Н. Н. ЩЕЛКУНОВ
А. П. ДИАНОВ

Микропроцессорные средства и системы

МОСКВА
«РАДИО И СВЯЗЬ»
1989

ББК 32.852
Щ 45
УДК 681.325.5—181.4

Рецензенты: И. И. Шагурин, Б. В. Шевкопляс

Редакции литературы по электронной технике

Щелкунов Н. Н., Дианов А. П.

Щ 45 Микропроцессорные средства и системы.—М.: Радио и связь, 1989.—288 с.: ил.

ISBN 5-256-00256-2.

Рассмотрены вопросы разработки микропроцессорных систем и их составных частей. Основное внимание уделено организации технических средств на базе широко используемых микропроцессорных комплектов серий КР580, К1810 и К1816. Изложение сопровождается большим числом практических примеров.

Для инженерно-технических работников, занимающихся проектированием средств автоматизации и вычислительной техники.

Щ $\frac{2302030700\text{-}145}{046(01)\text{-}89}$ 103-89 **ББК 32.852**

ISBN 5-256-00256-2

ПРЕДИСЛОВИЕ

Достигнутые к началу 70-х гг. успехи в области технологии интегральных микросхем и организации вычислительных устройств привели к появлению нового класса приборов — микропроцессоров. Сегодня микропроцессорная техника — индустриальная отрасль со своей методологией и средствами проектирования.

К настоящему времени накоплен большой практический опыт проектирования микропроцессоров и микропроцессорных систем, область применения которых постоянно расширяется. Отсутствие изданий, которые могли бы служить практическим руководством для специалистов, занимающихся проектированием средств автоматизации и вычислительной техники, побудило авторов написать данную книгу.

Основное внимание в книге уделено организации систем на базе широко распространенных микропроцессорных БИС серий КР580, К1810, К1816. Состав и организация типовой микропроцессорной системы рассматриваются в гл. 1. Там же вводится ряд основных определений и понятий. Функциональные и технические характеристики центральных процессоров, построенных на базе 8-разрядных микропроцессоров типов КР580ВМ80, К1821ВМ85А, приводятся в гл. 2. Вопросы, связанные с организацией подсистемы ввода-вывода (технические характеристики ряда периферийных БИС, организация типовых интерфейсов ввода-вывода и средств поддержки режима реального времени), рассматриваются в гл. 3. В гл. 4 вниманию читателей предлагаются однокристальные микроЭВМ типов К1816ВЕ48 и К1816ВЕ51. Своеобразие организации этих перспективных приборов обусловлено необходимостью размещения на одном кристалле всех компонентов микроЭВМ. Гл. 5 посвящена организации законченных одноплатных микроЭВМ на базе БИС типа КР580ВМ80, рассматриваются вопросы их программирования. В гл. 6 описаны характеристики 16-разрядных микропроцессоров типов К1810ВМ86 и К1810ВМ88. С методикой построения одноплатных микроЭВМ на базе 16-разрядного микропроцессора К1810ВМ86 читатель познакомится в гл. 7.

В книге вопросы общего характера сочетаются с конкретными практическими решениями, соответствующими современным тенденциям развития микропроцессорной техники. Так, при рассмотрении более сложных организаций новых микропроцессорных БИС теоретические рассуждения позволяют более полно оценить их преимущества и недостатки.

Книга написана на основе курса лекций, читаемых в Московском физико-техническом институте, а также практического опыта авторов, полученного при разработке и отладке ряда микропроцессорных систем.

ВВЕДЕНИЕ

Микропроцессор (МП) представляет собой функционально завершенное универсальное программно-управляемое устройство цифровой обработки данных, выполненное в виде одной или нескольких микропроцессорных БИС.

Микропроцессорные БИС относятся к новому классу микросхем, одной из особенностей которого является возможность программного управления работой БИС с помощью определенного набора команд. Эта особенность нашла отражение в программно-аппаратном принципе построения *микропроцессорных систем*, или *микросистем* (МС),— цифровых устройств или систем обработки данных, контроля и управления, построенных на базе одного или нескольких МП. Программно-аппаратный принцип построения МС является одним из основных принципов их организации и заключается в том, что реализация целевого назначения МС достигается не только аппаратными средствами, но и с помощью программного обеспечения — организованного набора программ и данных.

Микросистема общего назначения, содержащая кроме одного или нескольких МП память для хранения управляющих программ и данных, а также средства обмена информацией с периферийными устройствами ввода-вывода, называется *микропроцессорной ЭВМ* (микроЭВМ). Примерами периферийных устройств ввода-вывода, с которыми микроЭВМ может обмениваться информацией, служат алфавитно-цифровая клавиатура, дисплеи, принтеры, накопители на гибких магнитных дисках и др. МикроЭВМ, совмещенная с периферийными устройствами, называется *микровычислительным комплексом*.

При разработке МС приходится принимать во внимание большое число особенностей МП и микропроцессорных комплектов БИС: технологических, конструктивных, временны́х, энергетических, эксплуатационных, функциональных и др. Функциональные особенности характеризуют логическую организацию МП и микропроцессорных БИС, принципы их построения, использования и взаимодействия. Они являются основными, так как определяют прибор как функциональный элемент МС, раскрывают его потенциальные возможности логической и арифметической обработки информации.

На современном этапе развития микропроцессорной техники практически любую МС можно отнести к одному из трех классов:

системы на базе секционированных микропроцессорных БИС с микропрограммным управлением;

системы на основе однокристальных МП с программным управлением;

системы с сокращенным набором команд.

Основой элементной базы МС первого класса служат секционированные микропроцессорные комплекты БИС с микропрограммным управлением. Главными отличительными признаками этих комплектов являются: секционированность БИС, позволяющая из малоразрядных секций создавать многоразрядные МС; наличие независимых шин адреса, данных и управления, обеспечивающих разнообразие архитектур и поточную обработку; наличие встроенных трехстабильных выходных буферов с большим коэффициентом разветвления; микропрограммируемость.

Первые секционированные комплекты, выполненные на схемах транзисторно-транзисторной логики с диодами Шотки (ТТЛШ), были освоены в 1974 г. К ним относятся серии 3000 фирмы Intel (США) [16] и 2900 фирмы Advanced Micro Devices (США) [26]. На их основе был спроектирован 16-разрядный центральный процессор на плате 150×150 мм. Такой процессор содержал около 20 БИС, длительность его машинного цикла составляла 125 нс. К данному классу относятся отечественные микропроцессорные комплекты серий К585, К589, К1802, К1804 [40] и др. Следует отметить, что серия 2900 положила начало стандартному набору секционированных микропрограммируемых БИС, совершенствование которых продолжается до настоящего времени.

В основе систем второго класса лежат однокристальные МП с программным управлением и фиксированным набором команд. К приборам данного типа относится первый в мире промышленный 4-разрядный МП 4004 [9], выпущенный фирмой Intel в 1971 г. Этот год считают годом начала развития микропроцессорной техники. Однако надежды, возлагаемые на МП с программным управлением, полностью оправдались лишь с появлением 8-разрядного МП 8080 [59] в 1973 г. и 16-разрядного МП 8086 [27] в 1978 г., ставших типовыми представителями приборов данного класса.

Универсальность и большая функциональная насыщенность МП с программным управлением создали условия для разработки компактных и дешевых МС различного назначения [1, 17, 23, 44]. Затраты на проектирование таких систем по сравнению с МС первого класса значительно ниже, что объясняется наличием развитых средств проектирования и наборов вспомогательных и периферийных БИС, выполняющих разнообразные функции ввода-вывода и расширяющих функциональные возможности МС. Именно поэтому МС второго класса нашли самое широкое

распространение в нашей практической деятельности. Вопросы, связанные с их проектированием, являются основной темой данной книги.

Среди освоенных отечественной промышленностью однокристальных МП с программным управлением можно выделить две основные группы. Первую группу представляют МП серии К1801 [37], вторую — МП серий КР580 [36], К1821 [65] и К1810 [22]. Обе группы МП имеют функционально развитую организацию, снабжены наборами разнообразных периферийных БИС, что дает разработчику широкие возможности при проектировании МС. Более детальный анализ состава и функциональных возможностей этих приборов показывает, что МП КР580, К1821 и К1810 имеют ряд преимуществ по сравнению с МП К1801 и совместимыми с ними БИС других серий. Если последние разрабатывались для применения исключительно в микровычислительных комплексах, то организация первых сориентирована на управляющие системы. По-видимому, такая функциональная ориентация групп МП не предусматривалась заранее, а сложилась исторически.

Особую группу МС с программным управлением образуют микроЭВМ с хранимым в постоянной памяти прикладным программным обеспечением. Такие системы, встраиваемые в аппаратуру потребителя и предназначенные для управления ею в реальном масштабе времени, называются *программируемыми микроконтроллерами*. Интерес к организации микроконтроллеров и принципу их использования возрос с появлением в конце 70-х гг. однокристальных приборов этой группы.

В течение четырех лет, начиная с 1976 г., фирмой Intel разрабатывалось получившее широкое распространение семейство 8-разрядных однокристальных микроконтроллеров с программным управлением iMCS-48, базовым представителем которого является прибор 8048 [44, 60]. Заложенная в те годы архитектура семейства считается стандартной. Появилось множество кристаллов, архитектурно совместимых с iMCS-48. К их числу принадлежат отечественные однокристальные микроЭВМ типов К1816ВЕ35 и К1816ВЕ48 [17, 21].

Вычислительные возможности первых однокристальных микроЭВМ были исчерпаны уже к началу 80-х гг. Встала задача разработки новых микроконтроллеров, обладающих расширенными функциональными ресурсами. Среди предложенных новых архитектур однокристальных микроЭВМ следует выделить 8-разрядную архитектуру семейства микроконтроллеров iMCS-51 [32], предложенного фирмой Intel в 1981 г. Она удовлетворяет всем требованиям, предъявляемым к однокристальным микроконтроллерам, и является доминирующей до настоящего времени. Отечественной промышленностью освоено производство БИС типа К1816ВЕ51.

Новое направление развития микропроцессорной техники сложилось в 1984—1986 гг. Это МС третьего класса, которые известны как системы с сокращенным набором команд (RISC — Reduce Instruction Set Computer). Организация RISC подчинена задаче достижения максимальных скоростей. Основная ее особенность состоит в использовании небольшого набора часто используемых команд одинакового формата, которые могут быть выполнены за один микротакт центрального процессора. Более сложные редко используемые команды реализуются на программном уровне. Однако за счет значительного повышения скорости исполнения сокращенного набора команд средняя производительность RISC-процессоров оказывается выше, чем у обычных МП.

ГЛАВА 1.

ОРГАНИЗАЦИЯ МИКРОСИСТЕМ

1.1. Понятия организации и архитектуры

Под *организацией* МС понимают состав ее программно-аппаратных средств, связи между ними и их функциональные характеристики. Микросистемы имеют многоуровневую иерархическую организацию со многими составными компонентами на каждом уровне. С нижним уровнем функционального описания МС и ее составляющих связано понятие *физической организации* МС — ее *принципиальная схема.* Термин *логическая организация* относится к более высоким уровням описания МС. Так, логическая организация на уровне аппаратуры — это состав, функциональные связи и характеристики взаимодействия аппаратных модулей в процессе выполнения различных задач, которые обычно называют *структурной схемой* или *структурой.* Рассматривая логическую организацию на уровне программного обеспечения (ПО), говорят о вычислительной среде и ее особенностях.

Конечная цель проектирования МС — создание работоспособного и оптимального изделия на базе одного или нескольких МП. Возможность ее достижения определяется в первую очередь выбором рационального соотношения между программными и аппаратными средствами МС. Для этого вводится понятие архитектуры.

Архитектура МС — это функциональные возможности аппаратных средств МС, используемые для представления программ и данных, а также для управления процессом вычислений [29]. Архитектура служит примером вычислительной среды нижнего уровня, связанной непосредственно с аппаратурой МС.

Микросистемам, построенным на основе микропроцессорных комплектов (МПК) младшего поколения, свойственны более простые архитектуры, что было важно для интегральной технологии прошлого десятилетия. Однако вычислительные возможности и быстродействие этих систем, как правило, были низки. Усовершенствование технологических приемов позволило увеличить степень интеграции аппаратуры и перейти к сложным 16-разрядным архитектурам с виртуальной памятью, обеспечивающим параллельную обработку многих задач в реальном масштабе времени [55, 60].

Микросистема состоит из построенного на базе МП *центрального процессора* (ЦП), *основной памяти* для хранения программ и данных, а также *подсистемы ввода-вывода* (ВВ) для связи МС с внешней аппаратурой (рис. 1.1). Задача управления МС возлагается на ЦП, который связан с памятью и подсистемой ВВ через каналы памяти и ВВ соответственно. Центральный процессор считывает из памяти МС команды, которые образуют программу, и декодирует их. В соответствии с результатом декодирования команд он осуществляет выборку данных из памяти МС и *портов ввода*, обрабатывает их и пересылает обратно в память или *порты вывода* подсистемы ВВ. Существует также возможность ВВ данных из памяти на внешние устройства и обратно, минуя ЦП. В этом случае обмен данными выполняется через *канал прямого доступа к памяти* (ПДП) [20, 25, 41], управление которым возлагается на подсистему ВВ. Иногда выделяются *средства поддержки режима реального времени*, в простейшем случае разделяемые процессором и подсистемой ВВ.

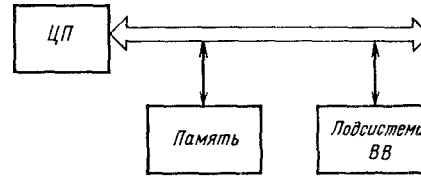


Рис. 1.1. Базовая организация микросистем

Каждый уровень организации МС и любая ее составная часть имеют достаточно сложную внутреннюю структуру, детализация которой приводит к появлению различных типов структур и вычислительных сред. В соответствии с используемым в МС принципом программного управления их организация в значительной степени определяется методологией построения больших вычислительных систем. Однако из-за особенностей производства МПК БИС и их применения организация МС приобрела ряд черт, не свойственных большим ЭВМ.

1.2. Архитектура типовой микросистемы

Основные типы архитектур. В большинстве современных микроЭВМ для хранения программ и данных используется одно пространство памяти. Такая организация получила название *архитектуры Дж. фон Неймана* — по имени математика, предложившего кодирование программ в формате, соответствующем формату данных. Программы и данные хранятся в едином пространстве, и нет никаких признаков, указывающих на тип информации в ячейке памяти. Содержимое ячейки интерпретируется оператором обработки, в качестве которого в простейшем случае выступает ЦП.

Однако почти все однокристальные микроЭВМ, представляющие класс однокристальных программируемых микроконтроллеров, выполнены по другой схеме, известной как *архитектура Гарвардской лаборатории*, в которой память программ CSEG

(Code Segment) и память данных DSEG (Data Segment) разделены и имеют свои собственные адресные пространства и способы доступа к ним. Такое разделение, позволившее реализовать компактно кодируемый набор машинных команд, экономно использовать память программ, и было применено при разработке однокристальных микроконтроллеров первых типов, имеющих всего лишь (1—2) К байт ($К=2^{10}$) управляющей памяти, расположенной на кристалле.

Дальнейшее совершенствование архитектур обоих типов состояло в выделении специального пространства данных небольшого объема, которое сегодня известно как набор *программно-доступных регистров* RSEG (Register Segment). В отличие от CSEG или DSEG регистры RSEG располагаются внутри ЦП в непосредственной близости от его *арифметическо-логического устройства* (АЛУ), что обеспечивает быстрый физический доступ к информации, хранящейся в них. В некоторые интервалы времени программа наиболее интенсивно работает лишь с небольшим объемом данных. Для вре́менного хранения этих данных и предназначена *регистровая область* — набор программно-доступных регисторов.

Область RSEG может быть как полностью изолирована от пространства данных DSEG, так и частично пересекаться с ним, что дает возможность рассматривать отдельные регистры МП как обычные ячейки памяти данных. Такая организация является целесообразной, если в МС поддерживается быстрый доступ ко всей или хотя бы некоторой части памяти данных DSEG.

Почти все современные МС имеют регистровые области независимо от того, к какому типу они принадлежат: неймановскому или гарвардскому. Внутренняя логическая организация RSEG очень разнообразна и играет определяющую роль в классификации архитектур. Функциональная структура регистровой области рассматривается в § 1.4. Пока отметим в ее составе лишь один регистр PC (Program Counter), который называется *программным счетчиком*. Данный регистр является неотъемлемой частью всех МС и связан с адресацией памяти программ. Именно он служит указателем следующего элемента программной последовательности, подлежащей выборке и исполнению.

Система ВВ в простейшем случае представляет набор адресуемых буферных схем и регистров (портов), через которые осуществляется связь с внешними и внутренними аппаратными средствами МС. Система ВВ обычно использует единый механизм адресации портов, размещенных в специальном пространстве ВВ микросистемы IOSEG (Input/Output Segment), логически изолированном от других пространств данных,— *изолированный ВВ*. К МС с изолированным ВВ относятся системы на базе МП КР580ВМ80, К1810ВМ86 и других, имеющих специальные *наборы команд ВВ*.

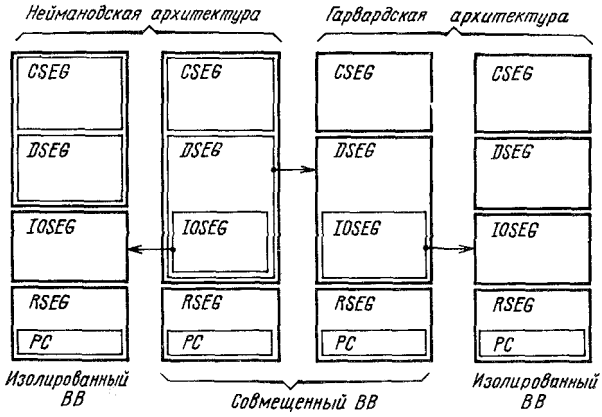


Рис. 1.2. Типовая организация памяти и пространства ввода-вывода

В некоторых системах логически изолированное пространство ВВ может отсутствовать. В этом случае в пространстве памяти данных DSEG выделяются области, в которых и размещаются порты,— *совмещенный ВВ*. Организация доступа к портам в таких МС ничем не отличается от процесса записи-считывания данных в память. Совмещенный ВВ используется в МС на базе МП серии К1801.

На рис. 1.2 представлены четыре типовых набора областей для хранения программ и данных. Стрелкой указан процесс изоляции отдельных областей, приводящий к появлению нового типового набора. Все наборы существуют реально, на их основе созданы те или иные серии микропроцессорных БИС. Каждый тип организации имеет свои преимущества и недостатки, учет которых позволяет создавать высокоэффективные системы различного применения.

Организация пространств памяти и ввода-вывода. В отличие от RSEG память программ CSEG и данных DSEG, а также область ВВ IOSEG организованы проще. В ряде случаев память МС с точки зрения программиста представляет собой линейно упорядоченный набор n-разрядных ячеек с произвольным доступом — *линейная память*. Каждой ячейке набора соответствует число, называемое ее *адресом*. Все адреса занимают целочисленный диапазон от 0 до 2^m-1, который образует *адресное пространство* памяти. Разрядность адреса m обычно равна 16, 18, 20, 24 или 32. В тех случаях, когда наименьшая адресуемая единица — байт ($n=8$), память имеет байтовую организацию.

Одним из примеров МС с памятью линейной организации байтового типа является 8-разрядная система на базе МП КР580ВМ80. Команды этого МП выполняют обращение к пространству памяти емкостью $2^{16}=64$К байт, как показано на рис. 1.3, а. В дальнейшем память МС будет представляться таким

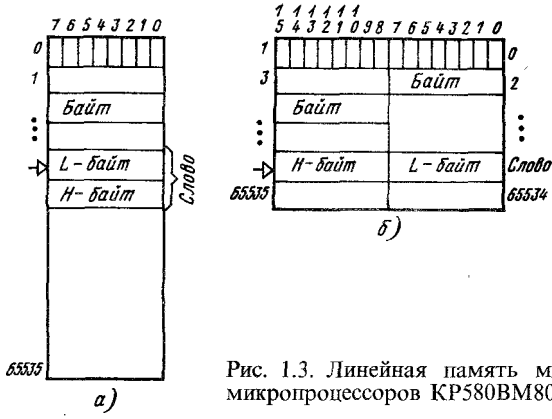


Рис. 1.3. Линейная память микросистем на базе микропроцессоров КР580ВМ80 (а) и К1801ВМ1 (б)

образом, чтобы ячейки со старшими адресами располагались ниже, чем с младшими. Нумерация отдельных разрядов в ячейке производится справа налево начиная с нуля, при этом разряд с нулевым номером является младшим.

При необходимости хранящиеся в памяти *программные объекты* команды и *операнды* (данные к командам) могут располагаться в соседних ячейках пространства памяти. Адресом объекта обычно служит наименьший из адресов ячеек, занимаемых им. Операция обращения к памяти предполагает считывание или запись всего объекта как единого целого. Например, слова в памяти МС на базе МП КР580ВМ80 хранятся в двух соседних байтах. Старшая часть слова занимает байт со старшим адресом, а младшая — байт с младшим адресом. При этом адрес младшего байта служит адресом слова (см. рис. 1.3, а).

Память большинства 16- и 32-разрядных МС также имеет байтовую организацию. Так, нижний уровень логического представления памяти МП К1810ВМ86 емкостью 1 Мбайт ($M=2^{20}$) аналогичен рассмотренному выше. Однако в данном МП существует более высокий уровень организации памяти, на котором в основном и работает программист (см. § 6.3).

Очень часто организация памяти предусматривает определенные ограничения на возможное расположение многобайтовых объектов. Так, в МС на базе 16-разрядного МП К1801ВМ1, которая также имеет память с байтовым доступом емкостью 64К байт, слова в памяти могут находиться только по четным адресам, как представлено на рис. 1.3, б. Тогда при доступе к слову значение младшего разряда его адреса, указывающего на байт в слове, во внимание не принимается, т. е. такая память имеет *границу слов*. Порядок расположения байтов внутри слова стандартный: сначала младший, затем старший байт слова.

Организацию такого вида имеют не только пространства памяти DSEG и CSEG, но и область BB IOSEG.

Командный цикл. Центральный процессор осуществляет ввод, обработку и вывод данных в соответствии с программой, хранимой в CSEG.

Программа — это упорядоченная последовательность команд и данных. Процесс исполнения программы заключается в последовательном выполнении команд, образующих программу.

Команда — это функционально завершенное элементарное действие, которое определяется типом используемых данных, источником их получения, операцией над ними, приемником размещения результата, а также источником получения следующей команды. Программист рассматривает команду как одно неделимое действие. На уровне физического обмена каждая команда представляет собой ряд типовых циклов обращения к системной магистрали.

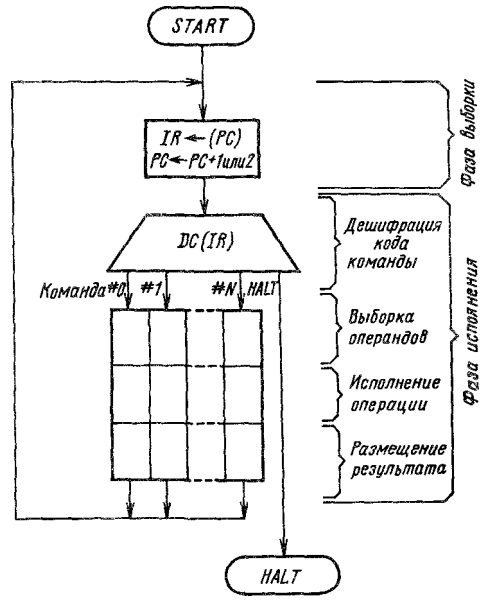


Рис. 1.4. Основные фазы работы микросистемы

Машинное представление команды в памяти МС называется ее *объектным кодом*. Объектный код команды состоит из ряда нулей и единиц. Однако человеку более понятна информация, представленная в символьной форме. Поэтому наряду с объектным кодом каждой команде приписывается ее символическое обозначение, или *мнемокод*, который используется при написании программ человеком с последующей их перекодировкой в машинное представление. Обычно существует взаимно-однозначное соответствие между мнемокодом и объектным кодом команды.

Время, необходимое для выполнения одной команды, называется *командным циклом* (рис. 1.4). Командный цикл делится на две фазы: *выборки* и *исполнения*. Работа ЦП заключается в непрерывном повторении чередующихся фаз командного цикла.

Основное содержание фазы выборки состоит в считывании первого байта (слова) команды из памяти МС и его ввод в специальный *регистр команд* IR (Instruction Register). Считывание байта (слова) происходит по адресу, хранящемуся в программном счетчике PC. Одновременно с этим содержимое PC увеличивается на 1 или 2, указывая на следующий элемент объектного кода. Фаза выборки одинакова для всех команд.

Фаза исполнения состоит в дешифрации содержимого IR и выполнении действий, определяемых этим содержимым. Состав и порядок действий фазы исполнения для каждой команды свой. Она также может включать считывание дополнительных байтов (слов) команды и соответствующего изменения PC, несколько дополнительных обращений к памяти программ и(или) данных для выборки операндов и размещения результата, циклы обращения к портам ВВ IOSEG.

В целом работа МС заключается в следующем. При включении источника питания или нажатии клавиши сброса RESET управление аппаратно передается на *стартовый адрес* памяти программ. Выбирается и исполняется первая команда, по результатам которой управление передается другой и т. д. При приеме специальной команды останова HLT МС приостанавливает свою работу до следующего пуска.

1.3. Структура типовой микросистемы

Магистраль микросистемы. На физическом уровне ЦП взаимодействует с памятью и подсистемой ВВ через единый набор системных шин — *внутрисистемную магистраль* (рис. 1.5), в общем случае состоящую из:

шины данных DB (Data Bus), по которой производится обмен данными между ЦП, памятью и подсистемой ВВ;

шины адреса AB (Address Bus), используемой для передачи адресов ячеек памяти и портов ВВ, к которым осуществляется обращение;

шины управления CB (Control Bus), реализующей функцию управления циклами обмена и работой системы.

Этот же набор шин применяется и для организации канала ПДП. Магистраль такого типа носит название *трехшинной с раздельными шинами передачи адреса и данных*. Примером внутрисистемной магистрали с тремя шинами служит широко распространенная магистраль И41 [33] и ей подобные.

Физическое совмещение линий связи ЦП с памятью и подсистемой ВВ было необходимым в связи с технологической особенностью производства БИС — ограниченным числом физических выводов, допустимых на кристалле. До середины 80-х гг. стандартная микропроцессорная БИС имела около 40 выводов, через которые требовалось связаться как с памятью, так и с подсистемой ВВ.

В некоторых МС с целью дальнейшего сокращения ширины физической магистрали вводят *совмещенную шину адреса/данных* AD (Address/Data Bus), по которой передаются как адреса, так и данные (рис. 1.6). Этап передачи адресной информации отделен по времени от этапа передачи данных и стробируется специальным сигналом ALE (Address Latch Enable), который включен в состав

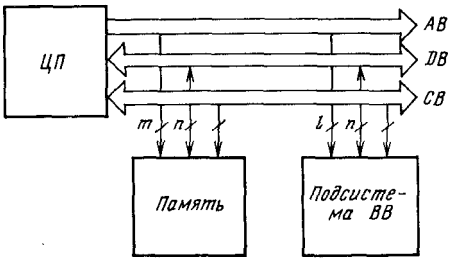


Рис. 1.5. МикроЭВМ с трехшинной магистралью

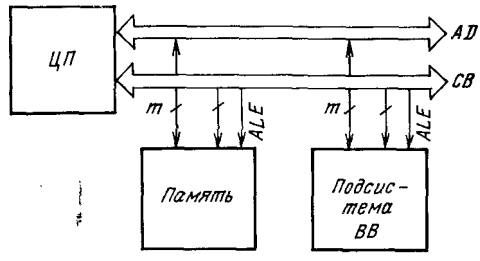


Рис. 1.6. МикроЭВМ с двухшинной магистралью

СВ. Данную магистраль обычно называют *двухшинной с совмещенными шинами передачи адреса и данных*. В качестве примера магистрали последнего типа можно привести внутрисистемную магистраль, формируемую в МП К1821ВМ85А, а также хорошо известную магистраль МПИ (межмодульный параллельный интерфейс) микросистем на базе БИС серии К1801.

Входящий в состав шины управления сигнал ALE используется для разделения функций, выполняемых совмещенной шиной AD. Для этой цели обычно служит срез ALE (переход из состояния 1 в 0), по которому присутствующая на шине AD адресная информация должна быть принята во внешний адресный регистр. При напряжении низкого уровня (0) на линии ALE шина AD выполняет функцию передачи данных.

Обычно каждый модуль МС с двухшинной магистралью, будь то модуль памяти или ВВ, содержит локальный адресный регистр RG, который представлен на рис. 1.7, или другие средства для запоминания адресной информации по срезу ALE. Для фиксации адресной информации может быть использован и один общий регистр (рис. 1.8), в результате ЦП с двухшинной магистралью преобразуется в подобный ему ЦП с тремя раздельными шинами. Построенная таким образом система относится уже к классу трехшинных МС.

Циклы обращения к магистрали. Физический обмен данными через магистраль выполняется словами или байтами в виде следующих друг за другом обращений к каналу. За один *цикл обращения к магистрали* между ЦП, памятью и подсистемой ВВ передается одно слово или байт. Существует несколько типовых циклов обмена. Среди них *чтение памяти и запись в память*. При изолированном ВВ добавляются *чтение порта ВВ* и *запись в порт ВВ*. В случае архитектуры гарвардского типа, когда памяти программ и данных физически разделены, вводится также цикл *чтения памяти программ*. При двухшинной организации МС для увеличения пропускной способности системной магистрали применяется комбинированный обмен типа *чтение-запись в память*, связанный с однократной передачей адреса в начале цикла обмена.

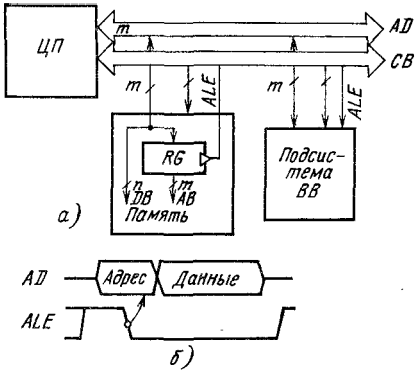


Рис. 1.7. Схема фиксации адреса (а) и временны́е диаграммы ее работы (б)

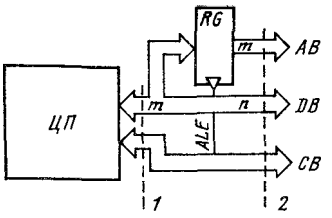


Рис. 1.8. Преобразователь магистрали: 1—двухшинная магистраль; 2—трехшинная магистраль

Рассмотренный выше командный цикл состоит из ряда циклов обращения к магистрали. Если командный цикл образует основное неделимое действие для программиста, то цикл обращения к магистрали является главным звеном работы МС.

В зависимости от числа типовых циклов обращения к магистрали шина СВ должна содержать определенное число линий для передачи сигналов синхронизации и управления доступом, среди которых:

MRDC (Memory Read Command)	Строб чтения памяти
MWTC (Memory Write Command)	Строб записи в память
IORC (Input/Output Read Command)	Строб чтения из порта ВВ
IOWC (Input/Output Write Command)	Строб записи в порт ВВ
PSEN (Program Segment Enable)	Строб чтения программной памяти

Иногда для передачи этих сигналов отводятся отдельные физические линии. В некоторых случаях такой набор сигналов заменяется другим, например:

RD (Read)	Строб чтения
WR (Write)	Строб записи
M/IO (Memory or Input/Output)	Выбор пространства памяти или ВВ
COD (Code)	Выбор памяти программ

Эти сигналы и образуют *физическую шину управления*. Нетрудно перейти от второго набора к первому, используя дополнительную внешнюю логику, входящую в преобразователь магистрали.

Компоненты, расположенные в пространстве памяти, управляются стробами MRDC и MWTC или RD/WR при $M/\overline{IO}=1$. Система ВВ реагирует на стробы IORC и IOWC или RD/WR при $M/\overline{IO}=0$, программная память — на строб PSEN или RD при $COD=1$. Временны́е диаграммы передачи данных через магистраль однотипны и имеют вид, представленный на рис. 1.9.

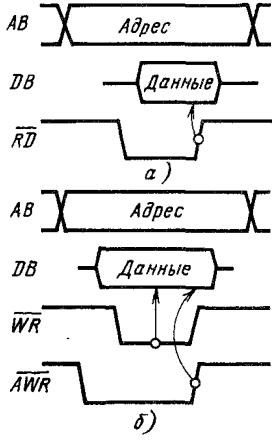


Рис. 1.9. Циклы чтения (а) и записи (б) трехшинной магистрали

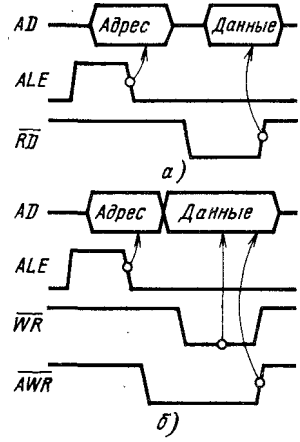


Рис. 1.10. Циклы чтения (а) и записи (б) двухшинной магистрали

Здесь обобщенный строб RD или WR может быть заменен любым другим с соответствующим направлением передачи данных. В случае МС с совмещенной шиной AD в состав шины управления добавляется сигнал ALE (рис. 1.10). Сигналы управления выбором $M/\overline{IO}$ и COD могут рассматриваться как расширение адресной информации, поэтому на них обычно накладываются те же временны́е ограничения, что и на адрес.

На диаграммах рис. 1.9, 1.10 выходные данные истинны в момент окончания строба RD, тогда как формируемые ЦП выходные данные истинны в течение действия сигнала WR. Магистрали с такими характеристиками применяются в простых МС, так как в них для вывода данных используются упрощенные выходные фиксирующие схемы со *статическим входом синхронизации*. Разработанные с расчетом на потенциальную запись периферийные БИС относятся к приборам первого поколения.

Улучшение скоростных характеристик МС потребовало перехода к магистралям, в которых истинность как входных, так и выходных данных гарантируется только в момент окончания стробов RD и WR, генерируемых с упреждением. Это, в свою очередь, усложнило организацию периферийных модулей и привело к необходимости применения выходных фиксирующих схем двухтактного типа или с *динамическим входом синхронизации*.

Стробы с новыми временны́ми ограничениями получили название улучшенных или модифицированных. Для их обозначения будем использовать букву A (Advance), стоящую впереди типового имени. Так, стробы улучшенного типа AWR, AMWC и AIOWC можно считать расширенными по сравнению с типовыми WR, MWTC и IOWC. Периферийные модули, спроектированные для работы с модифицированными стробами,

относятся к усовершенствованным приборам второго поколения [42].

Из представленных на рис. 1.9 и 1.10 временных диаграмм следует, что периферийные модули, такие как память или подсистема ВВ, успевают отреагировать на стробы RD и WR. В этом случае говорят о синхронном режиме работы системной магистрали, который обеспечивает безусловную передачу данных со скоростью, определяемой ЦП.

В некоторых случаях, например когда в МС используются медленно работающие модули или некоторые из модулей еще не готовы к очередному обмену по ряду причин, не зависящих от ЦП, длительности стробов RD и WR могут оказаться недостаточными для правильного выполнения операции обмена со стороны периферийного модуля. Тогда для организации надежного завершения магистральной операции в состав шины управления вводят специальную линию XACK (Exchange Acknowledge), служащую для передачи сигнала подтверждения обмена.

В каждом цикле обращения к каналу перед окончанием строба RD или WR ЦП проверяет линию XACK, контролируя установку сигнала подтверждения обмена. При подтверждении обмена ЦП завершает операцию на магистрали, в противном случае он переходит в *состояние ожидания подтверждения* WAIT, в котором остается до установления сигнала XACK (рис. 1.11). В состоянии WAIT ЦП тестирует линию XACK с периодом, равным, например, периоду следования импульсов CLK, формируемых генератором тактовых импульсов (ГТИ) процессора. Организованный таким образом обмен называется асинхронным режимом работы системной магистрали. На практике встречаются магистрали как с синхронным, так и асинхронным доступом.

Классическая организация магистрали с асинхронным доступом предполагает, что сигнал подтверждения XACK устанавливается в 1 выбранным периферийным модулем во время операции обмена и снимается после ее завершения (пассивная по умолчанию магистраль). Этой ситуации соответствуют временные диаграммы с низкими начальным и конечным уровнями напряжения сигнала XACK (см. рис. 1.11). Классическая организация свойственна большим МС. Для достижения максимальной производительности без состояния WAIT периферийные модули должны успевать возвращать сигнал подтверждения к моменту первой его проверки. Задержки ответа приводят к появлению состояний WAIT и снижению скорости обмена.

Существует еще один вариант организации асинхронного доступа, который предполагает изначальную готовность модулей к синхронному доступу. Этому варианту соответствуют диаграммы на рис. 1.11 с высокими начальным и конечным уровнями напряжения сигнала XACK. Устройство, не отвечающее требованиям синхронного доступа, должно устанавливать сигнал XACK в

0 до момента первого его тестирования, что приведет к переводу ЦП в состояние WAIT и приостановке операции обмена. Такое решение характерно для небольших однопроцессорных МС, так как позволяет максимально сократить объем логики подтверждения обмена.

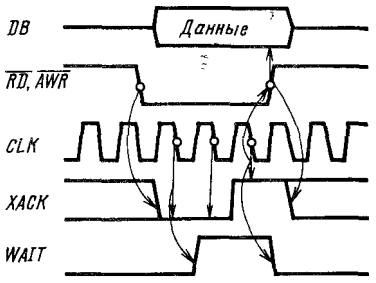


Рис. 1.11. Временны́е диаграммы цикла чтения-записи с подтверждением обмена

Внутри МС или в связанной с ней внешней среде могут возникать разнообразные события, которые требуют ее немедленной реакции на свое появление [18], т. е. вре́менной приостановки процесса обработки основной программы, и выполнения другой, предназначенной для обслуживания возникшего события. Реакция МС на событие завершается возвратом к основной программе. Так как состав и моменты возникновения этих событий заранее не известны, то процессы их обслуживания не могут быть учтены при составлении основной программы и должны выполняться параллельно и скрытно для нее.

Процесс обслуживания событий называется *прерыванием программы*. Каждое прерывающее программу событие сопровождается генерацией специального сигнала IRQi (Interrupt Request), i=0, 1, ..., который называется *радиальным запросом прерывания*, а вызываемые им программы — *программами обслуживания прерываний*. Аппаратные и программные средства, ответственные за организацию прерываний, образуют *систему прерываний*.

Часто в процессе связывания запроса на прерывание с программой его обслуживания участвует не только ЦП, но и периферийные модули МС. В этом случае ЦП получает общий сигнал запроса на прерывание INTR, называемый *векторным запросом* и свидетельствующий о появлении одного или нескольких радиальных запросов IRQi, и переходит к выполнению специального машинного цикла — *цикла ввода вектора прерывания*. Этот цикл состоит во вводе некоторой информации, однозначно связанной с номером i радиального запроса IRQi, которая используется ЦП для перехода к соответствующей программе обслуживания. В общем случае можно считать, что цикл ввода вектора прерываний подобен обычному циклу ввода данных, за исключением того, что сигнал IORC заменяется стробом *подтверждения прерывания* INTA (Interrupt Acknowledge). Это еще один типовой сигнал, входящий в состав шины СВ.

Прерывания изменяют общее состояние МС и могут влиять на ход выполнения основной программы. Однако их влияние не является прямым, так как осуществляется через размещаемые в памяти структуры данных — *области взаимодействия*. Основная программа взаимодействует с программами обслуживания пре-

рываний, считывая и модифицируя данные в этих областях. В остальном она создается и выполняется независимо от программ обслуживания, при этом прерывания рассматриваются как параллельно развивающийся изолированный процесс, контролирующий состояние области взаимодействия. Взаимодействие основной программы с процессом прерываний, связанное с получением внешней информации, подобно ВВ данных, поэтому система прерываний может рассматриваться как составная часть системы ВВ [4, 39, 40].

К рассмотренным выше циклам обращения к магистрали обычно добавляют еще два: *пуска* и *останова*, обеспечивающие запуск и соответственно останов МС. Они реализуют граничные точки основного алгоритма работы МС, представленного на рис. 1.4.

Типовые структуры МС. Память и подсистема ВВ включают отдельные функционально законченные модули, состав и организация которых раскрывают структуру соответствующей подсистемы. В представленной на рис. 1.12, а МС магистрального типа эти модули выполняются так, чтобы имелась возможность их подключения непосредственно к единой внутрисистемной магистрали. По такому *магистрально-модульному принципу* построено большинство современных МС.

В подсистеме памяти можно выделить модули *постоянного запоминающего устройства* (ПЗУ), используемые для хранения программ и констант. Они являются важнейшими компонентами организации такого класса МС, как программируемые микроконтроллеры (МК). При этом емкость ПЗУ может быть достаточно большой. Ко второму типу входящих в подсистему памяти стандартных модулей относятся *оперативные запоминающие устройства* (ОЗУ), предназначенные для хранения переменных и загружаемого извне объектного кода. Микроконтроллеры обычно имеют ОЗУ незначительных размеров, тогда как микроЭВМ

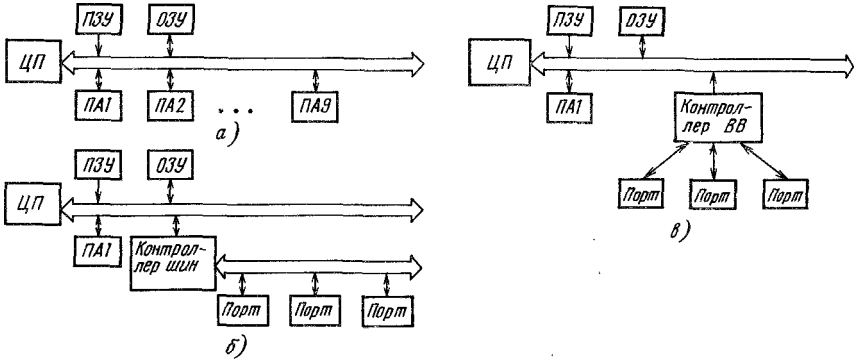


Рис. 1.12. Типовые структуры МС:
а — магистральная; б — магистрально-каскадная; в — магистрально-радиальная

общего назначения должны иметь ОЗУ достаточно большой емкости.

В составе подсистемы ВВ также можно выделить ряд функционально законченных устройств, которые оформляются в виде модулей, подключаемых непосредственно к единой магистрали МС. В простейшем случае это адресуемые ЦП буферные схемы и регистры подсистемы ВВ — порты. Более сложные программно-управляемые подсистемы ВВ, содержащие блоки портов, получили название *периферийных адаптеров* (ПА). В случае, когда средства ВВ предназначаются для управления специальным внешним оборудованием и реализации специальных функций ВВ, их называют *периферийными контроллерами.*

Наиболее сложными из современных средств обмена с внешними устройствами ВВ считаются *сопроцессоры ВВ* [58], которые работают по собственным программам, хранящимся в памяти МС, и по сути дела представляют собой отдельные МС.

Модули подсистем памяти и ВВ, реализуемые по магистральной или радиальной схеме, образуют *магистрально-каскадные* или *магистрально-радиальные структуры* МС, приведенные на рис. 1.12, б, в. В состав этих структур включаются специальные *контроллеры шин*, основное назначение которых — реализовать приоритетные соотношения при использовании магистрали.

В каждый момент времени на магистрали допускается только один активный модуль, в распоряжение которого отдаются все ресурсы магистрали. В простейших системах роль активного модуля всегда выполняет ЦП, который и организует управление магистралью. В более сложных системах со многими активными модулями (параллельно работающими другими ЦП, арифметическими или периферийными сопроцессорами, контроллерами ПДП) магистраль распределяется между ними в соответствии с последовательностью запросов на захват магистрали и приоритетными соглашениями. Эта задача возлагается на арбитра системной магистрали. В системах средней сложности функции арбитра выполняет МП.

Использование в МС единой магистрали обеспечивает выполнение за один рабочий цикл одной операции обмена данными между двумя (в общем случае любыми) модулями системы, например чтения командного слова или слова данных из памяти в МП, выдачи данных из процессора в порт подсистемы ВВ или наоборот и т. д.

1.4. Регистры микропроцессора

Функциональная неоднородность регистров. Регистровую область или набор программно-доступных регистров RSEG можно рассматривать как скоростное ОЗУ малой емкости, входящее в состав МП. Набор используется для временного хранения данных

и адресной информации, контролируемых программистом. Число регистров колеблется от 4 до 32. Короткая адресация регистровой области и быстрый доступ к ней обеспечивают создание эффективно исполняемых программ.

Регистры МП функционально неоднородны: одни служат для хранения данных или адресной информации, другие — для управления работой ЦП. В соответствии с этим все регистры можно разделить на регистры данных, указатели и регистры специального назначения. *Регистры данных* участвуют в арифметических и логических операциях в качестве источников операндов и приемников результата, *адресные регистры* или *указатели* используются для вычисления адресов данных и команд, расположенных в основной памяти. *Специальные регистры* служат для индикации текущего состояния ЦП и управления режимами его работы.

Функциональная специализация регистров определяется системой команд. При слабой специализации часть регистров обычно применяется для хранения как операндов, так и адресов. Их называют *регистрами общего назначения* (РОН).

Функциональная неоднородность RSEG связана с широким использованием *неявной* (подразумеваемой) *адресации* регистров, которая, в свою очередь, определяется стремлением к созданию коротких программ. Функциональная специализация затрудняет программирование, так как требует учета особенностей организации регистрового набора, присущих данному МП. Однако в результате объектный код команд исполняется быстрее и для его хранения требуется меньшая память.

На уровне символического кодирования команд для прямой ссылки на конкретные регистры МП им присваиваются имена, например A, B, C, D, SP, X, WP или R0, R1, R2 и т. д. Обычно эти имена отражают функциональное назначение регистра и способствуют пониманию символьных команд.

Адресные регистры. Ряд функций, которые возлагаются на RSEG, распределяются между регистрами МП. Так, для реализации различных методов *непрямой* (вычисляемой) *адресации* данных в составе блока регистров применяются *адресные регистры-указатели. Регистр косвенного адреса* DP (Data Pointer) содержит непосредственно адрес операнда, *регистр базы* BP (Base Pointer) используется для хранения начальных адресов массивов и записей, содержимое *индексного регистра* X (Index) является относительным адресом (индексом) операнда (рис. 1.13). Среди адресных регистров следует также отметить *регистры автоинкрементной* и *автодекрементной адресации*, которые автоматически увеличивают или уменьшают свое состояние до или после выполнения операции доступа в соответствии с длиной адресуемого ими операнда, и *регистры расширения адресного пространства* или *указатели сегментов и страниц*. (Подробно регистры данного типа будут рассмотрены в гл. 6.)

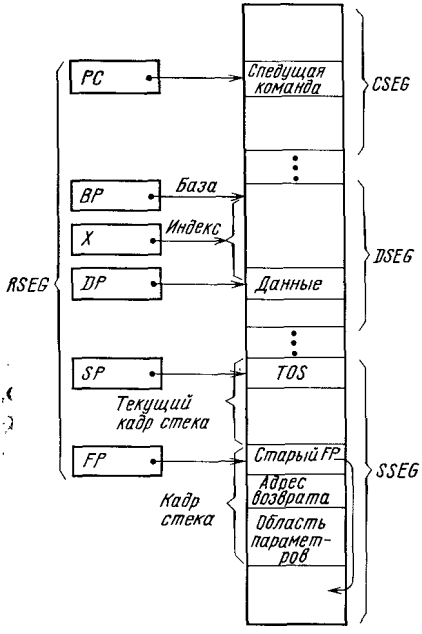


Рис. 1.13. Адресные регистры МС

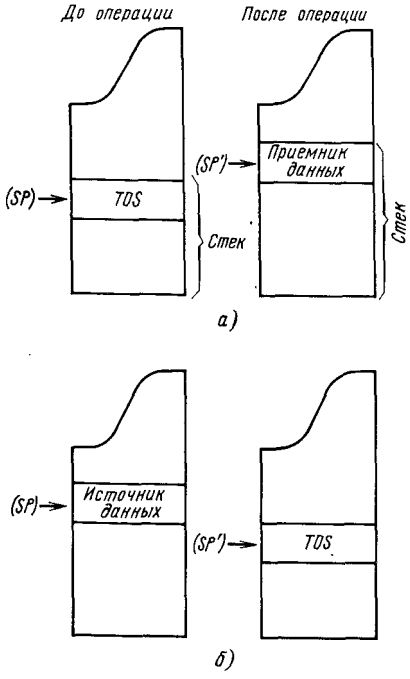


Рис. 1.14. Операции PUSH (а) и POP (б) типового стека

Очень важной является функция хранения адреса следующего подлежащего выборке элемента программной последовательности. Эту функцию выполняет *программный счетчик* PC. Бóльшая часть команд выполняется последовательно в порядке возрастания адресов памяти. Во время выборки очередной команды содержимое PC увеличивается на 1 или 2 для указания следующего байта или слова в последовательности команд. Процесс адресации следующего элемента командной последовательности, как правило, осуществляется неявно. В типовой команде, например,

```
MOV  dst,src          ;dst←src
```

где dst и src — указатели приемника и источника данных, какие-либо сведения об адресе следующей команды отсутствуют. Поэтому функция PC возлагается на конкретный регистр, содержимое которого автоматически инкрементируется после очередной выборки элемента командной последовательности. Изменение последовательности процесса выборки команд осуществляется специальными командами передачи управления, связанными с загрузкой PC адресом, отличным от адреса следующей команды.

В системах с предварительной выборкой команд каждый переход также связан с очисткой буфера предварительной вы-

борки. Регистр PC относится к классу указателей с автоинкрементированием.

Частным случаем регистра с автомодификацией является *указатель стека* SP (Stack Pointer). Он необходим для организации системного стека SSEG, который предназначен для хранения адресов возвратов и состояний процессора при вызове подпрограмм и обслуживании прерываний. Стек может также использоваться для временного хранения локальных переменных и передачи входных или выходных параметров при вызовах подпрограмм.

Стек — это область памяти с доступом типа «последний пришел — первый вышел» или LIFO (Last Input — First Output). Стек обычно заполняется в сторону уменьшения адресов, при этом указатель стека показывает на последнюю заполненную ячейку стека — *вершину стека* TOS (Top of Stack). Такой стек называется типовым, так как именно он применяется в большинстве МС. При записи в стек нового элемента данных (операция PUSH) содержимое SP уменьшается на 1 или 2 в зависимости от длины элемента (байт или слово) и затем используется в качестве адреса новой вершины, в которую заносится элемент. При считывании элемента данных из стека (операция POP) сначала считывается содержимое TOS, а затем содержимое SP увеличивается на 1 или 2 для адресации новой вершины стека. Работа стека показана на рис. 1.14. Исключительные удобства, предоставляемые стеком при вызове подпрограмм, привели к тому, что практически все современные МП имеют средства для его построения в виде SP.

При использовании стека для хранения локальных переменных и обмена параметрами между вызываемой и вызывающей процедурой может оказаться полезным специальный адресный регистр, указывающий на начало области параметров в стеке (см. рис. 1.13). Регистр с таким функциональным назначением называется *указателем кадра* FP (Frame Pointer). Действительно, значение SP непрерывно меняется, поэтому применять его в качестве точки отсчета при доступе к данным в стеке крайне неудобно. Процедуру доступа можно значительно упростить, если функцию точки отсчета отдать специально зарезервированному для этой цели указателю FP, который принадлежит к классу базовых регистров.

Специальные регистры. При выполнении команд АЛУ генерирует ряд сигналов-признаков, характеризующих результат операции. Функцию хранения этих сигналов, а также некоторых других выполняет специальный регистр слова состояния программы PSW (Program Status Word). С каждым признаком результата операции связывается одноразрядная переменная — *флажок*. Флажки группируются во входящее в состав PSW поле кода условия CC (Condition Code). Типовой состав флажков-признаков результата операции $F = F_{n-1}F_{n-2}\ldots F_1F_0$ следующий:

CF (Carry Flag)	Флажок переноса из старшего разряда АЛУ. Флажок CF используется наиболее часто. При арифметических операциях $CF=C_n$, при сдвигах $CF=F_{-1}$ или F_n выдвинутое при операциях значение младшего или старшего разряда
ZF (Zero Flag)	Флажок признака нуля, $\overline{ZF}=\bigvee_{i=0}^{n-1} F_i$
SF (Sign Flag)	Флажок знака результата, $SF=F_{n-1}$. Применяется при целочисленной арифметике со знаком
AF (Auxiliary Carry Flag)	Флажок дополнительного переноса или переноса из младшей тетрады, $AF=C_4$
OF (Overflow Flag)	Флажок арифметического переполнения. При выполнении арифметики в дополнительном коде $OF=C_n \oplus C_{n-1}$
PF (Parity Flag)	Флажок четности результата, $\overline{PF}=\bigoplus_{i=0}^{n-1} F_i$

Здесь C_i — входной перенос в i-й разряд, который одновременно является выходным переносом из (i−1)-го разряда сумматора АЛУ.

Удобно, если поле условия CC содержит один или несколько флажков пользователя U0, U1, ..., функциональное назначение которых определяет он сам. Обычно эти флажки служат для связи между отдельными частями программы. Состояние поля условия CC тестируется различными *командами условного типа.*

В состав PSW входит также ряд специальных флажков, управляющих работой МП:

IF (Interrupt Flag)	Маски и приоритеты прерываний, а также условия реакции на прерывания
TF (Trace Flag)	Флажок пошаговой трассировки, маска специального прерывания

Регистр PSW включает и различные модификаторы команд (например, флажок направления DF в МП К1810ВМ86), изменяющие реакцию МП на отдельные команды. Для обеспечения особых условий выполнения программ в PSW вводят специальный флажок, определяющий эти условия. Так, флажок H/$\overline{U}$ (Halt/User) в МП К1801ВМ2 служит для разрешения особых условий выполнения программ и реализации директив пультового терминала.

Упаковка всех флажков в одно слово дает возможность организовать их быструю засылку в память с последующим восстановлением. В некоторых сложных МП специальных регистров может быть несколько.

Регистры данных. Наиболее типичным представителем регистров данных является *аккумулятор* A (Accumulator), который используется для временного хранения исходных операндов и промежуточных результатов. С аккумулятором связано большинство команд арифметической и логической обработки. Ссылка на него, как правило, производится неявно с помощью кода операции:

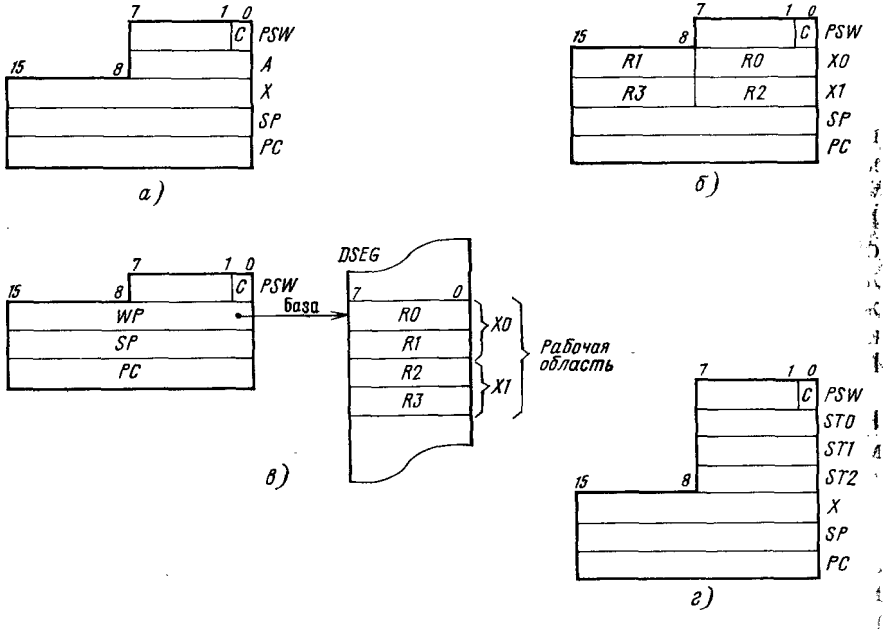


Рис. 1.15. Типовые составы RSEG:
а — с аккумулятором; б — с регистрами общего назначения; в — с рабочими областями; г — с вычислительным стеком

```
LDA     src             ;A←src
```

Неявная адресация позволяет не указывать в командах местоположения одного из операндов src и (или) результата операции dst, что существенно уменьшает длину их кода:

```
ANA     src             ;A←A AND src
```

Это очень важно в условиях ограниченной пропускной способности системной магистрали, особенно для 8-разрядных МС. Поэтому большинство первых МП, например КР580ВМ80, имели аккумулятор и были ориентированы на его интенсивное использование.

В составе МП может находиться один или несколько аккумуляторов. Так, в МП MC6809 два аккумулятора — A и B. Они имеют одноадресную систему команд, так как в коде команды явно указывается адрес лишь одного операнда. При этом предполагается, что источником другого операнда служит аккумулятор. Он же, как правило, применяется и для хранения результата операции:

```
ADDA    src             ;A←A+src
SUBB    src             ;B←B−src
```

Влияние аккумулятора на организацию МС настолько велико, что включение A в состав RSEG стало типичным, поэтому системы с аккумулятором выделили в отдельный класс. На рис. 1.15, а представлен минимальный набор программно-доступных регистров 8-разрядной МС аккумуляторного типа, в состав которых кроме аккумулятора входят индексный регистр X, указатели SP и PC, а также PSW с одним флажком переноса CF. Наличие полноразмерного индексного регистра в составе RSEG объясняется его функциональной универсальностью при адресации данных. Действительно, полноразмерный индексный регистр может выполнять функции BP, а при нулевом значении прямого компонента, задаваемого непосредственно в команде,— функции DP.

Другим примером регистров данных служат *рабочие регистры* R0, R1,... В отличие от аккумулятора они адресуются явно и могут интерпретироваться как сверхскоростные регистровые ОЗУ данных. Рабочие регистры используются в операциях как совместно с аккумулятором, так и без него. Некоторые рабочие регистры совмещают свою функцию хранения данных с функцией их адресации. В этом случае они приобретают функции РОН (МП К1801ВМ1). При необходимости для образования полноразмерного указателя регистры данных объединяются в пары. Ориентация архитектуры на рабочие регистры приводит к полутораадресной системе команд:

```
MOV     dst,reg         ;dst←reg
OR      reg,src         ;reg← reg OR src
```

В полутораадресных командах допускается только один операнд, хранящийся в памяти МС, тогда как другой должен находиться в регистровой области. Для двухместных операций приемником результата, как правило, является регистр—источник одного из операндов. Типовой минимальный набор RSEG для 8-разрядной МС с регистрами общего назначения приведен на рис. 1.15, б.

В ряде МП, предназначенных для работы в реальном масштабе времени, предусмотрены не один, а два (К1816ВЕ48) или даже четыре (К1816ВЕ51) набора рабочих регистров, один из которых резервируется для системных целей или обработки прерываний, а все остальные—для прикладных задач пользователя. В каждый момент времени доступен только один набор рабочих регистров, выбираемый специальным указателем WP (Work Pointer). Переключение доступного набора связано с перезагрузкой малоразрядного указателя WP.

Расширение разрядности указателя WP до полного размера адресного регистра и отображение набора рабочих регистров на основную память данных с базой WP приводит к типовой архитектуре с *рабочими областями* (см. рис. 1.15, в). Однако при передаче функции промежуточного хранения данных, выполняе-

мой ранее RSEG, DSEG, теряется быстрый доступ к промежуточным данным. Решение этой проблемы связано с реализацией части основной памяти данных на одном кристалле с ЦП и размещением рабочих областей в этом внутреннем сегменте ОЗУ.

Передача функции аккумулятора вершине стека TOS приводит к *стековым архитектурам*. Стековая организация МС дает возможность построить безадресные машины, объектный код которых имеет наименьшую длину. *Безадресные команды* стековой архитектуры оперируют элементами, находящимися на вершине стека и непосредственно под ней:

```
DEC                    ;PUSH(POP – 1)
ADD                    ;PUSH(POP + POP)
```

При выполнении операций исходные операнды извлекаются из стека, а результат передается на вершину TOS. Стековая архитектура обладает высокой вычислительной эффективностью [41], однако в классической форме она большого практического распространения не нашла. Это объясняется тем, что стек обычно размещается в основной памяти, доступ к которой требует отдельного цикла обращения к системной магистрали.

Для уменьшения времени доступа к стеку он должен быть физически приближен к АЛУ, например, за счет реализации на одном кристалле с ЦП внутренней памяти данных и размещении в нем стека. Совмещение стека с внутренней частью памяти может привести к сокращению разрядности указателя SP.

В другом случае в состав архитектуры вводят специальный стек, который размещается в регистровой области МП и используется исключительно для промежуточного хранения данных. Он обладает быстрым доступом и называется вычислительным. Глубина вычислительного стека невелика — составляет 3—8 машинных слова. Прямой доступ к содержимому указателя вершины вычислительного стека, как правило, отсутствует. Манипуляция содержимым указателя выполняется только через запись-считывание данных из стека. Примерный состав RSEG 8-разрядной МС с вычислительным стеком из трех регистров A, B, C приведен на рис. 1.15, г. Регистровые наборы такого типа можно встретить только в самых современных архитектурах МС [5].

Регистровые наборы микропроцессоров, выпускаемых промышленностью. Организация RSEG реальных МП отличается от приведенных выше типовых составов. Выпускаемые промышленностью МП, как правило, результат компромиссного выбора между несколькими типами организаций и, следовательно, могут быть отнесены к тому или иному классу лишь условно. Рассмотрим несколько примеров таких МП.

Приведенные на рис. 1.15 типовые наборы являются минимальными. Как правило, они подвергаются дальнейшему расширению в соответствии с теми или иными практическими соображениями.

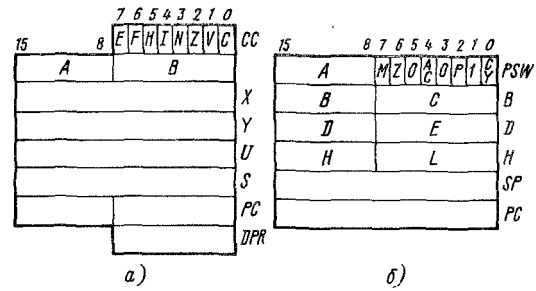


Рис. 1.16. Регистры микропроцессоров MC6809 (а) и 8080/85A (б)

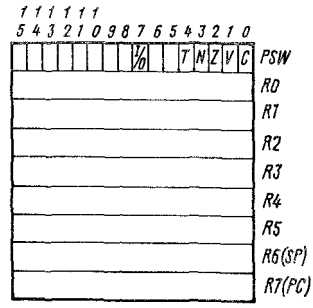


Рис. 1.17. Регистры микропроцессоров К1801ВМ1/ВМ2

Так, расширение набора с аккумулятором путем прямого дублирования его состава приводит к регистрам МП MC6809 фирмы Motorola [41], который представлен на рис. 1.16, а. В их составе два указателя стека S, U, два индексных регистра X, Y и два аккумулятора A, B. Набор дополнен также регистром выбора страницы DPR. Значительно расширен состав PSW. Другой путь увеличения аккумуляторного набора, выбранный фирмой Intel, привел к МП типа 8080/85A (рис. 1.16, б). К минимальному составу был добавлен блок из четырех 8-разрядных РОН B, C, D, E, образующих попарно два 16-разрядных адресных указателя B и D. Расширились и функции типового индексного регистра H, состоящего из двух 8-разрядных регистров H и L. Таким образом, оба МП следует отнести к классу приборов аккумуляторного типа, но с расширенными возможностями.

Архитектура ЦП однокристальных микроЭВМ типов К1816ВЕ48/ВЕ51 тоже имеет элементы организации аккумуляторного типа, которая расширена набором из восьми рабочих регистров. Если в приборе К1816ВЕ48 [61] предусмотрено два набора, то в К1816ВЕ51 [64] их уже четыре. Два первых регистра в наборах являются РОН и, следовательно, могут быть использованы для адресации данных в ОЗУ. Наборы располагаются в области внутренней памяти данных, как это предусмотрено архитектурой с рабочими областями. Однако их размещение в памяти данных строго регламентировано, поэтому для выбора набора используется одно- или двухразрядный указатель BS, входящий в состав PSW. Существует также ряд ограничений на размещение системного стека, который находится во внутреннем ОЗУ. (Более подробно организация однокристальных микроЭВМ данного типа рассмотрена в гл. 4.)

К МП с РОН принадлежат К1801ВМ1/ВМ2 [61] и им подобные, совместимые по архитектуре с ЦП мини-ЭВМ PDP-11. Практическая реализация 8-разрядных архитектур с РОН затруднена из-за сложностей, возникающих при кодировании системы

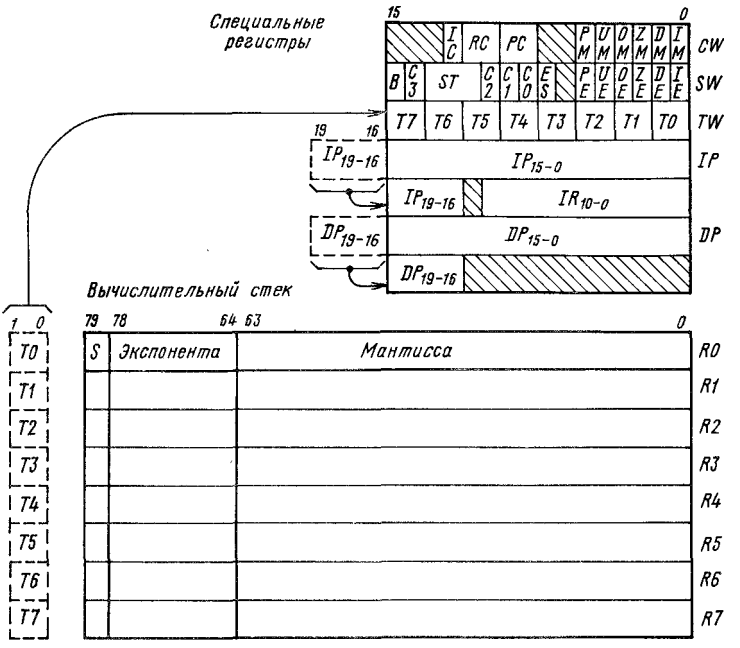


Рис. 1.18. Регистры сопроцессора 8087

команд. Поэтому архитектуры такого типа реально существуют только для 16-разрядных систем. На рис. 1.17 представлен базовый состав RSEG микропроцессоров К1801ВМ1/ВМ2, содержащий восемь 16-разрядных РОН R0—R7. Все регистры предназначены для хранения и данных, и адресной информации. Два старших выполняют также функции указателей SP и PC.

Применение РОН дает возможность создавать регулярную систему команд, большинство из которых используют любой регистр многими способами. Поэтому ряд 16- и 32-разрядных МП был ориентирован на широкое применение РОН. К их числу относятся такие МП, как Z8001/02 фирмы Zilog [41].

Как уже отмечалось, стековая организация еще не получила широкого распространения в промышленных приборах. Однако уже можно назвать некоторые процессоры, имеющие вычислительные стеки. Это прежде всего сопроцессор числовых данных 8087 (рис. 1.18) [61], блок рабочих регистров которого организован дополнительно в виде стека из восьми 80-разрядных элементов. Вершина стека адресуется с помощью 3-разрядного указателя ST, входящего в состав слова состояния SW. Все элементы рабочей области могут адресоваться либо явно как регистры ST0—ST7, отсчитываемые от вершины стека, либо неявно с помощью указателя ST. Программист может условно разделить регистро-

вый блок на вычислительный стек и стандартные рабочие регистры, используя различные способы доступа к ним. Кроме рабочих регистров в RSEG числового сопроцессора входят указатели, регистры управления и состояния.

Приведенных примеров вполне достаточно, чтобы показать, как разнообразны по своей структуре регистровые наборы МП, выпускаемые промышленностью. Однако несмотря на это многообразие все же удается найти возможность для их сравнения и оценки с помощью рассмотренных в данном параграфе типовых наборов RSEG.

1.5. Адресация данных

Представление адресной информации. Источниками и приемниками операндов команд служат регистровая память RSEG, память данных DSEG, а также порты ВВ IOSEG. В дальнейшем для обозначения какого-либо элемента регистровой области будем использовать символ reg, адреса системной памяти — addr, порта ВВ — port. Адрес операнда, считываемого или размещаемого в одном из пространств МС, называется исполнительным.

Исполнительные адреса строятся по сведениям, заложенным в командах. Метод построения исполнительного адреса называется способом адресации (mod). Примером одного из наиболее простых способов адресации служит прямое указание полного исполнительного адреса в команде. Однако такой способ адресации, обычно называемый прямым, оказывается неэффективным во многих случаях. В частности, при прямой адресации манипуляция исполнительными адресами связана с модификацией программы, что не всегда удобно, а иногда и невозможно.

Для повышения эффективности адресации операндов разработан ряд способов задания исполнительного адреса. Каждый способ эффективен только в конкретной ситуации размещения данных. Правильный выбор и использование всего набора позволяет обеспечить эффективный доступ к структурированным данным, таким как массивы, стеки, списки, а также перемещаемость программ и данных на этапе загрузки и выполнения, сократить длину программного кода и число обращений к магистрали, адресовать большую основную память в условиях малой разрядности ЦП.

Необходимость адресовать память большой емкости при коротком слове данных придает проблеме адресации в МС особо острый характер. Поэтому механизмы адресации памяти в МС оказываются наиболее совершенными и требуют минимального числа обращений к магистрали.

Сведения об адресе операнда могут быть заложены в команду в двух формах: *явной* и *неявной*. Явная форма предполагает наличие в команде специального поля, называемого *адресной*

частью команды, в котором эти сведения содержатся. Неявная форма обеспечивает передачу адресной информации через *именную часть команды* или *код операции* (ОР), т. е. сама операция несет определенные сведения о месте нахождения одного или всех операндов, используемых в ней.

Явная форма представления адресной информации более гибкая. Она позволяет связывать с одной операцией множество адресов, что повышает ее эффективность. Однако эта форма требует увеличения длины машинной команды, что снижает эффективность объектного кода. Неявная форма связывает часть адресной информации с операцией, сужая область ее действия, но длина машинной команды при этом существенно уменьшается. В реальных командах, как правило, применяются обе формы представления адресной информации. Широко распространена и такая форма представления адресных сведений: часть информации о месте нахождения одного и того же операнда передается в неявной форме, остальная информация представляется в адресном поле в явной форме.

В общем случае кодирование адресной информации операнда в команде сводится к указанию способа адресации mod и числовой информации, используемой при этом способе. Способ адресации как часть адресной информации может кодироваться в явной и неявной формах. Общепризнанной является явная форма представления mod, при которой в адресной части машинной команды резервируется специальное поле, а на уровне символьного кодирования — ряд специальных суффиксов, префиксов и указателей в поле операндов: #, @, %, (...), (...)+, −(...) и т. д. Например:

```
.disp       ;oper=(PC+disp)
(reg)+      ;oper=(reg),reg←reg+1
@addr       ;oper=((addr))
#data       ;oper=data
```

При неявной форме кодирования способ адресации передается ОР, тогда как дополнительная информация представляется явно. Примерами такого представления служат команды

```
STAX    reg         ;(reg)←A
MVI     reg, data   ;reg←#data
BR      disp        ;PC←PC+disp
```

Они эквивалентны следующим командам:

```
ST      A,@reg
MOV     reg,#data
JMP     .disp
```

Однокомпонентные способы адресации. Самым простым способом задания места расположения данных является включение его полного адреса в состав команды (рис. 1.19, а). Применительно к основной памяти данных МС такой способ адресации называется

прямым (ptr=addr). Прямая адресация дает возможность явно указать на любую конкретную ячейку памяти во всем адресном пространстве DSEG. Примером команды с прямой адресацией служит

```
INC     addr
;addr←(addr)+1
```

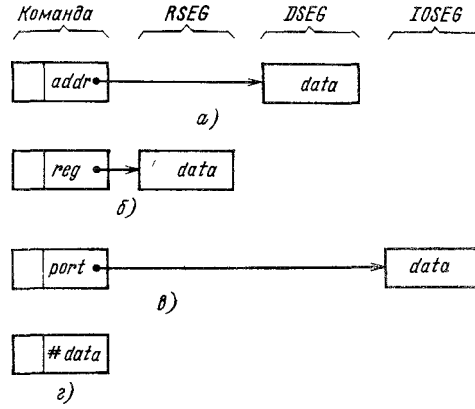


Рис. 1.19. Способы адресации данных:
а - прямая; б — регистровая; в — прямая адресация портов; г - непосредственная

Для явного задания полного m-разрядного адреса в команде должно быть выделено m разрядов. С целью сокращения числа разрядов во многих системах используется короткий вариант прямой адресации, обеспечивающий непосредственный доступ к ограниченной, заранее определенной области памяти. Как правило, это наиболее важная нижняя и (или) верхняя часть адресного пространства. *Короткая прямая адресация* может быть достаточно эффективной в МС гарвардского типа.

Неявная форма представления прямой адресации приводит к указанию конкретной ячейки системной памяти. Такой способ адресации используется крайне редко.

Основной недостаток прямой адресации состоит в том, что манипуляция адресом связана с изменением объектного кода, которое не всегда возможно. Например, в МС гарвардского типа доступ к CSEG с целью модификации объектного кода логически запрещен. В МС неймановского типа доступ к кодовому сегменту может быть тоже запрещен, но уже на физическом уровне, так как исполняемая программа записана в ПЗУ. В любом случае модификация объектного кода нежелательна, что связано со значительными трудностями при его анализе, например, с целью отладки. Использование прямой адресации ограничено случаем, когда расположение данных в памяти известно заранее и не изменяется в процессе исполнения программы. Для МС гарвардского типа могут быть предусмотрены специальные варианты доступа к CSEG с прямой адресацией для считывания констант и постоянных таблиц, которые могут храниться в ней. При этом выбор кодового сегмента кодируется неявно. Применительно к регистровой памяти вариант прямого включения адреса в команду приводит к *регистровой адресации* (ptr=reg) (рис. 1.19, б). Для прямой ссылки на регистр, входящий в состав небольшой регистровой памяти МС, обычно требуется несколько бит. Кроме того, отсутствуют циклы обращения к системной магистрали. Все это обеспечивает большую эффективность регистровой адресации.

Распределение данных по регистрам легко контролируется программистом и не вызывает особых затруднений при ссылке на них. Приведем примеры команд с регистровой адресацией:

```
MOV     reg,addr        ;reg←(addr)
NEG     reg             ;reg←−reg
```

В МС широко распространена *неявная форма регистровой адресации*, когда код операции подразумевает вполне конкретный регистр, например аккумулятор А:

```
LDA     addr        ;эквивалентно MOV A,addr
ANA     reg         ;эквивалентно AND A,reg
```

На уровне мнемокода регистровая адресация отличается от прямой использованием специальных имен, зарезервированных для обозначения регистров A,X,Ri и т. д. В некоторых системах вводится специальный символ регистровой адресации %:

```
CLR     %n              ;Rn←0
```

Здесь *n* — целое, указывающее на конкретный регистр области. Обычно регистровая адресация — единственный способ доступа к пространству регистров МС.

Прямая адресация портов (ptr=port) используется для адресации изолированного пространства портов ВВ (рис. 1.19, в). Доступ к портам обычно задается неявно кодом операции:

```
IN      port            ;A←port
OUT     port            ;port←A
```

Иногда для этой цели применяют резервные имена портов:

```
ANL     Pn              ;Pn←A AND Pn, n=0−7
```

Очень часто прямая адресация портов является единственным способом доступа к пространству ВВ. Встречается также вариант *прямой короткой адресации портов*, например, в системе команд К1810ВМ86.

В некоторых случаях источником данных служит сама команда (рис. 1.19, г). Тогда говорят о *непосредственной адресации* (prt = =#data), которая позволяет задавать константы как часть команды:

```
MOV     reg,#data       ;reg←data
ADD     reg,#data       ;reg←reg+data
```

Для более эффективного кодирования операций с часто используемым непосредственным операндом — *литералом* — обычно вводят специальные коды с *неявным определением данных*:

```
CLR     reg             ;reg←0
INC     reg             ;reg←reg+1
```

или варианты короткой непосредственной адресации:

```
INC     reg,#sdata      ;reg←reg+sdata
```

Здесь sdata может восприниматься как целое без знака или как целое со знаком. В любом случае перед выполнением операции sdata расширяется с учетом знака до полноразмерного операнда. Основной недостаток непосредственной адресацией состоит в отсутствии возможности манипуляции данными без модификации объектного кода, что привело к созданию ряда способов с *вычисляемым адресом*. Простейшим среди них является *косвенная адресация* (ptr = @addr или (addr)), схема выполнения которой показана на рис. 1.20, а. В командах с данным типом адресации используется не прямой адрес, а косвенный, указывающий на ячейку памяти с адресом операнда. Манипуляция исполнительным адресом сводится к изменению содержимого этой ячейки памяти и, следовательно, никак не связана с объектным кодом. К недостаткам косвенной адресации следует отнести двойное обращение к основной памяти и большую разрядность поля, отводимого под косвенный адрес.

От вышеуказанных недостатков косвенной адресации свободен ее вариант (рис. 1.20, б), в котором в качестве области, содержащей прямой адрес, используется не основная память данных, а регистровая. При этом все другие свойства, характерные для косвенной адресации, сохраняются, что объясняет широкое применение *регистровой косвенной адресации* (ptr = @reg или (reg)) практически во всех МС.

Для хранения прямого адреса требуется регистр достаточной длины. В ряде МС аккумуляторного типа для этой цели предусматриваются специальные адресные регистры. В МС с РОН регистры данных при необходимости объединяются в пары для образования адресных регистров. Примерами команд с косвенной регистровой адресацией являются

```
CLR     @reg        ;(reg)←0
INC     @reg        ;(reg)←(reg)+1
```

Пример неявного кодирования данного способа адресации можно привести из системы команд МП КР580ВМ80:

```
LDAX    B           ;A←(BC)
STAX    D           ;(DE)←A
```

Косвенная регистровая адресация портов (ptr = @reg или (reg)) через регистр может быть полезна при доступе к пространству IOSEG (рис. 1.20, в). Благодаря этому удается использовать одну и ту же программу для ВВ через порт, адрес которого заранее неизвестен и может меняться в процессе работы МС. Примеры команд ВВ:

```
IN      @reg        ;A←IOSEG(reg)
OUT     @reg        ;IOSEG(reg)←A
```

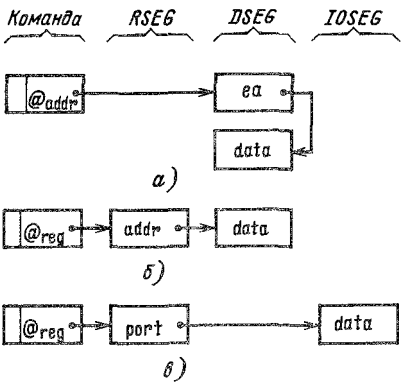


Рис. 1.20. Способы адресации данных:
а — косвенная; б — косвенная регистровая; в — косвенная регистровая адресация портов

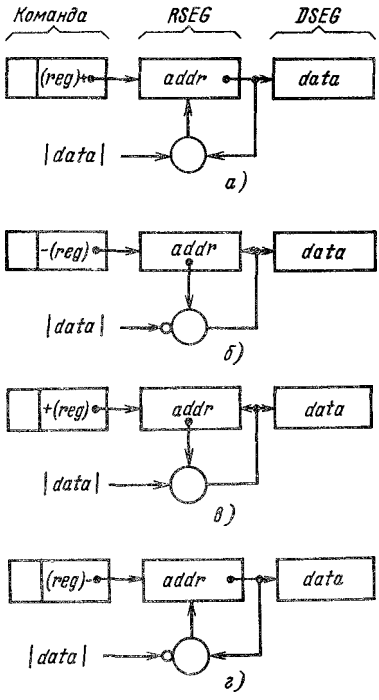


Рис. 1.21. Способы адресации с автомодификацией:
а — автоинкрементная; б — автодекрементная; в — преавтоинкрементная; г — поставтодекрементная

Для пошагового просмотра таблиц и списков удобно применять косвенную регистровую адресацию. Настройка на следующий адрес элемента выполняется программным способом с помощью операций, манипулирующих с адресным регистром:

```
LOOP:   MOVB    ptr, @reg       ;Поиск байта
        INC     reg             ;data в строке
        CMPB    ptr,#data       ;сравнение байтов, начи-
        JNZ     LOOP            ;ная с адреса (reg)
```

Очень часто данные располагаются в памяти последовательно (очереди, последовательности, файлы и т. д.). Для повышения эффективности работы со структурами такого типа используются *автоинкрементный* и *автодекрементный способы адресации* (ptr = = (reg)+, ptr = −(reg)).

При автоинкрементном способе (рис. 1.21, а) обеспечивается вычисление исполнительного адреса, как и при косвенном, но с последующим автоматическим увеличением содержимого адрес-

ного регистра на длину операнда, что позволяет подготовиться к выборке следующего элемента данных:

```
MOVB    dst,(reg)+          ;dst←(reg), reg←reg+1
```

При автодекрементной адресации (рис. 1.21, б) содержимое адресного регистра сначала уменьшается на длину операнда, а затем используется как исполнительный адрес:

```
MOVB    dst,-(reg)          ;reg←reg-1, dst←(reg)
```

Эти способы адресации позволяют исключать из программы настройку адреса, что существенно ускоряет процедуры просмотра потоковых структур.

Комбинирование автоинкрементных способов адресации позволяет организовать стеки с адресным регистром в качестве SP. Для загрузки данных в стек применяется автодекрементная адресация:

```
MOV     -(reg),src
```

для выталкивания — автоинкрементная:

```
MOV     dst,(reg)+
```

Доступ к вершине стека TOS осуществляется с помощью косвенной регистровой адресации:

```
MOV     dst,@reg
```

Указатель стека reg всегда указывает на вершину стека, который заполняется в сторону уменьшения адресов. Это соответствует принятой ранее стандартной организации стека.

Неявно заданный способ адресации с модификацией реализуется в операциях со стеком:

```
PUSH    src             ;-(SP)←src
POP     dst             ;dst←(SP)+
CALL    src             ;-(SP)←PC←src
RET                     ;PC←(SP)+
CALL    reg,src         ;-(SP)←reg←PC←src
RET     reg             ;PC←reg←(SP)+
```

В этом случае его называют стековым.

Процедура выборки программной последовательности также использует неявно заданную автоинкрементную адресацию по PC. Действительно, после считывания очередного элемента объектного кода указатель PC автоматически увеличивает свое содержимое на 1 или 2, в зависимости от длины программного элемента.

Непосредственную адресацию данных можно интерпретировать как автоинкрементную по PC. Например, программная последовательность

```
MOVB    reg,(PC)+
DB      data
```

эквивалентна команде с литералом

```
MOVB    reg,#data
```

Здесь DB (Define Byte) не имя какой-либо машинной команды, а символическое обозначение директивы размещения последовательности байтов данных. Эти данные должны быть указаны следом за директивой и при необходимости отделяются друг от друга запятыми. В нашем случае последовательность содержит один байт данных. После приема команды

```
MOVB    reg, (PC)+
```

указатель PC адресует байт data. После выполнения операции и следующей за ней автоматической модификации он будет указывать на начало новой команды.

Возможны два варианта определения автоинкрементной адресации (ptr = +(reg) и ptr = (reg)−), которые называют *преавтоинкрементной* и *поставтодекрементной* соответственно (рис. 1.21, в, г). Их совместное использование также обеспечивает организацию стека с указателем, адресующим TOS. Однако растет такой стек в сторону увеличения адресов.

В общем случае каждой паре автомодификации соответствуют по два типа стека. Эти стеки характеризуются следующими операциями доступа:

Операция	PUSH src	POP src
Стандартный стек	MOV−(reg),src	MOV dst,(reg)+
Стек типа 1	MOV (reg)+,src	MOV dst,−(reg)
Стек типа 2	MOV+(reg),src	MOV dst,(reg)−
Стек типа 3	MOV(reg)−,src	MOV dst,+(reg)

В отличие от стандартного стека и стека типа 2 указатель reg в двух новых стеках адресует первую свободную ячейку над TOS. При этом стек типа 1 растет в сторону уменьшения адреса, а стек типа 3 в сторону увеличения. Согласованность операций доступа стандартного стека с операцией выборки объектного кода, а также прямая адресуемость TOS по содержимому SP определили его широкое использование в качестве типового стека МС.

Комбинирование адресации с автомодификацией и косвенной позволяет построить более универсальные способы доступа по сравнению с рассмотренными выше — *косвенную автоинкрементную и автодекрементную адресацию* (ptr = @(reg)+, ptr = @−(reg)). Четыре способа представлены на рис. 1.22. В их основу положен стандартный вариант автомодификации. Достаточно широкое распространение находит лишь первая пара (рис. 1.22, а, б), которая характеризуется однократным обращением к памяти данных. Вторая пара способов (рис. 1.22, в, г) требует до трех обращений к памяти, что существенно снижает их практическую ценность.

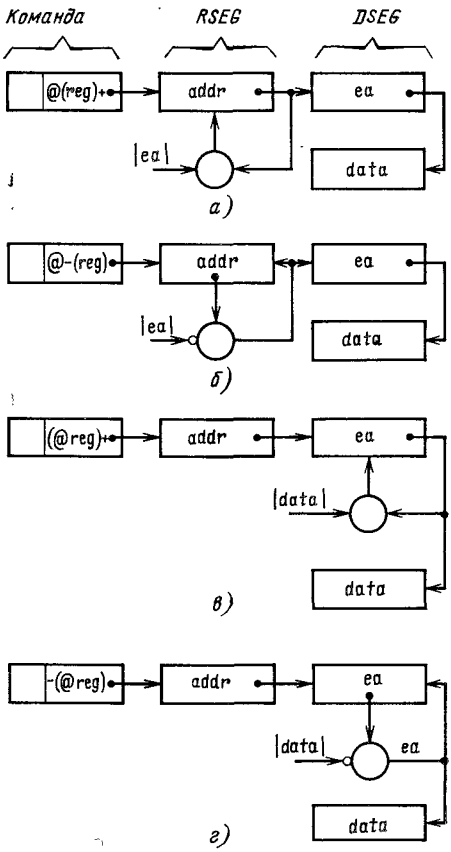


Рис. 1.22. Способы комбинированной адресации:

а — косвенная автоинкрементная; б — косвенная автодекрементная; в — прекосвенная автоинкрементная; г — прекосвенная автодекрементная

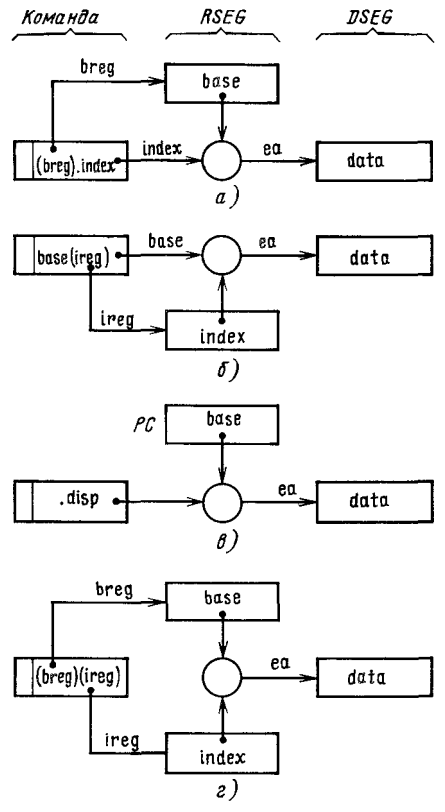


Рис. 1.23. Двухкомпонентные способы адресации:

а — по базе; б — индексная; в — относительная; г — двухкомпонентная косвенная регистровая

Многокомпонентные способы адресации. При многокомпонентных способах адресации для построения исполнительного адреса используется несколько источников адресной информации, содержимое которых суммируется. Каждое слагаемое — это прямо или косвенно заданный компонент (возможен вариант компонента с автомодификацией). В двухкомпонентном случае практический интерес представляют следующие комбинации способов адресации: прямая + косвенная регистровая, косвенная регистровая + косвенная регистровая, прямая + автоинкрементная. Наибольшее распространение нашла первая комбинация.

В случае *адресации по базе* (ptr = (breg).index) исполнительный адрес формируется по схеме, приведенной на рис. 1.23, а, когда

$$ea = (breg) + index,\ MOD\ 2^{|ea|}$$

Адресация по базе необходима для организации доступа к конкретным полям блока данных, адрес которого может быть неизвестен во время создания программы. Например, блок данных динамически создается в разных областях памяти или его размещение определяется в момент загрузки. В такой ситуации переменный адрес начала блока (базу) удобно располагать в регистре breg (базовом), а известный относительный адрес элемента (индекс) хранить непосредственно в команде. Допустим и вариант короткого индекса, уменьшающий длину объектного кода: $|index| \leqslant |ea|$. При этом индекс может восприниматься как целое либо без знака (offset), либо со знаком (disp). Во втором случае появляется возможность адресоваться относительно базы в обе стороны. При необходимости для выполнения операции суммирования (breg)+index производится расширение знака короткоразмерного индекса до полного размера.

Адресация по базе используется: для передачи данных в блоке параметров (базовый регистр служит указателем блока); для передачи данных через стек (базовый регистр выполняет роль указателя кадра FP); для передачи данных и организации переходов в программах, не зависящих от места их размещения в памяти (базовый регистр указывает на начало программы); для доступа к конкретному полю блока в связном списке (базовый регистр применяется для адресации элементов списка) или к одному и тому же полю динамически созданной записи (базовый регистр указывает на начало записи).

Схема вычисления исполнительного адреса при *индексной адресации* (ptr = base(ireg)) показана на рис. 1.23, б и похожа на схему вычисления по базе:

$$ea = base + (ireg),\ MOD\ 2^{|ea|}$$

Однако методика ее использования иная.

Индексная адресация применяется при последовательном доступе к элементам блока (массив, таблица, очередь), адрес размещения которого (база) известен. При этом номер элемента блока (индекс) является переменной, вычисляемой во время исполнения программы. Поэтому индекс удобно хранить в регистре ireg (индексном), а базу — в объектном коде команды. Отличие индексной адресации от адресации по базе состоит в том, что в объектном коде должен храниться полный адрес памяти $|base| = |ea|$, тогда как в качестве индексного регистра может использоваться регистр меньшей длины. При $|ireg| = |ea|$ индексная адресация включает адресацию по базе. По этой причине последнюю часто называют индексной. В случае нулевой базы индексная адресация с полноразмерным индексом совпадает с косвенной, тогда регистр косвенной адресации называется индексным.

