**САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ**

**БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

МДК 01.03. Разработка мобильных приложений

Специальность 09.02.07

«Информационные системы и программирование»

Квалификация «Программист»

Руководитель учебной практики:

Смирнова И.П.

Выполнил студент группы 01:

Борзухин А. Н.

Санкт-Петербург, 2023

Оглавление

[Лабораторная работа № 1 3](#_Toc129601309)

[1.1 Цель лабораторной работы 3](#_Toc129601310)

[1.2 Содержание работы 3](#_Toc129601311)

[Лабораторная работа № 2 7](#_Toc129601312)

[Исследование команд прямой адресации 7](#_Toc129601313)

[2.1 Цель лабораторной работы 7](#_Toc129601314)

[2.2 Содержание работы 7](#_Toc129601315)

[Лабораторная работа № 3 9](#_Toc129601316)

[Исследование команд непосредственной адресации 9](#_Toc129601317)

[3.1 Цель лабораторной работы 9](#_Toc129601318)

[3.2 Содержание работы 9](#_Toc129601319)

[Лабораторная работа № 4 11](#_Toc129601320)

[Исследование команд косвенной адресации 11](#_Toc129601321)

[4.1 Цель лабораторной работы 11](#_Toc129601322)

[4.2 Содержание работы 11](#_Toc129601323)

[Лабораторная работа № 5 13](#_Toc129601324)

[Исследование команд стековой адресации 13](#_Toc129601325)

[5.1 Цель лабораторной работы 13](#_Toc129601326)

[5.2 Содержание работы 13](#_Toc129601327)

[Лабораторная работа № 6 15](#_Toc129601328)

[Пример программы для микропроцессора 15](#_Toc129601329)

[6.1 Цель лабораторной работы 15](#_Toc129601330)

[6.2 Содержание работы 15](#_Toc129601331)

[Лабораторная работа № 7 17](#_Toc129601332)

[Лабораторная работа № 8 19](#_Toc129601333)

[Лабораторная работа № 9 21](#_Toc129601334)

[Лабораторная работа № 10 25](#_Toc129601335)

[Лабораторная работа № 11 28](#_Toc129601336)

[Лабораторная работа № 12 30](#_Toc129601337)

[Лабораторная работа №13 34](#_Toc129601338)

[Лабораторная работа №14 42](#_Toc129601339)

[Лабораторная работа №15 46](#_Toc129601340)

[Лабораторная работа № 16 50](#_Toc129601341)

[Логические операции 50](#_Toc129601342)

[Команды обнуления битов 52](#_Toc129601343)

[Команды логического сложения 54](#_Toc129601344)

[Команды восстановления битов 56](#_Toc129601345)

[Команды инвертирования чисел 59](#_Toc129601346)

[Инвертирование флага переноса 61](#_Toc129601347)

[Инвертирование заданных битов числа 63](#_Toc129601348)

[Сравнение чисел 65](#_Toc129601349)

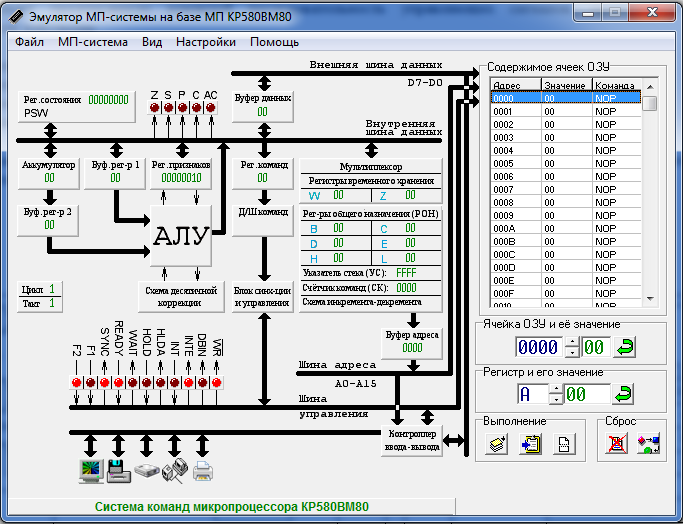
# Лабораторная работа № 1

## 1.1 Цель лабораторной работы

Познакомиться с МП KP580, познакомиться с программированием в машинных кодах и мнемокодах, научиться пользоваться средствами управления и клавиатурой устройств.

## 1.2 Содержание работы

Структура МП KP580



Буфер данных МП-системы и его значение

Регистр-аккумулятор (А) МП-системы

Регистр команд

Дешифратор команд

Буферный

Регистр1

Регистр признаков (флагов)

Буферный регистр МП-системы 2

Блок АЛУ

Блок десятичной коррекции значения регистра-аккумулятора

Блок синхронизации и управления

Таблица содержимого ОЗУ

панель редактирования значения ячейки ОЗУ

Панель редактирования значения содержимого выбранного регистра

Рисунок 1 – структура МП KP580

Таблица 1.2.1 − Программа прямой адресации

|  |  |  |
| --- | --- | --- |
| **Регистр** | **Назначение** | **Разрядность** |
| А | Регистр, в котором сохраняются непосредственные результаты выполнения арифметических и логических команд | 8-ми разрядный |
| HL | Адресный регистр, при косвенной адресации хранит 16-разрядный адрес основной памяти | 16-разрядный |
| PC | Функция счетчика команд | 16-разрядный |
| SP | Функция указателя стека | 16-разрядный |
| F и A | Образование стандартного регистрового набора микропроцессора с аккумулятором | 8-разрядные |
| B, C, D, E | Регистры общего назначения | 8-разрядные |
| F | Регистр флагов | 8-разрядый |

блок-схему функционирования МП во время выполнения команды сложения содержимого аккумулятора и регистра В, имеющую мнемоническое обозначение ADD B

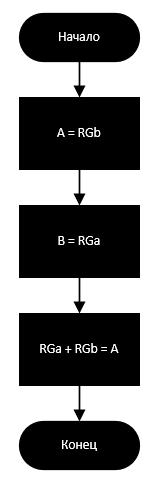


Рисунок 2 – Блок-схема выполнения команды сложения

Таблица 1.2.2 – флаги МП KP540

|  |  |  |
| --- | --- | --- |
| Обозначение флага | Признак флага | Условие установки флагов |
| C (Carry) | признак переноса | Флаг переноса C устанавливается при наличии переноса (при сложении) или заема (при вычитании) из старшего разряда аккумулятора, иначе сбрасывается |
| M (Minus) | признак отрицательного результата | Флаг знака M устанавливается, если знаковый бит результата операции (седьмой разряд аккумулятора) равен 1, иначе сбрасывается |
| Z (Zero) | признак нуля | Флаг нуля Z устанавливается, если результат операции в аккумуляторе равен нулю, иначе сбрасывается |
| P (Parity) | признак паритета/четности | Флаг четности P устанавливается, если результат операции в аккумуляторе содержит четное число единиц, иначе сбрасывается |
| AC (Auxiliary Carry) | признак половинного переноса | Флаг дополнительного переноса AC устанавливается при наличии переноса из третьего разряда аккумулятора в четвертый, иначе сбрасывается |

# Лабораторная работа № 2

# Исследование команд прямой адресации

## 2.1 Цель лабораторной работы

Рассмотреть особенности прямой адресации на МП, познакомиться с программированием в машинных кодах и мнемокодах, научиться пользоваться средствами управления и клавиатурой устройств.

## 2.2 Содержание работы

Прямая адресация — это такой метод адресации, когда каждая команда состоит из кода операции (первый байт) и адреса операнда, содержащегося во втором и третьем байтах (например, рассмотренная ниже команда LDA). Операнд находится в памяти, а в команде указывается его адрес. Прямая адресация − самая простая, но зато и самая неэкономичная с точки зрения объёма памяти и времени выполнения (13 машинных тактов). Примеры команд: JNP, CALL, LDA, STA.

Таблица 2.2.1 − Программа прямой адресации

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Команда | Комментарий |
| 0000 | 3A | LDA adr | Загрузить в аккумулятор значение по адресу 0A |
| 0001 | 0A | LDAX B |
| 0002 | 32 | STA adr | Сохранить значение из аккумулятора по адресу 0B |
| 0003 | 0B | DCX B |
| … | … | … |  |
| 000A | 10 | - |  |

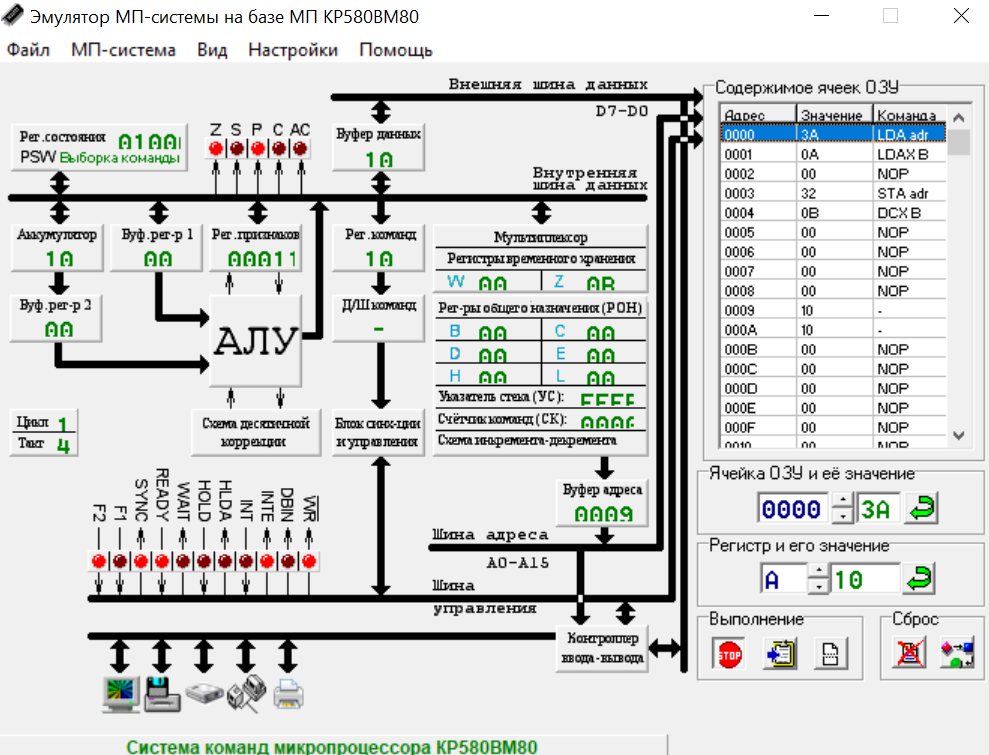


Рисунок 3 – Ввод операндов программы прямой адресации

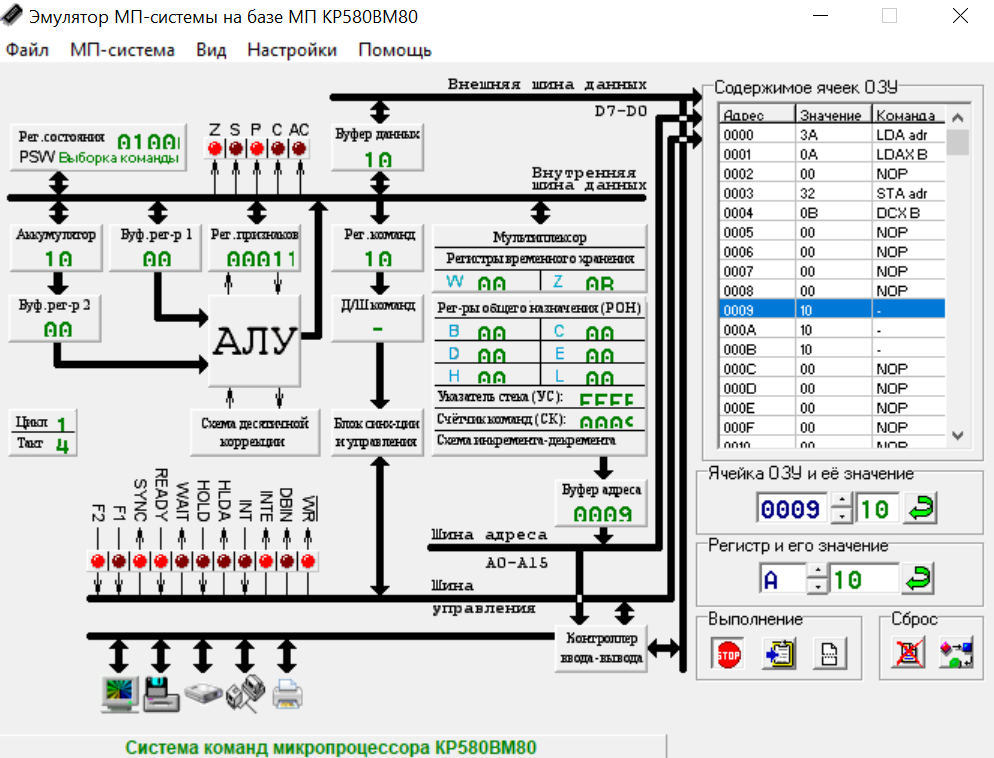


Рисунок 4 – Результат работы программы при прямой адресации

# Лабораторная работа № 3

# Исследование команд непосредственной адресации

## 3.1 Цель лабораторной работы

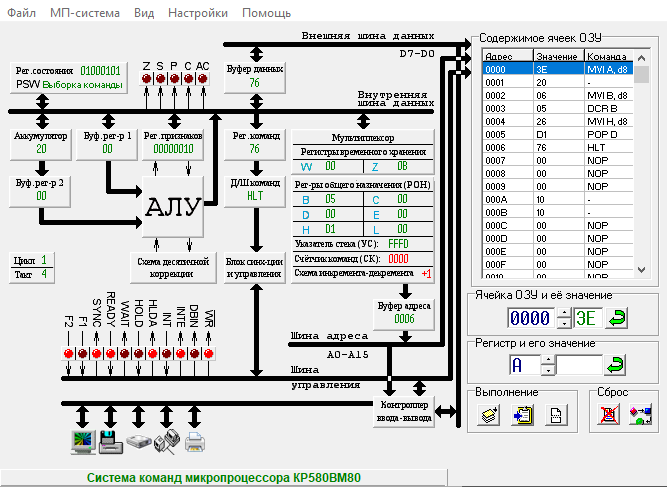
Рассмотреть особенности непосредственной адресации на МП, познакомиться с программированием в машинных кодах и мнемокодах, научиться пользоваться средствами управления и клавиатурой устройств.

