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 № *4*

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

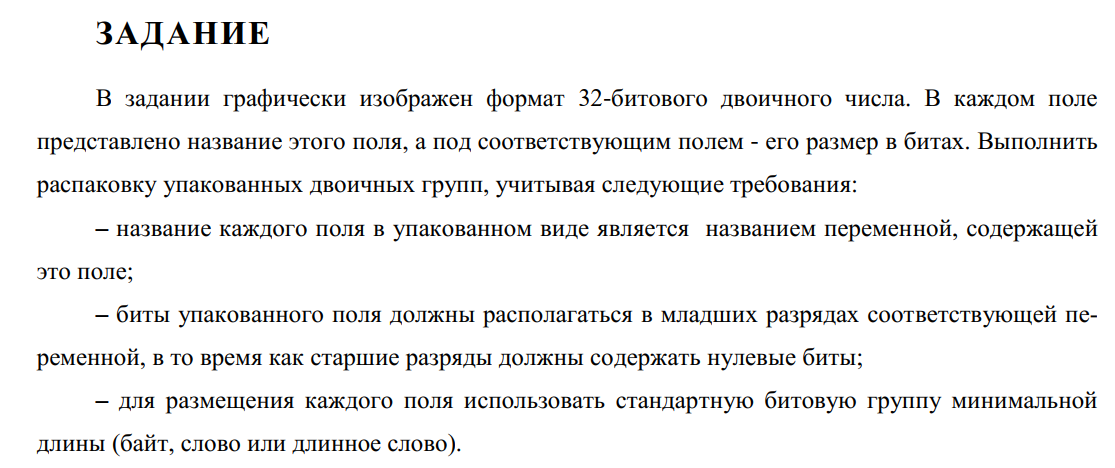
|  |  |  |
| --- | --- | --- |
| Виконав студент гр. | 525а | *Литвиненко А.В.* |
| 09.11.22 | (№ групи) | (П.І.Б.) |
| (підпис, дата) |  |  |
| Перевірив | канд. техн. наук, доцент | |
|  |  | *В. І. Дужий* |
| (підпис, дата) |  | (П.І.Б.) |

Харків – 2022

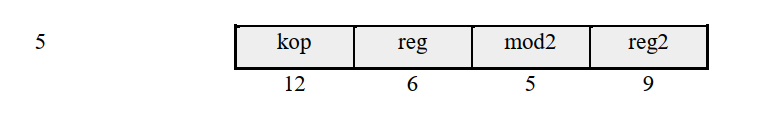
**Варіант 5**

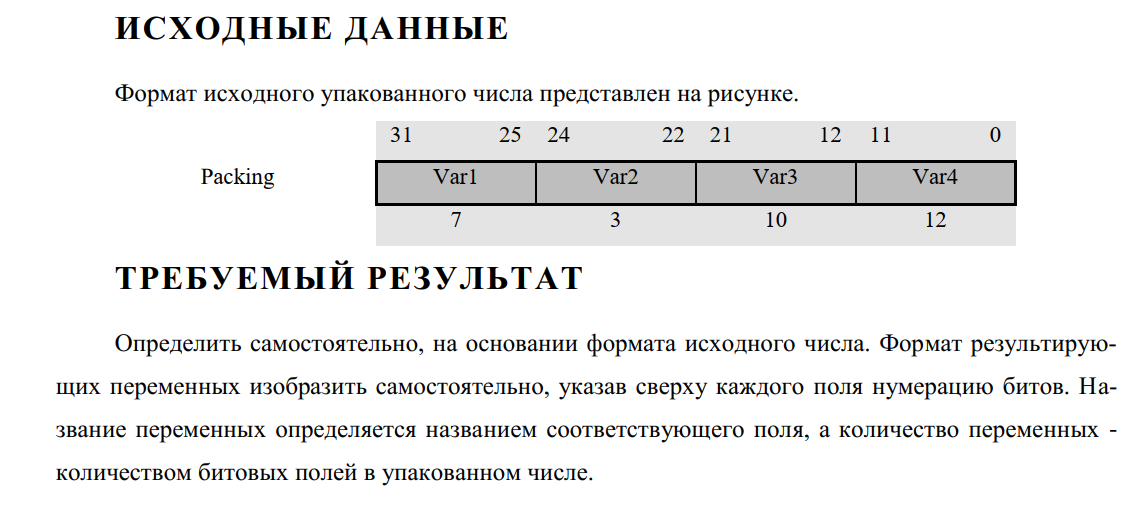
**Задача 1**

**Частина 1**. Постановка завдання

**Умова: **

**Вхідні дані:**



****

Частина 2. Опис алгоритму на псевдокоді

Вести вхідні дані з упакованими полями;

