Федеральное агентство по образованию
ГОУ ВПО «Уральский государственный технический университет – УПИ»

В.П. Мокрецов

МИКРОПРОЦЕССОРЫ И МПС

Часть 1

Архитектура микропроцессора К580ВМ80. Организация МП-систем

Учебное электронное текстовое издание
Научный редактор: доц., канд. техн. наук В.И. Паутов

Пособие предназначено для студентов специальностей 220101– Управление и информатика в технических системах, 230101 – Вычислительные машины, комплексы, системы и сети.

Описаны архитектура простейшего микропроцессора и семейство интегральных программируемых микросхем; организация различных подсистем микропроцессорных устройств и МП-систем. Рассмотрено программирование различных операций в системе команд процессора, приведены упражнения, контрольные вопросы и задания.

Екатеринбург
2007

ПРЕДИСЛОВИЕ

Современные микропроцессоры представляют собой весьма сложные по устройству изделия микроэлектроники. Многочисленные типы микропроцессоров характеризуются различными архитектурными решениями и функциональными возможностями. Микропроцессорная техника стремительно и многонаправленно развивается и совершенствуется, интегрируя новейшие достижения микроэлектроники и схемотехники.

Одним из важнейших направлений развития микропроцессорной техники является создание универсальных микропроцессоров, предназначенных для применения в вычислительных системах: персональных компьютерах, рабочих станциях, серверах, параллельных суперЭВМ.

Создание микроконтроллеров и цифровых сигнальных процессоров является также важнейшим направлением развития микропроцессорной техники. Однокристальные микроконтроллеры выпускаются различных типов, отличаются назначением, сложностью, архитектурными решениями и т.д.

Изучение микропроцессорных средств, являясь достаточно сложной задачей, требует оптимального выбора тематики и временных затрат на их освоение. Значительный интерес для изучения представляют первые семейства 8-разрядных микропроцессорных БИС: 8080, 8085 (Intel), Z80 (Zilog), MC6800, MC6809 (Motorola), MCS6500 (MOS Technology). Эти микросхемы стали доступными и классическими примерами построения на их основе микропроцессорных систем. Особенно подробно описаны в учебно-методической и инженерно-технической литературе микропроцессорные БИС семейства 8080 и 8085 и их отечественные аналоги серии К580 и К1821. Поэтому начальные знания о микропроцессорах и микропроцессорных системах целесообразно получить, используя в качестве примеров семейства БИС серии К580, 8080. Это позволит эффективно освоить простейшие средства и далее двигаться к более сложным микропроцессорам и МП-системам.

Число различных модификаций МК, представленных на мировом рынке, исключительно велико. Начало мощному клану МК с ядром MCS-51 положила фирма Intel, выпустив в 1980 г. МК 8051АН. Его аналог получил в России распространение под именем 1816ВЕ51, который хорошо описан в различных источниках. Для МК этого клана разработаны различные средства проектирования и отладки программного обеспечения. Поэтому для начального этапа изучения микроконтроллерных средств выбрано это семейство микроконтроллеров. На последующих этапах рассматриваются МК современных популярных семейств.

1. МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА

1.1. Структура микропроцессорного устройства

Наиболее широко применяется магистрально-модульный принцип построения микропроцессорных систем (МПС). Упрощенная структура такой МПС приведена ниже. Центральным устройством в системе является микропроцессор (МП), выполняющий арифметические и логические операции над данными, осуществляющий управление выборкой команд и данных из памяти и организующий взаимодействие всех устройств, входящих в систему.

Работа МП происходит под воздействием тактовых сигналов, вырабатываемых схемой синхронизации, часто выполняемой в виде отдельной микросхемы (генератора тактовых импульсов).

Программы работы МПС размещаются в модулях постоянных запоминающих устройств (ПЗУ) и оперативных запоминающих устройств (ОЗУ). Микропроцессорные устройства и МПС содержат различные средства ввода-вывода информации. Шины адреса, данных и управления объединяют все устройства в единую систему. Периферийные устройства подсоединяются к шинам через программируемые периферийные адаптеры, осуществляющие передачу информации в параллельном или последовательном кодах. Наличие программно-настраиваемых адаптеров делает весьма гибкой и функционально богатой систему ввода-вывода информации в МПС.

Для разработки микропроцессорных устройств и МПС промышленность выпускает семейства микропроцессорных БИС, позволяющих удобную реализацию всех необходимых подсистем. В настоящее время выпускаются разнообразные однокристальные микроконтроллеры разных классов, на базе которых можно создавать микропроцессорные устройства, микроЭВМ, МПС (см. рис.1.1).

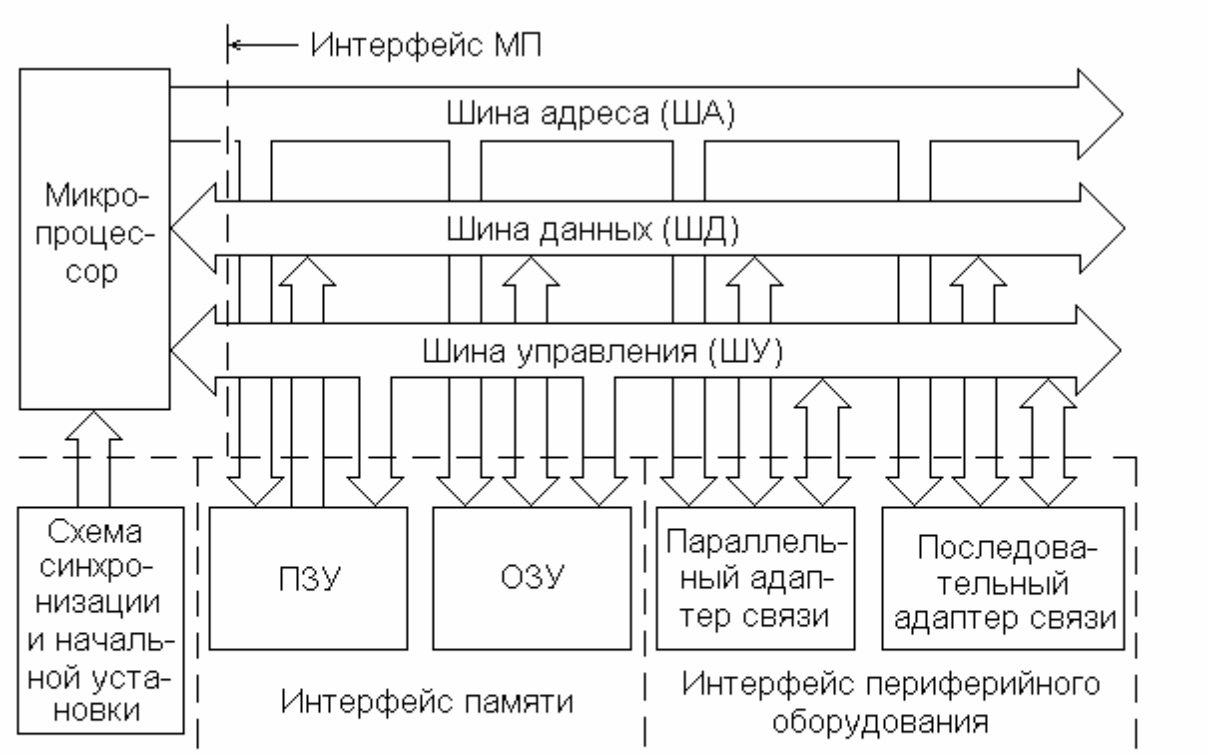


Рис.1.1. Типичная структура микропроцессорного устройства, МПС или Микро-ЭВМ

Подготовка списка команд называется программированием, а подготовленный список команд представляет собой программу, которая заносится в ПЗУ или ОЗУ и предписывает микропроцессору определенные действия. МП считывает из памяти команду и выполняет ее. Далее МП формирует адрес следующей команды в программе, считывает, выполняет и формирует адрес новой команды. Так функционирует ЭВМ.

1.2. Основные понятия и термины

Микропроцессорная техника (МПТ) включает технические и программные средства, используемые для построения различных микропроцессорных систем, устройств и персональных микроЭВМ.

МикроЭВМ – это цифровая ЭВМ, выполненная на микропроцессорных БИС и имеющая компактные размеры. Простейшая микроЭВМ состоит из пяти частей: устройства *ввода,* устройства *управления*, *арифметико-логического* устройства, *памяти* и устройства *вывода*.

Микропроцессор (МП) **–** это микросхема, выполняющая арифметические и логические операции над данными и осуществляющая программное управление вычислительным процессом.

Микропроцессорный комплект (МПК), или серия микропроцессорных БИС представляет собой набор взаимно совместимых микросхем, позволяющих разработку микроЭВМ, микропроцессорных систем (МПС).

Микропроцессорной системой обычно называют функционально законченное изделие, состоящее из одного или нескольких микропроцессорных устройств, выполненных на микропроцессорах или микроконтроллерах.

Микропроцессорное устройство (МПУ) представляет собой функционально и конструктивно законченное изделие, состоящее из нескольких микросхем, в состав которых входит микропроцессор; оно предназначено для выполнения определенного набора функций: получения, обработки, передачи, преобразования информации и управления.

Микроконтроллер (МК) представляет собой однокристальную микроЭВМ с небольшими вычислительными ресурсами и упрощенной системой команд, ориентированной на выполнение процедур логического управления различным оборудованием, или, в общем случае, системой команд, ориентированной на удобное выполнение определенного типа задач. Расширение сферы применения МК привлекло к развитию их архитектуры за счет размещения на

кристалле модулей, отражающих своими функциональными возможностями специфику решаемых задач. Такие модули называют периферийными. По составу периферийных устройств различают МК и **интегрированный процессор** (ИП). ИП определяет новый класс функционально емких однокристальных устройств, по количеству и составу периферийных устройств уступающий МК и занимающий промежуточное положение между МП и МК.

Логическая организация микропроцессоров и МПС направлена на достижение универсальности применения, высокой производительности вычислений. Логическую организацию МП и МПС называют их **архитектурой**.

Встроенной системой управления называют систему управления, конструктивно интегрированную в оборудование: станок, робот, стиральную машину, принтер, автомобиль и т.д.

Индустриальные компьютеры имеют развитый набор стандартных устройств сопряжения с объектом (УСО) и являются универсальными средствами управления для широкого круга объектов.

Промышленные контроллеры в отличие от однокристальных микроконтроллеров содержат средства адаптации языка программирования к конкретной области применения и средства объединения их в системы.

Универсальные МП предназначаются для применения в вычислительных системах, персональных ЭВМ, рабочих станциях и суперЭВМ.

Цифровые сигнальные процессоры (ЦСП) ориентированы на обработку в реальном времени цифровых потоков, образованных путем оцифрования аналоговых сигналов.

Производители МП и МК выпускают их в виде семейств, имеющих общее процессорное ядро, взаимодействующее с периферийными устройствами различной номенклатуры.

Анализ различных семейств МП, МК показывает, что наиболее жизнеспособными являются семейства, полученные путем селекции в их длительном эволюционном развитии.

Вопросы и задания

1.1. Назовите основные устройства ЭВМ и поясните их функциональное назначение.

1.2. Изобразите по памяти структурную схему ЭВМ.

1.3. Дайте определение понятий «микропроцессор», «микроконтроллер», «микропроцессорная система», «семейство микропроцессорных БИС».

1.4. Каково назначение и различие персональных и индустриальных компьютеров?

1.5. Каково назначение однокристальных и промышленных контроллеров?

1.6. Поясните понятие «встроенная микропроцессорная система управления».

1.7. Назовите основных производителей МП и МК.

1.8. Приведите примеры использования МП и МК.

2. МИКРОПРОЦЕССОР К580ВМ80

2.1. Структура микропроцессора

Микропроцессор К580ВМ80 (далее будем называть ВМ80) является аналогом микропроцессора 8080, выпущенного фирмой Intel в 1974 г.

Микропроцессор ВМ80 представляет собой функционально законченный однокристальный универсальный 8-разрядный микропроцессор с фиксированной системой команд.

В структурной схеме ВМ80 (рис.2.1) можно выделить следующие ее части: арифметико-логическое устройство (АЛУ), блок регистров, буферные схемы, управляющее устройство.

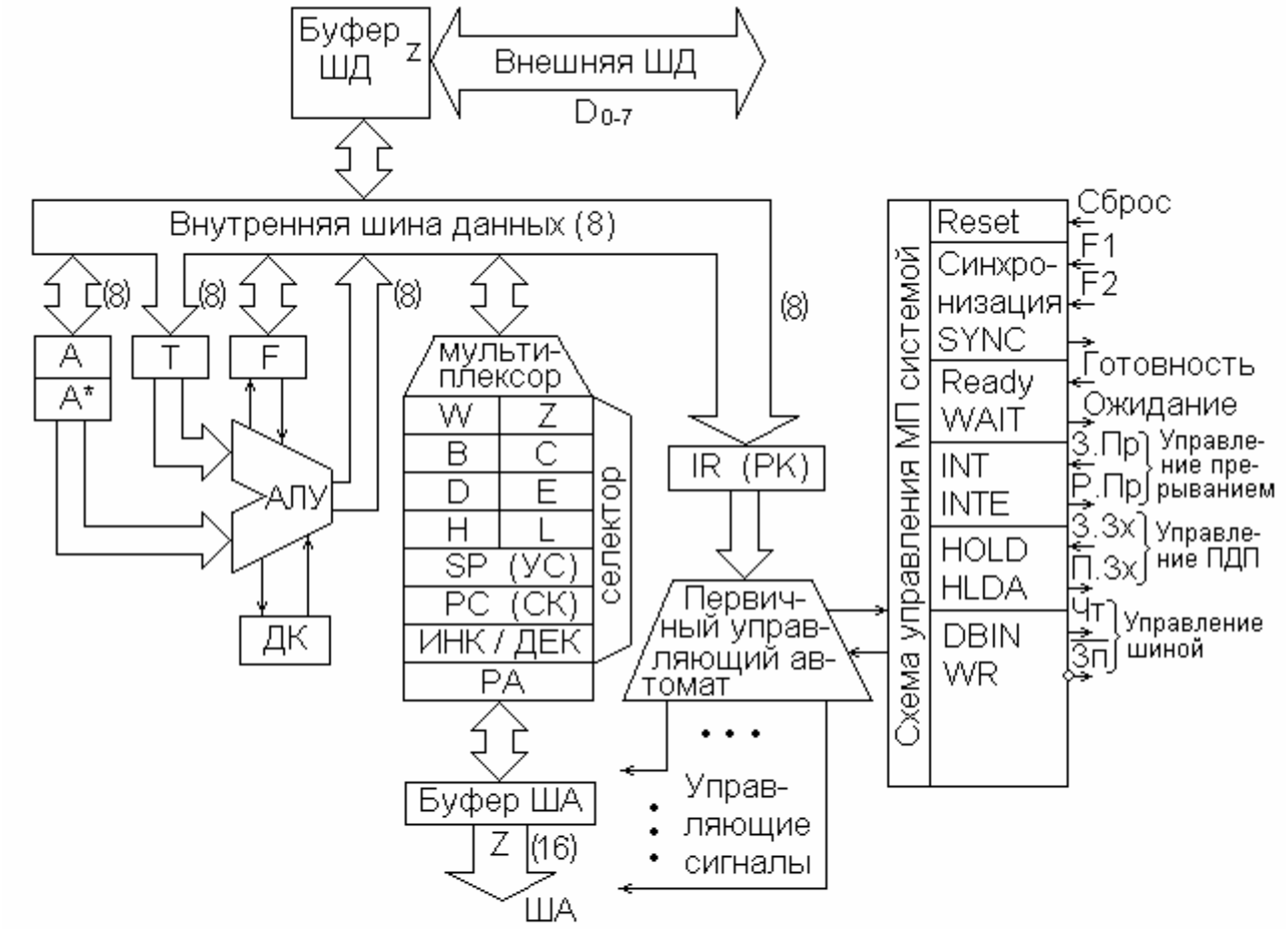


Рис.2.1. Структура 8-разрядного микропроцессора

Блок регистров ВМ80

Блок регистров ВМ80 содержит 8-разрядные программно-доступные регистры: **A** (регистр-аккумулятор); **B, C, D, E, H, L** (регистры общего назначения); **F** (регистр признаков-флагов); 16-разрядные специализированные регистры; программный счетчик СК **(PC**); регистр-указатель стека УС (**SP**); сдвоенный регистр косвенного адреса **HL** (H-регистр старшего полуадреса, L-регистр младшего полуадреса). А также имеются непосредственно недоступные программе регистры: 8-разрядный регистр команд (РК); 8-разрядные регистры временного хранения **T, W, Z**; 16-разрядный регистр адреса **PA**. Содержимое пар регистров B и C, D и E, H и L можно использовать в качестве слов двойной длины, т.е. 16-разрядных слов.

Регистры общего назначения (РОН) используются для хранения операндов, промежуточных и конечных результатов, а также адресов при косвенной адресации данных.

Регистр-аккумулятор А является особым в команде, в явном виде не адресуется, что позволяет упростить формат команд. Регистр А используется в качестве источника одного из операндов и приемника результата выполнения операции. Поэтому аккумулятор строится на основе двухступенчатых триггеров, что позволяет ему одновременно быть регистром-источником операнда и регистром-приемником результата выполнения операции. На использование аккумулятора в операции указывает код операции команды, т. е. в отношении аккумулятора применяется подразумеваемая адресация, что позволяет использовать одноадресные команды, имеющие сравнительно короткий формат.

К остальным программно-доступным регистрам применяется подразумеваемая или укороченная (регистровая) адресация, задаваемая коротким номером регистра в команде. Механизмы адресации будут рассмотрены ниже.

Использование регистра А и РОН позволяет при выполнении команд уменьшить количество обращений к памяти и этим самым повысить быстродействие микропроцессора.

В блоке регистров МП имеется схема инкремента/декремента, которая без привлечения АЛУ выполняет операцию прибавления/вычитания 1 над содержимым регистров. Схема инкремента/декремента позволяет реализовать процедуры автоматического задания приращений при операциях с адресами как в указателе стека (регистре SP), так и в программном счетчике (регистре PC).

При выполнении операций в МП возникает потребность в кратковременном хранении некоторых операндов и результатов выполнения операций. Для этого служат регистры временного хранения данных T, W, Z.

Для повышения эффективности операций с двухбайтными операндами и операций-пересылок двухбайтных адресов имеется возможность оперировать с содержимым регистровых пар B и C, D и E, H и L.

В состав блока регистров входит регистр-защелка адреса памяти **РА**. Любая пара регистров (BC, DE, HL) может быть использована для задания адресов команд и данных в программе. Такой адрес под воздействием соответствующих команд может быть загружен в регистр-защелку РА и модифицирован (посредством схемы инкремента/декремента) в процессе загрузки. Регистр-защелка РА передает адрес в буфер ША и далее в шину адреса.

Регистр команд **РК** служит для запоминания первого байта команды на все время ее выполнения. Код команды из этого регистра используется устройством управления для выработки последовательностей сигналов как для управления внутренними узлами МП, так и для управления внешними устройствами (памятью, портами и др.).

Арифметико-логическое устройство (АЛУ)

АЛУ выполняет арифметические и логические операции над 8-разрядными двоичными числами и представляет собой комбинационно-логическую схему. К одному из входов схемы АЛУ всегда подключен аккумулятор, к другому через регистр Т может быть подключен любой из регистров общего назначения.

АЛУ непосредственно связано с регистром признаков F, в соответствующих разрядах которого фиксируются особенности выполнения каждой операции: нулевой результат в аккумуляторе – **Z**, перенос из старшего/ заем в старший разряд – **CY**, знак результата – **S**, паритет – **P** и вспомогательный перенос из младшего полубайта – **AC**. Наличие в МП регистра признаков упрощает осуществление программных переходов в зависимости от состояния одного или более триггеров признаков. АЛУ позволяет в процессе регистровых «пересылок с перекосом» выполнять операции сдвига на один разряд вправо или влево.

В состав АЛУ входит комбинационная схема десятичной коррекции **ДК**, которая под воздействием специальной команды **DAA** преобразует результат сложения из двоичной формы в двоично-десятичную.

АЛУ выполняет простейшие арифметические операции сложения, вычитания; логические: конъюнкции, дизъюнкции, сложения по модулю 2; операции

сравнения, сдвига. Выполнение более сложных операций (умножение, деление, вычисление элементарных функций) может быть реализовано по разработанным подпрограммам.

Стековая память

Микропроцессор ВМ80 содержит средства для организации стековой памяти, позволяющей безадресное задание операндов. В общем случае стек представляет собой последовательность регистров или ячеек памяти, снабженных указателем стека, в котором автоматически при записи и считывании устанавливается адрес последней занятой ячейки стека (вершины стека). В стеке реализуется принцип обслуживания «последний пришел – первый вышел». Этот принцип при обращении к стеку реализуется автоматически. Поэтому команды записи и считывания не содержат адрес ячейки стека. Микропроцессор ВМ80 имеет только регистр-указатель стека (SP) и соответствующие цепи управления. Сам стек реализуется в оперативной памяти путем записи в указатель стека SP адреса ячейки памяти, являющейся исходной вершиной стека.

Буферные схемы

Двунаправленный буфер шины данных осуществляет логическое и электрическое разделение внутри процессорной шины данных и внешней системной ШД. Буфер состоит из регистра-защелки и выходной схемы с тремя состояниями, т. е. выходной схемы, обеспечивающей на выходе состояния логических 0 и 1 и третье состояние, реализующее полное электрическое отключение от нагрузки (высоко импедансное состояние). Наличие такого буфера дает возможность отключать устройства, подключенные к общей системной шине в МП- системе.

При выполнении операций вывода данных буферная схема передает в системную шину данных содержимое буферного регистра-защелки, на вход которого по внутренней шине с одного из регистров ВМ80 загружен код, подлежащий выдаче.

При выполнении ввода данных в микропроцессор внутренняя шина данных подключается к выходным цепям регистра-защелки буфера, загрузку которого из внешней ШД производит буферная схема под управлением команды.

Буферная схема переходит в высоко импедансное **Z**-состояние при выполнении МП-операций, не связанных с вводом или выводом.

Буферная схема шины адреса БА – однонаправленная, обеспечивает передачу адресов команд, данных и адресов устройств ввода-вывода. Выход буфера адреса также может переходить в Z-состояние. Это позволяет использовать ША другим устройствам, например контроллеру прямого доступа к памяти и др.

Устройство управления

Устройство управления (или схема управления) вырабатывает управляющие сигналы, необходимые для выполнения команды. В основе этой схемы действует цифровой автомат. В схеме управления можно выделить следующие средства:

- ***Регистр команд РК,*** служащий для приема кода команды (первого байта) и хранения этого кода на время ее выполнения;
- ***Дешифратор кода команды*** и выработки сигналов управления микрооперациями в соответствии с «зашитой» микропрограммой выполнения команды;
- ***Схему формирования*** машинных тактов, машинных циклов;
- ***Схему выборки РОН,*** участвующих в операции, представляющую собой дешифратор разрядов кода команды, указывающий номер регистра источника и регистра приемника операндов;
- ***Схему анализа переходов,*** которая анализирует состояние триггеров регистра признаков F при выполнении команд условных переходов;
- ***Схему формирования*** и выдачи кода типа текущего машинного цикла, на основе которого можно формировать расширенный набор сигналов для управления памятью, устройствами ввода-вывода, прерываниями, т. е. МП-системой;
- ***Схему анализа прерываний,*** содержащую триггер разрешения прерываний и триггер запроса прерываний. Анализ наличия запросов МП ВМ80

осуществляет в последнем такте последнего машинного цикла текущей команды;

- ***Схему анализа готовности,*** которая проверяет сигнал высокого уровня на линии «Ready» в конце второго такта машинного цикла;
- ***Схему анализа запроса шин*** (входная линия HOLD), обеспечивающую возможность реализации режима прямого доступа к памяти.

Вопросы и задания

2.1. Изобразите по памяти структуру МП ВМ80 и поясните назначение образующих ее узлов.

2.2. Поясните функции, выполняемые АЛУ.

2.3. Какие признаки формируются и при выполнении каких операций?

2.4. Каково назначение регистра IR инструкций (команд)?

2.5. Почему регистр-аккумулятор имеет структуру с двумя элементами памяти? Поясните на примере выполняемых команд.

2.6. Поясните назначение регистра T временного хранения операнда на примерах выполнения команд.

2.7. Поясните состояние узлов МП после воздействия сигнала Reset.

2.8. Поясните последовательность действий МП при начальном пуске (включении питания или действия сигнала Reset).

2.9. Каковы назначение блока десятичной коррекции и алгоритм его работы?

2.10. Каково назначение регистров A, B, C, D, E, H, L и ячейки M? Поясните на примерах команд, выполняемых микропроцессором.

2.2. Программная модель микропроцессора и МПС

Модель микропроцессора содержит только узлы, наиболее важные для понимания процесса его работы. Модель может содержать программно-доступные и программно-недоступные узлы. На рис.2.2 представлены программно-доступные (адресуемые в командах в явной или неявной форме) узлы МП, памяти и устройств ввода-вывода. На рис.2.3 указаны программно-

недоступные узлы МП, наиболее существенные для процесса выполнения команд. В программной модели выделены регистры **B, C, D, E, H, L, A**. Эти регистры предназначены для хранения данных. Они находятся в полном распоряжении программиста и единообразно участвуют в арифметических и логических операциях. К этой же группе отнесена так называемая ячейка памяти **M**, адрес которой помещен в регистровую пару **HL**. Ячейка памяти M также предназначена для хранения данных и находится в полном распоряжении программиста после того, как ее адрес загружен в регистровую пару HL. При этом она так же, как и РОН, единообразно участвует в арифметических и логических операциях. Все РОН, в том числе и ячейка M, могут использоваться для создания программно-управляемых счетчиков.

Регистр **A** выполняет функции аккумулятора. В двух операндных операциях он является источником одного операнда и приемником результата выполнения команды.

Регистр признаков **F** содержит признаки выполнения операций. Формат регистра F приведен на рис. 2.4. В регистре F фиксируются признаки: **CY –** признак переноса/заема. Если при выполнении команды возник перенос из старшего разряда или заем в старший разряд, то CY=1, иначе CY=0.

P – признак паритета. Если число единиц в байте результата четно, то P=1, иначе P=0.

AC – признак вспомогательного переноса. Если есть перенос между тетрадами байта, то AC=1, иначе AC=0.

Z – признак нуля. Если результат равен нулю, то Z=1, иначе Z=0.

S – признак «знака» принимает значение старшего разряда результата.

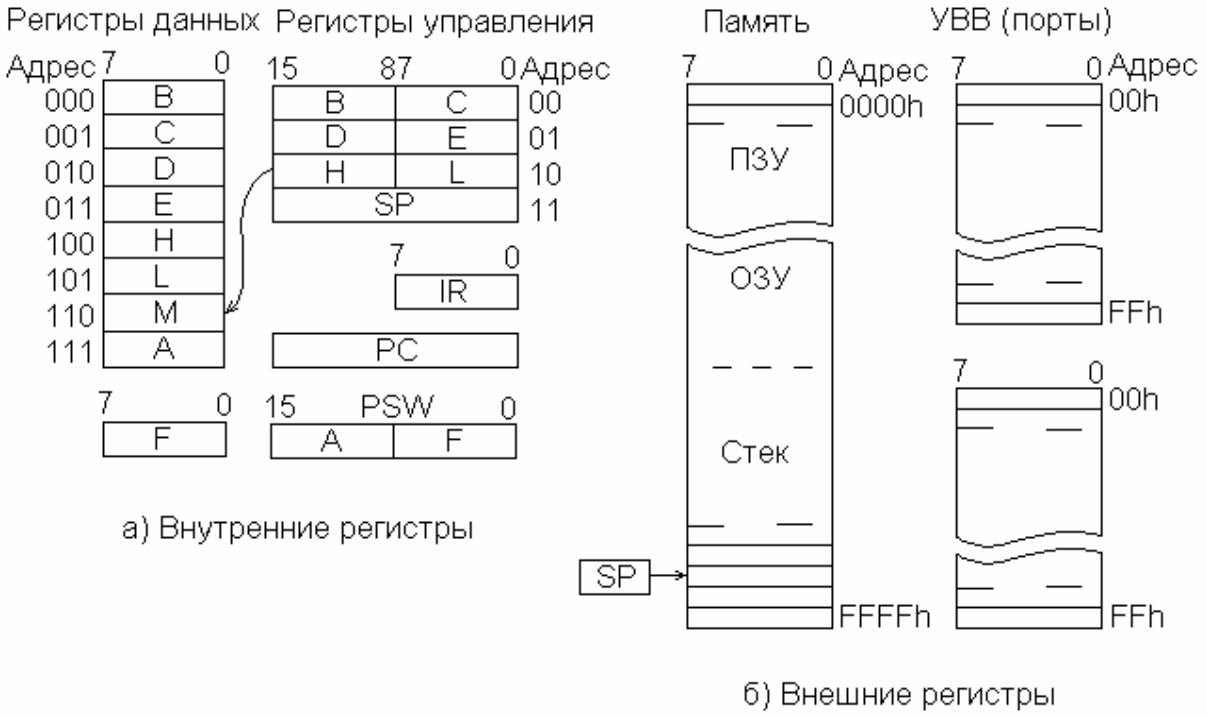


Рис.2.2. Программная модель МП (а) и МП системы (а и б)

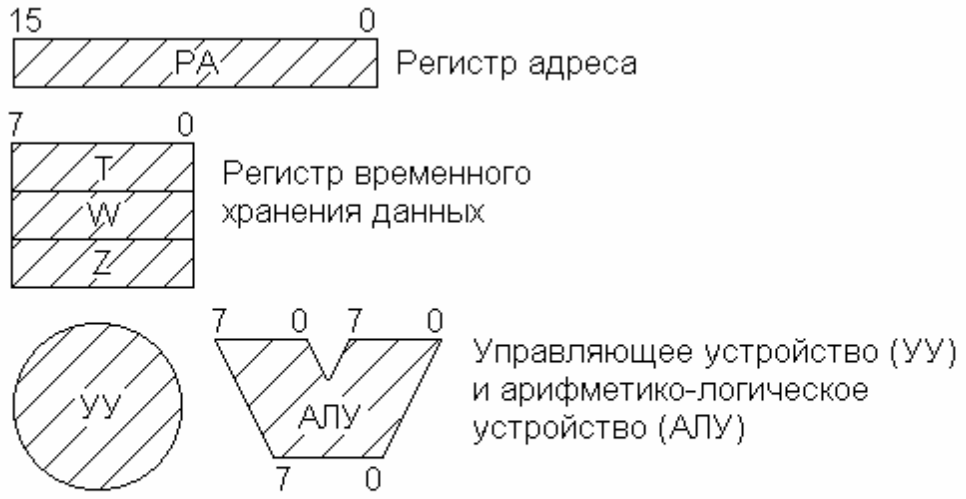


Рис.2.3. Программно-недоступные узлы МП

Регистр F

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	0	AC	0	P	1	CY

Рис.2.4. Регистр признаков F

Группы регистров управления **BC, DE, HL** являются указательными. Они предназначены для хранения 16-битных адресов, обеспечивают при этом косвенную адресацию и динамическое вычисление адреса ячейки памяти.

Указательный регистр **SP** содержит адрес ячейки оперативной памяти, являющейся текущей вершиной стека. Содержимое SP автоматически и последовательно уменьшается на 2 при выполнении команд записи в стек. А при выполнении команд считывания из стека содержимое регистра SP автоматически увеличивается на 2. Одна стековая команда записи осуществляет запись содержимого указанной в команде регистровой пары, т. е. два байта. При считывании из стека содержимое двух ячеек вершины стека заносится в регистровую пару, указанную в команде считывания.

Регистр инструкций IR доступен неявно. В этот регистр помещается первый байт команды, выбираемой из памяти при выполнении микропроцессором программы.

Указательный регистр PC выполняет функции основного адресанта, называется программным счетчиком или счетчиком команд (СК). Содержимое PC автоматически увеличивается на 1 при выборке очередного байта команды (или очередной команды) из памяти в процессе выполнения программы.

Регистровая пара **PSW** называется словом состояния процессора. Она образуется регистрами A и F.

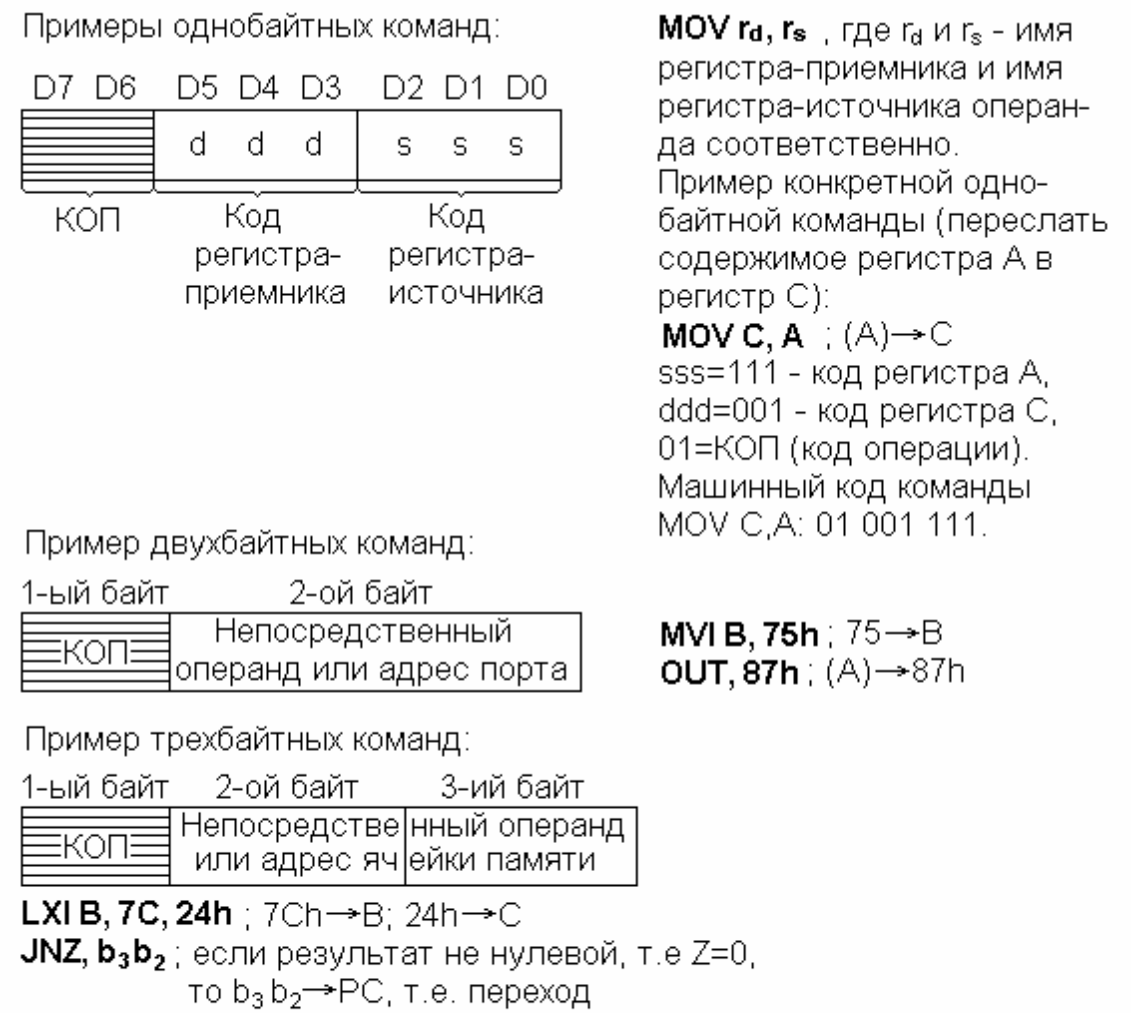


Рис.2.5. Форматы команд

2.3. Форматы команд и данных

Команды микропроцессора ВМ80 имеют однобайтный, двухбайтный и трехбайтный форматы. Поле команды содержит информацию (в кодах) о типе операции (код операции), об адресе операнда, типе обращения к памяти, о количестве байт в команде. Примеры команд приведены на рис. 2.5.

2.4. Режимы адресации

Способ определения источников и приемников операндов называют режимами адресации данных. Для микропроцессоров разработано около двух де-

сятков режимов адресации. В микропроцессоре ВМ80 используются четыре режима адресации данных.

1. Прямая (абсолютная) адресация. Это наиболее простая, но наименее экономичная адресация. В поле самой команды содержится полный 16-битный адрес операнда в памяти (рис. 2. 6, *а*). С помощью прямой адресации можно обращаться к любой ячейке в адресном пространстве.

2. Непосредственная адресация. Данные (байт или два байта) находятся непосредственно в команде во втором или во втором и третьем ее байтах (рис. 2. 6, *б*).

3. Регистровая адресация. Операндом является содержимое адресуемого в команде регистра (рис.2. 6, *в*). Команды с регистровой адресацией имеют однобайтный формат, выполняются достаточно быстро. В однобайтных командах с неявной адресацией подразумевается, что операнд находится в определенном внутреннем регистре МП и его специально адресовать не надо. Например, все команды сдвига микропроцессора ВМ80 или команда инвертирования оперируют содержимым аккумулятора.

4. Косвенная регистровая адресация (рис. 2. 6, *г*) позволяет компактно адресовать все пространство памяти. В этом режиме в поле команды содержится указание на регистровую пару, содержащую адрес операнда (данных). Косвенная адресация широко применяется при обращении к структурам данных типа массивов.

Вопросы и задания

2.11. Какие устройства образуют программную модель МП ВМ80 и МПС на его основе?

2.12. Убедитесь в своих знаниях программной модели МП ВМ80 и МПС на его основе, изобразив программно-доступные узлы.

2.13. Какие команды определяют доступ к устройствам ввода-вывода?

2.14. Назовите очень важные устройства МП, которые программно- недоступны.

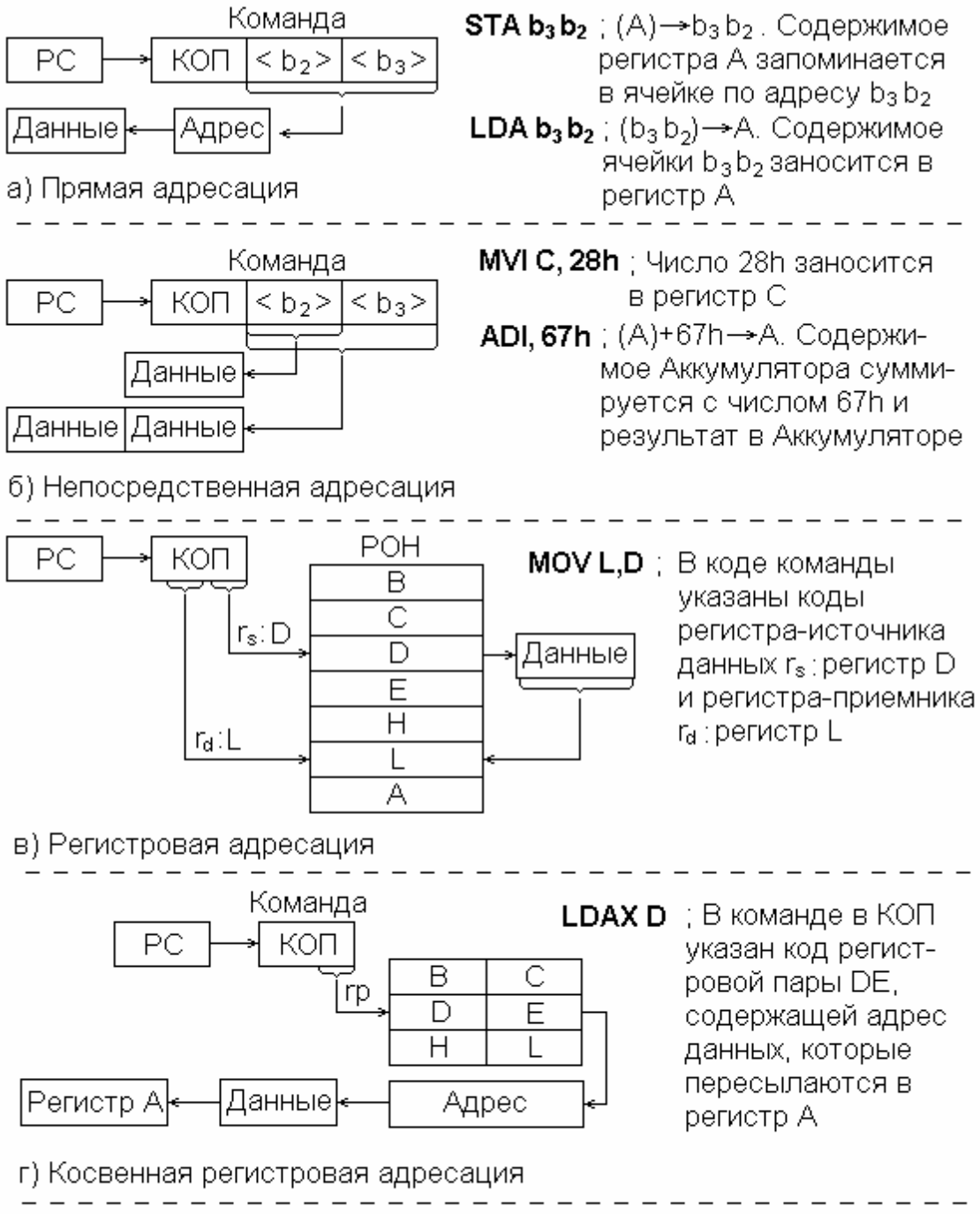


Рис.2.6. Режимы адресации данных

2.15. Поясните кодирование команд, их форматы и назначение.

2.16. Приведите примеры одно-, двух- и трехбайтных команд и определите по справочнику их машинные коды.

2.17. Поясните способы адресации к командам.

2.18. Какова адресация данных в МПС на МП ВМ80?

2.19. Какова адресация к стеку в МПС на ВМ80?

2.20. Определите способы адресации: STA 80C2h; IN, 58h;
ADI 4Ah; OUT, 58h;
MOV C, M; LHLD 94C2h;
MOV E, D; PUSH PSW.

