**САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ**

**БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

УП01

Специальность 09.02.07

«Информационные системы и программирование»

Квалификация «Программист»

Руководитель учебной практики:

Смирнова И. П.

Другие преподаватели:

Матысик И. А.

Полякова А. Н.

Фомин А. В.

Выполнил студент группа 493:

Сидоров антон дмитриевич

Санкт-Петербург, 2023

**Содержание**

[1 УП01.01. Разработка программных модулей 2](#_Toc128566337)

[2 УП01.02. Поддержка и тестирование программных модулей 2](#_Toc128566338)

[2.1 Задание расчёт сырья 2](#_Toc128566339)

[2.2 Unit-tests 3](#_Toc128566340)

[2.3 Test case 3](#_Toc128566341)

[3 УП01.03. Разработка мобильных приложений 4](#_Toc128566342)

[3.1 Цель работы 4](#_Toc128566343)

[3.2 Название работы 4](#_Toc128566344)

[3.3 Макеты окон 4](#_Toc128566345)

[3.4 Диаграмма базы данных 7](#_Toc128566346)

[3.5 Тестирование документации API 8](#_Toc128566347)

[4 УП01.04. Системное программирование 9](#_Toc128566348)

[4.1 Лабораторная работа 1 9](#_Toc128566349)

[4.2 Лабораторная работа №2. Исследование команд прямой адресации 12](#_Toc128566350)

[4.3 Лабораторная работа 3. Исследование команд непосредственной адресации 14](#_Toc128566351)

[4.4 Лабораторная работа 4. Исследование команд косвенной адресации 17](#_Toc128566352)

# УП01.01. Разработка программных модулей

# УП01.02. Поддержка и тестирование программных модулей

## Задание расчёт сырья

2.1.1 Задание

Для того чтобы в производстве могли быстро и одинаково рассчитывать количество необходимого сырья для производства той или иной продукции, необходимо разработать библиотеку классов.

Чтобы система правильно интегрировалась вам необходимо обязательно следовать правилам именования библиотек, классов и методов в них. В случае ошибок в рамках именования ваша работа не может быть проверена и ваш результат не будет зачтен. Классы и методы должны содержать модификатор public (если это реализуемо в рамках платформы), чтобы внешние приложения могли получить к ним доступ.

В качестве названия для библиотеки необходимо использовать: WSUniversalLib. Вам необходимо загрузить исходный код проекта с библиотекой в отдельный репозиторий с названием, совпадающим с названием проекта.

2.1.2 Спецификация метода

Метод должен принимать идентификатор типа продукции, идентификатор типа материала, количество необходимой продукции для производства, ширину продукции и длину продукции, а возвращать целое число – количество необходимого сырья с учетом возможного брака сырья.

|  |  |
| --- | --- |
| Библиотека классов | WSUniversalLib.dll |
| Название класса | Calculation |
| Название метода | GetQuantityForProduct() |
| Входящие  обязательные  параметры | int productType,  int materialType,  int count,  float width,  float length |
| Возвращаемые  параметры | int |

2.1.3 Программный код

## Unit-tests

## Test case

# УП01.03. Разработка мобильных приложений

## Цель работы

## Название работы

Разработка чата

## Макеты окон

3.1.1 Примечания

Окна в приложении могут отличаться от своих макетов.

3.1.2 Макеты окон

Макеты окон приложения показаны на рисунке 1.

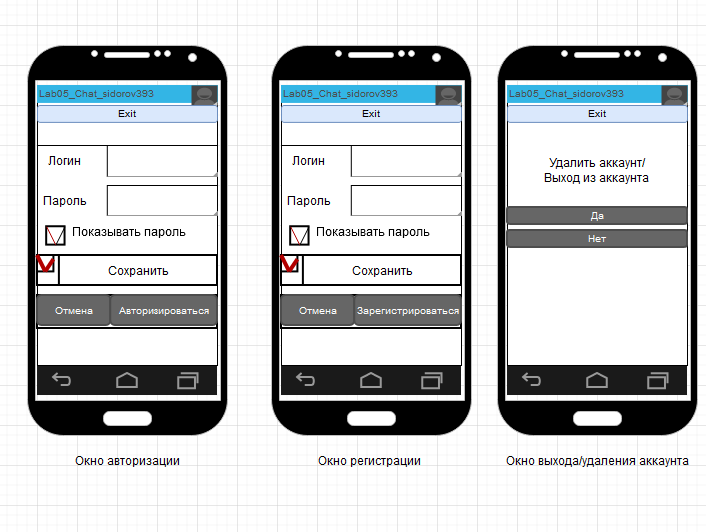


Рисунок 1 – Макеты окон

Также, есть, ещё окна, макеты которых показаны на рисункуе 2.

