**TASK # 1:**

Clear the sign bit of $t0 while leaving the other bits unchanged?

SOURCE CODE:

.data

input: .asciiz "\n enter an integer value: "

result: .asciiz "\n result is: "

.text

.globl main

main:

li $t0,0x00000000

la $a0,input

li $v0,4

syscall

li $v0,5

syscall

move $t1,$v0

and $t2,$t1,$t0

la $a0,result

li $v0,4

syscall

move $a0,$t2

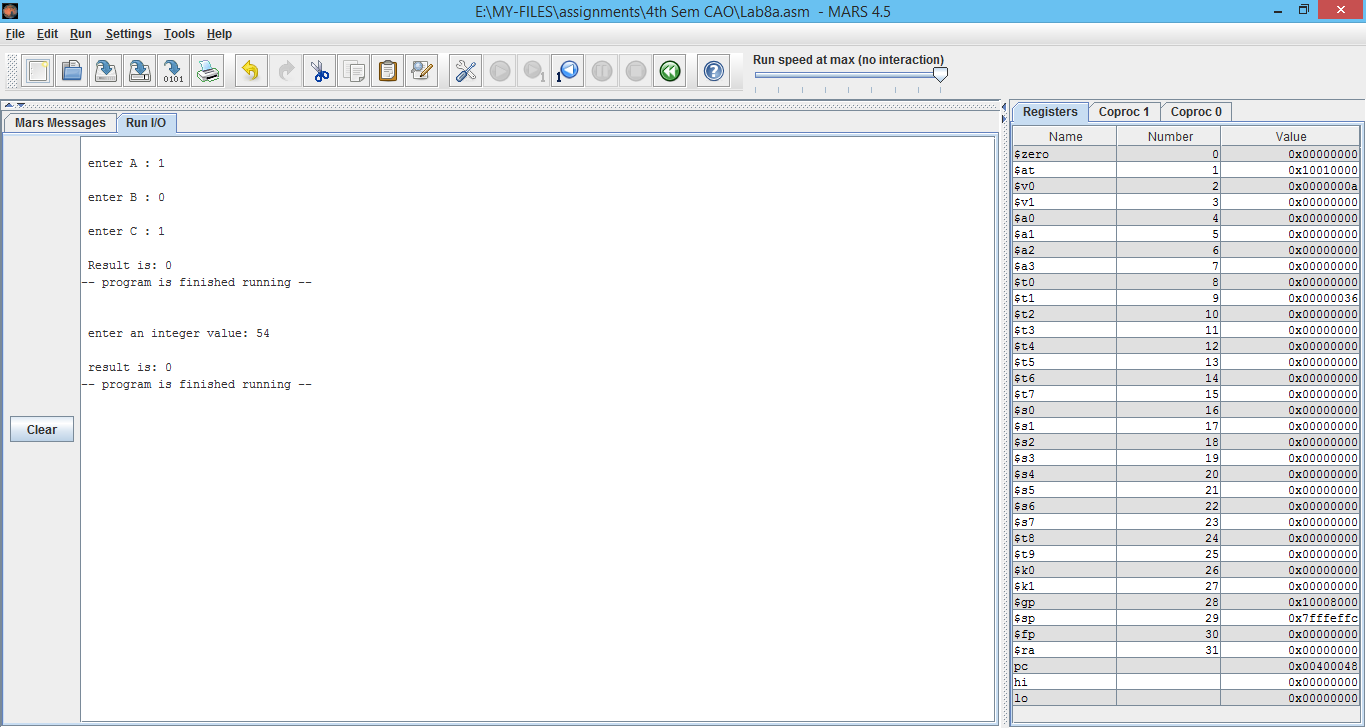
li $v0,1

syscall

li $v0,10

syscall

OUTPUT:



**TASK # 2:**

Set the most significant and least significant bits of $t0 while preserving the other bits?

