**NUST SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE**

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| Faculty Member: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Semester: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Section: \_\_\_\_\_\_\_\_\_\_\_ |

Department of Electrical Engineering

EE- 222: Microprocessor Systems

**LAB 03: CPU Flags**

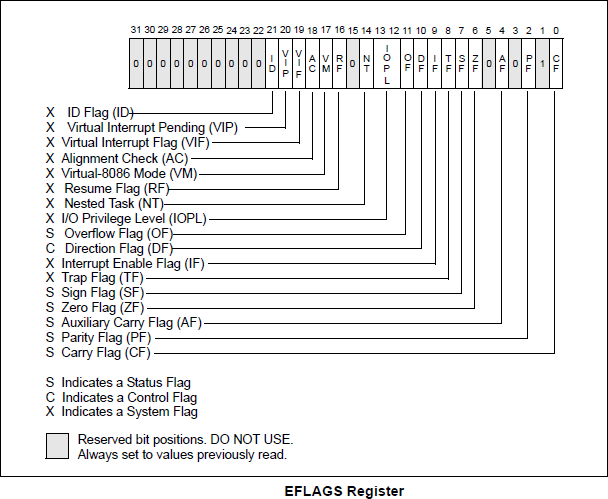
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student name | Reg. No. | Lab Report Marks / 10 | Viva Marks / 5 | Total/15 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Lab No. 3**

**Objective:** The aim of this lab is to learn about flags, their working and input/output in Assembly Language.

**Flags:**

The EFLAGS (or just Flags) register consists of individual binary bits that either control the operation of the CPU or reflect the outcome of some CPU operation.



1. **Status Flags:**

The Status flags reflect the outcomes of arithmetic and logical operations performed by the CPU.

1. The Carry flag (CF) is set when the result of an unsigned arithmetic operation is too large to fit into the destination.
2. The Overflow flag (OF) is set when the result of a signed arithmetic operation is either too large or too small to fit into the destination.
3. The Sign flag (SF) is set when the result of an arithmetic or logical operation generates a negative result.
4. The Zero flag (ZF) is set when the result of an arithmetic or logical operation generates a result of zero.
5. The Parity flag (PC) sums the number of bits that are set in a number, and indicates whether the sum is odd or even.

**1-** To check the effect of arithmetic instructions on the flags of the CPU using the program given below and provide the reasons for your answers with every assembly instruction in the space provided.

**2-** Write an assembly program that inputs an arrayA of byte sized 5 elements. Reverses the elements of the array i-e take the mirror image of array and store the sorted elements in another arrayB

**Note:** viva will be taken at the end of lab carrying **5 marks**

**Assembly Program**

**.data**

var1 SBYTE 7fh

var2 SBYTE -128

var3 SBYTE -1

**.code**

Mov al, var1

add al, 1 al = \_\_\_\_\_\_\_\_\_\_, SF=\_\_\_\_\_\_, CF=\_\_\_\_\_, OF=\_\_\_\_\_

Mov bh, 255

add bh, 1 bh = \_\_\_\_\_\_\_\_\_\_, ZF=\_\_\_\_\_\_, CF=\_\_\_\_\_, OF=\_\_\_\_\_

Mov ah, var2

dec ah ah = \_\_\_\_\_\_\_\_\_\_, ZF=\_\_\_\_\_\_, CF=\_\_\_\_\_, OF=\_\_\_\_\_

Mov al, 1

Sub al, 2 al = \_\_\_\_\_\_\_\_\_\_, SF=\_\_\_\_\_\_, CF=\_\_\_\_\_, OF=\_\_\_\_\_

Mov ax, +7fffh

Add ax, 2 ax = \_\_\_\_\_\_\_\_\_\_, SF=\_\_\_\_\_\_, CF=\_\_\_\_\_, OF=\_\_\_\_\_

Mov al, 5

neg al al = \_\_\_\_\_\_\_\_\_\_, SF=\_\_\_\_\_\_, CF=\_\_\_\_\_, OF=\_\_\_\_\_

**; repeat with -5 as operand**

Mov al, var3

add al, 1 ZF = \_\_\_\_\_\_\_\_\_\_, SF=\_\_\_\_\_\_, CF=\_\_\_\_\_, OF=\_\_\_\_\_

Mov ax, -128

sub ax, 1 ax = \_\_\_\_\_\_\_\_\_\_, SF=\_\_\_\_\_\_, CF=\_\_\_\_\_, OF=\_\_\_\_\_