CS61C: Great Ideas in Computer Architecture (a.k.a. Machine Structures)

Dr. Nick Weaver Lecturer (He/Him)





Computer Science 61C Spring 2022

- My primary specialty is Network Security and Network Measurement
 - Although a reformed hardware person, originally specializing in FPGAs, doing some of my own circuit board design these days...
 - After all, I hate drones, so...
- I will sprinkle a fair bit of security stuff throughout the lectures
 - Security is not an afterthought, but needs to be engineered in from the start: Since this class covers everything from the transistor to the cloud, I'll make security notes along the way
- What does lecturer mean?



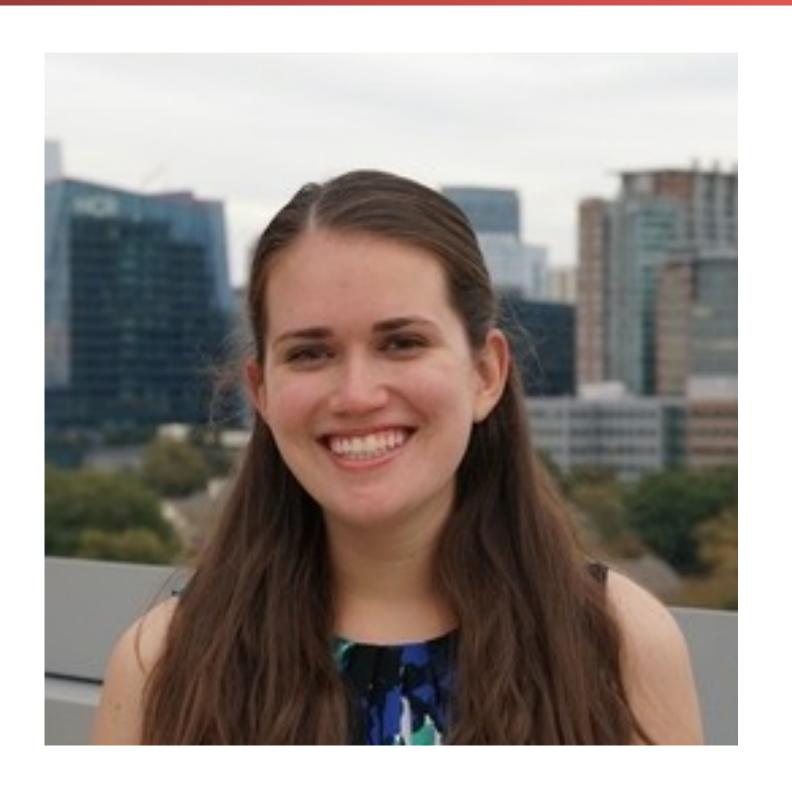




Connor McMahon (she/her)

Computer Science 61C Spring 2022

- Graduate Student in EECS
- Undergrad at Georgia Tech
- Former 61C TA





The Head TAs

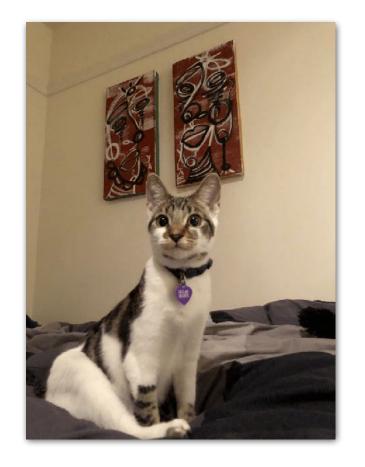
Computer Science 61C Spring 2022

McMahon and Weaver



Justin Yokota jyokota@

Hi everyone! I'm Justin, a student in the Fifth-Year Master's program for CS, who graduated as a Math/CS double major last semester. This semester, I'll be the head TA primarily responsible for content (ex. homeworks, projects, exams). Outside of classes, I tend to solve puzzles and play board games. Good luck!



Zoe Plaxco zapplaxco@

Hi! I love teaching and I bake a lot of bread! Come by my office hours to talk about plants, cooking, or existing in a late stage capitalist system. I am deeply passionate about breakfast foods. I am non-binary, and my pronouns are they/them. I am neurodivergent and queer



Jerry Xu jerryxu@

Howdy! I'm Jerry, EECS senior. I handle some of the infrastructure/software bits and bobs around here.



The TA Corps:

Computer Science 61C Spring 2022

- Too many to list on slides:
 - See https://cs61c.org/sp22/staff/



Agenda

Computer Science 61C Spring 2022

- What you need to know about this class
- Thinking about Machine Structures
- Great Ideas in Computer Architecture
- Number Representation



Course Information: https://cs61c.org/

Computer Science 61C Spring 2022 McMahon and Weaver



CS 61C Spring 2022 Calendar Staff Policies Resources OH Queue Venus Semesters

Great Ideas in Computer Architecture (Machine Structures)

CS 61C at UC Berkeley with Connor McMahon, Nicholas Weaver - Spring 2022

Lecture: Tuesday/Thursday 11:00AM - 12:30PM PT, Online

Week	Date	Lecture	Reading	Lab & Discussion	HW & Project
1	Tue 1/18	Lecture 1: Intro, Number Representation Slides Video	Course Policies		
	Thu 1/20	Lastura O. O Intra. Basica	K&R Ch. 1-5 C	Discussion 1: Number Rep Lab 0: Intro and Setup Due 1/24	
		Slides Video	Reference Slides Brian Harvey's Intro to C		
	Fri 1/21				Homework 1: Number Rep Due 1/28
2	Mon 1/24				Project 1: TBD Due 2/09
	Tue 1/25	Lecture 3: C Intro - Pointers, Arrays, Strings	K&R 5-6	Discussion 2: C Basics	
	Thu 1/27	Lecture 4: C Memory (Mis)Management	K&R 7.8.5, 8.7	Lab 1: C & CGDB Due 1/31	
	Fri 1/28				Homework 2: C Concepts Due 2/04



Course Information

Computer Science 61C Spring 2022

- Course Web: https://cs61c.org/
- Major tutoring support from CS-Mentors & paid tutors
- Textbooks: Average 15 pages of reading/week (can rent!)
 - highly recommended but not strictly required
 - Patterson & Hennessey, Computer Organization and Design, 5/e (RISC-V)
 - Kernighan & Ritchie, The C Programming Language, 2nd Edition
 - "ANSI" (old-school) C...
 - Barroso & Holzle, The Datacenter as a Computer, 2nd Edition (Online so don't need to buy)

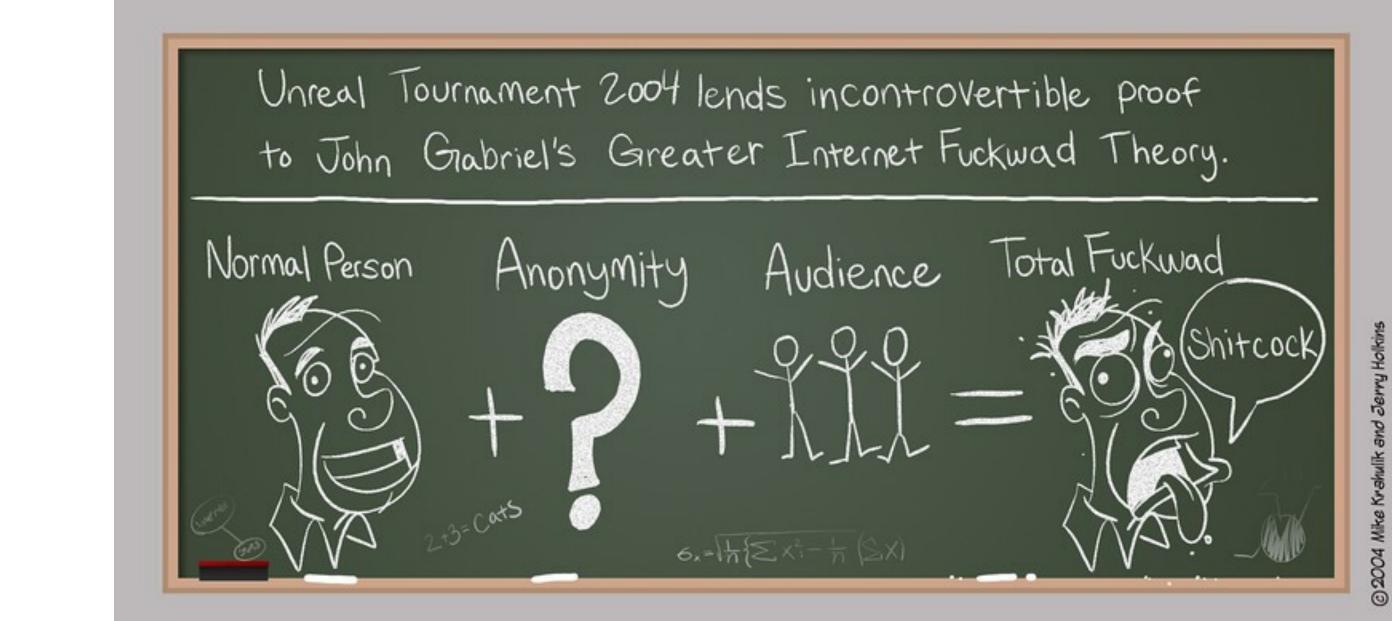


Piazza

Computer Science 61C Spring 2022

Berkeley EECS

- Piazza is an official channel
 - We will post announcements in it and we expect that, by posting announcements, you will read them
 - You can use this as a discussion forum to ask open questions
 - If you have private questions of the instructors and staff:
 do not use email, use a private question in Piazza
- Use Piazza, not email, to make requests of course staff
 - Piazza scales much better and allows us to keep track of things easier
- Non-enrolled students will not be allowed to join our Piazza forum, as we do not have enough bandwidth to answer questions from students who are not enrolled in the course.
- You can have posts be anonymous to other students but *not* to course staff.



Gradescope, Class Accounts, Concurrent Enrollment...

Computer Science 61C Spring 2022

- We will use Gradescope and PrairieLearn for all assignments
 - Midterm/Final with a regrade window
 - Homework on PL
 - Labs
 - Projects
- Class accounts:
 - https://inst.eecs.berkeley.edu/webacct
 - Login here with your Calnet ID to get course computer account starting on the first day of instruction.



Class Grading Policy: Fixed Bins or Curves, Whichever Is Better!

Computer Science 61C Spring 2022

- We have a set of fixed bins already defined
 - These bins are *guaranteed*: If you are in a bin, you will get *at least* that grade
- We have a departmental target (2.8-3.3 GPA, and we bias towards the high end)
 - http://www.eecs.berkeley.edu/Policies/ugrad.grading.shtml
- If bins result in above the target GPA ...
 - If students look to be doing better than expected ...
 we have smart students and don't need to make later assignments harder
- If bins are below our target GPA ... then we will curve the class



DSP...

Computer Science 61C Spring 2022

- We are happy to accommodate you, but we need to know
- So DSP students, please get your accommodations in now so we can schedule exams etc...
- We have a special GSI, Zoe Plaxco, and department course managers to handle/track a lot of the logistics
 - So don't be surprised if one of them reaches out to you
 - We also maintain access control: DSP information is only available on a need-to-know basis and most TAs don't need to know.



Late Policy for Projects... Slip Days!

Computer Science 61C Spring 2022

- Assignments due at 11:59:59 PM PT.
- You have 10 slip day tokens that apply to projects.
 No slip days for homework assignments (problem sets).
- Every day your project is late (even by a millisecond) we deduct a token
 - Can only use up to 4 slip day token on any given project/piece (e.g. 2A and 2B are separate)
- For projects, after you've used up all tokens, it's 1/3 of your points will be deducted per day.
 - (No credit if more than 3 days late.)
- No need for sob stories, just use a slip tokens!
- Grade in the end will use the optimal distribution of slip days between all your projects

Late Policy for Homeworks: 1/3rd every day

Computer Science 61C Spring 2022

- Every day a homework is late deducts 1/3 from the score
 - So zero credit after 3 days late
- Each homework has its own late counter
- If you need an extension due to extenuating circumstances, just ask on the extensions form!



Labs

Computer Science 61C Spring 202

- Labs are a key portion of the class that will help you with your projects
- We are trying a new lab format this semester
- You can either complete the lab on your own or in a lab section. There will be no lab checkoff
- If you attend lab section, the TA will guide you through the lab as a class to ensure that you can finish the lab in the allotted time



Labs

Computer Science 61C Spring 202

- Labs may be turned in one week late for half credit
- We will drop your lowest lab score at the end of the semester
- Lab 0 is completed on your own
 - No lab section available, but you can go to OH or ask questions on piazza if you need help
 - All other labs will have sections that you can attend, so if you have a question about them, go to a lab section or ask on piazza (no OH help)
- Lab 1 will have the option of attending a virtual section
- Hopefully the remaining labs will have both virtual and inperson sections



Beyond that: Extension Requests (update)

Computer Science 61C Spring 2022

- There are two forms on the web site:
 - One for normal extensions, one for if the manure is hitting the 2 MW Wind Turbine
- If you feel you have a need for an extension beyond slip days, use the appropriate form
- Our standard for granting extensions is very generous:
 Is it "reasonable"?
 - The TA responsible can say "yes" if its reasonable in their judgement (and their latitude is broad in terms of allowing extensions)
 - IF a TA might say "no", it instead becomes the instructors decision under the same "reasonable" standard



Exams

Computer Science 61C Spring 2022

- We will have two exams, a midterm and a final
 - There will be an alternate time slot immediately after the scheduled slot for those with a time conflict
 - There will be a second alternate time-slot skewed greatly for those on the other side of the planet
- Exams will be full hybrid
 - You can take an exam in person in a room like the Before Times at the scheduled time
 - You can take an exam remotely with remote proctoring
- You will default to in-person, but we will have a form a couple weeks before the exam to request alternates/remote exams
- You need to give us a reason but it is pretty much any reason:
 E.g. Nick takes exams better if his cat is on his lap.
 Berkeley EECS

Use Git and Push Often...

Computer Science 61C Spring 2022

- You will be using GitHub to host your projects for submission...
 - So use it for your normal workflow too
- Push your work back on a regular basis
 - It really prevents mistakes:
 "Ooops, go back" is the reason for version control
 - Your computer should be able to blow up, and you should only lose a couple hours work!
 - It gives a timestamp we can trust of when you wrote your code
 - Very useful if flagged for cheating
- Also, for any C coding, use Valgrind
- C is *not memory safe*, Valgrind will catch most of these errors when you make them. Berkeley EECS

Debugging and Office Hours...

Computer Science 61C Spring 202

- We are all here to help during office hours...
 - but we will *not* simply debug your project for you!
- In order to receive project assistance you must:
 - Have a test case which shows the problem
 - Have the debugger running and at a breakpoint that shows the problem
 - If it is a memory problem (segfault etc.) you must also have the project running in Valgrind to indicate where the problem is
- We also unfortunately have to enforce office hour time limits



Policy on Assignments and Independent Work

Computer Science 61C Spring 2022

- Projects can either be solo or in pairs, as can labs
- All homework is to be your work ALONE.
- You are encouraged to discuss your assignments with other students, but we expect that what you hand in is yours.
- It is NOT acceptable to copy (or even "start with") solutions from other students or the Web
- It is NOT acceptable to use PUBLIC GitHub archives (giving your answers away)
- We have tools and methods, developed over many years, for detecting this. You
 WILL be caught, and the penalties WILL be severe.
- Both Giver and Receiver are equally culpable and suffer equal penalties
 - If it is from a previous semester, the previous semester's students will also be reported to the student conduct office



Intellectual Honesty Policy: Detection and *Retribution*

Computer Science 61C Spring 2022

- We view those who would cheat as "attackers"
 - This includes sharing code on homework or projects, midterms, finals, etc...
 - But we (mostly) assume rational attackers:
 Benefit of attack > *Expected* cost
 - Cost of attack + cost of getting caught * probability of getting caught
- We take a detection and response approach
 - We use many tools to detect violations
 - "Obscurity is not security", but obscurity can help.
 Just let it be known that "We Have Ways"
 - We will go to DEFCON 1 (aka "launch the nukes") immediately
 - "Nick doesn't make threats. He keeps promises"





More On Academic Dishonesty...

Computer Science 61C Spring 2022

- We take cheating personally
 - We have an obligation to protect the value of this University for honest students:
 A critical threshold of cheating hurts everybody else
- Minimum penalty is negative points:
 - If we have just a 50% chance of catching you, you are provably better off just not doing the work. It is not rational to cheat
 - And penalties can go up from there:
 We can and have given Fs
 - Oh, and university policy says we can't reduce the penalty after you meet with us
- Nick takes and handles cheating cases personally
 - "I ought to of shot that dog myself, George.
 I shouldn't ought to of let no stranger shoot my dog."
 John Steinbeck, Of Mice and Men





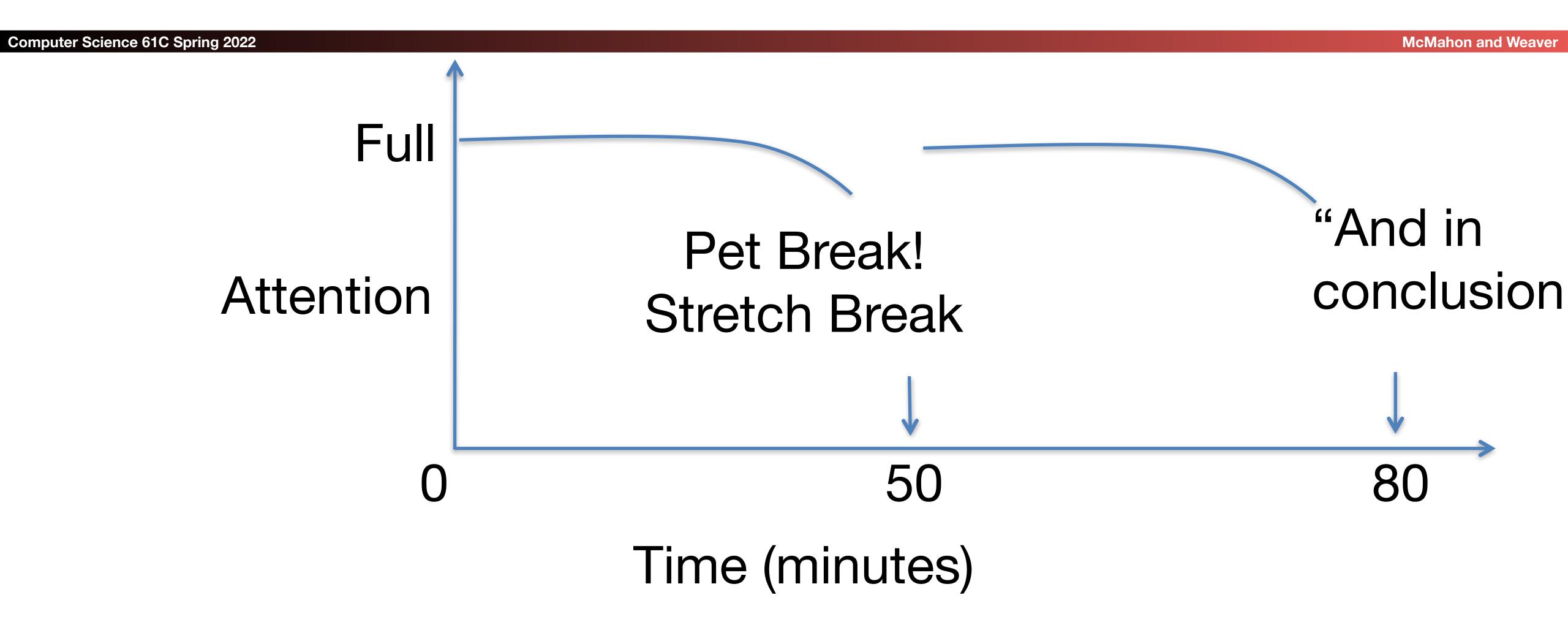
Stress Management & Mental Health...

Computer Science 61C Spring 2022

- We'll try to not over-stress you too much
 - But there really is a lot to cover and this really is a demanding major
- If you feel overwhelmed, please use the resources available
 - Academically: Ask on Piazza, Tutoring, Office hours, Guerrilla sections, etc...
 - We have lots and lots and lots of different ways for you to get help:
 Find the one that works for you!
 - Partially how we have scaled is not turning O(n) to O(log(n)) but allowing us to scale
 up the TA/tutor staff!
 - Non-Academic: Take advantage of University Health Services if you need to
 - Nick did! Zoloft (an antidepressant) and therapy saved my life, twice.



Architecture of a typical Lecture





Zoom And "Pet Time"

Computer Science 61C Spring 202

- Doing everything over Zoom sucks except for two things!
 - Chat and PETS!
- Use the chat to ask questions
 - We get more interactivity with the chat channel than we did with just "raise hand" in the lecture hall..
 - So we are going to use it!
- Lectures start at Berkeley Standard Time...
- But while we are on zoom, the 20 minutes before is "Pet Time"
 - We will bring our pets (when they are behaving) if we are operating remotely
 - You are encouraged to bring yours and share



Agenda

Computer Science 61C Spring 2022

- What you need to know about this class
- Thinking about Machine Structures
- Great Ideas in Computer Architecture
- Number Representation



CS61C is not really about C Programming

Computer Science 61C Spring 2022

- It is about the hardware-software interface
- What does the programmer need to know to make best use of the hardware.
- C is close to the underlying hardware, unlike other common languages like Python, Java, JavaScript, Go, R, Perl, Ruby, Haskell ...
 - Allows us to talk about key hardware features in higher level terms
 - Allows high performance
 - Only language which comes close is Rust...
- Also allows programmer to shoot themself in the foot in *amazingly* spectacular ways
- A goal in this class is for you to understand how C can be dangerous - leading to security loopholes and difficult bugs Berkeley EECS



Other Pedagogical Choices In This Class...





Computer Science 61C Spring 2022

Our processor of choice is the "RISC-V"

- Invented at Berkeley as a open-source hardware alternative to commercial processors, eg. x86, ARM
- Now gaining wide acceptance worldwide in academia and industry
- RISCs (Reduced Instruction Set Computers) by design are much simpler than tradition CISC architectures (x86 from Intel/AMD best examples) => simpler to learn, program, write compilers for, etc.
- RISC-V is our model for learning about processor hardware design
- We learn how to program at the processor instruction set level with RISC-V assembly language
- Our hardware design is in Logisim (schematics)
 - Later if you do hardware design, you'll only use schematics for circuit boards, everything else is Verilog or VHDL...
- But its harder to get up to speed on those Berkeley EECS

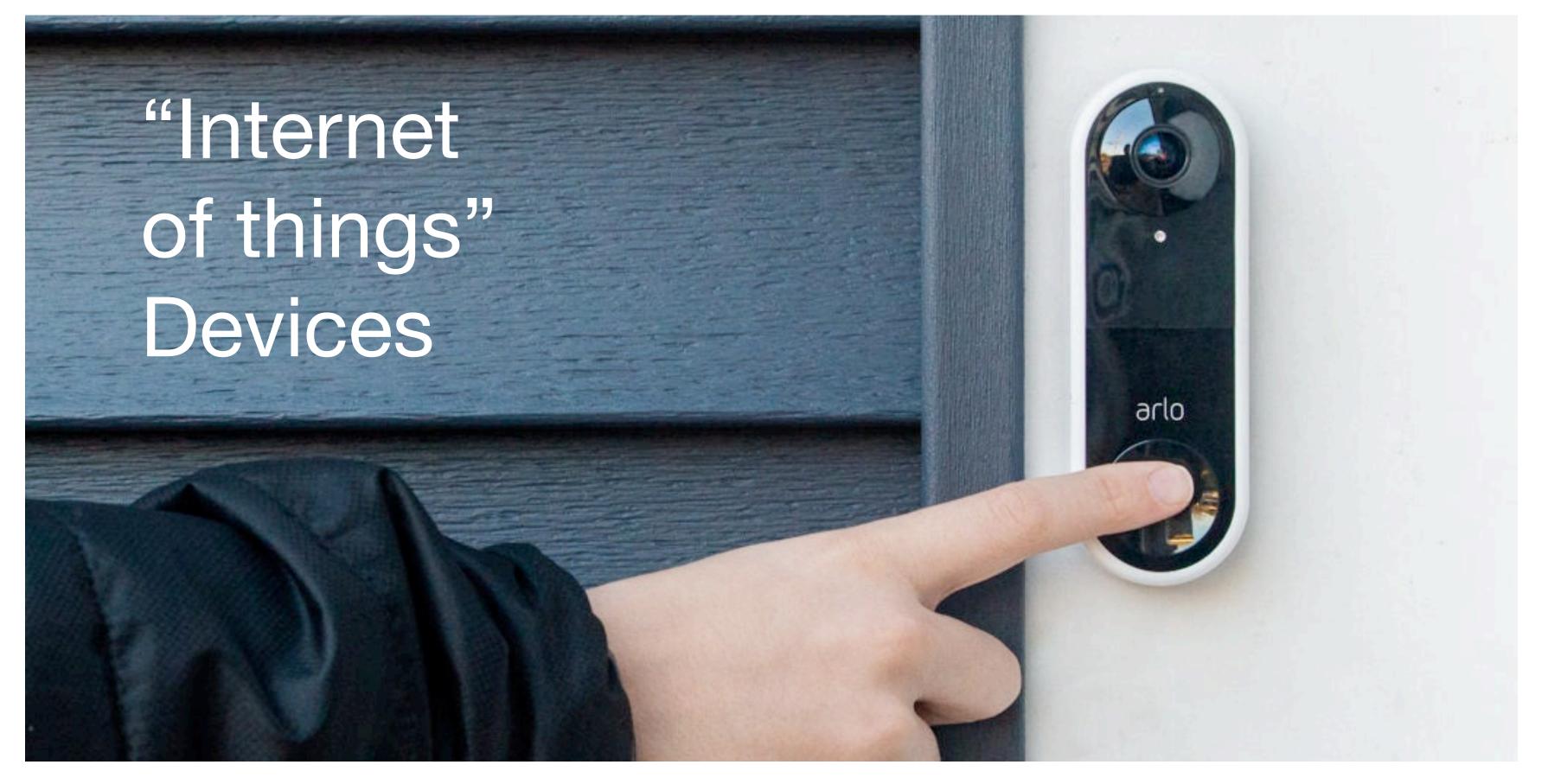
Modern 61C: From the small...

Computer Science 61C Spring 2022 **McMahon and Weaver** Personal Mobile Devices

To the very small...

Computer Science 61C Spring 2022

McMahon and Weaver





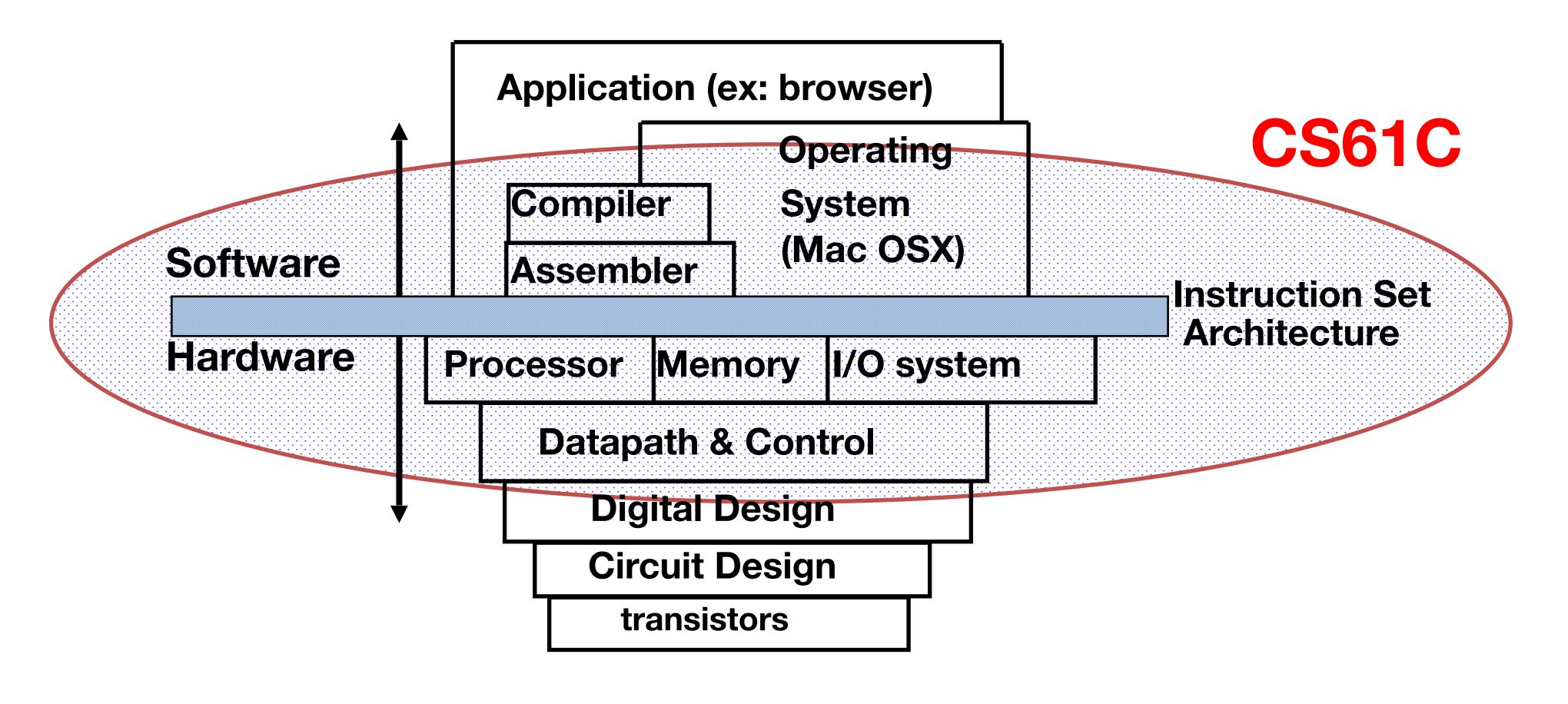
To the big...



Old School Machine Structures

Computer Science 61C Spring 2022

McMahon and Weaver



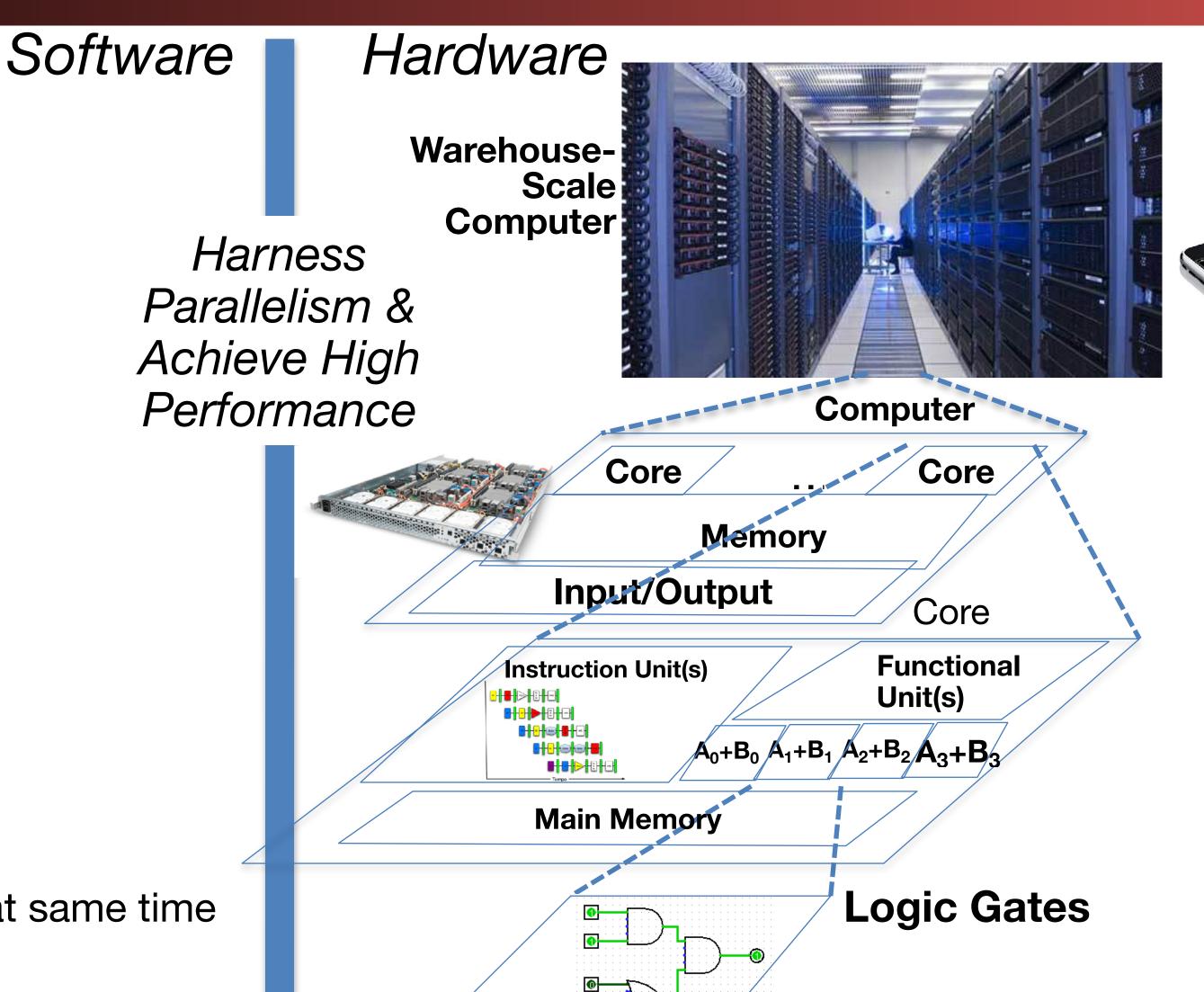


New School 61C: From the Data Center to the Hardware Logic Gates

Computer Science 61C Spring 2022

Parallel Requests
 Assigned to computer
 e.g., Search "giant military cats"

