COMP3211/COMP9211
Computer Architecture 2003

Oliver Diessel

Welcome

- Introduction
- Administrative matters

Introduction

- What is computer architecture?
- Why study computer architecture?
- How will we study computer architecture & organisation?

What is computer architecture?

- *Architecture* is those attributes visible to the programmer
  - Instruction set, number of bits used for data representation, I/O mechanisms, addressing techniques.
  - e.g. Is there a multiply instruction?
- *Organization* is how features are implemented
  - Control signals, interfaces, memory technology.
  - e.g. Is there a hardware multiply unit or is it done by repeated addition?

Architecture & Organization

- All Intel x86 family share the same basic architecture
- The IBM System/370 family share the same basic architecture
- This gives code compatibility
  - At least backwards
- Organization differs between different versions

Hierarchies

- Computers are complex systems
- The key to their description lies in recognizing their hierarchical nature
  - Hierarchical structures consist of interrelated subsystems, each of which is also hierarchical in nature
- The hierarchical nature of complex systems is essential to their design and description
  - Can deal with one level at a time
  - Can abstract the behaviour of lower level components

Structure & Function

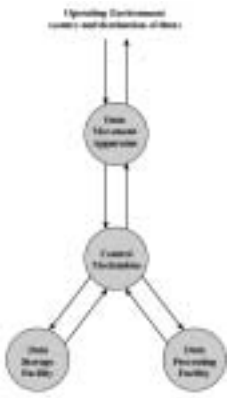
- The components of the computer system at any given level in the hierarchy are described and designed in terms of its structure and function
- *Structure* is the way in which components relate to each other
- *Function* is the operation of individual components as part of the structure

Function

- In essence, computers, and their components, simply perform the following functions:
  - Data processing
  - Data storage
  - Data movement
  - Control

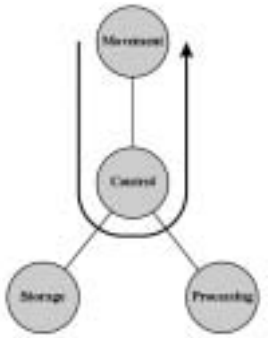
Functional view

- At this general level of description there are only a few possible operations



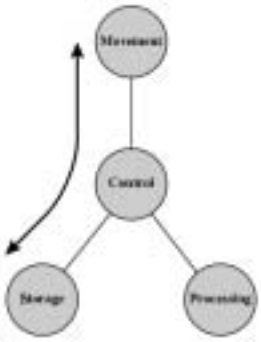
Operations (1) Data movement

- Example: Network routing



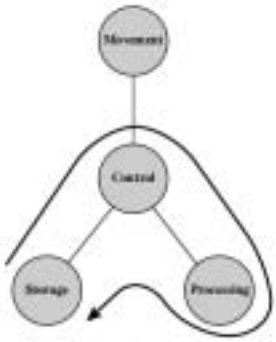
Operations (2) Storage

- E.g.: File download



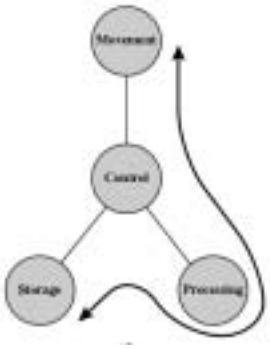
Operation (3) Processing from/to storage

- Transformations on data, e.g., file compression

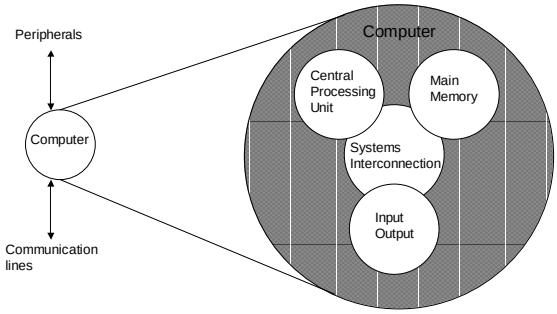


Operation (4) Processing from storage to I/O

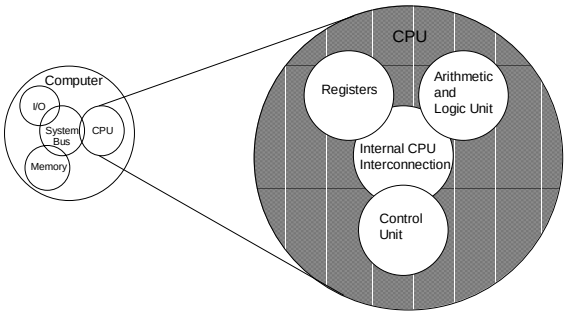
- E.g.: Decompressing and presenting an MPEG



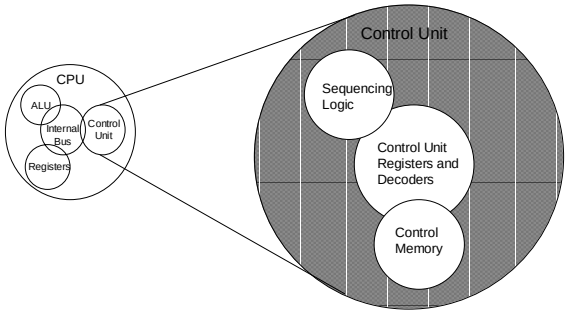
Structure - Top Level



Structure - The CPU



Structure - The Control Unit



Why study computer architecture?

