

Память

Второй основной составляющей любого компьютера является память. В идеале память должна быть максимально быстрой, довольно большой и чрезвычайно дешевой. Никакая современная технология не в состоянии удовлетворить все эти требования, поэтому используется другой подход. Система памяти создается в виде иерархии уровней. Верхние уровни обладают более высоким быстродействием, меньшим объемом и более высокой удельной стоимостью хранения одного бита информации, чем нижние уровни, иногда в миллиарды и более раз.

Обычное время доступа

Обычный объем

1 нс

Регистры

<1 Кбайт

2 нс

Кэш

4 Мбайт

10 нс

Основная память

512–2048 Мбайт

10 мс

Магнитный диск

200–1000 Гбайт



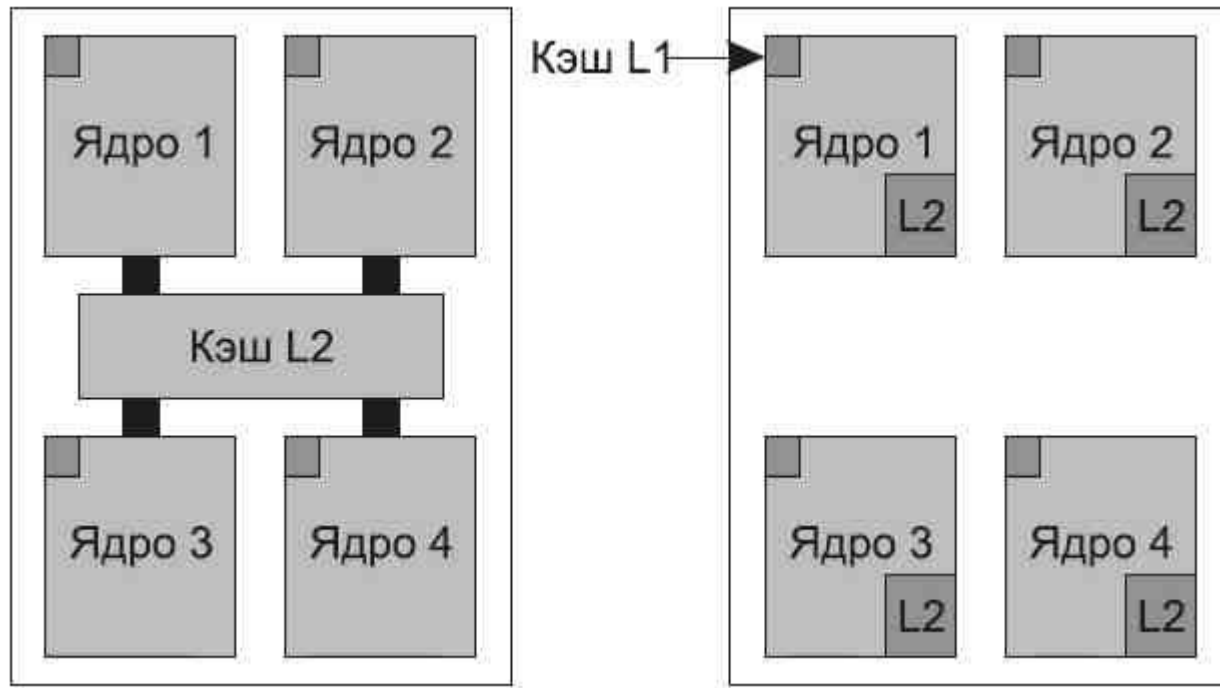
Верхний уровень состоит из внутренних регистров процессора. Они выполнены по той же технологии, что и сам процессор, и поэтому не уступают ему в быстродействии. Следовательно, к ним нет и задержек доступа. Внутренние регистры обычно предоставляют возможность для хранения 32×32 бит для 32-разрядного процессора или 64×64 бит — для 64-разрядного. В обоих случаях этот объем не превышает 1 Кбайт. Программы могут сами управлять регистрами (то есть решать, что в них хранить), без вмешательства аппаратуры.

Затем следует **кэш-память**, которая управляется главным образом аппаратурой. Оперативная память разделяется на **кэш-строки**, обычно по 64 байт, с адресами от 0 до 63 в кэш-строке 0, адресами от 64 до 127 в кэш-строке 1 и т. д. Наиболее интенсивно используемые кэш-строки оперативной памяти сохраняются в высокоскоростной кэш-памяти, находящейся внутри процессора или очень близко к нему. Когда программе нужно считать слово из памяти, аппаратура кэша проверяет, нет ли нужной строки в кэш-памяти. Если строка в ней имеется, то происходит результативное обращение к кэш-памяти, запрос удовлетворяется за счет кэш-памяти без отправки запроса по шине к оперативной памяти. Обычно результативное обращение к кэшу занимает по времени два такта. Отсутствие слова в кэш-памяти вынуждает обращаться к оперативной памяти, что приводит к существенной потере времени. Кэш-память из-за своей высокой стоимости ограничена в объеме. Некоторые машины имеют два или даже три уровня кэша, причем каждый из последующих медленнее и объемнее предыдущего.

В процессе кэширования строк оперативной памяти в кэше центрального процессора при каждом неудачном обращении к кэш-памяти в нее, как правило, будет вводиться новый элемент. При вычислении нужной кэш-строки для размещения нового элемента обычно используются некоторые старшие биты того адреса памяти, на который осуществляется ссылка. Например, при наличии 4096 кэш-строк по 64 байта и 32-разрядных адресов биты с 6-го по 17-й могли бы использоваться для определения кэш-строки, а биты с 0-го по 5-й — для определения байта в кэш-строке.

Многие современные процессоры имеют сразу два уровня кэш-памяти. Первый уровень, или **кэш L1**, всегда является частью самого процессора и обычно подает декодированные команды в процессорный механизм исполнения команд. У многих процессоров есть и второй кэш L1 для тех слов данных, которые используются особенно интенсивно. Обычно каждый из кэшей L1 имеет объем 16 Кбайт. Вдобавок к этому кэшу процессоры часто оснащаются вторым уровнем кэш-памяти, который называется **кэш L2** и содержит несколько мегабайт недавно использованных слов памяти. Различия между кэш-памятью L1 и L2 заключаются во временной диаграмме. Доступ к кэшу первого уровня осуществляется без задержек, а доступ к кэшу второго уровня требует задержки в один или два такта.

При разработке многоядерных процессоров конструкторам приходится решать, куда поместить кэш-память. На рисунке слева показан один кэш L2, совместно используемый всеми ядрами. Такой подход применяется в многоядерных процессорах корпорации Intel. Для сравнения на рисунке справа каждое ядро имеет собственную кэш-память L2. Именно такой подход применяет компания AMD.



Следующей в иерархии идет оперативная память. Это главная рабочая область системы памяти машины. Оперативную память часто называют оперативным запоминающим устройством (**ОЗУ**), или памятью с произвольным доступом (Random Access Memory (**RAM**)). В настоящее время блоки памяти имеют объем от сотен мегабайт до нескольких гигабайт, и этот объем стремительно растет. Все запросы процессора, которые не могут быть удовлетворены кэш-памятью, направляются к оперативной памяти.

Дополнительно к оперативной памяти многие компьютеры оснащены небольшой по объему неизменяемой памятью с произвольным доступом — постоянным запоминающим устройством (**ПЗУ**), оно же память, предназначенная только для чтения (Read Only Memory (**ROM**)). В отличие от ОЗУ она не утрачивает своего содержимого при отключении питания, то есть является энергонезависимой. ПЗУ программируется на предприятии-изготовителе и впоследствии не подлежит изменению. Эта разновидность памяти характеризуется высоким быстродействием и дешевизной. На некоторых компьютерах в ПЗУ размещается начальный загрузчик, используемый для их запуска. Такой же памятью, предназначенной для осуществления низкоуровневого управления устройством, оснащаются некоторые контроллеры ввода-вывода.

