Topics for the First Midterm

CS 315-01

Spring, 2015

The following topics may be covered on the first midterm.

• Chapter 1
  
  – Why knowledge of computer architecture is important for software developers.
  
  – Interpretations of performance: execution time, throughput, power consumption.
  
  – Hardware components: I/O, memory, datapath, control
  
  – Relation between high-level language, assembly language, and machine language.
  
  – Compiler, assembler.
  
  – Instruction set architecture
  
  – Application binary interface
  
  
  – Performance is the reciprocal of execution time, or execution time per program.
  
  – Wall clock vs CPU time.
  
  – Clock frequency vs. clock period.
  
  – The classic cpu performance equation:

\[
CPU \text{ time} = \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Frequency}}.
\]
- IPC and CPI
- Growth in processor performance since the 1980's. Its relation to power consumption. Why designers switched to multicore architectures.
- Consequences of parallelism for performance improvements
  - Moore’s law
  - Amdahl’s law
  - Relation between power consumption and CPU utilization.

- Chapter 2
  - MIPS registers and conventions for their usage.
  - MIPS memory layout: stack, heap, static data, program text.
  - Manipulation of stack pointer, return address.
  - MIPS core instructions
  - Basic use of Mars/Spim simulators
  - Instructions vs pseudoinstructions
  - Use of syscall
  - .text, .globl, .data, .asciiz
  - Branches and loops in MIPS
  - Function calls/returns in MIPS
  - Recursion
  - Words vs. bytes
  - Allocating memory on the stack
  - Program break, sbrk system call, allocating memory on the heap
  - Reading the green sheet
  - Representation of unsigned numbers: binary, octal, hexadecimal, decimal.
  - Converting between representations.
  - Representation of signed numbers: sign-magnitude, one’s complement, two’s complement
- Range of $n$-bit two’s complement integers.
- R-format instructions
- I-format instructions
- Storing large immediates in a register with \texttt{lui} and \texttt{ori}.
- Zero-extended vs Sign-extended immediates.
- J-format instructions
- Immediates in branch and jump instructions count words rather than bytes.
- Calculation of addresses in branch instructions (PC-relative addressing) and jump instructions.
- Dealing with branches and jumps to “distant” destinations.