INDEX

Serial No	Exp. No	Experiment Name	Date	Page No	Signature
01	01	An assembly program that can take input from user and addition two number.	24.09.22	1-4	
02	02	An assembly program that can take input from user and print as output	1.10.22	5-8	
03	03	write an assembly lang. program to convert upper to lower case and lower to upper case letter.	08.10.22	9-12	
04	04	write an assembly lang. program to divide two number.	08.10.22	13-16	

Problem No :- 01

Problem Name :- Implement Addition of two numbers in assembly language.

Theory :- We use add instruction for addition

ADD instruction adds an immediate data or contents of a memory location specified in the Instruction or a register to the contents of another register or memory location. The result is in the destination operand. That means memory to memory addition is not possible. Also the contents of the segment registers cannot be added using this instruction. All the condition code flags are affected depending upon the result. Hence one of the operands is initally moved to AL OR AX. Then using the add, 8 bit or 16bit addition is performed.

Apparatus :

(i) 8086 microprocessor kit/erau

(ii) Power supply.

Procedure :

1. Load contents of memory location 2050 in register AL

2. Copy content of register AL to register AL to register AH

3. Assign 0004 to cx register pair.

4. Do AND operation on content of AL with OF and store result in AL.

5. Rotate the contents of AH by executing ROL instruction using cx

6. Do AND operation on content of AH with OF and store result in AH

7. Add AL and AH content and store

result in AL
8. Store the content of AL in memory
location 2051

Source code :-

emu8086 - assembler and microprocessor emulator 4.08

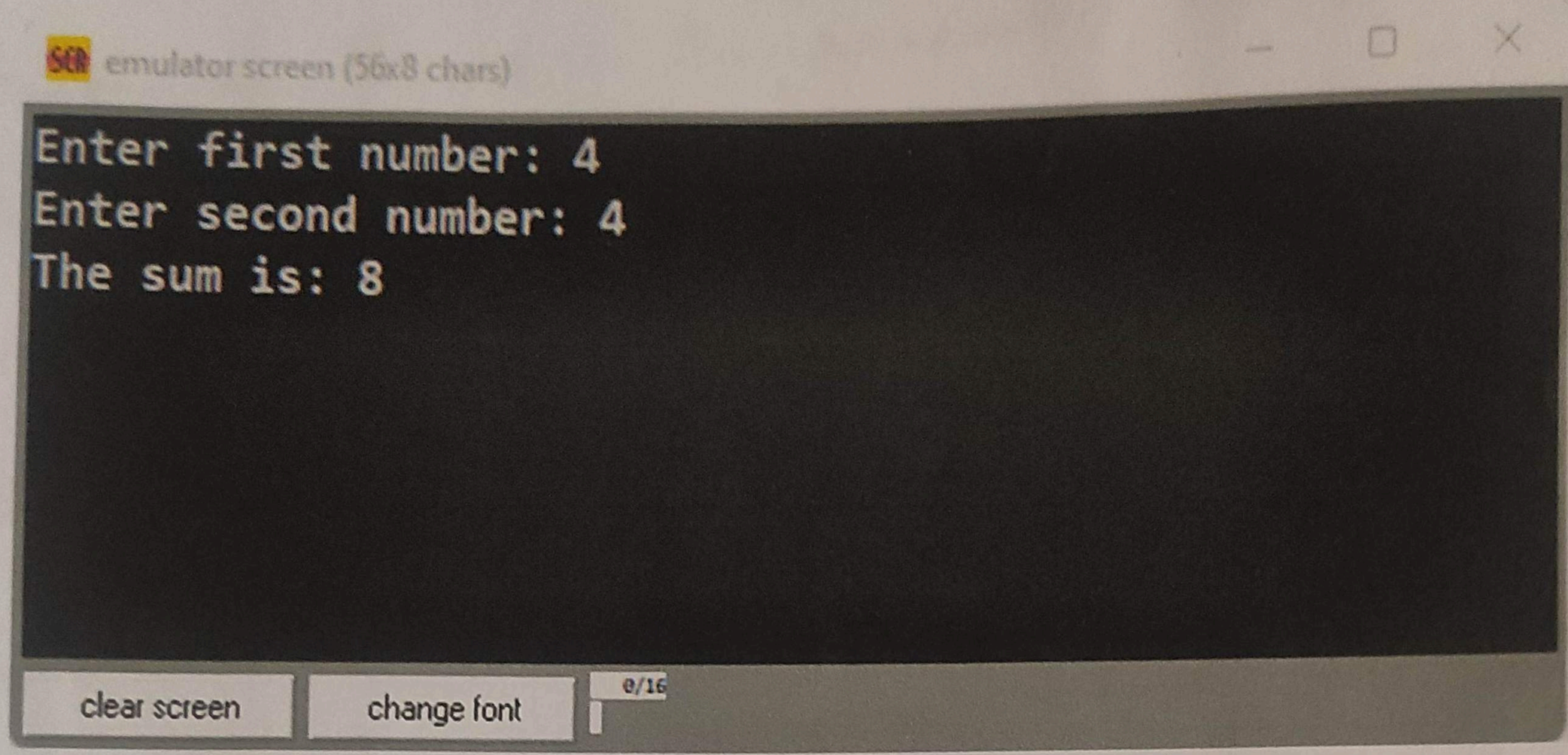
file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
.model small
.data
str1 db "Enter first number: $"
str2 db 13,10,"Enter second number: $"
str3 db 13,10,"The sum is: $"
.code
main proc
    mov ax,@data
    mov ds,ax
    lea dx,str1
    mov ah,9
    int 21h
    mov ah,1
    int 21h
    mov bl,al
    lea dx,str2
    mov ah,9
    int 21h
    mov ah,1
    int 21h
    mov bh,al
    sub bh,48
    sub bl,48
    lea dx,str3
    mov ah,9
    int 21h
    add bh,bl
    add bh,48
    mov dl,bh
    mov ah,2
    int 21h
    mov ah,4Ch
    int 21h
main endp
end
```

line: 27 col: 39 drag a file here to open

Output:-



