Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра систем управления

А.П. Пашкевич, О.А. Чумаков

МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

Конспект лекций
для студентов специальности
I-53 01 07 «Информационные технологии и управление
в технических системах»
дневной формы обучения

В 2-х частях

Часть 2

Минск 2006

УДК 004.31(075.8)
ББК 32.973.26-04 я 73
П 22

Р е ц е н з е н т :
доц. кафедры ЭВМ БГУИР,
канд. техн. наук Н.И. Силков

Пашкевич А.П.

П 22 Микропроцессорные системы управления: Конспект лекций для студ. спец. I-53 01 07 «Информационные технологии и управление в технических системах» дневн. формы обуч.: В 2 ч. Ч. 2 / А.П. Пашкевич, О.А. Чумаков. – Мн.: БГУИР, 2006. – 72 с.: ил.

ISBN 985-444-986-6 (ч. 2)

Во второй части издания рассмотрены вопросы, связанные с принципами построения однокристальных микроконтроллеров семейств MCS-48 и MCS-51, а также особенностями их применения в цифровых системах управления. Описаны их функциональные возможности, электрические параметры и система команд. Приводятся технические характеристики современных однокристальных микроконтроллеров, базирующихся на архитектуре MCS-51.

Часть I издана БГУИР в 2005 г.

УДК 004.31 (075.8)
ББК 32.973.26-04 я 73

ISBN 985-444-986-6 (ч. 2)
ISBN 985-444-865-7

Содержание

Введение

Большинство современных цифровых систем управления реализуются на базе *микроконтроллеров* (МК) – специализированных микропроцессорных устройств, ориентированных на выполнение управляющих функций. Интегрируя на одном кристалле высокопроизводительный процессор, память и набор периферийных устройств, микроконтроллеры позволяют с минимальными затратами реализовать высокоэффективные системы и устройства управления.

В настоящее время однокристальные микроконтроллеры являются наиболее массовыми представителями микропроцессорной техники, объем выпуска которых составляет около 2,5 млрд. штук в год. Их производством занимается ряд фирм (Intel, Motorola, Philips, Siemens, Atmel, Dallas, Temic, Oki, AMD и др.), которые предоставляют потребителям большую номенклатуру изделий, начиная с относительно простых 8-разрядных моделей и кончая 32-разрядными устройствами, ядром которых служат CISC-и RISC-процессоры.

В изделиях массового применения наибольшее распространение получили 8-разрядные однокристальные микроконтроллеры, которые имеют практически одинаковое вычислительное ядро, но отличаются объемом и типом внутренней памяти, номенклатурой размещенных на кристалле периферийных устройств и рядом технических характеристик. У истоков производства микроконтроллеров стоит фирма Intel с семействами восьмиразрядных микроконтроллеров MCS-48 и MCS-51. Первые модификации MCS-48/51 были выполнены по n-MOS технологии и являлись функционально завершенными однокристальными *микроЭВМ гарвардской архитектуры*, один из основных принципов которой состоит в логическом разделении адресных пространств памяти программ и данных. Последующие версии контроллеров MCS-51 стали изготавливать по более совершенной и низкопотребляющей CMOS технологии.

На сегодняшний день существует более 200 модификаций микроконтроллеров семейства MCS-51, выпускаемых почти 20-ю компаниями. Эти модификации включают в себя кристаллы с широчайшим набором периферии: от простых 20-выводных МК с одним таймером и резидентной памятью программ объемом 1 Кбайт до сложнейших 100-выводных кристаллов с 10-разрядными АЦП, массивами таймеров/счетчиков, аппаратными 16-разрядными умножителями и резидентной памятью программ до 64 Кбайт. Каждый год появляются все новые варианты представителей семейства MCS-51 с повышенным быстродействием, пониженным энергопотреблением, большим объемом памяти, а также с возможностью подключения сложных периферийных устройств на основе CAN и USB интерфейсов.

В данном курсе лекций рассматривается архитектура типичных представителей семейств микроконтроллеров MCS-48 и MCS-51, а также приводятся рекомендации по их использованию в цифровых системах управления.

1. Основы архитектуры микроконтроллеров

1.1. Основные типы микроконтроллеров

Микроконтроллер – это устройство, выполненное по технологии БИС и объединяющее на одном кристалле микропроцессор, память, порты ввода-вывода, а также средства поддержки работы в реальном масштабе времени (таймеры, обработчики прерываний и т.д.).

В однокристальных микроконтроллерах используется *гарвардская архитектура*. Согласно ее концепциям память программ и память данных имеют раздельные адресные пространства, при обращении к которым используются различные механизмы адресации. Содержимое памяти программ формируется либо на заводе-изготовителе (масочное программирование), либо с использованием программатора пользователя (электрическое программирование).

Современные микроконтроллеры можно разделить на три основных типа: а) встраиваемые 8-разрядные микроконтроллеры; б) 16- и 32-разрядные микроконтроллеры; в) цифровые сигнальные процессоры.

Во *встраиваемых микроконтроллерах* все необходимые ресурсы (память, таймеры, порты ввода-вывода и т.д.) располагаются на одном кристалле с процессорным ядром, благодаря чему обеспечивается их включение в систему управления с использованием минимального количества дополнительных компонентов. Для них характерны небольшие по современным понятиям объемы памяти программ (ROM объемом от 1 до 128 Кбайт) и памяти данных (RAM объемом от 64 байт до 4 Кбайта. Кроме того, обычно исключается возможность записи в память программ, что уменьшает вероятность ошибок, и имеется защита от несанкционированного копирования содержимого этой памяти. Типичный микроконтроллер для встроенных приложений потребляет ток от микроампер до десятка миллиампер и обычно работает в температурном диапазоне от –55 до +125°С. Программируются такие контроллеры на языках ассемблер или С.

В более сложных *микроконтроллерах (16- и 32-разрядных)* используют только внешнюю память, которая включает в себя как память программ (ROM), так память данных (RAM). Они применяются в системах, где требуется большой объем памяти и относительное небольшое количество устройств (портов) ввода/вывода. Типичным примером применения такого МК с внешней памятью является котроллер жесткого диска (HDD) с буферной кэш-памятью, который обеспечивает промежуточное хранение и распределение больших объемов данных (порядка нескольких мегабайт). Внешняя память позволяет такому микроконтроллеру работать с более высокой скоростью, чем встраиваемый МК.

Цифровые сигнальные процессоры (DSP) предназначены для цифровой обработки аналоговых сигналов, т.е. для решения задач, которые традиционно решала аналоговая схемотехника. Они обеспечивают ввод данных от аналогового источника, их обработку и формирование соответствующего выходного сигнала в

реальном масштабе времени. К сигнальным процессорам предъявляются специфические требования. От них требуются максимальное быстродействие, малые габариты, легкая стыковка с аналого-цифровыми и цифро-аналоговыми преобразователями, большая разрядность обрабатываемых данных и небольшой набор математических операций, обязательно включающий операцию умножения-накопления и аппаратную организацию циклов. На процессорах такого типа сделано большинство модемов, а также различные системы обработки звука и видео.

Для подключения внешних устройств в микроконтроллерах применяются как *параллельные* порты ввода/вывода, так и *последовательные* шины типа I2C (Inter-Integrated Circuit bus) или SPI (serial peripheral interface). Благодаря этим шинам для подключения периферийных микросхем к процессору достаточно двух-трех линий. Соответственно, уменьшается число выводов микросхем, позволяя уменьшить габариты устройств, повысить надежность и упростить разработку устройств.

Следует также различать *CISC* и *RISC*-микроконтроллеры. Команды RISC-процессоров очень просты и почти всегда выполняются за один такт. Для реализации операции, обрабатываемой CISC-процессором как одна команда, RISC должен выполнить последовательность микрокоманд. Достоинством CISC-микроконтроллеров является сокращенный объем программного кода и экономия усилий программиста, затраченных на его написание. Однако RISC-процессоры производительнее, занимают меньше места на кристалле и потребляют значительно меньше энергии, чем CISC. Поэтому в последнее время появляется все больше типов микроконтроллеров с RISC-архитектурой.

1.2. Память микроконтроллеров и их программирование

В микроконтроллерах используется три основных вида памяти: память программ, память данных, регистры. *Память программ* представляет собой постоянную память, предназначенную для хранения программного кода и констант. Она не изменяет содержимого в процессе выполнения программы. *Память данных* предназначена для хранения переменных в ходе выполнения программы. *Регистры* включают внутренние регистры процессора и регистры, используемые для управления периферийными устройствами. Следует заметить, что в микроконтроллерах семейств MCS-48/51 в качестве регистров используется специальная область внутренней (резидентной) памяти данных. Кроме того, наряду с внутренней памятью большинство микроконтроллеров могут использовать и внешнюю память программ и данных.

Для хранения программ обычно служит один из видов постоянной памяти: ROM (масочные ПЗУ), PROM (однократно программируемые ПЗУ), EPROM (электрически программируемые ПЗУ с ультрафиолетовым стиранием) или EEPROM (ПЗУ с электрической записью и стиранием, к этому виду также относятся современные микросхемы Flash-памяти). Все эти виды памяти

являются энергонезависимыми, т.е. содержимое такой памяти сохраняется после выключения питания.

Для программирования микроконтроллеров (т.е. записи в них программ) обычно используются специальные устройства – программаторы, подключаемые к персональному компьютеру. Исключением является микроконтроллеры с ROM-памятью, программирование которых осуществляется в процессе изготовления. Однако в настоящее время программирование EEPROM-памяти можно осуществлять и непосредственно на плате проектируемого или уже готового устройства. Такой способ получил название – ISP (In System Programming), он позволяет обновлять программное обеспечение микроконтроллера без удаления из платы, что дает большую экономию времени на этапе разработки.

Последние модели микроконтроллеров содержат встроенную Flash-память, которая функционально мало отличается от EEPROM. Основное различие состоит в способности стирания записанной информации. В памяти EEPROM стирание производится отдельно для каждой ячейки, а во Flash-памяти стирание осуществляется целыми блоками.

Для хранения данных используется RAM – оперативное запоминающее устройство. Число циклов чтения и записи в такой памяти неограниченно, но при отключение питания вся информация теряется.

1.3. Питание и управление энергопотреблением

Первые микроконтроллеры были рассчитаны на напряжение питания 5В. Однако в последнее время появляется все больше устройств с номинальным напряжением 3,3В и менее. Это связано в первую очередь с расширяющейся сферой применения микроконтроллеров в устройствах с автономным питанием.

Кроме того, во многих микроконтроллерах предусмотрена возможность управления потребляемой мощностью, путем их перевода программными средствами в режим хода или режим останова. *Режим холостого хода* характеризуется тем, что работа процессорного ядра приостанавливается, в рабочем состоянии остаются лишь генератор синхросигналов и сторожевой таймер. Регистры сохраняют свое значение. Выход из этого режима возможен по прерыванию или по сигналу сброса. Также может быть предусмотрен специальный таймер холостого режима, который периодически “пробуждает” микроконтроллер для выполнения активных действий. В этом режиме микроконтроллер потребляет около 30% номинальной мощности. В *режиме останова* работа внутри кристалла полностью остановлена. Выход из режима возможен по сигналу сброса (иногда по сигналу внешнего прерывания). В режиме останова потребляемый ток может достигать 1 мкА и менее.

2. Семейства MCS-48, MCS-51 и их модификации

2.1. Микроконтроллеры фирмы Intel

Микроконтроллеры Intel 8048 и 8051, изготовленные по технологии n-MOS, заложили основу архитектуры однокристальных устройств семейств MCS-48 и MCS-51. Следующим принципиальным шагом стал переход на CMOS-технологию (модификация 8xC51), что позволило реализовать режимы Idl (холостой ход) и PowerDown (пониженное энергопотребление). В дальнейшем фирмой Intel были выпущены микроконтроллеры 8xC51FA/FB/FC и 8xC51RA/RB/RC со встроенными специализированными таймером/счетчиком (PCA) и сторожевым таймером (WDT), позволяющими существенно снизить время реакции на внешние события. Архитектура этих микроконтроллеров оказалась настолько удачной, что стала промышленным стандартом.

Изначально наиболее “узкими” местами MCS-51 были 8-разрядное АЛУ и довольно большое время выполнения команд (12 тактов). Для устранения этих недостатков было создано семейство микроконтроллеров MCS-251/151, обеспечивающих адресацию до 16 Мбайт памяти и выполняющих 16-битные арифметические и логические операции за 2 такта. Система команд MCS-251 включает два набора инструкций – первый набор является копией системы команд MCS-51, а второй – содержит расширенные инструкции, реализующие преимущества архитектуры MCS-251. Перед использованием микроконтроллера его необходимо сконфигурировать, т.е. определить, какой из наборов станет активным после включения питания. Для пользователей, ориентированных на применение MCS-251 в качестве замены MCS-51, выпускаются микроконтроллеры MCS-151, уже запрограммированные на активизацию системы команд MCS-51.

В системах управления также широко используются 16-разрядные микроконтроллеры семейства MCS-96, основанные на несколько иной архитектуре. Базовый представитель этого семейства 8096 содержит 16-разрядный процессор, 8 Кбайт внутренней программной памяти с возможностью внешнего расширения до 64 Кбайт, 232 байта ОЗУ с возможностью внешнего расширения до 64 Кбайт, развитые средства поддержки режима реального времени, скоростной ввод-вывод. Он позволяет осуществлять операции в форматах “байт” (8 бит), “слово” (16 бит), а а также “двойное слово” (32 бит). Такие микроконтроллеры обладают усовершенствованной системой команд, а также разнообразием встроенной периферии (параллельные и последовательные порты, блоки процессоров событий, многоканальные аналого-цифровые преобразователи, широтно-импульсные модуляторы, трехфазные генераторы, генераторы меандра, сторожевые таймеры и пр.). Типичные области применения для этих микроконтроллеров – управление двигателями, модемы, контроллеры жестких и оптических дисков.

2.2. Микроконтроллеры фирмы Philips

Фирма Philips производит более 100 модификаций семейства MCS-51, в состав которых входят микроконтроллеры в корпусах от 24 до 80 выводов, работающие при тактовой частоте до 40 МГц и напряжении питания от 1,8 В. Изделия этой фирмы ориентированы на рынок бытовой электроники. В таких микроконтроллерах используется стандартное ядро MCS-51, дополненное широчайшим набором периферии, среди которой следует отметить: 10-разрядные АЦП; 8-разрядный ЦАП; широтно-импульсные модуляторы; массивы программируемых таймеров/счетчиков; интерфейсы I2C, CAN; специализированную периферию для телевизионной, видео- и аудиотехники. Кроме того, Philips выпускает микроконтроллеры MCS-51, обладающие функцией снижения электромагнитных помех (Lower EMI).

Новое оригинальное решение Philips по развитию архитектуры семейства MCS-51 реализовано в семействе микроконтроллеров 51XA (“расширенная архитектура 51”). При этом разработчики решили отказаться от достижения совместимости кодов команд нового микроконтроллера с 8051. Такое решение, с одной стороны, сделало невозможным прямую замену микроконтроллера MCS-51 на микроконтроллер семейства 51XA, но с другой стороны позволило создать полноценное 16-разрядное ядро, обеспечивающее почти 100-кратное увеличение быстродействия по сравнению с традиционной архитектурой MCS-51.

2.3. Микроконтроллеры фирмы Atmel

Фирма Atmel является одним из мировых лидеров по производству однокристальных микроконтроллеров. Архитектура MCS-51 реализована в ее контроллерах семейства AT89, которые имеют встроенную Flash-память для хранения программ и данных, а также обладают возможностью внутрисхемного программирования (загрузки программного кода) по последовательному интерфейсу. Имеются также модификации микроконтроллеров этого семейства, конструктивно исполненных в 20-выводных корпусах (вместо стандартных 40-выводных). Кроме того, выпускаются микроконтроллеры с архитектурой MCS-51, которые программируются по заказу покупателя на предприятии-изготовителе (семейство AT80) или допускают лишь однократное программирование пользователем (семейство AT87).

В настоящее время более широкое распространение получают новые изделия фирмы Atmel – 8-разрядные микроконтроллеры с AVR-ядром и 32-разрядные контроллеры с ARM-ядром. Они построены на базе RISC–процессоров и имеют более высокую производительность, чем MCS-51 при меньшем энергопотреблении.

AVR-архитектура реализована в семействе AT90 и объединяет 8-разрядный гарвардский RISC-процессор, 32 регистра общего назначения, каждый из которых может работать как аккумулятор, и развитую систему команд фиксированной

16-битной длины. Большинство команд выполняются *за один машинный такт* с одновременным исполнением текущей и выборкой следующей команды, что обеспечивает производительность до 1 MIPS на каждый МГц тактовой частоты.

Все микроконтроллеры AVR имеют встроенную Flash-память с возможностью внутрисхемного программирования через последовательный 4-проводной интерфейс. Их периферия включает: таймеры-счётчики, широтно-импульсные модуляторы, поддержку внешних прерываний, аналоговые компараторы, 10-разрядный 8-канальный АЦП, параллельные порты (от 3 до 48 линий ввода и вывода), интерфейсы UART и SPI, сторожеой таймер и устройство сброса по включению питания. Имеется три подсемейства таких микроконтроллеров (tiny AVR, classic AVR и mega AVR), отличающиеся производительностью и объемом памяти.

AMR-архитектура реализована в семействе AT91 и основана на 32-разрядном RISC-процессоре. Микроконтроллеры этого семейства отличаются наиболее высокой производительностью и выпускаются с различными комбинациями RAM, ROM и Flash-памяти. Их достоинством является также наличие EBI–интерфейса (External Bus Interface), обеспечивающего высокоскоростной доступ к дополнительной внешней памяти и внешним периферийным устройствам.

2.4. Микроконтроллеры фирмы Analog Devices

В устройствах обработки аналоговых сигналов широко применяются микроконтроллеры ADuC8xx со встроенным прецизионными АЦП и ЦАП семейства MicroConverter фирмы Analog Devices. В этих изделиях используется процессорное ядро 8051/52, встроенная Flash-память и блок ввода-вывода аналоговой информации, включающий многоканальные ЦАП и АЦП с разрядностью 12 бит и более. Имеется также встроенный датчик температуры и источник опорного напряжения с функциями термостабилизации.

В качестве типичного примера приведем технические характеристики микроконтроллера ADuC812, называемого также “однокристальной системой сбора данных”. Это устройство выполнено в 52-выводном корпусе и имеет 8-разрядный микропроцессор с тактовой частотой 12-16 МГц и системой команд MCS-51, 8 Кбайт резидентной Flash-памяти программ, 640 байт резидентной Flash-памяти данных и 256 байт резидентной RAM-памяти данных. Имеется также возможность адресации до 16 Мбайт внешней памяти данных и до 64 Мбайт внешней памяти программ. Аналоговый интерфейс включает в себя 8-канальный прецизионный 12-разрядный АЦП с быстродействием 5 мкс, два 12-разрядных ЦАП, DMA контроллер для передачи данных от АЦП в память. Встроенные периферийные модули обеспечивают последовательную передачу данных через порты типа UART, SPI и I2C. Питание осуществляется от источника с напряжением от 3 до 5 В, имеется схема слежения за напряжением питания и два режима пониженного энергопотребления.

Новые изделия этой фирмы ADuC702x основаны на 32-разрядном процессорном ARM-ядре и имеют расширенный набор аналоговых входов-выходов. Они предназначены для применения в бортовой автоэлектронике и промышленной автоматике, что обеспечивается аппаратной поддержкой шин типа LIN и LIN, а также простотой сопряжения с электромагнитными, ультразвуковыми и оптическими датчиками положения.

2.5. PIC-микроконтроллеры фирмы Microchip

Микроконтроллеры типа PIC (Peripheral Interface Controller) появились в конце 80-х годов и составили серьёзную конкуренцию контроллерам MCS-51. В их основу была положена 8-разрядная RISC-архитектура с системой простых однословных команд (в базовой модели PIC16C5x используется только 33 команды). Все команды (кроме команд перехода) выполняются за один машинный цикл (четыре такта).

Отличительной особенностью PIC-микроконтроллеров, реализованных по CMOS-технологии, является низкое энергопотреблении: 2 мА при питании 5 В на тактовой частоте 4 мГц; 15 мкА при питании 3 В на тактовой частоте 32 кГц и менее 3 мкА в режиме ожидания. Диапазон питающих напряжений составляет от 2 В до 6 В. Это обуславливает широкое применение таких микроконтроллеров в портативных приборах с батарейным питанием.

В настоящее время Microchip выпускает четыре основных семейства 8-разрядных PIC-микроконтроллеров, программно совместимых снизу вверх:

- PIC15Cx - базовое семейство с 12-разрядными командами и минимальной периферией;
- PIC12Cxxx - семейство с 12-разрядными командами, выпускаемое в миниатюрном 8-выводном исполнении;
- PIC16x/7x/8x/9x – среднее семейство с 14-разрядными командами, средствами аналогового ввода/вывода, контроллерами SPI, USART и I2C;
- PIC17C4x/5xx – высокопроизводительное семейство с 16-разрядными командами, увеличенным объёмом памяти, развитой системой ввода/вывода и повышенной тактовой частотой (до 33 МГц).

Перспективное пятое семейство PIC-контроллеров PIC18Cxxx будет иметь расширенное RISC-ядро, оптимизированное под использование нового Си-компилятора, адресное пространство программ до 2 Мбайт, до 4 Кбайт встроенной памяти данных и производительность 10 млн. операций в секунду.

3. Семейство микроконтроллеров MCS-48

3.1. Архитектура микроконтроллеров MCS-48

Микроконтроллеры семейства MCS-48 являются одним из первых представителей однокристальных ЭВМ, которые в отличие от классических процессоров имеют встроенную память, таймер и интерфейсы для связи с внешними устройствами. Это обусловило их широкое применение в системах управления, где преимущества однокристального исполнения оказались более существенными, чем некоторое снижение быстродействия, характерное для такой архитектуры. Первым представителем этого семейства стала микросхема i8048 разработанная фирмой Intel и использованная в клавиатуре первых компьютеров IBM PC. В её структуре использовано множество архитектурных элементов восьмиразрядных микропроцессоров 8080/85. Далее это семейство развивалось путём повышения степени интеграции, быстродействия (тактовой частоты) и снижения энергопотребления табл. 1.

Таблица 1

Микроконтроллеры, входящие в состав семейства MCS-48

Тип	Аналог	Тип ПЗУ	Объем ПЗУ, Кбайт	Объем ОЗУ, байт	Тактовая частота, МГц	Ток потребления, мА
Первое поколение						
8035	1816ВЕ35	внешнее	—	64	6	135
8048	1816ВЕ48	с УФ - стиранием	1	64	6	135
Второе поколение						
8039	1816ВЕ39	Внешнее	—	128	11	110
8049	1816ВЕ49	Масочное	2	128	11	110
Третье поколение						
80С35	1830ВЕ35	Внешнее	—	64	6	8
80С48	1830ВЕ48	Масочное ПЗУ	1	64	6	8

Все функции микроЭВМ в семействе MCS-48 реализуются с помощью единственной микросхемы. В состав семейства MCS-48 входит целый ряд микросхем с одной неизменной частью, – ядром, определяющим систему команд, что позволяет использовать для разработанного устройства микросхемы различных фирм-производителей без изменения принципиальной схемы устройства и программы. Отличия микросхем в семействе MCS-48 определяются периферийными компонентами, позволяющими выполнять как задачи управления

различными устройствами, так и реализовывать отдельные узлы аналоговой схемы.

Основу микроконтроллера (рис. 1) составляет двунаправленная восьмиразрядная системная шина, связывающая между собой все основные компоненты микросхемы:

- блок синхронизации и управления;
- арифметико-логическое устройство (АЛУ);
- блок таймеров/счётчиков;
- блок последовательного интерфейса и прерываний;
- программный счётчик;
- резидентную память данных (РПД) и память программ (РПП).

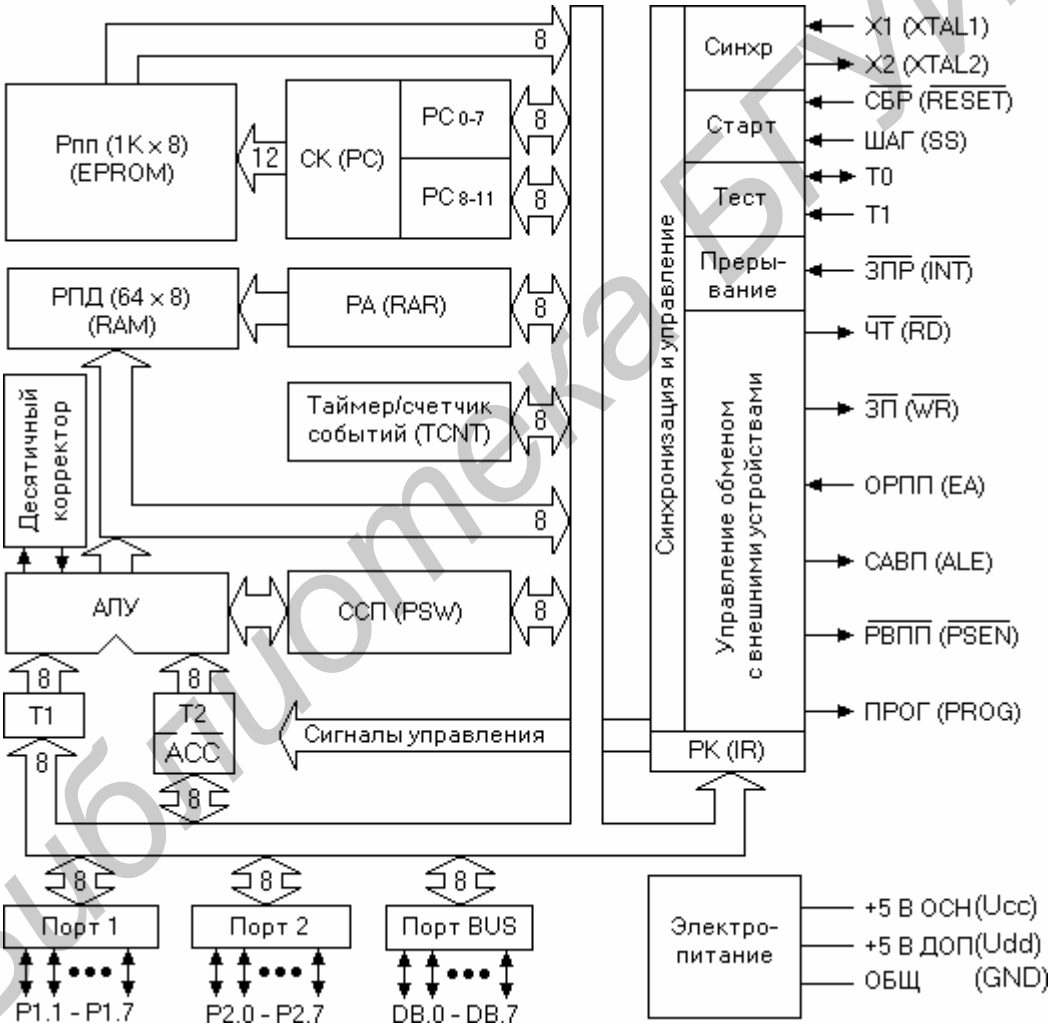


Рис. 1. Структурная схема микроконтроллера 1816ВЕ48

Конструктивно микроконтроллер выполнен в корпусе БИС с 40 внешними выводами, электрически совместимыми с элементами ТТЛ. Питание МК осуществляется с помощью двух выводов: первый, – для подачи основного, и второй, – дополнительного напряжения для питания резидентной памяти данных или программирования резидентной памяти программ. К выводам X1 и X2

подключается кварцевый резонатор или внешний источник синхронизации. Сигнал SS (ШАГ) совместно с сигналом ALE (САВП) позволяет при отладке выполнять программу с остановом после исполнения очередной команды. Выводы T1 и T0 опрашиваются командами условного перехода и могут использоваться для подсчёта внешних событий. Сигнал запроса прерывания ЗПР от внешнего источника вызывает подпрограмму обслуживания прерывания. Высокий уровень на входе EA (ОРПП) заставляет МК выполнять выборку команд только из внешней памяти программ, что позволяет тестировать прикладную программу при её отладке. Этот вывод также используется для подачи напряжения программирования +25В. Контроллер имеет три параллельных восьмиразрядных порта ввода/вывода, из которых порт BUS может использоваться в качестве внешней шины данных процессорного ядра контроллера.

3.2. Блок управления и синхронизации

Совместную работу блоков микроконтроллера во всех режимах работы координирует блок управления и синхронизации. В его состав входят: устройство синхронизации, логика ввода-вывода, регистр команд, дешифратор команд, логика управления ЭВМ.

Опорную частоту синхронизации определяет либо кварцевый резонатор, подключаемый к выводам X1 и X2, либо LC-цепь (рис. 2а и 2б соответственно). X1 является входом, а X2 – выходом генератора, способного работать в диапазоне частот от 1 до 6 МГц. На вход X1 может подаваться сигнал от источника внешней синхронизации рис. 2в. В состав генератора МК входят два счётчика с модулями пересчёта 3 и 5. Первый используется для формирования сигнала системной синхронизации (0,5 мкс). Этот же сигнал поступает на счётчик машинных циклов, на выходе которого через каждые пять сигналов синхронизации формируется сигнал ALE (2,5 мкс), идентифицирующий машинный цикл и использующийся в расширенных МК-системах для стробирования адреса внешней памяти.

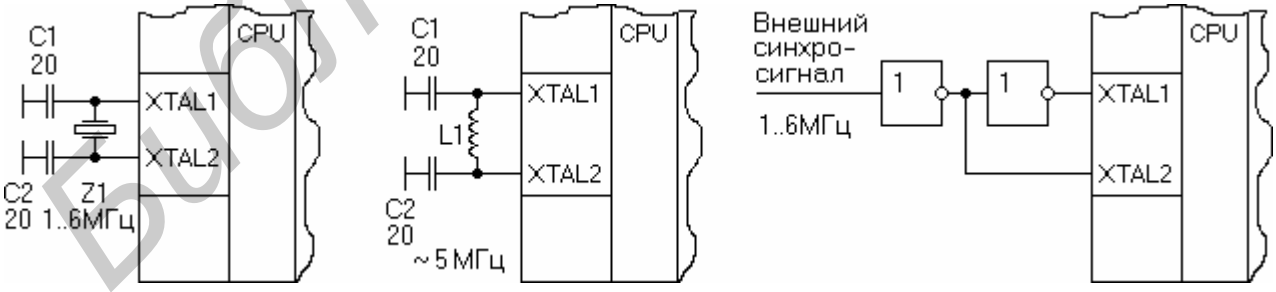


Рис. 2. Варианты схем синхронизации MCS-48

Сброс микроконтроллера в обслуживаемых МК-системах осуществляется с помощью кнопки “Сброс”, которая заземляет вход RESET (рис. 3а). В необслуживаемых МК-системах, ко входу RESET подключается конденсатор ёмкостью 1 мкФ, что обеспечивает подачу сигнала близкого к потенциалу земли,

длительностью не менее 50 мкс после установки в номинальное значение напряжения питания (рис. 3б). Сигнал RESET производит следующие действия: сбрасывает счётчик команд (PC) и указатель стека (SP); устанавливает порт BUS в высокоимпедансное состояние, а порты P1 и P2 – в режим ввода (FF в выходном буфере); выбирает банк регистров 0 и банк памяти 0; запрещает прерывания; останавливает таймер и выдачу синхросигнала на вывод T0; сбрасывает флаг переполнения таймера TF и флаги пользователя F0 и F1.

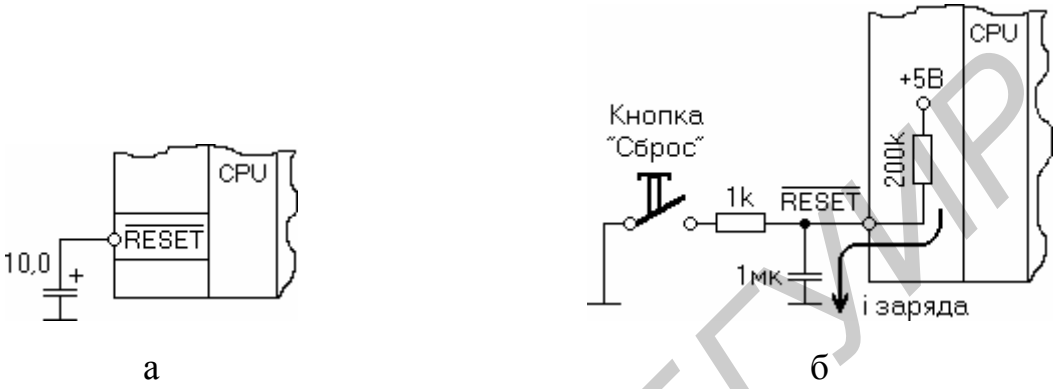


Рис. 3. Схемы начальной установки MCS-48

Для записи и хранения восьмиразрядного кода операции выполняемой команды предназначен регистр команд. Код операции, с помощью дешифратора команд и логики управления преобразуется в микропрограмму выполнения команды, считанной из ПЗУ.

3.3. Арифметико-логическое устройство

Восьмиразрядное арифметико-логическое устройство обеспечивает выполнение арифметических и логических операций и состоит из регистров временного хранения T1 и T2, аккумулятора (ACC), ПЗУ констант, арифметико-логического устройства и регистра слова-состояния программы (PSW).

Программно недоступные восьмиразрядные регистры временного хранения (T1 и T2) используются для временного хранения второго операнда при выполнении двухоперандных команд. Результат выполнения арифметико-логических операций или операций сдвига сохраняется в восьмиразрядном аккумуляторе. ПЗУ констант обеспечивает генерацию корректирующего кода при двоично-десятичном представлении данных, кода маски при битовых операциях и кода констант.

Арифметико-логическое устройство представляет собой схему комбинационного типа с последовательным переносом, предназначенную для выполнения операций сложения, вычитания и логических операций И, ИЛИ, исключающего ИЛИ, инкремента, декремента, инверсии, циклического сдвига вправо и влево, десятичной коррекции содержимого аккумулятора и обмена тетрад в байте.

При выполнении операций обработки данных в АЛУ, формируются флаги:

- переноса (C);
- вспомогательного переноса (AC);
- нулевого содержимого аккумулятора (Z);
- наличия единицы в заданном бите аккумулятора.

Следует заметить, что в регистре слова состояния программы PSW (рис. 4) сохраняются только два из этих флагов (C, AC). Остальные флаги формируются комбинационной схемой и не фиксируются, но логика условных переходов позволяет выполнять команды передачи управления (JZ, JNZ, JB0..JB7) без их фиксации триггерами. Кроме этого имеются условные переходы по “флагам пользователя” F0 и F1, по флагу переполнения таймера TF, а также по сигналам на тестируемых входах T0 и T1. Программист также имеет доступ к флагам рабочего банка регистров BS и выбранного банка внешней памяти программ MB. Кроме того, после окончания каждого машинного цикла опрашивается флаг разрешения/запрета прерывания.

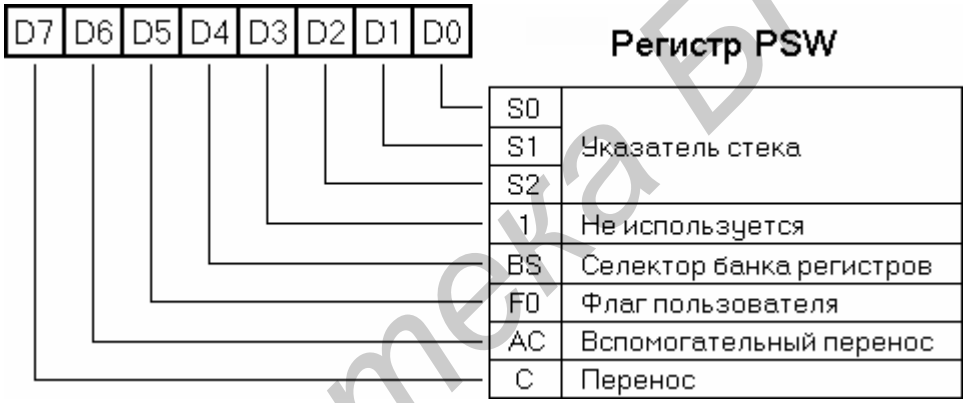


Рис. 4. Формат регистра слова-состояния программы PSW

Счётчик команд (Program Counter) формирует текущий 12-разрядный адрес памяти программ. В состав счётчика команд входят регистр PC и схема инкремента одиннадцати разрядов адреса (старший разряд PC не инкрементируется).

3.4. Память микроконтроллеров семейства MCS-48

В семействе микроконтроллеров MCS-48 подсистема памяти реализована по гарвардской архитектуре, что предполагает её физическое и логическое разделение на память программ и память данных с отдельным адресным пространством. В этих микроконтроллерах имеется три адресных пространства: память программ, внешняя память данных (ВПД) и внутренняя (резидентная) память данных (РПД). Такое построение памяти позволяет создавать максимально надёжные системы.

Схема подключения внешних микросхем памяти к микроконтроллерам семейства MCS-48 показана на рис. 5. Регистр адреса D2 на этой схеме

предназначен для запоминания младших восьми бит адреса, передаваемых через шину данных/памяти. Старшие четыре бита адреса передаются через шину адреса, совмещенную с портом P2. Выдача адреса сопровождается синхроимпульсом ALE, который позволяет запомнить младший байт адреса в регистре D2.

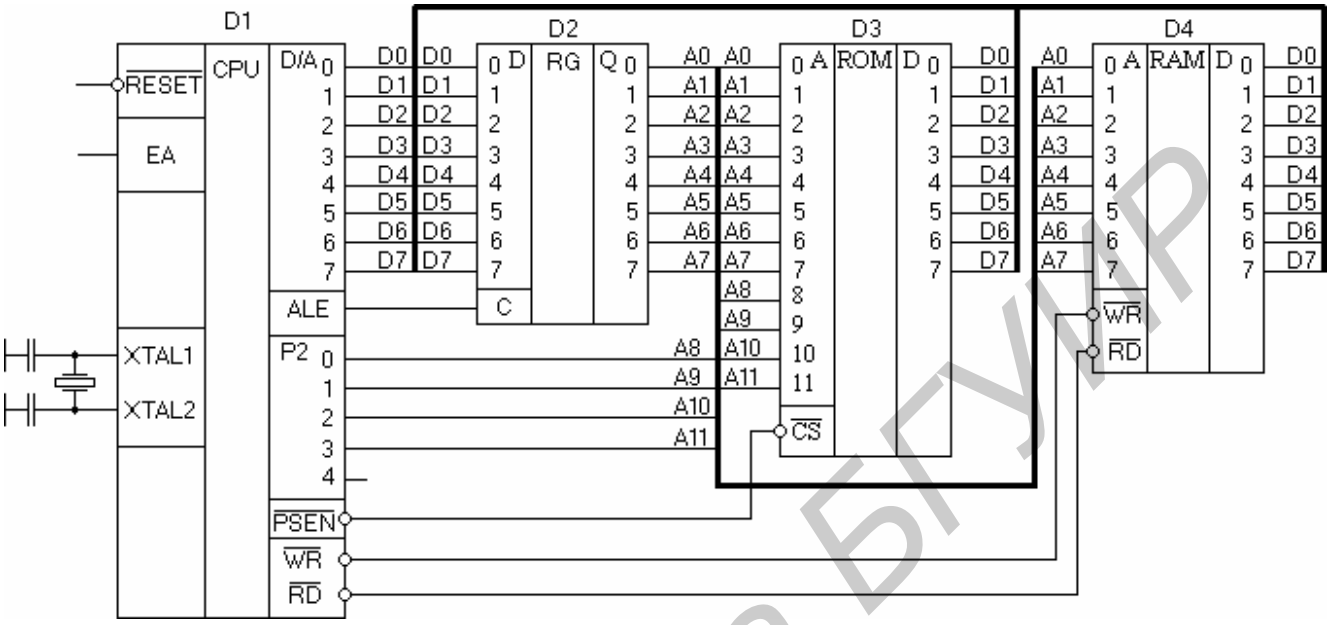


Рис. 5. Подключение внешней памяти к микроконтроллерам MCS-48

Для обращения к памяти программ D3 и к памяти данных D4 используется совмещённая шина данных/адреса BUS, а её текущее назначение определяется управляющими сигналами. При чтении памяти программ используется сигнал PSEN, а при чтении памяти данных, – сигнал RD. Запись информации в память данных осуществляется по сигналу WR.

3.5. Память программ

Память программ, размещённая на кристалле микроконтроллера, называется резидентной (РПП) и представляет собой постоянное запоминающее устройство. В семейство MCS-48 входят микроконтроллеры с различным объёмом резидентной памяти программ, а у 1816ВЕ35 РПП отсутствует, поэтому они могут работать только с внешней микросхемой ПЗУ ёмкостью до 4 Кбайт. Максимальное адресное пространство памяти программ составляет 4 Кбайта, поэтому счётчик команд двенадцатиразрядный (4 Кбайт = 4096 = 2^{12} байт). Адреса РПП находятся в диапазоне 000h..3FFh. После сброса микроконтроллера выполнение программы начинается с нулевого адреса, соответствующего внутреннему ПЗУ микроконтроллера.

Микроконтроллеры семейства MCS-48 имеют вывод EA, с помощью которого можно запретить работу внутренней памяти, подав на этот вход логический “0”. При этом, начиная с нулевого адреса, все обращения происходят

к внешней памяти программ. Поэтому доступ к внешней памяти программ осуществляется в двух случаях:

1. при подаче сигнала EA=0 независимо от адреса обращения,

2. в случае, если счётчик команд (PC) содержит число большее, чем 3FFh.

Следует заметить, что при выполнении программы инкрементируются только младшие 11 бит счётчика команд. Поэтому программа из предельного адреса 7FFh (если только по этому адресу не расположена команда передачи управления) перейдёт по адресу 0000h. Состояние старшего бита счётчика команд может быть изменено специальными командами (SEL MB0 и SEL MB1) совместно с командами безусловного перехода JMP и вызова подпрограмм CALL. При этом следует учитывать, что команды SEL MB0 и SEL MB1 изменяют только флаг MB и не воздействуют непосредственно на старший бит счётчика команд. Изменение этого бита осуществляется командами JMP и CALL, которые копируют текущее содержимое MB в старший бит счётчика команд. Поэтому для перехода из нулевого банка в первый требуется выполнить последовательность команд SEL MB1, JMP adr1.

При выполнении переходов между банками по командам SEL MB, необходимо следить за тем, чтобы подпрограммы, взаимно вызывающие друг друга, располагались в пределах одного банка памяти. Иначе возникает необходимость модификации флага MB в вызываемой подпрограмме и восстановления его при возврате в вызывавшую подпрограмму. Но если вызов такой подпрограммы носит условный характер, то проблема восстановления флага может оказаться неразрешимой.

Неполная (11-битная) инкрементация счётчика команд приводит к тому, что память программ разделяется на два банка ёмкостью по 2 Кбайта. Кроме того, в командах условного перехода задаётся 8-битный адрес, что обуславливает разбиение памяти программ на 16 страниц по 256 байт. В случае, когда в программе необходимо иметь много условных переходов, из-за небольшого размера страницы возникает проблема размещения соответствующих программных модулей на двух страницах. Для решения этой проблемы используют комбинации команд условного перехода с 8-разрядным адресом и команды безусловного перехода с 11-разрядным адресом. Адресное пространство памяти программ при этом распределено, как показано на рис. 6.

В резидентной памяти программ имеется три специализированных адреса:

- адрес 0, которому передаётся управление по сигналу RESET; по этому адресу должна находиться команда безусловного перехода к началу программы;
- адрес 3, по которому передаётся управление по сигналу внешнего прерывания;
- адрес 7, по которому передаётся управление по прерыванию от таймера/счётчика.

При обработке запросов прерываний в МК48 старший бит счётчика команд принудительно устанавливается в 0. Это приводит к необходимости располагать подпрограмму обслуживания прерывания и все вызываемые ею подпрограммы в нулевом банке памяти.

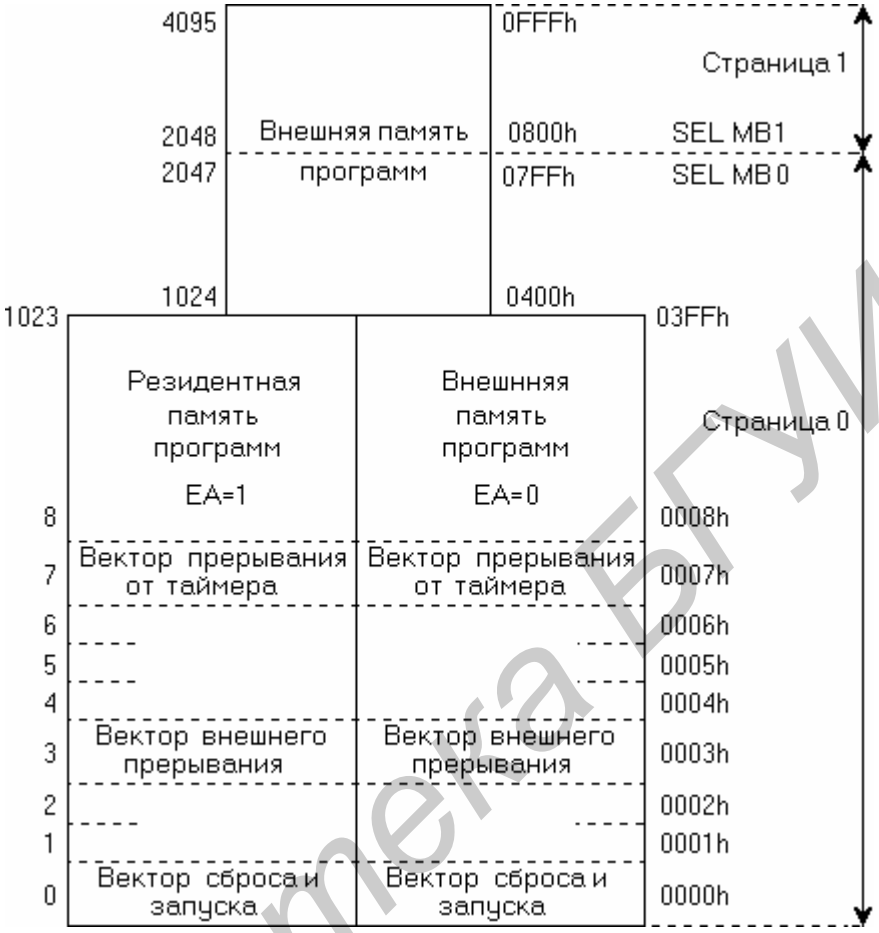


Рис. 6. Карта адресов памяти программ 1816ВЕ48

Семейство MCS-48 имеет несколько модификаций, отличающихся объёмом резидентной памяти программ. При использовании портов P1 и P2 в качестве дополнительных адресных расширителей объём внешней памяти программ может быть увеличен до 16 Мбайт.

3.6. Память данных

Резидентная память данных (РПД) ёмкостью 64 байта имеет в своём составе два банка рабочих регистров R0…R7, причём программист может выбрать один из них в качестве активного. Для размещения регистров используются ячейки памяти 0-7 и 24-31 (рис. 7), поэтому символу R0 в программе может соответствовать как ячейка памяти 0, так и ячейка с адресом 24. Такая организация позволяет сократить время обработки прерываний, так как вместо традиционного сохранения содержимого регистров в стеке можно переключить активный банк регистров. Выбор активного банка регистров выполняется командой SEL RB.

Любая ячейка РПД доступна по командам с косвенной адресацией через регистры R0, R1. При этом используются символы @R0 и @R1. Однако при адресации ячеек 0-7 и 24-31 предпочтительно использовать символы R0…R7, соответствующие прямой регистровой адресации.

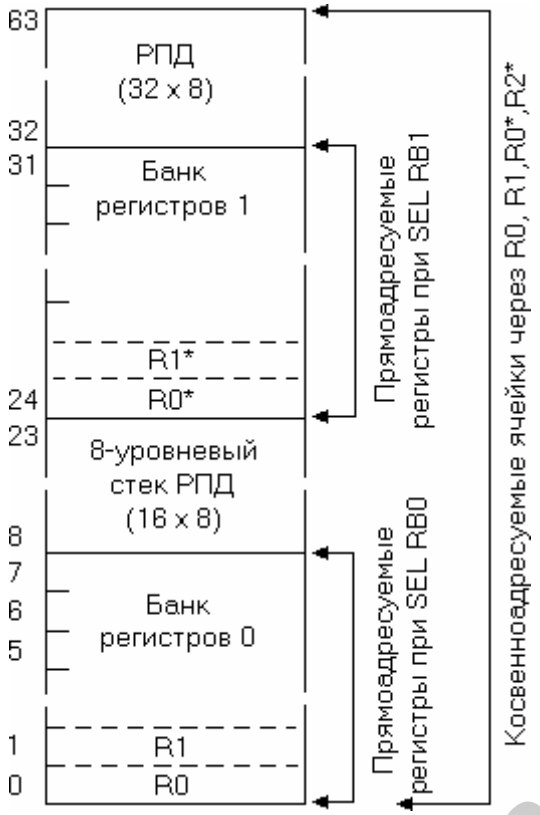


Рис. 7. Карта адресов памяти данных

В РПД размещается также стек объёмом 16 байт, имеющий 8 уровней по 2 байта (область памяти с адресами 8-23). В качестве указателя стека используются три разряда слова состояния программы PSW. В стеке сохраняется адрес возврата из подпрограммы (12 разрядов) и старшая тетрада PSW (4 разряда).

При уровне вложенности подпрограмм меньше восьми, незадействованные в стеке ячейки могут использоваться как ячейки РПД. При переполнении стека трёхбитный указатель стека переходит из состояния 7 в состояние 0.

Семейство MCS-48 не имеет команд загрузки байта в стек или его извлечения из стека, обычно используемые при обработке прерываний. Поэтому программист должен следить за тем, чтобы вложенные подпрограммы не использовали одни и те же рабочие регистры.

Практически все команды с обращением к РПД оперируют с одним байтом. Однако по командам вызова и возврата осуществляется доступ к двухбайтным словам. В памяти данных слова хранятся так, что старший байт располагается в ячейке с большим адресом. Отметим, что в памяти программ порядок расположения байтов по старшинству при хранении двухбайтных слов обратный.

В МК-системах с внешней памятью данных адресация осуществляется также через регистры R0 и R1 (символы @R0 и @R1), что обеспечивает доступ к памяти объёмом 256 байт.

3.7. Организация ввода/вывода информации

Для организации ввода/вывода информации микроконтроллеры семейства MCS-48 имеют 27 линий, в том числе три параллельных порта (BUS, P1 и P2) по восемь линий в каждом, а также три линии, которые могут использоваться для изменения хода программы по командам условного перехода и сигналу прерывания:

- INT – для ввода в МК сигнала запроса прерывания от внешнего источника либо для ввода тестируемого сигнала;
- T0 – для ввода тестируемого сигнала от двоичного датчика (по команде ENT0 CLK по этой линии выдается сигнал синхронизации);
- T1 – для ввода тестируемого сигнала или ввода импульсов для счётчика событий, который запускается по команде STRT CNT.

Порт BUS является двунаправленным и имеет буферные каскады с тремя состояниями. Он может использоваться либо в качестве шины данных/адреса при подключении внешней памяти, либо как параллельный порт ввода/вывода.

При подключении внешней памяти данных на линиях порта BUS сначала выставляется 8-разрядный адрес, сопровождаемый сигналом ALE. Затем по тем же линиям порта BUS производится передача байта данных, стробируемых сигналами RD или WR (при чтении или записи соответственно). Поэтому на время передачи данных адрес должен фиксироваться во внешнем регистре (см. рис. 5, регистр D2).

При подключении внешней памяти программ требуется формировать 12-разрядный адрес, младшие 8 разрядов которого передаются аналогично (по линиям порта BUS в сопровождении сигнала ALE) и также запоминаются во внешнем регистре. Старшие 4 разряда передаются через порт P2 (линии P2.0..P2.3) и в течение машинного цикла не изменяются, поэтому для старших разрядов адреса буферизация не требуется. Данные из внешней памяти программ передаются в микроконтроллер в сопровождении сигнала PSEN (РВПП); очевидно, что запись в память программ невозможна.

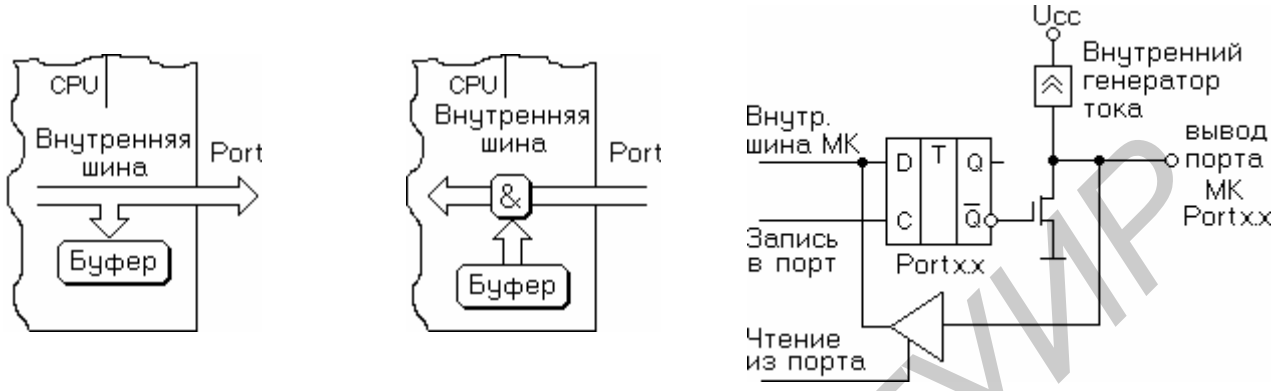
При работе порта BUS в режиме параллельного интерфейса вывод/вывод выполняется командами OUTL и INS. Кроме того, вводимые и выводимые через порт BUS данные можно модифицировать с помощью команд ORL BUS, #d и ANL BUS, #d, что позволяет выделять и обрабатывать в байте отдельный бит или группу бит. Такая операция называется маскированием.

Заметим, что обмен данными через порт BUS осуществляется командами пересылки MOVX, а также ввода/вывода OUTL и INS. При этом в случае записи во внешнюю память данных командой MOVX @R_i, A или выводе данных через порт командой OUTL BUS, A соответствующий байт фиксируется в буферном регистре и сохраняется на выводах микроконтроллера до выполнения следующей команды такого типа. Однако при чтении данных командой MOVX A, @R_i содержимое буфера уничтожается, в то время, как команда ввода INS не изменяет содержимое буферного регистра. Поэтому при работе с портом BUS не рекомендуется попеременное использование команд пересылки данных и команд ввода/вывода.

В отсутствие передач данных линии порта BUS отключаются от нагрузки, переходя в высокоимпедансное состояние.

Порты ввода/вывода P1 и P2 являются квазидвунаправленными. Это означает, что вывод информации осуществляется аналогично порту BUS (без

искажений), но при вводе производится операция поразрядного логического “И” между входными сигналами и содержимым буфера (рис. 8). Поэтому для ввода данных без изменений перед операцией чтения IN необходимо записать во все разряды буфера логические единицы. Заметим, что по сигналу RESET в буферные регистры портов P1 и P2 автоматически записывается FFh.



а б в

Рис. 8. Функциональная схема квазидвунаправленного порта:
а – в режиме вывода, б – в режиме ввода данных; в – для одного разряда порта

Для ввода/вывода информации через порты P1 и P2 используются команды IN и OUTL:

```
IN A, P1        ; ввести данные из порта P1 в аккумулятор
OUTL P2, A      ; вывести данные из аккумулятора в порт P2.
```

Квазидвунаправленная структура портов позволяет аппаратно выполнять *маскирование* требуемых линий при вводе, т.е. запись нулей в заданный разряд или группу разрядов. Например, если требуется обрабатывать только младшие 4 линии порта P2, то можно использовать последовательность команд:

```
MOV  A, #0Fh    ; запись маски в аккумулятор;
OUTL P2, A      ; загрузка маски в буфер порта P2;
IN  A, P2       ; ввод младшей тетрады и маскирование старшей,
```

выполнение которых обеспечивает запись в нулей в старшую тетраду аккумулятора. Следует учитывать, что текущее содержимое буфера P1 и P2 для чтения не доступно, поэтому программист должен быть уверен, что в буфере находится требуемая маска (в противном случае ее необходимо загрузить повторно).

Порты P1 и P2 архитектурно идентичны, за исключением четырёх младших линий порта P2, которые могут использоваться для:

- выдачи старшей тетрады адреса ВПП,
- подключения расширителя КР580ВР43 (рис. 9), позволяющего увеличить количество линий ввода/вывода на шестнадцать (четыре четырёхразрядных порта P4-P7).

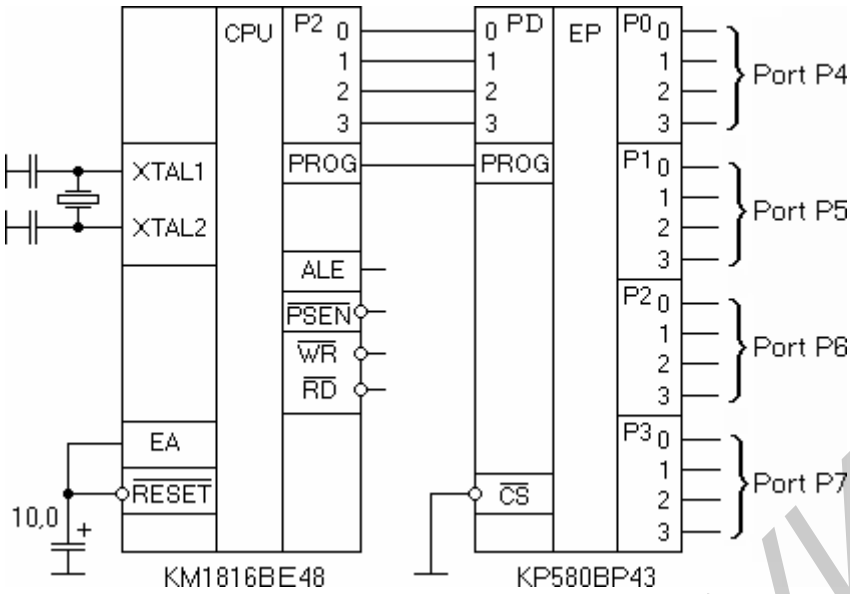


Рис. 9. Подключение расширителя ввода/вывода

Кроме обычных команд ввода/вывода IN, OUTL для изменения состояния линий P1, P2 можно использовать и логические команды ANL, ORL. Например:

ANL P1, #11100111b ;Установить логический “0” на линиях P1.3 и P1.4

ORL P2, #00011000b ;Установить логическую “1” на линиях P2.3 и P2.4

Такой приём позволяет заменить одной командой последовательность из трёх команд (ввод, логическая операция с аккумулятором, вывод).

3.8. Система обработки прерываний

Система прерываний представляет собой механизм, позволяющий реагировать на особые состояния, возникающие при работе системы управления (срабатывание датчика аварийной ситуации, пропадание питания и т.д.). С точки зрения программиста прерывание – это принудительная приостановка текущей программы и переход на выполнение другой программы (обработчика прерывания), после завершения которой управление возвращается прерванной программе.

Микроконтроллеры семейства MCS-48 позволяют обрабатывать прерывания от двух источников:

- внешнего сигнала на входе INT (адрес вектора прерывания 3);
- флага переполнения таймера-счётчика (адрес вектора прерывания 7).

Прерывание от внешнего источника возникает, когда на входе INT появляется низкий уровень. Состояние сигнала на этом входе анализируется во время выдачи сигнала ALE (в каждом машинном цикле), а обработка прерывания начинается после завершения текущей команды. При обработке внешнего прерывания выполняются следующие действия:

– содержимое счётчика команд (адрес возврата, 12 разрядов) и старшей тетрады слова состояния программы (флаги, 4 разряда) сохраняется в стеке;

– в счётчик команд записывается значение 003h (по этому адресу должна находиться команда JMP, обеспечивающая переход к обработчику прерывания, расположенному в MB0).

Обработка прерывания должна заканчиваться командой RETR, по которой:

– восстанавливается из стека содержимое счётчика команд (адрес возврата) и старшей тетрады слова состояния программы;
– продолжается выполнение прерванной программы.

Ко входу INT микроконтроллера через «монтажное ИЛИ» могут быть подключены несколько источников прерываний, однако другие запросы прерываний игнорируются до выхода из подпрограммы обработки по команде RETR. В этом случае источник прерывания определяется программно (необходимо предусмотреть возможность ввода сигналов от всех источников прерываний через порт P1 или P2).

Поскольку система прерываний MCS-48 реагирует на уровень сигнала на входе INT (а не на переход из “1” в “0”), то для исключения повторных прерываний сигнал INT должен быть снят до выполнения команды RETR. Эта операция должна выполняться автоматически, в противном случае обработчик прерываний должен снять низкий уровень с линии INT через какую-либо линию порта P1 или P2.

Прерывания от таймера/счётчика возникают, когда флаг переполнения TF устанавливается в “1”. Они обрабатываются аналогично, но переход осуществляется по адресу 007h. Кроме того, флаг TF сбрасывается автоматически при переходе на подпрограмму обработки прерывания.

Приоритет внешнего прерывания выше приоритета прерывания от внутреннего таймера. Это означает, что в случае одновременного запроса прерываний от внешнего источника и запроса от флага переполнения таймера приоритет остается за источником, воздействующим на вход INT.

При необходимости, в МК можно создать двухуровневую систему прерываний. Для этого необходимо загрузить в таймер/счётчик число FFh и перевести его в режим счётчика. Тогда изменение сигнала на входе T1 из “1” в “0” приведёт к прерыванию по вектору 007h.

Внешние прерывания могут быть запрещены командой DIS I и разрешены командой EN I. Для разрешения и запрещения прерываний от таймера используются команды EN TCNTI и DIS TCNTI. Сигнал RESET запрещает прерывания от обоих источников. Если внешние прерывания запрещены, то вход INT может быть проверен командой условного перехода JNI (при этом он используется в качестве дополнительного тестируемого входа подобно T0 и T1).

При входе в подпрограммы обслуживания прерывании старший бит счётчика команд PC_{11} принудительно устанавливается в нуль. Поэтому обработчики прерываний должны размещаться в банке памяти MB0.

3.9. Таймер/счётчик

MCS-48 имеет встроенный восьмиразрядный таймер/счётчик (рис. 10), который может подсчитывать импульсы от внутреннего генератора (режим таймера) или импульсы от внешнего источника (режим счётчика).

В режиме таймера через делитель частоты на 32 поступают сигналы основной синхронизации с периодом 2,5 мкс (400 кГц), поэтому состояние счётчика увеличивается на 1 каждые 80 мкс (12,5 кГц). Путём программной установки таймера/счётчика в исходное состояние и анализа флага переполнения могут быть реализованы различные временные задержки, лежащие в диапазоне от 80 мкс до 20,48 мс. Временные задержки, превышающие по длительности 256 состояний счётчика (20,48 мс), могут быть получены накоплением переполнений в рабочем регистре. Заметим, что период работы таймера 20,48 мс примерно соответствует частоте 50 Гц.

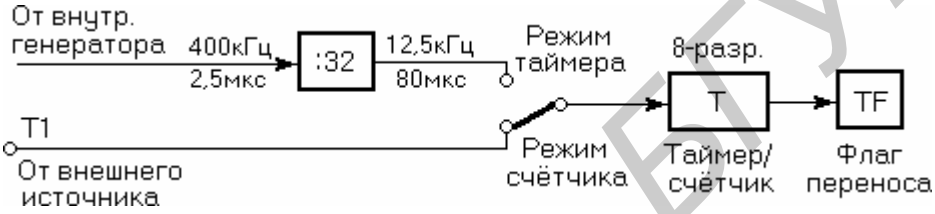


Рис. 10. Таймер микроконтроллеров семейства MCS-48

В режиме счётчика производится аппаратный подсчёт внешних импульсов, поступающих на тестируемый вход T1. При этом состояние счётчика изменяется в момент перехода входного сигнала из “1” в “0”. Количество подсчитанных импульсов можно прочитать командой MOV A, T. Минимальная длительность импульса на входе T1 равна 0,5 мкс, а минимальное время между импульсами – 7,5 мкс (при частоте кварцевого резонатора 6 МГц).

Из максимального состояния FFH таймер/счётчик переходит в начальное состояние 00H, при этом устанавливается флаг переполнения TF, что вызывает прерывание по вектору 007h (если оно разрешено командой EN TCNTI). Если прерывания от таймера/счётчика запрещены, то флаг переполнения может быть опрошен командой условного перехода JTF. Выполнение команды JTF, как и переход к подпрограмме обработки прерывания, сбрасывает флаг TF.

Выбор режима работы таймера/счётчика осуществляется специальными командами. Команда STRT T включает режим таймера, команда STRT CNT включает режим счётчика, а команда STOP TCNT останавливает как таймер, так и счётчик. Заметим, что сигнал RESET также останавливает таймер/счётчик.

Если требуется реализовать временную задержку, кратную 80 мкс, но меньшую 20,48 мс, то можно предварительно загрузить в таймер число, соответствующее количеству тактов в дополнительном коде. Например, для формирования временного интервала 4 мс (50 тактов по 80 мкс) загружается константа 256 – 50 = 206, что соответствует CEh:

```
MOV A, #CEh        ; количество периодов (в дополнительном коде)
```

```
        MOV T, A        ; загрузка таймера
        STRT T          ; запуск таймера
M0:     JTF M1          ; ожидание переполнения таймера
        JMP M0
M1:     STOP TCNT       ; остановка таймера
                        . . . . . .
```

На основе этой программы также можно реализовать задержки большей длительности. Для этого используется регистр, в который заносится требуемое число переполнений таймера, и организуется цикл, внутри которого осуществляется ожидание переполнения флага TF. Заметим, что наиболее эффективно таймер используется в режиме прерываний, когда для обнаружения переполнения не используются ресурсы процессора (при этом процессор может выполнять другую программу, обеспечивающую, например, отображение информации или анализ нажатия клавиш).

3.10. Система команд MCS-48

Система команд микроконтроллеров семейства MCS-48 содержит 96 базовых команд, имеющих длину один или два байта. Большинство команд являются однобайтными и выполняются за один машинный цикл длительностью 2,5 мкс (при частоте 6 МГц). Двухбайтные команды выполняются за 5 мкс.

В микроконтроллерах реализована как побайтовая (8 бит) так и потетрадная (4 бита) обработка данных. К достоинствам системы команд MCS-48 можно отнести: эффективный ввод/вывод, включая маскирование и возможность управления отдельными битами портов; возможность ветвления по значению отдельных бит; возможность обработки как двоичных, так и десятичных двоично-кодированных чисел. Форматы команд приведены на рис. 11.

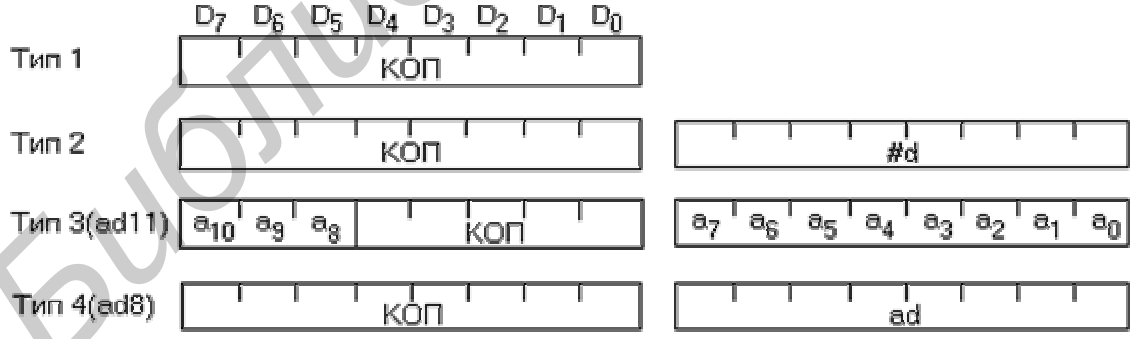


Рис. 11. Типы команд МК48
(КОП – код операции, #d – непосредственный операнд, $a_{10..0}$ – адрес)

Систему команд микроконтроллера условно можно разбить на пять групп: арифметические, логические, команды передачи данных, команды ветвления программ и передачи управления, команды управления режимами. В MCS-48 используются четыре способа адресации:

1. **Непосредственная**, – в теле команды содержится восьмибитный операнд (во втором байте команды):

```
MOV A, #05h        ; (A) ← 05h
```

2. **Прямая**, – в теле команды содержится адрес операнда (номер регистра):

```
MOV A, Rn          ; (A) ← (Rn)
```

3. **Косвенная**, – в теле команды содержится адрес адреса операнда (номер регистра, в котором расположен адрес ячейки памяти):

```
MOV A, @R0         ; (A) ← ((R0))
```

Косвенная адресация допускается только через два регистра: @R0 или @R1.

4. **Неявная**, – операнд или его адрес не указывается в явном виде, а определяется кодом операции (по умолчанию). Обычно таким операндом является содержимое аккумулятора:

```
DA A               ; двоично-десятичная коррекция A
```

Хотя в мнемонике этой команды аккумулятор указывается явно, в формате команды нет адресного поля (т.е. неявно подразумевается, что операндом может быть только аккумулятор).

3.11. Команды пересылки данных

С помощью этих команд можно произвести копирование либо обмен данных. Ячейка памяти или регистр, в котором располагается исходный байт называется *источником* S (source), а ячейка или регистр, в который производится копирование, – *приёмником* D (destination). Для рассматриваемого микроконтроллера команды пересылки записываются в виде MOV D, S, т.е. сначала указывается приёмник, а затем источник.

В семействе MCS-48 источником и приёмником могут быть аккумулятор A, регистры R0..R7, ячейки памяти, порты ввода/вывода, а также регистр таймера T и слово состояния процессора PSW. Однако система команд не позволяет переслать данные из любого источника в любой приёмник, поэтому большинство пересылок выполняется через аккумулятор (рис. 11).

Передача данных возможна в двух режимах: *пересылки* (загрузки) и *обмена*. При этом команды пересылки не изменяют содержимого источника, а команды обмена изменяют значения обоих операндов, участвующих в операции. Команды пересылки внутри микроконтроллера выполняются за один машинный цикл, обмен с внешней памятью и портами требует двух машинных циклов. Все команды пересылки (кроме MOV PSW, A) не изменяют флаги.

Схема возможных пересылок представлена на рис. 12, где показаны 9 типов операндов: аккумулятор A, регистры общего назначения R0..R7, PSW, таймер, порты, непосредственный операнд #d, внешняя и резидентная память данных, а также память программ (только для чтения). Большинство команд выполняет пересылку 8-битных операндов. Существуют также несколько команд,

оперирующих с четырёхбитными операндами (при обращении к 4-битным портам расширителя ввода/вывода P4..P7).

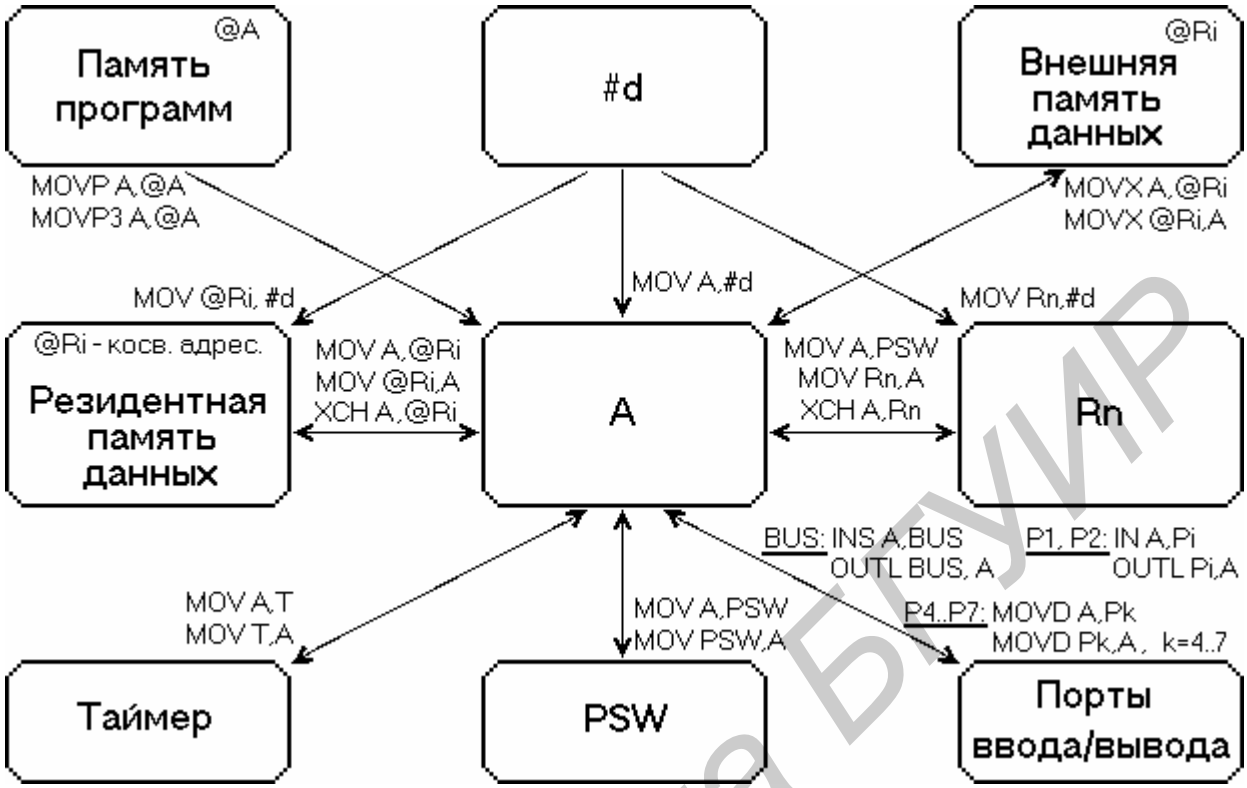


Рис. 12. Команды пересылки данных MCS-48

Так как в микроконтроллере существуют три независимых области памяти, а также порты ввода/вывода, то для обращения к ним используются различные команды:

- MOV, XCH – пересылка данных внутри микроконтроллера;
- MOVX – пересылка данных между аккумулятором и ВПД;
- MOVP – пересылка данных из памяти программ;
- IN, OUTL, MOVD – пересылка из портов в аккумулятор и наоборот.

Для обращения к резидентной памяти данных используется косвенная адресация через регистры R0 и R1:

MOV A, @R0 ; переслать в аккумулятор содержимое ячейки РПД с адресом, хранящемся в R0;

MOV @R1, A ; переслать содержимое аккумулятора в ячейку РПД с адресом, хранящемся в R1.

Для аккумулятора и регистров можно выполнить также команду обмена:

XCH A, R0 ; обменять содержимое аккумулятора с содержимым регистра R0 текущего банка.

В памяти программ наряду с командами можно располагать таблицы функций *sin x*, *cos x*, *sqrt x*, а также константы π, *e* и др. Их можно скопировать в аккумулятор с помощью команд с косвенной адресацией:

MOVP A, @A ; переслать в аккумулятор байт из текущей страницы памяти программ (адрес внутри страницы содержится в аккумуляторе);

MOVP3 A, @A ; переслать в аккумулятор байт из третьей страницы памяти программ (адреса – 300h..3FFh).

Для обращения к внешней памяти данных также используется косвенная адресация через регистры R0 и R1:

MOVX A, @R0 ; переслать в аккумулятор содержимое ячейки ВПД с адресом, хранящемся в R0;

MOVX @R1, A ; переслать содержимое аккумулятора в ячейку ВПД с адресом, хранящемся в R1.

Для работы с параллельными портами используется команды ввода/вывода IN и OUTL (для 8-разрядных операндов) и команды MOVD (для 4-разрядных операндов):

IN A, P0 ; ввести байт из порта P0 в аккумулятор;

OUTL P1, A ; вывести байт из аккумулятора в порт P1;

MOVD A, P4 ; ввести данные из порта P4 в младшую тетраду аккумулятора (старшая тетрада обнуляется);

MOVD P7, A ; вывести данные из младшей тетрады аккумулятора в порт P7.

При использовании команд пересылки данных следует учитывать, что большинство из них не изменяет флагов переноса C и полупереноса AC (за исключением команды MOV PSW, A). Кроме того, поскольку символы R0..R7 могут обозначать две различные области РПД, то необходимо корректно управлять выбором текущего банка регистров (команды SEL MB0 и SEL MB1, а также MOV PSW, A).

3.12. Арифметические команды

В системе команд микроконтроллера имеются следующие арифметические команды, позволяющие выполнять операции над 8-битными целыми числами:

- ADD – сложение однобайтных чисел;
- ADDC – сложение с учетом флага переноса;
- INC – инкрементирование (увеличение на 1);
- DEC – декрементирование (уменьшение на 1);
- DA – двоично-десятичная коррекция.

