1. **Объясните необходимость системы прерываний и цель введения приоритетов**

Блок прерываний является составной частью системы прерывания, назначение которой состоит в реагировании на некоторые особые ситуации, возникающие в процессе выполнения программ, путем прерывания выполнения текущей программы и переключения процессора на выполнение другой программы, обслуживающей соответствующую ситуацию. Такими особыми ситуациями могут быть переполнение разрядной сетки ЭВМ, неисправность аппаратуры и т.д. После обслуживания причины возникновения прерывания обычно осуществляется возврат к прерванной программе, которая продолжается о того места, где она была прервана.

Сигналы, несущие информацию о возникновении особых ситуаций, называется запросами прерывания - {Pi}. Запрос прерывания Рi поступает в блок прерываний УУ, где он запоминается и на его основе формируется специальная операция - операция прерывания, обеспечивающая прерывание текущей программы и переход к исполнению программы анализа возникшей ситуации. Сигналы запросов прерывания могут поступать одновременно от нескольких устройств ЭВМ по различным причинам, причем они МОГУТ поступать одновременно. Все сигналы запросов должны быть зафиксированы, а исполниться должен только один из них. Например, одновременно поступили сигнал прерывания от схемы контроля сумматора и сигнал прерывания по причине переполнения разрядной сетки. Очевидно, на второй не имеет смысла реагировать до тех пор, пока не будет устранена неисправность в сумматоре, вызвавшая первый сигнал.

Для выделения из множества сигналов запроса, поступивших в процессор, одного из них, который и вызовет прерывание, всем сигналам прерывания присваиваются приоритеты. Приоритет характеризуется целым числом 0,1,2,... . Чем меньше значение числа, тем выше приоритет. Приоритеты распределяются между причинами прерываний по степени важности и определяются в общем случае областью применения ЭБМ. Для выделения сигналов запроса прерывания блок прерываний содержит схему приоритета.

Исходя из этого функциями системы прерываний являются:

* прием и запоминание запросов прерывания;
* выделение запроса с высшим приоритетом из числа поступивших;
* прерывание выполнения текущей программы и запоминание информации, характеризующей состояние процессора в момент прерывания (обычно реализуется программным способом);
* организация перехода к выполнению программы, обрабатывающей выделенный запрос прерывания, - программы прерывания;
* сброс выделенного запроса прерываний;
* после исполнения программы прерывания организация возврата к прерванной программе, что выполняется путем восстановления состояния узлов и блоков процессора по запомненной ранее информации (обычно реализуется программным способом).

1. **Какие действия выполняет процессор при поступлении запроса на прерывание? Назовите возможные источники запросов прерывания.**

Система прерываний переводит процессор на выполнение потока команд, отличного от того, который выполняется до сих пор, с последующим возвратом к исходному коду.

Механизм обработки прерываний независимо от архитектуры вычислительной системы подразумевает выполнение некоторой последовательности шагов.

1. Установление факта прерывания (приём сигнала запроса на прерывание) и идентификация прерывания.

2. Запоминание состояния прерванного процесса вычислений. Состояние процесса выполнения программы определяется, прежде всего, значением счетчика команд (адресом следующей команды), содержимым регистров процессора, и может включать также спецификацию режима (например, режим пользовательский или привилегированный) и другую информацию.

3. Управление аппаратно передается на подпрограмму обработки прерывания.

4. Сохранение информации о прерванной программе, которую не удалось спасти на шаге 2 с помощью аппаратуры. В некоторых процессорах предусматривается запоминание довольно большого объема информации о состоянии прерванных вычислений.

5. Собственно выполнение программы, связанное с обработкой прерывания. Эта работа может быть выполнена той же подпрограммой, на которую было передано управление на шаге 3, но в операционных системах достаточно часто она реализуется путем последующего вызова соответствующей подпрограммы.

6. Восстановление информации, относящейся к прерванному процессу (этап, обратный шагу 4).

7. Возврат на прерванную программу.

Шаги 1-3 реализуются аппаратно, шаги 4-7 – программно.

В зависимости от источника прерывания делятся на три вида:

Внешние прерывания могут возникать от аппаратных устройств (вне прерываемого процесса), они сообщают о чем-то, требующим внимания процессора. Так же они могут возникать в результате действий пользователя за клавиатурой.

Так же эти прерывания называют аппаратными, так как возникают после подачи устройством электрического сигнала, который подается на специальный вход прерывания процессора.

* Прерывание, которое информирует систему о том, что требуемый сектор диска уже прочитан, его содержимое доступно программе.
* Прерывание, которое информирует систему о том, что завершилась печать символа на принтере и необходимо выдать следующий символ.
* Прерывания по нарушению питания.
* Нормальное завершение некоторой операции ввода-вывода (нажатие клавиши на клавиатуре).
* Прерывание по таймеру.

Внутренние прерывания вызываются событиями, которые связаны с работой процессора и являются синхронными с его операциями. Могут быть вызваны при обнаружении ошибки при выполнении программы. Это похоже на то, как если бы вы при чтении какого-либо предложения встретили «РБНСС ЗМЙ ЮДЙХМ ЯСМ ЖЯК», т.е. полнейшую бессмыслицу. Аналогичная ситуация может возникнуть и в компьютере. Процессор может встретить команды, не имеющие для него никакого смысла, или данные, которые нельзя обработать. Например, нулевой делитель, ошибки защиты памяти, обращение по несуществующему адресу, попытка выполнить привилегированную инструкцию в пользовательском режиме и т.п.

* при нарушении адресации (в адресной части выполняемой команды указан запрещенный или несуществующий адрес, обращение к отсутствующему сегменту или странице при организации механизмов виртуальной памяти);
* при наличии в поле кода не задействованной двоичной комбинации.
* при делении на нуль.
* при переполнении или исчезновении порядка.
* при обнаружении ошибок четности, ошибок в работе различных устройств аппаратуры средствами контроля.

В подробных ситуациях процессор генерирует прерывание, называемое особым случаем или исключением.

Программные прерывания не возникают непредвиденно. Идея прерываний настолько мощная, что с помощью прерывания программы могут запросить обслуживания, выполняемые другими программами. Программные прерывания возникают при выполнении особой команды процессора и представляют собой удобный способ вызова процедур ОС.

Пример (программные прерывания):

* привилегированная команда в режиме пользователя.
* адрес вне диапазона.
* нарушение защиты памяти.
* арифметическое переполнение, отсутствует страница.
* нарушение защиты сегмента.
* выход за границу сегмента.

Сигналы, вызывающие прерывания, формируются вне процессора или в самом процессоре, они могут возникать одновременно. Выбор одного из них для обработки осуществляется на основе приоритетов, приписанных каждому типу прерывания. Так, со всей очевидностью, прерывания от схем контроля процессора должны обладать наивысшим приоритетом (действительно, если аппаратура работает неправильно, то не имеет смысла продолжать обработку информации).

1. **Всегда ли возможно запретить прерывания? Каким образом это происходит?**

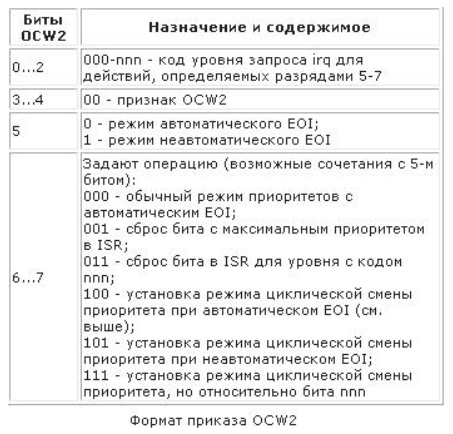
Самое простое решение состоит в запрете всех прерываний при входе процесса в критическую секцию и разрешение прерываний по выходу из нее. Если прерывания запрещены, невозможно прерывание по таймеру. Запрет на прерывания исключает передачу процессора другому процессу. Таким образом, запретив прерывания, процесс может спокойно считывать и сохранять общие данные, не опасаясь вмешательства конкурентов.

И все же было бы неразумно давать пользовательскому процессу полномочия для запрета прерываний. Представьте себе, что процесс отключил все прерывания и в результате какого-либо сбоя не включил их обратно. Операционная система на этом может закончить свое существование. К тому же, в многопроцессорной системе запрет прерываний влияет только на тот процессор, который выполняет инструкцию disable. Остальные процессоры продолжат работу и сохранят доступ к общим данным.

В то же время для ядра блокировка прерываний для некоторых команд может быть полезной при работе с переменными или списками. Возникновение прерывания в момент, когда, например, список готовых процессов находится в неопределенном состоянии, могло бы привести к условиям гонок. Итак, запрет прерываний бывает полезным в самой операционной системе, но для пользовательских процессов это решение в качестве механизма взаимного исключения неприемлемо.

1. **Приоритеты прерываний в К1810ВН59А. Как можно изменить приоритет с помощью управляющего слова OCW2?**

Шестое. OCW2 — управление приоритетом Подробности Родительская категория: Ввод-вывод Категория: Программирование контроллера прерываний i8259А Этот приказ (см. таблицу ниже) используется для управления приоритетом и учета особенностей завершения обслуживания прерывания в контроллере. Он определяет выполнение следующих действий: сбросить бит в ISR с наибольшим приоритетом; сбросить бит в ISR для определенного уровня прерываний; установить низший приоритет для определенного уровня; поменять приоритеты уровней с максимальным и минимальным приоритетами; поменять приоритеты уровней с заданным и минимальным приоритетами. Данный приказ посылается в порт 20h.



1. **Как осуществляется формирование начального адреса ППОП в К1810ВН59А?**

Основная функция – формирование начального адреса прерывающей программы. Любому запросу соответствует своя прерывающая программа. Существует три различных способа, используемых при формировании адреса:

1) Размещение прерывающих программ по фиксированным адресам. В некоторой постоянно распределенной области основной памяти по фиксированным адресам размещаются прерывающие программы, это размещение не меняется. Вход реализуется аппаратно, то есть адрес формируется аппаратно – это самый быстрый способ. Но уданного способа существуют серьезные ограничения:

a) привязка к адресам

b) количество причин прерывания должно быть достаточно малым

Этот способ применяется при малых системах прерывания и для тех причин, которые требуют немедленной реакции.

2) вход на основании слов состояния программы (PSW).

3) Векторное прерывание – самый распространенный способ. Данный способ является программно-аппартным, т.е. для любого запроса (для любого выделенного запроса) аппаратно формируется адрес вектора прерывания. Чаще всего эти адреса фиксированы. Адреса векторных прерываний хранятся в системной области памяти. На основе адреса вектора из таблицы векторов прерывания извлекается начальный адрес прерывающей программы. Это приводит к тому, что код запроса может быть малобитным, но таблица векторов прерывания должна хранится в начальной области памяти. В качестве вектора прерывания используются:

a) адрес начала прерывающей программы (применяется в PC)

b) команда безусловного перехода к программе

1. **Какие действия выполняются в ПКП и микропроцессоре при обращении к ППОП (после выработки сигнала запроса на прерывание INT) для микропроцессора К580ВМ80?**

ПКП выполняет следующие функции:

♦ фиксацию запросов на прерывание от 8 внешних источников;

♦ программное маскирование поступивших запросов;

♦ выделение приоритетного запроса по отношению к приоритету обслуживаемого прерывания с целью выявления возможности прерывания;

♦ присвоение фиксированных или циклически изменяемых приоритетов входам контроллера, на которые поступают запросы; 38

♦ программный опрос внешних устройств для определения, нуждается ли устройство в обмене;

♦ формирование номера вектора прерывания, а на его основе: 1. для микропроцессора К580ВМ80 выдачу кода операции команды CALL (перехода на подпрограмму) и 16-разрядного адреса таблицы переходов для этой подпрограммы;

1. **Какие действия выполняются в ПКП и микропроцессоре при обращении к ППОП (после выработки сигнала запроса на прерывание INT) для микропроцессора К1810ВМ86?**

ПКП выполняет следующие функции:

♦ фиксацию запросов на прерывание от 8 внешних источников;

♦ программное маскирование поступивших запросов;

♦ выделение приоритетного запроса по отношению к приоритету обслуживаемого прерывания с целью выявления возможности прерывания;

♦ присвоение фиксированных или циклически изменяемых приоритетов входам контроллера, на которые поступают запросы; 38

♦ программный опрос внешних устройств для определения, нуждается ли устройство в обмене;

