CSC2510 - Computer Organization

Lecture 1: Introduction

Philip Leong

Course Details

- Lecturer: Prof. Philip Leong
- Tutor: Miss Tu Zhou (Wed 1:30pm LHCG06, Thur 4:30pm LHC101)
- Lectures:
  - M9-10 changed from HCA 332 to ERB 401 (starting Mon 11th).
  - T5 ERB712
- Credit: 50% Final, 25% Mid-Term, 25% Homework

Thanks to Drs Y.S. Moon (CUHK), O. Mencer (Imperial), N. Dulay (Imperial) for some of the slides used in this course

What is a Computer?

- Apple Macintosh “Personal Computer” (circa 1990)

What is a Computer?

- Thinking Machines “Connection Machine”
What are they used for?

- Word processing
- Image processing
- Art: Music/Pictures/Movies creation, distribution
- Entertainment / Games
- AI - Robotics
- Simulation
- Communications e.g. Chat, Email, Video conferencing, etc.

What are they used for?

- Computer-Aided Design
- Air-Traffic Control
- Weather Prediction
- Weapons
- Designer Drugs
- Oil Exploration
- Human Genome Project
- Financial Markets
- Nuclear Reactor Control
- Exploring Space
- Art: Music, Pictures, Movies, creation and distribution

Computers & CPUs

COMPUTER
Apple
Power Macintosh
7100/66

CPU
Motorola
PowerPC 601/66

Course Aims

In this course we shall focus on the Organisation & Operation of the CPU. This involves computer architecture and assembly language programming

CPUs

- Intel Pentium
- Motorola/IBM PowerPC
- AMD K7
- ARM StrongArm
- Compaq (DIGITAL) Alpha
- Zilog Z80
- Motorola 68000
- 6502
- MIPS

- Interesting details about CPUs
  http://bwrc.eecs.berkeley.edu/CIC/archive/cpu_history.html
Why?

- Operating Systems
- Compilers
- Architecture
- Hardware
- Communications
- VLSI

Computer Organization

- Binary Numbers
- Data Representation
- Main Memory Organisation
- CPU Organisation & Operation
- CPU & Programming
- Input/Output Control

What will be covered in CSC2510?

- Lectures
  - Basic Knowledge
  - Machine Instructions & Assembly Language
  - Intel IA32 instruction set
  - I/O
  - Memory
- Tutorials
  - Reinforce lecture material on computer architecture
  - Assembly language programming

Overview of Organization

Input/Output Unit

- Input unit
  - Keyboard, mouse, microphone, CDROM etc
- Output unit
  - Graphical display, printer etc
- Use the collective term input/output unit (I/O unit)
  - Input units, output units, disk drive etc

Memory unit

- Memory used to store programs and data
- Unit of access is an n-bit word
  - Unique location is its address
  - Retrieval is in units of words
  - Commonly 32-bit today, moving to 64-bits
  - Typically 16-64 bit machines
- Primary storage: random-access memory (RAM)
- Secondary storage: hard disk, CDROM etc
**Processor**

- **Registers**
  - Small but fast storage of intermediate values in a computation
- **Arithmetic logic unit**: performs computations
  - e.g. add, divide, logical operations etc
  - c.f. calculator
  - Operands taken from registers
- **Control**
  - Orchestrates the transfer of data and sequencing of operations between memory, registers, ALU, I/O devices

**How a program is executed**

- Program and data reside on CDROM or hard disk
- Data stored in memory is fetched under program control into ALU and processed
- Processed information sent back to I/O unit
- All activities directed by control unit

**Processor detail**

- Machine has n registers
- PC used to keep track of execution of a program
- Points to the next instruction to be fetched from memory
- Memory address register (MAR) used to specify address for memory transfer
- Memory data register (MDR) contains data to be written or data read from memory
- Some instructions specify I/O operations

**Bus**

- A bus is used for communications between the processor, memory and I/O
  - advantages?
    - multiple buses are often used for higher performance

**System Software**

- Receive and interpret user commands
- Storage and retrieval from secondary storage
- Running standard application programs e.g. games, word processors, spreadsheets, editors, compilers, assemblers, linkers
- Interface to I/O units e.g. printer
- A key piece of system software is the operating system
Operating system (OS)

- Assigns computer resources to application programs
  - e.g. sharing of the I/O units, processor, memory, running applications programs

Multitasking

- Assume computer with 1 processor, 1 disk and 1 printer and application program is in machine code on disk
- t0: the OS loads the program from the disk to memory
- t1: program executes
- t2: program accesses disk
- t3-t4: program executes some more
- t4: program accesses printer
- t5: program terminates

Applications programs

- Normally written in high level language such as C, C++, Java, Fortran
  - Compiler translates to assembly language and then machine code which the computer can directly execute

Performance

- Most important measure is how quickly it executes your program (t5-t0 of earlier example)
  - Compiler, instruction set and hardware architecture, program all have impact on performance

Caching

- Optimising memory for speed (cache)
  - Commonly used data are copied to on-processor memory (cache) to reduce access time
  - Small memories can be made higher speed than large ones. In a computer, we need both.

Performance

- Performance (T=execution time, N=number of instructions, S=average cycles per instruction, R=clock rate in cycles per second)
  \[ T = \frac{N \times S}{R} \]
  - Processor circuits are controlled by a clock (R)
    - e.g. 3.2GHz Pentium operates at 3.2 GHz
      - Control circuit (but not memory) operates at this frequency
Pipelining and Superscalar

- Pipelining
  - Like a production line, instruction execution overlapped so greater parallelism is achieved
  - What parameters are improved?

- Superscalar operation
  - Execute several instructions simultaneously
  - Multiple ALUs

CISC vs RISC

- Reduced instruction set RISC
  - Large N, small S
- Complex instruction set computer
  - Small N, large S

Compiler

- Translates high level language such as C, C++, Java to machine instructions
- Aims to reduce N x S

Performance Measurement

- Computer architects use performance estimates to evaluate the effectiveness of new features
  - Practice is to use benchmarks available to all manufacturers e.g. SPEC benchmark
- Benchmark figures are for real computers
- There are n individual programs and the geometric mean compared with a 300MHz UltraSPARC-IIi machine is the benchmark value
  - Value indicates how much faster e.g. SPEC rating of 10 means it is 10 x faster than the UltraSPARC-IIi