COMP3211/COMP9211
Computer Architecture 2004

Oliver Diessel

Welcome

- What is this course about?
- Administration

What is “Computer Architecture”?

Computer Architecture =
  Instruction Set Architecture +
  Machine Organization

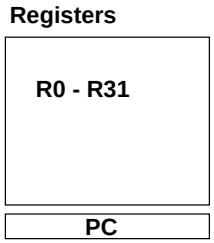
Example Instruction Set Architectures (ISAs)

• Digital Alpha	(v1, v3)	1992
• HP PA-RISC	(v1.1, v2.0)	1986
• Sun Sparc	(v8, v9)	1987
• SGI MIPS	(MIPS I, II, III, IV, V)	1986
• Intel x86	(8086,80286,80386, 80486,Pentium, MMX, ...)	1978

- PowerPC, ARM, IA64, …

MIPS R3000 ISA (Summary)

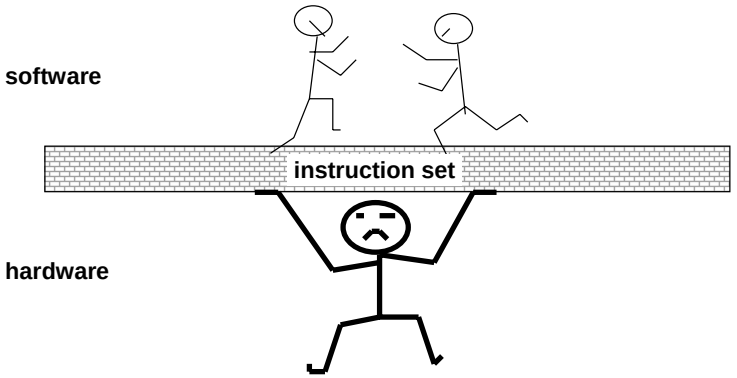
- Main instruction categories
  - Load/Store
  - Computational
  - Jump and Branch
  - Floating Point
    - coprocessor
  - Memory Management



