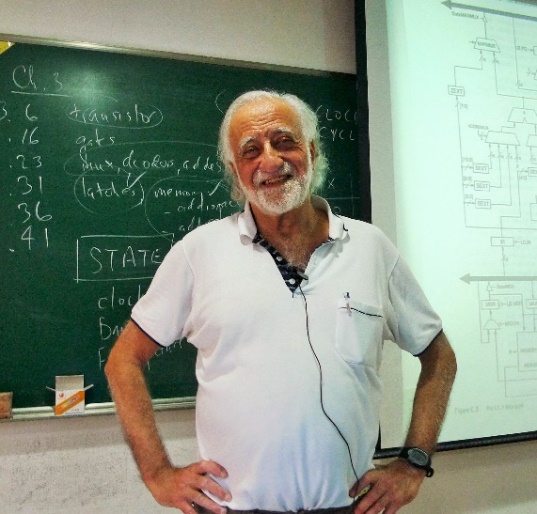
**计算机系统概论**

**Introduction to Computing Systems**



课程编号(Course No.): 21121420

学分(credit)：4.0

上课时间(time): **July 11 –July 28, 2023**

上课方式(mode): Zijingang Campus East2

面向对象(students)：students of CS, CE, SE, EE

预修要求(prerequisites)：no computer pre-requisites.

( Better have learned C programming)

课程网站(website):　 https://courses.zju.edu.cn/

**授课教师 (Professor):**

**--------------------------------------------------------------------------------------------------------------------------**

**Yale N. Patt**

Professor, Department of Electrical and Computer Engineering, University of Texas at Austin

<http://users.ece.utexas.edu/~patt>

**教授简介(Introduction to Yale Patt):**

**-------------------------------------------------------------------------------------------------------------------------**

Yale Patt教授，1939年生，现为德州大学奥斯汀分校电子与计算机工程学院教授，是工程学科的Ernest Cockrell, Jr. Centennial chair。**美国工程院院士，IEEE Fellow, ACM Fellow, 被IEEE Spetrum称为美国计算机界的卓越泰斗（luminary）,在美国乃至世界计算机体系结构领域有着广泛的影响力。**

他热爱并倾心教学，40余年教学生涯中，他几十年坚持为电子工程、计算机、计算机工程等专业大学新生上计算机系统入门课程、为本科生开设计算机组成课程、为高年级本科生和研究生开设好性能处理器设计课程。他编写的教材“计算机系统概论－从比特、门到C程序及更上层的系统”，站在计算机整体系统的高度将软硬件连贵起来进行阐述，为计算机后续专业课程打下基础，不仅是密歇根大学计算机专业的经典基础教材，也是美国多所知名大学，如德州大学、莱斯大学、明尼苏达大学、乔治亚理工学院、伊利诺伊大学和西北大学等校的计算机专业基础教材。他通过会议专题报告或公司短期教程的方式为计算机工业界教授不下5000位工程人员。鉴于他对计算机教育的杰出贡献，2000年获ACM Karl V. Karlstrom Outstanding Educator Award, 奖励他在发展计算机科学教育上，作为一个教师的杰出教学能力、奉献精神和突出成就。

他的研究兴趣始终定位于未来5-10年的计算机系统主流产品技术，并指导研究生设计这样的未来产品。1965年他首次研发WOS模块，这是第一个在单一硅片上上实现的复杂逻辑门电路模块。1984年他研发的HPS（高性能处理器系统）通过结合使用宽发射技术、进取型转移预测、动态调度、乱序执行、检查点按序交付（实现精确中断）多种技术，开发指令级并行性，他还提出了两级自适应转移预测技术。这些技术目前已被广泛应用在现代微处理器中。目前他仍在研究2018年的微处理器技术，包括可重配置的异构多核处理器技术，处理器器核与DRAM之间的高性能接口技术，通用GPU技术，多核环境下的高效预取技术，以及实时系统的高效用技术等。