Разновидностью адресации по базе является *относительная* (PC).disp, при которой в качестве базового регистра breg используется PC (рис. 1.23, в), как правило, указываемый неявно (ptr = .disp).

Отметим, что во всех случаях двухкомпонентной адресации индекс может быть целым числом со знаком. Это дает возможность адресоваться не только в сторону больших, но и меньших адресов. При относительной адресации смещение всегда целое число со знаком. Существует вариант *короткого смещения*, обеспечивающий более компактное кодирование команд.

Относительная адресация применяется для создания *позиционно-независимых программ*, т. е. программ, исполнение которых не зависит от их размещения в памяти. Примерами команд с относительной адресацией являются

```
MOV     reg,.disp       ;reg←(PC+disp)
JMP     .disp           ;PC←PC+disp
```

Каждому из вышеприведенных двухкомпонентных способов адресации соответствует вариант косвенной адресации:

```
@(breg).index       Косвенный по базе
@base(ireg)         Косвенный индексный
@.disp              Косвенный относительный
```

Здесь исполнительный адрес формируется по схеме

ea = (base + index)

Косвенные варианты требуют двойного обращения к памяти МС, поэтому они не получили широкого распространения.

Двухкомпонентная косвенная регистровая адресация (ptr = = (breg) (ireg)) может быть получена, когда обе компоненты — и база, и индекс — заданы косвенно (рис. 1.23, г). Способ симметричен по отношению к своим компонентам только при полноразмерном индексном регистре.

Трехкомпонентные способы адресации. Дальнейшего увеличения числа способов адресации можно достичь при переходе к *трехкомпонентным схемам* типа прямая + косвенная регистровая. Среди них

```
(breg).index (ireg)     Базированная с индексированием
base(ireg1) (ireg2)     С двойным индексированием
```

Адресация такого типа используется в МП К1810ВМ86, в котором

index = disp8/disp16
base = offset16

ГЛАВА 2.

МИКРОПРОЦЕССОРЫ КР580ВМ80/К1821ВМ85А

2.1. Вводные замечания

Архитектура однокристального 8-разрядного n-МОП микропроцессора КР580ВМ80 (ВМ80) второго поколения [1, 61] аналогична архитектуре известного МП 8080 [23, 40, 56]. Схемотехнические и архитектурные особенности ВМ80 в сочетании с n-канальной технологией позволили уменьшить время выполнения команд типов регистр-регистр до 1,6 мкс, регистр-память до 2,8 мкс при тактовой частоте 2,5 МГц, что соответствует быстродействию 500 тыс. операций в секунду.

Для работы ВМ80 применяется три источника питания: −5 В (ток потребления менее 1 мА), +5 В (ток потребления менее 70 мА) и +12 В (ток потребления менее 50 мА). Кристалл МП, содержащий около 5000 транзисторов, помещен в корпус с двухрядным расположением 40 выводов и шагом 2,5 мм между ними. Синхронизация МП осуществляется от двухфазного внешнего генератора, амплитуда тактовых импульсов которого 12 В.

На рис. 2.1 приведена базовая структурная схема МС, построенной на базе МП ВМ80. В системе используется общая для хранения программ и данных основная память емкостью 64 К байт и подсистема ВВ емкостью 256 байт. Двунаправленная 8-разрядная шина данных DAT, 16-разрядная шина адреса ADR и шина управления СВ служат для организации побайтового обмена между отдельными модулями системы.

2.2. Архитектура ВМ80

Блок программно-доступных регистров МП ВМ80 характеризуется большой функциональной неоднородностью. Практически каждый регистр МП выполняет присущую только ему функцию. Это сделано с целью более короткого кодирования системы команд. Однако из-за необходимости учитывать все функциональные особенности регистрового блока усложнилось программирование. Набор программно-доступных регистров МП ВМ80 приведен на рис. 2.2.

Восьмиразрядный аккумулятор А используется в большинстве команд логической и арифметической обработки. Обычно он адресуется неявно и служит как источником операнда, так и приемником результата. Благодаря этому в командах МП ВМ80 явно указывается только один операнд.

Признаки результата операции фиксируются во флажковом регистре F (Flags). Пять флажков CY, P, AC, Z и M упакованы в

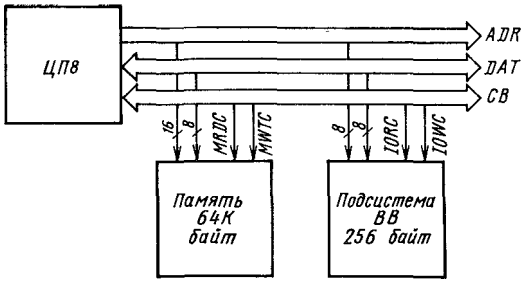


Рис. 2.1. Схема микроЭВМ на базе микропроцессора ВМ80

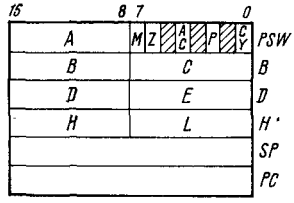


Рис. 2.2. Набор регистров микропроцессора ВМ80

байт, три разряда которого не используются. Флажки имеют следующее функциональное назначение:

CY (Carry)	Признак переноса из старшего разряда АЛУ
P (Parity)	Признак четного числа единиц в результате операции
AC (Auxiliary Carry)	Признак дополнительного переноса из младших четырех разрядов (младшей тетрады) АЛУ
Z (Zero)	Признак нулевого результата
M (Minus)	Знак результата

В некоторых командах пересылки флажковый регистр F совместно с аккумулятором A образует 16-разрядный регистр слова состояния программы PSW.

Шестнадцатиразрядный регистр H, как правило, служит адресным регистром. При косвенной регистровой адресации он хранит 16-разрядный исполнительный адрес основной памяти. В этом случае к нему ссылаются с помощью мнемоники M (Memory). В некоторых командах старший и младший байты 16-разрядного регистра H могут быть адресованы независимо и использованы как отдельные 8-разрядные регистры данных H (High Byte) и L (Low Byte) соответственно.

Для организации вызова подпрограмм и процедур обслуживания прерываний, а также ряда других функций в состав регистров введен 16-разрядный указатель стека SP. Он всегда указывает на вершину стека TOS, которая содержит последний введенный в стек элемент. Стек заполняется в сторону младших адресов. Такое направление заполнения для МС считается стандартным.

Программный счетчик PC выполняет свою обычную функцию адресации объектного кода. В фазе исполнения текущей команды он указывает на следующий подлежащий выборке элемент командной последовательности. В этот момент возможна его перезагрузка, которая приведет к изменению последовательной выборки команд.

Перечисленные выше 8-разрядные арифметические регистры F, A вместе с 16-разрядными адресными регистрами H, SP и PC образуют стандартный регистровый набор МП с аккумулятором (см. рис. 1.15, *а*). Набор МП ВМ80 расширен четырьмя 8-разрядными РОН: B, C, D и E. Эти регистры могут быть использованы либо как четыре регистра данных, либо как два 16-разрядных адресных регистра B и D, образованных парами BC и DE соответственно. Младшими байтами в парах являются C и E. Введение РОН позволило создать достаточно эффективный 8-разрядный МП с широкими функциональными возможностями.

В командах на регистры ссылаются как явно, кодируя его 3-разрядным полем, так и неявно с помощью кода операции. Код операции всегда подразумевает способ использования регистров. Широкому распространению неявной формы ссылки способствует функциональная неоднородность регистров.

Основная память МП ВМ80 рассматривается как линейный массив, состоящий из 64К байт. Формируемый микропроцессором 16-разрядный адрес дает ему возможность адресовать любой байт памяти. Слова в памяти хранятся в двух соседних байтах. В байте с младшим адресом хранится младшая половина слова, а в байте со следующим адресом — старшая. Адресом слова служит адрес его младшего байта.

В МП определено четыре способа задания места расположения данных в памяти: прямой, косвенный регистровый через 16-разрядный адресный регистр H, B или D, непосредственный и автоинкрементный (автодекрементный) через указатель стека SP (стековый). При прямой и непосредственной адресации данных могут быть доступны байты или слова, при косвенной адресации — только байты. Стековая адресация применяется только при работе со словами. Как способ адресации, так и тип операнда определяется неявно кодом операции.

В МП ВМ80 используется изолированное пространство ВВ. Эта отдельная область организована в виде массива из 256 8-разрядных портов ввода и 256 8-разрядных портов вывода. Допускается только один способ доступа к пространству ВВ — прямой, когда 8-разрядный адрес порта указывается непосредственно в команде.

2.3. Система команд ВМ80

В МП ВМ80 применяется довольно простой формат команд, приведенный на рис. 2.3. Команды имеют длину от 1 до 3 байт. Код операции всегда размещен в первом байте команды. Второй и, если необходимо, третий байты команды отводятся под непосредственные данные, адрес порта или ячейки памяти. В командах допускается явное задание только одного адреса

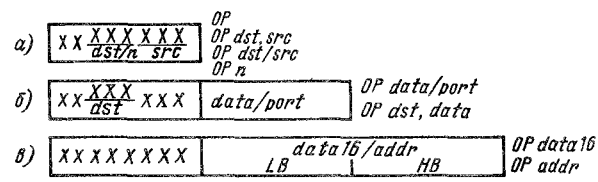


Рис. 2.3. Формат команд микропроцессора ВМ80:
а — однобайтовый; б — двухбайтовый; в — трехбайтовый

памяти. По этой причине систему команд МП следует отнести к классу одноадресных.

Система команд МП [11, 16] состоит из пяти групп: пересылки (14 команд, 28 операций), логической обработки (15 команд, 19 операций), арифметической обработки (14 команд, 29 операций), передачи управления (28 команд, 28 операций), управления процессором (7 команд, 7 операций). Всего в систему входят 78 базовых команд, содержащих 111 кодов операций. Их полный состав представлен в табл. 2.1—2.5. Наряду с мнемоникой и кодом операции (первый байт команды) таблицы содержат такие важнейшие для команд характеристики, как число обращений к системной магистрали и число периодов основной тактовой частоты, составляющих ее полный командный цикл. Здесь же представлена информация о влиянии команды на флажки регистра признаков F. При знаке «+» производится установка соответствующего флажка в зависимости от результата операции, знак «—» означает, что команда не влияет на него и, следовательно, оставляет без изменения. Длина команды в байтах легко определяется из ее мнемоники.

В условных командах число обращений к каналу и длительность машинного цикла в тактах зависят от выполнения условия. В таблицах указаны два значения невыполнения и выполнения условия.

Система команд МП ВМ80 в полном составе входит в систему команд МП К1821ВМ85А (ВМ85А) [65], обеспечивая их совместимость на уровне объектного кода. Они отличаются друг от друга лишь числом периодов основной тактовой частоты, необходимым для исполнения той или иной команды. По этой причине в таблицах даны два значения длительности командного цикла: одно для ВМ80, другое для ВМ85А. Кроме того, в табл. 2.5 введены две команды, не поддерживаемые архитектурой ВМ80. Это расширение системы команд ВМ85А по сравнению с ВМ80.

В мнемонике команд использованы следующие обозначения:

src	8-разрядный регистр-источник
dst	8-разрядный регистр-приемник
data	8-разрядный литерал
data16	16-разрядный литерал
addr	16-разрядный адрес памяти
port	8-разрядный адрес порта
n	Цифра от 0 до 7

Таблица 2.1

Мнемоника		Код	Число циклов ВМ80	Число тактов		Флажки	Содержание
				ВМ80	ВМ85А	C Y Z M P A C	
MOV	dst, src	01DDDSSS	1	5	4	– – – – –	dst ← src
MOV	dst, M	01DDD110	2	7	7	– – – – –	dst ← (HL)
MOV	M, src	01110SSS	2	7	7	– – – – –	(HL) ← src
MVI	dst, data	00DDD110	2	7	7	– – – – –	dst ← data
MVI	M, data	36	3	10	10	– – – – –	(HL) ← data
LDA	addr	3A	4	13	13	– – – – –	A ← (addr)
STA	addr	32	4	13	13	– – – – –	(addr) ← A
LDAX	B	0A	2	7	7	– – – – –	A ← (BC)
LDAX	D	1A	2	7	7	– – – – –	A ← (DE)
STAX	B	02	2	7	7	– – – – –	(BC) ← A
STAX	D	12	2	7	7	– – – – –	(DE) ← A
LXI	B,data,16	01	3	10	10	– – – – –	BC ← data16
LXI	D,data16	11	3	10	10	– – – – –	DE ← data16
LXI	H,data16	21	3	10	10	– – – – –	HL ← data16
LXI	SP,data16	31	3	10	10	– – – – –	SP ← data16
LHLD	addr	2A	5	16	16	– – – – –	HL ← (addr)
SHLD	addr	22	5	16	16	– – – – –	(addr) ← HL
SPHL		F9	1	5	6	– – – – –	SP ← HL
PUSH	S	C5	3	11	12	– – – – –	–(SP) ← BC
PUSH	D	D5	3	11	12	– – – – –	–(SP) ← DE
PUSH	H	E5	3	11	12	– – – – –	–(SP) ← HL
PUSH	PSW	F5	3	11	12	– – – – –	–(SP) ← PSW
POP	B	C1	3	10	10	– – – – –	BC ← (SP)+
POP	D	D1	3	10	10	– – – – –	DE ← (SP)+
POP	H	E1	3	10	10	– – – – –	HL ← (SP)+
POP	PSW	F1	3	10	10	+ + + + +	PSW ← (SP)+
XCHG		EB	1	4	10	– – – – –	DE ↔ HL
XTHL		E3	5	18	16	– – – – –	(SP) ↔ HL

Поля src и dst означают один из 8-разрядных регистров A, B, C, D, E, H или L. Для получения правильного кода операции следует в соответствующее поле кода записать двоичный код регистра согласно правилу

Регистр	B	C	D	E	H	L	A
Код	000	001	010	011	100	101	111

Представленная в табл. 2.1 группа содержит команды обмена между памятью и регистрами. Это наиболее часто встречающиеся в программах команды, которые занимают около 45% их общего числа. Группа команд пересылки никакого воздействия на флажковый регистр не оказывает, за исключением тех команд, которые используют флажковый регистр в качестве приемника данных.

Основу группы составляют следующие команды:

MOV, MVI	Перемещение
LDA, LDAX, LXI, LHLD	Загрузка
STA, STAX, SHLD	Сохранение

Эти команды оперируют как байтами MOV, MVI, LDA, STA, LDAX, STAX, так и словами LXI, LHLD, SHLD.

В командах пересылки байтов поля src и dst используются для указания 8-разрядных регистров A, B, C, D, E, H, L, а M обозначает косвенную адресацию через регистровую пару HL, которая должна содержать прямой адрес байта, участвующего в обмене. В составе команд также находятся операции загрузки аккумулятора с прямой (LDA) и косвенной (LDAX) адресацией через регистровые пары BC и DE, а также их обратные эквиваленты (STA, STAX). С учетом значимости 16-разрядного регистра H в составе группы предусмотрены операции загрузки LHLD и хранения SHLD его содержимого по прямому адресу. Команды MVI и LXI используют непосредственную адресацию, обеспечивающую загрузку 8- и 16-разрядного регистра или байта памяти некоторой константой.

Для начальной установки регистра SP предусмотрены две команды:

```
LXI  SP,data 16 ;SP ← data16
SPHL            ;SP ← HL
```

Если первая команда обеспечивает прямую загрузку SP константой, то с помощью второй можно организовать установку SP в соответствии со значением некоторой переменной. Такая операция удобна при реализации нескольких стеков.

Команды PUSH и POP организуют запись в стек и выборку из него содержимого 16-разрядных регистров B, D, H или PSW. Благодаря этим операциям создается удобный механизм сохранения и восстановления текущего контекста регистровой области или ее части для передачи параметров через стек, а также для выполнения других важных при программировании функций. Системный стек растет в сторону уменьшения адресов, а его указатель всегда адресует последний элемент стека или его вершину TOS.

В составе группы две команды обмена:

```
XCHG            ;DE↔HL
XTHL            ;(SP)↔HL
```

Они обеспечивают обмен между регистровой парой HL и регистровой парой DE или вершиной стека TOS. Первая операция позволяет использовать содержимое 16-разрядного регистра DE практически с тем же результатом, что и содержимое пары HL. Вторая поддерживает прямой доступ к TOS, что в ряде случаев очень важно. В тех ситуациях, когда TOS содержит адрес возврата, команда XTHL дает возможность доступа к нему с

целью модификации. Если применяется механизм передачи параметров через линейную область, т. е. параметры следуют непосредственно за командой вызова CALL, команда XTHL обеспечивает простую настройку на область параметров, их выборку и модификацию адреса возврата в соответствии с длиной области. Важно то, что все эти операции выполняются без потери содержимого HL.

Команда XTHL используется также для получения текущего состояния PC, например при организации относительной адресации. Фрагмент такого доступа имеет вид

```
        CALL    M1          ;–(SP)←M1, PC←M1
M1:     XTHL                ;HL↔(SP)
```

В данном случае HL получает значение метки M1, которое далее может быть употреблено произвольно. Иногда требуется произвести обмен между вершиной стека и регистром D, осуществляемый с помощью следующей последовательности:

```
        XCHG                ;DE↔HL
        XTHL                ;HL↔(SP)
        XCHG                ;DE↔HL
```

В системе команд отсутствует важная операция загрузки литерала в стек. Однако процедуру такого рода легко выполнить в виде

```
        LXI     H,data16    ; HL←data16
        PUSH    H           ;–(SP)←HL
```

Для сохранения содержимого HL можно использовать фрагмент

```
        PUSH    H           ;–(SP)←HL
        LXI     H, data16   ;HL←data16
        XTHL                ;(SP)↔HL
```

Группа команд логической обработки приведена в табл. 2.2. В ее составе четыре двухместных логических операции над байтами:

ANA, ANI	Логическое И
XRA, XRI	Исключающее ИЛИ
ORA, ORI	Логическое ИЛИ
CMP, CPI	Сравнение

В этих командах в качестве источника одного из операндов используют аккумулятор A, который одновременно служит и приемником результата. Источником второго операнда является

Таблица 2.2

Мнемоника	Код	Число циклов ВМ80	Число тактов		Флажки	Содержание
			ВМ80	ВМ85А	C Y Z M P A C	
ANA src	10100SSS	1	4	4	0+++0	A ← A AND src
XRA src	10101SSS	1	4	4	0+++0	A ← A XOR src
ORA src	10110SSS	1	4	4	0+++0	A ← A OR src
CMP src	*10111SSS*	*1*	*4*	*4*	+++++	*A − src*
ANA M	A6	2	7	7	0+++0	A ← A AND (HL)
XRA M	AE	2	7	7	0+++0	A ← A XOR (HL)
ORA M	B6	2	7	7	0+++0	A ← A OR (HL)
CMP M	BE	2	7	7	+++++	A − (HL)
ANI data	E6	2	7	7	0+++0	A ← A AND data
XRI data	EE	2	7	7	0+++0	A ← A XOR data
ORI data	F6	2	7	7	0+++0	A ← A OR data
CPI data	FE	2	7	7	+++++	A − data
RLC	07	1	4	4	+----	$A_7 \leftarrow A_6 \leftarrow ... A_0 \leftarrow A_7$
RRC	OF	1	4	4	+----	$A_0 \leftarrow A_1 \leftarrow ... A_7 \leftarrow A_0$
RAL	17	1	4	4	+----	$A_7 \leftarrow A_6 \leftarrow ... A_0 \leftarrow$ $\leftarrow CY \leftarrow A_7$
RAR	1F	1	4	4	+----	$A_0 \leftarrow A_1 \leftarrow ... A_7 \leftarrow$ $\leftarrow CY \leftarrow A_0$
CMA	2E	1	4	4	-----	A ← NOT A
CMC	3F	1	4	4	+----	CY ← NOT CY
STC	37	1	4	4	1----	CY ← 1

либо 8-разрядный регистр, кодируемый полем src, либо ячейка памяти, адресуемая парой HL, либо константа, заданная непосредственно в команде. Все команды влияют на флажки признаков результата, входящие в состав флажкового регистра. Заметим, что операция сравнения выполняется методом арифметического вычитания. По этой причине ее следовало бы отнести к группе арифметических команд.

Предусмотрена *еще одна типовая логическая операция* — логическое дополнение или инверсия CMA, которая работает только с неявно адресуемым аккумулятором. Особенность команды состоит в том, что она не влияет на состояние флажкового регистра.

Две команды STC и CMC дают возможность манипулировать флажком CY, устанавливая или инвертируя его. Сброс флажка может быть выполнен командой

```
ORA     A          ;A ← A OR A
```

При этом следует помнить, что состояние других флажков также изменяется.

Таблица 2.3

Мнемоника		Код	Число циклов ВМ80	Число тактов		Флажки	Содержание
				ВМ80	ВМ85А	$\overset{C}{Y}$ Z M P $\overset{A}{C}$	
ADD	src	10000SSS	1	4	4	+ + + + +	A ← A + A + src
ADC	src	10001SSS	1	4	4	+ + + + +	A ← A + src + CY
SUB	src	10010SSS	1	4	4	+ + + + +	A ← A − src
SBB	src	10011SSS	1	4	4	+ + + + +	A ← A − src − CY
ADD	M	86	2	7	7	+ + + + +	A ← A + (HL)
ADC	M	8E	2	7	7	+ + + + +	A ← A + (HL) + CY
SUB	M	96	2	7	7	+ + + + +	A ← A − (HL)
SBB	M	9E	2	7	7	+ + + + +	A ← A − (HL) − CY
ADI	data	C6	2	7	7	+ + + + +	A ← A + data
ACI	data	CE	2	7	7	+ + + + +	A ← A + data + CY
SUI	data	D6	2	7	7	+ + + + +	A ← A − data
SBI	data	DE	2	7	7	+ + + + +	A ← A − data − CY
INR	dst	00DDD100	1	5	4	− + + + +	dst ← dst + 1
DCR	dst	00DDD101	1	5	4	− + + + +	dst ← dst − 1
INR	M	34	3	10	10	− + + + +	(HL) ← (HL) + 1
DCR	M	35	3	10	10	− + + + +	(HL) ← (HL) − 1
DAA		27	1	4	4	+ + + + +	A ← 2/10-коррекция A
DAD	B	09	3	10	10	+ − − − −	HL ← HL + BC
DAD	D	19	3	10	10	+ − − − −	HL ← HL + DE
DAD	H	29	3	10	10	+ − − − −	HL ← HL + HL
DAD	SP	39	3	10	10	+ − − − −	HL ← HL + SP
INX	B	03	1	5	6	− − − − −	BC ← BC + 1
INX		13	1	5	6	− − − − −	DE ← DE + 1
INX	H	23	1	5	6	− − − − −	HL ← HL + 1
INX	SP	33	1	5	6	− − − − −	SP ← SP + 1
DCX	B	0B	1	5	6	− − − − −	BC ← BC − 1
DCX	D	1B	1	5	6	− − − − −	DE ← DE − 1
DCX	H	2B	1	5	6	− − − − −	HL ← HL − 1
DCX	SP	3B	1	5	6	− − − − −	SP ← SP − 1

Обычно в состав логических команд включают также и подгруппу сдвигов. В системе команд ВМ80 четыре команды сдвига: две вправо (RRC, RAR) и две влево (RLC, RAL). Определены операции циклического RRC, RLC и расширенного RAR, RAL сдвигов. В операциях участвует только флажок переноса CY, который всегда принимает значение выходного бита. Предварительная установка CY в 0 или 1 совместно с расширенным сдвигом дает возможность организовать отсутствующие в системе команд логические и арифметические сдвиги в обе стороны.

Группа команд арифметической обработки (табл. 2.3) осуществляет эффективную обработку числовой информации. В группе определены:

ADD, ADC, ADI, ACI, DAD	Сложение
SUB, SBB, SUI, SBI	Вычитание
INR, INX	Увеличение на 1
DCR, DCX	Уменьшение на 1
DAA	Десятичная коррекция

Предусмотрены операции как над байтами ADD, ADC, ADI, ACI, SUB, SBB, SUI, SBI, INR, DCR, DAA, так и над словами DAD, INX, DCX.

Во всех байтовых операциях сложения и вычитания используется аккумулятор как источник операнда и приемник результата. В качестве источника второго операнда применяется либо регистр src, либо ячейка памяти M, либо литерал data. В некоторых операциях (ADC, ACI, SBB и SBI) предусмотрен учет состояния флажка CY. В команде сложения DAD роль аккумулятора выполняет 16-разрядный регистр H, содержимое которого складывается с содержимым одного из 16-разрядных регистров B, D, H или SP. Эта команда очень важна при организации таблиц и списков. Следует отметить, что команда DAD оказывает влияние только на флажок CY.

Очень полезны операции увеличения и уменьшения на единицу. С их помощью довольно просто реализовать счетчики, часто необходимые в практике программирования. Операции применимы к 8- и к 16-разрядным регистрам, включая SP. Возможна также модификация байтовой ячейки памяти, адресуемой регистром HL.

Во многих МС предусмотрены специальные средства для работы с десятичными числами. Обработка числовых данных непосредственно в десятичной форме очень часто применяется в системах с десятичным ВВ, так как не требуется их промежуточного преобразования в двоичный формат и обратно. Для хранения десятичных чисел в памяти МС используется двоично-десятичный код (2/10-код) или код BCD (Binary Code Decimal). Различают два формата 2/10-кода: упакованный и распакованный.

В 2/10-коде упакованного формата каждая десятичная цифра от 0 до 9 представляется 4-разрядным двоичным эквивалентом от 0000B до 1001B соответственно, коды 1010B—1111B не используются. В одном байте удается разместить только две десятичные цифры и, следовательно, представить целое число из диапазона 0—99. Для представления многоразрядного десятичного числа отводится большее число байт (по байту на каждые две цифры). Например:

```
9272=1001  0010  0111  0010B
 380=0011  1000  0000B
```

Переход от одной системы представления к другой выполняется

заменой каждой десятичной цифры ее двоичным эквивалентом и наоборот.

В 2/10-коде распакованного формата для представления каждой десятичной цифры отводится один байт, младшая тетрада которого содержит двоичный код цифры, а содержимое старшей тетрады равно нулю. Приведенные выше десятичные числа в распакованном формате будут выглядеть следующим образом:

9272=00001001 00000010 00000111 00000010B
380=00000011 00001000 00000000B

Распакованный формат в сравнении с упакованным требует в 2 раза большей памяти, однако он очень хорошо согласуется с текстовым представлением десятичных цифр в коде КОИ-7. Для перевода распакованного 2/10-кода в текстовой формат достаточно в старшую тетраду каждой цифры записать код 0011B, а замена каждой старшей тетрады текстового представления десятичного числа нулями приведет к его переводу в распакованный формат 2/10-кода.

При необходимости знак десятичного числа кодируется отдельной тетрадой или байтом. В обоих случаях комбинация 0000B соответствует знаку плюс, а комбинация 1001B — знаку минус. Например, в случае упакованного формата

+921=0000 1001 0010 0001B
−350=1001 0011 0101 0000B

в распакованном формате

−350=00001001 00000011 00000101 00000000B

Арифметическая операция над десятичными числами выполняется в два этапа: сначала производится обычная двоичная операция над десятичными числами, затем десятичная коррекция полученного результата. При двухэтапном методе можно воспользоваться АЛУ двоичной арифметики. Для этого требуется лишь ввести в состав АЛУ специальный десятичный корректор, приводящий результат двоичной операции к десятичному виду. Каждой основной операции над десятичными числами необходим определенный корректор. Наибольшее распространение нашли следующие типы десятичной коррекции:

DAA	Десятичная коррекция сложения
DAS	Десятичная коррекция вычитания
AAA	Коррекция сложения в коде КОИ-7
AAS	Коррекция вычитания в коде КОИ-7
AAM	Коррекция умножения в коде КОИ-7
AAD	Коррекция деления в коде КОИ-7

Первые два типа поддерживают десятичную арифметику в упакованном формате, остальные — в распакованном. Для архитектур с аккумулятором или вычислительным стеком команды десятичной коррекции безадресные и работают либо с аккумулятором, либо с вершиной стека соответственно. Операции десятичной коррекции обычно являются байтно ориентированными, т. е. за один командный цикл допускается коррекция либо двухразрядного десятичного числа упакованного формата, либо одной десятичной цифры распакованного формата. Рассмотрим более подробно операцию AAA.

При двоичном сложении двух десятичных цифр в 2/10-коде с дополнительным учетом состояния флажка CF может быть получено двоичное число, находящееся в диапазоне 0—13H. Для его приведения к 2/10-коду должна быть применена операция десятичной коррекции. При попадании результата операции в диапазон 0—9 никакой коррекции не требуется. Если же результат превышает цифру 9, то его коррекция необходима. Она сводится к прибавлению числа 0F6H, благодаря чему результат становится верным. Десятичный перенос фиксируется флажком CF и может быть учтен при сложении более старших десятичных цифр. Реализовать такой десятичный корректор несложно.

По аналогии с AAA могут быть построены карты преобразований любых других коррекций для упакованного и распакованного форматов. При этом в случае упакованного формата может потребоваться дополнительный признак — перенос из младшей тетрады в старшую, который фиксируется флажком AF.

Команда десятичной коррекции для BM80 DAA поддерживает операцию сложения в упакованном 2/10-коде. Она используется для получения корректного результата после сложения двух байтов, представляющих собой числа в 2/10-коде. Например:

```
ADD B      ;A ← A2/10 + B2/10
DAA        ;A2/10 ← Коррекция A
```

Представленная в табл. 2.4 группа команд передачи управления содержит три основные операции, типичные для большинства МС:

JMP	Переход
CALL	Вызов подпрограммы
RET	Возврат из подпрограммы

Эти операции организуют безусловный переход, нарушая процесс последовательной выборки команд. Для поддержки условной передачи управления в состав группы введены три соответствующие модификации базовых операций:

Таблица 2.4

Мнемоника	Код	Число циклов ВМ80	Число тактов		Флажки					Содержание
			ВМ80	ВМ85А	CY	Z	M	P	AC	
PCHL	E9	1	5	6	–	–	–	–	–	PC ← HL
JMP addr	C3	3	10	10	–	–	–	–	–	PC ← addr
JC addr	DA	3	10	7/10	–	–	–	–	–	Если CY=1, то JMP addr
JNC addr	D2	3	10	7/10	–	–	–	–	–	Если CY=0, то JMP addr
JZ addr	CA	3	10	7/10	–	–	–	–	–	Если Z=1, то JMP addr
JNZ addr	C2	3	10	7/10	–	–	–	–	–	Если Z=0, то JMP addr
JM addr	FA	3	10	7/10	–	–	–	–	–	Если M=1, то JMP addr
JP addr	F2	3	10	7/10	–	–	–	–	–	Если M=0, то JMP addr
JPE addr	EA	3	10	7/10	–	–	–	–	–	Если P=1, то JMP addr
JPO addr	E2	3	10	7/10	–	–	–	–	–	Если P=0, то JMP addr
CALL addr	CD	5	17	18	–	–	–	–	–	–(SP) ← PC ← addr
CC addr	DC	3/5	11/17	9/18	–	–	–	–	–	Если CY=1, то CALL addr
CNC addr	D4	3/5	11/17	9/18	–	–	–	–	–	Если CY=0, то CALL addr
CZ addr	CC	3/5	11/17	9/18	–	–	–	–	–	Если Z=1, то CALL addr
CNZ addr	C4	3/5	11/17	9/18	–	–	–	–	–	Если Z=0, то CALL addr
CM addr	FC	3/5	11/17	9/18	–	–	–	–	–	Если M=1, то CALL addr
CP addr	F4	3/5	11/17	9/18	–	–	–	–	–	Если M=0, то CALL addr
CPE addr	EC	3/5	11/17	9/18	–	–	–	–	–	Если P=1, то CALL addr
CPO addr	E4	3/5	11/17	9/18	–	–	–	–	–	Если P=0, то CALL addr
RET	C9	3	11	10	–	–	–	–	–	PC ← (SP) +
RC	D8	1/3	5/11	6/12	–	–	–	–	–	Если CY=1, то RET
RNC	D0	1/3	5/11	6/12	–	–	–	–	–	Если CY=0, то RET
RZ	C8	1/3	5/11	6/12	–	–	–	–	–	Если Z=1, то RET
RNZ	C0	1/3	5/11	6/12	–	–	–	–	–	Если Z=0, то RET
RM	F8	1/3	5/11	6/12	–	–	–	–	–	Если M=1, то RET
RP	F0	1/3	5/11	6/12	–	–	–	–	–	Если M=0, то RET
RPE	E8	1/3	5/11	6/12	–	–	–	–	–	Если P=1, то RET
RPO	E0	1/3	5/11	6/12	–	–	–	–	–	Если P=0, то RET

Jcc Условный переход
Ccc Условный вызов подпрограммы
Rcc Условный возврат из подпрограммы

Каждая условная операция обеспечивает проверку восьми условий, в соответствии с которыми меняется значение поля cc. Мнемоника поля и соответствующее ему условие приведены ниже:

C	Carry	CY=1
NC	Not Carry	CY=0
Z	Zero	Z=1
NZ	Not Zero	Z=0
M	Minus	M=1
P	Positive	M=0
PE	Parity Even	P=1
PO	Parity Odd	P=0

Передача управления производится только в том случае, если выполняется условие, указанное в команде. Состав условий дает

Таблица 2.5

Мнемоника	Код	Число циклов ВМ80	Число тактов		Флажки					Содержание
			ВМ80	ВМ85А	C Y	Z	M	P	A C	
IN port	DB	3	10	10	–	–	–	–	–	A ← IOSEG(port)
OUT port	D3	3	10	10	–	–	–	–	–	IOSEG(port) ← A
RST n	11NNN111	3	11	11	–	–	–	–	–	—(SP) ← PC ← 8·n, n = 0 – 7
EI	FB	1	4	4	–	–	–	–	–	Разрешение прерываний
DI	F3	1	4	4	–	–	–	–	–	Запрет прерываний
RIM*	20	—	—	—	–	–	–	–	–	Чтение маски
SIM*	30	—	—	4	–	–	–	–	–	Установка маски
HLT	76	1	7	5	–	–	–	–	–	Останов
NOP	00	1	4	4	–	–	–	–	–	Нет операции

* Команды для ВМ85А.

возможность контролировать состояние четырех основных флажков CY, Z, M, P и не проверять пятый AC.

В операциях как условного, так и безусловного перехода и вызова подпрограмм используется полный прямой 16-разрядный адрес, обеспечивающий передачу управления в любую точку 64К-байтовой области пространства памяти. Однако во всех этих случаях манипуляция адресом как параметром затруднена.

В составе группы имеется еще одна команда, которая реализует передачу управления по содержимому регистровой пары HL:

```
PCHL                    ;PC ← HL
```

Применение данной команды решает важную проблему передачи управления по вычисляемому адресу. С ее помощью достаточно просто организовать многопутевое ветвление типа

```
if   Y = 0 then
     goto M0
else if Y = 1 then
     goto M1
...                     ;и так далее
end if
```

Группа команд управления процессором содержит семь кодов. В табл. 2.5 кроме этих команд включены еще две, которые действительны только для МП ВМ85А (см. § 2.7). В состав группы включены две команды ВВ с прямой адресацией портов из пространства IOSEG:

```
IN      port        ;A ← IOSEG (port)
OUT     port        ;IOSEG (port) ← A
```

Сюда же входит ряд команд по обслуживанию системы прерываний. Команды EI и DI разрешают и запрещают прием запросов, сбрасывая и устанавливая маску прерывания. Предусмотрена специальная однобайтовая команда RST n, n=0–7, представляющая собой укороченный вариант команды CALL addr при addr=8·n. Эта команда обеспечивает возможность программной инициализации процедур обслуживания прерываний и вызова операционной системы или ее специальных средств (см. § 2.4).

Команда HLT используется для приостановки работы МП. В состоянии останова HALT прибор сохраняет возможность приема и обслуживания запросов на прерывания при сброшенной маске. Поэтому МП может быть выведен из состояния HALT двумя путями: перезапуском и по сигналу прерывания.

Команда NOP представляет собой пустую операцию. По данной команде никаких действий МП не производит. Она может быть полезна для организации коротких пауз и установки «заплат» на объектный код при его модификации.

В целом система команд МП ВМ80 — хорошо согласованный набор средств общего назначения, используемых при программировании 8-разрядных МС на его основе. Интенсивное внедрение набора программно-доступных регистров аккумуляторного типа обеспечило короткое кодирование и быстрое исполнение команд, что очень важно для экономии объема системной памяти и увеличения быстродействия МС. К серьезным недостаткам системы команд следует отнести отсутствие относительной адресации, что привело к невозможности создания перемещаемого объектного кода. Все команды МП ВМ80 оперируют абсолютными адресами, что требует настройки объектного кода перед его исполнением. Второй недостаток — отсутствие средств защиты памяти. Эти недостатки дают нам право отнести архитектуру МП ВМ80 к классу достаточно простых однопользовательских систем.

2.4. Структурная схема ВМ80

Схема однокристального 8-разрядного МП ВМ80 приведена на рис. 2.4, на рис. 2.5 дано его условное графическое обозначение.

Схема содержит АЛУ, блок регистров, устройство управления и буфер данных. Обмен информацией между составными частями схемы осуществляется с помощью 8-разрядной внутренней шины. По шине передаются команды, адреса, данные, а также информация SW (Status Word) о состоянии процессора в текущем машинном цикле.

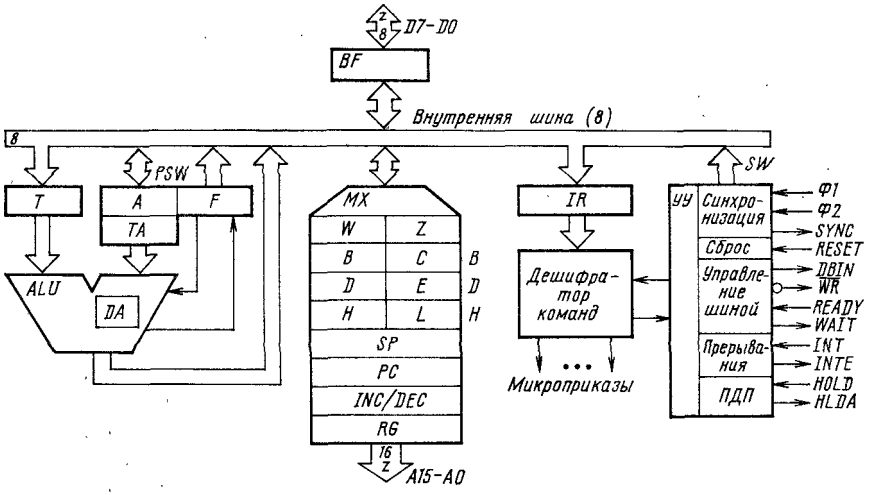


Рис. 2.4. Схема микропроцессора ВМ80

В состав интерфейса МП входят трехстабильная 16-разрядная шина адреса A15—A0, трехстабильная двунаправленная 8-разрядная мультиплексированная шина состояния/данных D7—D0, две линии двухфазной синхронизации Ф1, Ф2 и десять линий управления, из которых четыре входных, а шесть

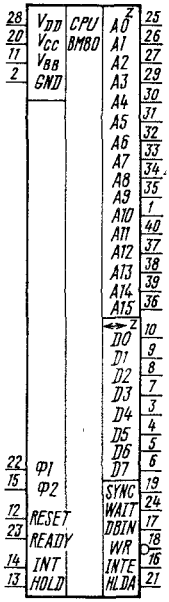


Рис. 2.5. Условное графическое обозначение микропроцессора ВМ80

выходных. Каждая линия управления имеет свое функциональное назначение:

RESET	Начальная установка
SYNC	Сигнал синхронизации по машинным циклам
DBIN	Строб ввода данных
WR	Строб вывода данных
READY	Сигнал готовности периферийных модулей к обмену
WAIT	Сигнал ожидания готовности
INTE	Разрешение прерывания
INT	Запрос векторного прерывания
HOLD	Запрос доступа к магистрали
HLDA	Подтверждение доступа к магистрали

Все действия МП синхронизированы вложенными друг в друга циклами трех уровней: командными, машинными и микротактами.

Командный цикл состоит из нескольких (от 1 до 5) машинных циклов, обозначаемых М1—М5. Каждый машинный цикл представляет собой цикл обращения к системной магистрали: выборку очередного байта команды или данных из памяти, запись в память, ввод или вывод данных. Исключение составляет команда DAD (сложить HL с парой регистров). Она содержит три машинных цикла, однако только в первом цикле М1 происходит обращение к памяти, тогда как два других отводятся для исполнения команды без обращения к магистрали. Цикл М1 всегда первый и возможно единственный цикл, который реализует фазу выборки команды. Его называют также циклом FETCH. В начале каждого машинного цикла МП вырабатывает сигнал SYNC. Реализация машинного цикла возлагается на устройство управления (УУ).

Машинные циклы выполняются по микротактам, определяемым как интервал времени между двумя соседними фронтами импульсов фазы Ф1. При частоте импульсов Ф1, равной 2 МГц, длительность микротакта составляет 500 нс. Один машинный цикл включает 3—5 микротактов Т1—Т5. В течение одного микротакта исполняется один микроприказ. Обычно цикл М1 длится 4—5 тактов, циклы М2—М5 содержат 3 такта. Сигнал SYNC выдается в первом микротакте каждого машинного цикла.

Для нормального функционирования МС недостаточно управляющих сигналов. Так, в данном наборе сигналов нельзя отличить циклы обращения к памяти от циклов ВВ. Расширение числа управляющих сигналов выполнено с помощью специального 8-разрядного слова состояния SW (Status Word), которое выдается через шину данных в первом такте Т1 каждого машинного цикла. Если сигналы управления и состояния используются в основном для управления процессом исполнения машинного цикла, то

сигналы SW кодируют тип машинного цикла. Последние определяют вид информации и направление ее движения через шину данных. Согласно условию $SYNC \cdot Ф1 = 1$ слово состояния SW фиксируется во внешнем регистре и используется для управления МС на уровне машинных циклов.

В состав SW входят следующие сигналы:

D0 (INTA)	Подтверждение прерывания. Выделяет машинные циклы обслуживания системы прерываний
D1 ($\overline{WO}$)	Запись/вывод. Свидетельствует о выдаче в данном цикле данных из МП
D2 (STACK)	Стек. Сигнал активен, если выполняется обращение к стеку, т. е. addr=(SP)
D3 (HLTA)	Подтверждение останова. Информирует о переходе МП в данном машинном цикле в состояние HALT
D4 (OUT)	Вывод. Свидетельствует об обращении к портам вывода. Адрес порта установлен на младшей и старшей половинах шины адреса
D5 (M1)	Цикл M1. Обозначает первый машинный цикл каждого командного цикла
D6 (INP)	Ввод. Свидетельствует об обращении к портам ввода. Адрес порта установлен на младшей и старшей половинах шины адреса
D7 (MEMR)	Чтение памяти. Выделяет циклы чтения данных из памяти

Всего возможно 11 наборов сигналов SW, разбивающих множество машинных циклов на 11 типов:

Тип	Мнемоника	Функциональное назначение
1	FETCH	Цикл M1 приема первого байта команды в регистр команд IR
2	MEMORY_READ	Цикл чтения данных из памяти по адресу, определяемому PC, BC, DE, HL или addr16
3	MEMORY_WRITE	Цикл записи данных в память по адресу, определяемому BC, DE, HL или addr16
4	STACK_READ	Цикл чтения из стека, чтение памяти по адресу, определяемому SP
5	STACK_WRITE	Цикл записи в стек, запись в память по адресу, определяемому SP
6	INPUT	Цикл ввода данных из порта в аккумулятор A
7	OUTPUT	Цикл вывода данных из аккумулятора в порт
8	INTERRUPT_M1	Первый цикл подтверждения прерывания
9	HALT	Цикл останова МП
10	HALT_INTERRUPT	Первый цикл подтверждения прерывания в состоянии HALT
11	INTERRUPT	Второй и третий циклы подтверждения прерывания

Описание содержания типовых циклов обращения к магистрали представлено в табл. 2.6.

Зная содержание допустимого множества машинных циклов, нетрудно построить их последовательности, реализующие ту или

Таблица 2.6

Слово состояния	FETCH	MEMORY READ	MEMORY WRITE	STACK READ
	1	2	3	4
D0 (INTA)	0	0	0	0
D1 (WO)	1	1	0	1
D2 (STACK)	0	0	0	1
D3 (HLTA)	0	0	0	0
D4 (OUT)	0	0	0	0
D5 (M1)	1	0	0	0
D6 (INP)	0	0	0	0
D7 (MEMR)	1	1	0	1
Команды шины управления	MEMR	MEMR	MEMW	MEMR

иную команду МП. Машинные циклы удобно представлять в следующем формате:

Номер цикла	Тип цикла	Состояние AB	Состояние DB
STAX B — запомнить содержимое A в памяти по адресу, определяемому регистровой парой BC:			
M1	FETCH	(PC)	⟨STAX B⟩
M2	MEMORY_WRITE	(BC)	(A)
CALL addr — вызов подпрограммы по адресу addr:			
M1	FETCH	(PC)	⟨CALL⟩
M2	MEMORY_READ	(PC) + 1	addr LB
M3	MEMORY_READ	(PC) + 2	addr HB
M4	STACK_WRITE	(SP) − 1	(PC) + 3 HB
M5	STACK_WRITE	(SP) − 2	(PC) + 3 LB
OUT port — вывод данных из аккумулятора в порт port:			
M1	FETCH	(PC)	⟨OUT⟩
M2	MEMORY_READ	(PC) + 1	port
M3	OUTPUT	port port	(A)

Во время выполнения каждого машинного цикла МП принимает или выдает информацию по шине данных. Направление передачи совпадает со стробами DBIN и $\overline{WR}$. В соответствии с этим все типы циклов можно разбить на циклы чтения и записи, связанные с вводом и выводом информации из МП:

Циклы чтения	Циклы записи
1 FETCH	
2 MEMORY_READ	3 MEMORY_WRITE
4 STACK_READ	5 STACK_WRITE
6 INPUT	7 OUTPUT
8 INTERRUPT_M1	
10 HALT_INTERRUPT	
11 INTERRUPT	

Временны́е диаграммы циклов приведены на рис. 2.6.

	STACK_ WRITE	INPUT	OUTPUT	INTERRUPT_MI	HALT	HALT_INTER-_RUPT	INTERRUPT
	5	6	7	8	9	10	11
	0	0	0	1	0	1	1
	0	1	0	1	1	1	1
	1	0	0	0	0	0	0
	0	0	0	0	1	1	0
	0	0	1	0	0	0	0
	0	0	0	1	0	1	0
	0	1	0	0	0	0	0
	0	0	0	0	1	0	0
	MEMW	I/OR	I/OW	INTA	—	INTA	INTA

В такте Т2 по срезу Ф2 выполняется проверка на активность сигнала READY. В случае его активности (READY=1) МП из такта Т2 переходит к такту Т3 обмена данными, реализуя синхронный протокол ВВ. Если же быстродействия внешнего модуля МС недостаточно для синхронного обмена с МП за один такт, он может задержать момент перехода к такту Т3 на целое число тактов TW. Для этого достаточно установить в 0 сигнал READY до момента его первой проверки и удерживать в этом состоянии то время, которое необходимо модулю для подготовки к обмену. В случае READY=0 МП из такта Т2 переходит в режим ожидания WAIT, о чем свидетельствует появление высокого уровня напряжения на линии WAIT. В данном режиме МП генерирует циклы TW ожидания сигнала READY и реализует асинхронный протокол обмена с внешними модулями МС. Продолжительность режима WAIT определяется длительностью действия напряжения низкого уровня на линии READY, которая контролируется в каждом такте TW по срезу Ф2.

При обнаружении высокого уровня напряжения сигнала READY МП из такта TW переходит к выполняющему обмен такту Т3. При этом сигнал WAIT устанавливается в 0. Для нормальной работы МП установку сигнала READY в 0 или 1 рекомендуется производить более чем за 120 нс до среза Ф2 и удерживать по крайней мере до его проверки.

Микропроцессор имеет несложные встроенные средства обслуживания запросов на прерывания с тремя управляющими сигналами:

INT	Запрос на прерывание
INTE	Разрешение прерывания
SW.INTA	Подтверждение прерывания

Управляемый программно с помощью команд EI и DI сигнал разрешения прерывания INTE маскирует вход INT запроса на прерывание. При условии INT·INTE=1, которое проверяется в

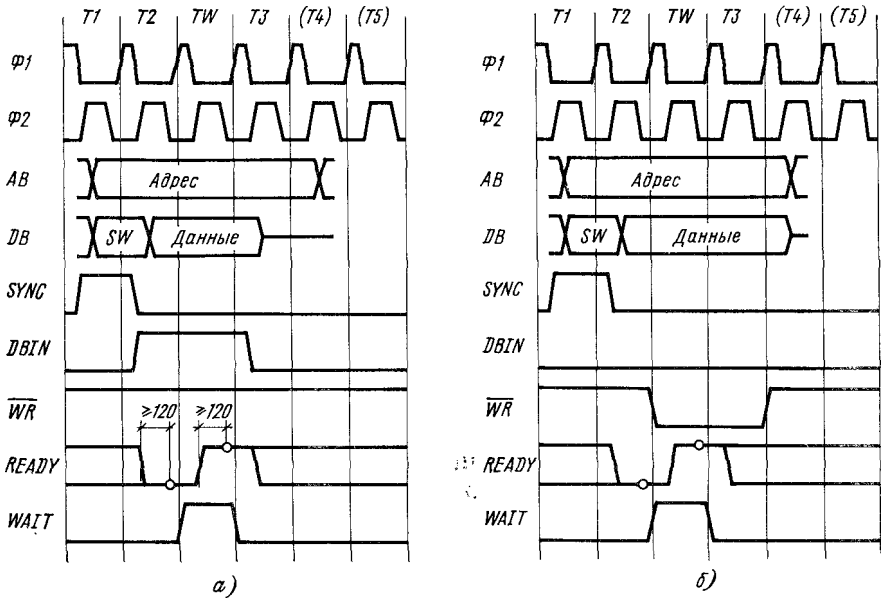


Рис. 2.6. Временны́е диаграммы циклов чтения (а) и записи (б) микропроцессора ВМ80

последнем такте каждого командного цикла (за исключением команды EI), МП должен перейти к процедуре обслуживания запроса. Благодаря этому гарантируется выполнение любой команды.

Процедура обслуживания запроса на прерывание заключается в идентификации источника прерывания, сохранении точки возврата в системном стеке и переходе к соответствующей подпрограмме обслуживания прерывания. Она начинается с машинного цикла INTERRUPT_M1, который похож на цикл FETCH, но отличается от него тем, что вместо сигнала SW.MEMR активен сигнал SW.INTA. Отличие самого цикла от цикла FETCH заключается также в том, что сбрасывается флаг INTE и PC не инкрементируется. Это обеспечивает сохранность точки возврата и игнорирование всех последующих запросов до специального разрешения командой EI. В остальном цикл INTERRUPT_M1 не отличается от цикла FETCH и состоит из ввода первого байта команды. Далее МП работает как обычно за исключением того, что при INT=1 сигнал SW.MEMR заменяется на SW.INTA, что предотвращает изменение PC.

Обычно в этом цикле генерируется команда CALL addr, в которой addr определяется источником прерывания и указывает на начало программы обслуживания. После выдачи команды CALL addr следует установить на линии INT напряжение низкого

уровня, что обеспечит переход к обычным машинным циклам с сигналом SW.MEMR. Поэтому в фазе исполнения команды CALL addr будут сформированы два цикла STACK—WRITE, которые обеспечат запись точки возврата прерванной программы в системный стек.

Для ускорения процедуры обработки прерываний применяется специальный укороченный вариант команды CALL addr — команда RST n, где addr = 8n, n = 0—7.

Использование команды RST n подразумевает резервирование первых 64 (8 × 8) байтов памяти под таблицу входов в подпрограммы обслуживания прерываний. Обычно по адресам 8n, n = 0—7, находятся команды JMP, передающие управление на подпрограммы обслуживания прерываний. Модификация адресной части команд JMP позволяет оперативно изменять входы в подпрограммы. Возврат из подпрограмм обслуживания прерывания обычно выполняется с помощью командной последовательности

```
EI          ;Разрешение прерывания, INTE ← 1
RET
```

В МП предусмотрены простейшие средства для управления захватом шины. Процедура захвата шины инициализируется внешними средствами установкой сигнала HOLD. Микропроцессор реагирует на этот сигнал по срезу Ф2. По фронту Ф1 в такте Т3 при вводе данных или в такте, следующем за Т3, при выводе данных МП формирует сигнал HLDA, а по фронту Ф2 в этом же такте шина данных переводится в высокоомное z-состояние. Шина адреса всегда переводится в z-состояние в такте, следующем за Т3. Такты Т4, Т5 и более (для команды DAD) совмещаются по времени с состоянием HOLD. Выход из состояния захвата производится после такта Т5 при обнаружении низкого уровня напряжения на линии HOLD. Затем МП переходит к исполнению такта Т1 следующего машинного цикла.

В цикле HALT обмена информации через шину данных не происходит. Этот цикл используется для вхождения в режим программного останова. Машинный цикл HALT следует за циклом FETCH, в котором принята команда HLT. При этом шина адреса и шина данных переводятся в z-состояние, а на выходе WAIT генерируется напряжение высокого уровня, как показано на рис. 2.7.

Находясь в состоянии HALT, МП не теряет возможности реагировать на сигнал HOLD, генерируя сигнал HLDA. Из состояния останова МП может быть выведен двумя путями: перезапуском сигналом RESET и прерыванием INT при разрешенной маске INTE. Временные диаграммы перезапуска сигналом RESET приведены на рис. 2.8. При перезапуске сбрасывается

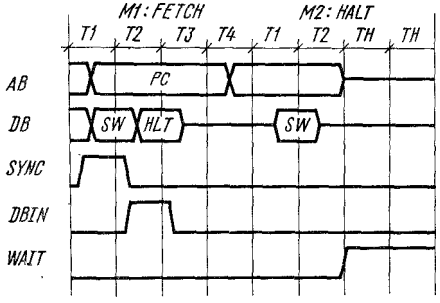


Рис. 2.7. Временны́е диаграммы машинного цикла HALT

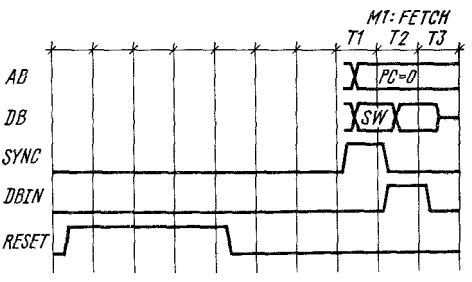


Рис. 2.8. Временны́е диаграммы перезапуска микропроцессора ВМ80

маска INTE и управление передается на стартовую ячейку, в качестве которой применяется ячейка с нулевым адресом.

2.5. Базовый комплект БИС серии КР580

Интерфейс однокристального МП, как правило, не обеспечивает его непосредственного соединения с системной магистралью по ряду причин. Во-первых, состав шин и логика управления обменом интерфейса МП и системной магистрали отличаются. Во-вторых, однокристальные МП имеют малые токовую и емкостную нагрузки. Это объясняется ограничениями, накладываемыми на максимальную рассеиваемую мощность кристалла (обычно 1,5—2 Вт). Кроме того, выходные буферы МП ВМ80 обеспечивают токовую нагрузку I_{OL} до 1,6 мА и емкостную C_L до 100 пФ, что недостаточно для больших систем.

Таким образом, для построения законченного модуля ЦП требуются дополнительные схемы. Важнейшими среди них являются ГТИ, системный контроллер, буферные регистры и шинные формирователи.

Генератор тактовых импульсов КР580ГФ24. Структура ГТИ и его условное обозначение приведены на рис. 2.9, временны́е диаграммы работы схемы — на рис. 2.10.

Выводы XTAL1 и XTAL2 служат для подключения кварцевого резонатора, а вывод TANK — для выбора его гармоники. Выход генератора буферизирован и выведен на линию OSC так, чтобы другие устройства МС могли его использовать. Основой схемы является делитель на 9, поэтому частота кварцевого резонатора должна быть в 9 раз больше, чем частота следования импульсов Ф1, Ф2. Частота кварцевого резонатора определяется быстродействием МС и лежит в пределах 4,5—22,5 МГц. При OSC = 22,5 МГц период следования синхроимпульсов Ф1 и Ф2 (длительность микротакта) 400 нс, что соответствует нижней границе диапазона, допустимого для МП ВМ80.

В состав схемы КР580ГФ24 входят также логические цепи для генерации строба $\overline{\text{STSTB}}$ (Status Strobe), синхронизации сигнала RDYIN (Ready Input) и формирования мощного выхода RESET, служащего для аппаратного сброса МС в исходное состояние.

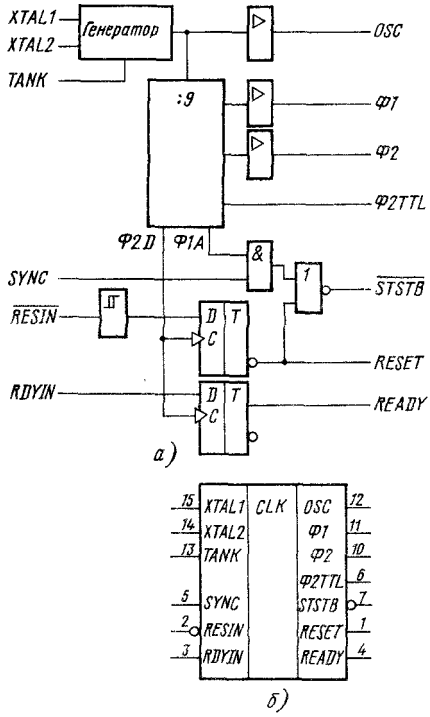


Рис. 2.9. Генератор тактовых импульсов КР580ГФ24:
а — структурная схема; б — условное графическое обозначение

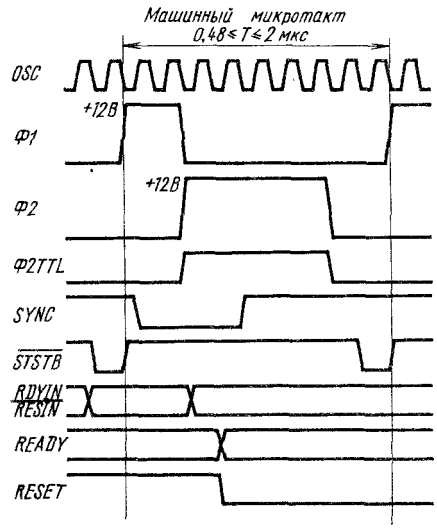


Рис. 2.10. Временные диаграммы работы генератора тактовых импульсов КР580ГФ24

Системный контроллер и формирователь шины КР580ВК28/ВК38. Специальная БИС типа КР580ВК28/ВК38 представляет собой системный контроллер и формирователь шины данных для МС на базе МП ВМ80. Схема формирует базовый набор управляющих стробов магистрали типа И41 [33] и обеспечивает двунаправленную буферизацию шины данных МП от основной памяти и устройств ВВ. В БИС использована биполярная ТТЛШ-технология, гарантирующая небольшие задержки и высокую нагрузочную способность.

Структурная схема системного контроллера КР580ВК28/ВК38 и его условное графическое обозначение приведены на рис. 2.11. Двунаправленный 8-разрядный шинный формирователь обеспечивает выход DB7—DB0 со стороны системной магистрали с током нагрузки до 10 мА и емкостью нагрузки до 100 пФ, а также изолирует шину данных МП D7—D0 от системной. Задержка, вносимая формирователем в шину данных, не превышает 40 нс. Формирователь выполнен по схеме с тремя состояниями.

В состав контроллера входит регистр-защелка, который по стробу $\overline{\text{STSTB}}$ фиксирует слово состояния SW, выдаваемое МП в начале каждого машинного цикла. Слово состояния определяет тип текущего машинного цикла в соответствии с табл. 2.8, в зависимости от которого логическая схема контроллера формирует один из пяти управляющих стробов системной магистрали: $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{I/OR}}$,

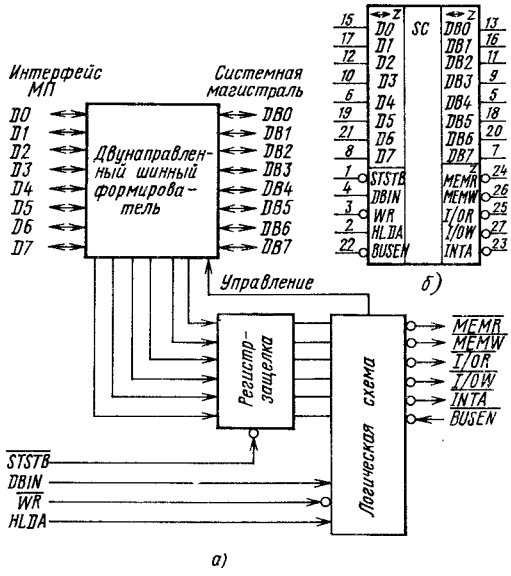


Рис. 2.11. Системный контроллер КР580ВК28/ВК38:
а — структурная схема; б — условное графическое обозначение

$\overline{\text{I/OW}}$, $\overline{\text{INTA}}$, также указанных в табл. 2.6. Временны́е диаграммы их генерации приведены на рис. 2.12 и зависят от длительности стробов DBIN и $\overline{\text{WR}}$ микропроцессора. Системный контроллер КР580ВК38 (ВК38) в отличие от КР580ВК28 (ВК28) формирует опережающие стробы $\overline{\text{MEMW}}$ и $\overline{\text{I/OW}}$ (см. § 1.3).

Строб $\overline{\text{INTA}}$ обычно используется для выбора порта вектора прерывания, изолированного от пространства ВВ. В МС, где требуется только один вектор прерываний, устройство КР580ВК28/ВК38 может автоматически в необходимый момент времени выдать команду RST7 на шину данных D7 — D0 без каких-либо дополнительных логических схем. Для этого выход $\overline{\text{INTA}}$ следует соединить с источником питания +12 В через резистр сопротивлением 1 кОм.

Трехстабильные выходные буферы шины данных и управляющих сигналов открываются асинхронно входным сигналом $\overline{\text{BUSEN}}$. При $\overline{\text{BUSEN}}=1$ буферы находятся в состоянии с высоким выходным сопротивлением.

Буферный регистр КР580ИР82/ИР83. Выполненный по биполярной технологии буферный регистр КР580ИР82/ИР83 используется в качестве 8-разрядного фиксатора или буфера. На рис. 2.13 представлены схема и условное обозначение регистра КР580ИР82, в отличие от которого регистр КР580ИР83 имеет инверсную выходную шину.

Основой схемы является 8-разрядный регистр-защелка со статическим синхровходом STB (Strobe). Запись данных в регистр разрешена при STB = 1. В противном случае регистр находится в режиме хранения. На выходе регистра имеется трехстабильный буфер, управляемый сигналом $\overline{\text{OE}}$ (Output Enable). Буфер обеспечивает выходной ток до 32 мА и емкость нагрузки до 300 пФ. Если управляющий сигнал $\overline{\text{OE}}$ активен, то данные регистра передаются на выход микросхемы. При $\overline{\text{OE}}=1$ выходной буфер закрыт и находится в высокоомном состоянии. Временны́е диаграммы работы регистров представлены на рис. 2.14.

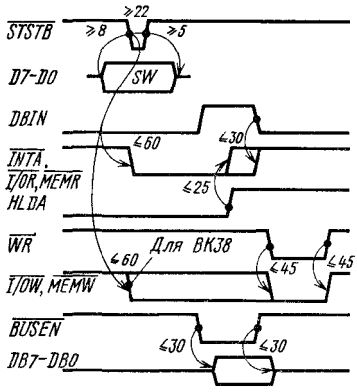


Рис. 2.12. Временны́е диаграммы работы системного контроллера КР580ВК28/ВК38

Шинный формирователь КР580ВА86/ВА87. Биполярная микросхема КР580ВА86/ВА87 предназначена для реализации 8-разрядных одно- и двунаправленных буферных схем с тремя состояниями на выходе. На рис. 2.15 приведены структурная схема и условное графическое обозначение БИС КР580ВА86. Формирователь КР580ВА87 отличается тем, что имеет инвертирующие буферы.

Трехстабильные буферные схемы выбираются только при низком уровне напряжения на линии $\overline{OE}$. При этом, если на входе T (Transmitter) высокий уровень напряжения, то открывается буфер для передачи из канала A в B. В противном случае осуществляется передача в обратном направлении. Временны́е диаграммы работы буфера представлены на рис. 2.16.

Восьмиразрядные каналы A и B шинного формирователя не эквивалентны. Так, если со стороны канала A ток нагрузки не должен превышать 16 мА, а емкость нагрузки 100 пФ, то со стороны канала B обеспечивается более высокая нагрузочная способность: максимальный ток нагрузки 32 мА, емкость нагрузки не более 300 пФ.

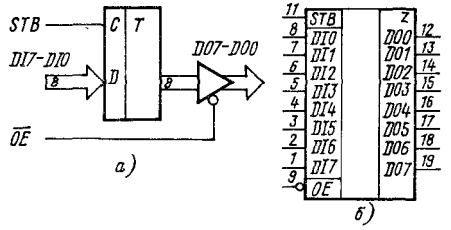


Рис. 2.13. Буферный регистр КР580ИР82:
а — структурная схема; б — условное графическое обозначение

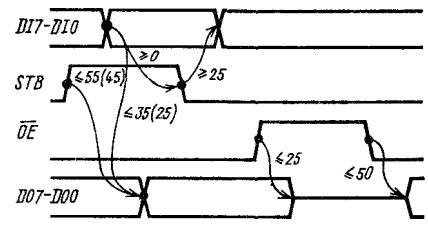


Рис. 2.14. Временны́е диаграммы работы регистров КР580ИР82/ИР83

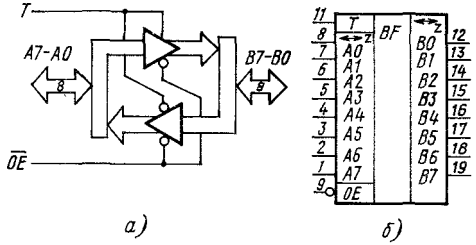


Рис. 2.15. Шинный формирователь КР580ВА86:
а — структурная схема; б — условное графическое обозначение

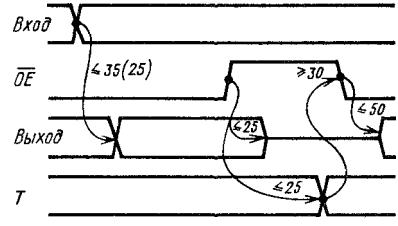


Рис. 2.16. Временны́е диаграммы работы формирователей КР580ВА86/ВА87

2.6. Центральный процессор на базе БИС серии КР580

Схема ЦП на базе БИС серии КР580 приведена на рис. 2.17 [23, 53]. В простейшем случае она может содержать только три кристалла: МП КР580ВМ80, генератор тактовых импульсов КР580ГФ24 и системный контроллер КР580ВК28/ВК38, применение которых гарантирует компактность ЦП при полном сохранении всех функциональных возможностей МП.

Чтобы выполнить функцию начальной установки МП, необходимо ко входу $\overline{\text{RESIN}}$ ГТИ подключать RC-цепочку. Наличие гистерезиса на входе $\overline{\text{RESIN}}$ (см. рис. 2.9, *а*) гарантирует быстрое переключение схемы в момент достижения порога переключения. Постоянная времени RC-цепочки рассчитывается из требований, накладываемых на длительность сигнала RESET.

Период следования синхросигналов Ф1, Ф2 микропроцессора КР580ВМ80 определяется кварцевым резонатором ZQ1. При выборе и установке кварцевого резонатора следует обеспечить минимальное сопротивление внешней цепи генератора XTAL2—XTAL1. Со стороны кварцевого резонатора генератор представляет индуктивный элемент, приводящий к снижению рабочей частоты. Для компенсации данного эффекта рекомендуется последовательно с кварцевым резонатором со стороны вывода XTAL2 включить конденсатор небольшой емкости. Для кварцевого резонатора частотой 18 МГц емкость 15—30 пФ. При повышении частоты может потребоваться уменьшение емкости конденсатора.

Кроме синхросигналов Ф1, Ф2 генератор формирует также системный сигнал CCLK (ток нагрузки $I_{0L} \leqslant 15$ мА) и сигнал системного сброса INIT ($I_{0L} \leqslant 15$ мА). Сигнал INIT предназначен для начальной установки системы в исходное состояние. Он реализуется при наличии сигнала, формируемого RC-цепочкой на входе $\overline{\text{RESIN}}$ генератора. Применяемые в схеме на рис. 2.17 элементы обеспечивают длительность сигнала сброса 0,35 с.

Сигнал подтверждения XACK используется при работе с медленными модулями памяти и ВВ. Он может быть задействован также для организации ожидания ЦП некоторого внешнего события. Микропроцессор КР580ВМ80 будет переведен в состояние ожидания WAIT, если сигнал XACK окажется неактивным в момент первой его проверки (начало второй трети такта Т2). В зависимости от размера и характеристик системы сигнал подтверждения может быть организован одним из двух способов.

В больших системах линия подтверждения XACK обычно выполняется с низким уровнем активности, что обеспечивает асинхронный способ доступа к модулям памяти и ВВ. Для этого перед входом RDYIN генератора следует поставить дополнительный инвертор. После того как выбранное устройство получит

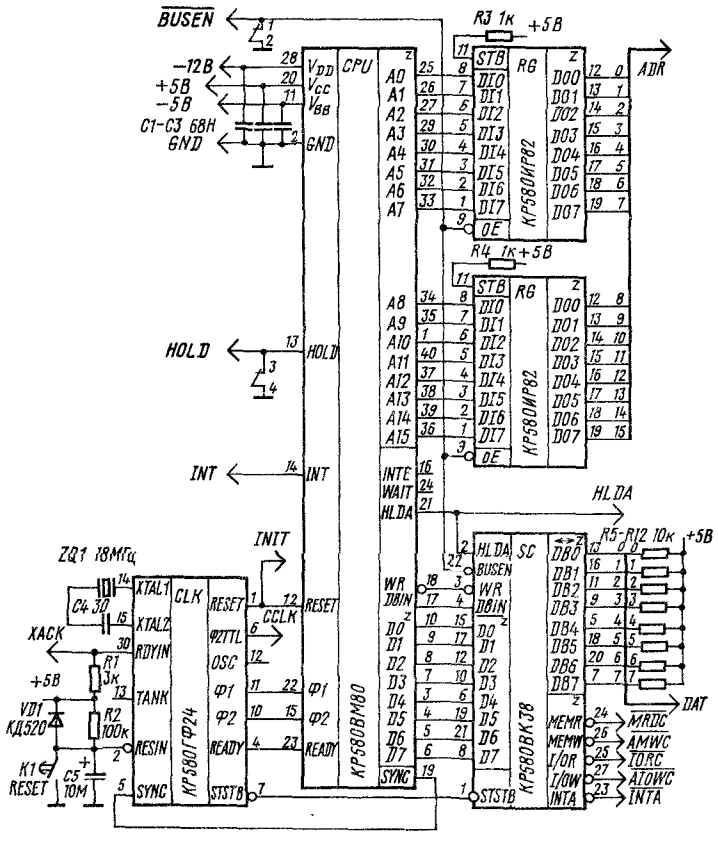


Рис. 2.17. Схема центрального процессора на базе БИС серии КР580

строб чтения или записи, оно генерирует сигнал подтверждения, формируемый на линии ХАСК по схеме с открытым коллектором. Для максимальной производительности системы ее модули должны возвращать сигнал подтверждения до его первой проверки.

В малых системах с быстродействующими устройствами рекомендуется использовать линию ХАСК с высоким уровнем активности. Процессор работает в синхронном режиме без тактов ожидания с максимальной для него скоростью. Следовательно, в устройствах, успевающих выполнить операцию в синхронном режиме, можно опустить логику подтверждения обмена, тем самым упростив их.

В схеме на рис. 2.17 использован системный контроллер типа КР580ВК38. От контроллера КР580ВК28 он отличается тем, что формирует упрежденные стробы $\overline{\text{I/OW}}$ и $\overline{\text{MEMW}}$ (см. рис. 2.12), рассчитанные на периферийные приборы второго

поколения. В контроллере КР580ВК28 эти стробы представляют собой задержанный на 4—45 нс строб $\overline{WR}$ микропроцессора КР580ВМ80, который генерируется за время такта Т3, т. е. уже после первой проверки сигнала подтверждения ХАСК.

Как показано на рис. 2.12, все командные стробы КР580ВК38 связаны со стробом STSTB и задержаны относительно его начала на 20—60 нс. Так как STSTB опережает на Т/3 фазу Ф2 такта Т2, то до момента первой проверки сигнала ХАСК еще остается 4Т/9—60 нс. При Т=400 нс это время примерно равно 120 нс, что вполне достаточно для сброса сигнала ХАСК. Временна́я диаграмма работы ЦП приведена на рис. 2.18.

Таким образом, применение контроллера КР580ВК28 возможно либо в системах с синхронным обменом, либо в системах асинхронного доступа с инверсной линией ХАСК. В МС, допускающих оба протокола обмена, рекомендуется использовать микросхему КР580ВК38. Во всех случаях нагрузочная способность командных линий $\overline{MRDC}$, $\overline{MWTC}$, $\overline{IORC}$ и $\overline{IOWC}$ следующая: $I_{0L} \leqslant 10$ мА, $C_L \leqslant 100$ пФ. Допустимый ток I_{0L} и емкость C_L линии $\overline{INTA}$ равны 5 мА и 100 пФ.

Системный контроллер обеспечивает также двунаправленную буферизацию шины данных МП без инверсии, доводя I_{0L} до 10 мА и C_L до 100 пФ. Дополнительная задержка в шине данных около 30 нс.

Удобно, но не обязательно системную шину данных DAT7—DAT0 подключать к источнику питания +5 В через резисторы 10 кОм. Это практически не скажется на нагрузочной способности шины. С другой стороны, чтение команды из несуществующей области памяти приведет к выполнению команды RST 7, это может быть использовано в отладочных и диагностических целях.

Адресная шина МП ВМ80 обеспечивает ток нагрузки I_{0L}= 1,8 мА и емкость нагрузки C_L до 100 пФ. Этой нагрузочной способности достаточно для построения одноплатного МК закрытого типа. В открытых для расширения МС требуется дополнительная буферизация адресной шины, которая может быть выполнена с помощью двух буферных регистров КР580ИР82 без инверсии. Буферизация адреса позволяет увеличить максимальные значения I_{0L} до 32 мА и C_L до 300 пФ. Вместе с этим в шине появляется дополнительная задержка 35 нс.

Возможна реализация адресного буфера на других микросхемах, например КР580ИР83, КР580ВА86/ВА87, К589ИР12, К589АП16/АП26 и т. д. Они имеют другие нагрузочные и временны́е характеристики и могут приводить к инверсии адреса.

Три сигнала ЦП — запрос на захват шины HOLD, подтверждение захвата HLDA (I_{0L}=1,8 мА) и разрешение шины BUSEN — служат для организации доступа к магистрали со стороны других активных модулей системы. Частным случаем такого модуля

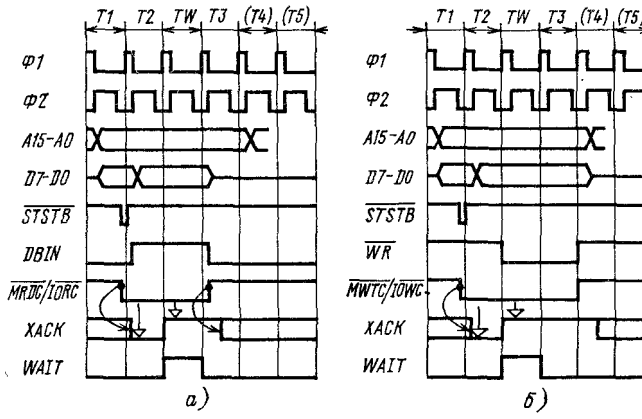


Рис. 2.18. Временные диаграммы циклов чтения (а) и записи (б) центрального процессора

является устройство с прямым доступом к памяти, управляемое БИС КР580ВТ57. Для запуска ЦП в системах, которые не имеют других активных модулей кроме ЦП, необходимо установить перемычки 1-2, 3-4.

2.7. Организация ВМ85А

Архитектурные отличия от ВМ80. Однокристальный МП К1821ВМ85А (ВМ85А) является усовершенствованным 8-разрядным МП, имеющим полную архитектурную совместимость со своим предшественником КР580ВМ80 [65].

Разработка нового МП была направлена на повышение производительности, скоростных характеристик и уменьшение числа БИС, необходимых для построения законченной МС. В результате удалось на одном кристалле кремния разместить устройство, функционально эквивалентное трем БИС: микропроцессору КР580ВМ80, генератору тактовых импульсов КР580ГФ24 и системному контроллеру КР580ВК28/ВК38.

С точки зрения программиста архитектура ВМ85А практически тождественна архитектуре ВМ80. В нем используется тот же самый набор программно-доступных регистров аккумуляторного типа (см. рис. 2.2). Сохранена логическая организация памяти и пространства ВВ. Система команд ВМ85А включает весь набор команд ВМ80 в их старой кодировке, что гарантирует полную совместимость с ПО микропроцессора ВМ80 на уровне объектного кода. Вместе с этим в состав системы команд ВМ85А введены две новые, что связано с расширением средств обработки прерываний и последовательным ВВ:

```
RIM             ;Чтение маски прерываний
SIM             ;Установка маски прерываний
```

Полный набор команд, включая новые, их характеристики и кодировка приведены в табл. 2.1—2.5.

Методология построения системы прерываний ВМ85А подчинена архитектуре ВМ80, однако число возможных источников прерываний на аппаратном уровне увеличено с одного до пяти. Наряду с типовым векторным запросом INTR (INT для ВМ80) введены еще четыре, имеющие фиксированные векторы прерываний. Это означает, что при появлении соответствующего запроса управление передается на ячейку с фиксированным адресом, приведенным в табл. 2.7. Из таблицы видно, что стартовые адреса подпрограмм обслуживания прерываний находятся в области точек входа по команде RST n, n=0—7, но расположены посередине между ними. Это разъясняет ряд наименований, принятых для запросов RST n.5, n=5—7.

Все запросы, за исключением TRAP, могут быть запрещены или разрешены одновременно с помощью команд EI и DI, управляющих общим флагом разрешения прерываний IEN. Существует также возможность раздельного маскирования запросов типа RST независимо друг от друга, которое выполняется с помощью новой команды SIM. По команде SIM обеспечивается установка нового состояния маски в соответствии с содержимым аккумулятора. При выполнении этой команды содержимое аккумулятора интерпретируется следующим образом:

A0	M5.5	Маска RST 5.5
A1	M6.5	Маска RST 6.5
A2	M7.5	Маска RST 7.5
A3	MSEN	Разрешение установки маски
A4	R7.5	Сброс триггера приема запроса RST 7.5
A5	—	Не используется
A6	SDEN	Разрешение вывода данных
A7	SOD	Последовательные данные для вывода через SOD

Установленная маска запрещает соответствующее прерывание. Смена маски в соответствии с A0—A2 выполняется только при разрешении ее установки: MSEN=1. В противном случае функция установки маски подавляется.

Таблица 2.7

Имя	Приоритет	Стартовый адрес	Вид сигнала
TRAP	1	24H	Переход из 0 в 1, затем 1
RST 7.5	2	3CH	Переход из 0 в 1
RST 6.5	3	34H	1
RST 5.5	4	2CH	1
INTR	5	Вводится при подтверждении прерывания	1

Текущее состояние масок прерываний может быть прочитано по команде RIM. Команда пересылает текущее состояние масок в аккумулятор в соответствии со следующим распределением:

A0	M5.5	Маска RST 5.5
A1	M6.5	Маска RST 6.5
A2	M7.5	Маска RST 7.5
A3	IEN	Флаг разрешения прерывания
A4	D5.5	Флаг запроса RST 5.5
A5	D6.5	Флаг запроса RST 6.5
A6	D7.5	Флаг запроса RST 7.5
A7	SID	Последовательные данные ввода через SID

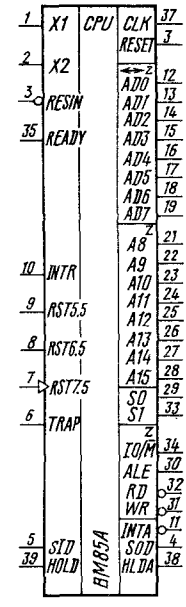
При включении источника питания или пуске МП все индивидуальные маски устанавливаются, а флаг разрешения прерывания IEN сбрасывается, что приводит к запрету прерываний. Прием какого-либо запроса на прерывание тоже вызывает общее запрещение всех маскируемых прерываний, но при сохранении состояния индивидуальных масок. Кроме этого запрос TRAP сохраняет и состояние флага общего разрешения, которое может быть прочитано только первой командой RIM. Вторая команда дает результат текущего состояния маски, показывающего общий запрет прерываний.

Запросы на прерывания строго упорядочены, как это показано в табл. 2.7. Высшим приоритетом обладает немаскируемый запрос TRAP, низший присвоен векторному прерыванию INTR. Установленная приоритетная схема разрешает конфликт при одновременном появлении нескольких запросов и не учитывает текущего приоритета программы. Так, прерывание с более низким приоритетом, разрешенное во время исполнения процедуры обслуживания запроса с более высоким приоритетом, может остановить последнюю, не учитывая соотношения приоритетов.

Еще одно важное отличие архитектуры BM85A от BM80 состоит в поддержке последовательной линии ВВ. В BM85A имеется возможность программного доступа к изолированному однобитовому пространству ВВ, реализуемого с помощью команд RIM и SIM. При исполнении данных команд наряду с чтением или установкой регистра масок производится ввод данных с линии в старший разряд аккумулятора или их вывод в обратном направлении. Вывод данных осуществляется только при установке флага разрешения вывода данных SDEN. В противном случае функция вывода подавляется. Управление функциями вывода и установки маски прерывания независимыми флажками SDEN и MSEN соответственно позволяет выполнить ВВ, не затрагивая систему прерываний.

Структурная схема BM85A. Если на уровне архитектуры МП BM85A аналогичен BM80, за исключением отмеченных выше отличий, то на физическом уровне они значительно отличаются друг от друга [43, 56].

Рис. 2.20. Условное графическое обозначение микропроцессора ВМ85А



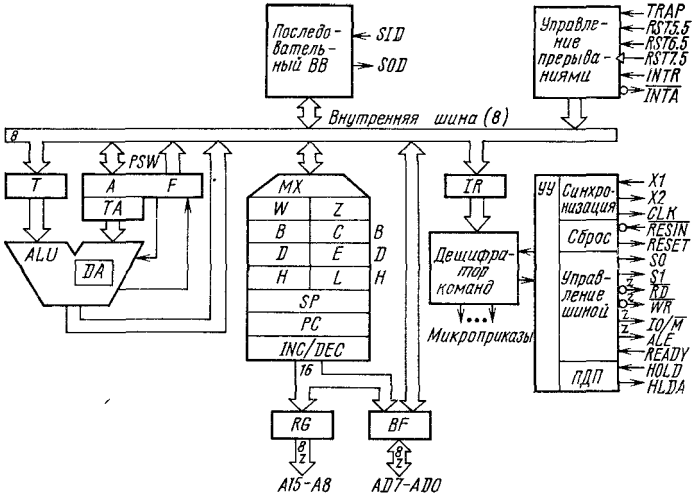


Рис. 2.19. Схема микропроцессора ВМ85А

Схема МП ВМ85А приведена на рис. 2.19, а его условное графическое изображение — на рис. 2.20.

По сравнению со схемой МП ВМ80 (см. рис. 2.4) в схеме ВМ85А используется совмещенная шина адреса/данных AD7—AD0, по которой передаются младшая часть адресной информации и 8-разрядные данные. Старшая часть адреса фиксируется в регистре адреса и выводится на шину A15—A8. Существенно видоизменен набор линий шины управления, которые обеспечивают прямое подключение кварцевого резонатора, а также управление периферийными БИС памяти и ВВ. Расширен и модифицирован состав физических линий для поддержки системы прерываний. Введен блок последовательного ВВ. Приведем физический интерфейс ВМ85А.

AD7—AD0	Двунаправленная трехстабильная мультиплексированная шина младшей части адреса и данных
A15—A8	Трехстабильная шина вывода старшей части адреса
X1, X2	Вход и выход усилителя для подключения внешнего кварцевого резонатора или RC-цепочки. Вход X1 может быть использован для приема внешних тактовых импульсов
CLK	Выход тактовых импульсов
RESIN	Вход для приема сигнала сброса МП в начальное состояние. По сигналу $\overline{\text{RESIN}}$ PC принимает нулевое значение, сбрасываются триггеры разрешения прерывания и состояния HLDA
RESET	Выходной сигнал системного сброса, синхронизированный тактовыми импульсами CLK

S0, S1	Состояние МП:	
	S1	S0 Назначение
	0	0 HALT (останов)
	0	1 WRITE (запись)
	1	0 READ (чтение)
	1	1 FETCH (выборка команды)
	Линия S1 может быть использована в качестве упреждающего сигнала $R/\overline{W}$	
$\overline{RD}$, $\overline{WR}$	Трехстабильные линии для вывода стробов чтения и записи. Данные действительны в конце строба	
$IO/\overline{M}$	Линия выбора системы памяти или устройств ВВ Имеет три состояния	
ALE	Строб разрешения фиксации адреса. Адрес действителен на срезе строба. Линия имеет три состояния. Может быть использован для стробирования информации о состоянии	
READY	Линия для приема подтверждения обмена во время стробов RD и WR	
INTR	Линия запроса векторного прерывания, который вызывает генерацию строба INTA. Предусмотрены программные средства запрещения (разрешения) приема сигнала. При сбросе прием запроса запрещен	
$\overline{INTA}$	Выходная линия для генерации строба подтверждения векторного прерывания после завершения текущего командного цикла. Используется аналогично стробу RD для приема вектора прерывания	
RST 5.5, RST 6.5, RST 7.5	Входы для принятия запросов прерывания типа RST n, n=5.5, 7.5 и 7.5 соответственно. Вход RST 7.5 имеет высший приоритет в группе. Приоритет группы выше приоритета INTR. Прерывания могут быть замаскированы независимо друг от друга	
TRAP	Вход немаскируемого прерывания типа RST n, n=4.5, высшего приоритета	
SID, SOD	Вход и выход последовательной передачи данных. Входные данные загружаются в старший разряд аккумулятора A7 по команде RIM, вывод данных осуществляется из A7 по команде SIM	
HOLD	Линия запроса захвата шины внешним модулем	
HLDA	Линия подтверждения захвата шины, активизируется в ответ на сигнал HOLD в конце текущего машинного цикла. При этом линии адреса/данных, а также RD, WR, $IO/\overline{M}$ и ALE переводятся в третье состояние	

Работа МП синхронизируется внешним кварцевым резонатором или RC-цепочкой, подключаемой непосредственно к выводам

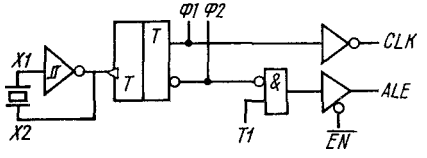


Рис. 2.21. Схема синхронизации микропроцессора BM85A

X1 и X2. Схема синхронизации представлена на рис. 2.21. Из схемы видно, что фронт сигнала на входе X1 переключает счетный триггер, который формирует две последовательности несовпадающих импульсов Ф1 и Ф2, используемые для тактирования внутренних схем МП. Внешний сигнал синхронизации CLK совпадает по фазе с импульсами Ф2. Сигнал ALE представляет собой один импульс Ф1, выделяемый в такте T1 каждого машинного цикла. Он является синхросигналом начала машинного цикла.

Основная тактовая частота сигнала CLK микропроцессоров ВМ85А и 8085А равна 3 МГц, что обеспечивает более высокую производительность, чем у стандартного МП ВМ80 с тактовой частотой 2,5 МГц. Существуют также БИС, рассчитанные на работу с частотой 5 МГц, например 8085А-2, что дополнительно повышает производительность МП данного типа.

В каждом машинном цикле МП обращается к магистрали для ввода или вывода одного байта информации, согласно временны́м диаграммам, изображенным на рис. 2.22. Работа канала синхронизируется стробами ALE, RD, WR и соответствует типовому протоколу на двухшинную магистраль (ср. с рис. 1.10). Каждый машинный цикл или цикл обращения к внешнему каналу содержит от 3 до 5 периодов T1—T5 сигнала CLK основной тактовой частоты. К этим тактам может быть добавлено произвольное число тактов ожидания готовности канала TW, которые включаются между T2 и T3. Непосредственно для ВВ информации отводятся лишь первые три такта совместно с тактами ожидания готовности. В такте T1 производится вывод адресной информации, а в тактах T2 и T3 — обмен данными.

При необходимости добавляются еще один или два такта для реализации операций внутри МП. В это время канал не используется.

Линия READY служит для организации обмена с медленными устройствами. При READY=1 реализуется синхронный режим работы, характеризующийся максимальной скоростью обмена без тактов ожидания, которую обеспечивает МП. В этом случае длительность стробов RD, WR минимальна и составляет 1,5T — 80 нс, где T — период CLK. Стробы задержаны на 50 нс относительно начала T2.

Проверка активности сигнала READY выполняется в середине T2 и всех следующих за ним тактов TW. Для организации асинхронного доступа этот сигнал должен быть установлен в 0 за 110 нс и удерживаться в таком состоянии вплоть до момента его первой проверки. Эти же временны́е ограничения характерны и для процесса установки сигнала готовности. Манипуляция сигналом READY дает возможность удлинить строб RD или WR до (1,5+N)T — 80 нс, где N — целое число периодов ожидания TW,

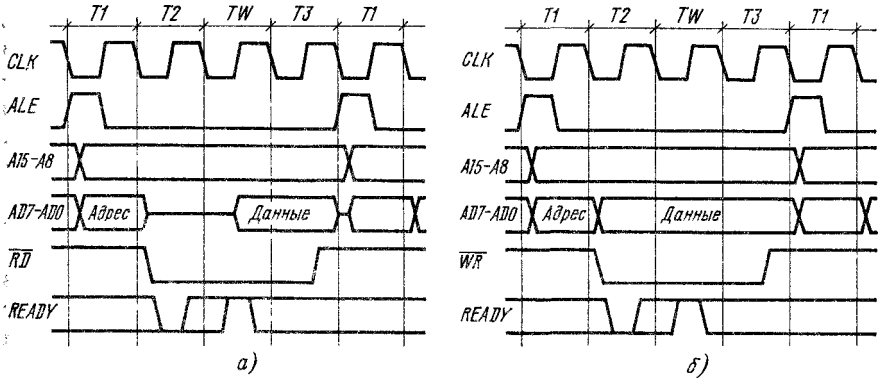


Рис. 2.22. Временны́е диаграммы циклов чтения (а) и записи (б) микропроцессора ВМ85А

обеспечив надежный обмен с медленной памятью или портами ВВ.

