***CP-2***

***GroupID***-11

***Group Members***: -

1. Aman Jaiswal (17114008)
2. Amit Vishwakarma (17114010)

***PID*** -62

***Problem*** -

Take a number N as an input from the user to store in memory. Convert the content of the variable N to binary-coded decimal (BCD) digits in the string variable S and save the string in memory. Print the output in the terminal.

1. ***C code for the given PID*** (***dec2BCD.c) –***

//GROUPID-11 (17114008\_17114010) AMAN JAISWAL & AMIT VISHWAKARMA

//DATE 28 OCTOBER 2018

//dec2BCD.c file used for converting an integer N to BINARY CODED DECIMAL

#include<stdio.h>

#include<string.h>

int main(){

char num[11];

char bcd[45];

int len;

char dig;

scanf("%s", num);

len=strlen(num);

bcd[0]='\0';

int i=0;

for( ;i<len;i++){

dig =num[i];

switch(dig){

case '0':

strcat(bcd,"0000 ");

break;

case '1':

strcat(bcd,"0001 ");

break;

case '2':

strcat(bcd,"0010 ");

break;

case '3':

strcat(bcd,"0011 ");

break;

case '4':

strcat(bcd,"0100 ");

break;

case '5':

strcat(bcd,"0101 ");

break;

case '6':

strcat(bcd,"0110 ");

break;

case '7':

strcat(bcd,"0111 ");

break;

case '8':

strcat(bcd,"1000 ");

break;

case '9':

strcat(bcd,"1001 ");

break;

default:

strcat(bcd,dig);

break;

}

}

printf("%s\n", bcd);

return 0;

}

1. ***MIPS Assembly language code written for the given PID*** (***decToBCD.asm) –***

##GROUPID-11 (17114008\_17114010) AMAN JAISWAL & AMIT VISHWAKARMA

##DATE 28 OCTOBER 2018

## decToBCD.asm file used for converting an integer N to BINARY CODED DECIMAL

##REGISTERS USED

## $vo -used to invoke a system call for taking input from the user or print/display something

## $a0 - used as an argument register

## $t2 store the digit from 0 to 9

## $t1 is used for comparing every digit of number with $t2

## bcd0 to bcd9 store binary corresponding to each digit

## str contains final output string

.data

enter\_num : .asciiz "Enter Number\n"

bcd\_out : .asciiz "BCD String\n"

bcd0 : .asciiz "0000 "

bcd1 : .asciiz "0001 "

bcd2 : .asciiz "0010 "

bcd3 : .asciiz "0011 "

bcd4 : .asciiz "0100 "

bcd5 : .asciiz "0101 "

bcd6 : .asciiz "0110 "

bcd7 : .asciiz "0111 "

bcd8 : .asciiz "1000 "

bcd9 : .asciiz "1001 "

str : .space 32

.text

main:

la $a0,enter\_num

li $v0,4

syscall

la $a0, str

li $a1, 32

li $v0, 8

syscall

la $t0,str

la $a0,bcd\_out

li $v0,4

syscall

loop:

lb $t1, 0($t0)

beqz $t1, terminate

else0:

li $t2, '0'

bne $t1 , $t2, else1

la $a0, bcd0

li $v0, 4

syscall

b iterate

else1:

li $t2, '1'

bne $t1 , $t2, else2

la $a0, bcd1

li $v0, 4

syscall

b iterate

else2:

li $t2, '2'

bne $t1 , $t2, else3

la $a0, bcd2

li $v0, 4

syscall

b iterate

else3:

li $t2, '3'

bne $t1 , $t2, else4

la $a0, bcd3

li $v0, 4

syscall

b iterate

else4:

li $t2, '4'

bne $t1 , $t2, else5

la $a0, bcd4

li $v0, 4

syscall

b iterate

else5:

li $t2, '5'

bne $t1 , $t2, else6

la $a0, bcd5

li $v0, 4

syscall

b iterate

else6:

li $t2, '6'

bne $t1 , $t2, else7

la $a0, bcd6

li $v0, 11

syscall

b iterate

else7:

li $t2, '7'

bne $t1 , $t2, else8

la $a0, bcd7

li $v0, 4

syscall

b iterate

else8:

li $t2, '8'

bne $t1 , $t2, else9

la $a0, bcd8

li $v0, 4

syscall

b iterate

else9:

li $t2, '9'

bne $t1 , $t2, def

la $a0, bcd9

li $v0, 4

syscall

b iterate

def:

la $a0, ($t1)

li $v0, 11

syscall

iterate:

addu $t0, $t0, 1

b loop

terminate:

exit:

li $v0, 10

syscall

1. ***Assembly language program file generated by C compiler GCC (decToBCD.s)-***

##GROUPID-11 (17114008\_17114010) AMAN JAISWAL & AMIT VISHWAKARMA

##DATE 28 OCTOBER 2018

##assembly language generated by c compiler decToBCD.s

.file 1 ""

.section .mdebug.abi32

.previous

.nan legacy

.module fp=32

.module nooddspreg

.abicalls

.rdata

.align 2

$LC0:

.ascii "%s\000"

.align 2

$LC1:

.ascii "0000 \000"

.align 2

$LC2:

.ascii "0001 \000"

.align 2

$LC3:

.ascii "0010 \000"

.align 2

$LC4:

.ascii "0011 \000"

.align 2

$LC5:

.ascii "0100 \000"

.align 2

$LC6:

.ascii "0101 \000"

.align 2

$LC7:

.ascii "0110 \000"

.align 2

$LC8:

.ascii "0111 \000"

.align 2

$LC9:

.ascii "1000 \000"

.align 2

$LC10:

.ascii "1001 \000"

