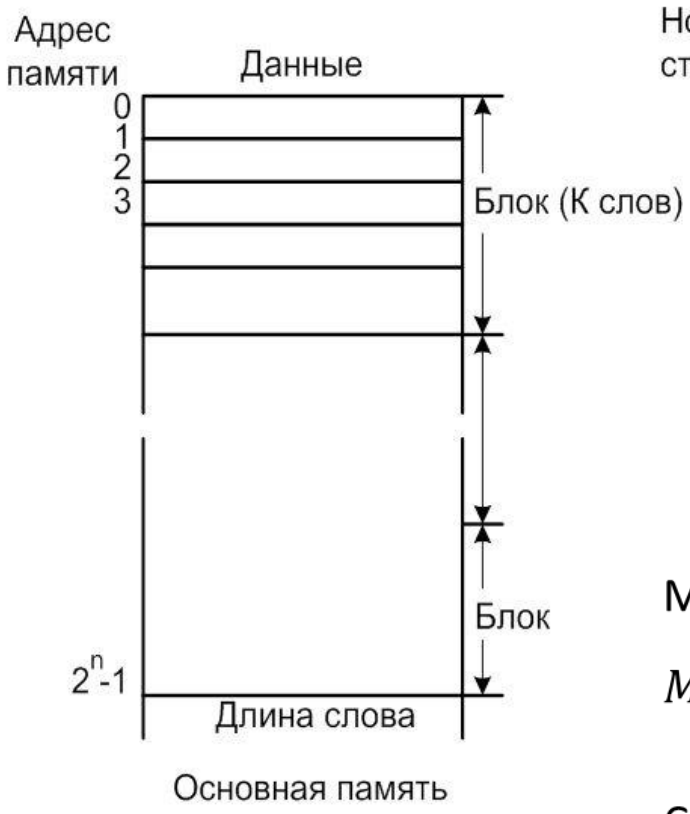
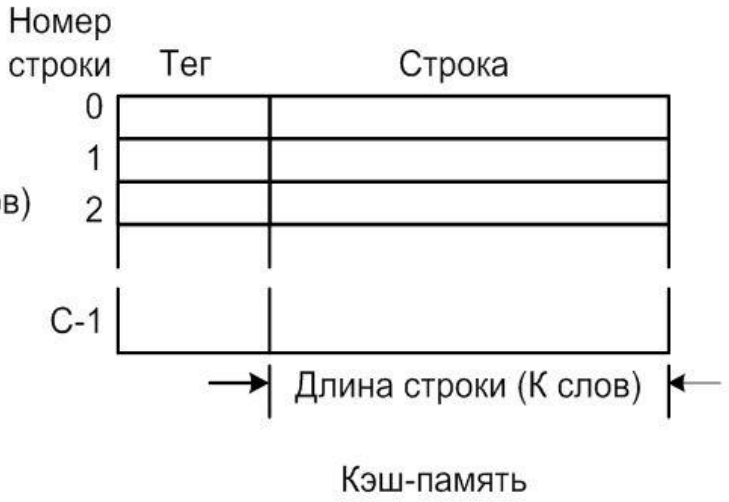
Идея предложено Уилксом в 1965 году при разработке ЭВМ Atlas. Автор назвал такую память подчиненной (slave). Позже стал использоваться термин cache, так как эта память скрыта от программиста.
Кэш-попадание - обнаружение слова в кэш памяти.
Кэш-промах - отсутствия слова кэш-памяти. В этом случае слово извлекается из основной памяти и одновременно в кэш пересылается блок данных, содержащий это слово.

Структура ЭВМ с кэш-памятью.





M – число блоков в основной памяти.

