

COMPUTER ORGANIZATION AND ARCHITECTURE

NAME: VAYYA VISHNUPRIYA

REG: AP22110010390

CLASS: CSE-F

LAB TASK 5

1. Write an assembly language program to perform division of 8-bit data.

CODE:

```
org 100h

; Initialize values

mov al,96h    ; Move 89h (137 in decimal) into AL
mov bl,10h    ; Move 10h (16 in decimal) into BL

; Perform division (AL / BL)

idiv bl      ; AL = quotient, AH = remainder
mov bl,al    ; Store quotient in BL
mov bh,ah    ; Store remainder in BH

; Convert first digit (quotient) to ASCII

and al,0f0h  ; Mask higher nibble of AL
shr al,4     ; Shift right 4 bits to get the first hex digit
add al,30h   ; Convert to ASCII (0-9)
cmp al,39h   ; Check if it's a number or letter (0-9)
jle print_first_digit1
```

```
add al,7    ; Convert to ASCII (A-F)
```

```
print_first_digit1:
```

```
    mov dl,al    ; Move the result to DL (for printing)
```

```
    mov ah,02h   ; Print function
```

```
    int 21h      ; Interrupt to print the character
```

```
; Convert second digit (quotient) to ASCII
```

```
mov al,bl    ; Move the quotient back into AL
```

```
and al,0fh   ; Mask the lower nibble of AL
```

```
add al,30h   ; Convert to ASCII (0-9)
```

```
cmp al,39h   ; Check if it's a number or letter (0-9)
```

```
jle print_second_digit1
```

```
add al,7     ; Convert to ASCII (A-F)
```

```
print_second_digit1:
```

```
    mov dl,al    ; Move the result to DL (for printing)
```

```
    mov ah,02h   ; Print function
```

```
    int 21h      ; Interrupt to print the character
```

```
; Print remainder (remainder is in BH)
```

```
; Convert first digit (upper nibble of remainder) to ASCII
```

```
mov al,bh    ; Move remainder into AL
```

```
and al,0f0h  ; Mask the higher nibble
```

```
shr al,4     ; Shift right 4 bits to get the first hex digit
```

```
add al,30h    ; Convert to ASCII (0-9)
cmp al,39h    ; Check if it's a number or letter (0-9)
jle print_first_rem_digit
add al,7      ; Convert to ASCII (A-F)
```

print_first_rem_digit:

```
    mov dl,al  ; Move the result to DL (for printing)
    mov ah,02h ; Print function
    int 21h    ; Interrupt to print the character
```

; Convert second digit (lower nibble of remainder) to ASCII

```
mov al,bh    ; Move remainder back into AL
and al,0fh   ; Mask the lower nibble
add al,30h   ; Convert to ASCII (0-9)
cmp al,39h   ; Check if it's a number or letter (0-9)
jle print_second_rem_digit
add al,7     ; Convert to ASCII (A-F)
```

print_second_rem_digit:

```
    mov dl,al  ; Move the result to DL (for printing)
    mov ah,02h ; Print function
    int 21h    ; Interrupt to print the character
```

OUTPUT:



2. Write a program in assembly language to perform division of 16-bit data.

CODE:

```
Org 100h  
mov ax,1780h  
mov bx,1000h  
div bx  
mov bx,ax  
mov cx,dx  
mov ah,ch  
and ah,0f0h  
shr ah,4
```

```
add ah,30h
cmp ah,39h
jle print_high_nibble32
add ah,7
print_high_nibble32:
    mov dl,ah
    mov ah,02h
    int 21h
mov ah,ch
and ah,0fh
add ah,30h
cmp ah,39h
jle print_low_nibble32
add ah,7
print_low_nibble32:
    mov dl,ah
    mov ah,02h
    int 21h

mov ah,cl
and ah,0f0h
shr ah,4
add ah,30h
cmp ah,39h
jle print_low_nibble24
add ah,7
```

print_low_nibble24:

mov dl,ah

mov ah,02h

int 21h

mov ah,cl

and ah,0fh

add ah,30h

cmp ah,39h

jle print_high_nibble24:

add ah,7

print_high_nibble24:

mov dl,ah

mov ah,02h

int 21h

mov ah, bh

shr ah, 4

add ah, 30h

cmp ah, 39h

jle print_high_nibble

add ah, 7

print_high_nibble:

mov dl, ah

mov ah, 02h

int 21h

mov ah, bh

and ah, 0fh

add ah, 30h

cmp ah, 39h

jle print_low_nibble

add ah, 7

print_low_nibble:

