

Режимы обмена информацией по системной магистрали

Материалы по дисциплине «Микропроцессорные системы»
Специальность «Компьютерные системы и комплексы»
Составитель: Торгашин Р.Г

ГБПОУ ВО "Борисоглебский техникум промышленных и информационных технологий"

2016 год

Режимы обмена информацией.

Микропроцессорная система обеспечивает большую гибкость работы, она способна настраиваться на любую задачу. Гибкость эта обусловлена прежде всего тем, что функции, выполняемые системой, определяются программой (программным обеспечением, software), которую выполняет процессор. Аппаратура (аппаратное обеспечение, hardware) остается неизменной при любой задаче.

Но гибкость микропроцессорной системы определяется не только этим. Настраиваться на задачу помогает еще и выбор режима работы системы, то есть режима обмена информацией по системной магистрали (шине).

Практически любая развитая микропроцессорная система (в том числе и компьютер) поддерживает три основных режима обмена по магистрали:

- программный обмен информацией;
- обмен с использованием прерываний (Interrupts);
- обмен с использованием прямого доступа к памяти (ПДП, DMA — Direct Memory Access).

Программный обмен

Программный обмен информацией является основным в любой микропроцессорной системе. Он предусмотрен всегда, без него невозможны другие режимы обмена. В этом режиме процессор является единоличным хозяином (или задатчиком, Master) системной магистрали. Все операции (циклы) обмена информацией в данном случае инициируются только процессором, все они выполняются строго в порядке, предписанном исполняемой программой. Такой метод часто называют PIO (Processor Input/Output)

Путь процессора по программе может быть линейным, циклическим, может содержать переходы (прыжки), но он всегда непрерывен и полностью находится под контролем процессора. Ни на какие внешние события, не связанные с программой, процессор не реагирует. Все сигналы на магистрали в данном случае контролируются процессором.

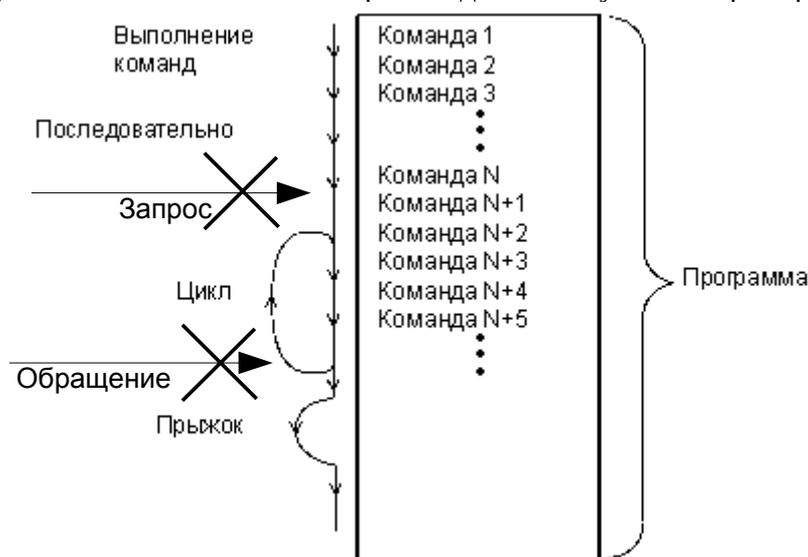


Рисунок 1: Программный обмен информацией

Взаимодействие процессора с «внешним миром» возможно только если оно закодировано в исполняемой программе. Таким образом, чтобы организовать реакцию на внешние события при программном обмене информацией нужно, чтобы процессор постоянно проверял факт наступления события. Такой метод называется методом опроса флага или polling.

Недостатком такого метода являются большие затраты процессорного времени на опрос флагов при отсутствии событий.

Проиллюстрируем данный метод:

Допустим вы готовите себе завтрак, поставив на плиту кипятиться молоко.

Естественно, на закипание молока надо реагировать, причем срочно. Как это организовать?

