# Міністерство освіти і науки України Львівський національний університет імені Івана Франка

**Факультет прикладної математики та інформатики**

Кафедра програмування

Лабораторна робота №8

**Логічні операції**

з курсу “Архітектура обчислюваних систем та схемотехніка”

Виконав: студент групи ПМІ-16 Процай Іван Сергійович

Львів – 2024

**Тема:** Логічні вирази

**Мета роботи:** Використовуючи середовище Visual Studio, освоїти

методи програмування з використанням логічних операцій на мові Assembler.

**Теоретичні відомості:**

or <операнд\_1>, <операнд\_2>

Ця операція виконує побітове логічне АБО, додаючи кожен біт операндів та зберігаючи результат у першому.

and <операнд\_1>, <операнд\_2>

Операція побітового логічного І. Вона виконує побітове множення операндів та зберігає результат у першому.

not <операнд>

Ця операція виконує бітове логічне заперечення.

xor <операнд\_1>, <операнд\_2>

Ця операція виконує бітове виключне АБО, зберігаючи результат у першому операнді. Часто використовується для обнулення регістрів (для обох операндів вказується один і той самий регістр).

shl (shr) <операнд>, <число\_бітів>

Ця операція здійснює зсув операнда вліво (вправо) на вказану кількість бітів. Біти, що вивільняються справа (зліва), заповнюються нулями.

rol (ror) <операнд>, <число\_бітів>

Ця операція виконує циклічний зсув операнда вліво (вправо) на вказану кількість бітів.

inc (dec) <операнд>

Ця операція збільшує (зменшує) значення операнда на одиницю.

**Хід роботи:**

1. У середовищі Visual Studio створив простий проект на мові C++.

2. Вибравши з контекстного меню проекту пункт Build Customization та підключив інструмент Microsoft Macro Assembler, активізувавши

опцію masm.

3. З використанням асемблерної вставки реалізував програму

для аналізу стану бітів двійкового масиву (підрахунку в масиві з восьми

чотирибітових слів кількості слів з парним числом одиниць в слові):

**Програма, що виконує підрахунок парних чисел в масиві arr.**

**Зображення, що містить текст, знімок екрана, Шрифт, програмне забезпечення

Автоматично згенерований описЗображення, що містить текст, Шрифт, знімок екрана, чорний

Автоматично згенерований опис**

Результат виконання

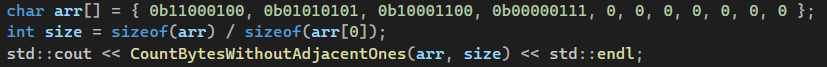
**Принцип роботи:**

1. Використовуючи вкладений цикл, код проходить через кожен елемент масиву та перевіряє кількість одиниць у його бінарному представленні.
2. Якщо кількість одиниць у бінарному представленні елемента є парною, воно збільшує лічильник парних чисел.
3. Нарешті, результат (кількість парних чисел) зберігається у змінній res.

Отже, цей код виконує ефективну перевірку та підрахунок парних чисел у масиві.

**Завдання варіанту 15 (Вивести цифри заданого трицифрового числа у порядку зростання.)**

Зображення, що містить текст, знімок екрана, програмне забезпечення

Автоматично згенерований опис 

Зображення, що містить текст, Шрифт, знімок екрана, чорний

Автоматично згенерований опис

Результат виконання

**Принцип роботи**:

1. Ініціалізуємо лічильник ecx як 0.
2. Завантажуємо адресу початку масиву arr у регістр esi.
3. Завантажуємо розмір масиву у регістр edx.
4. Починаємо цикл.
5. Завантажуємо значення байту у регістр al, переміщуємо вказівник на наступний елемент масиву.
6. Перевіряємо байт на наявність сусідніх одиниць та збільшуємо лічильник, якщо такі відсутні. Обгрунтування алгоритму:
   1. Беремо значення байту з масиву.
   2. Переміщуємося до наступного байту в масиві.
   3. Створюємо копію значення байту.
   4. Здійснюємо зсув копії байту на один біт управо.
   5. Виконуємо операцію "і" між копією та оригінальним значенням байту.
   6. Виконуємо операцію "і" між результатом попередньої операції та числом 85, яке має значення 01010101 у двійковій формі.
   7. Порівнюємо отриманий результат з нулем.
   8. Якщо результат цього порівняння не дорівнює нулю, це означає, що у байті є сусідні одиниці. У цьому випадку збільшуємо лічильник на одиницю.
7. Зменшуємо лічильник елементів масиву, повторюємо цикл та зберігаємо значення лічильника у відповідній змінній. Потім повертаємо різницю між розміром масиву та знайденою кількістю байтів без сусідніх одиниць.

**Висновок**: У цій лабораторній роботі ми досліджували використання логічних операцій у мові Assembler з використанням середовища Visual Studio. Під час виконання роботи ми освоїли базові операції логічного АБО, І, заперечення та виключного АБО, а також операції зсуву та циклічного зсуву.

Ми розглянули принципи побітового застосування цих операцій для виконання конкретних завдань, таких як підрахунок парних чисел у масиві та виведення цифр заданого трицифрового числа у порядку зростання.

Під час роботи були розглянуті алгоритми, які дозволили ефективно вирішити поставлені завдання. Робота демонструє вміння застосовувати базові логічні операції для виконання складних завдань програмування на рівні мови машинних команд.