## 3.2 Содержание работы

Непосредственная адресация – операнд непосредственно записан в коде команды (в мнемонике присутствует буква «I»). Перед операндом ставится запятая: MVI A, 20h. (возможно использование регистров A, B, C, D, E, H, L)

Таблица 3.2.1 − Программа косвенной адресации

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Команда | Комментарий |
| 0000 | 3E | MVI A, d8 | Положить в регистр A число 20h |
| 0001 | 20 | - |
| 0002 | 06 | MVI B, d8 | Положить в регистр B число 05h |
| 0003 | 05 | DCR B |
| 0004 | 26 | MVI H, d8 | Положить в регистр H число D1h |
| 0005 | D1 | POP D |
| 0006 | 76 | HLT | Завершить программу |

Рисунок 5 – Ввод операндов при непосредственной адресации

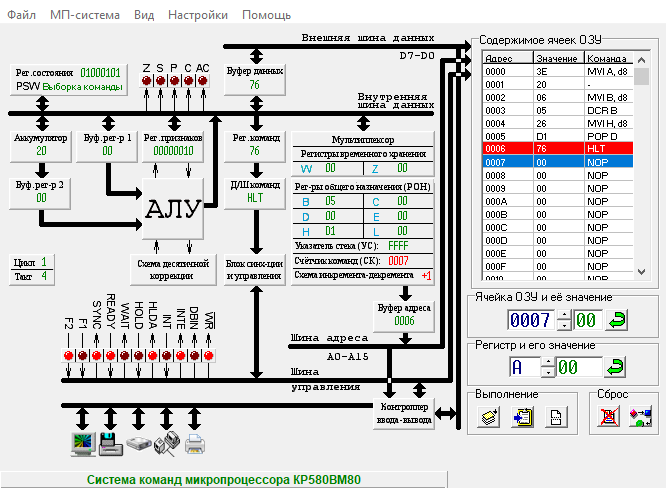


Рисунок 6 – Результат работы программы при непосредственной адресации

# Лабораторная работа № 4

# Исследование команд косвенной адресации

## 4.1 Цель лабораторной работы

Рассмотреть особенности косвенной адресации на МП, познакомиться с программированием в машинных кодах и мнемокодах, научиться пользоваться средствами управления и клавиатурой устройств.

## 4.2 Содержание работы

Косвенная адресация позволяет компактно адресоваться ко всем ячейкам памяти микропроцессора. В этом случае в поле операнда указывается код пары регистров, содержащих адрес памяти, по которому расположен байт данных. Адрес может содержаться в парах регистров HL (основной указатель адреса), а также в парах регистров ВС, DE (косвенная регистровая, при которой в формате команды

Таблица 4.2.1 − Программа косвенной адресации

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Команда | Комментарий |
| 0000 | 06 | MVI B, d8 | Положить в регистр B число 0 |
| 0001 | 00 | NOP |
| 0002 | 0E | MVI C, d8 | Положить в регистр C число 0A |
| 0003 | 0A | LDAX B |
| 0004 | 0A | LDAX B | Загрузка в аккумулятор значения по адресу из регистра BC |
| 0005 | 76 | HLT | Завершить программу |

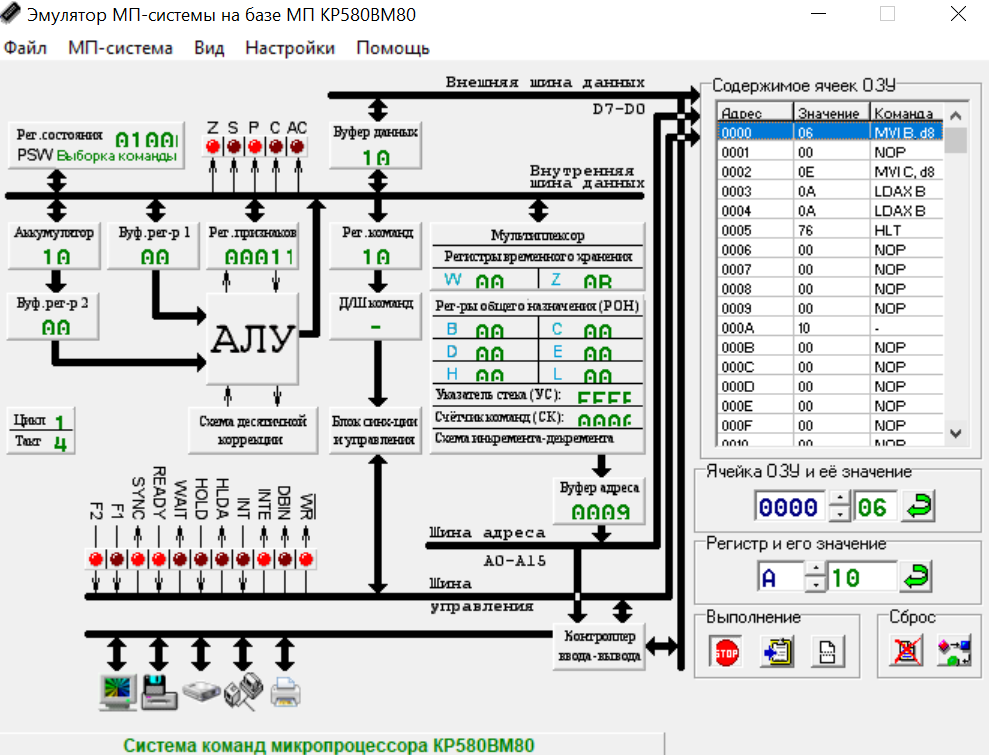


Рисунок 7 – Ввод операндов программы косвенной адресации

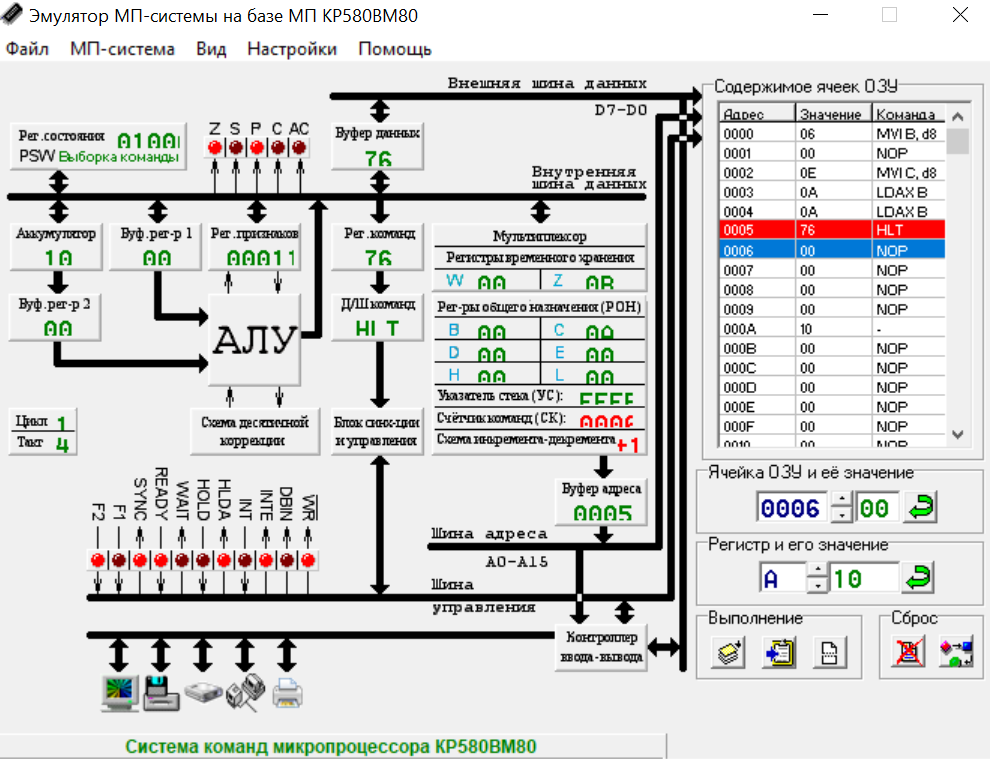


Рисунок 8 – Результат работы программы при косвенной адресации

# Лабораторная работа № 5

# Исследование команд стековой адресации

## 5.1 Цель лабораторной работы

Рассмотреть особенности выполнения простейших арифметических операций над целыми числами без знака на МП, познакомиться с программированием в машинных кодах и мнемокодах, научиться пользоваться средствами управления и клавиатурой устройств.

Выполнить операцию сложения двух однобайтных чисел X и Y.

## 5.2 Содержание работы

Источником или приёмником операнда является ячейка стековой памяти. Адрес вершины стека находится в указателе стека SP. При заполнении следующей ячейки указатель стека инкриминируется

Таблица 5.2.1 − Программа стековой адресации

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Команда | Комментарий |
| 0000 | 06 | MVI B, d8 | Положить в регистр B значение 10 |
| 0001 | 10 | - |
| 0002 | 16 | MVI D, d8 | Положить в регистр D значение 5A |
| 0003 | 5A | MOV E, D |
| 0004 | C5 | PUSH B | Положить в стек значение регистра B |
| 0005 | D5 | PUSH D | Положить в стек значение регистра D |
| 0006 | 50 | MOV D, B | Передать содержимое регистра B регистру D |
| 0007 | D1 | POP D | Достать из стека D |
| 0008 | C1 | POP B | Достать из стека B |
| 0009 | 76 | HLT | Завершить программу |

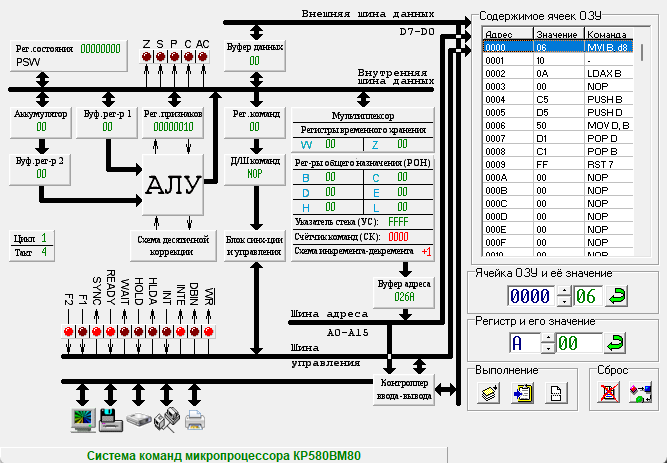


Рисунок 9 – Ввод операндов программы стековой адресации

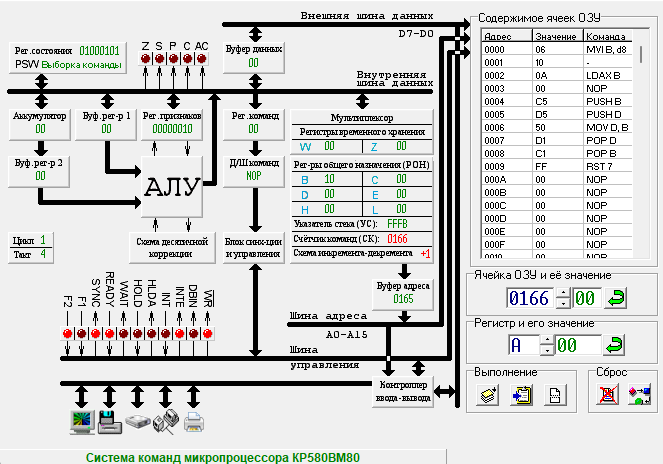


Рисунок 10 – Результат работы программы стековой адресации

# Лабораторная работа № 6

# Пример программы для микропроцессора

## 6.1 Цель лабораторной работы

Рассмотреть особенности выполнения простейших арифметических операций над целыми числами без знака на МП, познакомиться с программированием в машинных кодах и мнемокодах, научиться пользоваться средствами управления и клавиатурой устройств.

Выполнить операцию сложения двух однобайтных чисел X и Y.

## 6.2 Содержание работы

Таблица 6.2.1 − Программа сложения двух однобайтных чисел

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Мнемокод | Комментарий |
| 0000 | AF | ХRА А | Очистить аккумулятор |
| 0001 | ЗЕ | MVI A, 38 | Записать в аккумулятор  число 5610 = 38h16 |
| 0002 | 38 |
| 0003 | 06 | MVI B, A3 | Записать в регистр В число 16310 = a3h16 |
| 0004 | A3 |
| 0005 | 80 | ADD 8 | Сложить 38h16и a3h16 (5610 + 16310 = 21910 = db16) |
| 0006 | E7 | RST 7 | Прервать выполнение программы |

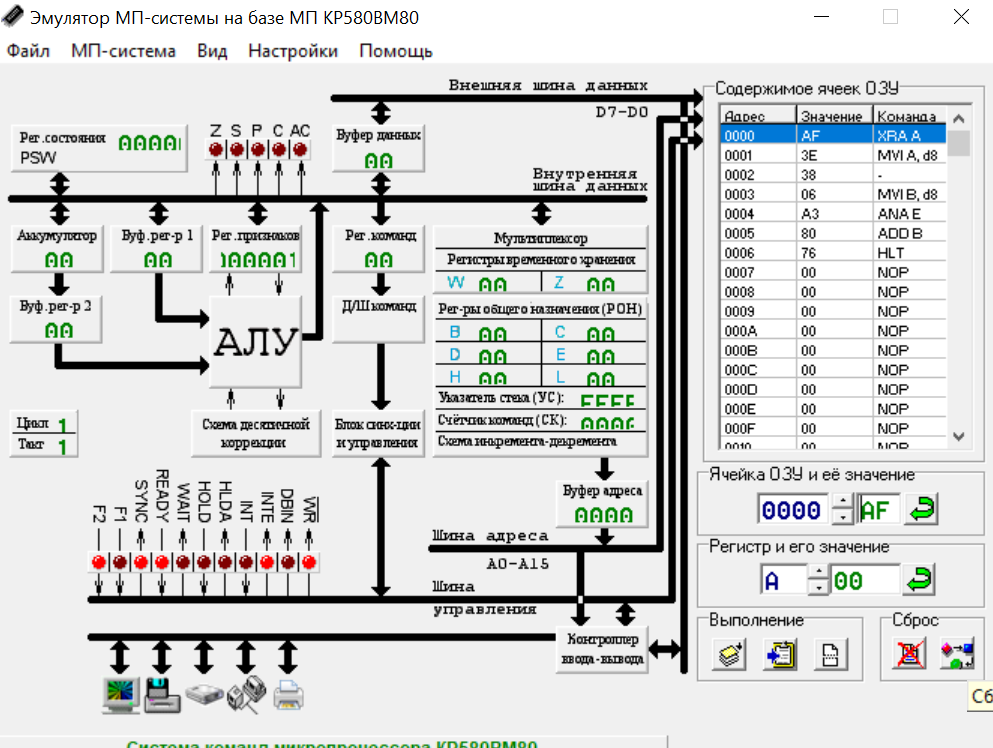


Рисунок 11 – Ввод операндов программы сложения двух чисел

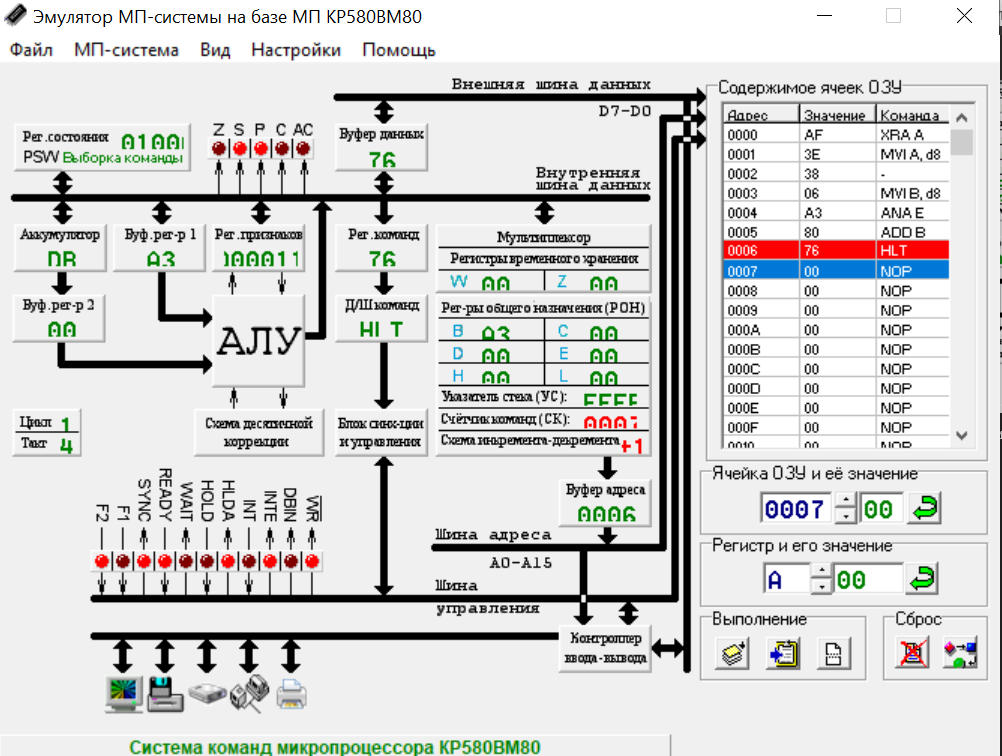


Рисунок 12 – Результат работы программы сложения двух чисел

# Лабораторная работа № 7

**Программа сложения двух однобайтных чисел X и У**

**Цель лабораторной работы:**  
  Задача состоит в том, чтобы, выполнив занесение однобайтных чисел в регистры А и В, сложить их и поместить результат сложения в аккумулятор.  
Сложение: 3710(2516) + 19910(С716) = 23610 (EC16)

