МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

Отчет по лабораторной работе №2

по дисциплине «Цифровые устройства и МЦ»

Вариант 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Выполнил: | студент группы  ИНБб-3301-02-00 |  |  | И.Д. Щукин |
|  |  |  |  |  |
| Проверил: | Преподаватель |  |  | М.А. Земцов |
|  |  |  |  |  |

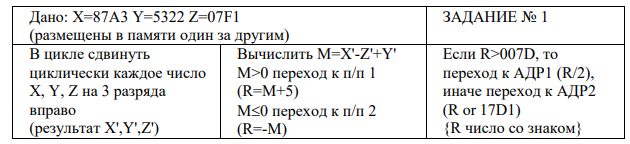
Работа защищена с оценкой «\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_ 2025 г.

г. Киров

2025

1. ЛАБОРАТОРНАЯ РАБОТА № 2 ««ПРИНЦИПЫ ВЫПОЛНЕНИЯ КОМАНД ВЕТВЛЕНИЯ, ОРГАНИЗАЦИЯ ЦИКЛОВ И ПОДПРОГРАММ»

**Цель работы**: изучение принципов выполнения команд ветвления, организации циклов и подпрограмм микропроцессоров с архитектурой x86.



Код программы:

.686

.model flat, stdcall

.stack 100h

.data

X dw 34723 ; Значения в десятичной системе

Y dw 5322

Z dw 2033

M dw 0

R dw 0

.code

ExitProcess PROTO STDCALL :DWORD

Start:

; Циклический сдвиг X, Y, Z на 3 разряда вправо

lea esi, [X] ; Загружаем адрес X в ESI

call shift\_right

lea esi, [Y] ; Загружаем адрес Y в ESI

call shift\_right

lea esi, [Z] ; Загружаем адрес Z в ESI

call shift\_right

; Вычисление M = X' - Z' + Y'

mov ax, [X]

sub ax, [Z]

add ax, [Y]

mov [M], ax

; Проверка M > 0

cmp ax, 0

jg label1 ; Если M > 0, переход к label1

jle label2 ; Если M <= 0, переход к label2

label1:

; R = M + 5

mov ax, [M]

add ax, 5

mov [R], ax

jmp exit ; Переход к завершению программы

label2:

; R = -M

mov ax, [M]

neg ax

mov [R], ax

check\_R:

; Проверка R > 125

cmp ax, 125

jg addr1 ; Если R > 125, переход к addr1

jmp addr2 ; Иначе переход к addr2

addr1:

; R = R / 2

mov ax, [R]

sar ax, 1 ; Арифметический сдвиг вправо на 1 (деление на 2)

mov [R], ax

jmp exit ; Переход к завершению программы

addr2:

; R = R or 6097

mov ax, [R]

or ax, 6097 ; Выполняем побитовую операцию OR с 6097

mov [R], ax

exit:

invoke ExitProcess, 0

; Подпрограмма для циклического сдвига вправо на 3 разряда

shift\_right:

mov ax, [esi] ; Загружаем значение по адресу, указанному в ESI

; Сдвиг вправо на 3 разряда

shr ax, 3

; Циклический сдвиг

mov bx, [esi] ; Сохраняем оригинальное значение в BX

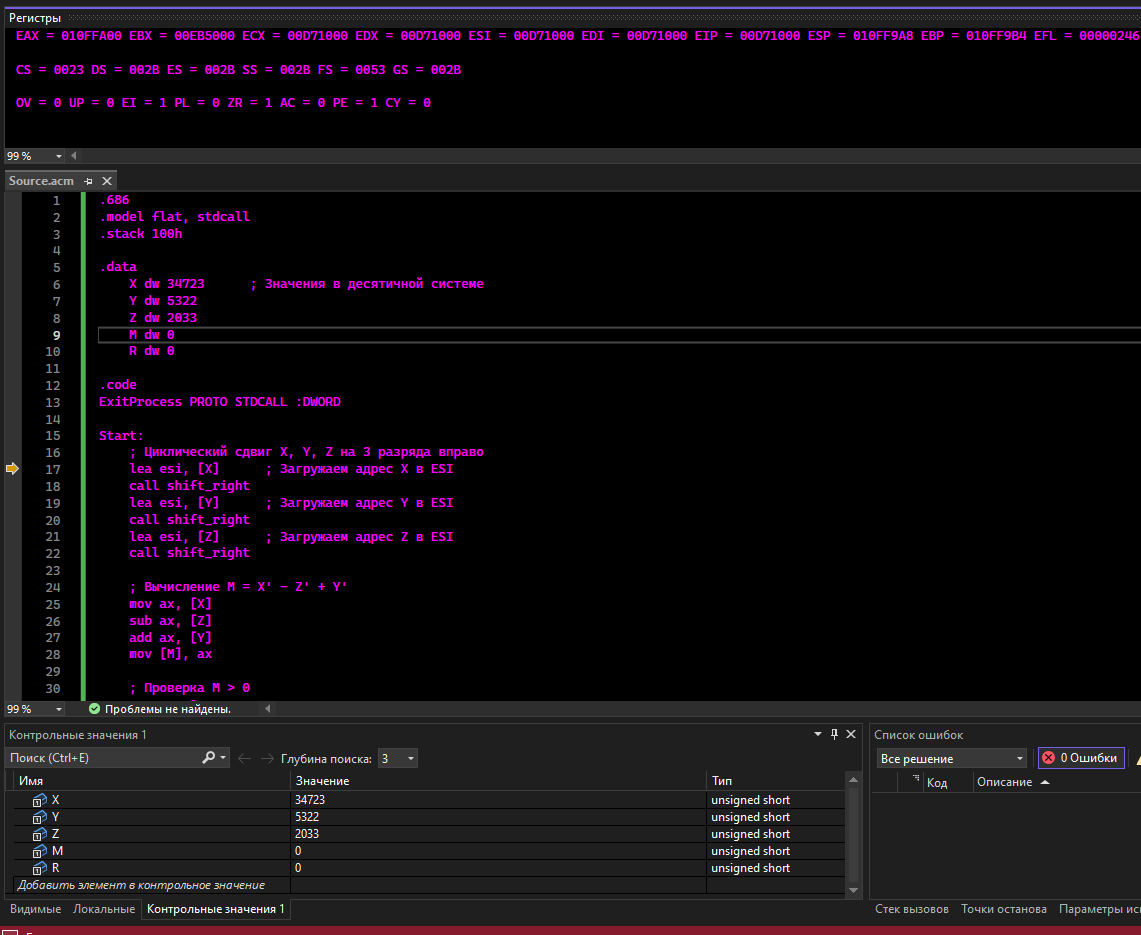
shl bx, 13 ; Сдвигаем оригинальное значение влево на 13 разрядов

