МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ

БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

Отчет по дисциплине «Цифровые устройства и МП»

Лабораторной работе №2

«ПРИНЦИПЫ ВЫПОЛНЕНИЯ КОМАНД ВЕТВЛЕНИЯ, ОРГАНИЗАЦИЯ ЦИКЛОВ И ПОДПРОГРАММ»

Выполнил: студенты гр. ИНБб – 3301                                              /И.А.Капустин/  
  
Проверил: педагог кафедры РЭС\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/М.А.Земцов/

Киров 2025

**Цель работы:**

Изучение принципов выполнения команд ветвления, организации циклов и подпрограмм микропроцессоров с архитектурой x86.

**Задание:**

|  |  |  |
| --- | --- | --- |
| Вариант | Исходные данные | Задание |
| 3 | X=6DA9 Y=11FA | |  |  |  | | --- | --- | --- | | В цикле 4 раза из Х вычесть Y (результатом будет L) | Вычислить M=L xor Y M>0 переход к п/п 1 (R=M & OFOF) M≤0 переход к п/п 2 (R=-M) | Если R=0, то переход к АДР1 (R=27E1 xor L), иначе переход к АДР2 (R+67A1 - L) | |

**Текст программы с комментариями:**

.686

.model flat,stdcall

.stack 100h

.data

X dw 28073 ;6DA9

Y dw 4602 ;11FA

I dw 0

M dw ?

L dw ?

R dw ?

.code

ExitProcess PROTO STDCALL :DWORD

Start:

mov ax,X; Запись значения X в регистр ax

for\_start:

cmp I,4 ; условие цикла, что он должен сработать 4 раза

jnl for\_end

sub ax,Y ; вычитание Y из X

add I,1 ; увеличиваем счетчик на 1

jmp for\_start

for\_end:

mov L,ax ;результат вычислений записан в L

mov ax,L

xor ax,Y ; Вычисление выражения M=L xor Y

mov M,ax ; Запись резульата в переменную M

cmp M,0

jg bolshe; Проверка M>0

jle menshe\_ili\_ravno; Проверка M<=0

bolshe:

call Podprog1;Вызов подпрограммы 1

jmp proverka\_R ; Переход к Проверка 0 или не 0

menshe\_ili\_ravno:

call Podprog2; Вызов подпрограммы 2

jmp proverka\_R ; Переход к Проверка 0 или не 0

proverka\_R: ; Проверка 0 или не 0

cmp R,0

je R\_zero; Проверка R=0

jne R\_nozero; Проверка R!=0

R\_zero: ; АДР1

mov ax,10209 ; Запись значения 27E1 в регистр ax

xor ax,L ; Вычисление R=27E1 xor L

mov R,ax ; Запись значения регистра ax в R

R\_nozero: ; АДР2

mov ax,R ; Запись значения R в регистр ax

add ax,26529 ; Вычисление выражения R+67A1 - L

sub ax,L ; Вычисление выражения R+67A1 - L

mov R,ax ; Запись значения регистра ax в R

exit:

Invoke ExitProcess,1

Podprog1 proc ;начало подпрограммы 1

and M,3855; Вычисление выражения R=M & 0F0F

mov R,ax; Запись значения R в регистр ax

ret

Podprog1 endp ;конец подпрограммы 1

Podprog2 proc ;начало подпрограммы 2

neg M; Вычисление выражения R=-M

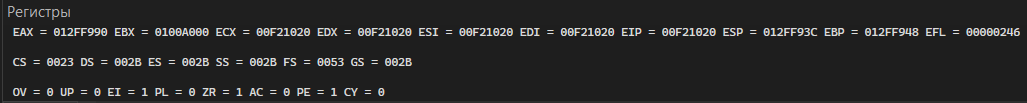
mov R,ax; Запись значения R в регистр ax