Особенностью процедур ВВ служит тот факт, что данные действительны только на срезе стробов RD и WR, т. е. протокол МП ВМ85А предполагает использование периферийных БИС второго поколения. Времена предустановки и удержания данных при выводе принимают значения $t_{DW} \geqslant 420$ нс, $t_{WD} \geqslant 80$ нс соответственно. Аналогично для цикла чтения $t_{DR} \geqslant 120$ нс, $t_{RDH} \geqslant 0$ нс.

Каждый командный цикл включает от одного до пяти машинных циклов М1—М5. Синхронизацию по командным циклам можно получить, выделяя с помощью сигналов состояния S1 и S0 все циклы М1 (FETCH). Этому типу цикла соответствует состояние S1 = S0 = 1. На рис. 2.23 приведен пример командного цикла IN port при работе без тактов ожидания. Из временны́х диаграмм видно, что команда выполняется за три цикла обращения к каналу: М1 — выборка кода операции, М2 — чтение из памяти второго байта команды и М3 — выдача содержимого аккумулятора в порт ВВ.

Следует отметить, что, подобно ВМ80, микропроцессор ВМ85А выводит 8-разрядный адрес порта как на старшую, так и на младшую половину 16-разрядной шины адреса. Модули ВВ могут быть ориентированы на использование только старшей половины шины А15—А8 и, следовательно, отпадет необходимость во внешнем адресном регистре для приема адреса по сигналу ALE.

Длительность командных циклов ВМ85А в периодах основной тактовой частоты отличается от длительности исполнения соответствующих команд в МП ВМ80. Эти отличия можно проследить по табл. 2.1—2.5, где для каждой команды указано число тактов, необходимых для ее исполнения как в одном, так и в другом

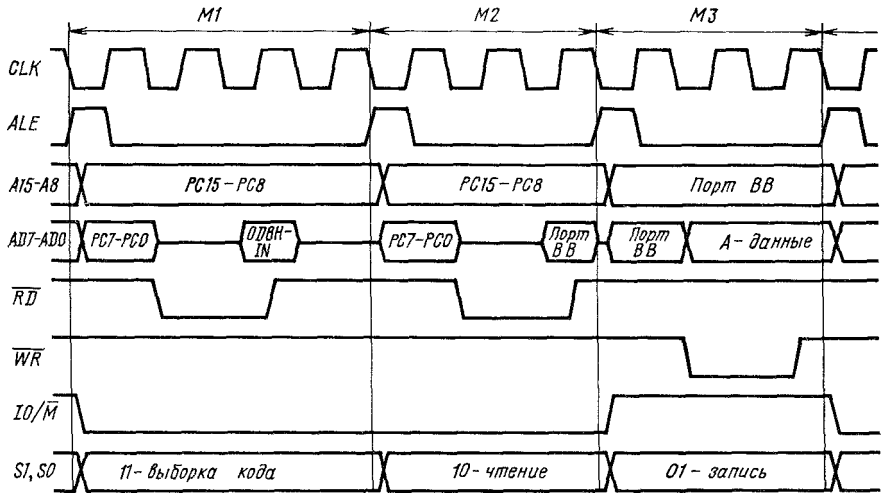


Рис. 2.23. Временны́е диаграммы командного цикла IN port

случае. Одни команды требуют меньшего, другие — большего числа тактов, что объясняется изменениями, внесенными в физическую организацию МП ВМ85А.

Как уже отмечалось, МП ВМ85А имеет пять линий для приема запросов на прерывание: INTR, RST 5.5, RST 6.5, RST 7.5 и TRAP. Линия INTR по своим функциям аналогична линии INT микропроцессора ВМ80. В ответ на запрос INTR генерируется один или три машинных цикла INTA с временны́ми диаграммами, эквивалентными циклу RD без тактов ожидания. Внешняя аппаратура отвечает на циклы INTA генерацией команды либо типа RSTn, n=0—7 (случай одного цикла INTA), либо типа CALL addr (случай трех циклов INTA), обеспечивая передачу управления на подпрограмму обслуживания прерывания.

Согласно методике приема запросов все входы можно разделить на три группы. К первой относятся входы статического типа RST 5.5, RST 6.5, а также INTR. Запрос на прерывание по этим входам фиксируется каждый раз, когда на них при сброшенной маске обнаружено напряжение высокого уровня. Время предустановки сигнала до начала Т1 цикла М1 $t_{INS} \geqslant 360$ нс и время его удержания $t_{INH} \geqslant 0$ нс. Для предотвращения повторной фиксации одного и того же запроса сигнал прерывания по статическому входу должен быть снят, прежде чем будет сброшена соответствующая маска.

Ко второй группе относится вход RST 7.5, который является входом динамического типа и фиксирует запрос на прерывание при каждом переходе сигнала из 0 в 1, даже при установленной маске и запрещенных прерываниях. Запрос сохраняется до тех пор,

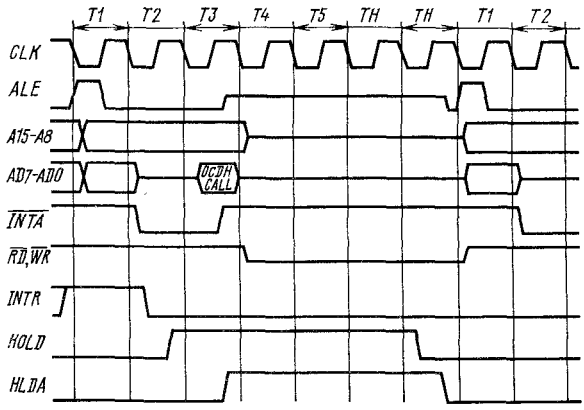


Рис. 2.24. Временны́е диаграммы цикла подтверждения прерывания и захвата шины

пока он не будет обслужен или не сброшен командой SIM. Сброс триггера осуществляется при установленном четвертом разряде аккумулятора R7.5. Сброс запроса реализуется также при перезапуске МП.

Третью группу образует вход TRAP, который не является чисто статическим или динамическим. Так, если для подтверждения прерывания на нем должно устанавливаться напряжение высокого уровня, то для фиксации нового запроса он должен генерировать напряжение низкого уровня, а затем вновь вернуться в исходное состояние (вход комбинированного типа). Это позволяет избежать ложных запусков из-за помех на линии по высокоприоритетному немаскируемому входу TRAP, используемому для фиксации важнейших для МС событий. Времена фиксации запроса TRAP эквивалентны временам предустановки и удержания сигналов по статическим входам.

Сигнал запроса на захват шины проверяется в начале каждого такта T3. Времена предустановки и удержания сигнала принимают значения $t_{HDS} \geqslant 170$ нс, $t_{HDH} \geqslant 0$ нс соответственно. При фиксации запроса шина освобождается в такте, непосредственно следующем за T3. Для этого за 110 нс до окончания T3 устанавливается сигнал подтверждения захвата $\overline{\text{HLDA}}$, а затем линии A15—A8; AD7—AD0, $\overline{\text{RD}}$, $\overline{\text{WR}}$, ALE, IO/$\overline{\text{M}}$ переводятся в третье состояние, тем самым освобождая магистраль для управления со стороны внешних модулей. При необходимости МП завершает такты T4, T5 и переходит в состояние HOLD, которое длится до снятия сигнала запроса. Временны́е диаграммы цикла захвата шины и подтверждения прерываний приведены на рис. 2.24.

Центральный процессор на базе BM85A. В отличие от BM80 микропроцессор BM85A представляет собой практически законченный однокристальный ЦП. Для его запуска необходим только кварцевый резонатор, подключенный к входам X1, X2, и схема

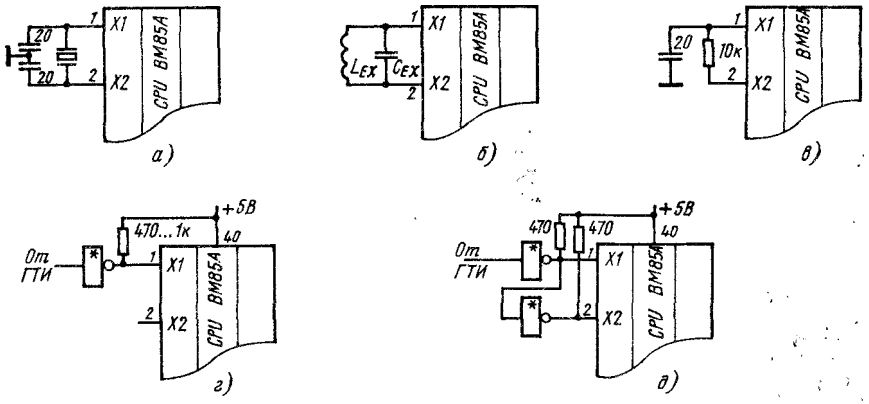


Рис. 2.25. Схемы тактирования микропроцессора ВМ85А:
а — внутренний генератор с кварцевым резонатором; б — внутренний генератор с настроенным LC-контуром; в — внутренний генератор с RC-цепочкой; г — внешний генератор 1—6 МГц; д — внешний генератор 6—10 МГц.

сброса на входе RESIN. Схема подключения кварцевого резонатора приведена на рис. 2.25, а. Конденсаторы емкостью 20 пФ на входах X1 и X2 могут потребоваться при запуске кварцевого резонатора с частотой 4 МГц и выше. Возможны другие варианты синхронизации МП. Схема на рис. 2.25, б обеспечивает частоту колебаний

$$f = 1/2\pi\sqrt{L_{EX}(C_{EX}+C_{IN})},$$

где C_{IN} — входная емкость между X1 и X2.

Если установка точной тактовой частоты не обязательна, то применяется схема, показанная на рис. 2.25, в, которая обеспечивает частоту колебаний около 3 МГц. В схеме на рис. 2.25, г,д синхронизация МП осуществляется от внешнего генератора.

В качестве схемы сброса рекомендуется цепочка, эквивалентная примененной в схеме на рис. 2.17. Вход READY может быть использован для организации асинхронного доступа к системной магистрали. Как и в ЦП на базе ВМ80, в ЦП на базе ВМ85А возможны два варианта построения системной линии подтверждения обмена, отличающиеся друг от друга уровнем активности. Выходы CLK и RESET применяются в качестве системных линий CCLK и INIT передачи тактовых импульсов и сигнала начальной установки соответственно.

Возможности выходов схемы ВМ85А по току составляют $I_{OL}=2$ мА $I_{OH}=400$ мкА. Зная требования по постоянному току, предъявляемые к обычным логическим элементам по входу:

Технология	I_{IH}, мкА	I_{IL}, мА
ТТЛ	40	1,6
ТТЛШ	40	2
n-МОП	10	0,01
ТТЛШ маломощная	20	0,4

можно оценить нагрузочную способность МП ВМ85А.

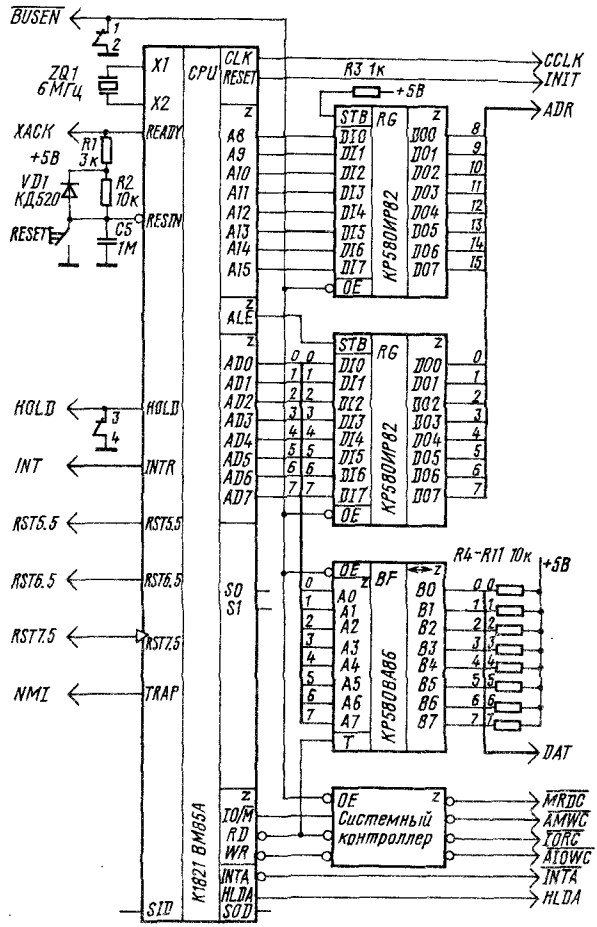


Рис. 2.26. Схема центрального процессора на базе микропроцессора ВМ85А

Для увеличения числа нагрузок по постоянному току сигнал должен быть буферизирован.

Другой фактор, который необходимо учитывать — это емкостная нагрузка C_L. Временные соотношения для ВМ85А гарантируются при $C_L \leqslant 150$ пФ. При превышении этой нагрузки временные соотношения ухудшаются. Отклонение емкостной нагрузки от 150 пФ в сторону уменьшения или увеличения влечет за собой изменение временных соотношений приблизительно на 0,2 нс/пФ.

Построенный таким образом ЦП оказывается ориентированным на двухшинную магистраль, шина управления которой состоит из следующих основных сигналов: ALE, $\overline{RD}$, $\overline{WR}$, IO/$\overline{M}$. Попытка реализовать трехшинную магистраль, эквивалентную магистрали на рис. 2.17, привела к схеме, представленной на

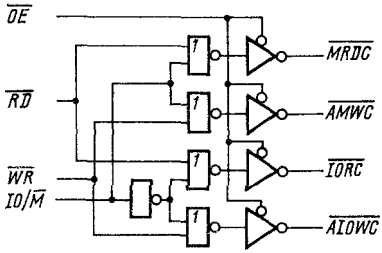


Рис. 2.27. Схема системного контроллера для микропроцессора ВМ85А

рис. 2.26. Буферные регистры КР580ИР82 и шинный формирователь КР580ВА86 служат не только для раздела совмещенной шины *AD*, но и для повышения нагрузочной способности системной магистрали до $I_{OL}=32$ мА и $C_L=300$ пФ. Возможно использование других схем подобного типа.

Системный контроллер представляет собой комбинационную схему (рис. 2.27). На выходах схемы предусмотрены трехстабильные буферы, управляемые сигналом $\overline{OE}$. Этот вход предназначен для перевода командных линий в высокоомное состояние при захвате магистрали внешним модулем, когда BUSEN=0. Логика данного типа может быть реализована как на микросхемах малой степени интеграции, так и на программируемых логических матрицах (ПЛМ) [50].

Сравнение схем на рис. 2.17 и 2.26 показывает, что применение МП ВМ85А выгодно за счет увеличения числа входных линий для приема запросов на прерывание, использования канала последовательного ВВ и перехода к единственному источнику питания +5 В. Не следует также забывать об улучшении скоростных свойств системного канала ЦП на базе МП ВМ85А.

Эффективность использования МП может быть повышена, если от трехшинной структуры вернуться к совмещенной магистрали AD. Однако в этом случае потребуется ряд специальных БИС памяти и ВВ, которые должны быть ориентированы на магистраль данного типа. Примером БИС такого типа служит семейство расширителей серии iMCS-85 фирмы Intel [56].

ГЛАВА 3.

ПОДСИСТЕМА ВВОДА-ВЫВОДА

3.1. Организация программно-управляемого обмена

Пространство ввода-вывода. Подсистема ВВ — один из трех компонентов МС, ответственный за связь с периферийными устройствами (ПУ), а в некоторых случаях и за организацию самих ПУ [3].

С точки зрения программиста, работающего на командном уровне, систему ВВ можно представить в виде пространства ВВ

IOSEG и ряда команд ВВ, имеющих к нему доступ [11]. Пространство IOSEG обычно организовано в виде набора n-разрядных ячеек — портов, каждый из которых может быть адресован независимо от других. Типичная разрядность порта для большинства МС равна восьми. В некоторых системах допускается объединение двух и более элементарных портов в многобайтовые, доступ к которым реализуется как к единому целому. Так, в МС на базе МП К1810ВМ86 допускаются и 8-, и 16-разрядные порты.

Подобно пространству памяти пространство ВВ, как правило, линейно упорядочено. Каждому элементарному порту поставлен в соответствие его адрес. Множество адресов занимает целочисленный диапазон от 0 до 2^p-1. Правила организации и адресации многобайтовых портов могут соответствовать аналогичным правилам размещения и адресации многобайтовых объектов в памяти МС.

В качестве примера приведем систему ВВ для МС на базе МП ВМ80. Пространство IOSEG такой системы организовано в виде 256 8-разрядных портов ВВ с линейно упорядоченными адресами от 0 до 2^8-1. В распоряжение пользователя предоставляются две команды ВВ, содержащиеся в системе команд:

```
IN   port          ;A←port
OUT  port          ;port←A
```

Архитектура ВМ85А позволяет расширить данное пространство ВВ. К нему добавляется изолированный одноразрядный порт ввода SID и изолированный одноразрядный порт вывода SOD, доступ к которым осуществляется с помощью команд

```
RIM                ;A7←SID
SIM                ;SOD←A7
```

Здесь А7 — старший разряд аккумулятора А (см. § 2.7). Далее рассматриваются другие организации подсистемы ВВ.

Программно-управляемый обмен. Между МС и ПУ происходит обмен информацией двух типов: служебной и собственно данных. Служебная информация от МС инициирует действия, связанные с обменом данных, и передается с помощью управляющих слов CW (Control Word). Служебные сообщения от ПУ, информирующие систему о его текущем состоянии, называются словами состояния SW (Status Word). В отличие от них данные передаются с помощью слов данных DW (Data Word) [3, 10].

Объем служебной информации, которой обмениваются ПУ и МС, а также ее интерпретация зависят от типа ПУ. Для наиболее простых устройств, таких как прямо управляемые клавишные матрицы или светодиодные линейки, служебная информация не нужна. В других случаях, например при взаимодействии с накопителем на гибких магнитных дисках, управляющая информа-

ция и данные о состоянии ПУ могут иметь значительный объем. В любом случае для связи с ПУ отводится ряд портов ВВ, через которые и проходит вся информация: сигналы управления, слова состояния или непосредственно данные. С точки зрения программиста множество портов ВВ образует пространство доступа к ПУ.

Отметим, что размер пространства доступа в общем случае не зависит от объема информации, которой обмениваются ПУ и МС. Распространена практика последовательной передачи массива информации через один и тот же порт. Это связано не только с экономией пространства IOSEG, но и с минимизацией ширины физического интерфейса ПУ, а также с его стандартизацией. Вместе с тем существует некоторая договоренность или правила обмена информацией между конкретным ПУ и МС, называемые протоколом обмена. Совокупность этих правил — основа для составления драйвера ПУ, представляющего набор подпрограмм, организующих взаимодействие с ним.

В тех случаях, когда процедуры обмена информацией с ПУ инициируются и выполняются непосредственно программой, реализуемой ЦП, говорят о программно-управляемом обмене. Программно-управляемый обмен не является единственным типом обмена. Но судя по аппаратным затратам, это наиболее эффективный тип обмена, поэтому он находит самое широкое применение в разнообразных МС.

Прямой ввод-вывод. В наиболее простом виде процедура ввода или вывода выполняется независимо от состояния ПУ. Такой вид обмена назван прямым или безусловным. Процедуры прямого ВВ в чистом виде возможны только при управлении всегда готовыми к обмену простейшими ПУ. К тому же они являются составными элементами более сложных процедур программно-управляемого обмена, к числу которых относится условный ВВ.

Простейшая процедура ввода информации из подсистемы ВВ связана с чтением отдельного порта пространства IOSEG. С этой целью физические линии подсистемы ВВ упаковываются в байтовые и многобайтовые наборы, состояние каждого из которых может быть передано на шину данных МС во время цикла чтения порта ВВ. Физическое управление вводом состояния конкретного набора лучше всего осуществлять с помощью шинного формирователя с тремя состояниями (рис. 3.1), открываемого стробом IORC при условии его выбора. Выбор или адресация формирователей возлагается на специальную декодирующую схему, называемую логикой выборки кристаллов, которая, в свою очередь, управляется сигналами шины адреса и при необходимости шины управления, ответственными за кодирование размера передаваемых данных: байта, слова, двойного слова.

Запись данных в порт соответствует их выводу в подсистему ВВ. В простейшем варианте порта вывода (рис. 3.2) данные фиксируются в регистре по стробу IOWC или AIOWC при условии

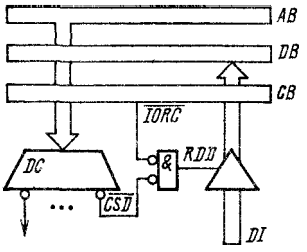


Рис. 3.1. Схема порта ввода

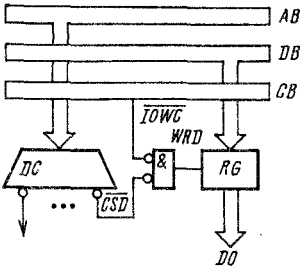


Рис. 3.2. Схема порта вывода

его выбора. Роль адресации регистра выполняет логика выборки кристаллов. При использовании строба $\overline{\text{IOWC}}$ данные могут фиксироваться его низким уровнем напряжения, при стробе $\overline{\text{AIOWC}}$ они должны восприниматься только по его фронту.

Примерами процедур прямого ввода служат следующие подпрограммы:

```
IXHL:   PUSH    PSW
        IN      CSD     ;Ввод данных из порта CSD
        MOV     M,A     ;Запись данных по адресу HL
        POP     PSW
        RET
 IAC:   MOV     A,C     ;Пересылка адреса порта
        STA     IAC+5   ;Модификация команды IN 0
        IN      0       ;Ввод данных
        RET
```

Процедуры для прямого вывода аналогичны:

```
OXHL:   PUSH    PSW
        MOV     A,M     ;Чтение данных по адресу HL
        OUT     CSD     ;Вывод данных
        POP     PSW
        RET
 OAC:   PUSH    PSW
        MOV     A,C     ;Пересылка адреса порта
        STA     OAC+7   ;Модификация команды OUT 0
        POP     PSW     ;Восстановление данных
        OUT     0       ;Вывод данных
        RET
```

Быстродействия шинных формирователей и регистровых схем обычно достаточно, чтобы поддержать обмен с максимальной для ЦП скоростью (синхронный режим работы системной магистрали). Поэтому подтверждение обмена сводится к как можно более быстрому возврату сигнала подтверждения XACK, образуемому стробами $\overline{\text{IORC}}$, $\overline{\text{IOWC}}$ или $\overline{\text{AIOWC}}$ по схеме

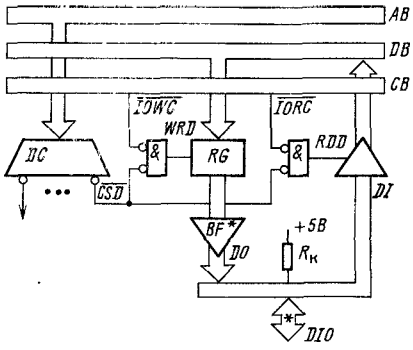


Рис. 3.3. Схема псевдодвунаправленного порта

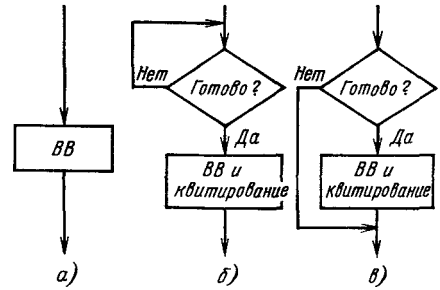


Рис. 3.4. Процедуры программно-управляемого обмена:
а — прямого; б — условного с занятием цикла; в — условного совмещенного

$$XACK = IORC \cdot CSD \quad \vee \quad IOWC \cdot CSD \vee \ldots$$

В системах с инверсным сигналом $\overline{XACK}$ частные сигналы подтверждения обмена формируются буферами с открытым коллектором и объединяются по принципу «монтажное ИЛИ». В малых МС с активной по умолчанию линией подтверждения XACK не требуется логика подтверждения (см. рис. 3.1, 3.2). Это может быть использовано для упрощения подсистемы ВВ.

Отметим, что в отличие от памяти данные, выводимые и вводимые через порты с одним и тем же адресом, в общем случае никак не связаны между собой. Вместе с тем прямое соединение выходных DO и входных DI линий портов ВВ дает возможность чтения выведенных ранее данных, что может быть полезным в ряде практических случаев, например при тестировании порта вывода или восстановлении его текущего состояния. Однако ввод внешних данных через такой порт невозможен.

Для поддержки двустороннего ВВ очень широко применяется псевдодвунаправленный порт (рис. 3.3). В данной схеме к выходам DO подключены буферы с открытым коллектором, допускающие объединение с линиями ввода внешних данных по схеме «монтажное ИЛИ». При этом нагрузочные резисторы могут быть встроены в порт. Установка на шине порта вывода напряжения высокого уровня обеспечивает ввод внешних данных без каких-либо искажений. При необходимости некоторые линии ввода могут быть выборочно замаскированы установлением на соответствующих им линиях DO напряжения низкого уровня. Допускается в качестве входных линий использовать лишь часть порта, тогда как оставшаяся часть может выполнять роль выходной шины. В целом схема, приведенная на рис. 3.3, достаточно универсальна, что объясняет ее широкое распространение, например, в однокристальных микроЭВМ.

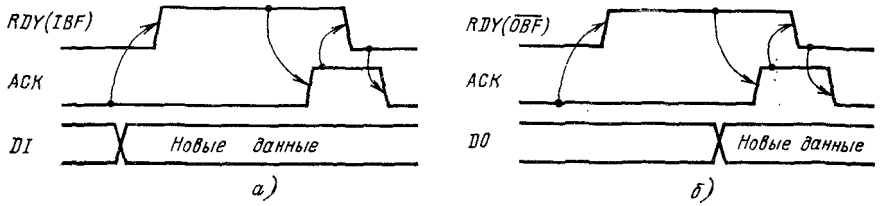


Рис. 3.5. Временные диаграммы условного обмена:
а — ввод; б — вывод

Условный ввод-вывод. Способность ПУ к скоростному ВВ данных не безгранична и, как правило, во много раз ниже скорости работы ЦП. Поэтому прежде чем приступить к чтению новых данных из порта ввода, необходимо удостовериться, что ПУ готово предоставить или уже предоставило эти данные. Иначе операция сведется к вводу недействительных или старых данных. Аналогичная ситуация складывается и при выводе данных, когда требуется проверка готовности ПУ к приему новых данных. В противном случае неразрешенный вывод со стороны ЦП может привести к потере нового или предыдущего элемента данных.

Типичное решение проблемы синхронизации обмена состоит в сопровождении операции условного ВВ специальным сигналом готовности RDY (Ready), генерируемым ПУ. Сигнал вводится в МС в составе слова состояния SW и служит для информирования ЦП о готовности ПУ принять или передать новые данные.

Наряду с прямым (рис. 3.4,а) существуют два типа условного ВВ: с занятием цикла (рис. 3.4,б) и совмещенный (рис. 3.4,в). В первом случае МС зависает на цикле ожидания готовности, тратя на это все машинное время. Во втором случае, если ПУ не готово к обмену, ЦП возвращается к основной задаче без выполнения операции ВВ. Однако он может снова проверить готовность ПУ к обмену и при удачном исходе выполнить ее.

После завершения операции ВВ сигнал готовности RDY должен быть снят и выставлен заново только при новой готовности к обмену. С этой целью ПУ следует проинформировать об окончании операции, для чего используется включенный в одно из управляющих слов CW сигнал подтверждения ACK (Acknowledgement). Протокол обмена служебной информацией такого типа (рис. 3.5) называется квитированием. Он обеспечивает надежную асинхронную передачу данных со скоростями, определяемыми ПУ.

Приведем пример процедуры условного ВВ с занятием цикла и программно-управляемым квитированием (см. рис. 3.4,б):

```
        PUSH    PSW
WAIT1:  IN      CSSC    ;Ввод слова состояния
        ANI     RDY     ;Выделение сигнала RDY
```

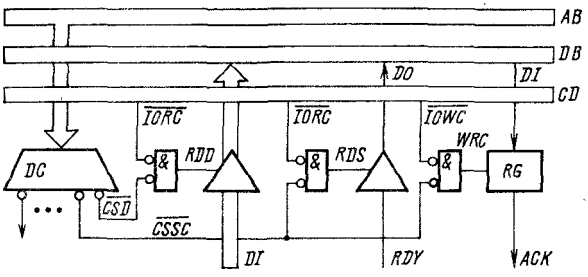


Рис. 3.6. Схема порта ввода с программным квитированием

```
        JZ      WAIT1       ;Если RDY=0, то ждать
        POP     PSW
;Процедура прямого ВВ
        PUSH    PSW
        MVI     A, ACK      ;Вывод сигнала
        OUT     CSSC        ;подтверждения ACK
WAIT2:  IN      CSSC        ;Выделение сигнала RDY
        JNZ     WAIT2       ;Если RDY=1, то ждать
        MVI     A, NACK     ;Сброс сигнала
        OUT     CSSC        ;подтверждения ACK
        POP     PSW
        RET
```

В сравнении с прямым условный ВВ с программным квитированием связан с увеличением аппаратных затрат (рис. 3.6). Однако это наиболее типичный вид обмена с ПУ. Он распространен в системах, где эффективность не связана с ожиданиями. При этом вариант совмещенного условного обмена обеспечивает оперативное отслеживание внешних событий и своевременную реакцию на их возникновение.

Признаком окончания операции может служить само обращение к порту данных. Это упрощает как схему порта, так и процедуру обмена, освобождая пользователя от работы с управляющим словом.

Очень часто на входе порта ввода (рис. 3.7, а) предусматривается регистр-защелка, фиксирующий входные данные по стробу STB (Strobe), генерируемому ПУ. Устанавливает флажок готовности IBF (Input Buffer Full), инициируя операцию ввода (рис. 3.7, б). Флажок сбрасывается автоматически при чтении содержимого порта ввода. Необходимо обратить внимание на логику формирования флажка готовности, который переключается только после завершения операции зашелкивания во входной регистр или считывания из него данных. Что произойдет, если это условие не будет выполнено?

Введение логики формирования флажка готовности в состав входного порта освободило ПУ от необходимости прямого управления им, придало процедуре обращения к порту с обеих

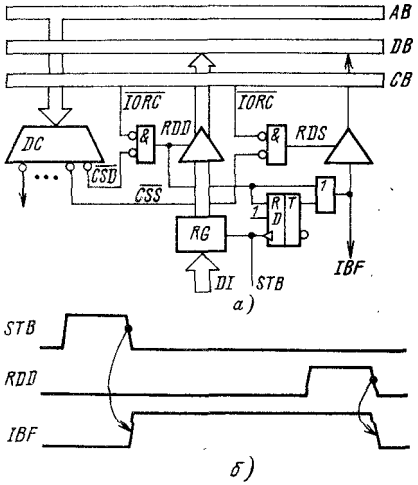


Рис. 3.7. Порт условного ввода:
а – структурная схема; б — временные диаграммы

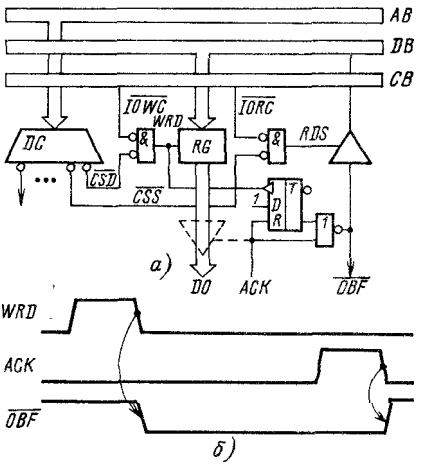


Рис. 3.8. Порт условного вывода:
а — структурная схема; б – временные диаграммы

сторон симметричный вид. Применение этой методики к выходному порту приводит к схеме на рис. 3.8, а, в которой роль флажка готовности выполняет флажок $\overline{\text{OBF}}$ (Output Buffer Full), сигналом окончания операции вывода и установки флажка OBF служит сигнал ACK, генерируемый ПУ. Процедуры условного ВВ упростились. Например, для условного ВВ с занятием цикла имеем

```
IA:     IN      CSS     ;Ввод слова состояния
        ANI     IBF     ;Выделение флажка IBF
        JZ      IA      ;Если IBF=0, то ждать
        IN      CSD     ;Иначе — ввод данных
        RET
OA:     PUSH    PSW
WAIT:   IN      CSS     ;Ввод слова состояния
        ANI     NOBF    ;Выделение флажка OBF
        JZ      WAIT    ;Если OBF=0, то ждать
        POP     PSW
        OUT     CSD     ;Иначе — вывод данных
        RET
```

При необходимости на выходе порта вывода может быть предусмотрен трехстабильный буфер, открываемый сигналом ACK. Это важно, например, при организации двунаправленного порта (рис. 3.9). Состояние входного и выходного регистров порта отображается флажками готовности IBF и OBF соответственно. Одновременная активизация линий ACK и STB реализует выходной порт с обратной связью.

Рассмотренные выше порты являются простейшими схемными элементами, на основе которых реализуется связь с ПУ. Они образуют первый, наиболее близкий к МС уровень аппаратных средств подсистемы ВВ. В ряде случаев этот уровень единственный.

3.2. Периферийные БИС

Адаптеры и контроллеры периферийных устройств. Под ПУ понимается любое устройство, обменивающееся информацией с МС. Роль физической среды обмена выполняют порты ВВ. Примерами ПУ могут служить клавиатура, индикаторы, счетчики/таймеры, преобразователи информации, исполнительные устройства, цифровые датчики и т. д. Некоторые из этих устройств или их определенные части могут встраиваться в подсистему ВВ и размещаться непосредственно за портами.

Периферийные устройства, встроенные в МС, могут либо представлять законченные автономные средства, например системный таймер или клавишный регистр, либо образовывать дополнительные средства связи с устройствами, расположенными вне МС. В качестве примеров внешних ПУ можно привести дисплеи на электронно-лучевых трубках (ЭЛТ), внешние запоминающие устройства, каналы передачи информации, измерительные приборы и т. д. Связь с устройствами подобного рода осуществляется встроенными в МС промежуточными преобразователями, которые совместно с относящимися к ним портами ВВ получили название периферийных адаптеров и контроллеров. В некоторых наиболее сложных случаях им присвоен статус периферийного сопроцессора.

Унификация интерфейсов внешних ПУ или периферийных интерфейсов привела к созданию ряда стандартных адаптеров и контроллеров. В их функции кроме ВВ входит: формирование периферийного интерфейса; физическое управление внешними ПУ в соответствии с командами ЦП; перенос протокола обмена с ЦП на более высокий уровень интеллектуальности; электрическое согласование сигналов.

Передача функции физического управления ПУ его периферийному адаптеру или контроллеру освобождает ЦП от трудоемких обязанностей низкоуровневого управления устройствами. Функции ЦП сводятся к общему управлению контроллеров ПУ с помощью высокоуровневых команд и слов состояния, что упрощает ПО, уменьшает загрузку ЦП и повышает общую производительность МС [7].

Благодаря успехам электронной промышленности во второй половине 70-х гг. удалось разработать множество однокристальных периферийных адаптеров и контроллеров, а также ряд вспомогательных схем к ним, которые называются периферийны-

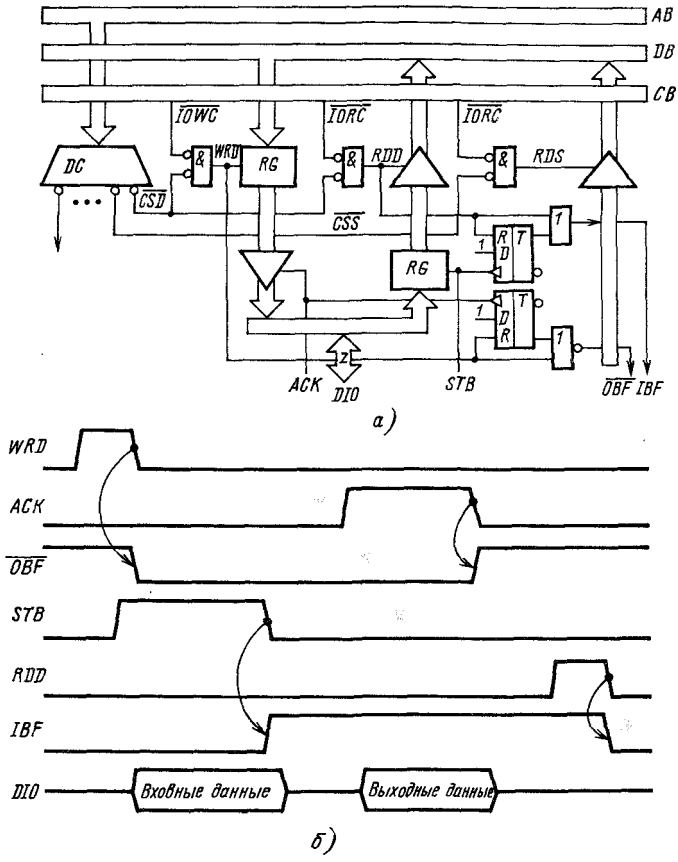


Рис. 3.9. Порт двунаправленного обмена:
а — структурная схема; б — временные диаграммы

ми БИС. Приведем далеко не полный набор таких БИС для МС на базе МП ВМ80/ВМ85А:

К589ИР12	Многорежимный буферный регистр
КР580ВВ51/ВВ51А	Программируемый связной адаптер
КР580ВИ53, 8254	Программируемый интервальный таймер
КР580ВВ55/ВВ55А	Программируемый периферийный адаптер
КР580ВТ57	Контроллер прямого доступа к памяти
КР580ВН59, К1810ВН59А	Программируемый контроллер прерываний
КР580ВГ71/ВГ72, К1818ВГ93	Контроллер накопителя на гибких магнитных дисках
8273	Контроллер линии передачи данных
КР580ВГ75	Контроллер ЭЛТ
КР580ВВ79	Контроллер клавиатуры и ЭЛТ
КР580ВК91А, КР580ВА93	Приемопередатчики шины IEEE-488
8292	Контроллер шины IEEE-488
8041	Универсальный периферийный интерфейс
К1810ВМ89	Сопроцессор ВВ

Этим приборы не только упрощают разработку ПО и повышают производительность МС, но и облегчают проектирование аппаратуры. Заменяя большие объемы аппаратных средств, которые в противном случае пришлось бы разрабатывать на основе малой логики, они повышают компактность и надежность систем. Если первые контроллеры (КР580ВВ51/ВИ53/ВВ55/ВТ57/ВН59) были ориентированы на решение общих прикладных задач, то последующие либо их улучшали (КР580ВВ51А/ВВ55А, К1810ВН59А), либо выполняли более узкие функции (КР580ВГ72/ВГ75/ВВ79/ВК91А и др.).

В любом случае эти устройства решали узкоспециальные задачи, такие как организация параллельного или последовательного интерфейса, распределение интервалов времени или управления ЭЛТ. Однако несмотря на большое число контроллеров и их повышенную гибкость остается еще много нерешенных проблем управления ПУ. Для их решения вводятся универсальные ПУ сопряжения типа 8041 фирмы Intel [59], которые представляют собой однокристальные микроЭВМ с системным интерфейсом подчиненного типа (см. § 4.9). Еще большую общность предоставляют сопроцессоры ВВ [58], например К1810ВМ89.

Стандартная шина для периферийных БИС. Периферийные БИС серии КР580 удовлетворяют электрическим и логическим спецификациям на микропроцессорную шину Microbus фирмы National Semiconductor (США) — унифицированную 8-разрядную шину, объединяющую в функционально законченный модуль отдельные компоненты, размещенные в непосредственной близости друг от друга. В 1978 г. она была признана стандартным интерфейсом периферийных БИС (стандарт P696-3 IEEE), обеспечивающим совместимость уже существующих микропроцессорных кристаллов с вновь разработанными [42].

Основная задача проектирования систем на базе МП и микропроцессорных БИС состоит в выборе внутрисистемной магистрали и сопряжении с ней приборов. Функциональные требования, накладываемые на внутрисистемную магистраль, рассмотрены в § 1.3. Они были сформулированы исходя из возможностей физических интерфейсов МП и ЦП на их основе. Другие требования предъявляются к интерфейсам периферийных БИС, призванным расширить функциональные возможности МС на аппаратном уровне. Попытка унифицировать эти требования привела к разработке ряда стандартных внутренних интерфейсов периферийных БИС, в числе которых шина Microbus. В случае простейших МС, состоящих из нескольких кристаллов, она может интерпретироваться как системная.

Двунаправленная трехстабильная шина данных D7 — D0 (Data Bus) используется для ВВ данных, информации о состоянии и управляющих слов, организованных в 8-разрядные наборы (рис. 3.10). В простейших случаях эти линии могут быть однонаправленными.

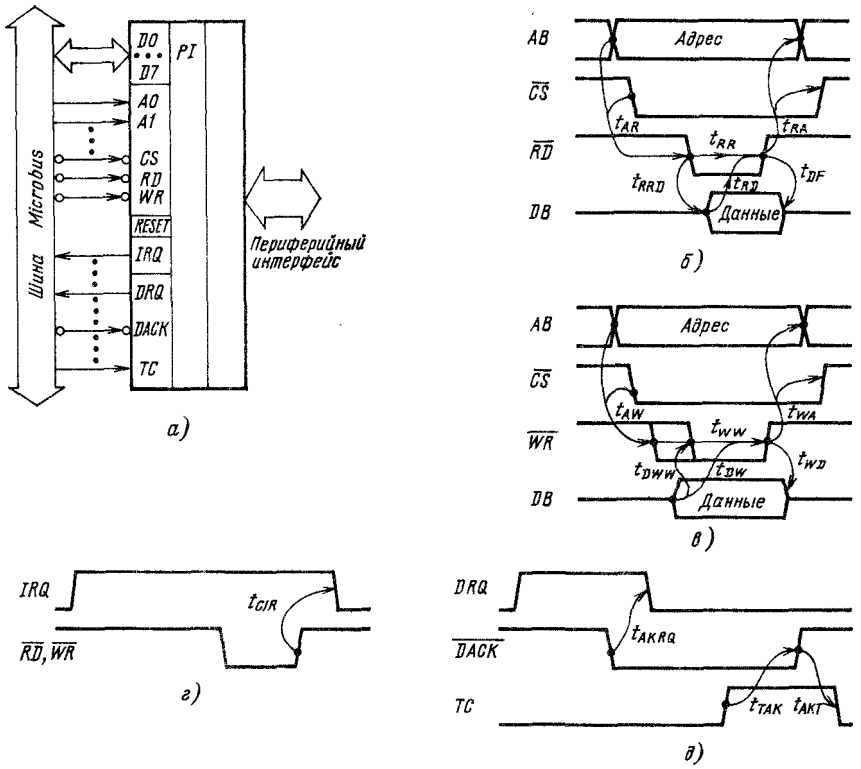


Рис. 3.10. Временны́е диаграммы сигналов шины

а — состав сигналов периферийной БИС; б — цикл чтения; в — цикл записи; г — цикл обработки запроса прерывания; д — цикл обработки запроса ПДП

Наличие напряжения низкого уровня на линии выбора кристалла $\overline{CS}$ (Chip Select) показывает, что производится обращение к данному прибору. Сигнал $\overline{CS}$ должен действовать в сочетании с другими сигналами, такими как строб чтения $\overline{RD}$ или записи $\overline{WR}$. Так, при напряжении низкого уровня на входной линии $\overline{RD}$ (Read) разрешает прибору выдачу информации на шину D7 — D0 при условии, что он выбран ($\overline{CS}=0$). Необходимо, чтобы на срезе строба RD на линиях шины D7 — D0 находились действительные данные. В случае невыполнения условия $\overline{CS}=0$ передатчики шины данных прибора должны находиться в высокоомном состоянии. Аналогично низкий уровень напряжения на входном контакте $\overline{WR}$ (Write) в совокупности с низким уровнем на входе $\overline{CS}$ дает возможность БИС принять данные из шины D7 — D0. Для приборов первого поколения информация на шине данных

D7—D0 должна быть истинной во время действия сигнала записи WR. БИС второго поколения требуют ее истинности только на срезе сигнала записи.

Во многих МС предполагается использование раздельных стробов чтения-записи при обращении к памяти — MEMR, MEMW и к портам ВВ — I/OR, I/OW (изолированный ВВ). В тех случаях, когда БИС содержит и память и порты ВВ, должны быть предусмотрены обе пары стробов чтения-записи.

Сигналы адресной шины A15—A0 применяются для адресации внутренних регистров БИС сопряжения. Сигналы адреса, поступающие на интерфейс, выбирают один из его внутренних регистров или ячеек памяти и в сочетании с входными сигналами CS, WR или RD определяют тип выполняемого обмена: ВВ данных, ввод управляющей информации или вывод состояния БИС.

Выходной сигнал готовности к обмену READY предназначен для реализации асинхронного обмена по шине Microbus. Стробы RD и WR будут сняты только после появления активного уровня на линии READY, что гарантирует успешный обмен данными с медленно действующими интерфейсными БИС. В приборах с быстрым доступом ответный сигнал READY может отсутствовать.

Сигнал сброса RESET (допускается вариант прямого сигнала RESET) выдается МП при включении источника питания или нажатии кнопки сброса. При получении данного сигнала интерфейсная БИС устанавливается в исходное состояние, при этом шина данных БИС должна быть отключена от системной шины Microbus.

Перекрестные устройства часто выдают в систему запросы на прерывание IRQ0, IRQ1, ..., которые служат признаками программно не предсказуемых событий. Сигнал прерывания IRQ, после того как он выработан асинхронно по отношению к другим сигналам шины, должен оставаться в активном состоянии до тех пор, пока не будет обработан. Он снимается только по срезу соответствующего строба RD или WR.

Для МС с ПДП в шине Microbus предусмотрены один обычный DRQ (Direct Request) и два специальных сигнала: DACK (Direct Acknowledge) и TC (Terminal Count). Каждый ПДП-канал формирует свой собственный сигнал запроса DRQ0, DRQ1, ..., на который МС реагирует выдачей специального сигнала подтверждения (квитанции) DACK. Этот сигнал должен быть эквивалентным сигналу выборки кристалла и иметь такие же временные характеристики. При этом в канале ПДП одновременно выполня-

ются операции чтения и записи. Еще один специальный сигнал TC выдается контроллером ПДП-каналу как указание о том, что настоящий цикл обмена является последним циклом для текущего блока данных.

Микросистемы, построенные с расчетом на универсальную 8-разрядную шину Microbus, обеспечиваются средствами прямого сопряжения с большинством выпускаемых сегодня периферийных БИС, ориентированных на стандарт шины Microbus. Благодаря этому возможна дальнейшая модификация МС, что очень важно для сложных МПК.

Вариантами, альтернативными интерфейсу Microbus, являются двухшинные интерфейсы внутриплатного уровня. Сюда прежде всего следует отнести двухшинную магистраль ВМ85А, обеспечивающую прямое подключение периферийных БИС и МП ВМ85А. Вторым широко распространенным в отечественной практике стандартом на внутрисистемную магистраль периферийных БИС является МПИ-шина (аналог Q-bus). Стандарт предназначен для МПК БИС серий К1801, К1809, К581, К588 и др.

Приведенная на рис. 2.17 схема 5-кристального ЦП не обеспечивает МС полным набором сигналов шины Microbus. Отсутствуют такие важные для системы линии, как $\overline{CS0}$, $\overline{CS1}$,... — линии выбора кристалла, IRQ0, IRQ1,... — линии запросов на прерывание, а также набор сигналов ПДП.

Сигналы $\overline{CS}$ обычно формируются из комбинации старших разрядов адресной шины ADR15 — ADR0 логикой выбора кристалла, которая представляет собой комбинационный дешифратор и легко реализуется с помощью ПЗУ, ПЛМ [13, 50] или логических схем малой степени интеграции. Для получения оставшихся сигналов шины Microbus необходимо в МС ввести контроллер обработки прерываний и контроллер ПДП (рис. 3.11), реализуемые с помощью специальных БИС типов КР580ВН59 и КР580ВТ57 соответственно [1].

3.3. Средства параллельного ввода-вывода

Многорежимный буферный регистр К589ИР12. Параллельный ВВ — один из наиболее простых и широко распространенных способов обмена информацией с ПУ, при котором практически не требуется никакого промежуточного преобразования данных. Поэтому параллельный обмен информацией производится непосредственно портами ВВ.

Для построения портов могут быть использованы различные БИС буферных регистров и шинных формирователей. При этом принципиальные схемы практически копируют вышеприведенные схемы соответствующих портов. Из числа специально разработанных для этой цели периферийных БИС следует отметить многорежимный буферный регистр (МБР) типа К589ИР12 (ИР12) [16].

Логическая схема МБР (рис. 3.12) явилась результатом анализа и обобщения схем простейших портов ВВ. В ее состав кроме 8-разрядного регистра-защелки с

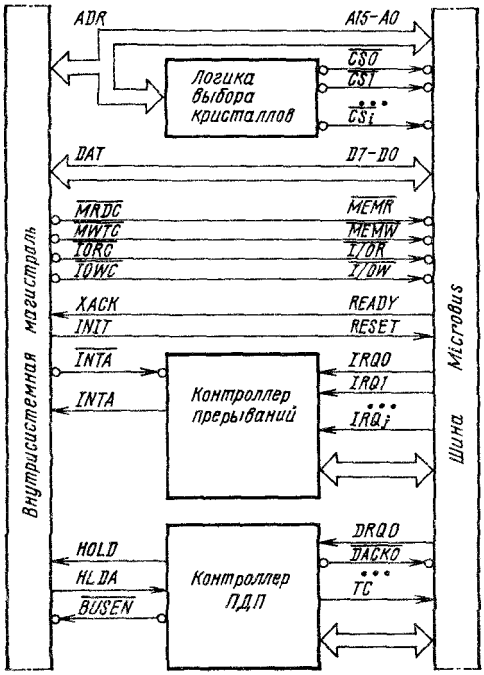


Рис. 3.11. Схема контроллера шины Microbus

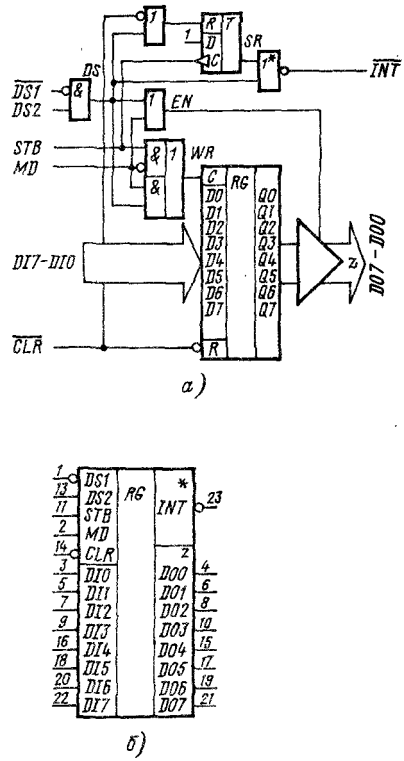


Рис. 3.12. Многорежимный буферный регистр ИР12:

а — структурная схема; б — условное графическое обозначение

трехстабильным буфером на выходе входит логика флажка состояния и средства начального сброса. Благодаря этому БИС К589ИР12 является удобным средством построения портов условного обмена. Схема выполнена по ТТЛШ-технологии и имеет 24-выводной корпус типа 239.24. Ток потребления $I_{CC} = 130$ мА.

Регистр-защелка принимает данные по входам D17 — D10 (Data Input), если на его *синхровход подано напряжение высокого уровня*

$$WR = DS \cdot MD \vee STB \cdot \overline{MD}$$

При возврате к напряжению низкого уровня происходит фиксация данных. Выходной буфер разрешает передачу состояния регистра без инверсии при

$$EN = MD \vee DS$$

В противном случае выходные линии D07 — D00 (Data Output) имеют высокое выходное сопротивление. Нагрузочная способность выходной шины следующая: $I_L = 100$ мА, $C_L = 300$ пФ. Задержка от входа DI до выхода DO составляет 30 нс.

Управляющий вход MD (Mode) определяет режим работы МБР. При MD = 1 (режим выходного порта) работа выходных буферов разрешена, а WR = DS. Схема

реализует выходной порт МС. При MD=0 (режим входного порта) состояние выходного буфера определяется сигналом DS (Device Select), а WR=STB. Схема реализует входной порт МС. При этом входы DS1, DS2 используются для выбора устройства со стороны ЦП, а вход STB — для фиксации данных в режиме входного порта и подтверждения считывания в режиме выходного порта со стороны ПУ.

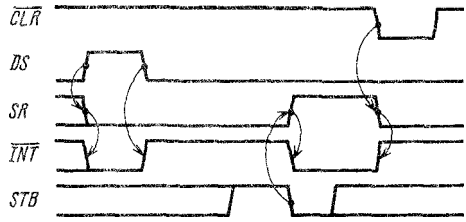


Рис. 3.13. Временные диаграммы работы многорежимного буферного регистра ИР12

В состав микросхемы включен триггер запроса обслуживания SR (Service Request), отражающий состояние МБР. Триггер устанавливается в 1 по срезу сигнала STB, а в исходное состояние при высоком уровне внутреннего сигнала DS (рис. 3.13). Дополнительный элемент ИЛИ на выходе триггера обеспечивает переключение выхода $\overline{INT}$ только по срезам сигналов STB и DS, т. е. после завершения операций записи или считывания данных, что соответствует логике установки флажка готовности. Выход запроса $\overline{INT}$ выполнен по схеме с открытым коллектором. Внутренний регистр и триггер запроса могут быть установлены в исходное состояние при подаче напряжения низкого уровня на вход $\overline{CLR}$ (Clear).

Программируемый периферийный адаптер ВВ55/ВВ55А. Использование БИС с организацией ИР12 позволяет реализовывать различные параллельные интерфейсы ВВ. Проектирование интерфейса осуществляется аппаратными средствами за счет предварительной коммутации управляющих входов БИС. Оперативное изменение такого интерфейса практически невозможно, а его функции ограничены. Это обстоятельство сужает область применения МБР, она ограничивается реализацией простейших параллельных интерфейсов с постоянной по времени логикой обмена.

В усложненном интерфейсе, когда логика обмена заранее не известна или характеристики процедур обмена во время работы МС должны меняться, используются программируемые периферийные адаптеры (ППА). В их состав входят программно-доступные регистры управляющих слов CW, которые и определяют режим работы адаптера. Программный доступ к управляющему регистру со стороны ЦП дает возможность оперативно управлять работой ППА и изменять характеристики интерфейса.

К числу ППА следует отнести следующие БИС:

КР580ВВ55, 8255	Программируемый периферийный адаптер фирмы Intel (базовый вариант)
КР580ВВ55А, 8255А	Программируемый периферийный адаптер фирмы Intel (улучшенный вариант)
MC6820	Программируемый периферийный адаптер фирмы Motorola (базовый вариант)
MC6821	Программируемый периферийный адаптер фирмы Motorola (улучшенный вариант)

Программируемый периферийный адаптер КР580ВВ55 (ВВ55) — это однокристальное программируемое устройство параллельного ВВ информации произвольного формата. В состав осуществляемых им процедур входит параллельный обмен данными с квитированием или без него как в режиме программного управления,

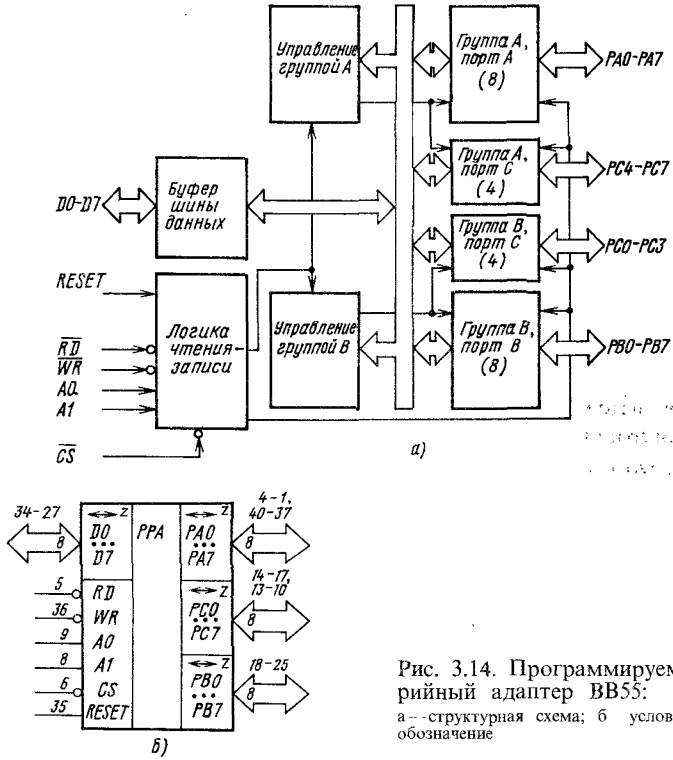


Рис. 3.14. Программируемый периферийный адаптер ВВ55:
а — структурная схема; б — условное графическое обозначение

так и по прерываниям. При этом обеспечивается организация не только однонаправленного, но и двунаправленного ВВ. Определение и переопределение типа интерфейса выполняется программными методами с помощью специальных процедур инициализации.

В состав ППА (рис. 3.14) входят три двунаправленных 8-разрядных порта, разбитых на две группы, два устройства управления группами портов и интерфейсная логика для согласования с системной магистралью. Организации портов, содержащих буферные регистры и шинные формирователи с тремя состояниями (рис. 3.15), значительно отличаются друг от друга. Схема управления содержит регистр управляющего слова CW, который доступен только для записи, чтение CW не допускается.

Обмен информацией между МП и внутренними регистрами ППА осуществляется через двунаправленный шинный формирователь и управляется сигналами $\overline{CS}$, A0, A1, $\overline{RD}$ и $\overline{WR}$ в соответствии с требованиями к шине Microbus. Адресные сигналы выбирают один из внутренних регистров, а стробы $\overline{RD}$ и $\overline{WR}$ управляют направлением передачи согласно табл. 3.1. Сигнал $\overline{CS}$ необходим для выбора кристалла.

Вход RESET служит для аппаратного сброса БИС в исходное состояние. Все внутренние регистры ППА, включая регистр управляющего слова CW, устанавливаются в 0. Сброс CW соответствует переводу всех портов в режим прямого ввода без квитирования.

Таблица 3.1

A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Операция
0	0	0	1	0	D←Порт A
0	1	0	1	0	D←Порт B
1	0	0	1	0	D←Порт C
1	1	0	1	0	Недопустимо
0	0	1	0	0	Порт A←D
0	1	1	0	0	Порт B←D
1	0	1	0	0	Порт C←D
1	1	1	0	0	Управление←D
X	X	1	1	0	Нет операции
X	X	X	X	1	Нет операции

Дальнейшая настройка ППА выполняется программно с помощью специального управляющего слова MS (Mode Selection), которое назначает режим работы каждому каналу. Эти режимы могут быть изменены в любое время. Для хранения MS используется регистр CW.

Регистр управляющего слова 7-разрядный. Запись в него осуществляется только при передаче в ППА управляющего слова с D7=1 (признак слова MS), которое определяет режим работы каждого канала ВВ в соответствии с форматом, приведенным на рис. 3.16,а. Каждое из управляющих устройств группы А или В принимает свою часть слова выбора режима. При записи нового управляющего слова все буферные регистры портов устанавливаются в 0.

Адаптер поддерживает три режима работы портов:

режим 0 — однонаправленный ВВ без квитирования (применим к любому из трех портов);

режим 1 — однонаправленный ВВ с квитированием (применим к портам А и В);

режим 2 — двунаправленный ВВ (допускается только для порта А).

При работе портов А и В в режимах 1 и 2 часть линий порта С из соответствующей группы используется для управления обменом с внешними ПУ. Функциональные отличия портов предопределены их структурной организацией.

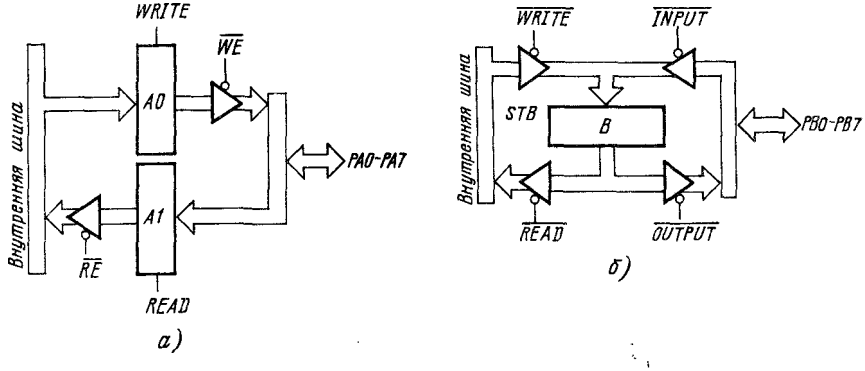


Рис. 3.15. Организация портов А (а) и В (б) программируемого периферийного адаптера

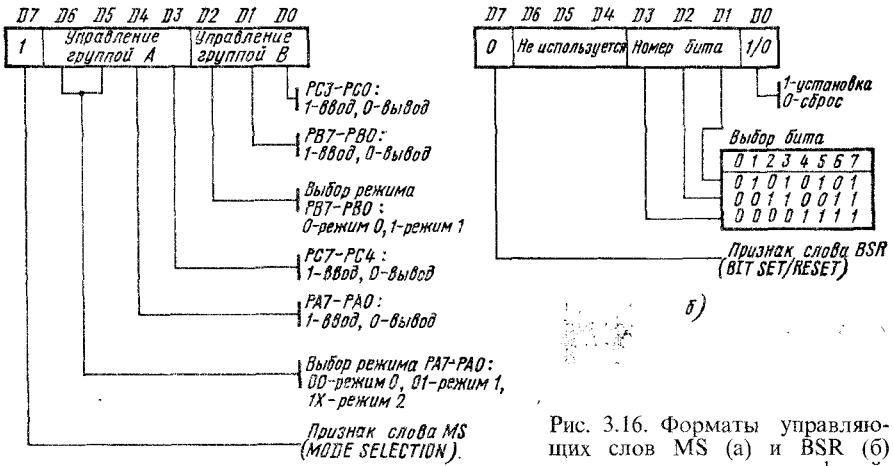


Рис. 3.16. Форматы управляющих слов MS (а) и BSR (б) программируемого периферийного адаптера

При D7=0 управляющее слово применяется для независимой установки (сброса) разрядов выходного порта С и носит название BSR (Bit Set/Reset). Выборочная манипуляция содержимым выходного буферного регистра порта С важна при его использовании в качестве шины управления ПУ. Формат слова BSR представлен на рис. 3.16,б.

В режиме 0 осуществляется прямой однонаправленный ВВ через любой из трех портов без каких-либо сигналов сопровождения. Данные вводятся или выводятся через выбранный канал в соответствии с временными диаграммами на рис. 3.17. В данном режиме интерфейс можно представить как набор параллельных линий ВВ, организованных в две байтовые и две 4-разрядные шины, причем каждая может быть применена либо для ввода, либо для вывода независимо от других (рис. 3.18). Входная информация адаптером не запоминается и читается при низком уровне напряжения сигнала на входе $\overline{RD}$. Выходная информация защелкивается в выходной буферный регистр выбранного порта по срезу системного сигнала $\overline{WR}$ и остается на выходе порта до нового цикла вывода или изменения режима.

Режим 1 обеспечивает организацию однонаправленного ВВ с квитированием через порты А и В. Входные и выходные данные фиксируются во внутренних регистрах ППА. Управление вводом (рис. 3.19, а) осуществляется сигналами

$\overline{STB}$ (Strobe)	Строб записи данных во входной регистр-защелку. Загрузка данных осуществляется по фронту $\overline{STB}$
IBF (Input Buffer Full)	Подтверждение загрузки данных. Сигнал устанавливается по срезу $\overline{STB}$ и сбрасывается по фронту $\overline{RD}$
INT (Interrupt)	Запрос на прерывание. Сигнал устанавливается по фронту $\overline{STB}$ и сбрасывается по срезу $\overline{RD}$. Используется для организации ввода по прерываниям

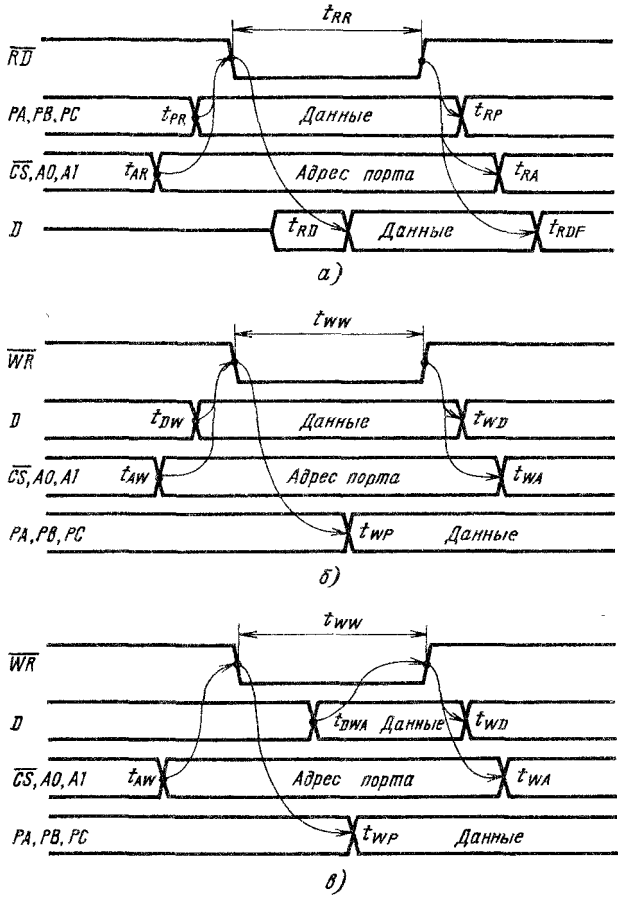


Рис. 3.17. Временные диаграммы режима 0:
а — ввод в режим 0; б — вывод в режим 0 (вариант ВВ55); в — вывод в режиме 0 (вариант ВВ55А)

Другой набор сигналов управления применяется при выводе данных (рис. 3.19, б):

$\overline{\text{OBF}}$ (Output Biffer Full)	Сгроб вывода новых данных. Низкий уровень сигнала $\overline{\text{OBF}}$ информирует о выводе новых данных. Сигнал $\overline{\text{OBF}}$ устанавливается по фронту $\overline{\text{WR}}$ и сбрасывается по срезу $\overline{\text{ACK}}$
$\overline{\text{ACK}}$ (Acknowledge)	Подтверждение приема выходных даниых со стороны внешнего устройства. Низкий уровень напряжения сигнала сообщает ППА, что данные приняты
INT (Interrupt)	Запрос на прерывание. Сигнал устанавливается по фронту $\overline{\text{ACK}}$ и сбрасывается по срезу $\overline{\text{WR}}$. Используется для организации вывода по прерываниям

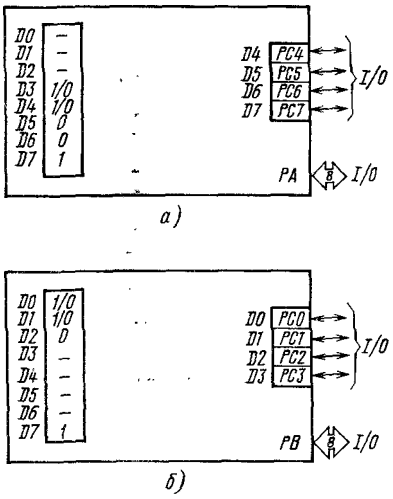


Рис. 3.18. Организация однонаправленного ВВ без квитирования:
а — группа А; б — группа В

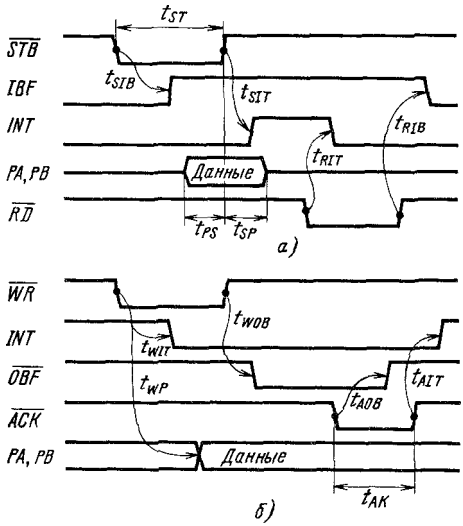


Рис. 3.19. Временные диаграммы режима 1:
а — ввод; б — вывод

Для генерации вышеуказанных сигналов управления применяются отдельные линии порта С в соответствии с рис. 3.20. Свободная от управления часть порта С может выполнять свою стандартную функцию ВВ в режиме 0.

Для управления обменом в режиме 1 со стороны ЦП предусмотрен программный доступ к линиям INT, IBF и $\overline{\text{OBF}}$. Доступ организован через операцию чтения порта С. На рис. 3.20 показано, как при этом интерпретируются отдельные разряды введенных данных, называемые словом состояния SW адаптера. В состав SW входят также флажки разрешения прерывания INTE, управление состоянием которых может быть выполнено с помощью команды BSR с соответствующим параметром. Генерация запроса на прерывание INT и установка связанного с ним одноименного флажка готовности в SW возможна только при установленном флажке INTE. Функция маскирования прерывания позволяет запретить или разрешить работу устройства ВВ, не затрагивая какие-либо другие устройства в структуре прерываний.

Особенности порта А (см. рис. 3.15 и 3.20) дают возможность организовать двунаправленный ВВ, называемый также режимом 2. В данном режиме линии PA7—PA0 выполняют роль двунаправленной трехстабильной шины, управляемой сигналами $\overline{\text{STB}}$, IBF, $\overline{\text{OBF}}$, $\overline{\text{ACK}}$ и INT согласно временным диаграммам на рис. 3.21.

Сигналы IBF и $\overline{\text{OBF}}$ информируют внешнее устройство о готовности принять или передать данные соответственно. Правила их формирования тождественны правилам режима 1. В соответствии с состоянием IBF и $\overline{\text{OBF}}$ внешнее ПУ либо генерирует очередные данные, сопровождая их стробом $\overline{\text{STB}}$, либо формирует сигнал подтверждения приема $\overline{\text{ACK}}$, готовясь к приему данных. Низкий уровень напряжения сигнала $\overline{\text{ACK}}$ открывает выходные буферы порта А, разрешая выдачу

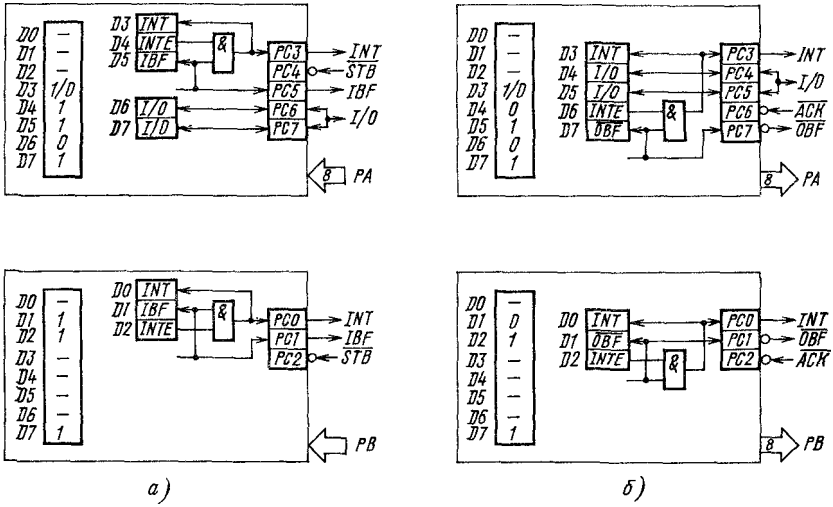


Рис. 3.20. Организация однонаправленного ВВ с квитированием:
а — ввод; б — вывод

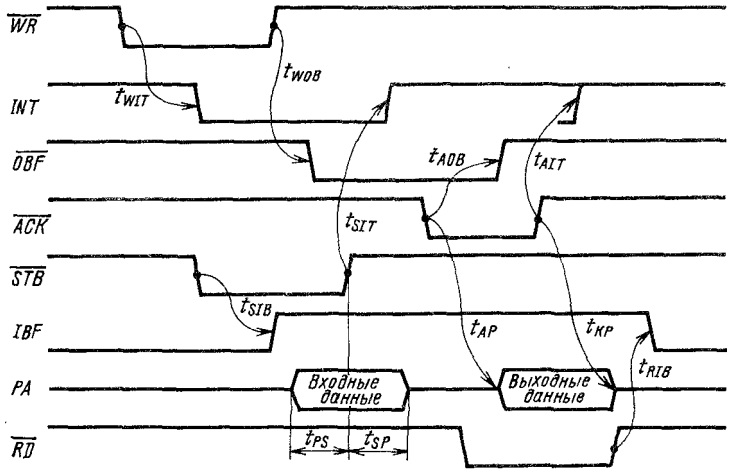


Рис. 3.21. Временны́е диаграммы режима 2

данных на шину. В остальных случаях шина PA имеет высокое выходное сопротивление.

Сигнал INT используется для организации ВВ по прерываниям. Логика его формирования приведена на рис. 3.22. При получении очередного запроса на прерывание ЦП читает слово состояния SW и по флажкам IBF, $\overline{\text{OBF}}$ уточняет статус порта A, выполняя ввод или вывод очередных данных. В SW предусмотрены два независимых флажка разрешения прерывания для ввода и вывода, что дает возможность переводить порт либо в режим ввода, либо в режим вывода выборочно.

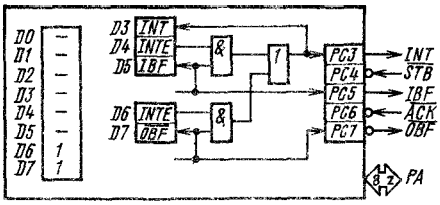


Рис. 3.22. Организация двунаправленного ввода-вывода

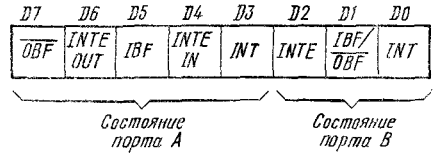


Рис. 3.23. Формат слова состояния SW программируемого периферийного адаптера

Порты А, В и С для работы в том или ином режиме программируются независимо друг от друга. Так, если порт В запрограммирован для ввода данных в режиме 1, то порт А может выполнять любую другую операцию обмена из числа возможных. Свободная от управления часть порта С также программируется либо для ввода, либо для вывода в режим 0, причем младшая половина порта независимо от старшей. Часть порта С, занятая под управление обменом, реализует функцию слова состояния в соответствии с форматом, представленным на рис. 3.23.

Адаптеры типа ВВ55 относятся к приборам первого поколения и являются эффективным средством построения систем ВВ. Их улучшенный вариант ВВ55А осуществляет работу с расширенными стробами записи, генерируемыми системным контроллером ВК38 и МП ВМ85А.

Интерфейс радиальный параллельный. В качестве примера практического использования МБР ИР12 и ППА ВВ55/ВВ55А рассмотрим задачу построения адаптера интерфейса радиального параллельного, удовлетворяющего требованиям шины Microbus, стандарт на которую был разработан Британским комитетом по стандартизации промышленных средств обработки информации в 1969 г. под названием BS 4421. В нашей стране стандарт известен как интерфейс радиальный параллельный (ИРПР) [33]. Базовый вариант интерфейса предусматривает одностороннюю асинхронную передачу информации по параллельной 16/8-разрядной шине D0—D15/D7 (D1—D16/D8) от источника к приемнику на расстояние до 15 м. (В скобках приводятся отечественные обозначения сигналов ИРПР.) Передача байта данных сопровождается двумя управляющими сигналами: управление приемника АС (Acceptance Control) или запрос приемника (ЗП) и управление источника SC (Source Control) или строб источника (СТР), приведенными на рис. 3.24. Данные истинны, когда АС=1 и SC=1. Однако циклы передачи данных могут начинаться и выполняться только при условии, что сигналы готовности приемника А0 (ГП) и источника S0 (ГИ) активны. Скорость передачи данных не регламентирована и зависит от быстродействия обеих сторон.

В состав интерфейса входит ряд необязательных вспомогательных линий. Это прежде всего линии контроля четности DP0 (Data Parity) и DP1 (KP0, KP1) для младшего и старшего байта

данных соответственно, а также линии состояния приемника A1—A8 (СП1—СП8) и линии состояния источника S1—S8 (СИ1—СИ8). Сигналы DP0, DP1 и S1—S8 выдаются аналогично данным, сигналы A1—A8 действительны только при AC=1 и SC=0. В некоторых случаях допускаются асинхронные сигналы состояния источника и приемника. Предусматривается также линия нуля Z (0 В) и экрана S (Э).

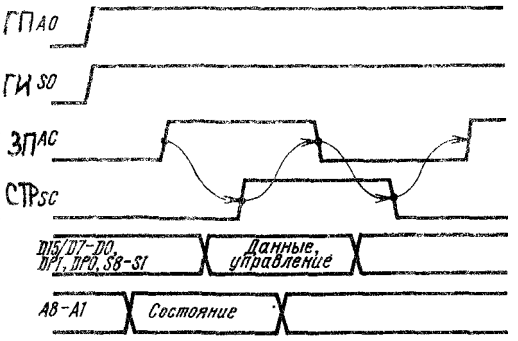


Рис. 3.24. Временные диаграммы протокола ИРПР

Из двух однонаправленных каналов может быть организован один двунаправленный ИРПР-канал с раздельными шинами данных. Для отличия одних и тех же функциональных линий, принадлежащих источнику и приемнику двунаправленного канала в их обозначение добавляется символ А (П)—приемная часть или S (И)—передающая часть, например ACA, SCA, ACS, SCS и т. д.

Функциональное назначение сигналов состояния источника и приемника конкретизируется в зависимости от использования интерфейса: печатающее устройство (ИРПР-ПЧ), дисплей (ИРПР-ВТ), фотосчитыватель, ленточный перфоратор (ИРПР-ПЛ) [33]. Например:

S1 (PV)	Наличие или отсутствие контроля
S2	Внимание
A1 (AE)	Ошибка четности
A2	Автономный режим

Уровень активности сигналов может быть как высоким (ИРПР-ЕС), так и низким (ИРПР-СМ).

Интерфейс ИРПР получил широкое распространение в практике связи МС с ПУ. Многие внешние устройства имеют встроенный интерфейс данного типа. Реализация интерфейса со стороны МС несложна и может быть выполнена на программном уровне (см. рис. 3.5) без дополнительных аппаратных затрат.

Интерфейс ИРПР-СМ используется для реального подключения стандартных ПУ: дисплеев, печатающих устройств, перфораторов, фотосчитывателей и т. д. Он является удобным средством обмена информацией между МС и внешним объектом, в качестве которого может выступать другая МС.

Реализация входного и выходного каналов адаптера ИРПР-СМ на базе МБР ИР12 приведена на рис. 3.25 и 3.26 соответственно. Внутренний интерфейс адаптера удовлетворяет требованиям,

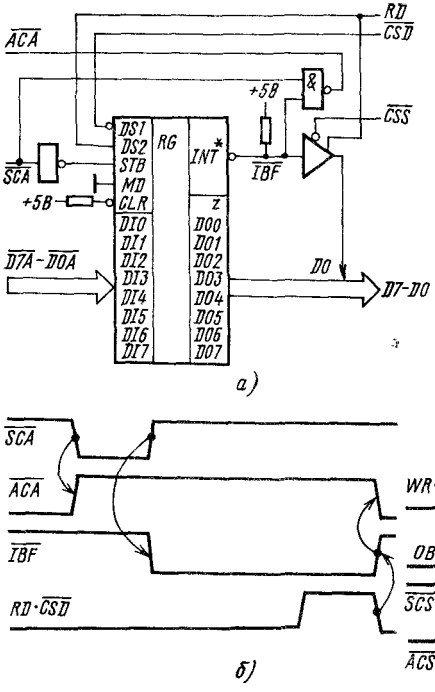


Рис. 3.25. Входной канал адаптера ИРПР-СМ:
а — структурная схема; б — временные диаграммы

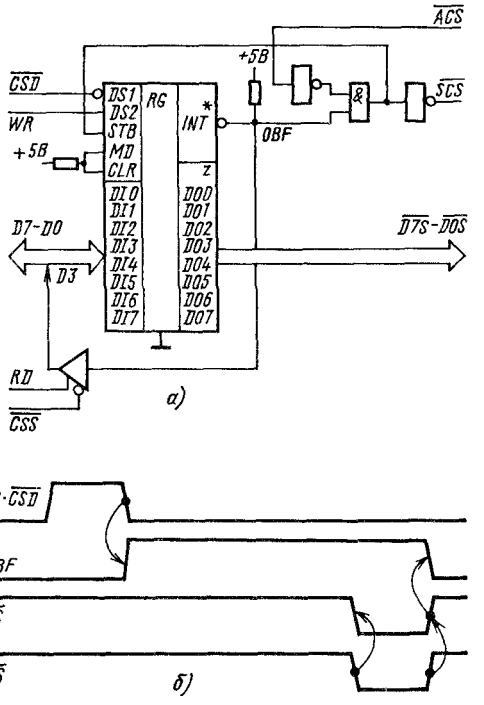


Рис. 3.26. Выходной канал адаптера ИРПР-СМ:
а — структурная схема; б — временные диаграммы

предъявляемым к шине Microbus. Учет низкого уровня активности входной и выходной шин данных ИРПР-СМ выполняется на программном уровне дополнительной инверсией данных. Согласование уровней активности внешней и внутренней шин данных возможно и на аппаратном уровне, однако это связано с вводом инвертирующего буфера, что приводит к росту аппаратных затрат. Нагрузочная способность выходного канала $I_{OL} = 100$ мА. При необходимости ВВ дополнительных сигналов состояния (управления) используется адекватное расширение входного и выходного каналов адаптера.

Простейший драйвер ИРПР-адаптера может содержать только две подпрограммы. Первая подпрограмма TTS — условный ввод с совмещением — возвращает в одном из флажков PSW, например в ZF, признак готовности ввода. При готовности (ZF = 0) возвращаются также введенные данные, для чего используется регистр A:

```
IBF     EQU     01H             ;Маска IBF
TTI:    IN      CSS             ;Ввод слова сос-
                                ;тояния
```

```
        CMA             ;Инверсия слова
                        ;состояния
        ANI     IBF     ;Выделение IBF
        RZ              ;Возврат, если
                        ;нет готовности
        IN      CSD     ;Ввод данных
        CMA             ;Прием нулево-
                        ;го байта
        RET             ;игнорируется
```

Вторая подпрограмма TTO — условный вывод с совмещением. Выводимые данные передаются через регистр A:

```
NOBF    EQU     08H     ;Маска OBF
TTO:    PUSH    B
        MOV     B, A    ;Сохранить сло-
                        ;во данных
        IN      CSS     ;Ввод слова сос-
                        ;тояния
        CMA             ;Инверсия слова
                        ;состояния
        ANI     NOBF    ;Выделение OBF
        MOV     A, B    ;Восстановление
                        ;данных
        POP     B
        RZ              ;Возврат, если
                        ;нет готовности
        CMA             ;Иначе — инвер-
                        ;сия данных
        OUT     CSD     ;Вывод данных
        CMA             ;При выводе ну-
                        ;левого байта
        RET             ;признак вывода
                        ;сбрасывается
```

На основе этих двух подпрограмм могут быть построены все остальные: ВВ с занятием цикла, ввод с эхо-выводом и т. д.

Анализ временны́х диаграмм работы ППА в режиме 1 (см. рис. 3.19) показывает, что при его проектировании были учтены логические требования интерфейса ИРПР. Это дает возможность реализовать ИРПР-адаптер с минимальным числом дополнительных внешних элементов (рис. 3.27). Схема построена на базе ППА ВВ55/ВВ55А и двух шинных формирователей типа ВА87. Последние используются как для промежуточной инверсии входных и выходных шин интерфейса, так и для создания необходимой нагрузочной способности по выходу $I_L = 32$ мА. Возможна их прямая замена на регистры ИР83.

Отличие ИРПР-адаптера состоит в необходимости его инициализации перед началом работы. Это может быть сделано с помощью специальной программы:

```
MS      EQU     0A7H        ;A — вывод в
                            ;режиме 1, B —
                            ;ввод
                            ;в режиме 1, C4,
                            ;C5 — вывод в
                            ;режиме 0
PAINI:  PUSH    PSW
        MVI     A, MS       ;Mode Selection
        OUT     PPA+3
        MVI     A,5         ;INTE_B←1,
                            ;разрешение
        OUT     PPA+3       ;прерываний по
                            ;каналу B
        MVI     A,0DH       ;INTE_A←1,
                            ;разрешение
        OUT     PPA+3       ;прерываний по
                            ;каналу A
        POP     PSW
        RET
```

Линии PC4 и PC5 ППА могут быть применены по усмотрению пользователя как в качестве входных (MS=0AFH), так и выходных (MS=0A7H). Напомним, что при инициализации ППА все регистры данных сбрасываются, следовательно, при выводе PC4=PC5=0. Управление их состоянием выполняется с помощью команд BSR.

Процедуры условного ВВ PAI, PAO могут иметь тот же самый вид, что и TTI, TTO для адаптера ИРПР-СМ на базе ИР12. Отличие состоит в отсутствии команд CMA, так как теперь инверсия осуществляется на аппаратном уровне. Кроме этого в процедурах используются разные адреса портов данных при вводе (CSD=PPA+1) и выводе (CSD=PPA). Адрес порта слова состояния CSS=PPA+3. Здесь под PPA понимается базовый адрес адаптера ВВ55.

3.4. Средства последовательного ввода-вывода

Интерфейс радиальный последовательный. Параллельные каналы обычно используются при расстоянии до 10—15 м. При больших расстояниях стоимость многопроводного кабеля достаточно высока. В этих случаях применяются каналы с последовательной (поразрядной) передачей данных, что позволяет снизить стоимость линий, однако уменьшается пропускная способность

канала. Функция стыка МС с последовательными каналами возлагается на связные адаптеры (СА).

К числу простейших интерфейсов последовательной передачи принадлежит интерфейс радиальный последовательный (ИРПС) или асинхронный старт-стоповый. Интерфейс предусматривает одностороннюю последовательную передачу данных словами по 5—8 бит со скоростями 1200, 2400, 4800, 9600 и 19 200 бит в секунду. Передаваемое слово обрамляется рамкой (рис. 3.28), состоящей из стартового бита, необязательного бита контроля четности и одного-двух стоповых битов. Полученная посылка носит название кадра. Размещение данных в кадре начинается с младших разрядов. Логические состояния линии кодируются значением либо напряжения, либо тока (табл. 3.2). Вход приемника и выход передатчика обозначаются соответственно SID и SOD.

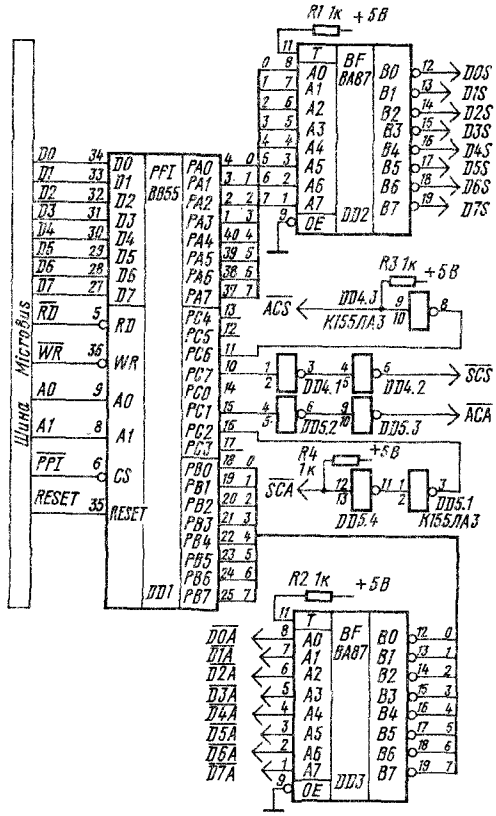


Рис. 3.27. Схема адаптера ИРПР-СМ на базе ВВ55/ВВ55А

Таблица 3.2

Логическое состояние	Линейный выход RS-232-С	Стык С2	Токовая петля	
			20 мА	40 мА
0	≥3 В	+5÷+15 В	0—3 мА	0—10 мА
1	<3 В	−5÷−15 В	15—25 мА	30—50 мА

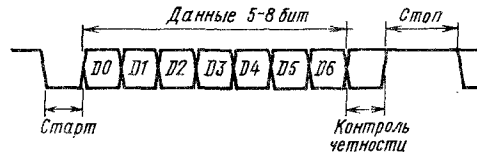


Рис. 3.28. Временны́е диаграммы протокола ИРПС

В некоторых случаях возможно расширение числа физических линий интерфейса введением сигналов управления и состояния, которыми обмениваются контроллер и ПУ, например готовности приема, запроса передачи. Питание линий осуществляется от сети передатчика (активный режим) или приемника (пассивный режим).

Программируемый связной адаптер ВВ51/ВВ51А. В простейшем случае функцию стыка последовательного канала с МС выполняют программируемые связные адаптеры (ПСА), в более сложных случаях — программируемые связные контроллеры и сопроцессоры. К числу однокристальных ПСА относятся БИС:

КР580ВВ51, 8251	Универсальный синхронно-асинхронный приемопередатчик (базовый вариант)
КР580ВВ51А, 8251А, S2657	Универсальный синхронно-асинхронный приемопередатчик (улучшенный вариант)
МС6850	Асинхронный последовательный интерфейс
К1801ВП1-035	Универсальный асинхронный приемопередатчик для МС с Q-шиной

Программируемый связной адаптер КР580ВВ51 (ВВ51) или универсальный синхронно-асинхронный приемопередатчик (УСАПП, USART — Universal Synchronous/Asynchronous Reciver/Transmitter) представляет собой однокристальное программируемое устройство, реализующее интерфейс МС с синхронно-асинхронными каналами последовательной связи. Он является аналогом прибора 8251 фирмы Intel.

