

2. Организация обмена информацией: Шины микропроцессорной системы и циклы обмена

В этой лекции речь идет об обмене информацией по шинам микропроцессорных систем, о циклах обмена информацией и их фазах, о принципах синхронизации обмена, принципах организации прерываний и ПДП.

Самое главное, что должен знать разработчик микропроцессорных систем — это принципы организации обмена информацией по шинам таких систем. Без этого невозможно разработать аппаратную часть системы, а без аппаратной части не будет работать никакое программное обеспечение.

За более чем 30 лет, прошедших с момента появления первых микропроцессоров, были выработаны определенные правила обмена, которым следуют и разработчики новых микропроцессорных систем. Правила эти не слишком сложны, но твердо знать и неукоснительно соблюдать их для успешной работы необходимо. Как показала практика, принципы организации обмена по шинам гораздо важнее, чем особенности конкретных микропроцессоров. Стандартные системные магистрали живут гораздо дольше, чем тот или иной процессор. Разработчики новых процессоров ориентируются на уже существующие стандарты магистралей. Более того, некоторые системы на основе совершенно разных процессоров используют одну и ту же системную магистраль. То есть магистраль оказывается самым главным системообразующим фактором в микропроцессорных системах.

Обмен информацией в микропроцессорных системах происходит в циклах обмена информацией. Под циклом обмена информацией понимается временной интервал, в течение которого происходит выполнение одной элементарной операции обмена по шине. Например, пересылка кода данных из процессора в память или же пересылка кода данных из устройства ввода/вывода в процессор. В пределах одного цикла также может передаваться и несколько кодов данных, даже целый массив данных, но это встречается реже.

Циклы обмена информацией делятся на два основных типа:

- **Цикл записи (вывода)**, в котором процессор записывает (выводит) информацию;
- **Цикл чтения (ввода)**, в котором процессор читает (вводит) информацию.

В некоторых микропроцессорных системах существует также цикл «чтение-модификация-запись» или же «ввод-пауза-вывод». В этих циклах процессор сначала читает информацию из памяти или устройства ввода/вывода, затем как-то преобразует ее и снова записывает по тому же адресу. Например, процессор может прочитать код из ячейки памяти, увеличить его на единицу и снова записать в эту же ячейку памяти. Наличие или отсутствие данного типа цикла связано с особенностями используемого процессора.

Особое место занимают циклы прямого доступа к памяти (если режим ПДП в системе предусмотрен) и циклы запроса и предоставления прерывания (если прерывания в системе есть). Когда в дальнейшем речь пойдет о таких циклах, это будет специально оговорено.

Во время каждого цикла устройства, участвующие в обмене информацией, передают друг другу информационные и управляющие сигналы в строго установленном порядке или, как еще говорят, в соответствии с принятым **протоколом обмена информацией**.

Длительность цикла обмена может быть постоянной или переменной, но она всегда включает в себя несколько периодов сигнала тактовой частоты системы. То есть даже в идеальном случае частота чтения информации процессором и частота записи информации оказываются в несколько раз меньше тактовой частоты системы.

Чтение кодов команд из памяти системы также производится с помощью циклов чтения. Поэтому в случае одношинной архитектуры на системной магистрали чередуются циклы чтения команд и циклы пересылки (чтения и записи) данных, но протоколы обмена остаются неизменными независимо от того, что передается — данные или команды. В случае двухшинной архитектуры циклы чтения команд и записи или чтения данных разделяются по разным шинам и могут выполняться одновременно.

2.1. Шины микропроцессорной системы

Прежде чем переходить к особенностям циклов обмена, остановимся подробнее на составе и назначении различных шин микропроцессорной системы.

Как уже упоминалось, в системную магистраль (системную шину) микропроцессорной системы входит три основные информационные шины: *адреса*, *данных* и *управления*.

Шина данных — это основная шина, ради которой и создается вся система. Количество ее разрядов (линий связи) определяет скорость и эффективность информационного обмена, а также максимально возможное количество команд.

