

Проект комиссии Президента
по модернизации и технологическому развитию экономики России
«Создание системы подготовки высококвалифицированных кадров
в области суперкомпьютерных технологий и
специализированного программного обеспечения»

Конспект лекций дисциплины

«Архитектура и программное обеспечение высокопроизводительных вычислительных систем»

Разработчик: к.ф.-м.н. Богословский Н.Н.

Москва

Тема № 2

Иерархическая организация памяти

Предположим, что на компьютере решается какая-нибудь очень большая задача, например, система линейных алгебраических уравнений с матрицей максимального размера. Если используется метод Гаусса, то исключать элементы можно в самых различных порядках. При этом общее число выполняемых арифметических операций остается одним и тем же или меняется не более чем в 2—3 раза. Однако, пропуская разные варианты метода для одной и той же системы, можно заметить, что общее время решения задачи меняется в десятки раз и более. Другая реальная ситуация. Пусть зафиксирован какой-либо вариант метода Гаусса, но теперь решаются системы различных размеров. При этом ожидается, что общее время решения задачи будет меняться пропорционально общему числу арифметических операций. Но и в этом случае встречаются неожиданности. Иногда для некоторых порядков время решения задачи может оказаться существенно большим.

На общее время решения задачи влияет много факторов. Но главные из них — это время выполнения арифметических операций и время взаимодействия с памятью. В методах типа Гаусса вся совокупность арифметических операций известна заранее. Поэтому, если время решения задачи почему-то неоправданно возрастает, то, скорее всего, есть определенная неаккуратность в использовании памяти.

Если для реализации алгоритма нужно не просто составить какую-нибудь программу, а добиться, чтобы она работала максимально быстро, то составитель программ должен хорошо знать архитектуру памяти и соблюдать вполне определенные правила работы с ней, что может дать ощутимый эффект в скорости работы программы. А это означает, что имеются причины для более глубокого обсуждения строения памяти и принципов ее использования. Тем более, если возникает необходимость решать большие задачи.

В любой ЭВМ, вне зависимости от ее архитектуры, программы и данные хранятся в памяти. Функции памяти обеспечиваются запоминающими устройствами (ЗУ), предназначенными для фиксации, хранения и выдачи информации в процессе работы ЭВМ. Процесс фиксации информации в ЗУ называется *записью*, процесс выдачи информации — *чтением* или *считыванием*, а совместно их определяют как *процессы обращения к ЗУ*.

Характеристики систем памяти

Перечень основных характеристик, которые необходимо учитывать, рассматривая конкретный вид ЗУ, включает в себя:

- месторасположения;
- емкость;
- единицу пересылки;
- метод доступа;
- быстродействие;
- физический тип;
- физические особенности;
- стоимость.

По месту расположения ЗУ разделяют на процессорные, внутренние и внешние. Наиболее скоростные виды памяти (регистры, кэш-память первого уровня) обычно размещают на общем кристалле с центральным процессором, а регистры общего назначения вообще считаются частью ЦП. Вторую группу (внутреннюю память) образуют ЗУ, расположенные на системной плате. К внутренней памяти относят основную память, а также кэш-память второго и последующих уровней (кэш-память второго уровня может также размещаться на кристалле процессора). Медленные ЗУ большой емкости (магнитные и оптические диски, магнитные ленты) называют внешней памятью, поскольку к ядру ЭВМ они подключаются аналогично устройствам ввода/вывода.

Емкость запоминающих устройств характеризуют числом битов либо байтов, которое может храниться в запоминающем устройстве. На практике применяются более крупные единицы, а для их обозначения к словам «бит» или «байт» добавляют приставки: кило, мега, гига, тера, пета, экса (kilo, mega, giga, tera, peta, exa). Стандартно эти приставки означают умножение основной единицы измерений на 10^3 , 10^6 , 10^9 , 10^{12} , 10^{15} и 10^{18} соответственно.

Важной характеристикой ЗУ является **единица пересылки**. Для основной памяти (ОП) единица пересылки определяется шириной шины данных, то есть количеством битов, передаваемых по линиям шины параллельно. Обычно единица пересылки равна длине слова, но не обязательно. Применительно к внешней памяти данные часто передаются единицами, превышающими размер слова, и такие единицы называются **блоками**.

При оценке быстродействия необходимо учитывать применяемый в данном типе ЗУ **метод доступа** к данным. Различают четыре основных метода доступа:

- **Последовательный доступ.** ЗУ с последовательным доступом ориентировано на хранение информации в виде последовательности блоков данных, называемых записями. Для доступа к нужному элементу (слову или байту) необходимо прочитать все предшествующие ему данные. Время доступа зависит от положения требуемой записи в последовательности записей на носителе информации и позиции

элемента внутри данной записи. Примером может служить ЗУ на магнитной ленте.

- **Прямой доступ.** Каждая запись имеет уникальный адрес, отражающий ее физическое размещение на носителе информации. Обращение осуществляется как адресный доступ к началу записи, с последующим последовательным доступом к определенной единице информации внутри записи. В результате время доступа к определенной позиции является величиной переменной. Такой режим характерен для магнитных дисков.
- **Произвольный доступ.** Каждая ячейка памяти имеет уникальный физический адрес. Обращение к любой ячейке занимает одно и то же время и может производиться в произвольной очередности. Примером могут служить запоминающие устройства основной памяти.
- **Ассоциативный доступ.** Этот вид доступа позволяет выполнять поиск ячеек содержащих такую информацию, в которой значение отдельных битов совпадает с состоянием одноименных битов в заданном образце. Сравнение осуществляется параллельно для всех ячеек памяти, независимо от ее емкости. По ассоциативному принципу построены некоторые блоки кэш-памяти.

Быстродействие ЗУ является одним из важнейших его показателей. Для количественной оценки быстродействия обычно используют три параметра:

- **Время доступа (T_d).** Для памяти с произвольным доступом оно соответствует интервалу времени от момента поступления адреса до момента, когда данные заносятся в память или становятся доступными. В ЗУ с подвижным носителем информации - это время, затрачиваемое на установку головки записи/считывания (или носителя) в нужную позицию.
- **Длительность цикла памяти или период обращения (T_c).** Понятие применяется к памяти с произвольным доступом, для которой оно означает минимальное время между двумя последовательными обращениями к памяти. Период обращения включает в себя время доступа плюс некоторое дополнительное время. Дополнительное время может требоваться для затухания сигналов на линиях, а в некоторых типах ЗУ, где считывание информации приводит к ее разрушению, - для восстановления считанной информации.
- **Скорость передачи.** Это скорость, с которой данные могут передаваться в память или из нее. Для памяти с произвольным доступом она равна $1/T_c$. Для других видов памяти скорость передачи определяется соотношением:

$$T_N = T_A + \frac{N}{R}$$

где T_N — среднее время считывания или записи N битов; T_A — среднее время доступа; R — скорость пересылки в битах в секунду.

Говоря о **физическом типе** запоминающего устройства, необходимо упомянуть три наиболее распространенных технологии ЗУ — это полупроводниковая память, память с магнитным носителем информации, используемая в магнитных дисках и лентах, и память с оптическим носителем — оптические диски.

В зависимости от примененной технологии следует учитывать и ряд **физических особенностей** ЗУ, например энергозависимость. В энергозависимой памяти информация может быть искажена или потеряна при отключении источника питания. В энергонезависимых ЗУ записанная информация сохраняется и при отключении питающего напряжения. Магнитная и оптическая память — энергонезависимы. Полупроводниковая память может быть как энергозависимой, так и нет, в зависимости от ее типа. Помимо энергозависимости нужно учитывать, приводит ли считывание информации к ее разрушению.

Стоимость ЗУ принято оценивать отношением общей стоимости ЗУ к его емкости в битах, то есть стоимостью хранения одного бита информации.

Иерархия запоминающих устройств

Память часто называют «узким местом» фон-неймановских ЭВМ из-за ее серьезного отставания по быстродействию от процессоров, причем разрыв этот неуклонно увеличивается. Так, если производительность процессоров ежегодно возрастает вдвое примерно каждые 1,5 года, то для микросхем памяти прирост быстродействия не превышает 9% в год (удвоение за 10 лет), что выражается в увеличении разрыва в быстродействии между процессором и памятью приблизительно на 50% в год.

Если проанализировать используемые в настоящее время типы ЗУ, выявляется следующая закономерность:

- чем меньше время доступа, тем выше стоимость хранения бита;
- чем больше емкость, тем ниже стоимость хранения бита, но больше время доступа.

При создании системы памяти постоянно приходится решать задачу обеспечения требуемой емкости и высокого быстродействия за приемлемую цену. Наиболее распространенным подходом здесь является построение системы памяти ЭВМ по иерархическому принципу. **Иерархическая память** состоит из ЗУ различных типов (рис. 1), которые, в зависимости от характеристик, относят к определенному уровню иерархии. Более высокий уровень меньше по емкости, быстрее и имеет большую стоимость в пересчете на бит, чем более низкий уровень. Уровни иерархии взаимосвязаны: все данные на одном уровне могут быть также найдены на более низком

уровне, и все данные на этом более низком уровне могут быть найдены на следующем ниже лежащем уровне и т. д.

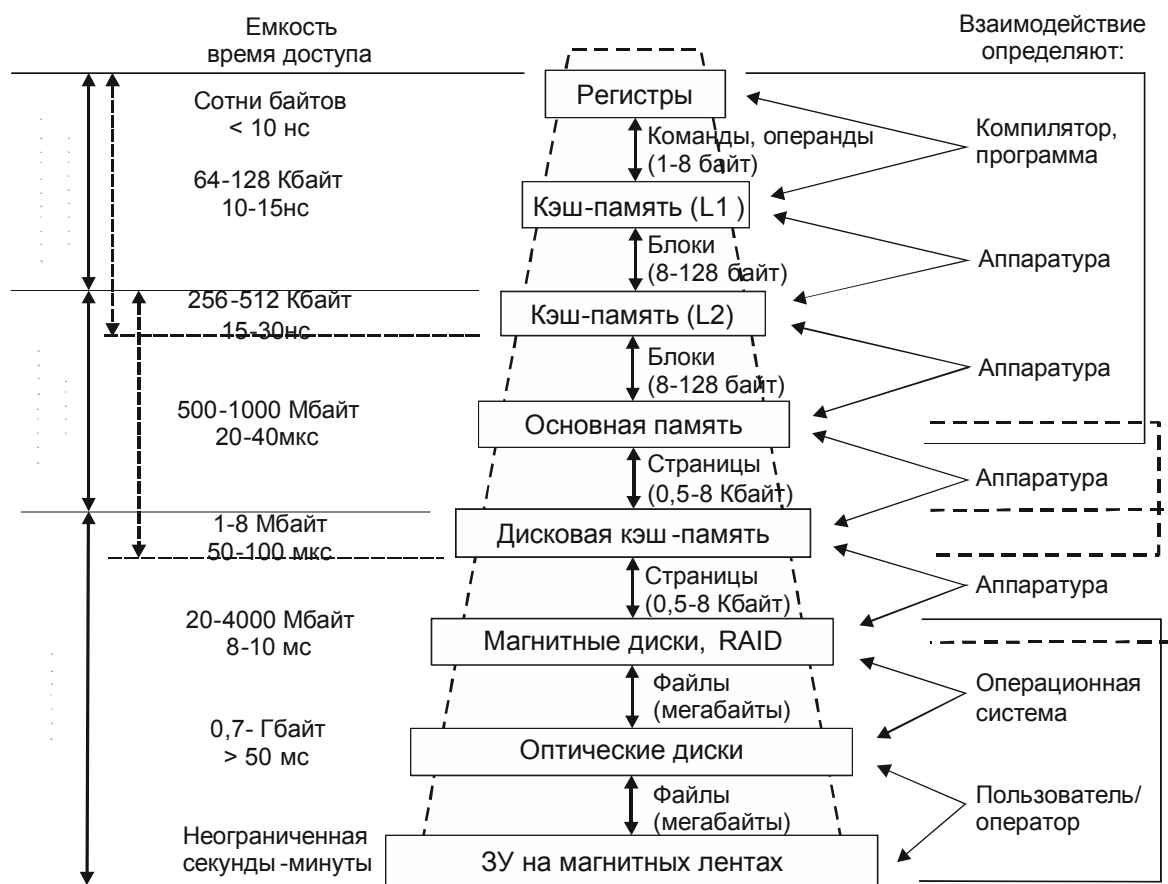


Рис. 1. Иерархия организации памяти.

Четыре верхних уровня иерархии образуют *внутреннюю память* ЭВМ, а все нижние уровни - это *внешняя* или *вторичная память*. По мере движения вниз по иерархической структуре:

- Уменьшается соотношение «стоимость/бит».
- Возрастает емкость.
- Растет время доступа.
- Уменьшается частота обращения к памяти со стороны центрального процессора

Если память организована в соответствии с пунктами 1-3, а характер размещения в ней данных и команд удовлетворяет пункту 4, иерархическая организация ведет к уменьшению общей стоимости при заданном уровне производительности.

Справедливость этого утверждения вытекает из принципа *локальности по обращению*. Если рассмотреть процесс выполнения большинства программ, то можно заметить, что с очень высокой вероятностью адрес очередной команды программы, либо следует непосредственно за адресом, по которому была считана текущая команда, либо расположен вблизи него. Такое расположение адресов называется *пространственной локальностью*.

стью программы. Обрабатываемые данные, как правило, структурированы, и такие структуры обычно хранятся в последовательных ячейках памяти. Данная особенность программ называется **пространственной локальностью данных**. Кроме того, программы содержат множество небольших циклов и подпрограмм. Это означает, что небольшие наборы команд могут многократно повторяться в течение некоторого интервала времени, то есть имеет место **временная локальность**. Все три вида локальности объединяет понятие **локальность по обращению**. Принцип локальности часто облекают в численную форму и представляют в виде так называемого правила «90/10»: 90% времени работы программы связано с доступом к 10% адресного пространства этой программы.

Из свойства локальности вытекает, что программу разумно представить в виде последовательно обрабатываемых фрагментов - компактных групп команд и данных. Помещая такие фрагменты в более быструю память, можно существенно снизить общие задержки на обращение, поскольку команды и данные, будучи один раз переданы из медленного ЗУ в быстрое, затем могут использоваться многократно, и среднее время доступа к ним в этом случае определяется уже более быстрым ЗУ. Это позволяет хранить большие программы и массивы данных на медленных, емких, но дешевых ЗУ, а в процессе обработки активно использовать сравнительно небольшую быструю память, увеличение емкости которой сопряжено с высокими затратами.