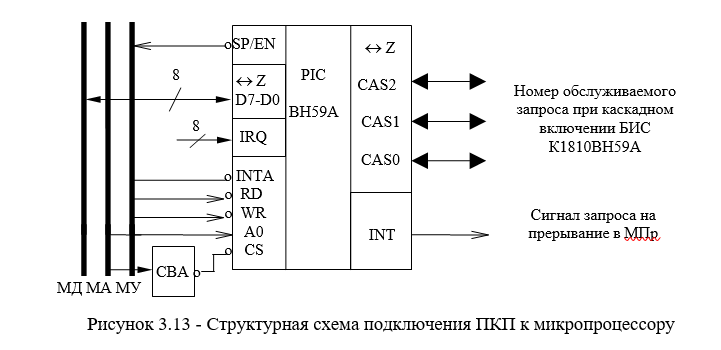
♦ формирование номера вектора прерывания, а на его основе:

2. для микропроцессора К1810ВМ86 и более старших моделей выдачу адреса (вектора) таблицы IDT векторов прерываний

1. **Каскадное включение БИС ПКП ВН59А. Поясните алгоритм работы схемы при поступлении запросов на входы IRQ ведущего ПКП и ведомого.**

Одна БИС ПКП предназначена для обработки до восьми входных сигналов запроса IRQ. На рисунке 3.13 приведена структурная схема подключения 8- уровневого ПКП к магистрали микропроцессора. В адресном пространстве портов ввода-вывода одна БИС ПКП представлена двумя адресами (портами) А7- А1=constant, A0=0 или 1. Сигналы ~RDIO, ~WRIO вырабатываются контроллером шины МПр при выполнении команд IN и OUT, сигнал ~INTA также вырабатыва- ется контроллером шины в ответ на поступление от ПКП сигнала запроса на пре- рывание INT.

Каскадное соединение нескольких БИС ПКП позволяет расширить число входных уровней до 64 (один ведущий и 8 ведомых). На рисунке 3.14 приведена схема каскадного включения для 22 входных сигналов IRQ для МПр серии К1810.



При начальной установке ПКП (S=0 в ICW1) обязательно в каждую БИС

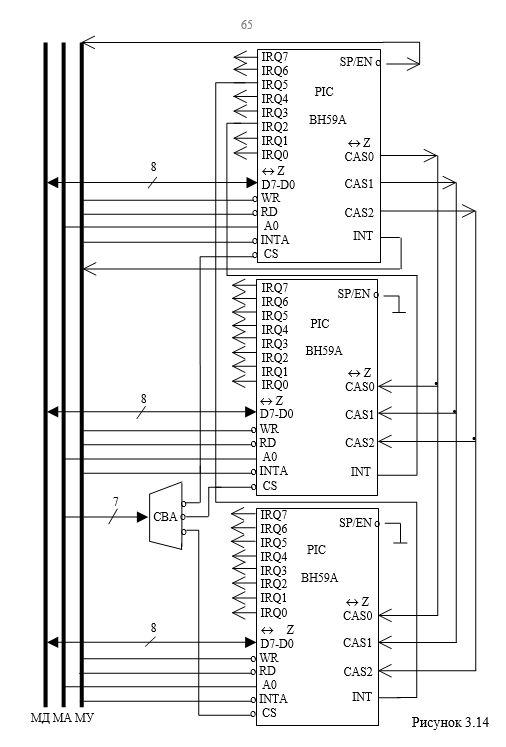
ПКП должно быть загружено управляющее слово ICW3: для ведущего ПКП код “1” содержат те разряды ICW3, к входам которых подключены выходы INT ведо- мых БИС; для ведомых БИС в разрядах D2-D0 ICW3 указывается номер микро- схемы, совпадающий с номером IRQ ведущей БИС, к которому подключен выход INT ведомой.

Для МПр серии К580 для ведущего ПКП вывод ~EN/~SP является входом и на него подается уровень логической единицы (напряжение питания), для ведо- мых БИС - уровень логического нуля, а для МПр серии К1810 вывод ~EN/~SP ве- дущего ПКП является выходом и сигнал ~EN поступает на вход шинного форми- рователя BA86 контроллера шины, а тип БИС ПКП (ведомая или ведущая) указы- вается в команде инициализации ICW4. Выводы ~EN/~SP ведомых БИС являются входами, на которые подается уровень логического нуля аппаратно, а при выдаче вектора на шину данных из ведомой БИС по второму ~INTA сигнал для шинного формирователя ~EN формируется ведущим ПКП, так как выводы ~EN/~SP нельзя объединять по монтажному "ИЛИ" (они не имеют третьего состояния).

