# **UCS 1512 - Microprocessor Lab**

# **End Semester Practical Examination - Batch 7**

**Reg no: 185001161 Name : Sreedhar V**

**Date :17/10/2020**

**1.a A block of 10 data is stored in the memory from XX00 to XX09 . Write an ALP using 8086 to transfer the data to the memory location YY00 to YY09 in the reverse order.**

**AIM :**

To program and execute the ALP for transferring a block of 10 data in memory from XX00 to XX09 to the memory location YY00 to YY09 in 8086 using an emulator.

**Algorithm:**

* + Move the address of data segment to register DS .
  + Move the address of extra segment to register ES.
  + Initialize count as 10 to transfer 10 block data.
  + Move offset of source(XX00 to XX09) to source index register(SI).
  + Move offset of destination(YY00 to YY09) to destination index register(DI)
  + Now increment SI register by 9 to point it to the last data of the block.
  + Now start a loop.
  + Set direction flag to 1 navigate in reverse order for source index register.
  + Load a value into AL.
  + Set direction flag to 0 to navigate in forward direction for destination index register.
  + Store the value in the destination.
  + Loop until count becomes zero.
  + Terminate the program.

**ALP:**

assume ds:data,cs:code,es:extra

data segment

source db 01h,10h,20h,30h,40h,50h,60h,70h,80h,90h

count dw 000Ah

data ends

extra segment

dest db ?

extra ends

code segment

org 0100h

start : mov ax,data

mov ds,ax

mov ax,extra

mov es,ax

mov cx,count

mov si,offset source

mov di,offset dest

add si,09h ;increment source offset by 9 to point at end

loop1:

std ; set the direction flag

lodsb ; load a byte from si into al

cld ; clear the direction flag

stosb ; Store byte into di from al

loop loop1

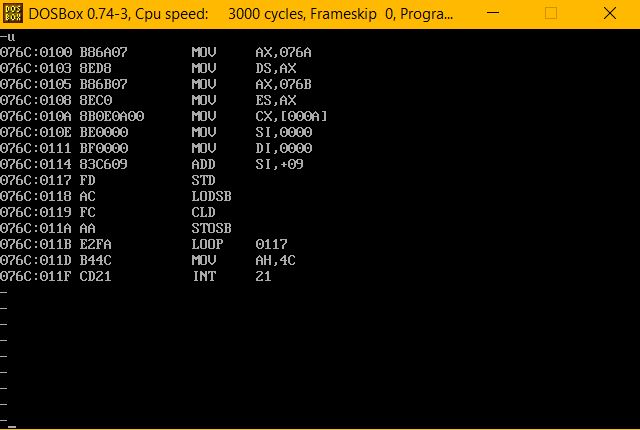
mov ah,4ch

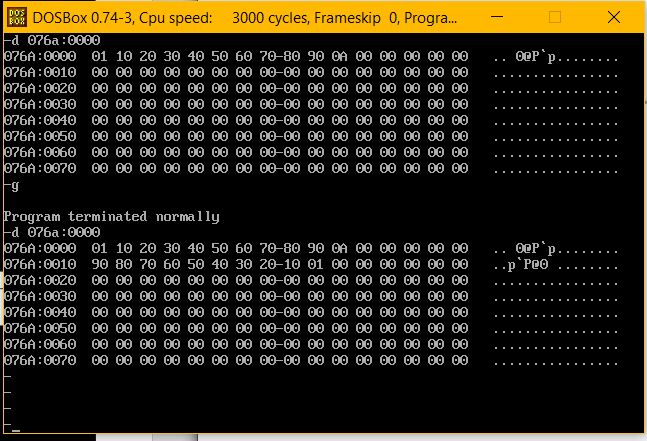
int 21h

code ends

end start

**Output Snapshot:**

****

****

**Result:**

Transferring a block of 10 data in memory from XX00 to XX09 to the memory location YY00 to YY09 is executed and verified using an emulator.

**1.b Write ALPs using 8086 to perform 32 bit addition and subtraction.**

**AIM :**

To program and execute the ALP for 32-bit addition and subtraction in 8086 using an emulator.

1. **Bit Addition:**

**Algorithm:**

* + Move the address of data segment to register DS.
  + Move the lower order nibble(16-bit) of op1 into AX register
  + Move the lower order nibble(16-bit) of op2 into BX register
  + Add AX and BX registers
  + Move the value in AX register into lower nibble(16-bit) of the result.
  + Move the higher order nibble(16-bit) of op1 into AX register
  + Move the higher order nibble(16-bit) of op2 into BX register
  + Add AX and BX registers with carry using ADC instruction.
  + Move the value in AX register into higher nibble(16-bit) of the result.
  + Check for the carry , if there is carry produced store 1 in carry.
  + Terminate the program.