Виділити з числа поле reg2 на C;

Виділити з числа поле mod2 на C;

Виділити з числа поле reg на C;

Виділити з числа поле kop на C;

Виділити з числа поле reg2 на асемблері;

Виділити з числа поле mod2 на асемблері;

Виділити з числа поле reg на асемблері;

Виділити з числа поле kop на асемблері;

Вивести значення змінних на С;

Вивести значення змінних на асемблері;

**Частина 3**. Схема алгоритму

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

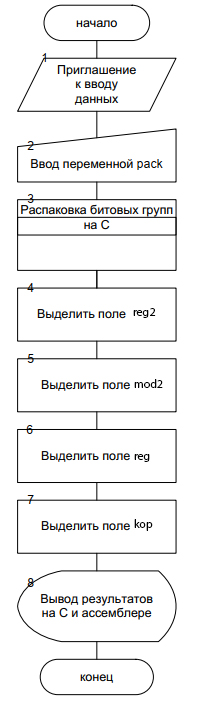


Рисунок 1 - Алгоритм перетворення

**Частина 4**. Розробка тестів

Таблиця 1 – Тестові набори

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Вхідні дані | Очікуваний результат | | | | Ціль тесту |
| Kop | Reg | Mod2 | Reg2 |
| 1 | FFFF FFFF | Fff | 3f | 1f | 1ff | Усі біти 1 |
| 2 | AAAA AAAA | Aaa | 2a | 15 | aa | Чередування 1 та 0 |
| 3 | 5555 5555 | 555 | 15 | a | 155 | Чередування 0 та 1 |
| 4 | FFF0 0000 | fff | 0 | 0 | 0 | Всі біти 1 в kop |
| 5 | 000F C000 | 0 | 3f | 0 | 0 | Всі біти 1 в reg |
| 6 | 0000 3E00 | 0 | 0 | 1f | 0 | Всі біти 1 в mod2 |
| 7 | 0000 01FF | 0 | 0 | 0 | 1ff | Всі біти 1 в reg2 |

**Частина 5**. Текст програми

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

/\*

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;

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

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

int main() {

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

printf("\n\tUnpacking byte groups");

while (1) {

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

printf("\nPlease, enter 8 16-bits numbers (exp, 5a9db8e4) : ");

scanf("%x", &value);

reg2 = value & 0x1ff;

mod2 = (value >> 9) & 0x1f;

reg = (value >> 14) & 0x3f;

kop = (value >> 20) & 0xfff;

\_\_asm {

mov eax,value

mov reg2\_a, ax

and reg2\_a, 0x1ff

shr eax, 9

mov mod2\_a, al

and mod2\_a, 0x1f

shr eax, 5

mov reg\_a, al

and reg\_a, 0x3f

shr eax, 6

mov kop\_a, ax

and kop\_a, 0xfff

}

cout << hex

<< "Bytes group kop (C++): " << (int)kop

<< "\nBytes group reg (C++): " << (int)reg

<< "\nBytes group mod2 (C++): " << (int)mod2

<< "\nBytes group reg2 (C++): " << (int)reg2

<< "\n\nBytes group kop (Asm): " << (int)kop\_a

<< "\nBytes group reg (Asm): " << (int)reg\_a

<< "\nBytes group mod2 (Asm): " << (int)mod2\_a

<< "\nBytes group reg2 (Asm): " << (int)reg2\_a

<< endl;

}

return 0;