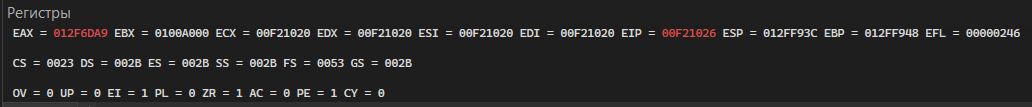
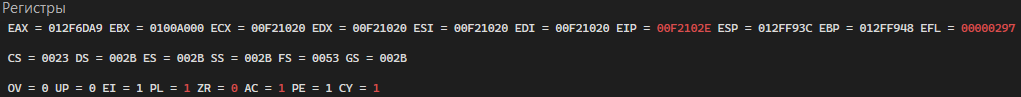
ret

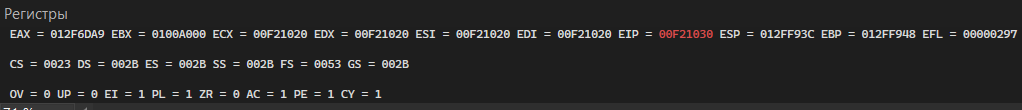
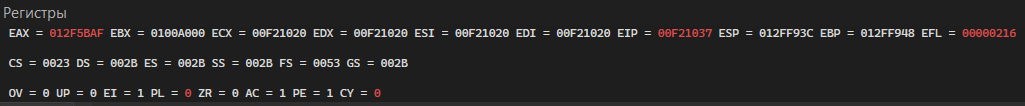
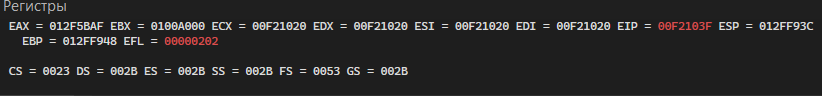
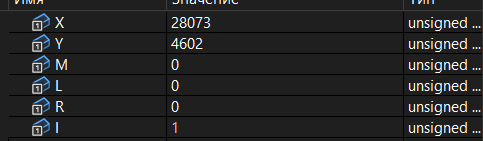
Podprog2 endp ;конец подпрограммы 2

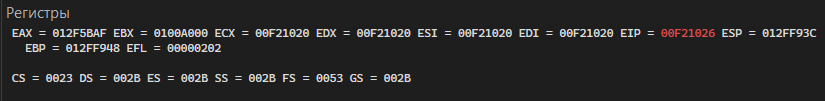
End Start

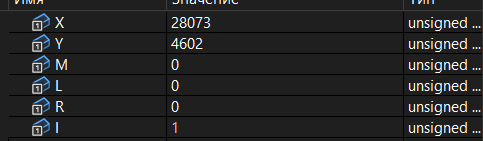
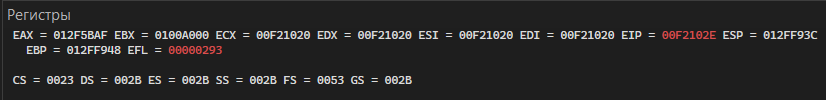
**Верификация программы: результаты расчета заданного выражения, скриншоты, показывающие содержимое регистров и значения переменных после каждого действия программы.**

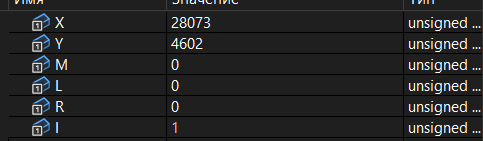
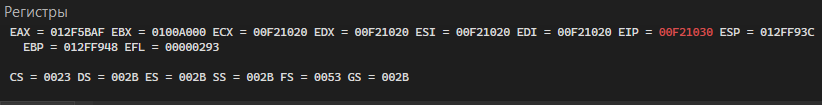
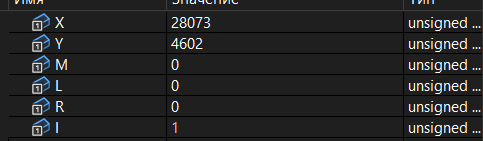
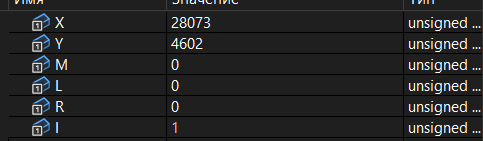
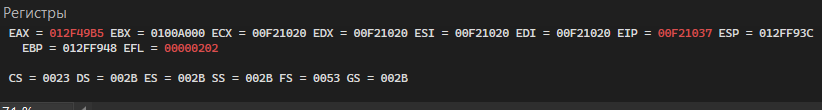
mov ax,X Запись значения X в регистр ax ****for\_start:

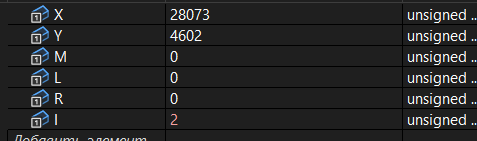
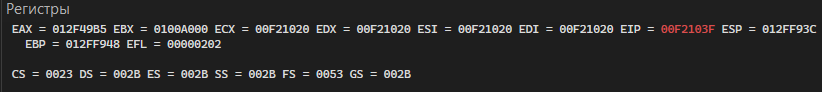
cmp I,4 ; условие цикла, что он должен сработать 4 раза ****jnl for\_end ****

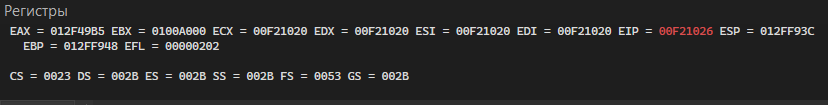
****sub ax,Y ; вычитание Y из X ****add I,1 ; увеличиваем счетчик на 1 ****  
jmp for\_start ****

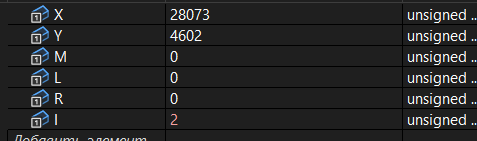
cmp I,4 ; условие цикла, что он должен сработать 4 раза  
****

