**Київський національний університет імені Тараса Шевченка**

**факультет радіофізики, електроніки та комп’ютерних систем**

**Комп’ютерні системи**

Лабораторна робота № 2

Тема: «Арифметичні операції над двійковими числами»

Роботу виконав

студент 3 курсу

спеціальності “КІ-СА”

Ситниченко Денис Вікторович

Київ 2021

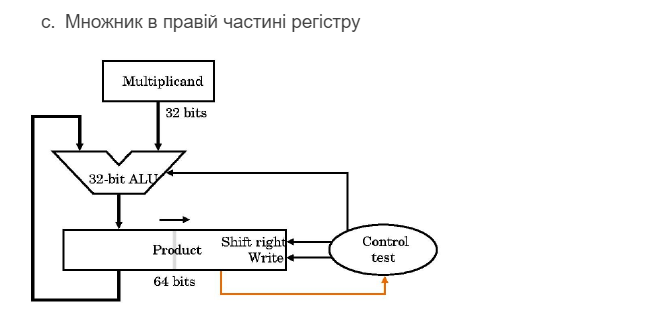
**Мета:** дослідити алгоритми, що використовуються в мікропроцесорах для множення та ділення цілих чисел та підходи до роботи з дійсними числами.

**Хід виконання роботи:**

**Варіанти завдань:**

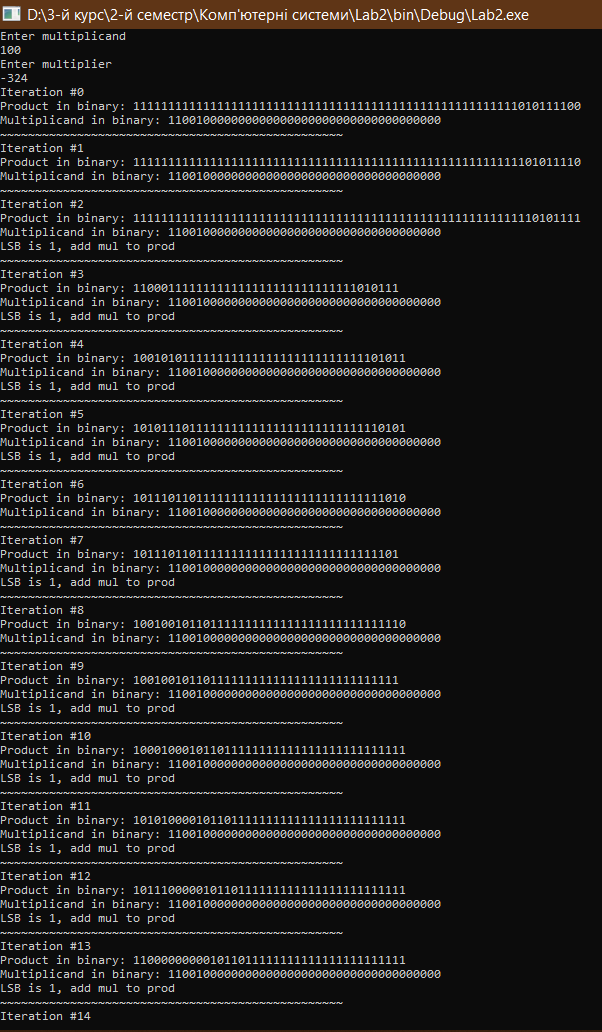
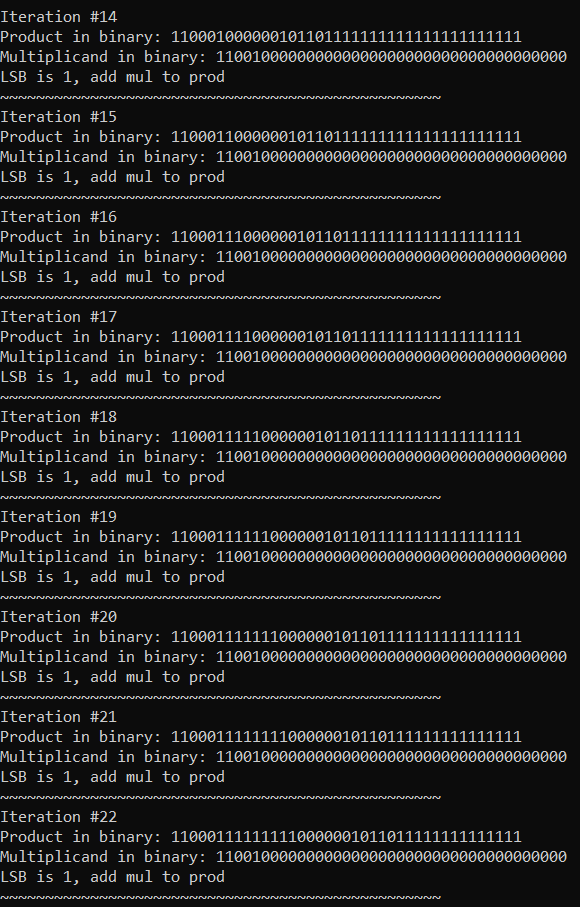
****