SOURCE CODE:

.data

input: .asciiz "\n enter an integer value: "

result: .asciiz "\n result is: "

.text

.globl main

main:

li $t0,0x01111110

la $a0,input

li $v0,4

syscall

li $v0,5

syscall

move $t1,$v0

and $t2,$t1,$t0

la $a0,result

li $v0,4

syscall

move $a0,$t2

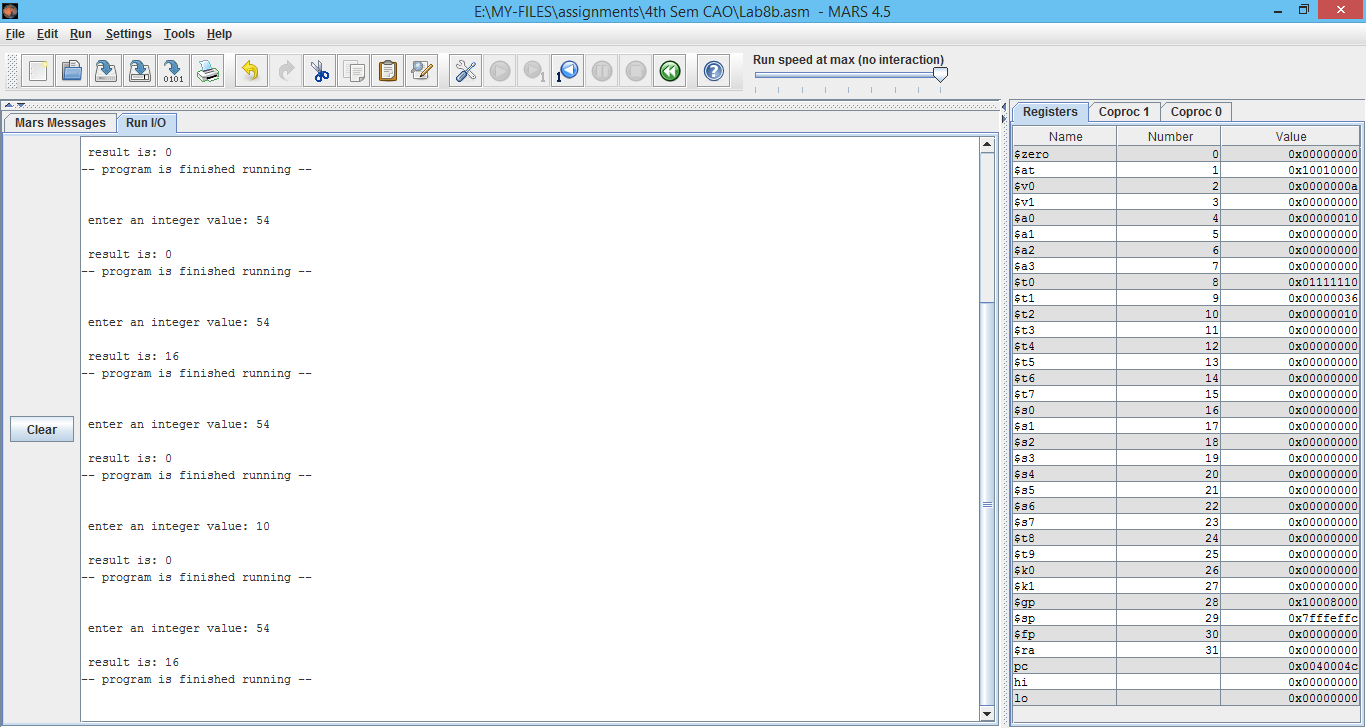
li $v0,1

syscall

li $v0,10

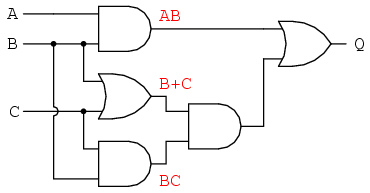
syscall

OUTPUT:



**TASK # 3:**

Implement following circuit into MIPS Assembly Language using AND,OR instructions.

****

SOURCE CODE:

.data

A: .asciiz "\n enter A : "

B: .asciiz "\n enter B : "

C: .asciiz "\n enter C : "

Result: .asciiz "\n Result is: "

.text

.globl main

main:

la $a0,A

li $v0,4

syscall

li $v0,5

syscall

move $t1,$v0

la $a0,B

li $v0,4

syscall

li $v0,5

syscall

move $t2,$v0

la $a0,C

li $v0,4

syscall

li $v0,5

syscall

move $t3,$v0

and $t4,$t1,$t2

or $t5,$t2,$t3

and $t6,$t2,$t3

and $t7,$t5,$t6

and $t8,$t4,$t7

la $a0,Result

li $v0,4

syscall

move $a0,$t8

li $v0,1

syscall

li $v0,10

syscall

OUTPUT:

