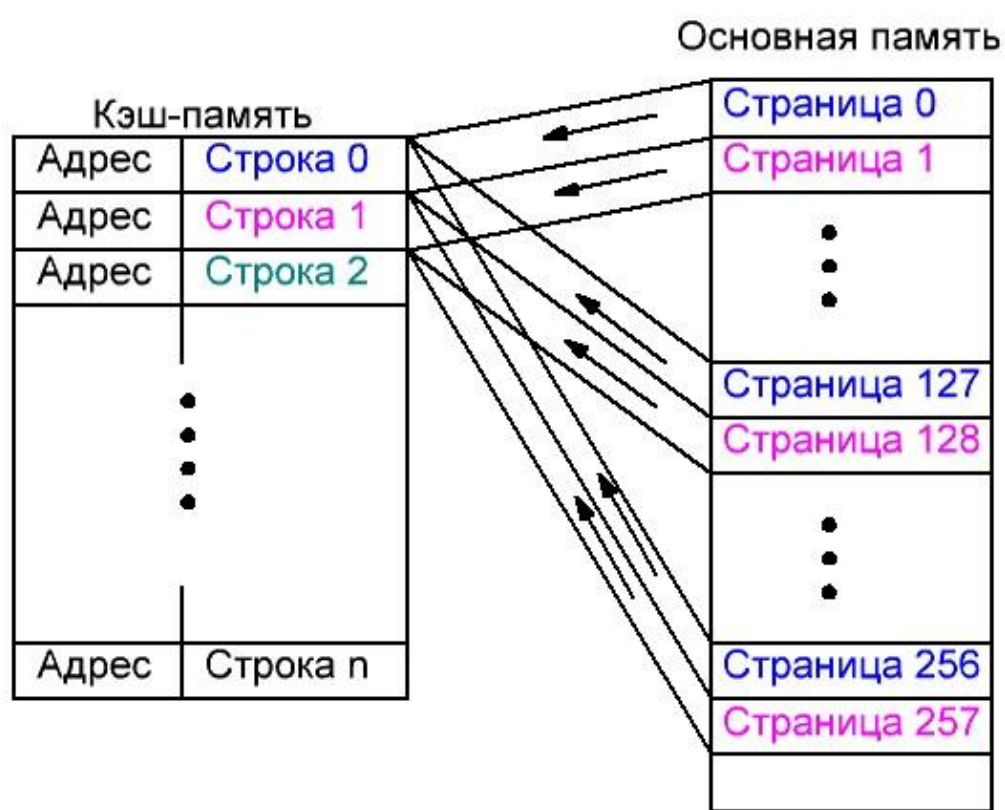


Основные структурные варианты построения кэш- памяти

Принципы размещения блоков в кэш-памяти

1. Кэш-память с прямым отображением (**direct mapped**). Каждый блок основной памяти имеет только одно фиксированное место, на котором он может появиться в кэш-памяти.



Кэш-память с прямым отображением – наиболее простая организация кэш-памяти.

При такой организации несколько блоков основной памяти строго соответствуют одной строке кэша. Для отображение адресов блоков основной памяти на адреса кэш-памяти обычно используются младшие разряды адреса блока. Таким образом, все блоки основной памяти, имеющие одинаковые младшие разряды в своем адресе, попадают на одну строку кэш-памяти.

Так как занимать строку в одно и то же время может только один блок основной памяти, то нужен специальный отличительный признак – **адресный тег**.

Поиск информации в кэш-памяти состоит в определении подмножества в которое попадает адрес ячейки основной памяти, выработанный процессором, затем в обращении к единственной соответствующей ей ячейке кэш-памяти и в единственном сравнении, устанавливающим, является ли эта ячейка желаемой.



- **Младшие разряды** (смещение) определяют положение слова в строке.
- **Средние разряды** позволяют выбрать одну из строк кэш-памяти.
- **Старшие разряды** образуют тег.

нные

ние

По адресу строки производится считывание.

Поле адресов считанной строки сравнивается с теговым адресом и если есть совпадение, вырабатывается сигнал “Попадание” (**Hit**) выдачи информации и затем при помощи мультиплексирования из строки данных выбирается искомое слово.

При загрузке из внешней памяти заменяется вся строка кэш-памяти при помощи блочной передачи, которые в современных системах осуществляются достаточно быстро.

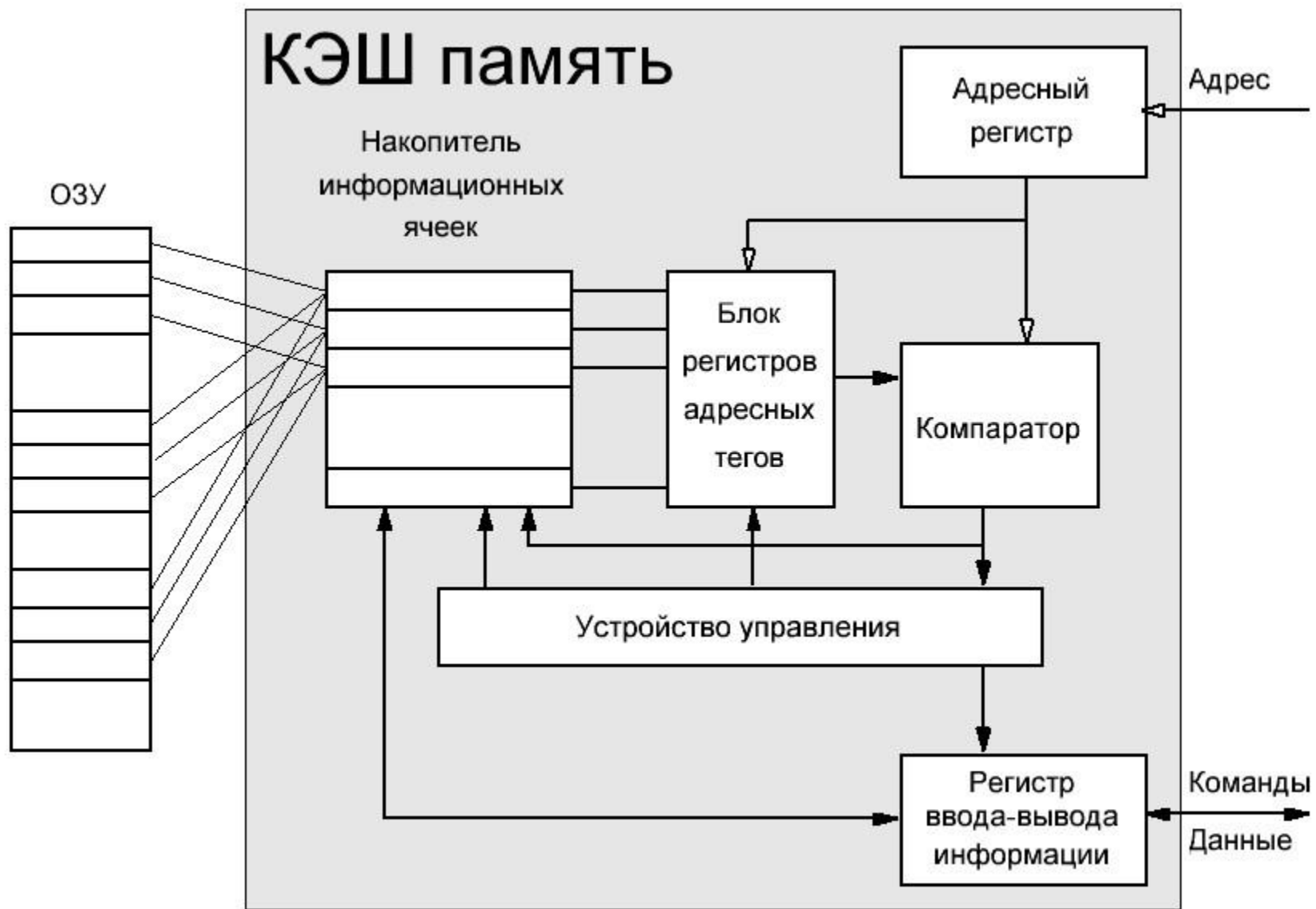


Рисунок 1 – Структура Кэш-памяти с прямым отображением

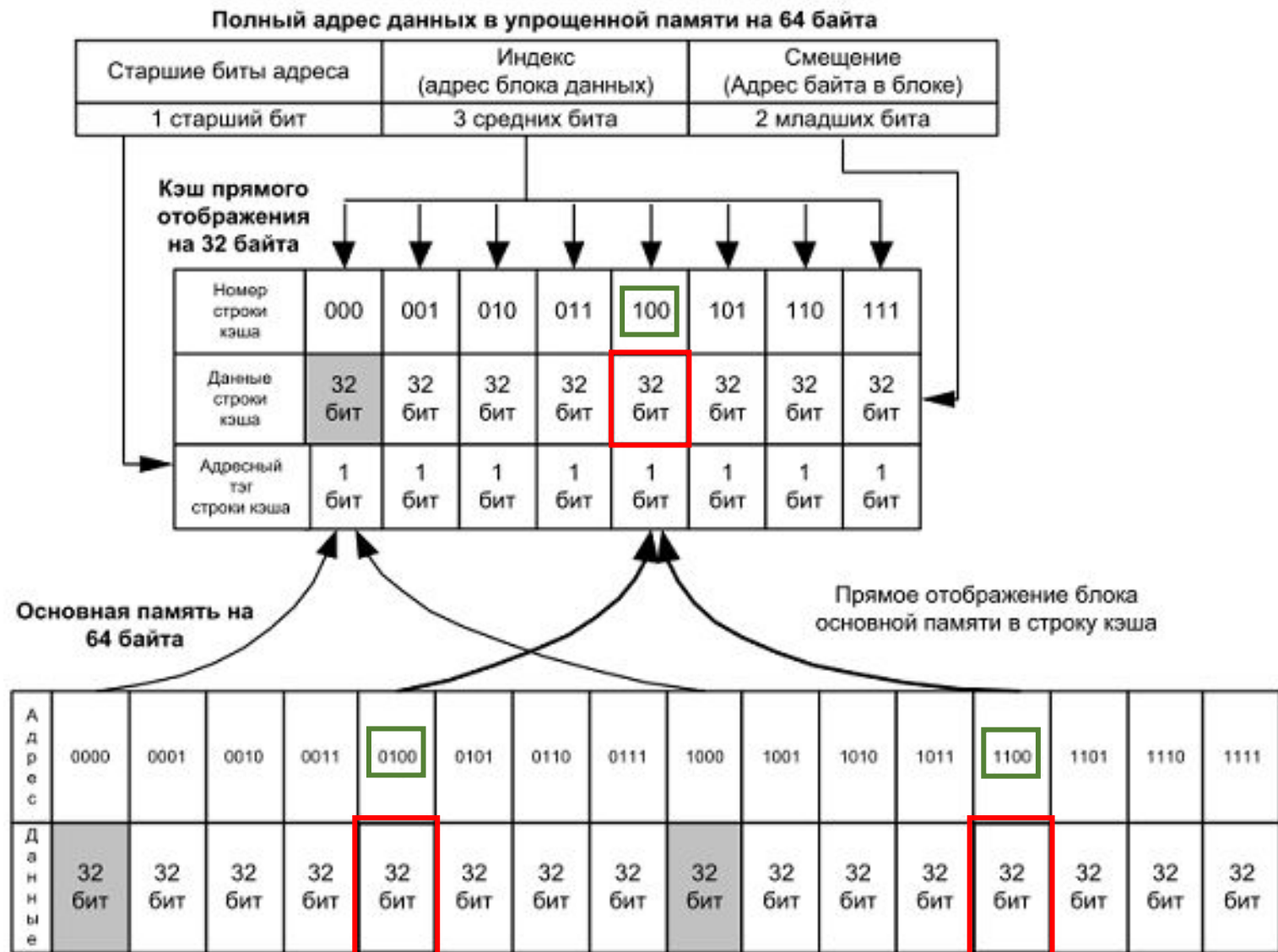


Рисунок 2 – Пример упрощенной кэш-памяти прямого отображения, имеющего вдвое меньший объем по сравнению с основной памятью.

Основное преимущество – простота аппаратной реализации и высокое быстродействие.

Необходим только **один компаратор**, **адресные теги** (указатель, который определяет какой блок в основной памяти представляет данный блок кэш-памяти. Эти теги обычно одновременно сравниваются с выработанным процессором адресом блока памяти.) **содержат меньшее число разрядов**, **накопитель может иметь форму стандартной памяти**. Это сокращает число логических уровней прохождения сигналов при поиске и выборке информации, что обеспечивает минимальное $t_{кп}$.

Основной недостаток – невысокое **число комбинаций размещения информации**.

Число комбинаций размещения информации из ОЗУ в ячейках кэш-памяти:

$$(n_{\text{ОЗУ}} / n_{\text{КП}})^{n_{\text{КП}}}$$

$n_{\text{ОЗУ}}$ – число ячеек ОЗУ

$n_{\text{КП}}$ – число ячеек кэш-памяти

2. Кэш-память с **полностью ассоциативным доступом** (FACM, *Fully Associated Cache Memory*). В структуре **FACM**, называемой также **структурой с произвольной загрузкой**, любую страницу можно загрузить в любую строку кэш-буфер. Структура такой кэш-памяти показана на рисунке 3.

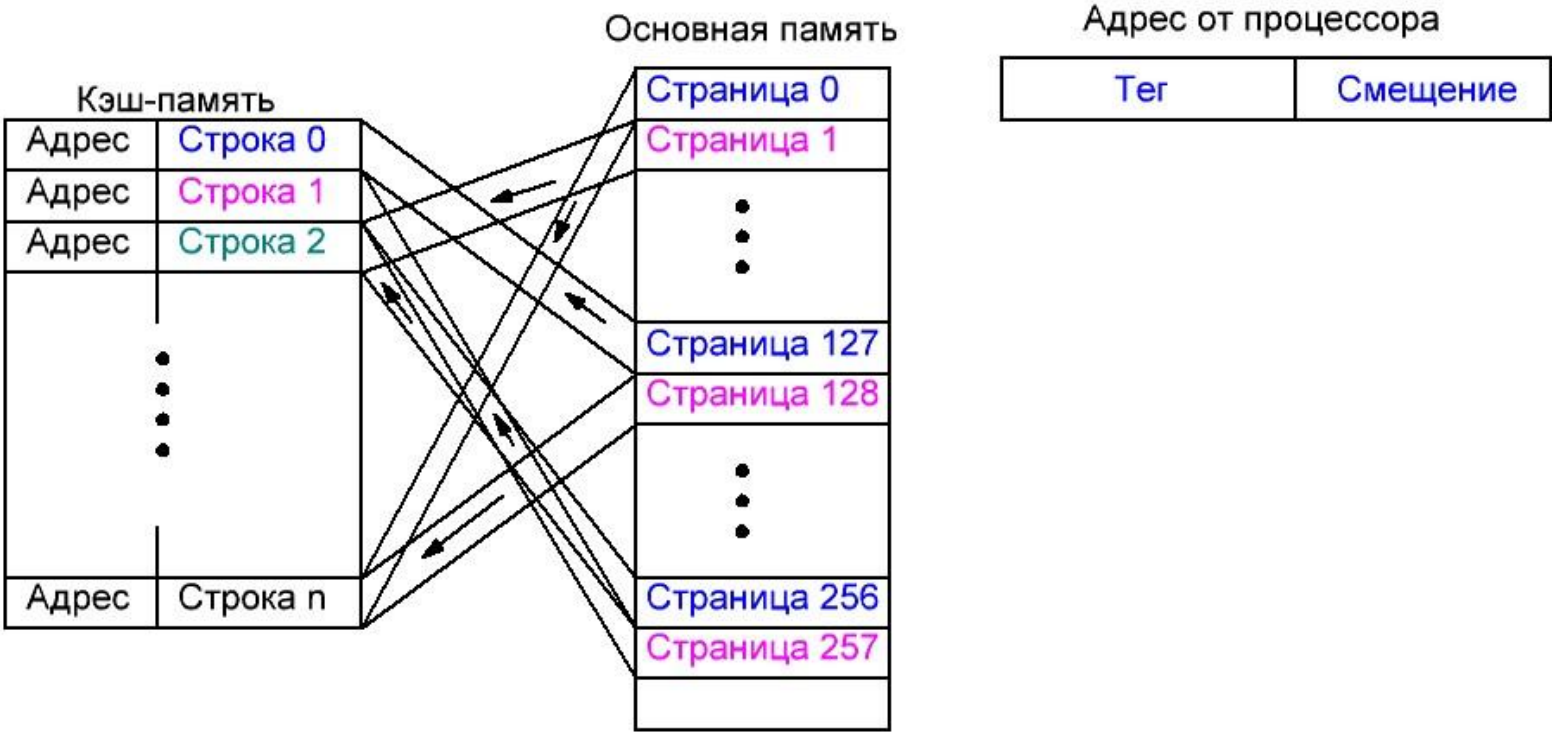
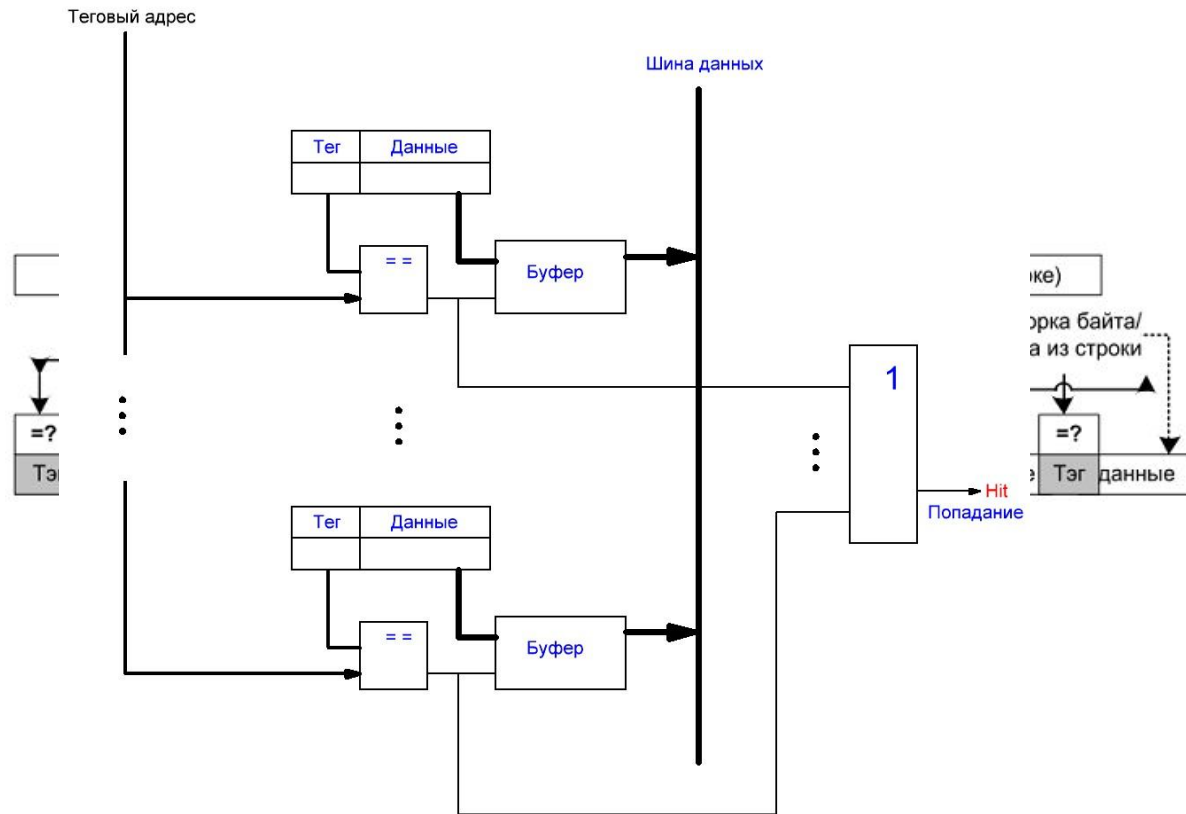


Рисунок 3 – Структура кэш-памяти с ассоциативным доступом

В качестве тега могут использоваться:

- **полный физический адрес**, если речь идет об адресации отдельных слов;
- **старшие разряды полного физического адреса** за вычетом младших (смещения), если смещение адресует слово в пределах строки, которые используются для адресации в пределах строки.