**ALP:**

assume ds:data,cs:code ; 32-bit add

data segment

opr1 dd 12345678h

org 0010h

opr2 dd 55555555h

org 0020h

carry db ?

res dd ?

data ends

code segment

org 0100h

start:

mov ax,data

mov ds,ax

mov ax,word ptr opr1

mov bx,word ptr opr2

add ax,bx

mov word ptr res,ax

mov ax,word ptr opr1 + 2

mov bx,word ptr opr2 + 2

adc ax,bx

mov word ptr res + 2,ax

jnc here

mov bh,01h

mov carry,bh

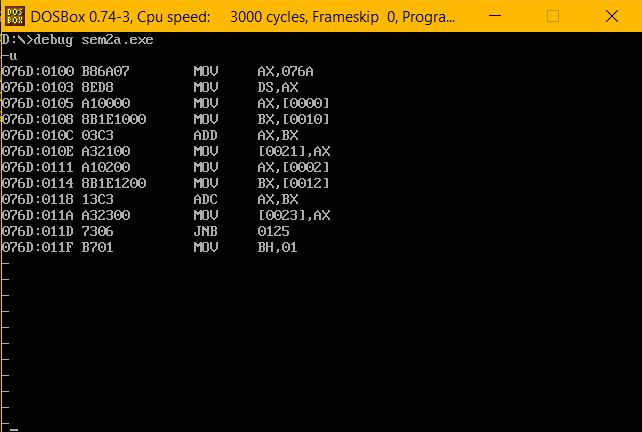
here : mov ah,4ch

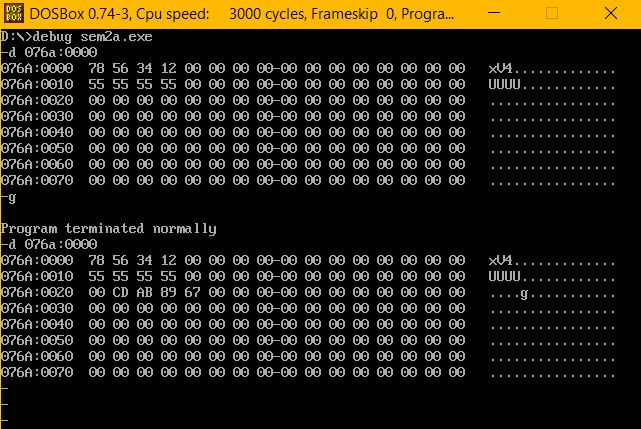
int 21h

code ends

end start

**Output Snapshot:**

****

****

1. **Bit Subtraction:**

**Algorithm:**

* + Move the address of data segment to register DS.
  + Move the lower order nibble(16-bit) of op1 into AX register
  + Move the lower order nibble(16-bit) of op2 into BX register
  + Subtract AX and BX registers using SUB instruction
  + Move the value in AX register into lower nibble(16-bit) of the result.
  + Move the higher order nibble(16-bit) of op1 into AX register
  + Move the higher order nibble(16-bit) of op2 into BX register
  + Subtract AX and BX registers with carry using SBB instruction.
  + Move the value in AX register into higher nibble(16-bit) of the result.
  + Check for the carry , if there is carry produced store 1 in carry.
  + Terminate the program.

**ALP:**

assume ds:data,cs:code ; 32-bit add

data segment

opr1 dd 9ABCDEF0h

org 0010h

opr2 dd 12345678h

org 0020h

carry db ?

res dd ?

data ends

code segment

org 0100h

start:

mov ax,data

mov ds,ax

mov ax,word ptr opr1

mov bx,word ptr opr2

sub ax,bx

mov word ptr res,ax

mov ax,word ptr opr1 + 2

mov bx,word ptr opr2 + 2

sbb ax,bx

mov word ptr res + 2,ax

jnc here

mov bh,01h

mov carry,bh

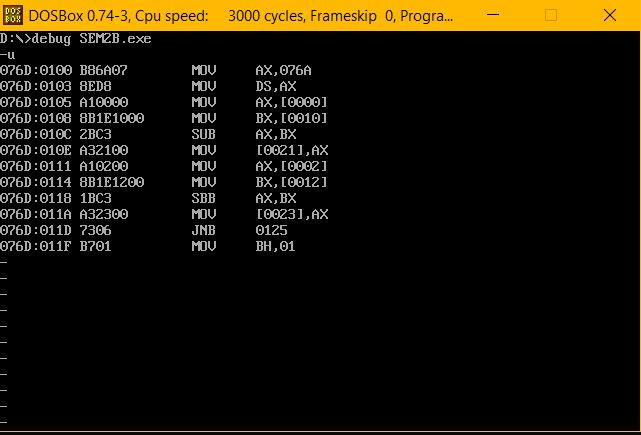
here : mov ah,4ch

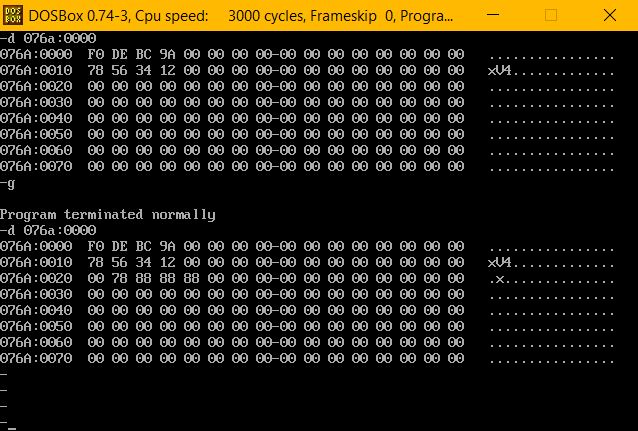
int 21h

code ends

end start

**Output Snapshot:**

****

****

**Result:**

32-Bit addition and subtraction is executed and verified in 8086 using an emulator.