CS 465 – Final Review

Fall 2018
Prof. Daniel Menasce

Questions

• What are the types of hazards in a datapath and how each of them can be mitigated?
• State and explain some of the methods used to deal with branch prediction.
Does the following code sequence must stall, can avoid stalls using forwarding only, or can execute without stalling or forwarding?

- lw $t0, 0($t0)
- add $t1, $t0, $t0

Question

- What is the purpose of the pipeline registers?
- What are the names given to these registers?
- Explain how information stored in these registers changes along the pipeline.
- How are these registers used to detect data hazards?
Explain This

Question

• Which of these elements influence the execution time of a program and why?
  – The level 1 cache hit rate
  – The hit rate to the TLB
  – The number of instructions handled per stage
  – The page fault rate
  – The ISA
  – The clock cycle duration
Question

• What are all the levels of a memory system hierarchy?
• How do size, access time, and cost/bit vary along the hierarchy?
• What property of programs is important for the performance of the memory system hierarchy?

Question

• A computer system takes 1,000 days to fail on average and the time to repair is on average one day. What is its availability?
Question

• Explain the Principle of Locality and discuss where it is applied and what it achieves.

Question

• What are the types of cache organization?
• Explain for each: block search, block placement, block replacement?
• Explain the two cache hit on write policies: write-through and write back (include write buffer)
• Explain approaches used for cache miss on write policies
Question

• Consider that a physical address is 32 bits and that a direct mapped cache has 1024 entries. Each block in the cache has 4 32-bit words. How many bits are used for the tag, cache index, and byte offset in the address?

Question

• Explain the three sources of misses known as the 3 C’s: compulsory, capacity, and conflict.
Question

• A direct mapped cache has 128 entries and each entry holds one block of 64 bits. What is the cache entry for a block that is stored in memory starting at byte 1152?

Question

• A 4-way set associative cache has 128 sets and each entry holds four blocks of 64 bits. What is the set entry for a block that is stored in memory starting at byte 1152?
Question

• Which of the following are true?
  – A. All cores share a single L1 cache
  – B. All cores share a single L2 cache
  – C. Each core has its own page table register
  – D. All cores share a single L3 cache
  – E. The TLB is stored in main memory
  – F. The page table needs to be accessed for every memory access

Explain This
Explain This

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

Valid

20

Physical page number

If 0 then page is not present in memory

18

Physical page number Page offset

Physical address

Explain This

Virtual page number

Valid Dirty Ref Tag

0 0 0

1 0 1

1 0 0

1 0 1

0 0 0

1 0 1

1 0 1

0 0 0

1 1 1

1 1 1

1 1 1

0 0 0

1 1 1

TLB

Physical page address

Physical memory

Page table

Valid Dirty Ref

Physical page or disk address

Disk storage

17

18
Consider a 4-way set associative cache with blocks of 4 words of 4 bytes each. The cache size is 256 K bytes. How many bits in the address are used for the tag, cache index, and block/word/byte?
Question

• A machine supports virtual memory and page sizes are 4 Kbytes long. A physical address is 32 bits long. How many page frames are there in main memory?
• Suppose that a virtual address is 34 bits long.
  – What is size of the virtual address space in pages?
  – How many entries are there in the page table?
  – Estimate the size of the page table in MB.

Question

• Consider the following table.

<table>
<thead>
<tr>
<th>Memory element</th>
<th>Access time</th>
<th>Hit rate</th>
<th>Miss Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 Cache</td>
<td>1 cycle</td>
<td>85%</td>
<td>5 cycles</td>
</tr>
<tr>
<td>L2 cache</td>
<td>5 cycles</td>
<td>99%</td>
<td>100 cycles</td>
</tr>
<tr>
<td>Memory</td>
<td>100 cycles</td>
<td>100%</td>
<td></td>
</tr>
</tbody>
</table>

– Compute the Average Memory Access Time in cycles
Question

• How is write on hit handled in virtual memory (write-through or write-back) and why?
• What is the typical page replacement policy used in virtual memory and how it is implemented?

Consider a VM system that uses an approximate LRU replacement scheme for main memory whose content is

<table>
<thead>
<tr>
<th>Valid bit</th>
<th>Reference bit</th>
<th>Dirty bit</th>
<th>Page frame number</th>
<th>Disk address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>-----</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-----</td>
<td>PF:3654</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>---</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>---</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>---</td>
</tr>
</tbody>
</table>

• Does a reference to virtual page 1 cause a page fault?
• If main memory only has space for 4 pages, which page would you replace to bring virtual page 1 into memory?
Question

• A program has to execute \( A \) arithmetic operations that take \( t \) seconds each. Ten percent of them cannot be executed in parallel. Give an expression for the program’s execution time when executed on \( p \) processors. What is the minimum possible execution time of this program?

Question

• What are vector machines? What kind of instructions they have in addition to conventional instructions? Give examples. What kind of applications are they useful for?