МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

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 микропроцессоры»

**Лабораторная работа №2**

**«ПРИНЦИПЫ ВЫПОЛНЕНИЯ КОМАНД ВЕТВЛЕНИЯ, ОРГАНИЗАЦИЯ ЦИКЛОВ И ПОДПРОГРАММ»**

Вариант №1

|  |  |  |
| --- | --- | --- |
| Выполнил: студент группы ИНБс–3301 |  | В.В. Эберлинг |
|  |  |  |
| Проверил: преподаватель кафедры РЭС |  | М.А. Земцов |

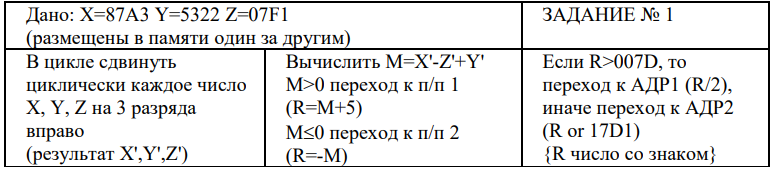
#### Киров 2025

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

**Ход работы:**

1. Исходные данные представлены в таблице 1.

Таблица 1 – Исходные данные



1. Текст программы

.686

.model flat,stdcall

.stack 100h

.data

X dw 87A3h

Y dw 5322h

Z dw 07F1h

R dd 0

.code

ExitProcess PROTO STDCALL :DWORD

Start:

xor eax, eax

xor ebx, ebx

mov ecx, 3

mov esi, offset X

shift\_loop:

mov ax, [esi]

ror ax, 3

push ax

add esi, 2

loop shift\_loop

mov ax, [esp+4]

add ax, [esp+2]

mov bx, [esp]

cmp eax, ebx

jg more

jle less

more: ; M>0

sub ax, [esp]

add ax, 5

jmp check

less: ; M<=0

mov bx, ax

mov ax, [esp]

sub ax, bx

check:

cmp eax, 007Dh

jg addr1

call pr2

jmp result

addr1:

call pr1

jmp result

pr1 proc

shr ax, 1

ret

pr1 endp

pr2 proc

or eax, 17D1h

ret

pr2 endp

result:

mov R, eax

exit:

Invoke ExitProcess,1

End Start

1. Верификация программы

|  |  |  |
| --- | --- | --- |
| X | 87A3 | 1000011110100011 |
| Y | 5322 | 0101001100100010 |
| Z | 7f1 | 0000011111110001 |

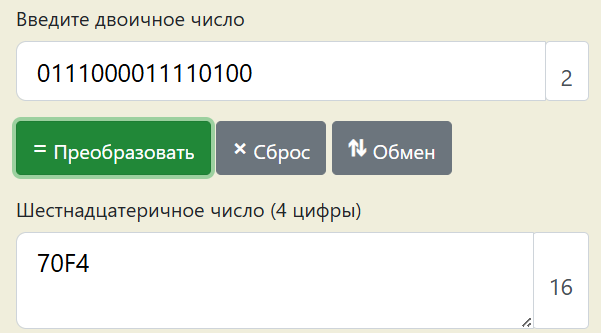


Рисунок 1.1 – Циклический сдвиг X вправо на 3 разряда

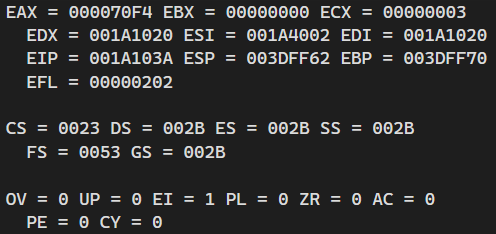


Рисунок 1.2 – Циклический сдвиг X вправо на 3 разряда в программе

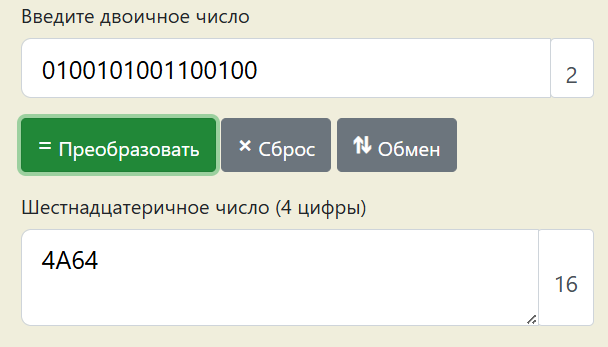


Рисунок 2.1 – Циклический сдвиг Y вправо на 3 разряда

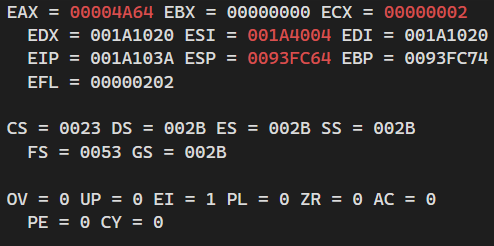


Рисунок 2.2 – Циклический сдвиг Y вправо на 3 разряда в программе

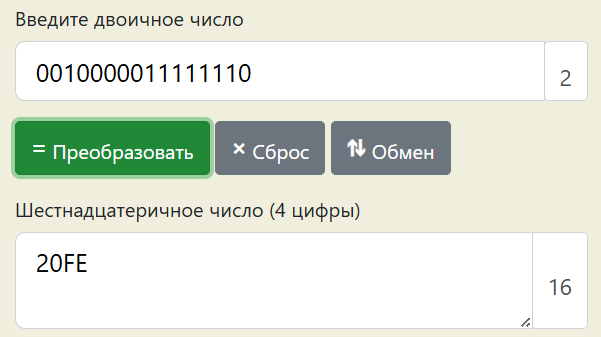


Рисунок 3.1 – Циклический сдвиг Z вправо на 3 разряда

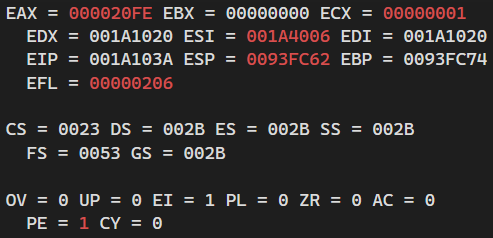


Рисунок 3.2 – Циклический сдвиг Z вправо на 3 разряда в программе

|  |  |  |
| --- | --- | --- |
| X’ | 70F4 | 0111000011110100 |
| Y’ | 4A64 | 0100101001100100 |
| Z’ | 20FE | 0010000011111110 |

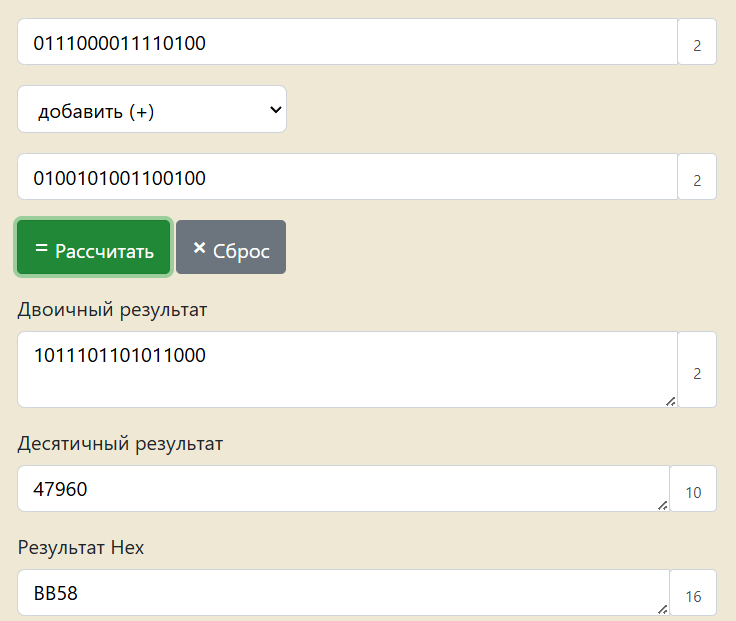


Рисунок 4.1 – Выполнение действия «X’ + Y’»

