МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА РАДИОЭЛЕКТРОННЫХ СРЕДСТВ

Отчет по дисциплине

«Цифровые устройства и микропроцессоры»

Лабораторная работа №2

«ПРИНЦИПЫ ВЫПОЛНЕНИЯ КОМАНД ВЕТВЛЕНИЯ, ОРГАНИЗАЦИЯ

ЦИКЛОВ И ПОДПРОГРАММ»

Вариант 9

Выполнил: студент группы ИНБб– 3301\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ю.А. Демина /

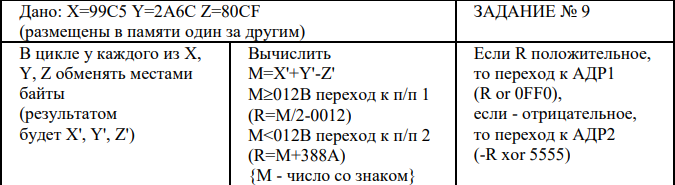
Проверил:\_ к.т.н. доцент кафедры РЭС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ М.А. Земцов /

Киров 2023

**Цель работы:** изучение принципов выполнения команд ветвления,

организации циклов и подпрограмм микропроцессоров с архитектурой x86.

**Задание:**



**Код программы:**

.686

.model flat,stdcall

.stack 100h

.data

X dw 099C5h

Y dw 02A6Ch

Z dw 080CFh

arr dw 099C5h,02A6Ch,080CFh // массив, состоящий из X,Y,Z

M dw ?

R dw ?

L dw ?

.code

ExitProcess PROTO STDCALL :DWORD

Start:

xor eax,eax // обнуление регистров

xor ebx,ebx

xor ecx,ecx

xor edx,edx

;task1 // Задание 1

MOV cx,3;

MOV bx,0

lp:

mov ax, arr[bx]

ROR ax,8 // Цикл меняет местами байты у X,Y,Z в массиве arr

mov arr[bx], ax

add bx,2

loop lp

;task2 // Задание 2

xor ebx,ebx

mov dx,arr[0] // записываем в регистр dx значение первого элемента массива (X’)

mov bx,arr[2] // записываем в регистр bx значение второго элемента массива (Y’)

add dx,bx // X’+Y’

mov cx,arr[4] // записываем в регистр сx значение третьего элемента массива (Z’)

sub dx,cx // (X’+Y’)-Z’

mov M,dx // M=(X’+Y’)-Z’

mov ax,M

cmp ax,012Bh // сравниваем M и 012B

jge com1 // если >=, то переход к com1, если <, то программа выполняется дальше

mov dx,388Ah

add ax,dx

mov R,ax // R=M+388A

jmp com2 // переход к com2

com1:

mov dx,0

mov ax,M

mov bx,2

div bx

mov dx,0012h

sub ax,dx

mov R,ax // R=M/2-0012

com2:

;task3 //Задание 3

mov ax,R

cmp ax,0h // сравниваем R и 0

jg adr1 // если R>0, то переходим к adr1

jl adr2 // если R<0, то переходим к adr2

adr1:

or ax,0FF0h

mov L,ax // L= R or 0FF0

jmp exit

adr2:

not R

mov cx, R

add cx,1

mov bx,5555h

xor cx,bx

mov L,cx // L= -R xor 5555

exit:

Invoke ExitProcess,L // результат

End Start

**Результат работы программы:**

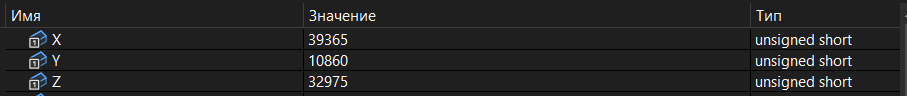
****

Рисунок 1 – Исходные данные

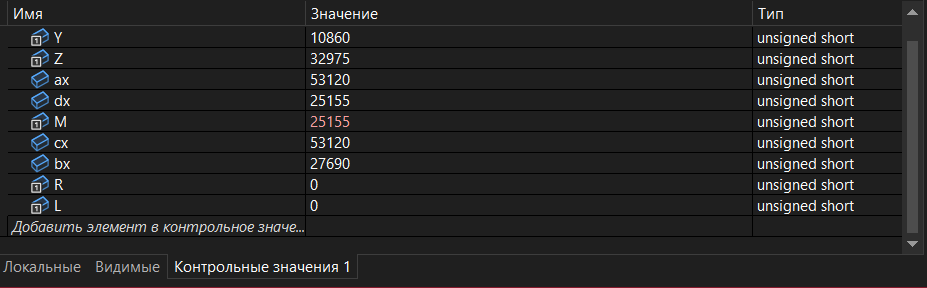


Рисунок 2 – полученное значение M

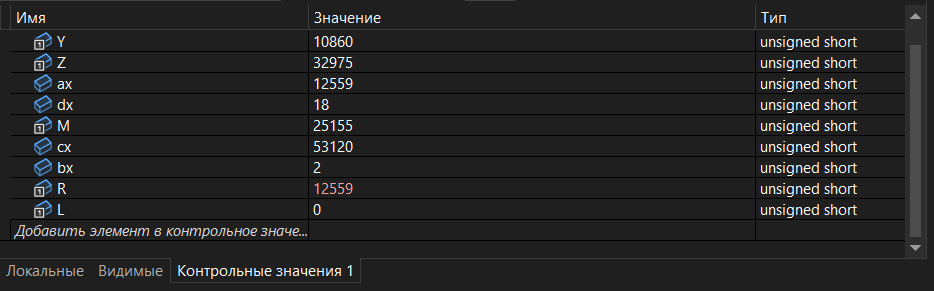


Рисунок 3 – полученное значение R

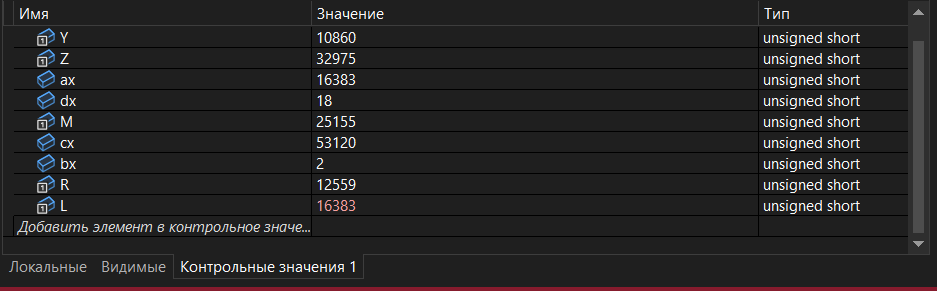


Рисунок 4 – полученное значение L

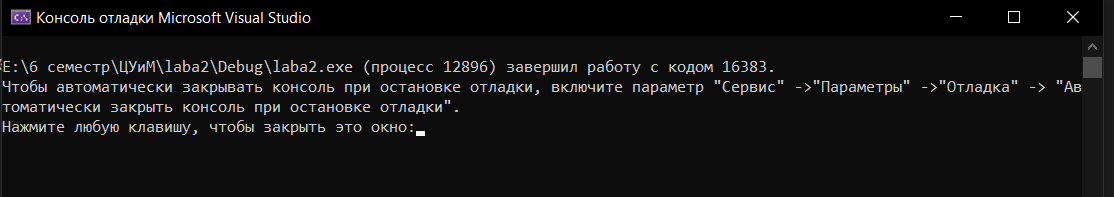


Рисунок 5 – результат работы программы

**Расчет вручную:**

1. **У каждого X,Y,Z обменять местами байты**

X= 99C516 = 3936510 = 10011001110001012

Y=2A6C16 = 1086010 = 00101010011011002

Z=80CF16 = 3297510 = 10000000110011112

Сдвиг на 8 бит: X’= 11000101100110012 = 5058510

Сдвиг на 8 бит: Y’= 01101100001010102 = 2769010

Сдвиг на 8 бит: Z’=11001111100000002 = 5312010

1. **Вычислить M=X’+Y’-Z’**

M= 50585+27690-53120=**25155**

1. **Условие 1**

012B16 = 29910

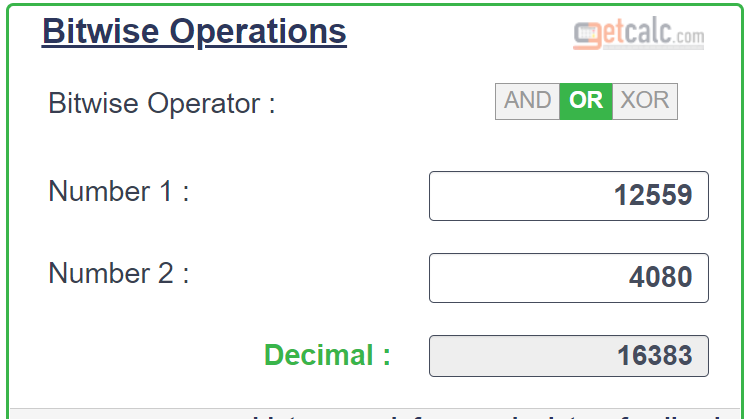
001216 = 1810

M > 012B (25155 > 299), значит выполняется п/п 1

R=M/2-0012 = 25155 / 2 -18 = **12559**

1. **Условие 2**

R > 0, значит выполняется переход к АДР1



0FF016 = 408010

L= R or 0FF0 = 12559 or 4080 = **16383**

**При каком X программа пойдет по противоположным ветвям?**

Например: X = FF00

**Расчет вручную:**

1. **У каждого X,Y,Z обменять местами байты**

X= FF0016 = 6528010 = 11111111000000002

Y=2A6C16 = 1086010 = 00101010011011002

Z=80CF16 = 3297510 = 10000000110011112

