Practical 3

**Aim**: Working with MIPS assembly language using Mars simulator.

1. **.** Write an ALP to perform 8-bit and 16 bit multiplication.

# Program/Procedure (8 bit multiplication):

.data

msg1: .asciiz "Enter the First Number:" msg2: .asciiz "Enter the Second Number:" msg3: .asciiz "Multiplication is:"

.text

#getting first number:- la $a0, msg1

li $v0, 4 syscall

li $v0, 5 syscall

move $t0, $v0

#getting second number:- la $a0, msg2

li $v0, 4 syscall

li $v0, 5

syscall

move $t1, $v0

#Multiplication of two numbers:- la $a0, msg3

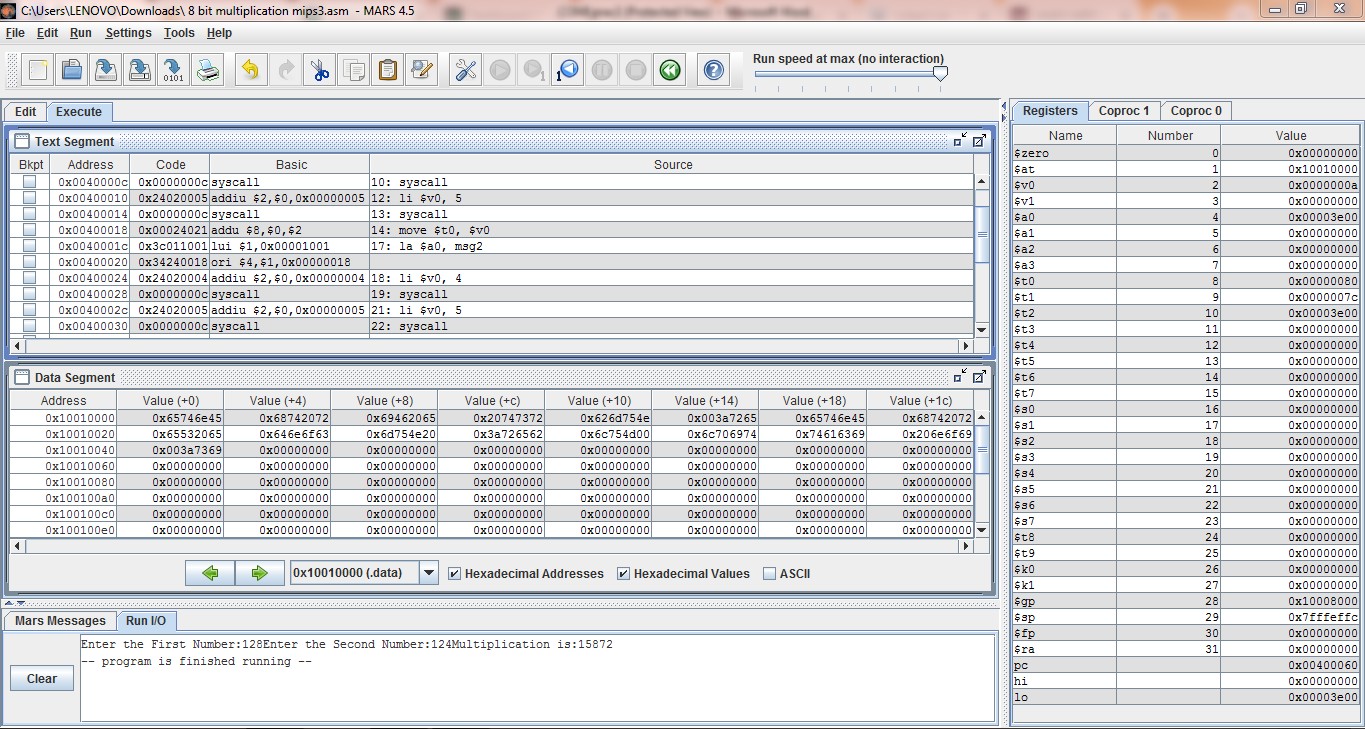
li $v0, 4 syscall

mul $t2,$t0,$t1

move $a0,$t2 li $v0,1 syscall

# Results:

#end program:- li $v0, 10 syscall



# Program/Procedure (16 bit multiplication):

.data

msg1: .asciiz "Enter the First Number:" msg2: .asciiz "Enter the Second Number:" msg3: .asciiz "Multiplication is:"