При любых обменах физический адрес запрашиваемой информации сравнивается с полями "тег" всех ячеек и при совпадении их в любой ячейке устанавливается сигнал "Попадание" (Hit).



Каждая ячейка кэш-памяти имеет свой компаратор и свой регистр адресного тега, что позволяет хранить в ней информацию из любой ячейки основного ОЗУ.

Для определения того, что блок кэш-памяти содержит достоверную или пригодную для использования информацию к тегу добавляется так называемый бит достоверности (*valid bit*).

Поиск информации осуществляется по некоторым "признакам" (адресным тегам, например адресам ячеек ОЗУ) в результате одновременного сравнения группы адресных тегов хранимых слов с заданным тегом.

При выполнении операции "чтение" и при значении сигнала Hit = 1 данные выдаются на шину данных, **если же совпадений нет (Hit = 0)**, то при чтении из основной памяти данные вместе с адресом помещаются в свободную или наиболее давно не используемую ячейку кэш-памяти.

При записи данные вместе с адресом сначала, как правило, размещаются в кэш-памяти (в обнаруженную ячейку при **Hit = 1** и свободную при **Hit = 0**) Копирование данных в основную память выполняется под управлением специального контроллера, когда нет обращений к памяти.

Число возможных комбинаций размещения информации из ОЗУ в ячейках гэш-памяти с ассоциативным доступом определяется выражением:

$$n_{\text{ОЗУ}}! / (n_{\text{ОЗУ}} - n_{\text{КП}})!$$

$n_{\text{ОЗУ}}$ – число ячеек ОЗУ

$n_{\text{КП}}$ – число ячеек кэш-памяти

Основное преимущество. Ассоциативный способ организации Кэш-памяти позволяет получить максимальное значение $P_{\text{КП}}$.

Основной недостаток. Повышенная сложность аппаратной организации, увеличенное время $t_{\text{КП}}$. Уменьшенное количество ячеек хранения по сравнению с Кэш-памятью организованной по методу прямого отображения.

Память типа **FACM** является весьма сложным устройством и используется только при малых емкостях, главным образом в специальных приложениях. В то же время этот вид кэш-памяти обеспечивает наибольшую функциональную гибкость и бесконфликтность адресов, так как любую единицу информации можно загрузить в любую ячейку кэш-памяти.

Сложность реализации полностью ассоциативного метода доступа в кэш-памяти заставляет искать иные структуры кэш-памяти, более экономичные по затратам аппаратных средств на их реализацию.

К числу таких структур относятся:

- **кэш-память с секторным отображением и**
- **кэш-память с множественно-ассоциативной архитектурой** (с ассоциацией по нескольким направлениям).

3. **Кэш-память с множественно-ассоциативным доступом (set associative).** Некоторый блок основной памяти может располагаться на ограниченном множестве мест в кэш-памяти. **Объединяет в себе прямой и ассоциативный методы доступа.**

Обычно множество представляет собой группу из двух или большего числа непересекающихся блоков в кэш-памяти. Если множество состоит из n блоков, то такое размещение называется **множественно-ассоциативным с n каналами.**

Для размещения блока прежде всего необходимо определить множество, которое определяется **индексом** – младшими разрядами адреса блока памяти :

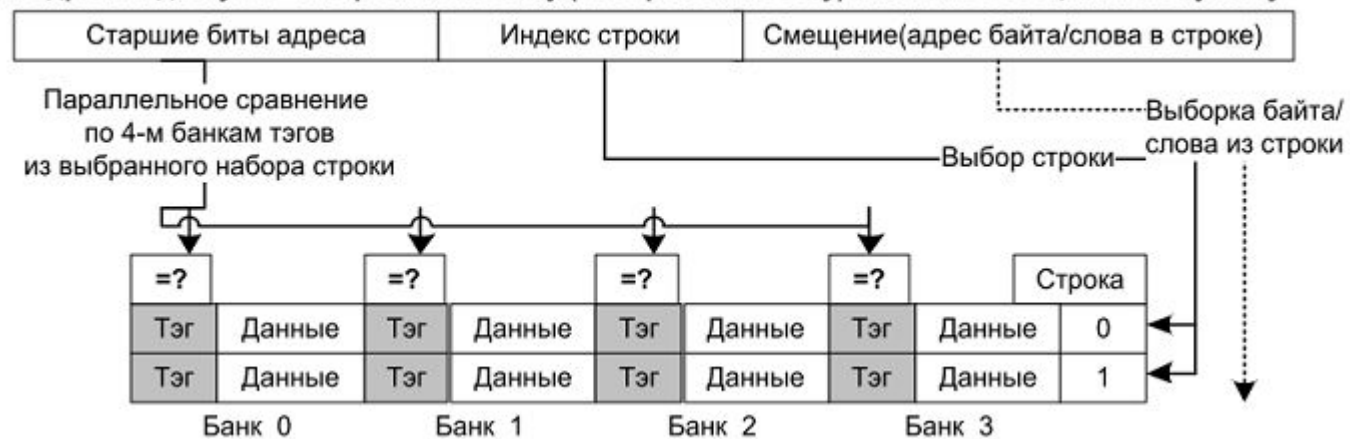
$$\begin{aligned} &(\text{адрес множества кэш-памяти}) = \\ &= (\text{адрес блока основной памяти}) \bmod (\text{число множеств в кэш-памяти}) \end{aligned}$$

