

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

**Experiment 8 Title: Assembly language programming to display content of flag register**

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

***Date of Performance: 01/04/2024***

**Batch: S2-1**                      **Timing: 3:00-5:00**                      **Date of Submission: 01/04/2024**

## Assembly language code

DATA\_SEG SEGMENT

*CHAR DB 17H*

MSG1 DB '-- -- -- -- OF DF IF TF SF ZF -- AX -- PF -- CF',0AH,'\$'

*FLAG DW ?*

DATA\_SEG ENDS

CODE\_SEG SEGMENT

*ASSUME CS:CODE\_SEG, DS:DATA\_SEG*

*START:*

```
MOV AX,DATA_SEG
```

MOV DS,AX #INITIALIZE DATA SEGMENT

*MOV AH,09H*                      #09 SERVICE

<i>LEA DX,MSG1</i>	<i>#ADDRESS OF MSG1 IS STORED IN DX</i>
--------------------	---

*INT 21H* *#DISPLAYING THE LINE*

MOV AL,0FFH #DISPLAY FLAG REGISTER

```
ADD AL,02H          #MAKING CARRY FLAG 1
```

CLD #CLEARING DIRECTION FLAG

*STI* *#SETTING INTERRUPT FLAG*

*PUSHF*                    *#PUSHING THE CONTENT OF FLAG REGISTER ONTO STACK*

*POP BX*                      #*POPING CONTENTS INTO BX REGISTER*

MOV FLAG,BX #SAVED TO MEMORY

*MOV CX,16*  
*MOV BX,8000H*

*#INITALIZE COUNTER*  
*#STORING 1000 0000 0000 0000*

*FLAG\_LOOP:*

*MOV AX,FLAG*

*#LOAD FROM MEMORY*

*AND AX,BX*

*#GETTING REQUIRED BIT*

*JZ DISP\_0*

*#IF BIT IS ZERO JUMP TO DISPLAY*

*MOV DL,31H*

*#STORING 1 IN DL*

*MOV AH,02H*

*#02H SERVICE*

*INT 21H*

*#DISPLAY THE LINE*

*MOV DL,' '*

*INT 21H*

*INT 21H*

*#DISPLAYING THE SPACES*

*JMP NEXT*

*#JUMP TO NEXT*

*DISP\_0:*

*MOV DL,30H*

*#STORING 0 IN DL*

*MOV AH,02H*

*#02H SERVICE*

*INT 21H*

*#DISPALY*

*MOV DL,' '*

*INT 21H*

*INT 21H*

*#DISPLAYING SPACES*

*NEXT:*

*ROR BX,1*

*#ROTATE RIGHT BX REGISTER BY 1*

*LOOP FLAG\_LOOP*

*#LOOP FOR ALL BITS*

*MOV AH,4CH*

*INT 21H*

*#EXIT TO DOS*

*CODE\_SEG ENDS*

*END START*

**Result:**

```
GUI Turbo Assembler x64
-- -- -- -- DF DF IF TF SF ZF -- AX -- PF -- CF
0 1 1 1 0 0 1 0 0 0 0 1 0 0 1 1

Press any key to exit...
```

**CONCLUSION:** *LO 2, LO 3 mapped.*

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