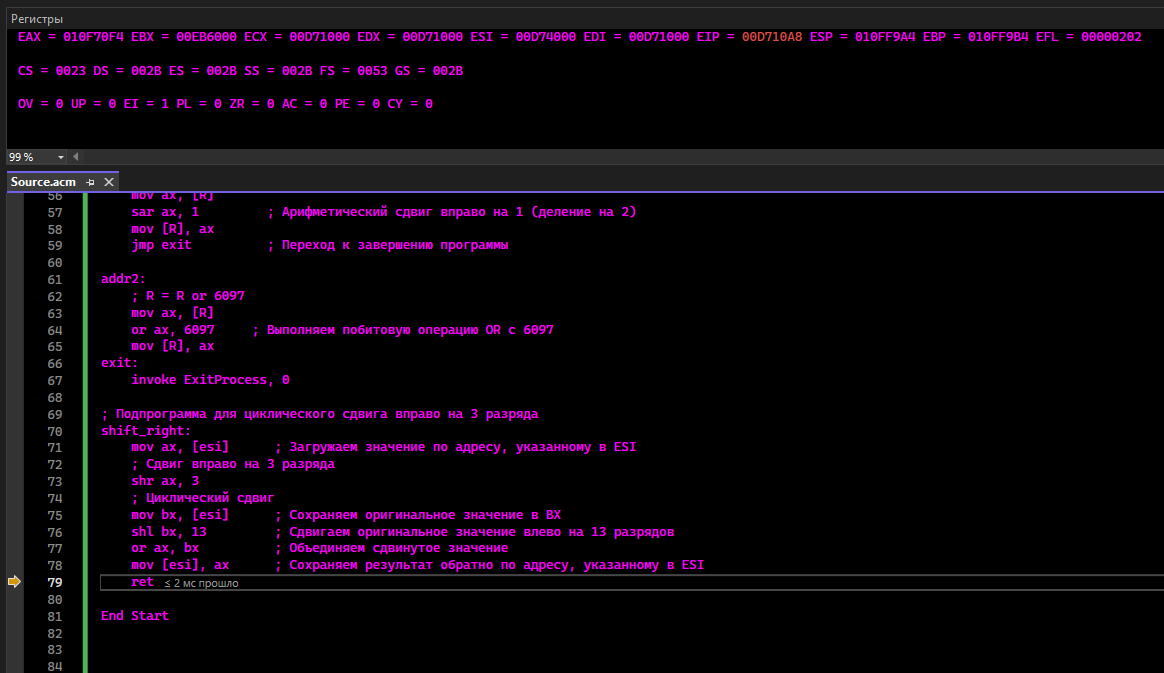
or ax, bx ; Объединяем сдвинутое значение

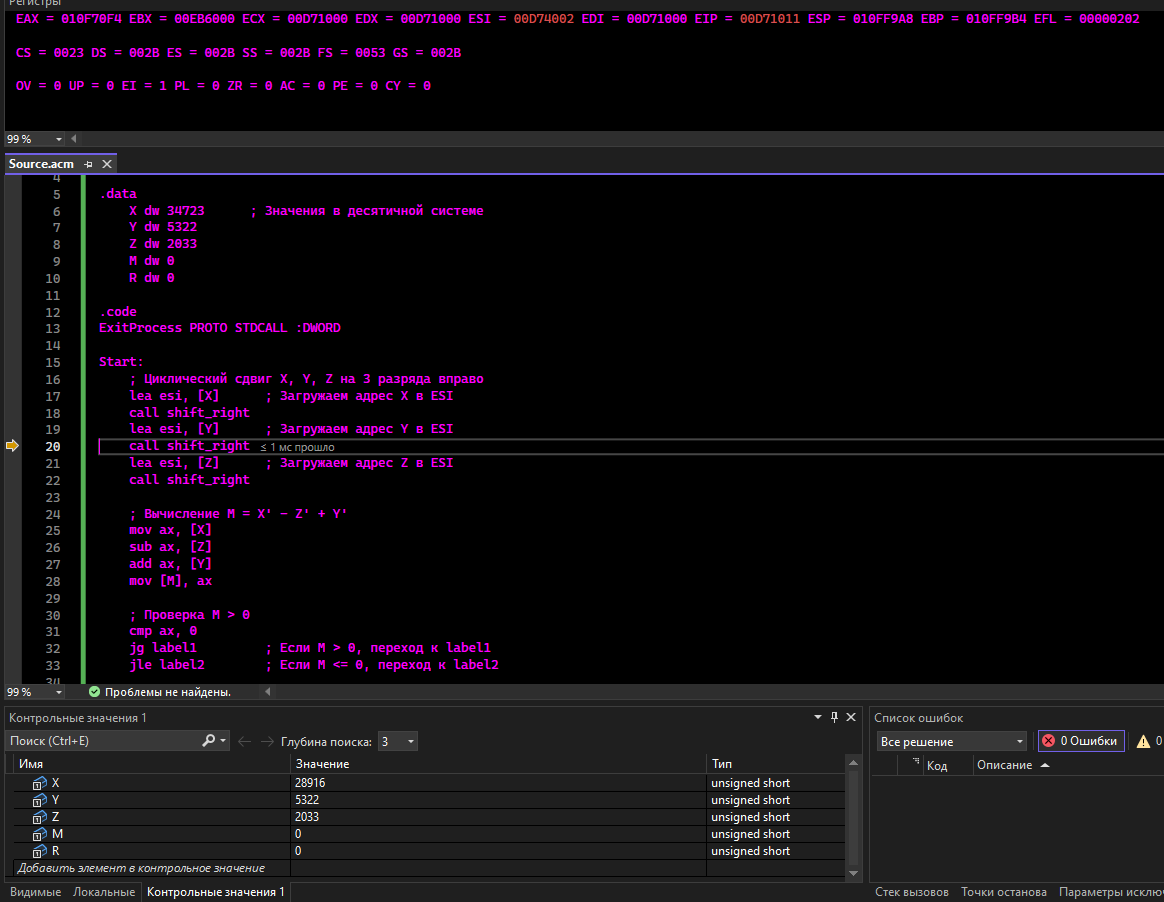
mov [esi], ax ; Сохраняем результат обратно по адресу, указанному в ESI

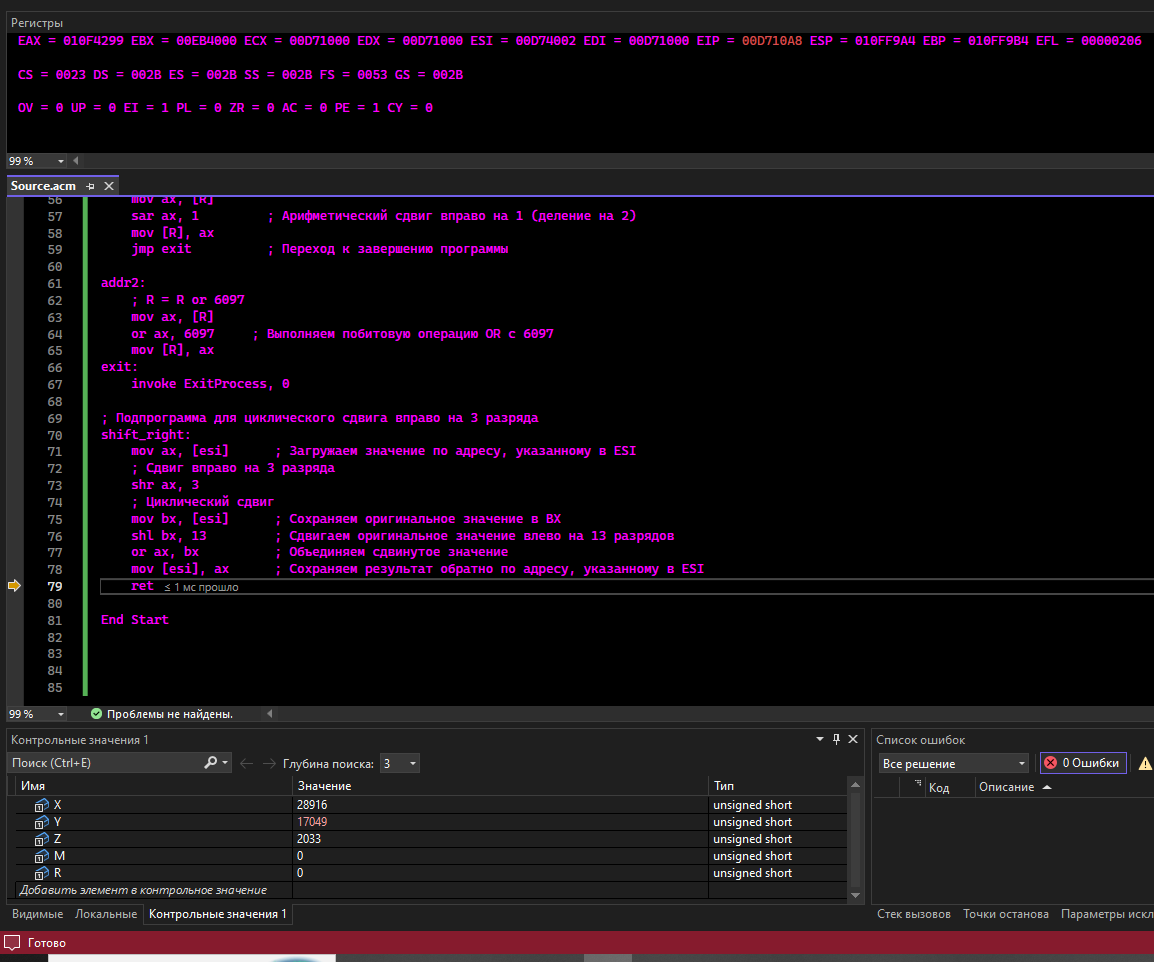
ret

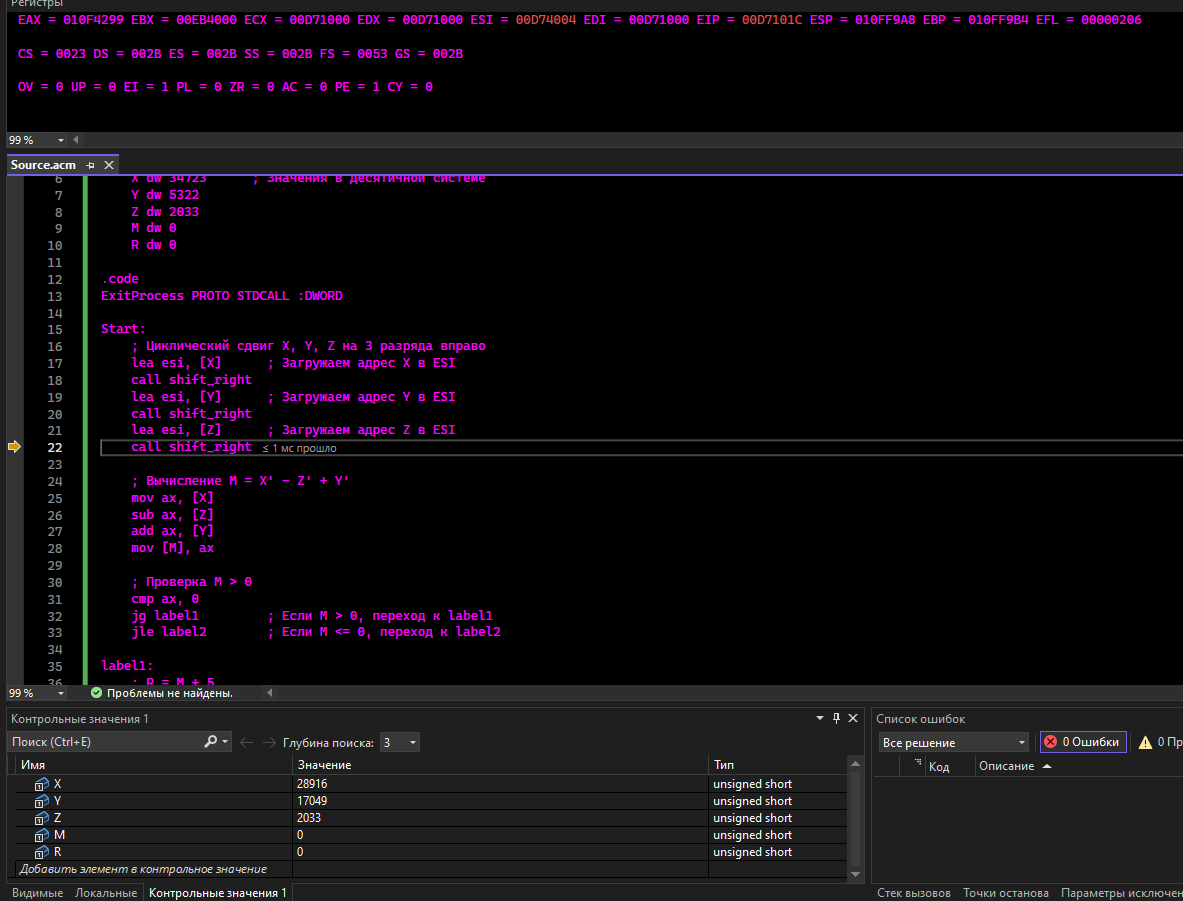
End Start

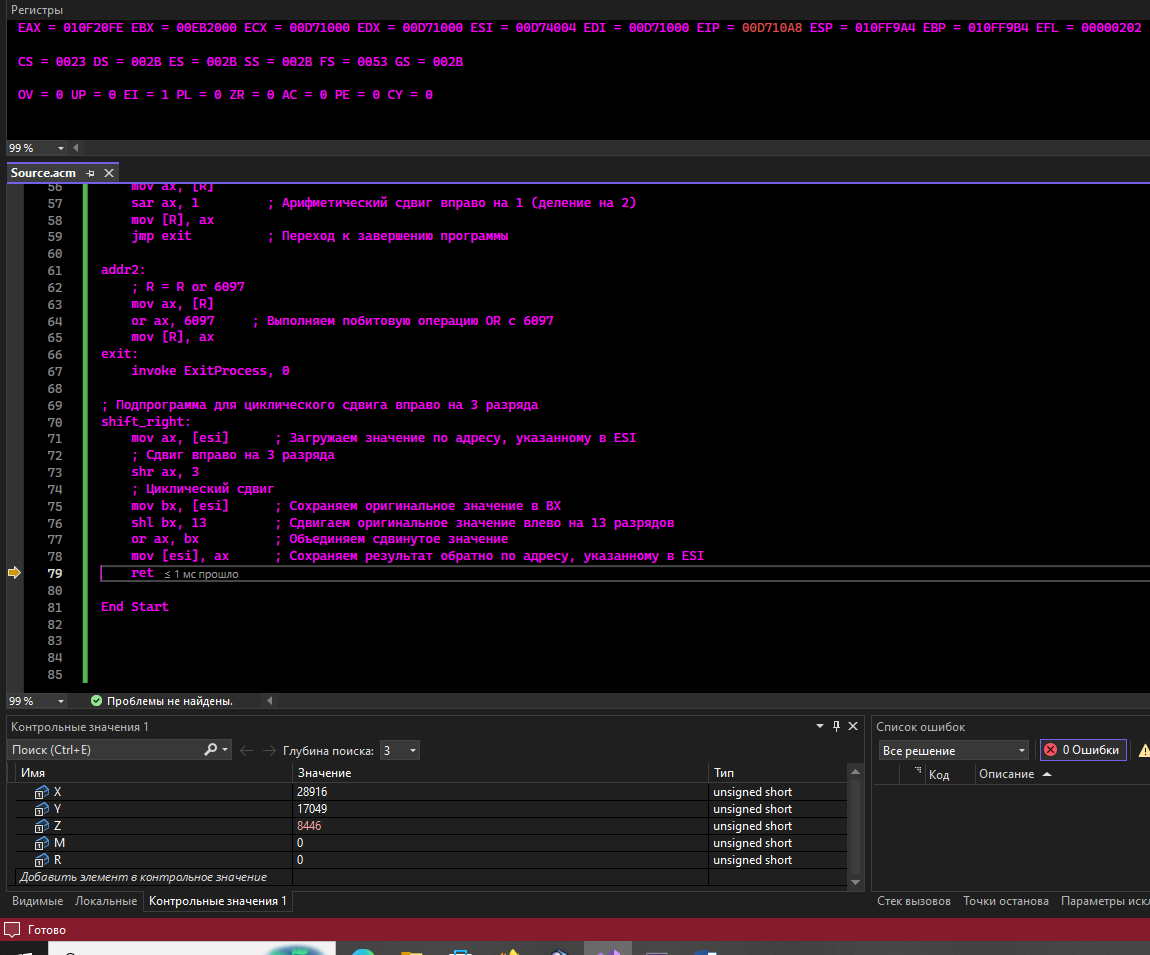


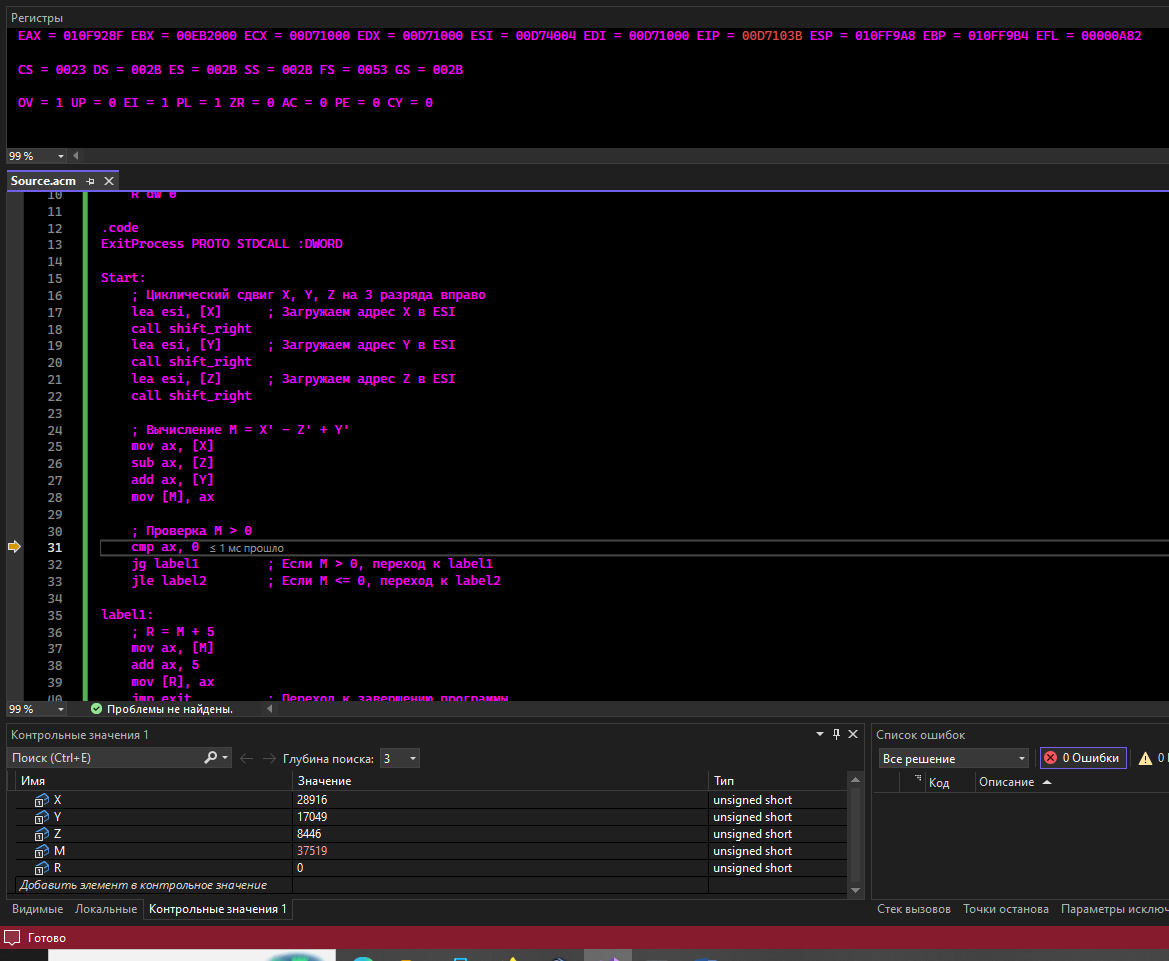


