

вых бит или вырабатывается сигнал ошибки паритета в схеме контроля четности, то принятые в текущем кадре биты данных игнорируются и контроллер ожидает поступления нового стартового бита.

Обмен данными с ВУ по последовательным линиям связи широко используется в микроЭВМ, особенно в тех случаях, когда не требуется высокой скорости обмена. Вместе с тем применение в них последовательных линий связи с ВУ обусловлено двумя важными причинами. Во-первых, последовательные линии связи просты по своей организации: два провода при симплексной и полудуплексной передаче и максимум четыре – при дуплексной. Во-вторых, в микроЭВМ используются внешние устройства, обмен с которыми необходимо вести в последовательном коде.

5.5. Способы обмена информацией в микропроцессорной системе

В ЭВМ применяются три режима ввода/вывода: программно-управляемый ВВ (называемый также программным или нефорсированным ВВ), ВВ по прерываниям (форсированный ВВ) и прямой доступ к памяти. Первый из них характеризуется тем, что инициирование и управление ВВ осуществляется программой, выполняемой процессором, а внешние устройства играют сравнительно пассивную роль и сигнализируют только о своем состоянии, в частности, о готовности к операциям ввода/вывода. Во втором режиме ВВ иницируется не процессором, а внешним устройством, генерирующим специальный сигнал прерывания. Реагируя на этот сигнал готовности устройства к передаче данных, процессор передает управление подпрограмме обслуживания устройства, вызвавшего прерывание. Действия, выполняемые этой подпрограммой, определяются пользователем, а непосредственными операциями ВВ управляет процессор. Наконец, в режиме прямого доступа к памяти, который используется, когда пропускной способности процессора недостаточно, действия процессора приостанавливаются, он отключается от системной шины и не участвует в передачах данных между основной памятью и быстродействующим ВУ. Заметим, что во всех вышеуказанных случаях основные действия, выполняемые на системной магистрали ЭВМ, подчиняются двум основным принципам [5, 6]:

1. В процессе взаимодействия любых двух устройств ЭВМ одно из них обязательно выполняет активную, управляющую роль и является

датчиком, второе оказывается управляемым, исполнителем. Чаще всего датчиком является процессор.

2. Другим важным принципом, заложенным в структуру интерфейса, является принцип квитирования (запроса-ответа): каждый управляющий сигнал, посланный датчиком, подтверждается сигналом исполнителя. При отсутствии ответного сигнала исполнителя в течение заданного интервала времени формируется так называемый тайм-аут, датчик фиксирует ошибку обмена и прекращает данную операцию.

5.5.1. Программно-управляемый ввод/вывод

Данный режим характеризуется тем, что все действия по вводу/выводу реализуются командами прикладной программы. Наиболее простыми эти действия оказываются для "всегда готовых" внешних устройств, например индикатора на светодиодах. При необходимости ВВ в соответствующем месте программы используются команды IN или OUT. Такая передача данных называется синхронным или безусловным ВВ.

Однако для большинства ВУ до выполнения операций ВВ надо убедиться в их готовности к обмену, т. е. ВВ является асинхронным. Общее состояние устройства характеризуется флагом готовности READY, называемым также флагом готовности/занятости (READY/BUSY). Иногда состояния готовности и занятости идентифицируются отдельными флагами READY и BUSY, входящими в слово состояния устройства.

Процессор проверяет флаг готовности с помощью одной или нескольких команд. Если флаг установлен, то иницируются собственно ввод или вывод одного или нескольких слов данных. Когда же флаг сброшен, процессор выполняет цикл из 2–3 команд с повторной проверкой флага READY до тех пор, пока устройство не будет готово к операциям ВВ (рис. 37). Данный цикл называется циклом ожидания готовности ВУ и реализуется в различных процессорах по-разному [2, 5, 6].

Основной недостаток программного ВВ связан с непроизводительными потерями времени процессора в циклах ожидания. К достоинствам следует отнести простоту его реализации, не требующей дополнительных аппаратных средств.



Рис. 37. Цикл программного ожидания готовности внешнего устройства

5.5.2. Организация прерываний в микроЭВМ

Одной из разновидностей программно-управляемого обмена данными с ВУ в микроЭВМ является обмен с прерыванием программы, отличающийся от асинхронного программно-управляемого обмена тем, что переход к выполнению команд, физически реализующих обмен данными, осуществляется с помощью специальных аппаратных средств [5, 6]. Команды обмена данными в этом случае выделяют в отдельный программный модуль – подпрограмму обработки прерывания. Задачей аппаратных средств обработки прерывания в процессоре микроЭВМ как раз и является приостановка выполнения одной программы (ее еще называют основной программой) и передача управления подпрограмме обработки прерывания. Действия, выполняемые при этом процессором, как правило, те же, что и при обращении к подпрограмме. Только при обращении к подпрограмме они инициируются командой, а при обработке прерывания – управляющим сигналом от ВУ, который называют "Запрос на прерывание" или "Требование прерывания".

Эта важная особенность обмена с прерыванием программы позволяет организовать обмен данными с ВУ в произвольные моменты времени, не зависящие от программы, выполняемой в микроЭВМ. Таким образом, появляется возможность обмена данными с ВУ в реальном масштабе времени, определяемом внешней по отношению к микроЭВМ средой. Обмен с прерыванием программы существенным обра-

зом экономит время процессора, затрачиваемое на обмен. Это происходит за счет того, что исчезает необходимость в организации программных циклов ожидания готовности ВУ (см. примеры 1 и 2, разд. 5.3), на выполнение которых тратится значительное время, особенно при обмене с медленными ВУ.

