15-740 Basic Computer Systems
Fall 1998
Syllabus

1. Electronic Connections

Much of the material for the course will be available electronically in the AFS directory:

`/afs/cs.cmu.edu/academic/class/15740-f98/public`

In the following subdirectories you will find:

doc Copies of many of the readings.

asst Subdirectories with supporting material for each assignment.

info Copies of handouts.

bin Executable programs

lib Program libraries

We will use the World Wide Web (WWW) to provide up-to-date information regarding all aspects of the course. Refer to Web page:

`http://www.cs.cmu.edu`
`/afs/cs.cmu.edu/academic/class/15740-f98/www/home.html`

You can obtain copies of all handouts (including assignments), track the course schedule, and learn all sorts of other interesting things from this Web page.

You should also follow the newsgroup `cmu.cs.class.cs740`. You are free to post on this group.

2. Logistics

Class meetings

Tuesdays and Thursdays, **1:20** - 3:20 p.m. in Wean Hall 5409A. (Note the change in the starting time.)

Instructors

Randal E. Bryant
7128 Wean Hall
268-8821
Randy.Bryant@cs.cmu.edu

Todd C. Mowry
8123 Wean Hall
268-3725
tcm@cs.cmu.edu

Course Secretary

Rosemary Battenfelder
8124 Wean Hall
268-3853
rosemary@cs.cmu.edu

Textbooks

Main text:

- Hennessy, J. L, and Patterson, D. A., *Computer Architecture: A Quantitative Approach, 2nd Edition*. Morgan Kaufmann, 1996.

  Note: don't try to get by with the first edition of this book. The second edition is much different and more aligned to the content of this course.

Supplemental texts (copies should be available in the E&S Library by the time we need them):

- Keshav, S., *An Engineering Approach to Computer Networking*, Addison-Wesley, 1997.
- Culler, D., and Singh, J.P., with Gupta, A. *Parallel Computer Architecture: A Hardware/Software Approach*, Morgan Kaufmann, 1998.

Non-CS Students

If you are not a graduate student in the Computer Science PhD program, you need permission to take this class. If you have not already done so, send a message to the instructor stating your status, why you want to take the class, and if you want to take the class for credit or as an auditor.

Introduction

The goal of this course is to provide a comprehensive overview, suitable for computer science graduate students, of the design and implementation of computer systems. We use the term "computer system" in a broad sense, comprising the combination of hardware, system software (e.g., operating systems, compilers), and methodology by which applications are mapped onto computers. The first part of the course covers systems based on a single processor, closely following the Hennessy and Patterson textbook. The middle part briefly covers networking, with TCP/IP as

the focus, using the Comer textbook. The third part deals with parallel computing systems, spanning from programming models to hardware realizations. Material for this part of the course will include both Hennessy and Patterson, as well as additional papers and book excerpts.

The lectures will cover these topics (approximately in this order):

1. Instruction sets and machine-level programming.
2. Processor implementation, including pipelining techniques.
3. Memory systems, including caches and virtual memory.
4. Input, output, and storage, particularly with respect to supporting multimedia applications.
5. Networking, including local and wide area networks
6. Parallel architectures and parallel programming.

Themes

The primary goal of this course is to help students become more skilled in their use of computer systems. Traditional computer architecture courses have a "builder-centric" focus, where the intended audience is students who plan to become computer architects. In contrast, with our "user-centric" focus, we assume our students will mainly be system users, including developers of system and applications software. Users can benefit greatly from understanding how computer systems work, as well as their strengths and weaknesses. This is particularly true in developing applications that require maximum perfomance from a number of system components, such as multimedia applications.

A second theme is to emphasize the role of evolving technology in setting the directions for future computer systems. Computer systems, more than any other field of computer science, has had to cope with the challenges of exploiting the rapid advances in hardware technology. Hardware that is either technologically infeasible or prohibitively expensive in one decade, such as bitmapped full color displays or gigabyte disk drives, becomes consumer products in the next. Technology that seems to have a bright future, such as magnetic bubble memories, never becomes competitive. Others, such as CMOS, move from being a niche technology to becoming dominant. In addition, computer systems must evolve to support changes in software technology, including advances in languages and compilers, operating systems, as well as changing application requirements. Rather than teaching a set of facts about current (but soon obsolete) technology, we therefore stress general principles that can track evolving technology.