.text

.align 2

.globl main

.set nomips16

.set nomicromips

.ent main

.type main, @function

main:

.frame $fp,104,$31 # vars= 72, regs= 2/0, args= 16, gp= 8

.mask 0xc0000000,-4

.fmask 0x00000000,0

.set noreorder

.cpload $25

.set nomacro

addiu $sp,$sp,-104

sw $31,100($sp)

sw $fp,96($sp)

move $fp,$sp

.cprestore 16

movz $31,$31,$0

addiu $2,$fp,36

move $5,$2

lw $2,%got($LC0)($28)

nop

addiu $4,$2,%lo($LC0)

lw $2,%call16(\_\_isoc99\_scanf)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,\_\_isoc99\_scanf

1: jalr $25

nop

lw $28,16($fp)

addiu $2,$fp,36

move $4,$2

lw $2,%call16(strlen)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,strlen

1: jalr $25

nop

lw $28,16($fp)

sw $2,28($fp)

sb $0,48($fp)

sw $0,24($fp)

b $L2

nop

$L16:

lw $2,24($fp)

addiu $3,$fp,24

addu $2,$3,$2

lbu $2,12($2)

nop

sb $2,32($fp)

lb $2,32($fp)

nop

addiu $2,$2,-48

sltu $3,$2,10

beq $3,$0,$L3

nop

sll $3,$2,2

lw $2,%got($L5)($28)

nop

addiu $2,$2,%lo($L5)

addu $2,$3,$2

lw $2,0($2)

nop

addu $2,$2,$28

j $2

nop

.rdata

.align 2

.align 2

$L5:

.gpword $L4

.gpword $L6

.gpword $L7

.gpword $L8

.gpword $L9

.gpword $L10

.gpword $L11

.gpword $L12

.gpword $L13

.gpword $L14

.text

$L4:

addiu $2,$fp,48

move $4,$2

lw $2,%call16(strlen)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,strlen

1: jalr $25

nop

lw $28,16($fp)

move $3,$2

addiu $2,$fp,48

addu $2,$2,$3

lw $3,%got($LC1)($28)

nop

lw $4,%lo($LC1)($3)

nop

swl $4,0($2)

swr $4,3($2)

addiu $3,$3,%lo($LC1)

lbu $4,4($3)

nop

sb $4,4($2)

lbu $3,5($3)

nop

sb $3,5($2)

b $L15

nop

$L6:

addiu $2,$fp,48

move $4,$2

lw $2,%call16(strlen)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,strlen

1: jalr $25

nop

lw $28,16($fp)

move $3,$2

addiu $2,$fp,48

addu $2,$2,$3

lw $3,%got($LC2)($28)

nop

lw $4,%lo($LC2)($3)

nop

swl $4,0($2)

swr $4,3($2)

addiu $3,$3,%lo($LC2)

lbu $4,4($3)

nop

sb $4,4($2)

lbu $3,5($3)

nop

sb $3,5($2)

b $L15

nop

$L7:

addiu $2,$fp,48

move $4,$2

lw $2,%call16(strlen)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,strlen

1: jalr $25

nop

lw $28,16($fp)

move $3,$2

addiu $2,$fp,48

addu $2,$2,$3

lw $3,%got($LC3)($28)

nop

lw $4,%lo($LC3)($3)

nop

swl $4,0($2)

swr $4,3($2)

addiu $3,$3,%lo($LC3)

lbu $4,4($3)

nop

sb $4,4($2)

lbu $3,5($3)

nop

sb $3,5($2)

b $L15

nop

$L8:

addiu $2,$fp,48

move $4,$2

lw $2,%call16(strlen)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,strlen

1: jalr $25

nop

lw $28,16($fp)

move $3,$2

addiu $2,$fp,48

addu $2,$2,$3

lw $3,%got($LC4)($28)

nop

lw $4,%lo($LC4)($3)

nop

swl $4,0($2)

swr $4,3($2)

addiu $3,$3,%lo($LC4)

lbu $4,4($3)

nop

sb $4,4($2)

lbu $3,5($3)

nop

sb $3,5($2)

b $L15

nop

$L9:

addiu $2,$fp,48

move $4,$2

lw $2,%call16(strlen)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,strlen

1: jalr $25

nop

lw $28,16($fp)

move $3,$2

addiu $2,$fp,48

addu $2,$2,$3

lw $3,%got($LC5)($28)

nop

lw $4,%lo($LC5)($3)

nop

swl $4,0($2)

swr $4,3($2)

addiu $3,$3,%lo($LC5)

lbu $4,4($3)

nop

sb $4,4($2)

lbu $3,5($3)

nop

sb $3,5($2)

b $L15

nop

$L10:

addiu $2,$fp,48

move $4,$2

lw $2,%call16(strlen)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,strlen

1: jalr $25

nop

lw $28,16($fp)

move $3,$2

addiu $2,$fp,48

addu $2,$2,$3

lw $3,%got($LC6)($28)

nop

lw $4,%lo($LC6)($3)

nop

swl $4,0($2)

swr $4,3($2)

addiu $3,$3,%lo($LC6)

lbu $4,4($3)

nop

sb $4,4($2)

lbu $3,5($3)

nop

sb $3,5($2)

b $L15

nop

$L11:

addiu $2,$fp,48

move $4,$2

lw $2,%call16(strlen)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,strlen

1: jalr $25

nop

lw $28,16($fp)

move $3,$2

addiu $2,$fp,48

addu $2,$2,$3

lw $3,%got($LC7)($28)

nop

lw $4,%lo($LC7)($3)

nop

swl $4,0($2)

swr $4,3($2)

addiu $3,$3,%lo($LC7)

lbu $4,4($3)

nop

sb $4,4($2)

lbu $3,5($3)

nop

sb $3,5($2)

b $L15

nop

$L12:

addiu $2,$fp,48

move $4,$2

lw $2,%call16(strlen)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,strlen