Первый путь — постоянно следить за молоком, но тогда вы ничего другого не сможете делать. Правильнее будет регулярно поглядывать на молоко, делая одновременно что-то другое. Это программный режим с опросом флага.

Обмен по прерываниям

Обмен по прерываниям позволяет оптимизировать механизм реакции процессора на внешние события.

Прерывание - «насильственный» перевода процессора с выполнения текущей программы на выполнение экстренно необходимой программы.

В рассмотренной выше ситуации с приготовлением завтрака, более простой путь— установить на кастрюлю с молоком датчик, который подаст звуковой сигнал при закипании молока, и спокойно заниматься другими делами. Правда, возможно, вам придется сначала закончить то, что вы начали делать, так что ваша реакция будет медленнее, чем в первом случае. Это и есть использование прерываний.

В режиме использования прерывания:

1. получив запрос прерывания от внешнего устройства (часто называемый IRQ — Interrupt ReQuest), процессор заканчивает выполнение текущей команды;
2. процессор сохраняет содержимое регистров;
3. процессор сохраняет в стек содержимое указателя команд (IP);
4. на основании вектора прерывания и таблицы векторов прерываний определяет адрес подпрограммы обработки прерываний
5. процессор переходит к программе обработки прерывания;
6. закончив выполнение программы обработки прерывания, он загружает сохраненное состояние и возвращается к прерванной программе с той точки, где его прервали.

Важно то, что вся работа, как и в случае программного режима, осуществляется самим процессором, внешнее событие просто временно отвлекает его. Реакция на внешнее событие по прерыванию в общем случае медленнее, чем при программном режиме.

Как и в случае программного обмена, здесь все сигналы на магистрали выставляются процессором, то есть он полностью контролирует магистраль. Для обслуживания прерываний в систему иногда вводится специальный модуль контроллера прерываний, но он в обмене информацией не участвует. Его задача состоит в том, чтобы упростить работу процессора с внешними запросами прерываний. Этот контроллер обычно программно управляется процессором по системной магистрали.

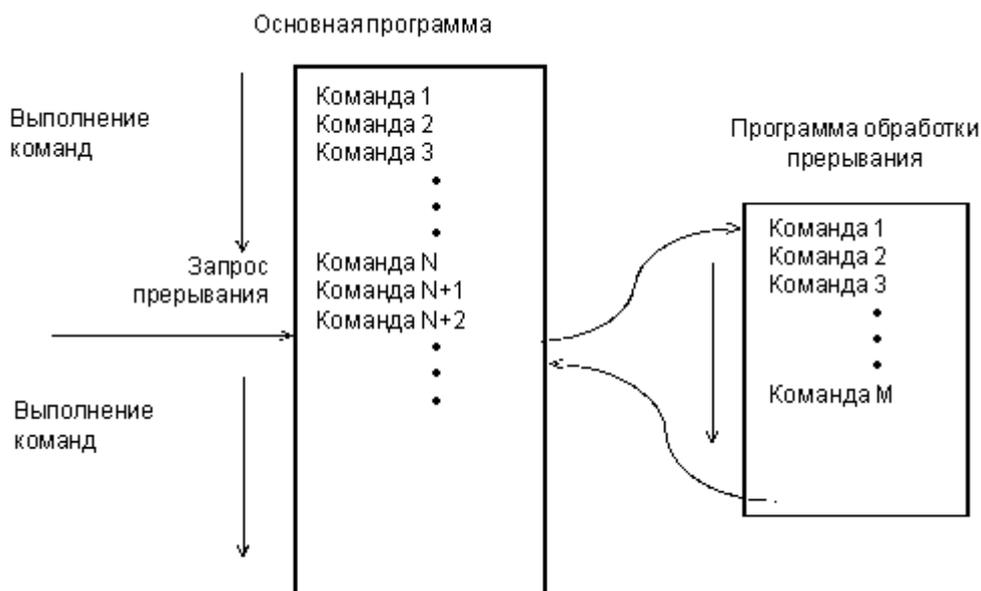


Рисунок 2: Обслуживание прерывания.