$$M = \frac{2^n}{K}$$

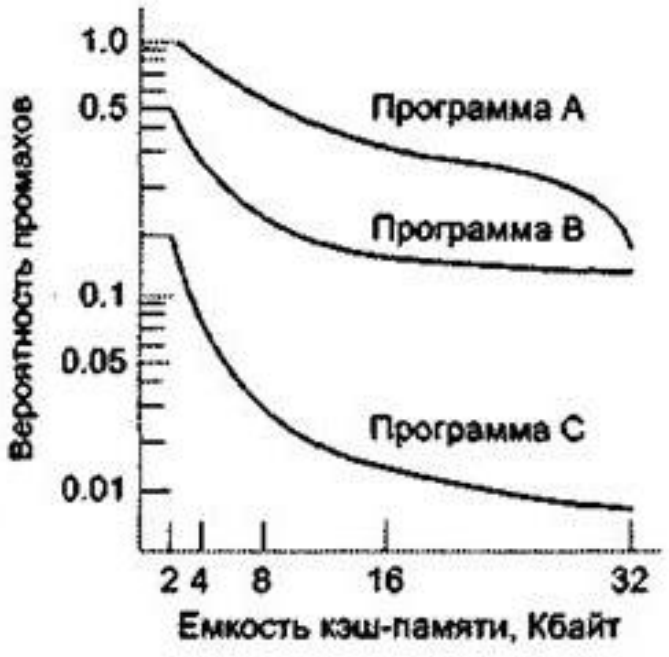
C << M – число строк в кэш-памяти.

Тег (признак) – содержит сведения о том, копия какого блока основной памяти в данный момент храниться в данной строке.

Факторы, влияющие на эффективность использования кэш-памяти:

- Ёмкость кэш-памяти;
- Размер строки;
- Способ отображения основной памяти на кэш;
- Алгоритм замещения информации в заполненной кэш-памяти;
- Алгоритм согласования содержимого основной и кэш-памяти;
- Число уровней кэш-памяти.

Зависимость вероятности промаха от ёмкости кэш-памяти.



Для большинства задач оптимальной является емкость кэш-памяти от 1 до 512 Кбайт.

Зависимость вероятности промаха от размера строки.

Вероятность промаха начинает расти когда размер строки становится излишне большим из-за:

- Большие размеры строки уменьшают число строк кэша, что приводит к необходимости часто и их смены;

- По мере увеличения размера строки каждое дополнительное слово оказывается дальше от запрашиваемого, поэтому такое дополнительное слово менее вероятно понадобиться в ближайшем будущем.

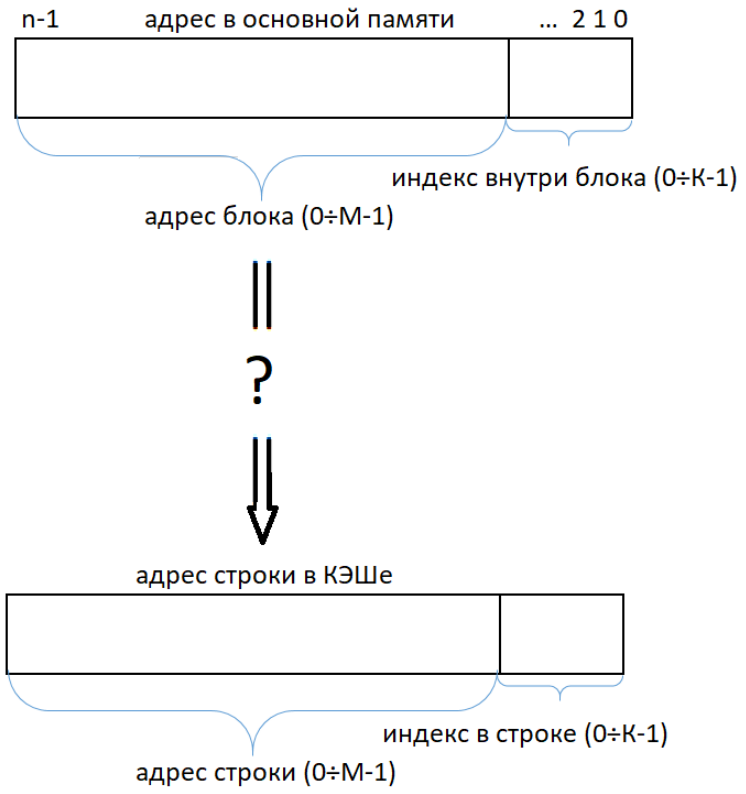
На практике размер строки обычно выбирают равным ширине шины данных, связывающей кэш-память с основной памятью.

Способы отображения оперативной памяти на кэш память.

Каким образом адрес блока данных в основной памяти, преобразуется в адрес строки в кэш-памяти?

