Computer Architecture

Appendix B - Virtual Memory

Virtual memory

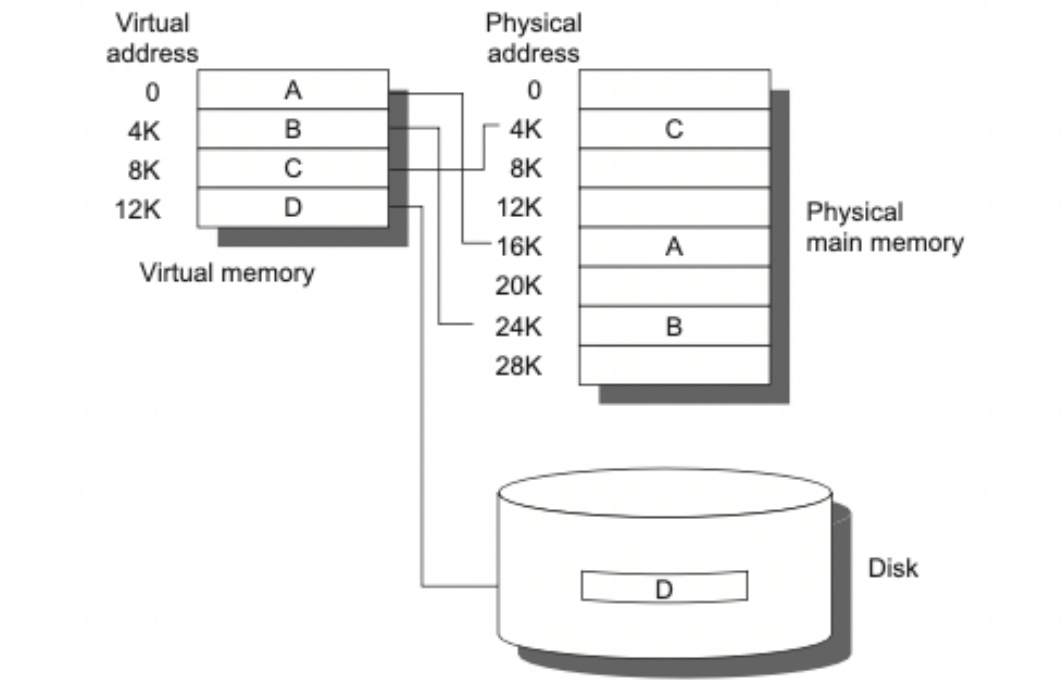

Figure B.19 The logical program in its contiguous virtual address space is shown on the left. It consists of four pages, A, B, C, and D. The actual location of three of the blocks is in physical main memory and the other is located on the disk.

Parameter	First-level cache	Virtual memory
Block (page) size	16–128 bytes	4096–65,536 bytes
Hit time	1–3 clock cycles	100–200 clock cycles
Miss penalty	8–200 clock cycles	1,000,000–10,000,000 clock cycles
(access time)	(6–160 clock cycles)	(800,000–8,000,000 clock cycles)
(transfer time)	(2–40 clock cycles)	(200,000–2,000,000 clock cycles)
Miss rate	0.1%–10%	0.00001%–0.001%
Address mapping	25–45-bit physical address to 14–20-bit cache address	32–64-bit virtual address to 25–45-bit physical address

Figure B.20 Typical ranges of parameters for caches and virtual memory. Virtual memory parameters represent increases of 10–1,000,000 times over cache parameters. Usually, first-level caches contain at most 1 MiB of data, whereas physical memory contains 256 MiB to 1 TB.

Differences between Caches and Virtual Memory

- Replacement on cache misses is primarily controlled by hardware.
- Virtual memory replacement is primarily controlled by software (the O/S).
- The “miss penalty” for VM is very long, which means:
  - It is important to make a good decision on which block (page) to replace, so O/S uses sophisticated algorithms to determine replacement, but
  - O/S can take some time to make a good decision without significantly increasing overall miss penalty
- Size of VM is determined by address space of processor, whereas cache size is independent of processor memory size

Paging vs Segmentation

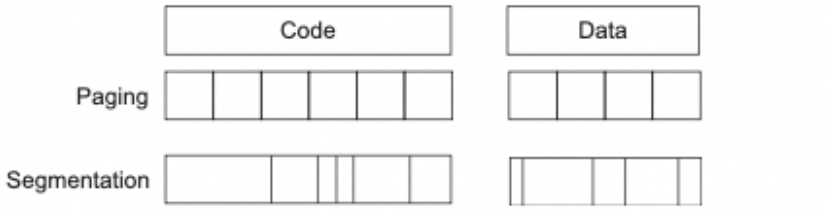

Figure B.21 Example of how paging and segmentation divide a program.

	Page	Segment
Words per address	One	Two (segment and offset)
Programmer visible?	Invisible to application programmer	May be visible to application programmer
Replacing a block	Trivial (all blocks are the same size)	Difficult (must find contiguous, variable-size, unused portion of main memory)
Memory use inefficiency	Internal fragmentation (unused portion of page)	External fragmentation (unused pieces of main memory)
Efficient disk traffic	Yes (adjust page size to balance access time and transfer time)	Not always (small segments may transfer just a few bytes)

Figure B.22 **Paging versus segmentation.** Both can waste memory, depending on the

Four Memory Hierarchy Questions

Q1: Where can a block be placed in the upper level? (*block placement*)

Q2: How is a block found if it is in the upper level? (*block identification*)

Q3: Which block should be replaced on a miss? (*block replacement*)

Q4: What happens on a write? (*write strategy*)

“Upper Level” —> Main Memory

Q1 - Where can a block be placed in main memory?

ANS - Main memory is fully associative

Q2 - How is a block found if it is main memory?

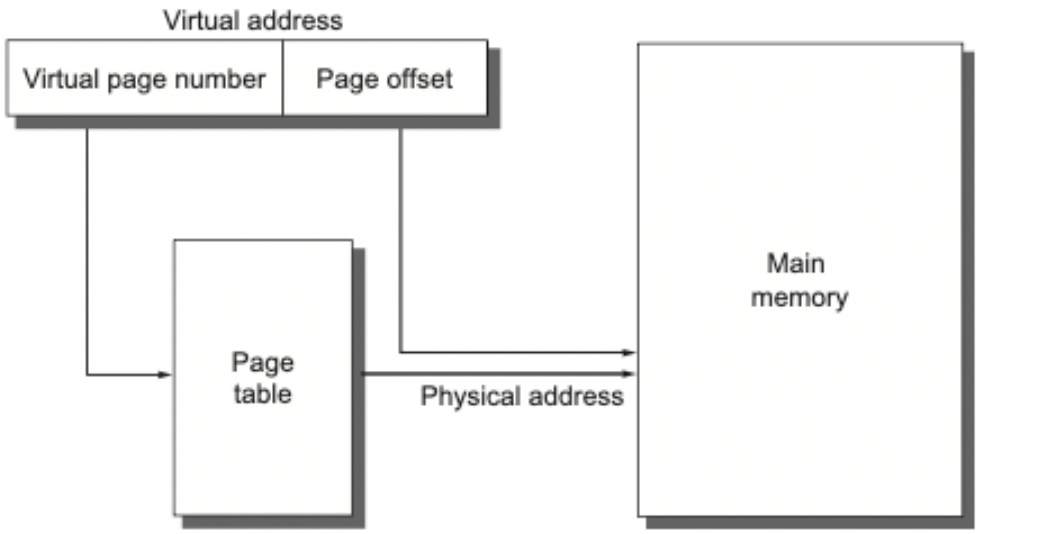

Figure B.23 The mapping of a virtual address to a physical address via a page table.

Q3 - Which block should be replaced on a Virtual memory miss?

- Due to the long miss penalty, O/S uses an LRU replacement policy
- Hardware helps by implementing:
  - Reference bit
  - Use bits

Q4 - What happens on a write?