Естественно, события, вызвавшие прерывания могут быть более или менее важными для работы системы. Например ошибка чтения страницы оперативной памяти важнее чем событие нажатия клавиши на клавиатуре. Поэтому прерываниям назначаются приоритеты, определяющие порядок их обработки.

Приоритеты могут обслуживаться как относительные и абсолютные

- Относительное обслуживание прерываний означает, что если во время обработки прерывания поступает более приоритетное прерывание, то это прерывание будет обработано только после завершения текущей процедуры обработки прерывания.

- Абсолютное обслуживание прерываний означает, что если во время обработки прерывания поступает более приоритетное прерывание, то текущая процедура обработки прерывания вытесняется, и процессор начинает выполнять обработку вновь поступившего более приоритетного прерывания. После завершения этой процедуры процессор возвращается к выполнению вытесненной процедуры обработки прерывания.

В зависимости от источника возникновения сигнала прерывания делятся на:

- **асинхронные, или внешние** (аппаратные) — события, которые исходят от внешних источников (например, периферийных устройств) и могут произойти в любой произвольный момент: сигнал от таймера, сетевой карты или дискового накопителя, нажатие клавиш клавиатуры, движение мыши. Факт возникновения в системе такого прерывания трактуется как запрос на прерывание (англ. Interrupt request, IRQ);

- **синхронные, или внутренние** — события в самом процессоре как результат нарушения каких-то условий при исполнении машинного кода: деление на ноль или переполнение стека, обращение к недопустимым адресам памяти или недопустимый код операции;

- **программные** (частный случай внутреннего прерывания) — инициируются исполнением специальной инструкции в коде программы. Программные прерывания как правило используются для обращения к функциям встроенного программного обеспечения (firmware), драйверов и операционной системы.

Внешние прерывания в зависимости от возможности запрета делятся на:

- **маскируемые** — прерывания, которые можно запрещать установкой соответствующих битов в регистре маскирования прерываний (в x86-процессорах — сбросом флага IF в регистре флагов);

•**немаскируемые** (англ. Non-maskable interrupt, NMI) — обрабатываются всегда, независимо от запретов на другие прерывания. К примеру, такое прерывание может быть вызвано сбоем в микросхеме памяти.

Прерывания, которые не могут быть обработаны в настоящий момент обычно помещаются в очередь. За администрирование такой очереди отвечает сам процессор или контроллер прерываний.

В реальных вычислительных системах за администрирование прерываний могут выполнять два контроллера прерываний.

Прямой доступ к памяти

Прямой доступ к памяти (DMA, DMA) — это режим, принципиально отличающийся от двух ранее рассмотренных режимов тем, что обмен по системной шине идет без участия процессора. Внешнее устройство, требующее обслуживания, сигнализирует процессору, что режим DMA необходим, в ответ на это процессор заканчивает выполнение текущей команды и отключается от всех шин, сигнализируя запрошившему устройству, что обмен в режиме DMA можно начинать.

В аналогии с кипячением молока такой путь состоит в том, чтобы соединить датчик на кастрюле с управлением плитой так, чтобы при закипании молока горелка была выключена без вашего участия (правда, аналогия с DMA здесь не очень точная, так как в данном случае на момент выполнения действия вас не отвлекают от работы).

Операция DMA сводится к пересылке информации из устройства ввода/вывода в память или же из памяти в устройство ввода/вывода. Когда пересылка информации будет закончена, процессор вновь возвращается к прерванной программе, продолжая ее с той точки, где его прервали. Это похоже на режим обслуживания прерываний, но в данном случае процессор не участвует в обмене. Как и в случае прерываний, реакция на внешнее событие при DMA существенно медленнее, чем при программном режиме.

Понятно, что в этом случае требуется введение в систему дополнительного устройства (контроллера DMA), которое будет осуществлять полноценный обмен по системной магистрали без всякого участия процессора. Причем процессор предварительно должен сообщить этому контроллеру DMA, откуда ему следует брать информацию и/или куда ее следует помещать. Контроллер DMA может считаться специализированным процессором, который отличается тем, что сам не участвует в обмене, не принимает в себя информацию и не выдает ее.

