National University of Computer and Emerging Sciences

Laboratory Manual

for

Computer Organization and Assembly Language Programming

Course Instructor	Ms. Sana Fatima
Lab Instructor(s)	Mr. Hazoor Ahmad Mr. Muhammad Salman Mubarik
Section	BSR-3A
Semester	Fall 2023

Exercise 1: [Bit Manipulation] Calculate the number of one bits in BX and complement an equal number of least significant bits in AX using MASK. HINT: Use the XOR instruction.

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 1011 1**101 1010**

Exercise 2: Write a subroutine to take a difference of two arrays and store result in the same array.

Exercise 3: Initialize AX with last 4 digits of your roll number (for example, if your roll number is 16L-1105 then AXshould 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 ⊙ is bitwise XOR operation.

Exercise 4: Write a subroutine that takes address of an array of integers, size of array, direction of rotation of array and number of rotations as parameters, and rotates the array left or right the number of times specified. Type of array is 'word'. Do not use registers for passing arguments to the sub-routine.

1	Address of array	
2	Size of array	A positive integer
3	Direction of Rotation	0 or 1 0= rotate left 1=rotate right
4	Number of rotations	Any positive integer

For instance: arr1: 1, 2, 3, 4

After single shifting left: 2, 3, 4, 1

After single shifting right: 4, 1, 2, 3

After shifting left twice the original array: 3 ,4 ,1, 2

Exercise 5: Write a subroutine **multiply** that receive two unsigned 8-bit integers and **return** product of them in back the 16-bit result. Don't use mul command write you own multiply code.

Write a subroutine series that receive two arrays and size of array (assume both arrays are same size), then it compute $S = \sum_{k=0}^{n-1} a(k) * b(k)$, use your multiply subroutine to multiply. then return 'S' to your main function. . (Assume your 'S' never exceed from 16 bits and is a local variable).

Write the main program that pass two arrays and size to series subroutine then get the 'S' and save in CX.

Do not use registers for passing arguments to the sub-routine. You may use a register for returning value from a sub-routine.

Post Lab

Exercise 1: You need to perform bit by bit comparison of two words. If the two words are equal then dx =1 otherwise, dx =0.

Exercise 2: Declare a 32byte buffer containing random data. Consider for this problem that the bits in these 32 bytes are numbered from 0 to 255. Declare another byte that contains the starting bit number. Write a program to copy the byte starting at this starting bit number in the AX register. Be careful that the starting bit number may not be a multiple of 8 and therefore the bits of the desired byte will be split into two bytes.