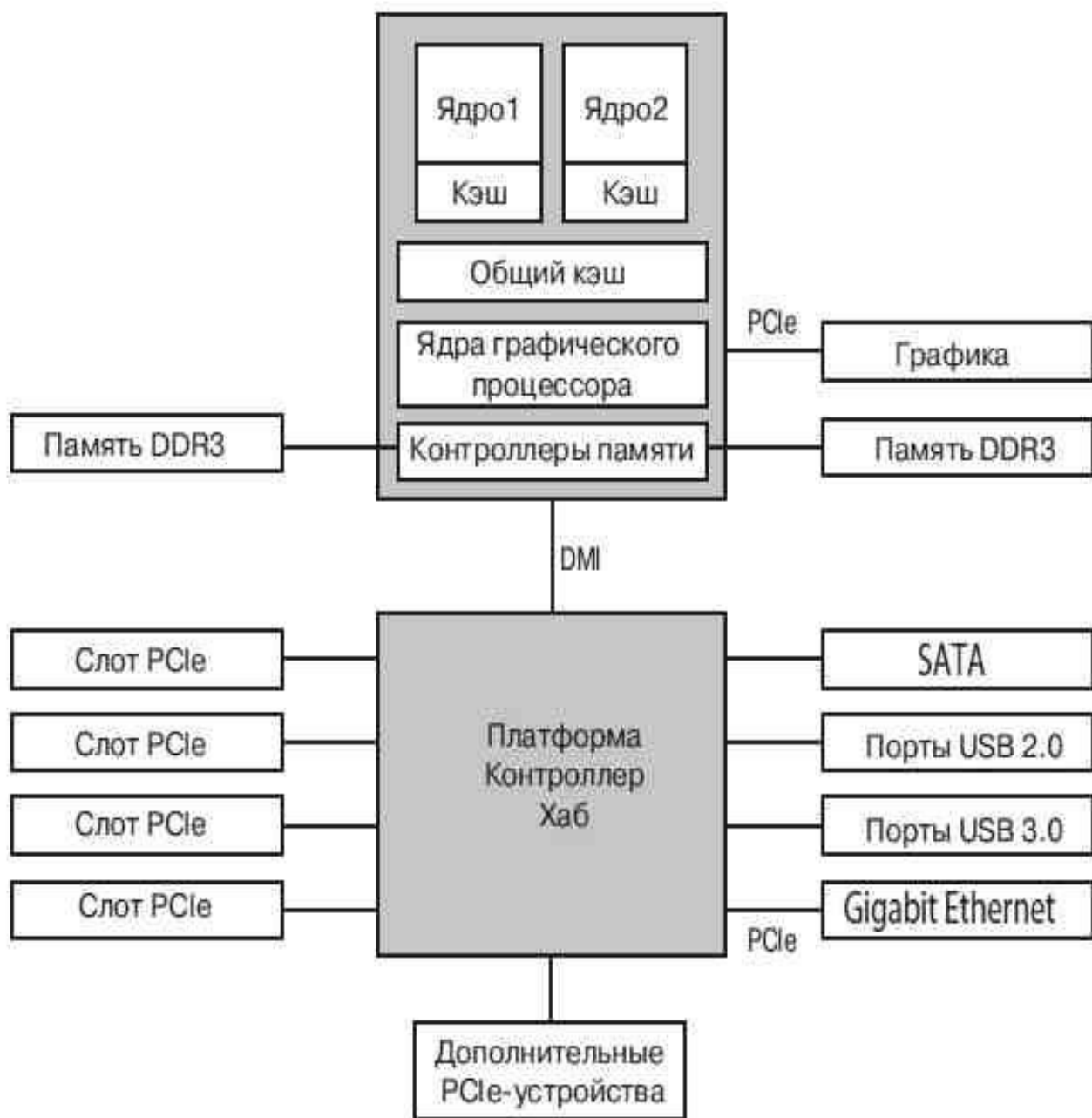
Существуют также другие разновидности энергонезависимой памяти, которые в отличие от ПЗУ могут стираться и перезаписываться, — электрически стираемые программируемые постоянные запоминающие устройства (**ЭСПЗУ**), они же **EEPROM** (Electrically Erasable PROM), и флеш-память. Однако запись в них занимает на несколько порядков больше времени, чем запись в ОЗУ, поэтому они используются для тех же целей, что и ПЗУ. При этом они обладают еще одним дополнительным свойством — возможностью исправлять ошибки в содержащихся в них программах путем перезаписи занимаемых ими участков памяти.

