ФГБОУ ВО «ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

«ПРИНЦИПЫ ВЫПОЛНЕНИЯ КОМАНД ВЕТВЛЕНИЯ, ОРГАНИЗАЦИЯ ЦИКЛОВ И ПОДПРОГРАММ»

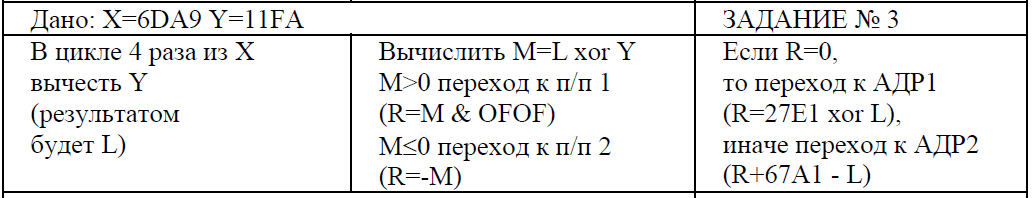
Лабораторная работа № 2

Вариант 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Выполнил: | студент группы ИНБс-33 | \_\_\_\_\_\_\_\_\_\_\_ |  | А.Ю. Бердников |
|  |  |  |  |  |
| Проверил: | доцент кафедры РЭС | \_\_\_\_\_\_\_\_\_\_\_ |  | М.А. Земцов |

Киров 2025

1. **Задание**

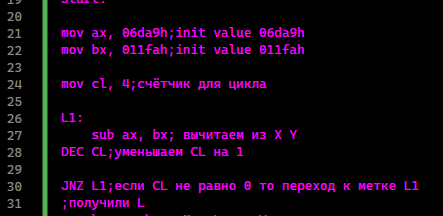


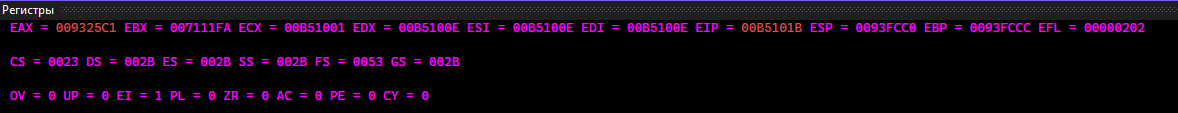
1. **Решение**

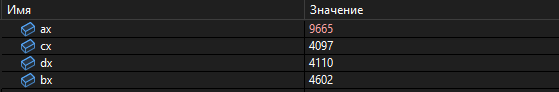
1) Работа программы при заданных значениях X и Y

1. В цикле 4 раза из Х вычитаем Y, результатом будет L

Вручную: L = X – 4\* Y = 25C1 = 9665.



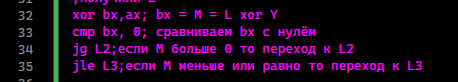


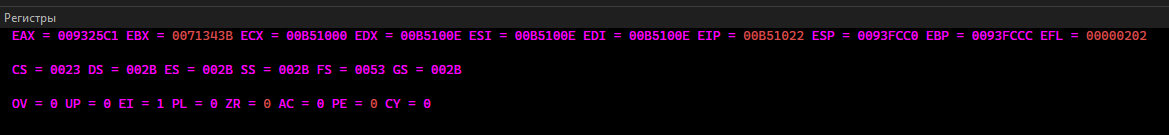


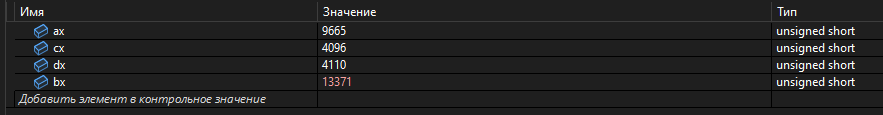
Значение регистра ax совпадает с рассчитанным вручную.

2. Вычисляем M = L xor Y

Вручную: M = L xor Y = 343B = 13371





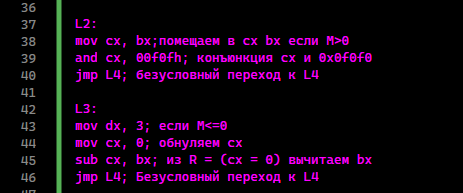


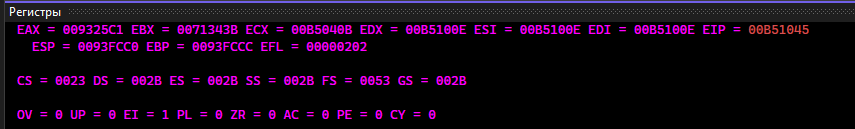
Значение регистра bx совпадает с рассчитанным вручную

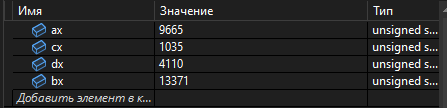
3. Если M > 0, то переход к метке L2, где R = M & 0f0f; Если M ≤ 0, то переход к метке L3, где R = -M. И переходим к метке L4.