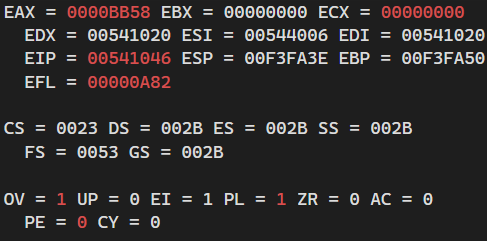


Рисунок 4.2 – Выполнение действия «X’ + Y’» в программе

Так как X’ + Y’ > Z переходим к п/п 1

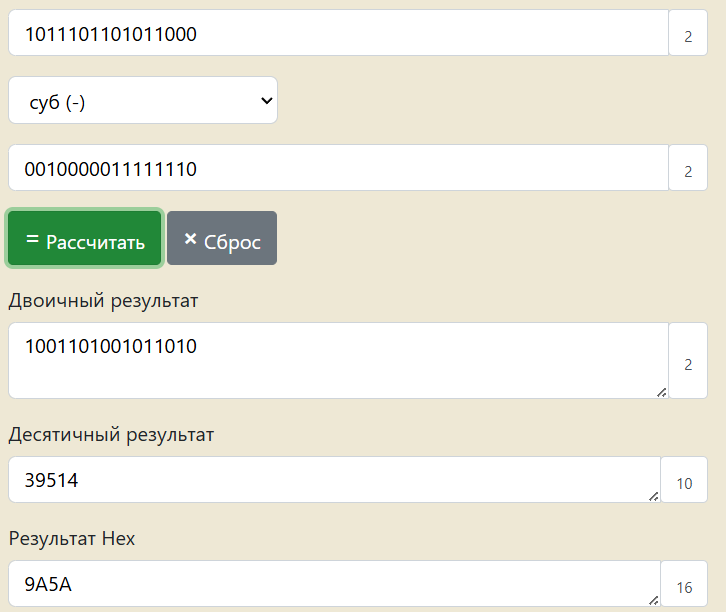


Рисунок 5.1 – Выполнение действия «X’ + Y’ – Z’»

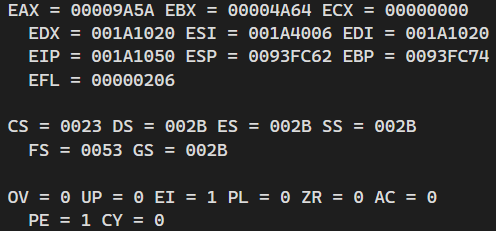


Рисунок 5.2 – Выполнение действия «X’ + Y’ – Z’» в программе

|  |  |  |
| --- | --- | --- |
| М | 9A5A | 1001101001011010 |

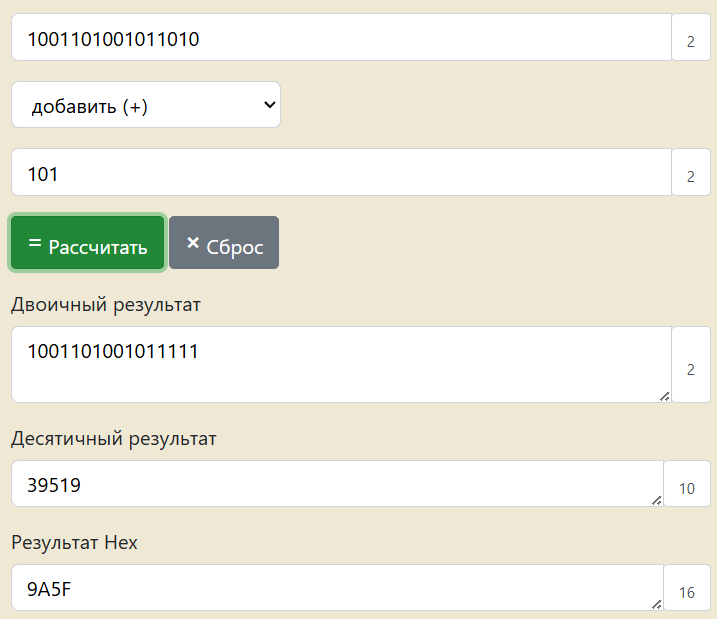


Рисунок 6.1 – Выполнение действия «М + 5»

