**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: AFFAN AHMAD \_\_\_ Enroll No: \_03-134221-003\_\_

## 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.

li $v0, 4

la $a0, prompt1

syscall

li $v0, 5

syscall

move $s0, $v0

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

li $v0, 4

la $a0, output1

syscall

li $v0, 1

move $a0, $s0

syscall

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

**EXPLANATION:**

IN first 6 line we taking A input in put from user , 7 TO 14 line we taking B input from user, 15 t0 8 line we use xor operator to swap A and B like A=A^B

B=A^B

A=A^B

,21 TO 26 line we print the first swap number (A=B),29 T0 34 line we print second output which is (B=A)and last two line program exit.

**My scenario is swapping four number taking from user** :

li $v0, 4

la $a0, prompt1

syscall

li $v0, 5

syscall

move $s0, $v0

li $v0, 4

la $a0, prompt2

syscall

li $v0, 5

syscall

move $s1, $v0

li $v0, 4

la $a0, prompt3

syscall

li $v0, 5

syscall

move $s2, $v0

li $v0, 4

la $a0, prompt4

syscall

li $v0, 5

syscall

move $s3, $v0

xor $s0, $s0, $s3

xor $s3, $s0, $s3

xor $s0, $s0, $s3

xor $s1, $s1, $s2

xor $s2, $s1, $s2

xor $s1, $s1, $s2

li $v0, 4

la $a0, output1

syscall

li $v0, 1

move $a0, $s0

syscall

li $v0, 4

la $a0, output2

syscall

li $v0, 1

move $a0, $s1

syscall

li $v0, 4

la $a0, output3

syscall

li $v0, 1

move $a0, $s2

syscall

li $v0, 4

la $a0, output4

syscall

li $v0, 1

move $a0, $s3

syscall

li $v0, 10

syscall

.data

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

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

prompt3: .asciiz "\nEnter the third number: "

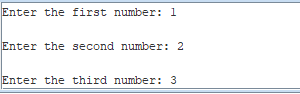
prompt4: .asciiz "\nEnter the fourth number: "

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

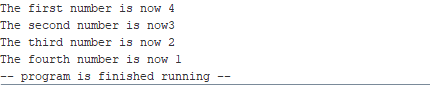
output2: .asciiz "\nThe second number is now"

output3: .asciiz "\nThe third number is now "

output4: .asciiz "\nThe fourth 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 = "

**error free :**

**.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**

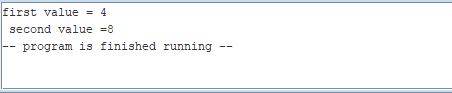
**addi $v0, $zero, 10 #Exit program**

**syscall**

**.data**

**result1: .asciiz "\nfirst value = "**

**result2:.asciiz"\n second 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 = "

**error free:**

**.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: .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.

.data

x:.asciiz"enter first number :"

y:.asciiz"enter second number :"

result:.asciiz"the output of nor operation is :"

.text

main:

li $v0 4

la $a0,x

syscall

li $v0, 5

syscall

move $t0 $v0

li $v0 4

la $a0,y

syscall

li $v0, 5

syscall

move $t1 $v0

nor $t2,$t0,$t1

li $v0 4

la $a0,result

syscall

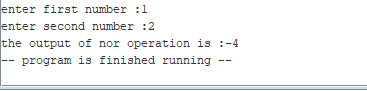
li $v0 ,1

move $a0,$t2

syscall

li $v0,10

syscall



* 1. NAND- take two input parameters, and return the NAND operation on those two parameter.

.data

x:.asciiz"enter first number :"

y:.asciiz"enter second number :"

result:.asciiz"the output of nor operation is :"

.text

main:

li $v0 4

la $a0,x

syscall

li $v0, 5

syscall

move $t0 $v0

li $v0 4

la $a0,y

syscall

li $v0, 5

syscall

move $t1 $v0

and $t2,$t0,$t1

not $t3, $t2

li $v0 4

la $a0,result

syscall

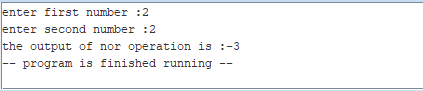
li $v0 ,1

move $a0,$t3

syscall

li $v0,10

syscall



* 1. NOT- take one input parameters, and return the NOT operation on that parameter.

.data

x:.asciiz"enter first number :"

y:.asciiz"enter second number :"

result:.asciiz"the output of not operation is :"

.text

main:

li $v0 4

la $a0,x

syscall

li $v0, 5

syscall

move $t0 $v0

not $t3,$t0

li $v0 4

la $a0,result

syscall

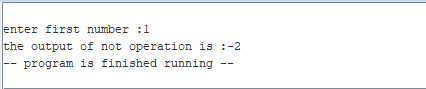
li $v0 ,1

move $a0,$t3

syscall

li $v0,10

syscall



* 1. Mult4 - take an input parameter, and return that parameter multiplied by 4 using only shift and add operations.

.data

x:.asciiz"enter first number :"

result:.asciiz"the output of nor operation is :"

.text

main:

li $v0 4

la $a0,x

syscall

li $v0, 5

syscall

move $t0 $v0

sll $t1,$t0,4

li $v0 4

la $a0,result

syscall

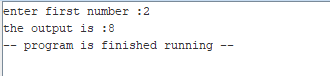
li $v0 ,1

move $a0,$t1

syscall

li $v0,10

syscall



* 1. Mult10 - take an input parameter, and return that parameter multiplied by 10 using only shift and add operations.

.data

x:.asciiz"enter first number :"

result:.asciiz"the output is :"

.text

main:

addi $t2,$zero,10

li $v0 4

la $a0,x

syscall

li $v0, 5

syscall

move $t0 $v0

sll $t1,$t0,1

mult $t0,$t2

mflo $t3

li $v0 4

la $a0,result

syscall

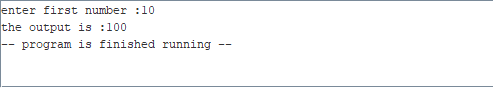
li $v0,1

move $a0,$t3

syscall

li $v0,10

syscall



**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**