МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського

«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп’ютерних систем та інфокомунікацій

Кафедра комп’ютерних систем, мереж і кібербезпеки (503)

Лабораторна робота № *5*

|  |  |
| --- | --- |
|  | *Упакування бітових груп* |
|  | (назва лабораторної роботи) |
| з дисципліни | *Архітектура комп'ютерів* |
|  | (шифр)  ХАІ**.**503**.**525a**.**03О**.**123-Комп'ютерна інженерія**,** ПЗ №9629619 |

|  |  |  |
| --- | --- | --- |
| Виконав студент гр. | 525а | *Литвиненко А.В.* |
| 14.11.2022 | (№ групи) | (П.І.Б.) |
| (підпис, дата) |  |  |
| Перевірив | канд. техн. наук, доцент | |
|  |  | *В. І. Дужий* |
| (підпис, дата) |  | (П.І.Б.) |

Харків – 2022

**Тема роботи:** упакування бітових груп

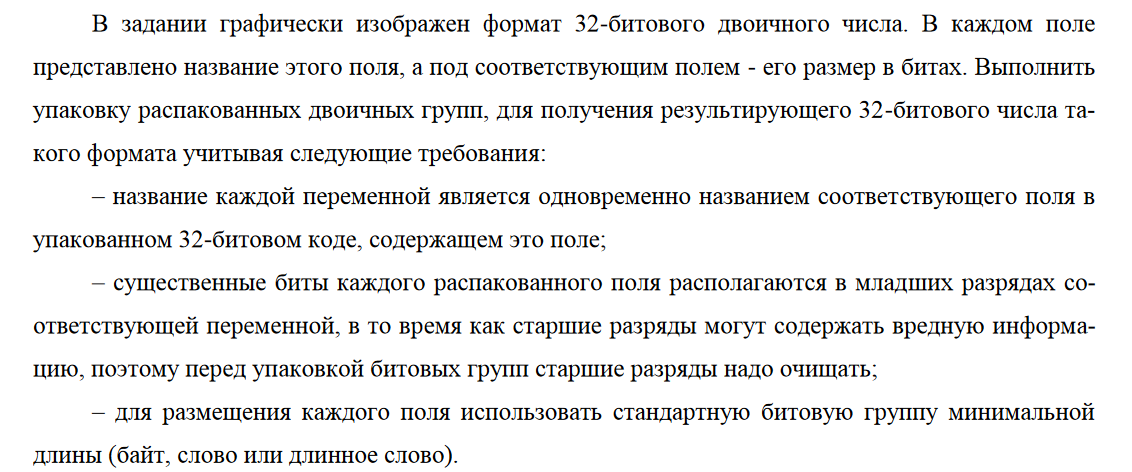
**Мета роботи**: вивчення логічних команд, команд сдвигу, алгоритмів упакування бітових груп, обробка чисел різної довжини.

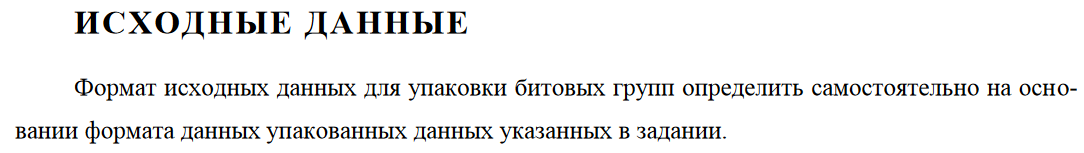
**Варіант 5**

**Задача 1**

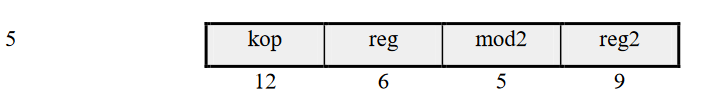
**Частина 1**. Постановка завдання

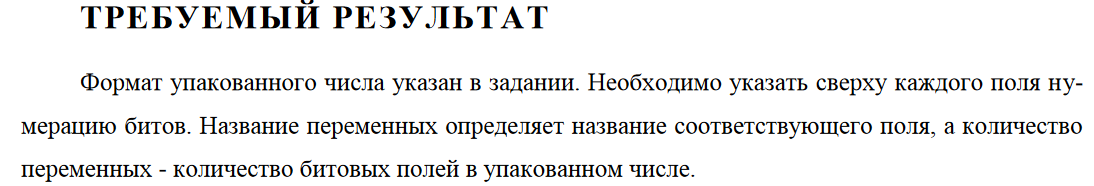
**Умова:**



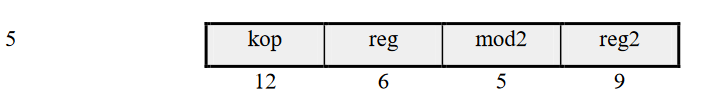


**Умова з додатка:**





**Частина 2**. Схема алгоритму



Ввести вхідні дані kop,reg,mod2,reg2;

Очистити старші біти в кожні змінній на C;

Об’єднати kop поле та число pack С;

Об’єднати reg поле та число pack С;

Об’єднати mod2 поле та число pack С;

Об’єднати reg2 поле та число pack С;

Очистити старші біти в кожні змінній на асемблері;

Об’єднати kop поле та число pack асемблері;

Об’єднати reg поле та число pack асемблері;

Об’єднати mod2 поле та число pack асемблері;

Об’єднати reg2 поле та число pack асемблері;

Вивести значення змінних на С;

Вивести значення змінних на асемблері;

На основі постановки завдання розроблений алгоритм, представлений на рисунку 1.

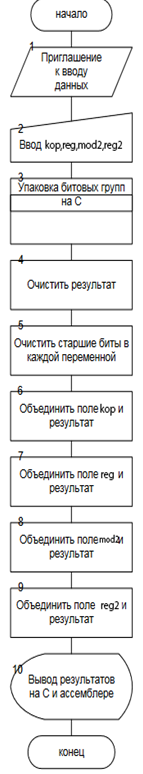


Рисунок 1 - Алгоритм перетворення

**Частина 3**. Розробка тестів

Таблиця 1 – Тестові набори

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Вхідні дані | | | | Вхідні дані | Очікуваний результат | Коментар |
| 1 | kop | reg | Mod2 | Reg2 |
| 2 | fff | 3f | 1f | 1ff | FFF 3F 1F 1FF | FFFF FFFF | Усі біти 1 |
| 3 | aaa | 2a | 15 | aa | AAA 2A 15 AA | AAAA AAAA | Чередування 1 та 0 |
| 4 | 555 | 15 | a | 155 | 555 15 A 155 | 5555 5555 | Чередування 0 та 1 |
| 5 | fff | 0 | 0 | 0 | FFF 0 0 0 | FFF0 0000 | Всі біти 1 в kop |
| 6 | 0 | 3f | 0 | 0 | 0 3F 0 0 | 000F C000 | Всі біти 1 в reg |
| 7 | 0 | 0 | 1f | 0 | 0 0 1F 0 | 0000 3E00 | Всі біти 1 в mod2 |
| 8 | 0 | 0 | 0 | 1ff | 0 0 0 1FF | 0000 01FF | Всі біти 1 в reg2 |
| 9 | 123 | 2a | e | 1ea | 123 2A E 1EA | 123A 9DEA | Різні біти |

**Частина 4**. Текст програми

Відповідно до розробленого алгоритму в середовищі Microsoft Visual Studio була написана програма, яка наведена нижче.

Main.c

/\*

File: unpack.cc

Unpacking bytes group

This program unpacking byte groups from unsigned int

Input datta:

unsigned int value, which consts byte groups:

12 + 6 + 5 + 9

31 0

kop reg mod2 reg2

12 6 5 9

\*/

#include <iostream>

#include <iomanip>

using namespace std;

unsigned long value, value\_a;

unsigned char mod2, reg, mod2\_a, reg\_a;

unsigned short kop, reg2, kop\_a, reg2\_a;

unsigned short tmp\_short;

int main() {

printf("\n\t\t(C) Lytvynenko A.V., 2022");

printf("\n\tPacking 32-bit number Value");

while (1) {

cout << "\nEnter 3 16-digest numbers for kop (for example, fff): ";

cin >> hex >> kop;

cout << "Enter 2 16-digest numbers for reg (for example, 3f): ";

cin >> tmp\_short; reg = tmp\_short;

cout << "Enter 2 16-digest numbers for mod2 (for example, 1f): ";

cin >> tmp\_short; mod2 = tmp\_short;

cout << "Enter 3 16-digest numbers for reg2 (for example, 1ff): ";

cin >> reg2;

kop\_a = kop;

reg\_a = reg;

mod2\_a = mod2;

reg2\_a = reg2;

reg2 &= 0x1ff;

mod2 &= 0x1f;

reg &= 0x3f;

kop &= 0xfff;

value = kop;

value = (value << 6) | reg;

value = (value << 5) | mod2;

value = (value << 9) | reg2;

