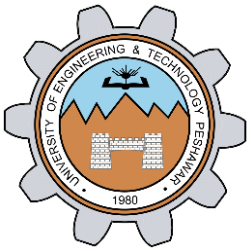
COA Lab

Lab #05



Fall 2021

CSE304L Computer Organization and Architecture Lab

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

Class Section: **B**

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

Student Signature: _____

Submitted to:

Engr. Amaad Khalil

January 2, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Task 1:

Write a program for MIPS architecture that contain all assembly instructions for R type I type and J type instructions.

Source code:

```
.data
        str: .asciiz"Enter first number: "
        str1: .asciiz"Enter second number: "
        str2: .asciiz"And: "
        str3: .asciiz"\nOr: "
        str4: .asciiz"\nXor: "
        str5: .asciiz"\nNor: "
        str6: .asciiz"\nAndi: "
        str7: .asciiz"\nOri: "
        str8: .asciiz"\nXori: "
        str9: .asciiz"\nShift left: "
        str10: .asciiz"\nShift Right: "
        str11: .asciiz"\nShift right arithm: "
        str12: .asciiz"\nShift left by var: "
        str13: .asciiz"\nShift right by var: "
        str14: .asciiz"\nShift right arithm. by var: "
        str15: .asciiz"\nAdd: "
        str16: .asciiz"\nAddu: "
        str17: .asciiz"\nSub: "
        str18: .asciiz"\nSubu: "
        str19: .asciiz"\nDiv: "
        str20: .asciiz"\nDivu: "
        str21: .asciiz"\nMult: "
        str22: .asciiz"\nMultu: "
        str23: .asciiz"\nMul: "
        str24: .asciiz"\nHi: "
        str25: .asciiz"\nLo: "
        str26: .asciiz"t0 is less than t1\n"
        str27: .asciiz"t0 is less than or equal to t1\n"
        str28: .asciiz"\nt0 is greater than t1\n"
        str29: .asciiz"t0 is greater than or equal to t1\n"
.text


main:
        li $v0,4      #system call code for printing string
        la $a0,str    #address of a string to print
        syscall       #print the string
        li $v0,5      #take the value from user
        syscall
        move $t0,$v0  #move the value to t0

        li $v0,4      #system call code for printing string
```

```
la $a0,str1   #address of a string to print
syscall       #print the string
li $v0,5      #take the value from user
syscall
move $t1,$v0  #move the value to t1

li $v0,4      #system call code for printing string
la $a0,str2   #address of a string to print
syscall       #print the string

and $t2,$t0,$t1 #and of t0 and t1 stored in t2
move $a0,$t2    #move t2 to a0
li $v0,1        #output the value
syscall

li $v0,4      #system call code for printing string
la $a0,str3   #address of a string to print
syscall       #print the string

or $t2,$t0,$t1  #or of t0 and t1 stored in t2
move $a0,$t2    #move t2 to a0
li $v0,1        #output the value
syscall

li $v0,4     #system call code for printing string
la $a0,str4  #address of a string to print
syscall      #print the string

xor $t2,$t0,$t1  #xor of t0 and t1 stored in t2
move $a0,$t2     #move t2 to a0
li $v0,1         #output the value
syscall

li $v0,4     #system call code for printing string
la $a0,str5  #address of a string to print
syscall      #print the string

nor $t2,$t0,$t1  #nor of t0 and t1 stored in t2
move $a0,$t2     #move t2 to a0
li $v0,1         #output the value
syscall

li $v0,4     #system call code for printing string
la $a0,str6  #address of a string to print
syscall      #print the string

andi $t2,$t0,7  #andi of t0 and 7 stored in t2
move $a0,$t2    #move t2 to a0
li $v0,1        #output the value
syscall
```

```
li $v0,4      #system call code for printing string
la $a0,str7   #address of a string to print
syscall       #print the string

ori $t2,$t0,7  #ori of t0 and 7 stored in t2
move $a0,$t2   #move t2 to a0
li $v0,1       #output the value
syscall

li $v0,4      #system call code for printing string
la $a0,str8   #address of a string to print
syscall       #print the string

xori $t2,$t0,7  #xori of t0 and 7 stored in t2
move $a0,$t2    #move t2 to a0
li $v0,1        #output the value
syscall

li $v0,4      #system call code for printing string
la $a0,str9   #address of a string to print
syscall       #print the string

sll $t2,$t0,2  #sll of t0 and 2 stored in t2
move $a0,$t2   #move t2 to a0
li $v0,1       #output the value
syscall

li $v0,4       #system call code for printing string
la $a0,str10   #address of a string to print
syscall        #print the string

srl $t2,$t0,2  #srl of t0 and 2 stored in t2
move $a0,$t2   #move t2 to a0
li $v0,1       #output the value
syscall

li $v0,4       #system call code for printing string
la $a0,str11   #address of a string to print
syscall        #print the string

sra $t2,$t0,2  #sra of t0 and 2 stored in t2
move $a0,$t2   #move t2 to a0
li $v0,1       #output the value
syscall


li $v0,4       #system call code for printing string
