

**Министерство образования и науки Российской Федерации**  
**Федеральное государственное бюджетное**  
**образовательное учреждение**  
**высшего образования**

**Новгородский государственный университет**  
**имени Ярослава Мудрого**

**Гурьянов С.А.**

**Система отладки микроконтроллеров**

**семейства MCS51**

**Великий Новгород**  
**2021**

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Новгородский государственный университет имени Ярослава Мудрого»  
Институт электронных и информационных систем

---

Кафедра радиосистем

**Гурьянов С.А.**

**Система отладки микроконтроллеров  
семейства MCS51**

Методическое пособие по курсам лабораторных работ  
и практических занятий

**Великий Новгород**

**2021**

**УДК 004.382.4**

**ББК**

**Гурьянов С. А.** Система отладки микроконтроллеров семейства MCS51: метод. пособие / С. А. Гурьянов ; Новгород. гос. ун-т им. Ярослава Мудрого. – Великий Новгород, 2021. — 34 с.

В учебном пособии рассмотрена структура лабораторного макета на основе микроконтроллера семейства MCS51, система команд микроконтроллера, примеры программ, приведено описание устройств ввода-вывода.

Учебное пособие отвечает новым образовательным стандартам и предназначено для подготовки специалистов по направлению подготовки:

- 11.03.01– «Радиотехника». ПРОФ Радиотехнические средства передачи, приема и обработки сигналов
- 11.04.01 – «Радиотехника». ПРОФ Радиотехнические средства передачи, приема и обработки сигналов
- 11.04.04 – «Электроника и нанoeлектроника» Магистерская программа «Системы и устройства передачи, приема и обработки сигналов»
- 15.03.06 – «Мехатроника и робототехника»

в дисциплинах:

- Цифровые устройства и микропроцессоры
- Проектирование цифровых устройств на ПЛИС и МК
- Проектирование цифровых устройств и систем
- Технологии систем управления и контроля
- Микропроцессорная техника в мехатронике и робототехнике

## СОДЕРЖАНИЕ

|  |    |
|--|----|
| ВВЕДЕНИЕ .....   | 5  |
| 1 МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА MCS51 .....                           | 6  |
| 1.1 Структура микроконтроллера I8051 .....                         | 6  |
| 1.2 Направление развития микроконтроллеров семейства MCS51 .....   | 10 |
| 2 РЕАЛИЗАЦИЯ ЛАБОРАТОРНОГО МАКЕТА.....                             | 12 |
| 2.1 Структура лабораторного макета.....                            | 12 |
| 2.2 Описание схемы электрической принципиальной макета .....       | 14 |
| 2.3 Работа лабораторного макета .....                              | 17 |
| 3 РАБОТА В СИСТЕМЕ ОТЛАДКИ MCS-51-IDE-XP .....                     | 18 |
| 3.1 Запуск программы .....   | 18 |
| 3.2 Подключение лабораторного макета к компьютеру.....             | 18 |
| 3.3 Работа с файлами .....   | 19 |
| 3.4 Компиляция .....   | 19 |
| 3.5 Загрузка и запуск программы.....                               | 20 |
| 3.6 Отладка .....  | 21 |
| 3.7 Остановка программы .....                                      | 22 |
| 3.8 Вывод на экран данных из памяти.....                           | 22 |
| 4 ТЕСТОВЫЕ ПРОГРАММЫ .....   | 24 |
| 4.1 Тестовая программа приема и передачи цифрового сигнала .....   | 24 |
| 4.2 Тестовая программа приема и передачи аналогового сигнала ..... | 25 |
| СПИСОК ЛИТЕРАТУРЫ.....   | 27 |
| Приложение А .....   | 28 |
| Схема электрическая принципиальная лабораторного макета .....      | 28 |
| Приложение Б .....   | 30 |
| Система команд микроконтроллера MCS51 .....                        | 30 |

## ВВЕДЕНИЕ

Методическое пособие предназначено студентам специальности

- 11.03.01 - «Радиотехника» ПРОФ Радиотехнические средства передачи, приема и обработки сигналов
- 11.04.01 – «Радиотехника» ПРОФ Радиотехнические средства передачи, приема и обработки сигналов
- 11.04.04 – «Электроника и нанoeлектроника» Магистерская программа Системы и устройства передачи, приема и обработки сигналов
- 15.03.06 – «Мехатроника и робототехника»

В пособии рассматривается лабораторный макет, реализованный на основе микроконтроллера семейства MCS51.

Приводится краткое описание данного микроконтроллера и его основных блоков.

Представлено описание схемы электрической принципиальной лабораторного макета и его конструкции.

Приводится описание необходимого программного обеспечения, используемого для написания программ, компиляции и их отладки.

Представлены примеры тестовых программ для работы с цифровыми сигналами, АЦП и ЦАП.

## 1 МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА MCS51

### 1.1 Структура микроконтроллера I8051

Первый микроконтроллер семейства MCS51 был выпущен компанией Intel в 1980 году. Микроконтроллеры семейства MCS-51 представляют собой функционально завершенные 8-разрядные однокристальные микро ЭВМ [2,3]. В настоящее время многие компании продолжают производство микроконтроллеров семейства MCS-51. Существуют реализации данного семейства для ПЛИС.

Все микроконтроллеры семейства MCS-51 имеют общую систему команд. Наличие дополнительного оборудования влияет только на количество регистров специальных функций, с помощью которых осуществляется настройка параметров работы соответствующих встроенных узлов.

Микроконтроллеры семейства MCS51 имеют в своем составе следующие аппаратные средства:

- 8-разрядный центральный процессор, в состав которого входят однобайтное арифметико-логическое устройство и схема аппаратурной реализации команд умножения и деления;
- запоминающее устройство программ (различного типа в зависимости от назначения),
- оперативное запоминающее устройство данных (резидентная память данных), используемое для организации регистровых банков, стеков и хранения пользовательских данных;
- не менее 32-х двунаправленных интерфейсных линий, индивидуально программируемых на ввод или вывод информации;
- два (или более) 16-битных таймера/счетчика, используемых для подсчета внешних событий, организации временных задержек и тактирования коммуникационного порта;

- асинхронный канал дуплексного последовательного ввода/вывода информации;
- блок двухуровневого векторного прерывания от пяти источников;
- тактовый генератор, схему синхронизации и управления,
- и другие устройства ввода-вывода (в зависимости от серии).

Структурная схема базового микроконтроллера семейства MCS51 представлена на рис.1.1.

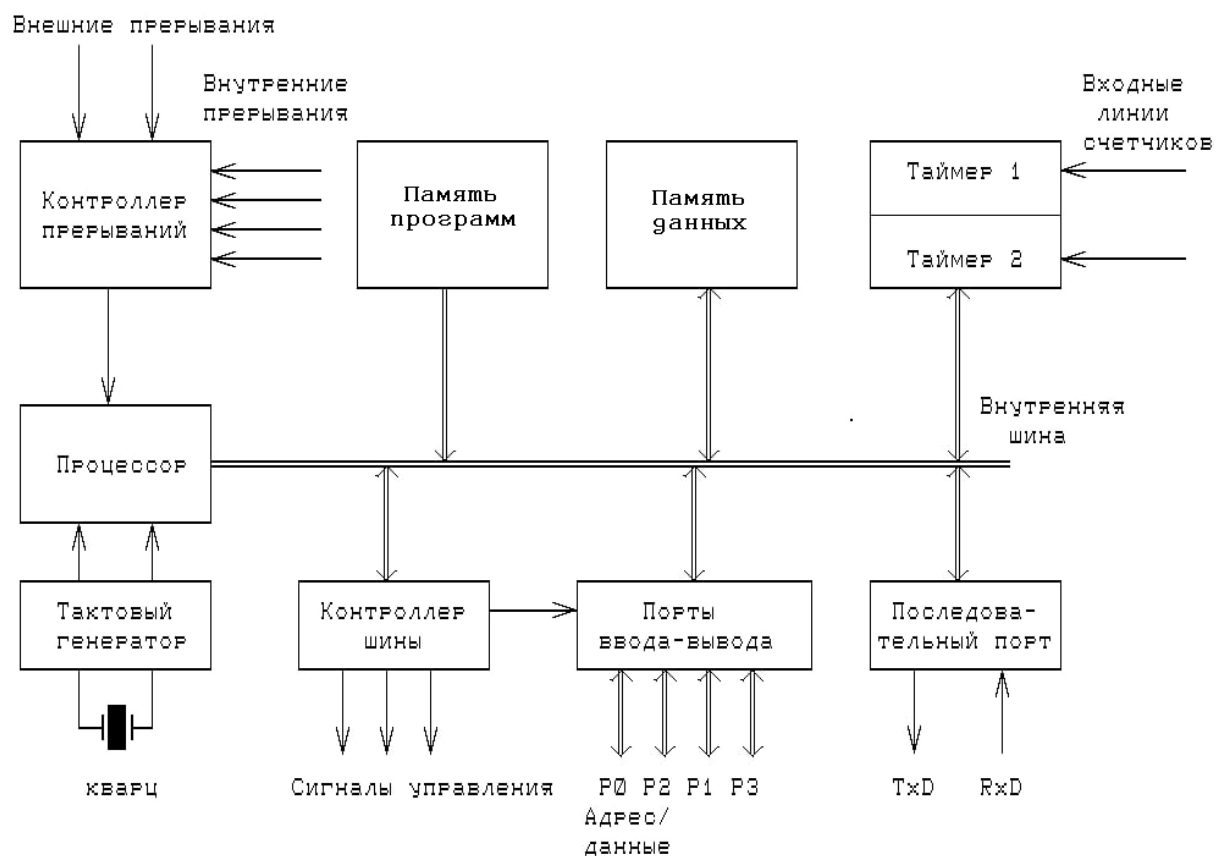


Рисунок 1.1 - Структурная схема микроконтроллера семейства MCS51

Семейство MCS51 имеет режим работы с внешними дополнительными БИС постоянной и оперативной памяти. Это свойство микроконтроллера используется при реализации лабораторного макета. При этом адресное пространство микроконтроллеров может быть расширено до 64 Кбайт. Также использование адресации внешних устройств как ячеек памяти позволит доста-