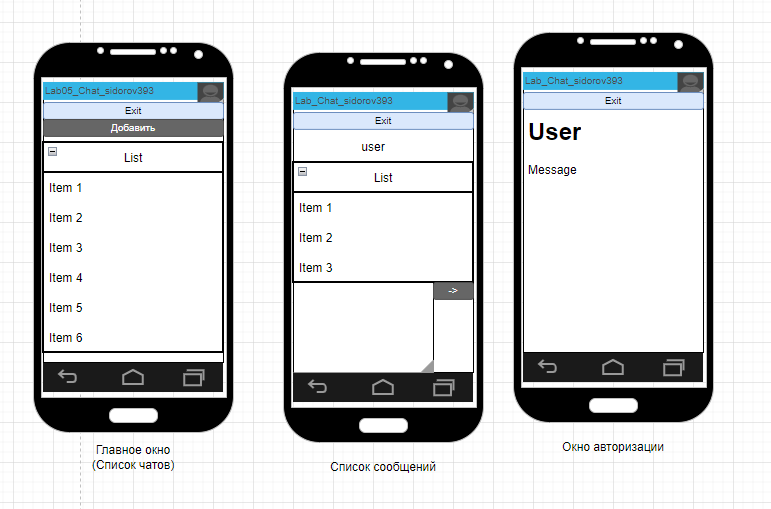


Рисунок 2 – Макеты окон

Окно главного и редактора url-ссылки меню показано на рисунке 3.

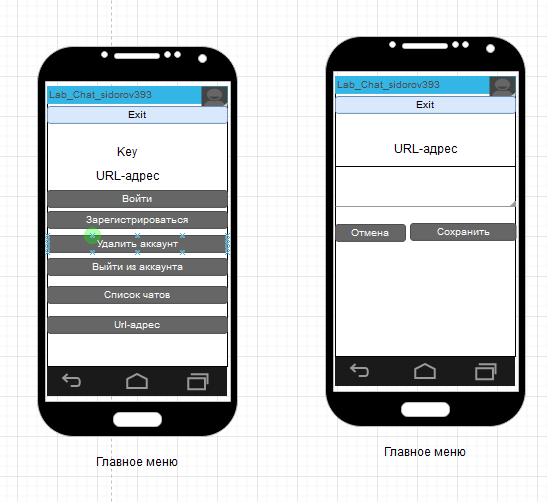


Рисунок 3 – Главное меню

3.1.3 Описание окон (функции)

Окно авторизации:

* Поле ввода логина
* Поле ввода пароля
* Флажок для показа/скрытия пароля
* Флажок для сохранения/несохранения данных на устройстве
* Кнопка отмены входа
* Кнопка входа в аккаунт

Окно регистрации:

* Поле ввода логина
* Поле ввода пароля
* Флажок для показа/скрытия пароля
* Флажок для сохранения/несохранения данных на устройстве
* Кнопка отмены входа
* Кнопка создания аккаунта и входа в него

Окно выхода из аккаунта:

* Предупреждающая надпись
* Кнопки подтверждения (да/нет)

Окно удаления аккаунта:

* Предупреждающая надпись
* Кнопки подтверждения (да/нет)

Окно списка чатов:

* Список чатов, с возможностью перейти к чату
* Кнопка добавления чата, перенаправляющая к списку аккаунтов

Окно списка аккаунтов:

* Список аккаунтов, с возможностью перейти к аккаунту и начать с ним чат

Окно редактора url-ссылки:

* Поле ввода url-адреса
* Кнопка сохранения url-адреса
* Кнопка отмены сохранения url-адреса

Главное меню (Первое открывающееся окно):

* Отображаемый url-адрес
* Отображаемый ключ сессии
* Кнопка авторизации, перенаправляющая на окно авторизации
* Кнопка регистрации, перенаправляющая на окно регистрации
* Кнопка выхода из аккаунта
* Кнопка удаления аккаунта
* Кнопка перехода к списку чатов
* Кнопка, перенаправляющая к редактору url-ссылок

Все окна, также, имеют кнопу выхода, перенаправляющую на предыдущее окно.