jnl for\_end  
****

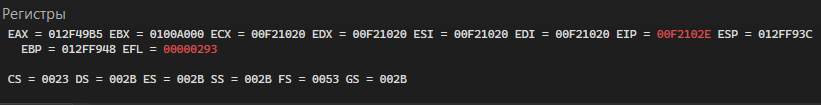
sub ax,Y ; вычитание Y из X  
****  
  
add I,1 ; увеличиваем счетчик на 1  


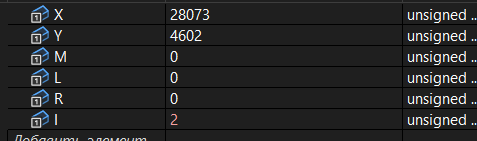
jmp for\_start  
****

cmp I,4 ; условие цикла, что он должен сработать 4 раза  
****

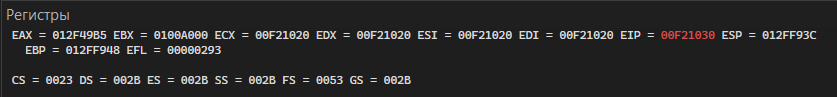
****

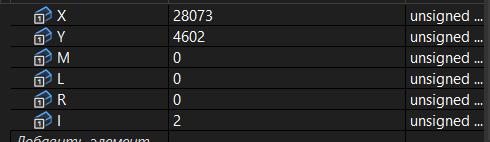
jnl for\_end

****

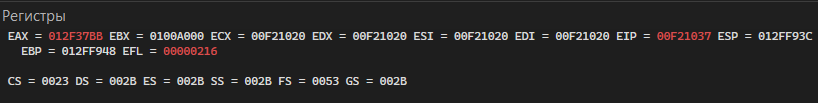
****

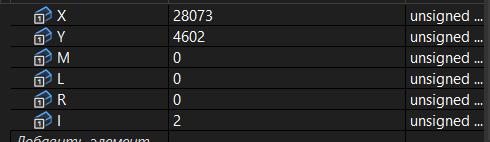
sub ax,Y ; вычитание Y из X

****

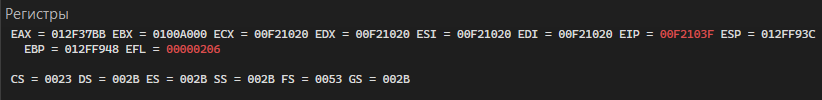
****

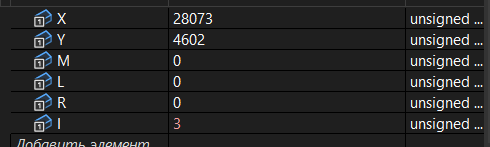
add I,1 ; увеличиваем счетчик на 1

****

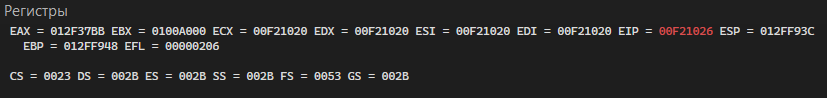
****

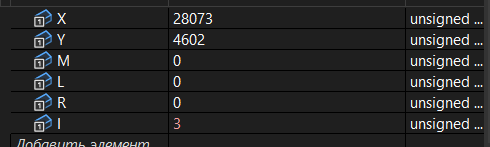
jmp for\_start

****

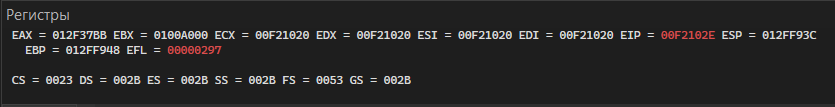
****

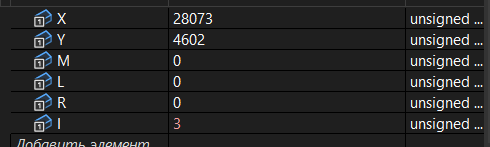
cmp I,4 ; условие цикла, что он должен сработать 4 раза

