

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ФАКУЛЬТЕТ РАДИОФИЗИКИ И ЭЛЕКТРОНИКИ  
Кафедра кибернетики

---

# МИКРОКОНТРОЛЛЕРЫ В СИСТЕМАХ УПРАВЛЕНИЯ И ИЗМЕРЕНИЯ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
К ЛАБОРАТОРНОМУ ПРАКТИКУМУ

Для студентов специальностей  
С 1-31 04 02 «Радиофизика»,  
С 1-31 04 03 «Физическая электроника»,  
специализации Е 1-25 01 10 08  
«Коммерческая инженерия» по разделу  
радиоэлектронных средств и информационных услуг»

В ДВУХ ЧАСТЯХ

Часть 2

---

МИНСК  
2004

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ФАКУЛЬТЕТ РАДИОФИЗИКИ И ЭЛЕКТРОНИКИ  
Кафедра кибернетики**

# **МИКРОКОНТРОЛЛЕРЫ В СИСТЕМАХ УПРАВЛЕНИЯ И ИЗМЕРЕНИЯ**

**Методические указания  
к лабораторному практикуму**

**В двух частях  
Часть 2**

**Для студентов специальностей:  
G 31 04 02 «Радиофизика»,  
G 31 04 03 «Физическая электроника»,  
E 25 01 10 «Коммерческая деятельность на рынке  
радиоэлектронных средств и информационных услуг»**

**МИНСК  
2004**

УДК 621.37(075.83)  
ББК 32.844.1р.я73  
М59

Автор - составитель  
**И. А. Шалатонин**

Рецензент  
доктор физико-математических наук,  
профессор *В. В. Ананасович*

Утверждено Ученым советом  
факультета радиофизики и электроники  
27 мая 2003 г., протокол № 11

**Микроконтроллеры** в системах управления и измерения:  
М59 Метод. указания к лабораторному практикуму: Для студентов специальностей G 1-31 04 02 «Радиофизика», G 1-31 04 03 «Физическая электроника», специализации E 1-25 01 10 08 «Коммерческая деятельность на рынке радиоэлектронных средств и информационных услуг»: В 2 ч. Ч. 2 / Авт.-сост. И. А. Шалатонин. – Мн.: БГУ, 2004. – 99 с.

Методические указания предназначены для студентов III–V курсов факультета радиофизики и электроники.

УДК 621.37(075.83)  
ББК 32.844.1р.я73

© БГУ, 2004

## СОДЕРЖАНИЕ

<b>1. Микроконтроллеры семейства MCS-51</b> .....	5
1.1. Базовая архитектура семейства .....	6
1.1.1. Стандартные порты ввода/вывода .....	10
1.1.2. Стандартные таймеры .....	12
1.1.3. Стандартный канал последовательного ввода/вывода .....	12
1.1.4. Стандартная система прерываний .....	13
1.1.5. Режимы работы .....	15
1.1.6. Система команд .....	15
1.1.7. Типы операндов .....	16
1.1.8. Способы адресации данных .....	16
1.1.9. Флаги результата .....	18
1.1.10. Символическая адресация .....	19
1.1.11. Адресация внешней памяти .....	19
1.2. Развитие в микроконтроллерах фирмы INTEL базовой архитектуры MCS-51 .....	20
1.3. Микроконтроллеры фирмы ATMEL .....	25
1.4. Микроконтроллеры фирмы DALLAS Semiconductor .....	27
1.5. Микроконтроллеры и средства разработки компании Silicon Labs (Cygnal) .....	30
1.6. Микроконтроллеры фирмы PHILIPS .....	34
1.7. Микроконтроллеры фирмы SIEMENS .....	37
<b>2. Лабораторные работы</b> .....	39
2.1. Лабораторная работа № 1. Программирование микро контроллера Intel8051 .....	39
2.1.1. Лабораторный стенд на базе микроконтроллера Intel8051 .....	39
2.1.2. Порядок выполнения лабораторной работы .....	42
2.2. Лабораторная работа № 2. Построение однопроводной сети 1 WIRE и организация обмена информации с iBUTTON DS199х с использованием МК DS5000T .....	46
2.2.1. Лабораторный стенд для изучения МПС на базе МК DS 5000T .....	46

2.2.2. Программное обеспечение для работы со стендом.....	46
2.2.3. Модуль для изучения Touch Memory DS1992 .....	49
2.2.4. Модуль для изучения адресуемых переключателей DS2405 .....	51
2.2.5. Работа со стендом для изучения микроконтроллеров Dallas Semiconductor .....	52
2.2.6. Порядок выполнения лабораторной работы .....	54
2.3. Лабораторная работа № 3. Программирование стандартных интерфейсов микроконтроллера на примере 8X552 фирмы PHILIPS .....	54
2.3.1. Краткие сведения из теории .....	55
2.3.2. Описание лабораторного стенда .....	61
2.3.3. Порядок выполнения лабораторной работы .....	67
2.4. Лабораторная работа № 4. Работа отладочной платы EV 552 с МК Atmel 89C51 .....	71
2.5. Лабораторная работа № 5. Работа отладочной платы C8051F300 с МК Cygnal F300.....	72
<b>Приложение 1</b> .....	<b>79</b>
<b>Приложение 2</b> .....	<b>82</b>
<b>Приложение 3</b> .....	<b>83</b>
<b>Приложение 4</b> .....	<b>84</b>
<b>Приложение 5</b> .....	<b>85</b>
<b>Приложение 6</b> .....	<b>94</b>

## 1. МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА MCS-51

Несмотря на непрерывное развитие и появление все новых и новых 16- и 32-разрядных микроконтроллеров, наибольшая доля мирового микропроцессорного рынка и по сей день остается за 8-разрядными устройствами. Среди всех 8-разрядных микроконтроллеров – семейство MCS-51 является несомненным чемпионом по количеству разновидностей и количеству компаний, выпускающих его модификации. Важную роль в достижении такой высокой популярности семейства 8051 сыграла открытая политика фирмы Intel, родоначальницы архитектуры, направленная на широкое распространение лицензий на ядро 8051 среди большого количества ведущих полупроводниковых компаний мира.

Основными производителями клонов 51-го семейства в мире являются фирмы Philips, Siemens, Intel, Atmel, Dallas, Temic, Oki, AMD, MHS, Gold Star, Winbond, Silicon Systems и ряд других. В рамках СССР производство микроконтроллера 8051 осуществлялось в Киеве, Воронеже (1816BE31/51, 1830BE31/51), Минске (1834BE31) и Новосибирске (1850BE31). Но в целом это направление в СССР и затем в СНГ не вышло за рамки базового варианта (8051).

Лидирующие позиции семейства MCS51 на рынке микроконтроллеров обусловлено целым рядом причин.

- Во-первых, в семействе x51 реализована очень удачная архитектура. Это способствовало тому, что семейство x51-совместимых микроконтроллеров «de facto» стало всемирным промышленным стандартом и широко распространено во всем мире.

- Во-вторых, x51-совместимые микроконтроллеры очень хорошо документированы. Как по аппаратной реализации этого семейства, так и по их применению и программированию опубликовано достаточно много разнообразной научно-технической литературы – книг, статей, рекомендаций по применению.

- В-третьих, за время существования этого семейства для него было разработано очень большое количество качественного и доступного программного обеспечения: компиляторов различных языков программирования: ASM51, C++, PL/M51, Fort51 и т. п., дизассемблеров, программных отладчиков, эмуляторов и т. п. Кроме того, существует большое количество программ и библиотек для различных научно-технических задач.

- В-четвертых, у большинства разработчиков, занимающихся микроконтроллерной техникой, имеется достаточно большой опыт общения с этим семейством микроконтроллеров, изучены многие особенности их поведения, выработаны приемы и способы отладки

В результате на сегодняшний день существует более 200 модификаций микроконтроллеров семейства 8051, выпускаемых почти 20 компаниями. Эти модификации включают в себя кристаллы с широчайшим спектром периферии: от простых 20-выводных устройств с одним таймером и 1К программной памяти до сложнейших 100-выводных кристаллов с 10-разрядными АЦП, массивами таймеров-счетчиков, аппаратными 16-разрядными умножителями и 64К программной памяти на кристалле. Каждый год появляются все новые варианты представителей этого семейства. Основными направлениями развития являются: увеличение быстродействия (повышение тактовой частоты и переработка архитектуры), снижение напряжения питания и потребления, увеличение объема ОЗУ и Flash-памяти на кристалле с возможностью внутрисхемного программирования, введение в состав периферии микроконтроллера сложных устройств типа системы управления приводами CAN и USB интерфейсов и т. п.

## **1.1. Базовая архитектура семейства**

Фирма Intel является родоначальницей архитектуры MCS-51, которая получила свое название от первого представителя этого семейства – микроконтроллера 8051, выпущенного в 1980 году на базе технологии NMOS. С точки зрения технологии микроконтроллер 8051 являлся для своего времени очень сложным изделием – в кристалле было использовано 128 тыс. транзисторов, что в 4 раза превышало количество транзисторов в 16-разрядном микропроцессоре 8086. Структурная схема MCS8051 приведена на рис. 1.

Основными элементами базовой архитектуры являются:

- 8-разрядное АЛУ на основе аккумуляторной архитектуры;
- 4 банка регистров, по 8 в каждом;
- встроенная память программ 4 Кбайт;
- внутреннее ОЗУ 128 байт;
- булевый процессор;
- 2 шестнадцатиразрядных таймера;

- контроллер последовательного канала (UART);
- контроллер обработки прерываний с двумя уровнями приоритетов;
- четыре 8-разрядных порта ввода/вывода, два из которых используются в качестве шины адреса/данных для доступа к внешней памяти программ и данных;
- встроенный тактовый генератор.

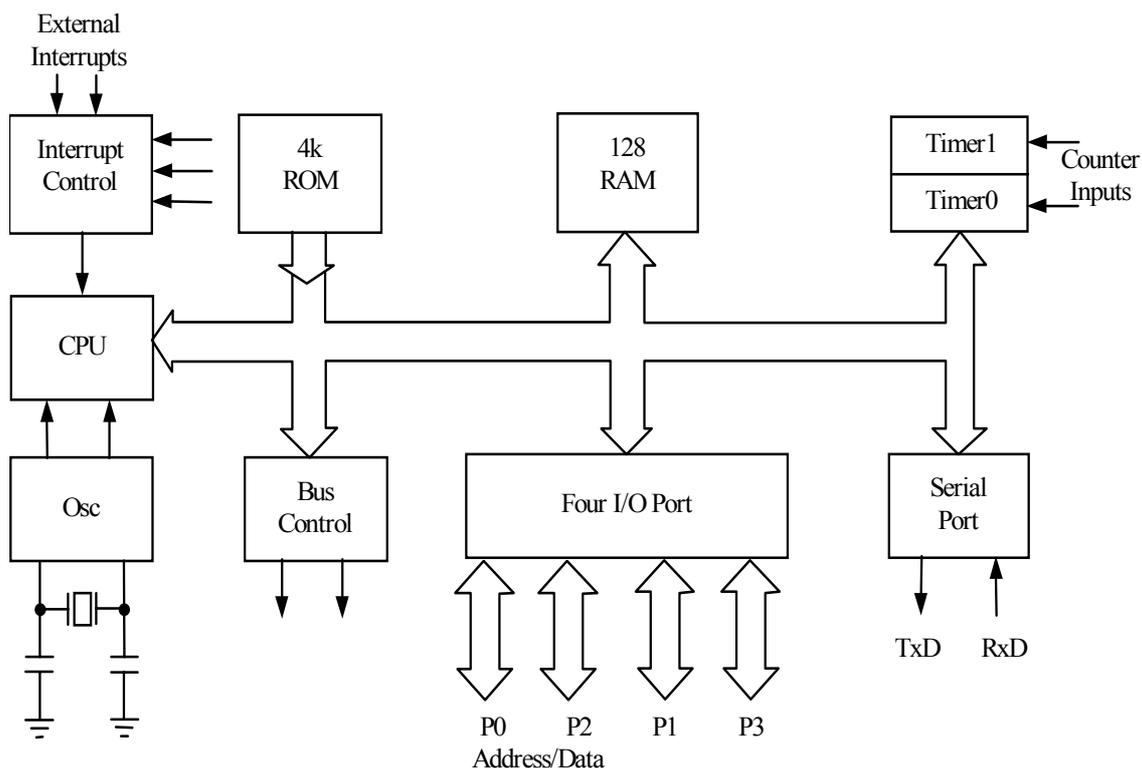


Рис. 1. Структурная схема микроконтроллера 8051

В микроконтроллерах семейства MCS8051, как машины с гарвардской архитектурой, память программ и данных имеет отдельные шины для передачи данных и адреса. Это позволяет сократить время выполнения команд, совместив процесс выборки команд и данных.

Архитектура МК 8051 ограничивает размер памяти программ 64 К. Для ее адресации используется 16-разрядная шина адреса. Память может быть прочитана, но не записана. Для управления считыванием используется сигнал PSEN (Program Store Enable). Имеются дополнительные 64 К памяти, которые могут быть считаны и записаны при помощи сигналов RD и WR. Обращение к этой памяти

может проводиться только косвенным образом через специальный регистр DPTR. Поскольку эта память достаточно быстрая, влияние особенностей гарвардской архитектуры на производительность заметно не сказывается. В связи с этим для обращения к этой памяти используются те же шины, что и для программной памяти. Описанная память носит название внешней памяти данных (External Data Memory) и располагается, как правило, вне кристалла.

Встроенная (Internal) память имеет внутренние (недоступные извне) отдельные 8-разрядные шины адреса и данных и сигналы управления. Фактически можно адресовать только 256 байтов памяти, но архитектурой предусмотрены 384 адресуемые ячейки (рис. 2).

Если в команде задана непосредственная адресация, то происходит обращение к одним 256 ячейкам, если задана косвенная адресация, то к другим 256 ячейкам. 128 младших ячеек при прямой и косвенной адресации совпадают. Старшие 128 ячеек, адресуемые прямым способом, являются регистрами встроенных устройств и носят название регистров специального назначения (Special Function Register SFR), для MCS51 существуют команды, позволяющие оперировать с битами внутренней памяти. 128 таких битов совпадают с битами шестнадцати байтов младшей половины внутренней памяти, и 128 битов совпадают с битами шестнадцати ячеек регистров SFR. Всего же может быть адресовано 256 битов.

Кроме SFR, для MCS51 выделено два типа регистров. Восемь PОН совпадают с младшими ячейками внутренней памяти. Два новых регистра, R0 и R1, могут быть также в памяти, под них отведено 32 байта, которые разбиты на четыре банка. Номер банка задается двумя битами 8-разрядного регистра состояния PSW. Такая конструкция PОН обеспечивает множественность способов обращения к ним. Специальные регистры относятся ко второму типу. 8-разрядный аккумулятор (A) применяется во всех арифметических операциях. Для операции умножения A работает в паре с 8-разрядным регистром B, а арифметические флаги содержатся в регистре состояния PSW. Указатель стека SP имеет 8 разрядов и используется для косвенной адресации вершины стека во внутренней памяти данных. Имеются также два 16-разрядных регистра – это счетчик программ PC и регистр косвенных ссылок DPTR.

Все перечисленные регистры являются неотъемлемой частью системы команд процессора и учтены в мнемонике языка ассемблера. К этим же регистрам имеется доступ через ячейки SFR, который

обеспечивается прямой адресацией к верхней половине внутренней памяти данных. В языке ассемблера для регистров в пространстве SFR зарезервированы мнемонические обозначения B, PSW, SP, ACC, DPL и DPH (для DPTR).



Рис. 2. Особенности структуры памяти микроконтроллеров с гарвардской архитектурой

Эти регистры кратко описаны ниже.

- Аккумулятор (A) используется как источник или приемник данных при арифметических операциях. Также может использоваться в некоторых других типах инструкций.

- Указатель стека (SP) – 8-битовый регистр, используемый для хранения позиции последнего байта данных, содержащихся в стеке. Стек может быть расположен в любом месте внутренней 128-байтовой сверхоперативной регистровой памяти. Указатель стека инкрементируется перед операцией PUSH и декрементируется после операции POP.

- Регистр B – основная функция – хранение исходных данных и результата при операциях умножения и деления.

- Слово состояния программы (PSW) содержит флаги, соответ-

ствующие результатам выполнения предыдущей операции, а также биты выбора банка рабочих регистров.

- Указатель данных (DPTR) используется для доступа к памяти данных, которая может быть расположена либо на шине памяти, либо на шине расширения. Программой пользователя может обрабатываться как два 8-битных регистра специальных функций или как 16-битный регистр в некоторых инструкциях

Регистры специальных функций с указанием их адресов приведены в табл. 1.

ТАБЛИЦА 1

**Регистры специальных функций**

Обозначение	Наименование	Адрес
* ACC	Аккумулятор	0E0H
* B	Регистр B	0F0H
* PSW	Регистр состояния программы	0D0H
SP	Указатель стека	81H
DPTR	Указатель данных (2 байта):	
DPL	Младший байт	82H
DPH	Старший байт	83H
* P0	Порт 0	80H
* P1	Порт 1	90H
* P2	Порт 2	0A0H
* P3	Порт 3	0B0H
* IP	Регистр приоритетов прерываний	0B8H
* IE	Регистр разрешения прерываний	0A8H
TMOD	Регистр режимов таймера/счетчика	89H
* ICON	Регистр управления таймера/счетчика	88H
TH0	Таймер/счетчик 0. Старший байт	8CH
TL0	Таймер/счетчик 0. Младший байт	8AH
TH1	Таймер/счетчик 1. Старший байт	8DH
TL1	Таймер/счетчик 1. Младший байт	8BH
* SCON	Управление последовательным портом	98H
SBUF	Буфер последовательного порта	99H
PCON	Управление потреблением	87H

\* – регистры, допускающие побитовую адресацию.

**1.1.1. Стандартные порты ввода/вывода**

Стандартный микроконтроллер 8051 реализован в корпусе DIP , имеющем 40 выводов. Кроме управляющих выводов, есть еще 32 вывода, которые отдельно программируются на ввод/вывод. 32 бита организованы в четыре 8-разрядных порта, обозначаемых как P0, P1, P2 и P3. Обращение к портам со стороны процессора происходит через 4 ячейки SFR. При записи в эти ячейки происходит

инициализация регистра защелки порта. При считывании снимается сигнал с физического входа. Со всеми разрядами портов допустимы битовые операции, таким образом можно считать и проинициализировать каждый ввод в отдельности.

Дополнительно порт или его разряд может обеспечивать альтернативную функцию. Для порта 0 это вставление младшего байта адреса и передача данных при обращении к внешней памяти данных или программ. На выводах порта 2 в этом случае появляется старшая половина 16-разрядного адреса.

Следует отметить основные особенности работы портов 0 и 2 в альтернативном режиме:

- для лучшей работы шины при передаче логической единицы в адресе или данных в соответствующем разряде порта 0 включается цепь подтягивания к питающему напряжению, а в порте 2 эта цепь изменяет свои параметры;

- для обеспечения корректного считывания данных в защелку порта 0 процессор автоматически прописывает все единицы;

- защелка порта 2 при передаче адреса не используется, а ее значение сохраняется.

Альтернативные функции портов 1 и 3 могут работать только при наличии в соответствующих разрядах защелок единичных значений. Эти значения устанавливаются при инициализации процессора автоматически, но в дальнейшем могут быть изменены или установлены только из программы. В базовом стандарте MSC51 порт 1 не имеет альтернативных функций, и они будут описаны по мере появления. Альтернативные же функции порта 3 являются стандартными для всех модификаций МК независимо от фирмы-производителя. Несмотря на это, возможны отклонения от описанного:

- P3.7 (RD) – выходной сигнал считывания внешней памяти данных;

- P3.6 (WR) – выходной сигнал записи внешней памяти данных;

- P3.5 (T1) – выходной сигнал для первого таймера;

- P3.4 (T0) – выходной сигнал для второго таймера;

- P3.3 (INT1) – вход прерывания;

- P3.2 (INT0) – вход прерывания;

- P3.1 (TXD) – вывод для передачи последовательных данных;

- P3.0 (RXD) – вывод для приема последовательных данных.

### 1.1.2. Стандартные таймеры

Предусмотрено два идентичных 16-разрядных регистра, которые могут работать в качестве счетчика или таймера. Доступ к каждому из них происходит через независимые пары 8-разрядных ячеек TH0, TLO для таймера 0 и TH1, TL1 для таймера 1. Управление таймерами осуществляется через 8-разрядные регистры TMOD и TCON. Регистр TCON позволяет запретить счет и узнать о возникшем переполнении. Регистр TMOD позволяет задать режим работы.

Выводы T0, T1, INT0 и INT1 порта 3 используются для управления счетчиками. Через T0 и T1 подаются подсчитываемые импульсы, а INT0 и INT1 могут быть использованы для запрещения работы счетчика/таймера.

Счетчики/таймеры функционируют в четырех различных режимах независимо друг от друга:

1. Первый режим обеспечивает 13-разрядный счет. Он нужен для совместимости с микроконтроллерами семейства MCS48, для которых существует только 8-разрядный таймер с 5-разрядным делителем.

2. Второй режим обеспечивает 16-разрядный счет.

3. Третий режим обеспечивает 8-разрядный счет с перезагрузкой при переполнении. TH0 (TH1) является собственно счетчиком, а TH1 (TL1) хранит константу перезагрузки.

4. Четвертый режим останавливает счетчик 1 и разбивает счетчик 0 на два независимых 8-разрядных счетчика. Основной счетчик/таймер располагается в TLO, а дополнительный TH0 может работать только как таймер и использует флаг переполнения и бит запрета счета таймера 1. Таймер 1 в это время работает без фиксации переполнения и может быть остановлен только переводом в четвертый режим.

В режиме таймера приращение счетчиков происходит за один машинный цикл.

### 1.1.3. Стандартный канал последовательного ввода/вывода

Микроконтроллеры семейства MCS51 имеют встроенный контроллер последовательного ввода/вывода, обеспечивающий передачу как в синхронном, так и в асинхронном режимах. Два вывода порта 3 ответственны за передачу и прием. По линии RXD происходит прием данных, а по линии TXD – передача. В качестве задающего генератора может быть использован таймер 1. Для

управления контроллером используется регистр SCON, а данные принимаются и передаются через регистр SBUF.

Максимальную скорость передачи – 1 бит за 12 машинных тактов – обеспечивает синхронный режим. В этом случае вывод TXD используется для выдачи синхроимпульсов, а прием и передача данных происходит на выводе RXD.

В асинхронном режиме предусмотрены дуплексные прием и передача восьми или девяти битов данных. Данным предшествует один стартовый бит и как минимум один завершающий. Минимальное время передачи одного бита 32 такта, что для 12 МГц соответствует скорости передачи 375 килобитов в секунду. Широкий диапазон перестраиваемых частот позволяет использовать все имеющиеся стандартные скорости передачи.

В стандартной архитектуре MCS51 не предусмотрено средств обнаружения ошибочных ситуаций, которые могут возникнуть при приеме.

#### **1.1.4. Стандартная система прерываний**

Стандартная система прерываний предусматривает 5 источников прерываний:

1. Прерывание от внешнего источника на выходе INT0.
2. Прерывание от таймера 0.
3. Прерывание от внешнего источника на выходе INT1.
4. Прерывание от таймера 1.
5. Прерывание от контроллера последовательного ввода/вывода.

Источники прерываний указаны в порядке их приоритета. Старший приоритет у INT0. Для разрешения прерывания используется регистр IE, в котором задействовано 6 битов. Пять битов регистра IP задают уровень приоритета каждого источника независимо. Существует два уровня приоритета.

Если одновременно возникает запрос от двух источников одного уровня, то срабатывает прерывание высшего приоритета. Если одновременно возникает запрос от двух источников одного приоритета, то срабатывает прерывание высшего уровня. Если в обработке находится прерывание высшего уровня, то никакие новые прерывания невозможны.

При обработке прерывания управление программой передается на фиксированный для данного источника адрес, а значение программного счетчика записывается в стек.

Из особенностей прерывания следует отметить, что внешние источники INTO и INT1 могут возникать по уровню или по срезу сигнала, в зависимости от программной установки. Прерывание от таймеров происходит при установке флагов переполнения. Прерывание от канала последовательного ввода/вывода возникает при появлении готовности либо приема, либо передачи. Источник прерывания может имитироваться, если программно установить нужный флаг готовности.

Карта нижней области памяти программ микроконтроллера 8051 показана на рис. 3. По адресу 0 располагается вектор сброса в начальное состояние (к этому же вектору происходит обращение при переполнении сторожевого таймера). При сбросе микроконтроллер переходит к выполнению команды, адрес которой задается этим вектором. После подтверждения прерывания управление передается по соответствующему адресу, то есть прерывание 0 вызывает переход по адресу 03h, прерывание 1 по адресу 0Bh и т. д. На первый взгляд в этом нет ничего особенного, все микроконтроллеры, представленные в данной книге, работают подобным образом. Но особенность заключается в том, что другие микроконтроллеры не предоставляют область памяти для непосредственного размещения программ обработки каждого вида прерываний. Восемь байт памяти программ, выделяемые для каждого обработчика прерываний, могут быть использованы, чтобы разместить обработчик целиком или обеспечить переход к более сложной программе обработки. Выделение восьми байт для размещения команд позволяет достаточно легко осуществлять обращение к простым процедурам обработки прерываний, например, таким, как выполнение сброса прерывания от таймера или увеличение содержимого счетчика реального времени.

Конечно, восьми байт недостаточно, чтобы сохранить содержимое регистров контекста прежде, чем перейти к обработке прерывания. Однако в 8051 нет флага нуля, поэтому содержимое слова состояния PSW сохраняется, если при обработке прерывания не используются команды сложения и вычитания, которые изменяют флаги переноса в регистре PSW. Простая установка битов или инкремент и декремент содержимого регистров выполняются без изменения содержимого PSW.

8-байтовое окно для обработчика прерывания 2	Адрес
8-байтовое окно для обработчика прерывания 1	00013h
8-байтовое окно для обработчика прерывания 0	0000Bh
Вектор сброса	00003h 00000h

Память программ

Рис. 3. Адресация прерываний в 8051

### 1.1.5. Режимы работы

Существует две основные технологии изготовления кристаллов устройств семейства MCS51. Вначале фирмой Intel была предложена технология NMOS, которая не предполагала использование других режимов функционирования. Применение технологии CMOS дало возможность использовать режимы низкого потребления. С этой целью в регистр PCON добавлены два бита – PD и IDL, которые ответственны за реализацию двух независимых режимов.

Режим низкого потребления (Power Down) приводит к полному останову процессора с прекращением работы тактового генератора. Выйти из этого режима можно только при полной инициализации процессора сигналом Reset.

Режим ожидания (Idle) останавливает только выполнение программ. Таймеры и встроенные контроллеры продолжают работать. Выход из этого режима происходит при возникновении любого прерывания.

### 1.1.6. Система команд

Система команд содержит 111 базовых команд, которые удобно разделить по функциональному признаку на пять групп: команды передачи данных, арифметических операций, логических операций, передачи управления и операции с битами. (Приложение 5.)

Система команд Intel8051 содержит 111 базовых команд, которые удобно разделить по функциональному признаку на пять групп: команды передачи данных, арифметических операций, логических операций, передачи управления и операций с битами.

Большинство команд имеют формат один или два байта и выполняются за один или два машинных цикла. При тактовой частоте 12 МГц длительность машинного цикла составляет 1 мкс.

Первый байт команды любых типа и формата всегда содержит код операции (КОП). Второй и третий байты содержат либо адреса операндов, либо непосредственные операнды.

### **1.1.7. Типы операндов**

Состав операндов Intel8051 включает в себя операнды трех типов: биты, тетрады, байты, 16-битные слова. Intel8051 имеет 128 программно управляемых флагов пользователя. Имеется также возможность адресации отдельных бит блока регистров специальных функций и портов. Для адресации бит используется прямой 8-битный адрес (bit). Косвенная адресация бит невозможна. Четырехбитные операнды используются только при операциях обмена (команды SWAP и XCHD). Восемьбитным операндом может быть ячейка памяти программ или данных (резидентной или внешней), константа (непосредственный операнд), регистры специальных функций (SFR), а также порты ввода/вывода. Порты и SFR адресуются только прямым способом. Байты памяти могут адресоваться также и косвенным образом через адресные регистры (R0, R1, DPTR и PC). Двухбайтные операнды это константы и прямые адреса, для представления которых используют второй и третий байты команды.

### **1.1.8. Способы адресации данных**

В Intel8051 используются следующие способы адресации данных:

- прямая,
- непосредственная,
- косвенная,
- неявная.

Способы адресации в микроконтроллере 8051 реализованы с учетом приоритета доступа к определенным массивам данных. В большинстве приложений часто приходится обращаться к очень небольшому количеству переменных, к некоторым переменным обращение происходит время от времени, а большая часть

переменных выбирается достаточно редко. При проектировании 8051 компания Intel определила для доступа к данным однобайтный цикл выполнения команд, который содержит 12 тактов. Доступ к регистрам текущего банка выполняется с помощью однобайтных команд за 12 тактов. Имеются также двухбайтные команды, выполняющие обращение к первым 256 адресам памяти за 24 такта (2 цикла), и трехбайтные команды для доступа к ячейкам памяти за пределами первых 256 байт, для выполнения которых требуется несколько циклов. При разработке программ следует стремиться оптимизировать размещение переменных в памяти данных, чтобы обращение к ним производилось за меньшее число циклов и требовало меньшего числа байт, занимаемых в памяти программ.

Первый из рассматриваемых способов – непосредственная адресация – не требует обращения к регистрам или памяти данных. При непосредственной адресации на обработку поступает операнд, который является частью команды. При программировании на языке ассемблера значение непосредственного операнда помечается символом #, например:

```
add A, #77;          Добавить 77 к содержимому аккумулятора.
```

Эта команда добавляет десятичное число 77 к содержимому аккумулятора, результат поступает обратно в аккумулятор.

Адресация к регистровым банкам обеспечивает доступ к одному из 8 байтов, размещенных в текущем банке. Это наиболее эффективный способ доступа к данным с точки зрения уменьшения объема памяти программ и сокращения числа требуемых командных циклов. Большинство команд, работающих с регистровыми банками, выполняются за один цикл и занимают только один байт. Регистры банка имеют имена начиная с R0 и по R7.

Прямая адресация отличается от регистровой тем, что можно получить доступ к любому байту в первых 256 ячейках памяти, указав 8-разрядный адрес. Ячейки с адресами от 080h по 0FFh не могут быть выбраны при помощи прямой адресации. Если написать команду:

```
mov A, 088h,
```

то в аккумулятор загрузится содержимое служебного регистра TCON (регистр управления таймером), а не содержимое ячейки памяти. Не следует путать прямую адресацию с непосредственной. Если в команде с непосредственной адресацией забыть поставить символ # перед операндом, то будет выполняться команда с прямой адресацией. Это является одной из распространенных синтаксических ошибок при

программировании микроконтроллеров 8051. Следует быть особенно внимательным при разработке и отладке прикладных программ, чтобы быть уверенным, что реализуется требуемый способ адресации.

Косвенно-регистрационная адресация осуществляется с помощью регистров R0 или R1 текущего банка. В этом случае содержимое регистра R0 или R1 используется как 8-разрядный адрес для обращения к первым 256 байтам памяти данных.

orl A,@R0

На выполнение косвенно-регистрационной адресации указывает символ @ перед именем регистра R0 или R1. Использование других регистров банка приведет к ошибке.

Регистр DPTR используется в качестве 16-разрядного индексного регистра. При этом можно указать смещение, которое добавляется к содержимому DPTR для формирования адреса операнда. Такой способ адресации, который называется косвенно-регистрационная со смещением, удобно использовать для доступа к отдельным элементам в структурах данных.

Когда необходимо получить доступ к данным, хранящимся в памяти программ, можно воспользоваться командой «move A, @A+PC». Эта команда адресует в памяти программ байт, который расположен со смещением относительно адреса текущей команды. Значение смещения, которое размещается в аккумуляторе A, должно быть заранее вычислено перед выполнением команды.

Оставшиеся способы адресации используются для изменения текущего содержимого программного счетчика.

### **1.1.9. Флаги результата**

Слово состояния программы PSW включает в себя четыре флага: C – перенос, AC – вспомогательный перенос, OV – переполнение, P – паритет и биты RS0, RS1.

Флаг паритета напрямую зависит от текущего значения аккумулятора. Если число единичных бит аккумулятора нечетное, то флаг P устанавливается, а если четное – сбрасывается. Все попытки изменить флаг P, присваивая ему новое значение, будут безуспешными, если содержимое аккумулятора при этом останется неизменным. Флаг AC устанавливается в случае, если при выполнении операции сложения/вычитания между тетрадами байта возник перенос/заем. Флаг C устанавливается, если в старшем бите результата возникает перенос или заем. При выполнении операций

умножения и деления флаг С сбрасывается, флаг OV устанавливается, если результат операции сложения/вычитания не укладывается в семи битах и старший (восьмой) бит результата не может интерпретироваться как знаковый. При выполнении операции деления флаг OV сбрасывается, а в случае деления на нуль устанавливается. При умножении флаг OV устанавливается, если результат больше 255.

Рассматривая содержимое PSW, можно заметить отсутствие флага нуля. Проверка на нуль в 8051 происходит по-другому. Команда «jz» проверяет содержимое аккумулятора А и выполняет переход, если его содержимое равно нулю.

Биты RS0, RS1 выбирают текущий 8-байтовый банк регистров. Два младших регистра каждого банка (R0, R1) могут использоваться для индексной адресации первых 256 байтов в памяти данных.

#### **1.1.10. Символическая адресация**

При использовании ассемблера Intel8051 (ASM 51) для получения объектных кодов программ допускается применение в программах символических имен регистров специальных функций (SFR), портов и их отдельных бит. Для адресации отдельных бит SFR и портов (такая возможность имеется не у всех SFR) можно использовать символическое имя бита следующей структуры:

< имя SFR или порта >.< номер бита >.

Например, символическое имя пятого бита аккумулятора будет следующим: ACC.5. Символические имена SFR, портов и их бит являются зарезервированными словами для ASM51, и их не надо определять с помощью директив ассемблера.

#### **1.1.11. Адресация внешней памяти**

Микроконтроллер 8051 может адресовать до 64 Кбайт внешней памяти программ и до 64 К внешней статической памяти данных SRAM. Микроконтроллер содержит встроенный интерфейс внешней памяти, который использует для обращения к ней линии портов P0 и P2. Каждый раз, когда при обращении к памяти программ или данных сформированный адрес выходит за границы внутренней памяти, активизируются внешние выводы и микроконтроллер 8051 пытается выбрать внешнюю память. При обращении к внешней памяти сначала выдаются младшие 8 бит адреса, затем по этим же линиям производится запись или чтение данных. Одной из интересных возможностей является использование общей внешней памяти для хранения программ и данных. Это достигается путем объединения сигналов PSEN и RD при помощи логической операции И (рис. 4). С

помощью такого включения можно реализовать схему, в которой микроконтроллер 8051 производит запись в свою память. Такая схема может использоваться в качестве экспериментального средства отладки для загрузки программы в память SRAM.

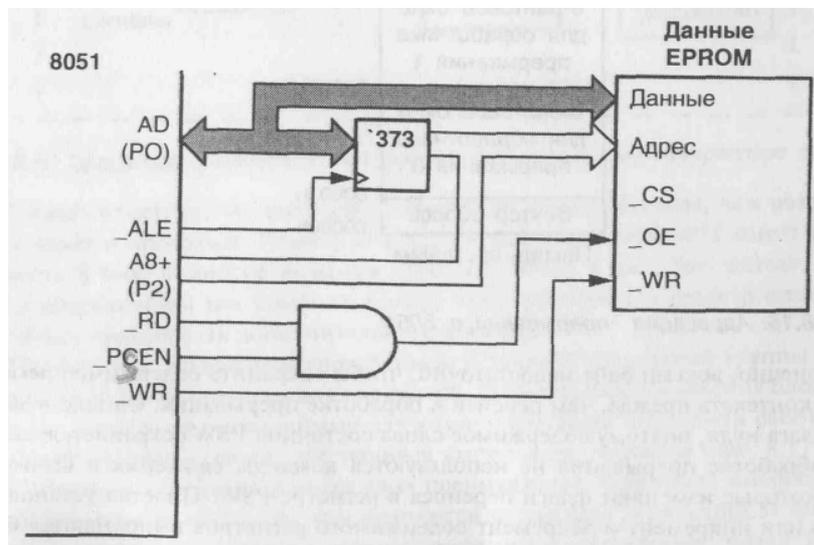


Рис. 4. Реализация общей памяти программ и данных

## 1.2. Развитие в микроконтроллерах фирмы INTEL базовой архитектуры MCS-51

Затем был выпущен микроконтроллер 80C52, который отличался увеличенным объемом памяти программ и данных на кристалле, был введен третий таймер с функциями выборки и сравнения и соответственно расширен контроллер прерывания.

Следующим принципиальным шагом в развитии MCS-51 стал перевод технологии изготовления на CHMOS. Это позволило реализовать режимы Idle и Power Down, позволившие резко снизить энергопотребление кристалла и открывшие дорогу к применению микроконтроллера в энергозависимых приложениях, например в автономных приборах с батарейным питанием.

И последним принципиальным этапом развития этого направления фирмой Intel в рамках 8-битной архитектуры стал выпуск микроконтроллеров 8xC51FA/FB/FC, которые для краткости часто обозначаются как 8xC51FX. Главной отличительной особенностью этой группы кристаллов является наличие у них массива программируемых счетчиков (PCA). Структурная схема PCA представлена на рис. 5.

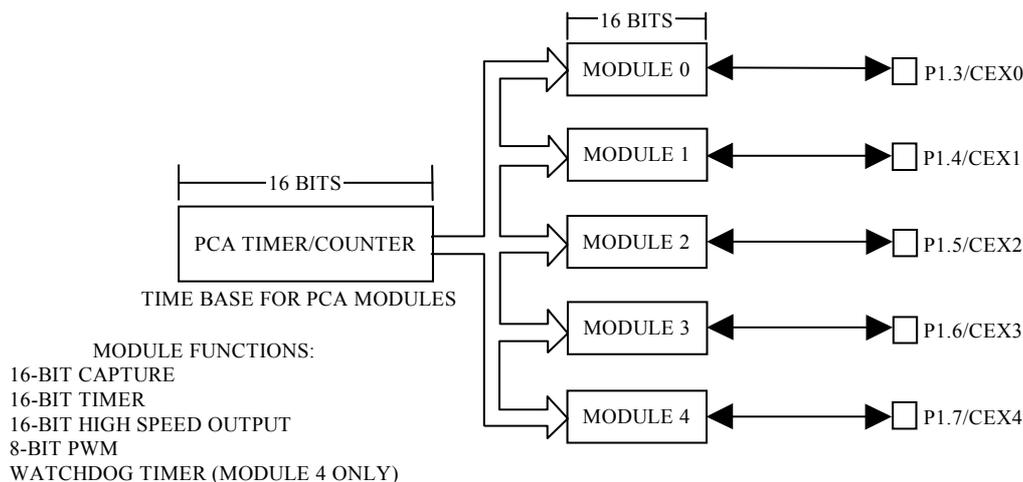


Рис. 5. Структурная схема PCA

В состав PCA входят:

- 16-разрядный таймер-счетчик;
- 5 шестнадцатиразрядных модулей выборки и сравнения, каждый из которых связан со своей линией порта ввода/вывода микроконтроллера.

Таймер-счетчик обслуживает все пять модулей выборки и сравнения, которые могут быть запрограммированы на выполнение одной из следующих функций:

- 16-битная выборка значения таймера по положительному фронту внешнего сигнала;
- 16-битная выборка значения таймера по отрицательному фронту внешнего сигнала;
- 16-битная выборка значения таймера по любому фронту внешнего сигнала;
- 16-битный программный таймер;
- 16-битное устройство скоростного вывода (HSO);
- 8-битный ШИМ.

Выполнение всех перечисленных функций происходит в PCA на аппаратном уровне и не загружает центральный процессор, что позволяет повысить общую пропускную способность системы, повысить точность измерений и обработки сигналов и снизить время реакции микроконтроллера на внешние события, что особенно важно для систем реального времени. Реализованный в 8xC51FX PCA оказался настолько удачным, что архитектура микроконтроллеров FX стала промышленным стандартом де-факто, а сам PCA многократно воспроизводился в различных модификациях микроконтроллеров разных фирм.

ТАБЛИЦА 2

## Список микроконтроллеров семейства 8051, выпускаемых фирмой Intel

Тип	Макс. част., МГц	ROM/ EPROM, байт	RAM, байт	Таймеры/ счетчики	Линии ввода/ вывода	Послед. каналы	АЦП, входы/ разр.	Другая периферия, особенности	Напряжение питания, В	Темп. диапазон, С	Корпус
8xC51BH (80C31BH)	24	4 К	128	2	32	UART	–	–	5+/-20 %	-40...+85	D40, L44, Q44
8xC52 (80C32)	24	8 К	256	3	32	UART	–	–	5+/-20 %	-40...+85	D40, L44, Q44
8xC54	33	16 К	256	3	32	UART	–	–	5+/-20 %	-40...+85	D40, L44, Q44
8xC58	33	32 К	256	3	32	UART	–	–	5+/-20 %	-40...+85	D40, L44, Q44
8xC5x-L	24	8...32 К	256	3	32	UART	–	Низковольтный вариант 8xC52/54/58	2,7...3,6	-40...+85	D40, L44, Q44
8xC51FA	24	8 К	256	3+PCA	32	UART	–	–	5+/-20 %	-40...+85	D40, L44, Q44
8xC51FB	24	16 К	256	3+PCA	32	UART	–	4 уровня IRQ, clock out	5+/-20 %	-40...+85	D40, L44, Q44
8xC51FC	24	32 К	256	3+PCA	32	UART	–	4 уровня IRQ, clock out	5+/-20 %	-40...+85	D40, L44, Q44
8xL51Fx	16	8...32 К	256	3+PCA	32	UART	–	Низковольтный вариант 8xC51Fx	2,7...3,6	-40...+85	D40, L44, Q44
8xC51RA	24	8 К	512	3+PCA+ Watch-dog	32	UART	–	4 уровня IRQ, clock out	5+/-20 %	-40...+85	D40, L44, Q44
8xC51RB	24	16 К	512	3+PCA+ Watch-dog	32	UART	–	4 уровня IRQ, clock out	5+/-20 %	-40...+85	D40, L44, Q44
8xC51RC	24	32 К	512	3+PCA+ Watch-dog	32	UART	–	4 уровня IRQ, clock out	5+/-20 %	-40...+85	D40, L44, Q44
8xC51GB	16	8 К	256	3+ PCAx2+ Watch-dog	48	UART	8x8	OFD, clock out	5+/-20 %	-40...+125	LCC-68
8xC152JA/JC	16,5	8 К	256	2	40	UART, GSC	–	DMAx2, Multi-Protocol Serial Communication	5+/-20 %	0...+70	DIP-48, LCC-68

Условные обозначения корпусов: D – DIP, L – LCC, Q – QFP, S – SSOP, SOIC.

Изначально наиболее “узкими” местами архитектуры MCS-51 были 8-разрядное АЛУ на базе аккумулятора и относительно медленное выполнение инструкций (для выполнения самых быстрых инструкций требуется 12 периодов тактовой частоты). Это ограничивало применение микроконтроллеров семейства в приложениях, требующих повышенного быстродействия и сложных вычислений (16- и 32-битовых). Насущным стал вопрос принципиальной модернизации старой архитектуры. Проблема модернизации осложнялась тем, что к началу 90-х годов уже была создана масса наработок в области программного и аппаратного обеспечения, и одной из основных задач разработки новой архитектуры была реализация аппаратной и программной совместимости со старыми разработками на базе MCS-51. Для решения этой задачи была создана совместная группа из специалистов компаний Intel и Philips, но позднее пути этих двух фирм разошлись. В результате в 1995 году появилось 2 существенно отличающихся семейства: MCS-251/151 у Intel и 51XA у Philips (на последнем мы остановимся ниже).

Основные характеристики архитектуры MSC-251:

- 24-разрядное линейное адресное пространство, обеспечивающее адресацию до 16М памяти;
- регистровая архитектура, допускающая обращение к регистрам как к байтам, словам и двойным словам;
- страничный режим адресации для ускорения выборки инструкций из внешней программной памяти;
- очередь инструкций;
- расширенный набор команд, включающий 16-битовые арифметические и логические инструкции;
- расширенное адресное пространство стека до 64К;
- выполнение самой быстрой инструкции за 2 такта;
- совместимость на уровне двоичного кода с программами для MCS-51.

Система команд MCS-251 построена на базе двух наборов инструкций – первый набор является копией системы команд MCS-51, а второй набор состоит из расширенных инструкций, реализующих

преимущества архитектуры MSC-251. Перед использованием микроконтроллера его необходимо сконфигурировать, т. е. с помощью программатора “прожечь” конфигурационные байты, определяющие, какой из наборов инструкций станет активным после включения питания. Если установить набор инструкций MCS-51, то в этом случае MSC-251 будет совместим с MCS-51 на уровне двоичного кода. Такой режим называется Binary Mode. Однако расширенные инструкции в этом режиме также доступны через “форточку” – зарезервированный код инструкции 0A5h. Естественно, длина каждой расширенной инструкции увеличивается в таком случае на 1 байт. Если же изначально установить набор расширенных инструкций, то в этом случае программы, написанные для MCS-51, потребуют перекомпиляции на кросс-средствах для MCS-51, т. к. теперь уже стандартные инструкции будут доступны через ту же «форточку» 0A5h и длина их также увеличится на 1 байт. Такой режим называется Source Mode. Он позволяет с максимальной эффективностью использовать расширенные инструкции и достигнуть наибольшего быстродействия, но требует переработки программного обеспечения.

Для пользователей, ориентированных на применение микроконтроллеров MCS-251, в качестве механической замены MCS-51 фирма Intel выпускает микроконтроллеры MCS-251 с уже запрограммированными битами конфигурации в состоянии Binary Mode. Такие микроконтроллеры получили индекс MCS-151.

Помимо самой Intel, микроконтроллеры MCS-251 по ее лицензии выпускает компания Temic Semiconductors.

В настоящее время Intel, устремленная на рынок Pentium-процессоров, сворачивает производство кристаллов MCS-51. В целом для конкретного разработчика этот факт может остаться и незамеченным, если только он не использует микроконтроллеры 8xC51GB и 80C152Jx – эти кристаллы не имеют своих точных аналогов среди изделий других фирм. Что же касается всех остальных микроконтроллеров семейства MCS-51, то все они многократно растиражированы другим компаниями-производителями микроконтроллеров и уход Intel с рынка микроконтроллеров никак не скажется на этом самом рынке, поскольку Intel давно уже не играла на нем ведущую роль.

ТАБЛИЦА 3

Тип	Макс. частота, МГц	ROM/ EPROM байт	RAM, байт	Таймеры/ счетчики	Линии ввода/ вывода	Послед. каналы	АЦП, входы/ разр.	Напряжение питания, В	Темп. диапазон, С	Корпус
8xC251SA	16	8 К	1 К	3+PCA+WDT	32	UART	–	4,5...5,5	–40...+85	D40, L44
8xC251SB	16	16 К	1 К	3+PCA+WDT	32	UART	–	4,5...5,5	–40...+85	D40, L44
8xC251SP	16	8 К	512	3+PCA+WDT	32	UART	–	4,5...5,5	–40...+85	D40, L44
8xC251SQ	16	16 К	512	3+PCA+WDT	32	UART	–	4,5...5,5	–40...+85	D40, L44
TSC8xC251G1	16	16 К	1 К	3+WDT	32	UART, I <sup>2</sup> C, SPI	–	4,5...5,5	–40...+85	L40, Q44
TSC8xC251A1	16	24 К	1 К	2+WDT	32	UART	4x8 bit	4,5...5,5	–40...+85	D40, L44, Q44
8Xc151SA	16	8 К	256	3+PCA+WDT	32	UART	–	4,5...5,5	–40...+85	D40, L44
8xC151SB	16	16 К	256	3+PCA+WDT	32	UART	–	4,5...5,5	–40...+85	D40, L44

Условные обозначения корпусов: D – DIP, L – LCC, Q – QFP.

### 1.3. Микроконтроллеры фирмы ATMEL

Очередным вкладом в повышение популярности семейства MCS-51 стали AT89 – микроконтроллеры фирмы Atmel, изготовленные по КМОП (CMOS) технологии, имеющие полностью статическую структуру и оснащенные встроенной Flash-памятью программ. Это позволило создавать более компактные и экономичные изделия, а также значительно облегчило разработку, отладку и запись в них программного обеспечения. Кроме того, в некоторых своих изделиях фирма Atmel начала использовать так называемый “X2 режим ускорения”, заключающийся в том, что машинный цикл стал выполняться за 6 тактов, а не за 12, как у первых микроконтроллеров семейства x51. Это позволяет им на тактовой частоте 20 МГц работать также быстро, как стандартные микроконтроллеры работали бы на частоте 40 МГц. Многие развитые микроконтроллеры фирмы Atmel начали оснащаться аналогово-цифровыми преобразователями (ADC – Analog to Digital Converters), охранными таймерами (WDT) и другими периферийными узлами.

ТАБЛИЦА 4

Тип	Макс. частота, МГц	FLASH	RAM, байт	EEPROM, байт	Таймеры/ счетчики	Линии ввода/ вывода	Послед. каналы	Другая периферия, особенности	Напряжение питания, В	Темп. диапа- зон, С	Корпус
AT89C1051	24	1 К	64	–	1	15	–	On-chip Analog Comparator	2,7...6,0	–55...+125	D20, S20
AT89C2051	24	2 К	128	–	2	15	UART	On-chip Analog Comparator	2,7...6,0	–55...+125	D20, S20
AT89C51	24	4 К	128	–	2	32	UART	–	4,0...6,0	–55...+125	D40, L44, Q44
AT89LV51	12	4 К	128	–	2	32	UART	–	2,7...6,0	–55...+125	D40, L44, Q44
AT89C52	24	8 К	256	–	3	32	UART	–	4,0...6,0	–55...+125	D40, L44, Q44
AT89LV52	12	8 К	256	–	3	32	UART	–	2,7...6,0	–55...+125	D40, L44, Q44
AT89C55	33	20 К	256	–	3	32	UART	–	4,0...6,0	–55...+125	D40, L44, Q44
AT89S8252	24	8 К	256	2К	3+ Watch- dog	32	UART + SPI	2 DPTR	2,7...6,0	–55...+125	D40, L44, Q44
AT89S53	24	12 К	256	–	3+ Watch- dog	32	UART + SPI	2 DPTR	4,0...6,0	–55...+105	D40, L44, Q44

Условные обозначения корпусов: D – DIP, L – LCC, Q – QFP, S – SSOP, SOIC.

Золотой ключик фирмы Atmel – это Flash-технология. Владея ею в совершенстве, Atmel строит свою политику на внедрении Flash в наиболее популярные микросхемы, использовавшие до того технологию EPROM – это и микросхемы памяти, и программируемая логика, и микроконтроллеры. Тот факт, что в качестве первого кандидата на прорыв на рынок микроконтроллеров, где господствовали такие монстры, как Philips, Siemens, Intel, Motorola, Mitsubishi и др., фирма Atmel избрала микроконтроллер 8051, свидетельствует об исключительно высокой популярности этого кристалла на рынке. Atmel – самая молодая и самая агрессивная среди всех компаний, выпускающих микроконтроллеры 51-го семейства. Она смогла предложить невероятно низкие цены на свою продукцию и быстро завоевала высокую популярность на рынке микроконтроллеров. Начиная с 1995 года компания ежегодно удваивает объемы продаж своих микроконтроллеров 51-го семейства. Среди оригинальных разработок Atmel:

- 20-выводные микроконтроллеры 89C2051/1051, открывшие дорогу микроконтроллерам 51-го семейства в сфере «1-долларовых» приложений;

- микроконтроллеры с каналом SPI, обеспечивающие возможность внутрисхемного программирования Flash. Эта функция может быть очень удобна в производстве, когда программирование микросхем осуществляется уже после их монтажа. Особенно это важно для мелкосерийного производства, так как позволяет постоянно совершенствовать программное обеспечение без изменения аппаратуры;

- внутренняя память данных на базе EEPROM, обеспечивающая хранение оперативных данных при отключенном питании микросхемы.

В планах фирмы на 1998 год – выпуск микроконтроллера с объемом Flash 128 Кбайт.

#### **1.4. Микроконтроллеры фирмы DALLAS Semiconductor**

Фирма Dallas Semiconductor была создана 1 февраля 1984 года и за свою историю поставила 215 основных видов продукции и свыше 1000 модификаций изделий. Фирма занимается разработкой, производством и сбытом интегральных микросхем и законченных микромодулей SIP STIK на их основе. Фирма является основным поставщиком энергонезависимой памяти. В корпусе микросхем энергонезависимой памяти размещается кристалл статического ОЗУ с низким потреблением питания и литиевая батарейка. Она обеспечивает сохранность информации в течение 10 лет. Энергонезависимая память обладает неограниченным числом циклов записи. Она является основой многих устройств, выпускаемых фирмой Dallas.

Dallas Semiconductor занимается разработкой, производством и сбытом интегральных микросхем и законченных микромодулей на их основе, которые называются STIC. Компания быстро завоевала репутацию новатора, создав ряд новых изделий, не выпускаемых другими фирмами.

Семейства высокопроизводительных МК 80C3X0 и 87C5X0 являются одними из наиболее производительных среди 8-разрядных МК. Они используют для выполнения команды только 4 такта по сравнению с 12 тактами у остальных совместимых 8051 прибора. Кроме этого, в состав микросхем входят 2 последовательных порта, монитор питания и сторожевой таймер.

Отличительной особенностью семейства МК DS500X является обеспечение высокого уровня защиты доступа к программам и данным. Механизм защиты включает в себя шифрование адресов и данных в памяти с использованием 64-разрядного ключа и стирание содержимого памяти и ключей при попытке несанкционированного доступа к памяти. МК DS5000T содержит также часы реального времени и встроенное энергонезависимое ОЗУ объемом до 64 Кб. Загрузка ОЗУ осуществляется через последовательный порт МК, после чего может быть произведено ее динамическое распределение для хранения данных и программ.

В результате разработки новых технологических процессов корпорация Dallas Semiconductor выпускает продукцию типа Soft Silicon – динамические, гибкие приборы на базе ИС, которые могут быть сформированы на последних стадиях производства или даже во время их применения. Выпуск продукции типа Soft Silicon стал возможным благодаря новейшим достижениям в технологии производства литиевых источников питания и непосредственному формированию рисунка интегральной схемы лучом лазера.

Семейство Soft Micro является линией совместимых с 8051 МК, поэтому нет ничего удивительного в том, что практически все члены семейства DS5000XX являются совместимыми не только программно с 8051-м, но и по выводам на корпусе. На данное семейство следует обратить особое внимание, так как именно оно наиболее широко из всех представлено на белорусском рынке и пользуется хорошим спросом у отечественных разработчиков благодаря своей простоте и привычности в пользовании.

Рассмотрим особенности данного семейства:

- совместимая с 8051 система команд;
- адресует 64 К памяти программ и данных;
- четыре 8-битных псевдодвухнаправленных порта ввода/вывода;
- 128 байт RAM;
- 1 UART;
- 5 прерываний, в том числе 2 внешних.

Это была общая характеристика семейства. Теперь предлагается рассмотреть каждый МК в отдельности.

Приняв во внимание вышесказанное, нетрудно догадаться, что в принципе нет необходимости перечислять все особенности этих МК, так как в основу их положен традиционный 8051, поэтому укажем только их отличия:

ТАБЛИЦА 5

Прибор	Техн. изг.	Память	Такт. част.	Часы р. в.	№ в серии
DS5000	Soft Micro	8 Kb	8Mhz	Нет	DS500-08-08
DS5000	Soft Micro	8 Kb	12Mhz	Нет	DS500-08-12
DS5000	Soft Micro	8 Kb	16Mhz	Нет	DS500-08-16
DS5000	Soft Micro	32 K	8Mhz	Нет	DS500-32-08
DS5000	Soft Micro	32 K	12Mhz	Нет	DS500-32-12
DS5000T	Soft Micro	32 K	16Mhz	Нет	DS500-32-16
DS5000T	Soft Micro	8 Kb	8Mhz	Есть	DS500T-08-08
DS5000T	Soft Micro	8 Kb	12Mhz	Есть	DS500T-08-12
DS5000T	Soft Micro	8 Kb	16Mhz	Есть	DS500T-08-16
DS5000T	Soft Micro	32 K	8Mhz	Нет	DS500T-32-08
DS5000T	Soft Micro	32 K	12Mhz	Нет	DS500T-32-12
DS5000T	Soft Micro	32 K	16Mhz	Нет	DS500T-32-16

На рис. 6. приведена структурная схема МК DS5000T.

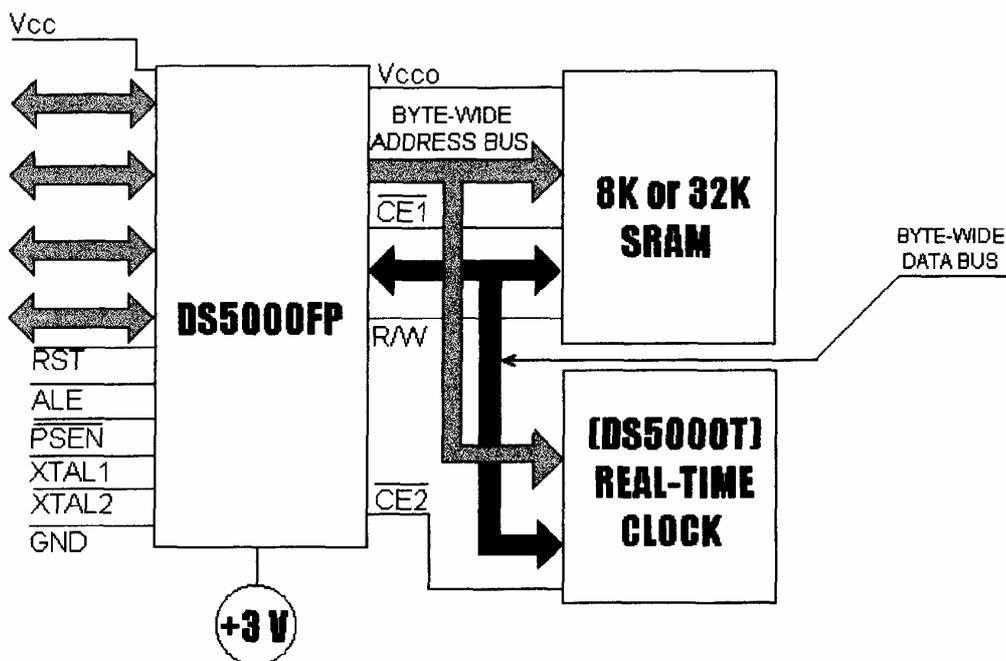


Рис. 6. Архитектура DS5000T

Рассмотрим также назначение выводов:

- Широки на рисунке обозначены порты ввода/вывода общего назначения (P1.0 – P1.7).
- RST – сброс текущего состояния прибора.
- XTAL1, XTAL2 – предназначены для подключения внешнего устройства к внутреннему осциллятору.
- GND – логическая “земля”.
- PSEN – разрешение размещения программы во внешней памяти.

- ALE – демultipлексирует мультимплексированную расширенную шину данных.
- Vcc – +5В.
- R/W – разрешение чтения/записи памяти данных.

DS5000T содержит четыре основные шины: внутренние шины адреса и данных, шину памяти и шину расширения. Все передающиеся в процессе выполнения программы адреса и данные проходят по внутренним шинам. Доступ к памяти данных и программ пользователя можно обеспечить, используя либо шину памяти, либо шину расширения. Аппаратно шина расширения идентична шине микроконтроллера 8051, то есть использует выводы портов 0 и 2. Шина памяти позволяет освободить выводы портов при доступе к памяти и использовать их по прямому назначению.

Все регистры процессора обозначаются как Регистры Специальных Функций (SFR). Они идентичны по номеру и функциям регистрам 8051.

Микроконтроллер DS5000T содержит внутреннюю регистровую память. Регистры адресуются числами от 00 до 7Fh и могут быть доступны команде MOV. Регистровая память включает в себя 4 8-байтовых банка рабочих регистров. Эта память не является частью внешней памяти данных.

## **1.5. Микроконтроллеры и средства разработки компании Silicon Labs (Cygnal)**

Подразделение микроконтроллеров (в минувшем компания Cygnal Integrated Products) компании Silicom Laboratories – один из признанных мировых лидеров в разработке и производстве высокопроизводительных 8051-совместимых микроконтроллеров. В настоящее время компанией разработано более 50 микроконтроллеров, делящихся на 10 семейств в зависимости от конфигурации. Микроконтроллеры Cygnal имеют ряд особенностей, выгодно отличающих их от микроконтроллеров других фирм.

Ядро Cygnal CIP-51 – это ядро стандартной 8-ми битной архитектуры и полностью совместимо с MCS-51 стандартом инструкций Intel. В отличие от микроконтроллеров, выполняющих одну инструкцию за 4-12 машинных тактов, ядро Cygnal CIP-51 выполняет до 70% инструкций за 1-2 машинных такта. Таким образом, ядро, работающее фактически на частоте тактового генератора, (а в семействе C8051F12x на частотах вдвое и вчетверо

выше) обеспечивает производительность 25 MIPS миллионов операций в секунду (в C8051F12x 50 и 100 MIPS). Преимущество в производительности достигает нескольких десятков раз по сравнению с аналогами. Ядро содержит дополнительный обработчик прерываний (от 12 до 22 источников) и от 3 до 5 шестнадцати битных таймеров общего применения.

В зависимости от модели МК Cygnal, в нем содержится от 2 до 128 Кбайт программируемой в системе Flash памяти, организованной секторами по 512 байт, каждый из которых может быть как памятью программ, так и памятью данных (EEPROM). От 256 байт до 8 Кбайт + 256 байт оперативной памяти.

В зависимости от семейства микроконтроллеры Cygnal имеют различные аппаратно встроенные интерфейсы:

- SMBus/I2C;
- SPI;
- UART (до 2-х интерфейсов, работающих независимо);
- промышленный CAN 2.0B Bosch;
- USB 2.0 (с поддержкой версии 1.0) + 1Кбайт FIFO ОЗУ;
- встроенный интерфейс доступа к внешней памяти данных с мультиплексированной или не мультиплексированной шиной адрес/данные.

*Встроенные аналого-цифровые функции микроконтроллера Cygnal:*

- многоканальные аналого-цифровые преобразователи на 8, 10, 12, 16 и 24 бит с производительностью от 100 ksps до 1000 ksps (тысяч преобразований в секунду) в зависимости от семейства;
- возможность работы АЦП в дифференциальном и одноканальном режимах;
- программируемый “оконный детектор”, позволяющий отслеживать аналоговую величину в заданном диапазоне значений;
- встроенные программируемые предусилители АЦП и высоковольтный дифференциальный усилитель (в семействе C8051F04x), допускающий напряжение до  $\pm 60\text{В}$  прямо на выводы процессора;
- встроенные двухканальные цифро-аналоговые преобразователи с разрешением 12 бит;
- встроенный датчик температуры;
- встроенные компараторы (от 1 до 3 в зависимости от семейства) с программируемым временем срабатывания и гистерезисом;
- возможность дифференцированного использования в качестве

источников опорных напряжений для АЦП и ЦАП внешнего напряжения, напряжения питания ядра, или встроенного источника опорного напряжения.

*Дополнительные функции МК компании Cygnal:*

- программируемый встроенный генератор, в том числе с прецизионной калибровкой, позволяющий микроконтроллеру “налету” переключаться между источниками тактовых сигналов (внешний – кварц, RC-цепочка) или встроенный и таким образом гибко регулировать потребляемую мощность;

- встроенная программируемая счетная матрица (ПСМ) с 3-6 модулями захвата/сравнения;

- встроенный ШИМ генератор, строящийся на базе ПСМ;

- встроенная коммутационная матрица (Crossbar), позволяющая перераспределять альтернативные функции портов;

- встроенный Watchdog таймер и двунаправленный Reset.

*Электрические характеристики МК:*

- Напряжение питания ядра от 2,7В до 3,6В;

- Минимальное напряжение сохранения информации в ОЗУ – 1,5В;

- Токи потребления на частоте ядра 25 МГц от 5 до 25 мА, в зависимости от типа микроконтроллера;

- Токи потребления в энергосберегающих режимах (IDLE и Shut Down) достигают единиц мкА.;

- Порты микроконтроллеров совместимы с 5В периферией и не требуют дополнительных буферных компонентов.

Микроконтроллеры семейства C8051F32x имеют встроенный стабилизатор напряжения для работы непосредственно от USB шины, который может использоваться для питания не только ядра, но и внешних устройств.

*Корпус и рабочая температура.* Все микроконтроллеры Cygnal производятся в современных корпусах для поверхностного монтажа, занимающих минимум места на плате. Это как стандартные TQFP и LQFP от 32 до 100 выводов, так и уникальные корпуса серии MLP, самый маленький из которых имеет размеры всего 3x3 мм.

Диапазон рабочих температур для всех микроконтроллеров от -40°C до +85°C. Температура хранения от -65°C до +150°C.

Перечень микроконтроллеров и их основные характеристики приведены в табл. 6.

ТАБЛИЦА 6

## Краткие характеристики и технические описания МК фирмы Cugnal

ТИП	MIPS	FLASH Кбайт	ОЗУ Байт	Интерфейсы	Таймеры 16 бит	АЦП/ЦАП	Компараторы, Датчик температуры, РСА модули	Тип корпуса	
C8051F300	25	8	256	SMBus/I2C, UART	3	АЦП 8 бит 500 ksp/s	1 комп, дат темп, 3 РСА	MLP-11	
C8051F301						-	1 комп, 3 РСА		
C8051F302						АЦП 8 бит 500 ksp/s	1 комп, дат темп, 3 РСА		
C8051F303						-	1 комп, 3 РСА		
C8051F304						-			
C8051F305						2			
C8051F310	25	16	1K+256	SMBus/I2C, SPI, UART	4	АЦП 10 бит 100 ksp/s	2 комп, дат темп, 5 РСА	32LQFP	
C8051F311								MLP-28	
C8051F320	25	16	1K+256	SMBus/I2C, SPI, UART, USB 2.0	4	АЦП 10 бит 200 ksp/s	2 комп, дат темп, 5 РСА	32LQFP	
C8051F321								MLP-28	
C8051F330	25	8	512+256	SMBus/I2C, SPI, UART	4	АЦП 10 бит 200 ksp/s, ЦАП 10 бит	1 комп, дат темп, 3 РСА	MLP-20	
C8051F331						-	1 комп, 3 РСА		
C8051F350	50	8	512+256	SMBus/I2C, SPI, UART	4	АЦП 24 бит, ЦАП 8 бит	1 комп, дат темп, 3 РСА	32LQFP	
C8051F351								MLP-28	
C8051F352						АЦП 16 бит, ЦАП 8 бит		32LQFP	
C8051F353								MLP-28	
C8051F206	25	8	1K+256	SPI, UART	3	АЦП 12 бит 100 ksp/s	-	48TQFP	
C8051F220			256			48TQFP			
C8051F221						32LQFP			
C8051F226			1K+256			48TQFP			
C8051F230			256			48TQFP			
C8051F231						32LQFP			
C8051F236			1K+256			48TQFP			
C8051F000	20	32	256	SMBus/I2C, SPI, UART	4	АЦП 12 бит 100 ksp/s, ЦАП 12 бит	2 комп, дат темп, 5 РСА	64TQFP	
C8051F001							48TQFP		
C8051F002						АЦП 12 бит 100 ksp/s, ЦАП 12 бит	1 комп, дат темп, 5 РСА	32LQFP	
C8051F005			2K+256			АЦП 12 бит 100 ksp/s, ЦАП 12 бит	2 комп, дат темп, 5 РСА	64TQFP	
C8051F006							48TQFP		
C8051F007						АЦП 12 бит 100 ksp/s, ЦАП 12 бит	1 комп, дат темп, 5 РСА	32LQFP	
C8051F010			256			АЦП 10 бит 100 ksp/s, ЦАП 12 бит	2 комп, дат темп, 5 РСА	64TQFP	
C8051F011							48TQFP		
C8051F012						АЦП 10 бит 100 ksp/s, ЦАП 12 бит	1 комп, дат темп, 5 РСА	32LQFP	
C8051F015			25			2K+256	АЦП 10 бит 100 ksp/s, ЦАП 12 бит	2 комп, дат темп, 5 РСА	64TQFP
C8051F016								48TQFP	
C8051F017							АЦП 10 бит 100 ksp/s, ЦАП 12 бит	1 комп, дат темп, 5 РСА	32LQFP
C8051F018	25	16	1K+256	SMBus/I2C, SPI, UART	4	АЦП 10 бит 100 ksp/s	2 комп, дат темп, 5 РСА	64TQFP	
C8051F019								48TQFP	
C8051F020	25	64	4K+256	SMBus/I2C, SPI,	5	АЦП 12 бит 100 ksp/s, 8 бит 500 ksp/s, ЦАП 12 бит	2 комп, дат темп, 5 РСА	100TQFP	
C8051F021				2 UART,		64TQFP			
C8051F022				Ext memory		АЦП 10 бит 100 ksp/s, 8 бит 500 ksp/s, ЦАП 12 бит		100TQFP	
C8051F023								64TQFP	

ТАБЛИЦА 6 (продолжение)

ТИП	MIPS	FLASH Кбайт	ОЗУ Байт	Интерфейсы	Таймеры 16 бит	АЦП/ЦАП	Компараторы, Датчик температуры, РСА модули	Тип корпуса
C8051F040	25	64	4K+256	SMBus/I2C, SPI,	5	АЦП 12 бит 100 ksp, 8 бит 500 ksp, ЦАП 12 бит	3 комп, дат темп, 6 РСА	100TQFP
C8051F041				2 UART,				64TQFP
C8051F042				Ext memory,				100TQFP
C8051F043				CAN 2.0B				64TQFP
C8051F060	25	64	4K+256	SMBus/I2C, SPI,	5	АЦП 16 бит 1 Msp, 12 бит 100 ksp, ЦАП 12 бит	3 комп, дат темп, 6 РСА	100TQFP
C8051F061				2 UART,				64TQFP
C8051F062				Ext memory,				100TQFP
C8051F063				CAN 2.0B				64TQFP
C8051F120	100	128	8K+256	SMBus/I2C, SPI,	5	АЦП 12 бит 100 ksp, 8 бит 500 ksp, ЦАП 12 бит	2 комп, дат темп, 6 РСА	100TQFP
C8051F121				2 UART,				64TQFP
C8051F122				Ext memory				100TQFP
C8051F123								64TQFP
C8051F124	50	128	8K+256	SMBus/I2C, SPI,	5	АЦП 12 бит 100 ksp, 8 бит 500 ksp, ЦАП 12 бит	2 комп, дат темп, 6 РСА	100TQFP
C8051F125				2 UART,				64TQFP
C8051F126				Ext memory				100TQFP
C8051F127								64TQFP
CP2101	12 Mbps		512+512	USB — UART	-	-	-	MLP-28

## 1.6. Микроконтроллеры фирмы PHILIPS

Фирму PHILIPS можно по праву назвать ведущей по количеству выпускаемых ею модификаций семейства 8051, их более 100. В его состав от PHILIPS входят микроконтроллеры в корпусах с числом выводов от 24 до 80, с тактовыми частотами до 40 МГц и напряжением питания от 1,8 В. Во всех этих устройствах используется стандартное ядро MCS-51, поэтому все временные и функциональные характеристики полностью соответствуют характеристикам микроконтроллеров INTEL. Фирма PHILIPS значительные усилия направила на интегрирование широкого спектра периферийных устройств на базе ядра 8051. Основные элементы периферии PHILIPS:

- АЦП с точностью преобразования до 10 разрядов;
- широтно-импульсные модуляторы;
- массивы программируемых счетчиков-таймеров;
- интерфейсы с процессорными шинами;
- EEPROM и FLASH на кристалле;
- специализированная периферия для телевизионной, видео- и аудиотехники.

В 1997 году фирма взяла курс на развитие Flash-технологии в производстве МК. Руководством PHILIPS официально заявлено, что в течение 1998–1999 гг. все выпускаемые микроконтроллеры будут переведены на FLASH-технологию.

С 1997 году фирма Philips переводит стандартные микроконтроллеры групп 80C51, 80C52/54/58 и 80C51FX на новую технологию, которую она сама назвала “New-and-Improved”, т. е. “Новая и улучшенная”. Какие же новые возможности появились у хорошо известных кристаллов после модернизации:

- максимальная тактовая частота кристаллов увеличена до 33 МГц;
- расширен диапазон напряжения питания от 2,7 до 5,5 В;
- количество аппаратных уровней прерываний увеличено до 4;
- во все кристаллы введена функция программируемого clock-out;
- UART заменен на улучшенный (enhanced);
- добавлена функция снижения электромагнитных помех (Lower EMI);
- добавлен второй DPTR;
- потребление энергии для питания микроконтроллера снижено на 50 %. В сочетании с 3-х вольтным питанием это может дать экономию до 75 % по сравнению с предыдущими образцами;
- снижена цена на 30 %.

Фактически такие новые возможности дают второе рождение старым кристаллам. Проблема для разработчика, однако, состоит в том, что маркировка микроконтроллеров после модернизации не изменилась, из-за чего возможна путаница между старыми и новыми модификациями.

Кроме того, фирма Philips выпустила группу микроконтроллеров, названную RX+. По сути, это дальнейшее развитие группы FX, в которой расширен объем внутреннего ОЗУ (512 байт, 1 Кбайт) и программной памяти (до 64 К). Группа RX+ обладает также всеми возможностями, предоставляемыми технологией “New-and-Improved”.

### **Микроконтроллеры с Flash-памятью**

В 1997 г. фирма Philips взяла четкий курс на развитие Flash-технологии в производстве микроконтроллеров. Отчасти это вызвано высокими технологическими возможностями Philips, отчасти успехами конкурентов, в первую очередь Atmel. В интервью журналу

ЕЕTimes David Fair, менеджер по маркетингу фирмы Philips, заявил, что “в ближайшие 3–5 лет микроконтроллеры на базе однократно программируемой EPROM будут постепенно исчезать и замещаться микроконтроллерами с Flash-памятью. Несмотря на то, что Flash-память дороже в производстве, чем EPROM, в конечном итоге для фирмы дешевле будет поддерживать единый технологический процесс”. Руководством фирмы Philips официально заявлено, что в течение 1998–99 годов все микроконтроллеры Philips будут переведены на Flash-технологию.

### **Микроконтроллеры семейства 80C51XA**

Фирма Philips предложила свое оригинальное решение по развитию архитектуры семейства 8051, выпустив семейство микроконтроллеров с индексом 51XA, что буквально переводится как «расширенная архитектура 51». Разработчики Philips решили отказаться от достижения совместимости кодов инструкций нового микроконтроллера с существующим ядром семейства 8051. Такое решение, с одной стороны, сделало невозможным прямую подмену микроконтроллера семейства 8051 на 51XA, с другой – развязало руки разработчикам для кардинальной переработки архитектуры на основе полноценного 16-разрядного ядра.

Основные характеристики семейства 8051XA:

- 16-разрядное АЛУ на базе регистровой архитектуры;
- 24-разрядное адресное пространство (16 Мбайт памяти) для памяти программ и данных;
- 8 16-разрядных регистров для выполнения арифметических и логических операций;
- расширенный набор инструкций;
- аппаратная поддержка мультизадачности;
- выполнение инструкций типа регистр-регистр за 100 нс;
- напряжение питания от 2,7 В.

Что же позволяет отнести этот совершенно оригинальный микроконтроллер к семейству 8051? Как уже упоминалось, программное обеспечение 8051 напрямую непереносимо на XA. Однако каждой инструкции 8051 обязательно соответствует ее аналог среди инструкций XA. Этот принцип использован в специальной программе-трансляторе исходного текста на ассемблере 8051 в исходный текст для ассемблера XA. Это позволяет говорить о совместимости XA с 8051 по исходному тексту. Кроме того, в XA реализована такая же структура памяти, как и в 8051, что оставляет

неизменными способы обращения к памяти программ, данных, регистрам специального назначения, применявшиеся в программах для семейства 8051, и обеспечивает совместимость с микроконтроллерами 8051 на уровне аппаратуры.

По расчетам Philips, архитектура ХА обеспечивает увеличение быстродействия до 100 раз по сравнению с традиционной архитектурой 8051.

ТАБЛИЦА 7

Тип	Макс. частота, МГц	ROM/ EPROM, байт	RAM, байт	Таймеры/ счетчики	Линии ввода/ вывода	Послед. каналы	АЦП, входы/ разр.	Другая периферия, особенности	Напряжение питания, В	Темп. диапазон, С	Корпус
P51XAG1x	30	8 К	512	3+WDT	32	2 UART	–		3,0...5,5	–40...+85	L44, Q44
P51XAG2x	30	16 К	512	3+WDT	32	2 UART	–		2,7...5,5	–40...+85	L44, Q44
P51XAG3x	30	32 К	512	3+WDT	32	2 UART	–		3,0...5,5	–40...+85	L44, Q44
P51XAC3x	25	32 К	1024	3+WDT	32	UART + CAN	–		2,7...5,5	–40...+85	D40, L44, Q44
P51XAS3x	25	32 К	1024	3+WDT+ PCA	80	2 UART + I <sup>2</sup> C	8x10 bit	ШИМ 6x8 bit, UPI,	2,7...5,5	–40...+85	Q80, Q100

Условные обозначения корпусов: D – DIP, L – LCC, Q – QFP, S – SSOP, SOIC.

Как мы видим, среди них имеются опции: FLASH, OTP-EPROM и ROM емкостью от 1 до 64 КБ и ROMLESS версии, способные адресовать до 64 КБ внешней памяти.

## 1.7. Микроконтроллеры фирмы SIEMENS

Фирма Siemens внесла несомненное своеобразие в развитие семейства 8051. Девизом фирмы можно, пожалуй, избрать: “Самые сложные микроконтроллеры 51-го семейства в мире”. Действительно, наряду с выпуском стандартных кристаллов, совместимых с 8051, 8052, Siemens разработала совершенно оригинальные модификации, нашедшие свое широкое применение в качестве промышленных high-end микроконтроллеров. Среди наиболее любопытных периферийных разработок Siemens:

- 10-разрядные АЦП;
- 6-канальный 10-разрядный ШИМ для управления электродвигателями постоянного тока;
- 29-канальный ШИМ;

- 8 DPTR;
- аппаратный блок умножения 16x16 и деления 32/16;
- блок загрузчика программной памяти (Bootstrap).

Совершенствуя технологию производства, фирма Siemens перевела все свои микроконтроллеры 51-го семейства, первоначально использовавшие лицензионное ядро фирмы Intel, на свое собственное ядро, после чего 51-е семейство Siemens сменило свое название и стало семейством C500. Перевод на новую технологию позволил повысить тактовые частоты микроконтроллеров, но ядро осталось, по сути, прежним. Все временные характеристики периферии и циклограммы шины адреса/данных полностью соответствуют характеристикам микроконтроллера 8051 фирмы Intel.

## **2. ЛАБОРАТОРНЫЕ РАБОТЫ**

### **2.1. Лабораторная работа № 1**

#### **Программирование микроконтроллера Intel8051**

Целью данной лабораторной работы является изучение

- системы команд,
- методов программирования микроконтроллера Intel8051,
- методов формирования временных задержек во встроенных микропроцессорных системах.

#### **2.1.1. Лабораторный стенд на базе микроконтроллера Intel8051**

Стенд (рис. 7) предназначен для изучения системы команд микроконтроллера семейства Intel 8051. Архитектура принципиальной схемы позволяет обрабатывать программы пользователя с использованием прерываний как на “чистой” схеме. Стенд оснащен наборным полем из 8 переключателей, служащих для ввода данных и полем из 8 светодиодов, предназначенных для вывода данных.

Стенд работает в двух режимах: режим загрузки и режим выполнения (отладки).

После сброса (кнопка “Reset”) стенд переходит в режим загрузки, в котором активизируется программа-загрузчик, находящаяся в ПЗУ. При этом ПЗУ (адреса 0 – 1FFFh) выполняет роль внешней памяти программ, а оперативное запоминающее устройство объемом 8 Кб подключено как внешняя память данных (адреса 0–1FFFh).

После загрузки программы пользователя в ОЗУ программа-загрузчик проводит аппаратное переконфигурирование системы, при этом ОЗУ становится внешней памятью программ, а в качестве внешней памяти данных используется набор индикаторов (запись в память по любому адресу) и переключателей (чтение памяти).

Переключение режимов работы производится набором мультиплексоров, управляемых сигналом Р 1.0. (Приложение 1, принципиальная схема стенда.)

Для вывода звуковой информации используется динамик, подключенный к линии D0 магистрали данных (как и индикатор D0 порта вывода).

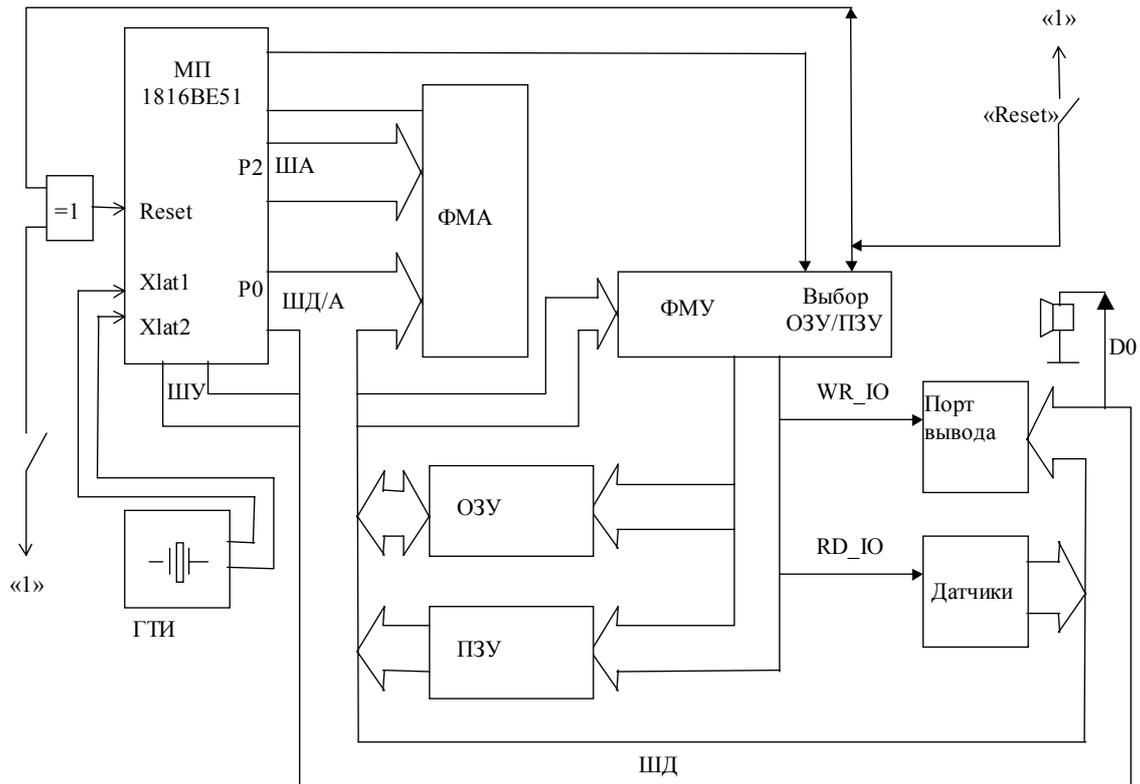


Рис. 7. Лабораторный стенд на базе микроконтроллера Intel 8051:

- ША – шина адреса (старшие разряды);
- ШД/А – шина данных / адреса (младшие разряды);
- ШД – шина данных;
- ГТИ – генератор тактовых импульсов;
- ФМА – формирователь адреса;
- ОЗУ – оперативное запоминающее устройство (8К);
- ПЗУ – постоянное запоминающее устройство (8К)

Пользователь создает программы в одном из текстовых редакторов, и затем, используя транслятор x8051.exe и редактор связей link.exe, получается файл с расширением HEX или TSK в зависимости от введенных параметров. Для упрощения работы рекомендуется использовать пакетный файл compile.bat, где в качестве параметра задается исходный файл (\*.A51), не указывая расширение. Например:

**compile.bat exampl**

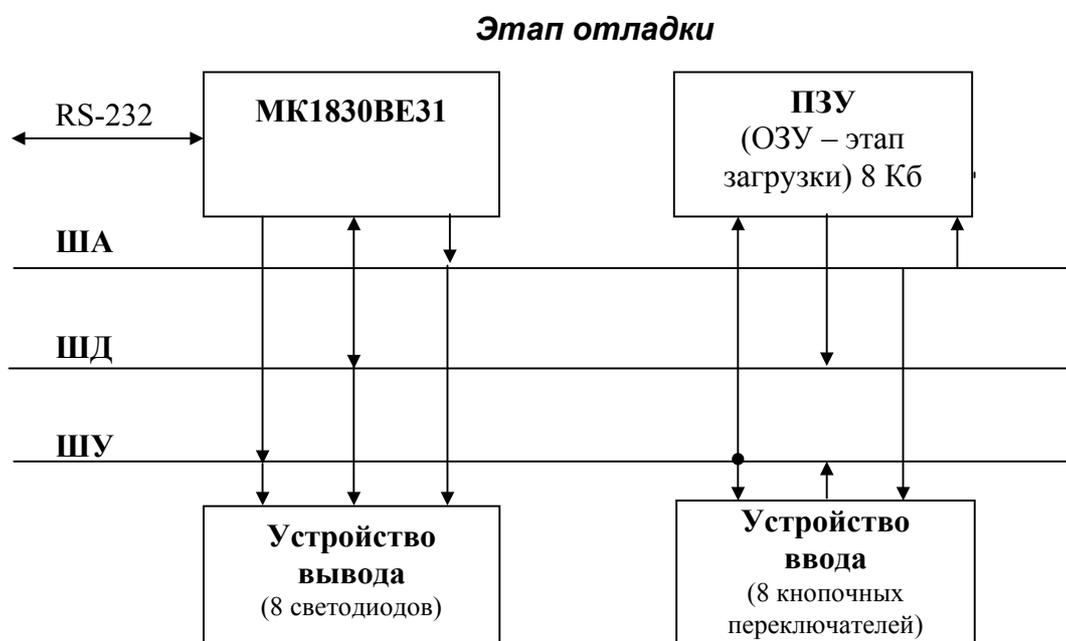
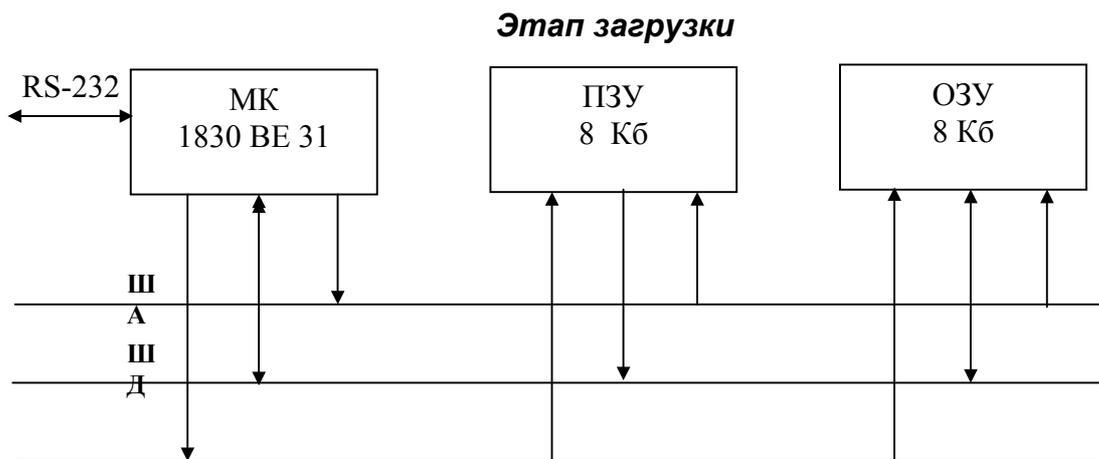


Рис. 8. Работа стенда на этапе загрузки/отладки

**Задание для домашней подготовки:**

- Изучить, пользуясь электронным учебником, структуру, систему команд, таймерную систему микроконтроллера Intel8051.
- Изучить структуру стенда, режима его работы (режим загрузки, режим отладки).
- Получить комплект заданий у преподавателя и написать программы на языке ассемблер.

## 2.1.2. Порядок выполнения лабораторной работы

### 1. Работа со стендом с использованием персонального компьютера.

Для загрузки программ в память стенда для изучения микроконтроллера 8051 используется программа BOOT51 со следующим синтаксисом:

**BOOT51** <имя загрузочного бинарного файла> /<номер последовательного порта>

Файл должен быть представлен в бинарном формате, предназначенном для записи в ПЗУ, и длиной, не превышающей 8192 байта. При правильном задании последовательного порта и корректном задании файла загрузка начнется при свечении на лицевой части стенда красного светодиода, означающего работу под управлением загрузчика, иначе следует произвести перезагрузку стенда путем нажатия <RESET> до появления свечения красного светодиода. После появления красного свечения произвести повторную загрузку. После загрузки управление передается программе пользователя путем аппаратного перенаправления сигналов управления внешними устройствами. Признаком исполнения является свечение зеленого светодиода. Для повторного запуска программы пользователя необходимо нажать кнопку <RESTART>.

**Пример:** Загрузить в ОЗУ стенда программу example.tsk.

Последовательность действий:

- Включить стенд.
- Проверить свечение *красного* светодиода на панели стенда, если он не светится, нажать на кнопку “Reset”.
- На персональном компьютере, предварительно подключенном к стенду с помощью последовательного (COM) интерфейса, перейти в каталог, содержащий программу **BOOT51.EXE**.
- Убедиться в наличии в этом каталоге файла **example.tsk**, в случае его отсутствия переписать файл в этот каталог.
- Запустить с программу BOOT51 со следующими параметрами:
  - **BOOT51.EXE example.tsk /2** (стенд подключен к компьютеру через последовательный порт 2) или
  - **BOOT51.EXE example.tsk /1** (стенд подключен к компьютеру через последовательный порт 1).

- Убедиться, что на панели стенда светится зеленый светодиод, что свидетельствует о работе пользовательской программы.

- В случае необходимости перезапустить программу с помощью нажатия кнопки “Restart” на панели стенда.

- Для окончания работы пользовательской программы нажать на кнопку “Reset” на панели стенда. Стенд перейдет под управление внутренней программы-монитора.

Возможные проблемы:

- BOOT51 выдал сообщение «Not found board BE51 on COM#» – проверьте подключение стенда и правильность указания номера последовательного порта;

- BOOT51 выдал сообщение «File not found» – проверьте правильность ввода имени файла. Не забудьте указать расширение файла;

- BOOT51 выдал сообщение «File too big» – размер файла превышает 8192 байта, файл не подлежит загрузке в ОЗУ стенда;

- BOOT51 выдал сообщение «File reading error» – произошла ошибка при чтении файла;

- BOOT51 выдал сообщение «Incorrect COM» – проверьте правильность указания номера последовательного порта;

- BOOT51 выдал сообщение «Load fault» – произошла ошибка при передаче данных от компьютера к стенду. Перезапустите стенд с помощью кнопки «Reset» и запустите программу BOOT51 еще раз;

- BOOT51 выдал сообщение «Incorrect command line» – проверьте правильность ввода строки параметров для программы BOOT51.

## ***2. Запустить на стенде разработанные дома программы и результат продемонстрировать преподавателю.***

### **ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ**

#### **Задача 1**

Прочитать число из порта ввода, прибавить к нему ненулевую константу и вывести на индикацию.

#### **Задача 2**

Прочитать число из порта ввода, проинвертировать и выдать на индикацию.

### **Задача 3**

В прочитанном из порта ввода числе переставить биты в обратном порядке. Результат вывести на индикаторы.

### **Задача 4**

Сложить два числа, находящихся в младших и старших 4-х разрядах порта ввода, и результат:

- 1) выдать на индикацию;
- 2) выдать на индикацию в двоично-десятичном коде.

### **Задача 5**

Написать программу вычисления количества единичных битов в числе, прочитанном из порта ввода (результат вывести на индикацию).

### **Задача 6**

Посчитать модуль разности количества единичных бит, стоящих на четных местах порта ввода, и количества единичных бит на нечетных местах порта ввода. Полученный результат выдать на индикатор.

### **Задача 7**

Получить два числа как старшие и младшие 4 разряда порта ввода. Если оба числа не превосходят 8, то посчитать их произведение, а если хотя бы одно число больше 8, посчитать их сумму. Результат вывести на индикатор в двоично-десятичном коде.

### **Задача 8**

Произвести преобразование числа, прочитанного из порта ввода по следующему закону перестановки битов:

- 0 → 3
- 1 → 2
- 2 → 4
- 3 → 7
- 4 → 0
- 5 → 6
- 6 → 1
- 7 → 5.

### **Задача 9**

Посчитать сумму чисел от 1 до заданного переключателями (порт ввода). Выдать результат на индикацию, если он не превышает 255, иначе выдать число 255.

### **Задача 10**

С использованием таймера-счетчика реализовать подпрограмму задержки в миллисекундах и проверить на примере “бегущей лампочки”.

### **Задача 11**

Написать программу счетчика секунд с использованием программируемого таймера-счетчика (текущее значение счетчика выводить на индикацию).

### **Задача 12**

Написать программу-секундомер, функционирующую следующим образом: счетчик секунд выводится на разряды 0–5 индикатора, а счетчик минут выводится на разряды 6–7 индикатора. Пуск/сброс секундомера осуществляется управляющим разрядом 0 порта ввода. Разряд 1 порта ввода используется для приостановки (паузы) счета.

### ***Контрольные вопросы***

1. Опишите базовую архитектуру семейства MCS-51 (кратко).
2. Опишите особенность структуры памяти семейства MCS-51.
3. Дайте краткий обзор микроконтроллеров семейства MCS-51, MCS-251, MCS-151 фирмы Intel.
4. Опишите режимы работы микроконтроллеров (Power Down, Idle).
5. Опишите организацию стандартных портов ввода/вывода микроконтроллеров семейства MCS-51.
6. Опишите таймеры микроконтроллеров семейства MCS-51.
7. Опишите работу канала последовательного ввода/вывода микроконтроллеров семейства MCS51.
8. Опишите стандартную систему прерываний микроконтроллеров семейства MCS-51.
9. Дайте краткий обзор системы команд микроконтроллеров семейства MCS-51.
10. Опишите организацию временных задержек без использования таймера.
11. Опишите организацию временных задержек с использованием таймера.
12. Опишите структуру и принцип работы лабораторного стенда по изучению микроконтроллеров семейства MCS-51.
13. Организация временных задержек с (без) использованием таймера.

## **2.2. Лабораторная работа № 2**

### **Построение однопроводной сети 1 WIRE и организация обмена информации с iBUTTON DS199x с использованием МК DS5000T**

Цель лабораторной работы:

Ознакомиться с интерфейсом 1 WIRE, научиться организовывать обмен данными с iBUTTON DS1992.

#### **2.2.1. Лабораторный стенд для изучения МПС на базе МК DS 5000T**

Стенд для изучения микроконтроллера DS5000T состоит из соединителя для DS5000T, контакта для подключения Touch Memory DS1992, согласователя уровней для последовательного порта и блока питания.

Для согласования уровней используется микросхема DS232A. Она включает в себя преобразователи уровней и источники напряжения для выработки управляющих уровней интерфейса RS-232. Один из преобразователей уровня используется для инициализации встроенного загрузчика DS5000T по сигналу DTR от компьютера.

Для подключения Touch Memory используется один из выводов двунаправленного порта с открытым стоком P 0.0. Кроме этого, выходы портов 0 и 1, а также входы встроенных таймеров и запроса прерывания выведены на 25-контактный разъем для подключения внешних устройств.

#### **2.2.2. Программное обеспечение для работы со стендом**

Управление стендом со стороны компьютера осуществляется программой KIT5K. Она поставляется вместе с отладочным стендом для DS5000T. KIT5K – среда, обеспечивающая интерфейс высокого уровня для загрузки прикладных программ в память микроконтроллера или установки его конфигурации при выполнении команды PROGRAM. Также она обеспечивает несколько команд, позволяющих непосредственно управлять ресурсами микроконтроллера. Они позволяют прямую установку регистра MCON, загрузку 40-битового ключа шифрования, установку и очистку замка защиты. Программа может просматривать, дизассемблировать и

изменять содержимое памяти микроконтроллера.

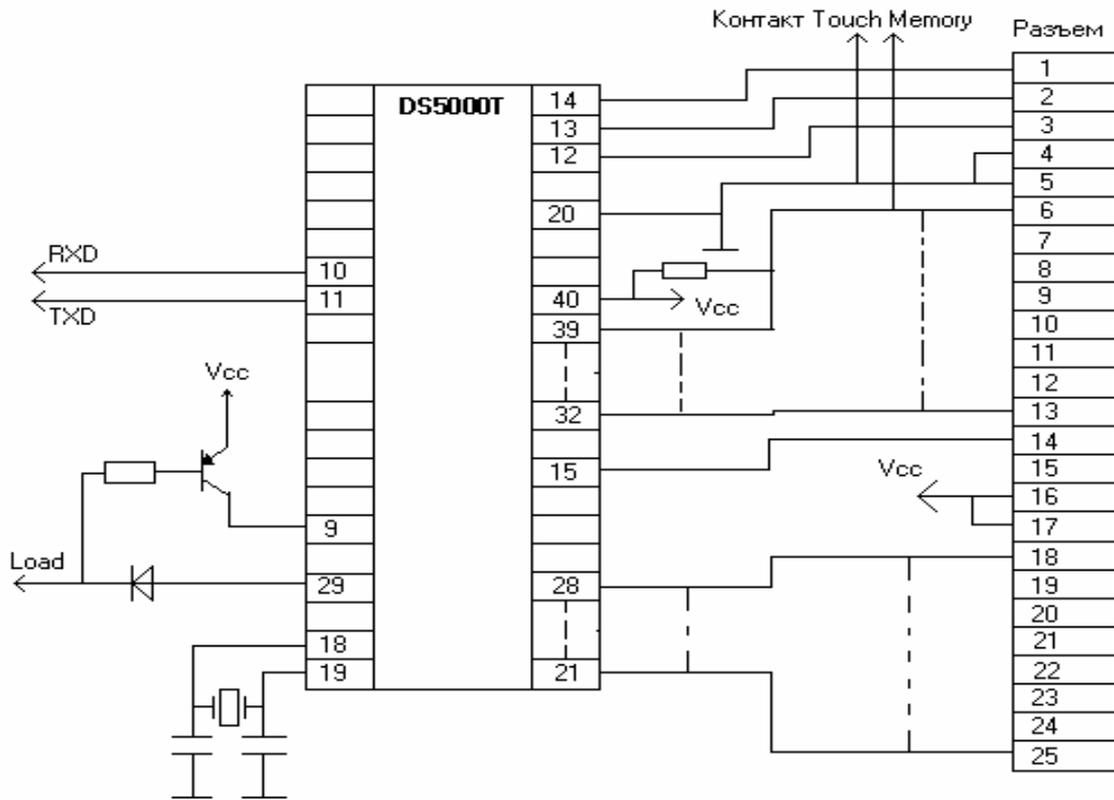


Рис. 9. Коннектор для DS5000T

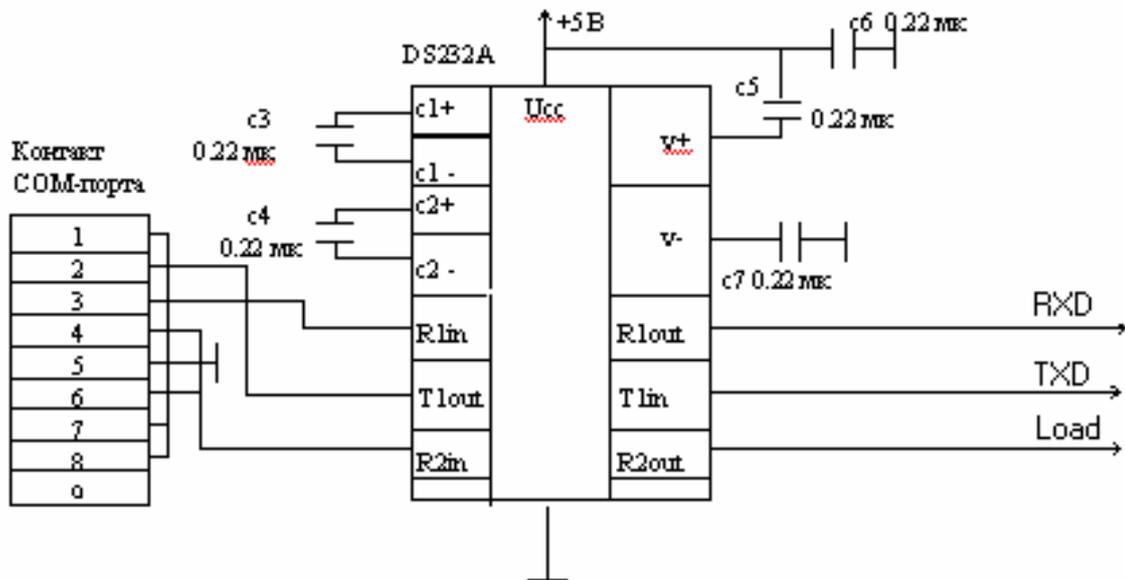


Рис. 10. Согласователь уровней

Для обмена данными между прикладной программой и компьютером в КИТ5К встроен простой режим терминала. В этом режиме данные с клавиатуры посылаются на последовательный порт и на экране выводятся данные, полученные с порта. Команды режима терминала представлены в табл. 8.

После запуска программы на экране появляется подсказка для ввода команд. КИТ5К может работать либо в диалоговом режиме, либо в пакетном режиме. Список команд представлен в табл. 9.

Загрузка программ в микроконтроллер осуществляется из файлов в Intel HEX формате. Исходный текст программы на ассемблере набирается в любом текстовом редакторе. Для преобразования его в HEX-файл необходимо следующее программное обеспечение: ассемблер ASM51 (или другой для микроконтроллера 8051), редактор связей RL51 (для объединения объектных файлов, содержащих части программы) и преобразователь формата ОН (Object to Hex, преобразует объектный файл в файл HEX формата).

ТАБЛИЦА 8

Команды режима терминала

capture	Разрешение сохранения полученных данных в файле.
kit	Переключение в режим отладчика.
posnow	Разрешение контроля за «снегом» для CGA монитора.
send	Послать указанный файл микроконтроллеру по последовательному порту.
snow	Запрещение контроля за «снегом» для CGA монитора.

**Примечание:** команды cd, cls, com, dir, dos, dtr, exit, help, logoff, logto, pgmode, quit, run, speed, type доступны в обоих режимах.

## Краткий список команд КИТ5К

cd	Переход в другой каталог или вывод названия текущего.
cls	Очистка экрана.
com	Назначение СОМ-порта, к которому подключен стенд.
dir	Вывод списка файлов в текущем каталоге.
display	Распечатка содержимого ОЗУ в виде чисел.
do	Выполнение списка команд из файла.
dos	Временный выход в DOS.
dtr	Переключение сигнала DTR между режимами загрузки и работы.
dump	Вывод содержимого ОЗУ в файл в Intel HEX формате.
edit	Редактирование памяти.
exit	Выход в DOS.
fill	Заполнение памяти константой.
help	Описание функций и синтаксиса команд КИТ5К.
key	Загрузка ключа шифрования.
load	Загрузка памяти из файла Intel HEX формата.
lock	Установить замок защиты.
logoff	Отключение ведения протокола выполняемых команд.
logto	Ведение протокола команд в указанном файле.
mcon	Ввод значения в регистр MCON.
partition	Установка разбиения памяти.
pgmode	Включение режима загрузки программ.
program	Автоматическая загрузка программ с использованием файла конфигурации.
quit	Выход в DOS.
range	Установка размера памяти.
Run	Запуск DS2250T для выполнения программы.
Speed	Установка скорости обмена данными.
Status	Вывод состояния на экран.
Term	Включение режима терминала.
Type	Вывод на экран указанного файла.
Unasm	Дизассемблирование содержимого памяти.
Unlock	Выключение замка защиты.
Verify	Сравнение Intel HEX файла и содержимого памяти.

### 2.2.3. Модуль для изучения Touch Memory DS1992

Touch Memory DS1992 представляет собой удобный носитель небольших объемов информации. Внешне это небольшой корпус в виде металлической таблетки с двумя контактами. Внутри его находится литиевая батарейка, ОЗУ, ПЗУ и схема управления. Обратиться к DS1992 можно, используя однопроводной интерфейс,

для работы которого необходимы только два контакта: земля и вывод данных. Передача данных по однопроводному интерфейсу осуществляется заданием временных интервалов сигналов на линии данных. Протокол обмена данными выглядит следующим образом:

1. Посылка импульса сброса устройством, контролирующим линию, и ожидание импульса присутствия, сигнализирующего о наличии на линии устройств.

2. Передача одной из четырех команд управления ПЗУ. В ПЗУ зашит идентификационный номер устройства, выжигаемый лазером при изготовлении. Он состоит из трех частей:

- 8 бит: идентификатор типа устройства. Для DS1992 он равен 08h.
- 48 бит: порядковый номер устройства. Не повторяется, т. е. уникален для каждого устройства.
- 8 бит: контрольный код, вычисляемый по алгоритму CRC.

Для управления ПЗУ используется одна из четырех команд:

- 1) читать ПЗУ: код 33h;
- 2) подобрать ПЗУ: код 55h;
- 3) искать ПЗУ: код F0h;
- 4) пропустить ПЗУ: код CCh.

После того как правильно выполнена одна из команд ПЗУ, DS1992 ожидает команду управления ОЗУ. ОЗУ состоит из 256-битного буфера данных и 1024 бит основной памяти. Сохранность ее содержимого обеспечивается литиевой батареей в течение 10 лет. Команды управления памятью обеспечивают запись данных в буфер, их проверку и копирование в основную память. Их список выглядит следующим образом:

- 1) записать в буфер; код 0Fh;
- 2) читать из буфера; код AAh;
- 3) копировать содержимое буфера в память; код 55h;
- 4) читать из памяти; код F0h.

После выполнения команды ОЗУ устройство ждет импульса сброса для начала нового цикла обмена данными.

Необходимые временные диаграммы обмена данными по однопроводному интерфейсу реализуются на однокристалльной микроЭВМ DS5000T. Ее двунаправленный порт ввода/вывода P0 имеет выводы с открытым стоком, к одному из которых подключается линия данных однопроводного интерфейса. Управляющая программа

для DS2250T обеспечивает прием байта данных по последовательному порту от компьютера, пересылку его на DS1992, прием ответного байта и передачу его компьютеру. Прием байта FEh вызывает посылку импульса сброса, определения наличия устройств на линии и передачу компьютеру результата: код символа “D” если на линии есть DS1992, или код символа “F” если устройств на линии нет.

Со стороны компьютера стендом управляет программа, выполняющая последовательность команд обмена данными. Эта программа определяет, к какому COM порту подключен стенд и подключена ли к нему Touch Memory. Для этого она посылает команду сброса и считывает данные с COM порта. Если прочитанный байт не равен коду символа “D”, то программа выдает сообщение об ошибке. Если программа получает байт кода “D”, то она посылает команду “читать ПЗУ”. Считав 8 байт ПЗУ, программа посылает команду “читать из памяти” и считывает 128 байт содержимого памяти. Затем выполняется команда сброса Touch Memory и последовательность команд записи в память, то есть команды “пропустить ПЗУ” и “запись в буфер”, затем опять “пропустить ПЗУ” и “копировать буфер в память”.

#### **2.2.4. Модуль для изучения адресуемых переключателей DS2405**

Характеристики переключателя DS2405:

- 1) Открытый сток (выход) управляется личным 64-битным регистрационным номером прибора.
- 2) Логический уровень вывода открытого стока определяется по однопроводной шине.
- 3) Нагрузочная способность выхода более 4 мА на 0,4 В.
- 4) Несколько DS 2405, подключенных к одной шине, могут быть идентифицированы и включены либо выключены независимо от других устройств, подключенных к этой же шине.
- 5) Уникальный 64-битный регистрационный номер (8 бит – код семейства, 48 бит – серийный номер, 8 бит – CRC-код) позволяет абсолютную идентификацию (нет двух приборов с повторяющимися номерами).
- 6) Сведение управления, адресации, информации и питания на один вывод (данные).
- 7) Прямое присоединение к одиночному выводу порта микропроцессора и связь со скоростью 16,3 Кбит/с.

8) Код семейства (8 бит) задает связь с устройством типа DS 2405.

9) CRC код (8 бит) гарантирует безошибочный выбор.

Адресуемый переключатель DS 2405 – транзистор с N канальным открытым стоком, который управляется по одному проводу. Он может быть открыт или закрыт подачей 64-битного идентификационного номера. Он включает в себя:

8-бит – идентификационный код устройства (для DS2405 он равен 05h).

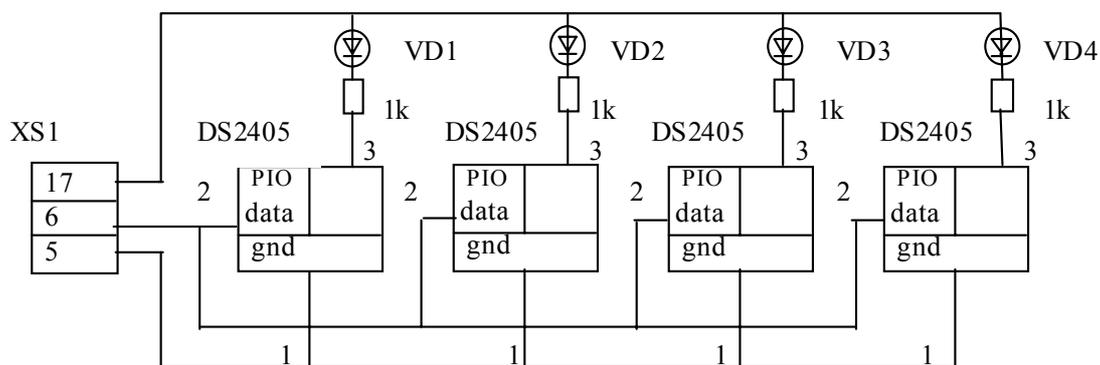


Рис. 11. Принципиальная электрическая схема модуля DS2405:  
XS1 – разъем стенда для внешних устройств VD1 – VD4 АЛ307

Управление модулем осуществляется программой, включающей в себя основные процедуры для реализации протокола однопроводного интерфейса. Она обеспечивает последовательный перебор номеров устройств, присутствующих на шине, включение и выключение устройства с текущим номером. При включении и выключении программа посылает номер устройства на последовательный порт и ожидает прихода данных на приемник последовательного порта для переключения к следующему устройству. Для отправки и приема данных используется режим терминала отладчика КИТ5К. Таким образом можно узнать адреса всех устройств на шине.

### 2.2.5. Работа со стендом для изучения микроконтроллеров Dallas Semiconductor

Стенд для изучения микроконтроллеров Dallas Semiconductor представляет собой микроконтроллер DS5000-16-32 с дополнительными внешними элементами, предназначенными для связи с компьютером и дополнительными устройствами.

Для работы со стендом необходим IBM-совместимый компьютер класса i486 и выше со свободным последовательным портом и установленным на нем программным обеспечением (ПО) для компиляции, загрузки и отладки программ.

Список необходимого ПО:

- ASM51.exe – ассемблер для 8051 микроконтроллеров. Используется для компиляции программ. Использование: ASM51.exe source.asm.

- Oh.exe – конвертор \*.obj-файлов, созданных ассемблером. Создает файлы в Intel Hex формате \*.hex. Использование oh.exe source, obj.

- Kit5k.exe – загрузчик программ в DS5000. Работает в командной строке.

Команды:

Corn n – указывает номер порта для обмена со стендом. Должна быть выполнена первой после запуска kit5k.

PGMODE – включает режим загрузки программ на стенде.

RUN – запуск программ.

PR имя\_файла.hex – загрузка программы.

Disp – распечатка дампа памяти.

UNASM address – дизассемблирование содержимого памяти контроллера.

Quit – выйти из программы.

Для работы со стендом необходимо подключить его к COM-порту компьютера с помощью нуль-модемного кабеля (9-контактный разъем на стенде). Блок питания стенда должен быть отключен от сети. Если для работы программы требуются дополнительные внешние модули, то их необходимо подключить к 25-контактному коннектору на стенде. После этого необходимо включить блок питания. Дальнейшая работа производится на компьютере. Последовательность действий:

1. Создать текст программы с помощью текстового редактора. Расширение файла – asm.

2. Откомпилировать программу с помощью ASM51. При отсутствии сообщений об ошибках он создаст файл с расширением obj.

3. Конвертировать obj-файл в Intel Hex формат с помощью программы Oh.exe (должен создаваться файл с расширением hex).

4. Запустить программу Kit5t.exe.

5. Ввести команду Com n, где n номер порта, который вы использовали для подключения стенда.

6. Для проверки работоспособности стенда наберите:

- PGMODE(Enter)
- DISP{Enter}

Если стенд работает должным образом, будет выведен дамп памяти. Иначе появится сообщение о невозможности обмена данными со стендом. При возникновении этого сообщения необходимо проверить работоспособность COM-порта и надежность соединения шлейфа со стендом.

7. Загрузить программу в память микроконтроллера с помощью команды PR.

8. Запустить программу командой RUN.

Пример:

```
Asm51.exe source.asm
```

```
Oh.exe source.obj
```

```
Kit5k.exe
```

```
kit:Com 1
```

```
kit: PR source.hex
```

```
kit:run
```

```
kit:quit
```

### **2.2.6. Порядок выполнения лабораторной работы**

1. Написать программу записи в iButtom DS1992 визитки студента.

2. Написать программу, обеспечивающую режим бегущих огней на светодиодах VD1VD4, подключенных к однопроводному интерфейсу посредством адресуемых переключателей DS2405.

Примечание: при написании программ использовать описанное выше программное обеспечение (ASM51.exe., Kit5k.exe).

## **2.3. Лабораторная работа № 3**

### **Программирование стандартных интерфейсов микроконтроллера на примере 8X552 фирмы PHILIPS**

Цель работы:

- Познакомиться с микроконтроллерами MCS51 фирмы PHILIPS, научиться программировать приборы, подключаемые по интерфейсу I<sup>2</sup>C.

- Научиться подключать и программировать ЖКИ индикатор.

### 2.3.1. Краткие сведения из теории

#### Схемы семейства MCS51 фирмы PHILIPS. Обзор 8X552

Схема 8X552 представляет собой высокопроизводительный МК, разработанный для применений в реальном масштабе времени, таких, как измерительная аппаратура, управление производством, автоматизированное управление (например, управление станками). В дополнение к стандартным функциям процессора 80C51, прибор обеспечивает ряд специальных аппаратных функций для этих применений (рис. 12).

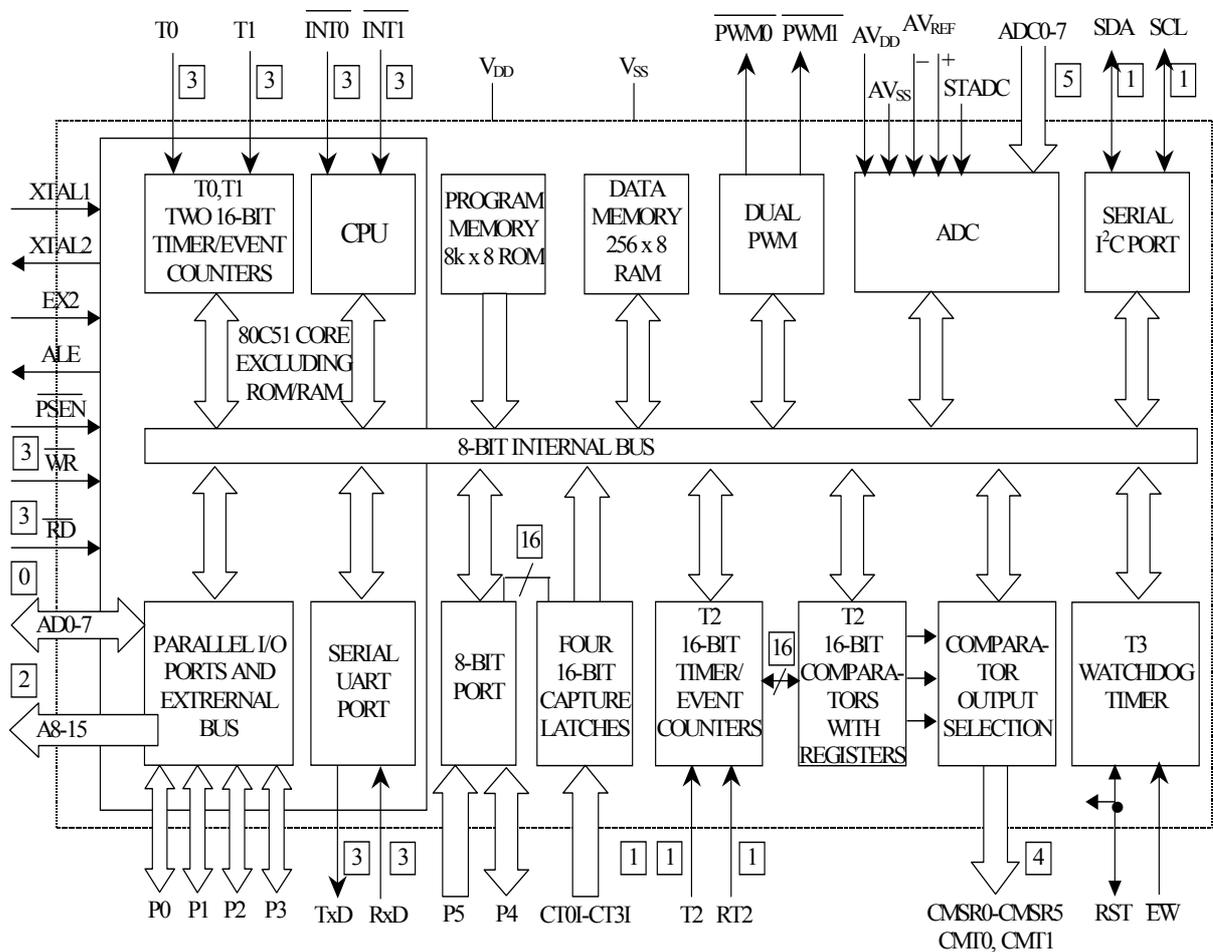


Рис. 12. Структурная схема МК 80X552

Однокристалльный 8-разрядный МК 80X552 изготавливается по усовершенствованной CMOS-технологии и является производным семейства микроконтроллеров 80C51. 8X552 использует мощную

систему команд процессора 80C51. Дополнительные специальные регистры помещены на кристалле для управления периферийными устройствами, находящимися на кристалле. Существуют три версии производимых схем:

- 1) 83C552: ПЗУ с масочным программированием на 8 Кб, ОЗУ на 256 байт;
- 2) 87C552: ЭППЗУ на 8 Кб, ОЗУ на 256 байт;
- 3) 80C552: версия 83C552 без ПЗУ.

80X552 содержит:

- энергозависимую память только для считывания программ;
- энергозависимую память 256x8 для считывания/записи данных;
- два 16-разрядных таймера/счетчика событий (идентичных таймерам серии 80C51);
- дополнительный 16-разрядный таймер, соединенный с защелками сбора данных и сравнения;
- структуру вложенных прерываний с 15 источниками и двумя уровнями приоритета;
- 8-входовый АЦП;
- двойной интерфейс ЦАП на основе ШИМ;
- два последовательных UART;
- таймер «будильника»;
- тактовый генератор.

Для систем, требующих расширенных возможностей, схема 8X552 может быть расширена, используя стандартные TTL-совместимые ЗУ и логику.

Рассмотрим структуру памяти микроконтроллера 80C552 (рис. 13):

1) Память программ (ПП). 8X552 содержит на кристалле ПП емкостью 8Кб, которая может быть расширена внешней до 64 Кб. Когда на выводе EA поддерживается высокий уровень, 8X552 вызывает команды из внутреннего ПЗУ, если адрес не превышает 1FFFh. Адреса от 2000h до FFFFh вызываются из внешней ПП. Когда на выводе EA поддерживается низкий уровень, все команды вызываются из внешней ПП. Ячейки ПЗУ с адресом от 0003h до 0073h используются ISR (программами обслуживания прерываний).

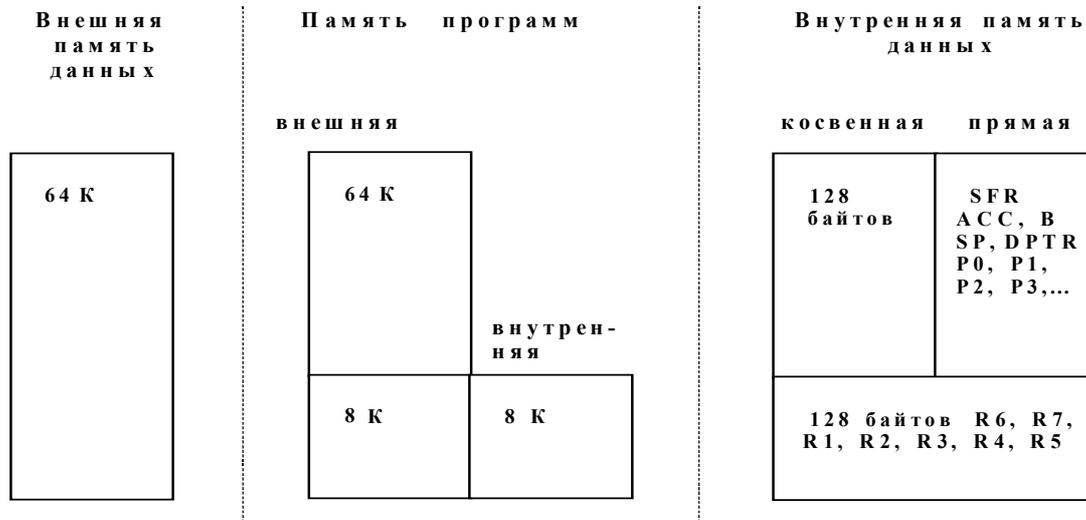


Рис. 13. Структура памяти МК 80C522

2) Память данных (ПД). Внутренняя ПД делится на три секции: младшие 128 байт ОЗУ, старшие 128 байт ОЗУ и 128-байтная область SFR. Младшие 128 байт ОЗУ адресуются прямо и косвенно. Пока ячейки ОЗУ от 128 до 255-й и SFR занимают одно и то же адресное пространство, они выбираются через различные режимы адресации. Позиции ОЗУ от 128 до 255-й адресуются только косвенно, а SFR – только прямо. Все другие аспекты внутреннего ОЗУ идентичны 80C51.

Стек может располагаться в любом месте во внутреннем ОЗУ путем загрузки 8-разрядного указателя стека. Глубина стека максимум 256 байт.

3) SFR (Special Function Registers). SFR (только прямо адресуемые) содержат все регистры 8X552, за исключением PC и четырех групп (банков) регистров. Большинство из 56 SFR используются для управления периферийными устройствами на кристалле. Стандартные SFR 80C51 представлены и функционируют идентично процессору 8X552, за исключением некоторых случаев, обсуждение которых выходит за рамки данной работы. Описание регистров и назначение отдельных битов приведено в табл. 10.

80X552 Special Function Registers

SYMBOL	DESCRIP-TION	DI-RECT ADD-RESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB				LSB				
ACC*	Accumulator	E0H	E7	E6	E5	E4	E3	E2	E1	E0	00H
ADCH#	A/D converter high	C6H									xxxxxxxB
ADCON#	Adc control	C5H	ADC.1	ADC.0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0	xx000000B
B*	B register	F0H	F7	F6	F5	F4	F3	F2	F1	F0	00H
CTCON#	Capture control	EBH	CTN3	CTP3	CTN2	CTP2	CTN1	CTP1	CTN0	CTP0	00H
TH3S#	Capture high 3	CFH									xxxxxxxB
CTH2#	Capture high 2	CEH									xxxxxxxB
CTH1#	Capture high 1	CDH									xxxxxxxB
CTH0#	Capture high 0	CCH									xxxxxxxB
CMH2#	Compare high 2	CBH									00H
CMH1#	Compare high 1	CAH									00H
CMH0#	Compare high 0	C9H									00H
CTL3#	Capture low 3	AFH									xxxxxxxB
CTL2#	Capture low 2	AEH									xxxxxxxB
CTL1#	Capture low 1	ADH									xxxxxxxB
CTL0#	Capture low 0	ACH									xxxxxxxB
CML2#	Compare low 2	ABH									00H
CML1#	Compare low 1	AAH									00H
CML0#	Compare low 0	A9H									00H
DPTR: DPH DPL	Data pointer (2 bytes) Data pointer high Data pointer low	83H 82H									00H 00H
IEN0*#	Interrupt enable 0	A8H	AE	AD	AC	AB	AA	A9	A8		
			EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0	00H
			EF	EE	ED	EC	EB	FA	E9	E8	
IEN1*#	Interrupt enable 1	E8H	ET2	ECM2	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0	00H
			BF	BE	BD	BC	BB	BA	B9	B8	
IP0*#	Interrupt priority 0	B8H	-	PAD	PS1	PS0	PT1	PX1	PT0	PX0	x0000000B
			FF	FE	FD	FC	FB	FA	F9	F8	
IP1*#	Interrupt priority 1	F8H	PT2	PCM2	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0	00H
P5#	Port 5	C4H	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	xxxxxxxB
			C7	C6	C5	C4	C3	C2	C1	C0	
P4#	Port 4	C0H	CMT1	CMT0	CMSR5	CMSR4	CMSR3	CMSR2	CMSR1	CMSR0	FFH
			B7	B6	B5	B4	B3	B2	B1	B0	
P3*	Port 3	B0H	-RD	-WR	T1	T0	-INT1	-INT0	TXD	RXD	FFH
			A7	A6	A5	A4	A3	A2	A1	A0	
P2*	Port 2	A0H	A15	A14	A13	A12	A11	A10	A9	A8	FFH
			97	96	95	94	93	92	91	90	
P1*	Port 1	90H	SDA	SCL	RT2	T2	CT3I	CT2I	CT1I	CT0I	FFH
			87	86	85	84	83	82	81	80	
P0*	Port 0	80H	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	FFH
PCON#	Power control	87H	SMOD	-	-	WLE	GF1	GF0	PD	IDL	00xx0000B
			D7	D6	D5	D4	D3	D2	D1	D0	
PSW*	Program status word	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	00H

\* SFRs are bit addressable.  
# SFRs are modified from or added to the 80C51 SFRs.

ТАБЛИЦА 10 (окончание)

SYMBOL	DESCRIPTION	DIRECT ADDRESS	BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION								RESET VALUE
			MSB				LSB				
PWMP# PWM1# PWM0#	PWM prescaler PWM register 1 PWM register 0	FEH FDH FCH									00H 00H 00H
RTE#	Reset/toggle enable	EFH	TP47	TP46	RP45	RP44	RP43	RP42	RP41	RP40	00H
SP	Stack pointer	81H									07H
SOBUF	Serial 0 data buffer	99H									xxxxxxxxB
			9F	9E	9D	9C	9B	9A	99	98	
SOCON*	Serial 0 control	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
S1ADR*	Serial 1 address	DBH	SLAVE ADDRESS							GC	00H
SIDAT*	Serial 1 data	DAH									00H
S1STA*	Serial 1 status	D9H	SC4	SC3	SC2	SC1	SC0	0	0	0	F8H
			DF	DE	DD	DC	DB	DA	D9	D8	
SICON#*	Serial 1 control	D8H	CR2	ENS1	STA	ST0	SI	AA	CR1	CR0	00H
STE#	Set enable	EEH	TG47	TG46	SP45	SP44	SP43	SP42	SP41	SP40	COH
TH1	Timer high 1	8DH									00H
TH0	Timer high 0	8CH									00H
TL1	Timer low 1	8BH									00H
TL0	Timer low 0	8AH									00H
TMH2#	Timer high 2	EDH									00H
TML2#	Timer low 2	ECH									00H
TMOD	Timer mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
			8F	8E	8D	8C	8B	8A	89	88	
TCON*	Timer control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TM2CON*	Timer 2 control	EAH	T2IS1	T2IS0	T2ER	T2B0	T2P1	T2P0	T2MS1	T2MS0	00H
			CF	CE	CD	CC	CB	CA	C9	C8	
TM2IR#*	Timer 2 int flag reg	C8H	T20V	CMI2	CMI1	CMI0	CTI3	CTI2	CTI1	CTI0	00H
T3#	Timer 3	FFH									00H

\* SFRs are bit addressable.

# SFRs are modified from or added to the 80C51 SFRs.

		7	6	5	4	3	2	1	0	
IEN1 (E6H)		ET2	ECM2	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0	
	(MSB)									(LSB)
		BIT		SYMBOL		FUNCTION				
	IEN1.7	ET2	Enable Timer T2 overflow interrupt(s)							
	IEN1.6	ECM2	Enable T2 Comparator 2 interrupt							
	IEN1.5	ECM1	Enable T2 Comparator 1 interrupt							
	IEN1.4	ECM0	Enable T2 Comparator 0 interrupt							
	IEN1.3	ECT3	Enable T2 Capture register 3 interrupt							
	IEN1.2	ECT2	Enable T2 Capture register 2 interrupt							
	IEN1.1	ECT1	Enable T2 Capture register 1 interrupt							
	IEN1.0	ECT0	Enable T2 Capture register 0 interrupt							

Figure 2. Timer T2 Interrupt Enable Register (IEN1)

Таким образом, однокристалльный микроконтроллер Philips PCB 80C552 является развитием семейства INTEL 8051. К числу отличительных особенностей микроконтроллера Philips PCB 80C552, важных с точки зрения использования устройства при решении данной задачи, следует отнести:

- наличие встроенного АЦП с возможностью работы в 8- или 10-разрядном формате преобразования;
- увеличенное число стандартных цифровых портов ввода/вывода (5);
- наличие (как и в устройстве INTEL 8051) полного дуплексного последовательного порта ввода/вывода (SIO0) с использованием внутреннего таймера микроконтроллера (Т1) в качестве генератора скорости передачи.

Микроконтроллер имеет 8 аналоговых входов, организованных как 8-разрядный порт P5. Имеются также встроенные программно-управляемый мультиплексор аналоговых сигналов “8 в 1” и устройство выборки-хранения.

Устройство выборки-хранения микроконтроллера обеспечивает достоверность младшего разряда выходного кода АЦ-преобразователя при скорости изменения входного аналогового сигнала до 10 В/мс.

Диапазон напряжений допустимых входных аналоговых сигналов АЦП микроконтроллера определяется путем задания потенциалов  $V_{ref-}$  (нижняя граница) и  $V_{ref+}$  (верхняя граница). При этом должно выполняться неравенство:  $0 \leq (V_{ref-}) < (V_{ref+}) \leq (V_{ss})$ , где  $V_{ss}$  – напряжение питания микроконтроллера (+5В).

Заданные при отладке контроллера значения  $(V_{ref-}) = 0В$  и  $(V_{ref+}) = +5В$  обеспечивают максимальный диапазон входных аналоговых сигналов АЦП.

Из пяти портов микроконтроллера для цифрового ввода/вывода полностью доступны 8-разрядные порты P1 и P4, а также старшие 6 разрядов порта P3 – таким образом, всего 22 линии цифрового ввода/вывода. Путем выбора режима работы таймера Т1 при частоте тактового генератора микроконтроллера 11,059 МГц скорость последовательного асинхронного обмена может быть с высокой степенью точности задана равной стандартным значениям скорости обмена интерфейса RS-232C (1200, 2400, 4800, 9600 и 19200 Бод/с).

Примечание: Диапазон рабочих температур

- РСF80C552 – (–40...+85 °С);
- РСВ80C552 – (0...+70 °С).

### 2.3.2. Описание лабораторного стенда

Структурная схема лабораторного стенда приведена на рис. 14.

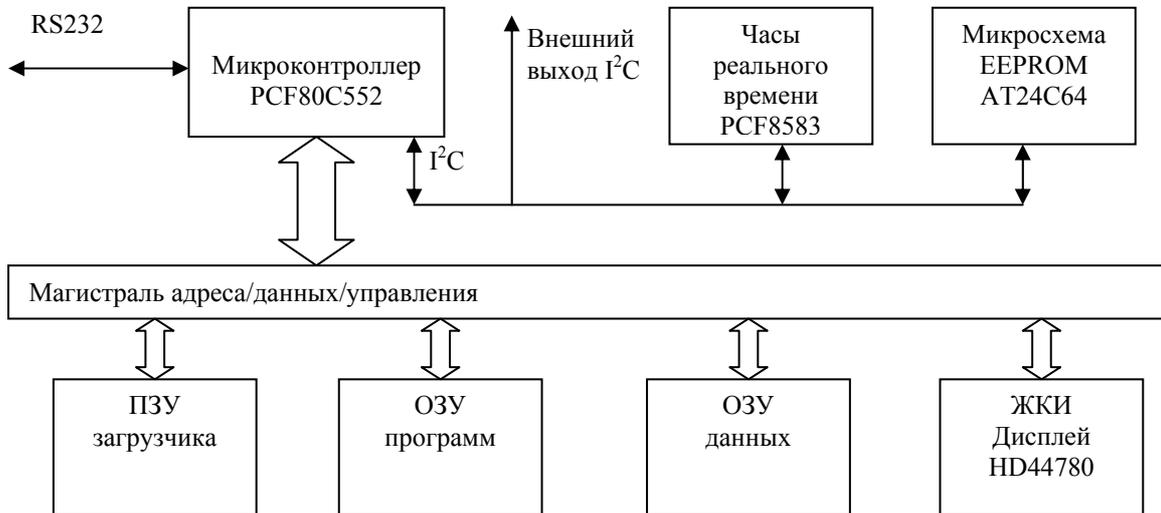


Рис. 14. Структурная схема лабораторного стенда EB-552

Отладочная плата EB-552 предназначена для макетирования устройств на базе микро-ЭВМ семейства MCS-51. Плата содержит микроконтроллер PCF80C552 фирмы Philips, установленный в панельку, и имеет возможность установки других микро-ЭВМ семейства MCS-51 в корпусах DIP-40 со стандартным расположением выводов. Непосредственно на плате располагается эмулятор ПЗУ, позволяющий с помощью входящей в комплект программы-загрузчика загружать из компьютера отлаживаемую программу в формате Intel HEX через интерфейс RS-232.

Помимо этого, плата содержит алфавитно-цифровой ЖКИ-модуль 24x2 или 20x4, энергонезависимые часы реального времени с интерфейсом I<sup>2</sup>C на базе микросхемы PCF8583, микросхему EEPROM AT24C64, а также разъем RJ-11 для подключения внешних устройств с шиной I<sup>2</sup>C. Все сигналы внутренней шины выведены вблизи макетного поля платы.

EB-552 – недорогое отладочное средство, позволяющее макетировать и отлаживать устройства на базе стандартных микро-ЭВМ семейства MCS-51 и усовершенствованного члена семейства MCS-51 80C552 фирмы Philips. Эта микро-ЭВМ программно и аппаратно совместима с 80C31 и в дополнение предоставляет 10-разрядный 8-канальный АЦП, аппаратную поддержку шины I<sup>2</sup>C (режимы Slave и Master), Watchdog таймер и дополнительные линии ввода/вывода.

Основной особенностью платы EB-552, обуславливающей удобство ее применения, является наличие непосредственно на плате эмулятора ПЗУ, позволяющего загружать отлаживаемую программу из компьютера и запускать на исполнение, а с помощью встроенного ЖКИ-модуля производить контроль результатов.

В состав платы входят: установленная в панельку микро-ЭВМ PCF80C552, панелька под установку микро-ЭВМ семейства MCS-51 в стандартных корпусах DIP-40 (для использования микро-ЭВМ в DIP-корпусах необходимо изъять из платы PCF80C552 и наоборот), 32 К ОЗУ, 32 К ОЗУ эмулятора, панелька под установку микросхем ПЗУ типа 27C256 (при поставке в нее установлена микросхема, содержащая программу-загрузчик, для работы платы в режиме отладки), ЖКИ-модуль 24x2 или 40x4 на базе контроллера HD44780, подключенный к системной шине микро-ЭВМ с отображением портов управления в область ОЗУ, часы/календарь PCF8583 и EEPROM AT24C64, подключенные через шину I<sup>2</sup>C, с возможностью управления с помощью встроенного в PCF80C552 контроллера или характерным для большинства микро-ЭВМ семейства MCS-51 чисто программным способом. Помимо этого, плата содержит ряд логических микросхем, выполняющих функции дешифрации и формирования управляющих сигналов для работы основных элементов, а также формирования сигналов выборки для подключаемых к системной шине устройств пользователя.

Область ПЗУ располагается по адресам \$0000...\$7FFF (знак \$ обозначает шестнадцатеричное основание), область эмулятора ПЗУ по адресам \$8000...\$FFFF, область ОЗУ \$0000...\$7DFF. Область адресов \$7E00...\$7FFF предназначена для операций ввода/вывода. Обращение к области ввода/вывода строится соответствующими сигналами CS0...CS7, причем первые 7 выведены вблизи макетного поля для подключения дополнительных устройств, а последний зарезервирован для управления ЖКИ-модулем. Вся область ввода/вывода \$200 адресов поделена на 8 областей по \$40 адресов. Сигнал CS0 строит обращение по адресам \$7E00...\$7E3F, CS1 по адресам \$7E40...\$7E7F и т. д.

Для управления ЖКИ-модулем имеется одна адресная линия (RS), выбирающая внутренний регистр, и линия выбора операции чтения/записи (R/W). Линия RS подключена к адресной линии A1, а R/W к A0, стробирующий ЖКИ-модуль сигнал CSLCD активизируется при обращении к области \$7FC0...\$7FFF. Таким

образом, для управления ЖКИ-модулем определены следующие адреса: \$7FC0 – запись в регистр IR ЖКИ-модуля), \$7FC1 – чтение регистра IR (BF и AC), \$7FC2 – запись в регистр DR, \$7FC3 – чтение регистра DR. Чтобы избежать шинных конфликтов и выхода компонентов из строя, не производите операции чтения по адресам \$7FC0 и \$7FC2, а операции записи по адресам \$7FC1 и \$7FC3.

Остальные периферийные микросхемы (EEPROM AT24C64 и часы PCF8583) подключены через шину I<sup>2</sup>C. Сигнальные линии шины I<sup>2</sup>C SCL и SDA подключены соответственно к портам P1.6 и P1.7, как в варианте с микро-ЭВМ PCF80C552, так и с микро-ЭВМ в 40-выводных корпусах DIP. Естественно, аппаратный контроллер шины UC доступен только в варианте с PCF80C552 либо при использовании микро-ЭВМ PCF80C652 в стандартном 40-выводном корпусе DIP, но содержащей контроллер типа I<sup>2</sup>C (а также АЦП), аналогичный PCF80C552. При использовании обычных микро-ЭВМ семейства MCS-51 обмен по шине UC реализуется программно. Нужно отметить, что программная реализация Master-абонента шины I<sup>2</sup>C не столь уж сложный процесс, и к такому решению нередко прибегают и при использовании микро-ЭВМ PCF80C552 и PCF80C652.

Для запуска отлаживаемой программы ее необходимо откомпилировать, начиная с адреса \$8000, в том числе и точки входов прерываний, которые соответствующим образом переносятся монитором-загрузчиком из области \$0000 в \$8000 командами JMP, и получить стандартный Intel HEX-файл.

После подачи на EB-552 напряжения питания или нажатия кнопки RES на экране отображается адрес E-mail фирмы КТЦ-МК, при этом плата находится в режиме ожидания получения HEX-файла через RS-232. HEX-файл передается из компьютера командой COPY в режиме 9600 бод с 1 старт-битом 8 битами данных. Для того чтобы установить этот режим COM-порта, нужно выполнить команду MS-DOS MODE COM2 96,n,8,1.

Когда HEX-файл полностью принят, программа автоматически запускается. Если в режиме ожидания HEX-файла, не передавая HEX-файл, нажать на кнопку INT0/T0, то появляются дополнительные возможности. Кнопка INT0 перебирает возможные альтернативы, а T0 позволяет выбрать необходимую:

- “Load Hex to Buffer” загружает HEX-файл без автоматического запуска;

- “Jump to 8000h” запускает выполнение программы с адреса \$8000;
- “View RAM, from 0000” – просмотр содержимого ОЗУ;
- “View eROM, from 8000” – просмотр содержимого эмулятора ПЗУ;
- “Check hardware” заводит тестирование микросхем ОЗУ и эмулятора ПЗУ, а также сканирует адреса шины УС, выводя адреса, по которым получен ответ.

Через разъем ХР<sub>3</sub> подается напряжение питания 5 В, наличие которого индицируется светодиодом VD1. ХР1 – розетка типа DB-9, через которую осуществляется соединение посредством интерфейса RS-232. Для формирования сигналов интерфейса применена микросхема ADM202, включенная стандартным способом. ХР2 – розетка типа RJ-11 для подсоединения внешних устройств к шине I<sup>2</sup>C. Кроме стандартных сигналов шины, на разъем дополнительно выведена линия питания +5В.

Конденсаторы С1, С9, С10, С12, С15 и С16 выполняют функции фильтрующих элементов, конденсатор С2 и резистор R5 образуют схему формирования сигнала сброса RST при включении питания или вручную – нажатием кнопки SB1. Тактовый генератор микро-ЭВМ включает в себя кварцевый резонатор ZQ2 и конденсаторы С13, С14. Для управления режимами работы микро-ЭВМ DD9 (DD11) на плате установлены съемные переключки SW3 и SW4. Первая управляет разрешением режима работы микроконтроллера с внешней памятью, вторая разрешает работу устройства восстановления работоспособности (Watchdog). Естественно, Watchdog доступен только при использовании микро-ЭВМ PCF80C552.

Для адресации и выборки микросхем внешней памяти и ЖКИ-модуля, а также формирования сигналов выборки устройств пользователя служат схемы управления на микросхемах DD3...DD7. На элементах DD6.1, DD6.2 и микросхеме DD4 собран дешифратор области ввода/вывода \$7E00... \$7FFF. Микросхема DD3 формирует сигналы CS0...CS7, стробирующие обращение к области ввода/вывода. Остальные элементы участвуют в формировании сигналов разрешения доступа к микросхемам ОЗУ (DD12) и эмулятора ПЗУ (DD13). Для формирования сигналов шины адреса, являющейся у микро-ЭВМ семейства MCS-51 мультиплексированной, служит регистр DD14 типа 74HC573, фиксирующий по сигналу ALE младшую половину (A0...A7) значения адреса (рис. 15).

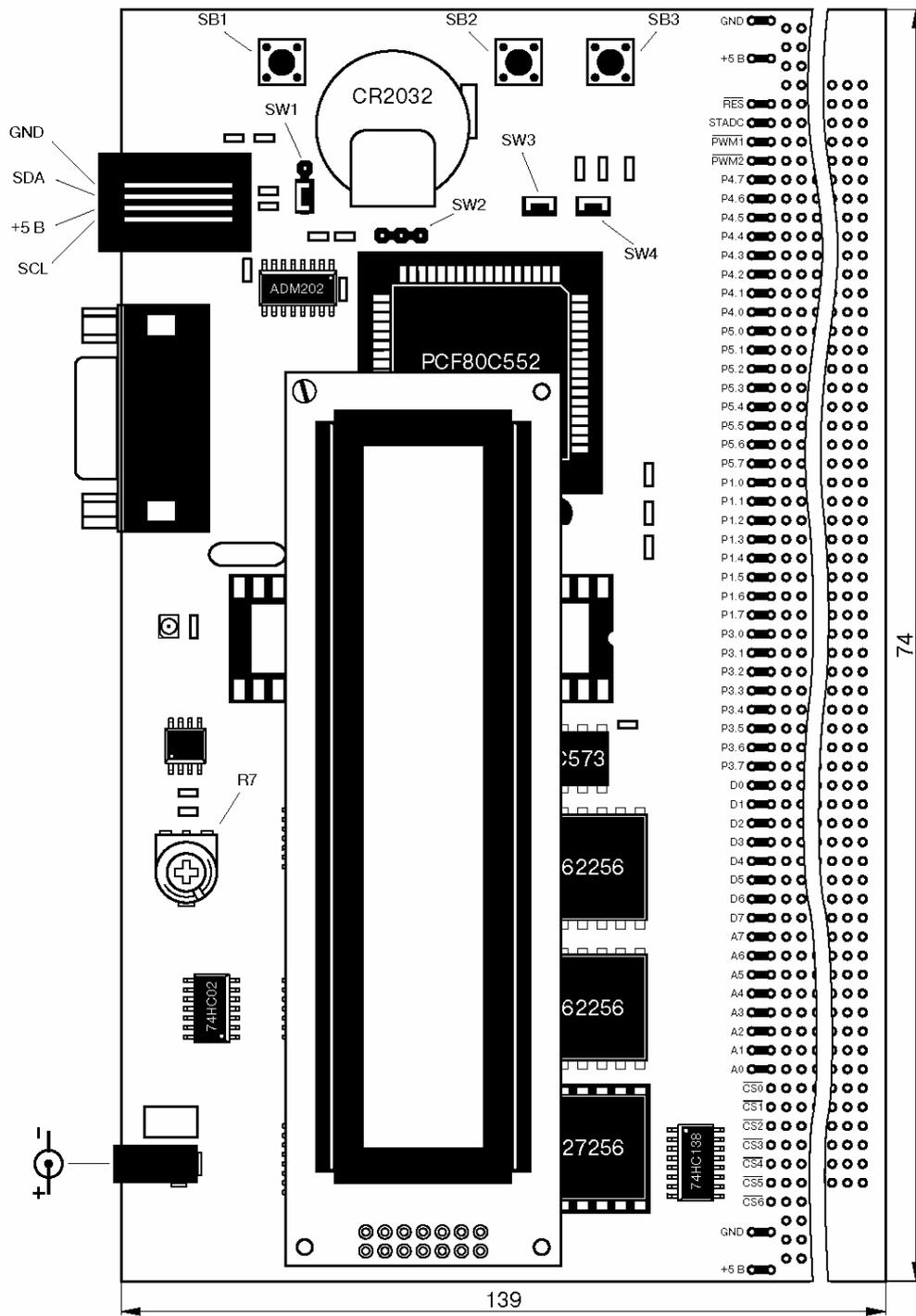


Рис. 15. Расположение элементов EB-552

Драйвер RS-232 и линии шины I<sup>2</sup>C подключены соответственно к портам P3.0, P3.7 и P1.6, P1.7, являющимися стандартными портами для исключения этих шин.

Микросхема часов PCF8583 включена стандартным образом. Часы энергонезависимы, при пропадании напряжения 5 В на разъеме XP3 переключаются на питание от литиевого гальванического элемента GB1. Коммутация источников питания производится диодами VD1 и VD2. Генератор опорной частоты 32768 Гц выполнен на кварцевом резонаторе ZQ1 и конденсаторе C1.

Подстроечный резистор R7 позволяет плавно изменять напряжение питания драйвера ЖК-панели путем изменения потенциала на входе V0. Это необходимо для правильной установки режима работы модуля, так как различные ЖК-материалы и конструкции ЖК-панелей могут иметь существенно различающиеся характеристики, что, к примеру, может привести к полному отсутствию изображения на одном ЖКИ-модуле, включенном в схему с настройкой резистора R7 для другого.

Кроме того, ЖКИ, предназначенные для работы при отрицательных температурах, требуют повышенного напряжения питания драйвера, для чего на вход V0 необходимо подать отрицательное напряжение (-1...-3 В). Для реализации этой возможности на плате EB-552 установлен маломощный преобразователь напряжения на основе микросхемы MAX660 (DA1).

В общем случае для активизации работы ЖКИ-модуля можно рекомендовать следующую последовательность действий. Подайте напряжение питания на плату EB-552 и поворачивайте движок резистора R7. После подачи напряжения питания модуль включается в режим развертки одной верхней строки. Сегменты этой строки должны менять свое состояние от прозрачного до непрозрачного, что является свидетельством работоспособности драйверов ЖКИ. Установите движок в такое положение, при котором изображение сегментов в верхней строке едва проступает на основном фоне ЖКИ. Теперь ЖКИ-модуль готов к приему и отображению информации. После того как вы добьетесь отображения на индикаторе какого-либо текста, вы сможете более точно отрегулировать изображение в соответствии с необходимой контрастностью и требуемым углом наблюдения.

Все сигналы внутренней шины EB-552 выведены вблизи монтажного поля платы и промаркированы (их относительное расположение соответствует указанному на принципиальной

электрической схеме), что позволяет быстро и с минимальными затратами расширить возможности ЕВ-552, собирая на монтажном поле различные устройства для конкретных применений.

### 2.3.3. Порядок выполнения лабораторной работы

- Используя методику, описанную в приложениях 2, 3, 4, запустить на стенде тестовые задачи.
- Получить у преподавателя задание. Написать программу и продемонстрировать ее работоспособность преподавателю.

#### Тестовые задачи по проверке работоспособности и формирования навыков программирования лабораторного стенда ЕВ-552(с решениями)

##### Задача 1

Написать программы для ПЭВМ и стенда ЕВ-552, обеспечивающие передачу байта через СОМ-порт со стороны ПЭВМ, его прием отладочной платой, инвертирование и передачу назад в ПЭВМ. ПЭВМ должна принять инвертированный байт, сложить его с переданным и результат вывести на экран ПЭВМ. Программу для ПЭВМ написать с использованием языка Паскаль, а для 80С552 – языка Ассемблера x8051.

**Решение:**

```

;-----
org      8000h
ajmp    INIT
org      8003h
reti
org      800Bh
reti
org      8013h
reti
org      801Bh
reti
org      8023h
reti
INIT:
mov IE,#00000000b
;-----Подготовка таймера
S9600 equ 0FDh;--F обмена - 9,6кГц
clr TR1
mov TMOD,#00100000b
mov TL1,#S9600
mov TH1,#S9600
;----- Подготовка УАПЧ
mov PCON,#00000000b
mov SCON,#01100010b
;Режим-1 / готовность
передачи (setb TI)
clr RI ;Сброс флага
готовности приемника
setb REN
;Разрешение приема
WAIT:
jnb RI,READY
ajmp WAIT
READY: mov A,SBUF ;Прием
clr RI
xrl A,#255
clr TI
mov SBUF,A ;Передача
ajmp WAIT
;*****
end

```

## Задача 2

Написать программу, выводящую на встроенный ЖКИ индикатор Ф.И.О студента. (Усложненный вариант задачи 2. Вывод ФИО осуществить в режиме бегущей строки.)

```
Решение:
;-----
org      8000h
ajmp    INIT
org      8003h
reti
org      800Bh
reti
org      8013h
reti
org      801Bh
reti
org      8023h
reti
;-----
org 8050h
DELAY:
djnz R7,DELAY
ret
WAIT_LCD:
mov DPTR,#7FC1h
LOOP1:
movx A,@DPTR
jb A.7,LOOP1
ret
WRITE_LCD_DR:
mov DPTR,#7FC2h
movx @DPTR,A
acall WAIT_LCD
ret
WRITE_LCD_IR:
mov DPTR,#7FC0h
movx @DPTR,A

INIT:
mov A,#1
acall WRITE_LCD_IR
mov A,#83h
acall WRITE_LCD_IR
mov A,#'I'
acall WRITE_LCD_DR
mov A,#'v'
acall WRITE_LCD_DR
mov A,#'a'
acall WRITE_LCD_DR
mov A,#'n'
acall WRITE_LCD_DR
mov A,#'o'
acall WRITE_LCD_DR
mov A,#'v'
acall WRITE_LCD_DR
mov A,#' '
acall WRITE_LCD_DR
mov A,#'I'
acall WRITE_LCD_DR
mov A,#'.'
acall WRITE_LCD_DR
mov A,#'I'
acall WRITE_LCD_DR
mov A,#'.'
acall WRITE_LCD_DR
this:
ajmp this
end
```

## Задача 3

Написать программу опроса встроенных на плату кнопок INT0 и T0. Осуществить вывод названия нажатой кнопки на индикатор.

```
Решение:
;-----
org      8000h
ajmp    INIT
org      8003h
reti
org      800Bh
reti
org      8013h
reti
org      801Bh
reti
;-----
org 8050h
DELAY:
djnz R7,DELAY
ret
WAIT_LCD:
mov DPTR,#7FC1h
LOOP1:
movx A,@DPTR
jb A.7,LOOP1
```

```

        ret
WRITE_LCD_DR:
    mov DPTR,#7FC2h
    movx @DPTR,A
    acall WAIT_LCD
    ret
WRITE_LCD_IR:
    mov DPTR,#7FC0h
    movx @DPTR,A
    acall WAIT_LCD
    ret
INIT:
    mov IE,#0
CLEAR:
    setb INTO
    setb T0
    mov A,#1
    acall WRITE_LCD_IR
MAIN:
    jnb INTO,GOT_I0
    jnb T0,GOT_T0
    ajmp MAIN
GOT_I0:
    mov A,#80h
    acall WRITE_LCD_IR
    mov A,#'I'
    acall WRITE_LCD_DR
    mov A,#'N'
    acall WRITE_LCD_DR
    mov A,#'T'
    acall WRITE_LCD_DR
    mov A,#'0'
    acall WRITE_LCD_DR
    mov A,#' '
    acall WRITE_LCD_DR
    mov A,#'P'
    acall WRITE_LCD_DR
    mov A,#'r'
    acall WRITE_LCD_DR
    mov A,#'e'
    acall WRITE_LCD_DR
    mov A,#'s'
    acall WRITE_LCD_DR
    mov A,#'s'
    acall WRITE_LCD_DR
    mov A,#'e'
    acall WRITE_LCD_DR
    mov A,#'d'
    acall WRITE_LCD_DR
THIS1:
    jnb INTO,THIS1
    ajmp CLEAR
GOT_T0:
    mov A,#80h
    acall WRITE_LCD_IR
    mov A,#'T'
    acall WRITE_LCD_DR
    mov A,#'0'
    acall WRITE_LCD_DR
    mov A,#' '
    acall WRITE_LCD_DR
    mov A,#'P'
    acall WRITE_LCD_DR
    mov A,#'r'
    acall WRITE_LCD_DR
    mov A,#'e'
    acall WRITE_LCD_DR
    mov A,#'s'
    acall WRITE_LCD_DR
    mov A,#'s'
    acall WRITE_LCD_DR
    mov A,#'e'
    acall WRITE_LCD_DR
    mov A,#'d'
    acall WRITE_LCD_DR
THIS2:
    jnb T0,THIS2
    ajmp CLEAR
END

```

## КОНТРОЛЬНЫЕ ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ

### Задача 1

Вычислить ряд чисел по закону:  $a_i = a_{i-2} * 3 + a_{i-1}$ , где  $a_1 = 1$ ,  $a_2 = 1$ . Результат от  $a_1$  до  $a_7$  вывести в верхнюю строку ЖКИ дисплея через пробел.

### Задача 2

Посчитать сумму чисел от 15 до 33, не пользуясь формулой арифметической прогрессии. Результат в десятичном виде выдать на ЖКИ дисплей.

### Задача 3

Заполнить первые 255 ячеек внешнего ОЗУ по закону: 0, 128, 1, 129, 2, 130, 3, 131... Вычислить 8-разрядную «контрольную сумму» полученного массива и выдать ее на дисплей в виде шестнадцатеричного числа.

Закон вычисления контрольной суммы:  $S = S \ll 1 + a_i$ , где операция « $\ll$ » – это операция циклического сдвига влево.

### Задача 4

Вывести на дисплей (выводя каждое шестнадцатеричное число через пробел и используя оба ряда индикатора) первые 16 байт, пришедшие от ПЭВМ по последовательному порту. Программу, отправляющую исходные данные с ПЭВМ, написать на языке высокого уровня.

### Задача 5

В полученном от ПЭВМ через последовательный порт числе переставить биты в обратном порядке. Результат отправить в ПЭВМ через последовательный порт. Написать программу для ПЭВМ для проверки правильности функционирования.

### Задача 6

Посчитать модуль разности количества единичных бит, стоящих на четных местах байта, принятого с ПЭВМ по последовательному порту, и количества единичных бит на нечетных местах этого байта. Полученный результат выдать на индикатор в десятичном виде.

### Задача 7

Написать программу-часы, выводящую на индикатор текущее время (чч:мм:сс), используя один из программируемых таймеров.

Усложненный вариант: модифицировать программу таким образом, чтобы при помощи кнопок INT0 и T0 была возможность

редактировать показания часов. Для примера взять обычные электронные часы.

### Задача 8

Написать программу-часы, используя микросхему часов реального времени.

### Задача 9

Написать программу, обеспечивающую формирование записной книжки на базе EEPROM AT24C64. Записать в нее список группы.

## 2.4. Лабораторная работа № 4

### Работа отладочной платы EV 552 с МК Atmel 89C51

Цель работы:

- Познакомиться с микроконтроллером Atmel 89C51, научиться программировать приборы, подключаемые к МК.
- Научиться подключать и программировать ЖКИ индикатор.

Для написания программ под микроконтроллер Atmel 89C51 используются программные средства, разработанные для программирования микроконтроллеров семейства MCS-51. В нашем случае используются компилятор – X8051.exe и линковщик (компоновщик) – Link.exe.

Этапы построения и отладки исполняемого файла:

1) Написание исходного текста в любом текстовом редакторе и сохранение его в файле с расширением \*.a51.

2) Компиляция исходного текста компилятором путем ввода в командной строке последовательности «X8051.exe *f\_name.a51*» (без кавычек). Если есть на этом этапе синтаксические ошибки, то исправляем их в исходном тексте. Если же ошибки не были обнаружены, то на выходе создается объектный файл *f\_name.obj*.

3) Следующий этап – это связывание меток путем запуска программы «link.exe», ввода имени объектного файла (в нашем случае *f\_name.obj*).

**Примечание.** На вопрос программы о смещении задать 0. На вопрос о ключе ввести ключ «X» (именно этот ключ обеспечивает получение исполняемого файла в формате \*.tsk).

4) Полученный файл в формате \*.tsk используем при программировании микроконтроллера с помощью программатора.

Если исходный текст программы был написан для запуска из внешнего ОЗУ, то для запуска ее из внутренней Flash-памяти МК следует внести коррективы в соответствии с нижеследующим примером.

**Тестовая задача 2.** Написать программу, выводящую на встроенный ЖКИ-индикатор ФИО студента.

Решение задачи 2:

- взять программу: C:\Shalat\Philips\fname.a51

- добавить в начало блока init команды:

```
init:
    mov a, #38h
call   write_LCD_IR
    mov a, #0Eh
call   write_LCD_IR
    .....
```

### **Порядок выполнения лабораторной работы**

1. Заменить на плате EB552 МК PCF80C552 на Atmel 89C51.
2. Запустить на стенде адаптированные тестовые задачи.
3. Получить у преподавателя задания. Написать программу и продемонстрировать результат преподавателю.

**Примечание.** Задачи запускать на выполнение в двух режимах:

- из внешней памяти программ;
- из внутренней памяти программ, предварительно записав их туда с помощью программатора.

#### **Внимание.**

При программировании МК Atmel 89C51 обращать особое внимание на правильность расположения ключа при установке МК в программатор!

## **2.5 Лабораторная работа № 5.**

### **Работа отладочной платы C8051F300 с МК Cygnal F300**

Цель работы: познакомиться с микроконтроллером Cygnal F300, научиться программировать приборы, подключаемые к МК.

## Эволюционный комплект C8051F300: структура платы, ресурсы платы и ПО к плате

Эволюционный комплект C8051F300 представлен на рис. 16.

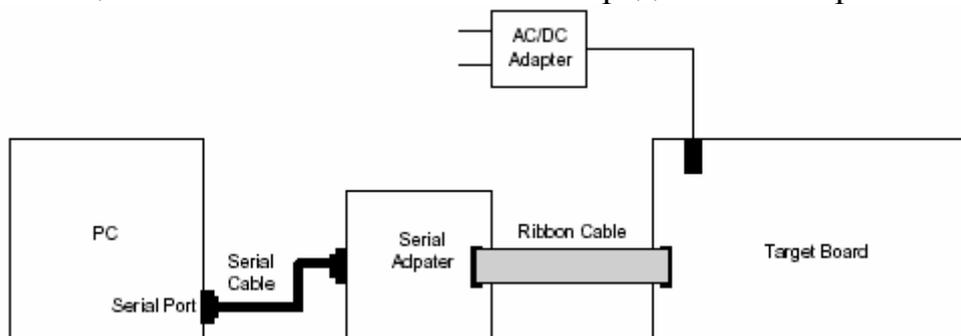


Рис. 16. Эволюционный комплект C8051F300

В состав эволюционного комплекта входят:

- Собственно эволюционная плата с установленным микроконтроллером C8051F320 (см. рис. 17)

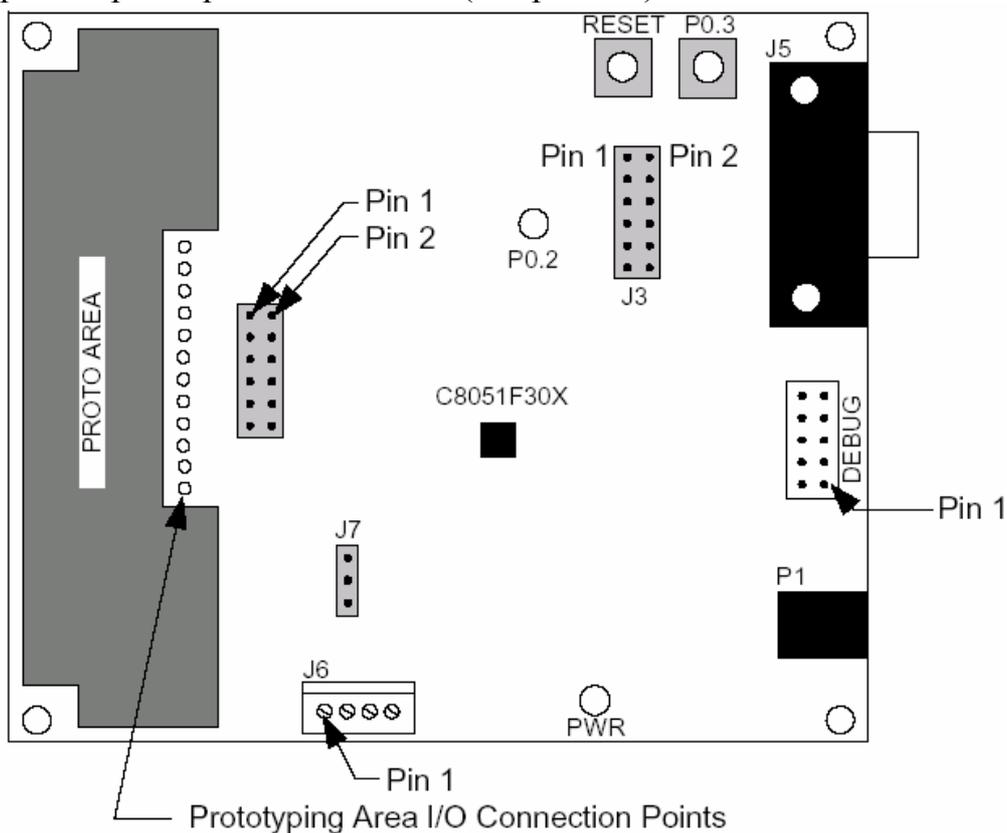


Рис. 17. Эволюционная плата C8051F300

- Адаптер-программатор, преобразующий входной интерфейс персонального компьютера RS-232C в интерфейс

программирования JTAG. В последнее время поставляется комбинированный адаптер ЕС2, поддерживающий два возможных интерфейса программирования (JTAG и I2C). Адаптер выполнен в пластмассовом корпусе с размерами 50x58x26 мм (см. рис. 18). Внутри находится контроллер программатора, выполненный на микросхеме С8051F012.

В качестве интерфейсной микросхемы RS-232С использована Sipex3223. Кроме того, имеется стабилизатор напряжения на 3,3В, кварцевый резонатор и два светодиода. Входной разъем для RS-232С — DB9F (розетка). Выходной разъем — двухрядный штырьковый разъем PLD-10 (2x5).

В нормальном рабочем режиме адаптер питается от эволюционной платы через 10-проводный плоский кабель. Напряжение питания — 3..6 В. Кроме того, адаптер может питаться от внешнего источника с выходным напряжением 5...9В, для чего предусмотрено стандартное гнездо.

- Настенный малогабаритный источник питания с размерами 55x42x35. Выходное напряжение — 9В, ток нагрузки — до 100 мА.

- Стандартный компьютерный кабель с двумя разъемами DB9F (розетка) - DB9 (вилка) для подсоединения адаптера к персональному компьютеру.

- 10-проводный плоский кабель с двумя разъемами DPS10 (2x5) для соединения адаптера и эволюционной платы.

- Техническое описание и руководство по эксплуатации.

- CD-ROM с программным обеспечением. В состав программного обеспечения входят: фирменная среда программирования Cygnal Integrated Development Tools; комплект программного обеспечения Evolution Keil uVision (макроассемблер, линкер, библиотекарь и эволюционная версия компилятора языка С с ограничением по размеру выходного кода 2 К); инсталляционная утилита Setup.exe; исходные тексты примеров и файлов определения регистров; документация.

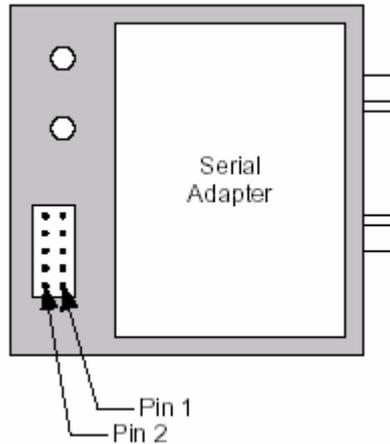


Рис. 18. Адаптер-программатор EC2

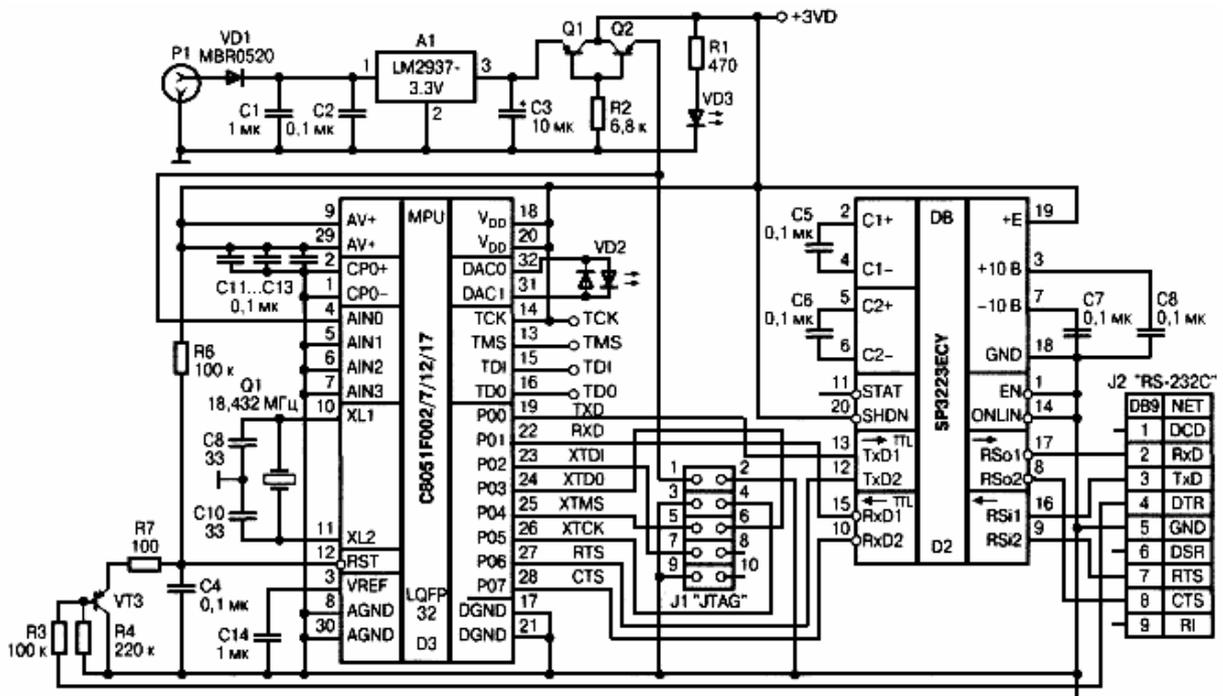


Рис. 19. Принципиальная схема адаптера-программатора EC-2 фирмы Cygnal

**Эволюционная плата семейства C8051F30x.** Принципиальная схема эволюционной платы C8051F300 разделена на три фрагмента. На рис. 20 показан узел микроконтроллера с обвязкой, на рис. 21 – стабилизатор напряжения и на рис. 22 – узел интерфейса RS-232C.

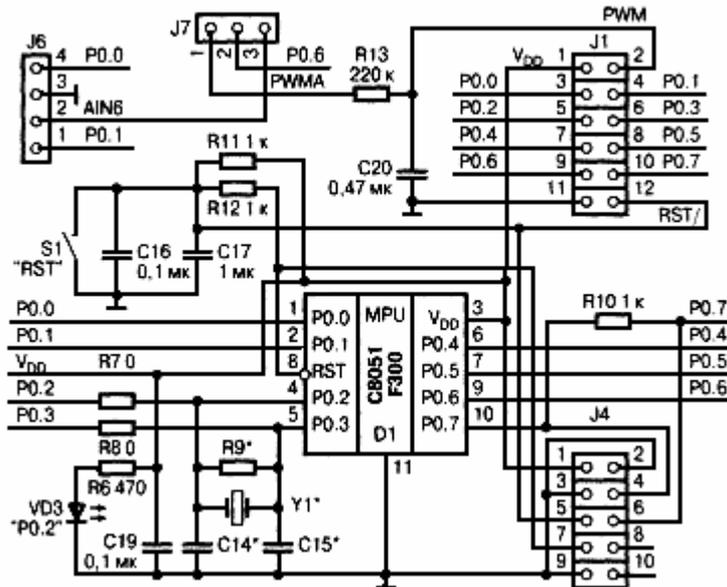


Рис. 20. Узел микроконтроллера C8051F300 с обвязкой

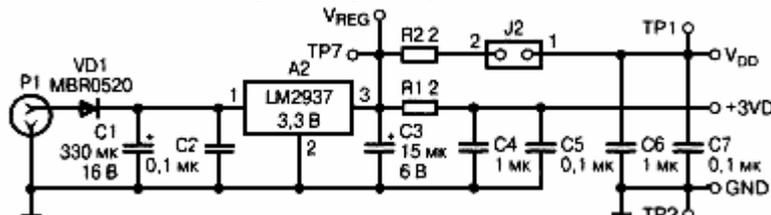


Рис. 21. Узел стабилизатора напряжения

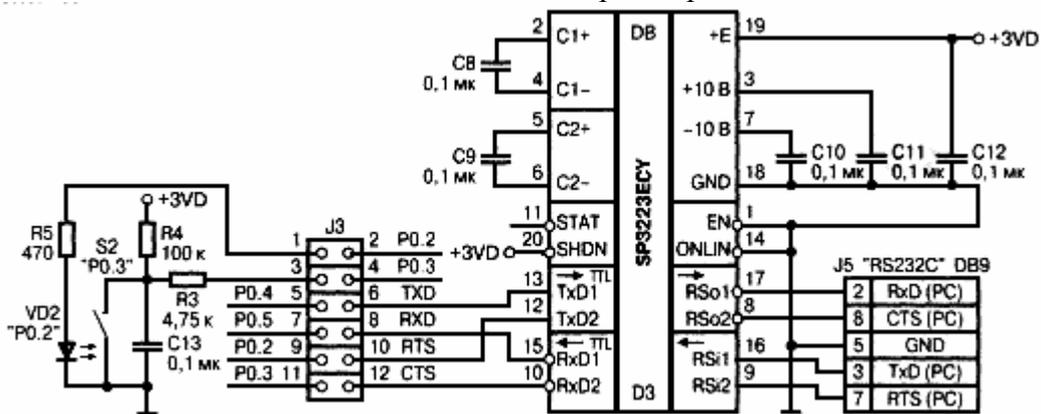


Рис. 22. Узел интерфейса RS-232C

Узел микроконтроллера содержит собственно микроконтроллер D1 (C8051F300), кнопку сброса S1, светодиод питания VD3 и набор штыревых разъемов. Перечень штыревых разъемов приведен в табл. 11.

Узел стабилизатора питания выполнен на микросхеме A2 (LM2937-3.3V) на напряжение 3,3 В по стандартной схеме. В табл. 12 приведены контакты выходного разъема эволюционной платы.

ТАБЛИЦА 11

## Перечень разъемов эволюционной платы С8051F3xx

Обозначение	Назначение
J1	Выходной разъем PLD12 (2x6) (см. табл. 12)
J2	Переключатель для подключения питания к микроконтроллеру
J3	Разъем для установки переключателей, которые предназначены для подключения сигналов от кнопки S2 (к линии порта P0.3), светодиода VD2 (к линии порта P0.2) и сигналов интерфейса RS-232C
J4	Разъем для подключения адаптера-программатора ЕС-2 (см. табл. 13)
J5	Выходной разъем интерфейса RS-232C (см. табл. 14)
J6	Клеммник винтовой для подключения внешних аналоговых сигналов типа 310-04-2-3
J7	Разъем для установки переключателя, который может подключить на вывод P0.6 либо клеммник винтовой J6/2, либо RC-фильтр для исследования широтно-импульсного модулятора PWM

ТАБЛИЦА 12

## Контакты выходного разъема J1 эволюционной платы С8051F300

Контакт	Описание	Контакт	Описание
1	VDD цифровое питание	2	PWM
3	P0.0	4	P0.1
5	P0.2	6	P0.3
7	P0.3	8	P0.5
9	P0.6	10	P0.7
11	GND	12	RST/

Узел интерфейса RS-232C также выполнен по стандартной схеме включения на микросхеме SP3223ESY. Контакты выходного разъема приведены в табл. 13. Размеры эволюционной платы составляют 85x103 мм.

Макетное поле одно, но довольно большое, имеет 11x26 металлизированных отверстий с шагом 2,54 мм. С двух сторон этого поля располагаются шины общего провода GND и питания VA+. Эволюционная плата выполнена по технологии SMD (поверхностного монтажа) с защитным покрытием (“зеленкой”). Описанный эволюционный комплект позволяет производить разработку и отладку тестового программного обеспечения, а также на практике изучать особенности работы встроенных аналоговых и цифровых узлов.

ТАБЛИЦА 13

Выходной разъем интерфейса RS-232C эволюционной платы C8051F300

Контакт	Описание
2	RxD (вход РС), в контроллере соединен с P0.5
3	TxD (выход РС), в контроллере соединен с P0.4
8	CTS (вход РС), в контроллере соединен с P0.3
7	RTS (выход РС), в контроллере соединен с P0.2
5	GND

ТАБЛИЦА 14

Контакты разъема программирования J4 по интерфейсу C2

Контакт	Описание
1	Вход питания от эволюционной платы 3,0... 3,6В
2	Цифровой общий провод
4	C2DAT (для интерфейса JTAG — TCK)
5	TMS
6	TDO
7	C2CLK (для интерфейса JTAG — TDI)
3, 8, 9, 10	Не используются

### Порядок выполнения лабораторной работы

1. Подключить отладочную плату к компьютеру, запустить среду разработки.
2. Разобраться с программой обработки показаний встроенного в микроконтроллер Cygnal C8051F300 температурного датчика. Пример программы на Ассемблере приведен в Приложении 6.
3. Написать программу, обеспечивающую включение/выключение светодиода P0.2 на отладочной плате с частотой 0,5 Гц. Процедура мерцания должна начинаться (завершаться) по нажатию на переключатель P0.3.

## ПРИЛОЖЕНИЕ 1

### ТИПЫ ФАЙЛОВ

Содержимое запоминающего устройства, которое должно быть обработано на компьютере или просто сохранено на дискете, называют *файлом*.

С точки зрения информатики термин «файл» обозначает последовательность бантов, размещенных на каком-либо носителе, которую можно легко защитить, копировать, перемещать или изменять с помощью обычного программного обеспечения.

#### Двоичные файлы

Очевидно, что записать байты такими, какими они представлены в запоминающем устройстве, наиболее просто. Такой файл называют бинарным, или двоичным.

Для примера рассмотрим последовательность из 22 десятичных значений, представленных ниже в виде обычного текста:

194	128	175	255	174	255	222	254
223	250	210	128	175	255	174	255
222	254	223	250	128	234		

Если мы запишем их в файл в виде 22 соответствующих байтов, листинг на экране или на принтере получится таким, как он показан ниже (по крайней мере, на ЭВМ, совместимой с ПК). Пример двоичного файла представлен в символах ASCII.

```
_5» « 1.И.-5» « |.и.<рп
```

Возможно, это самая компактная по количеству символов форма записи байтов, но и самая непонятная.

#### Десятичный файл

Более удобной формой записи является *свободный десятичный формат* (free format decimal). Чтобы сохранить данные в этом формате, байты надо записывать соответствующими десятичными числами, разделенными пробелами. Таким образом, до четырех байт текста в файле соответствуют одному байту в запоминающем устройстве. Взамен – наглядность изменений, которые легко внести

даже с помощью простейших программ, находящихся в распоряжении любого пользователя ПК.

### Шестнадцатеричный файл

Тот же самый принцип может быть применен и к шестнадцатеричным величинам, требующим только двух символов для представления байта информации. Если, кроме того, исключить разделительные пробелы, получим представление, известное под названием *сплошной гекс* (straight hex), то есть только шестнадцатеричные. Пример файла в формате straight hex:

```
C280AFFFAEFFDEFEDFFAD280AFFFAEFFDEFEDFFA80EA
```

Занимающий только два байта текста на один байт в запоминающем устройстве (и, возможно, несколько специальных символов в конце строки), формат straight hex – хороший компромисс между компактностью и наглядностью.

Однако разработаны и довольно широко используются улучшенные варианты форматов шестнадцатеричных файлов, из которых наиболее известны Intel Hex и Motorola S (названия соответствуют именам компаний-производителей радиоэлементов, разработавших эти форматы).

Главное их преимущество – возможность включать специальные контрольные ключи, позволяющие обнаруживать ошибки, возникающие во время передачи или копирования файлов, а также наличие информации, определяющей, для каких адресов или области адресов запоминающего устройства предназначены байты. Уменьшение размера файла может быть очень существенным в случае, если запоминающее устройство заполнено только частично.

### Файл Intel Hex

Ниже приведен пример в формате файла Intel Hex:

```
:10000000C280AFFFAEFFDEFEDFFAD280AFFFAEFFF1  
:06001000DEFEDFFA80EACB  
:00000001FF.
```

Каждая строка файла, или *запись*, начинается с девяти символов, которые имеют следующие стандартные значения:

символ 1 – символ «двоеточие»;

символы 2 и 3 – количество значащих байтов в записи (здесь 16, или 10h в шестнадцатеричном коде);

символы 4 и 5 – старший байт адреса размещения этой записи в запоминающем устройстве (в шестнадцатеричном коде);

символы 6 и 7 – младший байт адреса (в шестнадцатеричном коде);

символы 8 и 9 – не используются (00).

Далее следуют информационные байты, причем каждый состоит из двух символов, как и в формате `straight hex`, потом – два символа, представляющие байт контрольной суммы (`checksum`), а затем – символы «возврат каретки» и «перевод строки».

Контрольная сумма вычисляется как дополнение до двух сумм восьмиразрядных чисел без учета переноса всех полезных байтов, двух байтов адреса и байта длины строки.

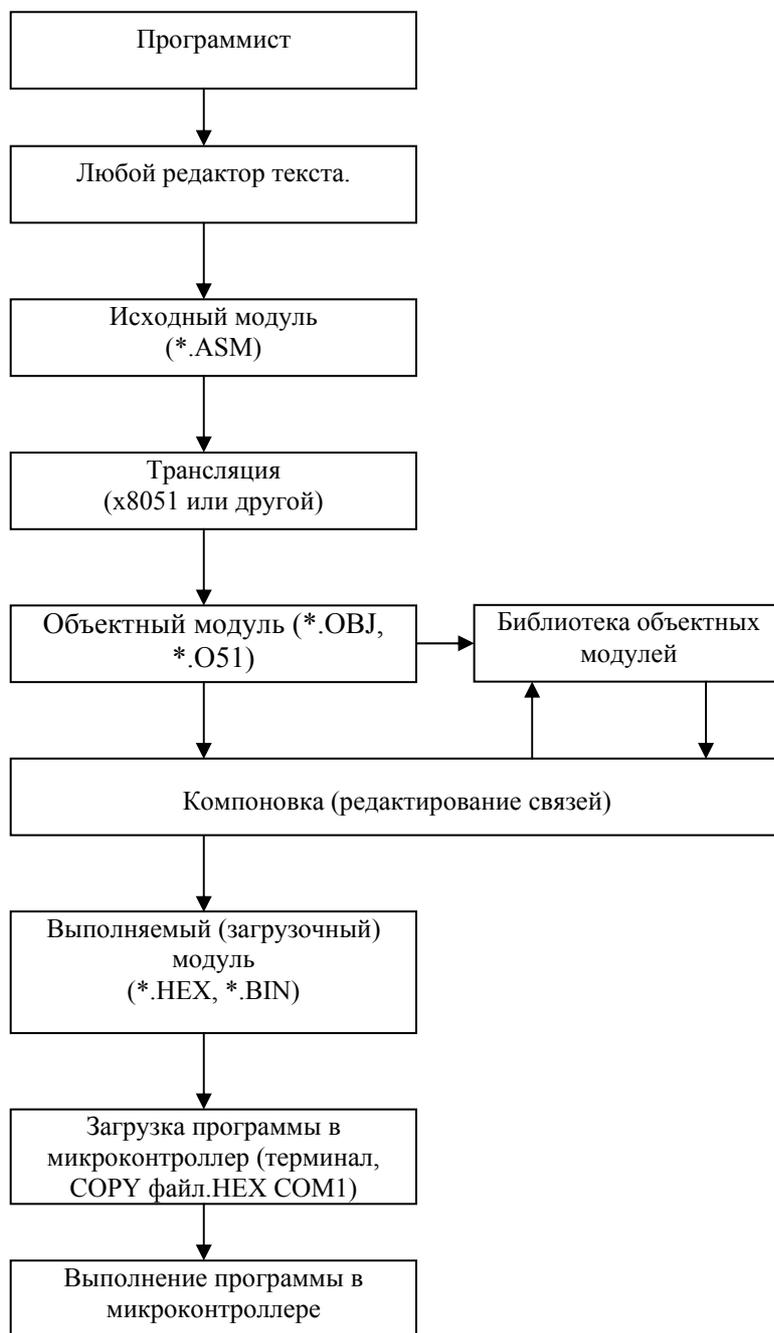
Последняя строка – «признак конца» – не содержит полезных байтов.

Приведенные стандарты важно знать, так как в большинстве имеющихся в продаже программаторов информация должна быть представлена в одном из описанных форматов. Соответственно многие программы и системы электронного проектирования генерируют коды для программирования в том или ином формате.

Достаточно часто может потребоваться перекодировка информации из одного формата в другой для устранения несовместимости оборудования и программного обеспечения разных производителей. Эту задачу решают относительно простые утилиты.

## ПРИЛОЖЕНИЕ 2

### ЭТАПЫ ПРОХОЖДЕНИЯ ЗАДАЧ ПОЛЬЗОВАТЕЛЯ С ВЫПОЛНЕНИЕМ ПРОГРАММЫ НА МИКРОКОНТРОЛЛЕРЕ



## ПРИЛОЖЕНИЕ 3

### ПРИМЕР ПАКЕТНЫХ ФАЙЛОВ, АВТОМАТИЗИРУЮЩИХ ПРОЦЕСС ТРАНСЛЯЦИИ ИСХОДНОГО КОДА С ПОМОЩЬЮ ТРАНСЛЯТОРА X8051

**Compile.bat** <имя файла>

```
@echo off
del *.o51 >nul
call a51.bat %1
pause
if exist %1.o51 call a51obj.bat %1
if exist %1.hex call upload.bat %1.hex
```

**A51.bat**

```
@Echo off
cls
echo ***** Translating ASM-x8051 ... *****
if exist %1.a51 goto t0
echo ERROR: Source file not found !!!
echo Example for use: a51[.bat] test8051
goto m2
:t0
x8051 %1.a51 -d >err
del %1.o51 >nul
ren %1.obj %1.o51
del %1.lst >nul
:m2
```

**A51obj.bat**

```
@Echo off
cls
rem path to file cr.cmd
if exist cr.cmd goto m0
echo Can't Find File cr.cmd !!!
echo Create It In Current Directory !
echo It Consist Of 4 Presses On <Enter> key,
echo and then use this bat file !!!
goto m2
:m0
cls
echo ***** Linking ASM-8051 ... *****
if exist %1.a51 goto t0
```

```

echo ERROR: In this directory must be this file with
echo extension .A51 !!!
echo example for use: test8051.a51
goto m2
:t0
if exist %1.o51 goto m1
echo OBJ file not found!!!
goto m2
:m1
rem ***** path to virtual disk cmd file
echo %1.o51 >a51.cmd
rem ***** Start Address For ASM-85 Programm *****
echo 0000 >>a51.cmd
copy/b a51.cmd+cr.cmd a51.cmd>nul
rem ***** linker name *****
link <a51.cmd
del a51.cmd >nul
del %1.o51 >nul
del %1.map >nul
if exist %1.tsk goto m2
echo ERROR IN OBJfile!!!
:m2

```

#### **Cr.cmd**

- (пустая строка)

-

-

d

## **ПРИЛОЖЕНИЕ 4**

### **ПРИМЕР ПАКЕТНЫХ ФАЙЛОВ, АВТОМАТИЗИРУЮЩИХ ПРОЦЕСС ТРАНСЛЯЦИИ ИСХОДНОГО КОДА С ПОМОЩЬЮ ТРАНСЛЯТОРА ASM51**

#### **Comp.bat <имя файла>**

```

asm51.exe %1.a51
oh.exe %1.obj
call upload.bat %1.hex

```

#### **upload.bat**

```

@mode com1 9600,n,8,1
@copy %1 com1

```

**Примечание.** Имя файла указывается без расширения.

**ПРИЛОЖЕНИЕ 5**

**БАЗОВАЯ СИСТЕМА КОМАНД  
ДЛЯ МИКРОКОНТРОЛЛЕРОВ С АРХИТЕКТУРОЙ MCS51**

**1. Группа команд передачи данных**

Название команды	Мнемокод	КОП	т	Б	Ц	Операция
Пересылка в аккумулятор из регистра ( $n = 0-7$ )	MOV A, Rn	11101rrr	1	1	1	(A) ← (Rn)
Пересылка в аккумулятор прямоадресуемого байта	MOV A, ad	11100101	3	2	1	(A) ← (ad)
Пересылка в аккумулятор байта из РПД ( $i = 0, 1$ )	MOV A, @Ri	1110011i	1	1	1	(A) ← ((Ri))
Загрузка в аккумулятор константы	MOV A, #d	01110100	2	2	1	(A) ← #d
Пересылка в регистр из аккумулятора	MOV Rn, A	11111rrr	1	1	1	(Rn) ← (A)
Пересылка в регистр прямоадресуемого байта	MOV Rn, ad	10101rrr	3	2	2	(Rn) ← (ad)
Загрузка в регистр константы	MOV Rn, #d	01111rrr	2	2	1	(Rn) ← #d
Пересылка по прямому адресу аккумулятора	MOV ad, A	11110101	3	2	1	(ad) ← (A)
Пересылка по прямому адресу регистра	MOV ad, Rn	10001rrr	3	2	2	(ad) ← (Rn)
Пересылка прямоадресуемого байта по прямому адресу	MOV add, ads	10000101	9	3	2	(add) ← (ads)
Пересылка байта из РПД по прямому адресу	MOV ad, @Ri	1000011i	3	2	2	(ad) ← ((Ri))
Пересылка по прямому адресу константы	MOV ad, #d	01110101	7	3	2	(ad) ← #d
Пересылка в РПД из аккумулятора	MOV @Ri, A	1111011i	1	1	1	((Ri)) ← (A)
Пересылка в РПД прямоадресуемого байта	MOV @Ri, ad	0110011i	3	2	2	((Ri)) ← (ad)

Пересылка в РПД константы	MOV @Ri, #d	0111011i	2	2	1	((Ri)) ← #d
Загрузка указателя данных	MOV DPTR, #dl6	10010000	13	3	2	(DPTR) ← #dl6
Пересылка в аккумулятор байта из ПП	MOVC A, @A+DPTR	10010011	1	1	2	(A) ← ((A) + (DPTR))
Пересылка в аккумулятор байта из ПП	MOVC A, @A+PC	10000011	1	1	2	(PC) ← (PC) + 1 (A) ← ((A) + (PC))
Пересылка в аккумулятор байта из ВПД	MOVX A, @Ri	11110001i	1	1	2	(A) ← ((Ri))
Пересылка в аккумулятор байта из расширенной ВПД	MOVX A, @DPTR	11100000	1	1	2	(A) ← ((DPTR))
Пересылка в ВПД из аккумулятора	MOVX @Ri, A	1111001i	1	1	2	((Ri)) ← (A)
Пересылка в расширенную ВПД из аккумулятора	MOVX @DPTR, A	11110000	1	1	2	((DPTR)) ← (A)
Загрузка в стек	PUSH ad	11000000	3	2	2	(SP) ← (SP) + 1 ((SP)) ← (ad)
Извлечение из стека	POP ad	11010000	3	2	2	(ad) ← (SP) (SP) ← (SP) - 1

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Обмен аккумулятора с регистром	XCH A, Rn	11001rrr	1	1	1	(A) ↔ (Rn)
Обмен аккумулятора с прямоадресуемым байтом	XCH A, ad	11000101	3	2	1	(A) ↔ (ad)

Обмен аккумулятора с байтом из РПД	XCH A, @Ri	1100011i	1	1	1	(A) ↔ ((Ri))
Обмен младшей тетрады аккумулятора с младшей тетрадой байта РПД	XCHD A, @Ri	1101011i	1	1	1	(A <sub>0-3</sub> ) ↔ (Ri) <sub>0-3</sub>

## 2. Группа команд арифметических операций

Название команды	Мнемокод	КОП	T	B	Ц	Операция
Сложение аккумулятора с регистром ( $n = 0-7$ )	ADD A, Rn	00101rrr	1	1	1	(A) ← (A) + (Rn)
Сложение аккумулятора с прямоадресуемым байтом	ADD A, ad	00100101	3	2	1	(A) ← (A) + (ad)
Сложение аккумулятора с байтом из РПД ( $i = 0, 1$ )	ADD A, @Ri	0010011i	1	1	1	(A) ← (A) + ((Ri))
Сложение аккумулятора с константой	ADD A, #d	00100100	2	2	1	(A) ← (A) + #d
Сложение аккумулятора с регистром и переносом	ADDC A, Rn	00111rrr	1	1	1	(A) ← (A) + (Rn) + (C)
Сложение аккумулятора с прямоадресуемым байтом и переносом	ADDC A, ad	00110101	3	2	1	(A) ← (A) + (ad) + (C)
Сложение аккумулятора с байтом из РПД и переносом	ADDC A, @Ri	0011011i	1	1	1	(A) ← (A) + ((Ri)) + (C)
Сложение аккумулятора с константой и переносом	ADDC A, #d	00110100	2	2	1	(A) ← (A) + #d + (C)
Десятичная коррекция аккумулятора	DA A	11010100	1	1	1	Если (A <sub>0-3</sub> ) > 9V((AC)=1), то (A <sub>0-3</sub> ) ← (A <sub>0-3</sub> ) + 6. Затем если (A <sub>4-7</sub> ) > 9V((C) = 1), то (A <sub>4-7</sub> ) ← (A <sub>4-7</sub> ) + 6
Вычитание из аккумулятора регистра и заема	SUBB A, Rn	10011rrr	1	1	1	(A) ← (A) - (C) - (Rn)

Вычитание из аккумулятора прямо-адресуемого байта и заема	SUBB A, ad	10010101	3	2	1	$(A) \leftarrow (A) - (C) - ((ad))$
Вычитание из аккумулятора байта РПД и заема	SUBB A, @Ri	1001011i	1	1	1	$(A) \leftarrow (A) - (C) - ((Ri))$
Вычитание из аккумулятора константы и заема	SUBB A, d	10010100	2	2	1	$(A) \leftarrow (A) - (C) - \#d$
Инкремент аккумулятора	INC A	00000100	1	1	1	$(A) \leftarrow (A) + 1$
Инкремент регистра	INC Rn	00001rrr	1	1	1	$(Rn) \leftarrow (Rn) + 1$
Инкремент прямо-адресуемого байта	INC ad	00000101	3	2	1	$(ad) \leftarrow (ad) + 1$
Инкремент байта в РПД	INC @Ri	0000011i	1	1	1	$((Ri)) \leftarrow ((Ri)) + 1$
Инкремент указателя данных	INC DPTR	10100011	1	1	2	$(DPTR) \leftarrow (DPTR) + 1$
Декремент аккумулятора	DEC A	00010100	1	1	1	$(A) \leftarrow (A) - 1$
Декремент регистра	DEC Rn	00011rrr	1	1	1	$(Rn) \leftarrow (Rn) - 1$
Декремент прямо-адресуемого байта	DEC ad	00010101	3	2	1	$(ad) \leftarrow (ad) - 1$
Декремент байта в РПД	DEC @Ri	0001011i	1	1	1	$((Ri)) \leftarrow ((Ri)) - 1$
Умножение аккумулятора на регистр В	MUL AB	10100100	1	1	4	$(B)(A) \leftarrow (A) \times (B)$
Деление аккумулятора на регистр В	DIV AB	10000100	1	1	4	$(A).(B) \leftarrow (A)/(B)$

### 3. Группа команд логических операций

Название команды	Мnemonic	КОП	Т	Б	Ц	Операция
Логическое И аккумулятора и регистра	ANL A, Rn	01011rrr	1	1	1	$(A) \leftarrow (A) \wedge (Rn)$
Логическое И аккумулятора и прямоадресуемого байта	ANL A, ad	01010101	3	2	1	$(A) \leftarrow (A) \wedge (ad)$
Логическое И аккумулятора и байта из РПД	ANL A, @Ri	0101011i	1	1	1	$(A) \leftarrow (A) \wedge ((Ri))$

Логическое И аккумулятора и константы	ANL A, #d	01010100	2	2	1	$(A) \leftarrow (A) \wedge \#d$
Логическое И прямо-адресуемого байта и аккумулятора	ANL ad, A	01010010	3	2	1	$(ad) \leftarrow (ad) \wedge (A)$
Логическое И прямо-адресуемого байта и константы	ANL ad, #d	01010011	7	3	2	$(ad) \leftarrow (ad) \wedge \#d$
Логическое ИЛИ аккумулятора и регистра	ORL A, Rn	01001rrr	1	1	1	$(A) \leftarrow (A) \vee (Rn)$
Логическое ИЛИ аккумулятора и прямо-адресуемого байта	ORL A, ad	01000101	3	2	1	$(A) \leftarrow (A) \vee (ad)$
Логическое ИЛИ аккумулятора и байта из РПД	ORL A, @Ri	0100011i	1	1	1	$(A) \leftarrow (A) \vee ((Ri))$
Логическое ИЛИ аккумулятора и константы	ORL A, #d	01000100	2	2	1	$(A) \leftarrow (A) \vee \#d$
Логическое ИЛИ прямо-адресуемого байта и аккумулятора	ORL ad, A	01000010	3	2	1	$(ad) \leftarrow (ad) \vee (A)$
Логическое ИЛИ прямо-адресуемого байта и константы	ORL ad, #d	01000011	7	3	2	$(ad) \leftarrow (ad) \vee \#d$
Исключающее ИЛИ аккумулятора и регистра	XRL A, Rn	01101rrr	1	1	1	$(A) \leftarrow (A) \vee (Rn)$
Исключающее ИЛИ аккумулятора и прямо-адресуемого байта	XRL A, ad	01100101	3	2	1	$(A) \leftarrow (A) \vee (ad)$
Исключающее ИЛИ аккумулятора и байта из РПД	XRL A, @Ri	0110011i	1	1	1	$(A) \leftarrow (A) \vee ((Ri))$
Исключающее ИЛИ аккумулятора и константы	XRL A, #d	01100100	2	2	1	$(A) \leftarrow (A) \vee \#d$
Исключающее ИЛИ прямо-адресуемого байта и аккумулятора	XRL ad, A	01100010	3	2	1	$(ad) \leftarrow (ad) \vee (A)$
Исключающее ИЛИ прямо-адресуемого байта и константы	XRL ad, #d	01100011	7	3	2	$(ad) \leftarrow (ad) \vee \#d$
Сброс аккумулятора	CLR A	11100100	1	1	1	$(A) \leftarrow 0$

Инверсия аккумулятора	CPL A	11110100	1	1	1	$(A) \leftarrow (\bar{A})$
Сдвиг аккумулятора влево циклический	RL A	00100011	1	1	1	$(A_{n+i}) \leftarrow (A_n),$ $n=0 \div 6,$ $(A_0) \leftarrow (A_7)$
Сдвиг аккумулятора влево через перенос	RLC A	00110011	1	1	1	$(A_{n+i}) \leftarrow (A_n),$ $n=0 \div 6,$ $(A_0) \leftarrow (C),$ $(C) \leftarrow (A_7)$
Сдвиг аккумулятора вправо циклический	RR A	00000011	1	1	1	$(A_n) \leftarrow (A_{n+i}),$ $n=0 \div 6,$ $(A_7) \leftarrow (A_0)$
Сдвиг аккумулятора вправо через перенос	RRC A	00010011	1	1	1	$(A_n) \leftarrow (A_{n+i}),$ $n=0 \div 6,$ $(A_7) \leftarrow (C),$ $(C) \leftarrow (A_0)$
Обмен местами тетрад в аккумуляторе	SWAP A	11000100	1	1	1	$(A_{0-3}) \leftrightarrow (A_{4-7})$

#### 4. Группа команд операций с битами

Название команды	Мнемокод	КОП	т	Б	Ц	Операция
Сброс переноса	CLR C	11000011	1	1	1	$(C) \leftarrow 0$
Сброс бита	CLR bit	11000010	4	2	1	$(b) \leftarrow 0$
Установка переноса	SETB C	11010011	1	1	1	$(C) \leftarrow 1$
Установка бита	SETB bit	11010010	4	2	1	$(b) \leftarrow 1$
Инверсия переноса	CPL C	10110011	1	1	1	$(C) \leftarrow (\bar{C})$
Инверсия бита	CPL bit	10110010	4	2	1	$(b) \leftarrow (\bar{b})$
Логическое И бита и переноса	ANL C, bit	10000010	4	2	2	$(C) \leftarrow (C) \wedge (b)$
Логическое И инверсии бита и переноса	ANL C, /bit	10110000	4	2	2	$(C) \leftarrow (C) \wedge (\bar{b})$
Логическое ИЛИ бита и переноса	ORL C, bit	01110010	4	2	2	$(C) \leftarrow (C) \vee (b)$

Логическое ИЛИ инверсии бита и	ORL C, /bit	10100000	4	2	2	$(C) \leftarrow (C) \vee (\bar{b})$
Пересылка бита в	MOV C, bit	10100010	4	2	1	$(C) \leftarrow (b)$
Пересылка переноса в бит	MOV bit, C	10010010	4	2	2	$(b) \leftarrow (C)$

### 5. Группа команд передачи управления

Название команды	Мнемокод	КОП	т	Б	Ц	Операция
Длинный переход в полном объеме памяти программ	LJMP ad 16	00000010	12	3	2	$(PC) \leftarrow ad\ 16$
Абсолютный переход внутри страницы в 2 Кбайта	AJMP ad 11	$a_{10}a_9a_800001$	6	2	2	$(PC) \leftarrow (PC) + 2$ $(PC_{0-10}) \leftarrow ad11$
Короткий относительный переход внутри страницы в 256 байт	SJMP rel	10000000	5	2	2	$(PC) \leftarrow (PC) + 2$ $(PC) \leftarrow (PC) + rel$
Косвенный относительный переход	JMP @A+DPTR	01110011	1	1	2	$(PC) \leftarrow (A) + (DPTR)$
Переход, если аккумулятор равен нулю	JZ rel	01100000	5	2	2	$(PC) \leftarrow (PC) + 2$ , если $(A) = 0$ , то $(PC) \leftarrow (PC) + rel$
Переход, если аккумулятор не равен нулю	JNZ rel	01110000	5	2	2	$(PC) \leftarrow (PC) + 2$ , если $(A) \neq 0$ , то $(PC) \leftarrow (PC) + rel$
Переход, если перенос равен единице	JC rel	01000000	5	2	2	$(PC) \leftarrow (PC) + 2$ , если $(C) = 1$ , то $(PC) \leftarrow (PC) + rel$
Переход, если перенос равен нулю	JNC rel	01010000	5	2	2	$(PC) \leftarrow (PC) + 2$ , если $(C) = 0$ , то $(PC) \leftarrow (PC) + rel$
Переход, если бит равен единице	JB bit, rel	00100000	11	3	2	$(PC) \leftarrow (PC) + 3$ , если $(b) = 1$ , то $(PC) \leftarrow (PC) + rel$

Переход, если бит равен нулю	JNB bit, rel	00110000	11	3	2	$(PC) \leftarrow (PC) + 3$ , если $(b) = 0$ , то $(PC) \leftarrow (PC) + rel$
Переход, если бит установлен, с последующим сбросом бита	JBC bit, rel	00010000	11	3	2	$(PC) \leftarrow (PC) + 3$ , если $(b) = 1$ , то $(b) \leftarrow 0$ и $(PC) \leftarrow (PC) + rel$
Декремент регистра и переход, если не нуль	DJNZ Rn, rel	11011rrr	5	2	2	$(PC) \leftarrow (PC) + 2$ , $(Rn) \leftarrow (Rn) - 1$ , если $(Rn) \neq 0$ , то $(PC) \leftarrow (PC) + rel$
Декремент прямо адресуемого байта и переход, если не нуль	DJNZ ad, rel	11010101	8	3	2	$(PC) \leftarrow (PC) + 2$ , $(ad) \leftarrow (ad) - 1$ , если $(ad) \neq 0$ , то $(PC) \leftarrow (PC) + rel$
Сравнение аккумулятора с прямоадресуемым байтом и переход, если не равно	CJNE A, ad, rel	10110101	8	3	2	$(PC) \leftarrow (PC) + 3$ , если $(A) \neq (ad)$ , то $(PC) \leftarrow (PC) + rel$ , если $(A) < (ad)$ , то $(C) \leftarrow 1$ , иначе $(C) \leftarrow 0$
Сравнение аккумулятора с константой и переход, если не равно	CJNE A, #d, rel	10110100	10	3	2	$(PC) \leftarrow (PC) + 3$ , если $(A) \neq \#d$ , то $(PC) \leftarrow (PC) + rel$ , если $(A) < \#d$ , то $(C) \leftarrow 1$ , иначе $(C) \leftarrow 0$ .
Сравнение регистра с константой и переход, если не равно	CJNE Rn, #d, rel	10111rrr	10	3	2	$(PC) \leftarrow (PC) + 3$ , если $(Rn) \neq \#d$ , то $(PC) \leftarrow (PC) + rel$ , если $(Rn) < \#d$ , то $(C) \leftarrow 1$ , иначе $(C) \leftarrow 0$
Сравнение байта в РПД с константой и переход, если не равно	CJNE @Ri, #d, rel	1011011i	10	3	2	$(PC) \leftarrow (PC) + 3$ , если $((Ri)) \neq \#d$ , то $(PC) \leftarrow (PC) + rel$ , если $((Ri)) < \#d$ , то $(C) \leftarrow 1$ , иначе $(C) \leftarrow 0$

Длинный вызов подпрограммы	LCALL ad16	00010010	12	3	2	(PC) ← (PC) + 3, (SP) ← (SP) + 1, ((SP)) ← (PC0–7), (SP) ← (SP) + 1, ((SP)) ← (PC8–15), (PC) ← ad 16
Абсолютный вызов подпрограммы в пределах страницы в 2 Кбайта	ACALL ad 11	a <sub>10</sub> a <sub>9</sub> a <sub>8</sub> 10001	11 6	2	2	(PC) ← (PC) + 2, (SP) ← (SP) + 1, ((SP)) ← (PC0–7), (SP) ← (SP) + 1, ((SP)) ← (PC8–15)
Возврат из подпрограммы	RET	00100010	1	1	2	(PC0–10) ← ad11
Возврат из подпрограммы обработки прерывания	RETI	00110010	1	1	2	(PC8–15) ← ((SP)), (SP) ← (SP) – 1, (PC0–7) ← ((SP)), (SP) ← (SP) – 1, (PC8–15) ← ((SP)), (SP) ← (SP) – 1, (PC0–7) ← ((SP)), (SP) ← (SP) – 1
Холостая команда	NOP	00000000	1	1	1	(PC) ← (PC) + 1

**Примечание:** Ассемблер допускает использование обобщенного имени команд JMP и CALL, которые в процессе трансляции заменяются оптимальными по формату командами перехода (AJMP, SJMP, LJMP) или вызова (ACALL, LCALL).

**ПРИЛОЖЕНИЕ 6**

**ПРИМЕР ПРОГРАММЫ, РЕАЛИЗУЮЩЕЙ ОБРАБОТКУ  
РЕЗУЛЬТАТОВ ИЗМЕРЕНИЯ ТЕМПЕРАТУРЫ ВСТРОЕННЫМ  
ДАТЧИКОМ МИКРОКОНТРОЛЛЕРА C8051F300**

```
-----  
; Temp_2.ASM  
-----  
; Copyright 2001, Cygnal Integrated Products, Inc.  
;  
; FILE:          Temp_2.ASM  
; DEVICE:       C8051F30x  
; ASSEMBLER:   Keil A51  
;  
; This program provides an example of how to configure the on-chip temperature sensor  
; with the ADC.  
; The calibrated on-chip 24.5MHz internal oscillator is used as the system clock source.  
;  
; The ADC is configured for left-justified mode, GAIN = 2, using Timer2 overflows as  
; the start-of-conversion source. Timer2 is configured in auto-reload mode to overflow  
; every 10ms. The ADC conversion complete interrupt handler reads the ADC value and  
; compares it with the expected value for room temperature (about 25 degrees C), stored  
; in ROOMCODE. If the measured temperature is below this value, the LED is turned  
; off. If the measured value is above ROOMCODE, the LED is turned on.  
;  
; The LED switch point can be easily modified by changing the value of  
; ROOMCODE.  
;  
-----  
;  
-----  
; EQUATES  
-----  
  
$INCLUDE (C8051F300.inc)  
  
LED      EQU    P0.2          ; LED on target board ('1' is LED ON)  
  
SYSCLK   EQU    24500        ; SYSCLK frequency in kHz  
  
TC_10ms  EQU    (SYSCLK / 12) * 10 ; number of timer counts in 10ms  
  
ROOMCODE EQU    0x95         ; 95 ADC value for 25 degrees C.  
  
-----  
; VARIABLES
```

```

;-----
;
;-----
; STACK

STACK    SEGMENT IDATA    ; declare STACK segment
         RSEG STACK
         DS 80h            ; reserve 128 bytes for stack

;-----
; MACRO DEFINITIONS
;-----

;-----
; RESET AND INTERRUPT VECTOR TABLE
;-----

        CSEG AT 0
        ljmp Main

        org 43h
        ljmp ADC0_ISR    ; ADC0 end of conversion interrupt

;-----
; MAIN PROGRAM CODE
;-----

Temp_2   SEGMENT CODE    ; declare CODE segment
         RSEG Temp_2    ; select CODE segment
         USING 0        ; using register bank 0

Main:
        ; Disable the WDT.
        anl PCA0MD, #NOT(040h) ; clear Watchdog Enable bit

        mov SP, #STACK-1    ; init stack pointer

        acall SYSCLK_Init    ; initialize system clock source

        acall PORT_Init     ; initialize crossbar and GPIO
        acall ADC0_Init     ; initialize ADC0 and temp sensor
        acall Timer2_Init   ; initialize Timer2

        setb TR2            ; START Timer2
        setb AD0EN         ; enable ADC

```

```

setb EA                ; enable global interrupts

sjmp $                 ; spin forever

;-----
; INTERRUPT VECTORS
;-----
;-----
; ADC0_ISR
;
; This ISR is activated on the completion of an ADC sample. When this event occurs,
; the ADC value is copied to the holding variable TEMPCODE, and is compared with
; the code for 25 degrees C. If the temperature is above 25 degrees C, the LED is turned
; on. If the temperature is below 25 degrees C, the LED is turned off.
;
ADC0_ISR:
    push PSW            ; preserve registers
    push acc

    clr AD0INT         ; clear ADC0 interrupt flag

    clr C
    mov a, ADC0        ; read ADC value
    subb a, #ROOMCODE  ; subtract room constant

    ; compare TEMPCODE with value expected for 25 degrees C
    ; if (TEMPCODE - ROOMDEG) < 0, then turn LED off, otherwise,
    ; turn it on.

    setb LED           ; turn LED on.
    jnc ADC0_ISR_END   ; exit if subtract result was
                        ; positive,
    clr LED            ; otherwise, turn LED off then exit

ADC0_ISR_END:
    pop acc
    pop PSW

    reti

;-----
; SUBROUTINES
;-----
;-----
; SYSCLK_Init
;-----

```

```

; This routine initializes the system time base by selecting the internal
; 24.5MHz oscillator as the system clock source. Also enables the missing
; clock detector.
;

```

```

SYSCLK_Init:

```

```

    mov  OSCICN, #07h          ; select internal oscillator at
                                ; 24.5MHz as the system clock
                                ; source
    mov  RSTSRC, #04h         ; enable missing clock detector

    ret

```

```

;-----
; PORT_Init
;-----

```

```

; This routine initializes the crossbar and GPIO pins.
;

```

```

PORT_Init:

```

```

    orl  XBR0, #04h           ; skip LED pin in crossbar
                                ; assignments
    mov  XBR2, #40h          ; Enable crossbar and weak pull-ups

    orl  P0MDOUT, #04h       ; enable P0.2 (LED on target board)
                                ; as push-pull
    orl  P0MDIN, #04h       ; enable P0.2 (LED) as digital
                                ; input

    ret

```

```

;-----
; Timer2_Init
;-----

```

```

; This routine initializes Timer2 in 16-bit auto-reload mode to overflow
; at 100Hz using SYSCLK/12 as its time base. Exits with Timer2
; stopped and Timer2 interrupts disabled.
;

```

```

Timer2_Init:

```

```

    mov  TMR2CN, #00h        ; STOP Timer2, clear TF2H and TF2L;
                                ; disable low-byte interrupt;
                                ; disable split mode; select
                                ; internal prescaler as timebase
    anl  CKCON, #NOT(060h)   ; Timer2 uses SYSCLK/12
    mov  TMR2RLH, #HIGH(-TC_10ms); init reload values
    mov  TMR2RLL, #LOW(-TC_10ms)
    mov  TMR2H, #0ffh        ; set to auto-reload immediately
    mov  TMR2L, #0ffh
    clr  ET2                 ; disable Timer2 interrupts

```

```

ret

;-----
; ADC0_Init
;-----
; This routine initializes ADC0 for monitoring the on-chip temp sensor at a
; gain of '2'. Enables the End of Conversion interrupt. Leaves ADC in
; disabled state.
;
;
ADC0_Init:
    mov  ADC0CN, #02h          ; ADC0 disabled, normal
                                ; track mode, ADC0 conversions
                                ; initiated by overflow of
                                ; Timer2
    mov  AMX0SL, #0f8h        ; select temp sensor as ADC0 input

                                ; ADC conversion clock <= 7.5 MHz
    mov  ADC0CF, #((SYSCLK / 7500) SHL 3)

    orl  ADC0CF, #02h         ; PGA gain = 2

    mov  REF0CN, #0eh         ; enable temp sensor; VREF = VDD;
                                ; bias generator is on
    orl  EIE1, #04h          ; enable ADC0 End of Conversion
                                ; Interrupt
    ret

;-----
; End of file.

END

```

Учебное издание

**МИКРОКОНТРОЛЛЕРЫ  
В СИСТЕМАХ УПРАВЛЕНИЯ  
И ИЗМЕРЕНИЯ**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
К ЛАБОРАТОРНОМУ ПРАКТИКУМУ**

Для студентов специальностей  
G 1-31 04 02 «Радиофизика»,  
G 1-31 04 03 «Физическая электроника»,  
специализации E 1-25 01 10 08  
«Коммерческая деятельность на рынке  
радиоэлектронных средств и информационных услуг»

**В ДВУХ ЧАСТЯХ**

**Часть 2**

**Автор - составитель  
Шалатонин Иван Алексеевич**

В авторской редакции

Технический редактор *Г. М. Романчук*  
Корректор *Л. Н. Масловская*

Ответственный за выпуск *Т. М. Турчиняк*

Подписано в печать 12.05.2004. Формат 60×84/16. Бумага офсетная.  
Гарнитура Таймс. Усл. печ. л. 4,88. Уч.-изд. л. 4,24. Тираж 50 экз. Зак.

Белорусский государственный университет.  
Лицензия на осуществление издательской деятельности  
№ 02330/0056804 от 02.03.2004.

220050, Минск, проспект Франциска Скорины, 4.

Отпечатано на копировально-множительной технике  
факультета радиофизики и электроники  
Белорусского государственного университета.  
220064, Минск, ул. Курчатова, 5.

№3 Содержание  
№4 - 11

Учебное издание

**МИКРОКОНТРОЛЛЕРЫ  
В СИСТЕМАХ УПРАВЛЕНИЯ  
И ИЗМЕРЕНИЯ**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
К ЛАБОРАТОРНОМУ ПРАКТИКУМУ**

Для студентов специальностей

С 1-31 04 02 «Радиофизика»,

С 1-31 04 03 «Вычислительная электроника,  
связанные с К 1-25 01 10 08

«Коммерческая деятельность на рынке  
радиоэлектронных средств и информационных услуг»

**В ДВУХ ЧАСТЯХ**

**Часть 2**

**Автор-составитель  
Шалахов Иван Александрович**

Ивановский институт

Технический редактор *Г. М. Рыжова*

Корректор *Л. В. Митюшина*

Ответственный за выпуск *Г. М. Рыжова*

Издано в печать 12.01.2004. Формат 60×84/16. Печать офсетная.  
Гарнитура Таймс. Усл. сетр. л. 4,88. Усл. изд. л. 8,21. Тираж 50 экз. За.

Копирование в сети, электронный университет

Информация об авторском праве и интеллектуальной собственности

№ 023/090/0094 от 01.01.2004

129010, Москва, проспект Фрунзенский Стороны, 4

Открыто для взаимодействия с общественностью

Информация о работе доступна в электронном

Базирование на государственном уровне университета

129084, Москва, ул. Кутузовская 3