- To better understand the structure and function of computers and how these impact on performance and program design
- To understand the tradeoffs among various components such as CPU clock speed & memory size
- To better appreciate the impact of technology trends and market forces upon computer design
- To be able to make more informed computer design, infrastructure development, and purchasing decisions

How will we study computer architecture and organisation?

- Top-down, with emphasis on the central components (CPU, memory, and interconnect)
- Learn concepts that apply to many systems rather than presenting a catalogue of techniques that can be found in a selection of machines
- Projects and exercises will be used to reinforce the concepts presented in lectures
- Develop familiarity with VHDL as a specification and simulation language

Administrative matters

- Course goals
- Lectures, Tutorials, and Labs
- Assignments, Quiz, and Final Exam
- Assessment and Supplementaries
- Staff, Text, References, and Support

Course goals

- Study the architecture & organisation of modern processors with emphasis on pipelined RISC machines
- Gain understanding of the design of the memory subsystem, I/O, and system level interconnect
- Become proficient in the use of VHDL for the description, simulation, and verification of architectural designs
- Complete a series of assignments leading to the design, implementation, and validation of a RISC system

Lectures

- Wed 2-3 EE LG1 & Fri 10-12 Biomed C
- Approximate schedule:
  - 1 week: Introduction and Background on context and performance
  - 2 weeks: Background on VHDL
  - 2 weeks: Interconnect & Buses
  - 2 weeks: Cache & Memory
  - 1 week: Instruction Set Architectures
  - 2 weeks: Single & Multi-cycle Datapath & Control
  - 2 weeks: Pipelining
  - 2 weeks: Current examples taken from PowerPC, Pentium, Thumb

Tutorials

- 1 hour tutorials are held each week commencing in Week 2
- Exercises will be released on the web in the week before they are held and solutions should be available during the week after the tutorial

Laboratory access

- No formally scheduled labs
- We will use Xilinx ISE 4.2 for designing and Modelsim 5.5 for simulating computer components and systems
- This software is available on vmware and should be distributed in pipe, bugle, piano, clavier, and organ labs
- You need to book your access when you need it
- Can be downloaded from the CSE server
- *Stay tuned for further exciting news!*

Assignments

- 4 assignments to be done in groups of 2 people from the same tutorial class
- Tutorial exercises will be designed to complement the assignments
- Planned schedule:
  - Ass 1 on Bus Transactions released Week 3, due Week 7;
  - Ass 2 on Cache & Memory released Week 6, due Week 9;
  - Ass 3 on Single-Cycle Datapath & Control released Week 8, due Week 12
  - Ass 4 on Pipelined System released Week 10, due Week 14

Quiz

- A 45 minute quiz will be held during the Wednesday lecture of Week 8, i.e., at 2.00pm on 17 September
- The quiz will cover all lecture, tutorial, and assignment material completed to the end of Week 7

Assessment

- Assignments
  - 40% weighting - distribution to be decided, but probably even
  - A 10% reduction in the mark obtained will apply for late submissions
  - Submissions will not be accepted after 5.00pm of the Friday following the week in which they are due
- Quiz
  - 15% weighting
- Final exam
  - 3 hours
  - 45% weighting
- Final mark = Assignments + Quiz + Exam
  - At least 40% must be achieved in the combined Quiz and Exam mark to pass this course, otherwise an FL will be given

Supplementaries

- If you miss the Final exam, you will only be offered a supplementary if you would have passed the course assuming you had scored at least 40% in the final
- If you are offered a supplementary exam as an additional assessment (having sat the Final already) and the result is satisfactory, then your final mark is set to 50

Staff

- Tutors
  - Jeremy Chan (jeremyc) Rm 501-10 Ext 4869
  - Newton Cheung (ncheung) Rm 501-03 Ext 4869
  - Andhi Janapsatya (andhij) Rm 501-02 Ext 7204
  - Lih Wen Koh (lwkoh)
  - Ivan Lu (ivanl) Rm 501-01 Ext 7204
  - Usama Malik (umalik) Rm 501-12 Ext 7205
- Administrator
  - Martin de Groot (martindg) Rm 301-06 Ext 7779
- Lecturer
  - Oliver Diessel (odiessel) Rm 502 Ext 5922

Text

- Computer Organization and Design: The Hardware/Software Interface, D.A. Patterson and J.L. Hennessy, 2nd Ed., Morgan Kaufmann, 1998.

Architecture References

- Computer Organization & Architecture: Designing for Performance, W. Stallings, 6th Ed., Prentice-Hall, 2003.
- Computer Organization, C. Hamacher, Z. Vranesic, and S. Zaky, 5th Ed., McGraw Hill, 2002.
- Structured Computer Organization, A.S. Tanenbaum, 4th Ed., Prentice-Hall, 1999.

VHDL References

- VHDL Starter's Guide, S. Yalamanchili, Prentice Hall, 1998.
- A VHDL Primer, J. Bhasker, 3rd Ed., Prentice Hall, 1999.
- The Designer's Guide to VHDL, P.J. Ashenden, 2nd Ed., Morgan Kaufmann, 2001.

Queries & Support

- Course material
  - **FIRST** contact your tutor - they build up a detailed knowledge about you and your needs during tutorials
  - If you still do not have a satisfactory answer, **THEN** email the lecturer, and if we cannot sort out the problem by mail, we can arrange a meeting
- Administrative issues
  - Please contact **Martin** for ALL administrative issues including tutorial rooms & allocations, assignment submissions, recorded marks, enrollment status, etc.
- Only email from your CSE account will be read

Next lecture

- Background
  - Part 1: Context for studies in computer architecture
  - Part 2: Performance measurement

Homework

- Acquire and skim through text
- Read Chapter 2