Computer Architecture
----A Quantitative Approach

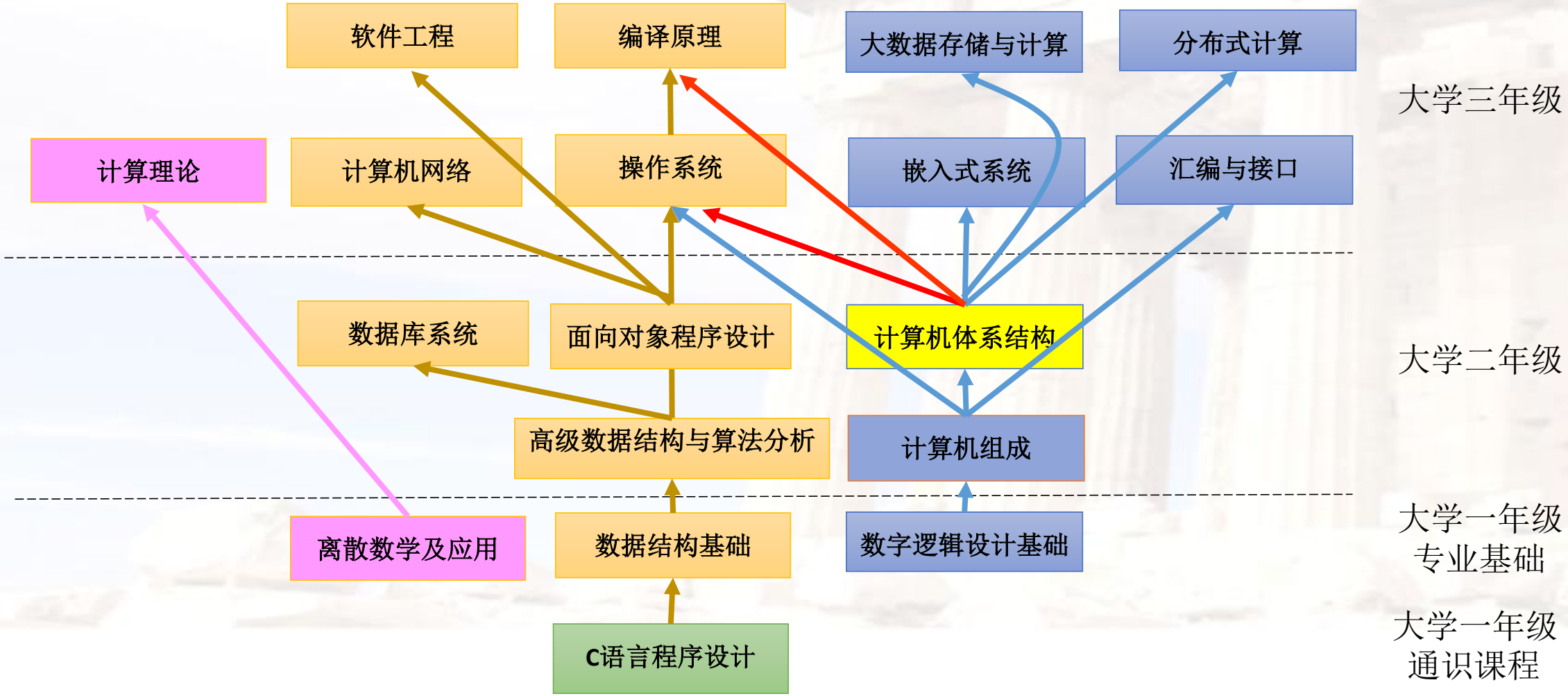
College of Computer Science & Technology
Jiang, Xiaohong
2024.fall

浙江大学 ZHEJIANG UNIVERSITY

Instructor & TA

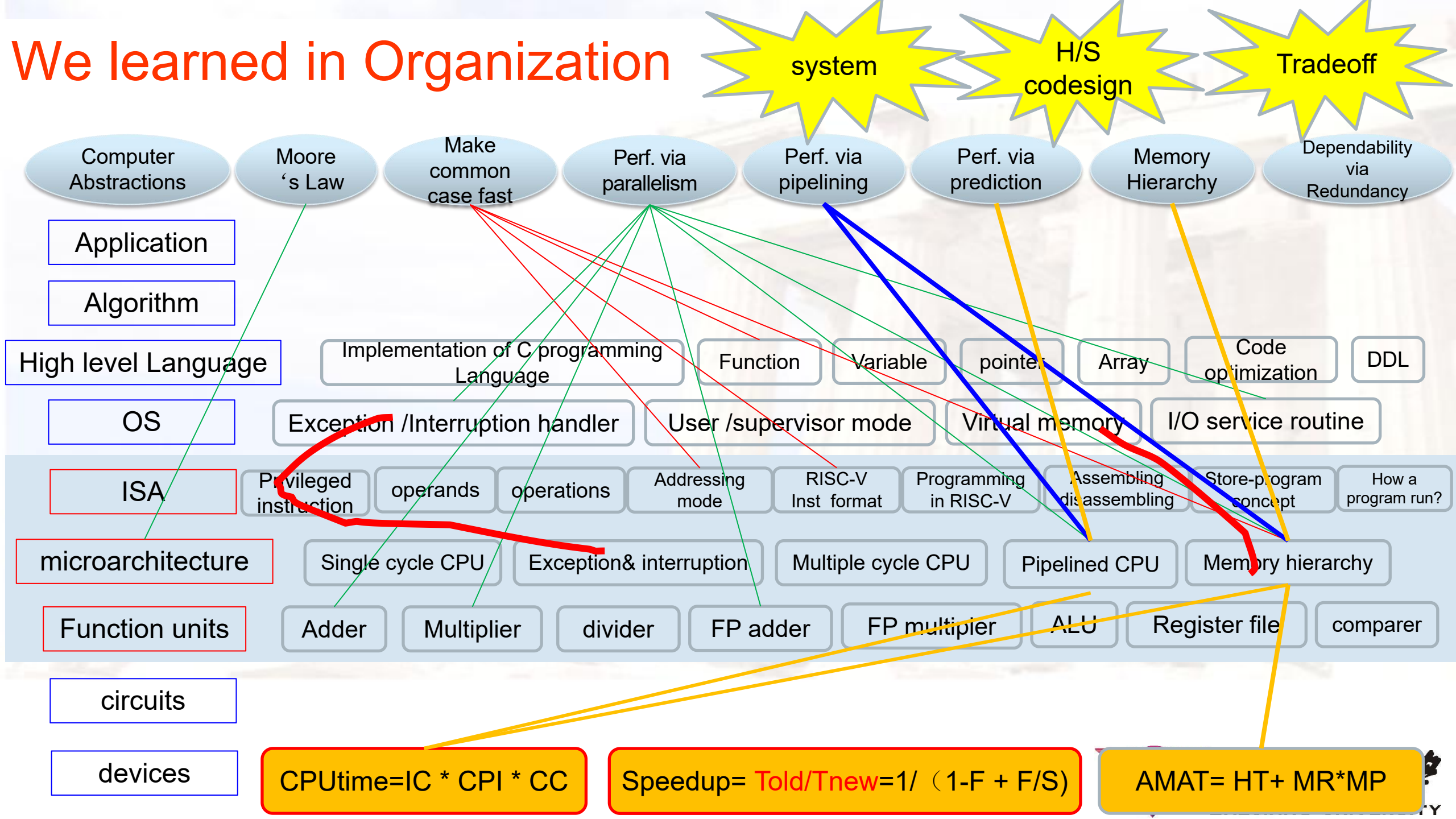
- **Instructor：** 姜晓红Jiang Xiaohong
  - Office : **Room520, Bld. of CaoGuangBiao,**
  - Mobile(short): 529114
  - Email: jiangxh@zju.edu.cn
  - Homepage: http://mypage.zju.edu.cn/jiangxh
- **TA：** 简英钊， 微信：shadow-_-dream
  黄鸿宇， 微信：skiwater_touchfish
- **Course Website:**
  - course.zju.edu.cn

Why we learn Computer Architecture?



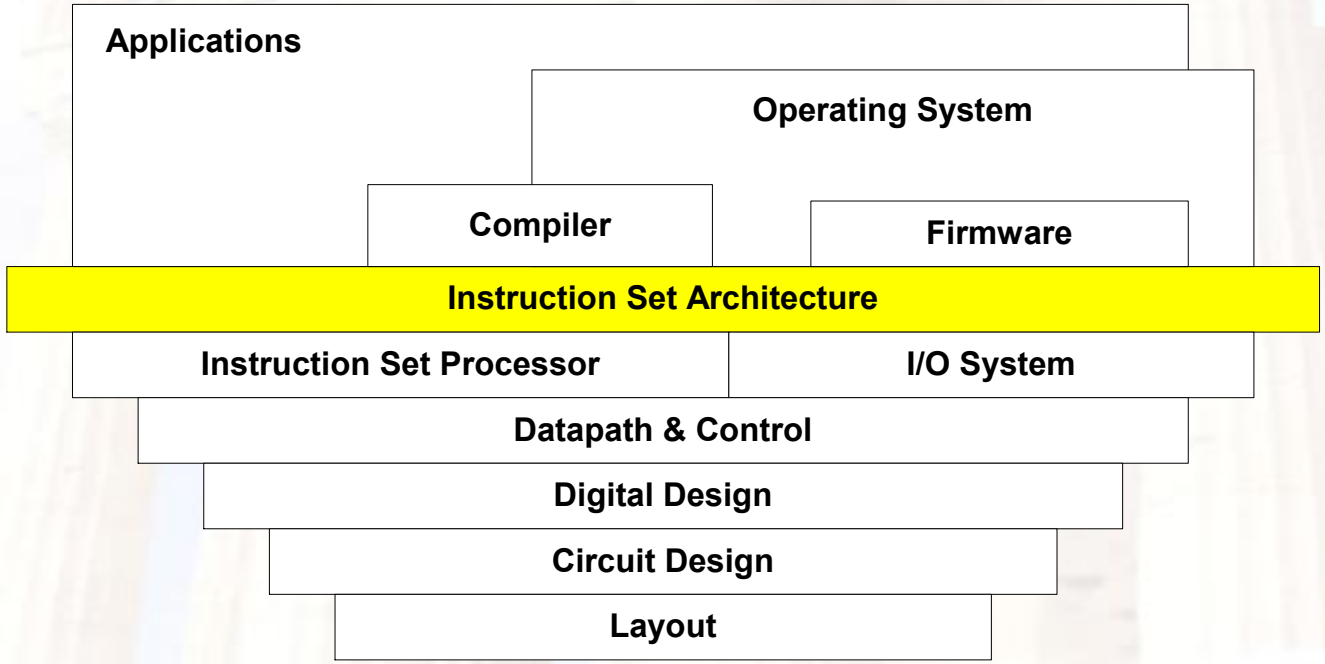
浙江大学 ZHEJIANG UNIVERSITY

We learned in Organization



Different from Organization

- **Simple pipelined-CPU is far away from the CPU of the real world.**
- **What we have learned is not enough for future research and development in Computer System**



- Part & **Whole**
- Bottom up & **Top down**
- How & **Why**

Course Objectives

The objective of this course

- systemically learn the fundamental concepts and design approaches of computer architecture using quantitative approaches from the view of the whole computer system.
- Learn the ideas and approaches to improve the performance of memory system(ch2).
- Learn the ideas and approaches to improve the performance of CPU via exploring ILP(ch3), DLP(ch4), and TLP(ch5).
- Grasp the hardware design tools and environment, design and implement the hardware with Verilog language in vivado environment on FPGA board.

Textbook

David A. Patterson, John L. Hennessy,

《Computer Architecture
– A Quantitative Approach》

6th Edition. July , 2019.
China Machine Press,
ISBN: 978-7-111-63110-1

John L. Hennessy （Stanford）

https://engineering.stanford.edu/people/john-hennessy

- Former President of Stanford University during 2000 – 2016 （17 billion）
- Current Alphabet Chairman
- "Godfather of Silicon Valley,“
- In 1981, Hennessy initiated a project at Stanford that focused on a simpler computer architecture known as RISC. During a sabbatical leave in 1984-85 he cofounded MIPS Computer Systems, now known as MIPS Technologies, which specializes in the production of microprocessors SPARC.
- Received Eckert-Mauchly Award in 2001
- Received Turing Award in 2017

David A. Patterson （UC Berkeley）

- UC Berkeley （1976 – 2016）
- Currently Google TPU

- He led the design and implementation of **RISC** I (the foundation of the SPARC architecture)
- Inventor of **RAID**
- involved in the Network of Workstations (NOW) project
- Research Accelerator for Multiple Processors (RAMP)
- Received ACM Eckert-Mauchly Award in ISCA 2008
- Received Turing Award in 2017

https://people.eecs.berkeley.edu/~pattrsn/

Text book evolution

1-3 edition	4-5th edtion	6th edtion
• MIPS • ILP （Instruction Level Parallelism） • Cost / Performance • PC	• MIPS / ARM • ILP （Instruction Level Parallelism） DLP（Data Level Parallelism） TLP（Thread Level Parallelism） • Cost / Performance dependability / power • PC embedded system Server	• RISC V • ILP（Instruction Level Parallelism） DLP (Data Level Parallelism） TLP（Thread Level Parallelism） RLP（Request LevelParallelism） • dependability Cost /Performance/power • PC Personal mobile device warehouse-scale computers domain-specific architecture

Updated Course Contents

教材： 5th Edition

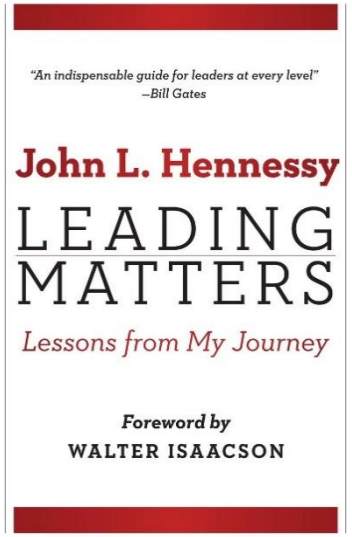
Contents in 2020	备注
Ch1 Fundamentals of computer design	
AppA Instruction Set Principles	MIPS
AppC Pipelining: Basic and Intermediate Concepts	Pipeline CPU
AppB Review of Memory Hierarchy	
Ch2 Memory Hierarchy Design	Improve cache perf.
AppD Storage Systems	
AppI Basic concepts of Multiprocessor	

教材： 6th Edition

Contents in 2021	备注
Ch1 Fundamentals of computer design	1.1-1.13
Ch2 Memory Hierarchy Design	2.1-2.9, app B3
Ch3 ILP and its expoloitation	App C2-C6，3.1-3.14 Branch prediction Dynamic scheduling Speculation
Ch4 DLP in Vector, SIMD, GPU	4.1-4.9 Vector. SIMD. GPU
Ch5 Thread Level Parallelism	5.1-5.11 Multiprocessor Cache coherence Synchronization
Ch6 Warehouse-scale computer to explore RLP & DLP	选讲
Ch7 Domain-specific Arch.	选讲

2018 interview

For pioneering a systematic, quantitative approach to the design and evaluation of computer architecture with enduring impact on the microprocessor industry.