- Parallel Threads
 Assigned to core
 e.g., Lookup, Ads
- Parallel Instructions
 >1 instruction @ one time
 e.g., 5 pipelined instructions
- Parallel Data
 >1 data item @ one time
 e.g., Add of 4 pairs of words
- Hardware descriptions
 All gates functioning in parallel at same time



McMahon and Weaver

IoT Device

Smart

Phone

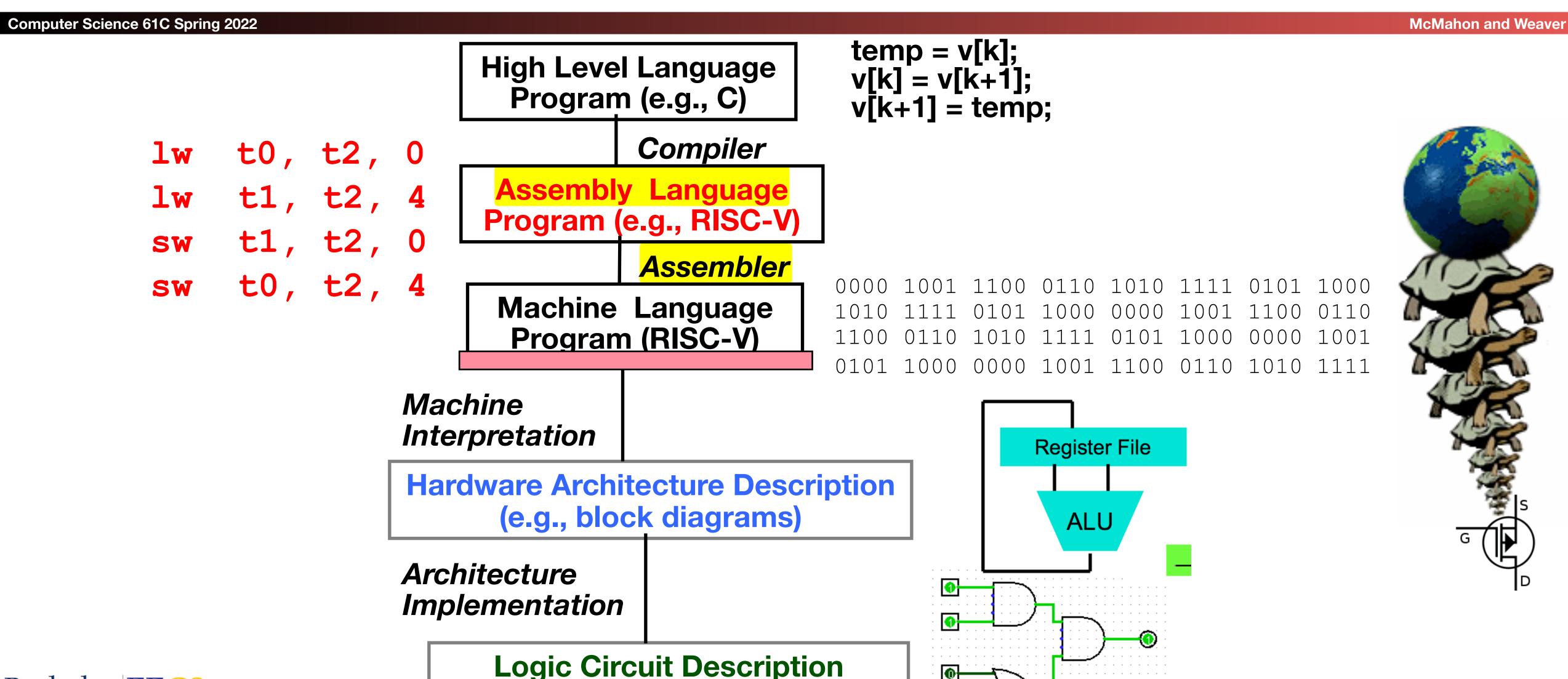
5 Great Ideas in Computer Architecture

Computer Science 61C Spring 2022

- Abstraction
 (Layers of Representation/Interpretation)
- 2. Moore's Law (Designing through trends)
- 3. Principle of Locality (Memory Hierarchy)
- 4. Parallelism & Amdahl's law (which limits it)
- 5. Dependability via Redundancy



Great Idea #1: Abstraction (Levels of Representation/Interpretation)



(Circuit Schematic Diagrams)

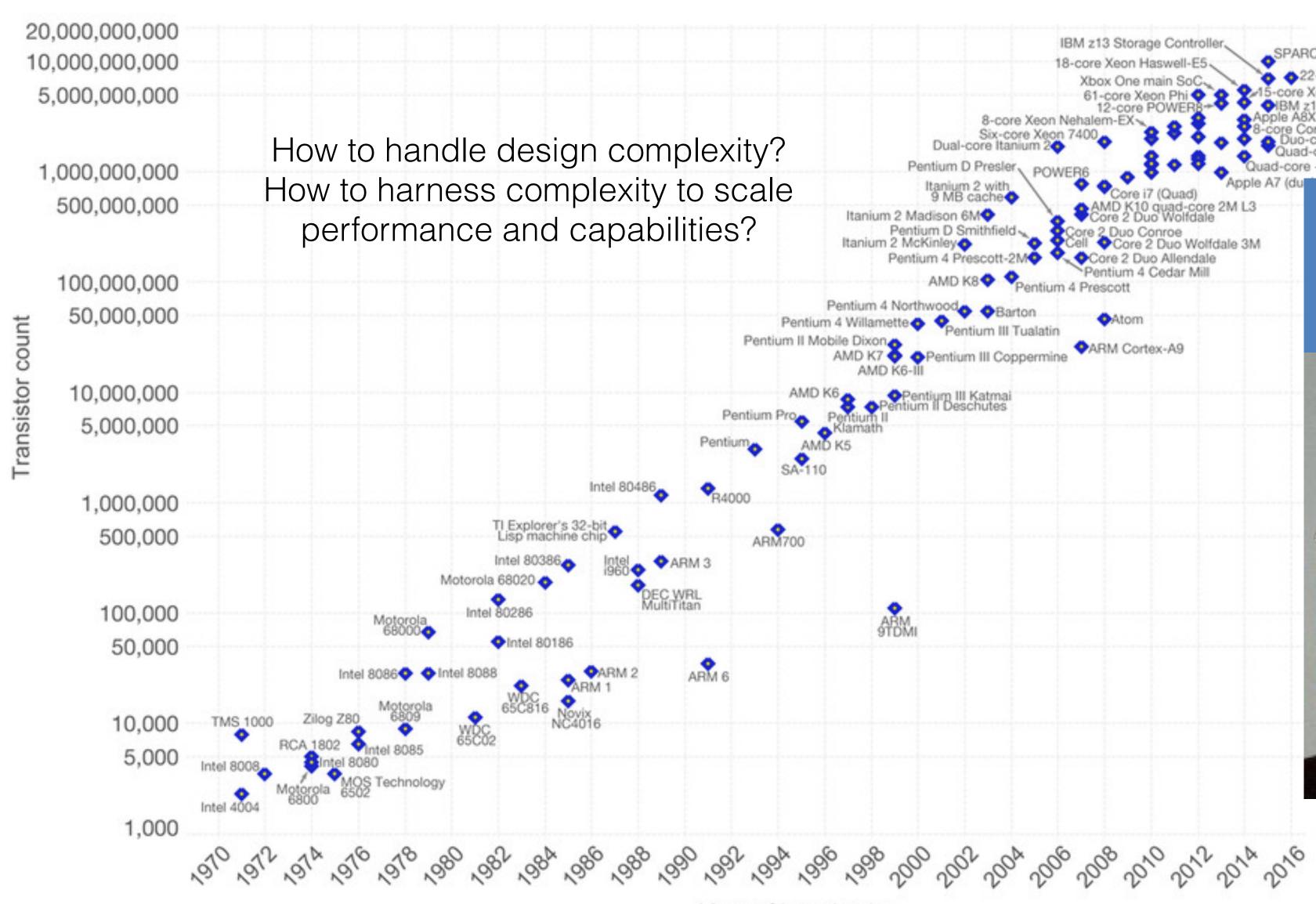
#2:

Moore's Law – The number of transistors on integrated circuit chips (1971-2016)



Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.





