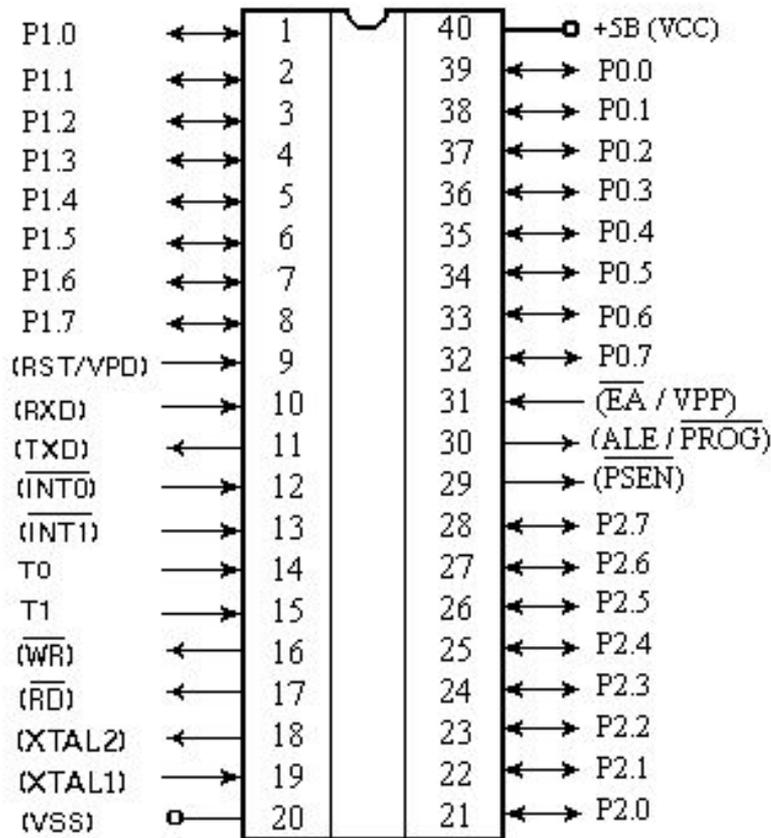




# Микропроцессорные СИСТЕМЫ

## АРХИТЕКТУРА МИКРОКОНТРОЛЛЕРА 8051

# Микроконтроллер 8051



Микроконтроллер выполнен на основе высокоуровневой n-МОП технологии. Через четыре программируемых параллельных порта ввода/вывода и один последовательный порт микроконтроллер взаимодействует с внешними устройствами. Основу структурной схемы (рис. 1) образует внутренняя двунаправленная 8-битная шина, которая связывает между собой основные узлы и устройства микроконтроллера: резидентную память программ (RPM), резидентную память данных (RDM), арифметико-логическое устройство (ALU), блок регистров специальных функций, устройство управления (CU) и порты ввода/вывода (P0-P3).

Микроконтроллеры рассматриваемого семейства являются типичными микропроцессорными устройствами с архитектурой SISC - со стандартным набором команд. Поэтому их система команд довольно обширна и включает в себя 111 основных команд. Их длина - один, два или три байта, причем большинство из них (94%) - одно- или двухбайтные. Все команды выполняются за один или два машинных цикла (соответственно 1 или 2 мкс при тактовой частоте 12 МГц), исключение - команды умножения и деления, которые выполняются за четыре машинных цикла (4 мкс). Микроконтроллеры семейства 8051 используют прямую, непосредственную, косвенную и неявную, адресацию данных.

В качестве операндов команд микроконтроллеры семейства 8051 могут использовать отдельные биты, четырехбитные цифры, байты и двухбайтные слова.

Все эти черты обычны для набора команд любого SISC-процессора и по сравнению с RISC набором команд обеспечивает большую компактность программного кода и увеличение быстродействия при выполнении сложных операций.

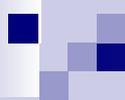
В то же время, набор команд семейства 8051 имеет несколько особенностей, связанных с типичными функциями выполняемыми микроконтроллерами - управлением, для которого типичным является оперирование с одноразрядными двоическими сигналами, большое число операций ввода вывода и ветвлений программы.

Наиболее существенная особенность системы команд рассматриваемых микроконтроллеров это возможность адресации отдельных бит в резидентной памяти данных. Кроме того, как отмечалось, некоторые регистры блока регистров специальных функций также допускают адресацию отдельных бит.

### **Арифметико-логическое устройство.**

8-битное арифметико-логическое устройство (ALU) может выполнять арифметические операции сложения, вычитания, умножения и деления; логические операции И, ИЛИ, исключающее ИЛИ, а также операции циклического сдвига, сброса, инвертирования и т.п. К входам подключены программно-недоступные регистры T1 и T2, предназначенные для временного хранения операндов, схема десятичной коррекции (DCU) и схема формирования признаков результата операции (PSW).

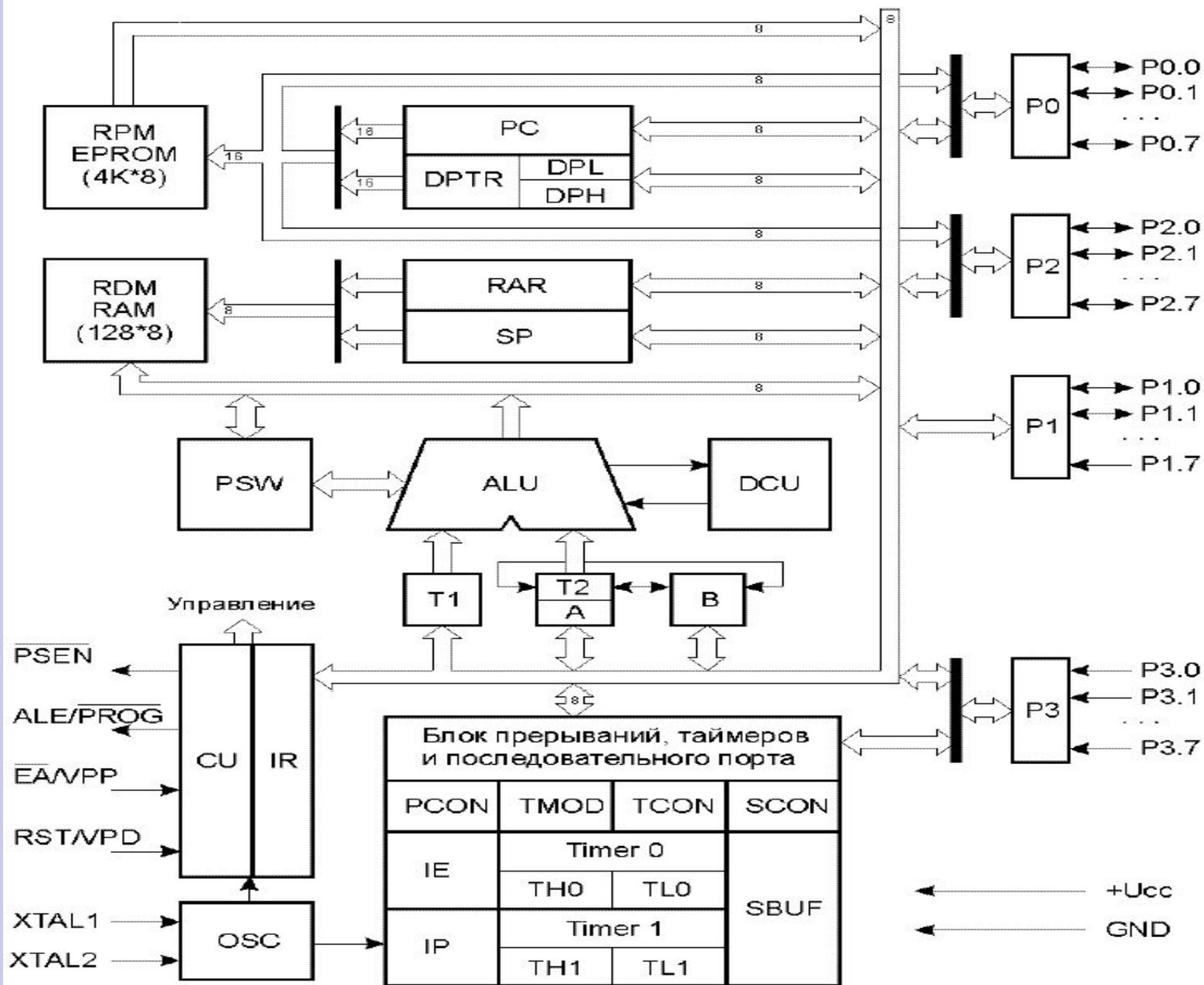
Простейшая операция сложения используется в ALU для инкрементирования содержимого регистров, продвижения регистра-указателя данных (RAR) и автоматического вычисления следующего адреса резидентной памяти программ. Простейшая операция вычитания используется в ALU для декрементирования регистров и сравнения переменных.



