Организация кэш-памяти

- Назначение
- Обмен данными
- Стратегия замещения
- Функции отображения

Назначение

- Кэш-память предназначена для хранения блоков данных и команд программы, выполняемой процессором в текущий момент времени
- Она представляет собой быстродействующую буферную память ограниченного объема, которая располагается между процессором и относительно медленной оперативной памятью (ОП)
- Физически кэш-память строится на микросхемах SRAM (Static Random Access Memory) и контроллере кэша.

Назначение

- Кэш-память должна иметь средства:

- реализующие процедуры обмена данными между оперативной памятью и кэш;
- определяющие, находится ли в кэш блок со словом, которое требуется процессору;
- выбирающие слот кэша, подлежащий замещению в случае промаха;
- выполняющие стратегию записи.

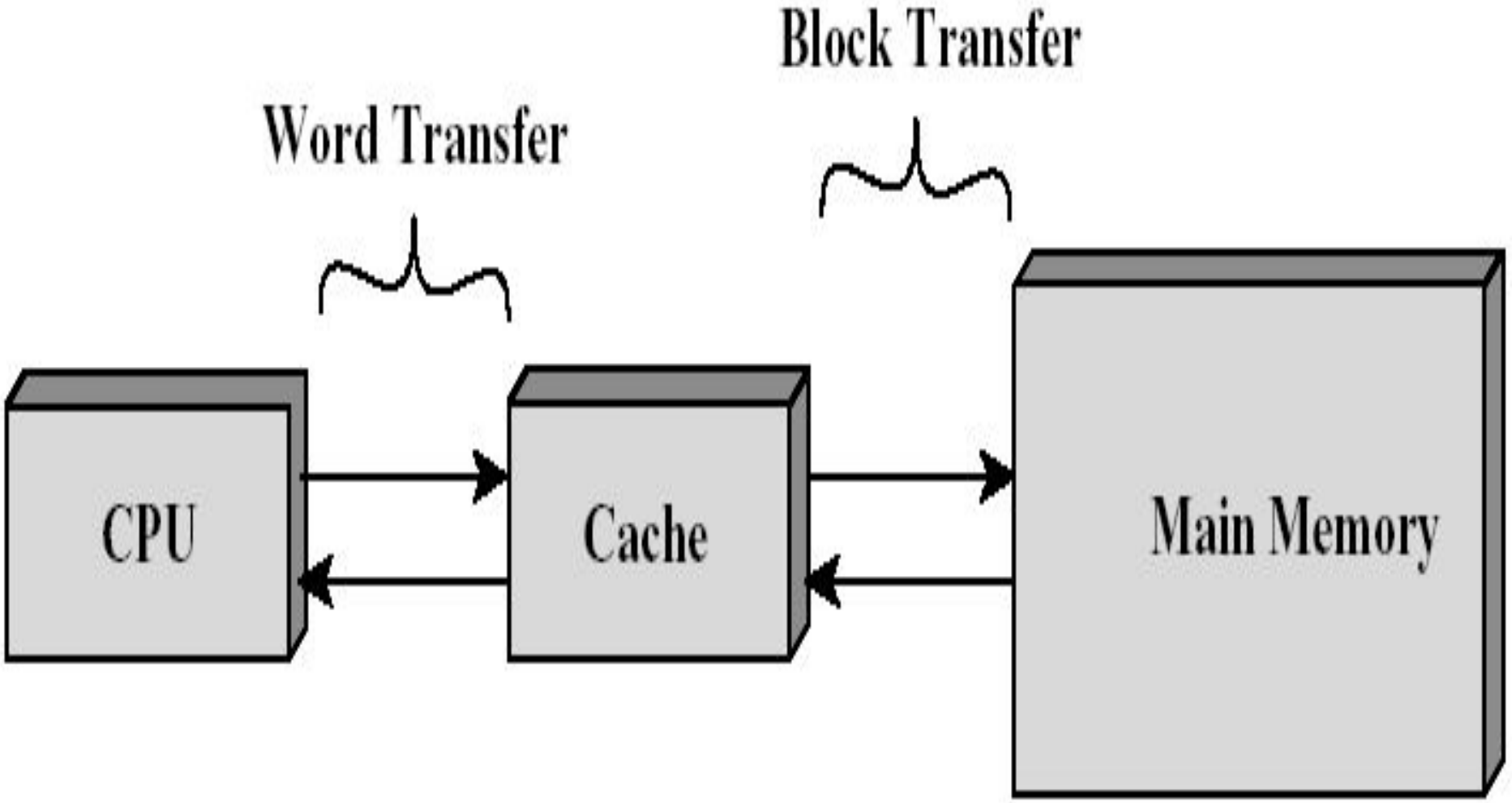
Концепция обмена данными между ОП и кэшем

- В компьютере имеется относительно большая и медленная ОП вместе с меньшей, более быстрой кэш-памятью
- Кэш содержит копии частей оперативной памяти
- Когда ЦП пытается читать слово из памяти, делается проверка, находится ли это слово в кэше

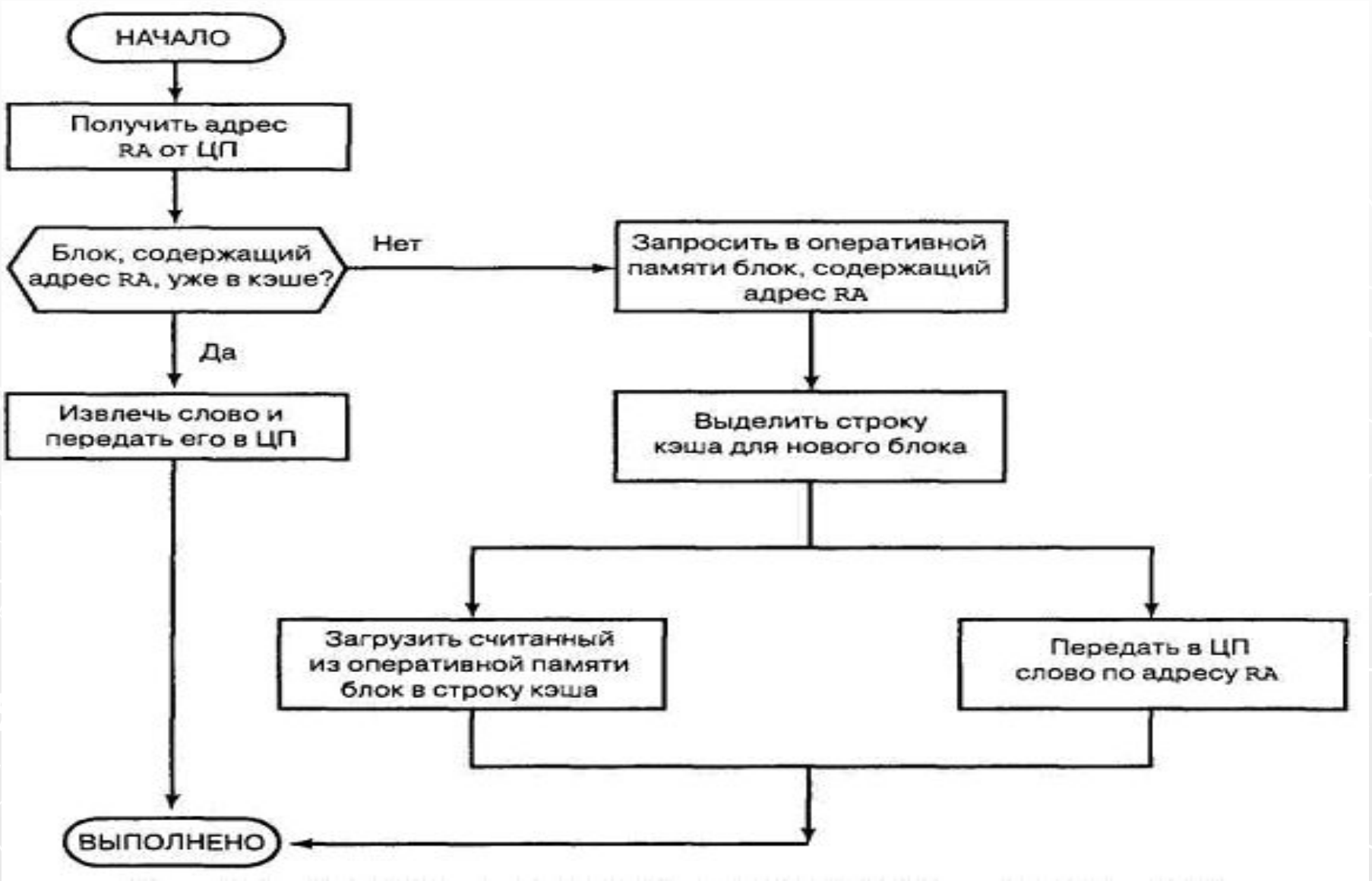
Концепция обмена данными между ОП и кэшем

- Если слово находится в кэш, оно поступает в ЦП
- В противном случае, блок оперативной памяти, состоящий из некоторого фиксированного числа байт, читается в кэш и затем слово поступает в ЦП

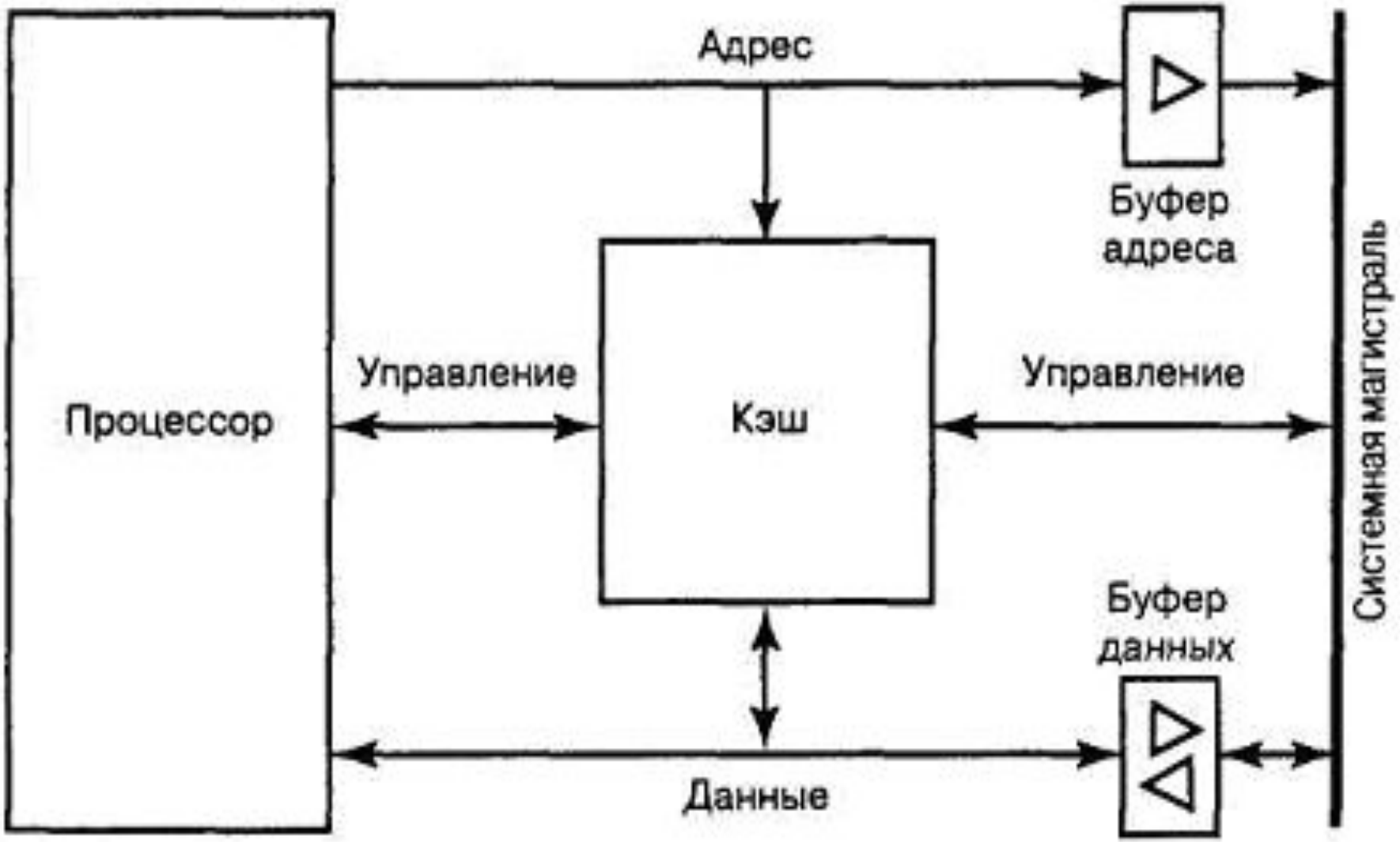
Взаимодействие системы кэш-ОП



Алгоритм выполнения операции чтения слова из кэша



Структурная схема блока кэш-памяти



Структурная схема блока кэш-памяти

- Кэш соединен с процессором линиями адреса, данных и управляющих сигналов
- Линии адреса и данных подключены также к буферам адреса и данных, которые имеют выход на системную магистраль, а через нее могут обмениваться данными с оперативной памятью

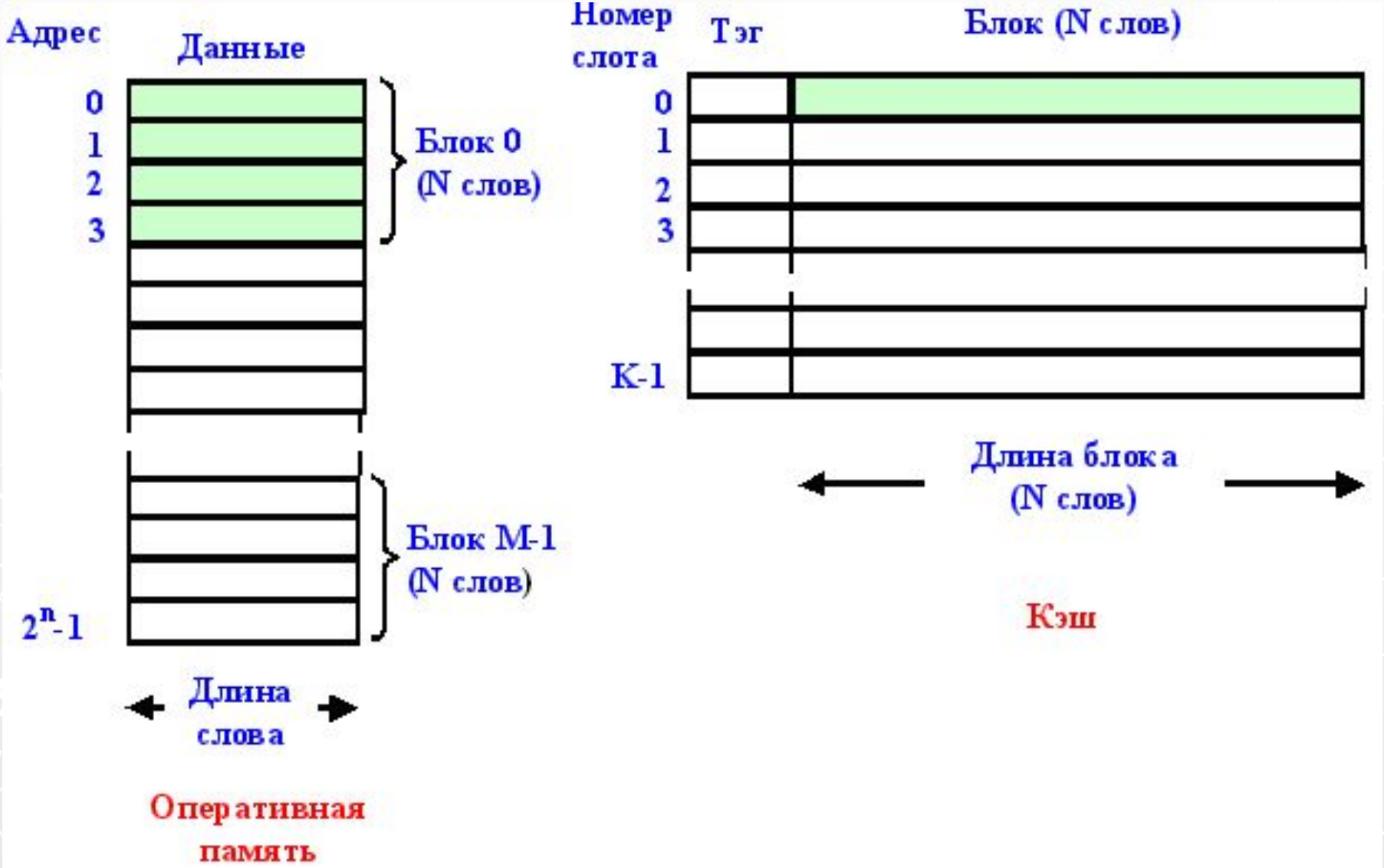
Структурная схема блока кэш-памяти

- Если интересующие процессор данные уже находится в кэше, буферы адреса и данных блокируются, и весь обмен идет, минуя системную магистраль
- Если же оказывается, что нужных процессору данных в кэше нет, затребованный процессором адрес загружается в буфер адреса и передается на системную магистраль
- Прочитанные из оперативной памяти данные помещаются в буфер данных и передаются из него в кэш и в процессор.

Понятие о локализации ссылок (locality reference)

- Обращения к памяти в процессе выполнения фрагмента программы имеют тенденцию "скапливаться" в ограниченной области ***(кластере — cluster)*** адресного пространства оперативной памяти
- По мере выполнения сложной программы текущий кластер смещается в адресном пространстве, но на коротком отрезке времени можно считать, что процессор обращается к фиксированному кластеру
- Таким образом, если блок данных выбран в кэш по запросу процессора, то, вероятно, что будущие обращения процессора будут к другим словам этого блока.

Логическая организация системы кэш - ОП



Логическая организация системы кэш - ОП

- Оперативная память состоит из 2^n адресуемых слов
- Каждое слово имеет уникальный адрес
- В системе кэш - ОП эта память рассматривается, как состоящая из ряда блоков фиксированной длины по N слов каждый
- Следовательно, ОП логически разбивается на $M = 2^n/N$ блоков

Логическая организация системы кэш - ОП

- Кэш состоит из K слотов (линий) по N слов каждый
- В связи с тем, что кэш-память значительно дороже, чем ОП, число слотов кэш значительно меньше, чем число блоков оперативной памяти (K<<M)
- Каждый блок ОП снабжается специальной меткой - тегом (tag), которая однозначно идентифицирует блок, находящийся в текущее время в кэше
- Тэг обычно представляет часть исполнительного адреса памяти

Логическая организация системы кэш - ОП

- Производительность системы кэш - ОП и компьютера в целом возрастает, когда кэш размещается в одном чипе с процессором
- В этом случае выходные цепи кэш могут непосредственно подсоединяться к арифметико-логическому устройству и регистрам процессора, что позволяет кэш работать на частоте процессора
- При этом снижается время доступа к кэш

Функционирование кэш

- Когда процессор формирует физический адрес обращения к памяти, этот адрес сначала посылается в кэш
- Происходит проверка, находится ли адресуемое слово в кэше
- Если процессор находит это слово в кэше, то данные извлекаются
- В этом случае произошло так называемое кэш-попадание (hit)

Функционирование кэш

- Если процессор не находит адресуемого слова в кэше, блок ОП, содержащий затребованное процессором слово, читается в кэш и одновременно слово поступает в процессор
- Такой случай называется кэш-промахом (miss)

Стратегия замещения

- При возникновении промаха контроллер кэш-памяти должен выбрать подлежащий замещению блок
- Используются три основных стратегии, определяющие какой блок должен быть выгружен из кэш-памяти

Стратегия замещения

- Random -случайно выбранный блок ;
- LFU (least frequently used) - наименее используемый блок ;
- LRU (least recently used) - наиболее давно использовавшийся блок

Стратегия замещения Random -случайно выбранный блок

- Блок-кандидат на замещение выбирается случайно; происходит замещение этого блока новым, запрашиваемым ЦП в текущий момент времени
- Преимущества:

  - Метод легко реализуем аппаратурой;

  - Быстрее остальных методов.
- Недостаток:

  - Выгруженный из кэша блок может потребоваться в следующем цикле обращения ЦП к памяти, что потребует выполнения алгоритма замещения для только что выгруженного из кэша блока.

Стратегия замещения
LFU -наименее используемый блок

- Для замещения выбирается блок в КЭШе, к которому было наименьшее количество ссылок
- Реализация алгоритма LFU требует применения для каждого блока кэша специального счетчика, с помощью которого вычисляется общее количество обращений к блоку за время пребывания его в кэше

Стратегия замещения LFU -наименее используемый блок

- **Недостатки:**

- Блок, загруженный в кэш последним имеет, наименьшее число обращений и будет выгружен в первую очередь, хотя он может потребоваться ЦП в ближайшее время
- Аппаратная реализация оказывается сложной и дорогой

Стратегия замещения
LRU - наиболее давно использовавшийся блок

- Метод LRU уменьшает вероятность выбрасывания блоков данных, которые вскоре могут потребоваться ЦП
- LRU характеризуется лучшим отношением производительность/стоимость по сравнению с другими методами и более чаще применяется в реальных системах

Стратегия замещения
LRU - наиболее давно использовавшийся блок

- Идея применения этого метода состоит в том, что блок данных, к которому ЦП не обращался в течение продолжительного периода времени, имеет мало шансов быть востребованным в ближайшем будущем
- Для этого в методе LRU используется механизм отслеживания: к какому блоку кэш процессор обращался наиболее давно

Стратегия замещения
LRU - наиболее давно использовавшийся блок

- Одна из возможных реализаций этого механизма состоит в использовании счетчика для каждого блока в кэш-памяти
- При каждом обращении к кэш все счетчики блоков инкрементируются за исключением того счетчика, к блоку которого произошло обращение
- Этот счетчик сбрасывается в ноль
- В этом случае замещению подлежит блок кэш-памяти, содержимое счетчика которого имеет наибольшее значение, т.е. блок не использовался дольше всех.

Стратегия записи

- Когда процессор выполняет процедуру записи слова в память, то для обеспечения нормальной работы системы процессор-память используется специальный механизм, обеспечивающий **когерентность данных**
- **Когерентность** предусматривает соответствие содержимого ОП и кэш-памяти
- Это соответствие достигается использованием механизмов ***сквозной записи*** **(*write through*)** или ***обратной записи*** **(*write back*)**

Стратегия записи

- ***Write through - сквозная запись***

  При сквозной записи слово одновременно изменяется как в кэш, так и в оперативной памяти
- Таким образом, при любом цикле записи, даже в случае попадания в кэш-память, происходит обращение к системной шине

Стратегия записи

- **Преимущество** - содержимое слотов кэш-памяти всегда соответствует содержимому блоков ОП
- **Недостаток -** снижение производительности процессора, так как циклы обращения к ОП требуют использования системной шины, работающей с тактовой частотой системной платы, которая значительно ниже тактовой частоты процессора

Стратегия записи

- ***Write back - обратная запись***
- При использовании стратегии обратной записи минимизируется количество обращений к ОП
- В механизме обратной записи каждый слот кэш-памяти снабжается специальным служебным битом, получившим название **бит модификации (бит М)**
- Если блок ОП загружается в кэш, то бит М сбрасывается в нулевое состояние

Write back

- В случае, если содержимое блока в кэше было изменено, **бит модификации** соответствующего слота устанавливается в **единичное состояние**
- Таким образом, все изменения в словах происходят только в кэш

Write back

- Лишь при выполнении алгоритма замещения слот, бит модификации которого установлен в единичное состояние, переписывается обратно в ОП
- Если блок в кэше не модифицировался, то обратное копирование отменяется, поскольку более низкий уровень памяти содержит те же самые данные, что и кэш-память

Write back

- **Преимущество**

  - блок ОП может находится в кэш-памяти длительное время и процессор имеет возможность неоднократно изменять содержимое блока без обращения к ОП

- **Недостатки**

- нарушение когерентности;
- необходимость использования дополнительного бита модификации для каждого слота приводит к усложнению аппаратной реализации кэша.

Стратегия записи

- В современных процессорах наиболее часто используется механизм обратной записи, хотя многие из них могут использовать как сквозную, так и обратную запись

Функции отображения

- Так как количество строк кэша меньше, чем блоков ОП, необходим механизм отображения блоков ОП в слоты кэша
- В зависимости от принятого в компьютере принципа размещения блоков в кэш-памяти применяются три типа их организации

Функции отображения

- Кэш прямого отображения
- Ассоциативная функция отображения
- Секционированная (наборно) ассоциативная функция отображения.

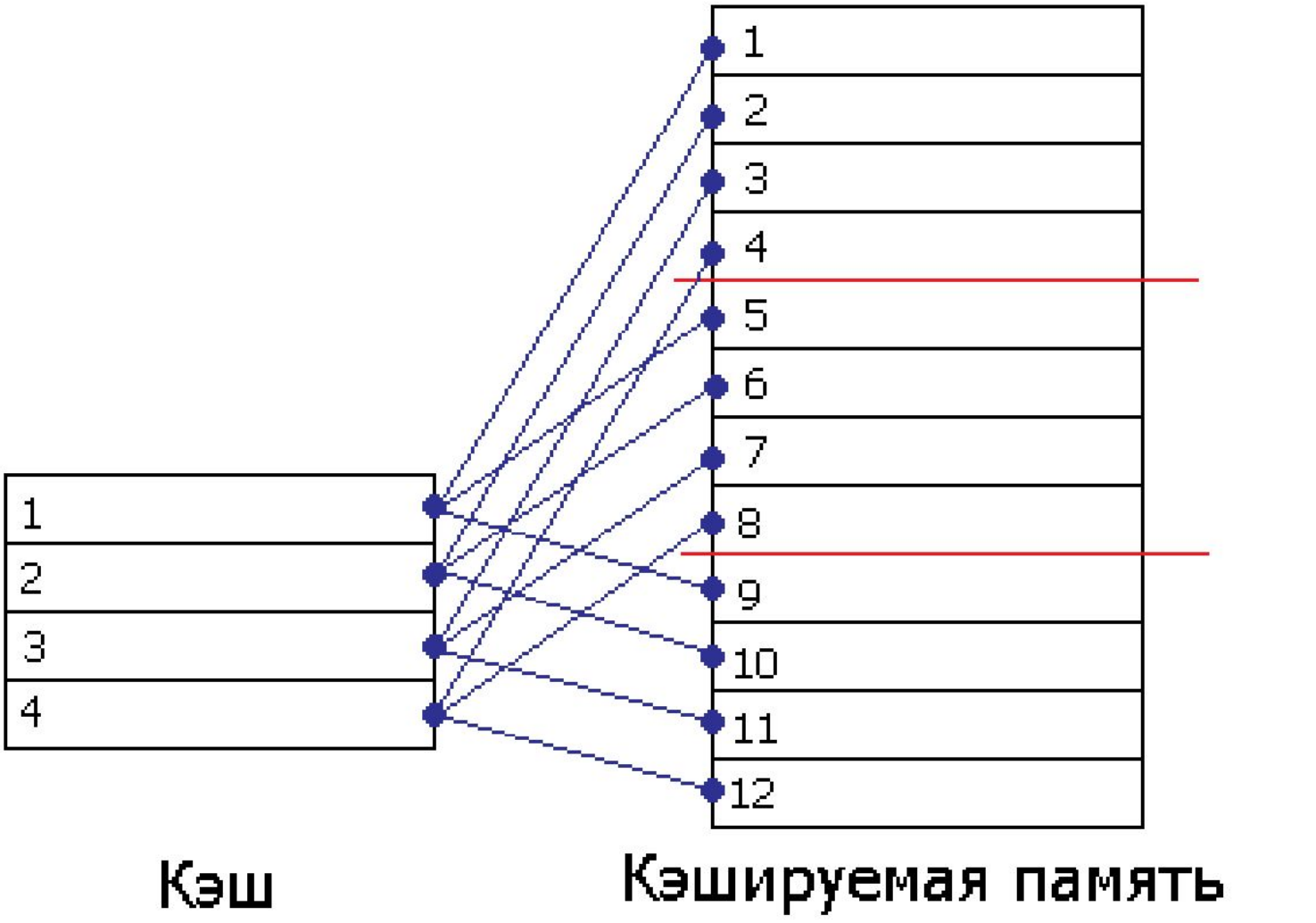
Исходные данные к рассматриваемым примерам

- Размер кэш-памяти составляет 16 Кбайт
- Обмен данными между ОП и кэш-памятью выполняется блоками размером по 4 байта каждый. В этом случае кэш организован как $4К=2^{12}$ слотов по 4 байта каждый
- Оперативная память состоит из 16 Мбайт Каждый байт адресуется 24-битным адресом (2^{24}=16 Мбайт);
- Принятый механизм адресации - байт

Архитектура кэш-памяти прямого отображения

- В кэш прямого отображения **(direct-mapping cache)** каждый блок ОП может размещаться только в одном фиксированном слоте (строке) кэша

Устройство кэша прямого отображения



Кэш прямого отображения

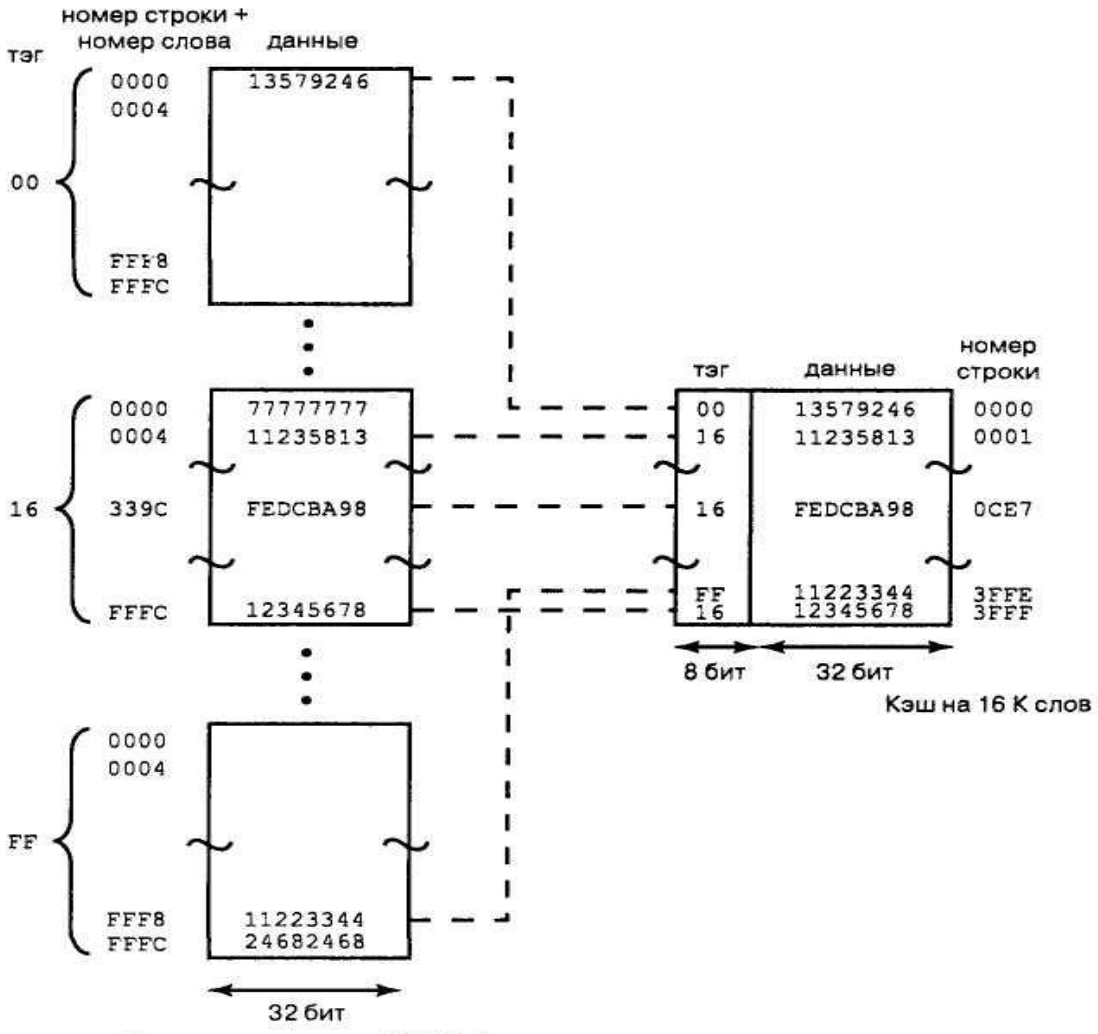
- Кэшируемая оперативная память разбивается на фреймы
- Размер каждого фрейма соответствует емкости кэш-памяти
- Предположим, что ОЗУ состоит из 1000 строк с номерами от 0 до 999, а кэш-память имеет емкость только 100 строк. В кэш-памяти с прямым отображением строки ОЗУ с номерами 0, 100, 200, ..., 900 могут сохраняться только в строке 0 КП

Кэш прямого отображения

- В рассматриваемом примере, когда размер кэш-памяти равен 16 Кбайт, а емкость ОП составляет 16 Мбайт, общее количество фреймов будет равно

  16 Мбайт/16 Кбайт=1 К
- Учитывая, что обмен данными между ОП и кэшем выполняется блоками по 4 байта, кэш память логически будет организована в виде 16 Кбайт/4 байта = 4 К слотов (линий), содержащих по 4 байта
- Аналогично будет представлен каждый фрейм ОП

Кэш прямого отображения



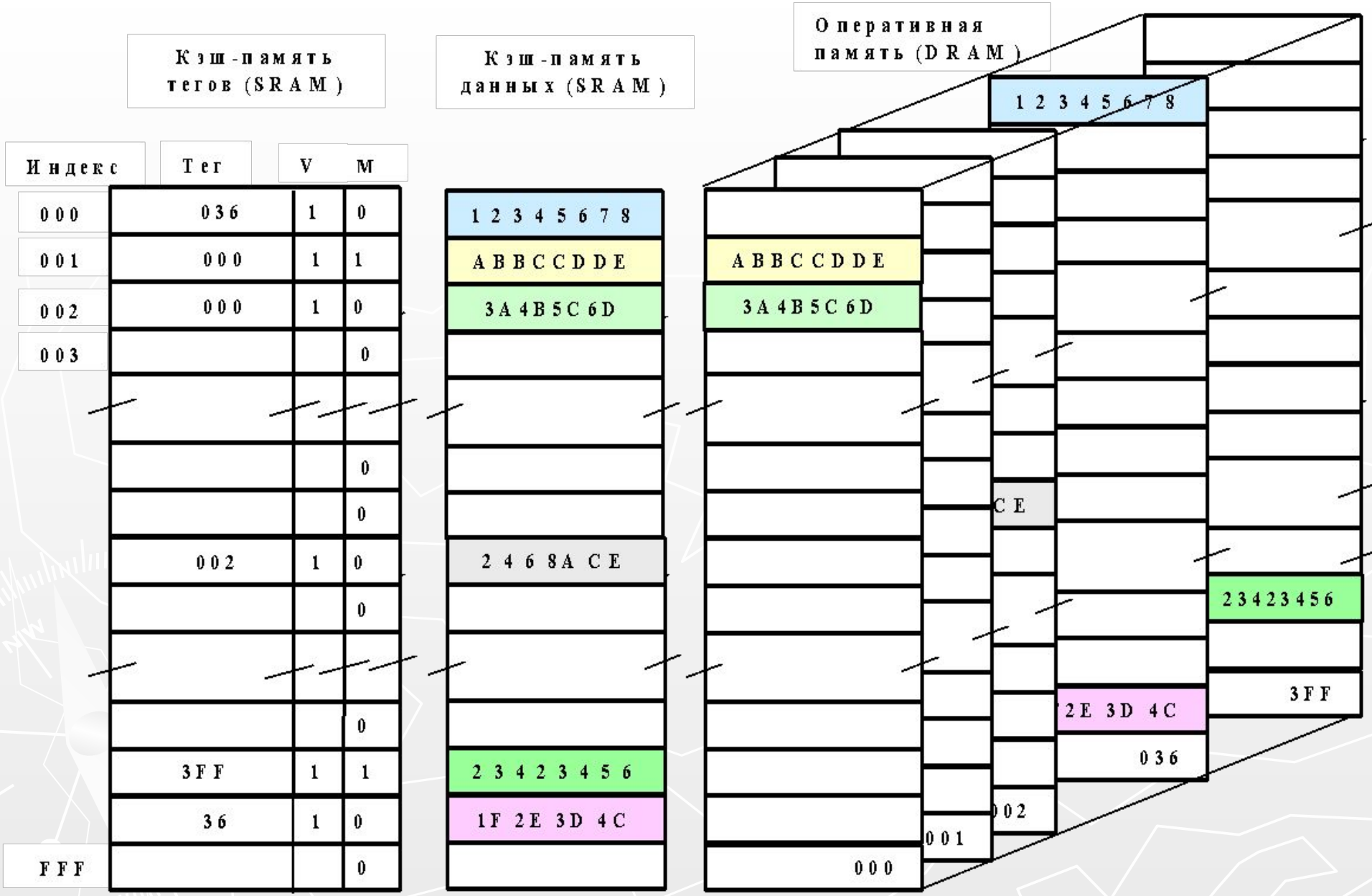
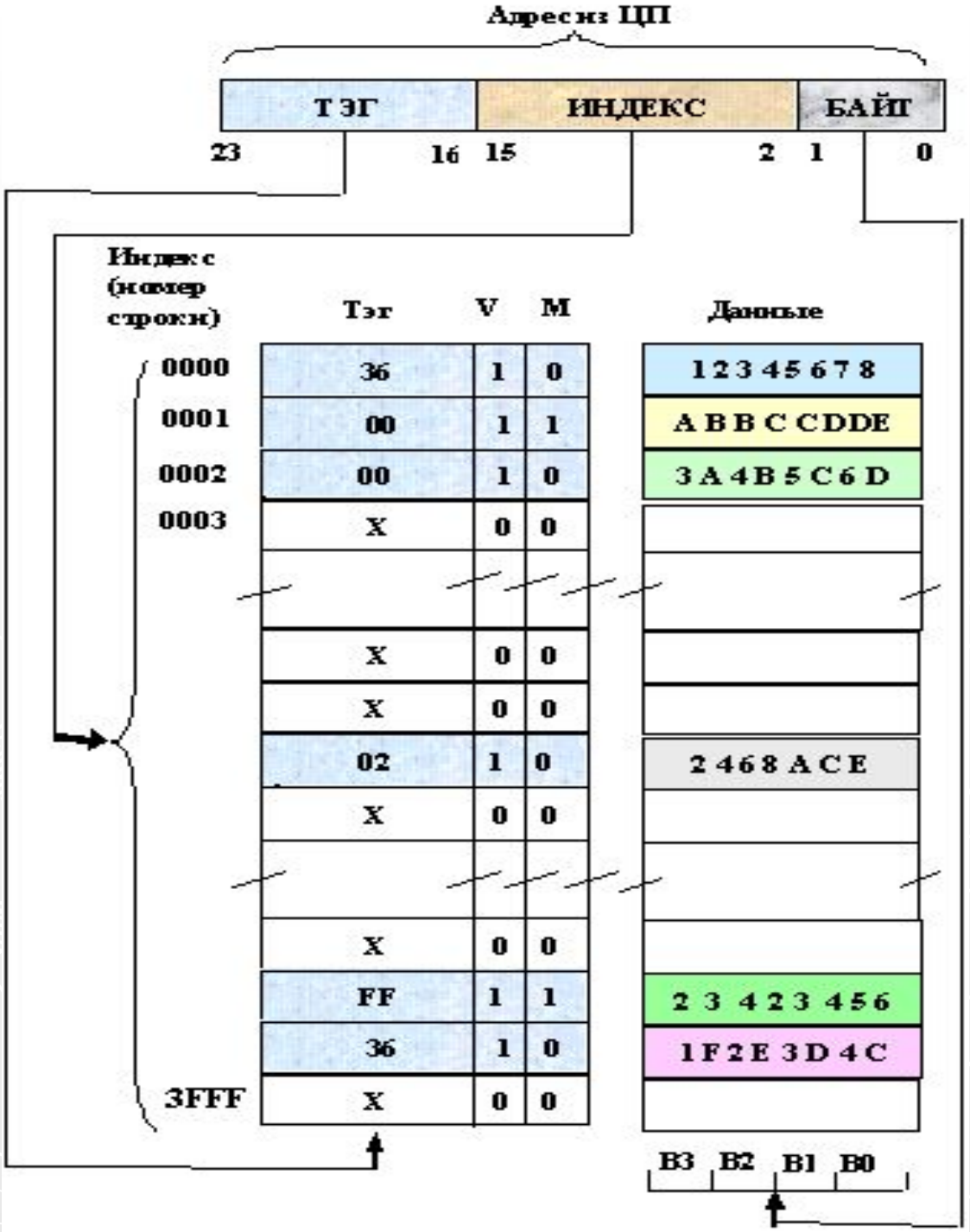

Кэш-память тегов (SRAM)
Кэш-память данных (SRAM)
Оперативная память (DRAM)
Индекс
Тег
V
M
000
036
1
0
001
000
1
1
002
000
1
0
003
0
0
0
002
1
0
0
0
3FF
1
1
36
1
0
FFF
0
1 2 3 4 5 6 7 8
A B B C C D D E
3A 4B 5C 6D
2 4 6 8A C E
2 3 4 2 3 4 5 6
1F 2E 3D 4C
A B B C C D D E
3A 4B 5C 6D
C E
2E 3D 4C
036
002
001
000
1 2 3 4 5 6 7 8
23423456
3FF

Рисунок 14.5 –


Адрес из ЦП
ТЭГ
ИНДЕКС
БАЙТ
23
16
15
2
1
0
Индекс (номер строки)
Тэг
V
M
Данные
0000
36
1
0
1 2 3 4 5 6 7 8
0001
00
1
1
A B B C C D D E
0002
00
1
0
3 A 4 B 5 C 6 D
0003
X
0
0
X
0
0
X
0
0
02
1
0
2 4 6 8 A C E
X
0
0
X
0
0
FF
1
1
2 3 4 2 3 4 5 6
36
1
0
1 F 2 E 3 D 4 C
3FFF
X
0
0
B3
B2
B1
B0

Кэш прямого отображения

- По индексу в кэш выбирается строка (слот) – 14 разрядов
- Происходит сравнение 8-разрядного тэга с содержимым области тэг, выбранной на первом шаге строке
- Если совпадают адрес-тэг и кэш-тэг, то hit, иначе – miss
- При hit загружается байт данных по содержимому поля байт

Кэш прямого отображения

- Кроме адресной части тега с каждым слотом связаны биты признаков ***действительности*** и ***модификации*** данных
- Каждый слот может быть действительным (valid), т.е. в текущий момент он достоверно отражает блок ОП, или пустым
- Для контроля когерентности данных, находящихся в слоте кэша и в блоке ОП, служит бит модификации (modified)

Кэш прямого отображения

- При обращении к памяти процессор может сформировать два типа запросов: чтение и запись
- Когда процессор генерирует запрос чтения из памяти, то сначала выполняется проверка: находится ли запрашиваемый байт данных в кэш памяти
- Если запрашиваемые процессором данные отсутствуют в кэше (***промах по чтению***), то генерируется обращение к ОП

Кэш прямого отображения

- Если произошло ***попадание по чтению***, то запрашиваемый процессором байт данных загружается в процессор
- При этом не требуется обращения к ОП, что способствует повышению производительности компьютера

Кэш прямого отображения

- Когда процессор генерирует запрос записи данных в память, то сначала происходит обращение к кэш
- Если запрашиваемый процессором блок отсутствует в кэш (***промах по записи***), то происходит обращение к ОП, запрашиваемый блок копируется из ОП в кэш и выполняется операция записи.

Кэш прямого отображения

- **Преимущества**

  - Простая схемная реализация;

  - Невысокая стоимость по сравнению с другими архитектурами КЭШей.

- **Недостатки**

  - Снижение производительности системы, когда в процессе выполнения программы процессору поочередно будут требоваться два или более блоков памяти, имеющие одинаковый индекс, но разные теги

Недостаток кэш прямого отображения

- Однако емкость КП при этом используется не в полной мере: несмотря на то, что часть кэш-памяти может быть не заполнена, будет происходить вытеснение из нее полезной информации при последовательных обращениях, например, к строкам 101, 301, 101 ОЗУ.

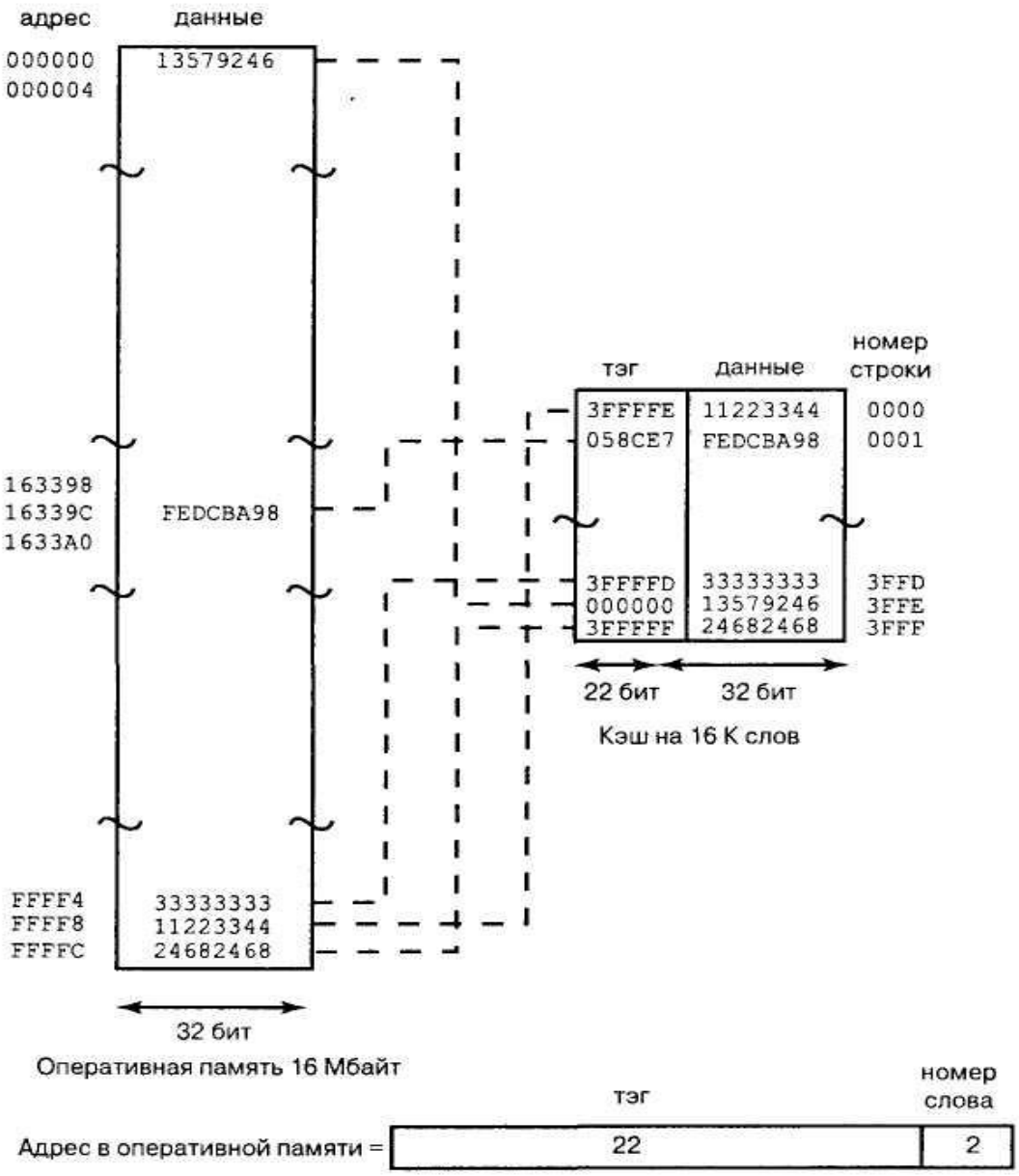
Ассоциативный кэш

- В ассоциативном кэше (Associative-mapping cache) блок данных может загружаться в любой свободный слот
- Такая организация кэша является самой гибкой по сравнению с рассмотренными типами структур

Ассоциативный кэш

- 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

Ассоциативный кэш



Ассоциативный кэш

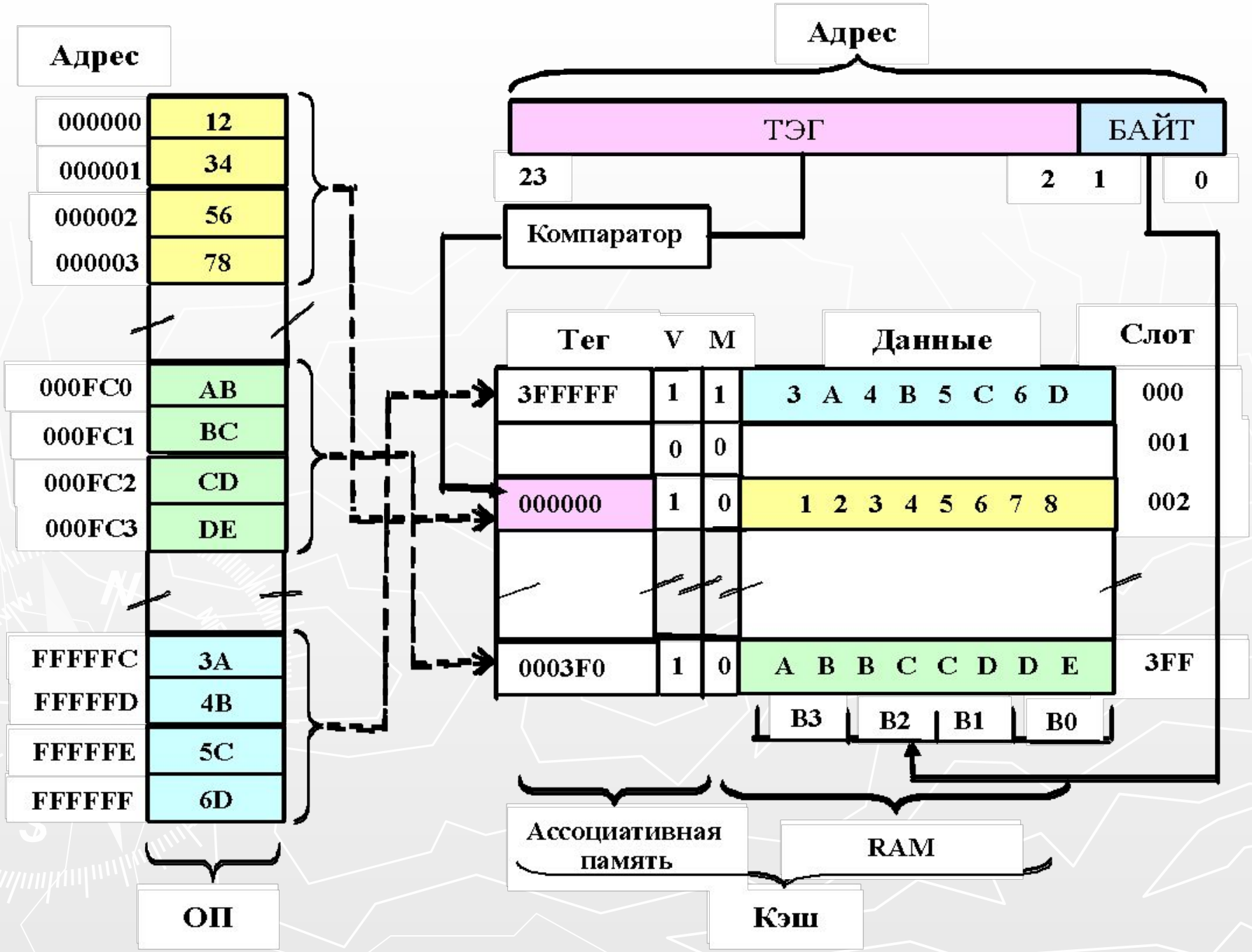
- Структурная схема ассоциативного кэша, представляет собой комбинацию ассоциативной памяти и памяти с произвольным доступом (RAM)
- В связи с тем, что каждая ячейка ассоциативной памяти значительно дороже, чем RAM ячейка, поэтому в ассоциативной памяти хранятся только адреса блоков ОП, в то время как данные могут размещаться в RAM ячейках кэша
- Это, с одной стороны, приводит к незначительному увеличению времени доступа, с другой стороны, снижает стоимость кэша

Ассоциативный кэш

- В каждой строке кэша нужно хранить помимо блока данных длиной в 4 байт (32 бит) еще и 22-разрядный код тэга этого блока
- Так 24-разрядному адресу 16339C будет соответствовать 22-разрядный код тэга 058CE7 (коды представляются в шестнадцатеричной нотации)
- Как формируется такой код, легко проследить, если воспользоваться двоичной нотацией

Таблица 1 – Формирование адреса номера строки кэш

Адрес в памяти	1	6	3	3	9	C	шестнадцатеричный
	0001	0110	0011	0011	1001	1100	Двоичный
Сдвиг вправо на 2 разряда и выбор старших 22 бит							
Тэг (старшие 22 разряда)	00	0101	1000	1100	1110	0111	Двоичный
	0	5	8	C	E	7	шестнадцатеричный


