Introduction

- Note: This is just a collection of some of the topics covered in class. For complete coverage refer to class notes.
- Basic trends
- Instruction Set Architecture
- RISC Processors

Review of MIPS Operand Addressing Modes

- Register addressing – operand is in a register
- Base (displacement) addressing – operand is at the memory location whose address is the sum of a register and a 16-bit constant contained within the instruction
- Immediate addressing – operand is a 16-bit constant contained within the instruction

Review of MIPS Instruction Addressing Modes

- PC-relative addressing – instruction address is the sum of the PC and a 16-bit constant contained within the instruction
- Pseudo-direct addressing – instruction address is the 26-bit constant contained within the instruction concatenated with the upper 4 bits of the PC

Course Administration

- Midterm exam:
  - March 21th at 5:00pm in the same classroom
  - Open Book
  - No notes allowed
- Lab 1 – Part 1 Due: today at 5pm
  - Extended deadline tonight at midnight with no penalty
  - Sunday March 11 with 10-point penalty
  - Wednesday March 14 with 20-point penalty
  - No submission after that
- Lab 1 part 2 Due: March 28th at 5pm. No extensions.
MIPS (RISC) Design Principles

- Simplicity favors regularity
  - fixed size instructions – 32-bits
  - small number of instruction formats
  - opcode always the first 6 bits
- Good design demands good compromises
  - three instruction formats
- Smaller is faster
  - limited instruction set
  - limited number of registers in register file
  - limited number of addressing modes
- Make the common case fast
  - arithmetic operands from the register file (load-store machine)
  - allow instructions to contain immediate operands

Numbers

- Binary representation
- MIPS arithmetic
- Add/subtract/Multiply/Divide
- Floating point numbers
- Floating point operations

IEEE 754 FP Standard Encoding

- Most (all?) computers these days conform to the IEEE 754 floating point standard
  \((-1)^{\text{sign}} \times (1+F) \times 2^{\text{E-bias}}\)
- Formats for both single and double precision
- F is stored in normalized form where the msb in the fraction is 1 (so there is no need to store it!) – called the hidden bit
- To simplify sorting FP numbers using integer comparisons
  - E comes before F in the word and
  - E is represented in excess (biased) notation

<table>
<thead>
<tr>
<th>Single Precision</th>
<th>Double Precision</th>
<th>Object Represented</th>
</tr>
</thead>
<tbody>
<tr>
<td>E (8)</td>
<td>F (23)</td>
<td>E (11)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>nonzero</td>
<td>0</td>
</tr>
<tr>
<td>1-254</td>
<td>anything</td>
<td>1-2046</td>
</tr>
<tr>
<td>255</td>
<td>0</td>
<td>2047</td>
</tr>
<tr>
<td>255</td>
<td>nonzero</td>
<td>2047</td>
</tr>
</tbody>
</table>

Summary: Evaluating ISAs

- Design-time metrics:
  - Can it be implemented, in how long, at what cost?
  - Can it be programmed? Ease of compilation?
- Static Metrics:
  - How many bytes does the program occupy in memory?
- Dynamic Metrics:
  - How many instructions are executed? How many bytes does the processor fetch to execute the program?
  - How many clocks are required per instruction?
  - How "lean" a clock is practical?

Best Metric: Time to execute the program!

MIPS Architecture

- Basic MIPS Architecture
- Components of MIPS Datapath
- Pipelining
  - Pipeline Hazards
    - structural hazards: attempt to use the same resource by two different instructions at the same time
    - data hazards: attempt to use data before it is ready
      - An instruction’s source operand(s) are produced by a prior instruction still in the pipeline
    - control hazards: attempt to make a decision about program control flow before the condition has been evaluated and the new PC target address calculated
      - branch instructions

Pipelining Issues

- Forwarding
- Hazard Detection
- Stalling
  - Possible approaches for control hazards
    - Stall (impacts CPI)
    - Move decision point as early in the pipeline as possible, thereby reducing the number of stall cycles
    - Delay decision (requires compiler support)
    - Predict and hope for the best
- Two types of stalls
- Dealing with Exception
Multiple Issue Introduction

- Superpipelining
- Multiple-issue:
  - Super scalar
  - VLIW
- Instruction vs Machine Parallelism
- Multiple-issue types:
  - In-order issue with in-order completion
  - In-order issue with out-of-order completion
  - Out-of-order issue with out-of-order completion
  - Out-of-order issue with out-of-order completion and in-order commit
- True data dependency, Output dependency, Antidependency

Next Lecture and Reminders

- Next lecture
  - March 28th
- Reminders
  - Midterm exam: March 21th @ 5:00pm
  - Lab 1 due today