

## EECS 370 Winter 09 Homework 6 Solutions

### Problem 1 (10 points)

Suppose you are given a system with the following specifications:

- 32-bit virtual address space
- 1GB of physical memory
- 2KB pages
- 4B page entries

Design a hierarchical virtual memory framework.

a) (2 points) How many virtual pages are there?

$2^{(32-11)} = 2^{21}$  virtual pages.

b) (2 points) How many physical pages?

$2^{(30-11)} = 2^{19}$  physical pages

c) (3 points) Assume each page table must fit in a single page. How many levels of page tables are required? How many entries are in each table?

Each page table entry is  $2^2$  bytes. For a page table to fit in one page, we can have at most:  $2^{11} / 2^2 = 2^9$  entries. Hence each level of page tables can represent at most 9 bits of the address. Because we have 21 bits to represent the virtual page number, and we can only use 9 bits per level, we need three levels of page tables.

d) (3 points) Assume your virtual memory system is optimally designed. (HINT: how many bits should your superpage number represent? Do you want this number to be high or low?) What is the worst-case page table size of this hierarchical configuration? What are the advantages of this hierarchical design over a "flat" design? What are the disadvantages?

Naïve design: 9 bits for superpage, 9 bits for page level 2, 3 bits for page level 3  
 $(1 + 2^9 + 2^{18})$  pages \* 2KB per page = 525.314 MB

Smart design: 3 bits for superpage, 9 bits for page level 2, 9 bits for page level 3  
 $(1 + 2^3 + 2^{12})$  pages \* 2KB per page = 8.210 MB

Hierarchical design takes up significantly less space on average since most addresses are not accessed. However, a lookup takes longer since you must follow 3 pointers to get to a page instead of one.

**Problem 2: Cache Addressing (5 points)**

A high-performance computer has a cache with the following characteristics:

- 50-bit addresses
- byte-addressable memory
- 16 megabyte cache size (1 megabyte =  $2^{20}$  bytes = 1,048,576 bytes)
- 2 kilobyte cache block size (1 kilobyte =  $2^{10}$  bytes = 1,024 bytes)
- 16-way set-associative cache

Address bit numbering: most significant bit is bit 49, least significant bit is bit 0.

a) What range of address bit numbers are used for each of the following?

<u>Field</u>	<u>Number of Bits</u>	<u>Bit Range</u>
Tag	30 bits	Bit20 – Bit49
Set Index	9 bits	Bit19 – Bit11
Block Offset	11 bits	Bit0 – Bit10

16MB cache size =  $2^{24}$  bytes in cache

2KB block size =  $2^{11}$  bytes/block

Block offset field is 11 bits

$(2^{24} \text{ bytes}) / (2^{11} \text{ bytes/block}) = 2^{13}$  blocks in cache

16-way set associative =  $2^4$  blocks/set

$(2^{13} \text{ blocks}) / (2^4 \text{ blocks/set}) = 2^9$  sets

Set index field is 9 bits

$(50 \text{ bits in address}) - (9 \text{ bits for set index}) - (11 \text{ bits for block offset}) = 30 \text{ bits}$

Tag field is 30 bits

b) For address **0x 3 2354 BA38 BE0F**, show the tag, set index, and block offset. You may state your answers in either binary or hexadecimal format.

<u>Field</u>	<u>Binary / Hexadecimal</u>
Tag	<b>0x3235 4BA3 = 11 0010 0011 0101 0100 1011 1010 0011</b>
Set Index	<b>0xF = 0 0000 1111</b>
Block Offset	<b>0x45F = 100 0101 1111</b>

**Problem 3 (10 points)**

Consider the following computer with a CPU, data cache, and memory.

- The system has a 12-bit virtual address size, and the size of a virtual memory page is 256 bytes. The computer has 65536 bytes of physical memory.
- The size of the CPU data cache is 512 bytes, with a block size of 16 bytes. The CPU data cache is 2-way set-associative with LRU replacement, and it is a virtually indexed cache. For a hit, the data cache can be accessed in a single clock cycle. For a miss, a cache block in memory can be read in 10 clock cycles, which includes the time for a TLB access.
- The CPU includes a fully associative TLB with 2 entries and an LRU replacement policy. For a TLB miss, the appropriate page table entry is supplied to the CPU after 25 clock cycles. The system uses a single level page table. Assume the TLB and the data cache are initially empty. The contents of the page table are shown below:

VPN	Valid	PPN
0	0	-
1	1	0xfd
2	1	0x4b
3	0	-
4	1	0x55
5	1	0x32
6	1	0x54
7	0	-
8	0	-
9	1	0xfe
A	1	0xfc
B	0	-
C	1	0xba
D	1	0x01
E	0	-
F	0	-

a) (7 points) Data reads from the following virtual addresses are performed (in the order listed):

0xce2, 0x258, 0xcea, 0x2d4, 0x15b, 0x241, 0xcee.

Please complete the table below to show the following for each read:

- Physical address (show it in hex)
- Data cache hit/miss (write HIT or MISS as appropriate)
- TLB hit/miss (write HIT or MISS as appropriate, or write NA if the TLB is not accessed)

Virtual Address	Physical address	Data cache hit/miss	TLB hit/miss/NA
0xce2			
0x258			
0xcea			
0x2d4			
0x15b			
0x241			
0xcee			

b) (3 points) A program running on the above described computer has a data cache miss rate of 15% and a data TLB miss rate of 10%. What is the average memory access latency for this program for data accesses? Please show your work and clearly state any assumptions you make.

## SOLUTIONS

a)

Virtual Address	Physical address	Data cache hit/miss	TLB hit/miss/NA
0xce2	0cbae2	MISS	MISS
0x258	0x4b58	MISS	MISS
0xcea	0xbaea	HIT	NA
0x2d4	0x4bd4	MISS	HIT
0x15b	0xfd5b	MISS	MISS
0x241	0x4b41	MISS	HIT
0xcee	0xbaee	HIT	NA

b)

Remember the cache is virtually indexed

1 = normal operation (data hit in cache)

0.15 = data cache miss rate, so 0.15 of the time you'll have misses

0.90 = of those misses, 90% will hit in the TLB, thus incurring a total penalty of 10

for just the cache miss part

0.15 = data cache miss rate

0.10 = of those misses, 10% are misses, thus incurring double penalty of 25 cycles for

TLB miss and 10 cycles for cache miss.

Therefore:

ANSWER:  $1 + 0.15 * 0.90 * 10 + 0.15 * 0.10 * (25 + 10) = 2.875$  cycles