Сдвиг на 8 бит: X’= 00000000111111112 = 25510

Сдвиг на 8 бит: Y’= 01101100001010102 = 2769010

Сдвиг на 8 бит: Z’=11001111100000002 = 5312010

1. **Вычислить M=X’+Y’-Z’**

M= 255+27690-53120=-**25175**

1. **Условие 1**

012B16 = 29910

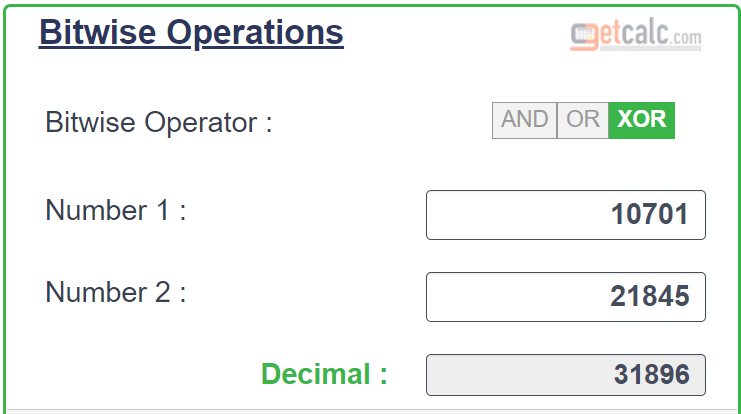
388A16 = 1447410

M < 012B (-25175 < 299), значит выполняется п/п 2

R=M+388A = -25175 + 14474 = -**10701**

1. **Условие 2**

R < 0, значит выполняется переход к АДР2



555516 = 2184510

L= -R xor 5555 = -(-10701) xor 21845 = **31896**

**Результат работы программы:**

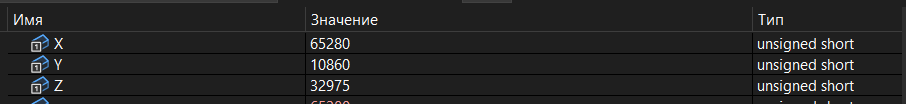


Рисунок 6 – Исходные данные

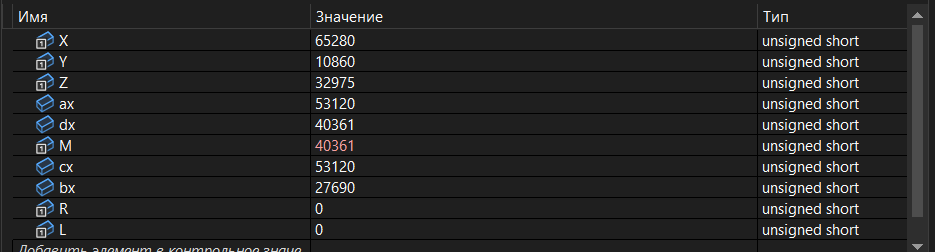
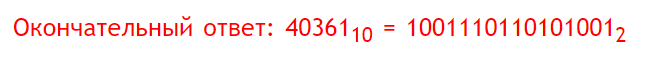
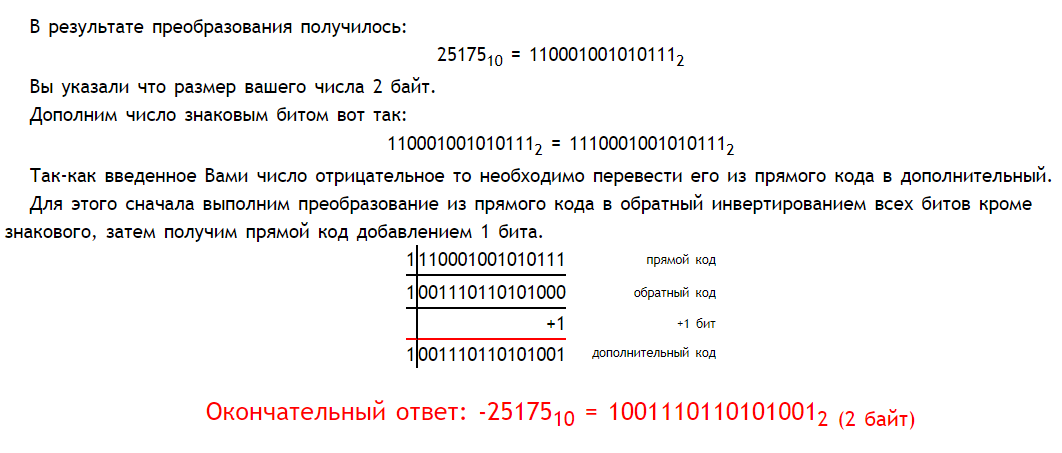