Далее, блок может размещаться на любом месте данного множества.

Адресация множественно-ассоциативной кэш-памяти осуществляется путем **деления адреса**, поступающего из процессора, на три части:

- **поле смещения** используется для выбора байта внутри блока кэш-памяти,
- **поле индекса** определяет номер множества,
- **поле тега** используется для сравнения.

Формат адреса и доступ к четырехканальному (четырехбанковому) частично-ассоциативному кэшу



При фиксированном общем размере кэш-памяти, **увеличение** степени **ассоциативности** приводит к **увеличению количества блоков** в множестве, при этом **уменьшается размер индекса** и **увеличивается размер тега**.

Выбор требуемого **множества** при обращении к кэш-памяти осуществляется **методом прямого отображения**, а **поиск** нужной **ячейки** внутри множества **ассоциативным методом**.

Число возможных комбинаций размещения информации в кэш-памяти с множественно-ассоциативным доступом имеет вид:

$$\left[\frac{(n_{OЗУ} / M_{КП})!}{(n_{OЗУ} - n_{КП} / M_{КП})!} \right]^{M_{КП}}$$

Если кэш-память имеет только одно подмножество $M_{КП} = 1$, этот вариант сводится к **полностью ассоциативному**, а при $M_{КП} = n_{КП}$ – к методу **доступа с прямым отображением**.

4. **Кэш-память с секторным отображением.** Это другой вариант кэш-памяти с множественно-ассоциативным методом доступа.

В кэш-памяти с секторным способом отображения накопитель также разделён на некоторое количество непересекающихся подмножеств $M_{кп}$. Но **доступ к каждому сектору** (подмножеству) не прямой, а **ассоциативный**. **Внутри** же каждого подмножества **поиск информации** осуществляется по **методу прямого отображения**.

Число возможных комбинаций размещения информации из ОЗУ в Кэш-памяти с секторным отображением равно:

$$\frac{(n_{ОЗУ} \cdot M_{кп} / n_{кп})!}{((n_{ОЗУ} \cdot M_{кп} / n_{кп}) - M_{кп})!} = \left(\frac{n_{ОЗУ} \cdot M_{кп}}{n_{кп}} \right)^{M_{кп}} \quad \text{при } n_{кп} \ll n_{ОЗУ}$$

При предельном случае $M_{кп} = n_{кп}$, соответствует **полностью ассоциативному** методу доступа, а при $n_{кп} \gg M_{кп}$ число комбинаций для секторной Кэш-памяти оказывается гораздо меньше по сравнению с множественно-ассоциативным методом доступа.

