# Computer Organization & Assembly Language

BS (CS) \_Fall\_2023

Lab\_5 Manual



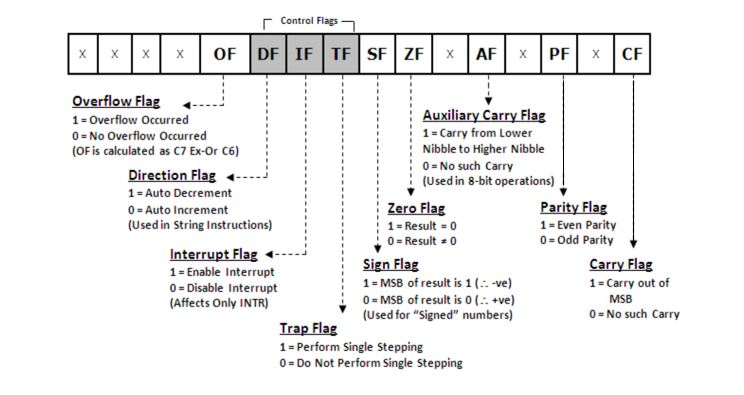
# Learning Objectives:

1. Flag Register
2. Jump instruction (JMP)
3. Loops

# **Lab 05**

# Flag Register:

These registers are used to store information about the state of the processor and the outcomes of various operations. Flag registers contain individual bits, each representing a specific condition or status of the CPU.



There are two types of flags

1. Status flags
2. Control flags

Status Flags:

* Allow the results of one instruction to influence later instructions.
* The arithmetic instructions use OF, SF, ZF, AF, PF, and CF
* Carry flag: Carry flag is set if carry is generated for unsigned integers.
* Overflow flag: Overflow flag is set if overflow is generated for signed integers.
* Zero flag: Zero flag is set if a zero is produced, as a result or as a comparison of two integers.
* Sign flag: Sign flag is set to one if the MSB of the destination operand is 1.

Controls flags:

These control flags serve specific purposes in controlling the operation of the CPU. Control flags in a CPU are essential for managing program execution, controlling interrupts, making decisions based on conditions, handling errors.

* Trap Flag (TF): When TF is set, the CPU enters single-step mode, where it executes one instruction at a time and generates a trap interrupt (interrupt 1) after each instruction.
* Interrupt Flag (IF): When IF is set, the CPU allows mask able interrupts to be processed. When it's cleared, mask able interrupts are disabled, preventing the CPU from responding to external interrupt requests.
* Directional Flag (DF):It control the direction of data movement in string operations. When DF is set, data moves from higher memory addresses to lower ones, and vice versa.

# Jump Instruction:

JMP is an unconditional jump command that is use to jump to a specific line. That line or block of code is identified by a label. Label can be any word other than the names of the variable used in the code or the keywords.

e.g., *START:*

*mov al,05*

*JMP START*

# Loops:

The keyword ‘loop’ runs a static loop of specific number of times and that number is stored in the Counter register, CX.

e.g., *mov cx,5*

*mov al,0*

*start:*

*inc al*

*loop start*

## Tasks

**For below questions(1, 2, 3), provide the values of the flags: Carry Flag (CF), Zero Flag (ZF), Sign Flag (SF), Overflow Flag (OF), and Parity Flag (PF) in binary after each operation.**

Question 1:

Consider two binary numbers, A = 1011 and B = 1101. Perform addition, subtraction, Multiplication, division operation. Provide the binary result along with flags value. Explain why flags is set or not set.

Question 2:

Take two binary numbers, X = 1110 and Y = 0101. Execute a multiplication, decrement, addition, subtraction, and Increment. Provide the binary result along with flags value. Explain flags is set.

Question 3:

Let's work with the binary numbers R = 1001 and S = 0010. Perform division, increment and decrement operation using the DIV, INC, DEC instruction.

**Use loops and labels to complete the following questions.**

Question 4:

Write a code to of generate even number from 1 to (the sum of last 2 digits) of your roll number save it in array.

Question 5:

Find the factorial of last digit of your roll number using loops.

Question 6:

Show the ASCII code of character a-z using loops.

## Submission Instruction

1. You must add your name and roll no in word file and each of your code files
2. Paste task wise screen short of your program with output into word. Add a two-line explanation of your program, how the flags are set etc.
3. Submit your .asm (all questions) files with word document (i22-1234\_LAB05.docx) in a zip folder. The name of your zip folder become lab number as well as your roll number, i-e i22-1234\_LAB05.zip
4. Failure to comply with the above instructions will result **0** in lab.