VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

MICROPROCESSOR AND INTERFACING

Date: 12/02/2021

Submitted By,

Name - Suraj P

Registration Number - 19BCE1044

Aim:

To verify the arithmetic operations for 8 bit and 16-bit numbers using 8086 processor by MASM611 assembler.

Tool Used:

Assembler - MASM 611

Algorithm:

8 BIT

The date and Month of DOB are taken as operands

1. Start

2.The Larger value is moved onto ah register and smaller in bh.

3. For addition and subtraction ,we use the command `add ah,bh` (ah = ah+bh) and `sub ah,bh` (ah = ah-bh) and the result is stored in ah register. The sum and diff stored in temporary variable in memory.

4. For multiplication instead of ah register , the larger data stored in al register. To generate product, we use `mul bh.` The result is stored in ax register(16 bits) and mov prod into a temporary variable.

5. For division the dividend (larger) stored in ax register and we use the command `div bh` .The quotient is stored in al and remainder in ah register. Allocate variable to store this quotient and rem.

6. In the data segment define the size allocated for each variable

7. Halt

16 BIT

The date + Month and year of DOB are taken as operands

1. Start

2.The Larger value is moved onto ax register and smaller in bx.

3. For addition and subtraction ,we use the command `add ax,bx` (ax = ax+bh) and `sub ax,bx` (ax = ax-bx) and the result is stored in ax register. The sum and diff are then stored in temporary variable.

4. For multiplication we use `mul bx`. The result is stored in ax and dx register as the value can cross 16-bit register capacity of ax and thus overflow stored in dx register. The product is stored in 2 variables each from ax and dx.

5. Before running the division we first clear the dx register using `mov dx,0h` .

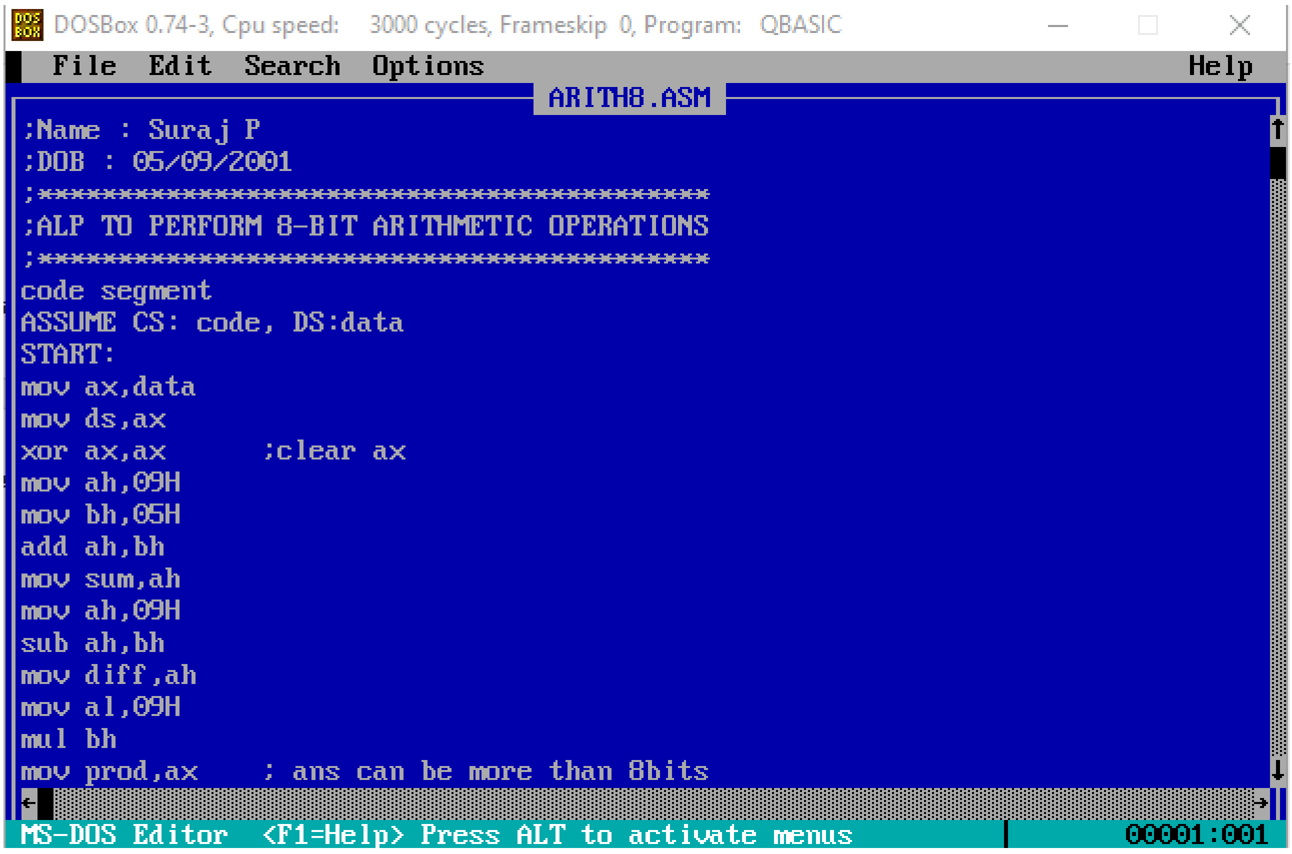
6. For division the dividend (larger) stored in ax register and divisor is chosen to be an 8-bit number stored in bx register. We use the command `div bh` .The quotient is stored in ax and remainder in dx register.