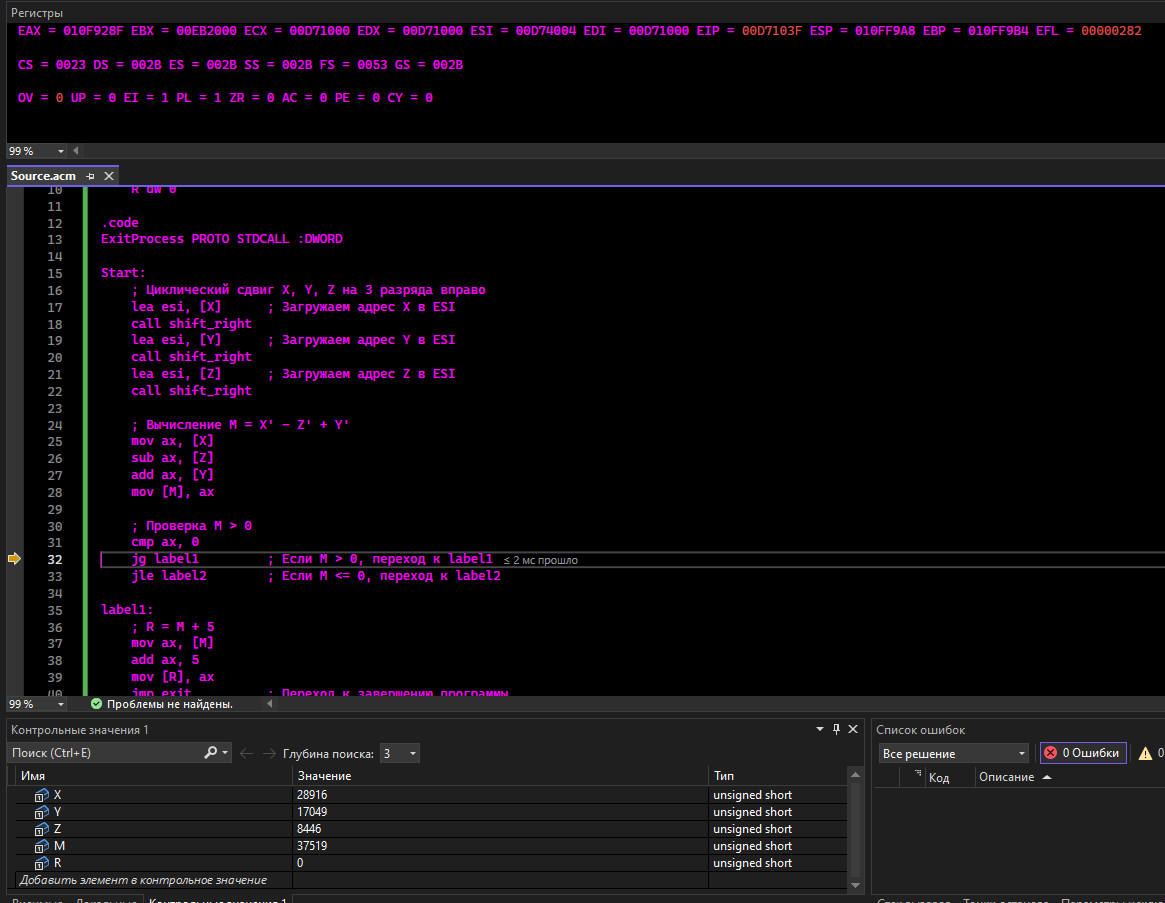


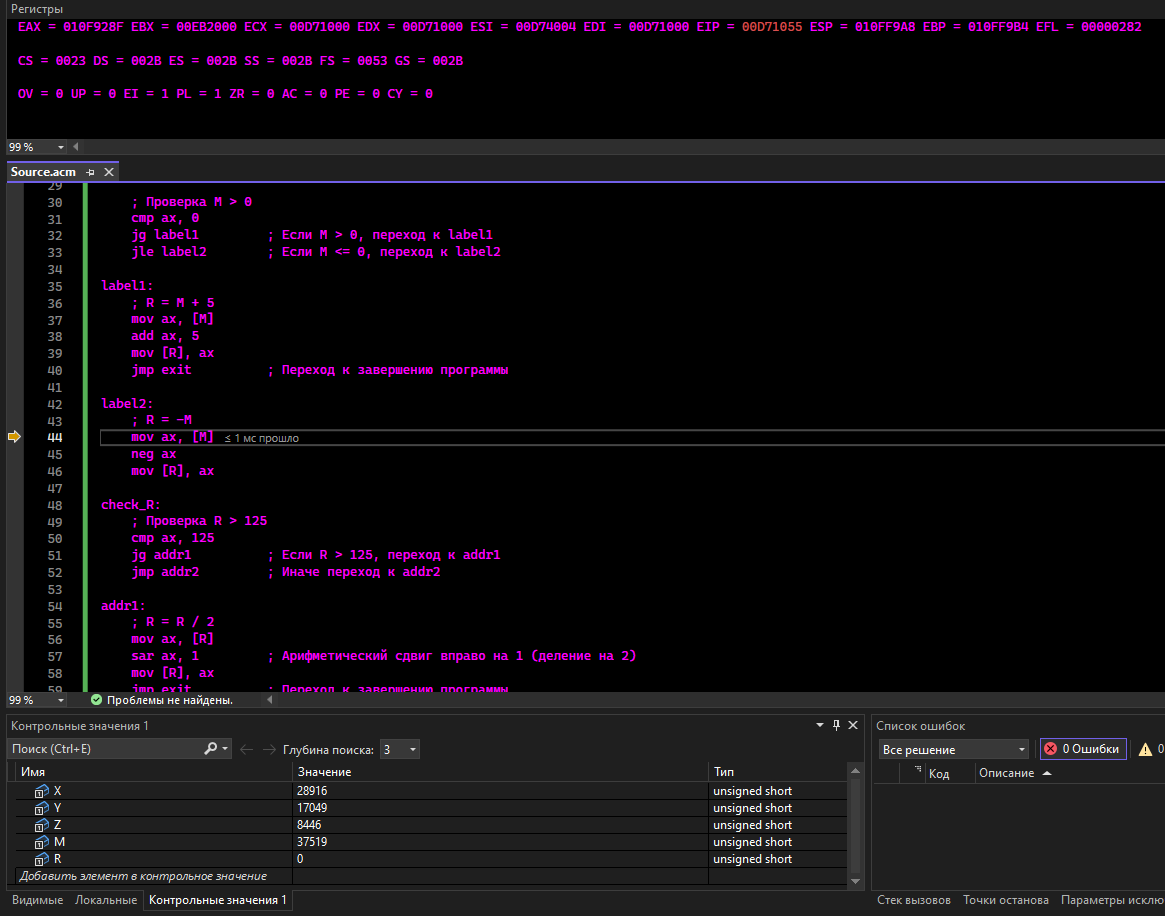


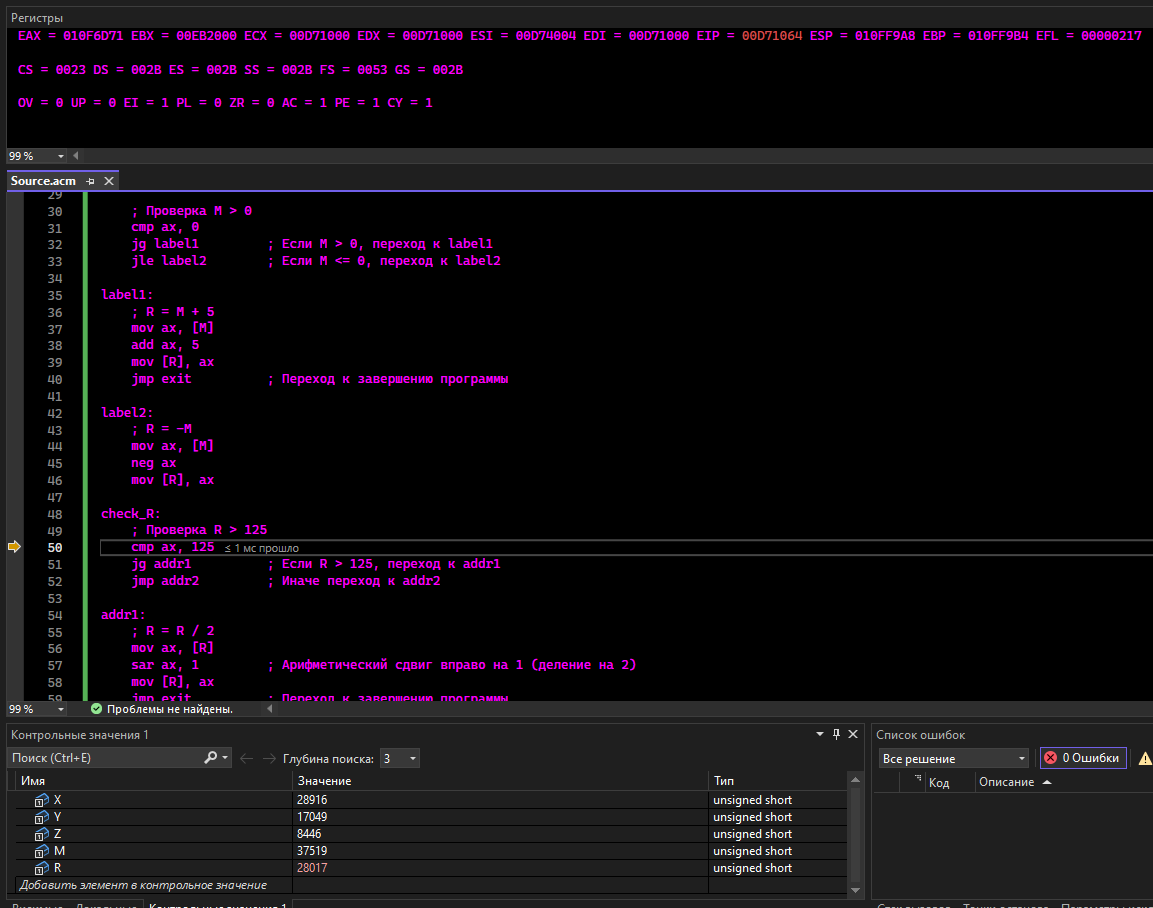


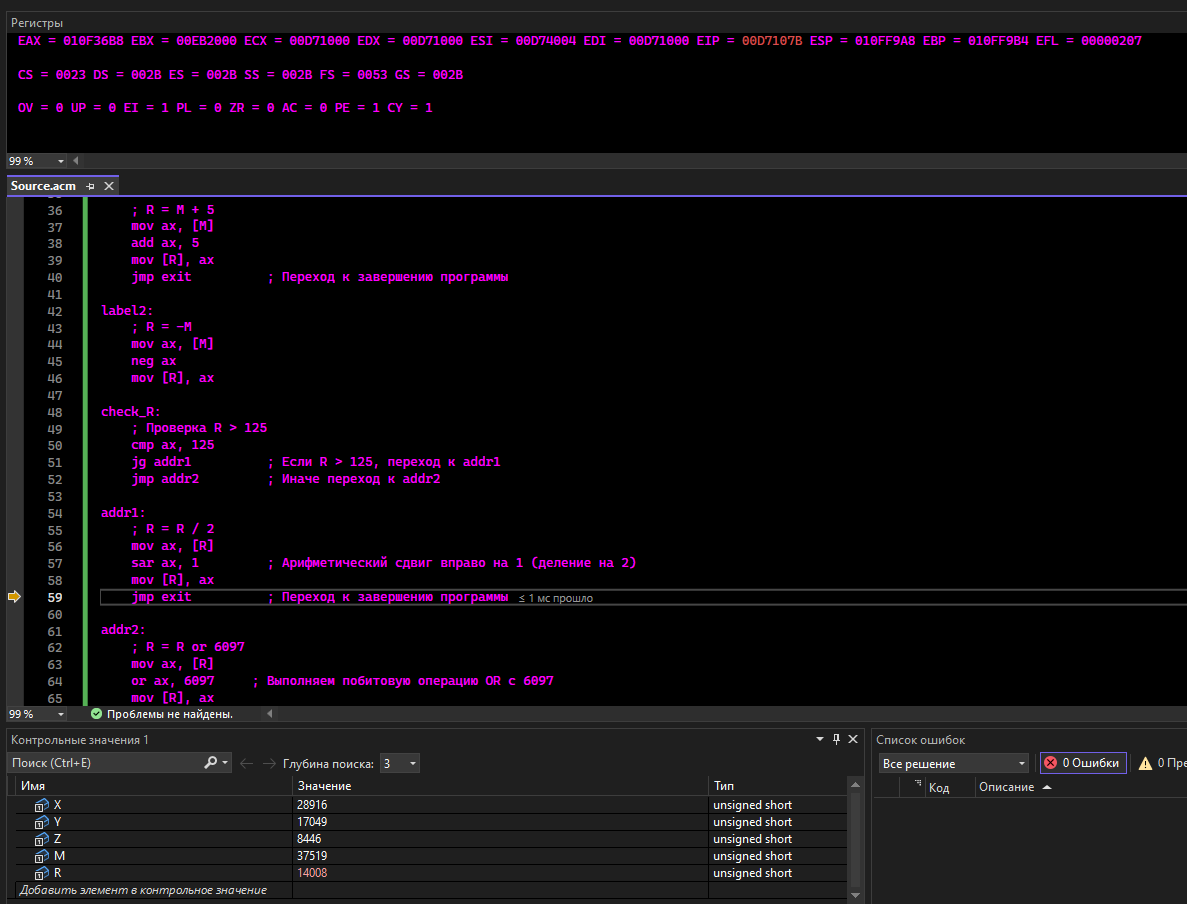












Таким образом, значение M должно быть равно 565.

Вывод: В ходе лабораторной работы были изучены основные принципы работы с ассемблером, включая выполнение арифметических операций, использование циклического сдвига и логических операций, а также взаимодействие с Windows API для вывода информации. Работа с ассемблером позволила глубже понять, как низкоуровневое программирование взаимодействует с аппаратным обеспечением и как реализуются базовые операции на уровне машинного кода.