****

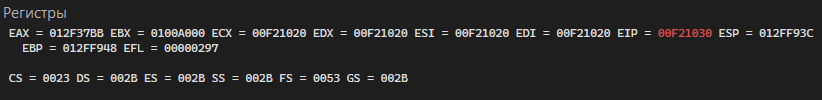
****

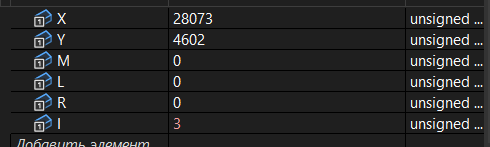
jnl for\_end

****

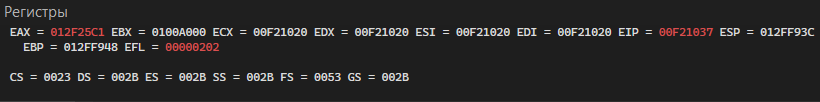
****

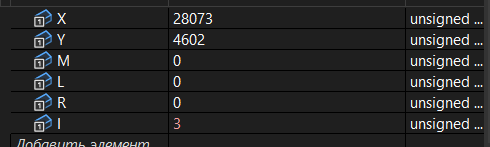
sub ax,Y ; вычитание Y из X

****

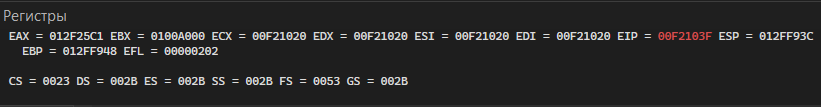
****

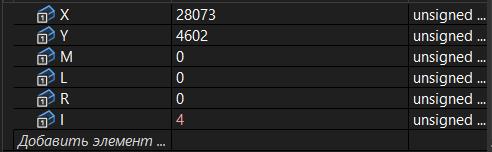
add I,1 ; увеличиваем счетчик на 1

****

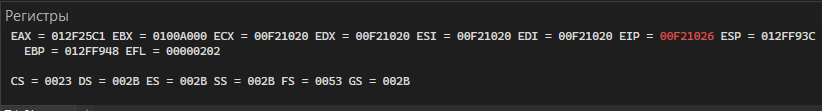
****

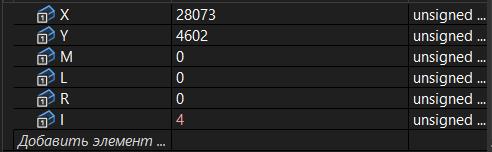
jmp for\_start

****

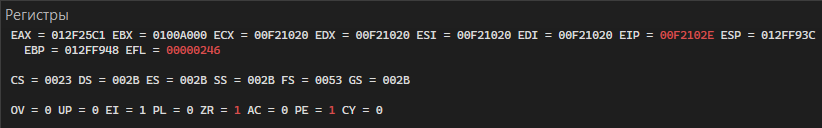
****

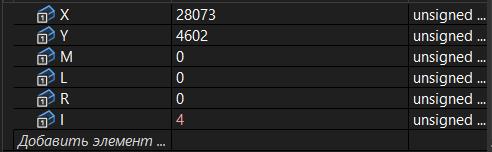
cmp I,4 ; условие цикла, что он должен сработать 4 раза

****

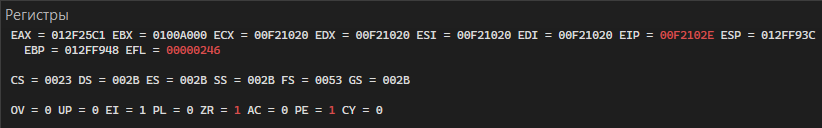
****

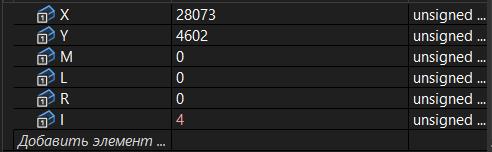
jnl for\_end

****

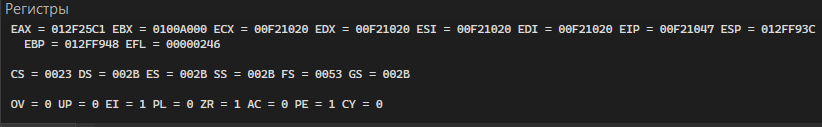
****

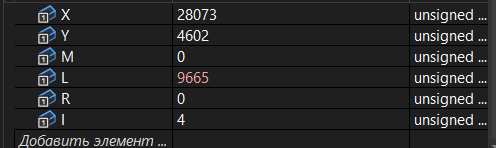
mov L,ax ;результат вычислений записан в L