Так как M = 13371 > 0, то мы переходим к метке L2.

R = M and 0f0f = 40B = 1035.





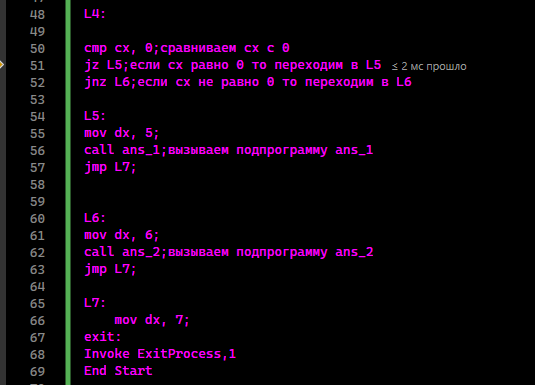


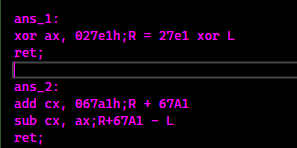
Значение регистра cx совпадает с рассчитанным вручную

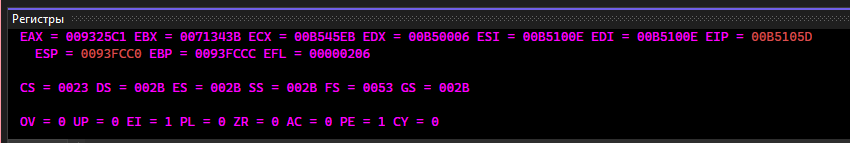
4. Если R = 0, то переходим к метке L5 подпрограмме anc\_1, где R = 27E1 xor L; если R ≠ 0, то переходим к подпрограмме anc\_2, где ответом будет R + 67A1 – L.

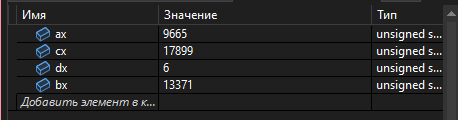
Так как R = 1035 ≠ 0, то переходим к подпрограмме anc\_2 и

ответ = R + 67A1 – L = 45EB = 17899.









Значение регистра cx совпадает с рассчитанным вручную

**Вывод**: алгоритм программы работает верно. Были изучены принципы выполнения команд ветвления, организации циклов и подпрограмм микропроцессоров с архитектурой x86.

**Программный код:**

.686

.model flat,stdcall

.stack 100h

.data

.code

ExitProcess PROTO STDCALL :DWORD

ans\_1:

xor ax, 027e1h;R = 27e1 xor L

ret;

ans\_2:

add cx, 067a1h;R + 67A1

sub cx, ax;R+67A1 - L

ret;

Start:

mov ax, 06da9h;init value 06da9h

mov bx, 011fah;init value 011fah

mov cl, 4;счётчик для цикла

L1:

sub ax, bx; вычитаем из X Y

DEC CL;уменьшаем CL на 1

JNZ L1;если CL не равно 0 то переход к метке L1

;получили L

xor bx,ax; bx = M = L xor Y

cmp bx, 0; сравниваем bx с нулём

jg L2;если M больше 0 то переход к L2

jle L3;если M меньше или равно то переход к L3

L2:

mov cx, bx;помещяем в cx bx

and cx, 00f0fh; конъюнкция cx и 0x0f0f0

jmp L4; безусловный переход к L4

L3:

mov dx, 3;

mov cx, 0; обнуляем cx

sub cx, bx; из R = (cx = 0) вычитаем bx

jmp L4; Безусловный переход к L4

L4:

cmp cx, 0;сравниваем cx с 0

jz L5;если cx равно 0 то переходим в L5

jnz L6;если сx не равно 0 то переходим в L6

L5:

mov dx, 5;

call ans\_1;вызываем подпрограмму ans\_1

jmp L7;

L6:

mov dx, 6;

call ans\_2;вызываем подпрограмму ans\_2

jmp L7;

L7:

mov dx, 7;

exit:

Invoke ExitProcess,1

End Start