CS2710 Computer Organization

Dr. Walter Schilling

Winter, 2012-2013

For the final exam, you may bring one 8.5 x 11 inch sheet of paper with notes.

1. Week 1 Lecture 1: Course Introduction Chapter 1 pgs 2-13
   (a) Draw the relationship between hardware, systems software, and applications software as a hierarchy
   (b) Explain the concept of a compiler
   (c) Draw a flowchart showing the basic steps of compilation for code.
   (d) Explain the relationship between high level languages, assembly language, and machine language
2. Week 1 Lecture 2: Computer Parts and Performance pgs 13-38
   (a) List the 5 classical components of a computer and show their relationships graphically.
   (b) Define instruction set architecture.
   (c) Define throughput and response time.
   (d) Explain the relationship between user CPU time and System CPU time.
   (e) Define the relationship between CPU execution time, CPU clock cycles, and Clock rate.
   (f) Given the average CPI for instruction classes, determine which implementation will execute faster.
   (g) Given an instruction profile, calculate the average CPI for the program.
3. Week 1 Lecture 3: Ohms Law and Power Handout, pgs. 39-40
   (a) Explain the relationship between current, voltage, and resistance (Ohms Law)
   (b) Calculate the power dissipated in a resistor based on voltage and current
   (c) Explain the relationship between clock rate and dissipated power.
4. Week 2 Lecture 4: Benchmarking pgs 44-55
   (a) Calculate the geometric mean of a set of numbers.
   (b) Explain the SPEC benchmarks.
   (c) Define Amdahl's law
   (d) Define MIPS
5. Week 2 Lecture 5: Instructions Part 1 pgs 76-87
   (a) Explain the difference between Harvard and Von Neumann architectures in a computer.
   (b) Define instruction set
   (c) Explain the concept of the source and destination registers for the MIPS instruction set.
   (d) Using the MIPS instruction set, explain how to add a set of variables.
   (e) Define the term computer register
   (f) Using the mips instruction set, initialize a register to a fixed value.
6. Week 2 Lecture 6: Instructions Part 2 and the SPIM Simulator pgs 87-94
   (a) Define alignment restriction
   (b) Define the terms "Big endian" and "little endian"
   (c) Define the term address
   (d) Define the term data transfer instruction

7. Week 3 Lecture 7: Numeric Representation pgs 87-94
    (a) Define the terms least significant bit and most significant bit.
    (b) Explain how unsigned integer numbers are represented in memory
    (c) Understand the limitations of using sign and magnitude to represent signed integer numbers
    (d) Explain how signed integer numbers are represented in memory using twos complement notation.
    (e) Convert twos complement numbers into decimal.
    (f) Convert decimal numbers into twos compliment format.
    (g) Explain the concept of sign extension
8. Week 3 Lecture 8: Programming Practice: Logical Operations and Assembly Code pgs 94-112
    (a) Define instruction format
    (b) List the MIPS instruction fields and explain their purpose
    (c) Explain the MIPS implements logical operations
    (d) Explain how MIPS implements a bitwise not operation.
    (e) List the MIPS conditional branch statements.
    (f) Draw a flowchart showing conditional branches executing.
    (g) Construct while, do-while, and for loops in MIPS assembly language
9. Week 3 Lecture 9: Procedures pgs 112-127
    (a) Define procedure
    (b) Explain the MIPS calling convention for procedure calls.
    (c) Define the term program counter
    (d) Explain the relationship between the caller and callee in a procedure call
    (e) Explain the purpose for the stack when invoking procedures
    (f) Explain the relationship between characters and strings.
    (g) Explain the three main mechanisms for representing strings in memory.
10. Week 4 Lecture 10: Compiling and Executing a Program pgs 139-148
    (a) Not to be covered on Midterm.
11. Week 4 Lecture 11: Arrays and Pointers pgs 157-161
    (a) Explain how arrays are declared in C.
    (b) Explain the relationship between arrays and pointers.
    (c) Declare an array in MIPS assembly language.
    (d) Construct MIPS assembly language code which manipulates data in arrays.
12. Week 4 Lecture 12: Addition, Subtraction, and Multiplication Chapter 3 pgs 224-235
    (a) Explain the relationship between addition and subtraction with twos compliment numbering systems.
    (b) Explain the concept of numeric overflow when dealing with twos compliment numbers.
    (c) Explain the concept of an exception.
    (d) Define multiplier and multiplicand.
    (e) Explain the concept of left and right shifting.
    (f) Explain how a computer may use left and right shifting to perform arithmetic multiplication sequentially.
    (g) Explain the advantage of using left or right shifts when multiplying or dividing by a power of 2.
    (h) Explain how hardware can be added to improve multiplication times.
    (i) Compare and contrast the time complexity of multiplication with addition and subtraction.
    (j) Explain how a computer deals with signed multiplication.
    (k) Explain the size relationship between the multiplier, multiplicand, and the product.