В состав ПСА (рис. 3.29) входят передатчик, приемник, буфер шины данных и ряд схем управляющего типа. Основу передатчика составляет 13-разрядный сдвиговый регистр, хранящий очередное выходное слово. Разряды 12, 11 регистра используются для формирования стоп-битов, 10 — для записи контрольного бита, разряды 9—2 предназначены для хранения данных, 1 — для формирования

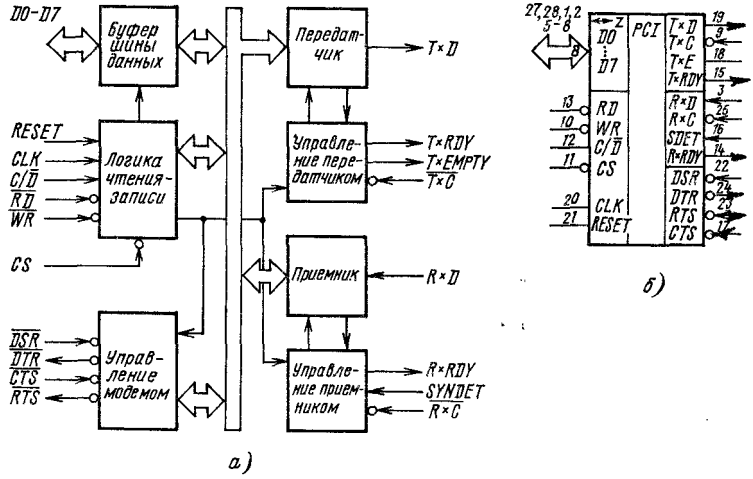


Рис. 3.29. Схема программируемого связного адаптера ВВ51:
а — структурная схема; б — условное графическое обозначение

старт-битов, последний применяется как выходной буфер для последовательного выталкивания слова на линию T×D (Transmitter Data). Управление работой передатчика осуществляется специальной схемой, которая отслеживает прием новых данных, при необходимости добавляет к ним контрольный бит, обрамляет старт-стоповыми битами и синхронизирует вывод из сдвигающего регистра.

Приемник содержит два 9-разрядных сдвиговых регистра. Информация со входа R×D (Resiver Data) последовательно поступает на один из входов (0—3) первого регистра в зависимости от длины передаваемого слова и затем на один из входов (0—3) второго регистра. Управление записью входной информации осуществляется схемой управления приемника, содержащей логику формирования синхроимпульсов приема, счетчик числа принятых битов, схему контроля четности, триггер ошибки четности PE (Parity Error), триггер ошибки кадра FE (Framming Error) и триггер ошибки переполнения OE (Overrun Error).

Буфер шины данных представляет собой 8-разрядное трехстабильное двунаправленное устройство для связи ПСА с МП, которое состоит из входного и выходного регистра данных, а также доступного для чтения регистра слова состояния SW.

Логика чтения-записи включает регистр режима, регистр команд и два регистра синхроимпульсов. Восьмиразрядный регистр режима предназначен для хранения слова выбора режима МС, тогда как регистр команд — для приема команды CI. Два 8-разрядных регистра синхроимпульсов хранят один или два символа синхронизации SYNC. Встроенный в логику чтения-записи компаратор реализует сравнение слов, хранящихся в регистрах синхроимпульсов и в регистрах приемника. Результат сравнения используется для управления работой БИС.

Схема управления модемом служит для обмена с ПУ управляющими сигналами:

$\overline{\text{DSR}}$ (Data Set Ready)	Готовность приемника модема. Проверяется программно
$\overline{\text{DTR}}$ (Data Terminal Ready)	Запрос готовности приемника модема. Управляется программно
$\overline{\text{RTS}}$ (Request To Send)	Запрос готовности передатчика модема. Управляется программно
$\overline{\text{CTS}}$ (Clear To Send)	Готовность передатчика модема. Разрешает ПСА передачу данных

Входные сигналы $\overline{\text{WR}}$ и $\overline{\text{RD}}$ определяют направление потока информации, передаваемой по шине данных из ЦП в ПСА и обратно. По стробу $\overline{\text{WR}}$ ЦП через буфер шины данных записывает в ПСА данные или управляющую информацию. Строб $\overline{\text{RD}}$ обеспечивает чтение данных или информации о состоянии ПСА. При отсутствии сигналов на входах $\overline{\text{WR}}$ и $\overline{\text{RD}}$ обмена информацией с ПСА не производится. Одновременная подача обоих сигналов запрещена. Логический уровень на входе C/$\overline{\text{D}}$ определяет вид информации при обмене, которая может быть либо словом управления/состояния (C/$\overline{\text{D}}$=1) либо байтом данных (C/$\overline{\text{D}}$=0). Все операции по обмену информацией (табл. 3.3) возможны только при низком уровне напряжения на линии выбора кристалла ($\overline{\text{CS}}$=0).

Перед началом работы ПСА должен быть установлен в исходное состояние либо с помощью сигнала сброса, генерируемого МС и подаваемого на вход

RESET микросхемы, либо программным способом с помощью команды CI.IR = 1. Для надежного сброса ПСА длительность сигнала RESET должна быть больше шести периодов тактовой частоты CLK.

Дальнейшее управление работой ПСА осуществляется двумя управляющими словами: MI (Mode Instruction) и CI (Command Instruction) в соответствии с табл. 3.4. Слово MI определяет основной режим работы ПСА, оно должно быть передано после операции сброса первым. Различают два формата MI (рис. 3.30), кодируемых двумя младшими разрядами слова: асинхронного (MI.0≠0 или MI.1≠0) и синхронного (MI.0=0 и MI.1=0) режимов. В обоих случаях имеется возможность программировать длину слова данных DW (поле L2—L1) и тип контроля (разряды EP и PEN). В асинхронном режиме программируется также число стоп-битов (поле S2—S1) и скорость передачи (поле B2—B1), в синхронном режиме можно задать число синхросимволов (SCS) и тип синхронизации (ESD). В синхронном режиме следом за MI должны быть загружены один или два синхросимвола SYNC. После их загрузки или загрузки MI асинхронного режима может быть подано управляющее слово CI.

Слово CI (рис. 3.31) используется для оперативного управления работой ПСА: разрешения (запрещения) приема (передачи), перехода в режим ожидания

Таблица 3.3

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Операция
0	0	1	0	D←Приемник
0	1	0	0	Передатчик←D
1	0	1	0	D←SW
1	1	0	0	CW←D
X	1	1	0	Нет операции
X	X	X	1	Нет операции

Таблица 3.4

Данные D7 - D0	Управляющие сигналы				
	C/$\overline{D}$	$\overline{WR}$	$\overline{RD}$	$\overline{CS}$	RESET
Произвольно	X	X	X	0	1
Mode Instruction	1	0	1	0	0
Синхросимвол 1	1	0	1	0	0
Синхросимвол 2	1	0	1	0	0
Command Instruction	1	0	1	0	0
Данные, Status Word	0	X	X	0	0
Command Instruction	1	0	1	0	0
Данные, Status Word	0	X	X	0	0

Примечание. Синхросимволы 1 и 2 могут отсутствовать.

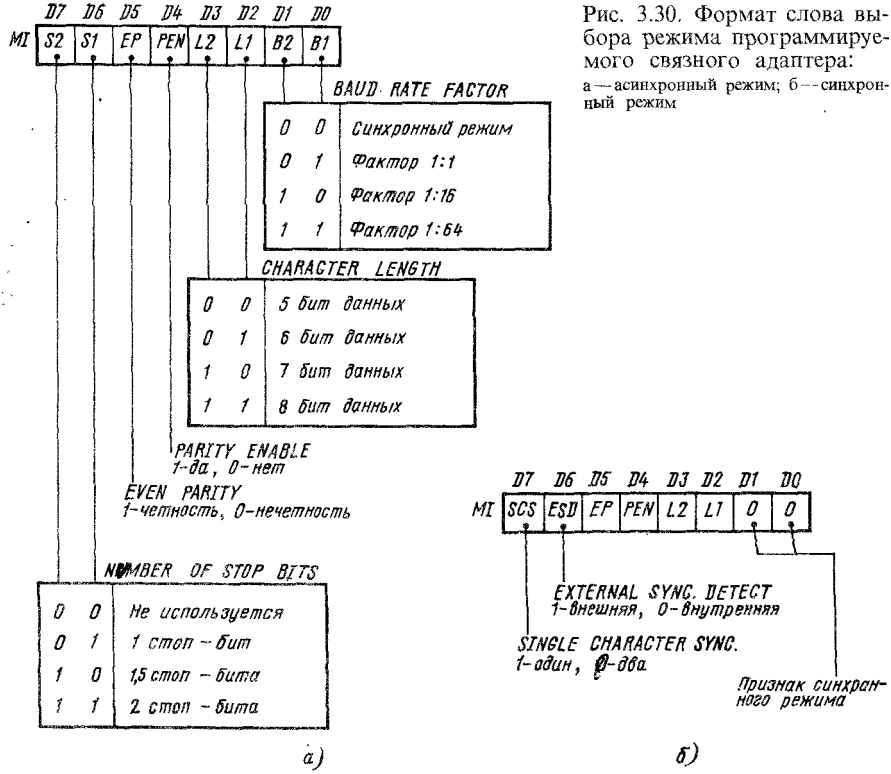


Рис. 3.30. Формат слова выбора режима программируемого связного адаптера:
а — асинхронный режим; б — синхронный режим

синхронизации, а также для программной установки ПСА в исходное состояние с целью его переинициализации (CI.IR = 1).

Для организации программно-управляемого обмена по условию в составе ПСА предусмотрено слово состояния SW (рис. 3.32), в состав которого входят флажки готовности передатчика T × RDY и приемника R × RDY. После выдачи слова данных флажок T × RDY устанавливается в 0 и снова в 1 после передачи DW в буфер передатчика. Аналогично работает флажок R × RDY при заполнении и считывании буфера принятых данных.

Кроме флажков готовности в состав слова состояния входят три признака ошибок. Наличие ошибки не прерывает работу ПСА. Триггеры ошибок устанавливаются в исходное состояние командой сброса ошибки (IC.ER = 1). Чтение SW возможно в любой момент времени, что позволяет МС управлять процессом передачи данных программными средствами.

Режим работы ПСА определяется программными средствами. После записи CI адаптер может быть переведен в один из основных режимов работы или в их допустимую комбинацию. К ним относятся: асинхронная передача, асинхронный прием, синхронная передача, синхронный прием с внутренней синхронизацией, синхронный прием с внешней синхронизацией.

При MI.0 ≠ 0 MI.1 ≠ 0 адаптер переходит в асинхронный старт-стоповый режим работы. Следующая за MI команда разрешения передачи (CI.T × EN = 1) при $\overline{CTS} = 0$ запускает передатчик. Установка флажка готовности T × RDY в 1

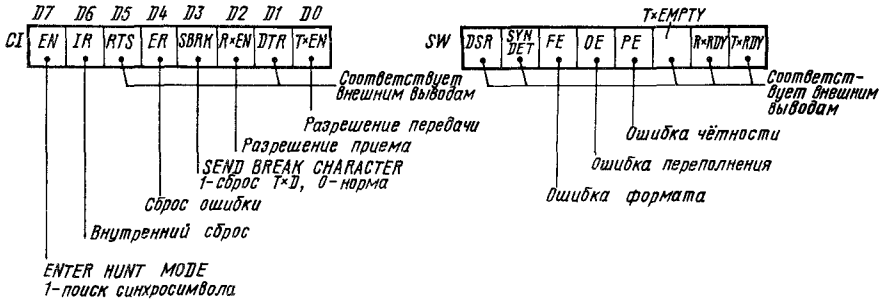


Рис. 3.31. Формат команды программируемого связного адаптера

Рис. 3.32. Формат слова состояния программируемого связного адаптера

сигнализирует МП о готовности передатчика принять данные (рис. 3.33, а). После выдачи данных флажок готовности T×RDY устанавливается в 0 и снова в 1 после передачи байта данных в выходной буфер передатчика. Если буфер еще занят выводом предыдущего байта данных, адаптер будет ждать его освобождения.

Записанные в буфер передатчика данные обрамляются старт-стоповыми битами и при необходимости битом контроля четности. Информация подается на выход T×D в последовательной форме согласно протоколу ИРПС (см. рис. 3.28) с частотой, кратной 1/1, 1/16 или 1/64 частоты на входе $\overline{T\times C}$ в зависимости от значения поля B2—B1 в MI. Смена данных на выходе T×D происходит по срезу $\overline{T\times C}$.

Если ПСА не имеет информации для передачи, то на выходе T×D устанавливается напряжение высокого уровня (сигнал маркирования). При выдаче сигнала маркирования на линии T×EMPTY также устанавливается напряжение высокого уровня, что используется для переключения направления передачи линии, работающей в полудуплексном режиме, на обратное направление. При подаче команды запрета передачи (CI.T×EN=0) на выходе T×D устанавливается напряжение низкого уровня (сигнал вклинивания), как и при подаче команды обрыва (CI.SBRK=1).

По команде разрешения приема (CI.R×EN=1) в асинхронном режиме приемник отслеживает срез сигнала на входе R×D, появление которого воспринимается как приход старт-бита (рис. 3.33, б). Истинность этого события проверяется вторично опросом состояния R×D в середине такта. Если старт-бит не подтверждается (R×D=1), то приемник возвращается в исходное состояние. При подтверждении старт-бита прием кадра разрешается. Считывание очередного бита выполняется в середине такта его передачи. Если при приеме будет обнаружена ошибка четности, то установится в 1 флажок PE. Если при приеме первого такта стоп-бита на линии R×D окажется напряжение низкого уровня, то зафиксируется ошибка формата FE. Прием стоп-бита сигнализирует об окончании текущего кадра. Принятые данные (5—8 бит) передаются в выходной регистр и устанавливается в 1 флажок готовности R×RDY. Если при этом предыдущий символ из регистра еще не был считан, то он теряется, что соответствует установке в 1 флажка переполнения OE. Флажок R×RDY устанавливается в 0 при считывании данных из буфера. Наличие ошибок не приостанавливает работу ПСА, а только информирует об их появлении. Линия SYNDET не используется.

Область применения асинхронного режима ограничена скоростью передачи данных 19,2 кБод. Для работы с высокоскоростными линиями вплоть до 56 (64 для BB51A) кБод применяется синхронный режим передачи.

В синхронный режим ПСА переводится при MI.0=0, MI.1=0. После записи MI должны быть переданы один (MI.SCS=1) или два (MI.SCS=0) синхросимвола. Передача разрешается по команде CI.T×EN=1 при $\overline{\text{CTS}}=0$. Логика установки флажка T×RDY аналогична его установке в асинхронном режиме. После передачи в ПСА слова данных и при $\overline{\text{CTS}}=0$ передатчик начнет выталкивать биты данных со скоростью следования $\overline{\text{T}\times\text{C}}$. Смена последовательных данных на выходе T×D определяется срезом $\overline{\text{T}\times\text{C}}$.

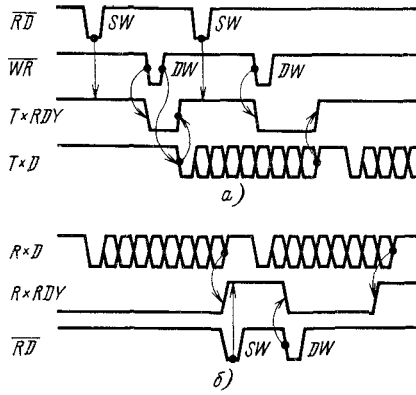


Рис. 3.33. Временные диаграммы передачи (а) и приема (б) в асинхронном режиме

Если новые данные от МП еще не пришли, а буфер передатчика уже пуст для предотвращения потери синхронизации в поток данных автоматически вставляются синхросимволы. При этом на выходе T×EMPTY формируется последовательность импульсов (по импульсу на синхросимвол), указывающих на передачу последнего бита каждого синхросимвола (рис. 3.34). Когда в передатчик ПСА поступают новые данные, на T×EMPTY устанавливается напряжение низкого уровня.

При синхронном приеме возможны два режима: с внутренней (MI.ESD=0) и внешней (MI.ESD=1) синхронизацией. В обоих случаях работа начинается по команде IC.R×EN=1. При этом в режиме внутренней синхронизации ПСА входит в состояние поиска синхросимволов (режим активного ожидания). Принимаемая по фронту $\overline{\text{R}\times\text{C}}$ информация R×D непрерывно сравнивается сначала с первым, а затем при необходимости и со вторым синхросимволом. При обнаружении синхросимволов во время приема последнего бита ПСА устанавливает на выходной линии SYNDET напряжение высокого уровня, что означает вхождение в синхронизацию. При сбросе и чтении SW на линии SYNDET устанавливается напряжение низкого уровня. При внешней синхронизации SYNDET работает на вход. Синхросимволы обнаруживаются внешними средствами, устанавливающими на линии SYNDET напряжение высокого уровня, что выводит ПСА из режима ожидания синхронизации и разрешает прием данных по каждому фронту $\overline{\text{R}\times\text{C}}$. Длительность сигнала SYNDET должна быть больше или равняться периоду $\overline{\text{R}\times\text{C}}$. Данные могут приниматься по частям при высоком уровне напряжения на линии SYNDET. Асинхронность SYNDET и $\overline{\text{R}\times\text{C}}$ может вызвать задержку данных на один период следования частоты $\overline{\text{R}\times\text{C}}$. При синхронном приеме данных ошибки четности и переполнения проверяются, как и в асинхронном.

В состав CI входят флажки, управляющие выходами $\overline{\text{DTR}}$ и $\overline{\text{RTS}}$. Аналогично SW содержит ряд флажков, отражающих текущее состояние физических линий $\overline{\text{DSR}}$, SYNDET, T×EMPTY, R×RDY, T×RDY. Такое дублирование позволяет реализовать обмен данными с ПУ как по методу программного опроса, так и по прерываниям. Максимальная задержка обновления информации в SW составляет 16 периодов CLK.

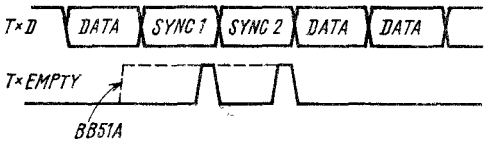


Рис. 3.34. Временные диаграммы передачи в синхронном режиме

Отличие состояний выхода T×RDY и флажка T×RDY заключается в том, что состояние последнего не зависит от состояний линии $\overline{CTS}$ и флажка CI.T×EN, тогда как состояние выхода T×RDY определяется произведением

$$T\times RDY = SW.T\times RDY \& CI.T\times EN \& \overline{CTS}$$

Устройство КР580ВВ51А (ВВ51А) и его аналоги 8251А, S2657 представляют собой усовершенствованный вариант стандартного ПСА типа ВВ51. Новый ПСА ориентирован на работу с такими высокопроизводительными микропроцессорами, как ВМ85А.

Микросхема ВВ51А обладает всеми свойствами ВВ51 и имеет следующие расширения:

введена двойная буферация данных с раздельными регистрами ВВ для управления, состояния, данных;

в асинхронном режиме приемник ПСА автоматически определяет и обрабатывает состояние вклинивания (R×D=0), при обнаружении которого флажок SYNDET устанавливается в 1, а уровень напряжения на выходной линии SYNDET становится высоким; при сбросе ПСА или обнаружении фронта R×D флажок SYNDET устанавливается в 0, а линия SYNDET возвращается в исходное состояние;

усовершенствована схема инициализации приемника, устраняющая его срабатывание в состоянии вклинивания;

по окончании передачи линия T×D возвращается в состояние маркирования до тех пор, пока не будет подана команда SBRK;

усовершенствована логика запрета передачи, отодвигающая запрет к моменту, когда все ранее записанные данные не будут переданы;

в синхронном режиме с внешней синхронизацией после вхождения в синхронизм (появление фронта на входной линии SYNDET) сигнал SYNDET может быть снят;

минимизирована возможность детектирования ложного сигнала синхронизации за счет использования двойного символа синхронизации, а также предварительной установки всех разрядов буферного регистра приемника в состояние 1 по команде ENTER HUNT MODE;

стробы $\overline{RD}$ и $\overline{WR}$ не влияют на внутренние устройства до прихода сигнала $\overline{CS}$;

улучшены временные параметры ПСА.

Адаптер ИРПС на базе ВВ51. С помощью ПСА ВВ51 достаточно просто реализуется ИРПС-адаптер (рис. 3.35), удовлетворяющий требованиям шины Microbus. В его состав еще входят две буферные схемы и генератор скорости передачи.

Существуют различные варианты построения буферных схем, в которых может быть предусмотрена также оптронная развязка линии. На рис. 3.35 приведено два наиболее простых варианта. Буферная логика обеспечивает передачу логического 0 высоким уровнем напряжения +12 В (R_{OUT}=1,3 кОм) и логической 1 низким уровнем — 12 В (R_{OUT}=300 Ом), что удовлетворяет требованиям на стык С2 (стандарт RS=232=С). В схемах предусмотре-

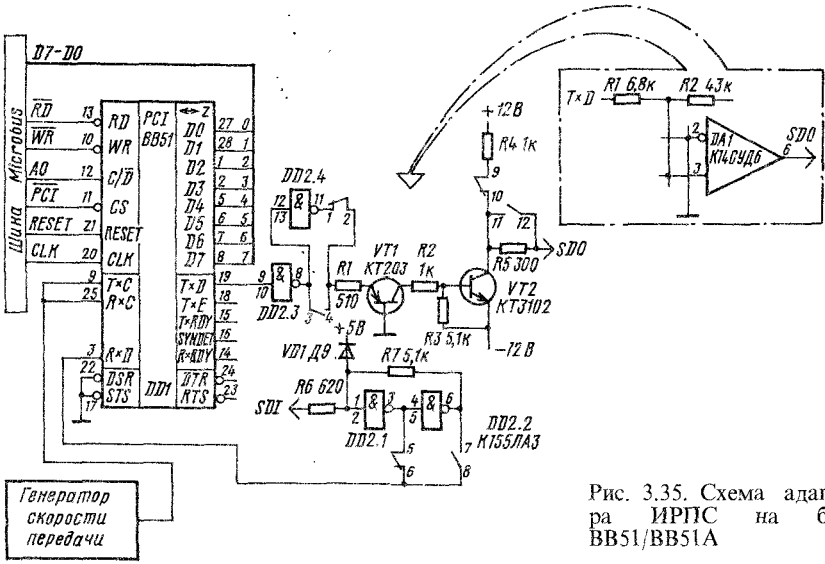


Рис. 3.35. Схема адаптера ИРПС на базе ВВ51/ВВ51А

на возможность устранения существующей инверсии во входных и выходных каналах последовательной связи, что может быть полезно при организации межмашинной связи магистрального типа. В этом случае для управления локальной сетью применяется магистраль с последовательным протоколом передачи. Доступ к магистрали регулируется с помощью специального маркера доступа, который передается от одного абонента сети к другому.

Для инициализации адаптера используется процедура SAINI, которая программирует ПСА для работы в асинхронном режиме с 7-битовым полем данных, контролем четности и двумя стоп-битами. Фактор скорости устанавливается равным 1:16. Процедура записывается следующим образом:

```
SAINI:  PUSH    PSW
        XRA     A           ;Перевод ПСА
        OUT     PCI+1       ;в состояние реагиро-
                            ;вания
        OUT     PCI+1       ;на команду RESET
        OUT     PCI+1
        MVI     A, 40H      ;Сброс
        OUT     PCI+1
        MVI     A, OFAH     ;MI
        OUT     PCI+1
        MVI     A,5         ;ВВ разрешен
        OUT     PCI+1
        CALL    SAI         ;Условный ввод
        POP     PSW
        RET
```

Перед выдачей командного слова выбора режима MI микросхема приводится в исходное состояние. Для этого в управляющий порт PCI+1 выдается три нулевых байта и команда сброса. Предварительная выдача трех нулей обеспечивает перевод БИС ВВ51 независимо от ее текущего состояния в рабочий режим, в котором она сможет правильно отреагировать на команду программного сброса (код 40Н). Команда с кодом 5 разрешает одновременную работу приемника и передатчика. После разрешения работы следует выполнить холостой цикл ввода по готовности, уничтожающий ложный символ, который может возникнуть в приемнике адаптера во время процедуры инициализации.

Процедуры условного ввода SAI и условного вывода SAO можно построить аналогично процедурам PAI и PAO, обеспечив единый механизм обращения к ним со стороны ПО.

Процедура условного ввода:

```
SAI:  IN    PCI+1     ;Ввод SW
      ANI   2         ;Выделение флажка
                      ;R×RDY
      RZ              ;Возврат, если не готов
      IN    PCI       ;Иначе — ввод данных
      RET
```

Процедура условного вывода:

```
SAO:  PUSH  B
      MOV   B,A       ;Сохранить данные вы-
                      ;вода
      IN    PCI+1     ;Ввод SW
      ANI   1         ;Выделение флажка
                      ;T×RDY
      MOV   A,B       ;Восстановление дан-
                      ;ных
      POP   B
      RZ              ;Возврат, если не готов
      OUT   PCI       ;Иначе — вывод данных
      RET
```

Процедуры SAI и SAO предполагают передачу символа через аккумулятор. Признаком выполнения ввода или вывода очередного символа служит сброшенный флажок нуля Z в слове состояния программы PSW микропроцессора.

Три вышеприведенных программы служат примером простейшего драйвера ВВ через ИРПС-адаптер. На его основе можно построить более сложные процедуры ВВ, такие, как ввод символа с занятием цикла и эхо-ответом или вывод байта в шестнадцатеричном формате HEX и т. д.

Скорость передачи определяется частотой тактовых импульсов, формируемых генератором скорости передачи. При этом не

следует забывать о дополнительном делении частоты на 16 внутренними средствами ПСА. Реализация генератора будет рассмотрена в § 3.7.

3.5. Система прерываний

Понятие прерывания центрального процессора. В тех случаях, когда в МС существует дефицит времени ЦП, а периферийные устройства работают медленно, используется их обслуживание по прерываниям. Например, алфавитно-цифровое печатающее устройство (АЦПУ) обеспечивает печать символа с номинальной скоростью 100 знаков в секунду или 1 знак за 10 мс. Если предположить, что цикл обращения к памяти составляет 2 мкс, то для вывода одного знака ЦП потребуется около 0,1 мс (см. процедуру РАО ИРПР-адаптера). Следовательно, из 10 мс (интервал вывода одного знака) 9,9 мс ЦП простаивает, ожидая готовности АЦПУ к приему нового знака.

Эффективность МС может быть существенно повышена, если отказаться от малопроизводительного ожидания готовности на программном уровне и передать функцию специальным аппаратным средствам. В это время ЦП может выполнять некоторую полезную работу, связанную с обработкой данных или обслуживанием других ПУ.

При готовности приступить к очередной операции ВВ устройство посылает в ЦП запрос на прерывание, по получении которого он временно приостанавливает выполнение текущей программы и переходит на обслуживание ПУ. После этого ЦП возвращается к прерванной программе, продолжая ее с момента приостанова. Обслуживание прерываний осуществляется в незаметном для основной программы режиме, поэтому их наличие прямо не влияет на работу последней, за исключением времени ее исполнения. Следует отметить, что обслуживание прерываний приводит к изменению общего состояния МС, что в дальнейшем может сказаться и на поведении основной программы, однако такое воздействие прерываний не будет прямым.

Обслуживание ВВ по прерываниям (рис. 3.36) является альтернативой программно-управляемому обмену (см. рис. 3.4). Если при чисто программном управлении как начало процедуры, так и непосредственное ее исполнение находятся под управлением программы, то обслуживание по прерываниям инициируется аппаратными средствами. Совокупность этих

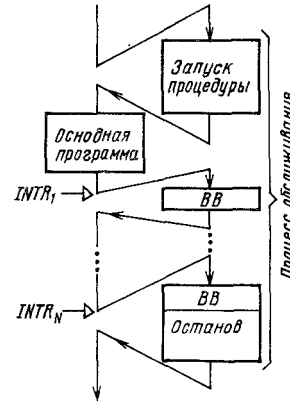


Рис. 3.36. Обслуживание ввода-вывода по прерываниям

средств, команд и программ их обслуживания называется системой прерываний. Прежде чем обслуживание ПУ по прерываниям будет начато, система прерываний должна быть настроена на это. Функция запуска МС возлагается на специальную процедуру инициализации обслуживания. После окончания работы с ПУ также может потребоваться специальная процедура, которая завершит эту работу.

Для того чтобы прерванная программа могла быть продолжена после обслуживания очередного запроса на прерывание с того места, на котором она была приостановлена, состояние ЦП должно быть восстановлено. Сохранность состояний сигналов управления в аппаратуре обеспечивается тем, что переход к обработке прерывания и возврат к прерванной программе осуществляется в строго определенные моменты времени, когда состояния этих сигналов однозначны и известны, т. е. в конце очередного командного цикла перед фазой выборки последующего. Состояние программно-доступных регистров может быть сохранено в памяти МС, а затем восстановлено непосредственно перед возвратом в прерванную программу. Этот процесс называется контекстным переключением и выполняется как программными, так и аппаратными средствами.

Организация радиальной системы прерываний. Физический интерфейс наиболее простой системы прерываний может быть представлен единственной линией IRQ (Interrupt Request), высокий уровень напряжения на которой воспринимается как запрос на прерывание. Для программиста такая система прерываний представляется в виде отдельной точки входа в процедуру обслуживания. Всякий раз, когда ЦП воспринимает запрос на прерывание, он активизирует процедуру обслуживания, передавая ее стартовый адрес в PC. Чтобы не потерялось старое содержимое PC, которое является адресом возврата в прерванную программу, оно должно быть автоматически где-то сохранено. Лучше всего для этой цели использовать системный стек, тогда возврат к прерванной программе будет заключаться в передаче управления по адресу на вершине стека. В этом же стеке может храниться и текущий контекст ЦП. При этом контекстные переключения выполняются с помощью обычных команд типа PUSH и POP.

Процесс обработки запроса на прерывание во многом подобен процессам вызова и возврата из подпрограмм. Однако в первом случае вызов осуществляется командой CALL, которая сформирована и подставлена в общую командную последовательность с помощью аппаратуры системы прерываний, во втором случае команда CALL действительно присутствует в последовательности инструкций. По этой причине запросы на прерывания часто называют аппаратными вызовами подпрограмм.

Для увеличения числа одновременно обслуживаемых источников прерываний в систему вводится несколько линий с фиксиро-

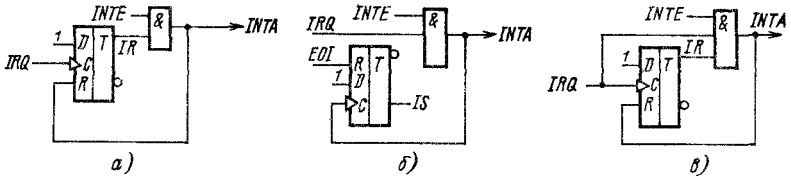


Рис. 3.37. Типы входов для приема запросов на прерывания:
а — динамический; б — статический; в — комбинированный

ванными стартовыми адресами подпрограмм обслуживания. Примером такой системы, получившей название радиальной, служит подсистема BM85A, организованная четырьмя входами: TRAP, RST 7.5, RST 6.5, RST 5.5 (см. табл. 2.7).

Обычно часть радиальных линий резервируется для приема внутренних прерываний ЦП, отражающих его критические состояния и требующих немедленного обслуживания. Остальные отводятся для приема внешних (по отношению к ЦП) запросов. В приведенном примере все четыре линии являются внешними.

Введенный при рассмотрении радиального запроса динамический вход IRQ (рис. 3.37, а) имеет низкую помехоустойчивость, так как любая помеха на линии может вызвать прерывание. Повышение помехоустойчивости связано с введением линий статического типа (рис. 3.37, б), от которых запрос на прерывание воспринимается исключительно по уровню. Однако в этом случае возникает проблема блокировки запроса с момента его подтверждения (иначе возможен повторный захват уже принятого к обслуживанию запроса), которая решается с помощью триггера блокировки IS (In Service), устанавливаемого при подтверждении прерывания. Сброс триггера выполняется программными средствами с помощью операции EOI (End of Interrupt) после снятия принятого к обслуживанию запроса. Существует также комбинированный тип входа, когда после фиксации запроса по перепаду требуется его подтверждение уровнем напряжения (рис. 3.37, в). Все типы входов находят самое широкое применение в практических системах прерываний. Так, в радиальной системе 8085A линия RST7.5 является входом динамического типа, RST6.5 и RST5.5 — статического, TRAP — комбинированного.

Практически во всех системах прерываний предусмотрен дополнительный механизм программно-управляемой блокировки запросов, который реализуется с помощью набора флажков, разрешающих или запрещающих восприятие запросов на прерывания. Эти флажки либо входят в состав PSW, либо упаковываются в отдельный регистр маски. Следует отметить, что маска не влияет на фиксацию запроса по динамическому входу триггером IR (Interrupt Request), однако дальнейшее его прохождение блокируется. В случае статической линии роль маски может выполнять триггер IS, например вход INT, и триггер INTE в МП BM80.

Контекстные переключения, выполняемые командами PUSH и POP, могут занимать значительное время. Для его сокращения вводятся специальные команды PUSHA и POPA (МП типов 80186 и 80286), которые сохраняют в стеке и восстанавливают сразу весь (A-All) набор регистров МП. Аналогично расширяют и функцию командной пары CALL, RET, преобразуя ее к новой взаимно согласованной паре TRAP, RTI (МП типа К1801ВМ1) или INT, IRET (МП типов К1810ВМ86/ВМ88, 80186, 80286), которая одновременно с PC сохраняет или восстанавливает PSW, а в некоторых случаях и другие регистры МП. Еще один механизм сохранения контекста связан с переключением регистровых наборов, как это выполнено в МП Z80 и однокристальных МК серии К1816.

В зависимости от числа подтвержденных запросов, одновременно находящихся на обслуживании, различают одно- и многоуровневые системы прерываний. В одноуровневой системе в каждый момент времени допускается лишь один подтвержденный запрос. Обработка всех других запросов откладывается до окончания текущего обслуживания. Блокировка запросов в одноуровневой системе осуществляется общим для всех триггером IS, фиксирующим факт обслуживания прерывания. Триггер устанавливается любым сигналом подтверждения прерывания INTA и сбрасывается командой окончания прерывания EOI. Примером одноуровневой системы с двумя источниками запросов служит система прерываний однокристального МК К1816ВЕ048.

Если несколько устройств одновременно запросили обслуживание, система прерываний выбирает одно из них. Выбор осуществляется на основании приоритета каждого из запросов, который отражает важность и срочность его обслуживания. При расстановке приоритетов учитываются: частота обслуживания запроса, длительность процесса обслуживания, последствия задержки обслуживания и др.

Рассматривая организацию приоритетов, следует выделить систему с фиксированными линейно упорядоченными приоритетами, которая является наиболее практичной и естественной.

Повышение гибкости системы приоритетов связано с их динамическим изменением по заданному алгоритму или при программировании. Однако в каждый отдельный момент времени все приоритеты строго упорядочены, что обеспечивает однозначный выбор одного из них.

Широко применяемой системой динамически изменяемых приоритетов является циклическая. В ней после каждого очередного обслуживания запроса происходит циклический сдвиг приоритетов с присвоением нижнего только что обработанному. Такая схема приводит к равномерному распределению «внимания» ЦП между всем множеством запросов и может быть использована при

обслуживании группы одинаковых устройств, когда выделение какого-либо из них нежелательно.

Многоуровневая система разрешает многократные (по числу уровней) прерывания одних процедур обслуживания другими. Для этого каждому уровню ставится в соответствие некоторое подмножество запросов из их общего числа и строго упорядоченный приоритет. Процедуры обслуживания некоторого уровня могут быть прерваны лишь запросами более высокого уровня. Фоновую работу ЦП, связанную с самым нижним приоритетом, может прервать любой запрос.

Для работы многоуровневой системы прерываний необходимо знать приоритет текущей процедуры, который соответствует приоритету процессора. Приоритеты запросов сравниваются с приоритетом процессора и, если последний выше, прерывают его работу. При подтверждении прерывания приоритет процессора повышается, принимая значение, равное приоритету запроса. Для разрешения конфликта внутри группы запросов одного уровня существует вторичная система приоритетов.

Примером МС с двухуровневой системой прерываний и пятью источниками запросов служит однокристальный МК К1816ВЕ51. В нем всем источникам программным методом назначается первый или второй уровень. Текущий приоритет процессора отображается с помощью двух IS-триггеров. Вторичная система приоритетов имеет фиксированную структуру и охватывает все пять источников.

Примером 8-уровневой системы прерываний с восемью источниками запросов является двухкристальная пара ВМ80+ВН59, запрограммированная для работы в режиме строгого вложения приоритетов. Здесь каждому уровню соответствует единственный источник запросов, поэтому необходимость во вторичных приоритетах отпадает. Для отображения приоритета процессора используется 8-разрядный регистр IS БИС КР580ВН59.

Расширение радиальной системы прерываний методом поллинга. Каждая внешняя радиальная линия IRQ с фиксированным вектором прерывания может быть превращена в магистраль, которая по схеме «монтажное ИЛИ» объединяет запросы от нескольких источников прерываний. Однако в этом случае после принятия общего запроса к обслуживанию возникает задача идентификации источника, выставившего запрос, и передачи управления на соответствующую процедуру обслуживания, которая решается только программными методами с помощью специальной процедуры POLL, называемой поллингом.

Функция поллинга состоит в последовательном опросе состояния всех устройств, связанных с данной линией запросов (рис. 3.38), и выявлении готового к обслуживанию. При опросе используются стандартные подпрограммы проверки готовности, которые входят в драйверы устройств:

```
POLL:  —
       —
       CALL    STATUS      ;Проверка готовности
       JNZ     SERVICE     ;и переход, если готов
```

Конкретный вид процедур проверки готовности STATUS и обслуживания SERVICE зависит от типа ПУ. В подпрограмме предполагается, что признак готовности возвращается через флажок нулевого результата Z. Простейший вид подпрограммы для портов условного ВВ следующий:

```
STATUS: IN      SW          ;Ввод слова состояния
        ANI     RDY         ;Выделение флажка го-
        RET                 ;товности
```

Опрашиваемый подпрограммой STATUS флажок готовности RDY обычно является и флажком потенциального запроса на прерывание IRQ. После обслуживания устройства (операция ВВ для портов) флажок RDY сбрасывается, автоматически снимая запрос на обслуживание. Затем может быть дано разрешение на прием новых запросов от устройств, которые к данному моменту уже выставлены. Однако такая схема действительна только для входов статического типа. При использовании динамических входов ряд новых запросов, пришедших от ПУ во время обслуживания, может быть потерян. Для их восстановления в конце текущего цикла обслуживания следует повторить процедуру POLL. Возврат к прерванной программе разрешается только после обслуживания всех устройств.

Процедура опроса состояния готовности может потребовать значительных временны́х затрат. Для их сокращения подпрограммы STATUS должны быть оптимизированы, а возможно, и объединены в единую подпрограмму. Однако это приведет к потере гибкости системы прерываний.

Организация векторной системы прерываний. Повышение эффективности системы прерываний связано с параметризацией команды CALL и передачей функции генерации этого параметра, названного вектором прерывания, внешним средствам. Интерфейс векторного запроса расширен шиной VECT для ввода вектора прерывания. Ее физическое совмещение с шиной данных потребовало включения в системную магистраль новой командной линии подтверждения прерывания $\overline{\text{INTA}}$. Временны́е диаграммы цикла ввода вектора прерывания аналогичны диаграммам цикла чтения памяти с заменой строба $\overline{\text{MRDC}}$ на строб $\overline{\text{INTA}}$.

Выпускаемые промышленностью МП осуществляют как радиальные так и векторные прерывания (табл.3.5). При этом вектор может представлять полную команду CALL (МП типов ВМ80/8085А), ее адресную часть, отдельное поле полного адреса и

т. д. В более совершенных системах векторного прерывания (МП типов ВМ86/ВМ88, К1801ВМ1) введен уровень преобразования физического вектора в стартовый адрес процедуры обслуживания, который выполняется с помощью таблицы прерываний IDT (Interrupt Descriptor Table). При этом вектор рассматривается как индекс таблицы, которая обычно размещается в системной памяти начиная с нулевого адреса. В новом МП 80286 [52] введен специальный базовый регистр IDTR, позволяющий размещать таблицу прерываний в любой части адресного пространства памяти.

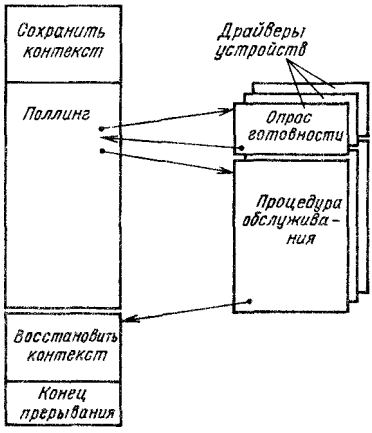


Рис. 3.38. Организация обслуживания по поллингу

Таблица IDT кроме стартового адреса может содержать дополнительную информацию, например начальное состояние PSW. В состав слова состояния программы обычно входит набор наиболее важных флажков и полей, управляющих системой прерываний, в частности приоритет процессора. Подтверждение прерывания автоматически сменяет их состояние, подготавливая систему прерываний к новому уровню обработки.

Аналогичным преобразованиям подвергаются и запросы радиальных прерываний. Обычно они соотносятся с некоторыми векторными прерываниями и называются запросами с фиксированными векторами.

Используя схему векторного прерывания с одним и тем же вектором легко получить линию запросов радиального типа. Так, для процессора на базе МП ВМ80 (см. рис. 2.17) не потребуется никаких вспомогательных схем. Действительно, при каждом

Таблица 3.5

Тип микропроцессора	Запросы		Тип микропроцессора	Запросы	
	векторного прерывания	с фиксированным вектором		векторного прерывания	с фиксированным вектором
КР580ВМ80	INT	Нет	К1816ВЕ51	—	INTO
К1821ВМ85А	INTR	TRAP			INT1
		RST7.5	К1810ВМ86	INTR	NM1
		RST6.5	К1801ВМ1	VIRQ	IRQ1
		RST5.5			IRQ2
		INT			IRQ3
			80286	INTR	NMI

стробе $\overline{\text{INTA}}$ МП будет принимать состояние свободной шины данных. Для приведенной на рис. 2.17 схемы ЦП с прямой шиной, нагруженной резисторами, подключенными к источнику питания +5 В, цикл INTA закончится вводом вектора 0FFH, что соответствует команде RST 7. В результате этого управление будет передано на стартовый адрес 0038H. В случае инверсной шины ввод ее свободного состояния содержит код 00H, соответствующий команде NOP.

Существует более простой метод ввода вектора, эквивалентного RST 7. Для этого следует свободный выход INTA системного контроллера ВК28/ВК38 подключить через резистор 1 кОм к источнику питания +12 В. Контроллер сам во время циклов INTA будет генерировать команду RST7 на внутреннюю шину данных МП ВМ80 независимо от состояния системной шины данных.

На основе вышеизложенной методики можно легко построить поллинговые расширения системы прерываний с одним уровнем. Для этого следует флажок разрешения прерывания INTE (вариант флажка IS) устанавливать только по окончании обслуживания. Разрешение прерываний выполняется командой EI (вариант операции EOI) перед возвратом RET. Особенность команды EI состоит в том, что прерывания разрешаются только после исполнения последующего командного цикла. Это предотвратит возможное переполнение стека при серии плотно следующих друг за другом запросов. Задача построения процедуры POLL облегчается тем, что линия INTR является статической.

Другой способ расширения системы прерываний состоит в проектировании внешних программно-управляемых средств, которые собирают вторичные (периферийные) радиальные запросы IRQ1—IRQN и формируют из них внутрисистемное векторное прерывание. Существуют два основных подхода при решении данной задачи. Первый подход—передача этих функций непосредственно ПУ и, следовательно, решается методом децентрализованного управления. Этот подход нашел свое практическое воплощение в МПК БИС, ориентированных на работу с Q-шиной. Второй подход состоит в передаче функции формирования векторного прерывания специальному устройству (рис. 3.39)—контроллеру прерываний. Сам контроллер прерываний может рассматриваться как расширение процессора, по этой причине его часто называют сопроцессором обработки прерываний. Данный подход удовлетворяет стандарту на системный интерфейс периферийных БИС типа Microbus, который предусматривает генерацию исключительно радиальных запросов.

Простейший вариант построения внешних средств формирования вектора средствами ПУ носит ярко выраженный шлейфовый характер (рис. 3.40). В нем копируется логика работы программы поллинга, перенося ее на аппаратные средства. Каждый запрос

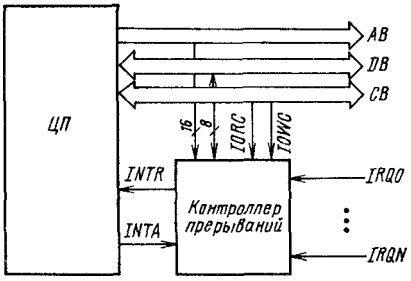


Рис. 3.39. Подключение контроллера прерываний к центральному процессору

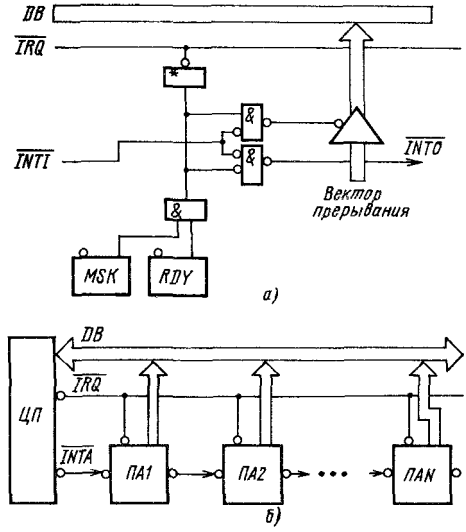


Рис. 3.40. Шлейфовая структура системы прерываний:
а — логика опроса периферийных устройств; б — схема подключения к центральному процессору

шлейфа может быть индивидуально замаскирован с помощью программно-доступного по записи триггера маски MSK. Обычно этот триггер входит в состав CW устройства и доступен для обратного контроля через SW. Вектор прерывания устройства может быть как фиксированным, так и программно-управляемым. При использовании шлейфа совместно с ЦП на базе ВМ80 вектор имеет вид 11NNN111, соответствующий команде RSTN, N=0—7.

Основной недостаток шлейфовой структуры — трудность управления приоритетами. Устройства, стоящие в INTA-цепочке ближе к ЦП, обладают более высоким приоритетом, поэтому изменение приоритетов сводится к изменению последовательности опроса, для чего может потребоваться специальная логика арбитража. Последняя может выполняться как централизованными, так и децентрализованными средствами.

3.6. Программируемый контроллер прерываний ВН59

Система векторных прерываний ВМ80/ВМ85А может быть построена различными способами. Наиболее эффективно она реализуется с помощью специальной БИС типа КР580ВН59 (ВН59), называемой программируемым контроллером прерываний (ПКП). Микросхема выполняется по *n*-МОП-технологии и размещается в 28-выводном корпусе с вертикальным расположением выводов. Она совместима с ТТЛ-схемами. Для работы БИС требуется единственный источник питания +5 В.

На основе ПКП ВН59 формируется 8-уровневая приоритетная система векторных прерываний для МП типов ВМ80/ВМ85А. Несколько контроллеров ВН59 могут соединяться каскадно для расширения числа уровней прерывания до 64. Прибор обеспечивает работу в нескольких режимах, позволяя оптимизировать

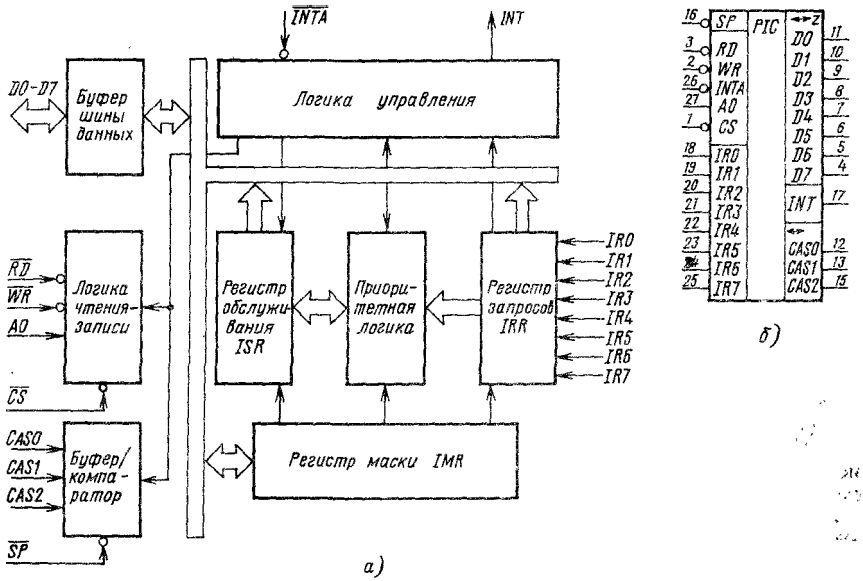


Рис. 3.41. Программируемый контроллер прерываний ВН59:

а — структурная схема; б — условное графическое обозначение

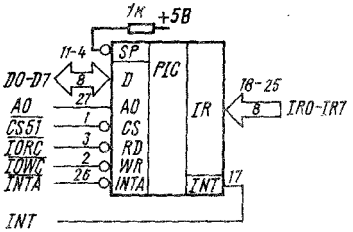


Рис. 3.42. Автономное включение ВН59

логику работы системы прерываний согласно требованиям, предъявляемым к МС. Структурная схема и условное графическое обозначение ПКП приведены на рис. 3.41.

Рассмотрим работу ВН59 в автономном режиме (рис. 3.42). Один или несколько запросов на прерывания (переход из 0 в 1) подаются на входы IR0—IR7 и запоминаются в регистре запросов IRR (Interrupt Request Register). Регистр обслуживания ISR (In Service Register) содержит все запросы, которые в данное время находятся в стадии обработки. Регистр маски IMR (Interrupt Mask Register) используется для маскирования отдельных уровней. Запрет некоторого уровня соответствует единице в соответствующем разряде IMR.

Приоритетная логика выбирает разрешенный запрос на прерывание с наивысшим приоритетом из числа поступивших и сравнивает его с текущим приоритетом запросов, находящихся на обслуживании. При превышении последнего ПКП генерирует сигнал INT (рис. 3.43). Микропроцессор подтверждает прием запроса INT генерацией строба INTA, под воздействием которого запрос с *высшим приоритетом* из IRR *фиксируется* в *соответствующем разряде* ISR. Принятый к обслуживанию IRR-бит сбрасывается и прием нового запроса разрешается. Одновременно с этим ПКП генерирует код команды CALL (0CDH), который принимается МП. В ответ ВМ80/ВМ85А инициирует еще два следующих друг за другом INTA-цикла. Во время этих циклов ПКП передает в МП полный адрес программы обслуживания прерывания, принятого к обработке: сначала младший, а затем старший байт адреса. Установленный в ISR бит остается в состоянии 1 до окончания процедуры обслуживания. В конце процедуры в ПКП

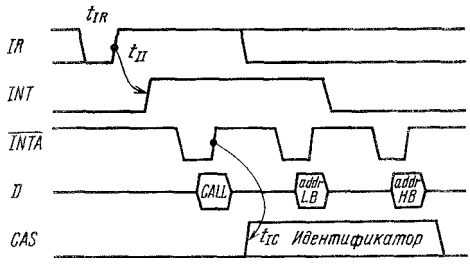


Рис. 3.43. Временные диаграммы работы ВН59

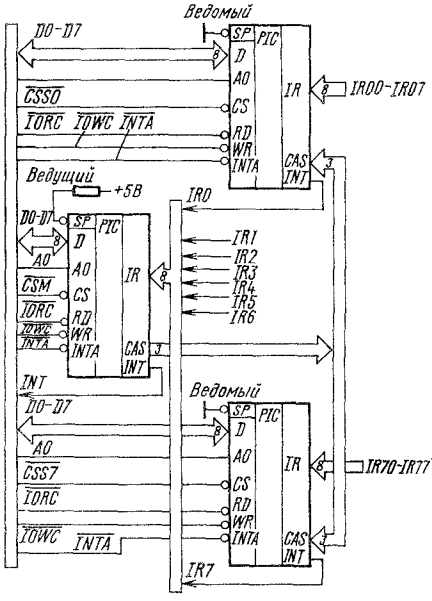


Рис. 3.44. Каскадное включение ВН59

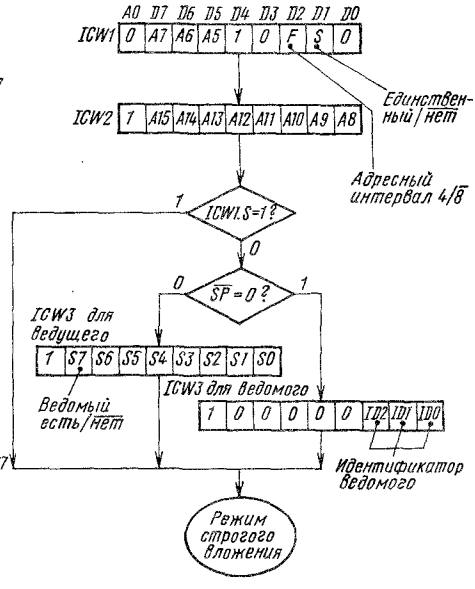


Рис. 3.45. Последовательность инициализации ВН59

должна быть передана специальная команда окончания прерывания EOI, которая сбрасывает соответствующий ISR-бит.

До тех пор, пока некоторый ISR-бит установлен, все запросы с равным или меньшим приоритетом игнорируются. В то же время запросы с более высоким приоритетом приводят к генерации сигнала INT, инициируя вложенные прерывания МП.

Каскадное соединение нескольких БИС типа ВН59 (рис. 3.44) позволяет довести число приоритетных уровней до 64. В данном случае одна БИС действует как ведущая ($\overline{SP}=1$), а другие как ведомые ($\overline{SP}=0$).

В каскадном режиме генерация кода команды CALL возлагается на ведущую БИС, а генерация адреса подпрограммы обслуживания — на ведомую. Для этого каждой ведомой БИС присваивается идентификационный код, соответствующий номеру линии запроса на прерывание ведущей БИС, к которой она подключена. Вместо генерации адреса ведущая БИС посылает по трехразрядной шине CAS

номер уровня, принятого к обслуживанию, на второй и третий стробы $\overline{\text{INTA}}$. Этот номер сравнивается компараторами ведомых БИС с присвоенными им идентификаторами, в результате выбирается один из них. Выбранная БИС завершает команду CALL, отвечая на второй и третий стробы $\overline{\text{INTA}}$ генерацией одного из восьми возможных адресов.

Отличительной чертой БИС типа ВН59 является ее программируемость. Программирование осуществляется двумя типами управляющих слов: командами инициализации ICW (Initialization Command Word) и управления OCW (Operation Command Word). Три команды инициализации ICW1—ICW3 загружаются перед началом работы и служат для установки БИС в исходное состояние. Команды управления OCW1—OCW3 могут быть переданы в ПКП в любое время после окончания инициализации. Они предназначены для оперативного управления работой контроллера. При каскадном включении каждая БИС программируется независимо от других.

При A0=0 и D4=1 входной байт интерпретируется как ICW1. При этом очищаются IRR, IMR и осуществляется последовательность действий, приведенная на рис. 3.45. Генерируемые ПКП восемь адресов в команде CALL (по адресу на каждый уровень) равно удалены друг от друга с интервалом либо 4, либо 8 байт, образуя в памяти МС таблицу входов в процедуры обслуживания прерываний размером 32 или 64 байт соответственно. Два первых обязательных слова ICW1 и ICW2 определяют базовый адрес таблицы входов. Адресный интервал таблицы программируется специальным битом F слова ICW1 (ICW1.F). При F=1 интервал равен 4, в противном случае—8. Адреса команды CALL формируются по схеме

$$addr = base + k \cdot NL,\quad k = 4 \text{ или } 8$$

и имеют формат, приведенный на рис. 3.46. Здесь через NL обозначен уровень запроса, принятого к обслуживанию. Программируемость базы таблицы входов base предполагает ее размещение в любом месте памяти МС с точностью до границы в 32 или 64 байт. При k=8 и base=0 таблица совпадает с системой входов по командам RSTn, n=0—7, микропроцессора ВМ80. При k=4 и base=4 обеспечивается совместимость со второй половиной таблицы входов по командам RSTn, n=4—7.5 микропроцессора ВМ85А. Кроме того, поддерживается перемещаемость таблицы входов, что может быть полезно в ряде практических случаев.

В составе ICW1 находится также бит S (Single), определяющий в каком режиме работает система прерываний: автономном (S=1) или каскадном (S=0). При работе в автономном режиме процедура инициализации завершается. В каскадном режиме в ПКП должна быть передана еще одна команда ICW3. Формат ICW3 ведущей БИС ($\overline{\text{SP}}=1$) отличается от формата ICW3 ведомого ($\overline{\text{SP}}=0$). В первом случае ICW3 сообщает контроллеру о множестве тех уровней, которые задействованы под каскадирование. Эти уровни выделяются установкой в 1 соответствующих разрядов ICW3. Во втором случае ICW3 используется для сообщения ведомой БИС идентификационного номера. Идентификатор должен соответствовать номеру уровня ведущей БИС, к которой подключена ведомая.

После инициализации микросхема готова к работе в режиме строго упорядоченных приоритетов. В данном режиме приоритеты уровней IR0—IR7 располагаются в порядке от 0 (высший) для IR0 до 7 (низший) для IR7. Дальнейшее управление работой схемы осуществляется с помощью OCW1—OCW3, форматы которых приведены на рис. 3.47.

Для установки регистра маски IMR применяется слово OCW1. С его помощью каждый уровень может быть замаскирован в произвольный момент времени независимо от других. При установке 1 в некотором разряде IMR запрещается

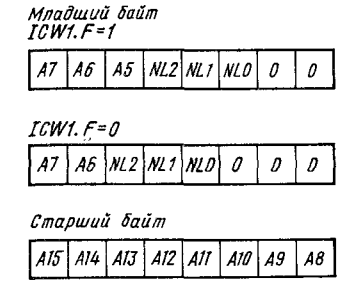


Рис. 3.46. Формат адреса команды CALL

прием прерываний по соответствующему уровню. Регистр маски воздействует и на IRR, и на ISR.

По окончании очередной процедуры обслуживания в контроллер должна быть передана команда EOI, которая сбрасывает ISR-бит с высшим приоритетом. Для этой цели используется OCW2 с кодом R=0, SL=0, EOI=1. При работе в каскадном режиме команда EOI выдается дважды: для ведущего и ведомого приборов.

В более общем случае OCW2 служит для установки в 0 произвольных разрядов ISR и циклического сдвига приоритетов с присвоением максимального значения любому из восьми возможных уровней. Если операция установки в 0 кодируется установкой в 1 бита OCW2.EOI, то операция циклического сдвига — бита OCW2.R (Rotate). В зависимости от состояния флажка SL (Special Level) возможны две ситуации: операция установки в 0 и (или) циклического сдвига относится к уровню с высшим в данный момент приоритетом из числа находящихся в обслуживании (SL=0); одна и (или) обе операции относятся к явно указываемому в OCW2 уровню (SL=1). В последнем случае номер уровня кодируется в трехразрядном поле L2—L0 (Level). Различные комбинации этих функций и соответствующие им коды приведены на рис. 3.47. На их основе можно построить достаточно мощные и гибкие системы прерываний с разнообразными характеристиками.

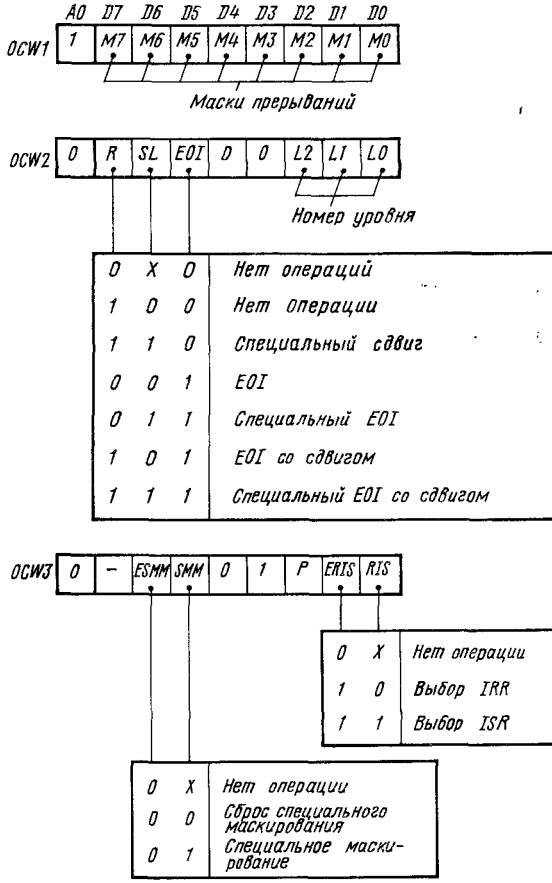


Рис. 3.47. Форматы OCW1—OCW3

Применение только команды EOI обеспечивает работу в режиме строго упорядоченных приоритетов, когда все запросы имеют фиксированные и строго упорядоченные приоритеты. Для обслуживания запросов с равными приоритетами должна быть использована команда EOI с циклическим сдвигом. При использовании команды сдвига одновременно со сбросом ISR-бита, имеющего высший текущий приоритет, реализуется циклический сдвиг приоритетов с присвоением

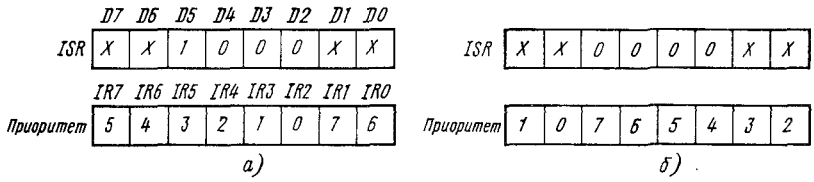


Рис. 3.48. Операция EOI со сдвигом:
а — до операции; б — после операции

низшего только что обслуженному уровню. На рис. 3.48 приведен пример циклического сдвига после обработки пятого уровня с приоритетом 3. Следует отметить, что циклический сдвиг не нарушает последовательности вложенных друг в друга прерываний, что обеспечивает правильный возврат из обслуживающих их подпрограмм.

В рассмотренных случаях в качестве сбрасываемого в регистре ISR-бита и (или) уровня с низшим приоритетом после циклического сдвига выступал уровень с высшим приоритетом из числа обслуживаемых в данное время, который однозначно определяется состоянием ISR и текущим распределением приоритетов. Прямая адресация уровня обеспечивает: сброс конкретного ISR-бита и, следовательно, завершение процедуры обслуживания соответствующего уровня; циклическое изменение приоритетов с явным указанием нижнего уровня системы приоритетов; сброс адресуемого ISR-бита и присвоение ему низшего приоритета за счет их циклического сдвига. Эти операции позволяют построить системы прерываний с разнообразными (в том числе динамически изменяемыми) структурами приоритетов.

В рассмотренной системе приоритетов находящийся на обслуживании уровень подавляет обработку уровней с меньшим приоритетом даже несмотря на его временное маскирование. Обработку прерываний более низкого уровня можно разрешить сбросом соответствующего ISR-бита, однако этот процесс необратим. Еще одна возможность состоит в установке с помощью OCW3 режима специального маскирования (см. рис. 3.47). В данном режиме каждый бит в регистре ISR запрещает только собственный уровень, но разрешает все остальные. Во время процедуры инициализации режим специального маскирования автоматически аннулируется.

Кроме управления режимом специального маскирования в функции OCW3 входит управление режимом поллинга и выбор регистра IRR или ISR для чтения его содержимого с помощью программ. Режим поллинга (опроса) предполагает ввод информации об источнике прерывания программным методом. Режим инициируется выдачей в ПКП слова OCW3 с установленным битом P (Polling). Контроллер ВН59 интерпретирует следующий цикл чтения при A0=0 как подтверждение прерывания и выдает на шину данных информацию о требующем обслуживания уровне с высшим приоритетом в формате, представленном на рис. 3.49. Если к моменту опроса запросов нет, то возвращается слово с флажком запроса I=0, в противном случае I=1, а в поле W2—W0 содержится код уровня с высшим приоритетом. В остальном работа ВН59 аналогична предыдущему случаю. Режим поллинга применяется в случаях, когда существует единая

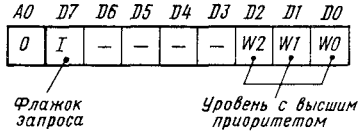


Рис. 3.49. Формат данных при опросе

программа обслуживания для нескольких запросов, а также для расширения числа уровней (сверх 64).

Следует отметить, что ведущие и ведомые контроллеры программируются независимо друг от друга. Поэтому они могут работать в различных режимах, благодаря чему расширяются возможности создания эффективных систем обработки прерываний.

Входящие в состав ВН59 регистры IMR, IRR и ISR доступны для чтения. Регистр маски IMR может быть прочитан в любой момент при A0=0 по команде $\overline{RD}$, регистры IRR и ISR — при A0=1. В каждый момент времени непосредственно доступен только один из регистров. Выбор регистра осуществляется словом OCW3 с помощью ERIS и RIS в соответствии с таблицей на рис. 3.47. После выбора содержимое регистра может считываться произвольное число раз до выбора другого. При приеме ICW1 устанавливается режим чтения содержимого IRR.

3.7. Средства счета времени

Программируемый интервальный таймер ВИ53/ВИ54. Среди периферийных БИС программируемый интервальный таймер (ПИТ, PIT — Programmable Interval Timer) КР580ВИ53 (ВИ53) занимает особое место. Это функционально законченное однокристальное ПУ встраиваемого типа, предназначенное для работы совместно с МП ВМ80. Интервальный таймер ВИ53 решает одну из наиболее общих проблем любой МС — генерацию точных временных интервалов под программным контролем. Микросхема выполнена по n-МОП-технологии в 24-выводном корпусе типа 2120.24 и имеет единственный источник питания +5 В.

В состав БИС (рис. 3.50) входят три 16-разрядных вычитающих счетчика (СТ) с частотой счета по входу CLK (Clock) до 2 МГц. Каждый СТ может работать в одном из шести программно-заданных режимов независимо от других. Все счетчики программно доступны для записи и чтения с помощью слов данных DW и могут работать как в двоичном коде, так и в 2/10-коде. Управление режимами выполняется с помощью управляющих слов CW (рис. 3.51), которые кроме режима (поле M) определяют код счета (двоичный или 2/10) и формат обмена данными с МП при операциях со счетчиками: только старшим байтом, только младшим байтом или всем словом (поле RL). Поле SC используется для указания счетчика, к которому относится очередное слово CW.

Связь ПИТ с МС осуществляется через двунаправленную 8-разрядную шину данных D7—D0 под управлением пяти сигналов A0, A1, $\overline{CS}$, $\overline{RD}$ и $\overline{WR}$ в соответствии с табл. 3.6. При двухбайтовом формате данных операция со счетчиками выполняется дважды: сначала записывается или считывается младший байт, затем — старший. Обслуживание СТ выполняется параллельно и независимо друг от друга. При подаче питания их состояния и режим работы оказываются неопределенными. Поэтому перед началом работы каждый СТ должен быть инициализирован индивидуально посылкой соответствующего слова состояния CW. Каждое CW, за исключением операции защелкивания (RL=0), сопровождается 1-2 байтами слова данных DW начального состояния выбранного СТ. Выполнение операции начинается только после загрузки последнего байта данных. Последовательность инициализации и переинициализации счетчиков произвольна. Допускается приостановка последовательности инициализации любого СТ на неопределенное время с последующим ее продолжением или началом новой.

Режим 0 (прерывание по окончанию счета). По окончании записи CW на

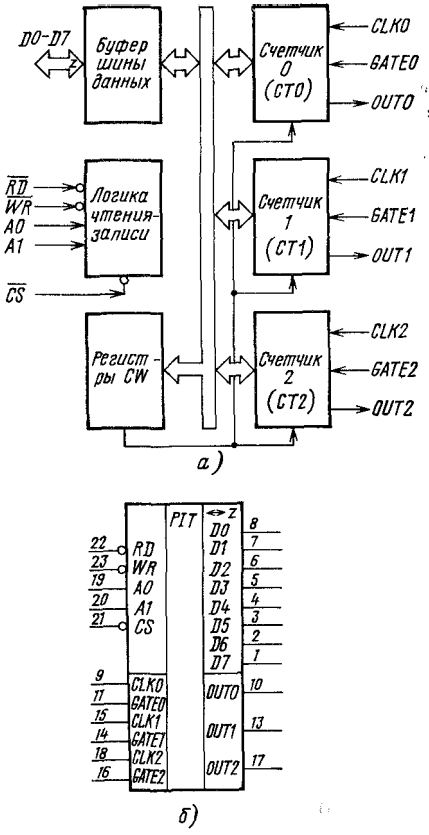


Рис. 3.50. Программируемый интервальный таймер ВИ53:

а — структурная схема; б — условное графическое обозначение

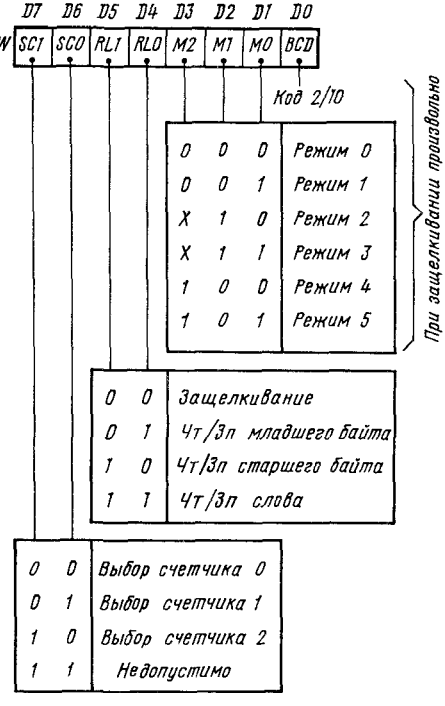


Рис. 3.51. Формат управляющего слова ВИ53

выходе OUT устанавливается 0 (рис. 3.52, а). После загрузки DW счетчик начинает вычитать по каждому срезу CLK. При переходе в 0 на выходе OUT устанавливается 1. Работа СТ при этом не останавливается. Перезапуск канала производится при загрузке новых данных DW. Запись первого байта останавливает

Таблица 3.6

A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Операция	A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Операция
0	0	0	1	0	D ← Счетчик 0	0	1	1	0	0	Счетчик 1 ← D
0	1	0	1	0	D ← Счетчик 1	1	0	1	0	0	Счетчик 2 ← D
1	0	0	1	0	D ← Счетчик 2	1	1	1	0	0	Управление ← D
1	1	0	1	0	Нет операции	X	X	1	1	0	Нет операции
0	0	1	0	0	Счетчик 0 ← D	X	X	X	X	1	Нет операции

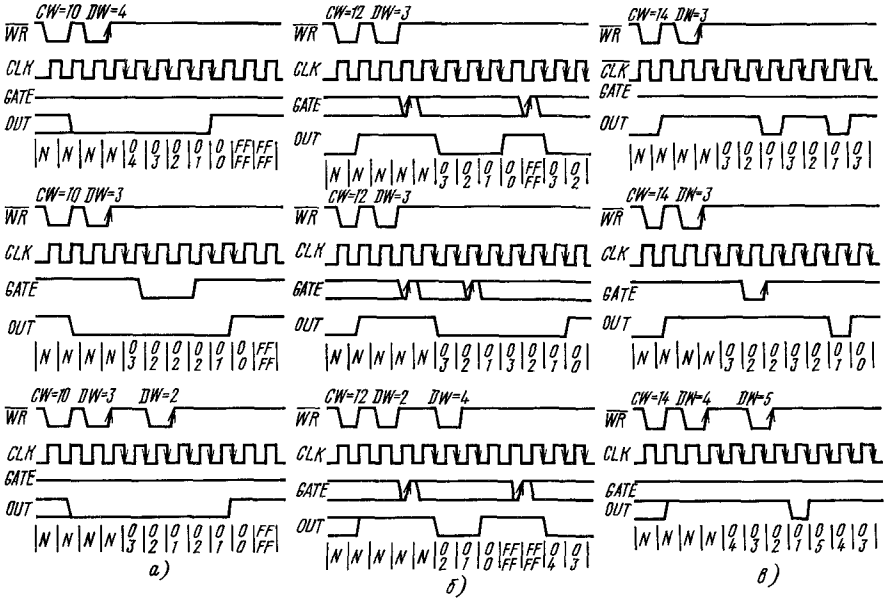


Рис. 3.52. Временные диаграммы работы программируемого интервального таймера:
а — режим 0, б - режим 1, в - режим 2

счет, второй байт запускает новый счет. Вход GATE разрешает счет при высоком и запрещает при низком уровне напряжения.

Режим 1 (программируемый одновибратор). Выход OUT генерирует 0 по первому срезу CLK после фронта GATE (рис. 3.52, б) и счетчик начинает считать. При переходе CT в 0 на выходе OUT устанавливается 1. Перезагрузка CT во время счета не изменяет длительности текущего импульса. Однако появление нового фронта GATE перезапускает CT с новым или старым значением. Чтение счетчика возможно в любое время.

Режим 2 (генератор частоты). Выход OUT=0 только в течение одного периода входной частоты CLK (рис. 3.52, в), который определяется значением DW. Перезагрузка CT не приводит к изменению длительности текущего периода, но влияет на длительность последующего. При подаче на управляющий вход GATE 0 осуществляется переход в 1 выхода OUT. Фронт GATE запускает CT из начального состояния. Может служить для аппаратной синхронизации счетчика.

Режим 3 (генератор прямоугольных импульсов). На выходе OUT 1 будет сохраняться до тех пор, пока не закончится одна половина счета (рис. 3.53, а). При нечетном DW на протяжении $(N+1)/2$ тактов удерживается 1 и на протяжении $(N-1)/2$ тактов — 0. При перезагрузке CT новое значение скажется на результат работы только при переходе OUT в другое состояние. В остальном режим подобен предыдущему.

Режим 4 (программная задержка строба). После записи CW на выходе OUT устанавливается 1 (рис. 3.53, б). Запуск счета осуществляется после загрузки DW. При достижении 0 на выходе OUT генерируется импульс длительностью в один период CLK, а CT продолжает работать. Перезагрузка CT во время счета приводит к перезапуску CT. Генерация 0 на входе GATE приостанавливает счет.

Таблица 3.7

Режим	Переход в 0	Переход в 1	Напряжение высокого уровня (1)
0	Запрет счета	Нет операции	Разрешение счета
1	Нет операции	Запуск с переходом OUT в 0	Нет операции
2, 3	Запрет счета с установкой OUT в 1	Запуск счета	Разрешение счета
4	Запрет счета	Нет операции	Разрешение счета
5	Нет операции	Запуск счета	Нет операции

Режим 5 (аппаратная задержка строба). Счетчик начинает работать только по фронту GATE (рис. 3.53, в). Новый фронт перезапускает текущий счет. В остальном режим подобен предыдущему.

Во всех режимах сигнал GATE (рис. 3.54) является управляющим: запрещает счет низким уровнем напряжения и (или) перезапускает фронтом. Функции входа GATE приведены в табл. 3.7.

В ряде случаев необходимо контролировать текущее состояние СТ, например, когда СТ используется в качестве счетчика событий или реальных часов. Существуют два метода чтения содержимого СТ. Первый метод состоит в обычном чтении 1—2 байтов выбранного СТ. Для устойчивого чтения текущего состояния СТ функция счета может быть подавлена с помощью входа GATE или

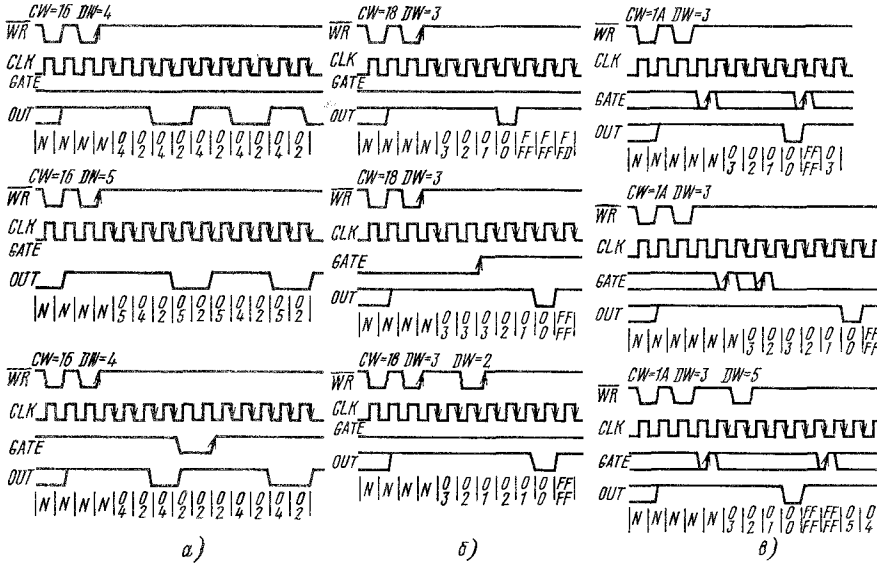


Рис. 3.53. Временные диаграммы работы программируемого интервального таймера:
а – режим 3, б – режим 4, в – режим 5

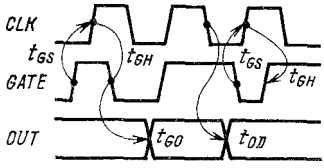


Рис. 3.54. Временные диаграммы периферийных сигналов программируемого интервального таймера

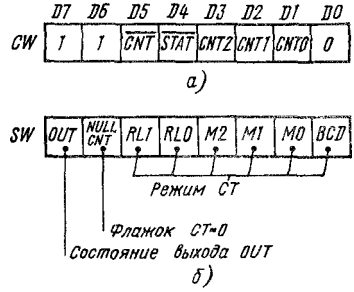


Рис. 3.55. Новые форматы слов ВИ54:
а — управляющее слово; б — слово состояния

внешним запретом импульсов CLK. Сначала читается младший, затем старший байт. Если формат данных обмена с СТ — слово, то оба байта должны быть сосчитаны до подачи на данный СТ новой команды.

Второй метод заключается в чтении содержимого СТ «на ходу» — без запрета его работы. Для этого в ПИТ должна быть послана специальная команда защелкивания (поле RL = 0), при которой поле выбора счетчика SC кодирует выбор СТ. Остальные разряды CW могут быть произвольными. По команде защелкивания текущее состояние выбранного СТ записывается в специальный регистр, что не мешает работе СТ. Следующая за командой операция чтения приводит регистр в исходное состояние. Операция чтения подвержена тем же ограничениям, что и в предыдущем случае.

Программируемый интервальный таймер ВИ54 является усовершенствованным архитектурно совместимым с ВИ53 прибором. В устройстве предусмотрена новая команда для чтения текущего состояния счетчиков, включая режим. Форматы команды и нового слова состояния SW приведены на рис. 3.55.

Организация общесистемных средств счета времени. Программируемый интервальный таймер ВИ53/ВИ54 является прибором широкого назначения. На его основе могут быть построены разнообразные времязависимые устройства, имеющие общесистемное значение: генераторы скорости передачи последовательных данных, часы суточного времени, средства контроля за длительностью обращения к системной магистрали и др. Учитывая широкий интерес к средствам такого типа, рассмотрим примеры их конкретной реализации для МС на базе ВМ80.

Генератор скорости. Генератор скорости передачи данных через последовательные каналы может быть построен на базе одного из счетчиков ПИТ. Например, для приведенного на рис. 3.35 адаптера ИРПС используется счетчик СТ2, который должен быть запрограммирован для работы в режиме генератора прямоугольных импульсов. На вход GATE счетчика необходимо подать напряжение высокого уровня, а вход CLK соединить с линией системной частоты CCLK. При перезапуске системы (включении питания или нажатии клавиши RESET) генератор скорости должен быть проинициализирован. Это делается с

помощью специальной процедуры инициализации VELINI, переводящий CT2 в режим 3 с обменом словами в двоичном коде:

```
VELINI:   PUSH  PSW
          MVI   A, 10110110B     ;Код команды
          OUT   PIT+3
          POP   PSW
          RET
```

Процедуру установки скорости передачи удобно оформлять в виде отдельной подпрограммы VEL, принимающей через регистровую пару BC 16-разрядный фактор скорости в двоичной форме:

```
VEL:      PUSH    PSW
          MOV     A,C       ;Младший байт фактора
          OUT     PIT+2     ;скорости
          MOV     A, B      ;Старший байт фактора
          OUT     PIT+2     ;скорости
          POP     PSW
          RET
```

С учетом дополнительного деления частоты на K внутренними средствами ПСА параметр x для подпрограммы VEL рассчитывается по формуле: x=CCLK/KV, где V—скорость передачи информации. Значения переменной x для стандартных скоростей передачи при CCLK=2 МГц и K=16 приведено в табл. 3.8.

Очевидно, что с помощью данной схемы скорость передачи 19 200 Гц с достаточной точностью не может быть получена. Для решения данной задачи необходим специальный подбор значений частоты CCLK, которая для BM80 связана с частотой выхода Ф2TTL ГТИ. Если использовать CCLK = 1,9384 МГц, то можно получить все стандартные частоты. Для этого потребуется кристалл с частотой OCS = 17,4456 МГц и параметр x, который представлен выше в скобках.

Часы суточного времени. Если функцию счета времени возложить на программные средства, то для реализации часов достаточно построить генератор меток реального времени. Для этой цели резервируется счетчик CT0 программируемого таймера,

Таблица 3.8

Требуемая скорость V, Гц	Параметр x	Действительная скорость V, Гц	Отклонение от стандарта, Гц
1 200	104 (112)	1 202	+2
2 400	52 (56)	2 404	+4
4 800	26 (28)	4 808	+8
9 600	13 (14)	9 616	+16
19 200	7 (7)	17 857	−1343

вход CLK которого соединяется с CCLK, на вход GATE подается напряжение высокого уровня, а выход OUT используется в качестве сигнала запроса на прерывание высшего приоритета IR0. Такая схема обеспечивает своевременную обработку меток реального времени через систему прерываний.

Контроллер прерываний ВН59 фиксирует запрос на прерывание по фронту с предварительным удержанием низкого уровня напряжения на входе не менее 120 нс. Поэтому СТ0 можно запрограммировать для работы как в режиме 2, так и в режиме 3. Остановимся на режиме 2:

```
        MVI     A,00110100B     ;Код команды
        OUT     PIT+3           ;для ПИТ
```

Процедура установки частоты следования меток реального времени подобна VEL:

```
CLK:    PUSH    PSW
        MOV     A,C             ;Младший байт
        OUT     PIT
        MOV     A,B             ;Старший байт
        OUT     PIT
        POP     PSW
        RET
```

Обычно эти метки следуют с частотой 50 Гц. Предполагая, что CCLK = 2 МГц, получаем стандартную процедуру инициализации генератора системного времени:

```
CLKINI: PUSH    PSW
        MVI     A,00110100B     ;Код команды
        OUT     PIT+3           ;для ПИТ
        LXI     B,40000         ;Слово данных ПИТ
        CALL    CLK             ;Вывод слова данных
        POP     PSW
        RET
```

Скорость счета может быть изменена в любой момент вызовом подпрограммы CLK.

Счет суточного времени может быть организован специальной программой [11], запускаемой по прерыванию IR0. Подпрограмма ведет счет в формате «часы, минуты, секунды, доли секунд» и использует 8 байт данных:

```
MODUL:  DB      50              ;Число системных ме-
                                ;ток
                                ;в секунде
        DB      60              ;Число секунд в ми-
                                ;нуте
        DB      60              ;Число минут в часе
```

```
        DB      24          ;Число часов в сутках
TIME:   DS      1           ;Доли секунд
        DS      1           ;Секунды
        DS      1           ;Минуты
        DS      1           ;Часы
```

Предполагается, что метки следуют с частотой 50 Гц:

```
CLOCK:  PUSH    PSW         ;Сохранить
        PUSH    B           ;состояние регистров
                            ;МП
        PUSH    D
        PUSH    H
        LXI     H,MODUL     ;Установить началь-
                            ;ные
        LXI     D,TIME      ;значения указателей
        MVI     B,4         ;Установить счетчик
                            ;циклов
NEXT:   LDAX    D           ;Выбрать текущее
                            ;значение
                            ;времени
        INR     A           ;Увеличить его на 1
        CMP     M           ;Сравнить с макси-
                            ;мальным
        JNZ     OK          ;Конец счета, если
                            ;меньше
        XRA     A           ;Иначе—
        STAX    D           ;сброс ячейки памяти
        INX     H           ;Модифицировать
        INX     D           ;указатели
        DCR     B           ;Уменьшить значение
                            ;счетчика
                            ;циклов на 1
        JNZ     NEXT        ;Переход, если цикл
                            ;не последний
OK:     STAX    D           ;Сохранить модифи-
                            ;цированное значение
                            ;текущего времени
        POP     H           ;Восстановить
        POP     D           ;состояние регистров
                            ;МП
        POP     B
        CALL    EOI         ;Конец прерывания
        POP     PSW
        EI                  ;Разрешить прерыва-
                            ;ние
        RET
```

Для получения текущего времени хранения в четырех соседних байтах с базой TIME необходимо просто прочитать их содержимое, например, с помощью процедуры:

```
TGET:   DI                      ;Запретить прерывания
        LDA     TIME+1          ;Секунды
        LHLD    TIME+2          ;Часы и минуты
        EI                      ;Разрешить прерывания
        RET
```

Программа возвращает в регистре A секунды, в L минуты, в H часы. Установка времени выполняется обратной процедурой, при этом доли секунд игнорируются:

```
TSET:   DI                      ;Запретить прерывания
        STA     TIME+1          ;Секунды
        SHLD    TIME+2          ;Часы и минуты
        EI                      ;Разрешить прерывания
        RET
```

Подпрограммы используются при запрещенных прерываниях, что обеспечивает монопольный доступ к разделяемому двумя параллельными процессами CLOCK и TGET/TSET блоку данных TIME. В противном случае могут быть ошибки, связанные с одновременностью доступа.

Временные затраты, необходимые для поддержки рассмотренных выше часов суточного времени при тактовой частоте BM80, равной 2 МГц, и частоте системных меток 50 Гц составляют 1,2%. В тех случаях, когда они не удовлетворяют пользователя, их можно снизить, за счет уменьшения частоты генерации системных меток, отказа от счета суточного времени с преобразованием в часы и перехода к счету в периодах системных меток, расширения аппаратурных средств поддержки часов или перехода к их полной аппаратной реализации.

Служба реального времени. Суточные часы являются основой для построения службы реального времени, в функции которой входит управление последовательностью заданий в строгом соответствии со временем. Обычно запросы на выполнение этих заданий упорядочиваются согласно времени начала их обслуживания. Время начала обслуживания является параметром запроса. Служба запускается во время обслуживания часов суточного времени. Она сверяет текущее время с временем начала обслуживания и при достижении последнего осуществляет запуск процедуры обслуживания. Сам запрос при этом из очереди удаляется.

Такая чисто программная организация службы времени имеет один недостаток — требует определенных затрат процессорного времени. А чем чаще работают часы суточного времени и реже моменты обслуживания запросов, тем менее эффективной оказывается служба времени программного уровня.

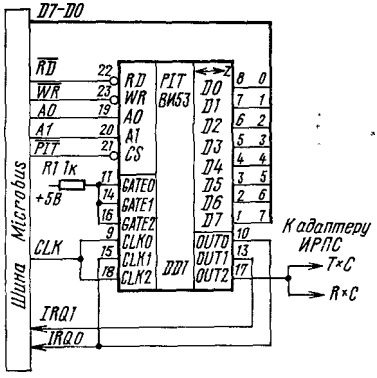


Рис. 3.56. Схема системного таймера

Решить проблему организации службы времени помогает устройство счета временны́х интервалов, в качестве которого применяется счетчик CT1 программируемого таймера. Для этого вход CLK счетчика соединяется с выходом OUT генератора часов суточного времени, на вход GATE подается напряжение высокого уровня, а выход OUT используется в качестве запроса IRQ1 первого уровня. Запросам на обслуживание от счетчика временны́х интервалов отведен более низкий приоритет по сравнению с запросами от генератора суточного времени, так как они следуют намного реже. При такой организации служба времени представляет собой системную процедуру, которая активизируется по прерываниям первого уровня. В функции процедуры входит запуск обслуживания очередного запроса из очереди и установка нового интервала ожидания до начала обслуживания следующего запроса. Отработка данного интервала возлагается на счетчик паузы CT1.

Счетчик CT1 программируется для работы в режиме 0 с помощью процедуры

```
DLINI:  PUSH   PSW
        MVI    A,01110000B   ;Команда ПИТ
        OUT    PIT+3
        POP    PSW
        RET
```

После ее выполнения на выходе OUT счетчика установится 0. Устройство запускается для отработки паузы, передаваемой через регистровую пару BC, следующим образом:

```
DELAY:  PUSH   PSW
        MOV    A,C           ;Младший байт зна-
                             ;чения паузы
        OUT    PIT+1
        MOV    A,B           ;Старший байт значе-
                             ;ния паузы
        OUT    PIT+1
        POP    PSW
        RET
```

Пауза передается в виде двоичного целого с периодом меток суточного времени. После отработки паузы счетчик CT1 свою

работу не приостановит. Для его остановки можно воспользоваться процедурой DLINI.

На рис. 3.56 представлена схема устройства на базе ПИТ, три счетчика которого зарезервированы для выполнения трех рассмотренных функций общесистемного значения. Это устройство, которое называется системным таймером, применяется в большинстве МС. Набор из шести процедур нижнего уровня (CLKINI, CLK, DLINI, DELAY, VELINI и VEL) образует драйвер системного таймера, который освобождает программиста от прямого управления работой устройства и предоставляет в его распоряжение стандартный интерфейс программного уровня. Такие компоненты, как часы суточного времени или служба реального времени, надстраиваются над драйвером, образуя второй слой ПО более высокого уровня.

ГЛАВА 4.

ОРГАНИЗАЦИЯ ОДНОКРИСТАЛЬНЫХ МИКРОКОНТРОЛЛЕРОВ

4.1. Вводные замечания

Отдельный класс МС представляют однокристальные микроЭВМ. Интеграция всех составных частей МС (ЦП, памяти, подсистемы ВВ, средств поддержки режима реального времени) привела к ряду ограничений на принципы ее организации, потребовала развития архитектуры в направлении, не свойственном для многокристальных компоновок.

Организация однокристальных микроЭВМ ориентирована на применение встраиваемых в изделие недорогих управляющих МС реального времени, рабочая программа которых расположена в ПЗУ системы. По этой причине находящаяся на кристалле физическая память микроЭВМ делится на постоянную для записи программ и оперативную для хранения различных переменных, а сами приборы называются однокристальными микроконтроллерами. Такому делению физической памяти способствовали и технологические ограничения, свойственные системам на одном кристалле.

Развитие интегральной технологии и расширение области применения однокристальных МК привели к дальнейшему совершенствованию архитектурных и структурных принципов их организации. Современные однокристальные МС обладают такими вычислительными ресурсами и возможностями управления в режиме реального времени, для получения которых раньше необходимы были более дорогие многокристальные компоновки. Применение МК на одном кристалле особенно эффективно в системах, где наряду с небольшой памятью требуются интенсивно используемые средства ВВ в реальном масштабе времени.