Predicts:
2X Transistors / chip
every 2 years

McMahon and Weaver

Gordon Moore Intel Cofounder B.S. Cal 1950!

Year of introduction

Interesting Times

Computer Science 61C Spring 202

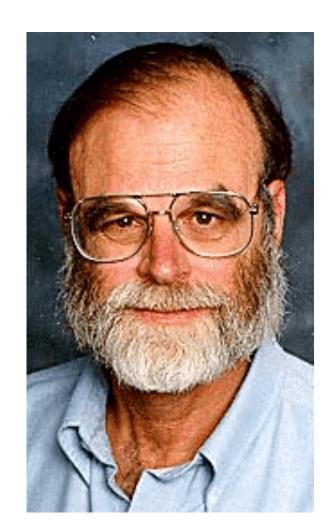
- Moore's Law meant that the cost of transistors scaled down as technology scaled to smaller and smaller feature sizes.
 - And the resulting transistors resulted in increased single-task performance
- But single-task performance improvements hit a brick wall years ago...
- State-of-the art commercial devices now 5nm
 - Latest TSMC/Apple silicon...
 This is only *part* of the reason the M1 processors are so fast!



Jim Gray's Storage Latency Analogy: How Far Away is the Data?

Computer Science 61C Spring 2022 Andromeda Tape Robot 2,000 Years Pluto Disk 2 Years 1.5 hr Sacramento Main Memory 100 This Campus 10 min On Board Cache 10 On Chip Cache This Room Registers 1 min

Head

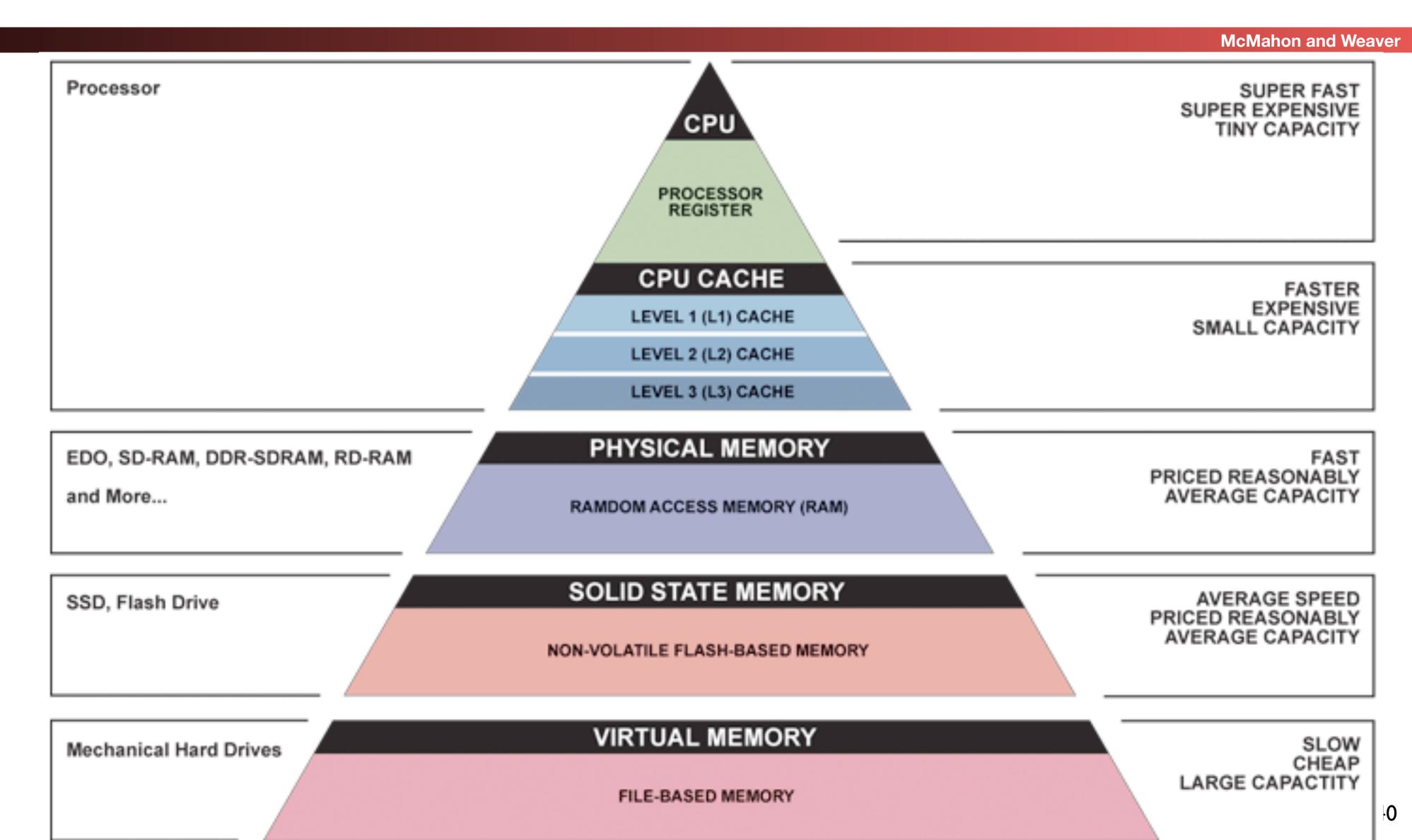


Jim Gray
Turing Award
B.S. Cal 1966
Ph.D. Cal 1969!

Great Idea #3: Principle of Locality/ Memory Hierarchy

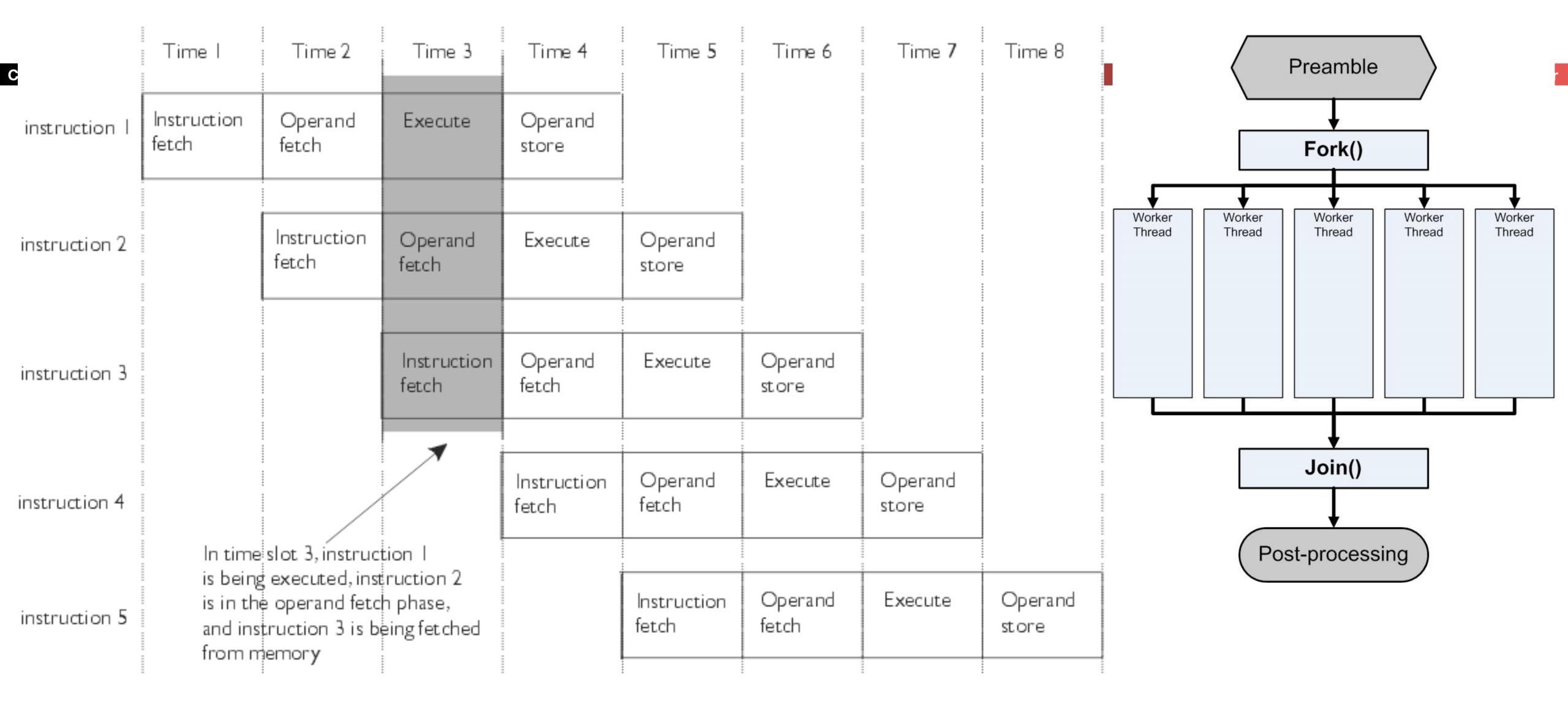
Memory
Hierarchy
effectively
creates a large
fast cheap
memory.

