5.6 Real-World Examples of ISAs

- We return briefly to the Intel and MIPS architectures from the last chapter, using some of the ideas introduced in this chapter.

5.6.1 Intel

- Intel introduced pipelining to their processor line with its Pentium chip.
- The first Pentium had two five-stage pipelines. Each subsequent Pentium processor had a longer pipeline than its predecessor with the Pentium IV having a 24-stage pipeline.
- The Itanium (IA-64) has only a 10-stage pipeline.

- Intel processors support a wide array of addressing modes.
- The original 8086 provided 17 ways to address memory, most of them variants on the methods presented in this chapter.
- Owing to their need for backward compatibility, the Pentium chips also support these 17 addressing modes.
- The Itanium, having a RISC core, supports only one: register indirect addressing with optional post increment.

5.6.2 MIPS

- MIPS was an acronym for *Microprocessor Without Interlocked Pipeline Stages*.
- The architecture is little endian and word-addressable with three-address, fixed-length instructions.
- Like Intel, the pipeline size of the MIPS processors has grown: The R2000 and R3000 have five-stage pipelines.; the R4000 and R4400 have 8-stage pipelines.

- The R10000 has three pipelines: A five-stage pipeline for integer instructions, a seven-stage pipeline for floating-point instructions, and a six-state pipeline for LOAD/STORE instructions.
- 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.

5.6.3 Java Virtual Machine

- The Java programming language is an interpreted language that runs in a software machine called the *Java Virtual Machine* (JVM).
- A 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.

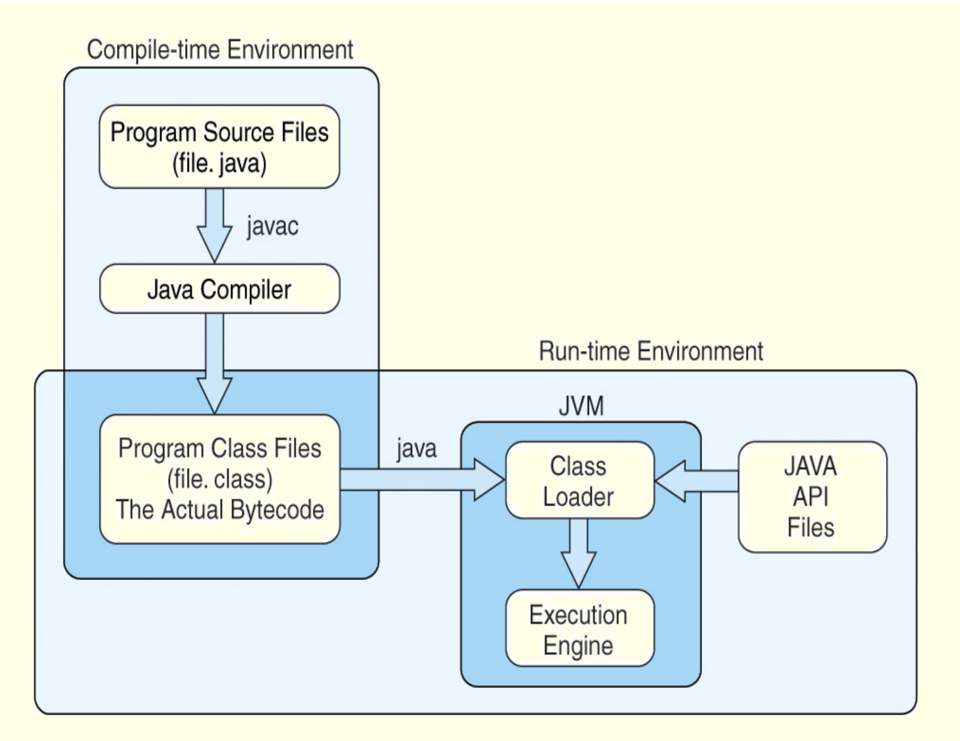

Figure 5.7 The Java Programming Environment

- 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!

P-code Languages:

- HLL translated (compiled) to P-Code
- Executed on a P-Machine that has the same function as the Java Virtual Machine.
- Examples: Pascal (for which P-Code was designed), Python and Perl.

5.6.3 ARM (Advanced RISC Machine)

- You may not have heard of ARM but most likely use an ARM processor every day. It is the most widely used 32-bit instruction architecture:
  - 95%+ of smartphones,
  - 80%+ of digital cameras
  - 40%+ of all digital television sets
- Founded in 1990, by Apple and others, ARM (Advanced RISC Machine) is now a British firm, ARM Holdings.
- ARM Holdings does not manufacture these processors; it sells licenses to manufacture.

- ARM is a load/store architecture: all data processing must be performed on values in registers, not in memory.
- It uses fixed-length, three-operand instructions and simple addressing modes
- ARM processors have a minimum of a three-stage pipeline (consisting of fetch, decode, and execute);
  - Newer ARM processors have deeper pipelines (more stages). Some ARM8 implementations have 13-stage integer pipelines

- ARM has 37 total registers but their visibility depends on the processor mode.
- ARM allows multiple register transfers.
  - It can simultaneously load or store any subset of the16 general-purpose registers from/to sequential memory addresses.
- Control flow instructions include unconditional and conditional branching and procedure calls
- Most ARM instructions execute in a single cycle, provided there are no pipeline hazards or memory accesses.

5. Summary

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