

Somnath Chattaraj

IT-A3-067 ASM Lab Assignment-1

Q1.) Write an Assembly Language Program to add two sixteen-bit numbers. The numbers are stored in DS: 0030H and DS: 0040H. Store the result in DS: 0050H, DS: 0051H, and DS: 0052H.

```
.model small
.stack 100h
.data
.code
main proc
    mov ax, @data
    mov ds, ax

    mov si, 0030h
    mov ax, [si]

    mov si, 0040h
    add ax, [si]
    adc dx, 0

    mov si, 0050h
    mov [si], ax
    inc si
    inc si
    mov [si], dx

    int 03h
main endp
end main
```

```
C:\>debug add16bit.exe
```

```
-t
```

```
AX=076C BX=0000 CX=001F DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076C CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
```

```
-e 076c:0030
```

```
076C:0030 00.05  FF.00
```

```
-e 076c:0040
```

```
076C:0040 04.04  40.00
```

```
-g=0000
```

```
AX=0009 BX=0000 CX=001F DX=0000 SP=0100 BP=0000 SI=0051 DI=0000
DS=076C ES=075A SS=076C CS=076A IP=001A  NU UP EI PL NZ NA PO NC
076A:001A CC          INT     3
```

```
-d ds:0050,0052
```

```
076C:0050 09 00 00
```

```
...
```

Q2.) Write an Assembly Language Program to subtract an 8-bit number stored in DS: 0030H from a number stored in DS: 0040H using 2's complement method. Store the result in DS: 0050H, and DS: 0051H.

```
..model small
.stack 100h
.data
.code
main proc
    mov ax, @data
    mov ds, ax

    mov si, 0030h
    mov al, [si]
    not al
    inc al

    mov si, 0040h
    add al, [si]

    jc L1
    not al
    inc al

L1:
    mov si, 0050h
    mov [si], al
    cmc
    mov ah, 00h
    adc ah, ah
    inc si
    mov [si], ah

    int 03h
main endp
end main
```

```

C:\>debug sub.exe
-t

AX=076C BX=0000 CX=0027 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076D CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0030
076C:0030 09.04

-e 076c:0040
076C:0040 04.09

-g=0000

AX=0005 BX=0000 CX=0027 DX=0000 SP=0100 BP=0000 SI=0051 DI=0000
DS=076C ES=075A SS=076D CS=076A IP=0026  NU UP EI PL NZ NA PO NC
076A:0026 CC          INT     3
-d ds:0050,0051
076C:0050 05 00          ..

```

Q3.) Write a program to transfer a block of 8 data bytes from memory location DS: 0030H to DS: 0040H.

```

.model small
.stack 100h
.data
.code
main proc
    mov ax, @data
    mov ds, ax
    mov es, ax

    mov si, 0030h
    mov di, 0040h
    mov cx, 0008h

    cld
l1:
    movsb
    loop l1

    int 03h
main endp
end main

```

```

C:\>debug loop.exe
-t
AX=076B BX=0000 CX=0015 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076C CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076b:0030
076B:0030  00.01  52.02  50.03  E8.04  EA.05  48.06  83.07  C4.08

-g=0000
AX=076B BX=0000 CX=0000 DX=0000 SP=0100 BP=0000 SI=0038 DI=0048
DS=076B ES=076B SS=076C CS=076A IP=0014  NU UP EI PL NZ NA PO NC
076A:0014 CC          INT     3
-d ds:0040,0047
076B:0040  01 02 03 04 05 06 07 08  .....

```

Q4.) Write an 8086 Assembly Language Program for the addition of 7 eight-bit numbers stored from DS: 0030H. Store the result in DS: 0050H and DS: 0051H.

```

.model small
.stack 100h
.data
.code
main proc
    mov ax, @data
    mov ds, ax

    mov al, 00h
    mov ah, 00h
    mov cl, 07h
    mov si, 0030h

l1:
    add al, [si]
    adc ah, 00h
    inc si
    loop l1

    mov si, 0050h
    mov [si], al
    inc si
    mov [si], ah

    int 03h
main endp
end main
END MAIN

```

```

C:\>debug addloop.exe
-t

AX=076C BX=0000 CX=001F DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076C CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0030
076C:0030 02.01  00.02  03.03  04.04  05.05  06.06  07.07

-g=0000

AX=001C BX=0000 CX=0000 DX=0000 SP=0100 BP=0000 SI=0051 DI=0000
DS=076C ES=075A SS=076C CS=076A IP=001E  NU UP EI PL NZ NA PO NC
076A:001E CC          INT     3
-d ds:0050,0051
076C:0050 1C 00          ..

