

Содержание

| | |
|--|----|
| Введение | 5 |
| Глава 1. Архитектура микроконтроллеров смешанного сигнала C8051Fxxx | 11 |
| 1.1. Архитектур а процессорного ядра CIP-51 | 11 |
| 1.1.1. Структура процессора и логическая организация МК | 11 |
| 1.1.2. Регистры специальных функций | 15 |
| 1.1.3. Слово состояния программы | 16 |
| 1.1.4. Система команд процессора | 17 |
| 1.2. Организация системы памяти | 23 |
| 1.2.1. Память программ | 23 |
| 1.2.2. Внутренняя память данных | 26 |
| 1.2.3. Внешняя память данных | 27 |
| 1.2.4. Интерфейс внешней памяти | 27 |
| 1.3. Организация ввода/вывода | 31 |
| 1.3.1. Схема выходного буфера порта ввода/вывода | 31 |
| 1.3.2. Структура ввода/вывода | 32 |
| 1.4. Система синхронизации | 36 |
| 1.4.1. Внутренний генератор синхронизации | 37 |
| 1.4.2. Внешний генератор синхронизации | 39 |
| 1.5. Система сброса | 40 |
| 1.5.1. Организация системы сброса | 41 |
| 1.5.2. Управление сторожевым таймером | 44 |
| 1.6. Система прерываний | 45 |
| 1.6.1. Логика прерываний | 46 |
| 1.6.2. Источники и приоритет прерываний | 48 |
| 1.7. Режимы уменьшенного потребления энергии | 50 |
| 1.7.1. Режим IDLE | 51 |
| 1.7.2. Режим STOP | 51 |
| 1.8. Таймеры | 52 |

| | |
|---|------------|
| 1.8.1. Таймеры и режимы их работы | 52 |
| 1.8.2. Программируемый массив счетчиков | 68 |
| 1.9. Контроллеры последовательной связи | 76 |
| 1.9.1. Контроллер SMBus | 76 |
| 1.9.2. Контроллер SPI | 84 |
| 1.9.3. Контроллер UART | 89 |
| 1.10. Аналоговые периферийные устройства | 97 |
| 1.10.1. Аналого-цифровые преобразователи | 97 |
| 1.10.2. Цифро-аналоговые преобразователи | 105 |
| 1.10.3. Схема формирования опорного напряжения | 106 |
| 1.10.4. Аналоговые компараторы | 108 |
| Глава 2. Средства разработки микроконтроллерных систем на базе МК C8051Fxxx | 111 |
| 2.1. Процесс разработки аппаратных и программных средств систем на базе МК | 112 |
| 2.2. Наборы средств проектирования | 115 |
| 2.3. Интегрированная среда разработки | 116 |
| 2.4. Мастер конфигурации МК | 121 |
| 2.5. Программное обеспечение фирмы Keil | 127 |
| 2.5.1. Ассемблер A51 | 127 |
| 2.5.2. Компилятор C51 | 130 |
| 2.5.3. Компоновщик BL51 | 133 |
| 2.5.4. Библиотекарь LIB51 | 134 |
| Глава 3. Проектирование аппаратных и программных средств | 135 |
| 3.1. Особенности системы команд и приемы программирования | 136 |
| 3.2. Программирование таймеров | 166 |
| 3.3. Обработка прерываний | 173 |
| 3.4. Программирование контроллеров последовательной связи | 181 |
| 3.5. Управление мощными нагрузками | 200 |
| 3.6. Подключение светодиодных и жидкокристаллических дисплеев | 213 |
| 3.7. Подключение клавиатур | 223 |
| 3.8. Использование аналоговых периферийных устройств | 246 |
| 3.9. Цифровая обработка сигналов | 259 |
| 3.10. Программирование энергонезависимой памяти | 265 |
| 3.11. Проектирование алгоритмов управления на основе конечно-автоматных моделей | 269 |
| Глава 4. Примеры построения измерительных и управляющих систем на базе МК SiLabs | 272 |
| 4.1. Система управления светофорами на пешеходном переходе | 272 |
| 4.2. Система управления электрическим бойлером | 279 |
| 4.3. Система управления автоматическими дверями | 288 |
| 4.4. Цифровой вольтметр | 294 |
| Заключение | 316 |
| Приложение 1. Включаемый файл ассемблера для МК 8051F020 | 317 |
| Приложение 2. Микроконтроллеры смешанного сигнала фирмы Silicon Laboratories | 322 |
| Литература | 328 |

Введение

Большинство создаваемых в наше время технических устройств, начиная от сложнейших объектов космической техники и кончая детскими игрушками, содержит одну или более так называемых встроенных систем (embedded system) автоматического управления.

Центральной частью такой системы является электронное управляющее устройство — контроллер (от англ. control — управление), снабженное соответствующими входными и выходными каналами, по которым передаются как аналоговые, так и дискретные (цифровые) сигналы.

С момента выпуска фирмой Intel первого микропроцессора 4004 в 1971 г. стало совершенно очевидным, что ядром контроллера должен быть универсальный цифровой процессор. Успехи микроэлектроники в последующие годы позволили разместить на одном кристалле с процессором устройства памяти и порты ввода/вывода — так появились однокристалльные микро-ЭВМ. А развитие теории цифровой обработки сигнала привело к появлению цифровых сигнальных процессоров (DSP — Digital Signal Processor). Особенностью однокристалльных DSP является наличие аналого-цифрового (АЦП) и цифро-аналогового (ЦАП) преобразователей, а также аппаратного умножителя-аккумулятора в составе арифметико-логического устройства. Наличие последнего позволяет резко сократить время выполнения операции умножения с накоплением суммы произведений, являющейся одной из основных операций алгоритмов цифровой обработки сигнала.

И наконец, желание разработчиков оптимизировать массовое производство встроенных систем сформировало концепцию технологии проектирования «Система на кристалле» (System on a Chip) и привело к появлению микроконтроллеров (МК), т. е. контроллеров встроенных систем, выполненных на одном кристалле. Согласно указанному выше такой МК должен с одинаковым успехом обрабатывать как цифровые, так и аналоговые сигналы, т. е. представлять собой определенный гибрид микро-ЭВМ и DSP. Иначе их называют микроконтроллерами смешанного сигнала (Mixed Signal Microcontroller).

Типовая структура такого МК с учетом всех современных тенденций имеет вид, показанный на **Рис. В.1**. Центральным элементом этой структуры является универсальное цифровое процессорное ядро. Наличие встроенного аппаратного умножителя-аккумулятора является серьезным преимуществом такого процессора в случае выполнения алгоритмов цифровой обработки сигнала.

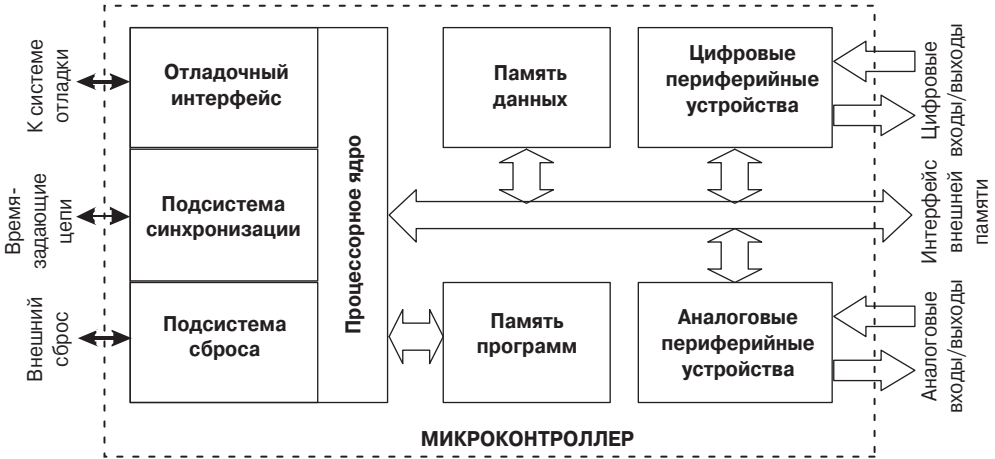


Рис. В.1. Типовая структура микроконтроллера смешанного сигнала.

Поскольку МК встраивается в объект управления, он вынужден работать в условиях значительных помех и колебаний питающего напряжения, приводящих к сбоям и временным отказам. Для распознавания таких ситуаций и последующего перезапуска работы микроконтроллера процессорное ядро снабжается специальной подсистемой сброса, содержащей, кроме стандартного входа внешнего сигнала сброса, целый ряд блоков контроля вычислительного процесса, формирующих сигнал сброса при обнаружении нештатной ситуации. В качестве примеров таких узлов можно привести сторожевой таймер (WatchDog Timer), страхующий МК от «зависания», и детектор пропадания синхронизации (Missing Clock Detector).

Вторая подсистема, характерная для процессорного ядра МК — это подсистема синхронизации (см. **Рис. В.1**). Такая подсистема включает, как правило, два генератора тактового сигнала: «внутренний» низко стабильный генератор для применения в системах, где не требуется точность по времени, и «внешний», с возможностью подсоединения к МК времязадающих элементов. Если в качестве такого элемента использовать, например, кварцевый резонатор, то можно с помощью МК производить высокоточные временные измерения или высокоточный синтез сигналов.

И наконец, третья подсистема — отладочный интерфейс — придает МК очень важное свойство внутрисистемного программирования и отладки (ISP — In System Programming and Debugging) с помощью компьютера и соответствующего программного обеспечения (средств разработки).

Использование в сочетании с этим интерфейсом в качестве памяти программ (см. **Рис. В.1**) МК энергонезависимого запоминающего устройства с электрическим стиранием (так называемая флэш-память — flash) позволяет по завершении отладки и отсоединения отладочного интерфейса продолжить функционирование системы в автономном рабочем режиме.

Память данных (оперативное запоминающее устройство — ОЗУ) МК может иметь несколько иерархических уровней с разными способами доступа. Часть этой памяти представляет собой так называемые регистры специальных функций (Special Function Registers) и содержит особую информацию, определяющую в том числе и внутреннюю конфигурацию МК. В современных МК число таких регистров может достигать нескольких сот.

Для расширения функциональных возможностей МК может снабжаться интерфейсом внешней памяти, позволяющим дополнительно подключать микросхемы запоминающих устройств.

Кроме того, совершенно очевидно, что в состав МК смешанного сигнала должны входить цифровые и аналоговые периферийные устройства (см. **Рис. В.1**).

Как правило, цифровые устройства представлены таймерами-счетчиками (Timer-Counter), программируемыми массивами счетчиков (Programmable Counter Array) и контроллерами последовательной связи (Serial Communication Controller) различных стандартов (UART, SMBus, SPI, CAN, USB и т. п.). Таймеры-счетчики позволяют решать задачи измерения временных интервалов и задачи формирования временных интервалов (например, ШИМ — широтно-импульсная модуляция — широко применяется для управления электродвигателями постоянного тока). Контроллеры последовательной связи применяются для построения микросетей из датчиков или исполнительных устройств, а также для подключения встроенной системы к последовательному порту компьютера. Такие микросети позволяют не только уменьшить количество и длину необходимых соединений, но и решить так называемую «проблему выводов» самого кристалла МК.

Аналоговые периферийные устройства представлены, как правило, АЦП и ЦАП, а также аналоговыми компараторами.

Нередко МК смешанного сигнала снабжают дополнительными устройствами, как, например, встроенным источником опорного напряжения, необходимым для работы АЦП и ЦАП, встроенным датчиком температуры, решающим, в частности, задачу температурной компенсации при подключении измерительных термопар к МК, и т. п.

8-битные МК смешанного сигнала фирмы Silicon Laboratories занимают одно из ведущих мест на рынке электронных компонентов этого типа в силу двух основных преимуществ:

- быстродействующее (тактовая частота до 100 МГц) процессорное ядро CIP-51, полностью программно совместимое с микроконтроллером фирмы Intel 8051, система команд которого является по сути международным стандартом, под нее разработано множество средств проектирования программ и готовых программ;
- широкий набор прецизионных аналоговых периферийных узлов.

Полный перечень всех производимых в настоящее время фирмой Silicon Laboratories МК и их параметры приведены в Приложении 2. Все микроконтроллеры образуют ряд семейств, имеющих особые свойства. Внутри каждого семейства имеется ряд модификаций, отличающихся количественными параметрами (в первую очередь количеством выводов корпуса). Обозначение каждого микроконтроллера содержит строку символов “С8051F”, после которых следует трехзначный десятичный номер модификации МК.

Базовым является семейство С8051F00х/С8051F01х (здесь и далее символом “х” обозначена последняя цифра модификации МК). МК этого семейства в зависимости от модификации имеют максимальную производительность 20 или 25 MIPS (Million Instructions per Second – миллионов команд в секунду), флэш-память программ объемом 32 или 16 Кбайт, память данных (ОЗУ) объемом 256, 1280 или 2304 байта, 8, 16 или 32 линии цифрового ввода/вывода, контроллеры последовательного интерфейса UART, SMBus, SPI, четыре 16-битных таймера, 5-канальный программируемый массив счетчиков, внутренний генератор тактового сигнала со стабильностью $\pm 20\%$ и максимальной частотой 16 МГц, 10- или 12-битный 8- или 4-канальный АЦП с максимальной скоростью преобразования до 100 тысяч выборок в секунду, два 12-битных ЦАП, встроенный датчик температуры, источник опорного напряжения и 2 аналоговых компаратора с программируемым гистерезисом. Наиболее функционально полным представителем этого семейства является МК С8051F005.

Дальнейшим расширением базового семейства являются микроконтроллеры семейства С8051F02х, которые содержат память программ и данных увеличенного объема (64 Кбайта и 4352 байта соответственно) плюс интерфейс внешней памяти, до 64-х линий цифрового ввода/вывода, дополнительный второй UART, дополнительный пятый таймер и дополнительный второй 8-битный 8-канальный АЦП с максимальной скоростью преобразования до 500 тысяч выборок в секунду (см. Приложение 2). Наиболее функционально полным представителем этого семейства является МК С8051F020.

Семейство С8051F04х является следующим шагом развития МК — в нем дополнительно имеется контроллер последовательного интерфейса CAN2.0В, количество каналов программируемого массива счетчиков увеличено до 6, частота внутреннего генератора тактового сигнала повышена до 24.5 МГц, а его стабильность улучшена до $\pm 2\%$, количество каналов главного АЦП увеличено до 13, количество аналоговых компараторов — до трех, а усовершенствованный входной усилитель позволяет подключать к МК аналоговые сигналы амплитудой до ± 60 В (см. Приложение 2). Наиболее функционально полным представителем этого семейства является МК С8051F040.

Еще более усовершенствованным является семейство С8051F06х: два одноканальных АЦП имеют разрешение 16 бит и обеспечивают скорость преобразования до 1 миллиона выборок в секунду, имеется контроллер прямого доступа к памяти (ПДП), а скорость преобразования третьего 10-битного 8-канального АЦП составляет 200 тысяч выборок в секунду (см. Приложение 2). Наиболее функционально полным представителем этого семейства является МК С8051F060.

Другое направление совершенствования МК, ориентированное на задачи цифровой обработки сигнала, представлено семейством С8051F12х/С8051F13х. Для этих МК характерны большие объемы памяти программ и данных (до 128 Кбайт и 8448 байт), а также производительность процессорного ядра до 50 и 100 MIPS и наличие на кристалле аппаратного умножителя-аккумулятора 16 x 16 бит (см. Приложение 2). Наиболее функционально полным представителем этого семейства является МК С8051F120.

Направление миниатюризации и удешевления МК представлено семействами «малого» класса. Прежде всего это семейство С8051F2хх, представители которого по сравнению с базовыми модификациями МК имеют уменьшенный объем памяти программ (8 Кбайт), сокращенный набор контроллеров последовательного интерфейса и меньшее количество таймеров, однако число коммутируемых каналов АЦП увеличено до 32 (см. Приложение 2). Наиболее функционально полным представителем этого семейства является МК С8051F206.

Это же направление представлено и в сверхминиатюрном семействе С8051F30х, представители которого имеют память программ 8, 4 или 2 килобайта, память данных — только 256 байт и всего 8 линий ввода/вывода, однако эти МК используют на печатной плате установочную площадку размером всего 3 x 3 мм² (см. Приложение 2). Наиболее функционально полным представителем этого семейства является МК С8051F300.

Наиболее мощным среди малогабаритных МК следует признать семейство С8051F31х, представители которого имеют такой же набор контроллеров последовательного интерфейса и таймеров-счетчиков, что и базовое семейство С8051F00х, однако имеют более быстродействующий АЦП с увеличенным количеством каналов и более точный внутренний генератор тактового сигнала (см. Приложение 2). Наиболее функционально полным представителем этого семейства является МК С8051F310.

На базе последнего семейства построены МК с контроллером последовательного интерфейса USB 2.0: С8051F320/1 и С8051F326/7. Эти МК имеют еще более точный ($\pm 1.5\%$) внутренний генератор тактового сигнала и усовершенствования в системе питания (см. Приложение 2).

Семейство С8051F33х является дальнейшим развитием сверхминиатюрного семейства С8051F30х. МК этого семейства имеют ОЗУ увеличенного объема (768 байт), большее количество линий ввода/вывода (17), более широкий набор цифровых периферийных узлов, более точный АЦП. Кроме того, в этом семействе имеются дополнительные усовершенствования: ЦАП с токовым выходом и дополнительный низкочастотный внутренний генератор с частотой до 10 КГц для микромощных применений (см. Приложение 2). Наиболее функционально полным представителем этого семейства является МК С8051F330. Этот МК имеет и специальную версию в корпусе типа PDIP, удобном для макетирования, — С8051F330D.

Семейство С8051F34х является усовершенствованным семейством МК с интерфейсом USB 2.0 (С8051F32х). В этих МК увеличено быстродействие процессорного ядра, увеличен объем памяти программ и данных, а также имеется интер-

фейс внешней памяти (см. Приложение 2). Наиболее функционально полным представителем этого семейства является МК С8051F340.

Семейство С8051F35х по объему памяти и количеству выводов вполне можно отнести к «малогабаритному» классу МК, однако отличительными особенностями этого семейства являются высокая производительность (50 MIPS) и наличие 8-канального прецизионного (16 и даже 24 бита) сигма-дельта АЦП со скоростью преобразования 1 тысяча выборок в секунду с широкодиапазонным (1...128) программируемым усилителем на входе (см. Приложение 2). Наиболее функционально полным представителем этого семейства является МК С8051F350.

И наконец, последняя разработка фирмы — семейство С8051F41х, которое сочетает высокую производительность, значительный объем памяти и малые габариты. Кроме того, МК этого семейства содержат 12-битные АЦП (до 24 каналов, скорость преобразования — 200 тысяч выборок в секунду), ЦАП (с токовым выходом) плюс встроенный стабилизатор напряжения питания и часы реального времени с батарейной поддержкой (см. Приложение 2). Наиболее функционально полным представителем этого семейства является МК С8051F410.

Таким образом, фирма Silicon Laboratories предоставляет разработчику широкую гамму микроконтроллеров, среди которых всегда можно выбрать оптимальный для конкретных условий применения. Фирма также снабжает разработчика удобными средствами разработки и качественной технической документацией.

Цель настоящей книги — помочь читателю изучить архитектурное построение микроконтроллеров С8051Fxxx, ознакомиться со средствами проектирования и освоить основные приемы программирования микроконтроллеров для выполнения заданных функций. Справочные данные, приведенные в книге, являются далеко не полными и никоим образом не заменяют оригинальные справочные материалы фирмы Silicon Laboratories. Книга рассчитана на подготовленного читателя, знакомого с основами вычислительной техники.

При подготовке книги автором был использован его опыт по преподаванию учебного курса «Микроконтроллеры» в Рыбинской государственной авиационной технологической академии в 1991—2001 гг. (г. Рыбинск, Россия) и в Инженерном академическом колледже ОРТ Брауде в 2002—2007 гг. (г. Кармиэль, Израиль). Поэтому формат и последовательность изложения материала отличаются от принятого в фирменных технических документах, использованных при подготовке рукописи. При написании книги в качестве основной литературы была использована оригинальная документация, размещенная на сайте www.silabs.com, поэтому в списке литературы указаны только другие источники информации, ссылки на которые имеются в тексте. Книга публикуется с официального разрешения фирмы Silicon Laboratories, однако она не несет никакой ответственности за достоверность материалов.

В приложенном к настоящей книге компакт-диске содержатся программные средства разработки для микроконтроллеров С8051Fxxx, оригинальная техническая документация (справочные данные и заметки по применению) и исходные файлы примеров программирования, описанных в книге.

Архитектура микроконтроллеров смешанного сигнала C8051Fxxx

В качестве основной модификации МК для первоначального изучения целесообразно выбрать C8051F020, так как он представляет собой наиболее совершенную версию базового варианта. После подробного ознакомления с этим МК читатель сможет самостоятельно ознакомиться с теми усовершенствованиями, которые были сделаны в последующих разработках МК.

1.1. Архитектура процессорного ядра CIP-51

1.1.1. Структура процессора и логическая организация МК

Процессор CIP-51 имеет одноадресную (аккумуляторную) архитектуру. Структура процессора (**Рис. 1.1**) содержит операционное и управляющее устройство. Центральной частью операционного устройства является арифметико-логическое устройство — АЛУ с двумя входами: первого и второго операндов, поступающих через регистры временного хранения TMP1 и TMP2 (temporary — временный). Источником первого операнда и местом сохранения результата для арифметических и логических операций служит аккумулятор А. Признаки результата операций (флаги) сохраняются в регистре PSW (Program Status Word — Слово Состояния Программы). Команды умножения и деления используют дополнительно регистр В, который другими командами может быть использован как РОН — регистр общего назначения (GPR — General Purpose Register). Другие РОН, а также ячейки сверхоперативной, так называемой внутренней памяти размещены в статическом ОЗУ — SRAM (Static Random Access Memory), доступ к которой осуществляется через регистр адреса SRAM. В этой же памяти размещен и стек, верхушку которого всегда определяет указатель стека — SP (Stack Pointer). Все перечисленные регистры и ячейки SRAM имеют размер 8 бит.

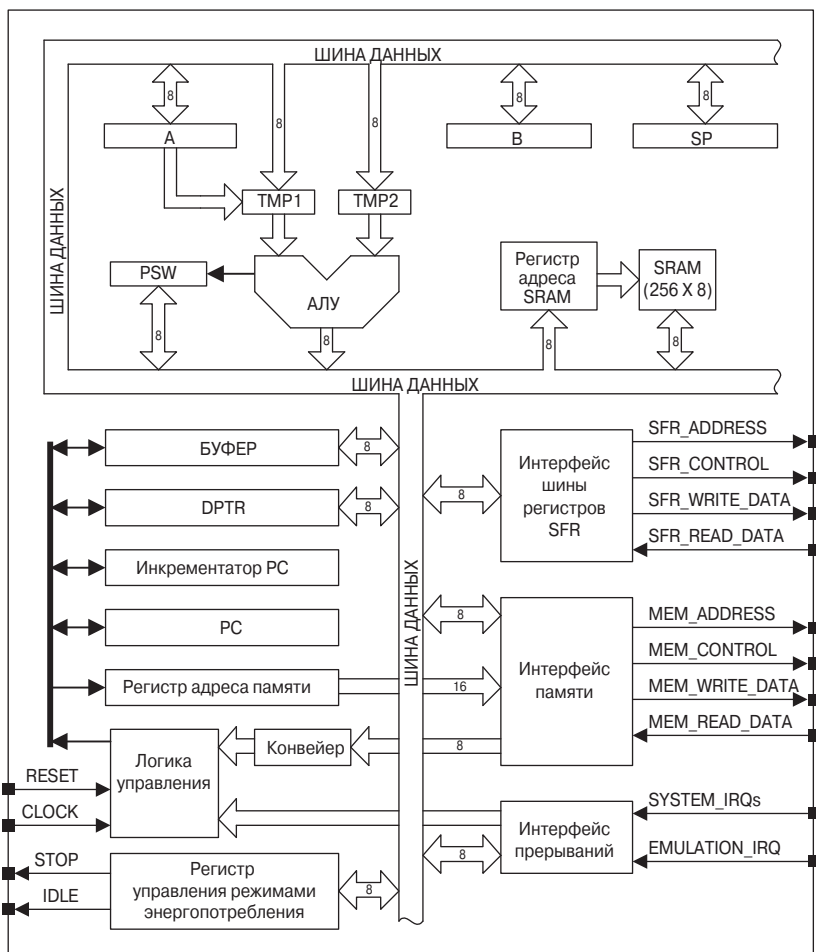


Рис. 1.1. Структура процессора CIP-51.

Центральной частью управляющего устройства процессора является 16-битный программный счетчик – PC (Program Counter), снабженный инкрементатором для поддержания естественного порядка следования команд. Второй 16-битный регистр DPTR (Data Pointer) является указателем данных, т. е. предназначен для хранения адреса данных, размещаемых в так называемой внешней памяти данных или в памяти программ МК. Третий 16-битный регистр Буфер (Buffer) недоступен для программиста и предназначен для временного хранения адресов переходов, содержащихся в командах переходов и вызовов подпрограмм. Все три указанных 16-битных регистра через регистр адреса памяти имеют выход на интерфейс памяти, к которому подключена память программ и внешняя память данных. Интерфейс памяти имеет все необходимые шины для осуществления операций чтения и записи: шину адреса MEM_ADDRESS, шину

управления MEM_CONTROL, выходную шину данных MEM_WRITE_DATA и входную шину данных MEM_READ_DATA. Информацией внешней памяти данных интерфейс обменивается с внутренней шиной данных процессора, а команды из памяти программ поступают через интерфейс на конвейер команд и далее на логику управления. Блок логики управления представляет собой конечный автомат, начальное состояние которого задается сигналом сброса RESET, а синхронизация осуществляется тактовыми импульсами CLOCK. Выходные сигналы этого блока управляют всеми элементами процессора, обеспечивая считывание и исполнение команд с использованием конвейера для повышения производительности.

Важнейшей частью МК являются также регистры специальных функций – SFR (Special Function Register), содержащие информацию, определяющую конфигурацию всех подсистем МК, и предназначенные для хранения данных периферийных узлов МК. Для доступа к этим регистрам используется специальный интерфейс шины регистров SFR, имеющий все необходимые шины для осуществления операций чтения и записи этих регистров: шину адреса SFR_ADDRESS, шину управления SFR_CONTROL, выходную шину данных SFR_WRITE_DATA и входную шину данных SFR_READ_DATA. Каждый регистр специальных функций имеет свой адрес в адресном пространстве SFR, там же размещаются и вышеупомянутые регистры процессора, к которым имеется доступ со стороны программы.

Как любой серьезный процессор, МК имеет интерфейс прерываний, обрабатывающий запросы прерываний – IRQ (Interrupt Request), генерируемые аппаратно (SYSTEM_IRQs) или программно (EMULATION_IRQ).