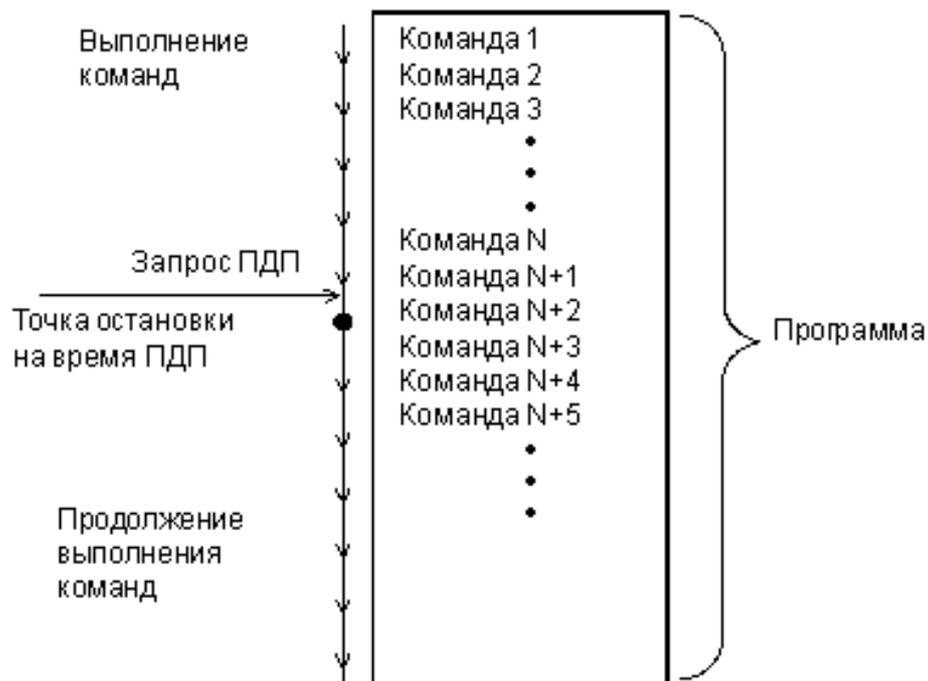


Рисунок 3: Обслуживание DMA



Рисунок 4: Информационные потоки в режиме DMA.

В принципе контроллер DMA может входить в состав устройства ввода/вывода, которому необходим режим DMA или даже в состав нескольких устройств ввода/вывода.

Для упрощения поиска устройств, при обмене информацией в режиме DMA, таким устройствам назначается канал DMA (так называемый номер канала DMA).

Обмен с помощью прямого доступа к памяти может обеспечить более высокую скорость передачи информации, чем программный обмен, так как процессор передает данные медленнее, чем специализированный контроллер DMA. Однако на практике это преимущество реализуется далеко не всегда. Скорость обмена в режиме DMA обычно ограничена возможностями магистрали. К тому же необходимость программного задания режимов контроллера DMA может свести на нет выигрыш от более высокой скорости пересылки данных в режиме DMA. Поэтому режим DMA применяется только там, где он необходим. Например этот режим используют все контроллеры ВЗУ в том числе контроллеры жестких дисков. А для контроллеров клавиатур такой режим не используется.

Если в системе уже имеется самостоятельный контроллер DMA, то это может в ряде случаев существенно упростить аппаратуру устройств ввода/вывода, работающих в режиме DMA.

На практике, хотя для жестких дисков обычно используется обмен DMA, иногда система может перевести устройство в режим PIO. Например, если при инициировании устройства обмен данными с ним был затруднен из-за дефектного шлейфа. В этом случае заметно сильное торможение при выполнении операций с файлами: запуске программ, открытии файлов.... А в описании ресурсов, используемых устройством, будет отсутствовать номер канала DMA.