LAB REPORT

Computer organization & Assembly Language

Agenda:
Learn to write Lab report
Finding factorial of a number

Description:

In this lab I have made a code of factorial to run in PCSpim software. In this code I have used following registers
t1, t2, t3, a0, v0. registers t1,t2,t3 are used for arithmetic and logic operations while accumulator a0 is used for displaying the results and v0 is a special purpose register loaded with immediate values for different operations as for display it is loaded with 1 and for exit it is loaded with 10.
There are four tags (labels) are used in this code main, factorial, display and exit.
Commands used in this code are li(load immediate), move,addi(add immediate), mul(multiply), bne(branch not equal),syscall .

Logic:

At start I have loaded value 5 in register t1 and 1 in t3 then as a temporary register I have used register t2 and moving 5 in it. Then to make loop I have added a label of factorial in which the value in t1 is decremented and multiplying it with t2 and finally the result is displayed.

Code:

```
.data
.text
.globl main

main:
        li $t1,5        #li is used to load immediate value 5 in register t1 which will use
                        for loop

        li $t3,1        #1 is loaded in t3 for comparing with t1

        move $t2,$t1    #the value in t1 is overwrite in t2 by using move instruction

factorial:              #label or tag in this program it used for looping

        addi $t1,$t1,-1 #immediate addition instructions as there are not any subi
                        instruction so for decrement the value in t1 I have added -1 in t1
                        result is stored again in t1

        mul $t2,$t2,$t1#multiplying two registers and store the result in t2

        bne $t1,$t3,factorial#compare t1 and t3 is equal or not if not equal then go back
                                to factorial (tag)

display:                #label for display

        move $a0,$t2    #moving result (t2) into accumulator a0
```

```
        li $v0,1        #v0 is a special purpose register

        syscall         #system call for displaying result

Exit:
        li $v0,10
        syscall         #system call for exit
```

Result: