|  |
| --- |
| FAST National University |
| **Matrix Operations**  **Midterm** |
|  |

**Computer Organization and Assembly Language**

|  |  |
| --- | --- |
| **Student Name** | Umamah Hussain |
| **Registration #** | 21L-1858 |
| **Instructor** | Hazoor Ahmad |
| **Class** | CS-A |
| **Section** | A |
| **Semester** | Fall 2022 |

Fast School of Computing

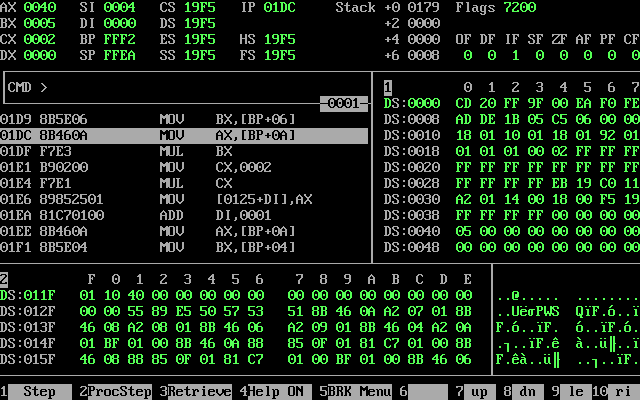
FAST-NU, Lahore, Pakistan

# Activity 1

## **Assembly Language Code**

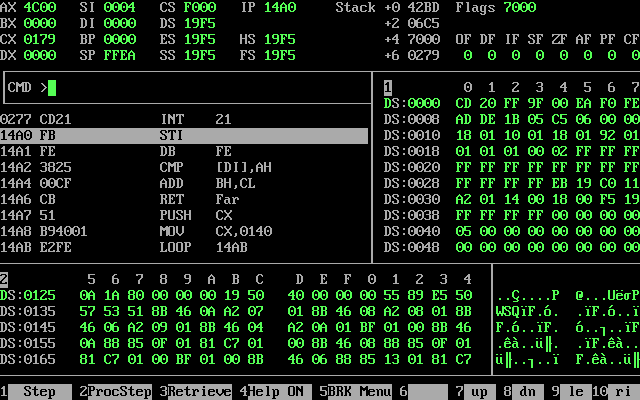
[org 0x0100]  
jmp start  
roll: db 1,8,5,8  
a: db 0  
b:db 0  
c:db 0  
d:db 0  
mr1:db 0,0,0,0  
mr2:db 0,0,0,0  
mr3:db 0,0,0,0  
mr4:db 0,0,0,0  
nr1: db 0,0  
nr2: db 0,0  
or1:dw 0,0,0  
or2: dw 0 ,0,0  
or3:dw 0,0,0  
loadrollno:  
push bp  
mov bp,sp  
push ax  
push di  
push bx  
push cx  
push es  
mov ax,[bp+10]  
mov byte [a],al  
mov ax,[bp+8]  
mov byte [b],al  
mov ax,[bp+6]  
mov byte [c],al  
mov ax,[bp+4]  
mov byte [d],al  
mov di,1  
movinm2:  
mov ax,[bp+10]  
mov byte [mr2+di],al  
add di,1  
mov ax,[bp+8]  
mov byte [mr2+di],al  
add di,1  
mov di,1  
  
movinm3:  
mov ax,[bp+6]  
mov byte [mr3+di],al  
add di,1  
mov ax,[bp+4]  
mov byte [mr3+di],al  
mov di,0  
movinn1:  
mov ax,[bp+4]  
mov byte [nr1+di],al  
add di,1  
mov ax,[bp+6]  
mov byte [nr1+di],al  
mov di,0  
movinn2:  
mov ax,[bp+8]  
mov byte [nr2+di],al  
add di,1  
mov ax,[bp+10]  
mov byte [nr2+di],al  
mov di,0  
dotproductor1:  
mov ax,[bp+10]  
mul ax  
mov [or1+di],ax  
  
mov ax,[bp+10]  
mov bx,[bp+8]  
mul bx  
mov cx,2  
mul cx  
add di,1  
mov [or1+di],ax  
mov ax,[bp+8]  
mul bx  
add di,1  
mov [or1+di],ax  
dotproductor2:  
mov di,0  
mov bx,[bp+6]  
mov ax,[bp+10]  
mul bx  
mov cx,2  
mul cx  
mov [or2+di],ax  
add di,1  
mov ax,[bp+10]  
mov bx,[bp+4]  
mul bx  
mov cx,ax  
mov ax,[bp+8]  
mov bx,[bp+6]  
mul bx  
add bx,cx  
mov ax,bx  
mov cx,2  
mul cx  
mov [or2+di],ax  
add di,1  
mov ax,[bp+8]  
mov bx,[bp+4]  
mul bx  
mov cx,2  
mul cx  
mov [or2+di],ax  
dotproduct3:  
mov di,0  
mov ax,[bp+6]  
mov bx,[bp+6]  
mul bx  
mov [or3+di],ax  
add di,1  
mov ax,[bp+6]  
mov bx,[bp+4]  
mul bx  
mov cx,2  
mul cx  
mov [or3+di],ax  
add di,1  
mov ax,[bp+4]  
mov bx,[bp+4]  
mul bx  
mov [or3+di],ax  
print:  
mov ax,0xb800  
mov es,ax  
mov si,0  
mov di,2  
ho:  
mov cx,[or1+si]  
add cx,0x30  
mov word[es:di],cx  
add si,2  
add di,2  
cmp si,6  
jne ho  
mov si,0  
hho:  
mov cx,[or2+si]  
add cx,0x30  
mov word[es:di],cx  
add si,2  
add di,2  
cmp si,6  
jne hho  
mov si,0  
mov di,320  
  
hhho:  
mov cx,[or3+si]  
add cx,0x30  
mov word[es:di],cx  
add si,2  
add di,2  
cmp si,6  
jne hhho  
  
  
pop cx  
pop bx  
pop di  
pop ax  
pop bp  
ret  
  
start:  
mov si,0  
mov ax,0  
l1:  
  
mov al,[roll+si]  
push ax  
add si,1  
cmp si,4  
jne l1  
  
call loadrollno  
mov ax, 0x4c00 ; terminate program  
int 0x21

## **Debugging Screenshots**



**ABOVE** **Screenshot Posted of OR1 only**

**THE BELOW SS IS OF OR2**



**THE BELOW SCREENSHOT IS OF OR3**