2.21. Поясните, почему ВМ80 может выполнять только установленные для него команды и какими его средствами это определяется.

2.22. На какие группы можно подразделить систему команд МП ВМ80?

2.5. Система команд МП

Систему команд МП можно представить в виде 76 мнемокодов (ассемблерных кодов). Здесь команды ВМ80 рассматриваются с позиции описания действий микропроцессора. Всю систему команд можно по функциональным признакам разделить на группы.

- Пересылки/загрузки.
- Положительных/отрицательных приращений.
- Арифметические.
- Логические.
- Сдвига в аккумуляторе.
- Передачи управления.
- Ввода-вывода.
- Специальные.

2.5.1. Команды пересылки/загрузки

В любой прикладной программе возникает необходимость передать содержимое регистра-источника (src-регистра) в регистр-приемник (dst-регистр). Эту операцию выполняют многочисленные команды пересылок dst←(src). При этом содержимое src-регистра и состояние флажков в регистре признаков F не изменяются. Обобщенный мнемокод команды пересылок можно представить в

виде **MOV dst, src;** содержимое регистра-источника пересылается в регистр-приемник, т. е. dst←(src).

в качестве src и dst могут быть указаны любые РОН (B, C, D, E, H, L), а также регистр A и ячейка M. Пересылки возможны из любого РОН, регистра A и ячейки M в любой РОН, регистр A и ячейку M. Микропроцессор не выполняет только пересылку из ячейки M в ячейку M.

В описании системы команд, приведенных в приложении 2 и методических указаниях к выполнению лабораторных работ [28], регистр-источник операнда обозначается $\mathbf{r_s}$, регистр-приемник $\mathbf{r_d}$, а обобщенный мнемокод регистровой пересылки в методических указаниях имеет вид: $\mathbf{MOV\ r_d, r_s; (r_s) \rightarrow r_d.}$

Пусть необходимо переслать содержимое регистра B в регистр A. Здесь src:=B, а dst:=A. Обобщенный мнемокод примет вид MOV A, B, т. е. содержимое регистра B пересылается в регистр A.

Рассмотрим машинный код команды MOV A, B, обобщенный вид которой приведен на рис. 2. 7. Кодирование команды MOV dst, src.

sss: = код рег. src; ddd: = код рег. dst. Кодирование конкретной команды MOV A, B (рис. 2.8). Код регистра B:=000=SSS. Надо поставить в поле src вместо sss код 000. Код регистра A:=111 надо поставить вместо ddd. Машинный код команды MOV A, B будет 0111 1000 B = 78h. Содержимое регистра B записалось в регистр A, при этом содержимое регистра B не изменилось.

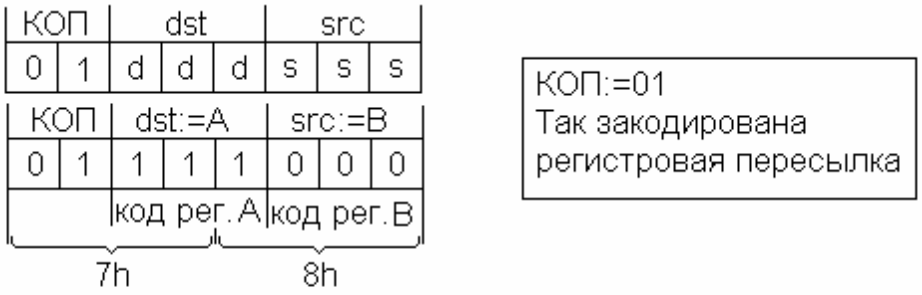


Рис.2.7. Машинный код команды MOV A,B

Рассмотрим действие команды MOV A, B (рис.2.8).

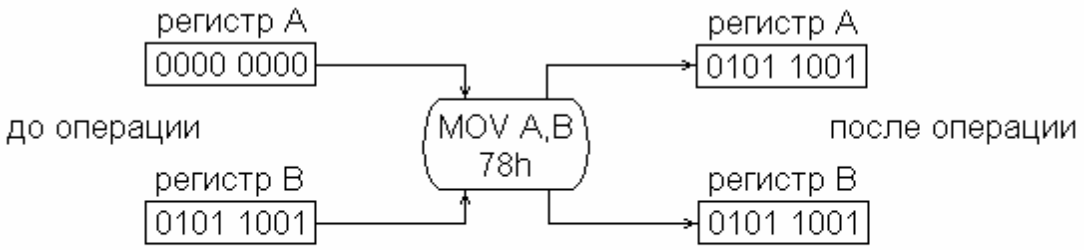


Рис.2.8. Действие команды MOV A,B

Непосредственная загрузка регистров.

Это двухбайтные команды, обобщенный вид мнемокода: **MVI r_d, b_2**, где b_2 – второй байт команды, является непосредственным операндом (числом), который загружается в регистр-приемник r_d.

Пусть необходимо число 75h загрузить в аккумулятор. Применим команду: MVI A, 75h; 75h → A.

Двухбайтные загрузки в регистровые пары. Обобщенный вид: **LXI r_p, b_3b_2,**

где r_p: B, D, H, SP, т. е. регистровые пары BC, DE, HL, SP, b_3b_2 – 16-разрядное число. Действие команды: $b_2 \rightarrow r_{PL}$, $b_3 \rightarrow r_{PH.}$ Пример. Загрузить указатель стека SP (рис. 2. 9). В результате выполнения этой команды b_2=A0h загружается в младшую часть указателя стека $SP_L \leftarrow$ A0h; b_3=28h загружается в старшую часть SP, т. е. $SP_H \leftarrow$ 28h.

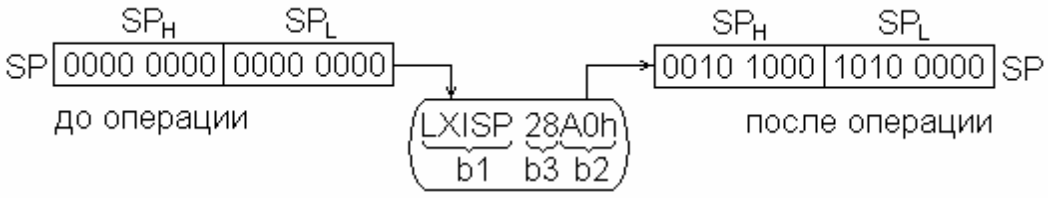


Рис.2.9. Загрузка указателя стека SP

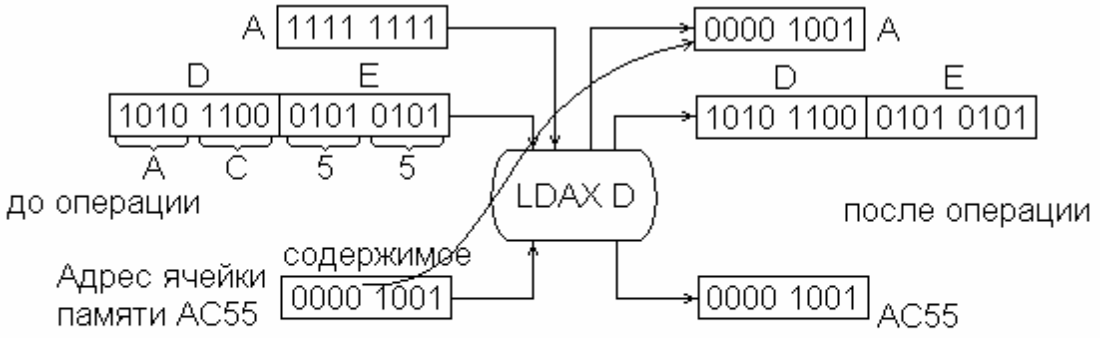


Рис.2.10. Выполнение команды LDAX D

Запоминания/загрузки регистра A и регистровой пары HL.

STA b_3 b_2; (A) → b_3b_2, т. е. содержимое A запоминается в ячейке b_3b_2.

LDA b_3 b_2; (b_3b_2) → A, т. е. содержимое ячейки b_3b_2 заносится в регистр A.

STAX r_p; (A) → (r_p), т. е. содержимое регистра A запоминается в ячейке, адрес которой находится в регистровой паре r_p: BC, DE.

LDAX r_p; ((r_p)) → A, т. е. содержимое ячейки, адресуемой указанной регистровой парой, заносится в регистр A. Действие команды LDAX D рассмотрено на рис. 2. 10. В результате содержимое ячейки, адрес которой находится в регистровой паре DE, записалось в регистр A.

SHLD b_3b_2; (L) → b_3b_2, (H) → b_3b_2+1. Содержимое HL пересылается в ячейки памяти с прямой адресацией.

Запись содержимого регистровых пар в стек.

PUSH r_p, где r_p: BC, DE, HL, PSW. (r_{PH}) → (SP)-1, (r_{PL}) → (SP)-2, (SP)-2 → SP.

Рассмотрим выполнение команды PUSH D (рис. 2.11).

При выполнении команды PUSH D МП выполняет следующие действия:

1) (SP)-1→ SP;

2) (D) → (SP) = 0F32;

3) (SP)-1 → SP;

4) (E) → (SP) = 0F31.

Вывод из стека в регистровые пары:

POP r_p. Эта команда считывает содержимое вершины стека (двух ячеек) и заносит в указанную регистровую пару $r_{p,}$ где r_p: BC, DE, HL, PSW.

Рассмотрим действие вывода из стека на примере выполнения команды POP D (рис. 2.12).

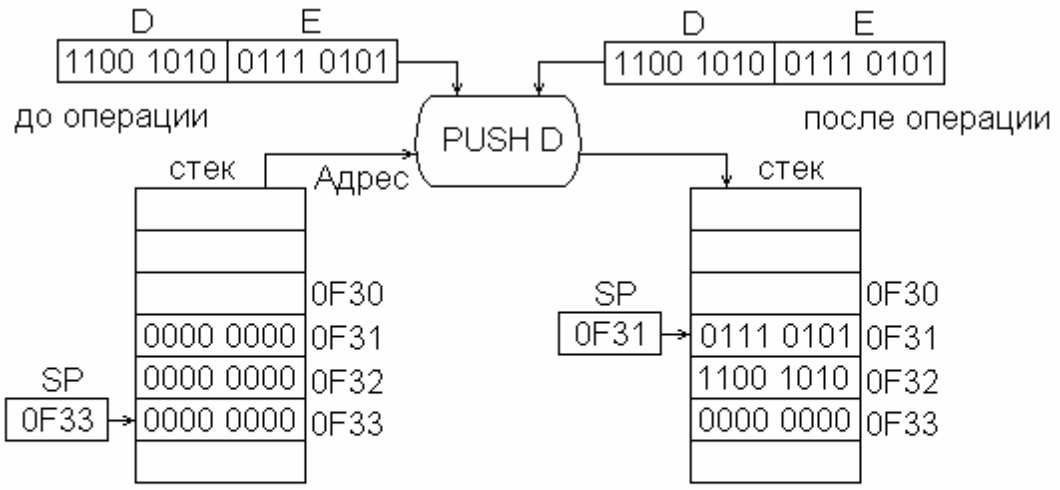


Рис.2.11. Выполнение команды PUSH D

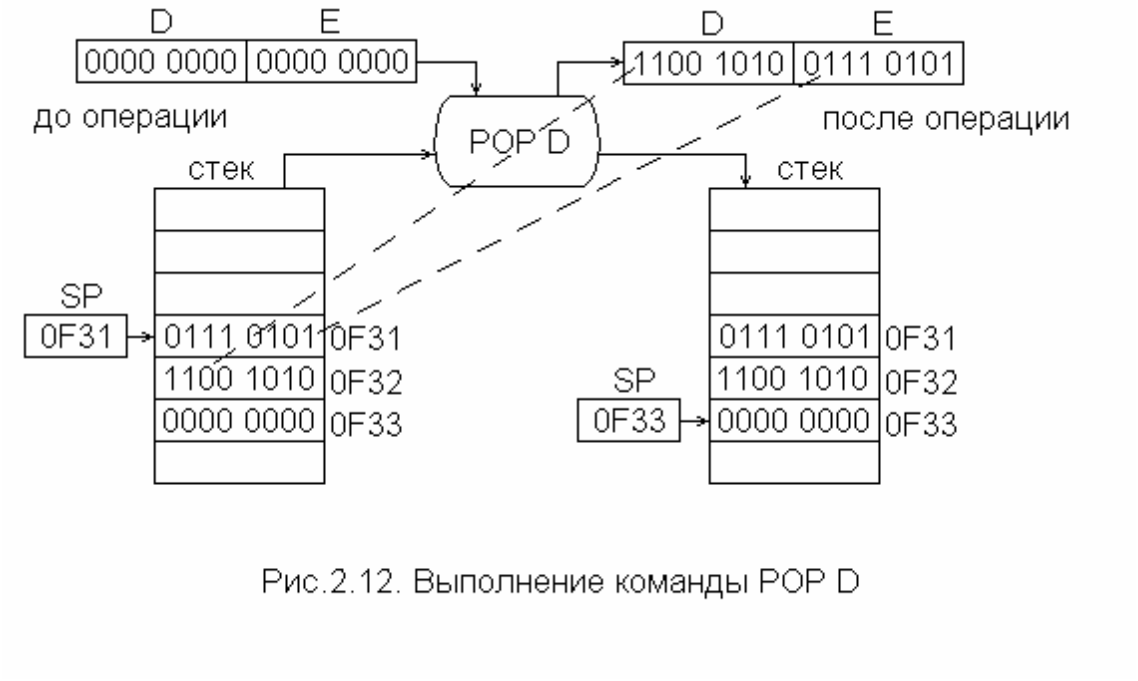


Рис.2.12. Выполнение команды POP D

При выполнении команды POP D МП выполняет следующие действия:

1) ((SP)) → D;

2) (SP)+1 → SP;

3) ((SP)+1) → E;

4) (SP)+1 → SP.

Обмен между регистрами DE и HL: XCHD; H ↔ D, L ↔ E.

Обмен вершины стека с HL: XTHL; (L) ↔ ((SP)), (H) ↔ ((SP)+1).

Пересылка содержимого HL в указатель стека SP: SPHL; (HL) → SP.

2.5.2. Команды положительных/отрицательных приращений

Признаки результата модифицируются, кроме CY:

S	Z	AC	P	CY
+	+	+	+	–

Положительные/отрицательные приращения РОН, регистра A и ячейки M.

INR r; (r)+1 → r,

DCR r; (r)-1 → r,

где r: B, C, D, E, H, L, M. A.

Положительные/отрицательные приращения регистровых пар:

INX r_p; (r_p)+1 → r_p ,

где r_p: BC, DE, HL.

DCX r_p; $(r_p)-1 \rightarrow r_p$,

При этом флаги не модифицируются.

Задания

2.23. Приведите примеры команд однобайтных и двухбайтных пересылок, загрузок и размещений.

2.24. Опишите последовательность действий МП в каждом машинном цикле при выполнении команды LDA 78A6h.

2.25. Вы можете проверить свои знания функционирования МП. Выберите любую команду и поясните, какие действия выполняются в каждом машинном цикле при ее выполнении.

2.26. Поясните выполнение команды PCHL.

2.27. Опишите по циклам выполнение команды SHLD.

2.28. Определите содержимое аккумулятора:

LXI H, 35E7h;	LXI H, 35E7h;
INX H;	SHLD 854Ch;
SHLD 854Ch;	INX H;
LDA 854Ch;	LDA 854Ch.

2.29. Определите содержимое ячейки памяти 1000h:

MVI A, 00h;

MOV B, A;

STA 1000h.

2.30. Определите содержимое указателя стека SP и ячейки B06Ah:

LXI SP B069h;

MVI A, 45h;

PUSH PSW.

2.5.3. Арифметические команды

Арифметические команды модифицируют все признаки в регистре F.

S	Z	AC	P	CY
+	+	+	+	+

Арифметические команды выполняют операции сложения, сложения с учетом переноса, вычитания и вычитания с учетом заема.

Команда сложения без учета бита переноса:

ADD r; (A)+(r) → A, где r: A, B, C, D, E, H, L, M.

Команда сложения с учетом бита переноса:

ADC r; (A)+(r)+(CY) → A.

Команда вычитания без учета бита заема:

SUB r; (A)-(r) → A.

Команда вычитания с учетом бита заема:

SBB r; (A)-(r)-(CY) → A.

Рассмотрим выполнение команды SBB E (рис. 2.13).

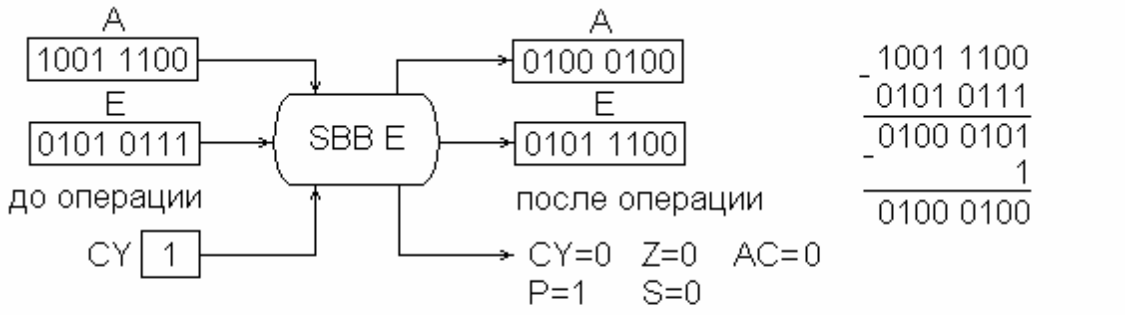


Рис.2.13. Выполнение команды SBB E

Арифметические команды с непосредственной адресацией. Сложение с непосредственным операндом:

ADI, b_2; (A)+b_2 → A, т. е. содержимое аккумулятора сложить с непосредственным операндом b_2.

ACI , b_2; (A)+b_2+(CY) → A, т. е. сложение содержимого регистра A, непосредственного операнда b_2 и бита переноса CY.

Вычитание непосредственного операнда:

SUI , b_2; (A)-b_2 → A, т. е. из содержимого аккумулятора вычесть непосредственный операнд b_2.

SBI , b_2; (A)-b_2-(CY) → A.

Команды сравнения **CMP r; CPI, b_2** выполняют операцию сравнения содержимого аккумулятора с содержимым регистра r или непосредственным операндом b_2. При этом содержимое аккумулятора не изменяется, но формируются признаки в регистре F.

CMP r; (A)-(r), где r: B, C, D, E, H, L, M, A.

CPI , b_2; (A)-b_2.

Команда сложения удвоенной точности:

DAD r_p; (HL)+(r_p) → HL, r_p: регистровые пары BC, DE, HL. При выполнении этой команды устанавливается только флаг CY.

Команда десятичной коррекции **DAA** применяется после выполнения операций сложения двоично-десятичных чисел с целью преобразования результата из двоичной формы в двоично-десятичную. При выполнении команды **DAA** реализуются следующие действия:

1. Если значение младшей тетрады аккумулятора больше кода 1001 или если флаг AC=1, то к содержимому аккумулятора прибавляется число 0110.
2. Если после этого старшая тетрада содержимого аккумулятора имеет код большей 1001 или если флаг CY=1, то к содержимому аккумулятора прибавляется число 0110 0000.

2.5.4. Команды логических операций

Признаки результата: S Z AC P CY

+	+	+	+	0
	+	0	0	

Команда побитной конъюнкции: **ANA r**; (A) & (r) → A.

Команда побитной дизъюнкции: **ORA r**; (A) v (r) → A.

Команда побитного сложения по модулю два:

XRA r; (A) ⊕ (r) → A. В этих командах r : B, C, D, E, H, L, M, A.

Рассмотрим выполнение команды ANA E (рис. 2.14).

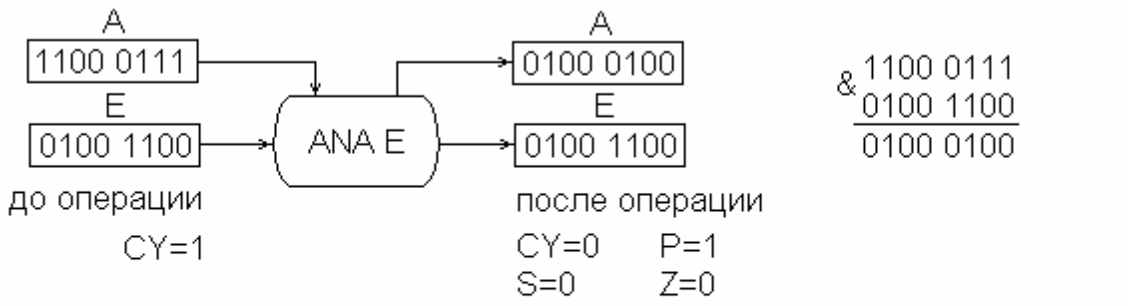


Рис.2.14. Выполнение команды ANA E

Логические команды с непосредственной адресацией (непосредственным операндом b_2).

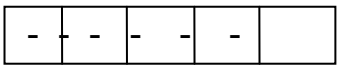
ANI , b_2; (A) & $b_2 \rightarrow$ A.

ORI , b_2; (A) v $b_2 \rightarrow$ A.

XRI , b_2; (A) $\oplus$ $b_2 \rightarrow$ A.

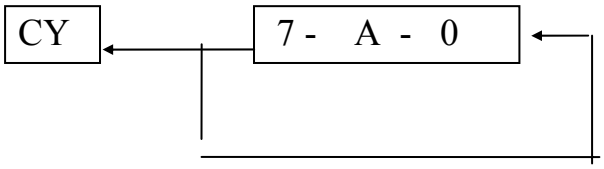
2.5.5. Команды сдвига в аккумуляторе на 1 разряд

Признаки результата: S Z AC P CY



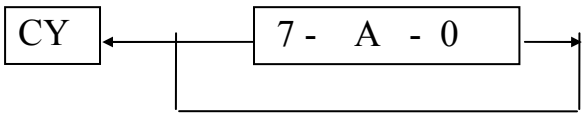
Сдвинуть циклически влево:

RLC;



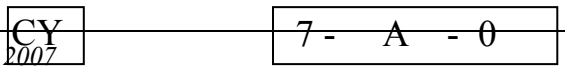
Сдвинуть циклически вправо:

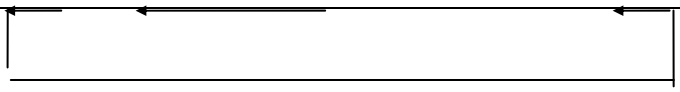
RRC;



Сдвинуть циклически влево через бит CY:

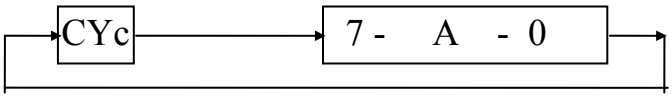
RAL;





Сдвинуть циклически вправо через бит CY:

RAR;



2.5.6. Команды передачи управления

Признаки результата: S Z AC P CY

-	+	-	-	-	-	

После выборки кода команды из памяти в программном счетчике PC формируется адрес следующей по порядку команды методом автоматического увеличения содержимого PC на единицу. Такой механизм ведения по программе имеет место на линейных ее участках. В разветвляющихся или циклических программах и при использовании подпрограмм встречаются переходы, когда выполняется не следующая по порядку команда, а команда, находящаяся в другом месте программной памяти. Команды, выполняющие такие переходы, называют командами передачи управления. Команды передачи управления осуществляют перезагрузку PC адресом передачи управления, который указывается в кодах этих команд.

Команды безусловной передачи управления:

PCHL; (H) $\rightarrow PC_H$, (L) $\rightarrow PC_L$. Перейти без условия по адресу, находящемуся в HL.

JMP b_3b_2; $b_2 \rightarrow PC_L$, $b_3 \rightarrow PC_H$. Перейти без условия по адресу b_3b_2.

Команды передачи управления по условию выполняют передачу управления, если условие выполняется, и не выполняют перехода, если указанное условие не выполняется. Условия определяются по значениям признаков в регистре флагов CY, P, Z, S по их единичному или нулевому значению. Восемь команд передачи управления приведены в табл. 2.1. Все восемь команд передачи управления по условию можно представить в виде одного мнемокода **Jcon** $\mathbf{b_3b_2}$ и одного кода команды 11CCC010, где con – переменная часть мнемокода, CCC – код условия передачи управления (см. табл. 2.1).

Таблица 2.1

con	CCC	Условия передачи управления
NZ	000	Если результат не нулевой, т. е. Z=0
Z	001	Если результат нулевой, т.е. Z=1
NC	010	Если не было переноса/заема, т. е. CY=0
C	011	Если был перенос/заем, т. е. CY=1
PO	100	Если результат нечетный, т. е. P=0
PE	101	Если результат четный, т. е. P=1
P	110	Если результат положительный, т. е. S=0
M	111	Если результат отрицательный, т. е. S=1

Рассмотрим выполнение МП одной команды передачи управления по условию Z=1 (рис. 2.15).

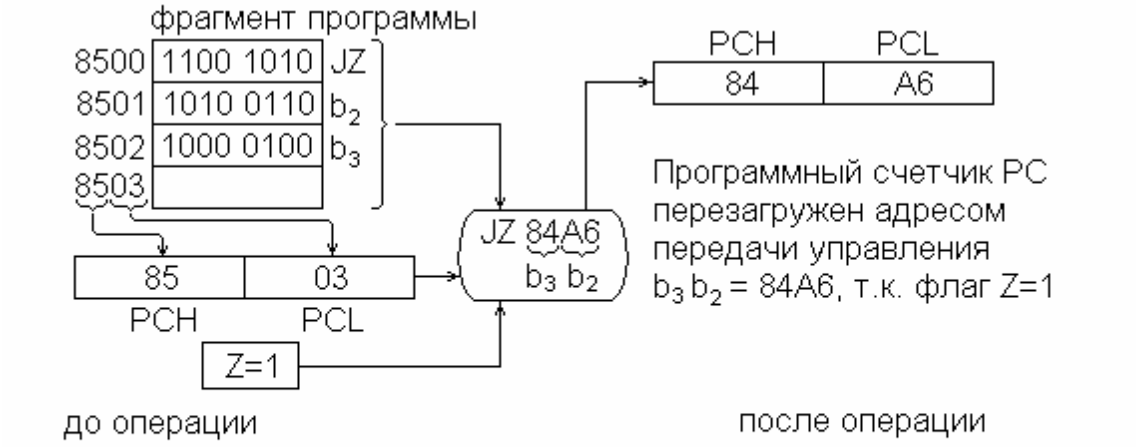


Рис.2.15. Выполнение команды передачи управления по условию Z=1

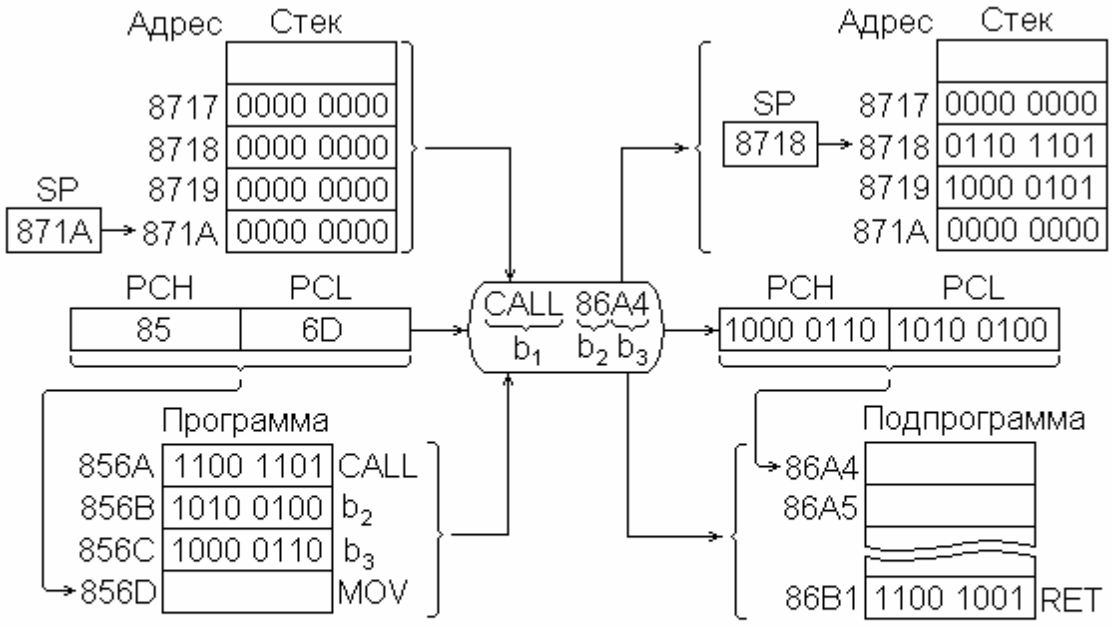


Рис.2.16. Выполнение команды CALL 86A4

Вызов подпрограмм (команда CALL b_3b_2). Команда вызова подпрограмм CALL b_3b_2 состоит из первого байта (код операции) и адреса вызываемой подпрограммы (b_3b_2). При выполнении этой команды текущее содержимое PC (т. е. адрес команды, следующей за CALL) автоматически запоминается в стеке. Этот адрес (называемый адресом возврата) возвращается в PC командой возврата **RET**, которой должно завершаться выполнение вызванной подпрограммы.

Для запоминания адресов возврата необходима магазинная память (стек), работающая по принципу: первый записанный в нее адрес возврата извлекается последним.

При выполнении команды CALL b_3b_2 выполняются следующие действия: (PC) $\rightarrow$ в стек, т. е. адрес возврата загружается в стек, $b_3b_2 \rightarrow$ PC, т. е. адрес подпрограммы загружается в PC.

Рассмотрим выполнение команды CALL 86 A4 (рис. 2.16). При выполнении команды CALL 86A4 реализуются следующие действия: программный счетчик, ведя по программе, считывает из нее команду CALL 86A4. Содержимое PC после считывания последнего байта команды CALL увеличивается на единицу и равно 856Dh. Далее начинается выполнение считанной в МП команды CALL 86A4. Содержимое PC=856D заносится в стек. PC загружается начальным адресом подпрограммы PC=86A4, и МП приступает к выполнению подпрограммы.

1) $(PC_H) \rightarrow ((SP)-1)$;

2) $(PC_L) \rightarrow ((SP)-2)$, т. е. (PC) $\rightarrow$ в стек;

3) (SP-2) $\rightarrow$ SP; новое содержимое SP;

4) $b_2 = A4 \rightarrow PC_{L;}$

5) $b_3 = 86 \rightarrow PC_{H;}$ т. е. начальный адрес подпрограммы загружен в PC.

Команда возврата из подпрограммы **RET:** $((SP)) \rightarrow PC_L$; $((SP+1)) \rightarrow PC_H$; SP: = (SP)+2.

Содержимое двух ячеек вершины стека заносится в PC. Таким способом реализуется возврат в «основную» программу.

Микропроцессор ВМ80 имеет восемь команд условных вызовов подпрограмм и восемь условных возвратов из подпрограмм. Проверяются единичные и нулевые значения флагов Z, CY, P, S. Если проверяемое в команде вызова или возврата условие выполняется, то осуществляется вызов или возврат. А если условие не выполняется, то выполняется следующая по порядку команда.

Команды условного вызова и условного возврата в таблице команд представлены в виде обобщенных мнемокодов команд (см. прил. 2)

C con b_3b_2 и **R con**.

В системе команд МП имеется особая 1-байтная команда вызова, предназначенная для обработки прерываний, введения контрольных точек при отладке программ. Она называется рестартом **RST n**. В коде команды RSTn значение n=11NNN111 три бита NNN формируется подсистемой прерываний или задаются программистом.

Имеется восемь команд RST n = 0, 1, 2, 3, 4, 5, 6, 7. Выполнение команды сводится к двум действиям:

- текущее содержимое PC загружается в стек;
- в PC передается код 0000 0000 00NN N000B. Здесь NNN есть двоичный код **n.**

Таким образом, в зависимости от NNN микропроцессор переходит к одной из восьми ячеек памяти, которые могут быть начальными адресами подпрограмм.

2.5.7. Команды операций ввода-вывода

IN , port; (port) → A, где port – 8-битный адрес порта ввода.

OUT , port; (A) → port, где port – 8-битный адрес порта вывода.

2.5.8. Специальные команды

Инвертирование аккумулятора CMA; (A) → A.

Установка и инвертирование бита CY STC; 1 → CY; CMC; (CY) → CY.

Разрешение прерываний EI.

Запрещение прерываний DI.

«Пустая» команда NOP, пропуск 4 тактов.

Команда останова HLT. МП воспринимает запросы прерывания и запросы шин.

Полный список команд приведен в приложении 2.

Вопросы и задания

2.31. Поясните выполнение команд ADD и ADC, SUB и SBB и какие признаки формируются в регистре F.

2.32. Поясните команду DAD. Обратите внимание, что при ее выполнении формируется только признак переноса CY.

2.33. Перечислите команды, позволяющие сравнивать двоичные коды.

2.34. Поясните отличие команд: INR H; DCR D; INX H; DCX D.

2.35. Занесите в аккумулятор десятичное число 73 в двоично-десятичном коде и определите содержимое аккумулятора и признаки результата после выполнения каждой из команд: ORI, 0Fh; ANI, F0h; XRA, A.

2.36. Содержимое какого устройства изменяют команды RLC, RRC, RAL и RAR? Поясните выполняемые ими операции.

2.37. Поясните, что общего и различного при выполнении команд: JMP 0000h; CALL 0000h; RST 0.

2.38. Для чего служат команды EI и DI?

2.6. Управление системой

В основе устройства управления МП используется цифровой автомат. Примерная схема алгоритма функционирования управляющего автомата в течение рабочего цикла выполнения команды приведена на рис. 2.17. Выполнение рабочего цикла команды начинается с опроса триггера прерывания. Если запрос прерывания поступил и прерывания разрешены (командой EI), то автомат формирует машинный цикл обработки прерывания, в котором управление передается подпрограмме обработки прерывания, и она выполняется. При отсутствии прерывания управляющий автомат создает цикл выборки команды из памяти и формирует адрес следующей команды.

Далее управляющий автомат дешифрирует код операции в команде и генерирует соответствующую коду операции серию управляющих сигналов, обеспечивающую выполнение в МП заданной операции.

Алгоритм работы управляющего автомата содержит условный оператор ожидания готовности операнда. Наличие такого оператора в алгоритме позволяет МП приспосабливаться для работы с различными видами внешней памяти, имеющей разное время доступа, а также с медленно действующими устройствами ввода-вывода (УВВ). Наличие в схеме алгоритма устройства управления, оператора ожидания готовности операнда, механизма анализа запросов на прерывание и запросов на захват шин позволяет МП формировать последовательность управляющих сигналов не только на основе команды, но и под воздействием внешних управляющих сигналов Ready, INT, HOLD.

Устройство управления МП в зависимости от кода текущей команды, состояния своего управляющего автомата, а также в зависимости от значений сигналов оповещения с шины управления МПС вырабатывает последовательности сигналов, реализующие процедуры системного обмена информацией.

В МП управляющий автомат в зависимости от сложности команды выполняет цикл команды за несколько (1 – 5) внутренних машинных циклов. Один машинный цикл требуется МП для одного обращения к памяти или УВВ. Машинный цикл МП ВМ80 может состоять из 3 – 5 тактов. Тактирование МП осуществляется от внешнего генератора сигналами F1, F2 (рис. 2.18, 2.19). В стандартном машинном цикле может быть от трех до пяти состояний автомата управления.

Микропроцессор ВМ80 приступает к анализу запросов на прерывание только после окончания выполнения текущей команды.

Из состояния останова МП может быть выведен сигналом прерывания INT или сигналом установки в исходное состояние Reset.

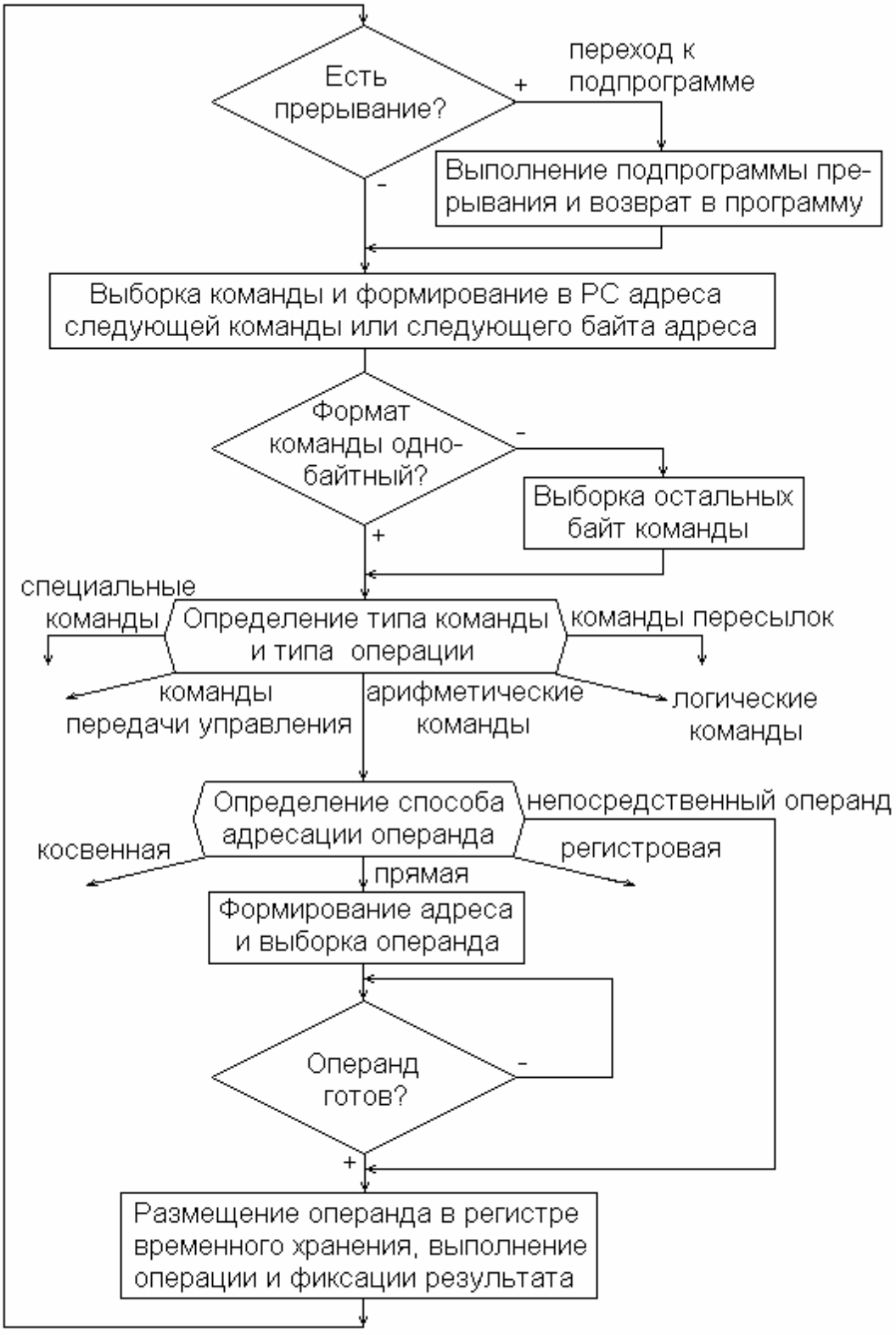


Рис.2.17. Рабочий цикл выполнения команды МП

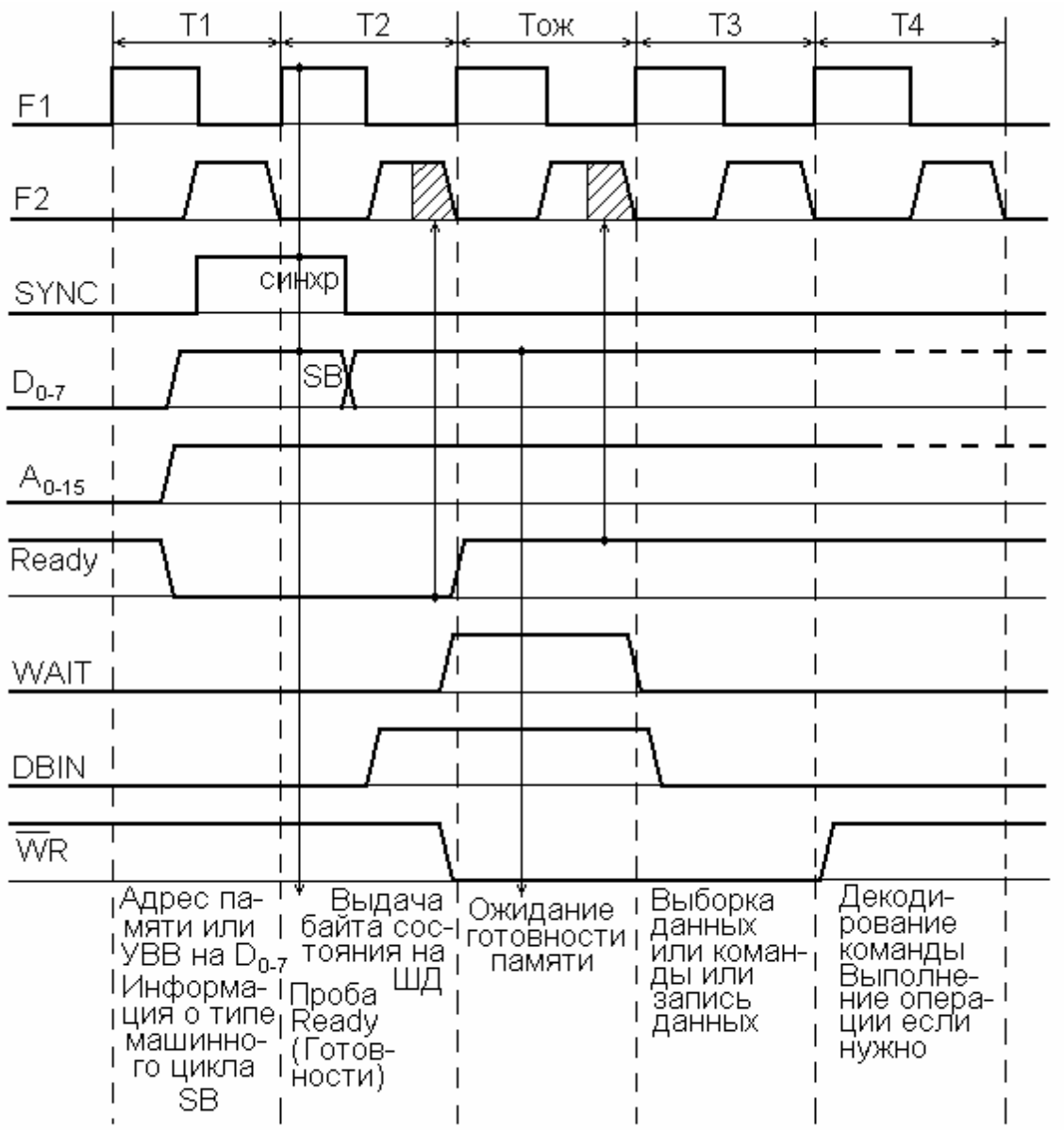


Рис.2.18. Временные диаграммы цикла выполнения команды

Временная диаграмма, изображенная на рис. 2.18, определяет основной цикл команды МП. В первом такте синхронизации Т1 МП выставляет на шине адреса код адреса A_{0-15} очередной команды. Одновременно на линии синхронизации **SYNC** появляется единичный сигнал, который идентифицирует информацию на шине данных D_{0-7} как байт состояния SB-процессора и загружает его в регистр системного контроллера К580ВК28 (рис. 2.19).

