

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ (МИИТ))**

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

**Отчет по практической работе**

«Логические команды и команды манипулирования битами»

по дисциплине

«Машинно-ориентированные языки программирования»

**Выполнил:** студент группы ТКИ-342

Фещенко В. А.

**Проверил:** доцент кафедры УиЗИ, к.т.н. Логинова Л. Н.

**Москва 2022 г.**

**Цель работы**

Изучение логических команд, команд манипулирования битами и получения навыка работы с ними.

**Задание**

1. Занести в память 32 разрядное шестнадцатеричное число в соответствии с вариантом из таблицы 6;
2. Подсчитать количество нулей и единиц в данном числе двумя разными способами;
3. Подсчитать количество парных нулей и парных единиц в данном числе;
4. В младшем байте числа обменять между собой биты 0-7, 1-6, 2-5, 3-4.

|  |  |
| --- | --- |
| Номер вариант | Исходное число(hex) |
| 14 | 14523678 |

Таблица 1 – Исходные данные

**Код программы**

#include <iostream>

#include <string>

#include <cmath>

std::string to\_binary(int num)

{

std::string binary;

binary.clear();

int bit = 0;

int\* arr = new int[8];

for (size\_t i = 0; i < 8; i++)

{

bit = num % 2;

arr[i] = bit;

num = num / 2;

}

for (int i = 7; i >= 0; i--)

{

binary.append(std::to\_string(arr[i]));

}

return binary;

}

int main()

{

uint32\_t num = 0x14523678;//0001'0100'0101'0010'0011'0110'0111'1000

size\_t zero = 32; //19

size\_t one = 0; //13

size\_t zeroPair = 0; //6

size\_t onePair = 0; //4

uint8\_t revers = 0; //30 0001'1110

uint8\_t straight = 0; // 0111'1000

\_\_asm

{

pushad

lea esi, num

mov eax, [esi]

jmp count

degree\_two:

jecxz if\_zero

mov edi, 2

mul edi

loop degree\_two

mov ebx, eax

jmp after\_degree

if\_zero:

mov ebx, 1

jmp after\_degree

count:

bsf ecx, eax

je result

push eax

mov eax, 1

jmp degree\_two

after\_degree:

pop eax

sub eax, ebx

inc one

dec zero

jmp count

result:

popad

}

std::cout << "zero: " << zero << " one: " << one << "\n";

zero = 0; //19

one = 0; //13

\_\_asm

{

pushad

lea esi, num

mov eax, [esi]

mov ecx, 32

second\_count:

bt eax, ecx

jc inc\_one

inc zero

loop second\_count

jmp result\_two

inc\_one:

inc one

loop second\_count

result\_two:

popad

}

std::cout << "another variant: \n" << "zero: " << zero << " one: " << one << "\n";

\_\_asm

{

pushad

lea esi, num

mov eax, [esi]

mov ecx, 0

the\_loop:

cmp ecx, 32

jge pair\_result

mov edi, 0

shld edi, eax, 2

cmp edi, 3

jz one\_pair\_counter

cmp edi, 0

jz zero\_pair\_counter

shl eax, 1

add ecx, 1

jmp the\_loop

one\_pair\_counter:

inc onePair

shl eax, 2

add ecx, 2

jmp the\_loop

zero\_pair\_counter:

cmp ecx, 31

jge pair\_result

inc zeroPair

shl eax, 2

add ecx, 2

jmp the\_loop

pair\_result:

popad

}

std::cout << "pairs: \n" << "zero's: " << zeroPair << " one's: " << onePair << "\n";

\_\_asm

{

pushad

lea esi, num

mov eax, [esi]

lea edi, straight

mov [edi], al

mov edx, 0

mov ebx, 0

mov bl, al

mov ecx, 8

swap:

sal dl, 1

shr bl, 1

adc dl, 0

loop swap

lea ecx, revers

mov [ecx], dl

popad

}

std::cout <<"straiht: " << to\_binary((int) straight) << "\nrevers: " << to\_binary((int) revers) << " \n";

return 0;

}

**Блок-схема алгоритма.**



Рисунок - Блок-схема алгоритма



Рисунок - Блок-схема первой подпрограммы



Рисунок - Блок-схема алгоритма второй подпрограммы

**Результат выполнения программы**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок - Результат выполнения программы

**Контрольные вопросы**

1. Приведите фрагменты программ на языке ассемблера, выполняющие следующие действия: сдвиг СХ на 2 бита вправо; сдвиг АХ на 2 бита влево; приравнивание значения 426 регистру СХ; сравнение значения переменной BYTE1 типа Byte и числа 2516.

Ответ:

ror cx, 2

rol ax, 2

mov cx, 426

cmp BYTE1, 25

1. Полагая, что DL содержит 1111'00012, а переменная BOOL типа Byte содержит 1110'00112, определите значение регистра DL после выполнения каждой отдельно взятой команды:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| а) | AND | DL,BOOL | b) | OR DL,BOOL |
| c) | XOR | DL,BOOL | d) | AND DL,0 |
| e) | XOR | DL,$FF | f) | NOT DL |

Ответ:

a) 11100001, b) 11110011, c) 00010010, d) 00000000, e) 00001110,

f) 00001110

1. Полагая, что DX содержит значение 1011'1001'1011'10012, определите, какое значение будет содержать DX после выполнения следующих отдельно взятых инструкций:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| a) | SHL | DL,1 | b) | SHL | DX,2 | c) | SHR DX,1 |
| d) | SAR | DX,2 | e) | SAL | DH,3 | f) | ROR DX,3 |
| g) | ROR | DL,3 | h) | RCL | DX,17 |  |  |

Ответ:

a) 10111001 01110010, b) 11100110 11100100, c) 01011100 11011100, d) 11101110 01101110, e) 11001011 10010000, f) 00110111 00110111, g) 10111001 00110111, h) 10111001 10111001

1. Чему будет равен регистр АX в результате следующего фрагмента программы:

mov cx, $0c

bsf ax, cx

Ответ: 0002

1. Чему будет равен регистр СX в результате следующего фрагмента программы (М – переменная типа Word):

mov cx, $2c

mov m, cx

bsf cx, m

Ответ:0002

1. Как приравнять одной командой регистр ECX к нулю десятью разными способами (можно использовать команды из всех лабораторных работ)? Начальное значение регистра ECX и флагов случайное.

Ответ:

xor ecx, ecx // 1

mov ecx, 0 // 2

shr ecx, 16 // 3

shr ecx, 16

sal ecx, 16 // 4

sal ecx, 16

and ecx, 0 // 5

sub ecx, ecx // 6

imul ecx, 0 // 7

loopM: // 8

loop loopM

or ecx, 0xFFFFFFFF // 9

inc ecx

loopM2: // 10

dec ecx

cmp ecx, 0

jne loopM2

**Вывод**

В ходе выполнения лабораторной работы были получены знания по работе с логическими командами и командами манипулирования битами.