Периодом становления архитектуры 8-разрядных однокристальных МК считают 1977—1979 гг., когда появились первые приборы этого класса: 8048 фирмы Intel, 3870 фирмы Mostek и 9940 фирмы Texas Instruments Inc. Приборы 3870, 9940 были программно совместимы с многокристальными системами и во многом

Таблица 4.1. **Состав однокристальных микроконтроллеров фирмы Intel**

Тип прибора	Память программ, байт		Память данных, байт		Число линий ВВ		ИРПС	Число и разрядность таймеров
	внутренняя	внешняя	внутренняя	внешняя	внутренних	внешних		
			iMCS-48					
8048	1К (ПЗУ)	4К	64	256	27	16	—	1×8
8748	1К (УСППЗУ)	4К	64	256	27	16	—	1×8
8035	—	4К	64	256	27	16	—	1×8
8049	2К (ПЗУ)	4К	128	256	27	16	—	1×8
8749	2К (УСППЗУ)	4К	128	256	27	16	—	1×8
8039	—	4К	128	256	27	16	—	1×8
8050	4К (ПЗУ)	—	256	256	27	16	—	1×8
8040	—	4К	256	256	27	16	—	1×8
8021	1К (ПЗУ)	—	64	—	21	16	—	1×8
8022*	2К (ПЗУ)	—	64	—	26	16	—	1×8
8041	1К (ПЗУ)	—	64	—	18+СА	16	—	1×8
8741	1К (УСППЗУ)	—	64	—	18+СА	16	—	1×8
8355	—	2К (ПЗУ)	—	—	—	16	—	—
8755	—	2К (УСППЗУ)	—	—	—	16	—	—
8155	—	—	—	256	—	22	—	1×8
8156	—	—	—	256	—	22	—	1×8
8243	—	—	—	—	—	16	—	—
			iMCS-51					
8051	4К (ПЗУ)	64К	128	64К	32	—	1	2×16
8751	4К (УСППЗУ)	64К	128	64К	32	—	1	2×16
8031	—	64К	128	64К	32	—	1	2×16
8052	8К (ПЗУ)	64К	256	64К	32	—	1	3×16
8032	—	64К	256	64К	32	—	1	3×16
8044	4К (ПЗУ)	64К	256	64К	32	—	1	2×16
8744	4К (УСППЗУ)	64К	256	64К	32	—	1	2×16
8344	—	64К	256	64К	32	—	1	2×16
			iMCS-96					
8394, 8396	8К (ПЗУ)	64К	232	64К	40+СВВ	—	1	2×16
8094, 8096		64К	232	64К	40+СВВ	—	1	2×16
8395*, 8397*	8К (ПЗУ)	64К	232	64К	40+СВВ	—	1	2×16
8095*, 8097*	—	64К	232	64К	40+СВВ	—	1	2×16

* Содержит АЦП.

дублировали их архитектурные признаки, МК 8048 имели оригинальную организацию.

В течение четырех лет, начиная с 1976 г., фирмой Intel было разработано семейство 8-разрядных однокристальных МК iMCS-48 [41], получившее широкое распространение, в основе которого лежит МК 8048 (табл. 4.1). В составе семейства 12 микроЭВМ с единой базовой архитектурой, но различными функциональными возможностями, реализованными непосредственно на кристалле. Семейство включает также ряд расширителей, согласованных с базовой архитектурой микроЭВМ и содержащих те части памяти программ и данных, а также средств ВВ, которые не включены в состав основного кристалла.

Базовая вычислительная среда iMCS-48 включает до 4К байт программного ПЗУ, формируемого как внутренними, так и внешними по отношению к микроЭВМ средствами, 64/128/256 байт внутренней и 256 байт внешней памяти данных, до 27 внутренних и 16 внешних линий ВВ, 8-разрядный таймер/счетчик, а также одноуровневую систему прерываний с двумя источниками запросов. Микроконтроллер 8021 является младшей моделью семейства, в которой отсутствуют средства подключения внешней памяти и система прерываний. На его основе создан простой прибор 8022 [31], в котором предусмотрен 8-разрядный аналого-цифровой преобразователь (АЦП) с двумя коммутируемыми аналоговыми входами. В отличие от остальных БИС 8041 [62] имеет встроенный системный адаптер (СА) для его подключения к шине более мощных систем в качестве программно-управляемого контроллера. Становление архитектуры МК было завершено к 1980 г., когда весь набор однокристальных МК и расширителей к ним был полностью освоен промышленностью. Сегодня элементы семейства iMCS-48 рассматриваются как стандартные компоненты для проектирования микропроцессорных средств и систем.

В 1980 г. фирмой Intel было разработано новое семейство однокристальных МК iMCS-51 [32, 57], базовым представителем которого является прибор 8051 (см. табл. 4.1). Новое семейство обеспечивает совместимость с архитектурой iMCS-48, но обладает более обширными адресными пространствами памяти программ и данных, усовершенствованными средствами ВВ и поддержки режима реального времени. В архитектуре предусмотрено до 64К байт ПЗУ, часть которого реализуется на кристалле, 128/256 байт внутреннего ОЗУ, до 64К байт внешнего ОЗУ, 32 линии физического ВВ, программируемый последовательный интерфейс, два или три 16-разрядных таймера/счетчика и двухуровневая система прерываний с пятью или шестью источниками запросов. Дальнейшее развитие получила система команд и способы доступа к отдельным элементам данных. В состав системы введены команды умножения и деления, реализован булев подпроцессор. Сейчас в семействе iMCS-51 более восьми однокристальных микроЭВМ с различными физическими возможностями. Развитие семейства продолжается.

В 1983 г., когда появилась возможность интеграции на одном кристалле кремния более чем 100 тыс. транзисторов, фирмой Intel было предложено семейство 16-разрядных однокристальных МК iMCS-96 [57]. В основе семейства лежит БИС 8096 (см. табл. 4.1), содержащая 120 тыс. транзисторов на одном кристалле, что позволило разместить на нем 16-разрядный ЦП, 8К байт программной памяти, 232 байт памяти данных, а также подсистему аналогового и цифрового ВВ с развитыми средствами поддержки режима реального времени, включая скоростной ВВ (СВВ). Практическое освоение перспективных 16-разрядных МК семейства iMCS-96, ориентированных на применение в 90-х гг., находится в начальной стадии. Областью их использования будут сложные управляющие устройства с повышенными арифметическими возможностями.

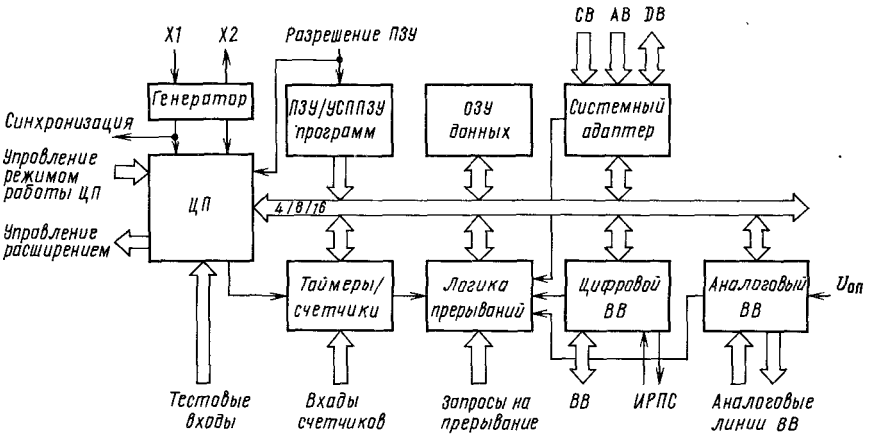


Рис. 4.1. Обобщенная схема однокристального микроконтроллера

Микроконтроллерные БИС (рис. 4.1) отличаются друг от друга, но общим для них является 8- или 16-разрядный ЦП, ПЗУ емкостью (1—8) К байт, ОЗУ емкостью 64—256 байт, значительное число линий цифрового ВВ (18—32). Все системы имеют достаточно эффективные наборы команд, содержащие до 70 и более различных кодов, в том числе мощные средства организации вычислений в режиме реального времени.

Имеется два типа памяти программ МК, обеспечивающих гибкость при переходе от проекта к промышленному изделию. Микроконтроллеры с ультрафиолетом стираемыми программируемыми ПЗУ (УСППЗУ) очень экономичны при разработке и отладке исходной системы. Их память команд может быть запрограммирована с помощью обычных программирующих систем [15]. При необходимости память можно полностью очистить засветкой ультрафиолетом через прозрачное окошко на верхней крышке корпуса БИС и ввести в нее новую программу.

4.2. Базовая организация BE48

Среди однокристальных МК отечественного производства следует отметить БИС К1816ВЕ48(ВЕ48) — базовый прибор МПК. Пока в составе комплекта пять элементов [17, 21]: К1816ВЕ48, К1816ВЕ49, К1816ВЕ48, К1816ВЕ35, К1816ВЕ39, отличающиеся функциональными возможностями памяти программ. В К1816ВЕ48/ВЕ49 используется маской программируемое ПЗУ, в К1816ВЕ48 — УСППЗУ, а К1816ВЕ35/ВЕ39 представляет модель без ПЗУ.

Семейство ВЕ48 является аналогом семейства iMCS-48. В основу его архитектуры положена организация гарвардского типа, ориентированная на интенсивное использование двух банков рабочих регистров и операций ВВ.

Представленная на рис. 4.2 структурная схема однокристальной вычислительной системы ВЕ48/ВЕ49/ВЕ50 содержит 8-разрядный

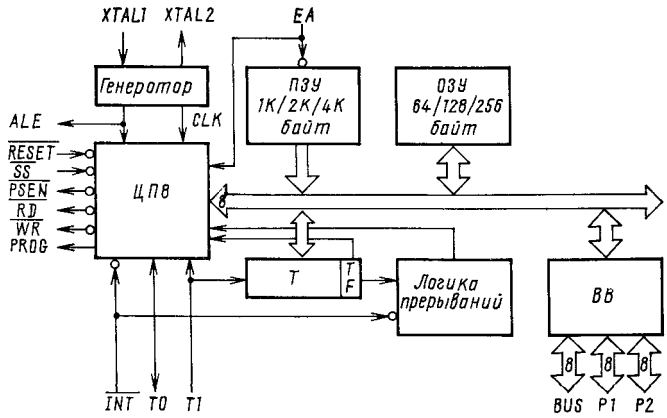


Рис. 4.2. Схема однокристальной вычислительной системы BE48/BE49/BE50

ЦП, (ЦП48), управляющее ПЗУ, внутреннее ОЗУ 24 линии прямого ВВ, три тестируемых входа, 8-разрядный таймер/счетчик Т и логику одноуровневой системы прерываний с двумя источниками запросов. Благодаря предусмотренным в схеме средствам внешнего расширения возможны вынос за пределы кристалла и (или) прямое увеличение размеров управляющей памяти до 4К байт, добавление дополнительного блока внешнего ОЗУ данных в 256 байт и 16 линий ВВ. Полученная в итоге система называется расширенной. При использовании в дальнейшем стандартной техники переключения банков внешние средства могут быть доведены до требуемого объема сверх норм, предусмотренных расширенной организацией.

Гарвардский принцип организации вычислительной среды предусматривает разделение памяти для хранения программ и данных. Управляющая память допускает только операцию считывания, память данных доступна и для записи, и для считывания.

Изоляция пространств памяти программ и данных позволила минимизировать длину машинных команд, что было очень важным в первых МК с ограниченными объемами ПЗУ. Как показывает практика, объем программного обеспечения МК обычно намного превышает объем памяти, отводимой под данные. Если бы в МК использовалась свойственная многокристальным микроЭВМ архитектура неймановского типа, то большая часть совмещенной памяти программ и данных была бы занята кодами команд. Вместе с тем основные механизмы адресации связаны исключительно с памятью данных. Их потенциальная ориентация на большой объем совмещенной памяти и реальное применение в рамках небольшой свободной от программ области привели бы к неэффективному использованию адресной

части команд и, как следствие, к увеличению объемов программ и снижению скорости их исполнения.

Память данных разбита на две полностью изолированные друг от друга 8-разрядные линейные области с различными способами доступа к ним. Внутренняя память является областью интенсивного обращения и служит только для хранения данных, внешняя — дополнительным расширением пространства данных и может быть с успехом использована для ВВ с отображением в память. Существует также возможность физического совмещения внешней памяти данных с частью управляющей памяти для организации единой области программ и данных, доступной как для операции чтения, так и записи.

Изолированное от памяти данных пространство ВВ представлено в виде трех встроенных 8-разрядных и четырех внешних 4-разрядных двунаправленных портов, причем каждый порт адресуется независимо от других. Имеются также три отдельные входные линии, состояние которых может быть проверено специальными командами условного перехода. Средства поддержки режима реального времени представлены двухрежимным 8-разрядным таймером/счетчиком и одноуровневой системой прерываний с двумя источниками запросов. Расширение специальных аппаратных функций может быть выполнено с помощью стандартных периферийных БИС из семейства ВМ80/ВМ85А, области ВВ которых отображаются на внешнюю память данных ВЕ48.

4.3. Набор регистров ВЕ48

Набор доступных программисту регистров ЦП ВЕ48 представлен на рис. 4.3. В состав набора входит 8-разрядный аккумулятор А, который выполняет свои обычные функции промежуточного хранения данных. При выполнении ряда операций А является как источником операнда, так и приемником результата операции. Ориентация на аккумулятор, который подразумевает неявную адресацию в большинстве команд различного типа, позволила существенно сократить длину программ и, следовательно, более экономно использовать объем управляющей памяти МС.

Среди типовых признаков результата операции — только два: CY — основной перенос из старшего разряда и AC — формируемый командами сложения и используемый командой DAA дополнительный перенос из младшей тетрады в старшую. Если CY представляет основное средство для организации многобайтовых операций, то AC (по аналогии с архитектурой ВМ80) поддерживает десятичную арифметику в 2/10-коде упакованного формата. Оба признака входят в состав 8-разрядного слова состояния программы PSW.

Для повышения вычислительной мощности предназначен набор из восьми рабочих регистров R0—R7 по 8 разрядов каждый. Регистры используются для вре́менного хранения данных, что уменьшает общее число обращений к памяти МС, связанное с дополнительными затратами на ее адресацию. Среди рабочих регистров следует выделить два первых R0 и R1, которые несут дополнительную функциональную нагрузку по адресации памяти данных. По этой причине их называют адресными регистрами или регистрами-указателями.

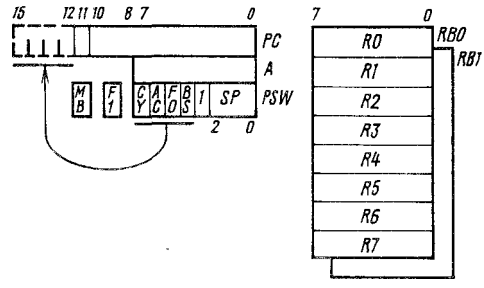


Рис. 4.3. Набор регистров центрального процессора BE48

В общем случае архитектура BE48 содержит два набора или банка рабочих регистров RB0 (Register Bank 0) и RB1, однако в каждый момент времени пользователю доступен только один из них. Выбор банка осуществляется с помощью одноразрядного указателя BS (Bank Select), входящего в состав PSW. Доступ к RB0 организуется при BS = 0, в противном случае разрешается доступ ко второму банку RB1. При начальной установке МК указатель BS сбрасывается, обеспечивая автоматический выбор RB0.

Средства переключения банков рабочих регистров дают возможность быстро менять текущий контекст процессора МС. Так, если RB0 зарезервировать под фоновую задачу, а RB1 применять для реализации процедур обслуживания прерываний, то можно существенно уменьшить время отклика МК на асинхронные запросы внешней среды, тем самым повысив скоростные характеристики прикладной системы. Одновременное использование четырех адресных регистров R0, R1, R0′ и R1′ обеспечивает быстрый доступ к четырем независимым структурам данных.

Счетчик команд PC имеет длину 12 разрядов, что позволяет обращаться к управляющей памяти емкостью 4К байт. Управляющая память разбита на два независимых банка MB0 (Memory Bank 0) и MB1 по 2К байт каждый, которые адресуются с помощью старшего разряда PC. Данный разряд программного счетчика отделен от 11 младших и никогда не изменяет своего значения при последовательной выборке команд. Следовательно, непрерывный переход из одного банка управляющей памяти в другой невозможен, поэтому используются команды передачи управления, которые содержат прямой 11-разрядный адрес внутри банка. Состояние старшего разряда PC определяется содержимым флажка MB (Memory Bank), управляемого индивидуально с помощью отдельных команд выбора RB0. При инициализации МК флажок MB и программный счетчик сбра-

сываются, обеспечивая обращение к первой команде нулевого банка.

Флажки-признаки операций CY и AC, а также указатель регистрового банка BS упакованы в старшей тетраде 8-разрядного слова состояния программы PSW. Четвертый разряд тетрады занимает флажок общего назначения F0, функциональный смысл которого определяется программистом. Существует ряд команд по его очистке, инвертированию и тестированию с условной передачей управления, позволяющих реализовать эффективную обработку однобитовых данных, различные флажковые и семафорные процедуры. Младшая тетрада PSW занята трехразрядным указателем стека SP.

Наряду с F0 имеется еще один флажок пользователя общего назначения F1. Его функции аналогичны F0, однако, в отличие от последнего, F1, как и флажок MB, не входит в состав PSW и располагается как независимый программно-доступный разряд. Это обстоятельство приводит к невозможности эффективной смены их состояний с возвратом в исходное при очередном переключении контекстов ЦП и ограничивает функциональные возможности F1 по сравнению с F0.

Входящий в состав PSW трехразрядный указатель стека обеспечивает организацию системного стека из восьми двухбайтовых ячеек. Стек служит для промежуточного хранения старшей тетрады PSW и определяемых текущим состоянием PC адресов возврата при обработке прерываний и вызовах подпрограмм. Набор признаков CY, AC, F0, BS совместно с 12-разрядным PC образуют двухбайтовый объект, автоматически загружаемый в стек или выбираемый обратно при очередной операции над ним. При этом старшее 4-разрядное поле объекта состоит из признаков. В отличие от архитектуры BM80 стек BE48 заполняется в сторону увеличения адресов и указывает на его первую свободную ячейку. При ограниченной глубине стека допускается до восьми уровней вложения, в противном случае возникает переполнение стека с переходом через границу на его начало. Переход через границу в обратном направлении возможен при попытке получить данные из уже пустого стека. При инициализации SP устанавливается в 0.

4.4. Организация памяти BE48

Пространство внутренней памяти данных DSEG содержит 256 байт, адресуемых косвенно через два адресных регистра R0, R1 выбранного в данный момент регистрового банка. В архитектуре отсутствует прямой способ адресации памяти, что привело к коротким однобайтовым форматам команд пересылки и обработки, за исключением команд с непосредственными данными.

Физическая оперативная память однокристальных МК, как правило, не покрывает всего пространства внутренней памяти данных из-за нехватки площади кристалла. Так, в приборах BE48 только первые 64 байта покрыты физическим ОЗУ. С усовершенствованием интегральной технологии ожидается увеличение объемов внутреннего физического ОЗУ МК и его постепенное приближение к потенциальному барьеру 256 байт. Например, в однокристальном МК BE49, имеющем такую же базовую архитектуру, что и BE48, объем физического ОЗУ на кристалле увеличен до 128 байт, а в 8050 все пространство в 256 байт покрыто физическим ОЗУ.

Особенностью архитектуры BE48 является совмещение обоих банков рабочих регистров и системного стека с пространством внутренней памяти данных, что дает возможность рассматривать хранящиеся в них объекты с двух сторон. Структура наложения представлена на рис. 4.4. Слова в памяти данных располагаются в двух соседних байтах. В байте с младшим адресом хранится младшая часть слова, его адрес является адресом всего слова. Такая форма хранения слов соответствует типовой и подобна используемой в BM80.

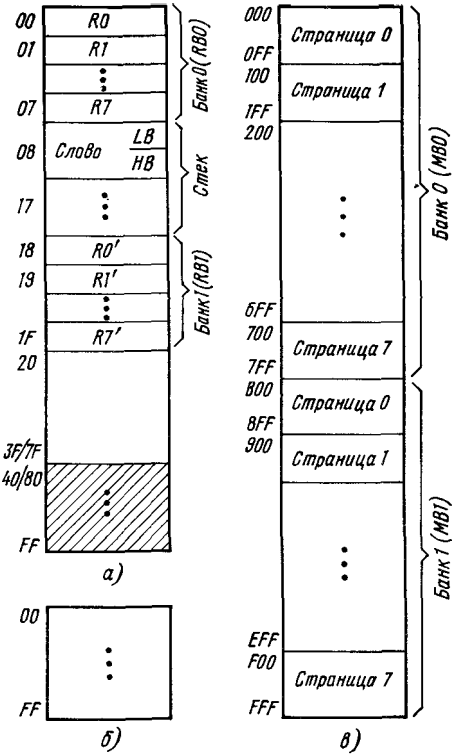


Рис. 4.4. Организация памяти BE48:
а — внутренняя память данных 256 байт, б — внешняя память данных 256 байт, в — память программ 4К байт

Возможен единственный способ адресации внутренней памяти данных — косвенный регистровый по R0 или R1. Однако часть ячеек все же оказывается доступной с помощью регистрового способа адресации, который может интерпретироваться как короткий прямой.

Совмещение системного стека и банков рабочих регистров с пространством внутренней памяти данных уменьшает и без того небольшой объем физического ОЗУ, свободного для хранения переменных. Этот недостаток частично компенсируется возможностью расширения ОЗУ внешними средствами. Архитектура BE48 предусматривает второй тип пространства памяти данных

XSEG, называемой внешней, хотя и с ограниченными ресурсами доступа. Существует только одна команда

```
MOVX    @Ri,A          ;XSEG (Ri)←A
MOVX    A,@Ri          ;A←XSEG (Ri)
```

с косвенным способом адресации через адресные регистры R0 и R1, позволяющая реализовать простой обмен с этой областью памяти общим объемом 256 байт. Тем не менее в некоторых прикладных системах потенциальная возможность такого расширения оказывается важной. Разработаны специальные микросхемы 8155/8156, с помощью которых можно получить расширение достаточно простыми средствами.

Кодовая память CSEG МК BE48 содержит два независимых банка по 2К байт. Банки разбиты на 256-байтовые страницы, по 8 страниц в каждом. Существует ряд ограничений на управление ходом выполнения программного кода внутри страниц и банков, а также на переходы через их границы.

В большинстве команд передачи управления используется короткий 8-разрядный адрес внутри текущей страницы, не обеспечивающий возможности перехода через границу страниц. Однако для последовательной выборки команд граница страниц прозрачна и может быть пересечена в любой момент времени.

В отличие от границы страниц граница банков полностью закрыта и для последовательной выборки. Поэтому единственный способ пересечения границы банка состоит в передаче управления по командам

```
JMP     addr11
CALL    addr11
RET
RETR
```

В первых двух командах используется прямой 11-разрядный адрес внутри банка, который загружается в младшие разряды PC. Состояние старшего разряда программного счетчика определяется по флажку выбора банка MB, управляемого отдельными командами:

```
SEL     MB0
SEL     MB1
```

В командах возврата полный 12-разрядный адрес, включающий разряд адресации банка, выбирается из стека системы. Несмотря на то, что программирование в пространстве программного кода, организованном таким образом, затруднено, результат может оказаться намного эффективнее, чем при работе в обычной однородной среде линейного типа.

Подобно памяти данных физическая память программ МК не покрывает все отводимое под нее пространство. Обычно на кристалле МК располагается (1—2) К байт ПЗУ. Оставшаяся часть может быть реализована внешними средствами, например с помощью специальной БИС 8355/8755. Однако, в отличие от памяти данных, внешняя и внутренняя памяти программ рассматриваются как единое целое. Различия между ними возникают только на уровне структуры и физического обмена.

Память программ изолирована от памяти данных, но в ней предусмотрена возможность хранения неизменяемых данных, например постоянных таблиц и массивов, что также снижает жесткие ограничения на объем используемых данных, связанные с малым объемом ОЗУ. Существуют два способа доступа к данным, расположенным в управляющей памяти: непосредственная адресация и косвенная регистровая страничная адресация через А. В последнем случае обеспечивается доступ либо к текущей странице, либо к странице 3, резервируемой специально для хранения констант различного типа.

Ячейка памяти программ с нулевым адресом является стартовой. При сбросе системы в РС и МВ записывается 0, что обеспечивает автоматическую передачу управления по стартовому адресу МК. Адреса 003 и 007 также являются точками входа в память программ обслуживания прерываний от внешнего запроса и таймера/счетчика соответственно.

4.5. Система ввода-вывода и служба реального времени BE48

Архитектура BE48 содержит 27 резидентных линий ВВ (рис. 4.5), организованных в три 8-разрядных порта BUS, P1, P2 и три тестируемые командами условного перехода независимые линии T0, T1, INT. Порты P1 и P2 имеют одинаковую структурную схему (рис. 4.6) и могут быть запрограммированы либо на ввод, либо на вывод данных. Выходные буфера портов имеют встроенные регистры, которые запоминают выводимые данные. При вводе данных информация поступает непосредственно в аккумулятор, минуя выходной буферный регистр, который должен находиться в состоянии 0FFH. В противном случае выполняется операция маскирования вводимых данных по схеме «монтажное ИЛИ» с содержимым выходного регистра. Это позволяет одни линии портов использовать для ввода, а другие — для вывода. Порт BUS отличается от P1 и P2 тем, что в нем применяется выходной буфер с тремя состояниями. Поэтому он не допускает одновременного смешивания ввода и вывода между своими разрядами.

Расширенное пространство ВВ содержит еще четыре 4-разрядных порта P4—P7, реализуемые внешними средствами. В системе команд предусмотрен ряд операций с прямым доступом к этим

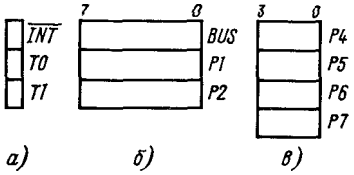


Рис. 4.5. Пространство ВВ ВЕ48:
а — тестируемые флажки; б — внутренние порты ВВ; в — внешние порты ВВ

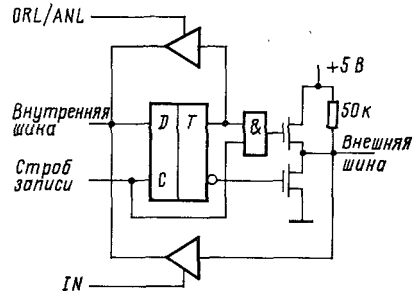


Рис. 4.6. Схема портов P1, P2

портам. Для осуществления этой возможности разработан специальный расширитель ВВ — микросхема 8243.

Архитектура ВЕ48 включает достаточно простые, но эффективные средства реального времени, среди которых 8-разрядный таймер/счетчик Т и одноуровневая система прерываний с двумя источниками запросов: внешними INT и внутренними TF.

Организация 8-разрядного таймера/счетчика ВЕ48 представлена на рис. 4.7. Специальная команда пересылки

```
MOV     A,T          ;A←T
MOV     T,A          ;T←A
```

обеспечивает программный доступ к его содержимому. Устройство работает в двух режимах: таймера и счетчика, в которые оно переводится по командам

```
START   T            ;Пуск в режиме таймера
START   CNT          ;Пуск в режиме счет-
                     ;чика
```

Устройство останавливает свою работу при подаче сигнала аппаратного сброса RESET или выполнении команды

```
STOP    TCNT         ;Останов таймера/счет-
                     ;чика
```

По команде START mode таймер/счетчик ведет непрерывный счет импульсов на своем входе по $\mathrm{mod}\, 2^8$. При переходе через границу инициируется запрос на прерывание и устанавливается флажок переполнения TF, который может быть проверен командой условного перехода

```
JTF     addr8        ;Если TF = 1, тогда
                     ;PC0-7←addr 8, TF←0
```

При выполнении команды перехода или при начальной установке по сигналу RESET флажок TF сбрасывается.

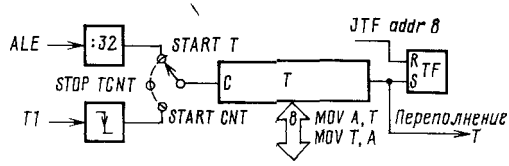


Рис. 4.7. Организация таймера/счетчика BE48

В режиме счетчика организуется подсчет числа переходов из 1 в 0 на внешнем входе T1. Максимальная скорость приращения счетчика составляет единицу на три машинных цикла. При этом напряжение низкого уровня на входе T1 должно удерживаться не менее одного, а высокого уровня не менее половины машинного цикла.

В режиме таймера устройство считает метки времени, генерируемые внутренними узлами МК. Эти метки являются результатом деления частоты следования машинных циклов ALE на 32. Во время выполнения команды START T этот делитель устанавливается в 0.

Используя начальную загрузку T и обнаруживая установку флажка TF, можно получить паузы от (1/32)ALE до 256(1/32)ALE. Организация пауз вне данного диапазона осуществляется с помощью программных циклов. Более короткие паузы можно также получить в режиме счетчика с T1, равным (1/3)ALE.

Встроенная в BE48 одноуровневая система прерываний с двумя источниками запросов, имеющими фиксированные векторы прерываний и приоритеты, представлена на рис. 4.8. Одноуровневость системы означает, что никакие новые запросы не воспринимаются до тех пор, пока не будет обслужен текущий.

В системе определены два источника: вход $\overline{\text{INT}}$ (стартовый адрес программы обслуживания 003H) и переполнение T (адресом программы обслуживания служит 007H). Каждый источник может быть замаскирован независимо от других специальными флажками EX и ET, управляемыми командами

```
EN      I       ;Разрешение внешнего
                ;прерывания INT
DIS     I       ;Запрет внешнего пре-
                ;рывания INT
EN      TCNT    ;Разрешение внутренне-
                ;го прерывания от T
DIS     TCNT    ;Запрет внутреннего
                ;прерывания от T
```

Флажки также сбрасываются при подаче сигнала RESET. В случае одновременной фиксации двух запросов приоритет отдается внешнему.

Внешнее прерывание по входу $\overline{\text{INT}}$ воспринимается, когда на нем появляется напряжение низкого уровня. Линия $\overline{\text{INT}}$ контроли-

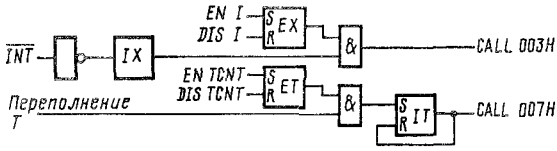


Рис. 4.8. Система прерываний BE48

руется во время ALE в последнем машинном цикле каждой команды. Результат тестирования запоминается во внутреннем триггере внешнего прерывания IX, установка которого инициирует формирование аппаратного эквивалента команды CALL 003H. При выполнении данной команды текущее содержимое PC и флажки PSW запоминаются в стеке, обеспечивая возможность возврата в точку прерывания. Программа обслуживания прерывания должна завершаться командой RETR, восстанавливающей PC и PSW. В начале второго цикла команда RETR разрешает прием новых запросов на прерывание. К этому моменту сигнал запроса $\overline{\text{INT}}$ должен быть снят внешними средствами. Обычно это делается автоматически в момент обслуживания устройства, выставившего запрос. Если такая возможность не осуществляется, то любая выходная линия BE48 может быть использована для выдачи специального сигнала подтверждения прерывания.

Сигнал переполнения от T фиксируется в триггере запроса IT. Он может быть замаскирован программно-управляемым флажком ET. При возникновении разрешенного прерывания генерируется аппаратная команда CALL 007H, сбрасывающая флажок IT. Сброс IT осуществляется автоматически при любом исполнении команды CALL 007H.

Частота выходного сигнала генератора OSC делится на 3 (рис. 4.9) для получения основной тактовой частоты CLK, которая может быть выведена на внешний вывод T0 по команде

```
ENT0    CLK     ;Разрешение вывода CLK на T0
```

Режим вывода отменяется только при общем сбросе МК.

4.6. Система команд BE48

Базовая архитектура BE48 содержит 96 машинных или 52 символьных команды, которые обеспечивают широкие возможности реализации сложных программных структур. Они имеют длину 1—2 байта, причем более 70% команд — однобайтовые. Все команды выполняются за один или два цикла и занимают соответственно 2,5 или 5 мкс при частоте 6 МГц на входе XTAL. Свыше 50% команд — одноцикловые, некоторые однобайтовые команды — двухцикловые. Команды объединены в пять групп: пересылки, обработки данных, манипуляции флажками, передачи управления, ВВ и управления режимом реального времени, которые приведены в табл. 4.2—4.6. Команды помеченные любым

знаком сноски, не входят в систему команд 8021, которая является усеченным вариантом системы BE48. Кроме того, команды с номером сноски выше первого не включены в систему 8022. Команды, помеченные третьим или четвертым знаком сноски, не входят в набор BE41. Единственная команда с четвертым знаком сноски не является базовой и определена только в КМОП МП. (Расширение BE41 будет рассмотрено в § 4.9.) Системы команд BE49/BE50 и BE48 тождественны.

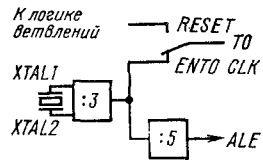


Рис. 4.9. Система синхронизации BE48

Группа команд пересылки, приведенная в табл. 4.2, дает возможность организовать передачу и обмен данными между аккумулятором и рабочими регистрами (или ячейками) внутренней памяти данных. В последнем случае используется только косвенный регистровый способ адресации через указатели R0, R1 выбранного банка рабочих регистров. Предусмотрены команды загрузки A, Rr, r=0—7, и внутреннего ОЗУ непосредственными данными. Все команды из подгруппы обмена оперируют байтами кроме двух, которые оперируют тетрадами, что может быть полезным при обработке 2/10-чисел и других 4-разрядных значений.

Аккумулятор может также выполнять обмен данными с внешней памятью XSEG, PSW и считывать константы из памяти программ CSEG. Допускается считывание либо из текущей

Таблица 4.2

Мнемоника		Число байтов	Число циклов	Код	CY	Описание
MOV	A,Rr	1	1	F8—FF	—	A←Rr, r=0−7
MOV	A,@Ri	1	1	F0—F1	—	A←(Ri), i=0−1
MOV	A,#data	2	2	23	—	A←data
MOV	Rr, A	1	1	A8—AF	—	Rr←A, r=0−7
MOV	@Ri, A	1	1	A0—A1	—	(Ri)←A, i=0−1
MOV	Rr,#data	2	2	B8—BF	—	Rr←data, r=0−7
MOV	@Ri, #data	2	2	B0—B1	—	(Ri)←data, i=0−1
MOV	A,PSW*[2]	1	1	C7	—	A←PSW
MOV	PSW, A*[2]	1	1	D7	+	PSW←A
XCH	A, Rr	1	1	28—2F	—	A↔Rr, r=0−7
XCH	A,@Ri	1	1	20—21	—	A↔(Ri), i=0−1
XCHD	A,@Ri	1	1	30—31	—	A_{0-3}↔$(Ri)_{0-3}$, i=0−1
SWAP	A	1	1	47	—	A_{0-3}↔A_{7-4}
MOVX	A,@Ri*[3]	1	2	80—81	—	A←XSEG(Ri), i=0−1
MOVX	@Ri, A*[3]	1	2	90—91	—	XSEG (Ri)←A, i=0−1
MOVP	A,@A	1	2	A3	—	A←CSEG (PC_{8-11}:A)
MOVP3	A,@A*[2]	1	2	E3	—	A←CSEG (3:A)

Таблица 4.3

Мнемоника		Число байтов	Число циклов	Код	CY	Описание
ADD	A,Rr	1	1	68—6F	+	A←A+Rr, r=0−7
ADD	A,@Ri	1	1	60—61	+	A←A+(Ri), i=0−1
ADD	A,#data	2	2	03	+	A←A+data
ADDC	A,Rr	1	1	78—7F	+	A←A+Rr+CY, r=0−7
ADDC	A,@Ri	1	1	70—71	+	A←A+(Ri)+CY, i=0−1
ADDC	A,#data	2	2	13	+	A←A+data+CY
ANL	A,Rr	1	1	58—5F	—	A←A AND Rr, r=0−7
ANL	A,@Ri	1	1	50—51	—	A←A AND (Ri), i=0−1
ANL	A,#data	2	2	53	—	A←A AND data
ORL	A,Rr	1	1	48—4F	—	A←A OR Rr, r=0−7
ORL	A,@Ri	1	1	40—41	—	A←A OR (Ri), i=0−1
ORL	A,#data	2	2	43	—	A←A OR data
XRL	A,Rr	1	1	D8—DF	—	A←A XOR Rr, r=0−7
XRL	A,@Ri	1	1	D0—D1	—	A←A XOR (Ri), i=0−1
XRL	A,#data	2	2	D3	—	A←A XOR data
INC	A	1	1	17	—	A←A+1
DEC	A	1	1	07	—	A←A−1
CLR	A	1	1	27	—	A←0
CPL	A	1	1	37	—	A←NOT A
DA	A	1	1	57	+	Десятичная коррекция A
RL	A	1	1	E7	—	$A_0 \leftarrow A_7$, $A_n \leftarrow A_{n-1}$, n=1−7
RLC	A	1	1	F7	+	$A_0 \leftarrow CY$, $A_n \leftarrow A_{n-1}$, $CY \leftarrow A_7$, n=1−7
RR	A	1	1	77	—	$A_7 \leftarrow A_0$, $A_n \leftarrow A_{n+1}$, n=0−6
RRC	A	1	1	67	+	$A_7 \leftarrow CY$, $A_n \leftarrow A_{n+1}$, $CY \leftarrow A_0$, n=0−6
INC	Rr	1	1	18—1F	—	Rr←Rr+1, r=0−7
INC	@Ri	1	1	10—11	—	(Ri)←(Ri)+1, i=0−1
DEC	Rr	1	1	C8—CF	—	Rr←Rr−1, r=0−7

страницы, адресуемой PC_{8-11}, либо из страницы 3, резервируемой специально для данного случая. Смещение внутри страницы определяется содержимым A.

Команды арифметической и логической обработки данных представлены в табл. 4.3. Все команды с двумя исходными операндами предполагают, что первый операнд находится в аккумуляторе. Вторым может быть содержимое рабочего регистра или байта внутренней памяти данных, либо литерал, кодируемый непосредственно в команде. Результат такой операции всегда пересылается в аккумулятор. В группе определены команды:

```
ADD     A,src          ;Сложение
ADDC    A,src          ;Сложение с переносом
ANL     A,src          ;Логическое И
```

```
ORL     A,src       ;Логическое ИЛИ
XRL     A,src       ;Логическое исключающее ИЛИ
```

Команды с одним источником, как правило, работают с аккумулятором. Среди них команды приращения и уменьшения содержимого А на единицу, очистки, инверсии и сдвига вправо или влево. Модификация А на единицу реализуется без фиксации переноса CY. Сдвиги строятся как с учетом состояния флажка CY, так и без него. Для усиления вычислительной мощности в состав команд обработки введены три команды по приращению или уменьшению на единицу содержимого R0—R7 и приращению содержимого любой ячейки внутренней памяти данных. Благодаря этому появляется возможность организации разнообразных счетчиков программными средствами.

В группе отсутствует операция вычитания, которую легко можно выполнить с помощью команд логического дополнения, приращения и арифметического сложения. Отсутствуют также какие-либо операции с данными из внешней памяти. Для их выполнения информация сначала должна быть переслана в аккумулятор.

В состав системы команд BE48 входит ряд команд по работе над одноразрядными флажками CY, F0 и F1 (табл. 4.4). Флажок CY является основным арифметическим признаком; F0, F1 носят общий характер, их функциональное назначение определяется пользователем. Над каждым из флажков предусмотрены команды очистки CLR и инверсии CPL.

Для выбора банков рабочих регистров и памяти программ используются флажки BS и MB соответственно. Прямая манипуляция их содержимым реализуется с помощью команды выбора SEL (см. табл. 4.4). Выбор нулевого банка RB0 или MB0 соответствует загрузке в BS или MB нуля, выбор первого банка RB1 или MB1 — единицы.

Система команд BE48 включает 16 команд перехода, из которых 14 условных (табл. 4.5). Только одна команда подгруппы осуществляет переход в любое место программной памяти емкостью 4К байт. Для этого она использует прямой 11-разрядный адрес внутри банка и текущее значение флажка выбора банка MB. Управление флажком MB реализуется отдельными командами SEL MB0 и SEL MB1 из группы битовой обработки. Остальные команды разрешают переход внутри текущей страницы.

Команда безусловной передачи JMPP @A организует переход по содержимому таблицы ветвлений, расположенной в текущей странице. Содержимое аккумулятора А применяется в качестве смещения в таблице, что дает возможность простейшими средствами реализовать программные переключатели. Команда DJNZ

Таблица 4.4

Мнемоника		Число байтов	Число циклов	Код	CY	Описание
CLR	C	1	1	97	0	CY←0
CPL	C	1	1	A7	+	CY←NOT CY
CLR	F0*[2]	1	1	85	—	F0←0
CPL	F0*[2]	1	1	95	—	F0←NOT F0
CLR	F1*[2]	1	1	A5	—	F1←0
CPL	F1*[2]	1	1	B5	—	F1←NOT F1
SEL	RB0*[2]	1	1	C5	—	BS←0
SEL	RB1*[2]	1	1	D5	—	BS←1
SEL	MB0*[3]	1	1	E5	—	MB←0
SEL	MB1*[3]	1	1	F5	—	MB←1

Rr, addr8 уменьшает содержимое одного из рабочих регистров на единицу и делает переход, если его содержимое не равно нулю. Данная команда очень эффективна при организации программных циклов. Двенадцать команд условного перехода тестируют на 0 содержимое A, состояние флажков CY, F0, F1, TF и внешних входов T0, T1, $\overline{\text{INT}}$. Ветвление внутри страницы выполняется в зависимости от результата тестирования. Последняя команда JBb addr8 дает возможность проверить состояние любого бита Ab, b=0–7, аккумулятора.

В составе группы передачи управления три команды для организации вызова и возврата из подпрограмм. Команда CALL addr11 подобно JMP addr11 использует 11-разрядный адрес внутри банка и состояние MB для выбора банка. По данной команде текущее значение 12-разрядного PC и четыре флажка BS, F0, AC и CY из PSW «заталкиваются» в стек. Команда RETR возвращает в исходное состояние PC и PSW. Команда RET восстанавливает только PC, давая возможность использовать флажки для передачи параметров.

Отсутствие средств чтения текущего состояния MB вызывает ряд трудностей при программировании. Так, при возврате из подпрограмм состояние MB в общем случае не определено, так как не известно, изменялся он подпрограммой или нет. Данное обстоятельство требует некоторого соглашения на правила манипуляции флажком при вызове подпрограмм. В частности, процедуры обслуживания прерывания, точки входов которых расположены в нулевом банке, вообще не должны изменять его состояние. Следовательно, эти процедуры должны быть полностью расположены в нулевом банке памяти программ.

Пустая команда NOP обеспечивает простой переход к следующему элементу программной последовательности. В КМОП-вариантах МК, например 80C49/80C39, предусмотрена еще одна команда IDL, которая не входит в базовый набор. Эта команда переводит МК в состояние останова, свободное от выполнения

Таблица 4.5

Мнемоника		Число байтов	Число циклов	Код	CY	Описание
JMP	addr11	2	2	aaa00100	—	PC←addr11, PC_{11}←MB
JMPP	@A	1	2	B3	—	PC_{0-7}←CSEG(PC_{8-11}:A)
DJNZ	Rr,addr8	2	2	E8—EF	—	Rr←Rr−1, если Rr≠0, то PC_{0-7}←addr8, n=0−7
JC	addr8	2	2	F6	—	Если CY=1, то PC_{0-7}←←addr8
JNC	addr8	2	2	E6	—	Если CY=0, то PC_{0-7}←←addr8
JZ	addr8	2	2	C6	—	Если A=0, то PC_{0-7}←←addr8
JNZ	addr8	2	2	96	—	Если A≠0, то PC_{0-7}←←addr8
JT0	addr8*[1]	2	2	36	—	Если T0=1, то PC_{0-7}←←addr8
JNT0	addr8*[1]	2	2	26	—	Если T0=0, то PC_{0-7}←←addr8
JT1	addr8	2	2	56	—	Если T1=1, то PC_{0-7}←←addr8
JNT1	addr8	2	2	46	—	Если T1=0, то PC_{0-7}←←addr8
JF0	addr8*[2]	2	2	B6	—	Если F0=0, то PC_{0-7}←←addr8
JF	addr8*[2]	2	2	76	—	Если F1=0, то PC_{0-7}←←addr8
JTF	addr8	2	2	16	—	Если TF=1, то PC_{0-7}←←addr8, TF←0
JNI	addr8*[3]	2	2	86	—	Если INT=0, то PC_{0-7}←←addr8
JBb	addr8*[2]	2	2	bbb10010	—	Если Ab=0, то PC_{0-7}←←addr8, b=0−7
CALL	addr11	2	2	aaa10100	—	(SP)←PSW_{4-7}, (SP)←←PC←addr11, PC_{11}←MB, SP←SP+1
RET		1	2	83	—	SP←SP−1, PC_{0-12}←(SP)
RETR*[2]		1	2	93	+	SP←SP−1, PC_{0-12}←(SP), PSW_{4-7}←(SP)
NOP		1	1	00	—	Нет операции
IDL*[4]		1	1	01	—	Перевод МК в режим с малым потреблением мощности

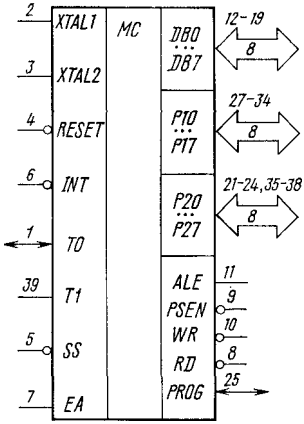


Рис. 4.10. Условное графическое обозначение BE48

каких-либо последующих команд. Тем не менее в состоянии останова поддерживается работа средств реального времени и при поступлении запроса на прерывание МК может перейти на подпрограмму его обслуживания. Выход из состояния останова осуществляется также при общем сбросе системы.

В табл. 4.6 приведены команды, которые обслуживают пространство ВВ, состоящее из трех 8-разрядных внутренних портов BUS, P1, P2 и четырех 4-разрядных внешних портов P4—P7. Ввод и вывод данных осуществляется через аккумулятор или его младшую половину. Предусматриваются логические команды И и ИЛИ содержимого портов BUS, P1, P2 с литералом, представляющие простейшую, но до-

Таблица 4.6

Мнемоника		Число байтов	Число циклов	Код	CY	Описание
IN	A,Pp	1	2	09—0A	—	A←Pp, p=1−2
OUTL	Pp,A	1	2	39—3A	—	Pp←A, p=1−2
ANL	Pp,#data*[2]	2	2	99—9A	—	Pp←Pp AND data, p=1−2
ORL	Pp,#data*[2]	2	2	89—8A	—	Pp←Pp OR data, p=1−2
INS	A, BUS*[3]	1	2	08	—	A←BUS
OUTL	BUS,A*[3]	1	2	02	—	BUS←A
ANL	BUS,#data*[3]	2	2	98	—	BUS←BUS AND data
ORL	BUS,#data*[3]	2	2	88	—	BUS←BUS OR data
MOVD	A,Pp	1	2	0C—0F	—	A_{0-3}←Pp, A_{4-7}←0, p=4−7
MOVD	Pp,A	1	2	3C—3F	—	Pp←A_{0-3}, p=4−7
ANLD	Pp,A	1	2	9C—9F	—	Pp←Pp AND A_{0-3}, p=4−7
ORLD	Pp,A	1	2	8C—8F	—	Pp←Pp OR A_{0-3}, p=4−7
MOV	A,T	1	1	42	—	A←T
MOV	T,A	1	1	62	—	T←A
STRT	T	1	1	55	—	Пуск T
STRT	CNT	1	1	45	—	Пуск CNT
STOP	TCNT	1	1	65	—	Останов T/CNT
EN	TCNTI*[2]	1	1	25	—	Разрешение прерываний от T/CNT
DIS	TCNTI*[2]	1	1	35	—	Запрет прерываний от T/CNT
ENT0	CLK*[3]	1	1	75	—	Разрешение CLK на T0
EN	I*[2]	1	1	05	—	Разрешение прерываний по INT
DIS	I*[2]	1	1	15	—	Запрет прерываний по INT

статочно эффективную обработку операндов из внутреннего пространства ВВ. При работе с внешними портами вторым операндом команд И и ИЛИ служит содержимое младшей тетрады А. Команды этого типа дают возможность манипулировать отдельными битами портов ВВ независимо от других.

В архитектурах 8021/8022 порт BUS заменен на стандартный порт P0. Поэтому операции ВВ с ним осуществляются в рамках общей команды

```
IN      A,Pp        ;A←Pp,  p=0−2
OUTL    Pp,A        ;Pp←A,  p=0−2
```

Подгруппа управления средствами реального времени (см. табл. 4.6) содержит команды, работающие с таймером/счетчиком и подсистемой прерываний. Сюда входят операции чтения и загрузки состояния устройства счета, его запуск в режиме таймера или счетчика, останов, а также четыре команды по разрешению и запрещению прерываний. Специальная операция ENTO CLK программирует выдачу синхроимпульсов CLK=(1/3)OSC на выход T0.

4.7. Физический интерфейс ВЕ48

Условное графическое обозначение МК ВЕ48 приведено на рис. 4.10. Приборы размещаются в стандартных 40-выводных корпусах типа 2123.40 с двухрядным расположением выводов. Приведем состав физического интерфейса ВЕ48:

V_{SS}	20	Общий
V_{DD}	26	Напряжение питания +5 В или +21 В в режиме программирования УСППЗУ
V_{CC}	40	Напряжение питания +5 В
PROG	25	Выходной строб для расширителя ВВ или программирующий импульс +18 В в режиме программирования УСППЗУ
P10—P17	27—34	Псевдодвунаправленный 8-разрядный порт ВВ
P20—P27	21—24, 35—38	Псевдодвунаправленный 8-разрядный порт ВВ, младшая тетрада которого P20—P23 используется для вывода старшей части адреса при обращении к внешней памяти программ или как канал связи с расширителем ВВ
DB0—DB7	12—19	Двунаправленный 8-разрядный канал ВВ данных либо синхронно со стробами $\overline{RD}$, $\overline{WR}$, либо асинхронно. При обращении к внешней памяти программ используется для выдачи младшего байта адреса синхронно с ALE и приема очередного байта программной последовательности синхронно с $\overline{PSEN}$. При обращении к внешней памяти данных служит для выдачи младшего байта адреса памяти синхронно с ALE и передачи

		данных в соответствии со стробами $\overline{RD}$, $\overline{WR}$
ALE	11	Строб приема адреса. Генерируется в начале каждого машинного цикла и может быть применен для синхронизации внешних цепей
$\overline{RD}$	8	Строб чтения данных. Генерируется в циклах чтения шины BUS. Используется как строб чтения внешней памяти данных
$\overline{WR}$	10	Строб выдачи данных. Генерируется в циклах вывода информации через шину BUS. Применяется как строб записи данных во внешнюю память
$\overline{RESET}$	4	Вход начальной установки. Используется в режимах программирования УСППЗУ
$\overline{PSEN}$	9	Строб чтения внешней памяти программ
EA	7	Разрешение доступа к внешней памяти. Применяется для запрета внутреннего ПЗУ в режимах эмуляции и отладки. Подача на вход напряжения +18 В переводит прибор в режим программирования УСППЗУ
$\overline{SS}$	5	Пошаговый режим. Используется совместно с ALE для организации пошагового исполнения команд
XTAL1, XTAL2	2, 3	Вход и выход для подключения кварцевого резонатора, а также вход для внешнего ГТИ
T0	1	Тестовый вход или выход для выдачи импульсов CLK, формируемых внутри МК
T1	39	Тестовый вход, а также вход для приема импульсов счета
$\overline{INT}$	2	Тестовый вход, а также вход для приема внешнего запроса на прерывание с низким уровнем активности

Встроенный в BE48 генератор работает в диапазоне частот 1—11 МГц. Вывод XTAL1 служит входом, а XTAL2 — выходом каскада усиления. Внешний кварцевый резонатор или RC-цепочка подключаются к XTAL1, XTAL2, обеспечивая необходимую для генерации внешнюю связь и сдвиг фаз.

В результате деления частоты OSC на 3 получается основная тактовая частота CLK, определяющая работу внутренних узлов МК. Синхросигнал машинных циклов ALE формируется путем дополнительного деления CLK на 5 и постоянно присутствует на одноименном выходе МК.

Каждый машинный цикл состоит из пяти тактов CLK, именуемых S1—S5. Для выполнения команды требуется один или два машинных цикла. Прием команды всегда осуществляется в первом такте S1 первого машинного цикла M1. Во втором такте S2 принятая команда декодируется, а PC получает приращение. Такты S3—S5 отводятся для внутреннего исполнения команды.

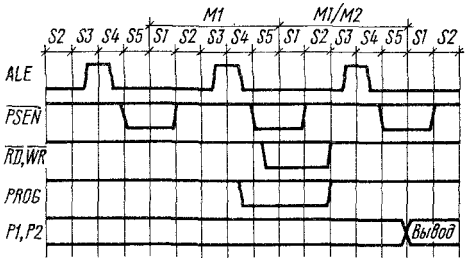


Рис. 4.11. Временны́е диаграммы командного цикла BE48

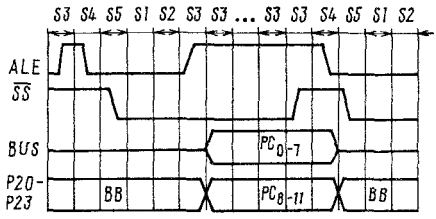


Рис. 4.12. Временны́е диаграммы пошагового режима работы

В случае двухбайтовой или одноцикловой команды одновременно с этим память программ предъявляет новое значение PC. Новый элемент программной последовательности считывается в такте S1 следующего машинного цикла. Для одноцикловой команды S1 — начало нового командного цикла.

Во втором машинном цикле двухбайтовая команда считывает второй байт программной последовательности, представляющий либо непосредственные данные, либо младшую часть адреса. Функциональное распределение тактов S1—S5 аналогично M1.

Однобайтовая двухцикловая команда с обращением к управляющей памяти во втором машинном цикле выполняется по аналогии с двухбайтовой. Все другие двухцикловые команды в такте S2 второго машинного цикла считывают данные либо из внешнего ОЗУ, либо из портов. В этом случае необходимая адресная информация передается в тактах S3—S5 первого машинного цикла. Обобщенная временна́я диаграмма тактирования командных циклов представлена на рис. 4.11.

Входные сигналы всех портов совместимы с ТТЛ, а на выходах может быть подключена стандартная ТТЛ-нагрузка. Вывод данных через порты P1, P2 осуществляется в такте S5 второго машинного цикла. При работе с портом BUS операции ВВ сопровождаются генерацией импульсов RD, WR, которые могут быть использованы для организации синхронного ВВ. Данные действительны на срезе стробов.

В МК BE48 предусмотрена возможность организации пошагового режима работы, что очень важно для отладки и проверки содержимого памяти программ. В этом случае программа исполняется по шагам, каждый шаг — это один машинный цикл. Управление началом следующего шага осуществляется с помощью сигнала на входе SS, как показано на рис. 4.12. При работе в пошаговом режиме содержимое порта BUS теряется, но может быть сохранено дополнительными

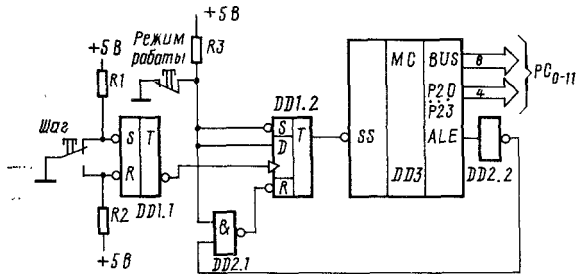


Рис. 4.13. Организация пошагового режима работы

внешними средствами, буферизирующими его состояние по срезу ALE.

Появление напряжения низкого уровня на входе $\overline{SS}$ вынуждает МК приостановить свою работу на этапе выборки следующего элемента командной последовательности. Микроконтроллер подтверждает состояние останова высоким уровнем напряжения выходного сигнала ALE и выводом на шину BUS и младшую половину порта P2 адреса следующей команды. Это состояние может продолжаться сколь угодно долго до момента перехода сигнала $\overline{SS}$ из 0 в 1, который выводит МК из режима останова, позволяя ему продолжать свою работу. Чтобы остановить МК на следующем шаге сигнал $\overline{SS}$ должен быть вновь активизирован сразу же после среза ALE. Простейшая схема организации пошаговой работы приведена на рис. 4.13.

При активации сигнала RESET выполняется общий сброс системы. Входная цепь встроенного в МК триггера Шмитта состоит из подключенного к источнику питания +5 В резистора и внешнего конденсатора емкостью 1 мкФ, что обеспечивает при включении питания формирование импульса системного сброса достаточной длительности. Длительность импульса, формируемого извне, должна быть не менее 50 мс при номинальном напряжении питания. Для стабилизации генератора требуется 5 машинных циклов. При AE = 1 во время импульса сброса ALE = 1 и PSEN = 1. При сбросе МК переходит в следующее исходное состояние:

PC←0
SP←0
BS←0
MB←0
BUS←z-состояние, кроме ситуации EA = 1
P1, P2←0FFH
DIS I, DIS TCNT
STOP TCNT
TF←0
F0, F1←0
Запрет вывода CLK на T0

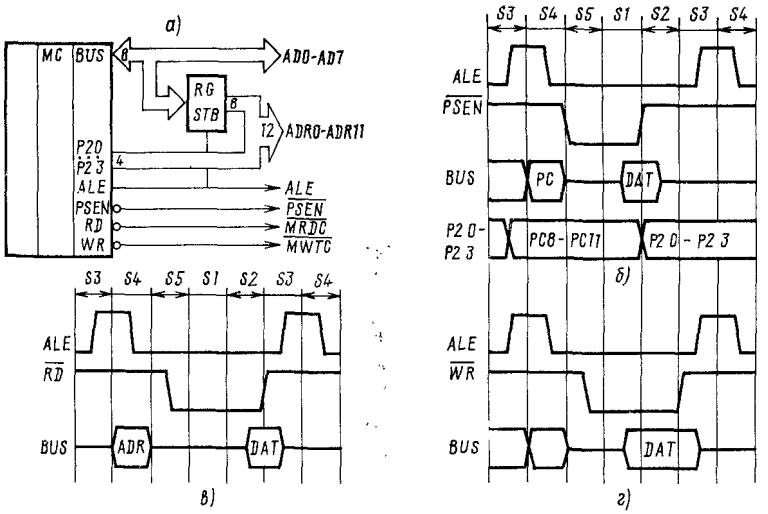


Рис. 4.14. Интерфейс расширения памяти:
а — состав; б — чтение памяти программ; в — чтение памяти данных; г — запись в память данных

4.8. Расширение внутренних ресурсов ВЕ48

Если размещенных на кристалле МК ресурсов оказывается недостаточно для реализации конкретной прикладной системы, они могут быть расширены внешними по отношению к МК средствами. Такое расширение легко выполняется за счет ввода в действие еще не использованных возможностей базовой архитектуры МК. Так, прямо адресуемая программная память может быть доведена до 4К байт, резервом дополнительной памяти данных служит поддерживаемое архитектурой пространство внешнего ОЗУ. На 16 линий может быть увеличена система ВВ. Кроме того, применяя различные способы расширения физических адресов, можно довести объемы внешних физических пространств до требуемого размера сверх норм, предусмотренных базовой архитектурой, без каких-либо ограничений сверху. Эффективная реализация специальных функций возможна с помощью стандартных периферийных БИС семейства ВМ80/ВМ85А.

Организация МК ВЕ48 поддерживает два способа прямого расширения своих вычислительных ресурсов. Первый способ — использование интерфейса расширения памяти, совместимого с системной магистралью ВМ85А. Второй способ предполагает активизацию интерфейса расширения ВВ.

В состав интерфейса расширения памяти входят двунаправленная 8-разрядная мультиплексированная шина адреса/данных BUS, шина старшей части адреса Р20—Р23, строб разрешения фиксации адреса ALE, строб чтения памяти программ PSEN, строб чтения

памяти данных $\overline{RD}$ и строб записи в память данных $\overline{WR}$. Состав линий и временны́е диаграммы работы интерфейса приведены на рис. 4.14. По срезу ALE передаваемая через шину BUS адресная информация должна запоминаться во внешнем адресном регистре. В случае обращения к памяти программ старшая часть адреса передается через 4-разрядную шину P20—P23. Тип операции определяется активацией одного из стробов $\overline{PSEN}$, $\overline{RD}$ и $\overline{WR}$. Данные считаются действительными на срезе стробов.

Цикл внешнего обращения к памяти программ инициируется каждый раз, когда содержимое PC выходит за пределы допустимого размера резидентной части ПЗУ, а также при EA = 1. Возможность полного отключения внутреннего ПЗУ используется при его эмуляции и отладке МС. Цикл обращения к внешней памяти данных является составной частью фазы исполнения команд:

```
MOVX    A,@Ri          ;A←XSEG(Ri),i=0-1
MOVX    @Ri,A          ;XSEG(Ri)←A, i=0-1
```

В операции внешнего расширения задействованы порты BUS и P2. Из временны́х диаграмм на рис. 4.14 видно, что состояние порта BUS при этом разрушается, а его выход после выполнения операции переходит в высокоимпедансное состояние. Это обстоятельство может быть учтено при организации двунаправленного обмена через шину BUS. Действительно, по команде

```
OUTL    BUS,A          ;BUS←A
```

шина BUS активизируется. Для перевода ее в z-состояние применяется команда типа MOVX, освобождающая шину для ввода информации. В отличие от порта BUS порт P2 также используется в операциях внешнего обращения к памяти программ, восстанавливая свое состояние после каждого цикла обращения.

Интерфейс расширения ВВ содержит 4-разрядную информационную шину команд/данных P20—P23 и линию управления PROG. Интерфейс активизируется по командам

```
MOVD    A,Pp           ;A0-3←Pp, A4-7←0,
                       ;p=4-7
MOVD    Pp,A           ;Pp←A0-3, p=4-7
ORLD    Pp,A           ;Pp←Pp OR A0-3,
                       ;p=4-7
ANLD    Pp,A           ;Pp←Pp AND A0-3,
                       ;p=4-7
```

Протокол интерфейса состоит в передаче через 4-разрядный канал P20—P23 кода операции и данных обмена, которые однозначно связаны с четырьмя вышеуказанными командами. Состав линии, временные диаграммы протокола и способ кодирования информа-

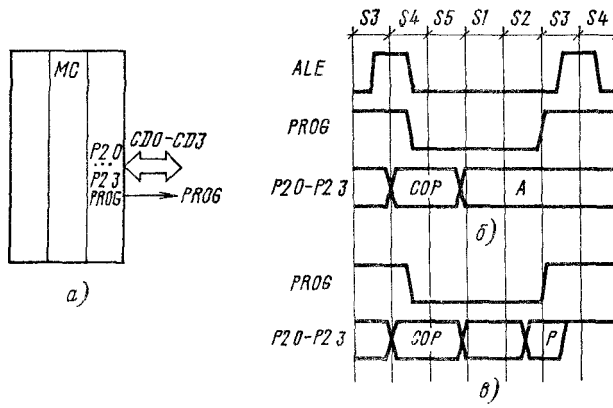


Рис. 4.15. Интерфейс расширения ВВ:
а — состав; б — вывод данных; в — ввод данных

ции приведены на рис. 4.15 и в табл. 4.7. Управление обменом осуществляется с помощью сигнала PROG, срез которого синхронизирует передачу кода операции обмена, а фронт — данных обмена.

Таблица 4.7

Код COP				Описание	Код COP				Описание
3	2	1	0		3	2	1	0	
0	0	P	P	MOVD A,Pp	1	0	P	P	ORLD Pp,A
0	1	P	P	MOVD Pp,A	1	1	P	P	ANLD Pp,A

В операциях участвует младшая тетрада аккумулятора. При чтении портов старшая тетрада A сбрасывается. Система команд предусматривает выполнение периферийной операции логического AND и OR содержимого порта с состоянием младшей тетрады аккумулятора. Пользователь может по своему усмотрению интерпретировать эти команды и команды пересылки, внося новый смысл в их содержание.

В отличие от цикла обращения к внешней памяти программ состояние младшей части порта P2 после операций обмена с внешней системой ВВ не восстанавливается. После чтения P20—P23 будут находиться в режиме ввода, после записи P20—P23 содержат данные, переданные в канал.

Дальнейшее расширение 4-разрядного пространства ВВ, как и областей памяти или данных, легко выполняется с помощью переключения банков. В этом случае ряд резидентных линий, например P24—P27, выполняют функции дополнительной адресной шины, указывающей на один из банков управляющей памяти, памяти данных или портов ВВ. Возможны различные отображения и наложения внешних областей пространств друг на друга,

формирующие прикладную вычислительную среду с требуемой организацией и конфигурацией. Функциональная совместимость интерфейса расширения памяти с шиной ВМ85А обеспечивает возможность прямого использования стандартных устройств памяти и периферийных БИС семейства ВМ80/ВМ85А.

4.9. Универсальный периферийный адаптер

Однокристальные МК представляют собой специальный класс функционально законченных МС, все ресурсы которых расположены на одном кристалле. Эти автономные устройства образуют стандартные системные ядра для построения компактных и недорогих средств цифровой обработки. Важнейшей областью применения однокристальных МК являются периферийные контроллеры ВВ, входящие в состав более мощных МС.

После промышленного освоения 8-разрядных МП и их улучшенных вариантов быстрыми темпами стала развиваться техника ВВ. Появилось множество однокристальных программируемых адаптеров и контроллеров ПУ, освобождающих главный процессор МС от ряда функций управления ВВ. Было достигнуто значительное повышение производительности МС и снижение сложности ее ПО.

Интеллектуальный интерфейс ВВ не только упрощает задачу программирования (так как многие функции нижнего уровня он выполняет своими собственными средствами), но и обеспечивает дополнительную гибкость системы. Такой интерфейс содержит программно-доступные регистры, управляющие его работой в различных режимах. Несмотря на программируемость все эти устройства спроектированы для выполнения узкоспециализированных задач ВВ. Архитектура однокристальных МК является идеальной средой проектирования действительно универсальных интеллектуальных интерфейсов различного типа.

К недостаткам МК относится отсутствие встроенного системного интерфейса, с помощью которого главный ЦП имел бы возможность взаимодействовать с МК. Системный интерфейс (СА) может быть реализован внешними средствами. Его размещение непосредственно на кристалле МК приводит к новому типу программируемых периферийных микроконтроллеров, подчиненных главному ЦП, к системной магистрали которого они подключены.

Типичным представителем приборов данного класса является универсальный периферийный адаптер (УПА, UPI — Universal Peripheral Interface) 8041 фирмы Intel (рис. 4.16). Прибор заключен в стандартный 40-выводный корпус (рис. 4.17), для его работы требуется один источник питания +5 В.

Устройство имеет организацию типа ВЕ48, которая расширена встроенным СА, удовлетворяющим требованиям шины Microbus на периферийные БИС

Таблица 4.8

A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Операция
0	0	1	0	D←DB, OBF←0
1	0	1	0	D←SW
0	1	0	0	DBB←D, F1←0, IBF←1
1	1	0	0	DBB←D, F1←1, IBF←1
X	1	1	0	Нет операции
X	X	X	1	То же

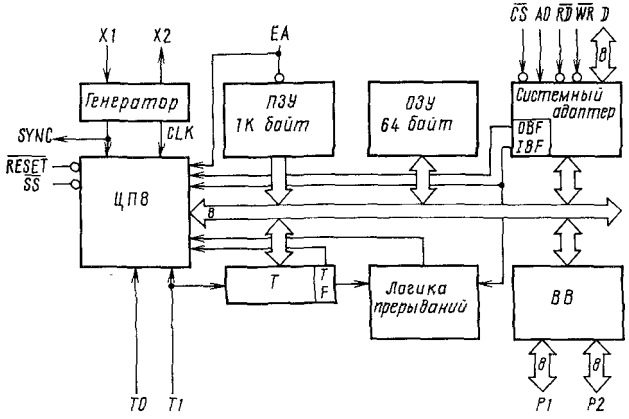


Рис. 4.16. Структурная схема универсального периферийного адаптера

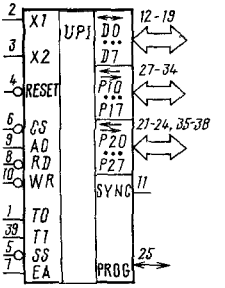


Рис. 4.17. Условное графическое обозначение универсального периферийного адаптера

второго поколения. В состав адаптера входят двунаправленный 8-разрядный буферный регистр слова данных DBB и 4-разрядный регистр слова состояния SW (рис. 4.18). По отношению к центральному процессору УПА является обычным ПУ, связь с которым осуществляется через названные регистры. Операции доступа к ним со стороны шины Microbus приведены в табл. 4.8.

Ввод-вывод данных сопровождается автоматическим сбросом/установкой флажков готовности OBF/IBF, обеспечивающих операцию условного обмена. Установка IBF в 1 вызывает генерацию запроса на прерывание со стартовым адресом 3 (внешнее прерывание INT для BE48). Ввод информации осуществляется как при A0 = 0, так и при A0 = 1, при этом состояние A0 фиксируется флажком F1. Ввод информации по адресу A0 = 0 резервируется для передачи непосредственно данных, по адресу A0 = 1 — для передачи управляющих слов. Флажок F0 доступен только для чтения в SW и носит общий характер.

Для УПА буфер DBB представляет собой порт, заменяющий BUS в BE48. Все флажки SW при этом программно-доступны. Если флажки F0, F1 представляют собой одноименные флажки пользователя BE48, то IBF и OBF — два новых флажка, которые отсутствуют в BE48.

Набор команд УПА совместим с системой BE48 на уровне объектного кода. Однако в наборе отсутствуют команды:
обращения к внешней памяти данных

```
MOVX    A, @Ri      ;A←XSEG(Ri), i=0−1
MOVX    @Ri, A      ;XSEG(Ri)←A, i=0−1
```

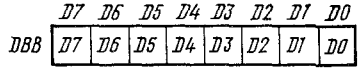


Рис. 4.18. Пространство связи с универсальным периферийным адаптером

выбора банков памяти программ

```
SEL     MB0     ;MB←0
SEL     MB1     ;MB←1
```

тестирования входа $\overline{\text{INT}}$

```
JNI     addr8   ;Если INT=1, то
                ;PC0-7←addr8
```

обслуживания порта BUS

```
INS     A,BUS   ;A←BUS
OUTL    BUS, A  ;BUS←A
ANL     BUS,#data ;BUS←BUS AND data
ORL     BUS,#data ;BUS←BUS OR data
```

и команда

```
ENT0    CLK     ;Разрешение выдачи CLK на
                ;T0
```

что связано с соответствующими изменениями в организации УПА. Введены также четыре новые команды, представленные в табл. 4.9.

Набор команд условного перехода внутри текущей страницы расширен двумя новыми:

```
JNIBF   addr8   ;Если IBF=0,  то
                ;PC0-7←addr8
JOBF    addr8   ;Если OBF=1, то
                ;PC0-7←addr8
```

которые позволяют организовать программное тестирование флажков IBF и OBF состояния буфера шины данных DBB. Такая проверка необходима при построении процедур программно-управляемого ВВ данных. Две другие команды

```
IN      A,DBB   ;A←DBB, IBF←0
OUT     DBB,A   ;DBB←A, OBF←1
```

связывают буфер ДВВ с аккумулятором, обеспечивая программный доступ к данным, переданным через шину данных. Команды обеспечивают согласованное управление флажками IBF и OBF, указывающими на состояние буфера DBB. При чтении буфера флажок IBF устанавливается в 0, а при записи в буфер флажок OBF устанавливается в 1. Так как запрос на внешнее прерывание $\overline{\text{INT}}$ микроконтроллера ВЕ48 в УПА используется для принятия запросов по флагу IBF, то команды

Таблица 4.9

Мнемоника	Число байтов	Число циклов	Код	CY	Описание
JNIBF addr8	2	2	D6	—	Если IBF=0, то PC_{0-7}←addr8
JOBF addr8	2	2	86	—	Если OBF=1, то PC_{0-7}←addr8
IN A,DBB	1	1	22	—	A←DBB, IBF←0
OUT DBB,A	1	1	02	—	DBB←A, OBF←1

```
EN      I      ;Разрешение прерываний по ;INT
DIS     I      ;Запрет прерываний по INT
```

получают другую интерпретацию.

4.10. Базовая организация BE51

Введение в организацию BE51. В состав однокристального МК BE51 (рис. 4.19) входит 8-разрядный ЦП, управляющее ПЗУ, внутреннее ОЗУ данных, 32 линии прямого ВВ, четыре тестируемых входа, канал последовательного ВВ, два или три 16-разрядных таймера/счетчика Т и логика двухуровневой системы прерываний с пятью или шестью источниками запросов [64]. Эти средства образуют резидентную часть МК, размещенную непосредственно на кристалле. Базовая организация предоставляет встроенные средства расширения своих ресурсов, которые предусматривают либо реализацию вне кристалла всей памяти программ, либо расширение памяти, имеющейся внутри кристалла до 64К байт. Имеется возможность подключения дополнительной внешней памяти данных в 64К байт. Дальнейшее расширение ресурсов может быть выполнено только с помощью внешних средств.

Для сокращения ширины физического интерфейса большинство логических линий совмещаются. Так, при обращениях к внешней памяти порт P0 выполняет роль совмещенной шины адреса/данных, а P2 — шины старшей части адреса. Все выводы порта P3 выполняют роль линий управления и специального ВВ.

В архитектуре BE51 и ее модификациях использован стандартный для МК принцип независимости сред для хранения

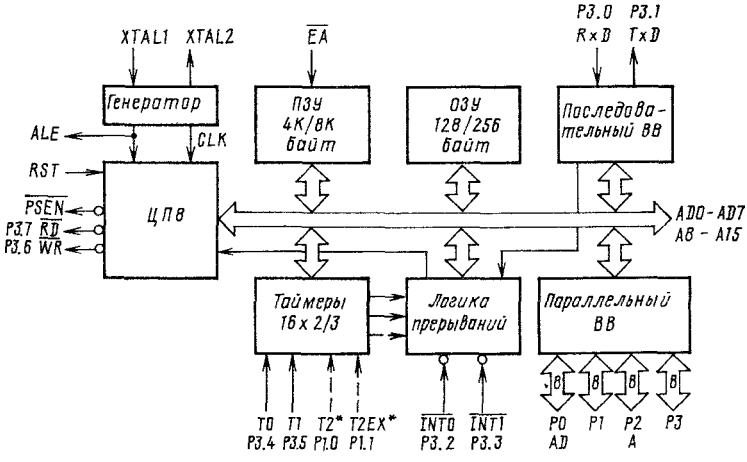


Рис. 4.19. Схема микроконтроллера BE51

программ и данных. Всего же архитектура BE51 включает пять типов пространств, четыре из которых являются областями данных:

RSEG	Пространство регистров
DSEG	Пространство внутренней памяти данных
BSEG	Битовое пространство данных
XSEG	Пространство внешней памяти
CSEG	Пространство программного кода

Однако пространство регистров, за исключением PC, и пространство битов, частично пересекаясь, физически совмещаются с пространством DSEG, образуя единую внутреннюю среду для хранения данных, имеющую достаточно сложную структуру. Принцип наложения нескольких пространств данных друг на друга уже встречался в архитектуре BE48. Это стандартный для однокристальных МС прием, позволяющий одни и те же физические данные рассматривать с разных позиций. В результате выбирается наиболее удобный для конкретного случая способ интерпретации тех или иных данных, в соответствии с которым и организуется доступ к ним.

Набор регистров BE51. Набор программно-доступных регистров процессора BE51 приведен на рис. 4.20. Он является расширением набора регистров BE48 (см. рис. 4.3), что обеспечивает совместимость архитектур BE48 и BE51 снизу вверх. Обе архитектуры относятся к классу аккумуляторных с переключаемыми банками рабочих регистров. Поэтому центральным регистром набора считается 8-разрядный аккумулятор A, выполняющий обычные функции основного арифметического регистра.

Регистр B служит расширением аккумулятора A, необходимым для осуществления операций умножения и деления, причем он является как источником, так и приемником операндов. Во всех других операциях регистр B выполняет функции, определяемые пользователем.

Регистр слова состояния программы кроме флажков, входящих в PSW BE48:

PSW.7	CY	Перенос из старшего разряда АЛУ
PSW.6	AC	Дополнительный перенос из младшей тетрады АЛУ
PSW.5	F0	Флажок пользователя общего назначения

включает также флажки

PSW.2	OV	Признак арифметического переполнения результата
PSW.1	P	Признак четности

Сюда же входит двухразрядное поле RS (Registers Select) выбора одного из четырех возможных банков рабочих регистров. Напомним, что в архитектуре BE48 таких банков было всего два. Флажки признаков результата CY, AC и OV, как правило,

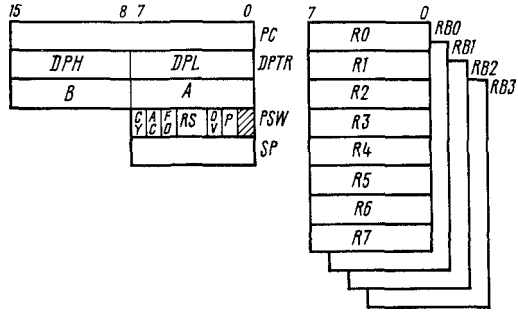


Рис. 4.20. Набор регистров BE51

отражают состояние последней арифметической операции, флажок P — четность содержимого A. Флажок переноса CY является аккумулятором булевого процессора. Функциональное назначение флажка F0 определяется пользователем в конкретной ситуации.

Шестнадцатиразрядный программный счетчик PC управляет последовательностью выполнения команд, хранящихся в программной памяти объемом до 64К байт. Указатель данных DPTR также имеет длину 16 разрядов, каждая его половина может быть адресована независимо от другой. Этот регистр используется в качестве адресного при пересылке констант из памяти программ и доступе к переменным из внешней памяти данных, а также для организации передачи управления.

Указатель стека образует системный стек глубиной до 256 байт. Он всегда содержит адрес последнего байта, занесенного в стек. Стек растет в сторону увеличения содержимого SP.

В МК BE51 предусмотрено четыре банка по восемь рабочих регистров R0—R7 в каждом, переключаемых полем RS слова состояния программы. Регистры выполняют общецелевые функции промежуточного хранения данных. По аналогии с BE48 два регистра R0 и R1 каждого банка реализуют также функции 8-разрядных указателей данных.

Использование наборов рабочих регистров позволяет существенно уменьшить длительность переключения контекстов ЦП, что очень важно для МС реального времени. Следует также отметить, что в BE51 отсутствует ряд ограничений, накладываемых на обработку подпрограмм и процедур обслуживания прерываний, которые свойственны BE48.

При сбросе МК все регистры устанавливаются в исходное состояние. Программный счетчик PC принимает значение 0000H, аккумулятор A—00H; B—00H, PSW—00H, SP—07H и DPTR—0000H. Сброс PC обеспечивает передачу управления по стартовому адресу 0000H, а установка SP в состояние 07H поддерживает совместимость со стеком BE48. Сброс PSW реализует выбор нулевого регистрового банка RB0, что также соответствует архитектуре BE48.

Организация памяти BE51. Пространство внутренней памяти DSEG имеет общий объем 256 байт. Однако организация BE51 предусматривает реализацию только первой его половины (128 байт). В МК 8052 DSEG используется в полном объеме.

Подобно архитектуре BE48 все банки рабочих регистров, а также системный стек в МК BE51 располагаются во внутренней памяти данных и могут рассматриваться как обычные ячейки памяти. Существуют два способа адресации памяти данных МК: прямой (direct) и косвенный (@Ri, i=0−1) через регистры R0, R1 выбранного в данный момент одного из банков RB0—RB3. При прямой адресации доступна только младшая половина адресного пространства внутренней памяти данных (128 байт), при косвенной обеспечивается доступ к любой ее ячейке (256 байт). Введение отсутствующей в BE48 прямой адресации значительно расширило возможности обработки данных МК, в частности появились средства прямого доступа в соседние регистровые банки и стек системы, интерпретируемые как обычные ячейки памяти.

Микроконтроллер BE51 имеет мощную и развитую подсистему ВВ и средства поддержки режима реального времени. Для их управления в МК предусмотрен ряд регистров, которые размещаются во второй половине прямо адресуемого пространства (рис. 4.21), образующей пространство специальных регистров (128 байт). Сюда же включены порты и основные регистры ЦП. Элементы, присутствующие только в модели 8052, отмечены знаком «*», как и на рис. 4.22—4.24.

Центральный процессор BE51 содержит специальную логику, предназначенную для выполнения нескольких однобитовых операций,—булев или одноразрядный процессор для вычисления булевых выражений. В основу булева процессора положен стандартный аккумуляторный принцип организации. В данном случае роль аккумулятора выполняет флажок переноса CY.

Для хранения булевых данных в архитектуре BE51 предусмотрено специальное одноразрядное линейно упорядоченное пространство BSEG объемом 256 байт, которое физически совмещено с байтовым пространством данных DSEG. При этом одна часть пространства BSEG попадает на обычные ячейки памяти DSEG и может рассматриваться как область общего назначения. Обычно она используется для хранения булевых переменных. Другая часть пространства BSEG попадает на ячейки памяти, совмещенные с регистрами МК, что обеспечивает независимый доступ к их отдельным разрядам. В булевом пространстве определена только прямая адресация bit.

На рис. 4.21 в байтах пространства с прямой адресацией, которые размещены в булевом пространстве, указаны диапазоны адресов BSEG, относящихся к их разрядам. Например, старший разряд аккумулятора A, отождествленного с ячейкой пространства памяти с прямой адресацией под адресом 0E0H, имеет адрес

а)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	R0	R1	R2	R3	R4	R5	R6	R7	R0′	R1′	R2′	R3′	R4′	R5′	R6′	R7′
1	R0″	R1″	R2″	R3″	R4″	R5″	R6″	R7″	R0‴	R1‴	R2‴	R3‴	R4‴	R5‴	R6‴	R7‴
2	00-07	08-0F	10-17	18-1F	20-27	28-2F	30-37	38-3F	40-47	48-4F	50-57	58-5F	60-67	68-6F	70-77	78-7F
3																
4																
5																
6																
7																

б)

	8	9	A	B	C	D	E	F
8	P0 80-87	SP	DPL	DPH			PCON	TCON 88-8F
9	P1 90-97							SCON 98-9F
A	P2 A0-A7							IE A8-AF
B	P3 B0-B7							IP B8-BF
C	C0-C7							T2* C2H C8-CF
D	PSW D0-D7							D8-DF
E	ACC E0-E7							E8-EF
F	B F0-F7							F8-FF

	9	A	B	C	D	E	F
8	SBUF						
9							
A							
B							
C		RC* AP2L	RC* AP2H	TL2*	TH2*		
D							
E							
F							

Рис. 4.21. Прямо адресуемая часть внутренней памяти данных (а) и память специальных регистров (б)

пространства BSEG, равный 0E7H. Двойная, а в ряде случаев тройная интерпретация отдельных данных дает возможность программисту выбирать тип доступа, наиболее подходящий для конкретного прикладного случая программирования. Все это позволяет повысить эффективность программного кода, его длину и скорость исполнения.

Пространство внешней памяти XSEG имеет объем 64К байт и реализуется внешними по отношению к МК средствами. Существует единственная команда, поддерживающая связь с данным пространством:

```
MOVX    A,@Ri          ;A←XSEG(P2:Ri),
                       ;i=0−1
MOVX    A,@DPTR        ;A←XSEG(DPTR)
MOVX    @Ri, A         ;XSEG(P2:Ri)←A,
                       ;i=0−1
MOVX    @DPTR, A       ;XSEG(DPTR)←A
```

В команде используются два типа адресации: косвенная регистровая по DPTR и страничная с номером страницы в P2 и смещением в R0, R1. Это дает право рассматривать организацию внешней памяти BE51 как линейную область или как область со страничной структурой.

Память программ CSEG адресуется 16-разрядным счетчиком PC и, следовательно, может иметь объем до 64К байт. Часть этой памяти (4К/8К байт с младшими адресами) может быть расположена на кристалле в виде программируемого маской ПЗУ или репрограммируемого электрически УСППЗУ. Она образует внутреннюю память программ. Оставшаяся часть, реализуемая внешними средствами вне кристалла, называется внешней памятью программ. С точки зрения программиста как внутренняя, так и внешняя память представляет собой единое пространство CSEG с равными правами доступа.

Среди особых точек пространства CSEG следует отметить

RESET	0000H	Стартовый адрес при сбросе системы
EXT10	0003H	Внешнее прерывание 0
TIMER0	000BH	Прерывание таймера/счетчика 0
EXTI1	0013H	Внешнее прерывание 1
TIMER1	001BH	Прерывание таймера/счетчика 1
SINT	0023H	Прерывание последовательного порта
TIMER2	002BH	Прерывание таймера/счетчика 2 (только для 8052)

Пространство CSEG — однородное линейное пространство, в котором определены два основных способа передачи управления: абсолютный addr16 и относительный с помощью 8-разрядного смещения rel со знаком. Однако для некоторых команд оно представляет набор 2К-байтовых страниц:

```
ACALL   addr11      ;+(SP)←PC, PC0-10←
                    ←addr11
AJMP    addr11      ;PC0-10←addr11
```

Кроме того, предусмотрен переход по смещению относительно базы DPTR:

```
JMP     @A+DPTR     ;PC←DPTR+A
```

Эти команды введены для поддержки совместимости с архитектурой BE48. Специальная команда пересылки

```
MOVC    A, @A+DPTR  ;A←CSEG(A+DPTR)
MOVC    A, @A+PC    ;A←CSEG(A+PC)
```

позволяет использовать содержимое программной памяти в качестве констант, доступных для чтения.

В МК BE51 имеется возможность совмещения внешней части CSEG с пространством XSEG. Такое совмещение поможет распространить на область CSEG операции и способы доступа к XSEG, в частности станет осуществимой операция записи, что может быть использовано при загрузке программ из внешней памяти.

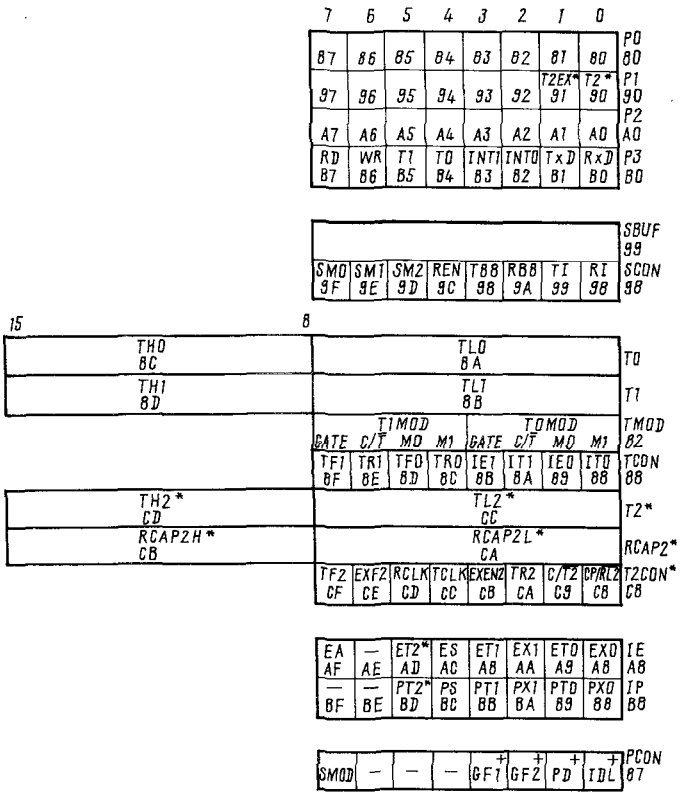


Рис. 4.22. Специальные регистры BE51

4.11. Периферийные средства BE51

Порты ввода-вывода. Подсистема ВВ микроконтроллера BE51 размещается непосредственно на кристалле. Для ВВ данных и управления процессом их передачи в состав МК введен ряд портов данных и регистров управления/состояния, совокупность которых образует набор специальных регистров (рис. 4.22).

Физическая система ВВ микроконтроллера BE51 состоит из четырех двунаправленных 8-разрядных портов P0—P3. Все порты ВВ отображены в пространстве внутренней памяти DSEG по адресам 80H, 90H, 0A0H, 0B0H и ничем не отличаются от обычных ячеек памяти. Отказ от изолированного пространства ВВ, используемого в архитектуре BE48, позволил увеличить вычислительную эффективность МС интенсивного ВВ, привел к более регулярной структуре набора команд BE51. Если ранее для увеличения возможностей арифметическо-логической обработки данных ВВ было необходимо вводить ряд новых команд:

```
ANL     port,A          ;port←port AND A
ORL     port,A          ;port←port OR A
```

то теперь любая команда с операндом из внутренней памяти может быть применена к содержимому портов P0—P3.

Кроме того, порты P0—P3 совмещены с битовым пространством BSEG, что обеспечивает доступ к отдельным его разрядам независимо от других. Порты P0—P3 занимают адреса BSEG: 80H—87H, 90H—97H, 0A0H—0A7H и 0B0H—0B7H соответственно.

Расширение пространства ВВ может быть выполнено за счет области XSEG или другими средствами с помощью самих портов P0—P3 и SBUF.

При обращении к внешней памяти программ или данных порты P0, P2 выполняют функции системных шин AD и AB соответственно. Младший байт адреса и данные передаются через P0 в мультиплексном режиме: сначала выводится адрес, а затем для передачи данных используется порт. Старший байт адреса формируется на P2. Линии порта P3 реализуют управление циклами обмена и другие специальные функции аппаратного уровня:

P3.0	R×D	Вход приемника последовательного канала
P3.1	T×D	Выход передатчика последовательного канала
P3.2	$\overline{\text{INT0}}$	Вход запроса на прерывание 0
P3.3	$\overline{\text{INT1}}$	Вход запроса на прерывание 1
P3.4	T0	Внешний вход таймера/счетчика 0
P3.5	T1	Внешний вход таймера/счетчика 1
P3.6	$\overline{\text{WR}}$	Строб записи в XSEG
P3.7	$\overline{\text{RD}}$	Строб чтения XSEG

В случае МК 8052 две линии P1 также выполняют специальную функцию

P1.0	T2	Внешний вход таймера/счетчика 2
P1.1	T2EX	Вход управления автоматической перезагрузкой таймера/счетчика 2

Порты P0—P3 имеют организацию, во многом схожую с организацией портов ВЕ48 (см. рис. 4.6). Отличие состоит лишь в выходном буфере, который значительно модифицирован. Так, в портах P0 и P2 в выходном буфере предусмотрены ключи, соединяющие их выводы с внутренними шинами AD и A соответственно. Поэтому при обращении к внешним средствам содержимое выходного регистра порта P2 не изменяется, а содержимое P0 становится равным 0FFH. Буферные каскады

порта P3 построены так, что для выполнения специальной операции соответствующий разряд выходного регистра P3 должен быть установлен в 1. Иначе на выходе будет 0.

