

Instructions	Opcode	Instructions	Opcode
MOV reg, reg MOV reg, mem MOV mem, reg	100010dw oorrmmmm disp	RCL reg, 1 RCL mem, 1	1101000w oo010mmmm disp
MOV reg, imm	1011wrrr data	RCR reg, 1 RCR mem, 1	1101000w oo011mmmm disp
MOV mem, imm	1100011w oo000mmmm disp data	ROL reg, 1 ROL mem, 1	1101000w oo000mmmm disp
MOVSX reg, reg MOVSX reg, mem	00001111 1011111w oorrrmmmm disp	ROR reg, 1 ROR mem, 1	1101000w oo001mmmm disp
MOVZX reg, reg MOVZX reg, mem	00001111 1011011w oorrrmmmm disp	RCL reg, CL RCL mem, CL	1101001w oo010mmmm disp
MUL reg MUL mem	1111011w oo100mmmm disp	RCR reg, CL RCR mem, CL	1101001w oo011mmmm disp
NEG reg NEG mem	1111011w oo011mmmm disp	ROL reg, CL ROL mem, CL	1101001w oo000mmmm disp
NOT reg NOT mem	1111011w oo010mmmm disp	ROR reg, CL ROR mem, CL	1101001w oo001mmmm disp
OR reg, reg OR reg, mem OR mem, reg	000010dw oorrmmmm disp	RCL reg, imm RCL mem, imm	1100000w oo010mmmm disp data
OR reg, imm OR mem, imm	1000000w oo001mmmm disp	RCR reg, imm RCR mem, imm	1100000w oo011mmmm disp data
POP reg	01011rrr	ROL reg, imm ROL mem, imm	1100000w oo000mmmm disp data
POP mem	10001111 oo000mmmm disp	ROR reg, imm ROR mem, imm	1100000w oo001mmmm disp data
PUSH reg	01010rrr	ADD reg, reg ADD reg, mem ADD mem, reg	000000dw oorrmmmm disp
PUSH mem	11111111 oo110mmmm disp	SUB reg, reg SUB reg, mem SUB mem, reg	001010dw oorrmmmm disp
PUSH imm	01101000 data	LOOP Label	11100010 disp
INC reg INC mem	1111111w oo000mmmm disp	DEC reg DEC mem	1111111w oo001mmmm disp

MOD = 11			Direct Effective Address			
R/M	W=0	W=1	R/M	MOD = 00	MOD = 01	MOD = 10
000	AL	AX	000	[BX]+[SI]	[BX]+[SI]+D ₈	[BX]+[SI]+D ₁₆
001	CL	CX	001	[BX]+[DI]	[BX]+[DI]+D ₈	[BX]+[DI]+D ₁₆
010	DL	DX	010	[BP]+[SI]	[BP]+[SI]+D ₈	[BP]+[SI]+D ₁₆
011	BL	BX	011	[BP]+[DI]	[BP]+[DI]+D ₈	[BP]+[DI]+D ₁₆
100	AH	SP	100	[SI]	[SI]+D ₈	[SI]+D ₁₆
101	CH	BP	101	[DI]	[DI]+D ₈	[DI]+D ₁₆
110	DH	SI	110	Direct Address	[BP]+D ₈	[BP]+D ₁₆
111	BH	DI	111	[BX]	[BX]+D ₈	[BX]+D ₁₆

National University of Computer and Emerging Sciences
Karachi Campus

- VII. Differentiate between CMP and SUB instruction.
- VIII. Elaborate through an example instruction, how does SCASW differ from LODSW?
- IX. Explain the difference between CBW and MOVSB instruction.
- X. Elaborate through an example, how does PUSHFD differ from PUSHAD?

CLO # 3: Apply translation of machine instructions into binary code and vice versa

Q2: Convert the following independent Assembly Language instructions to Machine Language code – give your answers in hexadecimal (binary answers will not be graded): [5 x 2 = 10 Marks]

- I. MOVZX CX, VAR1 ; Var1 = 18h
- II. POP 0F12C [BX]
- III. MOV BX, DX
- IV. INC [BP][DI] + 2C
- V. RCL [BX], 3

CLO # 3: Apply translation of machine instructions into binary code and vice versa

Q3: Convert the following hexadecimal machine codes to assembly language mnemonics. State what each of the byte fields mean: [5 x 2 = 10 Marks]

- I. 03 84 2B 1A
- II. 2B 1D
- III. F6 D3
- IV. 56
- V. 0B 0A

CLO # 2: Create basic assembly code using different type of addressing modes in x86 & RISC ISAs to solve simple-moderate problems

