***AIDS MICROPROCESSOR LAB S21 BATCH (2023-24)***

***Experiment 1 Title: Assembly language programming for MOV instruction in various addressing modes using software tool TASM 1.4***

***Name of student: Meet Raut Class Roll Number: 2201084***

***Date of Performance: 11/03/2024***

***Batch: S2-1 Timing: 3:00-5:00 Date of Submission: 05/02/2024***

***Assembly language code***

*data segment # Data segment*

*n1 db 15h # variable n1 define 8 bit number*

*n2 dw 1367h # variable n2 define 16 bit number*

*n3 dw 0 #Initialize n3 to 0*

*n4 dw 0 #unused variable*

*arr1 db 00h,12h,23h,34h,*

*45h,56h,67h,78h,89h,92h #Array arr1*

*arr2 db 20h,21h,22h,23h,*

*24h,25h,26h,27h,28h,29h #Array arr2*

*arr3 db 10 DUP(0) #Array arr3 with 10 elements initialized to 0*

*.*

*data ends # Assemble directives*

*code segment*

*assume cs:code, ds:data # Assemble directives defining cs and ds*

*start: # start of code segment*

*mov ax,data # moving data to ax register- initialization process*

*mov ds,ax # moving ax to dx register -- initialization process*

*;IMMEDIATE ADDRESING MODE*

*mov al,34h # Load immediate value 34h to al register*

*mov cx,1257h # Load immediate value 1257h to cx register*

*;REGISTER ADDRESSING MODE*

*mov ah,al # Copy the value from al to ah register*

*mov dx,cx # Copy the value from cx to dx register*

*;DIRECT ADDRESSING MODE*

*mov al,n1 #Load the value of variable n1 into al register*

*mov bx,n2 #Load the value of variable n2 into bx register*

*mov n3,al #Copy the value of al register to variable n3*

*mov bx,OFFSET arr1 #Load the offset address arr1 into bx register*

*mov si,OFFSET arr2 #Load the offset address arr2 into si register*

*mov di,OFFSET arr3 #Load the offset address arr3 into di register*

*;INDIRECT ADDRESSING MODE*

*mov cl,[bx] #Load the value at the address stored in bx into cl register*

*mov ch,[si] #Load the value at the address stored in si into ch register*

*mov [di],ch #Store the value of ch in the address stored in di*

*;BASE ADDRESSING MODE*

*mov dl,3[bx] #Load the value at the address (bx + 3) into dl register*

*;INDEXED ADDRESSING MODE*

*mov 7[di],dl #Store the value of dl at the address (di + 7)*

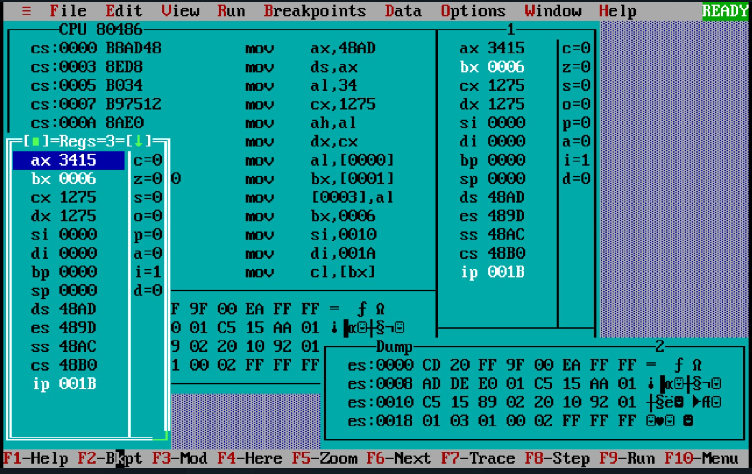
*mov ah,4ch #Set ah register for exit code*

