

Expt. No. 7



30/9/24

MASM - 32 BIT ADDITION

Aim :-

To write a 8086 program (in memory) to add 2 bit numbers

Algorithm :-

Step 1 : Start

Step 2 : Initialize data in data segment

Step 3 : Initialize data in code segment

Step 4 : Clear DX as 00H

Step 5 : Move the first number to AX register

Step 6 : Move second number to BX register

Step 7 : Move AX and DS registers

Step 8 : Add AX and BX

Step 9 : Move result from AX to result

Step 10 : Get the second segment (of 32 bit no)
by increment by 2, move num1+2 to AXStep 11 : Get the second segment (of 32 bit no) by
increment by 2, move num2+2 to AX

Step 12 : Add with carry AX and BX

Step 13 : Move AX to result+2 (remaining 16
bit of 32 bit number)

Step 14 : Add with carry DX and 00H

Step 15 : Move DX to result+4

Step 16 : Program will terminate control of
the operating system (INT 21H)

Step 17 : Code segment ends

Expt. No.



Step 18 : STOP

~~Result :-~~

~~program executed successfully and output
verified
16/10/2X~~

PROGRAM

```
DATA SEGMENT
NUM1 DW 1234H,5678H
NUM2 DW 0AB12H,0CDEFH
RESULT DW 3 DUP(?)
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:
MOV AX,DATA
MOV DS,AX
MOV DX,00H
MOV AX,NUM1
MOV BX,NUM2
CLC
ADD AX,BX
MOV RESULT,AX
MOV AX,NUM1+2
MOV BX,NUM2+2
ADC AX,BX
MOV RESULT+2,AX
ADC DX,00H
MOV RESULT+4,DX
MOV AH,4CH
INT 21H
CODE ENDS
END START
```

```
T
AX=2467  BX=CDEF  CX=003C  DX=0001  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0024  NV UP EI PL NZ NA PO NC
076B:0024 89160C00  MOV [000C],DX          DS:000C=0000
-T

AX=2467  BX=CDEF  CX=003C  DX=0001  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0028  NV UP EI PL NZ NA PO NC
076B:0028 B44C      MOV AH,4C
-T

AX=4C67  BX=CDEF  CX=003C  DX=0001  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=002A  NV UP EI PL NZ NA PO NC
076B:002A CD15      INT 15
-D DS:0000
076A:0000 34 12 78 56 12 AB EF CD-46 BD 67 24 01 00 00 00 4.xU....F.g$...
076A:0010 B8 6A 07 8E D8 BA 00 00-A1 00 00 8B 1E 04 00 F8 .j.....
076A:0020 03 C3 A3 08 00 A1 02 00-8B 1E 06 00 13 C3 A3 0A .....
076A:0030 00 83 D2 00 89 16 0C 00-B4 4C CD 15 08 56 E8 1F .....L...V..
076A:0040 D3 8B F0 0B F0 74 03 E9-B8 00 83 3E CE 49 01 75 .....t....>.I.u
076A:0050 50 A1 2C 4B 33 D2 E8 01-07 8B F0 89 56 FE 0B D0 P.,K3.....U...
076A:0060 75 06 B8 1F 00 E9 9A 00-8B 1D 2A E4 8A 47 04 50 u.....*..G.P
076A:0070 8A 45 10 03 45 02 50 8A-45 11 2A E4 03 45 04 50 .E..E.P.E.*..E.P
```



MASM-32 BIT SUBTRACTION

Aim:-

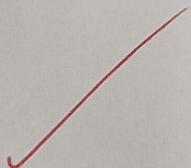
To write a 8086 program (in masm) to subtract two 32-bit numbers

Algorithm:-

- Step 1: Start
- Step 2: Initialize data in data segment
- Step 3: Initialize code in code segment
- Step 4: Clear DX as 00H
- Step 5: Move the first number to AX register
- Step 6: Move second number to BX register
- Step 7: Move AX and DS register
- Step 8: Subtract AX and BX
- Step 9: Move the result from AX to result
- Step 10: Get the second segment (of 32 bit no) by increment by 2, move num1+2 to AX
- Step 11: Get the second segment (of 32 bit no) by increment by 2, move num2+2 to BX
- Step 12: Subtract with borrow AX and BX
- Step 13: Move AX to result+2 (remaining 16 bit of 32 bit number)
- Step 14: Subtract with borrow DX and 00H
- Step 15: Move DX to result+4
- Step 16: Program will terminate control of the operating system (INT 21H)
- Step 17: Code segment ends