Шина данных всегда двунаправленная, так как предполагает передачу информации в обоих направлениях. Наиболее часто встречающийся тип выходного каскада для линий этой шины — выход с тремя состояниями.

Обычно *шина данных* имеет 8, 16, 32 или 64 разряда. Понятно, что за один цикл обмена по 64-разрядной шине может передаваться 8 байт информации, а по 8-разрядной — только один байт. Разрядность *шины данных* определяет и разрядность всей магистрали. Например, когда говорят о 32-разрядной системной магистрали, подразумевается, что она имеет 32-разрядную *шину данных*.

Шина адреса — вторая по важности шина, которая определяет максимально возможную сложность микропроцессорной системы, то есть допустимый объем памяти и, следовательно, максимально возможный размер программы и максимально возможный объем запоминаемых данных. Количество адресов, обеспечиваемых *шиной адреса*, определяется как 2^N , где N — количество разрядов. Например, 16-разрядная *шина адреса* обеспечивает 65 536 адресов. Разрядность *шины адреса* обычно кратна 4 и может достигать 32 и даже 64. *Шина адреса* может быть однонаправленной (когда магистралью всегда управляет только процессор) или двунаправленной (когда процессор может временно передавать управление магистралью другому устройству, например контроллеру ПДП). Наиболее часто используются типы выходных каскадов с тремя состояниями или обычные ТТЛ (с двумя состояниями).

Как в *шине данных*, так и в *шине адреса* может использоваться **положительная логика** или **отрицательная логика**. При положительной логике высокий уровень напряжения соответствует логической единице на соответствующей линии связи, низкий — логическому нулю. При отрицательной логике — наоборот. В большинстве случаев уровни сигналов на шинах — ТТЛ.

Для снижения общего количества линий связи магистрали часто применяется **мультиплексирование шин адреса и данных**. То есть одни и те же линии связи используются в разные моменты времени для передачи как адреса, так и данных (в начале цикла — адрес, в конце цикла — данные). Для фиксации этих моментов (стробирования) служат специальные сигналы на *шине управления*. Понятно, что мультиплексированная *шина адреса/данных* обеспечивает меньшую скорость обмена, требует более длительного цикла обмена (рис. 2.1). По типу *шины адреса* и *шины данных* все магистрали также делятся на мультиплексированные и немультимплексированные.

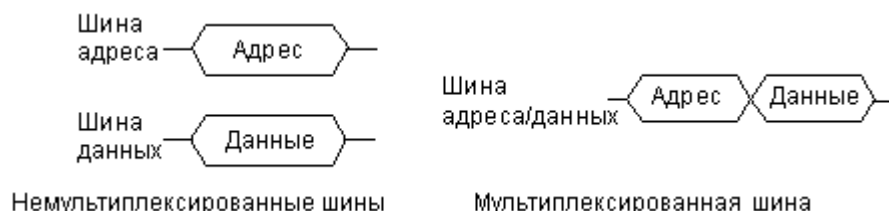


Рис. 2.1. Мультиплексирование шин адреса и данных.

В некоторых мультиплексированных магистралях после одного кода адреса передается несколько кодов данных (массив данных). Это позволяет существенно повысить быстродействие магистрали. Иногда в магистралях применяется частичное мультиплексирование, то есть часть разрядов данных передается по немультимплексированным линиям, а другая часть — по мультиплексированным с адресом линиям.

Шина управления — это вспомогательная шина, управляющие сигналы на которой определяют тип текущего цикла и фиксируют моменты времени, соответствующие разным частям или стадиям цикла. Кроме того, управляющие сигналы обеспечивают согласование работы процессора (или другого хозяина магистрали, задатчика, master) с работой памяти или

устройства ввода/вывода (устройства-исполнителя, slave). Управляющие сигналы также обслуживают запрос и предоставление прерываний, запрос и предоставление прямого доступа.