Discussion:- In the above addition program we used al, bl register because they are 16-bit register and we can adding 16-bit number, we have to use to x register. In this data section, we used variable type DB means "data type". It is used for an 8-bit variable. In data segment, we have initialized two variables a and b.

Problem No: 02

Problem Name: An assembly program that can take input from user and print as output.

Theory: In this experiment we will be introduced to the basic input and output operation using assembly language. we will use DOS interrupt (INT 21H) function calls to access the keyboard and video display. More details will also be given on the structure of assembly languge program.

Apparatus:

(i) 8086 microprocessor kit/emu

(ii) Power supply.

Procedure:

1. Load contents of memory location 2050

2. Copy content of register ax to register ah.

3. Move content al register to bl register

4. Assign dx register.

5. Move content bl register + dl register

6. Copy the register ah.

7. Printer the register.

8. Stop.

Source code :

emu8086 - assembler and microprocessor emulator 4.08

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
.model small
.data
str1 db "Enter the number: $"
str2 db 13,10,"The number: $"

.code
main proc
    mov ax,@data
    mov ds,ax

    lea dx,str1
    mov ah,9
    int 21h

    mov ah,1
    int 21h
    mov bl,al

    lea dx,str2
    mov ah,9
    int 21h

    mov dl,bl
    mov ah,2
    int 21h

    mov ah,4Ch
    int 21h

main endp
end
```

line: 23 col: 14 drag a file here to open

Output :

emulator screen (34x4 chars)

```
Enter the number: 4
The number: 4
```

clear screen change font 0/16

Discussion :- The program can only display a single digit inputted by the user. It also shown a message before taking in input & displaying output. The message in written inside ""(double quotation) and ended with ; sign which indicates end of a string.

Problem No: 03

Problem Name: Write an assembly language Program to convert an uppercase letter to a lowercase letter & lower to upper case.

Theory: To convert an uppercase letter to lowercase, we note that ASCII codes for the uppercase letter 'A' to 'Z' from a sequence from 65 to 90. The corresponding lowercase letters 'a' to 'z' have codes in sequence from 97 to 122, we say that ASCII codes form a collating sequence and we use this fact to sort textual information into alphabetical order. To convert from an uppercase character to its lowercase, we add 32 to the ASCII code. And lowercase to uppercase character we sub 32 to the ASCII code.