mov dl, ah

mov ah, 02h

int 21h

mov ah, bl

shr ah, 4

add ah, 30h

cmp ah, 39h

jle print_high_nibble2

add ah, 7

print_high_nibble2:

mov dl, ah

mov ah, 02h

int 21h

mov ah, bl

and ah, 0fh

add ah, 30h

cmp ah, 39h

jle print_low_nibble2

add ah, 7

print_low_nibble2:

mov dl, ah

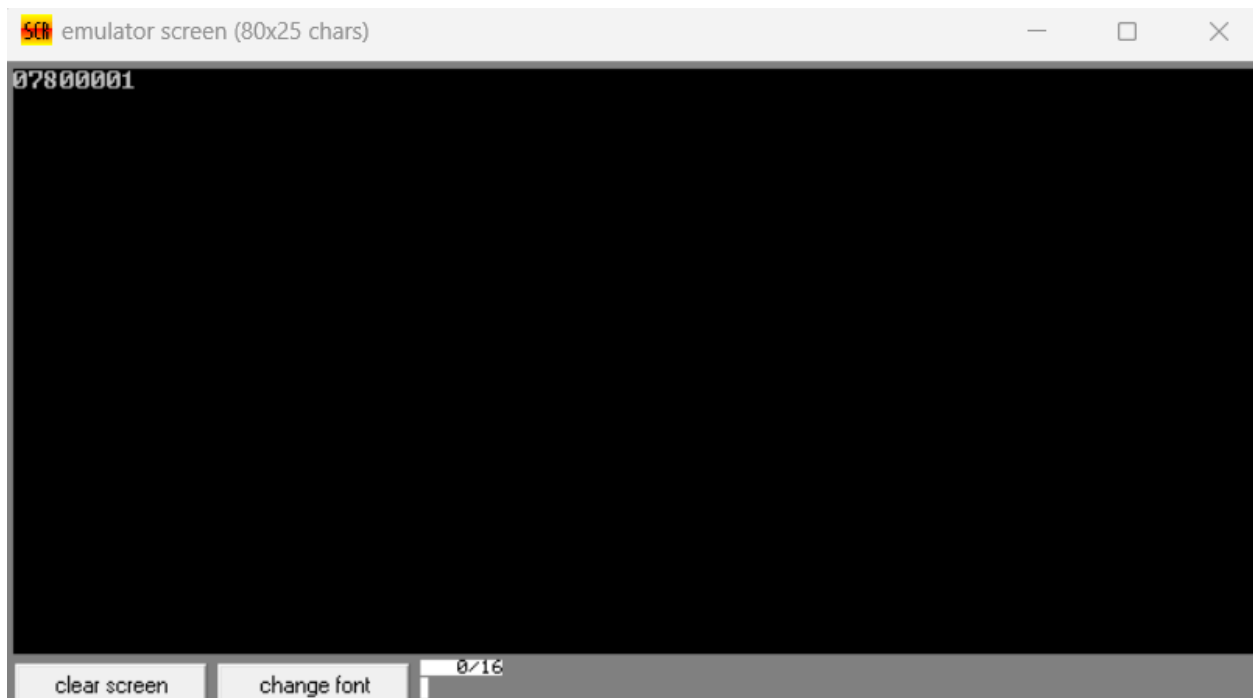
mov ah, 02h

int 21h

mov ah, 4ch

int 21h

OUTPUT:



GITHUB LINK: <https://github.com/vishnupriyavayya/COA-LAB-TASK-5>

The screenshot shows a GitHub repository page for 'COA-LAB-TASK-5' by user 'vishnupriyavayya'. The repository is public and has 1 branch and 0 tags. The main branch is 'main'. The repository contains two files: 'mycode1.lab task 5.asm' and 'mycode2.lab task 5.asm', both added via upload. The repository has 45061dc as the latest commit, made 'now', with 1 commit in total. The repository is currently empty, with a message 'Add a README' and a button 'Add a README'. The right sidebar shows the 'About' section with no description, website, or topics provided. It also shows 'Activity' with 0 stars, 1 watching, and 0 forks. The 'Releases' section shows no releases published, with a link to 'Create a new release'. The 'Packages' section shows no packages published, with a link to 'Publish your first package'.

github.com/vishnupriyavayya/COA-LAB-TASK-5

COA-LAB-TASK-5 Public

main 1 Branch 0 Tags

Go to file Add file Code

vishnupriyavayya Add files via upload 45061dc · now 1 Commit

mycode1.lab task 5.asm Add files via upload now

mycode2.lab task 5.asm Add files via upload now

README

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README

About

No description, website, or topics provided.

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package