Chapter 10: Virtual Memory

Lesson 09:

Caches and Virtual Memory

Objective

- Understand the use of virtual memory also incorporating caches
- Learn cache implementations which vary in whether they use virtual addresses or physical addresses to select a set that might contain the address being referred
- Understand virtual addressed virtually tagged, physically addressed physically tagged, virtual addressed virtually tagged and physically addressed physically tagged caches

Use of virtual memory also incorporating at caches as the first level or levels of their memory hierarchy

Incorporating caches virtual or physical memory at the first level

- Cache implementations vary in whether they use virtual addresses or physical addresses to select a set that might contain the address being referenced by an instruction and whether they use virtual or physical addresses to determine if a hit has occurred

Fully associative caches

- Only have one set
- Don’t use either the virtual or the physical address of a memory reference to select a set
- Designers of fully associative caches do have to decide whether to use virtual or physical addresses to determine if a hit has occurred

Caches virtually addressed virtually tagged

Caches virtually addressed virtually tagged

- Use virtual addresses to both select a set and to determine if a hit has occurred

Advantage of caches virtually addressed virtually tagged

- Only necessary to do an address translation if a cache miss occurs, but the disadvantage that all the data in the cache must be invalidated when a context switch occurs, to prevent the new program from accessing the old program's data

Disadvantage of caches virtually addressed virtually tagged

- All the data in the cache must be invalidated when a context switch occurs, to prevent the new program from accessing the old program's data

Use of process ID

- Adding process ID bits to each tag in the cache can eliminate this problem at the cost of increasing the amount of storage required for the tag array

Physically addressed physically tagged Caches

Physically addressed physically tagged Caches

- Use physical addresses to select sets and to determine if a hit has occurred have no problems with virtual address aliasing between programs
- Can take advantage of the fact that physical addresses are shorter than virtual addresses on many systems to reduce the size of their tag arrays

Disadvantage of Physically addressed physically tagged Caches

- Necessary to perform address translation before the cache is accessed, which increases the cache hit time

Physically addressed virtually tagged caches

Physically addressedvirtually tagged caches

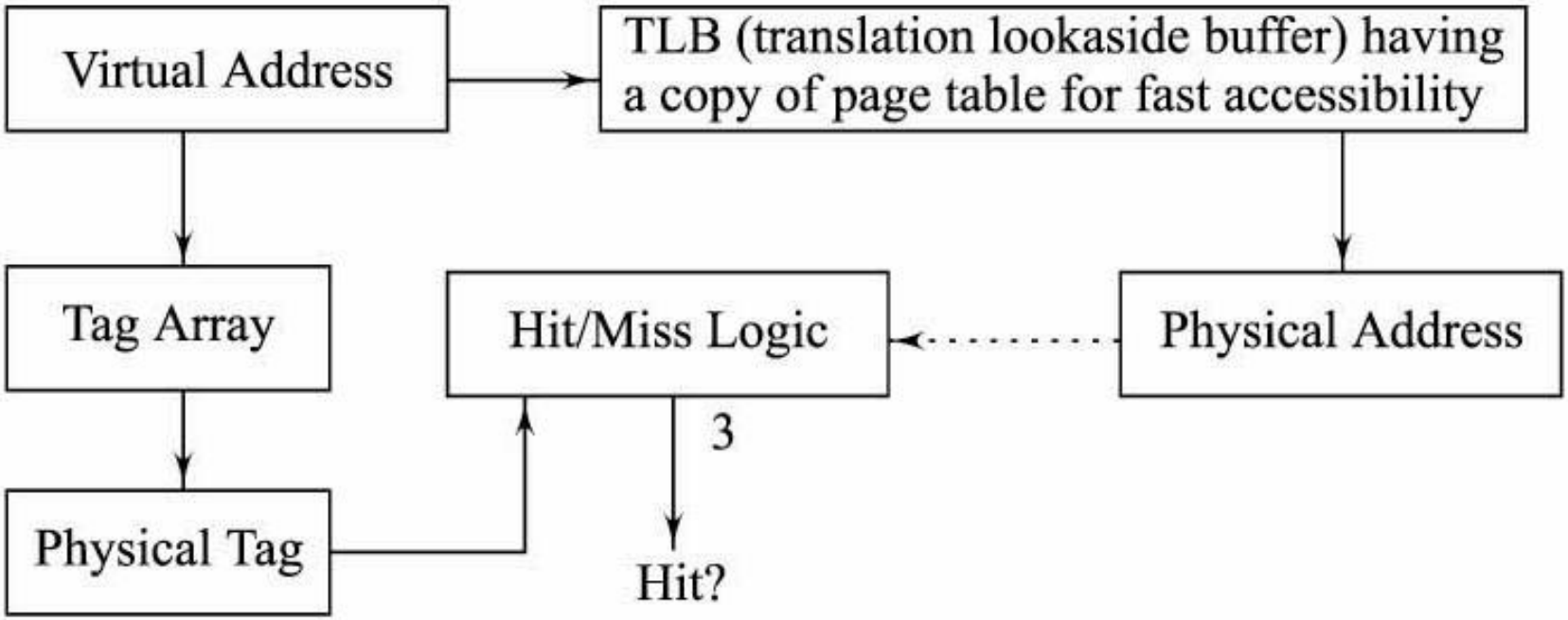
- Systems combine the worst aspects of virtual and physical addressing and are pretty much never used

Physically addressedvirtually tagged caches

- Using the physical address to select a set within the cache means that the cache access needs to wait for address translation to complete before it can begin, while the virtual tag means that the cache doesn't provide protection against the use of the same virtual address by multiple programs

Virtually addressed physically tagged caches

Virtually addressed physically tagged cache access



Virtually addressed physically tagged caches

- Using virtual addresses to select a set within the cache and physical addresses to determine if a hit has occurred allows the cache lookup to begin in parallel with address translation but provides protection, unlike virtually addressed/virtually tagged systems

Advantage

- As long as address translation takes less time than accessing the tag array, this type of cache can be as fast as a virtually addressed/virtually tagged cache, since the physical address isn't needed for hit/miss determination until after the tag array lookup completes

TLBs

- Tend to contain less data than a tag array
- Address translation generally faster than accessing the tag array unless a TLB miss occurs
- This combination of speed and protection makes virtually addressed/physically tagged caches the choice for most current systems

Level 1 cache

Level 1 cache

- Typically implemented as virtually addressed/physically tagged in systems with more than one level of cache memory
- Each level in the cache is accessed in sequence

Lower Level caches

Lower-level caches

- Usually physically addressed/physically tagged
- Address translation is performed during the level 1 cache access, so the physical address is available by the time that the accesses to the level 2 and lower caches begin, making it simpler and equally fast to use physically addressed/physically tagged caches for these levels

Parallel memory references at each level of caches

Check at each level of the cache in some systems

- In parallel on a memory reference
- Reduces the access time to lower levels of the cache
- Might use virtually addressed/physically tagged caches on all of their caches
- Allow accesses to all levels of the cache to begin immediately

Summary

We learnt

- Cache implementations vary in whether they use virtual addresses or physical addresses to select a set that might contain the address being referenced
- Virtual addressed virtually tagged, Physically addressed physically tagged cache, Virtual addressed virtually tagged cache and Physically addressed physically tagged caches

We learnt

- Level 1 caches typically implemented as virtually addressed/physically tagged in systems with more than one level of cache memory
- Lower level caches usually physically addressed/physically tagged
- In sequence or parallel on a memory reference to each level of caches

End of Lesson 09 on

Caches and Virtual Memory