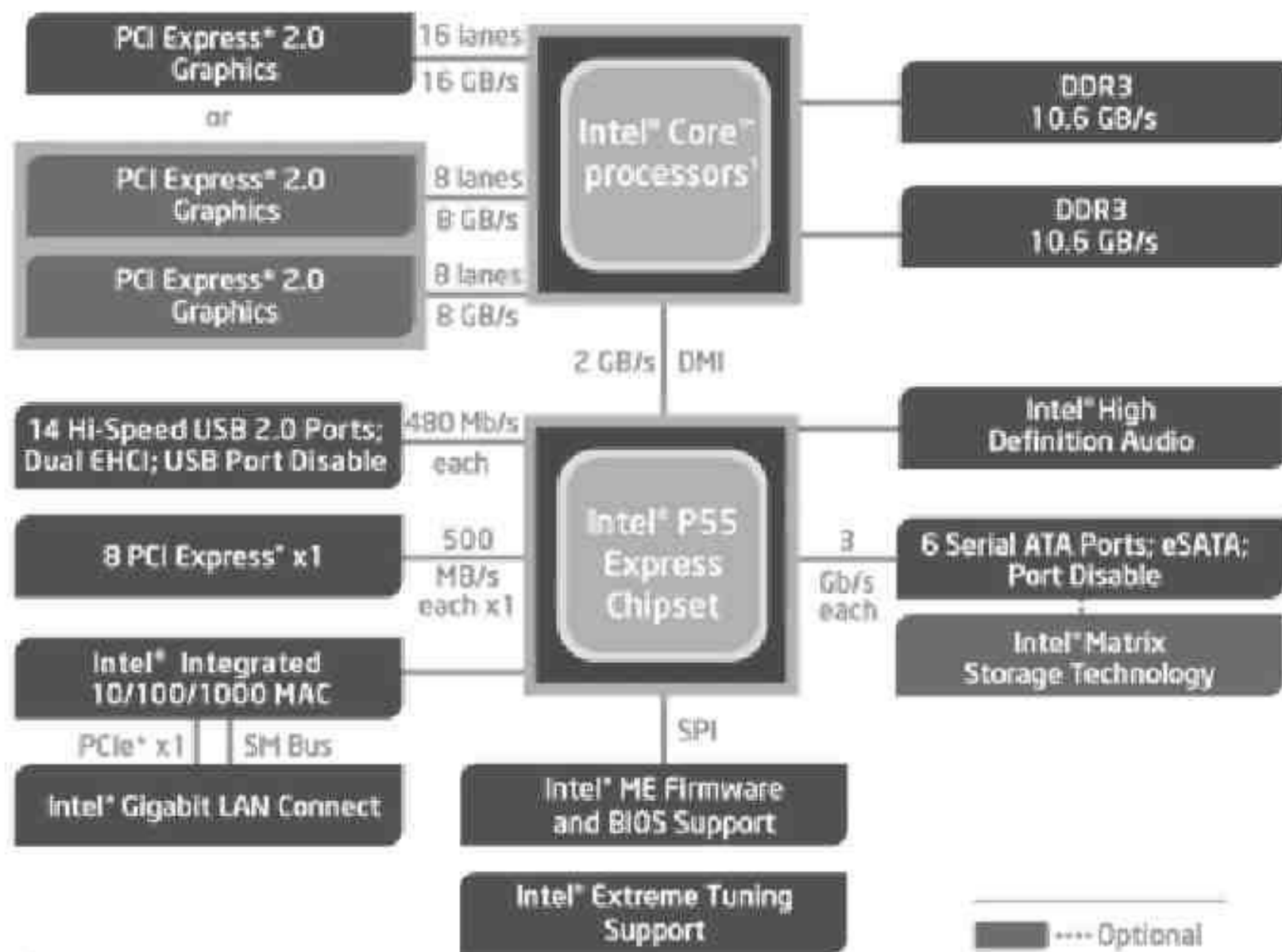
Флеш-память также обычно используется как носитель информации в портативных электронных устройствах.. По быстродействию флеш-память занимает промежуточное положение между ОЗУ и диском. Также, в отличие от дисковой памяти, если флеш-память стирается или перезаписывается слишком часто, она приходит в негодность.

Еще одна разновидность памяти — CMOS-память, которая является энергозависимой. Во многих компьютерах CMOS-память используется для хранения текущих даты и времени. CMOS-память и схема электронных часов, отвечающая за отсчет времени, получают питание от миниатюрной батарейки (или аккумулятора), поэтому значение текущего времени исправно обновляется, даже если компьютер отсоединен от внешнего источника питания. CMOS-память также может хранить параметры конфигурации, указывающие, например, с какого диска системе следует загружаться. Потребление энергии CMOS-памятью настолько низкое, что батарейки, установленной на заводе-изготовителе, часто хватает на несколько лет работы.

Шины

По мере увеличения скорости работы процессоров и памяти возможности единой шины по обеспечению всех процессов обмена данными достигли своего предела. В результате появились дополнительные шины как для более быстродействующих устройств ввода-вывода, так и для обмена данными между процессором и памятью. Вследствие этой эволюции массовая x86-система на данный момент имеет следующий вид.





¹ Compatible with:

У этой системы имеется множество шин (например, шина кэш-памяти, шина памяти, а также шины PCIe, PCI, USB, SATA и DMI), каждая из которых имеет свою скорость передачи данных и свое предназначение. Операционная система для осуществления функций настройки и управления должна знать обо всех этих шинах. Основной шиной является PCI (Peripheral Component Interconnect — интерфейс периферийных устройств).

Шина PCIe была придумана Intel в качестве преемницы более старой шины PCI, которая в свою очередь пришла на замену исходной шине ISA (Industry Standard Architecture — стандартная промышленная архитектура). Благодаря возможности передавать данные со скоростью в десятки гигабит в секунду шина PCIe работает намного быстрее своих предшественниц. Она сильно отличается от них и по своей природе. Вплоть до ее создания в 2004 году большинство шин были параллельными и совместно используемыми. Архитектура шин совместного использования означает, что для передачи данных разными устройствами используются одни и те же проводники. Таким образом, когда данные для передачи имеются сразу у нескольких устройств, для определения устройства, которому будет позволено использовать шину, требуется арбитр. В отличие от этого шина PCIe использует выделенные непосредственные соединения типа «точка — точка».

