16-bit operations