15-740 Basic Computer Systems
Fall 1997
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-f97/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-f97/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

Mondays, Wednesdays, and Fridays, 10:30 - 11:50 a.m. in Wean Hall 5409A.

Instructors

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

Bruce M. Maggs
4123 Wean Hall
268-7654
bmm@cs.cmu.edu

Course Secretary

Joan Maddamma
7121 Wean Hall
268-7656
Joan.Maddamma@cs.cmu.edu

Textbook

Hennessy, J. L, and Patterson, D. A., *Computer Architecture A Quantitative Approach, 2nd Edition*. Morgan Kaufman, 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.

Comer, D. E., *Internetworking with TCP/IP Volume 1: Principles, Protocols, and Architecture*, Prentice-Hall, 1995. This is a new edition. There should be reference copies in the E&S Library by the time we need them.

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 cache and virtual memory
4. Input, output, and storage, particularly with respect to supporting multimedia applications.
5. Shared memory multiprocessors.
6. Networking, including local and wide area networks
7. Parallel machine interconnection networks
8. Control parallel programming and machines
9. Data parallel programming and machines, including vector machines

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 or the liason Thomas Gross (`tg@cs.cmu.edu`).

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. However, we require you to write up your solutions individually, as well as to identify your collaborators in your writeup.

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 successfully complete the course with some deficiencies in their backgrounds. If you feel uncertain 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 MIPS architecture as our basis. This instruction set is very similar to DLX. It is documented more completely in [Kane92]. 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].

Shared memory Bus and directory-based schemes. Consistency models [HenPat96, Chs. 8.3–8.6, 8.8, and App. E].

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 [Comer95, Chs. 3–13]

Processor Interconnection Networks

Topologies Direct (hypercube, torus, ring) and indirect (butterfly) networks [Dally90].

Routing Virtual channels, deadlock avoidance [Leighton92] and [Dally90]

General Issues in Parallel Computing

Applications and technology [HenPat96, Chs. 8.1–8.2, 8.7, and 8.9–8.11].

Data Parallel Computing

Programming Model Algorithms and linguistic expression [Blelloch96]

Vector machines Vector machine design [HenPat96, App. B], memory subsystem [Smith+91].

Control Parallel Computing

Dynamic Threads Cilk [Cilk].

Message-passing processes PVM [Sunderam+94].

5. Schedule

The following is a tentative schedule. There might be some variations.

Class	*Date*	*Day*	*Topic*	*Reading*	*Homework*	*Who*
1	09/15	Mon	Introduction & Overview	H&P Ch. 1	#1 Out	BMM
2	09/17	Wed	MIPS/DLX programming I	H&P Ch. 2, Kane92		REB
3	09/19	Fri	MIPS/DLX programming II			REB
4	09/22	Mon	Instruction set comparison	H&P App. C, D		REB
5	09/24	Wed	Pipelined Implementation I	H&P Ch. 3.1–3.5	#1 Due, #2 Out	REB
6	09/26	Fri	Pipelined Implementation II	H&P Ch. 3.6–3.12		REB
7	09/29	Mon	Advanced Pipelining I	H&P Ch. 4.1-4.2		REB
8	10/01	Wed	Instruction Level Parallelism	H&P Ch. 4.5, 4.7		REB
9	10/03	Fri	Advanced Pipelining II	H&P Ch. 4.3, 4.4, 4.6		REB
10	10/06	Mon	Case Study	Gwennap	#2 Due, #3 Out	REB
11	10/08	Wed	Arithmetic Properties	H&P App. A.1–A.7		BMM
12	10/10	Fri	Arithmetic Implementation	H&P App. A.8–A.12		BMM
	10/13	Mon	*Mid-Semester Break*			
13	10/15	Wed	Memory Technology	H&P Ch. 1.7, 6.2		REB
14	10/17	Fri	Cache Memory	H&P Ch. 5.1–5.7	#3 Due	BMM
15	10/20	Mon	**Midterm Exam**		#4 Out	
16	10/22	Wed	Virtual Memory	H&P Ch. 5.8–5.13		BMM
17	10/24	Fri	Busses & I/O	H&P Ch. 6		BMM
18	10/27	Mon	Multimedia Computing			REB
19	10/29	Wed	Network technology I	H&P Ch. 7.1–7.8		BMM
20	10/31	Fri	Network technology II	H&P Ch. 7.9–7.13		BMM
21	11/03	Mon	Network Protocols	Comer Ch. 3–13	#4 Due, #5 Out	BMM
22	11/05	Wed	Interconnection networks I	H&P Ch. 7		BMM
23	11/07	Fri	Interconnection networks II			BMM
24	11/10	Mon	Message routing	Leighton, Dally		BMM
25	11/12	Wed	Parallel Machines	H&P Ch. 8.1–8.2		BMM
26	11/14	Fri	Data Parallel Programming	Blelloch		BMM
27	11/17	Mon	Control Parallel Programming	Cilk, Sunderam	#5 Due, #6 Out	BMM
28	11/19	Wed	Shared Memory Implementation	H&P Ch. 8.3–8.4		REB
29	11/21	Fri	Cache Protocol verification	Clarke		REB
30	11/24	Mon	Memory Consistency	H&P Ch. 8.5–8.7		REB
	11/26	Wed	*Thanksgiving vacation*			
	11/28	Fri	*Thanksgiving vacation*			
31	12/01	Mon	Processor/Network Interfaces	H&P Ch. 8.8–8.11		BMM
32	12/03	Wed	Vector Machines	H&P App. B, Smith		REB
33	12/05	Fri	Future Technology		#6 Due	REB
	12/09	Tue	**Final Exam (1–4pm)**			

Bibliography

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

`/afs/cs.cmu.edu/academic/class/15740-f96/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.

[Blelloch96] Blelloch, G. E., "Programming Parallel Algorithms," *Communications of the ACM*, March, 1996. A postscript version is available as `Blelloch.ps`. An online version is available as

`http://web.scandal.cs.cmu.edu/cacm.html`

[Cilk] *Cilk 5.1 Reference Manual*, pp. 1–48. Massachusetts Institute of Technology, September, 1997. Available as `Cilk.ps`

[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`.

[Comer95] Comer, D. E., *Internetworking with TCP/IP Volume 1: Principles, Protocols, and Architecture*, Third Edition, Prentice-Hall, 1995

[Dally90] Dally, W. J., "Network and Processor Architectures for Message-Driven Computers," *VLSI and Parallel Computation*, Suaya, R., and Birtwistle, G., *eds.*, Morgan Kaufmann, 1990, Chapter 3.1, pp. 140–169.

[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.

[Kane92] Kane, G., and Heinrich, J., *MIPS RISC Architecture*, Prentice-Hall, 1992. On reserve in E&S Library.

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

[Leighton92] Leighton, T., "Methods for Message Routing in Parallel Machines," *24th ACM Symposium on Theory of Computing (STOC '92)*, May, 1992, pp. 77–96.

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

[Smith+94] Smith, J. E., and Weiss, S., "PowerPC 601 and Alpha 21064: A Tale of Two RISCs," *IEEE Computer*, June, 1994, pp. 46–58.

[Smith+91] Smith, J. R., and Taylor, W. R., "Accurate Modeling of Interconnection Networks in Vector Supercomputers," *1991 International Conference on Supercomputing*, ACM, 1991, pp. 264–272.

[Sunderam+94] Sunderam, V. S., Geist, G. A., Dongarra, J., and Manchek, R., "The PVM Concurrent Computing System: Evolution, Experiences, and Trends," 1994. Available as `Sunderam.ps`.