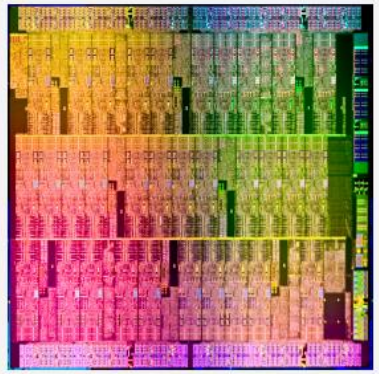
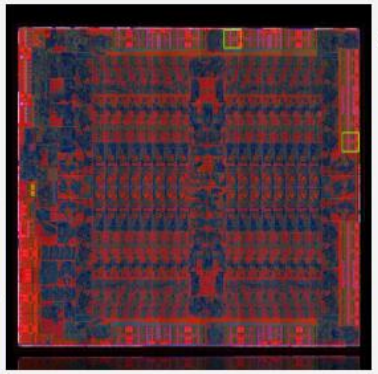
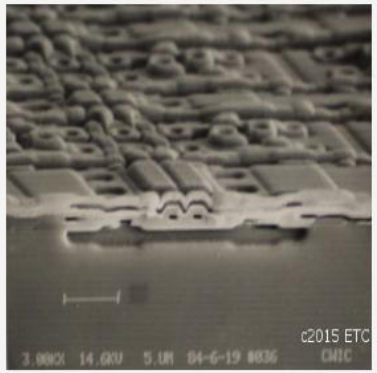
18-344: Computer Systems and the Hardware-Software Interface (Fall 2023)



Course Description

This course covers the design and implementation of computer systems from the perspective of the hardware software interface. The purpose of this course is for students to understand the relationship between the operating system, software, and computer architecture. Students that complete the course will have learned operating system fundamentals, computer architecture fundamentals, compilation to hardware abstractions, and how software actually executes from the perspective of the hardware software/boundary. The course will focus especially on understanding the relationships between software and hardware, and how those relationships influence the design of a computer system's software and hardware. The course will convey these topics through a series of practical, implementation-oriented lab assignments.

Course Staff & Logistics

- Prof. Greg Kesden
  - (412) 818-7813
  - HH A205
  - https://www.andrew.cmu.edu/~gkesden/schedule.html

- Teaching Assistants (18-344 Veterans):
  - Ashira Johara
  - Fiona Fisher
  - Jaehyun Lim
  - Yufei Shi
- Lecture: Mondays and Wednesdays, 3:30 – 4:20pm in WeH 5421
  - Some lectures are designed to run short, some to run long. We may leave early, we may spill into next week.

- Recitation: Friday, 10-10:50am in WeH 5421
  - Project focused + Reinforcement
- Office Hours (per website, TBD)
- 5 Labs (more later) + (Attendance/Homework | Exams)
- Gradescope (primarily for handin), Slack (Ephemeral questions) ,& Piazza (for continuous Q&A)
- Late policy: -10% for each day late. If you need schedule adjustments, please work it out *with us well in advance.*

Who am I?

https://www.andrew.cmu.edu/user/gkesden/

GREGORY KESDEN

Electrical and Computer Engineering (ECE)

- Associate Teaching Professor
- MS ECE Faculty Advisor

- Office: HH A205
- Voice: 412-268-8647 (Forwards to cell, 24x7)
- Fax: 877-225-2329 (Who uses FAXes these days?)
- Email: gkesden@andrew.cmu.edu
- Hangouts, Messenger, Skype, Etc: gkesden

- Virtual Office Hours: I'm not online right now. See schedule below or send email. Thanks!
- Fall 2023 Schedule: Office Hours, Class Times, Recurring Meetings, Etc.

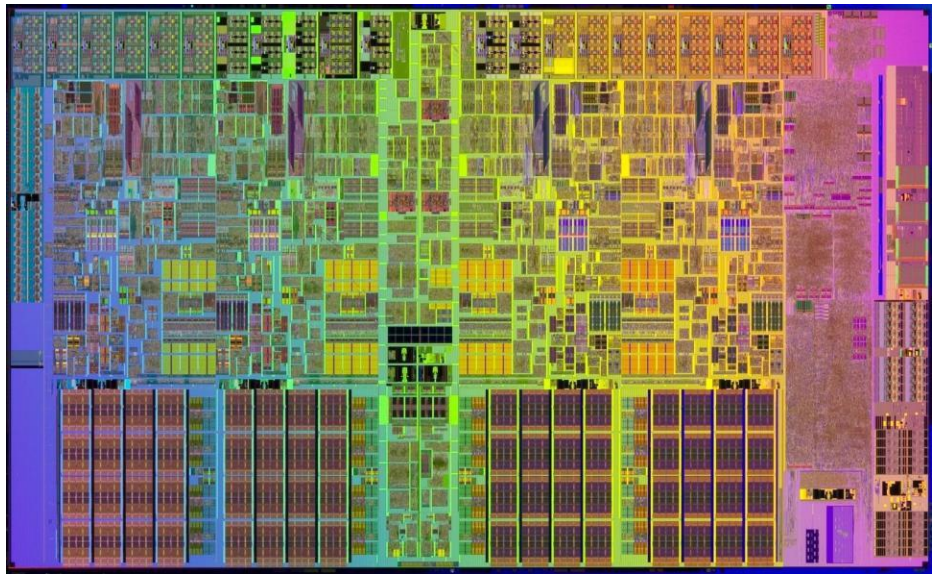
FALL 2023 CLASSES

- 18-213/613: Introduction to Computer Systems
- 18-344: Computer Systems and the Hardware-Software Interface
- 18-349: Embeddeed Systems (Canvas)

RECENT CLASSES

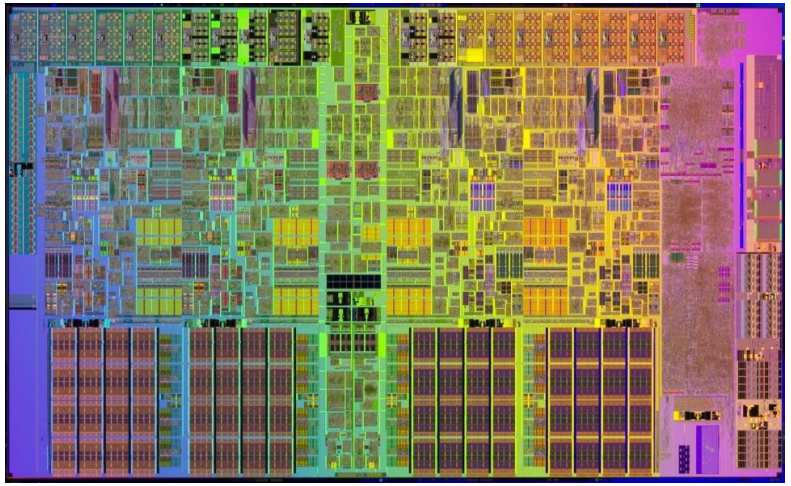
- 18-100: Introduction to ECE (Fall 2020, Canvas) (Spring 2021, Canvas) (Spring 2022, Canvas) (Fall 2022, Canvas)(Summer 2023, Canvas)
- 18-349/14-642: Introduction to Embedded Systems: (Fall 2019) (Fall 2020) (Fall 2021)
- 14-848: Cloud Infrastructure: (Fall 2017) / (Fall 2018) / (Fall 2019)
- 14-760: Adv. Real-World Data Networks (Spring 2019) / (Spring 2020)
- 14-740: Networks (Spring 2018)
- 14-736: Distributed Systems (Spring 2018) (Spring 2019) / (Spring 2020)
- 14-712: Operating Systems (Spring 2020)
- 18-537-C6: An Introduction to C and UNIX (Summer-1/Mini-5 2022)
- 18-537-D6: An Introduction to Python and ML(Summer-2/Mini-6) (Summer-2/Mini-6 2023)
- 14-513/18-613/18-600: Computer Systems: (Fall 2017) / (Fall 2018) / (Fall 2019) / (Summer 2020) / (Spring 2021) / (Fall 2021) / (Spring 2022) / (Summer-All 2022) / (Fall 2022) / (Spring 2023) / (Summer 2023)
- 15-418/618: Parallel Architecture and Programming (Spring 2018)

What is this course about?



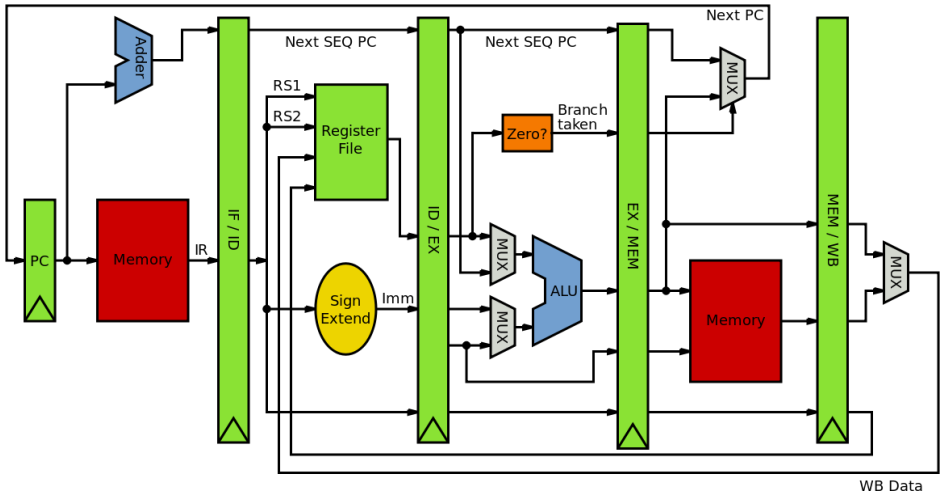
```
379  int walk_page_range(struct mm_struct *mm, unsigned long start,
380                  unsigned long end, const struct mm_walk_ops *ops,
381                  void *private)
382  {
383          int err = 0;
384          unsigned long next;
385          struct vm_area_struct *vma;
386          struct mm_walk walk = {
387                  .ops            = ops,
388                  .mm             = mm,
389                  .private        = private,
390          };
391
392          if (start >= end)
393                  return -EINVAL;
394
395          if (!walk.mm)
396                  return -EINVAL;
397
398          mmap_assert_locked(walk.mm);
399
400          vma = find_vma(walk.mm, start);
401          do {
402                  if (!vma) { /* after the last vma */
403                          walk.vma = NULL;
404                          next = end;
```

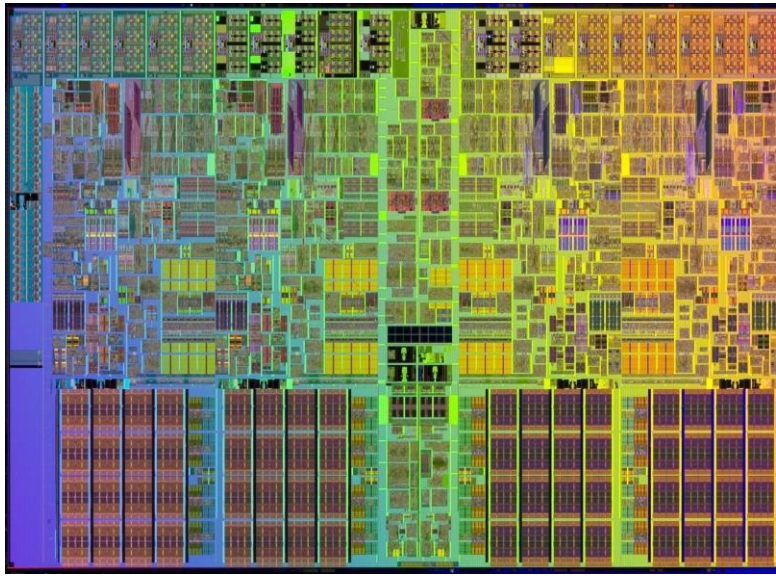
What is the hardware software boundary?



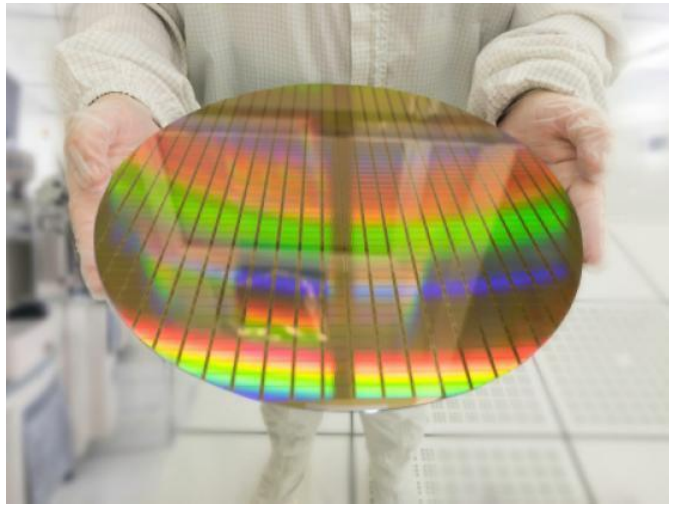
What is computer (μ)architecture?

Inst	Name	Opcode	funct3	funct7	Description (C)
add	ADD	0110011	0x0	0x00	`rd = rs1 + rs2`
sub	SUB	0110011	0x0	0x20	`rd = rs1 - rs2`
xor	XOR	0110011	0x4	0x00	`rd = rs1 ^ rs2`
or	OR	0110011	0x6	0x00	`rd = rs1 | rs2`
and	AND	0110011	0x7	0x00	`rd = rs1 & rs2`

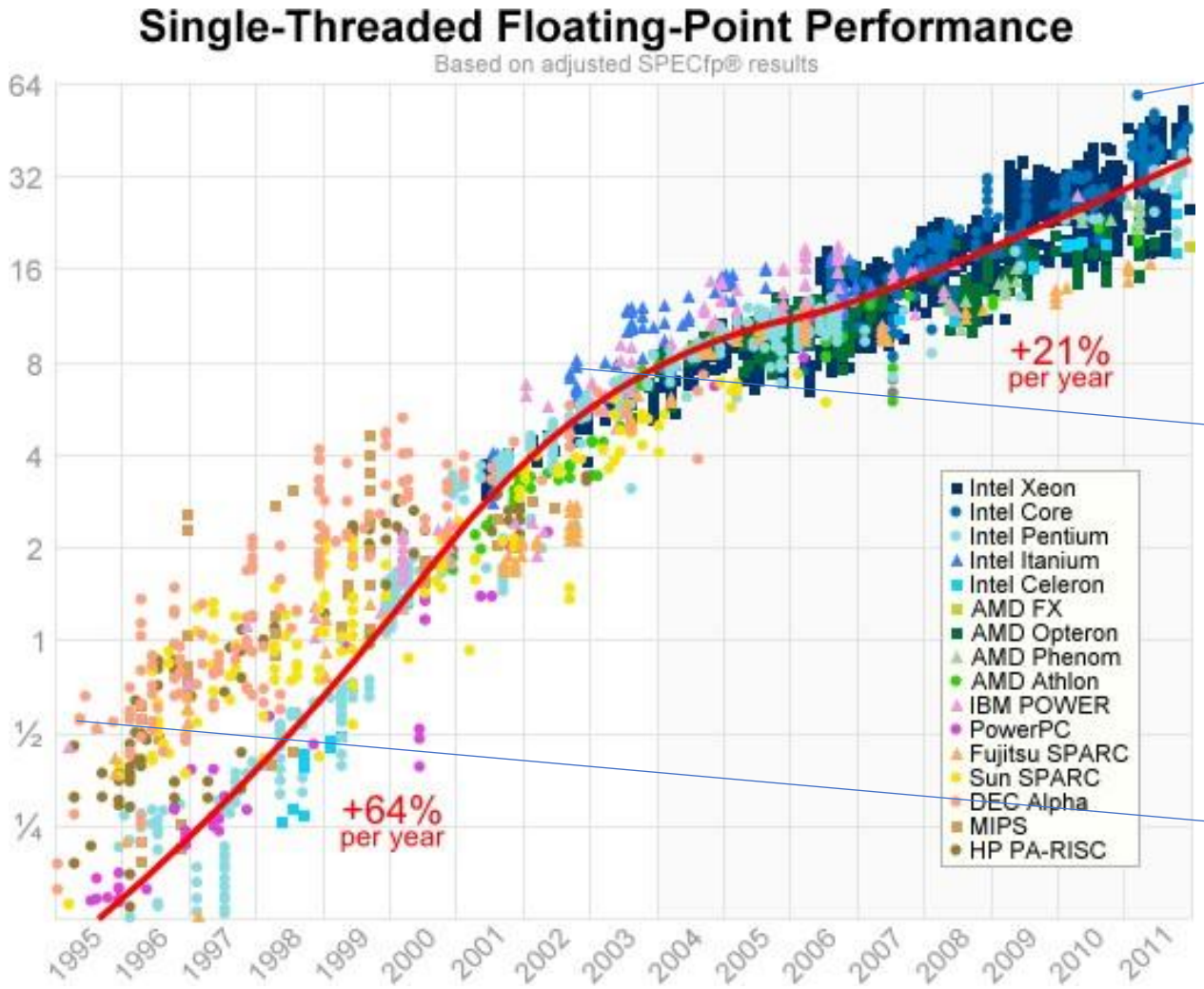


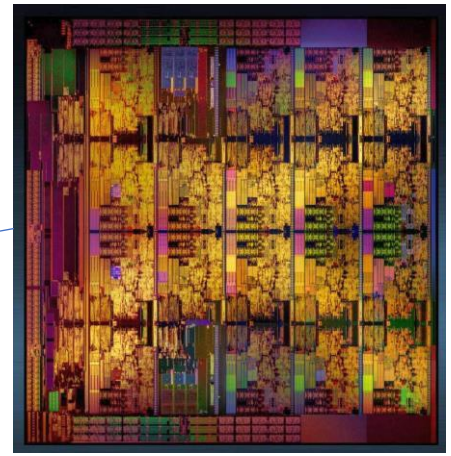


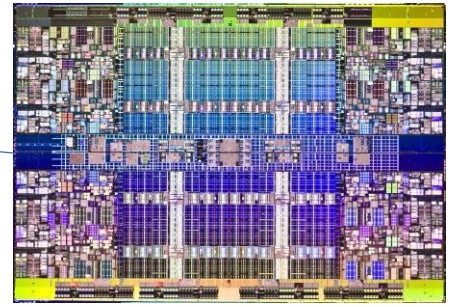
What constrains a computer system?

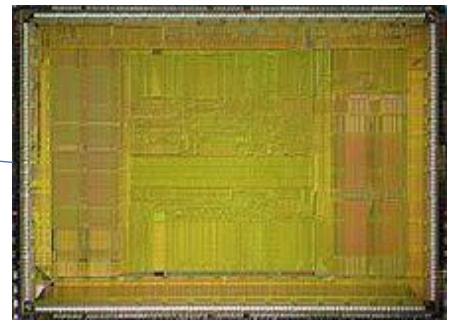


Why are these processors different?

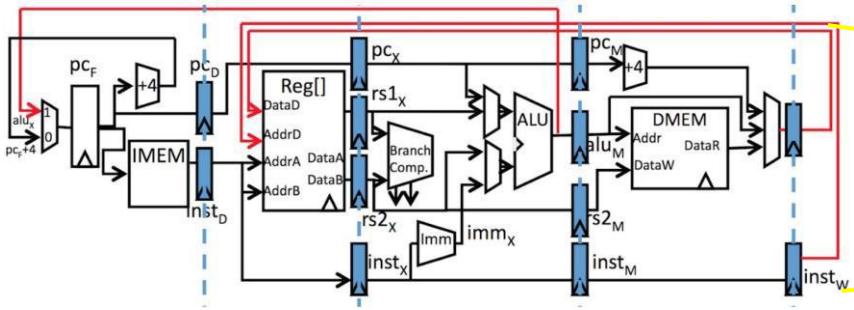


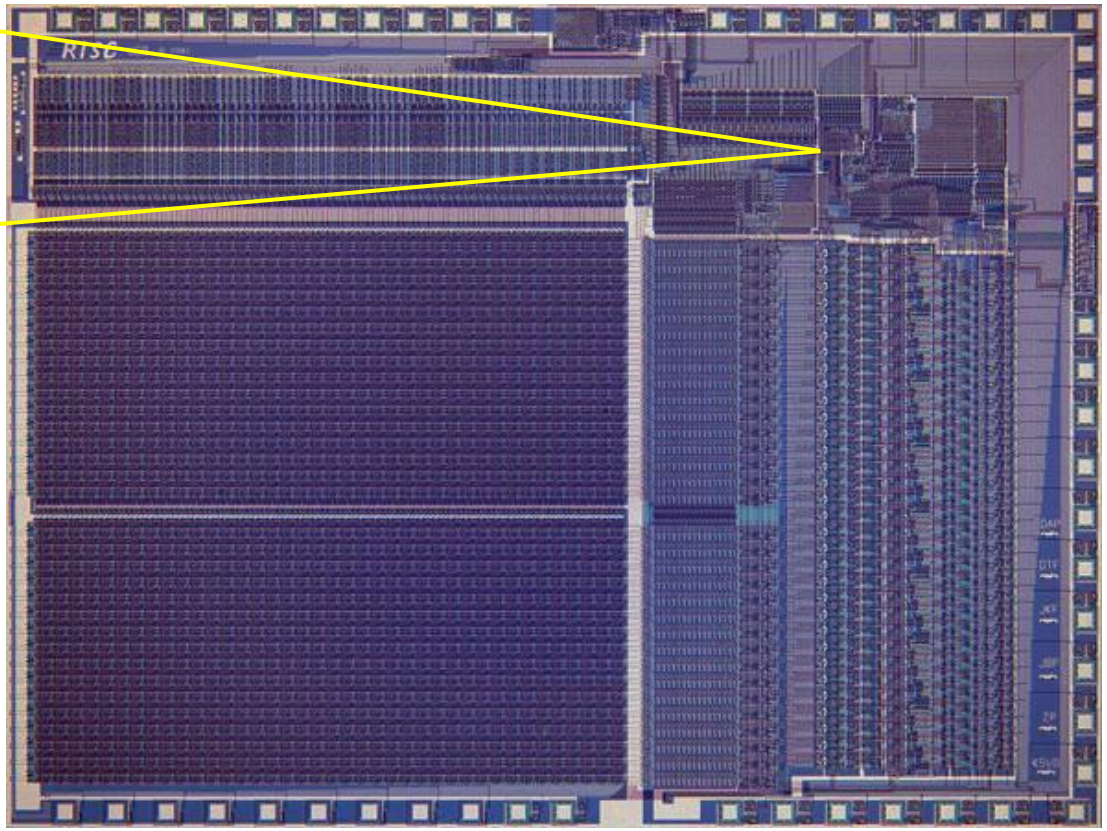


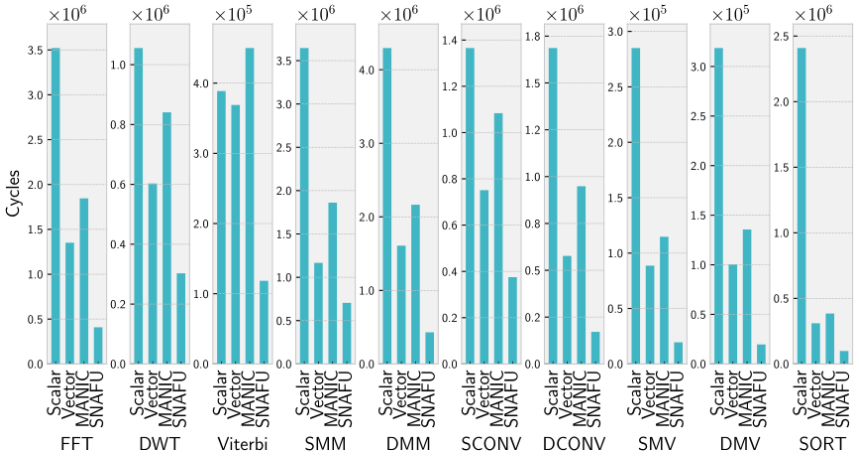




How do you measure a computer's performance?