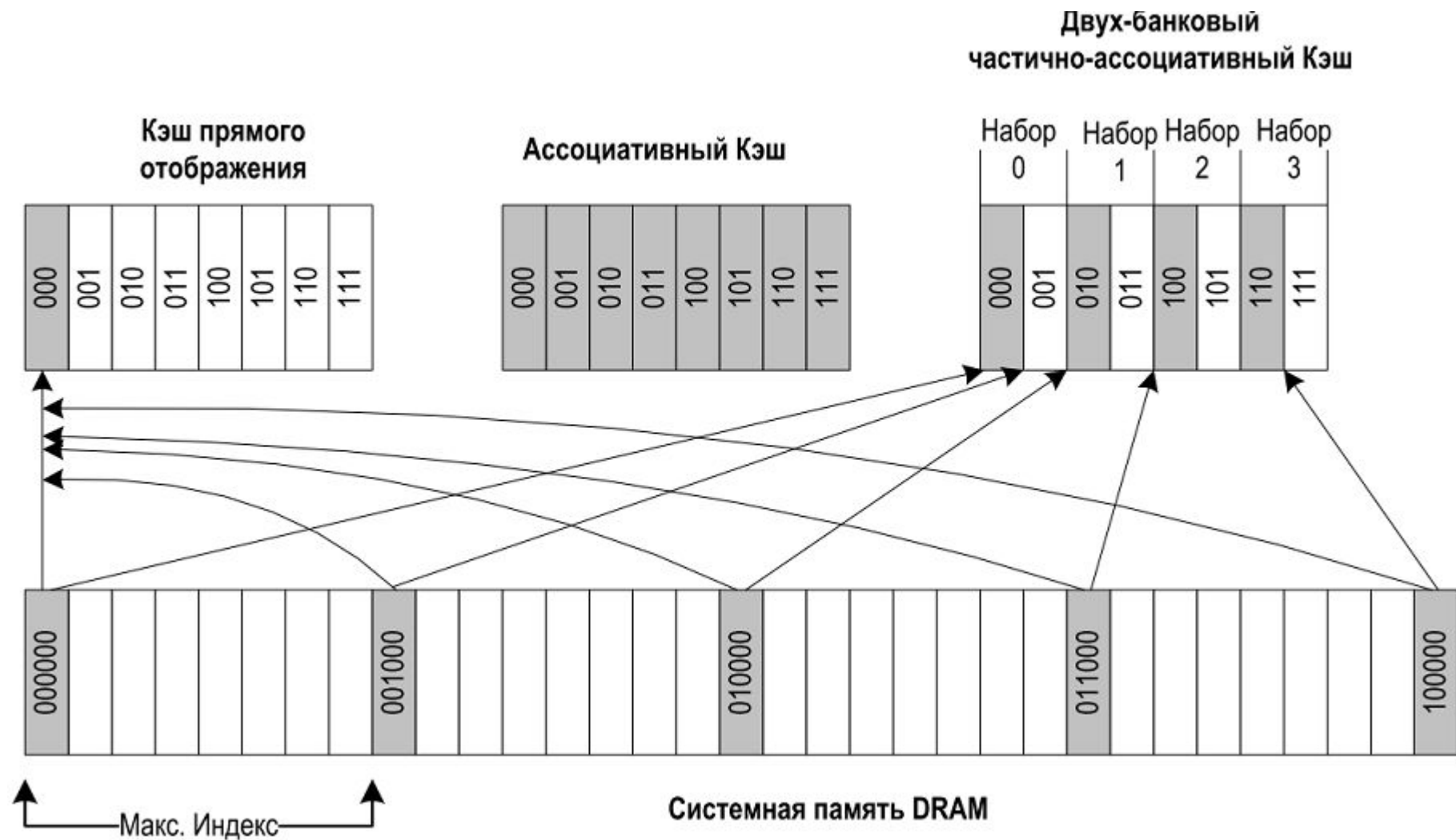


Рисунок 4 – Отображение блоков данных из основной памяти в кэш-память прямого отображения, ассоциативную кэш-память и частично-ассоциативную кэш-память.

Условие замещения блока кэш-памяти при промахе

При возникновении промаха, контроллер кэш-памяти должен **выбрать подлежащий замещению блок**.

При использовании кэш-памяти с **прямым отображением** применяются наиболее **простые аппаратные решения**. На попадание проверяется только один блок и только этот блок может быть замещен.

При **полностью ассоциативной** или **множественно-ассоциативной** организации кэш-памяти имеются несколько блоков, из которых надо выбрать кандидата в случае промаха. Как правило для замещения блоков применяются **две основных стратегии**: **случайная** (Random) и **LRU** (Least-Recently Used – давно не используемый блок).

Случайная. Чтобы иметь равномерное распределение, **блоки-кандидаты выбираются случайно (используется генератор случайных чисел) для определения номера банка**. В некоторых системах, чтобы получить воспроизводимое поведение, которое особенно полезно во время отладки аппаратуры, используют псевдослучайный алгоритм замещения.

Достоинство: простота реализации в аппаратуре.

LRU. Чтобы уменьшить вероятность выбрасывания информации, которая скоро может потребоваться, все обращения к блокам фиксируются. **Заменяется тот блок, который не использовался дольше всех**.

Недостаток: когда количество блоков для поддержания трассы увеличивается, алгоритм LRU становится все более дорогим и часто только приближенным.

Организация чтения блока кэш-памяти

При обращениях к кэш-памяти на реальных программах преобладают **обращения по чтению**.

Все обращения за командами являются обращениями по чтению и большинство команд не пишут в память.

В общем случае для ускорения операций с кэш-памятью **проводят оптимизацию кэш-памяти для выполнения операций чтения**.

Блок из кэш-памяти может быть прочитан в то же самое время, когда читается и сравнивается его тег.

Чтение блока начинается сразу как только становится доступным адрес блока. **Если чтение происходит с попаданием**, то **блок немедленно направляется в процессор**.

Если происходит промах, то ничего не происходит, поскольку от заранее считанного блока нет никакой пользы, правда нет и никакого вреда.

Организация записи блока кэш-

памяти

Обычно **операции записи** составляют менее 10% общего трафика памяти.

Так как процессор определяет размер записи (обычно от 1 до 8 байтов), то только эта часть блока может быть изменена.

В общем случае это подразумевает **выполнение над блоком последовательности операций чтение-модификация-запись**: **чтение оригинала блока, модификацию его части и запись нового значения блока**. Модификация блока не может начинаться до тех пор, пока проверяется тег, чтобы убедиться в том, что обращение является попаданием. Поскольку проверка тегов не может выполняться параллельно с другой работой, то **операции записи отнимают больше времени, чем операции чтения**.