Прерывание программы по требованию ВУ не должно оказывать на прерванную программу никакого влияния, кроме увеличения времени ее выполнения за счет приостановки на время выполнения подпрограммы обработки прерывания. Поскольку для выполнения подпрограммы обработки прерывания используются различные регистры процессора (счетчик команд, регистр состояния и т. д.), то информацию, содержащуюся в них в момент прерывания, необходимо сохранить для последующего возврата в прерванную программу.

Обычно задача сохранения содержимого счетчика команд и регистра состояния процессора возлагается на аппаратные средства обработки прерывания. Сохранение содержимого других регистров процессора, используемых в подпрограмме обработки прерывания, производится непосредственно в подпрограмме. Отсюда следует достаточно очевидный факт: чем больший объем информации о прерванной программе сохраняется программным путем, тем больше время реакции микроЭВМ на сигнал прерывания, и наоборот. Предпочтительными с точки зрения повышения производительности микроЭВМ (сокращения времени выполнения подпрограмм обработки, а следовательно и основной программы) являются уменьшение числа команд, обеспечивающих сохранение информации о прерванной программе, и реализация этих функций аппаратными средствами.

Формирование сигналов прерываний – запросов ВУ на обслуживание – происходит в контроллерах соответствующих ВУ. В простейших случаях в качестве сигнала прерывания может использоваться сигнал "Готовность ВУ", поступающий из контроллера ВУ в системный интерфейс микроЭВМ. Однако такое простое решение обладает существенным недостатком – процессор не имеет возможности управлять прерываниями, т. е. разрешать или запрещать их для отдельных ВУ. В результате организация обмена данными в режиме прерывания с несколькими ВУ существенно усложняется.

Для решения этой проблемы регистр состояния и управления контроллера ВУ (рис. 38) дополняют еще одним разрядом – "Разрешение прерывания". Запись 1 или 0 в разряд "Разрешение прерывания" производится программным путем по одной из линий шины системного интерфейса. Управляющий сигнал системного интерфейса "Запрос на прерывание" формируется с помощью схемы совпадения толь-

ко при наличии единиц в разрядах "Готовность ВУ" и "Разрешение прерывания" регистра состояния и управления контроллера [2, 5, 6].

Аналогичным путем решается проблема управления прерываниями в микроЭВМ, в целом. Для этого в регистре состояния процессора выделяется разряд, содержимое которого определяет, разрешены или запрещены прерывания от внешних устройств. Значение этого разряда может устанавливаться программным путем.

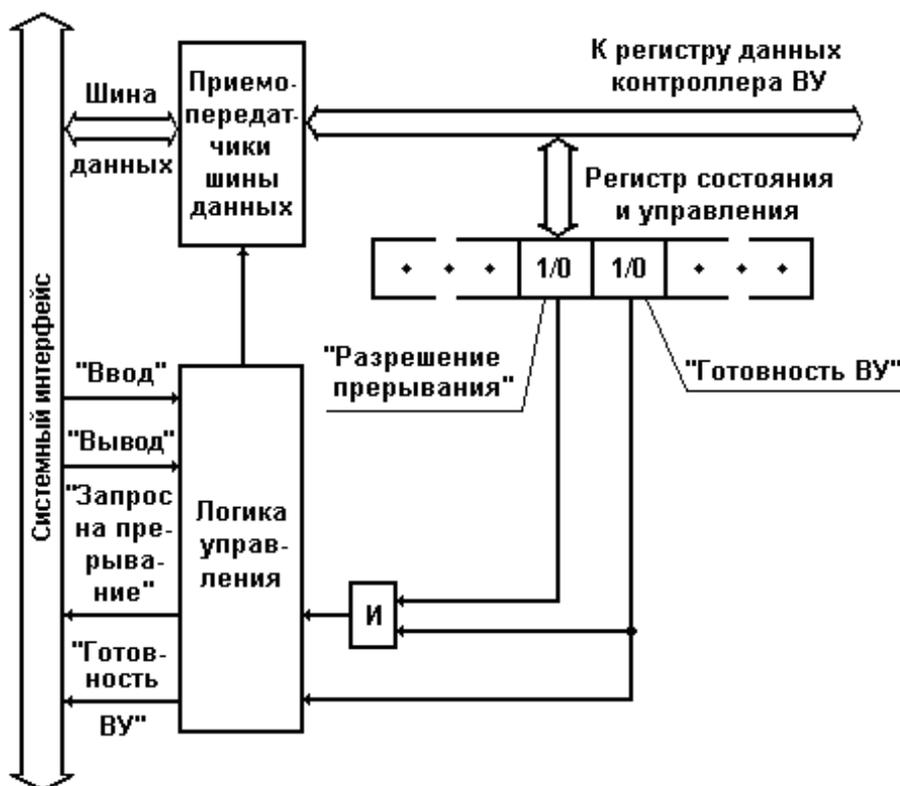


Рис. 38. Фрагмент блок-схемы контроллера ВУ с разрядом "Разрешение прерывания" в регистре состояния и управления

В микроЭВМ обычно используется одноуровневая система прерываний, т. е. сигналы "Запрос на прерывание" от всех ВУ поступают на один вход процессора. Поэтому возникает проблема идентификации ВУ, запросившего обслуживание, и реализации заданной очередности (приоритета) обслуживания ВУ при одновременном поступлении нескольких сигналов прерывания. Существуют два основных способа идентификации ВУ, запросивших обслуживания:

- программный опрос регистров состояния (разряд "Готовность ВУ") контроллеров всех ВУ;
- использование векторов прерывания.

