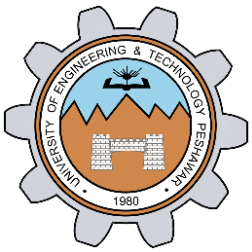
ARITHMETIC OPERATIONS IN QTSPIM (ASSEMBPLY LANGUAGE)

LAB # 01



Fall 2023

CSE-304L

Computer Organization & Architecture Lab

Submitted by: **AIMAL KHAN**

Registration No.: **21PWCSE1996**

Class Section: **A**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: 

Submitted to:

Dr. Bilal Habib

Thursday, October 5, 2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

ASSESSMENT RUBRICS COA LABS

LAB REPORT ASSESSMENT				
Criteria	**Excellent**	**Average**	**Nill**	**Marks Obtained**
1. Objectives of Lab	All objectives of lab are properly covered [Marks 10]	Objectives of lab are partially covered [Marks 5]	Objectives of lab are not shown [Marks 0]	
2. MIPS instructions with Comments and proper indentations.	All the instructions are well written with comments explaining the code and properly indented [Marks 20]	Some instructions are missing are poorly commented code [Marks 10]	The instructions are not properly written [Marks 0]	
3. Simulation run without error and warnings	The code is running in the simulator without any error and warnings [Marks 10]	The code is running but with some warnings or errors. [Marks 5]	The code is written but not running due to errors [Marks 0]	
4. Procedure	All the instructions are written with proper procedure [Marks 20]	Some steps are missing [Marks 10]	steps are totally missing [Marks 0]	
5. OUTPUT	Proper output of the code written in assembly [Marks 20]	Some of the outputs are missing [Marks 10]	No or wrong output [Marks 0]	
6. Conclusion	Conclusion about the lab is shown and written [Marks 20]	Conclusion about the lab is partially shown [Marks 10]	Conclusion about the lab is not shown[Marks0]	
7. Cheating			Any kind of cheating will lead to 0 Marks	
Total Marks Obtained:_____ Instructor Signature: _____				

Arithmetic Operation in Qtspim

(Assembly Language)

Objectives:

- Introduction to assembly language and Qtspim software
- How to write arithmetic operations
- How to write logical operations

Tasks:

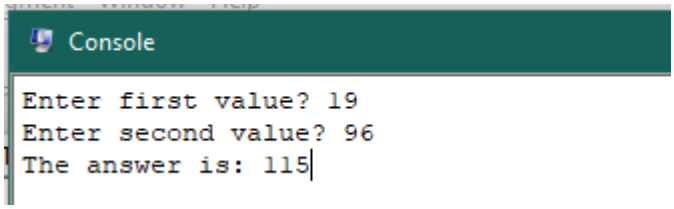
Task 1: Write an assembly language program which takes two numbers from user and add them and show the result on console.

Code:

```
.text
.globl main

main:
      li $v0, 4       # syscall 4 (print_str)
        la $a0, prompt1     # argument: string
        syscall         # print the string

      li $v0,5
      syscall
      move $t0,$v0

      li $v0,4
      la $a0, prompt2
      syscall

      li $v0,5
      syscall
      move $t1,$v0

      li $v0,4
      la $a0, answer
      syscall

      add $t2, $t0,$t1
      li $v0, 1
      move $a0, $t2
      syscall


.data
      prompt1:   .asciiz "Enter first value? "
```

```
      prompt2:   .asciiz "Enter second value? "
      answer:   .asciiz "The answer is: "
```

Output:

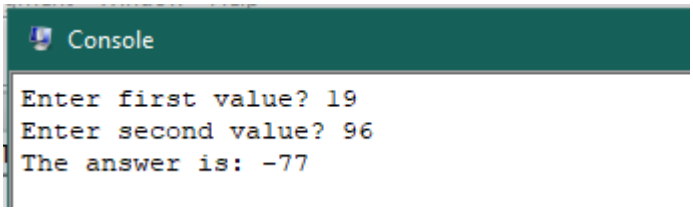


Task 2: Write an assembly language program which takes two numbers from user and subtract them and show the result on console

Code:

```
.text
.globl main

main:
      li $v0, 4       # syscall 4 (print_str)
      la $a0, prompt1     # argument: string
      syscall         # print the string

      li $v0,5
      syscall
      move $t0,$v0

      li $v0,4
      la $a0, prompt2
      syscall

      li $v0,5
      syscall
      move $t1,$v0

      li $v0,4
      la $a0, answer
      syscall

      sub $t2, $t0,$t1
      li $v0, 1
      move $a0, $t2
      syscall


.data
      prompt1:   .asciiz "Enter first value? "
      prompt2:   .asciiz "Enter second value? "
      answer:   .asciiz "The answer is: "
```

Output:

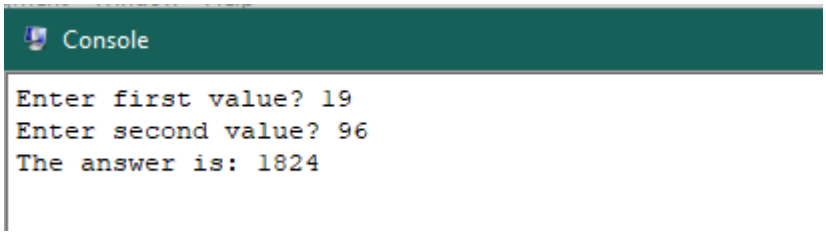


Task 3: Write an assembly language program which takes two numbers from user and multiply them and show the result on console.