```

Q5.) Write an 8086 Assembly Language Program for the addition of 5 sixteen-bit numbers stored from DS: 0030H. Store the result in DS: 0050H, DS: 0051H, DS: 0052H.

```

.model small
.stack 100h
.data
.code
main proc
    mov ax, @data
    mov ds, ax

    mov ax, 0000h
    mov bl, 00h
    mov cl, 05h
    mov si, 0030h

l1:
    add ax, [si]
    adc bl, 00h
    inc si
    inc si
    loop l1

    mov si, 0050h
    mov [si], ax
    inc si
    inc si
    mov [si], bl

    int 03h
main endp
end main

```

```

C:\>debug q5.exe
-t

AX=076C BX=0000 CX=0022 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076D CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-076c:0030
^ Error
-e 076c:0030
076C:0030 01.01 00.00 02.02 00.00 03.03 00.00 04.04 00.00
076C:0038 05.05 00.00

-g=0000

AX=000F BX=0000 CX=0000 DX=0000 SP=0100 BP=0000 SI=0052 DI=0000
DS=076C ES=075A SS=076D CS=076A IP=0021  NU UP EI PL NZ NA PO NC
076A:0021 CC          INT     3
-d ds:0050,0052
076C:0050 0F 00 00          ...

```

Q6.) Write an Assembly Language Program for the addition of five BCD numbers stored from DS: 0030H. Store the result in DS: 0040H and DS: 0041H.

```

.model small
.stack 100h
.data
.code
main proc
    mov ax, @data
    mov ds, ax

    mov ax, 0000h
    mov cx, 0005h
    mov si, 0030h

l1:
    add al, [si]
    daa
    adc ah, 00h
    inc si
    loop l1

    mov si, 0040h
    mov [si], ax

    int 03h
main endp
end main

```

```

AX=076B BX=0000 CX=001D DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076C CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076b:0030
076B:0030 00.01  52.02  50.03  E8.04  EA.05

-g=0000

AX=0015 BX=0000 CX=0000 DX=0000 SP=0100 BP=0000 SI=0040 DI=0000
DS=076B ES=075A SS=076C CS=076A IP=001C  NU UP EI PL NZ NA PE NC
076A:001C CC          INT     3
-d ds:0040,0041
076B:0040 15 00          ..

```

Q7.) Write an Assembly Language Program to subtract a BCD number stored in DS: 0040H from a BCD number stored in DS: 0050H. Store the result in DS: 0060H and DS: 0061H.

```

.model small
.stack 100h
.data
.code
main proc
    mov ax, @data
    mov ds, ax

    mov ax, 0000h
    mov si, 0040h
    mov al, [si]

    mov si, 0050h
    sub al, [si]
    das
    mov dl, 00h
    jnc l1

    mov cl, al
    mov al, 99h
    sub al, cl
    add al, 01h
    daa
    mov dl, 01h

l1:
    mov si, 0060h
    mov [si], al
    inc si
    mov [si], dl

    int 03h
main endp
end main

```

```

C:\>debug subbcd.exe
-t

AX=076C BX=0000 CX=002B DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076D CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0040
076C:0040 E4.09

-e 076c:0050
076C:0050 C4.04

-g=0000

AX=0005 BX=0000 CX=002B DX=0000 SP=0100 BP=0000 SI=0061 DI=0000
DS=076C ES=075A SS=076D CS=076A IP=002A  NU UP EI PL NZ NA PO NC
076A:002A CC          INT     3
-d ds:0060,0061
076C:0060 05 00          ..

