Northern University

of Business and Technology Khulna

Lab Report – 03

Course Code: CSE 3204

Course Title: Microprocessor and Assembly Language Programming Lab

Submitted By

Name : Hemendra Nath Mondal
Section : 6B
ID : 11220320873
Dept : Computer Science & Engineering

Submitted To

Name : Md. Abdul Halim Khan
Designation : Lecturer
Dept : Computer Science & Engineering

Lab Report – 03

Description:

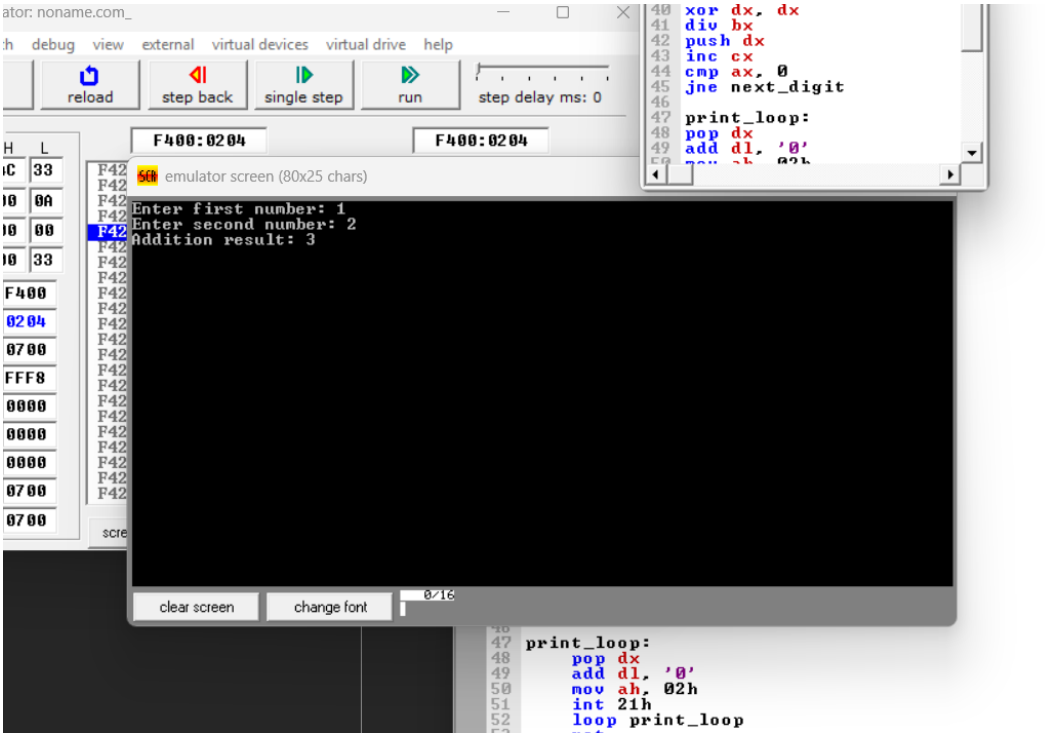
The three assembly programs perform basic arithmetic operations on two single-digit numbers input by the user. The addition program reads two digits, converts them from ASCII to numeric values, adds them, and then converts and prints the sum as decimal digits. The multiplication program similarly reads two digits, multiplies them using the MUL instruction which stores the result in a 16-bit register, and then converts and prints the potentially two-digit product. The division program takes a dividend and divisor, performs integer division using DIV, and then prints both the quotient and remainder after converting them to decimal characters. Each program handles user input and output carefully to display the correct arithmetic results in a readable format.

Code:

```
org 100h

mov ah, 09h
mov dx, offset msg1
int 21h

; Read first digit
mov ah, 01h
int 21h
sub al, '0'
mov bl, al

mov ah, 09h
mov dx, offset msg2
int 21h

; Read second digit
mov ah, 01h
int 21h
sub al, '0'
add bl, al

mov ah, 09h
mov dx, offset msg3
int 21h

; Convert and print result
mov ax, 0
mov al, bl
call print_num

mov ah, 4ch
int 21h
```

```
; --- Procedures ---
print_num:     ; print number in AX
    mov cx, 0
    mov bx, 10
next_digit:
    xor dx, dx
    div bx
    push dx
    inc cx
    cmp ax, 0
    jne next_digit

print_loop:
    pop dx
    add dl, '0'
    mov ah, 02h
    int 21h
    loop print_loop
    ret

msg1 db 'Enter first number: $'
msg2 db 13, 10, 'Enter second number: $'
msg3 db 13, 10, 'Addition result: $'
```

Output:

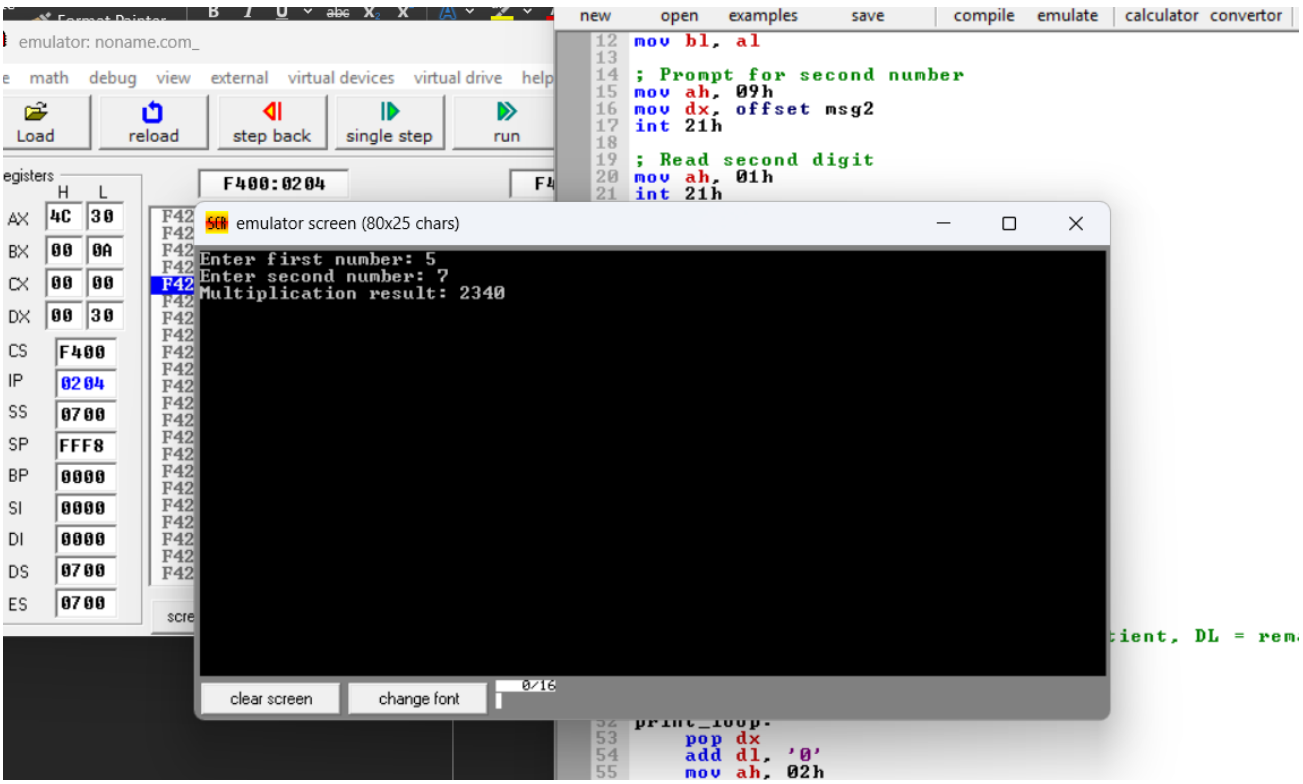


Code:

```
org 100h

; Prompt for first number
mov ah, 09h
mov dx, offset msg1
int 21h

; Read first digit
mov ah, 01h
int 21h
sub al, '0'
mov bl, al

; Prompt for second number
mov ah, 09h
mov dx, offset msg2
int 21h

; Read second digit
mov ah, 01h
int 21h
sub al, '0'
mov bh, al

; Perform multiplication
mov al, bl
mul bh        ; AL * BH = AX

; Show result message
mov ah, 09h
mov dx, offset msg3
int 21h

; AX contains result
call print_num

; Exit to DOS
mov ah, 4ch
int 21h

; --- Print procedure ---
print_num:
   mov cx, 0
   mov bx, 10
next_digit:
   xor dx, dx
```

```
    div bx      ; AX ÷ 10 → AL = quotient, DL = remainder
    push dx
    inc cx
    cmp ax, 0
    jne next_digit
print_loop:
    pop dx
    add dl, '0'
    mov ah, 02h
    int 21h
    loop print_loop
    ret

; --- Messages ---
msg1 db 'Enter first number: $'
msg2 db 13, 10, 'Enter second number: $'
msg3 db 13, 10, 'Multiplication result: $'
```

Output:



Code:

```
org 100h

mov ah, 09h
mov dx, offset msg1
int 21h

; Read dividend
mov ah, 01h
int 21h
sub al, '0'
mov bl, al

mov ah, 09h
mov dx, offset msg2
int 21h

; Read divisor
mov ah, 01h
int 21h
sub al, '0'
mov bh, al

; Divide
mov al, bl
mov ah, 0
div bh ; AL / BH → AL = quotient, AH = remainder

; Print quotient
mov ah, 09h
mov dx, offset msg3
int 21h
mov ax, 0
mov al, al
call print_num

; Print remainder
mov ah, 09h
mov dx, offset msg4
int 21h
mov ax, 0
mov al, ah
call print_num

mov ah, 4ch
int 21h
; --- Procedures ---
```

```
print_num:
    mov cx, 0
    mov bx, 10
next_digit:
    xor dx, dx
    div bx
    push dx
    inc cx
    cmp ax, 0
    jne next_digit
print_loop:
    pop dx
    add dl, '0'
    mov ah, 02h
    int 21h
    loop print_loop
    ret

msg1 db 'Enter dividend: $'
msg2 db 13, 10, 'Enter divisor: $'
msg3 db 13, 10, 'Quotient: $'
msg4 db 13, 10, 'Remainder: $'
```

Output: