PARSHVANATH CHARITABLE TRUST'S
A.P. SHAH INSTITUTE OF TECHNOLOGY
Department of Computer Science and Engineering
Data Science

Department of Computer Science Engineering Data Science

Academic Year: 2022-23 **Semester: IV**
Class / Branch: S.E.D.S. **Subject: Microprocessor Lab**

Experiment No. 2

1. **Aim: Write a menu driven Assembly Language Program to perform 16 bit addition subtraction,multiplication and division.**

2. **Software used:** tasm,tlink ,td,dosemu

3. **Theory :-**

   3.1 MACROS

   Macros are just like procedures, but not really. Macros look like procedures, but they exist only until your code is compiled, after compilation all macros are replaced with real instructions. If you declared a macro and never used it in your code, compiler will simply ignore it.

   Syntax:
   macro_name macro number_of_params
   <macro body> endm

   Example:

```
DISPLAY MACRO MSG
LEA DX,MSG
MOV AH,9
INT 21H
ENDM
```

   DISPLAY :- is the Name of the Macro. MACRO is the Keyword Used. MSG is the Argument Passed.

```
LEA DX,MSG      ; code inside macro
MOV AH,9
INT 21H
ENDM            ; is the end of Macro.
```

   The code which is used most of the time is written in between the macro for reducing the length of Code.

3.2 KEYBOARD INTERRUPTS

Taking Input from USER

i) MOV AH,0AH INT 21H

Keeps on taking input from user until terminated by '$'.The input is taken in a predefined buffer .

ii) MOV AH,01H INT 21H

Takes only one character from user. The input is taken in reg. AL

Display Messages

i) MOV AH,09H INT 21H

Displays a message terminated by '$'.
The Characters are taken in DX reg. (for word) or DL reg. (for byte) and Displayed. Example:

```
mov dx, offset msg
mov ah, 9
int 21h
ret
msg db "hello world $"
```

ii) MOV AH,02H INT 21H

Displays only single Character whose ASCII value is in DL reg. Example:

```
mov ah, 2
mov dl, 'a'
int 21h
```

4. Program

```
jumps

.model small

.data

a dw 2222h

b dw 1111h

menu db 10d,13d,"menu"

db 10d,"1. add two numbers"

db 10d,"2. subtract two numbers"
```

