

Организация памяти

Иерархии памяти

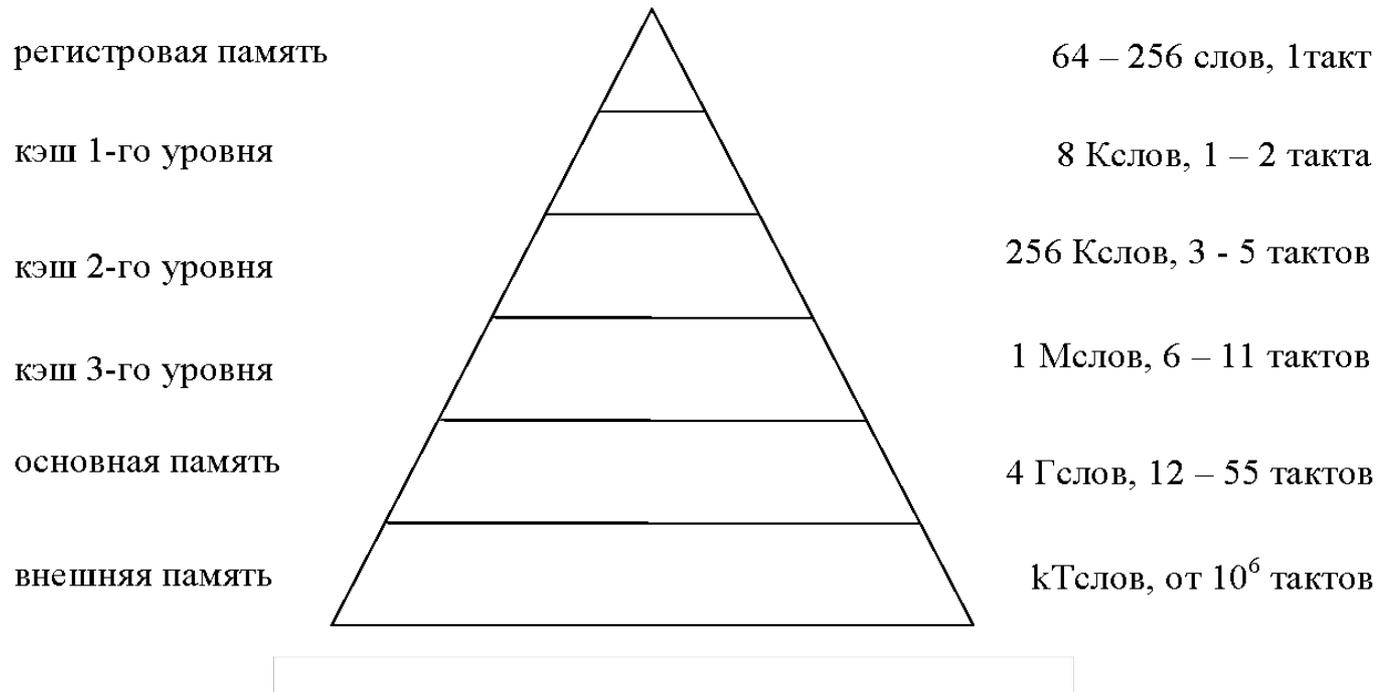
Идея **иерархической (многоуровневой)** организации памяти заключается в использовании на одном компьютере нескольких уровней памяти, которые характеризуются разным временем доступа к памяти и объемом памяти. (Время доступа к памяти это время между операциями чтения/записи, которые выполняются по случайным адресам.) Основой для иерархической организации памяти служит принцип **локальности ссылок во времени и в пространстве.**

- **Локальность во времени** состоит в том, что процессор многократно использует одни и те же команды и данные.
- **Локальность в пространстве** состоит в том, что если программе нужен доступ к слову с адресом A , то скорее всего, следующие ссылки будут к адресам, расположенным по близости с адресом A .

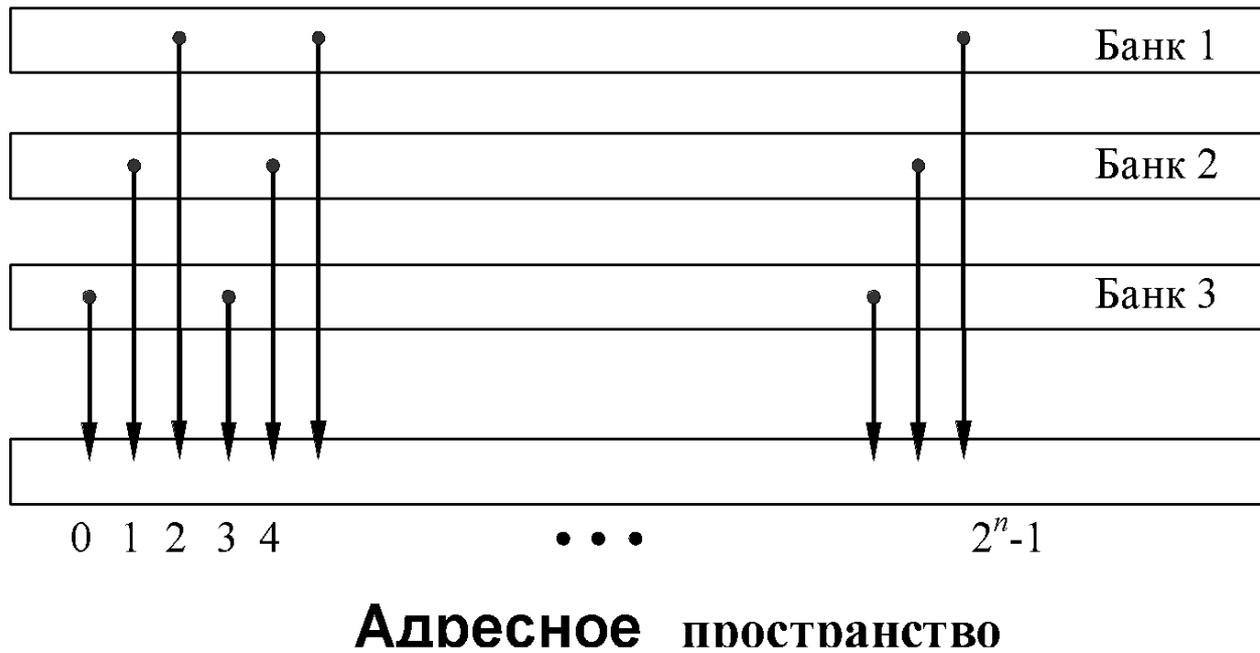
Из свойства локальности ссылок следует, что в типичном вычислении обращения к памяти концентрируются вокруг небольшой области адресного пространства и более того, выборка идет по последовательным адресам. Время доступа к иерархически организованной памяти уменьшается благодаря следующему

- **сокращению количества обращений к оперативной памяти**
- **совмещению обработки текущего фрагмента программы и пересылки данных из основной памяти в буферную память.**

Схема иерархического построения памяти



Интерливинг



Организация кэш-памяти

Кэш-память это высокоскоростная память небольшого размера с прямым доступом. Она предназначена для временного хранения фрагментов кода и данных. Кэш-память охватывает все адресное пространство памяти, но в отличие от оперативной памяти, она не адресуема и невидима для программиста.

Схема построения кэш-памяти

- Кэш-память построена на принципе **локальности ссылок во времени и в пространстве**.
- Кэш-контроллер загружает копии программного кода и данных из ОП в кэш-память блоками, равными размеру строки за один цикл чтения. Процессор читает из кэш-памяти по словам.
- Кэш-контроллер перехватывает запросы процессора к основной памяти и проверяет, есть ли действительная копия информации в кэш-памяти.

Структура кэш-памяти

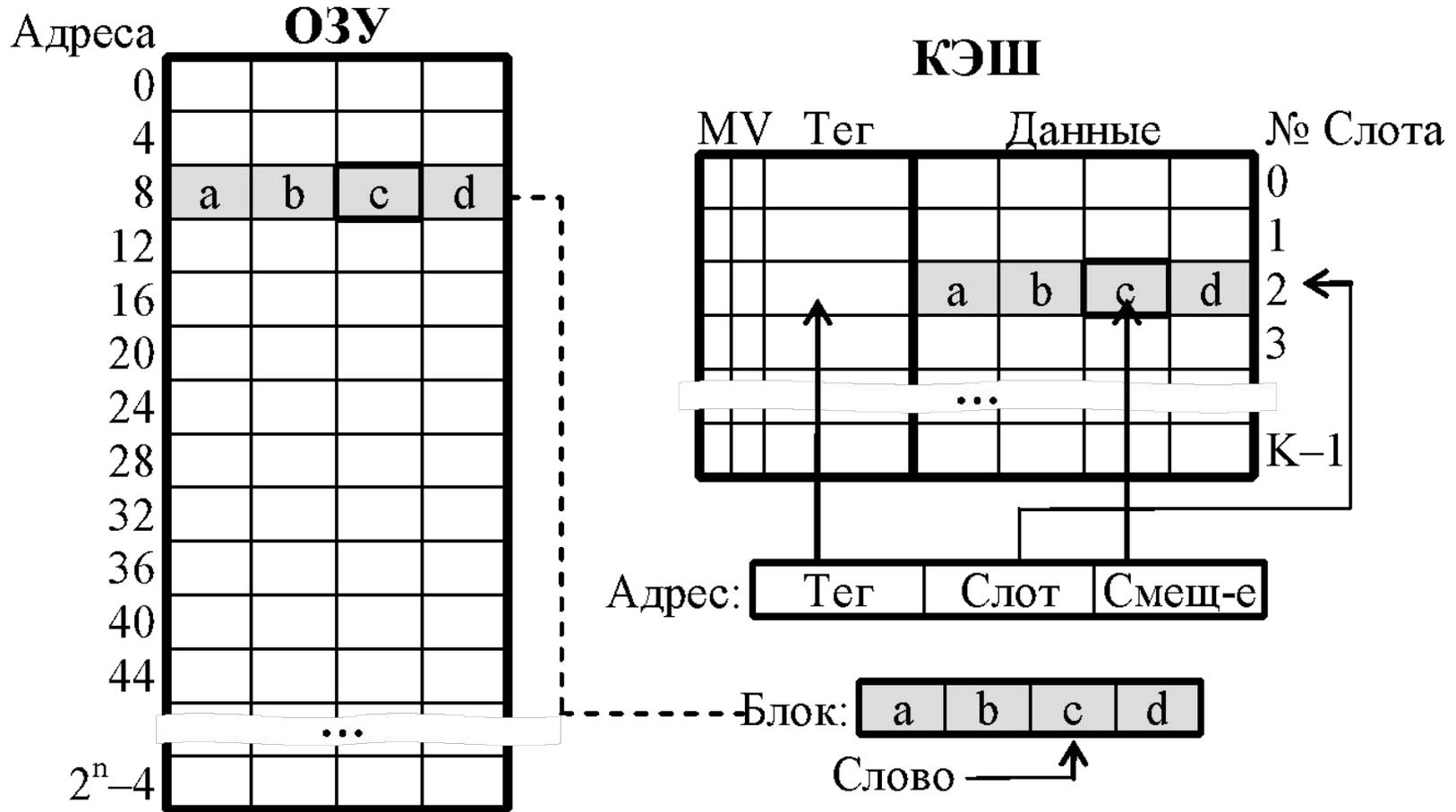
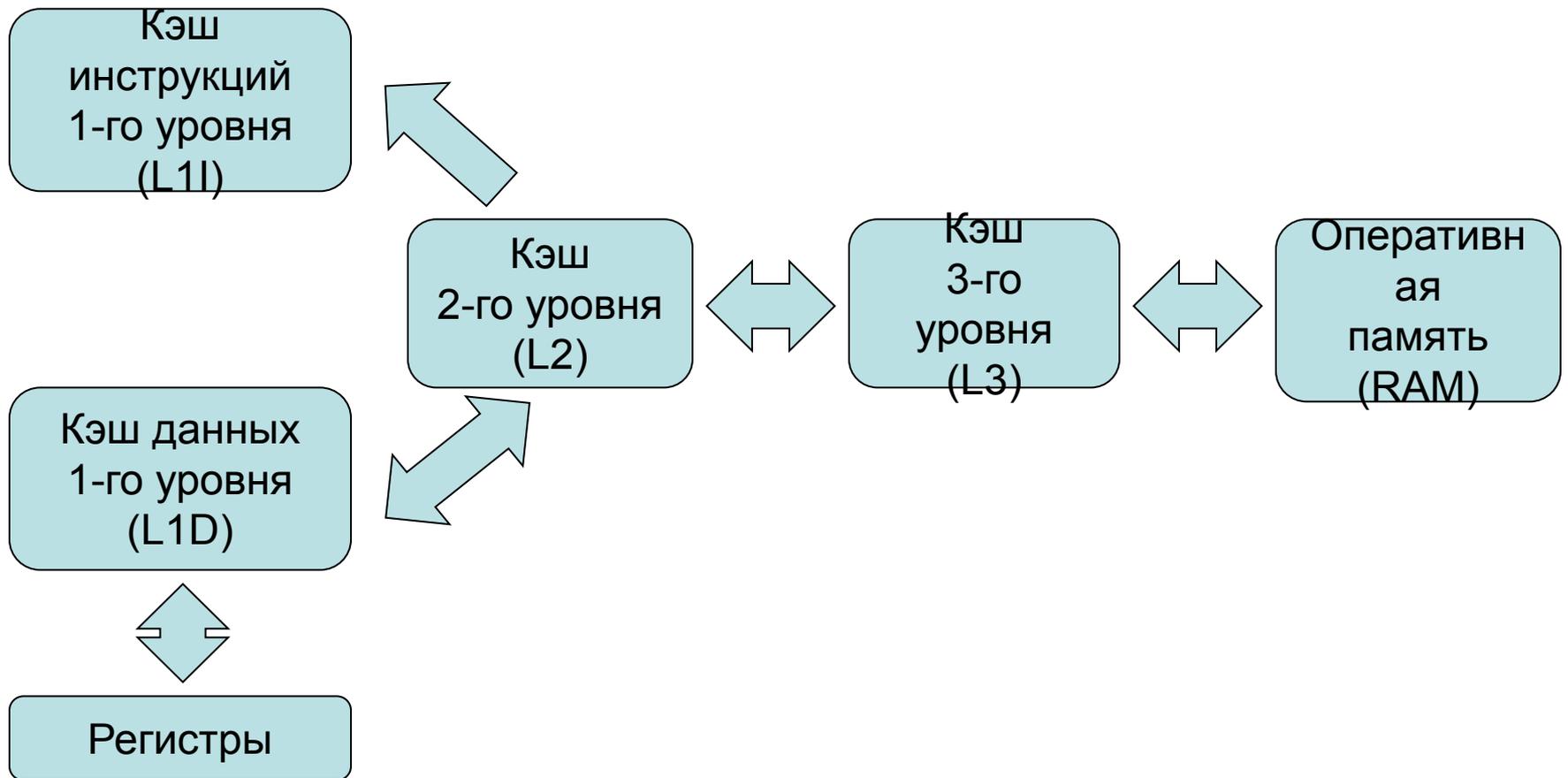


Схема иерархического построения памяти



Организация кэш-памяти

Когда контроллер помещает данные в кэш-память?

- Загрузка по требованию (**on demand**).
- Спекулятивная загрузка (**speculative load**). Алгоритм предполагает помещать данные в кэш-память задолго до того, как к ним произойдет реальное обращение. У кэш-контроллера есть несколько алгоритмов, которые указывают, какие ячейки памяти потребуются процессору в ближайшее время.

Организация кэш-памяти

Когда контроллер выполняет поиск данных в памяти?

- после фиксации промаха (*сквозной просмотр*).
- одновременно с поиском блока в кэш-памяти, в случае кэш-попадания, обращение к оперативной памяти прерывается (*отложенный просмотр*).

Основные вопросы организации кэш-памяти

- Алгоритм отображения адресов основной памяти в кэш-память.
- Алгоритм записи данных и команд из кэш-памяти в основную память.
- Алгоритм замещения строки в кэш-памяти.
- Размер кэш-памяти.
- Длина строки в кэш-памяти.

Алгоритмы отображения

- Прямой (**direct mapping**).
- Ассоциативный (**full associative mapping**).
- Множественно-ассоциативный (**set-associative mapping**).

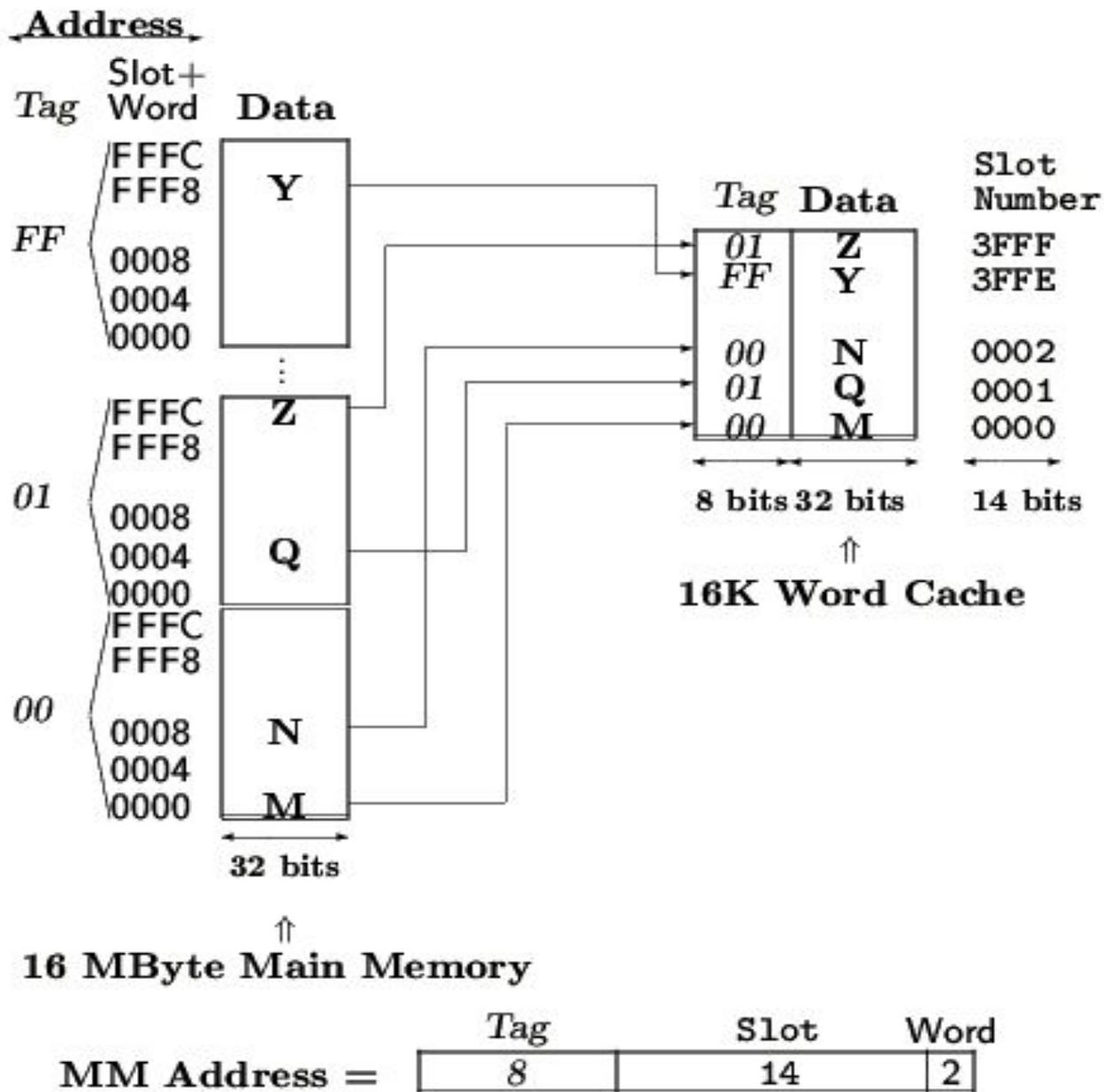


Figure 2: Direct Mapping

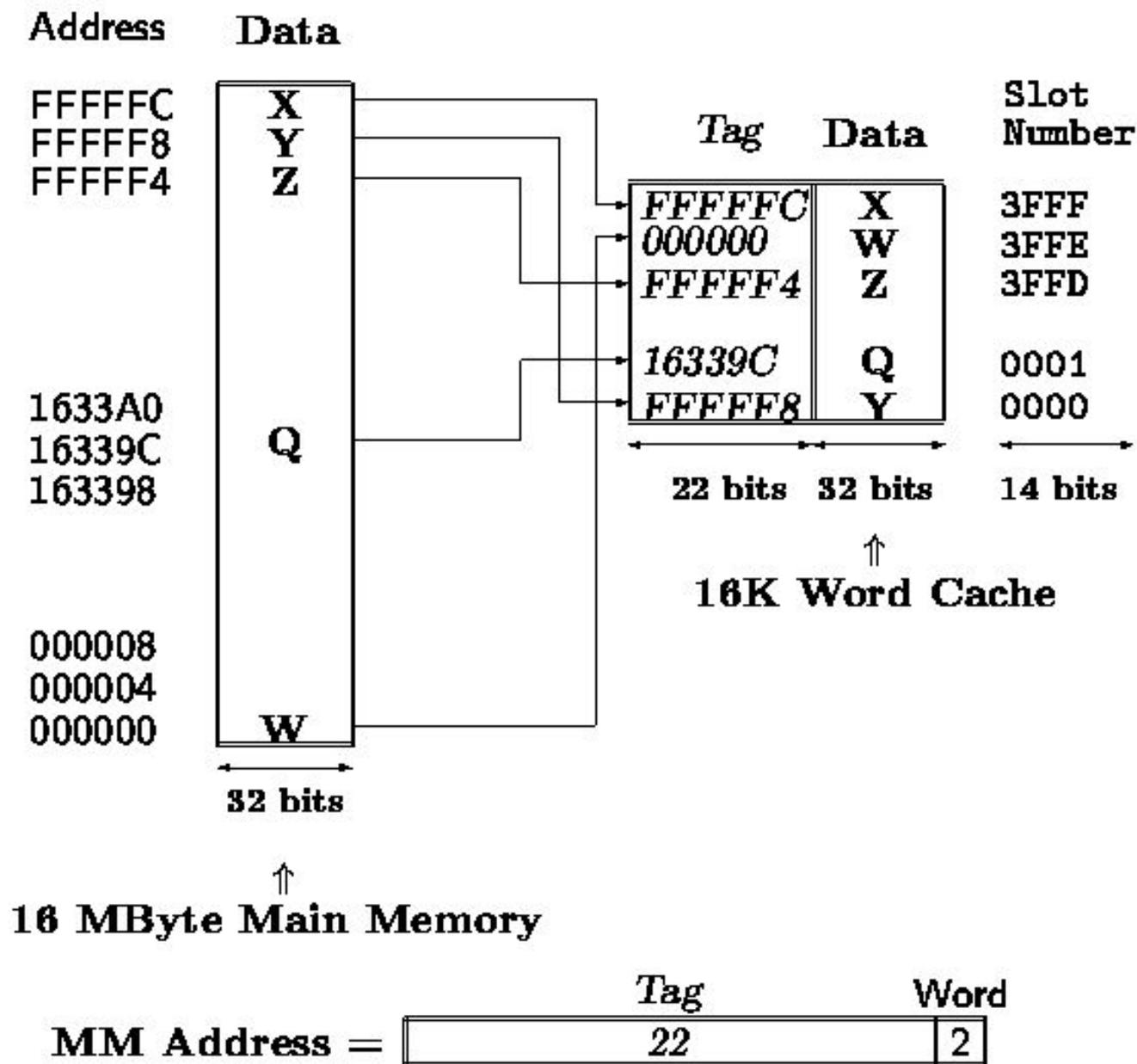
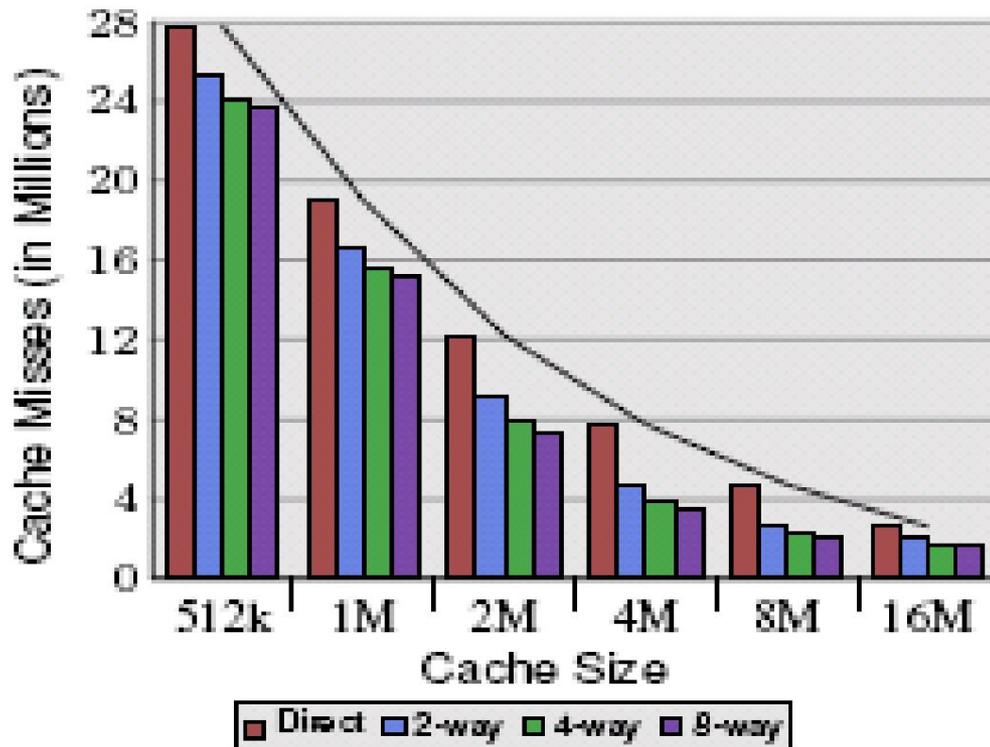
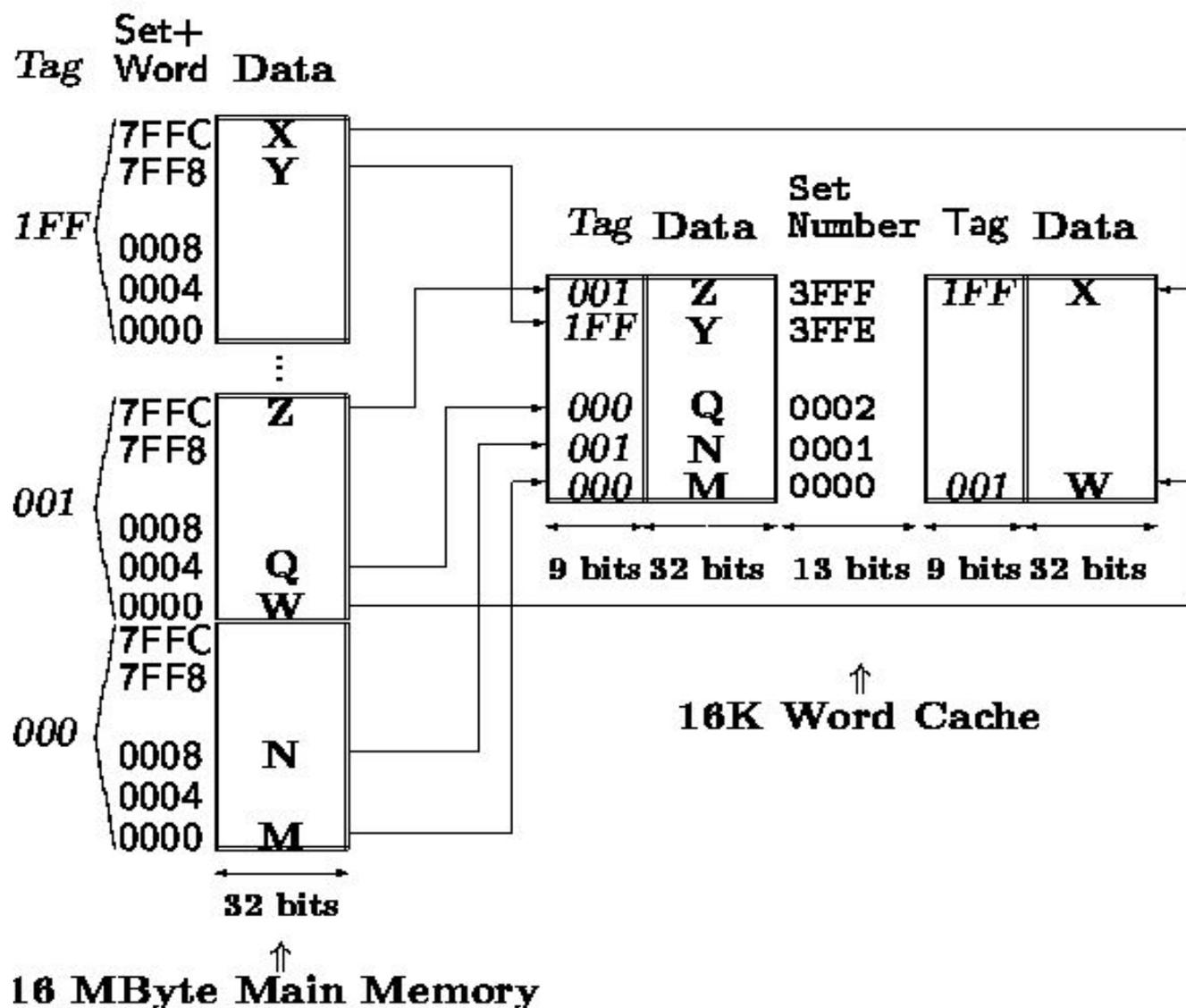


Figure 3: Associative Mapping

Зависимость количества промахов в кэш-память в зависимости от объема кэш-памяти и степени ассоциативности для длины троки 32 байта





MM Address =

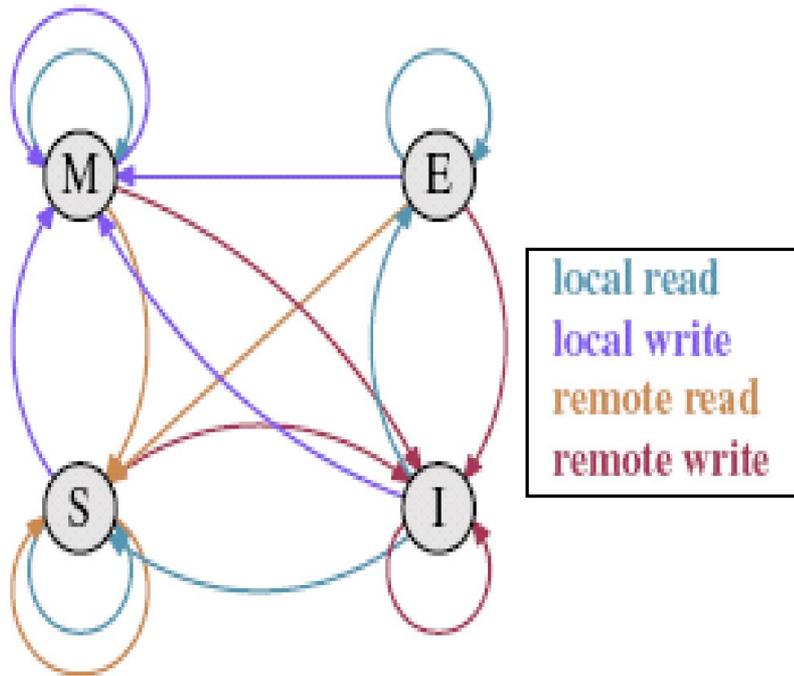
Tag	Slot	Word
9	13	2

Figure 4: Two-way Set-Associative Mapping

Алгоритмы записи

- Сквозная запись (**Write Through (WT)**).
- Сквозная запись с буферизацией (**Write Combining**).
- Обратная запись (**Write Back (WB)**).

Протокол MESI



- **M (Modified)**— данные в строке модифицированы, но не переписаны в ОП. Данные достоверны только в данной кэш-памяти.
- **E (Exclusive)**-- данные в строке совпадают с аналогичной строкой в ОП, но отсутствуют в других кэшах.
- **S (Shared)** – данные строки кэш-памяти не совпадают с данными аналогичной строкой в ОП и могут присутствовать в другой кэш-памяти.
- **I (Invalid)** – строка не содержит достоверных данных.

Алгоритм замещения (алгоритм псевдо-LRU)