Сигнал SYNC также свидетельствует о начале машинного цикла. По окончании сигнала SYNC буферная схема шины данных, расположенная в сис-

темном контроллере, переводит шину данных D_{0-7} в режим ввода, о чем свидетельствует единичный сигнал на линии **DBIN** шины управления.

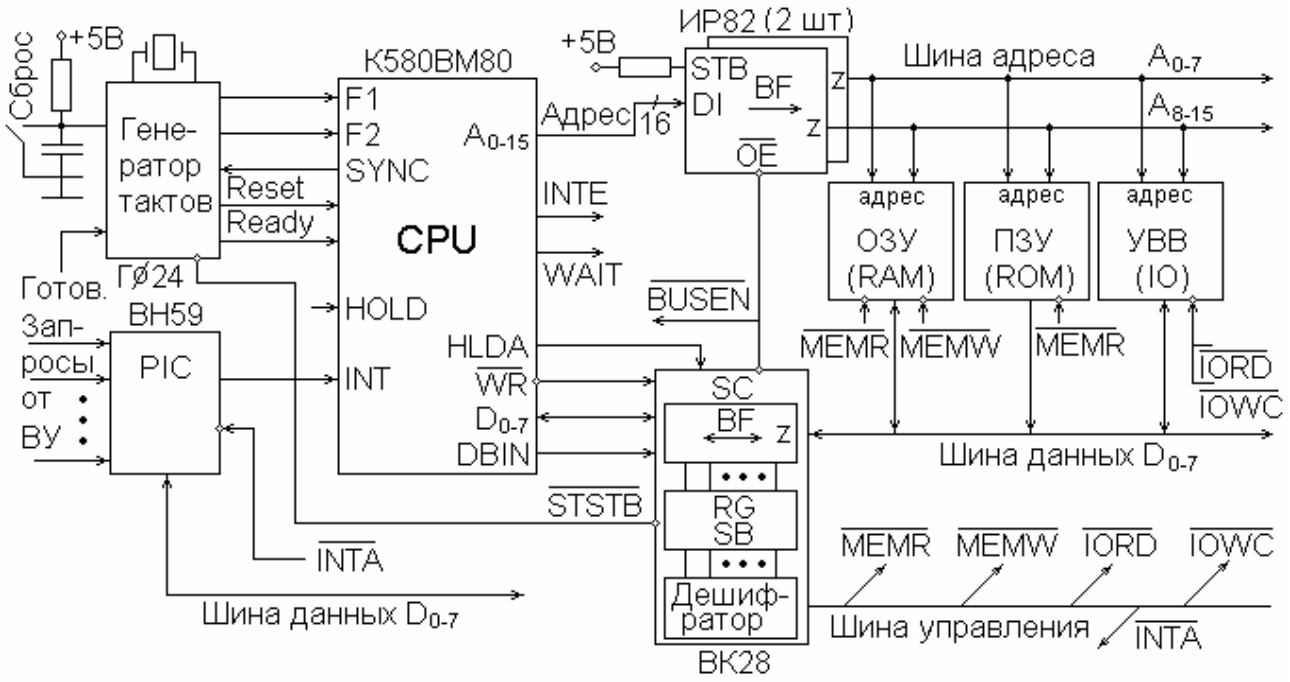


Рис.2.19. Структурная схема МПС

В такте T_2 МП осуществляет проверку готовности внешнего устройства (или памяти). Если внешнее устройство не формирует сигнал Готов (высокий уровень), то автомат управления МП переходит в состояние ожидания. В этом состоянии МП будет находиться до тех пор, пока на линии Ready не появится единичный сигнал, который будет свидетельствовать о том, что память или ВУ готовы к обмену. На временных диаграммах (см. рис. 2.18) приведен вариант отсутствия готовности в течение одного такта, и МП сформировал один такт ожидания Тож, в котором обнаружил единичный сигнал готовности Ready и перешел к рабочему такту T_3. В такте T_3 МП производит чтение или запись слова в память. Такты T_4 и T_5 отводятся для выполнения операции, заданной кодом операции команды. Выполнение некоторых сложных команд требует неоднократного прохода по циклу состояний автомата управления от T_1 до T_5.

Для нормального функционирования МПС недостаточно управляющих сигналов, формируемых МП на собственных выводах. Например, нельзя отличить циклы обращения к памяти от циклов обращения ввода-вывода. Расширение числа управляющих сигналов достигается с помощью специального 8-разрядного кода BS (байта состояния), который МП выдает через шину данных в первом такте T_1 каждого машинного цикла. Байт состояния BS содержит информацию о текущем машинном цикле. Всего 11 типов машинных циклов.

Тип	Мнемоника	Функциональное назначение.
1	FETCH	Цикл M1 приема первого байта команды в регистр IR .
2	MEMORY-READ	Цикл чтения данных из памяти по адресу, определяемому PC, BC, DE, HL.
3	MEMORY-WRITE	Цикл записи данных в память по адресу, определяемому BC, DE, HL, SP.
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.
10	HALT-INTERRUPT	Первый цикл подтверждения прерываний в состоянии останова.
11	INTERRUPT	Второй и третий циклы подтверждения прерывания.

Рассмотрим средства управления МПС на микропроцессоре ВМ80. Структурная схема ядра системы, приведенная на рис. 2.9, состоит из следующих блоков:

- микропроцессора ВМ80 (КР580ВМ80);
- генератора тактовых импульсов на микросхеме ГФ24 (КР580ГФ24);
- системного контроллера ВК28 или ВК38 (КР580ВК28, КР580ВК38);
- буферных регистров ИР82 (КР580ИР82);
- программируемого контроллера прерываний ВН59 (КР580ВН59);
- блоков оперативной и постоянной памяти (RAM, ROM);
- устройств ввода-вывода.

Назначение линий управления

F1 и **F2.** Это входные линии приема взаимно противофазных сигналов тактирования МП. Поступают с генератора тактовых импульсов (рис. 2.19).

SYNC. Выходная линия, на которую микропроцессор в начале каждого машинного цикла формирует сигнал синхронизации устройств, входящих в систему.

Ready. Входная линия приема сигнала от ВУ высоким уровнем, информирующая о готовности принять или выдать данные.

WAIT. Выходная линия, на которую МП выставляет сигнал ожидания (высокий уровень), если в такте Т2 отсутствует высокий уровень на линии Ready.

Reset. По этой линии поступает сигнал установки МП в исходное состояние. При этом в программный счетчик РС загружается стартовый адрес РС=0000h, с которого начинается пусковая программа. В регистр команд IR записывается код пустой команды NOP, т. е. IR=00h. Внутренние триггеры «разрешение прерывания» и «подтверждение захвата шины» устанавливаются в нулевое состояние. Состояние РОН и регистра признаков F по сигналу Reset не изменяется.

INT. Линия приема сигнала запроса на прерывание от ВУ. МП анализирует состояние этой линии в конце текущей команды или в состоянии «Останов». И если внутренний триггер «разрешения прерываний» установлен в состояние «разрешено» (командой EI), то МП формирует машинные циклы обработки прерывания, выставляя на шину данных байт состояния SB, на основе которого системный контроллер (см. рис. 2.19) вырабатывает сигналы INTA #.

INTE. Выходная линия, на которую МП формирует сигнал «разрешение прерывания».

HOLD. Входная линия приема сигнала запроса шин, поступающая от ВУ.

HLDA. Выходная линия, на которую микропроцессор выставляет сигнал «подтверждение захвата шин» в ответ на сигнал HOLD, при этом буферы на шинах адреса и данных устанавливает в Z-состояние.

DBIN Выходная линия, на которую МП формирует сигнал высокого уровня при выполнении команд чтения памяти или УВВ.

WR#. Выходная линия. При выполнении команд записи в память или УВВ МП выставляет сигнал низкого уровня.

Генератор тактовых импульсов формирует две взаимно противофазные импульсные последовательности F_1 и F_2. Микропроцессор воспринимает эти тактирующие сигналы, пересчитывает их с коэффициентом пересчета 3 – 5. Величина коэффициента пересчета определяется типом выполняемой в данный момент команды. В начале каждого машинного цикла микропроцессор формирует сигнал синхронизации (**SYNC**), который поступает на генератор тактовых импульсов. В генераторе тактовых импульсов с приходом сигнала **SYNC** формируется сигнал **STSTB#**, служащий для стробирования записи байта состояния SB процессора в регистр системного контроллера. Таким образом, в каждом машинном цикле МП формирует код типа текущего машинного цикла, который далее декодируется в системном контроллере. В результате декодирования кода типа машинного цикла формируется расширенный набор сигналов

шины управления для управления памятью, устройствами ввода-вывода и прерываниями. Все сигналы имеют активный низкий уровень:

MEMRD# – строб-сигнал чтения памяти.

MEMWR# – строб-сигнал записи в память.

IORD# – строб-сигнал чтения УВВ (портов).

IOWRC# – строб-сигнал записи в УВВ (порты).

INTA# – сигнал подтверждения прерывания. Используется для стробирования чтения адреса подпрограммы обработки прерывания, например из контроллера прерывания КР580ВН59.

П р и м е ч а н и е. Символы #, / используют для указания низкого активного уровня.

Вопросы и задания

2.39. Поясните процесс выполнения простейших команд MOV A,D; MOV C,M.

2.40. Выполните подробное пояснение выполнения команд LDA и LDAX.

2.41. Поясните процесс выполнения команды SHLD.

2.42. Каким образом осуществляется разделение адресных пространств памяти и портов?

2.43. Изобразите структурную схему МПС и поясните управление памятью, портами и прерываниями.

2.44. Поясните назначение байта состояния SB, формируемого МП в первом такте каждого машинного цикла.

2.45. Можно ли память стека расположить в отдельном адресном пространстве?

2.46.Поясните возможность МП осуществлять обмен данными с «медленной» памятью.

3. ИНТЕРФЕЙСНЫЕ БИС

Для создания МП-устройств, МП-систем выпускаются семейства взаимосовместимых БИС, называемые микропроцессорными комплектами (МПК).

МПК серии К580 можно рассматривать в качестве примера при построении МП-систем на базе 8-разрядных микропроцессоров первого поколения. Микросхемы серии К580 широко описаны в учебно-методической и инженерно-справочной литературе, что позволяет быстро изучить организацию отдельных подсистем и МП-систем в целом. Приведем основные БИС этой серии:

К580ГФ24 – генератор тактовых импульсов.

К580ИР82 – буферный регистр.

К580ВА86 – двунаправленный шинный формирователь.

К580ВК28 – системный контроллер.

К580ВК38 – системный контроллер.

К580ВВ55 – программируемый параллельный интерфейс.

К580ВВ51 – программируемый последовательный интерфейс.

К580ВИ53 – программируемый таймер.

К580ВИ54 – программируемый таймер.

К580ВН59 – программируемые контроллеры прерываний.

К580ВН59А – программируемые контроллеры прерываний.

К580ВТ57 – программируемые контроллеры прямого доступа к памяти.

К580ВТ37 – программируемые контроллеры прямого доступа к памяти.

К580ВВ79 – программируемый контроллер клавиатуры и индикации.

К580ВК91– приемопередатчики шины IEEE-488.

К580ВК92 – контроллер шины IEEE-488.

Наборы интерфейсных БИС упрощают разработку программного обеспечения (ПО), повышают производительность МПС и облегчают проектирование ее аппаратных средств.

Периферийные БИС серии К580 удовлетворяют электрическим и логическим спецификациям на микропроцессорную шину Microbus – унифицированную 8-разрядную шину, объединяющую в функционально законченные модули отдельные компоненты.

3.1. Программируемый параллельный адаптер

Для организации программно-управляемого обмена в параллельном коде с периферийными устройствами находит широкое применение микросхема КР580ВВ55.

Программируемый периферийный адаптер (ППА) КР580ВВ55 (далее ВВ55) применяется для параллельного обмена данными с квитированием и без него как в режиме программного управления, так и по прерываниям. При этом организация однонаправленного или двунаправленного обмена данными выполняется программированием ВВ55.

В состав микросхемы ВВ55 (рис. 3.1) входят три двунаправленных 8-разрядных порта, разделенных на две группы, два устройства управления группами портов и периферийная логика для согласования с системной шиной. Порты содержат буферные регистры и шинные формирователи с тремя состояниями. Схема управления содержит регистр управляющего слова CW, доступный только для записи.

Обмен информацией между МП и внутренними регистрами ВВ55 осуществляется через двунаправленный шинный формирователь и управляется сигналами CS, A0, A1, RD, WR согласно табл. 3.1. Адресные сигналы A0, A1 выбирают один из внутренних регистров, стробы RD и WR управляют направлением передачи, а сигнал CS определяет доступ к микросхеме.

Вход **Reset** служит для аппаратного сброса микросхемы в исходное состояние. Все регистры ВВ55, включая регистр управляющего слова **CW**, устанавливаются в состояние 0.

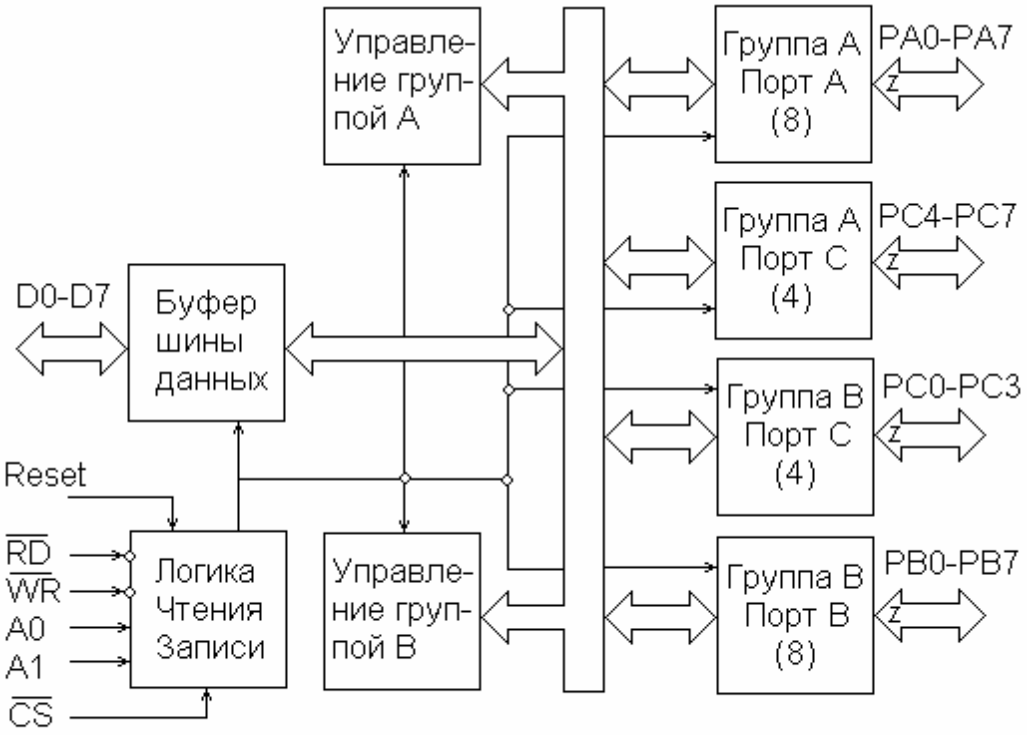


Рис.3.1. Структурная схема ВВ55

Таблица 3.1

A1	A0	RD	WR	CS	Операция
0	0	0	1	0	Чтение порта А
0	1	0	1	0	Чтение порта В
1	0	0	1	0	Чтение порта С
1	1	0	1	0	Недопустимо
0	0	1	0	0	Запись в порт А
0	1	1	0	0	Запись в порт В
1	0	1	0	0	Запись в порт С
1	1	1	0	0	Запись управляющего слова
x	x	x	x	1	Нет загрузки

Настройка микросхемы ВВ55 осуществляется с помощью управляющего слова **MS (Mode Selection),** которое назначает режим работы каждому порту (рис. 3.2, а). Каждое из управляющих устройств группы А или В принимает свою часть слова выбора режима **MS.**

Микросхему ВВ55 можно запрограммировать на следующие режимы работы ее портов:

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

режим 1 – однонаправленный ввод-вывод с квитированием (применим к портам А и В);

режим 2 – двунаправленный ВВ (только для порта А).

При работе портов А и В в режиме 1 и порта А в режиме 2 линии порта С используются для управления обменом с внешним периферийным устройством (ПУ).

С помощью управляющего слова **BSR (Bit Set/Reset)** осуществляется побитное программирование содержимого выходного буферного регистра порта С. Формат управляющего слова BSR приведен на рис. 3.2, б. Побитное программирование порта С необходимо при его использовании в качестве шины управления ПУ.

В режиме 0 осуществляется прямой однонаправленный обмен через любой из трех портов без сигналов сопровождения. В данном режиме порты А и В можно представить как две параллельные шины размером в один байт, а порт С – как две 4-разрядные параллельные шины. При этом каждую из указанных шин можно устанавливать на ввод или вывод независимо от других. Входные данные в микросхеме ВВ55 не запоминаются и читаются при низком уровне сигнала на входе RD. Выходная информация защелкивается в выходной буферный регистр выбранного порта по срезу системного сигнала WR и остается на выходе порта до нового цикла вывода или изменения режима.

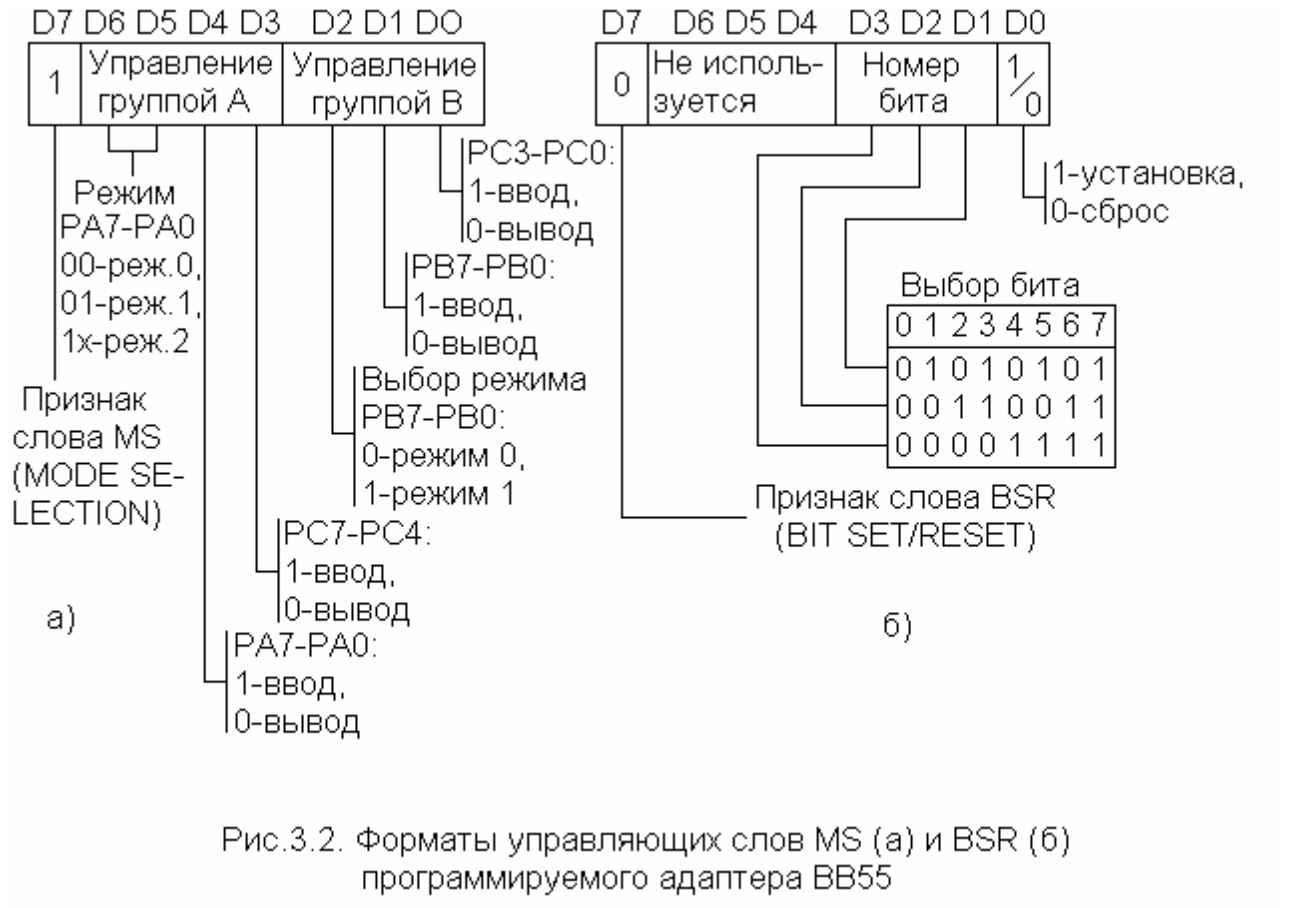


Рис.3.2. Форматы управляющих слов MS (а) и BSR (б) программируемого адаптера ВВ55

На рис. 3.3, а приведены временные диаграммы ввода данных из порта. Сигналами CS, A0, A1 МП выбирает соответствующий порт ВВ55, через который данные поступают на шину данных D_7-D_0 системы. При этом сигнал чтения RD открывает буфер на шине данных микросхемы в направлении из микросхемы на системную шину данных. Запись данных в порт приведена на рис. 3.3, б. Здесь также вначале осуществляется выборка порта сигналами CS, A0, A1. Далее МП формирует сигнал **WR**, выполняя команду вывода, затем выставляет данные на системную шину данных, которые записываются в выходной буфер выбранного порта.

Режим 1 обеспечивает однонаправленный обмен данными с квитированием через порты A и B. Входные и выходные данные фиксируются во внутренних регистрах портов A и B. Управление вводом (рис. 3.4, а) осуществляется сигналами:

STB (Strobe). Строб записи данных во входной регистр-защелку. Запись осуществляется по фронту STB.

IBF (Input Buffer Full). Подтверждение загрузки данных. Сигнал устанавливается по срезу STB и сбрасывается по фронту RD.

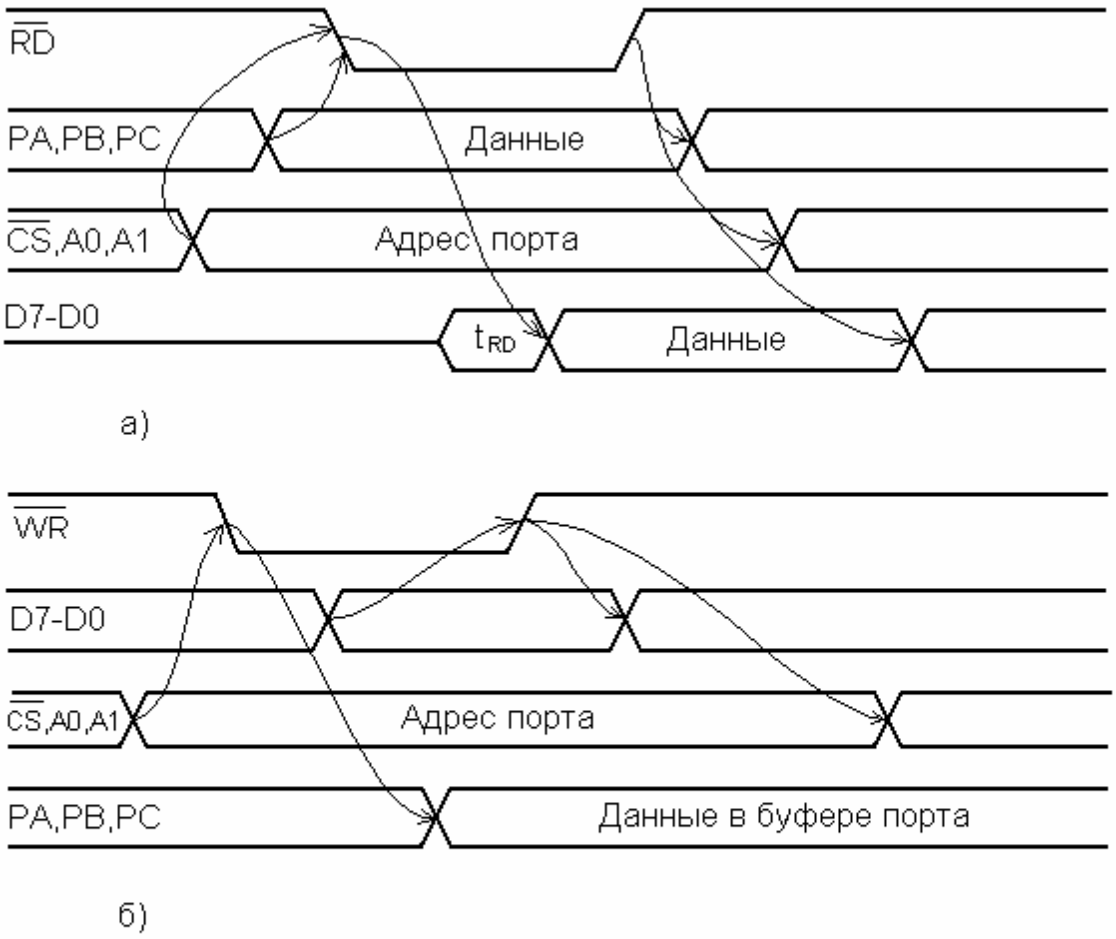


Рис.3.3. Временные диаграммы режима 0:
а) режим ввода; б) режим вывода

INT (Interrupt). Запрос на прерывание. Сигнал устанавливается по фронту STB и сбрасывается по срезу RD. Используется для организации ввода по прерываниям. Управление выводом (рис. 3.4, б) реализуется сигналами:

OBF (Output Buffer Full). Строб вывода новых данных. Устанавливается по фронту WR и сбрасывается по срезу АСК.

АСК (АСКnowledge). Подтверждение приема выходных данных со стороны ВУ, т. е. низким уровнем сигнала ВУ сообщается, что данные приняты.

INT (Interrupt). Запрос на прерывание. Сигнал устанавливается по фронту АСК и сбрасывается по срезу WR. Используется для обмена по прерываниям.

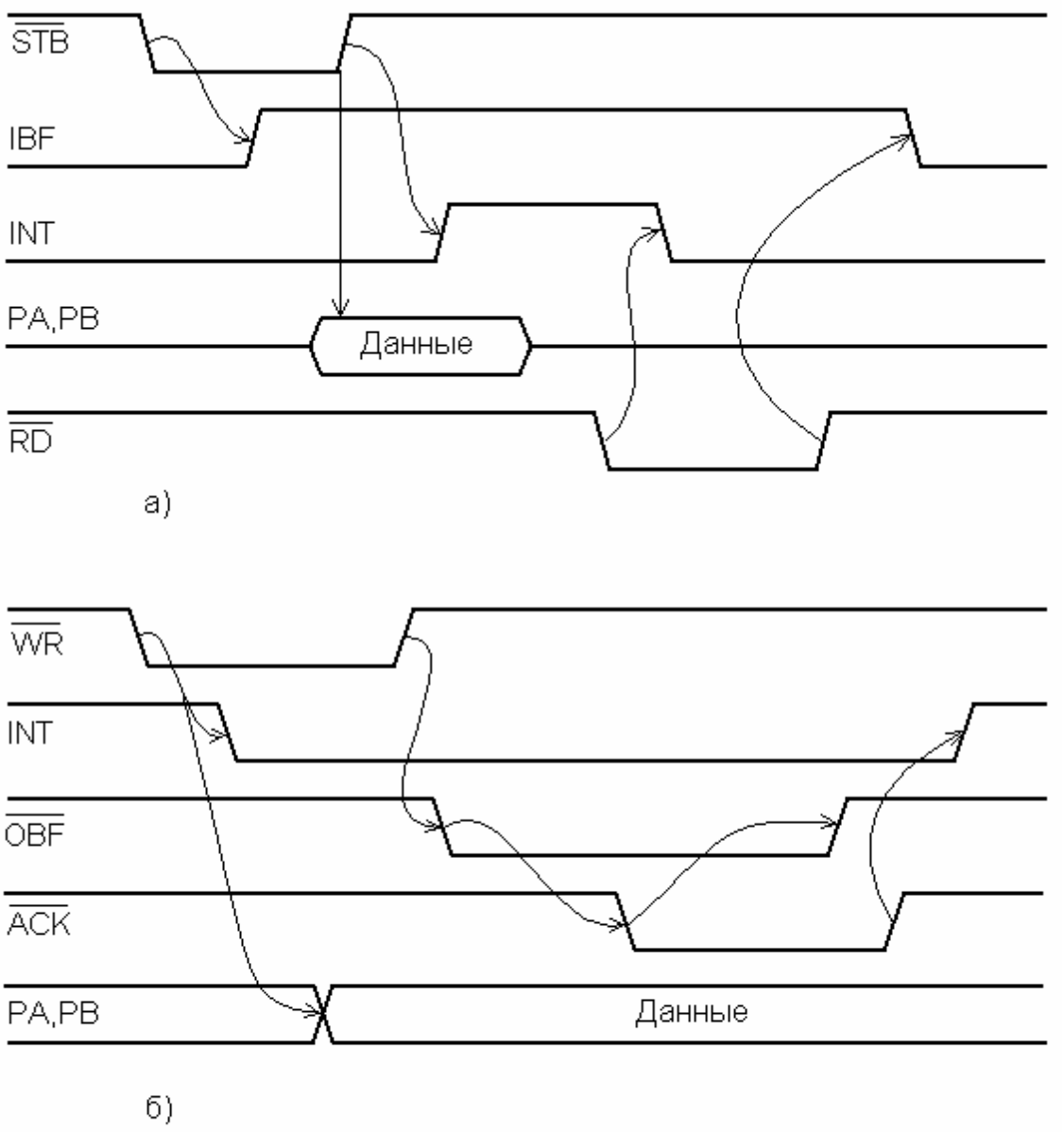


Рис.3.4. Временные диаграммы режима 1:
а - ввод; б - вывод

Для формирования вышеуказанных сигналов управления используются отдельные линии порта С в соответствии с рис. 3.5. Свободные от управления линии порта С могут использоваться в режиме 0.

Для управления обменом в режиме 1 со стороны МП предусмотрен программный доступ к линиям INT, IBF и OBF. Доступ организован через операцию чтения порта С. На рис. 3.5. показано, как при этом интерпретируются от-

дельные разряды введенных данных, называемые словом состояния **SW** адаптера, формат которого представлен на рис. 3.6.

В состав SW входят флажки разрешения прерываний **INTE**, управление состоянием которых может быть выполнено с помощью команды BSR. Генерация сигнала запроса на прерывание **INT** и установки связанного с ним одноименного флажка в SW возможна только при установленном флажке INTE. Запретить или разрешить работу устройства ВВ на микросхеме ВВ55 можно просто установив с помощью управляющего слова BSR 0 или 1 в соответствующий разряд порта С (см. рис. 3.5 и 3.6).

Порт А можно запрограммировать в режим двунаправленного ВВ, называемый также **режимом 2** (см. рис.3.7 и рис.3.8).

В режиме 2 линии PA7-PA0 исполняют роль двунаправленной трех- стабильной шины, управляемой сигналами STB, IBF, OBF, ACK и INT.

Сигналы IBF или OBF информируют ВУ о готовности принять или передать данные. В соответствии с состоянием IBF или OBF внешнее ПУ либо генерирует очередные данные, сопровождая их стробом STB, либо формирует сигнал подтверждения приема АСК, готовясь к приему данных. Низкий уровень сигнала АСК открывает выходные буферы порта А, разрешая выдачу данных на шину. В остальных случаях шина порта А находится в Z-состоянии.

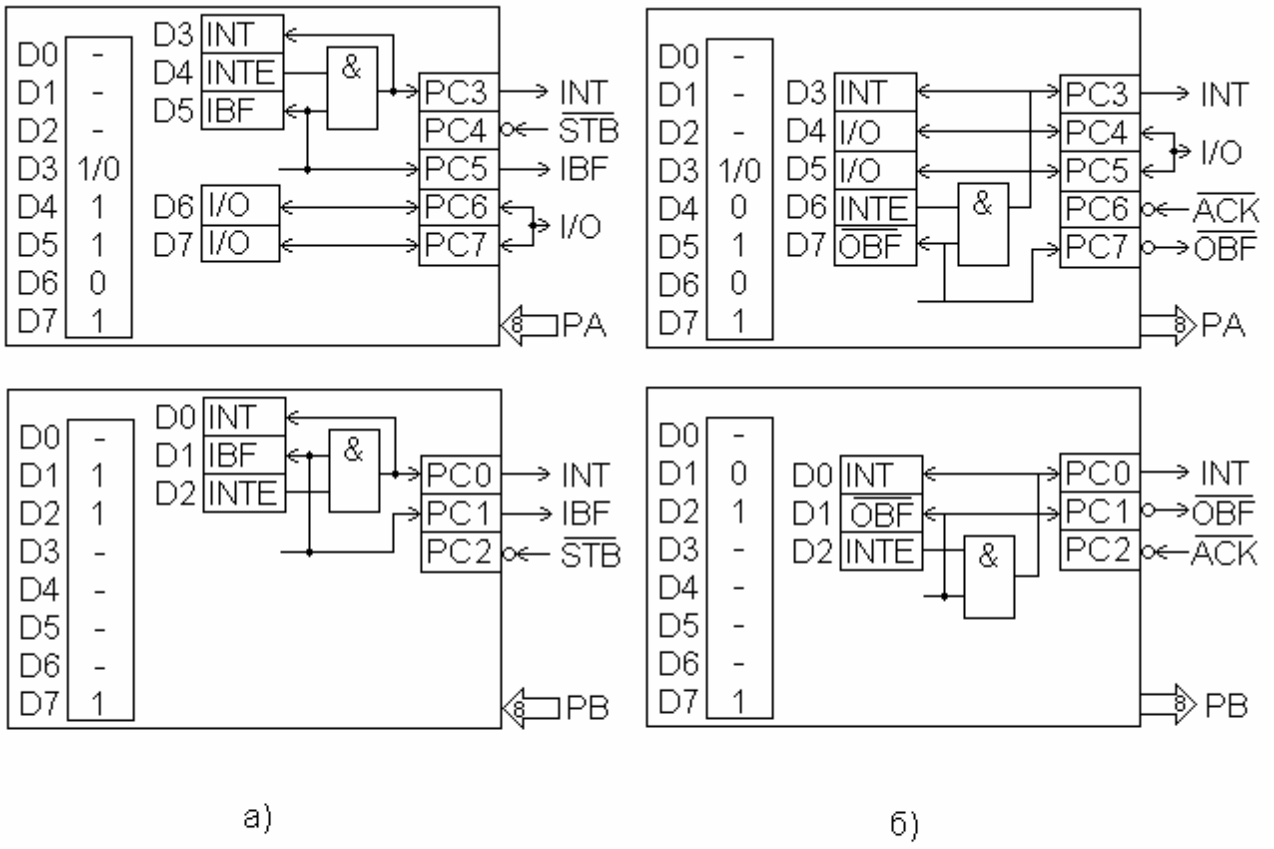


Рис.3.5. Организация однонаправленного обмена с квитированием:
а - ввод; б - вывод

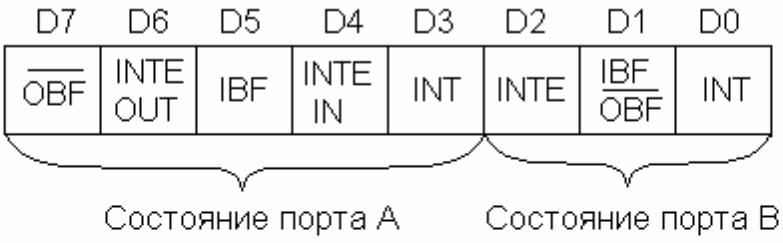


Рис.3.6. Формат слова состояния SW адаптера ВВ55

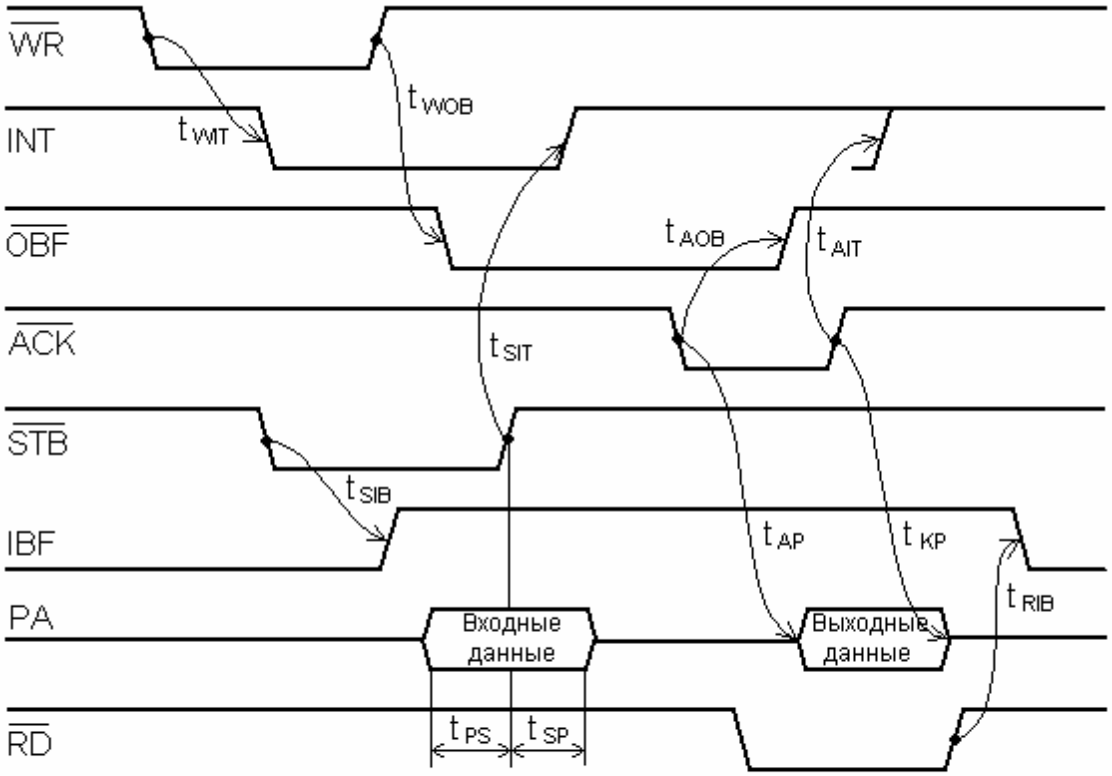


Рис.3.7. Временные диаграммы режима 2

Сигнал INT используется для организации ВВ по прерываниям. Логика его формирования пояснена на рис. 3.7 и 3.8. При получении очередного запроса на прерывание ЦП читает слово состояния SW (из порта С) и по флажкам IBF, OBF уточняет статус порта А, 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.

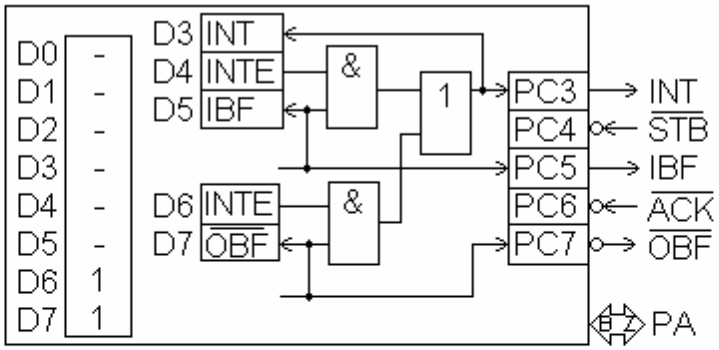


Рис.3.8. Организация двунаправленного ввода-вывода

Порты A, B, C для работы на указанные режимы программируются независимо друг от друга.

Пример подключения и программирования ППА.

Рассмотрим подключение микросхем ВВ55 в МП-систему, в которой она должна занимать адреса (см. табл. 3.1).

Таблица 3.1

Название регистра	Адрес	CS	Разряды ША							
			A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
Порт A	80h	0	1	0	0	0	0	0	0	0
Порт B	81h	0	1	0	0	0	0	0	0	1
Порт C	82h	0	1	0	0	0	0	0	1	0
Регистр CW	83h	0	1	0	0	0	0	0	1	1

$\overline{CS}$=00

Для обеспечения доступа к микросхеме по указанным адресам необходимо, чтобы сигнал ее выборки CS=0 формировался всегда, когда на линиях адреса A7 A6 A5 A4 A3 A2 находится код 100000.

На схеме подключения ППА в МПС рис. 3.9 приведены два варианта дешифраторов адреса, а также подключение к порту А печатающего устройства, к порту В – аналого-цифрового преобразователя.

