

1.4. Архитектура микропроцессорных систем

До сих пор мы рассматривали только один тип *архитектуры микропроцессорных систем* — *архитектуру* с общей, единой *шиной* для данных и команд (одношинную, или **принстонскую, фон-неймановскую архитектуру**). Соответственно, в составе системы в этом случае присутствует одна *общая память*, как для данных, так и для команд ([рис. 1.15](#)).



Рис. 1.15. Архитектура с общей шиной данных и команд.

Но существует также и альтернативный тип *архитектуры микропроцессорной системы* — это *архитектура* с *раздельными шинами* данных и команд (двухшинная, или гарвардская, *архитектура*). Эта *архитектура* предполагает наличие в системе *отдельной памяти* для данных и *отдельной памяти* для команд ([рис. 1.16](#)). Обмен процессора с каждым из двух типов *памяти* происходит по своей *шине*.

Архитектура с *общей шиной* распространена гораздо больше, она применяется, например, в персональных компьютерах и в сложных микрокомпьютерах. *Архитектура* с *раздельными шинами* применяется в основном в однокристалльных микроконтроллерах.

Рассмотрим некоторые достоинства и недостатки обоих архитектурных решений.

Архитектура с *общей шиной* (принстонская, фон-неймановская) проще, она не требует от процессора одновременного обслуживания двух *шин*, контроля обмена по двум *шинам* сразу. Наличие *единой памяти* данных и команд позволяет гибко распределять ее объем между кодами данных и команд. Например, в некоторых случаях нужна большая и сложная программа, а данных в *памяти* надо хранить не слишком много. В других случаях, наоборот, программа требуется простая, но необходимы большие объемы хранимых данных. Перераспределение *памяти* не вызывает никаких проблем, главное — чтобы программа и данные вместе помещались в *памяти* системы. Как правило, в системах с такой *архитектурой* *память* бывает довольно большого объема (до десятков и сотен мегабайт). Это позволяет решать самые сложные задачи.

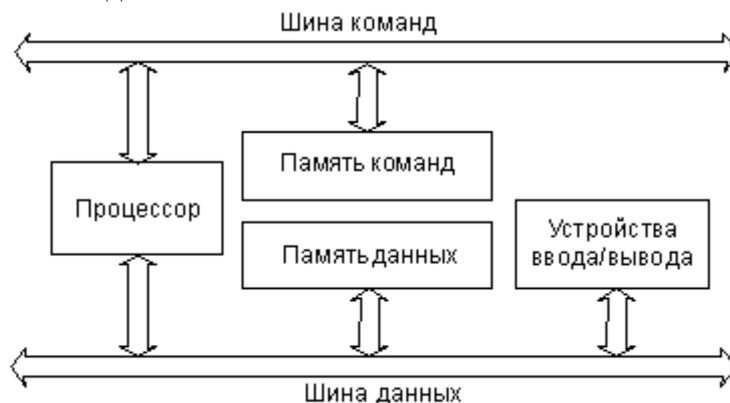


Рис. 1.16. Архитектура с раздельными шинами данных и команд.

Архитектура с *раздельными шинами* данных и команд сложнее, она заставляет процессор работать одновременно с двумя потоками кодов, обслуживать обмен по двум *шинам* одновременно. Программа может размещаться только в *памяти* команд, данные — только в *памяти* данных. Такая узкая специализация ограничивает круг задач, решаемых системой, так как не дает возможности гибкого перераспределения *памяти*. *Память* данных и *память* команд в этом случае имеют не слишком большой объем, поэтому применение систем с данной *архитектурой* ограничивается обычно не слишком сложными задачами.

В чем же преимущество *архитектуры* с двумя *шинами* (гарвардской)? В первую очередь, в быстродействии.

Дело в том, что при *единственной шине* команд и данных процессор вынужден по одной этой *шине* принимать данные (из *памяти* или *устройства ввода/вывода*) и передавать данные (в *память* или в *устройство ввода/вывода*), а также читать команды из *памяти*. Естественно, одновременно эти пересылки кодов по магистрали происходить не могут, они должны производиться по очереди. Современные процессоры способны совместить во времени выполнение команд и проведение циклов

обмена по системной *шине*. Использование конвейерных технологий и быстрой кэш-памяти позволяет им ускорить процесс взаимодействия со сравнительно медленной системной *памятью*. Повышение тактовой частоты и совершенствование структуры процессоров дают возможность сократить время выполнения команд. Но дальнейшее увеличение быстродействия системы возможно только при совмещении пересылки данных и чтения команд, то есть при переходе к *архитектуре* с двумя *шинами*.

В случае двухшинной *архитектуры* обмен по обоим *шинам* может быть независимым, параллельным во времени. Соответственно, структуры *шин* (количество разрядов кода адреса и кода данных, порядок и скорость обмена информацией и т.д.) могут быть выбраны оптимально для той задачи, которая решается каждой *шиной*. Поэтому при прочих равных условиях переход на двухшинную *архитектуру* ускоряет работу *микропроцессорной системы*, хотя и требует дополнительных затрат на аппаратуру, усложнения структуры процессора. *Память* данных в этом случае имеет свое распределение адресов, а *память* команд — свое.

