Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

Кафедра ИС

ОТЧЕТ

по лабораторной работе №5

### «Исследование методов адресации и программирования арифметических и логических операций»

Выполнил:

ст. гр. ИС/б-21-о

Куркчи А. Э.

Проверил:

Шишкевич Е. В.

Севастополь

2016

1. Цель работы

Изучить основные директивы языка ассемблера, исследовать их воздействие на процесс ассемблирования и формирования листинга программы.

Исследовать особенности функционирования блоков 16-разрядного микропроцессора при выполнении арифметических и логических операций и при использовании различных способов адресации. Приобрести практические навыки программирования на языке ассемблера МП 8086 арифметических и логических операций с применением различных способов адресации.

2. Постановка задачи

3.1. Изучить основные директивы ассемблера и их воздействие на процесс ассемблирования и формирования листинга программы. Повторить команды пересылки данных, а также команды арифметических и логических операций (выполняется в процессе домашней подготовки к лабораторной работе).

3.2. Изучить методы адресации, используемые в 16-разрядных процессорах (выполняется во время домашней подготовке к работе).

3.3. Составит программу, состоящую из следующих процедур обработки строк:

3.3.1. Заполнить 100+10*i* ячеек области памяти, начинающейся с адреса MAS рядом натуральных чисел. Здесь i – последняя цифра номера Вашей зачетной книжки.

3.3.2. Переслать массив слов из области памяти, начиная с адреса MAS1 в область с начальным адресом MAS2.

3.3.3. Найти в заданном массиве число, равное двум последним цифрам Вашей зачетной книжки и определить его индекс.

3.4. Переслать в память с адресом 2020:300 диагональные элементы матрицы размером 8×8. Значения элементов матрицы должны быть определены в сегменте данных программы.

3.5. Произвести отладку разработанных программ в пошаговом режиме и проследить за изменениями содержимого регистров

3.6. Произвести ассемблирование программы и получить объектный и исполняемый модуль программы в Ехе-формате и ее листинг.

3.7. Рассчитать время выполнения программы.