1: jalr $25

nop

lw $28,16($fp)

move $3,$2

addiu $2,$fp,48

addu $2,$2,$3

lw $3,%got($LC8)($28)

nop

lw $4,%lo($LC8)($3)

nop

swl $4,0($2)

swr $4,3($2)

addiu $3,$3,%lo($LC8)

lbu $4,4($3)

nop

sb $4,4($2)

lbu $3,5($3)

nop

sb $3,5($2)

b $L15

nop

$L13:

addiu $2,$fp,48

move $4,$2

lw $2,%call16(strlen)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,strlen

1: jalr $25

nop

lw $28,16($fp)

move $3,$2

addiu $2,$fp,48

addu $2,$2,$3

lw $3,%got($LC9)($28)

nop

lw $4,%lo($LC9)($3)

nop

swl $4,0($2)

swr $4,3($2)

addiu $3,$3,%lo($LC9)

lbu $4,4($3)

nop

sb $4,4($2)

lbu $3,5($3)

nop

sb $3,5($2)

b $L15

nop

$L14:

addiu $2,$fp,48

move $4,$2

lw $2,%call16(strlen)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,strlen

1: jalr $25

nop

lw $28,16($fp)

move $3,$2

addiu $2,$fp,48

addu $2,$2,$3

lw $3,%got($LC10)($28)

nop

lw $4,%lo($LC10)($3)

nop

swl $4,0($2)

swr $4,3($2)

addiu $3,$3,%lo($LC10)

lbu $4,4($3)

nop

sb $4,4($2)

lbu $3,5($3)

nop

sb $3,5($2)

b $L15

nop

$L3:

lb $2,32($fp)

nop

move $3,$2

addiu $2,$fp,48

move $5,$3

move $4,$2

lw $2,%call16(strcat)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,strcat

1: jalr $25

nop

lw $28,16($fp)

nop

$L15:

lw $2,24($fp)

nop

addiu $2,$2,1

sw $2,24($fp)

$L2:

lw $3,24($fp)

lw $2,28($fp)

nop

slt $2,$3,$2

bne $2,$0,$L16

nop

addiu $2,$fp,48

move $4,$2

lw $2,%call16(puts)($28)

nop

move $25,$2

.reloc 1f,R\_MIPS\_JALR,puts

1: jalr $25

nop

lw $28,16($fp)

move $2,$0

move $sp,$fp

lw $31,100($sp)

lw $fp,96($sp)