Простейшие операции автоматически образуют “тандемы” для выполнения таких операций, как, например, инкрементирование 16-битных регистровых пар. В ALU реализуется механизм каскадного выполнения простейших операций для реализации сложных команд. Так, например, при выполнении одной из команд условной передачи управления по результату сравнения в ALU трижды инкрементируется счётчик команд (PC), дважды производится чтение из RDM, выполняется арифметическое сравнение двух переменных, формируется 16-битный адрес перехода и принимается решение о том, делать или не делать переход по программе. Все перечисленные операции выполняются всего лишь за 2 мкс.

Важной особенностью ALU является его способность оперировать не только байтами, но и битами. Отдельные программно-доступные биты могут быть установлены, сброшены, инвертированы, переданы, проверены и использованы в логических операциях. Эта способность достаточно важна, поскольку для управления объектами часто применяются алгоритмы, содержащие операции над входными и выходными булевыми переменными, реализация которых средствами обычных микропроцессоров сопряжена с определенными трудностями.

Ниже приведена структурная схема микроконтроллера KM1816BE51.



Таким образом, ALU может оперировать четырьмя типами информационных объектов: булевыми (1 бит), цифровыми (4 бита), байтными (8 бит) и адресными (16 бит). В ALU выполняется 51 различная операция пересылки или преобразования этих данных. Так как используется 11 режимов адресации (7 для данных и 4 для адресов), то путем комбинирования операции и режима адресации базовое число команд 111 расширяется до 255 из 256 возможных при однобайтном коде операции.

### **Резидентная память программ и данных.**

Резидентные (размещённые на кристалле) память программ (RPM) и память данных (RDM) физически и логически разделены, имеют различные механизмы адресации, работают под управлением различных сигналов и выполняют разные функции

Память программ RPM имеет емкость 4 Кбайта и предназначена для хранения команд, констант, управляющих слов инициализации, таблиц перекодировки входных и выходных переменных и т.п. Память имеет 16-битную шину адреса, через которую обеспечивается доступ из программного счётчика РС или из регистра- указателя данных (DPTR).



DPTR выполняет функции базового регистра при косвенных переходах по программе или используется в операциях с таблицами. Память данных RDM предназначена для хранения переменных в процессе выполнения прикладной программы, адресуется одним байтом и имеет емкость 128 байт. Кроме того, к её адресному пространству примыкают адреса регистров специальных функций, которые перечислены в табл. 1.

Память программ, так же как и память данных, может быть расширена до 64 Кбайт путем подключения внешних микросхем.

Таблица 1. Блок регистров специальных функций

Символ	Наименование	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/счётчика	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 приёмопередатчиком	98H
SBUF	Буфер приёмопередатчика	99H
PCON	Регистр управления мощностью	87H

Примечание.  
Регистры, имена которых отмечены знаком (\*), допускают адресацию отдельных битов.

## **Аккумулятор, регистры общего назначения и флаги.**

Аккумулятор (A) является источником операнда и местом фиксации результата при выполнении арифметических, логических операций и ряда операций передачи данных. Кроме того, только с использованием аккумулятора могут быть выполнены операции сдвигов, проверка на нуль, формирование флага паритета и т.п. В распоряжении пользователя имеются 8 регистров общего назначения R0–R7 одного из четырёх возможных банков. При выполнении многих команд в ALU формируется ряд признаков операции (флагов), которые фиксируются в регистре PSW. В табл. 2 приводится перечень флагов PSW, даются их символические имена и описываются условия их формирования.

Таблица 2. Формат слова состояния программы - PSW.

Символ	Разряд	Имя и назначение
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. 3)
RS0	PSW.3	Выбор банка регистров. Устанавливается и сбрасывается программно для выбора рабочего банка регистров (табл. 3)
OV	PSW.2	Флаг переполнения. Устанавливается и сбрасывается аппаратно при выполнении арифметических операций
-	PSW.1	Не используется
P	PSW.0	Флаг паритета. Устанавливается и сбрасывается аппаратно в каждом цикле и фиксирует нечётное/чётное число единичных битов в аккумуляторе, т.е. выполняет контроль по четности

Таблица 3 Выбор рабочего 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

Наиболее “активным” флагом PSW является флаг переноса, который принимает участие и модифицируется в процессе выполнения множества операций, включая сложение, вычитание и сдвиги. Кроме того, флаг переноса (C) выполняет функции “булева аккумулятора” в командах, манипулирующих с битами. Флаг переполнения (OV) фиксирует арифметическое переполнение при операциях над целыми числами со знаком и делает возможным использование арифметики в дополнительных кодах. ALU не управляет флагами селекции банка регистров (RS0, RS1), их значение полностью определяется прикладной программой и используется для выбора одного из четырёх регистровых банков.

В микропроцессорах, архитектура которых опирается на аккумулятор, большинство команд работают с ним, используя неявную адресацию. В Intel 8051 дело обстоит иначе. Хотя процессор имеет в своей основе аккумулятор, он может выполнять множество команд и без его участия. Например, данные могут быть переданы из любой ячейки RDM в любой регистр, любой регистр может быть загружен непосредственным операндом и т.д. Многие логические операции могут быть выполнены без участия аккумулятора. Кроме того, переменные могут быть инкрементированы, декрементированы и проверены без использования аккумулятора. Флаги и управляющие биты могут быть проверены и изменены аналогично.

### **Регистры-указатели.**

8-битный указатель стека (SP) может адресовать любую область RDM. Его содержимое инкрементируется прежде, чем данные будут запомнены в стеке в ходе выполнения команд PUSH и CALL. Содержимое SP декрементируется после выполнения команд POP и RET. Подобный способ адресации элементов стека называют преинкрементным/постдекрементным. В процессе инициализации микроконтроллера после сигнала RST в SP автоматически загружается код 07H. Это значит, что если прикладная программа не переопределяет стек, то первый элемент данных в стеке будет располагаться в ячейке RDM с адресом 08H. Двухбайтный регистр-указатель данных DPTR обычно используется для фиксации 16-битного адреса в операциях с обращением к внешней памяти. Командами микроконтроллера регистр-указатель данных может быть использован или как 16-битный регистр, или как два независимых 8-битных регистра (DPH и DPL).

## **Регистры специальных функций.**

Регистры с символическими именами IP, IE, TMOD, TCON, SCON и PCON используются для фиксации и программного изменения управляющих бит и бит состояния схемы прерывания, таймера/счётчика, приёмопередатчика последовательного порта и для управления энергопотреблением. Их организация будет описана ниже при рассмотрении особенностей работы микроконтроллера в различных режимах.

## **Устройство управления и синхронизации**

Кварцевый резонатор, подключаемый к внешним выводам микроконтроллера, управляет работой внутреннего генератора, который в свою очередь формирует сигналы синхронизации. Устройство управления (CU) на основе сигналов синхронизации формирует машинный цикл фиксированной длительности, равной 12 периодам резонатора. Большинство команд микроконтроллера выполняется за один машинный цикл. Некоторые команды, оперирующие с 2-байтными словами или связанные с обращением к внешней памяти, выполняются за два машинных цикла. Только команды деления и умножения требуют четырех машинных циклов. На основе этих особенностей работы устройства управления производится расчёт времени исполнения прикладных программ.

На схеме микроконтроллера к устройству управления примыкает регистр команд (IR). В его функцию входит хранение кода выполняемой команды.