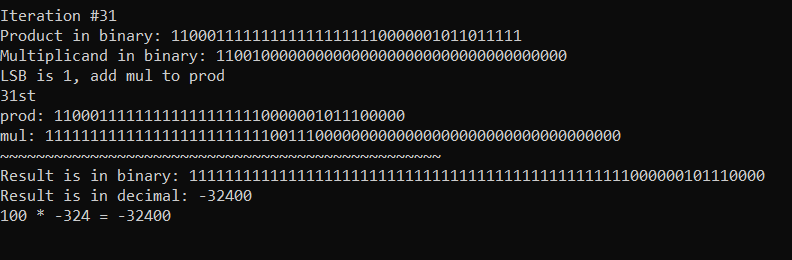
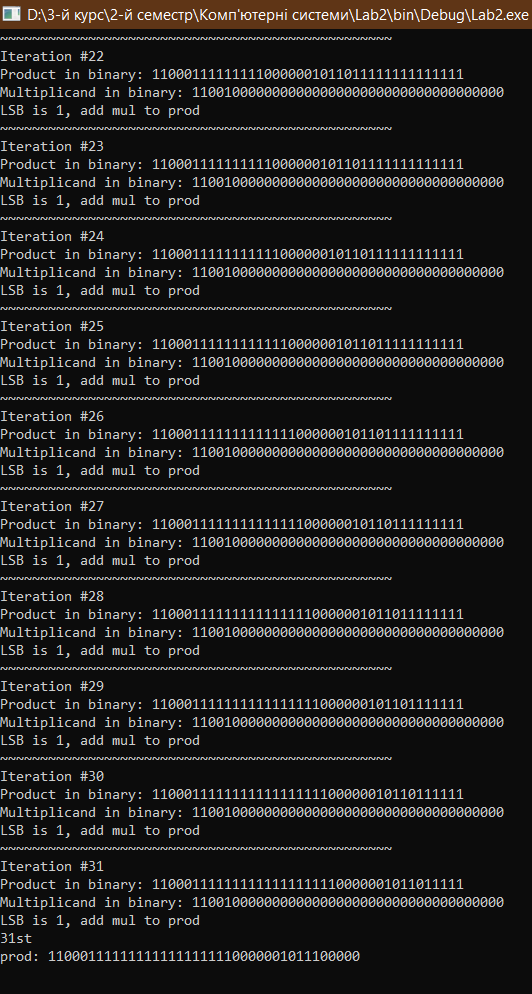
## **Множення двійкових чисел**



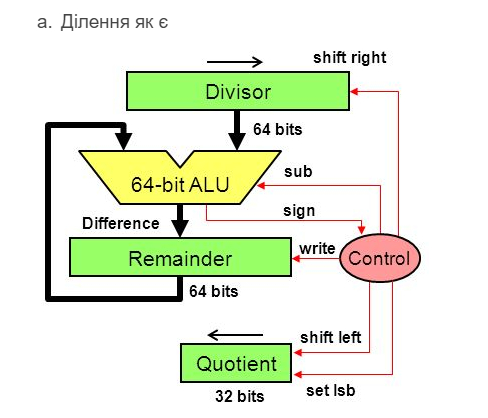
Алгоритм працює наступним чином: множник заноситься в в молодші 32-біти регістру product(prod), множне завантажується в регістр multiplicand(mul). На кожній ітерації зчитується наймолодший біт регістру product(prod) та в залежності від значення: якщо 1 – до старших 32 біт регістру product(prod) додається multiplicand(mul), якщо 0 – нічого не відбувається, після читання значення регістр product(prod) зсувається вправо на 1 біт. Так триває 32 ітерації, на останній ітерації для отримання правильного результату в разі, якщо множене було від’ємне, тобто знак був 1 – треба змінити значення в регістрі multiplicand(mul) на доповняльний код і додаи 1 до регістру product(prod) .

