Лекция

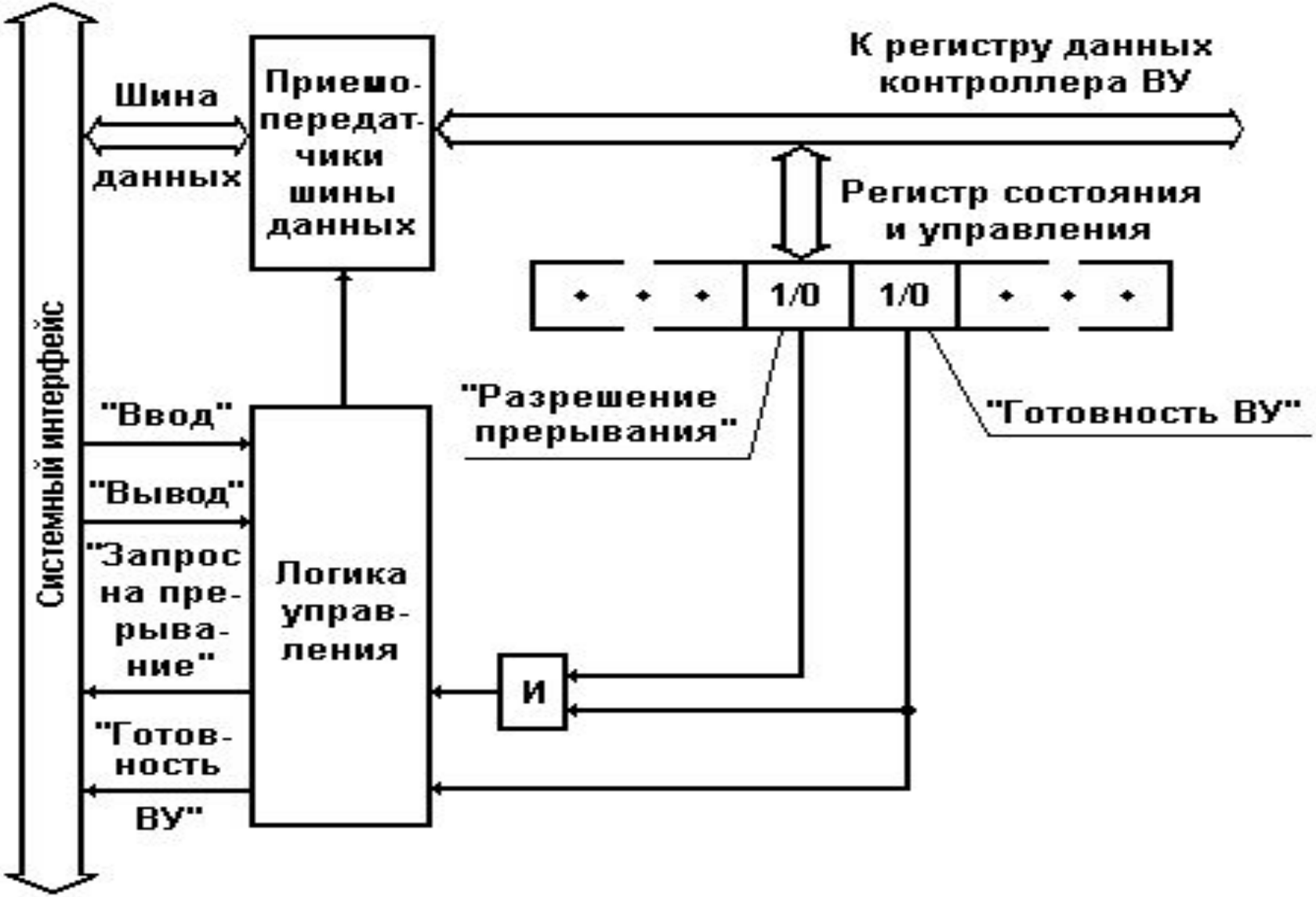
- **Обмен в режиме прерывания. Программные и аппаратные средства, обеспечивающие обмен в режиме прерывания**
- **Обмен в режиме прямого доступа к памяти**

Организация прерываний

Одной из разновидностей программно-управляемого обмена данными с ВУ в микрокомпьютере 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**. **Команды обмена данными** в этом случае выделяют в отдельный программный модуль - **подпрограмму обработки прерывания.** Задачей аппаратных средств обработки прерывания в процессоре микрокомпьютера 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 «Требование прерывания» или «Запрос на прерывание».

Регистр состояния контроллера ВУ дополняют еще одним разрядом - «Разрешение прерывания». Запись 1 или 0 в разряд «Разрешение прерывания» регистра состояния производится программным путем по одной из линий шины данных системного интерфейса. Управляющий сигнал системного интерфейса «Предоставление прерывания» формируется с помощью схемы совпадения только при наличии единиц в разрядах «Готовность ВУ» и «Разрешение прерывания» регистра состояния контроллера.


Системный интерфейс
Шина данных
Приемо-передатчики шины данных
К регистру данных контроллера ВУ
Регистр состояния и управления
1/0
1/0
"Разрешение прерывания"
"Готовность ВУ"
"Ввод"
"Вывод"
"Запрос на прерывание"
"Готовность ВУ"
Логика управления
И

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.

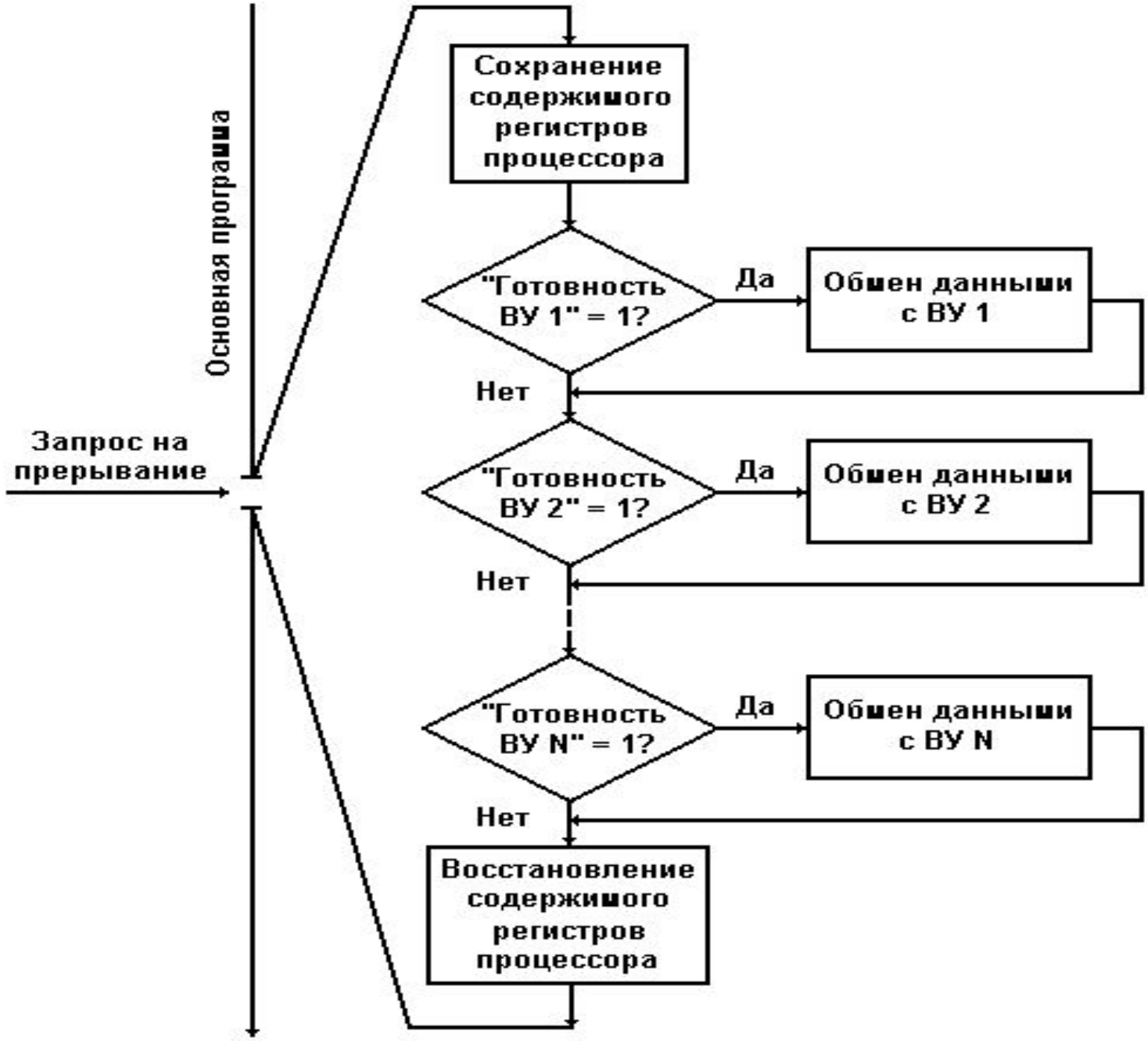
В микрокомпьютере обычно используется одноуровневая система прерываний, т. е. сигналы «Требование прерывания» от всех ВУ поступают на один вход процессора. Поэтому возникает проблема идентификации ВУ, запросившего обслуживание, и реализации заданной очередности (приоритета) обслуживания ВУ при одновременном 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 микрокомпьютера единой подпрограммы обслуживания прерываний от всех ВУ.