## Диаграмма базы данных

База данных предназначена для локального хранения данных об аккаунте и о url-ссылок.

Диаграмма базы данных представлена на рисунке 5.

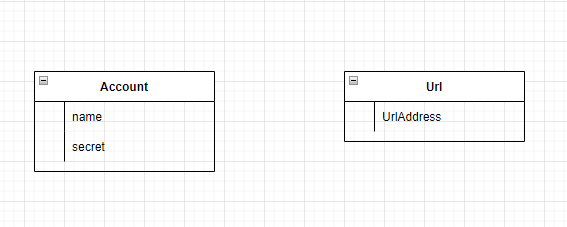


Рисунок 5 – Диаграмма базы данных

Таблица *Account* предназначена для хранения данных о пользователе: логин (*name*) и пароль (*secret*).

Таблица *UrlAddress* предназначена для хранения адреса сервера с API в поле *UrlAddress*.

## Тестирование документации API

# УП01.04. Системное программирование

## Лабораторная работа 1

4.1.1 Цель работы

Изучение архитектуры микропроцессора КР580.

4.1.2 структура МП

Числовые обозначения структуры представлены на рисунке 1.

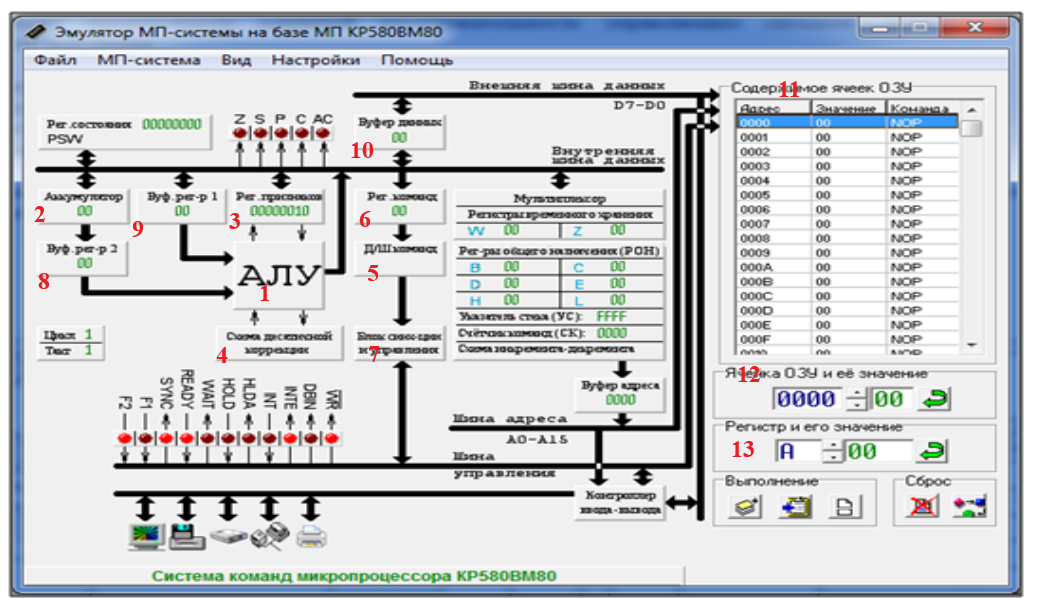


Рисунок 1 – структура МП.

Обозначения (Красным цветом на фотографии):

1. 8-разрядное арифметико-логическое устройство АЛУ (ALU)
2. аккумулятор (А)
3. регистр признаков RS, фиксирующий признаки, вырабатываемые АЛУ в процессе выполнения команды
4. десятичный корректор (DAA), выполняющий перевод информации из двоичной в двоично-десятичную форму
5. дешифратор команд (DCU)
6. регистр команд (IR), предназначенный для хранения первого байта команды, содержащего код операции
7. схема управления и синхронизации (CU), формирующая последовательности управляющих сигналов для работы ALU и регистров
8. однонаправленный 16-разрядный буферный регистр адреса (ВА)
9. двунаправленный 8-разрядный буферный регистр данных (BD)
10. регистр временного хранения операндов (RGb)
11. Пространство памяти и ввода-вывода
12. Значения ячеек памяти
13. Значения регистров