.text

#getting first number:- la $a0, msg1

li $v0, 4 syscall

li $v0, 5 syscall

move $t0, $v0

#getting second number:- la $a0, msg2

li $v0, 4 syscall

li $v0, 5 syscall

move $t1, $v0

#Multiplication of two numbers:- la $a0, msg3

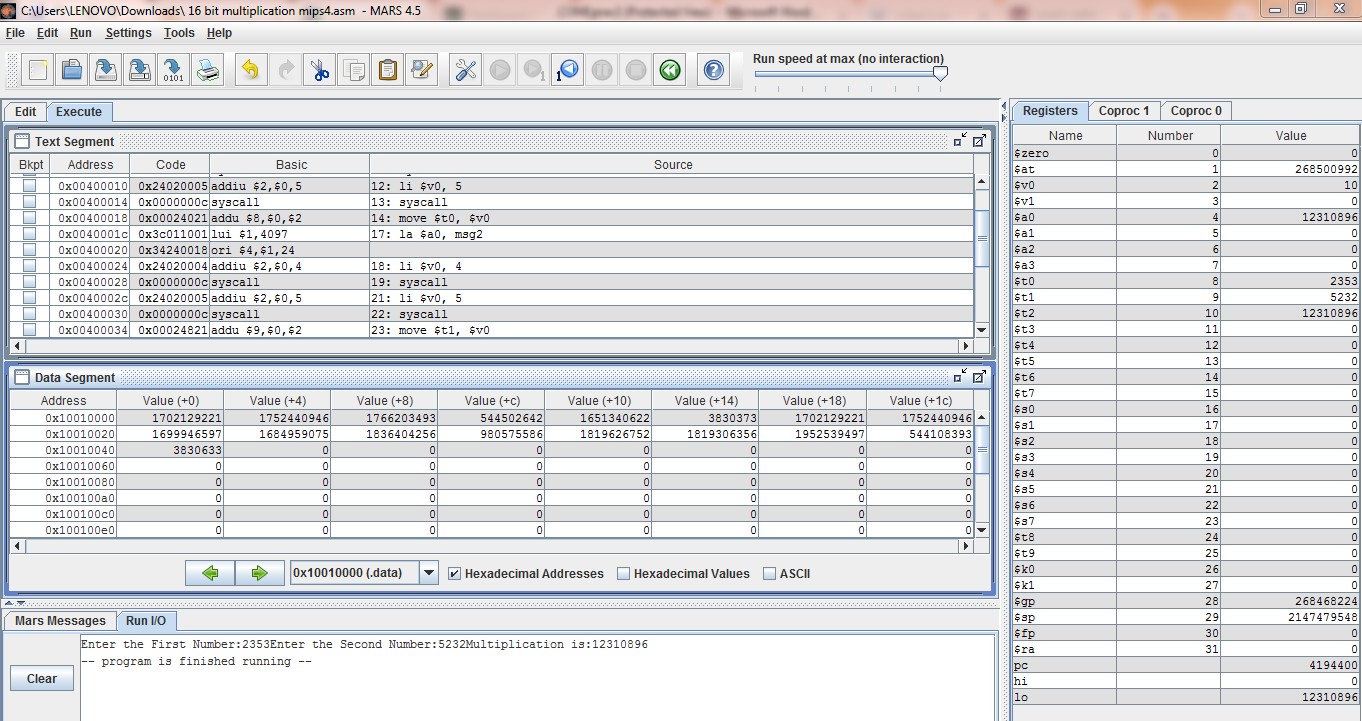
li $v0, 4 syscall

mul $t2,$t0,$t1

move $a0,$t2 li $v0,1 syscall

#end program:- li $v0, 10 syscall

# Results:



1. Write an ALP to perform 8-bit and 16-bit division.

# Program/Procedure (8 bit Division):

.data

msg1: .asciiz "Enter the First Number:" msg2: .asciiz "Enter the Second Number:" msg3: .asciiz "Division is:"