**Скріншоти роботи програми:**

****

****

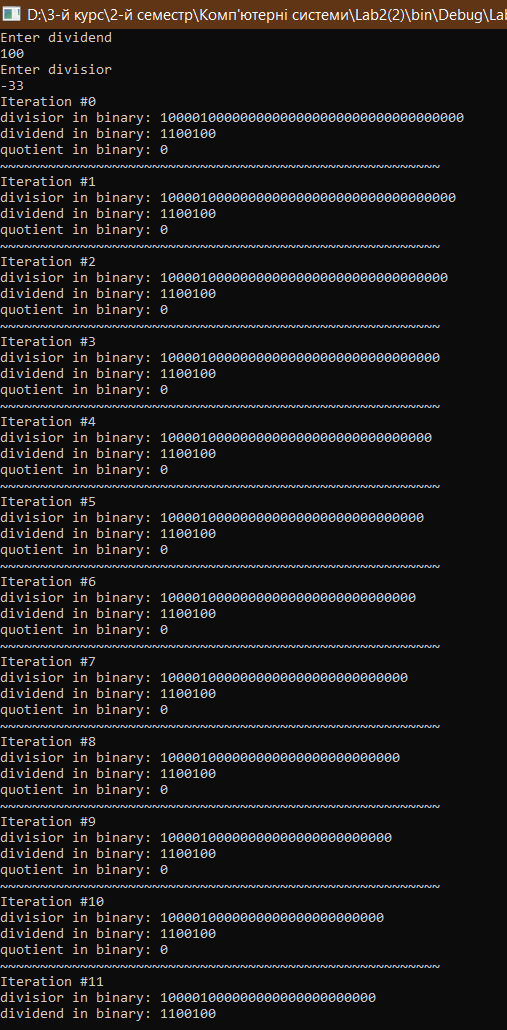
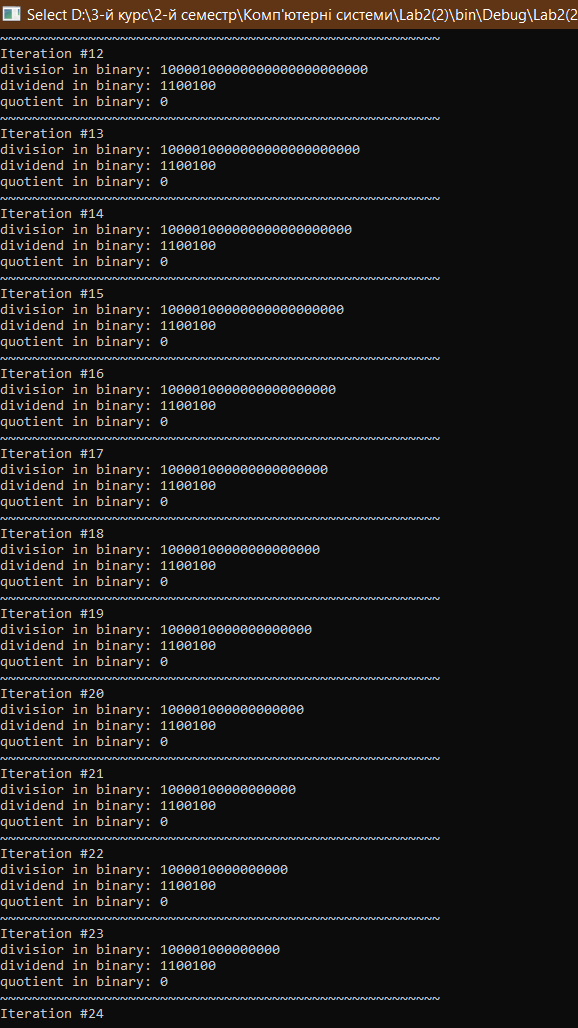
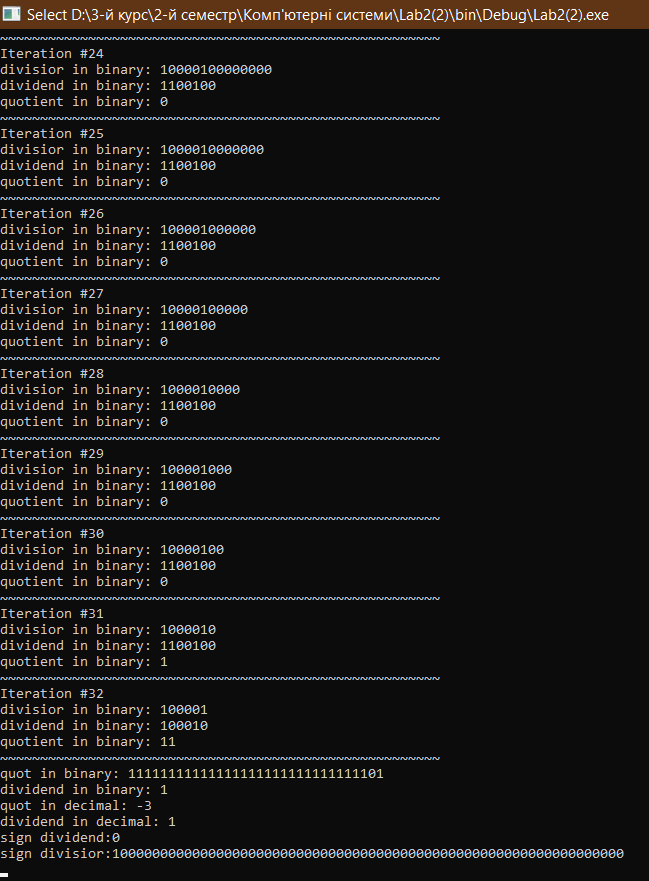
## **Ділення двійкових чисел**