Сигналы *шины управления* могут передаваться как в положительной логике (реже), так и в отрицательной логике (чаще). Линии *шины управления* могут быть как однонаправленными, так и двунаправленными. Типы выходных каскадов могут быть самыми разными: с двумя состояниями (для однонаправленных линий), с тремя состояниями (для двунаправленных линий), с открытым коллектором (для двунаправленных и мультиплексированных линий).

Самые главные управляющие сигналы — это стробы обмена, то есть сигналы, формируемые процессором и определяющие моменты времени, в которые производится пересылка данных по *шине данных*, обмен данными. Чаще всего в магистрали используются два различных строба обмена:

- Строб записи (вывода), который определяет момент времени, когда устройство-исполнитель может принимать данные, выставленные процессором на *шину данных*;
- Строб чтения (ввода), который определяет момент времени, когда устройство-исполнитель должно выдать на *шину данных* код данных, который будет прочитан процессором.

При этом большое значение имеет то, как процессор заканчивает обмен в пределах цикла, в какой момент он снимает свой строб обмена. Возможны два пути решения (рис. 2.2):

- При *синхронном обмене* процессор заканчивает обмен данными самостоятельно, через раз и навсегда установленный временной интервал выдержки ($t_{\text{выд}}$), то есть без учета интересов устройства-исполнителя;
- При *асинхронном обмене* процессор заканчивает обмен только тогда, когда устройство-исполнитель подтверждает выполнение операции специальным сигналом (так называемый режим handshake — рукопожатие).

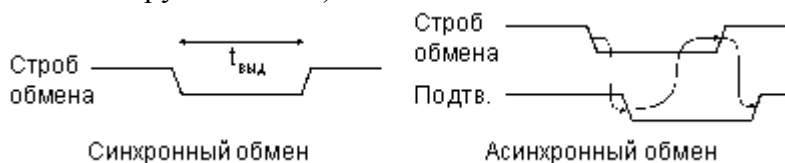


Рис. 2.2. Синхронный обмен и асинхронный обмен.

Достоинства *синхронного обмена* — более простой протокол обмена, меньшее количество управляющих сигналов. Недостатки — отсутствие гарантии, что исполнитель выполнил требуемую операцию, а также высокие требования к быстродействию исполнителя.

Достоинства *асинхронного обмена* — более надежная пересылка данных, возможность работы с самыми разными по быстродействию исполнителями. Недостаток — необходимость формирования сигнала подтверждения всеми исполнителями, то есть дополнительные аппаратурные затраты.

Какой тип обмена быстрее, *синхронный* или *асинхронный*? Ответ на этот вопрос неоднозначен. С одной стороны, при *асинхронном обмене* требуется какое-то время на выработку, передачу дополнительного сигнала и на его обработку процессором. С другой стороны, при *синхронном обмене* приходится искусственно увеличивать длительность строба обмена для соответствия требованиям большего числа исполнителей, чтобы они успевали обмениваться информацией в темпе процессора. Поэтому иногда в магистрали предусматривают возможность как *синхронного*, так и *асинхронного обмена*, причем *синхронный обмен* является основным и довольно быстрым, а *асинхронный* применяется только для медленных исполнителей.

По используемому типу обмена магистрали микропроцессорных систем также делятся на **синхронные и асинхронные**.

2.1. Циклы обмена по прерываниям

Циклы обмена в режиме прерываний строятся по тем же принципам, что и циклы программного обмена, но имеют ряд специфических особенностей.

Прерывания в микропроцессорных системах бывают двух основных типов:

- *векторные прерывания*, которые требуют проведения цикла чтения по магистрали;

- *радиальные прерывания*, которые не требуют никакого цикла обмена по магистрали.

Дело в том, что прерываний в микропроцессорной системе обычно бывает много. Поэтому процессору необходима информация о номере (или, как еще говорят, об адресе вектора) конкретного прерывания. Эта информация может быть передана процессору двумя путями.

При **векторном прерывании** код номера прерывания передается процессору тем устройством ввода/вывода, которое данное прерывание запросило. Для этого процессор проводит цикл чтения по магистрали, и по *шине данных* получает код номера прерывания. *Шина адреса* в данном цикле обычно не используется, так как устройство, запросившее прерывание, и так знает, что процессор будет обращаться именно к нему. В этом случае в магистрали достаточно всего одной линии *запроса прерывания* для всех устройств ввода/вывода. Так организованы прерывания, например, в магистрали Q-bus.