На каждом уровне иерархии информация разбивается на **блоки**, выступающие в качестве наименьшей информационной единицы, пересылаемой между двумя соседними уровнями иерархии. Размер блоков может быть фиксированным либо переменным. При фиксированном размере блока емкость памяти обычно кратна его размеру. Размер блоков на каждом уровне иерархии чаще всего различен и увеличивается от верхних уровней к нижним.

При доступе к командам и данным, например, для их считывания, сначала производится поиск в памяти верхнего уровня. Факт обнаружения нужной информации называют **попаданием** (hit), в противном случае говорят о **промахе** (miss). При промахе производится поиск в ЗУ следующего более низкого уровня, где также возможны попадание или промах. После обнаружении необходимой информации выполняется последовательная пересылка блока, содержащего искомую информацию, с нижних уровней на верхние. Следует отметить, что независимо от числа уровней иерархии пересылка информации может осуществляться только между двумя соседними уровнями.

При оценке эффективности подобной организации памяти обычно используют следующие характеристики:

- *коэффициент попаданий (hit rate)* - отношение числа обращений к памяти, при которых произошло попадание, к общему числу обращений к ЗУ данного уровня иерархии;
- *коэффициент промахов (miss rate)* - отношение числа обращений к памяти, при которых имел место промах; к общему числу обращений к ЗУ данного уровня иерархии;
- *время обращения при попадании (hit time)* - время, необходимое для поиска нужной информации в памяти верхнего уровня (включая выяснение, является ли обращение попаданием), плюс время на фактическое считывание данных;
- *потери на промах (miss penalty)* — время, требуемое для замены блока в памяти более высокого уровня на блок с нужными данными, расположенный в ЗУ следующего (более низкого) уровня.

Потери на промах включают в себя: *время доступа (access time)* - время обращения к первому слову блока при промахе и *время пересылки (transfer time)* — дополнительное время для пересылки оставшихся слов блока. Время доступа обусловлено задержкой памяти более низкого уровня, в то время как время пересылки связано с полосой пропускания канала между ЗУ двух смежных уровней.

Описание некоторого уровня иерархии ЗУ предполагает конкретизацию четырех моментов:

- *размещения блока* — допустимого места расположения блока на примыкающем сверху уровне иерархии;
- *идентификации блока* - способа нахождения блока на примыкающем сверху уровне;
- *замещения блока* - выбора блока, заменяемого при промахе с целью освобождения места для нового блока;
- *согласования копий (стратегии записи)* - обеспечения согласованности копий одних и тех же блоков, расположенных на разных уровнях, при записи новой информации в копию, находящуюся на более высоком уровне.

Самый быстрый, но и минимальный по емкости тип памяти - это внутренние регистры ЦП, которые иногда объединяют понятием *сверхоперативное запоминающее устройство* - СОЗУ. Как правило, количество регистров невелико, хотя в архитектурах с сокращенным набором команд их число может достигать до нескольких сотен. Основная память (ОП), значительно большей емкости, располагается несколькими уровнями ниже. Между регистрами ЦП и основной памятью часто размещают кэш-память, которая по емкости ощутимо проигрывает ОП, но существенно превосходит последнюю по быстродействию, уступая в то же время СОЗУ. В большин-

стве современных ЭВМ имеется несколько уровней кэш-памяти, которые обозначают буквой L и номером уровня кэш-памяти. На рис. 1 показаны два таких уровня. В последних разработках все чаще встречаются также третий уровень кэш-памяти (L3), причем разработчики ЭВМ говорят о целесообразности введения и четвертого уровня — L4. Каждый последующий уровень кэш-памяти имеет большую емкость, но одновременно и меньшее быстродействие по сравнению с предыдущим. Как бы то ни было, по «скорости» любой уровень кэш-памяти превосходит основную память. Все виды внутренней памяти реализуются на основе полупроводниковых технологий и в основном, являются энергозависимыми.

Долговременное хранение больших объемов информации (программ и данных) обеспечивается внешними ЗУ, среди которых наиболее распространены запоминающие устройства на базе магнитных и оптических дисков, а также магнитоленточные ЗУ.

Наконец, еще один уровень иерархии может быть добавлен между основной памятью и дисками. Этот уровень носит название дисковой кэш-памяти и реализуется в виде самостоятельного ЗУ, включаемого в состав магнитного диска. Дисковая кэш-память существенно улучшает производительность при обмене информацией между дисками и основной памятью.

Иерархия может быть дополнена и другими видами памяти. Так, некоторые модели ЭВМ фирмы IBM включают в себя так называемую расширенную память (expanded storage), выполненную на основе полупроводниковой технологии, но имеющую меньшее быстродействие и стоимость по сравнению с ОП. Строго говоря, этот вид памяти не входит в иерархию, а представляет собой ответвление от нее, поскольку данные могут передаваться только между расширенной и основной памятью, но не допускается обмен между расширенной и внешней памятью.

Кэш-память

В качестве элементной базы основной памяти в большинстве ЭВМ служат микросхемы динамических ОЗУ, на порядок уступающие по быстродействию центральному процессору. В результате процессор вынужден простаивать несколько тактовых периодов, пока информация из ОЗУ памяти установится на шине данных ЭВМ. Если ОП выполнить на быстрых микросхемах статической памяти, стоимость ЭВМ возрастет весьма существенно. Экономически приемлемое решение этой проблемы было предложено М. Уилксом в 1965 году в процессе разработки ЭВМ Atlas и заключается оно в использовании двухуровневой памяти, когда между ОП и процессором размещается небольшая, но быстродействующая буферная память. В процессе работы такой системы в буферную память копируются те участки ОП, к которым производится обращение со стороны процессора. В общепринятой терминологии - производится отображение участков

ОП на буферную память. Выигрыш достигается за счет ранее рассмотренного свойства локальности - если отобразить участок ОП в более быстродействующую буферную память и переадресовать на нее все обращения в пределах скопированного участка, можно добиться существенного повышения производительности ЭВМ.

Уилкс называл рассматриваемую буферную память *подчиненной* (slave memory). Позже распространение получил термин *кэш-память* (от английского слова cache - убежище, тайник), поскольку такая память обычно скрыта от программиста в том смысле, что он не может ее адресовать, и может даже вообще не знать о ее существовании. Впервые кэш-системы появились в машинах модели 85 семейства IBM 360.

В общем виде использование кэш-памяти поясним следующим образом. Когда ЦП пытается прочитать слово из основной памяти, сначала осуществляется поиск копии этого слова в кэше. Если такая копия существует, обращение к ОП не производится, а в ЦП передается слово, извлеченное из кэш-памяти. Данную ситуацию принято называть успешным обращением или *попаданием* (hit). При отсутствии слова в кэше, то есть при неуспешном обращении — *промахе* (miss), — требуемое слово передается в ЦП из основной памяти, но одновременно из ОП в кэш-память пересылается блок данных, содержащий это слово.

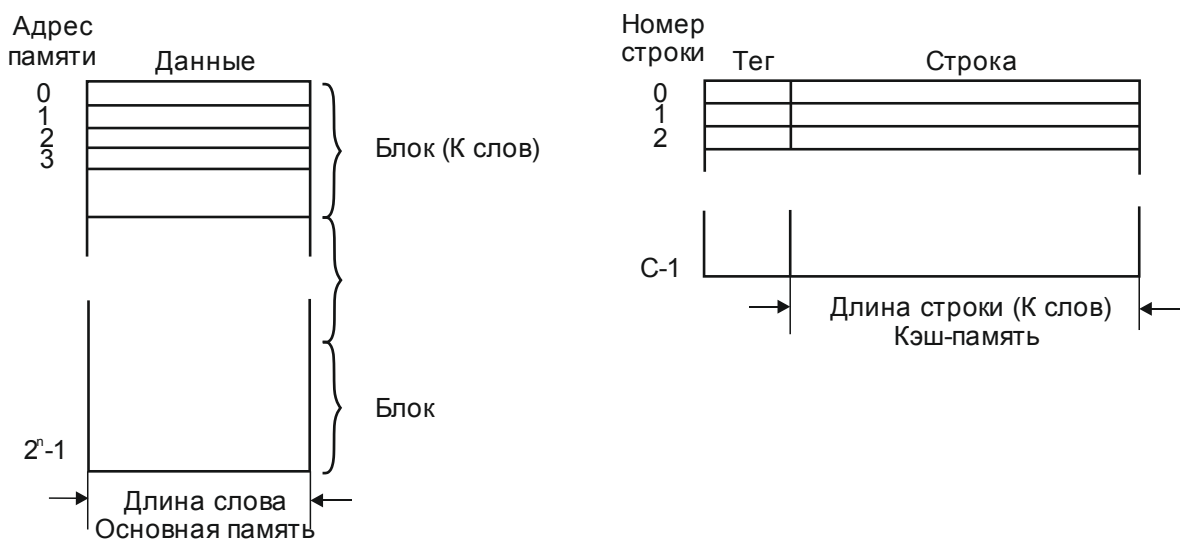


Рис. 2. Структура системы с основной и кэш-памятью.

На рис. 2 приведена структура системы с основной и кэш-памятью. ОП состоит из 2^n адресуемых слов, где каждое слово имеет уникальный n -разрядный адрес. При взаимодействии с кэшем эта память рассматривается как M блоков фиксированной длины по K слов в каждом ($M = 2^n/K$). Кэш-память состоит из C блоков аналогичного размера (блоки в кэш-памяти принято называть *строками*), причем их число значительно меньше числа блоков в основной памяти ($C \ll M$). При считывании слова из какого-либо блока ОП этот блок копируется в одну из строк кэша. Поскольку число

блоков ОП больше числа строк, отдельная строка не может быть выделена постоянно одному и тому же блоку ОП. По этой причине каждой строке кэш-памяти соответствует тег (признак), содержащий сведения о том, копия какого блока ОП в данный момент хранится в данной строке. В качестве тега обычно используется часть адреса ОП.

На эффективность применения кэш-памяти в иерархической системе памяти влияет целый ряд моментов. К наиболее существенным из них можно отнести:

- емкость кэш-памяти;
- размер строки;
- способ отображения основной памяти на кэш-память;
- алгоритм замещения информации в заполненной кэш-памяти;
- алгоритм согласования содержимого основной и кэш-памяти;
- число уровней кэш-памяти.

Емкость кэш-памяти

Выбор емкости кэш-памяти - это всегда определенный компромисс. С одной стороны, кэш-память должна быть достаточно мала, чтобы ее стоимостные показатели были близки к величине, характерной для ОП. С другой - она должна быть достаточно большой, чтобы среднее время доступа в системе, состоящей из основной и кэш-памяти, определялось временем доступа к кэш-памяти. В пользу уменьшения размера кэш-памяти имеется больше мотивировок. Так, чем вместительнее кэш-память, тем больше логических схем должно участвовать в ее адресации. Как следствие, интегральные микросхемы кэш-памяти повышенной емкости работают медленнее по сравнению с микросхемами меньшей емкости, даже если они выполнены по одной и той же технологии.

Реальная эффективность использования кэш-памяти зависит от характера решаемых задач, и невозможно заранее определить, какая ее емкость будет действительно оптимальной. Установлено, что для большинства задач близкой к оптимальной является кэш-память емкостью от 1 до 512 Кбайт.

Одноуровневая и многоуровневая кэш-память

Современные технологии позволяют разместить кэш-память и ЦП на общем кристалле. Такая внутренняя кэш-память строится по технологии статического ОЗУ и является наиболее быстродействующей. Емкость ее обычно не превышает 64 Кбайт. Попытки увеличения емкости обычно приводят к снижению быстродействия, главным образом из-за усложнения схем управления и дешифрации адреса. Общую емкость кэш-памяти ЭВМ увеличивают за счет второй (внешней) кэш-памяти, расположенной между внутренней кэш-памятью и ОП. Такая система известна под названием двухуровневой, где внутренней кэш-памяти отводится роль первого уровня

(L1), а внешней - второго уровня (L2). Емкость L2 обычно на порядок больше, чем у L1, а быстродействие и стоимость - несколько ниже. Память второго уровня также строится как статическое ОЗУ. Типичная емкость кэш-памяти второго уровня — 256 и 512 Кбайт, реже — 1 Мбайт, а реализуется она, как правило, в виде отдельной микросхемы, хотя в последнее время L2 часто размещают на одном кристалле с процессором, за счет чего сокращается длина связей и повышается быстродействие.

При доступе к памяти ЦП сначала обращается к кэш-памяти первого уровня. В случае промаха производится обращение к кэш-памяти второго уровня. Если информация отсутствует и в L2, выполняется обращение к ОП и соответствующий блок заносится сначала в L2, а затем и в L1. Благодаря такой процедуре часто запрашиваемая информация может быть быстро восстановлена из кэш-памяти второго уровня.

Среднее время доступа к одноуровневой кэш-памяти можно оценить как:

$$T = T_{L1h} + K_{L1m} \cdot T_{L1m}$$

где T_{L1h} - время обращения при попадании; K_{L1m} - коэффициент промахов; T_{L1m} - потери на промах. Для двухуровневой кэш-памяти имеем:

$$T = T_{L1h} + K_{L1m} \cdot T_{L1m} + K_{L2m} \cdot T_{L2m}$$

Потенциальная экономия за счет применения L2 зависит от вероятности попаданий как в L1, так и в L2. Ряд исследований показывает, что использование кэш-памяти второго уровня существенно улучшает производительность.

Количество уровней кэш-памяти не ограничивается двумя. В большинстве современных ЭВМ уже есть кэш-память третьего уровня (L3) и ведутся активные дискуссии о введении также и кэш-памяти четвертого уровня (L4). Характер взаимодействия очередного уровня с предшествующим аналогичен описанному для L1 и L2. Таким образом, можно говорить об иерархии кэш-памяти. Каждый последующий уровень характеризуется большей емкостью, меньшей стоимостью, но и меньшим быстродействием, хотя оно все же выше, чем у ЗУ основной памяти.

Основная память

Основная память (ОП) представляет собой единственный вид памяти, к которой ЦП может обращаться непосредственно (исключение составляют лишь регистры центрального процессора). Информация, хранящаяся на внешних ЗУ, становится доступной процессору только после того, как будет переписана в основную память.

Основную память образуют запоминающие устройства с произвольным доступом. Такие ЗУ образованы как массив ячеек, а «произвольный доступ» означает, что обращение к любой ячейке занимает одно и то же

время и может производиться в произвольной последовательности. Каждая ячейка содержит фиксированное число запоминающих элементов и имеет уникальный адрес, позволяющий различать ячейки при обращении к ним для выполнения операций записи и считывания.

Следствием огромных успехов в области полупроводниковых технологий стало изменение элементной базы основной памяти. На смену ЗУ на базе ферромагнитных колец пришли полупроводниковые микросхемы, использование которых в наши дни стало повсеместным.

Основная память может включать в себя два типа устройств: оперативные запоминающие устройства (ОЗУ) и постоянные запоминающие устройства (ПЗУ)

Преимущественную долю основной памяти образует ОЗУ, называемое оперативным, потому что оно допускает как запись, так и считывание информации, причем обе операции выполняются однотипно, практически с одной и той же скоростью, и производятся с помощью электрических сигналов. В англоязычной литературе ОЗУ соответствует аббревиатура RAM — *Random Access Memory*, то есть «память с произвольным доступом», что не совсем корректно, поскольку памятью с произвольным доступом являются также ПЗУ и регистры процессора. Для большинства типов полупроводниковых ОЗУ характерна энергозависимость — даже при кратковременном прерывании питания хранимая информация теряется. Микросхема ОЗУ должна быть постоянно подключена к источнику питания и поэтому может использоваться только как временная память.

Вторую группу полупроводниковых ЗУ основной памяти образуют энергонезависимые микросхемы ПЗУ (ROM - *Read-Only Memory*). ПЗУ обеспечивает считывание информации, но не допускает ее изменения (в ряде случаев информация в ПЗУ может быть изменена, но этот процесс сильно отличается от считывания и требует значительно большего времени).

Понятие виртуальной памяти

Для большинства типичных применений ЭВМ характерна ситуация, когда размещение всей программы в ОП невозможно из-за ее большого размера. В этом, однако, и нет принципиальной необходимости, поскольку в каждый момент времени «внимание» машины концентрируется на определенных сравнительно небольших участках программы. Таким образом, в ОП достаточно хранить только используемые в данный период части программ, а остальные части могут располагаться на внешних ЗУ (ВЗУ). Сложность подобного подхода в том, что процессы обращения к ОП и ВЗУ существенно различаются, и это усложняет задачу программиста. Выходом из такой ситуации было появление в 1959 году идеи виртуализации памяти, под которой понимается метод автоматического управления иерархической памятью, при котором программисту кажется, что он имеет дело

с единой памятью большой емкости и высокого быстродействия. Эту память называют виртуальной (кажущейся) памятью. По своей сути виртуализация памяти представляет собой способ аппаратной и программной реализации концепции иерархической памяти.

В рамках идеи виртуализации памяти ОП рассматривается как линейное пространство N адресов, называемое *физическим пространством памяти*. Для задач, где требуется более чем N ячеек, предоставляется значительно большее пространство адресов (обычно равное общей емкости всех видов памяти), называемое виртуальным пространством, в общем случае не обязательно линейное. Адреса виртуального пространства называют *виртуальными*, а адреса физического пространства - *физическими*. Программа пишется в виртуальных адресах, но поскольку для её выполнения нужно, чтобы обрабатываемые команды и данные находились в ОП, требуется, чтобы каждому виртуальному адресу соответствовал физический адрес. Таким образом, в процессе вычислений необходимо, прежде всего, переписать из ВЗУ в ОП ту часть информации, на которую указывает виртуальный адрес (отобразить виртуальное пространство на физическое), после чего преобразовать виртуальный адрес в физический (рис. 3).

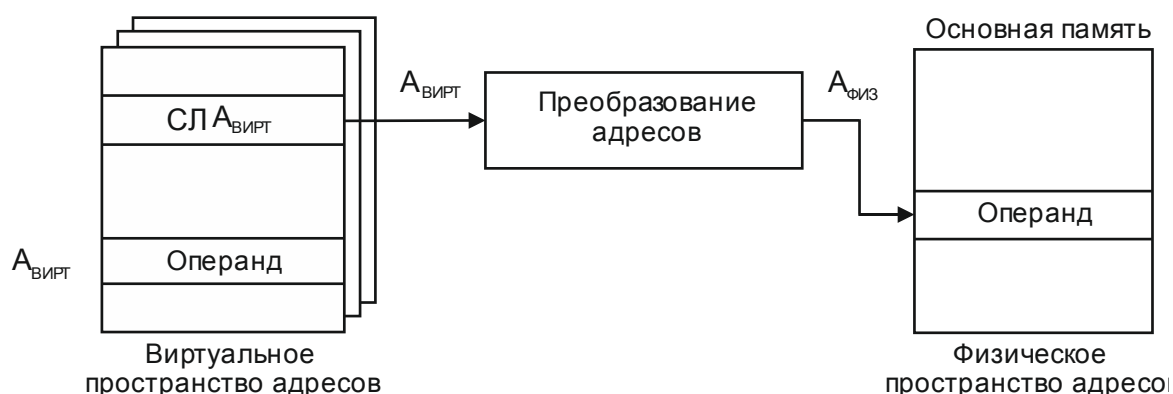


Рис. 3. Отображение виртуального адреса на физический.

Среди систем виртуальной памяти можно выделить два класса: системы с *фиксированным размером блоков* (страничная организация) и системы с *переменным размером блоков* (сегментная организация). Оба варианта обычно совмещают (сегментно-страничная организация).

Внешняя память

Важным звеном в иерархии запоминающих устройств является внешняя, или вторичная память, реализуемая на базе, различных ЗУ. Наиболее распространенные виды таких ЗУ — это магнитные и оптические диски и магнитоленточные устройства.

Магнитные диски

Информация в ЗУ на магнитных дисках (МД) хранится на плоских металлических или пластиковых пластинах (дисках), покрытых магнитным

материалом. Данные записываются и считываются с диска с помощью электромагнитной катушки» называемой *головкой считывания/записи*, которая в процессе считывания и записи неподвижна, в то время как диск вращается относительно нее. При записи на головку подаются электрические импульсы, намагничивающие участок поверхности под ней, причем характер намагниченности поверхности различен в зависимости от направления тока в катушке. Считывание базируется на электрическом токе, наводимом в катушке головки, под воздействием перемещающегося относительно нее магнитного поля. Когда под головкой проходит участок поверхности диска, в катушке наводится ток той же полярности, что использовался для записи информации. Несмотря на разнообразие типов магнитных дисков, принципы их организации обычно однотипны.

Организация данных и форматирование

Данные на диске организованы в виде набора концентрических окружностей, называемых дорожками. Каждая из них имеет ту же ширину, что и головка. Соседние дорожки разделены промежутками. Это предотвращает ошибки из-за смещения головки или из-за интерференции магнитных полей. Как правило, для упрощения электроники принимается, что на всех дорожках может храниться одинаковое количество информации. Таким образом, плотность записи увеличивается от внешних дорожек к внутренним.

Обмен информацией с МД осуществляется блоками. Размер блока обычно меньше емкости дорожки, и данные на дорожке хранятся в виде последовательных областей — секторов, разделенных между собой промежутками. Размер сектора равен минимальному размеру блока.

Типовое число секторов на дорожке колеблется от 10 до 100. При такой организации должны быть заданы точка отсчета секторов и способ определения начала и конца каждого сектора. Все это обеспечивается с помощью форматирования, в ходе которого на диск заносится служебная информация, недоступная пользователю и используемая только аппаратурой дискового ЗУ

Дисковая кэш-память

Концепция кэш-памяти применима и к дисковым ЗУ. Принцип кэширования дисков во многом схож с принципом кэширования основной памяти, хотя способы доступа к диску и ОП существенно разнятся. Если время обращения к любой ячейке ОП одинаково, то для диска оно зависит от целого ряда факторов. Во-первых, нужно затратить некоторое время для установки головки считывания/записи на нужную дорожку. Во-вторых, поскольку при движении головка вибрирует, необходимо подождать, чтобы она успокоилась. В-третьих, искомый сектор может оказаться под головкой также лишь спустя некоторое время.

Дисковая кэш-память представляет собой память с произвольным доступом, «размещенную» между дисками и ОП. Емкость такой памяти обычно достаточно велика - от 8 Мбайт и более. Пересылка информации между дисками и основной памятью организуется контроллером дисковой кэш-памяти. Изготавливается дисковая кэш-память на базе таких же полупроводниковых запоминающих устройств, что и основная память, поэтому в ряде случаев с ней обращаются как с дополнительной основной памятью. С другой стороны, в ряде операционных систем, таких как UNIX, в качестве дискового кэш используется область основной памяти.

В дисковой кэш-памяти хранятся блоки информации, которые с большой вероятностью будут востребованы в ближайшем будущем. Принцип локальности, обеспечивающий эффективность обычной кэш-памяти, справедлив и для дисковой, приводя к сокращению времени ввода/вывода данных от величин 20-30 мс до значений порядка 2-5 мс, в зависимости от объема передаваемой информации.

В качестве единицы пересылки может выступать сектор, несколько секторов, а также одна или несколько дорожек диска. Кроме того, иногда применяется пересылка информации, начиная с выбранного сектора на дорожке до ее конца. В случае пересылки секторов кэш-память заполняется не только требуемым сектором, но секторами, непосредственно следующими за ним, так как известно, что в большинстве случаев взаимосвязанные данные хранятся в соседних секторах. Этот метод известен также как *опережающее чтение* (read ahead).

В дисковых кэш обычно используется алгоритм сквозной записи. Специфика состоит в том, что далеко не всю информацию, перемещаемую между дисками и основной памятью, выгодно помещать в дисковый кэш. В ряде случаев определенные данные и команды целесообразно пересылать напрямую между ОП и диском. По этой причине в системах с дисковым кэшем предусматривают специальный динамический механизм, позволяющий переключать тракт пересылки информации: через кэш или минуя его.

Одна из привлекательных сторон дискового кэш в том, что связанные с ним преимущества могут быть получены без изменений в имеющемся аппаратном и программном обеспечении. Многие серийно выпускаемые дисковые кэш интегрированы в состав дисковых ЗУ. Примечательно, что архитектура кэш-памяти современных магнитных дисков типа «винчестер» реализует полностью ассоциативное отображение.