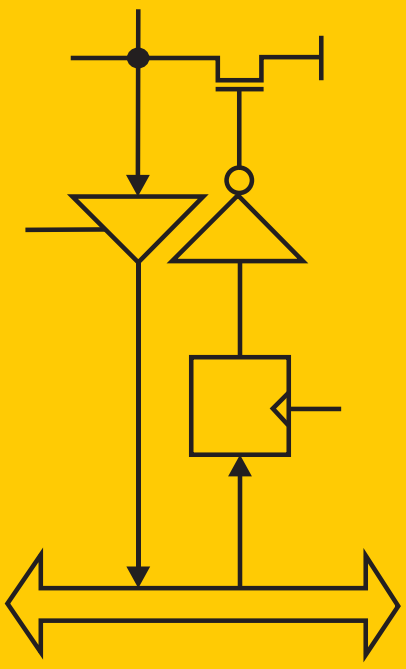
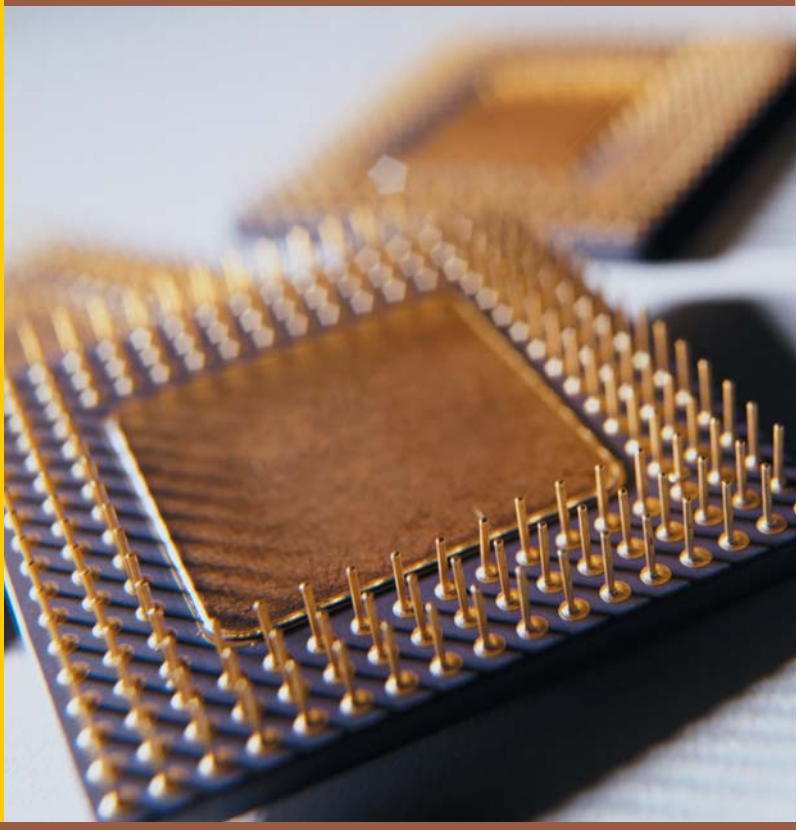
Магда Ю. С.





СОВРЕМЕННЫЕ МИКРОКОНТРОЛЛЕРЫ

АРХИТЕКТУРА, ПРОГРАММИРОВАНИЕ, РАЗРАБОТКА УСТРОЙСТВ

УДК **621.396.6**
ББК **32.872**
М12

Магда Ю. С.
М12 Современные микроконтроллеры. Архитектура, программирование, разработка устройств. — М.: ДМК Пресс, 2017. — 224 с.

ISBN 978-5-97060-551-6

В книге рассматривается широкий круг вопросов, связанных с практическим применением популярных микроконтроллеров 8051 и их расширений в системах управления и контроля. Основной упор сделан на практические аспекты разработки цифровых и аналоговых интерфейсов, использования таймеров, визуализации результатов измерений в системах сбора информации. Значительная часть материала посвящена практическому программированию в популярной среде разработки Keil uVision. Приводятся многочисленные примеры разработки несложных аппаратно–программных систем сбора аналоговой и цифровой информации, измерительных систем, систем управления внешними устройствами и т.д. Все приведенные в книге проекты разработаны и проверены на отладочном модуле Rita-51 фирмы Rigel Corp. и могут служить основой при разработке собственных проектов.

ББК 32.872
УДК 621.396.6

Юрий Степанович Магда

Современные микроконтроллеры
Архитектура, программирование, разработка устройств

Главный редактор Мовчан Д. А.
dmkpress@gmail.com
Корректор Теренина О. А.
Верстка и графика Старцевой Е. М.
Дизайн обложки Мовчан А. Г.

Формат 60x90 1/16

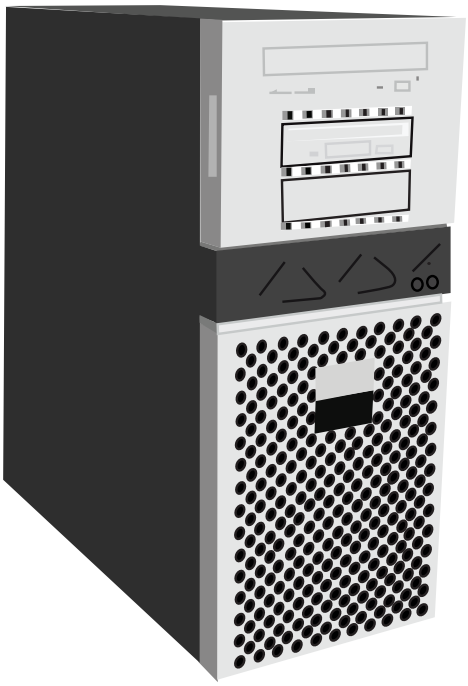
Усл. печ. л. 21. Тираж 100 экз.

Издательство «ДМК Пресс»
Сайт издательства: www.dmkpress.com

ISBN 978-5-97060-551-6

© Магда Ю. С.
© Оформление, ДМК Пресс, 2017

Оглавление



Программная архитектура микроконтроллеров 8051

1 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.

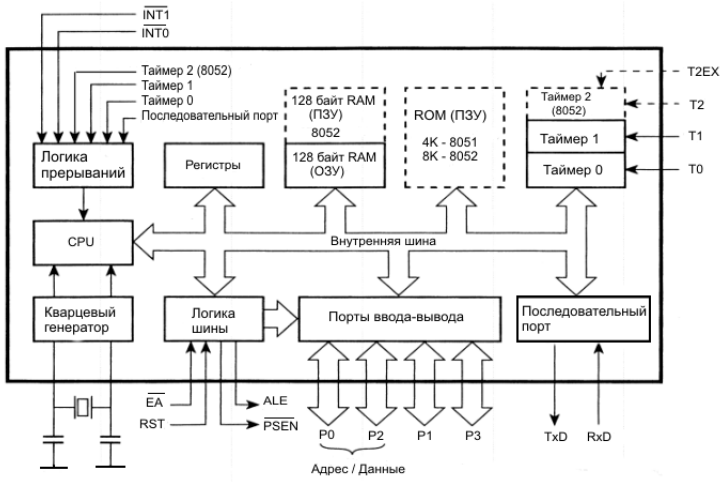


Рис. 1.1. *Функциональная схема аппаратной части 8051*

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 однокристального устройства с внешними выводами, обозначенными, как показано на рис. 1.2.

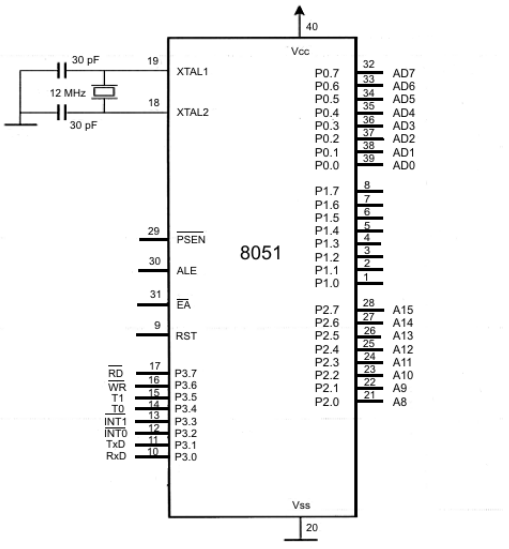


Рис. 1.2.
Схема расположения выводов 8051

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 INT0;

- 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.

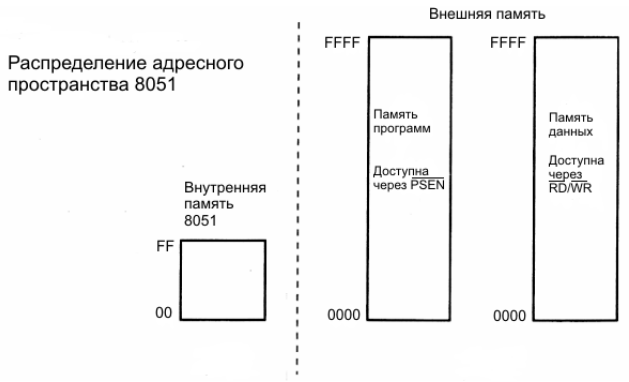


Рис. 1.3. *Общая структура памяти*

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*

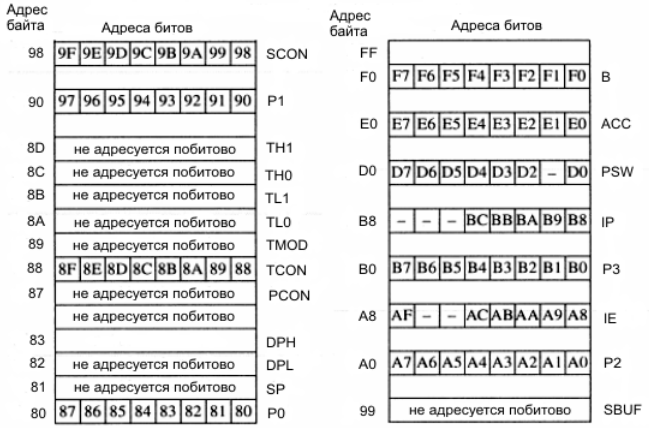


Таблица 1.1. *Назначение регистров специальных функций*

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
TL0	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 (TR0):

```
SETB    TCON.4
SETB    TR0
```

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.

Таблица 1.2. *Регистр слова состояния микроконтроллера*

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
F0	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

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

Таблица 1.3. *Битовые комбинации для выбора банка регистров*

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

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.

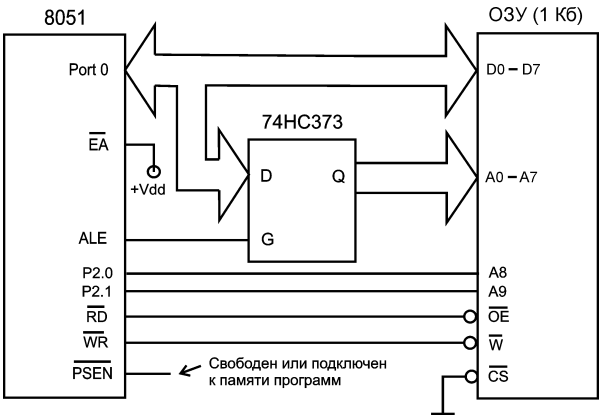
Нужно отметить одну важную особенность: при загрузке данных в стек адрес растет вверх, поэтому если в программе используются банки 1, 2 или 3, то указатель стека следует проинициализировать адресом из неиспользуемой области памяти, например 0x30, чтобы не перезаписать содержимое регистров одного из банков.

Двухбайтный регистр-указатель данных DPTR обычно используется для фиксации 16-разрядного адреса в операциях с обращением к внешней памяти. При работе с DPTR допускается 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.

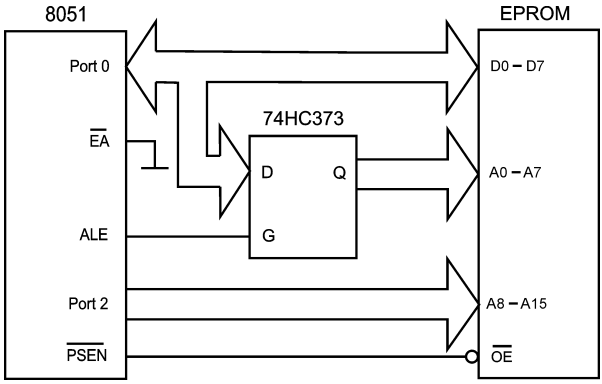


Рис. 1.7. *Подключение внешней памяти программ*

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 специфичных 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 add, A	(addr) ← (A)
MOV add, 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))

Таблица 1.4. *Команды пересылки данных (окончание)*

Мнемоника	Описание
MOVX @Ri, A	((Ri)) ← (A)
MOVX @DPTR, A	((DPTR)) ← (A)
PUSH addr	(SP) ← (SP) + 1, ((SP)) ← (addr)
POP addr	(addr) ← ((SP)), (SP) ← (SP) – 1
XCH A, Rn	(A) ↔ (Rn)
XCH A, addr	(A) ↔ (addr)
XCH A, @Ri	(A) ↔ ((@Ri))
XCHD A, @Ri	(A0 – 3) ↔ ((@Ri0 – 3))
MOV A, Rn	(A) ← (Rn)

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.

Таблица 1.5. *Команды арифметических операций*

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

Таблица 1.5. *Команды арифметических операций (окончание)*

Мнемоника	Описание
ADDC A, #direct8	(A) ← (A) + #direct8 + (C)
DAA	Десятичная коррекция аккумулятора
SUBB A, Rn	(A) ← (A) – (Rn) – (C)
SUBB A, addr	(A) ← (A) – (ad) – (C)
SUBB A, @Ri	(A) ← (A) – ((Ri)) – (C)
SUBB A, #direct8	(A) ← (A) – #direct8 – (C)
INC A	(A) ← (A) + 1INC Rn(Rn) < (Rn) + 1
INC Rn	(Rn) ← (Rn) + 1
INC addr	(addr) ← (addr) + 1
INC @Ri	((Ri)) ← ((Ri)) + 1
INC DPTR	(DPTR) ← (DPTR) + 1
DEC A	(A) ← (A) – 1
DEC Rn	(Rn) ← (Rn) – 1
DEC addr	(addr) ← (addr) – 1
DEC @Ri	((Ri)) ← ((Ri)) – 1
MUL AB	(B)(A) ← (A) × (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) ← (A) & (Rn)
ANL A, addr	(A) ← (A) & (addr)
ANL A, @Ri	(A) ← (A) & ((Ri))
ANL A, #direct8	(A) ← (A) & (#direct8)
ANL addr, A	(addr) ← (addr) & (A)
ANL addr, #direct8	(addr) ← (addr) & (#direct8)
ORL A, Rn	(A) ← (A) I (Rn)
ORL A, addr	(A) ← (A) I (addr)

Таблица 1.6. *Команды логических операций (окончание)*

Мнемоника	Описание
ORL A, @Ri	(A) ← (A) I ((Ri)
ORL A, #direct8	(A) ← (A) I (#direct8)
ORL addr, A	(addr) ← (addr) I (A)
ORL addr, #direct8	(addr) ← (addr) I (#direct8)
XRL A, Rn	(A) ← (A) ^ (Rn)
XRL A, addr	(A) ← (A) ^ (addr)
XRL A, @Ri	(A) ← (A) ^ ((Ri))
XRL A, #direct8	(A) ← (A) ^ (#direct8)
XRL addr, A	(addr) ← (addr) ^ (A)
XRL addr, #direct8	(addr) ← (addr) ^ (#direct8)
CLR A	(A) ← 0
CPL A	(A) ← ~(A)
SWAP A	(A0–3) – (A4–7)
RL A	Циклический сдвиг влево
RLC A	Сдвиг влево через перенос
RR A	Циклический сдвиг вправо
RRC A	Сдвиг вправо через перенос

Команды логических операций манипулируют байтами и позволяют выполнить следующие операции:

- логическое И (&);
- логическое ИЛИ (I);
- 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» (I). 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.

Таблица 1.7. *Команды битовых операций*

Мнемоника	Описание
CLR C	(C) ← 0
CLR bit	(bit) ← 0
SETB C	(C) ← 1
SETB bit	(bit) ← 1
CPL C	(C) ← ~(C)
CPL bit	(bit) ← ~ (bit)
ANL C, bit	(C) ← (C) & (bit)

Таблица 1.7. *Команды битовых операций (окончание)*

Мнемоника	Описание
ORL C, bit	(C) ← (C) ∨ (bit)
MOV C, bit	(C) ← (bit)
MOV bit, C	(bit) ← (C)

Последняя группа команд, которую мы рассмотрим, – это группа команд передачи управления микроконтроллера. В группе представлены команды безусловного и условного переходов, команды вызова подпрограмм и команды возврата из подпрограмм. Мнемоника команд и их описание представлены в табл. 1.8.

Таблица 1.8. *Команды передачи управления*

Мнемоника	Описание
LJMP addr16	Длинный безусловный переход по всем адресам памяти
AJMP addr11	Безусловный переход в пределах страницы 2 Кб
SJMP label	Безусловный переход на метку label в пределах страницы 256 байт
JMP @A+DPTR	Безусловный переход по косвенному адресу
JZ label	Переход на метку label, если нуль
JNZ label	Переход на метку label, если не нуль
JC label	Переход на метку label, если бит переноса установлен
JNC label	Переход на метку label, если бит переноса не установлен
JB bit, label	Переход на метку label, если бит установлен
JNB bit, label	Переход на метку label, если бит не установлен
JBC bit, label	Переход на метку label, если бит установлен с очисткой бита
DJNZ Rn, label	Переход на метку label, если содержимое Rn не равно
0DJNZ addr, label	Переход на метку label, если содержимое addr не равно
0CJNE A, addr, label	Сравнение аккумулятора с байтом и переход на метку label, если не равно
CJME A, #direct8, label	Сравнение аккумулятора с константой и переход на метку label, если не равно
CJNE Rn, #direct8, label	Сравнение регистра с константой и переход на метку label, если не равно
CJNE @Ri, #direct8, label	Сравнение байта памяти с константой и переход на метку label, если не равно
LCALL addr16	Длинный вызов подпрограммы по всем адресам памяти
ACALL addr11	Вызов подпрограммы в пределах страницы 2 Кб
RET	Возврат подпрограммы
RETI	Возврат подпрограммы обработки прерывания
NOP	Пустая операция

Рассмотрим, как работают команды передачи управления. Команда безусловного перехода LJMP осуществляет переход по абсолютному 16-битному адресу в пределах сегмента программ.

Действие команды AJMP аналогично команде LJMP, однако операндом являются лишь 11 младших разрядов адреса, что позволяет выполнить переход в пределах страницы размером 2 Кб. При этом содержимое счетчика команд вначале увеличивается на 2, затем заменяются 11 разрядов адреса.

В команде SJMP указан не абсолютный, а относительный адрес перехода. Величина смещения label рассматривается как число со знаком, поэтому переход возможен в пределах от –128 до +127 байт относительно адреса команды, следующей за командой SJMP.

Команда косвенного перехода JMP @A+DPTR позволяет вычислять адрес перехода в процессе выполнения самой программы. С помощью команд условного перехода можно проверять следующие условия:

- аккумулятор содержит нулевое значение (JZ);
- аккумулятор содержит ненулевое значение (JNZ);
- бит переноса C установлен (JC);
- бит переноса C не установлен (JNC);
- прямо адресуемый бит равен 1 (JB);
- прямо адресуемый бит равен 0 (JNB);
- прямо адресуемый бит равен 1 и очищается при выполнении команды (JBC).

Все команды условного перехода микроконтроллера 8051 оперируют с коротким относительным адресом из диапазона –128 – +127 относительно следующей команды.

Команда DJNZ предназначена для организации программных циклов. Указанные в команде регистр Rn или байт по адресу addr содержат счетчик повторений цикла, а смещение label – относительный адрес перехода к началу цикла. При выполнении команды содержимое счетчика уменьшается на 1 и проверяется на 0. Если значение содержимого счетчика не равно 0, то осуществляется переход на начало цикла, в противном случае выполняется следующая команда.

Команда CJNE удобна для реализации процедур ожидания событий. Операнды команды сравниваются между собой, после чего, в зависимости от результата сравнения, выполняется либо переход на метку label, либо выполняется следующая команда.

Действие команд вызова процедур полностью аналогично действию команд безусловного перехода – за исключением того, что они сохраняют в стеке адрес возврата.

Команда возврата из подпрограммы RET восстанавливает из стека значение содержимого счетчика команд, а команда возврата из процедуры обработки прерывания RETI, кроме того, разрешает прерывания. Следует отметить, что ассемблер 8051 допускает обобщенную мнемонику для команд безусловного перехода JMP и вызова подпрограмм CALL.

1.4. Система прерываний

Работа микроконтроллера 8051 в системах реального времени была бы невозможна без обработки событий, генерируемых внешними устройствами, и установки временных зависимостей между событиями в системе. Именно этим целям и служит логика обработки прерываний 8051, функциональная схема которой показана на рис. 1.8.

Классический микроконтроллер 8051 имеет 5 источников прерываний: два внешних прерывания, инициированных сигналами на входах, – INT0 (вывод P3.2) и INT1 (вывод P3.3); два прерывания таймеров – 0 и 1; прерывание последовательного порта (см. рис. 1.8). Очередность выполнения двух и более одновременно поступивших прерываний определяется их приоритетами.