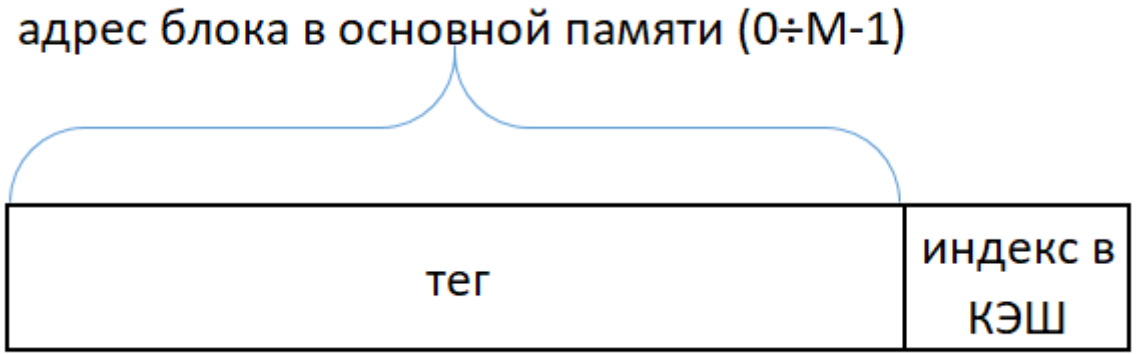
Известны три способа:

- Прямое отображение;
- полностью ассоциативное отображение;
- Множественно-ассоциативное отображение;



Прямое отображение.

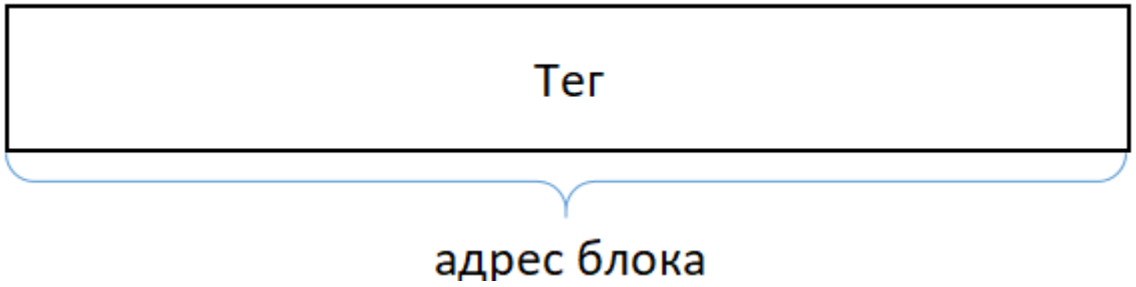
При прямом отображении адрес блока (0÷M-1) разбивается на две части:
- Поле тега – хранящее старшие разряды адресов в основной памяти;
- Поле индекса – определяющие номер строки в кэш-памяти.



Достоинства – простая реализация.
Недостаток – невысокая вероятность попадания.

Полностью ассоциативное отображение.

