



Name:  
Number:

Date: 18 November 2016

1	
2	
3	
4	
T	

## EENG410/INFE410 - MICROPROCESSORS I Midterm Exam

*Instructor: Prof. Dr. Hasan Demirel*

Read the Following Instructions Carefully:

1. The duration of the exam is **90** minutes.
2. Answer each question to a separate sheet on your answers booklet.

### QUESTIONS & SOLUTIONS

1. (%25) What will be the values of the specified register(s) and flags after the execution of the following instructions?

<p><b>a)</b> MOV AX, 389CH MOV BX, 1A8CH SHL BL, 1 ; CF=1, BX=1A18H ADC AH, BL</p> <table style="margin-left: 40px; border-collapse: collapse;"> <tr><td style="padding-right: 20px;">38H</td><td>0011 1000</td></tr> <tr><td>18H</td><td>0001 1000</td></tr> <tr><td style="border-bottom: 1px solid black;">+1</td><td style="border-bottom: 1px solid black;">+ 1</td></tr> <tr><td>AH=51H</td><td><b>0101 0001</b></td></tr> </table>	38H	0011 1000	18H	0001 1000	+1	+ 1	AH=51H	<b>0101 0001</b>	<p><b>b)</b> MOV AX, 1207H XOR AL, AH ; (1) AX=1215H MUL AH ; (2) AX=017AH SUB AH, AL ; (3)</p> <table style="margin-left: 40px; border-collapse: collapse;"> <tr><td style="padding-right: 20px;">(1) 07H</td><td>0000 0111</td></tr> <tr><td>12H</td><td>0001 0010</td></tr> <tr><td style="border-bottom: 1px solid black;">XOR</td><td style="border-bottom: 1px solid black;">XOR</td></tr> <tr><td>AL=15H</td><td><b>0001 0101</b></td></tr> </table> <p style="margin-left: 40px;">(2) AX=15H*12H= <b>017AH</b></p> <table style="margin-left: 40px; border-collapse: collapse;"> <tr><td style="padding-right: 20px;">(3) 01H</td><td>0000 0001</td></tr> <tr><td>7AH</td><td>1000 0110</td></tr> <tr><td style="border-bottom: 1px solid black;">-</td><td style="border-bottom: 1px solid black;">+</td></tr> <tr><td>AH=87H</td><td><b>1000 0111</b></td></tr> </table> <p style="margin-left: 40px;">01111010 (2s comp) 10000110</p>	(1) 07H	0000 0111	12H	0001 0010	XOR	XOR	AL=15H	<b>0001 0101</b>	(3) 01H	0000 0001	7AH	1000 0110	-	+	AH=87H	<b>1000 0111</b>
38H	0011 1000																								
18H	0001 1000																								
+1	+ 1																								
AH=51H	<b>0101 0001</b>																								
(1) 07H	0000 0111																								
12H	0001 0010																								
XOR	XOR																								
AL=15H	<b>0001 0101</b>																								
(3) 01H	0000 0001																								
7AH	1000 0110																								
-	+																								
AH=87H	<b>1000 0111</b>																								
<p>BX=<b>1A18H</b>, AX=<b>519CH</b>, CF=<b>0</b>, PF=<b>0</b>, AF=<b>1</b>, SF=<b>0</b>, ZF=<b>0</b></p>	<p>AX=<b>877AH</b>, CF=<b>1</b>, PF=<b>1</b>, AF=<b>0</b>, SF=<b>1</b>, ZF=<b>0</b></p>																								

<p><b>c)</b> SUB AX, AX ; AX=<b>0000H</b> SUB AX, 1 ; AX=<b>FFFFH</b> PUSH AX POPF</p>
<p>AX=<b>FFFFH</b> CF=<b>1</b>, PF=<b>1</b>, AF=<b>1</b>, SF=<b>1</b>, ZF=<b>1</b></p>

- 
2. (%25) Assume that an array of decimal numbers is given in A. Write an assembly language program that finds the minimum of the odd numbers and the maximum of the even numbers. Save the minimum to ODDMIN and maximum to EVENMAX variables in the data segment.

