16 Bit Arithmetic Operations using Trainer Kit

Ex No. : 2 Date: 09/02/2021

a. 16 bit Hexadecimal Addition

Aim:

To write an assembly language program to implement 16 bit hexadecimal addition.

Algorithm:

(a) Load input 1 into AX register
(b) Load input 2 into BX register
(c) Add the contents of AX & BX
(d) Move the result from AX register into memory
(e) Stop

Program:

```
MOV   AX, 1234H
MOV   BX, 1234H
ADD   AX,BX
MOV   [8500], AX
HLT
```

Table:

Memory	Label	Mnemonics	Operand	Opcode	Comments
8000		MOV	AX,1234H	B8,34,12	Move 1234H into AX register
8003		MOV	BX,1234H	BB,34,12	Move 1234H into BX register
8006		ADD	AX,BX	03,C3	Add the contents of AX&BX
8008		MOV	[8500],AX	A3,00,85	Move the sum from AX to memory
800B		HLT		F4	Stop

Sample input & output:

Input:

Input 1--- 1234H

Input 2--- 1234H

Output:

[8500] --- 68

[8501] --- 24

Result:

Thus 16 bit hexadecimal addition has been performed successfully and the output is displayed.

b. 16 bit Hexadecimal Subtraction

Aim:

To write an assembly language program to implement 16 bit hexadecimal subtraction.

Algorithm:

(a) Load input 1 into AX register
(b) Load input 2 into BX register
(c) Subtract the contents of AX & BX
(d) Move the result from AX register into memory
(e) Stop

Program:

```
MOV   AX, 1234H
MOV   BX, 1234H
SUB    AX,BX
MOV   [8500], AX
HLT
```

Table:

Memory	Label	Mnemonics	Operand	Opcode	Comments
8000		MOV	AX,1234H	B8,34,12	Move 1234H into AX register
8003		MOV	BX,1234H	BB,34,12	Move 1234H into BX register
8006		SUB	AX,BX	2B,C3	Subtract the contents of AX&BX
8008		MOV	[8500],AX	A3,00,85	Move the contents of AX to memory
800B		HLT		F4	Stop

Sample input & output:

Input:

Input 1--- 1234H

Input 2--- 1234H

Output:

[8500] --- 00

[8501] --- 00

Result:

Thus 16 bit hexadecimal subtraction has been performed successfully and the output is displayed.

c. 16 bit Hexadecimal Multiplication

Aim:

To write an assembly language program to implement 16 bit hexadecimal multiplication.

Algorithm:

(a) Load input 1 into AX register
(b) Load input 2 into BX register
(c) Load 0000 into DX register
(d) Multiply the contents of AX & BX
(e) Store the value in AX register (lower 16 bit) into memory
(f) Store the value in DX register (upper 16 bit) into memory
(g) Stop

Program:

```
MOV   AX,1234H
MOV   BX, 1234H
MOV   DX, 0000H
MUL   BX
MOV   [8500],AX
MOV   [8502],DX
HLT
```

Table:

Memory	Label	Mnemonics	Operand	Opcode	Comments
8000		MOV	AX,1234H	B8,34,12	Move 1234H into AX register
8003		MOV	BX,1234H	BB,34,12	Move 1234H into BX register
8006		MOV	DX,0000H	BA,00,00	Move 0000H into DX register
8009		MUL	BX	F7,E3	Multiply the contents of AX&BX
800B		MOV	[8500],AX	A3,00,85	Move the value in AX to memory
800E		MOV	[8502],DX	89,16,02,85	Move the value in DX to memory
8012		HLT		F4	Stop

Sample input & output:

Input:

Input 1--- 1234H

Input 2--- 1234H

Output:

[8500] --- 90

[8501] --- 5A

[8502] --- 4B

[8503] --- 01

Result:

Thus 16 bit hexadecimal multiplication has been performed successfully and the output is displayed.

d. 16 bit Hexadecimal Division

Aim:

To write an assembly language program to implement 16 bit hexadecimal division.

Algorithm:

(a) Load input 1 into AX register
(b) Load input 2 into BX register
(c) Load 0000 into DX register
(d) Divide the contents of DX(upper 16 bit) & AX (lower 16 bit) by BX
(e) Store the value in AX register (quotient) into memory
(f) Store the value in DX register (remainder) into memory
(g) Stop

Program:

```
MOV   AX,1234H
MOV   BX, 1234H
MOV   DX, 0000H
DIV     BX
MOV   [8500],AX   //quotient
MOV   [8502],DX   //remainder
HLT
```

Table:

Memory	Label	Mnemonics	Operand	Opcode	Comments
8000		MOV	AX,1234H	B8,34,12	Move 1234H into AX register
8003		MOV	BX,1234H	BB,34,12	Move 1234H into BX register
8006		MOV	DX,0000H	BA,00,00	Move 0000H into DX register
8009		DIV	BX	F7,F3	Divide DX&AX by BX
800B		MOV	[8500],AX	A3,00,85	Move the value in AX to memory
800E		MOV	[8502],DX	89,16,02,85	Move the value in DX to memory
8012		HLT		F4	Stop

Sample input & output:

Input:

Numerator --- 00001234H

Denominator --- 1234H

Output:

[8500] --- 01

[8501] --- 00

[8502] --- 00

[8503] --- 00

//[8500]&[8501] — Quotient

//[8502]&[8503] — Remainder

Result:

Thus 16 bit hexadecimal division has been performed successfully and the output is displayed.