****

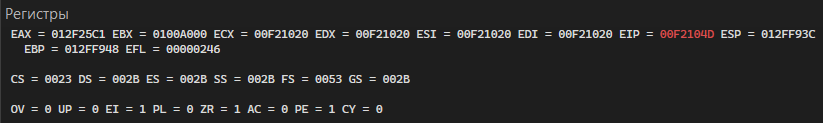
****

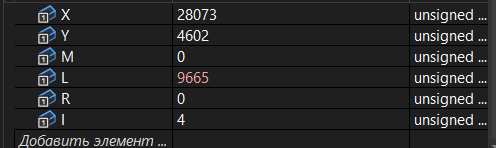
mov ax,L

****

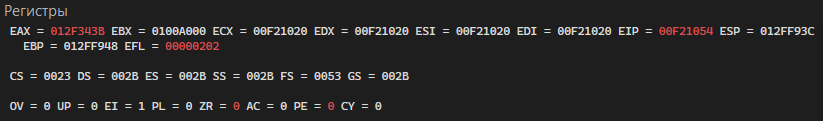
****

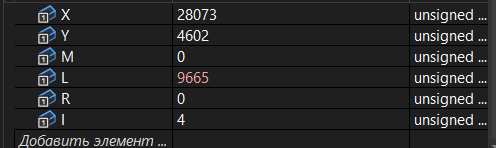
xor ax,Y ; Вычисление выражения M=L xor Y

****

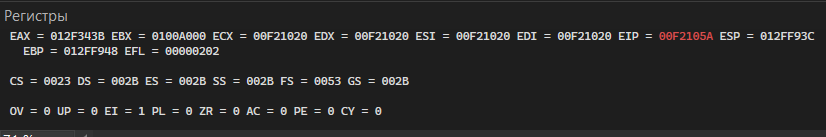
****

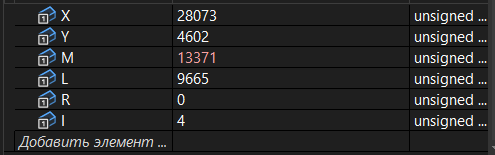
mov M,ax ; Запись резульата в переменную M

****

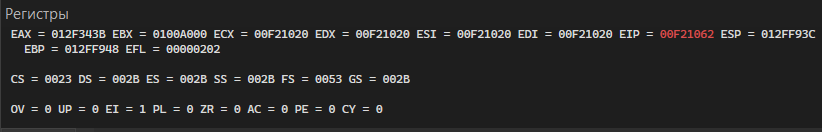
****

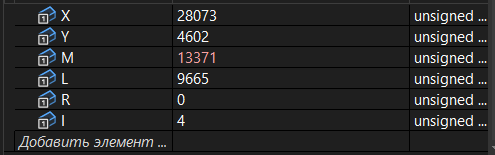
cmp M,0

****

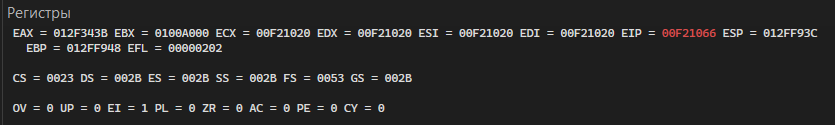
****

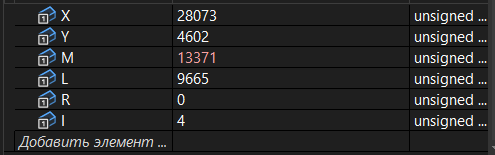
jg bolshe; Проверка M>0

****

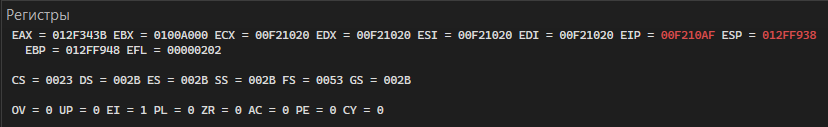
****

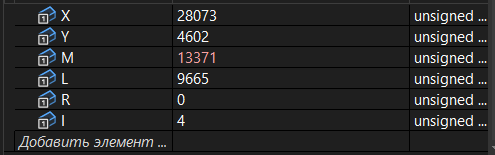
call Podprog1;Вызов подпрограммы 1

****

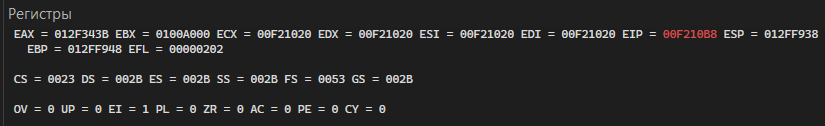
****

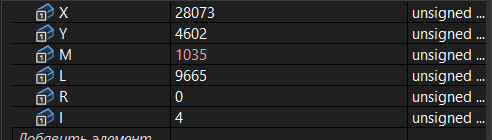
and M,3855; Вычисление выражения R=M & 0F0F