la $a0,str12   #address of a string to print
syscall        #print the string

sllv $t2,$t0,$t1  #sllv of t0 and t1 stored in t2
```

```
move $a0,$t2      #move t2 to a0
li $v0,1          #output the value
syscall

li $v0,4       #system call code for printing string
la $a0,str13   #address of a string to print
syscall        #print the string

srlv $t2,$t0,$t1  #srlv of t0 and t1 stored in t2
move $a0,$t2      #move t2 to a0
li $v0,1          #output the value
syscall

li $v0,4       #system call code for printing string
la $a0,str14   #address of a string to print
syscall        #print the string

srav $t2,$t0,$t1  #srav of t0 and t1 stored in t2
move $a0,$t2      #move t2 to a0
li $v0,1          #output the value
syscall


li $v0,4      #system call code for printing string
la $a0,str15  #address of a string to print
syscall       #print the string

add $t2,$t0,$t1
move $a0,$t2      #move t2 to a0
li $v0,1          #output the value
syscall

li $v0,4      #system call code for printing string
la $a0,str16  #address of a string to print
syscall       #print the string

addu $t2,$t0,$t1
move $a0,$t2      #move t2 to a0
li $v0,1          #output the value
syscall

li $v0,4      #system call code for printing string
la $a0,str17  #address of a string to print
syscall       #print the string

sub $t2,$t0,$t1
move $a0,$t2      #move t2 to a0
li $v0,1          #output the value
syscall

li $v0,4      #system call code for printing string
```

```
la $a0,str18  #address of a string to print
syscall      #print the string

subu $t2,$t0,$t1
move $a0,$t2     #move t2 to a0
li $v0,1         #output the value
syscall

li $v0,4     #system call code for printing string
la $a0,str19  #address of a string to print
syscall      #print the string


div $t0,$t1
mflo $a0
li $v0,1         #output the value
syscall

li $v0,4     #system call code for printing string
la $a0,str20  #address of a string to print
syscall      #print the string


divu $t0,$t1
mflo $a0
li $v0,1         #output the value
syscall

li $v0,4     #system call code for printing string
la $a0,str21  #address of a string to print
syscall      #print the string

mult $t1,$t0
mflo $a0
li $v0,1         #output the value
syscall


li $v0,4     #system call code for printing string
la $a0,str22  #address of a string to print
syscall      #print the string

multu $t1,$t0
mflo $a0
li $v0,1         #output the value
syscall


li $v0,4     #system call code for printing string
la $a0,str23  #address of a string to print
syscall      #print the string
```

```
        mul $t2,$t1,$t0
        move $a0,$t2
        li $v0,1        #output the value
        syscall

        li $v0,4     #system call code for printing string
        la $a0,str24  #address of a string to print
        syscall      #print the string

        mthi $t0
        mfhi $a0
        li $v0,1        #output the value
        syscall

        li $v0,4     #system call code for printing string
        la $a0,str25  #address of a string to print
        syscall      #print the string

        mtlo $t1
        mflo $a0
        li $v0,1        #output the value
        syscall

        blt $t0,$t1,Lab1   #check if t0 is less than t1
        j test2                    #jump to test2

Lab1:
        li $v0,4   #system call code for printing string
        la $a0,str26        #address of the string to print
        syscall             #print the string
test2:
        ble $t0,$t1,Lab2  #check if t0 is less than or equal to t1
        j test3                    #jump to test3
Lab2:
        li $v0,4   #system call code for printing string
        la $a0,str27        #address of the string to print
        syscall             #print the string
test3:
        bgt $t0,$t1,Lab3  #check if t0 is greater than t1
        j test4                    #jump test4
Lab3:
        li $v0,4   #system call code for printing string
        la $a0,str28        #address of the string to print
        syscall             #print the string
test4:
        bge $t0,$t1,Lab4 #check if t0 is greater than or equal to t1
        j exit                     #jump to exit
Lab4:
        li $v0,4   #system call code for printing string
        la $a0,str29        #address of the string to print
```

```
        syscall             #print the string
exit:
        li $v0,10 #Terminate the program
        syscall             #syscall
```

Output:

```
Enter first number: 8
Enter second number: 2
And: 0
Or: 10
Xor: 10
Nor: -11
Andi: 0
Ori: 15
Xori: 15
Shift left: 32
Shift Right: 2
Shift right arithm: 2
Shift left by var: 32
Shift right by var: 2
Shift right arithm. by var: 2
Add: 10
Addu: 10
Sub: 6
Subu: 6
Div: 4
Divu: 4
Mult: 16
Multu: 16
Mul: 16
Hi: 8
Lo: 2
t0 is greater than t1
t0 is greater than or equal to t1
```