точно просто подключить дополнительные устройства ввода-вывода цифровых и аналоговых сигналов (АЦП, ЦАП).

На рис. 1.2 представлено условное графическое обозначение базового микроконтроллера MCS51, а в таблице 1.1 – их назначение.

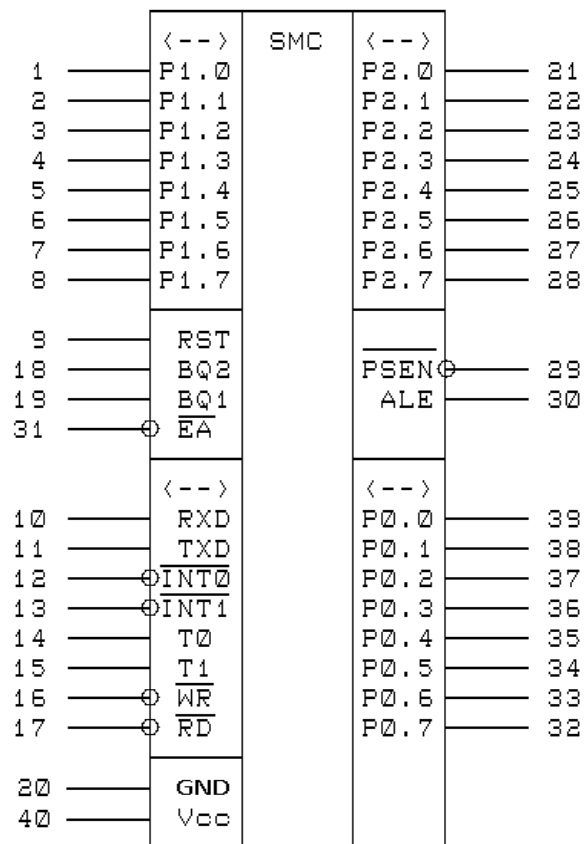


Рис. 1.2. Условное графическое обозначение микроконтроллера MCS51.

В приложении А представлена система команд микроконтроллера MCS51 [2,3].



Назначение выводов микроконтроллера MCS51

| N     | Обозначение | Назначение   |
|-------|-------------|--|
| 1-8   | P1.0 – P1.7 | 8-разрядный двунаправленный порт ввода/вывода P1.  |
| 9     | RST         | Reset. Сигнал общего сброса  |
| 10-17 | P3.0 - P3.7 | 8-разрядный двунаправленный порт ввода/вывода P3 с дополнительными (альтернативными) функциями:  |
| 10    | P3.0 – RXD  | Receive Data pin. Последовательные данные приемника  |
| 11    | P3.1 – TXD  | Transmit Date. Последовательные данные передатчика   |
| 12    | P3.2 - INT0 | Вход внешнего прерывания 0   |
| 13    | P3.3 - INT1 | Вход внешнего прерывания 1   |
| 14    | P3.4 - T0   | Вход таймера/счетчика 0  |
| 15    | P3.5 - T1   | Вход таймера/счетчика 1  |
| 16    | P3.6 – WR   | WRITE. Выход сигнала для записи во внешнюю память  |
| 17    | P3.7 – RD   | READ. Выход сигнала для чтения из внешней памяти   |
| 18    | BQ2         | Выводы для подключения кварцевого резонатора   |
| 19    | BQ1         |  |
| 20    | GND         | Общий вывод  |
| 21-28 | P2.0 - P2.7 | 8-разрядный двунаправленный порт ввода/вывода P2.<br>Выход адреса A8-A15 для внешней памяти  |
| 29    | PSEN        | Programm Store Enable. Сигнал разрешения внешней памяти программ   |
| 30    | ALE/PROG    | Adress Latch Enable. Строб адреса внешней памяти   |
| 31    | EA/VPP      | External Adress. Сигнал отключения резидентной памяти программ   |
| 32-39 | P0.0 – P0.7 | 8-разрядный двунаправленный порт ввода/вывода P0.<br>При обращении к внешней памяти мультиплексированная шина младшего байта адреса A0-A7 и байта данных |
| 40    | Vcc         | Напряжение питания +5v   |

## 1.2 Направление развития микроконтроллеров семейства MCS51

Основными направлениями развития этого семейства является [3]:

- улучшение электрических характеристик контроллера,
- увеличение объема резидентной памяти программ и данных,
- использование новых принципов организации архитектуры,
- расширение состава периферийных устройств,
- использование современных средств внутрисхемной диагностики и программирования.

Технические характеристики изменяются в следующих направлениях:

- Переход от NMOS технологии и CHMOS к полностью статической технологии, снижение напряжения питания до 1.8 В.
- Увеличение мощности выходных буферов до 24 мА.
- Увеличение тактовой частоты до 24-33 МГц (фирма Siemens до 40 МГц).
- Объем резидентной памяти программ увеличен до 32-64 Кбайт, а резидентной памяти данных - до 256-1024 байт (в некоторых модификациях – до 2 Кбайт).
- Объем линейного адресного пространства (внешнего) увеличился до 16 МБ (MCS-251).
- Одним из недостатков 8051 является неоптимальная диаграмма выборки, дешифрирования и выполнения команд, что связано, с частичным использованием гарвардской архитектуры. Дальнейшее развитие архитектуры связано с реализацией полного гарвардского принципа (раздельные магистрали для памяти команд и данных), использованием конвейерной обработки, увеличением области стека, расширением набора команд (обработка 16+ и +32 – разрядных данных, новые способы адресации и т.д.) при сохранении программной совместимости с младшими моделями семейства.

- Последние разработки MCS-51 выполняются, как правило, на базе флэш-памяти и использовании дополнительного ЭППЗУ для хранения данных.
- Состав периферийных устройств может включать: многоканальный АЦП 8 или 10-разрядный, схемы анализа и формирования временных событий, расширенное количество последовательных портов различного назначения (SPI, I2C, CAN, USB, JTAG), сторожевой таймер, широтно-импульсные модуляторы ШИМ и так далее.
- Расширено количество параллельных восьмиразрядных портов (до 6) и внутренних прерываний, увеличено число внешних и внутренних прерываний.
- Наряду с этим выпускаются МК с ограниченным составом периферии, но размещенных в малогабаритных корпусах с 14, 20, 24 внешними выводами.
- В базовой модели 8051 программирование осуществляется от внешнего программатора. Последние разработки включают резидентные средства программирования и отладки на базе последовательных портов и интерфейса JTAG.

Наиболее широко распространенными клонами MCS-51 являются: семейство AT89xxxxxx фирмы Atmel, 89Cxxx, 8xC5xx фирмы Philips, DS500x, DS80xxxx, DS87xxxx фирмы Dallas Semiconductor.

## 2 РЕАЛИЗАЦИЯ ЛАБОРАТОРНОГО МАКЕТА

### 2.1 Структура лабораторного макета

Структурная схема лабораторного макета представлена на рис. 2.1.

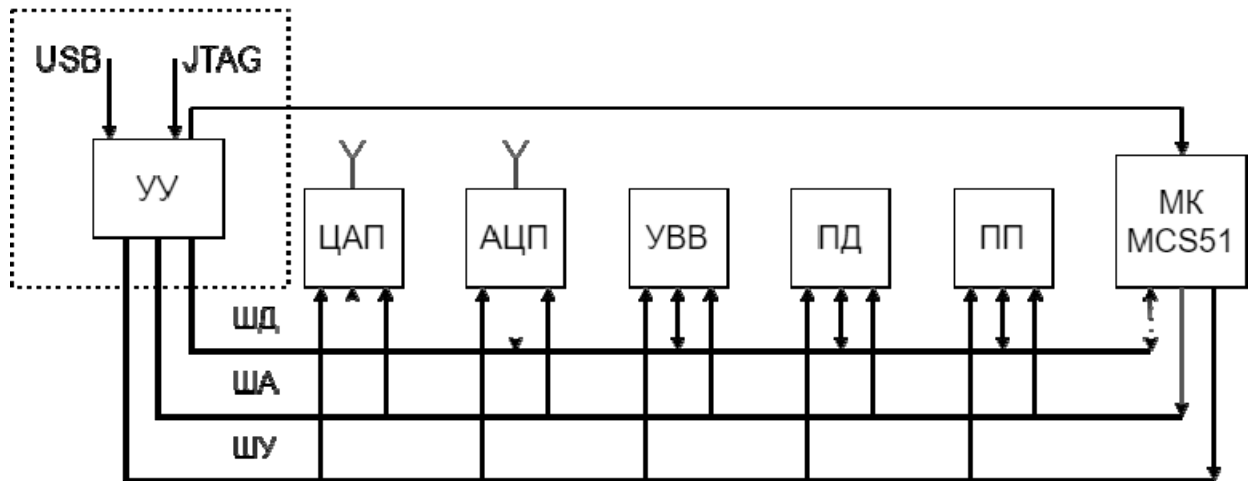


Рисунок 2.1 - Структурная схема лабораторного макета

При реализации лабораторного макета использовалась возможность микроконтроллера MCS51 работать с внешней памятью программ [2,3]. Поэтому лабораторный макет построен по классической структуре микропроцессорной системой с тремя шинами [1]. В этом случае исследуемый микроконтроллер для работы использует внешнюю память программ (ПП). Доступ к внешней памяти программ возможен по системным шинам адреса и данных от исследуемого микроконтроллера и от персонального компьютера через встроенное в макет устройство управления. При работе от персонального компьютера происходит запись исследуемой программы во внешнюю память программ лабораторного макета. Затем управление передается исследуемому микроконтроллеру для выполнения заданной программы.

Использование адресации внешних устройств как ячеек памяти позволяет подключить дополнительные устройства ввода-вывода цифровых (УВВ) и аналоговых сигналов (АЦП, ЦАП).

Структура лабораторного макета, представленная на рис.2.1, включает:

1. УУ –устройство управления;

2. МК MCS51 – исследуемый микроконтроллер;
3. ПП – внешняя память программ;
4. ПД – внешняя память данных;
5. УВВ – цифровые устройства ввода-вывода;
6. ЦАП - цифро-аналоговый преобразователь;
7. АЦП – аналого-цифровой преобразователь.

Доступ ко всем устройствам системы осуществляется по трем шинам управления: ША – шина адреса, ШД – шина данных и ШУ - шина управления. При такой организации доступ ко всем устройствам осуществляется как к ячейкам памяти с заданным адресом. Адреса устройств ввода-вывода формируются с помощью дешифратора адреса в УУ.

Связь лабораторного макета с персональным компьютером осуществляется по интерфейсу USB. Для отладки программы функционирования УУ используется встроенный интерфейс JTAG.

Конструкция лабораторного макета представлена на рисунке 2.2.

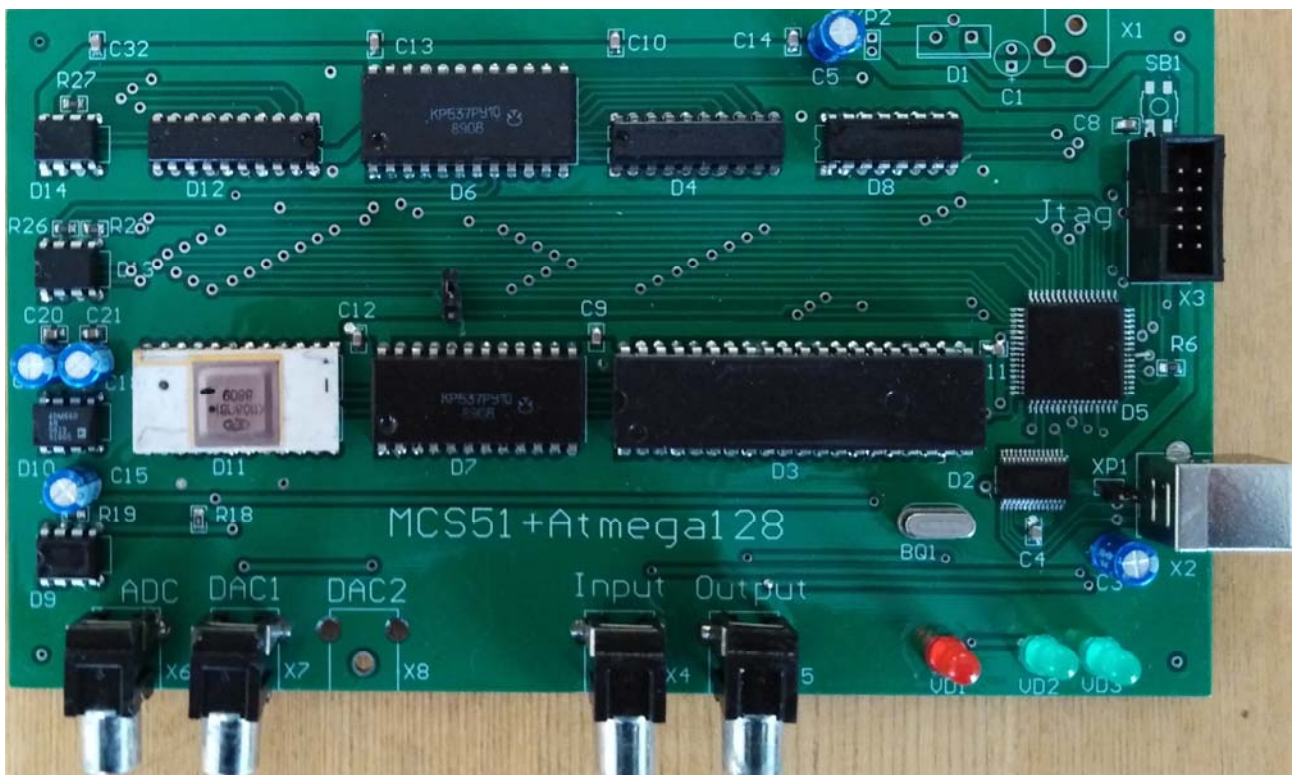


Рисунок 2.2 - Лабораторный макет MCS51

## 2.2 Описание схемы электрической принципиальной макета

Схема электрическая принципиальная макета представлена в Приложении А на 2-х листах.

Программирование макета осуществляется с помощью управляющего устройства (УУ), реализованного на микроконтроллере ATmega128 (D5). УУ принимает данные от персонального компьютера по специальному протоколу связи и записывает их во внешнюю память программ (ПП).

Функционирование макета осуществляется под управлением исследуемого микроконтроллера MCS51 (D3) с использованием внешней ПП.

Обращение к ПП осуществляется по трем системным шинам.

УУ служит для приёма и записи отлаживаемой программы в память программ (ПП) с персонального компьютера и передачи результатов работы программы на персональный компьютер для вывода на экран монитора. УУ реализовано на микроконтроллере Atmega128 (D5), имеющем поддержку работы с внешней памятью и встроенный интерфейс UART.

Для преобразования протокола шины UART в протокол USB используется USB-UART преобразователь на микросхеме FT232RL (D2).

Организация системы по классической структуре позволяет удобно обращаться к каждому устройству, по заданному для него адресу, который с помощью дешифратора преобразуется в необходимый сигнал управления (D8).

Адреса подключаемых устройств для микроконтроллера MCS51 распределены следующим образом (старшие 4 разряда адреса - A15, A14, A13, A12):

0000 – адрес памяти программ ПП CSP,

0001 – адрес памяти данных ПД CSO,

0010 – сигнал записи данных в ЦАП CSdac (при активном сигнале WR),

0011 – сигнал запуска АЦП Start (при активном сигнале WR),

0011 – сигнал чтения данных с АЦП Radc (при активном сигнале RD).

Программное обеспечение записывается во внешнюю память программ (ПП), реализованную на микросхеме K537PY10 объемом 2 Кбайта (D6). Адреса внешней памяти программ 0000h-03FFh.

Для записи больших массивов данных в макете реализовано внешнее оперативное запоминающее устройство данных на микросхеме K537PY10 объемом 2 Кбайта (D7). Адреса внешней памяти программ 0100h-07FFh.

Цифро-аналоговый преобразователь (ЦАП) (D12) и аналогово-цифровой преобразователь (АЦП) (D11) предназначены для изучения базовых принципов обработки и формирования аналоговых сигналов.

АЦП (D11) реализован на микросхеме - 1108ПВ1. На входе АЦП установлен операционный усилитель (D9). Первый каскад является буфером для входного сигнала (D9.1). Второй каскад предназначен для смещения входного аналогового сигнала на уровень  $U_{оп}/2 = 1.25В$  (D9.2) для получения однополярного входного сигнала АЦП от 0 до  $U_{оп}$ . Запуск АЦП осуществляется сигналом Start от дешифратора при выполнении соответствующей команды записи во внешнюю память данных. Чтение данных из АЦП осуществляется по сигналу Rdac при выполнении соответствующей команды чтения из внешней памяти данных. Входной сигнал для АЦП подается на разъем X6 «ADC».

В качестве ЦАП выбрана микросхема AD7528 (D12). Это сдвоенный 8-разрядный умножающий ЦАП. На выходах ЦАП включены буферные усилители со смещением выходного сигнала. Выходной сигнал данного канала имеет размах от 0 до  $U_{оп}$ , равного 2.5В. Выходной сигнал этого канала смещен на значение  $U_{оп}/2 = 1.25В$  на операционном усилителе (D13.2) для получения двухполярного сигнала размахом от  $-U_{оп}/2$  до  $+U_{оп}/2$ . Выходной сигнал этого канала подается на разъем X7 «DAC1». Опорным напряжением канала В ЦАП является выходной сигнал канала А, что позволяет реализовать функцию умножающего ЦАП. Выходной буфер этого канала реализован также на операционном усилителе (D14). Выходной сигнал этого канала может подаваться на разъем X8 «DAC2». Запись данных в ЦАП осуществляется по сигналу CSdac от дешифратора при выполнении соответствующей команды записи во внешнюю память данных.

УВВ организованы в виде разъёмов для подключения цифровых сигналов. Входы и выходы подключены через буферные резисторы к выводам мик-

роконтроллера. В макете используются биты порта P1 – P1.0 и P1.3. Разъем X4 «Input» соединен с контактом порта P1.0. На разъем X5 «Output» поступает сигнал с контакта порта P1.3.

Питание лабораторного макета реализуется через разъем X2 по USB кабелю от персонального компьютера.

Операционные усилители имеют двуполярное питание  $\pm 5\text{В}$ . Для формирования напряжения  $-5\text{В}$  для питания операционных усилителей используется инвертор напряжения на микросхеме ADM660 (D10).

Для индикации режимов работы используются три светодиода:

VD1 – режим работы микроконтроллера MCS51, подключен к выводу микроконтроллера P1.7;

VD2 – сигнал сброса микроконтроллера MCS51, включен при активном сигнале сброса;

VD3 - режим работы устройства управления, включен при программировании микроконтроллера MCS51.



### 2.3 Работа лабораторного макета

- В соответствии с заданием на лабораторную работу оформляется текст программы в любом тестовом редакторе или программе функционирования лабораторного макета MCS-51-IDE-XP. Текст программы сохраняется в файле с расширением \*.A51.
- Разработанная программа компилируется на персональном компьютере с помощью программы MCS-51-IDE-XP. Результатом компиляции является файл с расширением \*.HEX.
- Код программы из файла \*.HEX с помощью УУ лабораторного макета загружается во внешнюю память программ (ПП) макета.
- После окончания загрузки программы УУ формирует сигнал запуска для работы исследуемого микроконтроллера.
- Микроконтроллер MCS51 выполняет исследуемую программу.
- Циклическая программы выполняется в цикле без точки останова.
- Линейная программа останавливается на точке останова. По умолчанию точкой останова является адрес за последней командой программы. Программа MCS-51-IDE-XP позволяет перед запуском программы установить точку останова в любом месте исследуемой программы.
- Для остановки работы программы по нажатию кнопки останова в программе MCS-51-IDE-XP УУ формирует сигнал останова.
- После остановки работы исследуемого микроконтроллера программа MCS-51-IDE-XP считывает результаты работы программы (содержимое регистров, резидентной памяти, внешней памяти программ и внешней памяти данных исследуемого микроконтроллера) и выводит результаты работы на экран персонального компьютера.
- **При выполнении циклической программы и останове программой управления данные из исследуемого микроконтроллера не считываются.**

### 3 РАБОТА В СИСТЕМЕ ОТЛАДКИ MCS-51-IDE-XP

#### 3.1 Запуск программы

При запуске программы MCS-51-IDE-XP перед пользователем открывается главное окно программы (рис. 3.1). В верхней части располагается стандартное меню, включающее пункты «File», «Program», «Memory», «About». Основную часть окна занимает поле для редактирования кода пользовательской программы. В нижней части окна расположен статус-бар для получения вывода информационных сообщений.

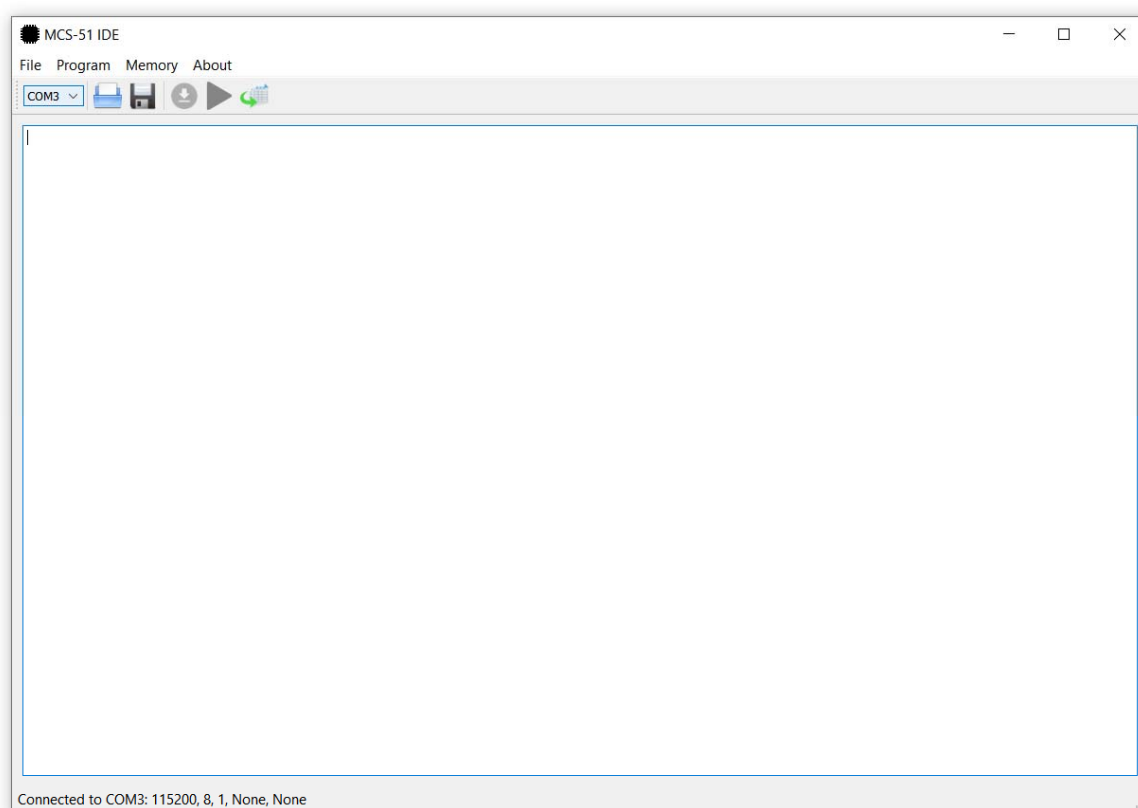


Рисунок 3.1 – Главное окно программы MCS-51-IDE-XP

#### 3.2 Подключение лабораторного макета к компьютеру

Для взаимодействия с УУ программе MCS-51-IDE-XP необходимо иметь подключение к COM-порту, к которому подключена система программирования MCS-51 (макет). При запуске программа автоматически подключается к порту, выбранному при последнем сеансе работы с программой. Если требуется изменить порт, это можно сделать с помощью выпадающего меню в верхней части главного окна программы (рис. 3.2).

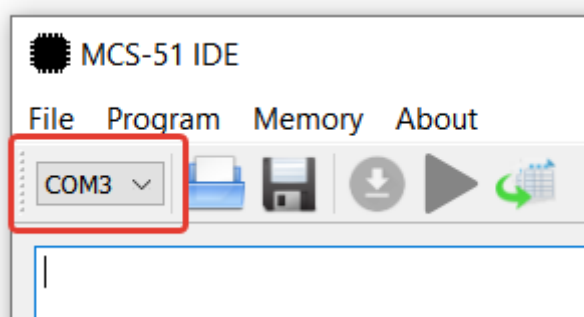


Рисунок 3.2 – Выбор порта COM-порту

После выбора порта переподключение происходит автоматически. При успешном подключении в статус-бар выводится соответствующее сообщение, в противном случае выводится сообщение об ошибке.

### 3.3 Работа с файлами

Программа работает с исходными файлами с расширением \*.A51. Это стандартное расширение для файлов, содержащих код программы на языке ассемблера для микроконтроллеров семейства MCS-51.

После запуска программы пользователь может написать новую программу в поле для редактирования кода, затем сохранить изменения в файле. Или открыть существующий файл при помощи меню File. Меню File содержит следующие пункт для удобной работы с файлами:

- File > New (Ctrl+N) – создание нового файла;
- File > Open (Ctrl+O) – открыть существующий файл;
- File > Save (Ctrl + S) – сохранить изменения в файле;
- File > Save As (Ctrl + Shift + S) – сохранить новый файл.

### 3.4 Компиляция

Для компиляции исполняемого файла необходимо в меню выбрать Program > Build (Ctrl+B). После чего программа предложит сохранить код программы в файле, если это не было сделано заранее, или автоматически сохранит уже открытый в программе файл.

Если компиляция прошла успешно, в статус-бар главного окна программы будет выведено сообщение об успешной компиляции. Если на этапе компиляции возникли ошибки, пользователю откроется окно с содержанием листингового файла \*.LST, в котором можно ознакомиться с полученными ошибками. Пример такого окна приведён на рис. 3.3.

```

C:/Dev/Study/pr.LST
05/28/2020 04:36:05 PAGE 1

A51 MACRO ASSEMBLER _DEV_STUDY_PR

MACRO ASSEMBLER A51 V8.2.7.0
NO OBJECT MODULE REQUESTED
ASSEMBLER INVOKED BY: C:\Dev\Study\build-FileSender-Desktop_Qt_5_14_2_MinGW_64_bit-Debug\debug\Compiler\A51.exe C:/Dev/
S
        tudy/pr.a51

LOC OBJ      LINE  SOURCE
0000 758911    1  Mov TMOD , #00010001b
0003 758C00    2  Mov TH0,#0
0006 758A00    3  Mov TL0,#0
                4  af
***
*** ERROR #A9 IN 4 (C:/Dev/Study/pr.a51, LINE 4): SYNTAX ERROR
                5
0009 2091FD    6  M0: jb P1.1, M0
                7
000C 3091FD    8  M1: jnb P1.1, M1
                9
000F D28C     10  Setb TR0
                11
0011 2091FD    12  M2: jb P1.1, M2
                13
0014 3091FD    14  M3: jnb P1.1,M3
                15
0017 C28C     16  Clr TR0
0019 A88A     17  Mov R0, TL0
001B A98C     18  Mov R1, TH0
                19  End
A51 MACRO ASSEMBLER _DEV_STUDY_PR
05/28/2020 04:36:05 PAGE 2

SYMBOL TABLE LISTING

```

Рисунок 3.3 – Окно программы с файлом .LST

Содержимое файла \*.LST можно в любой момент посмотреть с помощью команды меню Program > Open \*.LST File.

Если компиляция прошла успешно, пользователь получает возможность загрузить программу в ПП лабораторного макета для запуска, а также выбрать точку останова для осуществления отладки.

### 3.5 Загрузка и запуск программы

Для этого необходимо в меню выбрать Program > Run или нажать Ctrl+R. Программа проведёт компиляцию. В случае успеха создаётся файл \*.BIN с образом ПП, в который предварительно будет добавлена программа монитора и инструкции перехода к нему. Программа монитор необходима для считывания внутренних ресурсов исследуемого микроконтроллера для последующего вывода информации на экран. Затем содержимое файла отправляется через COM-порт в УУ, которое запишет образ и даст MCS-51 команду для начала выполнения программы.

Если в процессе создания образа ПП или при отправке данных возникает ошибка, пользователь получит о ней сведения в статус-баре. Также на экране появляется окно с уведомлением о возникшей ошибке (рис. 3.4).

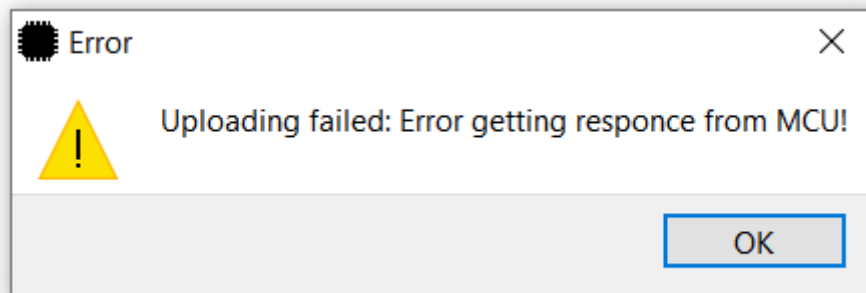


Рисунок 3.4 - Пример ошибки (Ошибка при получении ответа от УУ)

### 3.6 Отладка

Перед началом процедуры отладки можно выбрать в редакторе кода точку останова программы. Для этого необходимо на нужной строке кода нажать клавишу F9 или в меню выбрать Program > Toggle Breakpoint. Выбранная точка останова будет выделена жёлтым цветом (рис. 3.5). Выбор точки останова доступен только в случае успешной компиляции программы.

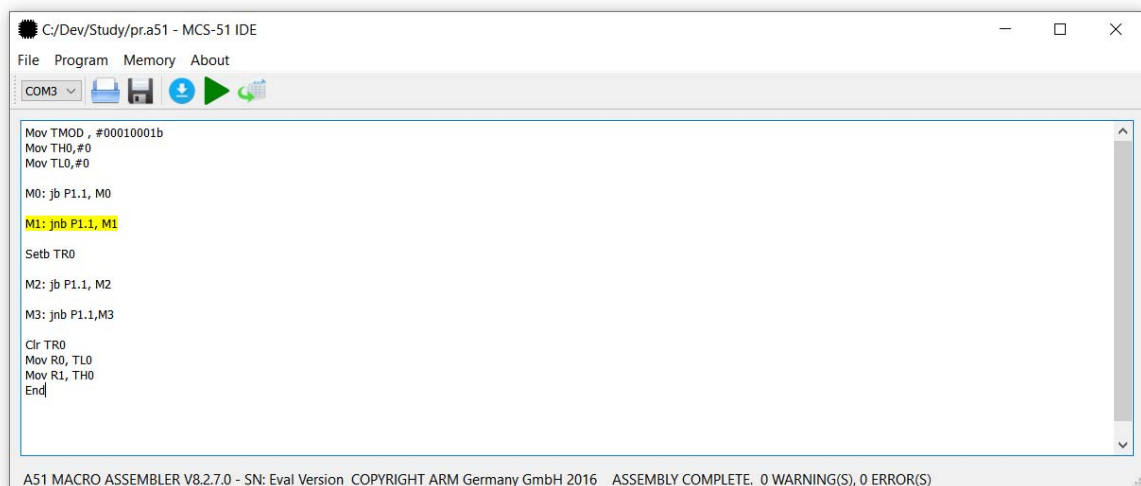


Рисунок 3.5 - Точка останова программы

Точку останова можно менять так же: нажимая клавишу F9 в нужном месте. Если точка останова не нужна, необходимо повторно нажать F9 на выделенной строке – точка останова будет удалена.

По умолчанию точкой останова является адрес за последней командой программы. Поэтому при циклической программе точка останова не работает.

При выбранной в коде программы точке останова в меню становится доступен пункт Program > Debug (F5). При выборе этого пункта программа созда-

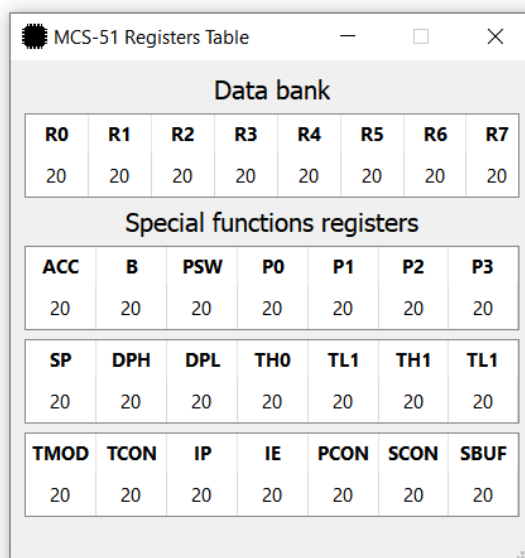
ет образ ПП с инструкцией перехода к монитору в месте выбранной точки останова и отправляет его [образ] в УУ для последующей записи и запуска выполнения программы микроконтроллером MCS-51. О возникающих в процессе ошибках пользователь будет уведомлён через статус-бар и уведомления.

### 3.7 Остановка программы

Если в коде программы пользователя имеется бесконечный цикл, выполнение программы можно остановить в меню Program > Stop Program. При использовании точки останова при успешной работе выполнение программы произойдет автоматически.

### 3.8 Вывод на экран данных из памяти

Для ознакомления с результатами выполнения пользовательской программы предусмотрена возможность просмотра состояния регистров МК MCS-51 с помощью команды меню Memory > Show Registers.



The screenshot shows a window titled "MCS-51 Registers Table" with a table of register values. The registers are organized into two main sections: "Data bank" and "Special functions registers".

| Data bank |    |    |    |    |    |    |    |
|-----------|----|----|----|----|----|----|----|
| R0        | R1 | R2 | R3 | R4 | R5 | R6 | R7 |
| 20        | 20 | 20 | 20 | 20 | 20 | 20 | 20 |

| Special functions registers |    |     |    |    |    |    |
|-----------------------------|----|-----|----|----|----|----|
| ACC                         | B  | PSW | P0 | P1 | P2 | P3 |
| 20                          | 20 | 20  | 20 | 20 | 20 | 20 |

| SP | DPH | DPL | TH0 | TL1 | TH1 | TL1 |
|----|-----|-----|-----|-----|-----|-----|
| 20 | 20  | 20  | 20  | 20  | 20  | 20  |

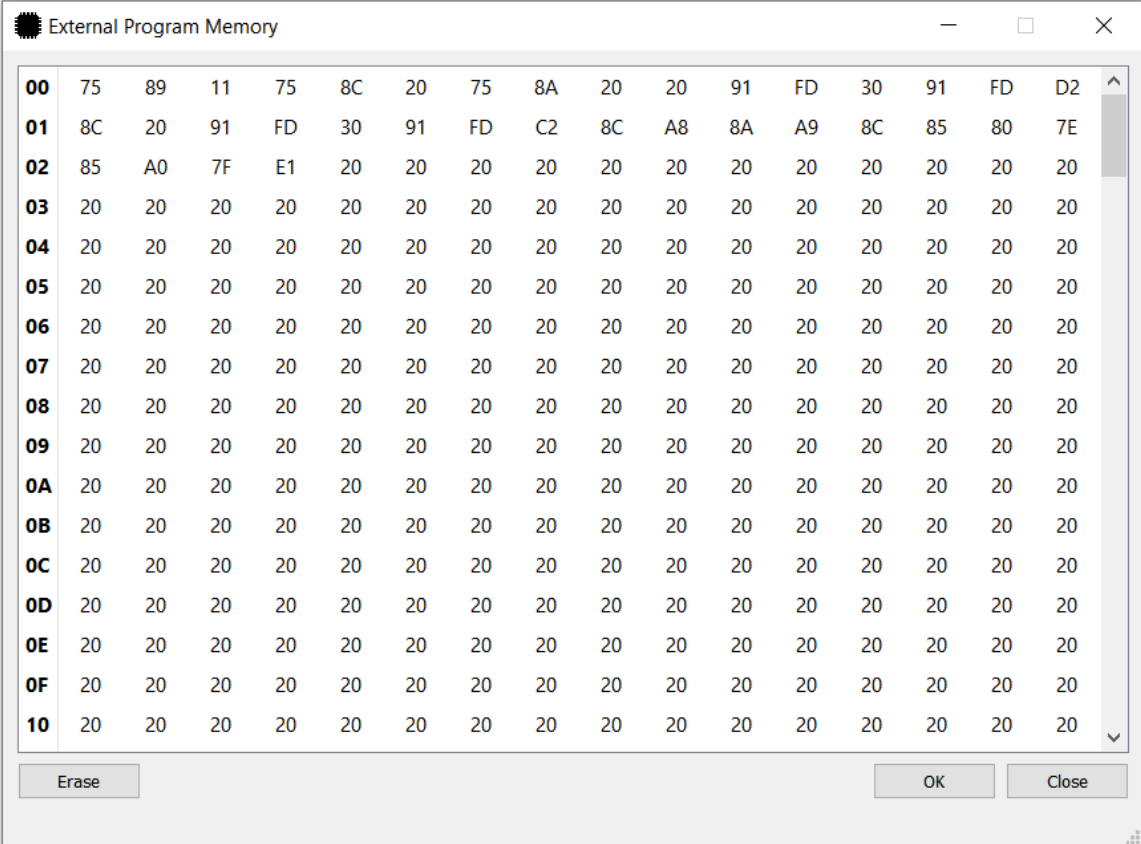
| TMOD | TCON | IP | IE | PCON | SCON | SBUF |
|------|------|----|----|------|------|------|
| 20   | 20   | 20 | 20 | 20   | 20   | 20   |

Рисунок 3.6 - Таблица с содержимым регистров ОЗУ

Для просмотра данных из памяти в меню Memory предусмотрены следующие пункты:

- Memory > Show RAM – вывод резидентной ПД MCS-51;
- Memory > Show Program Memory – вывод данных внешней ПП;
- Memory > Show Data Memory – вывод данных внешней ПД.

При выборе любого из пунктов меню на экране появится таблица с данными из соответствующей памяти (рис. 3.7). Необходимо заметить, что для просмотра данных из внешней ПП, внешней ПД или внутреннего ОЗУ МК MCS-51 выполнение программы будет принудительно остановлено.



| Address | 75 | 89 | 11 | 75 | 8C | 20 | 75 | 8A | 20 | 20 | 91 | FD | 30 | 91 | FD | D2 |
|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00      | 75 | 89 | 11 | 75 | 8C | 20 | 75 | 8A | 20 | 20 | 91 | FD | 30 | 91 | FD | D2 |
| 01      | 8C | 20 | 91 | FD | 30 | 91 | FD | C2 | 8C | A8 | 8A | A9 | 8C | 85 | 80 | 7E |
| 02      | 85 | A0 | 7F | E1 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 03      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 04      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 05      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 06      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 07      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 08      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 09      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 0A      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 0B      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 0C      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 0D      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 0E      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 0F      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 10      | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |

Рисунок 3.7 – Таблица с данными ПД

В нижней части окна просмотра ПП и ПД также расположена кнопка «Erase». При её нажатии в УУ будет отправлена команда стирания соответствующей памяти.

## 4 ТЕСТОВЫЕ ПРОГРАММЫ

### 4.1 Тестовая программа приема и передачи цифрового сигнала

Тестовая программа анализирует входной цифровой сигнал на входе «Input» X4 (контакт P1.0 D3) и повторяет цифровой сигнал на выходе «Output» X5 (контакт P1.3 D3).

*Программа передачи цифровых данных через бит переноса C.*

Org 0

; задание адресов устройств ввода-вывода

Inb equ p1.0 ; вход цифрового сигнала

Outb equ p1.3 ; выход цифрового сигнала

M1: mov C,Inb ; прием входного сигнала в признак C  
 mov Outb,C ; вывод сигнала из признака C  
 jmp M1 ; циклический переход

*Программа передачи цифровых данных с опросом входного сигнала.*

Org 0

; задание адресов устройств ввода-вывода

Inb equ p1.0 ; вход цифрового сигнала

Outb equ p1.3 ; выход цифрового сигнала

M1: jb Inb,M1 ; опрос входного сигнала (ожидание входного сигнала  
 равного 0)  
 clr Outb ; установка выходного сигнала в 0  
 M2: jnb Inb,M2 ; опрос входного сигнала (ожидание входного сигнала  
 равного 1)  
 setb Outb ; установка выходного сигнала в 1  
 jmp M1 ; циклический переход



#### 4.2 Тестовая программа приема и передачи аналогового сигнала

Тестовая программа измеряет входной аналоговый сигнал с помощью АЦП и выдает выходной сигнал на ЦАП без дополнительной обработки.

Входной аналоговый сигнал поступает на разъем «ADC» X6. АЦП (D11) реализован на микросхеме - 1108ПВ1. На входе АЦП установлен операционный усилитель (D9). Первый каскад является буфером для входного сигнала (D9.1). Второй каскад предназначен для смещения входного аналогового сигнала на уровень  $U_{оп}/2$  (D9.2). Напряжение преобразования  $U_{оп}$  в данном макете задано равным 2.5В. Поэтому входное измеряемое напряжение не должно превышать это значение. Входной двуполярный сигнал с уровнем от +1.25В до -1.25В преобразуется на входе АЦП с помощью ОУ в однополярный сигнал размахом от 0 до 2.5В.

Запуск АЦП осуществляется сигналом Start от дешифратора при выполнении соответствующей команды записи во внешнюю память данных. Чтение данных из АЦП осуществляется по сигналу Rdac при выполнении соответствующей команды чтения из внешней памяти данных.

Выходной сигнал формируется на выходе «DAC1» X7. В качестве ЦАП выбрана микросхема AD7528 (D12). На выходах ЦАП включены буферные усилители со смещением выходного сигнала. Канал А ЦАП работает с опорным напряжением 2.5В и формирует сигнал от 0 до 2.5В. Выходной сигнала этого канала смещен на значение  $U_{оп}/2$  (D13.2) для получения двуполярного сигнала от -1.25В до +1.25В.

Запись данных в ЦАП осуществляется по сигналу CSdac от дешифратора при выполнении соответствующей команды записи во внешнюю память данных.

Org 0

; задание адресов устройств ввода-вывода

Outb equ p1.3 ;выход цифрового сигнала

Start equ 3000h ;адрес ADC

Csdac equ 2000h ;адрес DAC

Adc\_dac:

setb Outb ;формирование цифрового синхросигнала

clr Outb

mov dptr, #Start ;адрес ADC

movx @dptr,a ;запуск ADC

movx a, @dptr ;чтение данных с ADC

mov dptr, #Csdac ;адрес DAC

movx @dptr,a ;запись данных в DAC

jmp adc\_dac ;циклический переход

end

## СПИСОК ЛИТЕРАТУРЫ

1. Структура микропроцессорной системы: Курс лекций. [Электронный ресурс] / Гурьянов С.А.; ФГБОУ «Новгородский государственный университет им. Ярослава Мудрого», Великий Новгород, 2012 г. –51с.

Режим доступа - <https://novsu.bibliotech.ru/Reader/BookPreview/-243>

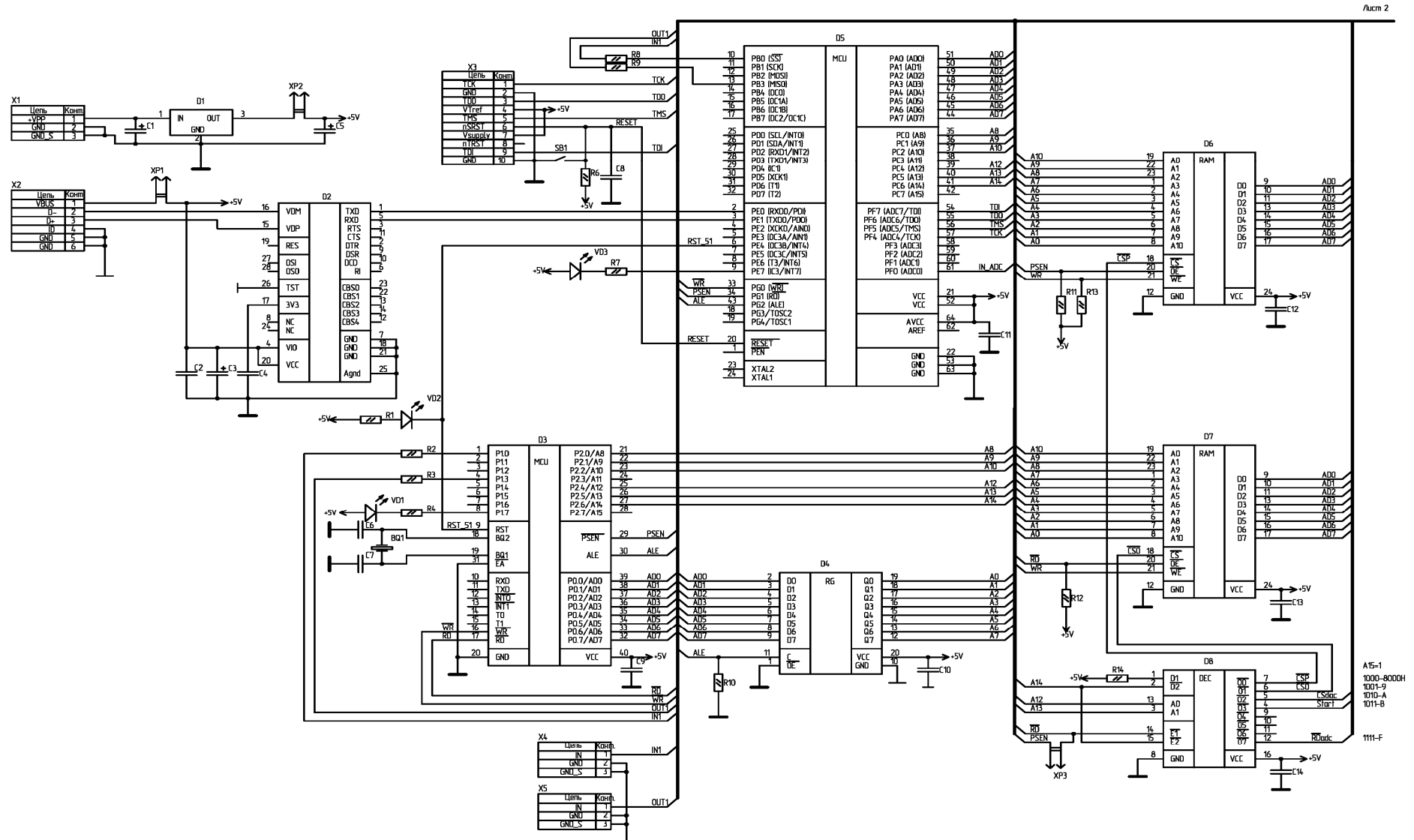
2. Микроконтроллер INTEL 8051. Конспект лекций. [Электронный ресурс] / Гурьянов С.А.; НовГУ. - В. Новгород, 2012. – 92с.

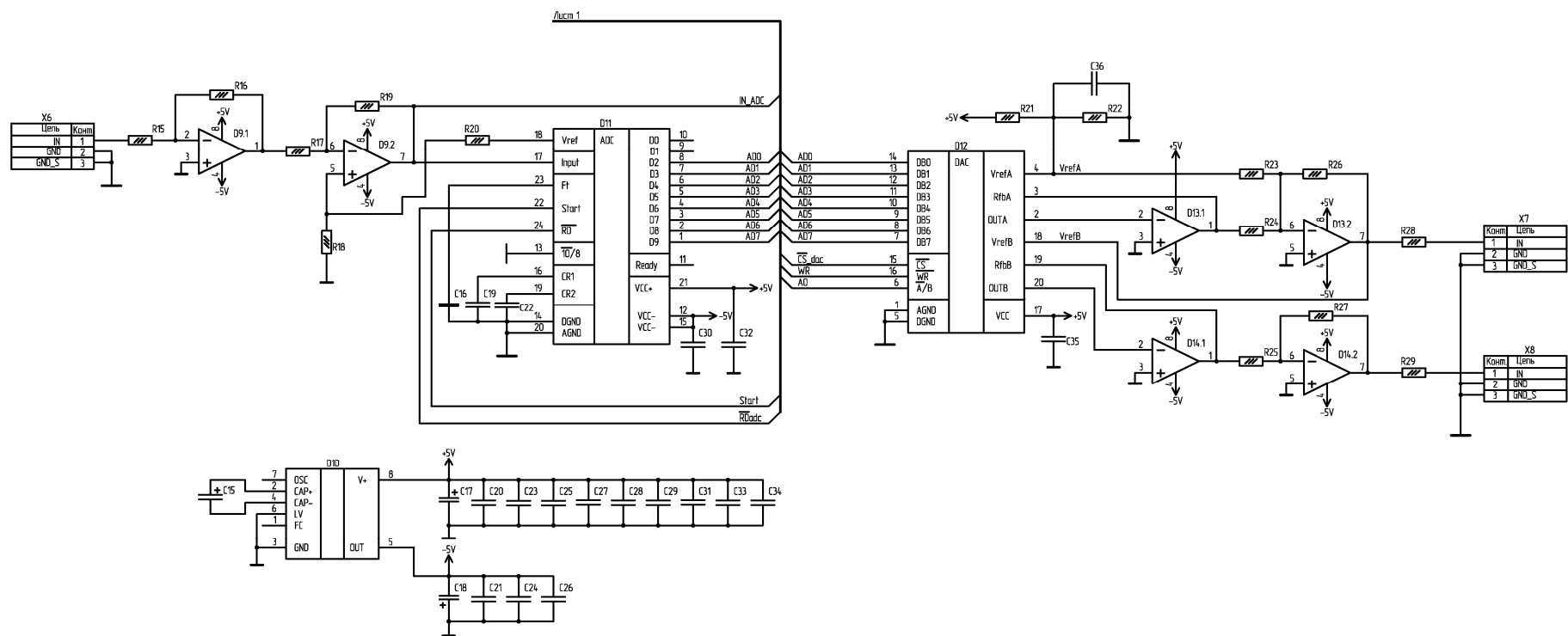
Режим доступа - <https://novsu.bibliotech.ru/Reader/BookPreview/-221>

3. Микроконтроллеры на базе архитектуры 8051.

[/http://www.atmel.com/ru/ru/products/microcontrollers/8051architecture/default.aspx](http://www.atmel.com/ru/ru/products/microcontrollers/8051architecture/default.aspx)

## Схема электрическая принципиальная лабораторного макета





Приложение Б  
Система команд микроконтроллера MCS51

Таблица Б1

Таблица обозначений и символов, используемых в системе команд

| Обозначение | Назначение   |
|-------------|--|
| A           | Аккумулятор  |
| B           | Регистр-расширитель аккумулятора   |
| Rn          | Регистры текущего банка регистров (n=0...7)  |
| Adr         | Прямо адресуемый 8-Битовый адрес ячейки данных внутреннего ОЗУ (0-127) или SFR (128-255) |
| @Ri         | Косвенно адресуемая 8-Битовая ячейка внутреннего ОЗУ данных (i=0,1)                      |
| #data8      | 8-Битовое непосредственное данные  |
| #data16     | 16-Битовое непосредственное данные   |
| Bit         | Бит с прямой адресацией, адрес которого находится во внутреннем ОЗУ данных или SFR       |
| РПД         | Резидентная (внутренняя) память данных   |
| ВПД         | Внешняя память данных  |
| ПП          | Память программ  |

Таблица команд микроконтроллера MCS51

| Мнемоника                                  | Функция   | Опе-<br>ранд                    | Фла-<br>ги |
|--|---|---------------------------------|------------|
| Команды передачи данных                    |   |                                 |            |
| MOV A, <источник>                          | Переслать байт из источника в A                 | Rn<br>Adr<br>@Ri<br>#data8      |            |
| MOV Rn, <источник>                         | Переслать байт из источника в Rn                | Adr<br>#data8                   |            |
| MOV Adr, <источник>                        | Переслать байт из источника в РПД по адресу Adr | A<br>Rn<br>Adr<br>@Ri<br>#data8 |            |
| MOV @Ri, <источник>                        | Переслать байт из источника в РПД по адресу @Ri | A<br>Adr<br>#data8              |            |
| MOV DPTR, #data16                          | Загрузить 16-Битовые данные в DPTR              |                                 |            |
| Команды передачи данных из памяти программ |   |                                 |            |
| MOVC A, @A+DPTR                            | Переслать байт из ПП по адресу A+DPTR в A       |                                 |            |
| MOVC A, @A+PC                              | Переслать байт из ПП по адресу A+PC в A         |                                 |            |
| Команды передачи данных из внешней памяти  |   |                                 |            |
| MOVX A, @Ri                                | Переслать байт из ВПД по адресу @Ri в A         |                                 |            |
| MOVX A, @DPTR                              | Переслать байт из ВПД по адресу @DPTR в A       |                                 |            |
| MOVX @Ri, A                                | Переслать байт из A в ВПД по адресу @Ri         |                                 |            |
| MOVX @DPTR, A                              | Переслать байт из A в ВПД по адресу @DPTR       |                                 |            |
| Команды работы со стеком                   |   |                                 |            |
| POP <Adr>                                  | Чтение из стека                                 |                                 |            |
| PUSH <Adr>                                 | Запись в стек                                   |                                 |            |
| Команды обмена данными                     |   |                                 |            |
| SWAP A                                     | Обмен тетрад внутри аккумулятора                |                                 |            |
| XCH A, <операнд>                           | Обмен содержимого аккумулятора с операндом      | Rn<br>Adr<br>@Ri                |            |

|                        |   |                            |               |
|------------------------|---|----------------------------|---------------|
| XCHD A, @Ri            | Обмен тетрад A и РПД с адресом @Ri          |                            |               |
| Арифметические команды |   |                            |               |
| ADD A, <операнд>       | Сложение A и операнда                       | Rn<br>Adr<br>@Ri<br>#data8 | AC<br>C<br>OV |
| ADDC A, <операнд>      | Сложение A и операнда с переносом           | Rn<br>Adr<br>@Ri<br>#data8 | AC<br>C<br>OV |
| DEC <операнд>          | Декремент операнда                          | A<br>Rn<br>Adr<br>@Ri      |               |
| DIV AB                 | Деление A на B                              |                            | C<br>OV       |
| INC <операнд>          | Инкремент операнда                          | A<br>Rn<br>Adr<br>@Ri      |               |
| INC DPTR               | Инкремент указателя данных                  |                            |               |
| DA A                   | Десятичная коррекция аккумулятора           |                            | AC<br>C       |
| MUL AB                 | Умножение A на B                            |                            | C<br>OV       |
| SUBB A, <операнд>      | Вычитание из A операнда с заемом            | Rn<br>Adr<br>@Ri<br>#data8 | AC<br>C<br>OV |
| Логические команды     |   |                            |               |
| ANL A, <операнд>       | Логическое "И" A и операнда                 | Rn<br>Adr<br>@Ri<br>#data8 | C=0           |
| ANL Adr, A             | Логическое "И" РПД с адресом Adr и A        |                            | C=0           |
| ANL Adr, #data8        | Логическое "И" РПД с адресом Adr и #data8   |                            | C=0           |
| CLR A                  | Сброс аккумулятора                          |                            |               |
| CPL A                  | Инверсия аккумулятора                       |                            |               |
| ORL A, <операнд>       | Логическое "ИЛИ" A и операнда               | Rn<br>Adr<br>@Ri<br>#data8 | C=0           |
| ORL Adr, A             | Логическое "ИЛИ" РПД с адресом Adr и A      |                            | C=0           |
| ORL Adr, #data8        | Логическое "ИЛИ" РПД с адресом Adr и #data8 |                            | C=0           |



|   |   |                            |     |
|---|---|----------------------------|-----|
| RL A                                      | Сдвиг содержимого аккумулятора влево                                |                            | C   |
| RLC A                                     | Сдвиг содержимого аккумулятора влево через флаг переноса C          |                            | C   |
| RR A                                      | Сдвиг содержимого аккумулятора вправо                               |                            | C   |
| RRC A                                     | Сдвиг содержимого аккумулятора вправо через флаг переноса C         |                            | C   |
| XRL A, <операнд>                          | Логическое "ИСКЛЮЧАЮЩЕЕ ИЛИ" A и операнда                           | Rn<br>Adr<br>@Ri<br>#data8 | C=0 |
| XRL Adr,A                                 | Логическое " ИСКЛЮЧАЮЩЕЕ ИЛИ" РПД с адресом Adr и A                 |                            | C=0 |
| XRL Adr,#data8                            | Логическое " ИСКЛЮЧАЮЩЕЕ ИЛИ" РПД с адресом Adr и #data8            |                            | C=0 |
| Команды безусловных переходов             |   |                            |     |
| AJMP <Метка>                              | Абсолютный переход на Метку   |                            |     |
| LJMP <Метка>                              | Длинный переход на Метку  |                            |     |
| SJMP <Метка>                              | Короткий переход на Метку   |                            |     |
| JMP @A+DPTR                               | Косвенный переход по адресу @A+DPTR                                 |                            |     |
| Команды условных переходов                |   |                            |     |
| CJNE A, <операнд>, <Метка>                | Сравнение двух операндов и переход на Метку, если они не равны      | Adr<br>#data8              | C   |
| CJNE Rn, #data8, <Метка>                  | Сравнение Rn и #data8 и переход на Метку, если они не равны         |                            | C   |
| CJNE @Ri, #data8, <Метка>                 | Сравнение @Ri и #data8 и переход на Метку, если они не равны        |                            | C   |
| DJNZ <операнд>, <Метка>                   | Декремент операнда и переход на Метку, если результат не равен нулю | Rn<br>Adr                  |     |
| JC <Метка>                                | Переход на Метку, если признак C установлен                         |                            | C   |
| JNC <Метка>                               | Переход на Метку, если признак C не установлен                      |                            | C   |
| JNZ <Метка>                               | Переход на Метку, если содержимое аккумулятора не равно нулю        |                            |     |
| JZ <Метка>                                | Переход на Метку, если содержимое аккумулятора равно нулю           |                            |     |
| Команды вызовов и возврата из подпрограмм |   |                            |     |
| ACALL <Метка>                             | Абсолютный вызов подпрограммы                                       |                            |     |
| LCALL <Метка>                             | Длинный вызов подпрограммы  |                            |     |
| RET                                       | Возврат из подпрограммы   |                            |     |
| RETI                                      | Возврат из прерывания   |                            |     |

| Команды работы с Битами    |  |  |   |
|----------------------------|--|--|---|
| ANL C, <Bit>               | Логическое "И" признака C и Бита                         |  | C |
| CLR <Bit>                  | Сброс Бита   |  |   |
| CPL <Bit>                  | Инверсия Бита  |  |   |
| MOV C, <Bit>               | Переслать Бит в признак C                                |  | C |
| MOV <Bit>, C               | Переслать признак C в Бит                                |  | C |
| ORL C, < Bit >             | Логическое "ИЛИ" признака C и Бита                       |  | C |
| ORL < Bit >, C             | Логическое "ИЛИ" признака C и Бита                       |  | C |
| SETB C                     | Установить Бит   |  | C |
| SETB <Bit>                 | Установить признак C                                     |  | C |
| Условные переходы с Битами |  |  |   |
| JB <Bit>, <Метка>          | Переход на Метку, если Бит установлен                    |  |   |
| JBC <Bit>, <Метка>         | Переход на Метку, если Бит установлен и сброс этого Бита |  |   |
| JNB <Bit>, <Метка>         | Переход на Метку, если Бит не установлен                 |  |   |