National University of Computer and Emerging Sciences

Lab Manual

Computer Organization and Assembly Language

Lab 04

Instructors	Rida Mahmood, Amna Sehar
Class	CS3
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

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

1- Calculate the number of one bits in BX and complement an equal number of least significant bits in AX. HINT: Use the XOR instruction and Rotate through carry

Sample Run:

Initial value of BX	Total No of 1 Bits in BX	Initial value of AX	AX after Complementing 7 least significant bits
1011 0001 1000 1001	7	1010 1011 1**010 0101**	1010 1 1**101 1010**

[Extended Operations]

2- Write a program that shifts a 64-bit number.

3- Write a program that adds two 64-bit numbers.

4- Write a program to multiply two 32-bit numbers and store the answer in a 64-bit location.

Sample Run:

```
        a:      dq 0xABCDD4E1     ; dq allocates 64 bit memory space. a is 32-bit number but
                                  it has space allocation of 64 bits
        b:      dd 0xAB5C32       ; 32-bit space for multiplier
result: dq 0x0                    ; result should be 0x73005CB8FF6FF2 verify on calculator
programmer's view
```

[Homework]

5- Write a program to swap the nibbles (4-bits = 1 nibble) in each byte of the AX register.

Sample Run:

AX before Swap	**1011 0010 0101 1101**	**0xB25D**
AX after Swap	**0010 1011 1101 0101**	**0x2BD5**

6- Write a program to swap every pair of bits in the AX register i.e. swap bit # 0 with bit # 1, bit # 2 with bit # 3 and so on.

Sample Run:

AX before Swap	**10 11 00 10 01 01 11 01**
AX after Swap	**01 11 00 01 10 10 11 10**