Spring 2020 Midterm Exam Review

When / Where

Monday, 13 April 2020 to 23:59 Friday 17 April 2020.

Print-home. *("Take-home" suggests violating the stay-at-home order.)*

Topics

Everything up to and including benchmarks. (Interrupts will not be covered.) (Spring 2020).

Conditions

Work on the exam alone.

E-mail questions to koppel@ece.lsu.edu. Don't be shy. Gotachas welcome.

You can use slides, references, and other materials provided for the course.

Do not seek out solutions to questions.

Format

Several problems, short-answer questions.

Resources

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

Statically Scheduled MIPS Study Guide:. . .
. . . `https://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.

Emphasis

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

ISA v. Implementation.

See Lecture Slides 1—`https://www.ece.lsu.edu/ee4720/2020/lsli01.pdf`

Why separating ISA from implementation is important.

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

MIPS ISA

See MIPS Overview—`https://www.ece.lsu.edu/ee4720/2020/lmips.s.html`

Write and read programs.

Understand how instructions are coded.

Statically Scheduled MIPS Implementations

See Lecture Slides 6—`https://www.ece.lsu.edu/ee4720/2020/lsli06.pdf`
and Statically Scheduled Study Guide—`https://www.ece.lsu.edu/ee4720/guides/ssched.pdf`

Unpipelined Implementation

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

Pipelined Implementation Versions

Basic (no bypassing).

ALU bypassing, branch in ID.

Many variations covered in class and in assignments.

Terminology

Dependency Definitions

Hazard Definitions

MIPS Implementation Skills

For a Given Pipelined Implementation

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

Add bypass paths.

Add functional units.

ISA Families: RISC, CISC, VLIW

See ISA Families Overview—`https://www.ece.lsu.edu/ee4720/2020/lrisc.s.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.

RISC to RISC Differences: MIPS v. SPARC

See ISA Families Overview—`https://www.ece.lsu.edu/ee4720/2020/lrisc.s.html`

Opcode fields (and their extensions).

Immediate operands, immediate sizes.

Integer branches.

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.

Compilers and Optimization

See Optimization Lecture Notes—`https://www.ece.lsu.edu/ee4720/2020/lopt.s.html`

Steps in building and compiling.

Basic optimization techniques, compiler optimization switches.

Profiling.

Compiler ISA and implementation switches.

How programmer typically uses compiler switches (options).

Benchmarks

See Lecture Slides 2—`https://www.ece.lsu.edu/ee4720/2020/lsli02.pdf`,
SPECcpu description—`http://www.spec.org/benchmarks.html`,
and the SPECcpu run and reporting rules—`https://www.spec.org/cpu2017/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.