Computer Architecture
(CSC-3501)
Lecture 15
(13 Mar 2008)

Seung-Jong Park (Jay)
http://www.csc.lsu.edu/~sjpark

Announcement

4.14 Real World Architectures

- MARIE shares many features with modern architectures but it is not an accurate depiction of them.
- In the following slides, we briefly examine two machine architectures.
- We will look at an Intel architecture, which is a CISC machine and MIPS, which is a RISC machine.
  - CISC is an acronym for complex instruction set computer.
  - RISC stands for reduced instruction set computer.

We delve into the "RISC versus CISC" argument in Chapter 9.

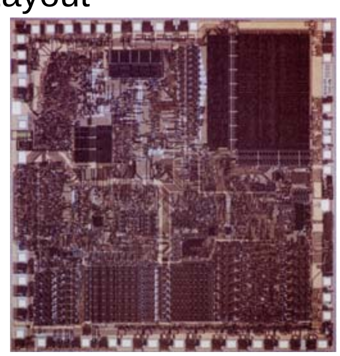
4.14 Real World Architectures

- The classic Intel architecture, the 8086, was born in 1979. It is a CISC architecture.
- It was adopted by IBM for its famed PC, which was released in 1981.
- The 8086 operated on 16-bit data words and supported 20-bit memory addresses.
- Later, to lower costs, the 8-bit 8088 was introduced. Like the 8086, it used 20-bit memory addresses.

What was the largest memory that the 8086 could address?

8086 CPU Layout

- 29,000 registers
- 5MHz
- 16 bit registers
- 20 address pin
- 1MB RAM



4.14 Real World Architectures

- The 8086 had four 16-bit general-purpose registers that could be accessed by the half-word.
- It also had a flags register, an instruction register, and a stack accessed through the values in two other registers, the base pointer and the stack pointer.
- The 8086 had no built in floating-point processing.
- In 1980, Intel released the 8087 numeric coprocessor, but few users elected to install them because of their cost.

4.14 Real World Architectures

- In 1985, Intel introduced the 32-bit 80386.
- It also had no built-in floating-point unit.
- The 80486, introduced in 1989, was an 80386 that had built-in floating-point processing and cache memory.
- The 80386 and 80486 offered downward compatibility with the 8086 and 8088.
- Software written for the smaller word systems was directed to use the lower 16 bits of the 32-bit registers.

4.14 Real World Architectures

- Currently, Intel's most advanced 32-bit microprocessor is the Pentium 4.
- It can run as fast as 3.8 GHz. This clock rate is nearly 800 times faster than the 4.77 MHz of the 8086.
- Speed enhancing features include multilevel cache and instruction pipelining.
- Intel, along with many others, is marrying many of the ideas of RISC architectures with microprocessors that are largely CISC.

4.14 Real World Architectures

- The MIPS family of CPUs has been one of the most successful in its class.
- In 1986 the first MIPS CPU was announced.
- It had a 32-bit word size and could address 4GB of memory.
- Over the years, MIPS processors have been used in general purpose computers as well as in games.
- The MIPS architecture now offers 32- and 64-bit versions.

4.14 Real World Architectures

- MIPS was one of the first RISC microprocessors.
- The original MIPS architecture had only 55 different instructions, as compared with the 8086 which had over 100.
- MIPS was designed with performance in mind: It is a *load/store* architecture, meaning that only the load and store instructions can access memory.
- The large number of registers in the MIPS architecture keeps bus traffic to a minimum.

How does this design affect performance?

Assembly Language

- To command a computer, you must understand its language.
  - Instructions: words in a computer's language
  - Instruction set: the vocabulary of a computer's language
- Instructions indicate the operation to perform and the operands to use.
  - Assembly language: human-readable format of instructions
  - Machine language: computer-readable format (1's and 0's)
- We introduce the MIPS architecture.
  - Developed by John Hennessy and his colleagues at Stanford and in the 1980's.
  - Used in many commercial systems, including Silicon Graphics, Nintendo, and Cisco
- Once you've learned one architecture, it's easy to learn others.