Организация кэш-памяти различает две стратегией выполнения записи:

1. **сквозная запись** — информация записывается в два места: в блок кэш-памяти и в блок более низкого уровня памяти (основная память).

2. **запись с обратным копированием** — информация записывается только в блок кэш-памяти.

Модифицированный блок кэш-памяти записывается в основную память только когда он замещается. Для сокращения частоты копирования блоков при замещении обычно с каждым блоком кэш-памяти связывается так называемый **бит модификации**. Этот бит состояния показывает был ли модифицирован блок, находящийся в кэш-памяти. Если он не модифицировался, то обратное копирование отменяется, поскольку более низкий уровень содержит ту же самую информацию, что и кэш-память.

Преимущества и недостатки подходов к организации

записи

При записи с обратным копированием **операции записи выполняются со скоростью кэш-памяти**, и несколько записей в один и тот же блок требуют только одной записи в память более низкого уровня. В этом случае обращения к основной памяти происходят реже, что уменьшает требования к быстродействию. Это очень привлекательно для мультипроцессорных систем.

При сквозной записи промахи по чтению не влияют на записи в более высокий уровень, и, кроме того, сквозная запись проще для реализации, чем запись с обратным копированием. Сквозная запись имеет также преимущество в том, что основная память имеет наиболее свежую копию данных. Это важно в мультипроцессорных системах, а также для организации ввода/вывода.

При промахе во время записи имеются две дополнительные возможности:

1. **разместить запись в кэш-памяти.** Блок загружается в кэш-память, вслед за чем выполняются действия аналогичные выполняющимся при выполнении записи с попаданием. Это похоже на промах при чтении и используется в кэш-памяти, реализующей запись с обратным копированием (в надежде, что последующая запись в этот блок будет перехвачена),
2. **не размещать запись в кэш-памяти.** Блок модифицируется на более низком уровне и не загружается в кэш-память. Такой метод используется в кэш-памяти со сквозной записью (поскольку последующая запись в этот блок все равно пойдет в память).

Основные параметры кэш-памяти

Основные параметры, которые характеризуют кэш-память различных типов:

- **Объем кэш-памяти** (сегмент данных и сегмент адресных тегов).
- **Размер строки** (блока) кэш-памяти.
- **Степень** (уровень) **ассоциативности** кэш-памяти (Прямое отображение, ассоциативный, частично-ассоциативный).
- **Алгоритм замещения данных** (выбор строки для удаления) при необходимости загрузки новых данных в кэш из основной памяти (LRU, Random)
- **Алгоритм записи данных** в кэш-память из ЦП (кэш с обратной записью и кэш со сквозной записью)

Объем кэш-памяти всегда влияет на общую производительность системы памяти по следующим причинам:

- При большем объеме больше данных может находиться внутри кэш-памяти и вероятность того, что нужные очередной команде данные резидентны в кэше, всегда будет выше (вероятность промахов уменьшается);
- Объем кэш-памяти может варьироваться от сотен байт до десятков Мегабайт в зависимости от области применения (ЦП для сотовых телефонов и рабочих станций, жесткий диск, сетевые контроллеры и файловые системы).

Размер строки кэш-памяти, как правило, определяется шириной транзакций с основной памятью и ее режимами работы. Например, для шины основной памяти шириной 32 бит и двойным темпом обмена данными, за один цикл обмена кэш-контроллер может послать или получить 64 бита данных. В пакетном режиме обмена (burst) за 4 цикла можно получить 256 бит. Поэтому для 32-разрядных систем размеры строки кэша варьируются между 64 и 256 бит.

Степень ассоциативности кэш-памяти сильно влияет на быстродействие.

- Кэш прямого отображения является наиболее быстродействующим, так как тест на попадание/промах производится одновременно с считыванием данных и в случае промаха считанные данные просто аннулируются.
- В ассоциативном и частично-ассоциативном кэшах данные появляются только после результата теста, что обуславливает дополнительную задержку.

Алгоритмы замещения данных являются актуальными только для полностью ассоциативного и частично-ассоциативного кэша.

В случае кэша прямого отображения выбора нет, для каждого значения индекса имеется только один блок, который должен быть замещен новыми данными и в этом случае содержание этого блока кэша просто объявляется недостоверным, либо в дополнение копируется в основную память, если данные были изменены.

Алгоритмы записи данных в кэш-память включают две основные группы:

- **Обратная запись** (Write Back), когда данные пишутся только в соответствующую строку кэш-память и дублируются в основную память только при замещении данных. Последовательность операций записи в один и тот же адрес кэша не вызывает избыточных операций записи в основную память и **уменьшает нагрузку на интерфейс основной памяти и общую шину системы**.
- **Сквозная запись** (Write Trough), когда данные пишутся одновременно в кэш-память и основную память с использованием FIFO-буфера записи (очередь записи). Буфер записи данных позволяет ЦП записывать последовательные блоки данных, не заботясь о фактическом моменте их появления в основной памяти. Прوماхи чтения не блокируют запись блоков в память из-за наличия параллельного тракта доступа к памяти с буфером записи.