Команды сложения ADD, ADDC и DA формируют два флага: C (перенос из старшего разряда) и AC (полуперенос, т.е. перенос из младшей тетрады в старшую). Эти флаги используются для выполнения условных переходов, двоично-десятичной коррекции, а также для сложения многобайтных чисел.

Заметим, что команда вычитания отсутствует, поэтому соответствующая операция выполняется программно (при помощи инверсии и сложения). При этом используются два способа, основанные на применении дополнительного кода

$$a - b = a + ДК(b) = a + (\overline{b} + 1),$$

либо обратного кода:

$$a - b = \overline{(\overline{a} + b)}.$$

Например:

$$2\text{E}_{16} - 04_{16} = 2\text{E}_{16} + ДК(04_{16}) = 2\text{E}_{16} + (\overline{04_{16}} + 1) = 2\text{E}_{16} + (\text{FB}_{16} + 1) = 2\text{A}_{16},$$

$$2\text{E}_{16} - 04_{16} = \overline{(\overline{2\text{E}_{16}} + 04_{16})} = \overline{(\text{D1}_{16} + 04_{16})} = \overline{\text{D5}_{16}} = 2\text{A}_{16}.$$

Более сложные операции, такие как умножение или деление, также могут быть реализованы программным способом.

Напомним, что команды сложения ADD и ADDC выполняют операции сложения над однобайтными двоичными числами, которые могут интерпретироваться двумя способами:

- 8-разрядное число без знака (unsigned) с диапазоном $0..255_{10}$;
- 8-разрядное число со знаком (signed) с диапазоном $-128_{10}..+127_{10}$, в котором старший разряд является знаковым (“1”означает минус, “0” – плюс).

Например, число FE_{16} может рассматриваться как 254_{10} либо как -2_{10}. При этом операции выполняются по законам двоичной арифметики.

Для сложения по законам десятичной арифметики используется последовательность команд

```
ADD A, …        ; двоичное сложение операндов
DA A            ; десятичная коррекция результата
```

При этом слагаемые представляются в двоично-десятичном коде, при котором каждой цифре десятичного числа выделяется отдельная тетрада, например $25_{10} = 0010'0101_{2/10}$ (следует учитывать, что в традиционной двоичной кодировке это число соответствует 37_{10}). В команде DA A реализован следующий алгоритм:

1. если после сложения значение младшей тетрады больше 9 или установился флаг AC, то к содержимому аккумулятора прибавляется 6;
2. если после этого значение старшей тетрады больше 9 или установлен флаг C, то к старшей тетраде прибавляется 6.

Например:

$$25_{10} + 09_{10} \rightarrow (25_{16} + 09_{16}) + \text{коррекция}$$

ADD: $25_{16} + 09_{16} = 2\text{E}_{16}$

DA A: $2\text{E}_{16} + 06_{16} = 34_{16} \rightarrow 34_{10}$

При вычитании по законам двоично-десятичной арифметики используется сложение с дополнительным кодом (дополнение до 100_{10}), которое осуществляется следующим образом:

$$a - b = a + ДК(b) = a + (100_{10} - b) \rightarrow a + (99_{16} - b) + 1,$$

где двоичное вычитание осуществляется с помощью обратного кода:

$$99_{16} - b = \overline{\overline{99} + b} = \overline{66 + b}.$$

При использовании арифметических операций следует учитывать, что команды INC и DEC не изменяют флаги C и AC.

3.13. Логические команды

Система команд микроконтроллеров MCS-48 позволяет выполнять следующие логические операции над байтами:

- ANL – логическое “И” аккумулятора с операндом;
- ORL – логическое “ИЛИ” аккумулятора с операндом;
- XRL – логическое “Исключающее ИЛИ” аккумулятора с операндом;
- CLR A – очистка содержимого аккумулятора (обнуление);
- CPL A – инвертирование содержимого аккумулятора.

Эти операции выполняются побитно, а результат помещается в аккумулятор. Команды ANL, ORL, XRL и CPL A не воздействуют на флаги. Команда CLR A очищает флаг C.

Существуют также логические команды (ANL, ORL), результат выполнения которых помещается в буфер порта ввода/вывода BUS, P1, P2. Аналогичные команды (ANLD, ORLD) имеются для четырёхразрядных портов P4..P7. Это позволяет эффективно управлять значениями отдельных бит при вводе/выводе информации.

```
ORL P1, #00001000b      ; установить третий бит порта P1;
ANL P2, #10111111b      ; сбросить шестой бит порта P2.
```

Кроме того, имеются команды, позволяющие выполнять операции над отдельными битами:

- CLR C; CLR F0; CLR F1 – сброс флагов;
- CPL C; CPL F0; CPL F1 – инвертирование флагов.

Для установки флагов в “1” используется последовательность команд CLR, CPL.

К группе логических команд относятся и команды циклического сдвига содержимого аккумулятора RL, RR, RLC, RRC (рис. 13). На основе команд RRC, RLC можно реализовать арифметический сдвиг, т.е. умножение и деление на 2 (при умножении перед сдвигом во флаг C необходимо занести 0, а при делении – скопировать в него знаковый разряд аккумулятора).

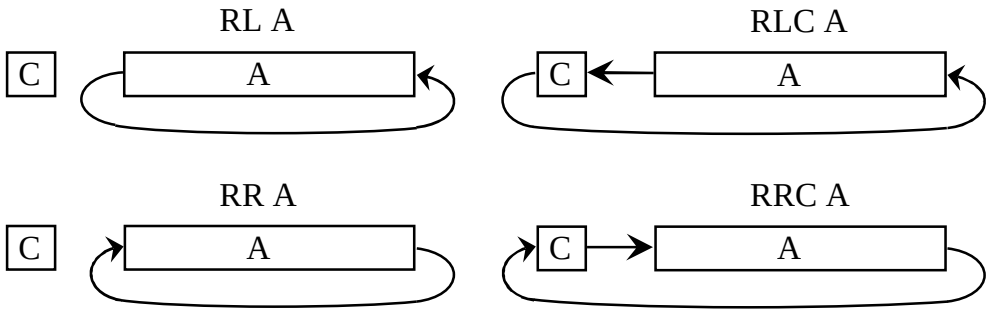


Рис. 13. Выполнение команд циклического сдвига

К этой группе команд относится и обмен местами старшей и младшей тетрад внутри аккумулятора (SWAP A), которая необходима при вводе/выводе тетрад в порты P4..P7.

3.14. Команды передачи управления

Команды передачи управления позволяют изменить естественный ход выполнения программы. Обычно при выполнении программы содержимое счётчика команд последовательно увеличивается на 1 или 2 (для однобайтных и двухбайтных команд соответственно). А команды передачи управления изменяют эту последовательность и приводят к записи в счётчик команд нового значения.

В микроконтроллерах семейства MCS-48 имеются две команды безусловного перехода:

- JMP ad11 – безусловный переход (с 11-разрядным адресом);
- JMPP @A – безусловный переход в текущей странице ПП

и 20 команд условного перехода (с 8-разрядным адресом):

- JZ, JNZ – переход по нулевому/ненулевому содержимому аккумулятора;
- JC, JNC – переход по установленному/сброшенному флагу C;
- JB0...JB7 – переход по единичному значению бита аккумулятора;
- JF0, JF1 – переход по единичному значению флагов F0 и F1;
- JTF – переход по переполнению таймера (установке флага TF);
- JT0, JNT0 – переход по высокому/низкому уровню на входе T0;
- JT1, JNT1 – переход по высокому/низкому уровню на входе T1;
- JNI – переход по низкому уровню на входе прерывания INT.

Команда безусловного перехода JMP задаёт 11-разрядный адрес ad11. Однако для полной адресации памяти программ объёмом 4 Кбайт требуется 12 разрядов. Поэтому она позволяют осуществить переход в пределах банка MB0 или MB1 объёмом по 2 Кбайт. Номер банка определяется флагом DBF, значение которого копируется в старший бит счётчика команд (PC_{11}). А сам флаг DBF управляется командами SEL MB0 и SEL MB1. Такая логика формирования адреса

перехода приводит к тому, что для переключения банков памяти программ требуется выполнить две команды:

```
SEL MB1          ; изменить номер банка во флаге DBF;
JMP ad11         ; перейти по адресу ad11 в банке MB1.
```

При этом собственно переключение банка ($PC_{11} = 1$) происходит по команде JMP, а команда SEL используется для подготовки переключения.

Остальные команды ветвления содержат только 8-разрядный адрес, который копируется в младшие разряды счётчика команд $PC_0..PC_7$. При этом его старшие разряды $PC_8..PC_{11}$ не изменяются. Поэтому переход осуществляется только в пределах одной страницы объёмом 256 байт. Это порождает проблему перехода с одной страницы на другую, которая решается путём комбинирования команд условного перехода и команды безусловного перехода JMP, например вместо команды JC ad8 выполняется последовательность команд:

Переход в пределах одной страницы	*Переход на другую страницу*
JC M1 . . . . . . M1:	JNC M0 JMP M1 M0: . . . . M1: . . . . . .

Если команда перехода с 8-разрядным адресом расположена на границе двух страниц (т.е. первый байт команды на одной странице, а второй – на следующей), то переход будет выполнен в пределах той страницы, где располагается второй байт команды.

В отличие от классических процессоров микроконтроллеры семейства MCS-48 не имеют команд перехода по положительному и отрицательному результату. Однако имеются команды JB0…JB7, позволяющие выполнить анализ любого бита аккумулятора. Поэтому для условного перехода по отрицательному результату используется команда JB7, а для перехода по положительному результату предварительно инвертируется содержимое аккумулятора.

Команды JC и JNC совместно с командой сложения ADD позволяют реализовать переходы по условиям “больше/меньше” некоторой пороговой величины. Например, для перехода по условию A > 05 используются команды:

```
ADD A, #0FBh     ; сложить аккумулятор с числом –6;
JC M0            ; если до сложения A > 05, то перейти на M1.
```

Но следует учитывать, что при этом изменяется содержимое аккумулятора.

Команда безусловного перехода с косвенной адресацией JMPP @A позволяет реализовать *множественные ветвления*, аналогично операторам SWITCH или CASE языков высокого уровня. При этом в памяти программ

располагается таблица команд JMP, а в аккумулятор помещается номер ячейки таблицы, содержащей команду JMP с требуемым адресом перехода. Эта таблица должна располагаться в той же странице памяти программ, что и команда JMPP, а ячейки таблицы нумеруются числами 0, 2, 4, ...

Для организации *циклов* удобно использовать команду DJNZ, сочетающую в себе декремент и переход по ненулевому результату. При этом счётчик циклов организуется в одном из регистров:

```
        MOV R4, #0Ah      ; инициализация счётчика цикла
M0:     . . . . . .
        . . . . . .       ; тело цикла
        . . . . . .
        DJNZ R4, M0       ; декремент R4 и повторение цикла, если R4≠0
```

Следует отметить, что тело цикла от метки M0 до команды DJNZ включительно должны находиться в пределах одной страницы памяти программ.

Сложные алгоритмы управления можно реализовать, применяя структурный стиль программирования, при котором задача разбивается на мелкие подзадачи, каждая из которых решается своей процедурой или функцией. Для этого используются следующие команды:

- CALL – вызов подпрограммы (с 11-разрядным адресом);
- RET – возврат из подпрограммы (без восстановления PSW);
- RETR – возврат из прерывания (с восстановлением PSW).

При вызове подпрограммы выполняются следующие действия:

– содержимое счётчика команд (адрес возврата, 12 разрядов) и старшей тетрады слова состояния (флаги, 4 разряда) сохраняются в стеке;

– в счётчик команд записывается 12-разрядный адрес подпрограммы, старший бит которого копируется из флага DBF, а остальные 11 разрядов являются аргументом команды CALL.

Глубина вложений подпрограмм ограничена ёмкостью стека (16 байт) и не должна превышать восьми. Подпрограмма должна заканчиваться командой RET, по которой:

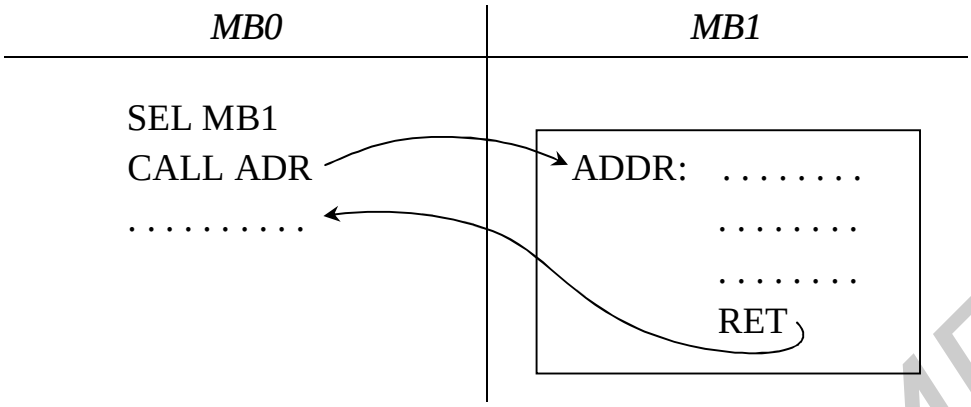
– восстанавливается из стека содержимое счётчика команд (адрес возврата);

– продолжается выполнение основной программы.

Заметим, что подпрограмма может заканчиваться командой возврата из прерывания RETR, которая кроме адреса возврата восстанавливает PSW.

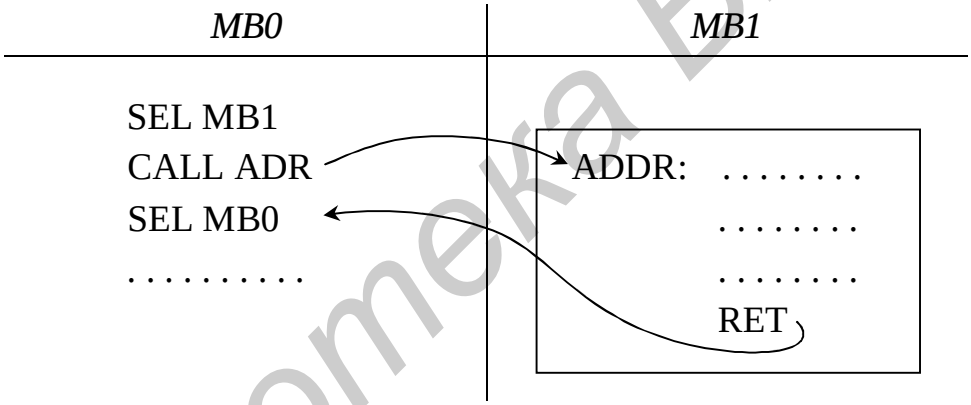
Подобно команде JMP, команда CALL позволяет обратиться в любое место текущего банка памяти программ. Кроме того, используя команды SEL MB0 и SEL MB1, можно вызвать подпрограммы из альтернативного (по отношению к текущему) банка. В этом случае перед вызовом подпрограммы необходимо

выбрать соответствующий банк памяти, а перед возвратом – восстановить старое значение DBF командой SEL:



Если в подпрограмме не восстановить значение номера банка во флаге DBF командой SEL MB0, то возврат все же будет выполнен правильно, в банк MB0 (так как в стеке сохранен полный 12-разрядный адрес возврата). Однако первая же команда JMP или CALL переключит банк на MB1.

Если к подпрограмме обращаться из разных банков памяти, то необходимо восстанавливать DBF в основной программе вслед за командой вызова:



Следует также учитывать, что подпрограммы обработки прерывания должны располагаться только в нулевом банке памяти MB0.

3.15. Команды управления режимами работы микроконтроллера

В эту группу входят команды управления таймером/счётчиком, прерываниями и флагами переключения банков регистров и банков памяти программ:

- STRT CNT – запуск счётчика;
- STRT T – запуск таймера;
- STOP TCNT – останов таймера/счётчика;
- EN I, DIS I – разрешение/запрещение внешних прерываний;
- EN TCNTI, DIS TCNTI – разрешение/запрещение прерывания от таймера/счётчика;
- SEL RB0, SEL RB1 – выбор текущего банка регистров;

- SEL MB0, SEL MB1 – выбор текущего банка памяти программ;
- ENT0 CLC – разрешение выдачи синхросигнала на выход T0.

Последняя из этих команд ENT0 используется для выдачи на вывод T0 синхроимпульсов с частотой $f_0 / 3$, где f_0 – частота кварцевого резонатора. Этот сигнал используется для синхронизации внешних устройств. Он может быть отключен только сбросом микроконтроллера.

Наличие команд переключения банков регистров позволяет при вызове подпрограмм и обработке прерываний эффективно использовать второй банк регистров в качестве рабочего, сохраняя параметры вычислительного процесса не в стеке, а в исходном банке регистров. При этом возврат к исходному банку регистров будет выполнен автоматически, если подпрограмма обработки прерывания оканчивается командой возврата RETR (с восстановлением PSW).

3.16. Загрузка прикладных программ в резидентную память

Процесс загрузки прикладных программ в резидентную память микроконтроллера (часто называемый программированием), требует использования специального устройства – программатора, обеспечивающего формирование требуемых сигналов напряжением +5 и +25 В, подаваемых на выводы микроконтроллера (рис. 14). При загрузке программ в РПП участвует большая группа выводов микроконтроллера, сигналы на которых должны формироваться в строго определенной последовательности. Эта временная последовательность приводится на рис. 15.

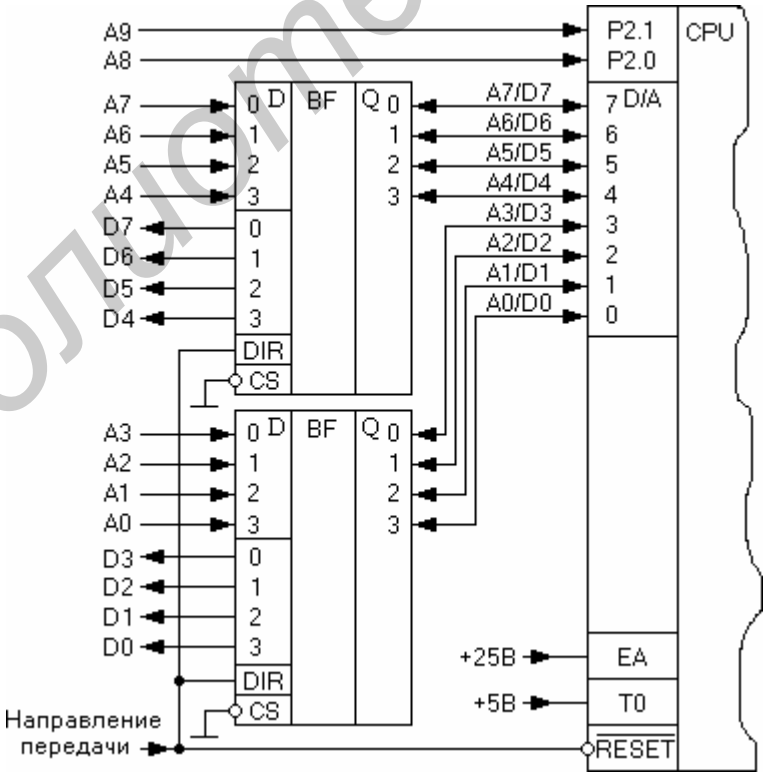


Рис. 14. Схема загрузки РПП с контролем вводимой информации

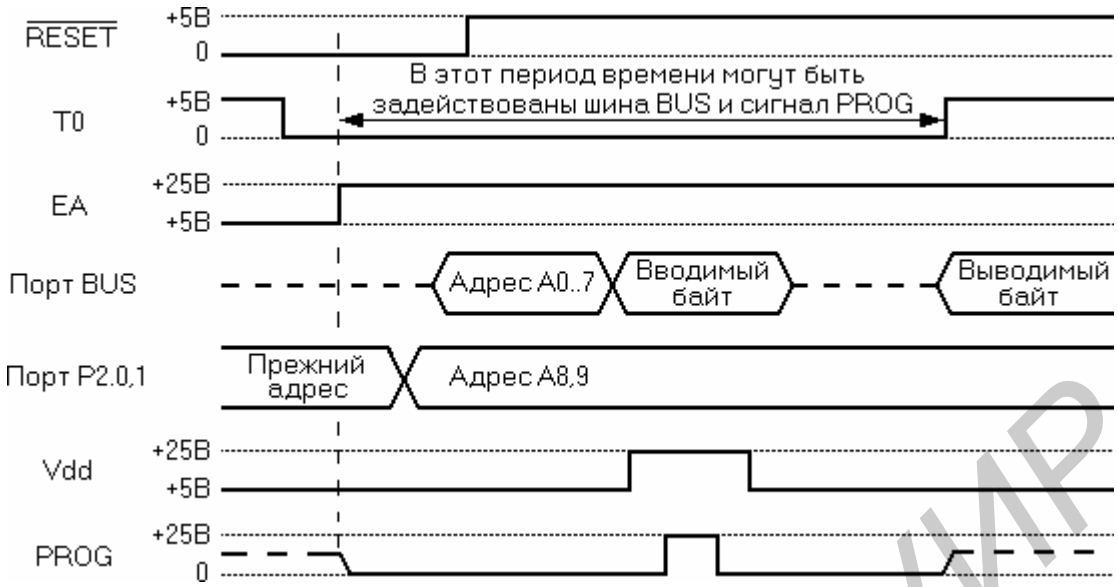


Рис. 15. Временная диаграмма режима загрузки РПП

Верификация прикладной программы, содержащейся в резидентной памяти МК, может быть выполнена путем прочтения её содержимого в режиме внешнего доступа. В процессе загрузки программы выполняется следующая последовательность действий:

1) программатор устанавливается в начальное состояние (U_{dd} = + 5 В, X1 синхросигнал с частотой 1..6 МГц, RESET = 0; T0 = +5 В, EA = +5 В);
2) микроконтроллер устанавливается в панельку программатора;
3) инициализируется режим загрузки (T0 = 0, EA = +25 В; PROG = 0);
4) выдаётся 10-разрядный адрес (8 разрядов на шину BUS и 2 разряда на линии P2.0, P2.1);
5) на вход RESET подаётся +5 В (защелка адреса);
6) выдаётся загружаемый байт (на шину BUS);
7) подаётся напряжение программирования (U_{dd}= + 25 В);
8) подаётся программирующий импульс (PROG = 25 В длительностью 50 мс);
7) снимается напряжение программирования (U_{dd}= + 5 В);
10) инициализируется режим проверки (T0 = + 5 В);
11) читается записанный байт (с шины BUS) и сравнивается с заданным;
12) производится переход к загрузке следующего байта (T0 = 0, RESET = 0) и процедура повторяется с п.4;
13) программатор возвращается в начальное состояние (T0 = +5 В);
14) микроконтроллер извлекается из панельки программатора.

При выборе программатора следует учитывать, что существуют как универсальные, так и специализированные устройства. Программатор может подключаться к компьютеру через стандартный порт ввода/вывода (параллельный либо последовательный), а также через специальный адаптер компьютера.

4. Семейство микроконтроллеров MCS-51

4.1. Архитектура микроконтроллеров MCS-51

Семейство микроконтроллеров MCS-51 основано на 8-разрядном процессорном ядре. В его состав входит целый ряд микросхем, отличающихся производительностью, объёмом встроенной памяти, количеством портов ввода/вывода, характеристиками таймеров и интерфейсных модулей. Все микроконтроллеры этого семейства работают с одной и той же системой команд, большинство из них выполняется в одинаковых корпусах с совпадающей цоколевкой. Это позволяет использовать для разработанного устройства микросхемы разных производителей без изменения принципиальной схемы и программы.

Первым представителем этого семейства является микросхема 8051 фирмы Intel, выпущенная в 1980 году на базе n-MOS технологии. В более поздних модификациях 80C51 использовалась технология CMOS, отличающаяся пониженным энергопотреблением. Отечественными аналогами семейства MCS-51 являются микросхемы 1816ВЕ31/51, 1830ВЕ31/51, 1834ВЕ31 и 1850ВЕ31. Типичный микроконтроллер этого семейства имеет 1 Кбайт внутренней памяти программ и 128 байт внутренней памяти данных (16 байт из которых допускают побитовую адресацию).

Первоначально при выполнении команд использовались простейшие методы синхронизации, при которых каждый машинный цикл состоял из 12 тактов. Поэтому при тактовой частоте 12 МГц команда выполнялась за 1..2 мкс. В дальнейшем число тактов на цикл удалось сократить до четырёх, двух, и даже одного, а тактовую частоту довести до 100 МГц.

На сегодняшний день существует более 200 модификаций микроконтроллеров семейства MCS-51, выпускаемых почти двадцатью компаниями (Philips, Siemens, Intel, Atmel, Dallas, Temic, Oki, AMD, MHS, Gold Star, Winbond, Silicon Systems и др., табл. 2). Эти модификации включают в себя кристаллы с широчайшим спектром периферии: от простых 20-выводных устройств с одним таймером и 1К программной памяти до сложнейших 100-выводных кристаллов с 10-разрядными АЦП, массивами таймеров-счетчиков, аппаратными 16-разрядными умножителями и 64 Кбайт программной памяти на кристалле. Основными направлениями их развития являются: повышение быстродействия, снижение напряжения питания и энергопотребления, увеличение объема внутренней памяти данных, использование внутренней FLASH-памяти программ, а также введение в состав периферии сложных устройств таких, как схемы управления электроприводами, CAN-интерфейсов распределённых систем управления, USB-интерфейсов и т.п.