Архитектура параллельной шины, подобная той, что используется в PCI, предполагает отправку каждого слова данных по нескольким проводникам. Например, в обычных шинах PCI одно 32-разрядное число отправляется по 32 параллельным проводникам. В отличие от этого в PCIe используется архитектура последовательной шины, и все биты сообщения отправляются по одному соединению, известному как дорожка (lane), что очень похоже на отправку сетевого пакета. Это существенно упрощает задачу, поскольку обеспечивать абсолютно одновременное прибытие всех 32 битов в пункт назначения уже не нужно. Но параллелизм все же используется, поскольку параллельно могут действовать сразу несколько дорожек. Например, для параллельной передачи 32 сообщений могут использоваться 32 дорожки. Из-за быстрого роста скоростей передачи данных таких периферийных устройств, как сетевые карты и графические адаптеры, стандарт PCIe обновляется каждые 3–5 лет. Например, 16 дорожек PCIe 2.0 предлагали скорость 64 Гбит/с. Обновление до PCIe 3.0 удваивает эту скорость, а обновление до PCIe 4.0 — удваивает еще раз.

В данной конфигурации центральный процессор общается с памятью через быструю шину DDR3, со внешним графическим устройством — через шину PCIe, а со всеми остальными устройствами — через концентратор по шине **DMI** (Direct Media Interface — интерфейс непосредственной передачи данных). Концентратор в свою очередь соединяет все другие устройства, используя для обмена данными с USB-устройствами универсальную последовательную шину, для обмена данными с жесткими дисками и DVD-приводами — шину SATA и для передачи Ethernet-кадров — шину PCIe.

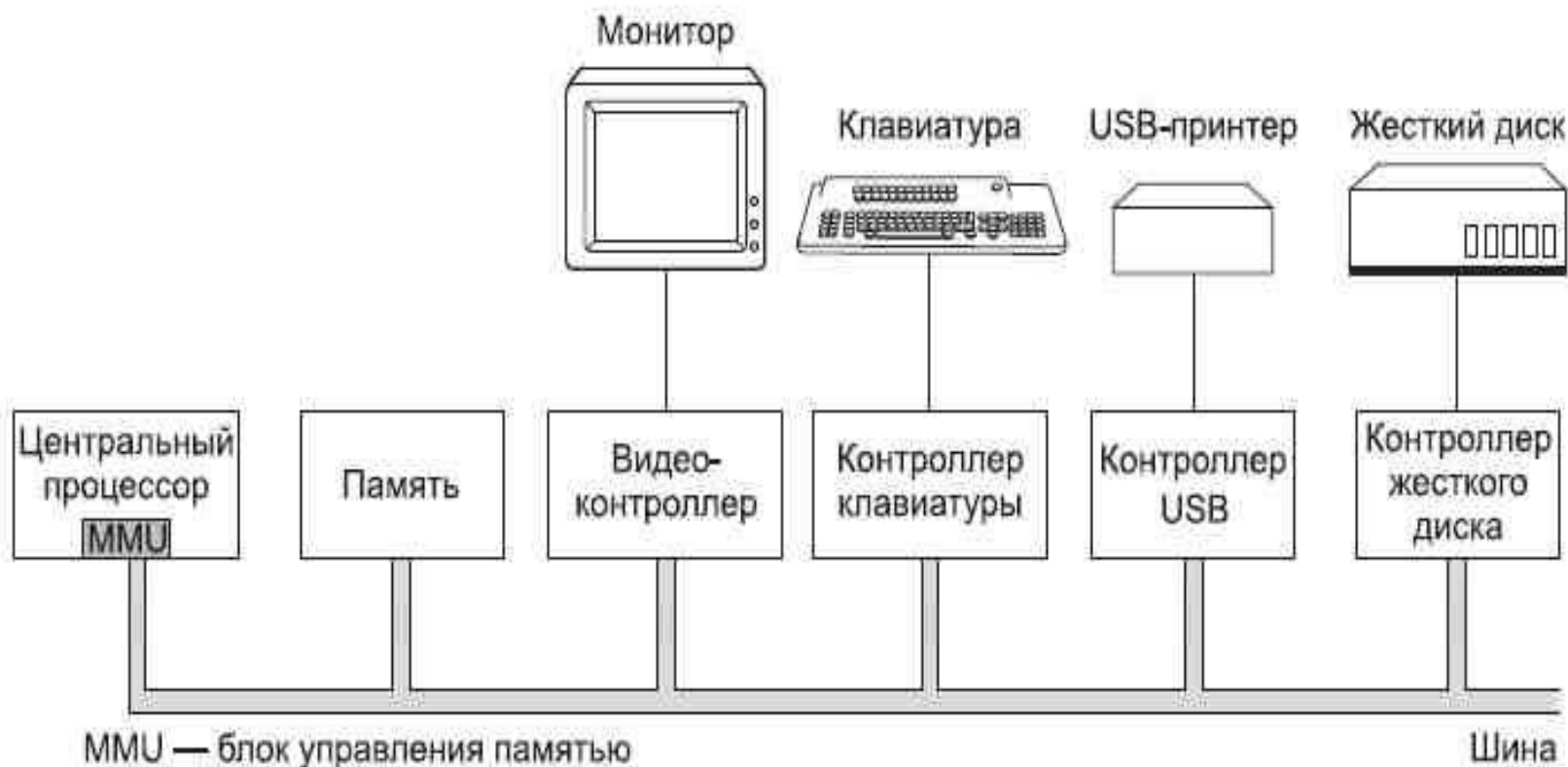
Шина **USB** (Universal Serial Bus — универсальная последовательная шина) была первоначально разработана для подключения к компьютеру всех низкоскоростных устройств ввода-вывода вроде клавиатуры и мыши. В USB используется небольшой разъем, имеющий (в зависимости от версии) от 4 до 11 контактов, часть из которых подводят к USB-устройствам питание или подключены к заземлению.