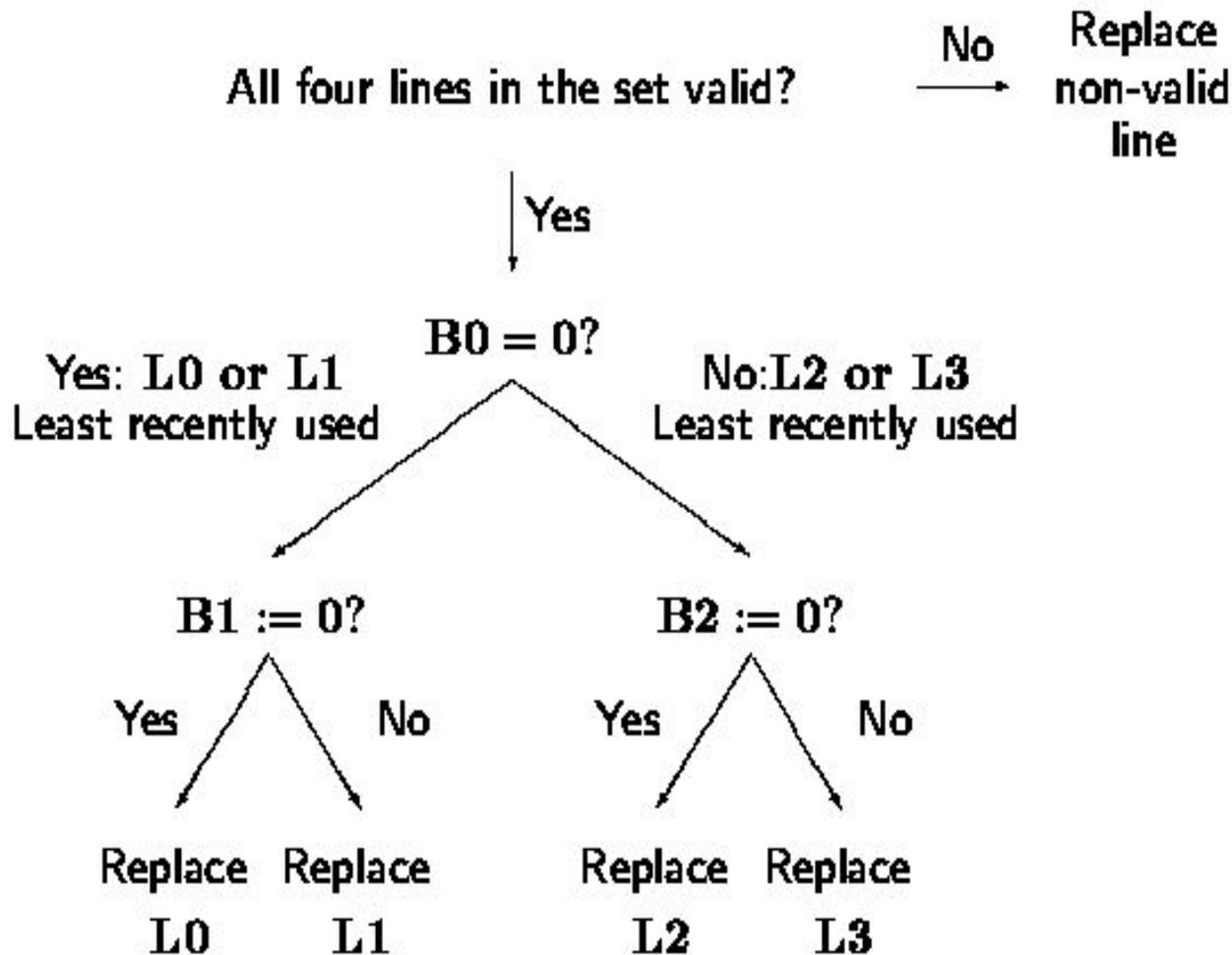


Figure 6: Intel 80486 on-chip cache replacement algorithm

Каким должен быть размер линии кэша-памяти?

- Размер линии должен быть как минимум в ширину канала памяти
- **Большой размер**
 - Более эффективное использование канала памяти при последовательном доступе
 - Позволяет уменьшать “ассоциативность” кэша и количество линий
- **Маленький размер**
 - Более эффективное использование канала памяти при произвольном доступе
 - Заполнение можно делать за одну транзакцию к памяти

Процессор	Pentium 4, Xeon (Northwood)	Itanium2	Opteron	Alpha 21264	PowerPC 970FX
Программные регистры	8 целочисл. (32 бит), 8 веществ. (80 бит), 16 векторных (128 бит)	128 целочисл. (64 бит), 128 веществ. (82 бит), 64 предикатных (1 бит), 8 регистров ветвлений (64 бит), 128 прикладных регистра	16 целочисл. (64 бит), 8 веществ. (80-бит), 16 векторных (128-бит)	32 целочисл. (64 бит), 32 веществ. (64 бит)	32 целочисл (64 бит), 32 веществ. (64 бит), 16 векторных (128-бит)
Аппаратные регистры	целочисл. 128 (32 бит), веществ. 128 (128 бит)	соответствуют программным	40 целочисл., 120 веществ.	80 целочисл., 72 веществ.	32 + 48 целочисл., 32 + 48 веществ., 16 + 16 векторных
Кэш данных L1	8 КВ, 4-way, строка 64 В,	16 КВ, 4-way, строка 64 В,	64 КВ, 2-way, строка 64 В,	64 КВ, 2-way, строка 64 В,	32 КВ, 2-way, строка 128 В,
Кэш команд L1	кэш трасс, 12 К микроопераций, 8-way	16 КВ, 4-way, строка 64 В,	64 КВ, 2-way, строка 64 В	64 КВ, 2-way, строка 64 В,	64 КВ, прямого отображения,
Кэш L2	512 КВ, 8-way, строка 64В,	256 КВ, 8-way, строка 128В,	1 МВ, 16-way, строка 64 В,	1 – 4 МВ, внешний,	512 КВ, 8-way, строка 128В,
Кэш L3		1.5 – 9 МВ, 12-way, строка 128В,			