



| | |
|---------------------------------------|--|
| Course Code: EE213 | Course Name: Computer Organization and Assembly Language |
| Instructor Name: Muhammad Danish Khan | |

Instructions:

- Attempt all questions, containing equal marks. Return the question paper.
- Read each question completely before answering it. There are **4 questions on 2 pages**.
- Make assumptions, if required, without contradicting any statement in the question paper.
- **Answers must be provided according to the questions sequence** given in the question paper. Examples/Comments are **necessary** where asked.
- **Provide only the hex-decimal values where asked.** Values are provided with their radixes, consider decimal otherwise. Consider 32bit architecture for all the instructions.
- This paper is subjective.

Time: 60 minutes.

Max Points: 40

Question 1(A)

Show the values of required registers/flags after execution of the following independent instructions:

(5 Points)

1. MOV AL, 42H
SHR AL, 1
AL = 21H
CF = 0

2. MOV AL, 0D4H
SAR AL, 1
AL = EAH
CF = 0

3. MOV AL, 0D4H
ROL AL, 4
AL = 4DH
CF = 1

4. MOV AL, 0D4H
STC
RCR AL, 3
AL = 3AH
CF = 1

5. MOV AL, 00001111B
TEST AL, 00000001B
CF = 0 ZF = 0

Question 1(B)

The *date* field of a file directory entry uses bits 8 to 15 for the YEAR, bits 4 to 7 for the month, and bits 0 to 3 for the DAY. Write instructions to copy the year to a byte variable *byear*.

(5 Points)

```
MOV AX, WORD PTR date
MOV byear, AH
```

Question 2 (A)

Given that op1, op2, op3 and X are integer variables, write equivalent assembly code for the following code snippet, *insert comments to clarify your code*: **(5 Points)**

```
while( op1 < op2 )
{
    op1++;
    if( op1 == op3 )
        X = 2;
    else
        X = 3;
}
```

```
MOV EAX, op1
L1:  CMP EAX, op2
     JB L2
     JMP EXIT
```

```
L2:  INC op1
     MOV EAX, op1
     CMP EAX, op3
     JE L3
     MOV X, 3
     JMP L1
```

```
L3:  MOV X, 2
     JMP L1
```

EXIT: ret

Question 2(B)

Given that VAR1, VAR2, and VAR3 are signed double words, write a code to store the largest one of these signed double words in VAR4. **(5 Points)**

| | | | | |
|---|--|--|--|--|
| MOV EAX, VAR1 MOV EBX, VAR2 MOV ECX, VAR3 | CMP EAX, EBX JG C2 C3: CMP EBX, EAX JG C4 C5: CMP ECX, EAX JG C6 RET | C2: CMP EAX, ECX JLE C3 MOV VAR4, EAX RET | C4: CMP EBX, ECX JLE C5 MOV VAR4, EBX RET | C6: CMP ECX, EAX JLE C7 MOV VAR4, ECX C7: RET |
|---|--|--|--|--|

Question 3

Write a code snippet for each of the following set of operations: **(10 Points)**

- I. Copy the second last element of the stack into ECX, without changing the stack contents.

```
MOV ECX, [ESP+4]
```

- II. Exchange the top two elements of the stack segment.

```
MOV EAX, [ESP]
MOV EBX, [ESP+4]
MOV [ESP], EBX
MOV [ESP+4], EAX
```

- III. Replace the contents of Accumulator with its mathematical cube (x^3).
- ```
MOVZX BX, AL
MUL AL
MUL BX
```
- IV. Loop through a DWORD array and calculate sum of all its elements in EAX.
- ```
MOV esi, 0
MOV eax, 0
MOV ECX, LENGTHOF array

L1: ADD EAX, [array+esi]
    ADD esi, 4
    LOOP L1
```
- V. Copy return address of a procedure into EBX.
- ```
main PROC
 CALL f1
 RET
main ENDP

f1 PROC
 MOV EBX, [ESP]
 RET
f1 ENDP
```

#### Question 4

Suppose an OCR machine report errors using an error byte at memory offset F600h, called status byte. The meaning of different bits of status byte is shown below:

| Bit | Message           | Meaning                                                 |
|-----|-------------------|---------------------------------------------------------|
| 0   | Short document    | The document just read is shorter than anticipated      |
| 1   | Long document     | The document just read is longer than anticipated       |
| 2   | Close feed        | Current document is too close to the preceding document |
| 3   | Multiple feed     | Two documents were detected at the same time            |
| 4   | Excessive skew    | The document is skewed (crooked) in the transport       |
| 5   | Document misfeed  | The document fails to feed into the transport           |
| 6   | Document jam      | The document jammed in the transport                    |
| 7   | Unspecified error | An unknown/unspecified error occurred                   |

Note that each bit represents a specific error condition indicated by the OCR machine. Now write a main procedure that wait in a loop to check a non-zero value of status byte. As soon as it gets a non-zero value, it checks all bits starting from bit 0 and use **WriteString** function to print the string shown in the Meaning column. Assume that the **WriteString** function takes address of string in register EDX. After printing the main procedure reset the status byte to zero and terminates. **(10 Points)**

```
monitor: MOV AL, status_byte
 CMP AL, 0
 JE monitor

 MOV ECX, 8
 MOV EBX, 0
 MOV BL, 1h
 again: TEST AL, BL
 JZ NEXT
 MOV EDX, STR1[EBX] ;str1 contains strings, element 0 empty.
 CALL WriteStr
 next: SHL BL, 1
 LOOP again
 JMP monitor
```

**BEST OF LUCK!**