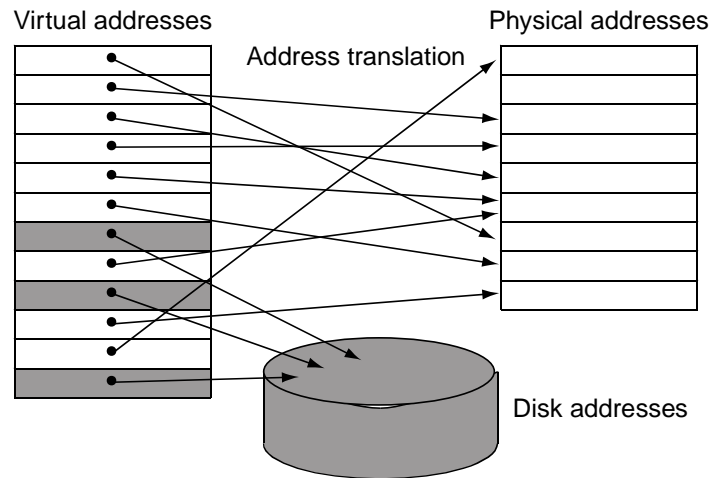# Virtual Memory

Instructor: Dmitri A. Gusev

Fall 2007

CS 502: Computers and Communications Technology

Lecture 11, October 10, 2007

# Virtual Memory

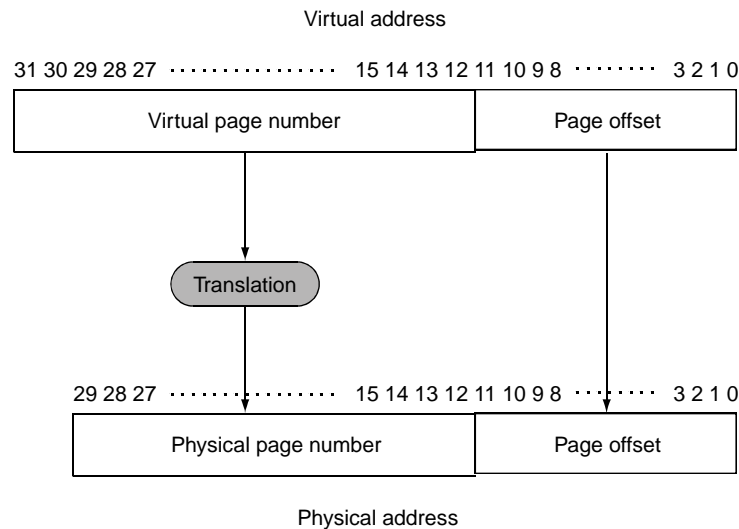- Main memory can act as a cache for the secondary storage (disk)

Virtual addresses

Physical addresses

Address translation
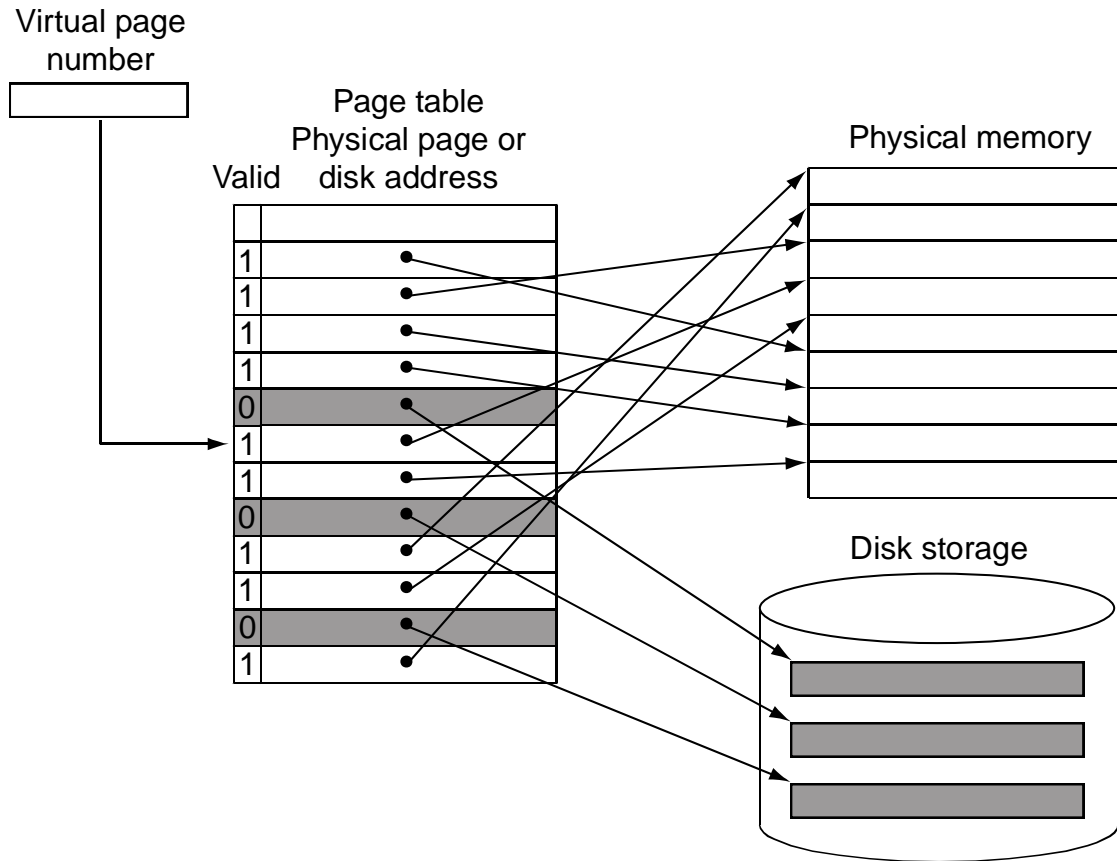
Disk addresses

Advantages:
- illusion of having more physical memory
- program relocation
- protection

# Pages: virtual memory blocks

- Page faults: the data is not in memory, retrieve it from disk
    - huge miss penalty, thus pages should be fairly large (e.g., 4KB)
    - reducing page faults is important (LRU is worth the price)
    - can handle the faults in software instead of hardware
    - using write-through is too expensive so we use writeback

Virtual address

31 30 29 28 27 ............... 15 14 13 12 11 10 9 8 ........ 3 2 1 0

| Virtual page number | Page offset |
|---|---|

Translation

29 28 27 ............... 15 14 13 12 11 10 9 8 ......... 3 2 1 0

| Physical page number | Page offset |
|---|---|

Physical address

# Page Tables

Virtual page number

Page table
Physical page or disk address

Valid

Physical memory

Disk storage

| Valid |
|-------|
| 1 |
| 1 |
| 1 |
| 1 |
| 0 |
| 1 |
| 1 |
| 0 |
| 1 |
| 1 |
| 0 |
| 1 |

# Page Tables

Page table register

Virtual address

31 30 29 28 27 ..................... 15 14 13 12 11 10 9 8 ....... 3 2 1 0

| Virtual page number | Page offset |
|---|---|

20

12

Valid        Physical page number

Page table

If 0 then page is not
present in memory

18

29 28 27 ........................... 15 14 13 12 11 10 9 8 ....... 3 2 1 0

| Physical page number | Page offset |
|---|---|

Physical address

# Making Address Translation Fast

- A cache for address translations:  translation lookaside buffer

TLB

Virtual page number

Valid Dirty Ref    Tag    Physical page address

| Valid | Dirty | Ref |
|---|---|---|
| 1 | 0 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |

Physical memory

Page table

Physical page or disk address

Valid Dirty Ref

| Valid | Dirty | Ref |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 0 | 1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

Disk storage
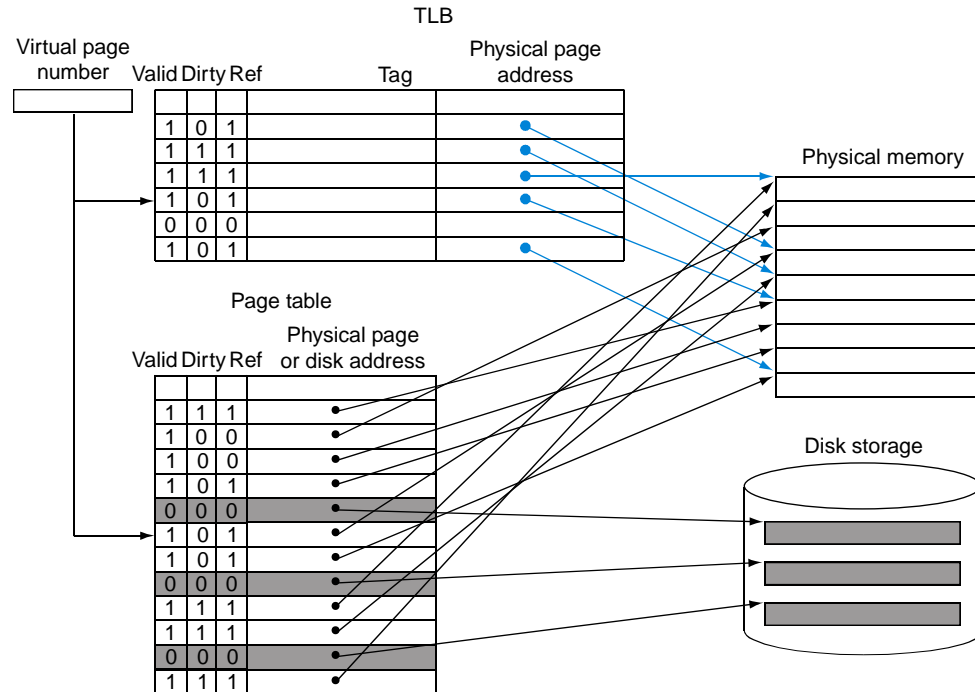
Typical values:    16-512 entries,
miss-rate:  .01% - 1%
miss-penalty:  10 – 100 cycles

# TLBs and caches

Virtual address

TLB access

TLB hit? — No → TLB miss exception

Yes → Physical address

Write?

No → Try to read data from cache

Yes → Write access bit on?

No → Write protection exception

Yes → Try to write data to cache

Cache hit? (read side) — No → Cache miss stall while read block

Yes → Deliver data to the CPU

Cache hit? (write side) — No → Cache miss stall while read block

Yes → Write data into cache, update the dirty bit, and put the data and the address into the write buffer

# TLBs and Caches

Virtual address

31 30 29 ···················· 14 13 12 11 10 9 ····· 3 2 1 0

| Virtual page number | Page offset |

20

12

Valid Dirty    Tag    Physical page number

TLB

TLB hit

20

| Physical page number | Page offset |

Physical address

| Physical address tag | Cache index | Block offset | Byte offset |

18

8

4

2

8

12

Valid    Tag    Data

Cache

Cache hit

32

Data

# Modern Systems

- 

| Characteristic | Intel Pentium P4 | AMD Opteron |
|---|---|---|
| Virtual address | 32 bits | 48 bits |
| Physical address | 36 bits | 40 bits |
| Page size | 4 KB, 2/4 MB | 4 KB, 2/4 MB |
| TLB organization | 1 TLB for instructions and 1 TLB for data<br><br>Both are four-way set associative<br><br>Both use pseudo-LRU replacement<br><br>Both have 128 entries<br><br>TLB misses handled in hardware | 2 TLBs for instructions and 2 TLBs for data<br><br>Both L1 TLBs fully associative, LRU replacement<br><br>Both L2 TLBs are four-way set associativity, round-robin LRU<br><br>Both L1 TLBs have 40 entries<br><br>Both L2 TLBs have 512 entries<br><br>TLB misses handled in hardware |

**FIGURE 7.34   Address translation and TLB hardware for the Intel Pentium P4 and AMD Opteron.** The word size sets the maximum size of the virtual address, but a processor need not use all bits. The physical address size is independent of word size. The P4 has one TLB for instructions and a separate identical TLB for data, while the Opteron has both an L1 TLB and an L2 TLB for instructions and identical L1 and L2 TLBs for data. Both processors provide support for large pages, which are used for things like the operating system or mapping a frame buffer. The large-page scheme avoids using a large number of entries to map a single object that is always present.

| Characteristic | Intel Pentium P4 | AMD Opteron |
|---|---|---|
| L1 cache organization | Split instruction and data caches | Split instruction and data caches |
| L1 cache size | 8 KB for data, 96 KB trace cache for RISC instructions (12K RISC operations) | 64 KB each for instructions/data |
| L1 cache associativity | 4-way set associative | 2-way set associative |
| L1 replacement | Approximated LRU replacement | LRU replacement |
| L1 block size | 64 bytes | 64 bytes |
| L1 write policy | Write-through | Write-back |
| L2 cache organization | Unified (instruction and data) | Unified (instruction and data) |
| L2 cache size | 512 KB | 1024 KB (1 MB) |
| L2 cache associativity | 8-way set associative | 16-way set associative |
| L2 replacement | Approximated LRU replacement | Approximated LRU replacement |
| L2 block size | 128 bytes | 64 bytes |
| L2 write policy | Write-back | Write-back |

**FIGURE 7.35   First-level and second-level caches in the Intel Pentium P4 and AMD Opteron.** The primary caches in the P4 are physically indexed and tagged; for a discussion of the alternatives, see the Elaboration on page 527.

# Modern Systems

- Things are getting complicated!

| MPU | AMD Opteron | Intrinsity FastMATH | Intel Pentium 4 | Intel PXA250 | Sun UltraSPARC IV |
|---|---|---|---|---|---|
| Instruction set architecture | IA-32, AMD64 | MIPS32 | IA-32 | ARM | SPARC v9 |
| Intended application | server | embedded | desktop | low-power embedded | server |
| Die size (mm$^2$) (2004) | 193 | 122 | 217 | | 356 |
| Instructions issued/clock | 3 | 2 | 3 RISC ops | 1 | 4 × 2 |
| Clock rate (2004) | 2.0 GHz | 2.0 GHz | 3.2 GHz | 0.4 GHz | 1.2 GHz |
| Instruction cache | 64 KB, 2-way set associative | 16 KB, direct mapped | 12000 RISC op trace cache (~96 KB) | 32 KB, 32-way set associative | 32 KB, 4-way set associative |
| Latency (clocks) | 3? | 4 | 4 | 1 | 2 |
| Data cache | 64 KB, 2-way set associative | 16 KB, 1-way set associative | 8 KB, 4-way set associative | 32 KB, 32-way set associative | 64 KB, 4-way set associative |
| Latency (clocks) | 3 | 3 | 2 | 1 | 2 |
| TLB entries (I/D/L2 TLB) | 40/40/512/512 | 16 | 128/128 | 32/32 | 128/512 |
| Minimum page size | 4 KB | 4 KB | 4 KB | 1 KB | 8 KB |
| On-chip L2 cache | 1024 KB, 16-way set associative | 1024 KB, 4-way set associative | 512 KB, 8-way set associative | — | — |
| Off-chip L2 cache | — | — | — | — | 16 MB, 2-way set associative |
| Block size (L1/L2, bytes) | 64 | 64 | 64/128 | 32 | 32 |

**FIGURE 7.36  Desktop, embedded, and server microprocessors in 2004.** From a memory hierarchy perspective, the primary differences between categories is the L2 cache. There is no L2 cache for the low-power embedded, a large on-chip L2 for the embedded and desktop, and 16 MB off chip for the server. The processor clock rates also vary: 0.4 GHz for low-power embedded, 1 GHz or higher for the rest. Note that UltraSPARC IV has two processors on the chip.

# Some Issues

- Processor speeds continue to increase very fast
  — much faster than either DRAM or disk access times
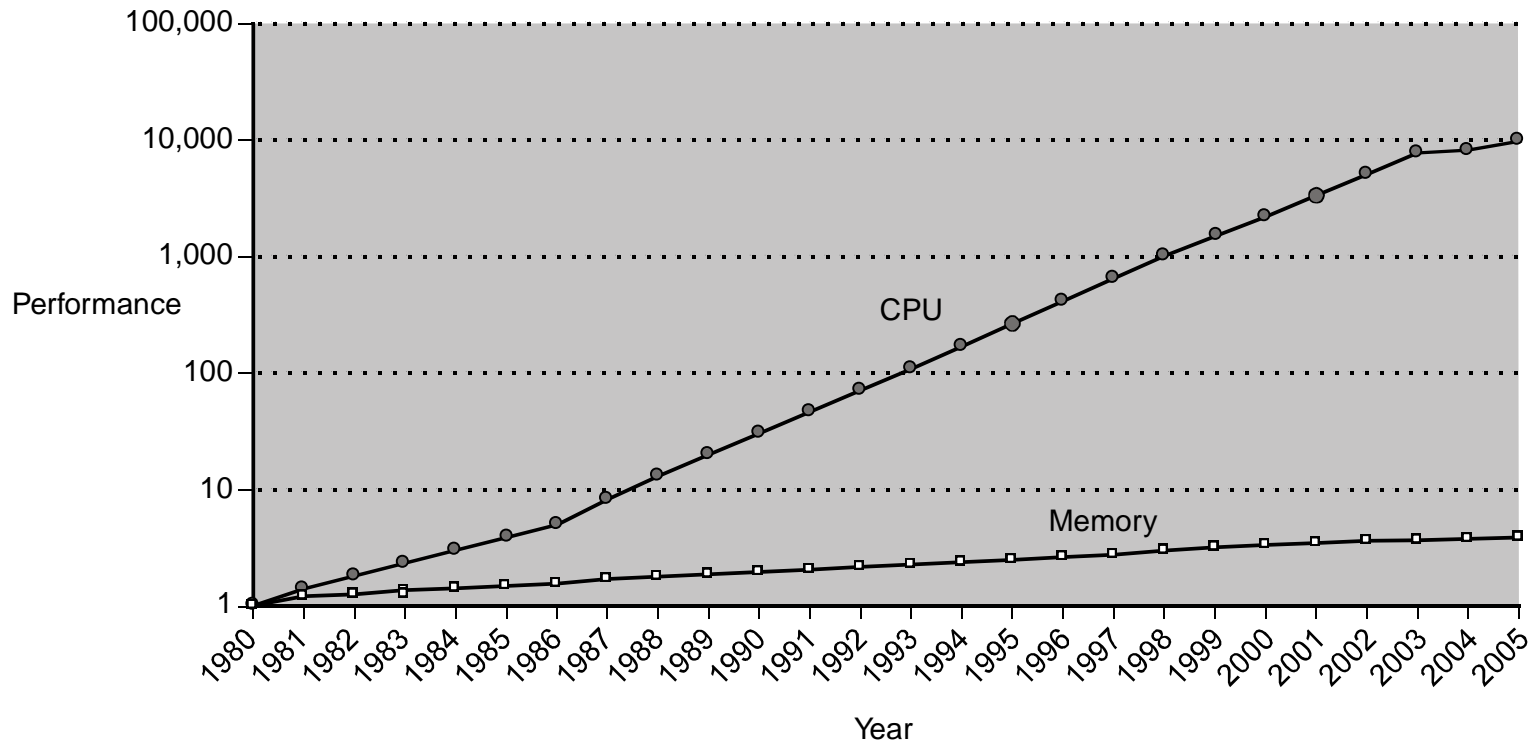


- Design challenge:  dealing with this growing disparity
  – Prefetching?  3rd level caches and more?  Memory design?