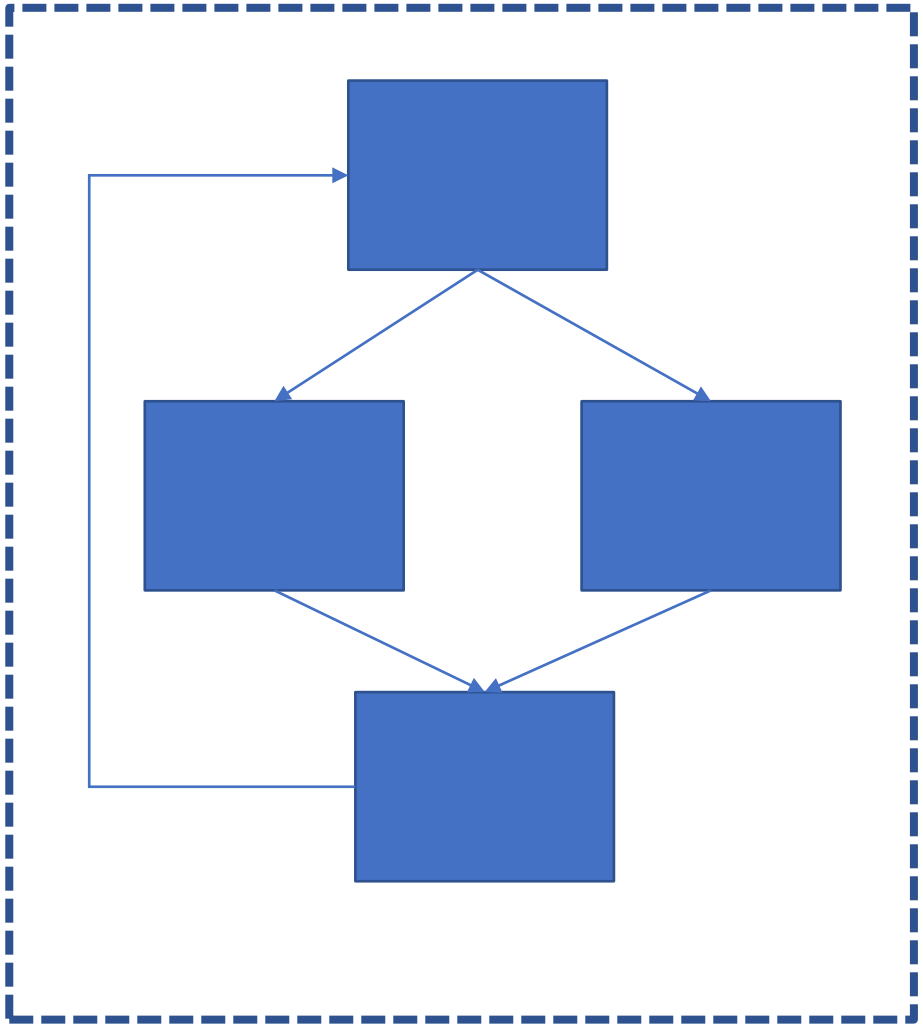






What makes software runnable?

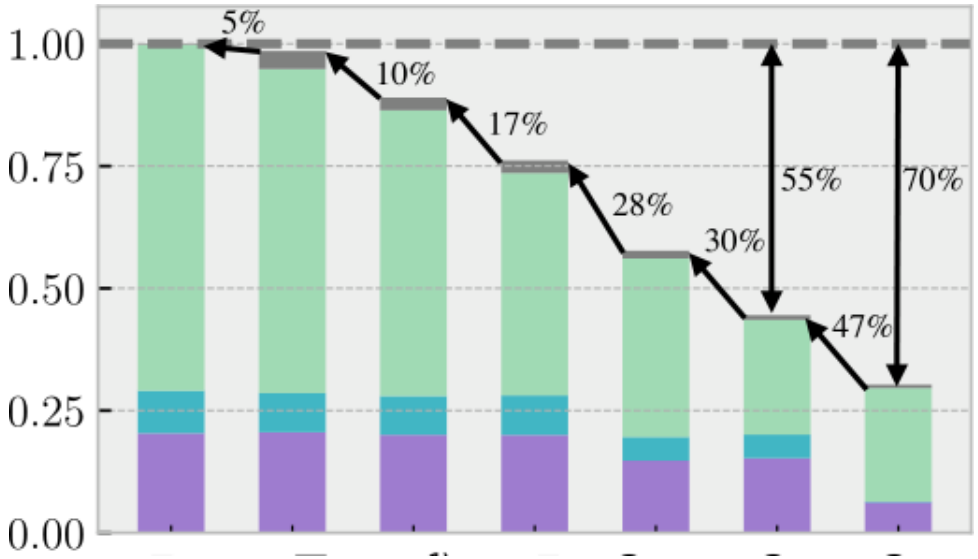
```
379  int walk_page_range(struct mm_struct *mm, unsigned long start,
380                  unsigned long end, const struct mm_walk_ops *ops,
381                  void *private)
382  {
383          int err = 0;
384          unsigned long next;
385          struct vm_area_struct *vma;
386          struct mm_walk walk = {
387                  .ops            = ops,
388                  .mm             = mm,
389                  .private        = private,
390          };
391
392          if (start >= end)
393                  return -EINVAL;
394
395          if (!walk.mm)
396                  return -EINVAL;
397
398          mmap_assert_locked(walk.mm);
399
400          vma = find_vma(walk.mm, start);
401          do {
402                  if (!vma) { /* after the last vma */
403                          walk.vma = NULL;
404                          next = end;
```

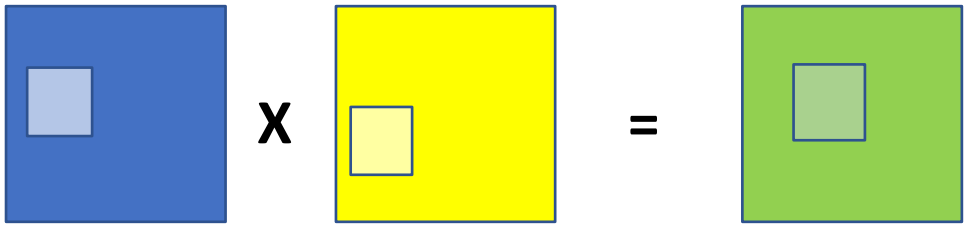


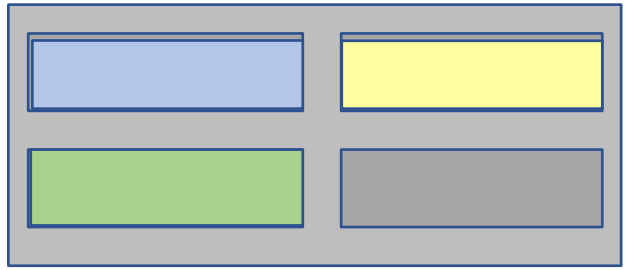
```
mov     0x18(%rdi),%r14
sub     0x10(%rdi),%r14
mov     0x8(%rdi),%rbp
add     %r14,%rsi
mov     %r14,%rdx
setb    %al
add     0x30(%rdi),%rsi
setb    %cl
shr     $0x3,%rdx
cmp     %rsi,(%rdi)
lea     0x64(%rsi,%rdx,1),%r12
cmovae  (%rdi),%rsi
cmp     %r12,%rsi
cmovae  %rsi,%r12
test    %rax,%rax
```

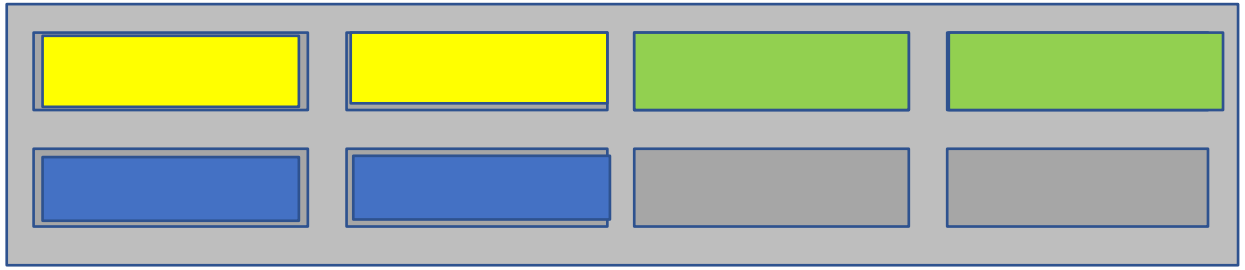
```
16aea:      48 c1 ea 03
16aee:      48 39 37
16af1:      4c 8d 64 16 64
16af6:      48 0f 43 37
16afa:      4c 39 e6
16afd:      4c 0f 43 e6
16b01:      48 85 c0
```

How do you *improve* software's performance?