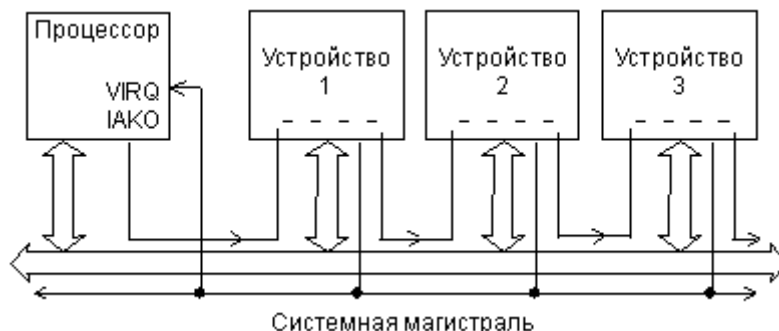


Рис. 2.8. Сигналы запроса и предоставления прерывания в магистрали Q-bus.

Схема распространения сигналов, участвующих в прерываниях на магистрали Q-bus, показана на [рис. 2.8](#). Упрощенная временная диаграмма цикла запроса и предоставления магистрали представлена на [рис. 2.9](#).



Рис. 2.9. Цикл запроса/предоставления векторного прерывания на магистрали Q-bus.

Запрос прерывания осуществляется отрицательным сигналом -VIRQ, который может формироваться каждым из устройств, запрашивающих прерывание. Тип выходного каскада для этого сигнала — ОК, чтобы избежать конфликтов между запрашивающими прерывания устройствами. Получив сигнал -VIRQ, процессор предоставляет прерывание (закончив предварительно выполнение текущей команды). Для этого он выставляет сигнал чтения данных -DIN и сигнал предоставления прерывания IAKO. Этот сигнал IAKO последовательно проходит через все устройства, которые могут запрашивать прерывания. Если устройство запросило прерывание, то оно не пропускает через себя этот сигнал. В результате получается, что если прерывания одновременно запросили два или более устройств, то сигнал предоставления прерывания получит только одно устройство, а именно то, которое ближе к процессору. Такой механизм разрешения конфликтов называется иногда географическим приоритетом (или цепочечным приоритетом, Daisy Chain). Получив сигнал IAKO, устройство, запросившее прерывание, должно снять свой сигнал -VIRQ.

Затем процессор проводит цикл безадресного чтения номера прерывания. В ответ на полученные сигналы -DIN и IAKO устройство, которому предоставлено прерывание, должно выдать на *шину адреса/данных* AD код номера прерывания (адреса вектора прерывания) и выставить сигнал подтверждения -RPLY. Процессор читает код номера прерывания и заканчивает цикл безадресного чтения снятием сигналов -DIN и IAKO.

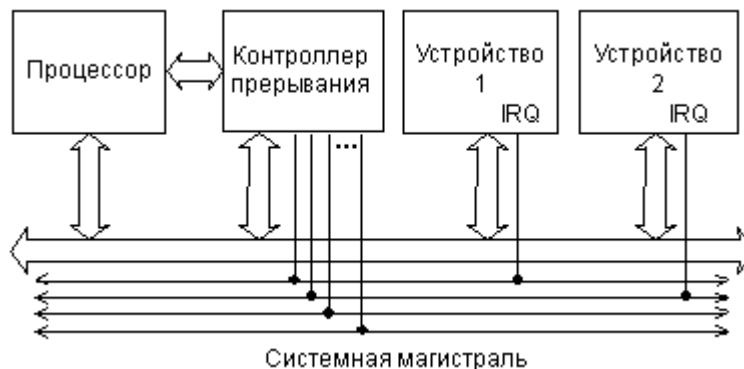


Рис. 2.10. Структура связей для организации радиальных прерываний на магистрали ISA.

