

22C:160 High Performance Computer Architecture

Homework 3

Assigned March 10, 06

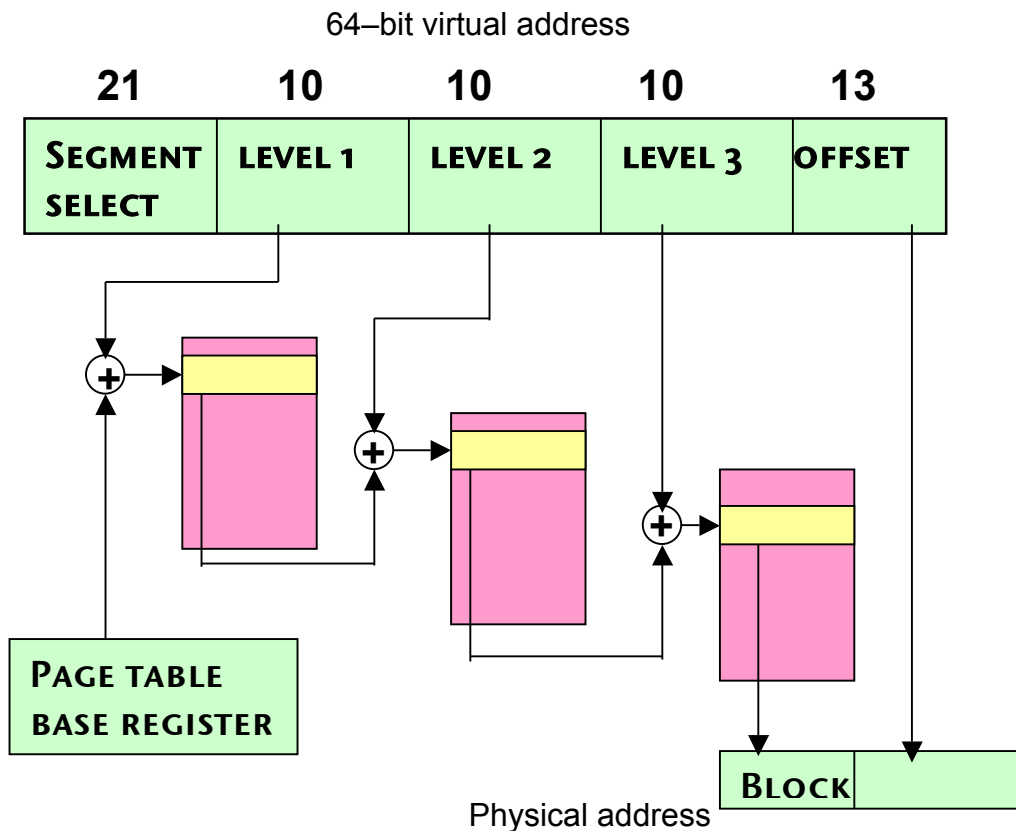
Due March 23, 06

Total Points = 50

There are two questions

Question 1 (25 points)

Linux running on IA-64 architectures use a three-level page table similar to the one discussed for Alpha 21264 in the class. The following diagram outlines the conversion scheme



The page size is 8 KB. The virtual address size is 64 bits, of which the most significant 21 bits are either all 0's (segment 0) or all 1's (segment 1). Each page table entry PTE (also called a page descriptor) has a size of 64 bits of which the most significant 32

bits contain the physical page frame (i.e. block) number of the next level, and the remaining 32-bits contain protection information, not relevant to this exercise. The physical address has a size of 45 bits (32 + 13 bits).

Now, convert the 64-bit virtual address (in segment 1) FFFF FFFF AABB CCDD (hex) into a 45-bit physical address. All the pages required to solve this problem are present in the physical memory, so there will be no page fault. For the purpose of your calculation, you can assume the following partial contents of the page table entries at any level:

Address of PTE	First 32 bits of the PTE
32 0's ## 1111 1111 11 ## 000	32 0's
28 0's ## 1111 ## 0111 0111 10 ## 000	0000 0000 ## 24 1's
1111 ## 28 0's ## 000	32 1's
32 0's ## 1101 0101 01 ## 000	28 0's ## 1111
28 1's ## 0000 ## 000	24 0's ## 1111 1111
All other addresses	32 1's

(The symbol ## denotes concatenation)

Assume that the **page table base register** points to page 0 starting from the physical address 0000 0000 0000 (hex). Clearly show the intermediate steps of your calculation.

Note. The address of a PTE = 32-bit page frame number ## 10 bit L1/L2/L3 field ## 000. The eight bytes X000 thru X111 store a PTE at location X of the page table. Check this out before answering the question.

Question 2 (25 points)

You purchased a computer system with the following features:

1. 95% of all memory accesses are cache hits.
2. Each cache block is two words, and the whole block is read on a cache miss.

3. The processor sends memory references to its cache at the rate of 1 billion per second, of which 75% are reads, and 25% are writes.
4. The memory system can support (i.e. transfer) 1 billion words per second, reads or writes.
5. The bus reads or writes one word per cycle.
6. At any time, 30% of the cache blocks are dirty (i.e. modified).
7. The cache always uses write-allocate on a write miss.

You are considering adding a peripheral to the system, and you want to know, "what fraction of the memory system bandwidth is already used", so that the rest could be used by the peripherals. Calculate **the percentage of the memory system bandwidth** used on the average in the following two cases:

- (a) The cache is write-through.
- (b) The cache is write-back.

Show all the intermediate steps.

Note. In write-through, only the desired word in the main memory is updated, not the entire block of two words.