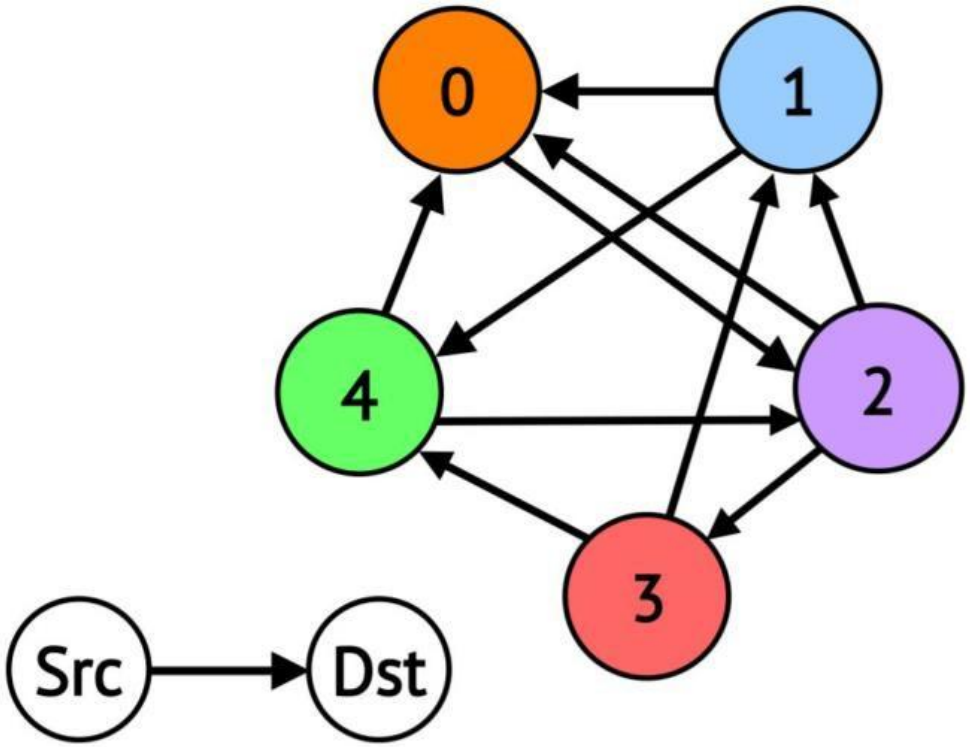


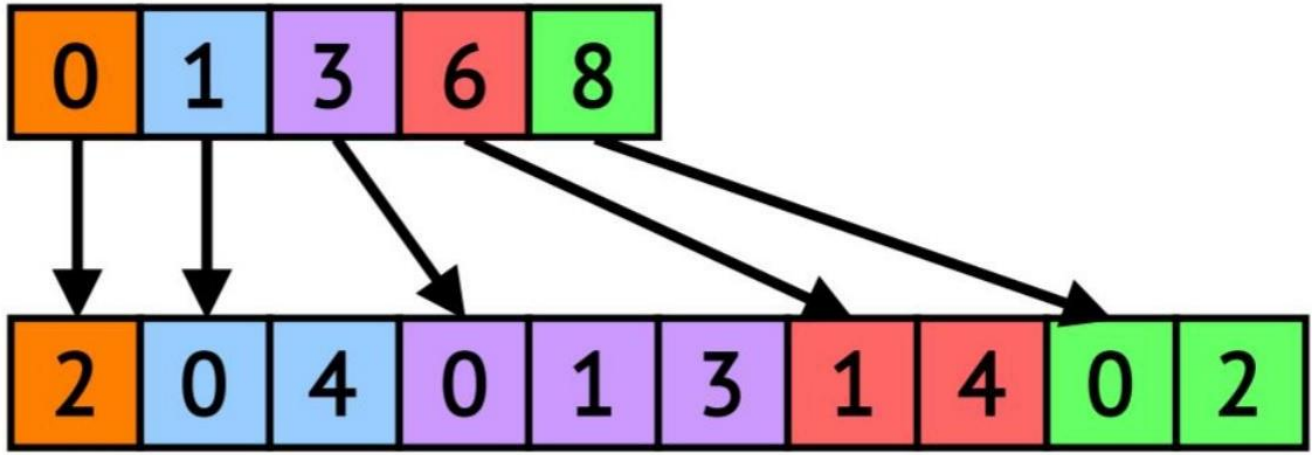




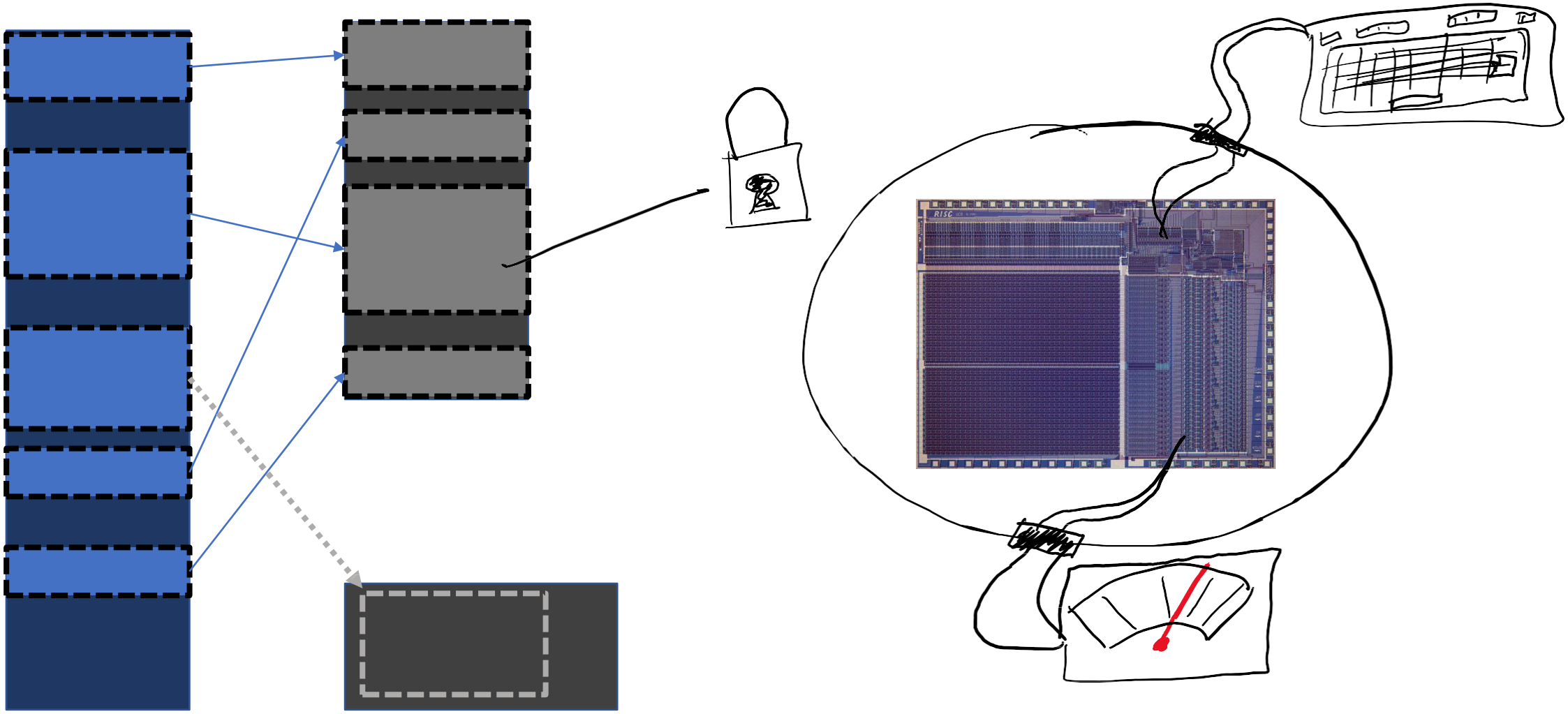


Are some programs intrinsically slow?

