

COMPUTER ORGANIZATION AND ARCHITECTURE – LAB TASK 4

NAME: VAYYA VISHNUPRIYA

REG NO: AP22110010390 (CSE-F)

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

```
org 100h    ; Set starting address
```

```
mov al, 03h    ; Load AL with 03h
```

```
mov bl, 03h    ; Load BL with 03h
```

```
mul bl        ; Multiply AL by BL, result in AX (AL * BL)
```

```
mov bl, al     ; Move result (AL) to BL for later use
```

```
mov ah, al     ; Move AL to AH
```

```
; Convert upper nibble to ASCII
```

```
and ah, 0F0h   ; Mask lower nibble, keep upper
```

```
shr ah, 4      ; Shift upper nibble to lower position
```

```
add ah, 30h    ; Convert to ASCII '0'-'9'
```

```
cmp ah, 39h    ; Compare with '9'
```

```
jle print_first_digit ; If less or equal to '9', skip next step
```

```
add ah, 7      ; Convert to ASCII 'A'-'F'
```

```
print_first_digit:
```

```
mov dl, ah     ; Move first digit to DL
```

```
mov ah, 02h    ; Prepare for output
```

```
int 21h       ; Print first digit
```

; Convert lower nibble to ASCII

mov ah, bl ; Move result (BL) back to AH

and ah, 0Fh ; Mask upper nibble, keep lower

add ah, 30h ; Convert to ASCII '0'-'9'

cmp ah, 39h ; Compare with '9'

jle print_sec_digit ; If less or equal to '9', skip next step

add ah, 7 ; Convert to ASCII 'A'-'F'

print_sec_digit:

mov dl, ah ; Move second digit to DL

mov ah, 02h ; Prepare for output

int 21h ; Print second digit

mov ah, 4Ch ; Prepare for program termination

int 21h ; Terminate program

OUTPUT:



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

```
org 100h      ; Set starting address
```

```
mov ax,0013h ;Load AX with 5678h
```

```
mov bx,0013h  ; Load BX with 1234h
```

```
mul bx        ; Multiply AX by BX, result in DX:AX
```

```
mov bx, ax     ; Move the lower 16 bits of the result (AX) into BX
```

```
; Convert and print the high nibble of BH
```

```
mov ah, bh     ; Move BH (high byte of BX) to AH
```

```
shr ah, 4      ; Shift right to isolate the high nibble
```

```
add ah, 30h    ; Convert to ASCII '0'-'9'
```

```
cmp ah, 39h    ; Compare with '9'
```

```
jle print_high_nibble ; If less than or equal, skip next step
```

```
add ah, 7       ; Adjust to ASCII 'A'-'F'
```

```
print_high_nibble:
```

```
mov dl, ah     ; Move the ASCII value to DL
```

```
mov ah, 02h    ; Set up for printing
```

```
int 21h        ; Print the high nibble of BH
```

```
; Convert and print the low nibble of BH
```

```
mov ah, bh     ; Move BH back to AH
```

```
and ah, 0fh    ; Mask the high nibble, keep the low nibble
```

```
add ah, 30h    ; Convert to ASCII '0'-'9'
```

cmp ah, 39h ; Compare with '9'

jle print_low_nibble ; If less than or equal, skip next step

add ah, 7 ; Adjust to ASCII 'A'-'F'

print_low_nibble:

mov dl, ah ; Move the ASCII value to DL

mov ah, 02h ; Set up for printing

int 21h ; Print the low nibble of BH

; Convert and print the high nibble of BL

mov ah, bl ; Move BL (low byte of BX) to AH

shr ah, 4 ; Shift right to isolate the high nibble

add ah, 30h ; Convert to ASCII '0'-'9'

cmp ah, 39h ; Compare with '9'

jle print_high_nibble2 ; If less than or equal, skip next step

add ah, 7 ; Adjust to ASCII 'A'-'F'

print_high_nibble2:

mov dl, ah ; Move the ASCII value to DL

mov ah, 02h ; Set up for printing

int 21h ; Print the high nibble of BL

; Convert and print the low nibble of BL

mov ah, bl ; Move BL back to AH

and ah, 0fh ; Mask the high nibble, keep the low nibble

add ah, 30h ; Convert to ASCII '0'-'9'

```
cmp ah, 39h    ; Compare with '9'
```

```
jle print_low_nibble2 ; If less than or equal, skip next step
```

```
add ah, 7      ; Adjust to ASCII 'A'-'F'
```

```
print_low_nibble2:
```

```
mov dl, ah     ; Move the ASCII value to DL
```

```
mov ah, 02h    ; Set up for printing
```

```
int 21h       ; Print the low nibble of BL
```

```
; Terminate the program
```

```
mov ah, 4ch    ; Set up for program termination
```

```
int 21h       ; Terminate the program
```

OUTPUT:



GITHUB LINK: <https://github.com/vishnupriyavayya/COA-LAB-TASK-4>

The screenshot shows a GitHub repository page for 'COA-LAB-TASK-4' by user 'vishnupriyavayya'. The repository is public and has 1 branch and 0 tags. The file list includes 'README.md' (Initial commit, 5 days ago), 'mycode1.lab task 4.asm' (Add files via upload, now), and 'mycode2.lab task 4.asm' (Add files via upload, now). The README file is selected, showing the title 'COA-LAB-TASK-4'. The right sidebar contains sections for 'About' (No description, website, or topics provided), 'Releases' (No releases published, Create a new release), and 'Packages' (No packages published, Publish your first package). The footer shows the GitHub logo and copyright information: © 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information.

github.com/vishnupriyavayya/COA-LAB-TASK-4

vishnupriyavayya / COA-LAB-TASK-4

Type to search

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

COA-LAB-TASK-4 Public

Pin Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags

Go to file Add file <> Code About

vishnupriyavayya Add files via upload 6603880 · now 2 Commits

README.md	Initial commit	5 days ago
mycode1.lab task 4.asm	Add files via upload	now
mycode2.lab task 4.asm	Add files via upload	now

README

COA-LAB-TASK-4

No description, website, or topics provided.

- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information