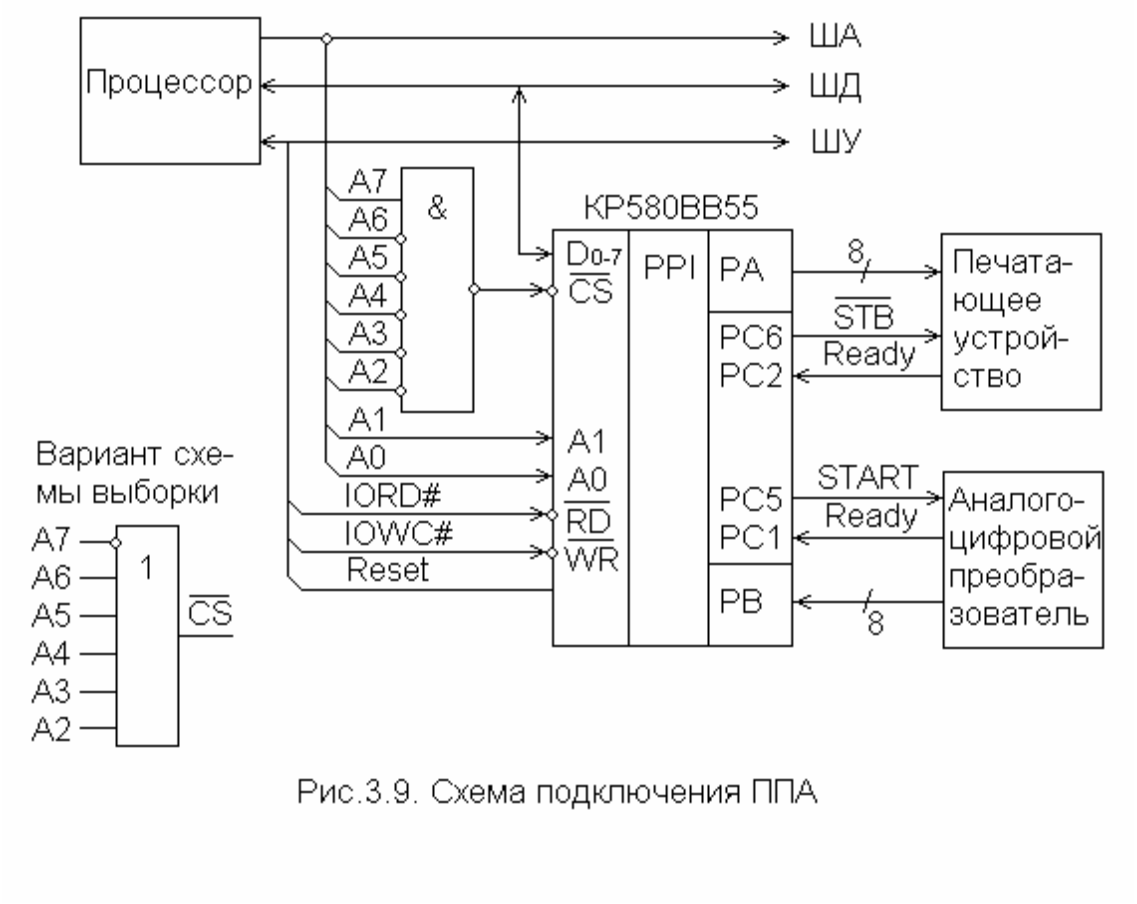


Рис.3.9. Схема подключения ППА

Рассмотрим программу управления печатающим устройством и аналого-цифровым преобразователем (см. рис. 3.9), использующую порты А, В и С в режимах 0.

MVI A, 83h; Запись управляющего слова SM.

OUT , 83h;

MVT A, 0Dh; Формирование сигнала STB=1, т.е. C6=1.

OUT , 83h;

MVI A, 0B; Формирование сигнала START, т.е. C5=1.

```
OUT , 83;
MVI A, 0A;        Сброс сигнала START, т.е. C5=0.
OUT , 83;
IN , port C;      Считывание SW, анализ готовности.
```

Далее анализ готовности принтера и АЦП.

Программа вывода на печать.

```
WAIT: IN, 82h;        Определение готовности принтера, т.е. C2=1.
      ANI, 04h;
      JZ, WAIT;       Ожидание готовности.
      MOV A, M;       Передача символа на печать.
      OUT, 80;
      MVI A, 0C;      Установка сигнала STB=0, т.е. C6=0.
      OUT, 83;
      INR A;          Установка STB =1.
      OUT, 83;
      RET
```

Задания

3.1. Разработайте подпрограмму ввода-вывода байта с АЦП через порт B в режиме 0 в соответствии со схемой подключения (см. рис. 3.9).

3.2. Разработайте схему подключения печатающего устройства и АЦП и программу управления обменом, используя режимы 1 портов A и B.

3.3. Определите достоинства режима 1 в сравнении с режимом 0 на основе анализа разработанных вами программ управления.

3.4. Перечислите и поясните функциональное назначение БИС серии К580.

3.5. Поясните режимы работы микросхемы ВВ55.

3.6. Составьте временные диаграммы обмена данными с квитированием.

3.7. Изобразите схему подключения ВВ55 к шине МПС.

3.8. Поясните использование SW при организации ввода-вывода данных.

3.2. Средства последовательного ввода-вывода

Параллельные каналы передачи данных применяются при расстоянии до 10 – 15м. При больших расстояниях стоимость многопроводного кабеля становится достаточно высокой, поэтому применяют каналы с последовательной (поразрядной) передачей данных. Известно много стандартов передачи данных в последовательных кодах**: RS-232C, RS-422, RS-485** и др.

Последовательная передача данных может быть в асинхронном или синхронном режимах. При **асинхронной** передаче (рис. 3.10) каждому передаваемому коду символа предшествует **старт-бит**, сигнализирующий приемнику о начале очередной посылки. Далее следуют **биты данных** (5 – 8) и возможно **бит паритете – P** (бит контроля четности).

Завершает посылку один или два **стоповых** бита, гарантирующих выдержку между соседними посылками. Старт-бит (всегда лог.0) обеспечивает простой механизм синхронизации приемника фронтом сигнала старт-бита. Внутренний генератор синхронизации приемника использует счетчик-делитель опорной частоты, обнуляемый в момент приема начала старт-бита. Этот счетчик формирует строб импульсы, по которым приемник фиксирует последующие принимаемые биты в середине их временных интервалов. Генераторы синхронизации передатчика и приемника, естественно, настраиваются на равные частоты (рис. 3.11). Чем выше частота передачи, тем больше влияние искажения фронтов на фазу принимаемого сигнала и тем больше погрешность привязки стробов к середине битового интервала.

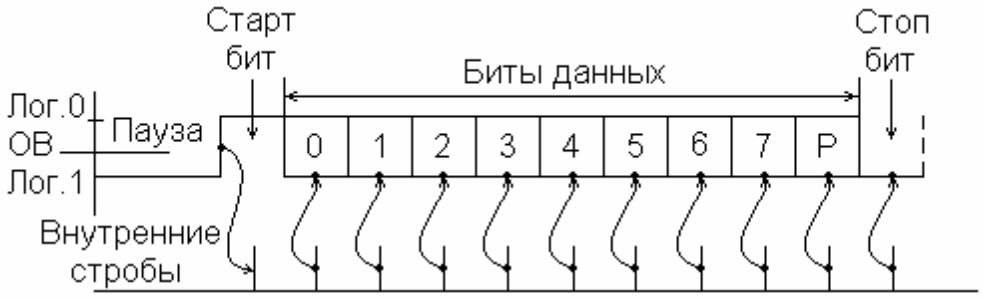


Рис.3.10. Формат асинхронной передачи

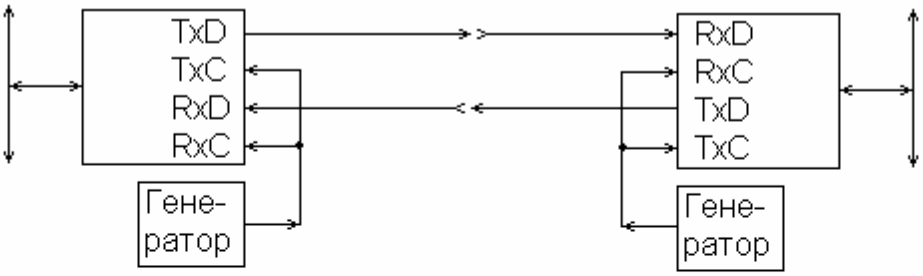


Рис.3.11. Схема дуплексной связи

Синхронный режим предполагает постоянную активность канала передачи данных. Посылка начинается с передачи кода синхросимвола (одного или двух). Далее коды символов передаются плотным потоком. Если у передатчика нет данных для передачи, то он заполняет паузу непрерывной посылкой кодов синхросимволов. В синхронном режиме необходима внешняя синхронизация приемника и передатчика, либо использование само-синхронизирующего кодирования, при котором на приемной стороне из принятого сигнала формируются импульсы синхронизации.

В качестве связных адаптеров применяются специальные микросхемы, например 8251A, 8250, 16450, 16550A (Intel). В качестве примера рассмотрим микросхему программируемого связного адаптера КР580ВВ51А, являющуюся аналогом микросхем 8251A.

Универсальный синхронно-асинхронный приемо-передатчик УСАПП (USART – Universal Synchronous/Asynchronous Resiver/Transmitter) КР580ВВ51 (далее ВВ51) представляет собой программируемую микросхему, реализующую интерфейс МПС с синхронно-асинхронными каналами последовательной связи. В состав ВВ51 (рис. 3.12) входят передатчик, приемник, буфер шины данных и схемы управления передатчиком, приемником, модемом.

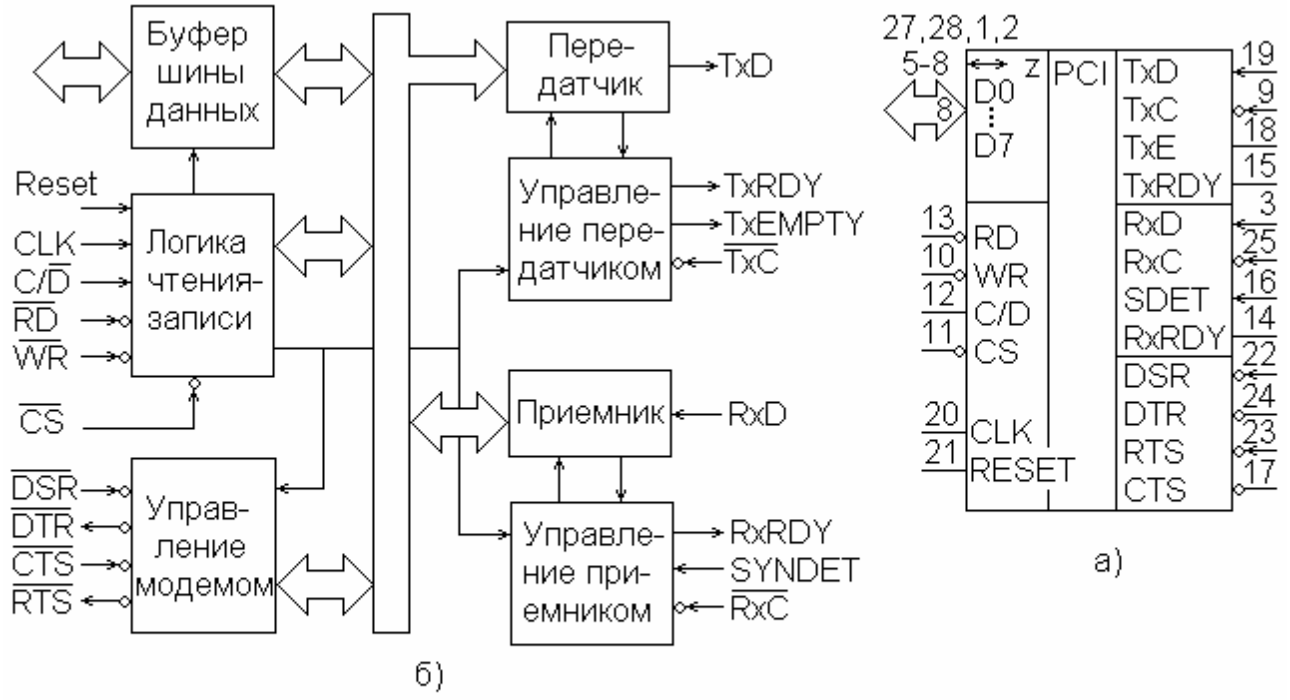


Рис.3.12. Структурная схема ВВ51

Основу передатчика составляет 13-разрядный сдвиговый регистр, хранящий очередной код передаваемого символа. Разряды 12 и 11 регистра используются для формирования стоп-битов, 10 – для записи контрольного бита, разряды 9 – 2 – для хранения данных, 1 – для формирования старт-битов, последний разряд 0 используется как выходной буфер. Схема управления передатчиком отслеживает прием новых данных, добавляет к ним контрольный бит, старт-бит, стоповый бит и синхронизирует вывод из регистра сдвига в линию TxD (Transmitter Data).

Таблица 3.2

Операция и вид информации	Управляющие сигналы				
	C/D	W/R	RD	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

Приемник состоит из входного формирователя, двух 9-разрядных регистров сдвига, двух регистров для хранения синхросимволов, схемы управления и схемы синхронизации. Информация со входа RxD (Resiver Data) последовательно поступает через входной буфер и далее в регистры сдвига. Управление записью входной информации осуществляется схемой управления приемником, 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).

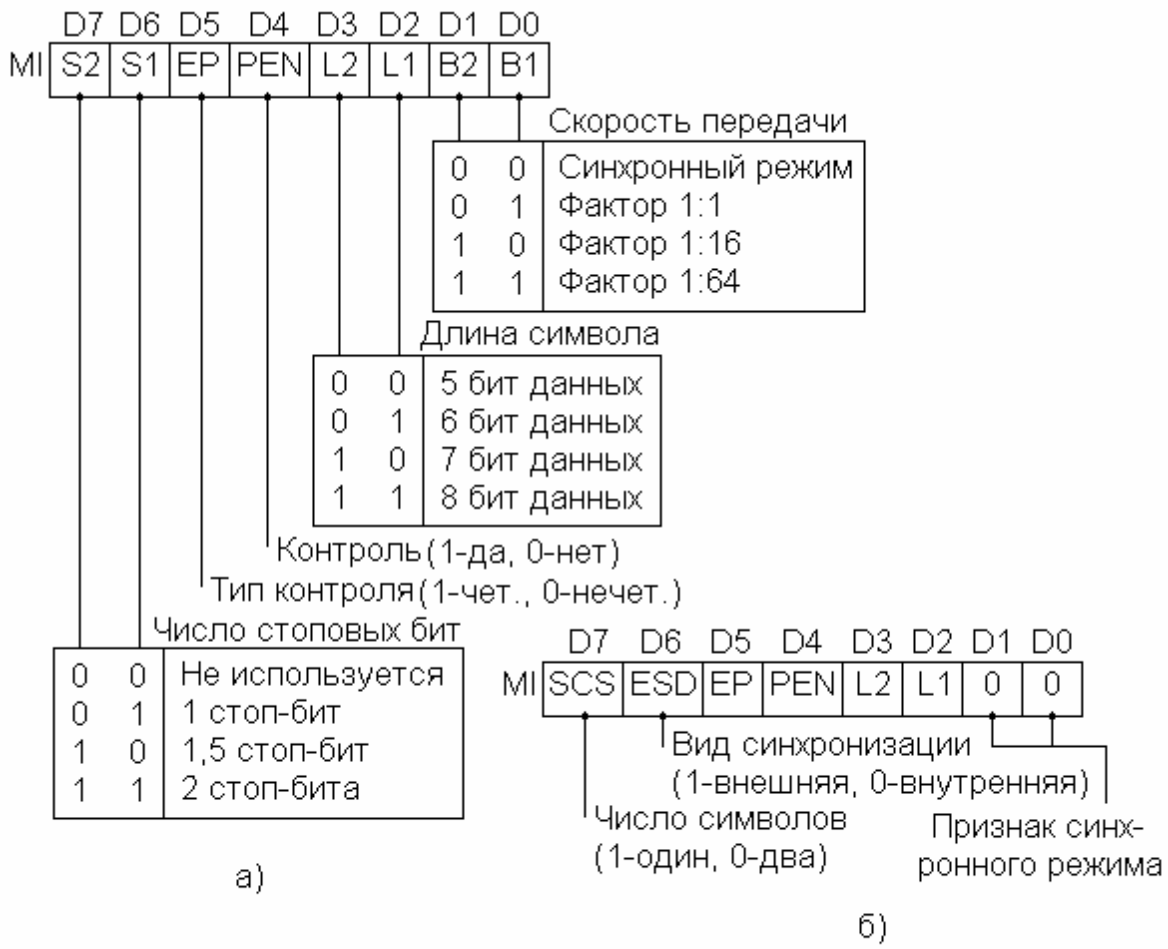


Рис.3.13. Формат инструкции режима MI
а - асинхронный режим, б - синхронный режим

Буфер шины данных представляет собой 8-разрядный трехстабильный двунаправленный буфер, обеспечивает связь УСАПП с МПС. Логика чтения-записи содержит регистр режима и регистр команд. Регистр режима служит для хранения управляющего слова выбора режима **MI**, а регистр команд – для приема командной инструкции **CI.**

Схема управления модемом служит для обмена с ПУ управляющими сигналами:

DSR (Data Set Ready). Готовность приемника модема проверяется программно.

DTR (Data Terminal Ready). Запрос готовности приемника модема. Управляется программно.

RTS (Reguest To Send). Запрос готовности передатчика модема.

Управляется программно.

CTS (Clear To Send). Готовность передатчика модема. Разрешает УСАПП передачу данных.

Входные сигналы WR и RD определяют направление потока информации, передаваемой шине данных из ЦП в УСАПП и обратно. Логический уровень на входе C/D=1 – запись или считывание управляющих слов, а при C/D=0 – запись или считывание данных. Все операции по обмену информацией возможны только при CS=0, т. е. когда микросхема выбрана.

Перед началом работы УСАПП необходимо установить в исходное состояние системным сигналом **Reset** либо программным способом с помощью команды **IR=1** (бит D6 в командной инструкции CI). Дальнейшее управление работой УСАПП осуществляется двумя управляющими словами: **MI (Mode Instruction) и CI (Command Instruction)** в соответствии с табл. 3.2. Управляющее слово MI определяет режим работы УСАПП и должно быть передано сразу после операции сброса. Имеется два формата MI (рис. 3.13), позволяющие установку асинхронного и синхронного режимов. В обоих режимах есть возможность программировать длину слова данных (поле L2 –L1) и тип контроля (разряды EP и PEN). Для асинхронного режима программируются также число стоп-бит (поле S2 – S1) и скорость передачи (поле B2 – B1). В синхронном режиме следом за MI должны быть закруглены один или два синхросимвола.

Слово CI (рис. 3.14) используется для оперативного управления работой УСАПП: разрешения/запрещения приема/передачи (перехода в режим ожидания синхронизации и установки микросхем в исходное состояние с целью ее переинициализации).

Для организации программно-управляемого обмена по условию можно использовать слово состояния **SW** (рис.3.15), в состав которого входят флажки готовности передатчика **TxRDY** и приемника **RxRDY**. После выдачи слова данных флажок **TxRDY** устанавливается в 0, а после записи в буфер передатчика данных DW снова устанавливается в 1. Аналогично устанавливается флажок **RxRDY** при заполнении и считывании буфера принятых данных.

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

УСАПП может работать в следующих режимах: асинхронная передача, асинхронный прием, синхронная передача, синхронный прием с внутренней или внешней синхронизацией.

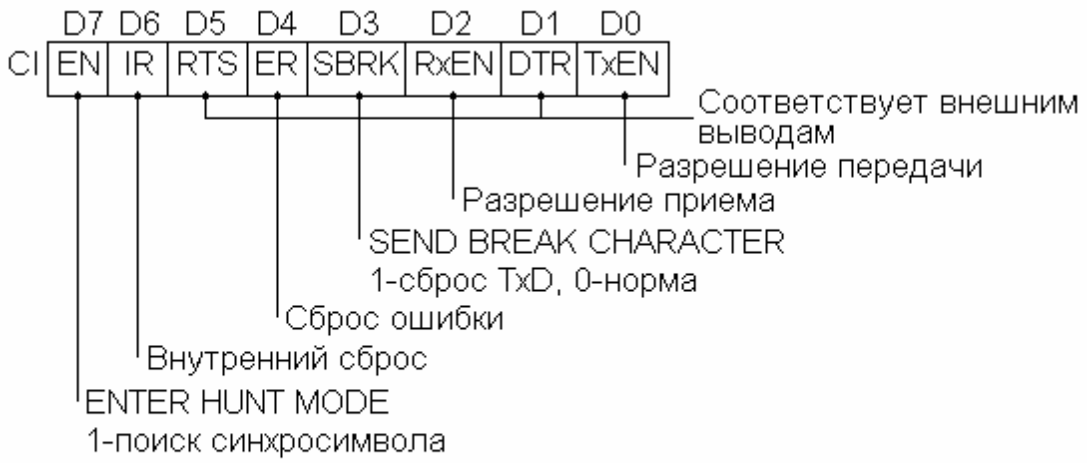


Рис.3.14. Формат командной инструкции CI

Если при приеме обнаруживается ошибка четности, то устанавливается флажок PE, а если при приеме первого такта стоп-бита на линии RxD окажется напряжение низкого уровня, то зафиксируется ошибка формата. Если предыдущий принятый символ из регистра еще не считан, то он теряется и устанавливается флажок переполнения OE.

Рис.3.15. Формат слова состояния SW

После передачи в УСАПП слова данных и при CTS#=0 передатчик начинает выталкивать биты данных со скоростью следования импульсов TxC# (рис. 3.16).

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 выводе **TxEMPTY** формируется импульс на каждый синхросимвол, указывающий на передачу последнего бита синхросимвола.

Информация на входе приемника RxD принимается по фронту сигнала RxC# и в синхронном режиме непрерывно сравнивается с синхросимволами (сначала с первым, а потом и со вторым). При обнаружении синхросимволов во время приема последнего бита на линии **SYNDET** устанавливается напряжение высокого уровня, что означает вхождение в синхронизацию.

При сбросе и чтении SW на линии SYNDET устанавливается низкий уровень напряжения. При внешней синхронизации SYNDET является входом синхросигналов.

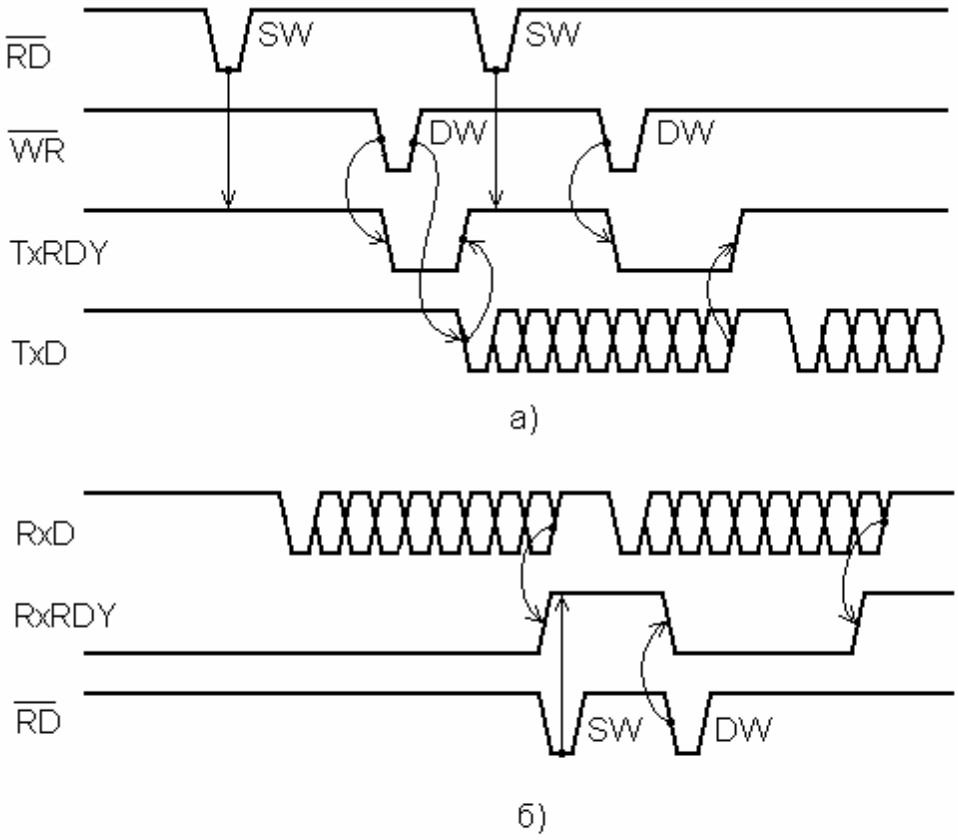


Рис.3.16. Временные диаграммы передачи (а)
и приема (б) в асинхронном режиме

В состав CI входят флажки, управляющие выходами **DTR# и RTS#.** Аналогично **SW** содержит ряд флажков, отражающих текущее состояние физических линий **DSR#, SYNDET, TxEMPTY, RxRDY, TxRDY**. Такое дублирование позволяет реализовать обмен данными с ПУ как по методу программного обмена, так и по прерываниям.

На рис. 3.16 приведены временные диаграммы передачи и приема данных в асинхронном режиме.

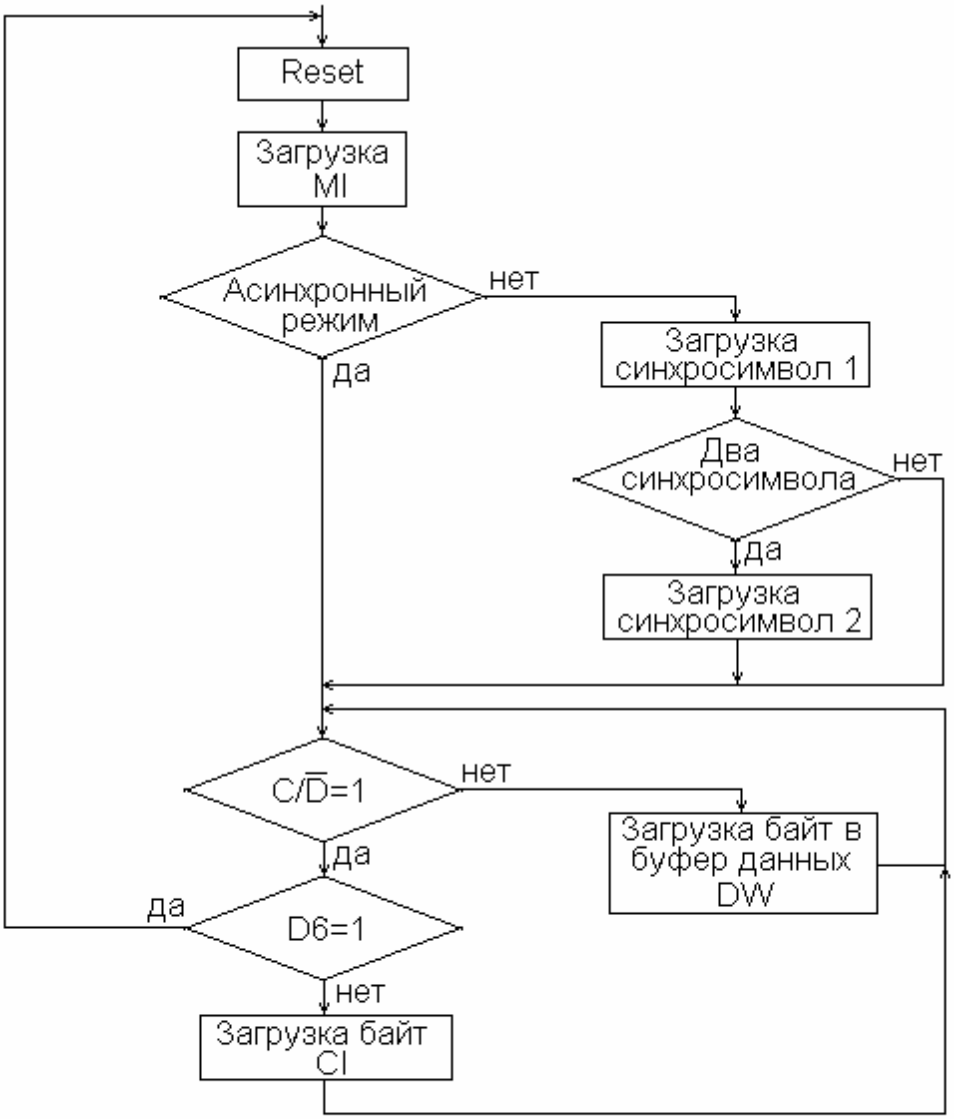


Рис.3.17. Последовательность инициализации ВВ51

Установка флажка готовности TxRDY в 1 сигнализирует МП о готовности передатчика принять данные (рис. 3.16,а). После выдачи данных флажок готовности TxRDY устанавливается в 0 и снова в 1 после передачи байта данных в выходной буфер передатчика. Если буфер еще занят выводом предыдущего байта данных, то происходит ожидание его освобождения.

Принятые данные передаются в выходной регистр, и устанавливается флажок готовности RxRDY (рис. 3.16, б), сигнализирующий МП о возможности их считывания. Флажок RxRDY устанавливается в 0 при считывании МП данных DW из буфера.

Последовательность инициализации УСАПП приведена на рис. 3.17.

Рассмотрим пример инициализации микросхемы ВВ51. Пусть необходимо установить: асинхронный режим, 7 бит в поле данных, контроль четности, 2 стоповых бита, фактор скорости 1/16. Микросхема имеет адреса в МПС: 30h и 31h.

```
XRA;          Обнуление аккумулятора.
OUT; 31h;     Перевод УСАПП в состояние
OUT; 31h;     реагирования на команду
OUT; 31h;     Reset.
MVI; A, 40h;  Формирование программного
OUT 31h;      Reset.
MVI A, FAh;   Загрузка управляющего слова
OUT , 31h;    режима MI
MVI A, 05h;   Разрешение ввода и вывода
OUT , 31h;
RET.
```

Применение.

Программируемые адаптеры связи применяются при организации последовательных каналов передачи данных. На рис. 3.18 приведен вариант организации каналов ввода и вывода данных на К580ВВ51.

Сигналы готовности передатчика и приемника TxRDY и RxRDY используются для вызова программ обмена.

Выходные сигналы передатчика TxD преобразуются на операционном усилителе ОУ в двухполярные стандартные сигналы интерфейса RS232C. А двухполярные сигналы из линии связи на аналоговом компараторе СА преобразуются в однополярные и поступают на вход приемника RxD.

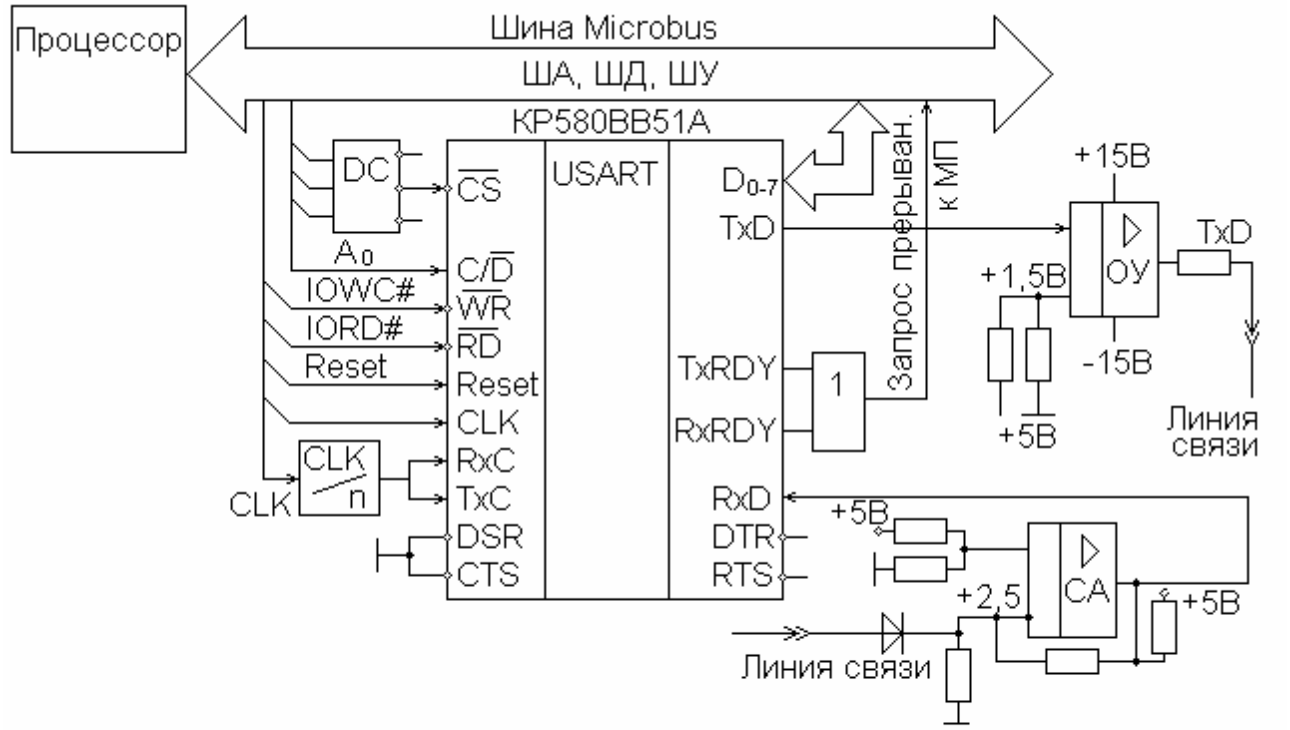


Рис.3.18. Схема подключения ВВ51А

Вопросы и задания

3.9. Разработайте подпрограммы ввода и вывода, в соответствии со схемой подключения (см. рис. 3.18).

3.10. Разработайте схему дешифратора, обеспечивающую микросхеме ВВ51 в МПС адреса 30h и 31h.

3.11. Поясните синхронизацию приема данных для асинхронных и синхронных режимов.

3.12. Почему с увеличением скорости передачи повышается вероятность сбоев принимаемых данных?

3.13. Поясните доступность при программировании следующих структур микросхемы ВВ51: регистров управляющих слов MI, CI, SW буферов передатчика и приемника.

3.14. Составьте структурную схему УСАПП и поясните режимы работы.

3.15. Изобразите схему подключения ВВ51 к микропроцессорной шине.

3.16. Поясните последовательность операций инициализации УСАПП.

3.3. Подсистемы прерываний

Обработка информации по прерываниям позволяет реализовывать эффективное взаимодействие с медленно действующими устройствами, а также оперативно откликаться на реальные события по запросам от ПУ.

Физический интерфейс наиболее простой системы прерываний является единственной линией IRQ (Interrupt Request), сигнал с которой воспринимается как запрос на прерывание. При приеме запроса на прерывание ЦП активизирует программу обслуживания прерывания, передавая ее начальный адрес в PC. Старое содержимое PC, являющееся адресом возврата в прерванную программу, автоматически сохраняется в стеке. Возврат в прерванную программу осуществляется последней командой в программе обслуживания прерываний (командой RET), которая адрес возврата из стека возвращает в PC. Начальный адрес программы обслуживания прерываний, загружаемый в PC, определяется МП. Такие входы запросов прерываний имеются, например, у микропроцессора К1821ВМ85 (TRAP, RST 7.5, RST 6.5, RST 5.5).

Вход INT микропроцессоров (ВМ80, ВМ85, Z80 и др.) называют маскируемым, что означает возможность разрешения или запрещения обслуживания запроса, поступившего на этот вход. Формирование начального адреса подпрограммы обработки прерываний требует получения от подсистемы прерывания кодов команды CALL или команды RSTn (для МП ВМ80, ВМ85, Z80).

В МПС на микропроцессорах платформы x86 стартовый адрес процедуры обслуживания прерываний определяется обращением к таблице прерываний через однобайтный вектор, формируемый аппаратными средствами подсистемы прерываний.

В зависимости от числа подтвержденных запросов прерывания, одновременно находящихся на обслуживании, различают одноуровневые и многоуровневые системы прерываний. Примером одноуровневой системы с двумя источниками запросов является система прерываний микроконтроллера К1816ВЕ48, а двухуровневой с пятью источниками запросов – К1816ВЕ51.

Для создания приоритетных многоуровневых систем прерываний применяются программируемые контроллеры прерываний К580ВН59 и К1810ВН59А. Микросхема ВН59 является аналогом микросхемы 8259 (Intel), используется в МПС на 8-разрядных МП (ВМ80, ВМ85, ВМ1 и др.), а ВН59А – в МПС и на 8-разрядных, и на 16-разрядных микропроцессорах платформы x86.

Программируемый контроллер прерываний КР580ВН59.

Программируемый контроллер прерываний предназначен для сбора и фиксации прерываний от восьми ВУ. Предусмотрена возможность изменения приоритета ВУ в процессе выполнения программы, программного маскирования на запрос для любого ВУ, расширения количества ВУ до 64 путем каскадного соединения одного ведущего с ведомыми (8 штук).

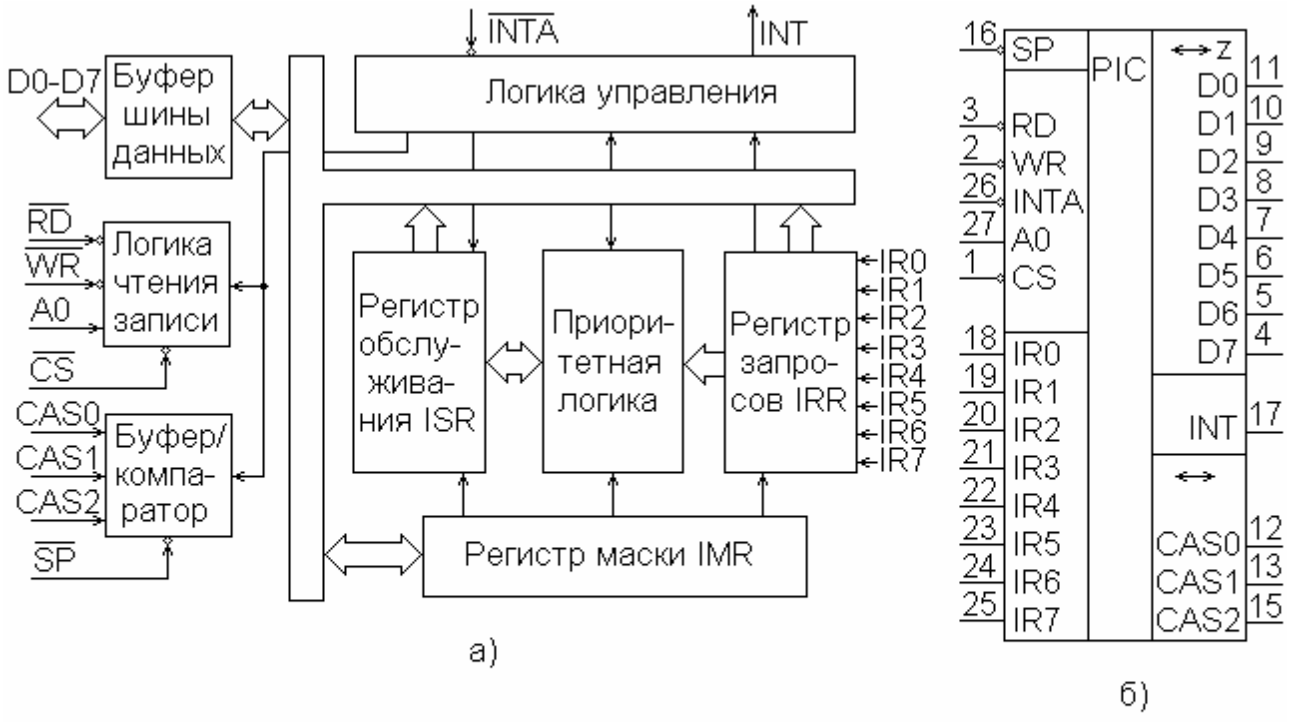


Рис.3.19. Программируемый контроллер прерываний ВН59
а) структурная схема
б) условное графическое обозначение

Функции контроллера

1. Фиксация запросов на прерывание.
2. Присвоение ВУ приоритетов с возможностью их изменения в процессе выполнения программы.
3. Независимое маскирование запросов на прерывание от ВУ.
4. Анализ приоритетов вновь поступивших и уже обслуживаемых запросов на прерывание.
5. Формирование запросов на прерывание (сигналов INT).
6. Формирование кода команды CALL и 8 адресов перехода.
7. Организация взаимодействия ведущего и ведомого ПКП.
8. Выдача на системную шину данных по запросам микропроцессора содержимого ряда регистров, характеризующих состояние подсистемы прерываний.

Рассмотрим работу ВН59 в автономном режиме (рис. 3.19). Запросы на прерывание от ВУ подаются на входы IRO-IR7 и запоминаются в регистре запросов IRR (Interrupt Request Reqister) (см. рис. 3.19). Регистр обслуживания ISR (In Service Reqister) содержит все запросы, которые в данное время находятся в стадии обслуживания. Регистр маски IMR (Interrupt Mask Reqister) используется для маскирования отдельных запросов. Запрет некоторого запроса осуществляется установкой 1 в соответствующем регистре IMR.

Приоритетная логика выбирает запрос на прерывание с наивысшим приоритетом из числа поступивших и сравнивает его с текущим приоритетом запросов, находящихся на обслуживании. Если приоритет выделенного из поступивших превышает приоритет текущего, то ВН59 генерирует сигнал INT к МП. Микропроцессор подтверждает прием запроса INT генерацией сигнала INTA#, под воздействием которого запрос с высшим приоритетом из IRR фиксируется в соответствующем разряде ISR. Принятый к обслуживанию сбрасывается в регистре IRR и прием нового запроса разрешается. Одновременно с этим ВН59 генерирует код команды CALL (0CDh), который по ШД принимается МП. В ответ ВМ80/ВМ85 инициируют еще два следующих друг за другом INTA-цикла, в которых ВН59 передает в МП полный адрес программы обслуживания

прерывания, принятого к обработке. Сначала передается младший байт, а затем старший байт адреса. Установленный в ISR бит остается в состоянии 1 до окончания процедуры обслуживания. В конце процедуры специальной командой окончания прерывания соответствующий разряд регистра ISR сбрасывается.