Организация прерываний с программным опросом готовности предполагает наличие в памяти микроЭВМ единой подпрограммы обслуживания прерываний от всех внешних устройств. Структура такой подпрограммы приведена на рис. 39.

Обслуживание ВУ с помощью единой подпрограммы обработки прерываний производится следующим образом. В конце последнего машинного цикла выполнения очередной команды основной программы процессор проверяет наличие требования прерывания от ВУ. Если сигнал прерывания есть и в процессоре прерывание разрешено, то процессор переключается на выполнение подпрограммы обработки прерываний.

После сохранения содержимого регистров процессора, используемых в подпрограмме, начинается последовательный опрос регистров состояния контроллеров всех ВУ, работающих в режиме прерывания. Как только подпрограмма обнаружит готовое к обмену ВУ, сразу выполняются действия по его обслуживанию. Завершается подпрограмма обработки прерывания после опроса готовности всех ВУ и восстановления содержимого регистров процессора.

Приоритет ВУ в микроЭВМ с программным опросом готовности внешнего устройства однозначно определяется порядком их опроса в подпрограмме обработки прерываний. Чем раньше в подпрограмме опрашивается готовность ВУ, тем меньше время реакции на его запрос и выше приоритет. Необходимость проверки готовности всех внешних устройств существенно увеличивает время обслуживания тех ВУ, которые опрашиваются последними. Это является основным недостатком рассматриваемого способа организации прерываний. Поэтому обслуживание прерываний с опросом готовности ВУ используется только в тех случаях, когда отсутствуют жесткие требования на время обработки сигналов прерывания внешних устройств.

Организация системы прерываний в микроЭВМ с использованием векторов прерываний позволяет устранить указанный недостаток. При такой организации системы прерываний ВУ, запросившее обслуживания, само идентифицирует себя с помощью вектора прерывания – адреса ячейки основной памяти микроЭВМ, в которой хранится либо первая команда подпрограммы обслуживания прерывания данного ВУ, либо адрес начала такой подпрограммы.

Таким образом, процессор, получив вектор прерывания, сразу переключается на выполнение требуемой подпрограммы обработки прерывания. В микроЭВМ с векторной системой прерывания каждое ВУ должно иметь собственную подпрограмму обработки прерывания.

