Spring 2011 Midterm Exam Review

mr-1

When / Where

Wednesday, 30 March 2011, 9:40-10:30 CDT

226 Tureaud (Here)

Conditions

Closed Book, Closed Notes

Bring one sheet of notes (both sides), 216 mm × 280 mm.

No use of communication devices, even for accessing Facebook `:-)`.

Format

Several problems, short-answer questions.

Resources

Solved tests and homework:. `http://www.ece.lsu.edu/ee4720/prev.html`

Statically Scheduled MIPS Study Guide:. . .
. . . `http://www.ece.lsu.edu/ee4720/guides/ssched.pdf`

Study Recommendations

Study this semester's homework assignments. Similar problems may appear on the exam.

Solve Old Problems—memorizing solutions **is not the same** as solving.

Following and understanding solutions **is not the same as** solving.

Use the solutions for brief hints and to check your own solutions.

Previous Midterms

Emphasis

mr-3

MIPS Programming and Instruction Use

Should be able to write MIPS programs.

Should be able to easily understand MIPS programs.

Should be able to use other ISAs' instructions in examples.

Not required to memorize instruction names, except for common MIPS instructions.

MIPS Implementation

Implementation Diagrams and Pipeline Execution Diagrams

They are a *team*, so study them together.

ISA Families

MIPS v. SPARC

Topics

mr-4

ISA v. Implementation.

See Lecture Slides 1—`http://www.ece.lsu.edu/ee4720/2011/lsli01.pdf`

Why separating ISA from implementation is important.

Why it's hard to design an ISA without bias towards first implementation.

mr-5

MIPS ISA

See MIPS Overview—`http://www.ece.lsu.edu/ee4720/2011/lmips.html`

Write and read programs.

Understand how instructions are coded.

mr-6

Statically Scheduled MIPS Implementations

See Lecture Slides 6—`http://www.ece.lsu.edu/ee4720/2011/lsli06.pdf` *and Statically Scheduled Study Guide*—`http://www.ece.lsu.edu/ee4720/guides/ssched.pdf`

Unpipelined Implementation

Understand relationship between insn format and connections to register file, etc.

Pipelined Implementations

Basic (no bypassing).

ALU bypassing, branch in ID.

mr-7

Dependency Definitions

Hazard Definitions

For a Given Pipelined Implementation

- Show pipeline execution diagrams.
- Show register contents at any cycle.
- Determine control hardware.
- Determine CPI.

ISA Families

mr-8

ISA Families: RISC, CISC, VLIW

See ISA Families Overview—`http://www.ece.lsu.edu/ee4720/2011/lrisc.html`

Key distinguishing features and their rationale.

RISC: Goal: Easy pipelining, simple (relatively) implementation.

Fixed-length instructions, balanced work, aligned memory access.

CISC: Goal: Powerful instructions, compact code size.

Flexible operand types, multiple-activity instructions.

VLIW: Goal: Easy superscalar implementation.

Bundled instructions, dependence information.

mr-9

RISC to RISC Differences: MIPS v. SPARC

See ISA Families Overview—`http://www.ece.lsu.edu/ee4720/2011/lrisc.html`

Opcode fields (and their extensions).

Immediate operands, immediate sizes.

Integer branches.

mr-10

CISC ISA Features

Goal: Powerful Instructions

Powerful: Many memory addressing modes.

Powerful: Many immediate sizes.

Powerful: Few limits on where addressing modes can be used.

mr-11

Compilers and Optimization

See Optimization Lecture Notes—`http://www.ece.lsu.edu/ee4720/2011/lopt.html`

Steps in building and compiling.

Basic optimization techniques, compiler optimization switches.

Profiling.

How programmer typically uses compiler switches (options).

mr-12

Benchmarks

See middle of Lecture Slides 2—`http://www.ece.lsu.edu/ee4720/2011/lsli02.pdf`, *SPEC-cpu description*—`http://www.spec.org/benchmarks.html`, *and the SPECcpu run and reporting rules*—`http://www.spec.org/cpu2006/Docs/runrules.html`

Benchmark types: kernel, synthetic, real progs.

SPECcpu Benchmark Suite

What suite measures.

Benchmark programs (types, how they were selected).

Why it's useful for computer engineers.

Why results might be trusted:

SPEC membership and their interests.

Rules for running benchmarks and disclosing results.

Interrupts and Exceptions and Traps

mr-13

Difference between interrupt, exception, trap.

Causes of exceptions, role of handler.

Privileged Mode.

Pipeline activity leading to execution of handler.

SPARC trap mechanism. (Trap base register, etc.).

Precise exceptions, achieving with floating-point operations.

mr-14

Floating-Point and other Long-Latency Operations

Types of operations. (Floating point and maybe load.)

Degree of pipelining: Initiation interval, latency.

Detecting functional unit structural hazards.

Detecting WB structural hazards: reservation register.

Detecting and handling RAW hazards: ID-stage v. pre-WB stall.

Control logic.

Handling WAW hazards.

Why precise exceptions more difficult in pipelined impl.