addiu $sp,$sp,104

j $31

nop

.set macro

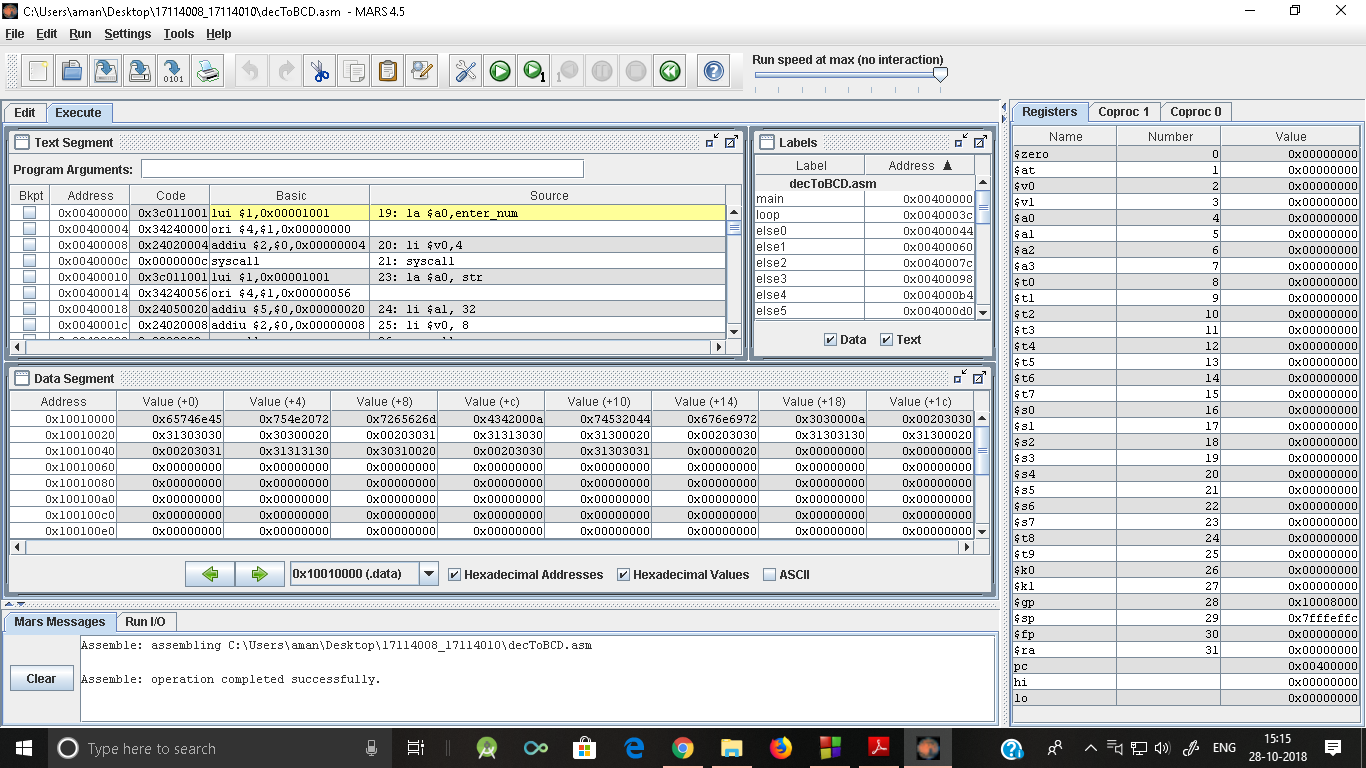
.set reorder

.end main

.size main, .-main