4.1.3 таблица регистров, которые имеет МП

|  |  |  |
| --- | --- | --- |
| **Регистр** | **Назначение** | **Разрядность** |
| регистрами общего назначения (РОН) | 8-разрядные регистры F, A вместе с 16-разрядными регистрами HL, SP и PC образуют стандартный регистровый набор микропроцессора с аккумулятором. Этот набор расширен четырьмя 8-разрядными регистрами общего назначения (РОН): B, C, D, E, которые в некоторых командах объединяются в 16-разрядные парные регистры BC и DE. Младшими регистрами пары являются соответственно регистры C и E. Введение РОН позволило создать достаточно эффективный микропроцессор с широкими функциональными возможностями. | 8 бит |
| аккумулятор А | используется в подавляющем большинстве команд логической и арифметической отработки. Обычно он адресуется неявно и служит как источником операнда, так и приемником результата. Благодаря этому в командах ВМ80А явно указывается только один операнд. | 8 бит |
| регистр HL | как правило, служит адресным регистром. При косвенной регистровой адресации он хранит 16-разрядный адрес основной памяти. В этом случае к нему ссылаются с помощью мнемоники М (Memory), например:  MOV A, M; содержимое ячейки (HL) заносится в аккумулятор | 16 бит |
| Регистры PC и SP | выполняют свою обычную функцию счетчика команд и указателя стека. |  |
|  |  |  |
|  |  |  |

**4.1.4 блок-схема функционирования МП во время выполнения команды сложения содержимого аккумулятора и регистра В, имеющая мнемоническое обозначение ADD B.**

Данная блок-схема показана на рисунке 1.

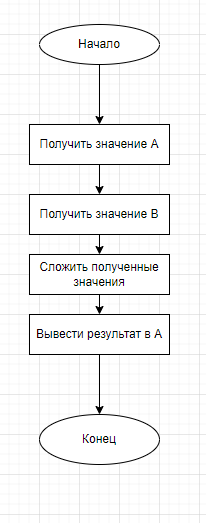


Рисунок 2 – Блок-схема

**4.1.5 таблица флагов**

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

## Лабораторная работа №2. Исследование команд прямой адресации

4.2.1 Цель лабораторной работы

Целью данной работы является ознакомление с командами микропроцессора КР580 для прямой адресации

4.2.2 Программный код

|  |  |  |  |
| --- | --- | --- | --- |
| 0000 | 3A | LDA adr | загрузить содержимое в аккумулятор |
| 0001 | 0A | LDAX B | из ячейки памяти с адресом Ah |
| 0002 | 00 | NOP |  |
| 0003 | 32 | STA adr | загрузить содержимое аккумулятора в |
| 0004 | 0B | DCX B | ячейку памяти с адресом Bh |
| 0005 | 00 | NOP |  |
| 0006 | 00 | NOP |  |
| 0007 | 76 | HLT | Остановка выполнения программы |
| 0008 | 00 | NOP |  |
| 0009 | 00 | NOP |  |
| 000A | 40 | MOV B, B | Значение в ячейке памяти |
| 000B | 00 | NOP |  |
| 000C | 00 | NOP |  |
| 000D | 00 | NOP |  |
| 000E | 00 | NOP |  |
| 000F | 00 | NOP |  |
| 0010 | 00 | NOP |  |

4.2.3 Ввод операндов при прямой адресации

Скрин ввода операндов и начала работы программы представлен на рисунке 3.

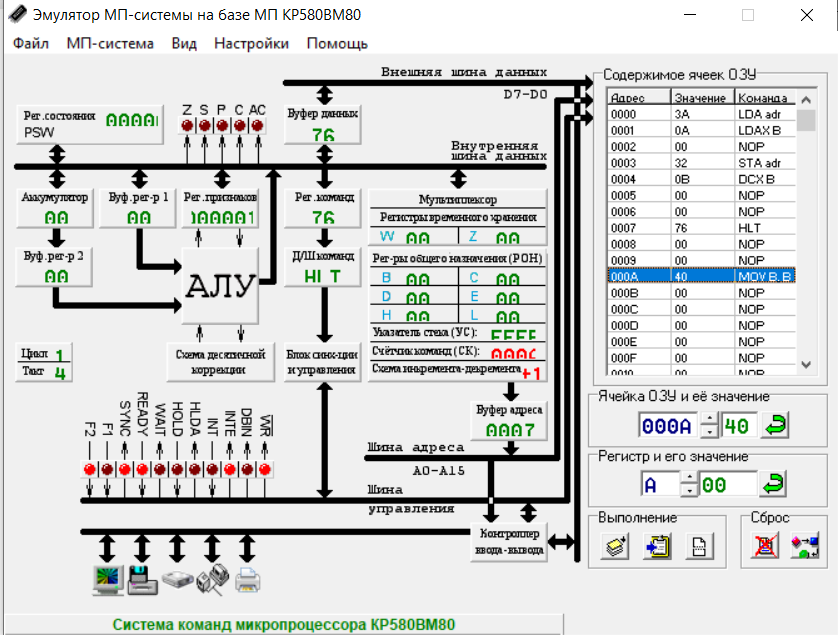


Рисунок 3 – Ввод операндов при прямой адресации

Операнд вводится в ячейку памяти *Ah*.

