

АРХИТЕКТУРА MCS-51

2.1. Особенности архитектуры MCS-51

Архитектура MCS-51 фирмы Intel была в свое время определена настолько удачно, что является сегодня, по существу, одним из стандартов «де-факто» на мировом рынке 8-разрядных микроконтроллеров. Эту архитектуру воспроизводит в том или ином виде в своих изделиях ряд фирм. По совокупному объему производства этих фирм микроконтроллеры MCS-51 занимают первое место.

Понятие «архитектура» микроконтроллеров далее трактуется как совокупность внутренних и внешних программно доступных ресурсов, системы команд, системы прерываний, функций ввода/вывода и протоколов обмена по магистрали. Архитектура воплощается производителем в виде набора связанных функционально-топологических модулей. Конкретный микроконтроллер представляет собой определенную комбинацию этих модулей, основой которой является операционное ядро («core» у фирмы Intel).

Исходная архитектура MCS-51 характеризуется следующими особенностями:

- архитектура «гарвардская», т.е. память программ и данных разделены;
- операционное ядро имеет «аккумуляторную» организацию, т.е. результат операции, как правило, помещается в регистр-аккумулятор;
- 8-разрядное АЛУ с аппаратным умножителем обрабатывает целочисленные операнды;

- имеются внутренние память программ (4Кбайт) и ОЗУ данных (128 байт);
- имеются четыре универсальных программируемых параллельных 8-разрядных порта ввода/вывода с возможностью реализации определенных альтернативных функций;
- набор блоков ввода/вывода включает два 16-разрядных программируемых счетчика/таймера и дуплексный последовательный порт;
- все регистры управления блоками ввода/вывода сведены в группу регистров спецфункций, которая имеет свой диапазон адресов в пространстве внутренней памяти данных.

Первая линия (product line) микроконтроллеров с исходной архитектурой MCS-51 была разработана на основе nMOS технологии (NMOS). Эта линия включала микросхемы 8051АН/8751ВН/8031АН с постоянной памятью программ, с перезаписываемой памятью программ (стираемой ультрафиолетом) и без внутренней памяти программ. Отечественной промышленностью до сих пор выпускается аналог микросхемы 8031АН – микроконтроллер К1816ВЕ31. Мощность, потребляемая микроконтроллерами этой группы, довольно велика.

Современные версии микроконтроллеров с исходной архитектурой MCS-51 выполнены с использованием комплементарной МОП технологии (CMOS). Они имеют маркировку 80C51/31 и выпускаются несколькими фирмами, поскольку дешевы и позволяют решать многие задачи. Технология CMOS позволила уменьшить рассеиваемую мощность при обычной работе и ввести особые режимы, дополнительные снижающие энергопотребление. Отечественным аналогом микросхемы 80C31 является микроконтроллер К1830ВЕ31.

В последние годы бурно развивается технология перезаписываемой flash-памяти программ, которая в значительной степени вытеснила память, стираемую ультрафиолетом. Это объясняется дороговизной металлокерамического корпуса с кварцевым стеклом, необходимого для УФ-микросхем. Микроконтроллеры с исходной архитектурой MCS-51 и flash-памятью выпускаются фирмами Atmel, Philips и другими под маркировкой 89C51. Микроконтроллеры с УФ-памятью остались, в основном, в виде однократно программируемых (ОТР) версий в пластмассовом корпусе без окна. Они имеют маркировку 87C51ОТР.

Усовершенствование технологии, рост степени интеграции позволил в рамках архитектуры MCS-51 расширить набор внутренних интерфейсных блоков, увеличить внутреннюю память программ и данных. Таким образом появились микроконтроллеры типа 80C52/54/58 и микроконтроллеры типа 80L52/54/58 с пониженным напряжением питания. Модернизированная архитектура MCS-51 характеризуется увеличенным до 256 байт внутренним ОЗУ данных, внутренней памятью программ 8/16/32 Кбайт,

наличием сторожевого таймера (watchdog timer). Расширение набора внутренних блоков привело к увеличению числа регистров спецфункций, несколько изменилась система прерываний. Наиболее широко используемыми микроконтроллерами с модернизированной MCS-51 архитектурой можно, в настоящее время, считать микросхемы типа 87C52/89C52/80C32, которые выпускаются большинством производителей, поддерживающих архитектуру MCS-51.

Такие фирмы, как Intel, Philips, Siemens, Dallas Semiconductor, Atmel, Winbond и ряд других, производят семейства микроконтроллеров с архитектурой MCS-51. В целях повышения производительности некоторые из них используют такие отличия, как расширенная система команд, сжатый цикл обмена по магистрали, увеличенное количество портов ввода/вывода.

В таблице указаны характеристики некоторых представителей семейств микроконтроллеров с архитектурой MCS-51.

2.2. Структура микроконтроллеров MCS-51

На рис. 2.1 приведена внутренняя структура микроконтроллера Intel 8051, соответствующая исходной архитектуре MCS-51.

Она включает следующий набор функциональных модулей:

- 8-разрядное АЛУ с аппаратной реализацией операций типа умножение;
- внутренняя память программ (4Кбайт) и ОЗУ данных (128 байт);
- четыре универсальных параллельных 8-разрядных порта ввода/вывода с возможностью реализации определенных альтернативных функций;
- два 16-разрядных программируемых счетчика/таймера;
- дуплексный последовательный порт.

Этот набор аппаратных средств и совокупность реализуемых функций делают микроконтроллеры семейства 8051 эффективным средством сбора, предобработки информации и управления объектами.

Внешний вид микросхем MCS-51 в 40-выводном корпусе и функции выводов представлены на рис. 2.2.

Функции выводов микроконтроллеров MCS-51 следующие.

Port 0. Двухнаправленный программируемый 8-разрядный параллельный порт ввода/вывода с возможностью установки в высокоимпедансное состояние. При работе в качестве выходов каждая линия обеспечивает нагрузочную способность, равную 8 входам маломощной серии LS TTL.

Микроконтроллеры с архитектурой MCS-51

Микро-контроллер (производитель)	Память программ	RAM	Таймеры	АЦП/ ЦАП	Fosc (МГц)	Особенности
8031АН Intel (снят с произ-водства)	-	128	2	-	12	Исходная архитектура, нМОП
K1816BE31 Россия	-	128	2	-	12	нМОП
80C31 Intel	-	128	2	-	12	КМОП
K1830BE31 Россия	-	128	2	-	12	КМОП
87C51 Intel AT87F51 Atmel AT89C51 Atmel	4K EPROM OTP 4K Flash	128	2	-	12, 16, 24, 33	3 бита защиты Flash с перезаписью
80C32 Intel	-	256	3	-	12, 16, 24, 33	КМОП
87C52 Intel	8K EPROM OTP	256	3	-	12, 16, 24, 33	КМОП
AT89C52 Atmel	8K Flash	256	3	-	12, 16, 24, 33	КМОП
80C52 Intel	8K ROM	256	3	-	12, 16, 20, 24	КМОП
87C54 Intel	16K EPROM OTP	256	3	-	12, 16, 20, 24	КМОП
80C54 Intel	16K ROM	256	3	-	12, 16, 24, 33	КМОП
87C58 Intel	32K EPROM OTP	256	3	-	12, 16, 24, 33	КМОП
80C58 Intel	32K ROM	256	3	-	12, 16, 24, 33	КМОП
87L52/54/58 Intel	8/16/32K OTP	256	3	-	12, 16, 20	Низковольтная серия (Vcc=3В), КМОП
80L52/54/58 Intel	8/16/32K ROM	256	3	-	12, 16, 20	
AduC812 Analog Devices	8K Flash	256	3	1/2	16	12-разрядные ЦАП и 2 АЦП, Vcc=(3,0-5,0)В

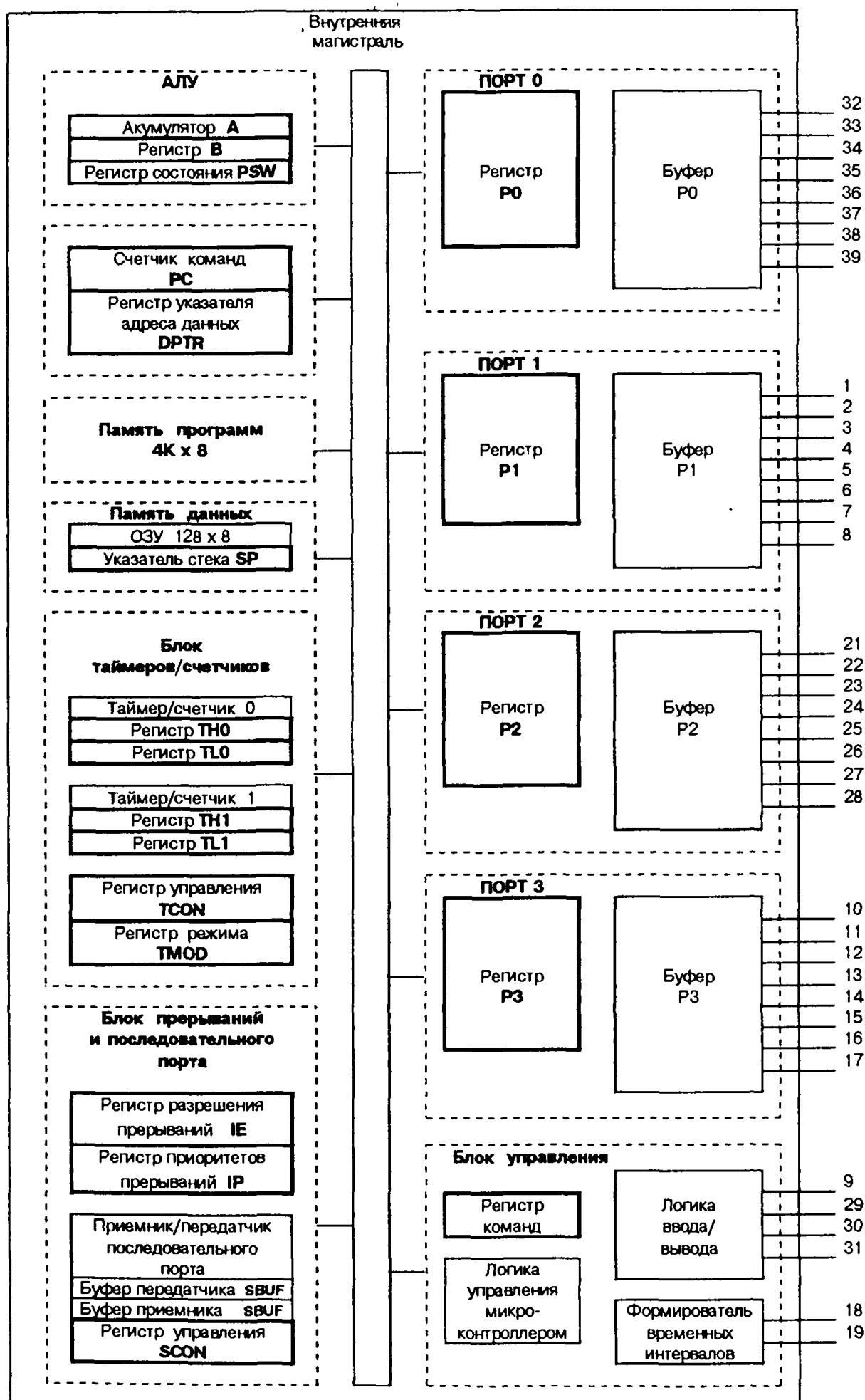


Рис. 2.1. Структура микроконтроллеров MCS-51

При обращении микроконтроллера к внешней памяти программ или данных порт работает в режиме мультиплексирования младшего байта адреса и 8-разрядной шины данных.

При записи в разряд регистра порта P0 логической «1» соответствующая линия порта переходит в режим высокоимпедансного входа. Для работы в режиме порта ввода/вывода необходимо внешнее подтягивание каждой линии порта к уровню логической «1».

Port 1. Двухнаправленный 8-разрядный параллельный порт ввода/вывода. При работе в качестве выходов каждая линия обеспечивает нагрузочную способность, равную 4 входам маломощной серии LS TTL. При записи в разряд регистра порта P1 логической «1» соответствующая линия порта переходит в режим высокоимпедансного входа со слабым подтягиванием сигнала к уровню логической «1».

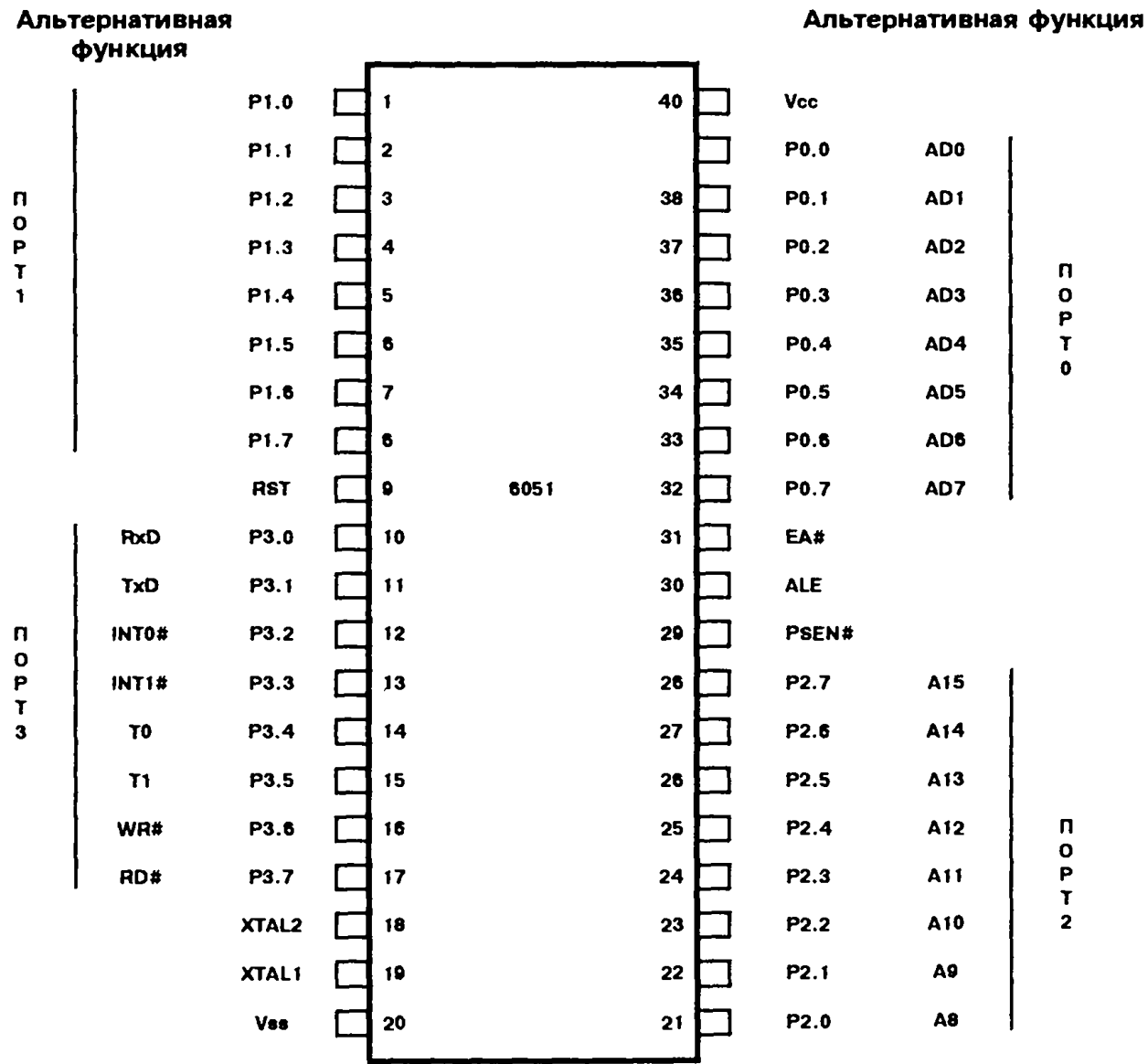


Рис. 2.2. Микроконтроллер MCS-51 в 40-выводном корпусе и функции выводов

Port 2. Двухнаправленный 8-разрядный параллельный порт ввода/вывода. При работе в качестве выходов каждая линия обеспечивает нагрузочную способность, равную 4 входам маломощной серии LS TTL. При записи в разряд регистра порта P2 логической «1» соответствующая линия порта переходит в режим высокоимпедансного входа со слабым подтягиванием сигнала к уровню логической «1».

При обращении микроконтроллера к внешней памяти программ или данных с использованием 16-разрядного адреса (команды MOVX @DPTR) через порт P2 выдается старший байт адреса. При обращении микроконтроллера к внешней памяти данных с использованием 8-разрядного адреса (команды MOVX @Ri) на выходы порта P2 выдается содержимое регистра специальных функций P2.

Port 3. Двухнаправленный 8-разрядный параллельный порт ввода/вывода. При работе в качестве выходов каждая линия обеспечивает нагрузочную способность, равную 4 входам маломощной серии LS TTL. При записи в разряд регистра порта P3 логической «1» соответствующая линия порта переходит в режим высокоимпедансного входа со слабым подтягиванием сигнала к уровню логической «1».

Выводы порта P3 могут выполнять альтернативные функции в соответствии со следующей таблицей.

Вывод	Альтернативная функция
P3.0.	RxD – вход приемника последовательного порта
P3.1.	TxD – выход передатчика последовательного порта
P3.2.	INT0 – вход внешнего прерывания 0
P3.3.	INT1 – вход внешнего прерывания 1
P3.4.	T0 – внешний вход таймера/счетчика 0
P3.5.	T1 – внешний вход таймера/счетчика 1
P3.6.	WR# – сигнал разрешения записи во внешнюю память данных
P3.7.	RD# – сигнал разрешения чтения из внешней памяти данных

RST. Вход инициализации. Высокий уровень на этом входе в течении двух машинных тактов запускает процесс инициализации микроконтроллера.

ALE. Активное значение сигнала на этом выходе разрешает фиксацию младшего байта адреса при обращениях к внешней памяти.

У микроконтроллеров типа 8751 с внутренней памятью программ это вывод имеет альтернативную функцию **PROG#**. При программировании внутренней памяти на него подается стробирующий сигнал.

PSEN#. Активное значение сигнала на этом выходе разрешает чтение из внешней памяти программ.

EA#. Сигнал на этом входе переключает источник кода при обращении к младшим 4 Кбайтам памяти программ. При EA = 0 и диапазоне адресов 0000H – FFFFH микроконтроллер 8051 выполняет цикл обращения

к внешней памяти программ, при $EA = 1$ обращение по одному из этих адресов приводит к чтению кода из внутренней памяти.

XTAL1. Вход инвертирующего усилителя для синхрогенератора.

XTAL2. Выход инвертирующего усилителя для синхрогенератора.

Vcc. Напряжение питания (+5 В).

Vss. Земля.

2.3. Организация памяти и программно доступные ресурсы

В архитектуре MCS-51 память программ и память данных разделены (гарвардская архитектура). Каждая из них имеет размер 64 Кбайта, выбор одной из двух матриц памяти осуществляется сигналами $PSEN$, $RD\#$, $WR\#$. Организация памяти в микроконтроллерах семейства 8051 иллюстрируется рис. 2.3.

Память программ может быть целиком внешней (сигнал $EA = 0$), либо при обращении по младшим 4 К адресов код извлекается из ячеек внутренней памяти микроконтроллера, а содержимое старших 60 К берется из внешней памяти системы (сигнал $EA = 1$).

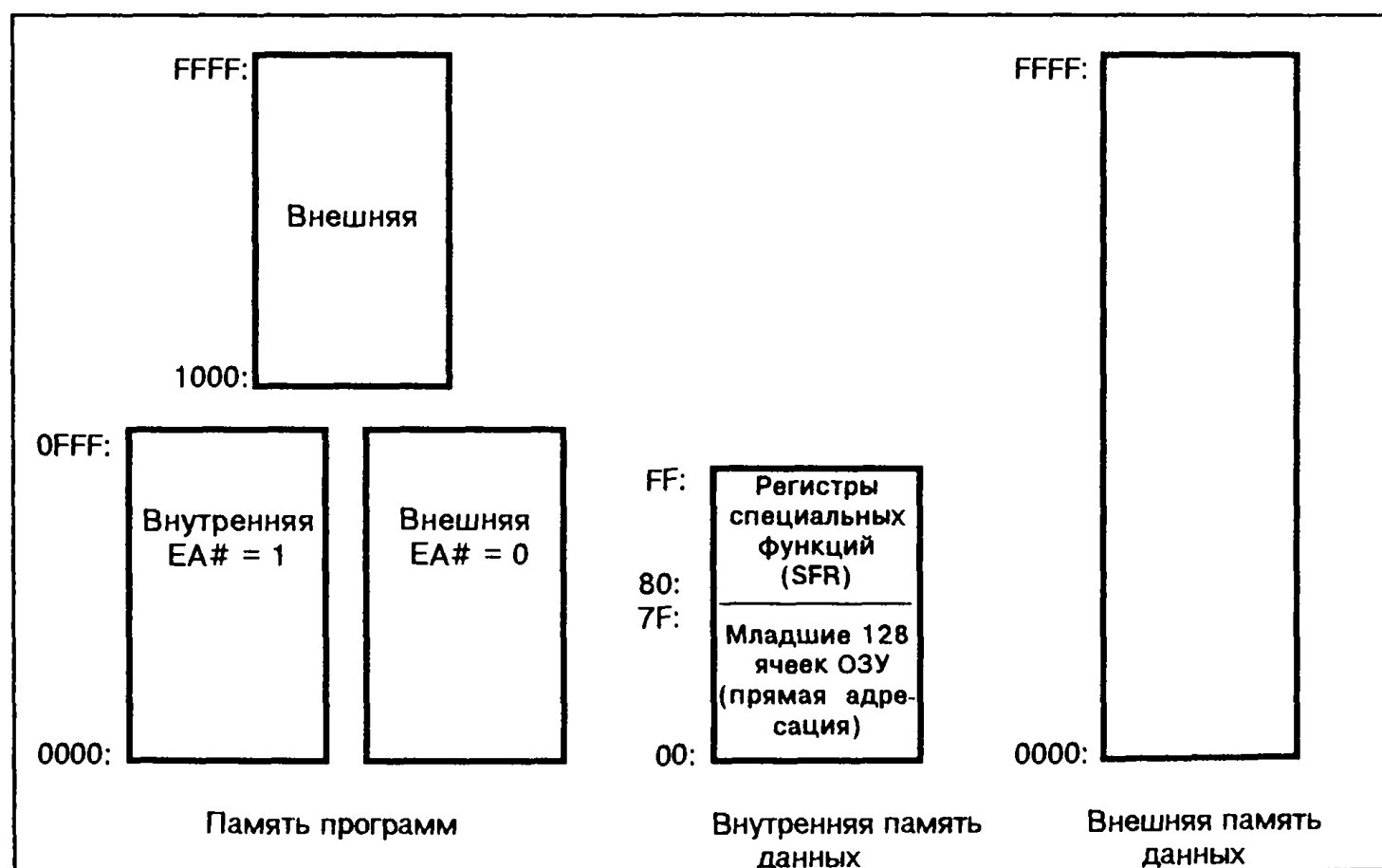


Рис. 2.3. Организация памяти в архитектуре MCS-51

Микроконтроллеры 87C51 имеют 3 бита секретности, предназначенных для защиты внутренней памяти программ от доступа извне. При установленных битах секретности запрещается не только чтение внутренней памяти извне, но и ее допрограммирование. Механизмы программирования и верификации внутренней памяти программ в данной книге рассматриваться не будут, поскольку в продаже имеются программаторы различных типов и стоимости. Возможности универсального профессионального программатора KROM описаны в главе 6.

Память данных делится на внешнюю и внутреннюю, каждая из них имеет свое пространство адресов. В архитектуре MCS-51 пространство адресов внутренней памяти данных объединяет все внутренние программно доступные ресурсы. Это пространство размером 256 байт в свою очередь делится на пространство адресов внутреннего ОЗУ (размером 128 байт) и пространство адресов регистров специальных функций.

Область внутреннего ОЗУ изображена на рис. 2.4, ячейки области занимают младшие адреса внутренней памяти данных с 00H по 7FH.

Шестнадцатеричный адрес									Десятичный адрес
7F:									127
2F:	7F	7E	7D	7C	7B	7A	79	78	47
2E:	77	76	75	74	73	72	71	70	46
2D:	6F	6E	6D	6C	6B	6A	69	68	45
2C:	67	66	65	64	63	62	61	60	44
2B:	5F	5E	5D	5C	5B	5A	59	58	43
2A:	57	56	55	54	53	52	51	50	42
29:	4F	4E	4D	4C	4B	4A	49	48	41
28:	47	46	45	44	43	42	41	40	40
27:	3F	3E	3D	3C	3B	3A	39	38	39
26:	37	36	35	34	33	32	31	30	38
25:	2F	2E	2D	2C	2B	2A	29	28	37
24:	27	26	25	24	23	22	21	20	36
23:	1F	1E	1D	1C	1B	1A	19	18	35
22:	17	16	15	14	13	12	11	10	34
21:	0F	0E	0D	0C	0B	0A	09	08	33
20:	07	06	05	04	03	02	01	00	32
1F:	R7	БАНК3							31
18:	R0								24
17:	R7	БАНК2							23
10:	R0								16
0F:	R7	БАНК1							15
08:	R0								8
07:	R7	БАНК0							7
00:	R0								0

Рис. 2.4. Структура внутреннего ОЗУ данных

В этой области памяти выделяются два особых фрагмента:

- младшие 32 адреса занимают четыре регистровых банка, каждый из которых содержит по восемь регистров общего назначения R0 – R7. Текущий банк определяется значением битов RS0, RS1 регистра PSW. Таким образом, младшие 32 ячейки ОЗУ, кроме адресов, имеют имена;
- ячейки с адресами в диапазоне 32 – 48 имеют прямо адресуемые биты, адреса битов этой части ОЗУ находятся в диапазоне 00H – 7FH.

На рис. 2.5 изображена область регистров специальных функций. В нее включены все программно доступные регистры (управления и данных) внутренних блоков ввода/вывода. Эта область формально занимает старшие 128 байт внутренней памяти данных, но обращение должно осуществляться по определенным адресам ячеек или отдельных битов. Обращение по промежуточным адресам приведет к ошибочному результату.

Шестнадцатеричный адрес									Десятичный адрес	Имя регистра	После сброса
0FF:											
0F0:	F7	F6	F5	F4	F3	F2	F1	F0	240	B	00H
0E0:	E7	E6	E5	E4	E3	E2	E1	E0	224	ACC	00H
0D0:	CY	AC	F0	RS1	RS0	OV		P	208	PSW	00H
	D7	D6	D5	D4	D3	D2		D0			
0B8:			PT2	PS	PT1	PX1	PT0	PX0	184	IP	XX000000B
			BD	BC	BB	BA	B9	B8			
0B0:	B7	B6	B5	B4	B3	B2	B1	B0	176	P3	FFH
0A8:	EA		ET2	ES	ET1	EX1	ET0	EX0	168	IE	0X000000B
	AF		AD	AC	AB	AA	A9	A8			
0A0:	A7	A6	A5	A4	A3	A2	A1	A0	160	P2	FFH
99:									153	SBUF	XXXXXXXXB
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI			
98:	9F	9E	9D	9C	9B	9A	99	98	152	SCON	00H
90:	97	96	95	94	93	92	91	90	144	P1	FFH
8D:									141	TH1	00H
8C:									140	TH0	00H
8B:									139	TL1	00H
8A:									138	TL0	00H
89:									137	TMOD	00H
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0			
88:	8F	8E	8D	8C	8B	8A	89	88	136	TCON	00H
87:									135	PCON	00XX0000B
83:									131	DPH	00H
82:									130	DPL	00H
81:									129	SP	07H
80:	87	86	85	84	83	82	81	80	128	P0	FFH

Рис. 2.5. Регистры специальных функций

Видно, что все регистры имеют как символические имена, так и адреса в качестве ячеек внутренней памяти. Часть регистров содержит прямо адресуемые биты. Адреса битов находятся в диапазоне 80H – F7H.

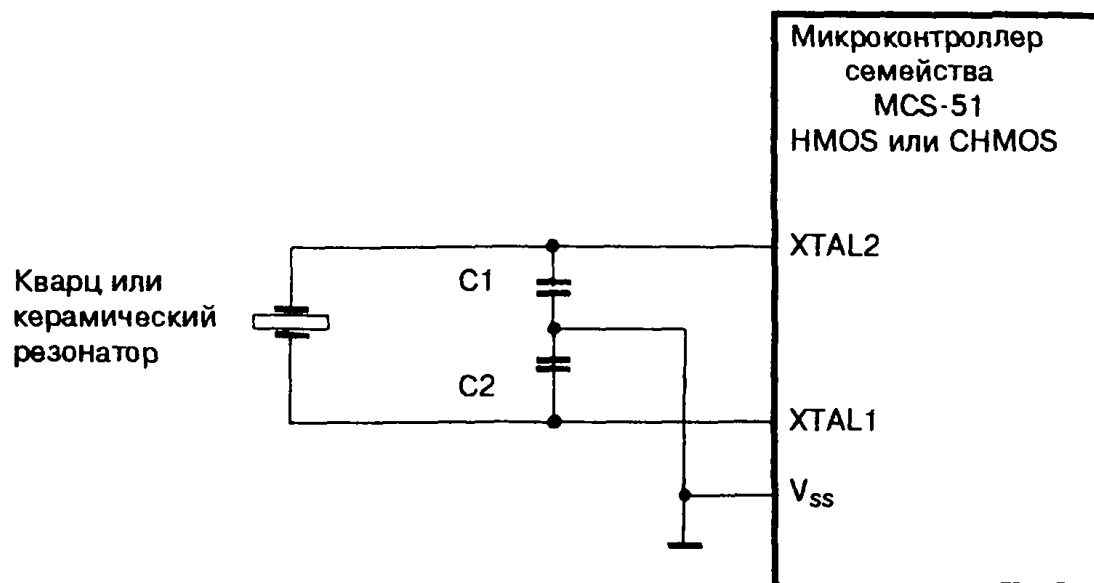
Назначение регистров специальных функций следующее:

Имя регистра	Назначение
ACC	Аккумулятор, основной операционный регистр
B	Дополнительный регистр для операций умножения и деления; в других командах может рассматриваться как обычный РОН
PSW	Регистр, хранящий слово состояния процессора
IP	Регистр приоритетов прерываний
P3	Регистр порта P3
IE	Регистр разрешения прерываний
P2	Регистр порта P2
SBUF	Регистр данных последовательного порта
SCON	Регистр управления последовательного порта
P1	Регистр порта P1
TH1	Старший регистр таймера/счетчика 1
TH0	Старший регистр таймера/счетчика 0
TL1	Младший регистр таймера/счетчика 1
TL0	Младший регистр таймера/счетчика 0
TMOD	Регистр режима таймеров/счетчиков
TCON	Регистр управления таймеров/счетчиков
PCON	Регистр управления энергопотреблением
SP	Регистр указателя стека
DPH, DPL	Указатель данных DPTR состоит из регистра старшего байта DPH и регистра младшего байта DPL, содержит 16-разрядный адрес для обращения к внешней памяти данных
P0	Регистр порта P0

Формат всех регистров специальных функций будет рассмотрен в разделах, посвященных описанию функций отдельных блоков ввода/вывода и системы прерываний.

2.4. Синхронизация, тактовая сетка, циклы команд

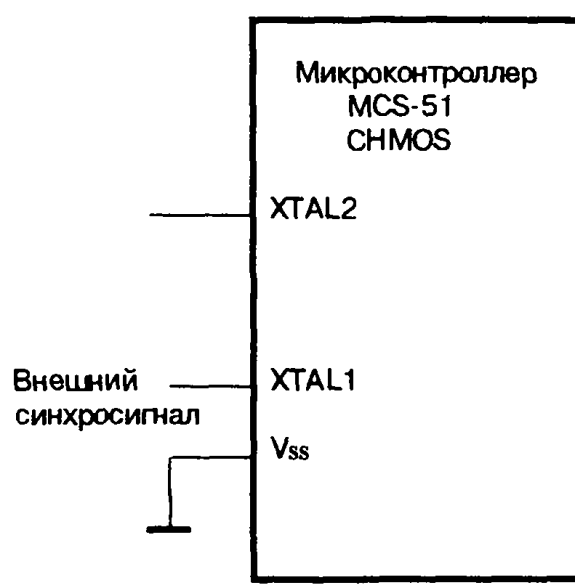
Синхронизация микроконтроллеров MCS-51 осуществляется с использованием внутреннего инвертирующего усилителя, который превращается в синхрогенератор посредством подключения к выводам XTAL1 и XTAL2 внешнего кварцевого резонатора. Вывод XTAL1 является входом, а вывод XTAL2 выходом внутреннего усилителя. Для синхронизации может быть



а)



б)



в)

Рис. 2.6. Синхронизация: использование внутреннего генератора (а); использование внешнего генератора для нМОП микросхем (б); использование внешнего генератора для кМОП микросхем (в)

Внутри микроконтроллера частота внешнего синхросигнала делится на 2. Период внутреннего синхросигнала определяет длительность интервала времени, который фирма Intel называет состоянием Si («state»). Этот интервал состоит из двух фаз P1 и P2. Более крупной единицей времени является машинный цикл, включающий шесть состояний S1 – S6. По длительности он равен 12 периодам внешнего синхросигнала. Машинный цикл служит в основном для целей внутреннего микропрограммного управления. При описании последовательности сигналов или событий фазам в машинном цикле присваивают номера с S1P1 по S6P2. Цикл выполнения

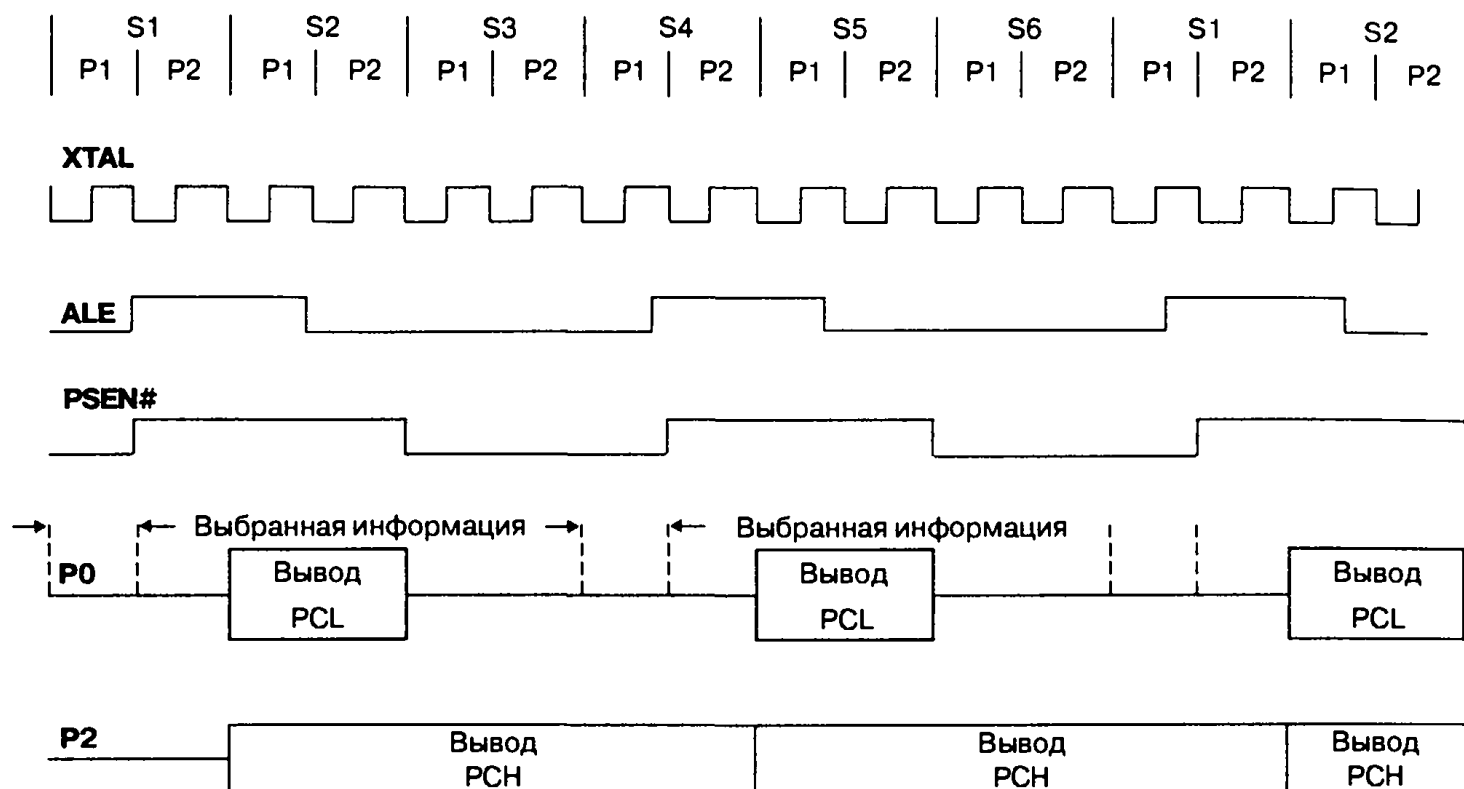
каждой команды состоит из одного, двух или четырех машинных циклов. Протоколы обмена микроконтроллеров семейства MCS-51 по магистрали реализованы в виде комбинаций следующих трех типов магистральных циклов (рис. 2.7):

- ввод кода команды из внешней памяти программ без обращения к внешней памяти данных;
- чтение из внешней памяти данных;
- запись во внешнюю память данных.

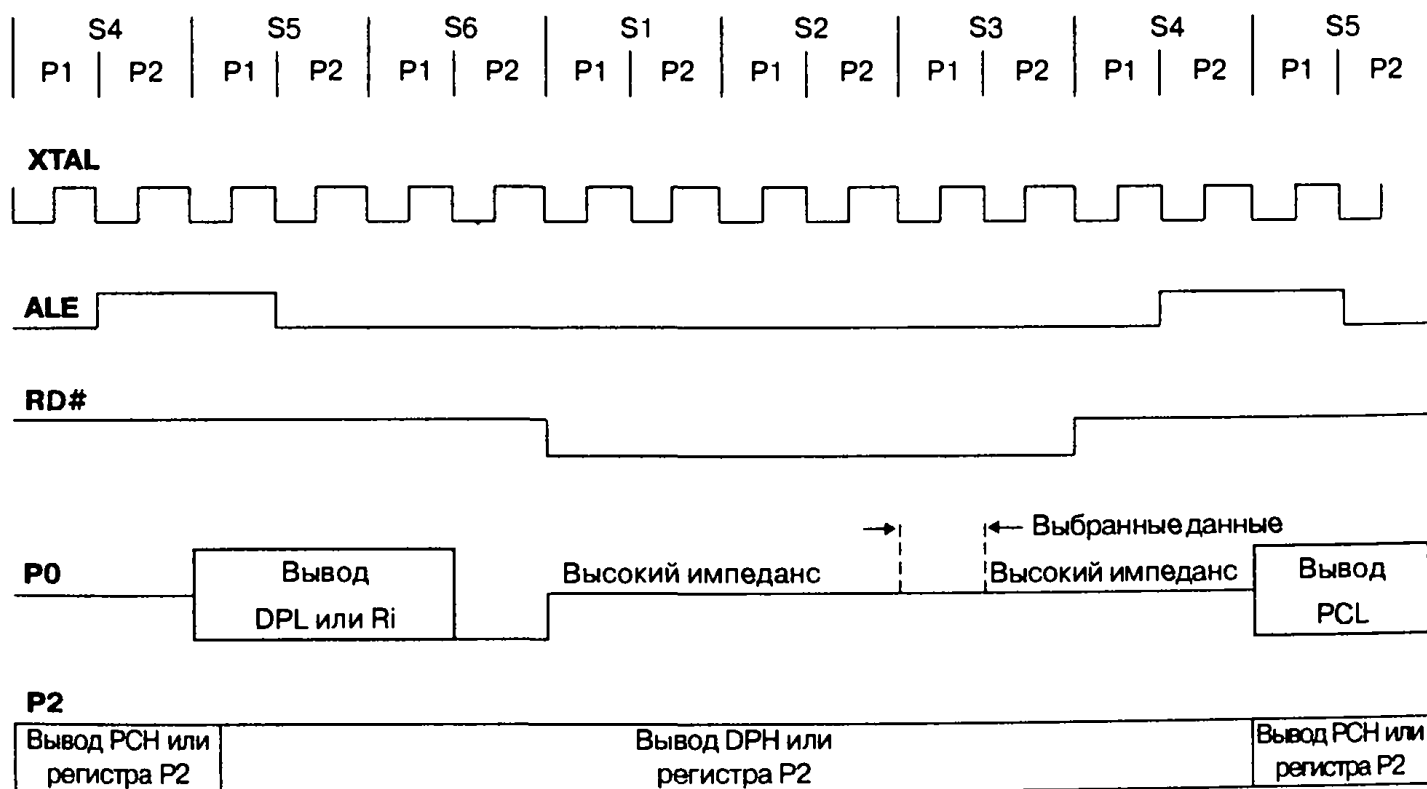
Первый тип магистрального цикла (рис. 2.7а) является основным, имеет длительность 6 состояний (12 периодов XTAL). Он характеризуется выработкой двух импульсов сигнала ALE, который служит для промежуточной фиксации младшего байта адреса при обращении к внешней памяти (из порта P0 в регистр-защелку). В этом типе магистрального цикла сигнал ALE стробирует адрес памяти программ, источником которого является счетчик команд (регистры PCN и PCL). Сигнал PSEN# является сигналом чтения кода из внешней памяти программ. В этом типе магистрального цикла он также вырабатывается дважды. Оба сигнала становятся активными первый раз на фазе S1P2, а второй раз на фазе S4P2. Во время второй выборки производится чтение следующего байта кода. Если этот байт в данной команде не нужен, он игнорируется, а счетчик команд не инкрементируется. Этот тип магистрального цикла является основным, поскольку отсутствие обращения к внешней памяти данных позволяет реализовать с его помощью самые короткие команды (рис. 2.8а,б) длительностью в один машинный цикл (6 тактов, 12 периодов F_{osc}). Поскольку сигнал ALE вырабатывается постоянно, с частотой $F_{osc}/6$, он может быть использован для целей внешней синхронизации.

Второй тип магистрального цикла (рис. 2.7б) имеет длительность 12 состояний и характеризуется тем, что второй сигнал ALE на фазе S1P2 второго машинного цикла не вырабатывается. Сигнал PSEN# остается пассивным с фазы S4P2 первого машинного цикла до фазы S6P1 следующего цикла, запрещая чтение байта кода. Вторым импульсом сигнала ALE стробирует фиксацию адреса памяти данных, источником которого в микроконтроллере является регистр DPTR (DPH и DPL) или регистр Ri (при косвенной адресации). Во втором случае содержимое регистра выдается через порт P0, а старшие восемь разрядов определяются значением в регистре порта P0. Далее, начиная с фазы S1P1 первого цикла до фазы S4P1 второго цикла, становится активным сигнал RD#. Он стробирует ввод данных через порт P0. Последующие импульсы сигналов ALE и PSEN# вводят код следующего байта команды.

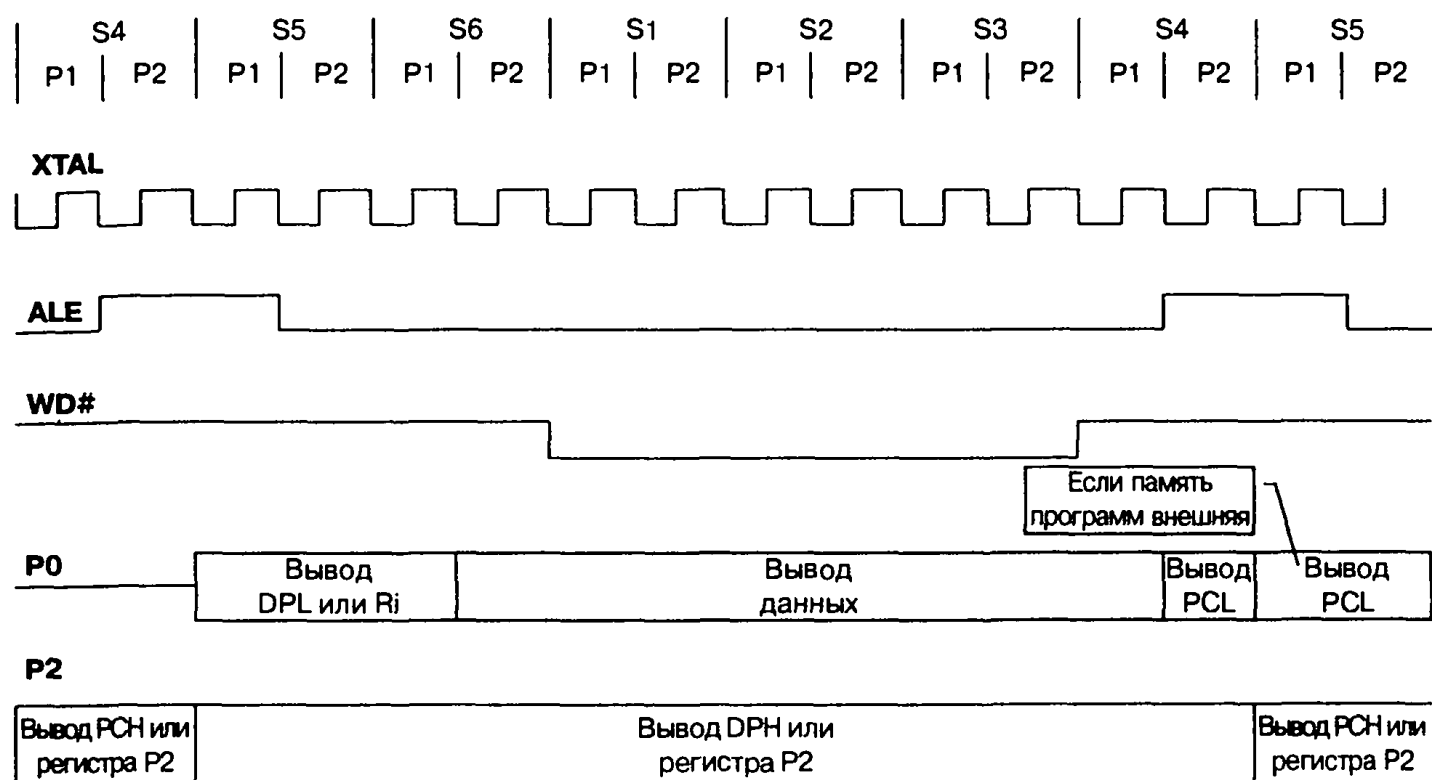
Третий тип магистрального цикла (рис. 2.7в) имеет длительность 12 состояний и в основном аналогичен второму типу. Отличие заключается в том, что активным становится сигнал разрешения записи WR#.



а) Выборка из внешней памяти программ



б) Чтение из внешней памяти данных



в) Запись во внешнюю память данных

Рис. 2.7. Типы магистральных циклов

Особенностью архитектуры MCS-51 является то, что циклы всех команд формируются из циклов магистрали и имеют строго определенную длительность. Примеры циклов ряда характерных команд приведены на рис. 2.8.

Самыми короткими по времени выполнения являются однобайтные и двухбайтные команды, выполняемые за один машинный цикл (рис. 2.8.а,б). Это команды с быстрыми операциями, не использующие обращений к внешней памяти данных. В однобайтных командах (рис. 2.8а) информация хранится в ячейках внутренней памяти данных или регистрах. По второму импульсу сигнала ALE читается код операции следующей команды, который игнорируется. В начале следующей команды он читается вновь. В двухбайтных командах (рис. 2.8б) во втором байте хранятся либо данные, либо адрес внутренней ячейки памяти. В этом случае второй импульс ALE стробирует ввод второго байта текущей команды.

Еще одну группу характерных команд составляют однобайтные команды длительностью два машинных цикла (рис. 2.8в). Это команды с медленными операциями, не использующие обращений к внешней памяти данных. Такими операциями являются, например, наращивание указателя данных DPTR, операции умножения и деления, возврат из подпрограммы. В командах такого типа после чтения кода операции по первому сигналу

ALE три раза подряд выбирается один и тот же следующий байт кода, который игнорируется.

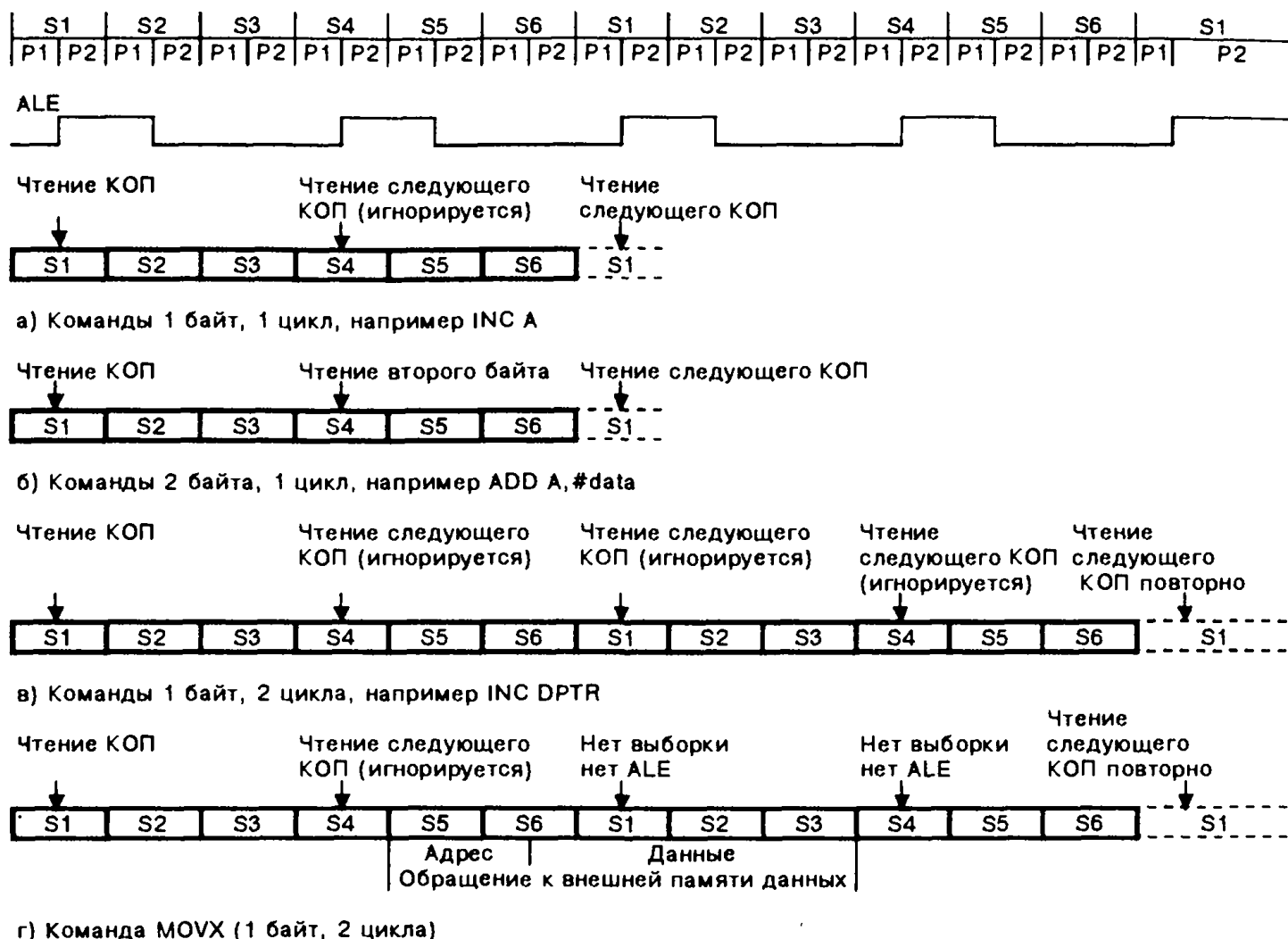


Рис. 2.8. Примеры циклов команд

Короткими, но медленными являются команды типа MOVX с обращением к внешней памяти данных. Они однобайтные, но выполняются за два машинных цикла (рис. 2.8г). После выборки кода операции команды по первому импульсу ALE следующий импульс выбирает следующий байт кода, который игнорируется. Далее импульс ALE стробирует адрес внешней памяти данных, по которому производится операция чтения или записи. По последнему импульсу ALE вводится следующий байт кода.

2.5. Методы адресации и система команд

Система команд микроконтроллеров MCS-51 ориентирована на организацию гибкого ввода/вывода данных через параллельные и последовательный порты, первичную обработку информации. Особое внимание уделено операциям с битами и передаче управления по их значению. Команды, выполняющие такие операции, составляют многочисленную группу и

образуют вместе с соответствующими аппаратными механизмами так называемый «булев процессор» (Boolean processor по терминологии фирмы Intel) в составе архитектуры MCS-51.

В целом систему команд MCS-51 можно охарактеризовать как весьма мощную и симметричную в том смысле, что программист имеет возможность использовать большинство операций с полным набором методов адресации и программно доступных ресурсов аппаратуры. В отличие от большинства RISC микроконтроллеров, система команд MCS-51 весьма удобна для программирования на ассемблере.

Разработчик изделия при создании рабочей программы работает с какой-либо кросс-системой, имеющей свою версию языка программирования. Подробнее о системах программирования будет сказано в главе 6. Ниже изложены основные правила, которые поддерживаются во всех системах программирования на языке ассемблер.

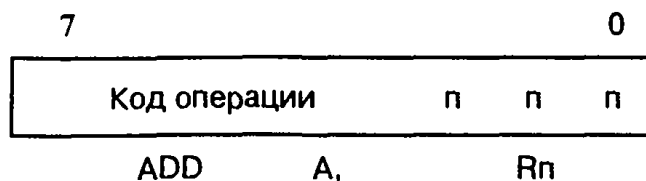
Каждая команда сообщает процессору выполняемую операцию и методы доступа к операндам. Командная строка языка ассемблер содержит метку (символический адрес), мнемонику (символическое имя) команды, операнды, комментарий. Имя команды однозначно связано с выполняемой ею операцией (действием). В командах микроконтроллеров MCS-51 могут обрабатываться до двух операндов. Первым при этом указывается регистр-приемник, вторым регистр-источник. Многие команды по умолчанию используют аккумулятор как один из источников операнда и/или как приемник результата.

Методы адресации представляют собой набор механизмов доступа к операндам. Одни из них просты и поэтому приводят к компактному формату команды и быстрому доступу к операнду, но объем доступных с их помощью ресурсов ограничен. Другие методы адресации позволяют оперировать со всеми имеющимися в системе ресурсами, но команда получается длинной, на ее ввод и выполнение тратится много времени. Набор методов адресации в каждой системе команд является компромиссным сочетанием известных механизмов адресации, выбранным проектировщиками архитектуры исходя из набора решаемых задач. При этом для процессоров с архитектурой CISC (complete instruction set computer), к которой принадлежит MCS-51, большую роль имеет удобство работы программиста.

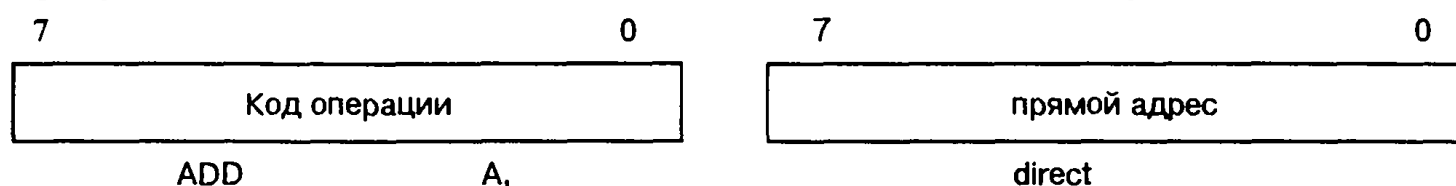
Команды микроконтроллеров семейства MCS-51 могут быть одно-, двух- и трехбайтными. Используется четыре основных метода адресации.

- **Регистровая адресация.** Операнд находится в одном из регистров общего назначения R0-R7 банка, номер которого определяется разрядами RS0, RS1 регистра PSW. Номер регистра определяется тремя младшими битами байта кода операции (КОП – первый байт команды). Регистровая адресация позволяет в одном байте указать код операции и адрес операнда. Поскольку операнд находится во внутренней ячейке, не требуется цик-

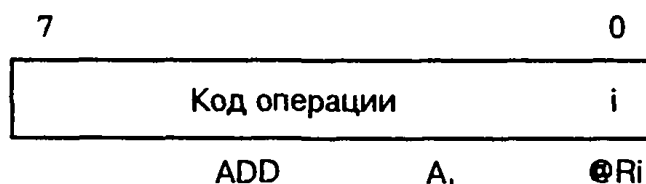
лов обращения к внешней памяти данных. Формат команды сложения ADD A,Rn следующий:



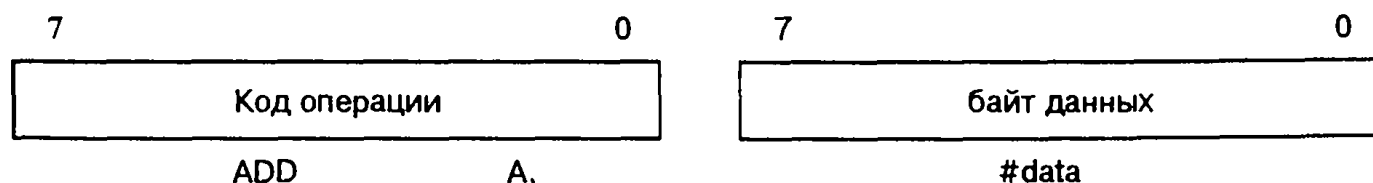
- **Прямая адресация.** Операнд находится во внутренней памяти данных, адрес ячейки определяет отдельный байт команды. Таким образом, команды с использованием прямой адресации имеют минимум два байта, доступны только 256 ячеек, обращения за операндом к внешней памяти не требуется. Формат команды сложения ADD A,direct следующий:



- **Косвенно-регистровая адресация.** Этот метод адресации использует регистры R0 и R1 в качестве указателей. Регистры берутся из банка, определяемого разрядами RS0, RS1 регистра PSW. Содержимое этих регистров используется как адрес для обращения к ячейке внутренней памяти данных, хранящей операнд. Команда получается однобайтной, номер регистра указывает младший бит. Длина команды при этом соответствует длине команды с использованием регистровой адресации, но косвенная адресация дает доступ ко всему пространству внутренней памяти данных. Формат команды сложения ADD A, @Ri следующий:



- **Непосредственная адресация.** Операнд находится в отдельном байте команды, это константа. Таким образом, команды с использованием непосредственной адресации имеют минимум два байта, обращения за операндом к внешней памяти не требуется. Формат команды сложения ADD A, #data следующий:



Отметим, что в общем случае для указания места нахождения каждого из двух операндов, а также приемника результата, в команде должен быть использован свой метод адресации. В «аккумуляторной» архитектуре MCS-51 результат всегда помещается в аккумулятор, поэтому адресуются только два операнда. Иногда один из операндов берется из определенного

регистра, что определяется уникальным кодом операции. По существу это пятый метод – *неявная адресация*.

Состояние процессора после выполнения очередной команды определяется состоянием аккумулятора и регистра PSW, который хранит «слово состояния процессора». Формат этого регистра и функции отдельных битов следующие:

PSW	CY	AC	F0	RS1	RS0	OV		P
------------	-----------	-----------	-----------	------------	------------	-----------	--	----------

Флаг	Бит	Функция
CY	PSW.7	Флаг переноса
AC	PSW.6	Флаг вспомогательного переноса (между тетрадами)
F0	PSW.5	Флаг 0, возможно программирование пользователем в общих целях
RS1	PSW.4	Первый бит селектора регистрового банка
RS0	PSW.3	Нулевой бит селектора регистрового банка
OV	PSW.2	Флаг переполнения
	PSW.1	Флаг, определяемый пользователем
P	PSW.0	Флаг чётности. Указывает чётное число "1" в аккумуляторе

При описании системы команд далее используются следующие обозначения:

R_n – один из регистров $R_0 - R_7$ в банке регистров, определенном битами регистра PSW;

Direct – адрес ячейки внутренней памяти данных.

@ R_i – ячейка внутренней памяти данных, косвенно адресуемая через регистры R_0 или R_1 текущего банка;

#data – 8-разрядная константа, включенная в команду;

#data 16 – 16-разрядная константа, включенная в команду;

addr 16 – 16-разрядный адрес перехода в командах LCALL и LJMP;

addr 11 – 11-разрядный адрес перехода в командах ACALL и AJMP;

rel – 8-разрядная константа со знаком (в дополнительном коде). Определяет смещение в команде SJMP и всех командах условных переходов;

bit – адрес бита во внутренней памяти данных.

Всего в системе команд семейства MCS-51 можно выделить 5 групп. Команды следующим образом изменяют значение флагов регистра PSW:

Команда	Флаг			Команда	Флаг		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	0	X		ANL C, /bit	X		
DIV	0	X		ORL C, bit	X		
DA	X			ORL C, /bit	X		
RRC	X			MOV C, bit	X		
RLC	X			CJNE	X		
SETB C	1						

Арифметические команды. Обеспечивают выполнение операций сложения и вычитания (в том числе с учетом переноса) над целочисленными 8-разрядными операндами, инкремент и декремент регистров и ячеек внутренней памяти данных, двоично-десятичную коррекцию содержимого аккумулятора. Краткая сводка этих команд приведена в следующей таблице, а более подробное их описание можно найти в Приложении 1.

Арифметические команды микроконтроллеров MCS-51

Мнемоника	Описание	Байт	Тактов	Действие
<u>АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ</u>				
ADD A,Rn	Сложение регистра с аккумулятором	1	12	ADD $(A) \leftarrow (A) + (Rn)$
ADD A,direct	Сложение ячейки внутренней памяти данных с аккумулятором	2	12	ADD $(A) \leftarrow (A) + (\text{direct})$
ADD A,@Ri	Сложение косвенно адресуемой ячейки внутренней памяти данных с аккумулятором	1	12	ADD $(A) \leftarrow (A) + ((Ri))$
ADD A,#data	Сложение непосредственного байта данных с аккумулятором	2	12	ADD $(A) \leftarrow (A) + \#data$
ADDC A,Rn	Сложение регистра с аккумулятором с учетом переноса	1	12	ADDC $(A) \leftarrow (A) + (C) + (Rn)$
ADDC A,direct	Сложение ячейки внутренней памяти данных с аккумулятором с учетом переноса	2	12	ADDC $(A) \leftarrow (A) + (C) + (\text{direct})$
ADDC A,@Ri	Сложение косвенно адресуемой ячейки внутренней памяти данных с аккумулятором с учетом переноса	1	12	ADDC $(A) \leftarrow (A) + (C) + ((Ri))$
ADDC A,#data	Сложение непосредственного байта данных с аккумулятором с учетом переноса	2	12	ADDC $(A) \leftarrow (A) + (C) + \#data$
SUBB A,Rn	Вычитание регистра из аккумулятора с учетом заема	1	12	SUBB $(A) \leftarrow (A) - (C) - (Rn)$
SUBB A,direct	Вычитание ячейки внутренней памяти данных из аккумулятора с учетом заема	2	12	SUBB $(A) \leftarrow (A) - (C) - (\text{direct})$
SUBB A,@Ri	Вычитание косвенно адресуемой ячейки внутренней памяти данных из аккумулятора с учетом заема	1	12	SUBB $(A) \leftarrow (A) - (C) - ((Ri))$
SUBB A,#data	Вычитание непосредственного байта данных из аккумулятора с учетом заема	2	12	SUBB $(A) \leftarrow (A) - (C) - \#data$
INC A	Инкремент аккумулятора	1	12	INC $(A) \leftarrow (A) + 1$
INC Rn	Инкремент регистра	1	12	INC $(Rn) \leftarrow (Rn) + 1$
INC direct	Инкремент ячейки внутренней памяти данных	2	12	INC $(\text{direct}) \leftarrow (\text{direct}) + 1$
INC @Ri	Инкремент косвенно адресуемой ячейки внутренней памяти данных	1	12	INC $((Ri)) \leftarrow ((Ri)) + 1$

Мнемоника		Описание	Байт	Тактов	Действие
DEC	A	Декремент аккумулятора	1	12	DEC $(A) \leftarrow (A) - 1$
DEC	Rn	Декремент регистра	1	12	DEC $(Rn) \leftarrow (Rn) - 1$
DEC	direct	Декремент ячейки внутренней памяти данных	2	12	DEC $(direct) \leftarrow (direct) - 1$
DEC	@Ri	Декремент косвенно адресуемой ячейки внутренней памяти данных	1	12	DEC $((Ri)) \leftarrow ((Ri)) - 1$
INC	DPTR	Инкремент указателя данных	1	24	INC $(DPTR) \leftarrow (DPTR) + 1$
MUL	AB	Умножение A на B	1	48	MUL $(A)_{7:0} \leftarrow (A) \times (B); (B)_{15:8}$
DIV	AB	Деление A на B	1	48	DIV $(A)_{15:8} \leftarrow (A) / (B); (B)_{7:0}$
DA	A	Двоично-десятичная коррекция аккумулятора	1	12	DA - содержимое аккумулятора в BCD IF $[(A_{3:0}) > 9] V[(AC) = 1]$ THEN $(A_{3:0}) \leftarrow (A_{3:0}) + 6$ AND IF $[(A_{7:4}) > 9] V[(C) = 1]$ THEN $(A_{7:4}) \leftarrow (A_{7:4}) + 6$

Особенностью системы команд MCS-51 является наличие в составе этой группы команд умножения и деления 8-разрядных целочисленных операндов. Получающийся 16-разрядный результат хранится в регистрах A и B. Результатом выполнения арифметических команд является, кроме нового содержимого аккумулятора, измененное значение регистра PSW.

Логические команды. Выполняют операции AND, OR, «исключающее ИЛИ», сдвиги содержимого аккумулятора вправо и влево, очистку аккумулятора и перестановку его тетрад. В качестве операндов могут использоваться непосредственные данные и содержимое ячеек внутренней памяти данных. Выполнение некоторых логических команд влияет на значение флага C (перенос) регистра PSW.

Логические команды микроконтроллеров MCS-51

Мнемоника		Описание	Байт	Тактов	Действие
ЛОГИЧЕСКИЕ ОПЕРАЦИИ					
ANL	A, Rn	AND регистра и аккумулятора	1	12	ANL $(A) \leftarrow (A) \wedge (Rn)$
ANL	A, direct	AND ячейки внутренней памяти данных и аккумулятора	2	12	ANL $(A) \leftarrow (A) \wedge (direct)$
ANL	A, @Ri	AND косвенно адресуемой ячейки внутренней памяти данных и аккумулятора	1	12	ANL $(A) \leftarrow (A) \wedge ((Ri))$
ANL	A, #data	AND непосредственного байта данных и аккумулятора	2	12	ANL $(A) \leftarrow (A) \wedge \#data$

Мнемоника		Описание	Байт	Тактов	Действие
ANL	direct,A	AND аккумулятора и ячейки внутренней памяти данных	2	12	ANL $(\text{direct}) \leftarrow (\text{direct}) \wedge (A)$
ANL	direct,#data	AND непосредственного байта данных и ячейки внутренней памяти данных	3	24	ANL $(\text{direct}) \leftarrow (\text{direct}) \wedge \#data$
ORL	A,Rn	OR регистра и аккумулятора	1	12	ORL $(A) \leftarrow (A) \vee (Rn)$
ORL	A,direct	OR ячейки внутренней памяти данных и аккумулятора	2	12	ORL $(A) \leftarrow (A) \vee (\text{direct})$
ORL	A,@Ri	OR косвенно адресуемой ячейки внутренней памяти данных и аккумулятора	1	12	ORL $(A) \leftarrow (A) \vee ((Ri))$
ORL	A,#data	OR непосредственного байта данных и аккумулятора	2	12	ORL $(A) \leftarrow (A) \vee \#data$
ORL	direct,A	OR аккумулятора и ячейки внутренней памяти данных	2	12	ORL $(\text{direct}) \leftarrow (\text{direct}) \vee (A)$
ORL	direct,#data	OR непосредственного байта данных и ячейки внутренней памяти данных	3	24	ORL $(\text{direct}) \leftarrow (\text{direct}) \vee \#data$
XRL	A,Rn	XOR регистра и аккумулятора	1	12	XRL $(A) \leftarrow (A) \vee (Rn)$
XRL	A,direct	XOR ячейки внутренней памяти данных и аккумулятора	2	12	XRL $(A) \leftarrow (A) \vee (\text{direct})$
XRL	A,@Ri	XOR косвенно адресуемой ячейки внутренней памяти данных и аккумулятора	1	12	XRL $(A) \leftarrow (A) \vee ((Ri))$
XRL	A,#data	XOR непосредственного байта данных и аккумулятора	2	12	XRL $(A) \leftarrow (A) \vee \#data$
XRL	direct,A	XOR аккумулятора и ячейки внутренней памяти данных	2	12	XRL $(\text{direct}) \leftarrow (\text{direct}) \vee (A)$
XRL	direct,#data	XOR непосредственного байта данных и ячейки внутренней памяти данных	3	24	XRL $(\text{direct}) \leftarrow (\text{direct}) \vee \#data$
CLR	A	Очистка аккумулятора	1	12	CLR $(A) \leftarrow 0$
CPL	A	Инверсия аккумулятора	1	12	CPL $(A) \leftarrow \neg(A)$
RL	A	Сдвиг аккумулятора влево	1	12	RL $(An+1) \leftarrow (An) \quad n = 0-6$ $(A0) \leftarrow (A7)$
RLC	A	Сдвиг аккумулятора влево через перенос	1	12	RLC $(An+1) \leftarrow (An) \quad n = 0-6$ $(A0) \leftarrow (C)$ $(C) \leftarrow (A7)$
RR	A	Сдвиг аккумулятора вправо	1	12	RL $(An) \leftarrow (An+1) \quad n = 0-6$ $(A7) \leftarrow (A0)$
RRC	A	Сдвиг аккумулятора вправо через перенос	1	12	RRC $(An) \leftarrow (An+1) \quad n = 0-6$ $(A7) \leftarrow (C)$ $(C) \leftarrow (A0)$
SWAP	A	Перестановка тетрад аккумулятора	1	12	SWAP $(A_{3:0}) \leftrightarrow (A_{7:4})$

Команды пересылки данных. Осуществляют передачу данных между ячейками внутренней памяти данных (команды MOV), внешней памяти данных (команды MOVBX) и аккумулятором.

Команды пересылки данных микроконтроллеров MCS-51

Мнемоника		Описание	Байт	Тактов	Действие
<u>ПЕРЕСЫЛКА ДАННЫХ</u>					
MOV	A,Rn	Пересылка из регистра в аккумулятор	1	12	MOV (A)←(Rn)
MOV	A,direct	Пересылка из ячейки внутренней памяти данных в аккумулятор	2	12	MOV (A)←(direct)
MOV	A,@Ri	Пересылка из косвенно адресуемой ячейки внутренней памяти данных в аккумулятор	1	12	MOV (direct)←((Ri))
MOV	A,#data	Пересылка непосредственного байта данных в аккумулятор	2	12	MOV (A)←#data
MOV	Rn,A	Пересылка из аккумулятора в регистр	1	12	MOV (Rn)←(A)
MOV	Rn,direct	Пересылка из ячейки внутренней памяти данных в регистр	2	24	MOV (Rn)←(direct)
MOV	Rn,#data	Пересылка непосредственного байта данных в регистр	2	12	MOV (Rn)←#data
MOV	direct,A	Пересылка из аккумулятора в ячейку внутренней памяти данных	2	12	MOV (direct)←(A)
MOV	direct,Rn	Пересылка из регистра в ячейку внутренней памяти данных	2	24	MOV (direct)←(Rn)
MOV	direct,direct	Пересылка из ячейки внутренней памяти данных в ячейку внутренней памяти данных	3	24	MOV (direct)←(direct)
MOV	direct,@Ri	Пересылка из косвенно адресуемой ячейки внутренней памяти данных в ячейку внутренней памяти данных	2	24	MOV (direct)←((Ri))
MOV	direct,#data	Пересылка непосредственного байта данных в ячейку внутренней памяти данных	3	24	MOV (direct)←#data
MOV	@Ri,A	Пересылка из аккумулятора в косвенно адресуемую ячейку внутренней памяти данных	1	12	MOV ((Ri))←(A)
MOV	@Ri,direct	Пересылка из ячейки внутренней памяти данных в косвенно адресуемую ячейку внутренней памяти данных	2	24	MOV ((Ri))←(direct)
MOV	@Ri,#data	Пересылка непосредственного байта данных в косвенно адресуемую ячейку	2	12	MOV ((Ri))←(data)
MOV	DPTR,data16	Загрузка 16-разрядной константы в указатель данных	3	24	MOV (DPTR)←#data _{15 0} DPH ←#data _{15 8} DPL ←#data _{7 0}
MOVC	A,@A+DPTR	Пересылка байта кода, связанного с DPTR, в аккумулятор	1	24	MOVC (A)←((A)+(DPTR))

Мнемоника	Описание	Байт	Тактов	Действие
MOVC A,@A+PC	Пересылка байта кода, связанного с PC, в аккумулятор	1	24	MOVC (PC) $\leftarrow$ (PC)+1 (A) $\leftarrow$ ((A)+(PC))
MOVX A,@Ri	Пересылка байта из внешней памяти данных (8-разр. адрес) в аккумулятор	1	24	MOVX (A) $\leftarrow$ ((Ri))
MOVX A,@DPTR	Пересылка байта из внешней памяти данных (16-разр. адрес) в аккумулятор	1	24	MOVX (A) $\leftarrow$ ((DPTR))
MOVX @Ri,A	Пересылка из аккумулятора в ячейку внешней памяти данных (8-разр. адрес)	1	24	MOVX ((Ri)) $\leftarrow$ (A)
MOVX @DPTR,A	Пересылка из аккумулятора в ячейку внешней памяти данных (16-разр. адрес)	1	24	MOVX (DPTR) $\leftarrow$ (A)
PUSH direct	Загрузка ячейки внутренней памяти данных в стек	2	24	PUSH (SP) $\leftarrow$ (SP)+1 ((SP)) $\leftarrow$ (direct)
POP direct	Выгрузка из стека в ячейку внутренней памяти данных	2	24	POP (direct) $\leftarrow$ ((SP)) (SP) $\leftarrow$ (SP)-1
XCH A,Rn	Обмен между регистром и аккумулятором	1	12	XCH (A) $\leftrightarrow$ ((Rn))
XCH A,direct	Обмен между ячейкой внутренней памяти данных и аккумулятором	2	12	XCH (A) $\leftrightarrow$ (direct)
XCH A,@Ri	Обмен между косвенно адресуемой ячейкой внутренней памяти данных и аккумулятором	1	12	XCH (A) $\leftrightarrow$ (Ri)
XCHD A,@Ri	Обмен младшими тетрадами между косвенно адресуемой ячейкой внутренней памяти данных и аккумулятором	1	12	XCHD (A _{3 0}) $\leftrightarrow$ ((Ri _{3 0}))

Существуют две команды MOVC, позволяющие читать в аккумулятор содержимое памяти программ. К этой группе относятся парные команды PUSH и POP, осуществляющие загрузку и выгрузку содержимого прямо адресуемой ячейки в стек (из стека). Имеются четыре команды взаимного обмена информацией между ячейками внутренней памяти данных (XCH и XCHD). Команды пересылки данных значения флагов в регистре PSW не изменяют.

Команды работы с битами. Эти команды устанавливают в «1» (SETB) или «0» (CLR) прямо адресуемый бит внутренней памяти данных, изменяют его значение на противоположное (CPL), выполняют операции AND и OR над флагом переноса C и прямо адресуемым битом (ANL и ORL), осуществляют пересылку значения между флагом C и прямо адресуемым битом (MOV), осуществляют передачу управления по состоянию флага C или прямо адресуемого бита (JC, JNC, JB, JNB, JBC).

Команды работы с битами микроконтроллеров MCS-51

Мнемоника	Описание	Байт	Тактов	Действие
<u>ОПЕРАЦИЯ С БУЛЕВЫМИ ПЕРЕМЕННЫМИ</u>				
CLR C	Очистка флага переноса	1	12	CLR (C)←0
CLR bit	Очистка бита	2	12	CLR (bit)←0
SETB C	Установка флага переноса	1	12	SETB (C)←1
SETB bit	Установка бита	2	12	SETB (bit)←1
CPL C	Инверсия флага переноса	1	12	CPL (C)←¬(C)
CPL bit	Инверсия бита	2	12	CPL (bit)←¬(bit)
ANL C,bit	AND бита и флага переноса	2	24	ANL (C)←(C) ∧ (bit)
ANL C,/bit	AND инверсии бита и флага переноса	2	24	ANL (C)←(C) ∧ ¬(bit)
ORL C,bit	OR бита и флага переноса	2	24	ORL (C)←(C) ∨ (bit)
ORL C,/bit	OR инверсии бита и флага переноса	2	24	ORL (C)←(C) ∨ ¬(bit)
MOV C,bit	Пересылка бита во флаг переноса	2	12	MOV (C)←(bit)
MOV bit,C	Пересылка флага переноса в бит	2	24	MOV (bit)←(C)
JC rel	Переход, если флаг переноса установлен	2	24	JC (PC)←(PC) + 2 IF (C)=1 THEN (PC)←(PC)+ rel
JNC rel	Переход, если флаг переноса не установлен	2	24	JNC (PC)←(PC) + 2 IF (C)=0 THEN (PC)←(PC)+ rel
JB bit,rel	Переход, если бит установлен	3	24	JB (PC)←(PC) + 3 IF (bit)=1 THEN (PC)←(PC)+ rel
JNB bit,rel	Переход, если бит не установлен	3	24	JNB (PC)←(PC) + 3 IF (bit)=0 THEN (PC)←(PC)+ rel
JBC bit,rel	Переход, если бит установлен, и очистка этого бита	3	24	JBC (PC)←(PC) + 3 IF (bit)=1 THEN (bit)←0 (PC)←(PC)+ rel

Команды этой группы, выполняющие логические операции и осуществляющие пересылку значения бита, изменяют значение флага C регистра PSW. Остальные команды на флаги не действуют.

Команды передачи управления. Передача управления может осуществляться безусловно или по условию, без возврата или с возвратом. В последнем случае для возврата необходимо хранить состояние процессора. Система команд MCS-51 кроме обычных «длинных» переходов с использованием 16-разрядного адреса унаследовала от семейства MCS-48 команды передачи управления с «коротким» 11-разрядным адресом.

Команды передачи управления микроконтроллеров MCS-51

Мнемоника	Описание	Байт	Тактов	Действие
<u>ПЕРЕДАЧА УПРАВЛЕНИЯ</u>				
ACALL addr11	Короткий переход с возвратом.	2	24	ACALL $(PC) \leftarrow (PC) + 2$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC_{7-0})$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC_{15-8})$ $(PC_{10-0}) \leftarrow \text{page address}$
LCALL addr16	Длинный переход с возвратом	3	24	LCALL $(PC) \leftarrow (PC) + 3$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC_{7-0})$ $(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (PC_{15-8})$ $(PC) \leftarrow \text{addr}_{15-0}$
RET	Возврат из подпрограммы	1	24	RET $(PC_{15-8}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ $(PC_{7-0}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$
RETI	Возврат из подпрограммы прерывания	1	24	RETI $(PC_{15-8}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$ $(PC_{7-0}) \leftarrow ((SP))$ $(SP) \leftarrow (SP) - 1$
AJMP addr11	Короткий переход без возврата	2	24	AJMP $(PC) \leftarrow (PC) + 2$ $(PC_{10-0}) \leftarrow \text{page address}$
LJMP addr16	Длинный переход без возврата	3	24	LJMP $(PC) \leftarrow \text{addr}_{15-0}$
SJMP rel	Короткий переход с 8-разрядным смещением	2	24	SJMP $(PC) \leftarrow (PC) + 2$ $(PC) \leftarrow (PC) + \text{rel}$
JMP @A+DPTR	Переход относительно DPTR	1	24	JMP $(PC) \leftarrow (A) + (DPTR)$

Мнемоника		Описание	Байт	Тактов	Действие
JZ	rel	Переход, если аккумулятор равен нулю	2	24	JZ $(PC) \leftarrow (PC) + 2$ IF (A) = 0 THEN $(PC) \leftarrow (PC) + rel$
JNZ	rel	Переход, если аккумулятор не равен нулю	2	24	JNZ $(PC) \leftarrow (PC) + 2$ IF (A) $\neq$ 0 THEN $(PC) \leftarrow (PC) + rel$
CJNE	A,direct,rel	Переход, если ячейка внутренней памяти данных не идентична аккумулятору	3	24	$(PC) \leftarrow (PC) + 3$ IF (A) < > (direct) THEN $(PC) \leftarrow (PC) + relative\ offset$ IF (A) < (direct) THEN (C) $\leftarrow$ 1 ELSE (C) $\leftarrow$ 0
CJNE	A,#data,rel	Переход, если непосредственный байт данных не равен аккумулятору	3	24	$(PC) \leftarrow (PC) + 3$ IF (A) < > (data) THEN $(PC) \leftarrow (PC) + relative\ offset$ IF (A) < (data) THEN (C) $\leftarrow$ 1 ELSE (C) $\leftarrow$ 0
CJNE	Rn,#data,rel	Переход, если непосредственный байт данных не равен регистру	3	24	$(PC) \leftarrow (PC) + 3$ IF ((Ri)) < > data THEN $(PC) \leftarrow (PC) + relative\ offset$ IF ((Ri)) < data THEN (C) $\leftarrow$ 1 ELSE (C) $\leftarrow$ 0
CJNE	@Ri,#data,rel	Переход, если непосредственный байт данных не равен косвенно адресуемой ячейке внутренней памяти данных	3	24	$(PC) \leftarrow (PC) + 3$ IF ((Ri)) < > data THEN $(PC) \leftarrow (PC) + relative\ offset$ IF ((Ri)) < data THEN (C) $\leftarrow$ 1 ELSE (C) $\leftarrow$ 0
DJNZ	Rn,rel	Декремент регистра и переход, если он не равен нулю	2	24	DJNZ $(PC) \leftarrow (PC) + 2$ (Rn) $\leftarrow$ (Rn) - 1 IF (Rn) > 0 or (Rn) < 0 THEN $(PC) \leftarrow (PC) + rel$
DJNZ	direct,rel	Декремент ячейки внутренней памяти данных и переход, если она не равна нулю	3	24	DJNZ $(PC) \leftarrow (PC) + 2$ (direct) $\leftarrow$ (direct) - 1 IF (direct) > 0 or (direct) < 0 THEN $(PC) \leftarrow (PC) + rel$
NOP		Нет операции	1	12	NOP $(PC) \leftarrow (PC) + 1$

Группа команд передачи управления включает четыре команды безусловного перехода без возврата («ультракороткая» SJMP, «короткая»

AJMP, «длинная» LJMP и переход по значению A+DPTR в команде JMP), две команды безусловного перехода с возвратом («короткая» ACALL и «длинная» LCALL), восемь команд условного перехода без возврата. Имеются две команды возврата: из обычной подпрограммы – команда RET и из подпрограммы обработки прерывания – команда RETI. Завершает эту группу и всю систему команд NOP – «нет операции».

Команды передачи управления значения флагов в регистре PSW не изменяют, кроме команд условного перехода CJNE, влияющих на значение флага C.

2.6. Система прерываний

Исходная система прерываний в архитектуре MCS-51 включает пять источников – два внешних и три внутренних. Система развивается, с появлением микроконтроллеров типа 80C52 число источников прерываний в архитектуре семейства постоянно увеличивается, что определяется увеличением количества внутренних блоков ввода/вывода. Некоторые усовершенствования претерпел механизм приоритетов прерываний. Здесь мы рассмотрим исходную систему прерываний, а все отличия будут указаны при описании старших версий микроконтроллеров.

Из пяти источников прерываний внешними являются входы INT0# и INT1#, а внутренними – два счетчика/таймера и последовательный порт. Активные сигналы на входах INT0#, INT1# устанавливают флаги IE0, IE1 регистра TCON. Таймер 0 и Таймер 1 используют для выработки запросов прерываний флаги переполнения TF0, TF1 в регистре управления TCON, а последовательный порт использует для этого два флага – приемника RI и передатчика TI в регистре управления SCON. Каждый из флагов последовательного порта может вызвать прерывание, а конкретный источник определяет процедура обслуживания посредством опроса.

Реакция на любой запрос, разрешенный к обслуживанию, заключается в передаче управления от текущей программы специальной процедуре обслуживания прерывания данного типа. Осуществляется эта передача посредством команды LCALL, код которой не выбирается из памяти, а формируется внутри микроконтроллера. Адрес перехода в архитектуре MCS-51 соответствует вектору прерывания и фиксирован для каждого источника:

Векторы системы прерываний

Источник прерывания	Вектор прерывания
Внешнее прерывание INT0	0003H
Таймер/счетчик T/C0	000BH
Внешнее прерывание INT1	0013H
Таймер/счетчик T/C1	001BH
Последовательный порт	0023H

Из таблицы видно, что интервал между адресами составляет всего 8 ячеек памяти программ, поэтому по первым четырем адресам может располагаться либо очень короткая процедура, либо команда JMP перехода на область памяти, где находится более длинная процедура обслуживания. По адресу последовательного порта места больше. Если по адресам векторов прерываний находятся команды JMP, то имеет место так называемая таблица переходов системы прерываний.

Когда запрос прерывания принят на обслуживание, выполняется следующая последовательность действий. Текущая команда выполняется до конца. Затем внутренне формируемая команда LCALL сохраняет текущее значение счетчика команд PC (адрес следующей команды) в стеке. Это позволяет после завершения процедуры обслуживания прерывания вернуться в прерванную программу. Далее эта команда загружает в PC адрес-вектор. Сохранение в стеке других регистров микроконтроллера, кроме PC, осуществляется программистом командами PUSH в начале процедуры обслуживания. В конце процедуры должны стоять соответствующие команды выгрузки POP. Этот процесс отдан на усмотрение разработчика, поскольку от объема сохраняемой информации сильно зависит время реакции на прерывание. Процедура обслуживания прерывания завершается командой RETI, которая выгружает из стека адрес возврата и помещает его в PC.

Каждый из источников прерываний может быть разрешен или запрещен с использованием соответствующего бита регистра IE. Кроме того, в этом регистре есть бит общего запрета всех прерываний.

При появлении запросов от нескольких источников очередность их обслуживания определяется с помощью механизма приоритетов прерываний. В архитектуре MCS-51 имеет место двухступенчатый механизм определения приоритетов.

IE	EA	-	-	ES	ET1	EX1	ET0	EX0
----	----	---	---	----	-----	-----	-----	-----

Имя бита	Номер бита	Функция
EA	IE.7	Запрещение запросов от всех источников, имеет место при EA = 0
.	IE.6	Зарезервирован
.	IE.5	Зарезервирован
ES	IE.4	Запрещение запроса от последовательного порта, имеет место при ES = 0
ET1	IE.3	Запрещение запроса от Таймера 1, имеет место при ET1 = 0
EX1	IE.2	Запрещение запроса по входу INT1#, имеет место при EX1 = 0
ET0	IE.1	Запрещение запроса от Таймера 0, имеет место при ET0 = 0
EX0	IE.0	Запрещение запроса по входу INT0#, имеет место при EX0 = 0

В исходной системе прерываний микроконтроллеров 8051АН приоритеты на первой ступени имеют два уровня: «высокий» и «низкий», и определяются значениями битов регистра IP.

IP	-	-	-	PS	PT1	PX1	PT0	PX0
----	---	---	---	----	-----	-----	-----	-----

Имя бита	Номер бита	Функция
-	IP.7	Зарезервирован
-	IP.6	Зарезервирован
-	IP.5	Зарезервирован
PS	IP.4	Определяет приоритет последовательного порта
PT1	IP.3	Определяет приоритет Таймера 1
PX1	IP.2	Определяет приоритет входа INT1#
PT0	IP.1	Определяет приоритет Таймера 0
PX0	IP.0	Определяет приоритет входа INT0#

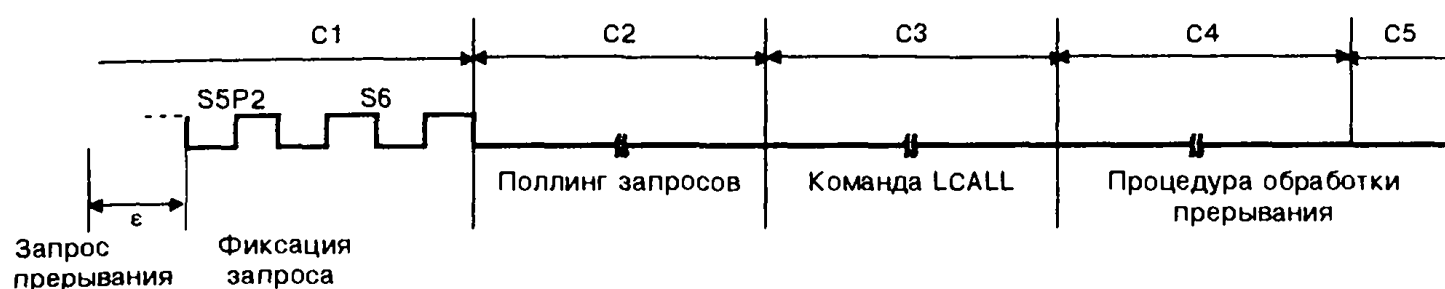
При «1» в определенном разряде регистра IP приоритет соответствующего источника прерывания высокий, а при «0» - низкий. При одновременном появлении нескольких запросов одного уровня очередность обслуживания определяется с помощью внутренней процедуры поллинга (последовательного опроса), который производится в порядке фиксированного старшинства источников внутри одного уровня приоритета. Этот порядок следующий:

Приоритеты прерываний при поллинге

Источник	Приоритет внутри уровня
Вход INT0#	Высший
Таймер 0	
Вход INT1#	
Таймер 1	
Последовательный порт	Низший

Действие механизма приоритетов прерываний заключается в выборе для обслуживания одного из источников при одновременном приходе нескольких запросов, а также в принятии решения о прерывании текущей процедуры обслуживания вновь поступившим запросом. Все источники прерываний проверяются на наличие запроса во время S5P2 каждого машинного цикла (рис. 2.9). В течении следующего машинного цикла анализируются биты приоритетов регистра IP и выполняется внутренняя процедура поллинга.

На основе двухступенчатого анализа выбирается запрос с наиболее высоким приоритетом из поступивших. Если какое либо прерывание уже обслуживается, то его процедуру может прервать только прерывание с более высоким приоритетом. Механизм приоритетов прерываний использует два внутренних программно недоступных флага текущего уровня обслуживания (первой ступени). Прием на обслуживание прерывания устанавливает флаг того уровня приоритета, к которому относится запрос. Этот флаг определяет порог чувствительности системы прерываний. Команда RETI в конце процедуры обслуживания очищает флаг своего уровня и, таким образом, ликвидирует порог чувствительности.



Наиболее быстрый вариант, когда C2 является последним машинным циклом команды (кроме RETI, а также команд обращения к IE и IP).

Рис. 2.9. Вызов процедуры обслуживания прерывания

Из рис. 2.9 видно, что между запросом и началом процедуры обслуживания проходит не менее трех машинных циклов.

Запрос прерывания не принимается на обслуживание и откладывается на более позднее время при следующих обстоятельствах:

- выполняется процедура обслуживания прерывания с более высоким приоритетом;
- текущий машинный цикл не является последним в цикле команды;
- выполняется команда RETI, команды обращения к регистрам IE, IP или следующие за ними.

В текст процедуры обслуживания любого источника прерывания рекомендуется включить команду сброса флага данного источника.

Особенности запросов внешних прерываний. По входам INT0#, INT1# могут восприниматься сигналы запросов, активными значениями которых является либо низкий уровень входного сигнала, либо перепад «1» – «0». Тип активного значения определяется битами IT0, IT1 регистра TCON. При Itx = 0 запрос фиксируется по низкому уровню сигнала на соответствующем входе INTx#, а при Itx = 1 по перепаду.

Если запрос прерывания формируется перепадом сигнала на входе INTx#, высокий и низкий уровни сигнала должны удерживаться не менее одного машинного цикла каждый.

При запросе в виде низкого уровня сигнала активное значение должно удерживаться на входе до начала обработки данного запроса. Далее сигнал должен стать пассивным до завершения процедуры обслуживания.

2.7. Параллельные порты

Все линии ввода/вывода микроконтроллеров в исходной архитектуре MCS-51 сгруппированы в четыре параллельных двунаправленных

8-разрядных порта P0-P3. Эти порты могут использоваться как для выполнения функций обмена данными (допуская при этом поразрядное программирование на ввод или на вывод), так и для выполнения альтернативных функций, таких как обращение к внешней памяти, прием запросов прерываний и другие.

При обращении к внешней памяти используются порты P0 и P2. Через порт P0 выдается младший байт адреса, который должен быть зафиксирован во внешнем регистре, поскольку вслед за ним на линиях порта появляются данные для записи или чтения. Через порт P2 выдается старший байт адреса. В микроконтроллерах типа 8051 и 8751 с внутренней памятью программ в условиях, когда внешняя память программ не используется, порты P0 и P2 могут служить для обмена данными как порты общего назначения. В микроконтроллерах с программируемой внутренней памятью программ порт P0 используется для ввода или вывода байта кода при записи и верификации. Младший байт адреса при программировании и чтении внутренней памяти вводится через порт P1, а старший байт адреса – через порт P2.

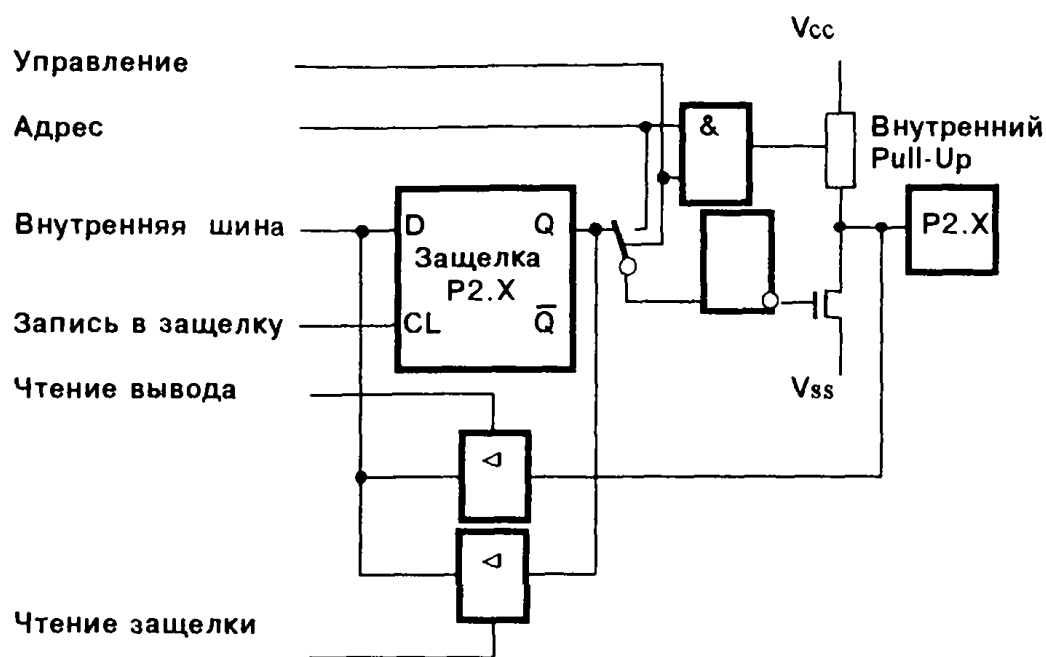
На рис. 2.10 приведены структурные схемы аппаратных средств, обслуживающих одну линию каждого из портов P0-P3. Все линии параллельных портов имеют защелки (типа D-триггера), которые объединены в группы по восемь и рассматриваются как регистры специальных функций P0-P3. Кроме защелки, каждая линия ввода/вывода имеет входной буфер и формирователь выходного сигнала. На рис. 2.10 видно, что на внутреннюю шину микроконтроллера может быть прочитано как состояние каждой из защелок, так и значения сигналов на внешних выводах. Часть команд микроконтроллера во время выполнения читают защелки (регистры портов), а другие – значение сигналов на линиях портов.

Для выполнения портом P3 альтернативных функций в соответствующий разряд регистра порта должна быть записана «1». Во время обращения к внешней памяти во все разряды порта P0 автоматически записываются «1», значения защелок порта P2 не изменяются.

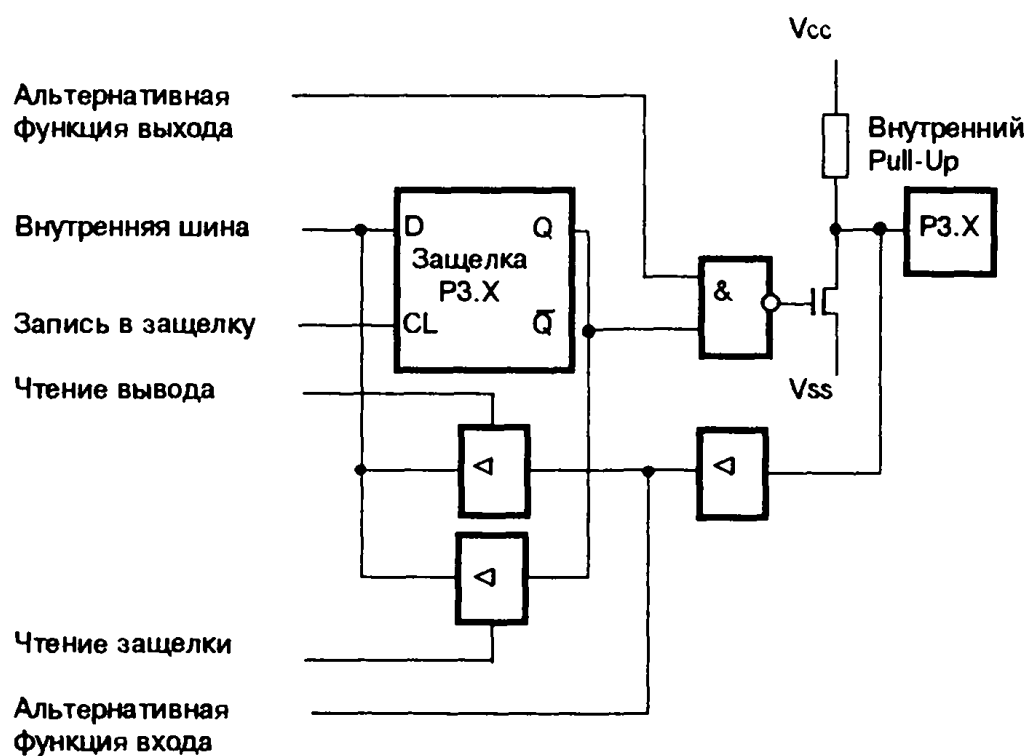
Чтобы вывести значение «0» или «1» на внешний вывод, необходимо записать это значение в соответствующий разряд регистра параллельного порта. Значение появится на внешнем выводе на фазе S1P1 первого машинного цикла следующей команды. Для использования линии порта в качестве входа, ее защелка должна содержать «1». При этом у портов P1, P2, P3 соответствующая линия внутри подтягивается к уровню «1», но может быть внешним источником переведена в состояние «0». Из-за такого механизма порты P1-P3 называют «квазидвунаправленными». Порт P0 элементов постоянного подтягивания к уровню «1» не имеет. Запись «1» в защелку порта P0 переводит соответствующую линию в высокоимпедансное состояние, поэтому этот порт является действительно двунаправленным. Верхний транзистор выходного формирователя открыт только



Рис. 2.10. Аппаратные средства портов P0-P3



В)



Г)

Рис. 2.10. Аппаратные средства портов P0-P3 (продолжение)

Поскольку функция записи «1» и функция чтения для выходного формирователя представляют собой одно и то же состояние с малым выходным током, для ускорения переключения внешних устройств выходные формирователи портов обеспечивают в течение первого машинного такта ($S1P1 - S1P2$) увеличенный ток.

Обновление данных в портах. В командах, которые изменяют значения в защелках, образующих регистры портов P0-P3, новые данные фиксируются на фазе $S6P2$ последнего машинного цикла. На выводе эти данные появляются на фазе $S1P1$ следующего цикла, поскольку выходной

буфер на фазе P2 текущего состояния удерживает значение, имевшееся на фазе P1 (рис. 2.11).

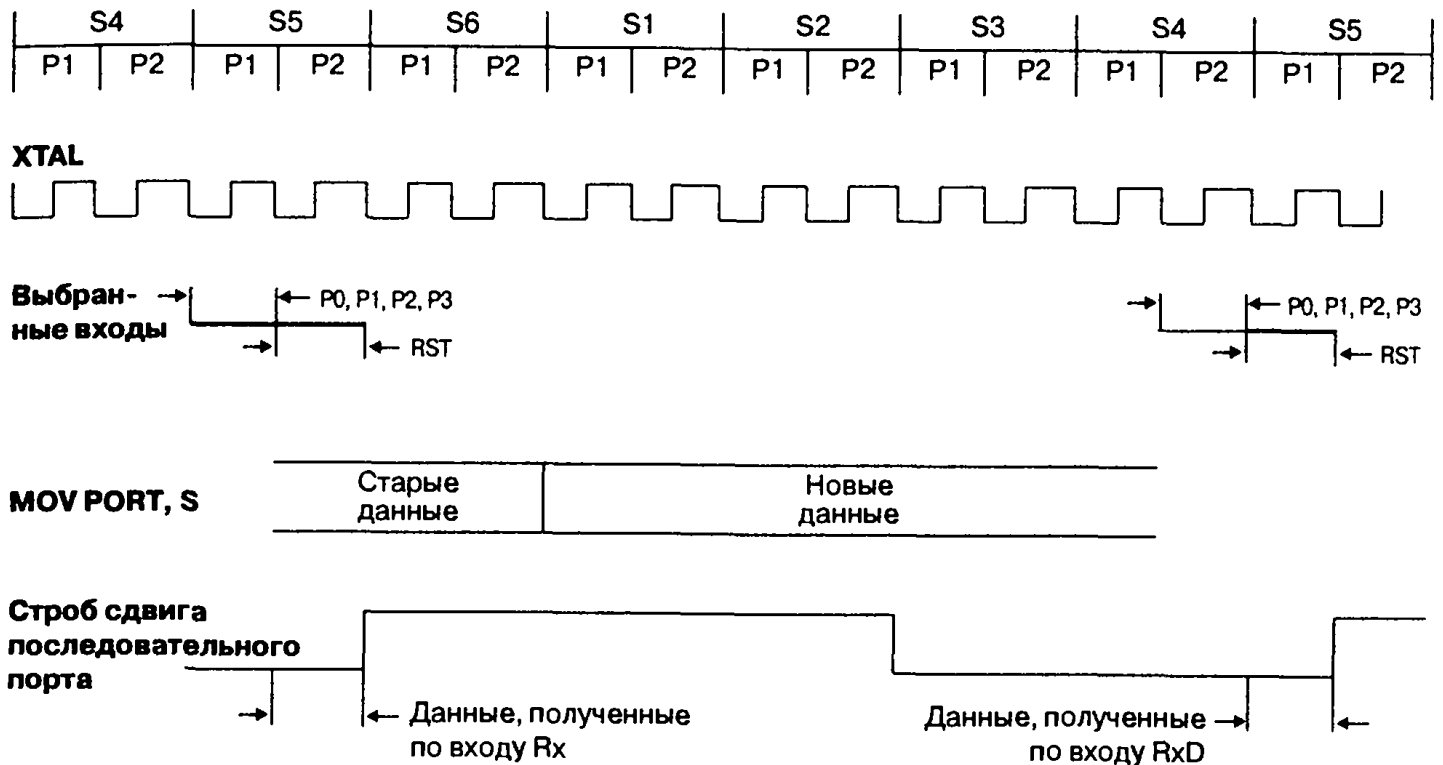


Рис. 2.11. Временные параметры обновления данных в портах

Если при такой операции на линии портов P1, 2 или 3 имеет место переход «0» – «1», то в такте S1 включается элемент "pull-up", обеспечивающий ток, приблизительно в сто раз превышающий обычный рабочий ток. Это ускоряет перезарядку емкостей нагрузки и сокращает время переключения.

Операции типа «чтение-модификация-запись». При обращении к портам P0-P3 одни команды читают состояние регистра порта (или отдельной защелки), а другие читают значение сигнала на выводе. Команды, читающие состояние регистра, затем выполняют некоторую операцию, результат которой вновь записывается в регистр. Они называются командами типа «чтение-модификация-запись». Если приемником является порт или бит порта, следующие команды обращаются к регистру (защелке) порта:

ANL	<port>,	<source>
ORL	<port>,	<source>
XRL	<port>,	<source>
JBC	<port.bit>,	<addr>
CPL	<port.bit>	
INC	<port>	
DEC	<port>	
DJNZ	<port>,	<addr>
CLR	<port.bit>	
SETB	<port.bit>	
MOV	<port.bit>,	<source>

Эти команды дают возможность изнутри правильно прочитать значение порта, когда он высоким значением открывает транзисторный клапан и на линии потенциал падает до логического «0».

2.8. Таймеры/счетчики

В исходной архитектуре MCS-51 два 16-разрядных таймера/счетчика, Таймер 0 и Таймер 1. Каждый из них независимо может быть запрограммирован на работу в качестве либо таймера (отсчет времени через подсчет внутренних импульсов синхронизации), либо счетчика (подсчет событий на внешнем входе). В обоих случаях переход через заранее установленный рубеж приводит к формированию запроса прерывания.

Выбор режима работы каждого из таймеров производится битами $C/Tx\#$ ($x = 0,1$) регистра TMOD. Подсчет событий производится посредством инкремента программно доступного регистра данных, который состоит из регистра младшего байта TLx и регистра старшего байта THx.

При работе в режиме таймера ($C/Tx\# = 0$) регистр TLx наращивается в каждом машинном цикле и частота счета равна $F_{osc}/12$.

В режиме счетчика ($C/Tx\# = 1$) регистр TLx наращивается при переходе сигнала на входе Tx из «1» в «0». Входы опрашиваются во время фазы S5P2 каждого машинного цикла. При обнаружении высокого уровня сигнала в одном цикле и низкого уровня в следующем цикле счетчик инкрементируется. Новое значение в регистре появляется во время фазы S3P1 цикла, следующего за тем, в котором был обнаружен переход. Поскольку на распознавание перехода уходит два машинных цикла, максимальная частота счета составляет $F_{osc}/24$. Ограничений на скважность входного сигнала нет, но для уверенной фиксации перепада сигнал должен удерживаться в каждом значении по крайней мере один машинный цикл.

Счетчики/таймеры обслуживаются регистром режима TMOD и регистром управления TCON.

TMOD	GATE1	C/T1#	M1.1	M1.0	GATE0	C/T0#	M0.1	M0.0
-------------	--------------	--------------	-------------	-------------	--------------	--------------	-------------	-------------

Имя бита	Номер бита	Функция
GATE1	TMOD.7	Бит управления Таймером 1. При GATE1=1 Таймер 1 работает всегда при TR1=1. При GATE1=0 для работы необходимо условие TR1=1 и INT1#=1.
C/T1#	TMOD.6	Бит выбора типа событий для Таймера 1. При C/T1#=1 он работает как счетчик, при C/T1#=0 как таймер.
M1.1	TMOD.5	Бит 1 определения режима работы Таймера 1.
M1.0	TMOD.4	Бит 0 определения режима работы Таймера 1.
GATE0	TMOD.3	Бит управления Таймером 0. При GATE0=1 Таймер 0 работает всегда при TR0=1. При GATE0=0 для работы необходимо условие TR0=1 и INT0#=1.

C/T0#	TMOD.2	Бит выбора типа событий для Таймера 0. При C/T0#=1 он работает как счетчик, при C/T0#=0 как таймер.
M1.0	TMOD.1	Бит 1 определения режима работы Таймера 0.
M0.0	TMOD.0	Бит 0 определения режима работы Таймера 0.

Биты M1 и M0 следующим образом определяют режимы работы таймеров/счетчиков:

M1	M0	Режим работы
0	0	Режим 0. THx как 8-разрядный таймер/счетчик. TLx как 5-разрядный делитель
0	1	Режим 1. 16-разрядный таймер/счетчик. THx и TLx включены последовательно
1	0	Режим 2. 8-разрядный таймер/счетчик TLx с автоперезагрузкой значением из THx
1	1	Режим 3. TL0 как 8-разрядный таймер/счетчик, управляемый битами управления Таймера 0. TH0 как 8-разрядный таймер/счетчик, управляемый битами управления Таймера 1. Таймер 1 не работает.

TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-------------	-----	-----	-----	-----	-----	-----	-----	-----

Имя бита	Номер бита	Функция
TF1	TCON.7	Флаг переполнения Таймера 1. Устанавливается при переходе счетного регистра таймера из состояния FFH в состояние 00H. Очищается при передаче управления на процедуру обработки прерывания.
TR1	TCON.6	Бит запуска Таймера 1. При TR1=1 счет разрешен.
TF0	TCON.5	Флаг переполнения Таймера 0. Устанавливается при переходе счетного регистра таймера из состояния FFH в состояние 00H. Очищается при передаче управления на процедуру обработки прерывания.
TR0	TCON.4	Бит запуска Таймера 0. При TR0=1 счет разрешен.
IE1	TCON.3	Флаг запроса прерывания по вход INT1#.
IT1	TCON.2	Бит селектора типа активного сигнала на входе INT1#. При IT1=1 активным является переход "1" – "0", при IT1=0 активным является низкий уровень сигнала.
IE0	TCON.1	Флаг запроса прерывания по вход INT0#.
IT0	TCON.0	Бит селектора типа активного сигнала на входе INT0#. При IT0=1 активным является переход "1" – "0", при IT0=0 активным является низкий уровень сигнала.

Выбор типа подсчитываемых событий для Таймера 0 и Таймера 1, т.е. назначение им функции таймера или счетчика, определяется значением бита управления C/Tx# регистра TMOD. Таймеры способны работать в трех режимах, выбор режима для каждого таймера производится комбинацией битов M1x, M0x того же регистра.

Счет разрешается битом TRx, если бит GATEx = 0. Когда счетное значение переходит из состояния все «1» в состояние все «0», устанавливается флаг запроса прерывания TFx. Установка бита GATEx = 1 дает возможность управлять таймером от входа INTx# и измерять таким обра-

зом ширину импульсов. Установка битов запуска не очищает счетные регистры.

Режим 0. В этом режиме имеет место 13-разрядный счетный регистр, в котором регистр THx работает как 8-разрядный счетчик, а регистр TLx используется как 5-битный предделитель.

Режим 1. Этот режим аналогичен режиму 0, но используются все 16 разрядов регистров THx и TLx.

Режим 2. В этом режиме регистр TLx работает 8-разрядный счетный регистр. После переполнения он автоматически перезагружается значением из регистра THx, который при этом своего предварительно запрограммированного значения не теряет.

Режим 3. Таймер 1 в этом режиме заблокирован, как если бы бит TR1 был сброшен.

Таймер 0 в режиме 3 работает как два независимых счетных регистра, причем регистр TL0 управляется битами управления Таймера 0, а регистр TH0 управляется битами управления Таймера 1.

2.9. Последовательный порт

Последовательный порт микроконтроллеров MCS-51 является дуплексным и обеспечивает работу в одном синхронном режиме (режим 0) и трех асинхронных режимах (режимы 1, 2, 3). Поддерживается также последовательный обмен в многопроцессорных системах.

При синхронном приеме синхросигнал выдается по линии TxD, а данные принимаются или передаются по линии RxD. При асинхронном обмене блок последовательного порта передает данные по линии TxD, а принимает по линии RxD. Прием и передача могут осуществляться одновременно, поскольку регистр данных SBUF состоит из двух частей. Чтобы передать данные необходимо произвести запись в регистр SBUF, а для приема информации необходимо прочитать содержимое регистра SBUF. Приемник буферизирован, т.е. возможен прием следующего байта, когда предыдущий еще находится в регистре SBUF. Однако, если подпрограмма обслуживания не успела прочесть байт данных из регистра SBUF к моменту завершения приема следующего байта, то этот следующий принятый байт затрет предыдущий. Прием и передача обслуживаются битами прерывания RI и TI, которые формируют общий запрос прерывания.

Управляется последовательный порт с помощью регистра SCON.

SCON

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

Имя бита	Номер бита	Функция
SM0	SCON.7	Бит 0 определения режима работы последовательного порта.
SM1	SCON.6	Бит 1 определения режима работы последовательного порта.
SM2	SCON.5	Разрешение обмена в многопроцессорных системах. В режиме 0 должен быть сброшен. В режиме 1 при SM2=1 бит RI не устанавливается, если принятый стоп-бит=0. В режимах 2 и 3 при SM2=1 бит RI не устанавливается, если принятый девятый бит данных RB8=0.
REN	SCON.4	Разрешение приема. При REN=1 прием разрешен.
TB8	SCON.3	9-й бит данных, который будет передаваться в режимах 2 и 3.
RB8	SCON.2	Значение устанавливается программно по усмотрению разработчика. В режиме 0 не используется. В режиме 1 при SM2=0 в него заносится принятый стоп-бит. В режимах 2 и 3 в него заносится 9-й принятый бит данных.
TI	SCON.1	Флаг запроса прерывания передатчика. Должен быть сброшен программно.
RI	SCON.0	Флаг запроса прерывания приемника. Должен быть сброшен программно.

Биты SM0 и SM1 следующим образом определяют режимы работы последовательного порта:

SM0	SM1	Скорость передачи	Режим работы
0	0	$F_{osc}/12$	Режим 0. Сдвиговый регистр.
0	1	переменная	Режим 1. Асинхронный приемо-передатчик, 8 бит данных в кадре.
1	0	$F_{osc}/64$ или $F_{osc}/32$	Режим 2. Асинхронный приемо-передатчик, 9 бит данных в кадре.
1	1	переменная	Режим 3. Асинхронный приемо-передатчик, 9 бит данных в кадре.

Кроме битов регистра SCON последовательный порт управляется также битом SMOD регистра PCON. В основном регистр PCON управляет режимами энергопотребления, назначение его битов описано в разделе 2.11 «Микроконтроллеры типа 8XC52». Бит SMOD этого регистра управляет опцией удвоения скорости обмена. При SMOD = 1 скорость обмена данными через последовательный порт в режимах 1, 2 и 3 удваивается.

Синхронный обмен (режим 0). В этом режиме по линии TxD выдается восемь синхроимпульсов, а по линии RxD принимается или передается байт данных. Передача и прием данных осуществляются начиная с младшего бита, сдвиг происходит в S6P2 каждого машинного цикла (рис. 2.12). Таким образом последовательный обмен производится с фиксированной частотой $F_{osc}/12$. Это максимально возможная скорость обмена через последовательный порт микроконтроллера.

Передача информации заключается в реализации следующих действий:

- запись в регистр SCON байта управления, очищающего биты SM0, SM1, REN;
- запись байта данных в регистр SBUF.

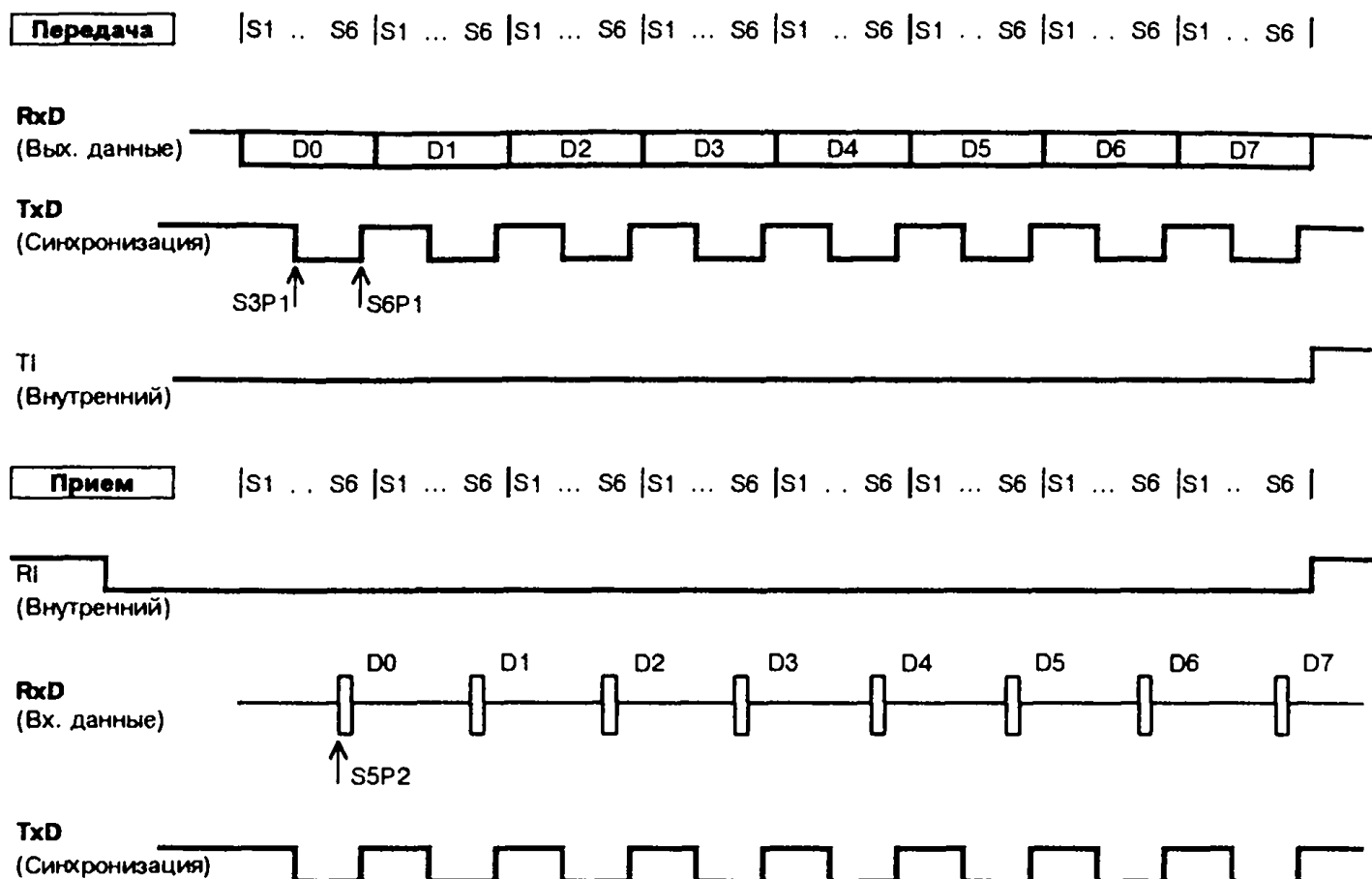


Рис. 2.12. Работа последовательного порта в режиме 0

Передача начинается в следующем (после записи в регистр SBUF) машинном цикле с выдвигания бита D0 на линию RxD в фазе S6P2. В S3P1 следующего цикла на линии TxD появляется «0» первого импульса синхронизации. Выдвижение битов данных продолжается в каждом цикле до девятого включительно, когда на линии RxD появляется старший бит D7. В начале десятого цикла линия RxD переводится в высокоуровневое состояние и устанавливается флаг RI, что свидетельствует об окончании передачи.

Прием информации инициируется записью в регистр SCON байта управления, очищающего биты SM0, SM1, RI и устанавливающего бит REN. Во втором цикле после этой операции записи в фазе S3P1 на линии TxD появляется «0» первого импульса синхронизации, а на фазе S5P2 выполняется ввод младшего бита данных D0 с линии RxD в сдвиговый регистр. Ввод повторяется в последующих циклах и после восьми сдвигов в регистр вводится старший бит D7, устанавливается флаг TI. После этого подпрограмма обслуживания может прочесть байт данных из SBUF.

Асинхронный обмен (режимы 1, 2, 3). При асинхронном обмене данные передаются по линии TxD, а принимаются по линии RxD.

В режиме 1 кадр данных состоит из 10 бит (рис. 2.13): старт-бит, восемь битов данных, стоп-бит. При приеме в разряд RB8 регистра SCON

записывается стоп-бит. Скорость обмена определяется переполнением таймера 1.

В режимах 2 и 3 кадр данных состоит из 11 бит (рис. 2.14): старт-бит, восемь битов данных, программируемый бит, стоп-бит. При приеме подпрограмма обслуживания может прочитать девятый бит из разряда RB8 регистра SCON. При передаче девятый бит записывается в TB8 регистра SCON. Пользователь может использовать девятый бит по своему усмотрению (часто это бит контроля).

Скорость обмена в режиме 2 равна $1/32$ ($SMOD = 1$) или $1/64$ ($SMOD = 0$) частоты синхросигнала. Скорость передачи в режиме 3 определяется переполнением таймера 1.

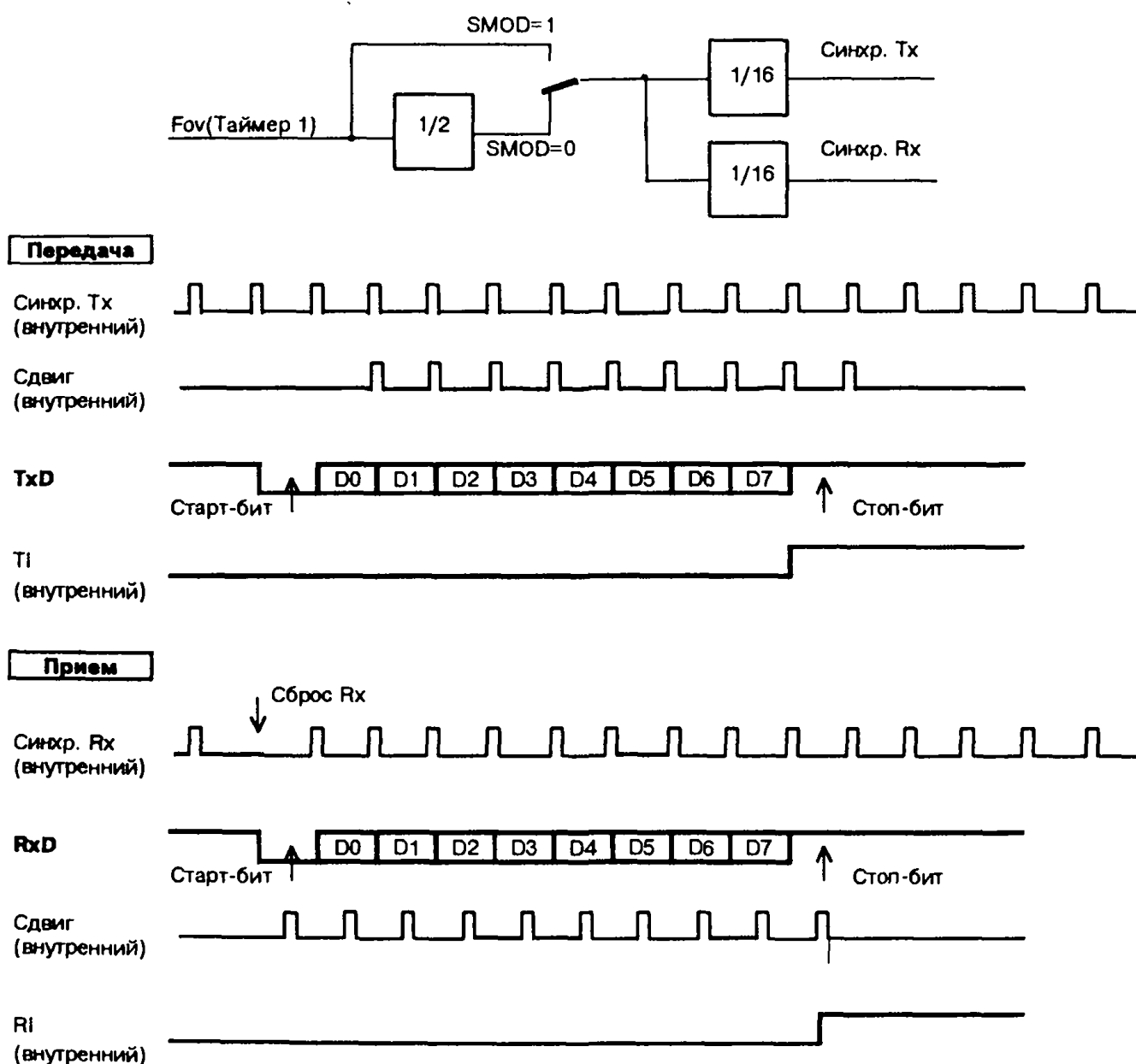


Рис. 2.13. Работа последовательного порта в режиме 1

Передача информации заключается в реализации следующих действий:

- запись в регистр SCON байта управления, определяющего биты SM0, SM1 (выбор режима) и очищающего бит REN. Для режимов 2 и 3 следует записать значение девятого бита данных в TB8;
- запись байта данных в регистр SBUF.

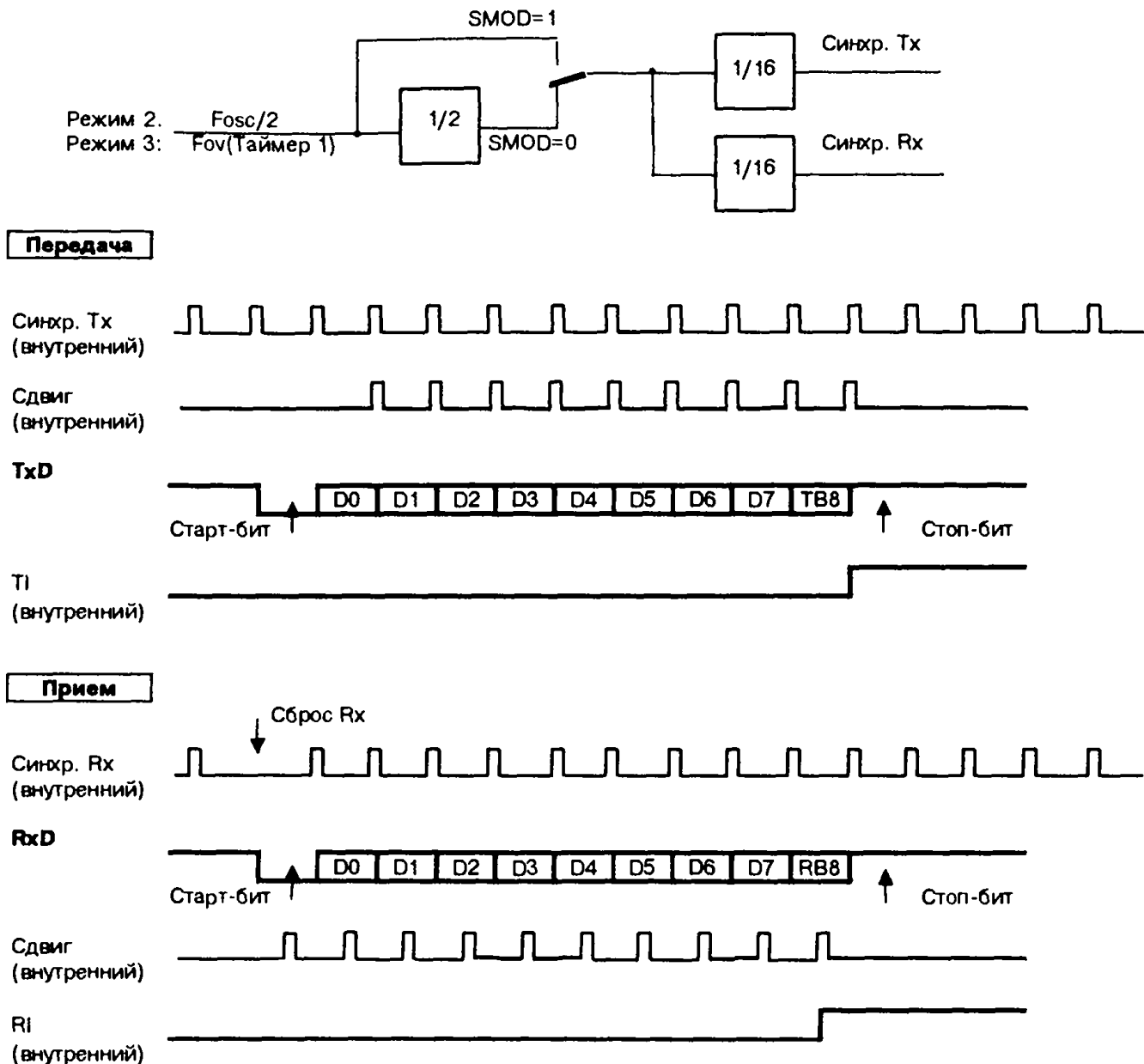


Рис. 2.14. Работа последовательного порта в режимах 2 и 3

Для инициализации приема следует установить бит REN регистра SCON. Прием данных начинается после идентификации перехода «1» – «0» на линии RxD.

При синхронизации последовательного порта от Таймера 1, работающего в режиме автоперезагрузки, скорость обмена определяется по формуле:

$$\text{Скорость} = (2^{SMOD} * F_{OSC}) / (32 * 12 * [256 - (TH)]),$$

где (TH) – содержимое регистра TH1 в десятичном представлении.

Обмен в многопроцессорных системах. Последовательный порт микроконтроллеров MCS-51 способен поддерживать с использованием Режимов 2 и 3 обмен в многопроцессорных системах. При этом передающий микроконтроллер играет роль ведущего, а принимающие микроконтроллеры – роль ведомых. Механизм такой передачи аппаратно поддерживается битом SM2 регистра SCON. Ведущий микроконтроллер посылает вначале посылки кадр, содержащий адрес одного из ведомых микроконтроллеров. Адрес отличается от данных тем, что его девятый бит данных установлен в 1, в то время как у кадра данных девятый бит равен 0. При SM2 = 1 кадр адреса вызывает прерывание, а кадр данных нет. Процедуры прерываний всех ведомых анализируют принятый адрес, при идентификации своего адреса микроконтроллер сбрасывает бит SM2 и читает последующие данные. Остальные ведомые оставляют бит SM2 установленным и продолжают выполнять текущую программу.

2.10. Режимы пониженного энергопотребления

У микроконтроллеров MCS-51, выполненных по CMOS технологии, существует два режима пониженного энергопотребления. Управление этими режимами осуществляется при помощи регистра PCON.

PCON

SMOD	SMOD0	–	–	GF1	GF0	PD	IDL
------	-------	---	---	-----	-----	----	-----

Имя бита	Номер бита	Функция
SMOD	PCON.7	Бит управления скоростью обмена последовательного порта. При SMOD = 1 скорость обмена в режимах 1, 2, 3 удваивается.
SMOD0	PCON.6	Бит выбора значения SCON.7 (начиная с 8XC52/54/58). При SMOD0 = 0 бит SCON.7 представляет флаг SM0, при SMOD0 = 1 бит SCON.7 представляет флаг FE .
–	PCON.5	Зарезервирован.
–	PCON.4	Зарезервирован.
GF1	PCON.3	Флаг общего назначения.
GF0	PCON.2	Флаг общего назначения.
PD	PCON.1	Бит управления режимом микропотребления. При PD = 1 этот режим включен.
IDL	PCON.0	Бит управления режимом холостого хода. При IDL = 1 этот режим включен.

Если установлены оба бита, PD и IDL, больший вес имеет бит PD и микроконтроллер переходит в режим микропотребления.

Режим холостого хода (Idle). В этом режиме центральный процессор отключается, система прерываний, счетчики/таймеры и другие блоки ввода/вывода продолжают функционировать. Счетчик команд, регистры и внутреннее ОЗУ сохраняют свои значения. Последней выполняемой ко-

мандой перед приостановкой процессора является команда, устанавливающая бит IDL. Потребляемая мощность в режиме холостого хода составляет около 15% от номинальной для микроконтроллеров типа 8XC51 и около 30% для микроконтроллеров типа 8XC51GB. Разница объясняется большим количеством блоков ввода/вывода у старших моделей.

Одним из двух возможных способов выхода из режима холостого хода является формирование любого разрешенного запроса прерывания. Принятое на обслуживание прерывание сбрасывает бит IDL регистра PCON. После команды RETI процедуры обслуживания возобновляется выполнение основной программы. Флаги GF0, GF1 регистра PCON могут использоваться процедурой прерывания для определения того, выполняется она в режиме холостого хода или в обычном режиме. Для этого, например, команда, устанавливающая флаг IDL, может установить определенную комбинацию в битах GF0, GF1. В процедуре прерывания наличие этой комбинации должно проверяться.

Другим способом выхода из состояния холостого хода является подача активного сигнала на вход RESET. Длительность этого сигнала должна быть не менее 24 периодов частоты синхронизации.

Режим микропотребления (Power down). Команда, в которой устанавливается бит PD является последней перед переходом в режим микропотребления. В этом режиме приостанавливается выполнение всех функций микроконтроллера, поскольку прекращает работать синхрогенератор. Состояние внутреннего ОЗУ данных сохраняется, содержимое регистров специальных функций теряется.

Выход из состояния микропотребления может быть осуществлен только подачей активного сигнала на вход RESET. Длительность сигнала должна быть не меньше 10 мс.

2.11. Микроконтроллеры типа 8xC52

Микроконтроллеры типа 8xC52 (80C52, 87C52, 80C32) реализуют модифицированную архитектуру MCS-51, в которой исходные принципы развиты с тем, чтобы оперировать ресурсами большего объема. Они изготавливаются только с использованием КМОП технологии, отличаются от микроконтроллеров типа 8051 наличием внутренней памяти программ объемом 8 Кбайт, внутреннего ОЗУ данных размером 256 байт и трех счетчиков/таймеров. Вывод P1.0 получил альтернативную функцию T2 (счетный вход Таймера 2), а вывод P1.1 альтернативную функцию T2EX (вход управления фиксацией/перезагрузкой значения Таймера 2). Последовательный порт этих микроконтроллеров может тактироваться от Таймера 2, он получил функции определения ошибки кадра и автоматического распознавания адреса. В систему прерываний введен второй регистр приорите-

тов IPH, работающий совместно с регистром IP. Кроме того, появился отладочный режим ONCE (on-circuit emulation).

Внутренняя память программ. Микроконтроллеры 80C52, 87C52, 89C52 имеют внутреннюю память программ размером 8 Кбайт (рис. 2.15). В первом случае это ПЗУ, программируемое при помощи маски на заводе-изготовителе. В остальных случаях это память, программируемая пользователем (ОТР или перезаписываемая flash).

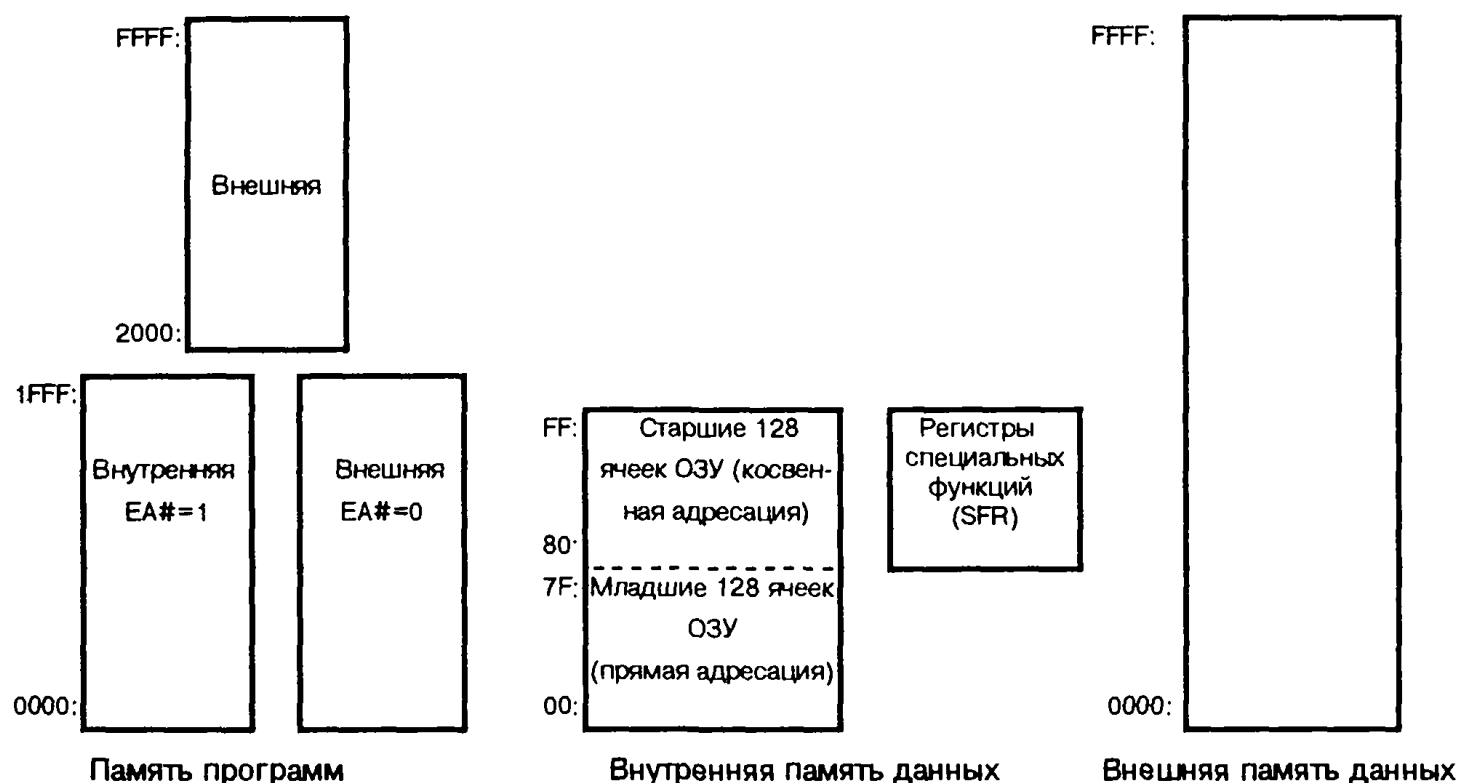


Рис. 2.15. Организация памяти микроконтроллеров типа 8X52

Внутренняя память программ имеет защиту от чтения информации. В микроконтроллерах разных фирм и модификаций это могут быть от одного до трех битов секретности и кодировочная таблица.

Внутренняя память данных и регистры специальных функций. Увеличение размера внутренней памяти данных привело к наложению старших 128 байт ОЗУ данных и пространства регистров специальных функций (рис. 2.15). Выбор той или иной области при обращении осуществляется аппаратурой микроконтроллера на основе используемого метода адресации. Обращение к старшим 128 байтам ОЗУ производится с использованием косвенно-регистровой адресации, а к регистрам специальных функций с использованием прямой адресации.

К набору регистров специальных функций добавлены регистры данных, перезагрузки и управления Таймера 2, а также регистр приоритетов прерываний IPH. Область регистров специальных функций микроконтроллеров 8xC52 можно представить так, как изображено на рис. 2.16.

F8								FF
F0	B 00000000							F7
E8								EF
E0	ACC 00000000							E7
D8								DF
D0	PSW 00000000							D7
C8	T2CON 00000000	T2MOD 00000000	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		CF
C0								C7
B8	IP x0000000	SADEN 00000000						BF
B0	P3 11111111						IPH x0000000	B7
A8	IE 00000000	SADDF 00000000						AF
A0	P2 11111111							A7
98	SCON 00000000	SBUF xxxxxxxx						9F
90	P1 11111111							97
88	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8F
80	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000			PCON 00x0000	87

Рис. 2.16. Регистры специальных функций микроконтроллеров 8xC52

Вновь добавленные регистры выделены, их функции рассмотрены далее.

Алгоритм формирования сигнала ALE у этих микроконтроллеров имеет небольшое отличие. Выдача этого сигнала на внешний вывод может быть запрещена с помощью установки младшего бита регистра с адресом 8EH. В результате на этом выводе удерживается «1», за исключением циклов обращения к внешней памяти при помощи команд MOVX и MOVC.

Таймер 2. Представляет собой 16-разрядный таймер/счетчик, который способен работать в трех режимах:

- режим захвата (фиксации) текущего счетного значения;
- режим прямого и обратного счета с автоперезагрузкой исходного значения;
- режим задающего генератора для последовательного порта.

Режимы работы определяются следующими комбинациями битов управления регистра T2CON:

RCLK+TCLK	CP/RL2#	TR2	Режим
0	0	1	16-разрядный таймер/счетчик с автоперезагрузкой
0	1	1	16-разрядный таймер/счетчик с захватом текущего значения
1	X	1	Задающий генератор для последовательного порта

Регистр данных Таймера 2 состоит из регистров спецфункций TH2 (старший байт) и TL2 (младший байт). Данные для автоперезагрузки хранятся в регистрах RCAP2H (старший) и RCAP2L (младший), а регистрами управления являются T2CON и T2MOD.

Выбор типа фиксируемых Таймером 2 событий определяется битом C/T2# регистра T2CON. При C/T2# = 1 он работает как счетчик, а при C/T2# = 0 как таймер. В обоих случаях битом CP/RL2# может быть установлен режим захвата счетного значения (CP/RL2# = 1), либо режим счета с автоперезагрузкой (CP/RL2# = 0). Захват значения из регистров данных TH2, TL2 и перезагрузка этих регистров исходным значением производится с использованием регистров RLAP2H, RLAP2L.

T2CON

TF2	EXF2	RCLK	TCLK0	EXEN2	TR2	C/T2#	CP/RL2#
-----	------	------	-------	-------	-----	-------	---------

Имя бита	Номер бита	Функция
TF2	T2CON.7	Флаг переполнения Таймера 2. Устанавливается при переходе счетного регистра таймера из состояния FFH в состояние 00H. Должен очищаться программно. TF2 не устанавливается, если RCLK=1 или TCLK=1.
EXF2	T2CON.6	Флаг внешнего события Таймера 2. Устанавливается по перепаду "1"- "0" на входе T2EX, если EXEN2=1. Является запросом прерывания от Таймера 2.
RCLK	T2CON.5	Бит выбора источника синхрочастоты для приемника последовательного порта в его режимах 1 и 3. При RCLK=1 источником является Таймер 2, при RCLK=0 источником является Таймер 1.
TCLK	T2CON.4	Бит выбора источника синхрочастоты для передатчика последовательного порта в его режимах 1 и 3. При TCLK=1 источником является Таймер 2, при TCLK=0 источником является Таймер 1.
EXEN2	T2CON.3	Бит разрешения внешнего события Таймера 2. При EXEN2=1 разрешена установка флага EXF2.
TR2	TCON.2	Бит запуска/останова Таймера 2. При TR2=1 таймер запускается.
C/T2#	T2CON.1	Бит выбора типа событий для Таймера 2. При C/T2#=1 он работает как счетчик, при C/T2#=0 как таймер.
CP/RL2#	TCON.0	Бит выбора режима Таймера 2. При CP/RL2#=1 по перепаду "1"- "0" на входе T2EX (если EXEN2=1) он переходит в режим захвата. При CP/RL2#=0 по перепаду "1"- "0" на входе T2EX (если EXEN2=1) или по переполнению Таймера 2 он переходит в режим автоперезагрузки. Если RCLK=1 или TCLK=1, этот бит игнорируется, а Таймер 2 работает в режиме перезагрузки по переполнению.

T2MOD	-	-	-	-	-	-	T20E	DCEN
--------------	---	---	---	---	---	---	------	------

Имя бита	Номер бита	Функция
T20E	T2MOD.1	Бит разрешения выдачи сигнала программируемой частоты от Таймера 2 на внешний вывод P1.0.
DCEN	T2MOD.0	Бит разрешения счета Таймера 2 в режиме автоперезагрузки. При T20E=1 счет разрешен в прямом или обратном направлении, в зависимости от значения сигнала на входе T2EX. При T20E = 0 (значение после сброса) счет разрешен в прямом направлении.

Бит DCEN этого регистра после сброса очищен. При этом Таймер 2 в режиме автоперезагрузки считает в прямом направлении. Если бит DCEN установлен, направление счета определяется значением внешнего сигнала на входе T2EX. Бит EXF2 при этом не является флагом прерывания и может служить в качестве индикатора переполнения при счете в прямом и обратном направлениях. При T2EX = 1 счет идет в прямом направлении, при переполнении устанавливается флаг прерывания TF2 и осуществляется перезагрузка счетных регистров TH2, TL2 заранее запрограммированными значениями из регистров RCAP2H, RCAP2L. При T2EX = 0 счет идет в обратном направлении, при достижении значения, занесенного в регистры RCAP2H, RCAP2L, устанавливается флаг прерывания TF2 и осуществляется перезагрузка счетных регистров TH2, TL2 16-разрядным значением 0FFFFH.

Таймер 2 имеет два флага запросов прерывания, TF2 и EXF2. Оба запроса обслуживаются с использованием одного вектора 2BH. Идентификацию запроса производит процедура обслуживания прерывания. Флаг TF2 устанавливается при переполнении счетного регистра Таймера 2 (RCLK = TCLK = 0), а флаг EXF2 устанавливается по перепаду «1» – «0» на входе T2EX, если EXEN2 = 1.

В режиме задающего генератора (RCLK = 1 или TCLK = 1) приемник или передатчик последовательного порта тактируются синхроимпульсами, формируемыми при переполнении счетного 16-разрядного регистра Таймера 2. При C/T2# = 0 скорость работы последовательного порта определяется формулой:

$$\text{Скорость} = F_{OSC} / [32 * (65536 - (RCAP2H, RCAP2L))].$$

Значение регистров RCAP2H, RCAP2L для нужной скорости обмена можно получить по формуле:

$$RCAP2H, RCAP2L = 65536 - F_{OSC} / (32 - \text{Скорость}).$$

$$F_{OUT} = F_{OSC} / [4 * (65536 - (RCAP2H, RCAP2L))].$$

Последовательный порт. У микроконтроллеров 8xC52 последовательный порт имеет функции определения ошибки кадра и автоматического распознавания адреса.

- **Определение ошибки кадра.** Эта функция реализуется через проверку уровня сигнала во время интервала времени, когда на линии должен проходить стоп-бит. Проверка выполняется при работе в режимах 1, 2, 3. Если стоп-бит в определенное время не обнаружен, устанавливается флаг ошибки кадра FE. Этот флаг делит с битом управления SM0 один и тот же разряд SCON.7. Бит SMOD0 (PCON.6) указывает при обращении на бит управления или на флаг ошибки кадра (см. выше формат регистра PCON). Флаг FE может проверяться после каждой операции приема. Сбрасываться этот флаг должен программно, прием правильного кадра не сбрасывает ранее установленного флага FE.
- **Автоматическое распознавание адреса.** При работе нескольких микропроцессоров по одному последовательному каналу ведущий может обратиться к ведомому по индивидуальному адресу, либо к группе ведомых по ширококвещательному адресу. Для хранения индивидуального адреса введен регистр спецфункций SADDR (адрес 0A9H), а для хранения маски – регистр SADEN (адрес 0B9H). Именно маска дает возможность адресовать в каждый момент одно или несколько устройств. Например:

Ведомый 1	
SADDR	11110001
SADEN	11111010
Адрес отклика	11110X0X
Ведомый 2	
SADDR	11110011
SADEN	11111001
Адрес отклика	11110XX1

В приведенном примере в адресе первого ведомого замаскирован бит 0, в то время как для выбора второго ведомого этот бит должен быть равен 1. Следовательно, для обращения к ведомому1 младший бит адреса должен быть равен 0 (например 11110000). Аналогично, бит1 = 0 для ведомого1 и замаскирован у ведомого2. Для обращения к ведомому2 бит1 адреса должен быть равен 1 (например 11110111). Для обращения к обоим ведомым одновременно могут быть использованы адреса, где бит0 = 1 и бит1 = 0 (широковещательными являются адреса 11110001 и 11110101).

Система прерываний. Включает шесть источников запросов – к исходной системе добавлено прерывание от Таймера 2, который имеет два флага (TF2 и T2EX). Это прерывание с вектором 002BH. Таблица приоритетов прерываний при внутреннем поллинге следующая:

Приоритеты прерываний 8052 при поллинге

Источник	Приоритет внутри уровня
Вход INT0#	Высший
Таймер 0	
Вход INT1#	
Таймер 1	
Последовательный порт	
Таймер 2, флаг TF2	Низший
Таймер 2, флаг T2EX	

Число уровней приоритетов увеличено до трех. Система прерываний 8x52 обслуживается регистрами IE, IP и IPH:

IE	EA	-	ET2	ES	ET1	EX1	ET0	EX0
IP	-	-	PT2	PS	PT1	PX1	PT0	PX0
IPH	-	-	PT2H	PSH	PT1H	PX1H	PT0H	PX0H

Имя бита	Номер бита	Функция
EA	IE.7	Запрещение запросов от всех источников, имеет место при EA = 0.
-	IE.6	Зарезервирован.
ET2	IE.5	Запрещение запроса от Таймера 2, имеет место при ET2 = 0.
ES	IE.4	Запрещение запроса от последовательного порта, имеет место при ES = 0.
ET1	IE.3	Запрещение запроса от Таймера 1, имеет место при ET1 = 0.
EX1	IE.2	Запрещение запроса по входу INT1#, имеет место при EX1 = 0.
ET0	IE.1	Запрещение запроса от Таймера 0, имеет место при ET0 = 0.
EX0	IE.0	Запрещение запроса по входу INT0#, имеет место при EX0 = 0.
-	IP.7	Зарезервирован.
-	IP.6	Зарезервирован.
PT2	IP.5	Определяет приоритет Таймера 2.
PS	IP.4	Определяет приоритет последовательного порта.
PT1	IP.3	Определяет приоритет Таймера 1.
PX1	IP.2	Определяет приоритет входа INT1#.
PT0	IP.1	Определяет приоритет Таймера 0.
PX0	IP.0	Определяет приоритет входа INT0#.
-	IPH.7	Зарезервирован.
-	IPH.6	Зарезервирован.
PT2H	IPH.5	Определяет приоритет Таймера 2, старший бит.
PSH	IPH.4	Определяет приоритет последовательного порта, старший бит.
PT1H	IPH.3	Определяет приоритет Таймера 1, старший бит.
PX1H	IPH.2	Определяет приоритет входа INT1, старший бит #.
PT0H	IPH.1	Определяет приоритет Таймера 0, старший бит.
PX0H	IPH.0	Определяет приоритет входа INT0#, старший бит.

Уровень приоритета конкретного источника определяется значением комбинации битов IPH.x и IP.x (x = 0..6) следующим образом:

IPH.x	IP.x	Приоритет
0	0	Уровень 0 (низший)
0	1	Уровень 1
1	0	Уровень 2
1	1	Уровень 3(высший)

Режим ONCE. Отладочный режим ONCE позволяет производить отладку и тестирование платы без удаления микроконтроллера (при запаянной микросхеме). Для перевода микроконтроллера в этот режим необходимо подать на вывод ALE сигнал логического «0» в процессе сброса и инициализации (сигнал PSEN# при этом имеет пассивный уровень «1») и удерживать его до снятия сигнала RESET.

В режиме ONCE линии порта P0 находятся в третьем состоянии. Линии других портов, а также сигналы ALE и PSEN находятся в состоянии логической «1». Это позволяет внешнему активному устройству управлять магистралью микропроцессорной системы.

Выход из режима ONCE производится посредством общего сброса.

2.12. Микроконтроллеры семейства AT89 фирмы Atmel

Микроконтроллеры с архитектурой MCS-51 фирмы Atmel – одно из первых семейств, все представители которого имеют версию с flash-памятью программ. Такой подход обеспечил популярность продукции фирмы Atmel среди разработчиков. Последовавшее снижение стоимости микроконтроллеров и выпуск семейства AVR, все представители которого также имеют flash-память и возможность внутрисистемного программирования, заставили такие фирмы, как Motorola, Philips, Texas Instruments, Microchip активизировать программы выпуска flash-микроконтроллеров.

Семейство микроконтроллеров фирмы Atmel с flash-памятью программ и ядром MCS-51 имеет маркировку «89». Это правило поддерживается другими производителями аналогичных микроконтроллеров – Philips, Temic, SST, ISSI, за исключением фирмы Winbond, выпускающей семейства W77 и W78. Номенклатура семейства AT89 приведена в следующей таблице.

В последней строке приведены параметры микроконтроллера, выпускаемого подразделением полупроводников фирмы Temic. Но, так как это подразделение в настоящее время приобретено фирмой Atmel и называется теперь Atmel Wireless and Microcontrollers, то можно включить микроконтроллер «51RD2» в семейство AT89.

Микроконтроллеры фирмы Atmel семейства AT89

Тип	Flash	RAM	Питание	Частота, МГц	Корпус	Комментарии
AT89C1051U	1K	128	3В, 5В	12, 24	DIP20, SOIC20	Iout,low = 20 mA
AT89C2051	2K	128	3В, 5В	12, 24	DIP20, SOIC20,	Iout,low = 20 mA
AT89C4051	4K	256	3В, 5В	12, 24	DIP20, SOIC20	Iout,low = 20 mA
AT89C51	4K	128	5В	24,33	DIP40, PLCC44, TQFP44	
AT89LV51	4K	128	3В, 5В	12	DIP40, PLCC44, TQFP44	
AT89C52	8K	256	5В	24,33	DIP40, PLCC44, TQFP44	
AT89LV52	8K	256	3В, 5В	12	DIP40, PLCC44, TQFP44	
AT89C55WD	20K	256	5В	24,33	DIP40, PLCC44, TQFP44	WD, 2 DPTR
AT89LV55	20K	256	3В, 5В	12	PLCC44, TQFP44	
AT89S8252	8K	256	5В	24	DIP40, PLCC44, TQFP44	ISP, SPI, WD, 2K EEPROM, 2 DPTR
AT89LS8252	8K	256	3В, 5В	12	DIP40, PLCC44, TQFP44	ISP, SPI, 2K EEPROM, 2 DPTR
AT89S53	12K	256	5В	24	DIP40, PLCC44, TQFP44	ISP, SPI, 2 DPTR
AT89LS53	12K	256	3В, 5В	12	DIP40, PLCC44, TQFP44	ISP, SPI, 2 DPTR
AT89S4D12	4K	128	3В	12	SOIC32	128K DataFlash, ISP, SPI
AT89C51RC	32K	256	5В	24,33	DIP40, PLCC44, TQFP44	256 bytes XRAM, 2 DPTR
AT89C51RD2	64K	256	3В, 5В	40	DIP40, PLCC44, VQFP44, VQFP64, PLCC68	ISP, Boot Flash, 1K XRAM, 2K EEPROM, 2 DPTR, PCA

В таблице использованы следующие обозначения:

Iout,low – нагрузочная способность выходов в состоянии логического «нуля».

ISP (In-System Programmability) – возможность программирования в системе.

Boot Flash – наличие области flash-памяти для размещения пользовательского монитора-загрузчика памяти.

XRAM – дополнительная RAM на кристалле, доступная как внешняя память данных с использованием команд MOVX.

WD – Watchdog, сторожевой таймер.

PCA (Programmable Counter Array) – многорежимный программируемый таймер.

Количество циклов перепрограммирования для микроконтроллеров семейства AT89 составляет не менее 1000.

По маркировке микроконтроллеров семейства AT89 можно определить ряд параметров кристалла.

Буква «С» после имени семейства (AT89C) указывает, что кристалл выполнен по CMOS технологии и параметры соответствуют аналогичным микросхемам других фирм (Intel 80C51, Philips 87C52 и т.д.).

Обозначение AT89LV отличает версии кристаллов с пониженным напряжением питания (2.7В).

Маркировку AT89S имеют микроконтроллеры, имеющие режим внутрисистемного программирования (ISP) через интерфейс SPI.

Микроконтроллеры AT89LS аналогичны AT89S, но имеют пониженное напряжение питания (2.7В).

Существует два основных ядра микроконтроллеров семейств MCS-51:

- C51, «родоначальник», стандартное ядро, включающее 128 байт RAM, 2 таймера-счетчика;
- C52, модифицированное ядро, имеет 256 байт RAM и 3 таймера-счетчика.

Маркировка C55 (или C54, C58 у других фирм) указывает на увеличенный объем памяти программ – от 16 до 32К.

Микроконтроллеры AT89C51RC и T89C51RD2 являются представителями серии «C51Rx», введенной фирмами Intel и Philips. Отличительные особенности этой серии – наличие дополнительной памяти данных на кристалле и второго указателя данных DPTR.

Рассмотрим подробнее особенности микроконтроллеров семейства AT89.

2.12.1. Программирование flash-памяти программ

В микроконтроллерах AT89C предусмотрен параллельный режим программирования flash-памяти. Для записи информации требуется подать напряжение программирования +12В, а для управления используются почти все линии портов. Поэтому процесс программирования flash-памяти выполняется на специализированных программаторах ПЗУ.

В микроконтроллерах AT89S кроме параллельного режима программирования имеется последовательный режим, через интерфейс SPI. При использовании последовательного режима не требуется внешнего напряжения программирования +12В, так как на кристалле предусмотрен преобразователь напряжения. Для загрузки flash-памяти необходимо управлять четырьмя линиями – RESET, SCK (P1.7), MOSI (P1.5), MISO (P1.6). Если при разработке проекта на базе микроконтроллеров AT89Sxx предусмотреть возможность подключения к перечисленным линиям внешнего программирующего устройства, то легко обеспечивается возможность об-

новления встроенного программного обеспечения (upgrade). Этот механизм называется внутрисистемным программированием – ISP (in-system programmability). Простой способ реализации цепей ISP в схеме показан на рис. 2.17.

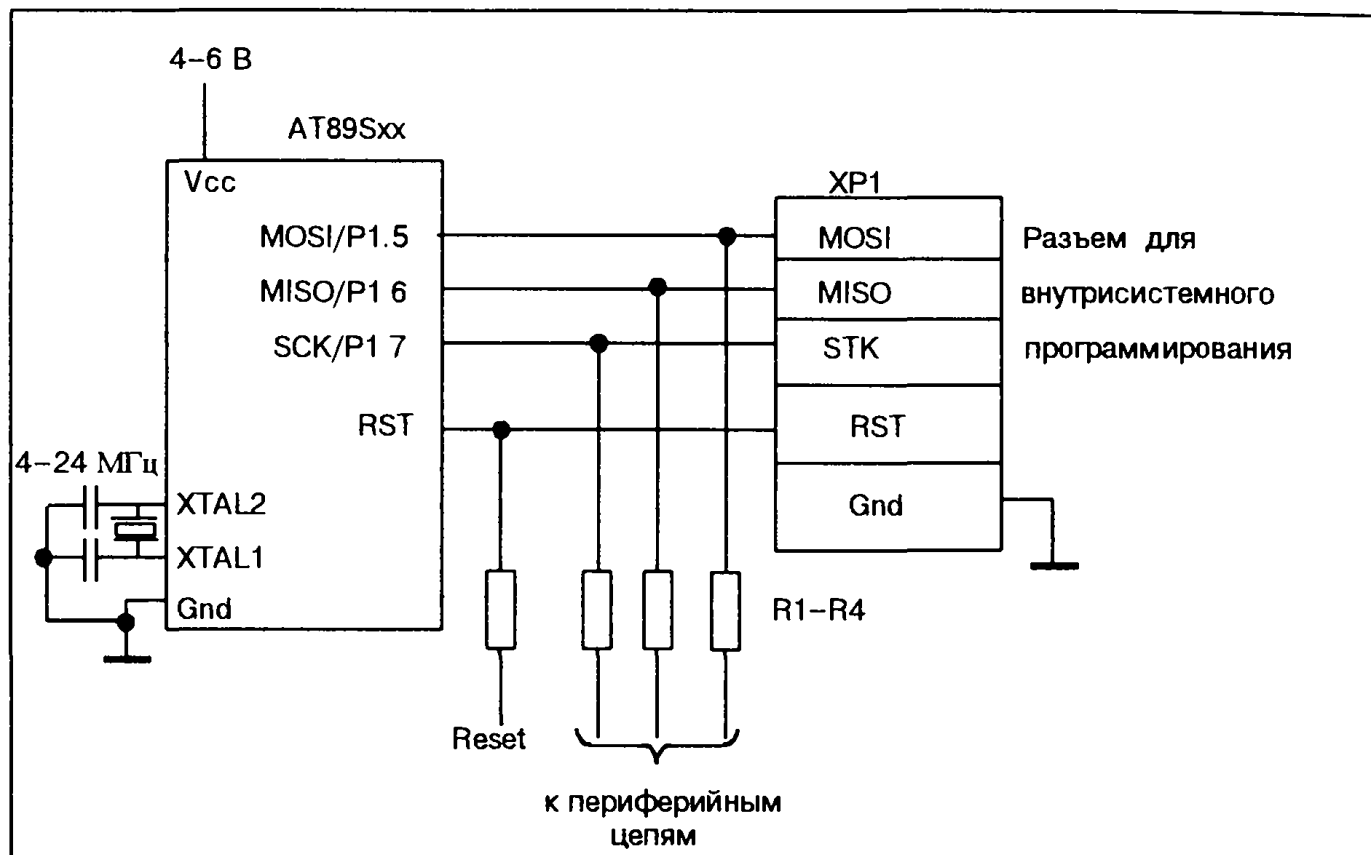


Рис. 2.17. Способ реализации цепей ISP

Программирующее устройство подключается непосредственно к микроконтроллеру через соединитель XP1. Для устранения конфликтов между сигналами программирования и периферийными устройствами, подключенными к тем же портам, установлены резисторы R1-R4. Алгоритм внутрисистемного программирования описан в параграфе 2.12.7.

Так как в интерфейсе типа SPI импульсы синхронизации формирует ведущее устройство (в нашем случае это внешнее программирующее устройство), а частота импульсов не ограничена снизу, то для программирования можно использовать, например, порт LPT компьютера. При этом временная диаграмма обмена формируется программно. В лаборатории «Микропроцессорные системы» МИФИ для этой цели используется кабель ByteBlaster, специфицированный фирмой Altera для внутрисистемной загрузки конфигурации ПЛИС.

2.12.2. Микроконтроллеры AT89S

Отличительными особенностями этой серии являются:

- EEPROM память данных (только у AT89S8252);

- последовательный интерфейс SPI;
- сторожевой таймер (Watchdog);
- два регистра-указателя данных DPTR.

Карта адресного пространства регистров специальных функций микроконтроллеров AT89S приведена на рис. 2.18.

F8								FF
F0	B 00000000							F7
E8								EF
E0	ACC 00000000							E7
D8								DF
D0	PSW 00000000					SPCH 000001xx		D7
C8	T2CON 00000000	T2MOD 00000000	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		CF
C0								C7
B8	IP x0000000							BF
B0	P3 11111111							B7
A8	IE 00000000		SPSH 00xxxxxx					AF
A0	P2 11111111							A7
98	SCON 00000000	SBUF xxxxxxx						9F
90	P1 11111111						WMCON 00000010	97
88	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8F
80	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000			SPDR xxxxxxx	PCON 00x0000

Рис. 2.18. Регистры специальных функций микроконтроллеров AT89Sxx

Управление доступом к EEPROM, сторожевым таймером и переключение указателей DPTR производится через регистр WMCON (адрес 096h). Биты этого регистра не входят в битовое адресное пространство микроконтроллера. Для установки битов можно использовать инструкцию `ORL WMCON, #OrMask`.

Если имя WMCON не поддерживается используемым компилятором, то можно объявить его как константу:

```
WMCON .equ 096h,
```

где OrMask – это константа, содержащая «1» в тех битах, которые нужно установить.

Аналогично, для сброса битов можно использовать инструкцию

ANL WMCON, #AndMask,

где AndMask – это константа, содержащая «0» в тех битах, которые нужно сбросить.

WMCON **PS2** **PS1** **PS0** **EEMWE** **EEMEN** **DPS** **WDTRST** **WDTEN**

Имя бита	Номер бита	Функция
PS2	WMCON.7	Установка периода срабатывания сторожевого таймера в диапазоне от 16 мс (по умолчанию) до 2048 мс.
PS1	WMCON.6	
PS0	WMCON.5	
EEMWE	WMCON.4	Бит разрешения записи в EEPROM. Перед записью нужно установить в "1". По окончании записи в EEPROM нужно установить в "0".
EEMEN	WMCON.3	Бит разрешения доступа к EEPROM. Когда бит установлен в "1" инструкции MOVX, использующие DPTR, обеспечивают доступ к EEPROM. Если бит равен "0", то производится обращение к внешней памяти данных.
DPS	WMCON.2	Выбор регистра DPTR. При DPS=0 в качестве DPTR используется пара DP0H:DP0L, а при DPS=1 пара DP1H:DP1L.
WDTRST RDY/BSY	WMCON.1	Бит сброса сторожевого таймера и флаг занятости EEPROM. При записи "1" в этот бит генерируется импульс сброса Watchdog и затем бит автоматически сбрасывается в "0". При чтении обращение идет к флагу, отражающему состояние EEPROM: если "0", то процесс записи байта данных не закончен, если "1", то EEPROM готова к записи.
WDTEN	WMCON.0	Бит разрешения сторожевого таймера. При WDTEN = 1 таймер Watchdog запущен и будет генерировать импульс Reset микроконтроллера с периодом, заданным битами PS2 - PS0.

В микроконтроллере AT89S53 память EEPROM отсутствует, поэтому биты 4 и 5 зарезервированы, а регистр называется WCON.

Память EEPROM. Этот вид энергонезависимой памяти отличается от flash-памяти тем, что допускает перезапись одного байта без стирания информации в остальных. Ячейка EEPROM занимает значительно больше места на кристалле, чем ячейка flash-памяти, поэтому стоимость EEPROM выше. Количество циклов перепрограммирования также выше – не менее 100000.

В микроконтроллерах AT89S8252 блок памяти данных EEPROM доступен так же, как внешняя память данных, инструкциями MOVX A,@DPTR и MOVX @DPTR,A. Для доступа к памяти EEPROM нужно установить бит EEMEN регистра WMCON. Для записи данных в EEPROM нужно дополнительно установить бит EEMWE регистра WMCON. В процессе записи бит RDY/BSY регистра WMCON находится в состоянии «0», по окончании записи этот бит переходит в состояние «1». Кроме того, окончание записи байта можно проверять циклическим чтением.

ем байта: пока запись не закончена старший бит данных читается в инверсном виде.

Пример процедуры записи байта в EEPROM:

```

mov A,#data      ;данные для записи
mov DPTR,#Address ;адрес в памяти
orl WMCON,#18h    ;установим биты разрешения EEMEN и EEMWE
movx @DPTR,A      ;инициируем запись
Loop:             ;цикл ожидания конца записи байта
mov A,WMCON       ;читаем содержимое регистра управления
jnb ACC.1, Loop    ;проверяем бит RDY/BSY
anl WMCON,#E7h    ;запись закончена, сбросим EEMEN и EEMWE

```

Watchdog – сторожевой таймер. Watchdog приобретает все большую популярность у производителей микроконтроллеров. Он предназначен для формирования сброса микроконтроллера на аппаратном уровне, если прикладная программа выполняет неконтролируемые действия, например, «зависла». Время срабатывания Watchdog устанавливается программно в пределах 16..2048 мс. В течение заданного времени программа должна выполнить процедуру сброса таймера Watchdog. Управление сторожевым таймером производится через регистр WMCON.

Для запуска Watchdog нужно установить бит WDTEN в «1». Период срабатывания таймера задается битами PS2-PS0:

PS2	PS1	PS0	Период
0	0	0	16 мс
0	0	1	32 мс
0	1	0	64 мс
0	1	1	128 мс
1	0	0	256 мс
1	0	1	512 мс
1	1	0	1024 мс
1	1	1	2048 мс

Для сброса сторожевого таймера следует записать «1» в бит WDTRST. Так как данный бит регистра WMCON разделяет функции с битом занятости EEPROM, то проверять его для оценки состояния Watchdog не имеет смысла.

Два регистра-указателя данных DPTR. Стандартное ядро MCS-51 имеет один 16-разрядный указатель данных для доступа к полному пространству памяти программ и данных. При использовании памяти данных объемом более 256 байт это единственный указатель, обеспечивающий доступ к памяти. В результате, при обработке сложных структур данных его требуется часто перезагружать и, что еще более неудобно, сохранять текущее состояние перед загрузкой. Введение второго 16-разрядного указателя облегчает программирование и позволяет ускорить обработку данных.

Второй регистр-указатель данных расположен по адресам 84h и 85h. В каждый момент времени доступен только один из двух регистров, поэтому все инструкции, работающие с DPTR, сохраняют свои функции. Бит DPS

регистра WMCON определяет, какой из двух указателей активен в данный момент. Если $DPS = 0$, то активен DP0, а если $DPS = 1$, то активен DP1. Таким образом, обеспечивается значительное повышение эффективности программы при работе одновременно с двумя массивами данных.

Последовательный интерфейс SPI. Serial Peripheral Interface (SPI) – это синхронный последовательный периферийный интерфейс. В микроконтроллерах AT89S максимальная скорость обмена равна 1,5 Мбит/сек. Для соединения используются 3 линии:

- SCK (P1.7) – Serial Clock, линия синхронизации; тактовые импульсы задает ведущее (Master) устройство
- MOSI (P1.5) – Master Output Slave Input, линия данных; является выходной у ведущего и входной у ведомого (Slave);
- MISO (P1.6) – Master Input Slave Output, линия данных; является входной у ведущего и выходной у ведомого.
- SS (P1.4) – Slave Select, линия выбора устройства определена для ведомого. Активный («0») уровень подключает устройство к магистрали SPI. Это позволяет объединять несколько ведомых устройств параллельно.

Подробно принцип обмена данными и временные диаграммы интерфейса SPI приведены в параграфах 3.6 и 4.7. Ниже описаны регистры управления SPI микроконтроллеров AT89S.

Регистр управления – SPI Control Register, размещен по адресу 0D5h.

SPCR	SPIE	SPE0	DORD	MSTR	CPOL	CPHA	SPR0	SPR1
-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------

Имя бита	Номер бита	Функция															
SPIE	SPCR.7	Бит разрешения прерываний SPI. Должен быть установлен в «1» совместно с битом ES регистра разрешения прерываний IE.															
SPE	SPCR.6	Бит разрешения SPI. При SPE=1 линии интерфейса подключены к порту P1 как альтернативные функции порта.															
DORD	SPCR.5	Порядок следования данных. При DORD=1 первым передается младший бит, при DORD=0 первым передается старший бит.															
MSTR	SPCR.4	Выбор ведущего. При MSTR=1 устройство является ведущим, при MSTR=0 – ведомым.															
CPOL	SPCR.3	Выбор полярности синхросигнала. При CPOL=1 линия SCK=1 в пассивном состоянии, при CPOL=0 линия SCK=0 в пассивном состоянии.															
CPHA	SPCR.2	Выбор фазы синхросигнала. Определяет активный фронт сигнала SCK.															
SPR1 SPR0	SPCR.1 SPCR.0	Выбор скорости передачи. Биты задают коэффициент деления тактовой частоты микроконтроллера: <table> <tr> <td>SPR1</td><td>SPR0</td><td>K</td></tr> <tr> <td>0</td><td>0</td><td>4</td></tr> <tr> <td>0</td><td>1</td><td>16</td></tr> <tr> <td>1</td><td>0</td><td>64</td></tr> <tr> <td>1</td><td>1</td><td>128</td></tr> </table>	SPR1	SPR0	K	0	0	4	0	1	16	1	0	64	1	1	128
SPR1	SPR0	K															
0	0	4															
0	1	16															
1	0	64															
1	1	128															

Регистр состояния – SPI Status Register, размещен по адресу 0AAh.

SPSR	SPIF	WCOL	-	-	-	-	-	-
-------------	-------------	-------------	---	---	---	---	---	---

Имя бита	Номер бита	Функция
SPIF	SPSR.7	Бит запроса прерывания SPI. Процедура обработки прерывания вызывается, если установлены биты SPCR.SPIE и IE.ES. Бит очищается автоматически, если производится чтение SPCR при установленном WCOL, а затем производится обращение к регистру SPDR.
WCOL	SPSR.6	Бит ошибки записи. Устанавливается, если попытка записи в регистр SPDR производилась в процессе передачи байта данных. Бит очищается автоматически, если производится чтение SPCR при установленном бите SPIF, а затем производится обращение к регистру SPDR.
-	SPSR.5	Зарезервирован
-	SPSR.4	Зарезервирован
-	SPSR.3	Зарезервирован
-	SPSR.2	Зарезервирован
-	SPSR.1	Зарезервирован
-	SPSR.0	Зарезервирован

Регистр данных – SPDR, размещен по адресу 086h. Запись данных в регистр, если устройство определено как ведущее (MSTR = 1), запускает цикл передачи байта. Если устройство определено как ведомое (MSTR = 0), то запись в регистр подготавливает данные для последующего чтения ведущим устройством.

2.12.3. Микроконтроллеры AT89C51RC/55WD

Особенностями этих микроконтроллеров являются:

- дополнительная память данных – только в AT89C51RC;
- сторожевой таймер (Watchdog);
- два регистра-указателя данных DPTR.

Ниже описаны регистры специальных функций, вновь включенные в данные кристаллы.

Вспомогательный регистр AUXR1 размещен по адресу 0A2h.

AUXR1	-	-	-	-	-	-	-	DPS
--------------	---	---	---	---	---	---	---	------------

Имя бита	Номер бита	Функция
-	AUXR1.7	Зарезервирован.
-	AUXR1.6	Зарезервирован.
-	AUXR1.5	Зарезервирован.
-	AUXR1.4	Зарезервирован.
-	AUXR1.3	Зарезервирован.
-	AUXR1.2	Зарезервирован.
-	AUXR1.1	Зарезервирован.
DPS	AUXR1.0	Выбор регистра DPTR. При DPS=0 в качестве DPTR используется пара DP0H:DP0L, а при DPS=1 пара DP1H:DP1L.

Вспомогательный регистр AUXR размещен по адресу 08Eh.

AUXR	-	-	-	WDIDLE	DISRTO	-	EXTRAM	DISALE
------	---	---	---	--------	--------	---	--------	--------

Имя бита	Номер бита	Функция
-	AUXR.7	Зарезервирован.
-	AUXR.6	Зарезервирован.
-	AUXR.5	Зарезервирован.
WDIDLE	AUXR.4	Запрещение работы сторожевого таймера в режиме Idle. Если WDIDLE=1, то в режиме Idle сторожевой таймер остановлен.
DISRTO	AUXR.3	Запрещение формирования сигнала RST. Если DISRTO=0, то при переполнении сторожевого таймера на входе RST формируется импульс сброса, если DISRTO=1, то RST является только входом.
-	AUXR.2	Зарезервирован.
EXTRAM только в 51RC	AUXR.1	Разрешение доступа к внешней памяти данных. Если EXTRAM=0, то инструкции MOVX обращаются к дополнительной памяти данных на кристалле, если EXTRAM=1, то обращение идет к внешней памяти данных, как в стандартном 89C51.
DISALE	AUXR.0	Запрещение выдачи сигнала ALE. Если бит установлен в «1», то ALE выдается только при выполнении инструкций MOVX и MOVC.

Сторожевой таймер в микроконтроллерах AT89C51RC и AT89C55WD. Программирование сторожевого таймера в данном микроконтроллере отличается от микроконтроллеров AT89Sxx. Для доступа к сторожевому таймеру введен регистр WDTRST с адресом 0A6h. Это регистр служит только для записи, чтение по данному адресу не имеет смысла.

Существует единственный способ обращения к WDTRST. Следует произвести два последовательных цикла записи в WDTRST кодов 01Eh и 0E1h. Типовая процедура имеет вид:

```

clr  IE          ;запретим прерывания, чтобы обеспечить
mov  WDTRST,#01Eh ;непрерывность двух циклов записи
mov  WDTRST,#0E1h ;в регистр
setb IE

```

Данная последовательность действий обеспечивает сброс сторожевого таймера. Эта же последовательность, выполненная впервые после сброса микроконтроллера, запускает сторожевой таймер.

Период срабатывания сторожевого таймера не программируется и равен 16383 машинных цикла микроконтроллера. Чтобы обеспечить нормальную работу программы, использующей режим Idle – пассивного ожидания асинхронного события – предусмотрена возможность запрещения работы Watchdog в этом режиме, установкой бита WDIDLE в регистре AUXR.

Дополнительная память данных (XRAM) в микроконтроллере AT89C51RC доступна так же, как внешняя память данных, используя инструкции MOVX. Для установки режима доступа к XRAM нужно сбросить бит EXTRAM в регистре AUXR (состояние после сброса микроконтроллера).

Два регистра-указателя данных DPTR в микроконтроллерах AT89C51RC и AT89C55WD функционируют так же, как в AT89Sxx.

Карта адресного пространства регистров специальных функций микроконтроллеров AT89C51RC и AT89C55WD приведена на рис. 2.19.

F8								FF
F0	B 00000000							F7
E8								EF
E0	ACC 00000000							E7
D8								DF
D0	PSW 00000000							D7
C8	T2CON 00000000	T2MOD 00000000	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		CF
C0								C7
B8	IP x0000000							BF
B0	P3 11111111							B7
A8	IE 00000000							AF
A0	P2 11111111		AUXR1 xxxxxxx0			WDTRST xxxxxxx0		A7
98	SCON 00000000	SBUF xxxxxxx						9F
90	P1 11111111							97
88	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR2 xxx00xxx	8F
80	P0 11111111	SP 00001111	DPL 00000000	DPH 00000000			PCON 00x0000	87

Рис. 2.19. Регистры специальных функций микроконтроллеров AT89C51RC, AT89C55WD

2.12.4. Микроконтроллер T89C51RD2

Рассмотрим кратко особенности микроконтроллера T89C51RD2.

EEPROM память данных доступна с помощью инструкций MOVX. Запись может производиться блоками от 1 до 64 байт. Начало и конец цикла записи контролируется через управляющий регистр.

Дополнительная память данных также доступна с помощью инструкций MOVX. Переключение адресных пространств производится аналогично AT89C51RC, адрес и функция управляющего бита совпадают.

Сторожевой таймер (Watchdog) управляется так же, как в AT89C51RC, но есть возможность программно установить период срабатывания таймера.

Размещение регистров-указателей данных **DPTR** и управление ими такие же, как в AT89C51RC.

Многорежимный программируемый таймер **PCA** аналогичен описанному в [1].

Имеется режим **повышенной производительности**, в котором машинный цикл содержит не 12 тактов, как в стандартном ядре MCS-51, а 6 тактов. Переключение между режимами производится программно. Режим называется «X2», на его поддержку указывает цифра 2 в маркировке микроконтроллера.

Режим **внутрисистемного программирования** отличается от соответствующего режима микроконтроллеров AT89Sxx. Программирование производится через интерфейс RS232 при стандартном напряжении питания +3 В или +5 В. Микроконтроллер имеет встроенную программу загрузки памяти. Кроме того имеется возможность программирования собственной памяти программ. Запись производится по секторам, при этом цикл стирания сектора выполняется автоматически. Имеется возможность использовать пользовательскую программу-загрузчик Flash памяти программ, при этом можно установить новый вектор Reset, отличный от нулевого.

2.12.5. Микроконтроллеры с уменьшенным числом выводов – AT89C1051/2051/4051

Вариант микроконтроллера с ядром MCS-51 в корпусе с 20 выводами был впервые введен фирмой Atmel. Основные особенности этой серии кристаллов:

- повышенная нагрузочная способность выходов в состоянии логического нуля; обеспечивается ток до 20мА, что позволяет без дополнительных буферных схем управлять светодиодными индикаторами и реле;
- наличие аналогового компаратора.

В этой серии микроконтроллеров используется стандартное ядро 8051, отсутствуют только порты P0 и P2.

Вид корпуса микроконтроллеров AT89Cx051 и функции выводов приведены на рис. 2.20.

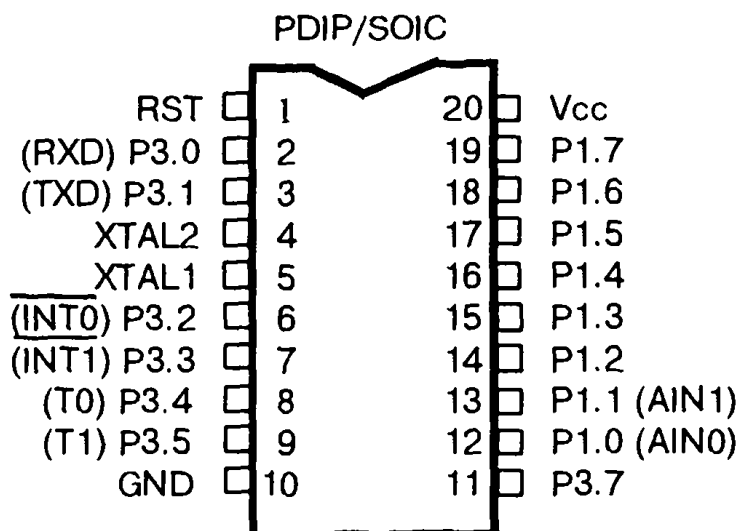


Рис. 2.20. Корпус и функции выводов микроконтроллеров AT89Cx051

2.12.6. Алгоритм последовательной загрузки flash-памяти микроконтроллеров AT89Sxx

Алгоритм последовательной загрузки flash-памяти программ этих микроконтроллеров состоит из следующей последовательности действий:

- подать питание $V_{cc}=2.7\text{ В} - 5\text{ В}$. Обеспечить тактовую частоту F_{osc} в диапазоне от 3 до 12 МГц. Установить $RST = 1$. Выдержать паузу 10 мс;
- послать команду «разрешение программирования» по линиям SCK, MOSI. Частота передачи должна быть меньше, чем $F_{osc}/40$;
- память программ и данных программируется по одному байту за цикл. Команда, адрес и данные передаются каждый раз в виде 3-байтной посылки. Типовое время записи байта равно 2.5 мс при $V_{cc}=5\text{ В}$. Запись производится автоматически;
- ячейку памяти программ или данных можно прочитать для проверки завершения цикла записи. Допускается циклический опрос данных после посылки команды записи;
- после окончания записи перевод RST в состояние «0» запускает программу на выполнение;

При формировании временной диаграммы последовательной загрузки flash-памяти следует придерживаться следующих правил.

1. Тактовый импульс SCK имеет форму 0-1-0.
2. Входные данные (MOSI) защелкиваются по переднему фронту SCK (0-1). Первым следует старший бит.

3. Выходные данные (MISO) обновляются по заднему фронту SCK (1 – 0). Первым следует старший бит. Чтение первого бита нужно производить до завершения импульса SCK.

Таблица команд последовательной загрузки flash-памяти программ имеет следующий вид.

Команды последовательной загрузки микроконтроллеров AT89Sxx

Команда	Байт 1	Байт 2	Байт 3	Описание
Разрешение программирования	1010 1100	0101 0011	xxxxxxx	Переводит кристалл в режим программирования после подачи RST = 1
Стирание памяти	1010 1100	xxxx x100	xxxxxxx	Стирание Flash и EEPROM
Чтение памяти программ	aaaa a001 для S53: aaaa aa01	мл. байт адреса	xxxxxxx	Старшие биты адреса передаются в байте 1. В цикле передачи 3-го байта производится чтение данных
Запись памяти программ	aaaa a010 для S53: aaaa aa10	мл. байт адреса	данные	Старшие биты адреса передаются в байте 1
Чтение памяти данных (только S8252)	00aa a101	мл. байт адреса	xxxxxxx	Старшие биты адреса передаются в байте 1. В цикле передачи 3-го байта производится чтение данных
Запись памяти данных (только S8252)	00aa a110	мл. байт адреса	данные	Старшие биты адреса передаются в байте 1
Запись битов защиты	1010 1100	BBBx x111	xxxxxxx	"0" в поле "B" устанавливает бит защиты.

Маски для установки битов защиты:

- 0111 1111 – запрещено выполнение инструкции MOVC, если инструкция размещается во внешней памяти программ, а читает код из внутренней памяти; состояние входа EA защелкивается по фронту сигнала RST, запрещена дальнейшая запись в память программ.
- 0011 1111 – то же, что в предыдущем пункте, дополнительно запрещено чтение памяти программ и данных.
- 0001 1111 – то же, что в предыдущем пункте, дополнительно запрещено выполнение программы, находящейся во внешней памяти.

2.13. Другие микроконтроллеры MCS-51 с flash-памятью программ

Микроконтроллеры с архитектурой (процессорным ядром) MCS-51 выпускают такие известные фирмы, как Philips, Siemens, Dallas Semiconductor, Winbond и ряд других.

Весьма интересны микроконтроллеры типа MCS-51, выпускаемые фирмой Winbond. Их основные характеристики приведены в следующей таблице.

Микроконтроллеры MCS-51 с Flash-памятью фирмы Winbond

Тип	Flash	RAM	Питание	Частота, МГц	Корпус	Комментарии
W78E51B	4K	128	5В	24, 40	DIP40, PLCC44, TQFP44	WD, Extra I/O port
W78LE51	4K	128	2.4В-5В	24	DIP40, PLCC44, TQFP44	WD, Extra I/O port
W78E52B	8K	256	5В	24, 40	DIP40, PLCC44, TQFP44	WD, Extra I/O port
W78LE52	8K	256	2.4В-5В	24	DIP40, PLCC44, TQFP44	WD, Extra I/O port
W78E54	16K	256	5В	24, 40	DIP40, PLCC44, TQFP44	WD, Extra I/O port
W78LE54	16K	256	2.4В-5В	24	DIP40, PLCC44, TQFP44	WD, Extra I/O port
W78E58	32K	256	5В	24, 40	DIP40, PLCC44, TQFP44	WD, Extra I/O port
W78LE58	32K	256	2.4В-5В	24	DIP40, PLCC44, TQFP44	WD, Extra I/O port
W78E516	64K	256	5В	24, 40	DIP40, PLCC44, TQFP44	WD, Extra I/O port, ISP, 256 XRAM
W78LE516	64K	256	2.4В-5В	24	DIP40, PLCC44, TQFP44	WD, Extra I/O port, ISP, 256 XRAM
W77E58	32K	256	5В	25, 40	DIP40, PLCC44, TQFP44	Turbo, 2 UART, 2DPTR, WD, 1K XRAM, Extra I/O port
W77LE58	32K	256	2.4В-5В	25	DIP40, PLCC44, TQFP44	Turbo, 2 UART, 2DPTR, WD, 1K XRAM, Extra I/O port
W77E468F	32K	256	5В	25, 40	QFP100	Turbo, 2 UART, 2DPTR, WD, 1K XRAM, 52 I/Os

В таблице приведены характеристики микроконтроллеров с flash-памятью. Однако в семействе микроконтроллеров Winbond почти все представители имеют варианты с масочным ПЗУ, с маркировкой W78C и W78L. При этом версии с расширенным напряжением питания и масочным ПЗУ специфицированы для работы при $V_{cc} = 1,8$ В, в то время, как версии с Flash памятью имеют нижнюю границу 2,4 В.

Рассмотрим особенности микроконтроллеров Winbond.

В микроконтроллерах, имеющих корпуса типа PLCC44 и TQFP44, существует дополнительный порт ввода-вывода P4 (Extra I/O port),

включающий 4 бита. В этот порт введены 2 дополнительных входа внешних прерываний Int2 и Int3.

Серия W77xx отличается тем, что может обеспечивать значительное повышение производительности по сравнению с классическими микроконтроллерами MCS-51. Для этого введен **режим Turbo**, использующий 4-тактный машинный цикл вместо 12-тактного. Данный режим впервые был введен в микроконтроллерах фирмы Dallas Semiconductor (80C320, 87C520, 87C530, только ROM и EPROM версии). Кроме сжатой временной диаграммы, в режиме Turbo исключены холостые циклы обращения к памяти, что позволяет еще в 1,5 раза повысить производительность. Поскольку тактовая частота повышена до 40 МГц, можно утверждать, что микроконтроллеры серии W77 на сегодняшний день имеют наибольшую производительность среди Flash микроконтроллеров MCS-51.

Дополнительные преимущества серии W77 – наличие двух регистров DPTR, двух последовательных интерфейсов UART, дополнительной памяти данных на кристалле. При подключении внешней памяти данных можно программно регулировать длительность циклов чтения и записи, что позволяет совмещать высокоскоростной микроконтроллер и ОЗУ среднего быстродействия.

Внутрисистемное программирование в микроконтроллерах типа W78E516 отличается от описанных для AT89S и T89C51RD2. Для загрузки flash-памяти используется последовательный порт RS232, но система команд отличается от Atmel; при этом для пользовательской программы-загрузчика имеется дополнительный блок flash-памяти объемом 4 Кбайт.

Маркировка семейств W77 и W78 отличается от стандартной «89» вполне обоснованно. Фирма Winbond использует для репрограммируемой памяти программ собственную технологию, называя эту память MTP – Multiple-Time Programmable. Для пользователя это отличие означает, что программаторы, использовавшиеся ранее, могут не поддерживать данные микроконтроллеры, так как при перепрограммировании требуется устанавливать два вида напряжения программирования, временные диаграммы также изменены.

Фирма Philips в настоящее время лидирует по общему объему и номенклатуре выпускаемых микроконтроллеров MCS-51. В программе поставок Philips представлены микроконтроллеры MCS-51 с напряжениями питания от 1,8 В, в корпусах от 20 до 80 выводов, имеющие такие периферийные блоки, как 8- и 10-разрядные АЦП, 8-разрядные ЦАП, ШИМ, интерфейсы I²C и CAN. Большая часть микроконтроллеров Philips имеет версии с EPROM и с масочным ПЗУ. В таблице приведены только версии с Flash памятью.

Серия Rx+ объединяет возможности Flash технологии и широкую номенклатуру периферийных устройств на кристалле. В микроконтроллерах серии Rx+ внутрисистемное программирование Flash производится через

интерфейс RS232. Система команд для загрузки Flash совпадает с микроконтроллерами T89C51RD2, но, в отличие от них, для внутрисхемной загрузки требуется подать напряжение программирования +12 В.

Микроконтроллеры MCS-51 с Flash-памятью фирмы Philips

Тип	Flash	RAM	Vcc	Частота, МГц	Корпус	Комментарии
P89C51RA+	8K	256	5В	33	DIP40, PLCC44, TQFP44	WD, PCA, 2 DPTR
P89C51RB+	16K	256	5В	33	DIP40, PLCC44, TQFP44	WD, PCA, 2 DPTR
P89C51RC+	32K	256	5В	33	DIP40, PLCC44, TQFP44	WD, PCA, 2 DPTR, 256 XRAM, ISP
P89C51RD+	64K	256	5В	33	DIP40, PLCC44, TQFP44	WD, PCA, 2 DPTR, 768 XRAM, ISP
P89C138	16K	256	5В	3.5-40	DIP40, PLCC44, TQFP44	
P89C238	32K	256	5В	3.5-40	DIP40, PLCC44, TQFP44	
P89C738	64K	256	5В	3.5-16	DIP40, PLCC44, TQFP44	256 XRAM
P89CE558	32K	256	5В	3.5-16	QFP80	WD, 2 PWM, 10 bit ADC, I2C, 768 XRAM, ISP

Для решения задач, требующих повышенной производительности микроконтроллеров, фирма Philips выпустила **семейство 80C51XA**. Семейство совместимо с MCS-51 только по особенностям архитектуры и системе команд на уровне мнемоник. Совместимость можно рассматривать в том плане, что наработанное для MCS-51 программное обеспечение может быть перетранслировано для установки на платформу 51XA с минимальными изменениями. Для облегчения переноса программного обеспечения с платформы MCS-51 на 51XA существует специальная программа-транслятор исходного текста на ассемблере 51 в исходный текст для ассемблера 51XA. В целом, семейство 51XA имеет более развитые способы адресации и систему команд, другую топологию выводов корпуса и другую кодировку инструкций процессора.

Основные характеристики семейства 51XA:

- 16-разрядное АЛУ;
- 24-разрядное адресное пространство (16 Мбайт памяти) для памяти программ и данных;
- восемь 16-разрядных регистров, организованных в виде регистрового файла, т.е. каждый может выполнять функцию аккумулятора;
- расширенный набор инструкций и способов адресации;
- аппаратная поддержка мультизадачности;
- напряжение питания от 2,7 В.

Архитектура ХА обеспечивает увеличение производительности на два порядка по сравнению с архитектурой MCS-51 (по оценкам фирмы Philips).

Сравнительно недавно выпущенные flash-микроконтроллеры фирм ISSI и SST не имеют принципиальных отличий от рассмотренных выше. Фирмы ISSI и SST специализируются на выпуске быстродействующих микросхем памяти – ОЗУ, flash. Выпуск этими фирмами микроконтроллеров с ядром MCS-51 и flash-памятью свидетельствует о том, что потребности рынка этих микросхем не удовлетворены усилиями Atmel, Philips, Winbond.

Микроконтроллеры MCS-51 с Flash-памятью фирмы ISSI

Тип	Flash	RAM	Vcc	Частота, МГц	Корпус	Комментарии
IS89C51	4K	128	5B	12,24,40	DIP40, PLCC44, TQFP44	Стандартная архитектура
IS89C52	8K	256	5B	12,24,40	DIP40, PLCC44, TQFP44	Стандартная архитектура
IS89LV52A	8K	256	3B, 5B	24	DIP40, PLCC44, TQFP44	Стандартная архитектура
IS89C54A	16K	256	5B	40	DIP40, PLCC44, TQFP44	Стандартная архитектура
IS89C58A	32K	256	5B	40	DIP40, PLCC44, TQFP44	Стандартная архитектура
IS89C64A	64K	256	5B	40	DIP40, PLCC44, TQFP44	Стандартная архитектура

Микроконтроллеры MCS-51 с Flash-памятью фирмы SST

Тип	Flash	RAM	Vcc	Частота, МГц	Корпус	Комментарии
SST89C54	16K	256	5B	33	DIP40, PLCC44, TQFP44	ISP, 4K Boot
SST89C58	32K	256	5B	33	DIP40, PLCC44, TQFP44	ISP, 4K Boot
SST89C59	64K	256	5B	33	DIP40, PLCC44, TQFP44	ISP, 4K Boot, 256 XRAM

В целом можно сделать вывод, что несмотря на то, что микроконтроллеры типа MCS-51 уступают по производительности современным RISC-микроконтроллерам, эта архитектура вряд ли будет полностью вытеснена новыми семействами, так как для определенного класса задач имеет преимущество по соотношению цена/качество. Большое значение имеет наличие развитых инструментальных средств и опыт разработчиков.