7.In data segment define the size allocated for each variable

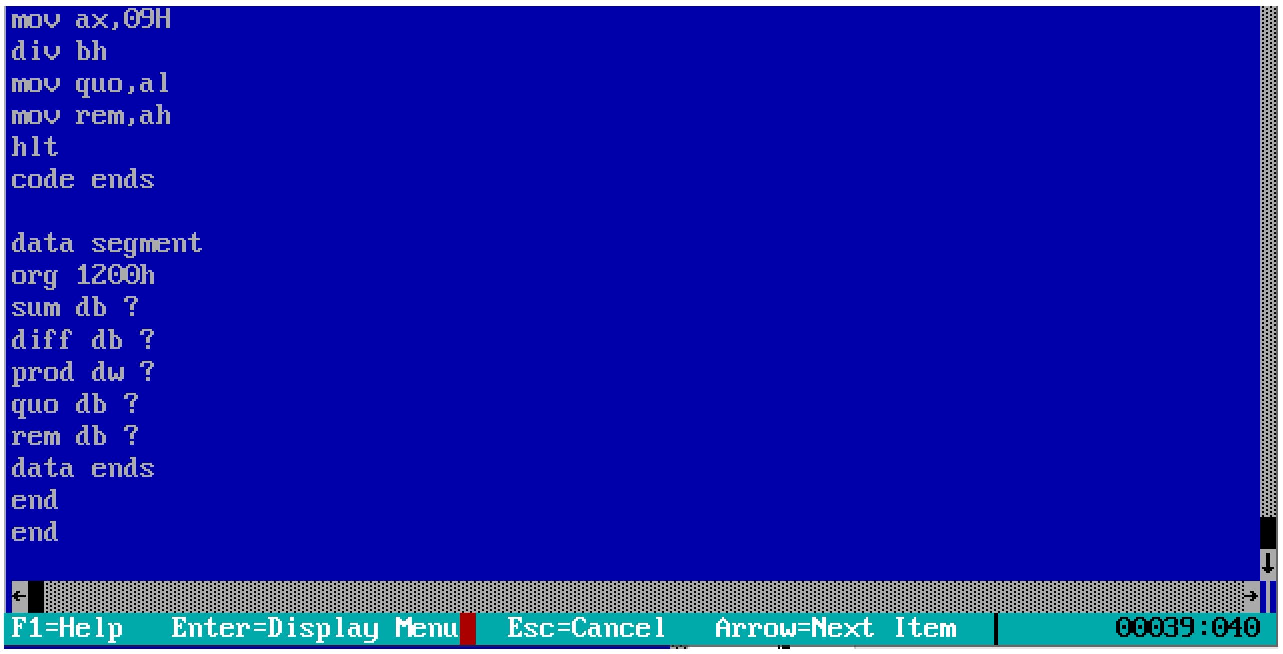
8. Halt

Program:

8 bit

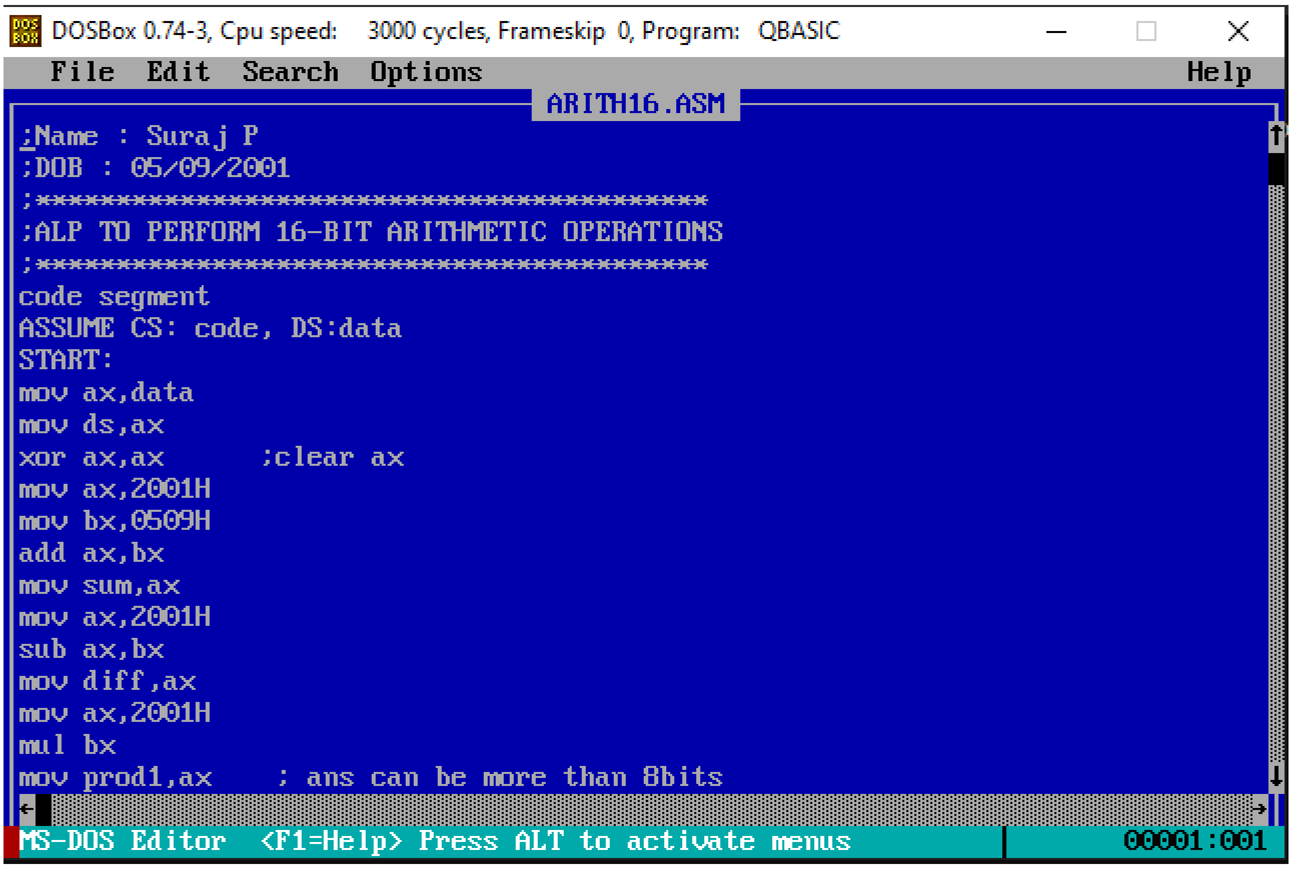

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: QBASIC
File Edit Search Options Help
ARITH8.ASM
MS-DOS Editor <F1=Help> Press ALT to activate menus 00001:001

```
;Name : Suraj P
;DOB : 05/09/2001
;****************************************
;ALP TO PERFORM 8-BIT ARITHMETIC OPERATIONS
;****************************************
code segment
ASSUME CS: code, DS:data
START:
mov ax,data
mov ds,ax
xor ax,ax      ;clear ax
mov ah,09H
mov bh,05H
add ah,bh
mov sum,ah
mov ah,09H
sub ah,bh
mov diff,ah
mov al,09H
mul bh
mov prod,ax    ; ans can be more than 8bits
```