This book is essential reading for those tasked with leading any complex enterprise in the academic, not-for-profit, or for-profit sector.

https://baijiahao.baidu.com/s?id=1607140815624945357&wfr=spider&for=pc

How ?

- Concepts, Ideas and Principles
- Quantitative approaches
- Hit the problem and right way to solve problem

- ***As a man sows, so he shall reap.***

  一分耕耘一分收获

Grading Policy:

- 16-20%: written homework
- 8-12%: pop quiz 2-3 times

28%

- 32%: Lab assignments
- 5-10%: Bonus

60% (<=60)

- lab grade = report (40%) + check(60%)

- 40%: Final exam
  - (close-book test with one A4 memo)

- Final grade = 40%(Final exam） + 60% (other <=100)

Lectures vs. Labs

Week	Theory (lectures)	Labs	%
1	Ch0: Course Introduction.	Lab1: Enhance CPU with forwarding and predict-NT	6
2	Ch1-1: CA design Lifecycle		
3	Ch1-2: trends of CA engineering		
4	Ch2-1: cache optimization	Lab2: Adding exception to CPU in Lab1	4
5	Ch2-2: VM and VA		
6	Ch3-0: ILP--supporting multicycle OP	Lab3: implement a 2-way set associative cache	3
7	Ch3-1: scoreboard		
8	Ch3-2: Tomasulo	Lab4: adding cache to CPU of Lab2	4
9	Ch3-3: hardware-based speculation		
10	Ch3-4: multiple issue processor	Lab5：expending CPU to support multicycle OP	7
11	Ch4-1: DLP—vector processor		
12	Ch4-2: SIMP，GPU		
13	Ch5-1: TLP-multiprocessor	Lab6：implement dynamic scheduling: scoreboard/Tomasulo	8
14	Ch5-2: cache coherence		
15	Ch5-3: synchronization		
16	Review		

Homeworks (16-20%)

- Total 4-5 times, once per chapter
- Submission deadline will be normally one week after assigned, and will be announced on **course website**.
- For doing homework, discussion is greatly encouraged, but every student is required to **Do and Submit** the homework individually on time.
- Submission Naming rule
  - StID_name_hw1.doc

Lab assignments

- Lab1----Implement pipelined CPU with forwarding paths and prediction-not-taken supporting RISC V 32i instructions.
- Lab2----Implement Interruption and Exception on CPU of Lab 1).
- Lab3----Implement a 2-way associative cache
- Lab4----Adding the cache into the pipelined CPU in Lab 2).
- Lab5----Expend the pipelined CPU to support multi-cycle operations: integer multiplier, integer divider / remainder operation, issue in order and complete out of order, detecting pipeline hazards (WAW, RAW)
- Lab6----Implement the Dynamic Scheduling (Scoreboard or Tomasulo), implement a dynamic scheduling pipelined CPU.

Labs (32%)

- Grading (report 40% + check 60%)/per lab
  - Lab1 3 weeks, 6% personally
  - Lab2 2 weeks, 4% personally
  - Lab3 1 weeks, 3% personally
  - Lab4 2 weeks, 4% personally
  - Lab5 3 weeks, 7% in group of 2 members
  - Lab6 4 weeks, 8% in group of 2 members
- You are encouraged to optimize the CPU performance in Lab1/2/4. You can get bonus if CPU performance is clearly improved（ >=10%）
- Each lab have up to 2 times to be checked. The 2nd check will have score <= 90.

Submission Policy:

- All the homeworks (individually) and lab reports (in group) are required to be submitted to the course website on time.
- Lab report and lab code are required to submit separately using different links under different deadline.
- Submission deadline will be announced on course website.
- All assignments in this course should be turned in by the specified due date. Late submission will be disgraded. (-10 points per day)

Honest Policy

- Be HONEST in your work!
- Found copy & be copied in the homework or lab report，you get ZERO for one submission and also get 10% off in the final grade !

Q&A

??