}

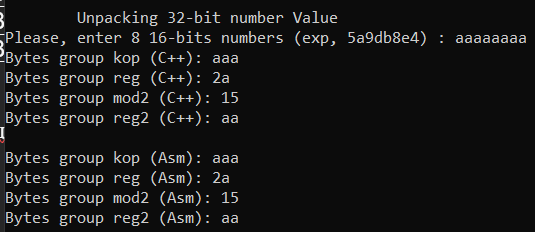
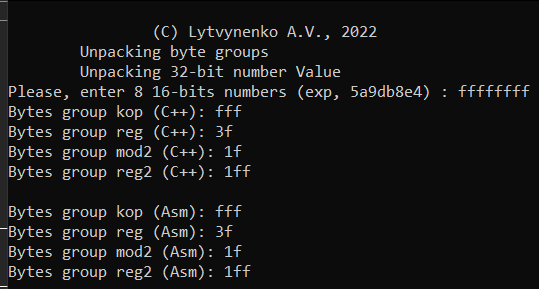
**Частина 6**. Тестування

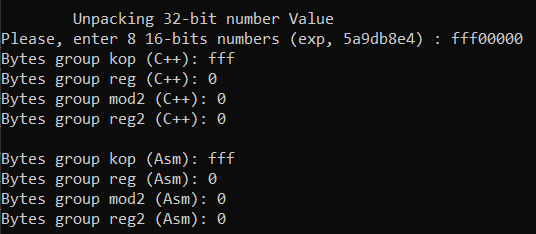
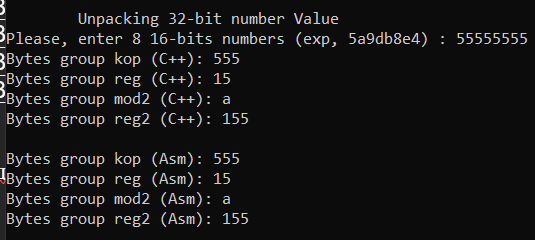
Результати тестування наведені в таблиці 2.

Таблиця 2 – Результати тестування

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Вхідні дані | Очікуваний результат | | | | Отриманий результат | | | | Статус |
| Kop | Reg | Mod2 | Reg2 | Kop | Reg | Mod2 | Reg |
| 1 | FFFF FFFF | Fff | 3f | 1f | 1ff | Fff | 3f | 1f | 1ff | ОК |
| 2 | AAAA AAAA | Aaa | 2a | 15 | aa | Aaa | 2a | 15 | aa | ОК |
| 3 | 5555 5555 | 555 | 15 | a | 155 | 555 | 15 | a | 155 | ОК |
| 4 | FFF0 0000 | fff | 0 | 0 | 0 | fff | 0 | 0 | 0 | ОК |
| 5 | 000F C000 | 0 | 3f | 0 | 0 | 0 | 3f | 0 | 0 | ОК |
| 6 | 0000 3E00 | 0 | 0 | 1f | 0 | 0 | 0 | 1f | 0 | ОК |
| 7 | 0000 01FF | 0 | 0 | 0 | 1ff | 0 | 0 | 0 | 1ff | ОК |

**Скриншот тестуваня:**





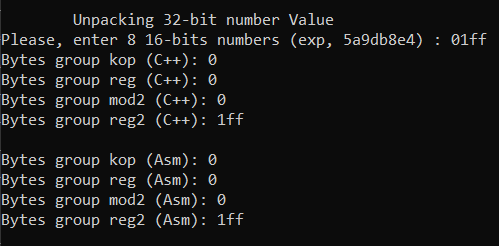
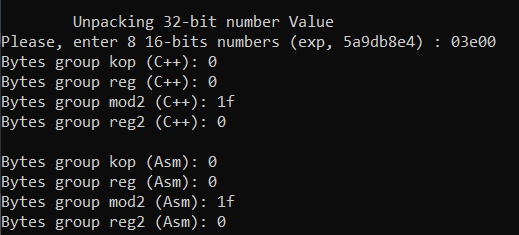
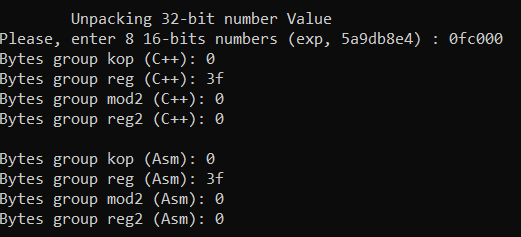


Рисунок 2 – скришоти тестування

**Висновки**

Під час цієї лабораторної роботи я навчився використовувати розпакування бітових груп з 32-бітого числа, попрактивувався у використанні логічних команд, команд сдвигів та алгоритмах розпаковки бітових груп.