A = 98,77,46,87,78,91,79,22,75,69,41,82

```
.MODEL SMALL
.STACK 64
.DATA
A      DB      98,77,46,87,78,91,79,22,75,69,41,82
ODDMIN DB      ?
EVENMAX DB     ?
.CODE
MAIN:  MOV AX,@DATA
       MOV DS,AX
       ; *****FINDING ODDMIN*****
       MOV CX,12
       MOV BH,0FFH
       MOV SI,OFFSET A
BACK:  MOV AL,[SI]
       SHR AL,1
       JNC NEXT
       MOV BL,[SI]
       CMP BL,BH
       JA  NEXT
       MOV BH,BL
NEXT:  INC SI
       LOOP BACK
       MOV ODDMIN,BH
       ; *****FINDING EVENMAX*****
       MOV CX,12
       MOV BH,00H
       MOV SI,OFFSET A
BACK2:MOV AL,[SI]
       SHR AL,1
       JC  NEXT2
       MOV BL,[SI]
       CMP BL,BH
       JB  NEXT2
       MOV BH,BL
NEXT2:INC SI
       LOOP BACK2
       MOV EVENMAX,BH
       MOV AH,4CH
       INT 21H
       END MAIN
```

---

---

3. (%25) Define three separate subroutines for each of the following tasks respectively.

- Multiply the two bytes pointed by SI and BX registers and add this multiplication to the word pointed by DI. The final result is to be saved to offset address 0300H in the data segment.

```
MULTIPLY:  MOV AL, [SI]
           MOV AH, [BX]
           MUL AH
           ADD AX, [DI]
           MOV [0300], AX
           RET
```

- Clear all the bits of the register AL if the least significant bit and the most significant bit of AL are both 0. Otherwise, all the bits of AL are set to 1.

```
MULTIPLY:  MOV BL, AL
           SHR BL, 1
           JC OVER
           MOV BL, AL
           SHL BL, 1
           JC OVER
           AND AL, 00000000B
           JMP NEXT
OVER:      OR AL, 11111111B
NEXT:     RET
```

- Multiply the least significant nibble of DL with the most significant nibble of BH. The result is to be saved into AL.

```
MULTIPLY:  AND DL, 0FH
           MOV AL, DL
           AND BH, 0F0H
           MOV CL, 4
           SHL BH, CL
           MUL BH
           RET
```

---

4. (%25) Assume that a string of name & surname is saved in S. The alphabetical characters in S can be in lowercase and/or uppercase letters. Name and surname are assumed to be separated by a space character and the string ends with a full stop "." character.

Write an assembly language program that will copy the name to *NAME* in lowercase and the surname to *SNAME* in uppercase letters. Assume that name and/or surname cannot exceed 20 characters.

The program should be general and work with every possible string with name & surname. However, you can consider the data segment definition given below in your program.

```
.DATA
S          DB    'Ahmed Goldsmith.'
NAME       DB    20 DUP(?)
SNAME     DB    20 DUP(?)
```

Hint: Uppercase characters are ordered between 'A' (41H) and 'Z' (5AH) and lowercase characters are ordered between 'a' (61H) and 'z' (7AH) in the in the ASCII Code table. For lowercase letters, bit 5 (d5) of the ASCII code is 1 where for uppercase letters it is 0. For example,

Letter	Binary	ASCII
'h'	01101000	68H
'H'	01001000	48H

```
.MODEL SMALL
.STACK 64
.DATA
S          DB      'Ahmed Goldsmith.'
NAME       DB      20 DUP(?)
SNAME      DB      20 DUP(?)
.CODE
MAIN: MOV AX,@DATA
      MOV DS,AX
      MOV SI,OFFSET S
      MOV BX,OFFSET NAME
      MOV DI,OFFSET SNAME
; ***Processing NAME (Assume that Name is all letters) ***
BACK1:MOV AL,[SI]
      CMP AL,\' \'
      JE SURNAME
      OR AL,00100000B
      MOV [BX],AL
      INC SI
      INC BX
      JMP BACK1
;***Processing SNAME (Assume that Surname is all letters) ***
SURNAME:INC SI
BACK2:MOV AL,[SI]
      CMP AL,\'.\'
      JE EXIT
      AND AL,11011111B
      MOV [DI],AL
      INC SI
      INC DI
      JMP BACK2
EXIT:MOV AH,4CH
      INT 21H
      END MAIN
```

---