Patt教授于2012年来浙大第一次上这门课，鉴于此课的重要性，也因为他对浙大学生的认可，他于2014起（除2018年）都来杭授课，授课效果一级棒！最近因两年疫情，采用的是Patt教授通过Zoom线上授课，实验课分小班线下由助教辅导、答疑、实验并验收。Patt教授的标准美语清晰易懂，讲课形象、生动、风趣，深获学生好评。总之，花暑期三周时间上这门课肯定不会后悔的。

**部分重要的世界级荣誉与获奖：**

**-------------------------------------------------------------------------------------------------------------------------**

**2021 IEEE HPCA 2021 Test of Time Award**, for "Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors," (HPCA, 2003), with Onur Mutlu, Jared Stark, and Chris Wilkerson. Aw arded, February, 2021.

**2014年，IEEE/ACM Inaugural Test of Time Influential Paper Award**，奖励在微处理器技术的前25年（(1968-1992)中最有影响力的10篇论文，期中Patt教授与人合作的论文占了4篇。

**2007年，IEEE/ACM ISCA最具影响力论文奖，**奖励他在ISCA’1992上发表的关于二级自适应转移预测器的论文，在过去15年中对微处理的研究、开发、产品设计及思想等产生的最重要影响。

**1996年，IEEE/ACM Eckert-Mauchly Award**，奖励他对于指令级并行性技术和超级标量处理器设计的重要贡献；**（计算机系统结构方向的最高奖）**；

**2016年，Benjamin Franklin Medal in Computer and Cognitive Science**。. 由本杰明.弗兰克林研究院颁发的奖励科学与工程七个领域的杰出人员（自1824年起）的富兰克林奖章。

**2000年，ACM Karl V. Karlstrom Outstanding Educator Award**, 奖励他在发展计算机科学教育上，作为一个教师的杰出教学能力、奉献精神和突出成就。

**1995年，IEEE Emannuel R. Piore Medal**，奖励他对于应用于商用高性能处理器的计算机体系结构方面的重要贡献；（IEEE技术奖章，信息处理领域）

**2013年，IEEE Harry H. Goode Award,** 奖励他在近半个世纪以来对信息处理技术的重要贡献，包括对于微处理器设计的远见卓识，突破性的教材，指导未来的领导者。

**1999年，IEEE Wallace W. McDowell Award**，奖励他通过计算机工程和教育双方面的成就对高性能微处理器设计工业产生重要影响；

**2011年，IEEE B. Ramakrishna Rau Award, 首位获奖者**，奖励他在微处理器设计领域的教学、指导、研究和服务多方面的重要贡献和积极的领导作用。

**2005年，IEEE Charles Babbage Award，**奖励他对于高性能处理器设计的根本性贡献。

**2000年，Outstanding Lecturer of the year,** National ACM lectureship Program(ACM杰出教师)。

**1996年，IEEE Computer Society Golden Core Award。**

**2016年，ACM SigMICRO 杰出服务奖。**

**2000-01，National ACM讲座计划，杰出讲师。**

**1998-99，National ACM讲座计划，杰出讲师。**

**课程简介(Course overview)：**

**--------------------------------------------------------------------------------------------------------------------------**