Кроме стандартного режима выполнения программы RUN МК может работать в режимах уменьшенного энергопотребления: холостого хода – IDLE и останова – STOP. Управление этими режимами осуществляет регистр управления режимами энергопотребления с помощью одноименных управляющих сигналов.

По сравнению с классическим МК Intel 8051, в котором все команды, за исключением команд умножения и деления, требовали для своего выполнения от 12 до 24 периодов синхронизации, а максимальная тактовая частота ограничивалась значением 12 МГц, CIP-51 выполняет 70% команд за 1 или 2 периода синхронизации, самая длинная команда требует 8 периодов, а максимальная тактовая частота составляет от 25 (для C8051F020) до 100 МГц (для некоторых модификаций МК). Это позволяет процессору CIP-51 достигать высокой производительности на относительно невысоких частотах синхронизации.

Логическая организация МК C8051F020 приведена на **Рис. 1.2**. Логически МК построен из четырех типовых блоков: процессора, памяти данных, отдельной памяти программ (так называемая Гарвардская архитектура) и блока ввода/вывода.

В составе процессора, кроме упомянутых выше аккумулятора ACC, регистра В, слова состояния программы PSW, указателя стека SP, указателя данных DPTR, логически разделенного на две 8-битные части: старшую DPH (high – высокий) и младшую DPL (low – низкий), и программного счетчика PC, входят четыре рабочих банка регистров: банк 0, банк 1, банк 2 и банк 3. Каждый из банков содержит по 8 регистров: R0...R7.

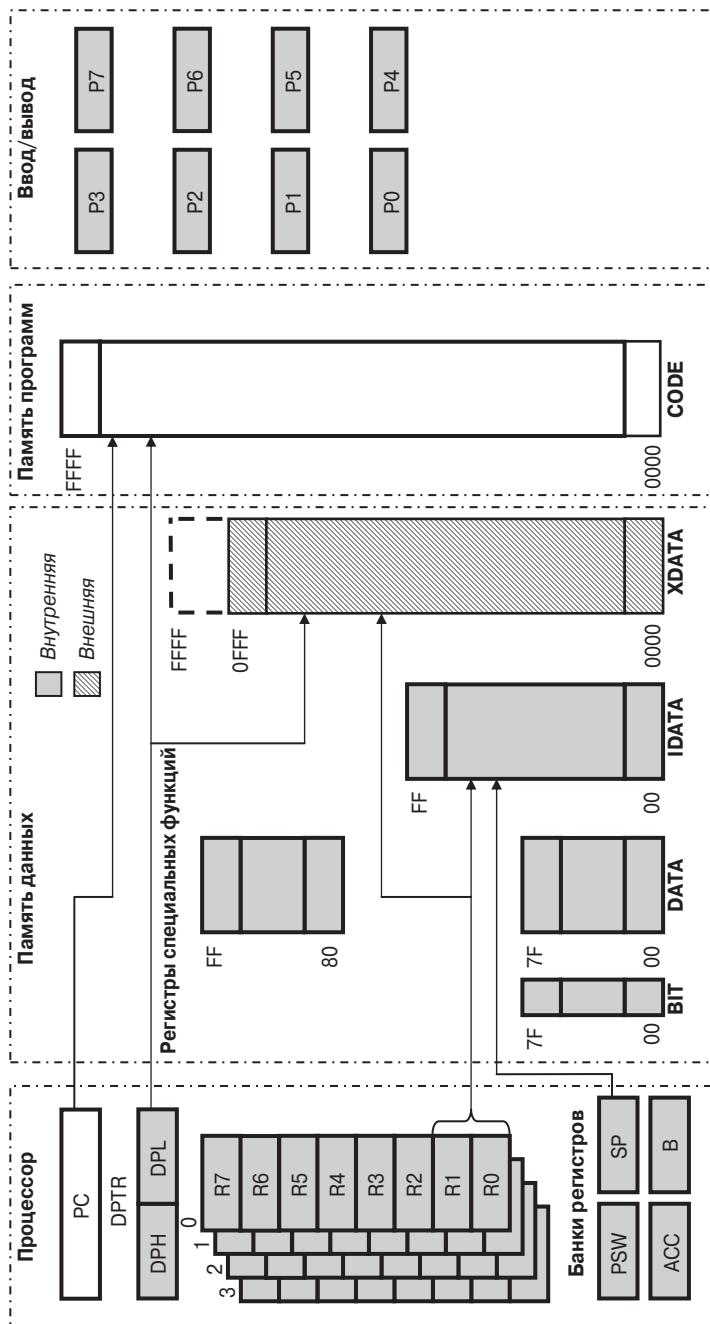


Рис. 1.2. Логическая организация МК.

Память данных разделена на четыре логических сегмента: BIT, DATA, IDATA и XDATA. Сегмент BIT, как это следует из его названия, предназначен для хранения булевых переменных. Его предельный объем — 128 бит, допустима только прямая адресация операндов с адресами от 00H до 7FH. Сегмент DATA имеет объем до 128 байт, допустима только прямая адресация операндов с адресами от 00H до 7FH, а старшие 128 адресов, начиная с 80H и кончая FFH, зарезервированы для обращения к регистрам специальных функций. Сегмент IDATA может иметь объем до 256 байт, однако здесь допустима только косвенная адресация (Indirect — косвенный). Из **Рис. 1.2** видно, что в качестве указателя могут выступать только регистры R0 и R1 активного банка регистров или указатель стека SP. Последнее свидетельствует о том, что именно в этом логическом сегменте размещен стек МК. Сегмент XDATA — сегмент внешней памяти (eXternal — внешний), его объем может составлять до 64 Кбайт, однако на кристалле МК C8051F020 имеется так называемое «внешнее» ОЗУ объемом только 4 Кбайта (конечный адрес 0FFFH). Через интерфейс внешней памяти можно подключить память данных объемом до 64 Кбайт. Из **Рис. 1.2** видно, что для этого сегмента допустима только косвенная адресация с полным адресом при использовании 16-битного указателя данных DPTR или косвенная страничная адресация при использовании как указателя одного из 8-битных регистров R0 или R1, при этом все адресное пространство будет содержать 256 страниц, из которых первые 8 могут находиться физически на кристалле МК, а остальные — снаружи.

Память программ образует логический сегмент с именем CODE, объемом до 64 Кбайт. Именно отсюда процессор считывает команды, используя в качестве указателя программный счетчик PC. Из **Рис. 1.2** видно, что есть возможность косвенной адресации к памяти программ с помощью указателя данных DPTR. В режиме чтения эта возможность используется для считывания таблиц постоянных данных, хранимых в памяти программ, а в режиме записи — для загрузки программных кодов в процессе отладки программ.

Блок ввода/вывода содержит восемь 8-битных портов P0...P7.

Следует отметить, что все логические элементы МК, выделенные серой заливкой, физически размещаются во внутренней памяти данных, объемом 256 байт, все выделенные регистры и порты ввода/вывода имеют прямые адреса в массиве адресов регистров специальных функций.