До тех пор, пока некоторый ISR-разряд установлен, все запросы с равным или меньшим приоритетом игнорируются, а запросы с более высоким приоритетом обслуживаются, инициируя вложенные прерывания.

Контроллер прерываний ВН59 может быть настроен на один из четырех режимов определения приоритетных запросов.

1. Режим фиксированных приоритетов. Входу IRQ присваивается наивысший приоритет, а приоритеты других входов убывают по мере возрастания их номера.
2. Векторные прерывания с циклическим перераспределением приоритетов. После каждого обслуженного прерывания вся система приоритетов изменяется по кругу, то есть последний обслуженный код имеет наименьший приоритет.
3. Векторные прерывания с адресуемыми распределениями приоритетов. В этом режиме номер входа, который имеет наивысший приоритет, указывается программно. Приоритеты остальных устройств распределяются по кругу.
4. Прерывание по результату опроса. Инициатива по обслуживанию прерывания исходит не от ВУ, а от микропроцессора. С помощью оперативного управляющего слова OCW3 МП устанавливается режим поллинга и осуществляется выбор регистров IRR или ISR для чтения содержимого. Контроллер ВН59 в цикле чтения при A0=0 выдает на шину данных код запроса с наивысшим приоритетом.

Программирование ВН59 осуществляется двумя типами управляющих слов: ICW (Initialization Command Word) и OCW (Operation Command Word). На рис. 3.20 представлена последовательность инициализации. Три управляющих слова ICW1-ICW3 загружаются перед началом работы и устанавливают БИС в режим фиксированных приоритетов. Для оперативного управления работой контроллера в любое время в него могут быть загружены команды управления OCW1-OCW3.

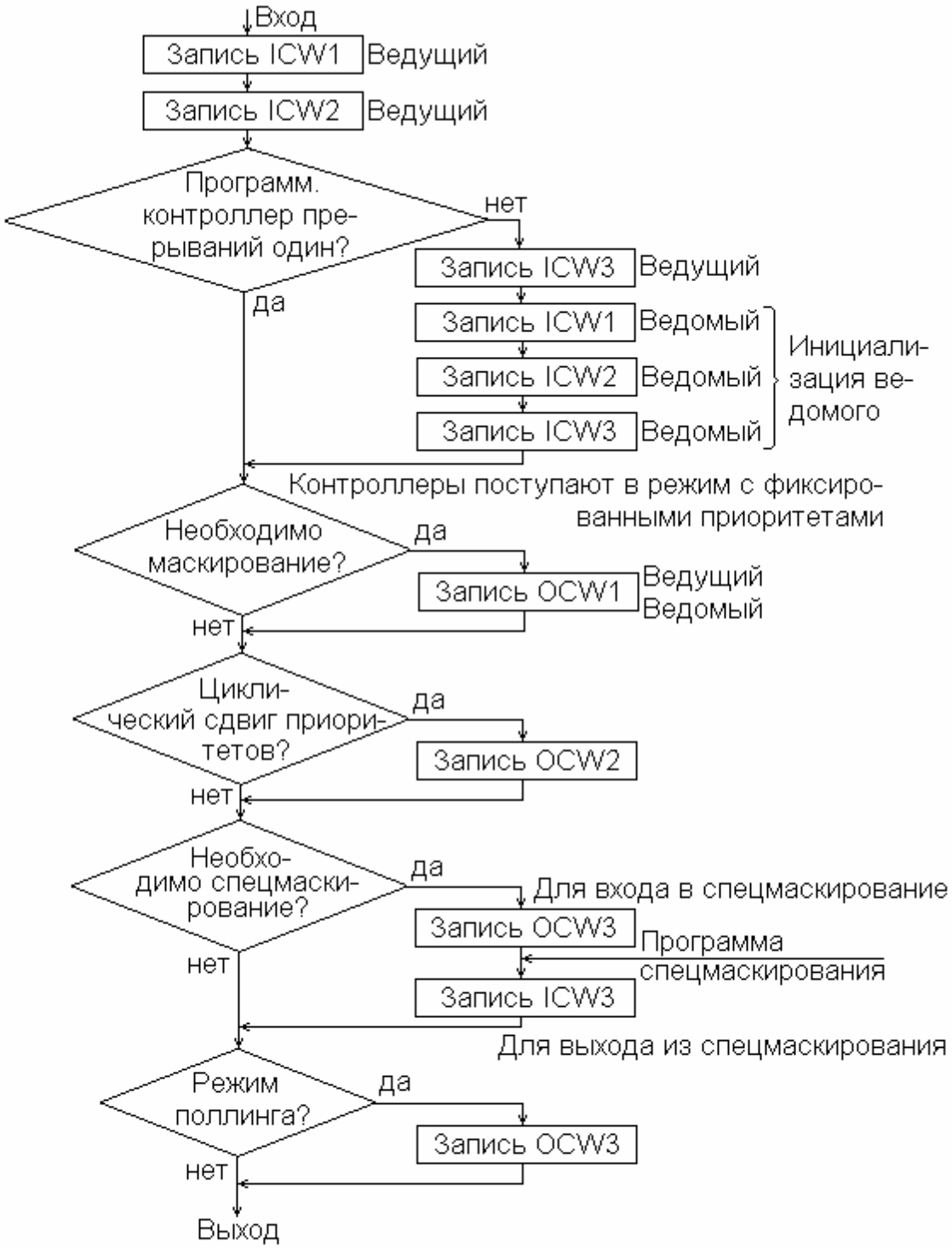


Рис.3.20. Последовательность инициализации и оперативного управления режимами

Два первых обязательных ICW1 и ICW2 определяют базовый адрес таблицы входов (рис. 3.21). Адресный интервал 4 или 8 задается битом F в ICW1. Начальные адреса подпрограмм обслуживания формируются по схеме:

addr = base + k × N, где k=4 или 8, N – номер запроса, принятого к обслуживанию.

В составе ICW1 бит S определяет отсутствие или наличие каскадирования ВН59. В каскадном варианте загружается еще команда ICW3 в ведущий и ведомый. ICW3 для ведущего содержит 1 в разрядах, соответствующих входам запросов, к которым подключены выходы INT ведомых. ICW3 для ведомого содержит код номера входа ведущего, к которому он подключен.

Рис.3.21. Форматы ICW1-ICW3

Рис.3.22. Форматы OCW1-OCW3

После инициализации ВН59 готов к работе в режиме фиксированных приоритетов. Дальнейшее управление работой схемы осуществляется с помощью команд OCW1 – OCW3, форматы которых приведены на рис. 3.22.

В произвольный момент времени и независимо от других каждый запрос командой OCW1 может быть замаскирован. При установке 1 в разряде IMR запрещается прием прерываний по данному входу.

Команда OCW2 служит для установки в 0 произвольных разрядов ISR и циклического сдвига приоритетов с присвоением максимального значения любому из восьми возможных уровней. Для обслуживания запросов с равными приоритетами используется циклический сдвиг (команда EOI). При использовании команды сдвига одновременно со сбросом ISR-бита, имеющего высший приоритет, реализуется циклический сдвиг приоритетов с присвоением низшего только что обслуженному уровню. Циклический сдвиг не нарушает последовательности вложенных друг в друга прерываний, что обеспечивает правильный возврат из обслуживающих их программ. Прямая адресация уровня в слове OCW2 позволяет сбросить конкретный ISR-бит и таким образом завершить

процедуру обслуживания этого запроса и циклическое изменение приоритетов с явным указанием нижнего уровня.

С помощью OCW3 устанавливается режим спецмаскирования (см. рис. 3.22). В этом режиме каждый бит в регистре ISR запрещает только собственный уровень, но разрешает все остальные.

Рис.3.23. Формат данных при опросе

Командой OCW3 осуществляется управление режимом поллинга и выбор регистра IRR или ISR для чтения его содержимого с помощью программ. Режим инициируется выдачей в ВН59 слова OCW3 с установкой бита P (Polling). Контроллер определяет следующий цикл чтения при A0=0 как подтверждение прерывания и выдает на шину данных OCW3. По этому слову определяется запрос с наивысшим приоритетом (рис. 3.23).

Если количество запросов больше 8, то применяются схемы каскадирования контроллеров ВН59 и ВН59А (8259 и 8259А). Один из контроллеров получает статус ведущего (выход SP подключается через резистор к источнику 5В), а остальные – ведомых (выходы SP заземляются). Выходные линии INT с ведомых подсоединяются ко входам запросов на прерывание ведущего (рис. 3.24).

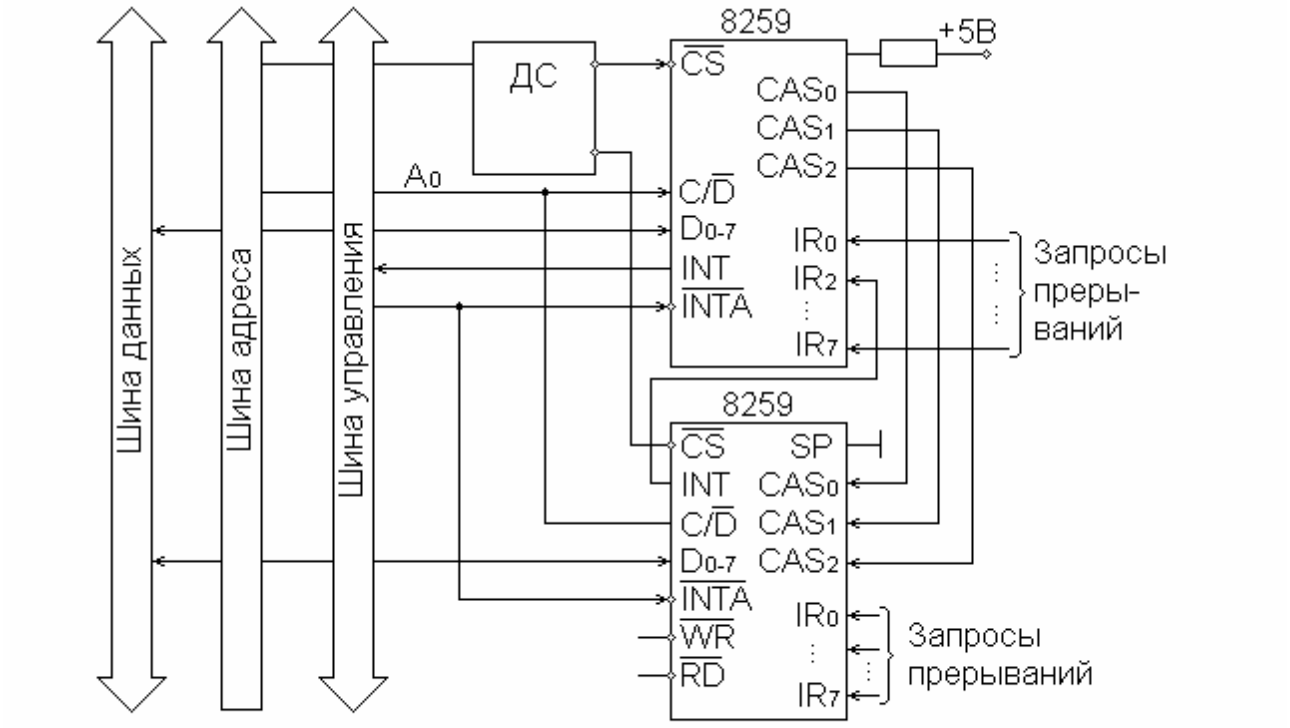


Рис.3.24. Схема каскадирования и подключения

Если ведущий ВН59, обрабатывая запросы, принимает на обслуживание запрос, поступивший с ведомого, то после получения первого импульса INTA ведущий выставляет на ШД первый байт команды CALL. А на шину каскадирования CAS0 – CAS2 ведущий сбрасывает код ведомого, запрос с которого принят к обслуживанию. Ведомый, код которого соответствует коду на линиях каскадирования, во втором и третьем циклах INTA выставляет на ШД младший и старший байты начального адреса подпрограммы обслуживания прерывания.
Пример инициализации ВН59.

Подсистема прерываний (см. рис. 3.24) выполнена на двух микросхемах К580ВН59 (или 8259). Ведомый контроллер подключен к входу IR2 ведущего. Пусть базовый адрес в МПС ведущего контроллера 20h, а ведомого 22h.

Таблица адресации подпрограмм обслуживания прерываний для ведущего начинается с базового адреса 4000h, т.е. для подпрограммы обслуживания прерываний по входу IRQ ведущего. Адреса подпрограмм обслуживания остальных запросов формируются через 4 байта. Базовый адрес подпрограммы

обслуживания запросов на ведомый равен 4040h, и далее адреса формируются через 8 байт.

```
MVI A, 14h;
OUT , 20h;    Запись ICW1 в ведущий.
MVI A,40h;
OUT , 21h;  Запись ICW2 в ведущий, т. е. старший байт адреса.
 MVI A, 04h;
OUT , 21h;   Запись ICW3 в ведущий.
MVI A, 54h;
OUT , 22h;    Запись ICW1 в ведомый.
MVI A, 40h;
OUT , 23h;   Запись ICW2 в ведомый, т. е. старшего байта адреса.
MVI A, 02h;
OUT , 23h;    Запись ICW3 в ведомый.
RET.
```

Вопросы и задания

3.17. Разработайте схему дешифратора для адресации ВН59 контроллеров (см. рис. 3.24).

3.18. Определите начальный адрес подпрограммы обслуживания прерывания по входу IR6, ведомого для схемы прерывания (см. рис. 3.24).

3.19. Разработайте подсистему прерываний для обслуживания запросов от 22 ПУ.

3.20. Поясните, какие средства имеют МП для организации обмена по запросам прерывания (на примере МП ВМ80, ВМ85, ВМ86).

3.21. Какова организация радиальной системы прерываний?

3.22. Поясните достоинства и недостатки приема запросов на прерывания в виде уровней напряжений (высокого/низкого) и фронтами сигнала.

3.23. Что понимается под одноуровневыми и многоуровневыми прерываниями?

3.24. Какие способы формирования начального адреса подпрограмм для обслуживания прерываний вам известны?

3.25. Приведите примеры векторной 8-уровневой системы прерывания.

3.26. Назовите функции программируемого контроллера ВН59.

3.27. Назовите управляющие слова инициализации ВН59 и их назначение.

3.28. Какие функции выполняют операционные управляющие слова OCW1–OCW3?

3.29. Изобразите по памяти структурную схему контроллера ВН59 и поясните назначение ее компонентов.

3.30. Назовите режимы обработки запросов по приоритетам контроллера ВН59.

3.4. Программируемый таймер К1810ВИ54

Программируемый таймер (ПТ) К1810ВИ54 предназначен для генерации времязадающих функций, программно-управляемых временных задержек с возможностью программного контроля их выполнения. Программируемые таймеры применяются в МПС, выполненных на базе МПК БИС К580, К1810, К1821, используемых в задачах управления и измерения в реальном масштабе времени с тактовой частотой до 8 Мгц. Конструктивно эти ПТ совместимы с ПТ типа К580ВИ53, отличаются от них повышенным быстродействием и расширенными функциональными возможностями.

Программируемый таймер К1810ВИ54 содержит три независимых канала, каждый из которых может быть запрограммирован на работу в одном из шести режимов для двоичного или двоично-десятичного счета. Структурная схема ПТ показана на рис. 3.25.

Назначение выводов.

CS# – выборка кристалла. Сигнал управляет входным буфером BD. При CS#=0 разрешается работа буфера.

RD# – чтение. Сигнал RD#=0 ориентирует входной буфер BD на вывод. ПТ выдает информацию в ЦП.

WR# – запись. Сигнал WR#=0 ориентирует входной буфер BD на ввод. ПТ принимает информацию от ЦП.

A0, A1 – адресные входы, по которым осуществляется адресация к одному из каналов:

A0=A1=00 – адрес канала 0;

A0=A1=01 – адрес канала 1;

A0=A1=10 – адрес канала 2;

A0=A1=11 – признак загрузки управляющего слова или команд.

CLK2 – CLK0 – входы тактовых сигналов для управления счетчиком/таймером. Срез сигнала на входе CLK приводит к уменьшению содержимого счетчика/таймера CE на единицу.

GATE2 – GATE0 – входы разрешения счета. При GATE=1 разрешается выполнение функций; для некоторых режимов работы разрешается поступление тактовых сигналов на вход счетчика/таймера, для других (импульсного генератора и генератора меандра) открывается выходной буфер OUT.

OUT2 – OUT0 – выходы счетчика таймера.

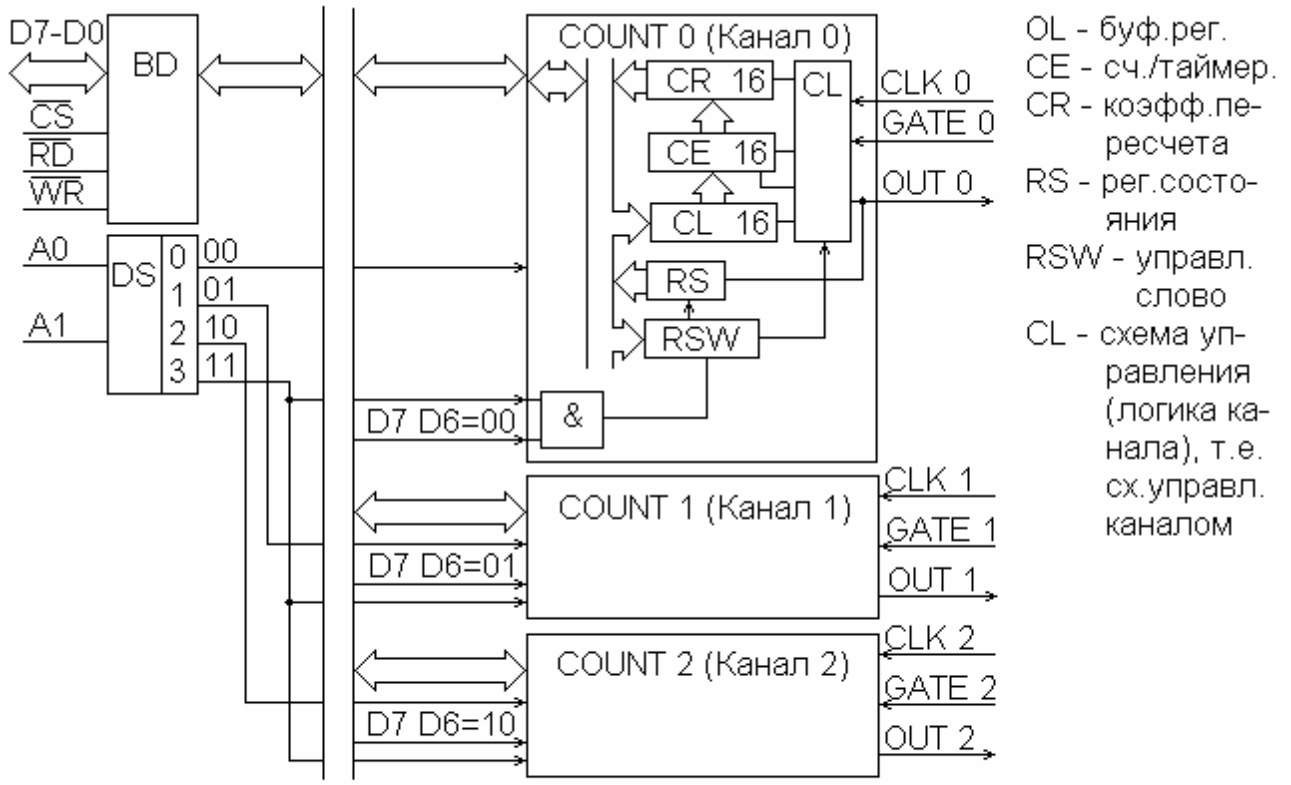


Рис.3.25. Структурная схема программируемого таймера ВИ54

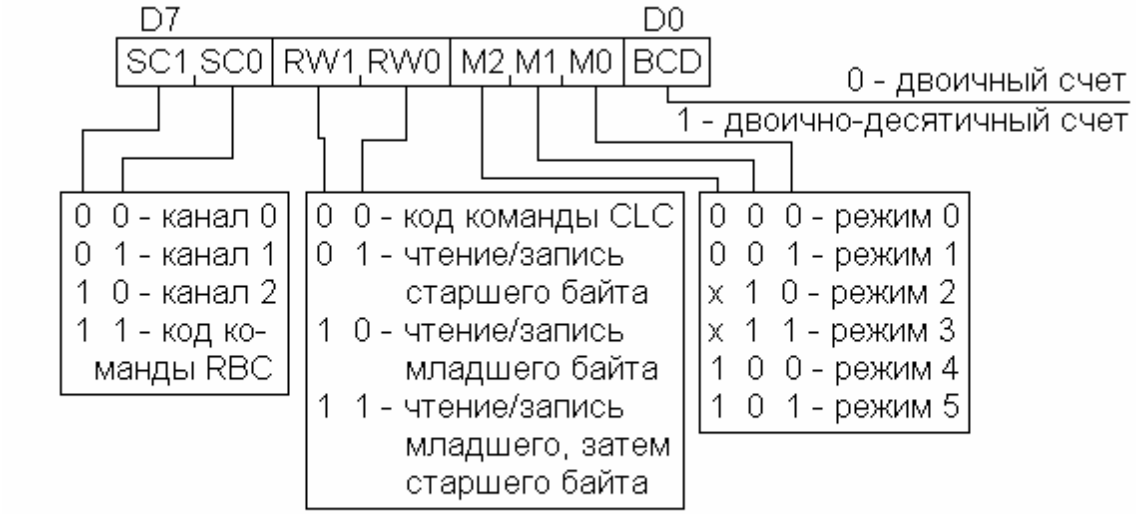


Рис.3.26. Формат управляющего слова ПТ

Структурная схема ПТ (см. рис. 3.25) содержит:

- буфер шины данных (BD) и логические схемы управления чтением/записью;

- дешифратор DS, с помощью которого выбирается один из трех каналов либо формируется признак загрузки управляющих слов или команд;
- три идентичных канала COUNT2-COUNT0, реализующих запрограммированную функцию.

Каждый канал содержит:

16-разрядный буферный регистр **OL**, служащий для запоминания и хранения мгновенного значения счетчика CE, которое в любое время может быть записано командой *Защелка* или *Чтение состояния* канала. После выполнения этих команд содержимое OL может быть считано в ЦП без остановки дальнейшего счета в регистре CE;

16-разрядный счетчик/таймер **CE**, работающий в режиме вычитания. Изменение содержимого CE осуществляется по срезу сигнала CLK при GATE=1;

16-разрядный регистр констант пересчета **CR**, служащий для хранения констант пересчета. Содержимое CR загружается в CE для счета в зависимости от запрограммированного режима;

8-разрядный регистр состояния канала **RS**, содержимое которого можно считывать в ЦП с помощью команды RBC – *Чтение состояния канала*. Содержимое этого регистра является словом состояния канала, формат которого представлен на рис. 3.26.

8-разрядный регистр управляющего слова **RSW**, предназначенный для его хранения. Слово загружается в RCW командой OUT с адресом, формирующим на входах A0, A11 код 11. Выбор конкретного канала осуществляется с помощью двух старших разрядов самого управляющего слова.

Схема управляющей логики канала CL осуществляет управление входом/выходом счетчика/таймера в зависимости от запрограммированного режима.

Программирование. ПТ относится к классу устройств функционально ориентированных и программно управляемых интерфейсных БИС. Поэтому перед

началом работы в него необходимо загрузить управляющее слово (УС) и константу пересчета. УС задает один из шести режимов работы, тип счета (двоичный или двоично-десятичный), порядок загрузки и размерность (один или два байта) константы. Времяимпульсная функция формируется на выходе OUT при GATE=1. Формирование функции осуществляется с помощью счетчика-таймера CE, работающего в режиме вычитающего счетчика по срезу сигнала CLK.

Программист может опросить состояние каналов ПТ с помощью специальных команд *Защелка* (чтение на лету – CLK) или *Чтение состояния канала* (RBC). Эти команды позволяют, не прерывая счета, опросить состояние счетчика/таймера CE. Кроме того, команда RBC позволяет прочитать содержимое регистра состояния канала, разряды которого несут информацию о запрограммированном режиме, состояния выхода OUT и флага «обновления».

Управляющее слово **CW**. Формат УС показан на рис. 3.26. Управляющие слова загружаются в регистры RSW каналов ПТ по командам вывода, формирующим на входах ПТ коды A0A1=11, CS#=0, WR#=0, RD#=1. Управляющие слова загружаются в тот канал ПТ, адрес которого указывается в самом формате УС, и сохраняются там во все время работы или до следующего программирования. Загрузка УС должна предшествовать загрузке констант.

В формате УС можно выделить четыре функциональных поля: SC, RW, M, BCD, с помощью которых задаются основные параметры работы канала.

Поле **SC** (разряды D7, D6) определяет адрес регистра RSW_n конкретного канала. Если в этом поле содержится код 11, то загружаемая информация воспринимается ПТ как команда *Чтение состояния канала* (см. далее описание команды RBC).

Поле **RW** (разряды D5, D4) определяет размерность и порядок загрузки констант. Если в поле RW заданы коды 01 или 10, то размер константы определен соответственно только старшим или только младшим байтом. Если в поле RW задан код 11, то размер константы – два байта; сначала загружаемый байт воспринимается как команда *Защелка* (см. далее описание команды CLC).

Поле **M** (разряды D3 – D1) задает один из шести режимов работы канала:

режим 0 (000) – прерывание от таймера;

режим 1 (001) – программируемый ждущий мультивибратор;

режим 2 (x10) – импульсный генератор частоты;

режим 3 (x11) – генератор импульсов со скважностью два;

режим 4 (100) – программно-запускаемый одновибратор;

режим 5 (101) – аппаратно-запускаемый одновибратор.

В режимах 2, 3 разряд D3 может принимать любое значение. (Подробно работа каналов в соответствующих режимах будет рассмотрена далее).

Поле **BCD** (разряд D0) определяет тип счета. При D=0 константа задается в двоичном коде и может принимать значения в диапазоне 0 – 65536. При D0=1 константа задается в двоично-десятичном коде в диапазоне 0 – 9999.

После загрузки УС необходимо загрузить в каналы константы пересчета. Константа пересчета загружается в ПТ также по командам вывода, но с адресом, формирующим на входах A0, A1 код соответствующего канала (00, 01, 10). Константа может быть задана байтом (старшим или младшим) или 16-разрядным словом (как это определено полем RW управляющего слова) и представлена двоичным или двоично-десятичным кодом (как определено полем BCD). Порядок загрузки каналов управляющими словами и константами строго определен. Возможны два варианта. Первый предполагает загрузку в любой последовательности сначала всех УС, затем констант пересчета. Второй предполагает загрузку управляющего слова для любого канала, а затем константы пересчета для этого же канала.

Общими и обязательными требованиями для загрузки УС и констант являются следующие: 1) загрузка УС должна опережать загрузку константы; 2) загрузка констант всегда должна выполняться до конца, как определено разрядами RW1, RW0 в формате УС.

Состояние счетчика/таймера CE можно прочитать тремя способами.

1. *Чтение по обычным командам ввода* позволяет прочесть состояние счетчика таймера CE в любой момент времени. Выполняется с помощью обычных команд ввода с адресом, формирующим на входах A0, A1 код соответст-

вующего канала. Необходимым условием для выполнения этой операции является остановка счета перед выполнением команд чтения, т.е. GATE=0. Операцию чтения необходимо выполнять до конца, т.е. нельзя прочитать только один байт CE, если константа задана 16-разрядным словом. Сначала считывается младший байт, затем старший.

2. *Чтение по команде* **CLC** *(Защелка)*. Команда CLC позволяет прочитать состояние счетчика/таймера CE в любой момент времени без остановки счета. Для этого программист должен загрузить эту команду в ПТ в определенный момент времени. Команда загружается в ПТ так же, как УС, т. е. по команде вывода с адресом, формирующим на входах A0, A1 код 11. Формат команды CLC представлен на рис. 3.27. Разряды D7, D6 задают адрес защелкиваемого канала. Эти разряды не могут принимать значение 11, являющиеся кодом команды RBC. Разряды D5, D4 являются кодом команды CLC и принимают значение 00. Разряды D3-D0 не участвуют в операции и могут принимать любое значение. Таким образом, существует три команды CLC, отличающиеся кодом в разрядах D7, D6:

CLC0=0000xxxx защелкивается канал 0;

CLC1=0100xxxx защелкивается канал 1;

CLC2=1000xxxx защелкивается канал 2.

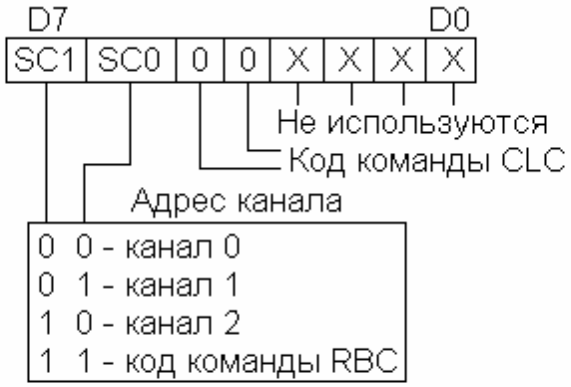


Рис.3.27. Формат команды CLC

После загрузки команды CLC выполняется операция чтения так же, как в первом способе, с теми же необходимыми условиями, кроме снятия сигнала

GATE. Фактически команда CLC записывает состояние счетчика/таймера CE в буферный регистр OL, из которого информация считывается без нарушения продолжения счета. Информация из OL может быть считана в любое время. Необходимо помнить, что если операция чтения не выполнена или выполнена не до конца, то новая команда CLC не воспринимается. Прочитать состояние регистра OL можно только после выполнения хотя бы одного цикла в CE.

3. *Команда* **RBC** (*Чтение состояния канала*) позволяет в любой момент времени прочитать слово состояния канала (SW), т.е. содержимое регистра RS, а также выполнить защелку одного или нескольких каналов одновременно. Формат команды RBC представлен на рис. 3.28. Она загружается в ПТ так же, как УС, т е. по команде вывода с адресом, формирующим код A0A1=11. Старшие разряды (D7D6=11) являются кодом, по которому ПТ распознает информацию на входе как команду RBC.

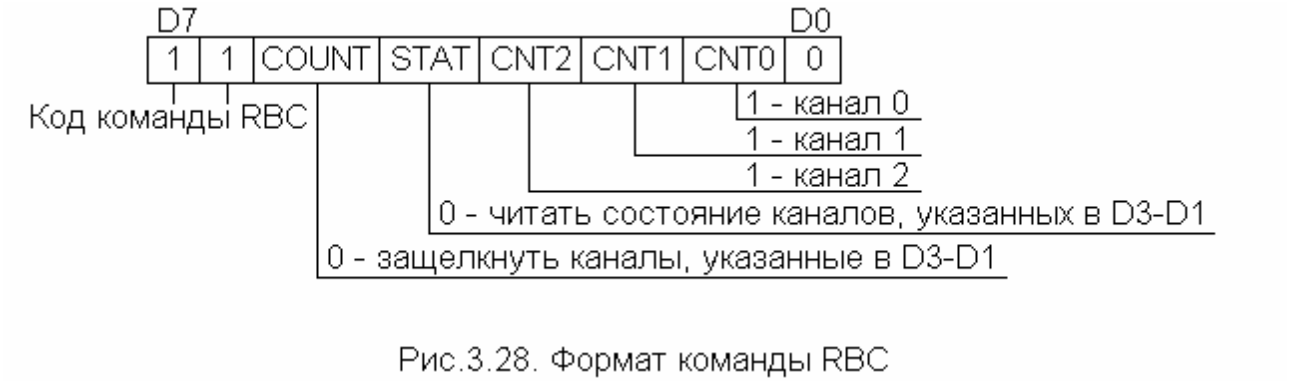


Рис.3.28. Формат команды RBC

Команда RBC может выполнять две операции: *Защелку*, что аналогично команде CLC, и *Чтение регистра состояния канала*. Эти операции задаются независимо кодами D5=0, D4=0. Возможно совмещение операций. При D5D4=11 операций нет. Особенностью этой команды является возможность выполнения операций одновременно в нескольких каналах. Установка кода D3D2D1=111 определяет операцию с соответствующим каналом.

Например, команда RBC=11001110 одновременно защелкивает/ перезаписывает состояние регистров CE всех трех каналов в собственные регистры OL, а также состояние каналов в RS и делает их доступными для чтения. Эта

команда эквивалентна трем командам CLC. Как и при команде CLC, новое «защелкивание» возможно только после полного считывания содержимого OL.

Слово-состояние канала **SW**. Каждый канал имеет регистр слова-состояния RS. Формат слова-состояния показан на рис. 3.29, из которого видно, что в процессе работы канала изменяются только два старших разряда слова состояния: D7 и D6. Остальные разряды соответствуют разрядам ранее загруженного управляющего слова, что позволяет контролировать правильность его загрузки.

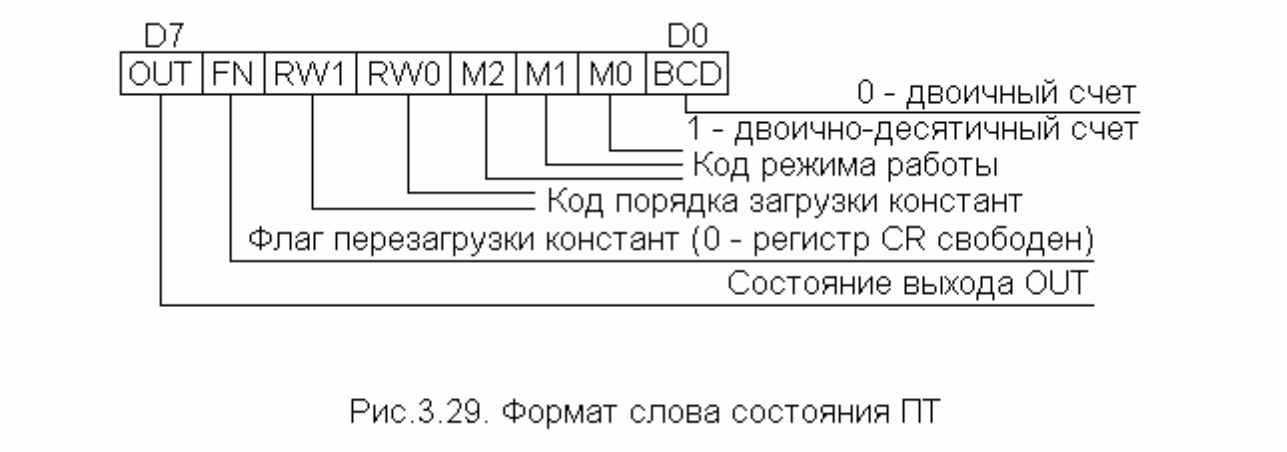


Рис.3.29. Формат слова состояния ПТ

Разряд D7 слова-состояния канала несет информацию о состоянии выхода канала OUT в момент выполнения команды RBC. Разряд D6 является флагом обновления регистра констант CR. Этот флаг особенно необходим для режимов 1 и 5, он позволяет определить, произошла ли загрузка константы из регистра CR в CE или нет. Напомним, что для этих режимов загрузка осуществляется аппаратно – по фронту сигнала GATE. Поэтому если флаг обновления установлен в нуль, то это означает, что ранее загруженная константа перезаписана из CR в CE и по ней осуществляется операция счета. В этом случае можно произвести загрузку новой константы, не нарушая предыдущего счета. По фронту следующего сигнала GATE начинается счет с новой константой.

Изменение состояния флага обновления при работе и программировании можно проиллюстрировать следующим примером:

при записи УС флаг FN=1;

при записи констант в CR FN=1;

при загрузке константы в CE FN=0.

Подключение ПТ к МПС приведено на рис. 3.30. Входы RD#, WR# ориентируют входной буфер ПТ на прием или выдачу, поэтому подключаются к соответствующим выходам ЦП (I/OR#, I/OW), на которых формируется сигнал низкого уровня при выполнении команд ввода-вывода. Вывод CS# является входом разрешения выбора ПТ. Нулевой потенциал на нем разрешает работу входного буфера ПТ. Этот вход используется для адресации ПТ, он подключается либо к одной из свободных шин адреса (принцип раздельной дешифрации), либо к выходу дешифратора ВУ. Выводы A0, A1 являются входами адресов внутренних регистров ПТ, они подключаются к шинам адреса, не занятым в адресации ПТ, обычно к A0, A1– шины адреса МПС.

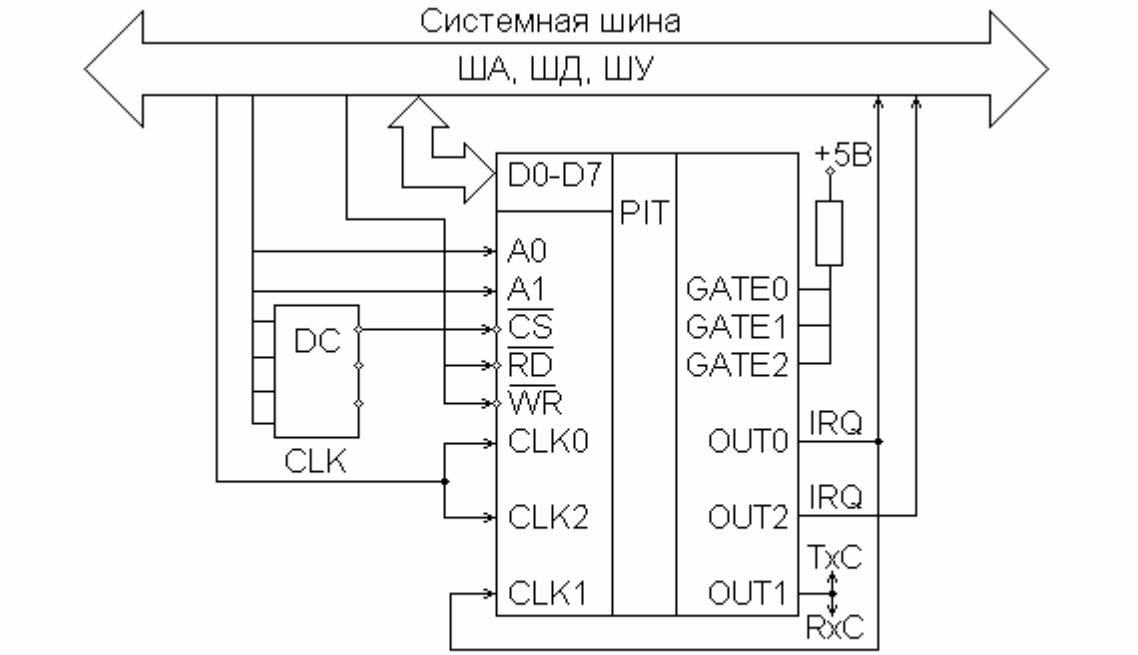


Рис.3.30. Схема подключения ПТ к шине МПС

Функционирование. Каналы ПТ независимо могут быть запрограммированы на работу в одном из шести режимов.

В режиме 0 – прерывание от таймера – низкий уровень сигнала на выводе OUT устанавливается сразу же после загрузки УС. Загрузка константы не оказывает влияния на этот выход. Счет разрешается положительным сигналом на входе GATE. Изменение состояния счетчика/таймера СЕ осуществляется по срезу сигнала CLK, причем по первому тактовому сигналу происходит загрузка

CE константой из CR, и только второй тактовый сигнал принимает участие в счете. После отсчета загруженного числа устанавливается сигнал OUT=1. Таким образом, сигнал OUT=0 удерживается на время N+1 тактовых периодов, где N – загруженная константа.

Если во время счета снять сигнал GATE, то счет приостанавливается, содержимое счетчика/таймера сохраняется. Новый положительный сигнал на входе GATE вызывает продолжение счета без перезагрузки CE содержимым CR. Загрузка новой константы во время счета приводит: при записи младшего байта к остановке текущего счета, а при записи старшего – к запуску нового цикла счета.

Контроль счетчика (выполнение команд CLC, RBC) в этом режиме возможен только после хотя бы одного цикла счета. На рис. 3.31 показана временная диаграмма работы ПТ в режиме 0.