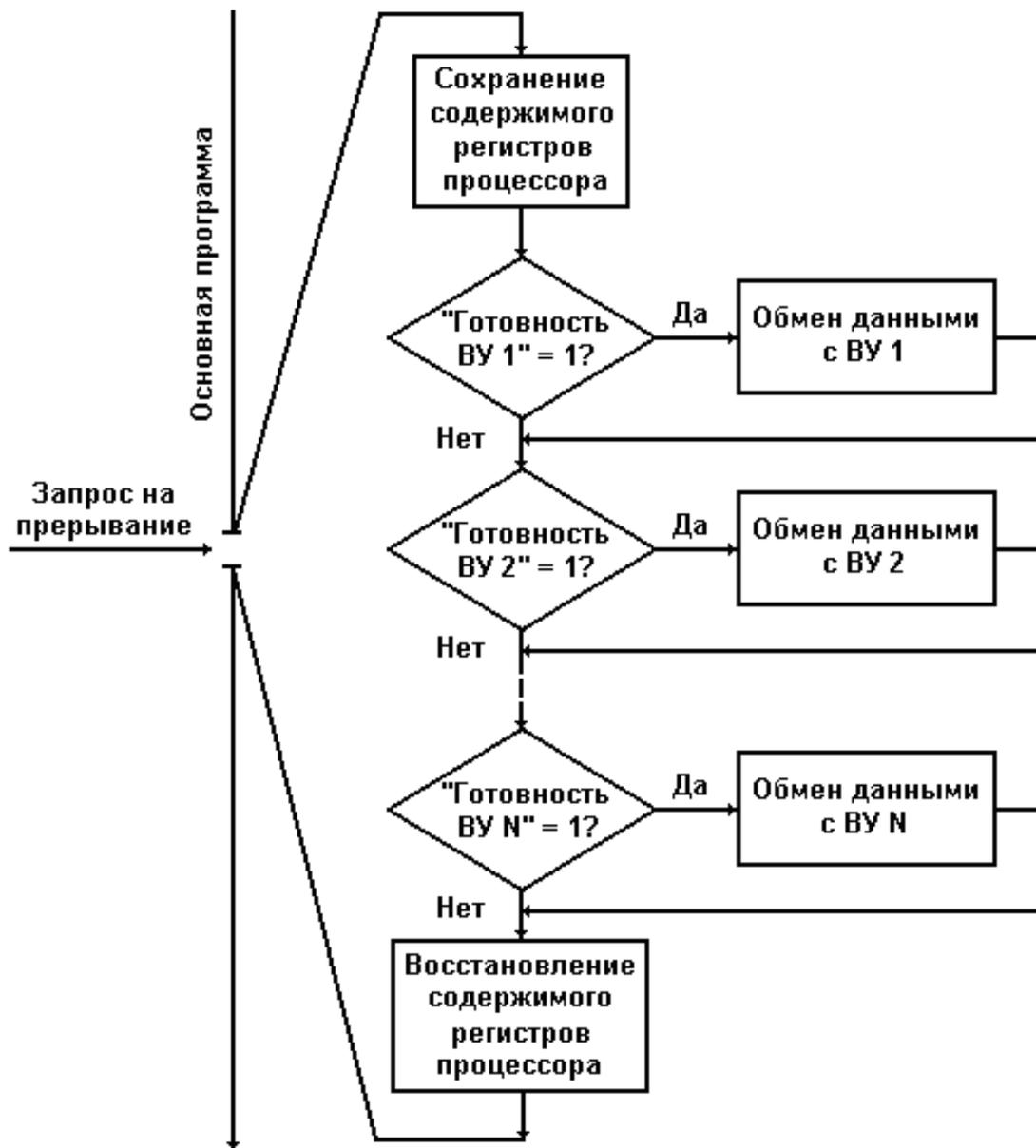


Рис. 39. Структура единой программы обработки прерываний и ее связь с основной программой

Различают векторные системы с интерфейсным и внеинтерфейсным вектором. В первом случае вектор прерывания формирует контроллер ВУ, запросившего обслуживания, во втором – контроллер прерываний, общий для всех устройств, работающих в режиме прерываний (IBM-совместимые персональные компьютеры).

Рассмотрим организацию векторной системы с интерфейсным вектором [5, 6]. Вектор прерывания выдается контроллером не одновременно с запросом на прерывание, а только по разрешению процессора, как это реализовано в схеме на рис. 40. Это делается для того, чтобы исключить одновременную выдачу векторов прерывания от нескольких ВУ. В ответ на сигнал контроллера ВУ "Запрос на прерывание" процессор формирует управляющий сигнал "Предоставление прерывания (вх.)", который разрешает контроллеру ВУ, запросившему обслуживание, выдачу вектора прерывания в шину адреса системного интерфейса. Для этого в контроллере используются регистр вектора прерывания и схема совпадения ИЗ. Регистр вектора прерывания обычно реализуется с помощью перемычек или переключателей, что позволяет пользователю устанавливать для конкретных ВУ требуемые значения векторов прерывания.

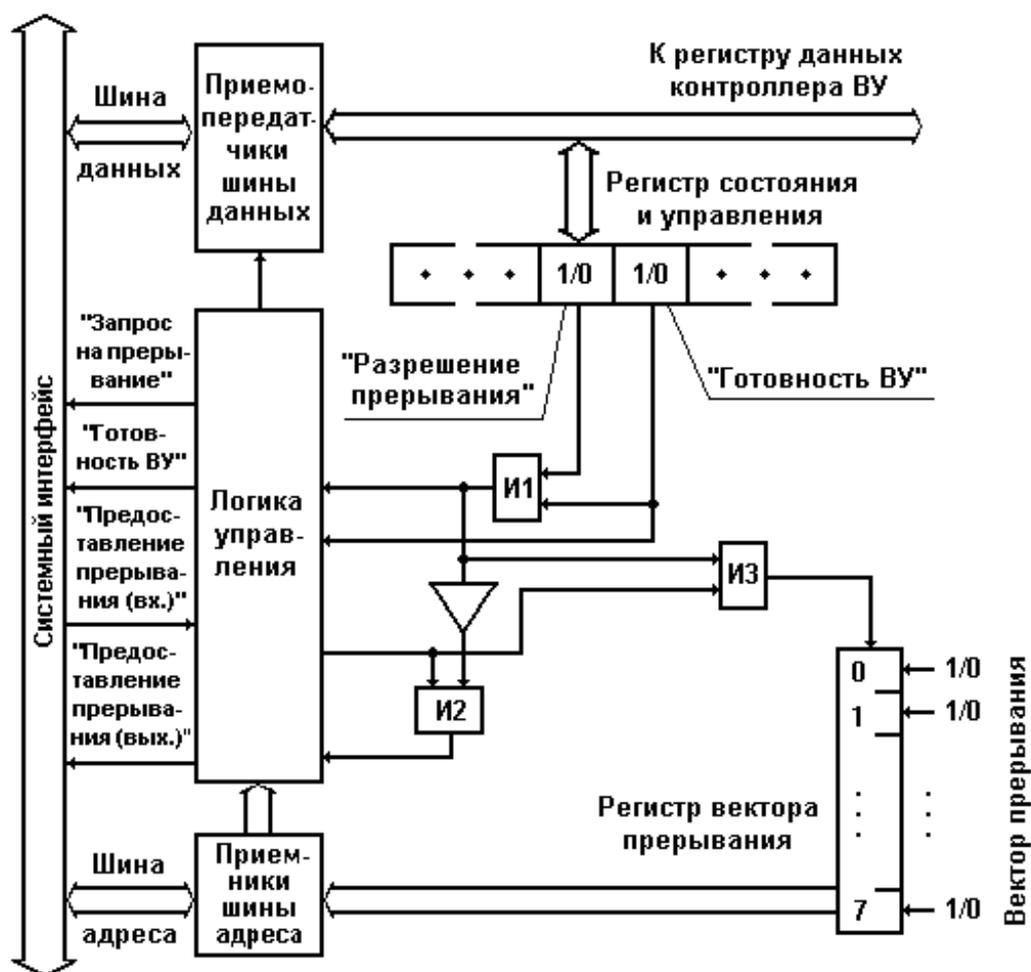


Рис. 40. Формирование векторов прерывания в контроллере ВУ

Управляющий сигнал "Предоставление прерывания (вых.)" формируется в контроллере ВУ с помощью схемы совпадения И2. Этот