### **Входные и выходные сигналы устройства управления и синхронизации:**

- PSEN – разрешение программной памяти,
- ALE – выходной сигнал разрешения фиксации адреса,
- PROG – сигнал программирования,
- EA – блокировка работы с внутренней памятью,
- VPP – напряжение программирования,
- RST – сигнал общего сброса,
- VPD – вывод резервного питания памяти от внешнего источника,
- XTAL – входы подключения кварцевого резонатора.

### **Параллельные порты ввода/вывода информации.**

Все четыре порта (P0-P3) предназначены для ввода или вывода информации побайтно. Каждый порт содержит управляемые регистр-защёлку, входной буфер и выходной драйвер.

Выходные драйверы портов 0 и 2, а также входной буфер порта 0 используются при обращении к внешней памяти. При этом через порт 0 в режиме временного мультиплексирования сначала выводится младший байт адреса, а затем выдается или принимается байт данных. Через порт 2 выводится старший байт адреса в тех случаях, когда разрядность адреса равна 16 бит. Все выводы порта 3 могут быть использованы для реализации альтернативных функций, перечисленных в табл. 4. Эти функции могут быть задействованы путем записи 1 в соответствующие биты регистра-защёлки (P3.0-P3.7) порта 3.

Таблица 4. Альтернативные функции порта P3.

Символ	Разряд	Имя и назначение
RD	P3.7	Чтение. Активный сигнал низкого уровня формируется аппаратно при обращении к внешней памяти данных
WR	P3.6	Запись. Активный сигнал низкого уровня формируется аппаратно при обращении к внешней памяти данных
T1	P3.5	Вход таймера/счётчика 1 или тест-вход
T0	P3.4	Вход таймера/счётчика 0 или тест-вход
INT1	P3.3	Вход запроса прерывания 1. Воспринимается сигнал низкого уровня или срез
INT0	P3.2	Вход запроса прерывания 0. Воспринимается сигнал низкого уровня или срез
TXD	P3.1	Выход передатчика последовательного порта в режиме UART. Выход синхронизации в режиме регистра сдвига
RXD	P3.0	Вход приёмника последовательного порта в режиме UART. Ввод/вывод данных в режиме регистра сдвига

Порт 0 является двунаправленным, а порты 1-3 - квазидвунаправленными. Каждая линия портов может быть использована независимо для ввода или вывода. По сигналу RST в регистры-защёлки всех портов автоматически записываются единицы, настраивающие их тем самым на режим ввода. Все порты могут быть использованы для организации ввода/вывода информации по двунаправленным линиям передачи. Однако порты 0 и 2 не могут быть использованы для этой цели в случае, если система имеет внешнюю память, связь с которой организуется через общую разделяемую шину адреса/данных, работающую в режиме временного мультиплексирования. Обращение к портам ввода/вывода возможно с использованием команд, оперирующих с байтом, отдельным битом, произвольной комбинацией битов. При этом в тех случаях, когда порт является одновременно операндом и местом назначения результата, устройство управления автоматически реализует специальный режим, который называется "чтение-модификация-запись". Этот режим обращения предполагает ввод сигналов не с внешних выводов порта, а из его регистразащёлки, что позволяет исключить неправильное считывание ранее выведенной информации. Этот механизм обращения к портам реализован в командах:

- ANL – логическое И, например, ANL P1,A;
- ORL – логическое ИЛИ, например, ORL P2,A;
- XRL – исключающее ИЛИ, например, XRL P3,A;
- JBC – переход, если в адресуемом бите единица, и последующий сброс бита, например, JBC P1.1, LABEL;
- CPL – инверсия бита, например, CPL P3.3;
- INC – инкремент порта, например, INC P2;
- DEC – декремент порта, например, DEC P2;
- DJNZ – декремент порта и переход, если его содержимое не равно нулю, например, DJNZ r, LABEL;
- MOV PX.Y,C – передача бита переноса в бит Y порта X;
- SET PX.Y – установка бита Y порта X; • CLR PX.Y – сброс бита Y порта X.

## Таймер/счётчик.

В составе микроконтроллера имеются регистровые пары с символическими именами TH0, TL0 и TH1, TL1, на основе которых функционируют два независимых программно-управляемых 16-битных таймера/счётчика событий (T/C0 и T/C1). При работе в качестве таймера содержимое T/C инкрементируется в каждом машинном цикле, то есть через каждые 12 периодов резонатора. При работе в качестве счётчика содержимое T/C инкрементируется под воздействием перехода из 1 в 0 внешнего входного сигнала, подаваемого на соответствующий (T0, T1) вход микроконтроллера. Опрос сигналов выполняется в каждом машинном цикле. Так как на распознавание перехода требуется два машинных цикла, то максимальная частота подсчёта входных сигналов равна  $1/24$  частоты резонатора. На длительность периода входных сигналов ограничений сверху нет. Для гарантированного прочтения входного считываемого сигнала он должен удерживать значение 1 как минимум в течение одного машинного цикла.

Для управления режимами работы и для организации взаимодействия таймеров с системой прерывания используются два регистра специальных функций TMOD и TCON, описание которых приводится в табл. 5-7. Для обоих T/C режимы работы 0, 1 и 2 одинаковы. Режимы 3 для T/C0 и T/C1 различны.

Таблица 5. Регистр режима работы таймера/счётчика.

Символ	Разряд	Имя и назначение
GATE	TMOD.7 для T/C1	Управление блокировкой. Если бит установлен, на входе "INTx" высокий уровень и бит управления "TRx" установлен. Если бит сброшен, то T/C разрешается, как только бит управления "TRx" устанавливается
	TMOD.3 для T/C0	Управление блокировкой. Если бит установлен, на входе "INTx" высокий уровень и бит управления "TRx" установлен. Если бит сброшен, то T/C разрешается, как только бит управления "TRx" устанавливается
C/T	TMOD.6 для T/C1	Бит выбора режима таймера или счётчика событий. Если бит сброшен, то работает таймер от внутреннего источника сигналов синхронизации. Если бит установлен, то работает счётчик от внешних сигналов на входе "Tx"
	TMOD.2 для T/C0	Бит выбора режима таймера или счётчика событий. Если бит сброшен, то работает таймер от внутреннего источника сигналов синхронизации. Если бит установлен, то работает счётчик от внешних сигналов на входе "Tx"
M1	TMOD.5 для T/C1	Режим работы (см. табл. 6)
	TMOD.1 для T/C0	Режим работы (см. табл. 6)
M0	TMOD.4 для T/C1	Режим работы (см. табл. 6)
	TMOD.0 для T/C0	Режим работы (см. табл. 6)

Таблица 6. Режимы работы таймера/счётчика.

M1	M0	Режим работы
0	0	“TLx” работает как 5-битный предделитель
0	1	16-битный таймер/счётчик. “ТНх” и “TLx” включены последовательно
1	0	8-битный автоперезагружаемый таймер/счётчик. “ТНх” хранит значение, которое должно быть перезагружено в “TLx” каждый раз по переполнению
1	1	Таймер/счётчик 1 останавливается. Таймер/счётчик 0: TL0 работает как 8-битный таймер/счётчик, и его режим определяется управляющими битами таймера 0. ТН0 работает только как 8-битный таймер, и его режим определяется управляющими битами таймера 1

Таблица 7. Регистр управления/статуса таймера.

Символ	Разряд	Имя и назначение
TF1	TCON.7	Флаг переполнения таймера 1. Устанавливается аппаратно при переполнении таймера/счётчика. Сбрасывается при обслуживании прерывания аппаратно
TR1	TCON.6	Бит управления таймера 1. Устанавливается/сбрасывается программой для пуска/останова
TF0	TCON.5	Флаг переполнения таймера 0. Устанавливается аппаратно. Сбрасывается при обслуживании прерывания
TR0	TCON.4	Бит управления таймера 0. Устанавливается/сбрасывается программой для пуска/останова таймера/счётчика
IE1	TCON.3	Флаг фронта прерывания 1. Устанавливается аппаратно, когда детектируется срез внешнего сигнала INT1. Сбрасывается при обслуживании прерывания
IT1	TCON.2	Бит управления типом прерывания 1. Устанавливается/сбрасывается программно для спецификации запроса INT1 (срез/низкий уровень)
IE0	TCON.1	Флаг фронта прерывания 0. Устанавливается по срезу сигнала INT0. Сбрасывается при обслуживании прерывания
IT0	TCON.0	Бит управления типом прерывания 0. Устанавливается/сбрасывается программно для спецификации запроса INT0 (срез/низкий уровень)

Режим 0. Перевод любого T/C в этот режим делает его 8-разрядным таймером, на вход которого подключен 5-битный делитель частоты на 32. В этом режиме таймерный регистр имеет разрядность 13 бит. При переходе из состояния “все единицы” в состояние “все нули” устанавливается флаг прерывания от таймера TF1. Входной синхросигнал таймера 1 разрешен (поступает на вход T/C), когда управляющий бит TR1 установлен в 1 и либо управляющий бит GATE (блокировка) равен 0, либо на внешний вход запроса прерывания INT1 поступает уровень 1. Установка бита GATE в 1 позволяет использовать таймер для измерения длительности импульсного сигнала, подаваемого на вход запроса прерывания.

Режим 1. Работа любого T/C в этом режиме такая же, как и в режиме 0, за исключением того, что таймерный регистр имеет разрядность 16 бит.

Режим 2. В этом режиме работа организована таким образом, что переполнение (переход из состояния “все единицы” в состояние “все нули”) 8-битного счётчика TL1 приводит не только к установке флага TF1, но и автоматически перезагружает в TL1 содержимое старшего байта (TH1) таймерного регистра, которое предварительно было задано программным путем. Перезагрузка оставляет содержимое TH1 неизменным. В режиме 2 T/C0 и T/C1 работают совершенно одинаково.

Режим 3. В этом режиме T/C0 и T/C1 работают по-разному. T/C1 сохраняет неизменным своё текущее содержимое. Иными словами, эффект такой же, как и при сбросе управляющего бита TR1 в нуль. В этом режиме TL0 и TH0 функционируют как два независимых 8-битных счётчика. Работу TL0 определяют управляющие биты T/C0 (C/T, GATE, TR0), входной сигнал INT0 и флаг переполнения TF0. Работу TH0, который может выполнять только функции таймера (подсчёт машинных циклов микроконтроллера), определяет управляющий бит TR1.

При этом ТН0 использует флаг переполнения TF1. Режим 3 используется в тех случаях, когда требуется наличие дополнительного 8-битного таймера или счётчика событий. Можно считать, что в режиме 3 микроконтроллер имеет в своем составе три таймера/счётчика. В том случае, если T/C0 используется в режиме 3, T/C1 может быть или включен, или выключен, или переведен в свой собственный режим 3, или может быть использован последовательным портом в качестве генератора частоты передачи, или, наконец, может быть использован в любом применении, не требующем прерывания.

### Последовательный порт

Через универсальный асинхронный приёмопередатчик UART (Universal Asynchronous Receiver-Transmitter) происходит передача информации, представленной последовательным кодом (младшими битами вперед), в полном дуплексном режиме обмена. В состав UART, называемого часто последовательным портом, входят принимающий и передающий сдвигающие регистры, а также специальный буферный регистр (SBUF) приёмопередатчика.

### Регистр SBUF

Представляет собой два независимых регистра: буфер приёмника и буфер передатчика. Загрузка байта в SBUF немедленно вызывает начало процесса передачи через последовательный порт. Когда байт считывается из SBUF, это значит, что его источником является приёмник последовательного порта. Запись байта в буфер приводит к автоматической переписи байта в сдвигающий регистр передатчика и инициирует начало передачи байта. Наличие буферного регистра приёмника позволяет совмещать операцию чтения ранее принятого байта с приёмом очередного байта. Если к моменту окончания приёма байта предыдущий байт не был считан, то он будет потерян.

Последовательный порт может работать в четырех различных режимах. Режим 0. Информация передается и принимается через вход приёмника RXD. Принимаются и передаются 8 бит данных. Через внешний выход передатчика TXD выдаются импульсы сдвига, которые сопровождают каждый бит. Частота передачи равна  $1/12$  частоты резонатора. Режим 1. Через TXD передаются или из RXD принимаются 10 бит: старт-бит (0), 8 бит данных и стоп-бит (1). Скорость приёма/передачи – величина переменная и задаётся таймером. Режим 2. Через TXD передаются или из RXD принимаются 11 бит: старт-бит, 8 бит данных, программируемый девятый бит и стоп-бит. При передаче девятый бит может использоваться для повышения достоверности передачи путём контроля по чётности и в него можно поместить значение признака паритета из PSW. Частота приёма/передачи выбирается программно и может быть равна  $1/32$  или  $1/64$  частоты резонатора в зависимости от SMOD. Режим 3. Совпадает с режимом 2, но частота приёма/передачи является величиной переменной и задаётся таймером.

3.9.2. Регистр SCON Регистр предназначен для управления режимом работы UART. Регистр содержит управляющие биты и девятый бит принимаемых или передаваемых данных RB8 и TB8, а также биты прерывания приёмопередатчика RI и TI. Функциональное назначение битов указано в табл. 8 и 9.

Таблица 8. Регистр управления/статуса UART.

Символ	Разряд	Имя и назначение
SM0	SCON.7	Биты управления режимом работы UART. Устанавливаются/сбрасываются программно (табл. 9)
SM1	SCON.6	Биты управления режимом работы UART. Устанавливаются/сбрасываются программно (табл. 9)
SM2	SCON.5	Бит управления режимом UART. Устанавливается программно для запрета приёма сообщения, в котором девятый бит равен 0
REN	SCON.4	Бит разрешения приёма. Устанавливается/сбрасывается программно для разрешения/запрета приёма последовательных данных
TB8	SCON.3	Передача бита 8. Устанавливается/сбрасывается программно для задания девятого передаваемого бита в режиме UART - 9 бит
RB8	SCON.2	Приём бита 8. Устанавливается/сбрасывается аппаратно для фиксации девятого принимаемого бита в режиме UART - 9 бит
TI	SCON.1	Флаг прерывания передатчика. Устанавливается аппаратно при окончании передачи байта. Сбрасывается программно после обслуживания прерывания
RI	SCON.0	Флаг прерывания приёмника. Устанавливается аппаратно при приёме байта. Сбрасывается программно после обслуживания прерывания

Таблица 9. Режим работы UART.

SM0	SM1	Режим работы UART
0	0	Сдвигающий регистр расширения ввода/вывода
0	1	UART - 8 бит. Изменяемая скорость передачи
1	0	UART - 9 бит. Фиксированная скорость передачи
1	1	UART - 9 бит. Изменяемая скорость передачи

Прикладная программа путём загрузки в два старших разряда SCON определяет режим работы UART. Во всех режимах передача инициируется любой командой, где SBUF указан как получатель байта. Приём в UART в режиме 0 происходит при условии RI=0 и REN=1. В режимах 1-3 приём начинается с приходом стартового бита, если REN=1. В TB8 программно устанавливается значение девятого бита данных, который будет передан в режиме 2 или 3. В RB8 фиксируется в режимах 2 и 3 девятый принимаемый бит данных. В режиме 1, если SM2=0, в бит RB8 заносится стоп-бит. В режиме 0 RB8 не используется. Флаг прерывания передатчика TI устанавливается аппаратно в конце периода передачи восьмого бита данных в режиме 0 и в начале периода передачи стоп-бита в режимах 1-3. Подпрограмма обслуживания этого прерывания должна сбрасывать бит TI. Флаг прерывания приёмника RI устанавливается аппаратно в конце периода приёма восьмого бита данных в режиме 0 и в середине периода приёма стоп-бита в режимах 1-3. Подпрограмма обслуживания прерывания должна сбрасывать бит RI.

## Работа UART в мультиконтроллерных системах.

В системах децентрализованного управления, которые используются для управления и регулирования в топологически распределенных объектах, возникает задача обмена информацией между множеством микроконтроллеров, объединенных в локальную вычислительно-управляющую сеть. Как правило, локальные сети на основе Intel 8051 имеют магистральную архитектуру с разделяемым моноканалом (коаксиальный кабель, витая пара, оптическое волокно), по которому осуществляется обмен информацией между контроллерами. Бит SM2 в SCON позволяет простыми средствами реализовать межконтроллерный обмен. Механизм обмена построен на том, что в режимах 2 и 3 программируемый девятый бит данных при приеме фиксируется в бите RB8. UART может быть запрограммирован таким образом, что при получении стоп-бита прерывание от приёмника будет возможно только при условии RB8=1. Ведущий контроллер всем ведомым передаёт широковещательное сообщение с байтом-идентификатором абонента, которое отличается от байтов данных только тем, что в его девятом бите содержится 1. Ведомые по этому признаку вызывают подпрограммы сравнения байта-идентификатора с кодом собственного сетевого адреса. Адресуемый контроллер сбрасывает свой SM2 и готовится к приёму блока данных. Остальные ведомые микроконтроллеры оставляют неизменными свои SM2=1 и передают управление основной программе. При SM2=1 информационные байты в сети прерывания не вызывают. В режиме 1 автономного микроконтроллера SM2 используется для контроля истинности стоп-бита. В режиме 0 SM2 не используется и должен быть сброшен.

## Скорость приёма/передачи.

Скорость зависит от режима работы UART. В режиме 0 частота зависит только от резонатора:  $f_0 = f_{рез}/12$ . За один машинный цикл передаётся один бит. В режимах 1-3 скорость зависит от значения управляющего бита SMOD в регистре специальных функций PCON (табл. 10). В режиме 2 частота передачи  $f_2 = (2SMOD/64)f_{рез}$ . В режимах 1 и 3 в формировании частоты передачи кроме управляющего бита SMOD принимает участие таймер 1. При этом частота передачи зависит от частоты переполнения (OVT1) и определяется следующим образом:  $f_{1,3} = (2SMOD/32)f_{OVT1}$ . Прерывание от таймера 1 в этом случае должно быть заблокировано. Сам T/C1 может работать и как таймер, и как счётчик событий в любом из трёх режимов. Однако наиболее удобно использовать режим таймера с автоперезагрузкой (старшая тетрада TMOD=0010B). При этом частота передачи определяется выражением  $f_{1,3} = (2SMOD/32)(f_{рез}/12)(256 - (TH1))$ . В табл. 11 приводится описание способов настройки T/C1 для получения типовых частот передачи данных через UART.

Таблица 10. Регистр управления мощностью PCON.

Символ	Разряд	Наименование и функция
SMOD	PCON.7	Удвоенная скорость передачи. Если бит установлен в 1, то скорость передачи вдвое больше, чем при SMOD=0
-	PCON.6-4	Не используются
GF1	PCON.3	Флаги, специфицируемые пользователем (флаги общего назначения)
GF0	PCON.2	Флаги, специфицируемые пользователем (флаги общего назначения)
PD	PCON.1	Бит пониженной мощности. При установке в 1 микроконтроллер переходит в режим пониженного энергопотребления
IDL	PCON.0	Бит холостого хода. Если бит установлен в 1, то микроконтроллер переходит в режим холостого хода

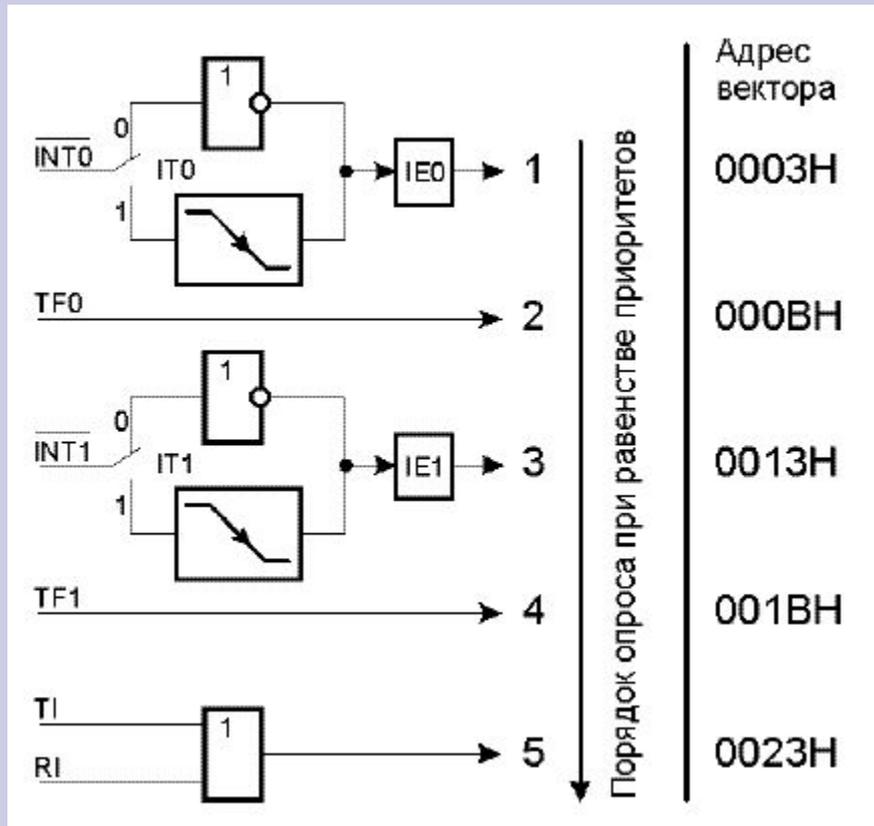
Примечание. При одновременной записи 1 в PD и IDL бит PD имеет преимущество. Сброс PCON выполняется путем загрузки в него кода 0XXX0000.

Таблица 11. Настройка таймера 1 для управления частотой работы UART.

Таймер/счётчик 1

Частота приёма/передачи (BAUD RATE)	Частота резонатора, МГц	SMOD Таймер/счётчик 1	С/Т Таймер/счётчик 1	Режим (MODE) Таймер/счётчик 1	Перезагружаемое число
Режим 0, макс.: 1 МГц	12	X	X	X	X
Режим 2, макс.: 375 кГц	12	1	X	X	X
Режимы 1,3: 62,5 кГц	12	1	0	2	0FFH
19,2 кГц	11,059	1	0	2	0FDH
9,6 кГц	11,059	0	0	2	0FDH
4,8 кГц	11,059	0	0	2	0FAH
2,4 кГц	11,059	0	0	2	0F4H
1,2 кГц	11,059	0	0	2	0E8H
137,5 Гц	11,059	0	0	2	1DH
110 Гц	6	0	0	2	72H
110 Гц	12	0	0	1	0FEEBH

## Система прерываний.



Внешние прерывания INT0 и INT1 (рис. 2) могут быть вызваны уровнем или переходом сигнала из 1 в 0 на входах микроконтроллера в зависимости от значений управляющих битов IT0 и IT1 в регистре TCON. От внешних прерываний устанавливаются флаги IE0 и IE1 в регистре TCON, которые инициируют вызов соответствующей подпрограммы обслуживания прерывания. Сброс этих флагов выполняется аппаратно только в том случае, если прерывание было вызвано по переходу (срезу) сигнала. Если же прерывание вызвано уровнем входного сигнала, то сбросом флага IE управляет соответствующая подпрограмма обслуживания прерывания путем воздействия на источник прерывания с целью снятия им запроса.

Рис. 2. Схема прерываний.

Флаги запросов прерывания от таймеров TF0 и TF1 сбрасываются автоматически при передаче управления подпрограмме обслуживания. Флаги запросов прерывания RI и TI устанавливаются UART аппаратно, но сбрасываться должны программой. Прерывания могут быть вызваны или отменены программой, так как все перечисленные флаги программно доступны. В блоке регистров специальных функций есть два регистра, предназначенных для управления режимом прерываний и уровнями приоритета. Форматы этих регистров, имеющих символические имена IE и IP описаны в табл. 12 и 13 соответственно.

Таблица 12. Регистр масок прерывания IE.

Символ	Разряд	Имя и назначение
EA	IE.7	Снятие блокировки прерываний. Сбрасывается программно для запрета всех прерываний независимо от состояний IE4-IE0
-	IE.6, 5	Не используются
ES	IE.4	Бит разрешения прерывания от UART. Установка/сброс программой для разрешения/запрета прерываний от флагов TI, RI
ET1	IE.3	Бит разрешения прерывания от таймера 1. Установка/сброс программой для разрешения/запрета прерываний от таймера 1
EX1	IE.2	Бит разрешения внешнего прерывания 1. Установка/сброс программой для разрешения/запрета прерываний
ET0	IE.1	Разрешение прерывания от таймера 0. Работает аналогично IE.3
EX0	IE.0	Разрешения внешнего прерывания 0. Работает аналогично IE.2

Таблица 13. Регистр приоритетов прерывания IP.

Символ	Разряд	Имя и назначение
-	IP.7-5	Не используются
PS	IP.4	Бит приоритета UART. Установка/сброс программой для назначения прерыванию от UART высшего/низшего приоритета
PT1	IP.3	Бит приоритета таймера 1. Установка/сброс программой для назначения прерыванию от таймера 1 высшего/низшего приоритета
PX1	IP.2	Бит приоритета внешнего прерывания 1. Установка/сброс программой для назначения прерыванию INT1 высшего/низшего приоритета
PT0	IP.1	Бит приоритета таймера 0. Работает аналогично IP.3
PX0	IP.0	Приоритет внешнего прерывания 0. Работает аналогично IP.2

Возможность программной установки/сброса любого управляющего бита в этих двух регистрах делает систему прерываний исключительно гибкой. Флаги прерываний опрашиваются в каждом машинном цикле. Ранжирование прерываний по приоритету выполняется в течение следующего машинного цикла. Система прерываний сформирует аппаратно вызов LCALL соответствующей подпрограммы обслуживания, если она не заблокирована одним из условий:

- в данный момент обслуживается запрос прерывания равного или более высокого уровня приоритета;
- текущий машинный цикл – не последний в цикле выполняемой команды;
- выполняется команда RETI или любая команда, связанная с обращением к регистрам IE или IP.

Примечание. Если флаг прерывания был установлен, но по одному из перечисленных условий не получил обслуживания и к моменту окончания блокировки уже был сброшен, то запрос прерывания теряется. По аппаратно сформированному коду команды LCALL система прерывания помещает в стек содержимое программного счётчика PC и загружает в PC адрес вектора прерывания соответствующей подпрограммы обслуживания. По этому адресу должна быть расположена команда безусловного перехода JMP к начальному адресу подпрограммы обслуживания прерывания. Эта подпрограмма в случае необходимости должна начинаться командами записи в стек PUSH слова состояния программы PSW, аккумулятора A, расширителя аккумулятора B, указателя данных DPTR и т.д. и заканчиваться командами восстановления из стека POP. Подпрограммы обслуживания прерывания обязательно завершаются командой RETI, по которой в программный счётчик перезагружается из стека сохранённый адрес возврата в основную программу. Команда RET также возвращает управление, но при этом не снимает блокировку прерывания.

## Система команд, особенности набора команд

### Типы команд.

Всего микроконтроллеры семейства 8051 выполняют 13 типов команд, они приведены в таблице. Как следует из нее, первый байт команды всегда содержит код операции (КОП), а второй и третий (если они присутствуют в команде) - адреса операндов или их непосредственные значения.

Тип	Первый байт	Второй байт	Третий байт
команды	D7...D0	D7...D0	D7...D0
тип 1	коп		
тип 2	коп	#d	
тип 3	коп	ad	
тип 4	коп	bit	
тип 5	коп	rel	
тип 6	коп	a7...a0	
тип 7	коп	ad	#d
тип 8	коп	ad	rel
тип 9	коп	ads	add
тип 10	коп	#d	rel
тип 11	коп	bit	rel
тип 12	коп	ad16h	ad16l
тип 13	коп	#d16h	#d16l

## Группы команд.

Все команды микроконтроллеры семейства 8051 можно разбить на пять функциональных групп:

1. пересылки данных;
2. арифметических операций;
3. логических операций;
4. операций над битами;
5. передачи управления.

## Обозначения, используемые при описании команд:

$R_n$  ( $n = 0, 1, \dots, 7$ ) - регистр общего назначения в выбранном банке регистров;

$@R_i$  ( $i = 0, 1$ ) - регистр общего назначения в выбранном банке регистров, используемый в качестве регистра косвенного адреса;

ad - адрес прямоадресуемого байта;

ads - адрес прямо адресуемого байта-источника;

add - адрес прямо адресуемого байта-получателя;

ad11 - 11-разрядный абсолютный адрес перехода;

ad16 - 16-разрядный абсолютный адрес перехода;

rel - относительный адрес перехода;

#d - непосредственный операнд;

#d16 - непосредственный операнд (2 байта);

bit - адрес прямо адресуемого бита;

/bit - инверсия прямо адресуемого бита;

A - аккумулятор;

PC - счетчик команд;

DPTR - регистр указатель данных;

( ) - содержимое ячейки памяти или регистра,

## Команды пересылки данных микроконтроллера 8051

Эта группа представлена 28 командами, их краткое описание приведено в таблице, где также указаны тип команды (Т) в соответствии с таблицей, ее длина в байтах (В) и время выполнения в машинных циклах (С).

Мнемокод	КОП	Т В С	Описание
MOV A, Rn	11101rrr	1 1 1	(A) <-- (Rn)
MOV A, ad	11100101	3 2 1	(A) <--(ad)
MOV A, @Ri	1110011i	1 1 1	(A) <-- ((Ri))
MOV A, #d	1110100	2 2 1	(A) <-- #d
MOV Rn, A	11111rrr	1 1 1	(Rn) <-- (A)
MOV Rn, ad	10101rrr	3 2 2	(Rn) <-- (ad)
MOV Rn, #d	01111rrr	2 2 1	(Rn) <-- #d
MOV ad, A	11110101	3 2 1	(ad) <--(A)
MOV ad, Rn	10001rrr	3 2 2	(ad) <-- (Rn)
MOV add, ads	10000101	9 3 2	(add) <-- (ads)
MOV ad, @Ri	1000011i	3 2 2	(ad) <-- ((Ri))
MOV ad, #d	1110101	7 3 2	(ad) <-- #d
MOV @Ri, A	1111011i	1 1 1	((Ri)) <-- (A)
MOV @Ri, ad	0110011i	3 2 2	((Ri)) <-- (ad)
MOV @Ri, #d	0111011i	2 2 1	((Ri)) <-- #d
MOV DPTR, #d16	10010000	3 3 2	(DPTR) <-- #d16
MOVC A, @A+DPTR	10010011	1 1 2	(A) <-- ((A)+(DPTR))
MOVC A, @A+pc	10000011	4 1 2	(PC) <-- (PC+1), (A) <-- ((A)+(PC))
MOVX A,@Ri	11100011	1 1 2	(A) <-- ((Ri))
MOVX a, @DPTR	11100000	1 1 2	(A) <-- ((DPTR))
MOVX @Ri, A	1111001i	1 1 2	((Ri)) <-- (A)
MOVX @DPTR, A	11110000	1 1 2	(DPTR) <-- (A)
PUSH ad	11000000	3 2 2	(SP) <-- (SP)+1, ((SP)) <-- (ad)
POP ad	11010000	3 2 2	(ad) <-- ((SP)), (SP) <-- (SP)-1
XCHA, Rn	11001rrr	1 1 1	(A) <-> (Rn)
XCHA, ad	11000101	3 2 1	(A) <->(ad)
XCHA, @Ri	11000111	1 1 1	(A) <->((@Ri))
A, @Ri	11010111	1 1 1	(A0-3)<->((@Ri0-3))

По команде MOV выполняется пересылка данных из второго операнда в первый. Эта команда не имеет доступа ни к внешней памяти данных, ни к памяти программ. Для этих целей предназначены команды MOVX и MOVC соответственно. Первая из них обеспечивает чтение/запись байт из внешней памяти данных, вторая - чтение байт из памяти программ.

По команде XCH выполняется обмен байтами между аккумулятором и ячейкой РПД, а по команде XCHD - обмен младшими тетрадами (битами 0 - 3).

Команды PUSH и POP предназначены соответственно для записи данных в стек и их чтения из стека. Размер стека ограничен лишь размером резидентной памяти данных. В процессе инициализации микроконтроллеры после сигнала сброса или при включении питающего напряжения в SP заносится код 07H. Это означает, что первый элемент стека будет располагаться в ячейке памяти с адресом 08H.

Группа команд пересылок микроконтроллера имеет следующую особенность - в ней нет специальных команд для работы со специальными регистрами: PSW, таймером, портами ввода-вывода. Доступ к ним, как и к другим регистрам специальных функций, осуществляется заданием соответствующего прямого адреса, т.е. это команды обычных пересылок, в которых вместо адреса можно ставить название соответствующего регистра. Например, чтение PSW в аккумулятор может быть выполнено командой:

`MOV A, PSW` , которая преобразуется Ассемблером к виду:

`MOV A, 0D0h (E5 D0)` , где E5 - код операции, а D0 - операнд (адрес PSW).

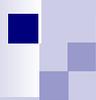
Кроме того, следует отметить, что в микроконтроллере аккумулятор имеет два различных имени в зависимости от способа адресации: A - при неявной адресации (например, `MOV A, R0`) и ACC - при использовании прямого адреса. Первый способ предпочтительнее, однако, не всегда применим.

## Команды арифметических операций микроконтроллера 8051

В данную группу входят 24 команды, краткое описание которых приведено в таблице. Из нее следует, что микроЭВМ выполняет достаточно широкий набор команд для организации обработки целочисленных данных, включая команды умножения и деления.

В таблице также указаны тип команды (Т) в соответствии с таблицей, ее длина в байтах (В) и время выполнения в машинных циклах (С)

Мнемокод	КОП	Т	В	С	Описание
ADD A, Rn	00101rrr	1	1	1	$(A) \leftarrow (A) + (Rn)$
ADD A, ad	100101	3	2	1	$(A) \leftarrow (A) + (ad)$
ADD A, @Ri	0010011i	1	1	1	$(A) \leftarrow (A) + ((Ri))$
ADD A, #d	00100100	2	2	1	$(A) \leftarrow (A) + \#d$
ADDC A, Rn	00111rrr	1	1	1	$(A) \leftarrow (A) + (Rn) + (C)$
ADDC A, ad	110101	3	2	1	$(A) \leftarrow (A) + (ad) + (C)$
ADDC A, @Ri	0011011i	1	1	1	$(A) \leftarrow (A) + ((Ri)) + (C)$
ADDC A, #d	110100	2	2	1	$(A) \leftarrow (A) + \#d + (C)$
DAA	11010100	1	1	1	Десятичная коррекция аккумулятора
SUBB A, Rn	10011rrr	1	1	1	$(A) \leftarrow (A) - (Rn) - (C)$
SUBB A, ad	10010101	3	2	1	$(A) \leftarrow (A) - (ad) - (C)$
SUBB A, @Ri	1001011i	1	1	1	$(A) \leftarrow (A) - ((Ri)) - (C)$
SUBB A, #d	10010100	2	2	1	$(A) \leftarrow (A) - \#d - (C)$
INC A	100	1	1	1	$(A) \leftarrow (A) + 1$
INC Rn	00001rrr	1	1	1	$(Rn) \leftarrow (Rn) + 1$
INC ad	101	3	2	1	$(ad) \leftarrow (ad) + 1$
INC @Ri	0000011i	1	1	1	$((Ri)) \leftarrow ((Ri)) + 1$
INC DPTR	10100011	1	1	2	$(DPTR) \leftarrow (DPTR) + 1$
DEC A	10100	1	1	1	$(A) \leftarrow (A) - 1$
DEC Rn	00011rrr	1	1	1	$(Rn) \leftarrow (Rn) - 1$
DEC ad	10101	3	2	1	$(ad) \leftarrow (ad) - 1$
DEC @Ri	0001011i	1	1	1	$((Ri)) \leftarrow ((Ri)) - 1$
MUL AB	10100100	1	1	4	$(B)(A) \leftarrow (A) * (B)$
DIV AB	10000100	1	1	4	$(A).(B) \leftarrow (A) / (B)$



По результату выполнения команд ADD, ADDC, SUBB, MUL и DIV устанавливаются флаги PSW, структура которых приведена в таблице.

Флаг C устанавливается при переносе из разряда D7, т. е. в случае, если результат не помещается в восемь разрядов; флаг 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, т. е. в случае, если результат не помещается в семь разрядов и восьмой не может быть интерпретирован как знаковый. Этот признак служит для организации обработки чисел со знаком.

Наконец, флаг P устанавливается и сбрасывается аппаратно. Если число единичных бит в аккумуляторе нечетно, то  $P = 1$ , в противном случае  $P = 0$ .

## Команды логических операций микроконтроллера 8051

В этой группе 25 команд, их краткое описание приведено в таблице. Нетрудно видеть, что эти команды позволяют выполнять операции над байтами: логическое И ( $/$ ), логическое ИЛИ ( $/$ ), исключающее ИЛИ ( $(+)$ ), инверсию (NOT), сброс в нулевое значение и сдвиг. Команды, оперирующие отдельными битами, описаны далее.

В таблице также указаны тип команды (Т) в соответствии с таблицей, ее длина в байтах (В) и время выполнения в машинных циклах (С)

Мнемокод	КОП	Т В С	Описание
ANL A, Rn	01011rrr	1 1 1	(A) <- (A) / (Rn)
ANL A, ad	1010101	3 2 1	(A) <- (A) / (ad)
ANL A, @Ri	1010111	1 1 1	(A) <- (A) / ((Ri))
ANL A, #d	1010100	2 2 1	(A) <- (A) / #d
ANL ad, A	1010010	3 2 1	(ad) <- (ad) / (A)
ANL ad, #d	1010011	7 3 2	(ad) <- (ad) / #d
ORL A, Rn	01001rrr	1 1 1	(A) <- (A) / (Rn)
ORL A, ad	1000101	3 2 1	(A) <- (A) / (ad)
ORL A, @Ri	0100011i	1 1 1	(A) <- (A) / ((Ri))
ORL A, #d	01000100	2 2 1	(A) <- (A) / #d
ORL ad, A	1000010	3 2 1	(ad) <- (ad) / A
ORL ad, #d	1000011	7 3 2	(ad) <- (ad) / #d
XRL A, Rn	01101rrr	1 1 1	(A) <- (A) (+) (Rn)
XRL A, ad	1100101	3 2 1	(A) <- (A) (+) (ad)
XRL A, @Ri	0110011i	1 1 1	(A) <- (A) (+) ((Ri))
XRL A, #d	1100100	2 2 1	(A) <- (A) (+) #d
XRL ad, A	1100010	3 2 1	(ad) <- (ad) (+) A
XRL ad, #d	1100011	7 3 2	(ad) <- (ad) (+) #d
CLR A	11100100	1 1 1	(A) <- 0
CPL A	11110100	1 1 1	(A) <- NOT(A)
SWAP A	11000100	1 1 1	(A0-3) <-> (A4-7)
RL A	100011	1 1 1	Циклический сдвиг влево
RLC A	110011	1 1 1	Сдвиг влево через перенос
RR A	11	1 1 1	Циклический сдвиг вправо
RRC A	10011	1 1 1	Сдвиг вправо через перенос

## Команды операций над битами микроконтроллера 8051

Группа состоит из 12 команд, краткое описание которых приведено в таблице. Эти команды позволяют выполнять операции над отдельными битами: сброс, установку, инверсию бита, а также логические И (/) и ИЛИ (/). В качестве "логического" аккумулятора, участвующего во всех операциях с двумя операндами, выступает признак переноса C (разряд D7 PSW), в качестве операндов могут использоваться 128 бит из резидентной памяти данных и регистры специальных функций, допускающие адресацию отдельных бит. В таблице также указаны тип команды (Т) в соответствии с таблицей, ее длина в байтах (В) и время выполнения в машинных циклах (С)

Мнемокод	КОП	Т В С	Описание
CLR C	11000011	1 1 1	(C) <- 0
CLR bit	11000010	4 2 1	(bit) <- 0
SETB C	11010011	1 1 1	(C) <- 1
SETB bit	11010010	4 2 1	(bit) <- 1
CPL C	10110011	1 1 1	(C) <- NOT(C)
CPL bit	10110010	4 2 1	(bit) <- NOT (bit)
ANL C, bit	10000010	4 2 2	(C) <- (C) / (bit)
ANL C, /bit	10110000	4 2 2	(C) <- (C) / NOT(bit)
ORL C, bit	1110010	4 2 2	(C) <- (C) / (bit)
ORL C, /bit	10100000	4 2 2	(C) <- (C) / NOT(bit)
MOV C, bit	10100010	4 2 1	(C) <- (bit)
MOV bit, C	10010010	4 2 2	(bit) <- (C)

## Команды передачи управления

Группа представлена командами безусловного и условного переходов, командами вызова подпрограмм и командами возврата из подпрограмм.

В таблице также указаны тип команды (Т) в соответствии с таблицей, ее длина в байтах (В) и время выполнения в машинных циклах (С)

Мnemonic	КОП	Т В С	Описание
LJMP ad16	00000010	12 3 2	Длинный безусловный переход по всей памяти
AJMP ad11	1	6 2 2	Безусловный переход в пределах страницы 2 Кбайт
SJMP rel	10000000	5 2 2	Безусловный переход в пределах страницы 256 байт
JMP @A+DPTR	1110011	1 1 2	Безусловный переход по косвенному адресу
JZ rel	1100000	5 2 2	Переход, если нуль
JNZ rel	1110000	5 2 2	Переход, если не нуль
JC rel	1000000	5 2 2	Переход, если бит переноса установлен
JNC rel	1010000	5 2 2	Переход, если бит переноса не установлен
JB bit, rel	100000	11 3 2	Переход, если бит установлен
JNB bit, rel	110000	11 3 2	Переход, если бит не установлен
JBC bit, rel	10000	11 3 2	Переход, если бит установлен со сбросом бита
DJMZ Rn, rel	11011rrr	5 2 2	Команда цикла
DJNZ ad, rel	11010101	8 3 2	Команда цикла
CJNE: A, ad, rel	10110101	8 3 2	Сравнение аккумулятора с байтом и переход, если не равно
CJME A, #d, rel	10110100	10 3 2	Сравнение аккумулятора с константой и переход, если неравно
CJNE: Rn, #d, rel	10111rrr	10 3 2	Сравнение регистра с константой и переход, если не равно
CJNE: @Ri, #d, rel	1011011i	10 3 2	Сравнение байта памяти с константой и переход, если не равно
LCALL ad16	10010	12 3 2	Длинный вызов подпрограммы во всей памяти
ACALL ad11	10001	6 2 2	Вызов подпрограммы в пределах страницы 2 Кбайт
RET	100010	1 1 2	Возврат подпрограммы
RETI	110010	1 1 2	Возврат подпрограммы обработки прерывания
NOP	0	1 1 1	Пустая операция

Команда безусловного перехода LJMP (L - long - длинный) осуществляет переход по абсолютному 16-битному адресу, указанному в теле команды, т. е. команда обеспечивает переход в любую точку памяти программ.

Действие команды AJMP (A - absolute - абсолютный) аналогично команде LJMP, однако в теле команды указаны лишь 11 младших разрядов адреса. Поэтому переход осуществляется в пределах страницы размером 2 Кбайт, при этом надо иметь в виду, что сначала содержимое счетчика команд увеличивается на 2 и только потом заменяются 11 разрядов адреса.

В отличие от предыдущих команд, в команде SJMP (S - short - короткий) указан не абсолютный, а относительный адрес перехода. Величина смещения rel рассматривается как число со знаком, а, следовательно, переход возможен в пределах - 128...+127 байт относительно адреса команды, следующей за командой SJMP.

Команда косвенного перехода JMP @A+DPTR позволяет вычислять адрес перехода в процессе выполнения самой программы.

Командами условного перехода можно проверять следующие условия:

1. JZ - аккумулятор содержит нулевое значение;
2. JNZ - аккумулятор содержит не нулевое значение
3. JC - бит переноса C установлен;
4. JNC - бит переноса C не установлен;
5. JB - прямо адресуемый бит равен 1
6. JNB - прямо адресуемый бит равен 0;
7. JBC - прямо адресуемый бит равен 1 и сбрасывается в нулевое значение при выполнении команды.

Все команды условного перехода рассматриваемых микроконтроллеров содержат короткий относительный адрес, т. е. переход может осуществляться в пределах -128... +127 байт относительно следующей команды.

Команда DJNZ предназначена для организации программных циклов. Регистр Rn или байт по адресу ad, указанные в теле команды, содержат счетчик повторений цикла, а смещение rel - 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 CJN удобна для реализации процедур ожидания внешних событий. В теле команды указаны "координаты" двух байт и относительный адрес перехода rel. В качестве двух байт могут быть использованы, например, значения содержимого аккумулятора и прямо адресуемого байта или косвенно адресуемого байта и константы. При выполнении команды значения указанных двух байт сравниваются и в случае, если они не одинаковы, осуществляется переход. Например, команда

`WAIT: CJNE A, P0, WAIT`

будет выполняться до тех пор, пока значения на линиях порта P0 не совпадут со значениями содержимого аккумулятора.

Действие команд вызова процедур полностью аналогично действию команд безусловного перехода. Единственное отличие состоит в том, 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 прерывание обслуженного уровня. Команды RET и RETI не различают, какой командой - LCALL или ACALL - была вызвана подпрограмма, так как и в том, и в другом случае в стеке сохраняется полный 16-разрядный адрес возврата.

В заключение следует отметить, что большинство Ассемблеров допускают обобщенную мнемонику JMP - для команд безусловного перехода и CALL - для команд вызова подпрограмм. Конкретный тип команды определяется Ассемблером, исходя из "длины" перехода или вызова,

## Таблицы адресов битовых областей памяти микроконтроллера 8051

### Память данных

Адрес байта	Адреса битов по разрядам							
	D7	D6	D5	D4	D3	D2	D1	D0
2FH	7F	7E	7D	7C	7B	7A	79	78
2EH	77	76	75	74	73	72	71	70
2DH	6F	6E	6D	6C	6B	6A	69	68
2CH	67	66	65	64	63	62	61	60
2BH	5F	5E	5D	5C	5B	5A	59	58
2AH	57	56	55	54	53	52	51	50
29H	4F	4E	4D	4C	4B	4A	49	48
28H	47	46	45	44	43	42	41	40
27H	3F	3E	3D	3C	3B	3A	39	38
26H	37	36	35	34	33	32	31	30
25H	2F	2E	2D	2C	2B	2A	29	28
24H	27	26	25	24	23	22	21	20
23H	1F	1E	1D	1C	1B	1A	19	18
22H	17	16	15	14	13	12	11	10
21H	0F	0E	0D	0C	0B	0A	9	8
20H	7	6	5	4	3	2	1	0

ПРИМЕЧАНИЕ. Адрес прямо адресуемых битов может быть записан либо в виде выражения (АдресБайта).(Разряд), например выражение 21.3 означает адрес третьего разряда ячейки памяти с адресом 21H, либо в виде абсолютного битового адреса, который для данного бита равен (см. таблицу) 0B.

## Регистры специальных функций

Адрес	Адреса битов по разрядам								Имя регистра
байта	D7	D6	D5	D4	D3	D2	D1	D0	Name
F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
...	...								...
E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
...	...								...
D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
...	...								...
B8H	-	-	-	BC	BB	BA	B9	B8	IP
...	...								...
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
...	...								...
A8H	AF	-	-	AC	AB	AA	A9	A8	IE
...	...								...
A0H	A7	A6	A5	A4	A3	A2	A1	A0	P2
...	...								...
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
...	...								...
90H	97	96	95	94	93	92	91	90	P1
...	...								...
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
...	...								...
80H	87	86	85	84	83	82	81	80	P0



ПРИМЕЧАНИЕ. Адрес прямо адресуемых битов может быть записан либо в виде выражения (НазваниеРегистра).(Разряд), например выражение `SCON.3` означает адрес третьего разряда регистра `SCON`, либо в виде абсолютного битового адреса, который для данного бита равен (см. таблицу) `9B`. Кроме того, некоторые биты управляющих регистров имеют собственные названия, так например данный бит имеет название `TB8`.