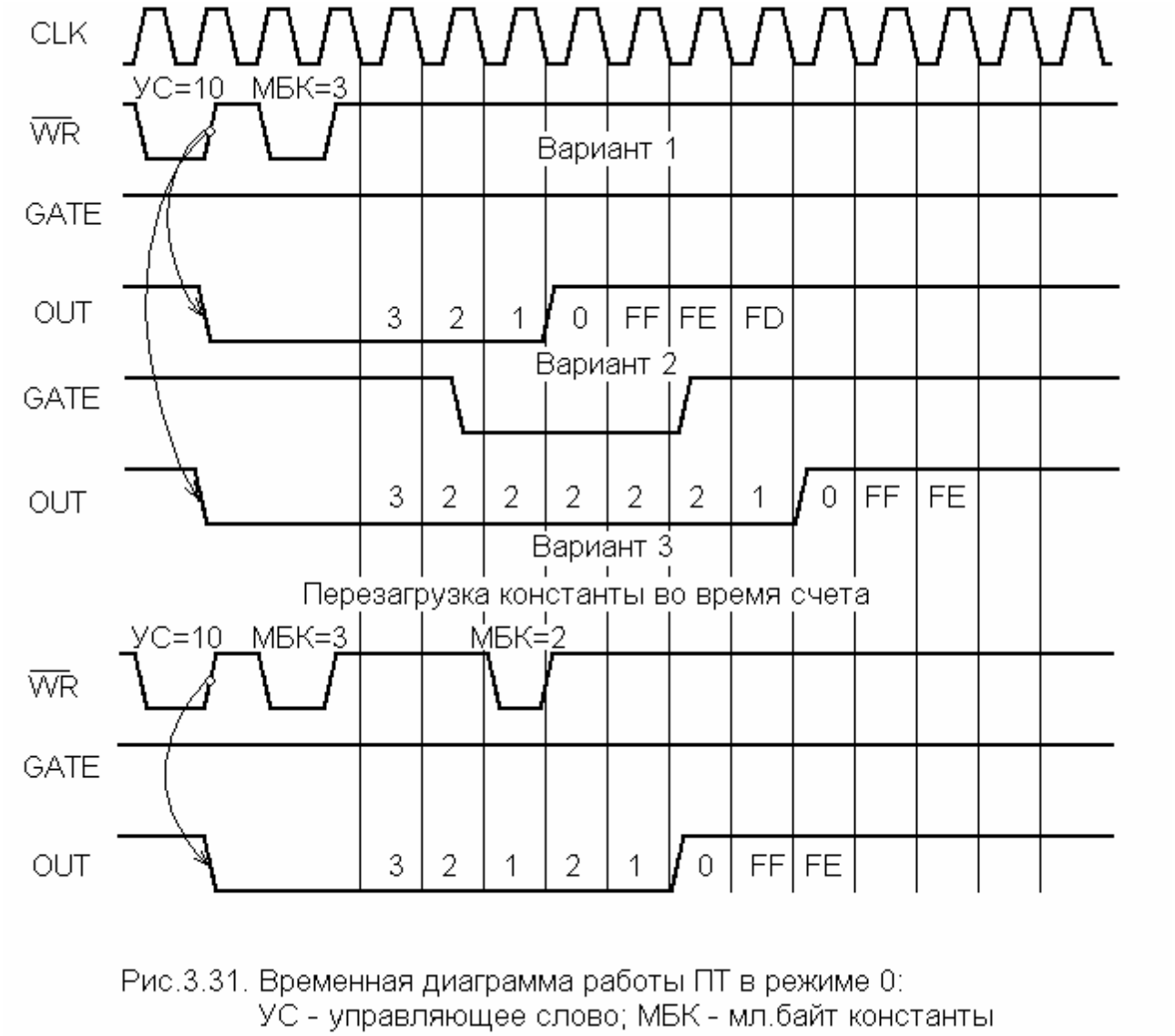


Рис.3.31. Временная диаграмма работы ПТ в режиме 0:
УС - управляющее слово; МБК - мл.байт константы

В режиме 1 -режим программируемого ждущего мультивибратора – на выходе OUT формируется сигнал низкого уровня длительностью $T=T_{CLK}N$, где T_{CLK} – период тактовых импульсов; N – константа. На выходе OUT по положительному фронту сигнала GATE устанавливается нулевой сигнал, который изменяется после окончания счета. Режим 1 является режимом с перезапуском. По каждому фронту сигнала на входе GATE регистр CE перезагружается содержимым CR. Это означает, что однажды загруженная константа участвует в счете всякий раз по фронту сигнала GATE, причем по фронту первого сигнала GATE флаг обновления устанавливается в нуль.

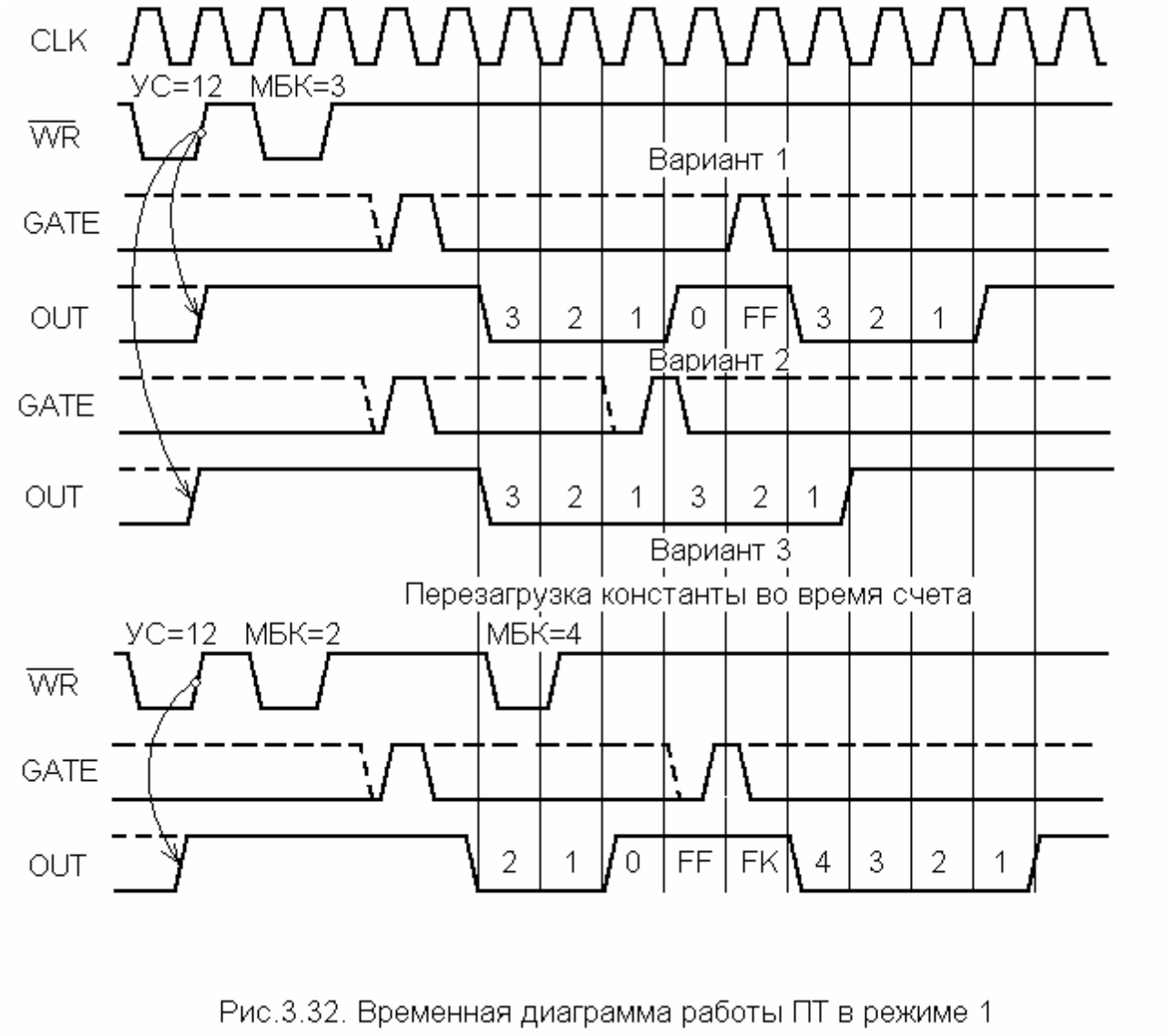


Рис.3.32. Временная диаграмма работы ПТ в режиме 1

Если во время счета в ПТ загружается новая константа, то она устанавливает флаг обновления в единицу, но не влияет на текущий счет. Новый счет начинается только по фронту следующего сигнала GATE. Выполнение команд CLC и RBC возможно только после выполнения хотя бы одного цикла счета. Временная диаграмма работы ПТ в режиме 1 показана на рис. 3.32.

В режиме 2 –режим импульсного генератора частоты – канал работает как делитель входной частоты F_{CLK} на N. Сразу же после загрузки УС на выходе OUT устанавливается единичный сигнал. При GATE=1 на выходе OUT с частотой F_{CLK}/N устанавливается нулевой сигнал на время одного периода CLK. Режим 2 является режимом с автозагрузкой, т. е. после окончания цикла счета СЕ автоматически перезагружается и счет повторяется. Перезагрузка канала новой константой не влияет на текущий счет, новый счет начинается по окон-

чании предыдущего. При GATE=0 на выходе OUT устанавливается напряжение высокого уровня и счет останавливается. При сигнале GATE=1 счет продолжается, что позволяет синхронизировать работу канала с внешними событиями. Выполнение команд CLC и RBC возможно для этого режима после окончания двух циклов счета. Временная диаграмма работы ПТ в режиме 2 показана на рис. 3.33 для трех различных вариантов.

Режим 3 – режим генератора импульсов со скважностью два – аналогичен режиму 2, за тем исключением, что на выходе OUT формируются импульсы с длительностью полупериода, равной $(N/2)T_{CLK}$ при четных N; $((N+1)/2)T_{CLK}$ для положительных и $((N-1)/2)T_{CLK}$ для отрицательных полупериодов при нечетных N.

Этот режим является режимом с автозагрузкой, т.е. перезагрузка CE константой из CR выполняется автоматически после окончания цикла счета. Перезагрузка константы во время счета не влияет на текущий счет, новый счет начинается по окончании предыдущего. Снятие сигнала GATE приостанавливает счет, установка его продолжает цикл счета. В этом режиме канал может работать только с константой больше трех. Выполнение команд CLC и RBC возможно только после двух циклов счета. Временная диаграмма работы ПТ в режиме 3 показана на рис. 3.34.

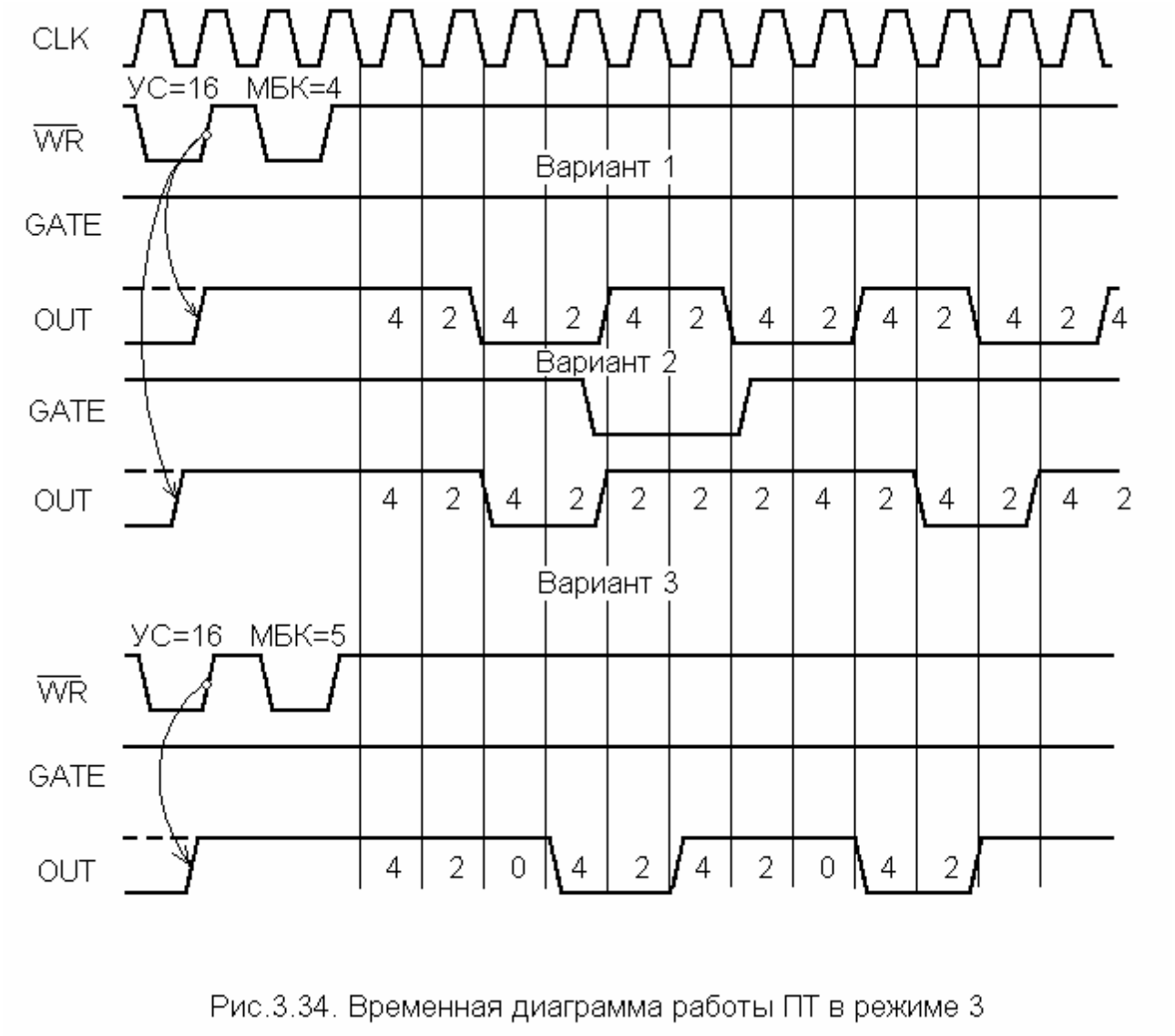


Рис.3.34. Временная диаграмма работы ПТ в режиме 3

В режиме 4 - программно-запускаемого одновибратора – по окончании отсчета числа, загруженного в счетчик/таймер, на выходе OUT устанавливается нулевой сигнал на время одного периода сигнала CLK. Высокий уровень сигнала на выходе OUT устанавливается сразу же после загрузки УС. Сигнал высокого уровня на входе GATE разрешает счет, причем первым тактовым сигналом происходит загрузка счетчика/таймера СЕ константой из CR, а второй тактовый сигнал начинает счет. Таким образом, сигнал длительностью, равной периоду тактовой частоты, устанавливается на выходе OUT через N+1 тактовых периодов. Если во время счета снимается сигнал CATE, то счет приостанавливается, текущее значение СЕ счетчика/таймера сохраняется. Новый положительный сигнал на GATE вызывает продолжение счета. Это режим одноразового выполнение функции таймером. Загрузка новой константы во время счета

приводит при записи младшего байта к остановке текущего счета, а при записи старшего – к запуску нового цикла счета. Выполнение команд CLC и RBC возможно только после окончания одного цикла счета.

Временная диаграмма работы ПТ в режиме 4 показана на рис. 3.35.

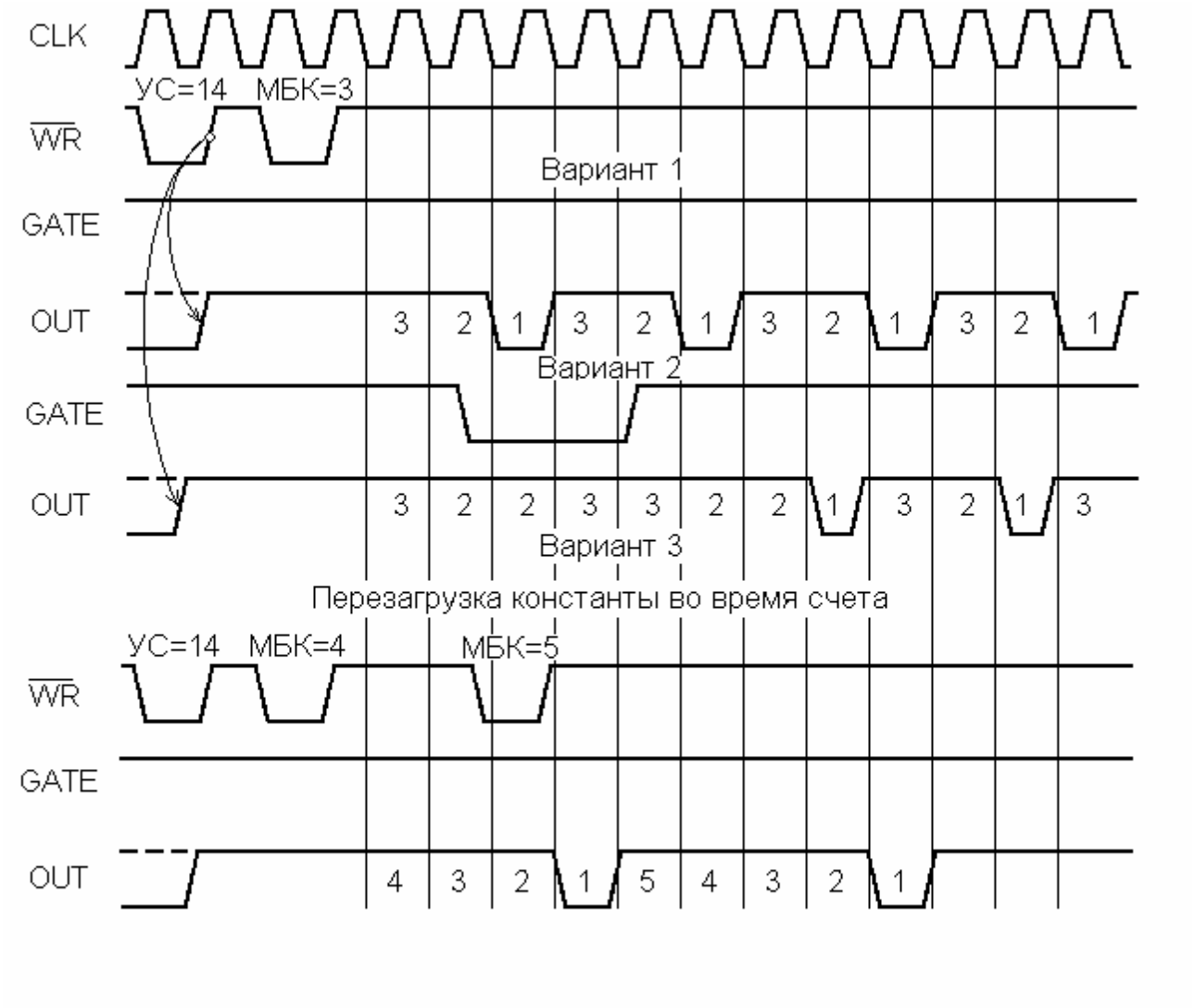


Рис.3.33. Временная диаграмма работы ПТ в режиме 2

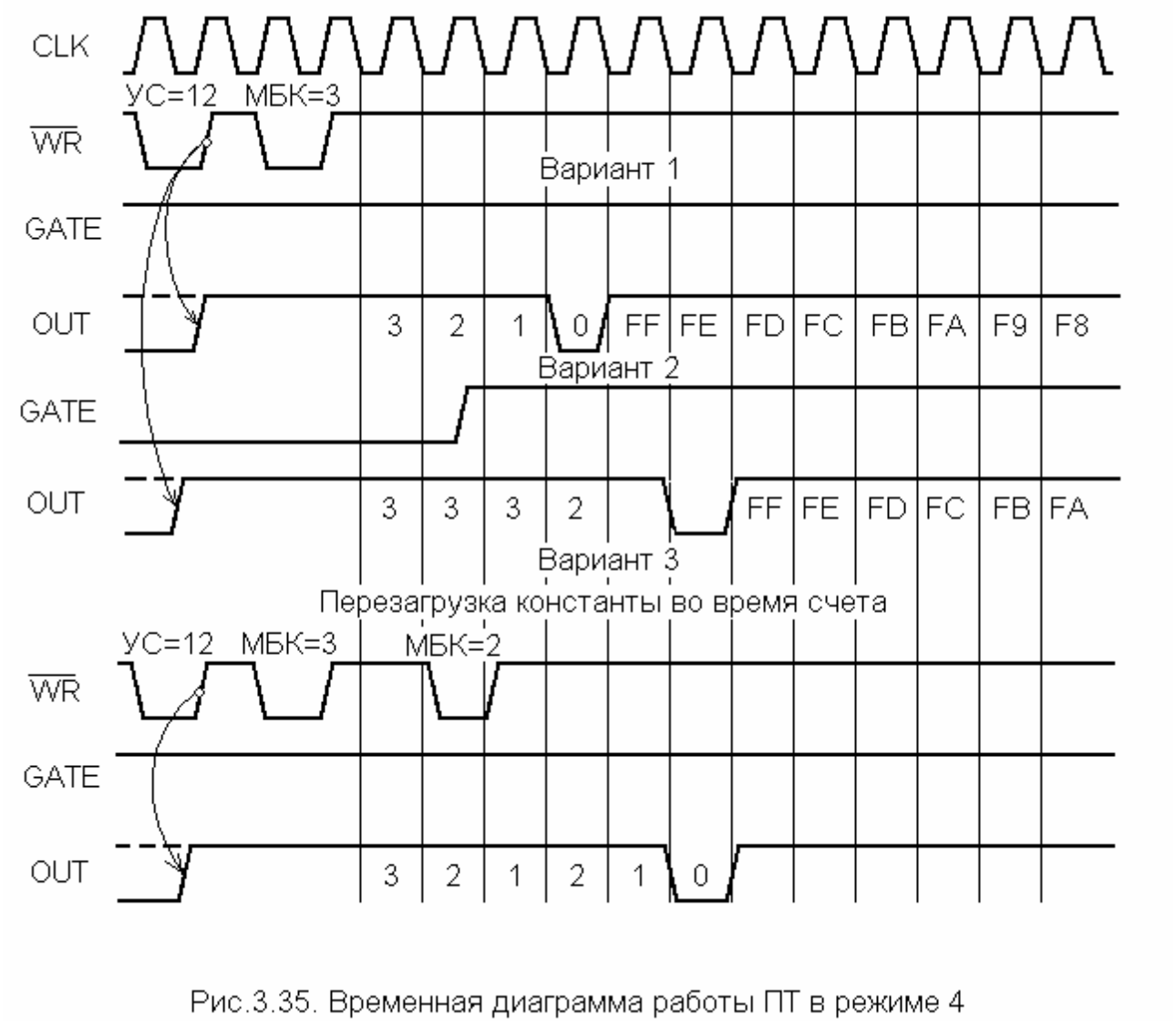


Рис.3.35. Временная диаграмма работы ПТ в режиме 4

Режим 5 – режим аппаратно-запускаемого одновибратора – аналогичен режиму 4 по способу формирования сигнала на выходе OUT и режиму 1 по действию сигнала GATE.

На выходе OUT устанавливается сигнал нулевого уровня на время одного периода CLD после отсчета загруженной в CE константы. Загрузка в CE константы из CR осуществляется по фронту сигнала GATE. Из этого следует, что по фронту GATE происходит новая загрузка CE из CR, причем первый фронт GATE устанавливает флаг обновления в нуль. Если во время счета в канал загружается новая константа, то эта операция устанавливает флаг обновления в единицу, но не влияет на текущий счет. Новый счет начинается только по фронту следующего сигнала GATE. Выполнение команд CLC и RBC возможно только после выполнения хотя бы одного цикла счета.

Временная диаграмма работы ПТ в режиме 5 показана на рис. 3.36.

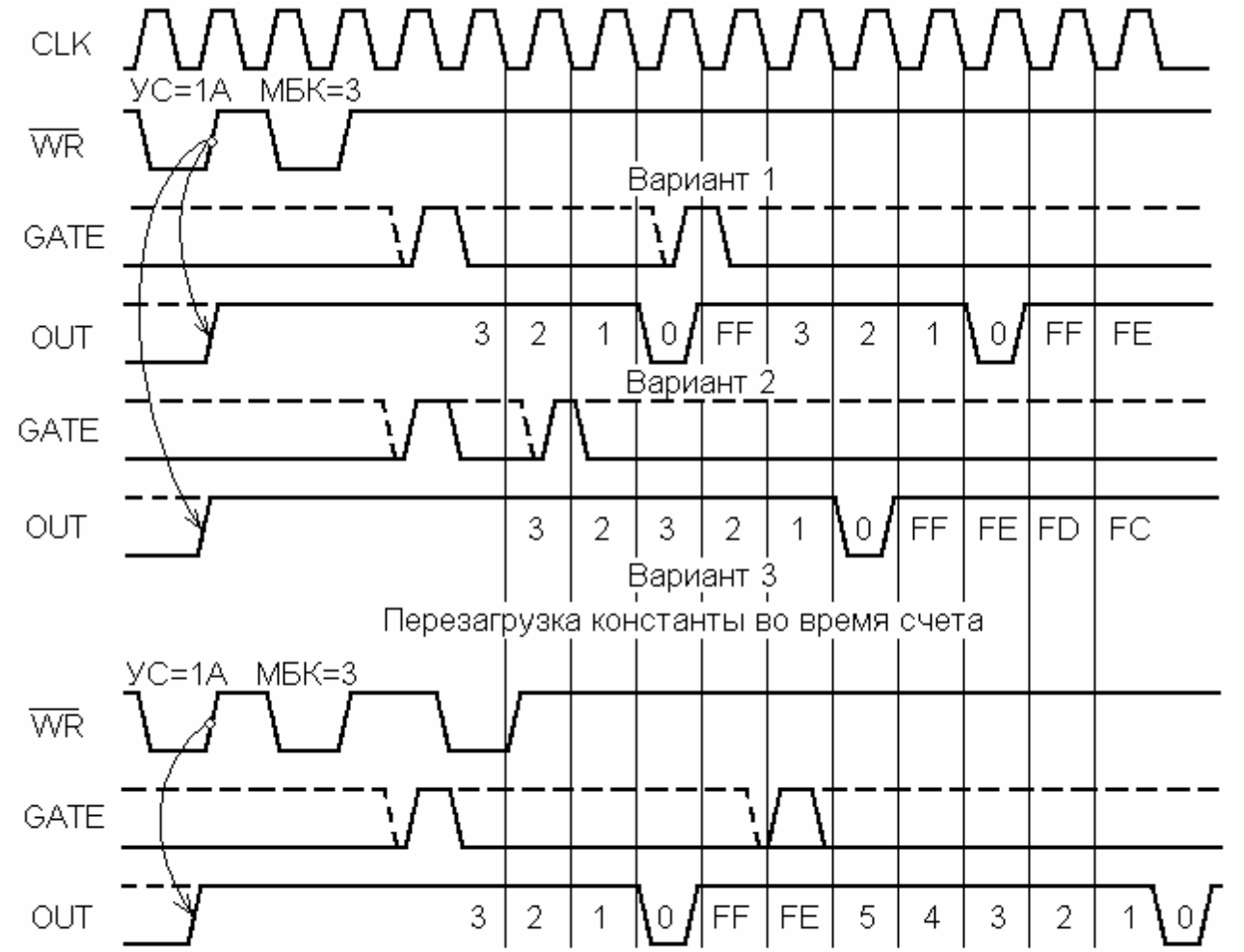


Рис.3.36. Временная диаграмма работы ПТ в режиме 5

Пример инициализации.

Сформировать три периодические импульсные последовательности сигналов. Схема подключения ВИ54 приведена на рис. 3.30. Адреса микросхемы ВИ54 – в системе 64h – 67h.

Все три счетчика необходимо установить в режим импульсного генератора частоты (режим 2) с автозагрузкой (см. рис. 3.33), а коэффициенты пересчета (деления) установить: 0007h, 0A9Ch и 52h.

Инициализация счетчика 0.

```
MVI  A,3Ch;          Загрузка в счетчике 0 управля-
OUT  , 67h;          ющего слова CW=0011 1100 B.
```

```
MVI  A, 07h;        Загрузка младшего байта коэф-
OUT  , 64h;         фициента деления в счетчик 0.
MVI  A, 00h;        Загрузка старшего байта коэф-
OUT  , 64h;         фициента деления в счетчик 0.
```

Допишите для счетчика 1 и для счетчика 2.

Задания

3.31. Разработайте фрагмент программ инициализации таймера ВИ54, выполняющего функции: счетчика внешних событий (N=2C8h); делителя частоты тактовых импульсов (N=8924) и схемы задержки на 2000 периодов CLK.

3.32. В соответствии с заданием в пункте 3.31 изобразите схему подключения таймера ВИ54 к МПС.

3.33. Составьте фрагменты программ вариантов считывания содержимого счетчиков без останова их работы.

3.34. Изучите режимы таймера по осциллограммам (см. рис.3.31 – 3.36).

3.35. Поясните адресацию к регистрам таймера.

3.36. Определите счетчик и его режим по управляющему слову (рис. 3.37).

1	0	1	1	1	0	0	0

Рис. 3.37. Формат управляющего слова

3.5. Контроллер прямого доступа к памяти К1810ВТ37

Общие принципы организации ПДП. Режим ПДП является самым скоростным способом обмена, который реализуется с помощью специальных аппаратных средств – контроллеров ПДП без использования программного обеспечения. Для осуществления режима ПДП контроллер должен выполнить ряд последовательных операций (рис. 3.38):

1) принять запрос DREQ на ПДП от ВУ;
2) сформировать запрос HRQ на захват шин для ЦП;

3) принять сигнал HLDA, подтверждающий этот факт после того, как ЦП войдет в состояние захвата (ШД, ША, ШУ в z-состояние);
4) сформировать сигнал DACK, сообщающий ВУ о начале выполнения циклов ПДП;
5) сформировать на ША адрес ячейки памяти, предназначенный для обмена;
6) выработать сигналы $\overline{\text{MR}}, \overline{\text{IOW}}$ и $\overline{\text{MW}}, \overline{\text{IOR}}$, обеспечивающие управление обменом;
7) по окончании цикла ПДП либо повторить цикл ПДП, изменив адрес, либо прекратить ПДП, снятием запроса на ПДП.

Циклы ПДП выполняются с последовательно расположенными ячейками памяти, поэтому контроллер ПДП должен иметь счетчик адреса ОЗУ. Число циклов ПДП определяется специальным счетчиком. Управление обменом осуществляется специальной логической схемой, формирующей в зависимости от типа обмена пары управляющих сигналов: $\overline{\text{MR}}, \overline{\text{IOW}}$ (циклы чтения), $\overline{\text{MW}}, \overline{\text{IOR}}$ (циклы записи). Из изложенного следует, что контроллер ПДП по запросу должен взять на себя управление системными шинами и выполнять совмещенные циклы чтения/вывода или записи/ввода до тех пор, пока содержимое счетчика циклов ПДП не будет равно нулю. На рис. 3.38 показана структурная схема МПС с контроллером ПДП.

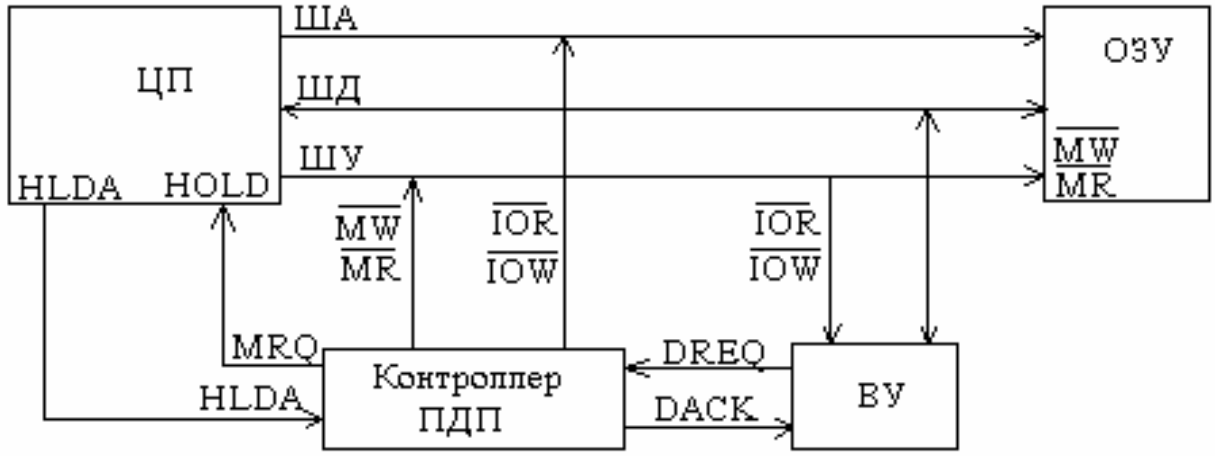


Рис. 3.38. Структурная схема МПС с контроллером ПДП

Контроллер ПДП К1810ВТ37 используется в составе МПС, выполненных на базе МПК К580, К1810, К1821, для реализации прямого доступа к памяти по четырем независимым каналам с положительным или отрицательным приращением адреса со скоростью до 1.6 Мбайт/с. КПДП позволяет реализовать передачу память – память, имея широкие возможности программного управления и каскадирования. Каждый канал может выполнить до 64К циклов ПДП и имеет возможность автоматической инициализации, т. е. повторения циклов ПДП с теми же параметрами.

Назначение выводов КПДП (рис. 3.39).

CLK – вход для подключения тактового генератора F_{CLK}=3 МГц.

$\overline{\mathbf{CS}}$ – выбор кристалла. $\overline{\mathrm{CS}} = 0$ разрешает работу КПДП.

RESET – сброс. Сигнал высокого уровня переводит КПДП в исходное состояние, устанавливая в нуль регистры команд, условий, временного хранения, а также устанавливая в единицу все разряды маски.

READY – готовность. Входной сигнал, используемый для синхронизации работы КПДП с медленнодействующими устройствами.

HLDA – подтверждение захвата. Входной сигнал, используемый ЦП для сообщения КПДП о возможности выполнения циклов ПДП.

DREQ3 – DREQ0 – входы запросов на ПДП от внешних устройств. Полярность запросов задается программно. Сигналы на этих входах должны удерживаться до прихода сигнала DACK. В исходном состоянии приоритет запросов естественный , DREQ0 имеет наивысший приоритет.

DB7 – DB0 – двунаправленная шина данных с буфером, имеющим z-состояние. В циклах ПДП на эти линии выдается восемь старших разрядов адресного кода, которые необходимо «защелкнуть» на внешнем регистре сигналом ADSTB. В режиме работы с ЦП по этим линиям осуществляется прием/передача данных.

$\overline{\textbf{IOR}}$ – чтение; как вход используется ЦП для чтения содержимого внутренних регистров КПДП; как выход в режиме ПДП разрешает выдачу данных из внешних устройств.

$\overline{\textbf{IOW}}$ – запись; как вход используется ЦП для загрузки данных в регистры КПДП; как выход в режиме ПДП разрешает запись данных в регистры внешних устройств.

$\overline{\textbf{EOP}}$ – окончание процесса. Вход/выход, используемый для указания окончания процесса передачи данных в режиме ПДП. Подавая на этот вход сигнал низкого уровня, можно прекратить передачу данных. После завершения передачи данных по одному из каналов на выходе устанавливается сигнал $\overline{\text{EOP}} = 0$. По этому сигналу (внешнему или внутреннему) снимается запрос, и обслуживание прекращается. Если установлен режим автоинициализации, то происходит загрузка рабочих регистров данного канала содержимым базовых регистров, а разряды регистра маски не меняются. В режимах без автоинициализации разряды маски и разряд ТС в слове-состоянии устанавливаются в соответствии с состоянием обслуженного канала. При передаче память – память вывод $\overline{\text{EOP}}$ ориентирован на выход, и по окончании счета на этом выходе формируется сигнал. Если вывод $\overline{\text{EOP}}$ не используется, то он должен быть подключен через резистор к шине питания (+5 В) для предотвращения формирования ложных сигналов окончания процесса.

A3 – A0 – адресные входы/выходы. Используются как входные в режиме работы с ЦП и для адресации к каналам и регистрам каналов КПДП. В режиме ПДП являются выходами, по которым передаются четыре младших разряда адреса ОЗУ.

A7 – A4 – адресные выходы, на которые в режиме ПДП передаются соответствующие разряды адреса ОЗУ. В режиме работы с ЦП переходят в z-состояние.

HRQ – выход запроса захвата шин. Запрос к ЦП для перехода в режим ПДП.

DACK3 – DACK0 – подтверждение ПДП. Выходные линии, на которые выдаются сообщения для ВУ о возможности выполнения циклов ПДП. Полярность сигнала задается программно. После сигнала RESET на выходах DACK устанавливается нуль.

AEN – разрешение адреса. AEN=1 устанавливается на время выдачи восьми старших разрядов адреса ОЗУ на линии DB7 – DB0.

ADSTB – строб адреса. Выход, на котором формируется импульс (строб), осуществляющий запись старших разрядов (A15 – A8) адреса ОЗУ с шин DB7 – DB0 во внешний буферный регистр.

$\overline{\textbf{MEMR}}$ – чтение из памяти. Выход, используемый в режиме ПДП для управления операцией чтения из памяти.

$\overline{\textbf{MEMW}}$ – запись в память. Выход, используемый в режиме ПДП для управления операцией записи в память.

$\mathbf{U_{cc}}$ – шина питания (+5 В).

GND – общий.

Структура КПДП (см. рис. 3.39). Контроллер включает четыре канала, каждый из которых состоит из четырех 16-разрядных регистров.

Регистр текущего адреса **CAR** хранит текущий адрес ячейки памяти при выполнении цикла ПДП. После выполнения цикла ПДП содержимое этого регистра увеличивается или уменьшается на единицу. Оно может быть прочитано или загружено с помощью двух команд ввода – вывода. Содержимое CAR может

быть обновлено по сигналу $\overline{EOP}$, если запрограммирован режим автоинициализации.

Регистр циклов ПДП **CWR** хранит число слов, предназначенных для передачи. При загрузке этого регистра необходимо помнить, что загружаемая константа должна быть на единицу больше числа слов, необходимых для передачи. При выполнении циклов ПДП регистр работает в режиме вычитающего счетчика. Разряд ТС регистра состояния устанавливается в единицу при переходе из нулевого состояния в состояние FFFFH. Чтение и запись содержимого регистра осуществляется двумя последовательно выполняемыми командами ввода – вывода. Содержимое CWR может быть обновлено при автоинициализации по сигналу $\overline{EOP}$ либо в регистре сохраняется значение FFFFH.

Регистр хранения базового адреса **BAR** *и регистр хранения базового числа циклов ПДП* **WCR** хранят базовые значения адреса и числа циклов ПДП, участвуют в автоинициализации. При начальной загрузке контроллера ПДП исходными параметрами происходит одновременная запись в регистры CAR, BAR, CWR и WCR. В процессе выполнения циклов ПДП содержимое BAR и WCR не изменяется. Прочитать состояние этих регистров невозможно.

Кроме того, каждый канал имеет 6-разрядный *регистр режима* **MR**, определяющий режим его работы. При загрузке этого регистра в младших разрядах D1, D0 указывается код номера канала. Назначение разрядов MR показано на рис. 3.40.

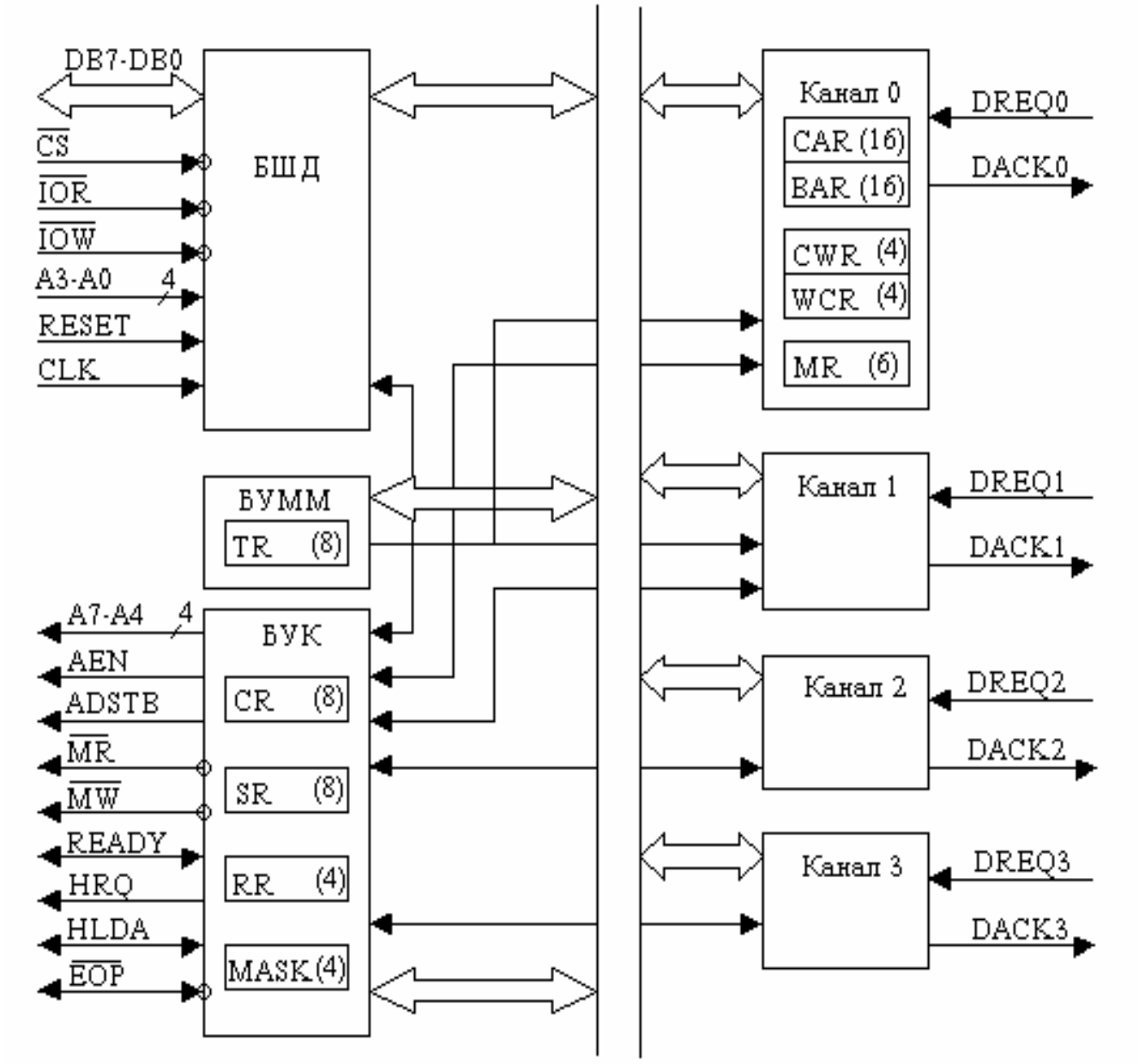


Рис. 3.39. Структурная схема КПДП

С помощью разрядов D2, D3 задается один из типов передачи – чтение, запись, проверка. Эти разряды могут принимать любые значения при D6D7=11. Разряд D4 определяет режим автозагрузки. Если D4=1, то при условии автозагрузки CAR и CWR загружаются параметрами BAR и WCR соответственно. Разряд D5 определяет режим изменения CAR. Если D5=0, после каждого цикла ПДП происходит увеличение содержимого CAR; если D5=1, то происходит уменьшение. Разряды D6, D7 определяют режимы работы канала – передачу по

запросу, одиночную передачу, блочную передачу, контроллер в режиме каскадирования.