```

Q8.) Write an Assembly Language Program to multiply two eight-bit numbers stored in DS: 0040H and DS: 0050H. Store the result in DS: 0060H.

```

.model small
.stack 100h
.data
.code
main proc
    mov ax, @data
    mov ds, ax

    mov ax, 0000h
    mov bl, 00h
    mov si, 0040h
    mov al, [si]

    mov si, 0050h
    mov bl, [si]
    mul bl

    mov si, 0060h
    mov [si], ax

    int 03h
main endp
end main

```



```

C:\>debug mul8bit.exe
-t

AX=076C BX=0000 CX=0020 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076C CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0040
076C:0040 09 02

-e 076c:0050
076C:0050 04 03

-g=0000

AX=0006 BX=0003 CX=0020 DX=0000 SP=0100 BP=0000 SI=0060 DI=0000
DS=076C ES=075A SS=076C CS=076A IP=001B  NU UP EI PL NZ NA PO NC
076A:001B CC          INT     3
-d ds:0060,0061
076C:0060 06 00          ..

```

Q9.) Write an Assembly Language Program to multiply two sixteen-bit numbers stored in DS: 0040H and DS: 0050H. Store the result in DS: 0060H.

```

.model small
.stack 100h
.data
.code
main proc
    mov ax, @data
    mov ds, ax

    mov ax, 0000h
    mov bx, 0000h

    mov si, 0040h
    mov ax, [si]
    mov si, 0050h
    mov bx, [si]
    mul bx

    mov si, 0060h
    mov [si], ax
    inc si
    inc si
    mov [si], dx

    int 03h
main endp
end main

```

```

C:\>debug mul16bit.exe
-t

AX=076C BX=0000 CX=0025 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076D CS=076A IP=0003  NV UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-e 076c:0040
076C:0040 02.04    40.00

-e 076c:0050
076C:0050 03.03    04.00

-g=0000

AX=000C BX=0003 CX=0025 DX=0000 SP=0100 BP=0000 SI=0062 DI=0000
DS=076C ES=075A SS=076D CS=076A IP=0020  NV UP EI PL NZ NA PO NC
076A:0020 CC          INT     3
-d ds:0060,0063
076C:0060 0C 00 00 00          ....

```

Q10.) Write an Assembly Language Program to divide 88H by 33H. Store the quotient in DS: 0060H and remainder in DS: 0061H

```

.model small
.stack 100h
.data
.code
main proc
    mov ax, @data
    mov ds, ax

    mov ax, 0000h
    mov al, 88h
    mov bl, 33h
    div bl

    mov si, 0060h
    mov [si], ax

    int 03h
main endp
end main

```

```

C:\>debug q10.exe
-t
AX=076B BX=0000 CX=0018 DX=0000 SP=0100 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=076C CS=076A IP=0003  NU UP EI PL NZ NA PO NC
076A:0003 8ED8          MOV     DS,AX
-g=0000
AX=2202 BX=0033 CX=0018 DX=0000 SP=0100 BP=0000 SI=0060 DI=0000
DS=076B ES=075A SS=076C CS=076A IP=0013  NU UP EI PL NZ NA PO NC
076A:0013 CC          INT     3
-d ds:0060,0061
076B:0060 02 22          ."

```

Q11.) Write an Assembly Language Program to divide 2222H by 55H. Store the quotient in DS: 0060H and remainder in DS: 0062H.

```

.model small
.stack 100h
.data
.code
main proc
    mov ax, @data
    mov ds, ax

    mov dx, 0000h
    mov ax, 2222h
    mov bx, 0055h
    div bx

    mov si, 0060h
    mov [si], ax
    inc si
    inc si
    mov [si], dx

    int 03h
main endp
end main

```

```
C:\>debug q11.exe
```

```
-g=0000
```

```
AX=0066  BX=0055  CX=001E  DX=0044  SP=0100  BP=0000  SI=0062  DI=0000
```

```
DS=076B  ES=075A  SS=076C  CS=076A  IP=0019  NU UP EI PL NZ NA PO NC
```

```
076A:0019 CC          INT      3
```

```
-d ds:0060,0063
```

```
076B:0060  66 00 44 00
```

```
f.D.
```