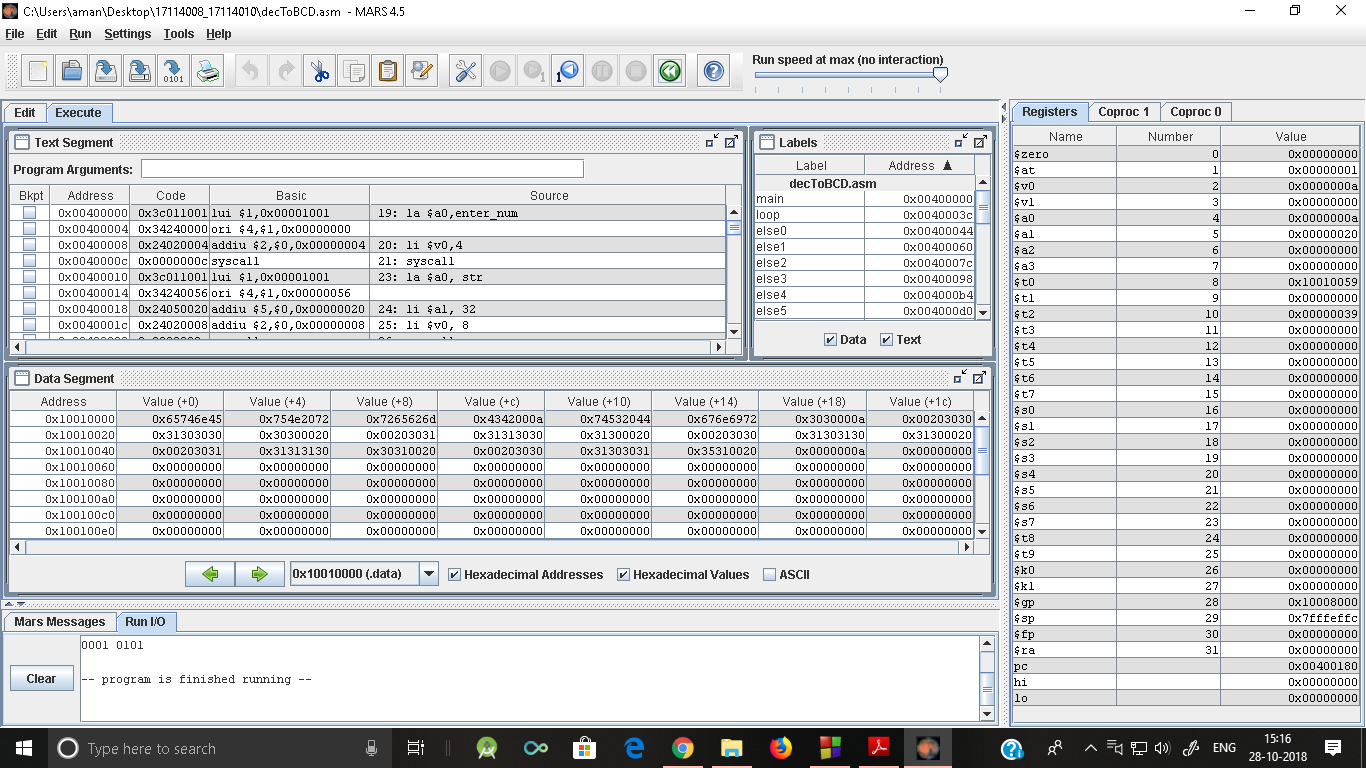
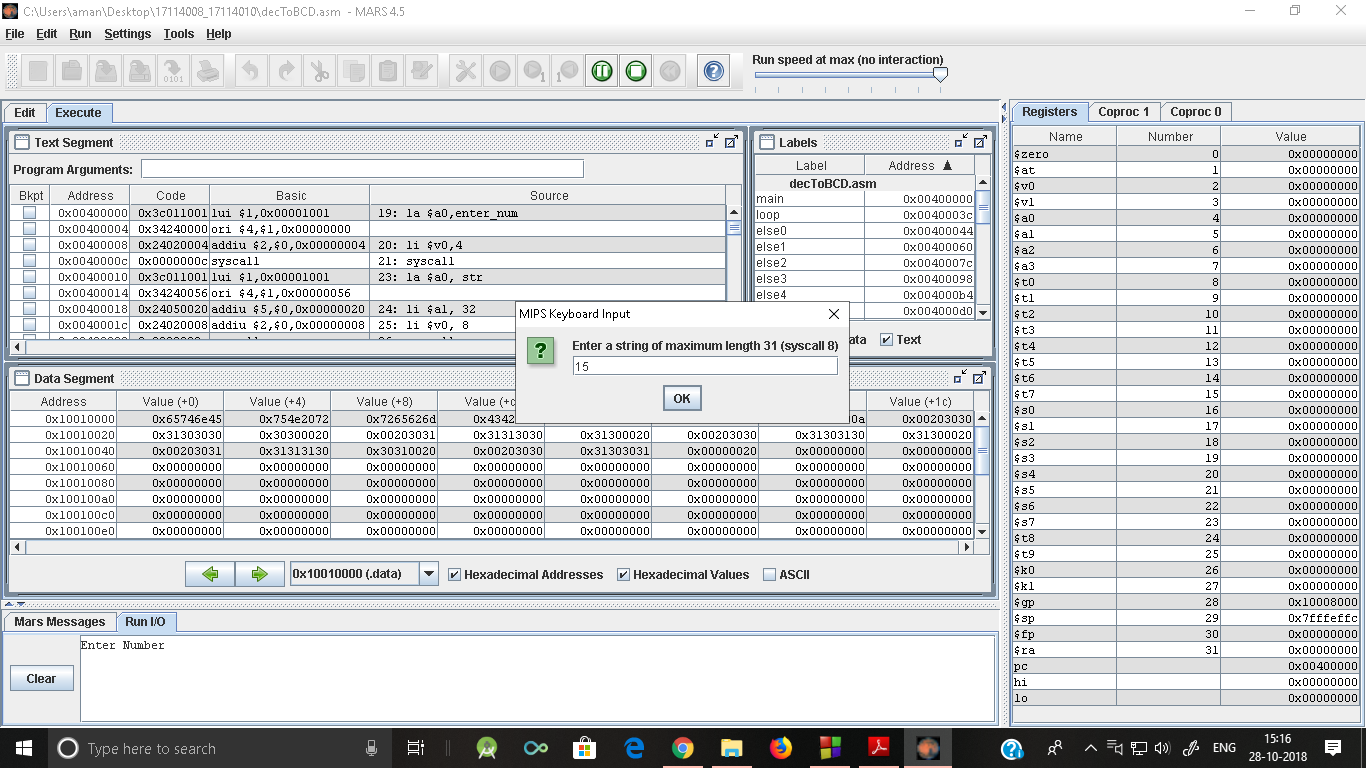
.ident "GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.9) 5.4.0 20160609"