При **радиальном прерывании** в магистрали имеется столько линий *запроса прерывания*, сколько всего может быть разных прерываний. То есть каждое устройство ввода/вывода, желающее использовать прерывание, подает сигнал *запроса прерывания* по своей отдельной линии. Процессор узнает о номере прерывания по номеру линии, по которой пришел сигнал *запроса прерывания*. Никаких циклов обмена по магистрали при этом не требуется. В случае *радиальных прерываний* в систему обычно включается дополнительная микросхема контроллера прерываний, обрабатывающая сигналы *запросов прерываний*. Именно так организованы прерывания, например, в магистрали ISA.

Упрощенная структура связей между устройствами, участвующими в обмене по прерываниям, на магистрали ISA показана на [рис. 2.10](#). Процессор общается с контроллером прерываний как по магистрали (чтобы задать ему режимы работы), так и вне магистрали (при обработке *запросов на прерывание*). Сигналы *запросов прерываний* IRQ распределяются между всеми устройствами магистрали. На каждую линию IRQ приходится одно устройство. Тип выходного каскада для этих линий — 2С, так как конфликты здесь не предусмотрены. *Запросом прерывания* является передний, положительный фронт сигнала IRQ. При одновременном поступлении сигналов IRQ от нескольких устройств порядок их обслуживания определяется контроллером прерываний.

Какой тип прерываний лучше — *векторный* или *радиальный*?

Векторные прерывания обеспечивают системе большую гибкость, в системе их может быть очень много. Но зато они требуют дополнительных аппаратных узлов во всех устройствах, запрашивающих прерывания, для обслуживания циклов безадресного чтения.

Радиальных прерываний в системе обычно не очень много (от 1 до 16). При этом типе прерываний, как правило, требуется введение в систему специального контроллера прерываний. Каждое *радиальное прерывание* требует введения дополнительной линии в *шину управления* системной магистрали. Но работать с *радиальными прерываниями* проще, так как все сводится только к выработке единственного сигнала IRQ, и никаких циклов обмена по магистрали не требуется.

2.3. Циклы обмена в режиме ПДП

Циклы обмена в режиме прямого доступа к памяти выполняются по тем же правилам, что и циклы программного обмена, и циклы предоставления прерываний.

Прежде чем начать обмен в режиме *ПДП*, устройство, которому необходим *ПДП*, должно запросить *ПДП* и получить его. Процедура запроса и предоставления *ПДП* очень похожа на процедуру запроса и предоставления прерывания. В обоих случаях устройство, требующее обслуживания, посылает сигнал запроса процессору. Однако в случае *ПДП* процессор обязательно должен предоставить *ПДП* запросившему устройству с помощью специальных сигналов, так как на время *ПДП* процессор отключается от магистрали. А при *радиальных прерываниях* предоставления прерывания от процессора не требуется.

На магистрали Q-bus запрос и предоставление *ПДП* организуются подобно запросу и предоставлению прерывания. Упрощенная структура связей устройств, участвующих в *ПДП*, показана на [рис. 2.11](#). Временная диаграмма запроса/предоставления *ПДП* очень близка к временной диаграмме запроса/предоставления прерывания (см. [рис. 2.9](#)). Сигнал запроса *ПДП*, называемый -DMR, передается всеми устройствами, нуждающимися в *ПДП*, по одной линии магистрали. Тип выходного каскада на этой линии — ОК. Процессор, получив сигнал -DMR, выдает сигнал предоставления *ПДП* DMGO, аналогичный сигналу IAKO. Этот сигнал также

проходит через все устройства последовательно, в результате чего *ПДП* получает только то устройство, которое находится ближе к процессору (географический приоритет). А затем устройство, получившее *ПДП*, проводит циклы обмена по магистрали, аналогично циклам программного обмена. В циклах *ПДП* информация читается из памяти и записывается в устройство ввода/вывода, или наоборот — читается из устройства ввода/вывода и передается в память.