Computer Science 61C Spring 2022





Great Idea #4: Parallelism





And Parallelism is Everywhere

Computer Science 61C Spring 2022

Most evident at the processor level:

- Raspberry Pi 4 B+
 - Quad core processor at 1.5 GHz
 - Each core is 3-issue, out-of-order superscalar
 - Plus GPU, 1-4 GB RAM, 2x USB3, 2x USB2, Gigabit Ethernet, 2x HDMI...
 - \$35-55
 - Nick is working on a board to turn the Compute Module 4 to power a fully autonomous, vision-guided drone...
- Compare with a Cray-1 from 1975:
 - 8 MB RAM, 80 MHz processor, 300MB storage, \$5M+
- Or modern high end servers:
 - 2u server which supports 4 processors each processor can have 20+ cores, so 80 processor cores!
- But, as we will see, present at every level of hardware design
 - multiple memory units, arithmetic units, logic gates all operate in parallel, etc.
- Parallelism is what makes hardware semantics different from single-thread software semantics.





The Caveat: Amdahl's Law

Computer Science 61C Spring 2022

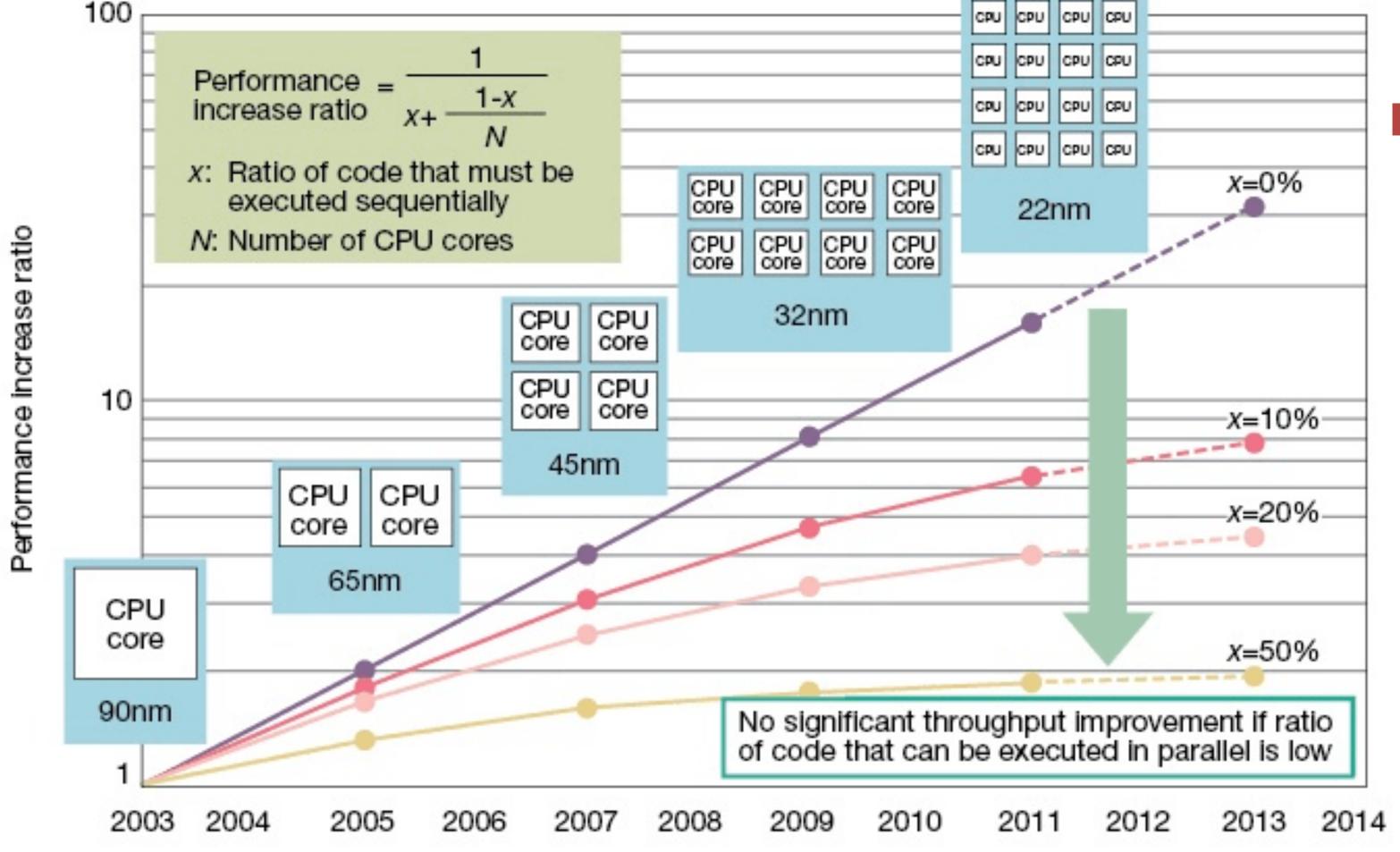
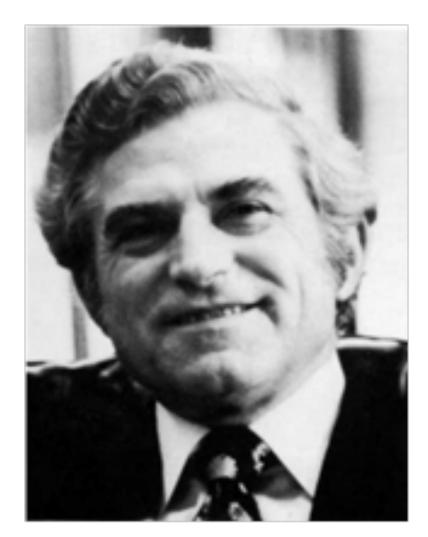


Fig 3 Amdahl's Law an Obstacle to Improved Performance Performance will not rise in the same proportion as the increase in CPU cores. Performance gains are limited by the ratio of software processing that must be executed sequentially. Amdahl's Law is a major obstacle in boosting multicore microprocessor performance. Diagram assumes no overhead in parallel processing. Years shown for design rules based on Intel planned and actual technology. Core count assumed to double for each rule generation.





Gene Amdahl Computer Pioneer

Great Idea #5: Failures Happen, so...

Computer Science 61C Spring 2022

- 4 disks/server, 50,000 servers
- Failure rate of disks: 2% to 10% / year
 - Assume 4% annual failure rate
- On average, how often does a disk fail?
 - a) 1 / month
 - b) 1 / week
 - C) 1 / day
 - d) 1 / hour



Coping with Failures

Computer Science 61C Spring 2022

McMahon and Weaver

- 4 disks/server, 50,000 servers
- Failure rate of disks: 2% to 10% / year
 - Assume 4% annual failure rate
- On average, how often does a disk fail?
 - a) 1 / month
 - b) 1 / week
 - C) 1 / day
 - d) 1 / hour

 $50,000 \times 4 = 200,000 \text{ disks}$ $200,000 \times 4\% = 8000 \text{ disks fail}$ $365 \text{ days } \times 24 \text{ hours} = 8760 \text{ hours}$

