ОС. Процессорный кеш

Лекция 3

Иерархия памяти

- Чем быстрее память тем больше цена
  - регистры и кеш на чипе (задержка ~1ns, размер несколько байт, 5k$ pgb)
  - кеш вне чипа (SRAM) (задержка ~10ns, размер несколько килобайт, 5k$ pgb)
  - основная память (DRAM) (задержка ~100ns, размер несколько гигабайт, 50$ pgb)
  - дисковая память (задержка ~10ms, размер несколько терабайт, <1$ pgb)
- Разница в производительности между CPU и RAM стала довольно значительной
- Пропускная способность памяти растет быстрее, чем уменьшается задержка
- Принцип локальности
  - пространственная локальность
  - временная локальность

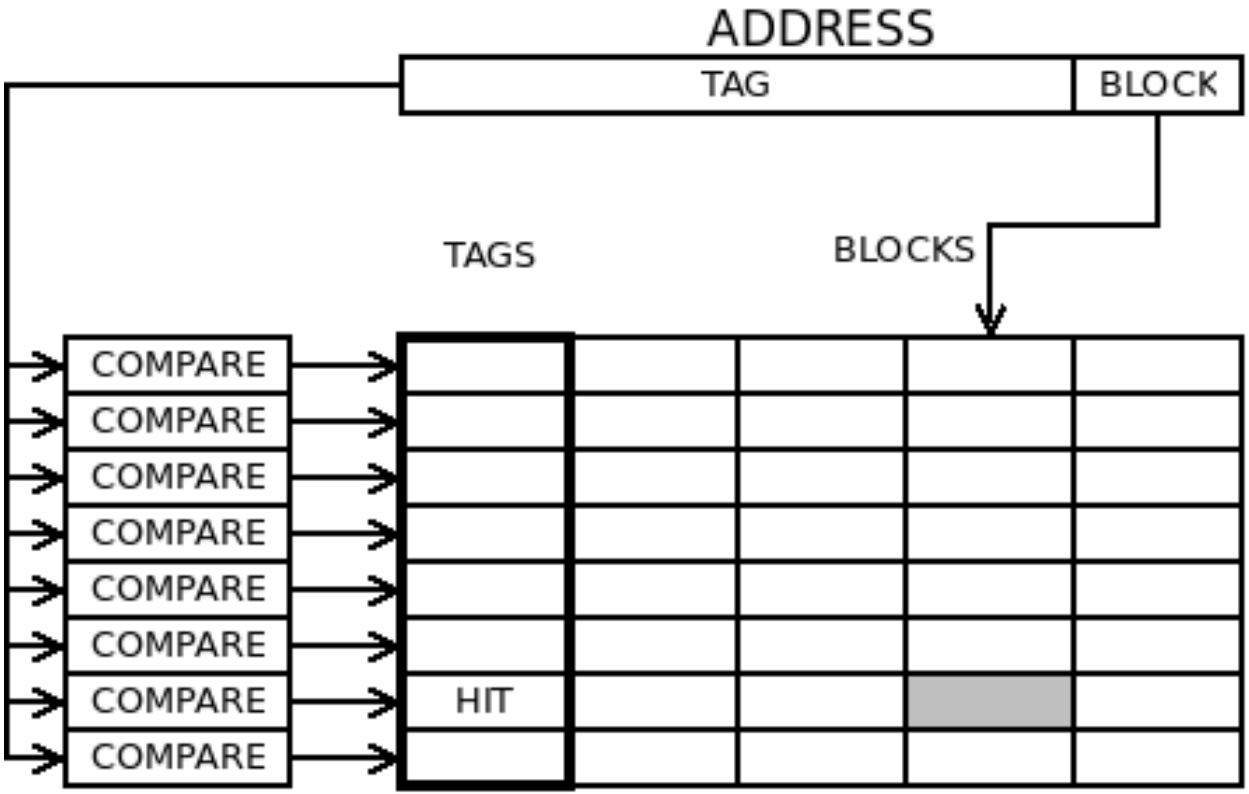
Линии кеша

- Линия хранит непрерывную последовательность байт из основной памяти
  - большая линия кеша - больше данных за раз подгружается в кеш
  - больше линий кеша - меньше вероятность, что данные уйдут из кеша
- Каждая линия имеет tag
  - позволяет узнать какой точно адрес занял линию
  - как правило используются старшие биты адреса
- Индекс линии в кеше определяется средними битами адреса
  - младшие биты не используются, потому что чтение происходит порциями равными размеру линии

Fully associative cache

- Любой адрес может занять любую позицию в кеше
  - у линий в кеше нет индекса - все они равнозначны
  - не страдает от конфликтов (нет индексов - нет конфликтов)
  - можно полностью утилизировать
- Недостатки
  - непрактичный (требует много элементов)
  - много элементов - больше задержка
  - вопрос с вытеснением линий кеша

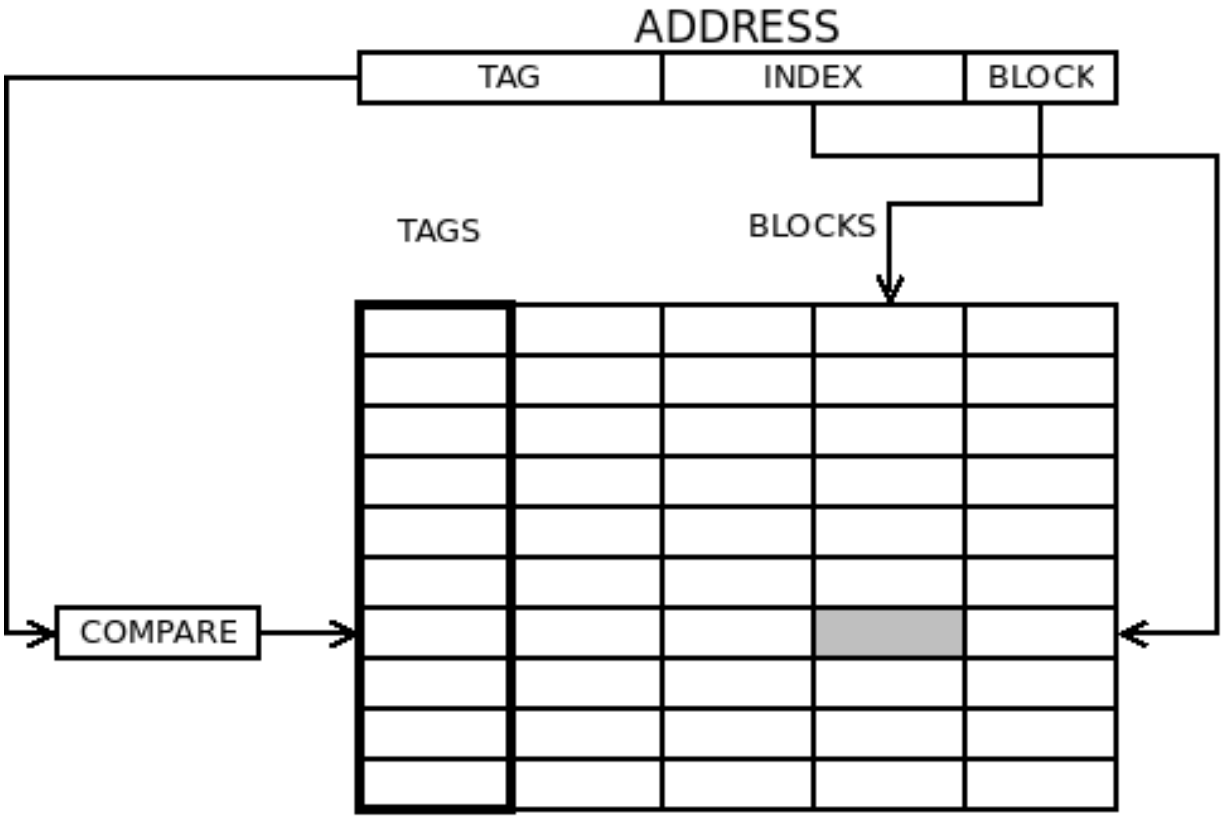
Fully associative cache



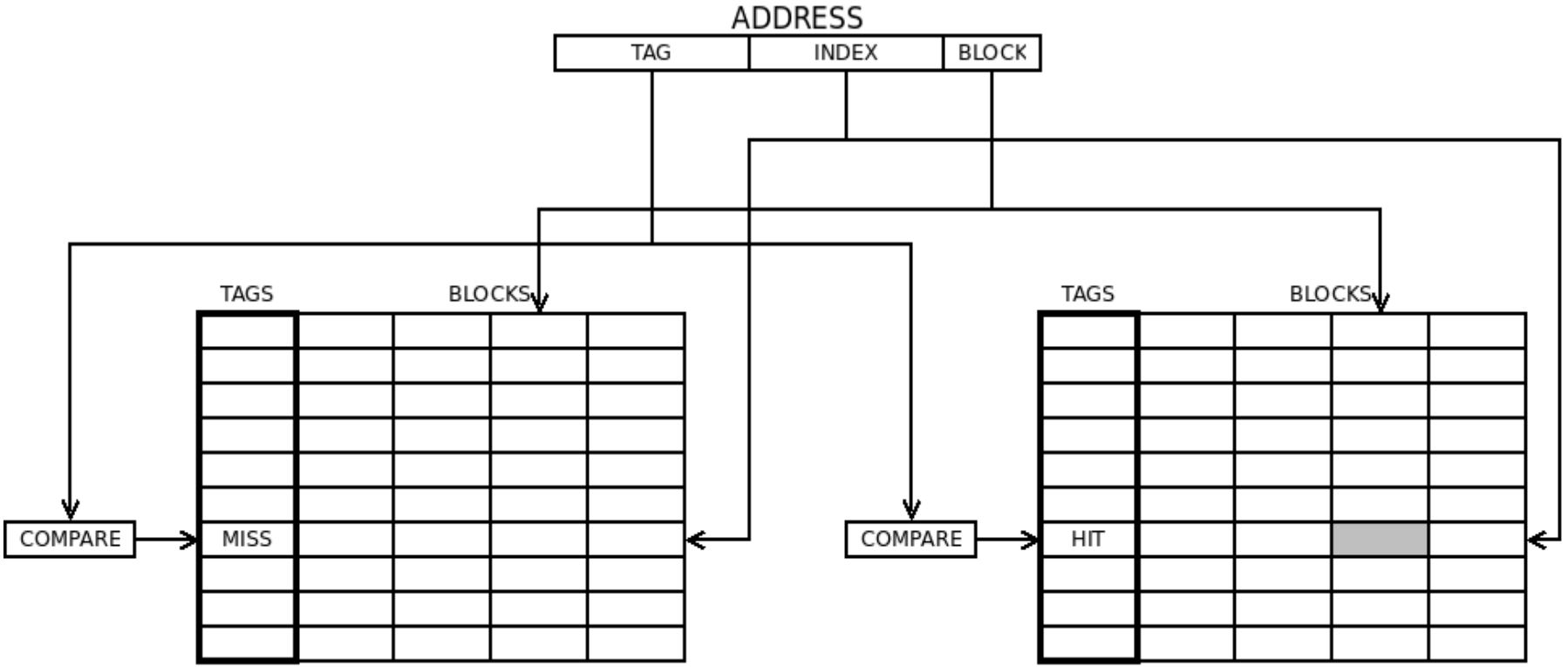
Direct mapped cache

- Каждый адрес может занять только одну строго определенную позицию в кеше
  - линия определяется средними битами адреса
  - страдает от конфликтов
  - плохая утилизация
- Достоинства
  - один компаратор
  - меньше элементов - быстрее доступ

Direct mapped cache



K-way associative cache



Cache coloring

- Проблема
  - все адреса с одинаковыми средними битами попадают в одни и те же линии кеша
  - одновременно могут использоваться только K объектов с одинаковыми средними битами
- Решение
  - сдвинем служебные данные в разных страницах SLAB-а на размер линии кеша
  - управляющие структуры из разных страниц будут иметь разный индекс, т. е. попадут в разные линии
  - получаем выигрыш при доступе к служебным структурам SLAB, за счет этого аллокация быстрее

Q&A