Computer Systems and Networks

ECPE 170 – Jeff Shafer – University of the Pacific

Modern Instruction Sets

Schedule

- **Today**
  - Finish Chapter 5 (instruction sets)
  - Quiz 4
- **Next week**
  - Chapter 6 – Memory systems
- **Beyond**
  - **Exam 2** – Tuesday, Nov 1st

Gradebook

- **Please check Sakai gradebook**
  - All grades correctly entered?

Homework 5.18

- **Addressing modes – what is loaded in the accumulator?**
  - Instruction: LOAD 500
  - Contents of memory:
  - Contents of R1: 200
- Value in ACC for:
  - **Immediate mode?**
  - **Direct mode?**
  - **Indirect mode?**
  - **Indexed mode?**

Address	Data
100	600
400	300
500	100
600	500
700	800

Homework 5.19

- **Speedup of a pipelined system? (both with infinity instructions and with finite instructions)**
- Scenario
  - Non-pipelined system: 200ns per instr (t_n)
  - Pipelined system: 40ns per stage (t_p)
  - Pipeline depth: 5
- **Theoretical speedup**
  - **For ∞ instructions?**
  - **For 200 instructions?**

$$S = \frac{nt_n}{(k+n-1)t_p} = \frac{200 \times 200ns}{(5+200-1)\times 40ns} = 4.90$$

Recap – Pipeline Pitfalls

- **Why might we not achieve this full speedup in a pipelined system?**
  - Unbalanced pipeline: We *assumed* that a 200ns-per-instruction unpipelined system could be converted to a 5-stage 40ns-per-stage system
    - If the pipeline stages aren't perfectly balanced, we have to clock at the slowest stage
    - We might only be able to achieve 50 or 60ns per stage…
  - **Hazards**
    - Data hazards / structural hazards / control hazards

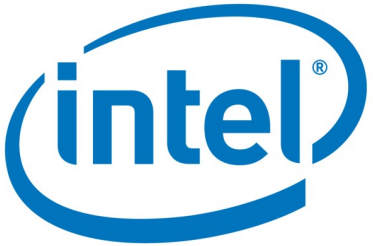
Real-World ISAs

Real-World Examples of ISAs

- What does **Intel** do?
- What does **MIPS** do?
- What does **Java** do?

Intel Design

- Processor is
  - Little endian
  - 2-address architecture (register-memory)
    - One operand can be a memory address, but the other must be a register number
  - Variable-length instructions
  - Variable-length data (1, 2, 4, 8 bytes)



Intel ISA + Pipelining

- Almost all Intel chips (286, 386, 486, etc…) have had some degree of pipelining
- Pipelining was first seriously applied to the **Intel 486** chip in 1989
  - Could complete an ALU instruction (coming from a register, going to a register) *every clock cycle*
- Pipelining got better with the **Pentium** chip in 1993
  - Double-wide: *Two instructions* are sent down the pipeline every cycle! (Requires two ALUs, etc…)

Intel ISA + Pipelining

- Pipeline depth changed over time:
  - Original Pentium: 5 stages
  - Pentium 2: 12 stages
  - Pentium 3: 14 stages
  - Pentium 4: 20-24 stages
  - Pentium 4 extreme edition: 31 stages
  - **Why were the pipelines getting longer?**
- Today
  - Core i7 has a 17-stage pipeline

Intel ISA + Addressing Modes

- Intel processors support a wide variety of addressing modes.
- The original 8086 (16-bit processor, released in 1978) provided **17 ways to address memory**
  - Displacement, register indirect, indexed, base indexed, etc…
  - http://www.ic.unicamp.br/~celio/mc404s2-03/addr_modes/intel_addr.html
- **How many addressing modes do you think the Pentium processor supported in 1993?**
  - All 17 (thank you, backwards compatibility)
- Other Intel chips (the Itanium product line) only support one addressing mode
  - Register indirect addressing with optional post increment
  - Reduces the amount of hardware required

MIPS Design

- MIPS was an acronym for *Microprocessor Without Interlocked Pipeline Stages*.
  - **Interlocked?** Hardware that detects a pipeline hazard and stalls the pipeline until hazard is resolved
  - Originally the MIPS omitted this hardware to simplify design
    - But software (the assembler) had to insert NOPs so programs wouldn't break!
    - NOPs made the program code bigger, so the hardware was eventually added back in…

MIPS Design

- MIPS processors used in many products, including PlayStation 2 / PSP
- The architecture:
  - Little endian
  - Word-addressable
  - Three-address, fixed-length instructions

MIPS + Pipelining

- Like Intel, the pipeline size of the MIPS processors has grown
  - R2000 and R3000 have 5-stage pipelines
  - R4000 and R4400 have 8-stage pipelines
  - R10000 has three pipelines:
    - 5-stage pipeline for integer instructions
    - 7-stage pipeline for floating-point instructions
    - 6-state pipeline for LOAD/STORE instructions