Проще всего преимущества двухшинной *архитектуры* реализуются внутри одной микросхемы. В этом случае можно также существенно уменьшить влияние недостатков этой *архитектуры*. Поэтому основное ее применение — в микроконтроллерах, от которых не требуется решения слишком сложных задач, но зато необходимо максимальное быстродействие при заданной тактовой частоте.

1.5. Типы микропроцессорных систем

Диапазон применения микропроцессорной техники сейчас очень широк, требования к *микропроцессорным системам* предъявляются самые разные. Поэтому сформировалось несколько типов *микропроцессорных систем*, различающихся мощностью, универсальностью, быстродействием и структурными отличиями. Основные типы следующие:

- микроконтроллеры — наиболее простой тип *микропроцессорных систем*, в которых все или большинство узлов системы выполнены в виде одной микросхемы;
- контроллеры — управляющие *микропроцессорные системы*, выполненные в виде отдельных модулей;
- микрокомпьютеры — более мощные *микропроцессорные системы* с развитыми средствами сопряжения с внешними устройствами.
- компьютеры (в том числе персональные) — самые мощные и наиболее универсальные *микропроцессорные системы*.

Четкую границу между этими типами иногда провести довольно сложно. Быстродействие всех типов *микропроцессоров* постоянно растет, и нередки ситуации, когда новый микроконтроллер оказывается быстрее, например, устаревшего персонального компьютера. Но кое-какие принципиальные отличия все-таки имеются.

Микроконтроллеры представляют собой универсальные устройства, которые практически всегда используются не сами по себе, а в составе более сложных устройств, в том числе и контроллеров. Системная *шина* микроконтроллера скрыта от пользователя внутри микросхемы. Возможности подключения внешних устройств к микроконтроллеру ограничены. Устройства на микроконтроллерах обычно предназначены для решения одной задачи.

Контроллеры, как правило, создаются для решения какой-то отдельной задачи или группы близких задач. Они обычно не имеют возможностей подключения дополнительных узлов и устройств, например, большой *памяти*, средств ввода/вывода. Их системная *шина* чаще всего недоступна пользователю. Структура контроллера проста и оптимизирована под максимальное быстродействие. В большинстве случаев выполняемые программы хранятся в постоянной *памяти* и не меняются. Конструктивно контроллеры выпускаются в одноплатном варианте.

Микрокомпьютеры отличаются от контроллеров более открытой структурой, они допускают подключение к системной *шине* нескольких дополнительных устройств. Производятся микрокомпьютеры в каркасе, корпусе с разъемами системной магистрали, доступными пользователю. Микрокомпьютеры могут иметь средства хранения информации на магнитных носителях (например, магнитные диски) и довольно развитые средства связи с пользователем (видеомонитор, клавиатура). Микрокомпьютеры рассчитаны на широкий круг задач, но в отличие от контроллеров, к каждой новой задаче его надо приспособлять заново. Выполняемые микрокомпьютером программы можно легко менять.

Наконец, компьютеры и самые распространенные из них — персональные компьютеры — это самые универсальные из *микропроцессорных систем*. Они обязательно предусматривают возможность модернизации, а также широкие возможности подключения новых устройств. Их системная *шина*, конечно, доступна пользователю. Кроме того, внешние устройства могут подключаться к компьютеру

через несколько встроенных портов связи (количество портов доходит иногда до 10). Компьютер всегда имеет сильно развитые средства связи с пользователем, средства длительного хранения информации большого объема, средства связи с другими компьютерами по информационным сетям. Области применения компьютеров могут быть самыми разными: математические расчеты, обслуживание доступа к базам данных, управление работой сложных электронных систем, компьютерные игры, подготовка документов и т.д.

Любую задачу в принципе можно выполнить с помощью каждого из перечисленных типов *микропроцессорных систем*. Но при выборе типа надо по возможности избегать избыточности и предусматривать необходимую для данной задачи гибкость системы.

В настоящее время при разработке новых *микропроцессорных систем* чаще всего выбирают путь использования микроконтроллеров (примерно в 80% случаев). При этом микроконтроллеры применяются или самостоятельно, с минимальной дополнительной аппаратурой, или в составе более сложных контроллеров с развитыми средствами ввода/вывода.

Классические *микропроцессорные системы* на базе микросхем процессоров и микропроцессорных комплектов выпускаются сейчас довольно редко, в первую очередь, из-за сложности процесса разработки и отладки этих систем. Данный тип *микропроцессорных систем* выбирают в основном тогда, когда микроконтроллеры не могут обеспечить требуемых характеристик. Наконец, заметное место занимают сейчас *микропроцессорные системы* на основе персонального компьютера. Разработчику в этом случае нужно только оснастить персональный компьютер дополнительными устройствами сопряжения, а ядро *микропроцессорной системы* уже готово. Персональный компьютер имеет развитые средства программирования, что существенно упрощает задачу разработчика. К тому же он может обеспечить самые сложные алгоритмы обработки информации. Основные недостатки персонального компьютера — большие размеры корпуса и аппаратурная избыточность для простых задач. Недостатком является и неприспособленность большинства персональных компьютеров к работе в сложных условиях (запыленность, высокая влажность, вибрации, высокие температуры и т.д.). Однако выпускаются и специальные персональные компьютеры, приспособленные к различным условиям эксплуатации.