Q4: Write a procedure ConvertBinToDec that takes an array of binary values as an input. You need to convert that binary value into its equivalent decimal value. Call WriteDec/WriteInt procedures are only allowed to display final result. [10 Marks]
consider the following example

input BYTE 0,0,1,0,1,0,0,0 ; Binary inputs given by user
call ConvertBinToDec
call WriteDec ; 40 is displayed

CLO # 2: Create basic assembly code using different type of addressing modes in x86 & RISC ISAs to solve simple-moderate problems

Q5: Write an assembly language procedure that adds the given two 256-bit numbers. Assume a 32-bit architecture. [10 Marks]

op1 QWORD 4 DUP(1234567812345678h)
op2 QWORD 4 DUP(8765432187654321h)

Computer Organization and Assembly language Final Exam

Date: December 18th 2024

Course Instructor(s)

Mr. Shoaib Rauf, Mr. Kashan Hussain, Mr. Aashir Mahboob, Ms. Atiya
Jokhio, Mr. M. Kariz Kamal, Mr. M. Usman, Mr. Nauraiz Subhan

Total Time (Hrs): 3

Total Marks: 100


Total Questions: 8

23K-0806

Roll No

3H

Section


Student Signature

Do not write below this line

INSTRUCTION: Attempt all the questions in-order.

CLO # 2: Identify use of processor register and addressing modes in various instructions for solving arithmetic data transfer and conditional processing

Q1: Briefly answer each of the following:

[10 x 2 = 20 Marks]

- Briefly discuss Pros and Cons of Register Parameters and Stack Parameters.
- Explain how a LOCAL directive is Different from ENTER instruction? Give two reasons.
- Consider the following Code:

```
Mysum Proc,  
A1:BYTE  
LOCAL A2[5]: BYTE  
MOV esi, Offset A1 ; Instruction 1  
MOV edi, ADDR A2 ; Instruction 2  
ret  
Mysum ENDP
```

Discuss whether above instructions are valid or not? Give reasons and write updated instruction if invalid.

- Consider the following code:

```
mov ax, 0h  
mov cx, 0Ah  
doLoop:  
dec ax  
loop doLoop
```

What is the value of the ax register after the completion of the doLoop?

- Using SUB and Shift instructions, multiply a number stored in EAX register by 27d.
- Support the best statement about LOOPZ:
 - LOOPZ instruction executes a block of code repeatedly by checking whether ECX is greater than zero and that Zero Flag is CLEAR.
 - LOOPZ instruction executes a block of code repeatedly by checking whether ECX is greater than zero and that Zero Flag is SET.
 - LOOPZ executes a block of code repeatedly by checking only whether ECX is equal to zero
 - LOOPZ instruction executes a block of code repeatedly by checking only whether Zero Flag is SET.

National University of Computer and Emerging Sciences
Karachi Campus

CLO # 4: Illustrate use of stack during a parametrized function/procedure call that uses local variables.

Q6: Write an assembly language program to count the number of words in the given string. [15 Marks]

```
.data
msg0 BYTE "string_1 Word Count is: ",0
string_1 BYTE "Nothing is worth more than this day",0
```

1 2 3 4 5 6 7

In the above string, collection of characters leading with space is a word. You must use STACK operations to perform the task. [Hint ASCII for space is 20h].

CLO # 2: Create basic assembly code using different type of addressing modes in x86 & RISC ISAs to solve simple-moderate problems

Q7: Using String Primitive instructions, write an assembly language program to find common characters from the following Arrays. The common characters are required to be stored in a new array and should be displayed. [15 Marks]

```
.data
Destination BYTE "abcdef",0
Source BYTE "cfge",0
Result BYTE 4 DUP(?)
```

CLO # 2: Create basic assembly code using different type of addressing modes in x86 & RISC ISAs to solve simple-moderate problems [5+5=10Marks]

Q8(a): Convert the following code into assembly language.

```
while N > 0
{
    if (N != 3) AND (N < A OR N > B)
        N = N - 2
    else
        N = N - 1
}
```

Handwritten: false
if (3 != 3)
1

Q8(b): In the following code sequence, show the value of AL (in hexadecimal format only) after each shift or rotate instruction has executed:

```
MOV AL,0D4H
SHL AL,3
MOV AL,0E4H
SAR AL,3
STC
MOV AL,0ABH
ROL AL,27
STC
MOV AL,10H
RCR AL,1
```

a. _____
b. _____
c. _____
d. _____

Handwritten: 15 (3 = 3) ✓

===== Good Luck =====