| Experiment No. 4 |
| --- |
| Implement a program to perform addition and subtraction of binary, hexadecimal number |
| Date of Performance: |
| Date of Correction: |

**Aim:** To implement a program that performs addition and subtraction operations on binary and hexadecimal numbers.

**Objective:** To understand and implement arithmetic operations on binary and hexadecimal numbers commonly used in computer systems.

**Theory:**

**Introduction:**

In digital systems and low-level computing, binary (base-2) and hexadecimal (base-16) number systems are widely used. Binary numbers represent the core data format of all computing systems, while hexadecimal offers a compact, human-readable form of binary. Understanding how to perform arithmetic operations (like addition and subtraction) in these systems is essential for tasks like instruction processing, memory addressing, and debugging.

**Binary Number System (Base-2):**

* Uses only two digits: 0 and 1.
* Each binary digit (bit) represents an increasing power of 2.
* Computers use binary because it maps directly to digital electronics: ON (1) and OFF (0).

**Binary Addition Rules:**

| **A** | **B** | **Carry In** | **Sum** | **Carry Out** |
| --- | --- | --- | --- | --- |
| **0** | **0** | **0** | **0** | **0** |
| **0** | **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **0** | **1** |
| **1** | **1** | **1** | **1** | **1** |

**Binary Subtraction Rules:**

| **A** | **B** | **Borrow In** | **Difference** | **Borrow Out** |
| --- | --- | --- | --- | --- |
| **0** | **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** | **0** |
| **0** | **1** | **0** | **1** | **1** |
| **1** | **1** | **0** | **0** | **0** |

Subtraction can also be implemented by adding the 2's complement of the subtrahend.

**Hexadecimal Number System (Base-16):**

* Uses digits 0–9 and letters A–F (A=10 to F=15).
* Each hex digit maps to 4 binary bits (e.g., A = 1010).
* Used in memory addresses, machine-level instructions, and debugging.

**Hexadecimal Arithmetic:**

* Addition and subtraction follow rules similar to decimal but use base-16.
* If a sum exceeds 15 (F), a carry is generated.
* Subtraction may involve borrowing if the subtrahend digit is greater than the minuend digit.

**Examples:**

**Binary Addition:**

1011 (11)

+ 1101 (13)

= 11000 (24)

**Hexadecimal Addition:**

A9

+ 6F

= 118 (carry handled)

**Applications:**

* Processor ALUs perform these arithmetic operations at the hardware level.
* Memory management often requires binary or hex arithmetic for address calculations.
* Instruction encoding in microcontrollers involves operations in hex.
* Essential in assembly language programming, debugging, and system-level diagnostics.

**Solution:**

**Conclusion:** We explored how binary and hexadecimal numbers are used in arithmetic operations and gained hands-on experience in how processors perform basic calculations.