Таблица 2

Микроконтроллеры, входящие в состав семейства MCS-51

Тип	ОЗУ, байт	ПЗУ, байт	Макс. частота, МГц	Таймеры-счётчики	Колич. прер.	Порты в/в
Зарубежные микроконтроллеры						
Atmel 89C1051	64	1K	24	1×16	3	2P
Atmel 89C2051	128	2K	24	2×16	6	1S, 2P
Intel 8051	128	4K	12	2×16	5	1S, 4P
Intel 8031	128	–	12	2×16	5	1S, 4P
Intel 8X52	256	8K	24	3×16	6	1S, 4P
Atmel 89C52	256	8K	24	3×16	8	1S, 4P
Atmel 89C55	256	20K	33	3×16	8	1S, 4P
Intel 8X54	256	16K	24	3×16	6	1S, 4P
Intel 8X58	256	32K	24	3×16	6	1S, 4P
Intel 8X51FA	128	8K	24	3+PCA	7	1S, 4P
Intel 8X51FB	256	16K	24	3+PCA	7	1S, 4P
Intel 8X51FC	256	32K	24	3+PCA	7	1S, 4P
Отечественные микроконтроллеры						
КР1816ВЕ51	128	4К	12	2×16	5	1S, 4P
КР1816ВЕ751	128	-	12	2×16	5	1S, 4P
КР1816ВЕ31	128	-	12	2×16	5	1S, 4P
КР1830ВЕ51	128	4К	16	2×16	5	1S, 4P
КР1830ВЕ751	128	-	16	2×16	5	1S, 4P
КР1830ВЕ31	128	-	16	2×16	5	1S, 4P

Основными структурными элементами базовой модели семейства микроконтроллеров MCS-51 являются (рис. 15):

- арифметико-логическое устройство;
- внутренняя память программ (4 Кбайт);
- внутренняя память данных (128 байт);
- два 16-разрядных таймера/счётчика;
- последовательный интерфейс (UART);
- блок обработки прерываний;
- четыре 8-разрядных порта ввода/вывода;
- тактовый генератор.

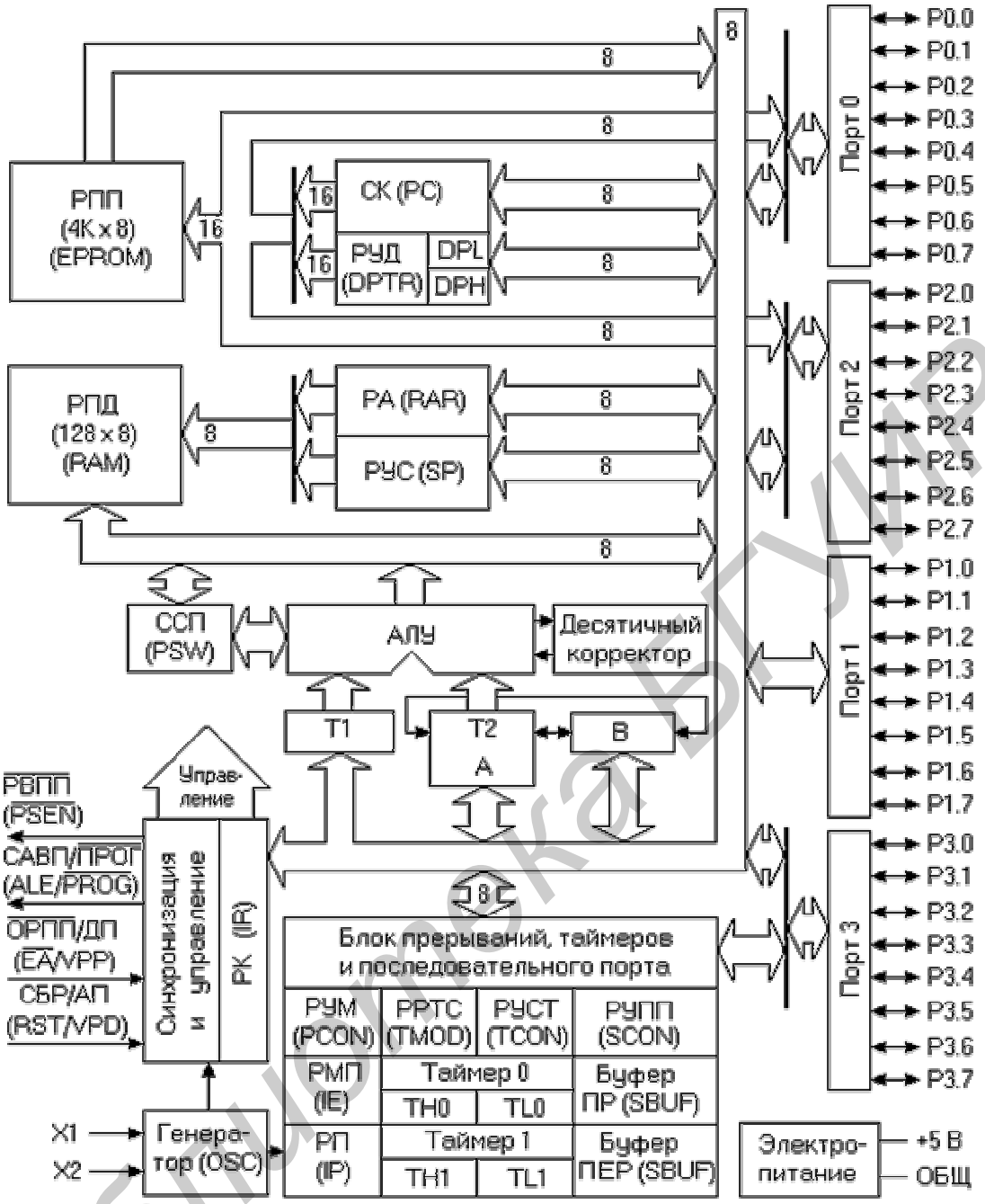


Рис. 16. Структурная схема микроконтроллера 1816ВЕ51

Конструктивно микроконтроллер выполнен в корпусе БИС с 40 внешними выводами, электрически совместимыми с элементами ТТЛ. Он имеет 32 вывода для подключения внешних устройств (4 восьмиразрядных порта Р0..Р3), два вывода питания, два вывода для кварцевого резонатора и четыре вывода для управляющих сигналов. Порт Р3 может быть запрограммирован пользователем на выполнение специальных (альтернативных) функций обмена информацией с внешней средой: последовательный ввод/вывод, чтение/запись, ввод тестируемых сигналов и сигналов прерывания.

4.2. Арифметико-логическое устройство

Восьмиразрядное арифметико-логическое устройство может выполнять арифметические операции сложения, вычитания, умножения и деления; логические операции И, ИЛИ, исключающее ИЛИ, а также операции циклического сдвига, сброса, инвертирования и т.п. В нём имеются программно недоступные регистры Т1 и Т2, предназначенные для временного хранения операндов, схема десятичной коррекции и схема формирования признаков.

Для реализации сложных команд в АЛУ используется механизм каскадного выполнения простейших операций, которые автоматически образуют «тандемы». Так, например, при выполнении одной из команд условной передачи управления по результату сравнения трижды инкрементируется счётчик команд, дважды производится чтение из РПД, выполняется арифметическое сравнение двух переменных, формируется 16-битный адрес перехода и принимается решение об условном переходе. Все перечисленные операции выполняются в АЛУ за 2 мкс.

Арифметико-логическое устройство может оперировать четырьмя типами данных: однобитными, четырёхбитными (тетрадами), 8-битными (байтами) и 16-битными. При создании прикладных программ разработчику доступны любые ячейки памяти данных и памяти программ, аккумулятор А, расширитель аккумулятора В, восемь регистров общего назначения R0..R7, слово состояния программы PSW, указатель стека SP а также ряд регистров специальных функций, приведенных в табл. 3.

Важной особенностью АЛУ является его способность оперировать не только байтами, но и битами. Эти операции выполняются специальным модулем АЛУ – «*булевым процессором*», который использует в качестве аккумулятора флаг переноса С. Этот процессор имеет побитово-адресуемое ОЗУ объёмом 128 бит, а также позволяет обращаться к 128 битовым полям регистров специальных функций. Побитно-адресуемыми являются все порты ввода-вывода, каждая линия которых может рассматриваться как однобитовый порт. Отдельно программно-доступные биты могут быть установлены, сброшены, инвертированы, переданы, проверены и использованы в логических операциях.

4.3. Память микроконтроллеров семейства MCS-51

В семействе микроконтроллеров MCS-51 память физически и логически разделена на память программ и память данных. В этих микроконтроллерах имеется три адресных пространства (рис. 16): память программ (ПП) – 64 Кбайт, внешняя память данных (ВПД) – 64 Кбайт, а также пространство внутренней (резидентной) памяти данных (РПД) и регистров специальных функций (РСФ) – 256 байт.

Память программ (ROM или EPROM) имеет суммарное адресное пространство объёмом 64 Кбайт, из которых 4 Кбайт размещается на кристалле

микроконтроллера (РПП). Область её нижних адресов используется при сбросе и обработке прерываний:

- 0000h – адрес перехода по сигналу RESET;
- 0003h – адрес вектора внешнего прерывания INT0;
- 000Bh – адрес вектора прерывания от таймера T0;
- 0013h – адрес вектора внешнего прерывания INT1;
- 001Bh – адрес вектора прерывания от таймера T1;
- 0023h – адрес вектора прерывания от последовательного порта.

При необходимости память программ может быть расширена за счёт дополнительных микросхем памяти, подключаемых к портам P0, P2 с помощью управляющих сигналов PSEN и ALE. Основное назначение памяти программ – хранение команд, которые считываются автоматически. Кроме того, в памяти программ можно хранить константы, таблицы функций и т.д., доступ к которым осуществляется командами MOVC.

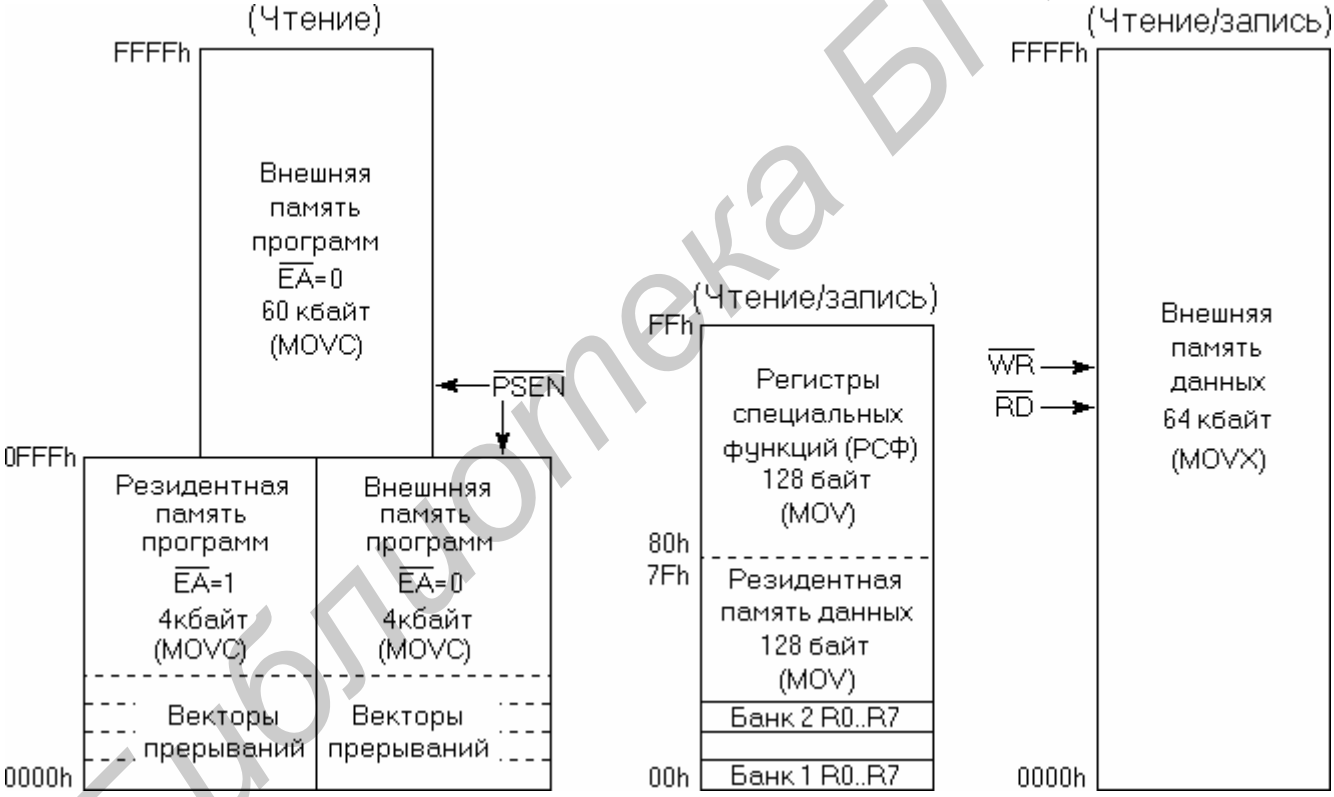


Рис. 17. Организация памяти микроконтроллеров MCS-51

Резидентная память данных (RAM) предназначена для хранения переменных в процессе выполнения прикладной программы. Она адресуется одним байтом и имеет ёмкость 128 байт. В ней размещается 4 банка рабочих регистров R0..R7, а также стек. Для банков регистров используются первые 32 байта. Регистры R0 и R1 в любом из банков могут использоваться в качестве регистров косвенного адреса @R0 и @R1. Следующие за регистровыми банками 16 байт образуют блок побитно-адресуемого пространства (128 бит с адресами от

00h до 7Fh). Все байты в нижней 128-байтной половине памяти могут адресоваться как прямо, так и косвенно. По умолчанию область стека начинается с адреса 0008h, так как при сбросе указатель стека SP = 0007h. Поэтому при использовании дополнительных банков регистров программист должен переопределить начальное значение SP.

К адресному пространству РПД примыкают адреса регистров специальных функций, которые перечислены в табл. 3. Они включают в себя регистры портов, таймеры, средства управления периферией и т.д. Часть из них допускает побитовую адресацию (128 бит с адресами 80h..FFh). Например, порт P0 можно адресовать двумя способами: как байт с адресом 80h, либо как набор битов P0.0..P0.7 с адресами 80h..87h. Карта адресов РПД и РСФ приведена на рис. 18. При обращении к РПД и РСФ используются команды MOV.

Таблица 3

Регистры специальных функций

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

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

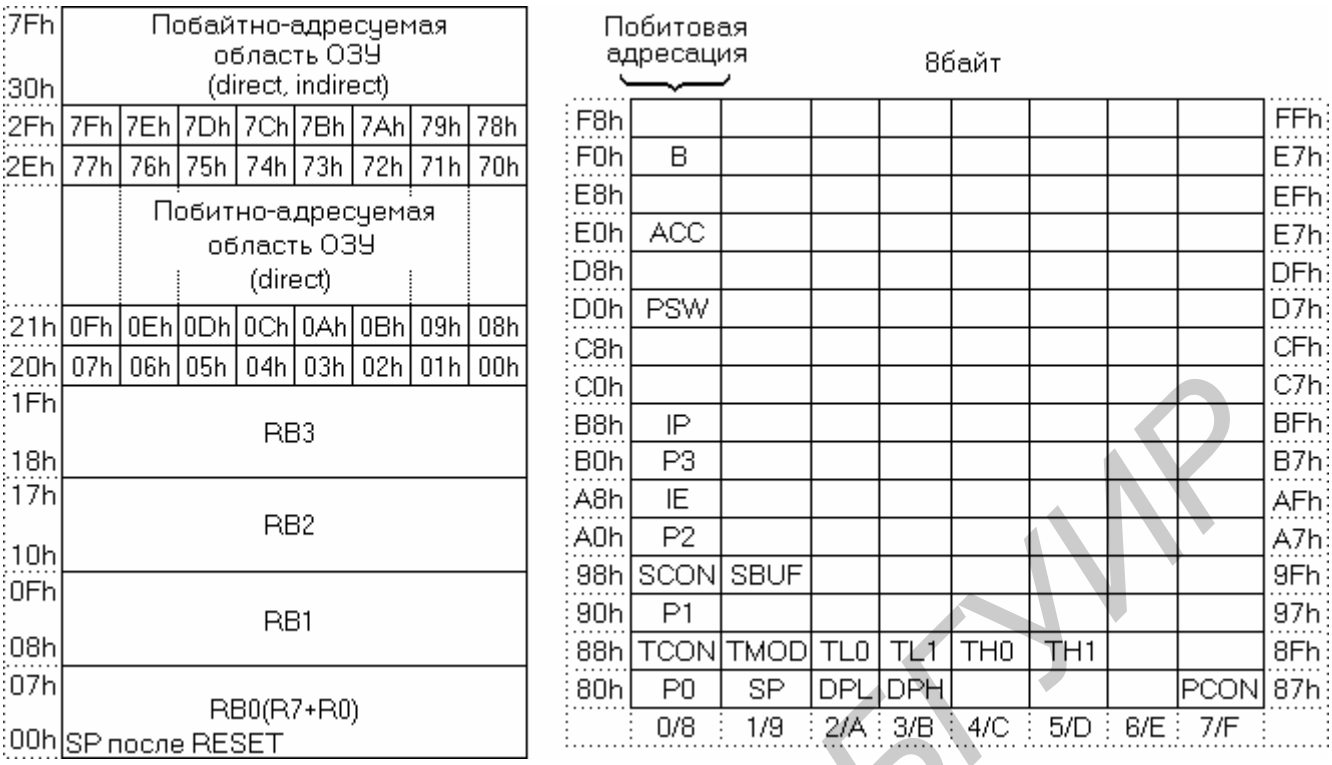


Рис. 18. Побайтовая и побитовая адресация РПД и РСФ

Следует заметить, что имеются варианты микроконтроллеров семейства MCS-51 с резидентной памятью данных объёмом 256 байт. В этом случае при обращении к области РСФ используется прямая адресация, а при обращении к верхней части области РПД – косвенная адресация.

Внешняя память данных (RAM) имеет независимое адресное пространство объёмом 64 Кбайт, при обращении к ней используются команды MOVX. Микросхемы внешней памяти данных подключаются к портам P0, P2 с помощью управляющих сигналов RD, WR и ALE. При необходимости адресное пространство ВПД можно совместить с адресным пространством ВПП, объединяя сигналы RD и PSE.

4.4. Параллельные порты ввода/вывода

Линии ввода/вывода микроконтроллеров семейства MCS-51 сгруппированы в четыре 8-разрядных порта P0..P3. Эти порты могут использоваться для функций обмена данными (допуская управление отдельными линиями), так и для выполнения альтернативных функций, таких как обращение к внешней памяти, приём запросов прерываний и др. Порт P0 является двунаправленным и аналогичен порту BUS микроконтроллеров MCS-48. Порты P1, P2, P3 являются квазидвунаправленными и также аналогичны портам P1, P2 семейства MCS-48. Чтобы вывести значение “0” или “1” на линию квазидвунаправленного порта, необходимо записать это значение в соответствующий разряд буферного регистра, а для использования этой линии в качестве входа её регистр должен содержать “1”.

При обращении к внешней памяти используются порты P0 и P2. Через порт P0 выдаётся младший байт адреса, который должен быть зафиксирован во внешнем регистре по сигналу ALE, так как затем на линиях P0 появляются данные для записи или чтения. Через порт P0 выдаётся старший байт адреса, внешняя фиксация которого не требуется.

Порт P1 предназначен для задания младшего байта адреса при программировании и проверке РПП микроконтроллера. При этом старший байт адреса передаётся через порт P2, а байт кода команды – через порт P0.

Линии порта P3 могут быть использованы для альтернативных функций, которые требуются в сложных системах управления:

- P3.7 – строб чтения из внешней памяти данных RD;
- P3.6 – строб записи во внешнюю память данных WR;
- P3.5 – вход тестируемого сигнала T1;
- P3.4 – вход тестируемого сигнала T0
- P3.3 – вход внешнего прерывания INT1;
- P3.2 – вход внешнего прерывания INT0;
- P3.1 – выход последовательного порта TxD;
- P3.0 – вход последовательного порта RxD.

Альтернативные функции могут быть задействованы путём записи “1” в соответствующие биты выходного буфера P3.0..P3.7.

4.5. Система прерываний MCS-51

Базовые модели микроконтроллеров 8051 имеют пять источников прерываний – два внешних (входы INT0, INT1) и три внутренних (от двух таймеров/счётчиков и один от последовательного порта). С появлением микроконтроллеров типа 8052 число источников прерываний увеличилось до 15 (в модели 8XC51GP). Адреса соответствующих векторов прерываний 0003h..0023h расположены в нижней области памяти программ (см. п. 4.3 и рис. 18). Поскольку интервал между адресами векторов прерываний составляет всего 8 байт, то по этим адресам располагается короткая процедура либо команда JMP, обеспечивающая переход на подпрограмму обслуживания прерывания (рис. 19).

Внешние прерывания INT0 и INT1 могут быть вызваны либо уровнем, либо переходом сигнала из “1” в “0” (в зависимости от значений управляющих бит IT0 и IT1 в регистре TCON). В обоих случаях прерывание устанавливает флаги IE0 и IE1 в регистре TCON, которые инициируют вызов соответствующей подпрограммы. Если прерывание вызвано уровнем, то для исключения повторных прерываний сигнал INT0 или INT1 должен стать пассивным до завершения подпрограммы обслуживания. Эта операция должна выполняться источником прерывания автоматически, в противном случае обработчик прерываний должен снять низкий уровень с линий INT через какую-либо линию портов. Кроме того,

перед возвратом из прерывания необходимо сбросить флаги IE0 и IE1. Во втором случае (прерывание по спаду импульса INT) сброс этих флагов выполняется аппаратно и дополнительных операций не требуется.

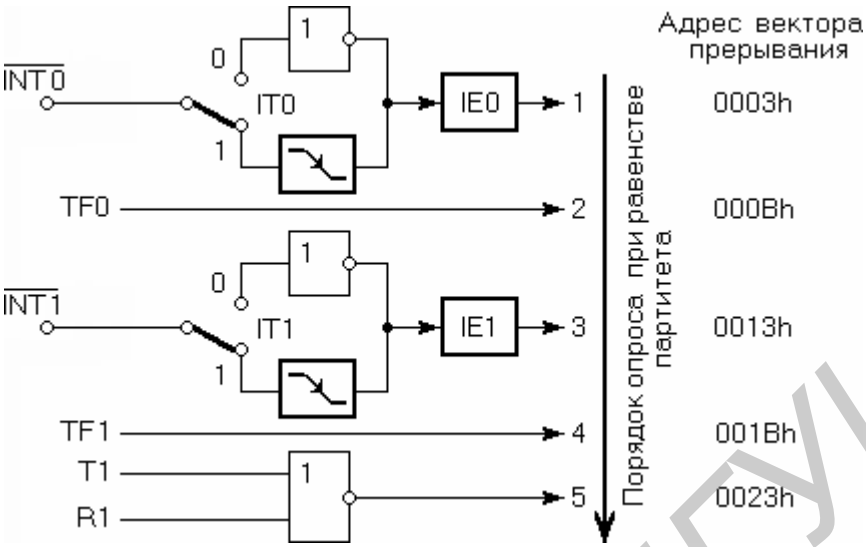


Рис. 19. Система прерываний MCS-51

Прерывания от таймеров инициируются при установке их флагов переполнения TF0 и TF1 в регистре TCON. Эти флаги сбрасываются автоматически при передаче управления подпрограмме обслуживания.

Прерывания от последовательного порта возникают в двух случаях: при загрузке байта во входной буфер (в режиме приёма) или освобождение выходного буфера (в режиме передачи). При этом аппаратно устанавливается один из флагов RI или TI регистра SCON, а конкретный источник определяется процедурой обслуживания посредством опроса. Сброс этих флагов выполняется программно.

Необходимо отметить, что любое из прерываний может быть инициировано программно, путём установки соответствующего флага: IE0, IE1, TF0, TF1, RI, TI. В результате будет вызвана та же подпрограмма, что и при установке этих флагов аппаратными средствами.

Каждый из источников прерываний может быть разрешен или запрещён при помощи соответствующего бита регистра маски прерываний IE, формат которого приведен на рис. 20. В этом регистре также имеется бит общего запрета всех прерываний EA.

При появлении запросов от нескольких источников очерёдность их обслуживания определяется двухступенчатым механизмом приоритетов. На первой ступени любому источнику можно присвоить «высокий» (“1”) или «низкий» (“0”) уровень приоритета. Для этого используются соответствующие биты регистра приоритетов IP (рис. 21). На второй ступени очерёдность обслуживания определяется адресом вектора прерывания, т.е. запросы с меньшим адресом обслуживаются в первую очередь. Если какое-либо прерывание уже обслуживается, то его процедуру может прервать только прерывание с более высоким приоритетом.

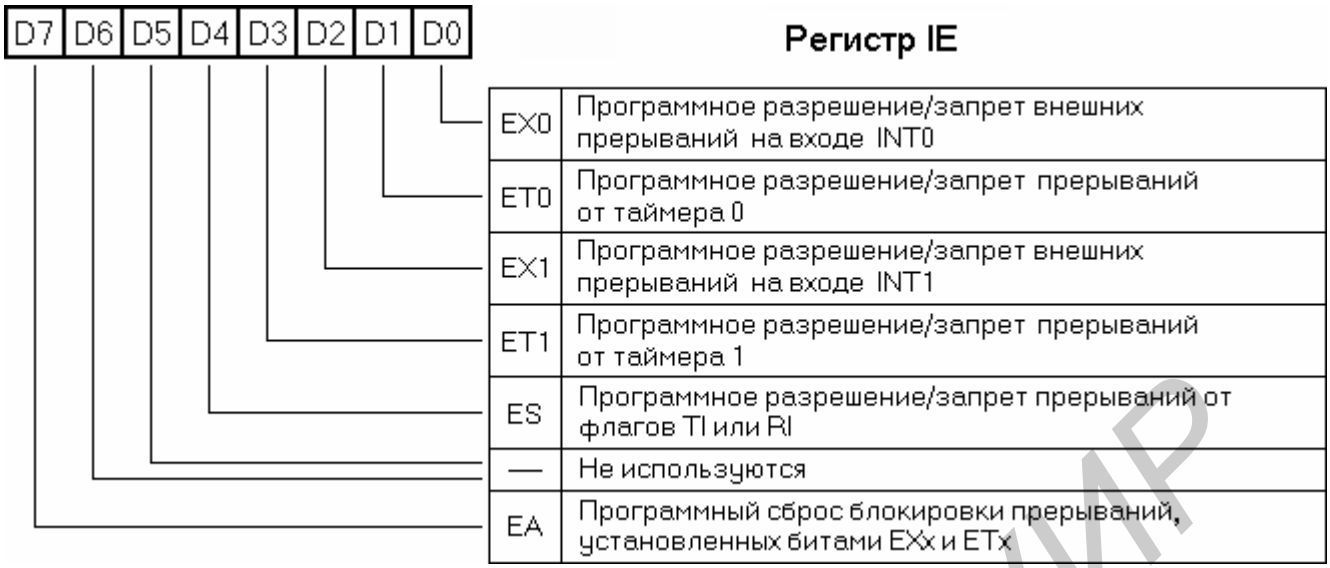


Рис. 20. Регистр маски прерываний IE

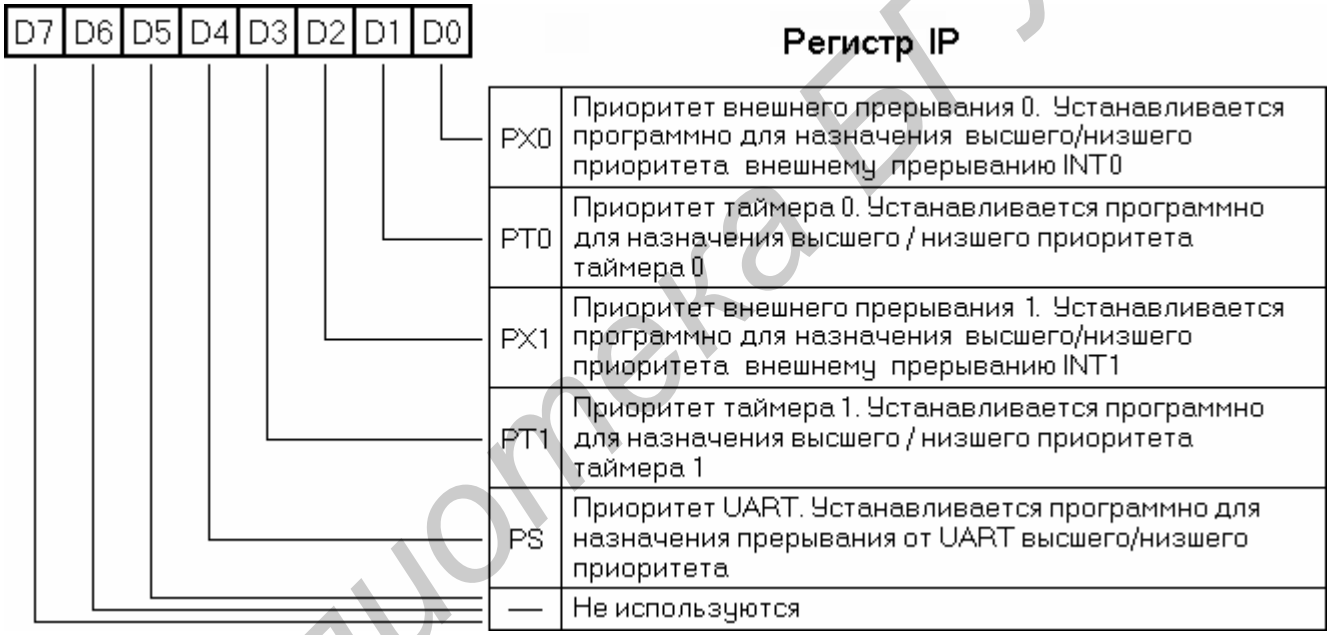


Рис. 21. Регистр приоритетов прерываний IP

Между запросом и началом обслуживания прерывания проходит не менее трёх машинных циклов. Обработка запроса может быть отложена, если:

- обслуживается запрос равного или более высокого уровня приоритета;
- текущий машинный цикл не является последним в цикле команды;
- выполняется команда RETI или любая команда, связанная с обращением к регистрам IE или IP.

Реакция на запрос, разрешённый к обслуживанию заключается в передаче управления от текущей программы специальной процедуре обслуживания прерывания. Для этого аппаратно формируется команда LCALL (заметим, что её код в программе не содержится). По команде LCALL текущее содержимое счётчика команд (адрес возврата) сохраняется в стеке. Затем в счётчик команд

загружается адрес вектора соответствующего источника прерывания, по которому располагается команда JMP. Сохранение в стеке других регистров микроконтроллера (A, B, PSW и т.д.) осуществляется программно командами PUSH в начале процедуры обслуживания. В конце этой процедуры должны выполняться соответствующие процедуры восстановления из стека POP. Подпрограммы обслуживания прерывания завершаются командой возврата RETI, по которой восстанавливается из стека прежнее содержимое счётчика команд. Заметим, что команда возврата из подпрограммы RET также восстанавливает содержимое счётчика команд, но не снимает блокировку прерываний.

4.6. Таймеры/счётчики

Микроконтроллеры MCS-51 имеют два 16-разрядных таймера/счётчика Timer0 и Timer1, каждый из которых независимо может быть запрограммирован на работу в качестве либо таймера либо счётчика. В режиме таймера производится подсчёт импульсов внутреннего тактового генератора, что позволяет формировать временные интервалы. В режиме счётчика производится подсчёт внешних импульсов на входах T0 и T1, которые соответствуют каким-либо событиям. В обоих случаях переполнение таймеров/счётчиков приводит к формированию запроса прерывания.

При работе в режиме таймера его содержимое инкрементируется в каждом машинном цикле, т.е. через каждые 12 периодов тактовой частоты. При работе в качестве счётчика инкрементирование осуществляется при переходе из “1” в “0” внешнего входного сигнала на входах T0, T1. Так как на распознавание перехода требуется два машинных цикла, то максимальная частота подсчёта входных сигналов равна 1/24 тактовой частоты. Для надёжной работы значение “1” на входах T0, T1 должно удерживаться не менее одного машинного цикла.

При использовании таймеров программисту доступны старшие байты счётчиков TH0, TH1, их младшие байты TL0, TL1, а также их флаги переполнения TF0, TF1. Для управления режимами работы и для организации взаимодействия таймеров с системой прерывания используются два регистра специальных функций: TMOD (регистр режима таймеров) и TCON (регистр управления таймерами). Формат этих регистров и назначение отдельных разрядов приведены на рис. 22 и 23. Таймеры/счётчики могут работать в четырёх режимах, причём режимы 0, 1 и 2 одинаковы обоих таймеров, а режим 3 различается.

В *режиме* 0 таймер/счётчик имеет разрядность 13, причём старший байт THx используется полностью, а из младшего байта TLx используются только 5 разрядов. Этот режим аналогичен таймеру микроконтроллеров MCS-48, где имеется 8-разрядный счётчик с 5-разрядным предделителем (рис. 24), однако частота переполнения таймера в MCS-51 примерно равна 8 мс.

Режим 1 аналогичен режиму 0, но используются все 16 разрядов регистров THx и TLx (рис. 25). Поэтому максимальное значение счётчика равно 65535, а максимальный период переполнения таймера – около 65 мс.

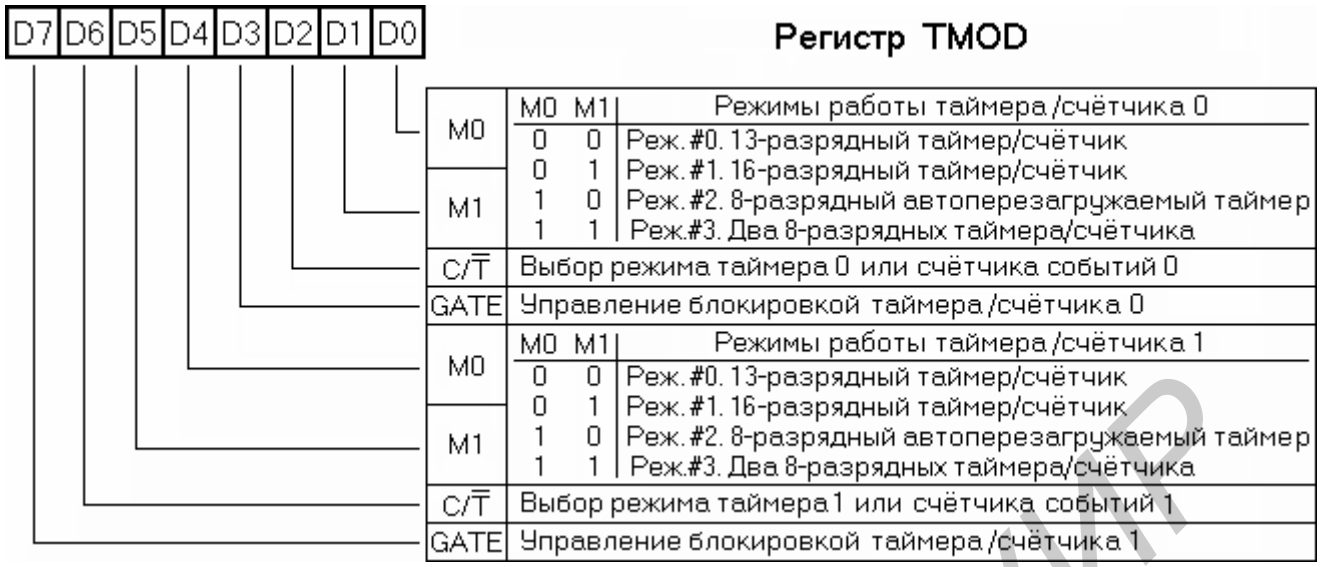


Рис. 22. Регистр режима таймеров/счётчиков TMOD

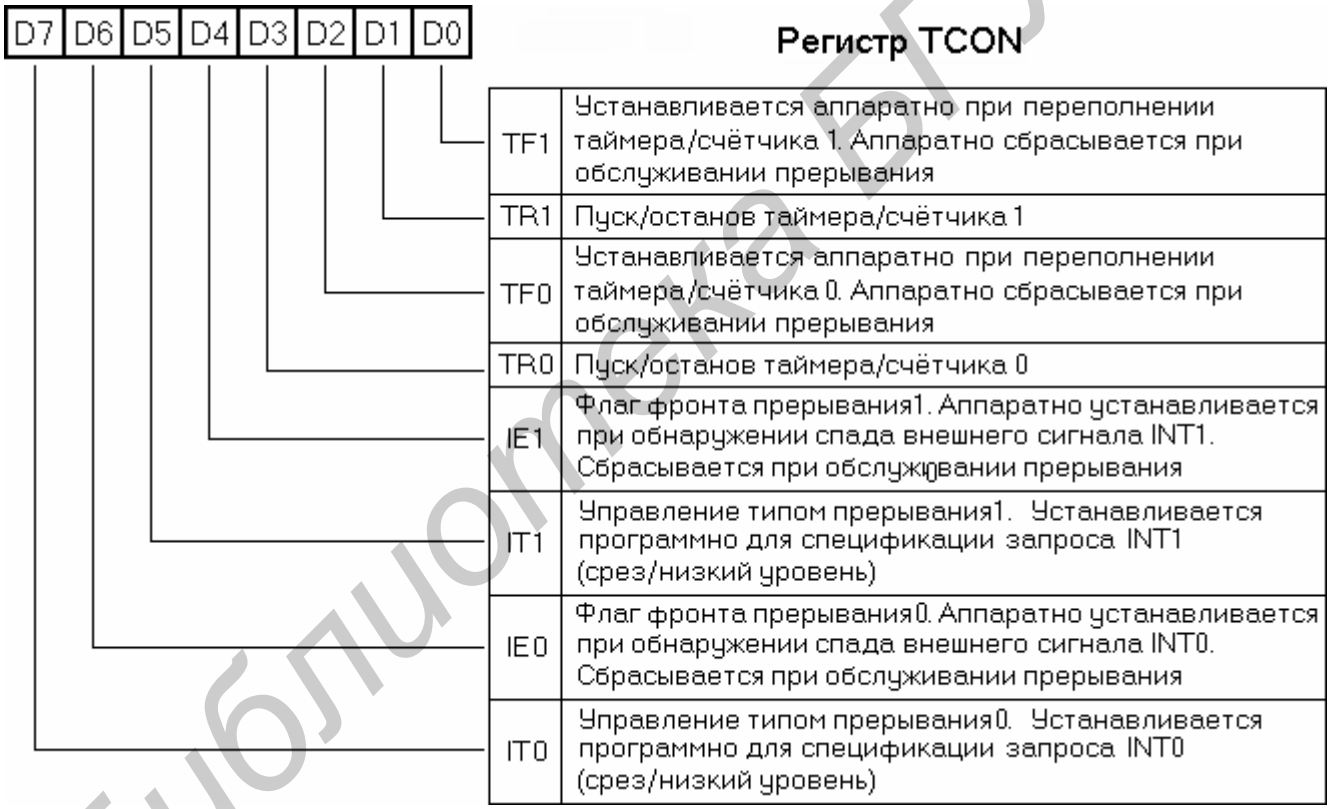


Рис. 23. Регистр управления таймеров/счётчиков TCON

В *режиме* 2 регистр TLx работает как 8-разрядный счётчик (рис. 26). После переполнения и установки флага TFx этот счётчик автоматически перезагружается значением из регистра THx, которое предварительно было задано программно. Перезагрузка не изменяет значение THx. Такая организация позволяет легко реализовать деление частоты на любой коэффициент в диапазоне от 1 до 256.

В *режиме* 3 таймеры работают по разному: Timer1 останавливается (эквивалентно TR1 = 0), а Timer0 разделяется на два независимых 8-разрядных счётчика TH0 и TL0 (рис. 27). При этом биты управления нулевого таймера/счётчика используются для TL0, а биты управления первого таймера/счётчика – для TH0, который может выполнять только функции таймера. Такой режим используется, если в системе управления требуется три таймера/счётчика.

В семействе MCS-51 отсутствуют специальные команды запуска и останова таймеров/счётчиков. Вместо этого используется биты TR0, TR1 (“1” – запуск, “0” – останов). При этом бит GATEx (блокировка) должен быть сброшен. Если GATEx = 1, то запуском таймеров/счётчиков управляет внешний вывод запроса прерывания INTx. Это позволяет аппаратно измерять длительность импульсов на входах прерываний.

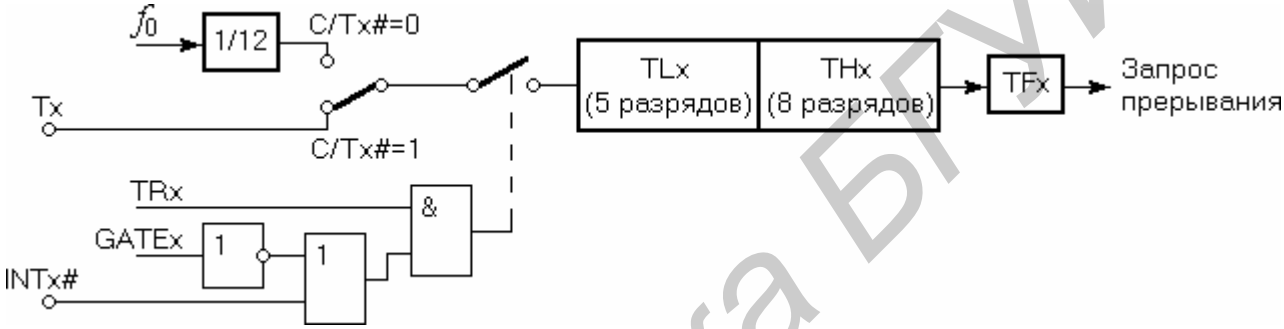


Рис. 24. Таймер/счётчик в режиме 0

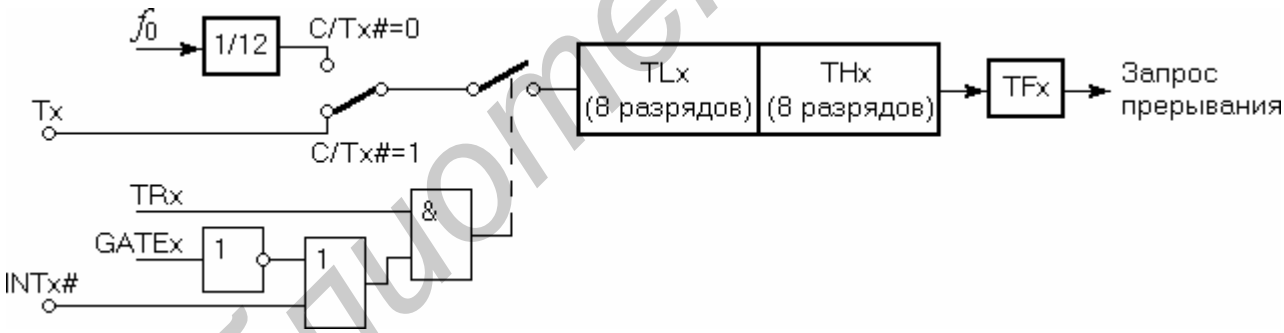


Рис. 25. Таймер/счётчик в режиме 1

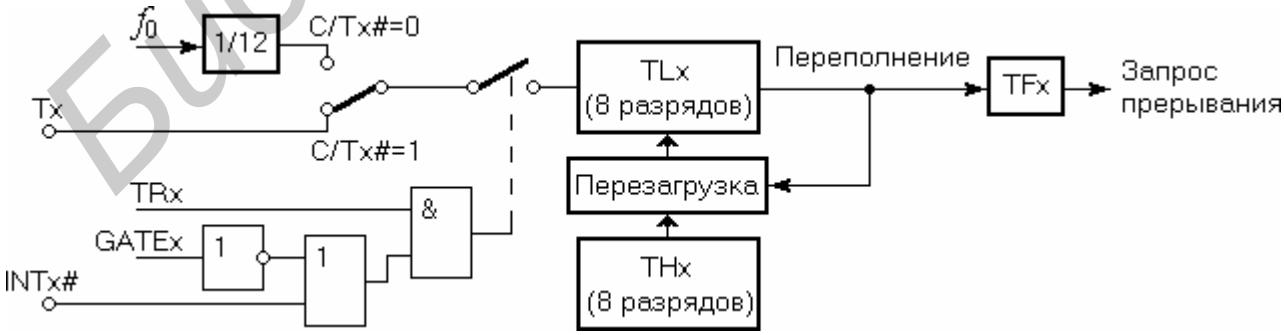


Рис. 26. Таймер/счётчик в режиме 2

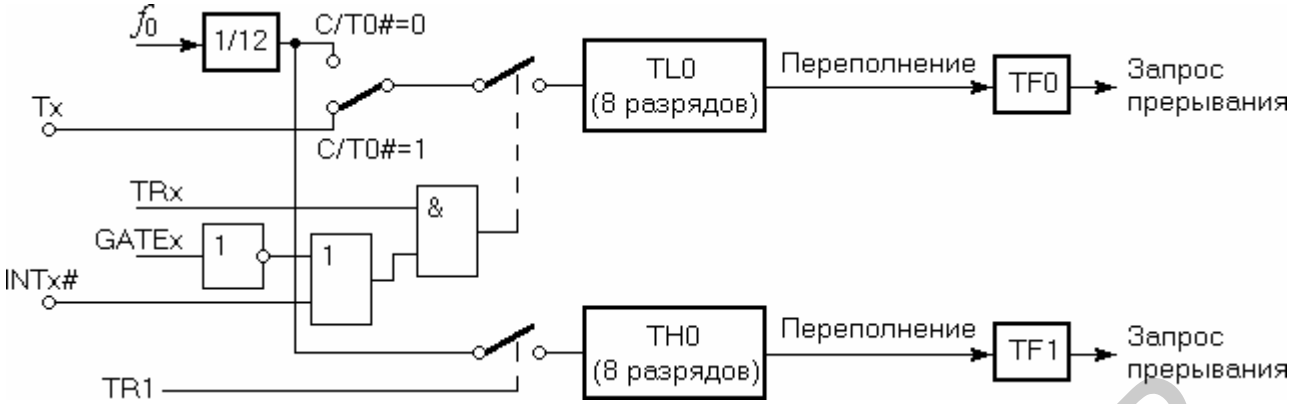


Рис. 27. Таймер/счётчик в режиме 3

На базе двух таймеров можно также измерить частоту внешних импульсов, подаваемых на вход T0. Для этого Timer0 переводится в режим счётчика и включается на определённый интервал времени (например, 1 мс), который формирует Timer1. В результате в регистры TH0, TL0 записывается частота измеряемого сигнала в килогерцах.

4.7. Последовательный интерфейс

Последовательный порт микроконтроллеров MCS-51, называемый также UART (Universal Asynchronous Receiver/Transmitter), является дуплексным и обеспечивает работу в синхронном либо асинхронном режимах. Его можно также использовать для обмена данными в системах управления на базе нескольких микроконтроллеров. Этот интерфейс поддерживает протокол обмена RS-232C, используемый в большинстве промышленных систем управления.

В состав UART входят регистры сдвига приёмника и передатчика, а также буферный регистр SBUF. Для управления последовательным портом используется регистр SCON (рис. 28), а также старший бит регистра управления энергопотреблением PCON.7, который позволяет задать удвоенную скорость передачи. Они позволяют установить режим работы (0 – синхронный; 1, 2, 3 – асинхронный), а также задать скорость передачи. В обоих случаях данные передаются начиная с младшего бита.

В режиме *синхронного* обмена (режим 0) по линии TxD выдаётся восемь синхроимпульсов, а по линии RxD принимается или передаётся байт данных. Поэтому линии TxD и RxD одного контроллера соединяются с соответствующими линиями другого контроллера (рис. 29). Передача очередного бита осуществляется каждый машинный цикл, поэтому обмен производится с фиксированной частотой $f_0 / 12$. Это максимально возможная скорость обмена через последовательный порт микроконтроллера.

При *асинхронном* обмене данные передаются по линии TxD, а принимаются по линии RxD. Поэтому линия TxD одного контроллера подключается к линии RxD другого, и наоборот (перекрёстное соединение). В этом случае используется старт/стопный принцип, при котором каждый передаваемый байт дополняется

стартовым и стоповыми битами, что позволяет начать передачу в любой момент времени. Приём и передача могут осуществляться одновременно, поскольку регистр SBUF состоит из двух частей – буфера приёмника и буфера передатчика. В режиме ожидания на линиях TxD удерживается высокий уровень, поэтому переход из “1” в “0” воспринимается как начало передачи старт-бита.

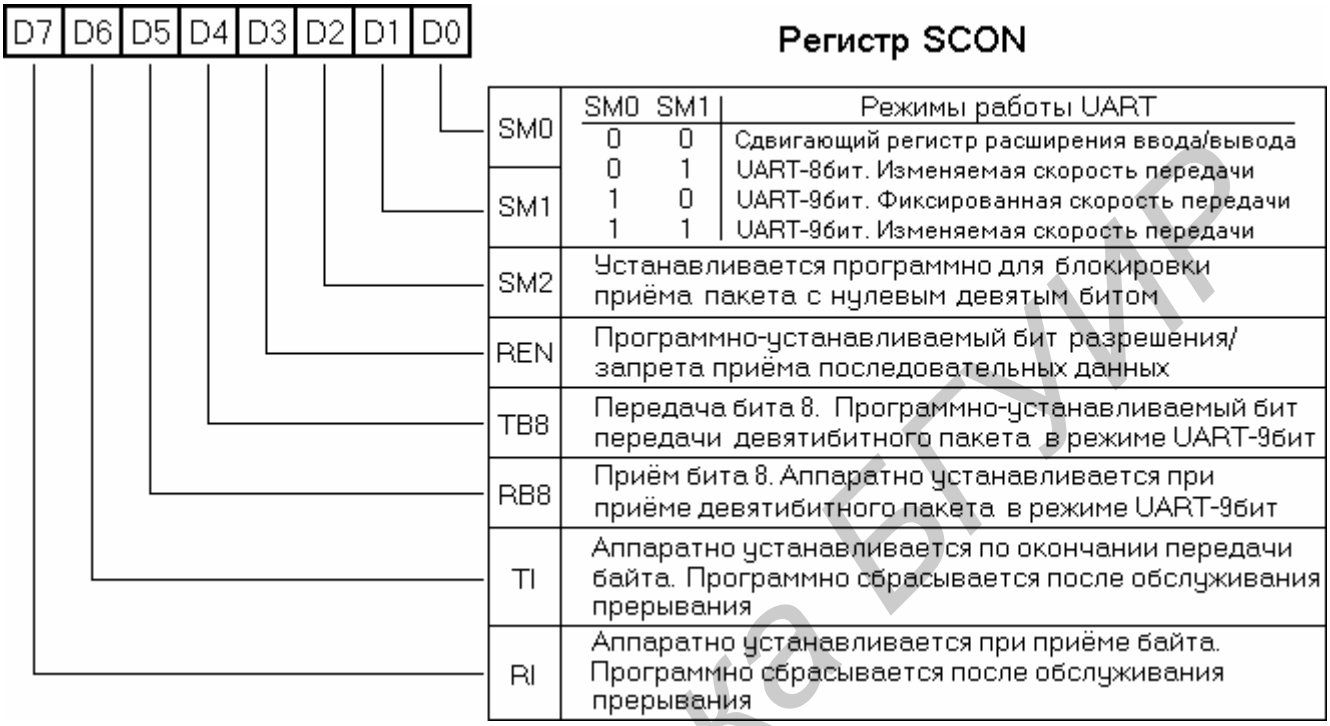


Рис. 28. Регистр управления режимами последовательного порта SCON

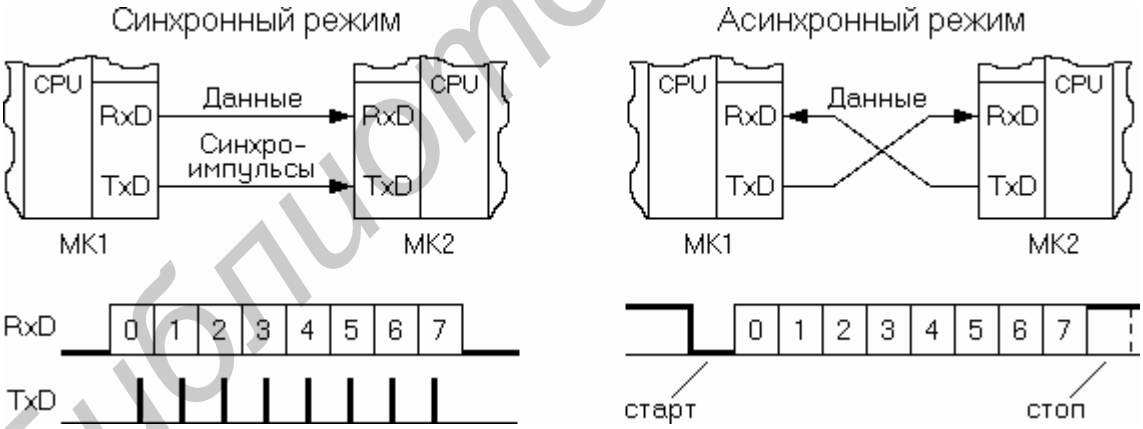


Рис. 29. Соединение микроконтроллеров при последовательной передаче данных

Чтобы передать данные необходимо записать байт в регистр SBUF, а для приёма информации необходимо прочитать содержимое этого регистра. После загрузки SBUF передаваемый байт копируется в регистр сдвига передатчика и побитно выдвигается на линию TxD. При приёме осуществляется обратная операция. Наличие буферного регистра приемника позволяет совмещать операцию чтения ранее принятого байта с приемом очередного байта. Если к

моменту окончания приема байта предыдущий байт не был считан из SBUF, то он будет потерян.

Последовательный интерфейс может работать в четырех различных режимах.

В *режиме* 0 осуществляется передача данных в синхронном режиме. Данные передаются и принимаются по линиям RxD, а по линиям TxD – синхросигналы. Передача начинается любой командой, по которой в SBUF поступает байт данных. Затем он копируется в сдвигающий регистр передатчика и в каждом машинном цикле сдвигается вправо (младшими битами вперёд), после чего очередной бит поступает на вывод RxD. В освобождающиеся старшие биты сдвигающего регистра передатчика записываются нули. При обнулении этого регистра устанавливается флаг TI (конец передачи) и линия RxD переводится в высокоуровневое состояние. При приёме этого байта в микроконтроллере должен быть сброшен флаг RI и установлен бит REN (разрешение приёма). После окончания приёма всех восьми бит устанавливается флаг RI (конец приёма), после чего принятый байт может быть прочитан из регистра SBUF.

В *режиме* 1 осуществляется асинхронная передача данных. При этом пакет данных состоит из 10 бит (рис. 29): старт-бит (“0”), 8 бит данных и стоп-бит (“1”). При приёме в разряд RB8 регистра SCON записывается стоп-бит. Для передатчика в регистре SCON необходимо сбросить бит REN, а для приёмника – установить его.

Передача инициируется загрузкой SBUF передатчика. Приём начинается при обнаружении перехода сигнала на входе RxD приёмника из состояния 1 в состояние 0, причём вход RxD опрашивается 16 раз за время передачи старт-бита. Если при этом обнаруживается значение, не равное 0, то приемник вновь возвращается к поиску перехода из 1 в 0. Такой механизм обеспечивает подавление ложных (сбойных) старт-бит.

Далее, при приёме каждого из информационных битов и стоп-бита, вход RxD опрашивается трижды (в состояниях 7, 8 и 9 внутреннего счётчика с коэффициентом деления 16). Это позволяет настроить приёмник на середину периода передачи бита и повысить помехоустойчивость за счет использования мажоритарного принципа (принятым значением считается то, которое было получено по меньшей мере дважды из трех раз). Если последний принятый бит равен 1, то приёмник устанавливает флаг RI, в противном случае принятый байт теряется и вновь начинается поиск переход из 1 в 0 на входе RxD. Потеря принятого байта также происходит, если из регистра SBUF не был прочитан предыдущий байт, т.е. не был сброшен флаг RI. Это должно выполняться программно.

В *режимах* 2 и 3 также осуществляется асинхронный обмен, но пакет данных состоит из 11 бит: старт-бит (“0”), 8 бит данных, программируемый девятый бит и стоп-бит (“1”). При передаче девятый бит записывается в разряд TB8 регистра SCON (перед загрузкой регистра SBUF инициирующего передачу).

При приёме девятый бит может быть прочитан из разряда TB8 регистра SCON. Программист может использовать девятый бит по своему усмотрению, например для контроля по чётности. Алгоритмы работы приёмника и передатчика в этих режимах аналогичны режиму 1, но имеются некоторые отличия в условиях окончания цикла приёма. Приём считается успешным, если принятый стоп-бит равен “1” и из регистра SBUF был прочитан предыдущий байт (сброшен флаг RI). Кроме того, если установлен разряд SM2 регистра SBUF, то независимо от состояния флага RI принимается и загружается в SBUF байт данных, который сопровождается “1” в девятом бите. Если при этом предыдущий байт не прочитан, то он теряется. Последний случай используется для организации обмена в многоконтроллерных системах.

Скорость обмена в асинхронном режиме определяется частотой переполнения первого таймера f_T либо частотой синхросигнала f_0. В режиме 1, 3 скорость обмена равна $f_T / 16$ или $f_T / 32$, а в режиме 2 – $f_0 / 32$ или $f_0 / 64$. При этом выбор конкретного значения скорости определяется флагом SCON – старшим разрядом регистра PCON. Например, если первый таймер настроить на режим 2 и использовать константу перезагрузки $F4_{16} = 244_{10}$, то при $f_0 = 12$ МГц этот таймер будет переполняться каждые $256 - 244 = 12$ мкс. Это соответствует частоте $f_T = 1/12$ мкс $\approx 83{,}3$ кГц, откуда скорость передачи равна 2,6 кбит/с (при SCON = 0). Поэтому при использовании кварцевого резонатора с частотой 11,059 МГц удаётся получить стандартное значение скорости передачи 2400 бит/с. Общая формула вычисления скорости передачи имеет вид:

$$f = \frac{f_0 / 12}{32 \cdot (256 - c)} \quad \text{при SMOD} = 0;$$

$$f = \frac{f_0 / 12}{16 \cdot (256 - c)} \quad \text{при SMOD} = 1;$$

где c – константа перезагрузки таймера. Очевидно, что при $f_0 = 12$ МГц максимальная скорость передачи достигает 62,5 кбит/с в режимах 1, 3, а режиме 2 – 375 кбит/с (для сравнения, в синхронном режиме 0 скорость составляет 1 Мбит/с).

Режимы 2, 3 позволяют построить многоконтроллерную систему, в которой обмен осуществляется через витую пару или коаксиальный кабель. При этом один контроллер должен быть ведущим (master), а остальные – ведомыми (slave). Поскольку команды ведущего контроллера должны восприниматься ведомыми, то выход master-передатчика должен быть подключен ко входу slave-приёмников и наоборот (рис. 30).

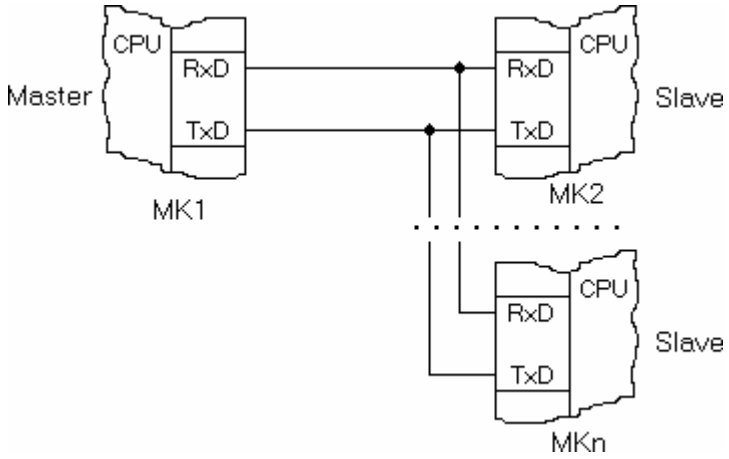


Рис. 30. Многоконтроллерная система

Команды ведущего контроллера передаются всем подчинённым контроллерам, но воспринимаются только тем контроллером, которому они адресованы. Для этого используется девятый бит, позволяющий отличать адресную информацию от данных. Обычно при передаче адреса девятый бит равен “1”, а при передаче данных и команд – “0”. Временная диаграмма работы многоконтроллерной шины приведена на рис. 31.

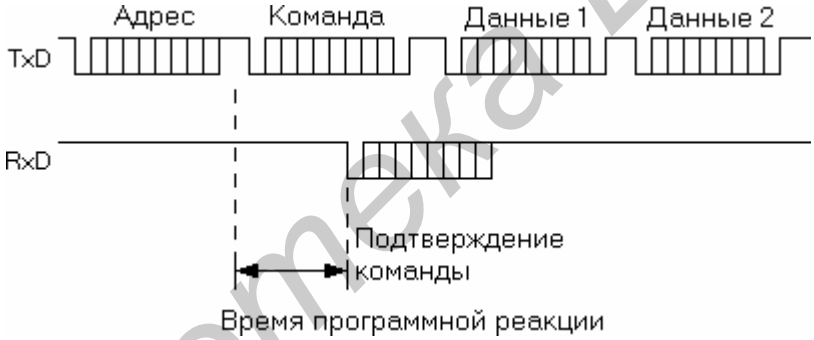


Рис. 31. Временная диаграмма многоконтроллерной системы

Если ведущему микроконтроллеру требуется передать блок данных, то он в режиме «широковещательной» передачи (всем ведомым) сначала выдаёт код адреса получателя (в его девятом бите содержится “1”). При получении адреса во всех ведомых микроконтроллерах происходят прерывания и вызов подпрограммы сравнения полученного адреса с кодом собственного сетевого адреса. Адресуемый микроконтроллер сбрасывает свой управляющий бит SM2 в “0” и готовится к приёму блока данных. Остальные ведомые микроконтроллеры, адрес которых не совпал с принятым, оставляют неизменным состояние SM2 = 1 и передают управление прерванной программе. В дальнейшем байты данных, поступающие в последовательный порт ведомых микроконтроллеров у которых SM2 = 1, прерывание не вызывают (не устанавливается флаг RI), т.е. игнорируются. Этот процесс продолжается до тех пор, пока ведущий микроконтроллер не передаст новый адрес получателя данных.

4.8. Управление энергопотреблением

У микроконтроллеров семейства MCS-51, выполненных по технологии CMOS (80C51), существуют два режима, позволяющих уменьшить потребление энергии в пассивном состоянии:

- режим холостого хода IDL (Idle), ток потребления 4,2 мА.
- режим пониженного энергопотребления PD (Power Down), ток потребления 50 мкА;

Управление этими режимами осуществляется при помощи битов IDL и PD регистра PCON (рис. 32). Если в них одновременно записана “1”, то преимущество имеет PD. Кроме того, регистр PCON содержит флаги пользователя GF0, GF1 и бит SMOD, управляющий скоростью передачи последовательного порта. У микроконтроллеров, выполненных по технологии n-МОП (8051), режимы пониженного энергопотребления отсутствуют.

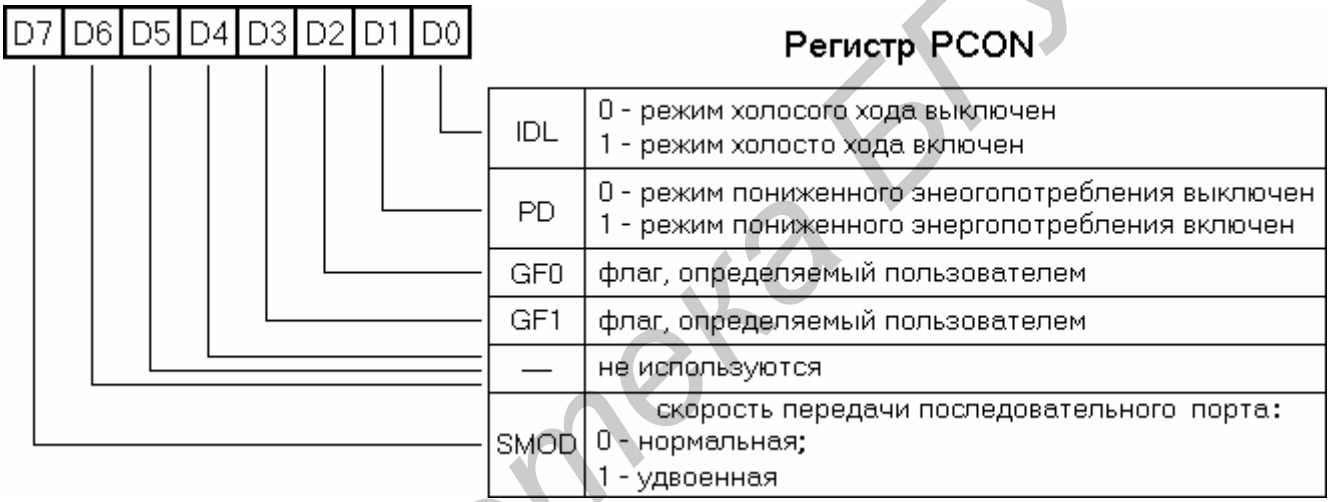


Рис. 32. Регистр управления мощностью PCON

В режиме *холостого хода* (IDL = 1) центральный процессор отключается, но тактовый генератор, система прерываний, таймеры/счётчики и другие блоки ввода/вывода продолжают функционировать. Счётчик команд, регистры и ячейки РПД сохраняют своё значение. На выводах всех портов удерживается состояние, которое было в момент перехода в режим холостого хода. На выводах ALE и PSEN формируется уровень “1”. При этом потребляемая мощность составляет от 15 до 30% от номинальной в зависимости от количества блоков ввода/вывода.

Выйти из режима холостого хода можно по сигналу RESET, либо по любому разрешённому запросу прерывания. Обработка прерывания приводит к аппаратному сбросу бита IDL и выходу из этого режима. После команды RETI (выход из подпрограммы обслуживания прерывания) возобновляется выполнение основной программы. Следует отметить, что выход из режима холостого хода по сигналу RESET приводит к перезапуску программы с нулевого адреса. Однако на перезапуск требуется не менее трёх машинных циклов, в течении которых возобновляется выполнение основной программы. Поэтому после команды,

переводящей контроллер в режим IDL, целесообразно расположить несколько команд NOP. Длительность сигнала RESET должна быть не менее 2 мкс (при тактовой частоте 12 МГц).

В *режиме пониженного энергопотребления* (PD = 1) блокируется работа всех узлов микроконтроллера, поскольку останавливается тактовый генератор. В этом режиме содержимое РПД и регистров специальных функций сохраняется, а на выходных линиях портов удерживаются значения, соответствующие содержимому их буферных регистров. Выходы сигналов ALE и PSEN сбрасываются. При этом электропитание осуществляется через вывод RST/VPD, а основное электропитание (V_{CC}) может быть отключено. Резервное напряжение V_{DD} в этом режиме может быть снижено до 2 В. Перед выходом из режима пониженного энергопотребления оба напряжения V_{CC}, V_{DD} должны быть восстановлены до номинального значения.

Заметим, что при одновременной записи логической “1” в PD и IDL, бит PD имеет преимущество.

Выход из режима пониженного энергопотребления возможен только по сигналу RESET. При этом переопределяются все регистры специальных функций, но содержимое РПД не изменяется. Длительность сигнала RESET должна быть не менее 10 мс, что обеспечивает надёжный запуск и стабилизацию генератора.

Рассмотренные выше режимы могут использоваться не только для снижения энергопотребления, но и для защиты от отказов основного блока питания. При этом можно обеспечить сохранность содержимого РПД с помощью маломощного (батарейного) аварийного источника, подсоединяемого к выводу RST/VPD. При получении прерывания от системы контроля электропитания микроконтроллер должен перегрузить в РПД основные параметры прерванной программы и выдать сигнал, разрешающий подключение к выводу RST/VPD аварийного источника питания. Все эти процедуры должны быть выполнены до того, как напряжение основного источника электропитания упадет ниже рабочего предела. После восстановления номинального значения напряжения в основной цепи электропитания выполняется системный сброс и источник аварийного электропитания может быть отключен.

4.9. Сброс в начальное состояние и синхронизация

В отличие от MCS-48, сброс микроконтроллеров семейства MCS-51 осуществляется путём подачи на вход RESET сигнала “1”. Для надёжности, этот сигнал должен быть удержан на входе RESET по меньшей мере в течение двух машинных циклов (24 периода резонатора).

Под воздействием сигнала RESET сбрасывается содержимое счётчика-команд PC, аккумулятора ACC и его расширителя B, слова-состояния PSW, регистра-указателя DPTR, а также регистра специальных функций TMOD, TCON, THx, TLx, IE, IP и SCON. Начальное состояние остальных регистров определяется следующим образом: в регистре PCON сбрасывается только старший бит SMOD,

в указатель стека SP загружается код 07h, а в буферы портов P0..P3 – код FFh. Состояние регистра SBUF не определено. Сигнал RESET не воздействует на содержимое ячеек РПД. Когда включается электропитание (V_{CC}), содержимое РПД неопределенно, за исключением операции возврата из режима пониженного энергопотребления.

На рис. 33 показана схема автоматического формирования сигнала RESET при включении электропитания. Время, необходимое для полного заряда ёмкости, обеспечивает уверенный запуск резонатора и его работу в течение более чем двух машинных циклов.

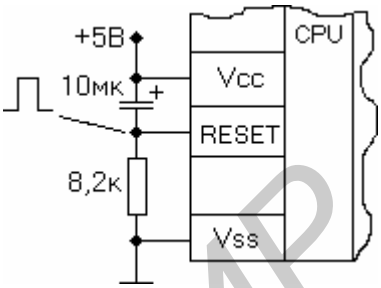
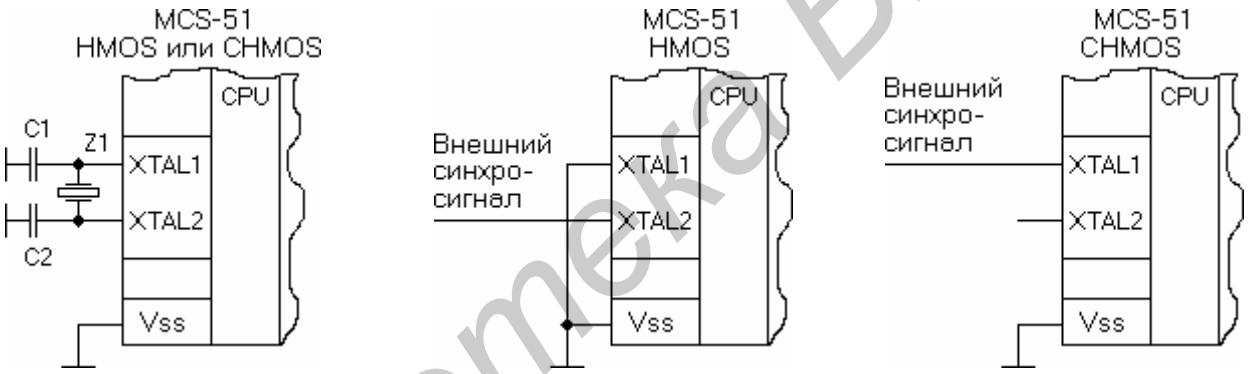


Рис. 33. Схема сброса при включении электропитания

Синхронизация микроконтроллеров (рис. 34) семейства MCS-51 осуществляется от внешнего тактового генератора или внутреннего генератора с внешним кварцевым резонатором, подключаемым к выводам XTAL1 и XTAL2. Отметим, что в HMOS-кристаллах сигнал внешнего генератора подаётся на вывод XTAL2, а в CHMOS – на вывод XTAL1.



а б в

Рис. 34. Синхронизация микроконтроллера:
а – от внутреннего генератора;
б, в – от внешнего генератора

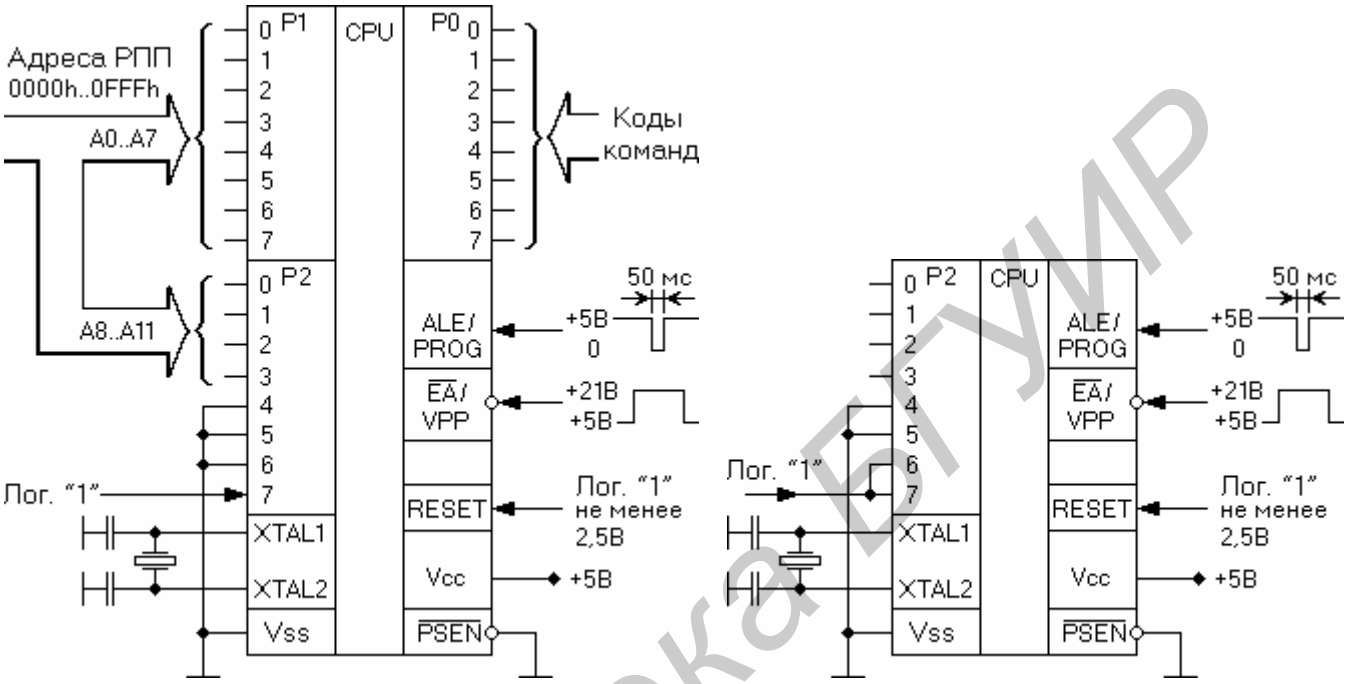
Внутри микроконтроллера частота синхросигнала делится на 12, образуя машинный цикл. Машинный цикл состоит из шести последовательных состояний S1..S6, каждое из которых делится на две фазы P1 и P2. При описании последовательности тактов используются обозначения S1P1, S1P2, S2P1,....,S6P2. Цикл выполнения каждой команды состоит из одного, двух или четырёх машинных циклов.

4.10. Загрузка прикладных программ

Большинство микроконтроллеров MCS-51 может быть запрограммировано пользователем. При необходимости программа может быть стёрта

ультрафиолетовым светом либо электрическими сигналами. Некоторые микроконтроллеры имеют средства защиты от копирования, запрещающие чтение из резидентной памяти программ.

В режиме программирования микроконтроллер должен работать на пониженной частоте 4..6 МГц так как на его внутренней шине P0..P2 осуществляется мультиплексирование адресной и кодовой информации (рис. 35, а)



а б

Рис. 35. Программирование микроконтроллера:
а – загрузка программ; б – запись бита защиты от копирования

В процессе программирования адресная информация подаётся на восемь линий порта P1 и четыре линии порта P2, а загружаемый байт – на линии порта P0. При этом выводы P2.4..P2.6 и PSEN должны быть заземлены, а на выводы P2.7 и RESET необходимо подать высокий уровень. На входе EA/VPP должен поддерживаться уровень +5 В, но в момент загрузки байта он должен быть увеличен до +21 В, а сигнал на входе ALE/PROG – на 50 мс сброшен в 0. После этого напряжение на входе EA/VPP возвращается к уровню +5 В. Источник напряжения +21 В (VPP) должен быть хорошо стабилизирован.

Аналогичным способом осуществляется *запись бита защиты*, который запрещает доступ к РПП любыми внешними средствами. Схема записи бита защиты показана на рис. 35 б. Процедура записи бита защиты такая же, как и при загрузке программ в РПП, но на вывод P2.6 должен подаваться уровень “1”. Сигналы на выводах портов P0, P1 и P2.0..P2.3 могут быть в любом состоянии. Однажды установленный бит защиты можно сбросить только путём полного стирания резидентной памяти программ.

4.11. Система команд семейства MCS-51

Система команд семейства MCS-51 содержит 111 базовых команд, которые подразделяются на пять групп:

- команды пересылки данных (Data Transfer);
- арифметические команды (Arithmetic);
- логические команды (Logic);
- команды ветвления (Program Branching);
- команды операций с битами (Boolean Variable Manipulation).

