

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

Новгородский государственный университет
имени Ярослава Мудрого

Гурьянов С.А.

МИКРОКОНТРОЛЛЕР INTEL 8051

Новгород

2012

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

Новгородский государственный университет
имени Ярослава Мудрого
Институт электронных и информационных систем

Кафедра радиосистем

Гурьянов С.А.
МИКРОКОНТРОЛЛЕР INTEL 8051

Курс лекций по дисциплине
"ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ И УПРАВЛЯЮЩИЕ СИСТЕМЫ"
для специальности 210302.65 "Радиотехника"
и направления 210300.68 "Радиотехника"
Системы и устройства передачи, приема и обработки сигналов

Новгород
2012

УДК 621.38

ББК

Гурьянов С.А. Микроконтроллер INTEL8051: Курс лекций / ФГБОУ «Новгородский государственный университет им. Ярослава Мудрого», Великий Новгород, 2012 г. – 89 с.

В учебном пособии рассмотрена структура микроконтроллера INTEL8051, система команд микроконтроллера, примеры программ, устройства ввода-вывода.

Учебное пособие отвечает новым образовательным стандартам и предназначено для подготовки специалистов по специальности 210302.65 "Радиотехника" и по направлению 210300.68 "Радиотехника" (Системы и устройства передачи, приема и обработки сигналов).

© Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
Новгородский государственный университет
имени Ярослава Мудрого, 2012

СОДЕРЖАНИЕ

	стр.
ВВЕДЕНИЕ	5
1. АРХИТЕКТУРА МИКРОКОНТРОЛЛЕРА I8051	8
2. СТРУКТУРА МИКРОКОНТРОЛЛЕРА I8051	13
2.1 Арифметико-логическое устройство	14
2.2 Организация памяти программ и данных	17
3. СИСТЕМА КОМАНД МИКРОКОНТРОЛЛЕРА I8051	22
3.1 Команды передачи данных	25
3.2 Арифметические команды	27
3.3 Логические команды с байтовыми переменными	28
3.4 Команды битового процессора	28
3.5 Команды ветвления и передачи управления	29
4. УСТРОЙСТВА ВВОДА/ВЫВОДА	31
4.1 Порты ввода-вывода	34
4.2 Таймер-счетчик	38
4.3 Последовательный интерфейс	44
4.4 Организация прерываний	51
5. БЛОК УПРАВЛЕНИЯ	57
6. НАЧАЛЬНАЯ УСТАНОВКА	62
7. ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРА I8051	65
8. ПРИМЕРЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ АССЕМБЛЕР	70
8.1 Программа преобразования системы счисления (2 -> 2/10)	70
8.2 Арифметические операции двойной точности	71
8.3 Вычисление синуса табличным методом	72
8.4 Передача данных последовательным кодом	73
8.5 Передача данных манчестерским кодом	73
8.6 Передача и прием данных через последовательный порт	74
8.7 Передача символьной строки через последовательный порт	76
8.8 Измерение длительности сигнала	77
8.9 Подсчет числа импульсов за заданный интервал времени	78
8.10 Измерение медленно изменяющегося сигнала	79
8.11 Реализация клавиатуры	80
СПИСОК ЛИТЕРАТУРЫ	81
ПРИЛОЖЕНИЕ: Система команд микроконтроллера I8051	82

ВВЕДЕНИЕ

В настоящее время в микропроцессорной технике выделился самостоятельный класс больших интегральных схем, называемых микроконтроллеры. Микроконтроллеры предназначены для построения "интеллектуального" оборудования различного назначения.

Архитектура микроконтроллеров - это результат эволюции архитектуры микропроцессоров и микропроцессорных систем. Данный процесс обусловлен стремлением существенно снизить аппаратные затраты и стоимость разрабатываемого оборудования. Эти цели достигаются путем повышения уровня интеграции больших интегральных схем.

Микроконтроллеры представляют собой приборы, конструктивно выполненные в виде одной большой интегральной схемы и включающие в себя все устройства, необходимые для реализации цифровой системы управления минимальной конфигурации: процессор, оперативное запоминающее устройство данных, постоянное запоминающее устройство команд, внутренний генератор тактовых сигналов, а также программируемые периферийные схемы для связи с внешними устройствами.

Использование микроконтроллеров в системах управления и измерения обеспечивает исключительно высокие показатели эффективности при достаточно низкой стоимости. Низкая стоимость объясняется возможностью построения многих систем на одной большой интегральной схеме. Эти показатели говорят о том, что в ближайшее время нет альтернативной элементной базы для построения управляющих, регулирующих, измерительных и формирующих систем.

Микроконтроллеры обладают значительными функционально-логическими возможностями и представляют собой эффективное средство автоматизации на основе применения цифровых средств и методов обработки данных и цифрового управления разнообразных объектов и процессов.

Часто микроконтроллеры называют однокристальными микро-ЭВМ. Это обосновывается тем, что такие микросхемы имеют ограниченный объем памяти для построения значительных контрольно-измерительных и вычислительных комплексов; логическое разделение адресного пространства памяти программ (ПЗУ) и памяти данных (ОЗУ); несколько упрощенную и ориентированную на задачи управления систему команд, в некоторой степени примитивные методы адресации команд и данных.

Специфическая организация ввода/вывода информации, структурная организация, набор команд микроконтроллеров лучше всего приспособлены для решения задач управления и регулирования в приборах, устройствах и системах автоматики, а не для решения задач обработки больших потоков данных с последующими вычислительными процедурами.

Начало развитию семейства однокристальных микроконтроллера положил выпуск фирмой INTEL микросхемы 8048, которая вначале восьмидесятых годов была заменена микросхемой 8051, положившей начало семейству MCS-51.

Этот микроконтроллер был чрезвычайно популярен. В настоящее время устарел и вытеснен более современными устройствами с 8051-совместимыми ядрами, производимыми более чем 20 независимыми производителями, такими как Atmel, Maxim IC (дочерняя компания Dallas Semiconductor), NXP (панее Philips Semiconductor), Winbond, Silicon Laboratories, Texas Instruments и Cypress Semiconductor.

При сохранении основной структуры 8-разрядного арифметико-логического устройства в системе команд произошли изменения, позволившие значительно повысить производительность микроконтроллера. Применение огромного количества битово-ориентированных операций позволило говорить о реализации битового процессора. Увеличилось количество команд пересылки без участия аккумулятора, были введены команды умножения, деления и новые команды вычитания. Появились усовершенствования при работе со стеком.

Изменения в структуре временного цикла микроконтроллера привели к значительному ускорению работы с внешней памятью программ и данных. Было расширено адресное пространство внутренней памяти программ и увеличено пространство внешней памяти программ и данных до 64 Кбайт. 16-битные регистры счетчика команд и указателя данных позволили напрямую обращаться ко всему диапазону адресов, что дало возможность реализации быстрой обработки больших массивов данных. Все программно-доступные узлы микроконтроллера сведены в специальную область памяти данных (SFR - Special Function Register), что позволило обращаться к ним как к обычным ячейкам резидентного ОЗУ.

Для расширения функциональных возможностей периферийных систем были введены:

- два 16-разрядных многофункциональных таймера/счетчика;

- последовательный порт с синхронным и асинхронным режимом передачи;
- 2-уровневая система прерываний;
- четыре параллельных порта ввода-вывода.

При сохранении потребления кристаллов была значительно увеличена максимальная тактовая частота микроконтроллеров до 12-16 МГц. Дальнейшее применение более совершенной и низко потребляющей CMOS технологии позволило также снизить потребление кристаллов микроконтроллера.

1. АРХИТЕКТУРА МИКРОКОНТРОЛЛЕРА I8051

Микроконтроллер I8051 является первым микроконтроллером семейства MCS-51. Данный микроконтроллер имеет в своем составе следующие аппаратные средства:

- 8-разрядный центральный процессор, в состав которого входят однокбайтное арифметико-логическое устройство и схема аппаратной реализации команд умножения и деления;
- постоянное запоминающее устройство программ емкостью 4 Кбайта,
- оперативное запоминающее устройство данных емкостью 128 байт, используемое для организации регистровых банков, стеков и хранения пользовательских данных;
- не менее 32-х двунаправленных интерфейсных линий, индивидуально программируемых на ввод или вывод информации;
- два 16-битных таймера/счетчика, используемых для подсчета внешних событий, организации временных задержек и тактирования коммуникационного порта;
- асинхронный канал дуплексного последовательного ввода/вывода информации со скоростью до 375 кбит/сек,
- блок двухуровневого векторного прерывания от пяти источников,
- синхрогенератор, схему синхронизации и управления.

Структурная схема микроконтроллера I8051 представлена на рис.1.1. Структура микроконтроллеров серии I8051 и их система команд позволяет в случае необходимости расширить функционально-логические возможности контроллеров. С использованием внешних дополнительных БИС постоянной и оперативной памяти адресное пространство микроконтроллеров может быть расширено до 64 Кбайт, а путем подключения различных интерфейсных БИС число связей микроконтроллеров с объектом управления может быть практически увеличено без ограничений.

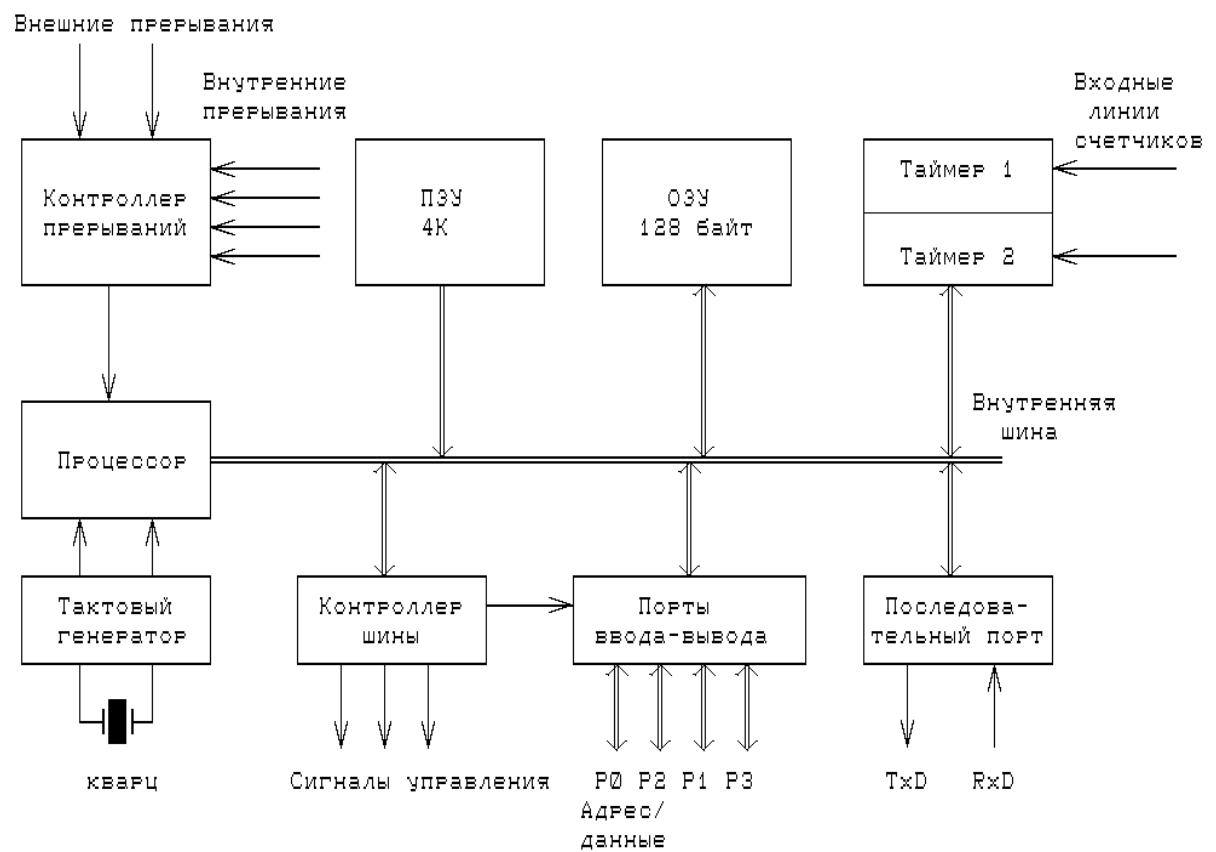


Рис. 1.1. Обобщенная структурная схема микроконтроллера I8051.

На рис.1.2 представлено условное графическое обозначение микроконтроллера I8051, а назначение выводов микроконтроллера дано в таблице 1.2.

Микроконтроллеры серии I8051 требуют одного источника питания напряжением +5В, рассеивают мощность около 1,5 Вт и работают в диапазоне температур от 0 до 70 С. По входам и выходам микроконтроллеры электрически совместимы с интегральными схемами ТТЛ. Конструктивно микроконтроллеры выполняются в металлокерамическом и пластмассовых корпусах типа 2123.40 (прямоугольном корпусе с 40 контактами).

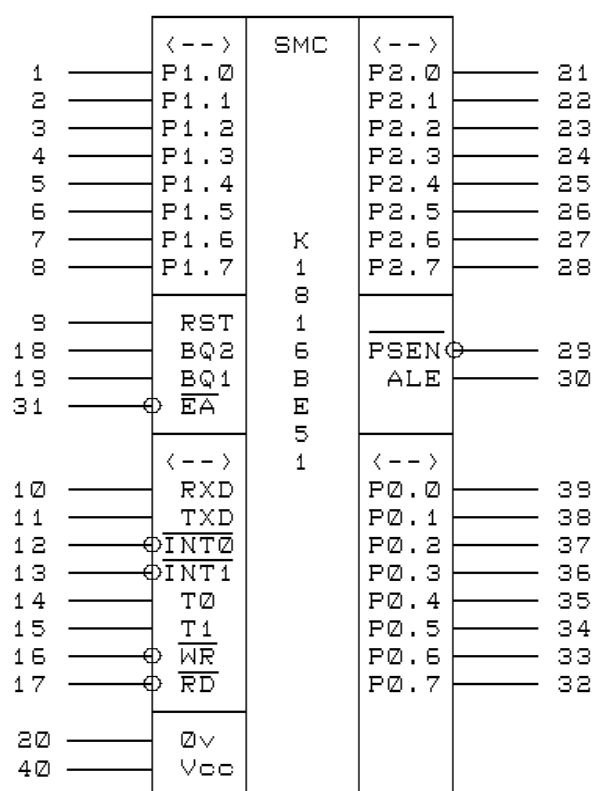


Рис. 1.2. Условное графическое обозначение микроконтроллера I8051.

Таблица 1.2. Назначение выводов микроконтроллера I8051

N вы- вода	Обозначение	Назначение	Тип
1-8	P1.0 – P1.7	8-разрядный двунаправленный порт ввода/вывода P1. Вход адреса A0-A7 при проверке внутреннего ПЗУ (РПЗУ).	вход/ выход

9	RST/VPD	Reset / Voltage Power Done. Сигнал общего сброса. Вывод резервного питания ОЗУ от внешнего источника.	вход
	P3.0 - P3.7	8-разрядный двунаправленный порт ввода/вывода P3 с дополнительными функциями:	вход/ выход
10	P3.0 - RXD	Receive Data pin. Последовательные данные приемника	вход выход
11	P3.1 - TXD	Transmit Date. Последовательные данные передатчика	вход вход
12	P3.2 - INT0	Вход внешнего прерывания 0	вход
13	P3.3 - INT1	Вход внешнего прерывания 1	вход
14	P3.4 - T0	Вход таймера/счетчика 0	выход
15	P3.5 - T1	Вход таймера/счетчика 1	
16	P3.6 - WR	WRITE. Выход стробирующего сигнала при записи во внешнюю память данных	выход
17	P3.7 - RD	READ. Выход стробирующего сигнала при чтении из внешней память данных	
18	BQ2	Выводы для подключения кварцевого резонатора	выход
19	BQ1		вход
20	Vss	общий вывод	вход
	P2.0 - P2.7	8-разрядный двунаправленный порт ввода/вывода P2 (при обращении к внешней памяти в режиме 16-разрядной адресации выводит старший байт адреса)	вход/ выход
21	P2.0 - A8		
22	P2.1 - A9		
23	P2.2 - A10		
24	P2.3 - A11		
25	P2.4 - A12		
26	P2.5 - A13		
27	P2.6 - A14		
28	P2.7 - A15		
29	PSEN	Programm Store Enable. Сигнал разрешения внешней памяти программ	выход
30	ALE/PROG	Adress Latch Enable / Programming. Строб адреса внешней памяти/программирование	вход/ выход
31	EA/VPP	External Adress / Voltage Power Programming. Сигнал отключения резидентной памяти программ / напряжение программирования	вход/ выход

32	P0.0 – P0.7	8-разрядный двунаправленный порт ввода/вывода P0 (при обращении к внешней памяти мультиплексированная шина младшего байта адреса и байта данных)	вход/ выход
33	P0.0 – AD0		
34	P0.1 – AD1		
35	P0.2 – AD2		
36	P0.3 – AD3		
37	P0.4 – AD4		
38	P0.5 – AD5		
39	P0.6 – AD6		
40	P0.7 – AD7		
40	Vcc	Напряжение питания +5v	вход

2. СТРУКТУРА МИКРОКОНТРОЛЛЕРА I8051

Функциональная схема микроконтроллера представлена на рис. 2.1.

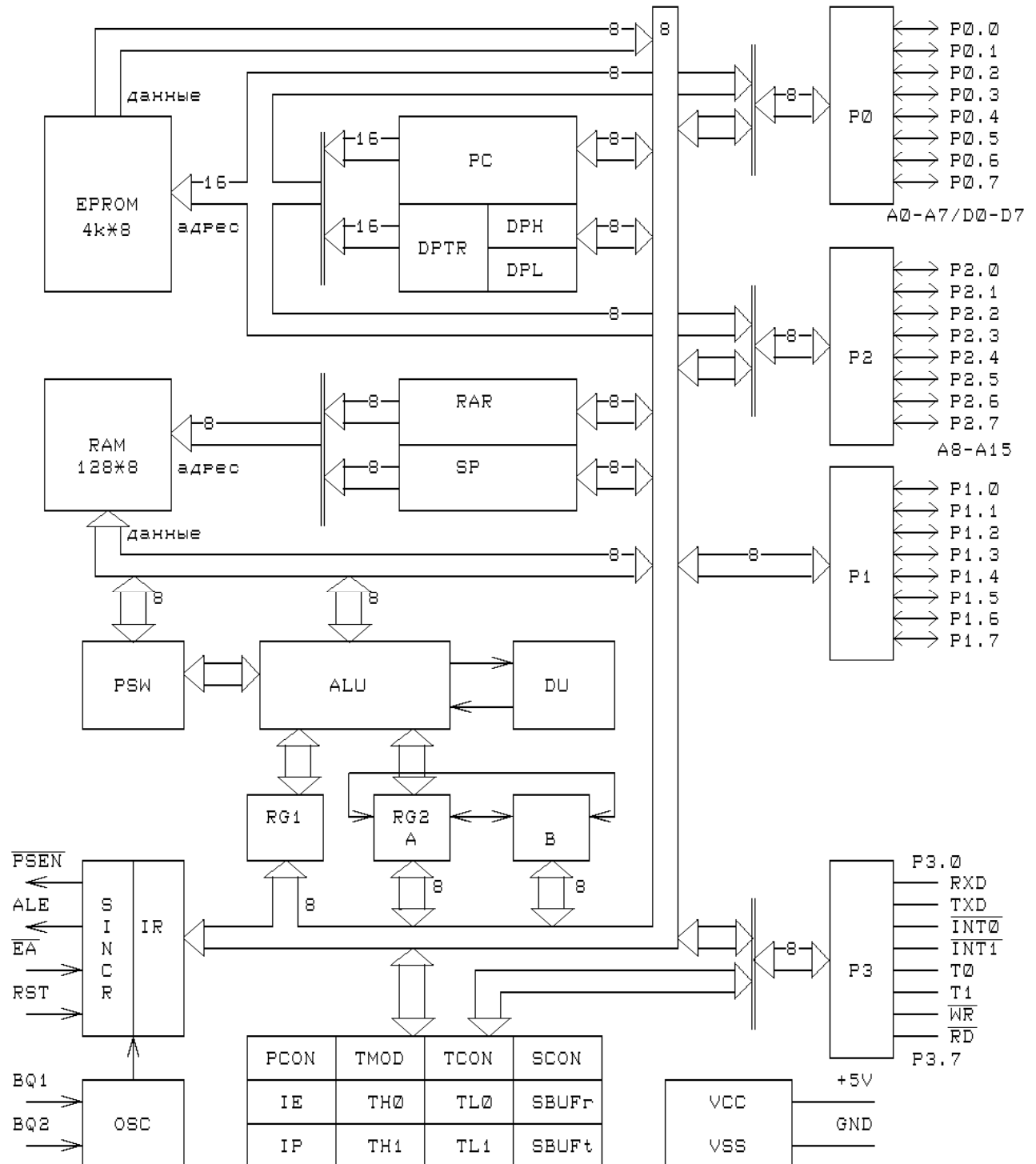


Рис. 2.1. Функциональная схема микроконтроллера I8051.

2.1 Арифметико-логическое устройство.

Основным блоком микроконтроллера является 8-битное арифметико-логическое устройство – ALU (АЛУ), выполняющее

- арифметические операции сложения, вычитания, умножения и деления,
- логические операции И, ИЛИ, исключающее ИЛИ,
- операции циклического сдвига, инвертирования и т.д..

Операционным блоком АЛУ является параллельный восьмиразрядный сумматор. Он представляет собой схему комбинационного типа с последовательным переносом, предназначенную для выполнения арифметических, логических операций.

Одним из операндов арифметических и логических операций всегда является аккумулятор А. В ходе арифметических операций умножения и деления также участвует регистр В. Для временного хранения операндов в арифметико-логическом устройстве используются программно недоступные регистры RG1 и RG2. Результаты выполнения операций арифметико-логического устройства фиксируются в схеме формирования признаков. При выполнении операций с двоично-десятичными числами используется схема десятичной коррекции DU.

Важной особенностью арифметико-логического устройства является его способность оперировать битами данных. Отдельные программно – доступные биты могут быть установлены, сброшены, инвертированы, переданы, проверены и использованы в логических операциях. За эту способность однокристалльный микроконтроллер часто называют булевым процессором.

Арифметико-логическое устройство микроконтроллера может оперировать четырьмя типами информации: булевскими (1 бит), цифровыми (4 бита), байтными (8 бит) и адресными (16 бит). В арифметико-логическом устройстве выполняется 51 различная операция пересылки или преобразования этих данных, при этом используется 11 режимов адресации (7 для данных и 4 для адресов). Это позволяет получить до 255 возможных при однобайтном коде операций.

Аккумулятор является одним из источников операнда и местом фиксации результата при выполнении арифметических, логических операций и ряда операций передачи данных. Кроме того, только с использованием аккумулятора могут быть выполнены операции сдвигов,

проверка на нуль, формирование флага паритета и т. п.. Однако многие команды выполняются и без участия аккумулятора: передача данных из ячейки резидентной памяти данных в любой регистр микроконтроллера, загрузка непосредственного операнда в любой регистр и т.д. Кроме того, переменные могут быть инкрементированы, декрементированы и проверены (test) без использования аккумулятора. Это является отличием данного микроконтроллера от большинства микропроцессоров, архитектура которых опирается на аккумулятор, используя адресацию "по умолчанию".

При выполнении многих команд в арифметико-логическом устройстве формируется ряд признаков операции (флагов), которые фиксируются в регистре слова состояния программы – PSW (ССП). В таблице 2.1 приводится перечень флагов слова состояния программы микроконтроллера.

Это флаг переноса C, вспомогательного переноса AC, флаг переполнения OV и флаг паритета P, которые формируются арифметико-логическим устройством при выполнении арифметических и логических операций.

Наиболее "активным" является флаг переноса C, который принимает участие и модифицируется в процессе выполнения множества операций, включая сложение, вычитание и сдвиги. Кроме того, флаг переноса выполняет функции "булевого аккумулятора" в командах работы с отдельными битами.

Флаг переполнения OV фиксирует арифметическое переполнение при операциях над целыми числами со знаком и делает возможным использование арифметики в дополнительных кодах.

Флаг паритета P устанавливается/сбрасывается при выполнении команд микроконтроллера и фиксирует нечетное/четное число единичных бит в аккумуляторе.

Флаг пользователя F0 специфицируется программистом. Два бита регистра слова состояния программы RS0 и RS1 позволяют выбирать один из четырех рабочих банков регистров, состоящих из восьми регистров общего назначения R0–R7. Причем арифметико-логическое устройство не управляет флагами RS0 и RS1 селекции банка регистров, и их значение полностью определяется прикладной программой.

Таблица 2.1. Формат слова состояния программы (PSW).

Символ	Позиция	Имя и назначение			
C	PSW.7	Флаг переноса. Устанавливается и сбрасывается аппаратурно или программой при выполнении арифметических и логических операций			
AC	PSW.6	Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратурными средствами при выполнении команд сложения и вычитания и сигнализирует о переносе или заёме в бите 3.			
F0	PSW.5	Флаг пользователя. Устанавливается, сбрасывается и проверяется программой как флаг, специфицируемый пользователем			
RS0 RS1	PSW.4	Выбор банка регистров. Устанавливается и сбрасывается программой для выбора рабочего банка регистров:			
	PSW.3	RS1	RS0	Банк	Адреса РПД
		0	0	0	00h - 07h
		0	1	1	08h - 0Fh
		1	0	2	10h - 17h
		1	1	3	18h - 1Fh
OV	PSW.2	Флаг переполнения. Устанавливается и сбрасывается аппаратурно при выполнении арифметических операций			
---	PSW.1	Не используется			
P	PSW.0	Флаг паритета. Устанавливается и сбрасывается аппаратурно в каждом цикле команды и фиксирует нечетное - четное число единичных бит в аккумуляторе.			

2.2 Организация памяти программ и данных

Контроллер I8051 имеет гарвардскую архитектуру – раздельное адресное пространство памяти программ и данных.

Организация памяти микроконтроллера I8051 представлена на рис. 2.2.

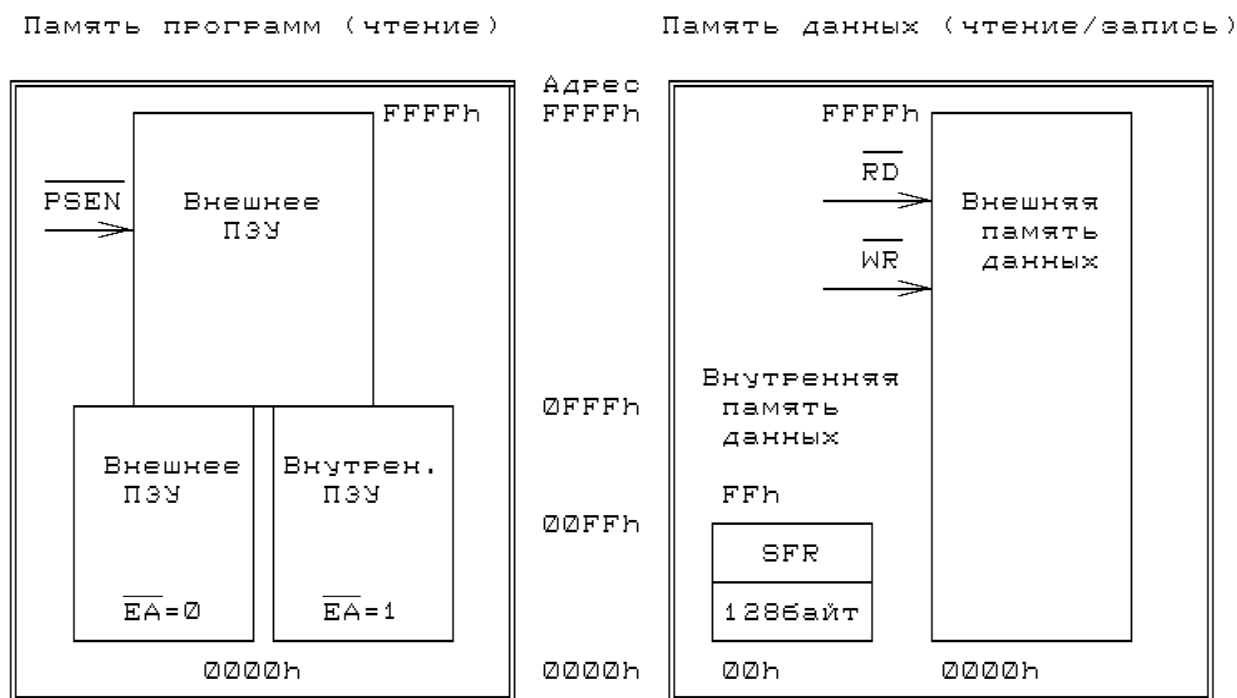


Рис. 2.2. Организация памяти микроконтроллера I8051

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

Память программ EPROM (постоянное запоминающее устройство – ПЗУ или стираемое программируемое постоянное запоминающее устройство – СППЗУ) располагается с адреса 0000h, имеет емкость 4 Кбайта и предназначена для хранения команд, констант, управляющих слов инициализации, таблиц перекодировки входных и выходных переменных и т. п.. Память программ, размещенная на кристалле микроконтроллера, называется резидентной памятью программ (РПП). Резидентная память программ имеет 16-битную шину адреса, через которую обеспечивается доступ к памяти из счетчика команд – РС (СК) или из регистра-указателя данных – DPTR (РУД). Регистр-указатель данных выполняет роль базового регистра при косвенной адресации памяти про-

грамм и косвенных переходах. Резидентная память программ доступна только для чтения.

Память данных RAM (оперативное запоминающее устройство) предназначена для хранения информации в процессе выполнения прикладной программы, адресуется одним байтом и имеет емкость 128 байт. Память данных, размещенная на кристалле микроконтроллера, называется резидентной памятью данных (РПД). Первые 32 байта сгруппированы в 4 банка по 8 регистров R0 – R7. Команды микроконтроллера могут обращаться к ним как к регистрам общего назначения R0 – R7 и как к ячейкам ОЗУ с прямой адресацией от 00h до 1Fh. Ячейки резидентного ОЗУ с адреса 20h до адреса 2Fh обеспечивают кроме прямой адресации, адресацию 128 битов (0–127).

Резидентная память данных имеет 8-битную шину адреса, через которую обеспечивается доступ к памяти из регистра адреса – RAR (РА) или из регистра-указателя стека – SP (PUC). Регистр-указатель стека используется микроконтроллером при работе со стеком. Через регистр адреса осуществляется прямая и косвенная адресация памяти данных.

К резидентной памяти данных примыкают адреса регистров специальных функций SFR (PCФ), которые перечислены в таблице 2.2. Обращение к указанным регистрам возможно несколькими способами: по символическому имени, непосредственно по адресу и побитово. При символическом обращении указывается имя адресуемого операнда, представленное в левом столбце таблицы 2.2. При прямой байтовой адресации указывается прямой адрес в области SFR, указанный в правом столбце таблицы 2.2. В таблице 2.3 представлены регистры специальных функций SFR при побитной адресации указанных регистров. Побитная адресация подразумевает также использование символического и прямого обращения к заданному биту.

Таблица 2.2. Блок регистров специальных функций (SFR) при байтной адресации

Символ	Наименование	Адрес
ACC *	Аккумулятор	0E0h
B *	регистр-расширитель аккумулятора	0F0h
PSW *	слово состояния программы	0D0h
SP	регистр/указатель стека	81h
DPTR	регистр-указатель данных (DPH) (DPL)	83h 82h
P0 *	порт 0	80h
P1 *	порт 1	90h
P2 *	порт 2	0A0h
P3 *	порт 3	0B0h
IP *	регистр приоритетов прерываний	0B8h
IE *	регистр разрешения прерываний	0A8h
TMOD	регистр режима таймера/счетчика	89h
TCON *	регистр управления/состояния таймера/счетчика	88h
TH0	старший байт таймера/счетчика 0	8Ch
TL0	младший байт таймера/счетчика 0	8Ah
TH1	старший байт таймера/счетчика 1	8Dh
TL1	младший байт таймера/счетчика 1	8Bh
SCON *	регистр управления последовательного порта	98h
SBUF	буфер последовательного порта	99h
PCON	регистр управления мощностью	87h

Примечание: Регистры, имена которых отмечены знаком (*), допускают адресацию отдельных бит.

Таблица 2.3. Регистры специальных функций при побитной адресации

Мнемоника бита регистра /битовые адреса									
SFR	7	6	5	4	3	2	1	0	Адрес
B	B.7 0F7h	B.6 0F6h	B.5 0F5h	B.4 0F4h	B.3 0F3h	B.2 0F2h	B.1 0F1h	B.0 0F0h	0F0h
A	ACC.7 0E7h	ACC.6 0E6h	ACC.5 0E5h	ACC.4 0E4h	ACC.3 0E3h	ACC.2 0E2h	ACC.1 0E1h	ACC.0 0E0h	0E0h
PSW	C 0D7h	AC 0D6h	F0 0D5h	RS0 0D4h	RS1 0D3h	OV 0D2h	- 0D1h	P 0D0h	0D0h
IP	- -	- -	PT2 0BDh	PS 0BCh	PT1 0BBh	PX1 0BAh	PT0 0B9h	PX0 0B8h	0B8h
P3	RD 0B7h	WR 0B6h	T1 0B5h	T0 0B4h	INT1 0B3h	INT0 0B2h	TXD 0B1h	RXD 0B0h	0B0h
IE	EA AFh	- -	ET2 ADh	ES ACh	ET1 ABh	EX1 AAh	ET0 A9h	EX0 A8h	0A8h
P2	P2.7 A7h	P2.6 A6h	P2.5 A5h	P2.4 A4h	P2.3 A3h	P2.2 A2h	P2.1 A1h	P2.0 A0h	0A0h
SCO N	SM0 9Fh	SM1 9Eh	SM2 9Dh	REN 9Ch	TB8 9Bh	RB8 9Ah	TI 99h	RI 98h	98h
P1	P1.7 97h	P1.6 96h	P1.5 95h	P1.4 94h	P1.3 93h	P1.2 92h	P1.1 91h	P1.0 90h	90h
TCO N	TF1 8Fh	TR1 8Eh	TF0 8Dh	TR0 8Ch	IE1 8Bh	IT1 8Ah	IE0 89h	IT0 88h	88h
P0	P0.7 87h	P0.6 86h	P0.5 85h	P0.4 84h	P0.3 83h	P0.2 82h	P0.1 81h	P0.0 80h	80h

Память программ и память данных микроконтроллера может быть расширена до 64 Кбайт путем подключения внешних микросхем памяти. При этом резидентная память данных используется совместно с внешней памятью данных. Различение памяти данных осуществляется программно методами адресации.

При обращении к внешней памяти данных в зависимости от типа команды микроконтроллер может обращаться к 256 байтам или к 64 Кбайтам. При обращении к 256 байтам формируется 8-разрядный адрес, выдаваемый через порт P0. При формировании 16-разрядного адреса, младший байт выдается через порт P0, а старший - через порт P2. Байт адреса, выдаваемый через порт P0, должен быть зафиксирован во

внешнем регистре по спаду сигнала ALE (Address Latch Enable - Строб адреса внешней памяти), т.к. в дальнейшем порт P0 используется как порт шины данных для обмена информацией. В зависимости от направления обмена формируются управляющие сигналы внешней памяти данных RD (Read - Чтение) и WR (Write - Запись). При работе только с резидентной памятью данных эти сигналы RD и WR не формируются.

Различие в конфигурации памяти программ определяется сигналом EA (External Address - Отключение резидентной памяти программ). При подключении EA к напряжению питания микроконтроллер работает с внутренней памятью программ. При выработке счетчиком команд адреса, превышающего 0FFFh (4 КБайта), происходит обращение к внешней памяти программ. При подключении вывода EA к нулевому потенциалу микроконтроллер работает только с внешней памятью программ. Для чтения из внешней памяти программ формируется сигнал PSEN (Program Store Enable - Разрешение внешней памяти программ) и 16-разрядный адрес (аналогично обращению к внешней памяти данных). При работе с резидентной памятью программ сигнал PSEN не формируется.

3. СИСТЕМА КОМАНД МИКРОКОНТРОЛЛЕРА I8051

Система команд однокристалльной микро-ЭВМ представляет большие возможности для обработки данных, обеспечивает реализацию логических, арифметических операций, а также управление в режиме реального времени. В микро-ЭВМ реализована побитовая, потетрадная (4 бита), побайтовая (8 бит) и 16 -разрядная обработка данных.

Микроконтроллер семейства I8051 - это 8-разрядная однокристалльная микро-ЭВМ. Поэтому ПЗУ, ОЗУ, регистры специального назначения, АЛУ и внешние шины микро-ЭВМ имеют байтовую организацию. Двухбайтовые данные используются только регистром-указателем (DPTR) и счетчиком команд (PC). Следует отметить, что регистр - указатель данных может быть использован как двухбайтовый регистр DPTR или как два однобайтовых регистра специального назначения DPH и DPL. Счетчик команд всегда используется как двухбайтовый регистр.

Синтаксис команд ассемблерного языка однокристалльной микро-ЭВМ состоит из мнемонического обозначения функции, вслед за которым идут операнды, указывающие методы адресации и типы данных. Различные типы данных или режимы адресации определяются установленными операндами. Например, аббревиатура "MOV" используется восемнадцатью различными командами для обработки трех типов данных (битов, байтов, адресов) в различных адресных пространствах.

Система команд микроконтроллера I8051 содержит 111 базовых команд, которые удобно разделить по функциональному признаку на пять групп: команды передачи данных, арифметических операций, логических операций, передачи управления и операций с битами.

В машинном коде команды занимают один, два или три байта. Первый байт команды любого типа и формата всегда содержит код операции (КОП). Второй и третий байты содержат либо адреса операндов, либо непосредственные операнды.

Состав операндов микроконтроллера I8051 включает в себя операнды четырех типов: биты, 4-битные цифры, байты и 16-битные слова. При этом используются регистровая, прямая, непосредственная, косвенная и неявная адресация.

Для адресации бит используется прямой 8-битный адрес (bit). При этом имеется возможность адресации отдельных бит блока регист-

ров специальных функций и портов. Косвенная адресация бит невозможна.

Четырехбитные операнды используются только при операциях обмена (команды SWAP и XCHD).

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

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

Трехбайтовые команды перехода и вызова LCALL, LJMP (с 16 - разрядным адресом) позволяют осуществлять переход и обращение по любому адресу адресного пространства памяти программ емкостью 64 Кбайт. Если необходим переход в пределах области памяти программ 2К, то можно использовать команды перехода и вызова с 11 - разрядным адресом (ACALL, AJMP). Переход внутри участка памяти, определяемый 8-разрядной величиной смещения, осуществляется по команде SJMP.

Команды выполняются за один, два или четыре (умножение и деление) машинных цикла. При тактовой частоте генератора 12 МГц длительность машинного цикла составляет 1мкс. Поэтому из 111 типов команд 64 выполняются за 1 мкс, 45 команд - за 2 мкс и две команды умножения и деления (MUL, DIV) выполняются за 4 мкс.

Все команды условных переходов осуществляются относительно содержимого счетчика команд с адресом перехода, вычисляемым центральным процессором во время выполнения команды.

В результате выполнения некоторых команд формируется словосостояние программы (PSW), включающее в себя четыре флага: С-перенос, АС-вспомогательный перенос, OV-переполнение и Р-паритет.

Флаг паритета напрямую зависит от текущего значения аккумулятора. Если число единичных бит аккумулятора нечетное, то флаг Р устанавливается, а если четное - сбрасывается. Все попытки изменить флаг Р, присваивая ему новое значение, будут безуспешными, если содержимое аккумулятора при этом останется неизменным.

Флаг АС устанавливается в случае, если при выполнении операции сложения/вычитания между тетрадами байта возник перенос/заем.

Флаг C устанавливается, если в старшем бите результата возникает перенос или заем. При выполнении операций умножения и деления флаг C сбрасывается.

Флаг OV устанавливается, если результат операции сложения /вычитания не укладывается в семи битах и старший (восьмой) бит результата не может интерпретироваться как знаковый. При выполнении операции деления флаг OV сбрасывается, а в случае деления на нуль устанавливается. При умножении флаг OV устанавливается, если результат больше 225. В таблице 3.1 представлены команды, модифицирующие флаги результата.

Таблица 3.1. Команды, модифицирующие флаги результата

Команды (мнемоника)	Флаги		
	C	OV	AC
ADD	x	x	x
ADDC	x	x	x
SUBB	0	x	
MUL	0	x	
DIV	0	x	
DA	x		
RRC	x		
RLC	x		
SETB C	1		
CLR C	0		
CPL C	C		
ANL C, bit	x		
ORL C, bit	x		
MOV C, bit	x		
CJNE	x		

Примечание: X- бит принимает значение 0 или 1.

Следует отметить, что операции над байтом 0D0h или битами 0D0h - 0D8h, являющимися разрядами словосостояния программы PSW, влияют на установку всех флагов. Система команд приведена в Приложении 1.

3.1 Команды передачи данных

Команды передачи данных можно разделить на несколько блоков в зависимости от операндов, участвующих в операции. На рис. 3.1 представлен граф возможных путей передачи данных в микроконтроллере I8051.

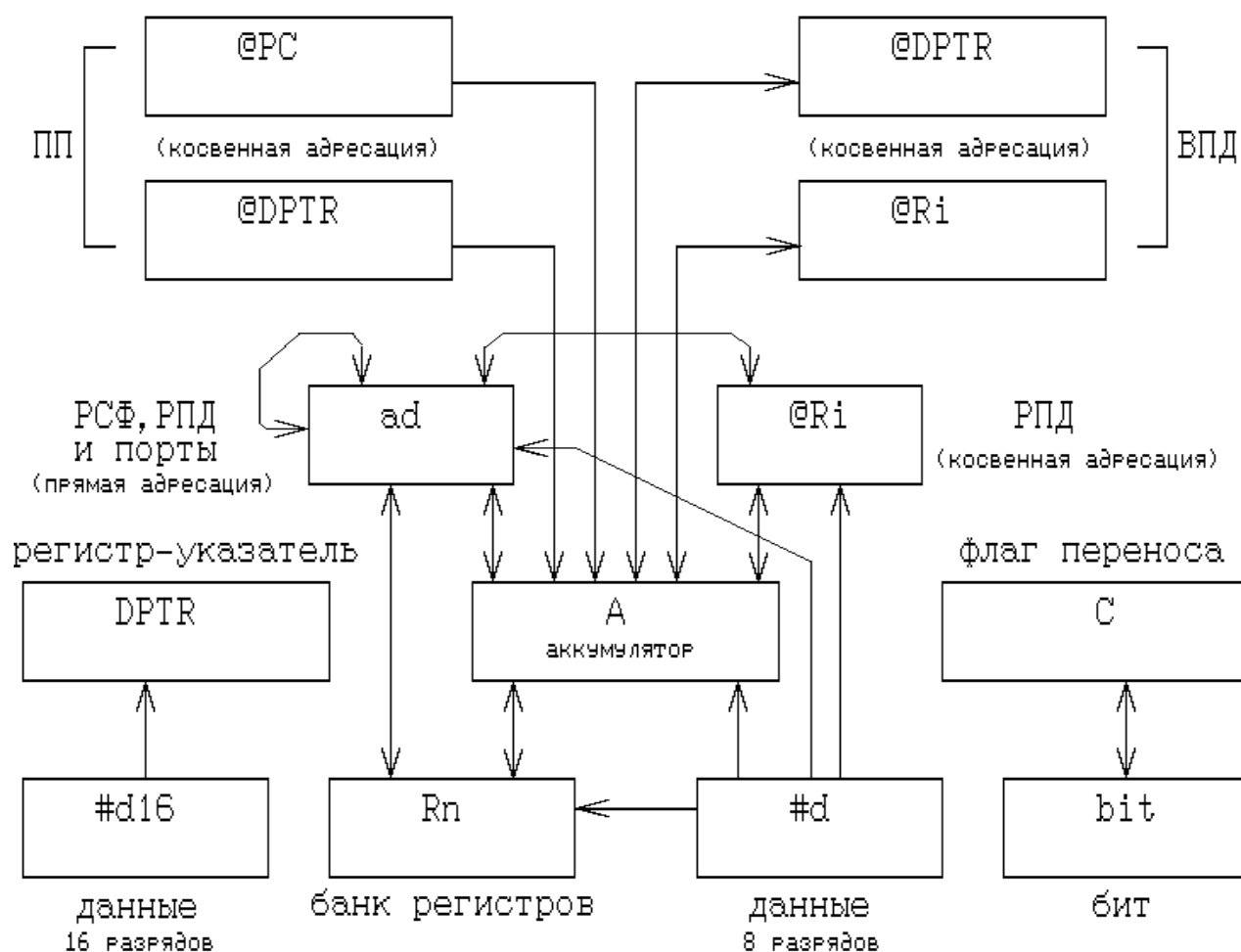


Рис. 3.1. Граф путей передачи данных в микроконтроллере I8051.

Основным блоком команд передачи данных являются команды работы с аккумулятором. Содержимое аккумулятора может быть обменено

- с содержимым рабочих регистров (выбранного банка) - Rn ;
- ячеек резидентной памяти данных, адресуемой с помощью косвенно-регистровой адресации - $@Ri$;
- младшие четыре разряда аккумулятора (разряды 3-0) могут быть обменены с младшими разрядами содержимого ячеек внутреннего ОЗУ, выбираемых с помощью косвенно-регистровой адресации;
- прямоадресуемых ячеек резидентного ОЗУ, регистров специальных функций и портов - ad ;

- ячеек 256-байтового блока внешнего ОЗУ данных, выбранного с использованием косвенно-регистровой адресации через регистры указатели R0 или R1 (выбранного банка регистров) - @Ri;

- ячеек внутри адресного пространства 64 Кбайт внешнего ОЗУ, выбранного с использованием косвенно-регистровой адресации через регистр-указатель данных DPTR - @DPTR.

Команды передачи в аккумулятор кодов команд и констант из памяти программ осуществляются с использованием косвенной адресации. Байт константы может быть передан в аккумулятор из ячейки памяти программ, адресуемой суммой базового регистра (PC или DPTR) и индексного регистра (содержимого A) - @DPTR+A или @PC+A.

К командам передачи данных, работающим с аккумулятором, относятся также команды записи 8-разрядной константы - #d. Запись 8-разрядной константы возможна также в область резидентной памяти данных с прямой и косвенной адресацией. 16-разрядная константа может быть записана в регистр-указатель данных DPTR.

Команды работы с прямоадресуемыми операндами позволяют производить обмен данными:

- с рабочими регистрами микроконтроллера;
- с ячейками резидентного ОЗУ, адресуемого с помощью косвенной адресации;
- между прямоадресуемыми ячейками резидентного ОЗУ, регистрами специальных функций и портами. Эти команды позволяют принимать данные из порта в ячейку внутреннего ОЗУ без использования рабочих регистров или аккумулятора.

В логическом процессоре к командам передачи данных относятся команды пересылки любого прямоадресуемого бита в бит переноса и наоборот.

3.2 Арифметические команды

В наборе команд однокристалльной микро-ЭВМ имеются следующие арифметические операции:

- сложение,
- сложение с учетом флага переноса,
- десятичная коррекция,
- вычитание с заемом,
- инкрементирование,
- декрементирование,
- умножение,
- деление.

В АЛУ производятся действия над целыми числами без знака. В двухоперандных операциях: сложение (ADD), сложение с переносом (ADDC) и вычитание с заемом (SUBB) аккумулятор является первым операндом и принимает результат операции. Вторым операндом может быть рабочий регистр выбранного банка рабочих регистров, ячейка внутренней памяти данных с косвенно-регистровой и прямой адресацией или байт непосредственных данных. Указанные операции влияют на флаги: переполнения, переноса, промежуточного переноса и флаг четности в слове состояния процессора (PSW).

Использование разряда переноса позволяет многократно повысить точность при операциях сложения (ADDC) и вычитания (SUBB).

Выполнение операций сложения и вычитания с учетом знака может быть осуществлено с помощью программного управления флагом переполнения (OV) регистра PSW. Флаг промежуточного переноса (AC) обеспечивает выполнение арифметических операций в двоично-десятичном коде.

Операции инкрементирования и декрементирования на флаги не влияют.

Существуют три арифметических операции, которые выполняются только на аккумуляторе: две команды проверки содержимого аккумулятора А (JZ, JNZ), и команда десятичной коррекции при сложении двоично-десятичных кодов.

При операции умножения содержимое аккумулятора А умножается на содержимое регистра В и результат размещается следующим образом: младший байт в регистре А, старший в регистре В.

В случае выполнения операции деления целое от деления помещается в аккумулятор А, остаток от деления – в регистр В.

3.3 Логические команды с байтовыми переменными

Система команд однокристальной микро-ЭВМ позволяет реализовать логические операции: "И", "ИЛИ", "ИСКЛЮЧАЮЩЕЕ ИЛИ" на регистре-аккумуляторе А и байте источнике. Вторым операндом (байтом источником) при этом может быть рабочий регистр в выбранном банке рабочих регистров, регистр внутреннего ОЗУ с косвенно-регистровой адресацией, прямоадресуемые ячейки внутреннего ОЗУ, регистры специального назначения и байт непосредственных данных.

Указанные логические операции могут быть реализованы на любом прямоадресуемом регистре внутреннего ОЗУ или регистре специального назначения с использованием в качестве второго операнда содержимого аккумулятора А или непосредственных данных.

Существуют логические операции, которые выполняются только на аккумуляторе: сброс и инвертирование всех восьми разрядов А, циклический сдвиг влево и вправо, циклический сдвиг влево и вправо с учетом переноса, обмен местами старшей и младшей тетрад (ниблов) внутри аккумулятора.

3.4 Команды битового процессора

Битовый процессор является частью архитектуры микроконтроллеров семейства I8051 и его можно рассматривать как независимый процессор побитовой обработки. Битовый процессор выполняет свой набор команд, имеет свое побитово-адресуемое ОЗУ и свой ввод-вывод.

Команды, оперирующие с битами, обеспечивают прямую адресацию 128 битов (адреса 0-127) в шестнадцати ячейках внутреннего ОЗУ (ячейки с адресами 20h-2Fh) и прямую побитовую адресацию регистров специального назначения, адреса которых кратны восьми: P0(80h),

TCON(88h), P1(90h), SCON(98h), P2(0A0h), IE(0A8h), P3(0B0h), IP(0B8h), PSW(0D0h), A(0E8h), B(0F0h).

Каждый из относительно адресуемых битов может быть установлен в "1", сброшен в "0", инвертирован, проверен. Могут быть реализованы переходы: если бит, установлен и если бит, не установлен. Осуществляется переход, если бит установлен, со сбросом этого бита, или этот бит может быть перезаписан в (из) разряда переноса. Между любым прямоадресуемым битом и флагом переноса могут быть произведены логические операции "И", "ИЛИ", где результат заносится в разряд флага переноса. Команды побитовой обработки обеспечивают реализацию сложных функций комбинаторной логики и оптимизацию программ пользователя.

3.5 Команды ветвления и передачи управления

Адресное пространство памяти программ микроконтроллера не имеет страничной организации, что позволяет свободно перемещать фрагменты программ внутри адресного пространства, при этом не требуется изменение номера страницы.

Перемещение отдельных фрагментов программы обеспечивает возможность использования перемещаемых программных модулей различными программами.

Команды 16-разрядных переходов и вызовов подпрограмм позволяют осуществлять переход в любую точку адресного пространства памяти программ объемом 64 Кбайт.

Команды 11-разрядных переходов и вызовов подпрограмм обеспечивают переходы внутри программного модуля емкостью 2 Кбайт.

В системе команд имеются команды условных и безусловных переходов относительно начального адреса следующей команды в пределах от PC-128 до PC+127.

Команды проверки отдельных разрядов позволяют осуществлять условные переходы по состоянию "0" или "1" прямоадресуемых битов.

Команды проверки содержимого аккумулятора (ноль/не ноль) позволяют осуществлять условные переходы по содержимому A.

Косвенно-регистровые переходы в системе команд микро-ЭВМ обеспечивают ветвление относительно базового регистра (содержимого DPTR или PC) со смещением, находящимся в аккумуляторе.

4. УСТРОЙСТВА ВВОДА/ВЫВОДА

К устройствам ввода-вывода микроконтроллера I8051 относятся

- четыре восьмиразрядных порта ввода-вывода, работающие в параллельном формате обмена;
- два 16-битных таймера/счетчика, используемых для подсчета внешних событий, организации временных задержек и тактирования коммуникационного порта;
- асинхронный канал дуплексного последовательного ввода/вывода информации со скоростью до 375 кбит/сек,
- блок двухуровневого векторного прерывания от пяти источников.

Порты микроконтроллера I8051 являются двунаправленными портами ввода-вывода и предназначены для обеспечения обмена информацией микроконтроллера с внешними устройствами, образуя 32 линии ввода-вывода. Обмен информацией может осуществляться побайтно или побитно. Каждый порт состоит из управляемого регистра-защелки, входного буфера и выходного драйвера.

Кроме основных функций ввода-вывода порты 0 и 2 используются при обращении к внешней памяти программ и данных. При этом через порт 0 в режиме временного мультиплексирования сначала выводится младший байт адреса внешней памяти, а затем выдается или принимается байт данных. Через порт 2 выводится старший байт адреса в тех случаях, когда разрядность адреса превышает 8 разрядов. Таким образом при работе с внешней памятью порты 0 и 2 не могут быть использованы для функций ввода-вывода.

Все выходы порта 3 имеют индивидуальную альтернативную функцию, представленные в таблице 4.1. Альтернативные функции инициализируются путем записи 1 в соответствующие биты регистра-защелки (P3.7 - P3.0) порта 3. По сигналу сброса микроконтроллера все порты автоматически настраиваются на режим ввода.

Порты ввода-вывода могут быть адресованы как ячейки памяти в адресном пространстве резидентной памяти данных:

при байтовом обмене	при битовом обмене
P0 - 80h	80h - 87h
P1 - 90h	90h - 97h
P2 - 0A0h	0A0h - 0A7h
P3 - 0B0h	0B0h - 0B7h.

Таблица 4.1. Альтернативные функции порта 3

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

Обращение к портам ввода-вывода возможно с использованием команд, оперирующих с байтом, отдельным битом и произвольной комбинацией бит. При этом возможна реализация специального режима работы "чтение-модификация-запись". В этом случае ввод сигналов осуществляется не с внешних выводов порта, а из его регистра-защелки, что позволяет исключить неправильное считывание ранее введенной информации. Два программируемых 16-битных таймера/счетчика могут быть использованы в качестве таймеров или счетчиков внешних событий. При работе в качестве таймера содержимое счетчика инкрементируется в каждом машинном цикле. Тактовая частота в этом случае равна частоте кварцевого резонатора, деленной на 12. При работе в качестве счетчика содержимое счетчика инкрементируется при переходе внешнего входного сигнала из 1 в 0. Максимальная частота входного сигнала не должна превышать $1/24$ частоты резонатора, что определяется спецификой фиксации входного перехода в таймере.

Для управления режимами работы и организации прерываний от таймеров используются регистры специальных функций:

TMOD – регистр режима работы таймера/счетчика (РРТС),

TCON – регистр управления/статуса таймера (РУСТ).

Блок последовательного интерфейса и прерываний предназначен для организации ввода-вывода последовательных потоков информации и организации прерывания программ.

Для управления режимами работы блока последовательного интерфейса и прерываний используются регистры специальных функций:

SCON - регистр управления/статуса последовательного интерфейса (РУПП),

IP - регистр приоритетов прерываний (РП),

IE - регистр масок прерывания (РМП).

4.1 Порты ввода-вывода

В микроконтроллерах I8051 существуют 4 многофункциональных 8-битовых порта ввода/вывода P0, P1, P2 и P3, предназначенные для обмена информацией с различными внешними устройствами, в том числе и с внешней памятью программ и данных.

Порты микроконтроллера I8051 являются двунаправленными портами ввода-вывода, образуя 32 линии ввода-вывода. Порт 0 является двунаправленным, а порты 1, 2 и 3 – квазидвунаправленными. Обмен информацией может осуществляться побайтно или побитно. Каждый порт состоит из управляемого регистра-защелки, входного буфера и выходного драйвера.

Кроме основных функций ввода-вывода порты могут выполнять ряд альтернативных функций. Каналы 0 и 2 используются при обращении к внешней памяти программ и данных. При этом через порт 0 в режиме временного мультиплексирования сначала выводится младший байт адреса внешней памяти, а затем выдается или принимается байт данных. При программировании внутренней памяти программ через порт 0 задаются данные программирования, и читается содержимое внутренней памяти программ. Через порт 2 выводится старший байт адреса (A8–A15) в тех случаях, когда при обращении к внешней памяти ее объем превышает 256 байт. При программировании и верификации внутренней памяти программ порт 2 применяется также для задания старших разрядов адреса. Таким образом, при работе с внешней памятью порты 0 и 2 не могут быть использованы для функций ввода/вывода.

Порт 1 используется для задания младшего байта адреса при программировании и чтении внутренней памяти программ микросхемы. При работе микроконтроллера порт 1 непосредственно может быть использован для функций ввода/вывода.

Порт 3 также может быть использован для реализации ввода/вывода, если нет необходимости в формировании дополнительных управляющих сигналов. Все выходы порта 3 имеют индивидуальную альтернативную функцию, наименование которых представлено на рис. 4.1.1, а назначение – в таблице 4.1.1. Альтернативные функции инициализируются путем записи 1 в соответствующие биты регистра-защелки (P3.7 – P3.0) порта 3. По сигналу сброса микроконтроллера все порты автоматически настраиваются на режим ввода.

Выходные линии портов ввода/вывода 1,2 и 3 могут быть нагружены на один вход ТТЛ-схемы. К порту 0 можно подключать два входа ТТЛ-схем.

Обращение к портам ввода/вывода возможно с использованием команд, оперирующих с байтом, отдельным битом и произвольной комбинацией бит.

Порты ввода/вывода в командах, написанных на языке Ассемблера, могут адресоваться с помощью символических имен при байтовой адресации – P0, P1, P2 и P3, при битовой адресации – P0.1, P1.3, P2.4, P3.7 и т.д., а также могут быть адресованы как ячейки памяти в адресном пространстве резидентной памяти данных:

при байтовом обмене при битовом обмене

P0 – 80h	80h – 87h
P1 – 90h	90h – 97h
P2 – 0A0h	0A0h – 0A7h
P3 – 0B0h	0B0h – 0B7h

MSB

LSB

RD	WR	T1	T0	INT1	INT0	TXD	RXD
Старший				Младший			

Рис.4.1.1. Наименование разрядов порта P3.

Таблица 4.1.1. Альтернативные функции порта P3

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

		данных в режиме сдвигающего регистра.
--	--	---------------------------------------

По физическому принципу обращения к портам ввода/вывода команды подразделяются на несколько типов.

Ряд команд микроконтроллера производит чтение данных непосредственно с входных линий портов. К ним относятся следующие команды пересылки данных:

MOV A , Pi - пересылка содержимого порта "i" в аккумулятор;
 MOV Rn , Pi - пересылка содержимого порта "i" в регистр Rn;
 MOV @Ri, Pi - пересылка содержимого порта "i" в ячейку памяти с адресом, записанным в регистре Ri;
 MOV C , Pi.j - пересылка бит "j" порта "i" в признак переноса.

Имеется набор команд, реализующих специальный режим работы "чтение-модификация-запись". В этом случае ввод сигналов осуществляется не с внешних выводов порта, а из его регистра-защелки, что позволяет исключить неправильное считывание ранее введенной информации. После считывания состояния защелки выходного порта происходит модификация этого значения и обратная запись в порт. Примеры подобных команд:

ANL Pi, A - логическое "И" содержимого порта "i" и аккумулятора;

ORL Pi, A - логическое "ИЛИ" содержимого порта "i" и аккумулятора;

XRL Pi, A - логическое "ИСКЛЮЧАЮЩЕЕ ИЛИ" содержимого порта "i" и аккумулятора;

JBC Pi.j,addr- переход, если бит "j" порта "i" равен 1, и сброс бита;

CPL Pi.j - инверсия бита "j" порта "i";

INC Pi - инкремент содержимого порта "i";

DEC Pi - декремент содержимого порта "i";

DJNZ Pi, addr - декремент содержимого порта "i" и переход, если не ноль;

MOV Pi.j, C - пересылка бита переноса в бит "j" порта "i";

CLR Pi.j - сброс бита "j" порта "i";

SETB Pi.j - установка бита "j" порта "i".

4.2 Таймер-счетчик

Микроконтроллеры семейства I8051 располагают двумя таймерами/счетчиками (новые модификации данного микроконтроллера могут иметь три таймера/счетчика).

Программируемые 16-битные таймеры/счетчики предназначены для подсчета внешних событий, организации программно-управляемых временных задержек и измерения временных интервалов.

В состав таймеров/счетчиков входят следующие узлы:

- 8-разрядный регистр режимов TMOD,
- 8-разрядный регистр управления/статуса TCON,
- два 16-разрядных регистра таймеров/счетчиков T/C0 и T/C1,
- схема управления.

Регистр режимов TMOD предназначен для установки режимов работы таймера/счетчика и определения источника его тактирования. Наименование разрядов данного регистра представлены на рис. 4.2.1, а назначение – рассмотрено в таблице 4.2.1.

MSB				LSB			
GATE1	C/T1	M1.1	M0.1	GATE0	C/T0	M1.0	M0.0
Старший				Младший			

Рис.4.2.1. Наименование разрядов регистра режима работы таймера/счетчика TMOD

Таблица 4.2.1. Регистр режима работы таймера/счетчика TMOD

Бит	Позиция	Функция
GATE1	TMOD.7	Управление блокировкой таймера 1. Если бит установлен, то таймер/счетчик 1 работает до тех пор, пока на входе INT1 высокий уровень и бит управления TR1 установлен. Если бит сброшен, то таймер/счетчик включается, когда устанавливается бит управления TR1.
C/T1	TMOD.6	Бит выбора режима таймера или счетчика событий для таймера 1. Если бит сброшен, то работает таймер от внутреннего источника сигналов синхронизации (таймер). Если бит установлен, то работает счетчик от внешних сигналов на входе T1 (счетчик).

M1.1	TMOD.5	Режим работы таймера/счетчика 1.
M0.1	TMOD.4	00 – 13-битный таймер/счетчик. TLi работает как 5-битный предделитель (режим 0). 01 – 16-битный таймер/счетчик. THi и TLi включены последовательно (режим 1). 10 – 8-битный автоперезагружаемый таймер/счетчик на базе TLi. THi хранит значение коэффициента счета, перезагружаемое в TLi по переполнению (режим 2). 11 – TL0 8-битный таймер/счетчик со всеми флагами управления, TH0 8-битный таймер.
GATE0	TMOD.3	Управление блокировкой таймера 0.
C/T0	TMOD.2	Бит выбора режима таймера или счетчика событий для таймера 0.
M1.0	TMOD.1	Режим работы таймера/счетчика 0.
M0.0	TMOD.0	

Регистр управления/статуса TCON предназначен для приема и хранения управляющего слова таймера/счетчика. Для этого используются старшие четыре разряда регистра TCON (TCON.7 – TCON.4). Наименование разрядов данного регистра представлены на рис. 4.2.2, а назначение – рассмотрено в таблице 4.2.2.

MSB

LSB

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Старший				Младший			

Рис.4.2.2. Наименование разрядов управления/статуса таймера/счетчика TCON

Таблица 4.2.2. Регистр управления/статуса таймера/счетчика TCON

Бит	Позиция	Функция
TF1	TCON.7	Флаг переполнения таймера 1. Устанавливается аппаратно при переполнении таймера/счетчика. Сбрасывается при обслуживании прерывания аппаратно.
TR1	TCON.6	Бит управления таймера 1. Устанавливается/сбрасывается программно для пуска/останова.
TF0	TCON.5	Флаг переполнения таймера 0.

TR0	TCON.4	Бит управления таймера 0.
IE1	TCON.3	Флаг запроса внешнего прерывания по входу INT1. Устанавливается аппаратно по прерыванию или программно. Сброс выполняется аппаратно только в случае, когда прерывание вызвано фронтом сигнала INT1. При работе с уровнем сигнала на входе INT1 сброс флага прерывания осуществляется программно.
IT1	TCON.2	Бит управления типом прерывания 1. Устанавливается/сбрасывается программно для спецификации вида прерывания по входу INT1: "0" – прерывание по уровню, "1" – прерывание по фронту (1/0).
IE0	TCON.1	Флаг запроса внешнего прерывания по входу INT0.
IT0	TCON.0	Бит управления типом прерывания 0.

16-разрядные регистры выполняют функцию хранения содержимого счета. Состоят из четырех 8-разрядных регистров TH0, TH1, TL0 и TL1. Регистры TH0 и TH1 содержат старший байт значения Т/С, а TL0 и TL1 – младший. Указанные регистры доступны для записи начальной величины счета, а также для контроля их содержимого. В процессе счета происходит инкрементирование содержимого счетчиков. При работе в качестве таймера содержимое счетчика инкрементируется в каждом машинном цикле. Тактовая частота в этом случае равна частоте кварцевого резонатора, деленной на 12. При работе в качестве счетчика содержимое счетчика инкрементируется при переходе внешнего входного сигнала из 1 в 0. Максимальная частота входного сигнала не должна превышать $1/24$ частоты резонатора, что определяется спецификой фиксации входного перехода в таймере. Переполнение счетчиков (переход из состояния 0FFh в состояние 00) служит признаком окончания счета и инициализирует прерывание от таймеров.

Схема управления генерирует флаги переполнения таймеров/счетчиков и запросов внешних прерываний, фиксирует в триггерах состояния сигналов INT0, INT1, T0 и T1 в каждом машинном цикле и синхронизирует работу регистров таймера/счетчика с работой центрального процессора микроконтроллера.

Режимы работы таймеров/счетчиков определяются значением управляющих битов регистра TMOD M0 и M1 и имеют четыре значения: 0, 1, 2 и 3. Режимы 0, 1 и 2 полностью идентичны для обоих тайме-

ров/счетчиков. Установка T/C0 в режим 3 влияет на режим работы T/C1.

В режиме 0 таймеры/счетчики работают на базе 13-разрядных регистров, образованных соответствующими регистрами THi и пятью младшими разрядами регистров TLi. Регистры TLi выполняют функцию делителя на 32, включенного на входе регистра THi. Работу таймера/счетчика в режиме 0 иллюстрирует рис. 4.2.3

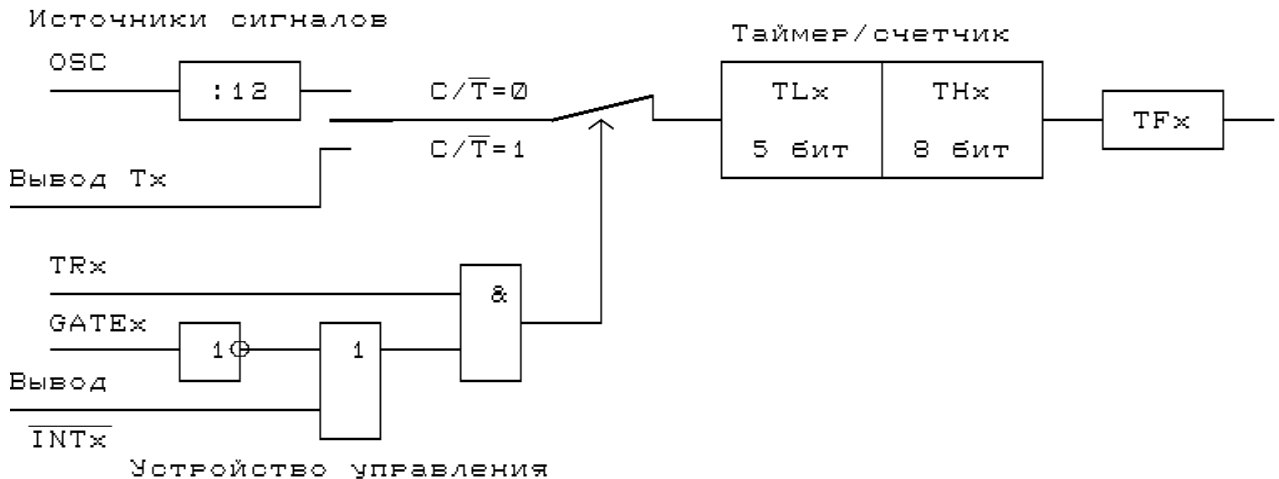


Рис. 4.2.3. Логика работы таймера/счетчика в режиме 0.

Счет таймера/счетчика начинается при установке бита TRx регистра TCON в состояние "1". Установка в "1" бита GATEx регистра TMOD разрешает управление таймером/счетчиком по входу INTx. В этом режиме можно использовать таймер/счетчик для измерения длительности импульсного сигнала, подаваемого на вход запроса прерывания INTx. Установка бита TRx в "0" запрещает счет независимо от состояния других битов.

Бит C/Tx определяет вид синхронизирующего сигнала. Установка бита C/Tx в состояние "0" определяет работу таймера/счетчика как таймера от внутреннего синхросигнала. При состоянии "1" бита C/Tx работа осуществляется от внешнего входного сигнала Tx в режиме счетчика.

В режиме 1 таймеры/счетчики работают на базе 16-разрядных регистров, образованных соответствующими регистрами THi и TLi. Принцип работы в режиме 1 идентичен режиму 0. Работу таймера/счетчика в режиме 1 иллюстрирует рис. 4.2.4.

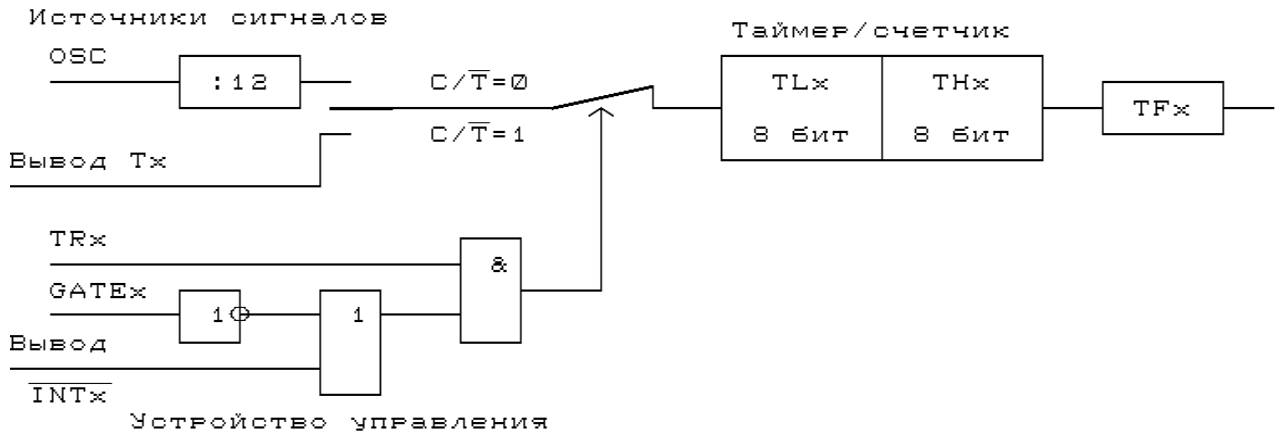


Рис. 4.2.4. Логика работы таймера/счетчика в режиме 1.

В режиме 2 используются 8-разрядные регистры TL0 и TL1. При переполнении этих регистров в процессе счета, т. е. при переходе из состояния 0FFh в состояние 00, происходит перезагрузка их значением из регистров TH0 и TH1 соответственно. Эти регистры загружаются программно, и процесс перезагрузки из THx в TLx не влияет на их содержимое. Работа таймера/счетчика в режиме 2 представлена на рис. 4.2.5.

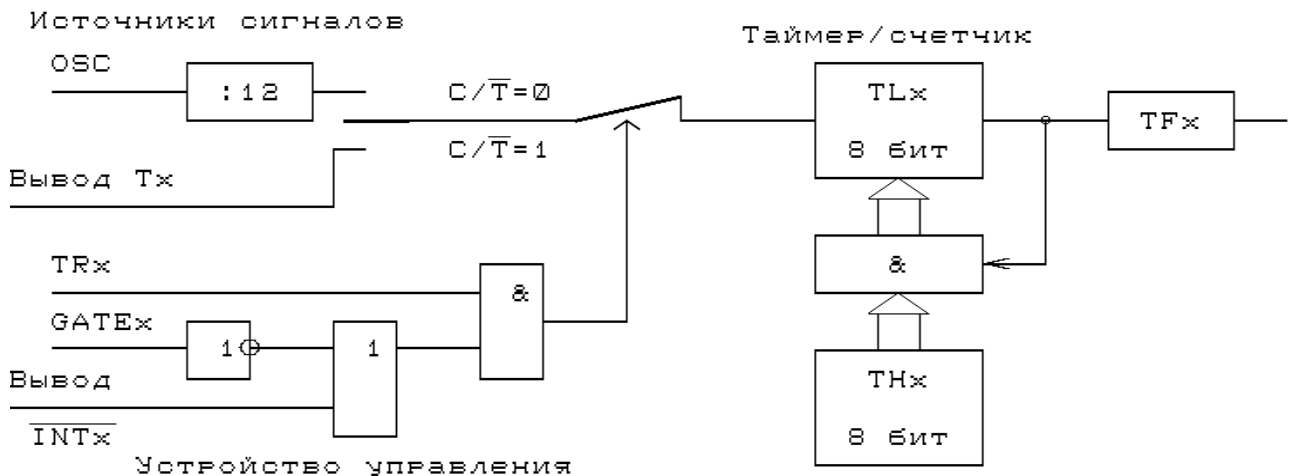


Рис. 4.2.5. Логика работы таймера/счетчика в режиме 2.

В режиме 3 работает только таймер/счетчик 0. Таймер/счетчик 1 в этом режиме заблокирован и сохраняет содержимое своих регистров TL1 и TH1 (как при TR1=0). Таймер/счетчик 0 представлен двумя независимыми 8-разрядными регистрами TL0 и TH0. Устройство на базе регистра TL0 работает со всеми битами управления, флагами и входными линиями таймера/счетчика 0. Устройство на основе регистра TH0 может работать только в таймерном режиме от внутреннего синхронизатора и использует управляющий бит (TR1) и флаг переполнения (TF1) таймера/счетчика 1. Остальные биты таймера/счетчика 1 с работой TH0 не связаны.

При работе таймера/счетчика 0 в режиме 3 приводит к лишению таймера/счетчика 1 управляющего бита включения TR1. По этой причине таймер/счетчик 1 в режимах 0,1,2 при сигнале GATE1=0 всегда включен и может быть использован в любом применении, не требующем прерываний.

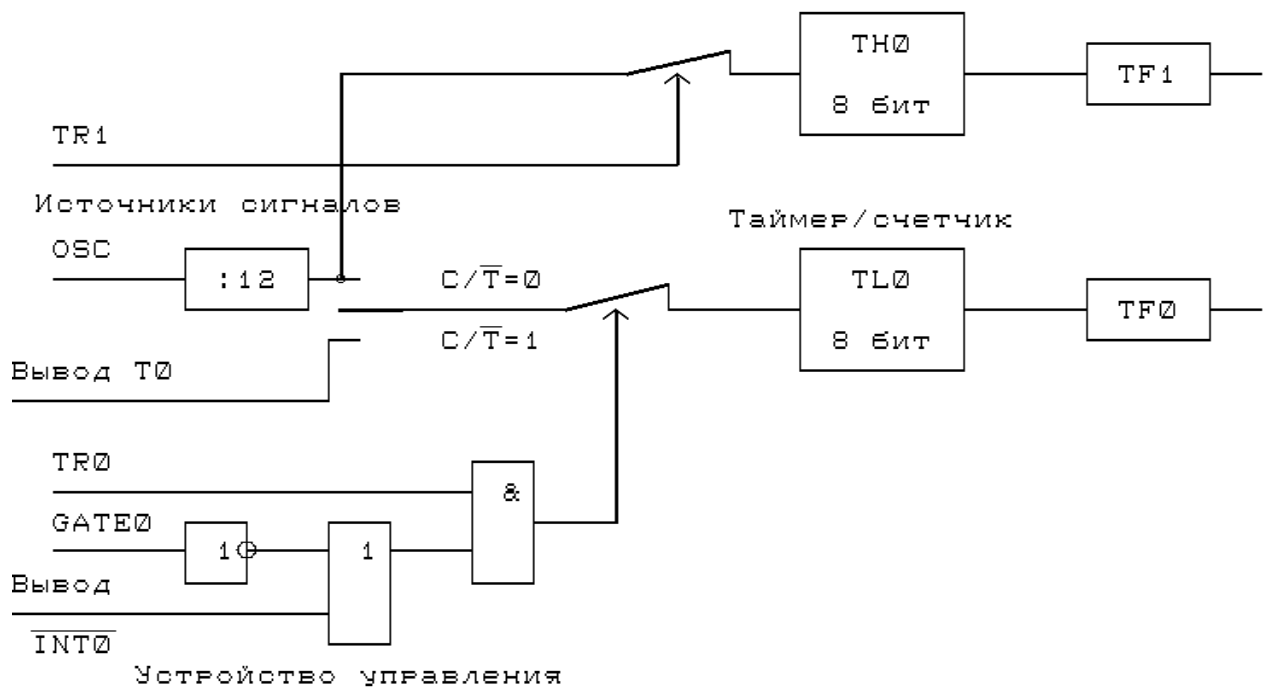


Рис. 4.2.6. Логика работы таймера/счетчика в режиме 3.

4.3 Последовательный интерфейс

Последовательный интерфейс микроконтроллера семейства I8051 предназначен для приема и передачи информации, представленной последовательным кодом, в полном дуплексном режиме. В состав последовательного интерфейса входят: принимающий и передающий сдвигающие регистры, специальный буферный регистр приемопередатчика (SBUF) и схемы управления.

Передача данных осуществляется после команды записи данных в буферный регистр. Далее данные автоматически переписываются в сдвигающий регистр передатчика, где преобразуются из параллельного формата представления в последовательный поток символов с необходимыми служебными битами и выдаются в последовательный канал связи с заданной скоростью передачи. При приеме информации происходит обратная операция преобразования формата принимаемых данных с соответствующим формированием служебных бит.

Управление работой последовательного порта осуществляется через специальный регистр управления/статуса SCON. Наименование разрядов данного регистра представлены на рис. 4.3.1, а назначение – рассмотрено в таблице 4.3.1.

MSB				LSB			
SM0	SM1	SM2	REN	TB8	RB8	TI	RI
Старший				Младший			

Рис. 4.3.1. Наименование разрядов регистра управления/статуса SCON последовательного интерфейса.

Данный регистр содержит управляющие биты, определяющие режим работы последовательного порта (SM0, SM1 и SM2, REN). А также включает в себя 9-ый бит принимаемых и передаваемых данных (RB8 и TB8) и биты прерывания приемопередатчика (RI и TI).

Последовательный порт микроконтроллера I8051 может работать в четырех различных режимах 0,1,2 и 3, определяемых битами управления режимом работы SM0 и SM1 в регистре управления/статуса SCON.

Таблица 4.3.1. Регистр управления/статуса последовательного интерфейса SCON.

Сим-вол	Позиция	Имя и назначение		
SM0 SM1	SCON.7	Биты управления режимом работы последовательного порта. Устанавливаются/сбрасываются программно.		
	SCON.6	SM0	SM1	Режим работы ПП
		0	0	сдвигающий регистр
		0	1	8 бит - изменяемая скорость передачи
		1	0	9 бит - фиксированная скорость
		1	1	9 бит - изменяемая скорость передачи
SM2	SCON.5	Бит управления режимом последовательного порта. Устанавливается программно для запрета приема сообщения, в котором девятый бит имеет значение 0.		
REN	SCON.4	Бит разрешения приема. Устанавливается/сбрасывается программно для разрешения/запрета приема последовательных данных		
TB8	SCON.3	Передача бита 8. Устанавливается/сбрасывается программно для задания девятого передаваемого бита в режиме последовательного интерфейса 9 бит.		
RB8	SCON.2	Прием бита 8. Устанавливается/сбрасывается аппаратурно для фиксации девятого принимаемого бита в режиме последовательного интерфейса 9 бит.		
TI	SCON.1	Флаг прерывания передатчика. Устанавливается аппаратурно при окончании передачи байта. Сбрасывается программно после обслуживания прерывания.		
RI	SCON.0	Флаг прерывания приемника. Устанавливается аппаратурно при приеме байта. Сбрасывается программно после обслуживания прерывания.		

В режиме 0 информация передается и принимается через внешний вывод входа приемника (RXD - линия порта ввода/вывода P3.0). Через вывод выхода передатчика (TXD - линия порта ввода/вывода P3.1) осуществляется передача синхроимпульсов, стробирующих принимаемые или передаваемые биты. Формат посылаемых данных - 8 бит. Частота

приема/передачи бита информации равна $1/12$ частоты кварцевого резонатора.

Передача начинается по любой команде, по которой в SBUF поступает байт данных (рис. 4.3.2). В момент времени S6P2 устройство управления микроконтроллера по сигналу "Запись в буфер" записывает байт в сдвигающий регистр передатчика и запускает блок управления передачей, который через один машинный цикл вырабатывает разрешающий сигнал "Посылка". При этом в момент S6P2 каждого машинного цикла содержимое сдвигающего регистра под воздействием сигнала "Сдвиг" сдвигается вправо (младшими разрядами вперед) и поступает на вывод RXD (выходные данные). В освобождающиеся биты сдвигового регистра записываются нули. При получении сигнала об освобождении сдвигового регистра блок управления снимает сигнал "Посылка" и устанавливает флаг TI окончания передачи. Одновременно формируется сигнал синхронизации сдвига на выводе микроконтроллера TXD.

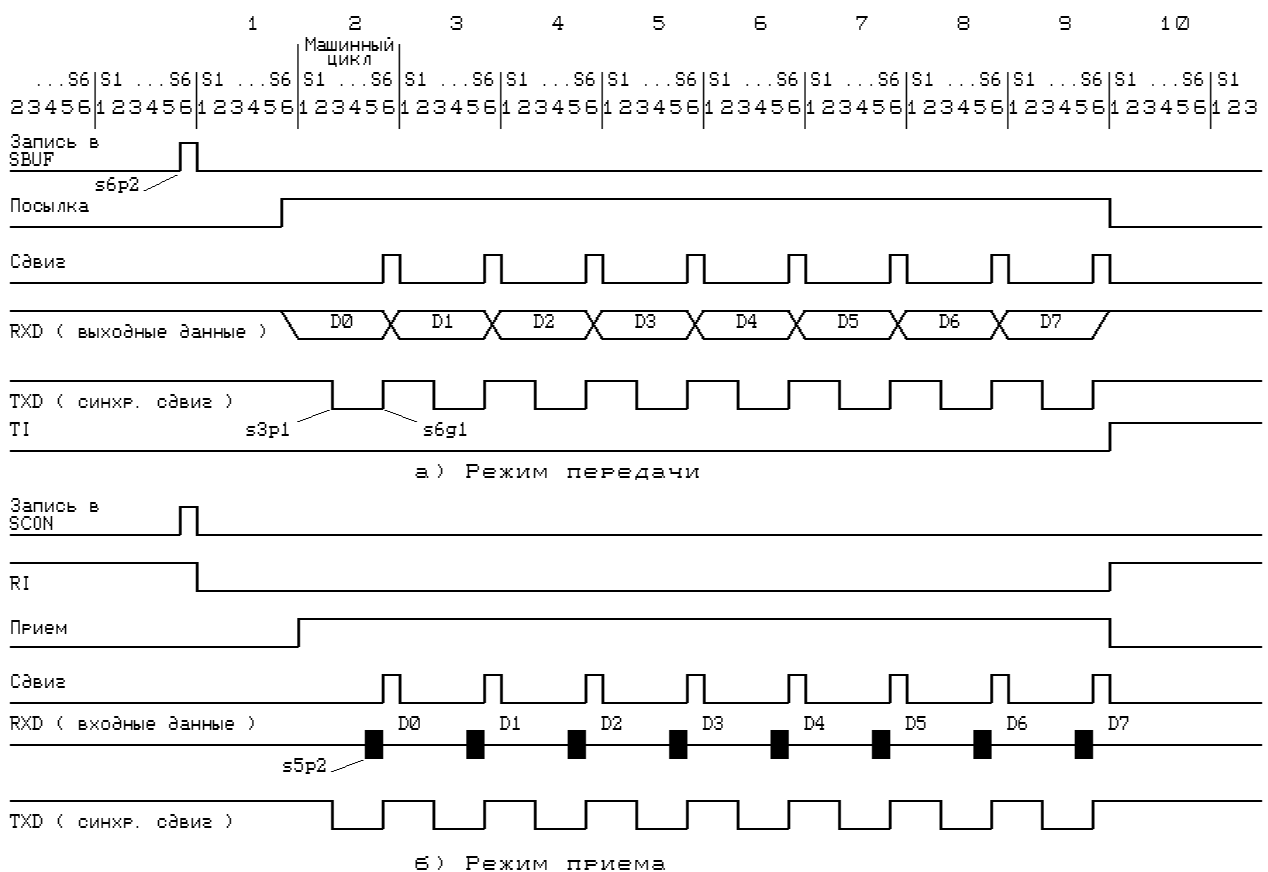


Рис. 4.3.2. Временная диаграмма работы последовательного порта в режиме 0.

Прием данных возможен при условии установки бита разрешения приема $REN=1$ и флага приема данных $RI=0$ в регистре SCON (таблица 4.3.1). После чего в следующем машинном цикле блок управления формирует разрешающий сигнал "Прием", по которому на выход TXD пере-

даются синхросигналы сдвига и в сдвигающем регистре приемника начинают формироваться значения бит данных, которые считываются с входа RXD в моменты S5P2 каждого машинного цикла. Сдвиг данных в регистре осуществляется сигналом "Сдвиг". После окончания приема блок управления переписывает содержимое сдвигающего регистра в буфер, снимает сигнал "Прием" и устанавливает флаг приема данных RI.

В режиме 1 информация передается через выход TXD, а принимается через вход RXD. Формат передаваемой информации – 10 бит (стартовый – "0", 8 – информационных бит и стоповый – "1"). Частота приема и передачи задается программированием таймера/ счетчика 1. На рис. 4.3.3 представлена временная диаграмма работы последовательного порта в режиме 1.

Передача начинается также по любой команде микроконтроллера, по которой в SBUF поступает байт данных. По сигналу "Запись в буфер", генерируемому в данный момент, записывается 1 в девятый бит сдвигающего регистра передатчика и запускается блок управления передачей, который формирует разрешающий сигнал "Посылка". По этому сигналу на вывод TXD сначала поступает "нулевой" старт – бит, а затем по разрешающему сигналу "Данные" – биты данных. Длительность битов данных равен 16 тактам внутреннего счетчика.

Прием начинается при обнаружении перехода сигнала на входе RXD из состояния 1 в состояние 0. Опрос входа RXD осуществляется 16 раз за период представления бита. При обнаружении стартового бита внутренний счетчик по модулю 16 сбрасывается для привязки к фазе входного сигнала. В состояниях 7,8 и 9 счетчика происходит опрос сигнала на входе RXD. Считанное значение принимаемого бита – это то, которое было получено по меньшей мере дважды из трех принятых значений в данном бите. При ложном приеме фронта стартового бита блок управления возвращается к поиску перехода из 1 в 0. Таким образом, обеспечивается подавление ложных выбросов по старт-биту. При приеме истинного стартового бита осуществляется прием остальных бит посылки. Блок управления устанавливает бит RB8 и формирует флаг RI, если в последнем такте сдвига выполняются два условия: бит RI=0, и либо SM2=0 (таблица 4), либо принятый стоп-бит равен 1. При несоблюдении данных условий произошла ошибка приема, и принятая последовательность бит сбрасывается и флаг приема RI не устанавливается.

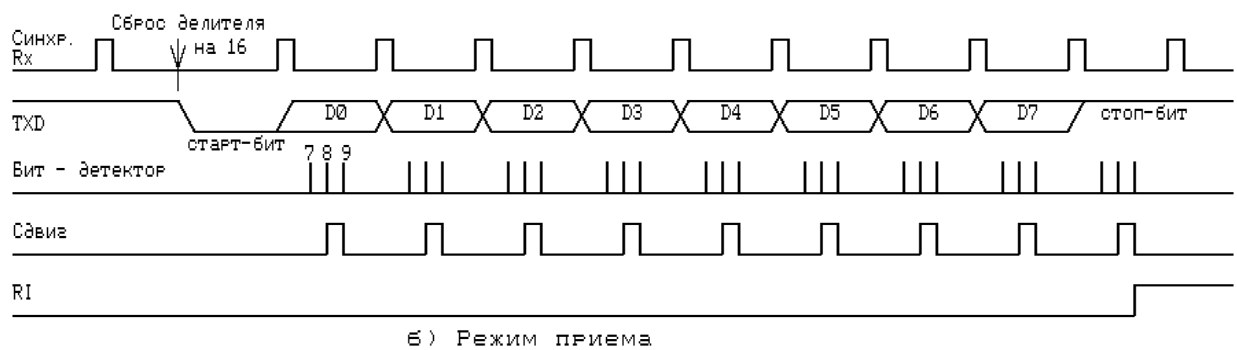
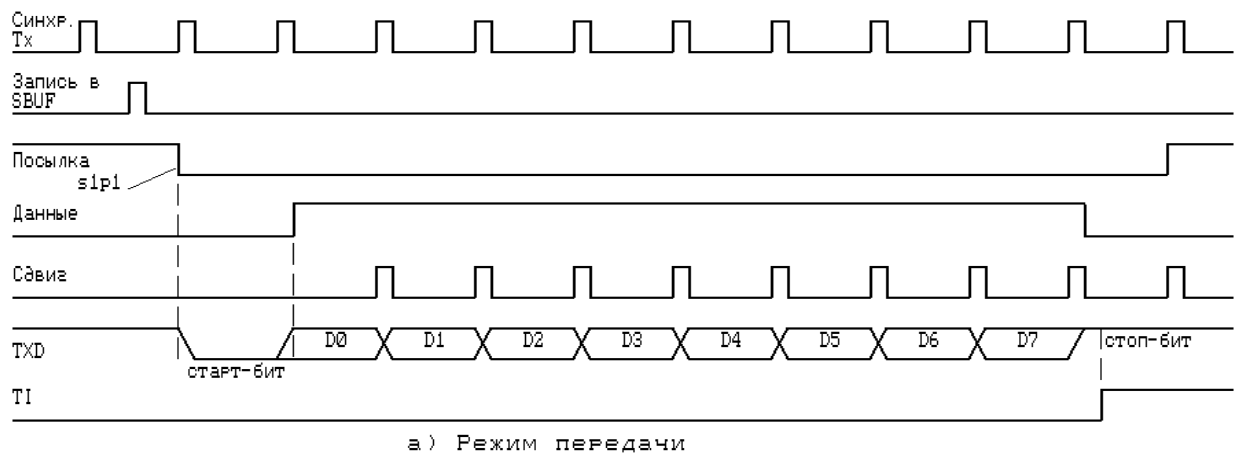


Рис. 4.3.3. Временная диаграмма работы последовательного порта в режиме 1.

В режимах 2 и 3 осуществляется передача информации по выходу TXD и прием по входу RXD. Формат посылки - 11 бит (стартовый - "0", 8 - информационных бит, программируемый 9-ый бит и стоповый -

"1"). При передаче 9-ый бит транслируется из бита TB8 регистра SCON, а при приеме - передается в бит RB8 регистра SCON. Данный бит чаще всего применяется для повышения достоверности передаваемой информации путем контроля по четности или нечетности. В данный бит может быть записано значение признака паритета из словостояния программы (PSW.0). Также в многопроцессорных системах 9-ый бит используется для идентификации адреса приемника передаваемой информации. Частота передачи задается программно и равна $1/32$ или $1/64$ частоты кварцевого резонатора в зависимости от управляющего бита SMOD регистра управления PCON (таблица 4.5.1). Как следует из рис. 4.3.4, представляющего временные диаграммы работы последовательного порта в режимах 2 и 3, работа в этих режимах отличается от работы в режиме 1 только наличием девятого программируемого бита. Вследствие этого изменяется условие окончания цикла приема: блок управления загружает бит RB8 и формирует флаг RI, если в последнем такте сдвига выполняются два условия: бит $RI=0$, и либо $SM2=0$, либо значение принятого девятого бита данных равно 1.

Режим 3 идентичен режиму 2 с тем отличием, что частота приема/передачи программируется таймером/счетчиком 1.

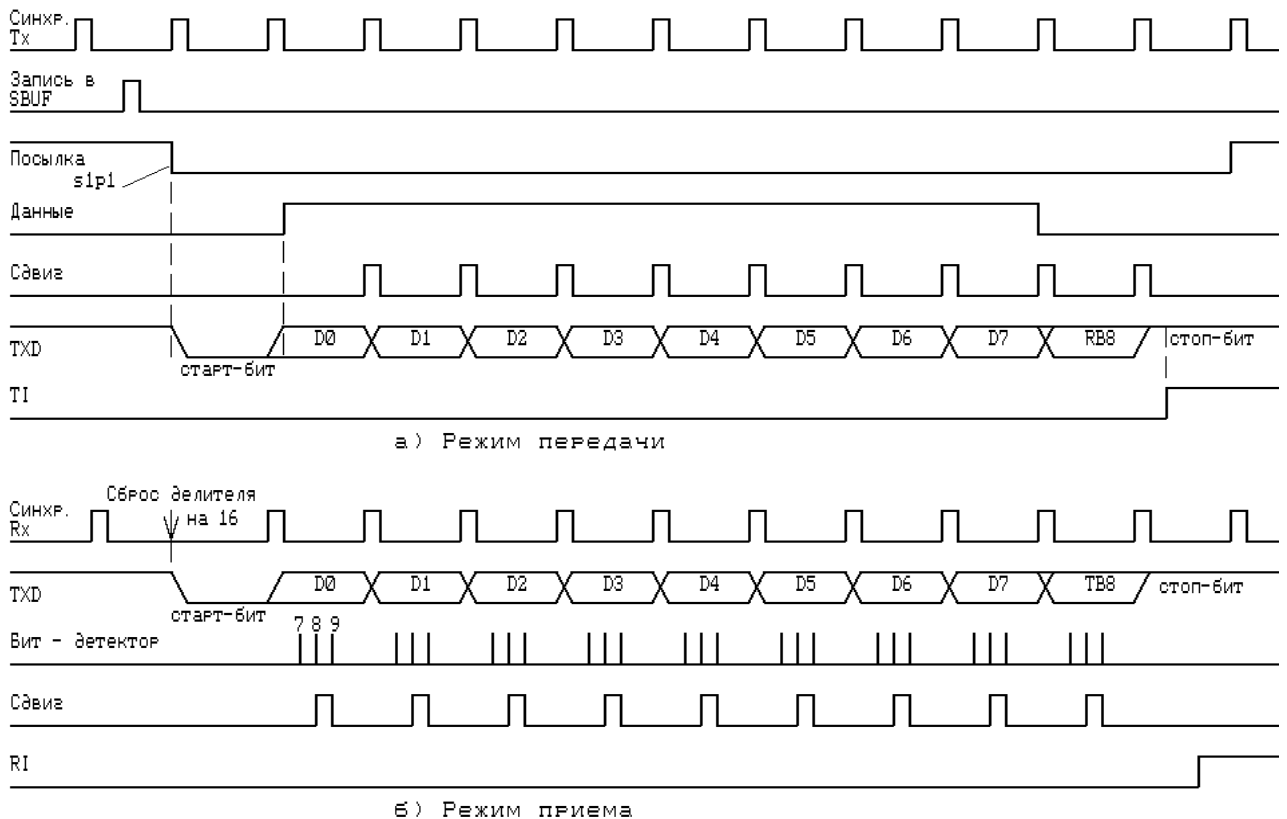


Рис. 4.3.4. Временная диаграмма работы последовательного порта в режиме 2 и 3.

Скорость приема/передачи последовательного интерфейса определяется режимом работы интерфейса и значением управляющего бита SMOD в регистре специальных функций (таблица 4.5.1). В режиме 0 частота передачи зависит только от резонансной частоты кварцевого резонатора $F_0 = F_{\text{рез}}/12$, и поэтому скорость передачи одного бита информации равна одному машинному циклу.

В режимах 1, 2 и 3 скорость обмена информацией зависит от значения бита SMOD в регистре управления PCON (таблица 4.5.1). В режиме 2 частота передачи определяется выражением $F_2 = (2\text{SMOD}/64) * F_{\text{рез}}$ и равна при SMOD=0 $(1/64) * F_{\text{рез}}$, а при SMOD=1 $(1/32) * F_{\text{рез}}$. В режимах 1 и 3 в формировании частоты обмена кроме бита SMOD участвует таймер 1. При этом частота передачи зависит от частоты переполнения FT1 таймера 1 и определяется выражением $F_{1,3} = (2\text{SMOD}/32) * FT1$. Наиболее удобно использовать работу таймера 1 с автоперезагрузкой, т.е. старшая тетрада TMOD = 0010b (таблица 4.2.1). При этом частота передачи определяется выражением $F_{1,3} = (2\text{SMOD}/32) * (F_{\text{рез}}/12) * (256 - (TH1))$. На рис. 4.3.5 представлена схема синхронизации последовательного интерфейса в режиме 2 от внутреннего задающего генератора при $F_{\text{вх}} = F_{\text{рез}}/2$ и в режиме 3 от таймера при $F_{\text{вх}} = FT1$.

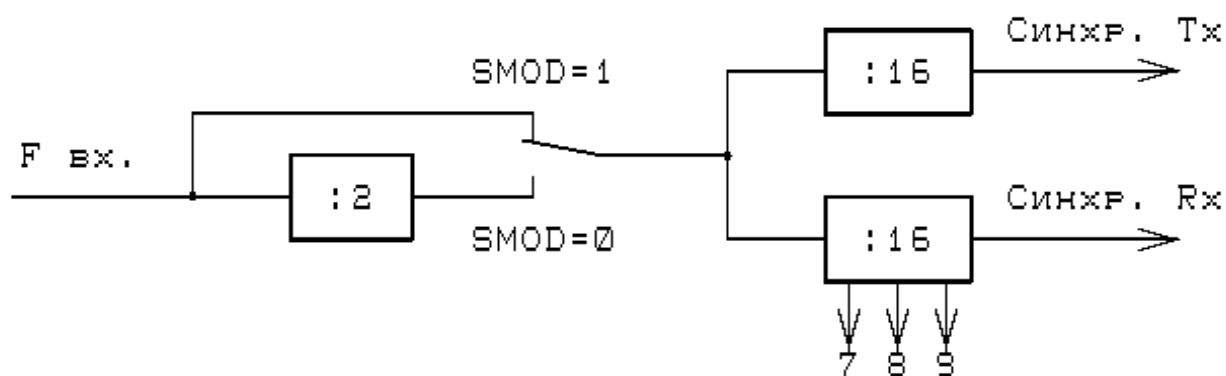


Рис. 4.3.5. Схема синхронизации последовательного интерфейса.

4.4 Организация прерываний

Однокристальная микроконтроллера серии I8051 имеет блок двух-уровневого векторного прерывания от пяти источников:

- два внешних прерывания INT0 и INT1,
- два прерывания от таймеров/счетчиков TF0 и TF1,
- прерывание последовательного порта TI или RI.

Упрощенная схема прерываний микроконтроллера показана на рис. 4.4.1.

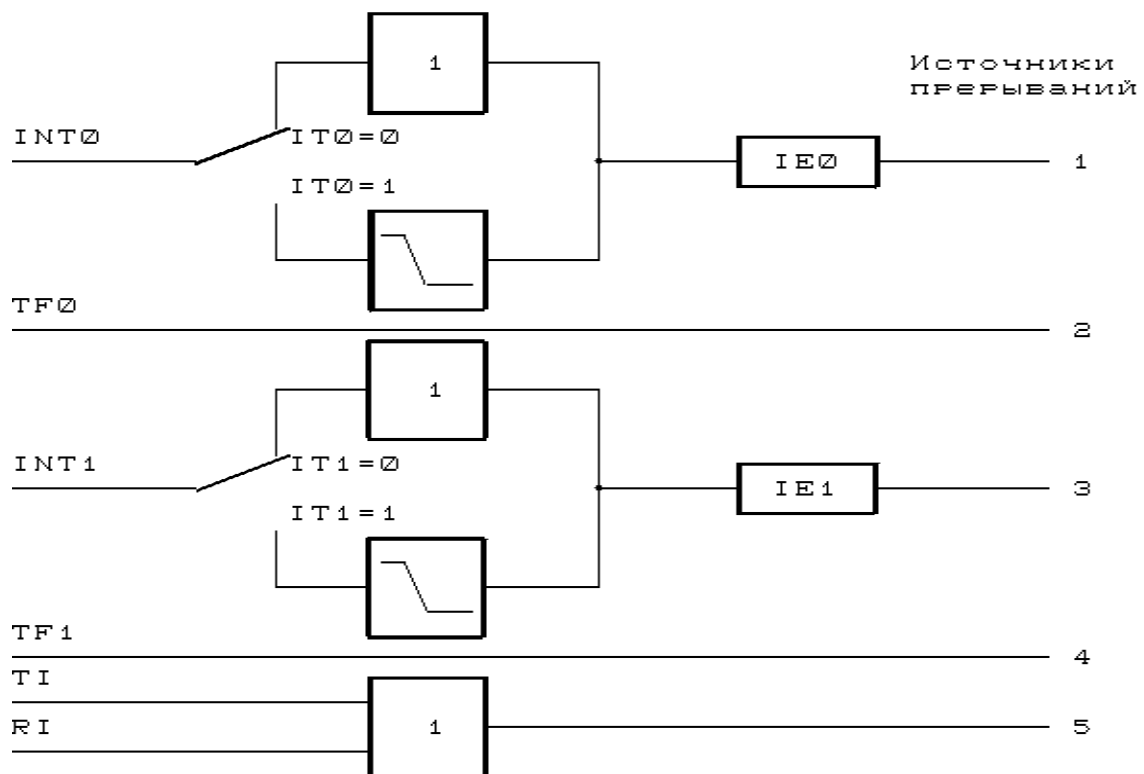


Рис. 4.4.1. Схема прерываний микроконтроллера

Управление прерываниями осуществляет регистр управления/ статуса таймера/счетчика TCON, назначение разрядов которого представлено на рис.4.2.2 и в таблице 4.2.2. Этот регистр управляют типом внешних прерываний (потенциал или переход) и содержит флаги прерываний от внешних источников: INT0 и INT1, и внутренних источников: таймеров/счетчиков T0 и T1, а также приемника RI и передатчика TI последовательного интерфейса.

Внешние прерывания INT0 и INT1 могут быть вызваны либо уровнем, либо переходом сигнала из 1 в 0 на входах микроконтроллера в зависимости от значений управляющих бит IT0 (TCON.0) и IT1 (TCON.2). При наличии прерывания от внешнего источника в регистре

TCON устанавливаются флаги IE0 (TCON.1) и IE1 (TCON.3) в зависимости от входа прерывания. Данные флаги инициируют вызов соответствующей подпрограммы обслуживания прерывания. Сброс этих флагов выполняется аппаратурно только в том случае, если прерывание было вызвано по переходу (срезу) сигнала. Если же прерывание вызвано уровнем входного сигнала, то сбросом флага IE управляет соответствующая подпрограмма. При обслуживании такого типа прерывания необходимо воздействовать на источник прерывания путем с целью снятия им запроса.

При прерывании от таймеров устанавливаются флаги запросов прерывания TF0 (TCON.5) и TF1 (TCON.7). Эти флаги сбрасываются автоматически при передаче управления подпрограмме обслуживания.

При прерывании от последовательного интерфейса флаги запросов прерывания устанавливаются в регистре управления/статуса SCON, назначение разрядов которого представлено на рис.4.3.1 и в таблице 4.3.1. Этот регистр управляет флагами приемника RI (SCON.0) и передатчика TI (SCON.1) последовательного интерфейса. Указанные флаги устанавливаются блоком последовательного интерфейса аппаратурно после окончания процесса передачи (флаг TI) или после окончания приема сигнала (флаг RI). Сброс данных флагов должен осуществляться программой обработки соответствующего прерывания.

Прерывания могут быть вызваны или отменены программой пользователя, так как все перечисленные флаги доступны программно и могут быть установлены или сброшены программой в любой момент времени.

В блоке регистров специальных функций имеется два регистра: предназначенных для управления режимом прерываний и уровнями приоритета. Это регистр разрешения прерываний IE, наименование разрядов которого представлено на рис. 4.4.2, а назначение - в таблице 4.4.1, и регистр приоритетов прерываний IP, представленный соответственно на рис. 4.4.3 и в таблице 4.4.2. Возможность программной установки/сброса любого управляющего бита в этих двух регистрах делает систему прерываний микроконтроллера исключительно гибкой.

MSB

LSB

EA	–	–	ES	ET1	EX1	ET0	EX0
Старший				Младший			

Рис. 4.4.2. Наименование разрядов регистра разрешения прерываний IE.

Таблица 4.4.1. Регистр разрешения прерываний

Символ	Позиция	Имя и назначение
EA	IE.7	Снятие блокировки прерываний. Сбрасывается программно для запрета всех прерываний независимо от состояний IE4 – IE0
	IE.6 IE.5	Не используются
ES	IE.4	Бит разрешения прерывания от последовательного порта. Установка/ сброс программой для разрешения/ запрета прерываний от флагов TI или RI
ET1	IE.3	Бит разрешения прерывания от таймера 1. Установка/сброс программой для разрешения/запрета прерываний от таймера 1.
EX1	IE.2	Бит разрешения внешнего прерывания 1. Установка/сброс программой для разрешения/запрета прерываний.
ET0	IE.1	Бит разрешения прерывания от таймера 0. Установка/сброс программно для разрешения/запрета прерываний от таймера 0.
EX0	IE.0	Бит разрешения внешнего прерывания 0. Установка/сброс программой для разрешения/запрета прерываний.

-	-	-	PS	PT1	PX1	PT0	PX0
Старший			Младший				

Рис. 4.4.3. Наименование разрядов регистра приоритетов прерываний IP.

Таблица 4.4.2. Регистр приоритетов прерываний

Символ	Позиция	Имя и назначение
	IP.7 IP.6 IP.5	Не используются
PS	IP.4	Бит приоритета прерывания последовательного интерфейса. Установка/сброс программой для присваивания прерыванию от последовательного интерфейса высшего/низшего приоритета.
PT1	IP.3	Бит приоритета прерывания таймера. 1. Установка/сброс программой для присваивания прерыванию от таймера 1 высшего/низшего приоритета.
PX1	IP.2	Бит приоритета внешнего прерывания 1. Установка/сброс программой для присваивания прерыванию внешнему прерыванию INT1 высшего/низшего приоритета.
PT0	IP.1	Бит приоритета прерывания таймера 0. Установка/сброс программой для присваивания прерыванию от таймера 0 высшего/низшего приоритета.
PX0	IP.0	Бит приоритета внешнего прерывания 0. Установка/сброс программой для присваивания прерыванию внешнему прерыванию INT0 высшего/низшего приоритета.

Каждый из указанных источников прерываний может быть индивидуально разрешен или запрещен установкой или сбросом соответствующего бита в регистре разрешения прерываний IE. Регистр IE содержит также бит EA, сброс которого запрещает все прерывания. С помощью регистра IP каждому прерыванию может быть установлен один из двух уровней приоритета: высокий или низкий. Это производится установ-

кой или сбросом соответствующего бита в регистре IP. Программа обработки прерывания с низким приоритетом может быть прервана программой с высоким приоритетом, но не может быть прервана программой с низким приоритетом. При одинаковом приоритете обработка производится в следующей последовательности:

Источник Приоритет

1. IE0 высший
2. TF0
3. IE1
4. TF1
5. RI+TI низший

Опрос флагов прерываний производится в момент S5P2 каждого машинного цикла. Ранжирование прерываний по уровню приоритета выполняется в течение следующего машинного цикла. Система прерываний сформирует аппаратно вызов (LCALL) соответствующей подпрограммы обслуживания, если она не заблокирована одним из следующих условий:

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

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

По аппаратно сформированному коду LCALL система прерывания помещает в стек только содержимое счетчика команд (PC) и загружает в счетчик команд адрес вектора соответствующей подпрограммы обслуживания, представленной на рис. 4.4.4. По адресу вектора должна быть расположена команда безусловной передачи управления (JMP) к начальному адресу подпрограммы обслуживания прерывания. Подпрограмма обслуживания в случае необходимости должна начинаться командами записи в стек (PUSH) слова состояния программы (PSW), аккумулятора, расширителя, указателя данных и других регистров и заканчиваться командами восстановления из стека (POP). Подпрограммы обслуживания прерывания обязательно завершаются командой RETI, по

которой в счетчик команд перезагружается из стека сохраненный адрес возврата в основную программу. Команда RET также возвращает управление прерванной основной программе, но при этом не снимает блокировку прерываний, что приводит к необходимости иметь программный механизм анализа окончания процедуры обслуживания данного прерывания.

		002Bh
Serial port		0023h
Timer 1	Память программ	001Bh
INT1		0013h
Timer 0		000Bh
INT0		0003h
Стартовый адрес L		0000h

Рис. 4.4.4. Адреса векторов прерываний

5. БЛОК УПРАВЛЕНИЯ

Блок управления микроконтроллера предназначен для выработки синхронизирующих и управляющих сигналов, обеспечивающих координацию работы всех внутренних ресурсов микроконтроллера. В состав блока управления входят: устройство синхронизации, регистр управления потреблением и логика управления микроконтроллера.

Синхронизация работы микроконтроллера осуществляется от генератора с внешним кварцевым резонатором или от внешнего источника тактовых сигналов. Кварцевый резонатор, подключаемый к внешним выводам X1 и X2 корпуса микроконтроллера, управляет работой внутреннего генератора, который в свою очередь формирует сигналы синхронизации.

устройство управления микроконтроллер на основе сигналов синхронизации формирует машинный цикл фиксированной длительности, равно 12 периодам резонатора или шести состояниям первичного управляющего автомата (S1-S6). Каждое состояние управляющего автомата содержит две фазы (P1, P2) сигналов резонатора. В фазе P1 выполняется операция в АЛУ, а в фазе P2 осуществляется межрегистровая передача. Весь машинный цикл состоит из 12 фаз, начиная с фазы S1P1 и кончая фазой S6P2, как показано на рис. 5.1.



Рис. 5.1. Диаграмма формирования машинных циклов микроконтроллера.

На диаграмме представлены сигналы резонатора BQ и строба адреса внешней памяти ALE. сигнал ALE формируется дважды за один машинный цикл (S1P2-S2P1 и S4P2-S5P1) и используется для управления процессом обращения к внешней памяти.

Большинство команд микроконтроллер выполняется за один машинный цикл. Некоторые команды, оперирующие с 2-байтными словами или связанные с обращением к внешней памяти, выполняются за два машинных цикла. Только команды деления и умножения требуют четырех машинных циклов. На основе этих особенностей работы устройства управления микроконтроллер производится расчет времени исполнения прикладных программ.

Конструкция регистра управления мощностью PCON определяется технологией изготовления микроконтроллера: n-МОП или КМОП. Для варианта изготовления по технологии n-МОП (серия I8051) регистр PCON имеет всего 1 бит – SMOD, управляющий скоростью передачи последовательного порта. Для варианта изготовления по технологии КМОП наименование разрядов данного регистра представлены на рис. 5.2, а назначение – рассмотрено в таблице 5.1.

Функции бита SMOD рассмотрены при описании работы последовательного порта в разделе 4.

Биты PCON.4 – PCON.6 зарезервированы для дальнейшего расширения семейства MCS51. При чтении значение этих разрядов не определено. Программист не должен записывать "1" в эти биты, т.к. они могут использоваться в дальнейших разработках однокристальных микроконтроллера семейства MCS51 для задания новых функций.

Биты GF1 и GF0 пользователь может задействовать по своему усмотрению.

MSB				LSB			
SMOD	–	–	–	GF1	GF0	PD	IDL
Старший				Младший			

Рис. 5.2. Наименование разрядов регистра управления PCON

Таблица 5.1. Регистр управления мощностью PCON.

Символ	Позиция	Имя и назначение
SMOD	PCON.7	Удвоенная скорость передачи. Если бит установлен в 1, то скорость передачи вдвое больше, чем при SMOD = 0.
	PCON.6 PCON.5 PCON.4	Не используются

GF1 GF0	PCON.3 PCON.2	Флаги, специфицируемые пользователем (флаги общего назначения)
PD	PCON.1	Бит пониженной мощности. При установке бита в 1 микроконтроллер переходит в режим пониженной потребляемой мощности.
IDL	PCON.0	Бит холостого хода. Если бит установлен в 1, то микроконтроллер переходит в режим холостого хода.

В однокристальных микроконтроллера семейства MCS51, выполненных по КМОП технологии, имеются два режима уменьшенного энергопотребления: режим холостого хода и режим микропотребления. Источником питания в этих режимах является вывод U_{ss} . Режимы уменьшенного потребления для КМОП однокристальная микроконтроллера индицируются установкой битов PD и IDL в регистре PCON. Воздействие этих битов на аппаратуру микроконтроллера показано на рис. 5.3. Если в PD и IDL одновременно записана "1", преимущество имеет бит PD.

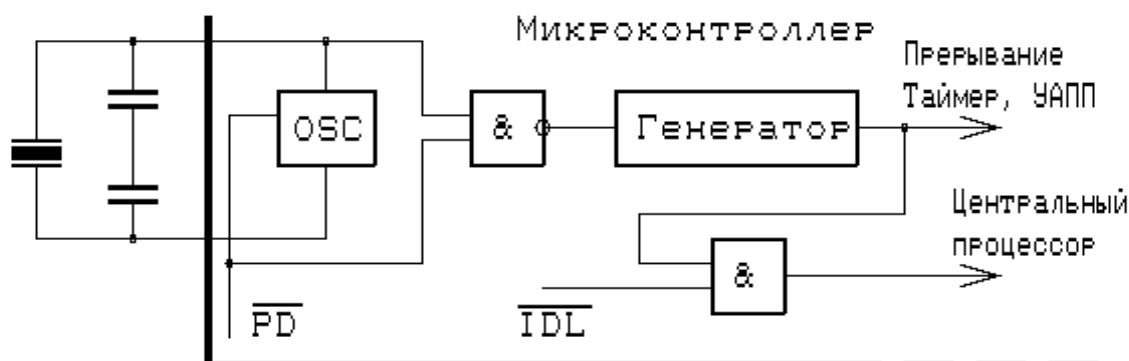


Рис. 5.3. Действие битов PD и IDL регистра PCON.

Для перехода в режим холостого хода необходима инструкция, которая устанавливает $PCON.0=1$ (IDL). В этом режиме блокируются функциональные узлы центрального процессора (CPU), что и уменьшает энергопотребление микросхемы. Сохраняются состояния указателя стека, программного счетчика, PSW, аккумулятора и всех других регистров, а также внутреннего ОЗУ данных.

Для окончания режима холостого хода имеются два способа. Активизация любого разрешенного прерывания автоматически приведет к установке $PCON.0=0$ и окончанию режима холостого хода. После исполнения команды RETI (выход из подпрограммы обслуживания прерывания) будет выполнена команда, которая следует за командой, переведшей

однокристалльную микроконтроллера в режим холостого хода. Биты GF0 и GF1 удобно использовать для индикации режима, в котором была вызвана программа обработки прерывания: произошло это при нормальной работе однокристалльного микроконтроллера или в режиме холостого хода. К примеру, команда, вызывающая режим холостого хода, может также устанавливать один или несколько флагов (GF0, GF1 или каких-либо других). Программа обработки прерывания, проверяя эти флаги, может определить предысторию своего вызова.

Другим способом окончания режима холостого хода является аппаратный сброс по входу RST длительностью не менее двух машинных циклов. Активный сигнал сброса на выводе RST асинхронно сбрасывает бит IDL (PCON.0). Поскольку тактовый генератор работает, однокристалльный микроконтроллер сразу после сброса IDL начинает выполнять программу с команды, следующей за командой, вызвавшей режим холостого хода. Между сбросом бита IDL и моментом, когда включится внутренний алгоритм сброса, может пройти до двух машинных циклов выполнения программы. Внутренние аппаратные средства микроконтроллера блокируют доступ к внутренней памяти данных в течение указанного времени, но не блокируют доступ к портам. Если при этом изменение информации на портах нежелательно, то необходимо следить, чтобы за командой, которая устанавливает бит IDL, не следовала непосредственно команда, записывающая информацию в порт или во внешнюю память данных.

Для перехода в режим микропотребления необходима инструкция, которая устанавливает PCON.1=1 (PD). В этом режиме задающий генератор выключается, прекращая тем самым работу всех узлов микроконтроллера, и сохраняется только содержимое ОЗУ. Единственным выходом из этого состояния является аппаратный сброс RST. В этом режиме работы напряжение UCC может быть уменьшено до 2В, и должно быть восстановлено до номинального перед выходом из режима микропотребления. Сброс следует удерживать в активном состоянии не менее 10 мс (время восстановления работы задающего генератора). При одновременной установке IDL=1 и PD=1 преимущество имеет бит PD.

В таблице 5.2 представлены состояния выводов микроконтроллера в режимах холостого хода и микропотребления.

Для микроконтроллеров серии I8051 (n-MOP) существует режим пониженного потребления. Во время нормальной работы внутреннее ОЗУ питается от UCC. Однако для микроконтроллера семейства MCS51, если

напряжение на выводе RST превышает UCC, оно становится источником питания для ОЗУ. Это реализовано с помощью двух внутренних диодов, с катодов которых берется питание ОЗУ, а аноды подключены соответственно к входу RST и к выводу питания однокристалльная микроконтроллера Ucc. Необходимо подчеркнуть, что для КМОП однокристалльная микроконтроллера данный режим отсутствует.

Логика управления ЭВМ в зависимости от режима работы однокристалльная микроконтроллера вырабатывает необходимый набор управляющих сигналов.

Таблица 5.2. Состояния выводов однокристалльная микроконтроллера в режимах холостого хода и микропотребления.

Режим	Память программ	ALE	PSEN	Порт P0	Порт P1	Порт P2	Порт P3
Холостого хода	Внутренняя	1	1	Данные	Данные	Данные	Данные
Холостого хода	Внешняя	1	1	Z	Данные	Адрес	Данные
Микропотребления	Внутренняя	0	0	Данные	Данные	Данные	Данные
Микропотребления	Внешняя	0	0	Z	Данные	Данные	Данные

6. НАЧАЛЬНАЯ УСТАНОВКА

Инициализация (сброс) микроконтроллера осуществляется сигналом RST (активный – высокий уровень напряжения) при условии подачи на микросхему внешнего сигнала синхронизации или при подключении кварца. Вход RST является входом внутреннего триггера Шмита.

Для гарантированного сброса микросхемы необходимо, чтобы длительность сигнала высокого уровня на входе RST была не менее двух машинных циклов микроконтроллера (24 периода частоты синхронизации F). Внешний сигнал сброса является асинхронным по отношению к внутренней синхронизации микроконтроллера. После окончания сигнала сброса на входе RST, т.е. появления уровня "0", проходит от 1 до 2 машинных циклов до начала формирования сигналов ALE и PSEN.

При подаче сигнала сброса на вход RST внутренний алгоритм сброса микроконтроллера производит следующие действия;

- устанавливает счетчик команд PC и все регистры специальных функций, кроме защелок портов P0-P3, указателя стека SP и регистра SBUF, в ноль (в регистрах специальных функций PCON, IP и IE резервные биты принимают случайные значения, а все остальные биты сбрасываются в ноль);

- подготавливает порты P0-P3 для приема данных, т.е. устанавливает фиксаторы-защелки портов P0-P3 в "1";

- указатель стека принимает значение равное 07H;

- в регистрах SBUF устанавливаются случайные значения;

- запрещает все источники прерываний, работу таймеров/счетчиков и последовательного порта;

- выбирает БАНК0 резидентного ОЗУ;

- выводы ALE и PSEN находятся в режиме ввода;

- содержимое резидентной памяти данных не изменяется, а после включения питания содержимое ячеек внутреннего ОЗУ данных принимает случайные значения.

Обобщенные данные по состояниям регистров после сброса адреса указаны в таблице 6.1.

Таблица 6.1.

Регистр	Информация
PC	0000h
ACC	00h
B	00h
PSW	00h
SP	07h
DPTR	0000h
P0-P3	0FFh
IP	XXX00000b
IE	0XX00000b
TMOD	00h
TCON	00h
TH0	00h
TL0	00h
TH1	00h
TL1	00h
SCON	00h
SBUF	Неопределенная
PCON	0XXX0000b для серии 1830 0XXXXXXXXb для серии I8051

Для n-МОП микроконтроллера автоматический сброс при включении питания UCS может быть реализован подключением входа RST к UCS через конденсатор емкостью 10 мкФ и к шине 0В через резистор 8,2 К. Для КМОП микроконтроллера этот резистор не требуется, однако его наличие не принесет вреда. КМОП микроконтроллера содержат внутренний резистор, включенный между RST и выводом 0В. Если использовать только внутренний резистор, емкость конденсатора может быть уменьшена до 1 мкФ.

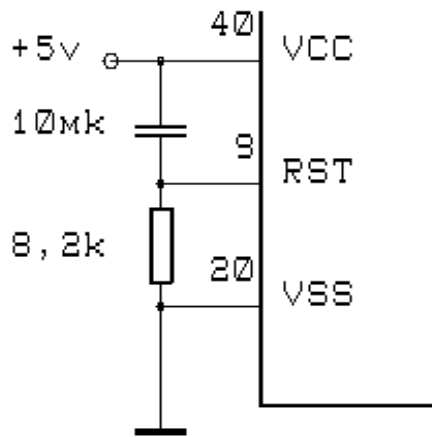


Рис. 6.1. Схема сброса при включении питания.

Включение питания без обеспечения гарантированного сброса может привести к тому, что микроконтроллер начнет выполнение программы с некоторого случайного адреса. Это объясняется тем, что счетчик команд PC не будет сброшен в 0000.

Семейство микроконтроллера I8051 содержит набор микросхем с внутренней памятью программ. Запись данных во внутреннюю память программ осуществляется в соответствии со стандартным алгоритмом программирования. Содержимое резидентной памяти программ для некоторых микросхем данной серии может быть "стерто" выдержкой под ультрафиолетовым источником света, а затем запрограммировано заново. Микроконтроллер имеет также средства защиты от несанкционированного считывания записанной программы с помощью установки битов защиты.

Программирование микроконтроллера осуществляется на частоте задающего генератора частотой 4–6 МГц. Генератор необходим для работы внутренней шины микроконтроллера, по которой происходит пересылка адреса и данных в соответствующие внутренние регистры. В таблице 7.1 приведены режимы работы микроконтроллера при программировании внутреннего ППЗУ микросхем серии I8051 (I8751).

Таблица 7.1. Режимы работы при программировании

Режим	RST	PSEN	ALE	EA	P2.7	P2.6	P2.5	P2.4
Программирование	1	0			1	0	0	0
Проверка	1	0	1	1	0	0	0	0
Программирование бита защиты памяти	1	0		U _{PR}	1	1	0	0

В таблице указано:

1 – уровень логической единицы на соответствующем выводе микроконтроллера;

0 – уровень логического нуля на соответствующем выводе микроконтроллера;

$U_{pr} = +21В + 0,5В$;

ALE – подается импульсом низкого логического уровня длительности 50 мс + 5 мс.

На рис. 7.1, 7.2 и 7.3 показаны схемы подачи сигналов на выводы микроконтроллера при работе в режимах, приведенных в таблице 7.1.

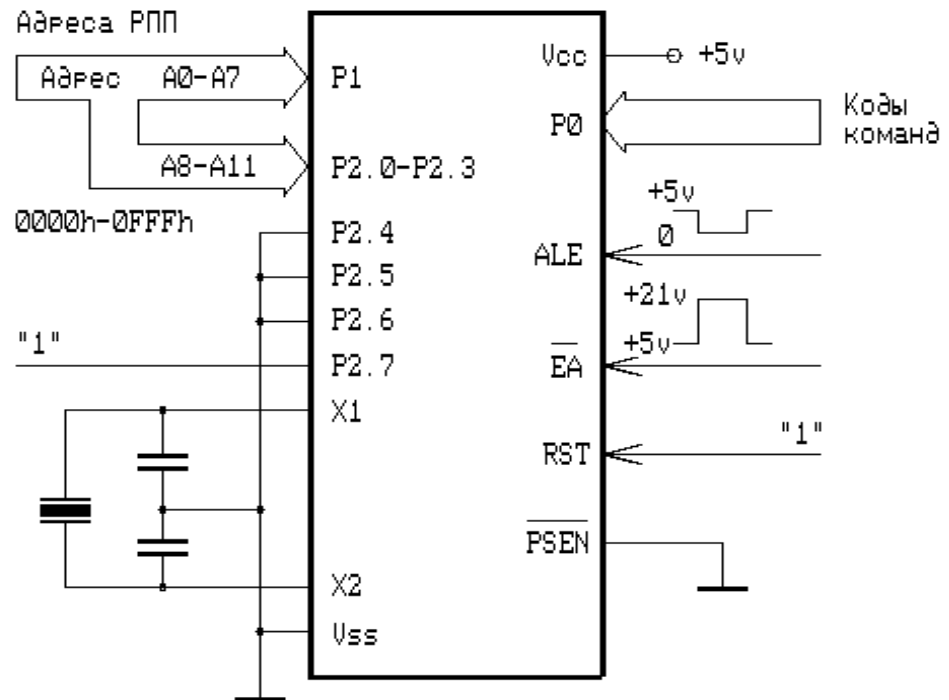


Рис. 7.1. Программирование микроконтроллера

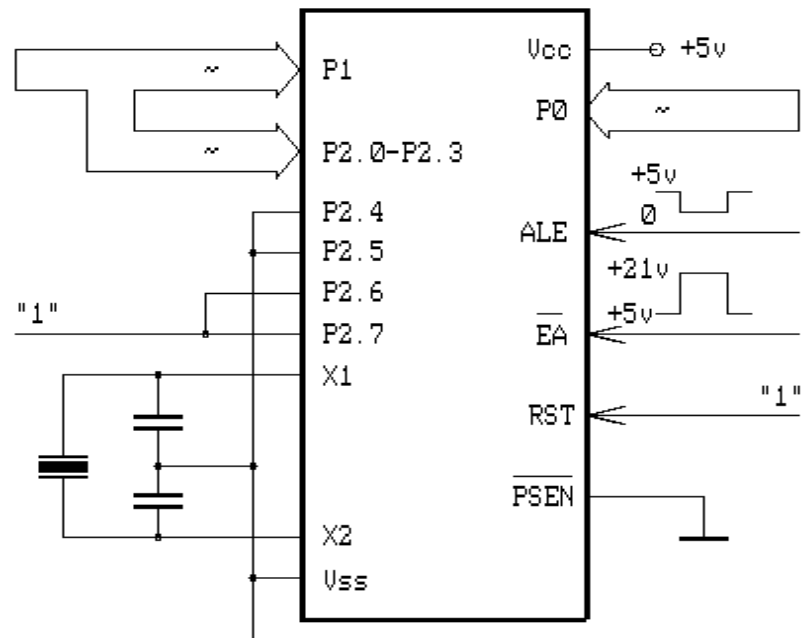


Рис. 7.2. Запись бита защиты.

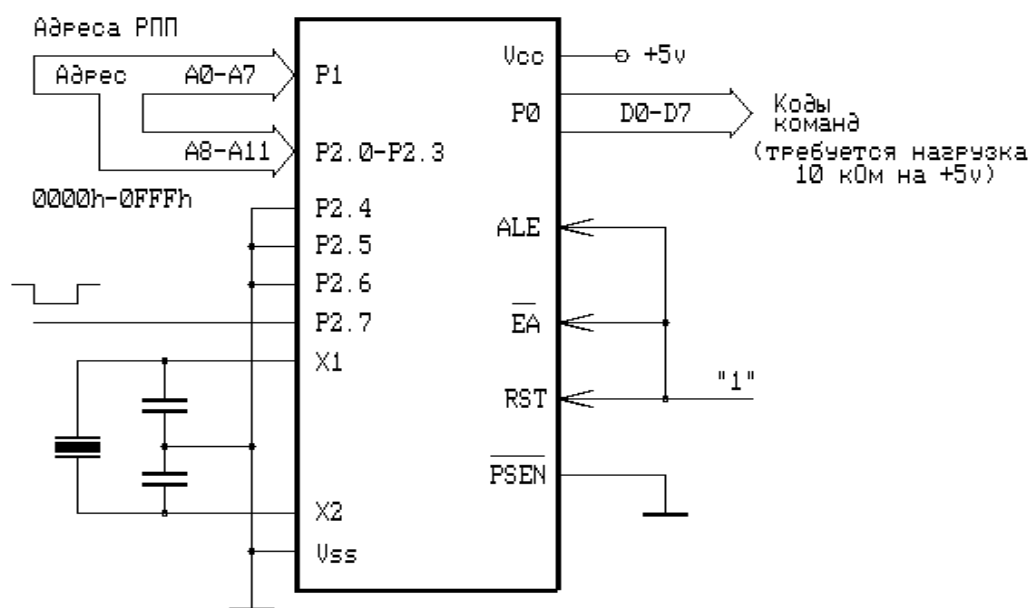


Рис. 7.3. Верификация программ.

В режиме программирования адрес ячейки ППЗУ, в которую необходимо записать информацию, подается на выводы портов P1 и P2. Младшие разряды адреса A0 – A7 подаются соответственно на выводы микроконтроллера P1.0 – P1.7, а старшие разряды адреса A8... A11 – соответственно на выводы P2.0 – P2.3.

Байт данных, который необходимо записать в адресуемую ячейку ППЗУ, подается на выводы порта P0.

На остальные выводы порта P2, а также на выводы микроконтроллера RST, PSEN и EA подаются уровни напряжения, приведенные для режима "Программирование" в табл. 10. Программирование ячейки ППЗУ

происходит при подаче импульса низкого уровня ALE длительностью $50 \text{ мс} + 5 \text{ мс}$. Сигнал EA удерживается в состоянии логической 1 до подачи импульса ALE. Затем напряжение на выводе EA повышается до $21 + 0,5 \text{ В}$, подается импульс ALE и сигнал EA вновь возвращается к уровню логической 1. Возможен также "быстрый" алгоритм программирования. В этом случае формируется сигнал ALE импульсами по 100 мкс, а суммарная длительность не превышает 50 мс. Формирование сигналов ALE прекращается при записи байта данных, полученное при проверке. Даже кратковременное импульсное превышение напряжения на выводе EA уровня $21,5 \text{ В}$ может вызвать необратимый отказ микросхемы. Поэтому источник программируемого напряжения UPR должен быть хорошо стабилизирован.

Режим проверки содержимого внутренней памяти программ возможен только в том случае, если не запрограммирован бит защиты памяти. В данном режиме возможно чтение внешней по отношению к микроконтроллеру аппаратурой содержимого внутренней памяти программ. Чтение возможно либо во время, либо после программирования микросхемы.

Адрес ячейки памяти программ, содержимое которой необходимо прочитать подается на выводы портов P1 и P2 аналогично режиму программирования. Другие выводы микроконтроллера должны поддерживаться в состояниях, приведенных для режима "Проверка" в табл. 7.1 и на рис. 7.3. Содержимое адресуемой ячейки памяти считывается с выводов порта P0 при подаче низкого логического уровня на вывод P2.7. При чтении вывод P2.7 может либо постоянно поддерживаться в состоянии "0", либо использоваться в качестве строб-сигнала чтения с активным низким уровнем. Для работы в данном режиме необходимо использовать подтягивающие резисторы, включенные между выводами порта P0 и напряжением питания.

Свойство защиты памяти заключается в наличии в составе ППЗУ специального бита, который, будучи запрограммирован, запрещает доступ к внутренней памяти программ любыми внешними по отношению к микроконтроллеру средствами. Включение микросхемы в режим программирования бита защиты памяти показано на рис. 7.2.. Процедура программирования в данном случае аналогична обычному программированию ППЗУ микроконтроллера за исключением того, что вывод P2.6 удерживается в состоянии логической 1. Порты P0, P1 и выводы P2.0-P2.3 могут находиться в произвольном состоянии. Остальные выводы микро-

контроллера должны удерживаться в состояниях, приведенных в табл.7.1 для режима "Программирование бита защиты памяти".

Если бит защиты памяти запрограммирован, его можно очистить только полным стиранием всего ППЗУ. При запрограммированном бите защиты памяти внутренняя память программ не может быть прочитана внешними по отношению к микроконтроллеру средствами, дальнейшее программирование ППЗУ становится невозможным и микроконтроллер теряет возможность работы с внешней памятью программ. Стирание ППЗУ очищает бит защиты памяти, и микросхема полностью восстанавливает свои функциональные возможности: может быть перепрограммирована и может работать с внешней памятью программ.

Стирание ППЗУ выполняется ультрафиолетовым излучением с длиной волны меньше 4000 ангстрем. Т. к. солнечный свет, а также лампы дневного света излучают волны в указанном диапазоне, они могут являться причиной порчи содержащейся в ППЗУ информации (критическая продолжительность облучения – 1 неделя на солнце или 3 года в комнате с лампами дневного света). Поэтому рекомендуется закрывать кварцевое окошко микросхемы защитной наклейкой. Рекомендуется стирание ППЗУ ультрафиолетовым излучением (длина волны 2537 ангстрем) с интегральной дозой как минимум 15 Вт-сек/см². При этом для стирания должно быть достаточным помещение микросхемы под ультрафиолетовую лампу с потоком 12000 мкВт/см² на время 20–30 минут и на расстояние около 2,5 см от лампы до кварцевого окошка микросхемы. После стирания все ячейки ППЗУ находятся в состоянии логической единицы.

8.1 Программа преобразования системы счисления (2 -> 2/10)

Входные параметры: исходное число в регистре A

Выходные параметры: результат в R1 и R0

BINBCD:

```
    mov B, #100
    div AB      ;Деление на 100 для определения числа
сотен (результат в A, остаток в B)
    mov R1, A

    mov A, #10
    xch A, B
    div AB      ;Деление на 10 для определения числа де-
сятков (десятки в A, единицы в B)

    swp A       ;Перемещение десятков в старшую тетраду
    add A, B     ;В регистре A число в двоично-десятичном
формате
    mov R0, A
```

8.2 Арифметические операции двойной точности

Входные параметры:

R0 - адрес 1 операнда

R1 - адрес 2 операнда

R2 - размерность чисел

Выходные параметры:

R0 - адрес результата

SUBSTR:

```
    clr C                                ;C=0
```

BEGIN:

```
    mov A,@R0                            ;уменьшаемое
    subb A,@R1                            ;вычитание (addc) - сложение
    mov @R0,A                            ;сохранение результата
    inc R0                                ;изменение указателей
    inc R1
    djnz R2,BEGIN                        ;цикл по R2
```

8.3 Вычисление синуса табличным методом

Входные параметры:

A - угол в градусах

Выходные параметры:

A - синус угла

Версия 1

sin:

```
100      inc A                ;наличие команды RET
101      movc A,@A+PC        ;PC=102
102      ret
```

sinus:

```
103      db 0                ;0
104      db 00000100b        ;0,017
105      db 00001001b        ;0,035
        .....
192      db 11111111b        ;0,999
```

Версия 2

sin:

```
100      mov DPTR, offset SINUS
101      movc A,@A+DPTR
102      ret
```

;

;

;

;

sinus:

```
XXX      db 0                ;0
        db 00000100b        ;0,017
        db 00001001b        ;0,035
        .....
        db 11111111b        ;0,999
```

8.4 Передача данных последовательным кодом

Входные параметры:

A - данные

Выходные параметры:

P1.0 - выходные данные

P1.1 - выходной синхросигнал

```

        mov R7,#8
loop:   rrc A      ;RLC
        mov P1.0,C
        ;nop
        ;nop
        ;nop
        setb P1.1
        clr P1.1
        djnz R7,loop

```

8.5 Передача данных манчестерским кодом

Каждый бит передается двумя интервалами - инверсным и прямым (Длительность одного интервала - 6 мкс)

Входные параметры:

A - данные

Выходные параметры:

P1.0 - выходные данные

```

        mov R7,#8
loop:   rrc A
        cpl C
        mov P1.0,C
        cpl C
        nop
        nop
        nop
        mov P1.0,C
        djnz R7,loop

```

8.6 Передача и прием данных через последовательный порт

Входные параметры:

A - данные

P - признак четности

Передача методом опроса готовности

S_P_OUT:

```

    mov C,P
    mov TB9,C
    jnb TI,$    ;M1: jnb TI,M1
    clr TI
    mov SBUF,A
    ret

```

Передача методом прерываний

```

INT_RSO:                ;вектор прерывания
    clr TI
    mov SBUF,A
    reti

```

Прием методом опроса готовности (с ожиданием)

SPIN:

```

    jnb RI,SPIN    ;jnb RI,$
    clr RI
    mov A,SBUF
    mov C,RB9      ;принятый признак
    xrl C,P        ;вычисленный признак
    ret

```

Прием методом опроса готовности (без ожидания)

SPIN:

```
    jnb RI, SP_RET
    clr RI
    mov A, SBUF
    mov C, RB9      ;принятый признак
    xrl C, P        ;вычисленный признак
```

SP_RET:

```
    ret
```

Прием методом прерываний

INT_RSI: ;вектор прерывания

```
    clr RI
    mov SBUF, A
    mov C, RB9      ;принятый признак
    xrl C, P        ;вычисленный признак
    reti
```

8.7 Передача символьной строки через последовательный порт

```

CR    equ 0Dh
LF    equ 0Ah

...
call STRING
db CR,LF          ;адрес в DPTR
db "Сообщение","$"
nop
...

STRING:
    pop DPH        ;Загрузка адреса сообщения
    pop DPL
STR1:
    clr A
    movc A,@A+DPTR ;Загрузка символа
STR2:
    jnb TI,$        ;Ожидание готовности
    clr TI
    mov SBUF,A      ;Выдача символа
    inc DPTR
    clr A
    movc A,@A+DPTR ;Загрузка символа
    cjne A,# "$",STR2 ;Цикл чтения

    mov A,#1
    jmp @A+DPTR     ;Возврат к основной программе

```

8.8 Измерение длительности сигнала

Входные данные:

IMP bit P3.0 ; входной цифровой сигнал

mov TMOD,#01010000b ;режим 1 для счетчика 1

mov TL1,#0

mov TH1,#0

WAIT2:

 jnb IMP, WAIT2 ;ожидание конца сигнала

WAIT0:

 jb IMP, WAIT0 ;ожидание начала сигнала

 setb TR1 ;TCON.6 - пуск счетчика 1

WAIT1:

 jnb IMP, WAIT1 ;ожидание конца сигнала

 clr TR1 ;останов счетчика 1

 mov A, TL1 ;считывание содержимого

 mov B, TH1 ;счетчика 1

8.9 Подсчет числа импульсов за заданный интервал времени

Входные данные:

T1-счетчик импульсов

T0-формирователь временного интервала

```

TIME    equ not (10000) + 1
;определение константы TIME для отсчета интервала 10 мс
        mov TMOD,#01010001b
; настройка T/C, 16 бит
; 1-счетчик импульсов
; 0-таймер (интервал времени)
        clr A
        mov TH1,A
        mov TL1,A

        mov TH0,#high(TIME)
        mov TL0,#low(TIME)

        orl TCON,#50h
;пуск счетчиков TR1 и TR0
        ;setb TR0
        ;setb TR1

WAIT:
        jbc TF0,EXIT
;TCON.5 - проверка переполнения таймера T/C0
        jmp WAIT          ;цикл, если TF0=0

EXIT:
        clr TR1           ;останов счетчика 1
        mov R1,TH1        ;(R0) (R1) - число импульсов
        mov R0,TL1

```

8.10 Измерение медленно изменяющегося сигнала

Входные данные:

P1 -вход данных ЦАП

INT0-выход компаратора

```

        mov R5,#8          ;счетчик
        clr A
        mov R3,A           ;значение шага ЦАП
        mov R4,A           ;измеряемое значение

        setb C

loop:
        mov A,R3           ;00
        rrc A              ;80 40 20
        mov R3,A           ;80 40 20
        orl A,R4           ;80 C0 A0
        mov P1,A
        jnb INT0,end_pr
        mov R4,A           ;80 - A0

end_pr:
        djnz R5,loop       ;время измрения 10 мкс * 8

```

8.11 Реализация клавиатуры

Алгоритм работы:

1. Вывод скан-кода
2. Прием кода нажатия
3. Если клавиша нажата, повторный прием
4. Если клавиша нажата, прием кода отжатия
5. Повторный прием кода отжатия
6. Перекодировка кода клавиши

PORT_STR equ P0

PORT_STLB equ P1

Сканирование клавиатуры

SCAN:

mov R4,#0 ; Сброс скан-кода

mov R7,#4 ; Счетчик сканирования

mov R6,11111110b ;Исходный байт сканирования(строка)

LOOP:

mov PORT_STR,R0 ; Вывод байта сканирования

rl A ; Сдвиг бита 0 влево

mov R6,A ; Сохранение текущего байта

mov A,PORT_STLB ; Ввод байта возврата (столбец)

; Обнаружение нажатой клавиши

; и перевод позиционного кода столбца

; в числовой код

mov R5,#3 ; Счетчик опросов

ROTATE:

rrc A ; Сдвиг байта возврата

jnc END_SCAN ; Выход при обнаружении 0

inc R4 ; Инкремент скан-кода

djnz R5,ROTATE

djnz R7,LOOP ; Счетчик циклов сканирования

jmp SCAN

END_SCAN:

СПИСОК ЛИТЕРАТУРЫ

1. Микушин А.В. Занимательно о микроконтроллерах. – СПб.: БХВ-Петербург, 2006. – 424с.: ил. – Библиогр.:с.418-419
2. Белов А.В. Конструирование устройств на микроконтроллерах. – СПб.: Наука и техника, 2005. – 255с.
3. Эрни Каспер. Программирование на языке Ассемблера для микроконтроллеров семейства i8051. – М.: Горячая линия-Телеком, 2003. – 191 с.
4. Схемотехника электронных систем. Микропроцессоры и микроконтроллеры / Гл.ред. Е.Кондукова. – СПб.: БХВ-Петербург, 2004
- 5 Григорьев В.Л. Программирование однокристальных микропроцессоров. – М.:Энергоатомиздат,1987.-228с.
- 6 Проектирование цифровых устройств на однокристальных микроконтроллерах /
В.В.Сташин, А.В.Урусов, О.Ф.Мологонцева.М.:Энергоатомиздат,1990.-224с.
- 7 Бобрыкин А.В., Липовецкий Г.П. и др. Однокристальные микроконтроллера. – М.: Бином, 1994. – 400с.
- 8 В.Я.Нерода, В.Э.Торбинский, Е.Л.Шлыков. Однокристальные микроконтроллера MCS-51. Архитектура. –М.: "Диджитал Компонентс", 1995. –164 с.
- 9 В. Б. Бродин, М.И.Шагурин Микроконтроллеры –М.: Издательство ЭКОМ, 1999. – 400 с.
10. <http://digital.sibsutis.ru/content.htm>

Список сокращений, символических имен и аббревиатур, примененных в таблице

A	- регистр-аккумулятор
addr	- прямой 8-битный адрес передачи управления
addr11	- прямой 11-битный адрес передачи управления
addr16	- прямой 16-битный адрес передачи управления
B	- регистр-расширитель аккумулятора
bit	- прямой 8-битный адрес бита
C	- признак переноса/заем
#data8	- 8- битный непосредственный операнд (константа)
#data16	- 16-битный непосредственный операнд (константа)
adr	- прямо адресуемый 8-битовый адрес ячейки внутренней памяти данных
sadr	- адрес ячейки приемника данных
dadr	- адрес ячейки источника данных
DPTR	- регистр-указатель данных
PC	- счетчик команд (СК)
Rn	- обобщенное имя рабочего регистра (R0 - R7)
SP	- регистр/указатель стека (РУС)
@Ri	- косвенно адресуемая ячейка памяти данных (0-256 байт)
@DPTR	- косвенно адресуемая ячейка памяти данных (0-64 кБайта)
[i-j]	- (i-j)-ые разряды операнда
<-	- занести
<-->	- поменять местами

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
Команды передачи данных				
MOV A, R0	E8	1	1	A ← R0
A, R1	E9	1	1	A ← R1
A, R2	EA	1	1	A ← R2
A, R3	EB	1	1	A ← R3
A, R4	EC	1	1	A ← R4
A, R5	ED	1	1	A ← R5
A, R6	EE	1	1	A ← R6
A, R7	EF	1	1	A ← R7
MOV A, adr	E5	2	1	A ← (adr)
MOV A, @R0	E6	1	1	A ← @R0
A, @R1	E7	1	1	A ← @R1
MOV A, #data8	74	2	1	A ← data8
MOV R0, A	F8	1	1	R0 ← A
R1, A	F9	1	1	R1 ← A
R2, A	FA	1	1	R2 ← A
R3, A	FB	1	1	R3 ← A
R4, A	FC	1	1	R4 ← A
R5, A	FD	1	1	R5 ← A
R6, A	FE	1	1	R6 ← A
R7, A	FF	1	1	R7 ← A
MOV R0, adr	A8	2	2	R0 ← (adr)
R1, adr	A9	2	2	R1 ← (adr)
R2, adr	AA	2	2	R2 ← (adr)
R3, adr	AB	2	2	R3 ← (adr)
R4, adr	AC	2	2	R4 ← (adr)
R5, adr	AD	2	2	R5 ← (adr)
R6, adr	AE	2	2	R6 ← (adr)
R7, adr	AF	2	2	R7 ← (adr)
MOV R0, #data8	78	2	1	R0 ← data8
R1, #data8	79	2	1	R1 ← data8
R2, #data8	7A	2	1	R2 ← data8

R3, #data8	7B	2	1	R3 <- data8
R4, #data8	7C	2	1	R4 <- data8
R5, #data8	7D	2	1	R5 <- data8
R6, #data8	7E	2	1	R6 <- data8
R7, #data8	7F	2	1	R7 <- data8
MOV adr, A	F5	2	1	(adr) <- A
MOV adr, R0	88	2	2	(adr) <- R0
adr, R1	89	2	2	(adr) <- R1
adr, R2	8A	2	2	(adr) <- R2
adr, R3	8B	2	2	(adr) <- R3
adr, R4	8C	2	2	(adr) <- R4
adr, R5	8D	2	2	(adr) <- R5
adr, R6	8E	2	2	(adr) <- R6
adr, R7	8F	2	2	(adr) <- R7
MOV adr, @R0	86	2	2	(adr) <- (R0)
adr, @R1	87	2	2	(adr) <- (R1)
MOV adr, #data8	84	2	2	(adr) <- #data8
MOV sadr, dadr	85	2	2	(sadr) <- (dadr)
MOV @R0, A	F6	1	1	(R0) <- A
@R1, A	F7	1	1	(R1) <- A
MOV @R0, adr	A6	2	1	(R0) <- (adr)
@R1, adr	A7	2	1	(R1) <- (adr)
MOV @R0, #data8	76	2	1	(R0) <- #data8
@R1, #data8	77	2	1	(R1) <- #data8
MOV DPTR, #data16	90	3	2	DPTR <- #data16
Команды передачи байта данных из памяти программ				
MOVC A, @A+DPTR	93	1	2	A <- (A+DPTR)
A, @A+PC	83	1	2	A <- (A+PC)
Команды передачи данных из внешней памяти данных				
MOVX A, @R0	E2	1	2	A <- (R0)
A, @R1	E3	1	2	A <- (R1)
MOVX A, @DPTR	E0	1	2	A <- (DPTR) расширенная ВПД
MOVX @R0, A	F2	1	2	(R0) <- A
@R1, A	F3	1	2	(R1) <- A
MOVX @DPTR, A	F0	1	2	(DPTR) <- A расширенная ВПД

Команды работы со стеком				
PUSH adr	CO	2	2	SP←-SP+1 (SP)←-(adr)
POP adr	BO	2	2	(adr)←-(SP) SP←-SP-1
Команды обмена данными				
XCH A, R0	C8	1	1	A <---> R0
A, R1	C9	1	1	A <-> R1
A, R2	CA	1	1	A <-> R2
A, R3	CB	1	1	A <-> R3
A, R4	CC	1	1	A <--> R4
A, R5	CD	1	1	A <--> R5
A, R6	CE	1	1	A <-> R6
A, R7	CF	1	1	A <-> R7
XCH A, adr	C5	2	1	A <--> (adr)
XCH A, @R0	C6	1	1	A <-> (R0)
A, @R1	C7	1	1	A <-> (R1)
XCHD A, @R0	D6	1	1	A[0...3] <-> (R0[0...3])
A, @R1	D7	1	1	A[0...3] <-> (R1[0...3])
Команды арифметических операций				
Сложение				
ADD A, R0	28	1	1	A←- A+R0
A, R1	29	1	1	A←- A+R1
A, R2	2A	1	1	A←- A+R2
A, R3	2B	1	1	A←- A+R3
A, R4	2C	1	1	A←- A+R4
A, R5	2D	1	1	A←- A+R5
A, R6	2E	1	1	A←- A+R6
A, R7	2F	1	1	A←- A+R7
ADD A, adr	25	2	1	A←- A+(adr)
ADD A, @R0	26	1	1	A←- A+(R0)
A, @R1	27	1	1	A←- A+(R1)
ADD A, #data8	24	2	1	A←- A+data8
ADDC A, R0	38	1	1	A←- A+R0+C
A, R1	39	1	1	A←- A+R1+C
A, R2	3A	1	1	A←- A+R2+C

A, R3	3B	1	1	A ← A+R3+C
A, R4	3C	1	1	A ← A+R4+C
A, R5	3D	1	1	A ← A+R5+C
A, R6	3E	1	1	A ← A+R6+C
A, R7	3F	1	1	A ← A+R7+C
ADDC A, adr	35	2	2	A ← A+(adr)+C
ADDC A, @R0	36	1	1	A ← A+(R0)+C
A, @R1	37	1	1	A ← A+(R1)+C
ADDC A, #data8	34	2	2	A ← A+data8+C
Десятичная коррекция				
DA A	E4	1	1	Если A[0-3]>9 или AC=1,
				то A ← A+06h, и далее
				если A[4-7]>9 или C= 1,
				то A ← A+60h
Вычитание				
SUBB A, R0	98	1	1	A ← A-R0-C
A, R1	99	1	1	A ← A-R1-C
A, R2	9A	1	1	A ← A-R2-C
A, R3	9B	1	1	A ← A-R3-C
A, R4	9C	1	1	A ← A-R4-C
A, R5	9D	1	1	A ← A-R5-C
A, R6	9E	1	1	A ← A-R6-C
A, R7	9F	1	1	A ← A-R7-C
SUBB A, adr	95	2	1	A ← A-(adr)-C
SUBB A, @R0	96	1	1	A ← A-(R0)-C
A, @R1	97	1	1	A ← A-(R1)-C
SUBB A, #data8	94	2	1	A ← A- #data8-C
Инкремент				
INC A	04	1	1	A ← A+1
INC R0	08	1	1	R0 ← R0+1
R1	09	1	1	R1 ← R1+1
R2	0A	1	1	R2 ← R2+1
R3	0B	1	1	R3 ← R3+1
R4	0C	1	1	R4 ← R4+1
R5	0D	1	1	R5 ← R5+1

R6	0E	1	1	R6 <- R6+1
R7	0F	1	1	R7 <- R7+1
INC adr	05	2	1	(adr) <- (adr)+1
INC @R0	06	1	1	(R0) <- (R0)+1
@R1	07	1	1	(R1) <- (R1)+1
INC DPTR	A3	1	2	DPTR <- DPTR+1
Декремент				
DEC A	14	1	1	A <- A-1
DEC R0	18	1	1	R0 <- R0-1
R1	19	1	1	R1 <- R1-1
R2	1A	1	1	R2 <- R2-1
R3	1B	1	1	R3 <- R3-1
R4	1C	1	1	R4 <- R4-1
R5	1D	1	1	R5 <- R5-1
R6	1E	1	1	R6 <- R6-1
R7	1F	1	1	R7 <- R7-1
DEC adr	15	2	1	(adr) <- (adr)-1
DEC @R0	16	2	1	(R0) <- (R0)-1
@R1	17	2	1	(R1) <- (R1)-1
Умножение				
MUL AB	A4	1	4	A <- мл. байт A*B B <- ст. байт A*B
Деление				
DIV AB	84	1	4	A <- A:B A - частное B - остаток
Команды логических операций				
Операция "И"				
ANL A,R0	58	1	1	A <- A & R0
A,R1	59	1	1	A <- A & R1
A,R2	5A	1	1	A <- A & R2
A,R3	5B	1	1	A <- A & R3
A,R4	5C	1	1	A <- A & R4
A,R5	5D	1	1	A <- A & R5
A,R6	5E	1	1	A <- A & R6

A, R7	5F	1	1	A <-A & R7
ANL A, adr	55	2	1	A <-A & (adr)
ANL A, @R0	56	1	1	A <-A & (R0)
A, @R1	57	1	1	A <-A & (R1)
ANL A, #data8	54	2	1	A <-A & data8
ANL adr, A	52	2	1	(adr) <- (adr) & A
ANL adr, #data8	53	3	2	(adr) <- (drc) & #data8
Операция "ИЛИ"				
ORL A, R0	48	1	1	A <-A or R0
A, R1	49	1	1	A <-A or R1
A, R2	4A	1	1	A <-A or R2
A, R3	4B	1	1	A <-A or R3
A, R4	4C	1	1	A <-A or R4
A, R5	4D	1	1	A <-A or R5
A, R6	4E	1	1	A <-A or R6
A, R7	4F	1	1	A <-A or R7
ORL A, adr	45	2	1	A <-A or (adr)
ORL A, @R0	46	1	1	A <-A or (R0)
A, @R1	47	1	1	A <-A or (R1)
ORL A, #data8	44	2	1	A <-A or data8
ORL adr, A	42	2	1	(adr) <- (adr) or A
ORL adr, #data8	43	2	2	(adr) <- (drc) or #data8
Операция "Исключающее ИЛИ"				
XRL A, R0	68	1	1	A <- A xor R0
A, R1	69	1	1	A <- A xor R1
A, R2	6A	1	1	A <- A xor R2
A, R3	6B	1	1	A <- A xor R3
A, R4	6C	1	1	A <- A xor R4
A, R5	6D	1	1	A <- A xor R5
A, R6	6E	1	1	A <- A xor R6
A, R7	6F	1	1	A <- A xor R7
XRL A, adr	65	2	1	A <- A xor (adr)
XRL A, @R0	66	1	1	A <- A xor (R0)
A, @R1	67	1	1	A <- A xor (R1)
XRL A, #data8	64	2	1	A <-A xor data8

XRL adr, A	62	2	1	(adr) <- (adr) xor A
XRL adr, #data8	63	3	2	(adr) <- (drc) xor #data8
Команды логических операций. Команды сдвигов				
RL A	23	1	1	A[n+1] <- A[n], A[0] <- A[7] и C <- A[7],
RLC A	33	1	1	A[n+1] <- A[n], A[0] <- C, C <- A[7]
RR A	03	1	1	A[n] <- A[n+1], A[7] <- A[0] и C <- A[0]
RRC A	13	1	1	A[n] <- A[n+1], A[7] <- C, C <- A[0]
CLR A	E4	1	1	A <- 0
CPL A	F4	1	1	A <- $\overline{A}$
SWAP A	C4	1	1	A[0...3] <- A[4...7] A[4...7] <- A[0...3]
Команды операций с битами				
CLR C	C3	1	1	C <- 0
CLR bit	C2	2	1	bit <- 0
SETB C	B3	1	1	C <- 1
SETB bit	B2	2	1	bit <- 1
CPL C	B3	1	1	C <- $\overline{C}$
CPL bit	B2	2	1	bit <- bit
ANL C, bit	82	2	2	C <- C & bit
ANL C, $\overline{\text{bit}}$	B0	2	2	C <- C & $\overline{\text{bit}}$
ORL C, $\overline{\text{bit}}$	72	2	2	C <- C or $\overline{\text{bit}}$
ORL C, bit	A0	2	2	C <- C or bit
MOV C, bit	A2	2	1	C <- bit
MOV bit, C	97	2	1	bit <- C
Холостая команда				
NOP	00	1	1	PC <- PC+1
Команды передачи управления. Безусловный переход				
LJMP addr16	02	3	2	PC <- addr16
AJMP addr11	01	2	2	
	21	2	2	

двоичный код	41	2	2	
команды	61	2	2	
	81	2	2	
a10a9a8 00001	A1	2	2	
a7.....a0	C1	2	2	
Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
	E1	2	2	
SJMP addr	80	2	2	PC ← PC+addr
JMP @A+DPTR	73	1	2	PC ← (A+DPTR)
Условный переход				
JZ addr	60	2	2	PC ← PC+2; если A=0, то PC ← PC+addr
JNZ addr	70	2	2	PC ← PC+2; если A<>0, то PC ← PC+addr
JC addr	40	2	2	PC ← PC+2; если C=1, то PC ← PC+addr
JNC addr	50	2	2	PC ← PC+2; если C=0, то PC ← PC+addr
JB bit, addr	20	3	2	PC ← PC+3; если bit=1, то PC ← PC+addr
JNB bit, addr	30	3	2	PC ← PC+3; если bit=0, то PC ← PC+addr
JBC bit, addr	10	3	2	PC ← PC+3; если bit=1, то bit ← 0 PC ← PC+addr
DJNZ R0, addr	D8	2	2	PC ← PC+2; Rn ← Rn-1, и если Rn≠0, то PC ← PC+addr
R1, addr	D9	2	2	
R2, addr	DA	2	2	
R3, addr	DB	2	2	
R4, addr	DC	2	2	
R5, addr	DD	2	2	
R6, addr	DE	2	2	
R7, addr	DF	2	2	
DJNZ adr, addr	D5	2	2	PC ← PC+2;

				(adr) <- (adr)-1, и если (adr)≠0, то PC <- PC+addr
Команды передачи управления				
Условный переход				
CJNE A, #data8, addr	B4	3	2	PC <- PC+3; а если #data8 ≠ A, то PC <- PC+addr
CJNE A, adr ,addr	B5	3	2	PC <- PC+3; а если (adr) ≠ A, то PC <- PC+addr
CJNE R0, #data8 ,addr	B8	3	2	PC <- PC+3;
R1, #data8, addr	B9	3	2	а если Rn ≠A,
R2, #data8, addr	BA	3	2	то PC <- PC+addr
R3, #data8, addr	BB	3	2	
R4, #data8, addr	BC	3	2	
R5, #data8, addr	BD	3	2	
R6, #data8, addr	BE	3	2	
R7, #data8, addr	BF	3	2	
CJNE @R0, #data8, addr	B6	3	2	PC <- PC+3; а если @Ri ≠data8,
@R1, #data8, addr	B7	3	2	то PC <- PC+addr
Команды вызова подпрограмм				
LCALL addr16	12	3	2	PC <- PC+3,
				(SP) <- PC[0...7],
				(SP+1) <- PC [8... 15],
				SP <- SP+2, PC <- addr16,
ACALL addr11	11	2	2	PC <- PC+2,
	31	2	2	(SP) <- PC[0...7],
двоичный код	51	2	2	(SP+1) <- PC[8. ..15],
команды	71	2	2	SP <- SP+2,
	91	2	2	PC[0. ..10] <- addr11
a10a9a8 10001	B1	2	2	
a7..... a0	DI	2	2	
	FI	2	2	

Команды возврата из подпрограмм				
RET	22	1	2	PC[8...15] <- (SP)
				PC[0..7] <- (SP-1)
				SP <- SP-2
RETI	32	1	2	PC[8...15] <- (SP)
				PC[0..7] <- (SP-1)
				SP <- SP-2