Name : Khushi Nitinkumar Patel

PRN : 2020BTECS00037

Experiment 14 : Write X86/64 ALP to perform multiplication of two 8-bit hexadecimal numbers. Use successive addition and add and shift method (Use of 64-bit registers is expected).

Source Code :

```
scall macro x,y                ;macro to take input and output
lea dx,x        mov ah,y
      int 21h
endm

.model small
.data

menu db 10d,13d,"                MENU For Multiplication"
db 10d,"1. Successive Addition"      db 10d,"2. Shift and
Add method"      db 10d,"3. Exit"
    db 10d
    db 10d,"Enter your choice: $"


      m1 db 10d,13d,"Enter First Number: $"
m2 db 10d,13d,"Enter Second Number: $"
m3 db 10d,13d,"Answer: $"
```

```
        nwline db 10d,13d,'$'

        choice db 1 dup('0')
num1 db 2 dup('0')        num2
db 2 dup('0')

.code        mov
ax,@data
        mov ds,ax

main:
        scall menu,09h
mov ah,01h
        int 21h

        cmp al,'3'
        jae exit

        mov [choice],al

        scall m1,09h
call numinput
        mov [num1],bl

        scall m2,09h call
        numinput
        mov [num2],bl

        mov al,[choice]
```

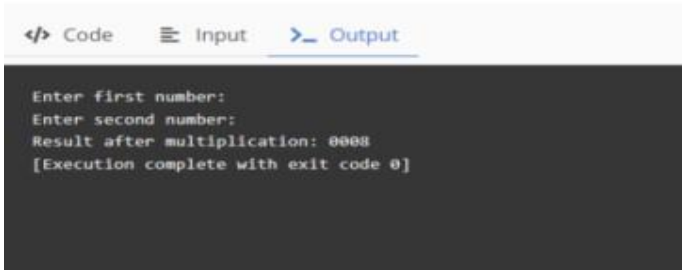
```
        cmp al,'1'
je case1
cmp al,'2'
        je case2
         exit:
       mov ah,4Ch
       int 21h

case1:
        mov bl,[num1]
mov cl,[num2]
        mov ax,0              ;ax to store answer
mov bh,0        mov ch,0

        cmp cl,0              ;check multiplication with 0 condition
je skip4 loop3:
        add ax,bx        loop loop3             ;auto-
decrement cx and jmp skip4:
        mov bx,ax             ;backup ax in bx
scall m3,09h
        call numdisplay           ;display answer from bx register jmp
main


case2:
        mov bl,[num1]        mov dl,[num2]
mov ax,0              ;ax to store answer
mov dh,0 mov bh,0

        mov cl,16 up1:
          shl ax,1
rol bx,1       jnc
```

```
down1       add
ax,dx down1:
        loop up1
mov bx,ax


      mov bx,ax           ;backup ax in bx
scall m3,09h
      call numdisplay      ;display answer from bx register


jmp main
numinput proc
      mov bl,0h
      mov ch,02h

;code to input 2 digit numbers loop1:
      mov ah,01h
      int 21h

      cmp al,39h
jbe skip1       sub
al,07h skip1:
      sub al,30h

      cmp ch,01H
      je skip2
      rol al,04H
skip2:
      add bl,al
```

```
        dec ch
jnz loop1

ret endp    ;End of
Procedure

numdisplay proc
mov dx,bx        mov
ch,04h ;code to display
4 digits loop2:
        rol dx,04h
rol bx,04h

        and dl,0Fh
          cmp dl,09h
jbe skip3
          add dl,07h
skip3:
add dl,30h

          mov ah,02h
int 21h


          mov dx,bx
dec ch        jnz
loop2

        scall nwline,09h ret
```

```
endp    ;End of Procedure

end     ;End of Program
```

Output :



Conclusion :

a. Shift-and-Add multiplication method adds the multiplicand X to itself Y times,where Y denotes the multiplier.

b. I used RAX Register(64-bit register).