F1=Help Enter=Display Menu Esc=Cancel Arrow=Next Item 00039:040

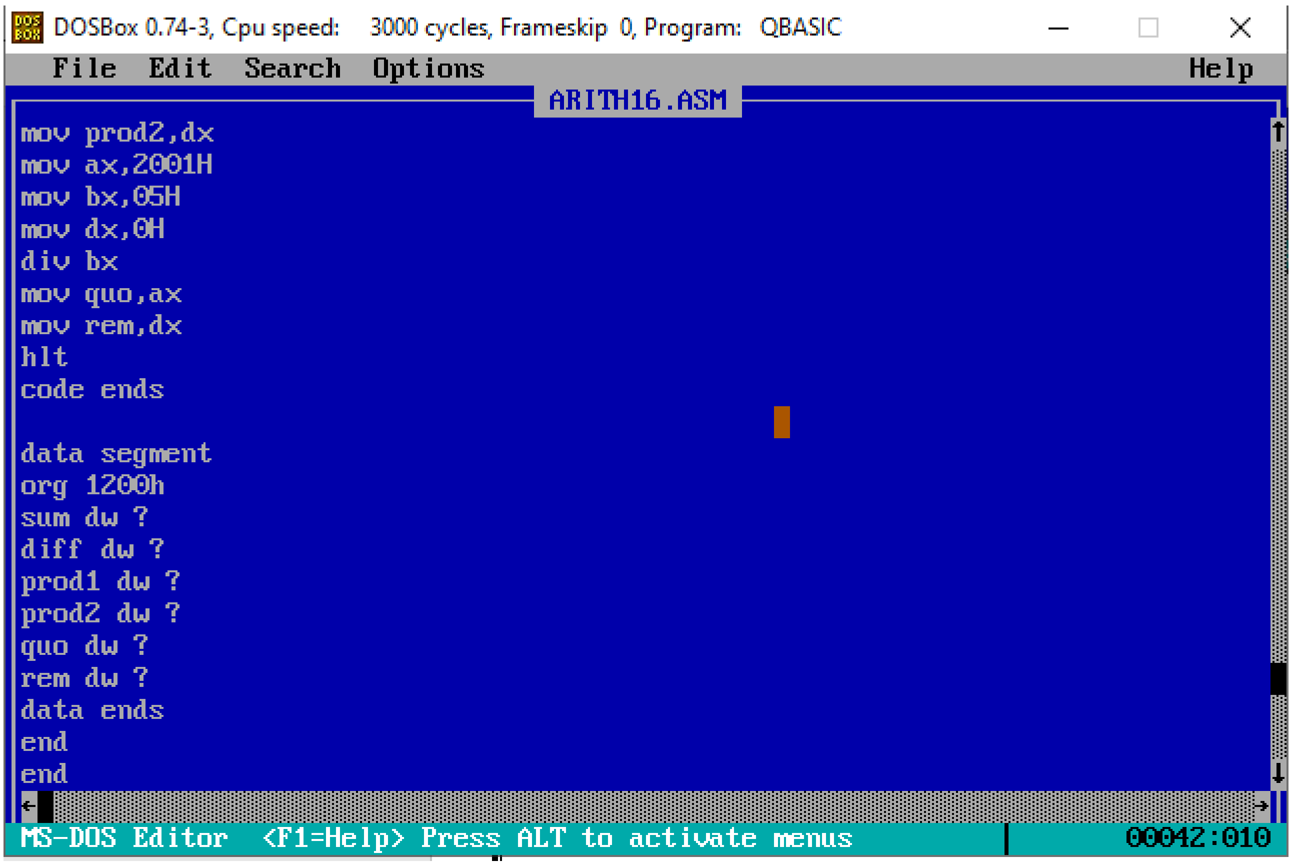
```
mov ax,09H
div bh
mov quo,al
mov rem,ah
hlt
code ends

data segment
org 1200h
sum db ?
diff db ?
prod dw ?
quo db ?
rem db ?
data ends
end
end
```

16 bit


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: QBASIC
File Edit Search Options Help
ARITH16.ASM

```
;Name : Suraj P
;DOB : 05/09/2001
;****************************************
;ALP TO PERFORM 16-BIT ARITHMETIC OPERATIONS
;****************************************
code segment
ASSUME CS: code, DS:data
START:
mov ax,data
mov ds,ax
xor ax,ax      ;clear ax
mov ax,2001H
mov bx,0509H
add ax,bx
mov sum,ax
mov ax,2001H
sub ax,bx
mov diff,ax
mov ax,2001H
mul bx
mov prod1,ax     ; ans can be more than 8bits
```

```
MS-DOS Editor  <F1=Help> Press ALT to activate menus        00001:001
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: QBASIC
File Edit Search Options Help
ARITH16.ASM

```
mov prod2,dx
mov ax,2001H
mov bx,05H
mov dx,0H
div bx
mov quo,ax
mov rem,dx
hlt
code ends

data segment
org 1200h
sum dw ?
diff dw ?
prod1 dw ?
prod2 dw ?
quo dw ?
rem dw ?
data ends
end
end
```

```
MS-DOS Editor  <F1=Help> Press ALT to activate menus        00042:010
```

Sample Input:

8 bit

Num1 : 09H

Num2 : 05H

16 bit

For Addition, Subtraction, Multiplication

Num1 : 2001H

Num2 : 0509H

For Division

Num1 : 2001H

Num2 : 05H

Sample Output:

Addition (8 bit): 0EH

Addition (16 bit): 250AH

Subtraction (8 bit): 04H

Subtraction (16 bit): 1AF8H

Multiplication (8 bit): 002DH

Multiplication (16 bit): A12509 H

Division (8 bit):

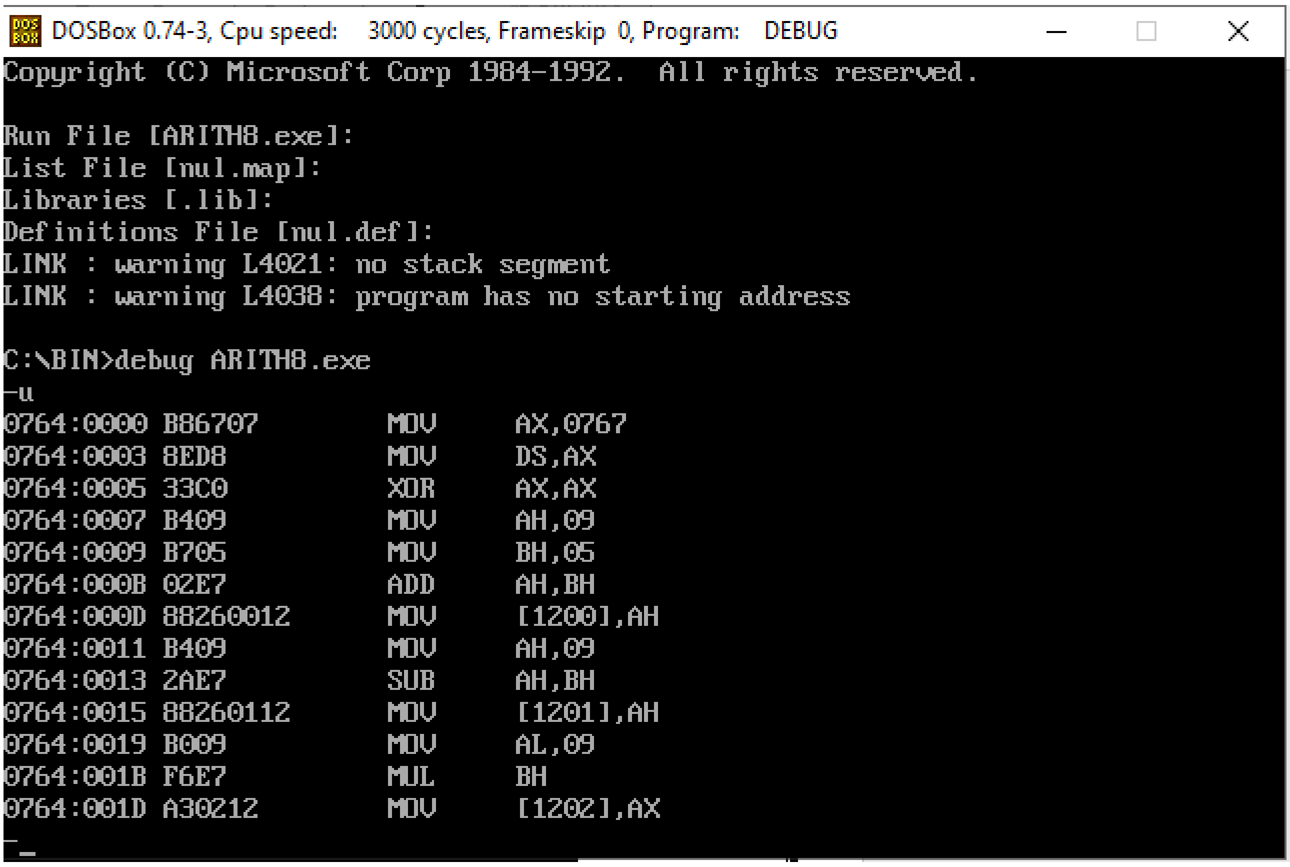
Quotient : 01H **Remainder:** 04H

Division (16 bit):

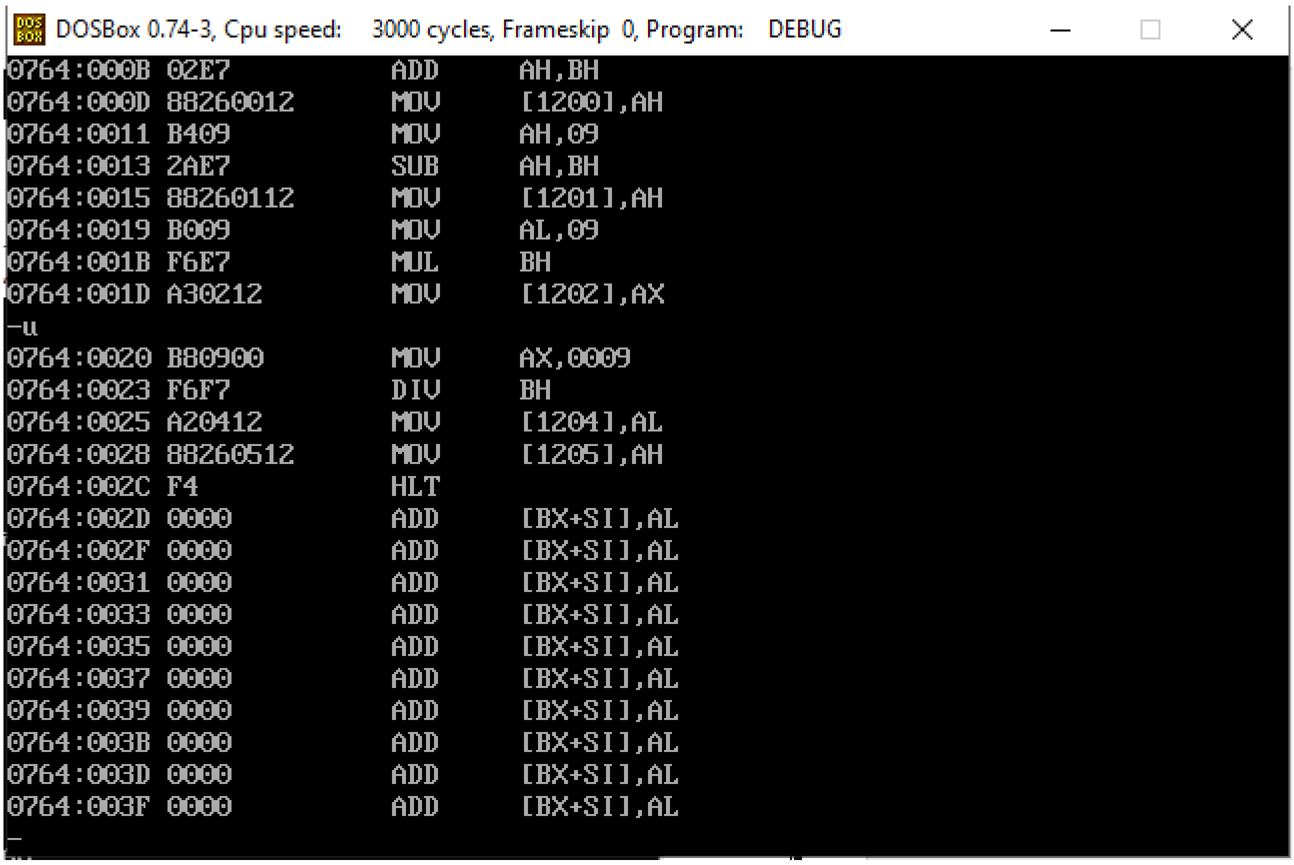
Quotient : 0666H **Remainder:** 0003H

Snapshot of the Output:

8 bit

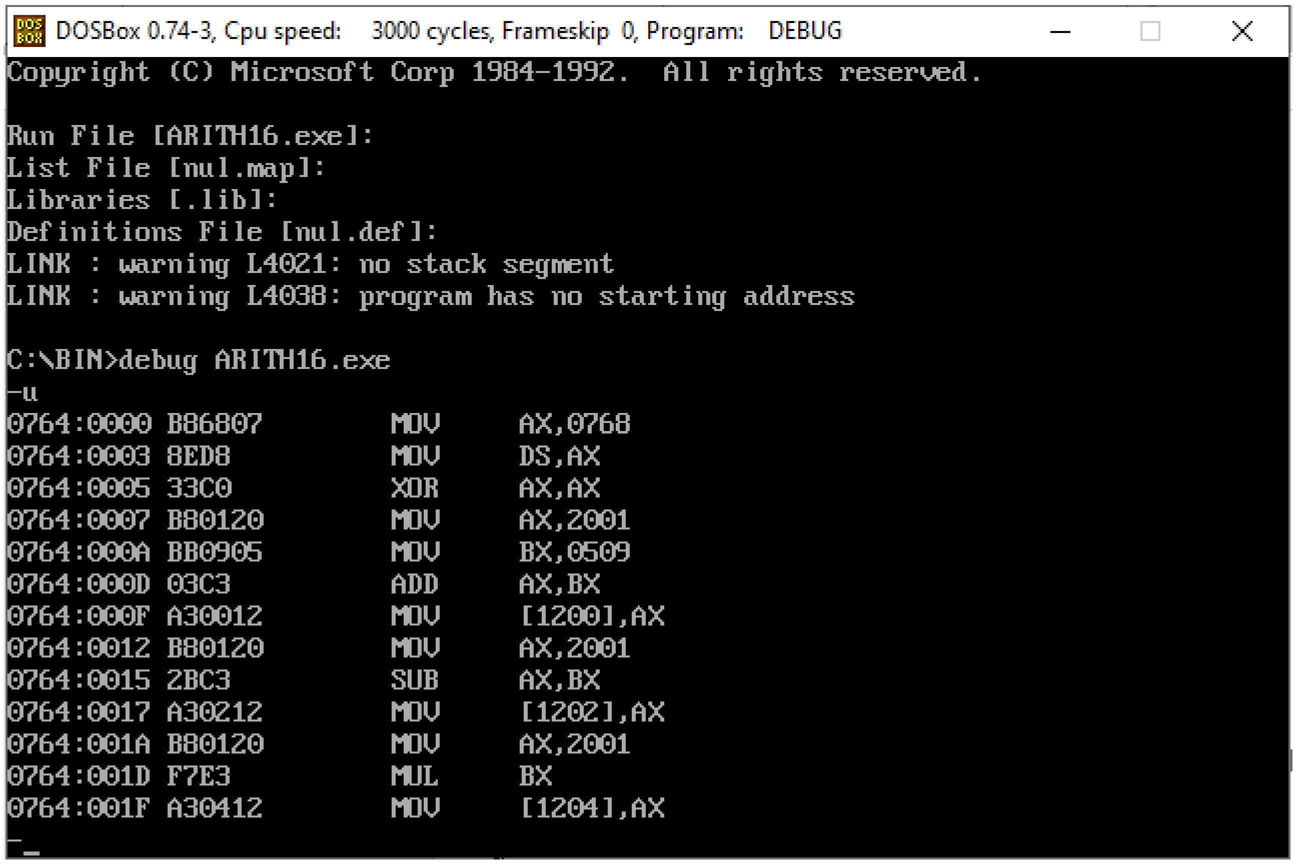

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
Copyright (C) Microsoft Corp 1984-1992.  All rights reserved.

Run File [ARITH8.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:
LINK : warning L4021: no stack segment
LINK : warning L4038: program has no starting address

C:\BIN>debug ARITH8.exe
-u
0764:0000 B86707        MOV     AX,0767
0764:0003 8ED8          MOV     DS,AX
0764:0005 33C0          XOR     AX,AX
0764:0007 B409          MOV     AH,09
0764:0009 B705          MOV     BH,05
0764:000B 02E7          ADD     AH,BH
0764:000D 88260012      MOV     [1200],AH
0764:0011 B409          MOV     AH,09
0764:0013 2AE7          SUB     AH,BH
0764:0015 88260112      MOV     [1201],AH
0764:0019 B009          MOV     AL,09
0764:001B F6E7          MUL     BH
0764:001D A30212        MOV     [1202],AX
-_
```

