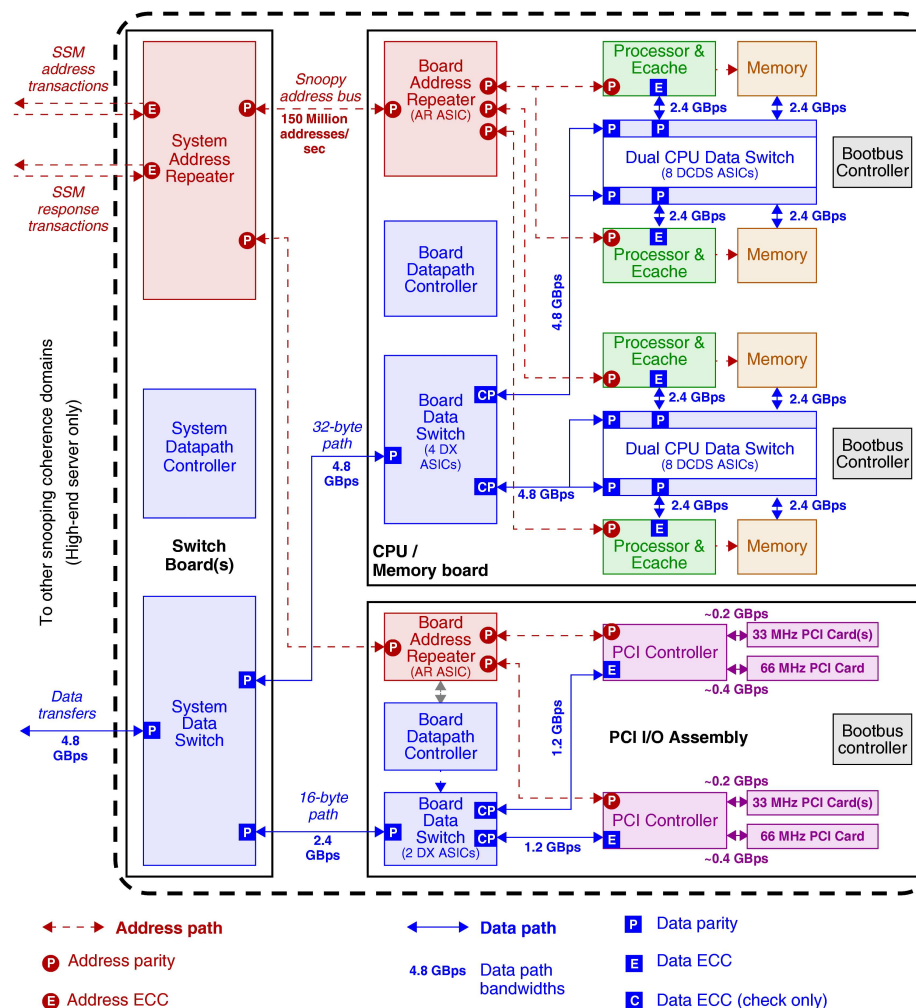


Welcome to COMP2300 – Introduction to Computer Systems

UltraSPARC III Cu processor layout



(a rather advanced computer system!)

Course Contact

- Course web site: <http://cs.anu.edu.au/Student/comp2300>
- Course coordinator & lecturer (M/2, O, N and Assignment 2):
Peter Strazdins CSIT N219, 6125-55140, comp2300@cs
- Lecturer (D, C, P, M/2 and Assignment 1)
Peter Christen CSIT N326, 6125 5960, [peter.christen \[at\] anu.edu.au](mailto:peter.christen@anu.edu.au)
- Course Tutors: Peter Janes and Andreas Pfeil
- Phorum: <http://cs.anu.edu.au/phorum>
 - [comp2300.announcements](#): posting from lecturers only
 - [comp2300.talk](#): for your use

Course Schedule

- Lectures: Three one hour lectures per week, five modules:
 - Digital building blocks (4)
 - C language (4)
 - PeANUt or “Assembly Level Machine Organisation” (9)
 - Memory Systems and Modern Machines (5)
 - Operating System Concepts (4)
 - Interconnection Networks (2)
- Other Lectures: 1 introductory and 1 exam preparation.
- Slides on the web site before lecture
- Laboratories: 6 supervised, 1 unsupervised and 2 homework
 - → Register NOW via <http://cs.anu.edu.au/streams>
 - will contain new and examinable material; see web page for times
- Assignments: 2
- more details on the course schedule page

Lecture slides

- Lectures slides will be made available on the COMP2300 web site **at least one day before** the corresponding lecture.
- We expect students to print (the 4-up) lectures slides and bring them to the lectures.
- Printing lectures slides after lectures and learning from them is not useful (the lectures slides are deliberately terse and incomplete)
- Rather, add notes onto lectures slides during the lectures.

Course Assessment

- see the assessment web page
- Assignments:
 - plagiarism issues and unacceptable vs acceptable collaboration: see sections 6.4 and 6.5 of the DCS Student Handbook
 - students have **two weeks** from the electronic release of the assignment marks. After then the marks will be final.
 - release of marks announced on the web site and at lectures
- Final Exam
 - 3 hours, write on exam, 1 page A4 notes (both side, printed or hand written), no calculator
 - to obtain an HD you will be expected to have read around the course – e.g. from the text books and material from further links on web site
 - former exam papers available from course web page; note not all questions are applicable

References and Text Books

- Specification of the PeANUt Computer 2005 (or 2004) Must buy!
 - **How to get it:** pay at MCC between 9.30am and 4.30pm before end of Friday 3rd March. Take receipt to DCS office.

- a reference book on C programming Strongly recommended
 - Pure C Programming*, Amir Afzal, 1999,
 - The C Programming Language, Brian Kernighan and Dennis Ritchie, 1988
 - C Programming Made Simple by Conor Sexton, 1997
 - C Programming for Engineering and CS, H.H. Tan and T.B.] D'Orazio, 1999
 - probably any other ANSI (i.e.] relatively modern!) C programming text

- A book on Computer Architecture Strongly recommended
 - Computer Systems: A Programmers Perspective*, Bryant and O'Hallaron, 2003
 - Structured Computer Organization, A.S. Tanenbaum, 1998 (4th Ed)
 - Computer Organization, C. Hamacher, Z. Vranesic and S. Zaky, 2002 (5th Ed)
 - Computer Architecture - A Quantitative Approach, J.L. Hennessey and D.A. Patterson, 2003 (3rd Ed),

- further details (publishers, ISBNs) from course text web page

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

Why study computer systems?

-
- From the course web page:
- Better understanding how computers work
- Machine oriented programming
- Operating system oriented programming
- Middleware programming (e.g. JVM)
- Better programmer
 - Faster programs
 - Safer programs

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, assembly language, register, procedures, interrupts...

4. Memory Systems and Modern Architectures

Virtual memory, page replacement, latency, cache, ...

5. Operating System Concepts

Processes, schedule, devices, file systems ...

6. Interconnection Networks

Communication model, switched/packet, Ethernet, GRID ...

What now.. things to do

- Register for a laboratory group with <http://cs.anu.edu.au/streams>
- Get the PeANUt reading brick
- Consider buying a computer architecture book
- Locate a book about C
- Have a look at the course's web site
- Inspect phorum
- Have a look at number systems (link on web site)