4.2.4 Результат работы модели при прямой адресации

Скрин результата работы программы представлен на рисунке 4.

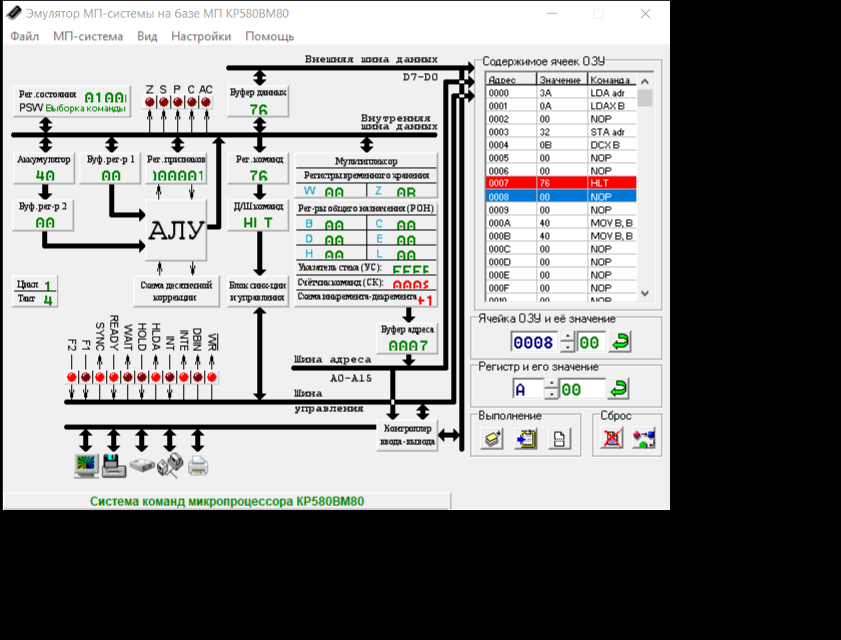


Рисунок 4 – Результат работы модели при прямой адресации

Результат выводится в ячейку памяти *Bh*.

4.2.5 Заключение

Освоено использование команд для прямой адресации.

## Лабораторная работа 3. Исследование команд непосредственной адресации

4.3.1 Цель лабораторной работы

Целью данной работы является ознакомление с командами микропроцессора КР580 для непосредственной адресации

4.3.2 Программный код

|  |  |  |  |
| --- | --- | --- | --- |
| **№ Ячейки** | **Значение** | **Команда** | **Комментарий** |
| 0000 | 3E | MVI A, d8 | Положить в регистр A (аккумулятор) |
| 0001 | 20 | - | число 20h |
| 0002 | 06 | MVI B, d8 | Положить в регистр B |
| 0003 | 05 | DCR B | число 5h |
| 0004 | 26 | MVI H, d8 | Положить в регистр H |
| 0005 | D1 | POP D | число D1h |
| 0006 | D7 | RST 2 | Перейти к команде по в ячейке с адресом 10h |
| 0007 | 00 | NOP |  |
| 0008 | 00 | NOP |  |
| 0009 | 00 | NOP |  |
| 000A | 00 | NOP |  |
| 000B | 00 | NOP |  |
| 000C | 00 | NOP |  |
| 000D | 00 | NOP |  |
| 000E | 00 | NOP |  |
| 000F | 00 | NOP |  |
| 0010 | 00 | NOP |  |
| 0011 | 00 | NOP |  |
| 0012 | 00 | NOP |  |
| 0013 | 00 | NOP |  |
| 0014 | 76 | HLT | Остановить работу программы |
| 0015 | 00 | NOP |  |

4.3.3 Ввод операндов при непосредственной адресации

Скрин ввода операндов и начала работы программы представлен на рисунке 5.