Code:

```
.text
.globl main

main:
      li $v0, 4       # syscall 4 (print_str)
        la $a0, prompt1     # argument: string
        syscall         # print the string

      li $v0,5
      syscall
      move $t0,$v0

      li $v0,4
      la $a0, prompt2
      syscall

      li $v0,5
      syscall
      move $t1,$v0

      li $v0,4
      la $a0, answer
      syscall

      mul $t2, $t0,$t1
      li $v0, 1
      move $a0, $t2
      syscall


.data
      prompt1:   .asciiz "Enter first value? "
      prompt2:   .asciiz "Enter second value? "
      answer:   .asciiz "The answer is: "
```

Output:



Task 4: Write an assembly language program which takes two numbers from user and divide them and show the result on console.

Code:

```
.text
.globl main

main:
      li $v0, 4       # syscall 4 (print_str)
        la $a0, prompt1     # argument: string
        syscall         # print the string

      li $v0,5
      syscall
      move $t0,$v0

      li $v0,4
      la $a0, prompt2
      syscall

      li $v0,5
      syscall
      move $t1,$v0

      li $v0,4
      la $a0, answer
      syscall

      div $t2, $t0,$t1
      li $v0, 1
      move $a0, $t2
      syscall

.data
      prompt1:   .asciiz "Enter first value? "
      prompt2:   .asciiz "Enter second value? "
      answer:   .asciiz "The answer is: "
```

Output:

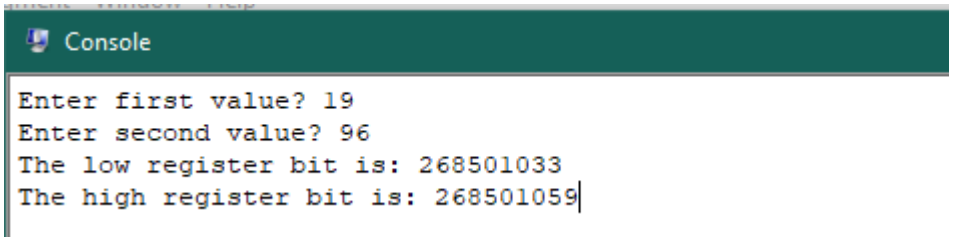
Task 5: Write assembly program to multiply two numbers using MULT and extract the bit from high and low registers to general purpose registers.

Code:

```
.text
.globl main

main:
      li $v0, 4       # syscall 4 (print_str)
        la $a0, prompt1     # argument: string
        syscall         # print the string

      li $v0,5
      syscall
      move $t0,$v0

      li $v0,4
      la $a0, prompt2
      syscall

      li $v0,5
      syscall
      move $t1,$v0

      li $v0,4
      la $a0, low
      syscall

      mult $t0,$t1
      li $v0, 1
      mfhi $t3
      syscall

      li $v0,4
      la $a0, high
      syscall

      mult $t0,$t1
      li $v0, 1
      mflo $t4
      syscall


.data
```

```
      prompt1:   .asciiz "Enter first value? "
      prompt2:   .asciiz "Enter second value? "
      low:   .asciiz "The low register bit is: "
      high:   .asciiz "\nThe high register bit is: "
```

Output:



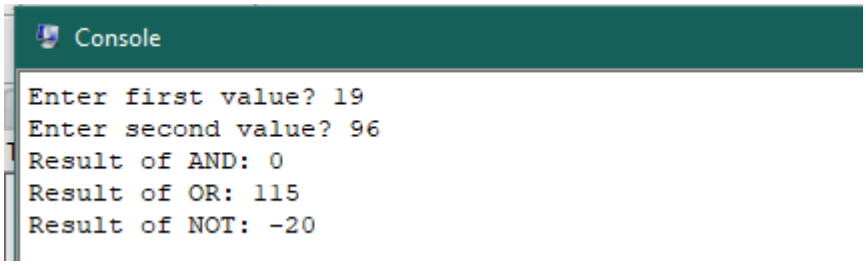
Task 6: Write program to perform AND, OR, NOT operations in MIPS.

Code:

```
.text
.globl main

main:
      li $v0, 4
        la $a0, prompt1
        syscall

      li $v0,5
      syscall
      move $t0,$v0

      li $v0,4
      la $a0, prompt2
      syscall

      li $v0,5
      syscall
      move $t1,$v0

      li $v0,4
      la $a0, AND
      syscall

      and $t2,$t0,$t1
      li $v0, 1
      move $a0, $t2
      syscall

      li $v0,4
      la $a0, OR
      syscall

      or $t3,$t0,$t1
```

```
      li $v0, 1
      move $a0, $t3
      syscall

      li $v0,4
      la $a0, NOT
      syscall

      not $t4,$t0
      li $v0, 1
      move $a0, $t4
      syscall


.data
      prompt1:   .asciiz "Enter first value? "
      prompt2:   .asciiz "Enter second value? "
      AND:   .asciiz "Result of AND: "
      OR:   .asciiz "\nResult of OR: "
      NOT:   .asciiz "\nResult of NOT: "
```

Output:



Reference:

To view my codes, please refer to my GitHub Account.

Conclusion:

In this lab I have learnt the basics of assembly language and how to perform arithmetic and logical operation on values taken from user and display the result on console.

The End.