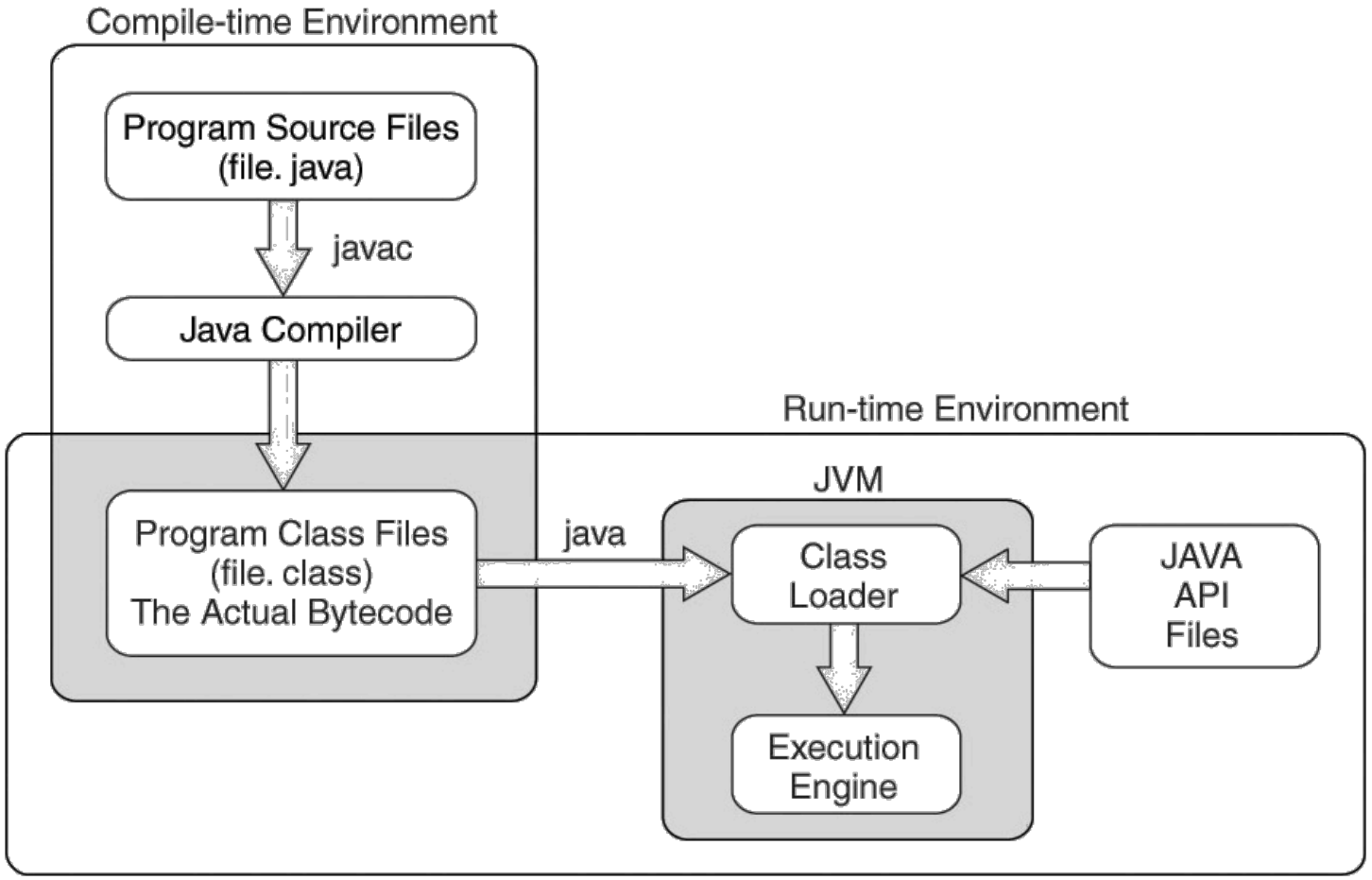
MIPS ISA

- In all MIPS ISAs, only the LOAD and STORE instructions can access memory
- The ISA uses only base addressing mode
- The assembler accommodates programmers who need to use immediate, register, direct, indirect register, base, or indexed addressing modes
  - *Better software allows for simpler hardware while still giving programmers flexibility…*

Java Design

- Java is a programming language
  - **Why are we discussing it in a chapter focused on processor design?**
- Java is an interpreted language that runs in a software machine called the Java Virtual Machine (JVM)
  - The JVM is written in a native language for a wide array of processors, including MIPS and Intel
  - Like a real machine, the JVM has an ISA all of its own, called **bytecode**. This ISA was designed to be compatible with the architecture of any machine on which the JVM is running

Java Design


Compile-time Environment
Program Source Files
(file. java)
javac
Java Compiler
Run-time Environment
JVM
Program Class Files
(file. class)
The Actual Bytecode
java
Class
Loader
JAVA
API
Files
Execution
Engine

Java ISA

- Java bytecode is a stack-based language
  - Most instructions are zero address instructions
- The JVM has four registers that provide access to five regions of main memory
- All references to memory are offsets from these registers. Java uses no pointers or absolute memory references
- Java was designed for platform interoperability, not performance!
  - Bare minimum design that allows a JVM to be built on top of virtually any other type of processor

Java Design

- **Does the JVM have to be implemented in software?**
  - No! You could build a hardware chip that runs Java bytecode directly

Chapter 5 Conclusion

- ISAs are distinguished according to their bits per instruction, number of operands per instruction, operand location and types and sizes of operands
- Endianness is another major architectural consideration
- CPU can store store data based on
  1. A stack architecture
  2. An accumulator architecture
  3. A general purpose register architecture