Computer Architecture

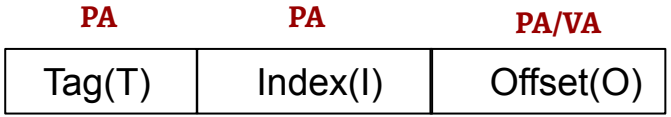
Cache Optimizations

Debadatta Mishra, CSE, IITK

Classifying cache misses

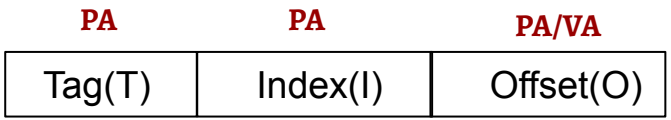
- Compulsory: First reference misses or cold misses
- How to measure? # of cache misses with infinite cache size
- How to reduce? Hardware/Software Prefetching
- Capacity misses: Non-compulsory misses occurring when the cache is full
- How to measure? Cache misses in fully associative cache (with Belady) - Compulsory misses
- How to reduce? Increased cache size, better cache replacement, prefetching
- Conflict misses: In set-associative or direct mapped caches, # of cache misses because of conflict in the set i.e., cache miss on access to a previously evicted block where the eviction was because of a full set
- How to measure? Total cache misses in k-way set associative cache - # of misses in fully associative cache
- How to reduce? Increased associativity, Other techniques: randomized mapping, victim cache …

Cache Types (Mapping)

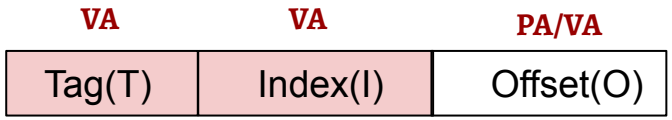


- Physically indexed and physically tagged (PIPT)
- Advantages/disadvantages?

Cache Types (Mapping)

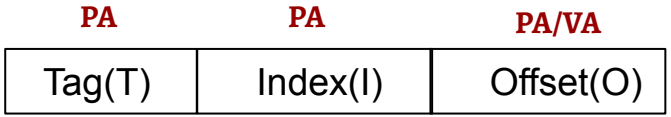


- Physically indexed and physically tagged (PIPT)
- Advantages/disadvantages? Independent of virtual address space. Issue: cache OP possible only after address translation

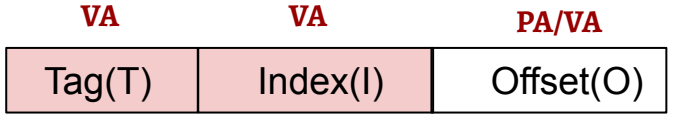


- Virtually indexed and virtually tagged (VIVT)
- Advantages/disadvantages?

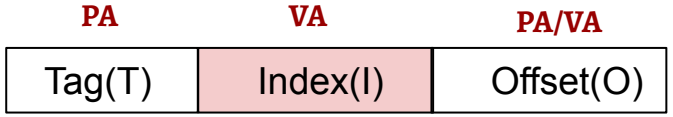
Cache Types (Mapping)



- Physically indexed and physically tagged (PIPT)
- Advantages/disadvantages? Independent of virtual address space. Issue: cache OP possible only after address translation

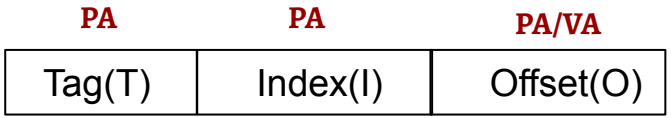


- Virtually indexed and virtually tagged (VIVT)
- Advantages/disadvantages? Cache OP independent of address translation. Issues: protection enforcement, context switches, synonyms or aliases

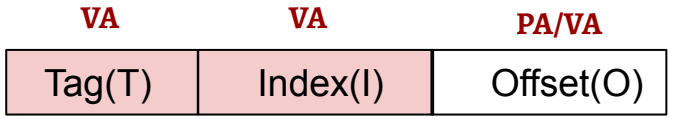


- Virtually indexed and physically tagged (VIPT)

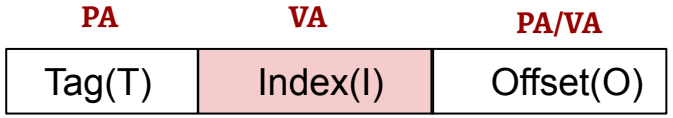
Cache Types (Mapping)



- Physically indexed and physically tagged (PIPT)
- Advantages/disadvantages? Independent of virtual address space. Issue: cache OP possible only after address translation



- Virtually indexed and virtually tagged (VIVT)
- Advantages/disadvantages? Cache OP independent of address translation. Issues: protection enforcement, context switches, synonyms or aliases



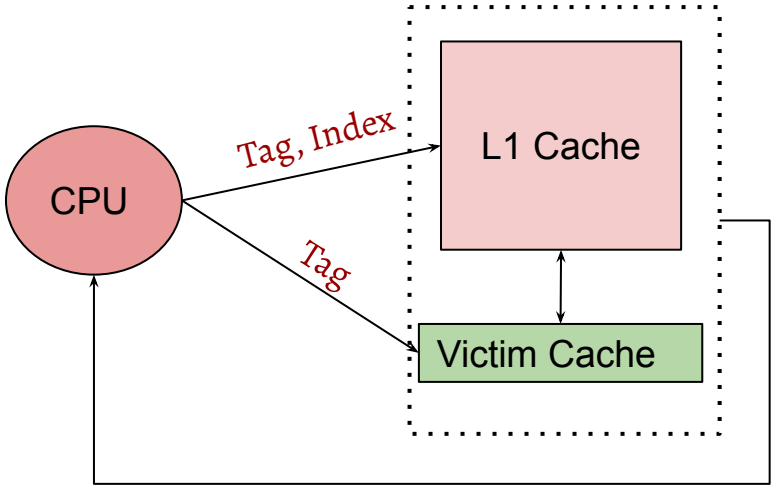
- Virtually indexed and physically tagged (VIPT)
- Data read can start after index mapping. Page size depends on block size and # of index bits

Reducing hit time

- Small caches with lower associativity ⇒ Reduced lookup time
- How to achieve lookup performance of direct mapped caches while reducing conflict misses?

Reducing hit time

- Small caches with lower associativity ⇒ Reduced lookup time
- How to achieve lookup performance of direct mapped caches while reducing conflict misses?



Victim cache (proposed by Jouppi in 1990)

- Direct mapped L1 cache with a fully associative small cache to hold evicted blocks
- Shown to be effective in reducing conflicts

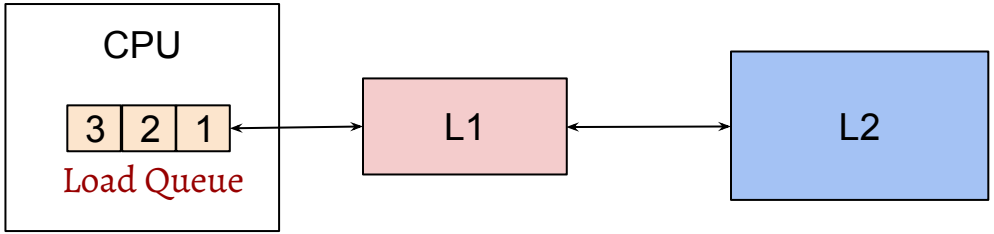
Reducing hit time

- Small caches with lower associativity ⇒ Reduced lookup time
- How to achieve lookup performance of direct mapped caches while reducing conflict misses?

Way prediction

- Extra-bits to predict the way in a set-associative cache for the next access
- Start tag comparison and reading data from the predicted cache line
- If mispredicted, check other blocks (extra cycles needed)
- Used in MIPS R10000 and ARM Cortex-A8

Non-blocking caches



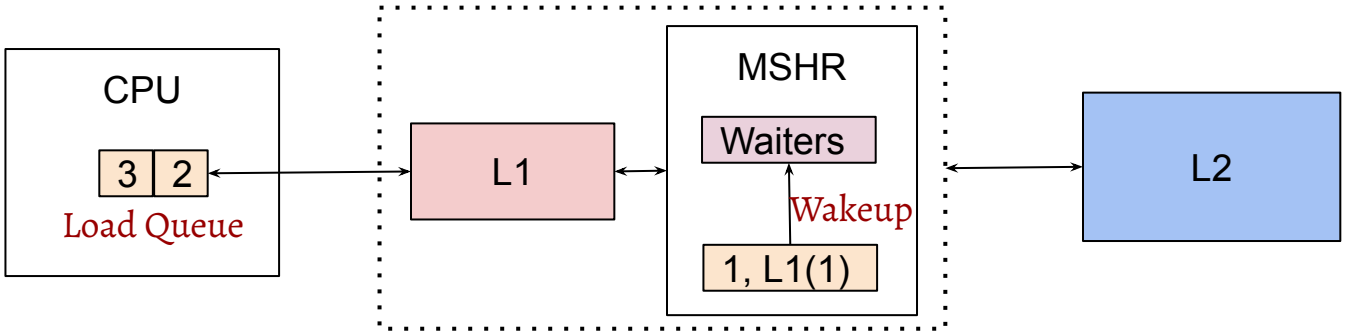
If '1' is a miss at L1, '2' can not be issued till the miss is handled at lower layers

Even worse, if '2' and '3' are cache hits at L1, may result in unnecessary stalls

- How to allow cache lookup operations to proceed while there are outstanding cache misses?

Non-blocking caches with MSHR

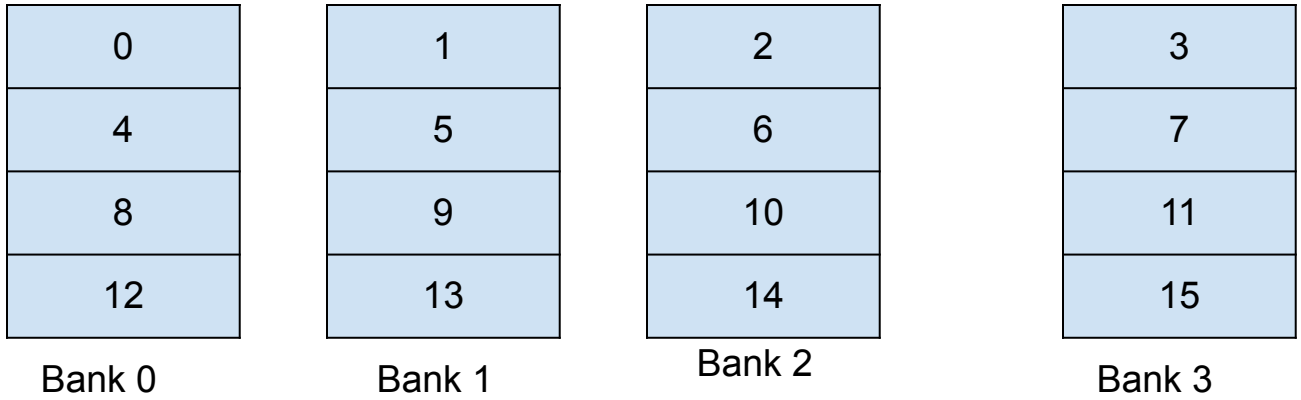
- How to allow cache lookup operations to proceed while there are outstanding cache misses?



- Miss status holding register (MSHR) holds information for outstanding cache misses
- On L1 miss, allocate a MSHR entry or merge with an existing entry
- MSHR allows lookup and merge (reduced bandwidth usage)
- Wake up waiting requests on arrival of required sub blocks from down the hierarchy
- Stall only when MSHR is full

Increase cache bandwidth with multiple banks

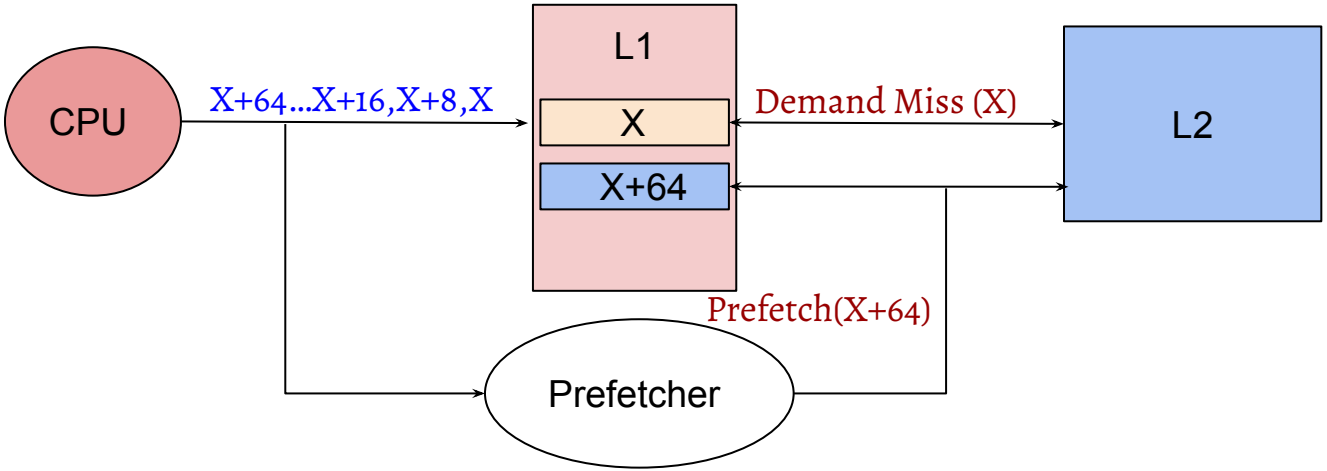
- Can we support more than one concurrent cache operations?



- Split the monolithic cache into banks (used in ARM and Intel)
- Maximum throughput when accesses spread across banks
- Sequential interleaving works well in many cases

Hardware prefetching

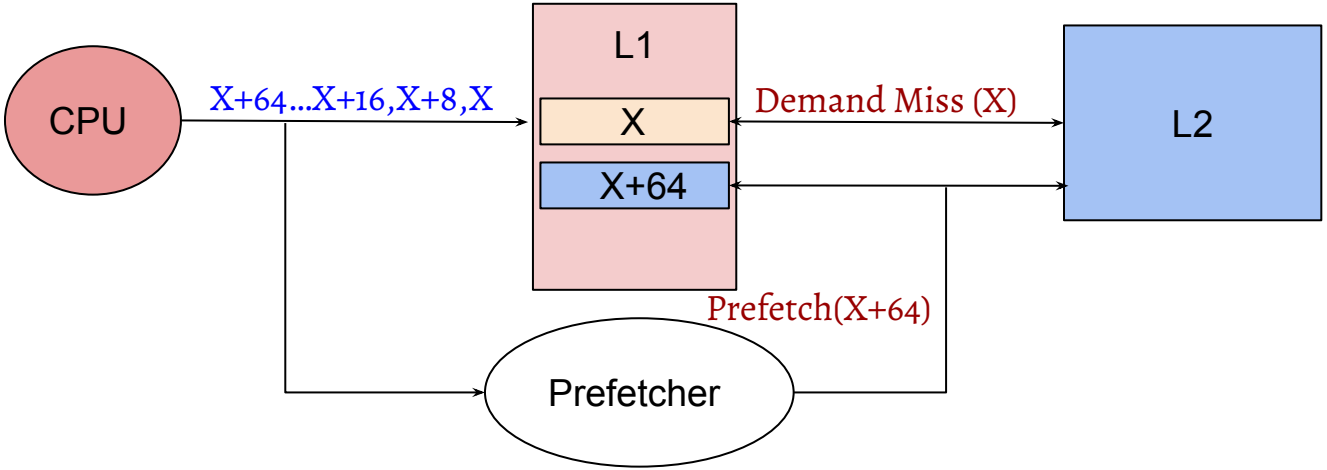
- Prefetch blocks before they are required by the CPU by predicting future access



- Design parameters: Degree of prefetch, prefetch distance
- Policy: which address to prefetch? Next line prefetcher, Stride prefetcher
- How good is a prefetcher?

Hardware prefetching

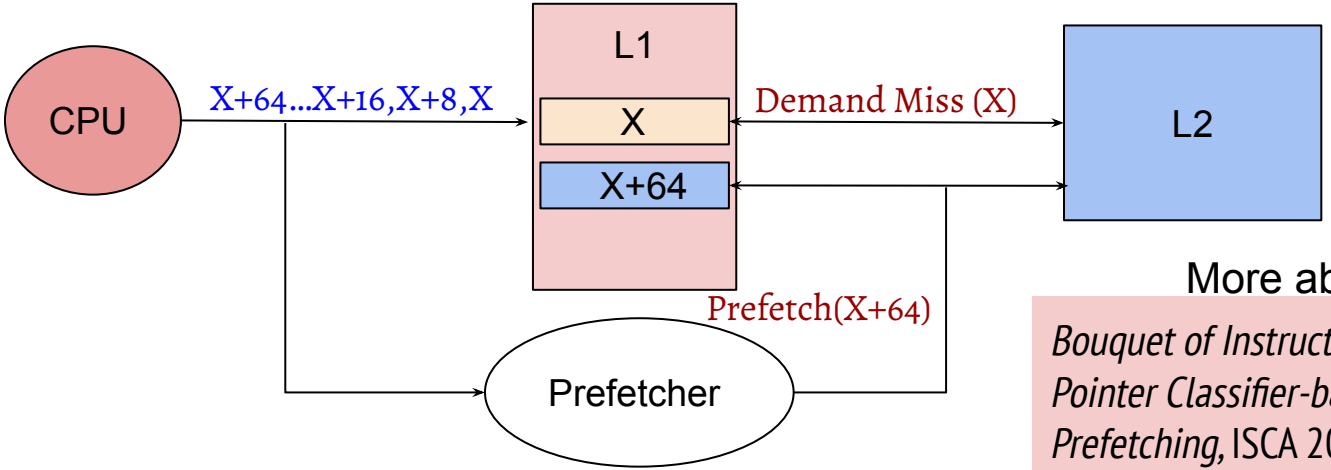
- Prefetch blocks before they are required by the CPU by predicting future access



- How good is a prefetcher?
  - Prefetch accuracy (Hits due to prefetch / Total prefetches)
  - Pollution (Misses due to eviction caused by the prefetcher)
  - Timeliness (Prefetched block status: in cache or in flight)

Hardware prefetching

- Prefetch blocks before they are required by the CPU by predicting future access



More about prefetchers

Bouquet of Instruction Pointers: Instruction Pointer Classifier-based Spatial Hardware Prefetching, ISCA 2020

- How good is a prefetcher?
  - Prefetch accuracy (Hits due to prefetch / Total prefetches)
  - Pollution (Misses due to eviction caused by the prefetcher)
  - Timeliness (Prefetched block status: in cache or in flight)

Software techniques

- Allocation
  - Alignment of structures/objects, placement of members, inlining vs. redirection
- Program logic restructuring
  - Loop interchange
  - Blocking
- Software prefetching
  - Use software prefetch instructions
  - X86 prefetch instruction
  - GCC: __builtin_prefetch