UCS1512 – Microprocessors Lab
End Semester Practical Examination

Shivanirudh S G
CSE-C
185001146

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 store a block of 10 data in reverse from XX00 to XX09 in YY00 to YY09.

Algorithm:

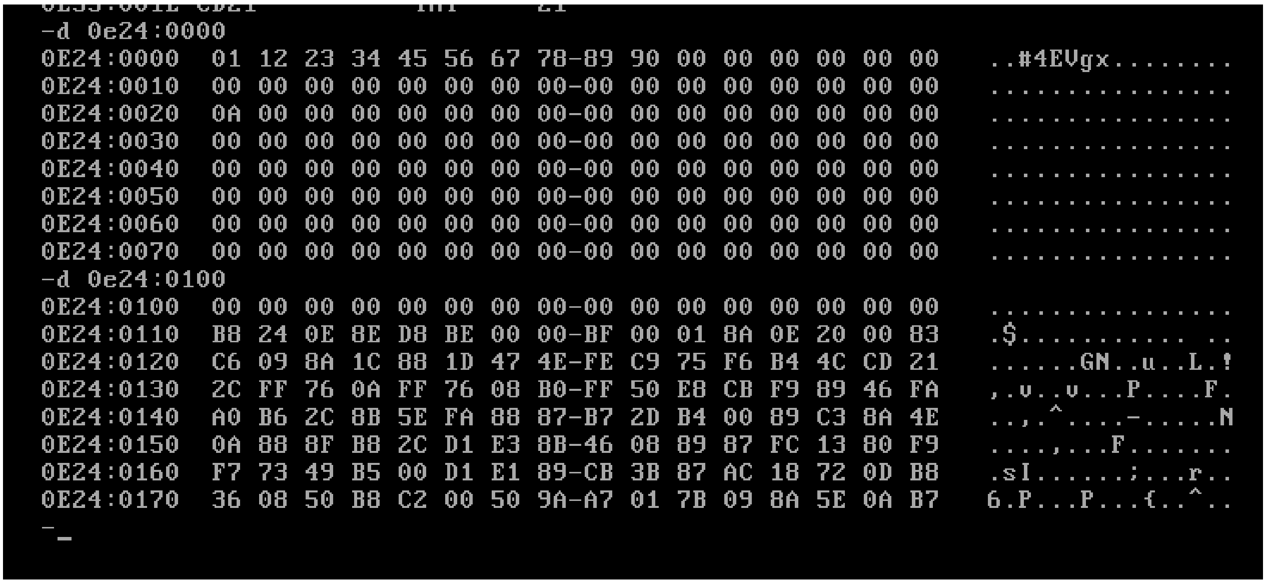
1. Move data to accumulator and then to the DS register.
2. Move the offsets of source and dest to SI and DI registers respectively.
3. Move the value of count to CL register.
4. Move the offset of source in SI to the end by adding one less than count.
5. Under label HERE, move the value at offset marked by SI register to BL register.
6. Move the value in BL register to offset marked by DI register.
7. Increment DI and decrement SI.
8. Decrement CL and jump back to label HERE if not zero.

Program:

```
;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.

assume ds:data, cs:code
data segment
org 00H
source db 01H, 12H, 23H, 34H, 45H, 56H, 67H, 78H, 89H, 90H
org 100H
dest db ?
org 20H
count db 0AH
data ends
code segment
start: mov ax, data
mov ds, ax

mov si, offset source
mov di, offset dest
mov cl, count
```

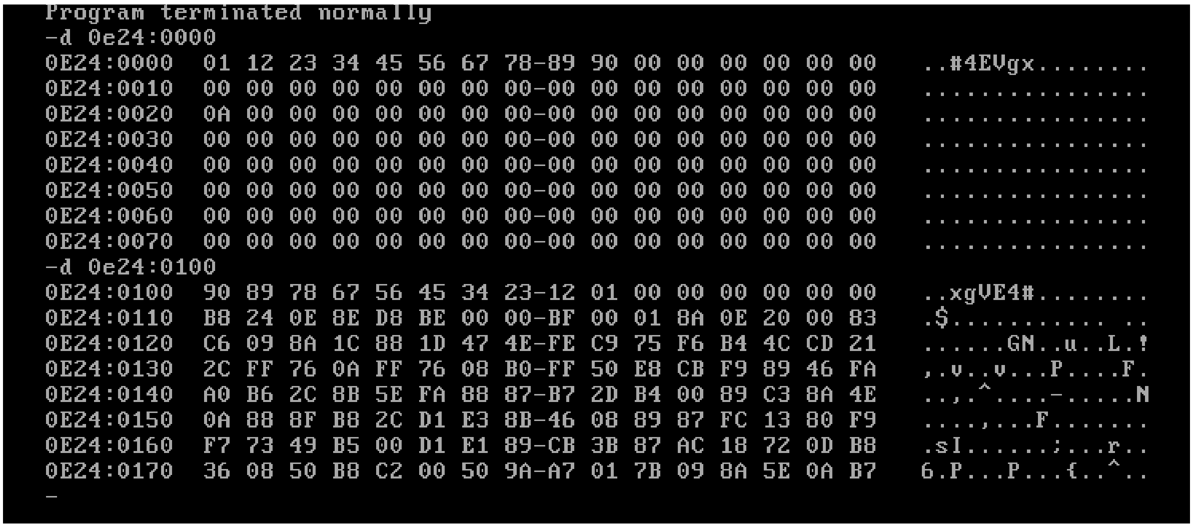
```
add si, 09H                    ;To move SI to end of the block

here: mov bl, [si]             ;Move from source to destination using
mov [di], bl                   ;BL register
inc di                          ;Increment DI for next destination
dec si                         ;Decrement SI for next source
dec cl
jnz here                        ;Continue till CL = 0
mov ah, 4ch
int 21h
code ends
end start
```

Input:



```
0E33:001E CD21          INT     21
-d 0e24:0000
0E24:0000  01 12 23 34 45 56 67 78-89 90 00 00 00 00 00 00   ..#4EVgx........
0E24:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0020  0A 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-d 0e24:0100
0E24:0100  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0110  B8 24 0E 8E D8 BE 00 00-BF 00 01 8A 0E 20 00 83   .$........... ..
0E24:0120  C6 09 8A 1C 88 1D 47 4E-FE C9 75 F6 B4 4C CD 21   ......GN..u..L.!
0E24:0130  2C FF 76 0A FF 76 08 B0-FF 50 E8 CB F9 89 46 FA   ,.v..v...P....F.
0E24:0140  A0 B6 2C 8B 5E FA 88 87-B7 2D B4 00 89 C3 8A 4E   ..,.^....-.....N
0E24:0150  0A 88 8F B8 2C D1 E3 8B-46 08 89 87 FC 13 80 F9   ....,...F.......
0E24:0160  F7 73 49 B5 00 D1 E1 89-CB 3B 87 AC 18 72 0D B8   .sI......;...r..
0E24:0170  36 08 50 B8 C2 00 50 9A-A7 01 7B 09 8A 5E 0A B7   6.P...P...{..^..
-_
```

Output:



```
Program terminated normally
-d 0e24:0000
0E24:0000  01 12 23 34 45 56 67 78-89 90 00 00 00 00 00 00   ..#4EVgx........
0E24:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0020  0A 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-d 0e24:0100
0E24:0100  90 89 78 67 56 45 34 23-12 01 00 00 00 00 00 00   ..xgVE4#........
0E24:0110  B8 24 0E 8E D8 BE 00 00-BF 00 01 8A 0E 20 00 83   .$........... ..
0E24:0120  C6 09 8A 1C 88 1D 47 4E-FE C9 75 F6 B4 4C CD 21   ......GN..u..L.!
0E24:0130  2C FF 76 0A FF 76 08 B0-FF 50 E8 CB F9 89 46 FA   ,.v..v...P....F.
0E24:0140  A0 B6 2C 8B 5E FA 88 87-B7 2D B4 00 89 C3 8A 4E   ..,.^....-.....N
0E24:0150  0A 88 8F B8 2C D1 E3 8B-46 08 89 87 FC 13 80 F9   ....,...F.......
0E24:0160  F7 73 49 B5 00 D1 E1 89-CB 3B 87 AC 18 72 0D B8   .sI......;...r..
0E24:0170  36 08 50 B8 C2 00 50 9A-A7 01 7B 09 8A 5E 0A B7   6.P...P...{..^..
-
```

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

Aim:

To perform 32-bit addition and subtraction.