这是面向计算机工程、电子工程专业学生的第一门专业课程。课程旨在为同学们学习计算机科学和工程的一系列后续专业课程打下坚实的基础。课程的指导思想是，在专业教育的初期就让学生对计算机系统有一个较为完整的理解，对于后续的计算机组成与体系结构、操作系统、数据库、网络、算法设计以及软件工程等课程的深入理解有很大的帮助。课程将采用一种动机驱动的自底向上的授课方法：即在引入和概要介绍一些概念及其重要性之后，把这些新的概念与同学已经理解的知识关联起来。从作为开关的晶体管开始，到构建逻辑门，再到更复杂的逻辑结构，然后是带门控的锁存器，最后到存储器的实现。然后学习计算机的指令周期，特定的小教学样机计算机LC-3的实现。样机LC-3虽小，但五脏俱全，具有现代处理器的重要结构，同时足够简单以保证学生能完全理解计算机原理。第一次的编程作业是关于LC-3的机器指令，然后介绍汇编语言以及汇编程序是如何工作的。后续的编程作业将用到LC-3的汇编语言。课程还会介绍好的编程风格、程序调试。提供LC-3的模拟器让学生调试、测试自己的汇编程序。用LC-3汇编程序写的系统服务例程执行I/O操作，可以实现从键盘输入数据并在显示器上输出结果。用户程序通过自陷（Trap）指令和自陷向量调用服务例程。最后加入子程序调用和返回指令使得LC-3的指令系统是一个完备的指令系统。

This is the first course in computing for students of computer engineering and electrical engineering. The objective is to provide a strong foundation that a serious student can build on in later courses across the spectrum of computer science and engineering. The idea is that a more complete understanding of the fundamentals early in your education will help you acquire a deeper understanding of more advanced topics later, whether that topic is in computer architecture, operating systems, data base, networks, algorithm design, software engineering, or whatever. I call the approach "motivated" bottom-up. That is, after providing some overview of why a new concept is important, we attempt to tie that new concept to what you already understand. Starting with the transistor as a switch, we build logic gates, then more complex logic structures, then gated latches, culminating in an implementation of memory. From there, we study the computer's instruction cycle, and then a particular computer, the LC-3 (for Little Computer 3). We got it wrong the first couple of times! The LC-3 captures the important structures of a modern computer, while keeping it simple enough to allow full understanding. The first programming assignment is in the machine language of the LC-3. From there, we move up to Assembly Language, and learn how an assembler works. The remaining programming assignments are in LC-3 Assembly Language. We cover good programming style and practice, and teach debugging from the gitgo. An LC-3 Simulator allows the student to debug his/her own programs. Input (via the keyboard) and output (via the monitor) both use physical device registers. System service routines, written in LC-3 Assembly Language are used to perform I/O functions. They are invoked by user programs by the TRAP instruction and corresponding trap vector. Subroutine calls and returns complete the LC-3 instruction set.

**课程内容(Course contents):**

July 11, 14：00-17：30 Lec1: Chapter 1,2. (Professor Patt, )

July 12, 14：00-17：30 Lec2: Chapter 3. (Professor Patt)

July 13, 14：00-17：30 Lec3: Chapter 4 and first half of Chapter 5. (Professor Patt)

July 14, 14：00-17：30 Lec4: Rest of Chapter 5 and Chapter 6. (Professor Patt)

July 15, Weekend no class

July 16, Weekend no class

July 17, 14：00-17：30 Lec5: Chapter 7. (Professor Patt)

July 18, 14：00-17：30 Lec6: Chapter 8，16.（Stack， Arrays）(Professor Patt)

July 19, 14：00-17：30 Lec7: Chapter 8，19. （Character Strings， Linked lists）(Professor Patt)

July 20, 14：00-17：30 Lec8: Chapter 9. I/0 (Professor Patt)

July 21, 14：00-17：30 Lec9: Chapter 9. Trap，Interrupts (Professor Patt)

July 22, Weekend no class

July 23, Weekend no class

July 24, 14：00-17：30 Lec10: Chapter 10. Calculator(stack arithmetic, data conversion) (Professor Patt)

July 25, 14：00-17：30 Lec11: Chapter 8. Recursion. (Professor Patt)

July 26, 14：00-17：30 Lec12: Chapter 16,19 revisited, but with C (Professor Jiang)

July 27, 14：00-17：30 Lec13: Chapter 14, revisited, but with C (Professor Jiang)

July 28, 14：00-17：30 Lec14: Chapter 17, revisited, but with C (Professor Jiang)

final exam 14:00-16:30 arranged in the afternoon on First Friday of fall semester 2023

**实验与课程作业(Lab assignments & Homeworks)：**

**-------------------------------------------------------------------------------------------------------------------------**

Homework will be assigned in the class and lab assignments (programs) will used to reinforce the material in lecture.