Контроллер ПДП включает три функциональных блока, которые выполняют функции управления. *Буфер шины данных* служит для согласования работы контроллера с ЦП. Некоторые сигналы, обеспечивающие эти функции, используются для управления передачей данных в циклах ПДП. *Блок управления контроллером* при передаче память – память включает один 8-разрядный регистр TR временного хранения данных, обеспечивающий хранение байта в цикле передачи память – память на время изменения адреса.

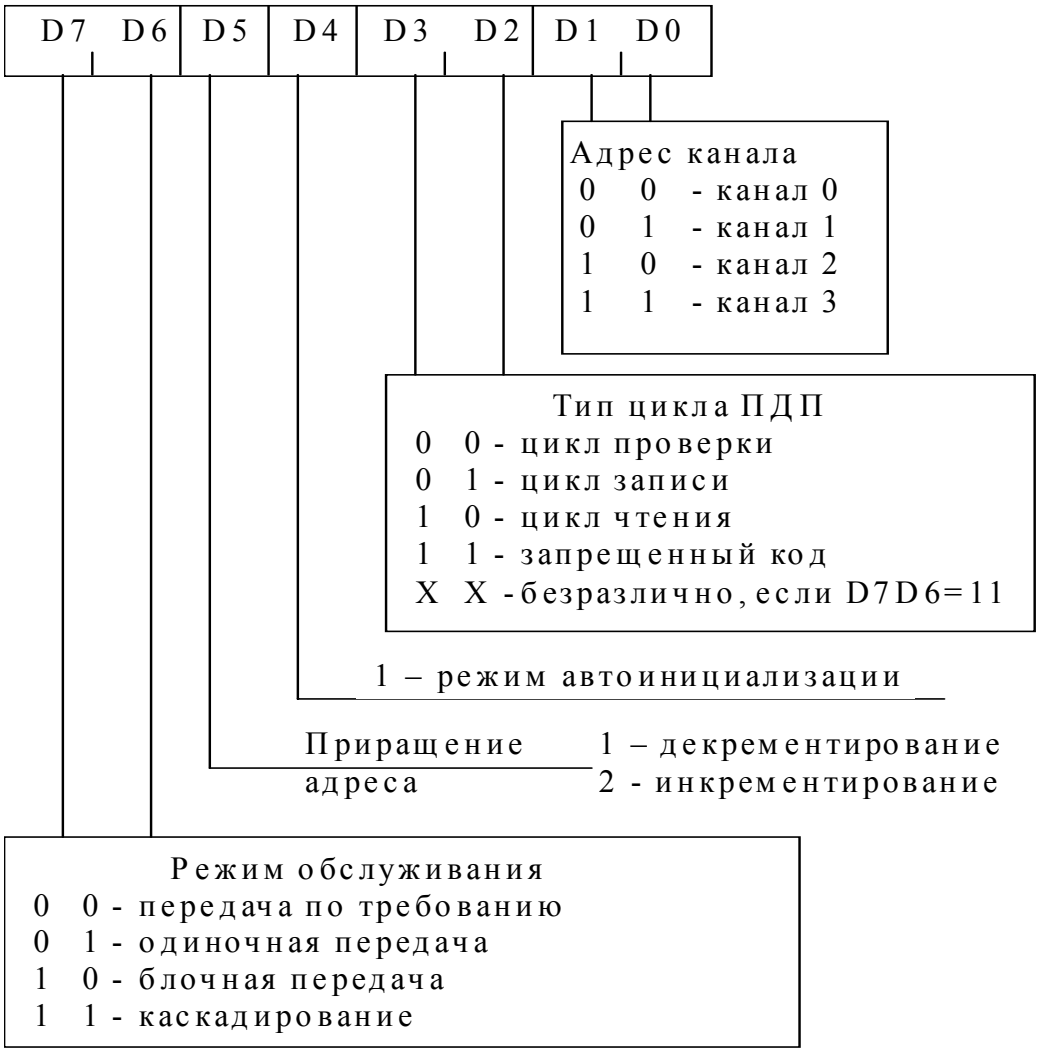


Рис. 3.40. Формат команды установки режима MR

Последнее загруженное в этот регистр слово сохраняется там до поступления сигнала RESET. *Блок управления режимом ПДП* вырабатывает необходимые сигналы управления при передаче данных в циклах ПДП. Включает два 8-разрядных и два 4-разрядных регистра.

Регистр команд **CR** определяет основные параметры работы канала. Загрузка CR осуществляется командой вывода от ЦП, а сброс – по сигналу RESET или команде общего сброса. Назначение разрядов регистра показано на рис. 3.41. Разряды D0, D1 используются для задания режимов работы каналов 0 и 1 в режиме память – память. Разряд D2 инициализирует контроллер для выполнения ПДП, разряд D3 определяет режим выполнения циклов ПДП. Если D3=1, циклы ПДП выполняются с пропуском одного такта при изменении адреса в пределах младшего байта. Разряд D4 устанавливает режим приоритетов. Если D4=1, запросу обслуженного канала присваивается наинизший приоритет – режим вращения приоритета. Разряд D5 устанавливает режим удлиненного цикла записи. Если D5=1, сигналы $\overline{\text{IOW}}$ и $\overline{\text{MEMW}}$ вырабатываются с двойной длительностью. Разрядами D6, D7 программируются уровни запросов на ПДП (DREQ) и сигналов подтверждения ПДП (DACK).

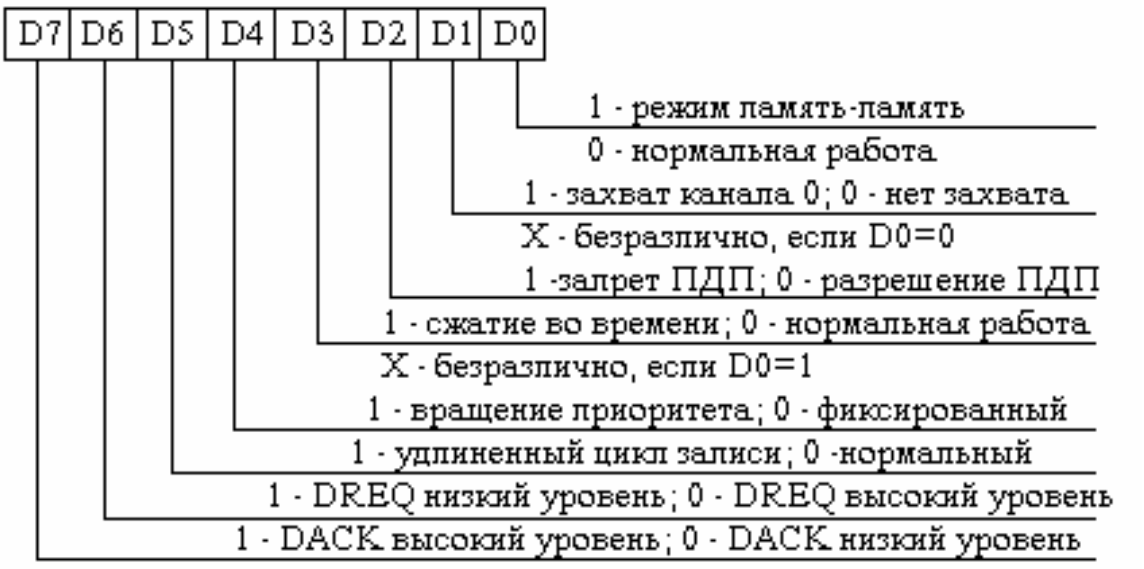


Рис. 3.41. Формат команды управления CR

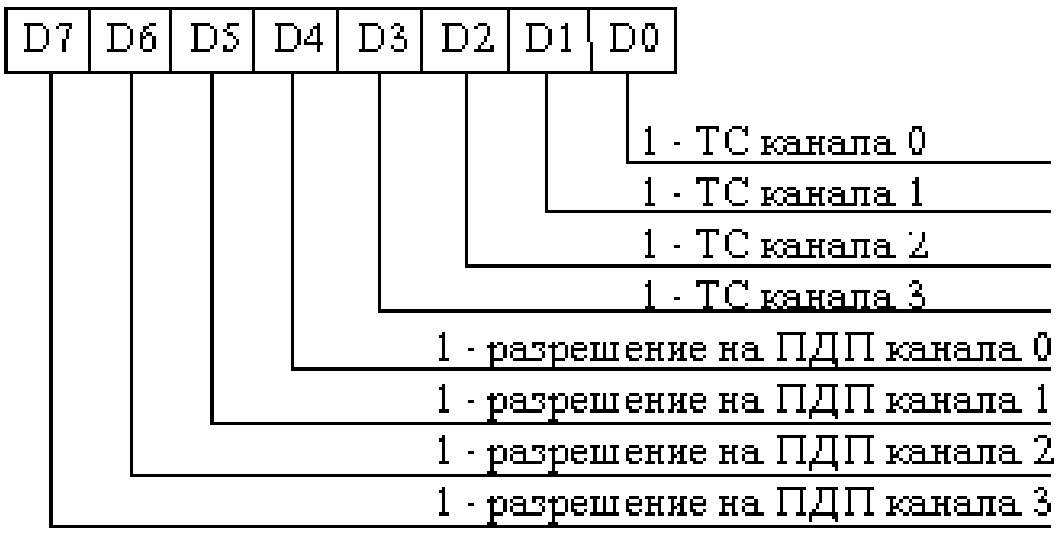


Рис. 3.42. Формат слова-состояния

Регистр условий **SR**. Разряды D3 – D0 этого регистра устанавливаются аппаратно при возникновении сигнала ТС, т. е. после окончания циклов ПДП или по внешнему сигналу $\overline{\text{EOP}}$. Эти разряды сбрасываются (устанавливаются в нуль) сигналом RESET и после выполнения команды чтения содержимого этого регистра. Разряды D4 – D7 устанавливаются программно при необходимости обслуживания по соответствующему каналу. Назначение разрядов SR показано на рис. 3.42.

Регистр запросов **RR**. Контроллер может обслуживать запросы на ПДП, формируемые как аппаратно – по входам DREQ, так и программно – по состоянию разрядов (регистров) запросов RR. Каждый разряд этого регистра соответствует запросу по одному из каналов. Разряды этого регистра не маскируются и устанавливаются раздельно программно или сигналами ТС и $\overline{\text{EOP}}$. Программная установка этих разрядов осуществляется командой, формат которой представлен на рис. 3.43. Сброс всех разрядов RR осуществляется сигналом RESET. Для обработки программного запроса контроллер должен быть запрограммирован в режиме блочной передачи.

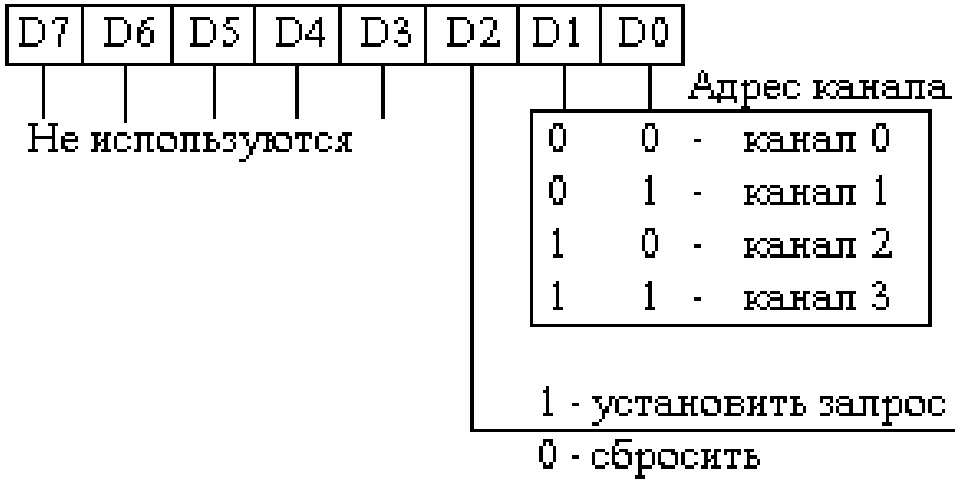


Рис. 3.43. Формат команды установки запросов на ПДП

Регистр маски **MASK**, с помощью которого могут быть замаскированы сигналы DREQ каждого канала. Разряды MASK могут быть установлены специальной командой одновременно (рис. 3.44) или раздельно (рис. 3.45). Кроме того, если канал не запрограммирован на режим автозагрузки, после появления сигнала $\overline{\text{EOP}}$ соответствующий разряд регистра устанавливается в единицу. Все разряды MASK устанавливаются в нули сигналом RESET либо командой CMR (Clear Mask Register).

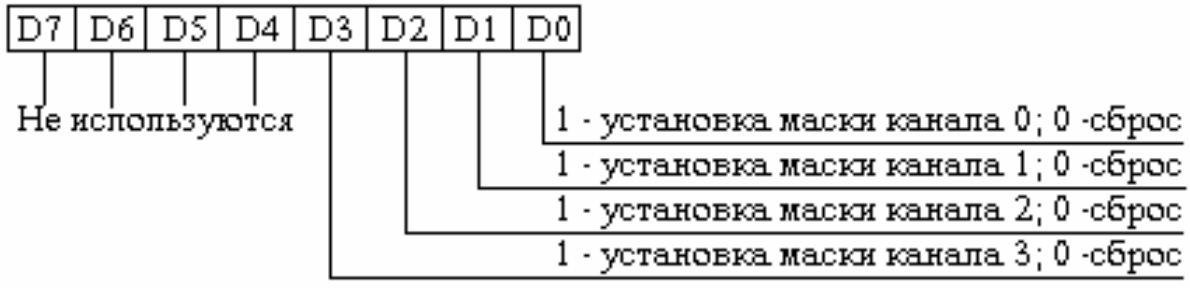


Рис. 3.44. Команда установки всех разрядов маски

Режим работы ПДП. Контроллер ПДП может работать в двух основных режимах: с ЦП и выполнения циклов ПДП. В режиме работы с ЦП контроллер воспринимается им как внешнее устройство, а после загрузки управляющих

слов переходит в пассивное состояние S1. В этом состоянии контроллер находится до тех пор, пока на вход одного из каналов не поступит запрос на ПДП DREQ или этот запрос не будет выставлен программно от ЦП. Обнаружив запрос на ПДП, контроллер переходит в состояние S0 и выставляет сигнал запроса на захват системной шины HRQ, ожидая от ЦП сигнала подтверждения захвата HLDA.При получении сигнала HLDA контроллер начинает выполнять циклы ПДП.

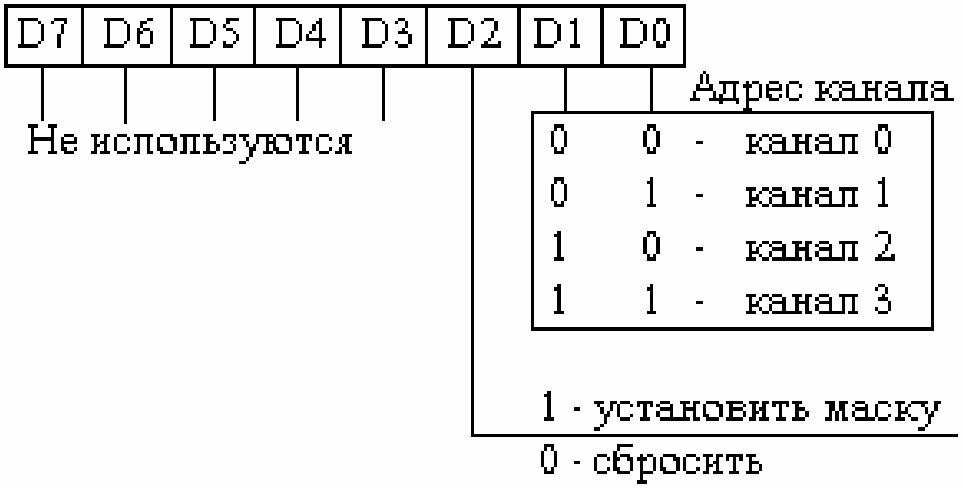


Рис. 3.45. Команда установки разряда маски

Различают четыре рабочих состояния при выполнении этих циклов: S1 – S4. Если при выполнении циклов ПДП на вход READY подать нуль, контроллер между тактами S2/S3 и S4 выполняет такты ожидания SW. Состояние SW характеризуется активностью линий передачи данных. При передаче информации в режиме память – память необходимо выполнить два полных цикла чтения и записи, поэтому для передачи одного слова контроллер выполняет два цикла ПДП по четыре такта в каждом: S11 – S14 для чтения из памяти и S21 – S24 для записи в память.

Временная диаграмма работы контроллера в циклах ПДП представлена на рис. 3.46. В пассивном состоянии происходит опрос входов запросов на ПДП и возможно взаимодействие с ЦП при помощи обычных команд ввода – вывода. Так как взаимодействие с ЦП КПДП чаще осуществляет словом из двух байтов,

то для правильного их выбора контроллер использует внутренний триггер, указывающий на операцию с младшим или старшим байтом слова. Этот триггер сбрасывается сигналом RESET или командой общего сброса, указывая на операцию с младшим байтом. После выполнения операции с младшим байтом он устанавливается в единицу, указывая старший байт.

Контроллер может быть запрограммирован для выполнения следующих четырех режимов работы ПДП. *В режиме одиночной передачи* осуществляется передача одного байта, при этом содержимое счетчика циклов ПДП (CWR) уменьшается, а содержимое адресного регистра (CAR) уменьшается или увеличивается на единицу. Бит окончания передачи (TC) в регистре условий устанавливается в единицу, когда содержимое CWR примет значение FFFFH. Вход DREQ должен поддерживаться в активном состоянии до прихода сигнала DACK. Если DREQ остается активным и после передачи одного байта, сигнал HRQ снимается, а новый цикл передачи возможен с приходом очередного сигнала HLDA.

В режиме блочной передачи циклы ПДП осуществляются до момента установления бита TC в регистре условий, т. е. когда счетчик циклов ПДП CWR примет значение FFFFH или передача остановится по внешнему сигналу $\overline{\text{EOP}}$. Циклы передачи могут быть возобновлены, если канал был запрограммирован на автоинициализацию.

В режиме передачи по требованию циклы ПДП продолжаются до тех пор, пока не установится разряд TC в регистре условий либо не придет сигнал $\overline{\text{EOP}}$, либо не снимется сигнал DREQ.

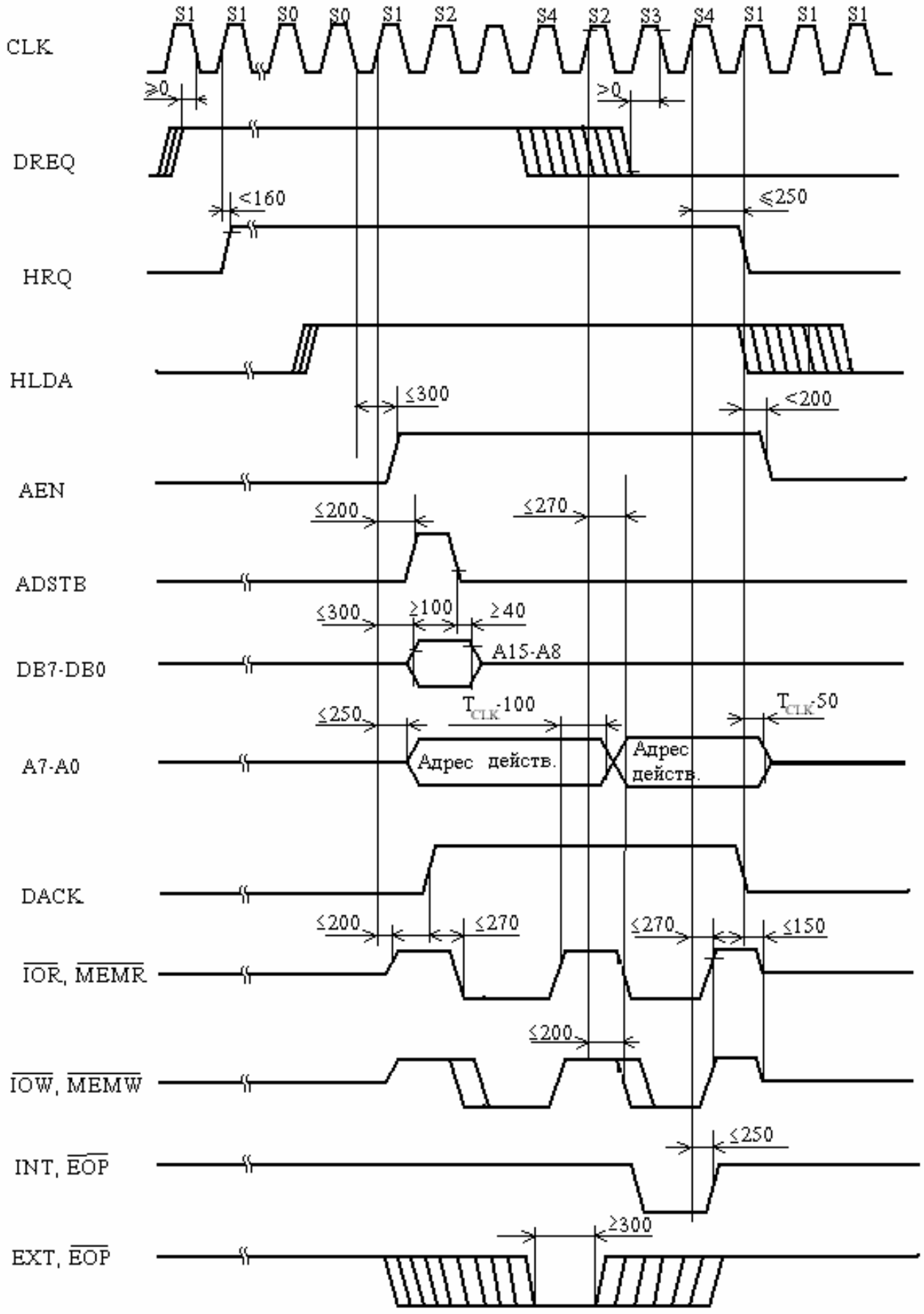


Рис. 3.46. Временная диаграмма работы КПДП

В этом режиме передача может осуществляться, пока внешнее устройство не закончит передачу информации. Автоинициализацию в этом режиме можно осуществлять после окончания передачи сигналом $\overline{\text{EOP}}$, внешним или вырабатываемым по признаку ТС.

Режим передачи память–память позволяет осуществлять перемещение блоков информации в поле оперативной памяти. Для реализации этого режима используются параметры каналов 0 и 1. Передача инициализируется программно установкой DREQ в канале 0. После прихода сигнала HLDA=1 контроллер за четыре такта считывает данные из ячейки памяти с адресом из регистра CAR канала 0 и записывает их в регистр временного хранения TR, а затем за четыре такта записывает эти данные в ячейку памяти с адресом из CAR канала 1. Когда содержимое регистра циклов ПДП CWR примет значение FFFFH, установится разряд ТС и передача закончится. Канал 0 может быть запрограммирован на передачу информации без изменения адреса, что позволяет заполнить ячейки блока ОЗУ константой.

Типы передачи ПДП. Во всех режимах ПДП возможны три основных типа передачи. *Запись данных* – осуществляется передача данных от внешнего устройства к ОЗУ. Контроллер в этом случае активизирует сигналы $\overline{\text{MEMW}}$ и $\overline{\text{IOR}}$. *Чтение данных* – осуществляется передача данных от ОЗУ к внешнему устройству, активизируются сигналы $\overline{\text{MEMR}}$ и $\overline{\text{IOW}}$. В случае *проверки или псевдопередачи* контроллер выполняет действия такие же, как в цикле чтения/записи, но сигналы управления не вырабатываются. В этом случае сигнал READY не воспринимается. Кроме того, контроллер может быть запрограммирован для выполнения дополнительных функций.

Автоинициализация осуществляется, если установлен соответствующий разряд в регистре условий, и по сигналу $\overline{\text{EOP}}$. При автоинициализации содержимое базовых регистров BAR и WCR загружается в регистры текущих значений CAR и CWR. Разряды маски при этом не меняются. После автоинициализации контроллер готов к работе и возобновляет действие с приходом очередного сигна-

ла DREQ. Для автоинициализации обоих каналов в режиме память–память регистры циклов ПДП CWR должны программироваться идентично.

Контроллер может быть запрограммирован для обслуживания каналов с жестко *заданными приоритетами* либо с их *циклическим изменением*. При жестко заданном приоритете наивысший приоритет устанавливается за каналом с меньшим номером. При циклическом изменении самый низкий приоритет присваивается каналу после его обслуживания. Это позволяет обслужить все каналы поочередно.

Таблица 3.3

A3	A2	A1	A0	Команда	Операция
1	0	0	0	Ввод	Чтение регистра состояния
1	0	0	0	Вывод	Запись в регистр команд управления
1	0	0	1	То же	Запись в регистр запросов
1	0	1	0	>>	Установка всех разрядов маски
1	0	1	1	>>	Запись в регистр режима
1	1	0	0	>>	Установка режима ввода младшего байта
1	1	0	1	Ввод	Чтение регистра временного хранения
1	1	0	1	Вывод	Общий сброс
1	1	1	0	То же	Сброс всех разрядов маски
1	1	1	1	>>	Установка разряда маски

Программирование контроллера. Программирование контроллера осуществляется от ЦП командами ввода–вывода и возможно только в пассивном состоянии или при наличии на входе HLDA напряжения низкого уровня, если даже присутствует сигнал HRQ. Начальную инициализацию контроллера необходимо осуществить сразу же после включения напряжения питания по всем каналам (если даже они не используются), загружая команды и константы. Адреса внутренних регистров контроллера определяются кодом на выводах A3–A0. В табл.3.3 показаны коды на A3–A0, соответствующие выполняемым командам ЦП, а в табл.3.4 – коды на A3–A0, соответствующие адресам регистров КПДП.

Таблица 3.4

A3	A2	A1	A0	Команда	Операция
0	0	0	0	Вывод	Загрузка мл/ст байта в рег BAR и CAR канала 0
0	0	0	0	Ввод	Чтение содержимого CAR канала 0
0	0	0	1	Вывод	Загрузка мл/ст байта в рег WCR и CWR канала 0
0	0	0	1	Ввод	Чтение содержимого CWR канала 0
0	0	1	0	Вывод	Загрузка мл/ст байта в рег BAR и CAR канала 1
0	0	1	0	Ввод	Чтение содержимого CAR канала 1
0	0	1	1	Вывод	Загрузка мл/ст байта в рег WCR и CWR канала 1
0	0	1	1	Ввод	Чтение содержимого CWR канала 1
0	1	0	0	Вывод	Загрузка мл/ст байта в рег BAR и CAR канала 2
0	1	0	0	Ввод	Чтение содержимого CAR канала 2
0	1	0	1	Вывод	Загрузка мл/ст байта в рег WCR и CWR канала 2
0	1	0	1	Ввод	Чтение содержимого CWR канала 2
0	1	1	0	Вывод	Загрузка мл/ст байта в рег BAR и CAR канала 3
0	1	1	0	Ввод	Чтение содержимого CAR канала 3
0	1	1	1	Вывод	Загрузка мл/ст байта в рег WCR и CWR канала 3
0	1	1	1	Ввод	Чтение содержимого CWR канала 3

Подключение контроллера к системной шине (рис. 3.47). Восемь старших разрядов адреса выдаются на ШД и должны быть записаны сигналом ADSTB во внешний регистр. Линия AEN используется для того, чтобы разряды адреса остались действующими на ША в течение трех тактовых периодов цикла ПДП. Линии A7 – A0 подключаются непосредственно к ША. Сигналы MEMR, MEMW, IOR, IOW управляют в циклах ПДП соответственно ОЗУ и буфером ВУ.

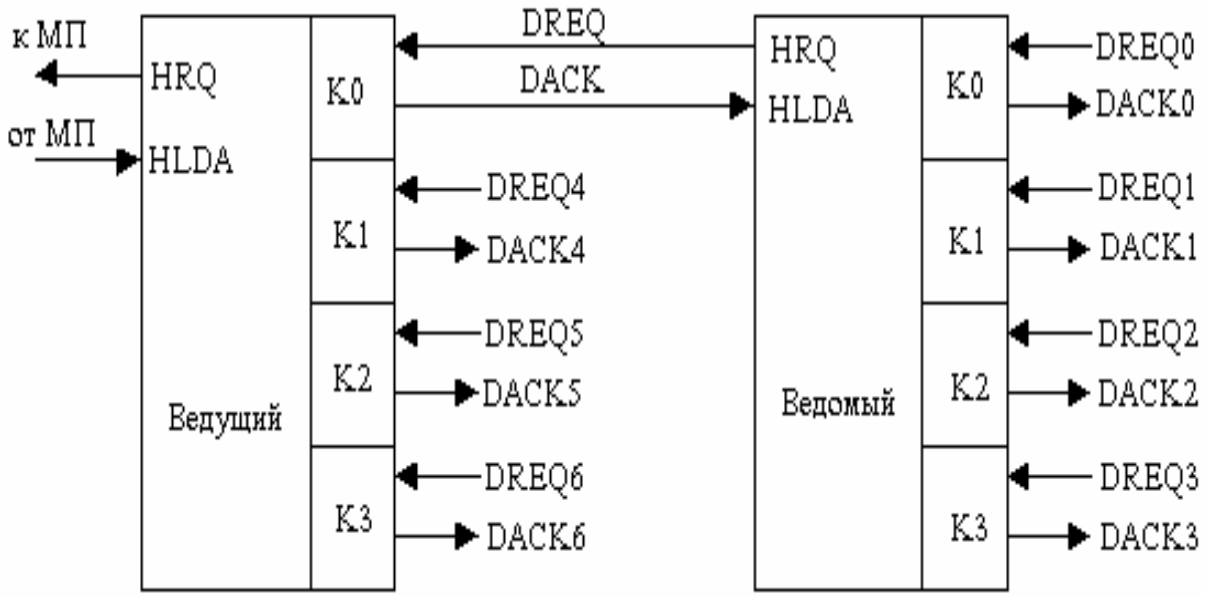


Рис. 3.47. Схема каскадирования КПДП

Каскадирование КПДП. Примером каскадного использования КПДП может быть IBM PC/AT, в котором к шине адреса со смещением на 1 байт подключили второй 8237A (рис. 3.48). Его 16-битные регистры адреса способны управлять линиями адреса A16 – A1 (младший байт A0 всегда 0). Таким образом, второй КПДП обеспечивает передачу по два байта. Вторая микросхема 8237A подключена как ведущая и создает три 16-битных канала ПДП.

Вопросы и задания

3.37. В каких режимах работают КПДП и его функции в системе?

3.38. Определите программно-доступные регистры и их адресацию.

3.39. Составьте схему подключения К1810ВТ37 к шинам адреса и данных МПС.

3.40. Составьте программу инициализации контроллера для блочного обмена по одному каналу.

3.41. Поясните реализацию режима ПДП на ВТ37.

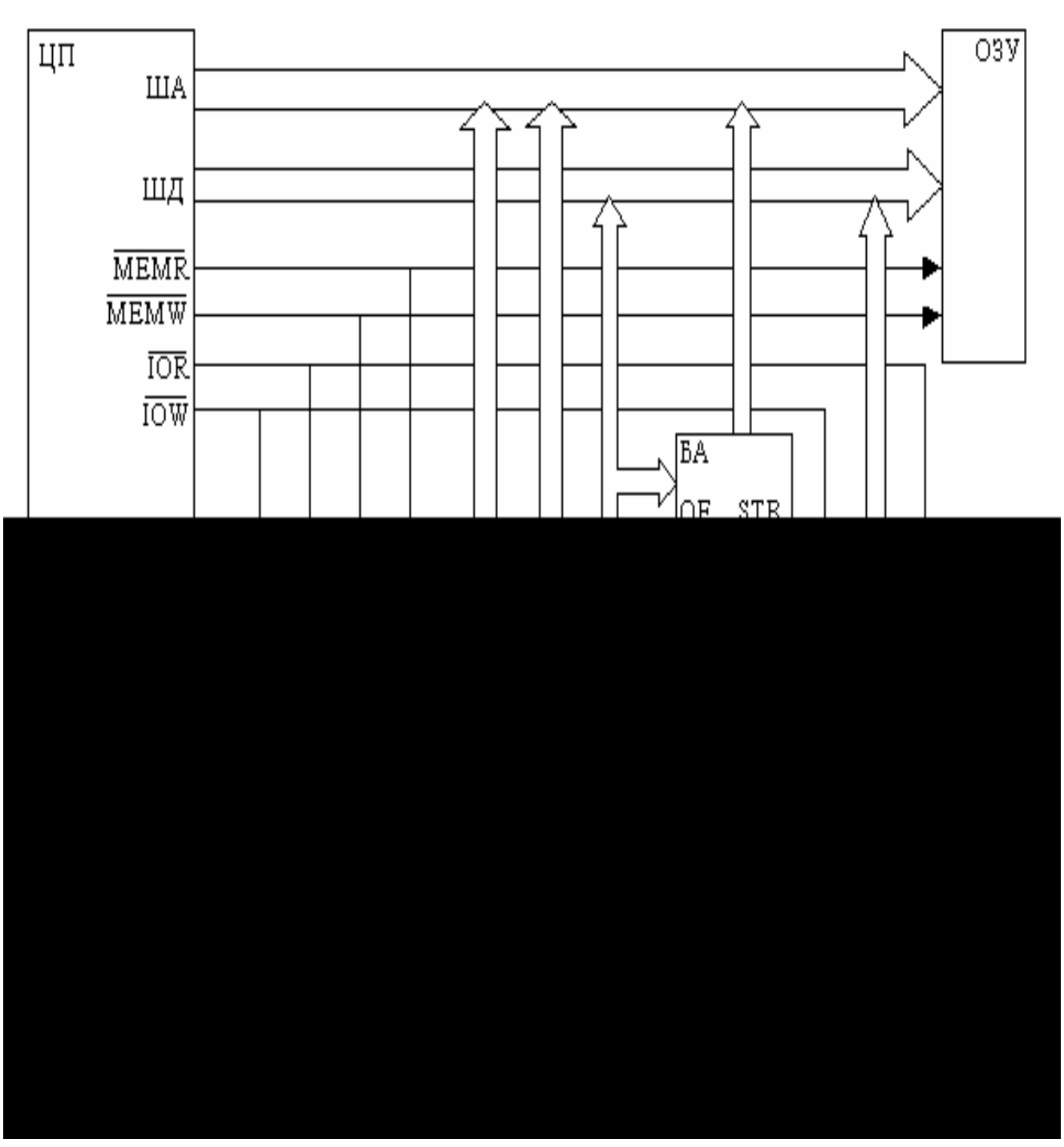
3.42. Поясните адресацию к регистрам КПДП при программировании.

3.43. Как формируется 16-битный адрес КПДП при управлении обменом?

3.44. Поясните предоставление ПДП по запросам с ведомого КПДП.

3.45. Какие приоритеты запросов поддерживает 8237 (ВТ37)?

3.46. В какой последовательности необходимо производить загрузку регистров контроллера при его программировании?



4. ЭЛЕМЕНТЫ ПРОГРАММИРОВАНИЯ

Целью данного раздела является более детальное изучение вычислительных возможностей микропроцессоров методом исследования программной реализации различного рода операций.

На начальном этапе предлагается ручное ассемблирование с определением машинных кодов, что позволяет детально изучить

- принцип кодирования команд;
- форматы команд и данных;
- способы адресации данных и команд;
- систему команд микропроцессора;
- программную модель микропроцессора и МПС;
- программные модели интерфейсных БИС и контроллеров устройств.

В качестве примера оформления рассмотрим программу суммирования содержимого ячейки 8400h и порта 01h с выводом суммы в порт 02h и запоминанием в ячейке 8401h.

В табл. 4.1 представлена программа, выполняющая определение указанной суммы. Таблица для записи программы имеет следующие поля: адреса, кода команды, ассемблерного кода (мнемокода) и комментариев. Поле метки используется при наличии команд передачи управления и в данной программе остается пустым.

Поле мнемокода содержит символьное обозначение кода команды на языке ассемблера.

Поле операнда содержит информацию о регистрах, данных и адресах, объединенных соответствующей операцией. Программа ассемблер может выдать соответствующий машинный код, используя только поля мнемокода и операнда. Списку команд можно назначить ячейки памяти, как это сделано в табл. 4.1.

Таблица 4.1

Адрес (h-код)	Машинный код (h-код)	Метка	Мнемокод	Операнд	Комментарий
8500 8501 8502	21 00 84		LXI	H, 8400h	Загрузка указателя адреса, т.е. 8400h → HL
8503 8504	DB 01		IN	01	Ввод числа из порта 01 в аккумулятор
8505	86		ADD	M	Сложение содержимого ячейки М и регистра А, т.е. (A)+(M) → A
8506	23		INX	H	Увеличить указатель адреса, т.е. (HL)+1 → HL
8507			MOV	M, A	Заполнить сумму в ячейке 8401
8508 8509	D3 02		OUT	02	Вывод суммы в порт 02,т.е. (A) → 02
850A	76		HLT		Останов ЭВМ

Поле комментариев не учитывается ассемблером и ограничивается его перепечаткой. Это поле очень важно для понимания процессов и событий в программе.

После составления программы на ассемблере мнемокоды и операнды можно ручным способом перевести на машинный язык, используя систему команд выбранного микропроцессора. Каждому байту кода команды и операнду назначить последовательно ячейки памяти. Эту операцию определения машинных кодов и назначения ячеек памяти может выполнить ассемблер.

Но для достижения вышеуказанных целей на начальных этапах изучения микропроцессоров и их программирования рекомендуется эти преобразования выполнять вручную.

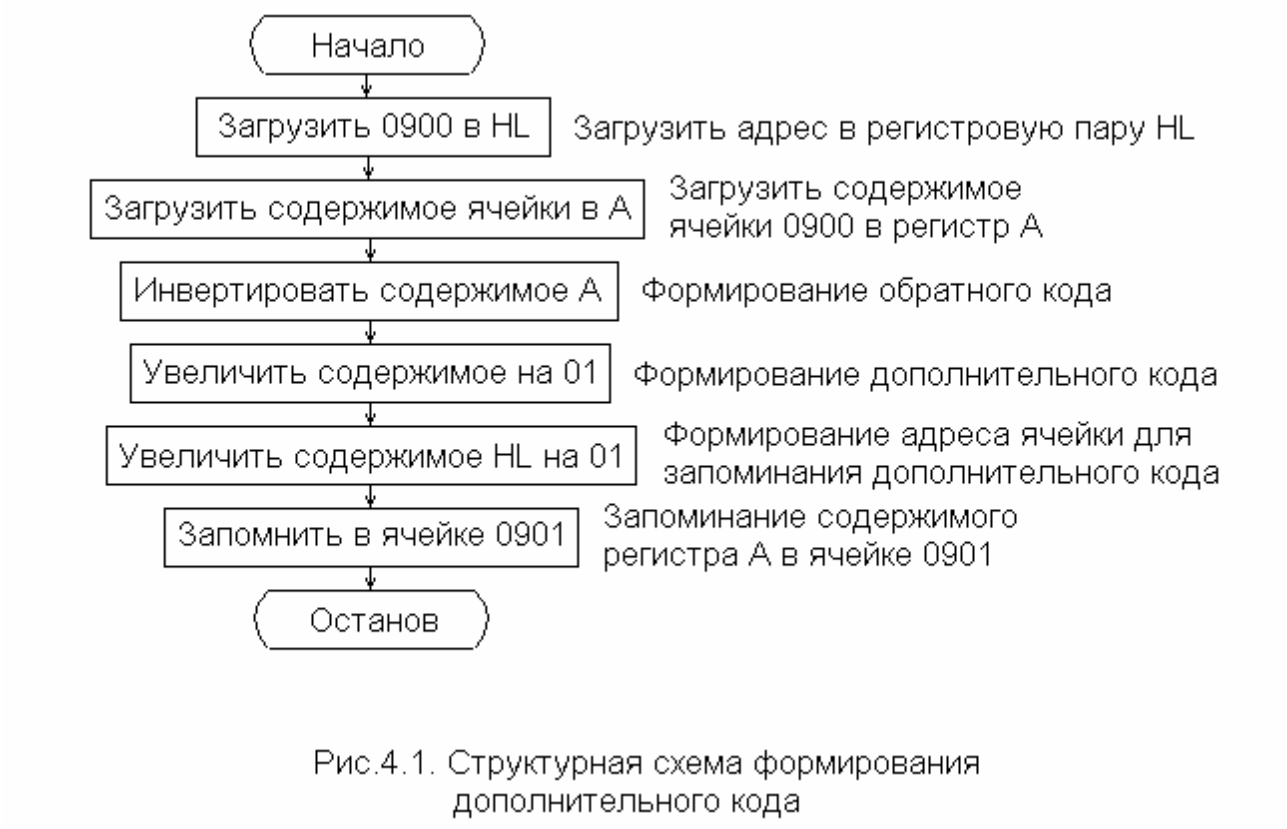


Рис.4.1. Структурная схема формирования дополнительного кода

4.1. Последовательные программы

Последовательной называют программу, в которой нет ветвлений. Для примера рассмотрим программу формирования дополнительного кода числа, считываемого из ячейки 0900h. Структурная схема программы формирования дополнительного кода приведена на рис. 4.1.

Задания

4.1. Разработайте алгоритм и программу формирования дополнительного кода.

4.2. Разработайте программы, выполняющие над числами в ячейках 0902 и 0903 следующие операции: а) сложение; б) вычитание;

в) сложение с предварительным умножением слагаемых на два;

г) вычитание с предварительным делением чисел на два;

д) сложение с получением результата в двоично-десятичном коде.

Примечание. Результаты вычислений разместите в ячейках памяти.