- Always write-back
- Dirty bit

Translation Lookaside Buffer (TLB)

Connection between hardware and software

- TLB serves as a “cache” for page table

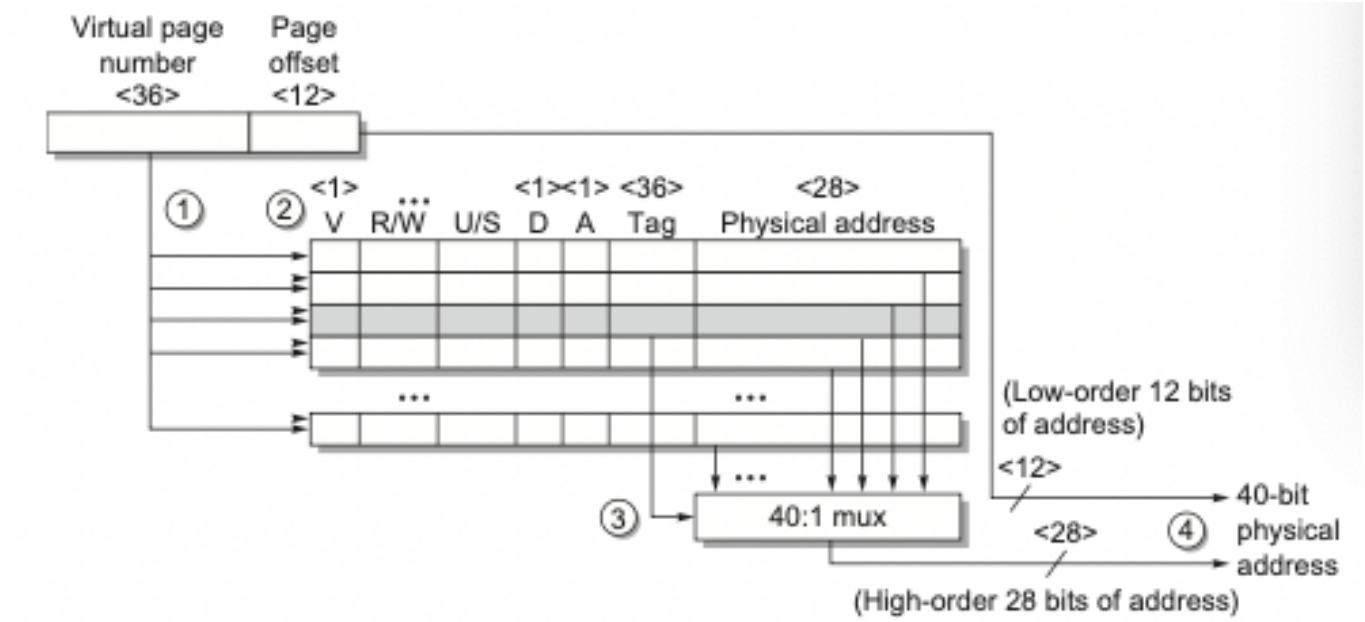

Figure B.24 Operation of the Opteron data TLB during address translation. The four

Page Table Fields

- V (Valid) or Present Bit - indicates if page number is mapped to a “frame” in main memory
- R/W - (Read/Write) - determines permissions for page
- U/S - (User/Supervisor) - specifies type of page
- D (Dirty) - specifies if page has been written to
- A (Accessed) - set if page has been recently accessed
- E (Execute) - Specifies that this page contains executable code
- Tag - Page number that this entry maps
- Frame No - Physical Address where this page is in physical memory

Virtual Memory Operation

1. CPU provides Virtual Address, containing Page Number and Page Offset.
2. Page number is compared with tag in TLB
3. If a match, check permissions. If OK, then:
   main memory address = physical address | page offset
4. If no match, hardware generates a “page fault” (exception).
   O/S performs a paging operation - page is brought into main memory from backing store, updates page table and TLB.
5. CPU resumes operation

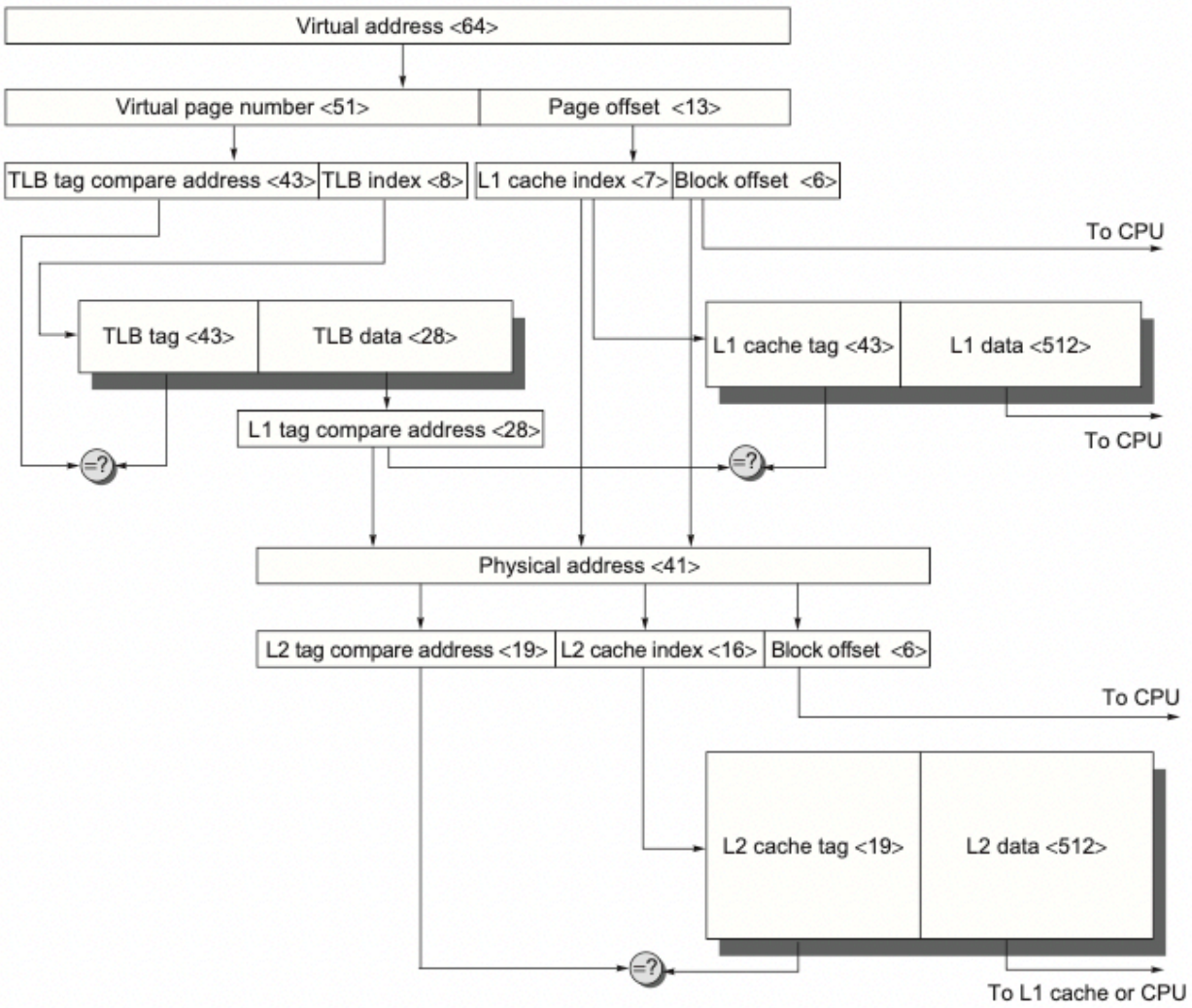

Figure B.25 The overall picture of a hypothetical memory hierarchy going from virtual address to L2 cache

Parameter	Description
Block size	1 PTE (8 bytes)
L1 hit time	1 clock cycle
L2 hit time	7 clock cycles
L1 TLB size	Same for instruction and data TLBs: 40 PTEs per TLBs, with 32 4 KiB pages and 8 for 2 MiB or 4 MiB pages
L2 TLB size	Same for instruction and data TLBs: 512 PTEs of 4 KiB pages
Block selection	LRU
Write strategy	(Not applicable)
L1 block placement	Fully associative
L2 block placement	4-way set associative

Figure B.28 Memory hierarchy parameters of the Opteron L1 and L2 instruction and data TLBs.