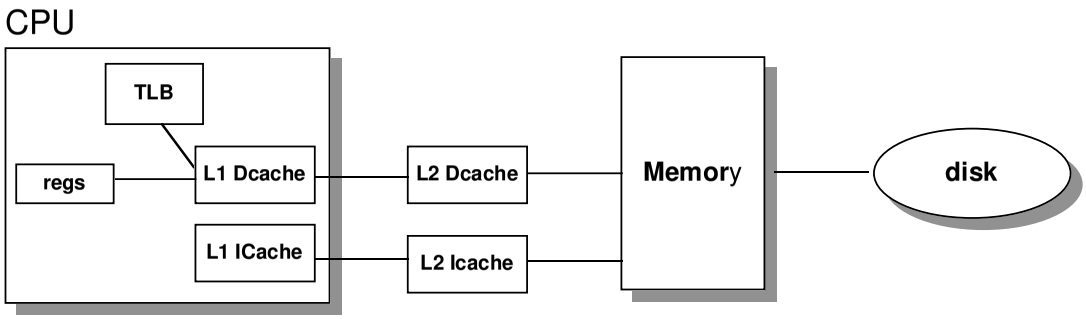
Multilevel Caches and Virtual Memory

April 11, 1996

Topics:

- multilevel caches
- virtual memory basics
- VM address translation

Multi-level caches

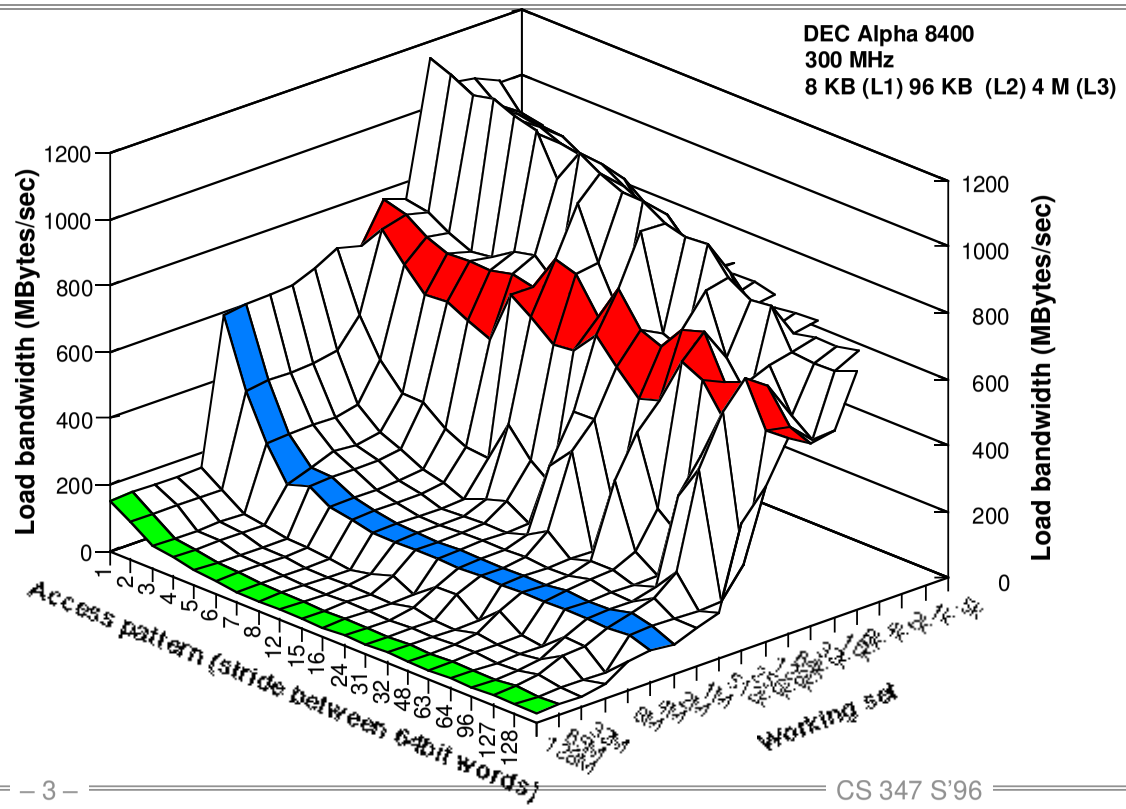


size:	200 B	64 KB	256K SRAM	32 MB DRAM	2 GB
speed:	5 ns	10 ns	10 ns	100 ns	10 ms
$/Mbyte:			$256/MB	$30/MB	$0.30/MB
block size:	4 B	8 B	64 KB	4 KB	

larger, slower, cheaper

larger block size, higher associativity, more likely to write back

The memory mountain range



Virtual memory aims

Overcome physical address limitations

- **important but secondary**

Relocation transparency

- **eliminated nasty code and data overlays common in the 60's.**

Allow multitasking

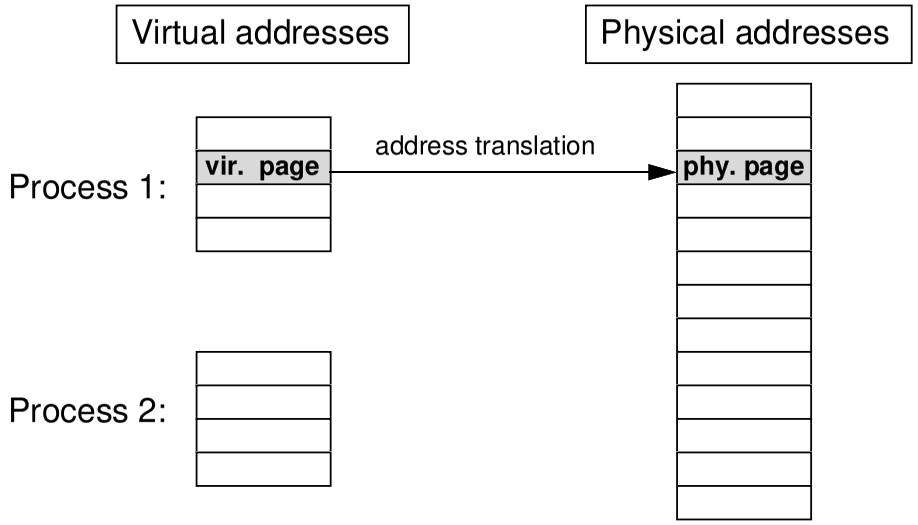
- **multiple users**
- **background tasks**
  - print spooler
  - Internet servers
  - AFS clients

Each executing task called a "process"

- **each process run in private address space of "virtual addresses"**

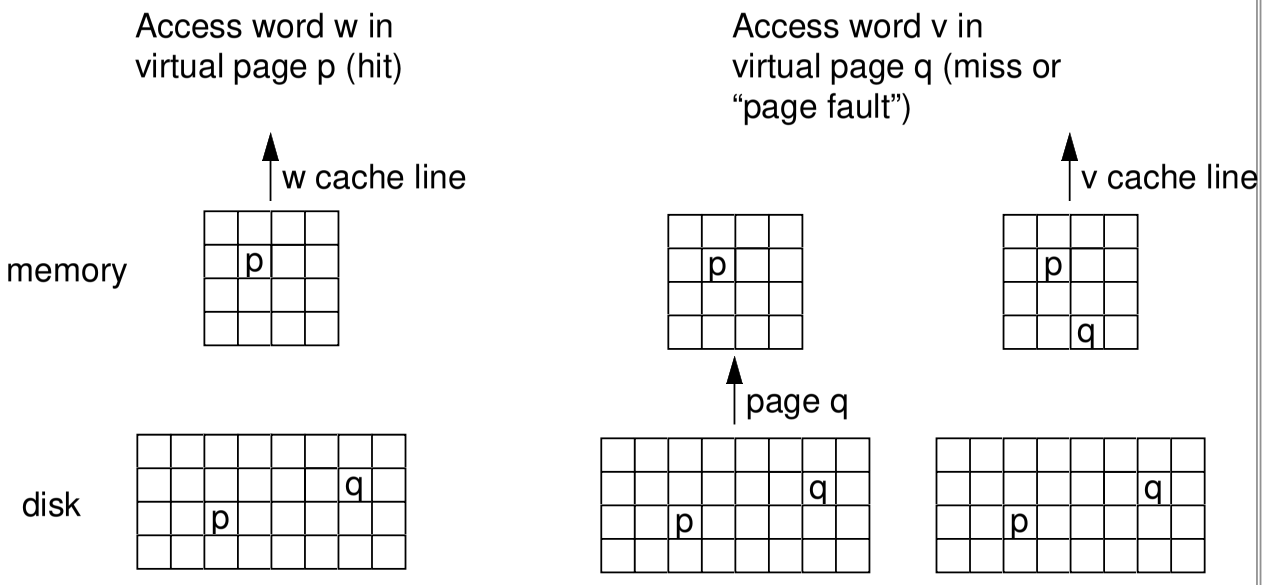
Virtual address spaces

Key idea: virtual and physical address spaces are divided into equal-sized blocks known as “virtual pages” and “physical pages”



What if the virtual address spaces are bigger than the phyiscal address space?

VM as another level in the hierarchy



VM design issues

Everything driven by enormous cost of misses:

- hundreds of thousands of clocks.
- vs units or tens of clocks for cache misses.
- disks are high latency, hand bandwidth devices
- disk performance: 10 ms access time, 10 MBytes/sec transfer rate

Large block sizes:

- 4KBytes - 16 KBytes are typical
- amortize high access time
- reduce miss rate by exploiting spatial locality

VM design issues

Fully associative page placement:

- eliminates conflict misses
- every miss is a killer, so worth the lower hit time

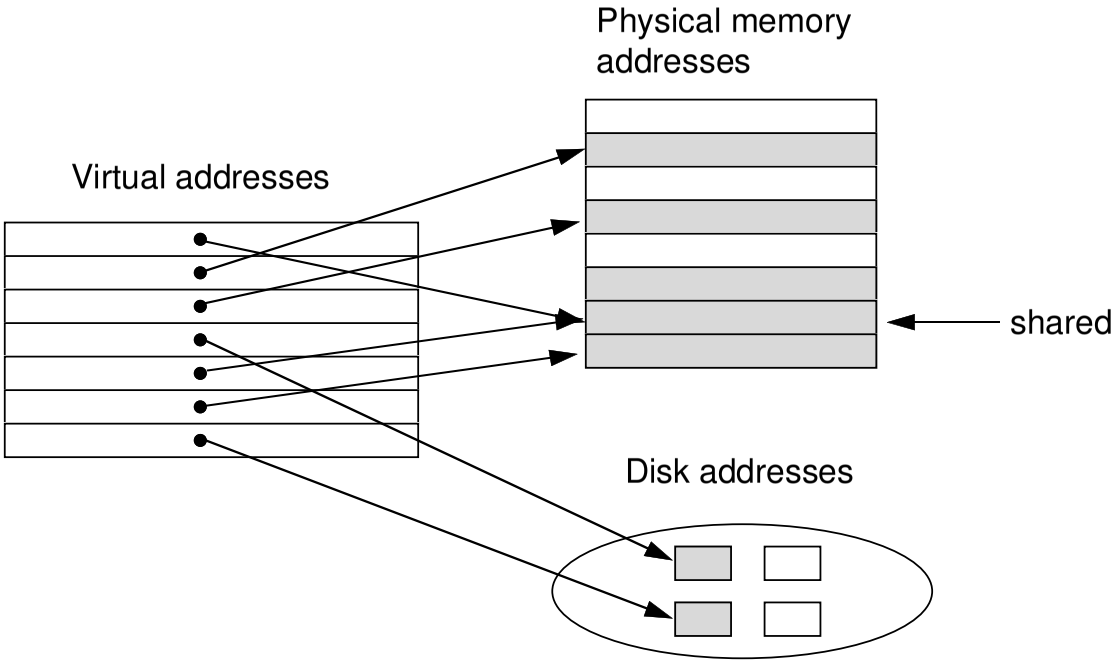
Use smart replacement algorithms

- handle misses in software
- miss penalty is so high anyway, no reason to handle in hardware
- small improvements pay big dividends

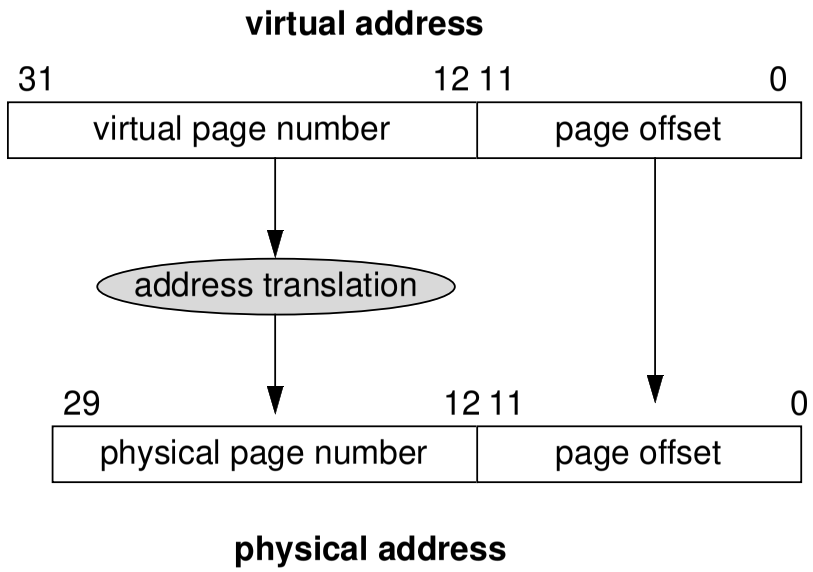
Write back only:

- disk access too slow to afford write through + write buffer

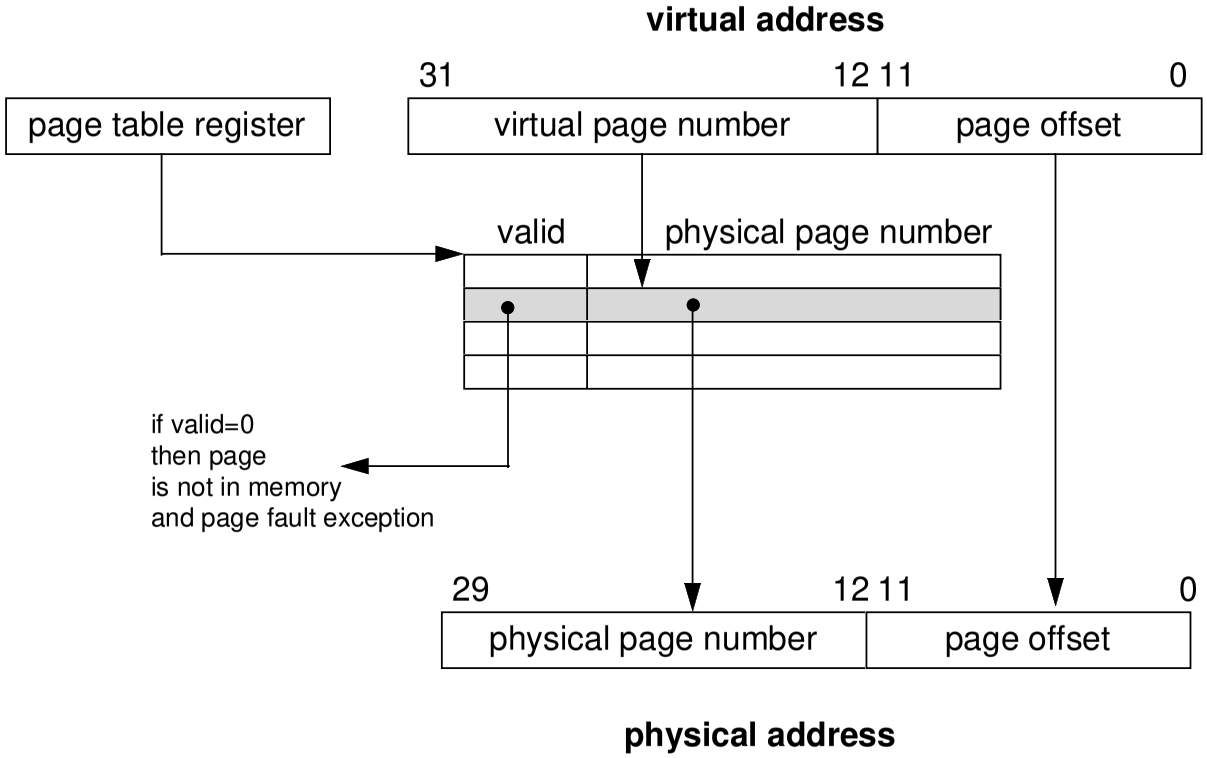
Virtual memory with paging



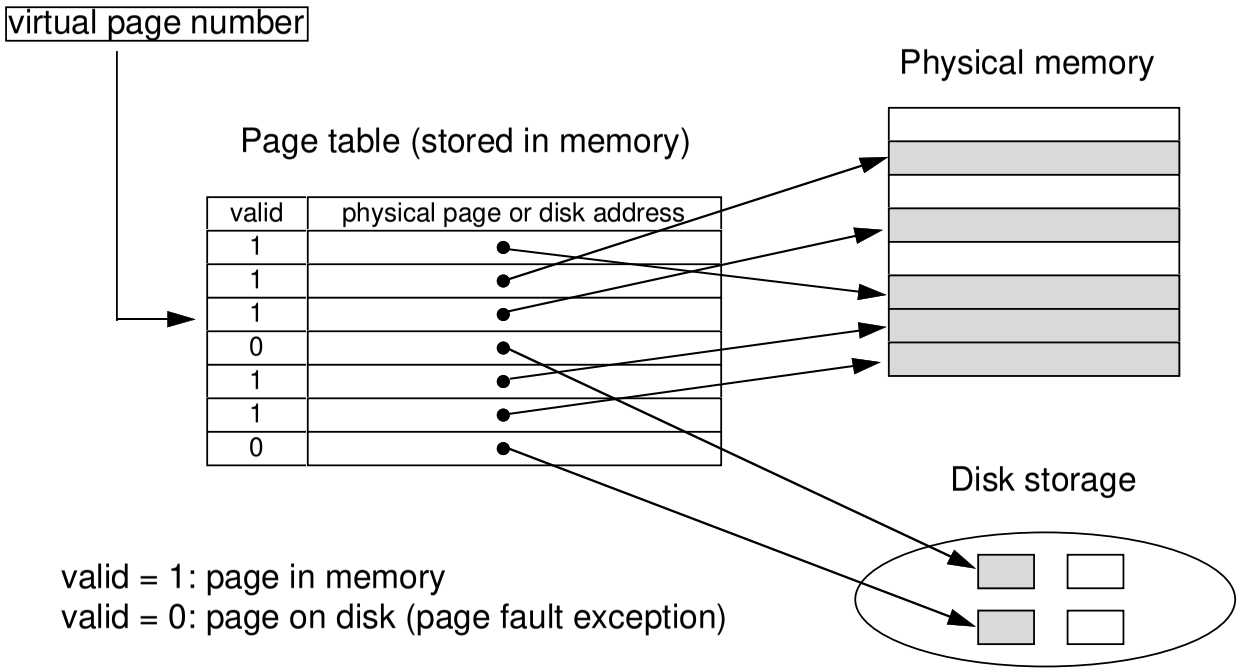
Address translation



Address translation with a page table

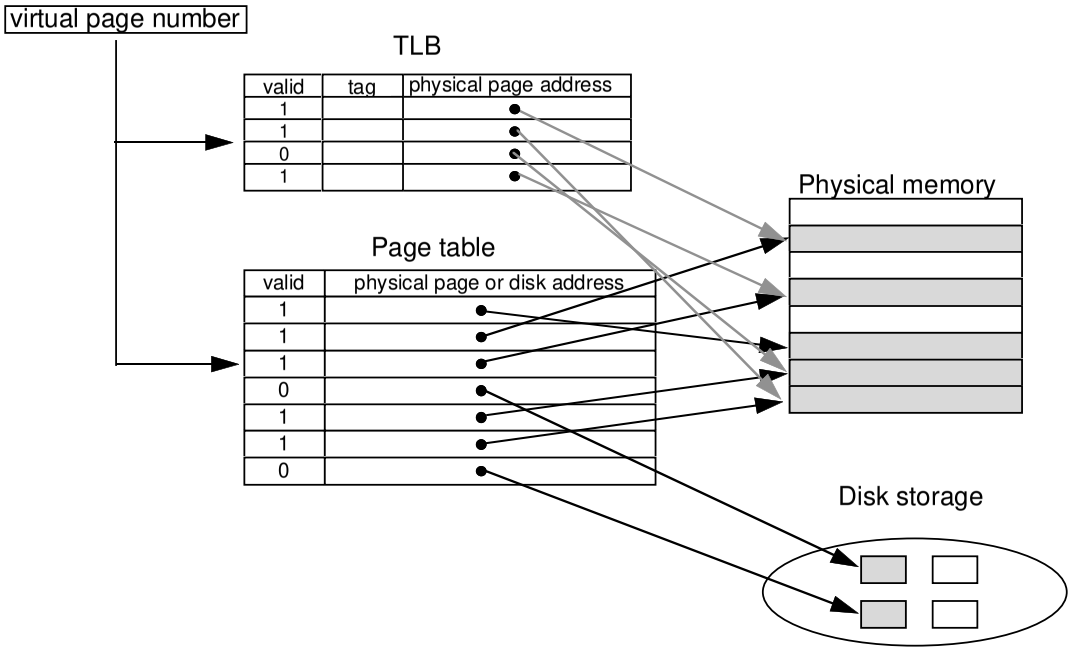
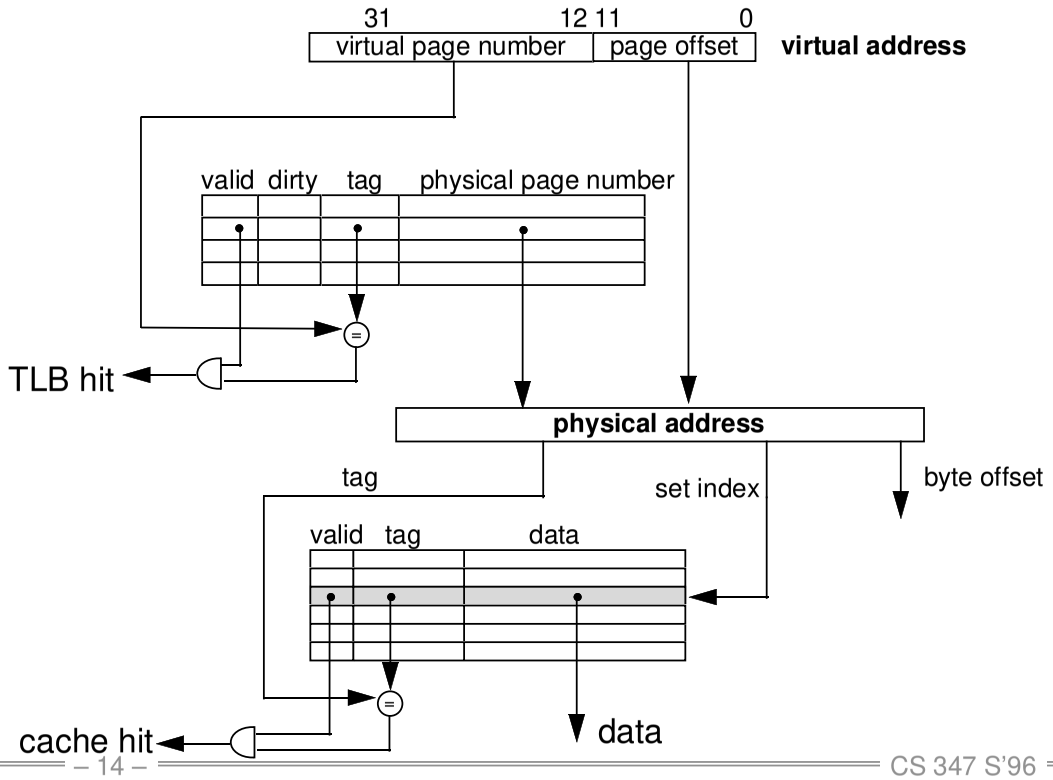


Page table summary



Translation lookaside buffer (TLB)

TLB is an associative cache of virtual to physical mappings for the page table.



Address translation with a TLB