При каскадном включении, если сигнал INT, поступивший в МПр, вырабо- тан сигналом IRQ от ведущего ПКП, то формирование адреса ППОП или адреса вектора таблицы IDT осуществляется этой же БИС аналогично тому, как это про- исходит при работе с одной БИС. Если же выработка сигнала INT вызвана по сиг- налу IRQ, поступившему на вход ведомой микросхемы, то формирование адреса ППОП осуществляется следующим образом. Для МПр серии К580 при поступле- нии первого сигнала ~INTA1 ведущая БИС выдает на шину данных код команды CALL, а на шину CAS2-CAS0 - код номера ведомой БИС, вызвавшей прерывание. Ведомые БИС по этой же шине принимают этот код и сравнивают со своим номе- ром, загруженным в ICW3 при начальной инициализации. По второму и третьему сигналам ~INTA код адреса ППОП выдается на шину данных той ведомой БИС, номер которой совпал с кодом на шине CAS2-CAS0.

Для МПр серии К1810 по сигналу ~INTA1 выполняется только выдача но- мера запроса IRQ, вызвавшего выработку сигнала INT на шину CAS2-CAS0 из ве- дущего ПКП, а по ~INTA2 по совпавшему номеру ведомая БИС ПКП выдает на шину данных адрес вектора прерывания таблицы IDT, а ведущая БИС сопровож- дает его выдачей сигнала ~EN, если в команде ICW4 ведущей был запрограмми- рован бит разрешения данных ~EN.

Заметим, что ведущая и ведомые БИС должны иметь различные адреса пор- тов ввода-вывода, могут программироваться независимо друг от друга и на раз- личные режимы обслуживания запросов.



В ПКП ВН59А устранен недостаток нарушения режима вложенности обра- ботки запросов при каскадном включении от ведомых ПКП путем введения до- полнительного режима строгого упорядочения приоритетов, который задается командой ICW4 (бит PSV D4=1) только в ведущем ПКП. При этом в ведущем ПКП снимается требование строгого упорядочения приоритетов и разрешается прием новых запросов со входов IRQ, только что принятых к обслуживанию (или еще называют разрешение приоритета ведомого), если бит AEOI=0. С точки зре- ния технической реализации это означает, что схема сравнения СОМ разрешает выделение приоритетного запроса, если его номер меньше или равен номеру ISR- бита ведущего, установленного в “1” (если PSV=0, то сравнение производится только на меньше (рисунок 3.2б)).

В этом случае ведущий ПКП разрешает незамаскированным IRQ с более высоким приоритетом в ведомой БИС проходить на выход INT ведущего, даже если соответствующий бит ISR регистра в ведущем установлен в "1". То есть, ес- ли в ведомом ПКП во время обработки запросов, приходит запрос с более высо- ким приоритетом, то новый IRQ вновь сформирует сигнал INT в ведущем ПКП.

При использовании режима строгого упорядочения приоритетов несколько изменяется и организация возврата из ППОП. Для входов IRQ ведущего ПКП в ППОП необходима загрузка одного OCW2 только в ведущий ПКП, а для IRQ ве- домых ПКП требуется загрузка одного или двух OCW2 по следующему алгорит- му:

1. Сначала OCW2 неконкретного EOI загружается в ведомый ПКП, который вызвал прерывание.
2. Затем считывается и проверяется значение ISR регистра этого ведомого (предварительно в OCW3 должен быть установлен режим чтения ISR регистра, если он не был установлен при инициализации или при установке режима спе- циального маскирования или программного опроса).
3. Если регистр ISR ведомого ПКП равен нулю, то OCW2 загружается в ве- дущий ПКП, иначе OCW2 в ведущий не загружается.

Например, два ведомых ПКП подключаются ко входам IRQ1 и IRQ2 веду- щего. Тогда приоритеты входов IRQ распределяются следующим образом:

Ведущий: IRQ0 - высший приоритет

Ведомый 1: IRQ0, IRQ1, IRQ2, IRQ3, IRQ4, IRQ5, IRQ6, IRQ7

Ведомый 2: IRQ0, IRQ1, IRQ2, IRQ3, IRQ4, IRQ5, IRQ6, IRQ7

Ведущий: IRQ3, IRQ4, IRQ5, IRQ6, IRQ7 - низший приоритет.