Алгоритм працює наступним чином: ділене записується в ліву частину Remainder (молодші 32 біти) дільник записується в старші 32 біти регістру Divisior. Після чого дільник віднімається від діленого, якщо після цього ділене більше або дорівнює 0, то регістр Quotient зсувається вліво на 1 біт і його наймолодший біт стає 1, в іншому випадку, коли залишок менше 0 – до діленого додається дільник та регістр Quotient зсувається вліво на 1 біт, після чого регістр Divisor зсувається вправо на 1 біт і починається нова ітерація. В алгоритмі 33 ітерації, бо на першій відбувається фактично віднімання нуля від Remainder. В результаті роботи алгоритму в регістрі Quotient матимемо частку від ділення, а в молодших 32-х бітах регістру Remainder буде остача.

Щоб цей алгоритм працював зі знаковими числами, потрібно виконувати його над абсолютними значеннями діленого та дільника, після чого перетворити значення частки та/або остачі в їх доповняльний код. Якщо знаки діленого і дільника відрізняються, то значення частки має бути від’ємним, якщо ділене від’ємне, то й остача має бути також від’ємною.

**Скріншоти роботи програми:**

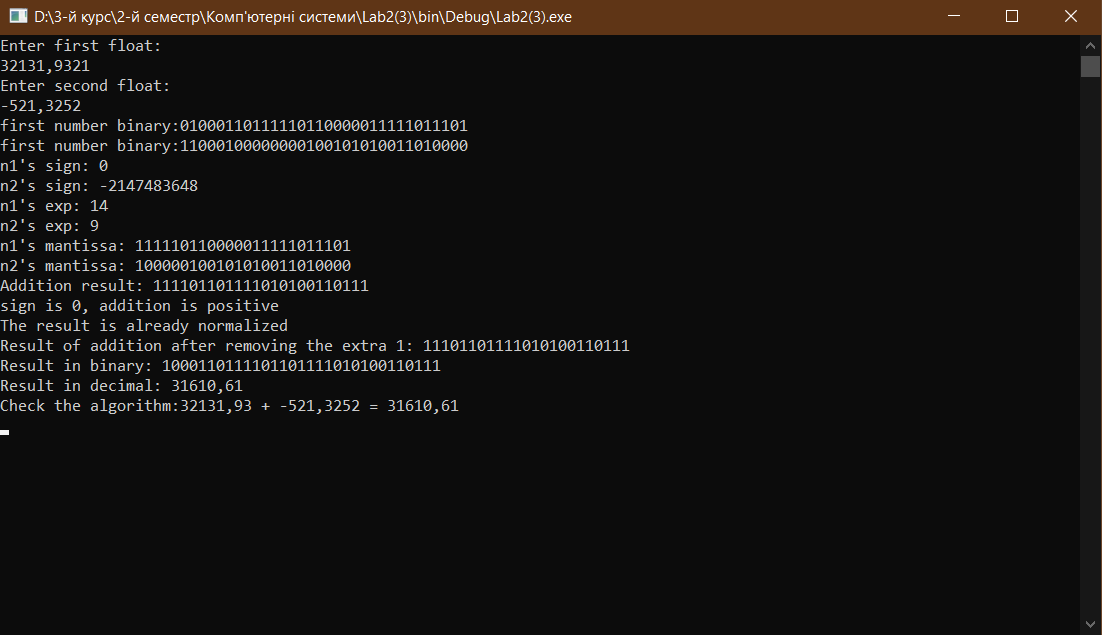


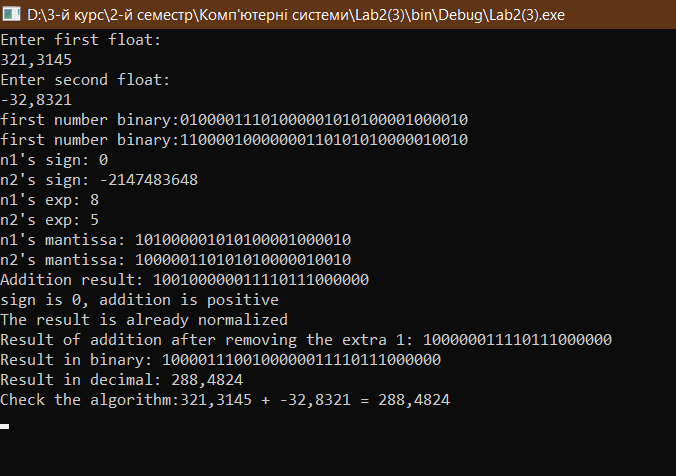
## **Робота з IEEE 754 Floating Point (Представити лише ключові кроки при виконанні операцій)**

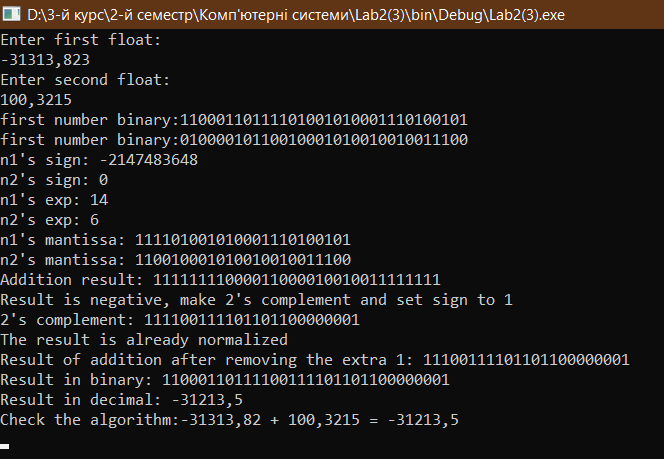