сигнал используется для организации последовательного аппаратного опроса готовности ВУ и реализации тем самым требуемых приоритетов ВУ. Процессор при поступлении в него по общей линии системного интерфейса "Запрос на прерывание" сигнала прерывания формирует управляющий сигнал "Предоставление прерывания (вх.)", который поступает сначала в контроллер ВУ с наивысшим приоритетом (рис. 41). Если это устройство не требовало обслуживания, то его контроллер пропускает сигнал "Предоставление прерывания" на следующий контроллер, иначе дальнейшее распространение сигнала прекращается и контроллер выдает вектор прерывания на адресноинформационную шину.

Аппаратный опрос готовности ВУ производится гораздо быстрее, нежели программный. Но если обслуживания запросили одновременно два или более ВУ, обслуживание менее приоритетных ВУ будет отложено на время обслуживания более приоритетных, как и в системе прерывания с программным опросом.



Рис. 41. Реализация приоритетов ВУ в микроЭВМ с векторной системой прерываний. С интерфейсным вектором (ППР (вх.) – "Предоставление прерывания (входной)"; "ППР (вых.) – Предоставление прерывания (выходной)")

Рассмотренная векторная система прерываний практически полностью соответствует системе прерываний, реализованной в микроЭВМ "Электроника-60". Восемьразрядный вектор прерывания в "Электронике-60" указывает одну из ячеек памяти с адресами от 0 до 255, в которой размещается адрес начала подпрограммы обработки прерывания. В следующей за указанной вектором прерывания ячейке памяти хранится новое содержимое регистра состояния процессора, загружаемое в него при переключении на подпрограмму обработки прерывания. Один из бит нового содержимого регистра состояния процессора запрещает или разрешает прерывания от других ВУ, что позволяет ВУ с

более высоким приоритетом прерывать подпрограммы обслуживания ВУ с меньшим приоритетом и наоборот.

Векторная система с внеинтерфейсным вектором прерывания используется в IBM-совместимых персональных компьютерах. В этих компьютерах контроллеры внешних устройств не имеют регистров для хранения векторов прерывания, а для идентификации устройств, запросивших обслуживания, используется общий для всех ВУ контроллер прерываний. Ниже приведен пример контроллера прерываний INTEL 8259A.

БИС программируемого контроллера прерываний (ПКП) представляет собой устройство, реализующее до восьми уровней запросов на прерывания с возможностью программного маскирования и изменения порядка обслуживания прерываний. За счет каскадного включения БИС ПКП число уровней прерывания может быть расширено до 64 (в архитектуре персонального компьютера IBM PC AT).

Структурная схема ПКП приведена на рис. 42 [2, 5, 6].

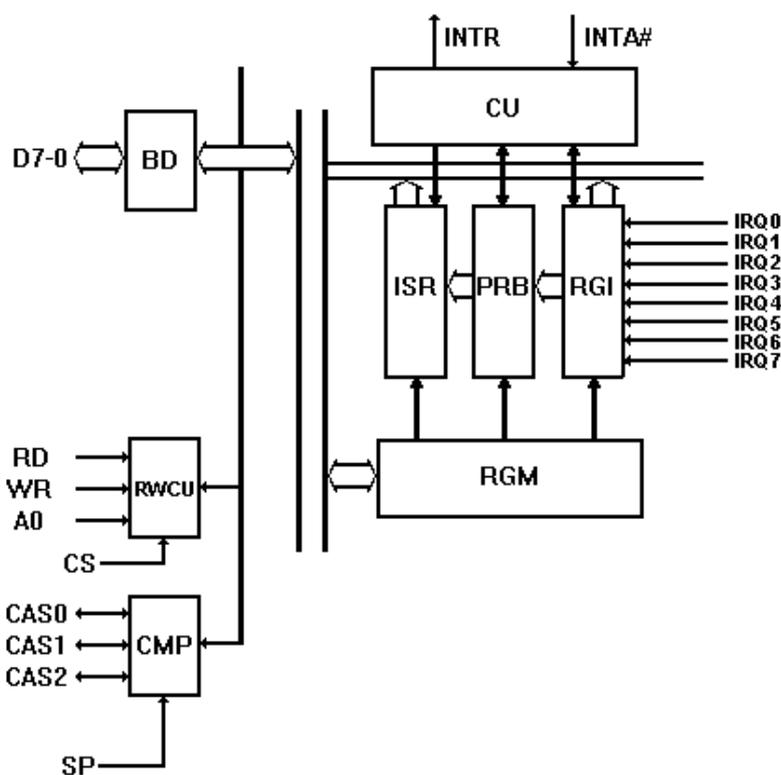


Рис. 42. Контроллер прерываний Intel 8259A

В состав БИС входят:

RGI – регистр запретов прерываний; хранит все уровни, на которые поступают запросы IRQx;

PRB – схема принятия решений по приоритетам; схема идентифицирует приоритет запросов и выбирает запрос с наивысшим приоритетом;

ISR – регистр обслуживаемых прерываний; сохраняет уровни запросов прерываний, находящиеся на обслуживании ПКП;

RGM – регистр маскирования прерываний; обеспечивает запрещение одной или нескольких линий запросов прерывания;

BD – буфер данных; предназначен для сопряжения ПКП с системной шиной данных;

RWCU – блок управления записью/чтением; принимает управляющие сигналы от микропроцессора и задает режим функционирования ПКП;

СMP – схема каскадного буфера-компаратора; используется для включения в систему нескольких ПКП;

CU – схема управления; вырабатывает сигналы прерывания и формирует трехбайтовую команду CALL для выдачи на шину данных.