USB является централизованной шиной, в которой главное (корневое) устройство опрашивает устройства ввода-вывода каждую миллисекунду, чтобы узнать, есть ли у них данные для передачи. Стандарт USB 1.0 мог обеспечить совокупную скорость передачи данных 12 Мбит/с, в USB 2.0 скорость была поднята до 480 Мбит/с, а пиковая скорость в USB 3.0 составила никак не меньше 5 Гбит/с. Любое USB-устройство может быть подключено к компьютеру и приступить к работе немедленно, не требуя его перезагрузки.

SCSI (Small Computer System Interface — интерфейс малых вычислительных систем) является высокоскоростной шиной, предназначенной для высокопроизводительных дисков, сканеров и других устройств, нуждающихся в значительной пропускной способности. В наши дни эти шины встречаются в основном в серверах и рабочих станциях. Скорость передачи данных может достигать 640 Мбайт/с.

Устройства ввода-вывода

Центральный процессор и память не единственные ресурсы, которыми должна управлять операционная система. С ней также активно взаимодействуют и устройства ввода-вывода информации. Из следующего рисунка видно, что устройства ввода-вывода обычно состоят из двух компонентов: самого устройства и его контроллера.



Контроллер представляет собой микросхему или набор микросхем, которые управляют устройством на физическом уровне. Он принимает от операционной системы команды, например считать данные с помощью устройства, а затем их выполняет.

Довольно часто непосредственное управление устройством очень сложно и требует высокого уровня детализации, поэтому задачей контроллера является предоставление операционной системе простого интерфейса. Для осуществления всей этой работы контроллеры часто содержат маленькие встроенные компьютеры, запрограммированные на выполнение подобных задач.

Другим компонентом является само устройство. Устройства имеют довольно простые интерфейсы, поскольку они, во-первых, обладают весьма скромными возможностями, а во-вторых, должны отвечать общим стандартам. Соблюдение последнего условия необходимо для того, чтобы, к примеру, любой контроллер SATA-диска смог работать с любым SATA-диском. В наше время SATA является стандартным типом дисков на многих компьютерах. Поскольку интерфейс устройства скрыт его контроллером, все операционные системы видят только интерфейс контроллера, который может существенно отличаться от интерфейса самого устройства.

Так как все типы контроллеров отличаются друг от друга, для управления ими требуется различное программное обеспечение. Программа, предназначенная для общения с контроллером, выдачи ему команды и получения поступающих от него ответов, называется **драйвером устройства**.

Ввод и вывод данных можно делать тремя различными способами. В простейшем из них пользовательская программа производит системный вызов, который транслируется ядром в процедуру вызова соответствующего драйвера. После этого драйвер приступает к процессу ввода-вывода. В это время он выполняет очень короткий цикл, постоянно опрашивая устройство и отслеживая завершение операции (обычно занятость устройства определяется состоянием специального бита). По завершении операции ввода-вывода драйвер помещает данные (если таковые имеются) в нужное место, и возвращает управление. Затем операционная система возвращает управление вызывающей программе. Этот способ называется **активным ожиданием** или **ожиданием готовности**, а его недостаток заключается в том, что он загружает процессор опросом устройства об окончании работы.

Второй способ заключается в том, что драйвер запускает устройство и просит его выдать прерывание по окончании выполнения команды (завершении ввода или вывода данных). Сразу после этого драйвер возвращает управление. Затем операционная система блокирует вызывающую программу, если это необходимо, и переходит к выполнению других задач. Когда контроллер обнаруживает окончание передачи данных, он генерирует **прерывание**, чтобы просигнализировать о завершении операции.

Прерывания играют очень важную роль в работе операционной системы, поэтому рассмотрим их более подробно. На рисунке внизу показан процесс ввода-вывода.



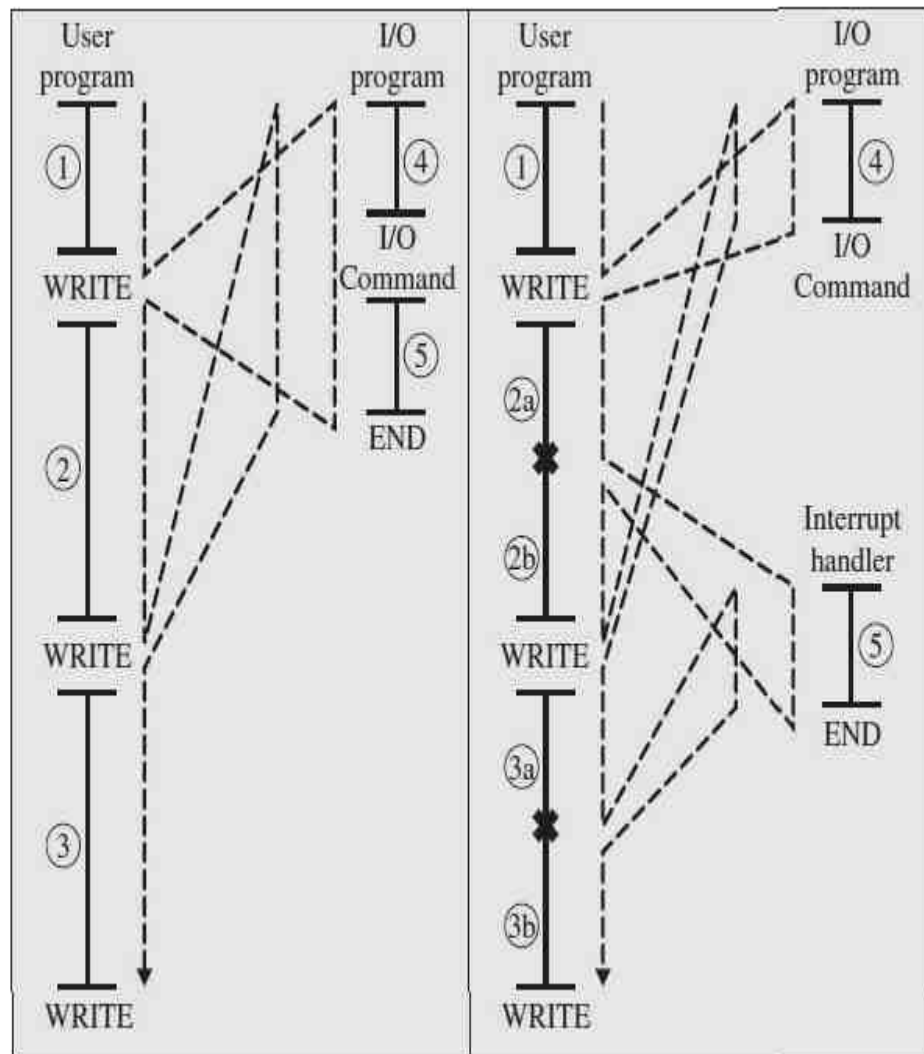
На первом этапе драйвер передает команду контроллеру, записывая информацию в его регистры. Затем контроллер запускает само устройство. На втором этапе, когда контроллер устройства завершает чтение или запись заданного ему количества байтов, он выставляет сигнал для микросхемы контроллера прерываний, используя для этого определенные линии шины. На третьем этапе, если контроллер прерываний готов принять прерывание (а он может быть и не готов к этому, если обрабатывает прерывание с более высоким уровнем приоритета), он выставляет сигнал на контакте микросхемы центрального процессора, информируя его о завершении операции. На четвертом этапе контроллер прерываний выставляет номер устройства на шину, чтобы процессор мог его считать и узнать, какое устройство только что завершило работу (поскольку одновременно могут работать сразу несколько устройств).

Как только центральный процессор решит принять прерывание, содержимое счетчика команд и слова состояния программы помещаются, как правило, в текущий стек и процессор переключается в режим ядра. Номер устройства может быть использован как индекс части памяти, используемой для поиска адреса обработчика прерываний данного устройства. Эта часть памяти называется **вектором прерываний**.

Когда обработчик прерываний (являющийся частью драйвера устройства, выдающего запрос на прерывание) начинает свою работу, он извлекает помещенные в стек содержимое счетчика команд и слова состояния программы и сохраняет их, а затем опрашивает устройство для определения его состояния. После завершения обработки прерывания обработчик возвращает управление ранее работавшей пользовательской программе — на первую же еще не выполненную команду.

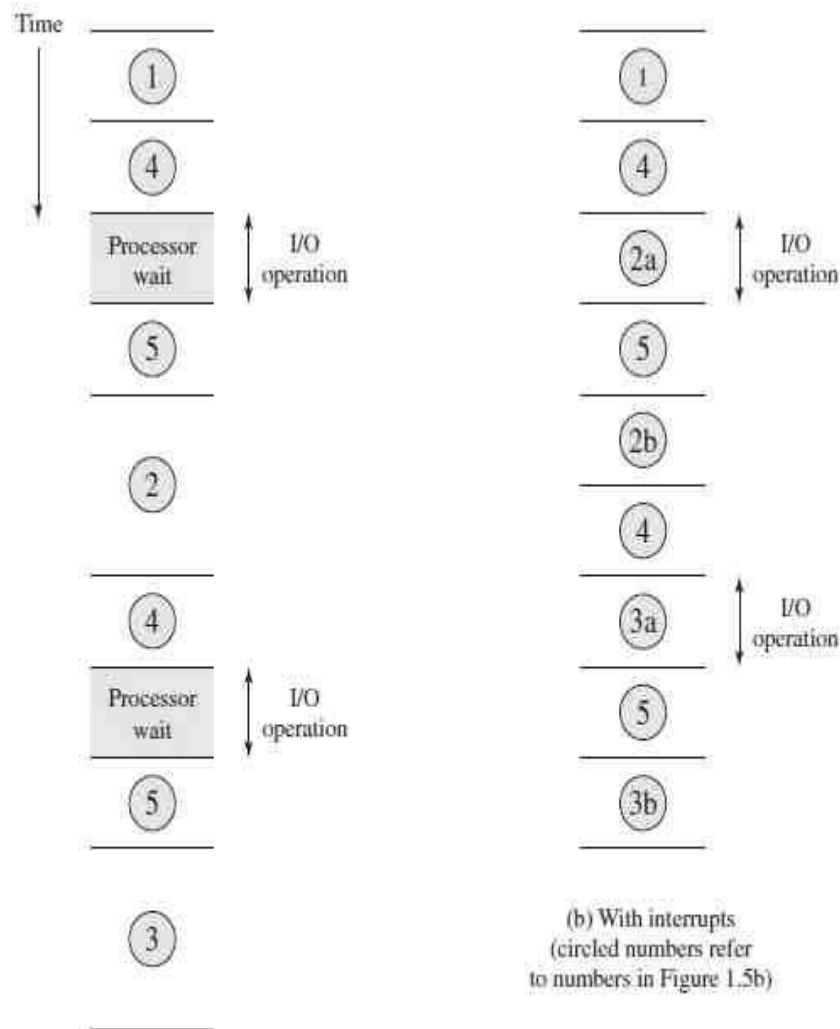
При третьем способе ввода-вывода используется специальный контроллер **прямого доступа к памяти** (Direct Memory Access (**DMA**)), который может управлять потоком битов между оперативной памятью и некоторыми контроллерами без постоянного вмешательства центрального процессора. Центральный процессор осуществляет настройку контроллера DMA, сообщая ему, сколько байтов следует передать, какое устройство и адреса памяти задействовать и в каком направлении передать данные, а затем дает ему возможность действовать самостоятельно. Когда контроллер DMA завершает работу, он выдает запрос на прерывание, который обрабатывается в ранее рассмотренном порядке.

Прерывания часто происходят в очень неподходящие моменты, например во время работы обработчика другого прерывания. Поэтому центральный процессор обладает возможностью запрещать прерывания с последующим их разрешением. Пока прерывания запрещены, любые устройства, закончившие свою работу, продолжают выставлять свои запросы на прерывание, но работа процессора не прекращается, пока прерывания снова не станут разрешены. Если за время запрещения прерываний завершится работа сразу нескольких устройств, контроллер решает, какое из них должно быть обработано первым, полагаясь обычно на статические приоритеты, назначенные каждому устройству. Побеждает устройство, имеющее наивысший приоритет, которое и обслуживается в первую очередь. Все остальные устройства должны ожидать своей очереди.



(a) No interrupts

(b) Interrupts; short I/O wait



(a) Without interrupts
(circled numbers refer
to numbers in Figure 1.5a)

(b) With interrupts
(circled numbers refer
to numbers in Figure 1.5b)