В отличие от MCS-48 здесь имеются команды умножения, деления, вычитания, операций над битами, операций со стеком и расширенный набор команд передачи управления.

Команда может иметь *длину* один, два или три байта, причем первый байт содержит код операции, а второй и третий - адреса операндов либо непосредственные операнды. Большинство команд (94 из 111) состоят из одного или двух байт.

Время выполнения команды составляет один, два или четыре машинных цикла. При тактовой частоте 12 МГц длительность машинного цикла равна 1мкс, при этом 64 команды выполняются за 1 мкс, 45 команд - за 2 мкс и 2 команды (умножение и деление) - за 4 мкс. Заметим, что в отличие от MCS-48, здесь нет прямой связи между длиной команды и числом машинных циклов, требуемых для ее выполнения.

В командах MCS-51 используется *четыре типа операндов* - биты, тетрады (4 бита), байты и 16-битные слова. Однобитный операнд может содержаться в любом из 128 флагов РПД, а также в отдельных разрядах буферов ввода/вывода и некоторых регистров специальных функций. Четырехбитные операнды используются только при операциях обмена (команды SWAP и XCHD). Восьмибитные операнды могут размещаться в аккумуляторе, регистрах R0..R7, в ячейках памяти программ или данных (резидентной или внешней), в регистрах специальных функций, а также в буферах портов ввода/вывода. Двухбайтные операнды – это константы и прямые адреса, для представления которых используются второй и третий байты команды.

Система команд MCS-51 поддерживает те же методы адресации, что и MCS-48, однако имеет некоторую специфику, обусловленную большим объемом внешней памяти программ и данных. При задании адреса операнда используются следующие методы:

- **непосредственная адресация** - операнд содержится непосредственно в команде и может занимать один или два байта ($data_8$, $data_{16}$);
- **прямая адресация –** в команде содержится однобайтный адрес операнда adr_8 (используется только для резидентной памяти данных и регистров специальных функций);

- **регистровая адресация** – в команде имеется трехбитовое поле, определяющее номер регистра, содержащего операнд; выбор одного из четырех регистровых банков осуществляется программированием битов селектора банка (RS1, RS0) в PSW.
- **косвенно-регистровая адресация** - в команде содержится ссылка на регистр, содержащий адрес операнда (используется при обращении к РПД и ВПД через 8-разрядные регистры R0 и R1, указатель стека SP, а также через 16-разрядный регистр-указатель DPTR);
- **индексная адресация –** адрес операнда находиться сложением 8–разрядного индекса, содержащегося в аккумуляторе, и 16-разрядной базовой величины, содержащейся в DPTR или PC (используется только при чтении из памяти программ, при выборке из таблиц).
- **неявная адресация** – в команде отсутствует специальное адресное поле, определяющее адрес операнда, но его расположение полностью определяется кодом операции (используется в командах, которые по умолчанию выполняют операции только с аккумулятором или DPTR).

Следует отметить, что, в отличие от других процессоров, преимущества регистровой адресации по времени выполнения команд здесь не столь существенные (за один машинный цикл выполняются и некоторые команды с прямой либо косвенной адресацией).

При программировании на ассемблере для ASM51 обычно используются символические адреса регистров специальных функций и их отдельных бит. Например, ACC.5 - символическое имя пятого бита аккумулятора; SMOD (либо PCON.7) - символическое имя бита удвоенной скорости передачи последовательного интерфейса. Эти символические адреса являются зарезервированными словами ASM51, и их не надо определять с помощью директив ассемблера.

4.12. Команды пересылки данных

Команды этой группы обеспечивают выполнение операций копирования (пересылка байтов и битов между памятью и регистрами процессора). Они делятся на три типа:

- команды пересылки, использующие РПД и РСФ (команды типа MOV);
- команды пересылки, использующие ВПД (команды типа MOVX);
- команды пересылки, использующие ПП (команды типа MOVC).

Эти команды не модифицируют флаги результата, за исключением команд загрузки PSW и аккумулятора (флаг паритета). Большую часть команд данной группы составляют команды передачи и обмена байтов. Команды пересылки бит представлены далее в группе команд битовых операций

Команды MOV (табл. 4) позволяют пересылать данные между аккумулятором, регистрами общего назначения, ячейками резидентной памяти

данных и регистрами специальных функций. В приведенной таблице используются следующие обозначения: src (source) – источник; dest (destination) – приемник; Rn – регистр общего назначения R0…R7; Ri – регистр-указатель адреса при косвенной адресации R0, R1; #ad8 – восьмиразрядный адрес; #d8 – восьмиразрядный непосредственный операнд; #d16 – 16-разрядный непосредственный операнд.

Следует заметить, что в отличие от MCS-48 пересылки между ячейками памяти здесь могут выполняться без участия аккумулятора. Кроме того, в зависимости от метода адресации аккумулятор имеет два различных имени: A – при неявной адресации (например, MOV A, R0) и ACC – при использовании прямого адреса. В последнем случае аккумулятор рассматривается как регистр специальных функций с адресом E0h, для которого также допускается побитовый доступ. Использование неявной адресации аккумулятора предпочтительнее, но не всегда возможно (например при обращении к его отдельным битам).

Таблица 4

Команды пересылки, использующие РПД и РСФ

Мнемокод	**Операция**	**Операнды**
MOV A, src	A ← src	src = Rn, @Ri, ad8, #d8
MOV dest, A	dest ← A	dest = Rn, @Ri, ad8

MOV dest, src	dest ← src	src = Rn, @Ri, ad8, #d8
		dest = Rn, @Ri, ad8

MOV DPTR, #d16	DPTR ← #d16	DPTR
PUSH src	SP ← SP + 1 @SP ← src	src = ad8
POP dst	@SP ← dst SP ← SP – 1	dst = ad8
XCH A, byte	A ↔ byte	byte = Rn, @Ri, ad8
XCHD A, @Ri	Low_A ↔ Low_byte	byte = @Ri

В первую группу также входят команды PUSH и POP, обеспечивающие соответственно запись данных в стек и чтение из стека. В MCS-51 стек располагается в резидентной памяти данных, поэтому размер стека ограничен лишь ее объемом. При этом следует учитывать, что по сигналу RESET в указатель стека заносится код 07h, поэтому первый элемент стека располагается в ячейке памяти с адресом 08h, используемой для регистра R0 банка RB1. Поэтому в сложных системах после сброса следует переопределить начальное значение стека. Кроме пересылок команды первой группы могут осуществлять обмен байтами между аккумулятором и ячейками РПД (команда XCH), а также обмен младшими тетрадами, т.е. битами 0 - 3 (команда XCHD).

Следует отметить, что MCS-51 не имеет специальных команд ввода-вывода и команд для работы с регистрами специальных функций. Доступ к ним осуществляется обычными командами MOV с использованием прямого адреса (допускается также использование символических адресов). Например, пересылка PSW в аккумулятор выполняется командой MOV A, PSW, которая преобразуется ассемблером к виду MOV A, 0D0h. Аналогично, вывод из аккумулятора в порт P0 осуществляется командой MOV P0, A эквивалентной MOV A, 80h. Ещё одна особенность команд типа MOV проявляется в микроконтроллерах с расширенным объемом РПД, где обращение к верхним 128 байтам резидентной памяти данных осуществляется только в режиме косвенной адресации (при использовании прямого адреса выполняется обращение к РСФ).

Команды MOVX (табл. 5) позволяют пересылать данные между аккумулятором и ячейками ВПД. При этом используется косвенная адресация через 16-разрядный регистр-указатель DPTR либо 8-разрядные регистры R0, R1. В первом случае обеспечивается доступ к любой из 65 536 ячеек ВПД, в втором – к странице объемом 256 байт (старший байт адреса, т.е. номер страницы, определяется текущим содержимым буфера порта P2).

Таблица 5

Команды пересылки, использующие ВПД

Мнемокод	**Операция**	**Разрядность адреса**
MOVX A, @Ri	A ← @Ri	8
MOVX @Ri, A	@Ri ← A	8
MOVX A, @DPTR	A ← @DPTR	16
MOVX @DPTR, A	@DPTR ← A	16

Команды MOVC (табл. 6) используются для чтения данных, размещенных в памяти программ (констант и таблиц функций). При обращении к таблице обычно применяется первая из этих команд, причём регистр DPTR устанавливается на точку начала таблицы, а в аккумулятор помещается индекс табличной ячейки. При использовании второй команды следует учитывать, что в момент вычисления суммы A+PC содержимое счётчика команд PC уже увеличено на 1 (т.е. указывает на следующую команду).

Таблица 6

Команды пересылки из памяти программ

Мнемокод	**Операция**
MOVC A, @A+DPTR	A ← @ (A+DPTR)
MOVC A, @A+PC	A ← @ (A+PC)

4.13. Команды арифметических операций

Микроконтроллеры MCS-51 имеют следующие арифметические команды: сложение ADD, сложение с учётом переноса ADC, вычитание с заёмом SUBB, инкремент INC, декремент DEC, десятичная коррекция DA A, умножение MUL и деление DIV (табл. 7). Большинство из них выполняет операции над 8-разрядными операндами (кроме INC DPTR, которая обеспечивает инкремент 16-разрядного регистра-указателя).

Таблица 7

Команды арифметических операций

Мнемокод	Операция	Операнды	Флаги
ADD A, byte	A ← A + byte	byte = Rn, @Ri, ad8, #d8	C, AC, OV, P
ADDC A, byte	A ← A + byte + flagC	byte = Rn, @Ri, ad8, #d8	C, AC, OV, P
SUBB A, byte	A ← A – byte – flagC	byte = Rn, @Ri, ad8, #d8	C, AC, OV, P
INC byte	byte ← byte + 1	byte = A, Rn, @Ri, ad8	–
DEC byte	byte ← byte – 1	byte = A, Rn, @Ri, ad8	–
INC DPTR	DPTR ← DPTR+1	DPTR	–
MUL AB	BA ← B·A	A, B	C=0, OV
DIV AB	A(B) ← A/B	A, B	C=0, OV
DA A	Двоично- десятичная коррекция	A	C, AC, OV, P

При выполнении арифметических операций формируется флаги переноса C, дополнительного переноса AC, переполнения OV и паритета P, которые помещаются в регистр PSW (рис. 36). Они формируются следующим образом:

- C = 1, если в старшем бите результата возникает перенос или заем;
- AC = 1, если возникает перенос/заем между тетрадами;
- P = 1, если число единичных бит в аккумуляторе нечетно;
- OV = 1, если старший бит результата не может быть интерпретирован как знаковый (например, при суммировании положительных чисел получен отрицательный результат).

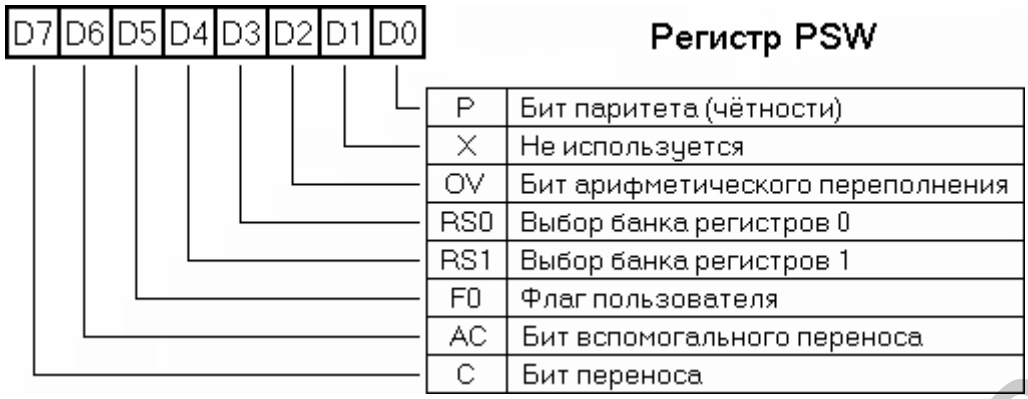


Рис. 36. Формат слова состояния PSW

Отметим, что все четыре флага формируются только при выполнении операций сложения и вычитания, а операции умножения и деления флаг C сбрасывают (табл. 8). При умножении флаг OV устанавливается, если результат больше 255. При делении флаг OV устанавливается при делении на нуль. Особенности формирования флагов C и OV поясняются примерами, приведенными ниже (рис. 37, 38).

$FF_{16} + FF_{16} = FE_{16}$	$7F_{16} + 7F_{16} = FE_{16}$	$01_{16} + 01_{16} = 02_{16}$	$81_{16} + 81_{16} = 02_{16}$
(-1) + (-1) = (-2)	(+127) + (+127) = (-2)	(+1) + (+1) = (+2)	(-127) + (-127) = (+2)
C=1, OV=0	C=0, OV=1	C=0, OV=0	C=1, OV=1

Рис. 37. Формирование флагов C и OV при сложении

$FF_{16} - 01_{16} = FE_{16}$	$7F_{16} - 81_{16} = FE_{16}$	$01_{16} - FF_{16} = 02_{16}$	$81_{16} - 7F_{16} = 02_{16}$
(-1) – (+1) = (-2)	(+127) – (-127) = (-2)	(+1) – (-1) = (+2)	(-127) – (+127) = (+2)
C=0, OV=0	C=1, OV=1	C=1, OV=0	C=0, OV=1

Рис. 38. Формирование флагов C и OV при вычитании

Поскольку в MCS-51 отсутствует команда вычитания без учета заёма, то для выполнения такой операции используется команда SUBB (перед этим программно сбрасывается флаг переноса).

Команда десятичной коррекции DAA позволяет выполнить сложение чисел, представленных в двоично-десятичном коде. Она корректирует 8-битовое число, полученное в результате выполненной ранее команды сложения ADD или ADDC. Если содержимое младшей тетрады аккумулятора превышает 9 или AC=“1”, то к ней прибавляется 06h, при этом получается младшая двоично-десятичная цифра. Указанное сложение не изменяет содержимое флага AC, но может установить флаг C, если перенос из младшей тетрады распространяется через всю старшую тетраду (в противном случае C не изменяется). Далее, если C=“1” или содержимое старшей тетрады превышает 9, то содержимое аккумулятора увеличивается на 60h, создавая старшую двоично-десятичную цифру. При этом флаг C

устанавливается (не изменяется), если имеется (отсутствует) перенос из старшего бита аккумулятора.

Операция умножения MUL выполняется над содержимым аккумулятора A и регистра B, полученный результат размещается в двух байтах AB (младший байт в B, старший - в A). Для операции деления DIV в аккумулятор помещается частное, а в регистр B – остаток. При этом надо учитывать, что операнды интерпретируются как 8-разрядные целые числа без знака.

4.14. Команды логических операций

Система команд MCS-51 включает в себя стандартные логические операции И, ИЛИ, исключающее ИЛИ (ANL, ORL, XRL), а также операции инвертирования CPL, сброса CLR, циклического сдвига RL, RLC, RR, RRC и обмена тетрадами SWAP (см. табл. 8). Операции ANL, ORL, XRL выполняются поразрядно над байтами, располагаемыми в аккумуляторе, в ячейках резидентной памяти данных или регистрах специальных функций. Остальные операции выполняются только над содержимым аккумулятора.

В отличие от MCS-48 здесь имеется возможность производить операцию XRL с содержимым буферов портов P0…P3, что может быть эффективно использовано для инверсии их отдельных линий.

Таблица 8

Команды логических операций

Мнемокод	Операция	Операнды	Флаги
ANL A, byte	A ← A & byte	byte = Rn, @Ri, ad8, #d8	P
ANL ad8, byte	ad8 ← ad8 & byte	byte = A, #d8	–
ORL A, byte	A ← A ∨ byte	byte = Rn, @Ri, ad8, #d8	P
ORL ad8, byte	ad8 ← ad8 ∨ byte	byte = A, #d8	–
XRL A, byte	A ← A ⊕ byte	byte = Rn, @Ri, ad8, #d8	P
XRL ad8, byte	ad8 ← ad8 ⊕ byte	byte = A, #d8	–
CLR A	A ← 00h	A	P=1
CPL A	A ← NOT(A)	A	P=1
RL A	$A_{k+1} \leftarrow A_k$; $k = 0 \div 6$, $A_0 \leftarrow A_7$	A	–
RLC A	$A_{k+1} \leftarrow A_k$; $k = 0 \div 6$, $A_0 \leftarrow C$; $C \leftarrow A_7$	A	C, P
RR A	$A_{k+1} \rightarrow A_k$; $k = 0 \div 6$, $A_0 \rightarrow A_7$	A	–
RRC A	$A_{k+1} \rightarrow A_k$; $k = 0 \div 6$, $A_0 \rightarrow C$; $C \rightarrow A_7$	A	C, P
SWAP A	$A_{7\ldots4} \rightarrow A_{3\ldots0}$	A	–

4.15. Команды операций над битами

Микроконтроллеры MCS-51 содержат в своем составе “булевый” процессор, имеющий свое побитово-адресуемое ОЗУ, свои порты ввода/вывода и выполняющий свой набор команд. При выполнении этих команд флаг C выполняет функции аккумулятора.

Внутреннее ОЗУ “булевого” процессора имеет объём 256 бит. В него входят 128 прямоадресуемых бит из области РПД (ячейки с адресами 20h-2Fh), а также битовые поля некоторых регистров специальных функций (адреса которых кратны восьми): ACC, B, P0…P3, TCON, SCON, IE, IP, PSW. Побитно-адресуемыми являются, в частности, все линии ввода-вывода, каждая линия которых может рассматриваться как однобитовый порт. При работе с битами используется только прямая адресация.

Каждый из отдельно адресуемых бит может быть установлен в “1”, сброшен в “0”, инвертирован, передан во флаг C или принят из него. Между любым прямоадресуемым битом и флагом C могут быть произведены логические операции И, ИЛИ, результат которых заноситься во флаг C. Кроме того, по состоянию адресуемого бита можно проверить и выполнить соответствующий условный переход. Список команд операций над юитами приведён в табл 9, где bit обозначает прямой восьмиразрядный адрес.

Таблица 9

Команды побитовой обработки

Мнемокод	Операция	Мнемокод	Операция
ANL C, bit	C ← C & bit	CLRC	C ← 0
ANL C, /bit	C ← C & NOT(bit)	CLR bit	bit ← 0
ORL C, bit	C ← C ∨ bit	SETB C	C ← 1
ORL C, /bit	C ← C ∨ NOT(bit)	SETB bit	C ← bit
MOV C, bit		CPL C	C ←NOT(C)
MOV bit, C	bit ← C	CPL bit	bit ←NOT(bit)

4.16. Команды передачи управления

В эту группу входят команды условного и безусловного перехода, а также команды вызова подпрограмм и возврата из них (табл. 10). большинстве из них используется прямая адресация, т.е. адрес перехода целиком (или его часть) содержится в самой команде передачи управления.

В зависимости от разрядности указываемого адреса и способа его формирования различают четыре типа команд безусловного перехода:

- LJMP ad16 – *длинный переход* (в команде содержится полный 16-битный адрес, обеспечивается переход по всему адресному пространству объёмом 64 Кбайт, длина команды – три байта);

- AJMP ad11 – *абсолютный переход* (в команде содержится 11 младших бит адреса, обеспечивается переход в пределах текущей страницы объёмом 2048 байт, длина команды – два байта);
- SJMP rel8 – *короткий относительный переход* (в команде задается 8-разрядное смещение адреса относительно содержимого счётчика команд, обеспечивается переход в пределах –128…+127 байт относительно адреса следующей команды, длина команды – два байта);
- JMP @A+DPTR – *косвенный переход* (в аккумуляторе задается 8-разрядное смещение адреса относительно DPTR, обеспечивается переход в пределах –128…+127 байт относительно этого регистра, длина команды – один байт).

На практике редко возникает необходимость перехода в пределах всего адресного пространства и чаще используются укороченные команды перехода, занимающие меньше места в памяти программ. При программировании на ассемблере допускается использование мнемонического обозначения JMP, поэтому выбор конкретного варианта команды безусловного перехода LJMP, AJMP или SJMP осуществляется компилятором.

Таблица 10

Команды передачи управления

Мнемокод	Операция	Мнемокод	Операция
LJMP ad16	Длинный безусловный переход	DJNZ Rn, rel8	DEC Rn и переход, если byte ≠ 0
AJMP ad11	Абсолютный безусловный переход	DJNZ byte, rel8	DEC byte и переход, если byte ≠ 0
SJMP rel8	Короткий безусловный переход (относительный)	CJNE A, byte, rel8	Переход, если A ≠ byte
JZ rel8	Переход, если A = 0	CJNE A, #d8, rel8	Переход, если A ≠ #d8
JNZ rel8	Переход, если A ≠ 0	CJNE Rn, #d8, rel8	Переход, если Rn ≠ #d8
JC rel8	Переход, если flagC = 1	CJNE @Ri, #d8, rel8	Переход, если @Ri ≠ #d8
JNC rel8	Переход, если flagC = 0	LCALL ad16	Вызов подпрограммы (длинный)
JB bit, rel8	Переход, если bit = 1	ACALL ad11	Вызов подпрограммы (абсолютный)
JNB bit, rel8	Переход, если bit = 0	RET	Возврат из подпрограммы
JBC bit, rel	Переход, если bit = 1; затем CLR bit	RETI	Возврат из прерывания

Команды условных переходов используют только относительную адресацию, аналогичную SJMP. Они осуществлять ветвление по следующим условиям: аккумулятор содержит нуль (JZ); содержимое аккумулятора не равно нулю (JNZ); перенос равен единице (JC); перенос равен нулю (JNC); адресуемый бит равен единице (JB); адресуемый бит равен нулю (JNB).

Команда DJNZ предназначена для организации циклов, при этом в отличие от MCS-48 в качестве счётчика может использоваться не только регистр, но и любой прямоадресуемый байт (например, ячейка РПД). В процедурах ожидания какого-либо события удобно использовать команду CJNE, обеспечивающую сравнение байта с однобайтным операндом и переход в случае неравенства.

Все команды переходов, за исключением CJNE и JBC, не оказывают воздействия на флаги. Команда CJNE устанавливает флаг C, если первый операнд оказывается меньше второго. Команда JBC сбрасывает флаг C в случае перехода.

Для обращения к подпрограммам используются команды LCALL, ACALL. Эти команды в отличие от команд перехода LJMP, AJMP сохраняют в стеке адрес возврата в основную программу. Для возврата из подпрограммы необходимо выполнить команду RET. Команда RETI отличается от команды RET тем, что разрешает прерывания обслуженного уровня.

Литература

1. Микропроцессорные системы: Учебное пособие для вузов / Под ред. Д.В. Пузанкова. – СПб.: Политехника, 2002. – 935 с.
2. Бродин В.Б., Шагурин М.И. Микроконтроллеры: Архитектура, программирование, интерфейс. – М.: ЭКОМ, 1999. – 400 с.
3. Предко М. Руководство по микроконтроллерам: В 2 т. / Пер.с англ.; Под ред. И.И. Шагурина, С.Б. Лужанского. – М.: Постмаркет, 2001. – Т1 – 416 с., Т2 – 488 с.
4. Схемотехника электронных систем: Микропроцессоры и микроконтроллеры // Бойко В., Гуржий А., Жуйков В. и др.. – СПб.: BHV-Санкт-Петербург, 2004. – 464 с.
5. Бродин В.Б., Калинин А.В. Системы на микроконтроллерах и БИС программируемой логики. – М.: ЭКОМ, 2002. – 399 с.
6. Микропроцессорные системы и микроЭВМ в измерительной технике: Учеб. пособие для вузов / Под ред. А.Г. Филиппова. – М.: Энергоатомиздат, 1995. – 368 с.
7. Каспер Э. Программирование на языке Ассемблера для микроконтроллеров семейства i8051. – М.: Горячая линия – Телеком, 2003. – 191 с.
8. Микропроцессоры в системах автоматического управления: INTEL 8XC196MC: Учеб. пособие / Н.Г. Бутырин, О.П. Кан, А.Л. Логинов, А.Н. Щербина. 1995. – 114 с.
9. Кузьминов А.Ю. Интерфейс RS232. Связь между компьютером и микроконтроллером. – М: Радио и связь, 2004. – 168 с.
10. Каратаев Н.В. Микроконтроллеры для бытовой аппаратуры. – М.: Додэка-XXI, 2001. – 207 с.
11. Козаченко В.Ф. Микроконтроллеры: Руководство по применению 16-разряд. микроконтроллеров Intel MCS-196/296 во встроенных системах управления. – М.: ЭКОМ, 1997. – 686 с.
12. Шагурин И.И. Микропроцессоры и микроконтроллеры фирмы Motorola: Справ.пособие. – М. : Радио и связь, 1998. – 556 с.
13. Таверньe К. PIC-микроконтроллеры: Практика применения: Пер.с фр. – М.: ДМК Пресс, 2002. – 272 с.
14. Яценков В.С. Микроконтроллеры Microchip. Практическое руководство. – М.: Горячая линия – Телеком, 2002. – 296 с.
15. Корнеев В.В., Киселев А.В. Современные микропроцессоры. – М.: Нолидж, 2000.

Св. план 2006, поз. 37

Учебное издание

Пашкевич Анатолий Павлович,
Чумаков Олег Анатольевич

МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

Конспект лекций
для студентов специальности
I-53 01 07 «Информационные технологии и управление
в технических системах»
дневной формы обучения

В 2-х частях

Часть 2

Ответственный за выпуск О.А. Чумаков

Подписано в печать 2.06.2006 г.	Формат 60×84 1/16.	Бумага офсетная.
Гарнитура «Таймс».	Печать ризографическая.	Усл. печ. л. 4,3.
Уч.-изд.л. 4,1.	Тираж 250 экз.	Заказ 189.

Издатель и полиграфическое исполнение: Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0056964 от 01.04.2004. ЛП №02330/0131518 от 30.04.2004.
220013, Минск, П.Бровки, 6.