Установка ПКП в исходное состояние и "настройка" его на определенный режим обслуживания прерываний происходит с помощью двух типов команд: команд инициализации (ICW) и команд управления операциями (OCW).

Программируемый контроллер прерываний (ПКП) имеет 16 входов запросов прерываний (IRQ 0 – IRQ 15). Контроллер состоит из двух каскадно включенных контроллеров – выход INTR (запрос на прерывание) второго контроллера подключен ко входу IRQ 2 первого контроллера. В качестве примера отметим, что к линии IRQ 0 подключен системный таймер, к линии IRQ 1 – клавиатура, к линии IRQ 8 – часы реального времени и т. д.

Упрощенная схема взаимодействия контроллера прерываний с процессором и контроллером шины имеет следующий вид (см. рис. 43).

Эта схема функционирует следующим образом. Пусть в некоторый момент времени контроллер клавиатуры с помощью единичного сигнала по линии IRQ 1 известил контроллер прерываний о своей готовности к обмену. В ответ на запрос контроллер прерываний генерирует сигнал INTR (запрос на прерывание) и посылает его на соответствующий вход процессора. Процессор, если маскируемые прерывания разрешены (т. е. установлен флаг прерываний IF в регистре флагов процессора), посылает на контроллер шины сигналы R# – чтение, C# – управление и IO# – ввод/вывод, определяющие тип цикла шины. Контроллер шины, в свою очередь, генерирует два сигнала подтверждения прерывания INTA# и направляет их на контроллер прерываний. По

второму импульсу контроллер прерываний выставляет на шину данных восьмибитный номер вектора прерывания, соответствующий данной линии IRQ.

В режиме реального адреса ("реальном" режиме) векторы прерываний хранятся в таблице векторов прерываний, которая находится в первом килобайте оперативной памяти. Под каждый вектор отведено 4 байта (2 байта под адрес сегмента и 2 байта под смещение), т. е. в таблице может содержаться 256 векторов. Адрес вектора в таблице – номер вектора * 4.

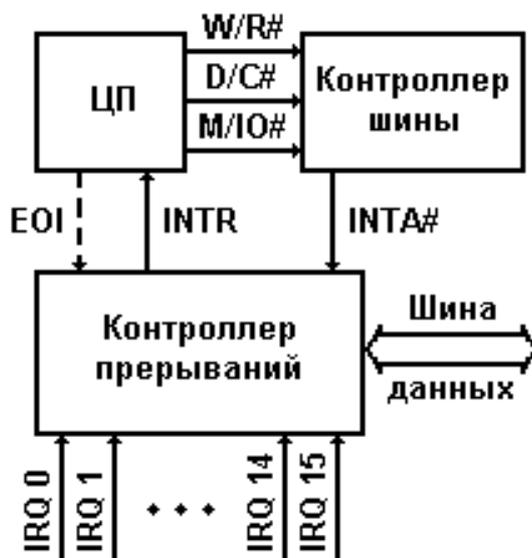


Рис. 43. Упрощенная схема взаимодействия контроллера прерываний с процессором и контроллером шины в IBM-совместимых ПК

Далее процессор считывает номер вектора прерывания. Сохраняет в стеке содержимое регистра флагов, сбрасывает флаг прерываний IF и помещает в стек адрес возврата в прерванную программу (регистры CS и IP). После этого процессор извлекает из таблицы векторов прерываний адрес подпрограммы обработки прерываний для данного устройства и приступает к ее выполнению.

Процедура обработки аппаратного прерывания должна завершаться командой конца прерывания EOI (End of Interruption), посылаемой контроллеру прерываний. Для этого необходимо записать байт 20h в порт 20h (для первого контроллера) и в порт A0h (для второго).

В IBM PC/XT/AT используется режим прерываний с фиксированными приоритетами. Высшим приоритетом обладает запрос по линии

IRQ 0, низшим – IRQ 7. Так как второй контроллер подключен к линии IRQ 2 первого контроллера, то приоритеты линий IRQ в порядке убывания приоритета располагаются следующим образом: IRQ 0, IRQ 1, IRQ 8 – IRQ 15, IRQ 3 – IRQ 7. Если запрос на обслуживание посылают одновременно два устройства с разными приоритетами, то контроллер обслуживает запрос с большим приоритетом, а запрос с меньшим приоритетом блокирует. Блокировка сохраняется до получения команды EOI.

5.5.3. Организация прямого доступа к памяти

Одним из способов обмена данными с ВУ является обмен в режиме прямого доступа к памяти (ПДП, DMA – direct memory access). В этом режиме обмен данными между ВУ и основной памятью микроЭВМ происходит без участия процессора. Обменом в режиме ПДП управляет не программа, выполняемая процессором, а электронные схемы, внешние по отношению к процессору. Обычно схемы, управляющие обменом в режиме ПДП, размещаются в специальном контроллере, который называется контроллером прямого доступа к памяти [5, 6].