Совмещение специальных функций с портом P3 позволило расширить число портов ВВ с трех для ВЕ48 до четырех для ВЕ51 при тех же ограничениях на ширину физического интерфейса. Однако это усложнило выходной буфер P3.

Порты P1—P3 имеют встроенную нагрузку, тогда как для порта P0, выполненного по схеме с открытым коллектором, требуется внешняя нагрузка. Каждый вывод портов P0—P3 может быть использован в качестве выходного независимо от других. Для перевода какого-либо вывода в режим входа в соответствующий разряд выходного регистра должна быть записана 1. При сбросе МК состояние всех портов устанавливается равным 0FFH.

Следует отметить, что в отличие от ВЕ48 в МК ВЕ51 все 32 вывода портов P0—P3 тестируются индивидуально с помощью условных команд ветвления:

```
JB      bit, rel    ;Если BSEG(bit)=1, то
                    ;SJMP rel
JNB     bit, rel    ;Если BSEG(bit)=0, то
                    ;SJMP rel
JBC     bit, rel    ;Если BSEG(bit)=1, то
                    ;SJMP rel
                    ;и BSEG(bit)←0
```

Последняя команда после тестирования всегда на выводе порта, как и в любом другом разряде из пространства BSEG, устанавливает 0.

Последовательный канал связи. В состав ВЕ51/8052 входит дуплексный канал последовательной связи с буферизацией, который может быть запрограммирован для работы в одном из четырех режимов:

режим 0—синхронный последовательный ВВ со скоростью OSC/12;

режим 1—асинхронный с 10-битовым кадром и переменной скоростью передачи;

режим 2—асинхронный с 11-битовым кадром и фиксированной скоростью передачи OSC/32 или OSC/64;

режим 3—асинхронный с 11-битовым кадром и переменной скоростью передачи.

Входные и выходные данные хранятся в буферном регистре SBUF с адресом 99H. Управление работой приемопередатчиков осуществляется через слово управления и состояния SCON, расположенное в регистре по адресу 98H:

SCON.0	RI	Флаг прерывания приемника
SCON.1	TI	Флаг прерывания передатчика

SCON.2	RB8	Восьмой бит приемника в режимах 2 и 3. В режиме 1, если SM2=0, то отображает стоп-бит. В режиме 0 не используется
SCON.3	TB8	Восьмой бит передатчика в режимах 2 и 3
SCON.4	REN	Разрешение приема
SCON.5	SM2	Запрещение приема кадров с нулевым восьмым битом данных. В режиме 0 должен быть сброшен
SCON.6	SM1	Младший разряд для кодирования номера режима
SCON.7	SM0	Старший разряд для кодирования номера режима:

SM0	SM1	Режим	SM0	SM1	Режим
0	0	0	1	0	2
0	1	1	1	1	3

Таймеры/счетчики. К стандартным средствам поддержки режима реального времени относятся таймеры/счетчики и подсистема прерываний. Если таймеры необходимы для организации системных меток реального времени и отработки временных интервалов, то подсистема прерываний обеспечивает своевременную реакцию МК на асинхронные события, происходящие как внутри МС, так и вне ее.

В состав BE51 входят два 16-разрядных таймера/счетчика CT0, CT1. Еще один (CT2) добавлен в архитектуре 8052. Состояние таймеров/счетчиков отражается программно-доступными регистровыми парами (TH0, TL0), (TH1, TL1) и (TH2, TL2) соответственно, размещенными в пространстве DSEG по адресам (8CH, 8AH), (8DH, 8BH) и (OCDH, OCCH).

Таймеры/счетчики CT0—CT2 могут быть запрограммированы для работы либо в качестве таймера, либо в качестве счетчика. Функция таймера состоит в счете числа машинных циклов, следующих с частотой OSC/12. Функция счетчика заключается в отслеживании числа переходов из 1 в 0 на соответствующих входах T0, T1, T2.

Управление режимом работы CT0, CT1 осуществляет регистр TMOD (Timer/Counter Mode), который расположен по адресу 89H. Регистр разбит на два 4-разрядных подрегистра T0MOD и T1MOD, которые ответственны за управление CT0 и CT1 соответственно:

TMOD.0	M0	Младший бит поля управления режимом CT0
TMOD.1	M1	Старший бит поля управления режимом CT0:

M0	M1	Режим	M0	M1	Режим
0	0	0	1	0	2
0	1	1	1	1	3

TMOD.2	C/$\overline{T}$N	Выбор функции таймера или счетчика CT0. При C/$\overline{T}$=0 выбирается функция таймера, в противном случае счетчика
TMOD.3	GATE	Флажок управления работой CT0. При GATE=1 работа разрешается, если $\overline{INT0}$=1 и TR0=1 (см. TCON). При GATE=0 работа счетчика зависит только от состояния TRO

TMOD.4	M0	То же, но для CT1
TMOD.5	M1	То же, но для CT1
TMOD.6	C/$\overline{T}$	То же, но для CT1
TMOD.7	GATE	То же, но для CT1

За управление CT0, CT.1 также ответствен регистр TCON (Timer/Counter Control), расположенный по адресу 88H:

TCON.0	IT0	Управление типом входа $\overline{INT0}$. При IT0=0 программируется динамический по срезу тип входа, в противном случае — статический
TCON.1	IE0	Флажок запроса прерывания $\overline{INT0}$ при динамическом входе. При подтверждении прерывания сбрасывается
TCON.2	IT1	То же, что и IT0, но для $\overline{INT1}$
TCON.3	IE1	То же, что и IE0, но для $\overline{INT1}$
TCON.4	TR0	Флажок программного запуска/останова CT0
TCON.5	TF0	Флажок переполнения CT0, который вызывает запрос прерывания. При подтверждении прерывания сбрасывается
TCON.6	TR1	То же, что и TR0, но для CT1
TCON.7	TF1	То же, что и TF0, но для CT1

Младшая половина регистра используется для управления входами запроса на прерывания $\overline{INT0}$ и $\overline{INT1}$, старшая — для управления непосредственно CT0 и CT1.

Для управления работой CT2 служит регистр T2CON (Timer/Counter 2 Control), расположенный по адресу 0C8H:

T2CON.0	CP/$\overline{RL2}$	Флажок захвата/автозагрузки CT2. При CP/$\overline{RL2}$=1 захват текущего состояния CT2 осуществляется по срезу T2EX, если EXEN2=1. При CP/$\overline{RL2}$=0 разрешается автозагрузка CT2, если он переполнен или по срезу T2EX, если EXEN2=1. Этот флажок не принимается во внимание при RCLK=1 или TCLK=1, т. е. когда CT2 работает в качестве генератора скорости
T2CON.1	C/$\overline{T2}$	Управление функцией таймера (C/$\overline{T2}$=0) или счетчика (C/$\overline{T2}$=1)
T2CON.2	TR2	Программный запуск CT2 при TR2=1 и останов в противном случае
T2CON.3	EXEN2	Флажок разрешения динамическому по срезу входу T2EX осуществлять захват/автозагрузку CT2, если он не работает в режиме генератора скорости последовательного порта. Функция активируется при EXEN2=1

T2CON.4	TCLK	При TCLK=1 CT2 используется передатчиком последовательного канала, работающего в режиме 1 или 3, в качестве генератора скорости передачи. При TCLK=0 генератором служит CT1
T2CON.5	RCLK	То же, что и TCLK, но для приемника последовательного канала
T2CON.6	EXF2	Флажок выполнения захвата/автозагрузки, вызванного изменением состояния на T2EX при EXEN2=1. При разрешенном прерывании вызывает запрос. Сбрасывается только программным способом
T2CON.7	TF2	Флажок переполнения CT2. Также вызывает запрос на прерывание. Функция установки подавляется при TCLK=1 или RCLK=1

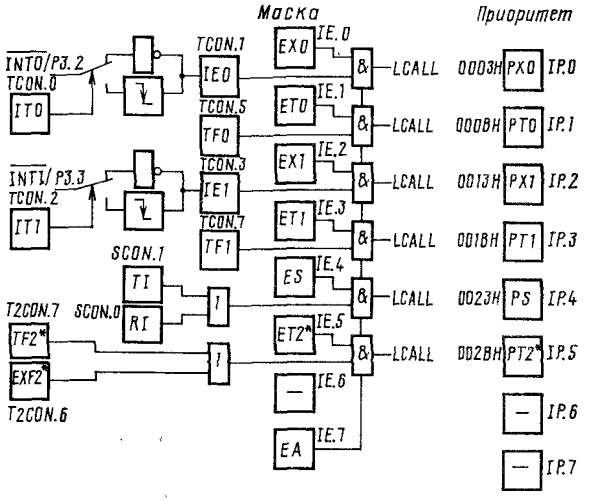
При захвате текущего состояния или при повторной автозагрузке в управлении работой CT2 участвует также 16-разрядный регистр RCAP2 (Reload/Capture Timer/Counter 2), расположенный в двух соседних 8-разрядных ячейках с адресами 0CAH (RCAP2L) и 0CBH (RCAP2H).

Система прерываний. Архитектура BE51 поддерживает двухуровневую приоритетную систему прерываний с пятью (или шестью в случае 8052) источниками запросов на обслуживание, имеющими фиксированные векторы прерываний. Программное управление системой осуществляется через два 8-разрядных регистра: IP (Interrupt Priopity) — регистр приоритета прерываний и IE (Interrupt Enable) — регистр разрешения прерываний. Условная схема системы прерываний представлена на рис. 4.23.

Для приема внешних запросов на прерывание служат линии INT0 и INT1, которые могут быть запрограммированы на срабатывание как по переходу из одного состояния в другое, так и по уровню входного сигнала независимо друг от друга. Управление типом входа осуществляется флажками IT0/TCON.0 (Interrupt Type 0) и IT1/TCON.2. При ITi=1 устанавливается режим срабатывания по переходу из 1 в 0, в противном случае — по напряжению низкого уровня на входе INTi, i=0—1. Запросы на прерывание от внешних источников $\overline{\text{INT0}}$ или $\overline{\text{INT1}}$ устанавливают флажки IE0/TCON.1 (Interrupt Edge 0) и IE1/TCON.3. В случае работы по переходу эти флажки сбрасываются автоматически при входе в соответствующую процедуру обслуживания прерывания. В случае режима работы по уровню флажки отслеживают состояние сигналов на входных линиях $\overline{\text{INT0}}$ и $\overline{\text{INT1}}$, повторяя все их изменения.

Источниками внутренних запросов могут быть: флажок TF0/TCON.5 (Timer Flag 0) — признак переполнения CT0, флажок TF1/TCON.7 — признак переполнения CT1, а также флажок TI/SCON.1 (Transmit Interrupt) или флажок RI/SCON.0 (Recive

Рис. 4.23. Схема системы прерываний BE51



Interrupt). Еще одним источником внутреннего прерывания в случае 8052 могут быть флажки TF2 — признак переполнения CT2 или EXF2, входящие в состав T2CON. Флажки внутренних запросов TF0 и TF1 очищаются автоматически при переходе к соответствующей процедуре обслуживания, тогда как состояние флажков TI, RI, а также TF2, EXF2 не изменяется. Процедурой обслуживания прерываний эта информация используется для уточнения источника запроса по методу поллинга, а затем сбрасывается с помощью подходящей для этого команды.

Все флажки, которые фиксируют запросы на прерывания, могут быть установлены программным способом, что дает возможность реализовать вызов соответствующих процедур обслуживания непосредственно из программ. Этому способствует дублирование входов $\overline{\text{INT0}}$ и $\overline{\text{INT1}}$ программно-управляемыми флажками IE0 и IE1. Каждый из пяти (или шести) источников прерываний может быть замаскирован независимо от других с помощью регистра маски IE:

IE.0	EX0	Маска IE0 или $\overline{\text{INT0}}$
IE.1	ET0	Маска TF0
IE.2	EX1	Маска IE1 или $\overline{\text{INT1}}$
IE.3	ET1	Маска TF1
IE.4	ES	Маска TI+RI
IE.5	ET2	Маска TF2+EXF2
IE.6	—	Не используется
IE.7	EA	Общее разрешение прерываний

Старший разряд регистра используется для блокировки или разрешения работы всей системы прерываний.

Каждому из пяти (или шести) источников прерываний присваивается один из двух уровней приоритета сбросом или установкой соответствующего разряда в регистре приоритета IP:

IP.0	PX0	Приоритет IE0 или $\overline{\text{INT0}}$
IP.1	PT0	Приоритет TF0
IP.2	PX1	Приоритет IE1 или $\overline{\text{INT1}}$
IP.3	PT1	Приоритет TF1
IP.4	PS	Приоритет TI+RI
IP.5	PT2	Приоритет TF2+EXF2
IP.6	—	Не используется
IP.7	—	Не используется

Установка разряда в 1 соответствует высокому приоритету, в 0 — низкому. Процедура обслуживания низкоприоритетного уровня может быть прервана запросом более высокого уровня. Высокоуровневое прерывание не может быть остановлено никаким другим. Для решения конфликтных ситуаций одновременного появления нескольких запросов одного уровня используется схема вторичного арбитража, устанавливающая строгое отношение предпочтения между всеми источниками запросов:

Источник	Приоритет
IE0	0 (высший)
TF0	1
IE1	2
TF1	3
RI+TI	4
TF2+EXF2	5 (низший)

Запросы на прерывания могут быть приняты к обслуживанию в конце каждого командного цикла, за исключением исполнения RETI или команды с любым видом доступа к регистрам IE и IP. Данное исключение гарантирует выполнение хотя бы еще одной инструкции после команд обращения к IE или IP и RETI, прежде чем возникнет новое прерывание программы. При фиксации запроса аппаратура МК генерирует команду

LCALL vect

обеспечивая переход к стартовому адресу vect соответствующей процедуры обслуживания. С каждым источником связан свой стартовый адрес (вектор прерывания):

EXTI0	IE0	0003H
TIMER0	TF0	000BH
EXTI1	TE1	0013H
TIMER1	TF1	001BH
SINT	RI+TI	0023H
TIMER2	TF2+EXF2	002BH

В некоторых случаях эта команда одновременно с переходом к стартовому адресу сбрасывает флажок, вызвавший данное прерывание. При переходе к процедуре обслуживания текущее состояние PC загружается в стек, обеспечивая возврат по команде RETI, заканчивающей каждую процедуру обслуживания. Эта команда отличается от обычной инструкции возврата RET тем, что сообщает системе прерываний об окончании текущей процедуры обслуживания, что необходимо для управления двухуровневой системой приоритетов.

В состав BE51 входит еще один общецелевой регистр управления PCON (Power Control), расположенный по адресу 87H. Состав этого регистра зависит от технологии изготовления МК. При n-МОП-технологии PCON содержит всего лишь один флажок SMOD. Полный состав флажков можно встретить только в КМОП-вариантах:

PCON.0	IDL	Установка бита активизирует режим свободного состояния
PCON.1	PD	Установка бита активизирует режим пониженной мощности
PCON.2	GF0	Общецелевой флажок
PCON.3	GF1	Общецелевой флажок
PCON.4	—	Не используется
PCON.5	—	Не используется
PCON.6	—	Не используется
PCON.7	SMOD	Активизирует двойную скорость приема/передачи последовательного канала, работающего в режимах 1, 2 и 3

Все управляющие регистры кроме PCON совмещены с пространством BSEG, что обеспечивает доступ к отдельным их разрядам с помощью команд булевой группы. При сбросе МК управляющие регистры принимают нулевые значения за исключением следующих:

IP (BE51)	XXX00000B	SBUF	XXXXXXXXB
IP (BE52)	XX000000B	PCON (n-МОП)	0XXXXXXXB
IE (BE51)	0XX0000B	PCON (КМОП)	0XXX0000B
IE (BE52)	0X000000B		

4.12. Система команд BE51

Система насчитывает 111 команд, из них 49 однобайтовых, 45 двухбайтовых и 17 трехбайтовых. Появление трехбайтовых команд связано с расширением объемов CSEG и XSEG, а также вводом прямой адресации в DSEG. Все команды выполняются за один или два машинных цикла (12 тактов ALE) за исключением команд MUL и DIV, которые требуют четыре цикла. Большинст-

Таблица 4.10

Мнемоника	CY	OV	AC	Мнемоника	CY	OV	AC
ADD	+	+	+	CLR C	0	–	–
ADDC	+	+	+	CPL C	+	–	–
SUBB	+	+	+	ANL C,bit	+	–	–
MUL	0	+	–	ANL C,/bit	+	–	–
DIV	0	+	–	ORL C,bit	+	–	–
DA	+	–	–	ORL C,/bit	+	–	–
RRC	+	–	–	MOV C,bit	+	–	–
RLC	+	–	–	CJNE	+	–	–
SET C	1	–	–				

Таблица 4.11

Мнемоника	Число циклов	Код	Флажки	Описание
			C A F R R O P I C 0 S S V	
MOV A,Rn	1	E8—EF	– – – – – – +	A ← Rn, n=0–7
MOV A,direct	1	E5	– – – – – – +	A ← (direct)
MOV A,@Ri	1	E6—E7	– – – – – – +	A ← (Ri), i=0–1
MOV A,#data	1	74	– – – – – – +	A ← data
MOV Rn,A	1	F8—FF	– – – – – – –	Rn ← A, n=0–7
MOV Rn,direct	2	A8—AF	– – – – – – –	Rn ← (direct), n=0–7
MOV Rn,#data	1	78—7F	– – – – – – –	Rn ← data, n=0–7
MOV direct,A	1	F5	– – – – – – –	(direct) ← A
MOV direct,Rn	2	88—8F	– – – – – – –	(direct) ← Rn, n=0–7
MOV direct,direct	2	85	– – – – – – –	(direct) ← (direct)
MOV direct,@Ri	2	86—87	– – – – – – –	(direct) ← (Ri), i=0–1
MOV direct,#data	2	75	– – – – – – –	(direct) ← data
MOV @Ri,A	1	F6—F7	– – – – – – –	(Ri) ← A, i=0–1
MOV @Ri,direct	2	A6—A7	– – – – – – –	(Ri) ← (direct), i=0–1
MOV @Ri,#data	1	76—77	– – – – – – –	(Ri) ← data, i=0–1
MOV DPTR,#data 16	2	90	– – – – – – –	DPTR ← data16
MOVC A,@A+DPTR	2	93	– – – – – – +	A ← CSEG (A+DPTR)
MOVC A,@A+PC	2	83	– – – – – – +	A ← CSEG (A+PC)
MOVX A,@Ri	2	E2—E3	– – – – – – +	A ← XSEG (P2:Ri), i=0–1
MOVX A,@DPTR	2	E0	– – – – – – +	A ← XSEG (DPTR)
MOVX @Ri,A	2	F2—F3	– – – – – – –	XSEG (P2:Ri) ← A, i=0–1
MOVX @DPTR,A	2	F0	– – – – – – –	XSEG (DPTR) ← A
PUSH direct	2	C0	– – – – – – –	+ (SP) ← (direct)
POP direct	2	D0	± ± ± ± ± ± ±	(direct) ← (SP)
XCH A,Rn	1	C8—CF	– – – – – – +	A↔ Rn, n=0–7
XCH A,direct	1	C5	– – – – – – +	A ↔ (direct)
XCH A,@Ri	1	C6—C7	– – – – – – +	A ↔ (Ri), i=0–1
XCHD A,@Ri	1	D6—D7	– – – – – – +	$A_{0-3} \leftrightarrow (Ri)_{0-3}$, i=0–1

во двухбайтовых команд одноцикловые, а все трехбайтовые команды — двухцикловые. Это объясняется тем, что за один

машинный цикл в BE51 может вводиться до двух байтов программного кода.

Все множество команд BE51 удобно разбить на пять групп: пересылки (28), логической обработки (25), арифметической обработки (24), передачи управления (17), булевого процессора (17).

В процессе исполнения команды влияют на ряд флажков-признаков результата, входящих в состав PSW (табл. 4.10). Признак P устанавливается всякий раз, когда приемником результата служит аккумулятор, включая операции пересылки.

Группа команд пересылки (табл. 4.11) содержит команды MOV (пересылки данных между DSEG и RSEG), MOVC (между CSEG и A), MOVX (между XSEG и A или RSEG), команды обращения к стеку PUSH и POP, а также две команды обмена XCH и XCHD. Наиболее емкой инструкцией является команда MOV, которая использует четыре способа адресации: регистровый (A, Rn, DPTR), прямой (direct), косвенный (@Ri) и непосредственный (#data, #data 16). Для указания приемника служат три способа адресации (кроме непосредственного), для указания источника все четыре. Трехбайтовая команда MOV direct, direct обеспечивает пересылку между двумя любыми ячейками памяти, включая регистры МК. Тем не менее для обмена с регистрами предусмотрены специальные двух- и однобайтовый форматы:

```
MOV     Rn,direct
MOV     direct,Rn
MOV     A,Rn
MOV     Rn,A
```

Их использование позволяет существенно сократить длину программного кода. Специальная команда

```
MOV     DPTR,#data16
```

позволяет загрузить 16-разрядный указатель DPTR значением data16.

При выполнении команды MOVC считывания данных из программной памяти могут быть применены два способа адресации: по базе DPTR и относительный. В обоих случаях целое без знака смещение (индекс) хранится в аккумуляторе. Приемником результата также служит аккумулятор. Команда дает возможность выполнять быструю перекодировку по таблицам, осуществлять доступ к массивам памяти.

Обращение к внешней памяти осуществляется с помощью команды MOVX. Обмен производится по байтам между аккумулятором и ячейкой внешней памяти данных. Ячейка XSEG может быть адресована двумя способами: косвенно через 16-разрядный указатель DPTR и странично косвенно через 8-разрядный указатель Ri, i=0−1. В последнем случае регистром страниц служит выходной регистр P2.

Таблица 4.12

Мнемоника	Число циклов	Код	Флажки C A F R R O P Y C 0 S S V	Описание
ANL A,Rn	1	58—5F	– – – – – – +	A ← A AND Rn, n=0–7
ANL A,direct	1	55	– – – – – – +	A ← A AND (direct)
ANL A,@Ri	1	56—57	– – – – – – +	A ← A AND (Ri), i=0–1
ANL A,#data	1	54	– – – – – – +	A ← A AND data
ANL direct A	1	52	– – – – – – –	(direct) ← (direct) AND A
ANL direct,#data	2	53	– – – – – – –	(direct) ← (direct) AND data
ORL A,Rn	1	48—4F	– – – – – – +	A ← A OR Rn, n=0–7
ORL A,direct	1	45	– – – – – – +	A ← A OR (direct)
ORL A,@Ri	1	46—47	– – – – – – +	A ← A OR (Ri), i=0–1
ORL A,#data	1	44	– – – – – – +	A ← A OR data
ORL direct A	1	42	– – – – – – –	(direct) ← (direct) OR A
ORL direct,#data	2	43	– – – – – – –	(direct) ← (direct) OR data
XRL A,Rn	1	68—6F	– – – – – – +	A ← A XOR Rn, n=0–7
XRL A,direct	1	65	– – – – – – +	A ← A XOR (direct)
XRL A,@Ri	1	66—67	– – – – – – +	A ← A XOR (Ri), i=0–1
XRL A,#data	1	64	– – – – – – +	A ← A XOR data
XRL direct,A	1	62	– – – – – – –	(direct) ← (direct) XOR A
XRL direct,#data	2	63	– – – – – – –	(direct) ← (direct) XOR data
CLR A	1	E4	– – – – – – +	A ← 0
CPL A	1	F4	– – – – – – –	A ← NOT A
RL A	1	23	– – – – – – –	$A_7 \leftarrow A_6 \leftarrow \ldots \leftarrow A_0 \leftarrow A_7$
RLC A	1	33	+ – – – – – +	CY ← $A_7 \leftarrow A_6 \leftarrow \ldots \leftarrow A_0 \leftarrow$ CY
RR A	1	03	– – – – – – –	$A_0 \leftarrow A_1 \leftarrow \ldots \leftarrow A_7 \leftarrow A_0$
RRC A	1	13	+ – – – – – +	CY ← $A_0 \leftarrow A_1 \leftarrow \ldots \leftarrow A_7 \leftarrow$ CY
SWAP A	1	C4	– – – – – – –	$A_{4-7} \leftrightarrow A_{0-3}$

Операции PUSH и POP используют только прямой способ адресации, однако это не мешает им манипулировать содержимым регистров, которые рассматриваются как ячейки памяти.

Две типовые операции обмена XCH и XCHD дополняют одностороннюю пересылку двусторонней. При выполнении операции XCH обмену подлежат байты, при XCHD — младшие тетрады байтовых операндов.

Приведенная в табл. 4.12 группа команд логических операций содержит три типовые двухместные операции: ANL — логическое И, ORL — логическое ИЛИ и XRL — логическое исключающее ИЛИ. Источником первого операнда и одновременно приемником результата служит либо аккумулятор A, либо прямо адресуемая ячейка памяти. Второй операнд задается одним из четырех основных методов адресации. В состав группы входит также ряд одноместных операций: CLR — очистки, CPL — логического дополнения (инверсия), а также RL, RLC, RR и RRC — операции

Таблица 4.13

Мнемоника	Число циклов	Код	Флажки							Описание
			C Y	A C	F 0	R S	R S	O V	P	
ADD A,Rn	1	28—2F	+	+	—	—	—	+	+	A ← A+Rn, n=0—7
ADD A,direct	1	25	+	+	—	—	—	+	+	A ← A+(direct)
ADD A,@Ri	1	26—27	+	+	—	—	—	+	+	A ← A+(Ri), i=0—1
ADD A,#data	1	24	+	+	—	—	—	+	+	A ← A+data
ADDC A,Rn	1	38—3F	+	+	—	—	—	+	+	A ← A+Rn+CY, n=0—7
ADDC A,direct	1	35	+	+	—	—	—	+	+	A ← A+(direct)+CY
ADDC A,@Ri	1	36—37	+	+	—	—	—	+	+	A ← A+(Ri)+CY, i=0—1
ADDC A,#data	1	34	+	+	—	—	—	+	+	A ← A+data+CY
SUBB A,Rn	1	98—9F	+	+	—	—	—	+	+	A ← A—Rn—CY, n=0—7
SUBB A,direct	1	95	+	+	—	—	—	+	+	A ← A—(direct)—CY
SUBB A,@Ri	1	96—97	+	+	—	—	—	+	+	A ← A—(Ri)—CY, i=0—1
SUBB A,#data	1	94	+	+	—	—	—	+	+	A ← A—data—CY
INC A	1	04	+	+	—	—	—	+	+	A ← A+1
INC Rn	1	08—0F	—	—	—	—	—	—	—	Rn ← Rn+1, n=0—7
INC direct	1	05	—	—	—	—	—	—	—	(direct) ← (direct)+1
INC @Ri	1	06—07	—	—	—	—	—	—	—	(Ri) ← (Ri)+1, i=0—1
INC DPTR	2	A3	—	—	—	—	—	—	—	DPTR ← DPTR+1
DEC A	1	14	—	—	—	—	—	—	+	A ← A—1
DEC Rn	1	18—1F	—	—	—	—	—	—	—	Rn ← Rn—1, n=0—7
DEC direct	1	15	—	—	—	—	—	—	—	(direct) ← (direct)—1
DEC @Ri	1	16—17	—	—	—	—	—	—	—	(Ri) ← (Ri)—1, i=0—1
MUL AB	4	A4	0	—	—	—	—	+	+	BA ← A×B
DIV AB	4	84	0	—	—	—	—	+	+	AB ← A/B
DA A	1	4	+	—	—	—	—	—	+	A ← 2/10-коррекция A

циклического и расширенного сдвигов вправо и влево. Все операции манипулируют содержимым только аккумулятора A. Сюда же включена операция обмена тетрад в аккумуляторе SWAP, которая может интерпретироваться как циклический сдвиг байта на четыре разряда.

В состав группы команд арифметической обработки (табл. 4.13) входят: операция сложения ADD, сложения с учетом переноса ADDC, вычитания с учетом займа SUBB, увеличения и уменьшения на единицу INC и DEC, десятичная коррекция сложения в 2/10-коде упакованного формата DA, умножение MUL и деление DIU. Операции выполняются над беззнаковыми целыми числами.

В операциях сложения и вычитания первым операндом и приемником результата служит аккумулятор. В качестве второго операнда выступает либо рабочий регистр Rn, n=0—7, выбранного регистрового банка, либо ячейка памяти данных, адресуемая прямо direct или косвенно @Ri, i=0—1, либо непосредственные данные #data. Операции INC и DEC применимы к аккумулятору,

Таблица 4.14

Мнемоника	Число циклов	Код	Флажки	Описание
			C A F R R O P Y C 0 S S V	
ACALL addr11	2	aaa10001	– – – – – – –	+ (SP) ← PC, PC_{0-10} ← ← addr11
LCALL addr16	2	12	– – – – – – –	+ (SP) ← PC, PC ← ← addr16
RET	2	22	– – – – – – –	PC ← (SP)–,
RETI	2	32	– – – – – – –	PC ← (SP)–, конец прерывания
AJMP addr11	2	aaa00001	– – – – – – –	PC_{0-10} ← addr11
LJMP addr16	2	02	– – – – – – –	PC ← addr16
SJMP rel	2	80	– – – – – – –	PC ← PC + rel
JMP @A + DPTR	2	73	– – – – – – –	PC ← DPTR + A
JZ rel	2	60	– – – – – – –	*Если* A = 0, то PC ← ← PC + rel
JNZ rel	2	70	– – – – – – –	Если A ≠ 0, то PC ← ← PC + rel
CJNE A, direct, rel	2	B5	+ – – – – – –	Если A ≠ (direct), то PC ← PC + rel
CJNE A, #data, rel	2	B4	+ – – – – – –	Если A ≠ data, то PC ← ← PC + rel
CJNE Rn, #data, rel	2	B8 — BF	+ – – – – – –	Если Rn ≠ data, то PC ← ← PC + rel, n = 0 – 7
CJNE @Ri, #data, rel	2	B6 – – B7	+ – – – – – –	Если (Ri) ≠ data, то PC ← ← PC + rel, i = 0 – 1
DJNZ Rn, rel	2	8 — DF	– – – – – – –	Rn ← Rn – 1. Если Rn ≠ ≠ 0, то PC ← PC + rel, n = 0 – 7
DJNZ direct, rel	2	D5	– – – – – – –	(direct) ← (direct) – 1. Если (direct) ≠ 0, то PC ← ← PC + rel
NOP	1	00	– – – – – – –	Нет операции

одному из рабочих регистров или к ячейке памяти, адресуемой как прямо, так и косвенно. Кроме этого операция увеличения на единицу может быть применена к содержимому регистра указателя DPTR.

В операциях целочисленного умножения и деления без знака участвуют аккумулятор и регистр B. При умножении 8-разрядное значение A умножается на 8-разрядное значение B, а 16-разрядный результат записывается в пару BA. При этом регистр B хранит старшую часть произведения. Флажок переполнения OV устанавливается, если произведение больше 255. При делении 8-разрядного значения A на 8-разрядное значение B частное записывается в A, а остаток в B. При попытке деления на 0 устанавливается флажок переполнения. Операция десятичной коррекции для сложения DA осуществляется стандартным способом (см. § 2.3).

В составе группы команд передачи управления (табл. 4.14) находятся команды перехода AJMP, LJMP, SJMP, JMP, условного перехода JZ, JNZ, CJNE, вызова ACALL, LCALL, возврата RET, RETI и модификации с условным переходом DJNZ. Сюда же включена пустая команда NOP.

В командах передачи управления широко применяется относительная адресация, которая поддерживает перемещаемые программные модули. В качестве относительного адреса выступает 8-разрядное смещение rel со знаком, обеспечивающее ветвление от текущего положения PC в обе стороны на ± 127 байт. Для перехода в любую другую точку 64К-байтового адресного пространства может быть использован либо прямой addr16, либо косвенный @A+DPTR адрес. В последнем случае содержимое A интерпретируется как целое без знака. Вариант короткой прямой адресации addr11 внутри 2К-байтовой текущей страницы введен для совместимости с архитектурой BE48.

Все эти типы адресации могут быть применены только к операции перехода, а для операции вызова допустимы только прямой addr16 и внутристраничный addr11 способы адресации. Во всех условных операциях может использоваться только относительная адресация.

Когда МК BE51 опознает запрос на прерывание, она генерирует одну из команд типа LCALL addr16, что автоматически обеспечивает запоминание адреса возврата в стеке. Однако в отличие от BE48 в BE51 нет автоматически сохраняемой информации о состоянии. При этом логика прерываний перестает срабатывать на запросы того уровня, который был принят к обслуживанию. Для понижения уровня прерывания служит команда возврата из прерывания RETI, которая кроме операции, эквивалентной RET, включает операцию разрешения прерывания данного уровня.

К типовым условным операциям BE51 относятся также операции JZ и JNZ, JC и JNC. Две последние включены в группу булевых. Однако появилась новая операция «Сравнить и перейти» CJNE. По данной команде операнд сначала сравнивается по правилам вычитания целых чисел с константой и в соответствии с результатом сравнения выставляется флажок CY. Затем в случае несовпадения с константой выполняется ветвление. Сравнивая аккумулятор, регистр или ячейку памяти с последовательностью констант, получаем удобный способ проверки на совпадения, например с целью выявления особых случаев. По сути дела команда CJNE является элементом оператора языков высокого уровня типа CASE.

Дальнейшее развитие получила и команда DJNZ. Теперь программист в качестве счетчика может использовать не только один из рабочих регистров Rn, $n=0-7$, но и любую ячейку памяти данных DSEG.

Таблица 4.15

Мнемоника	Число циклов	Код	Флажки							Описание
			CY	AC	F0	RS	RS	OV	P	
MOV C,bit	1	A2	+	–	–	–	–	–	–	CY ← BSEG(bit)
MOV bit,C	2	92	–	–	–	–	–	–	–	BSEG(bit) ← CY
CLR C	1	C3	0	–	–	–	–	–	–	CY ← 0
CLR bit	1	C2	–	–	–	–	–	–	–	BSEG(bit) ← 0
SETB C	1	D3	1	–	–	–	–	–	–	CY ← 1
SETB bit	1	D2	–	–	–	–	–	–	–	BSEG(bit) ← 1
CPL C	1	B3	+	–	–	–	–	–	–	CY ← NOT CY
CPL bit	1	B2	–	–	–	–	–	–	–	BSEG(bit) ← NOT BSEG(bit)
ANL C,bit	2	82	+	–	–	–	–	–	–	CY ← CY AND BSEG(bit)
ANL C,/bit	2	B0	+	–	–	–	–	–	–	CY ← CY AND NOT BSEG (bit)
ORL C,bit	2	72	+	–	–	–	–	–	–	CY ← CY OR BSEG(bit)
ORL C,/bit	2	A0	+	–	–	–	–	–	–	CY ← CY OR NOT BSEG (bit)
JC rel	2	40	–	–	–	–	–	–	–	*Если CY=1, то* SJMP *rel*
JNC rel	2	50	–	–	–	–	–	–	–	Если CY=0, то SJMP rel
JB bit,rel	2	20	–	–	–	–	–	–	–	Если BSEG(bit)=1, то SJMP rel
JNB bit,rel	2	30	–	–	–	–	–	–	–	Если BSEG(bit)=0, то SJMP rel
JBC bit,rel	2	10	–	–	–	–	–	–	–	Если BSEG(bit)=1, то SJMP rel и BSEG(bit) ← 0

Ряд команд, предназначенных для выполнения операций пересылки, проверки условий и логической обработки булевых (одноразрядных) переменных, образует отдельную группу (табл. 4.15). В качестве одного из операндов они применяют флажок переноса CY, в качестве другого служит прямо адресуемый элемент пространства BSEG. Флажок CY при выполнении операций И и ИЛИ может рассматриваться как булевый аккумулятор.

В группу входят также операции безусловного и условного переходов с относительным 8-разрядным смещением rel. Условный переход может быть осуществлен как при установленном JB, так и при сброшенном JNB бите. Наличие команд с прямым адресом проверяемого бита объясняет отсутствие специальных команд проверки тестовых входов, которые были в BE48. Особо следует отметить операцию JBC, которая реализует ветвление при установленном бите и одновременно с этим сбрасывает его в 0. Такая операция полезна в системах, решающих много задач, при организации семафоров, которые вводятся для защиты коллективно используемых ресурсов МС.

Семафор представляет собой расположенный в памяти флажок, информирующий о состоянии связанного с ним ресурса: 1 — «Свободно»; 0 — «Занято». Захват ресурса допускается только в случае,

если он свободен, затем семафор должен быть переведен в состояние «Занято». Захват ресурса с помощью команды JBC осуществляется следующим образом:

```
WAIT:   JBC     bit,OK      ;Проверка семафора bit
        SJMP    WAIT        ;Ресурс занят, ожидание
OK:     —                   ;Ресурс захвачен
```

4.13. Функциональное описание BE51

Синхронизация микроконтроллера. Приборы семейства BE51/8052 размещаются в стандартных 40-выводных корпусах (рис. 4.24), для их работы требуется единственный источник питания +5 В. Встроенный в схему генератор рассчитан на работу с кварцевым резонатором (диапазон частот 3,5—12 МГц), подключенным к выводам XTAL1 и XTAL2 (рис. 4.25). Возможно также использование внешнего ГТИ с подачей тактовых импульсов на вывод XTAL1.

При делении частоты OSC на 2 получается основная внутренняя тактовая частота CLK. Первая половина периода CLK называется фазой P1, вторая — фазой P2. Каждый машинный цикл состоит из шести периодов CLK, называемых состояниями S1, S2, ..., S6 или 12 периодов OSC, называемых фазами S1P1, S1P2, S2P1, S2P2, ..., S6P1, S6P2. Каждый машинный цикл сопровождается генерацией двух стробов ALE длительностью в один период CLK, которые размещаются в фазах S1P2 — S2P1 и S4P2 — S5P1 соответственно. Период следования ALE (машинный полуцикл) равен 6 периодам OSC.

Командный цикл МК содержит несколько машинных циклов и отсчитывается от фазы S1P1. По фазе S1P2 в IR фиксируется код операции. Второй байт двухбайтовой команды читается в S4P2 того же машинного цикла, третий — в S1P2 следующего. Таким образом, для ввода каждого байта требуется один машинный полуцикл. Во время всех оставшихся полуциклов в фазах S1P2 и S4P2 читается код операции следующей размещенной в памяти команды. Однако байт в IR не вводится и PC не инкрементируется, что приводит к его игнорированию. Ввод повторяется до окончания текущего командного цикла, который всегда завершается в фазе S6P2. Вслед за этим начинается новый командный цикл с вводом в IR кода операции следующей по исполнению команды.

Другую временну́ю последовательность исполнения имеет однобайтовая двухцикловая команда обращения к внешнему ОЗУ данных MOVX. Доступ к ОЗУ реализуется во втором машинном цикле, поэтому первый машинный цикл соответствует общей схеме, а во втором отсутствуют пустые операции FETCH. Во время второго машинного цикла первый строб ALE отсутствует, так как доступ к внешнему ОЗУ требует всего машинного цикла.

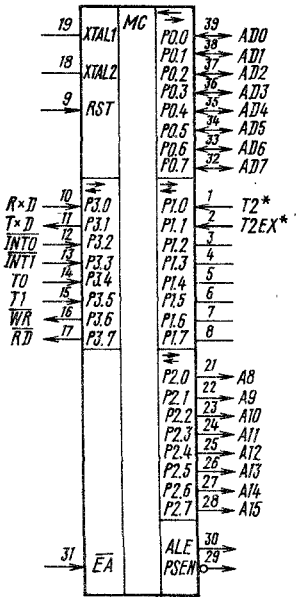


Рис. 4.24. Условное графическое обозначение BE51

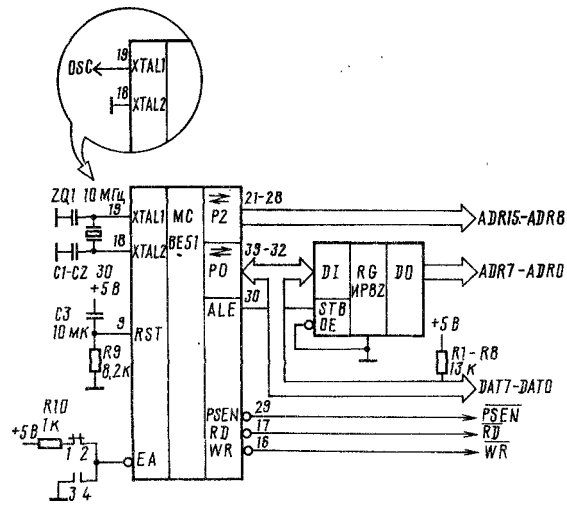


Рис. 4.25. Схема включения BE51

Запись результата операции в порты P0—P3 реализуется в фазе S6P2 последнего машинного цикла соответствующей команды, однако на выходе порта новое значение появляется только в следующей фазе S1P1. При записи в порты P1—P3 информации, требующей перехода из 0 в 1, через выходную цепь в течение S1P1 и S1P2 пропускается импульс тока, амплитуда которого в 100 раз превышает номинальное значение. Это делается для повышения скорости переключения выходных сигналов. Порт P0 такими свойствами не обладает.

Выходы портов P1—P3 обеспечивают управление четырьмя маломощными ТТЛ-входами ($I_{OL}=1{,}6$ мА). Нагрузочная способность порта P0 $I_{OL}=3{,}2$ мА. Однако для его работы требуются внешние нагрузочные резисторы, за исключением случая работы в режиме AD.

Считывание данных, присутствующих на входе портов P0, P1, выполняется по фазе S5P1 последнего машинного цикла команды, а на входе портов P2, P3—по фазе S5P2.

Вход RST служит для приведения МК в исходное состояние. Сигнал запроса воспринимается всякий раз, когда на входе RST удерживается напряжение высокого уровня более двух машинных циклов (24 периода OSC) при условии, что генератор запущен. В течение следующего машинного цикла формируется внутренний

сигнал сброса, который повторяется до тех пор, пока вход RST активен. При внутреннем сбросе установка регистров МК осуществляется следующим образом:

PC	0000H	T2CON(BE52)	00H
ACC	00H	THO	00H
B	00H	TL0	00H
PSW	00H	TH1	00H
SP	07H	TL1	00H
DPTR	0000H	TH2	00H
P0—P3	0FFH	TL2	00H
IP(BE51)	XXX00000B	PCAP2H(BE52)	00H
IP(BE52)	XX000000B	PCAP2L(BE52)	00H
IE(BE51)	0XX00000B	SCON	00H
IE(BE52)	0X000000B	SBUF	XXH
TMOD	00H	PCON(n-МОП)	0XXXXXXXB
TCON	00H	PCON(КМОП)	0XXX0000B

Схема автоматического сброса МК при включении источника питания приведена на рис. 4.25. Указанные на схеме элементы RC-цепочки обеспечивают длительность импульса RST, достаточную для стабилизации генератора и последующего за ней сброса МК.

Микроконтроллеры семейства BE51 осуществляют режим хранения записанных в их внутреннюю память данных при выключенном источнике питания +5 В. В данном режиме запасное напряжение питания поступает на вход RST, а ток потребления очень мал. При подаче напряжения +5 В запасной источник должен оставаться во включенном состоянии еще два машинных цикла, в течение которых выполняется общий сброс МК.

В КМОП-вариантах для перехода в режим с малым потреблением используются флажки PD и IDL в PCON. При установке флажка IDL работа генератора не прекращается и, следовательно, все внутренние ПУ продолжают работать. При установке флажка PD работа генератора прекращается. Выход из состояния IDL=1 может быть осуществлен по прерыванию или с помощью общего сброса. Выход из режима останова с пониженной мощностью (PD=1) может произойти только с помощью общего сброса.

В МК не предусмотрен вход пошаговой отладки, как это сделано в BE48. С этой особенностью BE51 связано усложнение отладочных средств для МС на его основе. Приходится либо разрабатывать специальные диагностические кристаллы, например 8051E, либо проектировать сложные аппаратные средства, подставляющие в «прозрачном» для пользователя режиме специальные циклические программы. Однако существует еще одна возможность организации пошагового режима работы, которая может быть использована в простейших системах отладки. Для этого резервируется один из входов $\overline{INT0}$ или $\overline{INT1}$, который

программируется для работы в режиме статического входа. Простая программа обеспечивает тестирование, например входа $\overline{INT0}$, с последующим выходом из прерывания:

```
JNB     P3.2,$      ;Ждать пока INT0=0
JB      P3.2,$      ;Ждать пока INT0=1
RETI                ;Возврат и выполне-
                    ;ние одной команды
```

После генерации на входе $\overline{ИНТ0}$ одного импульса (срез на входе $\overline{INT0}$) выполняется команда RETI. Так как следующий запрос на прерывание может быть воспринят только при выполнении одной команды после RETI, гарантируется исполнение очередной инструкции из программы пользователя.

Интерфейс расширения. Как и в BE48, внутренние ресурсы BE51/8052 могут быть расширены внешними средствами. При этом память программ CSEG достигает 64К байт, а память данных увеличивается за счет ввода в действие внешней памяти XSEG, максимальная емкость которой тоже 64К байт. Внешняя часть CSEG и область XSEG могут быть объединены в одно физическое пространство. Расширение области ВВ выполняется за счет части пространства XSEG (совмещенный ВВ).

Для связи со средствами расширения МК имеет встроенную системную магистраль, которая физически частично совмещена с портами P0, P1 и P3. В состав магистрали входят двунаправленная шина AD (порт P0), шина старшей части адреса A (порт P2), стробы фиксации адреса ALE, чтения памяти программ $\overline{PSEN}$, записи $\overline{WP}$ (линия P3.6) и чтения $\overline{RD}$ (линия P3.7) внешней памяти данных XSEG.

По срезу ALE передаваемая через порт P0 младшая часть адреса фиксируется во внешнем регистре (см. рис. 4.25). Старшая часть принимается из порта P2. Стробы $\overline{PSEN}$ ($I_{0L}=3{,}2$ мА), $\overline{RD}$ и $\overline{WR}$ определяют тип доступа. При чтении CSEG (операция $\overline{PSEN}$) данные считываются по фронту строба $\overline{PSEN}$ (фаза S1P1 и S4P1) (рис. 4.26, а), при чтении XSEG (операция $\overline{RD}$) — по фазе S3P1 (рис. 4.26, б), при записи в XSEG (операция $\overline{WR}$) данные действительны на всем стробе (рис. 4.26, в).

Цикл внешнего обращения к CSEG инициируется всякий раз при выходе адреса за пределы внутреннего ПЗУ, а также при $\overline{EA}=0$. Возможность отключения внутреннего ПЗУ используется в системах эмуляции МК и отладки МС на его основе. Цикл обращения к внешней памяти инициируется по команде MOVX.

В операциях обращения к внешней памяти участвуют порты P0 (совмещенная магистраль AD), P2 (магистраль старшего адреса), а

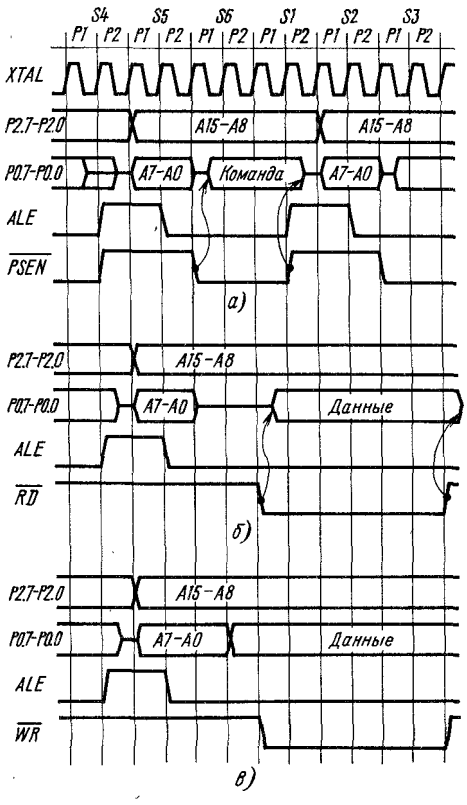


Рис. 4.26. Временные диаграммы обращения к внешней памяти:
а — считывание CSEG; б — считывание XSEG; в — запись в XSEG

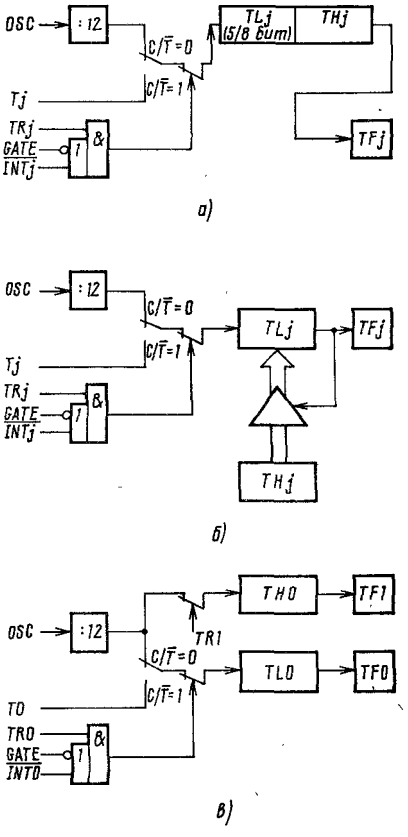


Рис. 4.27. Работа таймеров/счетчиков CT0, CT1:
а — в режиме 0 и 1; б — в режиме 2; в — в режиме 3 (только CT0)

также линии P3.6 (строб $\overline{WR}$) и P3.7 (строб $\overline{RD}$) порта P3. При работе порта P0 в качестве линии AD содержимое связанного с ним выходного регистра теряется и принимает значение 0FFH. Содержимое выходного регистра порта P2 при работе в режиме адресной шины сохраняется и поступает на выводы порта в тех машинных циклах и полуциклах, когда нет обращения к внешней памяти.

Физическое совмещение пространств CSEG и XSEG выполняется объединением стробов $\overline{PSEN}$ и $\overline{RD}$ в один строб чтения.

Работа таймеров/счетчиков. Каждый из таймеров/счетчиков CT0 — CT1 (CT2) выполняет функцию таймера ($C/\overline{T}=0$) или счетчика ($C/\overline{T}=1$). Функция таймера состоит в счете синхроим-

пульсов, следующих с частотой OSC/12, функция счетчика — в счете числа переходов из 1 в 0 на выходах T0, T1 или T2.

В этом режиме соответствующий вход тестируется в течение фазы S5P2 каждого машинного цикла. При обнаружении состояния 1 в одном цикле и состояния 0 в следующем за ним значение связанного со входом счетчика увеличивается на 1. В фазе S3P1 нового машинного цикла модифицированное значение отражается в регистровой паре. Так как процедура обнаружения перехода длится два машинных цикла, то максимальная скорость счета не должна превышать OSC/24. Существует единственное ограничение, накладываемое на временную последовательность входных импульсов, подлежащих счету: каждый уровень (1 или 0) должен удерживаться неизменным по крайней мере в течение одного машинного цикла. В рамках выбранной функции определены четыре режима работы для CT0, CT1 (M0, M1) и три для CT2 (CP/$\overline{RL2}$, RCLK, TCLK).

Режим 0 таймеров/счетчиков CT0, CT1 подобен режиму работы CT микроконтроллера BE48 с предварительным делителем на 32 (рис. 4.27,а). В данном режиме счетный регистр имеет длину 13 бит (5 младших разрядов TLj и THj, j=0−1), три старших разряда TLj игнорируются. Сигнал переполнения счетчика фиксируется флажком TFj. Счет разрешается при TRj=1 и GATE=0 либо TRj=1 и $\overline{INTj}$=1. Установка GATE=1 дает возможность таймеру измерять ширину импульсов на входе $\overline{INTj}$. Установка флажка TRj не влияет на содержимое THj, TLj.

Режим 1 подобен режиму 0, за исключением того, что в нем используется полный 16-разрядный счетчик TH, TL.

Режим 2 конфигурирует 8-разрядный счетчик TLj с автозагрузкой содержимым THj (рис. 4.27,б). Сигнал переполнения TLj не только устанавливает флажок TFj, но и перезагружает TLj содержимым THj.

Режим 3 для CT0 и CT1 осуществляется по-разному. В случае CT0 два счетных регистра TL0 и TH0 рассматриваются независимо друг от друга (рис. 4.27,в). Логика управления работой TL0 аналогична режиму 1. Второй счетчик работает в режиме таймера, управляемого флажком TR1. Сигнал переполнения фиксируется флажком TF1. В случае CT1 режим 3 останавливает счет, подобно тому, как это было при TR1=0 в других режимах. Использование режима 3 позволяет увеличить число счетчиков МК BE51 до трех, что может быть необходимо в некоторых практических случаях.

В МК 8052 имеется еще один 16-разрядный счетчик CT2, который может работать в одном из трех основных режимов: захвата (CP/$\overline{RL2}$=1), автозагрузки (CP/$\overline{RL2}$=0) и генератора скорости (RCLK=1 или TCLK=1). В зависимости от состояния флага EXEN2 определены два варианта организации режима захвата (рис. 4.28,а). При EXEN2=0 сигнал переполнения CT2

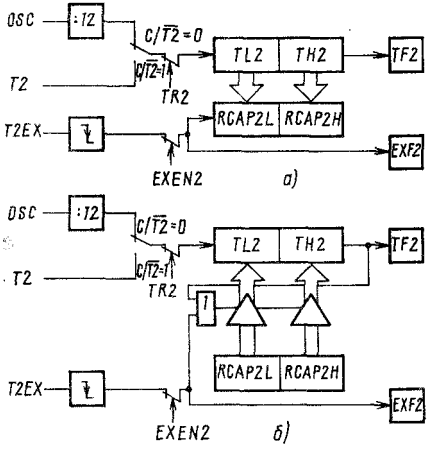


Рис. 4.28. Работа таймера/счетчика СТ2 в режимах захвата (а) и автозагрузки (б)

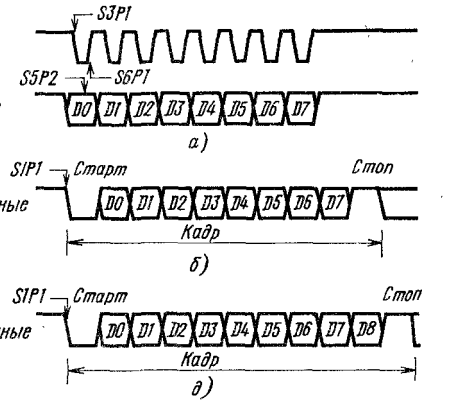


Рис. 4.29. Форматы передачи данных: а — режим 0; б — режим 1; в — режим 2 и 3

устанавливает флажок переполнения TF2. Дополнительно к этому при EXEN2 = 1 каждый переход из 1 в 0 на входе T2EX вызывает перезагрузку 16-разрядного регистра СТ2 значением, хранящимся в CAP2, и установку флажка EXF2.

В режиме автозагрузки (рис. 4.28,б) каждое переполнение СТ2 кроме установки флага TF2 вызывает перезагрузку СТ2 значением из CAP2. Одновременно с этим при EXEN2 = 1 реализуется возможность перезагрузки СТ2 переходом из 1 в 0 на входе T2EX и установку флажка EXF2.

При RCLK = 1 или TCLK = 1 устройство работает в режиме генератора скорости приемника или передатчика последовательного канала связи.

Работа последовательного канала связи. Встроенный в BE51 последовательный канал связи может быть запрограммирован для работы в одном из четырех режимов. В режиме 0 (рис. 4.29,а) данные принимаются или передаются последовательно через линию R × D со скоростью OSC/12 младшими битами вперед по 8 разрядов за операцию. Для синхронизации внешних средств используется линия T × D. Передача инициируется всякий раз, когда новые данные записываются в SBUF. Признаком окончания передачи служит установка флажка TI. Операция ввода данных активизируется при разрешенном приеме (REN = 1) по сбросу флажка RI. Установка флажка RI свидетельствует о готовности введенных данных для считывания из SBUF.

Выдаваемый на линию T × D синхросигнал переходит в состояние 0 в фазе S3P1 второго машинного цикла, следующего за циклом исполнения команды записи данных в SBUF или команды

сброса RI при REN = 1 (рис. 4.29,а). Затем он переключается через каждые шесть тактов, переходя в состояние 1 в фазе S6P1 и возвращаясь в состояние 0 в фазе S3P1 до тех пор, пока не будут переданы или приняты все 8 бит. Флажок RI или TI в зависимости от выполняемой операции устанавливается после фронта восьмого синхроимпульса. Прием выходных данных внешними средствами следует осуществлять по фронту синхросигнала. Ввод данных от внешних средств выполняется перед очередным фронтом синхросигнала в фазе S5P2.

В отличие от режима 0 в трех оставшихся режимах реализуется асинхронный обмен данными, форматы которых приведены на рис. 4.29,б,в. Для повышения помехоустойчивости приема в режимах 1—3 каждый бит данных опрашивается трижды. Для этого период передачи бита данных делится на 16 интервалов. Опрос осуществляется в интервалах 7—9. Решение о состоянии бита данных принимается голосованием «два из трех».

До прихода стартового бита приемник проверяет вход R × D 16 раз за период. При обнаружении перехода из 1 в 0 на входе R × D приемник начинает счет периодов передачи данных. В интервалах 7—9 первого периода приемник проверяет правильность генерации стартового бита. Если стартовый бит не подтверждается, то переход принимается за помеху, в противном случае реализуется операция последовательного приема 8 или 9 бит данных, которые запоминаются в регистре SBUF и во флажке RB8 (режим 2 и 3) при приеме последующего стопового бита. Одновременно с этим устанавливается флажок готовности приемника RI, свидетельствующий о приеме очередного кадра.

Данные передаются на вход T × D после их записи в SBUF независимо от состояния TI. Передача стартового бита начинается в фазе S1P1 машинного цикла, следующего за первым сигналом переполнения счетчика, который используется в качестве генератора скорости. Таким образом, начало передачи данных оказывается синхронизированным по отношению к генератору скорости. Кадр завершается выдачей стопового бита. Перед началом передачи стопового бита устанавливается флажок TI, свидетельствующий об окончании передачи данных.

В режимах 1—3 кадр данных имеет форматы, представленные на рис. 4.29,б,в. Если данные D0—D7 доступны через SBUF, то разряд D8 — через флажки TB8 и RB8 управляющего слова SCON. В режиме 2 скорость передачи в зависимости от значения флажка SMOD в регистре PCON может быть равной либо OSC/32 (SMOD = 0), либо OSC/64 (SMOD = 1). В режиме 1 и 3 скорость передачи определяется частотой переполнения CT1 или CT2 (RCLK = 1 или TCLK = 1). В случае SMOD = 1 скорость передачи совпадает с частотой переполнения, а при SMOD = 0 скорость передачи в 2 раза меньше. Таким образом, манипулируя флажком

SMOD, можно понизить скорость передачи в режимах 1—3 в 2 раза.

Флажки RI и TI способны осуществлять запрос на прерывание. В режимах 1—3 установка флажка SM2 разрешает установку флажка RI и генерацию запросов от него только при приеме кадра с D8=1. Для режима 1 это соответствует контролю кадра на стоповый бит. Для режима 2 и 3 такая организация работы обеспечивает разделение кадров на два класса в зависимости от состояния RB8. В противном случае принятые данные в SBUF не записываются и безвозвратно теряются.

Возможность реагировать только на кадры с RB8=1 учитывается при построении межмашинной магистральной сети, когда несколько подчиненных процессоров связаны через единую последовательную магистраль с главным процессором. Кадры с RB8=1 широковещательные, так как всегда воспринимаются всеми подчиненными процессорами. Они могут применяться для установления связи с одним из подчиненных узлов с последующим обменом информацией с помощью обычных кадров (RB8=0), которые игнорируются всеми процессорами, за исключением адресованного. Для вызова следующего процессора используется новый широковещательный кадр с новым адресом.

ГЛАВА 5.

ОРГАНИЗАЦИЯ ОДНОПЛАТНЫХ МИКРОКОНТРОЛЛЕРОВ НА БАЗЕ КР580ВМ80

5.1. Одноплатный микроконтроллер мМС1204

Микроконтроллеры — наиболее массовый класс микроЭВМ, встраиваемых в различные системы управления реального времени. Они отличаются достаточными для хранения рабочих программ и данных емкостями ПЗУ и ОЗУ соответственно, развитой системой ВВ, малыми габаритными размерами и мощностью потребления. Для построения МК могут быть использованы различные наборы микропроцессорных БИС. Лучше всего для этой цели подходят n-канальные БИС серии КР580, которые были положены в основу семейства одноплатных МК мМС1200.

Базовой моделью семейства является МК мМС1204 [52, 53], который представляет собой законченную одноплатную МС общего назначения с магистрально-модульной архитектурой открытого типа (рис. 5.1). Основой МС служит шина типа И41. На плате отсутствуют какие-либо средства для реализации специальных функций, ориентированных на конкретные применения.

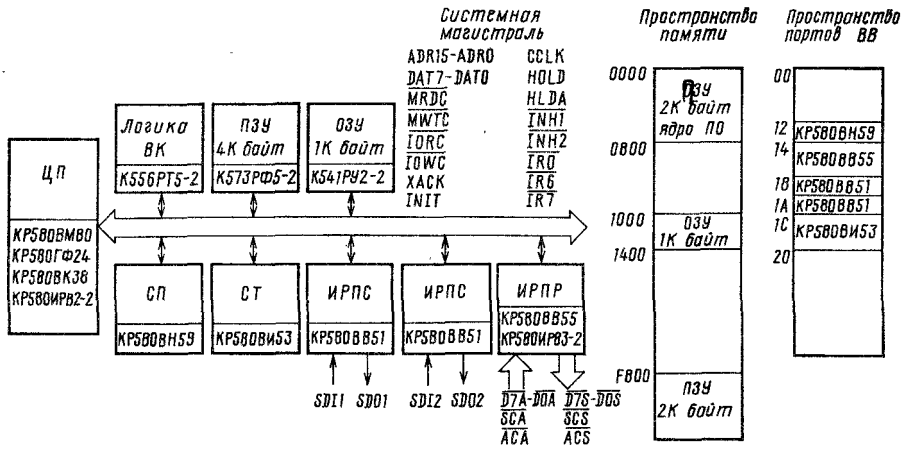


Рис. 5.1. Схема одноплатного микроконтроллера мМС1204

Специфичность и разнородность таких средств привели бы к перегрузке платы и ее неэффективному использованию. Поэтому основное внимание было уделено интегрированию действительно универсального базового набора программно-аппаратных средств, обладающего функциональной завершенностью и обеспечивающего применение МК в качестве ядра информационных систем.

В состав МК (см. рис. 5.1) входят: 8-разрядный ЦП, ПЗУ, ОЗУ, два последовательных интерфейса типа ИРПС и параллельный интерфейс типа ИРПР. Системный таймер (СТ) совместно с 8-уровневой системой прерываний (СП) обеспечивает поддержку режима реального времени [49], который характерен для многих применений МК. Внутрисистемная магистраль организует многоплатные расширения МК с помощью специальных и системных модулей, таких как модуль аналогового ВВ или системная память соответственно.

Схема ЦП и памяти МК приведена на рис. 5.2. При построении ЦП на базе МП КР580ВМ80 была использована типовая схема (см. рис. 3.7), в состав которой кроме МП входит ГТИ КР580ГФ24 и системный контроллер КР580ВК38.

В МК используется активная по умолчанию линия подтверждения обмена ХАСК, что гарантирует компактность как одноплатного ядра, так и многоплатного его расширения (см. § 2.6). Дополнительная буферизация шины адреса увеличила ее нагрузочную способность до $I_{OL}=32$ мА, $C_L=300$ пФ. Нагрузочная способность шины данных: $I_{OL}=10$ мА, $C_L=100$ пФ. Дополнительные задержки в шинах адреса и данных составляют 35 и 30 нс соответственно.

Память МК должна включать как постоянную для хранения программ и констант, так и оперативную для хранения перемен-

ных. Понятно, что емкость ПЗУ должна быть намного больше емкости ОЗУ, конкретные значения которых зависят от области применения МК и многих других факторов. Опыт показывает, что большое число современных управляющих программ занимают область около 4К байт и более. По этой причине на плате МК следует предусмотреть одно или два места для установки микросхем УСППЗУ типа К573РФ2/РФ4/РФ5/РФ6 [36]. Также могут быть использованы любые другие ПЗУ емкостью (2—32)К байт, совместимые по разводке, например, с УСППЗУ 27128 или 27256 фирмы Intel. Применение двух кристаллов К573РФ2/РФ5 данного семейства обеспечивает минимальную емкость ПЗУ 4К байт. Две 32К-байтовые микросхемы покрывают все адресное пространство МК.

Схема МК на рис. 5.2 имеет два места для установки УСППЗУ К573РФ2/РФ5 или ППЗУ КР556РТ7. При включении напряжения питания или нажатии клавиши RESET управление передается на ячейку памяти с нулевым адресом. Поэтому одна микросхема занимает первые 2К байт адресного пространства МК, другая, размещенная в старшей части пространства,— область 0F800H—0FFFFH. По усмотрению пользователя это УСППЗУ может быть перемещено в любую другую область.

Другой подход состоит в использовании универсальной панельки (рис. 5.3), допускающей применение любой микросхемы с цоколевкой семейства К573РФ2/РФ4/РФ5/РФ6. Каждый тип микросхем устанавливается в панельку одним из двух возможных способов (табл. 5.1). В зависимости от типа УСППЗУ ставятся перемычки, обеспечивающие правильное подключение микросхемы. На рис. 5.3,а приведена схема подключения БИС К573РФ2/РФ5, которая должна быть вставлена в панельку вторым способом.

Оперативная память МК может быть небольшой. В системе на рис. 5.2 это ОЗУ 1К байт, которое реализуется на двух микросхемах статического типа К541РУ2 [36] с организацией $1К \times 4$. Базовый адрес ОЗУ 1000H. Лучше всего для этой цели подходят микросхемы с байтовой организацией, например 2К-байтовая микросхема К537РУ8/РУ9 [36]. Наряду с линиями выбора кристалла $\overline{CE}$ и разрешения записи $\overline{WE}$ они имеют отдельную линию разрешения считывания $\overline{OE}$, которая подключается непосредственно к командной линии $\overline{MRDC}$.

Для подключения кристаллов ОЗУ и ПЗУ к системной магистрали требуется дополнительная логика, которую удобно формировать на биполярных ППЗУ или ПЛМ [13, 50]. В МК эта логика реализуется с помощью схемы, представленной на рис. 5.4.

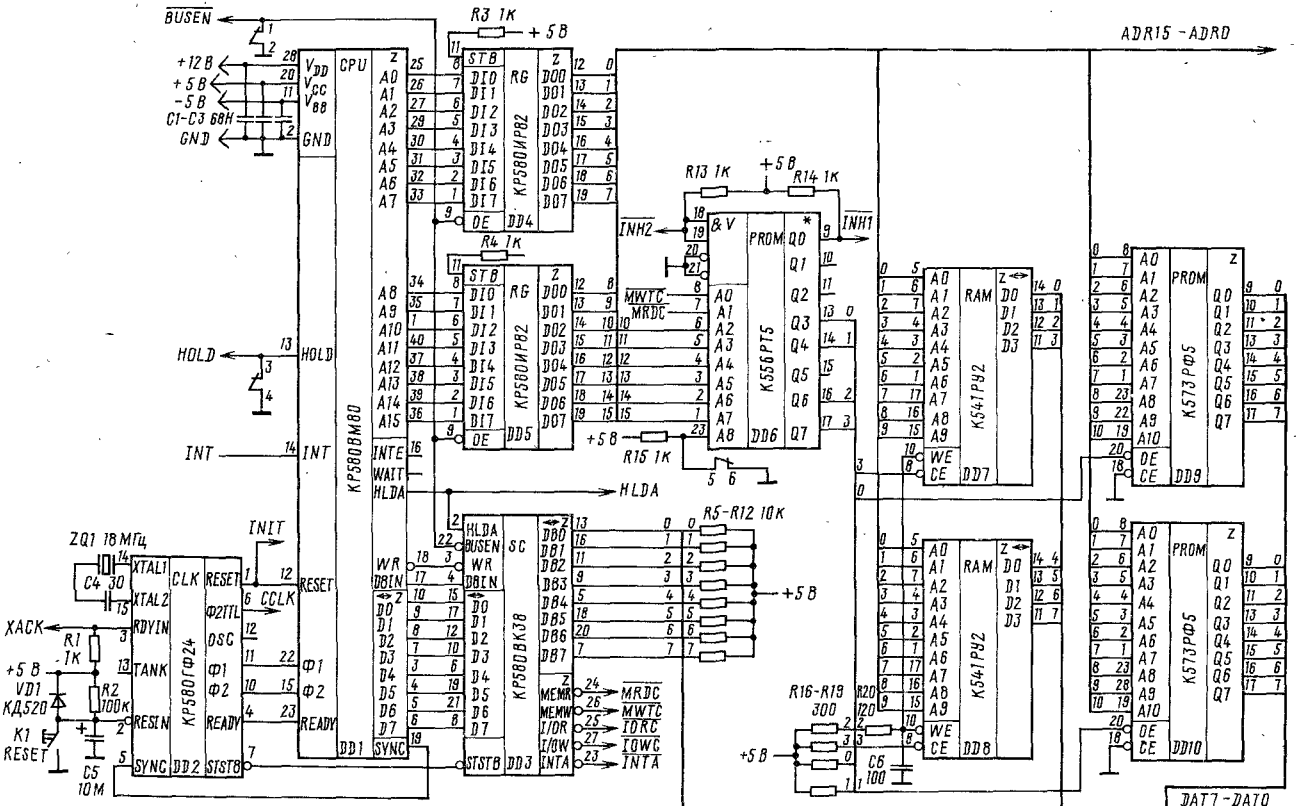


Рис. 5.2. Схема центрального процессора и памяти микроконтроллера мМС1204

Таблица 5.1

Емкость, байт,	Тип УСППЗУ	Перемычки	Способ установки
16К	К573РФ2/РФ5	3-4, 7-8	2
32К	К573РФ4	1-2, 7-8	2
64К	К573РФ6	1-2, 7-8, 11-12	1
128К	27128	1-2, 5-6, 11-12	1
256К	27256	1,2, 5-6, 9-10	1

Таблица 5.2

Адрес ПЗУ	Состояние
001	F6
005	F6
011	7E
012	3E
OF9	EE
OFD	EE
Остальные	FF

При необходимости расширение памяти МК легко выполняется с помощью дополнительной платы памяти. Одной платы вполне достаточно, чтобы покрыть все адресное пространство МС в любых сочетаниях «оперативная память — постоянная память». Системный сигнал $\overline{\text{INH1}}$ является общим сигналом выборки ОЗУ/ПЗУ, размещенных на плате МК. Он служит для запрета работы системной памяти в моменты обращения ЦП к локальным ресурсам. Сигнал $\overline{\text{INH1}}$ формируется буфером с открытым коллектором, что необходимо для его объединения по схеме «монтажное ИЛИ» с аналогичными сигналами запрета от других модулей системы.

Для построения логики выборки кристаллов ПЗУ/ОЗУ используется младшая часть адресного пространства ППЗУ К556РТ5, содержимое которого представлено в табл. 5.2. Оставшаяся часть ППЗУ может быть запрограммирована для другого распределения областей ОЗУ и ПЗУ в пространстве памяти МК. Применение программируемой пользователем логики позволило сократить число корпусов на плате и стандартизировать монтажную схему независимо от распределения физической памяти в адресном пространстве микроЭВМ.

Следует учитывать, что при использовании ППЗУ в моменты его переключения возможно появление на выходах микросхемы кратковременных ложных выбросов. Особую опасность эти выбросы представляют только для линии $\overline{\text{WE}}$ ОЗУ. Для их сглаживания рекомендуется включать дополнительную RC-цепочку.

Размещенную на плате МК память можно отключить, если активизировать линию $\overline{\text{INH2}}$. При этом становится безусловно пассивной линия $\overline{\text{INH1}}$ и открывается доступ к внешней системной памяти. Управление линией $\overline{\text{INH2}}$ осуществляется внешними средствами. Сигнал $\overline{\text{INH2}}$ может быть полезен при запрете

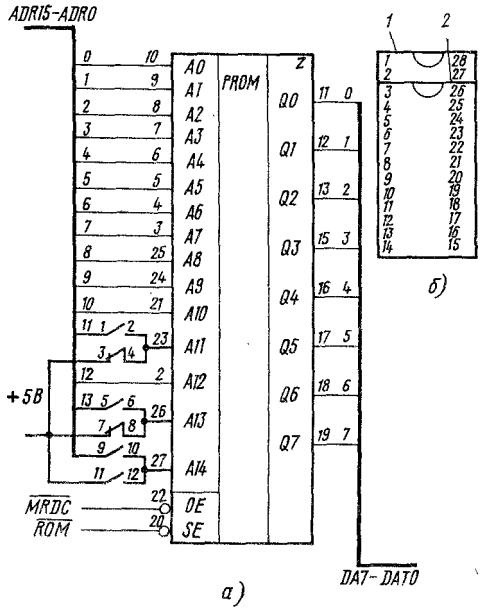


Рис. 5.3. Универсальная панелька:
а — схема подключения; б — установка микросхемы (1, 2 — способы установки)

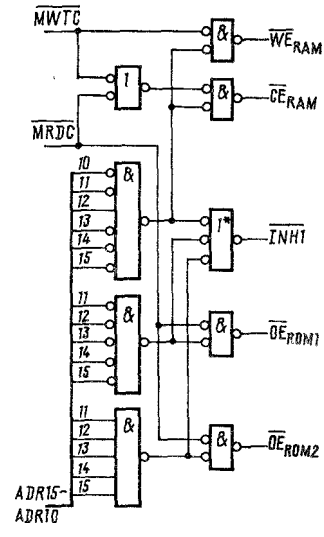


Рис. 5.4. Схема подключения приборов памяти

доступа в местную память со стороны внешних активных модулей. Для этого достаточно выполнить соединение $\overline{INH2}$ = = BUSEN.

5.2. Средства ввода-вывода и поддержки режима реального времени

Система ВВ микроконтроллера приведена на рис. 5.5. Она содержит три стандартных канала обмена данными с внешними устройствами: два последовательных типа ИРПС и один параллельный типа ИРПР. В основу последовательных каналов положены две микросхемы ПСА КР580ВВ51, которые инициализируются для работы в асинхронном старт-стоповом режиме. Один канал резервируется для местного подключения дисплея, другой может быть использован по усмотрению пользователя. Типовым вариантом его применения служит реализация радиальной межмашинной связи. Адаптер последовательного канала построен по схеме (см. рис. 3.27), описанной в § 3.4. Там же дана методика работы с ним.

Параллельный 8-разрядный интерфейс типа ИРПР реализован на базе ППА КР580ВВ55 и двух буферных регистров КР580ИР83 с инверсией. Максимальный ток нагрузки I_{0L} выходной шины

данных 32 мА. Каналы РВ и РА ППА программируются для работы в режиме соответственно ввода и вывода со стробированием. Линии РС4 и РС5 программируются на вывод и могут быть использованы по усмотрению пользователя, например для формирования выходных сигналов $\overline{INH2}$ и тестового контроля. В типовом варианте ИРПР-канал используется для подключения к МК алфавитно-цифрового печатающего устройства.

Микроконтроллер мМС1204 содержит ряд стандартных средств поддержки режима реального времени. Они включают системный таймер и контроллер обработки прерываний. Построенный на базе БИС КР580ВИ53 системный таймер содержит три 16-разрядных счетчика. Первый счетчик резервируется для формирования системных меток реального времени. Обычно они следуют с частотой 50 Гц, которая может быть изменена пользователем МК. Второй счетчик отведен для отработки пауз и интервалов в единицах системного времени. Он служит для индикации момента наступления следующего события. При частоте следования меток 50 Гц максимальная длительность интервала составляет 1310,72 с. На основе третьего счетчика построен генератор скорости передачи данных через последовательные каналы. Использование счетчиков в других целях не предусмотрено.

Восьмиуровневая система прерываний реализована на основе БИС КР580ВН59. Часть линий (IR5—IR1) используется для приема запросов от внутриплатных источников. В качестве последних выступают: счетчик меток реального времени (IR1), счетчик временных интервалов (IR2), выходной канал ИРПР (IR3), приемники ИРПС (IR4 и IR5). Для уменьшения времени отклика на запросы каждому источнику отведен индивидуальный уровень прерывания. Запрет отдельных прерываний осуществляется установкой соответствующих разрядов в регистре маски контроллера КР580ВН59.

Линии $\overline{IR0}$, $\overline{IR6}$ и $\overline{IR7}$ выведены на системную магистраль для приема запросов от внешних источников, число которых не должно превышать трех. В МК не предусмотрено расширение числа уровней методом каскадирования. Для этого в системах с бо́льшим числом источников прерываний линии на запросы объединяются по схеме «монтажное ИЛИ» в три группы. Поиск источника внутри группы осуществляется программными процедурами поллингового типа. Возможность каскадирования контроллера КР580ВН59 будет обеспечена, если на разъеме системной магистрали предусмотреть три линии каскадного расширения CAS2—CAS0.

Массивы портов периферийных БИС размещаются в пространстве ВВ микросистемы начиная с адреса 12Н. Последовательность размещения БИС (их базовые адреса) следующая: КР580ВН59 (12Н), КР580ВВ55 (14Н), первый КР580ВВ51 (18Н),

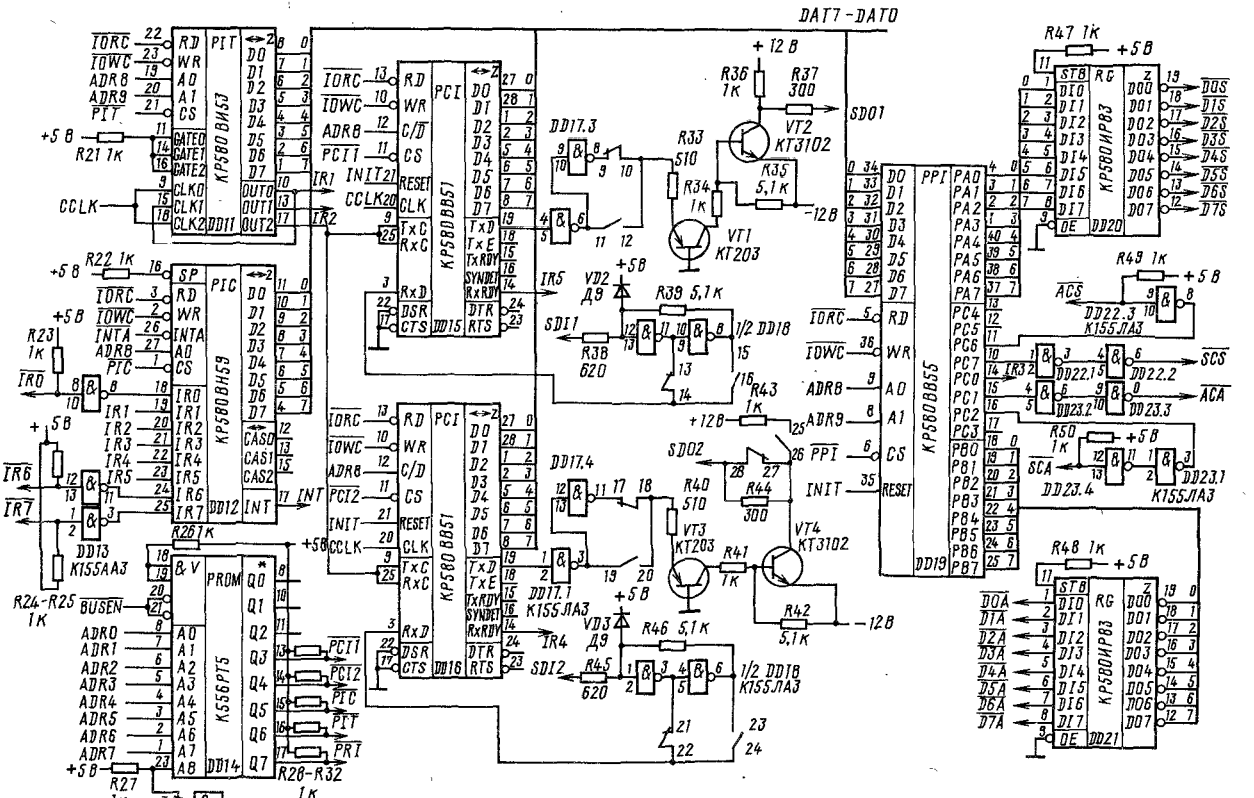


Рис. 5.5. Схема подсистемы ВВ

Таблица 5.3

Адрес ПЗУ	Состояние	Адрес ПЗУ	Состояние	Адрес ПЗУ	Состояние
012	DF	017	7F	01C	BF
013	DF	018	F7	01D	BF
014	7F	019	F7	01E	BF
015	7F	01A	EF	01F	BF
016	7F	01B	EF	Остальные	FF

второй КР580ВВ51 (1АН), КР580ВИ53 (1СН). Для более равномерного распределения нагрузки для адресации внутриплатных портов используются линии ADR8, ADR9 старшей половины адресной шины. Логика выборки кристаллов реализована на младшей половине ППЗУ К556РТ5, содержимое которой представлено в табл. 5.3. Поступающий на один из входов выборки кристалла ППЗУ К556РТ5 сигнал BUSEN обеспечивает выключение местных средств ВВ, если магистраль захвачена внешним модулем. В противном случае возможно нарушение работы системы.

Строгий отбор наиболее часто применяемых функций и их интеграция на одной плате обеспечили создание программируемого МК общего назначения, который может быть использован как в одно-, так и в многоплатных вариантах. По сути МК является модулем, служащим при построении микроЭВМ ее центральным элементом. Для этого он должен быть расширен отдельной платой основного ОЗУ [14] и средствами связи с внешней памятью. В автономном варианте МК может быть полезен при построении «разумного» сетевого контроллера, через который осуществляется подключение к локальной сети дисплеев, принтеров, накопителей и других ПУ со стандартными интерфейсами типа ИРПР или ИРПС. Другая область применения МК связана с разработкой специализированных автоматизированных систем и комплексов. Более широкому распространению МК способствует его архитектурная совместимость с серийно выпускаемым вычислительным комплексом СМ 1810. Этот комплекс может рассматриваться как инструментальная система проектирования программного обеспечения МК [15, 30].

5.3. Программирование системы ввода-вывода

В состав модуля мМС1204 входит несколько программируемых устройств: два связных адаптера КР580ВВ51, периферийный адаптер, построенный на базе КР580ВВ55, интервальный таймер КР580ВИ53 и контроллер прерываний КР580ВН59. Приведем значения и мнемонику базовых адресов этих периферийных кристаллов:

```
PSA1    EQU     18H     ;ПСА1
PSA2    EQU     1AH     ;ПСА2
PPA     EQU     14H     ;ППА
PIT     EQU     1CH     ;Системный таймер
PIC     EQU     12H     ;Контроллер прерываний
```

Аппаратное окружение устройства в значительной степени предопределяет их функциональное назначение и режим работы. Доопределение функций выполняется на программном уровне. Удобно ввести стандартный низкоуровневый слой ПО, реализующий непосредственное управление аппаратурой МК. Использование такого типового программного слоя, содержащего ряд драйверов физических устройств, освобождает программиста от учета их аппаратных особенностей. Организуется новый, не зависящий от аппаратуры программный интерфейс управления общесистемными функциями МК [52].

Для программиста ПУ представляется в виде набора основных и вспомогательных подпрограмм, поддерживающих процесс обмена с ним. Этот набор называют драйвером ПУ. Драйвер ВВ скрывает от программиста техническую организацию устройства, освобождает его от необходимости учитывать множество мелких деталей, связанных с конкретным протоколом обмена. Вместо этого он предоставляет в распоряжение программиста интерфейс программного уровня.

Основное назначение драйвера состоит в организации нового чисто программного интерфейса, благодаря которому разрывается прямая связь прикладного ПО с конкретным интерфейсом ПУ командного уровня. Удобнее всего программный интерфейс представить в виде набора подпрограмм

```
DRIVER:  JMP     INIT      ;Процедура инициализации
         JMP     INPUT     ;Ввод байта
         JMP     OUTPUT    ;Вывод байта
```

Каждая подпрограмма выполняет свою процедуру обмена или обслуживания ПУ, число и сложность которых может изменяться. В общем случае взаимодействие с каждым ПУ осуществляется с помощью программируемого адаптера или контроллера. Поэтому в состав драйвера следует ввести специальную процедуру инициализации INIT контроллера, которая настраивает его для работы в конкретном режиме. Примером простого драйвера ИРПР-адаптера на базе ВВ55 служит набор из трех рассматриваемых в § 3.3 подпрограмм

```
PA:      JMP     PAINI     ;Инициализация
         JMP     PAI       ;Условный ввод байта
         JMP     PAO       ;Условный вывод байта
```

По аналогии с драйвером РА был построен и драйвер для управления ИРПС-адаптером на базе ВВ51 (см. § 3.4):

```
SA:     JMP     SAINI       ;Инициализация
        JMP     SAI         ;Условный ввод
        JMP     SAO         ;Условный вывод
```

В составе мМС1204 два драйвера типа SA: S1 и S2, которые отличаются лишь базовыми адресами БИС связного адаптера.

Стандартизация интерфейса ВВ делает независимым ПО следующего уровня от типа используемого адаптера, в качестве которого может выступать как ППА, так и любой ПСА. Появляется возможность определить ряд дополнительных виртуальных устройств ВВ, отвечающих данному стандарту.

Во многих случаях может оказаться полезным осуществление операции ввода не с какого-либо внешнего устройства, а непосредственно из памяти МК. Для этого может использоваться драйвер буфера памяти BF:

```
BFINI:  SHLD    BFPTR       ;Инициализация указателя
        RET                 ;буфера
BFI:    PUSH    H
        LHLD    BFPTR
        MVI     A, 0FFH     ;Установка признака
        ORA     A           ;готовности ZF=1
        MOV     A, M        ;Чтение данных
        INX     H           ;Модификация указателя
        SHLD    BFPTR       ;буфера
        POP     H
        RET
```

В драйвере определены процедура инициализации, устанавливающая указатель BFPTR начала буфера, и процедура ввода. Указатель BFPTR размещается в ОЗУ и служит для указания следующего подлежащего выборке байта памяти. При выборке байта содержание указателя увеличивается на единицу. Аналогично может быть определена более редкая операция вывода в память.

Иногда удобно использовать виртуальное устройство фиктивного вывода, позволяющее осуществлять кажущийся вывод, который в действительности отсутствует. Драйвер ВВ такого устройства рассмотрен ниже. Он поддерживает единственную операцию вывода по готовности:

```
BBO:    PUSH    B
        MOV     B,A
        MVI     A,0FFH      ;Установка признака
        ORA     A           ;готовности
        MOV     A,B
        POP     B
        RET
```

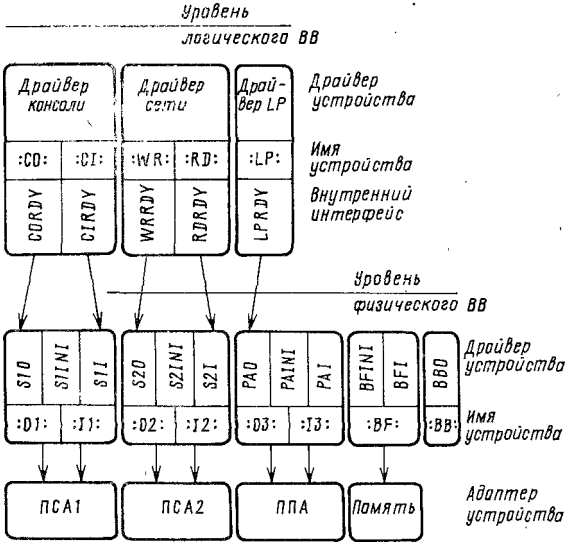


Рис. 5.6. Организация подсистемы ВВ мМС1204

Унификация программного интерфейса создает у программиста иллюзию работы с некоторым стандартным ПУ. Такое устройство будем называть логическим или виртуальным, подчеркивая факт его существования на программном уровне. Замена драйвера со стандартным программным интерфейсом никак не сказывается на ПО, что повышает его мобильность и независимость от аппаратной организации МС.

Драйверы PA, S1, S2, BF и BB совместно с адаптерами организуют физическую систему ВВ мМС1204, в составе которой могут быть выделены восемь устройств физического ввода или вывода. Среди них :O1: и :I1: (передатчик и приемник ПСА1), :O2: и :I2 (ПСА2), :O3: и :I3: (ППА), :BF: (буфер памяти) и :BB: (устройство фиктивного вывода).

Над системой физического вывода надстраивается аппаратно независимая система логического ВВ (рис. 5.6). Набор виртуальных устройств составляет основу логической системы ВВ. Типичным примером простой логической системы ВВ для бездисковых МС служит система из пяти виртуальных устройств, таких как :CI: (ввод с консоли), :CO: (вывод на консоль), :LP: (вывод на принтер), :RD: (ввод с устройства чтения), :WR: (вывод на устройство записи).

Консоль — это устройство для ведения диалога оператора с МС. В качестве консоли могут выступать различные физические устройства, например дисплей, пишущая машинка, телетайп или перфоленточный фотосчитыватель с заранее заготовленной перфолентой и т. д. Однако с точки зрения программиста обмен информацией с любым из этих устройств на логическом уровне

адекватен. Поддержка этой адекватности возлагается на драйверы физических устройств. Аналогичный подход может быть применен и к логическому принтеру, и к дополнительному логическому устройству чтения-записи, выполняющему роль внешнего ОЗУ. В наиболее простых случаях :RD: — физическое устройство чтения с перфоленты или приемник из канала связи с другой МС, а :WR: — устройство записи на перфоленту или передатчик в канал связи. Внутренний интерфейс драйверов логической системы ВВ согласуется со стандартными правилами передачи параметров при вызовах физической системы ВВ, благодаря чему логические устройства могут назначаться и переназначаться на соответствующие физические устройства для реализации той или иной конфигурации ВВ.

Пользователь может самостоятельно расширить систему физического ВВ. Находясь в рамках стандартного интерфейса связи между физической и логической системой ВВ, можно переобозначить логические устройства на соответствующие физические и воспользоваться готовыми ресурсами логической среды ВВ для организации ввода или вывода на конкретное физическое устройство.

Функции логических устройств на рис. 5.6 выполняются с помощью 15-байтовой таблицы связи, располагаемой в области ОЗУ МК:

```
CIRDY:    JMP    S11    ;:CI:=:I1:
CORDY:    JMP    S10    ;:CO:=:O1:
LPRDY:    JMP    PAO    ;:LP:=:O3:
RDRDY:    JMP    S21    ;:RD:=:I2:
WRRDY:    JMP    S20    ;:WR:=:O2:
```

По аналогии с драйверами физических устройств (физических драйверов) могут быть построены и драйверы логических устройств (логические драйверы), которые образуют второй слой логической системы ВВ. Если в функции первого слоя входит прямое управление физическим ПУ и поддержка интерфейса ПУ, то основным назначением второго слоя следует считать расширение набора типовых операций ВВ, облегчающих процесс прикладного программирования. Программист должен лишь подобрать нужную ему стандартную операцию ВВ из числа тех, которые поддерживают драйвер логического уровня.