.text

#getting first number:- la $a0, msg1

li $v0, 4 syscall

li $v0, 5

syscall

move $t0, $v0

#getting second number:- la $a0, msg2

li $v0, 4 syscall

li $v0, 5 syscall

move $t1, $v0

#Division of two numbers:- la $a0, msg3

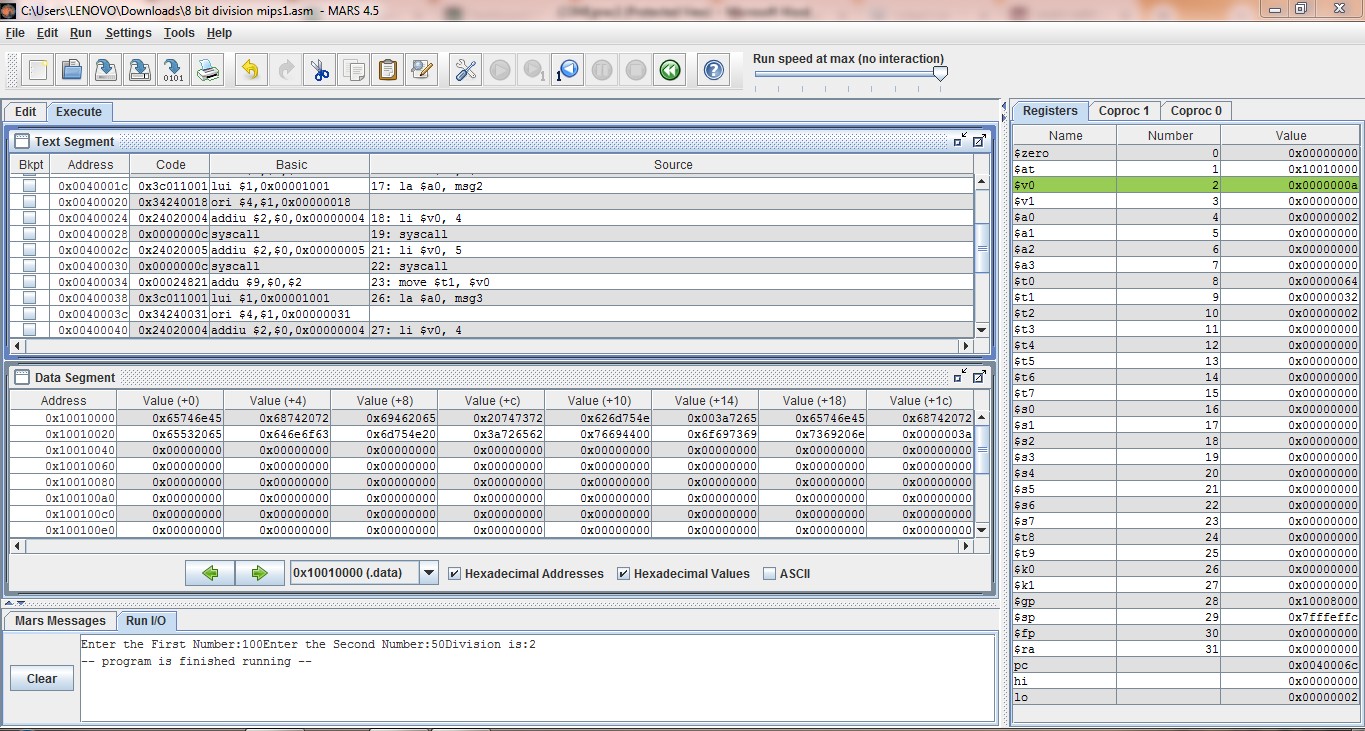
li $v0, 4 syscall

div $t2,$t0,$t1

move $a0,$t2 li $v0,1 syscall

#end program:- li $v0, 10 syscall

# Results:



* + **Program/Procedure (16 bit Division):**

.data

msg1: .asciiz "Enter the First Number:" msg2: .asciiz "Enter the Second Number:" msg3: .asciiz "Division is:"