ADDITION

Algorithm:

1. Move data to the accumulator and then to the DS register.
2. Move lower word of operand 1 to AX register, and higher word to BX register.
3. Move value 00H to CL register.
4. Add lower word of operand 2 to AX using ADD AX, opr2_l.
5. Add the higher word of operand 2 to BX with carry using ADC BX, opr2_h.
6. Jump to label HERE if no carry is generated.
7. Increment CL register.
8. Under label HERE, move value in AX to res_l and value in BX to res_h.
9. Move value in CL to carry.

Program:

```
;1.b Write ALP using 8086 to perform 32 bit addition

assume ds:data, cs:code
data segment
org 00H
opr1_l dw 0ABCDH            ;Operand 1 is 1234 ABCD H.
opr1_h dw 1234H
org 10H
opr2_l dw 1234H             ;Operand 2 is FBCD 1234 H.
opr2_h dw 0FBCDH
org 20H
res_l dw 0000H              ;Expected result: (Operand 1 + Operand 2)
res_h dw 0000H              ;0E01 BE01
carry db 00H                ;Carry: 01
data ends
code segment
start: mov ax, data
mov ds, ax

mov cl, 00h                 ;To store carry

mov ax, opr1_l
mov bx, opr1_h

add ax, opr2_l              ;Add lower words
adc bx, opr2_h              ;Add higher words with carry

jnc here
```

```
inc cl                                  ;Increment if carry generated

here: mov res_l, ax
mov res_h, bx
mov carry, cl

mov ah, 4ch
int 21h
code ends
end start
```

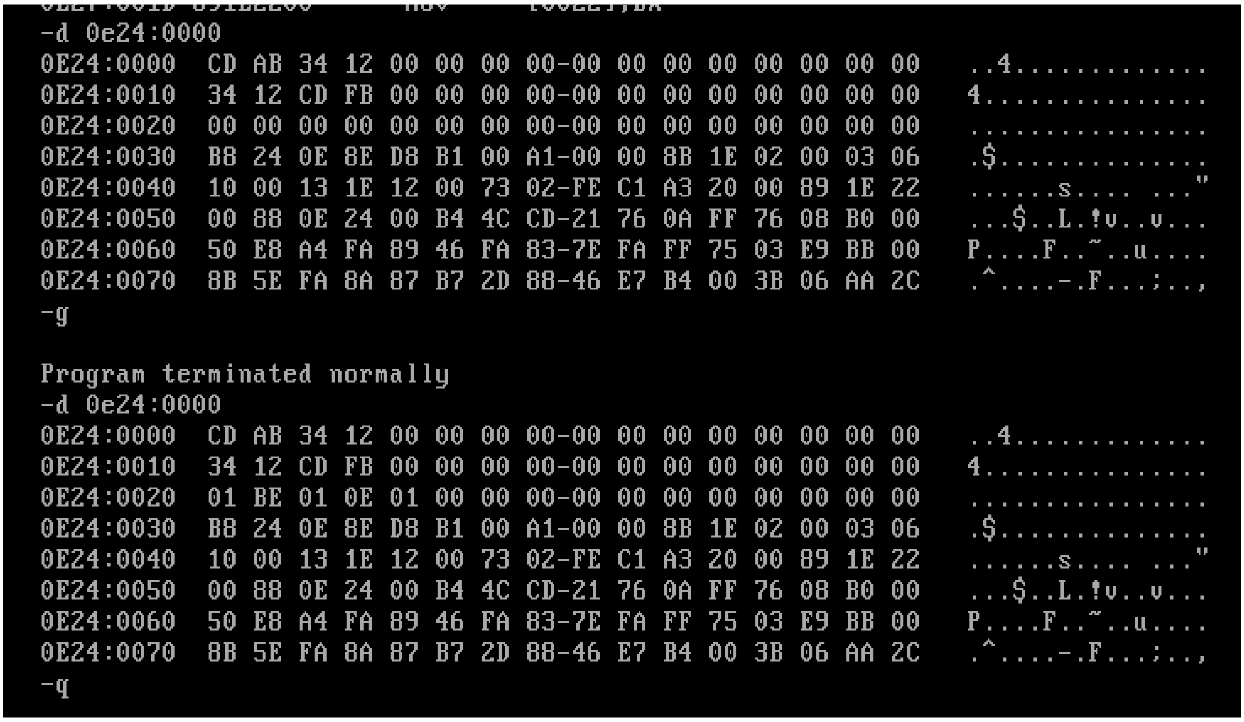
Input/Output:

Input:

    Operand 1 : 1234 ABCD
    Operand 2 : FBCD 1234

Output:

    Result : 0E01 BE01
    Carry: 01



```
-d 0e24:0000
0E24:0000  CD AB 34 12 00 00 00 00-00 00 00 00 00 00 00 00   ..4.............
0E24:0010  34 12 CD FB 00 00 00 00-00 00 00 00 00 00 00 00   4...............
0E24:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0030  B8 24 0E 8E D8 B1 00 A1-00 00 8B 1E 02 00 03 06   .$..............
0E24:0040  10 00 13 1E 12 00 73 02-FE C1 A3 20 00 89 1E 22   ......s.... ..."
0E24:0050  00 88 0E 24 00 B4 4C CD-21 76 0A FF 76 08 B0 00   ...$..L.!v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  CD AB 34 12 00 00 00 00-00 00 00 00 00 00 00 00   ..4.............
0E24:0010  34 12 CD FB 00 00 00 00-00 00 00 00 00 00 00 00   4...............
0E24:0020  01 BE 01 0E 01 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0030  B8 24 0E 8E D8 B1 00 A1-00 00 8B 1E 02 00 03 06   .$..............
0E24:0040  10 00 13 1E 12 00 73 02-FE C1 A3 20 00 89 1E 22   ......s.... ..."
0E24:0050  00 88 0E 24 00 B4 4C CD-21 76 0A FF 76 08 B0 00   ...$..L.!v..v...
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-q
```

SUBTRACTION

Algorithm:

1. Move data to the accumulator and then to the DS register.
2. Move lower word of operand 1 to AX register, and higher word to BX register.
3. Move value 00H to CL register.
4. Subtract lower word of operand 2 from AX using SUB AX, opr2_l.
5. Subtract the higher word of operand 2 from BX with carry using SBB BX, opr2_h.
6. Jump to label HERE if no borrow is generated.
7. Negate AX and BX registers using NEG AX and NEG BX.
8. Decrement BX register to adjust for 2’s complement
7. Increment CL register.
8. Under label HERE, move value in AX to res_l and value in BX to res_h.
9. Move value in CL to borrow.

Program:

```
;1.b Write ALP using 8086 to perform 32 bit subtraction

assume ds:data, cs:code
data segment
org 00H
opr1_l dw 0ABCDH           ;Operand 1 is 1234 ABCD H
opr1_h dw 1234H
org 10H
opr2_l dw 0FFFFH           ;Operand 2 is ABCD FFFF H
opr2_h dw 0ABCDH
org 20H
res_l dw 0000H             ;Expected result: (Operand 1 - Operand 2)
res_h dw 0000H             ;9999 5432
borrow db 00H              ;Borrow: 01
data ends
code segment
start: mov ax, data
mov ds, ax

mov cl, 00h                ;To store borrow
mov ax, opr1_l
mov bx, opr1_h

sub ax, opr2_l             ;Subtract lower words
sbb bx, opr2_h             ;Subtract higher words with carry

jnc here
neg ax                    ;2's complement of AX.
neg bx                    ;2's complement of BX.
dec bx                     ;Decrement BX to adjust for 2's complement,
                          ;as (BX)(AX) is a single value
```

```
inc cl                              ;Increment if borrow occurs

here: mov res_l, ax
mov res_h, bx
mov borrow, cl

mov ah, 4ch
int 21h
code ends
end start
```