1.1.2. Регистры специальных функций

МК C8051F020 имеет 122 регистра специальных функций. Прямой адрес любого из этих регистров можно определить из адресной карты, показанной в **Табл. 1.1**. Для определения шестнадцатеричного адреса надо взять первую цифру номера строки (указан слева) и добавить цифру номера колонки, на пересечении которых размещен соответствующий регистр. Причем, если строка обозначена номером, оканчивающимся на 0, надо взять номер колонки без скобок, если — на 8, то надо взять символ в скобках. Например, адрес регистра SP будет 81H, а адрес регистра TL0 будет 8AH.

Таблица 1.1. Адресная карта регистров специальных функций

| | | | | | | | | |
|----|--------|---------|----------|----------|----------|----------|----------|---------|
| F8 | SPI0CN | PCA0H | PCA0CPH0 | PCA0CPH1 | PCA0CPH2 | PCA0CPH3 | PCA0CPH4 | WDTCN |
| F0 | B | SCON1 | SBUF1 | SADDR1 | TL4 | TH4 | EIP1 | EIP2 |
| E8 | ADC0CN | PCA0L | PCA0CPL0 | PCA0CPL1 | PCA0CPL2 | PCA0CPL3 | PCA0CPL4 | RSTSRC |
| E0 | ACC | XBR0 | XBR1 | XBR2 | RCAP4L | RCAP4H | EIE1 | EIE2 |
| D8 | PCA0CN | PCA0MD | PCA0CPM0 | PCA0CPM1 | PCA0CPM2 | PCA0CPM3 | PCA0CPM4 | |
| D0 | PSW | REF0CN | DAC0L | DAC0H | DAC0CN | DAC1L | DAC1H | DAC1CN |
| C8 | T2CON | T4CON | RCAP2L | RCAP2H | TL2 | TH2 | | SMB0CR |
| C0 | SMB0CN | SMB0STA | SMB0DAT | SMB0ADR | ADC0GTL | ADC0GTH | ADC0LTL | ADC0LTH |
| B8 | IP | SADEN0 | AMX0CF | AMX0SL | ADC0CF | P1MDIN | ADC0L | ADC0H |
| B0 | P3 | OSCXCN | OSCICN | | | P74OUT | FLSCL | FLACL |
| A8 | IE | SADDR0 | ADC1CN | ADC1CF | AMX1SL | P3IF | SADEN1 | EMI0CN |
| A0 | P2 | EMI0TC | | EMI0CF | P0MDOUT | P1MDOUT | P2MDOUT | P3MDOUT |
| 98 | SCON0 | SBUF0 | SPI0CFG | SPI0DAT | ADC1 | SPI0CKR | CPT0CN | CPT1CN |
| 90 | P1 | TMR3CN | TMR3RLL | TMR3RLH | TMR3L | TMR3H | P7 | |
| 88 | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | CKCON | PSCTL |
| 80 | P0 | SP | DPL | DPH | P4 | P5 | P6 | PCON |
| | 0(8) | 1(9) | 2(A) | 3(B) | 4(C) | 5(D) | 6(E) | 7(F) |

■ Бит-адресуемые регистры

Все регистры, адрес которых оканчивается на 80H или 88H, являются бит-адресуемыми, т. е. каждый двоичный разряд имеет битовый адрес, причем этот адрес лежит в диапазоне 80H...FFH, являющемся продолжением диапазона адресов сегмента ВIT (см. Рис. 1.2). Этот адрес строится по следующей схеме:

$$a_7a_6a_5a_4a_3b_2b_1b_0,$$

где $a_7a_6a_5a_4a_3$ – старшие пять бит адреса бит-адресуемого регистра;
 $b_2b_1b_0$ – номер бита (справа налево, начиная с 0).

Так, например, адрес 5-го бита аккумулятора ACC.5 будет E5H, а адрес 3-го бита регистра TCON.3 будет 8BH.

Расшифровку аббревиатур имен регистров специальных функций и их битов можно уточнить по тексту включаемого файла (язык ассемблера), приведенного в Приложении 1.

1.1.3. Слово состояния программы

Слово состояния программы сохраняется в регистре PSW и включает флаги переноса CY (carry), дополнительного переноса AC (auxiliary carry), переполнения OV (overflow), паритета P (parity), два флага пользователя F0 и F1, а также биты выбора банка регистров RS1 и RS0 (register bank select). Полная информация об этом регистре приведена на Рис. 1.3.

| | | | | | | | |
|------------------------|-------|-------|-------|-------|-------|----------------|-------|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R |
| CY | AC | F0 | RS1 | RS0 | OV | F1 | P |
| Значение после сброса: | | | | | | Адрес SFR: 00H | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |

Рис. 1.3. Регистр слова состояния программы (PSW).

Бит 7 CY — флаг переноса.

Устанавливается, если в последней арифметической операции имел место перенос (сложение) или заем (вычитание), и обнуляется в противном случае.

Бит 6 AC — флаг дополнительного переноса.

Устанавливается, если в последней арифметической операции имел место перенос в старший нибл (сложение) или заем из старшего нибла (вычитание), и обнуляется в противном случае.

Бит 5 F0 — флаг пользователя 0.

Бит-адресуемый флаг общего назначения для использования под управлением программы.

Биты 4, 3 RS1, RS0 — биты выбора банка регистров.

| RS1 | RS0 | Банк регистров |
|-----|-----|----------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 3 |

Бит 2 OV — флаг переполнения.

Устанавливается, если имело место переполнение при сложении или вычитании чисел со знаком, а также при умножении и делении, и обнуляется в противном случае.

Бит 1 F1 — флаг пользователя 1.

Бит-адресуемый флаг общего назначения для использования под управлением программы.

Бит 0 P — флаг паритета.

Устанавливается, если сумма восьми бит аккумулятора нечетна, и обнуляется, если четна.

1.1.4. Система команд процессора

Система команд процессора CIP-51 насчитывает 111 команд. Краткое описание системы команд приведено в Табл. 1.2. Для каждой команды указана мнемоника, описание операции, выполняемой командой, а также указано количество байтов для каждой команды и время выполнения в циклах частоты синхронизации. В конце таблицы расшифрованы условные обозначения регистров, операндов и методов адресации.

Таблица 1.2. Система команд

| Мнемоника | Описание | Количество байтов | Количество циклов |
|--------------------------------|--|-------------------|-------------------|
| АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ | | | |
| ADD A,Rn | Сложение регистра с А | 1 | 1 |
| ADD A,direct | Сложение прямо адресованного байта с А | 2 | 2 |
| ADD A,@Ri | Сложение косвенно адресованного байта с А | 1 | 2 |
| ADD A,#data | Сложение непосредственно байта с А | 2 | 2 |
| ADDC A,Rn | Сложение регистра с А и переносом | 1 | 1 |
| ADDC A,direct | Сложение прямо адресованного байта с А и переносом | 2 | 2 |
| ADDC A,@Ri | Сложение косвенно адресованного байта с А и переносом | 1 | 2 |
| ADDC A,#data | Сложение непосредственно байта с А и переносом | 2 | 2 |
| SUBB A,Rn | Вычитание регистра из А с заемом | 1 | 1 |
| SUBB A,direct | Вычитание прямо адресованного байта из А с заемом | 2 | 2 |
| SUBB A,@Ri | Вычитание косвенно адресованного байта из А с заемом | 1 | 2 |
| SUBB A,#data | Вычитание непосредственно байта из А с заемом | 2 | 2 |
| INC A | Инкремент А | 1 | 1 |
| INC Rn | Инкремент регистра | 1 | 1 |
| INC direct | Инкремент прямо адресованного байта | 2 | 2 |
| INC @Ri | Инкремент косвенно адресованного байта | 1 | 2 |
| DEC A | Декремент А | 1 | 1 |
| DEC Rn | Декремент регистра | 1 | 1 |
| DEC direct | Декремент прямо адресованного байта | 2 | 2 |
| DEC @Ri | Декремент косвенно адресованного байта | 1 | 2 |
| INC DPTR | Инкремент указателя данных | 1 | 1 |
| MUL AB | Умножение А на В | 1 | 4 |
| DIV AB | Деление А на В | 1 | 8 |
| DA A | Десятичная коррекция А | 1 | 1 |
| ЛОГИЧЕСКИЕ ОПЕРАЦИИ | | | |
| ANL A,Rn | Логическое И регистра с А | 1 | 1 |
| ANL A,direct | Логическое И прямо адресованного байта с А | 2 | 2 |
| ANL A,@Ri | Логическое И косвенно адресованного байта с А | 1 | 2 |
| ANL A,#data | Логическое И непосредственно байта с А | 2 | 2 |
| ANL direct,A | Логическое И А с прямо адресованным байтом | 2 | 2 |
| ANL direct,#data | Логическое И непосредственно байта с прямо адресованным байтом | 3 | 3 |
| ORL A,Rn | Логическое ИЛИ регистра с А | 1 | 1 |
| ORL A,direct | Логическое ИЛИ прямо адресованного байта с А | 2 | 2 |
| ORL A,@Ri | Логическое ИЛИ косвенно адресованного байта с А | 1 | 2 |

Таблица 1.2. Система команд

(продолжение)

| Мнемоника | Описание | Количество байтов | Количество циклов |
|-------------------------|---|-------------------|-------------------|
| ORL A,#data | Логическое ИЛИ непосредственно байта с А | 2 | 2 |
| ORL direct,A | Логическое ИЛИ А с прямо адресованным байтом | 2 | 2 |
| ORL direct,#data | Логическое ИЛИ непосредственно байта с прямо адресованным байтом | 3 | 3 |
| XRL A,Rn | Исключающее ИЛИ регистра с А | 1 | 1 |
| XRL A,direct | Исключающее ИЛИ прямо адресованного байта с А | 2 | 2 |
| XRL A,@Ri | Исключающее ИЛИ косвенно адресованного байта с А | 1 | 2 |
| XRL A,#data | Исключающее ИЛИ непосредственно байта с А | 2 | 2 |
| XRL direct,A | Исключающее ИЛИ А с прямо адресованным байтом | 2 | 2 |
| XRL direct,#data | Исключающее ИЛИ непосредственно байта с прямо адресованным байтом | 3 | 3 |
| CLR A | Очистка А | 1 | 1 |
| CPL A | Инвертирование А | 1 | 1 |
| RL A | Циклический сдвиг А влево | 1 | 1 |
| RLC A | Циклический сдвиг А влево через перенос | 1 | 1 |
| RR A | Циклический сдвиг А вправо | 1 | 1 |
| RRC A | Циклический сдвиг А вправо через перенос | 1 | 1 |
| SWAP A | Обмен ниблов в А | 1 | 1 |
| ПЕРЕСЫЛКА ДАННЫХ | | | |
| MOV A,Rn | Пересылка регистра в А | 1 | 1 |
| MOV A,direct | Пересылка прямо адресованного байта в А | 2 | 2 |
| MOV A,@Ri | Пересылка косвенно адресованного байта в А | 1 | 2 |
| MOV A,#data | Пересылка непосредственно байта в А | 2 | 2 |
| MOV Rn,A | Пересылка А в регистр | 1 | 1 |
| MOV Rn,direct | Пересылка прямо адресованного байта в регистр | 2 | 2 |
| MOV Rn,#data | Пересылка байта непосредственно в регистр | 2 | 2 |
| MOV direct,A | Пересылка А в прямо адресованный байт | 2 | 2 |
| MOV direct,Rn | Пересылка регистра в прямо адресованный байт | 2 | 2 |
| MOV direct,direct | Пересылка прямо адресованного байта в прямо адресованный байт | 3 | 3 |
| MOV direct,@Ri | Пересылка косвенно адресованного байта в прямо адресованный байт | 2 | 2 |
| MOV direct,#data | Пересылка непосредственно байта в прямо адресованный байт | 3 | 3 |
| MOV @Ri,A | Пересылка А в косвенно адресованный байт | 1 | 2 |
| MOV @Ri,direct | Пересылка прямо адресованного байта в косвенно адресованный байт | 2 | 2 |
| MOV @Ri,#data | Пересылка непосредственно байта в косвенно адресованный байт | 2 | 2 |
| MOV DPTR,#data16 | Загрузка указателя данных 16-битной константой | 3 | 3 |

Таблица 1.2. Система команд

(продолжение)

| Мнемоника | Описание | Количество байтов | Количество циклов |
|------------------------|--|-------------------|-------------------|
| MOVC A,@A+DPTR | Пересылка байта из памяти программ относительно DPTR в A | 1 | 3 |
| MOVC A,@A+PC | Пересылка байта из памяти программ относительно PC в A | 1 | 3 |
| MOVX A,@Ri | Пересылка байта из внешней памяти данных в A (8-битный адрес) | 1 | 3 |
| MOVX @Ri,A | Пересылка A во внешнюю память данных (8-битный адрес) | 1 | 3 |
| MOVX A,@DPTR | Пересылка байта из внешней памяти данных в A (16-битный адрес) | 1 | 3 |
| MOVX @DPTR,A | Пересылка A во внешнюю память данных (16-битный адрес) | 1 | 3 |
| PUSH direct | Загрузка прямо адресованного байта в стек | 2 | 2 |
| POP direct | Извлечение прямо адресованного байта из стека | 2 | 2 |
| XCH A,Rn | Обмен регистра с A | 1 | 1 |
| XCH A,direct | Обмен прямо адресованного байта с A | 2 | 2 |
| XCH A,@Ri | Обмен косвенно адресованного байта с A | 1 | 2 |
| XCHD A,@Ri | Обмен младшей тетрады косвенно адресованного байта с A | 1 | 2 |
| ОБРАБОТКА БИТОВ | | | |
| CLR C | Очистка переноса | 1 | 1 |
| CLR bit | Очистка прямо адресованного бита | 2 | 2 |
| SETB C | Установка переноса | 1 | 1 |
| SETB bit | Установка прямо адресованного бита | 2 | 2 |
| CPL C | Инвертирование переноса | 1 | 1 |
| CPL bit | Инвертирование прямо адресованного бита | 2 | 2 |
| ANL C,bit | Логическое И прямо адресованного бита и переноса | 2 | 2 |
| ANL C,/bit | Логическое И инверсии прямо адресованного бита и переноса | 2 | 2 |
| ORL C,bit | Логическое ИЛИ прямо адресованного бита и переноса | 2 | 2 |
| ORL C,/bit | Логическое ИЛИ инверсии прямо адресованного бита и переноса | 2 | 2 |
| MOV C,bit | Пересылка прямо адресованного бита в перенос | 2 | 2 |
| MOV bit,C | Пересылка переноса в прямо адресованный бит | 2 | 2 |
| JC rel | Переход, если перенос равен 1 | 2 | 2/3 |
| JNC rel | Переход, если перенос равен 0 | 2 | 2/3 |
| JB rel | Переход, если прямо адресованный бит равен 1 | 2 | 3/4 |
| JNB rel | Переход, если прямо адресованный бит равен 0 | 2 | 3/4 |
| JBC bit,rel | Переход, если прямо адресованный бит равен 1, и очистка бита | 3 | 3/4 |

Таблица 1.2. Система команд

(продолжение)

| Мнемоника | Описание | Количество байтов | Количество циклов |
|-----------------------|---|-------------------|-------------------|
| ПРОГРАММНЫЕ ВЕТВЛЕНИЯ | | | |
| ACALL addr11 | Абсолютный вызов подпрограммы | 2 | 3 |
| LCALL addr16 | Длинный вызов подпрограммы | 3 | 4 |
| RET | Возврат из подпрограммы | 1 | 5 |
| RETI | Возврат из подпрограммы обработки прерывания | 1 | 5 |
| AJMP addr11 | Абсолютный переход | 2 | 3 |
| LJMP addr16 | Длинный переход | 3 | 4 |
| SJMP rel | Короткий переход (относительный адрес) | 2 | 3 |
| JMP @A+DPTR | Косвенный переход относительно DPTR | 1 | 3 |
| JZ rel | Переход, если A равен нулю | 2 | 2/3 |
| JNZ rel | Переход, если A не равен нулю | 2 | 2/3 |
| CJNE A,direct,rel | Сравнение прямо адресованного байта и A и переход, если не равно | 3 | 3/4 |
| CJNE A,#data,rel | Сравнение непосредственно байта и A и переход, если не равно | 3 | 3/4 |
| CJNE Rn,#data,rel | Сравнение непосредственно байта и регистра и переход, если не равно | 3 | 3/4 |
| CJNE @Ri,#data,rel | Сравнение непосредственно байта и косвенно адресуемого байта и переход, если не равно | 3 | 4/5 |
| DJNZ Rn,rel | Декремент регистра и переход, если не равно нулю | 2 | 2/3 |
| DJNZ direct,rel | Декремент прямо адресованного байта и переход, если не равно нулю | 3 | 3/4 |
| NOP | Нет операции | 1 | 1 |

Примечание. Условные обозначения регистров, операндов и методов адресации:

Rn — регистры R0...R7 выбранного регистрового банка.

@Ri — ячейка внутренней памяти данных, косвенно адресуемая через регистры R0...R1.

rel — 8-битное знаковое смещение (в дополнительном коде) относительно первого байта следующей команды. Используется командой SJMP и всеми командами условных переходов.

direct — 8-битный адрес внутренней памяти данных. Это может быть адрес прямо адресуемой ячейки памяти (00H...7FH) или адрес регистра специальных функций (80H...FFH).

#data — 8-битная константа.

#data16 — 16-битная константа.

bit — прямо адресуемый бит внутренней памяти данных или регистра специальных функций.

addr11 — 11-битный адрес, используемый командами ACALL и AJUMP. Этот адрес должен быть внутри 2-Кбайтной страницы памяти программ и является первым байтом следующей команды.

addr16 — 16-битный адрес, используемый командами LCALL и LJUMP. Этот адрес может быть любым внутри 64-Кбайтного адресного пространства памяти программ.

Единственный неиспользованный код операции (0A5h) выполняет те же действия, что и команда NOP.

Все мнемоники © Intel Corporation 1980.

Система команд содержит пять групп команд: арифметические операции, логические операции, операции пересылки данных, операции обработки битов и программные ветвления.

В группе команд арифметических операций представлены операции сложения, сложения с переносом, вычитания с заемом (простого вычитания нет!), инкремента, декремента, умножения, деления и десятичной коррекции после сложения. Все эти операции производятся над байтовыми операндами. Операнды операций умножения и деления размещаются в регистрах А и В, в этой же паре регистров записывается и результат. После умножения старший байт двухбайтного произведения размещается в регистре В, а младший — в регистре А. После деления А на В частное размещается в регистре А, а остаток — в В. Кроме того, в этой группе команд есть одна 16-битная операция — инкремент указателя данных INC DPTR.

В группе команд логических операций представлены логические операции И, ИЛИ, исключающее ИЛИ, очистка и инвертирование аккумулятора, а также циклические сдвиги аккумулятора влево и вправо, как простые, так и сдвиги через перенос. В эту группу включена также команда свопинга аккумулятора SWAP A, позволяющая менять местами старший и младший nibble (nibble — полубайт) аккумулятора, что очень удобно для алгоритмов обработки двоично-десятичных кодов.

В группе команд пересылки данных имеются команды пересылки данных между ячейками внутренней памяти данных (MOV), между аккумулятором и ячейками внешней памяти данных (MOVX), пересылки из памяти программ в аккумулятор (MOVC). Имеется также команда MOV DPTR,#data16, позволяющая производить загрузку 16-битного указателя данных. Кроме того, в эту группу включены команды загрузки и извлечения стека и команды обмена операндами, среди которых одна из команд (XCHD A,@Ri) предназначена для обработки двоично-десятичных кодов и производит обмен только младших nibbles.

Группа команд обработки битов включает команды очистки, установки и инвертирования битов, логические операции И, ИЛИ, пересылки битов и команды условных переходов по состоянию битов. Особая команда JBC bit,rel позволяет выполнить одновременно условный переход и очистку бита, что очень удобно для обработки флагов. Следует отметить, что роль аккумулятора в двухоперандных битовых операциях играет бит переноса (в мнемонике команды он обозначается как C).

Группа команд программных ветвлений содержит команды вызова и возврата из подпрограмм (в том числе из подпрограмм обработки прерываний), команды безусловных и условных переходов, а также комплексные команды: сравнения и условного перехода, декремента и условного перехода — удобные для организации циклов.

И наконец, в составе команд имеется «холостая» команда NOP — нет операции.