Рисунок 7 – полученное значение M

40361=-25175





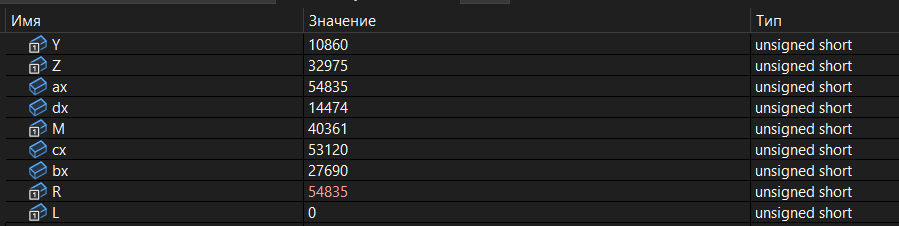
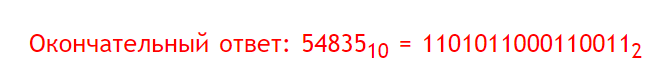
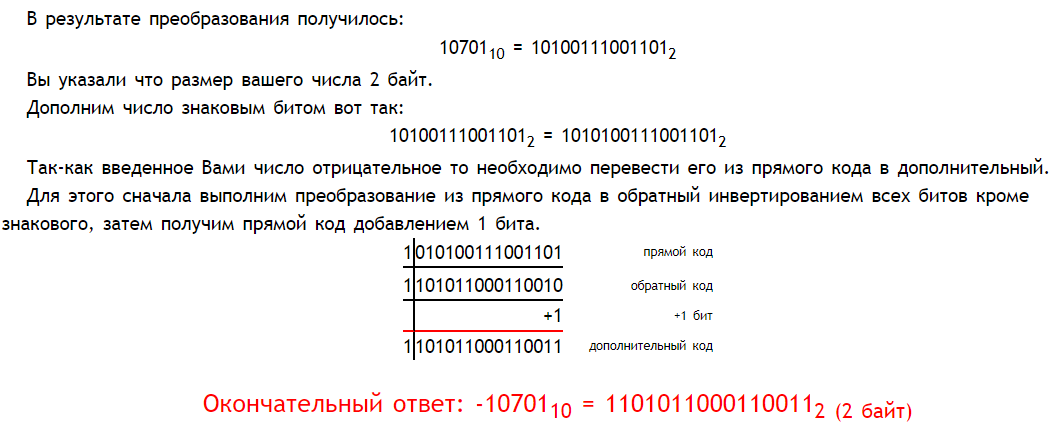


Рисунок 8 – полученное значение R

54835=-10701





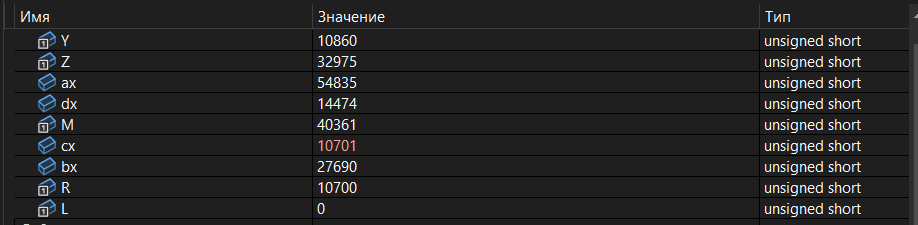


Рисунок 9 – полученное значение -R

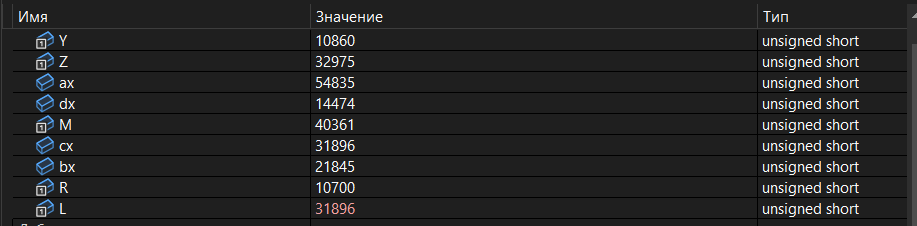


Рисунок 10 – полученное значение L

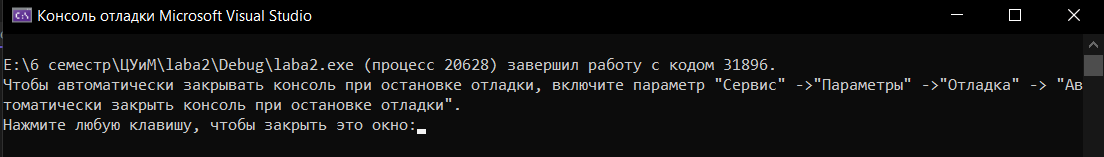


Рисунок 11 – результат работы программы

**Вывод:**

* изучены принципы выполнения команд ветвления, организации циклов и подпрограмм микропроцессоров с архитектурой x86
* Результаты расчёта выражения на ассемблере и вручную совпадают