A third theme of the course is that "hands-on" exercises generally provide more insight and useful learning than to do ones involving just paper and pencil. All of the class assignments involve programming and using computer systems, although in a diversity of ways.

Placing Out

Students who have had a graduate-level computer architecture course or have equivalent work experience will find that much of the course material is familiar, particularly in the earlier parts of

the course. Most likely, however, you will find that the focus and style of the course is significantly different from what you have had before, and that the pace is sufficiently fast, that you will not be bored. If you feel strongly that you should be able to "place out" either partially or completely, you should contact the instructor.

Policies

Grades will be based on homeworks, a midterm exam, and a final exam.

For the homework assignments, we encourage limited collaboration among members of the class. Namely, we allow you to work in groups of up to three people in solving the homework problems.

To pass this course, you are expected to demonstrate competence in the major topics listed above. The final exam counts for 25%, the midterm counts for 15%, and the homework for 60%.

Late assignments will not be accepted without prior arrangement.

3. Prerequisites

The following would constitute full preparation for the course. Many students successfully complete the course with some deficiencies in their backgrounds. If you feel uncertain about whether you have adequate preparation, discuss this with one of the instructors. For each of the areas, a reference is suggested for self study.

- A working knowledge of C, including pointers and memory allocation [Kernighan+88].
- An introductory compiler course, especially aspects of code generation such as data formats, procedure linkages, and translation of control constructs [Aho+86, Chs. 1–2, 7–9].
- An introductory operating system course, especially aspects of protection, scheduling, and concurrency [Silberschatz+91, Chs. 4.1–4.3, 5.1–5.4, 5.7, and 7–9]
- Bit-level representation of and manipulation of numbers [HenPat96, App. A.1–A.2].

4. Syllabus

Uniprocessor design

Instruction Set Design Classes of instruction set architectures, addressing memory and operands, compiler perspective [HenPat96, Ch. 2]. We will actually use the Alpha architecture as our basis. This instruction set is very similar to DLX. Comparison between RISC [HenPat96, App. C] and CISC [HenPat96, App. D] architectures.

Pipelined processor implementation The DLX pipeline, including techniques for handling hazards, multicycle operations, and exceptions [HenPat96, Ch. 3].

Advanced pipelining Instruction level parallelism, branch prediction, and multiple issue. [HenPat96, Ch. 4.1–4.7, and 4.9–4.11]

Case Study The case study will focus on the DEC Alpha 21264. This is the highest performance processor ever designed. It employs a number of architectural and circuit features to maximize performance.

Memory Hierarchies

Caches Processor-memory interface, indexing, associativity, write policies [HenPat96, Chs. 5.1–5.6].

Virtual Memory Memory mapping, protection, page tables, and TLBs [HenPat96, Chs. 5.7–5.12].

Protocol verification Temporal logic model checking [Clarke+92]

Input, Output, and Storage

Storage devices Characteristics of disks and other devices [HenPat96, Chs. 6.1–6.2].

Buses I/O subsystems, [HenPat96, Chs. 6.3, 6.6–6.7, and 6.9–6.11]

Reliability and performance I/O performance [HenPat96, Chs. 6.4, 6.8], system reliability and use of redundancy [HenPat96, Chs. 6.5].

Local and Wide Area Networks

Network technology Ethernet, token ring, packet switching, ATM [HenPat96, Ch. 7].

Network protocols TCP/IP as an example of a layered protocol [Keshav+97, Chs. 3,15–16]

Parallel Architectures and Programming

Overview and Fundamentals Communication architecture, shared address space, message passing, data parallel processing, etc. [HenPat96, Ch. 8.1], [CulSin98, Ch. 1].

Parallel Programming The parallelization process, case studies, programming for performance [HenPat96, Ch. 8.2], [CulSin98, Chs. 2–3].

Cache Coherence Snoop-based coherence, directory-based coherence [HenPat96, Ch. 8.3–8.4 and App.E], [CulSin98, Chs. 5.1,6, and 8].

Memory Consistency Models Sequential and relaxed consistency [HenPat96, Ch. 8.6], [CulSin98, Chs. 5.2 and 9.1].

Latency Tolerance Prefetching and multithreading [HenPat96, Ch. 8.7], [CulSin98, Ch. 11].