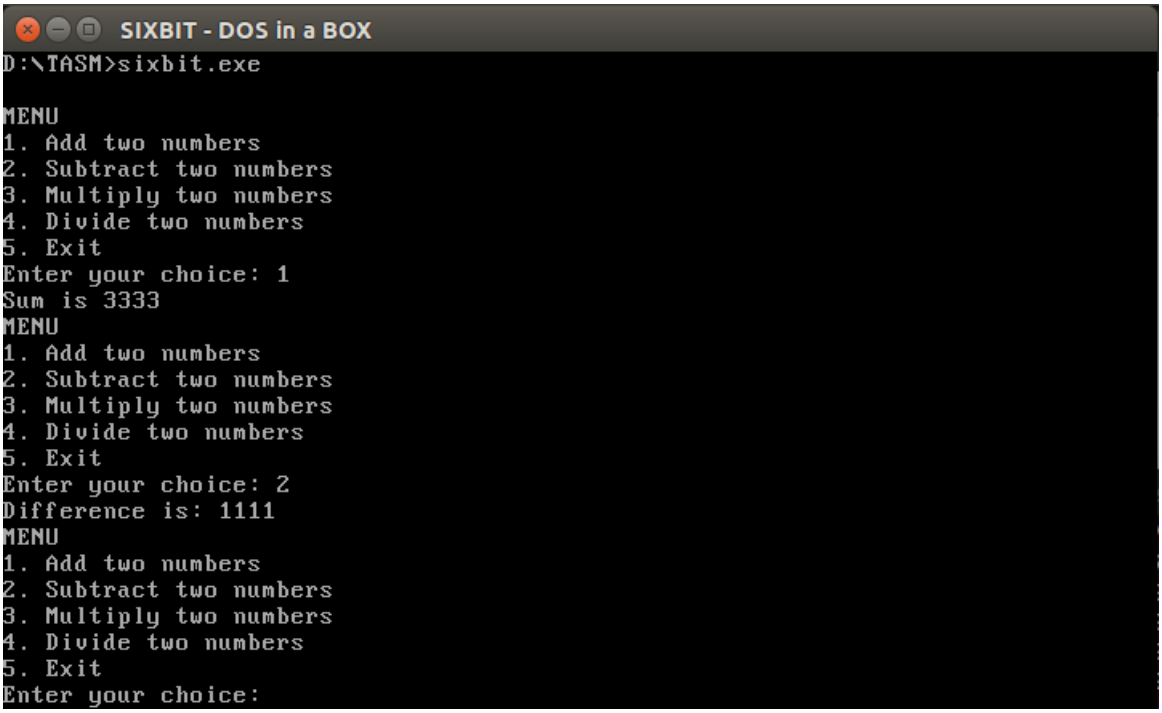
```
db 10d,"3. multiply two numbers"
db 10d,"4. divide two numbers"
db 10d,"5. exit"
db 10d,"enter your choice: $"
m1 db 10d,"sum is $"
m2 db 10d,"difference is: $"
m3 db 10d,"product is: $"
m4 db 10d,"quotient is: $"
print macro xx
lea dx,xx
mov ah,09h
int 21h
endm
.code
start:
mov ax,@data
mov ds, ax
main:
print menu
mov ah,01h
int  21h
cmp  al,'1'
je case1
cmp  al,'2'
je case2
cmp  al,'3'
je case3
cmp  al,'4'
je case4
cmp  al,'5'
je exit
exit:mov ah,4ch
    int 21h
```

```
case1:
print m1
      mov     ax, a          ; load number1 in ax
      mov     bx, b          ; load number2 in bx
      add     ax, bx         ; add numbers. result in ax
      mov     ch, 04h        ; count of digits to be displayed
      mov     cl, 04h        ; count to roll by 4 bits
      mov     bx, ax         ; result in reg bx
l2:   rol     bx, cl         ; roll bx so that msb comes to lsb
      mov     dl, bl         ; load dl with data to be displayed
      and     dl, 0fh        ; get only lsb
      cmp     dl, 09         ; check if digit is 0-9 or letter a-f
      jbe     l4
      add     dl, 07         ; if letter add 37h else only add 30h
l4:   add     dl, 30h
      mov     ah, 02         ; function 2 under int 21h (display character)
      int     21h
      dec     ch             ; decrement count
      jnz     l2
jmp main
case2:
print m2
      mov     ax, a          ; load number1 in ax
      mov     bx, b          ; load number2 in bx
      sub     ax, bx                 ; subtract numbers. result in ax
      mov     ch, 04h                ; count of digits to be displayed
      mov     cl, 04h                ; count to roll by 4 bits
      mov     bx, ax                 ; result in reg bx
l1:   rol     bx, cl                 ; roll bx so that msb comes to lsb
      mov     dl, bl                 ; load dl with data to be displayed
      and     dl, 0fh                ; get only lsb
      cmp     dl, 09                 ; check if digit is 0-9 or letter a-f
      jbe     l3
```

```
        add     dl, 07                  ; if letter add 37h else only add 30h
l3:     add     dl, 30h
        mov     ah, 02                  ; function 2 under int 21h (display character)
        int     21h
        dec     ch                      ; decrement count
        jnz     l1
jmp main

case3:
print m3
        mov     ax, a           ; load number1 in ax
        mov     bx, b           ; load number2 in bx
        mul     bx              ; multiply numbers. result in dx and ax
        mov     si, ax
        mov     bx, dx          ; result in reg bx
        mov     dh, 2
l5:     mov     ch, 04h         ; count of digits to be displayed
        mov     cl, 04h         ; count to roll by 4 bits
l6:     rol     bx, cl          ; roll bx so that msb comes to lsb
        mov     dl, bl          ; load dl with data to be displayed
        and     dl, 0fh         ; get only lsb
        cmp     dl, 09          ; check if digit is 0-9 or letter a-f
        jbe     l7
        add     dl, 07          ; if letter add 37h else only add 30h
l7:     add     dl, 30h
        mov     ah, 02          ; function 2 under int 21h (display character)
        int     21h
        dec     ch              ; decrement count
        jnz     l6
        dec     dh
        cmp     dh, 0
        mov     bx, si
        jnz     l5
```

```
jmp main
case4:
print m4
      mov     ax,a          ; load number1 in ax

mov dl,00
mov dh,00
       mov     bx, b          ; load number2 in bl
       div     bx             ; divide numbers. quotient in al and rem in ah
       mov     ch, 04h        ; count of digits to be displayed
       mov     cl, 04h        ; count to roll by 4 bits
       mov     bx, ax              ; result in reg bx
lo1:   rol      bx, cl               ; roll bx so that msb comes to lsb
       mov     dl, bl              ; load dl with data to be displayed
       and     dl, 0fh              ; get only lsb
       cmp     dl, 09h              ; check if digit is 0-9 or letter a-f
       jbe     lo3
       add     dl, 07h              ; if letter add 37h else only add 30h
lo3:   add     dl, 30h
       mov     ah, 02h              ; function 2 under int 21h (display character)
       int     21h
       dec     ch                  ; decrement count
       jnz     lo1

jmp main
end start
```

Output:



.

5. Conclusion :-