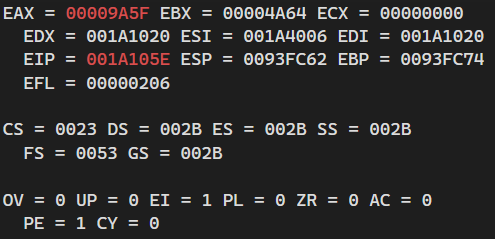


Рисунок 6.2 – Выполнение действия «М + 5» в программе

|  |  |  |
| --- | --- | --- |
| R | 9A5F | 1001101001011111 |

Так как R > 007D переходим к АДР1

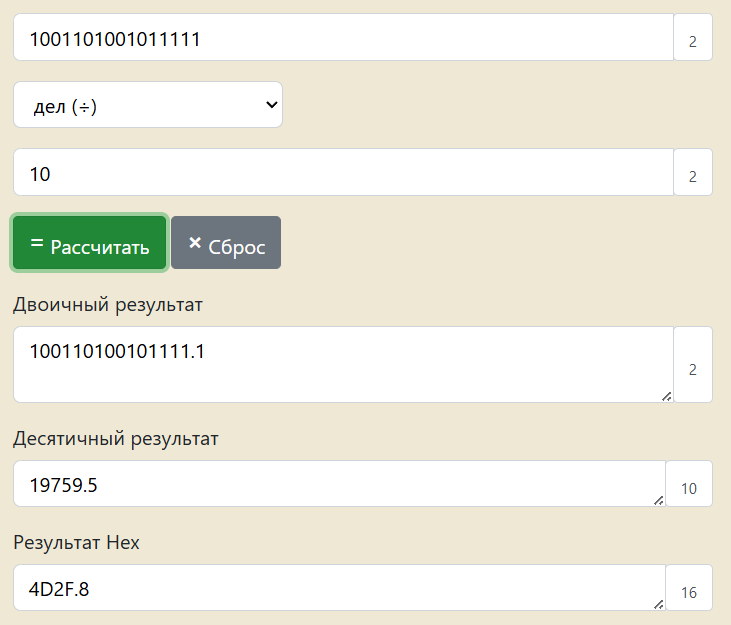


Рисунок 7.1 – Выполнение действия «R / 2»

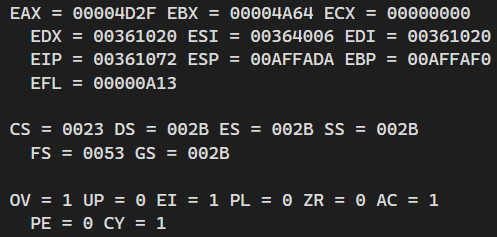


Рисунок 7.2 – Выполнение действия «R / 2» в программе

|  |  |  |
| --- | --- | --- |
| R | 4D2F | 100110100101111 |

1. Проверка ветвления программы

Чтобы проверить правильность работы программы, поменяем Z так, чтобы программа пошла по другим веткам.

В десятичной системе счисления X’ + Y’ = 47960, следовательно, чтобы программа пошла после первого ветвления по другой ветке Z’ должно быть больше или равно 47960, а по второй, чтобы меньше или равно 48085 (48960 + 125).

BB58 <= Z’ <= BBD5

Возьмём минимально возможное для проверки ветвления значение Z’, тогда Z будет равно 56005.

|  |  |  |
| --- | --- | --- |
| Z’ | BB58 | 1011101101011000 |
|  |  |  |
| Z | DAC5 | 1101101011000101 |

Сейчас так как X’ + Y’ < Z переходим к п/п 2.