John Hennessy

- President of Stanford University
- Professor of Electrical Engineering and Computer Science at Stanford since 1977
- Coinvented the Reduced Instruction Set Computer (RISC)
- Developed the MIPS architecture at Stanford in 1984 and cofounded MIPS Computer Systems
- As of 2004, over 300 million MIPS microprocessors have been sold

David Patterson

- Writes textbooks with John Hennessy
- First in his family to graduate from college (UCLA, 1969)
- Professor of Computer Science at UC Ber keley since 1977, where he coinvented RI SC, the Reduced Instruction Set Compute r
- In 1984, he developed the SPARC archite cture used by Sun Microsystems
- Invented RAID (Redundant Array of Inexp ensive Disks) and NOW (Network of Work stations)

Architecture Design Principles

The underlying design principles that we will follow, as arti culated by Hennessy and Patterson:

1. Simplicity favors regularity
2. Make the common case fast
3. Smaller is faster
4. Good design demands good compromis es

Instructions: Addition

- The most common operation computers perform is additio n.

High-level code	MIPS assembly code
`a = b + c;`	`add a, b, c`

- add: the mnemonic indicates what operation to perform
- b, c: source operands on which the operation is performed
- a: destination operand to which the result is written

Instructions: Subtraction

- Subtraction is similar to addition. Only the mnemonic ch anges.

High-level code	MIPS assembly code
`a = b - c;`	`sub a, b, c`

- sub: the mnemonic indicates what operation to perform
- b, c: source operands on which the operation is performe d
- a: destination operand to which the result is written

Design Principle 1

Simplicity favors regularity

- Consistent instruction format.
- Same number of operands – easier to encode and han dle in hardware.

Instructions: More Complex Code

- More complex code is handled by multiple MIPS instruct ions.

High-level code
```
a = b + c - d;
// single line comment
/* multiple line
   comment */
```

MIPS assembly code
```
add t, b, c  # t = b + c
sub a, t, d  # a = t - d
```

Design Principle 2

Make the common case fast

- MIPS includes only simple, commonly used instructions.
- So, the hardware to decode and execute the instruction can be sim ple, small, and fast.
- More complex instructions (that are less common) can be performe d using multiple simple instructions.
- MIPS is a *reduced instruction set computer* (RISC), with a small nu mber of simple instructions.
- Other architectures, such as Intel's IA-32 found in many PC's, are *complex instruction set computers* (CISC).
- Include complex instructions that are rarely used, such as the "strin g move" instruction that copies a string (a series of characters) fro m one part of memory to another.

Operands

- A computer needs a physical location from which to retri eve binary operands.
- A computer retrieves operands from:
  - Registers
  - Memory
  - Constants (also called *immediates*)

Operands: Registers

- Memory is slow.
- Most architectures have a small set of (fast) registers.
- MIPS has thirty-two 32-bit registers.
- MIPS is called a 32-bit architecture because it operates o n 32-bit data.

(A 64-bit version of MIPS also exists, but we will consid er only the 32-bit version.)

Design Principle 3

Smaller is Faster

- MIPS includes only a small number of registers.
- Just as retrieving data from a few books on your table is faster than retrieving data from 1000 books, retrieving d ata from 32 registers is faster than retrieving it from 100 0 registers or a large memory.

Chapter 4 Conclusion

- The major components of a computer system ar e its control unit, registers, memory, ALU, and d ata path.
- A built-in clock keeps everything synchronized.
- Control units can be microprogrammed or hardw ired.
- Hardwired control units give better performance, while microprogrammed units are more adaptabl e to changes.

Chapter 4 Conclusion

- Computers run programs through iterative fetch-decode-execute cycles.
- Computers can run programs that are in machin e language.
- An assembler converts mnemonic code to mach ine language.
- The Intel architecture is an example of a CISC a rchitecture; MIPS is an example of a RISC archit ecture.