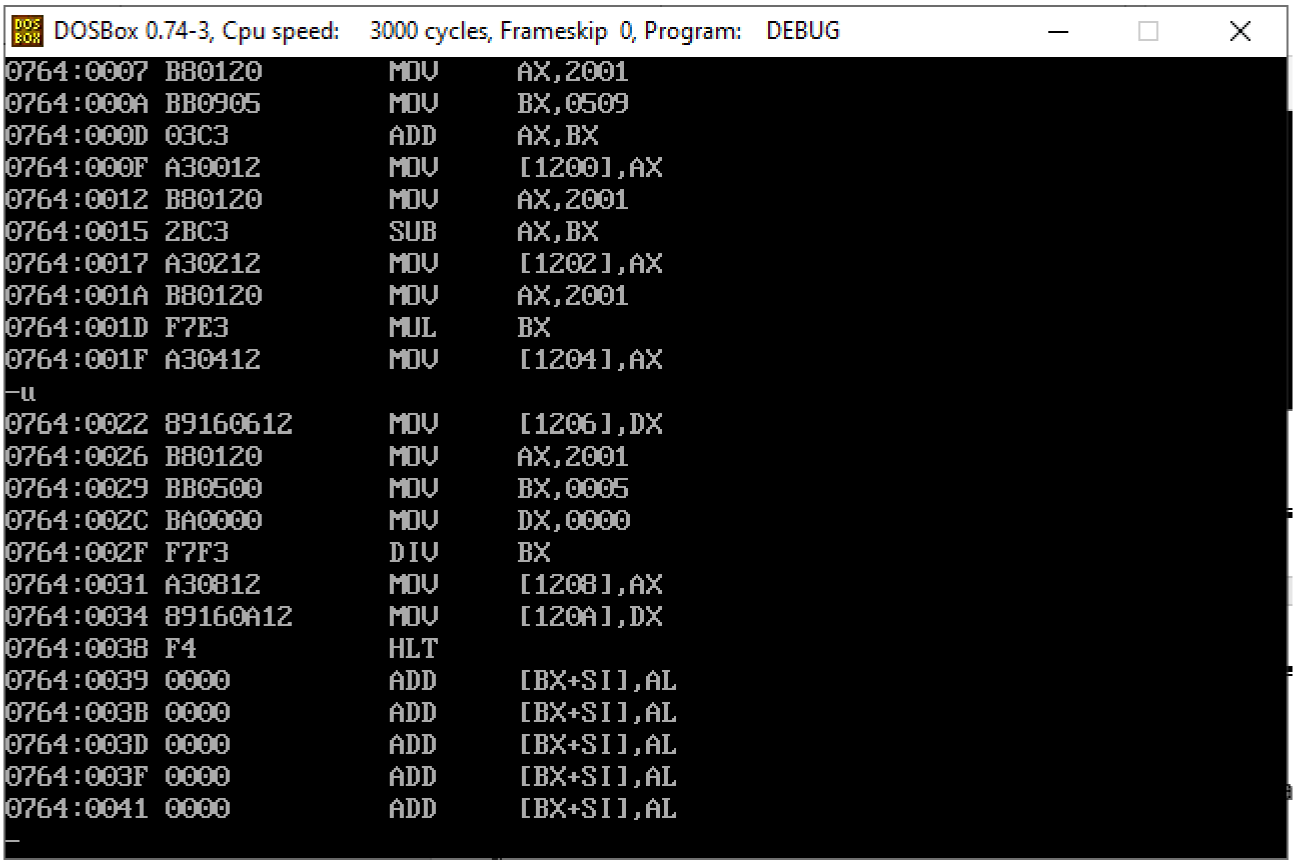

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
0764:000B 02E7          ADD     AH,BH
0764:000D 88260012      MOV     [1200],AH
0764:0011 B409          MOV     AH,09
0764:0013 2AE7          SUB     AH,BH
0764:0015 88260112      MOV     [1201],AH
0764:0019 B009          MOV     AL,09
0764:001B F6E7          MUL     BH
0764:001D A30212        MOV     [1202],AX
-u
0764:0020 B80900        MOV     AX,0009
0764:0023 F6F7          DIV     BH
0764:0025 A20412        MOV     [1204],AL
0764:0028 88260512      MOV     [1205],AH
0764:002C F4            HLT
0764:002D 0000          ADD     [BX+SI],AL
0764:002F 0000          ADD     [BX+SI],AL
0764:0031 0000          ADD     [BX+SI],AL
0764:0033 0000          ADD     [BX+SI],AL
0764:0035 0000          ADD     [BX+SI],AL
0764:0037 0000          ADD     [BX+SI],AL
0764:0039 0000          ADD     [BX+SI],AL
0764:003B 0000          ADD     [BX+SI],AL
0764:003D 0000          ADD     [BX+SI],AL
0764:003F 0000          ADD     [BX+SI],AL
-
```

16 bit


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

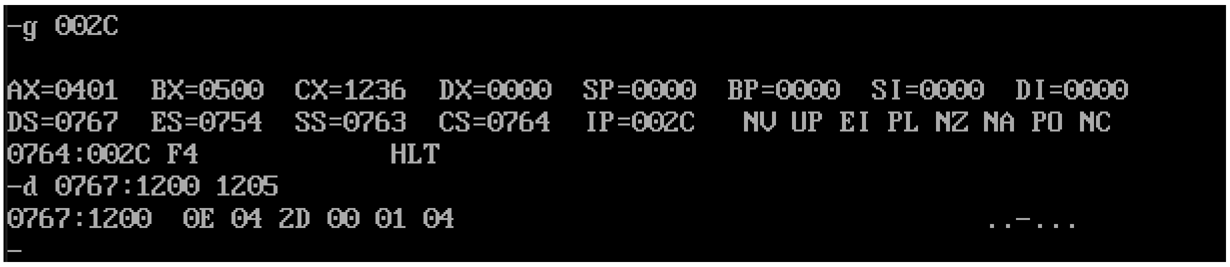
```
Copyright (C) Microsoft Corp 1984-1992.  All rights reserved.

Run File [ARITH16.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:
LINK : warning L4021: no stack segment
LINK : warning L4038: program has no starting address

C:\BIN>debug ARITH16.exe
-u
0764:0000 B86807        MOV     AX,0768
0764:0003 8ED8          MOV     DS,AX
0764:0005 33C0          XOR     AX,AX
0764:0007 B80120        MOV     AX,2001
0764:000A BB0905        MOV     BX,0509
0764:000D 03C3          ADD     AX,BX
0764:000F A30012        MOV     [1200],AX
0764:0012 B80120        MOV     AX,2001
0764:0015 2BC3          SUB     AX,BX
0764:0017 A30212        MOV     [1202],AX
0764:001A B80120        MOV     AX,2001
0764:001D F7E3          MUL     BX
0764:001F A30412        MOV     [1204],AX
-_
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
0764:0007 B80120        MOV     AX,2001
0764:000A BB0905        MOV     BX,0509
0764:000D 03C3          ADD     AX,BX
0764:000F A30012        MOV     [1200],AX
0764:0012 B80120        MOV     AX,2001
0764:0015 2BC3          SUB     AX,BX
0764:0017 A30212        MOV     [1202],AX
0764:001A B80120        MOV     AX,2001
0764:001D F7E3          MUL     BX
0764:001F A30412        MOV     [1204],AX
-u
0764:0022 89160612      MOV     [1206],DX
0764:0026 B80120        MOV     AX,2001
0764:0029 BB0500        MOV     BX,0005
0764:002C BA0000        MOV     DX,0000
0764:002F F7F3          DIV     BX
0764:0031 A30812        MOV     [1208],AX
0764:0034 89160A12      MOV     [120A],DX
0764:0038 F4            HLT
0764:0039 0000          ADD     [BX+SI],AL
0764:003B 0000          ADD     [BX+SI],AL
0764:003D 0000          ADD     [BX+SI],AL
0764:003F 0000          ADD     [BX+SI],AL
0764:0041 0000          ADD     [BX+SI],AL
-
```

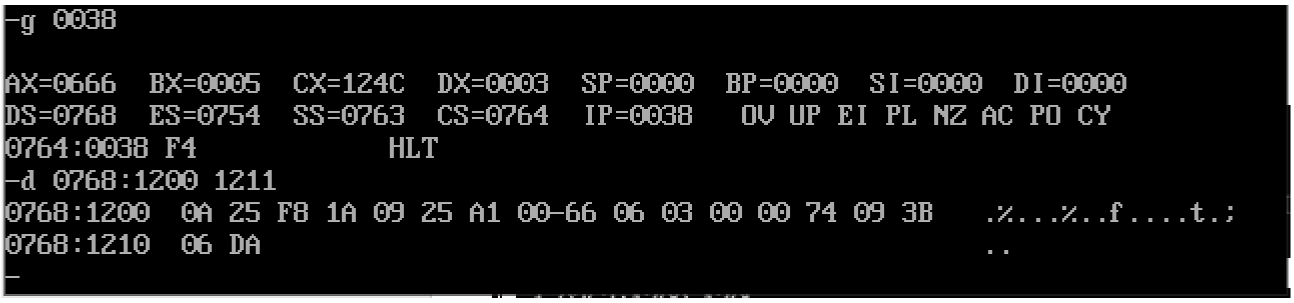
Register/Memory Contents for I/O

8 bit



