

**National University of Computer and Emerging
Sciences**

Lab Manual

Computer Organization and Assembly Language



Lab 04

Instructor	Rida Mahmood
Class	DS3
Sections	A1, A2
Semester	Fall 2022

Fast School of Computing

FAST-NU, Lahore, Pakistan

Objectives

- How to interpret the different types of jumps
- How to use the different types of registers and how to manipulate them in assembly language
- How to perform arithmetic operations with registers and conditional jumps
- How to use the debugger for viewing the available registers and their function

Contents

Objectives	2
ACTIVITY 1	2
ACTIVITY 2	2
ACTIVITY 3	3
ACTIVITY 4	3
ACTIVITY 5	3
REFERENCES	3

Note for all questions: You can make as many memory variables as you need

ACTIVITY 1:

Initialize *AX* with last 4 digits of your roll number (for example, if your roll number is 16L-1105 then *AX* should be initialized with 1105).

Once initialized, write a program to swap every pair of bits in the *AX* register as shown in

Table

below:

AX	Contents of AX (Your Roll #)			
Before	000 0	010 0	010 1	000 1
After	000 0	100 0	101 0	001 0

ACTIVITY 2:

Modify your program in Activity 1 to swap two bits as shown in **Table** below:

AX	Contents of AX (Your Roll #)			
Before	000 0	010 0	010 1	000 1

After	000	000	010	010
	0	1	1	0

ACTIVITY 3

Modify your program in Activity 1 & 2 to swap two nibbles as shown in **Table** below:

AX	Contents of AX (Your Roll #)			
Before	000	010	010	0001
	0	0	1	
After	010	000	000	0101
	0	0	1	

ACTIVITY 4:

Initialize **AX** with last 4 digits of your roll number (for example, if your roll number is 16L-1105 then **AX** should be initialized with 1105). Store **AX** in **BX**. Make a memory variable *f*, initialize it with 0 and compute

$$f = (A || B) \& \& (A \odot 0x1BCD)$$

$||$ is bitwise OR operation, $\&\&$ is bitwise AND operation whereas \odot is bitwise XOR operation.

ACTIVITY 5:

Initialize **AX** with last 4 digits of your roll number (for example, if your roll number is 16L-1105 then **AX** should be initialized with 1105). Store **AX** in **BX**. Make a 32-bit memory variable *f*, initialize it with 0 and compute

$$f = (A \times B) + \{A, B\}$$

\times is **Multiplication** operation, $+$ is **Addition** operation whereas $\{A, B\}$ **concatenates**

16-bit **A** and

B to form **32-bit** number.

ACTIVITY 6:

Write a program to swap every pair of bits in the AX register

REFERENCES

- ["http://www.dosbox.com/download.php?main=1"](http://www.dosbox.com/download.php?main=1)
- <http://sourceforge.net/projects/nasm>

- <http://www.nasm.us/>
- <http://www.programmersheaven.com/download/21643/download.aspx> (AFD)