\_\_asm {

and reg2\_a, 0x1ff

and mod2\_a, 0x1f

and reg\_a, 0x3f

and kop\_a, 0xfff

sub eax, eax

mov ax, kop\_a

shl eax, 6

or al, reg\_a

shl eax, 5

or al, mod2\_a

shl eax, 9

or ax, reg2

mov value\_a, eax

};

cout << hex

<< "Packed bytes group kop (C++): " << value

<< "\nPacked bytes group kop (ASM): " << value\_a;

}

return 0;

}

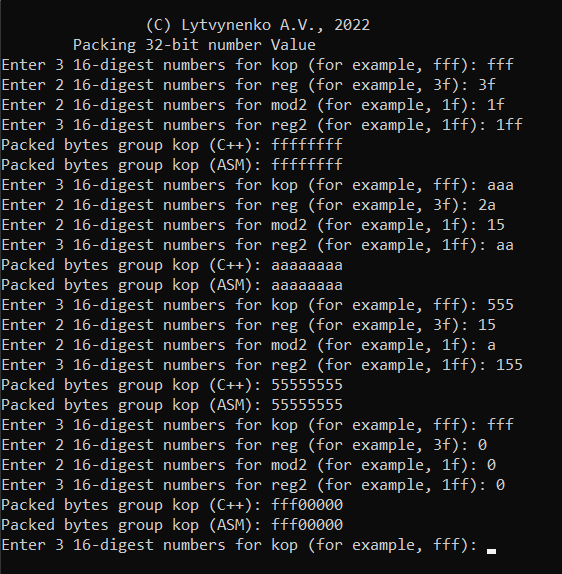
**Частина 5**. Тестування

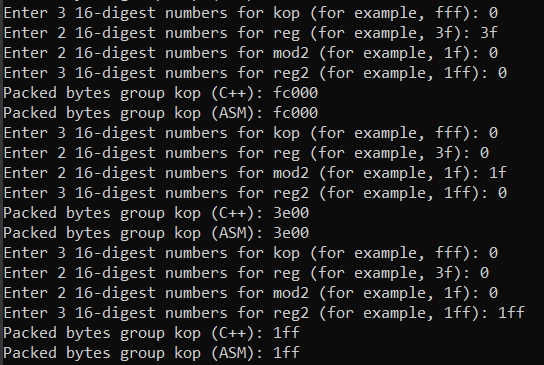
Результати тестування наведені в таблиці 2.

Таблиця 2 – Результати тестування

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Вхідні дані | | | | Очікуваний результат | Отриманий результат | Результат тестування |
| kop | reg | Mod2 | Reg2 |
| 1 | fff | 3f | 1f | 1ff | FFFF FFFF | FFFF FFFF | Good |
| 2 | aaa | 2a | 15 | aa | AAAA AAAA | AAAA AAAA | Good |
| 3 | 555 | 15 | a | 155 | 5555 5555 | 5555 5555 | Good |
| 4 | fff | 0 | 0 | 0 | FFF0 0000 | FFF0 0000 | Good |
| 5 | 0 | 3f | 0 | 0 | 000F C000 | 000F C000 | Good |
| 6 | 0 | 0 | 1f | 0 | 0000 3E00 | 0000 3E00 | Good |
| 7 | 0 | 0 | 0 | 1ff | 0000 01FF | 0000 01FF | Good |
| 8 | 123 | 2a | e | 1ea | 123A 9DEA | 123A 9DEA | Good |

**Скриншот тестування:**





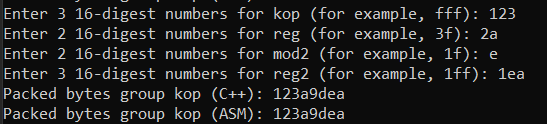


Рисунок 2 – скришот тестування

**Висновки**

Під час цієї лабораторної роботи я вивчив логічні команди, також команди здвигу, алгоритми упакування, бітові групи, обробку чисел з різними довжинами, покращив свої практичні навички у Сі та асемблері.