Unsigned and Signed Operations

Thought Process

To perform subtraction of two numbers $A - B$, it must be transformed into an addition operation:

$$A - B = A + (-B)$$

Instead of manually subtracting one number from another, computers **process it as an addition of complements**, which effectively reverses the negative sign of the subtrahend. This simplifies hardware implementation and makes it efficient for digital circuits.

Subtraction Using Radix Complement

To subtract two numbers $A - B$ using the radix complement, follow these steps:

1. Use **radix complements** to convert the operation into an addition operation.
2. If the result generates a carry out, **discard the carry** and the result is correct.
3. If no carry is generated, take the **negative radix complement** of the result to get the magnitude of the correct answer.

```
Example 1: Calculate 5 − 3 in base-2.

Take the 2's complement of 3 (0011) to get -3 (1101).

     0 1 0 1   (5)
+    1 1 0 1   (2's complement of 3)
------------
   1 0 0 1 0   (sum of 5 and 2's complement of 3)
-  1 0 0 0 0   (since there is a carry out, discard end carry)
------------
     0 0 1 0   (answer)
```

```
Example 2: Calculate 3 − 5 in base-2.

Take the 2's complement of 5 (0101) to get -5 (1011).

     0 0 1 1   (3)
+    1 0 1 1   (2's complement of 5)
------------
     1 1 1 0   (since there is no carry out, take the negative complement)
------------
   - 0 0 1 0   (answer)

Note that while the initial answer 1110 is equal to -2 in signed representation,
taking the complement would allow us to easily get the magnitude of the result.
```

```
Example 3: Calculate −5 − 3 in base-2.

Take the 2's complement of 5 (0101) to get -5 (1011).
Take the 2's complement of 3 (0011) to get -3 (1101).

     1 0 1 1   (2's complement of 5)
+    1 1 0 1   (2's complement of 3)
------------
   1 1 0 0 0   (sum of 5 and 2's complement of 3)
-  1 0 0 0 0   (since there is a carry out, discard end carry)
------------
     1 0 0 0   (answer)
```

Subtraction of Using Diminished Radix Complement

Subtracting two numbers $A - B$ using the diminished radix complement follows similar steps to the radix complement, albeit with some differences. These differences are highlighted in **green**:

4. Use **diminished radix complements** to convert the operation into an addition operation.
5. If the result generates a carry out, **discard the carry and add 1 to the result** to account for the diminished complement.
6. If no carry is generated, take the **negative diminished radix complement** of the result to get the magnitude of the correct answer.

```
Example 1: Let X be 101 0100 and Y be 100 0011. Find X − Y.

Take the 1's complement of 100 0011 to get 011 1100.

     1 0 1 0 1 0 0   (X)
+    0 1 1 1 1 0 0   (1's complement of Y)
------------------
   1 0 0 1 0 0 0 0   (sum of X and 1's complement of Y)
-  1 0 0 0 0 0 0 0   (since there is a carry out, discard end carry)
+               1    (and then add 1)
------------------
     0 0 1 0 0 0 1   (answer)
```

```
Example 2: Let X be 101 0100 and Y be 100 0011. Find Y − X.

     1 0 0 0 0 1 1   (Y)
+    0 1 0 1 0 1 1   (1's complement of X)
------------------
     1 1 0 1 1 1 0   (since there is no carry out, take the negative complement)
------------------
   - 0 0 1 0 0 0 1   (answer)
```

Overflow and Carry in Subtraction

In unsigned subtraction, overflow occurs if **there is no carry-out from the most significant bit** (MSB).

- If carry occurs, the result is correct.
- If carry does not occur, it indicates that the result has wrapped around, leading to an incorrect (negative-like) value.

```
(No Overflow, Carry Occurs)

Example 1: Let X be 9 and Y be 5 in base-10. Find X − Y.

Take the 2's complement of 5 (0101 → 1011).

Add to 9:

     1 0 0 1   (X)
+    1 0 1 1   (2's complement of Y)
------------
   1 0 1 0 0

The leftmost carry-out is 1 (carry occurs), meaning no overflow occurred. The final
result is 0100, which is 4 (correct).
```

```
(Overflow, No Carry Occurs)

Example 2: Let X be 5 and Y be 9 in base-10. Find X − Y.

Take the 2's complement of 9 (1001 → 0111).

Add to 5:

     0 1 0 1   (X)
+    0 1 1 1   (2's complement of Y)
------------
     1 1 0 0

No carry-out (MSB doesn't produce a carry), meaning overflow has occurred. The
result 1100 is incorrect in unsigned arithmetic, as it represents 12, but should
have been negative (which isn't allowed in unsigned representation).
```

Overflow and Carry in Addition

In unsigned integer addition, overflow occurs when the result of the addition **exceeds the maximum representable value** for the given number of bits. This happens when there is a carry-out from the most significant bit (MSB), meaning the result is too large to fit within the fixed bit width.

- If a carry is generated from the leftmost bit, overflow has occurred.
- If no carry is generated, the result is correct.

```
(No Overflow, No Carry-Out)

Example 1: Let X be 5 and Y be 6 in base-10. Find X + Y.

     0 1 0 1   (X)
+    0 1 1 0   (Y)
------------
     1 0 1 1

Since there is no carry-out from the MSB, the result fits within the 4-bit range
0 to 15, so no overflow occurred.
```

```
(Overflow, Carry-Out Occurs)

Example 2: Let X be 9 and Y be 7 in base-10. Find X + Y.

     1 0 0 1   (X)
+    0 1 1 1   (Y)
------------
   1 0 0 0 0

Since the leftmost carry bit is 1 (result is 5 bits long: 10000), overflow occurred
because the result exceeds the 4-bit maximum value (1111 or 15 in decimal). The
actual stored result will be 0000 in base-2, which is incorrect.
```