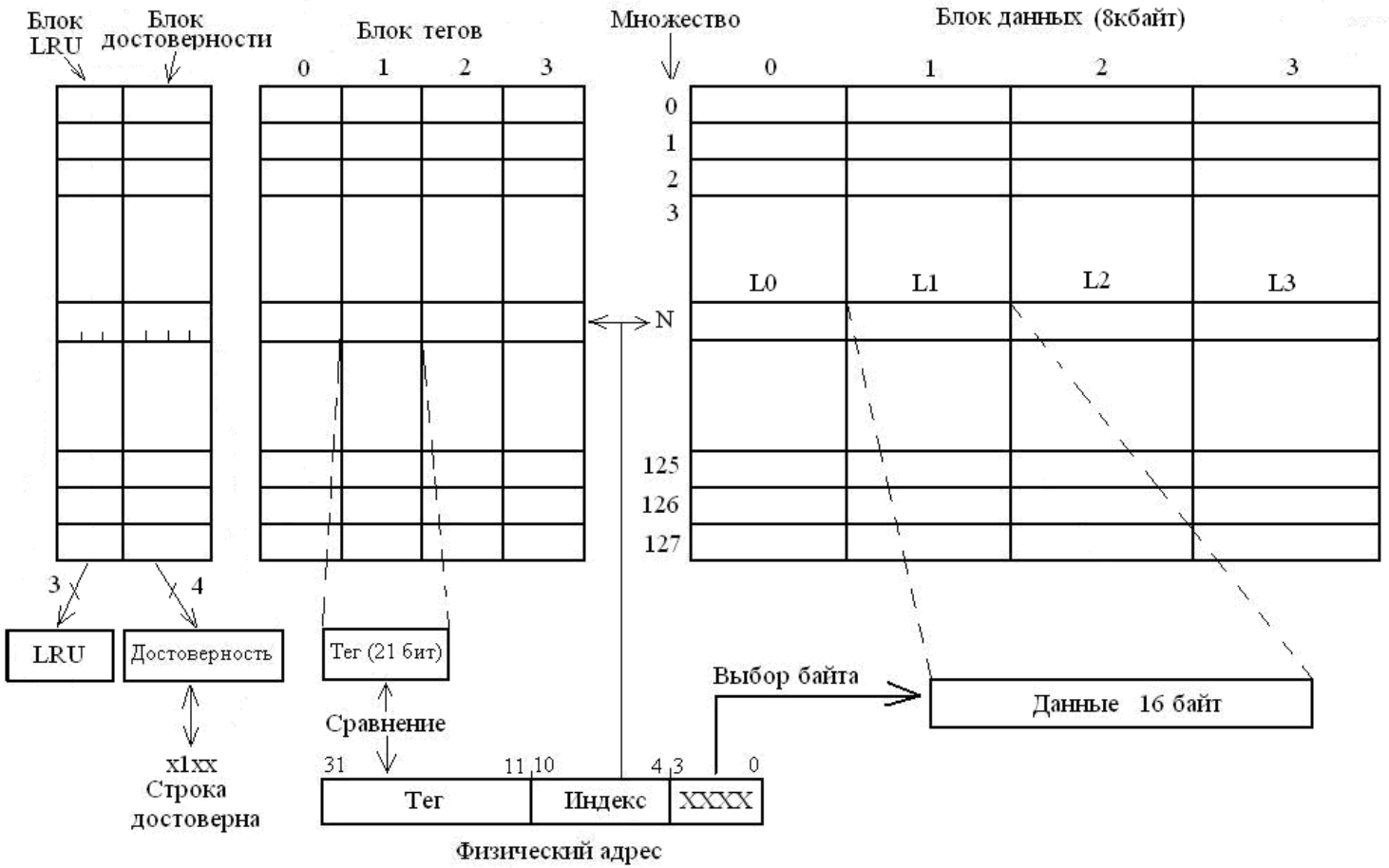
Любой блок основной памяти может находиться в любой строке кэш-памяти. В этом случае адрес блока основной памяти является тегом, а поле индекса отсутствует.
Для реализации необходима дорогостоящая ассоциативная память.



Множественно ассоциативный отображение.

Представляет собой компромиссное решение, сочетающее достоинства прямого и ассоциативного способов отображения.
В процессе Intel 486 используется совмещённый кэш для команд и данных ёмкостью 8 Кбайт = 4 × 128 × 16 байт.

Структура кеш-памяти 486 процессора.

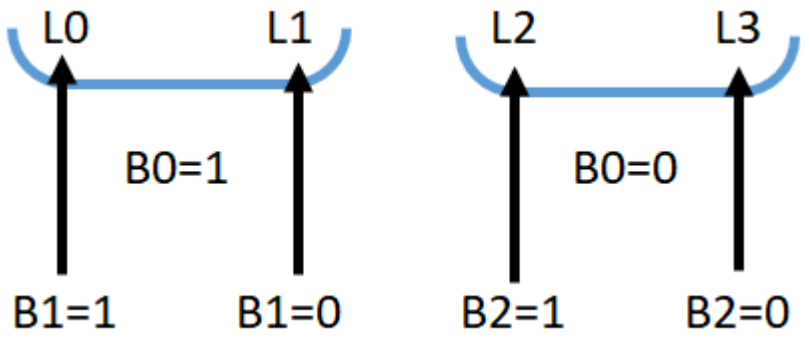


Алгоритмы замещения информации в заполненной кэш-памяти.

1. На основе наиболее давнего использования (LRU – least Recently Used). Замещается та строка в кэш-памяти, к которой дольше всего не было обращений.
2. Алгоритм работающий по принципу FIFO. Замещается строка дольше всего находившееся в кэше.
3. На основе наименее часто использовавшейся строки (LFU – Least Frequently Used). Заменяется та строка, к которой было меньше всего обращений.
4. Произвольный выбор строки для замены. Заменяемая строка выбирается случайным образом.

Алгоритм LRU.

Пусть L0,L1,L2,L3 – строки в кэш-памяти.
Каждому множеству в блоке LRU соответствует 3 бита B0,B1,B2, которые модифицируются при каждом попадании следующим образом.



Выбор заменяемой строки определяется содержим B0, B1, B2.