Пусть запросы не маскируются и поступает запрос на вход IRQ4 первого ведомого:

1. ситуация: AEOI=0 и PSV=0 для ведущего ПКП.

После первого ~INTA ISR регистр ведущего равен: 00000010 ISR регистр 1-го ведомого равен: 00010000

Тогда ни один из вновь поступивших запросов с более высоким приорите- том IRQ0-IRQ3 первого ведомого не может прервать выполнение текущей ППОП до сброса ISR-бита в ведущем ПКП. Только один запрос IRQ0 ведущего может прервать текущую ППОП.

1. ситуация: AEOI ведущего = 1, PSV=0.

По второму сигналу ~INTA в ведущем ПКП будет автоматически сброшен ISR-бит с наивысшим приоритетом, если не установлен режим автоматического сдвига приоритетов (дополнительная команда OCW2) и теперь запросы IRQ0- IRQ3 первого ведомого и IRQ0 ведущего с более высоким приоритетом могут прервать выполнение текущей ППОП, но и запросы по входам IRQ2-IRQ7 веду- щего с более низким приоритетом могут прервать выполнение текущей ППОП, включая IRQ0-IRQ7 второй ведомой через вход IRQ2 ведущей БИС ПКП.

1. ситуация: AEOI=0, PSV=1 для ведущей.

В данной ситуации запросы только IRQ0-IRQ3 первой ведомой и IRQ0 ве- дущей с более высоким приоритетом могут прервать выполнение текущей ППОП. При этом необходимо учитывать, что при обработке прерывания от веду- щего ПКП (запросы IRQ0, IRQ3-IRQ7 ведущего) могут прерывать свои собствен- ные ППОП, для чего при входе в ППОП программист должен предусмотреть их маскирование по этим входам до окончания выполнения ППОП.

В данном примере также необходимо учитывать в ППОП от ведомых БИС последовательность сброса ISR-бита в ведомом и ведущем ПКП по ранее рас- смотренному алгоритму. Например, при обработке прерывания IRQ4 от первого ведомого поступает запрос IRQ2 , а затем IRQ0 от первого ведомого. Тогда в ISR- регистре первого ведомого будет сформирован код 00010101 и если при возврате из прерывания в ППОП ведомого ПКП сразу сбросить ISR-бит в ведомом и веду- щем, то сразу будут разрешены прерывания от источников с более низким при- оритетом от ведущего и второго ведомого ПКП до завершения обработки ППОП с более высоким приоритетом (рисунок 3.16).

1. **Ответ в восьмом.**
2. **Какие требования (особенности) должен учитывать программист для реализации режима программного опроса?**

Режим программного опроса используется для определения источника пре- рывания непосредственно программистом путем последовательного опроса ис- точников запросов на прерывание IRQ многократной загрузкой управляющего слова OCW3 и чтением в МПр слова состояния ПКП. Для реализации этого режи- ма предварительно прерывания в МПр должны быть запрещены (в регистре слова состояния процессора флаг разрешения прерываний IF должен быть сброшен). Тогда в ответ на поступление сигнала прерывания INT микропроцессор игнориру- ет выдачу сигналов "Подтверждения прерывания" ~INTA, а функции опроса со- стояния аппаратуры прерывания возлагаются на программиста.

Следовательно, для перехода в режим последовательного опроса необходи- мо при инициализации или перед установкой режима программного опроса запре- тить прерывания в МПр, установить режим программного опроса загрузкой ко- манды OCW3, а затем прочитать в МПр при А0=0 слово состояния ПКП в форма- те, представленном на рисунке 3.12. Младшие разряды считанного байта слова состояния ПКП указывают номер внешнего устройства (вектора прерывания) с наивысшим приоритетом, а единица в старшем разряде байта указывает на нали- чие запроса на прерывание INT от этого устройства. После считывания байта сло- ва состояния ПКП в МПр в БИС ПКП автоматически снимается режим программ- ного опроса. Обслуживание запроса заключается в программном декодировании информации, получаемой от ПКП, и выполнении соответствующей подпрограм- мы, реализующей обмен между МПр и выбранным устройством. При этом все функции по сохранению необходимой информации в стеке и формированию ад- реса ППОП возлагаются на программиста.