Apparatus:

(i) 8086 microprocessor kit / emu

(ii) Power supply.

Procedure :-

(i) Get data segment and initailize DS.

(ii) Get first message by DX register and display string function by AH register

(iii) Read the character function

(iv) Read a small letter in AL register.

(v) Convert it and store it.

(vi) Again read the second message

(vii) Read the character function

(viii) Read a upper letter in AL register

(ix) Convert it small an store it

(x) Stop the program.

Source code :-

edit: G:\upper case & lower case.asm

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
.MODEL SMALL
.STACK 100H
.DATA
msg1 db "Enter Upper Case: $"
msg2 db 10,13, "Output Case: $"
msg3 db 10,13, "Enter Lower Case: $"
msg4 db 10,13, "Output Case: $"
var1 db ?
.CODE
MAIN PROC
      MOV AX ,@DATA
      MOV DS , AX
      LEA DX, msg1
      MOV AH, 9
      INT 21H
      MOV AH,1
      INT 21H
      MOV var1,AL
      MOV BL, var1
      ADD BL, 20H
      LEA DX, msg2
      MOV AH, 9
      INT 21H
      MOV AH,2
      MOV DL,BL
      INT 21H
      LEA DX, msg3
      MOV AH, 9
      INT 21H
      MOV AH,1
      INT 21H
      MOV var1,AL
      MOV BL,var1
      SUB BL, 20H
      LEA DX, msg4
      MOV AH, 9
      INT 21H
      MOV AH,2
      MOV DL,BL
      INT 21H
      EXIT:
      MOV AH,4CH
      INT 21H
      MAIN ENDP
END MAIN
```

drag a file here to open

line: 3 col: 6

emulator screen (40x6 chars)

```
Enter Upper Case: F
Output Case: f
Enter Lower Case: n
Output Case: N
```

clear screen | change font | 0/16

Discussion:- To convert upper to its lower case, need add 32 to the ASCII code and lower to upper convert, need ~~add~~ subtract 32. The number 32 is obtained by subtracting the ASCII code for 'A' from the ASCII code for 'a' (i.e. 'A' → 'a' = 97 - 65 = 32)

Problem No: 04

Problem Name: Write an assembly language Program to divide two numbers in 8086.

Theory: The division operation two elements - a quotient and a remainder. In case of multiplication, overflow does not occur because double-length registers are used to keep the product. However, in case of division, overflow may occur. The processor generates an interrupt if overflow occurs. The Div instruction is used for unsigned data and the IDIV is used for signed data.

Apparatus:

(i) 8086 microprocessor kit lemu

(ii) Power supply

Procedure :-

1. Start the program by loading the HL pair registers with address of memory location.

2. Move the data to B Register.

3. Load the second data into accumulator.

4. Compare the two numbers to check carry.

5. Subtract two numbers.

6. Increment the value of carry.

7. Check whether the repeated subtraction in over.

8. Then store the results (quotient and remainder) in given memory location.

9. Terminate the program.

Source Code :-

edit: G:\mycode.asm

File edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
.model small
.data
n1 dw 20d
n2 db 5d
msg3 db "Questiont is:$"
msg4 db 10d,"Remainder is:$"
que db 0
rem db 0
.code

mov ax,@data
mov ds,ax
mov ax,n1
mov bl,n2
div bl
mov que,al
mov rem,ah
mov ah,09h
mov dx,offset msg3
int 21h
mov dl,que
add dl,48d
mov ah,2
int 21h
mov ah,09h
mov dx,offset msg4
int 21h
mov dl,rem
add dl,48d
mov ah,2
int 21h

mov ah,4ch
int 21h
end
```

line: 10 col: 1 drag a file here to open

output:

emulator screen (38x5 chars)

```
Questiont is:4
                Remainder is:0
```

clear screen change font 0/16

Discussion of Div executes unsigned division div divides a 16-, 32-, or 64-bit register value by a register or memory byte, word, or long (divisor). The quotient is stored in the AL, AX, or EAX register respectively. The remainder is stored in AH, DX, or EDX. The size of the divisor (8, 16, or 32 bit operand) determines the particular register used as the dividend.