**Practical No. 12**

**Title:** Finding the 1’s and 2’s Complement of a 16-bit Number

**Objective:** To determine the **1’s complement** and **2’s complement** of a 16-bit number using binary operations.

**Theory:** In binary arithmetic, **complements** are used for signed number representation and arithmetic operations.

**1’s Complement:**

The **1’s complement** of a number is obtained by **inverting all bits** (changing 0 to 1 and 1 to 0).

**Example (16-bit number):**

Original: 1011 0010 1100 1010

1’s Complement: 0100 1101 0011 0101

**2’s Complement:**

The **2’s complement** of a number is found by:

1. Taking its **1’s complement**.
2. Adding **1** to the least significant bit (LSB).

**Example:**

1’s Complement: 0100 1101 0011 0101

Add 1: +0000 0000 0000 0001

------------------------------------

2’s Complement: 0100 1101 0011 0110

The **2’s complement** is used for representing negative numbers in binary and performing subtraction operations.

**Materials/Tools Required:**

* Microprocessor/microcontroller (e.g., 8085/8051)
* Assembler/Simulator
* Computer system with programming software
* Binary calculator (optional)

**Procedure:**

**Finding the 1’s Complement:**

1. **Load the 16-bit number** into a register pair (e.g., HL register in an 8085 processor).
2. **Invert all bits** (Use the **CMA (Complement Accumulator)** instruction for both higher and lower bytes).
3. **Store the Result** in memory or display it.

**Finding the 2’s Complement:**

1. **Use the 1’s Complement Result.**
2. **Add 1** to the least significant byte using the ADD instruction.
3. **Store the 2’s Complement Result** in memory.

**Observations:**

* **1’s complement** flips all bits without modifying value representation.
* **2’s complement** provides a signed representation for negative numbers.
* If a number is already in 2’s complement form, taking another 2’s complement restores the original number.

**Conclusion:**

The 1’s and 2’s complement of a 16-bit number are successfully determined using bitwise operations and addition. This process is crucial in **signed number representation and arithmetic operations** in digital systems.

**Applications :**

* Used in **binary subtraction** using 2’s complement method.
* Essential for **negative number representation** in microprocessors.
* Applied in **digital logic circuits and ALU operations** for arithmetic calculations.