Драйверы логических устройств могут иметь развитый набор функций ВВ. Так, входящий в состав мМС1204 драйвер консоли и принтера включает 20 процедур ВВ:

```
CO      Вывод символа из аккумулятора А на консоль
COTXT   Вывод текста на консоль. Регистр ВС — указатель текста. Код 00 —
        терминатор текста
COLF    Перевод строки
COTLE   Вывод текста в соответствии с COTXT и перевод строки
```

COH	Вывод значения младшей тетрады на консоль в шестнадцатеричном (HEX) формате
COH2	Вывод содержимого A на консоль в HEX-формате
COH4	Вывод содержимого BC на консоль в HEX-формате
MSGER	Вывод текста «ERROR» с HEX-кодом ошибки, лежащим в A
CI	Ввод символа из консоли в A. Символ вводится без дублирования. Бит четности устанавливается равным нулю
CIO	Ввод символа из консоли в A с дублированием
HEXA	Преобразование лежащего в A HEX-символа в его двоичный эквивалент. Если исходный символ — не шестнадцатеричная цифра, то он не преобразуется и флажок CY=1
CIH	Ввод в A шестнадцатеричной цифры в коде КОИ-7 и ее преобразование в двоичный эквивалент. При обнаружении ошибки флажок CY=1
CIH2	Ввод с консоли двух HEX-цифр в коде КОИ-7 и их преобразование в двоичный эквивалент, возвращаемый через A. При обнаружении ошибки флажок CY=1
CIH21	Ввод с консоли второй HEX-цифры в то время, когда в младшей тетраде A уже есть двоичный эквивалент первой. Используется после CI и HEXA при CY=0
CIH4	Ввод четырех HEX-цифр в коде КОИ-7 и их преобразование в двоичный эквивалент, возвращаемый через регистровую пару BC. При обнаружении ошибки флажок CY=1
CIH41	Ввод оставшихся трех HEX-цифр в то время, когда в младшей тетраде A уже есть двоичный код первой, и их совместное преобразование в двоичную форму, возвращаемую через BC. При обнаружении ошибки флажок CY=1. Применяется после CI и HEXA при CY=0
COLP	Вывод передаваемого через A символа на консоль с его копированием на принтер
LP	Вывод передаваемого через A символа на принтер
COPY	Включение копирования
NCOPY	Выключение копирования

Вывод на консоль может сопровождаться копированием диалога на принтер. Управление копированием (его включение и выключение) осуществляется специальными процедурами COPY и NCOPY.

5.4. Программирование средств поддержки режима реального времени

В состав аппаратуры МК мМС1204 входит ряд стандартных средств поддержки режима реального времени. Среди них два системных таймера (таймер системного времени и таймер временных интервалов), реализованные на базе ПИТ КР580ВИ53 и 8-уровневая система прерываний, в основе которой лежит ПКП КР580ВН59. Третий таймер ПИТ резервируется в качестве генератора скорости передачи ПСА1 и ПСА2.

Аппаратная логика мМС1204 предусматривает строгую специализацию всех счетчиков ПИТ и их резервирование исключительно для общесистемных целей. Благодаря этому удается определить простой драйвер системного таймера, содержащий четыре стандартные подпрограммы:

```
; Установка скорости приема-передачи ПСА1 и ПСА2
VEL:    PUSH    PSW
        MVI     A,0B6H          ;Код команды ПИТ
        OUT     PIT+3
        MOV     A,C             ;Передача параметра:
        OUT     PIT+2           ;младший байт
        MOV     A,B             ;и старший байт
        OUT     PIT+2           ;фактора скорости
        POP     PSW
        RET
; Установка периода следования системных меток
CLK:    PUSH    PSW
        MVI     A,34H           ;Код команды ПИТ
        OUT     PIT+3
        MOV     A,C             ;Передача параметра:
        OUT     PIT             ;младший байт
        MOV     A,B             ;и старший байт
        OUT     PIT             ;периода системных меток
        POP     PSW
        RET
; Запуск счетчика паузы
DELAY:  PUSH    PSW
        MOV     A,C             ;Передача параметра:
        OUT     PIT+1           ;младший байт
        MOV     A,B             ;и старший байт
        OUT     PIT+1           ;длительности паузы
        POP     PSW
        RET
; Процедура сброса счетчика паузы
DLINI:  PUSH    PSW
        MVI     A,70H           ;Код команды ПИТ
        OUT     PIT+3
        POP     PSW
        RET
```

Драйвер поддерживает следующие процедуры: VEL — установку скорости передачи по каналам связи, CLK — установку частоты следования системных меток времени, DELAY — запуск счетчика задержки и DLINI — инициализацию счетчика задержки. Процедура DLINI подготавливает счетчик CT1 ПИТ для работы в режиме 0 (прерывание по окончанию счета) и устанавливает на его выходе 0 (отсутствие запроса на прерывание IR2). Процедура DELAY загружает счетчик задержки CT1 параметром, после чего он начинает счет. По окончанию счета генерируется запрос на прерывание IR2 по

второму уровню. Более детально работа счетчика была рассмотрена в § 3.7.

Программируемый контроллер прерываний КР580ВН59 обеспечивает различные режимы работы системы прерываний. Важно определить ее стандартный способ функционирования, для поддержки которого может быть использован ряд специальных системных подпрограмм нижнего уровня. В качестве последнего удобно выбрать широко используемый режим строгого вложения приоритетов. Находясь в рамках данного режима, можно написать драйвер ПКП, включающий четыре стандартные подпрограммы:

```
; Инициализация ПКП
ICINI:  DI              ;Запрет прерываний
        PUSH    PSW
        MVI     A,0E0H
        ANA     C
        ORI     16H     ;Адресный интервал — 4,
        OUT     PIC     ;режим работы — автономный
        MOV     A,B
        OUT     PIC+1
        MVI     A, 0FFH ;Установка маски
        OUT     PIC+1
        POP     PSW
        EI              ;Разрешение прерываний
        RET
; Конец прерывания
EOI:    PUSH    PSW
        MVI     A,20H   ;EOI
        OUT     PIC
        POP     PSW
        RET
; Чтение маски прерываний
MIN:    IN      PIC+1
        RET
; Запись маски прерываний
MOUT:   OUT     PIC+1
        RET
```

Процедура ICINI инициализирует ПКП для работы в данном режиме. В качестве входного параметра через регистровую пару BC процедуре ICINI передается базовый адрес ITAB таблицы векторов, которая имеет вид

```
ITAB:                   ;База таблицы
IL0:    JMP     ADDR0   ;Пользователь IR0
        DB      0
```

```
IL1:    JMP     ADDR1   ;Системное время
        DB      0
IL2:    JMP     ADDR2   ;Система отсчета пауз
        DB      0
IL3:    JMP     ADDR3   ;Приемник ППА
        DB      0
IL4:    JMP     ADDR4   ;Приемник ПСА2
        DB      0
IL5:    JMP     ADDR5   ;Приемник ПСА1
        DB      0
IL6:    JMP     ADDR6   ;Пользователь IR6
        DB      0
IL7:    JMP     ADDR7   ;Пользователь IR7
```

Здесь ADDRN, N=0–7, служит указателем точки входа в подпрограмму обслуживания прерывания соответствующего уровня. Следует помнить, что таблица должна располагаться по 32-байтовой границе. Инициализация включает команду разрешения прерывания EI при полностью установленных масках ПКП (прерывания по всем уровням запрещены). Для управления маской служит специальная пара процедур MIN и MOUT.

При каждом включении напряжения питания или нажатии клавиши сброса ПО мМС1204 инициализирует систему прерываний с базой таблицы векторов прерываний. При этом все векторы принимают нулевое значение. Пользователь управляет средствами поддержки режима реального времени, используя вышеуказанные процедуры.

В заключение отметим, что введение низкоуровневого слоя ПО, непосредственно управляющего аппаратурой МК, позволило организовать стандартную среду программирования, свободную от аппаратных особенностей ее реализации. Система двунаправленного ВВ обладает достаточной гибкостью для создания различных физических конфигураций по вводу и выводу данных. Система поддержки режима реального времени обеспечивает простой интерфейс по организации и управлению процессами на более высоком уровне.

ГЛАВА 6.

МИКРОПРОЦЕССОР К1810ВМ86

6.1. Вводные замечания

Однокристальный высокопроизводительный 16-разрядный МП К1810ВМ86 (ВМ86) выполнен по n-МОП-технологии и имеет архитектуру, однотипную с МП 8086 фирмы Intel [27, 41, 58].

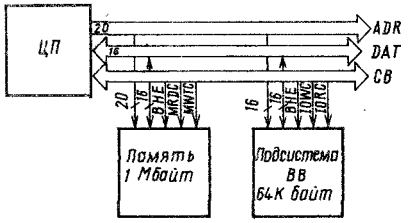


Рис. 6.1. Схема микроЭВМ на базе ВМ86

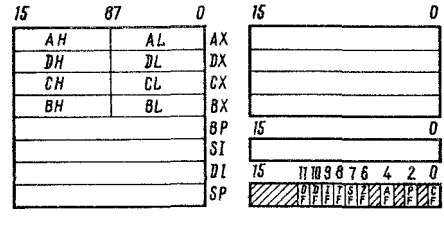


Рис. 6.2. Набор регистров ВМ86

Напряжение питания микросхемы +5 В, ток потребления не более 340 мА. Конструктивно БИС ВМ86 выполнена в корпусе типа 2123.40 с 40 двурядными выводами и шагом 2,5 мм между ними.

При выполнении операций типа регистр-регистр быстродействие МП с тактовой частотой 5 МГц составляет 2 млн. операций в секунду.

Схема микроЭВМ на основе данного МП приведена на рис. 6.1. Она содержит 20-разрядную шину ADR физического адреса, 16-разрядную шину DAT данных и шину управления. В системе используется 1 Мбайт физической памяти и изолированная от нее подсистема ВВ с 16-разрядными адресами портов ВВ. Обмен информацией осуществляется как словами, так и байтами.

Архитектура МП ВМ86 является базовой для ряда новых 16- и 32-разрядных МП типов К1810ВМ88, 80186/188 [57], 80286 [58] и 80386 [63]. Так, если архитектура К1810ВМ88 (ВМ88) тождественна ВМ86, а различия между ними наблюдаются только на уровне структурной схемы, то МП 80186/188 имеют улучшенную по отношению к базовой систему команд, включающую ряд новых операций высокого уровня. В архитектурах 80286 и 80386 системы команд получили дальнейшее развитие.

6.2. Организация регистров ВМ86

Набор программно-доступных регистров МП ВМ86 [17, 22] имеет большую функциональную неоднородность. По этой причине его нельзя назвать блоком РОН. Система команд МП ВМ86 широко использует неявную и укороченную форму представления адресной информации. В связи с этим ряд команд оперирует одним-двумя определенными регистрами из всего набора или некоторым его подмножеством. Это обстоятельство затрудняет программирование МП, так как требует от программиста учета всех функциональных особенностей, присущих его регистрам. Однако в результате широкого применения таких укороченных форм адресации объектный код МП оказывается намного короче, чем он был бы при использовании блока РОН. Время исполнения

такого кода за счет сокращения времени выборки команд будет существенно меньше.

Микропроцессор ВМ86 содержит 14 программно-доступных регистров: восемь общецелевых регистров адреса/данных AX—SP, четыре сегментных регистра CS—ES, программный счетчик IP и флажковый регистр F. Все регистры имеют по 16 разрядов. Их организация приведена на рис. 6.2.

Во время выполнения программы регистры AX—SP содержит адреса или данные. Первые четыре регистра AX—BX, как правило, используются для хранения данных, поэтому их называют регистрами данных. В командах с байтовыми операндами каждая половина регистров данных AL—BH может быть адресована отдельно. Эти подрегистры предназначены для хранения однобайтовых данных. Регистры BP—SP обычно называют адресными регистрами, так как они выполняют функцию хранения 16-разрядных адресов. Адресные регистры BP, SI и DI служат базовыми и индексными регистрами при различных способах адресации. Для этой же цели может быть использован и регистр BX. Указатель стека SP предназначен для организации стека, хранящего локальные переменные, адреса возвратов из подпрограмм и процедур обслуживания прерываний. Он всегда указывает на верхний элемент стека, который заполняется в сторону уменьшения адресов. Такая организация стека соответствует стандартной.

Существует и более узкая специализация этих регистров. Так, регистр AX (Accumulator) выполняет роль 16-разрядного, а AL—байтового аккумулятора. Для работы с ним предусмотрен полный набор команд со специальным коротким форматом. Такие операции, как ВВ данных, осуществляют обмен, только используя эти регистры. Широкое применение аккумулятора при обработке данных дает возможность программисту сократить длину программы.

Регистр данных DX (Data) служит расширением аккумулятора до 32 разрядов, выполняя при этом функции старшего слова. Такое расширение необходимо при осуществлении операций 16-разрядного умножения и деления. В ряде команд ВВ DX может использоваться в качестве регистра косвенного адреса порта ВВ.

Регистр BX (Base) кроме базового или индексного может служить регистром косвенной адресации. Такая возможность предусматривается для совместимости систем команд ВМ86 и ВМ80. Еще одно назначение регистра BX—указатель кадра FP (Frame Pointer).

В некоторых командах регистр CX (Counter) играет роль счетчика циклов, а младший байт CL применяется при реализации команд параметрического сдвига.

Адресные регистры BP—DI также имеют узкую специализацию. Так, регистры SI (Source Index) и DI (Destination Index)

могут быть использованы в качестве регистров косвенной адресации. В командах обработки строк SI и DI являются регистрами автоинкрементной или автодекрементной адресации. Тип адресации определяется флажком DF в регистре F. Если DF=1, то реализуется автодекрементный способ адресации, в противном случае — автоинкрементный. При этом SI указывает на исходную строку, а DI — на результирующую строку. Регистры SI и DI в паре с регистрами BX и BP (Base Pointer) могут быть использованы при различных способах базовой адресации с индексированием.

Сегментные регистры CS — ES служат для указания четырех сегментов, располагаемых в основной памяти.

Базовые адреса этих сегментов однозначно определяются содержимым сегментных регистров:

$$base20 = 16 \cdot sel$$

где sel — значение сегментного регистра, известное также под названием селектор. Регистр CS (Code Segment) указывает на сегмент кода, служащего для хранения исполняемых МП программ. Регистр SS (Stack Segment) указывает на стековый сегмент, резервируемый под системный стек. Два оставшихся регистра DS (Data Segment) и ES (Extra Segment) служат для указания на основной и дополнительный сегменты данных, используемые для хранения переменных.

Функцию счётчика команд в МП ВМ86 выполняет регистр IP (Instraction Pointer). Он содержит 16-разрядное значение смещения (offset) в сегменте кода, указывающее на следующий элемент программной последовательности, который подлежит выборке.

Слово состояния программы F (Flags) содержит девять флажков:

CF (Carry Flag)	Флажок переноса
PF (Parity Flag)	Флажок четности
AF (Auxiliary Carry Flag)	Флажок дополнительного переноса
ZF (Ziro Flag)	Флажок нулевого результата
SF (Sign Flag)	Флажок знака
TF (Trap Flag)	Флажок пошагового исполнения
IF (Interrupt Enable Flag)	Флажок разрешения прерывания по входу INT
DF (Direction Flag)	Флажок направления
OF (Overflow Flag)	Флажок переполнения

Флажки признаков результата CF, PF, AF, ZF, SF и OF имеют стандартное назначение (см. § 1.4). Флажки TF, IF, DF относятся к классу управляющих работой процессора. Следует отметить, что структура младшего байта F полностью соответствует структуре флажкового регистра МП ВМ80 (см. рис. 2.2).

6.3. Организация памяти ВМ86

В МП ВМ86 существует два уровня представления памяти. На нижнем физическом уровне память организована в виде монолитного 1М-байтового блока $1M \times 8$, каждый байт которого может быть адресован отдельно. Такое пространство и его адрес называют линейными. В МП ВМ86 линейный адрес эквивалентен физическому, т. е. тому, который присутствует на адресной шине.

Слова в линейной памяти занимают два любых соседних байта. В байте с младшим адресом хранится младшая часть слова. Его адрес служит адресом всего слова и может быть как четным, так и нечетным. Двойные слова размещаются как два соседних слова. Младшее слово располагается по младшему адресу, его адрес служит адресом двойного слова.

Для адресации блока в 1 Мбайт необходим 20-разрядный линейный адрес addr20. Однако программист оперирует не этими физическими адресами, а так называемыми логическими адресами. Каждый логический адрес — это пара sel: offset, состоящая из 16-разрядного селектора сегмента sel и 16-разрядного смещения внутри сегмента offset.

Множество логических адресов образует сегментированное адресное пространство логической памяти, состоящей из ряда изолированных друг от друга частей — сегментов. Селектор может рассматриваться как указатель сегмента. Всего возможно 2^{16} логических сегментов. Каждый сегмент хранит функционально законченную часть объектного кода или отдельный набор данных к нему. Отдельный сегмент должен быть отведен для стека той или иной программы. В общем случае сегментированная память отображает модульную организацию программ, что может быть использовано для поддержки модульного программирования. Размер логического сегмента не фиксирован и зависит от его конкретного функционального назначения.

Второй компонент логического адреса — смещение — определяет расстояние от начала сегмента до объекта, расположенного внутри него. Разрядность данной компоненты определяет максимально возможный размер сегмента 64К байт. В целом объем логического пространства адресов составляет 4 Гбайт ($Г = 2^{30}$).

Возникает задача отображения логических сегментов в линейную физическую память. Селектор сегмента однозначно связан с базовым линейным адресом сегмента, определяющим его расположение в линейном пространстве:

$$base20 = 16 \cdot sel$$

Линейный адрес объекта формируется из логического как сумма базы сегмента и смещения объекта в нем:

$$addr20 = base20 + offset = 16 \cdot sel + offset$$

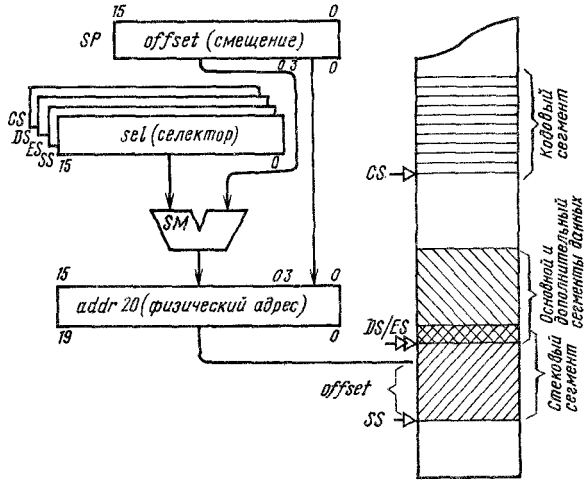


Рис. 6.3. Схема преобразования адресов

Один и тот же линейный адрес представляется множеством логических адресов.

В командах, как правило, используется неявный способ задания селектора, в качестве которого всегда выступает содержимое одного из четырех сегментных регистров. Вторая компонента логического адреса offset может быть определена различными традиционными способами.

Сегментные регистры CS—ES выделяют в памяти объемом 1 Мбайт четыре сегмента до 64К байт каждый. Сегменты выравнены по границе 16 байт (граница параграфа) и могут частично или полностью перекрываться друг с другом, как показано на рис. 6.3. Эти сегменты в отличие от остальных будем называть открытыми для доступа, вкладывая в это *понятие* тот смысл, что их селекторы хранятся в сегментных регистрах и, следовательно, доступ к ним открыт.

Неявный способ задания сегментного регистра обусловливает функциональную специализацию сегментов. В зависимости от используемого сегментного регистра различают кодовый сегмент, стековый сегмент, а также основной и дополнительный сегменты данных.

Регистр CS служит для указания кодового сегмента, который хранит объектный код. Содержимое CS используется для определения базы сегмента всякий раз, когда выполняется выборка объектного кода. При этом смещением в сегменте выступает значение программного счетчика IP.

При обращении к стеку (offset=SP) селектор извлекается из регистра SS. Поэтому указываемый им сегмент называется стековым и служит для организации системного стека.

Таблица 6.1

Вид обращения к памяти	Селектор по умолчанию	Возможные варианты	Смещение offset
Выборка команды	CS	—	—
Обращение к стеку	SS	—	—
Обращение к переменной кроме нижеуказанных	DS	CS, ES, SS	Исполнительный адрес
Обращение к переменной по базе BP	SS	CS, ES, DS	Исполнительный адрес
Обращение к исходной строке	DS	CS, ES, SS	SI
Обращение к результирующей строке	ES	—	DI

Данные обычно располагаются в основном сегменте данных, база которого хранится в DS. Смещение offset в сегменте определяется одним из традиционных способов формирования исполнительного адреса.

В правилах адресации данных существует три исключения. Так, если при обращении к данным используется один из способов базовой адресации по BP, то в качестве указателя сегмента служит SS. Этот вариант адресации эффективен при передаче параметров через текущий кадр системного стека. Указателем кадра является адресный регистр BP.

При обработке строк регистр DS служит указателем сегмента исходной строки, смещение в сегменте извлекается из индексного регистра SI. При этом результирующая строка размещается в дополнительном сегменте данных, указателем которого служит ES. Смещение строки-результата определяется по содержимому DI. Одновременное применение двух сегментов позволяет реализовать эффективную передачу данных между сегментами.

Эти используемые по умолчанию правила выбора сегментных регистров приведены в табл. 6.1. В ряде случаев существует возможность замены сегментного регистра, которая реализуется явным указанием сегментного регистра с помощью специального байта-префикса, предшествующего команде. Однако эти возможности ограничены. С помощью префикса можно изменить только сегмент данных, за исключением обращения к строке-результату. Выборка команды, обращение к стеку и размещение результирующей строки всегда выполняется с помощью CS, SS и ES соответственно.

Сегментирование памяти позволяет оформлять позиционно независимые и динамически перемещаемые программные модули, которые сами не должны изменять содержимое сегментных регистров, т. е. все смещения и переходы в программе должны выполняться относительно неизвестных, но фиксированных значений сегментных регистров.

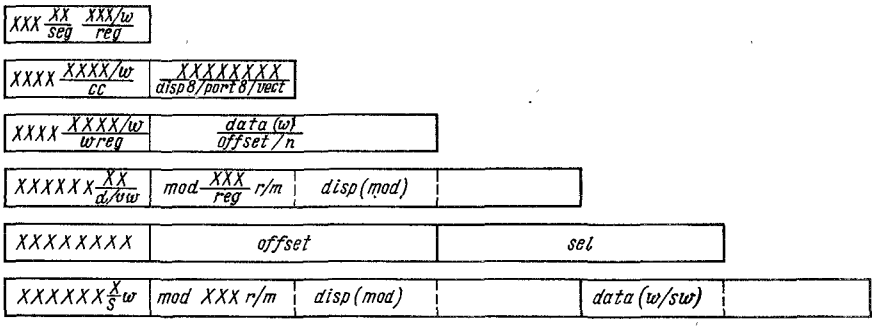


Рис. 6.4. Формат команд ВМ86

6.4. Формат команд ВМ86

Команды МП занимают в памяти от 1 до 6 байт и имеют один из представленных на рис. 6.4 форматов. Перед командой может присутствовать один или несколько байтов-префиксов, фиксирующих определенный режим исполнения команды, например префикс замены сегментного регистра. Большое число форматов объясняется стремлением разработчиков реализовать эффективное кодирование команд.

Код операции (биты X), как правило, содержится в первом байте, иногда для этой цели используется также часть или весь второй байт. В форматах команд, оперирующих как словами, так и байтами, отведен специальный бит w, служащий признаком длины операнда: при w=1 операнд имеет длину слова, при w=0 операндом является байт. Если при этом применяется непосредственный операнд, то поле data(w) будет иметь соответственно два или один байт в зависимости от значения w. В одном из двух форматов команд с непосредственным операндом предусмотрен специальный бит s — признак расширения байта в слово с учетом знака, дающий возможность кодирования 16-разрядных непосредственных данных в одном байте. Если слово упаковывается в байт, то w=1 и s=1.

Поля reg и seg используются в командах с регистровым способом адресации для указания на регистры МП. Поле reg длиной 3 бита кодирует один из восьми регистров адреса/данных. При байтовом операнде в качестве этих регистров выступают байтовые регистры AL — BH. При 16-разрядном операнде это поле кодирует регистры AX — SP. Кодирование выполняется согласно табл. 6.2. Двухразрядное поле seg применяется для ссылки на один из четырех сегментных регистров. Правила кодирования указаны в табл. 6.3.

Для команд с прямой адресацией в логическом пространстве адресов предусмотрено два формата. В одном из них зарезервировано 32-разрядное поле для указания полного логического адреса

Таблица 6.2

Код reg	Регистр		Код reg	Регистр	
	w=0	w=1		w=0	w=1
0	AL	AX	4	AH	SP
1	CL	CX	5	CH	BP
2	DL	DX	6	DH	SI
3	BL	BX	7	BH	DI

seg : offset. При этом за кодом команды следует смещение offset, а потом селектор sel, как этого требуют правила хранения двойных слов в памяти.

Таблица 6.3

Код	Регистр
0	ES
1	CS
2	SS
3	DS

В другом формате, использующем внутрисегментную прямую адресацию, предусмотрено 16-разрядное поле для кодирования только смещения offset полного логического адреса. Альтернативой внутрисегментного прямого способа адресации является относительный. Короткий относительный адрес указывается в 8-разрядном поле disp8, которое интерпретируется как представленное в дополнительном коде целое со знаком и дает возможность осуществить внутрисегментную адресацию в пределах $+127 \div -127$ от текущего положения IP.

В большинстве команд применяемый способ доступа к данным определяется байтом mod XXX r/m или mod reg r/m. Он всегда следует непосредственно за кодом команды и называется постбайтом. Первый тип постбайта используется в однооперандных командах, когда поля mod и r/m определяют одну из 32 возможных схем вычисления адреса операнда согласно табл. 6.4, а свободные биты XXX служат для расширения поля кода операции. Второй тип постбайта определяет местоположение двух операндов. В качестве одного из операндов всегда выступает содержимое регистра адреса/данных, кодируемого полем reg в соответствии с табл. 6.2, другой определяется полями mod и r/m.

Из табл. 6.4 можно получить обобщенную схему вычисления адреса

$$\text{offset} = \text{base} + \text{index} + \text{disp}$$

которая содержит три компонента. В качестве базы используется содержимое базового регистра BX или BP, в качестве индекса index — содержимое одного из индексных регистров SI или DI. Смещение кодируется непосредственно в команде и интерпретируется как целое со знаком. Сложение выполняется по модулю 2^{16}, т. е. с циклическим переходом через границу в 64К байт. В зависимости от значения смещение disp может быть как 8- так и 16-разрядным. В обоих случаях смещение следует непосредственно за постбайтом в одноименном поле disp(mod).

Таблица 6.4

Код		Схема вычисления адреса	Код		Схема вычисления адреса
mod	r/m		mod	r/m	
0	0	(BX)+(SI)	1	4	(SI)+disp8
	1	(BX)+(DI)		5	(DI)+disp8
	2	(BP)+(SI)		6	(BP)+disp8
	3	(BP)+(DI)		7	(BX)+disp8
	4	(SI)			
	5	(DI)	2	0	(BX)+(SI)+disp16
	6	offset		1	(BX)+(DI)+disp16
	7	(BX)		2	(BP)+(SI)+disp16
				3	(BP)+(DI)+disp16
1	0	(BX)+(SI)+disp8		4	(SI)+disp16
	1	(BX)+(DI)+disp8		5	(DI)+disp16
	2	(BP)+(SI)+disp8		6	(BP)+disp16
	3	(BP)+(DI)+disp8		7	(BX)+disp16
			3	x	reg (см. табл. 6.2)

Таблица 6.5

Наименование	Обозначение	Операнд
Регистровая	reg	(reg)
Прямая	addr16	M(addr16)
Косвенная регистровая	[rireg]	M((rireg))
По базе с 8-разрядным смещением	[bireg]. disp8 [bireg+disp8]	M((bireg)+disp8)
По базе с 16-разрядным смещением или индексная	[bireg]. disp16 [bireg+disp16]	M((bireg)+disp16)
	addr16 [bireg] [addr16+bireg]	M(addr16+(bireg))
По базе с индексированием без смещения	[breg] [ireg] [breg+ireg]	M((breg)+(ireg))
По базе с индексированием и 8-разрядным смещением	[breg] [ireg]. disp8 [breg+ireg]. disp8 [breg] [ireg+disp8] [breg+ireg+disp8]	M((breg)+(ireg)+disp8)
По базе с индексированием и 8-разрядным смещением или с двойным индексированием	[breg] [ireg]. disp16 [breg+ireg]. disp16 [breg] [ireg+disp16] [breg+ireg+disp16]	M((breg)+(ireg)+ +disp16)
	addr16 [breg] [ireg] addr16 [breg+ireg] [addr16+breg] [ireg]	M(addr16+(breg)+ +(ireg))
Непосредственная	data	data

Один или два компонента из трех могут отсутствовать. В зависимости от этого различают семь базовых способов адресации памяти данных, которые представлены в табл. 6.5. Эти

Таблица 6.6

Назначение	Мнемоника	Регистры						
		AX	BX	CX	DX	BP	SI	DI
Регистр косвенного адреса	rireg	—	+	—	—	—	+	+
Базовый или индексный регистр	bireg	—	+	—	—	+	+	+
Базовый регистр	breg	—	+	—	—	+	—	—
Индексный регистр	ireg	—	—	—	—	—	+	+

способы вместе с регистровым и непосредственным образуют полный набор методов адресации данных. Составы используемых в табл. 6.5 условных обозначений адресных регистров приведены в табл. 6.6.

Одноразрядный признак d/v выполняет двойную функцию. В некоторых командах он задает направление операции (функция d). При d=1 поля mod r/m постбайта определяют операнд-источник src. При d=0 ситуация меняется, поля mod r/m относятся к dst, а поле reg играет роль src. В командах сдвига и вращения этот флажок используется в качестве модификатора кода операции (функция v). Если v=0, то реализуется обычный сдвиг вправо (влево) на один разряд, в противном случае (v=1) выполняется параметрический сдвиг на число разрядов, кодируемых содержимым CL.

Поле port8 содержит короткий 8-разрядный адрес порта, расположенного в начальном диапазоне 0—255 адресов пространства ВВ. В других случаях это поле резервируется для кодирования типа вектора прерывания vect.

В командах условного перехода 4-разрядное поле cc применяется для кодирования 16 условий перехода. Младший разряд поля cc указывает на инверсию условия, а три старших определяют тип условия согласно табл. 6.7, который зависит от типа используемых данных. В соответствии с этим множество условий разделено на три группы: логических условий C—PO, анализа чисел со знаком L—NLE и сравнений чисел без знака B—NBE.

При символическом кодировании команд ВМ86 на языке ассемблера ASM86 придерживаются несколько необычных принципов. Один из них состоит в применении групповой мнемоники и отказе от различного рода модификаторов именной части команды, указывающих на типы операндов и способы доступа к ним. Это означает, что одно и то же имя, например MOV, используется как для 8-, так и 16-разрядных операндов. При этом допускаются всевозможные способы адресации, включая непосредственный. В системе команд ВМ80 для данных случаев применялись различные имена. Принцип групповой мнемоники приводит к тому, что символической команде с одним и тем же именем

Таблица 6.7

Мнемоника cc	Код cc	Условие перехода	Наименование
C	2	CF=1	Перенос
NC	3	CF=0	Нет переноса
S	8	SF=1	Отрицательный
NS	9	SF=0	Положительный
E/Z	4	ZF=1	Равно/нулевой
NE/NZ	5	ZF=0	Не равно/не нулевой
O	0	OF=1	Переполнение
NO	1	OF=0	Нет переполнения
P/PE	A	PF=1	Четный
NP/PO	B	PF=0	Нечетный
L/NGE	C	SF⊕OF=1	Меньше/не больше и не равно
GE/NL	D	SF⊕OF=0	Больше или равно/не меньше
LE/NG	E	(SF⊕OF) V ZF=1	Меньше или равно/не больше
G/NLE	F	(SF⊕OF) VZF=0	Больше/не меньше и не равно
B/NAE	2	CF=1	Меньше/не больше и не равно
AE/NB	3	CF=0	Больше или равно/не меньше
BE/NA	6	CFVZF=1	Меньше или равно/не больше
A/NBE	7	CFVZF=0	Больше/не меньше и не равно

Примечание. C — Carry, N — Not, S — Sign, E — Equal, Z — Ziro, O — Overflow, P — Parity, PE — Parity Even, PO — Parity Odd, L — Less, G — Great, B — Below, A — Above.

соответствует несколько машинных команд с разными форматами и кодами операций.

В условиях неизменной именной части вся информация о типах операндов и способах доступа к ним должна содержаться в *адресных полях команды*. Если способ доступа кодируется формой записи адресного выражения, что соответствует общепринятому, то тип используемых операндов связан с типом адресного выражения, введение которого служит еще одной особенностью языка ассемблера ASM86, сближающей его с языком высокого уровня.

Основу любых адресных выражений образуют такие простейшие элементы, как константы, переменные и метки. Обычно на уровне языка ассемблера они ничем не отличаются друг от друга. Язык ASM86 различает их, присваивая в момент определения различные атрибуты, включая тип (TYPE). Так, константам (возможно именованным), не имеющим ничего, кроме значения, присваивается тип CONSTANT (константа). Переменные имеют не только имя и значение, но и смещение (OFFSET), сегмент (SEGMENT) и тип, который может быть трех видов: BYTE (байт), WORD (слово) и DWORD (двойное слово). Меткам кроме имени тоже присваиваются такие атрибуты, как смещение, сегмент и тип.

Различают два типа меток: NEAR (внутрисегментные) и FAR (межсегментные). Если OFFSET определяет смещение переменной или метки внутри сегмента, то атрибут SEGMENT связан с именем и соответственно с селектором логического сегмента, в котором была определена данная переменная или метка. Наконец, TYPE указывает на разрядность переменной в байтах (1, 2 и 4) или число байтов, необходимых для представления метки. Внутрисегментная метка требует только 16-разрядного поля для кодирования нового значения смещения, загружаемого в IP, межсегментная метка — двойного слова для указания полного логического адреса seg:offset загружаемого в CS:IP соответственно.

Каждое адресное выражение также имеет тип, связанный с типом входящих в его состав элементов: констант, переменных, меток. Этот тип проверяется на допустимость его применения в конкретной операции и затем используется для генерации адекватной машинной команды. Так, если метки образуют основу операндов в командах передачи управления, то переменные — основу адресных выражений во всех других командах. Например, по команде MOV AX,NAME будет сформирована машинная команда с непосредственным операндом, если NAME — поименованная константа, или с прямой адресацией, если NAME — переменная типа WORD. В остальных случаях будет выдано сообщение об ошибке.

В некоторых случаях невозможно определить тип адресного выражения. Например, DEC [BX] может быть интерпретировано, не только как декрементировать байт, адресуемый BX, но и как декрементировать слово. Такие адресные выражения называются анонимными ссылками. Иногда может возникнуть необходимость в применении некоторого объекта по назначению, отличному от того, которое предписано его типом, например осуществить переход по значению переменной или использовать имя метки в качестве переменной для организации доступа к программному коду, как к обычному элементу данных. Во всех этих ситуациях появляется необходимость в явном определении типа адресного выражения. Для этого требуются специальные операторы указания типа [38]

Оператор	Тип	Оператор	Тип
BYTE PTR	BYTE	NEAR PTR	NEAR
WORD PTR	WORD	FAR PTR	FAR
DWORD PTR	DWORD		

Доступ к отдельным атрибутам с целью работы с ними обеспечивает другой набор операторов, возвращающих значение соответствующего атрибута и имеющих тип CONSTANT. Среди них:

OFFSET	Возвращает смещение объекта
SEG	Возвращает селектор объекта
TYPE	Возвращает тип объекта

Проверки на типы, свойственные языкам высокого уровня, оберегают программы от ошибок, связанных с неправильным использованием имен. Однако они вносят ограничения, не характерные для низкоуровневых языков ассемблера. Возможность применения операторов указания типа и операторов доступа к атрибутам снимают эти ограничения, позволяет программисту использовать символьное программирование.

6.5. Система команд ВМ86

Микропроцессор ВМ86 обладает развитой системой команд, включающей операции умножения и деления. Эти команды манипулируют байтами, словами, строками, 2/10-числами в упакованном и распакованном форматах [35, 38]. На уровне языка ассемблера в систему команд ВМ86 входит система команд ВМ80. Однако система команд ВМ86 имеет ряд недостатков, которые устранены в новом высокоинтегрированном однокристальном МП 80186 [54]. Системы команд МП 80186 и ВМ86 полностью совместимы на уровне объектного кода. Вместе с тем в состав команд МП 80186 много команд, которых не было в ВМ86 [46, 47].

В параграфе приводится набор команд нового МП 80186. Те команды, которых нет в составе команд ВМ86, выделены знаком «*». Таким образом, можно получить представление о дальнейшем усовершенствовании и расширении системы команд ВМ86.

Систему команд МП ВМ86 удобно разбить на шесть групп: пересылки, арифметической обработки, логической обработки, обработки строк, передачи управления, управления процессором. Рассмотрим их более подробно.

Основной формат двухадресной команды OP dst, src имеет вид

code(dw) mod red r/m disp(mod)

Один из операндов кодируется полем reg и, следовательно, должен быть регистром. Второй операнд кодируется полями mod r/m и может быть как ячейкой памяти, так и регистром. Какой из операндов dst или src является регистром, определяет признак направления d в машинной инструкции: dst = reg, при d = 1, src = reg, при d = 0. В зависимости от mod может потребоваться дополнительная 8- или 16-разрядная адресная компонента disp(mod), следующая непосредственно за постбайтом. Данный формат допускает операции как с байтами (w = 0), так и со словами (w = 1).

В формате не предусмотрена возможность кодирования непосредственной адресации. Поэтому для команд вида OP dst, data применяется специальный формат

Таблица 6.8

Мнемоника	Формат		
MOV dst, src	100010dw	mod reg r/m	
MOV dst, data	1100011w	mod 0 r/m	data (w)
MOV reg, data	1011wreg	data (w)	
MOV acc, offset	1010000w	offset	
MOV offset, acc	1010001w	offset	
MOV seg, src	8E	mod 0 seg r/m	
MOV dst, seg	8C	mod 0 seg r/m	
PUSH src	FF	mod 6 r/m	
PUSH reg	01010 reg		
PUSH seg	000seg110		
PUSH data*	011010S0	data (s)	
PUSHA*	60		
POP src	8F	mod 0 r/m	
POP reg	01011 reg		
POP seg	000seg111		
POPA*	61		
XCHG dst reg	1000011w	mod reg r/m	
XCHG AX, reg	10010 reg		
IN acc, port8	1110010w	port8	
IN acc, DX	1110110w		
OUT port8, acc	1110011w	port8	
OUT DX, acc	1110111w		
XLAT	D7		
LEA reg, src	8D	mod reg r/m	
LDS reg, src	C5	mod reg r/m	
LES reg, src	C4	mod reg r/m	
LAHF	9F		
SAHF	9E		
PUSHF	9C		
POPF	9D		

Примечание. В команде MOV seg, src seg≠1 (CS), в команде POP seg seg≠1 (CS), в команде XCHG AX, reg reg≠0 (AX), в командах LEA reg, src LDS reg, src LES reg, src mod≠11 (reg)

code (w/sw) mod code r/m disp (mod) data (w/sw)

Существуют также часто используемые короткоформатные варианты этих двухадресных команд. Операции с аккумулятором AL или AX: OP acc, data имеют формат

code (w) data (w)

Команды пересылки. Группа команд пересылки представлена в табл. 6.8. Наиболее мощная команда этой группы MOV dst, src предусматривает использование в качестве источника src или приемника dst одного из регистров адресов/данных AL—BH (w=0) или AX—SP (w=1):

```
MOV     dst, reg        ;признак d=0
MOV     reg, dst        ;признак d=1.
```

Второй операнд определяется одним из восьми способов адресации, приведенных в табл. 6.5 и кодируемых полями mod r/m постбайта.

Направление пересылки определяется признаком d в машинной команде.

Постбайт не обеспечивает поддержку непосредственного способа адресации. По этой причине введен специальный формат команды

```
MOV     dst, data
```

где data — адресное выражение типа CONSTANT. В обоих случаях пересылке подлежат либо байт (w=0), либо слово (w=1). Указателем типа данных в первом случае служит мнемоника регистра, во втором — тип dst.

Для реализации эффективной работы с наиболее часто применяемыми операциями загрузки регистров в состав машинных команд пересылки введены короткоформатные варианты

```
MOV     reg, data
MOV     acc, offset
MOV     offset, acc
```

Хотя эти команды и являются частными случаями вышеприведенных, но их формат по крайней мере на один байт короче первых, что дает возможность построить более эффективный код. Указателем типа данных служит имя регистра, использованного в командах.

Примерами команды пересылки могут служит следующие:

```
MOV     AX, WORD  PTR [BX]
MOV     BP, OFFSET  BUF
MOV     BYTE  PTR [BX],0
```

В составе набора имеются две команды загрузки сегментных регистров и считывания их содержимого. Команда загрузки CS не определена, т. е. нет возможности выполнить прямую загрузку сегментного регистра CS. Это объясняется тем, что смена селектора кодового сегмента без изменения содержимого IP — смещения внутри кодового сегмента — приведет к практически не имеющей смысла передаче управления. По этой причине загрузка CS допускается только в командах межсегментной передачи управления. Следует отметить, что загрузка сегментных регистров непосредственными данными отсутствует. Для выполнения такой операции требуется двухшаговая процедура

```
MOV     AX, SEG  VAR
MOV     DS, AX
```

обеспечивающая загрузку селектора переменной VAR в регистр DS.

Команды PUSH и POP организуют доступ к системному стеку стандартного типа. При этом пересылке подлежат только слова. Команды обеспечивают загрузку и выборку из стека операндов, хранимых как в памяти, так и в регистровой области МП. В последнем случае допускается более короткий формат кодирования. Операция загрузки CS из стека независимо от IP не определена по тем же причинам, что и ранее.

К недостаткам команд группы пересылок ВМ86 следует отнести отсутствие возможности загрузки непосредственных данных в стек. Необходимость такой загрузки возникает довольно часто при использовании стека для передачи параметров. В рамках системы команд ВМ86 для этого требуется двушаговая процедура

```
MOV    AX, data
PUSH   AX
```

Этот недостаток устранен в новом МП 80186, в систему команд которого введена команда PUSH data.

Охват командами PUSH и POP всего набора регистров, включая сегментные, обеспечивает быстрое переключение выборочного контекста МП. Однако процесс сохранения и восстановления полного контекста МП, необходимый при переключениях задач и обработки прерываний, требует довольно длительного времени. Поэтому в состав команд 80186 введены две новые команды PUSHA (PUSH All) и POPA (POP All), обеспечивающие быстрое переключение контекста. По команде PUSHA в стек последовательно загружаются содержимое всех восьми регистров адресов/данных. Загрузка выполняется в следующем порядке: AX, CX, DX, BX, начальное значение SP, BP, SI и DI. Команда POPA восстанавливает регистры в обратной последовательности. При этом выталкиваемое из стека значение SP теряется из-за ненадобности.

В составе группы команд пересылки ВМ86 находится команда обмена XCHG с форматами двух типов. Один из них обеспечивает обмен данными между регистром и памятью/регистром. Допускается обмен как словами, так и байтами. Другой более короткий формат может быть применен для кодирования операции обмена между аккумулятором AX и регистром reg (reg$\neq$AX).

В МП ВМ86 используется изолированный ВВ. Для обмена данными с пространством ВВ имеются две команды ввода IN и две команды вывода OUT. Первая пара команд IN и OUT работает с коротким 8-разрядным адресом, обеспечивающим доступ к первым 256 портам ВВ с младшими адресами. При обращении к портам со старшими адресами реализуется второй вариант команд IN и OUT с косвенной регистровой адресацией, когда регистр DX служит 16-разрядным указателем порта. Во всех случаях роль источника или приемника данных выполняет

аккумулятор, порт при этом может быть как 8-, так и 16-разрядным. Примеры команд ВВ:

```
IN    AX, 32H
OUT   DX, AL
```

Специальная команда XLAT осуществляет замену содержимого AL байтом из 256-байтовой таблицы преобразования, адресуемой регистром BX. Индекс таблицы определяется первоначальным значением AL. Класссическим примером использования этой команды служит преобразование кодов.

По команде LEA в регистр reg загружается исполнительный адрес (смещение) операнда src. Операнд источника должен быть расположен в памяти. Эта команда применяется для предварительной настройки соответствующих регистров на объекты. Например, команда XLAT требует предварительной настройки регистра BX, которая может быть выполнена следующим образом:

```
LEA    BX, TAB
```

где TAB — переменная, обозначающая массив из 256 байт.

Две другие команды обеспечивают пересылку 32-разрядных слов, обычно интерпретируемых как полные указатели seg:offset логического пространства адресов. В команде LDS селектор загружается в регистр DS, тогда как команда LES предусматривает загрузку селектора в дополнительный сегментный регистр ES. Смещение offset всегда загружается в один из регистров reg. Одновременная загрузка полного логического адреса в пару SS:SP отсутствует. Для этого используется пара команд

```
MOV    SS, WORD PTR PTRADDR+2
MOV    SP, WORD PTR PTRADDR
```

После команды MOV ss, src прерывания не проверяются, что обеспечивает согласование значений SS и SP. При другой последовательности инициализации стека необходимо предусмотреть защиту от возможного появления запросов на прерывания, переход на процедуры обслуживания которых может разрушить содержимое некоторых ячеек памяти из-за несогласованности SS и SP. Этот недостаток устранен в 32-разрядном МП 80386, в состав команд которого введена команда LSS, реализующая загрузку полного логического адреса в пару SS:reg.

Команда LAHF копирует содержимое младшего байта флажкового регистра F в регистре AH. Команда SAHF выполняет обратную пересылку. Эти команды предназначены для обеспечения совместимости ВМ86 и ВМ80 на уровне ассемблера, так как дают возможность рассматривать регистр AX как аналог регистра PSW микропроцессора ВМ80.

Таблица 6.9

Мнемоника	Формат		
ADD dst, src	000000dw	mod reg r/m	
ADD dst, data	100000sw	mod 0 r/m	data (sw)
ADD acc, data	0000010w	data (w)	
ADC dst, src	000100dw	mod reg r/m	
ADC dst, data	100000sw	mod 2 r/m	data (sw)
ADC acc, data	0001010w	data (w)	
INC dst	1111111w	mod 0 r/m	
INC reg	01000reg		
SUB dst, src	001010dw	mod reg r/m	
SUB dst, data	100000sw	mod 5 r/m	data (sw)
SUB acc, data	0010110w	data (w)	
SBB dst, src	000110dw	mod reg r/m	
SBB dst, data	100000sw	mod 3 r/m	data (sw)
SBB acc, data	0001110w	data (w)	
DEC dst	1111111w	mod 1 r/m	
DEC reg	01001reg		
CMP dst, src	001110dw	mod reg r/m	
CMP dst, data	100000sw	mod 7 r/m	data (sw)
CMP acc, data	0011110w	data (w)	
NEG dst	1111011w	mod 3 r/m	
AAA	37		
DAA	32		
AAS	3F		
DAS	2F		
MUL src	1111011w	mod 4 r/m	
IMUL src	1111011w	mod 5 r/m	
IMUL reg, src, data*	011010s1	mod reg r/m	data (s)
DIV src	1111011w	mod 6 r/m	
IDIV src	1111011w	mod 7 r/m	
AAM	D4	0A	
AAD	D5	07	
CBW	98		
CWD	99		
BOUND reg, src*	62	mod reg r/m	

Примечание. В команде BOUND reg, src mod$\neq$11

Команды PUSHF и POPF являются частным случаем операции доступа к системному стеку. Они расширяют возможности операции загрузки и считывания из стека на флажковый регистр F. В системе команд BM86 отсутствует какая-либо команда по установке/сбросу флажка трассировки TF. Эта операция может быть выполнена только с помощью команды POPF.

Все команды пересылки, за исключением SAHF и POPF, не изменяют состояния флажкового регистра.

Команды арифметической обработки. Представленная в табл. 6.9 группа содержит ряд типичных для МП команд арифметической обработки:

ADD	Сложение
ADC	Сложение с переносом
SUB	Вычитание
SBB	Вычитание с заемом
CMP	Арифметическое сравнение
INC	Увеличение на 1
DEC	Уменьшение на 1
NEG	Изменение знака

Все эти команды кодируются стандартными для ВМ86 способами. Они работают как со словами, так и с байтами. Операнды команд могут быть либо целым без знака, либо целым со знаком. В последнем случае применяется дополнительный код представления чисел. В данном случае различие между беззнаковой целочисленной арифметикой и со знаком состоит не в способе выполнения операций, а в интерпретации содержимого операндов и флажков признаков результата. Флажки устанавливаются типовым образом.

Набор команд сложения и вычитания расширен одноадресными операциями умножения и деления целых со знаком и без него:

MUL	Умножение без знака
IMUL	Умножение со знаком
DIV	Деление без знака
IDIV	Деление со знаком

Операции могут выполняться со словами и с байтами, что определяется типом src. Источником одного из операндов и приемником результата служит неявно адресуемый 8-, 16- или 32-разрядный аккумулятор, в качестве которого используется AL, AX или пара DX:AX соответственно. При этом DX представляет старшую часть аккумулятора. Операции умножения и деления выполняются по схеме

$$\text{accN} \cdot \text{src N} \rightarrow \text{acc2N}$$
$$\text{acc2N:src N} \rightarrow \text{acc2N}, \quad N = 8 \text{ или } 16$$

После выполнения операции деления в младшей части аккумулятора хранится N-разрядное частное, а в старшей — N-разрядный остаток. При делении на нуль генерируется прерывание типа 0.

При выполнении операций деления и умножения могут быть полезны две вспомогательные команды расширения чисел со знаком:

CBW	Расширение байта AL в слово AX
CWD	Расширение слова AX в двойное слово DX:AX

В состав арифметической группы введены команды для поддержки 2/10-арифметики в упакованном формате:

DAA	Десятичная коррекция AL при сложении
DAS	Десятичная коррекция AL при вычитании

а также неупакованном, например в коде КОИ-7:

AAA	Коррекция AL в коде КОИ-7 при сложении
AAS	Коррекция AL в коде КОИ-7 при вычитании
AAM	Коррекция AL в коде КОИ-7 при умножении
AAD	Коррекция AL в коде КОИ-7 при делении

При выполнении десятичной арифметики предполагается, что исходные данные уже представлены в соответствующем формате. Для представления конечного результата в коде КОИ-7 необходимо с каждой распакованной 2/10-цифрой, еще находящейся в аккумуляторе, выполнить действие

```
OR  AL, 30H
```

К недостаткам арифметической группы можно отнести отсутствие операций умножения и деления на литерал. Этот недостаток частично устранен в МП 80186.

В МП 80186 возможности команды IMUL расширены. Введено 16-разрядное умножение на литерал, имеющее два варианта записи:

```
IMUL  reg, data
IMUL  reg, src, data
```

Схема выполнения этих команд соответственно имеет вид

reg · data → reg
src · data → reg

Нетрудно заметить, что первый вариант является специальным случаем второго, когда src = reg.

В МП 80186 также предусмотрена команда проверки границ BOUND, благодаря которой упрощается работа с массивами данных. Для использования этой команды индекс массива помещается в один из регистров reg, а границы массива — в два соседних слова памяти (сначала нижняя, а затем верхняя). Команда проверяет вхождение индекса в интервал, отмеченный границами (граничные точки входят в интервал). В случае выхода за границы генерируется прерывание типа 5.

Команды логической обработки. Наиболее часто используемые операции логической обработки включены в набор команд МП ВМ86 (табл. 6.10). Среди них:

AND	Логическое И
OR	Логическое ИЛИ
XOR	Исключающее ИЛИ
TEST	Логическое сравнение
NOT	Инверсия

Таблица 6.10

Команда	Формат		
AND dst, src	001000dw	mod reg r/m	
AND dst, data	1000000w	mod 100 r/m	data (w)
AND acc, data	0010010w	data (w)	
TEST reg, src	1000010w	mod reg r/m	
TEST dst, data	1111011w	mod 000 r/m	data (w)
TEST acc, data	1010100w	data (w)	
OR dst, src	000010dw	mod reg r/m	
OR dst, data	1000000w	mod 001 r/m	data (w)
OR acc, data	0000110w	data (w)	
XOR dst, src	001100dw	mod reg r/m	
XOR dst, data	1000000w	mod 110 r/m	data (w)
XOR acc, data	0011010w	data (w)	
NOT dst	1111011w	mod 010 r/m	
ROL dst, 1/CL	110100vw	mod 000 r/m	
ROR dst, 1/CL	110100vw	mod 001 r/m	
RCL dst, 1/CL	110100vw	mod 010 r/m	
RCR dst, 1/CL	110100vw	mod 011 r/m	
SHL/SAL dst, 1/CL	110100vw	mod 100 r/m	
SHR dst, 1/CL	110100vw	mod 101 r/m	
SAR dst, 1/CL	110100vw	mod 111 r/m	
ROL dst, cnt*	1100000w	mod 000 r/m	cnt
ROR dst, cnt*	1100000w	mod 001 r/m	cnt
RCL dst, cnt*	1100000w	mod 010 r/m	cnt
RCR dst, cnt*	1100000w	mod 011 r/m	cnt
SHL/SAL dst, cnt*	1100000w	mod 100 r/m	cnt
SHR dst, cnt*	1100000w	mod 101 r/m	cnt
SAR dst, cnt*	1100000w	mod 111 r/m	cnt

Для кодирования команд применены стандартные для BM86 способы. Возможности адресации здесь те же, что и у команд арифметической обработки. Операции с аккумулятором имеют укороченный формат.

В рассматриваемую группу включен полный набор команд логического и арифметического сдвига на одну или несколько позиций вправо (влево), что определяется вторым операндом. Когда второй операнд равен единице (v=0), команды реализуют обычные сдвиги вправо (влево) на одну позицию. Когда в качестве второго операнда используется символ CL (v=1), команды образуют группу параметрических сдвигов на число позиций, определяемое содержимым CL. В составе операций сдвига следующие команды:

SAR,	SAL	Арифметический сдвиг
SHR,	SHL	Логический сдвиг
ROR,	ROL	Циклический сдвиг
RCR,	RCL	Расширенный сдвиг через CF

Среди недостатков группы команд логической обработки можно отметить отсутствие команд сдвига по литералу и команд

Таблица 6.11

Мнемоника		Формат
MOVS	type PTR [DI], seg:type PTR [SI]	1010010w
MOVSB		A4
MOVSW		A5
CMPS	type PTR [DI], seg:type PTR [SI]	1010011w
CMPSB		A6
CMPSW		A7
SCAS	type PTR [DI]	1010111w
SCASB		AE
SCASW		AF
LODS	seg:type PTR [SI]	1010110w
LODSB		AC
LODSW		AD
STOS	type PTR [DI]	1010101w
STOSB		AA
STOSW		AB
INS	type PTR [DI], DX	0110110w
INSB		6C
INSW		6D
OUTS	DX, seg:type PTR [SI]	0110111w
OUTSB		6E
OUTSW		6F
REP	minstr	F2
REPE/REPZ	cinstr	F3
REPNE/REPNZ	cinstr	F2

поразрядной обработки. Первый недостаток устранен в МП 80 186, в состав команд которого введен полный набор операций сдвига по литералу. Команды, манипулирующие битами, будут реализованы только в МП 80 386.

Команды обработки строк. В системе команд ВМ86 существует ряд средств по организации процедур обработки строк. Сюда входят базовые команды и префиксы повторения, кодировка которых приведена в табл. 6.11.

Базовые строковые операции используют регистр SI для адресации элемента исходной строки и регистр DI для указания на элемент строки-результата. Последняя всегда располагается в сегменте дополнительных данных. Элемент строки может быть байтом или словом. Во время выполнения операции указатели операндов SI и DI автоматически модифицируются на один или два в зависимости от типа элемента. Направлением модификации управляет флажок DF. Если DF=0, то реализуется увеличение указателей, если DF=1 — уменьшение.

В группе имеется пять базовых команд:

MOVC	Пересылка элемента строки
CMPS	Сравнение элементов строк методом вычитания [SI] — [DI]
SCAS	Сравнение элемента строки с содержимым аккумулятора по схеме acc — [DI]

LODS	Загрузка элемента строки в аккумулятор
STOS	Передача содержимого аккумулятора в результирующую строку

Операции CMPS и SCAS устанавливают ряд признаков результата, сам результат не возвращают. Размер операнда определяется типом адресных выражений

```
OP      WORD PTR [DI],WORD PTR [SI]
```

Существует вариант использования модификатора имени команды (суффикс B или W) для указания размера операндов, например MOVSB, LODSW. В этом случае адресные выражения не нужны и, следовательно, отсутствует возможность замены сегмента, в котором расположена строка-источник.

Любая из базовых команд может иметь префикс повторения, показывающий, что базовая операция будет повторена несколько раз. При этом регистр CX применяется в качестве счетчика числа повторений. Выход из цикла реализуется по нулевому значению CX. Проверка счетчика повторений на нуль выполняется перед каждой базовой операцией. Это означает, что нулевое начальное значение счетчика CX не вызовет никаких действий.

Существует несколько вариантов префикса повторений:

REP	Повторить
REPE/REPZ	Повторить, пока равно
REPNE/REPNZ	Повторить, пока не равно

Первый префикс используется для повторения команд пересылки minstr:MOVS, STOS известное число раз. Он реализует итерацию

```
while  CX≠0  do
       begin  CX: = CX−1;  minstr
       end
```

Четыре других варианта префикса служат для повторения команд сравнения cinstr: CMPS, SCAS. В отличие от первого эти префиксы осуществляют выход из цикла при CX=0 и ZF=1 или ZF=0 соответственно. Алгоритм их работы можно определить следующим образом:

```
while  CX≠0   do
       begin   CX:=CX−1;  cinstr
               if  ZF = 0/1  then  exit
       end
```

Такой префикс дает возможность сравнивать цепочки, находить конкретные символы в них, выполнять другие практически важные операции.

Более сложные процедуры обработки строк реализуются с помощью команд управления циклами типа LOOP, которые будут

рассмотрены ниже. Комбинируя базовые операции с префиксами повторения и командами управления итерациями, можно построить мощные и эффективные процедуры символьной обработки. При этом очень полезной может оказаться команда трансляции XLAT, преобразующая коды символов и разбивающая символы на различные классы.

В МП 80186 добавлено две новых базовых команды типа minstr, которые обеспечивают организацию блочного ВВ:

INS	Ввод элемента строки
OUTS	Вывод элемента строки

Они работают примерно так же, как и команда пересылки элемента строки MOVS, за исключением того, что одним из операндов служит порт, адресуемый DX. Здесь также существуют варианты указания типа элемента с помощью модификатора имени.

Команды передачи управления. В группе команд передачи управления (табл. 6.12) прежде всего следует выделить:

JMP	Переход
CALL	Вызов подпрограммы
RET	Возврат из подпрограммы

Эти команды обеспечивают передачу управления как внутри сегмента (типа NEAR), так и с выходом из него (типа FAR).

Таблица 6.12

Мнемоника		Формат		
JMP	disp8	E8	disp8	
JMP	disp	E9	disp	
JMP	src16	FF	mod 100 r/m	
JMP	sel:offset	EA	offset	sel
JMP	src32	FF	mod 101 r/m	
CALL	disp	E8	disp	
CALL	src16	FF	mod 010 r/m	
CALL	sel:offset	9A	offset	sel
CALL	src32	FF	mod 011 r/m	
near RET		C3		
near RET	n	C2	n	
far RET		CB		
far RET	n	CA	n	
Jcc	disp8			
LOOP	disp8			
LOOP (Z/E)	disp8			
LOOP (NZ/NE)	disp8			
JCXZ	disp8			

Примечание. В командах JMP src16, JMP src32, CALL src166, CALL src32 mod≠11

Поэтому каждая команда имеет несколько кодов и форматов, выбираемых в зависимости от типа операнда expr.

Существуют четыре разновидности каждой команды в соответствии с четырьмя допустимыми типами операнда expr: NEAR, PTR, FAR PTR, WORD PTR и DWORD PTR. Они используются для внутрисегментной и межсегментной прямой передачи управления, а также внутрисегментной и межсегментной косвенной передачи управления соответственно. В последнем случае место расположения адреса передачи управления, кодируемого словом или двойным словом, определяется одним из способов адресации с помощью постбайта.

Передача управления внутри текущего сегмента связана с изменением только содержимого IP, тогда как передача управления между сегментами требует изменения содержимого не только IP, но и селектора CS. При вызовах подпрограмм это обстоятельство учитывается также тем, что в первом случае в стеке сохраняется только IP, а во втором — пара CS:IP.

Когда объявляется вход в подпрограмму, ему присваивается тип NEAR или FAR. Явно это делается с помощью специальной директивы объявления PROC [10, 35]:

```
SUB1    PROC  NEAR
        —
        RET
SUB1    ENDP
SUB2    PROC  FAR
        —
        RET
SUB2    ENDP
```

Тогда обращение CALL SUB1 должно быть выполнено в NEAR-формате, а CALL SUB2 — в FAR-формате. В первом случае RET имеет NEAR-форму, а во втором он реализуется в FAR-форме. Считается, что подпрограммы типа NEAR необъявленные. Согласование типов подпрограмм, их вызовов и возвратов возлагается на программиста.

Для каждого типа команд возврата существует две ее разновидности:

```
RET
RET     n16
```

Здесь n16 — целое, которое используется для увеличения содержимого SP после возврата. Эта команда может быть применена для удаления параметров, записанных в стек выполнением команды вызова.

Команды условной передачи управления, имеющие вид Jcc disp8, могут осуществлять или не осуществлять переход в зависимости от состояния флажков регистра F. Каждая из этих 18

команд опрашивает соответствующую комбинацию флажков для определения условия перехода в соответствии с табл. 6.7. Если условие не выполняется, то управление передается следующей команде. При выполнении условия управление передается по адресу $ +disp8. Здесь адресное значение disp8 интерпретируется как целое со знаком, обеспечивающее относительную передачу управления в диапазоне $-128 \div +127$ байт. Такой тип передачи управления называют коротким (SHORT).

Существует также вариант безусловной команды JMP с относительным адресом. Для явного указания на использование JMP типа SHORT применяется ключевое слово SHORT:

```
JMP     SHORT  M1
```

Читателю предлагается его сравнить с обычным переходом типа NEAR

```
JMP     M1
```

Команды управления итерацией служат для организации программных циклов. Они используют регистр CX в качестве счетчика. Подобно командам условного перехода команды управления итерацией предполагают относительную адресацию в пределах $-128 \div +127$ байт и, следовательно, являются короткими (SHORT):

```
LOOP  disp8 ;CX←CX−1;
            ;if CX≠0 then IP←IP+disp8
            ;             else exit;
LOOPE/      ;CX←CX−1;
/LOOPZ      ;if CX=0 and ZF=1 then IP←IP+disp8
disp8       ;             else exit;
LOOPNE/     ;CX←CX−1;
LOOPNZ      ;if CX≠0 and ZF=0 then IP←IP+disp8
disp8       ;             else exit;
JCXZ  disp8 ;if CX=0 then IP←IP+disp8
            ;             else exit;
```

Последнюю команду целесообразно применять в начале цикла для его обхода по CX=0, т. е. для исполнения нулевых периодов цикла.

Команды управления процессором. В состав группы (табл. 6.13) входит ряд команд управления прерываниями, которые позволяют программам активизировать процедуры обслуживания прерываний:

```
INT   vect      ;−(SP)←F
                ;TF←0,  IF←0
                ;−(SP)←CS←(4·vect+2)
                ;−(SP)←IP←(4·vect)
```

Таблица 6.13

Команда		Формат		
ENTER	n,l*	C8	n16	18
LEAVE*		C9		
INT	vect	CD	vect	
INT	3	CC		
INTO		CE		
IRET		CF		
CLC		F8		
CMC		F5		
STC		F9		
CLD		FC		
STD		ED		
CLI		FA		
STI		FB		
HLT		F4		
NOP		90		
LOCK		F0		
WAIT		9B		
ESC	op, ptr	10011XXX	mod XXX r/m	
seg:		001seg110		

Если vect=3, то формируется укороченный однобайтовый вариант инструкции прерывания

```
INT     3
```

Еще одна инструкция прерывания

```
INT     0                ;if OF=1   then INT 4
                         ;          else exit;
```

используется после арифметических или логических команд для активации процедур обслуживания переполнения. По специальной команде выполняется операция выхода из процедуры обслуживания прерывания

```
IRET                     ;IP←(SP)+
                         ;CS←(SP)+
                         ;F←(SP)+
```

Более подробно информация о прерываниях изложена в § 6.6.

Группа из семи команд позволяет изменять содержимое флажков CF, DF и IF в регистре F:

```
CLC                      ;CF←0
CMC                      ;CF←NOT  CF
STC                      ;CF←1
CLD                      ;DF←0
STD                      ;DF←1
CLI                      ;IF←0
STI                      ;IF←1
```

Если первые три команды поддерживают обычное управление флажков переноса CF, то две следующие необходимы для указания направления размещения строк. Последние две команды эквивалентны командам запрета и разрешения маскируемых прерываний по входу INTR.

Команда HLT вызывает переход МП в состояние останова. Процессор может быть выведен из данного состояния либо подачей активного уровня на линию RESET, либо при получении запроса на прерывание от внешних средств. Команда HLT является альтернативой бесконечному программному циклу в ситуациях ожидания запроса на прерывания.

Команда NOP не вызывает никаких действий МП.

Префикс LOCK может быть использован в многопроцессорной системе для организации доступа к общему ресурсу, такому, как буфер, указатель, блок данных. В таких системах одновременный доступ к одному и тому же ресурсу со стороны двух и более процессоров может привести к ошибке. Для предотвращения этого в состав МС вводятся специальные системные объекты-семафоры (см. § 4.12), которые обеспечивают монопольное владение ресурсом одним процессором. Захват ресурса другими процессорами разрешается только после его освобождения.

Пусть байт SEM — семафор типа: 0 — «Свободно»; 1 — «Занято». Операция тестирования семафора и его установки в случае состояния «Свободно» может быть модифицирована в операцию считывания семафора с одновременной его установкой в состояние «Занято» и последующим анализом считанного состояния. При этом наиболее важным является этап считывания и установки. Считывание и установка должны выполняться одновременно. Этот этап можно легко осуществить с помощью команды XCHG — обмена состояния семафора с регистром, который предварительно устанавливается в необходимое состояние. Однако фаза исполнения команды XCHG состоит из нескольких циклов обращения к каналу. Для предотвращения захвата магистрали и перехвата семафора в этот период необходимо использовать префикс LOCK. Например:

```
        MOV     AL,1
WAIT:   LOCK
        XCHG    AL,SEM
        TEST    AL,AL
        JNZ     WAIT
        ; Доступ к ресурсу
        MOV     SEM,0
```

Команда WAIT переводит МП в состояние бесконечного ожидания активного уровня на входе TEST. Эта команда оказывается полезной при синхронизации программы с некоторыми внешними событиями.

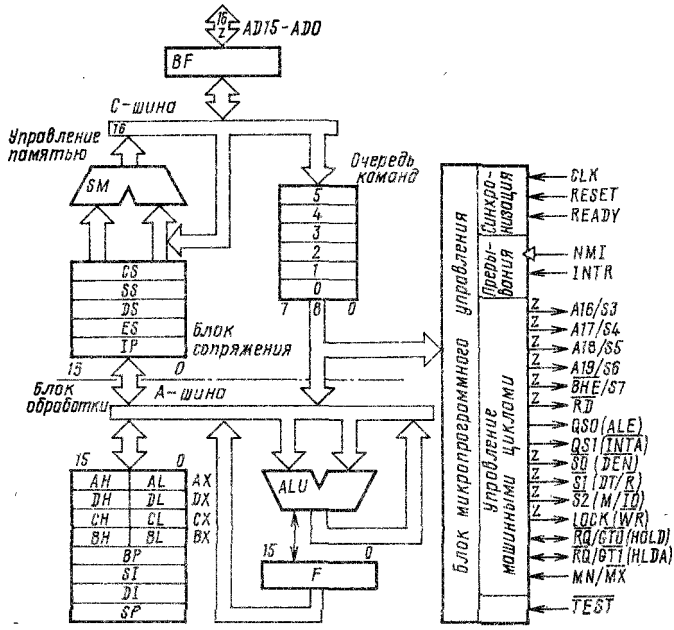


Рис. 6.5. Схема микропроцессора ВМ86

Команда ESC предоставляет возможность другому процессору (сопроцессору) получить команду, кодируемую в поле op, а также хранящийся в памяти и кодируемый полем src операнд. При этом функция генерации адреса в соответствии с src возлагается на основной процессор.

6.6. Структурная схема ВМ86

Организация циклов обращения к магистрали. Представленная на рис. 6.5 структурная схема МП ВМ86 разбита на два асинхронно работающих блока: блок сопряжения (БС) и блок обработки (БО). Блок сопряжения формирует физические 20-разрядные адреса, осуществляет опережающую выборку команд и обмен данными, управляет шиной микропроцессора. Выбранная командная последовательность хранится в буферном файле типа FIFO (очередь), глубина которого 6 байт. Всякий раз, когда в очереди освобождаются два байта и отсутствует запрос на доступ к каналу для передачи операнда, БС выбирает из памяти следующее слово командной последовательности.

Блок обработки извлекает байты команд из очереди и исполняет их. При необходимости обмена операндом с памятью БО обращается к блоку сопряжения, который и реализует этот запрос. Любое нарушение последовательного хода выполнения

программы инициирует очистку и повторное заполнение очереди команд. Совмещение операций выборки и исполнения команд привело к существенному повышению производительности МП при тех же частотных свойствах канала и быстродействии памяти МС.

Существуют два режима работы МП: минимальной и максимальной конфигурации, которые отличаются друг от друга составом входных (выходных) управляющих сигналов (рис. 6.6). Режим программируется с помощью входа MN/$\overline{MX}$.

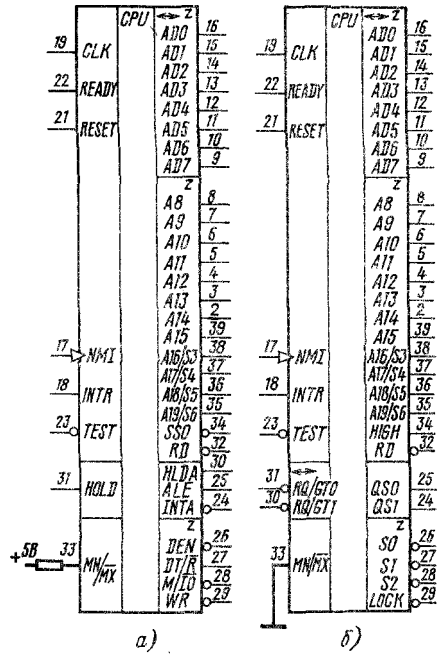


Рис. 6.6. Условное графическое обозначение микропроцессора ВМ86:
а – минимальный режим; б – максимальный режим

Работа МП осуществляется синхронно с тактовыми импульсами, поступающими на вход CLK. Стандартная частота следования тактовых импульсов 5 МГц обеспечивает микротакт МП, равный 200 нс. Границей раздела микротактов Т (тактов) считается срез синхроимпульса. Скважность следования импульсов равна 1/3. Фронт и срез CLK должны быть достаточно крутыми ($\leqslant$10 нс). В конструкцию МП включены динамические ячейки, для сохранения состояния которых необходима минимальная частота 2 МГц. Поэтому блокировка CLK с целью реализации одношагового или циклического режимов работы запрещена.

При необходимости совершить обмен данными с памятью или выбрать очередной элемент программной последовательности МП инициирует цикл обращения к магистрали. Каждый такой цикл содержит не менее четырех тактов Т1—Т4 и состоит в чтении и записи слова (байта) из МП. Временные диаграммы циклов чтения и записи показаны на рис. 6.7. В такте Т1 МП по линиям A19/S6—A16/S3, AD15—AD0 и $\overline{BHE}$/S7 передает адресную информацию A19—A0, $\overline{BHE}$. При адресации портов ВВ на линиях A19—A16 устанавливается 0. В цикле записи МП помещает данные на шину AD15—AD0 в тактах Т2—Т4. В цикле чтения такт Т2 используется для переключения шины адресов/данных на чтение. В это время шина находится в высокоомном состоянии. Прием данных осуществляется в Т3 и Т4.

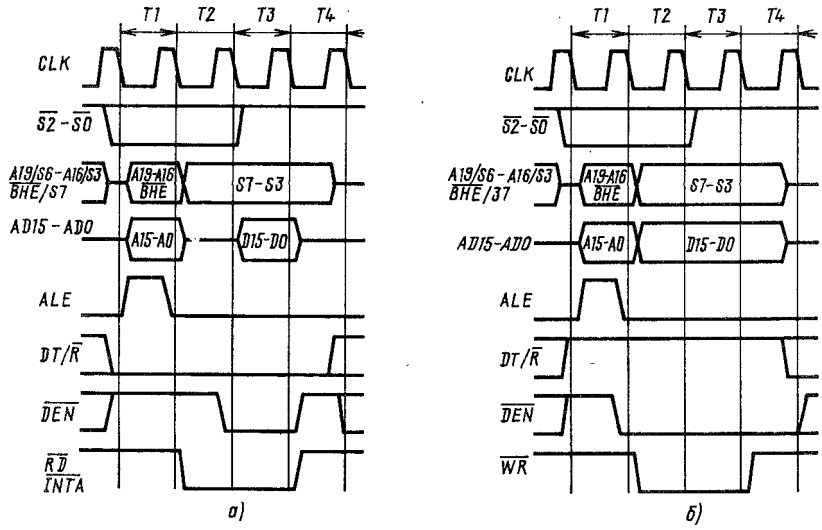


Рис. 6.7. Временные диаграммы циклов чтения (а) и записи (б) ВМ86

В такте Т3 любого машинного цикла МП опрашивает вход готовности ПУ к обмену READY. Если на входе READY сигнала готовности нет, то после такта Т3 МП входит в состояние ожидания, которое длится целое число тактов TW. После прихода сигнала READY МП завершает машинный цикл тактом Т4 и переходит к следующему циклу обращения к магистрали. Может случиться, что необходимости в таком обращении нет, тогда появляются холостые такты Т5.

Вход READY, используемый для асинхронного доступа к медленным периферийным модулям памяти и ВВ, не синхронизируется. Поэтому должны быть предусмотрены внешние средства его синхронизации импульсами CLK.

Параллельно с передачей данных в тактах Т2, Т3 (TW) и Т4 на линиях $\overline{\text{BHE}}$/S7 и A19/S6—A16/S3 присутствует информация о состоянии процессора S7—S3. Сигнал S6 всегда находится в состоянии 0, сигнал S5 индицирует состояние флажка разрешения прерывания IF, а сигналы S4, S3 кодируют сегментный регистр, применяемый для формирования физического адреса согласно табл. 6.14. Сигнал S5 обновляется в начале каждого такта CLK.

Передаваемая в первом такте Т1 информация A19—A0 и $\overline{\text{BHE}}$ используется для адресации памяти и портов ВВ. Память на физическом уровне представляет собой набор 16-разрядных ячеек, разбитых на два байтовых банка: H и L по 512К байт каждый. Если H-банк состоит из старшей половины ячеек и связан со старшим байтом 16-разрядной шины данных, то L-банк содержит младшую половину ячеек и состыкован с младшим байтом шины

Таблица 6.14

S4	S3	Сегментный регистр
0	0	ES
0	1	SS
1	0	CS, ВВ или прерывание
1	1	DS

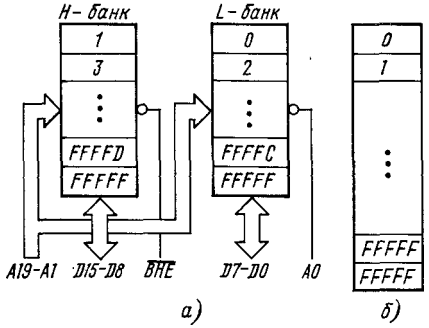


Рис. 6.8. Физическая (а) и логическая (б) организации памяти

данных. Адресные линии A19—A1 применяются для адресации ячейки. Линия A0 определяет байт в ячейке памяти (рис. 6.8).

Если A0 = 0, то адресуется либо младший байт, либо слово ячейки в зависимости от длины операнда, иначе при A0 = 1 адресуется только ее старший байт. В обоих случаях операция со старшим байтом разрешена, если $\overline{BHE} = 0$. Этот сигнал является признаком того, что старшая половина шины данных D15—D8 участвует при обмене.

Запись-считывание байтов и размещенных в одной ячейке по четному адресу слов реализуется за один машинный цикл. Обмен со словом, размещенным по нечетному адресу, выполняется за два машинных цикла. Сначала по старшей половине шины данных передается младший байт слова (A0 = 1, $\overline{BHE} = 0$), затем адрес получает приращение и по младшей половине шины данных производится обмен со старшим байтом слова. С учетом этого рекомендуется слова в памяти размещать по четным адресам. К данному замечанию особенно критично содержимое SP, так как обмен со стеком всегда выполняется по словам.

В МП используется логически изолированная от памяти структура ВВ с физически совмещенным интерфейсом. Для адресации портов ВВ существует отдельное адресное пространство в 64 К байт.

Адресное пространство портов ВВ организовано аналогично памяти в виде двух байтовых банков по 32 К байт каждый. Байтовые порты подключаются либо к старшей, либо к младшей половине шины данных. Для равномерного распределения нагрузки рекомендуется их разделить поровну между двумя банками. Шестнадцатиразрядные устройства подсоединяются ко всей шине данных. Однако при нечетном адресе обмен с такими портами будет происходить за два цикла.

Логический раздел единой магистрали между памятью и системой ВВ осуществляют управляющие сигналы, генерируемые

МП. Эти сигналы указывают, какую функцию в текущем машинном цикле выполняет магистраль: обмен с памятью или операцию ВВ. Способ формирования сигналов раздела зависит от режима работы микросхемы.

Режим минимальной конфигурации ($MN/\overline{MX}=1$) позволяет без дополнительных внешних средств создавать МС простого вида. В этом режиме все необходимые сигналы управления МП вырабатывает сам.

Рассмотрим более подробно управляющие сигналы этого режима:

ALE	Выходной сигнал, стробирующий в такте T1 адресную информацию. Используется для ее записи во внешний буферный регистр
$M/\overline{IO}$	Выходной сигнал, который отличает обмен с памятью ($M/\overline{IO}=1$) от обмена с портом ВВ ($M/\overline{IO}=0$)
$DT/\overline{R}$	Управляющий сигнал, информирующий о направлении обмена. При $DT/\overline{R}=1$ осуществляется цикл записи, в противном случае — цикл чтения. Применяется для управления направлением передачи в шинных буферах
$\overline{RD}$, $\overline{WR}$	Строб чтения и строб записи соответственно. Сигнал $\overline{RD}$ инициирует выдачу данных из памяти или порта ВВ в шину данных. Строб $\overline{WR}$ исполняет процедуру записи данных в память или порт ВВ. Считается, что данные действительны на фронте $\overline{WR}$
$\overline{DEN}$	Инверсный сигнал, разрешающий передачу данных через шину данных

В режиме максимальной конфигурации ($MN/\overline{MX}=0$) управляющая информация передается через шину состояния $\overline{S2}$—$\overline{S0}$ в закодированном виде согласно табл. 6.15. Диаграмма вывода состояния $\overline{S2}$—$\overline{S0}$ представлена на рис. 6.9. Дешифрация состояния и формирование необходимых управляющих сигналов возлагается на внешние средства — системный контроллер, который тактируется импульсами CLK.

Код состояния присутствует на этих выводах в части такта T4 предыдущего цикла, а также тактах T1 и T2 текущего цикла. В конце каждого машинного цикла шина $\overline{S2}$—$\overline{S0}$ переводится в пассивное состояние (код 111). Момент окончания пассивного состояния запускает системный контроллер на формирование управляющих сигналов нового цикла в соответствии с кодом $\overline{S2}$—$\overline{S0}$.

За счет кодирования основной управляющей информации ряд выводов МП освобождается. Эти выводы применяются для генерации дополнительной управляющей информации, упрощающей построение сложных МС. Так, выходы QS0, QS1 служат для

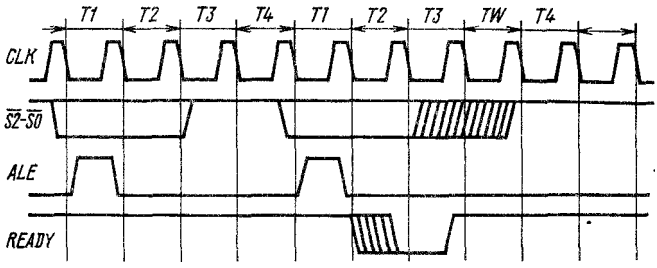


Рис. 6.9. Временные диаграммы вывода сигналов $\overline{S2}-\overline{S0}$.

вывода информации о состоянии очереди команд в соответствии с табл. 6.16 и могут быть использованы сопроцессором для отслеживания ее прохождения.

Линия $\overline{TEST}$ применяется командой WAIT для привязки программ к внешним сигналам. По команде WAIT МП опрашивает линию $\overline{TEST}$. При отсутствии сигнала на линии МП переходит в режим с периодической проверкой линии через каждые 5 тактов. Вход может быть использован для синхронизации программы с внешней аппаратурой, например с сопроцессором.

Вход RESET служит для сброса МП в начальное состояние. При запуске одновременно с включением напряжения питания на вход RESET подают сигнал, который после достижения напряжением питания номинального значения выдерживают не менее 50 мкс. При перезапуске МП на вход RESET подают сигнал длительностью не менее четырех периодов тактовой частоты (рис. 6.10).

Таблица 6.15

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Тип цикла	$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Тип цикла
0	0	0	Подтверждение прерывания	1	0	0	Выборка команды
0	0	1	Чтение порта ВВ	1	0	1	Чтение памяти
0	1	0	Запись в порт ВВ	1	1	0	Запись в память
0	1	1	Останов	1	1	1	Пассивное состояние

Таблица 6.16

QS1	QS0	Состояние очереди
0	0	Нет операции
0	1	Очистка очереди
1	0	Выборка первого байта команды
1	1	Выборка следующего байта команды

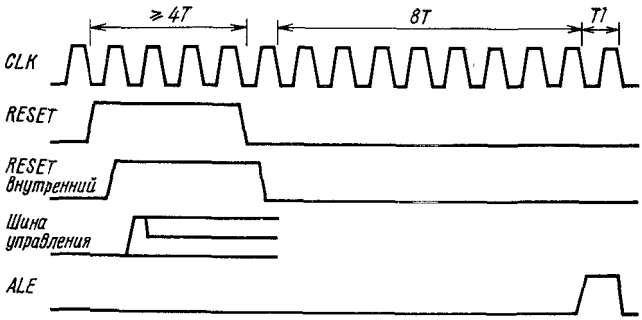


Рис. 6.10. Пуск микропроцессора ВМ86

Сигнал сброса RESET, синхронизированный тактами CLK, поступает на внутренние узлы МП. Микропроцессор устанавливает флажок IF разрешения прерывания в 0 и переводит свои выходы в состояния, указанные в табл. 6.17, в которых они остаются на все время действия RESET.

После окончания сигнала RESET МП в течении восьми первых тактов устанавливает свои регистры в исходное состояние согласно табл. 6.18 и переходит к выборке команды из ячейки с адресом 0FFFF0H. Как правило, по этому стартовому адресу находится команда межсегментного перехода FAR JMP.

Обработка прерываний. Микропроцессор ВМ86 имеет достаточно простую и эффективную векторную систему обработки прерывания. Каждому запросу поставлен во взаимно однозначное соответствие его вектор, который служит для определения адреса подпрограммы обслуживания прерывания. В МП используется 256 векторов. Способ определения вектора зависит от типа запроса. Рассмотрим более подробно запросы, поступающие от внешней аппаратуры.

Для приема этих запросов служат два входа: NMI — линия немаскируемых запросов и INTR — линия маскируемых векторных запросов. Маской служит разряд IF из флажкового регистра. Состояние флажка IF доступно внешней аппаратуре в тактах T2 — T4 через шину A18/S5. Прерывание по входу INTR воспринимается всякий раз, когда INTR · IF = 1.

В ответ на запрос по входу INTR МП инициирует процедуру ввода вектора прерывания, состоящую из двух последовательных машинных циклов INTA, разделенных двумя холостыми тактами T5. В минимальном режиме эти циклы аналогичны циклам ввода (RD-циклы), но вместо строба RD генерируется строб INTA и состояние адресной шины не определено. Для максимального режима цикл INTA идентифицируется кодом на шине S2 — S0, а генерация INTA-строба возлагается на системный контроллер.

Таблица 6.17

Вывод	Состояние
AD15—AD0	z
A19/S6—A16/S3	z
$\overline{\text{BHE}}$/S7	Через 1 в z
M/$\overline{\text{IO}}$ ($\overline{\text{S2}}$)	Через 1 в z
DT/$\overline{\text{R}}$ ($\overline{\text{S1}}$)	Через 1 в z
$\overline{\text{DEN}}$ ($\overline{\text{S0}}$)	Через 1 в z
$\overline{\text{WR}}$ ($\overline{\text{LOCK}}$)	Через 1 в z
$\overline{\text{RD}}$	Через 1 в z
$\overline{\text{INTA}}$	1
ALE	0
HLDA	0
$\overline{\text{RQ}}$/GT0—$\overline{\text{RQ}}$/GT1	1
QS1—QS0	0

Таблица 6.18

Регистр	Состояние
F	Сброшен
IP	0000H
CS	FFFFH
DS	0000H
SS	0000H
ES	0000H
Очередь команд	Пусто

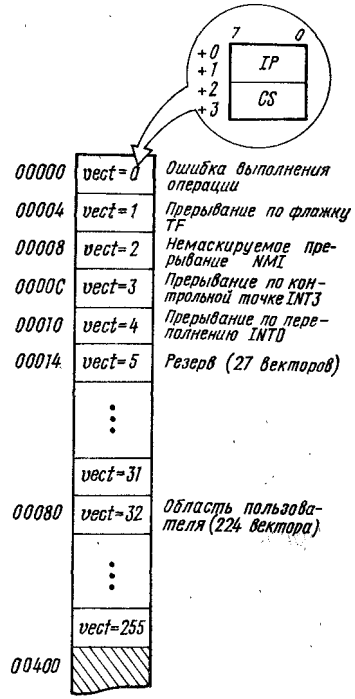


Рис. 6.11. Таблица прерываний IDT

Первый INTA-цикл информирует интерфейс МС о начале процедуры. В течение этого цикла никаких данных МП не принимает. Во втором цикле по младшей половине 16-разрядной шины принимается вектор прерывания, который лежит в диапазоне 0—255. В максимальном режиме с такта Т2 первого цикла до такта Т2 второго генерируется сигнал $\overline{\text{LOCK}}$, препятствующий захвату магистрали между INTA-циклами. В минимальном режиме сигнал M/$\overline{\text{IO}}$=0, что свидетельствует об обмене с портом ВВ. Состояние входа HOLD до конца процедуры игнорируется.

Вход немаскируемого прерывания NMI является динамическим. Запрос по нему воспринимается только при поступлении положительного перепада сигнала, который затем должен быть удержан в активном состоянии более двух тактов. Запросам по

входу NMI присвоен фиксированный вектор 2, поэтому процедура ввода номера для них не выполняется.

Кроме двух источников внешних прерываний МП имеет три источника специальных внутренних прерываний: по переполнению, по ошибке деления и при пошаговой трассировке. Прерывание по переполнению возникает в результате исполнения команды INTO, если флажок переполнения OF устанавливается в 1. Этот тип прерывания имеет вектор, равный 4.

Прерывание по ошибке деления происходит, если частное операции деления DIV или IDIV превышает максимально допустимое значение, например при делении на нуль. Прерыванию по ошибке деления присвоен вектор 0. Эти два вида прерывания (по переполнению и по ошибке деления) полезны для обнаружения арифметических ошибок.

Прерывание пошаговой трассировки вырабатывается после исполнения любой команды, перед началом выполнения которой флажок трассировки TF был установлен в 1. Прерывание реализует оперативную отладку программ в режиме пошаговой работы. Вектор данного типа прерывания равен 1.

Существует еще один источник внутреннего векторного прерывания — программный. Этот тип прерываний возникает при исполнении двухбайтовой команды INT vect. В этой команде vect — число из диапазона 0—255, которое является вектором прерывания. Определен также однобайтовый вариант команды INT 3, используемой для организации контрольных точек останова в отлаживаемой программе. Программное прерывание дублирует любой тип аппаратных прерываний, что может быть полезным для проверки подпрограмм их обслуживания.

Вектор прерывания, считанный с шины из кода команды или определенный по типу радиального запроса, служит индексом обращения к таблице прерываний IDT (рис. 6.11). Таблица размещается в первом 1К байте памяти. Под каждый вектор отводится по 4 байта, в которых хранятся начальные состояния IP и CS подпрограммы обслуживания в указанной последовательности. После определения вектора прерывания МП считывает значение IP и CS из таблицы IDT. Адресом новых значений IP и CS служит учетверенное значение вектора vect. При формировании физического адреса обращения к таблице никакие сегментные регистры не используются, о чем свидетельствуют сигналы состояния $S4=1$, $S3=0$.

Затем МП сбрасывает флажки IF и TF, выполняет загрузку в стек старого содержимого регистров F и CS, выбирает код операции первой команды подпрограммы обслуживания и «вталкивает» в стек прежнее значение IP. Эти действия он выполняет в указанной последовательности. После записи в стек старого содержимого IP МП возобновляет свою обычную работу. Сигнал

C5 (состояние флажка IF) устанавливается в 0 в такте T2 после считывания нового значения CS.

Количество тактов от конца команды и до начала выполнения подпрограммы обслуживания зависит от типа прерывания. Для запросов по входу INTR оно равно 61, для программных прерываний и прерываний при пошаговой трассировке — 51, для команды INT 3 — 52, для команды INT 0 — 53 такта. Все процедуры прерываний должны оканчиваться командой IRET.

Если в МП поступает одновременно несколько запросов на прерывания, он обслуживает их в соответствии с приоритетом:

Прерывание по флагу TF (высший приоритет)
Немаскируемое прерывание NMI
Программное прерывание INT vect
Векторное прерывание INTR (низший приоритет)

Эта последовательность нарушается, когда одновременно три запроса с высшим приоритетом (за исключением INTR) активны. В данном случае последовательность обработки следующая:

Программное прерывание INT vect
Немаскируемое прерывание NMI
Прерывание по флажку TF

Управление захватом шины. Микропроцессор имеет мощные программно-аппаратные средства управления захватом шины. Эти средства применяются для построения каналов ПДП и мультипроцессорных МС.

