

2. Управляющие восьмиразрядные микроконтроллеры семейства MCS-51 фирмы Intel и совместимые с ними.

Несмотря на непрерывное развитие и появление все новых и новых 16- и 32-разрядных микроконтроллеров и микропроцессоров, наибольшая доля мирового микропроцессорного рынка и по сей день остается за 8-разрядными устройствами. Согласно данным компании Semico Research Corp., Phoenix, в 1996 году общий мировой объем продаж микроконтроллеров всех типов составил \$11,4 миллиарда, при этом \$5,56 миллиарда (или 48.6%) пришлось на долю 8-разрядных кристаллов. Это в 2,5 раза больше объема продаж ближайших конкурентов: 16-разрядных микроконтроллеров (\$2.1 млрд.) и DSP (\$2.4). По всем прогнозам аналитических компаний на ближайшие 5 лет лидирующее положение 8-разрядных микроконтроллеров на мировом рынке сохранится.

В настоящее время среди всех 8-разрядных микроконтроллеров - семейство MCS-51 является несомненным чемпионом по количеству разновидностей и количеству компаний, выпускающих его модификации. Оно получило свое название от первого представителя этого семейства - микроконтроллера 8051, выпущенного в 1980 году на базе технологии HMOS. Удачный набор периферийных устройств, возможность гибкого выбора внешней или внутренней программной памяти и приемлемая цена обеспечили этому микроконтроллеру успех на рынке. С точки зрения технологии микроконтроллер 8051 являлся для своего времени очень сложным изделием - в кристалле было использовано 128 тыс. транзисторов, что в 4 раза превышало количество транзисторов в 16-разрядном микропроцессоре 8086.

Важную роль в достижении такой высокой популярности семейства 8051 сыграла открытая политика фирмы Intel, родоначальницы архитектуры, направленная на широкое распространение лицензий на ядро 8051 среди большого количества ведущих полупроводниковых компаний мира.

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

Все микроконтроллеры из семейства MCS-51 имеют общую систему команд. Наличие дополнительного оборудования влияет только на количество регистров специального назначения.

Основными производителями клонов 51-го семейства в мире являются фирмы Philips, Siemens, Intel, Atmel, Dallas, Temic, Oki, AMD, MHS, Gold Star, Winbond, Silicon Systems и ряд других.

В рамках СССР производство микроконтроллера 8051 осуществлялось в Киеве, Воронеже (1816BE31/51, 1830BE31/51), Минске (1834BE31) и Новосибирске (1850BE31).

Микроконтроллеры данного семейства выпускаются в PLCC, DIP и QFP корпусах и могут работать в следующих температурных диапазонах:

- коммерческий (0°C — +70°C);
- расширенный (-40°C — +85°C);
- для военного использования (-55°C — +125°C).

Примерами микроконтроллеров семейства MCS-51 с расширенными возможностями могут Расширения микроконтроллеров MCS-51/52 служить 8XC51FA, 8XC51GB, 80C152.

2.1. Структурная организация микроконтроллера i8051.

2.1.1. Общие характеристики.

Микроконтроллер семейства 8051 имеют следующие аппаратные особенности:

- внутреннее ОЗУ объемом 128 байт;
- четыре двунаправленных побитно настраиваемых восьмиразрядных порта ввода-вывода;
- два 16-разрядных таймера-счетчика;
- встроенный тактовый генератор;
- адресация 64 Кбайт памяти программ и 64 Кбайт памяти данных;
- две линии запросов на прерывание от внешних устройств;
- интерфейс для последовательного обмена информацией с другими микроконтроллерами или персональными компьютерами.

Микроконтроллер 8751 снабжен УФ ПЗУ объемом 4 Кбайт.

Функциональная схема микроконтроллера семейства 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).

2.1.2. 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).

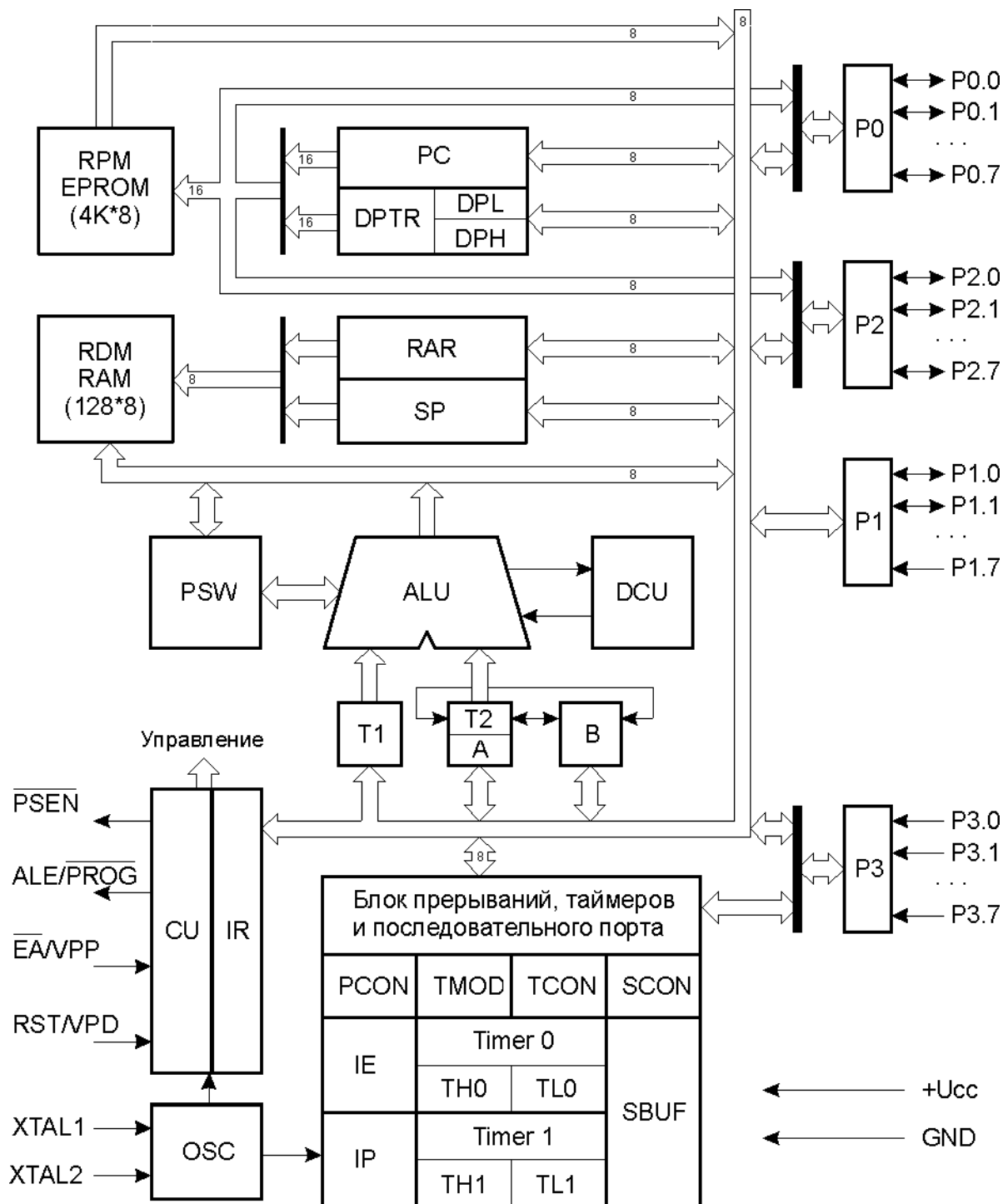


Рис. 1. Структурная схема микроконтроллера KM1816BE51

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 “танделы” 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.

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.

2.1.3. Назначение выводов микроконтроллера 8051.

P1.0	□ 1	40	□ Vcc
P1.1	□ 2	39	□ P0.0 (AD0)
P1.2	□ 3	38	□ P0.1 (AD1)
P1.3	□ 4	37	□ P0.2 (AD2)
P1.4	□ 5	36	□ P0.3 (AD3)
P1.5	□ 6	35	□ P0.4 (AD4)
P1.6	□ 7	34	□ P0.5 (AD5)
P1.7	□ 8	33	□ P0.6 (AD6)
RESET	□ 9	32	□ P0.7 (AD7)
(RxD) P3.0	□ 10	31	□ EA/Vpp
(TxD) P3.1	□ 11	30	□ ALE/ $\overline{\text{P}}\text{ROG}$
(INT0) P3.2	□ 12	29	□ PSEN
(INT1) P3.3	□ 13	28	□ P2.7 (A15)
(T0) P3.4	□ 14	27	□ P2.6 (A14)
(T1) P3.5	□ 15	26	□ P2.5 (A13)
(WR) P3.6	□ 16	25	□ P2.4 (A12)
(RD) P3.7	□ 17	24	□ P2.3 (A11)
XTAL2	□ 18	23	□ P2.2 (A10)
XTAL1	□ 19	22	□ P2.1 (A9)
Vss	□ 20	21	□ P2.0 (A8)

Рис. 2. Назначение выводов 8051.

Обозначения на этом рисунке:

- U_{ss} — потенциал общего провода ("земли");
- U_{cc} — основное напряжение питания +5 В;
- X1, X2 — выводы для подключения кварцевого резонатора;
- RST — вход общего сброса микроконтроллера;
- PSEN — разрешение внешней памяти программ; выдается только при обращении к внешнему ПЗУ;
- ALE — строб адреса внешней памяти;
- EA — отключение внутренней программной памяти; уровень 0 на этом входе заставляет микроконтроллер выполнять программу только внешнего ПЗУ; игнорируя внутреннее (если последнее имеется);
- P1 — восьми битный квази двунаправленный порт ввода/вывода: каждый разряд порта может быть запрограммирован как на ввод, так и на вывод информации, независимо от состояния других разрядов;

- P2 — восьми битный квази двунаправленный порт, аналогичный P1; кроме того, выводы этого порта используются для выдачи адресной информации при обращении к внешней памяти программ или данных (если используется 16-битовая адресация последней). Выводы порта используются при программировании 8751 для ввода в микроконтроллер старших разрядов адреса;
- P3 — восьми битный квази двунаправленный порт, аналогичный P1; кроме того, выводы этого порта могут выполнять ряд альтернативных функций, которые используются при работе таймеров, порта последовательного ввода-вывода, контроллера прерываний, и внешней памяти программ и данных;
- P0 — восьми битный двунаправленный порт ввода-вывода информации: при работе с внешними ОЗУ и ПЗУ по линиям порта в режиме временного мультиплексирования выдается адрес внешней памяти, после чего осуществляется передача или прием данных.

3. Организация ОЗУ, ПЗУ и регистров микроконтроллера 8051.

3.1.1. Память программ (ПЗУ).

Как и у большинства микроконтроллеров, у микроконтроллеров семейства 8051, память программ и память данных являются самостоятельными и независимыми друг от друга устройствами, адресуемыми различными командами и управляющими сигналами.

Объем **встроенной памяти программ**, расположенной на кристалле микроконтроллера 8051 и 8751, равен 4 Кбайт. При обращении к **внешней памяти программ** все микроконтроллеры семейства 8051 всегда используют 16-разрядный адрес, что обеспечивает им доступ к 64 Кбайт ПЗУ. Микроконтроллер обращается к программной памяти при чтении кода операции и операндов (используя счетчик команд PC), а также при выполнении команд переноса байта из памяти программ в аккумулятор. При выполнении команд переноса данных адресация ячейки памяти программ, из которой будут прочитаны данные, может осуществляться с использованием как счетчика PC, так и специального двухбайтового регистра-указателя данных DPTR.

3.1.2. Память данных (ОЗУ).

Объем **расположенной на кристалле памяти данных**—128 байт. Объем **внешней памяти** данных может достигать 64 Кбайт. Первые 32 байта организованы в четыре банка регистров общего назначения, обозначаемых соответственно банк 0 — банк 3. Каждый из них состоит из восьми регистров R0 — R7. В любой момент программе доступен только один банк регистров, номер которого содержится в третьем и четвертом битах слова состояния программы PSW (см. ниже).

Оставшееся адресное пространство может конфигурироваться разработчиком по своему усмотрению: в нем располагаются стек, системные и пользовательские области данных. Обращение к ячейкам памяти данных возможно двумя способами. Первый способ — прямая адресация ячейки памяти. В этом случае адрес ячейки является операндом соответствующей команды. Второй способ — косвенная адресация с помощью регистров R0 или R1: перед выполнением соответствующей команды в один из них должен быть занесен адрес ячейки, к которой необходимо обратиться.

Для обращения к внешней памяти данных используется только косвенная адресация с помощью регистров R0 и R1 или с помощью 16-разрядного регистра-указателя DPTR. Он относится к группе регистров специальных функций, и с его помощью можно адресовать все 64 Кбайта внешней памяти.

Часть памяти данных представляет собой так называемую битовую область, в ней имеется возможность при помощи специальных битовых команд адресовываться к каждому разряду ячеек памяти. Адрес прямо адресуемых битов может быть записан либо в виде (Адрес Байта).(Разряд), например выражение 21.3 означает третий разряд ячейки памяти с адресом 21H, либо в виде абсолютного битового адреса. Соответствие этих двух способов адресации можно определить по таблице.

Адрес байта	Адреса битов по разрядам								
	Adr	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	09	08	
20H	07	06	05	04	03	02	01	00	

Таблица.1. Адреса битовых областей памяти микроконтроллера 8051

ПРИМЕЧАНИЕ. Адрес прямо адресуемых битов может быть записан либо в виде выражения (Адрес Байта).(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.

3.1.3.Регистры специальных функций.

Адрес байта	Адреса битов по разрядам								Имя регистра
Adr	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

Таблица.2. Карта адресуемых битов в блоке регистров специальных функций

ПРИМЕЧАНИЕ. Адрес прямо адресуемых битов может быть записан либо в виде выражения (Название Регистра).(Разряд), например выражение SCON.3 означает адрес третьего разряда регистра SCON, либо в виде абсолютного битового адреса, который для данного бита равен (см. таблицу) 9В. Кроме того, некоторые биты управляющих регистров имеют собственные названия, так например данный бит имеет название ТВ8

3.1.4.Регистры специальных функций.

К адресному пространству памяти данных примыкает адресное пространство регистров специальных функций SFR (Special Function Register).

Адрес	Символ	Наименование
0E0H	*ACC	Аккумулятор (Accumulator)
0F0H	*B	Регистр расширитель аккумулятора (Multiplication Register)
0D0H	*PSW	Слово состояния программы (Program Status Word)
080H	*P0	Порт 0 (SFR P0)
090H	*P1	Порт 1 (SFR P1)
0A0H	*P2	Порт 2 (SFR P2)
0B0H	*P3	Порт 3 (SFR P3)
081H	SP	Регистр указатель стека (Stack Pointer)
083H	DPH	Старший байт регистра указателя данных DPTR (Data Pointer High)
082H	DPL	Младший байт регистра указателя данных DPTR (Data Pointer Low)
08CH	TH0	Старший байт таймера 0 ()
08AH	TL0	Младший байт таймера 0 ()
08DH	TH1	Старший байт таймера 1 ()
08BH	TL1	Младший байт таймера 1 ()
089H	TMOD	Регистр режимов таймеров счетчиков (Timer/Counter Mode Control Register)
088H	*TCON	Регистр управления статуса таймеров (Timer/Counter Control Register)
0B8H	*IP	Регистр приоритетов (Interrupt Priority Control Register)
0A8H	*IE	Регистр маски прерывания (Interrupt Enable Register)
087H	PCON	Регистр управления мощностью (Power Control Register)
098H	*SCON	Регистр управления приемопередатчиком (Serial Port Control Register)
099H	SBUF	Буфер приемопередатчика (Serial Data Buffer)

Таблица.3. Размещение регистров специальных функций в пространстве SFR

Примечание. Регистры, символ которых отмечен знаком (*), допускают адресацию своих отдельных бит при использовании команд из группы команд операций над битами.

Адреса, по которым расположены эти регистры, приведены в таблице.

Отметим, что регистры занимают только часть 128-байтового адресного пространства. Те ячейки памяти с адресами 80H-0FFH, которые не заняты регистрами, физически отсутствуют, на кристаллах микроконтроллеров семейства 8051 при обращении к ним можно прочитать лишь код команды возврата.

Регистры специальных функций управляют работой блоков, входящих в микроконтроллер.

- Регистры-защелки **SFR** параллельных портов **P0...P3** - служат для ввода-вывода информации.
- Две регистровые пары с именами **TH0, TL0** и **TH1, TL1** представляют собой регистры, двух программно-управляемых 16-битных таймеров-счетчиков.
- Режимы таймеров-счетчиков задаются с использованием регистра **TMOD**, а управление ими осуществляется с помощью регистра **TCON**.
- Для управления режимами энергопотребления микро-ЭВМ используется регистр **PCON**.
- Регистры **IP** и **IE** управляют работой системы прерываний микро-ЭВМ,
- регистры **SBUF** и **SCON** — работой приемопередатчика последовательного порта.
- Регистр-указатель стека **SP** в микро-ЭВМ рассматриваемого семейства — восьми битный. Он может адресовать любую область внутренней памяти данных. В отличие от микропроцессора KP580BM80, у микро-ЭВМ семейства 8051 стек «растет вверх», т.е. перед выполнением команды **PUSH** или **CALL** содержимое **SP** инкрементируется, после чего производится запись информации в стек. Соответственно при извлечении информации из стека регистр **SP** декрементируется после извлечения информации. В процессе инициализации микро-ЭВМ 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.
- Регистр-указатель данных **DPTR** чаще всего используют для фиксации 16-битного адреса в операциях обращения к внешней памяти программ и данных. С точки зрения программиста он может выступать как в виде одного 16-битного регистра, так и в виде двух независимых регистров **DPL** и **DPH**
- Аккумулятор (**ACC**) 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 АЛУ формируется ряд признаков операции (флагов), которые фиксируются в регистре **PSW**.
- Регистр **B** используется как источник и как приемник при операциях умножения и деления, обращение к нему, как к регистру **SFR**, производится аналогично аккумулятору.
- При выполнении ряда команд в арифметико-логическом устройстве (АЛУ) формируются признаки операций — флаги, которые фиксируются в регистре **PSW**.

3.1.5.Регистр флагов (PSW).

Символ	Позиция	Имя и назначение																				
P	PSW.0	Флаг приоритета. Устанавливается и сбрасывается аппаратно в каждом цикле команды и фиксирует нечетное/четное число единичных бит в аккумуляторе																				
-	PSW.1	Не используется																				
OV	PSW.2	Флаг переполнения. Устанавливается и сбрасывается аппаратно при выполнении арифметических операций																				
RS0 - RS1	PSW.3 - PSW.4	Биты выбора используемого банка регистров. Могут быть изменены программным путем																				
		<table border="1"> <thead> <tr> <th>RS0</th> <th>RS1</th> <th>Банк</th> <th>Границы адресов ОЗУ</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>00H - 07H</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>08H - 0FH</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> <td>10H - 17H</td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>18H - 1FH</td> </tr> </tbody> </table>	RS0	RS1	Банк	Границы адресов ОЗУ	0	0	0	00H - 07H	1	0	1	08H - 0FH	0	1	2	10H - 17H	1	1	3	18H - 1FH
		RS0	RS1	Банк	Границы адресов ОЗУ																	
		0	0	0	00H - 07H																	
		1	0	1	08H - 0FH																	
0	1	2	10H - 17H																			
1	1	3	18H - 1FH																			
F0	PSW.5	Флаг пользователя. Может быть установлен, сброшен или проверен программой пользователя																				
AC	PSW.6	Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратными средствами при выполнении команд сложения и вычитания и сигнализирует о переносе или заеме в бите 3 аккумулятора																				
C	PSW.7	Флаг переноса. Устанавливается и сбрасывается как аппаратно, так и программным путем																				

Таблица.4. Перечень флагов, их символические имена и условия формирования

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 четырех 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.

3.1.6. 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.

3.2. Организация портов ввода вывода микроконтроллера 8051.

3.2.1. Общие сведения.

- Количество портов - 4. Название - P0...P3, они адресуются как регистры специальных функций.
- Разрядность - 8 с возможностью побитной адресации разрядов.
- Направление обмена информацией через порты - все порты двунаправленные, причем имеется возможность в каждом порту часть разрядов использовать для ввода данных, а часть для вывода.

3.2.2. Альтернативные функции.

Из-за ограниченного количества выводов корпуса ИМС микроконтроллера, большинство выводов используется для выполнения двух функций - в качестве линий портов и для альтернативных функций

- **Порты P0 и P2** используются при обращении к внешней памяти. При этом на выходах P0 младший байт адреса внешней памяти мультиплексируется с вводимым/выводимым байтом. Выходы P2 содержат старший байт адреса внешней памяти, если адрес 16-разрядный. При использовании восьмиразрядного адреса портом P2 можно пользоваться для ввода-вывода информации обычным образом. При обращении к внешней памяти в P0 автоматически заносятся 1 во все биты. Информация в P2 при этом остается неизменной.
- **Порт P3** помимо обычного ввода и вывода информации используется для формирования и приема специальных управляющих и информационных сигналов. Разряды порта (все или частично) при этом могут выполнять следующие альтернативные функции:

Альтернативные функции могут быть активированы только в том случае, если в соответствующие биты порта P3 предварительно занесены 1. Неиспользуемые альтернативным образом разряды могут работать как обычно.

Вывод порта

Альтернативная функция

P3.0	RXD - вход последовательного порта
P3.1	TXD - выход последовательного порта
P3.2	INT0 - внешнее прерывание 0
P3.3	INT1 - внешнее прерывание 1
P3.4	T0 - вход таймера-счетчика 0
P3.5	T1 - вход таймера-счетчика 1
P3.6	WR - строб записи во внешнюю память данных
P3.7	RD - строб чтения из внешней памяти данных

3.2.3. Устройство портов.

Каждый из портов содержит регистр-защелку (SFR P0 — SFR P3), выходную цепь и входной буфер.

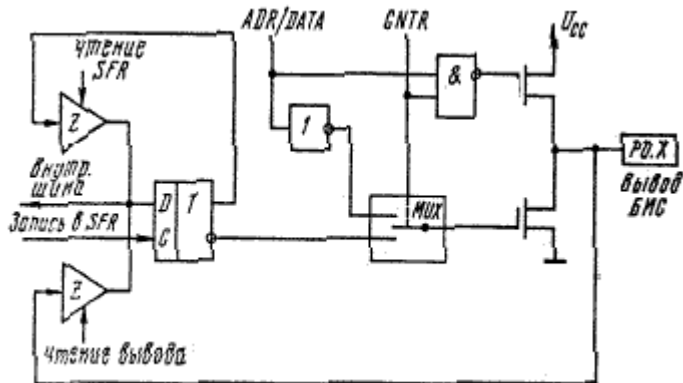


Рис.3. Порт P0

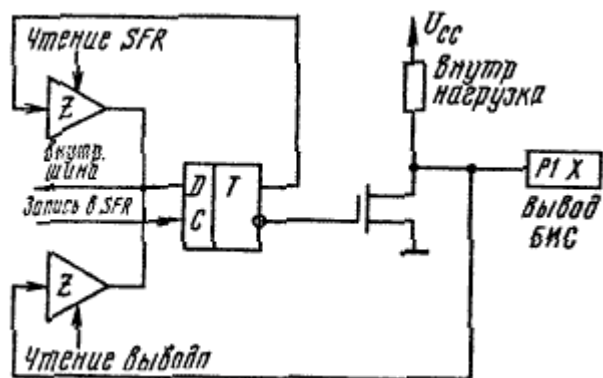


Рис.4. Порт P1

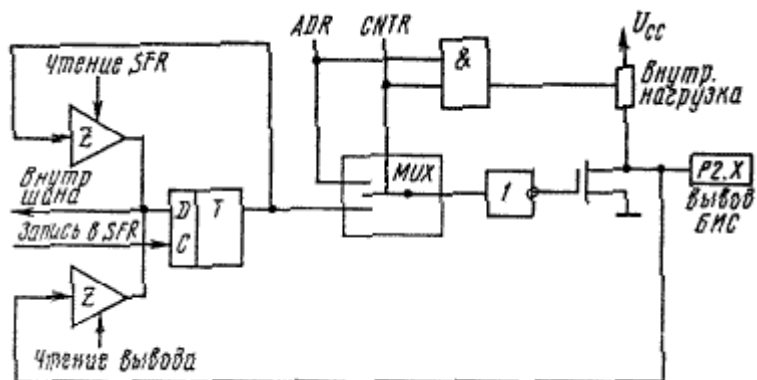


Рис.5. Порт P2

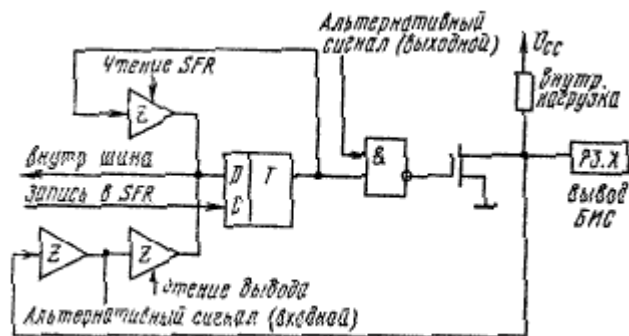


Рис.6. Порт P3

На рисунке изображены функциональные схемы регистров-защелок и буферов ввода-вывода всех портов микро-ЭВМ 8051. Каждый из разрядов регистра-защелки SFR является D-триггером, информация в который заносится с внутренней шины данных микроконтроллера по сигналу «Запись в SFR Px» (x= 0, 1, 2, 3) от центрального процессорного элемента (CPU). С прямого выхода D-триггера информация может быть выведена на внутреннюю шину по сигналу «Чтение SFR Px» от CPU, а с вывода микросхемы («из внешнего мира») по сигналу «Чтение выводов Px». Одни команды активизируют сигнал «Чтение SFR PI», другие - «Чтение выводов PI».

3.2.4. Особенности электрических характеристик портов.

Выходные каскады триггеров SFR портов P1 — P3 выполнены на полевых транзисторах с внутренней нагрузкой, в то время как аналогичные каскады триггеров SFR P0—на транзисторах с открытым стоком. Каждая линия любого из портов может независимо использоваться как для ввода, так и для вывода информации (для линий портов P0 и P2 это справедливо тогда, когда они не используются для обращения к внешней памяти).

Для перевода любой линии портов P1 — P3 в режим ввода информации необходимо в соответствующий разряд SFR занести 1. При этом выходной полевой транзистор отключается. Внутренний нагрузочный резистор как бы «подтягивает» потенциал вывода к напряжению питания, в то время как внешняя нагрузка может сделать его нулевым. Выходные каскады порта P0 имеют иную структуру. Нагрузочный полевой транзистор линии порта включен только тогда, когда порт выводит 1 при обращении к внешней памяти. В остальных случаях нагрузочный транзистор отключен. Таким образом, при работе в режиме обычного ввода-вывода информации (как, например, порт P1) выходные каскады порт P0 представляют собой ступени на транзисторах с открытым стоком. Запись 1 в соответствующий бит SFR отключает и второй транзистор, что приводит к тому, что вывод БИС оказывается под «плавающим» потенциалом. Это позволяет использовать линии порта P0 как выходы с высоко импедансным состоянием. Поскольку выходные каскады портов P1 — P3 имеют внутреннюю нагрузку, при переводе в режим ввода информации они становятся источниками тока для микросхемы или транзистора, нагруженных на данный.

3.3. Таймеры / счетчики микроконтроллеров семейства 8051.

В базовых моделях семейства имеются два программируемых 16-битных таймера/счетчика (T/C0 и T/C1), которые могут быть использованы как в качестве таймеров, так и в качестве счетчиков внешних событий. В первом случае содержимое соответствующего таймера/счетчика (далее для краткости T/C) инкрементируется в каждом машинном цикле, т.е. через каждые 12 периодов колебаний кварцевого резонатора, во втором оно инкрементируется под воздействием перехода из 1 в 0 внешнего входного сигнала, подаваемого на соответствующий (T0, T1) вывод микро-ЭВМ 8051. Так как на распознавание периода требуются два машинных цикла, максимальная частота подсчета входных сигналов равна 1/24 частоты резонатора. На длительность периода входных сигналов ограничений сверху нет. Для гарантированного прочтения входной сигнал должен удерживать значение 1, как минимум, в течение одного машинного цикла микро-ЭВМ.

Для управления режимами работы T/C и для организации их взаимодействия с системой прерываний используются два регистра специальных функций (TMOD и TCON), описание которых приведено ниже.

3.3.1.Регистр режима работы таймера/счетчика TMOD

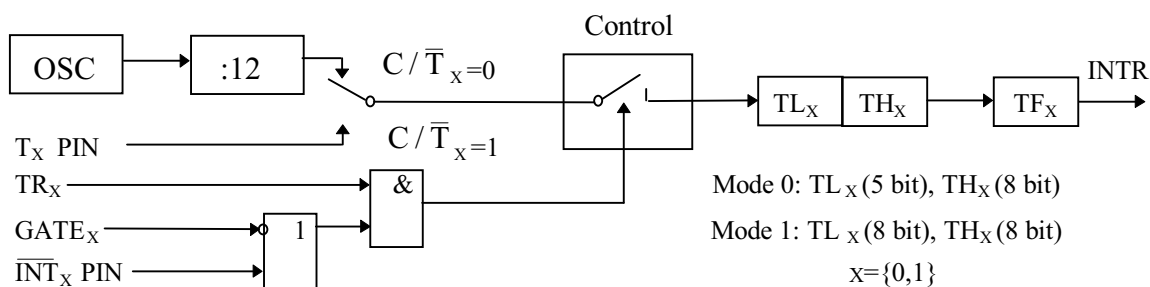
Символ	Позиция	Имя и назначение	
GATE	TMOD.7 для T/C1 и TMOD.3 для T/CO	Управление блокировкой. Если бит установлен, то таймер/счетчик "x" разрешен до тех пор, пока на входе "INTx" высокий уровень и бит управления "TRx" установлен. Если бит сброшен, то T/C разрешается, как только бит управления "TRx" устанавливается	
C/T	TMOD.6 для T/C1 и TMOD.2 для T/CO	Бит выбора режима таймера или счетчика событий. Если бит сброшен, то работает таймер от внутреннего источника сигналов синхронизации. Если; установлен, то работает счетчик от внешних сигналов на входе "Tx"	
M1	TMOD.5 для T/C1 и TMOD.1 для T/CO	Режим работы	
		M1M0	
		0 0	Таймер BE48. "TLx" работает как 5-битный предделитель
		0 1	16 битный таймер/счетчик. "THx" и "TLx" включен последовательно
		1 0	8-битный авто перезагружаемый таймер/счетчик. "THx" хранит значение, которое должно быть перезагружено в "TLx" каждый раз по переполнению
M0	TMOD.4 для T/C1 и TMOD.0 для T/CO	1 1	Таймер/счетчик 1 останавливается. Таймер/счетчик 0: TLO работает как 8-битный таймер/счетчик, и его режим определяется управляющими битами таймера 0. TH0 работает только как 8 битный таймер, и его режим определяется управляющими битами таймера 1

3.3.2.Регистр управления/статуса таймера TCON.

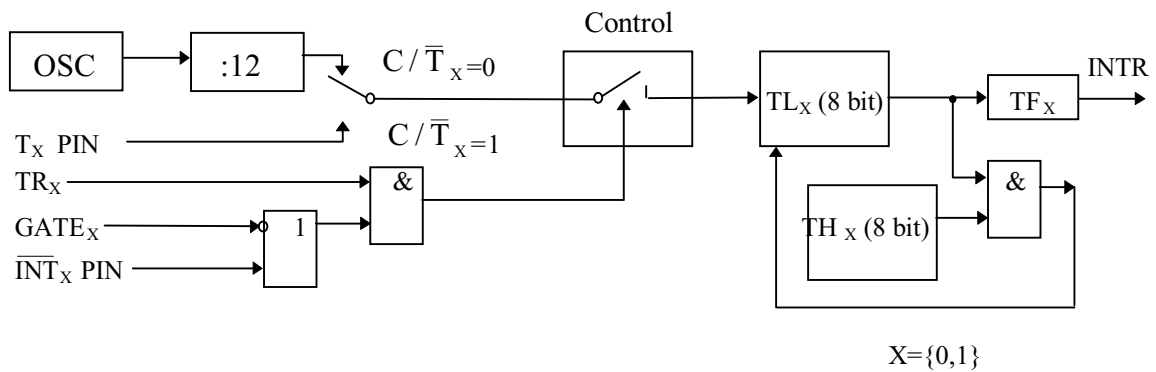
Символ	Позиция	Имя и назначение
TF1	TCON.7	Флаг переполнения таймера 1. Устанавливается аппаратно при переполнении таймера/счетчика. Сбрасывается при обслуживании прерывания аппаратно
TR1	TCON.6	Бит управления таймера 1. Устанавливается, / сбрасывается программой для пуска/останова
TF0	TCON.5	Флаг переполнения таймера 0. Устанавливается аппаратно. Сбрасывается при обслуживании прерывания
TR0	TCON.4	Бит управления таймера 0. Устанавливается / сбрасывается программой для пуска/останова таймера/счетчика
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
IT1	TCON .0	Бит управления типом прерывания 0. Устанавливается / сбрасывается программно для спецификации запроса INT0 (срез/низкий уровень)

3.3.3.Режимы работы таймеров-счетчиков.

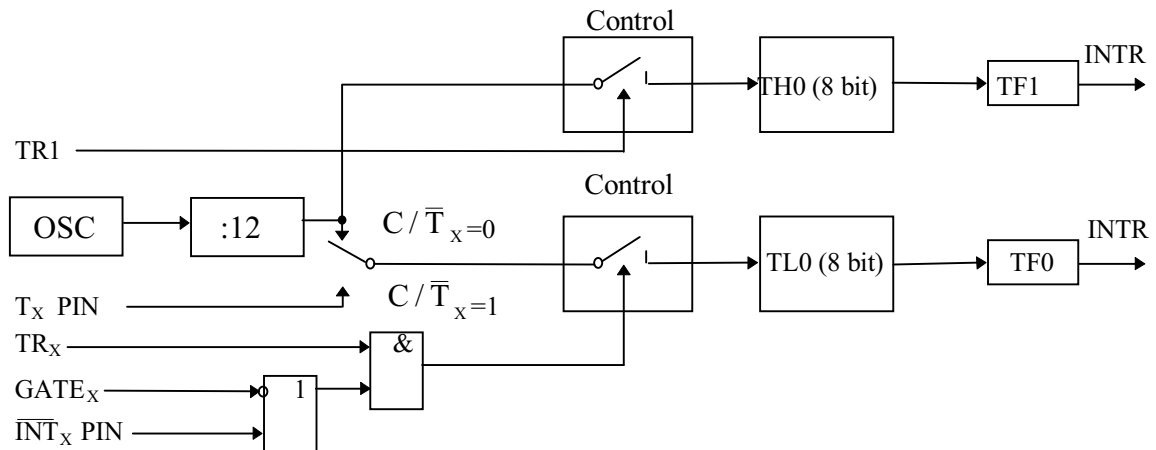
Как следует из описания управляющих бит TMOD, для обоих T/C режимы работы 0, 1 и 2 одинаковы. Режимы 3 для T/CO и T/C1 различны. Рассмотрим кратко работу T/C в каждом из режимов.



а - логика работы T/CO и T/C1 в режимах 0 и 1



б - логика работы T/C0 и T/C1 в режиме 2



в - логика работы T/C0 в режиме 3

Рис.7. Логика работы T/C0 и T/C1 в режимах 0, 1, 2 и 3

- **Режим 0.** Перевод любого T/C в режим 0 делает его похожим на таймер KM1816BE48 (восьми битный счетчик), к входу которого подключен пяти-битный предделитель частоты на 32. Работу T/C в режиме 0 на примере T/C1 иллюстрирует рис а. В этом режиме таймерный регистр имеет разрядность 13 бит. При переходе из состояния "все единицы" в состояние "все нули" устанавливается флаг прерывания от таймера TF 1. Входной синхросигнал таймера 1 разрешен (поступает на вход T/C1), когда управляющий бит TR1 установлен в 1 либо управляющий бит GATE (блокировка) равен 0, либо на внешний вывод запроса прерывания INT1 поступает уровень 1. Отметим попутно, что 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") восьмибитного счетчика TL1 приводит не только к установке флага TF1 (см. рис. б), но и автоматически перезагружает в TL1 содержимое старшего байта (TH 1) таймерного регистра, которое предварительно было задано программным путем. Перегрузка оставляет содержимое TH1 неизменным. В режиме 2 T/C0 и T/C1 также работают совершенно одинаково.
- **Режим 3.** В режиме 3 T/C0 и T/C1 работают по-разному. T/C1 сохраняет неизменным свое текущее содержимое. Иными словами, эффект такое же как и при сбросе управляющего бита TR1 в 0. Работу T/C0 иллюстрирует рис. в. В режиме 3 TL0 и TH0 функционируют как два независимых восьмибитных счетчика. 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 микро-ЭВМ), определяет управляющий бит TR1. При этом TH0 использует флаг переполнения TF1. Режим 3 используется в тех случаях, когда требуется наличие дополнительного восьми битного таймера или счетчика событий. Можно считать, что в этом режиме микро-ЭВМ 8051 имеет в своем составе три таймера/счетчика. В случае же, если T/C0 используется в режиме 3, T/C1 может быть или выключен, или переведен в режим 0, 1 или 2, или может быть использован последовательным портом в качестве генератора частоты передачи.

В модернизированных моделях микроконтроллеров семейства MCS-51 может иметься третий таймер счетчик T/C2 и (или) блок программных счетчиков PCA, которые тоже могут быть использованы для отсчета временных интервалов.

3.4. Последовательный порт микроконтроллера 8051.

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

Кроме того, работой последовательного порта управляют два служебных регистра -

- Регистр управления/статуса приемопередатчика SCON
- Бит SMOD регистра управления мощностью PCON

Запись байта в буфер приводит к автоматической переписи байта в сдвигающий регистр передатчика и инициирует начало передачи байта. Наличие буферного регистра приемника позволяет совмещать операцию чтения ранее принятого байта с приемом очередного. Но если к моменту окончания приема байта предыдущий не был считан из SBUF, то он будет потерян.

Последовательный порт 8051 может работать в четырех различных режимах.

- **Режим 0.** Информация и передается, и принимается через вывод входа приемника (RXi TXi). Принимаются или передается 8 бит данных. Через вывод выхода передатчика (TXD; выдаются импульсы сдвига, которые сопровождают каждый бит. Частота передачи бита информации равна 1/12 частоты кварцевого резонатора
- **Режим 1.** В этом режиме передаются через вывод TXD или принимаются через RXD 10 бит информации: старт-бит (0), 8 бит данных и стоп-бит (1) при приеме информации в бит RB8 регистра управления/статуса приемопередатчика SCON заносится стоп-бит. Скорость приема/передачи — величина переменная и задается таймером.
- **Режим 2.** В этом режиме через вывод TXD передаются или через RXD принимаются 11 бит информации: старт-бит, 8 бит данных, программируемый девятый бит и стоп-бит. При передаче девятый бит данных может принимать значение 0 или 1 или, например, для повышения достоверности передачи путем контроля по четности в него может быть помещено значение признака паритета из слова состояния программы (PSW.0). При приеме девятый бит данных помещается в бит RB8 SCON, а стоп-бит, в отличие от режима 1, теряется. Частота приема/передачи выбирается программой и может быть равна либо 1/32, либо 1/64 частоты резонатора в зависимости от управляющего бита SMOD.
- **Режим 3.** совпадает с режимом 2 во всех деталях, за исключением частоты приема/передачи, которая является величиной переменной и задается таймером.

Во всех случаях передача инициализируется инструкцией, в которой данные перемещаются в SBUF. Прием инициализируется при обнаружении перепада из 1 в 0 на входе приемника. При этом в режиме 0 этот переход должен сопровождаться выполнением условий $R1 = 0$ и $REN = 1$ (см. табл. 8), а для остальных режимов - $REN = 1$.

3.5. Регистр управления/статуса приемопередатчика SCON.

Управление режимом работы приемопередатчика осуществляется через специальный регистр с символическим именем SCON. Этот регистр содержит не только управляющие биты, определяющие режим работы последовательного порта, но и девятый бит принимаемых или передаваемых данных (RB8 и TB8) и биты прерывания приемопередатчика (R1 и T1).

Прикладная программа путем загрузки в старшие биты регистра SCON двухбитного кода определяет режим работы приемопередатчика. Во всех четырех режимах работы передача инициализируется любой командой, в которой буферный регистр SBUF указан как получатель байта. Как уже отмечалось, прием в режиме 0 осуществляется при условии, что $R1 = 0$ и $REN = 1$, в остальных режимах - при условии, что $REN = 1$.

В бите TB8 программно устанавливается значение девятого бита данных, который будет передан в режиме 2 или 3. В бите RB8 в этих режимах фиксируется девятый принимаемый бит данных. В режиме 1 в бит RB8 заносится стоп-бит. В режиме 0 бит RB8 не используется.

Флаг прерывания передатчика T1 устанавливается аппаратно в конце периода передачи стоп-бита во всех режимах. Соответствующая подпрограмма обслуживания прерывания должна сбрасывать бит TL.

Флаг прерывания приемника R1 устанавливается аппаратно в конце периода приема восьмого бита данных в режиме 0 и в середине периода приема стоп-бита в режимах 1, 2 и 3. Подпрограмма обслуживания прерывания должна сбрасывать бит RI.

3.5.1. Функциональное назначение бит регистра управления/статуса приемопередатчика SCON.

Символ	Позиция	Имя и назначение															
SM0	SCON.7	Биты управления режимом работы приемопередатчика. Устанавливаются/сбрасываются программно см. примечание 1															
SM1	SCON.6	<table border="1"> <thead> <tr> <th>SM0</th> <th>SM1</th> <th>Режим работы приемопередатчика</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Сдвигающий регистр расширения ввода/вывода</td> </tr> <tr> <td>0</td> <td>1</td> <td>8 битовый приемопередатчик, изменяемая скорость передачи</td> </tr> <tr> <td>1</td> <td>0</td> <td>9 битовый приемопередатчик. Фиксированная скорость передачи</td> </tr> <tr> <td>1</td> <td>1</td> <td>9 битовый приемопередатчик, изменяемая скорость передачи</td> </tr> </tbody> </table>	SM0	SM1	Режим работы приемопередатчика	0	0	Сдвигающий регистр расширения ввода/вывода	0	1	8 битовый приемопередатчик, изменяемая скорость передачи	1	0	9 битовый приемопередатчик. Фиксированная скорость передачи	1	1	9 битовый приемопередатчик, изменяемая скорость передачи
		SM0	SM1	Режим работы приемопередатчика													
		0	0	Сдвигающий регистр расширения ввода/вывода													
		0	1	8 битовый приемопередатчик, изменяемая скорость передачи													
1	0	9 битовый приемопередатчик. Фиксированная скорость передачи															
1	1	9 битовый приемопередатчик, изменяемая скорость передачи															
SM2	SCON.5	Бит управления режимом приемопередатчика. Устанавливается программно для запрета приема сообщения, в котором девятый бит имеет значение 0															
REN	SCON.4	Бит разрешения приема. Устанавливается/сбрасывается программно для разрешения/запрета приема последовательных данных															
TB8	SCON.3	Передача бита 8. Устанавливается/сбрасывается программно для задания девятого передаваемого бита в режиме 9-битового передатчика															
RB8	SCON.2	Прием бита 8. Устанавливается/сбрасывается аппаратно для фиксации девятого принимаемого бита в режиме 9-битового приемника															
TI	SCON.1	Флаг прерывания передатчика. Устанавливается аппаратно при окончании передачи байта. Сбрасывается программно после обслуживания прерывания															
RI	SCON.0	Флаг прерывания приемника. Устанавливается аппаратно при приеме байта. Сбрасывается программно после обслуживания прерывания															

3.5.2. Скорость приема/передачи информации через последовательный порт.

Скорость приема/передачи, т.е. частота работы приемопередатчика в различных режимах, определяется различными способами.

В режиме 0 частота передачи зависит только от резонансной частоты кварцевого резонатора f_{PE3} :

$$f = f_{PE3}/12.$$

За машинный цикл последовательный порт передает один бит информации. В режимах 1, 2 и 3 скорость приема/передачи зависит от значения управляющего бита SMOD в регистре специальных функций PCON.

3.5.3. Регистр управления мощностью PCON.

Символ	Позиция	Наименование и функция
SMOD	PCON.7	Удвоенная скорость передачи. Если бит установлен в 1, то скорость передачи вдвое больше, чем при SMOD = 0. По сбросу SMOD = 0.
	PCON.6	Не используется
	PCON.5	Не используется
	PCON.4	Не используется
GF1 GF0	PCON.3 PCON.2	Флаги, специфицируемые пользователем (флаги общего назначения)
PD	PCON.1	Бит пониженной мощности. При установке бита в 1 микро-ЭВМ переходит в режим пониженной потребляемой мощности
IDL	PCON.0	Бит холостого хода. Если бит установлен в 1, то микро-ЭВМ переходит в режим холостого хода

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

В режиме 2 частота передачи определяется выражением

$$f = 2^{SMOD} f_{PE3}/64.$$

Иными словами, при SMOD = 0 частота передачи равна 1/64 частоты f_{PE3} , а при SMOD = 1 — 1/32 частоты f_{PE3} .

В режимах 1 и 3 в формировании частоты передачи, кроме управляющего бита SMOD, принимает участие таймер 1. При этом частота передачи f зависит от частоты переполнения f_{OVL1} определяется следующим образом:

$$f = 2^{SMOD} f_{OVL1}/32$$

Прерывание от таймера 1 в этом случае должно быть заблокировано. Сам же таймер может работать как в режиме таймера, так и в режиме счетчика. Номер режима (0, 1, 2) роли не играет. Наиболее типично использование его в режиме таймера с автоперезагрузкой (старшая тетрада TMOD = 0010B). При этом частота передачи определяется выражением:

$$f = 2^{\text{SMOD}} \cdot f_{\text{PE3}} / (32 \times 12 \times (256 - \text{TH1})).$$

Частота приема/передачи (BAUD RATE)	Частота резонатора МГц	Таймер/счетчик 1			
		SMOD	C/T	Режим (MODE)	Перезагружаемое число
Режим 0, макс: 1 МГц	12	X	X	X	X
Режим 2, макс: 375 КГц	12	1	X	X	X
Режим 1, 3: 62,2 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	0F4H
137,5 Гц	11,059	0	0	2	1DH
110 Гц	6	0	0	2	72H
110 Гц	12	0	0	1	0FEEBH

Таблица.5. Настройка таймера 1 для управления частотой работы приемопередатчика.

Отметим, что скорости приема и передачи могут различаться.

Предельно низких частот приемопередачи можно достичь при использовании таймера в режиме 1 (16-битный таймер) и разрешении прерываний от таймера (старший полубайт TMOD = 0001B). Перезагрузка 16-битного таймера должна осуществляться программным путем.

3.6. Система прерываний микроконтроллера 8051.

Упрощенная схема прерываний микро-ЭВМ 8051 показана на рисунке.

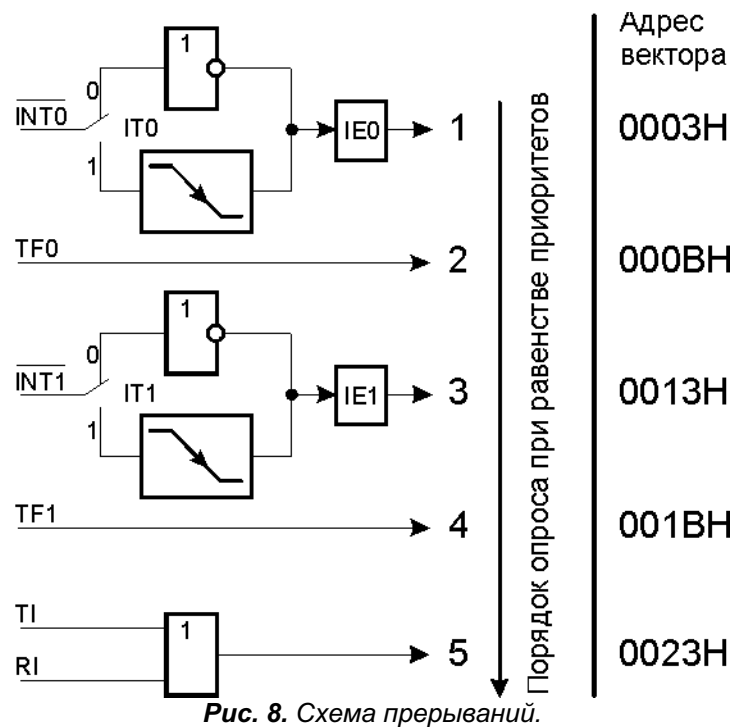


Рис. 8. Схема прерываний.

Внешние прерывания INT 0 и INT 1 могут быть вызваны либо уровнем, либо переходом сигнала из 1 в 0 на входах 8051 в зависимости от значений управляющих бит IT0 и IT1 в регистре TCON. От внешних прерываний устанавливаются флаги IE0 и IE1 в регистре TCON, которые инициируют вызов соответствующей программы 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 I должна управлять соответствующая подпрограмма обслуживания прерывания путем воздействия на источник прерывания с целью снятия им запроса.

Флаги запросов прерывания от таймеров TF0 и TF1 сбрасываются автоматически при передаче управления подпрограмме обслуживания. Флаги запросов прерывания RI и TI устанавливаются блоком управления приемопередатчика аппаратно, но сбрасываться должны программным путем.

Прерывания могут быть вызваны или отменены программой, так как все названные флаги программно доступны и могут быть установлены/ сброшены программой с тем же результатом, как если бы они были установлены/сброшены аппаратными средствами.

В блоке регистров специальных функций есть два регистра, предназначенных для управления режимом прерываний IE и уровнями приоритета IP. Возможность программной установки/сброса любого управляющего бита в этих двух регистрах делает систему прерываний 8051 исключительно гибкой.

В более сложных модификациях микроконтроллеров семейства MCS-51 количество периферийных устройств увеличено, что приводит к необходимости использовать один вектор прерывания для нескольких устройств (разделение подпрограмм обслуживания прерываний в этом случае необходимо реализовать программно), либо добавить еще два регистра - режима (маски) и приоритета прерываний.

3.6.1.Регистр масок прерывания (IE).

Символ	Позиция	Имя и назначение
EA	IE.7	Снятие блокировки прерывания. Сбрасывается, программно для запрета всех прерываний независимо от состояний IE.4 - IE.0
	IE.6	Не используется
	IE.5	Не используется
ES	IE.4	Бит разрешения прерывания, от приемопередатчика Установка/сброс программой для разрешения/запрета прерываний от флагов TI или RI .
ET1	IE.3	Бит разрешения прерывания от таймера. Установка/сброс программой для разрешения/запрета прерываний от таймера 1
EX1	IE.2	Бит разрешения внешнего прерывания 1. Установка/сброс программой для разрешения/запрета прерывания 1
ET0	IE.1	Бит разрешения прерывания от таймера 0. Установка/сброс программой для разрешения/запрета прерываний от таймера 0 .
EX0	IE.0	Бит разрешения внешнего прерывания 0. Установка/сброс программой для разрешения/запрета прерывания 0

3.6.2.Регистр приоритетов прерываний (IP).

Символ	Позиция	Имя и назначение
-	IP.7 - IP.5	Не используется
PS	IP.4	Бит приоритета приемопередатчика. Установка/сброс программой для присваивания прерыванию от приемопередатчика высшего/низшего приоритета
PT1	IP.3	Бит приоритета таймера 1. Установка/сброс программой для присваивания прерыванию от таймера 1 высшего/низшего приоритета
PX1	IP.2	Бит приоритета внешнего прерывания 1. Установка/сброс программой для присваивания высшего/низшего приоритета внешнему прерыванию INT1
PT0	IP.1	Бит приоритета таймера 0. Установка/сброс программой для присваивания прерыванию от таймера 0 высшего/низшего приоритета
PX0	IP.0	Бит приоритета внешнего прерывания 0. Установка/сброс программой для присваивания высшего/низшего приоритета внешнему прерыванию INT0

3.6.3.Выполнение подпрограммы прерывания.

Система прерываний формирует аппаратный вызов (LCALL) соответствующей подпрограммы обслуживания, если она не заблокирована одним из следующих условий:

- в данный момент обслуживается запрос прерывания равного или 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

Отметим, что если флаг прерывания был установлен, но по одному из указанных выше условий не получил обслуживания и к моменту окончания блокировки уже сброшен, то запрос прерывания теряется и нигде не запоминается.

По аппаратно сформированному коду LCALL система прерывания помещает в стек только содержимое счетчика команд (PC) и загружает в него адрес вектора соответствующей подпрограммы обслуживания. По адресу вектора должна быть расположена команда безусловной передачи управления (JMP) к начальному адресу подпрограммы обслуживания прерывания. В случае необходимости она должна начинаться командами записи в стек (PUSH) слова состояния программы (PSW), аккумулятора, расширителя, указателя данных и т.д. и должна заканчиваться командами восстановления из стека (POP). Подпрограммы обслуживания прерывания должны завершаться командой RETI, по которой в счетчик команд перезагружается из стека сохраненный адрес возврата в основную программу. Команда RET также возвращает управление прерванной основной программе, но при этом не снимают блокировку прерываний,

что приводит к необходимости иметь программный механизм анализа окончания процедуры обслуживания данного прерывания.

3.7. Работа с внешней памятью микроконтроллера 8051.

Обращения к внешней памяти подразделяются на обращения к внешней памяти программ и обращения к внешней памяти данных. В первом случае для формирования сигнала, активирующего ПЗУ с программой, используется сигнал PSEN, во втором — сигналы RD и WR, активизирующие ОЗУ с данными.

Если используется 16-битовый адрес, старшие восемь бит выводятся через порт P2, где они сохраняются в течение всего цикла обращения к внешней памяти. Отметим, что выходные каскады порта P2 имеют внутреннюю нагрузку, несколько отличающуюся от P1 и P3, благодаря чему в SFR P2 при выводе адресной информации вовсе не обязательно защелкивать все единицы. Добавим также, что при выводе адресной информации информация из SFR P2, хотя и не присутствует на выводах микроЭВМ, но и не теряется, восстанавливаясь на них после окончания обращений к внешней памяти (если в процессе этих обращений SFR P2 не был модифицирован).

Если при обращении к внешней памяти данных используется восьми битный адрес, то на выводах порта остается та же информация, которая там была до начала обращения к внешней памяти. Это позволяет организовать постраничную адресацию внешней памяти данных.

Как уже отмечалось, на выводах порта P0 младший байт адреса мультиплексируется с данными. Сигналы адреса/данных задействуют оба полевых транзистора выходного каскада порта P0. Таким образом, в этом случае выводы P0 уже не являются выводами с открытым стоком и не требуют внешних нагрузочных элементов.

Сигнал ALE используется для фиксации младшего байта адреса во внешнем регистре-защелке. Адресная информация достоверна в момент окончания сигнала ALE.

Выводимый в цикле записи байт заносится в P0 непосредственно перед активацией сигнала WR и остается неизменным до окончания этого сигнала. В цикле чтения данные на выводах P0 для достоверного считывания должны быть установившимися к моменту окончания сигнала RD.

Во время обращения к внешней памяти CPU записывает 0FFH в SFR P0, уничтожая, таким образом, хранимую там информацию. Таким образом, использовать для записи порт P0 при работе с внешней памятью надо с известной долей осторожности.

Обращение к внешней памяти программ возможно в двух случаях:

- когда сигнал EA активен, т.е. имеет нулевой уровень,
- когда программный счетчик PC содержит число больше 0FFH.

Следовательно, при использовании микро-ЭВМ, не имеющей встроенного ПЗУ или не использующей его, на входе EA должен присутствовать сигнал с нулевым уровнем.

Когда CPU работает с внешней памятью программ, все линии порта P2 используются для вывода старшего байта адреса и не могут быть использованы для обычного ввода\вывода информации. При этом, как отмечалось выше, в SFR P2 может быть занесена любая информация — адресная информация, выводимая через P2, не зависит от состояния его SFR.

3.8. Режимы микроконтроллера 8051 с пониженным энергопотреблением.

Во многих вариантах применения микро-ЭВМ энергопотребление является одним из основных параметров. В этих случаях целесообразно использовать КМОП версии микро-ЭВМ. В них предусмотрены дополнительные возможности снижения энергопотребления, отсутствующие в стандартных n-МОП изделиях. Ранее выпускались и n-МОП версии микро-ЭВМ, имевшие режимы работы с пониженным энергопотреблением. В настоящее время их выпуск почти повсеместно прекращен. Однако время от времени они все же будут попадать в руки радиолюбителей, поэтому трудно гарантировать наличие или отсутствие этих режимов в тех или иных конкретных n-МОП изделиях.

КМОП микро-ЭВМ имеют два режима с пониженным потреблением тока: режим холостого хода (XX) и режим выключенного напряжения питания (ВНП). Последний иногда называют режимом пониженного энергопотребления. По мнению авторов настоящей статьи, это вряд ли корректно, так как режим XX также характеризуется пониженным энергопотреблением в сравнении с обычным режимом работы микро-ЭВМ. Дословно второй режим, называемый в англоязычной литературе режимом "Power Down Mode", можно перевести так, как указано выше. Этот перевод корректнее отражает суть режима - в нем допустимо вообще отключить от микро-ЭВМ питающее напряжение, подаваемое на вход Ucc (вывод 40 микро-ЭВМ). для сохранения содержимого внутренней ОЗУ в этом случае нужно запитать от резервного источника. Резервное питающее напряжение подают через вход RST (вывод 9).

В режиме XX (IDL = 1) генератор микро-ЭВМ работает, тактовые сигналы поступают на систему прерываний, последовательный порт и таймеры/счетчики. Все регистры сохраняют свое содержимое, на выводах всех портов удерживаются логические состояния, которые были на них в момент перехода в режим XX. Однако сигнал генератора синхронизирующий CPU, отключается. В режиме ВНП (PD = 1) генератор останавливается. Прекращается тактирование не только CPU, но и последовательного порта, таймеров/счетчиков, системы прерываний. Как и в режиме XX, состояние регистров, резидентного ОЗУ и выводов портов остается неизменным.

Режимы ХХ и ВНП активизируются при установке соответствующих битов в специальном функциональном регистре-регистре управления мощностью PCON. Адрес этого регистра 87H, его описание приведено в табл.

Отметим следующие особенности этого регистра. Если одновременно установлены в 1 биты IDL и PD, то последний имеет преимущество - микро-ЭВМ переходит в режим ВНП. Содержимое регистра PCON после сброса - 0XXX0000.

В n-МОП версиях микроЭВМ регистр PCON содержит, как правило, только SMOD. Остальные четыре бита присутствуют только в КМОП устройствах. Пользовательские программы никогда не должны заносить 1 в незанятые биты (PCON4 - PCON6), так как они могут использоваться в следующих модификациях микроЭВМ. Последнее распространяется на все регистры и на все адреса в области регистров специальных функций, которые не заняты регистрами. Разработчики предполагают использовать их в новых изделиях. Однако для достижения совместимости уже разработанного программного обеспечения с новыми изделиями дополнительные возможности последних будут включаться установкой в 1 битов в соответствующих регистрах. Поэтому пользовательская программа, устанавливающая в 1 неиспользуемые биты, будет нормально работать на микро-ЭВМ, имеющихся в наличии сегодня, но вовсе не обязательно будет работать на новых микро-ЭВМ, несмотря на их полную программную совместимость с семейством 8051.

3.8.1.Режим ХХ.

В этот режим микро-ЭВМ переводится любой командой, устанавливающей в 1 бит PCON.0. Она оказывается последней в цепочке выполняемых команд: в режиме ХХ выполнение программы приостанавливается, так как на CPU перестает поступать сигнал тактового генератора. Однако содержимое внутреннего ОЗУ и регистров специальных функций остается неизменным, выходы портов удерживают значения, которые были на них до перехода в режим ХХ, на таймеры/счетчики, приемопередатчик и на систему прерываний продолжают поступать тактовые сигналы. На выводах ALE и PSEN устанавливаются сигналы единичного уровня.

Состояние выводов портов зависит от типа ОЗУ, с которым микро-ЭВМ обменивалась информацией перед тем как перейти в режим ХХ. При работе с внутренним ОЗУ на выводах портов присутствуют данные из соответствующих SFR (естественно, если порт в режиме вывода информации). При работе с внешним ОЗУ выходы порта 0 переходят в высокоимпедансное состояние, а на выводах порта 2. сохраняется адресная информация. На выводах портов 1 и 3 присутствуют данные из SFR портов.

Вывести микро-ЭВМ из режима ХХ можно двумя способами. Так, вызов любого из прерываний приведет к аппаратному стиранию бита PCON.0, прекращающему ХХ. Прерывание будет обслужено, и очередной после RETI выполняемой командой будет та, которая следует за командой, приведшей к переходу микро-ЭВМ в режим ХХ. флаги GF0 и GF1 могут использоваться для индикации того, произошло ли прерывание во время нормальной работы или во время ХХ. Например, команда, запускающая этот режим, может также устанавливать один или оба флага. Когда ХХ прекращен прерыванием, сервисная программа прерывания может проверять состояние флагов.

Другой способ прекращения ХХ - с помощью аппаратного сброса. Поскольку синхрогенератор продолжает работать, аппаратный сброс должен поддерживаться в активном состоянии, только в течение двух машинных циклов (24 периодов колебаний).

Сигнал сброса стирает бит PCON.0. В этот момент CPU возобновляет выполнение программы с самого начала. Как показано на рис. 14, перед началом обработки алгоритма внутреннего запуска могут иметь место два или три машинных цикла выполнения программы. Встроенное в микросхему устройство в это время препятствует доступу к внутреннему ОЗУ, но доступ к выводам порта не ограничен. Чтобы исключить возможность появления неопределенных выходных сигналов на выводах порта, команда, следующая за вызывающей ХХ, не должна быть командой, записывающей информацию в SFR порта или во внешнее ОЗУ данных. Напомним, что после аппаратного сброса содержимое SFR переопределяется.

3.8.2.Режим ВНП.

Команда, устанавливающая в 1 бит PCON.1, переводит микро-ЭВМ в режим ВНП. В нем генератор микро-ЭВМ, как отмечалось, останавливается, прекращает функционирование не только CPU, но и таймеры/счетчики, приемопередатчик, система прерываний. При наличии основного или резервного источника питающего напряжения встроенное ОЗУ и регистры SFR сохраняют свое содержимое. Состояние портов не отличается от состояния при переходе в режим ХХ (см. выше). Однако в отличие от режима ХХ на выводах ALE и PSEN устанавливаются сигналы с нулевым уровнем.

Единственный способ выйти из этого режима - аппаратный сброс. Он переопределяет содержимое всех SFR, но не меняет содержимого встроенного ОЗУ.

В рассматриваемом режиме напряжение питания Ucc может снижаться вплоть до 2 В. Следует, однако, позаботиться, чтобы снижалось оно не раньше перехода микроЭВМ в режим ВНП и восстанавливалось до прежнего значения прежде, чем она выйдет из этого режима. Сигнал сброса, возвращающий микро-ЭВМ в обычный режим, не должен подаваться раньше, чем Ucc достигнет своего рабочего значения, и должен поддерживаться в активном состоянии достаточно долго, чтобы генератор успел запуститься и его колебания стабилизировались (обычно - не менее 10 мс).

4. Система команд микроконтроллера семейства 8051.

4.1.1. Общая характеристика.

Микро-ЭВМ 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). Микро-ЭВМ семейства 8051 используют прямую, непосредственную, косвенную и неявную, адресацию данных

В качестве операндов команд микро-ЭВМ 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 микро ЭВМ это возможность адресации отдельных бит в резидентной памяти данных. Кроме того, как отмечалось, некоторые регистры блока регистров специальных функций также допускают адресацию отдельных бит. Карты адресов отдельных бит в резидентной памяти данных и в блоке регистров специальных функций.

4.1.2. Типы команд

Всего микро-ЭВМ 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 команды	Первый байт D7...D0	Второй байт D7...D0	Третий байт 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

Таблица. 6. Типы команд

4.1.3. Типы операндов

Состав операндов включает в себя операнды четырех типов: биты, 4-битные цифры, байты и 16-битные слова.

Микроконтроллер имеет 128 программно-управляемых флагов пользователя. Имеется также возможность адресации отдельных битов блока регистров специальных функций и портов. Для адресации битов используется прямой 8-битный адрес (bit). Косвенная адресация битов невозможна. Карты адресов отдельных битов представлены на рис. 6 .

Четырехбитные операнды используются только при операциях обмена SWAP и XCHD.

Восьмибитным операндом может быть ячейка памяти программ (ПП) или данных (резидентной (РПД) или внешней (ВПД)), константа (непосредственный операнд), регистры специальных функций, а также порты ввода/вывода. Порты и регистры специальных функций адресуются только прямым способом. Байты памяти могут адресоваться также и косвенным образом через адресные регистры R0, R1, DPTR и PC.

Двухбайтные операнды - это константы и прямые адреса, для представления которых используются второй и третий байты команды.

Адреса	(D ₇)								(D ₀)
7FH									
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	09	08	
20H	07	06	05	04	03	02	01	00	
1FH	Банк 3								
18H									
17H	Банк 2								
10H									
0FH	Банк 1								
08H									
07H	Банк 0								
00H									

Рис. 9. Карта адресуемых битов в резидентной памяти данных

4.1.4. Группы команд.

Система команд семейства MCS-51 содержит 111 базовых команд, которые по функциональному признаку можно подразделить на пять:

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

Формат команд - одно-, двух- и трехбайтовый, причем большинство команд (94) имеют формат один или два байта. Первый байт любых типа и формата всегда содержит код операции, второй и третий байты содержат либо адреса операндов, либо непосредственные операнды.

Состав операндов включает в себя операнды четырех типов: биты, nibbles (4 разряда), байты и 16-битные слова. Время исполнения команд составляет 1, 2 или 4 машинных цикла. При тактовой частоте 12 мГц длительность машинного цикла составляет 1 мкс, при этом 64 команды исполняются за 1 мкс, 45 команд - за 2 мкс и 2 команды (умножение и деление) - за 4 мкс.

Набор команд MCS-51 поддерживает следующие режимы адресации.

Прямая адресация (Direct Addressing). Операнд определяется 8-битным адресом в инструкции. Эта адресация используется только для внутренней памяти данных и регистров SFR.

Косвенная адресация (Indirect Addressing). В этом случае инструкция адресует регистр, содержащий адрес операнда. Данный вид адресации может применяться при обращении как к внутреннему, так и внешнему ОЗУ. Для указания 8-битных адресов могут использоваться регистры R0 и R1 выбранного регистрового банка или указатель стека SP.

Для 16-битной адресации используется только регистр "указатель данных" (DPTR - Data Pointer).

Регистровая адресация (Register Instruction). Данная адресация применяется для доступа к регистрам R0+R7 выбранного банка. Команды с регистровой адресацией содержат в байте кода операции трехбитовое поле, определяющее номер регистра. Выбор одного из четырех регистровых банков осуществляется программированием битов селектора банка (RS1, RS0) в PSW.

Непосредственная адресация (Immediate constants). Операнд содержится непосредственно в поле команды вслед за кодом операции и может занимать один или два байта ($data_8$, $data_{16}$).

Индексная адресация (Indexed Addressing). Индексная адресация используется при обращении к памяти программ и только при чтении. В этом режиме осуществляется просмотр таблиц в памяти программ. 16-битовый регистр (DPTR или PC) указывает базовый адрес требуемой таблицы, а аккумулятор указывает на точку входа в нее. Адрес элемента таблицы находится сложением базы с индексом (содержимым аккумулятора).

Другой тип индексной адресации применяется в командах "перехода по выбору" (Case Jump). При этом адрес перехода вычисляется как сумма указателя базы и аккумулятора.

Неявная адресация (Register-Specific Instructions). Некоторые инструкции используют индивидуальные регистры (например, операции с аккумулятором, DPTR), при этом данные регистры не имеют адреса, указывающего на них; это заложено в код операции

4.1.5. Обозначения, используемые при описании команд.

Rn (n = 0, 1, ..., 7) – регистр общего назначения в выбранном банке регистров;

@Ri (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,

4.1.6. 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 (Б) и время выполнения в машинных циклах (Ц).

Таблица.7. Команды передачи данных

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Пересылка в аккумулятор из регистра (n=0÷7)	MOV A, Rn	11101rrr	1	1	1	(A) ← (Rn)
Пересылка в аккумулятор прямоадресуемого байта	MOV A, ad	11100101	3	2	1	(A) ← (ad)
Пересылка в аккумулятор байта из РПД (i=0,1)	MOV A, @Ri	1110011i	1	1	1	(A) ← ((Ri))
Загрузка в аккумулятор константы	MOV A, #d	01110100	2	2	1	(A) ← #d
Пересылка в регистр из аккумулятора	MOV Rn, A	11111rrr	1	1	1	(Rn) ← (A)
Пересылка в регистр прямоадресуемого байта	MOV Rn, ad	10101rrr	3	2	2	(Rn) ← (ad)
Загрузка в регистр константы	MOV Rn, #d	01111rrr	2	2	1	(Rn) ← #d
Пересылка по прямому адресу аккумулятора	MOV ad, A	11110101	3	2	1	(ad) ← (A)
Пересылка по прямому адресу регистра	MOV ad, Rn	10001rrr	3	2	2	(ad) ← (Rn)
Пересылка прямоадресуемого байта по прямому адресу	MOV add, ads	10000101	9	3	2	(add) ← (ads)
Пересылка байта из РПД по прямому адресу	MOV ad, @Ri	1000011i	3	2	2	(ad) ← ((Ri))
Пересылка по прямому адресу константы	MOV ad, #d	01110101	7	3	2	(ad) ← #d
Пересылка в РПД из аккумулятора	MOV @Ri, A	1111011i	1	1	1	((Ri)) ← (A)
Пересылка в РПД прямоадресуемого байта	MOV @Ri, ad	0110011i	3	2	2	((Ri)) ← (ad)
Пересылка в РПД константы	MOV @Ri, #d	0111011i	2	2	1	((Ri)) ← #d
Загрузка указателя данных	MOV DPTR, #d16	10010000	13	3	2	(DPTR) ← #d16
Пересылка в аккумулятор байта из ПП	MOVC A, @A+DPTR	10010011	1	1	2	← ((A) +(DPTR))
Пересылка в аккумулятор байта из ПП	MOVC A, @A+PC	10000011	1	1	2	(PC) ← (PC)+1, (A) ← ((A)+(PC))

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Пересылка в аккумулятор байта из ВПД	MOVX A, @Ri	1110001i	1	1	2	(A) ← ((Ri))
Пересылка в аккумулятор байта из расширенной ВПД	MOVX A, @DPTR	11100000	1	1	2	(A) ← ((DPTR))
Пересылка в ВПД из аккумулятора	MOVX @Ri, A	1111001i	1	1	2	((Ri)) ← (A)
Пересылка в расширенную ВПД из аккумулятора	MOVX @DPTR, A	11110000	1	1	2	((DPTR)) ← (A)
Загрузка в стек	PUSH ad	11000000	3	2	2	(SP) ← (SP) + 1, ((SP)) ← (ad)
Извлечение из стека	POP ad	11010000	3	2	2	(ad) ← (SP), (SP) ← (SP) - 1
Обмен аккумулятора с регистром	XCH A, Rn	11001rrr	1	1	1	(A) ↔ (Rn)
Обмен аккумулятора с прямоадресуемым байтом	XCH A, ad	11000101	3	2	1	(A) ↔ (ad)
Обмен аккумулятора с байтом из РПД	XCH A, @Ri	1100011i	1	1	1	(A) ↔ ((Ri))
Обмен младших тетрад аккумулятора и байта РПД	XCHD A, @Ri	1101011i	1	1	1	(A _{0...3}) ↔ ((Ri) _{0...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 микро-ЭВМ 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 микро-ЭВМ 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.

4.1.7. Команды арифметических операций 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 (Б) и время выполнения в машинных циклах (Ц).

Таблица.8. Арифметические операции.

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Сложение аккумулятора с регистром (n=0÷7)	ADD A, Rn	00101rrr	1	1	1	(A) ← (A) + (Rn)
Сложение аккумулятора с прямоадресуемым байтом	ADD A, ad	00100101	3	2	1	(A) ← (A) + (ad)
Сложение аккумулятора с байтом из РПД (i = 0, 1)	ADD A, @Ri	0010011i	1	1	1	(A) ← (A) + ((Ri))
Сложение аккумулятора с константой	ADD A, #d	00100100	2	2	1	(A) ← (A) + #d
Сложение аккумулятора с регистром и переносом	ADDC A, Rn	00111rrr	1	1	1	(A) ← (A) + (Rn) + (C)
Сложение аккумулятора с прямоадресуемым байтом и переносом	ADDC A, ad	00110101	3	2	1	(A) ← (A) + (ad) + (C)
Сложение аккумулятора с байтом из РПД и переносом	ADDC A, @Ri	0011011i	1	1	1	(A) ← (A) + ((Ri)) + (C)
Сложение аккумулятора с константой и переносом	ADDC A, #d	00110100	2	2	1	(A) ← (A) + #d + (C)
Десятичная коррекция аккумулятора	DA A	11010100	1	1	1	Если (A _{0...3}) > 9 или ((AC)=1), то (A _{0...3}) ← (A _{0...3}) + 6, затем если (A _{4...7}) > 9 или ((C)=1), то (A _{4...7}) ← (A _{4...7}) + 6
Вычитание из аккумулятора регистра и заёма	SUBB A, Rn	10011rrr	1	1	1	(A) ← (A) - (C) - (Rn)
Вычитание из аккумулятора прямоадресуемого байта и заема	SUBB A, ad	10010101	3	2	1	(A) ← (A) - (C) - ((ad))
Вычитание из аккумулятора байта РПД и заема	SUBB A, @Ri	1001011i	1	1	1	(A) ← (A) - (C) - ((Ri))
Вычитание из аккумулятора константы и заема	SUBB A, d	10010100	2	2	1	(A) ← (A) - (C) - #d
Инкремент аккумулятора	INC A	00000100	1	1	1	(A) ← (A) + 1

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Инкремент регистра	INC Rn	00001rrr	1	1	1	$(Rn) \leftarrow (Rn) + 1$
Инкремент прямоадресуемого байта	INC ad	00000101	3	2	1	$(ad) \leftarrow (ad) + 1$
Инкремент байта в РПД	INC @Ri	0000011i	1	1	1	$((Ri)) \leftarrow ((Ri)) + 1$
Инкремент указателя данных	INC DPTR	10100011	1	1	2	$(DPTR) \leftarrow (DPTR) + 1$
Декремент аккумулятора	DEC A	00010100	1	1	1	$(A) \leftarrow (A) - 1$
Декремент регистра	DEC Rn	00011rrr	1	1	1	$(Rn) \leftarrow (Rn) - 1$
Декремент прямоадресуемого байта	DEC ad	00010101	3	2	1	$(ad) \leftarrow (ad) - 1$
Декремент байта в РПД	DEC @Ri	0001011i	1	1	1	$((Ri)) \leftarrow ((Ri)) - 1$
Умножение аккумулятора на регистр B	MUL AB	10100100	1	1	4	$(B)(A) \leftarrow (A) \cdot (B)$
Деление аккумулятора на регистр B	DIV AB	10000100	1	1	4	$(B).(A) \leftarrow (A)/(B)$

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 $P = 0$.

4.1.8. 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 (\wedge), логическое ИЛИ (\vee), исключающее ИЛИ (\oplus), инверсию (NOT), сброс в нулевое значение и сдвиг. В таблице также указаны тип команды (Т) в соответствии с таблицей, ее длина в байтах (Б) и время выполнения в машинных циклах (Ц).

Таблица.9. Логические операции

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Логическое И аккумулятора и регистра	ANL A, Rn	01011rrr	1	1	1	$(A) \leftarrow (A) \text{ AND } (Rn)$
Логическое И аккумулятора и прямоадресуемого байта	ANL A, ad	01010101	3	2	1	$(A) \leftarrow (A) \text{ AND } (ad)$
Логическое И аккумулятора и байта из РПД	ANL A, @Ri	0101011i	1	1	1	$(A) \leftarrow (A) \text{ AND } ((Ri))$
Логическое И аккумулятора и константы	ANL A, #d	01010100	2	2	1	$(A) \leftarrow (A) \text{ AND } \#d$
Логическое И прямоадресуемого байта и аккумулятора	ANL ad, A	01010010	3	2	1	$(ad) \leftarrow (ad) \text{ AND } (A)$
Логическое И прямоадресуемого байта и константы	ANL ad, #d	01010011	7	3	2	$(ad) \leftarrow (ad) \text{ AND } \#d$
Логическое ИЛИ аккумулятора и регистра	ORL A, Rn	01001rrr	1	1	1	$(A) \leftarrow (A) \text{ OR } (Rn)$
Логическое ИЛИ аккумулятора и прямоадресуемого байта	ORL A, ad	01000101	3	2	1	$(A) \leftarrow (A) \text{ OR } (ad)$
Логическое ИЛИ аккумулятора и байта из РПД	ORL A, @Ri	0100011i	1	1	1	$(A) \leftarrow (A) \text{ OR } ((Ri))$
Логическое ИЛИ аккумулятора и константы	ORL A, #d	01000100	2	2	1	$(A) \leftarrow (A) \text{ OR } \#d$
Логическое ИЛИ прямоадресуемого байта и аккумулятора	ORL ad, A	01000010	3	2	1	$(ad) \leftarrow (ad) \text{ OR } (A)$
Логическое ИЛИ прямоадресуемого байта и константы	ORL ad, #d	01000011	7	3	2	$(ad) \leftarrow (ad) \text{ OR } \#d$
Исключающее ИЛИ аккумулятора и регистра	XRL A, Rn	01101rrr	1	1	1	$(A) \leftarrow (A) \text{ XOR } (Rn)$
Исключающее ИЛИ аккумулятора и прямоадресуемого байта	XRL A, ad	01100101	3	2	1	$(A) \leftarrow (A) \text{ XOR } (ad)$
Исключающее ИЛИ аккумулятора и байта из РПД	XRL A, @Ri	0110011i	1	1	1	$(A) \leftarrow (A) \text{ XOR } ((Ri))$
Исключающее ИЛИ аккумулятора и константы	XRL A, #d	01100100	2	2	1	$(A) \leftarrow (A) \text{ XOR } \#d$
Исключающее ИЛИ прямоадресуемого байта и аккумулятора	XRL ad, A	01100010	3	2	1	$(ad) \leftarrow (ad) \text{ XOR } (A)$
Исключающее ИЛИ прямоадресуемого байта и константы	XRL ad, #d	01100011	7	3	2	$(ad) \leftarrow (ad) \text{ XOR } \#d$
Сброс аккумулятора	CLR A	11100100	1	1	1	$(A) \leftarrow 0$
Инверсия аккумулятора	CPL A	11110100	1	1	1	$(A) \leftarrow \text{NOT}(A)$
Сдвиг аккумулятора влево циклический	RL A	00100011	1	1	1	$(A_{n+1}) \leftarrow (A_n), n=0 \div 6, (A_0) \leftarrow (A_7)$
Сдвиг аккумулятора влево через перенос	RLC A	00110011	1	1	1	$(A_{n+1}) \leftarrow (A_n), n=0 \div 6, (A_0) \leftarrow (C), (C) \leftarrow (A_7)$
Сдвиг аккумулятора вправо циклический	RR A	00000011	1	1	1	$(A_n) \leftarrow (A_{n+1}), n=0 \div 6, (A_7) \leftarrow (A_0)$
Сдвиг аккумулятора вправо через перенос	RRC A	00010011	1	1	1	$(A_n) \leftarrow (A_{n+1}), n=0 \div 6, (A_7) \leftarrow (C), (C) \leftarrow (A_0)$
Обмен местами тетрад в аккумуляторе	SWAP A	11000100	1	1	1	$(A_{0...3}) \leftrightarrow (A_{4...7})$

4.1.9. 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 (\wedge) и ИЛИ (\vee). 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 (Б) и время выполнения в машинных циклах (Ц).

Таблица.10. Операции с битами

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Сброс переноса	CLR C	11000011	1	1	1	(C) ← 0
Сброс бита	CLR bit	11000010	4	2	1	(b) ← 0
Установка переноса	SETB C	11010011	1	1	1	(C) ← 1
Установка бита	SETB bit	11010010	4	2	1	(b) ← 1
Инверсия переноса	CPL C	10110011	1	1	1	(C) ← NOT(C)
Инверсия бита	CPL bit	10110010	4	2	1	(b) ← NOT(b)
Логическое И бита и переноса	ANL C, bit	10000010	4	2	2	(C) ← (C) AND (b)
Логическое И инверсии бита и переноса	ANL C, /bit	10110000	4	2	2	(C) ← (C) AND (NOT(b))
Логическое ИЛИ бита и переноса	ORL C, bit	01110010	4	2	2	(C) ← (C) OR (b)
Логическое ИЛИ инверсии бита и переноса	ORL C, /bit	10100000	4	2	2	(C) ← (C) OR (NOT(b))
Пересылка бита в перенос	MOV C, bit	10100010	4	2	1	(C) ← (b)
Пересылка переноса в бит	MOV bit, C	10010010	4	2	2	(b) ← (C)

4.1.10. Команды передачи управления микроконтроллера 8051.

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 (Б) и время выполнения в машинных циклах (Ц).

Таблица.11. Команды передачи управления

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Длинный переход в полном объеме ПП	LJMP ad16	00000010	12	3	2	(PC) ← ad16
Абсолютный переход внутри страницы в 2 Кб	AJMP ad11	a ₁₀ a ₉ a ₈ 00001	6	2	2	(PC) ← (PC) + 2, (PC ₀₋₁₀) ← ad11
Короткий относительный переход внутри страницы в 256 байт	SJMP rel	10000000	5	2	2	(PC) ← (PC) + 2, (PC) ← (PC) + rel
Косвенный относительный переход	JMP @A+DPTR	01110011	1	1	2	(PC) ← (A) + (DPTR)
Переход, если аккумулятор равен нулю	JZ rel	01100000	5	2	2	(PC) ← (PC) + 2, если (A)=0, то (PC) ← (PC) + rel
Переход, если аккумулятор не равен нулю	JNZ rel	01110000	5	2	2	(PC) ← (PC) + 2, если (A) ≠ 0, то (PC) ← (PC) + rel
Переход, если перенос равен единице	JC rel	01000000	5	2	2	(PC) ← (PC) + 2, если (C)=1, то (PC) ← (PC) + rel
Переход, если перенос равен нулю	JNC rel	01010000	5	2	2	(PC) ← (PC) + 2, если (C)=0, то (PC) ← (PC) + rel
Переход, если бит равен единице	JB bit, rel	00100000	11	3	2	(PC) ← (PC) + 3, если (b)=1, то (PC) ← (PC) + rel
Переход, если бит равен нулю	JNB bit, rel	00110000	11	3	2	(PC) ← (PC) + 3, если (b)=0, то (PC) ← (PC) + rel
Переход, если бит установлен, с последующим сбросом бита	JBC bit, rel	00010000	11	3	2	(PC) ← (PC) + 3, если (b)=1, то (b) ← 0 и (PC) ← (PC) + rel
Декремент регистра и переход, если не нуль	DJNZ Rn, rel	11011rrr	5	2	2	(PC) ← (PC) + 2, (Rn) ← (Rn) - 1, если (Rn) ≠ 0, то (PC) ← (PC) + rel
Декремент прямоадресуемого байта и переход, если не нуль	DJNZ ad, rel	11010101	8	3	2	(PC) ← (PC) + 2, (ad) ← (ad) - 1, если (ad) ≠ 0, то (PC) ← (PC) + rel
Сравнение аккумулятора с прямоадресуемым байтом и переход, если не равно	CJNE A, ad, rel	10110101	8	3	2	(PC) ← (PC) + 3, если (A) ≠ (ad), то (PC) ← (PC) + rel, если (A) < (ad), то (C) ← 1, иначе (C) ← 0
Сравнение аккумулятора с константой и переход, если не равно	CJNE A, #d, rel	10110100	10	3	2	(PC) ← (PC) + 3, если (A) ≠ #d, то (PC) ← (PC) + rel, если (A) < #d, то (C) ← 1, иначе (C) ← 0
Сравнение регистра с константой и переход, если не равно	CJNE Rn, #d, rel	10111rrr	10	3	2	(PC) ← (PC) + 3, если (Rn) ≠ #d, то (PC) ← (PC) + rel, если (Rn) < #d, то (C) ← 1, иначе (C) ← 0
Сравнение байта в РГД с константой и	CJNE @Ri, #d, rel	1011011i	10	3	2	(PC) ← (PC) + 3, если ((Ri)) ≠ #d, то

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
переход, если не равно						$(PC) \leftarrow (PC) + rel, \text{если } ((Ri)) < \#d, \text{ то } (C) \leftarrow 1, \text{ иначе } (C) \leftarrow 0$
Длинный вызов подпрограммы	LCALL ad16	00010010	12	3	2	$(PC) \leftarrow (PC) + 3, (SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (PC_{0..7}), (SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (PC_{8..15}), (PC) \leftarrow ad16$
Абсолютный вызов подпрограммы в пределах страницы в 2 Кб	ACALL ad11	a ₁₀ a ₉ a ₈ 10001	6	2	2	$(PC) \leftarrow (PC) + 2, (SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (PC_{0..7}), (SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (PC_{8..15}), (PC_{0-10}) \leftarrow ad11$
Возврат из подпрограммы	RET	00100010	1	1	2	$(PC_{8..15}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1, (PC_{0..7}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$
Возврат из подпрограммы обработки прерывания	RETI	00110010	1	1	2	$(PC_{8..15}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1, (PC_{0..7}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$
Пустая операция	NOP	00000000	1	1	1	$(PC) \leftarrow (PC) + 1$

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:

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

Все команды условного перехода рассматриваемых микро-ЭВМ 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 обслуживаемого 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.