

Chapter 10: Virtual Memory

Lesson 05:

Translation Lookaside Buffers

Objective

- Learn that a page table entry access increases the latency for a memory reference
- Understand that how use of translation-lookaside-buffers (TLBs) reduces the access latency many times
- Develop ability to solve problems relating to TLB hit and miss

Translation lookaside buffers

Disadvantage of using page tables for address translation

- A page table must be accessed on each memory reference
- On a system with a single-level page table, this doubles the number of memory accesses required
- Each load or store operation requires one memory reference to access the appropriate page table entry and one to perform the actual load or store

Disadvantage

- Greatly increases the latency (waiting period) of a memory reference
- Problem is even greater on systems that have multilevel page tables
- Multiple memory references are required to traverse the page table

Reducing penalty for first the page table access

- Processing delay due to need to access first the page table
- Solution— Incorporate translation lookaside buffers (TLBs)
- TLB— act as caches for the page table

TLB

- Whenever a program performs a memory reference, the virtual address sent to the TLB to determine if it contains a translation for the address
- Yes— the TLB returns the physical address of the data, and the memory reference continues
- No—, a *TLB miss* occurs, and the system searches the page table for the translation

Translation lookaside buffer

- Organization of memory like in a fully associative Content addressable memory

Example

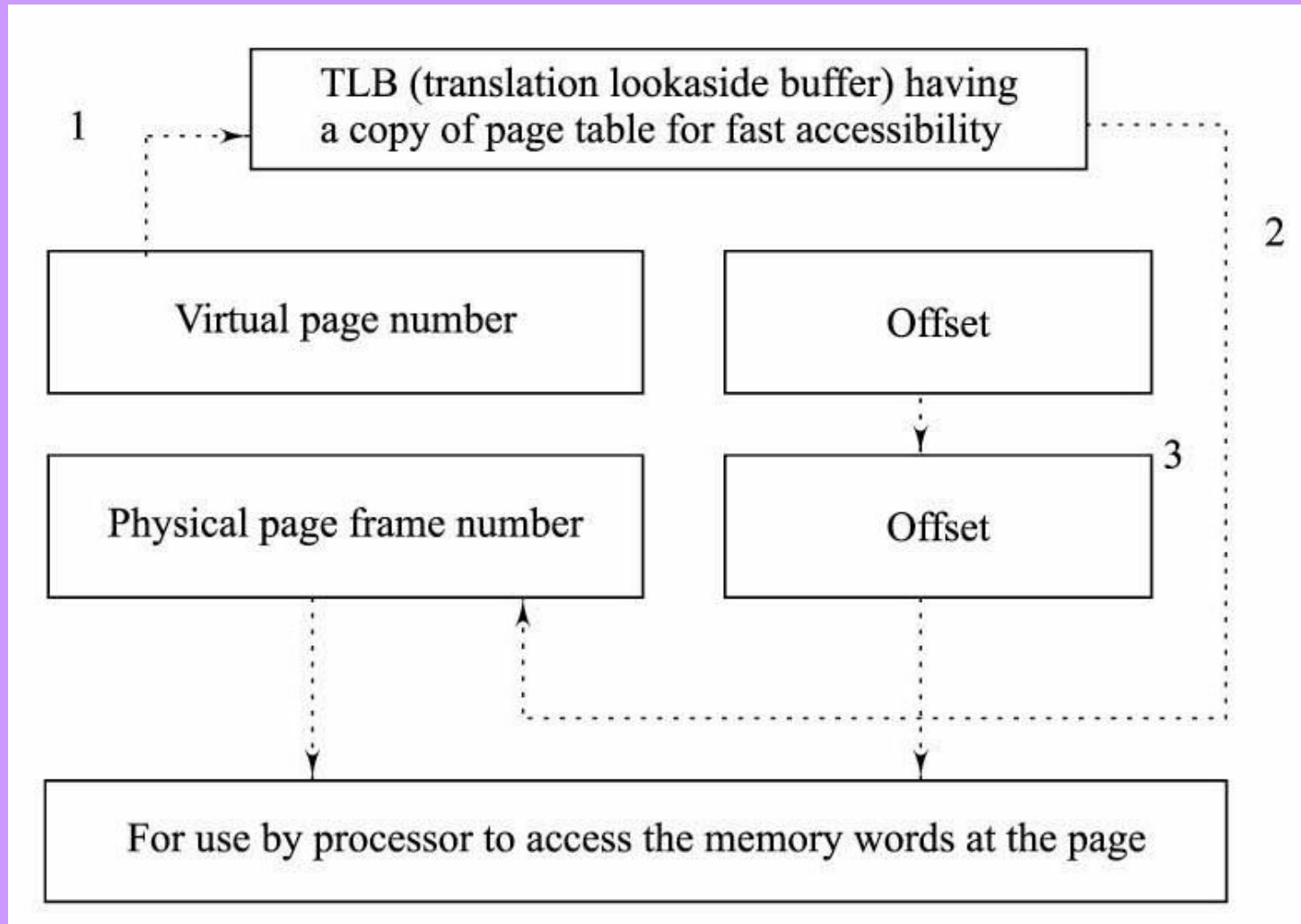
- A TLB has a hit rate of 95 percent and the TLB miss penalty $T_{\text{miss}} = 150$ cycles
- Assume that, on a TLB hit, address translation takes $T_{\text{hit}} = 0$
- Compute the average time required for an address translation

Solution

- Similar to a formula for the caches
- The average time for an address translation is $T_{\text{hit}} \times P_{\text{hit}} + T_{\text{miss}} \times P_{\text{miss}}$
- Plugging in the probabilities and delay numbers, an average address translation time of $(0 \times 0.95 + 0.05 \times 150) = 7.5$ cycles
- A reduction of 20 times over the translation time without the TLB

Address translation using translation lookaside buffers

Address Translation in a case when a TLB used



TLB hit and miss and

Hit in the TLB

- TLB contains a translation for the virtual address and the physical address of the reference can be used to complete the memory reference in hardware without software involvement
- When a page is evicted from the main memory, translations for the page are evicted from the TLB as well
- A TLB hit means that the physical page containing the address is mapped in memory

TLB miss, but the page mapped

- System accesses the page table to find the translation for the virtual address
- Copies that translation into the TLB, and the memory reference proceeds
- TLB misses generally take a relatively short time to resolve, because the system just has to access the page table

Time to resolve TLB miss when the page mapped

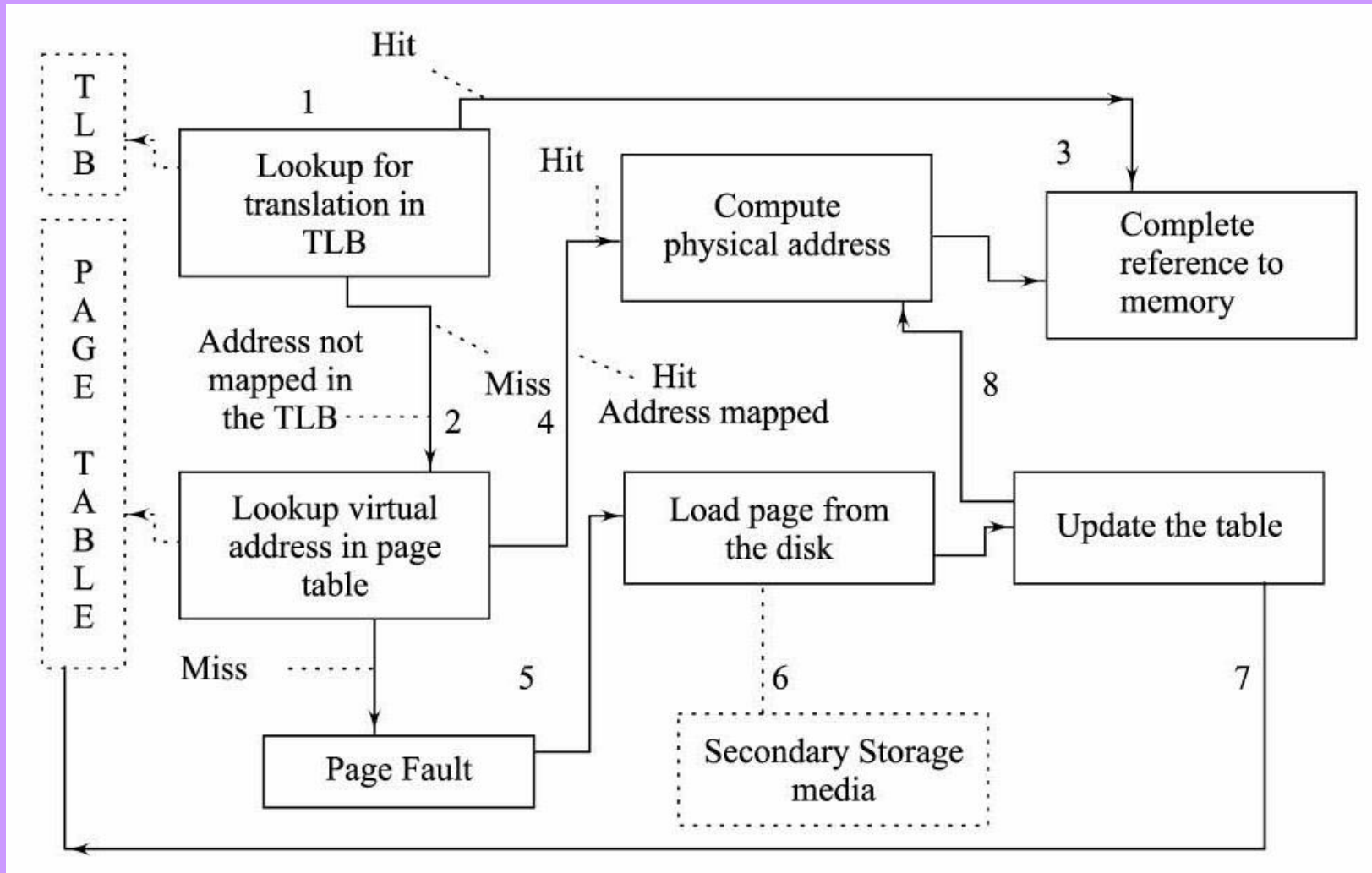
- Assuming no page faults occur while accessing the page table
- TLB misses can usually be resolved in a few hundred cycles
- User program just waits until the TLB miss has been resolved

TLB miss, and the page not mapped

- The system accesses the page table, determines that the address is not mapped, and a page fault occurs

Operating System Handling of the Page Faults

Address translation process in cases when page fault may or may not occur



Action on page fault

- The operating system then loads the page's data from the disk in the same manner as a virtual memory system that does not contain a TLB
- Some OS can switch to another process when the page fault resolution takes a long time

Operating System Handling of the Page Faults

- TLB misses and page faults are handled very differently
- The difference in the amount of time it takes to resolve each event
- TLB misses generally take a relatively short time to resolve and the user program just waits until the TLB miss has been resolved

Page faults

- Require accessing the disk to fetch the page
- Accessing a hard disk typically takes several milliseconds, an amount of time that is comparable to the amount of time that the operating system lets a program run before giving another program access to the processor

Context switch to another process or thread

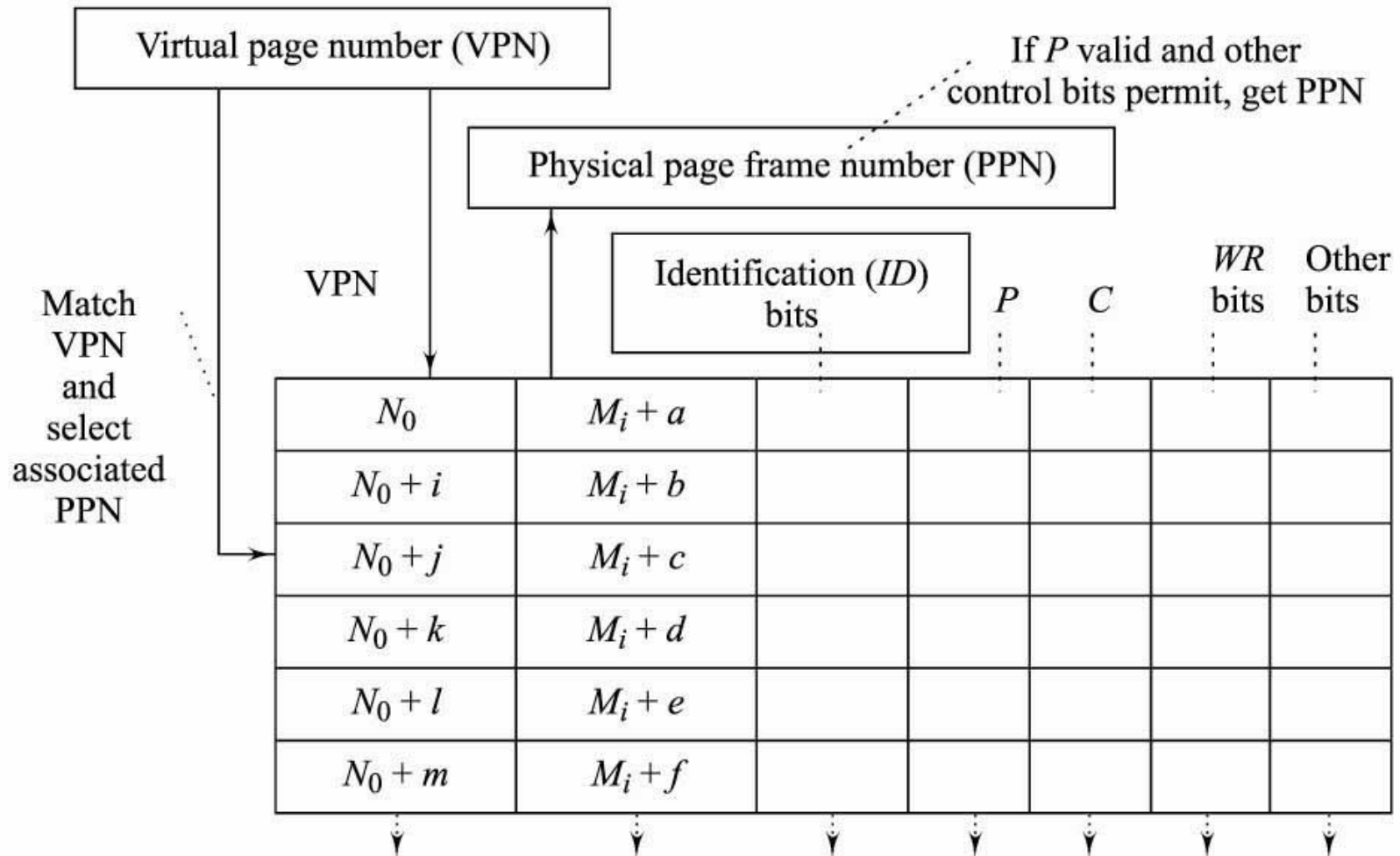
- Switch to process, which is concurrently being processed
- Many operating systems switch programs (processes or threads) 60–100 times per second, allowing each program to execute for 16.7 to 10 ms between context switches

TLB organisation

TLB entry with VPN and PPN with P, C and WR bits

- VPN
- PPN
- P presence bit
- C change indication bit
- WR write protect bit
- Process ID in some systems
- Other control bits

TLB entries with VPN and PPN with P, C and WR bits



TLB organisation

- Similar fashion to caches
- Having an associativity and a number of sets
- Example— a 128-entry, 4-way set-associative TLB would have 32 sets, each containing 4 entries in each set

TLB organisation

- TLB sizes usually described in terms of the number of entries, or translations, contained in the TLB
- Amount of space taken up by each entry mostly irrelevant to the performance of the system

TLB Size and Number of entries

- TLBs typically much smaller than a system's cache(s)
- A relatively small number of TLB entries to describe the working set of a program
- Example— TLBs with 128 entries common in processors built in the mid-1990s that had 32 to 64 kB of first-level cache and 4 kB pages

Bits in an entry in a TLB

- TLBs typically contain many more bits than would be required to describe the data contained in the cache, because it is desirable for the TLB to contain translations for data that resides in the main memory as well as in the cache
- Each entry in a TLB refers to the much more data (all bytes of the page frame) than a cache line

Example

- A given processor has 46-bit and 32-bit virtual and physical addresses
- The page size is 4 kB and the processor's TLB has 256 entries
- 8-way set-associative
- Find many bits are required for each TLB entry

Solution

- Number of entries = 256 entries in the TLB
- Total storage is $256 \times$ the size of each entry
- Each entry needs to contain a valid bit, a change (dirty) bit, the PPN of the page, and the VPN minus the number of bits used to select the set in the TLB

Solution

- With 32-bit addresses and 4 kB pages, the VPN and PPN are $(46 - 12)$ and $(32 - 12) = 34$ and 20 bits each
- With 256 entries and 8-way set-associativity, there are $(256 \div 8) = 32$ sets in the TLB
- So 5 bits of the VPN are used to select a set
- Each TLB entry is of $(34 - 5 + 20 + 2) = 51$ bits

Summary

We learnt

- A page table entry access increases the latency for a memory reference and therefore TLBs used
- Action on TLB miss
- Action on TLB miss as well as page fault
- Understand that how use of translation-lookaside-buffers (TLBs) reduces the access latency many times

We learnt

- TLB entry format
- TLB organization
- Develop ability to solve problems relating to TLB hit and miss

End of Lesson 05 on
Translation Lookaside Buffers