Таблица 2 - Программа PRG 1 сложения двух однобайтных чисел

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Мнемокод | Комментарий |
| 0000 | AF | ХRА А | Очистить аккумулятор |
| 0001 | ЗЕ | MVI A, 38 | Записать в аккумулятор |
| 0002 | 25 |  | число X (3710 = 2516) |
| 0003 | 06 | MVI B, A3 | Записать в регистр В |
| 0004 | C7 |  | число У (19910 = C716) |
| 0005 | 80 | ADD B | Сложить X и Y. Сложить 2516 и C716 (3710 + 19910 = 23610 = EC16) |
| 0006 | 32 | STA adr | Записать содержимое аккумулятора в |
| 0007 | 0D |  | ячейку 000D |
| 0008 | 00 |  |  |
| 0009 | E7 | RST 7 | Прервать выполнение программы |

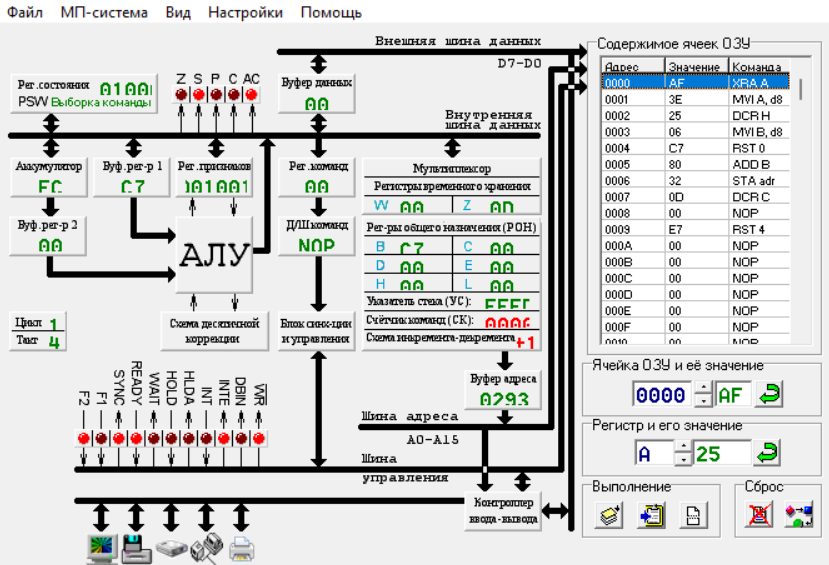


Рисунок 13 – До выполнения программы

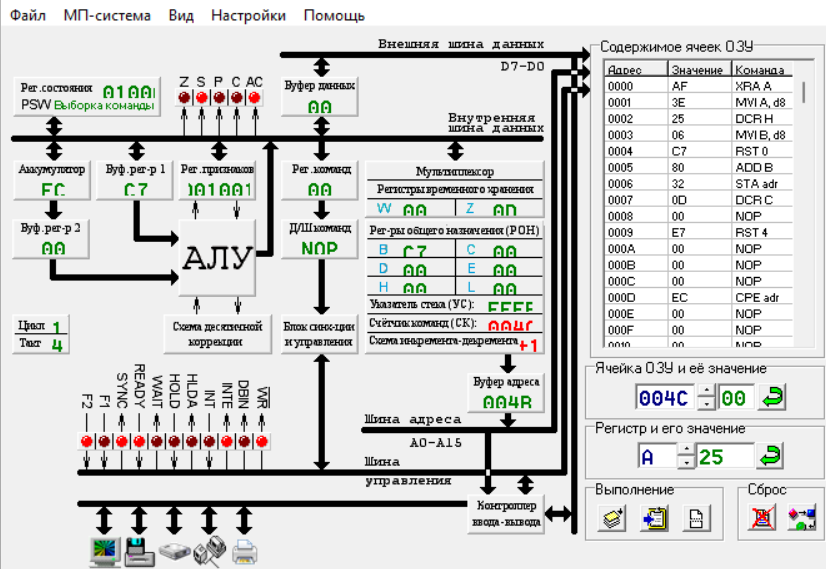


Рисунок 14 – После выполнения задания

# Лабораторная работа № 8

**Программа вычитания двух однобайтных чисел X и У**

**Цель лабораторной работы:**

Задача состоит в том, чтобы, выполнив занесение однобайтных чисел в регистры А и В, сложить их и поместить результат вычитания в аккумулятор.

Вычитание: 23510(EB16) - 7810(4E16) = 15710 (9D16)

Таблица 3 - Программа PRG 2 вычитания двух однобайтных чисел

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код команды | Мнемокод | Комментарий |
| 0000 | AF | ХRА А | Очистить аккумулятор |
| 0001 | ЗЕ | MVI A, 38 | Записать в аккумулятор |
| 0002 | EB |  | число X (23510 = EB16) |
| 0003 | 06 | MVI B, A3 | Записать в регистр В |
| 0004 | 4E |  | число У (7810 = 4E16) |
| 0005 | 90 | SUB B | Вычесть X и Y. Вычесть 4E16 из EB16 23510(EB16) - 7810(4E16) = 15710 (9D16) |
| 0006 | 32 | STA adr | Записать содержимое аккумулятора в |
| 0007 | 0D |  | ячейку 000D |
| 0008 | 00 |  |  |
| 0009 | E7 | RST 7 | Прервать выполнение программы |

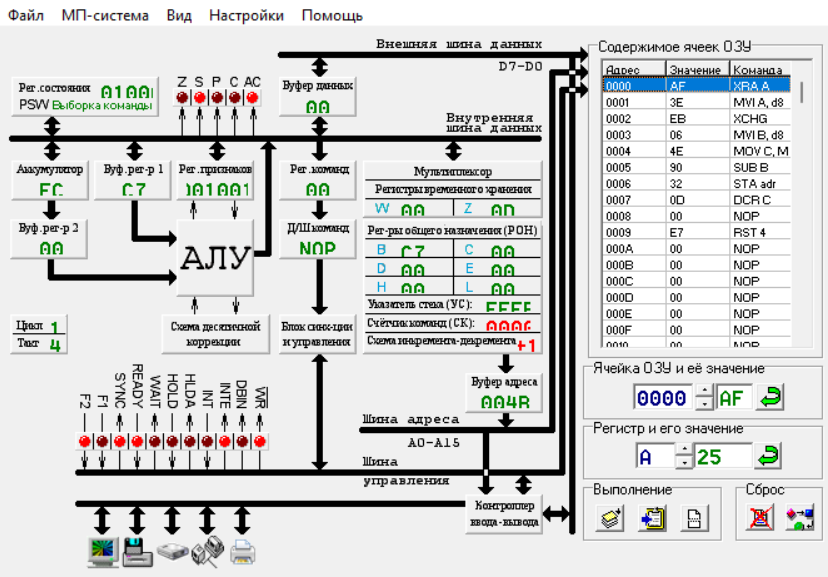
=

Рисунок 15 – До выполнения программы

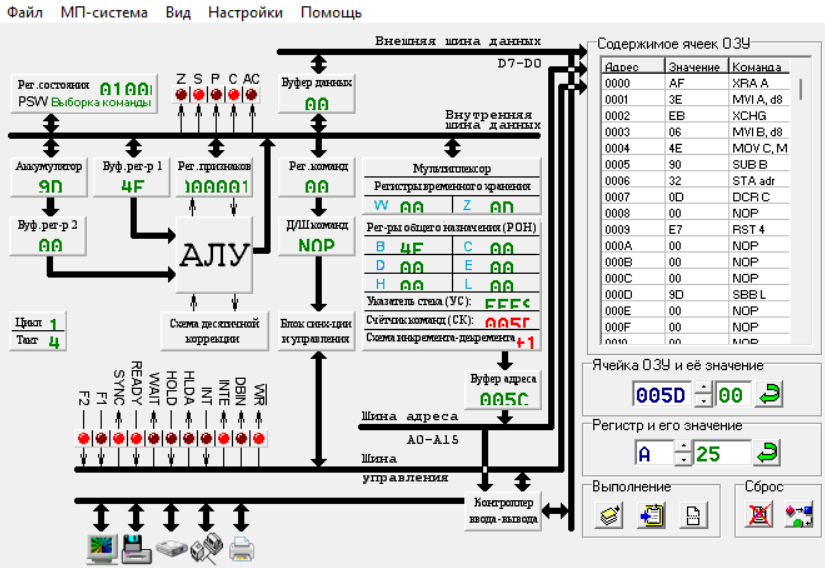


Рисунок 16 – После выполнения задания

# Лабораторная работа № 9

**Сложение массива однобайтных чисел**

**Цель лабораторной работы:**

Сложить массив: 13310(8516) + 5410(3616) + 4510(2D16) + 12510(7D16) + 3510(2316) + 1210(C16) + 17810(8216) = 53410(21616)

Вывод старшего байта выводится в ячейку 0020, младшего в 0021. Должно получится  
0020 – 02, 0021 – 16.

Таблица 1.3 – Программа PRG 3 сложения массива однобайтных чисел

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес | Код  команды | Метка | Мнемоника | Комментарий |
| 0000  0001  0002 | 21  50  00 | PRG 3: | LXI H,0050 | Загрузить в регистры HL, адрес первого слагаемого |
| 0003  0004 | 0E  07 |  | MVI C, 07 | Загрузить в регистр С количество слагаемых |
| 0005 | AF |  | XRA A | Очистить аккумулятор |
| 0006 | 47 |  | MOV B, A | Очистить регистр В |
| 0007 | 86 | M1: | ADD M | Прибавить к содержимому аккумулятора число из массива слагаемых |
| 0008  0009  000А | D2  0D  00 |  | INC M2 | Если переноса нет, то идти на М2 |
| 000В  000С | 04  B7 |  | INR B  ORA A | Увеличить содержимое регистра В на 1  Очистить флаг переноса |
| 000D | 23 | M2: | INX H | Указать на следующий адрес слагаемого |
| 000E | 0D |  | DCR C | Уменьшить содержимое регистра С на 1 |
| 000F  0010  0011 | C2  07  00 |  | INZ | Если не все слагаемые, то идти на M1 |
| 0012 | 32 |  | STA adr | Поместить данные из аккумулятора в ячейку  0021 |
| 0013 | 21 |  |  |
| 0014 | 00 |  |  |  |
| 0015 | AF |  | XRA A | Очистить аккумулятор |
| 0016 | 80 |  | ADD B | Добавить к аккумулятору значение из регистра B |
| 0017 | 32 |  | STA adr | Поместить данные из аккумулятора в ячейку  0020 |
| 0018 | 20 |  |  |
| 0019 | 00 |  |  |  |
| 001A | FF |  |  | Прервать выполнение программы |
| 0050 | 85 |  |  | Массив чисел |
| 0051 | 36 |  |  |
| 0052 | 2D |  |  |
| 0053 | 7D |  |  |
| 0054 | 23 |  |  |
| 0055 | C |  |  |
| 0056 | 82 |  |  |

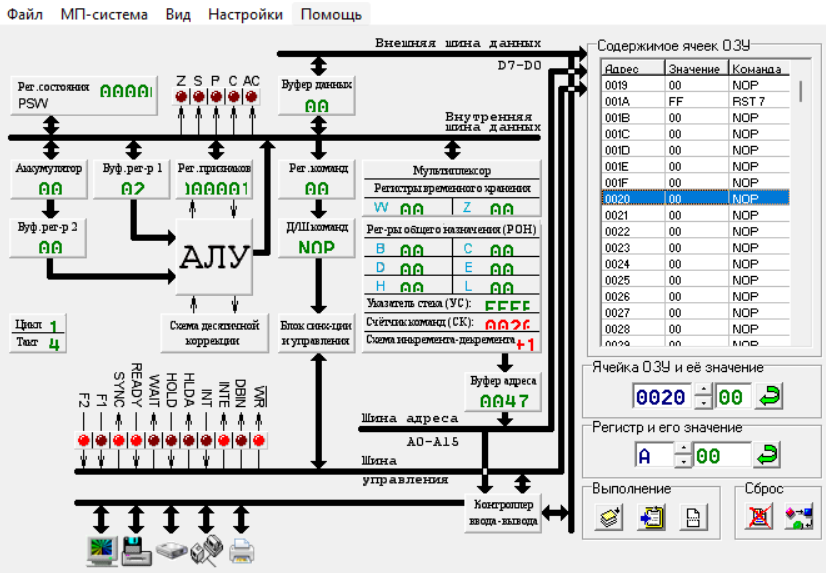


Рисунок 17 – До выполнения программы

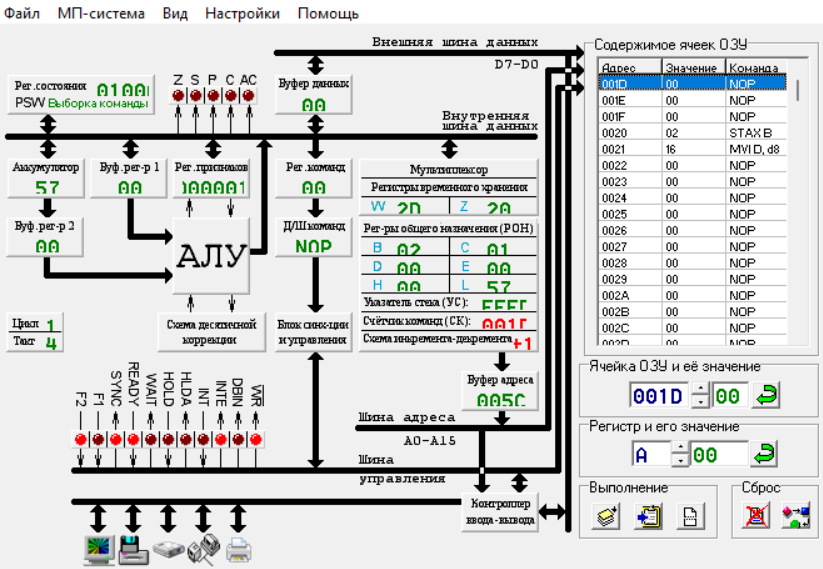


Рисунок 18 – После выполнения задания

# Лабораторная работа № 10

**Сложение двухбайтовых десятичных чисел**

**Цель лабораторной работы:**

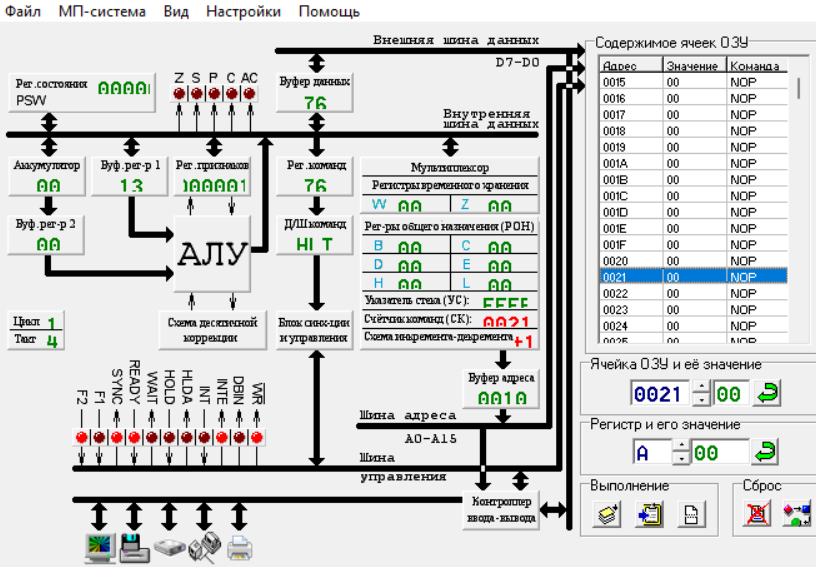
Выполнить операцию сложения двух двухбайтных чисел 204810 + 491510. Результат поместить в ячейки 0020 и 0021.

Таблица 1 - Программа PRG 1 сложения двух десятичных чисел

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 0000 | 3E | MVI A,8A | Младший байт первого слагаемого заносится в аккумулятор 0016 → A |
| 0001 | 00 |
| 0002 | 06 | MVI B,D8 | Младший байт первого слагаемого заносится в аккумулятор 3316 → B |
| 0003 | 33 | INX SP |  |
| 0004 | 80 | ADD B | Сложение A и B |
| 0005 | 32 | STA adr | Переместить байт из ячейки Bв ячейку Dс адресом 21 |
| 0006 | 50 | MOV D,B |  |
| 0007 | 00 | NOP |  |
| 0008 | 3E | MVI A,D8 | Старший байт первого слагаемого заносится в аккумулятор 0816 → A |
| 0009 | 08 | - |  |
| 000А | 06 | MVI B,D8 | Старший байт второго слагаемого заносится в аккумулятор 1316 → B |
| 000В | 13 | INX D |  |
| 000С | 88 | ADC B | Сложить |
| 000D | 32 | STA adr | Переместить байт из ячейки Bв ячейку Dс адресом 20 |
| 000E | 51 | MOV D,C |  |
| 000F | 00 | NOP |  |
| 0010 | 76 | HTL | остановка |

Данная программа выполняет операцию 80016 + 133316 = 1B3316

(204810 + 491510 = 696310).

Рисунок 19 – до выполнения программы

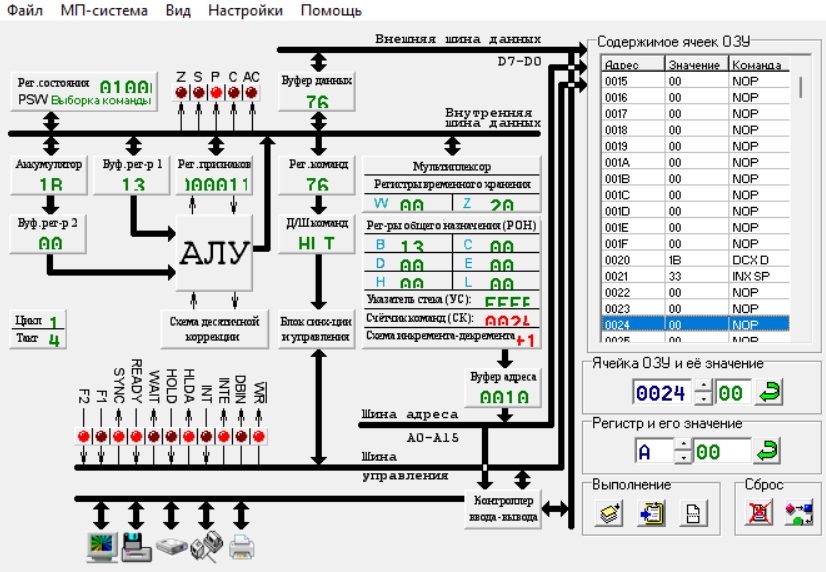


Рисунок 20 – после выполнения программы

# Лабораторная работа № 11

**Вычитание одинаковых по длине чисел**

**Цель лабораторной работы:**

Выполнить операцию вычитания двух двухбайтных чисел 6285110 – 2325110. Результат поместить в ячейки 0020 и 0021.

Таблица 2 - Программа PRG 2 вычитания двух чисел.

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 0000 | 3E | MVI A,d8 | Младший байт первого слагаемого заносится в аккумулятор 8316 →A |
| 0001 | 83 |
| 0002 | D6 | SUI d8 | Вычитание из A число D3 |
| 0003 | D3 |
| 0004 | 32 | STA adr | Переместить байт из ячейки Bв ячейку Dс адресом 21 |
| 0005 | 21 | MOV D,B |
| 0006 | 00 | NOP |  |
| 0007 | 3E | MVI A,21 | Старший байт первого слагаемого заносится в аккумулятор F516 → A |
| 0008 | F5 |
| 0009 | DE | SBI d8 | Вычитание из A число 5A |
| 000А | 5A |
| 000В | 32 | STA adr | Переместить байт из ячейки Bв ячейку Dс адресом 20 |
| 000С | 20 | MOV D,C |
| 000D | 00 | NOP |  |
| 000E | 76 | HLT | остановка |

Данная программа выполняет операцию F58316 – 5AD316 = 9AB016

(6285110 – 2325110 = 3960010)

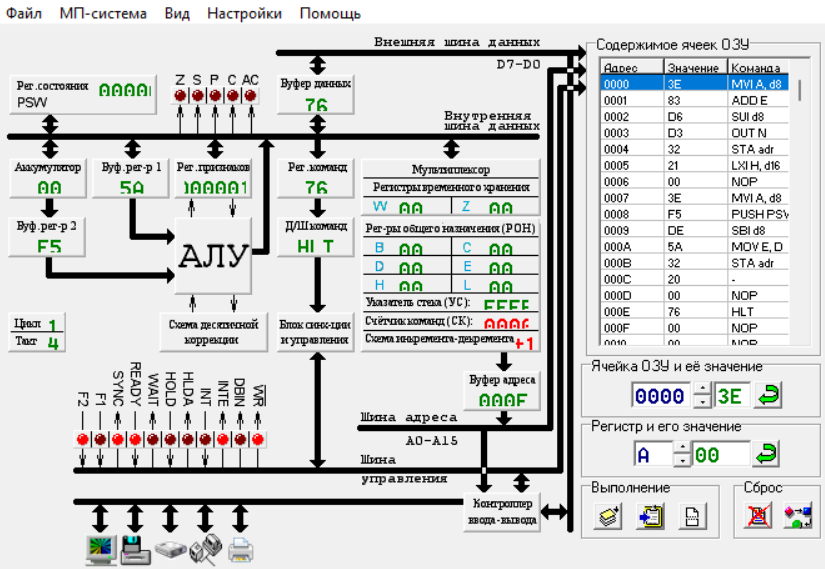


Рисунок 21 – до выполнения программы

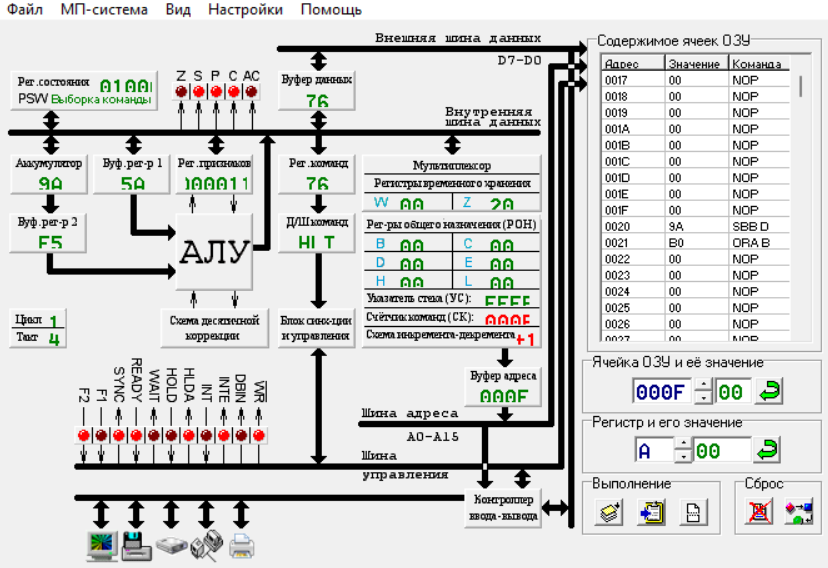


Рисунок 22 – после выполнения программы

# Лабораторная работа № 12

**Изучение алгоритма ветвления**

**Цель лабораторной работы:**

Рассмотреть особенности выполнения алгоритма ветвлений на МП, познакомиться с программированием в машинных кодах и мнемокодах, научиться пользоваться средствами управления и клавиатурой устройств.

Значение числа x кладется в ячейку с адресом 0030 и результат по адресу 0031 соответственно.

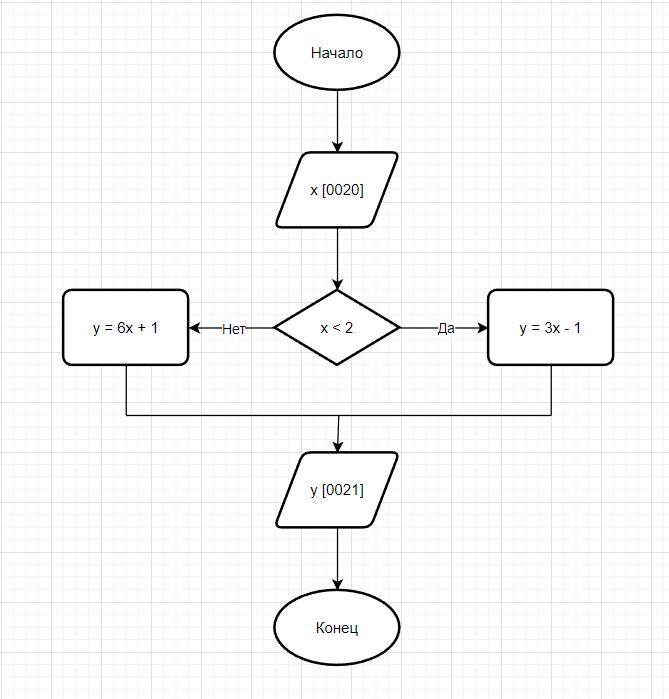


Рисунок 23 - Блок-схема алгоритма ветвления

Таблица - Программа PRG 1 с ветвлением.

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 0000 | 3A | LDA adr | Загрузить в аккумулятор число по адресу 0020 |
| 0001 | 20 |
| 0002 | 00 |
| 0003 | FE | CPI d8 | Сравнить число 2 со значением в аккумуляторе |
| 0004 | 02 |
| 0005 | FA | JM adСr | Перейти по метке 0012, если больше или равно 2 |
| 0006 | 12 |
| 0007 | 00 |
| 0008 | 47 | MOV B, A | Поместить содержимое аккумулятора в регистр B |
| 0009 | 80 | ADD B | Умножение через сложение |
| 000А | 80 |
| 000В | 06 | MVI B, d8 | Поместить число 1 в регистр B |
| 000С | 01 |
| 000D | 90 | SUB B | Вычесть число 1 |
| 000E | 32 | STA adr | Поместить значение аккумулятора в ячейку 0021 |
| 000F | 21 |
| 0010 | 00 |
| 0011 | 76 | HLT | Остановить программу |
| 0012 | 47 | MOV B, A | Поместить содержимое аккумулятора в регистр B |
| 0013 | 80 | ADD B | Умножение через сложение |
| 0014 | 80 |
| 0015 | 80 |
| 0016 | 80 |
| 0017 | 80 |
| 0018 | 06 | MVI B, d8 | Поместить 1 в регистр B |
| 0019 | 01 |
| 001A | 80 | ADD B | Прибавить число 1 |
| 001B | 32 | STA adr | Поместить значение аккумулятора в ячейку 0021 |
| 001C | 21 |
| 001D | 00 |
| 001E | 76 | HLT | Остановить программу |

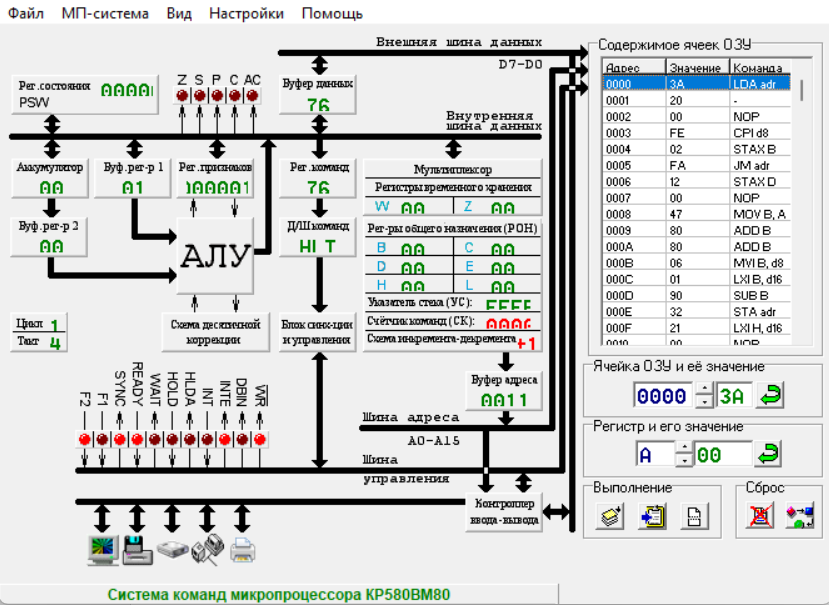


Рисунок 24 – До выполнения программы

Необходимо взять тестовые значения для проверки программы. При x = 1 уравнение будет следующим: 3 \* 1 – 1 = 2. При x = 2 уравнение будет следующим: 6 \* 2 + 1 = 13 (D16). Итого в результате выполнения программы мы должны получить следующие результаты 216 и D16.

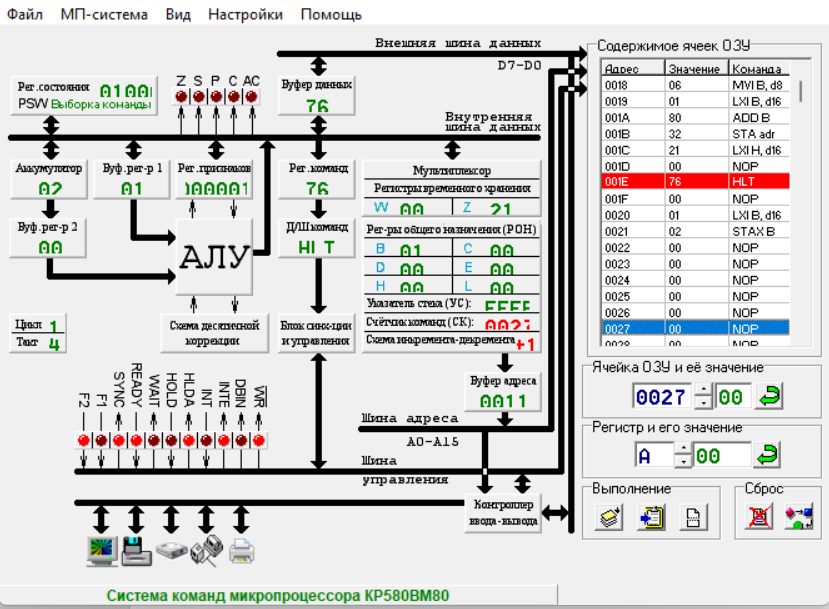


Рисунок 25 – результат выполнения команды для х = 1

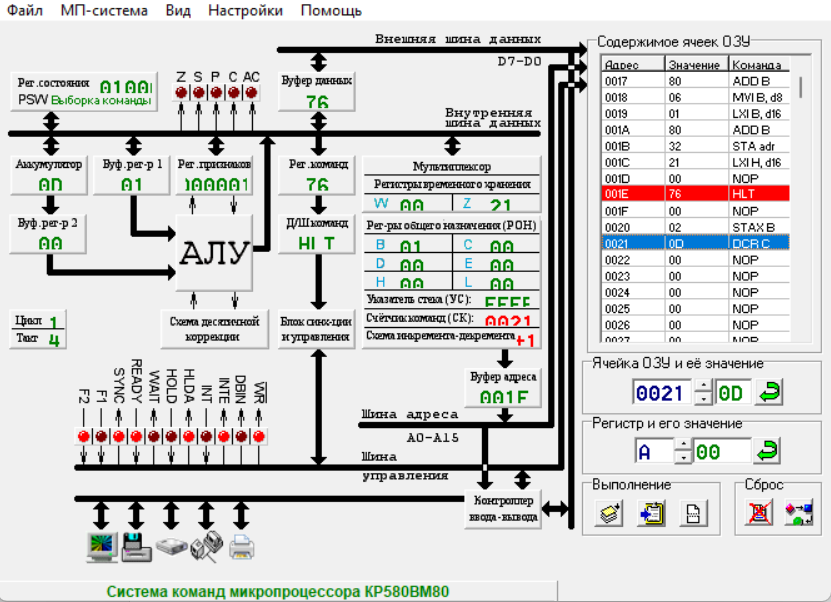


Рисунок 26 – Результат выполнения программы при х = 2

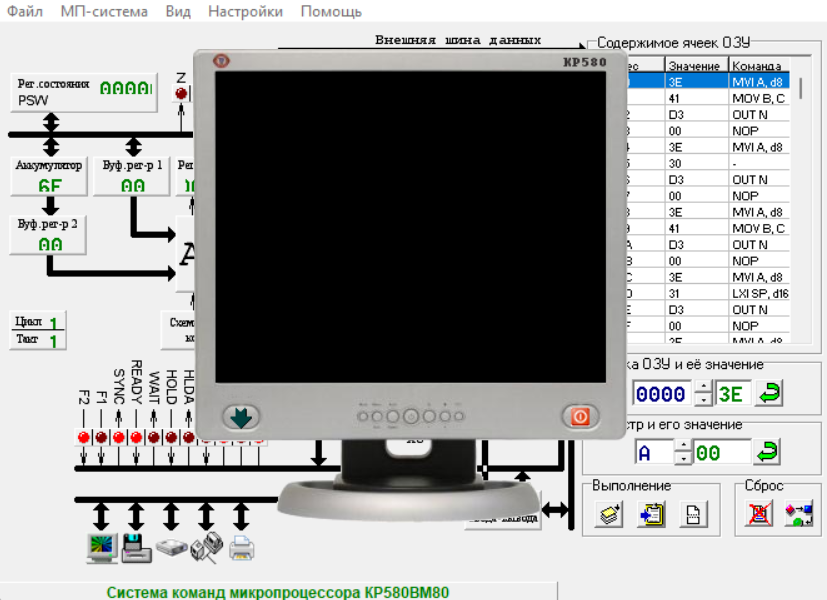
# Лабораторная работа №13

**Вывод текстовой информации на экран**

**Цель лабораторной работы:**  
 Рассмотреть особенности вывода информации на экран МП, познакомиться с программированием в машинных кодах и мнемокодах, научиться пользоваться средствами управления и клавиатурой устройств. Необходимо вывести на экран монитора KP580 номер группы, фамилию и имя разными цветами.

Таблица - Программа PRG 1 с выводом “01-AndrewBorzuhin” на экран.

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес | Код | Мнемоника | Комментарий |
| 0000 | 3E | MVI A, d8 | Загрузка кода синего цвета |
| 0001 | 41 | MOV B, C |  |
| 0002 | D3 | OUT N |  |
| 0003 | 00 | NOP |  |
| 0004 | 3E | MVI A, d8 | Загрузка кода символа 0 |
| 0005 | 30 | - |  |
| 0006 | D3 | OUT N |  |
| 0007 | 00 | NOP |  |
| 0008 | 3E | MVI A, d8 | Загрузка кода синего цвета |
| 0009 | 41 | MOV B, C |  |
| 000A | D3 | OUT N |  |
| 000B | 00 | NOP |  |
| 000C | 3E | MVI A, d8 | Загрузка кода символа 1 |
| 000D | 31 | LXI SP, d16 |  |
| 000E | D3 | OUT N |  |
| 000F | 00 | NOP |  |
| 0010 | 3E | MVI A, d8 | Загрузка кода зеленого цвета |
| 0011 | 21 | LXI H, d16 |  |
| 0012 | D3 | OUT N |  |
| 0013 | 00 | NOP |  |
| 0014 | 3E | MVI A, d8 | Загрузка кода символа - |
| 0015 | 2D | DCR L |  |
| 0016 | D3 | OUT N |  |
| 0017 | 00 | NOP |  |
| 0018 | 3E | MVI A, d8 | Загрузка салатового цвета |
| 0019 | 31 | LXI SP, d16 |  |
| 001A | D3 | OUT N |  |
| 001B | 00 | NOP |  |
| 001C | 3E | MVI A, d8 | Загрузка буквы A |
| 001D | 41 | MOV B, H |  |
| 001E | D3 | OUT N |  |
| 001F | 00 | NOP |  |
| 0020 | 3E | MVI A, d8 | Загрузка сиреневого цвета |
| 0021 | 51 | MOV D, C |  |
| 0022 | D3 | OUT N |  |
| 0023 | 00 | NOP |  |
| 0024 | 3E | MVI A, d8 | Загрузка буквы n |
| 0025 | 6E | MOV H, L |  |
| 0026 | D3 | OUT N |  |
| 0027 | 00 | NOP |  |
| 0028 | 3E | MVI A, d8 | Загрузка бирюзового цвета |
| 0029 | 71 | MOV M, C |  |
| 002A | D3 | OUT N |  |
| 002B | 00 | NOP |  |
| 002C | 3E | MVI A, d8 | Загрузка буквы d |
| 002D | 64 | MOV L, M |  |
| 002E | D3 | OUT N |  |
| 002F | 00 | NOP |  |
| 0030 | 3E | MVI A, d8 | Загрузка синего цвета |
| 0031 | 41 | MOV B, C |  |
| 0032 | D3 | OUT N |  |
| 0033 | 00 | NOP |  |
| 0034 | 3E | MVI A, d8 | Загрузка буквы r |
| 0035 | 72 | MOV L, C |  |
| 0036 | D3 | OUT N |  |
| 0037 | 00 | NOP |  |
| 0038 | 3E | MVI A, d8 | Загрузка коричневого цвета |
| 0039 | 11 | LXI D, d16 |  |
| 003A | D3 | OUT N |  |
| 003B | 00 | NOP |  |
| 003C | 3E | MVI A, d8 | Загрузка буквы e |
| 003D | 65 | MOV M, E |  |
| 003E | D3 | OUT N |  |
| 003F | 00 | NOP |  |
| 0040 | 3E | MVI A, d8 | Загрузка зеленого цвета |
| 0041 | 21 | LXI H, d16 |  |
| 0042 | D3 | OUT N |  |
| 0043 | 00 | NOP |  |
| 0044 | 3E | MVI A, d8 | Загрузка буквы w |
| 0045 | 77 | DCR L |  |
| 0046 | D3 | OUT N |  |
| 0047 | 00 | NOP |  |
| 0048 | 3E | MVI A, d8 | Загрузка сиреневого цвета |
| 0049 | 51 | MOV D, C |  |
| 004A | D3 | OUT N |  |
| 004B | 00 | NOP |  |
| 004C | 3E | MVI A, d8 | Загрузка буквы B |
| 004D | 42 | MOV B, L |  |
| 004E | D3 | OUT N |  |
| 004F | 00 | NOP |  |
| 0050 | 3E | MVI A, d8 | Загрузка бирюзового цвета |
| 0051 | 71 | MOV M, C |  |
| 0052 | D3 | OUT N |  |
| 0053 | 00 | NOP |  |
| 0054 | 3E | MVI A, d8 | Загрузка буквы o |
| 0055 | 6F | MOV M, D |  |
| 0056 | D3 | OUT N |  |
| 0057 | 00 | NOP |  |
| 0058 | 3E | MVI A, d8 | Загрузка коричневого цвета |
| 0059 | 11 | LXI D, d16 |  |
| 005A | D3 | OUT N |  |
| 005B | 00 | NOP |  |
| 005C | 3E | MVI A, d8 | Загрузка буквы r |
| 005D | 72 | MOV H, L |  |
| 005E | D3 | OUT N |  |
| 005F | 00 | NOP |  |
| 0060 | 3E | MVI A, d8 | Загрузка синего цвета |
| 0061 | 41 | MOV B, C |  |
| 0062 | D3 | OUT N |  |
| 0063 | 00 | NOP |  |
| 0064 | 3E | MVI A, d8 | Загрузка буквы z |
| 0065 | 7A | MOV L, L |  |
| 0066 | D3 | OUT N |  |
| 0067 | 00 | NOP |  |
| 0068 | 3E | MVI A, d8 | Загрузка голубого цвета |
| 0069 | 61 | MOV H, C |  |
| 006A | D3 | OUT N |  |
| 006B | 00 | NOP |  |
| 006C | 3E | MVI A, d8 | Загрузка буквы u |
| 006D | 75 | MOV H, L |  |
| 006E | D3 | OUT N |  |
| 006F | 00 | NOP |  |
| 0070 | 3E | MVI A, d8 | Загрузка зеленого цвета |
| 0071 | 21 | LXI H, d16 |  |
| 0072 | D3 | OUT N |  |
| 0073 | 00 | NOP |  |
| 0074 | 3E | MVI A, d8 | Загрузка буквы h |
| 0075 | 68 | MOV L, M |  |
| 0076 | D3 | OUT N |  |
| 0077 | 00 | NOP |  |
| 0078 | 3E | MVI A, d8 | Загрузка зеленого цвета |
| 0079 | 21 | LXI H, d16 |  |
| 007A | D3 | OUT N |  |
| 007B | 00 | NOP |  |
| 007C | 3E | MVI A, d8 | Загрузка буквы i |
| 007D | 69 | MOV L, E |  |
| 007E | D3 | OUT N |  |
| 007F | 00 | NOP |  |
| 0080 | 3E | MVI A, d8 | Загрузка зеленого цвета |
| 0081 | 21 | LXI H, d16 |  |
| 0082 | D3 | OUT N |  |
| 0083 | 00 | NOP |  |
| 0084 | 3E | MVI A, d8 | Загрузка буквы n |
| 0085 | 6E | MOV L, A |  |
| 0086 | D3 | OUT N |  |
| 0087 | 00 | NOP |  |
| 0088 | 76 | HLT | Остановка программы |

Рисунок 27 – До выполнения программы

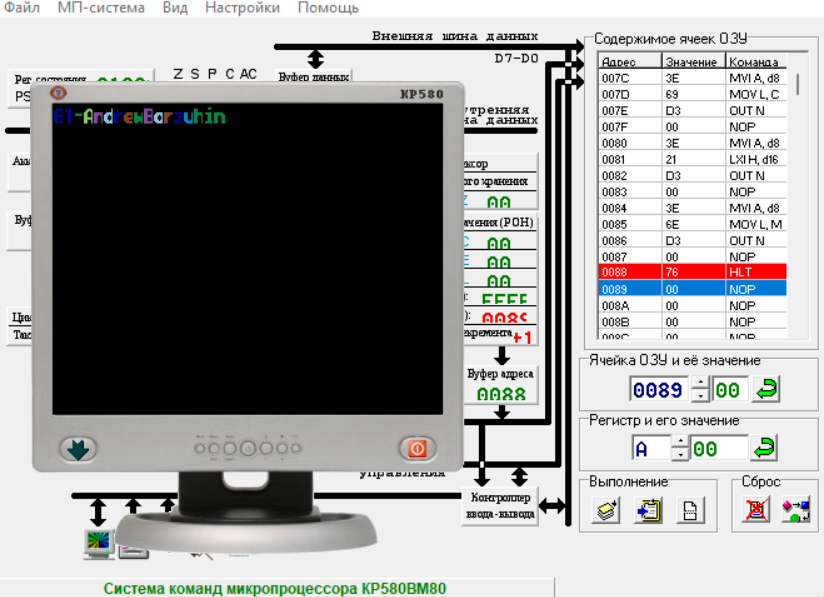


Рисунок 28 – После выполнения программы

# ****Лабораторная работа №14****

**Умножение целых чисел без знака**

**Цель лабораторной работы:**  
 Рассмотреть особенности умножения целых чисел на МП без знака, познакомиться с программированием в машинных кодах и мнемокодах, научиться пользоваться средствами управления и клавиатурой устройств.

Необходимо провести операцию умножения 5710 \* 2810 = 159610(3916 \* 1С16 = 63С16)

Результат поместить в ячейки 0020 и 0021.