```
-g 002C

AX=0401  BX=0500  CX=1236  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=0767  ES=0754  SS=0763  CS=0764  IP=002C   NV UP EI PL NZ NA PO NC
0764:002C F4            HLT
-d 0767:1200 1205
0767:1200  0E 04 2D 00 01 04                                ..-...
-
```

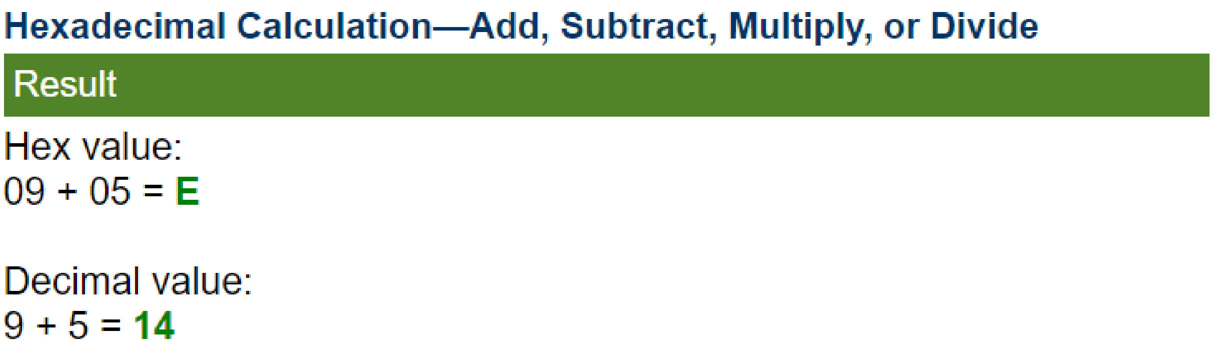
16 bit



