One’s Complement and Two’s Complement Representations (with Signed and Unsigned Numbers)

---

1. What are Unsigned and Signed Numbers?

1.1 Unsigned Numbers

- **Definition**:
  Unsigned numbers are **always non-negative** — only positive integers and zero.
- **Representation**:
  All bits represent the magnitude directly (no sign bit).
- **Example (4 bits)**:
  `0000` (0), `0001` (1), ..., up to `1111` (15).

1.2 Signed Numbers

- **Definition**:
  Signed numbers can be **positive** or **negative**. One bit (usually the leftmost) is used to represent the **sign**.
  - `0` → Positive
  - `1` → Negative
- **Representation**:
  Different methods can represent signed numbers:
  - One’s Complement
  - Two’s Complement (most common today)

---

2. Memory Space Example: 8 Bits

Suppose you have **8 bits** available. Let’s explore:

2.1 Unsigned Numbers

- **Total possible values**:
  28=2562^8 = 25628=256 different numbers.
- **Range**:

    - Minimum: 0 (`00000000`)
    - Maximum: 255 (`11111111`)
- **List**:
  0, 1, 2, ..., 255

2.2 Signed Numbers (Two's Complement or One's Complement)

Signed numbers divide the range into positive and negative numbers.

- **Total possible values**:
  2^8 = 256 different combinations.
- **Range**:
    - In **One's Complement**:
        - Range:
        - i.e., **−127 to +127**
        - (Two representations for zero: `00000000` and `11111111`)
    - In **Two's Complement**:
        - Range:
        - i.e., **−128 to +127**
        - (Only one zero: `00000000`)

Type	Total Combinations	Range
Unsigned (8 bits)	256	0 to 255
Signed (8 bits, One's Complement)	256	-127 to +127
Signed (8 bits, Two's Complement)	256	-128 to +127

---

3. Recap: One's Complement and Two's Complement

3.1 One's Complement

- **How**:
    - Positive numbers → as-is (regular binary)
    - Negative numbers → flip all bits (1 → 0, 0 → 1)
- **Note**:
  Two different zeros exist: positive zero and negative zero.

3.2 Two's Complement

- **How**:
  - Positive numbers → as-is
  - Negative numbers → flip all bits + add 1
- **Note**:
  Only one unique zero exists.

---

4. How to Find Complements

4.1 Finding One's Complement

- Step 1: Write the positive binary number.
- Step 2: Invert (flip) each bit.

Example:
Decimal +6 → `00000110`
One's complement of -6 → `11111001`

4.2 Finding Two's Complement

- Step 1: Write the positive binary number.
- Step 2: Invert each bit.
- Step 3: Add 1 to the result.

Example:
Decimal +6 → `00000110`
One's complement → `11111001`
Add 1 → `11111010` (this is Two's complement of -6)

---

5. Two Worked Examples

Example 1: Represent -9 in One's and Two's Complement (8 bits)

- Binary of +9: `00001001`

One's Complement:

- Invert: `11110110`

Two's Complement:

- Add 1 → `11110110` + `00000001` = `11110111`

Representation	Value
Positive 9	00001001
One’s Complement of -9	11110110
Two’s Complement of -9	11110111

Example 2: Represent -13 in One’s and Two’s Complement (8 bits)

- Binary of +13: `00001101`

One’s Complement:

- Invert: `11110010`

Two’s Complement:

- Add 1 → `11110010` + `00000001` = `11110011`

Representation	Value
Positive 13	00001101
One’s Complement of -13	11110010
Two’s Complement of -13	11110011