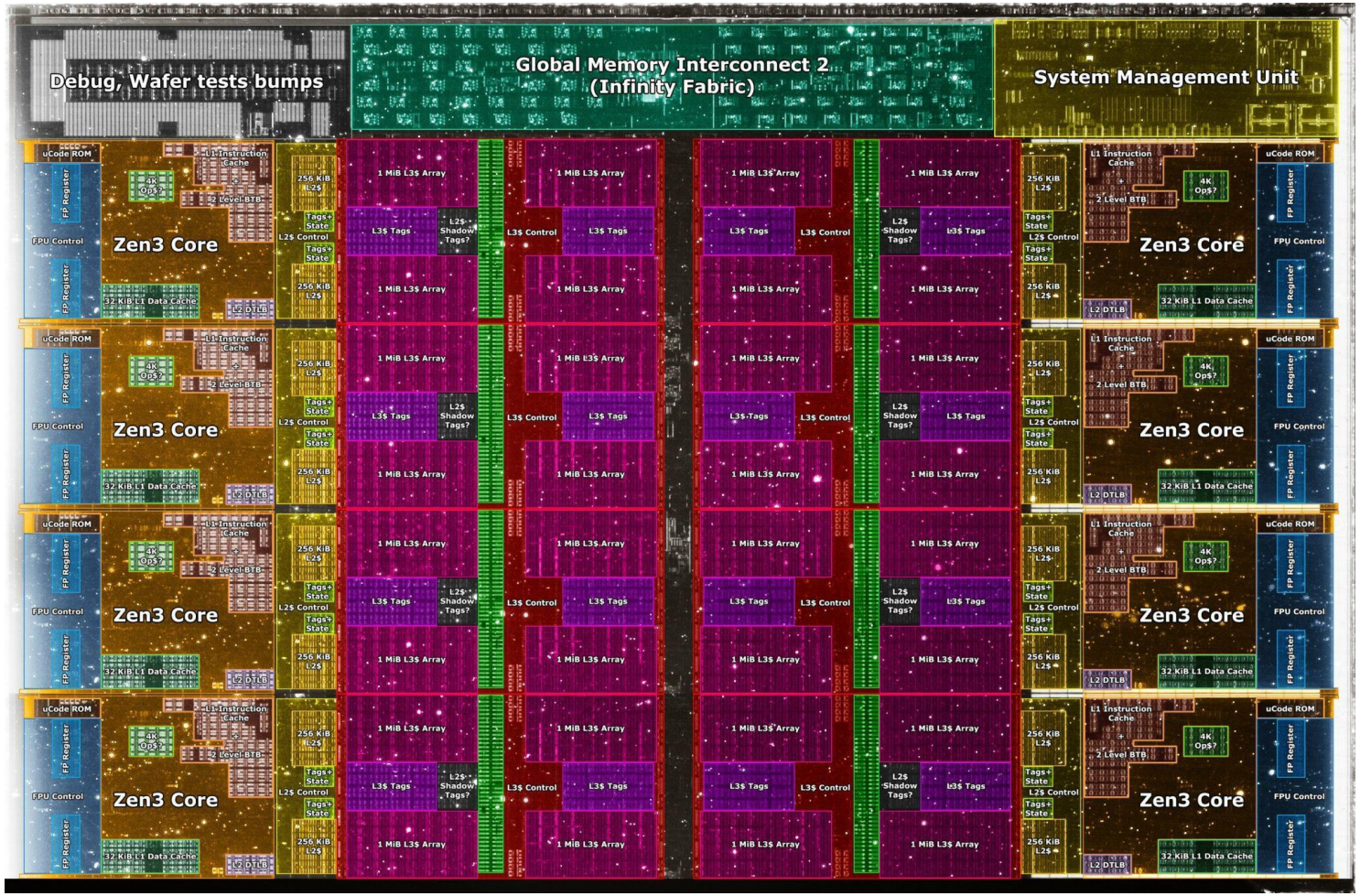




What are the lower-extremities of software?

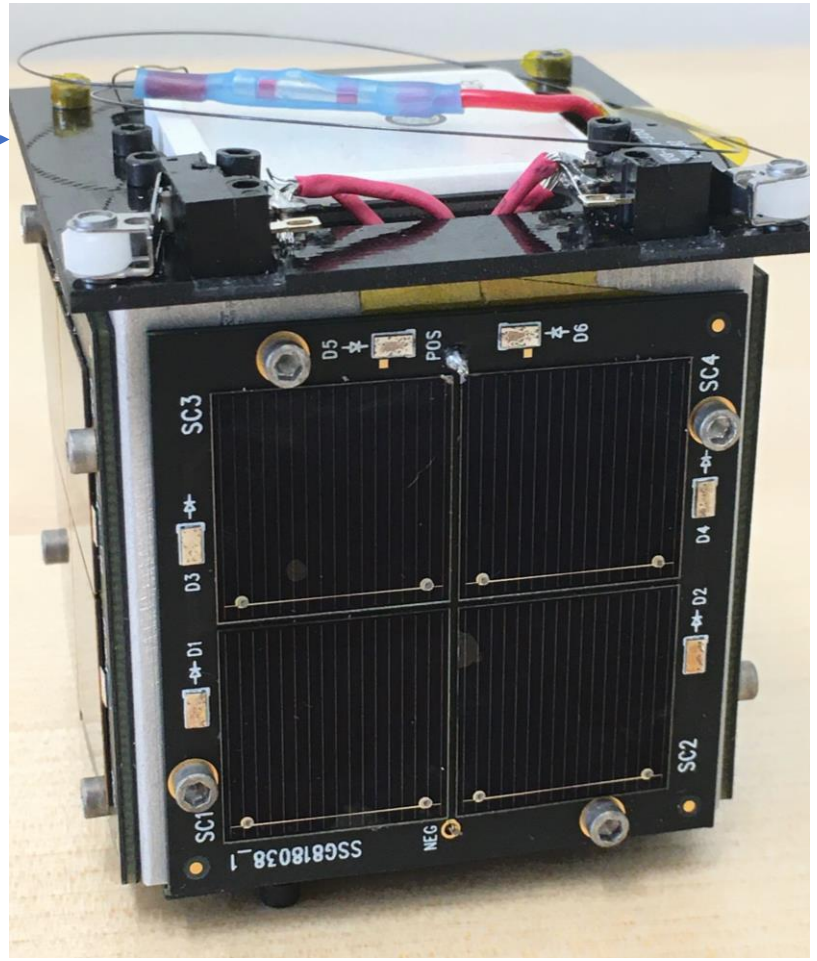


How (& why?) to do two things at the same time?



How does it all fit together in real systems?

Tartan-Artibeus-1 Batteryless Energy-harvesting Nanosatellite System



What are we doing this semester?

- Series of labs (Lab 0 – "bootstrapping" goes out today, due in 1 week)
  - Do all labs except lab 0 in pairs. The point is not to make you code or experiment for the rest of your life. **We want you to learn by doing**.
- Labs released via AFS: /afs/ece.cmu.edu/class/ece344/assign
  - For each lab, *handout.txt* is your main reference & documentation. **READ IT CAREFULLY** because it will tell you what you need to know to complete the lab
- Labs are **not** just "code & submit". Instead, you'll be building, studying, measuring, and evaluating systems. Coding it up is step 1.

