ASSIGNMENT 7 REPORT

DIWAKAR PRAJAPATI(2018CS10330)
SATWIK BANCHHOR (2018CS10385)

AIM:
To make a MIPS Simulator in C++.

DESIGN DETAILS:
In assignment 6 we had made Simulator for MIPS program in VHDL. In this we have made the Simulator in C++. The implementation is a bit different here.

IMPLEMENTATION:
Register array: Array of integer of size 32.
Memory array: Array of string of size 4096.
Label array: Array of string of size 20.
Label_Address array: Array of int of size 20.
The extra feature added is of labelling. We can make components by labelling the parts of code for easier understanding while branching, jumping and calling functions.

The memory is being initialise with program and rest as "[NULL]" (a symbol for 00000000000000000000000000000000)
While initialising the memory, if any label is encountered, then is is getting stored in the label array and corresponding address is stored in the the corresponding index in Label_address array(ultimately generating a mapping from label to address). Now when any jump of and branch instruction occurs, then it fetches the label and looks up in the array to get address of the label and then jumps to that label.
I have made a function which converts a register ($s0,$rt, $t0, $at , $ra) to its corresponding index of the register.

TEST CASE

```
//SUM TILL N
//reg[1]=1,reg[3]=N, output in reg[16].
label1 :
add $s0 $zero $zero
add $v0 $zero $zero
label2 :
add $v0 $v0 $at
add $s0 $s0 $v0
beq $v0 $v1 label3
j label2
label3 :
```

```
//STORE FIRST N FIBONACCI IN MEMORY
//reg[1]=0, reg[2]=1, reg[3]=N, output in memory from 4096 to
4096-N+1
label1 :
sub $sp $sp $at
sw $a0 0 $sp
sub $sp $sp $at
sw $a1 0 $sp
add $v0 $v0 $at
label2 :
add $a2 $a1 $a0
sub $sp $sp $at
sw $a2 0 $sp
lw $a0 1 $sp
lw $a1 0 $sp
add $v0 $v0 $at
beq $v0 $v1 label3
j label2
label3 :
```