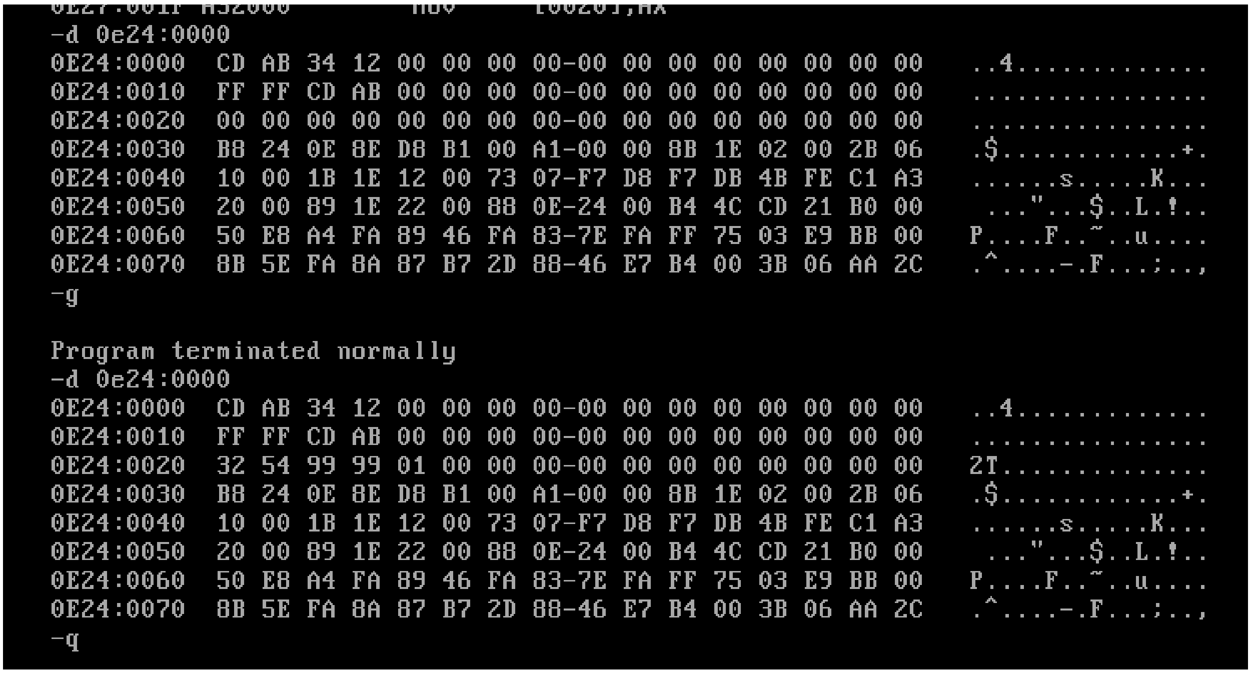
Input/Output:

Input:

Operand 1 : 1234 ABCD
Operand 2 : ABCD FFFF

Output:

Result : 9999 5432
Borrow: 01



```
0E27:001F A32000        MOV     [0020],AX
-d 0e24:0000
0E24:0000  CD AB 34 12 00 00 00 00-00 00 00 00 00 00 00 00   ..4.............
0E24:0010  FF FF CD AB 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0030  B8 24 0E 8E D8 B1 00 A1-00 00 8B 1E 02 00 2B 06   .$............+.
0E24:0040  10 00 1B 1E 12 00 73 07-F7 D8 F7 DB 4B FE C1 A3   ......s.....K...
0E24:0050  20 00 89 1E 22 00 88 0E-24 00 B4 4C CD 21 B0 00    ..."...$..L.!..
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-g

Program terminated normally
-d 0e24:0000
0E24:0000  CD AB 34 12 00 00 00 00-00 00 00 00 00 00 00 00   ..4.............
0E24:0010  FF FF CD AB 00 00 00 00-00 00 00 00 00 00 00 00   ................
0E24:0020  32 54 99 99 01 00 00 00-00 00 00 00 00 00 00 00   2T..............
0E24:0030  B8 24 0E 8E D8 B1 00 A1-00 00 8B 1E 02 00 2B 06   .$............+.
0E24:0040  10 00 1B 1E 12 00 73 07-F7 D8 F7 DB 4B FE C1 A3   ......s.....K...
0E24:0050  20 00 89 1E 22 00 88 0E-24 00 B4 4C CD 21 B0 00    ..."...$..L.!..
0E24:0060  50 E8 A4 FA 89 46 FA 83-7E FA FF 75 03 E9 BB 00   P....F..~..u....
0E24:0070  8B 5E FA 8A 87 B7 2D 88-46 E7 B4 00 3B 06 AA 2C   .^....-.F...;..,
-q
```

Result:

8086 programs were written to perform transfer and storing of block of data in reverse, 32-bit addition and subtraction, and results observed.