****

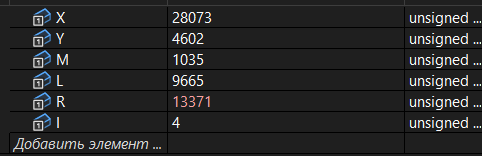
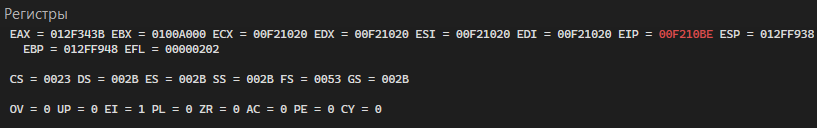
****

mov R,ax; Запись значения R в регистр ax

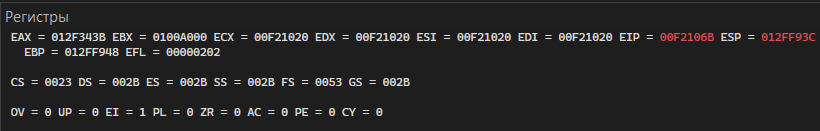
****

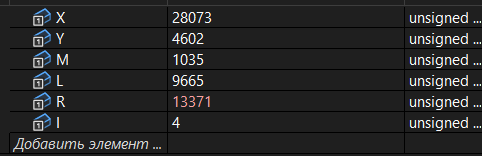
****

ret

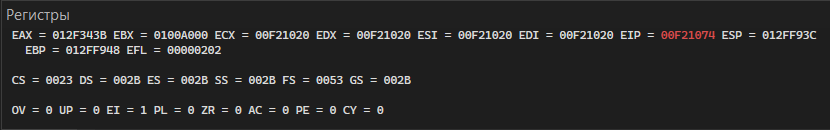
****

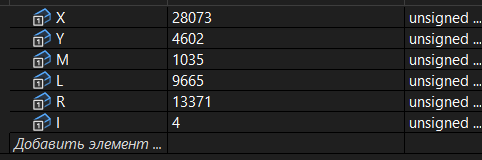
jmp proverka\_R ; Переход к Проверка 0 или не 0

