

Review Session 7

Fri. Mar 2, 2:15 pm – 3:05 pm
Thornton 102

1. Agenda

- Example problems on cache, exceptions & virtual memory
- Homework 4 hints

1

2. Example Problems

Virtual cache and exceptions:

A cache is physically tagged if the tag portion is extracted from the physical address. Similarly a cache is virtually tagged if the tag is from the virtual address. Consider a system that supports virtual memory with paging and each process on the system has its own virtual memory address space.

2

a) What is the main advantage of using a virtually tagged cache?

b) Name two disadvantages of a virtually tagged cache.

c) Compared to a physically tagged cache, does a virtually tagged cache require additional field(s), for correctness or performance? If so, what field(s) is needed?

d) For the following types of exceptions, state whether the process should resume execution after the exception. If execution should be resumed, state whether it should resume at the instruction that caused the exception or the instruction after.

Exception Types	Resume?	Re-execute faulting instruction?
Misaligned address		
Page fault		
Memory protection violation		
Syscall		

Consider a 5-stage pipelined MIPS processor in which branches are resolved in the ID stage (with a dedicated comparator) and there is a branch delay slot to eliminate control hazards. Suppose we execute the following code sequence on this processor:

```
        or    $3, $1, $2
        or    $6, $4, $5
        beq   $3, $6, there
        lw    $7, 0($3)
        sll   $1, $7, 2
        :
        :
there:  add   $9, $7, $8
        :
        :
```

7

e) If the branch is not taken, what instructions should be executed after the branch? What if the branch is taken?

8

- f) Suppose the `lw` instruction causes a TLB miss. What should the exception PC (EPC) be set to? Is that sufficient to allow the program to resume execution correctly? If not, what other information is necessary? (Hint: Compare the code sequence with and without TLB miss exception.)

9

3. Homework hints

Problem 1:

- Equation for write through cache:

$$\begin{aligned} \text{AMAT}_{\text{ThisLevel}} &= \text{HitRate}_{\text{ThisLevel}} \times \text{HitTime}_{\text{ThisLevel}} + \\ &\quad \text{MissRate}_{\text{ThisLevel}} (\text{HitTime}_{\text{ThisLevel}} + \text{AMAT}_{\text{NextLevel}}) \\ &= \text{HitTime}_{\text{ThisLevel}} (\text{HitRate}_{\text{ThisLevel}} + \text{MissRate}_{\text{ThisLevel}}) + \\ &\quad \text{MissRate}_{\text{ThisLevel}} (\text{AMAT}_{\text{NextLevel}}) \\ &= \text{HitTime}_{\text{ThisLevel}} + \text{MissRate}_{\text{ThisLevel}} (\text{AMAT}_{\text{NextLevel}}) \end{aligned}$$

- Hit time is 1 cycle if unspecified, but not 0 seconds!
- Perfect write through buffer i.e. no matter how much data is fed into the buffer by the CPU, it would never result in a stall.

- Equation for write back cache:

$$AMAT_{\text{ThisLevel}} = \text{HitTime}_{\text{ThisLevel}} + \text{MissRate}_{\text{ThisLevel}}(AMAT_{\text{NextLevel}}) + \text{MissRate}_{\text{ThisLevel}} [\text{DirtyRate}_{\text{ThisLevel}} (AMAT_{\text{NextLevel}})]$$

- CPI equation:

$$\begin{aligned} CPI_{\text{overall}} &= CPI_{\text{base}} + \text{InstrReadFreq} \times \text{CyclesPerInstrRead} + \\ &\quad \text{DataReadFreq} \times \text{CyclesPerDataRead} + \\ &\quad \text{DataWriteFreq} \times \text{CyclesPerDataWrite} \\ \text{CyclesPerWhateverAction} &= AMAT_{\text{WhateverAction}} \times \text{ClockFrequency} \end{aligned}$$

Problem 2:

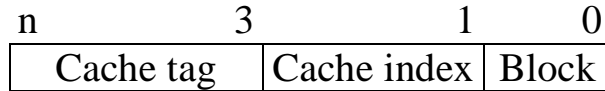
- Part a: consider the relative frequency of hits and misses are encountered.
- Part c: consider what happens when the block size is greater than 1 word e.g. B words in general?
- Part f: what is the relationship between the array and step size values so that in the 2-way set associative case data would compete for the same cache line, but not so with a 4-way set associative cache?

Problem 3

- You need not worry about the cache address part.
- Both virtual and physical addresses have the same page size thus page offset bits.
- Given that the TLB has 2 ways, how many sets (i.e. lines) are there in each way? How many bits are needed to index these sets?
- Remember, just like with caches, set indexing bits are not store anywhere in the TLB!

Problem 4:

- Address breakdown:



<i>Address</i>	<i>Hit or Miss</i>	<i>Address</i>	<i>Hit or Miss</i>
<i>1</i>		<i>17</i>	
<i>15</i>		<i>56</i>	
<i>29</i>		<i>6</i>	
<i>4</i>		<i>1</i>	
<i>56</i>		<i>28</i>	
<i>41</i>		<i>4</i>	
<i>15</i>		<i>20</i>	
<i>18</i>		<i>56</i>	

<i>Index</i>	<i>Sets[1]</i>				<i>Sets[0]</i>			
	<i>Vld</i>	<i>Tag</i>	<i>Blk[1]</i>	<i>Blk[0]</i>	<i>Vld</i>	<i>Tag</i>	<i>Blk[1]</i>	<i>Blk[0]</i>
<i>0</i>								
<i>1</i>								
<i>2</i>								
<i>3</i>								

Problem 5:

- Part a: let N be the number of cycles between each byte of data. 1000 out of the N are used to process the data, the remaining are used on polling.
- Part a: how many polling cycles in N – 1000 cycles?
- Part a: how many times do we need to do this?
- Part b: again let N be the number of cycles between each byte of data.
- How many out of the N are used for interrupt and to process the data? The remainder is used for other tasks between each interrupt.
- How many interrupts do we see?

Problem 6:

- Work out the effective bandwidth of each part of the system in MB/s. Recalling that the system is only as fast as the slowest component, what is the bandwidth of data transfer you should use in your calculation of disk access?

Problem 7:

- Part a: assume that the platter has 2 heads, and that $1 \text{ GB} = 2^{30}$ bytes.
- Part b: overhead refers to both initialization, rotational delay and seek time, so the calculation is for the maximum transfer bandwidth.
- Part c: here the average access time refers to only seek time, controller overhead and rotational delay, excluding the transfer time.
- Part e: needs to be done with both full transfer bandwidth of disk (corresponding to large contiguous data reads) as well as with effective bandwidth when reading 64 kB blocks.
- Part g: think about performance impact and I/O data usage.