В минимальном режиме для этих целей используются вход HOLD принимающий запросы доступа к шине, и выход HLDA, генерирующий сигнал подтверждения доступа. Вход HOLD является асинхронным и опрашивается по фронту каждого тактового импульса. При поступлении напряжения высокого уровня МП в середине последнего такта текущего цикла канала T4 или холостого такта T5 выдает сигнал подтверждения захвата по выходу HLDA. Одновременно с этим МП переводит в высокоомное состояние шину AD, шины $\overline{BHE}$/S7, A19/S6 — A16/S3 и управляющие линии $\overline{RD}$, $\overline{WR}$, DT/$\overline{R}$, M/$\overline{IO}$, $\overline{DEN}$, логически отключаясь от магистрали. При снятии сигнала HOLD МП в следующем такте сбрасывает сигнал подтверждения HLDA и возвращает себе управление магистралью (рис. 6.12а).

В максимальном режиме для управления захватом шины МП служат линии $\overline{RQ}$/$\overline{GT0}$, $\overline{RQ}$/$\overline{GT1}$ и $\overline{LOCK}$.

Двунаправленные линии $\overline{RQ}$/$\overline{GT0}$ и $\overline{RQ}$/$\overline{GT1}$ имеют одно и то же функциональное назначение, однако приоритет у линии $\overline{RQ}$/$\overline{GT0}$ выше. Процедура захвата шины в максимальном режиме состоит из трех этапов: запроса, разрешения, освобождения

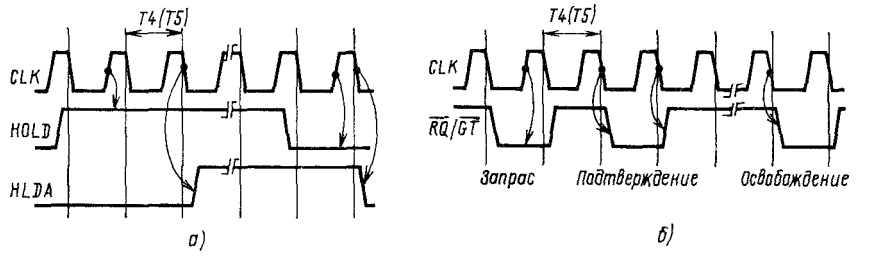


Рис. 6.12. Временные диаграммы захвата шины для минимального (а) и максимального (б) режимов

(рис. 6.12, б). Процедура на захват шины инициируется внешним модулем подачей на один из входов $\overline{RQ}/GT0$ или $\overline{RQ}/GT1$ импульса длительностью Т. В конце текущего машинного цикла или после очередного холостого такта Т5 МП генерирует ответный импульс, подтверждающий захват. Этот импульс передается по той же линии и имеет длительность Т. В следующем такте МП переводит выходы $\overline{BHE}$/S7, A19/S6—A16/S3, AD15—AD0, а также линии $\overline{S2}$—$\overline{S0}$ LOCK и $\overline{RD}$ в третье состояние, логически отключаясь от шины. После окончания цикла обмена внешний модуль генерирует еще один импульс длительностью Т, который извещает о возвращении управления шиной спустя два такта. Обмен импульсами должен происходить синхронно. Запрос по входу $\overline{RQ}/GT1$, пришедший одновременно с запросом $\overline{RQ}/GT0$, будет разрешен спустя один-два такта после освобождения магистрали. Запросы на захват шины имеют более высокий приоритет, чем прерывания.

$\overline{LOCK}$ — инверсный программно-управляемый выход блокировки захвата шины. Сигнал вырабатывается по префиксу LOCK и поддерживается в течение всего цикла выполнения команды, следующей за ним. При блокировке МП не воспринимает запросы на захват шины, обеспечивая себе возможность единоличного доступа к памяти. Такой тип доступа важен при организации семафоров. Сигнал блокировки $\overline{LOCK}$ не влияет на обработку запросов прерывания. В минимальном режиме сигнал $\overline{LOCK}$ отсутствует. Однако префикс LOCK все же используется для задержки выдачи подтверждения сигнала HLDA на время выполнения команды.

6.7. Структурные отличия ВМ88

Архитектура МП ВМ88 тождественна архитектуре ВМ86. Программное обеспечение одного МП может быть использовано другим без изменения. Отличия в их организации наблюдаются только на структурном уровне.

Схема микроЭВМ, построенной на основе МП ВМ88, приведена на рис. 6.13. Она, как и схема на рис. 6.1, содержит 20-разрядную шину физического адреса ADR, шину данных DAT и шину управления СВ. В системе используется физическая память емкостью до 1 М байта и изолированная система ВВ с пространством портов до 64 К байт. Однако обмен с памятью и внешними приборами ВВ выполняется через 8-разрядную шину данных DAT по байтам. По этой причине в магистрали отсутствует линия $\overline{\text{BHE}}$, которая предназначена для управления передачей данных через старшую часть шины данных DAT15—DAT0. Там, где в МС (см. рис. 6.1) требовался один цикл обращения к магистрали, теперь необходимы два таких цикла.

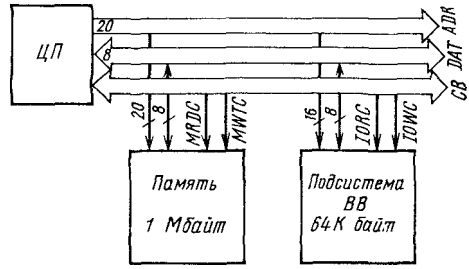


Рис. 6.13. Схема микроЭВМ на базе ВМ88

С другой стороны, системная шина на рис. 6.13 полностью эквивалентна шине микроЭВМ на рис. 2.1, построенной на базе МП ВМ80. Это означает совместимость внешних по отношению к ЦП аппаратных модулей как одной, так и другой МС.

Представленная на рис. 6.14 структурная схема МП ВМ88 аналогична схеме МП ВМ86 (см. рис. 6.5), за исключением того, что внешний обмен данными выполняется по 8 бит. Шестнадцатиразрядные операнды считываются или записываются при помощи двух последовательных циклов обращения к магистрали. Поэтому производительность ВМ88 несколько хуже, чем у ВМ86, а в остальном с точки зрения программиста оба процессора не различимы.

В процессоре ВМ88 длина очереди уменьшена до 4 байт, тогда как в ВМ86 она составляет 6 байт или 3 слова. Длина очереди была сокращена для уменьшения времени занятия системной магистрали блоком сопряжения, необходимого для заполнения очереди. Вместе с этим был оптимизирован алгоритм предварительной выборки. Так, если в ВМ86 новое слово программного кода считывается из памяти и вводится в буфер всякий раз, когда в очереди освобождается два байта, то в ВМ88 новый байт программного кода пересылается в буфер при наличии в нем хотя бы одного места. Алгоритм такого типа сглаживает возможные колебания длины очереди, обеспечивая практически постоянное ее заполнение.

Блоки обработки обоих процессоров совершенно идентичны и работают с одинаковыми скоростями. Поэтому скорость работы ЦП на базе ВМ88 ограничивается производительностью его блока сопряжения. Поддержка всегда заполненной очереди команд

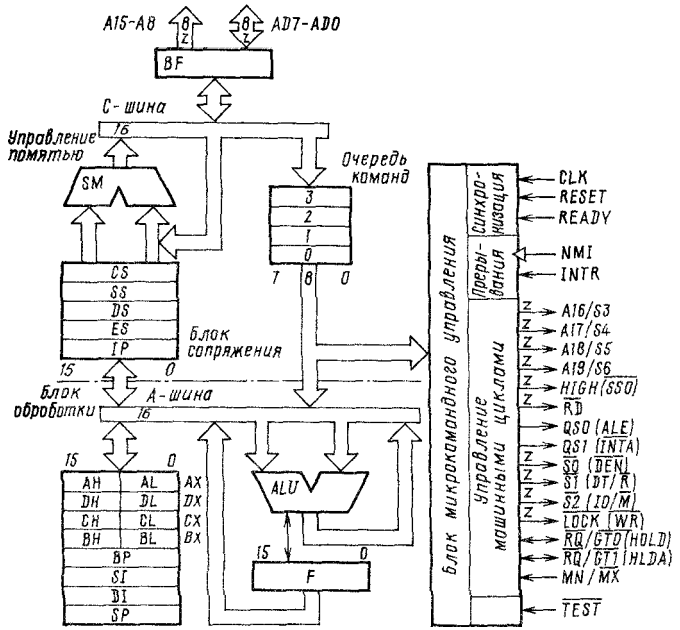


Рис. 6.14. Схема микропроцессора ВМ88

заставляет ЦП работать с максимальной для него скоростью, определяемой скоростью работы блока обработки. Однако в случае следующих друг за другом ряда простых команд очередь может быстро опустеть и производительность ЦП будет определяться скоростью выборки команд из памяти.

Микропроцессор ВМ88, подобно ВМ86, может работать как в минимальном, так и в максимальном режиме, в соответствии с этим меняется состав и функциональное назначение его аппаратного интерфейса. Программирование режима выполняется с помощью входа MN/$\overline{\text{MX}}$. Внесенные в структуру ВМ88 изменения повлияли и на его интерфейс, который несколько отличается от интерфейса ВМ86.

Во-первых, двунаправленные линии AD15—AD8 заменены на однонаправленную шину A15—A8, служащую только для выдачи адресной информации. Сигналы A15—A8 запоминаются внутри ЦП и выдаются на одноименные линии в продолжении всего машинного цикла работы с шиной аналогично старшим адресным линиям ЦП ВМ85ГА.

Во-вторых, отсутствует необходимость в сигнале $\overline{\text{BHE}}$. В максимальном режиме освободившаяся линия не используется. На ней всегда поддерживается напряжение высокого уровня. В минимальном режиме через данную линию выводится сигнал $\overline{\text{SS0}}$, функционально эквивалентный сигналу $\overline{\text{S0}}$ максимального

режима, но с другими временны́ми параметрами. Его временные параметры совпадают с параметрами сигнала IO/M. Сигнал SS0 совместно с сигналами DT/$\overline{R}$ и IO/$\overline{M}$ обеспечивает полную информацию о типе текущего машинного цикла в соответствии с табл. 6.19. Отметим, что сигнал IO/$\overline{M}$ инвертирован по отношению к однотипному сигналу МП ВМ86. Это сделано для обеспечения совместимости с интерфейсом ВМ85А. Существует еще одно отличие ВМ88 в минимальном режиме. При входе в состояние останова сигнал ALE задерживается на один такт, чтобы позволить внешним схемам запомнить информацию о состоянии.

Процессор ВМ88 размещается в стандартном 40-выводном корпусе с двурядным расположением выводов. Условное графическое обозначение микросхемы приведено на рис. 6.15. Функциональное назначение выводов микросхемы совпадает с функциональным назначением тех же выводов для ВМ86 (ср. с рис. 6.6). Это упрощает проектирование МС на их основе, дает возможность ввести процедуры автоматического распознавания типа МП. Такого рода распознавания важны для компонентов внешнего расширения процессоров, работающих в максимальном режиме.

Процедура распознавания и настройки на конкретный МП выполняется при включении напряжения питания или нажатии

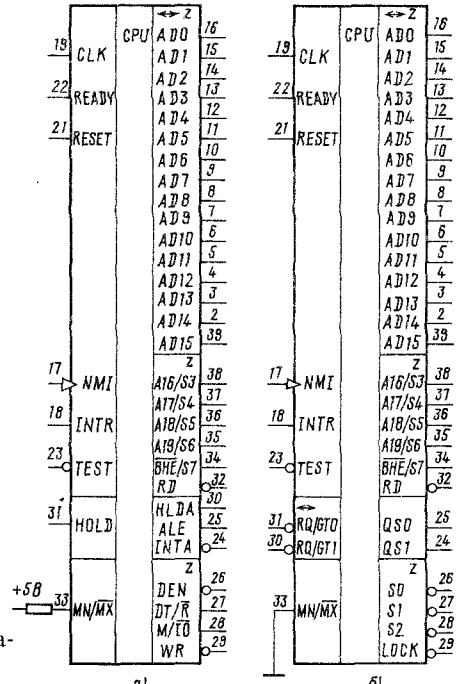


Рис. 6.15. Условное графическое обозначение микропроцессора ВМ88:
а — минимальный режим; б — максимальный режим

клавиши RESET, когда управление передается на стартовый адрес МП. В качестве стартового входа в обеих микросхемах используется логический адрес 0FFFFH:0000H. Однако в первом цикле обращения к магистрали МП ВМ86 считывает первое слово программной последовательности ($\overline{BHE}=0$), тогда как в ВМ88 эта линия всегда находится в состоянии HIGH ($\overline{BHE}=1$). Состояние линии $\overline{BHE}$ в первом машинном цикле после сброса однозначно связывается с типом МП.

Конфигурация локальной шины ВМ88 полностью совпадает с мультиплексированным интерфейсом ПУ семейства ВМ85А, что позволяет разработчику создавать системы с минимальным числом корпусов. Подобная конфигурация дала возможность использовать всю вычислительную мощность процессоров ВМ88 совместно с высокоинтегрированными периферийными кристаллами семейства ВМ85А.

Таблица 6.19

Состояние			Тип цикла	Состояние			Тип цикла
IO/$\overline{M}$	DT/$\overline{R}$	$\overline{SS0}$		IO/$\overline{M}$	DT/$\overline{R}$	$\overline{SS0}$	
1	0	0	*Подтверждение прерываний*	0	0	0	Выборка команды
1	0	1	Чтение порта ВВ	0	0	1	Чтение памяти
1	1	0	Запись в порт ВВ	0	1	0	Запись в память
1	1	1	Останов	0	1	1	Пассивное состояние

ГЛАВА 7.

ОРГАНИЗАЦИЯ ОДНОПЛАТНЫХ МИКРОКОНТРОЛЛЕРОВ НА БАЗЕ К1810ВМ86

7.1. Принцип построения

Главной структурной особенностью современных МС является магистрально-модульный принцип их построения, регламентирующий способ межмодульных связей. Согласно этому принципу МС разбивается на ряд функционально-законченных устройств — модулей. Связь между модулями реализуется с помощью единой внутрисистемной магистрали, что подразумевает общий для всех модулей состав шин, единый способ представления информации на них и общие правила исполнения всех процедур передачи информации через шину.

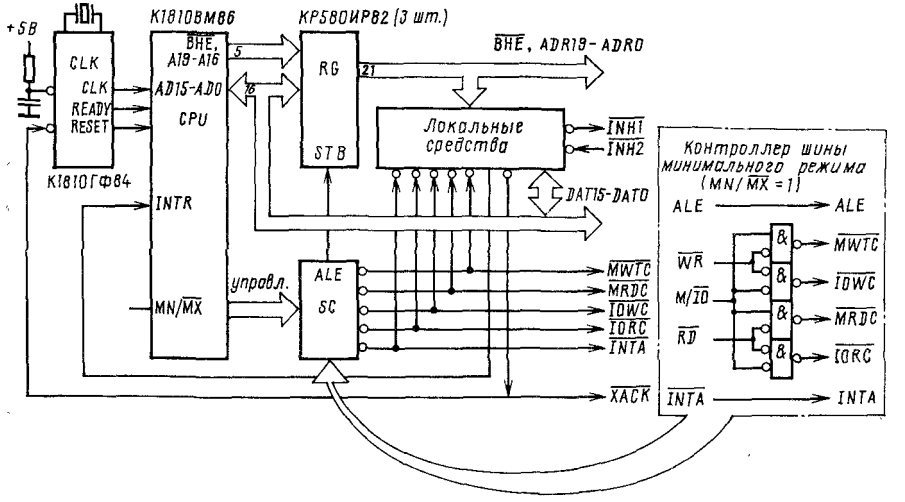


Рис. 7.1. Схема микросистемы на базе микропроцессора К1810ВМ86

На рис. 7.1 представлена обобщенная схема МС на базе МП К1810ВМ86. Уникальным свойством этого микропроцессора является возможность выбора с помощью входа MN/$\overline{\text{MX}}$ одного из двух его рабочих режимов, в наибольшей степени подходящего к конкретному применению. В соответствии с этим меняется и логика работы контроллера шины.

Минимальный режим (MN/$\overline{\text{MX}}$=1) оптимизирован для малых МС с одним процессором. Микропроцессор практически непосредственно обеспечивает управление системной магистралью И41 (см. рис. 6.1).

Максимальный режим (MN/$\overline{\text{MX}}$=0) предназначен для создания мультипроцессорных систем различной конфигурации. В этом режиме МП формирует промежуточную локальную шину, для преобразования которой в системную магистраль И41 требуется более сложная логика. Данная логика реализуется однокристальным системным контроллером К1810ВК88.

В качестве генератора используется БИС К1810ГФ84 (ГФ84). Усиление и буферизация шины адреса выполняют три буферных регистра типа КР580ИР82/ИР83. Для двунаправленной буферизации шины данных (на рис. 6.1 не показано) могут быть использованы две микросхемы КР580ВА86/ВА87. Приведенный на рис. 6.1 ЦП по структуре аналогичен ЦП на базе МП КР580ВМ80 (см. рис. 2.17). Он также формирует системную магистраль, совместимую с магистралью И41 [33], включая линию $\overline{\text{BHE}}$.

Один из практически возможных вариантов подключения к 16-разрядной внутримодульной магистрали типа И41 ПЗУ, ОЗУ

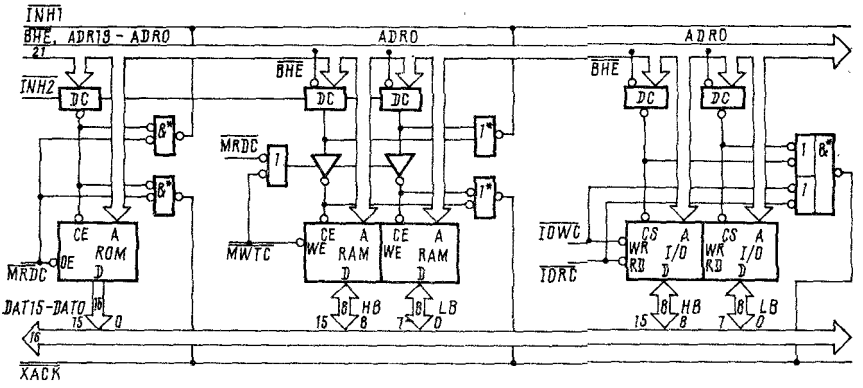


Рис. 7.2. Схема подключения локальных средств

статического типа, а также периферийных БИС ВВ приведен на рис. 7.2. Конфигурации такого вида могут быть применены для построения одноплатных микроЭВМ на базе МП К1810ВМ86.

Схема на рис. 7.2 включает устройство выборки БИС, которое генерирует сигналы выборки БИС $\overline{CS}$ и $\overline{CE}$. Оперативная память и область ВВ содержат по два 8-разрядных банка каждый. Выбор банка осуществляется сигналами $\overline{BHE}$ (H-банк) и $\overline{ADR0}$ (L-банк), в результате реализуется обмен либо байтами, либо словами. В состав интерфейса шины также входит логика подтверждения операции обмена, формирующая сигнал $\overline{XACK}$, и логика запрета, основанная на сигналах $\overline{INH1}$ и $\overline{INH2}$. Сигналы подтверждения $\overline{XACK}$ и запрета $\overline{INH1}$, $\overline{INH2}$ формируются по схеме «монтажное ИЛИ».

В схеме использованы типовые варианты приборов памяти и ВВ. Микросхемы ПЗУ, так же как и периферийные БИС имеют линию $\overline{OE}$ или $\overline{RD}$ для стробирования выхода при чтении данных. Интерфейсом такого типа обладают УСППЗУ серии К573 и ППЗУ серии К556, а также периферийные БИС ВВ серий КР580 и К1810. Вместе с тем большинство ОЗУ статического типа, например К541РУ1/РУ2, К537РУ3, не имеют линии для приема команды чтения. Это усложняет логику их подключения к магистралям типа И41. Другой тип микросхем, например К537РУ8 или К537РУ9, имеющий вместе с линией выборки кристалла вход $\overline{OE}$ для команды чтения, подключается к шине так же, как БИС ВВ.

7.2. Генератор тактовых импульсов ГФ84

Микросхема К1810ГФ84 [61] является однокристальным ГТИ, специально спроектированным для МП К1810ВМ86. В состав микросхемы (рис. 7.3) входит

стабилизируемый кварцем генератор, делитель на 3, а также логика синхронизации сигналов готовности и сброса.

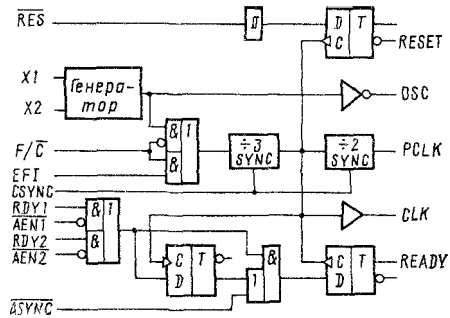


Рис. 7.3. Схема генератора тактовых импульсов К1810ГФ84

Встроенный в БИС генератор рассчитан для работы с внешним кварцевым резонатором, который подключается к входу X1 и выходу X2. Частота кварцевого резонатора должна быть в 3 раза больше, чем требуемая МП частота CLK. Для достижения наиболее стабильной работы входных цепей генератора рекомендуется точки X1 и X2 подключить к общей шине через резисторы сопротивлением 510 Ом. Полезно также последовательно с кварцевым резонатором подключить конденсатор небольшой емкости (около 15 пФ). Выход генератора подключается непосредственно к выводу OSC, так что внешние средства при необходимости могут воспользоваться основной тактовой частотой ГТИ.

Основная системная частота CLK получается путем деления частоты OSC на 3. Длительность импульсов на выходе CLK составляет 1/3 периода, что соответствует требованиям, предъявляемым МП. Еще один выходной синхросигнал PCLK представляет собой меандровую последовательность импульсов с частотой, равной половине частоты CLK. В случае стандартного значения частоты CLK 5 МГц, частота PCLK составляет 2,5 МГц. Этот синхросигнал предназначен для формирования основной тактовой последовательности CCLK для ПУ. Установочный вход CSYNC позволяет синхронизировать CLK и PCLK, например, с другим ГТИ.

Микросхема предусматривает возможность работы от внешнего генератора импульсов OSC, которые подаются на вход EFI. Выбор между внутренним и внешним генератором OSC реализуется с помощью управляющего входа F/$\overline{C}$. При F/$\overline{C}$=0 выбирается внутренний генератор, в противном случае — внешний.

Логика формирования сигнала сброса RESET включает триггер Шмитта и синхронизирующий D-триггер, срабатывающий по срезу CLK. Применение на входе RES пороговой схемы позволяет формировать сигнал сброса при нажатии клавиши RESET с помощью простой RC-цепочки подобно тому, как это сделано на рис. 2.17 при использовании БИС КР580ГФ24. Полученный на выходе RESET сигнал сброса соответствует требованиям, накладываемым на него МП К1810ВМ86 (см. § 6.6).

Логика формирования сигнала готовности READY предусматривает синхронизацию входного сигнала готовности RDY1 или RDY2. Выбор входа осуществляется управляющими сигналами $\overline{AEN1}$ и $\overline{AEN2}$. Симметричность пары RDY, $\overline{AEN}$ допускает использование в качестве входа готовности инверсную линию $\overline{AEN}$, тогда как прямая линия RDY может служить управляющей.

Различают два типа входных сигналов готовности: асинхронный и синхронный, в соответствии с которыми предусматриваются и два типа синхронизации. Выбор типа синхронизации осуществляется по уровню напряжения на линии $\overline{ASYNC}$. При $\overline{ASYNC}$=0 реализуется двухступенчатая логика синхронизации асинхронного сигнала готовности, в противном случае — одноступенчатая логика синхронизации синхронного сигнала готовности.

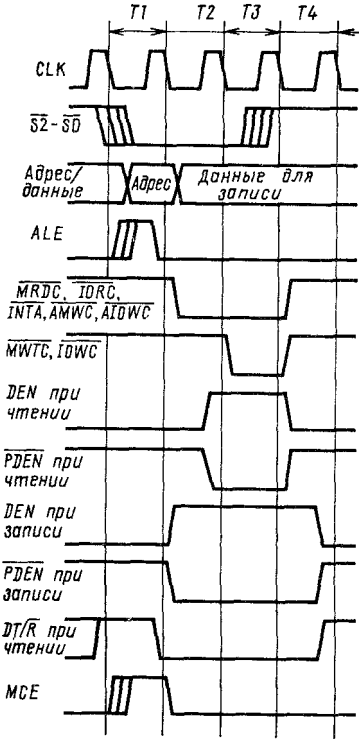


Рис. 7.5. Схема системного контроллера К1810ВГ88

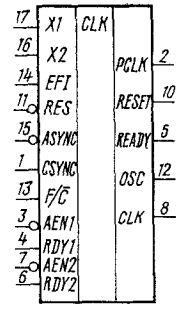


Рис. 7.4. Условное графическое обозначение генератора тактовых импульсов К1810ГФ84

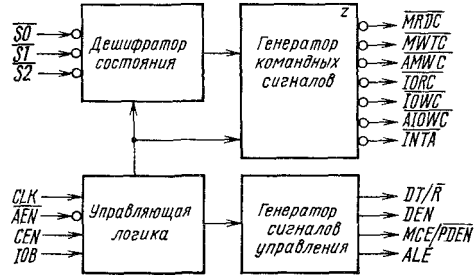


Рис. 7.6. Временные диаграммы работы системного контроллера К1810ВГ88

В случае двухступенчатой синхронизации ($\overline{ASYNC}=0$) переход входного сигнала готовности из 0 в 1 будет синхронизироваться сначала по фронту, а затем по срезу CLK. Переход входного сигнала из 1 в 0 будет всегда синхронизироваться только по срезу CLK. Такая схема синхронизации ориентирована на использование с асинхронными системными каналами с неготовым по умолчанию сигналом ответа $\overline{XACK}$.

При одноступенчатой синхронизации ($\overline{ASYNC}=1$) входной сигнал готовности тактируется только срезом CLK. Этот способ применим в системах, которые гарантируют синхронность формирования сигнала подтверждения обмена или готовности.

Вход $\overline{ASYNC}$ имеет встроенный резистор, подключенный к шине питания. Поэтому допускается вход $\overline{ASYNC}$ оставлять свободным, что соответствует ситуации $\overline{ASYNC}=1$.

Микросхема К1810ГФ84 упакована в 18-выводный корпус типа 2104.18. Условное обозначение и распределение сигналов по выводам приведено на рис. 7.4.

7.3. Системный контроллер ВГ88

Микросхема К1810ВГ88 [61] является однокристальным устройством, предназначенным для согласования локального интерфейса МП К1810ВМ86/ВМ88, работающего в максимальном режиме, с системной магистралью типа И41. Системный контроллер (рис. 7.5) содержит дешифратор состояния, управляющую логику, генератор командных сигналов и генератор сигналов управления. Работа БИС синхронизируется тактовой последовательностью CLK, которая используется для синхронизации МП (рис. 7.6).

Дешифратор состояния фиксирует генерируемые МП сигналы состояния $\overline{S2}$—$\overline{S0}$ (Status) и декодирует их для определения типа выполняемого цикла. Фиксация очередного состояния осуществляется по срезу сигнала S2VS1VS0, формируемого внутренней логикой дешифратора.

Декодированные в соответствии с табл. 6.16 сигналы состояния используются для генерации ряда командных стробов:

$\overline{MRDC}$	Чтение памяти
$\overline{MWTC}$	Запись в память
$\overline{AMWC}$	Запись в память с упреждением
$\overline{IORC}$	Чтение портов
$\overline{IOWC}$	Запись в порты
$\overline{AIOWC}$	Запись в порты с упреждением
$\overline{INTA}$	Подтверждение прерывания

В зависимости от числа и способа формирования команд различают два основных режима работы контроллера: режим системной шины и режим шины ВВ, программируемые с помощью управляющего вывода IOB (Input/Output Bus).

В *режиме системной шины* (IOB = 0) все семь стробов возможны. Однако их генерация разрешается только при активном сигнале $\overline{AEN}$ (Address Enable), который обычно формирует внешняя логика шинного арбитража. Управляющий сигнал $\overline{AEN}$ может рассматриваться как сигнал разрешения доступа к шине, обеспечивающий возможность параллельной работы на одной магистрали нескольких активных устройств.

В *режиме шины ВВ* (IOB = 1) разрешается генерация только четырех команд: $\overline{IORC}$, $\overline{IOWC}$, $\overline{AIOWC}$ и $\overline{INTA}$. При этом их формирование уже не зависит от состояния сигнала AEN, т. е. функция арбитража в данном режиме не поддерживается. Вместе с командами системный контроллер генерирует ряд сигналов, управляющих формирователем шины данных и буферными регистрами шины адреса:

ALE	Строб фиксации адреса
$\overline{DEN}$	Разрешение работы драйвера шины данных
DT/$\overline{R}$	Прием/передача данных
MCE/$\overline{PDEN}$	Разрешение работы главного модуля или разрешение работы драйвера шины ВВ

Строб ALE (Address Latch Enable) разрешает фиксацию адреса в буферных регистрах МС (ALE = 1). Сигнал DEN (Data Enable) определяет временной ин-

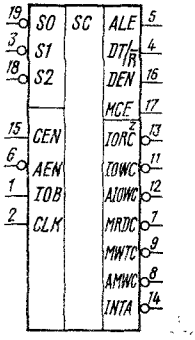


Рис. 7.7. Условное графическое обозначение контроллера К1810ВГ88

тервал подключения локальной шины МП к внешней шине (DEN=1), а сигнал DT/$\overline{R}$ (Data Transmission/Receipt) управляет направлением передачи данных от МП (DT/$\overline{R}$=1) или к МП (DT/$\overline{R}$=0). Обычно эти сигналы используются для управления работой двунаправленного формирователя шины данных. Вывод MCE/$\overline{PDEN}$ имеет две функции, которые соответствуют двум режимам работы контроллера.

В *режиме системной магистрали* (IOB=0) генерируется сигнал MCE (Master Slave Enable), который действует только в цикле подтверждения прерывания. Он информирует о работе ведущего контроллера прерываний и может быть использован при организации каскадного расширения в том случае, когда ведущий контроллер находится на локальной шине МП, а подчиненные — на системной шине. (Более подробно об этом см. § 7.6.)

В *режиме шины ВВ* (IOB=1) вывод выполняет функцию $\overline{PDEN}$ (Peripheral Data Enable). Сигнал $\overline{PDEN}$ разрешает передачу данных с или на шину ВВ, служащую шиной ПУ. Временные параметры сигнала такие же, как у управляющего сигнала DEN. Еще один входной управляющий сигнал CEN (Command Enable) служит для перевода командных линий в третье состояние (CEN=0).

Микросхема К1810ВГ88 выполнена по биполярной технологии и упакована в 20-выводный кристаллодержатель 2140Ю.20 с вертикальным расположением выводов (рис. 7.7). Для работы БИС требуется один источник питания +5 В, ток потребления $I_{CC} \leqslant 230$ мА. Нагрузочная способность командных линий: $I_{OL}=32$ мА, $C_L=300$ пФ, линий управления: $I_{OL}=16$ мА, $C_L=80$ пФ.

7.4. Центральный процессор на базе БИС серии К1810

На рис. 7.8 представлена схема ЦП на базе МП 1810ВМ86. В ее основе лежит типовая 8-кристальная структура [34], в состав которой кроме МП входят ГТИ К1810ГФ84, системный контроллер К1810ВК88, реализованный на трех микросхемах КР580ИР82 21-разрядный адресный регистр и двунаправленный 16-разрядный формирователь шины данных на двух микросхемах КР580ВА86.

В представленной схеме ЦП используется максимальный режим работы МП. Это сделано с целью унификации внутримодульной шины и обеспечения постепенного перехода на сопроцессорные расширения центрального МП. В качестве элементов расширения рассматриваются сопроцессор числовой обработки К1810ВМ87 и сопроцессор 80130 операционной системы RMX86 [62]. Линии RQ/$\overline{GT}$ и $\overline{TEST}$ резервируются для такого типа расширений ЦП.

Микропроцессор ВМ86 требует наличия синхросигналов CLK с крутыми фронтами ($\leqslant 10$ нс) и скважностью 3, частота следования которых 2—5 МГц. Существование минимального значения частоты следования импульсов CLK объясняется тем, что в МП ВМ86 использованы запоминающие элементы динамического типа, поэтому какая-либо манипуляция с синхросигналом запрещена.

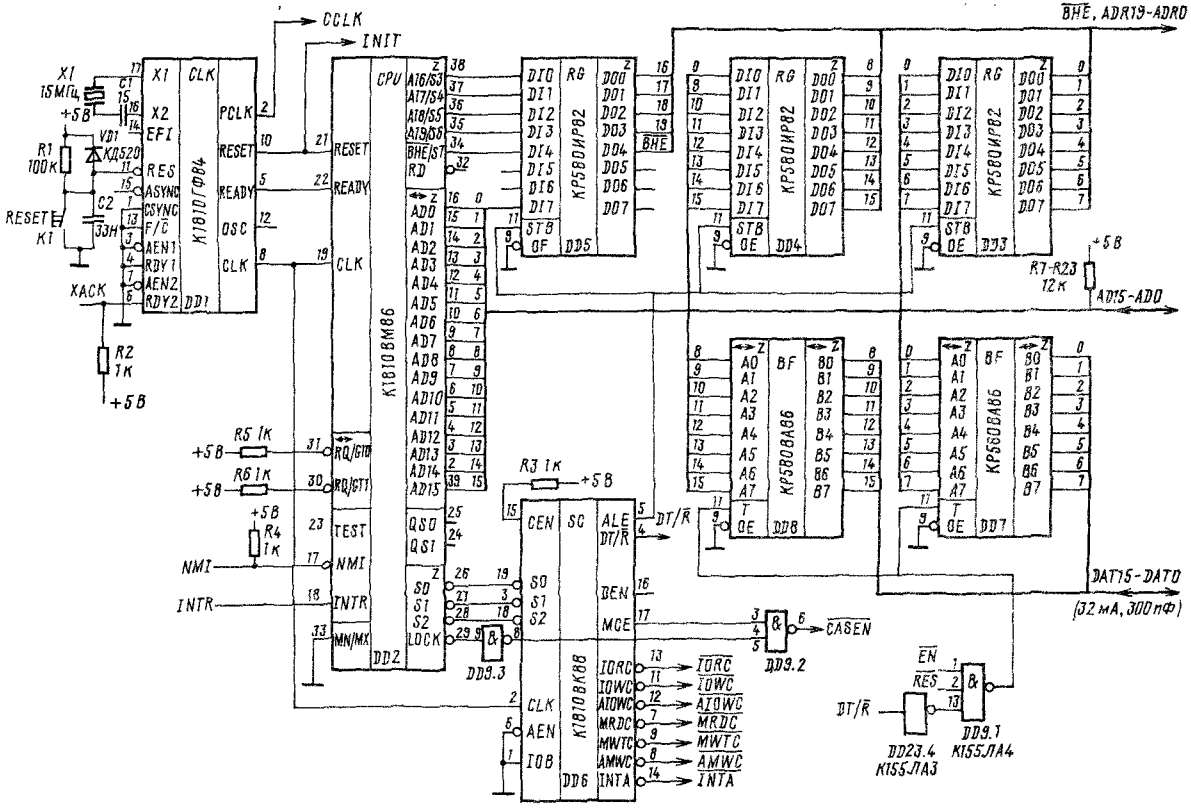


Рис. 7.8. Схема центрального процессора на базе К1810ВМ86

Всем этим требованиям удовлетворяет тактовый сигнал, сформированный ГТИ на базе К1810ГФ84. Устройство работает от кварцевого резонатора (режим $F/\overline{C}=0$), имеющего в 3 раза большую частоту колебаний, чем желаемая для CLK. Для получения более точного и стабильного сигнала рекомендуется использовать основную частоту кварцевого резонатора.

Между кварцевым резонатором и выводом X2 следует включить конденсатор небольшой емкости (около 15 пФ), которая должна быть такой, чтобы коэффициент усиления цепи обратной связи генератора не был меньше 1. Для этого необходимо, чтобы сумма сопротивлений кварцевого резонатора и конденсатора на рабочей частоте не превышала 1 кОм. (Объяснение причин включения дополнительной емкости дано в § 2.6.)

Генерируемый ГТИ тактовый сигнал ПУ PCLK (он же — сигнал системных тактовых импульсов CCLK, выходящий на системную магистраль) имеет скважность 2 и частоту колебаний, равную половине частоты CLK. Вариант прямого использования PCLK в качестве системных тактов обеспечивает выходной ток $I_{OL} \leqslant 5$ мА и допустимую емкость нагрузки $C_L \leqslant 100$ пФ.

Для общего сброса МП ВМ86 требует подачи на свой вход RESET импульса длительностью более четырех периодов тактовой частоты CLK, за исключением случая включения источника питания, когда длительность импульса должна составлять как минимум 50 мкс. Генератор ГФ84 имеет встроенную логику, упрощающую формирование данного сигнала. Для формирования сигнала RESET достаточно на входе $\overline{RES}$ организовать простую RC-цепочку (рис. 7.8), которая при подаче напряжения питания или нажатии клавиши RESET автоматически обеспечивает формирование импульса сброса необходимой длительности. Этот импульс также используется в качестве сигнала общего сброса INIT всей системы с максимальным выходным током $I_{OL}=5$ мА. Для увеличения нагрузочной способности линии сброса INIT используются дополнительные магистральные усилители.

Еще одной функцией, которую может поддерживать ГТИ, является прием и синхронизация двух сигналов готовности ПУ к обмену. Обычно один сигнал готовности поступает от локальной шины МП, а другой — от системной магистрали.

Условие готовности: $RDY1 \cdot \overline{AEN1} + RDY2 \cdot \overline{AEN2} = 1$,

где RDY1, RDY2 — сигналы готовности от ПУ, а $\overline{AEN1}$ и $\overline{AEN2}$ — управляющие сигналы, определяющие, с какой шиной в данном машинном цикле работает МП. Линии готовности обычно формируются по схеме с открытым коллектором и могут иметь как высокий (ХАСК), так и низкий ($\overline{ХАСК}$) уровень активности. (Преимущества и недостатки применения того или иного варианта обсуждались в § 2.6.) Используя симметрию элементов пары

RDY, AEN, пользователь выбирает соответствующий вход (прямой или инверсный) для приема сигнала готовности от ПУ. Оставшийся служит для вспомогательных целей, например для управления выбором шины. В схеме на рис. 7.8, ориентированной на одномагистральную архитектуру, используется одна линия готовности или подтверждения к обмену ХАСК с высоким уровнем активности (готов по умолчанию).

Дополнительную возможность обеспечивает вход ASYNC, управляющий способом синхронизации принятого сигнала готовности. При ASYNC=0 выполняется двухступенчатая синхронизация сигнала по фронту и срезу тактовой частоты CLK. Если на ASYNC подается напряжение высокого уровня, то реализуется процедура одноступенчатой синхронизации по срезу CLK. Ориентация на линию ХАСК с высоким уровнем активности требует применения одноступенчатого способа синхронизации, когда обусловленная процедурой синхронизации задержка сигнала готовности минимальна. В такте Т3 каждого машинного цикла МП опрашивает вход готовности READY. Для правильного приостанова МП в текущем цикле необходимо, чтобы сигнал ХАСК был установлен в 0 за 35 нс до начала Т3.

Еще одна линия CSYNC (внешняя синхронизация) позволяет нескольким ГТИ ГФ84 синхронизировать свою работу. В простейшем случае вывод CSYNC должен быть заземлен.

Ориентация на максимальный режим работы МП ВМ86 потребовала введения в состав ЦП системного контроллера ВК88, однако возможна достаточно простая имитация его работы (рис. 7.9). Не исключен также вариант перехода к минимальному режиму работы, когда требуемая для формирования сигналов управления системной шины дополнительная логика имеет минимальный объем.

Системный контроллер ВК88 запрограммирован для работы в режиме системной магистрали (IOB=0). Напряжение низкого уровня на входе разрешения адреса AEN указывает контроллеру на монопольное владение системной шиной, благодаря чему команды магистрали формируются контроллером с минимально возможными задержками.

Контроллер ВК88 работает синхронно с МП ВМ86. Он принимает код текущего машинного цикла S2—S0 (см. табл. 6.16) и формирует весь набор сигналов, необходимый для его выполнения. Временные диаграммы циклов ввода и вывода в максимальном режиме без тактов ожидания приведены на рис. 7.10.

В соответствии с кодом S2—S0 системный контроллер формирует пять стандартных управляющих линий IORC, IOWC, MRDC, MWTC и INTA, нагрузочная способность которых

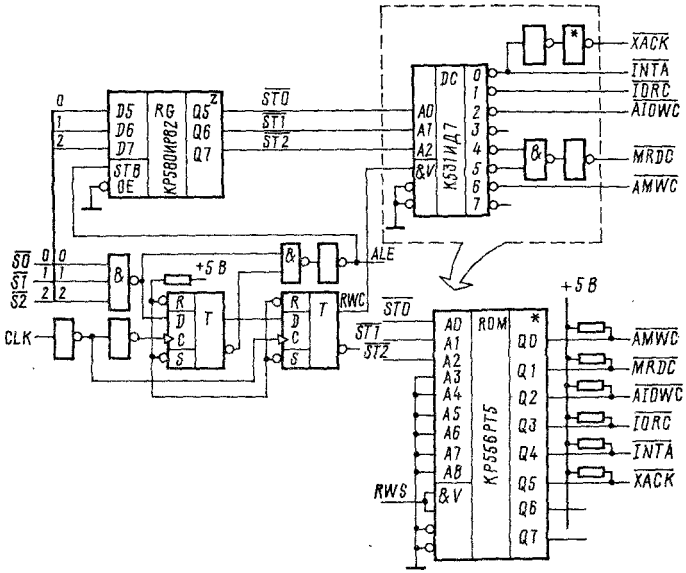


Рис. 7.9. Схема контроллера шины для ВМ86, работающего в максимальном режиме (MN/$\overline{MX}$=0)

$I_{OL} \leqslant 32$ мА, $C_L \leqslant 300$ пФ. Две линии упреждающего управления $\overline{AIOWC}$ и $\overline{AMWC}$ имеют ту же нагрузочную способность. Выбор между стандартным и упреждающим вариантами соответствующей линии управления возлагается на пользователя. Стробы $\overline{IOWC}$ и $\overline{MWTC}$ (см. рис. 7.10,б) генерируются в третьем такте машинного цикла, поэтому для обеспечения возможности работы системной шины с тактами ожидания рекомендуется остановить свой выбор на упреждающем управлении. В данном случае время между началом строба управления и первой проверкой сигнала XACK около 130 нс. Это то время, которое предоставляется ПУ на принятие решения и установления сигнала XACK в 0 для введения тактов ожидания.

Кроме сигналов системного управления контроллер формирует ряд вспомогательных сигналов, используемых в интерфейсной логике ЦП: ALE — строб съема адреса с мультиплексных линий $\overline{BHE}$/S7, A19/S6 — A16/S3, AD15 — AD0; DT/$\overline{R}$ — сигнал управления направлением передачи через шинный драйвер; DEN — строб включения шинного драйвера; MCE — строб разрешения считывания адресной информации CAS2 — CAS0 с ведущего контроллера прерываний в цикле подтверждения прерывания. Максимальный выходной ток этих линий $I_{OL} \leqslant 16$ мА, емкость нагрузки $C_L \leqslant 80$ пФ.

По сигналу ALE адресная информация фиксируется в 21-разрядном буферном регистре, построенном на основе трех

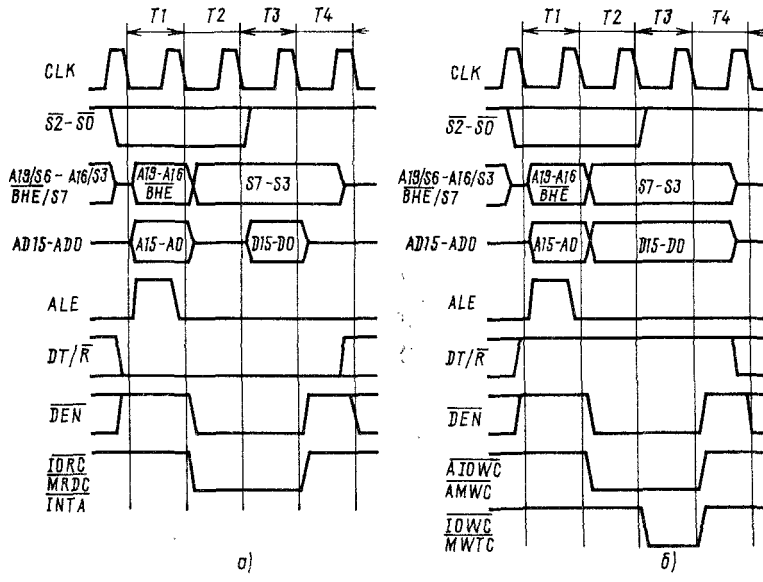


Рис. 7.10. Временны́е диаграммы циклов чтения (а) и записи (б) центрального процессора

микросхем КР580ИР82. Выходные линии регистра организуют мощную демультиплексированную адресную шину ($I_{OL} \leqslant 32$ мА, $C_L \leqslant 300$ пФ), обеспечивающую адресацию 1 М байт памяти и 64 К байт пространства ВВ. Возможна замена регистров КР580ИР82 на регистры КР580ИР83 с инверсией или на любые другие аналогичного типа.

Мультиплексная шина адреса/данных AD19—AD0 служит внутриплатной шиной данных: все резидентные ресурсы МК могут подключаться к ней. Мультиплексная шина AD19—AD0 имеет малый выходной ток $I_{OL} \leqslant 2$ мА и емкость нагрузки $C_L \leqslant 100$ пФ, что может не обеспечить в многоплатных расширениях гарантированных характеристик по переменному току.

Для устранения этого недостатка в ЦП введен двунаправленный 16-разрядный драйвер межмодульной шины данных (две микросхемы КР580ВА86). При ориентации на системную магистраль с инверсной шиной данных микросхемы КР580ВА86 должны быть заменены на КР580ВА87 или эквивалентные им.

В случае обращения к ресурсам, расположенным вне платы ($\overline{RES}=1$ и $\overline{EN}=1$), драйвер управляется сигналом DT/$\overline{R}$. В противном случае ($\overline{RES}=0$ и/или $\overline{EN}=0$) драйвер осуществляет передачу информации с мультиплексной шины на системную, что дает возможность наблюдать за процессом прохождения данных

внутри модуля. Это свойство МК может быть использовано в отладочных и тестовых целях.

7.5. Программируемый контроллер прерываний ВН59А

В настоящее время разработан новый контроллер прерываний К1810ВН59А (ВН59А), совместимый на уровне как интерфейса, так и управляющих кодов со своим предшественником ВН59 (см. § 3.6). Последний может быть заменен прибором ВН59А без внесения изменений в схему и ПО. Новый ПКП выполняет ряд функций, недоступных приборам ВН59. Важнейшая среди них — это обслуживание системы прерываний МП ВМ86/ВМ88.

Микросхема ВН59А выполнена по n-МОП-технологии, размещается в 28-выводном корпусе 2121.28 с двурядным вертикальным расположением выводов, совместима по выходам с ТТЛ-схемами. Для ее работы требуется источник питания +5 В.

Схема ПКП ВН59А аналогична схеме на рис. 3.41, за исключением вывода $\overline{SP}$. Этот вывод выполняет двойную функцию ($\overline{SP}/\overline{EN}$) и реализован в виде двунаправленной линии. В первом случае он используется как вход $\overline{SP}$ для программирования ведущего/ведомого прибора. Во втором случае он выполняет роль выходной линии $\overline{EN}$, указывающей направление передачи данных в/из прибора и необходимой для управления внешними двунаправленными шинными формирователями. Условное графическое обозначение ВН59А совпадает с ВН59 (см. рис. 3.41,б).

Перед началом работы прибор ВН59А, подобно ВН59, должен быть инициализирован в соответствии с процедурой на рис. 7.11. Данная процедура отличается от инициализации ВН59 (рис. 3.45) наличием четвертого слова ICW4, которое ранее отсутствовало. Необходимость в ICW4 программируется битом IC4 в ICW0, который для ВН59 был всегда в состоянии 0. Отсутствие ICW4 эквивалентно его принятию со всеми флажками, установленными в 0. Как будет видно далее, этот случай полностью соответствует режиму работы ВН59. Единственное исключение составляет дополнительно появившийся флажок LTIM в ICW0, программирующий тип входов IR7—IR0. Новый прибор поддерживает два типа запросов на прерывание: по переходу из 0 в 1 в соответствии с ВН59 (LTIM=0) и по высокому уровню напряжения (LTIM=1). Потенциальный вариант входов IR7—IR0 предполагает удержание 1 на линии запроса до окончания первого строба $\overline{INTA}$ и установку на ней 0 к моменту разрешения прерывания. В противном случае произойдет повторный захват того же самого сигнала прерывания.

Новое слово ICW4 обеспечивает программное управление рядом функций, отсутствовавших в ВН59. Главнейшей среди них является совместимость с МП типа ВМ86/ВМ88. Перевод ВН59А для работы в режим ВМ86 осуществляется установкой флажка ICW4. MPM в 1. В данном режиме общее функционирование прибора не изменяется, за исключением фазы ответа на стробы $\overline{INTA}$. В отличие от ВМ80/ВМ85А микропроцессоры ВМ86/ВМ88 генерируют два строба $\overline{INTA}$ (см. § 6.6). Отвечая на первый строб $\overline{INTA}$ контроллер прерываний устанавливает в 1 соответствующий бит в ISR, одновременно устанавливая его в 0 в регистре IRR, и в случае каскадного соединения выставляет идентифицирующий код ведомого контроллера на линиях CAS2—CAS0 (рис. 7.12). Никакие данные в этом цикле на шину данных не выводятся. Во время второго цикла $\overline{INTA}$ автономный или один

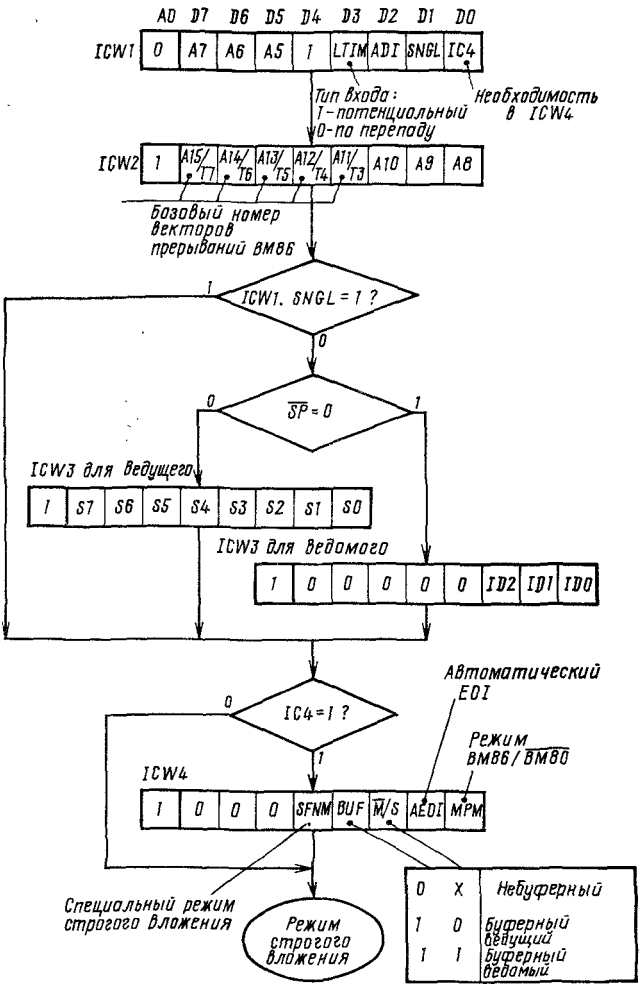


Рис. 7.11. Последовательность инициализации ВН59А

из ведомых контроллеров выдает на шину данных 8-разрядный вектор прерывания в формате, представленном на рис. 7.13. Микропроцессор ВМ86/ВМ88 использует эту информацию для вычисления адреса входа в таблицу векторов прерываний. Кроме того, в режиме ВМ86 функция ADI (управления адресным интервалом) подавляется. Состояние поля А7—А5 в ICW1, а также А10—А8 в ICW2 игнорируется. В остальном работа ВН59А подобна работе ВН59.

Следующее расширение ВН59А состоит в возможности автоматического выполнения операции конца прерывания EOI (ICW4.AEOI = 1). В данном режиме ВН59А отрабатывает неспециальную команду EOI сразу же по окончании последнего строба $\overline{\text{INTA}}$: второго в режиме ВМ86 и третьего в режиме ВМ80. В результате отпадает необходимость в генерации команды EOI. Режим AEOI может-

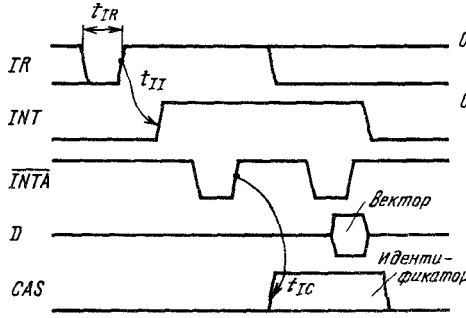


Рис. 7.12. Временные диаграммы работы BH59A в режиме BM86

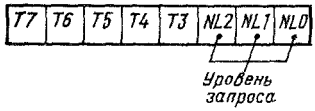


Рис. 7.13. Формат вектора прерываний

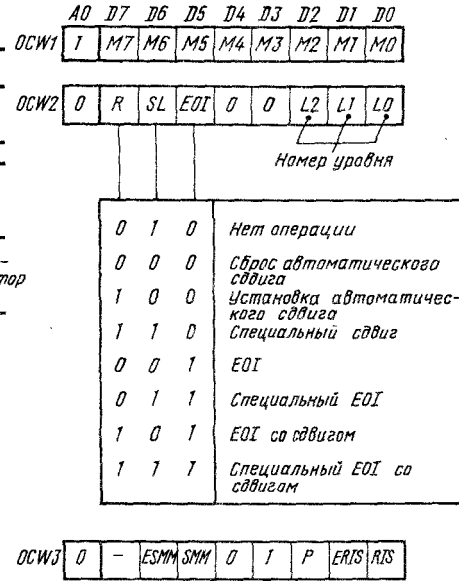


Рис. 7.14. Форматы OCW1 — OCW3 BH59A

быть применен, если не требуется строгого упорядочения приоритетов в рамках отдельного контроллера.

Чтобы добиться автоматического циклического сдвига при AEOI в BH59A предусмотрен специальный триггер. Он устанавливается в 1 словом OCW2 (рис. 7.14) при R=1, SL=0, EOI=0, а в 0 при R=0, SL=0, EOI=1. В BH59 (см. рис. 3.47) эти комбинации не использовались. В остальном функции OCW1 — OCW3 обоих приборов совпадают.

В больших системах может потребоваться дополнительный буфер, согласующий интерфейс BH59A с системной магистралью. Возникает проблема управления этим буфером, решаемая с помощью линии $\overline{SP}/\overline{EN}$. Режим буферизации программируется установкой флажка BUF в 1 в слове ICW4. В данном режиме линия $\overline{SP}/\overline{EN}$ становится выходной и на ней устанавливается низкий уровень напряжения всякий раз, когда контроллер BH59A выдает информацию на шину данных.

Подобная модификация линии $\overline{SP}/\overline{EN}$ заставляет пересмотреть правила программирования ведомых и ведущих контроллеров. Теперь для этой цели используется специальный флажок $\overline{M}/S$ в слове ICW4. Выходы $\overline{SP}/\overline{EN}$ всех приборов объединяются по схеме «монтажное ИЛИ», образуя единую линию управления буфером данных.

В приборе BH59A осуществляется специальный режим строгого упорядочения приоритетов, программируемого флажком SFNM в слове ICW4. Установка флажка SFNM в 1 снимает требование строгого упорядочения приоритетов и разрешает прием новых запросов с линии, только что принятой к обслуживанию. Обычно этот режим используется в ведущем контроллере, в то время, как ведомые контроллеры продолжают работать в рамках системы с обычным строгим

упорядочением приоритетов. В результате при получении очередного запроса от ведомого контроллера последний не блокируется, как это было ранее, и может генерировать новые запросы с более высоким приоритетом.

По окончании процедуры обслуживания прерывания программа должна проверить, было ли обслуженное прерывание единственным, исходящим от ведомого ВН59А. Это делается путем выдачи в ведомый прибор неспециальной команды EOI с последующей проверкой состояния регистра ISR. Если ISR пуст, неспециальная команда EOI посылается в ведущий контромер, в противном случае команда EOI не генерируется.

7.6. Одноплатный микроконтроллер мМС1212

В условиях широкой компьютеризации особую роль приобретает создание развитых семейств программируемых МК. Семейства МК призваны обеспечить быструю компоновку и модификацию системы с теми или иными функциональными и скоростными свойствами с целью перекрытия ряда требований, которые предъявляет к ним конкретная задача применения. Пример проектирования базового модуля такого семейства был приведен в гл. 5, где представлен универсальный 8-разрядный одноплатный МК мМС1204, построенный на базе БИС серии КР580.

В ряде прикладных случаев быстродействие в 500 тыс. операций в секунду МК типа мМС1204 оказывается недостаточным. Поэтому предлагается использовать новый высокоэффективный одноплатный 16-разрядный МК мМС1212 с быстродействием до 2 млн. операций в секунду [51, 54]. В основу МК положен однокристальный МП К1810ВМ86 повышенного быстродействия.

Особенностью МК мМС1212, как и его 8-разрядного предшественника, является тщательно отобранный функционально полный набор аппаратных средств общесистемного профиля, организованных по открытому для расширения магистрально-модульному принципу с ориентацией на стандартную шину типа И41. Открытость архитектуры МК предопределяет возможность расширения его функций модулями системного расширения из состава семейства и специальными модулями, спроектированными исключительно для конкретного применения.

Архитектурная, функциональная, электрическая и конструктивная совместимость модулей мМС1212 и мМС1204 снизу вверх позволяет считать их элементами одного семейства. Вместе с этим получили развитие новые свойства, не характерные для систем на базе МП ВМ80, но важные для многих случаев применения МП ВМ86.

Микроконтроллер мМС1212 (рис. 7.15) собран на одной плате функционально законченной МС общего назначения. В состав МК входят 16-разрядный ЦП, ПЗУ, ОЗУ и два последовательных адаптера с интерфейсом типа ИРПС, представляющие собой резидентные средства ВВ. Для осуществления работы в режиме реального времени в состав микроконтроллера включены ядро

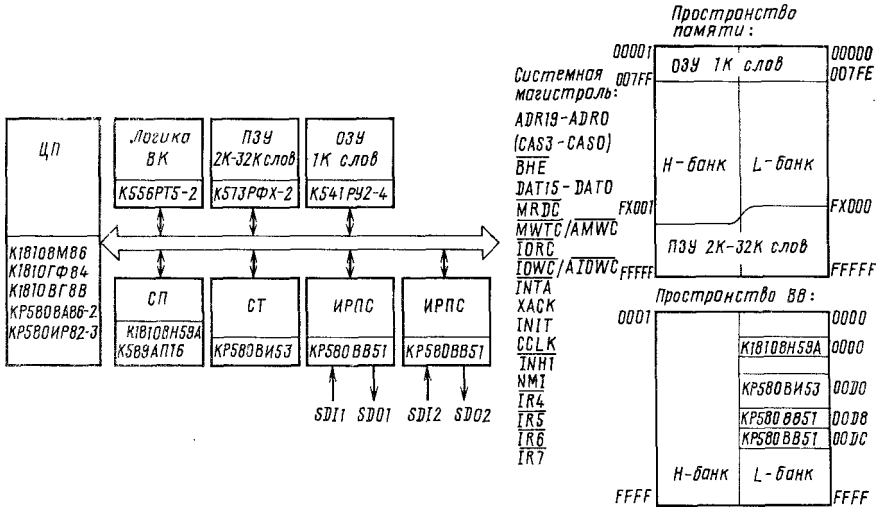


Рис. 7.15. Схема одноплатного микроконтроллера мМС1212

расширяемой многоуровневой системы прерываний (СП) и таймер общесистемного времени (СТ). Все внутримодульные средства связаны между собой с помощью стандартной 16-разрядной шины типа И41, реализующей возможность многоплатного расширения функций МК в зависимости от конкретного применения. Состав системной магистрали мМС1212 и области размещения его внутримодульных ресурсов в пространствах памяти и портов ВВ МП ВМ86 также приведены на рис. 7.15.

Центральный процессор МК построен по схеме (см. рис. 7.8), которая подробно описана в § 7.4. Процессор формирует полный набор сигналов системной магистрали и служит для управления всеми остальными модулями МК.

В составе подсистемы памяти МК мМС1212 ОЗУ используется для хранения данных, ПЗУ — для хранения констант и программного кода. ОЗУ статического типа реализовано на четырех БИС К541РУ2 (1К × 4 бит) [39] и размещено в области 00000Н — 007FFH 1М-байтового адресного пространства памяти МК. Первая половина ОЗУ резервируется под организацию 256-элементной таблицы векторов прерываний, вторая предназначена для организации системного стека и хранения ряда переменных. Реальный раздел резидентного ОЗУ между таблицей прерываний и оставшейся областью пользователя зависит от конкретного приложения МК.

Оперативное запоминающее устройство организовано в виде двух 8-разрядных банков данных по 1 Мбайт каждый. Микроконтроллер осуществляет операции чтения и записи как слова, так и

любого отдельного байта. Управление доступом производится сигналами $\overline{MRDC}$, $\overline{MWTC}$, $\overline{BHE}$ и ADR0.

На плате МК имеются рассмотренные в § 5.1 и приведенные на рис. 5.3 универсальные панельки для установки двух 8-разрядных кристаллов УСППЗУ типа К573РФ2/РФ4/РФ5/РФ6, ЭСППЗУ К558РР3 или любых других со стандартной разводкой выводов, например 27256. Выбор типа кристаллов зависит от конкретного приложения МК. Правила установки кристаллов в МК указаны на рис. 7.16.

При включении питания или нажатии клавиши RESET управление в МП ВМ86 передается по логическому адресу 0FFFFH:0000H, что соответствует физическому адресу 0FFFF0H. По этой причине последние 16 байт 1М-байтового адресного пространства памяти резервируется для инициализации МС. В первых пяти байтах этой области должна располагаться команда длинного перехода (FAR JMR) с 32-разрядным адресом sel:offset, загружаемым в пару CS:IP соответственно. Для организации этого перехода резидентную часть ПЗУ (2К—32К слов) лучше всего поместить в верхнюю область адресного пространства.

Постоянное запоминающее устройство организовано в виде единого 16-разрядного блока, считывание информации из которого всегда осуществляется словами без учета состояния линий $\overline{BHE}$ и ADR. Интерпретация данных возлагается на приемник информации — микропроцессор.

Система ВВ микроконтроллера содержит два последовательных канала связи типа ИРПС, реализованных на базе БИС программированного связного адаптера КР580ВВ51. Предполагается, что ПСА будут инициированы для работы в асинхронном старт-стоповом режиме. Один канал резервируется для подключения системной консоли, а второй может быть использован по усмотрению пользователя. Типичным вариантом его применения служит организация межмашинной связи.

Реализация средств ВВ включена в принципиальную схему рис. 7.17. Встроенные в последовательные каналы буферные схемы были обсуждены в § 3.4.

В качестве генератора скорости приема/передачи данных через ПСА используется один из трех счетчиков программируемого интервального таймера. Два других зарезервированы для общесистемных целей. Первый используется в качестве генератора меток системного времени, обычно следующих с частотой 20—100 Гц. Второй выполняет функцию счетчика пауз в единицах системного времени. Использование счетчиков ПИТ в других целях не предусмотрено.

В МК мМС1212 использована каскадно расширяемая система прерываний, резидентное ядро которой реализовано на трех микросхемах: К1810ВН59А, К589АП16 и К155ЛА3 (DD13), см. рис. 7.17. Высокоуровневые входы ведущего программируемого

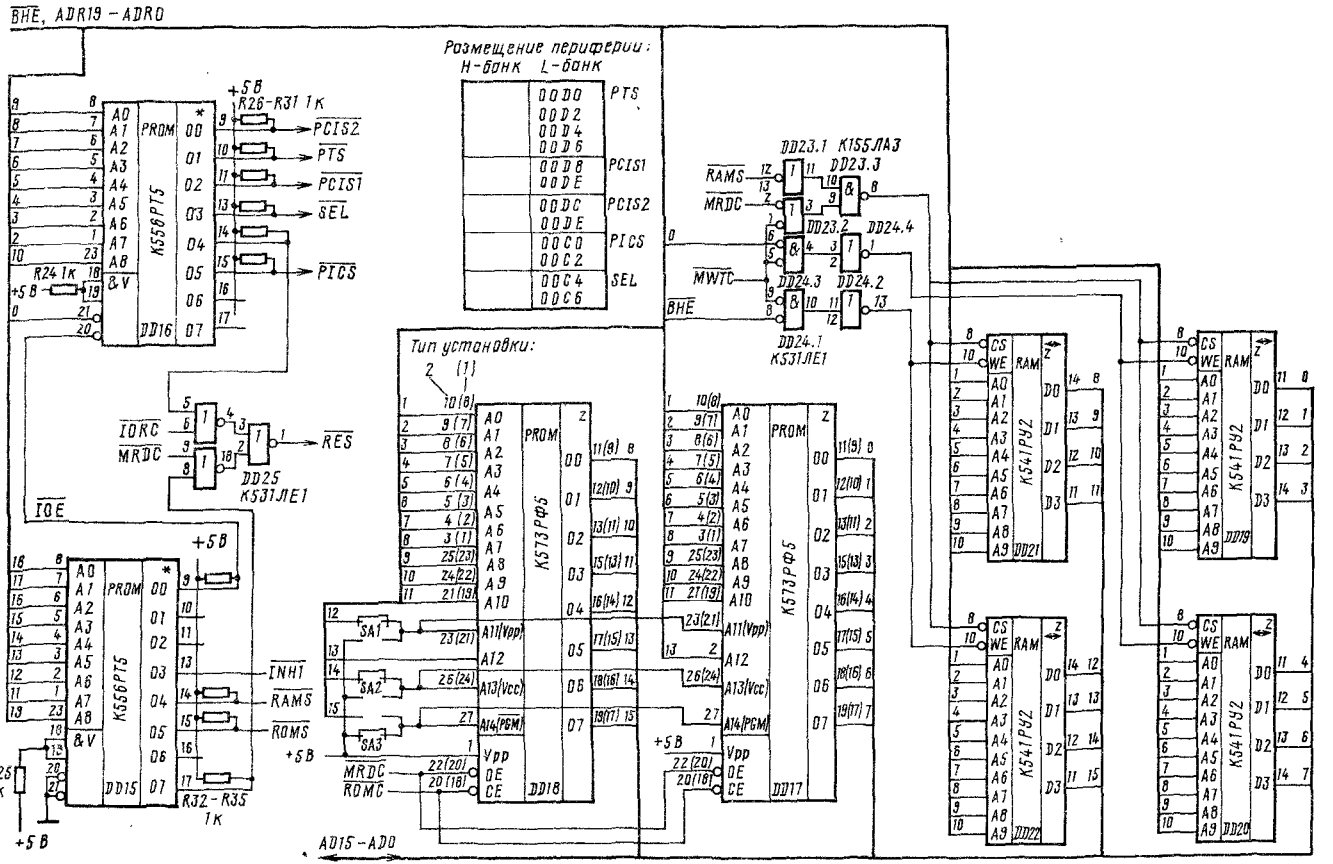


Рис. 7.16. Схема памяти

контроллера прерываний IR3—IR0 связаны с запросами по готовности приемников ПСА2 и ПСА1, с сигналом окончания счета паузы и метками системного времени. Оставшиеся свободными для пользователя линии IR7—IR4 контроллера прерываний выведены на системную магистраль с низким уровнем активности. В разряд пользовательских входит также линия немаскируемого прерывания NMI с фиксированным вектором прерывания, равным 2.

В схеме на рис. 7.17 предусмотрены средства расширения системы прерываний по методу каскадирования. Для этого БИС ВН59А должна быть запрограммирована для работы в каскадном режиме в качестве ведущей. Ведомые контроллеры подключаются к пользовательским уровням запросов на прерывания IR7—IR4. Информация CAS2—CAS0 о выборе подчиненного контроллера передается через линии ADR10—ADR8 системной шины адреса.

Генерируемый системным контроллером сигнал MCE активен в каждом из двух INTA-циклов, формируемых МП при приеме вектора прерывания. Этот перекрывающийся со стробом ALE сигнал позволяет пропустить адрес ведомого ВН59А CAS2—CAS0 на мультиплексную шину адреса/данных для запоминания в буферном адресном регистре. Однако выдача адреса линий CAS2—CAS0 на мультиплексную шину МП в первом цикле INTA не рекомендуется, так как гарантия о ее освобождении МП наступает лишь через 80 нс после начала цикла. Запрет сигнала MCE в первом цикле выполняет сигнал LOCK (см. рис. 7.8), который активизируется в такте Т2 первого цикла и снимается в такте Т2 второго цикла. Поскольку контроллер ВН59А выдает действительный адрес каскада только во втором цикле, то никакая информация не теряется.

При отсутствии БИС ВН59А последняя может быть заменена на ВН59 без изменения принципиальной и монтажной схем. В этом случае линия INTA не используется и должна быть отсоединена от ВН59. Ввод номера источника запросов выполняется по команде поллинга, а его декодирование — программными средствами, составляющими основу подпрограммы обслуживания прерывания с номером вектора прерывания 0FFH. Для этой и других целей мультиплексная шина адреса/данных должна быть подключена через дополнительные резисторы к источнику питания +5 В.

Методика подключения БИС памяти к 16-разрядной шине И41 и других ПУ была изложена в § 7.1. Для организации логики выборки БИС в МК мМС1212 использованы две микросхемы ППЗУ К556РТ5 (512×8 бит) DD15 и DD16 (рис. 7.16). При размещении в адресном пространстве блоков резидентного ОЗУ и ПЗУ реализуется БИС DD15. Эта же микросхема с помощью сигнала IOE выделяет область ВВ (0000Н—07FFH) в 1 К слов для последующего размещения в ней портов всех резидентных ПУ

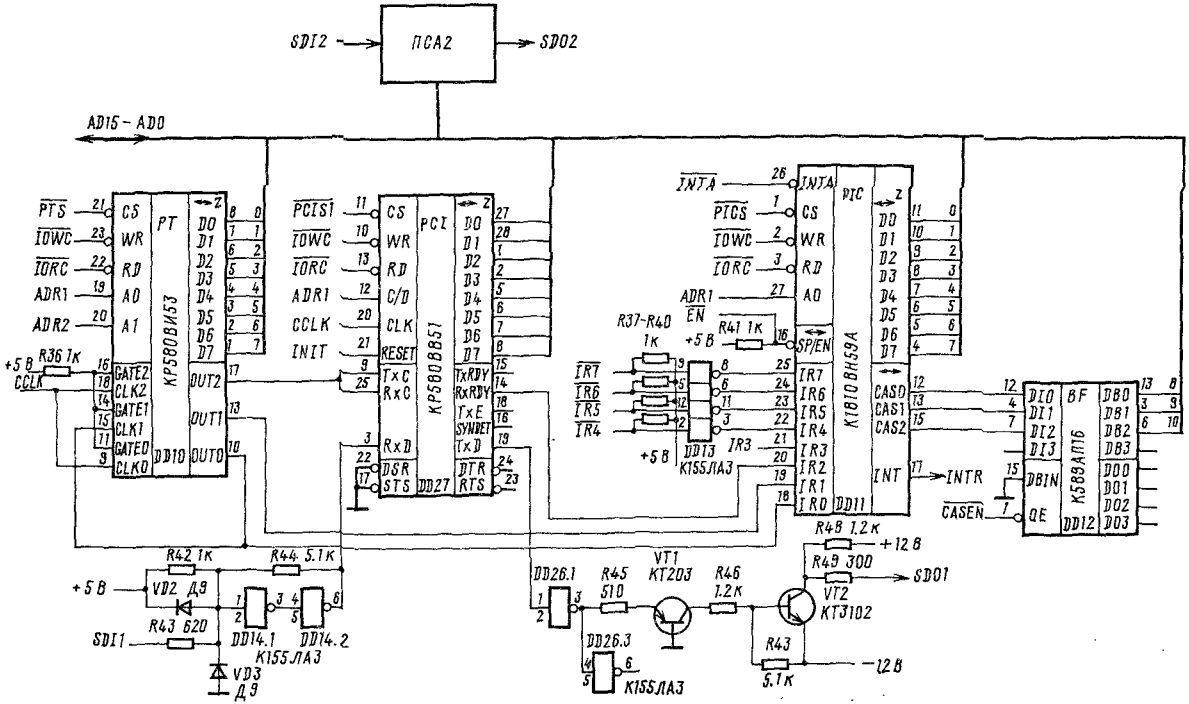


Рис. 7.17. Схема средств ВВ и поддержки реального времени

МК. Содержимое ППЗУ DD15 для случая, когда емкость ПЗУ МК равна 16К словам (21728×2) приведено в табл. 7.1. Генерируемый логикой выборки кристаллов сигнал $\overline{\text{INH1}}$:

$$\overline{\text{INH1}} = \text{ROMS} + \text{RAMS}$$

служит сигналом запрета работы основной памяти [14], которая может иметь физические ячейки в области адресов резидентного ОЗУ/ПЗУ. Линия $\overline{\text{INH1}}$ выполнена по схеме с открытым коллектором. Вторая ступень логики выборки кристаллов ПУ реализована на DD16, содержимое ППЗУ которой приведено в табл. 7.2. Все 8-разрядные массивы портов программируемых периферийных БИС размещены в L-банке пространства ВВ. По этой причине БИС DD16 открывается лишь при условии доступа к L-банку (ADR0=0) без учета состояния $\overline{\text{BHE}}$. Вспомогательный сигнал $\overline{\text{SEL}}$ используется в отладочных целях для синхронизации осциллографа.

Дополнительная внешняя логика формирует сигнал $\overline{\text{RES}}$, корректирующий направление передачи драйвера шины данных. При чтении данных с устройств, расположенных непосредственно на плате, шинный драйвер автоматически включается на передачу. При вводе вектора прерывания возникают особые сложности, которые связаны с тем, что ведущий контроллер находится на мультиплексной шине, а дополнительные ведомые (возможно, отсутствующие) — на системной шине данных. Правильное решение дает схема на рис. 7.17, в которой генерируемый главным контроллером ВН59А по двунаправленной линии SP/$\overline{\text{EN}}$ сигнал $\overline{\text{EN}}$ (выдача данных) используется для коррекции направления передачи через шинный драйвер. При этом вопрос о правильном вводе вектора прерывания как в каскадном, так и в одиночном режиме работы контроллера прерываний решается автоматически.

Таблица 7.1

Адрес ПЗУ	Состояние	Адрес ПЗУ	Состояние	Адрес ПЗУ	Состояние
000	66	15F	57	1BF	57
10E	57	16F	57	1CF	57
11F	57	17F	57	1DF	57
12F	57	18F	57	1EF	57
13F	57	19F	57	1FF	57
14F	57	1AF	57	Остальные	FF

Таблица 7.2

Адрес ПЗУ	Состояние
00C	CF
02C	ED
06C	EB
08C	E7
0AC	ED
0EC	EE
Остальные	FF

СПИСОК ЛИТЕРАТУРЫ

1. **Алексенко А. Г., Галицин А. А., Иванников А. Д.** Проектирование радиоэлектронной аппаратуры на микропроцессорах.— М.: Радио и связь, 1984.—272 с.
2. **Алексенко А. Г., Шагурин В. И.** Микросхемотехника.—М.: Радио и связь, 1982.—416 с.
3. **Балашов Е. П., Григорьев В. Л., Петров Г. А.** Микро- и мини-ЭВМ: Учеб. пособие для вузов.—Л.: Энергоатомиздат, 1984.—376 с.
4. **Балашов Е. П., Пузанков Д. В.** Микропроцессоры и микропроцессорные системы: Учеб. пособие для вузов/Под ред. В. Б. Смолова.—М.: Радио и связь, 1981.—328 с.
5. **Баррон И., Кэвил П., Мэй Д.** Транспьютер с быстродействием 5 млн. операций/с и более//Электроника.—1982.—№ 24.—С. 26—35.
6. **Березенко А. И.** Микропроцессорные комплекты общего применения.—М.: Машиностроение, 1982.—120 с.
7. **Бистон Дж.** Микроконтроллеры второго поколения, реализующие специализированные функции//Электроника.—1978.—№ 24.—С. 52—60.
8. **Брусенцов Н. П.** Микрокомпьютеры.—М.: Наука, 1985.—208 с.
9. **Вайда Ф., Чакань А.** МикроЭВМ: Пер. с венг.—М.: Энергия, 1980.—360 с.
10. **Григорьев В. Л.** Программирование однокристальных микропроцессоров.—М.: Энергоатомиздат, 1987.—288 с.
11. **Григорьев В. Л.** Программное обеспечение микропроцессорных систем.—М.: Энергоатомиздат, 1983.—208 с.
12. **Гилмор Ч.** Введение в микропроцессорную технику: Пер. с англ.—М.: Мир, 1984.—334 с.
13. **Дианов А. П., Щелкунов Н. Н.** Методика программирования микросхем ПЗУ// Микропроцессорные средства и системы.—1985.—№ 3.—С. 75—79.
14. **Дианов А. П., Щелкунов Н. Н.** Организация динамической памяти микросистем//Микропроцессорные средства и системы.—1987.—№ 4.—С. 75—80.
15. **Дианов А. П., Щелкунов Н. Н.** Система проектирования микропроцессорных устройств//Микропроцессорные средства и системы.—1987.—№ 5.—С. 83—86.
16. **Каган Б. М., Сташин В. В.** Микропроцессоры в цифровых системах.—М.: Энергия, 1979.—192 с.
17. **Каган Б. М., Сташин В. В.** Основы проектирования микропроцессорных устройств автоматики.—М.: Энергоатомиздат, 1987.—304 с.
18. **Каган Б. М.** Электронные вычислительные машины и системы: Учеб. пособие для вузов.—2-е изд.—М.: Энергия, 1985.—542 с.
19. **Клингман Э.** Проектирование микропроцессорных систем: Пер. с англ.—М.: Мир, 1980.—576 с.
20. **Клингман Э.** Проектирование специализированных микропроцессорных систем: Пер. с англ.—М.: Мир, 1985.—363 с.
21. **Кобылинский А. В., Линоветский Г. П.** Однокристальные микроЭВМ серии К1816//Микропроцессорные средства и системы.—1986.—№ 1.—С. 10—19.
22. **Кобылинский А. В., Москалевский А. Н., Темчеико В. А.** Однокристальный высокопроизводительный 16-разрядный микропроцессор КМ1810ВМ86//Микропроцессорные средства и системы.—1986.—№ 1.—С. 28—33.
23. **Коффрон Дж.** Технические средства микропроцессорных систем. Практический курс: Пер. с англ.—М.: Мир, 1983.—344 с.
24. **Левенталь Л.** Введение в микропроцессоры.—М.: Энергоатомиздат, 1983.—463 с.
25. **Лиьят А.** Архитектура малых вычислительных систем: Пер. с англ./Под ред. К. В. Поселева.—М.: Мир, 1981.—187 с.
26. **Мнк Дж., Брик Дж.** Проектирование микропроцессорных устройств с разрядно-модульной организацией. В 2-х кн.: Пер. с англ.—М.: Мир, 1984.—Кн. 1.—253 с.—Кн. 2.—223 с.

27. **Микрокомпьютер** 8086, обладающий возможностями 8- и 16-разрядных процессоров/Кац, Морс, Полман, Ривенел // Электроника.— 1978.— № 4.— С. 23—31.
28. **Микропроцессорные** БИС и микро-ЭВМ: Построение и применение/А. А. Васенков, Н. М. Воробьев, В. Л. Дшхунян и др.; Под ред. А. А. Васенкова.— М.: Сов. радио, 1980.— 280 с.
29. **Микропроцессоры.** В 3-х кн.: Учеб. пособие для вузов/Под ред. Л. Н. Преснухина.— М.: Высшая школа, 1986.— Кн. 1.— 495 с.; Кн. 2.— 383 с; Кн. 3.— 351 с.
30. **Микропроцессоры:** Системы программирования и отладки/В. А. Мясников, М. Б. Игнатьев, А. А. Кочкин, Ю. Е. Шейнин; Под ред. В. А. Мясникова, М. Б. Игнатьева.— М.: Энергоатомиздат, 1985.— 272 с.
31. **Монолитный** микрокомпьютер с аналого-цифровым преобразователем/В. Чек, Е. Чжен, Г. Хилл и др. // Электроника.— 1978.— № 11.— С. 36—44.
32. **Муи Дж.** Микрокомпьютер для эмуляции // Электроника.— 1980.— № 16.— С. 48—53.
33. **Мячев А. А., Иванов В. В.** Интерфейсы вычислительных систем на базе мини- и микро-ЭВМ/Под ред. Б. Н. Наумова.— М.: Радио и связь, 1986.— 248 с.
34. **Неверески У. Дж.** 8- и 16-разрядные процессоры — наиболее сложные КМОП-схемы с высокоуровневой архитектурой // Электроника.— 1984.— № 7.— С. 27—35.
35. **Погорелый С. Д., Слободянюк Т. Ф.** Программное обеспечение микропроцессорных систем.— Киев: Техника, 1985.— 240 с.
36. **Полупроводниковые** БИС запоминающих устройств: Справочник/В. В. Баранов, Н. В. Бекин, А. Ю. Гордонов и др.; Под ред. А. Ю. Гордонова и Ю. Н. Дьякова.— М.: Радио и связь, 1987.— 360 с.
37. **Проектирование** цифровых систем на комплектах микропрограммируемых БИС/С. С. Булгаков, В. М. Мещеряков, В. В. Новоселов, Л. А. Шумилов; Под ред. В. Г. Колесникова.— М.: Радио и связь, 1984.— 240 с.
38. **Система** команд микропроцессора КМ1810ВМ86/А. В. Кобылинский, А. В. Береза, Н. Г. Сабадаш и др. // Микропроцессорные средства и системы.— 1986.— № 2.— С. 3—9.
39. **Соботка З., Стары Я.** Микропроцессорные системы: Пер. с чеш.— М.: Энергоатомиздат, 1981.— 496 с.
40. **Соучек Б.** Микропроцессоры и микроЭВМ: Пер. с англ.— М.: Сов. радио, 1979.— 520 с.
41. **Уокерли Дж.** Архитектура и программирование микроЭВМ. В 2-х кн.: Пер. с англ.— М.: Мир, 1984.— Кн. 1.— 496 с.; Кн. 2.— 341 с.
42. **Форс.** Стандартная микропроцессорная шина, упрощающая задачи разработчиков микрокомпьютеров // Электроника.— 1978.— № 15.— С. 33—41.
43. **Хоуп Г.** Проектирование цифровых вычислительных устройств на интегральных схемах: Пер. с англ.— М.: Мир, 1984.— 400 с.
44. **Хилбурн Дж., Джулич П.** МикроЭВМ и микропроцессоры: Пер. с англ./Под ред. С. Д. Пашкеева.— М.: Мир, 1979.— 463 с.
45. **Шевкопляс Б. М.** Микропроцессорные структуры. Инженерные решения.— М.: Радио и связь, 1986.— 264 с.
46. **Шумейкер К.** Микропроцессор, содержащий ряд периферийных функциональных блоков и упрощающий построение микросистем // Электроника.— 1983.— № 9.— С. 51—59.
47. **Шумейкер К.** 16-разрядный СБИС-микропроцессор с высокими техническими характеристиками // Электроника.— 1983.— № 10.— С. 45—51.
48. **Щелкунов Н. Н., Дианов А. П.** Одноплатный 16-разрядный микроконтроллер общего назначения // Микропроцессорные средства и системы.— 1987.— № 1.— С. 77—83.
49. **Щелкунов Н. Н., Дианов А. П.** Программирование микросистем реального времени // Микропроцессорные средства и системы.— 1985.— № 4.— С. 31—35.
50. **Щелкунов Н. Н., Дианов А. П.** Процедуры программирования логических матриц // Микропроцессорные средства и системы.— 1986.— № 2.— С. 71—76.

51. **Щелкунов Н. Н., Дианов А. П.** Техника программирования 16-разрядных микроконтроллеров // Микропроцессорные средства и системы.— 1987.— № 2.— С. 11—14.
52. **Щелкунов Н. Н., Дианов А. П.** Техника программирования 8-разрядных микроконтроллеров // Микропроцессорные средства и системы.— 1986.— № 6.— С. 23—28.
53. **Щелкунов Н. Н., Дианов А. П.** Универсальный одноплатный микроконтроллер // Микропроцессорные средства и системы.— 1986.— № 5.— С. 65—69.
54. **APX-186** High Integration 16-bit Microprocessor.—Intel, 1982.
55. **APX-286/10** High Performance Microprocessor with Memory Management and Protection.—Intel, 1983.
56. **MCS-80/85** Microprocessors.—Intel, 1977.
57. **Microcontroller** Handbook.—Intel, 1985.
58. **The 8086** Family Users Manual.—Intel, 1979.
59. **UPI-41** Users Manual.—Intel, 1977.
60. **80386** High Performance Microprocessor with Integrated Memory Management.—Intel, 1985.
61. **Микропроцессоры** и микропроцессорные комплекты интегральных микросхем: Справочник. В 2-х т./Н. Н. Аверьянов, А. И. Березенко, Ю. И. Борщенко и др.; Под ред. В. А. Шахнова.— М.: Радио и связь, 1988.— Т. 1.— 368 с.— Т. 2.— 368 с.
62. **Лю Ю-Чжен, Гибсон Г.** Микропроцессоры семейства 8086/8088. Архитектура, программирование и проектирование микрокомпьютерных систем: Пер. с англ.— М.: Радио и связь, 1987.— 512 с.
63. **Ушкар М. Н.** Микропроцессорные устройства в радиоэлектронной аппаратуре.— М.: Радио и связь, 1988.— 128 с.
64. **Перспективные** однокристальные ЭВМ/М. Г. Весноватов, Г. В. Карацюба, В. В. Павлов, В. А. Старшова // Микропроцессорные средства и системы.— 1987.— № 2.— С. 7, 8.
65. **Басманов А. С., Широков Ю. Ф.** Микропроцессоры и однокристальные микроЭВМ: Номенклатура и функциональные возможности/Под ред. В. Г. Дорачева — М.: Энергоатомиздат, 1988.— 128 с.

ОГЛАВЛЕНИЕ

Предисловие 3
Введение 4

Глава 1. **Организация микросистем** 8

1.1. Понятия организации и архитектуры 8
1.2. Архитектура типовой микросистемы 9
1.3. Структура типовой микросистемы 14
1.4. Регистры микропроцессора 21
1.5. Адресация данных 31

Глава 2. **Микропроцессоры КР580ВМ80/К1821ВМ85А** 42

2.1. Вводные замечания 42
2.2. Архитектура ВМ80 42
2.3. Система команд ВМ80 44
2.4. Структурная схема ВМ80 56
2.5. Базовый комплект БИС серии КР580 64
2.6. Центральный процессор на базе БИС серии КР580 68
2.7. Организация ВМ85А 71

Глава 3. **Подсистема ввода-вывода** 82

3.1. Организация программно-управляемого обмена 82
3.2. Периферийные БИС 90
3.3. Средства параллельного ввода-вывода 95
3.4. Средства последовательного ввода-вывода 108
3.5. Система прерываний 119
3.6. Программируемый контроллер прерываний ВН59 127
3.7. Средства счета времени 133

Глава 4. **Организация однокристальных микроконтроллеров** 143

4.1. Вводные замечания 143
4.2. Базовая организация ВЕ48 146
4.3. Набор регистров ВЕ48 148
4.4. Организация памяти ВЕ48 150
4.5. Система ввода-вывода и служба реального времени ВЕ48 153
4.6. Система команд ВЕ48 156
4.7. Физический интерфейс ВЕ48 163
4.8. Расширение внутренних ресурсов ВЕ48 167
4.9. Универсальный периферийный адаптер 170
4.10. Базовая организация ВЕ51 173
4.11. Периферийные средства ВЕ51 179
4.12. Система команд ВЕ51 187
4.13. Функциональное описание ВЕ51 195

Глава 5. **Организация одноплатных микроконтроллеров на базе КР580ВМ80** 203

5.1. Одноплатный микроконтроллер мМС1204 203
5.2. Средства ввода-вывода и поддержки режима реального времени 208
5.3. Программирование системы ввода-вывода 211
5.4. Программирование средств поддержки режима реального времени 216

Глава 6. **Микропроцессор К1810ВМ86** 219

6.1. Вводные замечания 219

6.2. Организация регистров ВМ86 220
6.3. Организация памяти ВМ86 223
6.4. Формат команд ВМ86 226
6.5. Система команд ВМ86 232
6.6. Структурная схема ВМ86 248
6.7. Структурные отличия ВМ88 258

Глава 7. **Организация одноплатных микрокоитроллеров на базе К1810ВМ86** 262

7.1. Принципы построения 262
7.2. Генератор тактовых импульсов ГФ84 264
7.3. Системный контроллер ВГ88 267
7.4. Центральный процессор на базе БИС серии К1810 268
7.5. Программируемый контроллер прерываний ВН59А 274
7.6. Одноплатный микроконтроллер мМС1212 277
Список литературы 284

Производственное издание

ЩЕЛКУНОВ НИКОЛАЙ НИКОЛАЕВИЧ и ДИАНОВ АЛЕКСАНДР ПЕТРОВИЧ

МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

Заведующий редакцией Ю. Н. Рысев
Редактор И. П. Леонтьева
Художественный редактор Н. С. Шеин
Переплет художника Н. А. Пашуро
Технический редактор Г. З. Кузнецова
Корректор Л. А. Буданцева

ИБ № 1583

Слано в набор 30.01.89. Подписано в печать 30.06.89. Т.-23740. Формат $60\times88^{1}/_{16}$. Бумага офсет. № 2. Гарнитура таймс. Печать офсет. Усл. печ. л. 17,64. Усл. кр.-отт. 17,64. Уч.-изд. л. 20,78. Тираж 25 000 экз. Изд. № 21978. Зак. № 948. Цена 1 р. 40 к.

Издательство «Радио и связь». 101000 Москва, Почтамт, а/я 693

Ордена Октябрьской Революции и ордена Трудового Красного Знамени МПО «Первая Образцовая типография» Государственного комитета СССР по печати. 113054, Москва, Валовая, 28.