.text

#getting first number:- la $a0, msg1

li $v0, 4 syscall li $v0, 5 syscall

move $t0, $v0

#getting second number:- la $a0, msg2

li $v0, 4 syscall li $v0, 5 syscall

move $t1, $v0

#Division of two numbers:- la $a0, msg3

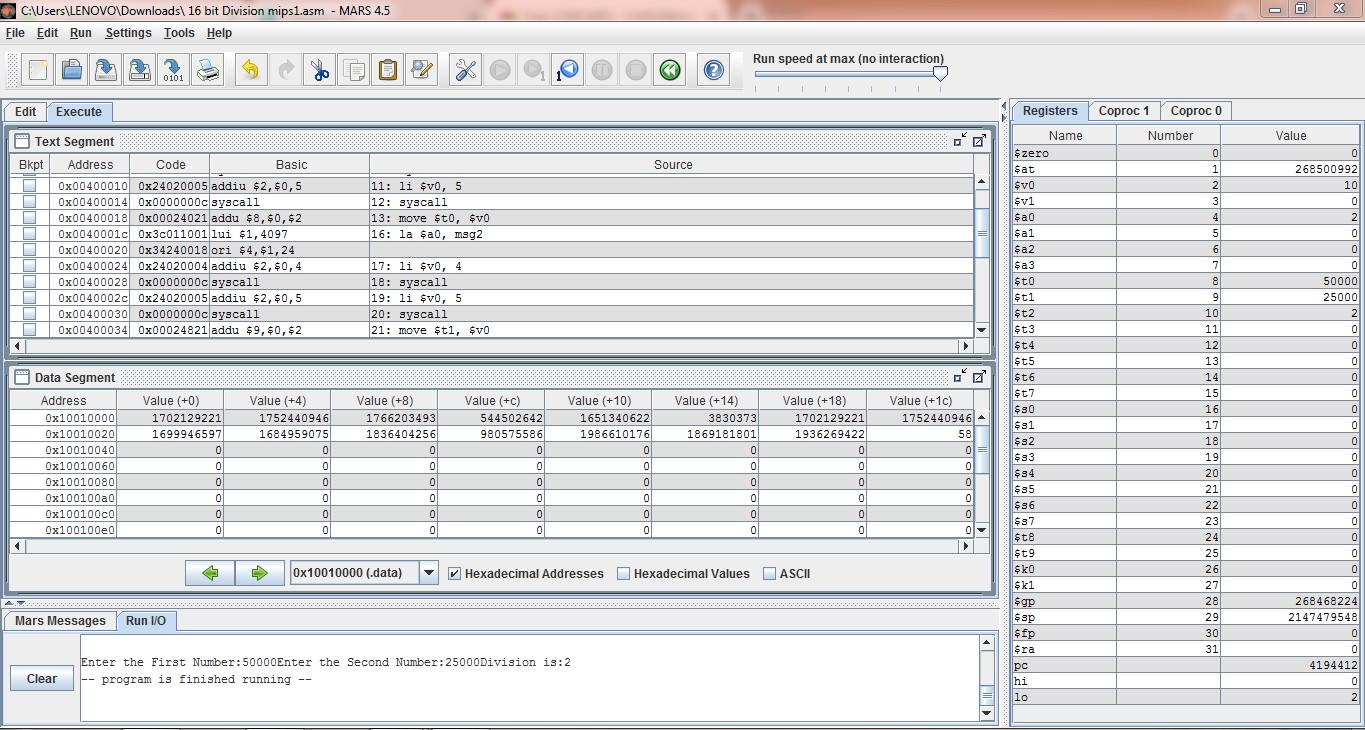
li $v0, 4

syscall

div $t2,$t0,$t1 move $a0,$t2 li $v0,1 syscall

#end program:- li $v0, 10 syscall

# Results:



**Conclusion:**

From this practical I learnt about MIPS programs ,how to deal with them and how to write a code in MIPS assembly language.

And I came to know about the new platform where we do MIPS problem i.e. MARS.

**Sign: Date:**