```
-g 0038

AX=0666  BX=0005  CX=124C  DX=0003  SP=0000  BP=0000  SI=0000  DI=0000
DS=0768  ES=0754  SS=0763  CS=0764  IP=0038   OV UP EI PL NZ AC PO CY
0764:0038 F4            HLT
-d 0768:1200 1211
0768:1200  0A 25 F8 1A 09 25 A1 00-66 06 03 00 00 74 09 3B   .%...%..f....t.;
0768:1210  06 DA                                            ..
-
```

Manual Verification:

8 bit

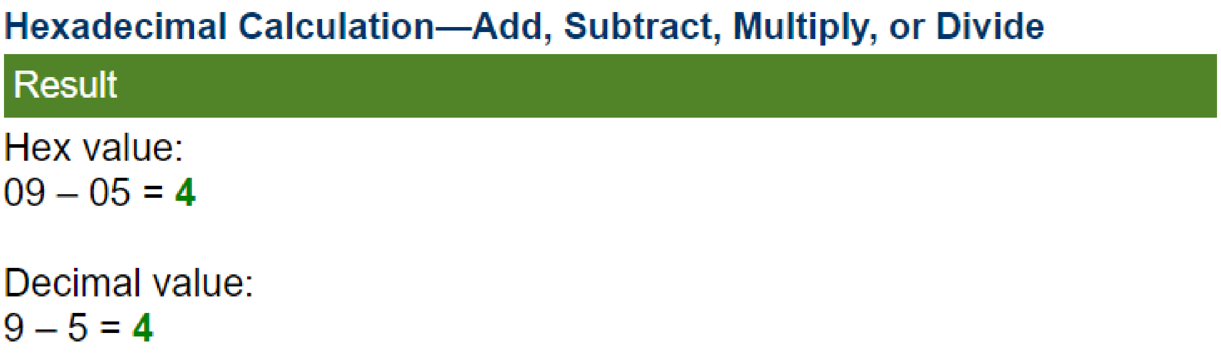


Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
09 + 05 = **E**

Decimal value:
9 + 5 = **14**

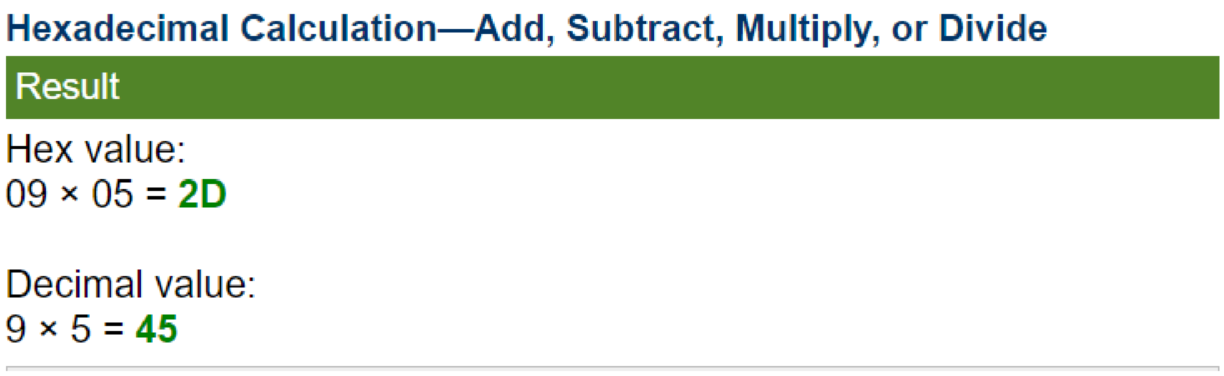


Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
09 – 05 = **4**

Decimal value:
9 – 5 = **4**

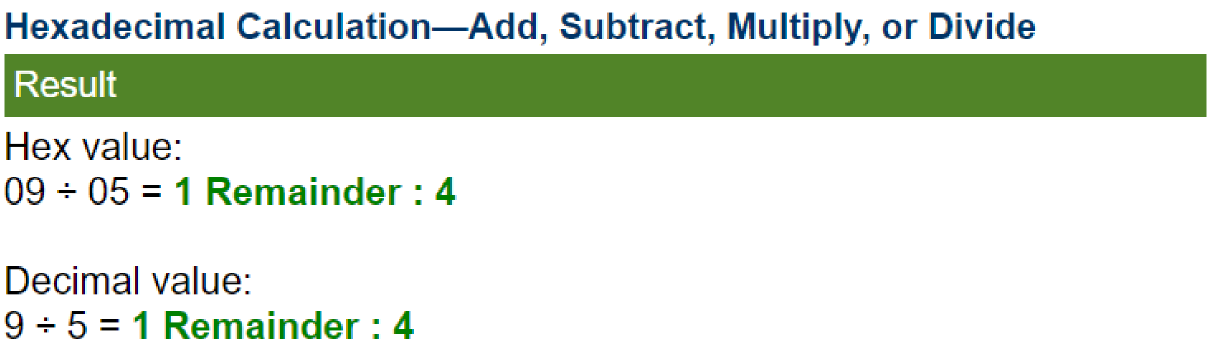


Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
09 × 05 = **2D**

Decimal value:
9 × 5 = **45**



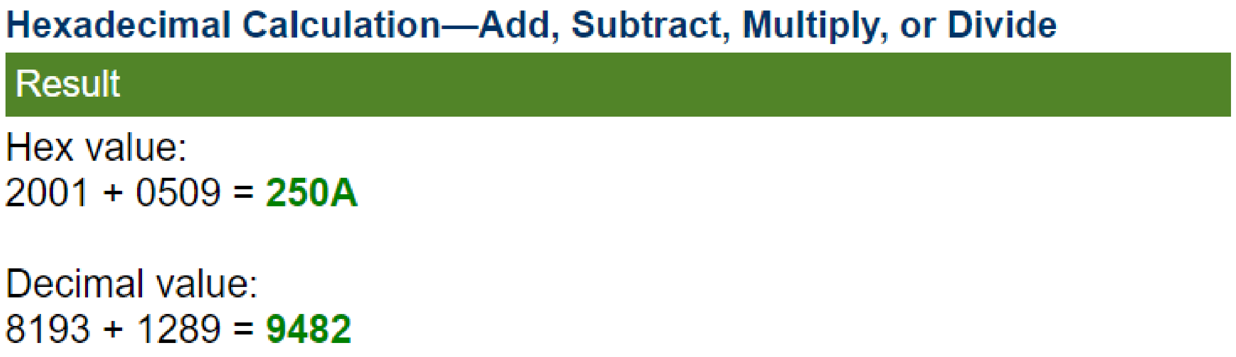
Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
09 ÷ 05 = **1 Remainder : 4**

Decimal value:
9 ÷ 5 = **1 Remainder : 4**

16 bit

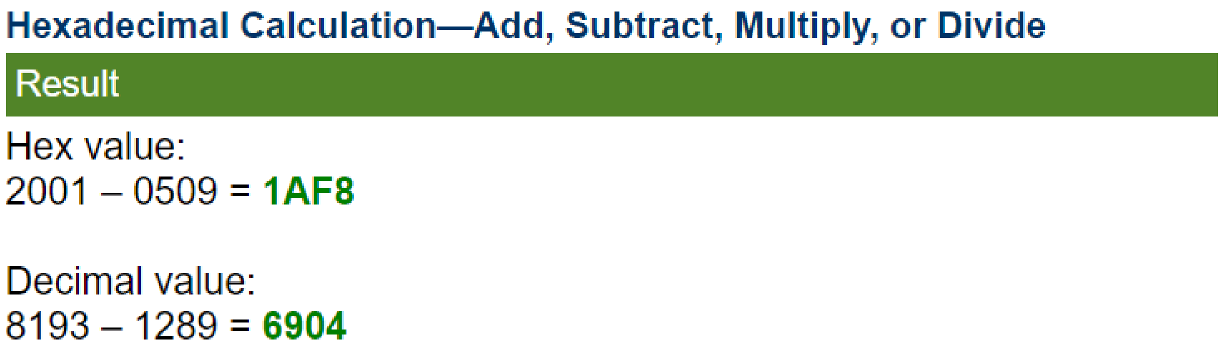


Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
2001 + 0509 = **250A**

Decimal value:
8193 + 1289 = **9482**

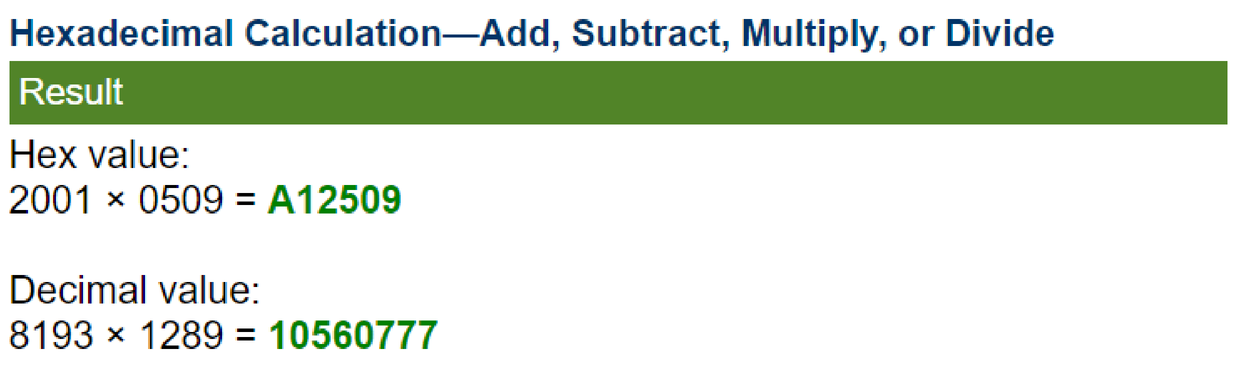


Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
2001 – 0509 = **1AF8**

Decimal value:
8193 – 1289 = **6904**

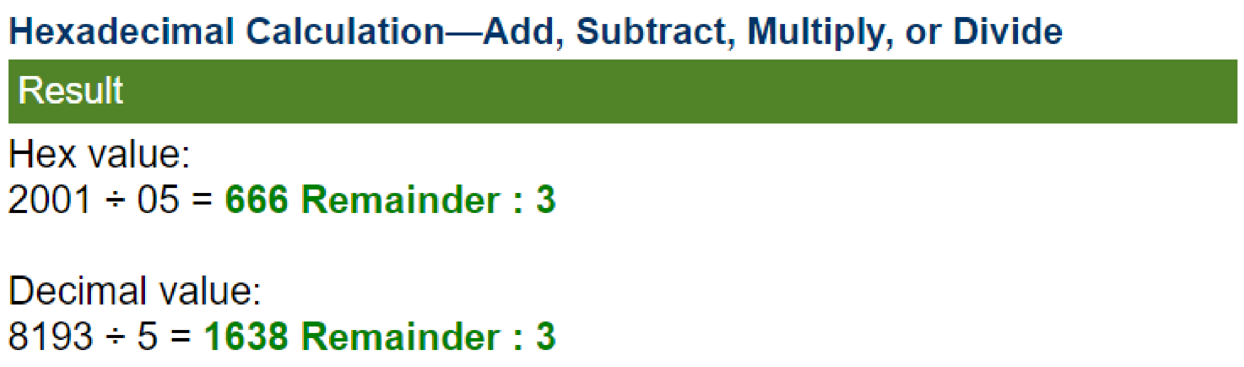


Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
2001 × 0509 = **A12509**

Decimal value:
8193 × 1289 = **10560777**



Hexadecimal Calculation—Add, Subtract, Multiply, or Divide

Result

Hex value:
2001 ÷ 05 = **666 Remainder : 3**

Decimal value:
8193 ÷ 5 = **1638 Remainder : 3**

Result:

Hence all operations-addition, subtraction, multiplication and division in both 8-bit as well as 16-bit representation have been performed and verified using the MASM611 application in DOSBOX.