# ECE3210 Microprocessor Engineering

## Homework 4

1. **Chapter 4. 8**
   8B07 => 1000 1011 0000 0111

   D=1: REG ← R/M

   W=1: word

   MOD =00 : No displacement

   REG =000: AX

   R/M = 111: DS:[BX]

   MOV AX, DS:[BX]

2. **Chapter 4. 9**

   8B9E004C --> 1000 1011 1001 1110 0000 0000 0100 1100

   D=1: REG ← R/M

   W=1: word

   MOD =10 : 16-bit displacement

   REG =011: BX

   R/M = 110: SS:[BP]

   MOV BX, SS:[BP+4C00]

3. **Chapter 4.10**

   MOV SI, [BX+2]

   1000 1011 0111 0111 0000 0002

   8B 77 02

4. **Chapter 4.11**
   MOV ESI, [EAX]
   67 66 8B 30 => 1000 1011 0011 000

| R/M Code | Function |
|---|---|
| 000 | DS:[EAX] |
| 001 | DS:[ECX[ |
| 010 | DS:[EDX] |
| 011 | DS:[EBX] |
| 100 | Uses scaled-index byte |
| 101 | SS:[EBP]* |
| 110 | DS:[ESI] |
| 111 | DS:[EDI] |

*Note:* See text section, Special Addressing Mode.

5. **Chapter 4.15**

   32

6. **Chapter 4.17**

   CS

7. What is the output of the following program

   Solution: The modified string is: ABCDEABCIJ

```
.DATA
     NEWLINE         DB 0DH,0AH,'$'
     MESSAGE1   DB 'The modified string is: $'
     STRING     DB 'ABCDEFGHIJ','$'

.CODE
        .STARTUP
     MAIN PROC FAR
          ;setup
          LEA   SI, STRING
          LEA   DI, STRING+5
          CLD

          PUSH ES
          PUSH DS
          POP ES

          MOVSB
          MOVSB
          MOVSB

          POP ES

          LEA         DX,MESSAGE1              ;PRINT MESSAGE
          MOV         AH,9
```

```
        INT     21H

        LEA     DX,STRING               ;PRINT MESSAGE
        MOV     AH,9
        INT     21H

        .EXIT
    MAIN ENDP

END
```

**8.** Repeat question 10, replacing the setup code with

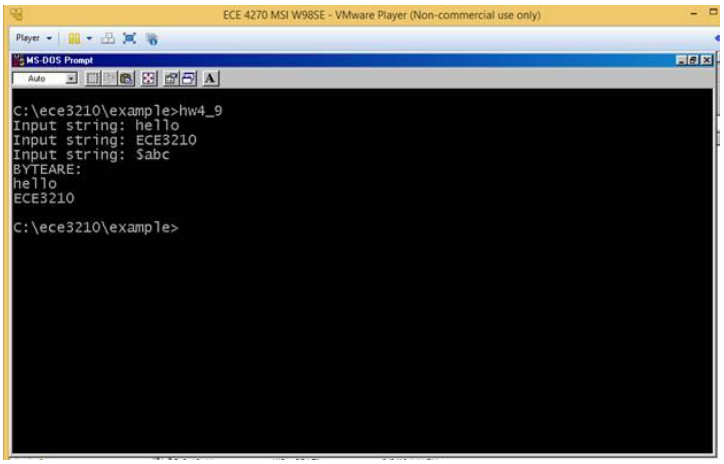
```
LEA  SI, STRING+9
LEA DI, STRING+7
STD
```

Solution: The modified string is: ABCDEJIJIJ

**9.** Write a program that reserves a 1024 bytes long area named *byteArea*. Use a dialog to input string one at a time into another 100 bytes area named *stringIn*. Copy the first string to the beginning of *byteArea*, then append carriage return (0DH) and linefeed (0AH) characters to the end of the copy. Copy the next string from *stringIn* to *byteArea*, starting right after the first string's linefeed. Terminate input when the first character of *stringIn* is $ and append a byte ('$') to the destination right after the last string's linefeed. Finally, display all the characters in *byteArea* on screen. The result should be the strings that were entered, one per line. Use REP MOVSB instruction. Include a screenshot.

```
.MODEL MEDIUM

.STACK 100H

.DATA
        BYTEARE   DB 1024 DUP (?)
        BUFFER DB 100
        COUNT  DB ?
        STRINGIN DB 100 DUP ('$')

        NEWLINE             DB 0DH,0AH,'$'
        MESSAGE1DB 'Input string: $'
        MESSAGE2DB 'BYTEARE: ', 0DH, 0AH, '$'

.CODE
    .STARTUP
        MAIN PROC FAR
                PUSH ES
                PUSH DS
                POP ES
                CLD
```

```
                    LEA DI, BYTEARE
BEGIN:              LEA     SI, STRINGIN

                    LEA DX,MESSAGE1 ;PRINT MESSAGE
                     MOV AH,9
                    INT 21H

                    MOV DX, OFFSET BUFFER
                    MOV AH, 0AH
                    INT 21H

                    LEA DX,NEWLINE ;PRINT MESSAGE
                     MOV AH,9
                     INT 21H

                    CMP STRINGIN, '$'
                    JE FINISH
                    MOV CX, 0
                    MOV CL, COUNT
                    REP MOVSB

                    MOV CX, 2H
                    LEA SI, NEWLINE
                    REP MOVSB
                    JMP BEGIN
FINISH:             MOV BYTE PTR[DI], '$'
                    POP ES

                    LEA             DX,MESSAGE2             ;PRINT MESSAGE
                    MOV             AH,9
                    INT             21H

                    LEA             DX,BYTEARE              ;PRINT MESSAGE
                    MOV             AH,9
                    INT             21H

                    .EXIT
          MAIN ENDP

END
```




----------End----------