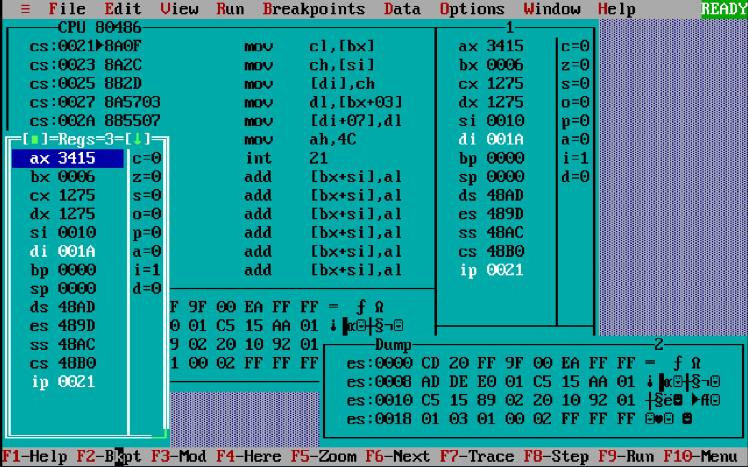
*int 21h # breakpoint interrupt*

*code ends # Assembler directives to end code*

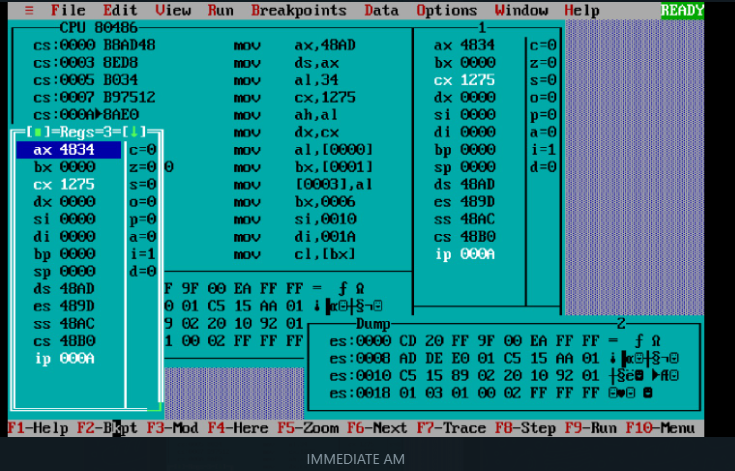
*end start # Assembler directives to end start*

*Result :*

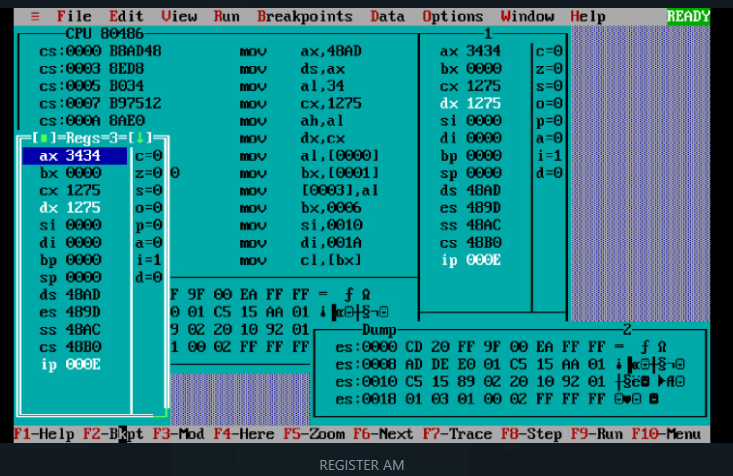




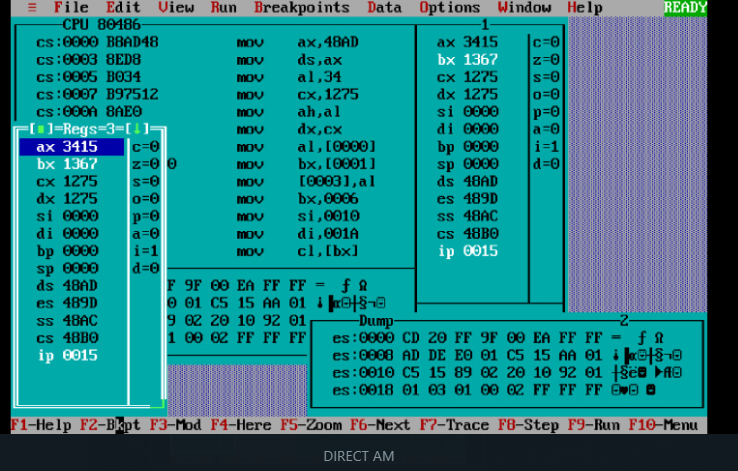
**(i) IMMEDIATE ADDRESSING MODE:**

****

**(ii) REGISTER ADDRESSING MODE:**

****

**(iii) DIRECT ADDRESSING MODE:**

****

***CONCLUSION: LO 1, LO 3 mapped.***

***---------------------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-----------------------------------***