0	0	X – заменяется L0
0	1	X – заменяется L1
1	X	0 – заменяется L2
1	X	1 – заменяется L3.

Алгоритм согласования содержимого кэш-памяти и основной памяти.

Причины несоответствия:

1. Изменение информации в кэш памяти процессором.
2. изменение содержимого основной памяти устройствами ввода/вывода.

Для устранения первой причины используются методы:

- Метод сквозной записи (write through).
- Метод обратной записи (write back).

По методу сквозной записи, прежде всего обновляется слово, хранящиеся в основной памяти. Если в кэше есть копия, она также обновляется. Если в кэше нет копии, то либо из основной памяти в кэш пересылается блок, содержащий измененное слово (сквозная запись с отображением), либо нет (сквозная запись без отображения).

Существует модификация метода, известная как метод буферизованной сквозной записи.

В соответствии с методом обратной записи слово заносится только в кэш. Если строки в кэше нет, то соответствующий блок пересылается сначала из основной памяти в кэш. При замещении строки, она пересылается основную память.

Смешанные и разделенной кэш-память.

Если кэш используется и для команд и для данных, то она называется смешанной, а соответствующая архитектура – Принстонской.
Если кэш разделён на две части - кэш команд и кэш данных, то такая кэш-память называется разделенной, а соответствующая архитектура - Гарвардской.
Достоинства смешанной кэш-памяти является более высокая вероятность попаданий, поскольку в ней оптимальный баланс между командами и данными устанавливается автоматически.
Достоинством разделенного кэша является возможность одновременной выборки команд и данных, что особенно важно для машин с конвейерной обработкой.

Одноуровневая и многоуровневая кэш память.

Ёмкость кэш памяти L1 не превышает 64 Кбайт. Попытки увеличения ёмкости обычно приводят к снижению быстродействия.
Поэтому общую ёмкость кэш-памяти увеличивают за счет второй (внешней) кэш памяти L2. ёмкость L2 обычно на порядок больше чем у L1, а быстродействие и стоимость несколько ниже.
Типичная ёмкость L2 - 256 – 512 кбайт и реализуется она, как правило, в виде отдельной микросхемы.

$T_{\text{дост } L1} = T_{L1h} + K_{L1m}T_{L1m}$, где
T_{L1h} – время обращения при попадании.
T_{L1m} – потери на промах.
T_{L1m} – коэффициент промахов.
$T_{\text{дост } L1}$ – время доступа к одноуровневые кэш-памяти.
$T_{\text{дост } L2} = T_{L1h} + (K_{L1m}T_{L2h}) + (K_{L2m}T_{L2m})$, где
$T_{\text{дост } L2}$ – время доступа к двухуровневый кэш памяти.

Для ускорения обмена информацией между центральным процессором и L2 вводят специальную шину, называемую шиной заднего плана. Шина переднего плана связывает центральный процессор с основной памятью.

Кэш процессора Pentium III.

Кэш L2 может быть реализован вне процессора, как в версии Katmoi. Он имеет размер 512 Кбайт, реализован на основе SRAM. Имеет 4-х канальную организацию. Может использовать протоколы сквозной и обратной записи. Ширина шины составляет 64 бита.

В процессоре Pentium III CooperMine L2 интегрирован в кристалл процессора. Имеет размер 256 Кбайт и 8-канальную организацию. Так как он внутри кристалла, то соединён с процессором 256 разрядной шиной.

Кэш процессора Pentium IV.

У процессора Pentium IV может быть до трёх уровней кэш памяти. Кэш первого уровня состоит из отдельных кэша команд и кэша данных. Кэш данных размером 8 Кбайт имеет 4 канальную множественно-ассоциативную организацию. Размер блока составляет 64 байта. Для записи данных в кэш применяется протокол сквозной записи. Целочисленные данные извлекаются из кэша за 2 такта. Pentium IV могут работать на тактовой частоте свыше 1.3 ГГц. Для доступа к данным требуется менее 2 нс. В Кэше команд содержатся не обычные машинные команды, а их декодированные версии.

Кэш 2-го уровня имеет ёмкость 256 Кбайт. Имеет 8-канальный множественно-ассоциативный организацию. Размер блока составляет 128 байт. Для записи данных используется протокол обратной записи. Время доступа составляет 7 тактов.

Оба кэша и L1 и L2 реализованы на микросхеме процессора. Архитектура Pentium IV позволяет добавить в микросхему и кэш 3-уровня. Однако он используется только в процессорах для серверных систем.