1. ***Screenshots taken after simulating the assembly code-***

******

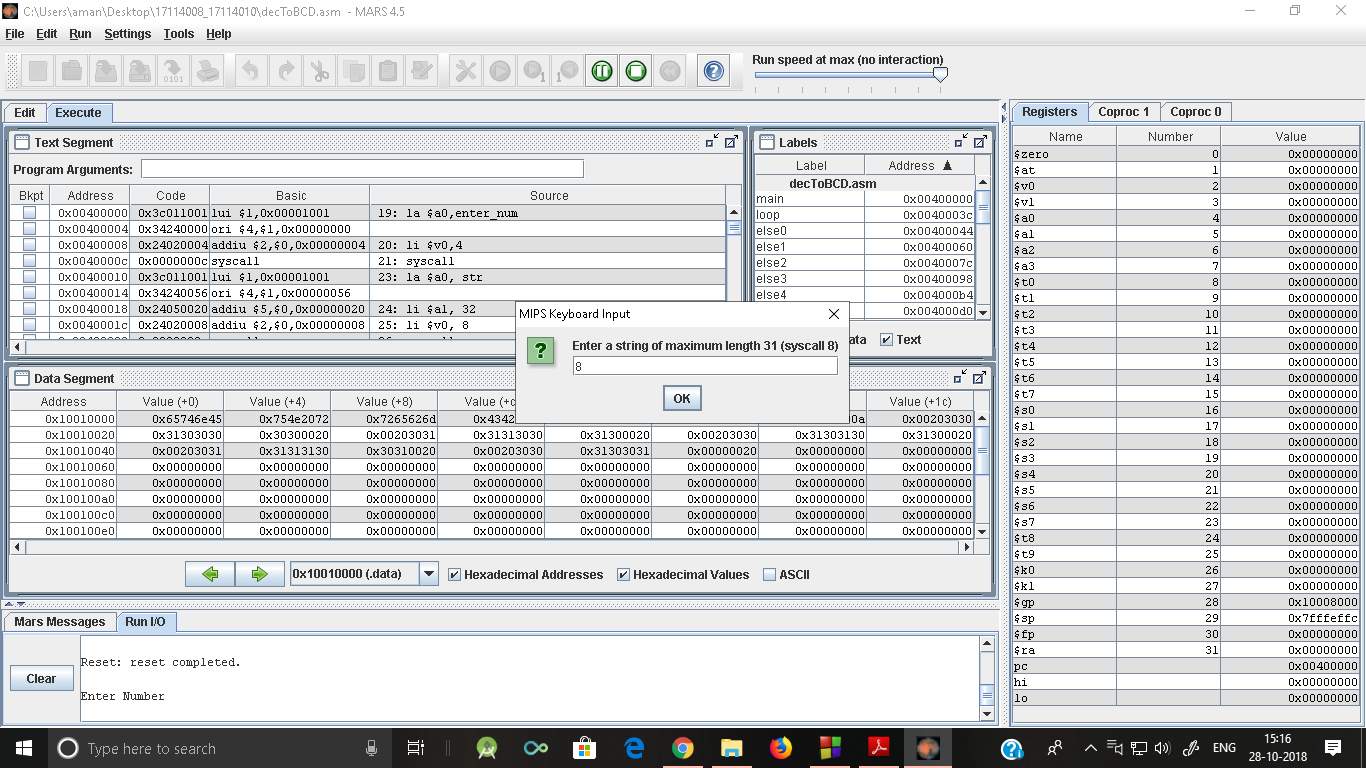
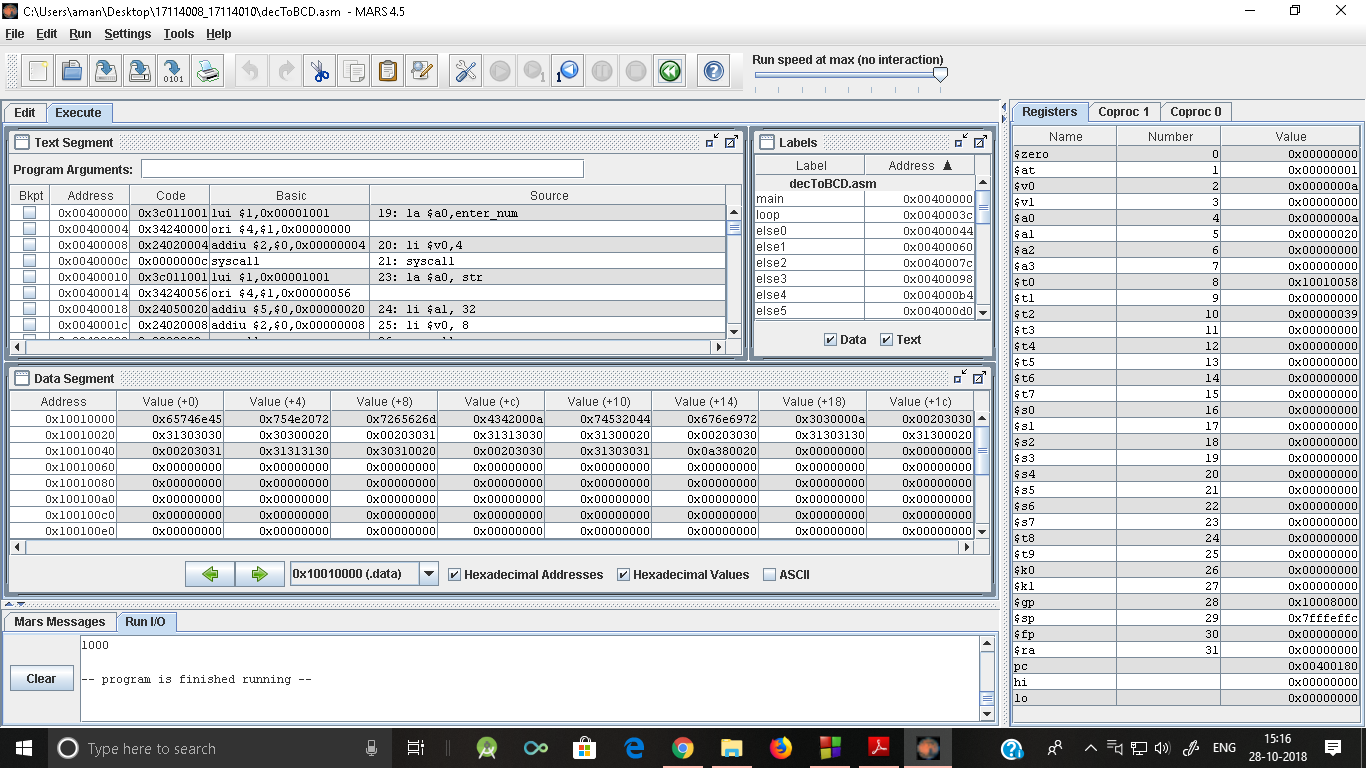
* Input -15