Multiprocessor Interconnects Communication performance, topologies, routing, case studies [HenPat96, Ch. 7.5], [CulSin98, Ch. 10].

Synchronization Locks and barriers [HenPat96, Ch. 8.5], [CulSin98, Ch. 7.9].

Case Study SGI Origin 2000 [CulSin98, Ch. 8.5].

5. Schedule

Table 1 shows the tentative schedule. There might be some variations.

Table 1: CS 740, Fall 1998, Tentative Schedule.

Class	Date	Day	Topic	Reading	Homework	Who
1	9/15	Tu	Performance & Technology	H&P Ch. 1	#1 Out	TCM
2	9/17	Th	Alpha/DLX Programming	H&P Ch. 2		TCM
3	9/22	Tu	Instruction Set Comparison	H&P App. C & D		TCM
4	9/24	Th	Pipelined Implementation	H&P Ch. 3	#1 Due, #2 Out	TCM
5	9/29	Tu	Advanced Pipelining	H&P Ch. 4		TCM
6	10/1	Th	Instr. Level Parallelism			TCM
7	10/6	Tu	Computer Arithmetic	H&P App. A		REB
8	10/8	Th	Cache Memory	H&P Ch. 5	#2 Due, #3 Out	REB
9	10/13	Tu	Virtual Memory			REB
10	10/15	Th	Alpha 21264 Case Study	Gwennap		REB
11	10/20	Tu	Multimedia Integration			REB
12	10/22	Th	Input/Output & Storage	H&P Ch. 6	#3 Due, #4 Out	REB
13	10/27	Tu	**Midterm**, Network Mechanisms	H&P Ch. 7, Keshav		ALF
14	10/29	Th	Network Protocols	Keshav		ALF
15	11/3	Tu	Multiprocessor Arch. Overview	H&P Ch. 8, CSG Ch. 1		TCM
16	11/5	Th	Parallel Programming	CSG Ch. 2 & 3	#4 Due, #5 Out	TCM
17	11/10	Tu	Cache Coherence	CSG Ch. 5, 6, & 8		TCM
18	11/12	Th	Cache Protocol Verification	Clarke		REB
19	11/17	Tu	Mem. Consistency & Latency Hiding	CSG Ch. 9.2 & 11		TCM
20	11/19	Th	Multiprocessor Interconnects	H&P Ch. 8, CSG Ch. 10	#5 Due, #6 Out	TCM
21	11/24	Tu	Synchronization, Origin Case Study	CSG Ch. 7, Laudon97		TCM
	11/26	Th	*Thanksgiving*			
22	12/1	Tu	Future Technology			REB
23	12/3	Th	Application Case Study		#6 Due	REB
	12/9	We	**Final Exam (1-4pm)**			

Bibliography

Many of the papers are available as postscript files in the directory

`/afs/cs.cmu.edu/academic/class/15740-f98/public/doc`

The file names are indicated for these documents. Copies of all other papers are on reserve in the E&S Library.

References

[Aho+86] Aho, A. V. and Sethi, R. and Ullman J. D., *Compilers*. Addison-Wesley, 1986. Background.

[Clarke+92] Clarke, E.M., *et al*, "Verification of the Futurebus+ Cache Coherence Protocol," Technical Report CMU-CS-92-206, 1992. Available as `Clarke.ps`.

[CulSin98] Culler, D., and Singh, J. P., with Gupta, A., *Parallel Computer Architecture: A Hardware/Software Approach*. Morgan Kaufman, 1998.

[Gwennap] Gwennap, L., "Digital 21264 Sets New Standard," *Microprocessor Report*, October 28, 1996. Available as `21264-article.ps`.

[HenPat96] Hennessy, J. L, and Patterson, D. A., *Computer Architecture A Quantitative Approach, 2nd Edition*. Morgan Kaufman, 1996. Main textbook.

[Kernighan+88] Kernighan, B. W., and Ritchie, D. M., *The C Programming Language, 2nd edition*, Prentice Hall, 1988. Background.

[Keshav+97] Keshav, S., *An Engineering Approach to Computer Networking*, Addison-Wesley, 1997.

[Silberschatz+91] Silberschatz, A., Peterson, J., and Galvin, T., *Operating System Concepts, 3rd Edition*, Addison-Wesley, 1991. Background.