|  |
| --- |
| FAST National University |
| **Lab 6** |
|  |

**Computer Organization and Assembly Language**

|  |  |
| --- | --- |
| **Student Name** | Muhammad Ali Butt |
| **Registration #** | 21L-7646 |
| **Instructor** | Hazoor Ahmad |
| **Class** | Lab-CS2A |
| **Section** | BCS-3A2 |
| **Semester** | Fall 2022 |

Fast School of Computing

FAST-NU, Lahore, Pakistan

# Activity 1

## **Assembly Language Code**

[org 0x0100]

jmp start

clrscr:

push ax

push es

push di

mov ax, 0xb800

mov es, ax

mov di, 0

nextchar:

mov word [es:di], 0x0720

add di, 2

cmp di, 4000

jb nextchar

pop di

pop es

pop ax

ret

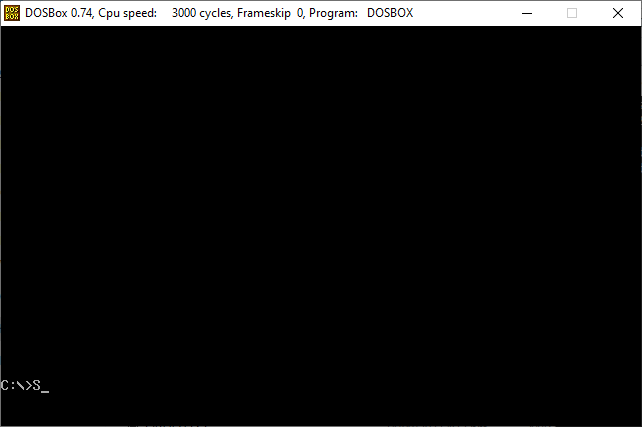
start:

call clrscr

mov ax, 0x4c00

int 0x21

## **Debugging Screenshots**



# Activity 2

## **Assembly Language Code**

[org 0x0100]

jmp start

asciis: db '0000'

clrscr:

push ax

push es

push di

mov ax, 0xb800

mov es, ax

mov di, 0

nextchar:

mov word [es:di], 0x0720

add di, 2

cmp di, 4000

jb nextchar

pop di

pop es

pop ax

ret

ASCII\_convert: ;Roll number will be stored in ax

push es

push bx

push dx

push di

push si

mov bx, 0xb800

mov es, bx

mov si, 3 ;Will be used to store the ASCII codes in memory

mov bx, 16 ;Will be used for division

nextdigit:

mov dx, 0

div bx

add dl, 0x30

mov [asciis+si], dl

dec si

cmp ax, 0

jnz nextdigit

pop si

pop di

pop dx

pop bx

pop es

ret

start:

call clrscr

mov ax, 0x7646

call ASCII\_convert ;ASCIIs will be stored in memory

mov bx, 0xb800

mov es, bx

mov bx, 0

mov si, 0

mov cx, 4

mov ah, 0x07

printRollNum:

mov al, [asciis+bx]

mov [es:si], ax

add bx, 1

add si, 2

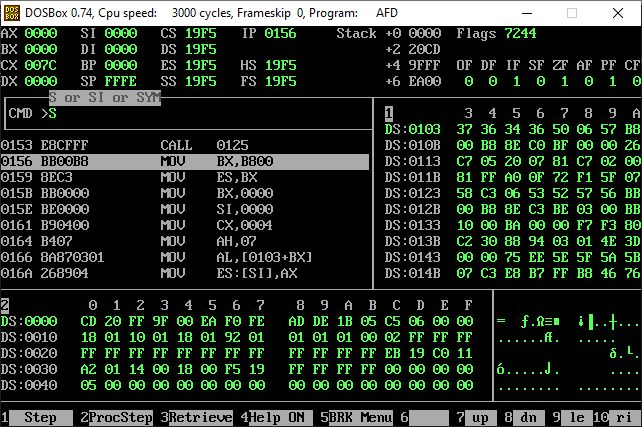
loop printRollNum

mov ax, 0x4c00

int 0x21

## **Debugging Screenshots**

ASCII codes (37, 36, 34, 36) stored in memory:



# Activity 3

## **Assembly Language Code**

[org 0x0100]

jmp start

sentence: db 'My name is Ali and my roll number is '

asciis: db '0000'

clrscr:

push ax

push es

push di

mov ax, 0xb800

mov es, ax

mov di, 0

nextchar:

mov word [es:di], 0x0720

add di, 2

cmp di, 4000

jb nextchar

pop di

pop es

pop ax

ret

ASCII\_convert: ;Roll number will be stored in ax

push es

push bx

push dx

push di

push si

mov bx, 0xb800

mov es, bx

mov si, 3 ;Will be used to store the ASCII codes in memory

mov bx, 16 ;Will be used for division

nextdigit:

mov dx, 0

div bx

add dl, 0x30

mov [asciis+si], dl

dec si

cmp ax, 0

jnz nextdigit

pop si

pop di

pop dx

pop bx

pop es

ret

start:

call clrscr

mov ax, 0x7646

call ASCII\_convert ;ASCIIs will be stored in memory

mov bx, 0xb800

mov es, bx

mov ah, 0x07

mov bx, 0

mov di, 0

mov cx, 37 ;Storing size of the sentence in cx

printSentence: ;Printing the sentence

mov al, [sentence+bx]

mov [es:di], ax

inc bx

add di, 2

loop printSentence

mov cx, 4

mov bx, 0

printRollNum:

mov al, [asciis+bx]