Output -0001 0101



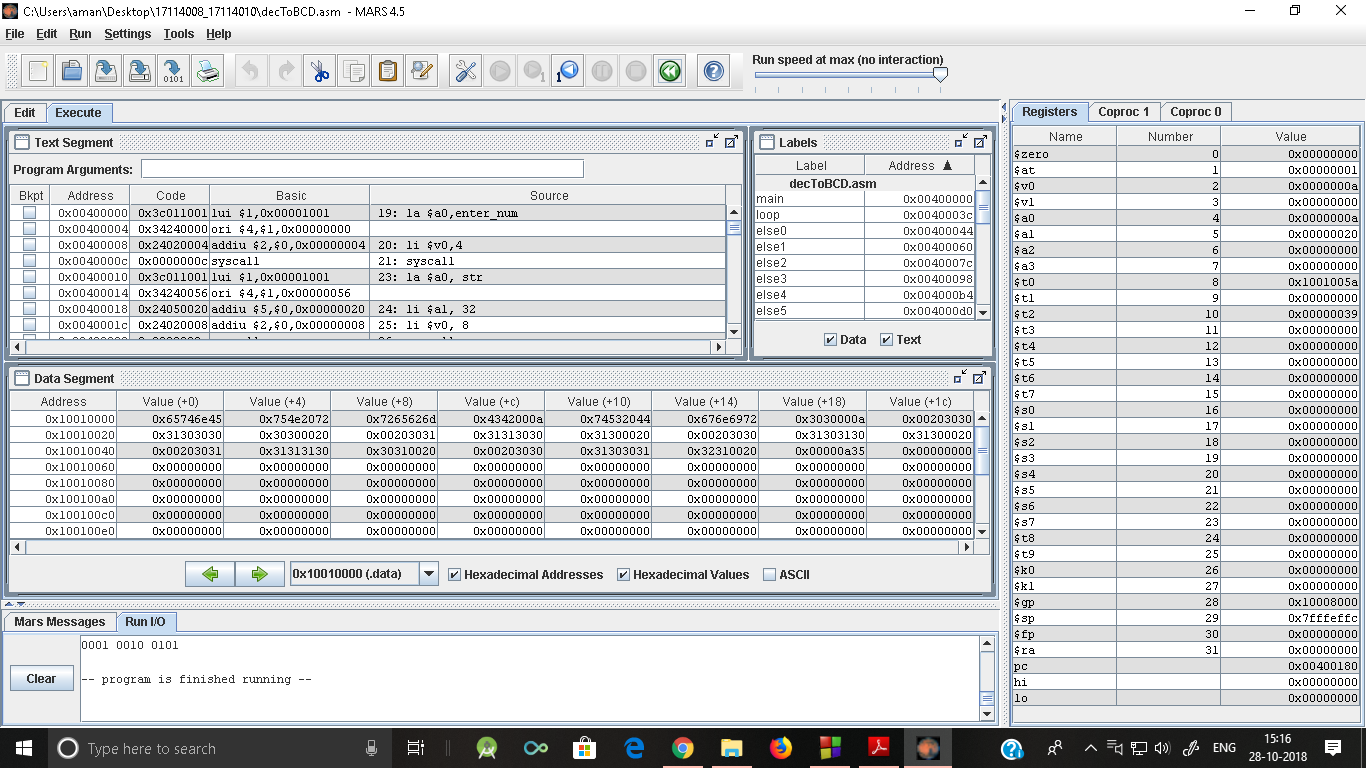
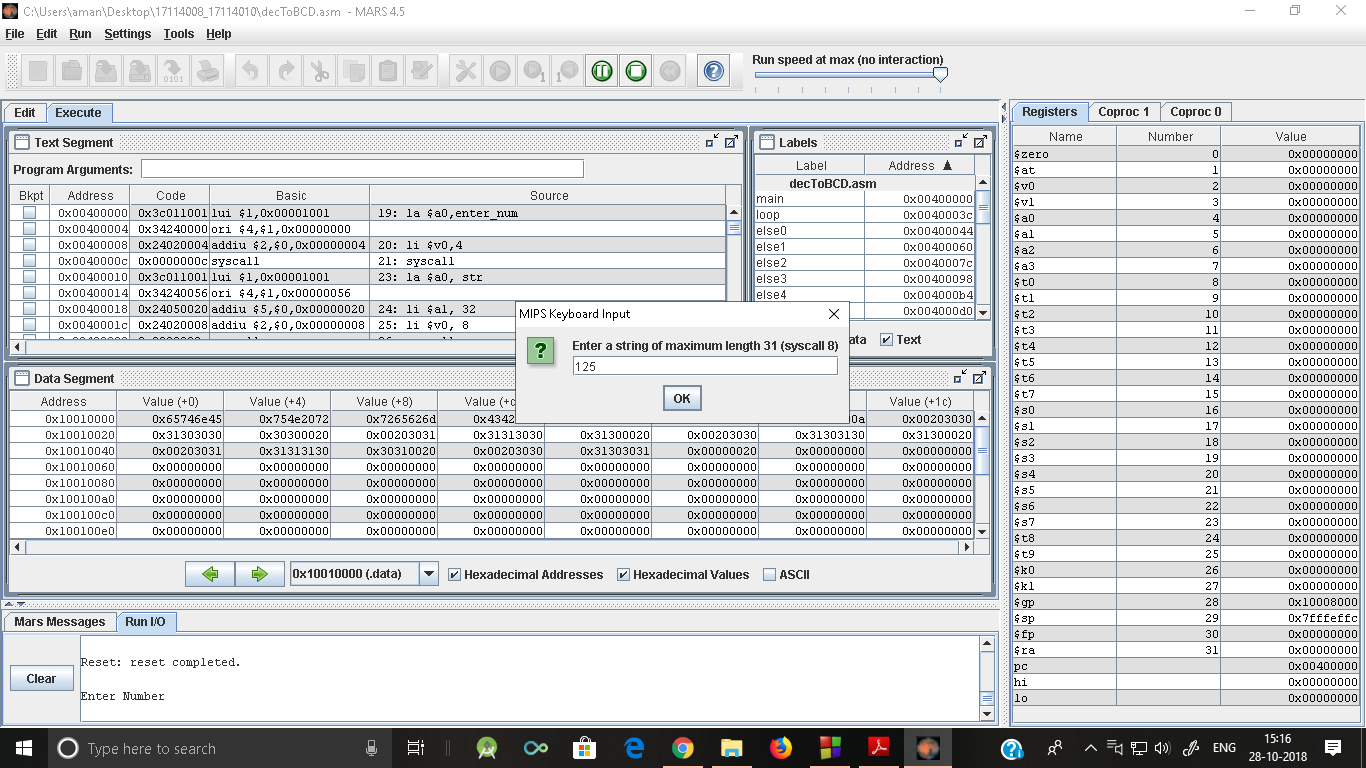
* ***Input-8***