Приоритет ВУ в микрокомпьютере с **программным опросом готовности ВУ однозначно определяется порядком их опроса в подпрограмме обработки прерываний.**


Основная программа
Запрос на прерывание
Сохранение содержимого регистров процессора
"Готовность ВУ 1" = 1?
Да
Обмен данными с ВУ 1
Нет
"Готовность ВУ 2" = 1?
Да
Обмен данными с ВУ 2
Нет
"Готовность ВУ N" = 1?
Да
Обмен данными с ВУ N
Нет
Восстановление содержимого регистров процессора

При организации системы прерываний с использованием векторов 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 из контроллера ВУ вектор прерывания, сразу переключается на выполнение требуемой подпрограммы обработки прерывания. В микрокомпьютере с векторной системой прерывания каждое ВУ должно иметь собственную подпрограмму обработки прерывания.

Вектор прерывания выдается контроллером не одновременно с запросом на прерывание, а только по разрешению процессора.

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.

Маскируемые прерывания

Маскируемые прерывания используются для сигнализации о событиях в устройствах. Реакция процессора на маскируемые прерывания может быть задержана сбросом его внутреннего флага IF (инструкция CLI запрещает прерывания, STI – разрешает). По возникновении события, требующего реакции, адаптер (контроллер) устройства формирует *запрос прерывания*, который поступает на вход *контроллера прерываний*. Задача контроллера прерываний – довести до процессора запрос прерывания и сообщить вектор, по которому выбирается программная процедура обработки прерываний. В IBM PC-совместимых компьютерах применяется два основных типа контроллеров прерываний:

- *Периферийный контроллер прерываний* (Peripheral Interrupt Controller, PIC) программно совместим с традиционным контроллером 8259A, использовавшимся еще в первых моделях IBM PC. Со времен IBM PC/AT применяется связка из пары каскадно соединенных контроллеров PIC, позволяющая обслуживать до 15 линий запросов прерываний.