mov [es:di], ax

add bx, 1

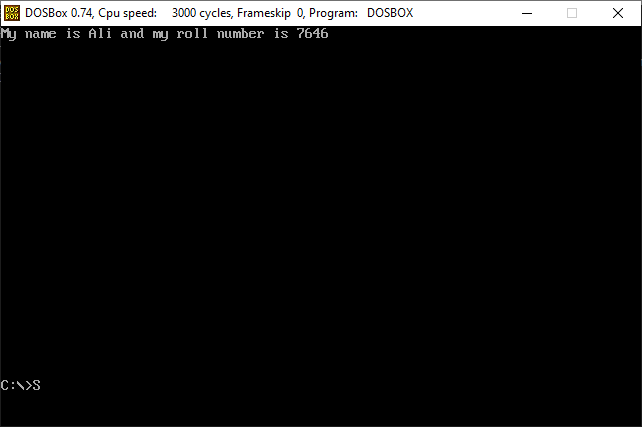
add di, 2

loop printRollNum

mov ax, 0x4c00

int 0x21

## **Debugging Screenshots**



# Activity 4

## **Assembly Language Code**

[org 0x0100]

jmp start

digits: db 0, 0, 0, 0

r: dw 0

l: dw 0

clrscr:

push ax

push es

push di

mov ax, 0xb800

mov es, ax

mov di, 0

nextchar:

mov word [es:di], 0x0720

add di, 2

cmp di, 4000

jb nextchar

pop di

pop es

pop ax

ret

Digit\_convert: ;Roll number will be stored in ax

push bx

push dx

push si

mov bx, 0xb800

mov si, 3 ;Will be used to store the digits in memory

mov bx, 16 ;Will be used for division

nextdigit:

mov dx, 0

div bx

mov [digits+si], dl

dec si

cmp ax, 0

jnz nextdigit

pop si

pop dx

pop bx

ret

calRandL:

push ax

push dx

mov dx, 0

mov al, [digits]

add dx, ax

mov al, [digits+1]

add dx, ax

mov al, [digits+2]

add dx, ax

mov al, [digits+3]

add dx, ax

shr dx, 2

mov [r], dx

add dx, 3

mov [l], dx

pop ax

pop dx

ret

start:

call clrscr

mov ax, 0x7646

call Digit\_convert

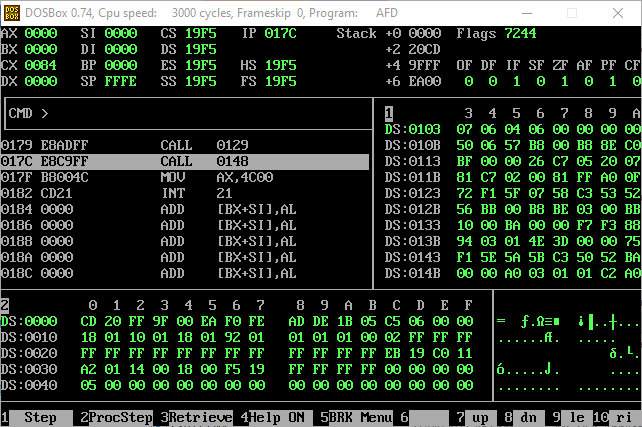
call calRandL

mov ax, 0x4c00

int 0x21

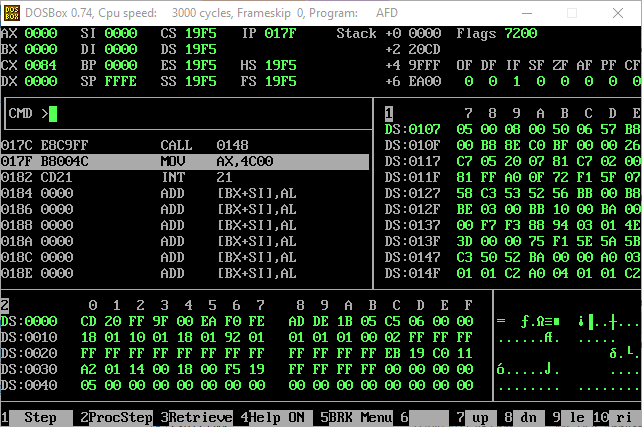
## **Debugging Screenshots**

Digits of the roll number stored in memory:



Values of r and l (5 and 8 respectively) stored in memory:

=



# Activity 5

## **Assembly Language Code**

[org 0x0100]

jmp start

digits: db 0, 0, 0, 0

r: dw 0

l: dw 0

f: dd 0

clrscr:

push ax

push es

push di

mov ax, 0xb800

mov es, ax

mov di, 0

nextchar:

mov word [es:di], 0x0720

add di, 2

cmp di, 4000

jb nextchar

pop di

pop es

pop ax

ret

Digit\_convert: ;Roll number will be stored in ax

push bx

push dx

push si

mov bx, 0xb800

mov si, 3 ;Will be used to store the digits in memory

mov bx, 16 ;Will be used for division

nextdigit:

mov dx, 0

div bx

mov [digits+si], dl

dec si

cmp ax, 0

jnz nextdigit

pop si

pop dx

pop bx

ret

calRandL:

push ax

push dx

mov dx, 0

mov al, [digits]

add dx, ax

mov al, [digits+1]

add dx, ax

mov al, [digits+2]

add dx, ax

mov al, [digits+3]

add dx, ax

shr dx, 2

mov [r], dx

add dx, 3

mov [l], dx

pop ax

pop dx

ret

calF:

push ax

push bx

start:

call clrscr

mov ax, 0x7646

call Digit\_convert

call calRandL

call calF

mov ax, 0x4c00

int 0x21