***Output-1000***



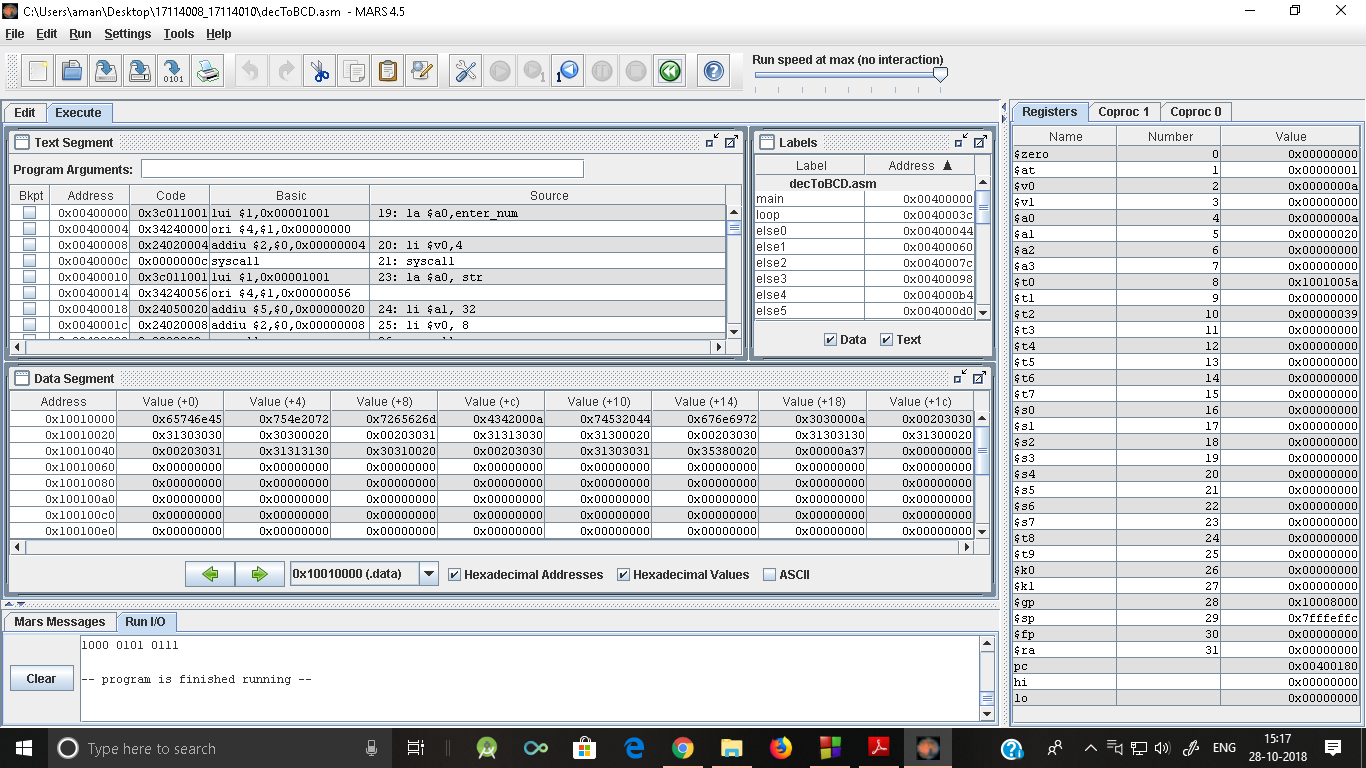
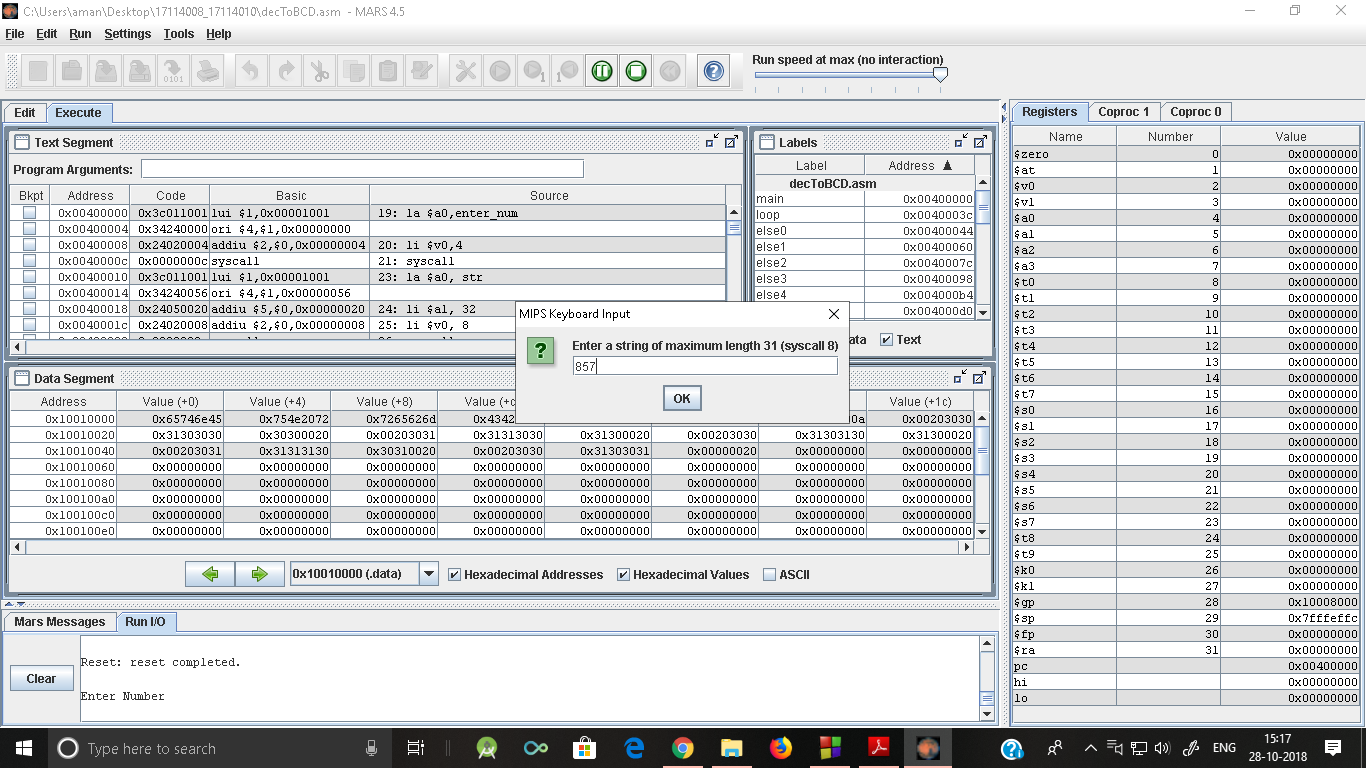
* ***Input-125***

***Output-0001 0010 0101***



* ***Input -857***

***Output-1000 0101 0111***



Thank you!