Обмен данными в режиме ПДП позволяет использовать в микроЭВМ быстродействующие внешние запоминающие устройства, такие, например, как накопители на жестких магнитных дисках, поскольку ПДП может обеспечить время обмена одним байтом данных между памятью и ВЗУ, равное циклу обращения к памяти.

Для реализации режима прямого доступа к памяти необходимо обеспечить непосредственную связь контроллера ПДП и памяти микроЭВМ. Для этой цели можно было бы использовать специально выделенные шины адреса и данных, связывающие контроллер ПДП с основной памятью. Но такое решение нельзя признать оптимальным, так как это приведет к значительному усложнению микроЭВМ в целом, особенно при подключении нескольких ВЗУ. В целях сокращения количества линий в шинах микроЭВМ контроллер ПДП подключается к памяти посредством шин адреса и данных системного интерфейса. При этом возникает проблема совместного использования шин системного интерфейса процессором и контроллером ПДП. Можно выделить два основных способа ее решения: реализация обмена в режиме ПДП с "захватом цикла" и в режиме ПДП с блокировкой процессора.

Существуют две разновидности прямого доступа к памяти с "захватом цикла". Наиболее простой способ организации ПДП состоит в том, что для обмена используются те машинные циклы процессора, в

которых он не обменивается данными с памятью. В такие циклы контроллер ПДП может обмениваться данными с памятью, не мешая работе процессора. Однако возникает необходимость выделения таких циклов, чтобы не произошло временного перекрытия обмена ПДП с операциями обмена, инициируемыми процессором. В некоторых процессорах формируется специальный управляющий сигнал, указывающий циклы, в которых процессор не обращается к системному интерфейсу. При использовании других процессоров для выделения таких циклов необходимо применение в контроллерах ПДП специальных селективных схем, что усложняет их конструкцию. Применение рассмотренного способа организации ПДП не снижает производительности микроЭВМ, но при этом обмен в режиме ПДП возможен только в случайные моменты времени одиночными байтами или словами.

Более распространенным является ПДП с "захватом цикла" и принудительным отключением процессора от шин системного интерфейса. Для реализации такого режима ПДП системный интерфейс микроЭВМ дополняется двумя линиями для передачи управляющих сигналов "Требование прямого доступа к памяти" (ТПДП, DRQ – Dma ReQuest) и "Предоставление прямого доступа к памяти" (ППДП, DACK – Dma ACKnowledged).

Управляющий сигнал ТПДП формируется контроллером прямого доступа к памяти. Процессор, получив этот сигнал, приостанавливает выполнение очередной команды, не дожидаясь ее завершения, выдает на системный интерфейс управляющий сигнал ППДП и отключается от шин системного интерфейса. С этого момента все шины системного интерфейса управляются контроллером ПДП. Контроллер ПДП, используя шины системного интерфейса, осуществляет обмен одним байтом или словом данных с памятью микроЭВМ и затем, сняв сигнал ТПДП, возвращает управление системным интерфейсом процессору. Как только контроллер ПДП будет готов к обмену следующим байтом, он вновь "захватывает" цикл процессора и т. д. В промежутках между сигналами ТПДП процессор продолжает выполнять команды программы. Тем самым выполнение программы замедляется, но в меньшей степени, чем при обмене в режиме прерываний.

Применение в микроЭВМ обмена данными с ВУ в режиме ПДП всегда требует предварительной подготовки, а именно: для каждого ВУ необходимо выделить область памяти, используемую при обмене, и указать ее размер, т. е. количество записываемых в память или читаемых из памяти байт (слов) информации. Следовательно, контроллер ПДП должен обязательно иметь в своем составе регистр адреса и счетчик байт (слов). Перед началом обмена с ВУ в режиме ПДП процессор

должен выполнить программу загрузки. Эта программа обеспечивает запись в указанные регистры контроллера ПДП начального адреса выделенной ВУ памяти и ее размера в байтах или словах в зависимости от того, какими порциями информации ведется обмен. Сказанное не относится к начальной загрузке программ в память в режиме ПДП. В этом случае содержимое регистра адреса и счетчика байт слов устанавливается переключателями или перемычками непосредственно на плате контроллера.

Блок-схема простого контроллера ПДП, обеспечивающего ввод данных в память микроЭВМ по инициативе ВУ в режиме ПДП "Захват цикла", приведена на рис. 44.