БИС программируемого контроллера прерываний **Intel 8259A**

Контроллер прерываний (ПКП) представляет собой устройство, реализующее до восьми уровней запросов на прерывания, **с возможностью программного маскирования и изменения порядка обслуживания прерываний.**

Маскируемые прерывания

- *Усовершенствованный периферийный контроллер прерываний* (Advanced Peripheral Interrupt Controller, APIC) введен в компьютеры для поддержки мультипроцессорных систем на базе процессоров 4–5-го поколений (486 и Pentium) и используется поныне для более поздних моделей процессоров. Помимо поддержки мультипроцессорных конфигураций, современный контроллер APIC позволяет увеличивать число доступных линий прерываний и обрабатывать запросы прерываний от устройств PCI, посылаемые через механизм сообщений (MSI). Компьютер, оснащенный контроллером APIC, обязательно имеет возможность функционировать и в режиме, совместимом со стандартной связкой пары PIC. Этот режим включается по аппаратному сбросу (и включению питания), что позволяет использовать старые ОС и приложения MS-DOS, «не знающие» APIC и мультипроцессирования.

Традиционная схема формирования запросов прерываний с использованием пары контроллеров PIC изображена на рис. 4.4.

Маскируемые прерывания

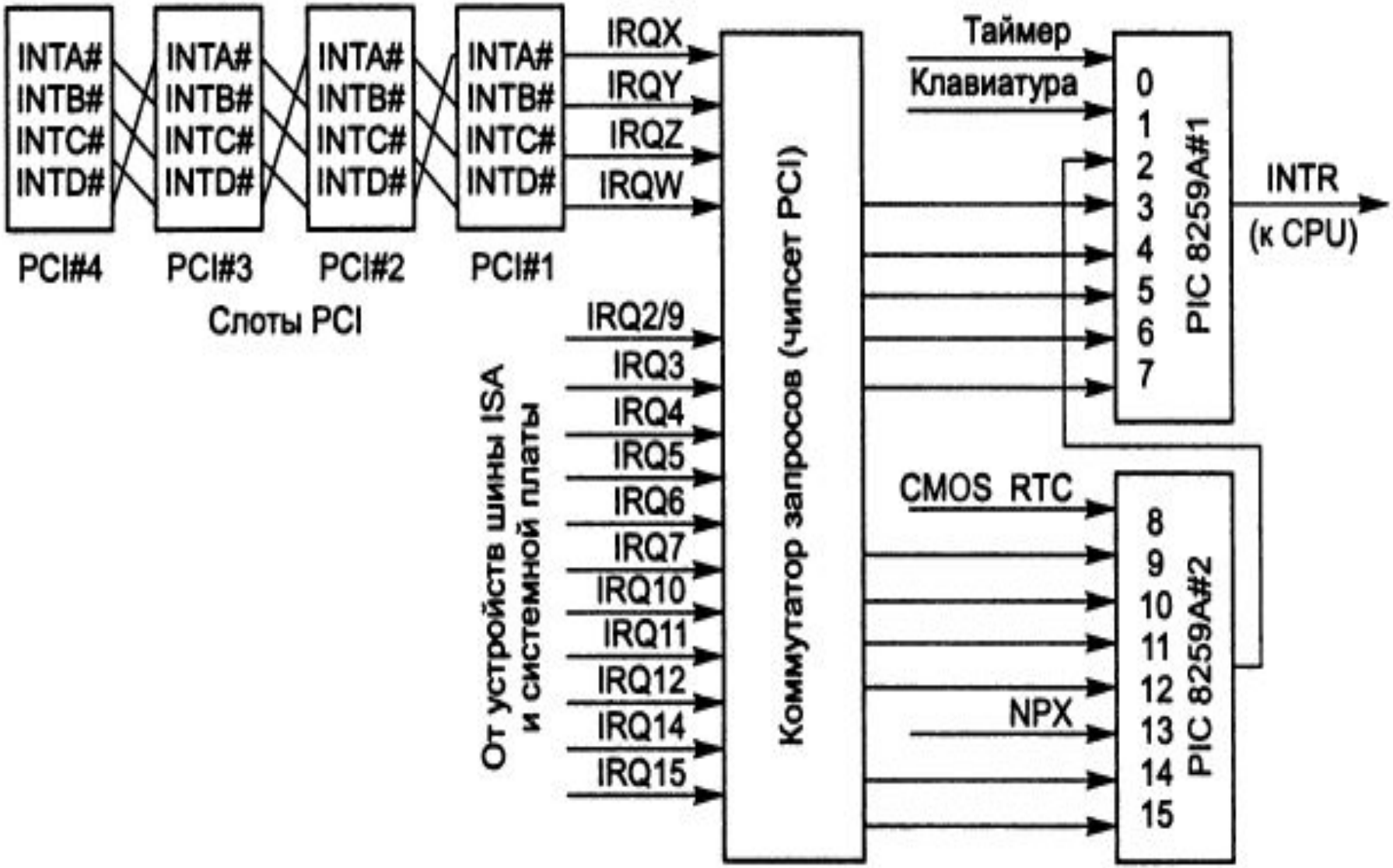


Рис. 4.4. Коммутация запросов прерываний

Аппаратные прерывания PC AT

Master/ Slave	Запрос на прерывание	Источник прерывания	Приоритет	Код вектора
___	NMI*	Ошибка паритета памяти или другая неисправимая ошибка в системе	1	02h
MASTER	IRQ0	Системный таймер	2	08h
MASTER	IRQ1	Клавиатура	3	09h
SLAVE	IRQ8	Часы реального времени	4	70h
SLAVE	IRQ9	Резерв	5	71h
SLAVE	IRQ10	Резерв	6	72h
SLAVE	IRQ11	Резерв	7	73h
SLAVE	IRQ12	Резерв	8	74h
SLAVE	IRQ13	Ошибка сопроцессора	9	75h
SLAVE	IRQ14	НМД	10	76h
SLAVE	IRQ15	Резерв	11	77h
MASTER	IRQ3	RS232	12	0Bh
MASTER	IRQ4	RS232	13	0Ch
MASTER	IRQ5	Параллельный порт 2	14	0Dh
MASTER	IRQ6	НГМД	15	0Eh
MASTER	IRQ7	Параллельный порт 1	16	0Fh

Маскируемые прерывания