Адрес
000000
000001
000002
000003
12
34
56
78
000FC0
000FC1
000FC2
000FC3
AB
BC
CD
DE
FFFFFC
FFFFFD
FFFFFE
FFFFFF
3A
4B
5C
6D
ОП
Адрес
ТЭГ
БАЙТ
23
2 1
0
Компаратор
Тег
V
M
Данные
Слот
3FFFFF
1
1
3 A 4 B 5 C 6 D
000
0
0
001
000000
1
0
1 2 3 4 5 6 7 8
002
0003F0
1
0
A B B C C D D E
3FF
B3
B2
B1
B0
Ассоциативная память
RAM
Кэш

Ассоциативный кэш

- Когда процессор формирует адрес обращения к памяти, то старшие 22 разряда этого адреса загружаются в поле тега, а 2 младших разряда - в поле выбора байта
- Содержимое адресного поля тега параллельно сравнивается с идентификаторами блоков, представленными в ассоциативной памяти кэша
- Если значение **ТЕГ** совпадает с содержимым одного из полей ассоциативной памяти, то фиксируется ***кэш-попадание*** и производится выборка байта из соответствующей строки кэша
- Адрес байта указывается в двухразрядном поле **БАЙТ**

Ассоциативный кэш

- **Преимущество**

  - Позволяют так организовать обновление строк, что вероятность обнаружения нужного слова данных в кэше *(кэш-попадания — hit in cache)* значительно возрастает, а это значит, что значительно реже нужно обращаться за новыми данными к оперативной памяти.

- **Недостатки**

  - Сложная схемная реализация, необходимая для одновременного сравнения тегов во всех слотах кэша;
  - Высокая стоимость.

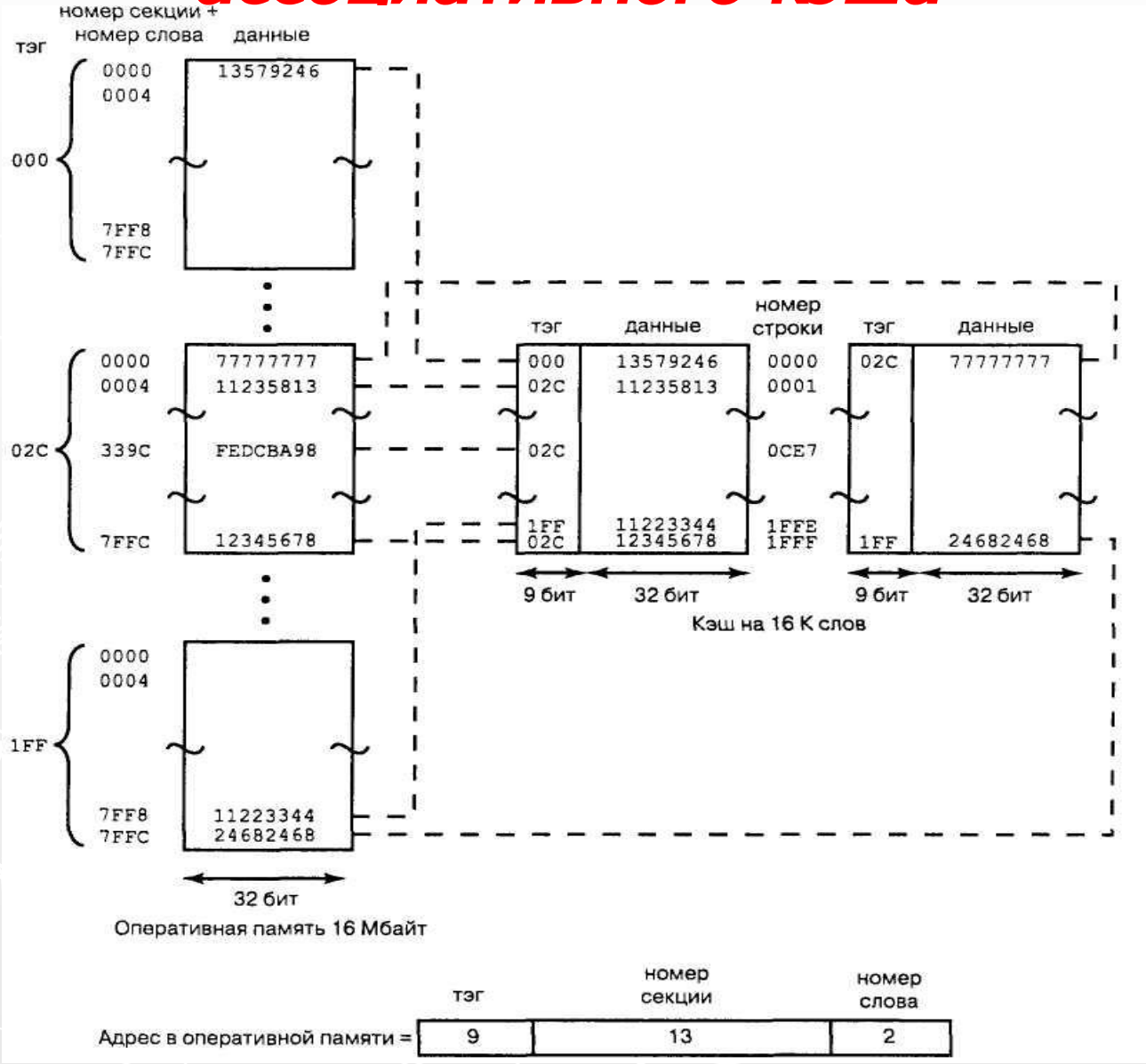
Секционированный ассоциативный кэш

- Такой кэш является дальнейшим совершенствованием кэша прямого отображения
- Решение проблемы кэша прямого отображения состоит в построении памяти, позволяющей хранить более одного блока с одинаковым индексом

Секционированный ассоциативный кэш

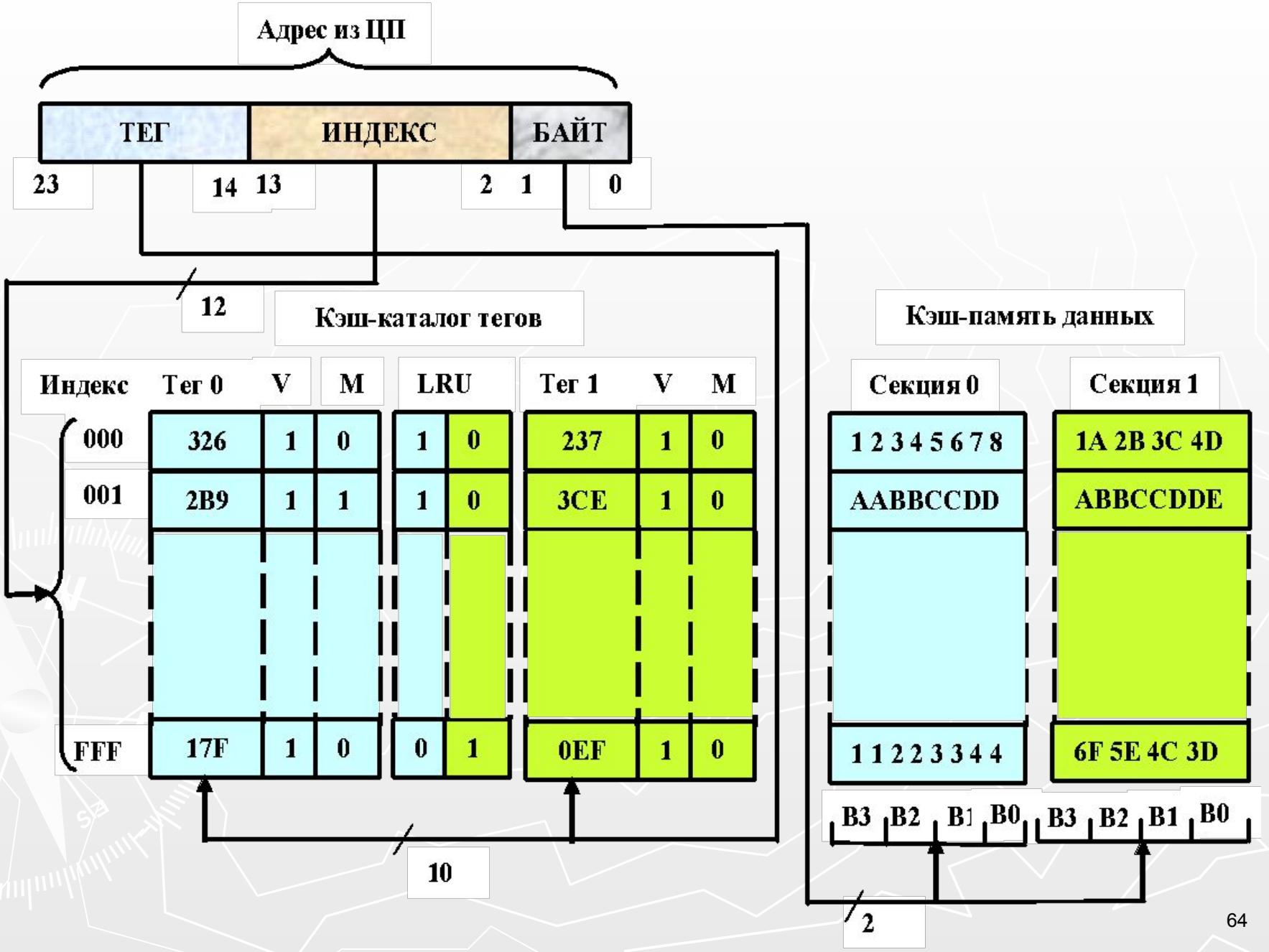
- Например, секционированный ассоциативный кэш, имеющий w наборов памяти, может хранить w блоков данных с одинаковыми индексами вместе с их тегами
- Архитектуру наборно-ассоциативного кэша можно рассматривать как несколько параллельно работающих кэш прямого отображения
- Контроллеру такого кэша приходится принимать решение о том, в какую секцию следует помещать очередной блок данных