Course Calendar

Date	Topic
08/25/2025	Introduction to the Hardware/Software Boundary [pptx] / [pdf]
08/27/2025	von Neumann Architectures [pptx] / [pdf]
09/01/2025	**No class - Labor Day**
09/03/2025	Computer Architecture Basics [pptx] / [pdf]
09/08/2025	ISAs: The RISC-V ISA [pptx] / [pdf]
09/10/2025	Pipelines and Hazards [pptx] / [pdf]
09/15/2025	Control hazards and Branch Prediction [pptx] / [pdf]
09/17/2025	Caches and Memory Hierarchy [pptx] / [pdf]
09/22/2025	Cache Replacement Policies and Enhancements [pptx] / [pdf]
09/24/2025	Introduction to Performance Evaluation [pptx] / [pdf]
09/29/2025	Design Space Exploration [pptx] / [pdf]
10/01/2025	Advanced Architecture: Superscalar and Out of Order [pptx] / [pdf]
10/06/2025	Advanced Architecture: Superscalar and Out of Order (Part 2)) [pptx] / [pdf]
10/08/2025	Exam #1
10/13/2025	**Fall Break**
10/15/2025	**Fall Break**

Date	Topic
10/20/2025	Advanced Dataflow (Energy Minimal) [pptx] / [pdf]
10/22/2025	Virtual Memory [pptx] / [pdf]
10/27/2025	The Compiler Is Here to Help (And, wrapping up VM) [pptx] / [pdf]
10/29/2025	Sparse Problems Introduction [pptx] / [pdf]
11/03/2025	Sparse Problems Optimization (Propagation Blocking) [pptx] / [pdf]
11/05/2025	Introduction to Data Center Computing [pptx] / [pptx]
11/10/2025	Parallelism, Coherency, and Concurrency Basics [pptx] / [pdf]
11/12/2025	Synchronization and Transactional Memory [pptx] / [pdf]
11/17/2025	Consistency, Coherency, and Understanding the Model [pptx] / [pdf]
11/19/2025	Multicore Interconnect Networks [pptx] / [pdf]
11/24/2025	Meltdown and Spectre [pptx] / [pdf]
11/26/2025	**No Class -- Thanksgiving**
12/01/2025	Review/Wrap-Up
12/03/2025	Exam #2

Diving into Computer Architecture

Date	Topic
08/25/2025	Introduction to the Hardware/Software Boundary [pptx] / [pdf]
08/27/2025	von Neumann Architectures [pptx] / [pdf]
09/01/2025	**No class - Labor Day**
09/03/2025	Computer Architecture Basics [pptx] / [pdf]
09/08/2025	ISAs: The RISC-V ISA [pptx] / [pdf]
09/10/2025	Pipelines and Hazards [pptx] / [pdf]
09/15/2025	Control hazards and Branch Prediction [pptx] / [pdf]
09/17/2025	Caches and Memory Hierarchy [pptx] / [pdf]
09/22/2025	Cache Replacement Policies and Enhancements [pptx] / [pdf]
09/24/2025	Introduction to Performance Evaluation [pptx] / [pdf]
09/29/2025	Design Space Exploration [pptx] / [pdf]
10/01/2025	Advanced Architecture: Superscalar and Out of Order [pptx] / [pdf]
10/06/2025	Advanced Architecture: Superscalar and Out of Order (Part 2)) [pptx] / [pdf]
10/08/2025	Exam #1
10/13/2025	**Fall Break**
10/15/2025	**Fall Break**

Broad introduction to computer architecture, understanding the architecture / microarchitecture distinction, what is an ISA?, what limits computer performance?, how to think about hardware using abstractions, Amdahl's Law

ILP & Dealing with Hazards

Date	Topic
08/25/2025	Introduction to the Hardware/Software Boundary [pptx] / [pdf]
08/27/2025	von Neumann Architectures [pptx] / [pdf]
09/01/2025	**No class - Labor Day**
09/03/2025	Computer Architecture Basics [pptx] / [pdf]
09/08/2025	ISAs: The RISC-V ISA [pptx] / [pdf]
09/10/2025	Pipelines and Hazards [pptx] / [pdf]
09/15/2025	Control hazards and Branch Prediction [pptx] / [pdf]
09/17/2025	Caches and Memory Hierarchy [pptx] / [pdf]
09/22/2025	Cache Replacement Policies and Enhancements [pptx] / [pdf]
09/24/2025	Introduction to Performance Evaluation [pptx] / [pdf]
09/29/2025	Design Space Exploration [pptx] / [pdf]
10/01/2025	Advanced Architecture: Superscalar and Out of Order [pptx] / [pdf]
10/06/2025	Advanced Architecture: Superscalar and Out of Order (Part 2)) [pptx] / [pdf]
10/08/2025	Exam #1
10/13/2025	**Fall Break**
10/15/2025	**Fall Break**

Introduction to ILP, pipelining and what gets in its way, how control flow happens at execution time, branch prediction

Caches & Memory Hierarchies

Date	Topic
08/25/2025	Introduction to the Hardware/Software Boundary [pptx] / [pdf]
08/27/2025	von Neumann Architectures [pptx] / [pdf]
09/01/2025	**No class - Labor Day**
09/03/2025	Computer Architecture Basics [pptx] / [pdf]
09/08/2025	ISAs: The RISC-V ISA [pptx] / [pdf]
09/10/2025	Pipelines and Hazards [pptx] / [pdf]
09/15/2025	Control hazards and Branch Prediction [pptx] / [pdf]
09/17/2025	Caches and Memory Hierarchy [pptx] / [pdf]
09/22/2025	Cache Replacement Policies and Enhancements [pptx] / [pdf]
09/24/2025	Introduction to Performance Evaluation [pptx] / [pdf]
09/29/2025	Design Space Exploration [pptx] / [pdf]
10/01/2025	Advanced Architecture: Superscalar and Out of Order [pptx] / [pdf]
10/06/2025	Advanced Architecture: Superscalar and Out of Order (Part 2)) [pptx] / [pdf]
10/08/2025	Exam #1
10/13/2025	**Fall Break**
10/15/2025	**Fall Break**

Memory is the real problem! Caches, memory hierarchies: an architect's view, cache replacement and other cache optimizations

Principled Performance Analysis

Date	Topic
08/25/2025	Introduction to the Hardware/Software Boundary [pptx] / [pdf]
08/27/2025	von Neumann Architectures [pptx] / [pdf]
09/01/2025	**No class - Labor Day**
09/03/2025	Computer Architecture Basics [pptx] / [pdf]
09/08/2025	ISAs: The RISC-V ISA [pptx] / [pdf]
09/10/2025	Pipelines and Hazards [pptx] / [pdf]
09/15/2025	Control hazards and Branch Prediction [pptx] / [pdf]
09/17/2025	Caches and Memory Hierarchy [pptx] / [pdf]
09/22/2025	Cache Replacement Policies and Enhancements [pptx] / [pdf]
09/24/2025	Introduction to Performance Evaluation [pptx] / [pdf]
09/29/2025	Design Space Exploration [pptx] / [pdf]
10/01/2025	Advanced Architecture: Superscalar and Out of Order [pptx] / [pdf]
10/06/2025	Advanced Architecture: Superscalar and Out of Order (Part 2)) [pptx] / [pdf]
10/08/2025	Exam #1
10/13/2025	**Fall Break**
10/15/2025	**Fall Break**

Measuring a system in a meaningful way, understanding performance measurement pitfalls, Pareto analysis, design space iteration & exploration, Amdahl's Law (again)

Microarchitectural Optimizations

Date	Topic
08/25/2025	Introduction to the Hardware/Software Boundary [pptx] / [pdf]
08/27/2025	von Neumann Architectures [pptx] / [pdf]
09/01/2025	**No class - Labor Day**
09/03/2025	Computer Architecture Basics [pptx] / [pdf]
09/08/2025	ISAs: The RISC-V ISA [pptx] / [pdf]
09/10/2025	Pipelines and Hazards [pptx] / [pdf]
09/15/2025	Control hazards and Branch Prediction [pptx] / [pdf]
09/17/2025	Caches and Memory Hierarchy [pptx] / [pdf]
09/22/2025	Cache Replacement Policies and Enhancements [pptx] / [pdf]
09/24/2025	Introduction to Performance Evaluation [pptx] / [pdf]
09/29/2025	Design Space Exploration [pptx] / [pdf]
10/01/2025	Advanced Architecture: Superscalar and Out of Order [pptx] / [pdf]
10/06/2025	Advanced Architecture: Superscalar and Out of Order (Part 2)) [pptx] / [pdf]
10/08/2025	Exam #1
10/13/2025	**Fall Break**
10/15/2025	**Fall Break**

Going beyond IPC=1, advanced ILP techniques, superscalar & out-of-order execution, vector processors, Very Large Instruction Word processors, other more exotic architectures

Midterm Exam

Date	Topic
08/25/2025	Introduction to the Hardware/Software Boundary [pptx] / [pdf]
08/27/2025	von Neumann Architectures [pptx] / [pdf]
09/01/2025	**No class - Labor Day**
09/03/2025	Computer Architecture Basics [pptx] / [pdf]
09/08/2025	ISAs: The RISC-V ISA [pptx] / [pdf]
09/10/2025	Pipelines and Hazards [pptx] / [pdf]
09/15/2025	Control hazards and Branch Prediction [pptx] / [pdf]
09/17/2025	Caches and Memory Hierarchy [pptx] / [pdf]
09/22/2025	Cache Replacement Policies and Enhancements [pptx] / [pdf]
09/24/2025	Introduction to Performance Evaluation [pptx] / [pdf]
09/29/2025	Design Space Exploration [pptx] / [pdf]
10/01/2025	Advanced Architecture: Superscalar and Out of Order [pptx] / [pdf]
[illegible]	[illegible]
10/08/2025	Exam #1
[illegible]	**Fall Break**
10/15/2025	**Fall Break**

The midterm covers everything from the start of the course up to this point. The labs & homeworks will evaluate what you learn during the second half of the course.

