National University of Computer and Emerging Sciences

Lab Manual

Computer Organization and Assembly Language

Lab 04

Instructor	Hazoor Ahmad
Class	CS3
Sections	A, D, H, K
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

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	0000	0100	0101	0001
After	0000	1000	1010	0010

ACTIVITY 2:

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

AX	Contents of AX (Your Roll #)			
Before	0000	0100	0101	0001
After	0000	0001	0101	0100

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	0000	0100	0101	0001
After	0100	0000	0001	0101

ACTIVITY 4:

Initialize $\boldsymbol{AX}$ with last 4 digits of your roll number (for example, if your roll number is 16L-1105 then $\boldsymbol{AX}$ should be initialized with 1105). Store $\overline{\boldsymbol{AX}}$ in $\boldsymbol{BX}$. 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 $\boldsymbol{AX}$ with last 4 digits of your roll number (for example, if your roll number is 16L-1105 then $\boldsymbol{AX}$ should be initialized with 1105). Store $\overline{\boldsymbol{AX}}$ in $\boldsymbol{BX}$. 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 $\{\boldsymbol{A}, \boldsymbol{B}\}$ **concatenates** 16-bit **A** and **B** to form **32-bit** number.

ACTIVITY 6:

Differentiate between Near, Far and Short Jumps. Write your own assembly language programs and demonstrate how these jumps have been taken.

REFERENCES

- "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)