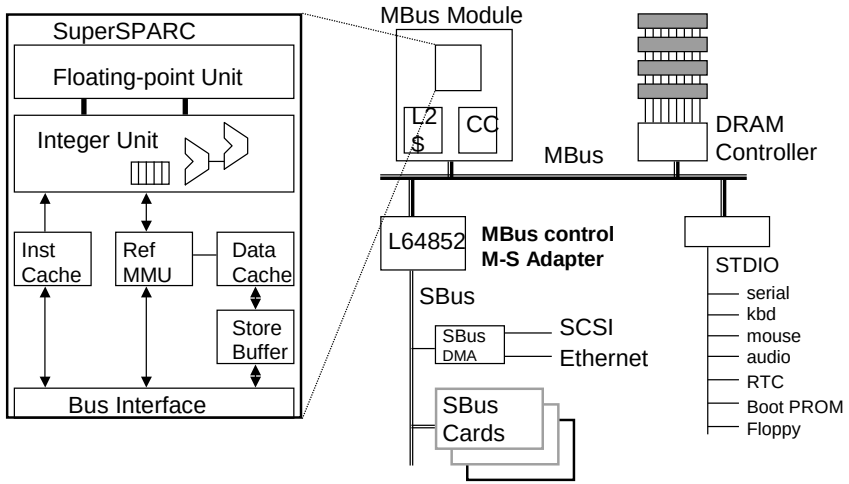
3 Instruction Formats: all 32 bits wide

OP	rs	rt	rd	sa	funct
OP	rs	rt	immediate		
OP	jump target				

The Instruction Set: a Critical Interface



Example Organization

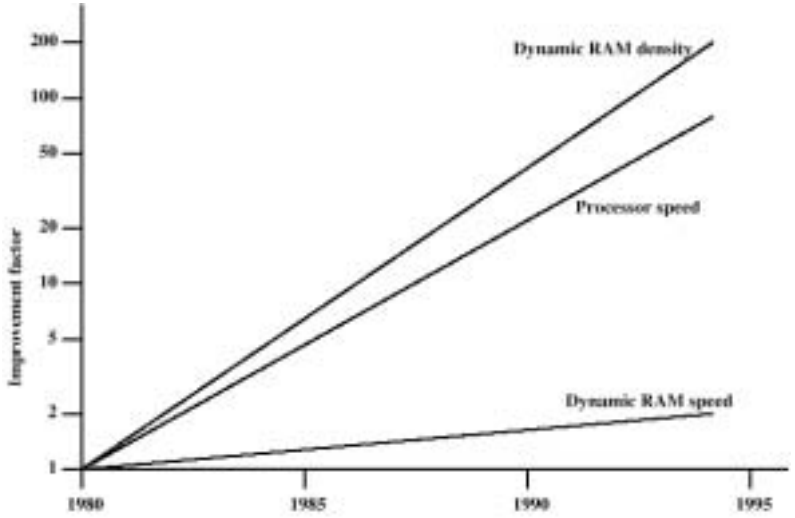


TI SuperSPARC^{tm} TMS390Z50 in Sun SPARCstation20

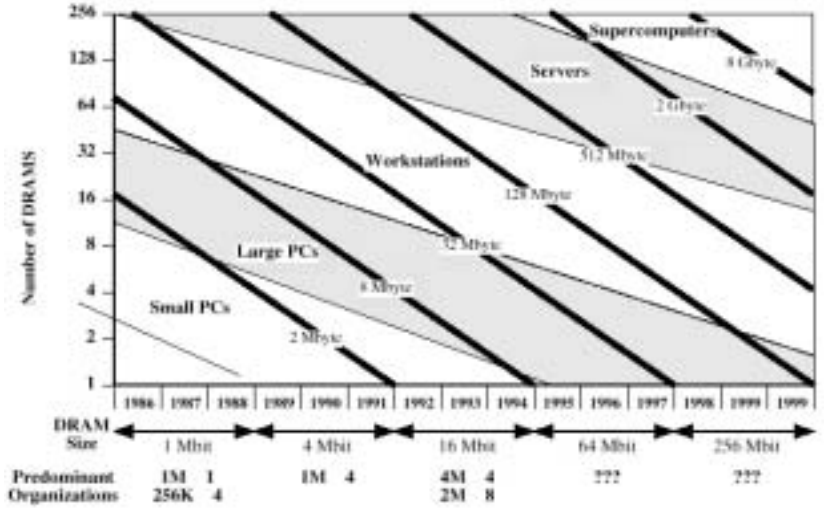
Technology ⇒ dramatic change

- Processor
  - Logic capacity: about 30% per year
  - Clock rate: about 20% per year
- Memory
  - DRAM capacity: about 60% per year (4x every 3 years)
  - Memory speed: about 10% per year
  - Cost per bit: about 25% per year
- Disk
  - Capacity: about 60% per year

DRAM and Processor Characteristics



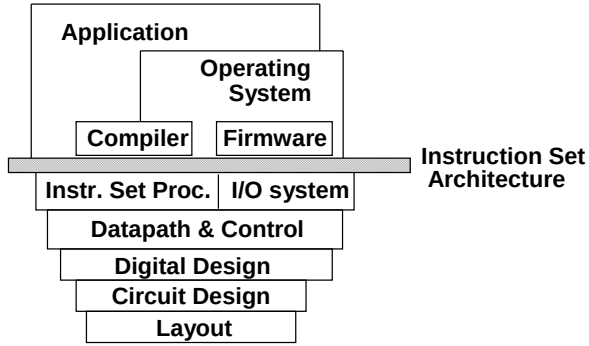
Trends in DRAM use



Solution approaches

- Increase number of bits retrieved at one time
  - Make DRAM “wider” rather than “deeper”
- Reduce frequency of memory access
- Change DRAM interface
  - Cache
  - More complex cache and cache on chip
- Increase interconnection bandwidth
  - High speed buses
  - Hierarchy of buses

What is “Computer Architecture”?



- Coordination of many levels of abstraction
- Under a constantly & rapidly changing set of forces
- Involves Design, Measurement, and Evaluation

Course Content

Computer Architecture and Engineering

Instruction Set Design	Computer Organization
Interfaces	Hardware Components
Compiler/System View	Logic Designer's View
-"Building Architect"	-"Construction Engineer"

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?

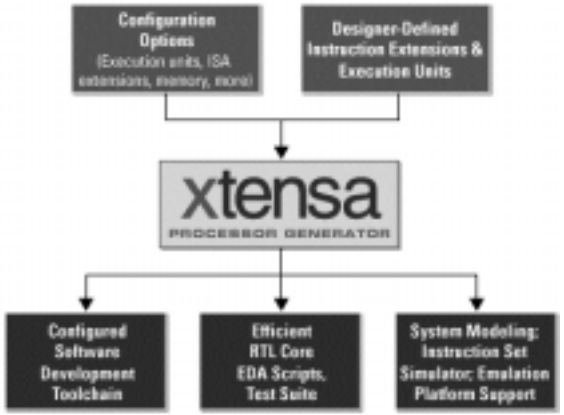
Architecture & Organization

- All Intel x86 family share the same basic architecture
- This gives code compatibility
  - At least backwards
- However, organization varies significantly between versions