PROGRAM:

```
DATA SEGMENT  
NUM2 DW 1234H,5678H  
NUM1 DW 0AB12H,0CDEFH  
RESULT DW 3 DUP(?)  
DATA ENDS  
CODE SEGMENT  
ASSUME CS:CODE,DS:DATA  
START:  
MOV AX,DATA  
MOV DS,AX  
MOV DX,00H  
MOV AX,NUM1  
MOV BX,NUM2  
CLC  
SUB AX,BX  
MOV RESULT,AX  
MOV AX,NUM1+2  
MOV BX,NUM2+2  
SBB AX,BX  
MOV RESULT+2,AX  
SBB DX,00H  
MOV RESULT+4,DX  
MOV AH,4CH  
INT 21H  
CODE ENDS  
END START
```





Expt. No.

Step 18 : STOP

~~RESULT :-~~

✓ Program executed successfully and output
has been verified.

~~Geethan
16/10/~~

OUTPUT:

```

AX=7777 BX=5678 CX=003C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0760 CS=076B IP=0024 NV UP EI PL ZR NA PE NC
076B:0024 89160000 MOV [000C],DX DS:000C=0000

-T

AX=7777 BX=5678 CX=003C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0760 CS=076B IP=0028 NV UP EI PL ZR NA PE NC
076B:0028 B44C MOV AH,4C

-T

AX=4C77 BX=5678 CX=003C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076A ES=075A SS=0760 CS=076B IP=002A NV UP EI PL ZR NA PE NC
076B:002A CD21 INT 21

-D DS:0000
076A:0000 34 12 78 56 12 AB EF CD-DE 98 77 77 00 00-00 00 4 xU . . . . . . . .
076A:0010 B8 6A 07 8E DB BA 00 00-A1 04 00 8B 1E 00 00 F8 . j . . . . . . . .
076A:0020 2B C3 A3 00 00 A1 06 00-BB 1E 02 00 1B C3 A3 0A + . . . . . . . .
076A:0030 00 83 DA 00 B0 16 0C 00-B4 4C CD 21 56 FE 05 0C . . . . . . . . L tU
076A:0040 00 52 50 E8 FA 48 83 C4-04 50 E8 7B 0E 83 C4 04 , RP . H . P . C
076A:0050 3D FF FF 74 0B E9 ED 00-C4 5E FC 26 8A 47 0C 2A = . t . . ^ & G *
076A:0060 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83 , RP . . . . RP . H
076A:0070 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6 , P . . . . P . S

```

Expt. No. 9



2/11

MASM STRING CONCATENATION

Aim:-

To write a MASM program to implement string concatenation

Algorithm:-

Step1 : Start

Step2 : Define data section and declare str1 and str2 inside a null terminates (0) at the end of each string

Step3 : Create a buffer result

Step4 : Begin code section

Step5 : Load str1 into CS register

Step6 : Load address of result buffer into register

Step7 : Set ECX to the length of str1 including the null terminator

Step8 : Use the rep movsh instruction to copy the character of str1 to the result buffer

Step9 : Load the address of str2 into the ESI register

Step10 : Set EC to the length of str2

Step11 : Use rep movsh instruction to copy the characters of str2, into the result buffer

Step12 : STOP

Result:-

✓ Program has been executed and output verified

PROGRAM:

```
DATA SEGMENT
MSG1 DB "HELLO$"
MSG2 DB "WORLD$"
DATA ENDS
ASSUME CS:CODE,DS:DATA
CODE SEGMENT
START:
MOV AX,DATA
MOV DS,AX
LEA DX,MSG1
MOV AH,09H
INT 21H
LEA DX,MSG2
MOV AH,09H
INT 21H
MOV AH,4CH
INT 21H
CODE ENDS
END START
```