3. Текст программы

data segment

table db 0**,**1**,**2**,**3**,**4**,**5**,**6**,**7

db 8**,**9**,**0**,**1**,**2**,**3**,**4**,**5

db 6**,**7**,**8**,**9**,**0**,**1**,**2**,**3

db 4**,**5**,**6**,**7**,**8**,**9**,**0**,**1

db 2**,**3**,**4**,**5**,**6**,**7**,**8**,**9

db 0**,**1**,**2**,**3**,**4**,**5**,**6**,**7

db 8**,**9**,**0**,**1**,**2**,**3**,**4**,**5

db 6**,**7**,**8**,**9**,**0**,**1**,**2**,**3

ends

stack segment

dw 128 dup**(**0**)**

ends

code segment

start**:**

**mov** **ax,** stack

**mov** **ss,** **ax**

**call** matrix

**call** gen

**call** move

**call** find

**jmp** finish

set\_data**:**

**mov** **ax,** data

**mov** **ds,** **ax**

**ret**

matrix**:**

**call** set\_data;

**mov** **si,** 0

**mov** **di,** 00300h

**mov** **ch,** 8

**call** matrix\_cycle

**ret**

matrix\_cycle**:**

**call** set\_data

**mov** **cl,** **[si]**

**add** **si,** 9

**mov** **ax,** 02020h

**mov** **ds,** **ax**

**mov** **[di],** **cl**

**inc** **di**

**dec** **ch**

**jnz** matrix\_cycle

**ret**

gen**:**

**call** set\_data

**mov** **di,** 40h

**mov** **[di],** 1

**inc** **di**

**mov** **ch,** 130

**mov** **cl,** 107

**mov** **dh,** 13

**call** gen\_cycle

**ret**

gen\_cycle**:**

**mov** **bh,** **[di** **-** 1**]**

**mov** **ax,** 2

**mul** **bh**

**add** **al,** **dh**

**div** **cl**

**mov** **[di],** **ah**

**inc** **di**

**dec** **ch**

**jnz** gen\_cycle

**ret**

move**:**

**mov** **ch,** 130

**mov** **di,** 0

**mov** **si,** 40h

**call** move\_cycle

**ret**

move\_cycle**:**

**call** set\_data

**mov** **bh,** **[si]**

**inc** **si**

**mov** **ax,** 00800h

**mov** **ds,** **ax**

**mov** **[di],** **bh**

**inc** **di**

**dec** **ch**

**jnz** move\_cycle

**ret**

find**:**

**call** set\_data

**mov** **ch,** 130

**mov** **si,** 40h

**call** find\_cycle

**ret**

find\_cycle**:**

**mov** **ah,** **[si]**

**inc** **si**

**mov** **al,** 63

**cmp** **ah,** **al**

**jz** success

**dec** **ch**

**jnz** find\_cycle

**mov** **ax,** 4c01h

**ret**

success**:**

**mov** **ax,** 4c00h

**mov** **dx,** **si**

**sub** **dx,** 40h

**ret**

finish**:**

**int** 21h

ends

end start

4. Выполнение программы

На рисунке 4.1 представлен скриншот выполнения генерации целочисленных значений по адресу 0710:0040.

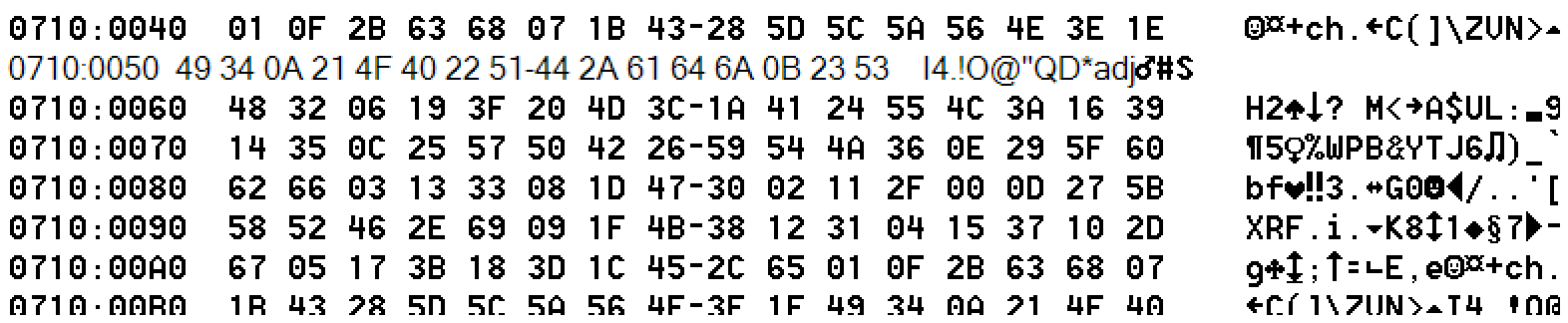


Рисунок 4.1 – Генерация чисел

На рисунке 4.2 представлен скриншот выполнения пересылки данных из первого адреса во второй – 0800:0000.

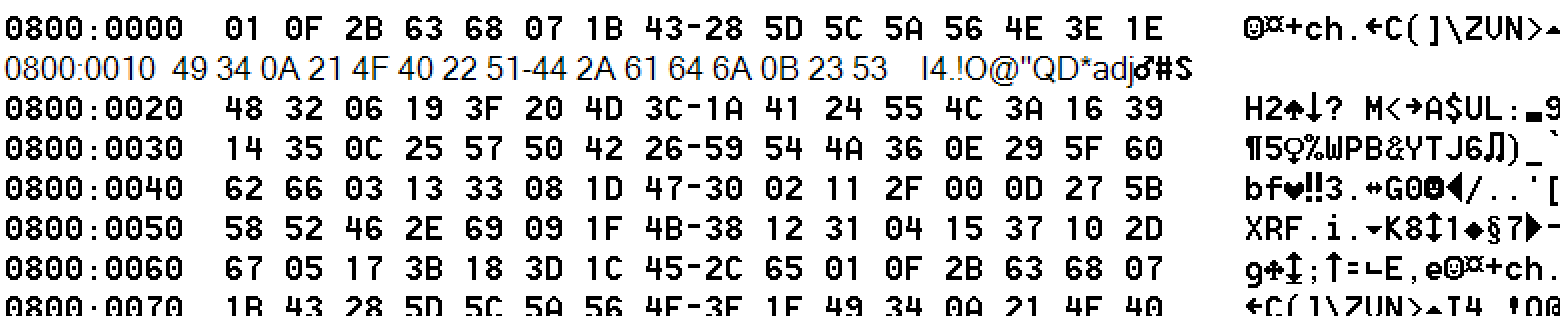


Рисунок 4.2 – Пересылка данных

На рисунке 4.3 представлен скриншот выполнения поиска в сгенерированных числах, последних двух цифр зачетной книжки – 63. Индекс смещения в массиве представлен в регистре DX, а код выхода программы в AX.