D7 D6 D5 D4 D3 D2 D1 D0

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I NT | | X | X | X | X | W2 | W1 | | W0 |
|  |  | | | | | | |  | |
|  | | | | | | | | |

Номер приоритетного запроса

1 - наличие запроса на прерывание

0 - отсутствие запроса на прерывание

Рисунок 3.12 - Формат слова состояния ПКП для режима программного опроса

Режим последовательного опроса обычно применяется в тех случаях, когда существует единая программа обработки прерывания для нескольких запросов IRQ, а также для расширения числа уровней IRQ (более 64).

CTI ; IF:=0 запрещение прерываний в микропроцессоре

. . . . . . . . . .

MOV AL, 01001110b ; OCW3=01001110b, AC:=OCW3

OUT 30h, AL ; А0=0, запись в ПКП OCW3, бит Р=D2=1

. . . . . . .. . . . . .

IN AL, 30h ; чтение в МПр слова состояния ПКП,

; OCW3:=01001010b - снятие режима ПО.

Далее выполняется анализ бита D7 в AL и если D7=1, то производится об- ращение к ППОП по команде CALL.

1. **Ответ в 10-м**
2. **От каких параметров зависит количество управляющих слов начальной инициализации ICW, загружаемых в ПКП?**

Запись управляющих слов для различных каналов можно производить в любой последовательности. Из таблицы 4.2 видно, что в за- висимости от комбинаций управляющих сигналов на входах ~CS, ~RD, ~WR, A1, A0 к шине данных D7-D0 подключаются различные каналы таймера и выполняет- ся запись либо управляющего слова CW, либо начальное значение счета в RgB, а считывать либо текущее значение счета из счетчиков, либо слово состояния каж- дого канала таймера.

ICW (Initialization Control Word) — управляющее слово инициализации. Всего имеются четыре таких слова с жесткой внутренней структурой — ICW1...ICW4. Эти слова предназначены для задания режима работы контроллера. Количество этих слов (4) определено количеством режимов: FNM, ARM, SRM, PM.

1. **Для чего нужна вложенность прерываний? Опишите режим работы «спецмаскирование» ПКП ВН59А**

При вложенных прерываниях, процедура обработки текущего прерывания может быть прервана (отложена) при поступлении запроса на прерывание, имеющего более высокий уровень приоритета. После обработки прерывания с более высоким уровнем приоритета процессор возвращается к прерванной процедуре и продолжает обработку данного прерывания до ее окончания или до нового прерывания. Очевидно, что процедура обработки прерывания с более высоким уровнем может быть в свою очередь прервана прерыванием с еще более высоким уровнем приоритета и т.д. При этом прерывания, имеющие более низкий уровень приоритета по сравнению с текущим, обычно запрещаются (маскируются).

Режим специального маскирования служит для того, чтобы разрешить пре- рывания от источников с более низким приоритетом, чем у запросов, находящих- ся в обработке в текущий момент времени. В данном режиме каждый бит в реги- стре обслуживания ISR запрещает только собственный уровень, но разрешает все остальные (как с более высоким, так и с более низким приоритетом), т.е. в общем случае регистр маски IMR совместно с регистром обслуживания ISR выполняют функцию маскирования поступающих запросов IRQ. В этом режиме запросы, по- ступившие на входы с более низкими приоритетами, обслуживаются до тех пор, пока не будет отменен режим специального маскирования. Формат команды OCW3 приведен на рисунке 3.11, который определяет режимы специального мас- кирования и последовательного опроса (считываемых прерываний), а также воз- можности чтения и выдачи в микропроцессор состояния регистра запросов пре- рываний IRR и регистра обслуживания ISR.

Режим специального маскирования назначается динамически после загруз- ки команды OCW3 и начала обслуживания запроса. Для перехода в режим специ- ального маскирования необходимо выполнить следующие команды:

MOV AL, 6Ah ; OCW3=01101010b, AC:=OCW3

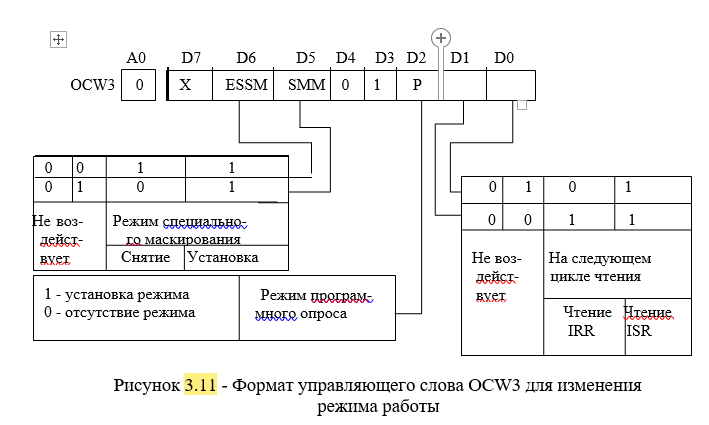
OUT 30h, AL ; A0=0, запись OCW3 в ПКП с адресом 30h.