Таблица - Программа PRG 1 с умножением целых чисел без знака.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № Ячейки | Значение | Метка | Команда | Комментарий |
| 0000 | 00 |  | NOP | Не используется |
| 0001 | 1E |  | MVI E, d8 | Записать в регистр E число 3916(5710) |
| 0002 | 39 |  | DAD SP |
| 0003 | 0E |  | MVI C, d8 | Записать в регистр C число 1С16 (2110) |
| 0004 | 1С |  | INR E |
| 0005 | AF |  | XRA A | Очистить аккумулятор |
| 0006 | 67 |  | MOV H, A | Очистить регистр A |
| 0007 | 6F |  | MOV L, A | Очистить регистр L |
| 0008 | 57 |  | MOV D, A | Очистить регистр D |
| 0009 | 83 |  | ADD E | A = A + E |
| 000A | C8 |  | RZ | Если А = 0, то выход |
| 000B | AF |  | XRA A | Очистить аккумулятор |
| 000C | 81 |  | ADD C | A – множитель |
| 000D | C8 |  | RZ | Если А = 0, то выход |
| 000E | 1F | C1: | RAR | Арифметический сдвиг аккумулятора вправо |
| 000F | D2 |  | JNC adr | Флаг переноса не установлен (JUMPIFNOCARRY) |
| 0010 | 13 |  | INX D |
| 0011 | 00 |  | NOP |
| 0012 | 19 |  | DAD D | Сдвиг множимого влево |
| 0013 | EB |  | XCHG | пару регистров DE с парой HL (EXCHANGE) |
| 0014 | 29 |  | DAD H | Прибавить к паре регистров HL содержимое пары регистров H |
| 0015 | EB | C2: | XCHG | Обменять пару регистров DE с парой HL (EXCHANGE) |
| 0016 | B7 |  | ORA A | Логическое сложение «ИЛИ» аккумулятора. В данном случае очистить флаг переноса |
| 0017 | C2 |  | JNZ adr | Флаг нуля не установлен (JUMPIFNOTZERO) |
| 0018 | 0E |  | MVI C, d8 |
| 0019 | 00 |  | NOP |
| 001A | 22 |  | SHLD adr | Поместить содержимое регистра L по адресу 0020h, содержимое регистра H помещается в следующую ячейку |
| 001B | 20 |  | - |
| 001C | 00 |  | NOP |
| 001D | 76 |  | HLT | Остановить выполнение программы |

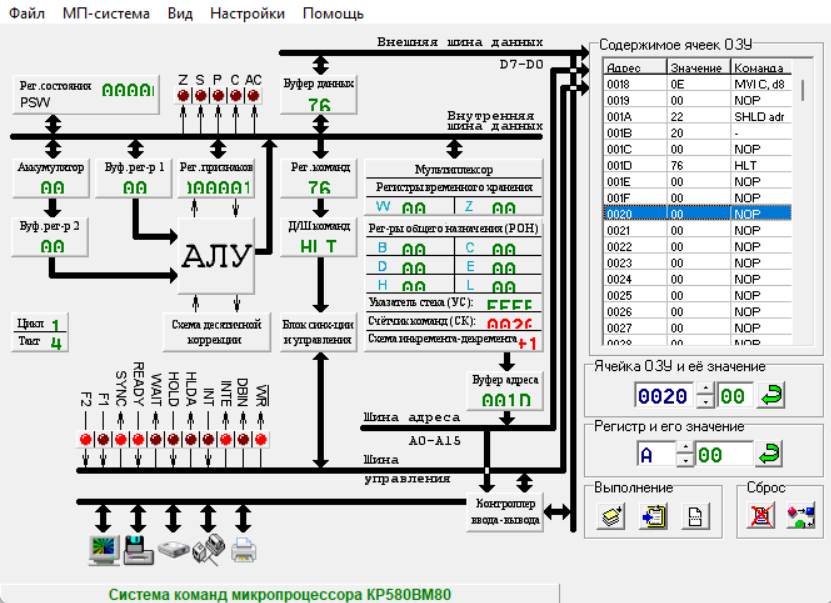


Рисунок 29 – До выполнения программы

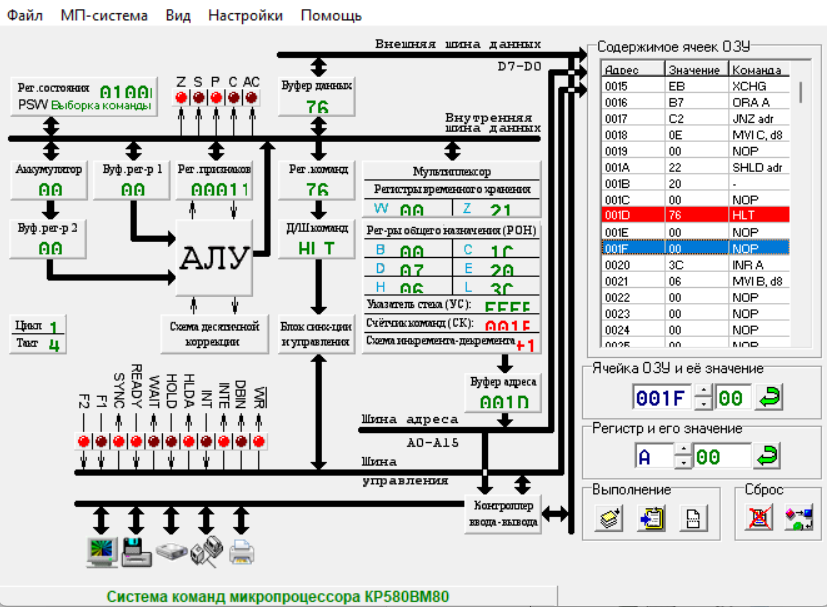


Рисунок 30 – Результат выполнения программы

# ****Лабораторная работа №15****

**Деление целых чисел без знака**

**Цель лабораторной работы:**  
 Изучение алгоритмов деление двух однобайтных чисел, составление и выполнение программы деления двух однобайтных чисел без знака.

Необходимо провести операцию деления 18910 / 2710 = 710 (BD16 / 1816 = 716). Результат поместить в ячейку 0030.

Таблица - Программа PRG 1 с делением целых чисел без знака.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № Ячейки | Значение | Метка | Команда | Комментарий |
| 0000 | 11 | DIVB: | LXI D, 14A0 | ; Загрузка делимого и делителя в пару DE |
| 0001 | BD |  |  | ; |
| 0002 | 18 |  |  | ; |
| 0003 | 21 |  | LXI H, 0008 | ;Загрузка счетчика в регистр Lи очистка регистра Н |
| 0004 | 08 |  |  | ; |
| 0005 | 00 |  |  | ; |
| 0006 | 0Е |  | MVI C, 0 | ; Очистка регистра промежуточного остатка |
| 0007 | 00 |  |  | ; |
| 0008 | F0 | NXTB: | MOV A, E | ; Загрузка делимого в аккумулятор |
| 0009 | 17 |  | RAL | ; Смещение делимого влево на один бит |
| 000А | 5F |  | MOV E, A | ; Пересылка делимого в регистр Е |
| 000В | 79 |  | MOV A, C | ; Загрузка промежуточного остатка в А |
| 000С | 17 |  | RAL | ; Смещение промежуточного остатка влево |
| 000D | 92 |  | SUB D | ; Вычитание делителя от А |
| 000Е | D2 |  | JNC NOADD | ; Если С = 0 - переход на метку NOADD |
| 000F | 12 |  |  | ; |
| 0010 | 00 |  |  | ; |
| 0011 | 82 |  | ADD D | ; Восстановление остатка |
| 0012 | 4F | NOADD: | MOV C, A | ; Пересылка промежуточного остатка в регистр С |
| 0013 | 3F |  | CMC | ; Инверсия признака переполнения |
| 0014 | 7C |  | MOV A, H | ; Запись доли в А |
| 0015 | 17 |  | RAL | ; Смещение доли |
| 0016 | 67 |  | MOV H, A | ; Сохранение доли |
| 0017 | 2D |  | DCR L | ; Уменьшение регистра L |
| 0018 | C2 |  | JNZ NXTB | ; Организация цикла |
| 0019 | 08 |  |  | ; |
| 001А | 00 |  |  | ; |
| 001В | 7C |  | MOV A, H | ; Пересылка младшего байта результата в А |
| 001С | 32 |  | STA 0050 | ; Сохранение содержимого А в памяти по адресу 0030 |
| 001D | 30 |  |  | ; |
| 001E | 00 |  |  | ; |
| 001F | 79 |  | MOV A, H | ; Пересылка старшего байта результата в А |
| 0020 | 32 |  | STA 0051 | ; Сохранение содержимого А в памяти по адресу 0031 |
| 0021 | 31 |  |  | ; |
| 0022 | 00 |  |  | ; |
| 0023 | 76 |  | HLT | ; Прекращение выполнения программы |

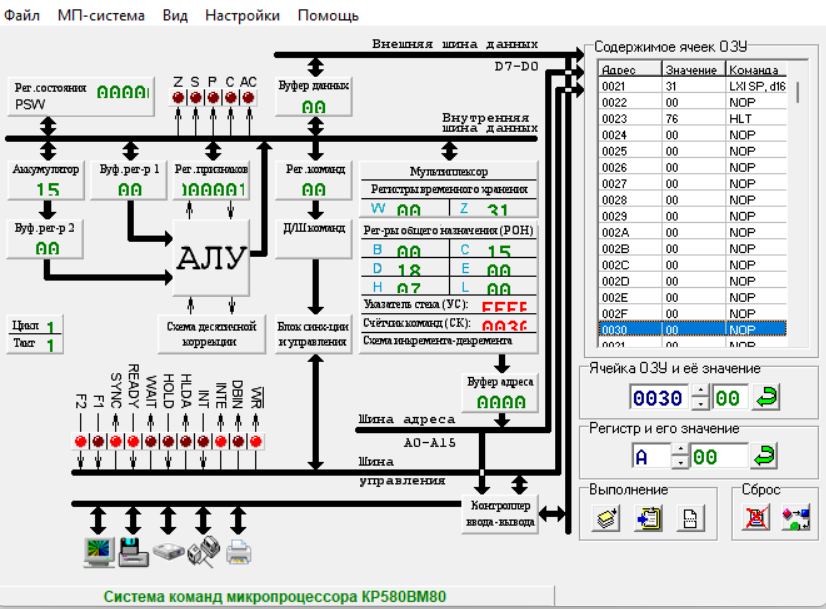


Рисунок 31 – до выполнения программы

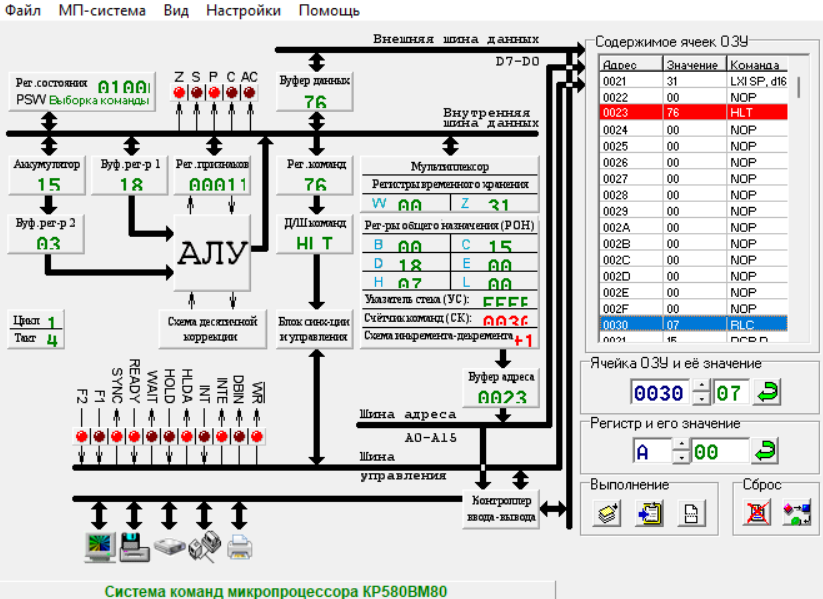


Рисунок 32 – после выполнения программы.

# Лабораторная работа № 16

## Логические операции

**Цель лабораторной работы:**

Исследовать команды логического умножения. Логически умножить числа 210 и 201.

21010 = 1101 00102 = D216

20110 = 1100 10012 = C916

Результатом логического умножения 21010 и 20110 будет число 19210 (1100 00002 и С016)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| & | **1** | **1** | **0** | **1** | **0** | **0** | **1** | **0** |
| **1** | **1** | **0** | **0** | **1** | **0** | **0** | **1** |
| = | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** |

Таблица - Программа PRG 1 выполняющее логическое умножение.

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Команда | Комментарий |
| 0000 | 3E | MVI A, d8 | Загрузить число 210 в десятичной системе счисления в регистр A |
| 0001 | D2 | - |
| 0002 | 06 | MVI B, d8 | Загрузить число 201 в десятичной системе счисления в регистр B |
| 0003 | C9 | - |
| 0004 | A0 | ANA B | A = A & B |
| 0005 | 32 | STA adr | Поместить результат в ячейку 000A |
| 0006 | 0A | - |
| 0007 | 00 | NOP |
| 0008 | 76 | HLT | Остановить программу |



Рисунок 33 – до выполнения программы

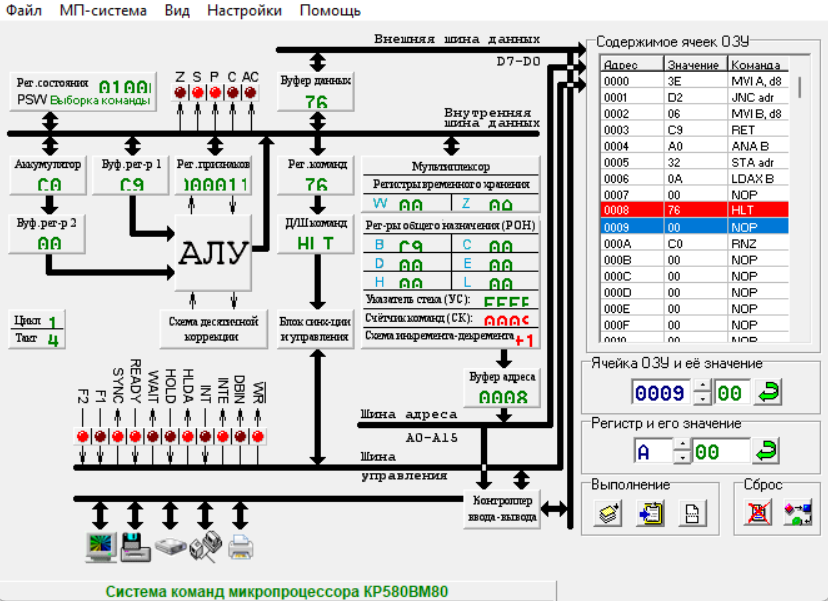


Рисунок 34 – Результат выполнения программы

## Команды обнуления битов

**Цель лабораторной работы:**

Исследовать команды маскирования. Найти код символа “o” (6F) и при помощи маски обнулить 5й бит. В результате должен получиться код прописной буквы “O” (4F).