Для додавання 2 чисел з плаваючою крапкою, необхідно спочатку отримати знак, порядок, та мантису чисел із двійкового представлення кожного з чисел. Перший(найстарший) біт відповідає за знак, 8 бітів після нього – порядок, 23 біти після нього - це мантиса числа. До мантиси також потрібно додати 1 у 24-й біт. Спочатку порівнюються порядки чисел, у разі співпадіння, переходимо до додавання, а якщо ні – то урівнюємо порядки шляхом зсування направо мантиси числа з меншим порядком на різницю порядків двох чисел. Після цього мантиси додаються як звичайні двійкові числа. Якщо в результаті додавання вийшло від’ємне число, то мантису потрібно інвертувати, тим самим перевівши в доповняльний код, щоб вона стала додатньою, а також позначити, що знак результуючого числа буде 1. Після додавання потрібно нормалізувати результат, тобто щоб найстарша 1-ця мантиси була в 24-у біті, для цього відповідно треба зсувати мантису вправо, якщо в результаті додавання вона вийшла в 25-у біті, або вліво (якщо відбувалось віднімання) потрібну кількість разів, паралельно з чим треба інкрементувати чи декрементувати порядок результуючого числа. Після цього потрібно зібрати кінцеве число, в найстарший біт записується знак, наступні 8 – порядок, в наступні 23 мантиса, але в мантиси прибирається 1-а одиниця, бо вона є схованою.

**Скріншоти роботи програми:**







**Висновок**: алгоритми множення та ділення цілих чисел- це ітеративні алгоритми, в яких к-сть ітерацій задається розрядністю операндів. При роботі з числами з рухомою комою треба розуміти, що вони мають обмежену точність, тобто при додаванні чисел які відрізняються на декілька порядків – мантиса меншого з чисел буде зсунута вправо і як результат її частина буде втрачена.