****

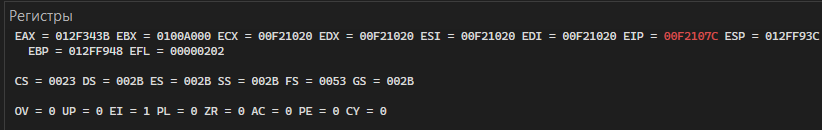
****

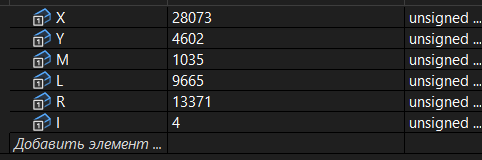
cmp R,0

****

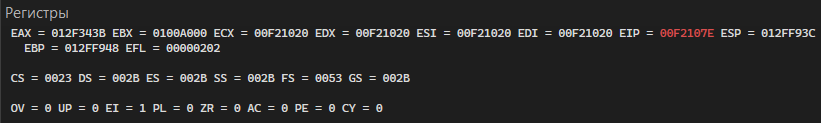
****

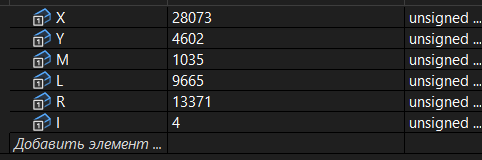
je R\_zero; Проверка R=0

****

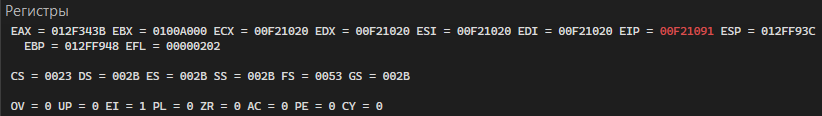
****

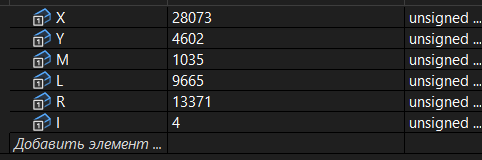
jne R\_nozero; Проверка R!=0

****

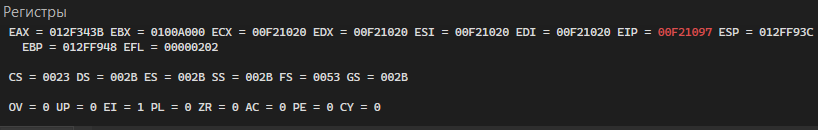
****

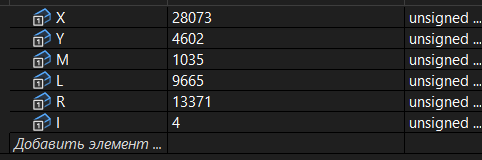
mov ax,R ; Запись значения R в регистр ax

****

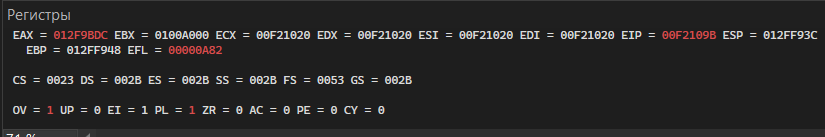
****

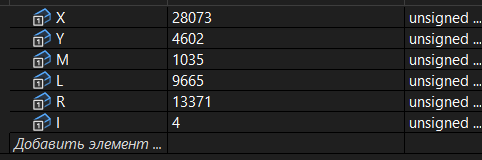
add ax,26529 ; Вычисление выражения R+67A1 - L

****

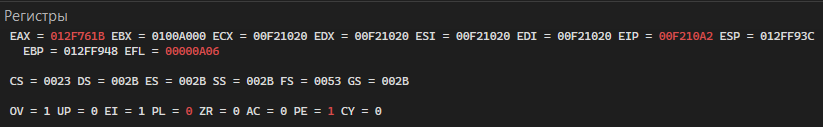
****

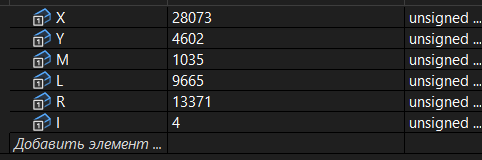
sub ax,L ; Вычисление выражения R+67A1 - L

****

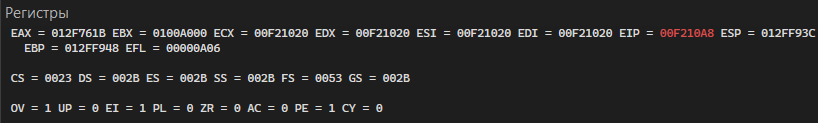
****

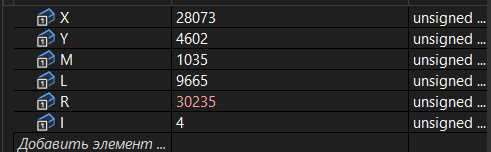
mov R,ax ; Запись значения регистра ax в R

****

****

Invoke ExitProcess,1

****

****

**Вывод:**Я изучил принципы выполнения команд ветвления, организацию циклов и подпрограмм микропроцессоров с архитектурой x86.