После того как конфигурирование системы прерываний произведено (проинициализирован контроллер прерываний, устройствам назначены линии запросов и установлены указатели на процедуры обработки), отработка маскируемых аппаратных прерываний происходит следующим образом:

1. Устройство по событию прерывания возбуждает назначенную ему линию запроса прерывания.
2. Контроллер принимает *сигналы запросов от источников прерываний* (сигналы IRQx) и при наличии незамаскированного запроса подает сигнал *общего запроса прерывания* (сигнал INTR) процессору x86.
3. Процессор, реагируя на запрос (когда прерывания флагом IF разрешены), сохраняет в стеке содержимое регистра флагов и адрес возврата, после чего формирует *шинный цикл INTA* (Interrupt Acknowledge — подтверждение прерывания), который доводится до контроллера прерываний.
4. В момент получения сигнала INTA контроллер прерываний фиксирует состояние своих входов запросов — к этому моменту их состояние могло измениться: например, появились новые запросы или пропал запрос от «нетерпеливого» устройства. Контроллер анализирует поступившие запросы в соответствии с запрограммированной схемой приоритетов и посылает процессору *вектор прерывания*, соответствующий самому приоритетному незамаскированному запросу, присутствующему на входе контроллера в момент подачи шинной команды *INTA*. При этом контроллер выполняет и некоторые действия в соответствии с установленной приоритетной политикой, учитывающие, какой именно вектор был послан (какой из запросов пошел на обслуживание).

Маскируемые прерывания

5. Получив вектор прерывания, процессор по его номеру вызывает соответствующую процедуру обработки прерывания. Если данный вектор прерывания используется не только для аппаратных прерываний, но и для исключений и/или программных прерываний, то процедура в первую очередь должна определить, к какому из этих типов относится данное событие. Для этого процедура может обратиться к контроллеру PIC (прочитать регистр ISR) и проанализировать состояние регистров процессора.

   Дальнейшие шаги описывают случай аппаратного прерывания.