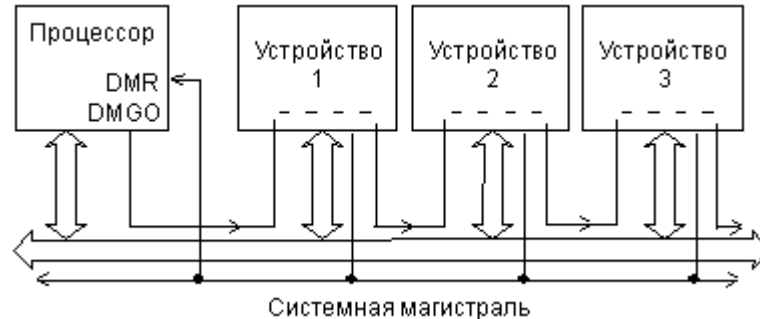


Рис. 2.11. Структура связей запроса/предоставления *ПДП* на магистрали Q-bus.

На магистрали ISA запрос/предоставление *ПДП* очень напоминает организацию *радиальных прерываний* (рис. 2.12). Точно так же в системе существует контроллер *ПДП*, к которому сходятся сигналы запроса *ПДП*, называемые DRQ, и от которого расходятся сигналы предоставления *ПДП*, называемые -DACK. К каждому каналу *ПДП* (пара сигналов DRQ и -DACK) подключается только одно устройство, запрашивающее *ПДП*. Тип выходных каскадов для этих сигналов — 2С. Устройство, нуждающееся в *ПДП*, посылает сигнал запроса DRQ и получает в ответ сигнал предоставления -DACK. После этого контроллер *ПДП* проводит циклы обмена по магистрали между устройством ввода/вывода и памятью.

2.3. Прохождение сигналов по магистрали

При организации обмена по магистралям и шинам разработчику необходимо учитывать несколько важных моментов, связанных как с особенностью распространения сигналов по шинам, так и с самой природой шин. В противном случае микропроцессорная система может попросту не работать или работать неустойчиво, хотя вся логика цифровых устройств, входящих в систему, будет спроектирована безошибочно.

В случае, когда системная шина (магистраль) микропроцессорной системы является внешней, а не скрыта внутри микросхемы, необходимо учитывать особенности распространения сигналов по длинным линиям. Хотя в большинстве случаев длина магистрали не слишком велика, не превышает 1—2 десятков сантиметров, это все равно оказывает большое влияние на синхронизацию обмена.

На прохождение сигналов по магистрали влияют следующие факторы:

- конечная величина задержки распространения сигналов по линиям магистрали;
- различие задержек распространения сигналов по разным линиям шины;
- одновременное выставление сигналов на линии шины;
- искажение фронтов сигналов, проходящих по линиям магистрали;
- отражение сигналов от концов линий связи (рис. 2.14).



Рис. 2.14. Прохождение сигналов по шине.

Для учета всех этих факторов разработчики стандартных магистралей обмена и стандартных протоколов обмена всегда закладывают необходимые задержки между сигналами,

участвующими в обмене. Кроме того, задержки между сигналами выбираются таким образом, чтобы устройство, которому адресован тот или иной сигнал, имело достаточно времени для его обработки. Если разрабатывается новая магистраль, все это тоже надо учитывать.

Поэтому пытаться «модернизировать» какой-то стандартный протокол и ускорять обмен по магистрали путем уменьшения задержек, предусмотренных стандартом, очень опасно. Точно так же опасно, не изменяя протокола обмена, пытаться увеличить длину магистрали, увеличивая тем самым задержки распространения сигналов по линиям и шинам. Особенно чувствительны к такого рода «модернизациям» синхронные магистрали, в которых не предусмотрено обязательное подтверждение выполнения каждой операции.

Например, длительность фазы адреса в цикле обмена выбирается таким образом. В течение адресной фазы все сигналы всех разрядов кода адреса, пусть даже и сформированные процессором не одновременно, должны дойти до устройства-исполнителя по своим проводам шины. А устройство-исполнитель должно этот код адреса принять и обработать (то есть отличить).