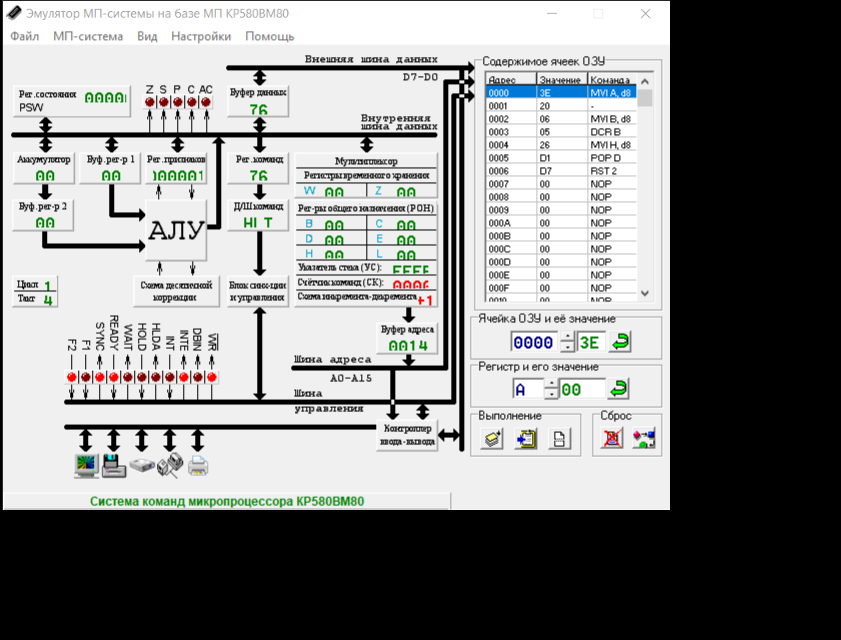


Рисунок 5 – Ввод операндов при непосредственной адресации

В регистр *A* будет вводиться число *20h*, В регистр *B* – *5h*, *H – D1h*.

4.3.4 Результат работы модели при непосредственной адресации

Скрин результата работы программы представлен на рисунке 6.

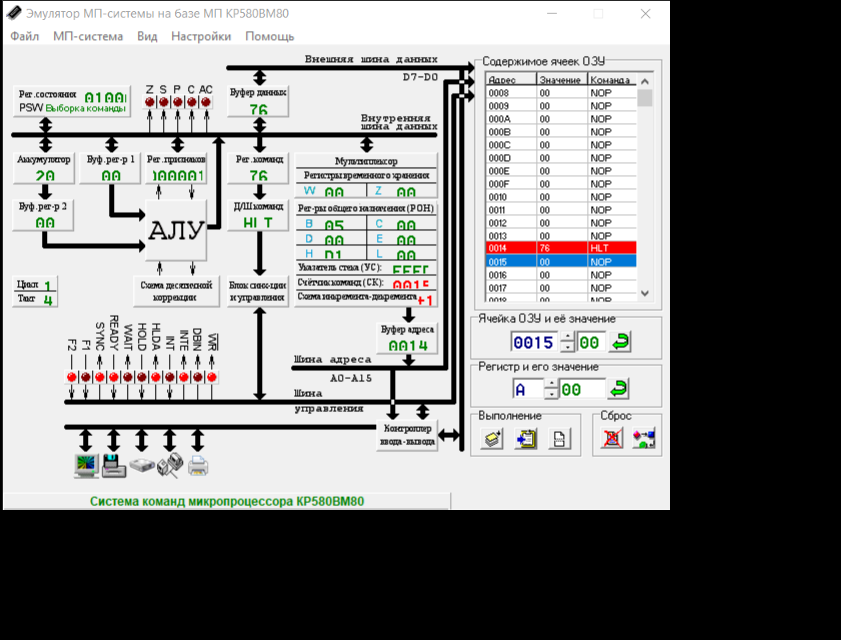


Рисунок 6 – Результат работы модели при непосредственной адресации

Значения регистров приведены в таблице ниже:

|  |  |  |  |
| --- | --- | --- | --- |
|  | A= | 20 |  |
| B= | 05 | C= | 00 |
| D= | 00 | E= | 00 |
| H= | D1 | L= | 00 |
| W= | 00 | Z= | 00 |

4.3.5 Заключение

Освоено использование команд для непосредственной адресации.

## Лабораторная работа 4. Исследование команд косвенной адресации

4.4.1 Цель лабораторной работы

Целью данной работы является ознакомление с командами микропроцессора КР580 для косвенной адресации

4.4.2 Программный код