EEL 4709C: Computer Design Midterm Review

Fall, 2009

Contents

- Introduction on Computer Design
- Performance Evaluation
- Processor Architecture
- Memory hierarchy

Introduction on Computer Design

- Major components in a computer
- The Von Newman Computer model
- Processor/memory performance gap
- Moore's law
- Computer architecture vs computer organization

Performance Evaluation

- Response time vs. throughput
- CPU time and its calculation
- Performance evaluation
  - Total/average execution time
  - Weight execution time
  - Normalized geometric mean
- Amdhal's law
  - speedup

Processor Architecture

- Processor structure and instruction cycles
  - Program and instruction
  - Instruction execution cycles
- Pipelining basic
  - Pipeline principles
  - Pipeline hazards
- Interrupt
- Buses

Processor Architecture

- Processor structure and instruction cycles
- Pipelining basic
- Interrupt
  - What/why/how
  - Multiple interrupts
- Buses
  - What
  - Data/Address/Control buses
  - Single buses vs bus hierarchy
  - Shared vs dedicated bus
  - Centralized vs distributed arbitration

Memory Hierarchy

- What/why
  - The principle of locality
- Cache and cache design
  - Mapping strategies and address structures
  - Replacement algorithm
  - Writing policy
  - Performance of memory system
- Internal memory
  - Types: Volatile/Non-Volatile
  - Types of RAM
  - Types of ROMs

Memory Hierarchy

- External memory
  - Types
  - Total disk access time = $t_s+t_r+t_t$
- Virtual memory
  - What
  - Virtual address to physical address translation
  - Using Translation Lookaside Buffer (TLB) to save the translation results

Questions?

Examples

- Suppose a processor has the following characteristics.
  - 40 bit address bus
  - 64KB direct mapped first level cache with block size of 32bytes
  - 2MB 4way set associative second level cache with block size of 256 bytes
- For physical memory address 0x20689832
- Values of tag, index, block offset for the first level and second level cache?

Solution

- 1st level:
  - Block size of 32 = 2^5 → block offset 5 bit
  - Total number of blocks = 64Kb/32 = 2^{11} → index size = 11
  - Tag size = 40 – 11 – 5 = 26
  - Address 0x0020689832 (hex)

  = 0b 0000 0000 0010 0000 0110 1000 1001 1000 0011 0010 (binary)
  - block offset = 0b 10010 = 0x 12
    index = 0b 100 1100 0001 = 0x 4C1
    Tag = 0b 0000 0000 0010 0000 0110 1000
    = 0x 002068

Solution

- 2nd level:
  - Block size of 256 = 2^8 → block offset 8 bit
  - Total number of blocks = 2Mb/256 = 2^{11} → set index size = 11
  - Tag size = 40 – 11 – 8 = 21
  - Address 0x0020689832 (hex)

  = 0b 0000 0000 0010 0000 0110 1000 1001 1000 0011 0010 (binary)
  - block offset = 0b 0011 0010 = 0x 32
    index = 0b 000 1001 1000 = 0x 098
    Tag = 0b 0 0000 0000 0100 0000 1101
    = 0x 00040D

Examples

- A hypothetical microprocessor runs at 200Mhz, and one instruction takes 4 cycles
- A user program has 1,000,000 instructions
- Two output operations are required and may potentially be overlapped, each for different devices (A and B)
- Output operation on A takes 0.3s, Output operation on B takes 0.1 s
- If interrupts are used, the interrupt service handlers (ISH) for A and B are 10,000 and 20,000 instructions, respectively.
- Ignore any other overhead
- Ques: What is the maximal possible speedup with/without using interrupt?

Solution

- Cycle time = 50 ns = 0.05 us = 5×10^{-8} ms
- Time for executing 1,000,000 inst = 1,000,000 x 4 x cycle_time = 0.2 ms.
- Without interrupt: T_wo = 0.2 + 300 + 100 = 400.2 ms
- With interrupt
  - T_w = 0.2 + (10,000 + 20,000) x 4 * cycle_time = 0.206 ms
- Speedup = 400.2 / 0.206

Examples

- A computer has a cache, main memory, and a disk used for virtual memory. If a referenced word is in the cache, 20ns are required to access it. If it is in main memory but not in the cache, 60ns are needed to load it into the cache, and then the reference is started again. If the word is not in main memory, 12ms are required to fetch the word from disk, followed by 60ns to copy it to the cache, and then the reference is started again. The cache hit ratio is 0.9 and the main memory hit ratio is 0.99. What is the average time in nanoseconds required to access a referenced word on this system?

- Average memory access time (AVAT) = Hit_time_{L1} + $Miss\_Rate_{L1}$ * $Miss\_Penalty_{L1}$
- $Miss\_Penalty_{L1}$ = Hit_Time_{L2} + $Miss\_Rate_{L2}$ * $Miss\_Penalty_{L2}$
- $Miss\_Penalty_{L2}$ = Hit_Time_{L3} + $Miss\_Rate_{L3}$ * $Miss\_Penalty_{L3}$

- Hit_time_{L1} = 20ns $Miss\_Rate_{L1}$ = 1- 0.9
- Hit_time_{L2} = 60ns $Miss\_Rate_{L2}$ = 1- 0.99
- Hit_time_{L3} = 12ms = 12 x 10^6 ns $Miss\_Rate_{L3}$ = 0

- So
- AVAT = 20 + 0.1 * (60 + 0.01 * (12 x 10^6 + 0)) ns