Labs(Q&A) will be in the classrooms each class day in the morning 9:00am—11:30am from July 11 to July 28 except weekend.  
Lab0 Introduction to LC3 simulation, learn to use the simulation platform

Lab1, small program in machine language to get started and run on LC3   
Lab2, data structure problem in LC-3 assembly language running on LC3  
Lab3, data structure problem in LC-3 assembly language running on LC3  
Lab4, recursive problem, in LC-3 assembly language running on LC3  
Lab5, Prog 2 done in C ( Array and pointer)  
Lab6, due July 30, Prog 4 done in C (recursive problem)

**课程成绩(Grading):**

**-------------------------------------------------------------------------------------------------------------------------**

Final examination 40%

Homework 20%

Lab Assignments: 40%

**课程教材(Textbook):**

**-----------------------------------------------------------------**

《Introduction to Computing System—from bits and gates to C and beyond.》3rd edition.

Writer: Yale N. Patt and Sanjay J. Patel;

Publisher: China Machine Press.

**Contents of the Textbook:**

**------------------------------------------------------------------**

Ch1 Welcome Aboard   
Ch2 Bits, Data Types, and Operations   
Ch3 Digital Logic Structures   
Ch4 The Von Neumann Model   
Ch5 The LC-3   
Ch6 Programming   
Ch7 Assembly Language   
Ch8 I/O   
Ch9 TRAP Routines and Subroutines   
Ch10 And, Finally...The Stack  
Ch11 Introduction to Programming in C   
Ch12 Variables and Operators   
Ch13 Control Structures   
Ch14 Functions   
Ch15 Testing and Debugging   
Ch16 Pointers and Arrays   
Ch17 Recursion   
Ch18 I/O in C   
Ch19 Data Structures

Appendix A The LC-3 ISA  
Appendix B From LC-3 to x86   
Appendix C The Microarchitecture of the LC-3   
Appendix D The C Programming Language   
Appendix E Useful Tables   
Appendix F Selected Solutions (available at website)

Lec1： Chapter1 Welcome Aboard   
Chapter2 Bits, Data Types, and Operations

Lec2, Chapter3 Digital Logic Structures

Lec3, Chapter4 The Von Neumann Model

Chapter5-1 The LC-3

Lec4, Chapter5-2 The LC-3

Chapter6 Programming

Lec5, Chapter7 Assembly Language

Lec6 Chapter16. Data structures （ pointers and arrays）

Lec7 Chapter19. Data structures, continued

Lec8 Chapter8. I/O

Lec9 Chapter9. TRAP Routines and Subroutines

Lec10 Chapter10. And, Finally...The Stack

Lec11 Chapter14 Functions

Chapter17. Recursion

Lec12 Chapter16,19 revisited, but with C

Lec13 Chapter14 revisited, but with C

Lec14 Chapter17 revisited, but with C

第一课 第一章 欢迎词   
第二章 位、数据表示和操作

第二课 第三章 数字逻辑结构

第三课 第四章 冯 诺依曼模型

第五章-1 LC-3模型机

第四课 第五章-2 LC-3模型机

第六章 程序设计

第五课 第七章 汇编语言

第六课 第十六章. 数据结构（ 指针和数组）

第七课 第十九章 数据结构（续）

第八课 第八章 I/O

第九课 第九章 TRAP例程和子程序

第十课 第十章 栈

第十一课 第十四章 函数

第十七章. 递归

第十二课 第16/19章 数据结构（用C语言）

第十三课 第十四章 函数（用C语言）

第十四课 第十七章 递归（用C语言）