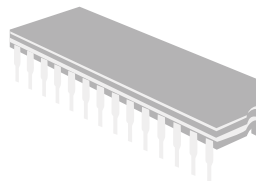
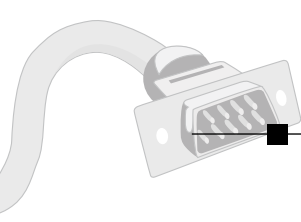


# Оглавление

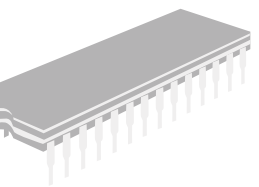
---

<b>Введение</b> .....	6
<b>Глава 1. Программная архитектура микроконтроллеров 8051</b> .....	10
<b>1.1.</b> Структура внутренней памяти 8051.....	12
<b>1.2.</b> Подключение внешней памяти программ и данных.....	16
<b>1.3.</b> Система команд микроконтроллера семейства 8051 .....	17
<b>1.4.</b> Система прерываний .....	23
<b>1.5.</b> Параллельные порты ввода/вывода данных .....	29
<b>Глава 2. Программирование и отладка в среде Keil uVision</b> .....	32
<b>2.1.</b> Преимущества и недостатки языков высокого уровня.....	33
<b>2.2.</b> Создание программ в Keil C51 .....	34
<b>2.3.</b> Синтаксис Keil C51 .....	45
<b>2.3.1.</b> Символы, ключевые слова и идентификаторы .....	45
<b>2.3.2.</b> Форматы данных в Keil C51 .....	48
<b>2.3.3.</b> Специальные ключевые слова Keil C51 .....	49
<b>2.3.4.</b> Операторы и выражения в Keil C51 .....	54
<b>2.3.5.</b> Файлы заголовков Keil C51 .....	55
<b>2.4.</b> Управление вводом/выводом в Keil C51 .....	57
<b>2.5.</b> Операции с памятью .....	59
<b>2.6.</b> Программирование ввода/вывода через последовательный порт.....	60
<b>2.7.</b> Интерфейс с языком ассемблера .....	64
<b>2.7.1.</b> Встроенный ассемблерный код .....	64
<b>2.7.2.</b> Подпрограммы на ассемблере.....	71





2.8. Программирование на языке ассемблера в среде Keil.....	74
2.9. Отладка программ в среде Keil uVision .....	83
<b>Глава 3. Использование последовательного порта .....</b>	<b>92</b>
3.1. Запись данных в последовательный порт .....	94
3.2. Чтение данных из последовательного порта.....	102
3.3. Прерывание последовательного порта .....	103
3.4. Работа с последовательным портом в Keil C51.....	108
3.5. Интерфейс систем на базе 8051 с персональным компьютером .....	110
<b>Глава 4. Встроенные таймеры .....</b>	<b>117</b>
4.1. Режим работы таймера в качестве 16-разрядного таймера .....	119
4.2. Прерывания таймеров .....	124
4.3. Режим автоперезагрузки.....	128
4.4. Счетчики событий.....	130
4.5. Таймер 2 .....	133
4.5.1. Режим автоперезагрузки таймера 2.....	134
4.5.2. Режим захвата таймера 2.....	137
4.6. Аппаратно-программные решения с использованием таймеров .....	145
4.6.1. Измерение частоты.....	145
4.6.2. Широтно-импульсная модуляция .....	153
<b>Глава 5. Обработка дискретных сигналов .....</b>	<b>158</b>
5.1. Обработка входных данных с использованием SPI.....	161
5.2. Пользовательские интерфейсы ввода дискретных данных .....	174
5.3. Пользовательские интерфейсы вывода дискретных данных .....	186
<b>Глава 6. Ввод/вывод аналоговых сигналов .....</b>	<b>192</b>
6.1. Обработка аналоговых входных сигналов.....	193
6.2. Использование цифро-аналоговых преобразователей .....	205





---

<b>Глава 7. Отображение информации в системах с микроконтроллерами 8051</b> .....	208
<b>7.1.</b> Применение семисегментных индикаторов .....	209
<b>7.2.</b> Применение жидкокристаллических индикаторов .....	213
<b>Заключение</b> .....	224

# Введение

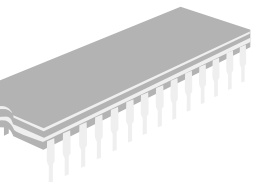
---

Разработка систем управления и контроля с использованием однокристалльных микроконтроллеров в настоящее время переживает настоящий бум. Системы на базе микроконтроллеров используются практически во всех сферах жизнедеятельности человека, и каждый день появляются все новые и новые области применения этих устройств. В последнее время в связи с бурным развитием электроники и схемотехники расширились возможности и самих микроконтроллеров, позволяющие выполнять многие задачи, ранее недоступные для реализации, такие, например, как обработка аналоговых сигналов. Одним из наиболее ранних микроконтроллеров, появившихся на рынке, является микроконтроллер 8051, разработанный фирмой Intel более двадцати лет назад. Несмотря на столь приличный возраст, классический 8051 и его клоны в настоящее время остаются одними из наиболее популярных при разработке систем управления и контроля. Хорошо продуманная архитектура и интуитивно понятная система команд оказывают решающее влияние на выбор многих разработчиков аппаратно-программных систем.

Да и сами микроконтроллеры линейки 8051 постоянно развиваются, предлагая разработчику все новые и новые возможности. На основе базового кристалла 8051 созданы и успешно применяются устройства с развитой периферией и большими объемами памяти. Программирование микроконтроллеров в настоящее время значительно упростилось благодаря инструментальным средствам высокого уровня, разработанным ведущими фирмами. Сегодня микроконтроллеры можно программировать на языках C, Pascal, Basic, Forth и др., что во многом облегчает жизнь программистам, не знакомым с аппаратной частью этих устройств.

Эта книга посвящена практическим аспектам разработки систем на основе микроконтроллера 8051. В отличие от большинства книг по данной тематике, здесь приводятся примеры создания реальных систем и решения задач, с которыми разработчики сталкиваются каждый день. Любой разработчик знает, какие трудности возникают при решении реальной задачи, такой, например, как создание системы сбора и анализа аналоговой информации. В первую очередь это трудности с получением необходимой информации по разработке более-менее сложных систем, поскольку ни одна фирма в мире или разработчик такую информацию так просто не отдадут.

В настоящее время на рынке присутствует очень много литературы и документации по микроконтроллерам для разработчиков начального уровня, но очень мало литературы для более опытных категорий читателей. Предлагаемая вашему вниманию книга призвана восполнить этот пробел. Здесь наряду с основными сведениями об аппаратно-программной архитектуре микроконтроллеров 8051 приводятся многочисленные проекты разработки систем обработки аналоговых и дискретных данных, вывода и отображения информации. Все примеры





аппаратно-программных систем, приведенные в книге, разработаны и протестированы на плате развития Rita-51 фирмы Rigel Corp. и являются полностью работоспособными. Читатели без особого труда смогут адаптировать и усовершенствовать приведенный в книге программный код при разработке собственных систем обработки данных и контроля.

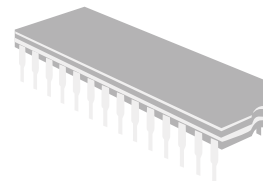
Книга рассчитана на широкий круг читателей – от начинающих до опытных разработчиков – и может оказаться полезной для всех, кто желает самостоятельно изучить аппаратно-программную архитектуру микроконтроллеров 8051 и применить эти знания на практике.

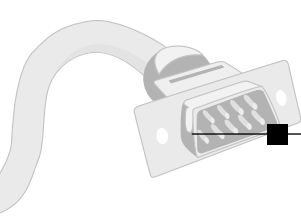
## Структура книги

Структура книги такова, что материал можно изучать выборочно, отдельными главами или последовательно, начиная с первой главы. Это позволяет различным категориям читателей изучать тот материал, который им более всего интересен.

Книга состоит из 7 глав; краткий обзор каждой из них:

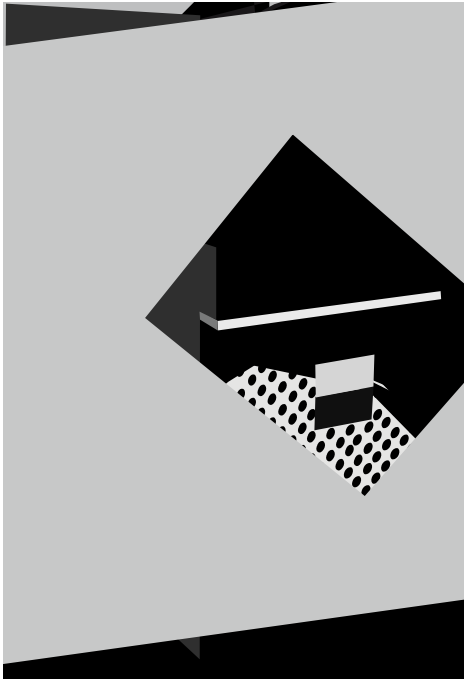
- глава 1 «Программная архитектура микроконтроллеров 8051». В этой главе рассматриваются общие вопросы функционирования микроконтроллеров семейства 8051, аппаратная и программная архитектура базовой модели, включая систему прерываний, ввода/вывода, интерфейсы с внешней памятью программ и данных. Значительная часть материала посвящена описанию системы команд микроконтроллера 8051;
- глава 2 «Программирование и отладка в среде Keil uVision». Материал этой главы посвящен вопросам разработки и отладки программ для 8051 в популярной среде Keil uVision. Рассматриваются вопросы программирования на языке C51 и ассемблере, отладки программного обеспечения. Приводятся практические примеры разработки, компиляции и отладки программ в среде Keil uVision;
- глава 3 «Использование последовательного порта». В этой главе детально проанализированы принципы функционирования последовательного порта микроконтроллера 8051. Рассматриваются многочисленные примеры программного кода обмена данными с использованием последовательного порта. Приводятся практические примеры создания программного интерфейса с персональным компьютером, работающим под управлением операционной системы Windows XP;
- глава 4 «Встроенные таймеры». Эта глава содержит материал по программированию таймеров микроконтроллеров 8051/8052. Дается исчерпывающая информация по аппаратно-программной архитектуре таймеров, подкрепленная многочисленными примерами программирования. Значительная часть главы посвящена применению таймеров в разработке реальных систем с детальным анализом программного кода;
- глава 5 «Обработка дискретных сигналов». Материал главы посвящен вопросам разработки и программирования систем обработки дискретных сигналов. В главе проанализированы основные аппаратно-программные интерфейсы I2C и SPI, а также приведены примеры разработки систем ввода-вывода дискретных данных с использованием этих интерфейсов. Большое внимание уделено разработке пользовательских интерфейсов с детальным анализом их программного кода;
- глава 6 «Ввод/вывод аналоговых сигналов». В этой главе рассматривается широкий круг вопросов, связанных с обработкой аналоговых входных сигналов, а также принципы генерации аналоговых выходных сигналов. На практических примерах показано создание реальных систем сбора данных при использовании аналого-цифровых преобразователей с детальным анализом программного кода. Приводится практический пример разработки цифро-аналогового преобразователя;





- глава 7 «Отображение информации в системах с микроконтроллерами 8051». Материал главы затрагивает вопросы, связанные с отображением визуальной информации в системах на базе микроконтроллера 8051. Рассмотрены практические примеры построения простых систем отображения информации с использованием семисегментных светодиодных и жидкокристаллических индикаторов. Анализируются вопросы создания пользовательских систем отображения информации с использованием жидкокристаллических дисплеев.

Автор благодарит коллектив издательства «ДМК Пресс» за помощь при подготовке книги к изданию. Особая признательность жене Юлии за поддержку и помощь при написании книги.



# Программная архитектура микроконтроллеров 8051

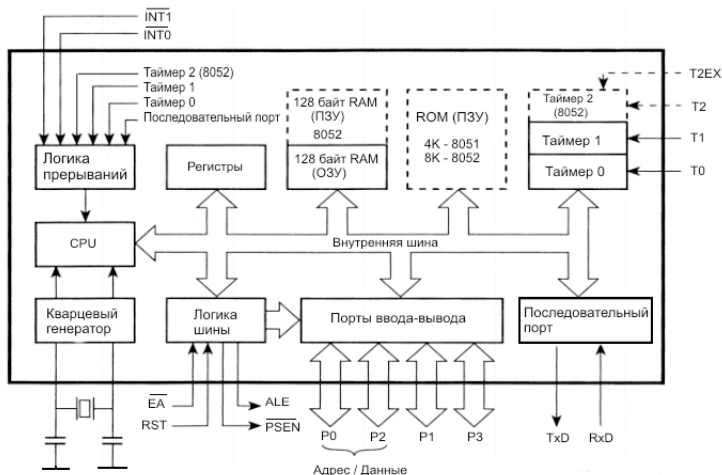
<b>1.1.</b>	Структура внутренней памяти 8051 .....	12
<b>1.2.</b>	Подключение внешней памяти программ и данных .....	16
<b>1.3.</b>	Система команд микроконтроллера семейства 8051.....	17
<b>1.4.</b>	Система прерываний.....	23
<b>1.5.</b>	Параллельные порты ввода/вывода данных .....	29

# Программная архитектура микроконтроллеров 8051

В этой главе мы рассмотрим основные функциональные узлы популярных микроконтроллеров семейства 8051/8052 и принципы их работы. Здесь же вкратце рассмотрим и систему команд 8051, которая нам пригодится при создании аппаратно-программных проектов последующих глав.

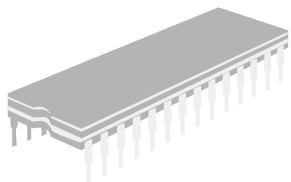
Аппаратная архитектура 8051 представлена на рис. 1.1.

**Рис. 1.1.**  
Функциональная  
схема аппаратной  
части 8051

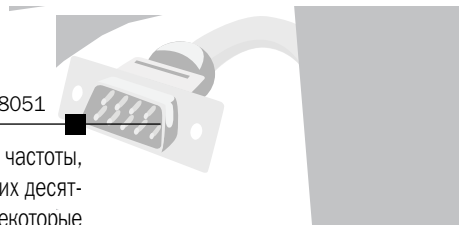


В микроконтроллере 8051 все вычисления выполняются в арифметико-логическом устройстве, являющемся частью базового процессорного модуля (CPU). Обмен данными, находящимися в оперативной памяти микроконтроллера, а также считывание команд выполняется по внутренней шине 8051. По этой шине осуществляется и обмен данными с портами ввода/вывода P1 – P3, с последовательным портом и таймерами. Внутренний контроллер шины формирует необходимые сигналы (EA, ALE, PSEN, RD/WR) для работы с внешней памятью программ и данных, а также сигнал сброса/начальной установки RST.

Микроконтроллеры 8051 рассчитаны на работу с системами реального времени, которые могут генерировать определенные сигналы, требующие немедленной реакции микроконтроллера. Для обработки таких сигналов (или событий) служит аппаратно реализованная логика прерываний, позволяющая обрабатывать сигналы внешних источников, таймеров и последовательного порта.

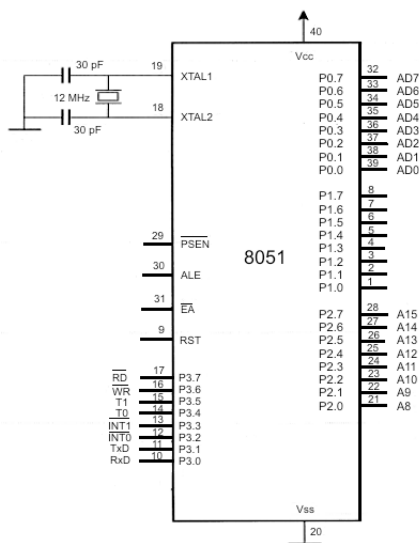






Скорость выполнения операций в системе на базе 8051 зависит от тактовой частоты, с которой работает кристалл и которая может варьироваться от единиц до нескольких десятков мегагерц. В архитектуру классического микроконтроллера 8051 были внесены некоторые изменения (к двум существующим таймерам добавлен третий, а также расширена внутренняя память), которые привели к созданию устройства 8052, наиболее популярного в настоящее время.

Микроконтроллер 8051 реализован в виде однокристалльного устройства с внешними выводами, обозначенными, как показано на рис. 1.2.



**Рис. 1.2.**

Схема расположения выводов 8051

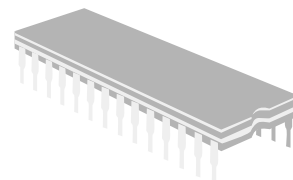
Входные и выходные сигналы микроконтроллера 8051 имеют следующие назначения:

- XTAL и XTAL2 – входы подключения кварцевого резонатора для работы генератора тактовой частоты микроконтроллера;
- PSEN – сигнал, используемый при обращении к внешней памяти программ;
- ALE – выходной сигнал разрешения фиксации адреса при обращении к внешней памяти программ/данных;
- EA – сигнал, блокирующий работу с внутренней памятью;
- RST – сигнал общего сброса;
- P0 – P3 – выводы портов ввода/вывода микроконтроллера;
- Vss и Vcc – выводы подачи напряжения питания.

Порты P0, P2 и P3 помимо функционирования в режиме ввода/вывода дискретных сигналов могут выполнять, в зависимости от аппаратной конфигурации, и другие функции. Так, через порт P0 при обращениях к внешней памяти выставляются младшие 8 бит 16-разрядного адреса, а затем, в фазе записи/чтения данных, через этот порт идет обмен данными. Порт P2 при обращениях к внешней памяти служит источником старших 8 бит 16-разрядного адреса.

Выводы порта P3 микроконтроллера 8051 имеют следующие альтернативные назначения:

- P3.0 – вход приема данных в последовательный порт;
- P3.1 – выход передачи данных с последовательного порта;
- P3.2 – вход внешнего прерывания INTO;



- P3.3 – вход внешнего прерывания INT1;
- P3.4 – вход управления таймером 0;
- P3.5 – вход управления таймером 1;
- P3.6 – выход сигнала записи в память;
- P3.7 – выход сигнала чтения из памяти.

Для использования альтернативных функций порта P3 необходимо настроить соответствующим образом программное обеспечение системы 8051.

### 1.1. Структура внутренней памяти 8051

Микроконтроллеры 8051 оперируют двумя типами памяти: памятью программ и памятью данных. Память данных может быть реализована как комбинация размещенного на кристалле (резидентного или on-chip) статического ОЗУ и внешних микросхем памяти. Для простых аппаратно-программных конфигураций с применением 8051 бывает достаточно резидентной памяти самого микроконтроллера.

Программный код размещается в памяти программ, которая физически может быть реализована в виде однократно программируемого устройства (EPROM), перепрограммируемого устройства (EEPROM) или флеш-памяти. Если для записи программ используется EPROM или EEPROM, то программный код обычно располагается во внешнем по отношению к микроконтроллеру устройстве. В подавляющем большинстве современных микроконтроллеров 8051 память программ располагается во флеш-памяти, находящейся, так же как и резидентная память данных, на одном кристалле.

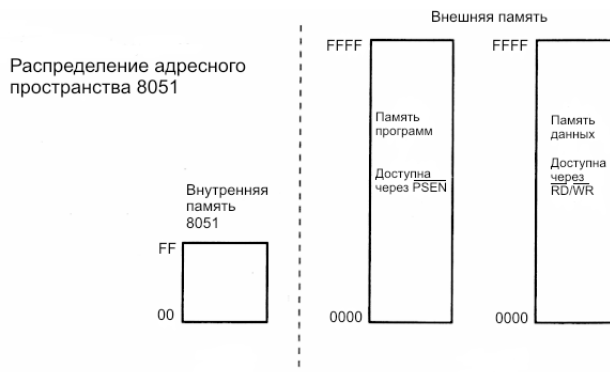
Память программ и память данных физически и логически разделены, имеют различные механизмы адресации, работают под управлением различных сигналов и выполняют разные функции.

Память программ может иметь максимальный объем, равный 64 Кб, что обусловлено использованием 16-разрядной шины адреса. Во многих случаях емкость памяти программ, размещенной на кристалле 8051, ограничена 4, 8 или 16 Кб. В память программ кроме команд могут записываться константы, управляющие слова инициализации, таблицы перекодировки входных и выходных переменных и т.п. Доступ к содержимому памяти программ осуществляется посредством 16-битовой шины адреса. Сам адрес формируется с помощью либо программного счетчика (PC), либо регистра-указателя данных (DPTR). DPTR выполняет функции базового регистра при косвенных переходах по программе или используется в операциях с таблицами.

Общая структура памяти микроконтроллера 8051 показана на рис. 1.3.

Рис. 1.3.

Общая структура памяти



Рассмотрим более подробно резидентную (on-chip) память микроконтроллера 8051. Резидентная память, изображенная в левой части рис. 1.3, состоит из двух частей: внутреннего ОЗУ размером 128 байт и памяти, выделяемой для регистров специальных функций (Special Function Registers, SFR). Внутреннее ОЗУ имеет структуру, показанную на рис. 1.4.



**Рис. 1.4.**  
Внутреннее ОЗУ

Для доступа к данным, размещенным во внутреннем ОЗУ, используется однобайтовый адрес. Архитектура внутренней памяти данных 8051 позволяет обращаться к отдельным битам данных в специально выделенной области внутреннего ОЗУ, начиная с адреса 0x20 и заканчивая 0x2F (см. рис. 1.4). Таким образом, в указанном диапазоне адресов можно обращаться к 128-битовым переменным с помощью команд битовых операций SETB и CLR. Битовые переменные нумеруются, начиная с 0x0 и заканчивая 0x7F. Это не означает, что нельзя обращаться к этим ячейкам памяти, как к байтам при обычных операциях с памятью.

Например, для установки бит 0 и 1 в области памяти начиная с 0x20, можно выполнить команды

```
SETB 00h
SETB 01h
```

или команду

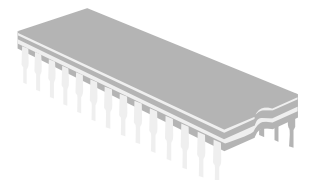
```
ORL 20h, 0x3
```

Во втором случае обращение выполняется к байту по адресу 0x20, а установка соответствующих битов выполняется операцией «логическое ИЛИ».

Во внутреннем ОЗУ микроконтроллера 8051 выделены 4 банка регистров общего назначения. При включении микроконтроллера банком по умолчанию становится банк 0 (см. рис. 1.4). При этом регистру R0 соответствует адрес 0x00, регистру R1 – адрес 0x01, наконец, регистру R7 при использовании банка 0 соответствует адрес 0x07. Если банком по умолчанию становится, например, банк 1, то регистру R0 будет соответствовать адрес 0x08, регистру R1 – адрес 0x09 и регистру R7 – адрес 0x0F.

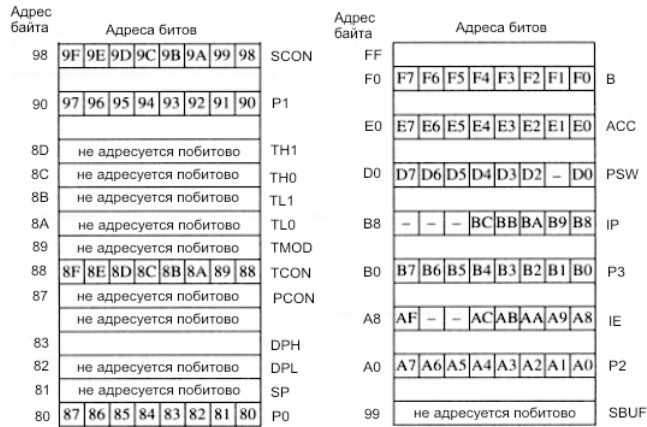
К адресному пространству внутреннего ОЗУ начиная с адреса 0x80 примыкают и адреса регистров специальных функций (рис. 1.5).

Регистры специальных функций (Special Function Registers, SFR) предназначены для управления ходом вычислительных операций, а также отвечают за инициализацию, настройку и управление портами ввода/вывода, таймерами, последовательным портом. Кроме того, регистры специальных функций содержат информацию о приоритетах прерываний, а также биты



управления разрешением прерываний. Регистры специальных функций с указанием их назначения перечислены в табл. 1.1.

**Рис. 1.5.**  
Регистры специальных функций



**Таблица 1.1.**  
Назначение регистров специальных функций

Обозначение	Описание	Адрес
A	Аккумулятор	0E0H
B	Регистр-расширитель аккумулятора	0F0H
PSW	Слово состояния программы	0D0H
SP	Регистр-указатель стека	81H
DPTR	Регистр-указатель данных (DPH)	83H
	(DPL)	82H
P0	Порт 0	80H
P1	Порт 1	90H
P2	Порт 2	0A0H
P3	Порт 3	0B0H
IP	Регистр приоритетов прерываний	0B8H
IE	Регистр маски прерываний	0A8H
TMOD	Регистр режима таймера/счетчика	89H
TCON	Регистр управления/статуса таймера	88H
TH0	Таймер 0 (старший байт)	8CH
TLO	Таймер 0 (младший байт)	8AH
TH1	Таймер 1 (старший байт)	8DH
TL1	Таймер 1 (младший байт)	8BH
SCON	Регистр управления приемопередатчиком	98H
SBUF	Буфер приемопередатчика	99H
PCON	Регистр управления мощностью	87H

Некоторые регистры специальных функций допускают побитовую адресацию. При этом обращение к отдельным битам такого регистра возможно как с помощью обычных функций для работы с байтами, так и с помощью команд побитовых операций. Например, для запуска таймера 0 можно выполнить команду ассемблера

ORL TCON, #10h

или одну из команд установки бита TCON.4 (TRO):

```
SETB    TCON.4
SETB    TRO
```

Рассмотрим смысл некоторых регистров специальных функций и начнем с аккумулятора и регистра слова состояния (регистра флагов).

Аккумулятор (A) является источником операнда и фиксирует результат при выполнении арифметических, логических операций и ряда операций передачи данных. Кроме того, некоторые операции можно выполнить только с использованием аккумулятора: например, операции сдвигов, проверку на нуль, формирование флага паритета и т.п.

В распоряжении программиста имеются 8 регистров общего назначения R0 – R7 одного из четырех банков. При выполнении многих команд в арифметико-логическом устройстве микроконтроллера формируется ряд признаков операции (флагов), которые фиксируются в регистре PSW. Перечень флагов PSW, их символические имена и условия формирования приведены в табл. 1.2.

Обозначение	Бит	Описание
C	PSW.7	Флаг переноса. Устанавливается и сбрасывается аппаратно или программно при выполнении арифметических и логических операций
AC	PSW.6	Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратно при выполнении команд сложения и вычитания и сигнализирует о переносе или займе в бите 3
FO	PSW.5	Флаг 0. Может быть установлен, сброшен или проверен программой как флаг, специфицируемый пользователем
RS1	PSW.4	Выбор банка регистров. Устанавливается и сбрасывается программно для выбора рабочего банка регистров (табл. 1.3)
RS0	PSW.3	
OV	PSW.2	Флаг переполнения. Устанавливается и сбрасывается аппаратно при выполнении арифметических операций
–	PSW.1	Не используется
P	PSW.0	Флаг четности. Устанавливается и сбрасывается аппаратно в каждом цикле и фиксирует нечетное/четное число единичных битов в аккумуляторе, т.е. выполняет контроль по четности

**Таблица 1.2.**

*Регистр слова состояния микроконтроллера*

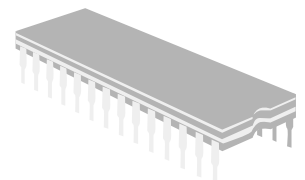
Установки битов RS0 – RS1 при выборе банка регистров показаны в табл. 1.3.

RS1	RS0	Банк	Границы адресов
0	0	0	00h–07h
0	1	1	08h–0Fh
1	0	2	10h–17h
1	1	3	18h–1Fh

**Таблица 1.3.**

*Битовые комбинации для выбора банка регистров*

Среди регистров специального назначения есть регистры, выполняющие функции адресации данных, находящихся в памяти. К ним относятся 8-разрядный указатель стека (SP) и регистр-указатель DPTR.



Указатель стека может адресовать любую область внутренней памяти данных микроконтроллера, при этом содержимое этого регистра инкрементируется перед выполнением команд PUSH и CALL и декрементируется после выполнения команд POP и RET. В процессе инициализации микроконтроллера после сигнала RST в указатель стека автоматически загружается код 0x07. Это значит, что если программа не переопределяет содержимое указателя стека, то первый элемент данных в стеке будет располагаться в ячейке памяти 0x08.

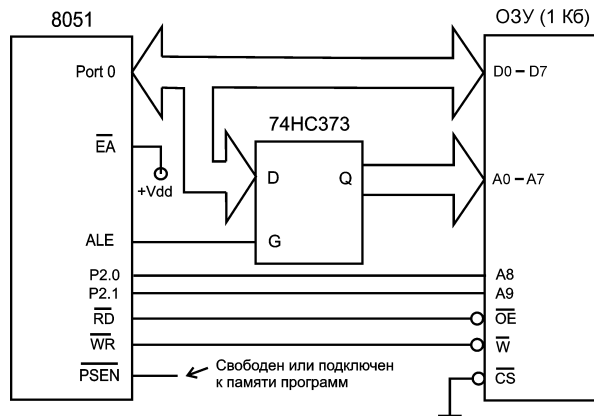
Нужно отметить одну важную особенность: при загрузке данных в стек адрес растет вверх, поэтому если в программе используются банки 1, 2 или 3, то указатель стека следует проинициализировать адресом из неиспользуемой области памяти, например 0x30, чтобы не перезаписать содержимое регистров одного из банков.

Двухбайтный регистр-указатель данных DPTR обычно используется для фиксации 16-разрядного адреса в операциях с обращением к внешней памяти. При работе с DPTR допускает использование старшего и младшего байтов регистра (DPH и DPL соответственно).

## 1.2. Подключение внешней памяти программ и данных

Внешняя память данных может быть подключена к микроконтроллеру приблизительно по такой схеме, какая изображена на рис. 1.6.

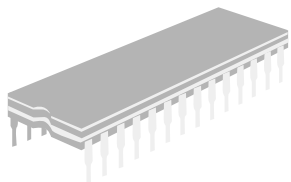
**Рис. 1.6.**  
Подключение  
внешней памяти  
данных

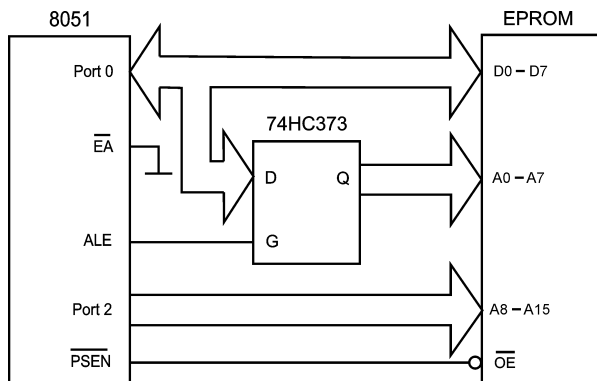


Здесь показан интерфейс микроконтроллера 8051 с внешним модулем ОЗУ емкостью 1 Кб. Адресация памяти по этой схеме реализована следующим образом:

1. Младшие 8 бит адреса выводятся стандартным образом через порт P0 и запоминаются в регистре-защелке 74HC373 по спаду сигнала ALE.
2. На шину адреса подаются старшие биты адреса, из которых используются разряды A8 и A9, устанавливаемые на выводах P2.0 и P2.1 и предназначенные (вместе с установленными в регистре 74HC373 линиями A0 – A7) для выбора адреса в пределах 1 Кб.
3. По низкому уровню одного из сигналов RD (чтение) или WR (запись) осуществляется требуемая операция, при этом байт данных считывается/записывается через порт P0.

Память программ, так же как и память данных, может быть расширена до 64 Кб путем подключения внешних микросхем. Стандартная схема подключения внешней памяти программ осуществляется по схеме, показанной на рис. 1.7.



**Рис. 1.7.**

Подключение  
внешней  
памяти  
программ

Так же как и при обращении к памяти данных, младшая часть адреса памяти формируется портом P0 и запоминается в регистре-защелке по спаду ALE, а старший байт адреса выводится через порт P2. Считывание команды выполняется при подаче низкого уровня на линию PSEN. Поскольку вывод EA подключен к общему проводу, то внутренняя память программ отключается и микроконтроллер при включении начинает выполнение программы с адреса 0x0000 внешней памяти.

### 1.3. Система команд микроконтроллера семейства 8051

Микроконтроллеры семейства 8051 являются микропроцессорными устройствами с архитектурой CISC со стандартным набором команд, характерных для данной архитектуры. Система команд 8051-совместимых устройств включает 111 основных команд размером от одного до трех байт, но большая часть этих команд – одно- или двухбайтовая. Почти все команды выполняются за один или два машинных цикла, что по времени приблизительно равно 1–2 мкс при тактовой частоте 12 МГц, за исключением команд умножения и деления, которые требуют для выполнения четыре машинных цикла.

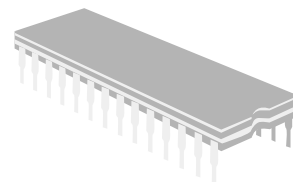
Команды микроконтроллеров 8051 используют прямую, непосредственную, косвенную и неявную адресацию данных. При этом в качестве операндов команд могут выступать отдельные биты, четырехбитовые комбинации (тетрады), байты и слова из двух байт.

В набор команд семейства 8051 входит ряд команд, обеспечивающих выполнение специфических функций управления, например, манипуляции с отдельными битами. Особенностью системы команд 8051 является возможность адресации отдельных бит в памяти данных, а также отдельных бит регистров специальных функций.

По выполняемым функциям команды микроконтроллера 8051 можно разделить на несколько групп:

- пересылки данных;
- арифметических операций;
- логических операций;
- операций над битами;
- передачи управления.

Рассмотрим эти группы команд более подробно, но перед этим условимся при описании мнемоники команд использовать следующие обозначения:



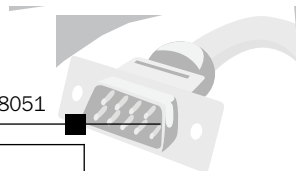
- Rn (n = 0...7) – регистр общего назначения в выбранном банке регистров;
- @Ri (i = 0, 1) – регистр общего назначения в выбранном банке регистров, используемый для формирования косвенного адреса;
- addr – адрес байта;
- src – адрес байта-источника;
- dst – адрес байта-приемника;
- addr11 – 11-разрядный абсолютный адрес перехода;
- addr16 – 16-разрядный абсолютный адрес перехода;
- label – относительный адрес перехода;
- #direct8 – непосредственный операнд размером 1 байт;
- #direct16 – непосредственный операнд размером 2 байта;
- bit – адрес прямо адресуемого бита;
- ~bit – инверсия прямо адресуемого бита;
- A – регистр-аккумулятор;
- PC – регистр-счетчик команд;
- DPTR – 16-разрядный регистр-указатель данных;
- ( ) – содержимое ячейки памяти или регистра.

Команды пересылки данных микроконтроллера 8051 включают 28 команд, краткое описание которых приведено в табл. 1.4.

**Таблица 1.4.**  
Команды пересылки данных

Мнемоника	Описание
MOV A, Rn	(A) ← (Rn)
MOV A, addr	(A) ← (addr)
MOV A, @Ri	(A) ← ((Ri))
MOV A, #direct8	(A) ← #direct8
MOV Rn, A	(Rn) ← (A)
MOV Rn, addr	(Rn) ← (addr)
MOV Rn, #direct8	(Rn) ← #direct8
MOV addr, A	(addr) ← (A)
MOV addr, Rn	(addr) ← (Rn)
MOV dst, src	(dst) ← (src)
MOV addr, @Ri	(addr) ← ((Ri))
MOV addr, #direct8	(addr) ← #direct8
MOV @Ri, A	((Ri)) ← (A)
MOV @Ri, addr	((Ri)) ← (addr)
MOV @Ri, #direct8	((Ri)) ← #direct8
MOV DPTR, #direct16	(DPTR) ← #direct16
MOVC A, @A+DPTR	(A) ← ((A) + (DPTR))
MOVC A, @A+PC	(PC) ← (PC+1), (A) ← ((A)+(PC))
MOVX A, @Ri	(A) ← ((Ri))
MOVX A, @DPTR	(A) ← ((DPTR))





Мнемоника	Описание
MOVX @Ri, A	$((Ri)) \leftarrow (A)$
MOVX @DPTR, A	$((DPTR)) \leftarrow (A)$
PUSH addr	$(SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (addr)$
POP addr	$(addr) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$
XCH A, Rn	$(A) \leftrightarrow (Rn)$
XCH A, addr	$(A) \leftrightarrow (addr)$
XCH A, @Ri	$(A) \leftrightarrow ((Ri))$
XCHD A, @Ri	$(A0 - 3) \leftrightarrow ((Ri0 - 3))$
MOV A, Rn	$(A) \leftarrow (Rn)$

**Таблица 1.4.**  
Команды  
пересылки  
данных  
(окончание)

Команда MOV выполняет пересылку данных из второго операнда в первый. Эта команда не работает с данными, находящимися во внешней памяти данных или в памяти программ. Для работы с данными, находящимися во внешней памяти данных, предназначены команды MOVX, а для работы с константами, записанными в память программ, – команда MOVC. Первая из них обеспечивает чтение/запись байтов из внешней памяти данных, вторая – чтение байтов из памяти программ.

Команды XCH выполняют обмен байтами между аккумулятором и ячейкой памяти, а команда XCHD выполняет обмен данными между младшими тетрадами (биты 0–3).

Команды PUSH и POP предназначены для записи данных в стек и их чтения из стека соответственно. При этом размер стека ограничен лишь размером памяти данных, расположенной на кристалле. В процессе инициализации микроконтроллера после сигнала сброса или при включении питающего напряжения в указатель стека SP заносится код 07H. Таким образом, первый элемент стека будет располагаться в ячейке памяти с адресом 08H.

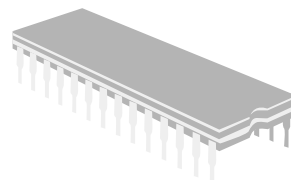
В группе команд пересылок микроконтроллера нет команд для работы с регистрами специальных функций (таймерами, портами ввода/вывода и т.д.). Доступ к таким регистрам осуществляется по их прямому адресу или с использованием их мнемоники, записанной в специальном файле (для программ на ассемблере, например, таким файлом может быть 8051.MCU).

Следует отметить, что регистр-аккумулятор имеет два различных имени в зависимости от способа адресации (A – при неявной адресации, например MOV A, R0; ACC – при использовании прямого адреса).

В группу команд арифметических операций 8051 входят 24 команды, выполняющие операции по обработке целочисленных данных, включая команды умножения и деления. Перечень команд этой группы приведен в табл. 1.5.

Мнемоника	Описание
ADD A, Rn	$(A) \leftarrow (A) + (Rn)$
ADD A, addr	$(A) \leftarrow (A) + (addr)$
ADD A, @Ri	$(A) \leftarrow (A) + ((Ri))$
ADD A, #direct8	$(A) \leftarrow (A) + \#direct8$
ADDC A, Rn	$(A) \leftarrow (A) + (Rn) + (C)$
ADDC A, addr	$(A) \leftarrow (A) + (addr) + (C)$
ADDC A, @Ri	$(A) \leftarrow (A) + ((Ri)) + (C)$

**Таблица 1.5.**  
Команды  
арифметических  
операций



**Таблица 1.5.**

*Команды арифметических операций (окончание)*

Мнемоника	Описание
ADDC A, #direct8	$(A) \leftarrow (A) + \#direct8 + (C)$
DAA	Десятичная коррекция аккумулятора
SUBB A, Rn	$(A) \leftarrow (A) - (Rn) - (C)$
SUBB A, addr	$(A) \leftarrow (A) - (ad) - (C)$
SUBB A, @Ri	$(A) \leftarrow (A) - ((Ri)) - (C)$
SUBB A, #direct8	$(A) \leftarrow (A) - \#direct8 - (C)$
INC A	$(A) \leftarrow (A) + 1$ INC Rn(Rn) < (Rn) + 1
INC Rn	$(Rn) \leftarrow (Rn) + 1$
INC addr	$(addr) \leftarrow (addr) + 1$
INC @Ri	$((Ri)) \leftarrow ((Ri)) + 1$
INC DPTR	$(DPTR) \leftarrow (DPTR) + 1$
DEC A	$(A) \leftarrow (A) - 1$
DEC Rn	$(Rn) \leftarrow (Rn) - 1$
DEC addr	$(addr) \leftarrow (addr) - 1$
DEC @Ri	$((Ri)) \leftarrow ((Ri)) - 1$
MUL AB	$(B)(A) \leftarrow (A) \times (B)$

Результат выполнения команд ADD, ADDC, SUBB, MUL и DIV влияет на флаги слова состояния (PSW) следующим образом:

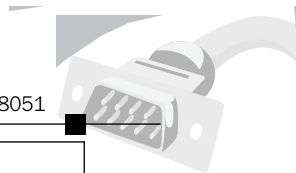
- флаг переноса C устанавливается при переносе из разряда D7 в том случае, если результат операции не помещается в восемь разрядов; флаг дополнительного переноса AC устанавливается при переносе из разряда D3 в командах сложения и вычитания и служит для реализации десятичной арифметики. Этот признак используется командой DAA;
- флаг OV устанавливается при переносе из разряда D6 в случае, если результат операции не помещается в семь разрядов и восьмой не может быть интерпретирован как знаковый. Этот признак служит для организации обработки чисел со знаком;
- флаг четности P устанавливается и сбрасывается аппаратно. Если число единичных битов в аккумуляторе нечетно, то  $P = 1$ , в противном случае  $P = 0$ .

Следующая группа команд, которую мы рассмотрим, – это команды логических операций. Группа содержит 25 команд (табл. 1.6).

**Таблица 1.6.**

*Команды логических операций*

Мнемоника	Описание
ANL A, Rn	$(A) \leftarrow (A) \& (Rn)$
ANL A, addr	$(A) \leftarrow (A) \& (addr)$
ANL A, @Ri	$(A) \leftarrow (A) \& ((Ri))$
ANL A, #direct8	$(A) \leftarrow (A) \& (\#direct8)$
ANL addr, A	$(addr) \leftarrow (addr) \& (A)$
ANL addr, #direct8	$(addr) \leftarrow (addr) \& (\#direct8)$
ORL A, Rn	$(A) \leftarrow (A)   (Rn)$
ORL A, addr	$(A) \leftarrow (A)   (addr)$



Мнемоника	Описание
ORL A, @Ri	$(A) \leftarrow (A) \mid ((Ri))$
ORL A, #direct8	$(A) \leftarrow (A) \mid (\#direct8)$
ORL addr, A	$(addr) \leftarrow (addr) \mid (A)$
ORL addr, #direct8	$(addr) \leftarrow (addr) \mid (\#direct8)$
XRL A, Rn	$(A) \leftarrow (A) \wedge (Rn)$
XRL A, addr	$(A) \leftarrow (A) \wedge (addr)$
XRL A, @Ri	$(A) \leftarrow (A) \wedge ((Ri))$
XRL A, #direct8	$(A) \leftarrow (A) \wedge (\#direct8)$
XRL addr, A	$(addr) \leftarrow (addr) \wedge (A)$
XRL addr, #direct8	$(addr) \leftarrow (addr) \wedge (\#direct8)$
CLR A	$(A) \leftarrow 0$
CPL A	$(A) \leftarrow \sim(A)$
SWAP A	$(A0-3) \leftrightarrow (A4-7)$
RL A	Циклический сдвиг влево
RLC A	Сдвиг влево через перенос
RR A	Циклический сдвиг вправо
RRC A	Сдвиг вправо через перенос

**Таблица 1.6.**  
Команды  
логических  
операций  
(окончание)

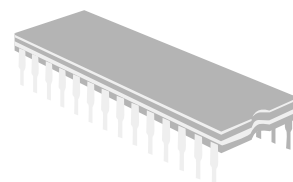
Команды логических операций манипулируют байтами и позволяют выполнить следующие операции:

- логическое И (&);
- логическое ИЛИ (|);
- исключающее ИЛИ (^);
- инверсию (~);
- очистку байта;
- обычные и циклические сдвиги.

Команды операций над битами микроконтроллера 8051 включают 12 команд, позволяющих выполнять операции над отдельными битами: сброс, установку, инверсию, а также «логическое И» (&) и «логическое ИЛИ» (|). В качестве операндов могут выступать 128 бит из внутренней памяти данных микроконтроллера, а также регистры специальных функций, допускающие адресацию отдельных битов. Перечень команд и их мнемоника показаны в табл. 1.7.

Мнемоника	Описание
CLR C	$(C) \leftarrow 0$
CLR bit	$(bit) \leftarrow 0$
SETB C	$(C) \leftarrow 1$
SETB bit	$(bit) \leftarrow 1$
CPL C	$(C) \leftarrow \sim(C)$
CPL bit	$(bit) \leftarrow \sim(bit)$
ANL C, bit	$(C) \leftarrow (C) \& (bit)$

**Таблица 1.7.**  
Команды  
битовых  
операций



Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

[e-Univers.ru](http://e-Univers.ru)