(l) Construct simple MIPS assembly language code to perform multiplication oeprations.

13. Week 5 Lecture 13: Binary Division pgs 236-242

    (a) Perform binary division of two numbers.
    (b) Define dividend, divisor, quotient, and remainder.
    (c) Explain how division is accomplished in computer hardware.
    (d) Construct a program which uses MIPS integer multiplication and division

14. Week 5 Lecture 14: Floating Point Computations pgs 242-270

    (a) Define floating point number.
    (b) Define the terms fraction and exponent when dealing with floating point numbers.
    (c) Define overflow and underflow in relation to floating point numbers.
    (d) Convert a floating point number from binary to decimal format.
    (e) Convert a floating point number from decimal to floating point format.
    (f) Draw the algorithm for adding two floating point numbers.
    (g) Calculate the result of adding two floating point numbers together.

15. Week 5 Lecture 15: Floating Point Continued and Real World Ramifications pgs 275-279

    (a) Draw the algorithm for multiplying two floating point numbers together.
    (b) Calculate the result of the multiplication of two floating point numbers.
    (c) Explain the ramifications of the Pentium division bug.

16. Week 6 Lecture 16: Midterm Exam Review None

    (a)

17. Week 6 Lecture 17: Midterm Exam None

    (a)
    (b)

18. Week 6 Lecture 18: The Processor and its Datapath Chapter 4 pgs 300-316

    (a) Draw an abstract view of the core of a microprocessor. (Fig 4.3)
    (b) Compare and contrast combinational elements with state elements.
    (c) Define edge triggered clocking.
    (d) Define the term program counter.
    (e) Draw the data path segment responsible for fetching instructions and incrementing the program counter.
    (f) Define sign extend.
    (g) Define branch target address

19. Week 7 Lecture 19: Pipelining pgs 330-344

    (a) Define pipelining
    (b) Calculate the speedup achieved by pipelining for a given number of instructions.
    (c) Define how pipelining improves computer performance.
    (d) Define structural hazard, data hazard, and control hazard.
    (e) Define the term stall.
    (f) Explain the concept of forwarding.

20. Week 7 Lecture 20: Branch Prediction pgs 375-391

    (a) Define branch prediction.
    (b) Draw a state machine for a 2 bit branch prediction scheme
    (c) Explain the impact on the compiler of branch delay.

(d) Define the concept of a vectored interrupt

21. Week 7 Lecture 21: Cache Basics Chapter 5 pgs 452-475

    (a) Define temporal and spatial locality.
    (b) Define hit rate and miss rate.
    (c) Define the term cache
    (d) Explain how a direct mapped cache determines the location in the cache.
    (e) Given a memory address and cache size information, perform the calculations to convert an address to a cache block number.

22. Week 8 Lecture 22: Cache Handling pgs 475-492

    (a) Explain the relationship between miss rate and block size in a cache.
    (b) Construct a flowchart explaining how a cache miss is handled.
    (c) Compare and contrast write through and write back caching schemes.
    (d) Define the term write buffer.
    (e) Perform cache related calculations to analyze the impact of having a cache on a computer system.

23. Week 8 Lecture 23: Cache Design pgs 475-492, pgs 518-525

    (a) Define set associative cache and fully associative cache.
    (b) Compare and contrast the performance of set associative caches, direct mapped caches, and fully associative caches.
    (c) Explain the operation of the LRU replacement scheme.
    (d) Explain the concept of a multi-level cache.
    (e) Explain the three C model for cache.

24. Week 8 Lecture 24: IO Introduction and Reliability Chapter 6 pgs 570-575

    (a) Draw a picture showing the connection between the processor and memory mapped IO devices.
    (b) Define the terms reliability, dependability, and availability

25. Week 9 Lecture 25: Disk Based Storage pgs 575-579

    (a) Define the terms track, sector, and seek as pertaining to a nonvolatile storage medium.
    (b) Calculate the average rotational latency for a disk given disk performance criteria.
    (c) Calculate the disk read time for a hard drive given performance characteristics for the disk.
    (d) Based on desired criteria, choose appropriate performance characteristics for a given drive based on need.

26. Week 9 Lecture 26: Flash Storage and IO Operation pgs 580-586

    (a) Explain the limitations of flash memory.
    (b) Define wear leveling.
    (c) Define the term IO Transaction