Архитектура двухканального секционно-ассоциативного кэша



Секционированная ассоциативная функция отображения

- Для хранения данных в таком кэше используются две секции (секция **0** и секция **1**)
- Количество слотов в каждом наборе составляет 2^{12} = 4K по 4 байта в каждом слоте


Адрес из ЦП
ТЕГ
ИНДЕКС
БАЙТ
23
14
13
2
1
0
12
Кэш-каталог тегов
Кэш-память данных
Индекс
Тег 0
V
M
LRU
Тег 1
V
M
Секция 0
Секция 1
000
326
1
0
1
0
237
1
0
1 2 3 4 5 6 7 8
1A 2B 3C 4D
001
2B9
1
1
1
0
3CE
1
0
AABBCCDD
ABBCCDDE
FFF
17F
1
0
0
1
0EF
1
0
1 1 2 2 3 3 4 4
6F 5E 4C 3D
B3
B2
B1
B0
B3
B2
B1
B0
10
2

Секционированная ассоциативная функция отображения

- Такой кэш должен содержать два каталога тегов
- Каталог ***Тег 0*** используется при работе с набором **0** данных, соответственно, каталог ***Тег 1*** работает с набором 1
- Каждая строка кэш-каталога тегов имеет собственный бит ***действительности (V)*** и ***модификации (M)***

Секционированная ассоциативная функция отображения

- Для реализации алгоритма замещения используются два служебных бита, обозначенных на рисунке ***LRU - Least Recently Used***
- Кандидатом на замещение обычно выбирается слот, последнее обращение к которому было раньше
- Например, при обращении к слоту, входящему в набор 0, бит LRU этого слота набора 0 устанавливается в 1, а соответствующий бит LRU набора 1 сбрасывается в 0

Секционированная ассоциативная функция отображения

- Значение индекса в адресе обращения процессора определяет две строки в каталоге тегов
- Далее происходит параллельное сравнение индексируемых тегов для всех каналов кэша с содержимым поля ТЕГ в адресе запроса
- Если схема сравнения обнаруживает запрашиваемый адресом обращения тег, то соответствующие данные из одного из наборов памяти данных кэш перемещаются в процессор

Секционированная ассоциативная функция отображения

- В противном случае затребованный процессором блок данных загружается из ОП
- При загрузке блока данных в кэш происходит замещение содержимого одного из индексируемых слотов
- Выбор замещаемого слота реализуется с помощью алгоритма LRU

Секционированная ассоциативная функция отображения

- Замещаемый блок из кэша записывается в ОП, если он был модифицирован в течении времени пребывания в кэш (если бит М=1)
- Если в замещаемом блоке бит М=0, то новый блок из ОП замещает старый без перезаписи в ОП, сокращая тем самым частоту копирования блоков из кэш-памяти в ОП

Секционированная ассоциативная функция отображения

- Если ЦП сформировал запрос по записи, то в случае кэш-попадания запись данных выполняется только в кэш при реализации ***стратегии обратной записи***
- При этом бит модификации соответствующего слота в наборе устанавливается в 1
- Модифицированный блок кэш-памяти будет записан в ОП только когда он будет замещаться

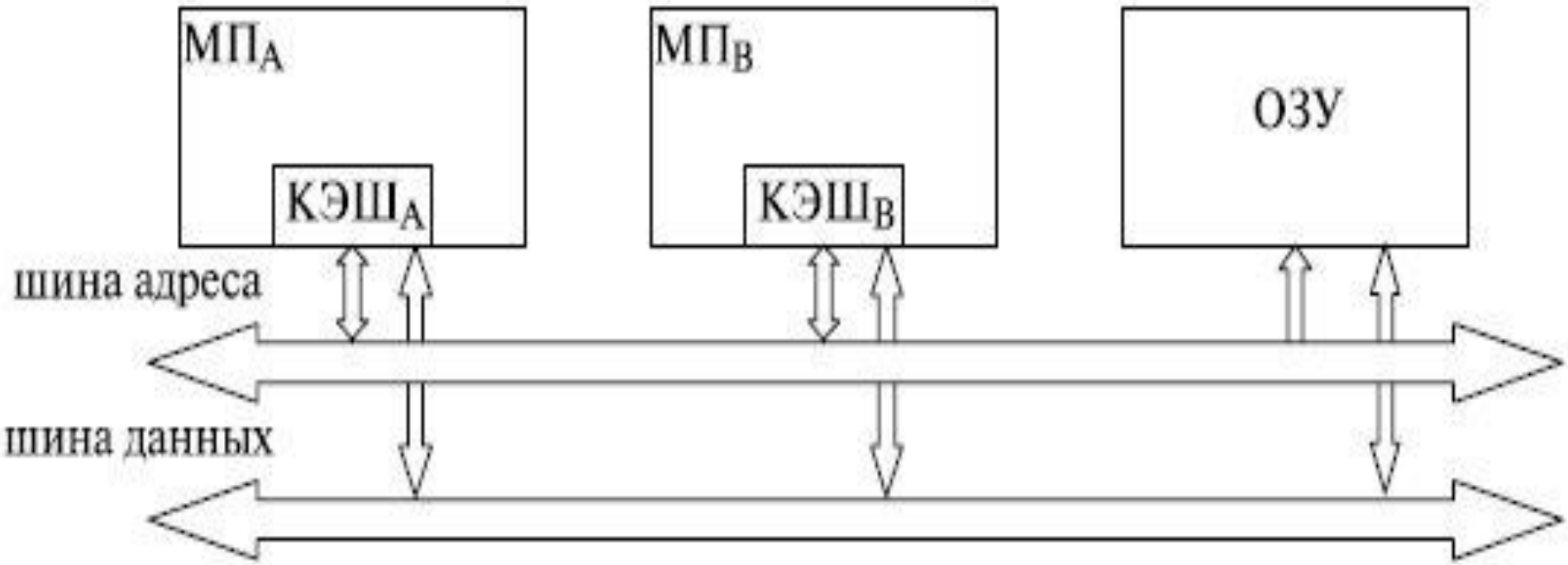
Секционированный ассоциативный кэш

- Секционированная ассоциативная кэш-память широко используется в современных процессорах
- Так в процессорах Intel P6 нашла применение четырех канальная секционно - ассоциативная кэш-память (уровень L1) для хранения команд
- Кэш-память данных имеет архитектуру двух - канального секционно - ассоциативного кэша

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 мультипроцессорной системы с общей оперативной памятью



- Предположим, что МП А загрузил некоторую строку данных из ОЗУ в свою внутреннюю КП и изменил данные в этой строке в процессе работы
- Существует два основных механизма обновления оперативной памяти:

- **сквозная запись**, которая подразумевает, что как только изменилась информация во внутренней кэш-памяти, эта же информация копируется в то же место оперативной памяти
- **обратная запись**, при которой микропроцессор после изменения информации во внутреннем кэше отражает это изменение в оперативной памяти не сразу, а лишь в тот момент, когда происходит вытеснение данной строки из кэш-памяти в оперативную. То есть существуют определенные моменты времени, когда информация, предположим, по адресу 2000 имеет разные значения: микропроцессор ее обновил, а в оперативной памяти осталось старое значение. Если в этот момент другой микропроцессор (МП В), использующий ту же оперативную память, обратится по адресу 2000 в ОЗУ, то он прочитает оттуда старую информацию, которая к этому времени уже не актуальна.

- 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);
- **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.

- 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

Протокол MESI

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

1 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;

Протокол MESI

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

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

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

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

- 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**
- 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).

Выводы

- В лекции рассмотрены общие принципы функционирования кэш-памяти микропроцессора, организация кэш-памяти с прямым отображением, полностью ассоциативной и множественно-ассоциативной КП
- Рассмотрены основные механизмы обновления оперативной памяти: кэширование со сквозной и с обратной записью
- Представлена организация внутренней кэш-памяти микропроцессора
- Разобраны способы обеспечения согласованности кэш-памяти микропроцессоров в мультипроцессорных системах