DHA Suffa University

Department of Computer Science

Computer Organization & Assembly Language

Fall 2017

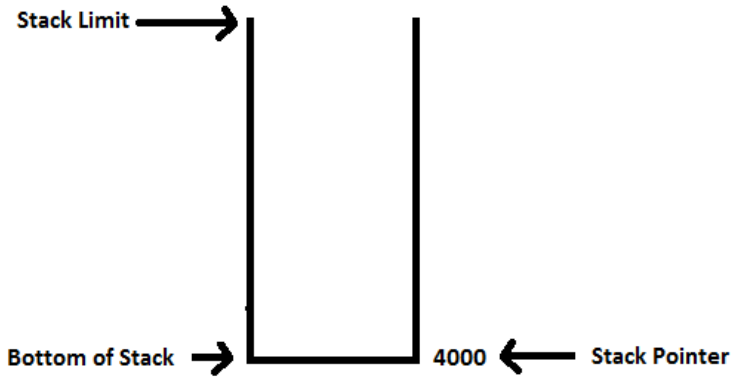
Lab # 10 (Recursive Procedures & Stack)

Objective:

To understand the recursive calls of procedures using Stack.

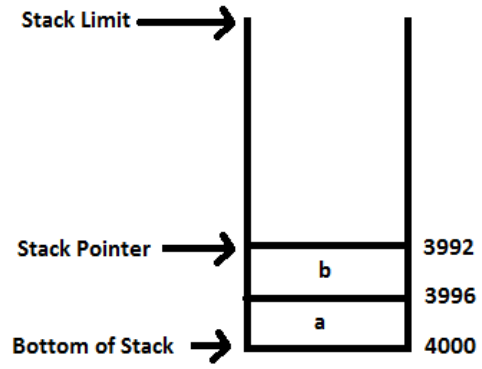
Use of Stack in MIPS:

While working with procedures in MIPS one should have clear concept of how stack works. Look at the figure below:



Initially when Stack is empty then Stack Pointer points at the Bottom of Stack as shown in the above figure. When something needs to be pushed on the Stack then it is needed to decrement the Stack Pointer. In MIPS $SP register contains the Stack Pointer value.

If a variable "a" is pushed on the Stack then Stack Pointer would be decreased by 4. If any other variable "b" is pushed on the Stack then Stack pointer would be further decreased by 4. After these two consecutive pushes Stack would loo k like below figure:



When it is needed to pop a value from stack then first increase the stack pointer by 4 and then pop that value and similarly do the same procedure to pop other values present in the stack.

MIPS Program for computing sum of natural numbers using Stack:

```
.text
.globl main
main:
        li $a0,7        #passing the argument to function addition(7);
        li $v1,0        # initialization of $v1

        jal addition    #function call

        move $a0,$v1    #moving the result to $a0 for printing
        li $v0,1
        syscall
        li $v0,10       #exit code
        syscall

addition:

        sub $sp,$sp,4           #moving $sp up
        sw $ra,($sp)            #storing $ra in stack

        beq $a0,0,Base          #checking base case

        add $v1,$v1,$a0
        sub $a0,$a0,1

        jal addition

        Base:                   #base case
                lw $ra,($sp)    #poping $ra from stack
                add $sp,$sp,4   #updating $sp

                jr $ra          #jumping to $ra
```