**Bahria University, Lahore Campus**

Department of Computer Sciences

Lab Journal 09

**(Spring 2023)**

|  |  |  |
| --- | --- | --- |
| Course: | **Computer Architecture & Organization Lab** |  |
| Course Code: | CEL 221 | Max Marks: 20 |
| Faculty’s Name: | Maryam Munawar | Lab Engineer: |

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Enroll No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Lab Tasks:

### Task1: 5 Marks

Write a program to retrieve two numbers from a user, and swap those number using only the XOR operation. You should not use a temporary variable to store the numbers while swapping them. Your program should include a proper and useful prompt for input, and print the results in a meaningful manner.

# Purpose: Implement a swap using only xor operations and no extra memory.

.text

#prompt and read first number

li $v0, 4

la $a0, prompt1

syscall

li $v0, 5

syscall

move $s0, $v0

#prompt and read second number

li $v0, 4

la $a0, prompt2

syscall

li $v0, 5

syscall

move $s1, $v0

xor $s0, $s0, $s1

xor $s1, $s0, $s1

xor $s0, $s0, $s1

# output first number

li $v0, 4

la $a0, output1

syscall

li $v0, 1

move $a0, $s0

syscall

# output second number

li $v0, 4

la $a0, output2

syscall

li $v0, 1

move $a0, $s1

syscall

li $v0, 10

syscall

.data

prompt1: .asciiz "\nEnter the first number: "

prompt2: .asciiz "\nEnter the second number: "

output1: .asciiz "\nThe first number is now "

output2: .asciiz "\nThe second number is now

**Task 2: 5 Marks**

Correct the following programs.

**Program 1**

.text

main:

li $v0, 4

la $a0, result1

syscall

li $v0, 1

li $a0, 4

syscall

li $v0, 4

la $a0, result2

syscall

li $v0, 1

li $a0, 8

syscall

addi $v0, $zero, 10 #Exit program

syscall

.data

result1: .ascii "\nfirst value = "

**Program 2**

.text

main:

li $v0, 4

la $a0, result1

syscall

li $v0, 4

li $a0, 4

syscall

li $v0, 4

la $a0, result2

syscall

li $v0, 1

li $a0, 8

syscall

addi $v0, $zero, 10 #Exit program

syscall

.data

result1: .asciiz "\nfirst value = "

result2: .asciiz "\nsecond value = "

**Task 3: 10 Marks**

There are many algorithms presented in this text that would lend themselves to be included as subprograms in the utils.asm file. Implement some or all of the following into the utils.asm file, properly documenting them, and include programs to test them.

* 1. NOR subprogram - take two input parameters, and return the NOR operation on those two parameter.
  2. NAND- take two input parameters, and return the NAND operation on those two parameter.
  3. NOT- take one input parameters, and return the NOT operation on that parameter.
  4. Mult4 - take an input parameter, and return that parameter multiplied by 4 using only shift and add operations.
  5. Mult10 - take an input parameter, and return that parameter multiplied by 10 using only shift and add operations.

**Lab Grading Sheet :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Max Marks** | **Obtained Marks** | **Comments(*if any*)** |
| a. | 5 |  |  |
| b. | 5 |  |  |
| c. | 10 |  |  |
|  |  |  |  |
| **Total** | **20** |  | **Signature** |

**Note : Attempt all tasks and get them checked by your Instructor**