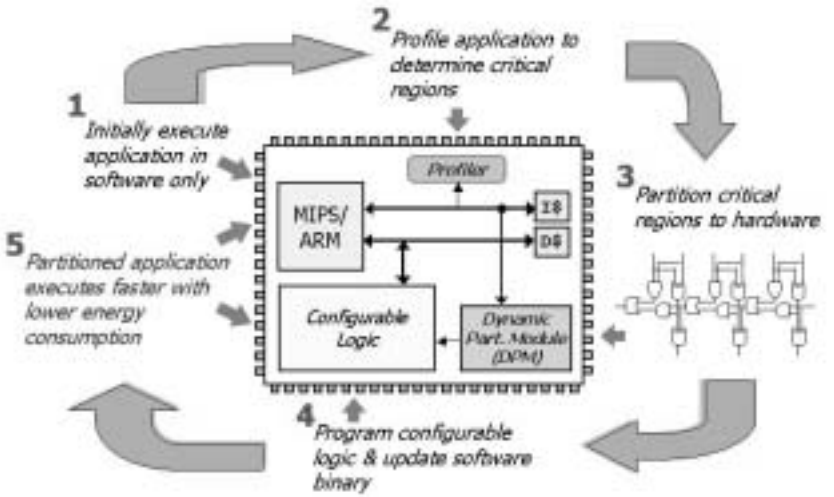
Why study computer architecture?

- 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

Xtensa: User configured architecture



WARP: Dynamic architectures



Administrative matters

- Course goals
- Lectures, Tutorials, and Labs
- Assignments, Quizes, 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
- Practice the use of VHDL for the description, simulation, and verification of architectural designs
- Complete a series of exercises 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
  - 1.5 weeks: Instruction Set Architectures
  - 3 weeks: Single & Multi-cycle Datapath & Control
  - 2 weeks: Pipelining
  - 1.5 weeks: Current examples taken from PowerPC, Pentium, Thumb
  - 1.5 weeks: Cache & Memory
  - 2 weeks: Interconnect & Buses

Tutorials & Labs

- Tutorials and labs will be held in alternating weeks
- 10 groups of approximately 20 students – 5 groups will do a tutorial and the rest will do a lab in *even-numbered weeks*, then swap activities in the following *odd-numbered week*
- Start Week 2; Finish Week 13
- 6 tutes & 6 labs all up
- COMPULSORY!
- Need to re-register for tutes & labs using Sirius
- Pick one “even” tutorial time and one “odd” lab time, or one “odd” tutorial time and one “even” lab time
- Eg WED12EVEN tute and MON10ODD lab means you do a tute in even-numbered weeks and a lab in odd-numbered weeks

Tutorials

- 1 hour long – held for 2 weeks – attend one
- Exercises will be released on the web by the Monday of the week they are first held and solutions should be available during the week after the tutorial is last held
- Involve hand-in exercises worth 5%

Labs

- 2 hours long – supervised for 1.5 hours
- Run for 2 weeks – attend one
- 2 tutors per group: intensive assistance, so use resource wisely
- We will initially use Xilinx ISE 4.2 for designing and Modelsim 5.5 for simulating computer components and systems
- We will then move on to using SimpleScalar for simulating and analyzing system performance
- Involve hand-in exercises worth 5%

Assignments

- 2 assignments to be done either individually or in groups of 2 people from the same lab class; groups of 3 may be allowed with Lecturer's permission – but must be good reasons
- Lab exercises will be designed to complement the assignments
- Planned schedule:
  - VHDL Ass 1 on Pipelined Datapath released Week 3, due Week 9;
  - SimpleScalar Ass 2 on System Architecture released Week 7, due Week 14
- Each assignment will contribute 15% to the final assessment for the course
- Assignments will offer extensions for bonus marks

Quizes

- There will be TWO 45 minute quizes held during Wednesday lecture period:
  - Quiz 1 in Week 2 and Quiz 2 in Week 8
- Quiz 1 will cover assumed background on combinational and sequential logic design, single cycle computer operation, and VHDL covered in DSS – worth 5%
- Quiz 2 will cover all lecture, tutorial, and lab material completed to the end of Week 7 – worth 10%

Assessment

- Tute & lab exercises
  - 5% each - 10% total - must be handed in on due date
- Assignments
  - 15% each - 30% total
  - 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
- Quizes
  - 5% and 10% respectively - 15% total
- Final exam
  - 3 hours
  - 45% weighting
- Final mark = Assignments + Quiz + Exam
  - You must score at least 40% of the available combined Quiz and Exam marks 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 and you submit documentation to NSQ
- 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
  - Marco Della Torre (marcodt)
  - Lih Wen Koh (lwkoh) Rm 510-06 Ext 57205
  - Ivan Lu (ivanl) Rm 510-01 Ext 57204
  - Boris Savkovic (boriss)
  - Charley Wong (cwong)
- Administrator
  - Martin de Groot (martindg) Rm 301-06 Ext 57779
- Lecturer
  - Oliver Diessel (odiessel) Rm 502 Ext 55922

Text

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

Queries & Support

- Course material
  - FIRST contact your tutor or lab demonstrator
  - THEN, if they cannot help, email me, and if I cannot help by mail, we can arrange a meeting
- Administrative issues
  - Please contact Martin for ALL administrative issues including tutorial & lab allocations, assignment submissions, recorded marks, enrolment 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