# 15-740 Basic Computer Systems
# Fall 1995
# 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-f95/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-f95/www/home.html
```

You can obtain copies of all handouts (including assignments), track the course schedule, and learn all kinds 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, 1:00 - 2:50 p.m. in 5409 Wean Hall

**Instructors**

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

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

**Teaching Assistant**

Yirng-An Chen
7205 Wean Hall
268-3620
yachen@cs.cmu.edu

**Course Secretary**

Joan Maddamma
7124 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.

**Non-CS Students**

If you are not a CS graduate student, 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 textbook. The second part deals with systems consisting of multiple, connected processors to support networking and parallel computing. Material for this part of the course will include both the textbook and 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
5. Networking, including local and wide area networks
6. Parallel machine interconnection networks
7. Data parallel programming and machines, including vector machines
8. Shared memory multiprocessors
9. Control parallel programming and machines

**Policies**

Grades will be based on homeworks and a final exam.

For the homework assignments, we encourage limited collaboration among members of the class. Namely, we will let you work in groups of up to three people in solving the homework problems. However, we will 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 counts for 35%, the homework for 65%.

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]. 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** IBM/Motorola Power and PowerPC architectures and implementations [HenPat96, Ch. 4.8], and [Smith+94]. This architecture blends RISC and CISC concepts, and there are several generations of implementations to compare.

### 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].

### 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 [Comer91, Chs. 1–10]

### 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** [Trew+91], and [HenPat96, Chs. 8.1–8.2, 8.7, and 8.9–8.11].

**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]

## Data Parallel Computing

**Programming Model** Algorithms and linguistic expression [Blelloch95]

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

**Other hardware support** The CM-5 [Leiserson+93].

## Control Parallel Computing

**Dynamic Threads** Cilk [Blumofe+95].

**Message-passing processes** PVM [Sunderam+94] and MPI [Dongarra+95].

**Message-passing machines** [TBA]

## 5. Schedule

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

| Class | Date | Day | Topic | Reading | Homework | Lecturer |
|---|---|---|---|---|---|---|
| 1 | 09/19 | Tu | Introduction & Overview | H&P Ch. 1 | #1 Out | REB |
| 2 | 09/21 | Th | MIPS/DLX programming | H&P Ch. 2 | | REB |
| 3 | 09/26 | Tu | Instruction set comparison | H&P App. D | | REB |
| 4 | 09/28 | Th | Pipelined Implementation | H&P Ch. 3 | #1 Due, #2 Out | REB |
| 5 | 10/03 | Tu | Advanced Pipelining | H&P Ch. 4 | | REB |
| 6 | 10/05 | Th | Instr. level parallelism | | | REB |
| 7 | 10/10 | Tu | Case Study | H&P App. C, Smith94 | | REB |
| 8 | 10/12 | Th | Computer Arithmetic | H&P App. A | #2 Due, #3 Out | YAC |
| 9 | 10/17 | Tu | Cache Memory | H&P Ch. 5 | | BMM |
| 10 | 10/19 | Th | Virtual Memory | | | BMM |
| 11 | 10/24 | Tu | Input/Output & Storage | H&P Ch. 6 | | REB |
| 12 | 10/26 | Th | Network mechanisms | H&P Ch. 7 | #3 Due, #4 Out | BMM |
| 13 | 10/31 | Tu | Network Protocols | Comer, Ch. 1--10 | | BMM |
| 14 | 11/02 | Th | Parallel computing overview | Trew | | BMM |
| 15 | 11/07 | Tu | Interconnection networks | H&P Ch. 7 | | BMM |
| 16 | 11/09 | Th | Message routing | Leighton, Dally | #4 Due, #5 Out | BMM |
| 17 | 11/14 | Tu | Data Parallel Programming | Blelloch | | BMM |
| 18 | 11/16 | Th | Vector Machines | H&P App. B, Smith91 | | BMM |
| 19 | 11/21 | Tu | Data parallel machines | Leiserson | | BMM |
| | 11/23 | Th | Thanksgiving vacation | | | |
| 20 | 11/28 | Tu | Shared Memory Machines | H&P Ch. 8 | #5 Due, #6 Out | REB |
| 21 | 11/30 | Th | Cache Protocol verification | Clarke | | REB |
| 22 | 12/05 | Tu | Control Parallel Programming | Blumofe | | BMM |
| 23 | 12/07 | Th | Message passing machines | Dongarra, Sunderam | #6 Due | BMM |
| | 12/11 | M | Final Exam | | | |

## Bibliography

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

`/afs/cs.cmu.edu/academic/class/15740-f95/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.

[Blelloch95] Blelloch, G. E., “Programming Parallel Algorithms,” *Communications of the ACM*, to appear. Available as `Blelloch.ps`.

[Blumofe+95] R. D. Blumofe, *et al*, "Cilk: An Efficient Multithreaded Runtime System," *Principles and Practice of Parallel Programming (PPoPP '95)*, 1995. Available as `Blumofe.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`.

[Comer91] Comer, D. E., *Internetworking with TCP/IP Volume 1: Principles, Protocols, and Architecture*, Prentice Hall, 1991, pp. 1–158. In reference section of E&S Library, Call Number 004.6C73i2.

[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.

[Dongarra+95] Dongarra, J. J., Otto, S. W., Snir, M., and Walker, D., "An Introduction to the MPI Standard," April, 1995. Available as `Dongarra.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.

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

[Leiserson+93] Leiserson, C., *et al*, "The Network Architecture of the Connection Machine CM-5," *ACM Symposium on Parallel Algorithms and Architectures (SPAA '92)*, June, 1992, pp. 272–285. Available as `Leiserson.ps`

[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`.

[Trew+91] Trew, A., and Wilson, G., *Past, Present, and Parallel: A Survey of Available Parallel Computing Systems*, Springer-Verlag, 1991, pp. 1–12.