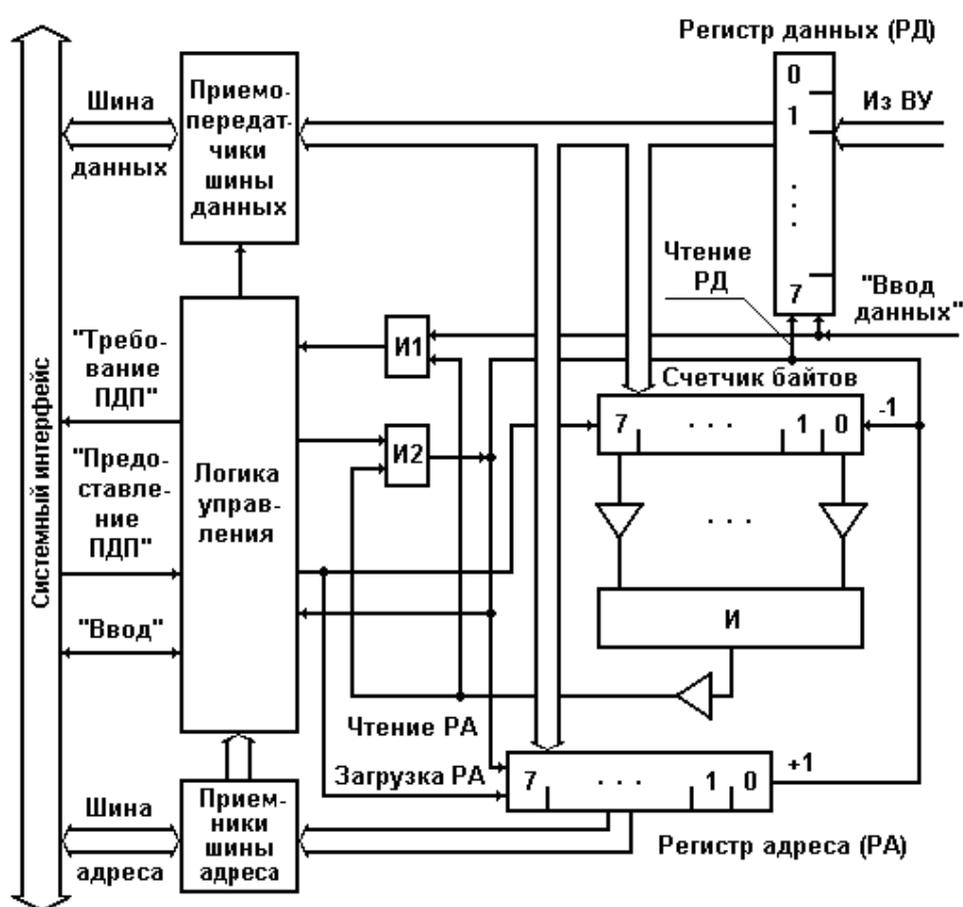


Рис. 44. Контроллер ПДП для ввода данных из ВУ в режиме "Захват цикла" и отключением процессора от шин системного интерфейса

Перед началом очередного сеанса ввода данных из ВУ процессор загружает в регистры его контроллера следующую информацию: в счетчик байт – количество принимаемых байт данных, а в регистр адреса – начальный адрес области памяти для вводимых данных. Тем са-

мым контроллер подготавливается к выполнению операции ввода данных из ВУ в память микроЭВМ в режиме ПДП.

Байты данных из ВУ поступают в регистр данных контроллера в постоянном темпе. При этом каждый байт сопровождается управляющим сигналом из ВУ "Ввод данных", который обеспечивает запись байта данных в регистр данных контроллера. По этому же сигналу и при ненулевом состоянии счетчика байт контроллер формирует сигнал ППДП. По ответному сигналу процессора ППДП контроллер выставляет на шины адреса и данных системного интерфейса содержимое своих регистров адреса и данных, соответственно. Формируя управляющий сигнал "Вывод", контроллер ПДП обеспечивает запись байта данных из своего регистра данных в память микроЭВМ. Сигнал ППДП используется в контроллере и для модификации счетчика байт и регистра адреса. По каждому сигналу ППДП из содержимого счетчика байт вычитается единица, и как только содержимое счетчика станет равно нулю, контроллер прекратит формирование сигналов "Требование прямого доступа к памяти".

На примере простого контроллера ПДП мы рассмотрели только процесс подготовки контроллера и непосредственно передачу данных в режиме ПДП. На практике любой сеанс обмена данными с ВУ в режиме ПДП всегда иницируется программой, выполняемой процессором, и включает два следующих этапа.

1. На этапе подготовки ВУ к очередному сеансу обмена процессор в режиме программно-управляемого обмена опрашивает состояние ВУ (проверяет его готовность к обмену) и посылает в ВУ команды, обеспечивающие подготовку ВУ к обмену. Такая подготовка может сводиться, например, к перемещению головок на требуемую дорожку в накопителе на жестком диске. Затем выполняется загрузка регистров контроллера ПДП. На этом подготовка к обмену в режиме ПДП завершается и процессор переключается на выполнение другой программы.

2. Обмен данными в режиме ПДП начинается после завершения подготовительных операций в ВУ по инициативе либо ВУ, как это было рассмотрено выше, либо процессора. В этом случае контроллер ПДП необходимо дополнить регистром состояния и управления, содержимое которого будет определять режим работы контроллера ПДП. Один из разрядов этого регистра будет инициировать обмен данными с ВУ. Загрузка информации в регистр состояния и управления контроллера ПДП производится программным путем.

Наиболее распространенным является обмен в режиме прямого доступа к памяти с блокировкой процессора. Он отличается от ПДП с "захватом цикла" тем, что управление системным интерфейсом переда-

ется контроллеру ПДП не на время обмена одним байтом, а на время обмена блоком данных. Такой режим ПДП используется в тех случаях, когда время обмена одним байтом с ВУ сопоставимо с циклом системной шины.

В микроЭВМ можно использовать несколько ВУ, работающих в режиме ПДП. Предоставление таким ВУ шин системного интерфейса для обмена данными производится на приоритетной основе. Приоритеты ВУ реализуются так же, как и при обмене данными в режиме прерывания, но вместо управляющих сигналов "Требование прерывания" и "Предоставление прерывания" (рис. 40) используются сигналы "Требование прямого доступа" и "Предоставление прямого доступа", соответственно.