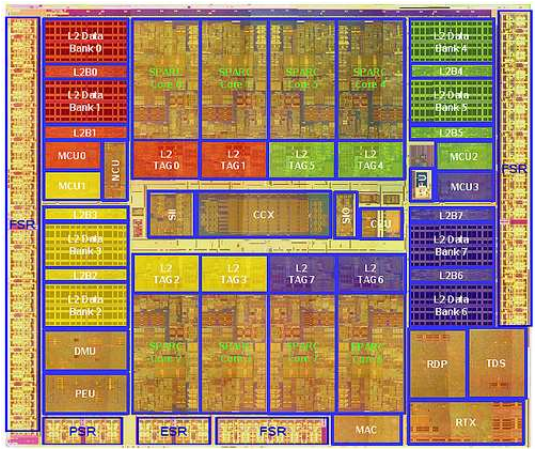
Welcome to COMP2300 – Introduction to Computer Systems

UltraSPARC T2
(Niagara-2)
multicore chip layout

a rather advanced computer system!
(courtesy of T. Okazaki, Flickr)



Course Contact

- course web site: http://cs.anu.edu.au/Student/comp2300
- course coordinator & lecturer:
  Peter Strazdins CSIT N219, 6125-5140, `comp2300@cs`
- course tutors: Jie Cai, Peter Strazdins and Li Zhou
- forums accessible by StReAMS:
  - Annoucements: postings from lecturers only
  - Discussion: for your use

Course Schedule

- lectures: three one hour lectures per week, five modules:
  - **D**igital building blocks (4)
  - **C** language (4)
  - **P**eANUt or "Assembly Level Machine Organisation" (9)
  - **M**emory Systems and Modern Machines (5)
  - **O**perating System Concepts (4)
  - Interconnection **N**etworks (1)
- other lectures: 1 introductory, mid-semester exam, and 1 exam preparation.
- tutorial/laboratories: 9 supervised, and 1 unsupervised
  - → register NOW via http://cs.anu.edu.au/streams
  - important to come prepared! will contain examinable material!
- assignments: 3
- more details on the course schedule page

Lecture Slides

- this year's lectures slides will be made available on the COMP2300 web site on the **day before** the corresponding lecture
  - can use last year's if need them earlier
- we advise you to print (the 4-up) lectures slides and bring them to the lectures
- printing lectures slides after lectures and learning from them may not be very useful (deliberately terse and incompl. . .)
- rather, annotate your notes onto lectures slides during the lectures
- reading from texts etc is necessary for developing your understanding

Course Assessment

- see the assessment web page
- designed to promote engagement; test *understanding* rather than *rote learning*
- Tute/Lab Marks: 10%
- Assignments: due weeks 6, 9 and 12 30%
  - this year's theme: the 'out-shuffle'
  - note plagiarism issues and unacceptable vs acceptable collaboration: producing a solution with the aid of another's solution is cheating!
- redeemable Mid-Semester Exam: tentatively Thu wk 7; covers **D** & **C** modules 20%
- Final Exam: 40%
  - 2+1 hours, 1 page A4 notes (both sides, printed or hand written), no calculator
  - former exam papers available from course web page; note that not all questions are still applicable
- to obtain a D or HD, you will be expected to have read around the course

  e.g. from the text books & further links on web site

References and Text Books

- Specification of the PeANUt Computer 2009 (or 2004–8) free this year!
  - why is a hardcopy useful?
- a reference book on C programming strongly recommended
  - C Programming: A Modern Approach*, K.N. King, 2008
  - Pure C Programming, Amir Afzal, 1999
  - The C Programming Language, Brian Kernighan and Dennis Ritchie, 1988
  - C Programming Made Simple by Conor Sexton, 1997
  - probably any other ANSI (i.e. relatively modern!) C programming text
- a book on Computer Architecture strongly recommended
  - The Essentials of Computer Organization and Architecture*, Null and Lobur, 2006
  - Computer Systems: A Programmers Perspective*, Bryant and O'Hallaron, 2003
  - Computer Systems, J Stanley Warford, 2003 (3rd ed)
  - Structured Computer Organization, A.S. Tanenbaum, 2005 (5th ed)
- further details (publishers, ISBNs) from course text web page

Discussion: Your Views: What are 'Computer Systems'

we will record your thoughts in these notes:

- what comes into your mind when you hear the term 'computer systems'?
- what aspects of computing/IT would you classify as 'computer systems'?
- what kind of tasks would you expect work in 'computer systems' to involve?
- what kind of skills would you expect to be required?

What's the Course all about?

- the hardware-software interface
- how the computer works
- system-oriented programming
- low-level programming
- programmer's view of computer hardware
- field of computer systems is based on 2 fundamental concepts:
  - abstraction: multiple levels of detail/description;
    - manage complexity, interfaces, standards
  - virtualization: give the *appearance* of a capability or service; decouple services from underlying physical resources
    - simplicity, flexibility, better resource sharing
- the computer systems mind-set:
  - real systems are very COMPLEX!
  - discern which parts are essential (for task-at-hand) first; (only) understand in detail those

Why Study Computer Systems?

- from IEEE/ACM Computing Curricula 2001, on the course web page:
- better understanding of how computers work
- machine-oriented programming
- operating system oriented programming
- middleware programming (e.g. JVM)
- better programmer
  - faster programs
  - safer programs
  - complex software systems
- disruptive technologies are now emerging, potentially affecting all of IT

What do we cover?

1. Digital Building Blocks
   number systems, data representation, logic gates, machine code, architectures, history, ...
2. C Programming
   functions, compiling, system oriented (Unix, Linux), ...
3. PeANUT computer
   architecture, registers, machine and assembly language, procedures, exceptions...
4. Memory Systems and Modern Architectures
   virtual memory, page replacement, latency, cache, instruction set design, ...
5. Operating System Concepts
   processes, scheduling, devices, file systems ...
6. Interconnection Networks
   communication model, switched/packet, Ethernet

What now. . . things to do

- register for a laboratory group with http://cs.anu.edu.au/streams
- select texts (C, computer systems)
- have a look at the course's web site
- inspect the course Discussion forum
- have a look at number systems (link on web site)