Virtual Memory

10/20/2025	Advanced Dataflow (Energy Minimal) [pptx] / [pdf]
10/2[illegible]/2025	Virtual Memory [pptx] / [pdf]
10/2[illegible]/2025	The Compiler Is Here to Help (And, wrapping up VM) [pptx] / [pdf]
10/29/2025	Sparse Problems Introduction [pptx] / [pdf]
11/03/2025	Sparse Problems Optimization (Propagation Blocking) [pptx] / [pdf]
11/05/2025	Introduction to Data Center Computing [pptx] / [pptx]
11/10/2025	Parallelism, Coherency, and Concurrency Basics [pptx] / [pdf]
11/12/2025	Synchronization and Transactional Memory [pptx] / [pdf]
11/17/2025	Consistency, Coherency, and Understanding the Model [pptx] / [pdf]
11/19/2025	Multicore Interconnect Networks [pptx] / [pdf]
11/24/2025	Meltdown and Spectre [pptx] / [pdf]
11/26/2025	**No Class -- Thanksgiving**
12/01/2025	Review/Wrap-Up
12/03/2025	Exam #2

Virtual Memory, virtualization basics, bad ways to do VM, hardware/software co-design for virtualization, VM advanced topics, huge pages, TLB design

Sparse Computation

10/20/2025	Advanced Dataflow (Energy Minimal) [pptx] / [pdf]
10/22/2025	Virtual Memory [pptx] / [pdf]
10/27/2025	The Compiler Is Here to Help (And, wrapping up VM) [pptx] / [pdf]
10/29/2025	Sparse Problems Introduction [pptx] / [pdf]
11/03/2025	Sparse Problems Optimization (Propagation Blocking) [pptx] / [pdf]
11/05/2025	Introduction to Data Center Computing [pptx] / [pptx]
11/10/2025	Parallelism, Coherency, and Concurrency Basics [pptx] / [pdf]
11/12/2025	Synchronization and Transactional Memory [pptx] / [pdf]
11/17/2025	Consistency, Coherency, and Understanding the Model [pptx] / [pdf]
11/19/2025	Multicore Interconnect Networks [pptx] / [pdf]
11/24/2025	Meltdown and Spectre [pptx] / [pdf]
11/26/2025	**No Class -- Thanksgiving**
12/01/2025	Review/Wrap-Up
12/03/2025	Exam #2

Introduction to sparsity, what is a sparse problem and why is one difficult?, understanding the performance limiters, sparse problem optimization strategies, open problems

Parallelism & Concurrency

0/20/2025	Advanced Dataflow (Energy Minimal) [pptx] / [pdf]
0/22/2025	Virtual Memory [pptx] / [pdf]
0/27/2025	The Compiler Is Here to Help (And, wrapping up VM) [pptx] / [pdf]
0/29/2025	Sparse Problems Introduction [pptx] / [pdf]
1/03/2025	Sparse Problems Optimization (Propagation Blocking) [pptx] / [pdf]
1/05/2025	Introduction to Data Center Computing [pptx] / [pptx]
1/10/2025	Parallelism, Coherency, and Concurrency Basics [pptx] / [pdf]
1/12/2025	Synchronization and Transactional Memory [pptx] / [pdf]
1/17/2025	Consistency, Coherency, and Understanding the Model [pptx] / [pdf]
1/19/2025	Multicore Interconnect Networks [pptx] / [pdf]
1/24/2025	Meltdown and Spectre [pptx] / [pdf]
1/26/2025	**No Class -- Thanksgiving**
2/01/2025	Review/Wrap-Up
2/03/2025	Exam #2

Parallel computation, parallel architectures concurrency, memory consistency models, synchronization, atomics, transactional memory networks on chip

End of semester: computer architecture today

/20/2025	Advanced Dataflow (Energy Minimal) [pptx] / [pdf]
/22/2025	Virtual Memory [pptx] / [pdf]
/27/2025	The Compiler Is Here to Help (And, wrapping up VM) [pptx] / [pdf]
/29/2025	Sparse Problems Introduction [pptx] / [pdf]
/03/2025	Sparse Problems Optimization (Propagation Blocking) [pptx] / [pdf]
/05/2025	Introduction to Data Center Computing [pptx] / [pptx]
/10/2025	Parallelism, Coherency, and Concurrency Basics [pptx] / [pdf]
/12/2025	Synchronization and Transactional Memory [pptx] / [pdf]
/17/2025	Consistency, Coherency, and Understanding the Model [pptx] / [pdf]
/19/2025	Multicore Interconnect Networks [pptx] / [pdf]
/24/2025	Meltdown and Spectre [pptx] / [pdf]
/26/2025	**No Class -- Thanksgiving**
/01/2025	Review/Wrap-Up
/03/2025	Exam #2

(Subject to change) What is happening in the field of computer architecture and systems today? What are the exciting new ideas? Reconfigurable dataflow machines, encrypted computing, energy-harvesting computers, open-ended Q/A

Lab 0: Bootstrapping

- 18-344 has lots of moving parts. Lab 0 is about figuring them all out
  - You'll be using each of them again in subsequent labs.
- SPEC2017: collection of benchmark programs designed for evaluating computer architectures
  - Needlessly complex and very difficult to change infrastructure. SPEC will be a pain, but will give you the Real Computer Systems Experience.
- Pin: binary instrumentation tool used to insert code into program binaries to implement computer architecture simulators
- Destiny: memory modeling tool useful for evaluating the time & energy to access different cache/memory designs

Lab 1: Branch Prediction

- As you will learn, (and as you may recall from 213) microarchitectures predict the outcome of their branch instructions
- Write a branch predictor simulator
- Evaluate different implementations
  - Cost, accuracy, implementation feasibility
- Write-up in English prose characterizing and explaining your design, and with quantitative evaluation of your design's performance.

Lab 2: Memory Hierarchy Design Space

- Cache hierarchies are big complex microarchitectural components
- Which is best for a given set of programs?
- You will start by writing a cache, and then a whole memory hierarchy.
- Then you will run a design space exploration process to optimize your memory hierarchy implementation, subject to physical constraints and performance & efficiency goals
- You will explore different replacement policies
- Write-up in English prose summarizing your design space exploration and conclusions, including quantitative evaluation.

Lab 3: Virtual Memory

- Virtual memory provides process isolation and access control using paging and some hardware support
- You will implement an emulation of a page table and the basic functions that you will use to manipulate the page table
- You will implement simulated hardware to accelerate translate of virtual to physical addresses and evaluate its impact on system performance
- You will study your implementation and quantitatively analyze your page table and hardware support
- Write-up describing your design, including quantitative evaluation of your system and its performance

Lab 4: Sparse Workload Optimization

- Sparse workloads are programs that work on datasets that have sparse structure, like graphs that have lots of vertices and far fewer edges (sparse adjacency matrix). Sparse workloads are hard to cache.
- Given a simple unoptimized sparse workload implementation, you will use a highly specialized optimization called Propagation Blocking to optimize, yielding a much higher performance implementation
- You will study your optimized version, and quantitatively analyze your design choices.
- Write-up including description of your implementation and summary of quantitative results

Lab 5: Synchronization for Parallel Code

- Parallel programs require synchronizing to avoid confusing interactions between threads.
- There are many ways to implement synchronization
- You will implement different synchronization mechanisms (from spin locks to transactional memory) for two performance-sensitive test programs
- You will quantitatively study your implementations and their performance on the two test programs
- Write-up including description of your synchronization alternatives and a quantitative analysis of performance