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 20

Выполнил: студент группы ИНБб– 3301\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Д.А. Шатов /

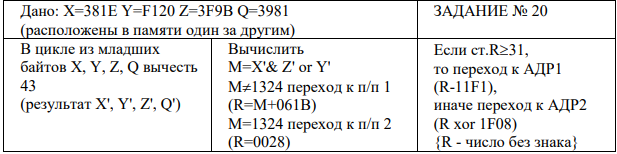
Проверил:\_ к.т.н. доцент кафедры РЭС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ М.А. Земцов /

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 0381eh

Y dw 0f120h

Z dw 03f9bh

Q dw 03981h

ms dw 0381eh, 0f120h, 03f9Bh, 03981h

M dw ?

R dw ?

L dw ?

.code

ExitProcess PROTO STDCALL :DWORD

Start:

xor eax,eax

xor ebx,ebx

xor ecx,ecx

xor edx,edx

;task 1

mov bx,0

mov cx,4

mov ax,ms[0]

lp:

mov ax, ms[bx]

sub al,02bh

mov ms[bx],ax

add bx,2

loop lp

;task 2

mov ax,ms; x

mov bx,ms[2]; y

mov cx,ms[4]; z

and ax,bx

or ax,cx

mov M,ax

cmp M,01324h

jne part

add R,00028h

jmp next

part: add M,0061bh

mov ax,M

mov R,ax

;task 3

next:

mov ax,R

cmp ah,031h

jge adr1

jnge adr2

adr1:

sub ax,011f1h

mov L,ax

jmp exit

adr2:

mov bx,01f08h

xor ax,bx

mov L,ax

exit:

Invoke ExitProcess,L

End Start

**Результат работы программы:**

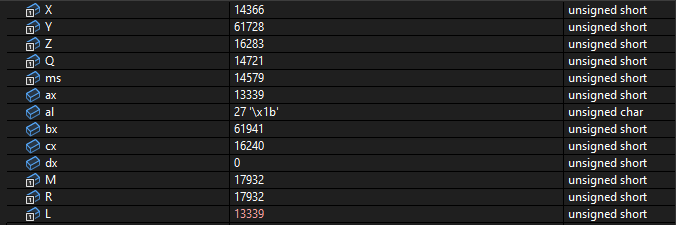


Рисунок 1 – Исходные данные

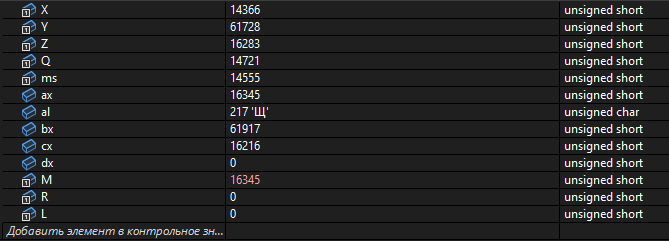


Рисунок 2 – Полученное значение M



Рисунок 3 – Полученное значение R



Рисунок 2 – Полученное значение L

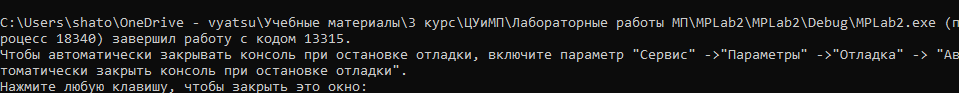


Рисунок 2 – результат работы программы

**Расчет вручную:**

1. **В цикле из младших байтов X, Y, Z вычесть 43**

X= 391E16 = 1436610 = 111000000111102

Y=F12016 = 6172810 = 11110001001000002

Z=3F9B16 = 1628310 = 111111100110112

Вычитание из мл. байта: млX’= млX – 4316 = 000111102 – 10000112 = 110110112 = 21910 => X’ = 111000110110112

Вычитание из мл. байта: млY’= млY – 4316 = 001000002 – 10000112 = 110111012 = 22110 => Y’ = 11110001110111012

Вычитание из мл. байта: млZ’= млZ – 4316 = 100110112 – 10000112 = 010110002 = 8810 => Z’ = 111111010110002

1. **Вычислить M=X’ & Y’ or Z’**

M = 111000110110112 & 11110001110111012 or 111111010110002 =00111111110110012 = **1634510**

1. **Условие 1**

132416 = 490010

061B16 = 156310

M ≠ 012B (16345 ≠ 1563), значит выполняется п/п 1

R = M + 061B = 1634510 + 156310 = **1790810**

1. **Условие 2**

ст.R ≥ 3116(4910), значит выполняется переход к АДР1

L = R – 11F116 = 1790810 – 459310 = **1331510**

1. **При каком X и Z программа пойдет по противоположным ветвям?  
   (только X невозможен, т.к. вычисляется M=X’ & Y’ or Z’, где Z’ > 156310)**

Например: X = 43 и Z = 1367

**Расчет вручную:**

1. **В цикле из младших байтов X, Y, Z вычесть 43**

X= 4316 = 6710 = 00000000010000112

Y= F12016 = 6172810 = 11110001001000002

Z= 136716 = 496710 = 00010011011001112

Вычитание из мл. байта: млX’= млX – 4316 = 010000112 – 10000112 = 02 = 010 => X’ = 00000000000000002

Вычитание из мл. байта: млY’= млY – 4316 = 001000002 – 10000112 = 110111012 = 22110 => Y’ = 11110001110111012

Вычитание из мл. байта: млZ’= млZ – 4316 = 11001112 – 10000112 = 110112 = 3610 => Z’ = 00010011001001002

1. **Вычислить M=X’ & Y’ or Z’**

M = 00000000000000002 & 11110001110111012 or 000100110010010012 =1100001101112 = **490010**

1. **Условие 1**

132416 = 490010

061B16 = 156310

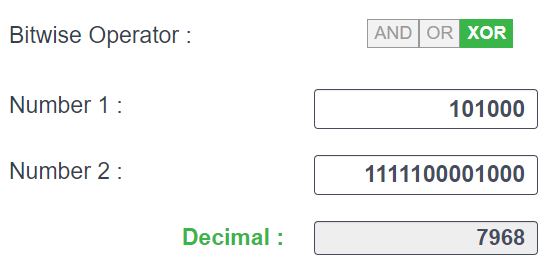
M ≠ 1324 (4900 ≠ 4900), значит выполняется п/п 2

R = 2816 = **4010**

1. **Условие 2**

ст.R ≥ 3116(4010<4910), значит выполняется переход к АДР2

L = R xor 1F0816 = 1010002 xor 11111000010002 = **796810**



**Результат работы программы:**

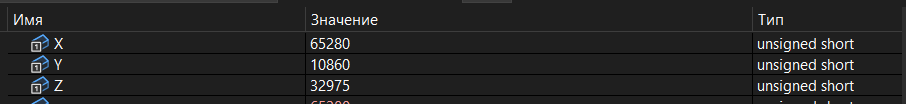


Рисунок 6 – Исходные данные



Рисунок 7 – полученное значение M



Рисунок 8 – полученное значение R



Рисунок 9 – полученное значение L

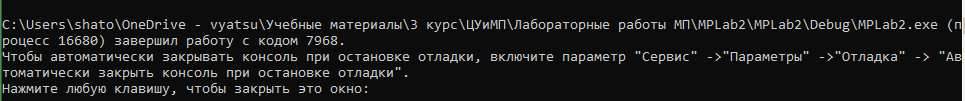


Рисунок 11 – результат работы программы

**Вывод:**

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