С помощью байта-маски 1011 1112 (4316) сбрасывается 5-й биты байта 0110 11112 (6F16). Здесь выполняется операция логического умножения чисел:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| & | **0** | **1** | **1** | **0** | **1** | **1** | **1** | **1** |
| **0** | **1** | **0** | **1** | **1** | **1** | **1** | **1** |
| = | **0** | **1** | **0** | **0** | **1** | **1** | **1** | **1** |

В результате данного действий получается код буквы O из o.

Таблица – Программа PRG 2 выполняющая логическое умножение

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Команда | Комментарий |
| 0000 | 3E | MVI A, d8 | Загрузка кода символа “o” в A |
| 0001 | 6F | - |
| 0002 | 06 | MVI B, d8 | Загрузка маски для обнуления 5 бита в B |
| 0003 | 5F | MOV E, A |
| 0004 | A0 | ANA B | A = A & B |
| 0005 | 32 | STA adr | Положить результат в 000A |
| 0006 | 0A | - |
| 0007 | 00 | NOP |
| 0008 | 76 | HLT | Остановить программу |

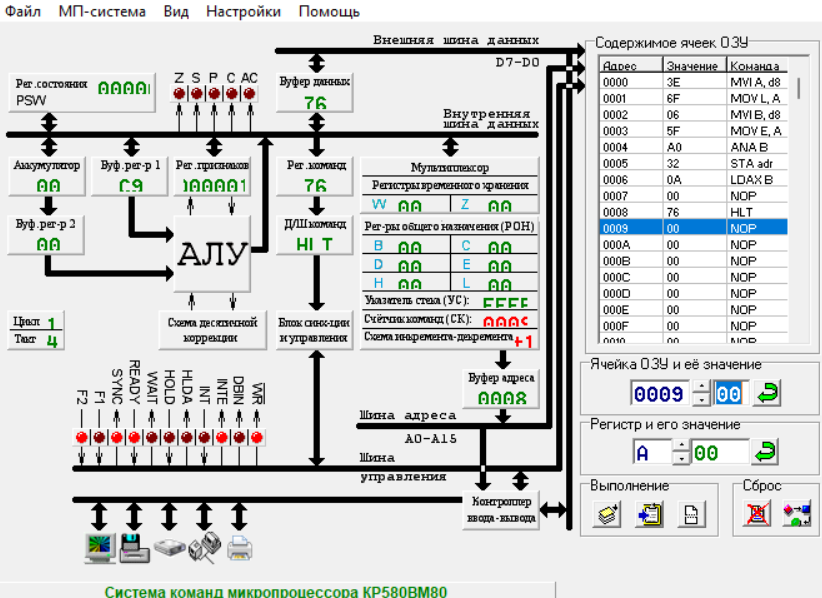


Рисунок 35 – до выполнения программы

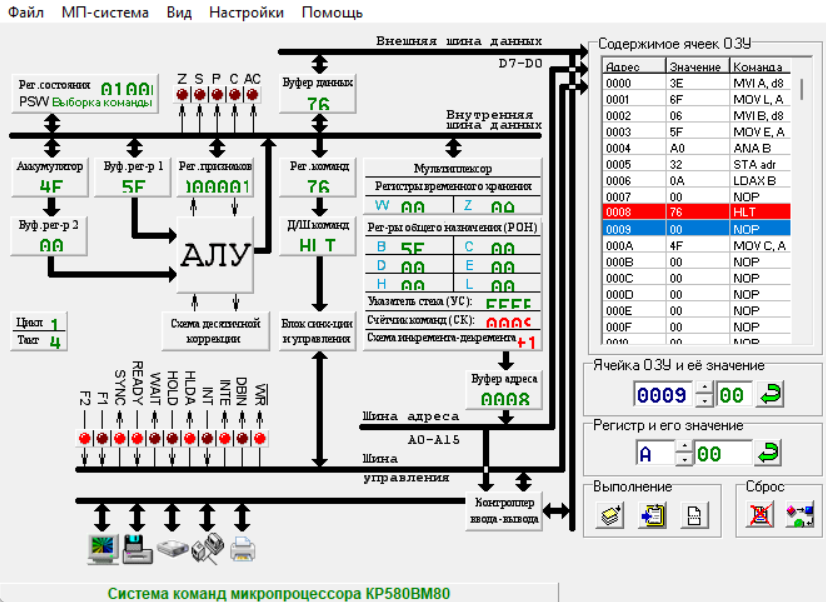


Рисунок 36 – после выполнения программы.

## Команды логического сложения

**Цель лабораторной работы:**

Исследовать команды логического сложения. Сложить числа 151 и 214.

15110 = 1001 01112 = 9716

21010 = 1101 01102 = D616

Здесь выполняется операция логического сложения чисел:

1001 01112 (9716) + 1101 01102 (D616) = 1101 01112 (D716)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | **1** | **0** | **0** | **1** | **0** | **1** | **1** | **1** |
| **1** | **1** | **0** | **1** | **0** | **1** | **1** | **0** |
| = | **1** | **1** | **0** | **1** | **0** | **1** | **1** | **1** |

Таблица – Программа PRG3 выполняющая логическое сложение

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Команда | Комментарий |
| 0000 | 3E | MVI A, d8 | Поместить 97 в регистр A |
| 0001 | 97 | - |
| 0002 | 06 | MVI B, d8 | Поместить D6 в регистр B |
| 0003 | D6 | - |
| 0004 | B0 | ORA B | A = A дизъюнкция B |
| 0005 | 32 | STA adr | Поместить результат в ячейку 000A |
| 0006 | 0A | - |
| 0007 | 00 | NOP |
| 0008 | 76 | HLT | Остановить программу |

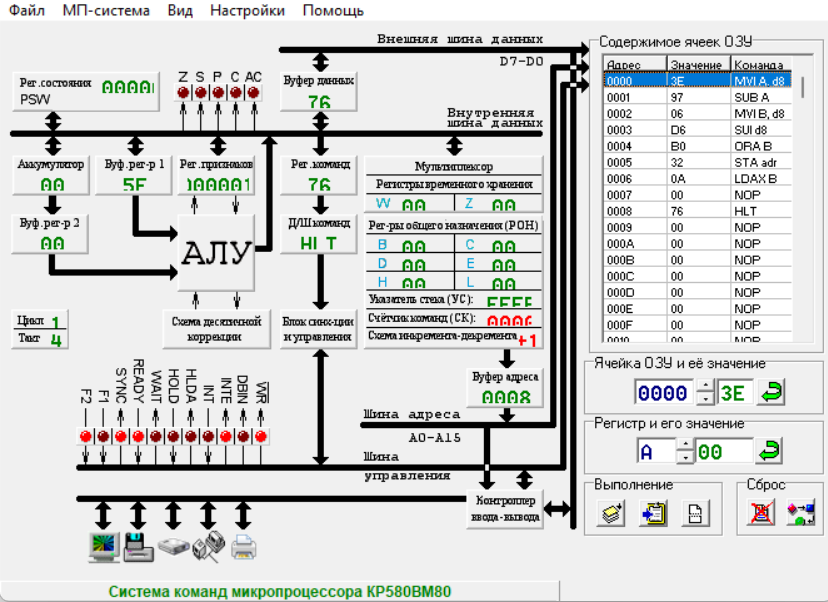


Рисунок 37 – до выполнения программы

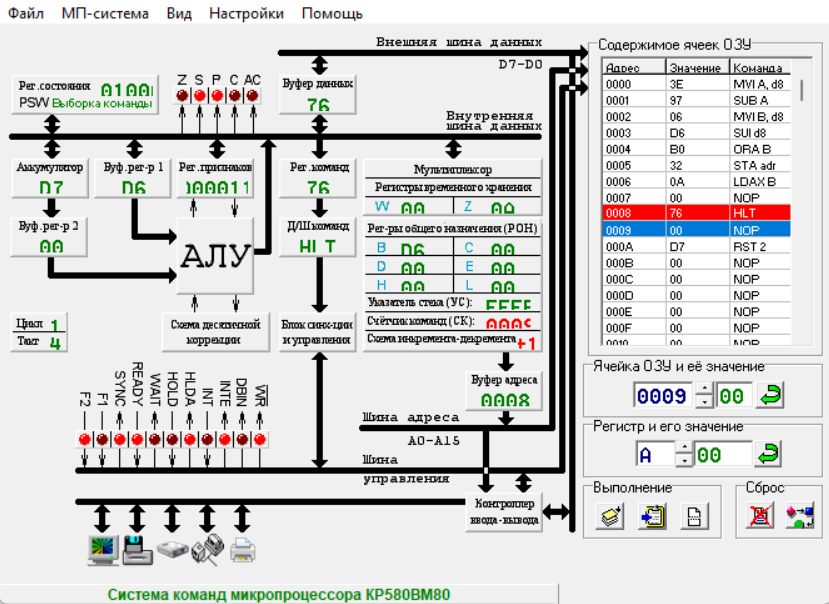


Рисунок 38 – результат работы программы

## Команды восстановления битов

**Цель лабораторной работы:**

Исследовать команд восстановлений битов. В соответствии с таблицей кодировки символов ASCII найти код символа и при помощи маски восстановить пятый бит, чтобы восстановить код буквы “m”.

С помощью байта-маски 0110 00002 (6016) устанавливаются единицы в 2-м и 6-м битах байта 0100 11012 (4D16).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| + | **0** | **1** | **0** | **0** | **1** | **1** | **0** | **1** |
| **0** | **1** | **1** | **0** | **0** | **0** | **0** | **0** |
| = | **0** | **1** | **1** | **0** | **1** | **1** | **0** | **1** |

Таблица – Программа PRG4 выполняющая восстановление бита

|  |  |  |  |
| --- | --- | --- | --- |
| № ячейки | Значение | Команда | Комментарий |
| 0000 | 3E | MVI A, d8 | Положить 4D в регистр A |
| 0001 | 4D | - |
| 0002 | 06 | MVI B, d8 | Положить 60 в регистр B |
| 0003 | 60 | - |
| 0004 | B0 | ORA B | A = A диз B (маска) |
| 0005 | 32 | STA adr | Положить результат в 000A |
| 0006 | 0A | - |
| 0007 | 00 | NOP |
| 0008 | 76 | HLT | Остановить программу |

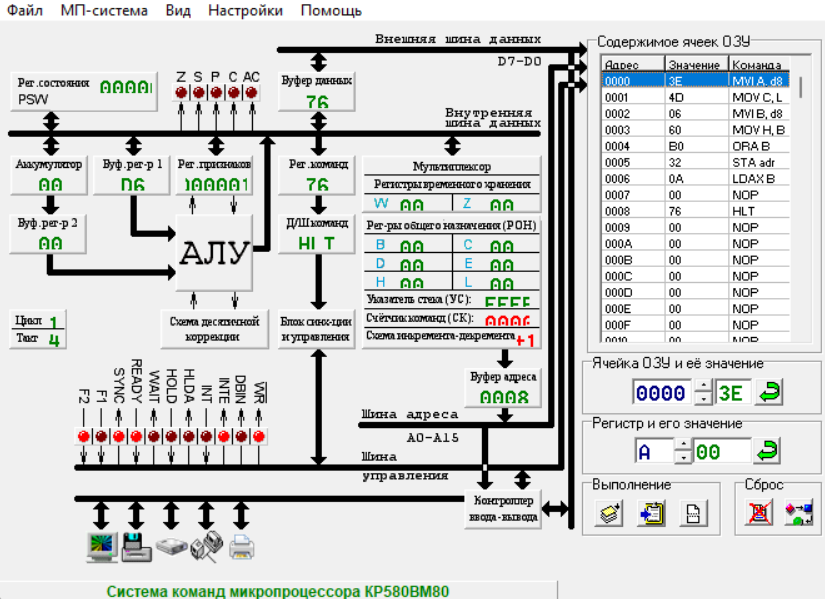


Рисунок 39 – до выполнения программы

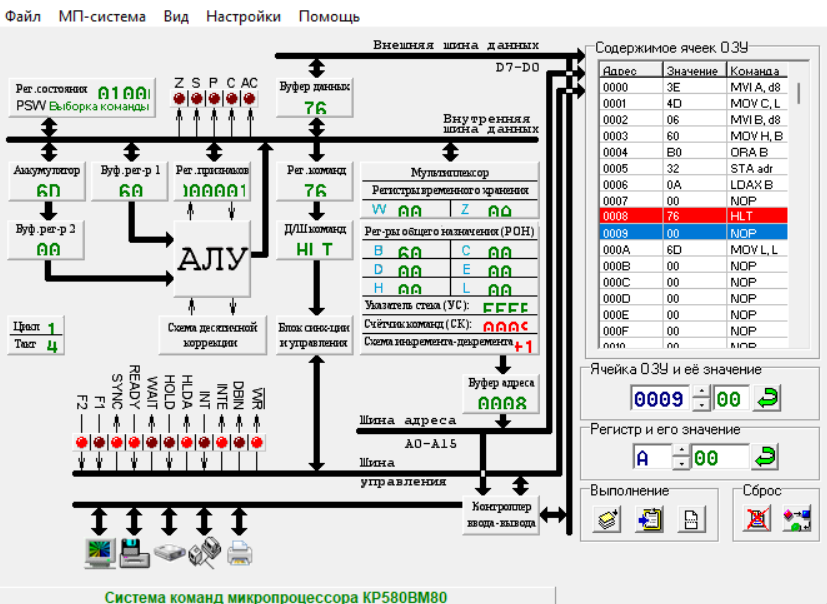


Рисунок 40 – результат выполнения программы

## Команды инвертирования чисел

**Цель лабораторной работы:**

Исследовать команды инвертирования чисел. Инвертировать число 19410   
(1100 00102).

Здесь выполняется операция инвертирования числа

Не 1100 00102 (С216) = 0011 11012 (3D16)

|  |
| --- |
| Не |
| **1** | **1** | **0** | **0** | **0** | **0** | **1** | **0** |
|  | **0** | **0** | **1** | **1** | **1** | **1** | **0** | **1** |

Таблица – Программа PRG5 выполняющая инверсию числа

|  |  |  |
| --- | --- | --- |
| Значение | Команда | Комментарий |
| 3E | MVI A, d8 | Загрузка числа C2 в аккумулятор |
| C2 | - |
| 2F | CMA | Инверсия аккумулятора |
| 32 | STA adr | Положить результат по адресу 000A |
| 0A | - |
| 00 | NOP |
| 76 | HLT | Остановить программу |

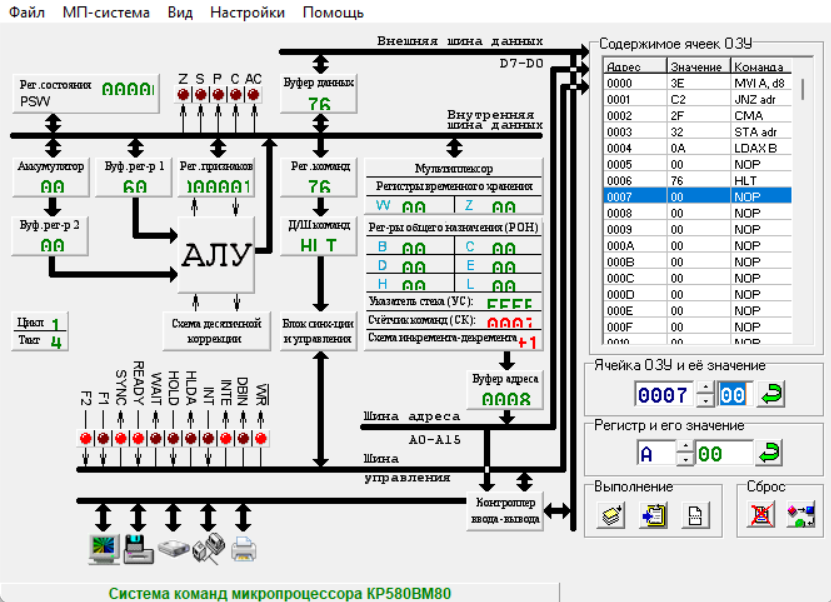


Рисунок 41 – до выполнения программы

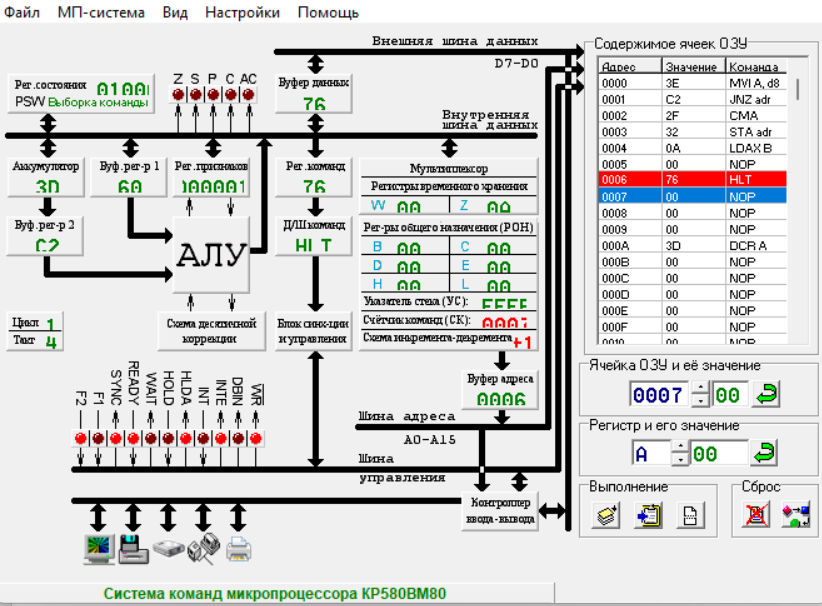


Рисунок 42 – результат выполнения программы

## Инвертирование флага переноса

**Цель лабораторной работы:**

Исследовать команды переноса флага.

Таблица – Программа, выполняющая инверсию флага переноса

|  |  |  |  |
| --- | --- | --- | --- |
| № ячейки | Значение | Команда | Комментарий |
| 0000 | 3F | CMC | Инверсия флага переноса |
| 0001 | 76 | HLT | Остановить программу |

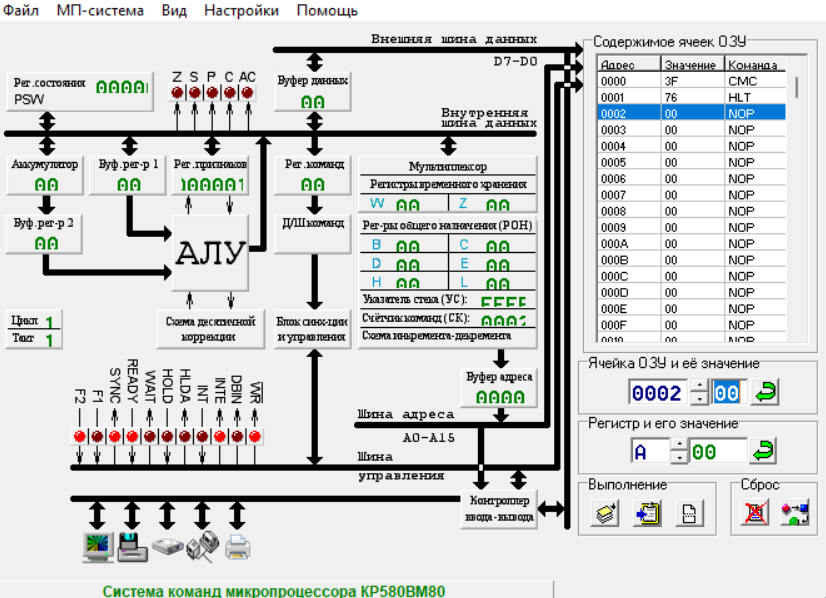


Рисунок 43 – до выполнения программы.

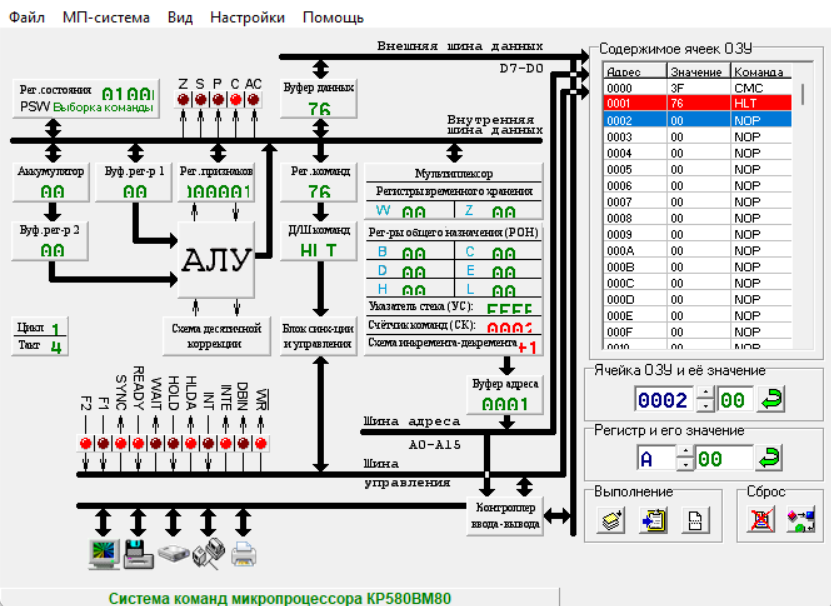


Рисунок 44 – результат работы программы

## Инвертирование заданных битов числа

**Цель лабораторной работы:**

Исследовать команды инвертирования чисел. Инвертировать 2й, 4й, 5й бит числа 8110(0101 00012 и 5116).

Здесь выполняется операция инвертирования 2, 4 и 5 бита байта 0101 00012 (5116) с помощью байта-маски 001101002 (3416).



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |

В результате получается число 0110 01012 (6516).

Таблица – Программа PRG6 выполняющая инвертирование заданных битов числа

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Команда | Комментарий |
| 0000 | 3E | MVI A, d8 | Загрузить число 51 в регистр A |
| 0001 | 51 | - |
| 0002 | 06 | MVI B, d8 | Загрузить число 34h в регистр B |
| 0003 | 34 | INR M |
| 0004 | A8 | XRA B | A = A ^ B |
| 0005 | 32 | STA adr | Положить результат в 000A |
| 0006 | 0A | - |
| 0007 | 00 | NOP |
| 0008 | 76 | HLT | Завершить программу |

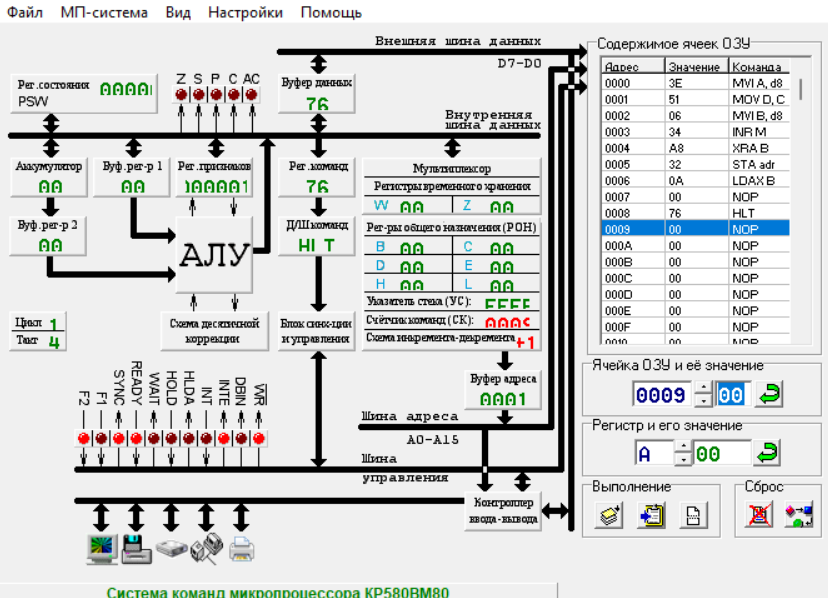


Рисунок 45 – до выполнения программы

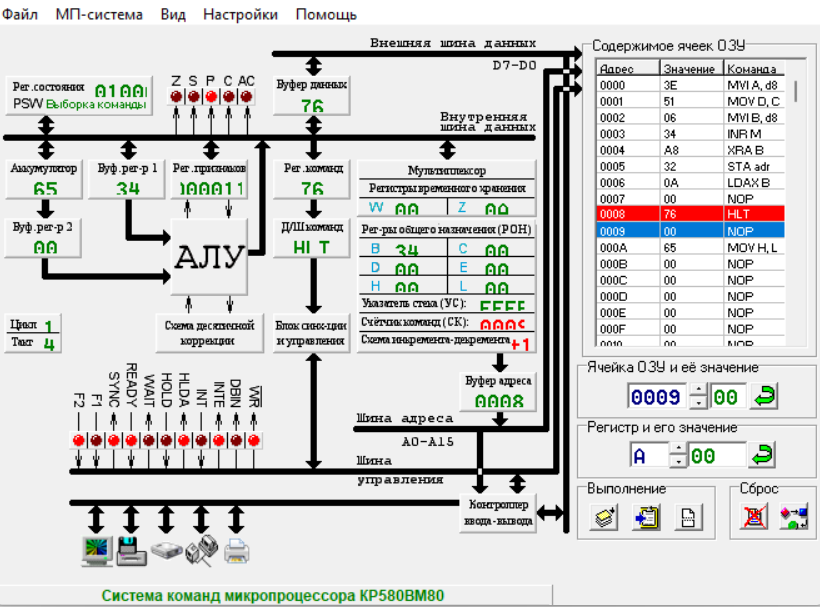


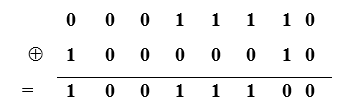
Рисунок 46 – результат выполнения программы

## Сравнение чисел

**Цель лабораторной работы:**

Исследовать команды поразрядного сравнения чисел. Сравнить числа 3010(0001 11102) и 13010(1000 00102).

Здесь выполняется операция поразрядного сравнения числа 0001 11102 (1E16) и 1000 00102(8216).



В результате получается число 1001 11002 (9C16).

Таблица PRG – Программа, выполняющая сравнение чисел

|  |  |  |  |
| --- | --- | --- | --- |
| № Ячейки | Значение | Команда | Комментарий |
| 0000 | 3E | MVI A, d8 | Загрузить число 1E в регистр A |
| 0001 | 1E | - |
| 0002 | 06 | MVI B, d8 | Загрузить число 82 в регистр B |
| 0003 | 82 | - |
| 0004 | A8 | XRA B | A = A ^ B |
| 0005 | 32 | STA adr | Поместить результат в ячейку 000A |
| 0006 | 0A | - |
| 0007 | 00 | NOP |
| 0008 | 76 | HLT | Завершить программу |

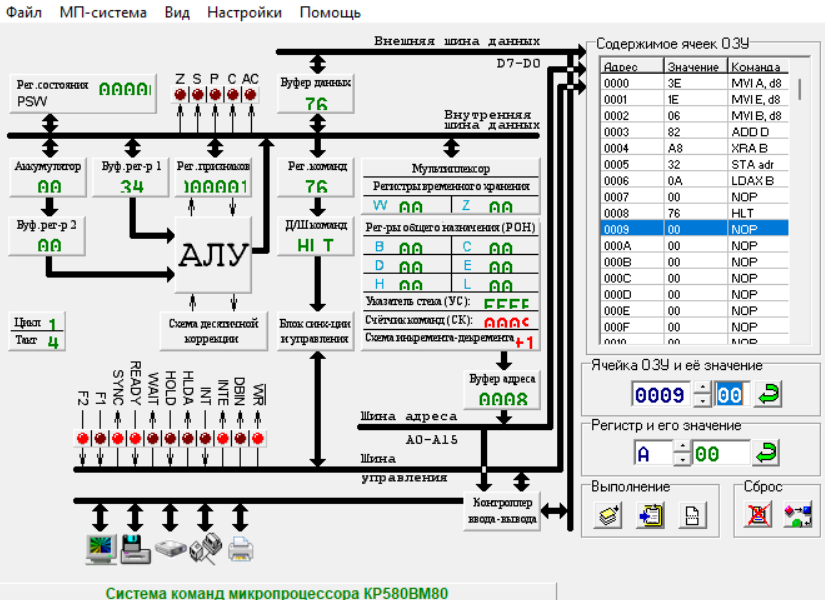


Рисунок 47 – до выполнения программы

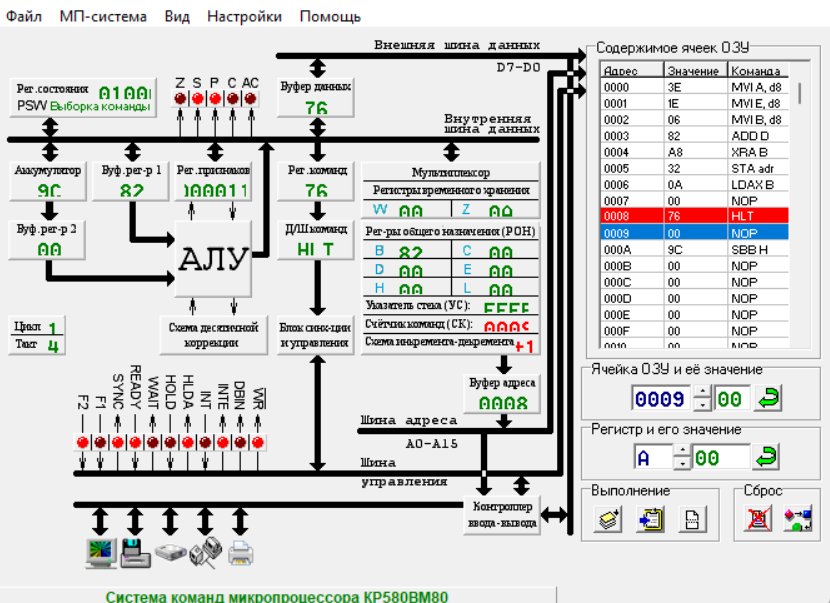


Рисунок 48 – результат выполнения программы