6. Процедура обработки прерывания идентифицирует источник прерывания – определяет устройство, его вызвавшее. При разделяемом использовании несколькими устройствами данного номера запроса (следовательно, и вектора) идентифицировать источник прерывания можно только последовательными обращениями к регистрам каждого из этих устройств. При этом следует учитывать возможность поступления запросов от нескольких устройств одновременно или в процессе обработки прерывания от одного из них.

7. Процедура обслуживает устройство-источник прерывания — выполняет «полезные» действия, связанные с событием, о котором и сигнализировало устройство. Это обслуживание должно обеспечить и *снятие сигнала запроса* прерывания от данного устройства. В случае разделяемых прерываний источников может быть несколько, и все они требуют обслуживания.

   Если обработка прерывания занимает значительное время, в течение которого требуется реакция системы на более приоритетные запросы, то после критической секции в обработчик включают инструкцию `STI`, устанавливающую флаг разрешения прерываний (`IF`) в процессоре. С этого момента возможны *вложенные прерывания*, прерывающие работу данного обработчика при поступлении другого, более приоритетного прерывания.

8. Процедура обработки прерывания посылает контроллеру *команду EOI* (End Of Interrupt — завершение обработки прерывания), по которой контроллер разрешает последующий прием сигнала с обслуженного входа и менее приоритетных. Это должно быть сделано после снятия сигнала прерывания от обслуженных устройств, иначе контроллер после *EOI* пошлет повторный запрос. Обработчик прерывания, для которого запрос поступил от ведомого контроллера, должен послать *EOI* как ведомому, так и ведущему контроллеру. Фрагмент обработчика от инструкции подачи команды *EOI* до завершения (инструкции `IRET`) должен быть *непрерываемым*, то есть являться *критической секцией*. Если обработчик разрешал вложенные прерывания, то перед инструкцией подачей команды *EOI* должна присутствовать инструкция `CLI`, запрещающая прерывания.

Маскируемые прерывания

9. Завершается обработка прерывания инструкцией IRET, по которой процессор возвращается к выполнению прерванного потока инструкций, предварительно извлекая из стека содержимое регистра флагов (с установленным IF), и аппаратные прерывания снова оказываются разрешенными.

Режимы работы ПКП

- **Режим полного вложения** (приоритет от высшего IRQ к низшему) – основной режим. Пока установлен разряд в ISR все последующие запросы с таким же или более низким приоритетом игнорируются, **подтверждаются** лишь **запросы с более высоким приоритетом.**

- **Циклический режим** (используется **круговой порядок** использования **приоритетов**). Последнему обслуживаемому запросу присваивается низший приоритет, следующему по кругу – наивысший.

Идея – если устройство обслуживается, то остальные устройства должны быть обслужены перед следующим обслуживанием данного устройства.

- **Маскирование прерываний**. Позволяет устройствам с более низким приоритетом получить возможность генерировать прерывания. Режим специального маскирования **разрешает прерывания всех уровней кроме уровней, обслуживаемых в данный момент**.

- **Специальный режим полного вложения** программируется любым ведущим контроллером при инициализации (игнорируются запросы с приоритетом ниже, чем приоритет обрабатываемого в данный момент запроса, **обслуживаются все запросы с равным или более высоким приоритетом**).

Немаскируемые прерывания

Немаскируемые прерывания (Non-Maskable Interrupt, NMI) в PC используются для сигнализации о фатальных аппаратных ошибках. На немаскируемое прерывание процессор реагирует всегда (если завершено обслуживание предыдущего немаскируемого прерывания); этому прерыванию соответствует фиксированный вектор 2. Сигнал на линию NMI (вход процессора) приходит от схем контроля памяти (четности или ECC), от линий контроля шины ISA (IOCHK) и шины PCI (SERR#). Сигнал NMI блокируется до входа процессора установкой в 1 бита 7 порта 070h, отдельные источники разрешаются и идентифицируются битами порта 061h:

- бит 2 R/W (ERP) – разрешение контроля ОЗУ и сигнала SERR# шины PCI;
- бит 3 R/W (EIC) – разрешение контроля шины ISA;
- бит 6 R (IOCHK) – ошибка контроля на шине ISA (сигнал IOCHK#);
- бит 7 R (PCK) – ошибка четности ОЗУ или сигнал SERR# на шине PCI.

