Name : VIRENDRA · VIJAY · PALEKAR
PRN : 201041045
Subject: MICRORROCESSOR

EXPERIMENT – 4

- AIM : Assembly program to display the contents of the flag register.
- HARDWARE/SOFTWARE REQUIRED : MASM / TASM 5.0
- EXPLANATION : To display the contents of flag registers pushf and pop instruction. Each bit of flag register is then masked off with 1 and all 0's (i.e 1000 0000 0000 0000 (16 bit) à 8000h) and based on the result of masking either 0 (30h) or 1 (31h) is get displayed on the screen. Each bit of the above 16 bit number gets shifted in right direction by 1 position before making to obtain the next bit position of flag register. This whole procedure gets repeated 16 times.
- ALGORITHM :

1 Start
2 Initialize data segment through AX register in the DS register.
3 Display the flag bit names as "XXXXODITSF ZF X AF X PF X CF"
   a. DO ANY ARITHMETIC OPERATION ADDITION / SUB

```
MOV AX, 1234h
ADD AX, OFFFFH ; CF AF, SF
```

Name : VIRENDRA · VIJAY · PALEKAR
PRN : 201041045

MOV AX, 1234h
SUB AX, 1234h : 2f 5f

4 Push the contents of flag register to a stack.
5 Pop the contents of stack to register to any 16 bit register (say BX = 0000 0100 1000 1001)
6 Move the contents of BX to temporary variable say t
7 Move the 8000h number to AX. (AX ← 8000h)
8 Move the count as 16 to CX register.
9 Move the contents of temporary variable t to BX
10 And the contents of BX and AX
11 If zero flag is set the goto the step no 14 otherwise goto step no 12
12 Move the 31h to DL register
13 Make the unconditional jump to step no 15
14 Move the 30h to DL register.
15 Preserve the (8000h) number from AX in t1 temporary variable
16 Display the contents of DL register.
17 Move the contents of t1 to AX register back (As while displaying 30h or 31h AH register get modified as 02h function is moved of INT 21h)
18 Rotate the contents of AX by 1 positions in right direction.
19 Repeat step no 5 to 17 till count CX reaches to 0.
20 Stop.

Name: Virendra Vijay Palekar

PRN: 201041045

Sub: Microprocessor

EXPERIMENT NO.4

Program:

```
Data Segment
msg db 0dh,0ah,"-- -- -- -- OF DF IF TF SF ZF -- AF -- PF -- CF $"
newl db 0dh,0ah,"$"
flag dw ?
Data ends

Code Segment
assume CS:Code,DS:Data
start:
 mov ax,Data
 mov DS,ax


Mov ax,3345H
mov bx,99H
add ax,bx


 mov dx,offset msg
 mov ah,09h
```

```
 int 21h

 mov dx,offset newl
 mov ah,09h
 int 21h

; Cli ;
; stc
; std
 ;or

;MOV AX,1234h
;ADD AX,0FFFFH  ; CF AF, SF
;MOV AX,1342h
;SUB AX,1342H  ; zf sf
 pushf

 pop bx

 mov flag,bx

 mov cx,16
 mov bx,8000h

loops:
 mov ax,flag
```

```
 and ax,bx
 jz zero
 mov dl,31h
 mov ah,02h
 int 21h
 jmp space

zero: mov dl,30h
 mov ah,02h
 int 21h

space: mov dl,' '
 mov ah,02h
 int 21h

 mov ah,02h
 int 21h
 ror bx,1

 loop loops

 mov ah,4ch
 int 21h
Code ends
end start
```

LST File:

```
Microsoft (R) Macro Assembler Version 6.11                    02/24/22
17:59:27

prog4.asm
                                                              Page 1 - 1


 0000                                                         Data
Segment

 0000 0D 0A 2D 2D 20 2D                                       msg db 0dh,0ah,"--
-- -- -- OF DF IF TF SF ZF -- AF -- PF -- CF $"

      2D 20 2D 2D 20 2D

      2D 20 4F 46 20 44

      46 20 49 46 20 54

      46 20 53 46 20 5A

      46 20 2D 2D 20 41

      46 20 2D 2D 20 50

      46 20 2D 2D 20 43

      46 20 24

 0033 0D 0A 24                                                newl db
0dh,0ah,"$"

 0036 0000                                                    flag dw ?

 0038                                                         Data
ends


 0000                                                         Code
Segment

                                                              assume
CS:Code,DS:Data
```

```
0000                                    start:
0000  B8 ---- R                    mov ax,Data
0003  8E D8                           mov DS,ax


0005  B8 3345                       Mov ax,3345H
0008  BB 0099                       mov bx,99H
000B  03 C3                         add ax,bx


000D  BA 0000 R                    mov dx,offset msg
0010  B4 09                         mov ah,09h
0012  CD 21                         int 21h

0014  BA 0033 R                    mov dx,offset newl
0017  B4 09                         mov ah,09h
0019  CD 21                         int 21h

                                        ; Cli ;
                                        ; stc
                                        ; std
                                         ;or

                                        ;MOV
AX,1234h
                                        ;ADD
AX,0FFFFH  ; CF AF, SF
```

```
                                        ;MOV
AX,1342h
                                        ;SUB
AX,1342H  ; zf sf
001B  9C                                pushf

001C  5B                                pop bx

001D  89 1E 0036 R                      mov flag,bx

0021  B9 0010                           mov cx,16
0024  BB 8000                           mov bx,8000h

0027                                    loops:
0027  A1 0036 R                         mov ax,flag
002A  23 C3                             and ax,bx
002C  74 08                             jz zero
002E  B2 31                             mov dl,31h
0030  B4 02                             mov ah,02h
0032  CD 21                             int 21h
0034  EB 06                             jmp space

0036  B2 30                             zero: mov
dl,30h
0038  B4 02                             mov ah,02h
003A  CD 21                             int 21h
```

```
003C  B2 20                    space: mov dl,' '
003E  B4 02                    mov ah,02h
0040  CD 21                    int 21h

0042  B4 02                    mov ah,02h
0044  CD 21                    int 21h
0046  D1 CB                    ror bx,1

0048  E2 DD                    loop loops

004A  B4 4C                    mov ah,4ch
004C  CD 21                    int 21h
004E                           Code
ends

                               end start
```

Segments and Groups:

N a m e	Size	Length	Align	Combine	Class
Code		16 Bit	004E	Para	Private
Data		16 Bit	0038	Para	Private

Symbols:

N a m e	Type	Value	Attr
flag	Word	0036	Data
loops	L Near	0027	Code
msg	Byte	0000	Data
newl	Byte	0033	Data
space	L Near	003C	Code

start	L Near	0000	Code
zero	L Near	0036	Code

0 Warnings

0 Errors

Output: