Учебная дисциплина

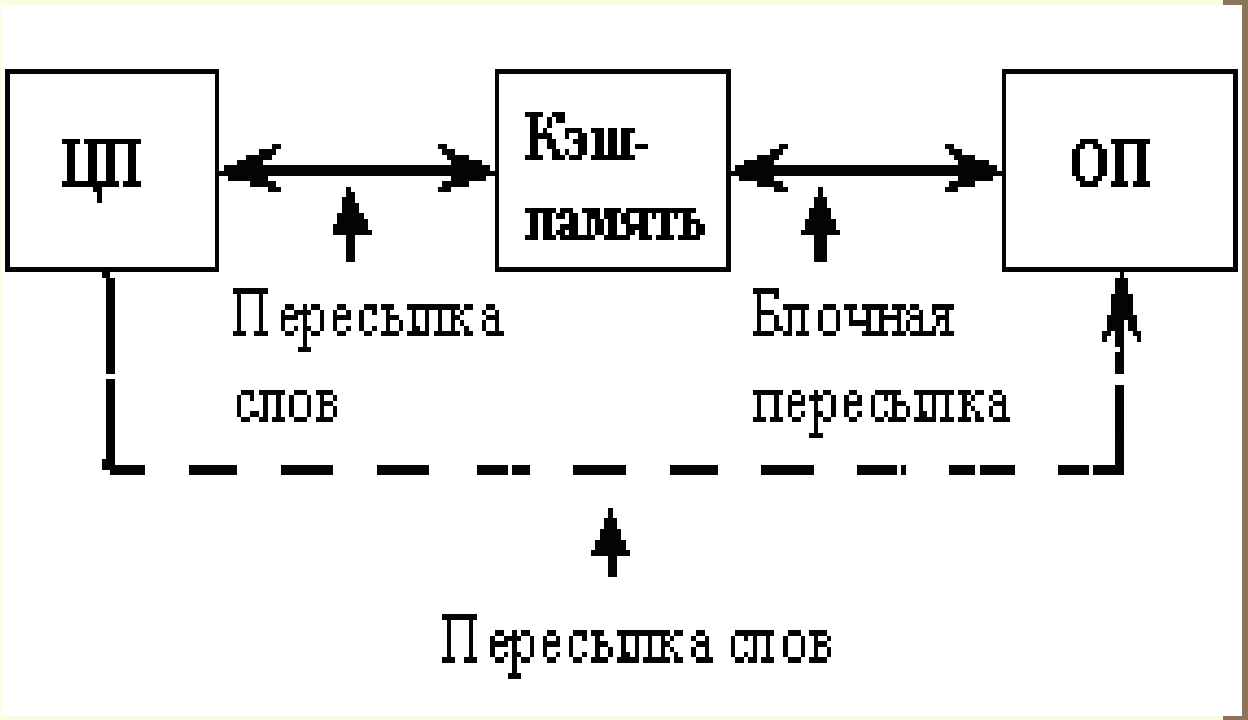
Вычислительные средства АСОИУ

(5 семестр)

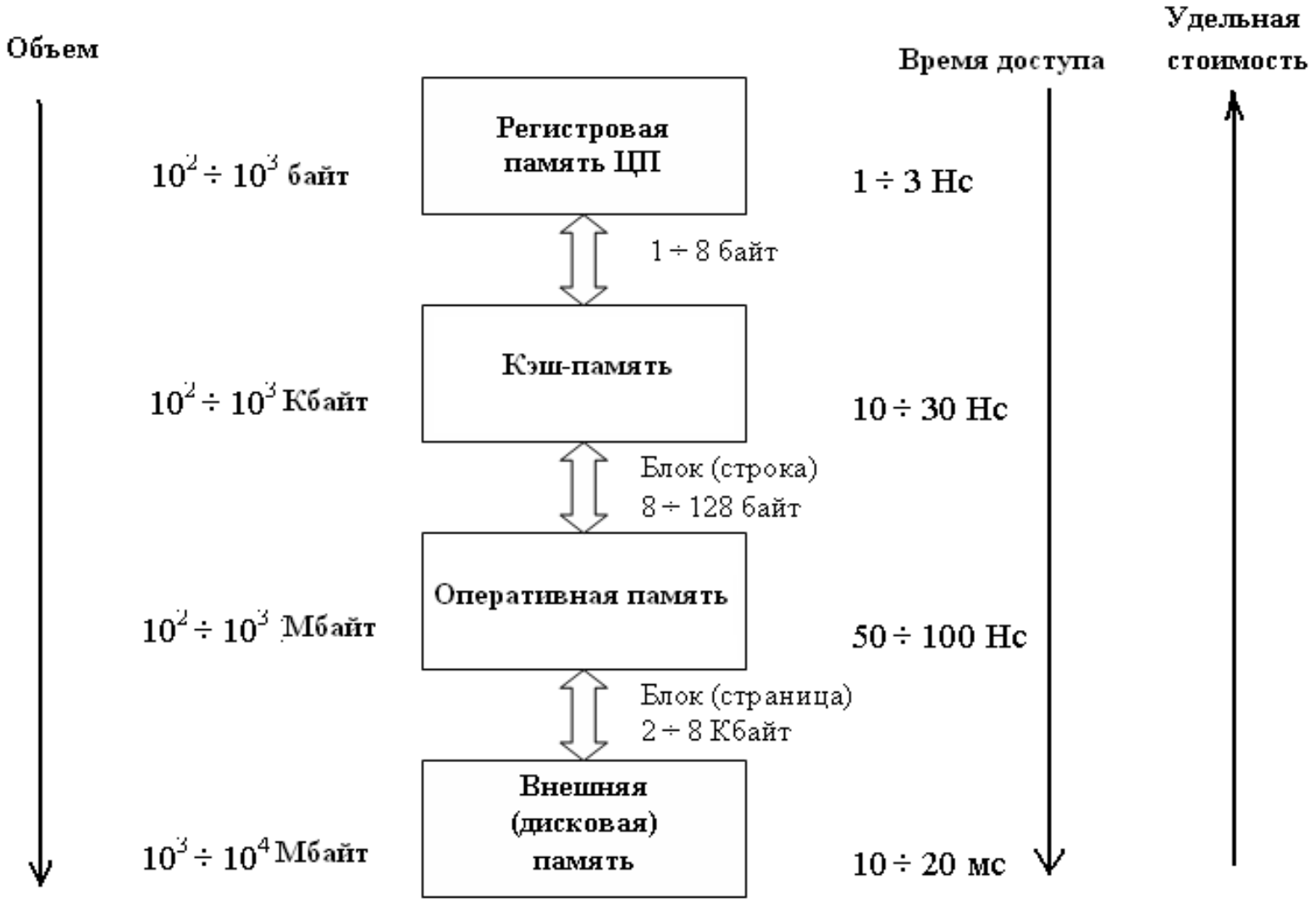
Часть5

Кэш- память

- **Принципы организации кэш-памяти**



Упрощённая схема иерархии памяти



Кэш-память

- **Кэш-память** (КП), или **кэш**, представляет собой организованную в виде ассоциативного *запоминающего устройства* (АЗУ) быстродействующую *буферную память* ограниченного объема, которая располагается между регистрами процессора и относительно медленной основной памятью.

Кэш-память

- Кэш-память хранит наиболее часто используемую информацию совместно с ее признаками (тегами), в качестве которых выступает часть адресного кода.

Структура кэш-памяти

- Кэш-память и основная память разделяются на блоки одинакового объема, размер которых обычно составляет 8-128 байт. Обычно блок ОП и кэш-памяти называют *строкой (line)*. Обмен между ОП и кэш-памятью носит блочный характер.
- В кэш-памяти содержатся копии тех блоков ОП, к которым в последнее время выполнялись обращения со стороны ЦП.

- В процессе работы отдельные блоки информации копируются из основной памяти в *кэш-память*. При обращении процессора за командой или данными сначала проверяется их наличие в КП. Если необходимая *информация* находится в кэше, она быстро извлекается. Это **кэш-попадание**. Если необходимая *информация* в КП отсутствует (**кэш-промах**), то она выбирается из основной памяти, передается в *микропроцессор* и одновременно заносится в *кэш-память*.

Механизм сохранения информации в кэш-памяти

- При включении микропроцессора в работу вся информация в его кэш-памяти недостоверна.
- При обращении к памяти микропроцессор, как уже отмечалось, сначала проверяет, не содержится ли искомая информация в кэш-памяти.
- Для этого сформированный им физический адрес сравнивается с адресами ячеек памяти, которые были ранее кэшированы из ОЗУ в КП..

Механизм сохранения информации в кэш-памяти

- Для того чтобы уже следующее обращение к КП приводило как можно чаще к **кэш-попаданиям**, передача из оперативной памяти в *кэш-память* происходит не теми порциями (байтами или словами), которые востребованы микропроцессором в данном обращении, а так называемыми **строками**.

Механизм сохранения информации в кэш-памяти

- *Длина строки* превышает максимально возможную длину востребованных микропроцессором данных. Обычно она составляет от 16 до 64 *байт* и выровнена в памяти по границе соответствующего раздела

Механизм сохранения информации в кэш-памяти

Высокий *процент* **кэш-попаданий** в этом случае обеспечивается благодаря тому, что в большинстве случаев программы обращаются к ячейкам памяти, расположенным вблизи от ранее использованных. Это свойство, называемое ***принципом локальности ссылок***, обеспечивает эффективность использования КП.

Механизм сохранения информации в кэш-памяти

- Например, микропроцессору для своей работы потребовалось 2 байта информации. Если строка имеет длину 16 *байт*, то в *кэш* переписываются не только нужные 2 байта, но и некоторое их окружение.
  Когда *микропроцессор* обращается за новой информацией, в силу локальности ссылок, скорее всего, обращение произойдет по соседнему адресу.

Механизм сохранения информации в кэш-памяти

- Когда очередной сформированный микропроцессором *физический адрес* выйдет за пределы строки *кэш*-памяти (произойдет **кэш-промах**), будет выполнена подкачка в *кэш* новой строки, и вновь ряд последующих обращений вызовет **кэш-попадания**.

Механизм сохранения информации в кэш-памяти

- Чем длиннее используемая при обмене между оперативной и кэш-памятью строка, тем больше *вероятность* того, что следующее обращение произойдет в пределах этой строки.

принцип локальности ссылок

- Причем выделяются два вида локальности: *пространственная* и *временная*. Кроме того, принцип локальности рассматривается как в отношении команд, так и в отношении данных.

принцип локальности ссылок

- *Пространственная локальность в отношении команд* характеризуется тем, что вероятность выборки команды по следующему адресу, по сравнению с адресом исполняемой команды, намного больше, чем вероятность выборки команды по любому другому адресу. Этот принцип проявляется на линейных участках программы.

принцип локальности ссылок

- *Пространственная локальность в отношении данных* выражается в том, что вероятность обращения к слову данных по следующему адресу намного больше вероятности обращения к данным по любому другому адресу.

принцип локальности ссылок

- *Временной аспект принципа локальности обращений в отношении команд* предполагает большую вероятность обращения к команде по одному и тому же адресу в течение небольшого интервала времени.

принцип локальности ссылок

- *В отношении данных временной аспект принципа локальности обращений* означает большое значение вероятности обращений к одному и тому же слову данных в течение небольшого интервала времени.

Оценка эффективности кэш-памяти

- Численной оценкой эффективности принятого принципа построения кэш-памяти является ***процент удачных обращений (процент кэш-попаданий),*** определяемый как отношение числа обращений к памяти, реализуемых через кэш, к общему числу обращений. Как правило, ***в современных компьютерах процент кэш-попаданий составляет 95÷98 %.***

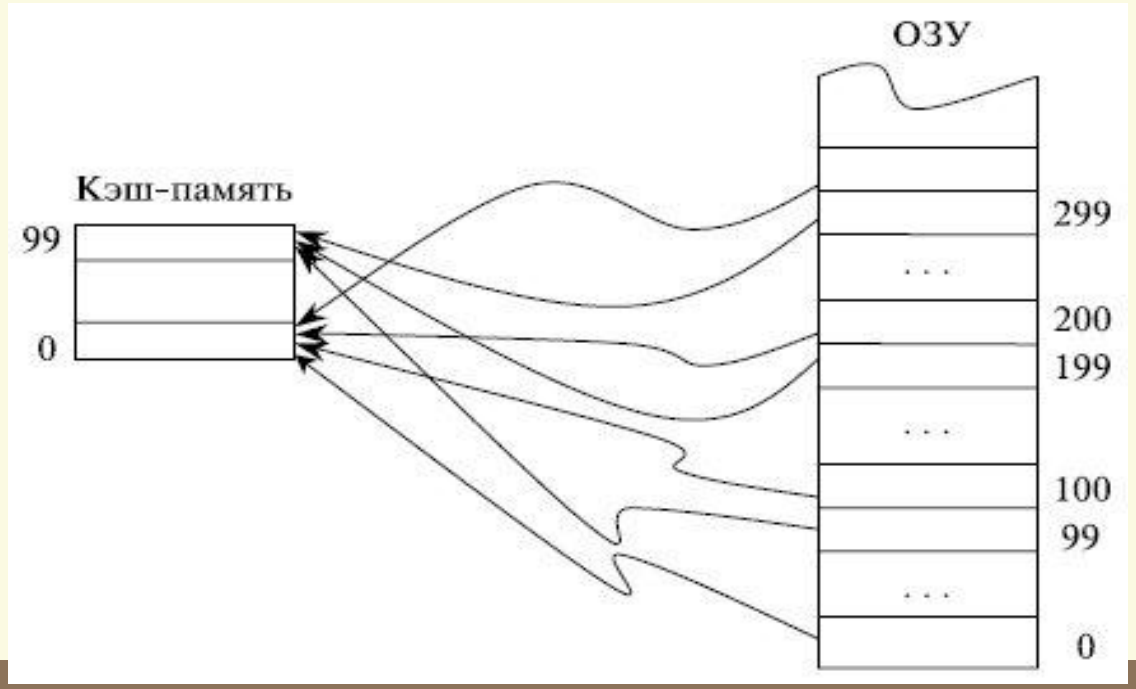
- Поэтому при выборе длины строки должен быть разумный *компромисс* между соотношением времени обращения к оперативной и *кэш*-памяти и вероятностью достаточно удаленного перехода от текущего адреса при выполнении программы. Обычно *длина строки* определяется в результате моделирования аппаратно-программной *структуры системы*.

Типы кэш-памяти

- 1. Кэш-память с прямым отображением.
- 2. Полностью ассоциативная кэш-память.
- 3. Множественно-ассоциативная кэш-память.

Кэш-память с прямым отображением.

- каждая строка *ОЗУ* имеет только одно фиксированное *место*, на котором она может находиться в *кэш-памяти*



Кэш-память с прямым отображением.

- Предположим, что *ОЗУ* состоит из 1000 строк с номерами от 0 до 999, а *кэш-память* имеет емкость только 100 строк. В *кэш*-памяти с прямым отображением строки *ОЗУ* с номерами 0, 100, 200, ..., 900 могут сохраняться только в строке 0 КП и нигде иначе, строки 1, 101, 201, …, 901
- *ОЗУ* - в строке 1 КП, строки *ОЗУ* с номерами 99, 199, …, 999 сохраняются в строке 99 *кэш*-памяти (.

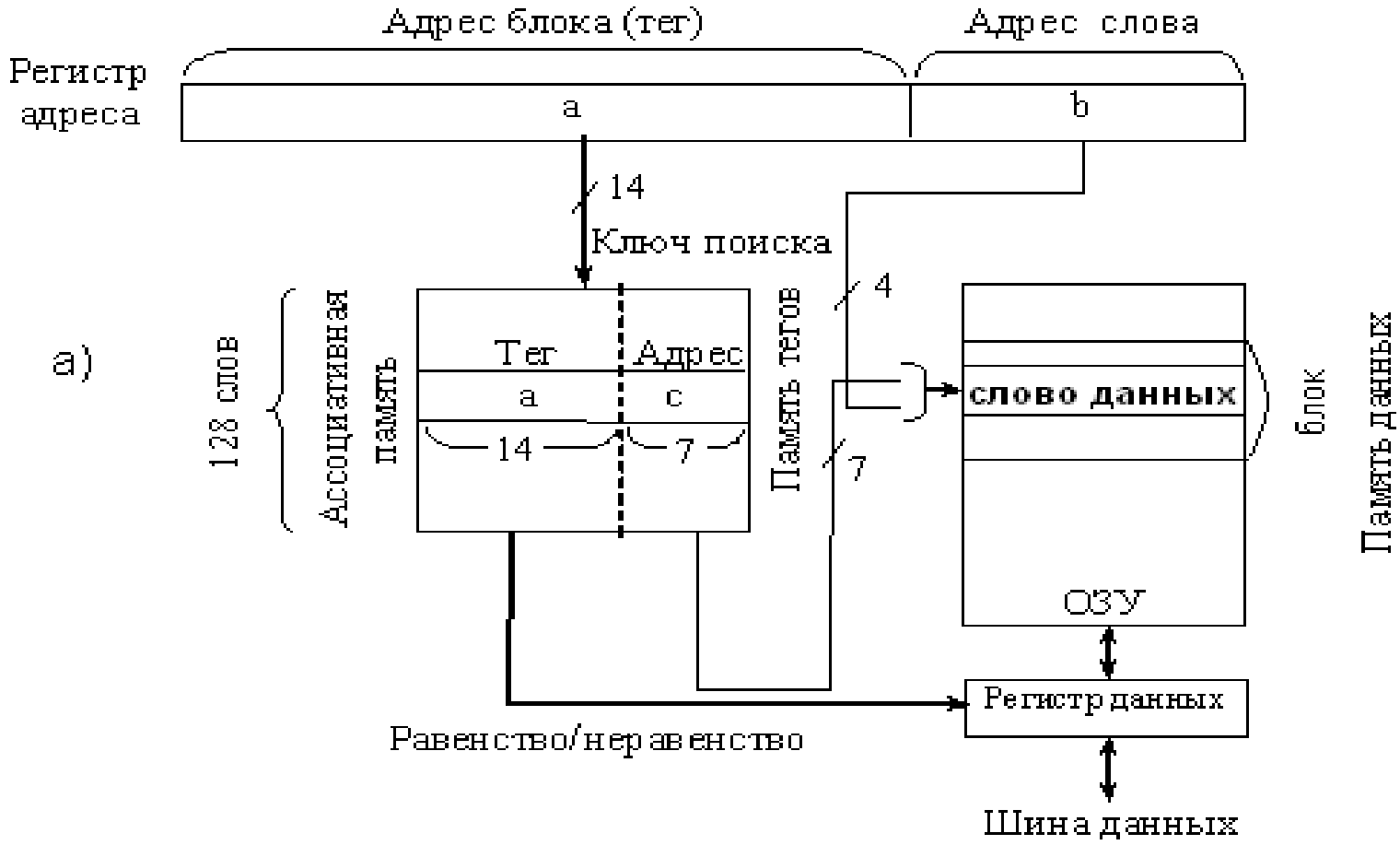
Кэш-память с прямым отображением.

- Кэш-память разделяется на две части: память тегов и память данных. Память данных состоит из 128 блоков (по разрядности поля *b* в адресе основной памяти), каждый блок содержит 16 слов (по разрядности поля *c*). В частном случае, слово может соответствовать байту. Емкость кэш-памяти составляет 2 Кслов.

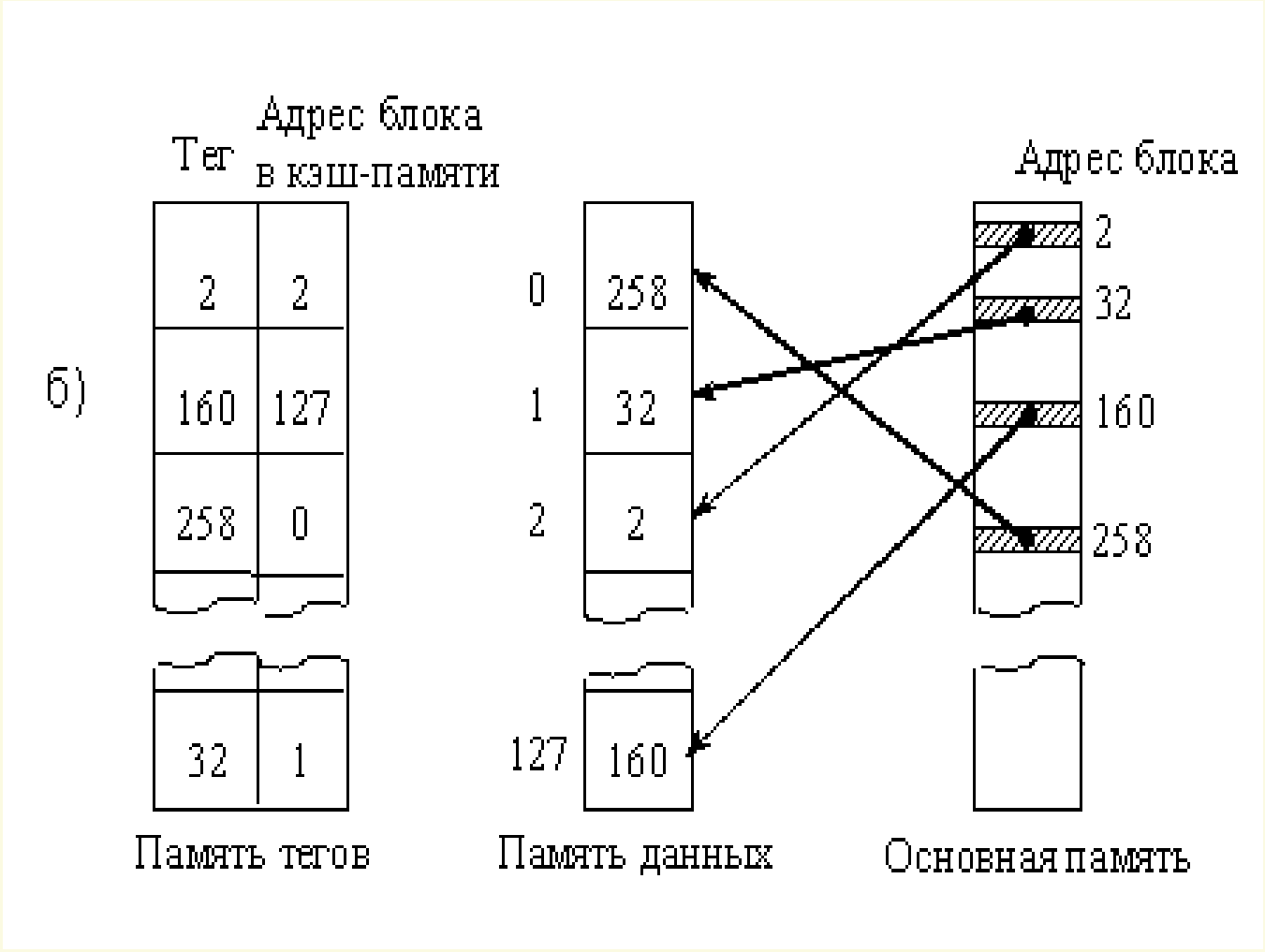
Полностью ассоциативная кэш-память

- Каждая строка *ОЗУ* может располагаться в любом месте такой *кэш*-памяти.
- В полностью ассоциативной *кэш*-памяти максимально используется весь ее объем: *вытеснение* сохраненной в КП информации проводится лишь после ее полного заполнения. Однако *поиск* в *кэш*-памяти, организованной подобным образом, представляет собой трудную задачу.

Полностью ассоциативная кэш-память



Полностью ассоциативная кэш-память



Полностью ассоциативная кэш-память

- Память тегов реализуется как ассоциативная, то есть обращение к ней производится не по адресу, а по некоторому ключу (признаку, тегу). Ключом поиска в памяти тегов является 14-разрядный адрес блока.

Полностью ассоциативная кэш-память

- В ассоциативной памяти выделяются два регистровых блока:
- - регистры тегов (TR1, …, TRn);
- - регистры данных (DR1, …, DRn).

Полностью ассоциативная кэш-память

- Между теговыми регистрами и регистрами данных существует взаимно однозначное соответствие. Для поиска информации в ассоциативной памяти во входной регистр ITR (Input Tag Register) записывается тег, по которому осуществляется поиск данных. Этот тег поступает на один из входов схем совпадений (компараторов) CMP1, …, CMPn. На другой вход каждого компаратора подается тег из соответствующего тегового регистра.

Полностью ассоциативная кэш-память

- При совпадении содержимого одного из теговых регистров с входным тегом на соответствующем компараторе будет выработан сигнал разрешения обращения E1, …, En к соответствующему регистру данных. Выходы всех компараторов объединяются на элементе ИЛИ, выход которого связан с входом разрешения обращения выходного регистра данных ODR

Полностью ассоциативная кэш-память

- При подаче сигнала чтения R в регистры данных и сигнала записи W в выходной регистр данных, на выходе ассоциативной памяти появятся данные с тегом, соответствующим входному.
- В случае несовпадения входного тега с содержимым теговых регистров вырабатывается соответствующий сигнал на выходе инвертора. Этим сигналом отмечается факт отсутствия искомой информации в ассоциативной памяти.

Множественно-ассоциативная кэш-память

- Компромиссом между этими двумя способами организации *кэш*-памяти служит **множественно-ассоциативная** КП, в которой каждая строка *ОЗУ* может находиться по ограниченному множеству мест в *кэш*-памяти.

Множественно-ассоциативная кэш-память

- В кэш-памяти с множественно-ассоциативным отображением осуществляется разделение блоков кэш-памяти и, соответственно, основной памяти на ряд множеств (групп). Для блоков ОП, принадлежащих одному множеству, реализуется принцип прямого отображения, то есть все блоки ОП из одного множества должны отображаться на определенное число блоков кэш-памяти соответствующего множества гораздо меньшей мощности.

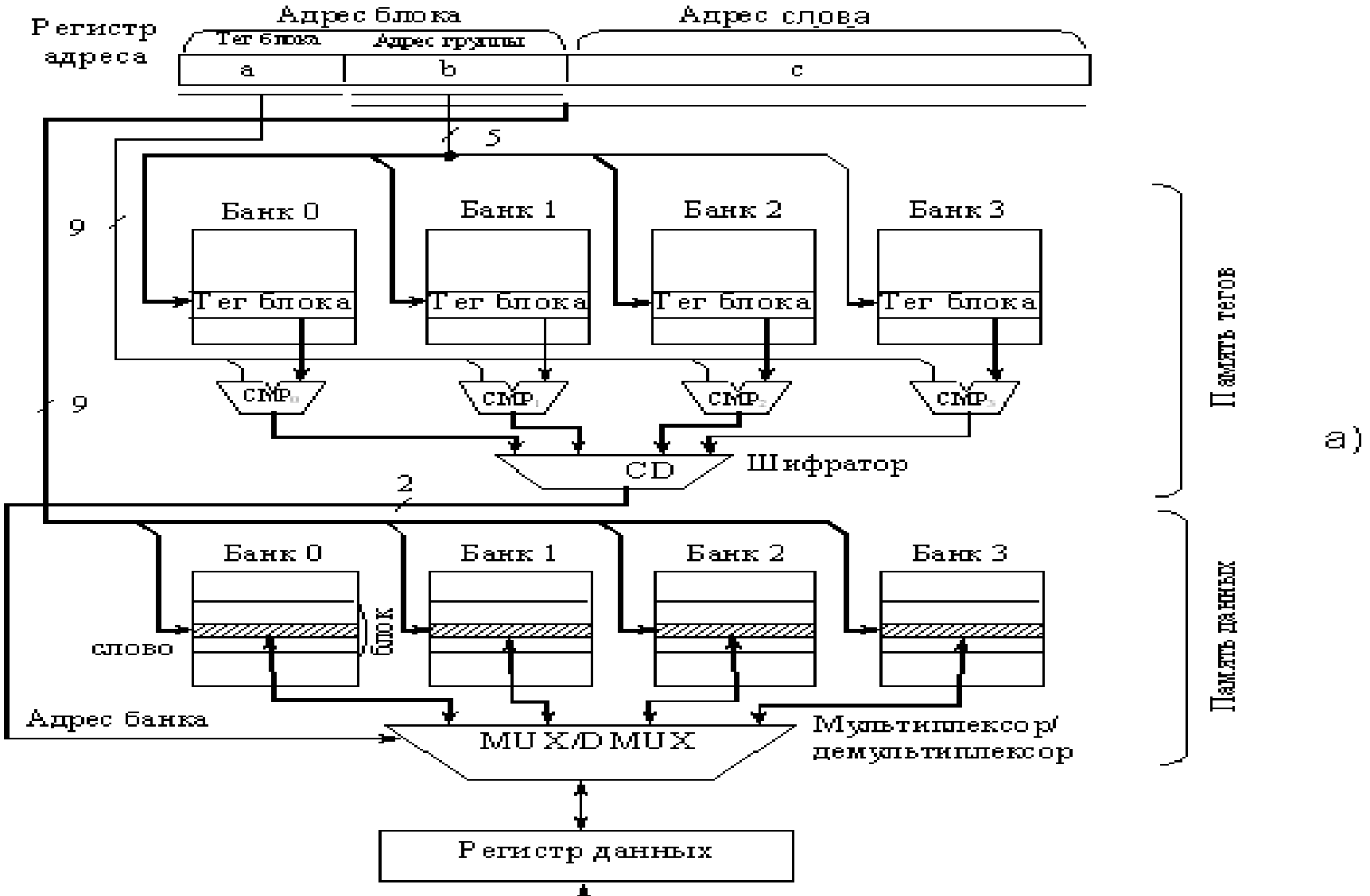
Множественно-ассоциативная кэш-память

- В свою очередь принцип отображения блоков ОП на блоки кэш-памяти внутри выбранного множества является ассоциативным: любой блок ОП, принадлежащий выбранному множеству, может помещаться на место любого блока кэш-памяти, принадлежащего соответствующему множеству.

Множественно-ассоциативная кэш-память

- Память данных состоит из 128 блоков, которые разделяются на 32 множества (группы) по 4 блока в множестве. Каждому блоку памяти данных соответствует определенный тег из памяти тегов. Память тегов представляет собой матрицу из 32 строк и 4 столбцов.

Множественно-ассоциативная кэш-память



Множественно-ассоциативная кэш-память

- В свою очередь, ОП состоит из 32 множеств (групп) по 512 блоков в множестве. Каждый из них претендует на занятие одного из 4-х мест в кэш-памяти. Подобная реализация называется кэш-памятью с четырехканальным доступом.

Множественно-ассоциативная кэш-память

- При обращении к кэш-памяти из памяти тегов выбираются 4 тега блоков из одного множества (по одному из каждого банка) для параллельного сравнения с входным тегом из регистра адреса на компараторах CMP0 – CMP3.

Множественно-ассоциативная кэш-память

- При совпадении одного из тегов блоков с входным тегом на выходе шифратора (кодера) формируется адрес банка для управления пересылкой данных между регистром данных и памятью данных через мультиплексор/демультиплексор.
- Банк соответствует столбцу памяти тегов и памяти данных. Каждый банк снабжается собственными схемами управления доступом, что допускает параллельное обращение ко всем банкам по одному адресу.

Стратегии замещения информации в кэш-памяти

- *1. LRU* - замещается строка, к которой дольше всего не было обращений;
- 2. FIFO - замещается самая давняя по пребыванию в кэш-памяти строка;
- 3. Random - замещение проходит случайным образом.

Влияние стратегий замещения

- При увеличении емкости кэша, естественно, уменьшается вероятность **кэш-промаха**, но даже при незначительной на сегодняшний день емкости кэш-памяти в 16 Кбайт около 95 % обращений происходят к КП, минуя оперативную память

Влияние стратегий замещения

- Чем больше степень ассоциативности кэш-памяти, тем больше вероятность **кэш-попадания** за счет более полного заполнения КП (время поиска информации в КП в данном анализе не учитывается);

Влияние стратегий замещения

- Механизм *LRU* обеспечивает более высокую вероятность **кэш-попадания** по сравнению с механизмом случайного замещения Random, однако этот выигрыш не очень значителен.

Влияние стратегий замещения

- Соответствие между данными в оперативной памяти и в *кэш*-памяти обеспечивается внесением изменений в те области *ОЗУ*, для которых данные в *кэш*-памяти подверглись изменениям. Существует два основных способа реализации этих действий: со сквозной записью (writethrough) и с обратной записью (write-back).

- ***Обновление основной памяти:***
- 1. Кэширование со сквозной записью.
- 2. Кэширование с обратной записью.

Кэширование со сквозной записью.

- При записи **кэширование со сквозной записью** обновляет *основную память* параллельно с обновлением информации в КП. Это несколько снижает *быстродействие* системы, так как *микропроцессор* впоследствии может вновь обратиться по этому же адресу для записи информации, и предыдущая пересылка строки *кэш*-памяти в *ОЗУ* окажется бесполезной. Однако при таком подходе содержимое соответствующих друг другу строк *ОЗУ* и КП всегда идентично. Это играет большую роль в мультипроцессорных *системах с общей* оперативной памятью.

Кэширование с обратной записью

- **Кэширование с обратной записью** модифицирует строку *ОЗУ* лишь при *вытеснении* строки *кэш*-памяти, например, в случае необходимости освобождения места для записи новой строки из *ОЗУ* в уже заполненную КП.

Кэширование с обратной записью

- Промежуточное положение между этими подходами занимает способ, при котором все строки, предназначенные для передачи из КП в *ОЗУ*, предварительно накапливаются в некотором буфере.
- Передача осуществляется либо при *вытеснении* строки, как в случае **кэширования с обратной записью**, либо при необходимости согласования *кэш*-памяти нескольких микропроцессоров в мультипроцессорной системе, либо при заполнении буфера.

Организация внутренней кэш-памяти микропроцессора

- Внутренний *кэш* 32-разрядного универсального микропроцессора является общим при обращении как к командам, так и к данным. Обращение ведется по физическим адресам

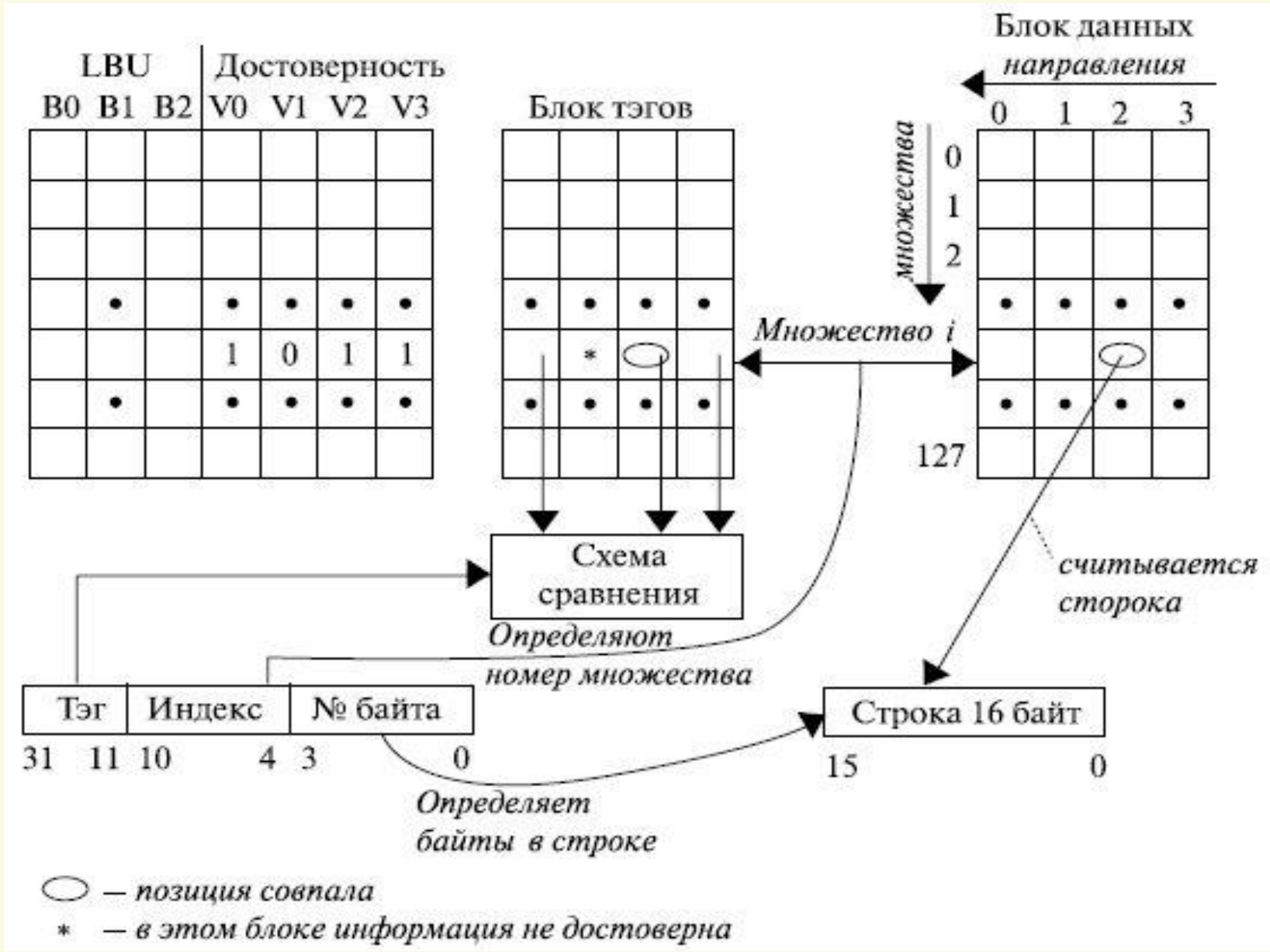
Организация внутренней кэш-памяти микропроцессора

- *Кэш-память* обычно реализуется в виде ассоциативного ЗУ, в котором для каждой строки сохраняются дополнительные сведения, называемые тегом, или признаком, в качестве которого выступает адресный код или его часть. Когда в АЗУ подается *адрес*, с ним одновременно сравниваются все теги.

Структурный состав кэш-памяти

- ***КП состоит из следующих блоков :***
- ***блока данных,***
- ***блока тегов,***
- ***блока достоверности и LRU.***

- ***Блок данных содержит 8 Кбайт данных и команд. Он разделен на 4 массива (направления), каждый из которых состоит из 128 строк.***


LBU
Достоверность
B0 B1 B2 V0 V1 V2 V3
1 0 1 1
Блок тэгов
Блок данных
направления
0 1 2 3
множества
0
1
2
127
Множество i
Схема
сравнения
считывается
сторока
Определяют
номер множества
Тэг
Индекс
№ байта
31 11 10 4 3 0
Строка 16 байт
15 0
Определяет
байты в строке
— позиция совпала
* — в этом блоке информация не достоверна

Пояснение структуры кэш-памяти

- Строка содержит данные из 16 последовательных адресов памяти начиная с адреса, кратного 16. *Индекс массивов* блока данных, состоящий из 7 *бит*, соответствует 4 строкам КП, по одной из каждого массива. Четыре строки КП с одним и тем же индексом называются множеством.

Пояснение структуры кэш-памяти

- В **блоке тегов** имеется один *тег* длиной 21 *бит* для каждой строки данных в КП. Блок тегов также разделен на 4 массива по 128 тегов. *Тег* содержит старшие 21 *бит* физического адреса данных, находящихся в соответствующей строке КП.

Пояснение структуры кэш-памяти

- В **блоке достоверности** и *LRU* содержится по одному 7-разрядному значению для каждого из 128 множеств строк КП: 4 бита достоверности (V) по одному на каждую строку *множества* и 3 бита (B0 ... B2), *управляющие* механизмом *LRU*. Биты достоверности показывают, содержит ли строка достоверные (V = 1) или недостоверные (V = 0) данные. При программной очистке КП и аппаратном сбросе процессора все биты достоверности сбрасываются в 0.

Пояснение структуры кэш-памяти

- *Адресация кэш*-памяти осуществляется путем разделения старших 28 *бит* физического адреса на 2 части. Младшие 7 *бит* из этих разрядов (разряды 10…4 физического адреса) образуют *поле* индекса и определяют множество, в котором могут храниться данные. Старшие 21 *бит* (разряды 31…11 физического адреса) служат полем тега и применяются для определения того, находится ли *информация* с данным физическим адресом в какой-либо строке выбранного *множества*.

Поиск в кэш-памяти с заданным физическим адресом

- 1. Физический адрес, по которому происходит обращение, разбивается на 3 поля: Тег, Индекс, № байта. 7 разрядов A10…A4 поля индекса определяют одно из 128 множеств.
- 2. В выбранном множестве содержатся 4 строки с информацией.

Поиск в кэш-памяти с заданным физическим адресом

- Чтобы определить, присутствует ли нужная информация в одной из строк этого множества, проводится сравнение старших 21 бита физического адреса (поле Тег) с тегами строк выбранного множества. Сравнение проводится только для достоверных строк, то есть тех, у которых в блоке достоверности установлен бит достоверности V = 1.

Поиск в кэш-памяти с заданным физическим адресом

- 3. Если для одной из строк ее тег и разряды A31...A11 физического адреса совпали, то это означает, что произошло **кэш-попадание** и необходимая информация есть в кэш-памяти.
- 4. Считывается найденная строка из 16 байт. Искомый байт в ней определяется 4 младшими разрядами физического адреса (A3…A0).

Поиск в кэш-памяти с заданным физическим адресом

- Если на этапе 3 совпадения не произошло или все строки множества недостоверны, эта ситуация определяется как **кэш-промах**. В этом случае по сформированному микропроцессором физическому адресу выполняется обращение к оперативной памяти. Из *ОЗУ* извлекается нужная информация, и содержащая ее строка записывается в свободную строку выбранного множества.

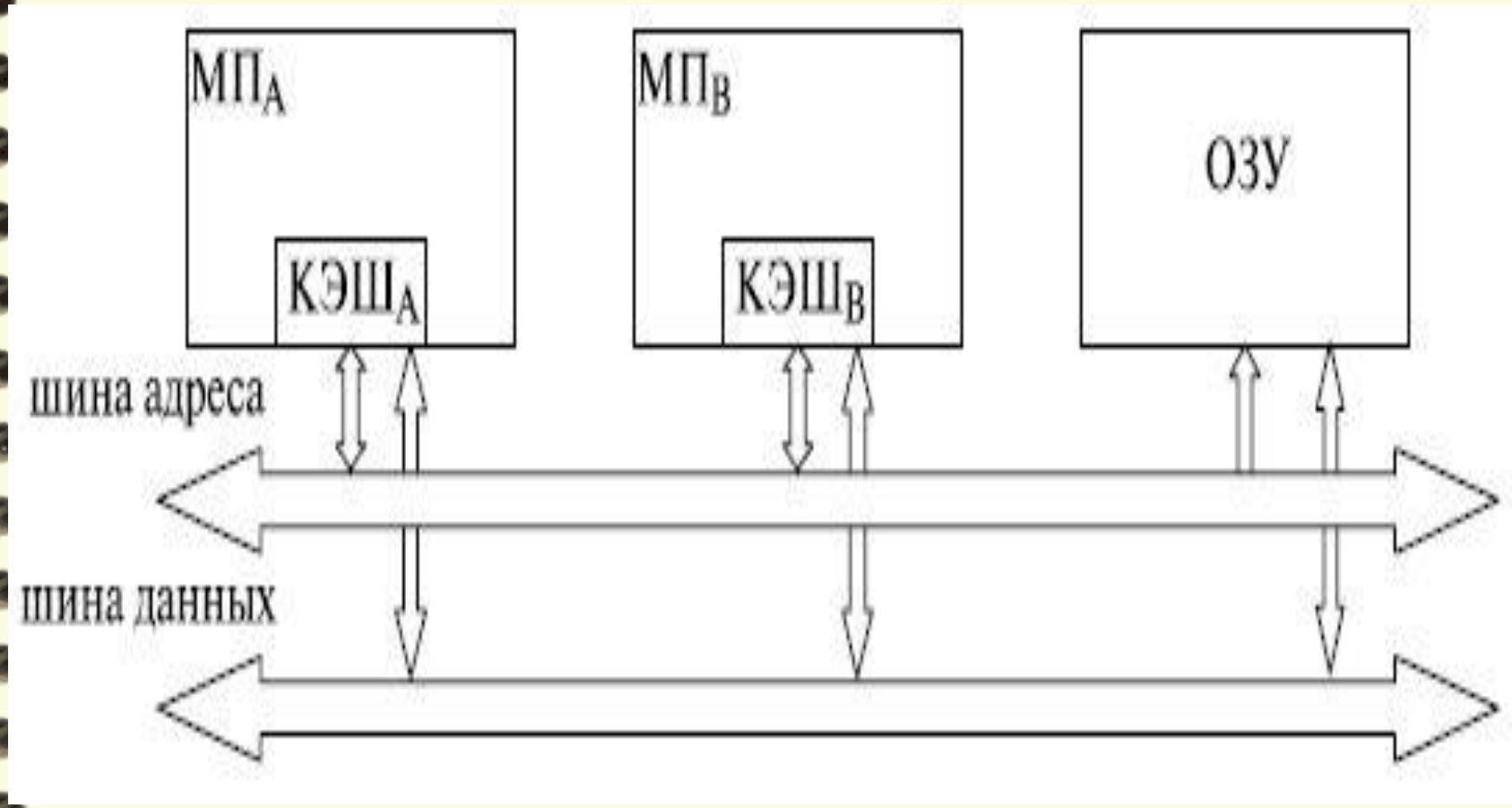
Поиск в кэш-памяти с заданным физическим адресом

- Старшие 21бит физического адреса записываются в поле тега этой строки. Если все строки в выбранном множестве достоверны, то замещается строка, к которой дольше всего не было обращений согласно механизму *LRU*. Этот механизм действует точно так же, как и при *вытеснении* строк из **буфера ассоциативной трансляции** *TLB*.

Обеспечение согласованности кэш-памяти микропроцессоров в мультипроцессорных системах

- Рассмотрим особенности работы *кэш-памяти* в том случае, когда одновременно несколько микропроцессоров используют общую оперативную *память* .
- В этом случае могут возникнуть проблемы, связанные с кэшированием информации из оперативной памяти в *кэш-память* микропроцессоров.

Структура мультимикропроцессорной системы с общей оперативной памятью



Обеспечение когерентности

- Для обеспечения согласованности (*когерентности*) памяти в мультипроцессорных системах используются аппаратные *механизмы*, позволяющие решить эту проблему. Такие *механизмы* называются **протоколами когерентности кэш-памяти**. Эти протоколы призваны гарантировать, что любое считывание элемента данных возвращает последнее по времени записанное в него *значение*.

Существует два класса протоколов *когерентности*

- **протоколы на основе справочника** (directory based): информация о состоянии блока физической памяти содержится только в одном месте, называемом справочником (физически справочник может быть распределен по узлам системы);

Существует два класса протоколов *когерентности*

- **протоколы наблюдения** (*snooping*): каждый кэш, который содержит копию данных некоторого блока физической памяти, имеет также соответствующую копию служебной информации о его состоянии; централизованная система записей отсутствует; обычно кэши расположены на *общей шине*, и контроллеры всех кэшей наблюдают за шиной (просматривают ее), чтобы определять, какие обращения по адресам в пределах этого блока происходят со стороны других микропроцессоров.

Обеспечение когерентности

- В мультипроцессорных *системах с общей* памятью наибольшей популярностью пользуются **протоколы наблюдения**, поскольку для опроса состояния кэшей они могут использовать уже существующее физическое соединение - шину памяти.
- Для поддержания *когерентности* применяется два основных метода.

Обеспечение когерентности

- Один из методов заключается в том, чтобы гарантировать, что *процессор* должен получить исключительные *права доступа* к элементу данных перед выполнением записи в этот *элемент данных*. Этот тип протоколов называется протоколом **записи с аннулированием** (write invalidate protocol), поскольку при выполнении записи он аннулирует другие копии.

Обеспечение когерентности

- Альтернативой протоколу записи с *аннулированием* является обновление всех копий элемента данных в случае записи в этот *элемент данных*.
- Этот тип протокола называется протоколом *записи с обновлением* (write update protocol), или протоколом *записи с трансляцией* (write *broadcast* protocol).

протокол MESI, который относится к группе **протоколов наблюдения с аннулированием**.

- Этот протокол использует 4 признака состояния строки *кэш*-памяти микропроцессора, по первым буквам которых и называется протокол:

признаки

- Измененное состояние (Modified): информация, хранимая в кэш-памяти микропроцессора А, достоверна только в этом кэше; она отсутствует в оперативной памяти и в кэш-памяти других микропроцессоров;

признаки

- исключительная копия (Exclusive): информация, содержащаяся в кэше А, содержится еще только в оперативной памяти;

признаки

- разделяемая информация (Shared): информация, содержащаяся в кэше А, содержится в кэш-памяти по крайней мере еще одного МП, а также в оперативной памяти;

признаки

- недостоверная информация (*Invalid*): в строке кэш-памяти находится недостоверная информация.

Примеры

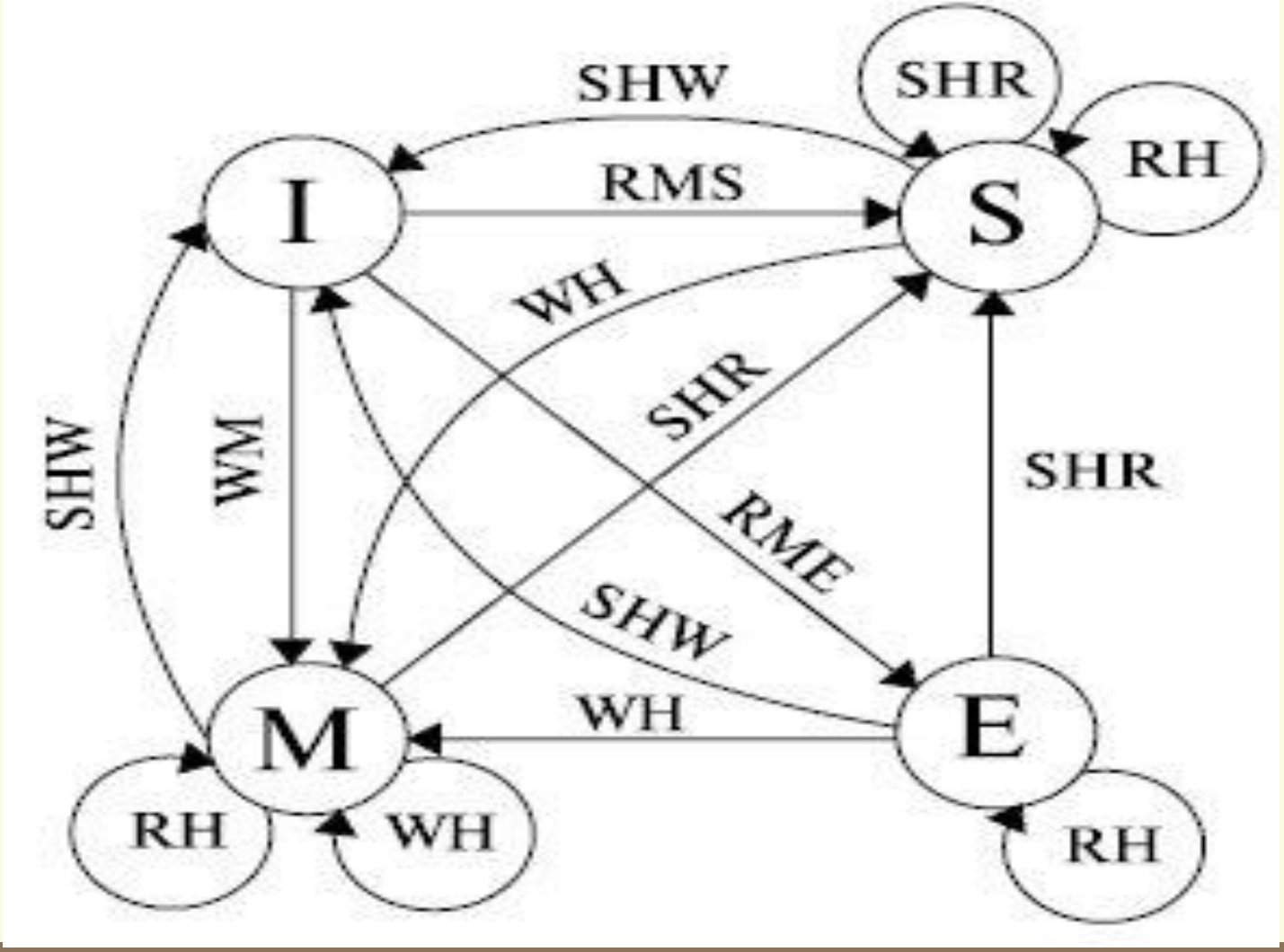
- При работе микропроцессора А с точки зрения обеспечения *когерентности* памяти возможны следующие ситуации:
- RH (Read *Hit*) - **кэш-попадание** при чтении;
- WH (Write *Hit*) - **кэш-попадание** при записи;
- RME (Read Miss Exclusive) - **кэш-промах** при чтении;

Примеры

- *RMS* (Read Miss Shared) - **кэш-промах** при чтении, но соответствующий блок есть в кэш-памяти другого микропроцессора;
- WM (Write Miss) - **кэш-промах** при записи;
- SHR (Snoop *Hit* Read) - обнаружение копии блока при прослушивании операции чтения другого кэша;
- SHW (Snoop *Hit* Write) - обнаружение копии блока при прослушивании операции записи другого кэша.

- В силу того, что в рассматриваемой мультипроцессорной системе микропроцессоры связаны общей шиной, в том числе и шиной адреса, принимая информацию по адресным линиям, *микропроцессор* определяет, было ли обращение по адресам, содержащимся в его
- *кэш*-памяти, со стороны других микропроцессоров.

MESI-диаграмма обеспечения когерентности кэш-памяти



Пояснение к MESI-диаграмме

- Пусть блок *кэш*-памяти находится в состоянии ***Modified***, то есть достоверная *информация* находится только в *кэш*-памяти данного МП. Тогда в случае обнаружения при прослушивании адресной шины обращения со стороны другого микропроцессора для чтения информации по входящим в данную строку адресам *микропроцессор* должен ***передать*** эту ***строку кэш-памяти в ОЗУ***, откуда она уже будет прочитана другим микропроцессором.

Пояснение к MESI-диаграмме

- При этом состояние строки в *кэш*-памяти рассматриваемого микропроцессора изменится с ***модифицированного на разделяемое (Shared).***

Пояснение к MESI-диаграмме

- Если строка *кэш*-памяти находилась в состоянии ***Invalid,*** то есть *информация* в ней была недостоверной, то по отношению к этой строке следует рассматривать только ситуации, связанные с **кэш-промахами**.

Пояснение к MESI-диаграмме

- Так, если произошел **кэш-промах** при выполнении *операции* записи, то необходимая строка будет занесена в *кэш-память* данного МП, в эту строку будут записаны измененные данные, и она приобретет статус исключительного владельца новой информации (Modified).