Выйти из режима можно по командам:

MOV AL, 4Ah ; OCW3=01001010b, AC:=OCW3

OUT 30h, AL ; A0=0, запись OCW3 в ПКП с адресом 30h.

В ППОП при установленном режиме специального маскирования сброс ISR-бита при возврате из прерывания выполняется по команде специфического конца прерывания OCW2 формата SEOI (рисунок 3.10), в которой номер сбрасы- ваемого ISR-бита указывается непосредственно в команде (поле L2-L0). Исполь- зование других форматов команд OCW2, например, неспецифического конца пре- рывания (сброс ISR-бита с наивысшим приоритетом) приведет к сбросу ISR-бита чужой ППОП. Например, если выполнение ППОП3 было прервано запросом с более низким приоритетом IRQ5, то при возврате из ППОП5 в ISR-регистре будет сброшен третий бит, а не пятый.



Следует заметить, что во время начальной установки ПКП (при загрузке команды ICW1) режим специального маскирования автоматически аннулируется.

1. **В чем отличие последовательности вызова ППОП при использовании вектора прерывания и без него?**

Режим программного опроса используется для определения источника пре- рывания непосредственно программистом путем последовательного опроса ис- точников запросов на прерывание IRQ многократной загрузкой управляющего слова OCW3 и чтением в МПр слова состояния ПКП. Для реализации этого режи- ма предварительно прерывания в МПр должны быть запрещены (в регистре слова состояния процессора флаг разрешения прерываний IF должен быть сброшен). Тогда в ответ на поступление сигнала прерывания INT микропроцессор игнориру- ет выдачу сигналов "Подтверждения прерывания" ~INTA, а функции опроса со- стояния аппаратуры прерывания возлагаются на программиста.

Следовательно, для перехода в режим последовательного опроса необходи- мо при инициализации или перед установкой режима программного опроса запре- тить прерывания в МПр, установить режим программного опроса загрузкой ко- манды OCW3, а затем прочитать в МПр при А0=0 слово состояния ПКП в форма- те, представленном на рисунке 3.12. Младшие разряды считанного байта слова состояния ПКП указывают номер внешнего устройства (вектора прерывания) с наивысшим приоритетом, а единица в старшем разряде байта указывает на нали- чие запроса на прерывание INT от этого устройства. После считывания байта сло- ва состояния ПКП в МПр в БИС ПКП автоматически снимается режим программ- ного опроса. Обслуживание запроса заключается в программном декодировании информации, получаемой от ПКП, и выполнении соответствующей подпрограм- мы, реализующей обмен между МПр и выбранным устройством. При этом все функции по сохранению необходимой информации в стеке и формированию ад- реса ППОП возлагаются на программиста.

D7 D6 D5 D4 D3 D2 D1 D0

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I NT | | X | X | X | X | W2 | W1 | | W0 |
|  |  | | | | | | |  | |
|  | | | | | | | | |

Номер приоритетного запроса

1 - наличие запроса на прерывание

0 - отсутствие запроса на прерывание

Рисунок 3.12 - Формат слова состояния ПКП для режима программного опроса

Режим последовательного опроса обычно применяется в тех случаях, когда существует единая программа обработки прерывания для нескольких запросов IRQ, а также для расширения числа уровней IRQ (более 64).

CTI ; IF:=0 запрещение прерываний в микропроцессоре

. . . . . . . . . .

MOV AL, 01001110b ; OCW3=01001110b, AC:=OCW3

OUT 30h, AL ; А0=0, запись в ПКП OCW3, бит Р=D2=1

. . . . . . .. . . . . .

IN AL, 30h ; чтение в МПр слова состояния ПКП,

; OCW3:=01001010b - снятие режима ПО.

Далее выполняется анализ бита D7 в AL и если D7=1, то производится об- ращение к ППОП по команде CALL.