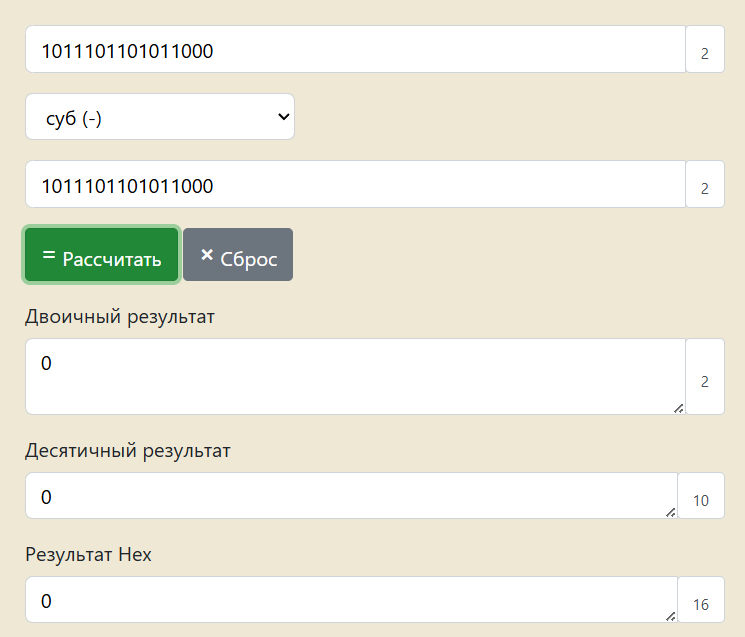


Рисунок 8.1 – Выполнение действия «– (X’ + Y’ – Z’)»

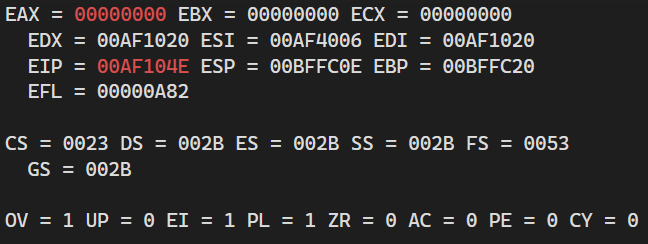


Рисунок 8.2 – Выполнение действия «– (X’ + Y’ – Z’)» в программе

|  |  |  |
| --- | --- | --- |
| М | 0 | 0000000000000000 |

Так как R < 007D переходим к АДР2

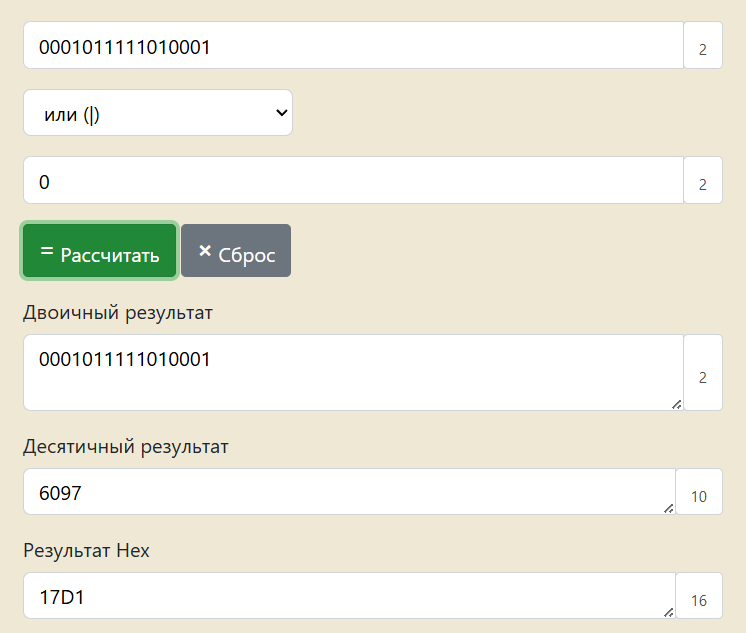


Рисунок 9.1 – Выполнение действия «R or 17D1»



Рисунок 9.2 – Выполнение действия «R or 17D1» в программе

**Вывод:** написана программа на ассемблере, реализующая вычисления в соответствие с вариантом и использующая подпрограммы, также были подобраны входные данные таким образом, чтобы проверить правильность выполнения программы при всех вариантах ветвлений и переходов.