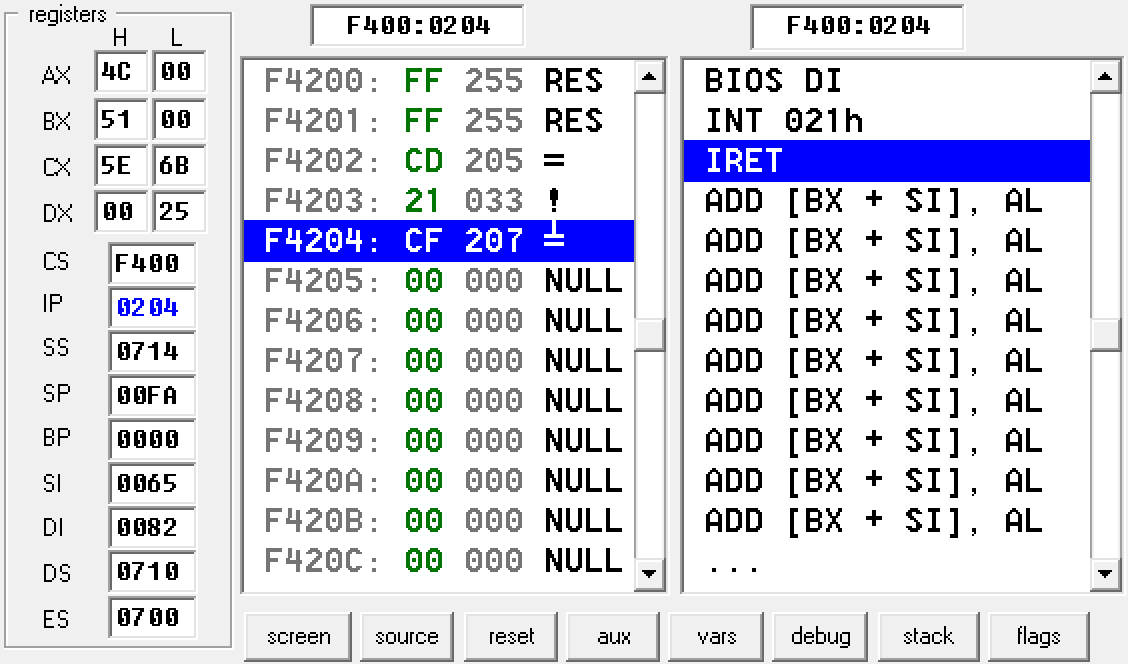


Рисунок 4.3 – Поиск цифр

На рисунке 4.4 представлен скриншот выполнения вывода диагональных элементов матрицы по адресу 2020:0300.

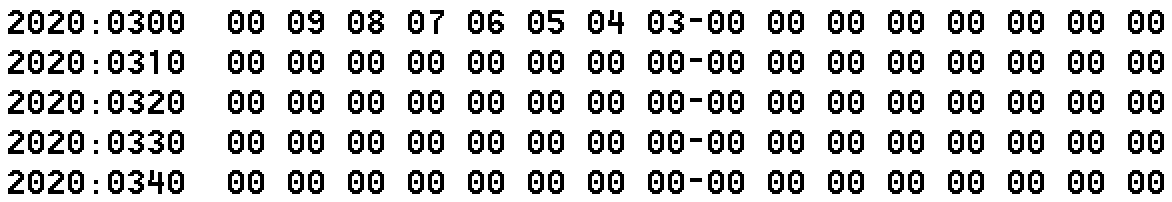


Рисунок 4.4 – Диагональные элементы матрицы

Вывод

В ходе лабораторной работы были изучены основные директивы языка ассемблера, исследованы их взаимодействия на процессы ассемблирования и формирования листинга программы. Исследованы особенности функционирования блоков 16-разряздного микропроцессора при выполнения арифметических и логических операций и при использовании различных способов адресации. Приобретены практические навыки программирования на языке ассемблера МП 8086 арифметических и логических операций с применением различных способов адресации.