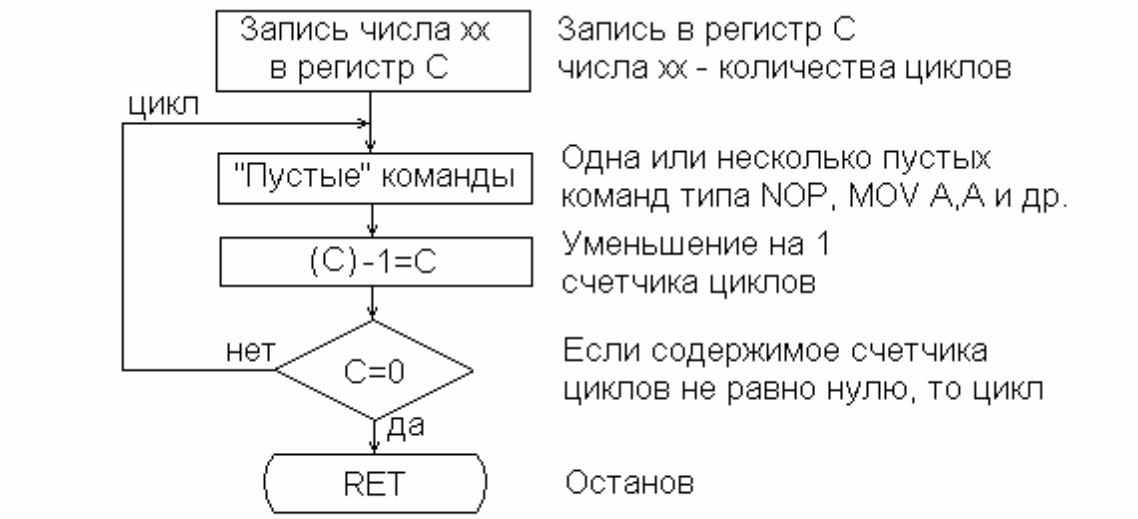


Рис.4.2. Схема формирования программной задержки

4.2. Циклические программы

Для организации циклов в программах в качестве счетчиков используются регистры A, B, C, D, E, H, L, M. Можно использовать и регистровые пары в качестве счетчиков, но команды INX, DCX не формируют признаков, что несколько затрудняет формирование ветвлений в программах по содержимому регистровых пар.

На рис. 4.2 приведена схема формирования программной задержки, использующая выполнение набора пустых команд в циклах. Счетчиком циклов является регистр C.

Задания

4.3. Разработайте подпрограмму задержки в соответствии с алгоритмом (рис.4.2).

4.4. Разработайте программную задержку на 1 с, применяя вложенные циклы.

4.3. Операции с массивами данных

Рассмотрим типовую процедуру сбора и формирования в ОЗУ МП-системы массива данных, вводимых из порта ввода IPORT. Начальный адрес

массива данных – BASE, а регистр C используется как счетчик данных. Регистровую пару HL удобно использовать как указатель данных. Схема процедуры сбора данных приведена на рис.4.3.

Задания

4.5. Разработайте программу сбора данных в соответствии с рис. 4.3 и организации 100 байтного массива в ОЗУ.

4.6. Введите в программу пункта 4.5 обработку массива:

а) нахождение суммы массива (без учета возможного переполнения);

б) нахождение максимального числа в массиве данных, представленных как целое без знака.

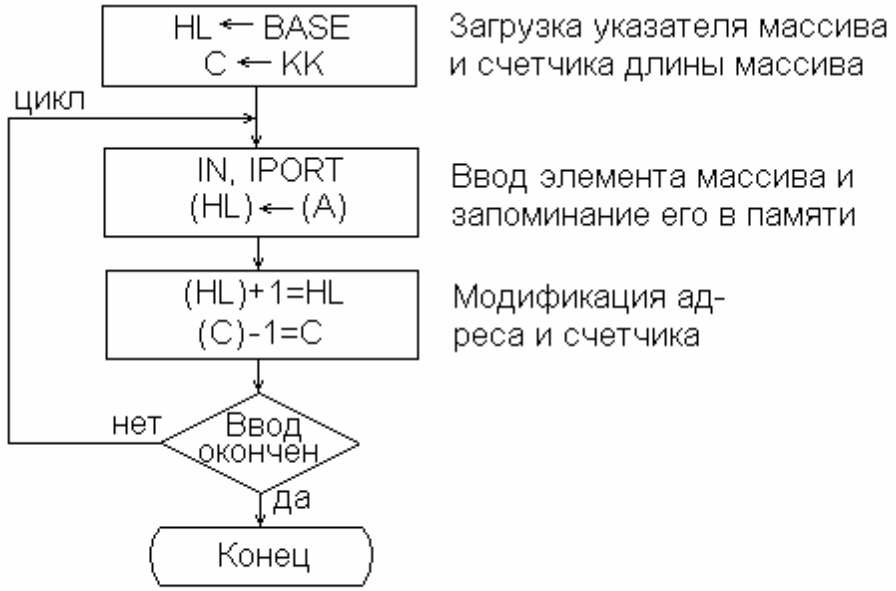


Рис.4.3. Схема процедуры выбора данных

4.4. Программная реализация типовых функций управления

При управлении оборудованием и технологическими процессами часто необходимо опрашивать состояние двоичного датчика, ожидать определенного события в объекте управления, формировать временные задержки и управляющие сигналы.

Рассмотрим ожидание замыкания контакта **К** в результате возникновения определенных событий на объекте управления рис. 4.4. Контакт двоичного дат-

чика **К** подключен к порту ввода линии D3. Программа имеет символическое имя WAIT (ожидание) и может быть многократно использована основной управляющей программой контроллера по команде CALL WAIT.

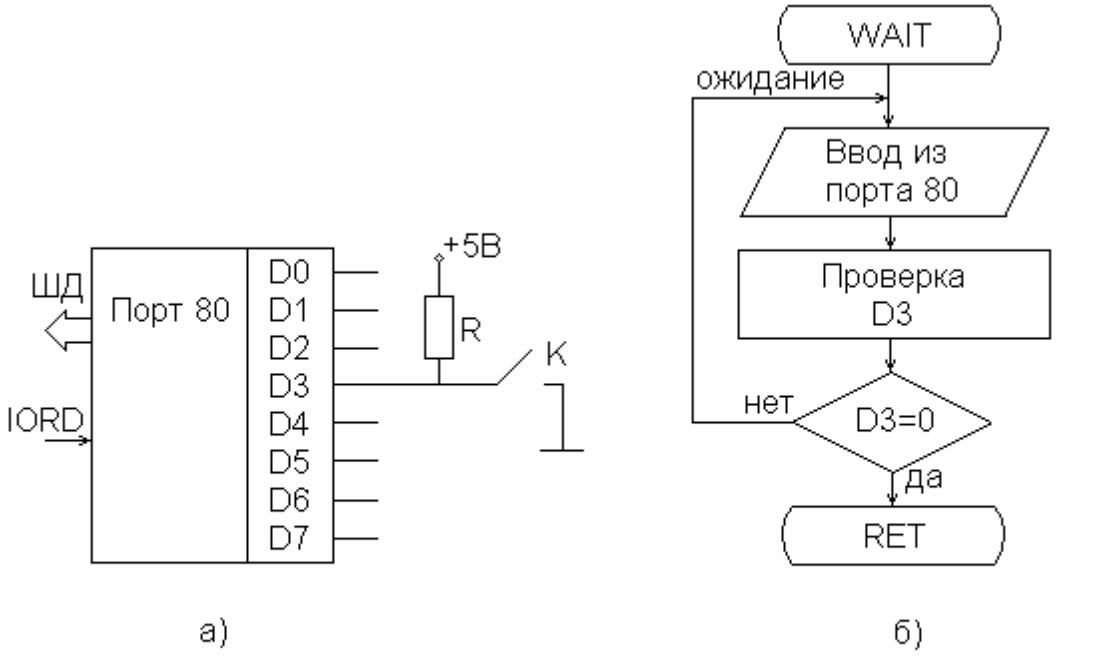


Рис.4.4. Схема (а) и алгоритм (б) ожидания события

Текст программы WAIT

```
WAIT:   IN 80h;      ввод в регистр А из порта 80.
        ANI 08h;     маскирование всех разрядов.
        JNZ WAIT;    переход в цикле ожидания, если D3=1.
        RET;         возврат в основную программу, если D=0.
```

Задания

4.7. Разработайте алгоритм и программу опроса двоичного датчика.

4.8. Разработайте схему подключения и программу формирования управляющего сигнала для включения и выключения осветительной лампы.

4.9. Разработайте схему и программу формирования последовательности импульсов.

4.10. Разработайте схему подключения и программу управления RS-триггером.

4.11. Разработате схему подключений и программу, определяющую факт замыкания контакта К (см. рис. 4.4), используя запрос на прерывание.

4.5. Моделирование функций алгебры логики

Моделирование функций алгебры логики (ФАЛ) заданной *полной таблицей истинности*, в которой всем набором входных переменных (X_4, X_3, X_2, X_1) ставится в соответствие вектор выходных значений ФАЛ (Y_8, Y_7, Y_6, Y_5, Y_4, Y_3, Y_2, Y_1). В качестве примера рассмотрим систему ФАЛ заданных в табл. 4.2, систему восьми ФАЛ для четырех входных переменных.

Метод программной реализации ФАЛ, заданных полной таблицей истинности:

- в память МПС записывается массив векторов выходных значений ФАЛ для всех 2^n наборов значений входных переменных (в порядке возрастания номера набора);
- считываемое извне значение входных переменных интерпретируется как номер строки в массиве, суммируется с начальным адресом массива и образует адрес строки выходных значений ФАЛ.

Метод моделирования ФАЛ, заданной СДНФ:

- в памяти МПС формируется массив входных наборов термов, на которых ФАЛ принимают значение 1;
- после получения текущего входного набора обслуживающая программа ведет поиск терма, равного текущему входному набору и, в случае обнаружения равного, присваивает ФАЛ значение 1. А если равенства не будет обнаружено для всех термов в массиве, то ФАЛ присваивается значение 0.

Таблица 4.2

Входные переменные				Выходные значения ФАЛ							
X4	X3	X2	X1	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1
0	0	0	0	0	1	1	0	1	1	1	1
0	0	0	1	1	0	1	0	0	1	1	1
0	0	1	0	1	0	0	1	1	0	1	1
0	0	1	1	0	1	0	0	0	1	0	1

0	1	0	0	1	0	1	0	0	0	1	0
0	1	0	1	0	1	1	0	1	1	1	0
0	1	1	0	0	1	0	0	0	1	1	0
0	1	1	1	1	0	1	1	1	1	1	0
1	0	0	0	1	1	0	0	1	1	1	1
1	0	0	1	0	0	0	1	0	1	1	0
1	0	1	0	1	0	0	0	1	0	0	0
1	0	1	1	0	1	1	0	1	0	1	0
1	1	0	0	1	1	0	1	0	0	1	0
1	1	0	1	0	0	1	1	1	0	0	1
1	1	1	0	1	0	0	0	1	1	1	1
1	1	1	1	1	1	0	1	0	1	0	1

Задания

4.12. Разработайте программу табличного варианта моделирования ФАЛ, заданных в табл. 4.2. Входные переменные введите из порта 80h или ячейки памяти 8700h.

4.13. Разработайте программу моделирования одной ФАЛ, заданной в табл. 4.2, используя метод сравнения текущего набора переменных с единичными наборами заданной ФАЛ. Переменные введите из порта 80h или ячейки памяти 8700h.

ПРИЛОЖЕНИЕ 1

КОНТРОЛЬНЫЕ ЗАДАНИЯ

По курсу «Микропроцессорные средства» студенты-заочники выполняют четыре контрольных задания. Для определения номера варианта задания необходимо номер зачетной книжки разделить нацело на 20. Остаток от деления принять за номер варианта.

Каждое контрольное задание должно иметь титульный лист, на котором указывается номер контрольного задания, номер зачетной книжки, фамилия и номер академической группы.

Для выполнения контрольных заданий можно пользоваться любой литературой и не только той, которая рекомендована для выполнения задания. При оформлении отчета по контрольной работе ссылка на литературу с указанием страниц обязательна.

Контрольное задание 1

Разработать структурную схему центрального процессорного модуля и подсистем микроЭВМ, указанных в таблице вариантов – П1.

- Выбрать необходимые микросхемы.
- Составить структурную схему микроЭВМ.
- Выполнить краткое описание функций, выполняемых подсистемами, и входящих в них БИС.
- Предлагаемые типы микропроцессоров: К580ВМ80, 8080, К1810ВМ86, К1810ВМ88, К1821ВМ85, 8085, Z80, К1856ВМ1, К580ВМ1, К1816ВЕ51, К1816ВЕ31 или другие микропроцессоры и микроконтроллеры, МП БИС, выбранные вами самостоятельно.

Примечание. Центральный процессорный модуль содержит микропроцессор, генератор тактов, шинные формирователи.

Пример выполнения задания 1

Задание. Разработать подсистему памяти программ емкостью 4 Кбайта к однокристальной микроЭВМ КМ1816ВЕ48 с использованием стандартных перепрограммируемых постоянных запоминающих устройств.

На рис. П.1 изображена схема подключения памяти программ к однокристальной микроЭВМ КМ1816ВЕ48 с использованием стандартных перепрограммируемых постоянных запоминающих устройств.

Схема состоит из микросхем:

- однокристальной микроЭВМ КМ1816ВЕ48;
- БИС ППЗУ типа К573РФ1 - 3 шт.;
- дешифратора на микросхеме К155ИД4;
- многорежимного буферного регистра К589ИР12.

Однокристальная микроЭВМ КМ1816ВЕ48 содержит резидентную (внутреннюю) память программ емкостью 1 Кбайт, ОЗУ данных емкостью 64 байта, арифметико-логическое устройство, устройство управления, три порта ввода-вывода, внутренний таймер.

Для создания подсистемы ППЗУ емкостью 4 Кбайта применены три микросхемы типа К573РФ1, каждая из которых имеет емкость 1 Кбайт.

Выборка микросхем К573РФ1 осуществляется сигналами с дешифратора К155ИД4, на котором декодируется информация, поступающая по линиям Р23 и Р22 порта Р2 микросхемы КМ1816ВЕ48. Десятиразрядная шина адреса организована путем объединения восьми разрядов ДВ..ДВ7 порта В (порта BUS), поступающих через буферный регистр К589ИР12, и двух разрядов Р20 и Р21 порта Р2, поступающих через буферные усилители на К155ЛП4. Шина данных создана на двунаправленных шинных формирователях К589АП16, подключенных к порту В микросхемы КМ1816ВЕ48.

Таким образом, на трех микросхемах К573РФ1 и с учетом резидентной памяти микросхемы К1816ВЕ48 суммарная емкость ППЗУ составляет 4 Кбайта.

Таблица П.1

Номер варианта	Емкость ПЗУ Кбайт	Емкость ОЗУ Кбайт	Количество портов ввода-вывода		Счетчиков	Таймеров	Входов прерываний
			параллельных	последов-ных			
0	4	4	2	2	1	1	2
1	4	8	2	3	2	1	4
2	4	6	3	2	1	1	4
3	4	2	3	3	2	2	2

4	6	6	1	1	1	1	4
5	6	6	2	1	2	2	4
6	6	8	1	2	1	1	2
7	6	2	1	3	2	2	2
8	6	4	3	1	1	1	2
9	8	2	2	1	2	2	2
10	8	3	3	2	1	3	2
11	8	6	2	2	2	1	4
12	8	5	2	2	1	2	4
13	8	16	3	1	2	1	4
14	8	4	1	3	1	2	4
15	7	2	1	3	2	1	2
16	5	4	2	2	1	2	2
17	16	6	3	1	2	1	2
18	16	8	1	3	1	2	2
19	9	2	2	2	2	1	4

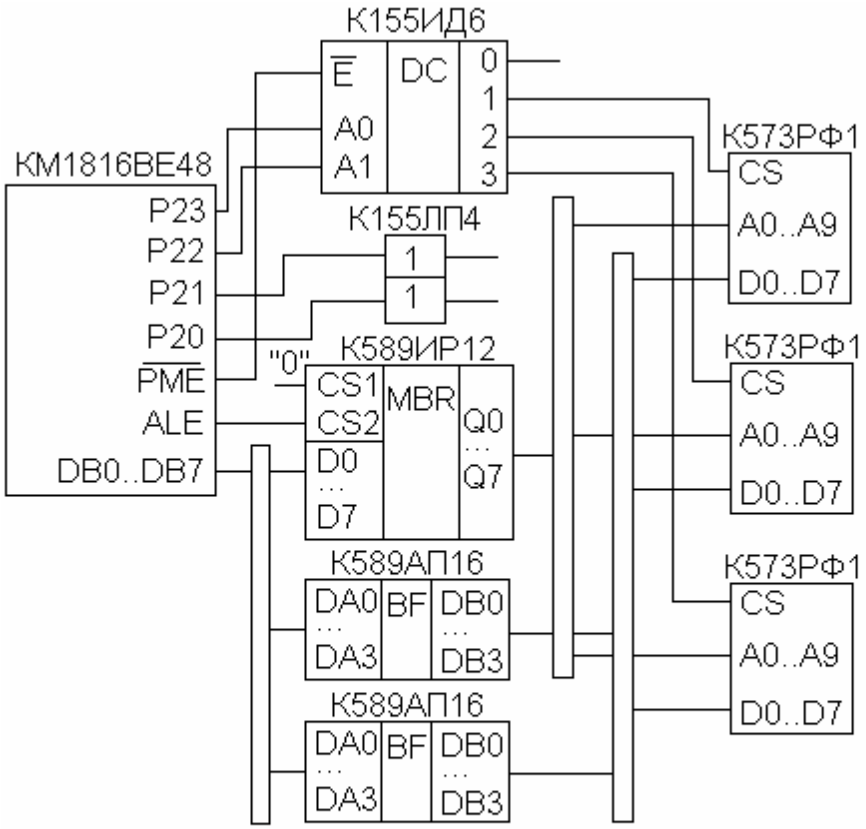


Рис.П.1. Структурная схема подсистемы ППЗУ емкостью 4 Кбайта

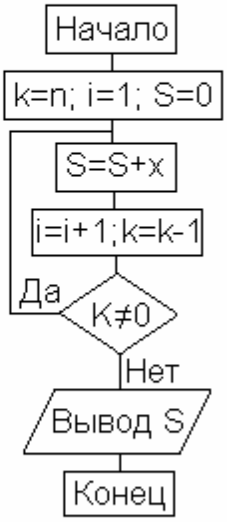


Рис.П.2.

Контрольное задание 2

Разработать алгоритм и программу на ассемблере и в машинных кодах микропроцессора КР580ВМ80 (или Z80, К1858ВМ1). Варианты заданий приведены в таблице П3.

Исходные данные и результаты вычислений можно размещать в регистрах общего назначения МП или в ячейках памяти по адресам 8400..8600h (h - обозначение шестнадцатиричной системы счисления).

Выбор исходных данных осуществляется самостоятельно.

Пример выполнения задания 2

Разработать алгоритм и программу вычисления суммы компонент вектора X по формуле $S= \sum X_i$ (i=1, .. , n). Блок-схема алгоритма задачи приведена на рис. П1, программа приведена в табл. П2.

Пусть: $S= \sum (1+2+3+4+5+6+7)$.

Распределение входных данных и результата в памяти.

8460-X1=1 8464-X5=5

8461-X2=2 8465-X6=6

8462-X3=3 8466-X7=7

8463-X4=4 8467-S=28

Таблица П2

Адрес ячеек памяти	Код	Метка	Мнемокод	Комментарий
8400	06 07		MVI B 07	Загрузка счетчика
8402	97		SUB A	Обнуление аккумулятора
8403	21 60 84		LXI H 84 60	Загрузка указателя адреса
8406	86	MI	ADD M	Сложение содержимого аккумулятора и ячейки 8460
8407	23		INX H	Инкремент указателя адреса
8408	05		DCR B	Декремент счетчика
8409	C2 06 84		JNZ 8406	Повторение при K≠0
840C	77		MOV M, A	Засылка результата в ячейку памяти
840D	76		HLT	Конец

Таблица П3

Вариант	Формулировка задачи
0	Вычисление эелементов вектора Z по элементам векторов X и Y $Z_i=X_i+Y_i$, где i=1, .. , n
1	Вычисление эелементов вектора Z по элементам векторов X и Y $Z_i=X_i-Y_i$, где i=1, .. , n
2	Вычисление эелементов вектора Z по элементам векторов X и Y $Z_i=X_i+Y_i+1$, где i=1, .. , n
3	Вычисление эелементов вектора Z по элементам векторов X и Y $Z_i=X_i-Y_i+1$, где i=1, .. , n
4	Вычисление суммы компонент S по формуле $S=\sum(x_i+1)$
5	Вычисление элементов вектора Z по элементов векторов X и Y $Z_i=\max(X_i, Y_i)$, где i=1, .. , n
6	Вычисление элементов вектора Z по элементам вектора X по формуле $Z=\begin{cases} X_i, \text{ если } X_i<0 \\ 0, \text{ если } X_i\geq 0, \text{ где i=1, .. , n} \end{cases}$
7	Вычисление элементов вектора Z по элементам векторов X и Y $Z_i=\min(X_i, Y_i)$, где i=1, .. , n
8	Вычисление элементов вектора Z по элементам вектора X по формуле $Z=\begin{cases} 4, \text{ если X=4} \\ 5, \text{ если X не равно 4, где i=1, .. , n} \end{cases}$
9	Вычисление элементов вектора Z по элементам вектора X по формуле $Z=\begin{cases} X, \text{ если } X_i=Y_i \\ 0, \text{ если } X_i \text{ не равно } Y_i, \text{ где i=1, .. , n} \end{cases}$
10	Вычисление элементов вектора Z по элементам векторов X и Y . $Z\ i = (2X_i,+ 2Y_i)$, где i=1, .. , n
11	Вычисление элементов вектора Z по элементам векторов X и Y . $Z\ i = (0.5X_i,+ 0.5Y_i)$, где i=1, .. , n
12	Вычисление элементов вектора Z по элементам векторов X и Y . $Z\ i = (X_i,+ 2Y_i)$, где i=1, .. , n
13	Вычислить сумму положительных элементов вектора Z $S= \sum Z_i$, если $Z_i>0$, где i=1, .. , n
14	Вычислить сумму отрицательных элементов вектора Z $S= \sum Z_i$, если $Z_i<0$, где i=1, .. , n
15	Организация временной задержки T=1сек. программным способом.

	Длительность такта МП равна 0,5 мкс.
16	Вычисление элементов вектора Z по элементам векторов X и Y $Z_i=2X_i+Y_i$, где i=1, .. , n
17	Вычисление элементов вектора Z по элементам векторов X и Y $Z_i=X_i - 0,5Y_i$, где i=1, .. , n
18	Вычисление элементов вектора Z по элементам векторов X и Y $Z_i=0,5X_i+Y_i+1$, где i=1, .. , n
19	Вычисление элементов вектора Z по элементам векторов X и Y $Z_i=0.5X_i-0,5Y_i+1$, где i=1, .. , n
20	Вычисление суммы компонент S по формуле $S= \sum (0,5x_i+1)$
21	Вычисление элементов вектора Z по элементам векторов X и Y $Z_i=\max(X_i, 0,5Y_i)$, где i=1, .. , n
22	Вычисление элементов вектора Z по элементам векторов X и Y $Z_i=\max(2X_i, 0,5Y_i)$, где i=1, .. , n

Контрольное задание 3

Разработка функциональной схемы простейшего микропроцессорного устройства (контроллера):

- Выбор основных микросхем.
- Распределение адресных пространств памяти и устройств ввода-вывода.
- Разработка принципиальных схем дешифраторов выборки модулей памяти и устройств ввода-вывода.
- Оформление функциональной схемы контроллера и краткого описания его компонентов.

Исходные данные индивидуально формулируются преподавателем или используются те же, что и для задания 1.

Контрольное задание 4

Создание минимального программного обеспечения контроллера, разработанного в задании 3:

- Разработка пусковой и тестирующей программы модулей ПЗУ и ОЗУ контроллера;
- Выбор режимов программируемых интерфейсных БИС и их краткое описание;
- Разработка программ инициализации интерфейсных БИС на выбранные режимы.

ПРИЛОЖЕНИЕ 2

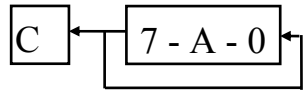
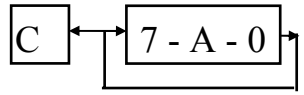
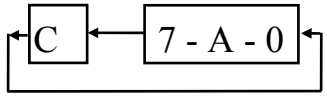
Команды микропроцессора КР580ВМ80А

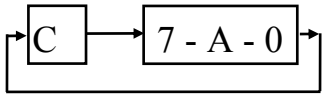
№	Мнемоника команды	Код первого байта команды	Б	Ц	Т	Операции	Пояснение
Пересылки однобайтовые							Все признаки результата не изменяются.
1	MOV r_d, r_s	01dddsss	1	1	5	$(r_s) \rightarrow (r_d)$	Переслать в РОН с прямой регистровой адресацией.
2	MOV r, M	01ddd110	1	2	7	$(M) \rightarrow r$	Переслать из памяти с косвенной адресацией по (HL) в РОН
3	MOV M, r	01110sss	1	2	7	$(r) \rightarrow M$	Переслать из РОН в память с косвенной адресацией по (HL)
4	MVI r, b2	00ddd110	2	2	7	$b2 \rightarrow r$	Переслать непосредственный операнд в РОН
5	MVI M, b2	00110110	2	3	10	$b2 \rightarrow M$	Переслать непосредственный операнд в память с косвенной адресацией по (HL)
6	STA b3b2	00110010	3	4	13	$(A) \rightarrow b3b2$	Переслать содержимое аккумулятора в память по прямому адресу b3b2
7	LDA b3b2	00111010	3	4	13	$(b3b2) \rightarrow A$	Переслать из памяти с прямой адресацией в аккумулятор
8	STAX rp	00dd0010	1	2	7	$(A) \rightarrow (rp)$	Переслать из аккумулятора в память с косвенной адресацией по (rp); $rp \neq H$
9	LDAX rp	00ss1010	1	2	7	$((rp)) \rightarrow A$	Переслать из памяти с косвенной адресацией по (rp) в аккумулятор; $rp \neq H$

№	Мнемоника команды	Код первого байта команды	Б	Ц	Т	Операции	Пояснение
10	OUT b2	11010011	2	3	10	(A) → b2	Переслать из аккумулятора в порт вывода с номером b2
11	IN b2	11011011	2	3	10	(b2) → A	Переслать из порта ввода с номером b2 в аккумулятор.
Пересылки двухбайтовые							Все признаки результата не изменяются (кроме 21)
12	LXI rp,b3b2	00dd0001	3	3	10	b2 → $rp_{мл}$ b3 → $rp_{ст}$	Загрузить непосредственный операнд b3b2 в регистровую пару rp
13	LXI SP, b3b2	00110001	3	3	10	b3b2 → SP	Загрузить в указатель стека код b3b2
14	SHLD b3b2	00100010	3	5	16	(L) → b3b2 (H) → b3b2 + 1	Переслать содержимое HL в память с прямой адресацией
15	LHLD b3b2	00101010	3	5	16	(b3b2) → L (b3b2 + 1) → H	Переслать из памяти с прямой адресацией в пару HL
16	XCHG	11101011	1	1	4	H ↔ D L ↔ E	Обмен содержимым пары HL и DE
17	SPHL	11111001	1	1	5	(HL) → SP	Переслать содержимое пары HL в указатель стека
18	PUSH rp	11ss0101	1	3	11	$(rp)_{ст}$→ (SP) – 1 $(rp)_{мл}$→ (SP) – 2 (SP) – 2 → SP	Загрузить в стек содержимое регистровой пары rp

№	Мнемоника команды	Код первого байта команды	Б	Ц	Т	Операции	Пояснение
19	PUSH PSW	11110101	1	3	11	(A) → (SP) – 1 (F) → (SP) – 2 (SP) – 2 → SP	Загрузить слово состояния процессора PSW в стек
20	POP rp	11dd0001	1	3	10	((SP)) → $r_{мл}$ ((SP) + 1 → $rp_{ст}$ (SP) + 2 → SP	Извлечь содержимое верхушки стека в пару rp
21	POP PSW	11110001	1	3	10	((SP)) → F ((SP) + 1 → A (SP) + 2 → SP	Извлечь содержимое верхушки стека в регистры F, A. Изменяются все признаки результата
22	XTHL	11100011	1	5	18	(L) ↔ ((SP)) (H) ↔ ((SP) + 1)	Обмен содержимым пары HL и верхушки стека. Значение SP не изменяется
Арифметические операции в аккумуляторе							Все признаки результата изменяются
23	ADD r	10000sss	1	1	4	(A) + (r) → A	Сложить содержимое аккумулятора и РОН
24	ADD M	10000110	1	2	7	(A) + (M) → A	Сложить с памятью, косвенно адресуемой по (HL)
25	ADI b2	11000110	2	2	7	(A) + b2 → A	Сложить с непосредственным операндом
26	ADC r	1000 1sss	1	1	4	(A) + (r) + (CY) → A	Сложить с РОН и с битом CY (переносом)
27	ADC M	1000 1110	1	2	7	(A) + (M) + (CY) → A	Сложить с памятью и с битом CY

№	Мнемоника команды	Код первого байта команды	Б	Ц	Т	Операции	Пояснение
28	ACI b2	1100 1110	2	2	7	(A) + b2 + (CY) → A	Сложить с непосредственным операндом и с битом CY
29	SUB r	1001 0sss	1	1	4	(A) – (r) → A	Вычесть содержимое РОН. При заёме бит CY: = 1
30	SUB M	1001 0110	1	2	7	(A) – (M) → A	Вычесть содержимое памяти, косвенно адресуемой по (HL)
31	SUI b2	1101 0110	2	2	7	(A) – b2 → A	Вычесть непосредственный операнд
32	SBB r	1001 1sss	1	1	4	(A) – (r) – (CY) → A	Вычесть содержимое РОН и бит CY (заём)
33	SBB M	1001 1110	1	2	7	(A) – (M) – (CY) →A	Вычесть содержимое памяти, косвенно адресуемой по (HL), и бит CY
34	SBI b2	1101 1110	2	2	7	(A) – b2 – (CY) → A	Вычесть непосредственный операнд и бит CY
35	CMP r	1011 1sss	1	1	4	(A) – (r)	Сравнить содержимое аккумулятора и РОН. Разность никуда не записывается
36	CMP M	1011 1110	1	2	7	(A) – (M)	Сравнить содержимое аккумулятора и памяти, косвенно адресуемой по (HL)
37	CPI b2	1111 1110	2	2	7	(A) – b2	Сравнить содержимое аккумулятора с непосредственным операндом
38	DAA	0010 0111	1	1	4		Десятичная коррекция в аккумуляторе
Арифметические операции в РОН							Изменяются признаки S, Z, AC, P, а признак переноса CY не изменяется (кроме 43 – 45)
39	INR r	00dd d100	1	1	5	(r) + 1 → r	Инкремент РОН

№	Мнемоника команды	Код первого байта команды	Б	Ц	Т	Операции	Пояснение
40	DCR r	00dd d101	1	1	5	$(r) - 1 \rightarrow r$	Декремент РОН
41	INR M	0011 0100	1	3	10	$(M) + 1 \rightarrow M$	Инкремент памяти, косвенно адресуемой по (HL)
42	DCR M	0011 0101	1	3	10	$(M) - 1 \rightarrow M$	Декремент памяти, косвенно адресуемой по (HL)
43	INX rp	00dd 0011	1	1	5	$(rp) + 1 \rightarrow rp$	Инкремент пары rp. Признаки результата не изменяются
44	DCX rp	00dd 1011	1	1	5	$(rp) - 1 \rightarrow rp$	Декремент пары rp. Признаки результата не изменяются
45	DAD rp	00dd 1001	1	3	10	$(HL) + (rp) \rightarrow HL$	Сложить 2 – байтовые операнды. rp = B, D, H, SP. Изменяется только признак переноса CY
Сдвиги в аккумуляторе							
46	RLC	0000 0111	1	1	4	$CY \leftarrow (a_7)$ $(a_7) \rightarrow a_0$ $a_{i+1} \rightarrow (a_i)$	Сдвинуть циклически влево 
47	RRC	0000 1111	1	1	4	$CY \leftarrow (a_0)$ $(a_0) \rightarrow a_7$ $(a_{i+1}) \rightarrow (a_i)$	Сдвинуть циклически вправо 
48	RAL	0001 0111	1	1	4	$CY \leftarrow (a_7)$ $(CY) \rightarrow a_0$ $a_{i+1} \leftarrow (a_i)$	Сдвинуть циклически влево через бит переноса CY 

№	Мнемоника команды	Код первого байта команды	Б	Ц	Т	Операции	Пояснение
49	RAR	0001 1111	1	1	4	$(a_0) \rightarrow CY$ $(CY) \rightarrow a_7$ $(a_{i+1}) \rightarrow a_i$	Сдвинуть циклически вправо через бит переноса CY 
Логические операции в аккумуляторе и регистре F							Изменяются признаки S, Z, P. Признак AC не изменяется. Признак CY устанавливается в 0. (Кроме 59 – 61)
50	ANA r	1010 0sss	1	1	4	$(A) \,\&\, (r) \rightarrow A$	Поразрядная конъюнкция с РОН
51	ANA M	1010 0110	1	2	7	$(A) \,\&\, (M) \rightarrow A$	Поразрядная конъюнкция с памятью, косвенно адресуемой по (HL)
52	ANI b2	1110 0110	2	2	7	$(A) \,\&\, b2 \rightarrow A$	Поразрядная конъюнкция с непосредственным операндом.
53	ORA r	1011 0sss	1	1	4	$(A) \vee (r) \rightarrow A$	Поразрядная дизъюнкция с РОН
54	ORA M	1011 0110	1	2	7	$(A) \vee (M) \rightarrow A$	Поразрядная дизъюнкция с памятью, косвенно адресуемой по (HL)
55	ORI b2	1111 0110	2	2	7	$(A) \vee b2 \rightarrow A$	Поразрядная дизъюнкция с непосредственным операндом
56	XRA r	1010 1sss	1	1	4	$(A) \oplus (r) \rightarrow A$	Поразрядно сложить по модулю 2 с РОН
57	XRA M	1010 1110	1	2	7	$(A) \oplus (M) \rightarrow A$	Поразрядно сложить по модулю 2 с памятью, косвенно адресуемой по (HL)
58	XRI b2	1110 1110	2	2	7	$(A) \oplus b2 \rightarrow A$	Поразрядно сложить по модулю 2 с непосредственным операндом

№	Мнемоника команды	Код первого байта команды	Б	Ц	Т	Операции	Пояснение
59	CMA	0010 1111	1	1	4	$(\overline{A}) \rightarrow A$	Инвертировать аккумулятор. Признаки результата не изменяются
60	STC	0011 0111	1	1	4	$1 \rightarrow CY$	Установить бит переноса CY = 1. Остальные признаки результата не изменяются
61	CMC	0011 1111	1	1	4	$(\overline{CY}) \rightarrow CY$	Инвертировать бит переноса CY. Остальные признаки результата не изменяются
Команды управления							Все признаки результата не изменяются
62	JMP b3b2	1100 0011	3	3	10	b3b2 → PC	Перейти безусловно по адресу b3b2
63	PCHL	1110 1001	1	1	5	(H) → $PC_{ст}$ (L) → $PC_{мл}$	Перейти безусловно по адресу, находящемуся в HL
64	Jcon b3b2	11CC C010	3	3	10	Если (условие) = да, то b3b2 → PC	Перейти по условию. Если условие CCC подтверждается, то переход, иначе – исполнение следующей команды
65	CALL b3b2	1100 1101	3	5	17	(PC) → стек b3b2 → PC	Адрес возврата сохранить в стеке и перейти на подпрограмму по адресу b3b2
66	Ccon b3b2	11CC C100	3	3 5	11 17	Если (условие) = да, то (PC) → стек, b3b2 → PC	Перейти на подпрограмму по условию. Если условие CCC подтверждается, то переход, иначе – исполнение следующей команды

№	Мнемоника команды	Код первого байта команды	Б	Ц	Т	Операции	Пояснение
67	RET	1100 1001	1	3	11	((SP)) → $PC_{мл}$ ((SP)) + 1 → $PC_{ст}$ (SP) + 2 → SP	Вернуться безусловно из подпрограммы или из программы обслуживания прерывания
68	Rcon	11CC C000	1	1 3	5 11	Если (условие) = да, то (стек) → PC	Вернуться по условию. Если условие CCC подтверждается, то возврат, иначе – исполнение следующей команды
69	RST n	11NN N111	1	1	11	(PC) → стек 0...0NNN000 → PC	Рестарт по вектору прерывания NNN. Адрес возврата сохранить в стеке и перейти по адресу 0000 0000 00NN N000
70	EI	1111 1011	1	1	4	РПР:=1	Разрешить прерывания вслед за следующей после EI командой
71	DI	1111 0011	1	1	4	РПР:=0	Запретить прерывания вслед за командой DI
72	NOP	0000 0000	1	1	4	Нет операций	«Пустая» команда; пропуск четырех тактов
73	HLT	0111 0110	1	1	7	СТОП:= 1 ОЖИД:= 1	Команды останова. В PC находится в адрес следующей команды. МП может воспринимать запросы прерывания и запросы к магистрали

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Микропроцессорные системы: Учебное пособие для вузов / Е. К. Александров [и др.]; Под общ. Ред. Д. В. Пузанкова. – СПб.: Политехника, 2002. – 935 с.
2. Ларионов А. М. [и др.]. Вычислительные комплексы, системы и сети. Л.: Энергоатомиздат. 1987. – 288 с.
3. Каган Б. Н. Основы проектирования микропроцессорных устройств автоматики М.: Энергоатомиздат, 1987. – 304 с.
4. Щелкунов Н. Н., Дианов А. П. Микропроцессорные средства и системы. М.: Радио и связь, 1989. – 288 с.
5. Григорьев В. Л. Програмное обеспечение микропроцессорных систем. М.: Энергоатом издат, 1983. – 208 с.
6. Токхайм Р. Микропроцессоры. Курс и упражнения. М.: Энергоатомиздат, 1987. – 336 с.
7. Григорьев В. Л. Микропроцессор i486. Архитектура и программирование. Кн. 1 Программная архитектура. Кн. 2. Аппаратная архитектура. Кн.3. Устройство с плавающей точкой. Кн.4. Справочник по системе команд. М.: ГРАНАЛ, 1993. – 346 с и 382 с.
8. Гук. М. Процессоры Intel: от 8086 до Pentium II. Спб.: Питер, 1998. – 224 с.
9. Григорьев В. Л. Программирование однокристальных микропроцессоров. М.: Энергоатомиздат, 1987. – 184 с.
10. Михальчук В .М. [и др.]. Микропроцессоры 80x86, Pentium. Архитектура, функционирование, программирование, оптимизация кода. Мн.: Битрикс, 1994. – 426 с.
11. Казаринов Ю. М. Микропроцессорный комплект К1810. Структура, программирование и применение. Справочная книга. М.: Высшая школа, 1990. – 269 с.

12. Сташин В. В. [и др.]. Проектирование цифровых устройств на однокристальных микроконтроллерах. М.: Энергоатомиздат, 1990. – 224 с.
13. Боборыкин А. В. [и др.]. Однокристальные микроЭВМ. Справочник. М.: МИКАП, 1994. – 400 с.
14. Шевкопляс Б. В. Микропроцессорные структуры. Инженерные решения. Справочник. М.: Радио и связь, 1990. – 512 с.
15. Козаченко В. Ф. Микроконтроллеры: руководство по применению 16-разрядных микроконтроллеров Intel MCS – 196/296 во встроенных системах управления. М.: ЭКОМ, 1997. – 668 с.
15. Гук М. Аппаратные средства PC. Энциклопедия . Спб.: ПитерКом, 1998. – 816 с.
16. Новиков Ю. В. Разработка устройств сопряжения для персонального компьютера типа IBM PC. Практическое пособие. М.: ЭКОМ., 1997. – 224 с.
17. Харп Г. [и др.]. Транспьютеры. Архитектура и прогрммное обеспечение. М.: Радио и связь, 1993. – 304 с.
18. Русак И. М. [и др.]. Технические средства ПЭВМ. Мн.: Высш. шк., 1996. – 504 с.
19. Фрир Дж. Построение вычислительных систем на базе перспективных микропроцессоров. М.: Мир, 1990. – 413 с.
20. Майоров В. Г. Гаврилов А. И. Практический курс программирования микропроцессорных систем. М.: Машиностроение, 1990. – 272 с.
21. Гуртовцев А. Л. Гудыменко С. В. Программы для микропроцессоров. Справочное пособие. Мн.: Высш. шк., 1989. – 352 с.
22. Хвощ С. Т. [и др.]. Микропроцессоры и микроЭВМ в системах автоматического управления. Справочник. Л.: Машиностроение, 1987. – 612 с.
24. Лихтцидлер Б. Я. [и др.]. Микропроцессоры и вычислительные устройства в радиотехнике. Учебное пособие. К.: Выща шк., 1988. – 272 с.
25. Федорков Б. Г. [и др.] Микросхемы ЦАП и АЦП: функционирование, параметры, применение. М.: Энергоатомиздат, 1990. – 320 с.

26.Шварце Х., Хольцгрехе Г. В. Использование компьютеров в регулировании и управлении. М.: Энергоатомиздат, 1990. – 176 с.

27. Лебедев О. Н. [и др.]. Изделия электронной техники. М.: Радио и связь, 1994. – 248 с.

28. Учебно-отладочное устройство «ЭЛЕКТРОНИКА 580». Принцип работы. Элементы программирования: Методические указания к лабораторному практикуму / В. П. Мокрецов, Д. Г. Матюнин. Екатеринбург: изд. УГТУ, 2002. 40 с.

ОГЛАВЛЕНИЕ

Учебное электронное текстовое издание

Мокрецов Василий Петрович

МИКРОПРОЦЕССОРЫ И МПС

Часть 1

Архитектура микропроцессора К580ВМ80. Организация МП-систем

Редактор *И.Г. Южакова*
Компьютерная верстка *А.А. Гребенщикова*

Рекомендовано РИС ГОУ ВПО УГТУ-УПИ
Разрешен к публикации 18.04.07.
Электронный формат – PDF
Формат 60×90 1/8

Издательство ГОУ ВПО УГТУ-УПИ
620002, Екатеринбург, ул. Мира, 19
e-mail: sh@uchdep.ustu.ru

Информационный портал
ГОУ ВПО УГТУ-УПИ
http://www.ustu.ru