

Средства симуляции ЦП и ОС и изучение поведения программ

Лекция №8



Державин Андрей
Шурыгин Антон

➤ **Квиз**

- Сверхоперативная память
- Устройство кэшей
- Моделирование

Код викторины:

00729331



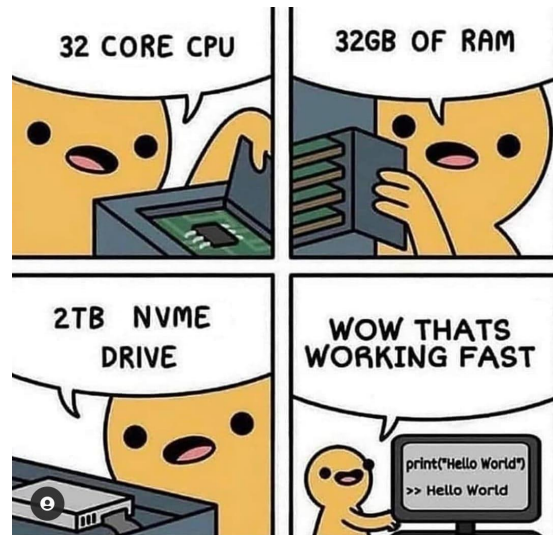
- Квиз

➤ **Сверхоперативная память**

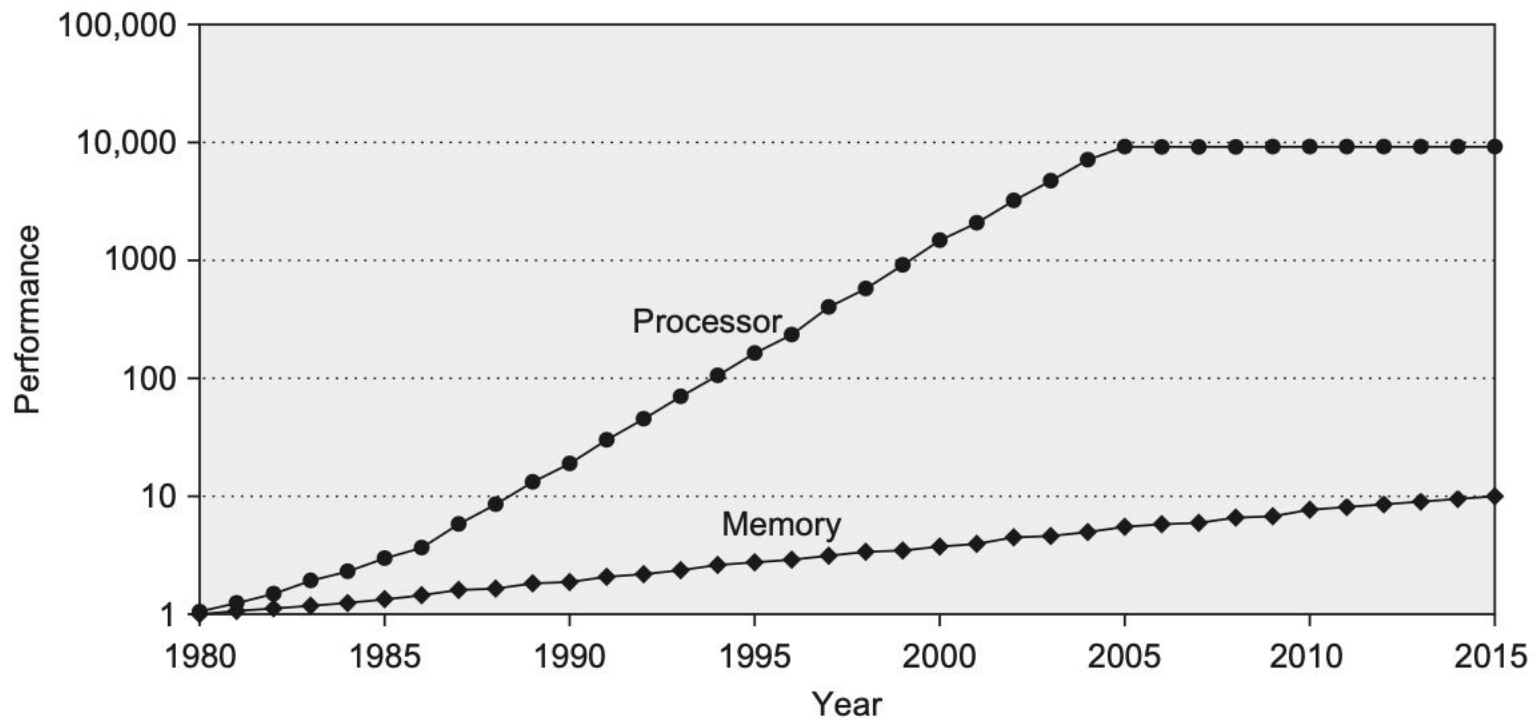
- Устройство кэшей
- Моделирование

Стена памяти

- Долгое время микропроцессоры стремительно ускорялись
- Было замечено, что улучшение микропроцессоров не дает существенного ускорения программ
 - Скорость оперативной памяти растет не так быстро
 - Процессор вынужден простаивать ожидая данные из памяти
- Данная проблема получила название *стена памяти (memory wall)*
 - При неизменном принципе организации вычислений, рост производительности систем будущего ограничен скоростью оперативной памяти

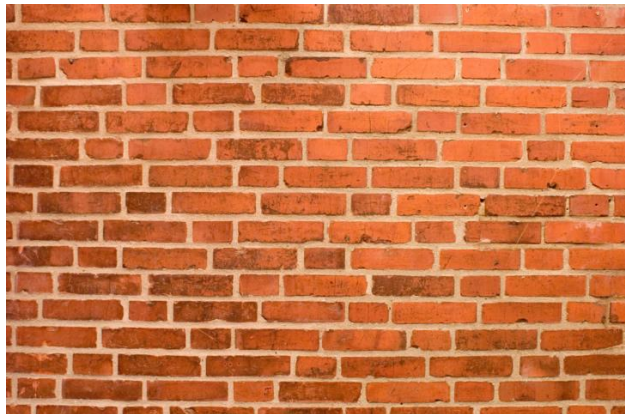


Стена памяти



Сверхоперативная память

- Представляет собой устройство с быстрым доступом, хранящее копию некоторого подмножества данных основной памяти
- Две области применения
 - **Частичное** решение проблемы “стены памяти”
 - Быстрая синхронизация многопроцессорных систем



Сверхоперативная память

- Рассмотрим общую схему работы и предпосылки кэшей
- Какая программа будет работать быстрее и почему?
 - image - двумерный массив

```
for (int y = 0; y < height; ++y) {  
    for (int x = 0; x < width; ++x) {  
        foo(image[y][x]);  
    }  
}
```

```
for (int x = 0; x < width; ++x) {  
    for (int y = 0; y < height; ++y) {  
        foo(image[y][x]);  
    }  
}
```


Сверхоперативная память

- Концепция кэшей основывается на двух принципах локальностей, выполняющихся во многих алгоритмах:
 - Временная - данные, к которым обращались недавно, будут запрошены вновь
 - Пространственная - соседние данные скорее всего также будут запрошены
- Идея: храним наиболее “нужные” данные
- Два способа попаданий данных в кэш
 - явный запрос загрузки по первому обращению
 - предварительная загрузка (prefetching) ручная либо автоматическая

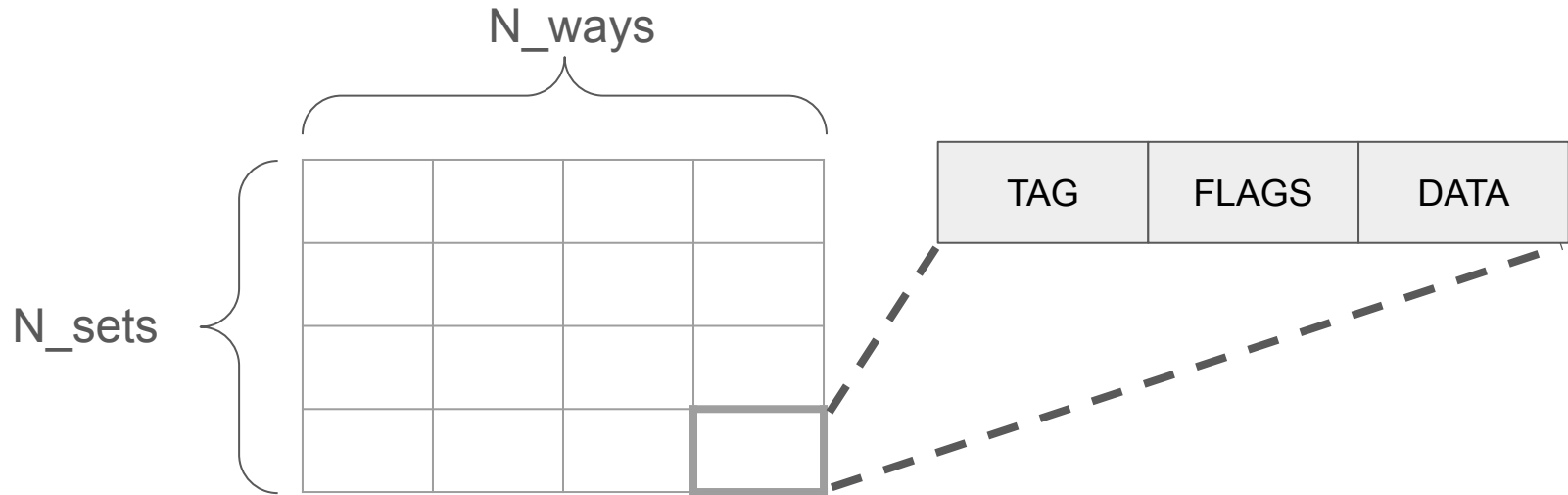
Сверхоперативная память. Операции

- В процессе работы с кэш-памятью возникают разные ситуации
- Ёмкость кэша невелика => иногда нужно освободить часть данных для новых
 - Старые данные вытесняются (**eviction**)
- Данные, хранящиеся в кэше становятся неактуальными
 - Выполняется сброс всех данных (**flush**)
- Выбор данных для вытеснения обычно определяется соответствующей политикой

- Квиз
- Сверхоперативная память
- **Устройство кэшей**
- Моделирование

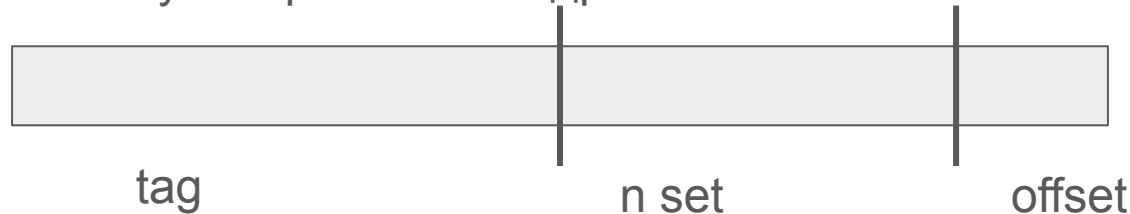
Устройство кэша

- Рассмотрим общий принцип организации кэш-памяти
- Единица хранимых данных - *кэш-линия* (обычно 32/64 байта)



Устройство кэша. Общий алгоритм

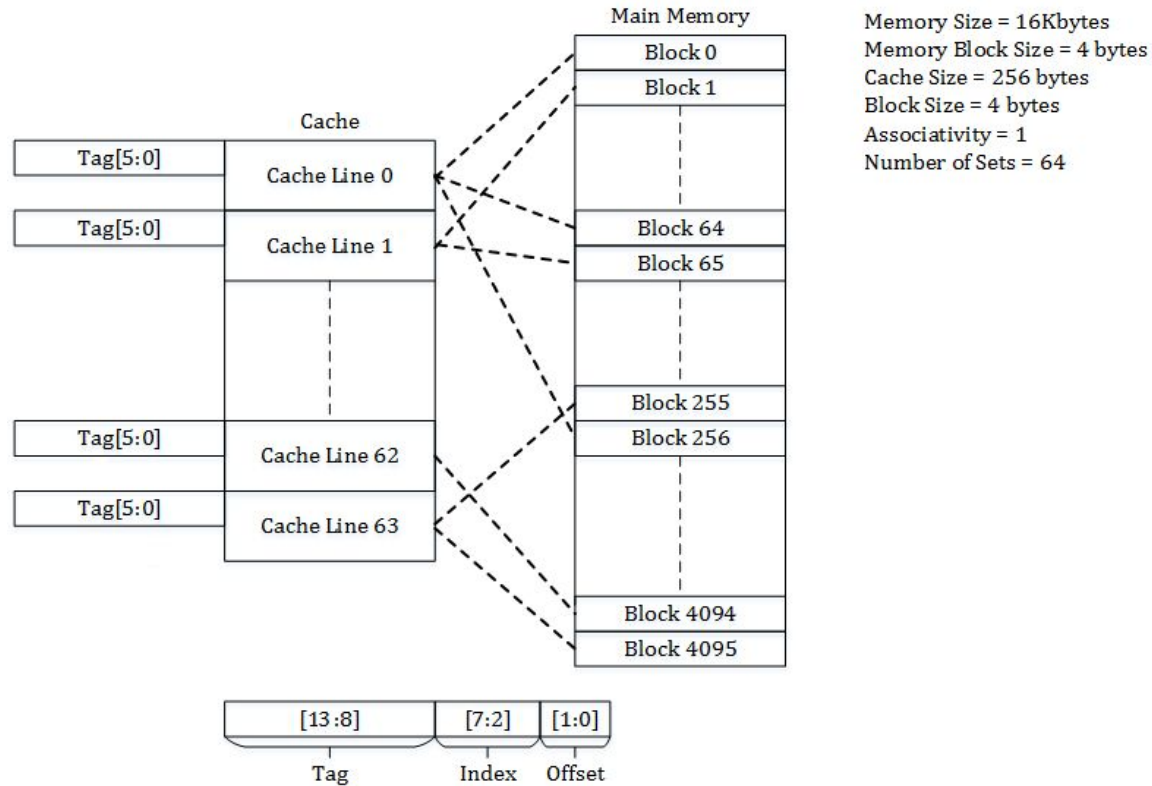
- Какая основная задача кэша?
 - Быстро найти соответствие “адрес-данные” (cache-hit), либо его отсутствие (cache miss)
- Из адреса в памяти выделяются смещение внутри линии, тэг и номер сета
- Выбирается указанный сет и осуществляется поиск вхождения с таким же тэгом среди N_ways (считается, что он параллельный)
- Проверяется валидность записи с помощью флага
- Каким образом лучше разбивать адрес на составные части?



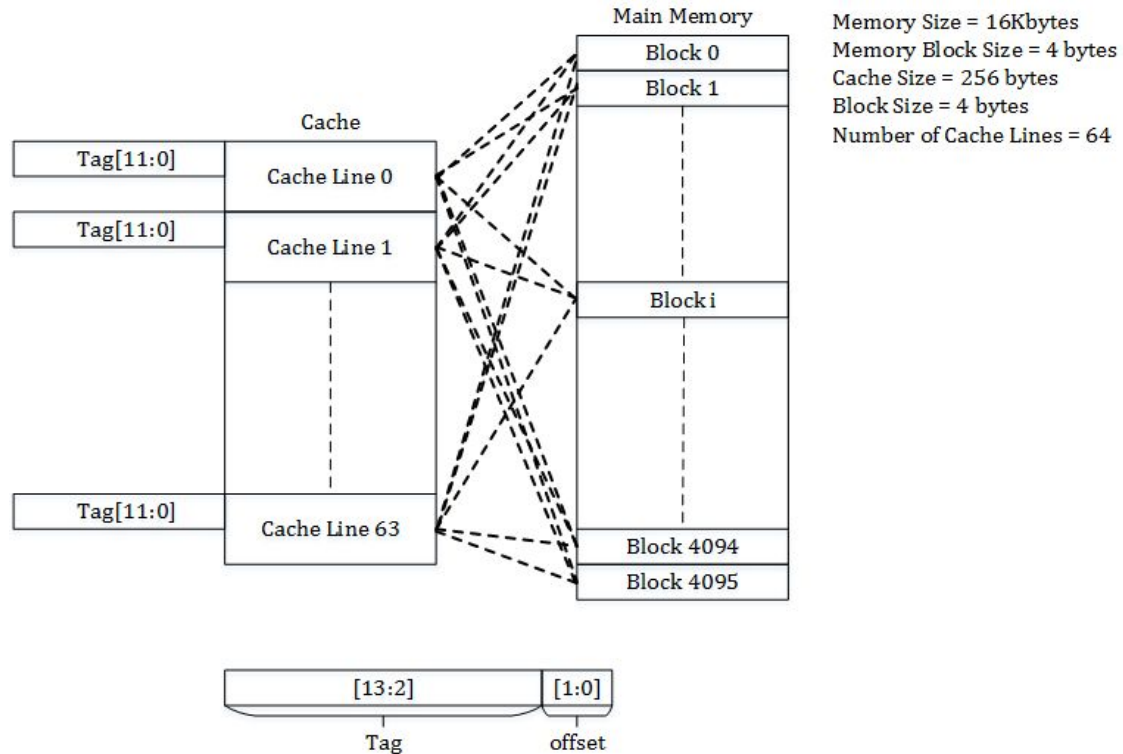
Устройство кэша. Параметры

- Рассмотрим параметры обобщённого кэша.
- Размер одной линии данных - S_{line}
- Ассоциативность - N_{ways} . Определяет число способов разместить одну и ту же линию в кэше
 - $N_{ways} == 1$ - Кэш прямого отображения (*direct mapped cache*)
- N_{sets} - число сетов
 - $N_{sets} == 1$ - Полностью ассоциативный кэш (*fully associative cache*)
- Ёмкость кэша. Выражается из вышеуказанных величин
 - $C = N_{ways} * N_{sets} * S_{line}$

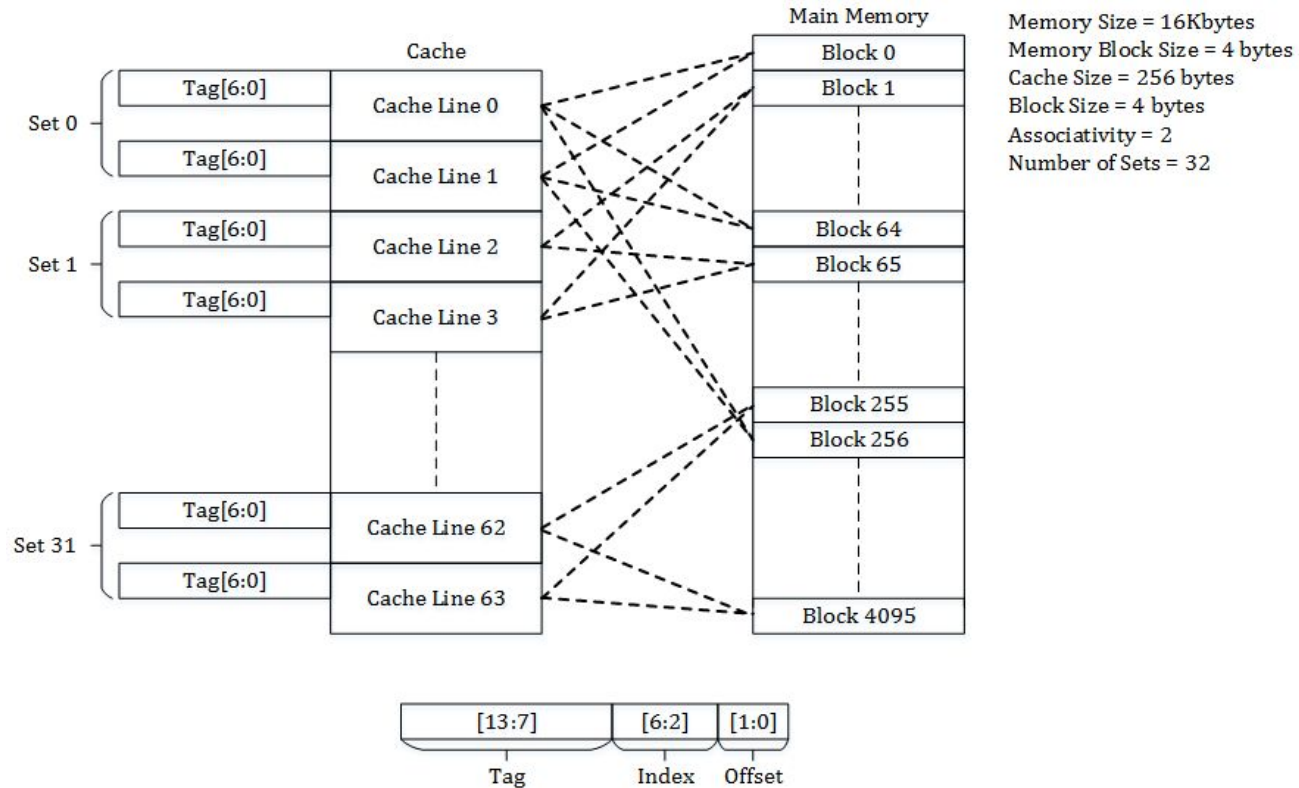
Устройство кэша. Direct-mapped cache



Устройство кэша. Fully-associative cache



Устройство кэша. Set-associative cache



Устройство кэша. Политики вытеснения

- Что делать, если произошел промах?
 - Выбрать свободную/невалидную линию и поместить новые данные
- Что делать, если нет свободных мест?
 - Нужно выбрать линию для вытеснения
- Как выбрать ячейку для вытеснения?
 - Существуют разные подходы
- Какая политика вытеснения считается “Идеальной”?
 - Кэш Беладзи - вытесняем линию, к которой дольше всего не будет обращений
- К сожалению, мы не можем заглянуть в будущее
- Опираемся на историю обращений для принятия выбора линии для вытеснения

Устройство кэша. Политики вытеснения

- Вытеснять всегда первую ячейку
 - Подходит только для direct mapped cache
- Вытеснять случайную ячейку
 - Может давать не оптимальные результаты
- FIFO - First In, First Out
 - Очередь внутри сэта, вытесняется самая старая ячейка внутри сэта
 - Требуется потоковое чтение памяти
- LRU - least recently used
- LFU - least frequently used
- MRU - most recently used
- Много других политик...

Устройство кэша. Трансляция адресов

- В современных процессорах используется абстракция виртуальной памяти
- Поиск в кэше может осуществляться по физическому, виртуальному или комбинации адресов
- При использовании виртуальных адресов следует учитывать следующие факторы:
 - Задержки - трансляция адреса требует некоторое время. Для ускорения процесса могут использоваться TLB, хранящие результаты трансляций
 - Наложение - несколько виртуальных адресов могут попадать в один физический
 - Проверять, что только одна линия с данным физ. адресом лежит в кэше

Устройство кэша. Адресация

- 4 вида адресации в кэшах:
- PIPT (Physically indexed, physically tagged)
 - простые, но медленные, нет aliasing
- VIVT (Virtually indexed, virtually tagged)
 - Быстрые, но есть aliasing и гомонимы (один виртуальный адрес на много физ.)
- VIPT (Virtually indexed, physically tagged)
 - Можно искать кэш линию одновременно с трансляцией
 - Обнаружение гомонимов
- PIVT (Physically indexed, virtually tagged)
 - не дают существенных преимуществ
 - чисто академический интерес

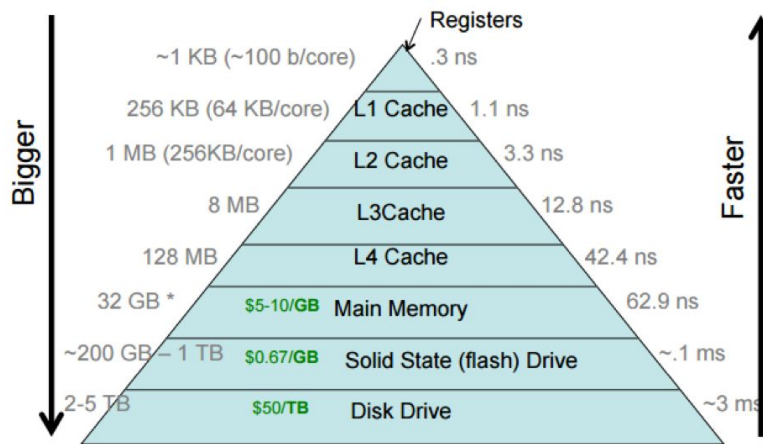
Устройство кэша. Иерархия

- Давайте увеличим N_ways и N_sets чтобы вместить как можно больше линий
- Все ли хорошо?
 - К сожалению, площадь на кристалле и допустимое энергопотребление ограничены



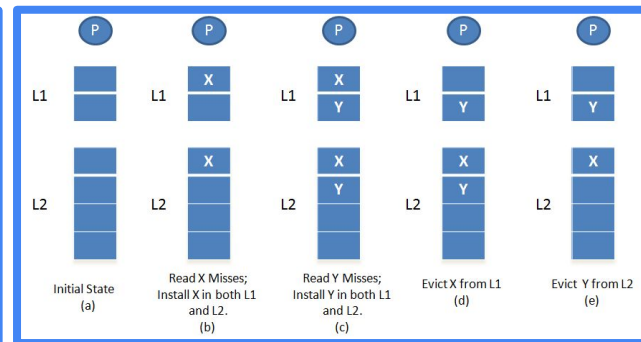
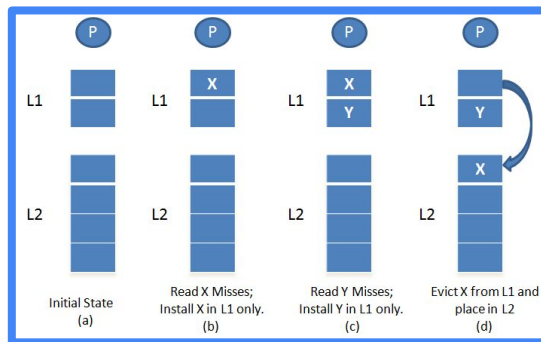
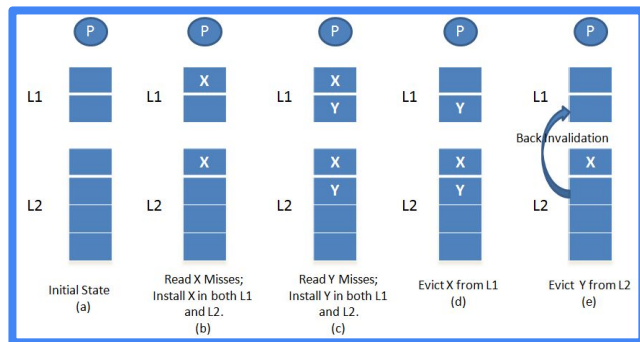
Устройство кэша. Иерархия

- Нельзя разместить большой кэш на кристалле
- Организуем кэш второго уровня L2
 - Располагается вне CPU => больше задержка
 - Имеет больший размер
- Данный подход можно масштабировать на следующие уровни



Устройство кэша. Иерархия

- Может ли одна линия находиться в нескольких кэшах?
- Есть 3 варианта
 - $L1 \subset L2$ - *инклюзивные кэши*
 - $L1 \cap L2 = \emptyset$ - *эксклюзивные кэши*
 - Ни то, ни другое - *non-inclusive non-exclusive (NINE)*



Устройство кэша. Кэш кода и данных

- В архитектуре фон-Неймана код и данные хранятся в одном пространстве памяти
- Множество используемых адресов, а также паттерны доступа к ним различаются у данных и кода
 - Может оказывать сильную нагрузку на кэши
- Отдельно выделяют кэш инструкций (IC) и кэш данных (DC).
 - Нужно следить за своевременной инвалидацией IC

Кэши в многопроцессорных системах

- Доступ к памяти имеет сразу несколько процессоров
- Требуется обеспечить согласованность данных, лежащих в кэшах разного уровня с реальным состоянием ОЗУ

Модели согласованности

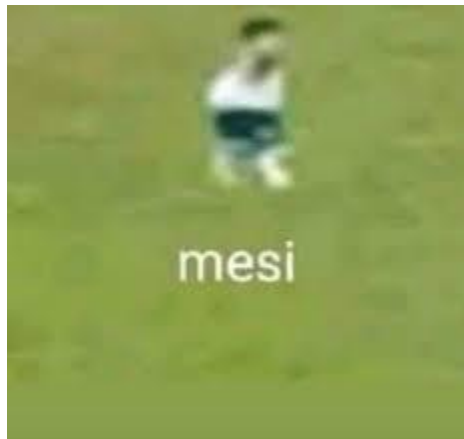
- Представляют собой некоторый контракт между программами и памятью
- Среди существующих моделей можно выделить следующие:
 - Строгая согласованность - чтение всегда возвращает актуальное значение
 - Последовательная - все процессы видят одинаковую последовательность записей
 - Причинная - фокусируется на зависимостях между операциями

Политики записи

- При записи данных в кэш, нужно знать момент времени, когда обновление придет в память
- 4 политики записи:
 - WT - write through - моментальная запись в память
 - WB - write back - обновление при вытеснении линии
 - WC - write combining - накопление буфера для записи для сброса в память
 - Не следует использовать со слабой согласованностью
 - Подходит для GPU
 - UC - uncacheable - запрет кэширования (для периферии)

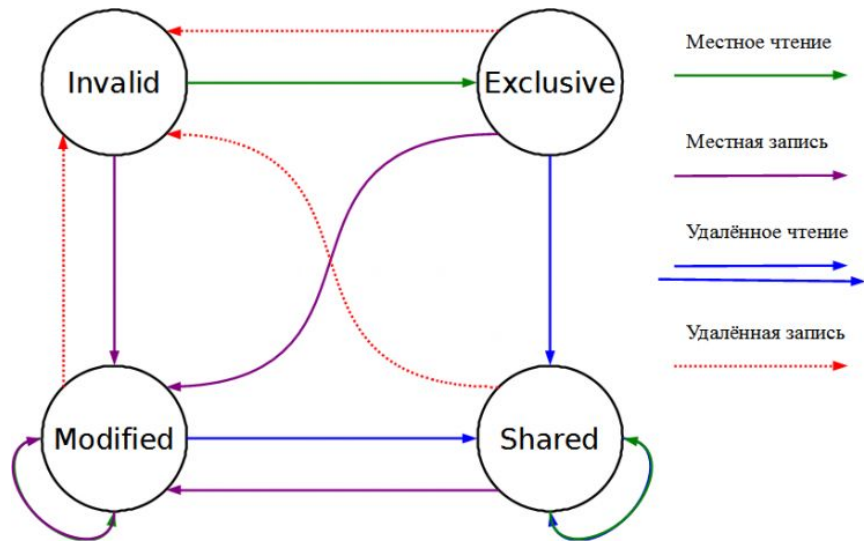
Протоколы когерентности

- Нужно поддерживать модель согласованности по всей иерархии кэшей
- При изменении линии генерируется сообщение, распространяющееся по иерархии
- Большинство современных протоколов есть вариации протокола MESI



Протоколы когерентности. MESI

- Каждая линия имеет только одно из состояний
 - М - модифицированная, находится только в одном кэше. Может быть записана
 - Е - эксклюзивная, как М, только данные идентичны ОЗУ
 - S - разделяемая на несколько кэшей. При записи всегда идет запрос на шину
 - Все записи помечаются I
 - I - невалидная линия (либо пустая)



- Квиз
- Сверхоперативная память
- Устройство кэшей
- **Моделирование**

Моделирование кэшей

- Как мы можем промоделировать кэши?
- “Честно”
 - полное моделирование иерархии кэшей, хранение данных
 - Поиск тэга осуществляется последовательно
- Модель задержек
 - Наличие кэшей выражается только в задержках доступов в память
 - Упрощает разработку модели и эксперименты с ней
 - Не подходит для транзакционной памяти

Влияние на симуляцию

- Подключим модель кэшей к функциональному симулятору
- Как изменится скорость?
 - Сильно упадёт - симулятор стал по сути потактовым
- Можно подключать симуляцию кэшей только на время изучения приложения
 - Кэши будут пустыми
- Будем “разогревать” кэши

Загрузка ОС. Кэши
отключены

Прогрев кэшей. Кэши
включены. Измерений
нет

Изучение приложения.
Измерения включены