Прерывания могут быть внешними и внутренними.
Внешние прерывания относятся, к непланируемым прерываниям.

Внутренние прерывания возникают внутри микропроцессора во время вычислительного процесса. К их возбуждению приводит одна из двух причин:

– ненормальное внутреннее состояние микропроцессора, возникшее при обработке некоторой команды программы. Такие события принято называть исключительными ситуациями, или просто исключениями.

– обработка машинной команды int xx. Этот тип прерываний называется программным. Это – планируемые прерывания, так как с их помощью программист обращается в нужное для него время за обслуживанием своих запросов либо к операционной системе, либо к BIOS, либо к собственным программам обработки прерываний.

(номером прерывания).
В реальном режимеВ реальном режиме (RM) таблица векторов прерываний расположена в первом килобайте памяти начиная с адреса 0000:0000 и содержит 256 векторов прерываний. В защищённом режимеВ реальном режиме (RM) таблица векторов прерываний расположена в первом килобайте памяти начиная с адреса 0000:0000 и содержит 256 векторов прерываний. В защищённом режиме (PM) адрес в физической памяти и размер таблицы прерываний определяется 48-битнымВ реальном режиме (RM) таблица векторов прерываний расположена в первом килобайте памяти начиная с адреса 0000:0000 и содержит 256 векторов прерываний. В защищённом режиме (PM) адрес в физической памяти и размер таблицы прерываний определяется 48-битным регистром IDTR.
В IDT используются следующие типы прерываний: аппаратные прерывания, программные прерывания и прерывания, зарезервированные процессором, называемые исключениями (для RM первые пять, для PM первые 32) на случай возникновения некоторых событий (деление на ноль, ошибка трассировки, переполнение).
В реальном режиме элементом IDT является 32-битныйВ реальном режиме элементом IDT является 32-битный адрес обработчика прерывания.
В защищённом режиме элементом IDT является шлюзВ защищённом режиме элементом IDT является шлюз прерывания длиной 8 байтВ

Организация прямого доступа к памяти

Одним из способов обмена данными с ВУ является обмен в режиме прямого доступа к памяти (ПДП).

В этом режиме **обмен данными между ВУ и основной памятью компьютера 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.

Организация прямого доступа к памяти

Необходимость реализации в микрокомпьютере обмена данными в режиме ПДП вызывается двумя основными факторами.

Во-первых, **при использовании режима ПДП появляется возможность начальной загрузки программ в основную память из устройства ввода**.

Во-вторых, что и является наиболее важным, **режим ПДП обеспечивает возможность использования быстродействующих внешних запоминающих устройств, таких, например, как накопители на жестких магнитных дисках.**

В целях сокращения количества линий в шинах микрокомпьютера **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.

Наиболее распространенным является ПДП **с «захватом цикла» и принудительным отключением процессора от шин системного интерфейса**. Для реализации такого режима ПДП системный интерфейс микрокомпьютера дополняется двумя линиями для передачи управляющих сигналов «Требование прямого доступа к памяти» (ТПДП) и «Предоставление прямого доступа к памяти» (ППДП).

Применение в компьютере 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 байтов (слов) информации.

На практике любой сеанс обмена данными с ВУ в режиме ПДП всегда инициируется программой, реализуемой 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 его подготовку к обмену.

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.

Прямой доступ в память с блокировкой процессора *отличается от ПДП с «захватом цикла» тем, что управление системным интерфейсом передается контроллеру ПДП не на время обмена одним байтом, а на время обмена блоком данных.* Такой режим ПДП необходим в тех случаях, когда время обмена одним байтом с ВУ сопоставимо с циклом процессора. В этом случае процессор не успевает выполнить хотя бы одну команду между очередными операциями обмена в режиме ПДП.