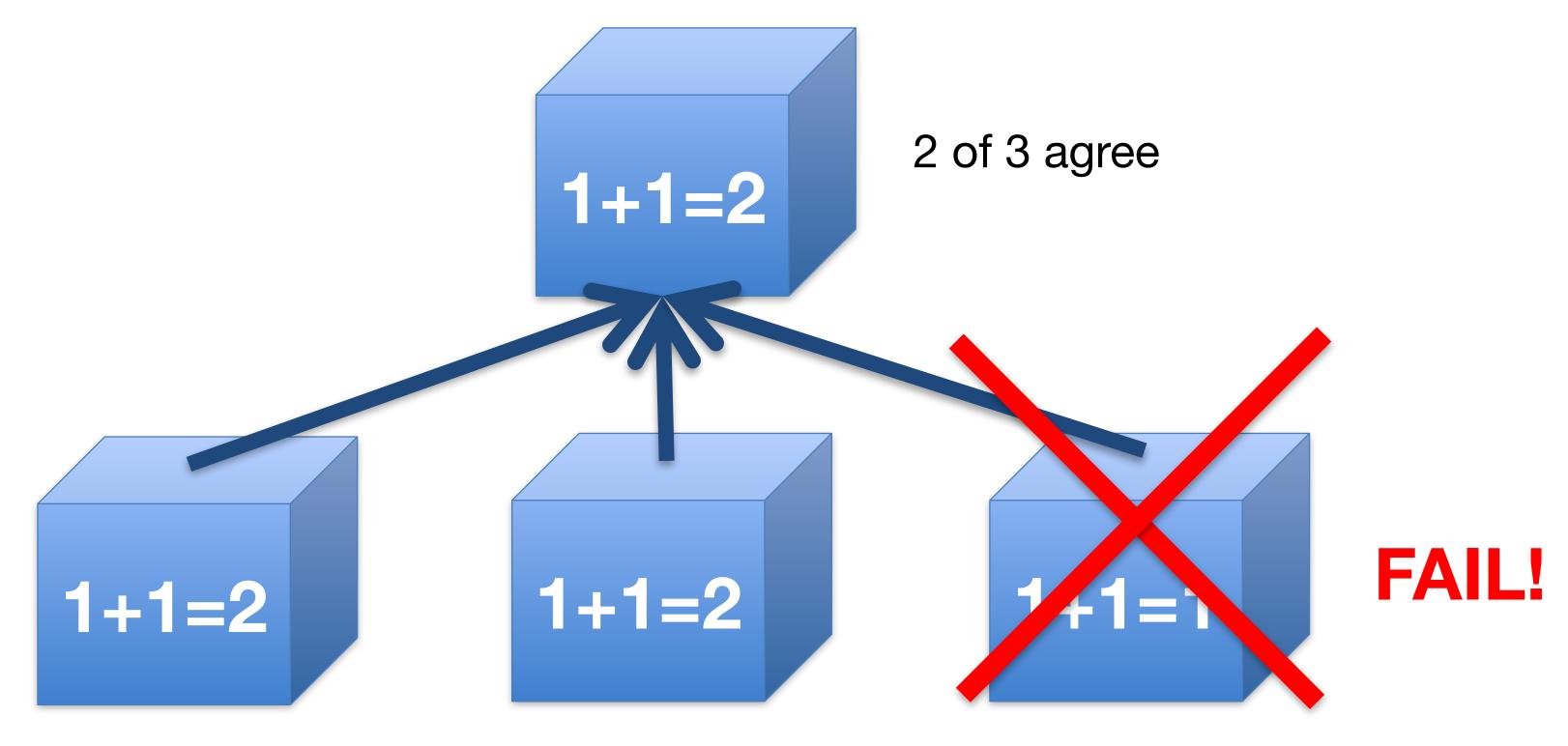


Great Idea #5: Dependability via Redundancy

Computer Science 61C Spring 2022

McMahon and Weaver

 Redundancy so that a failing piece doesn't make the whole system fail



Increasing transistor density reduces the cost of redundancy



Great Idea #5: Dependability via Redundancy

Computer Science 61C Spring 2022

- Applies to everything from datacenters to storage to memory to instructors
 - Redundant datacenters so that can lose 1 datacenter but Internet service stays online
 - Redundant computers was Google's original internal innovation
 - Redundant disks so that can lose 1 disk but not lose data (Redundant Arrays of Independent Disks/RAID)
 - Redundant memory bits so that can lose 1 bit but no data (Error Correcting Code/ECC Memory/"Chipkill" memory)
 - Redundant instructors!







Summary

Computer Science 61C Spring 2022

- CS61C: Learn 5 great ideas in computer architecture to enable high performance programming via parallelism, not just learn C
 - Abstraction
 (Layers of Representation/Interpretation)
 - 2. Moore's Law
 - 3. Principle of Locality/Memory Hierarchy
 - 4. Parallelism
 - 5. Dependability via Redundancy



Agenda

Computer Science 61C Spring 2022

- What you need to know about this class
- Thinking about Machine Structures
- Great Ideas in Computer Architecture
- Number Representation



What is Binary?

Computer Science 61C Spring 2022

- A method of representing numbers using a string of 0s and 1s
- Why do computers use binary?
 - The fundamental building block of a computer is a transistor which can only represent two values. We decided to label these two values as 0 and 1
- Denoted by prepending a 0b or appending a subscripted 2 to the string



Terms

Computer Science 61C Spring 2022

- Bit
 - 1 binary digit
- Nibble
 - 4 bits
- Byte
 - 8 bits
- Base
- The number of different digits that a system has to represent numbers
 Berkeley EECS

- Most significant bit (MSB)
 - The bit in the highest position
 - 1 0 0 1 0 1
- Least significant bit (LSB)
 - The bit in the lowest position

1	0	0	1	0	1	0	1
---	---	---	---	---	---	---	---

Binary Table

Computer Science 61C Spring 2022 McMahon and Weaver

Leading zeros don't change the value!

Decimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111



Powers of 2

Computer Science 61C Spring 2022 McMahon and Weaver

Decimal	Binary(0b)	Powers of 2
0	0000	
1	0001	20
2	0010	21
3	0011	
4	0100	22
5	0101	
6	0110	
7	0111	
8	1000	23
9	1001	
10	1010	
11	1011	
12	1100	
13	1101	
14	1110	
15	1111	



Powers of 2

Computer Science 61C Spring 2022

McMahon and Weaver

The exponents correspond to the bits that are 1!

Decimal	Binary(0b)	Powers of 2
0	0000	
1	0001	20
2	0010	21
3	0011	21+20
4	0100	2 ²
5	0101	22+20
6	0110	22+21
7	0111	22+21+20
8	1000	23
9	1001	23+20
10	1010	23+21
11	1011	23+21+20
12	1100	23+22
13	1101	23+22+20
14	1110	23+22+21
15	1111	23+22+21+20



Powers of 2

Computer Science 61C Spring 2022

McMahon and Weaver

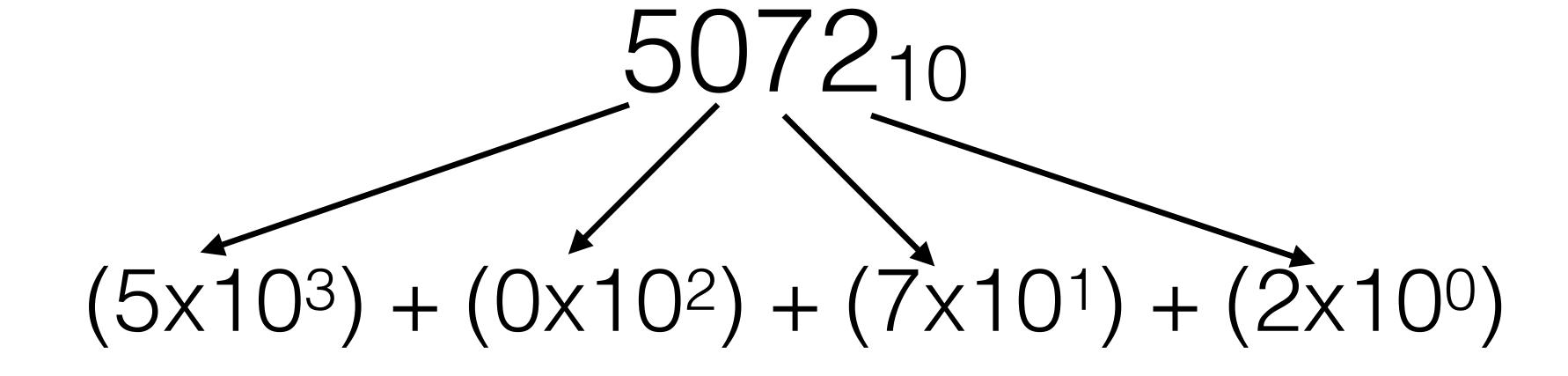
Result
1
2
4
8
16
32
64
128
256
512
1024
2048
4096



Decimal Notation

Computer Science 61C Spring 2022

McMahon and Weaver



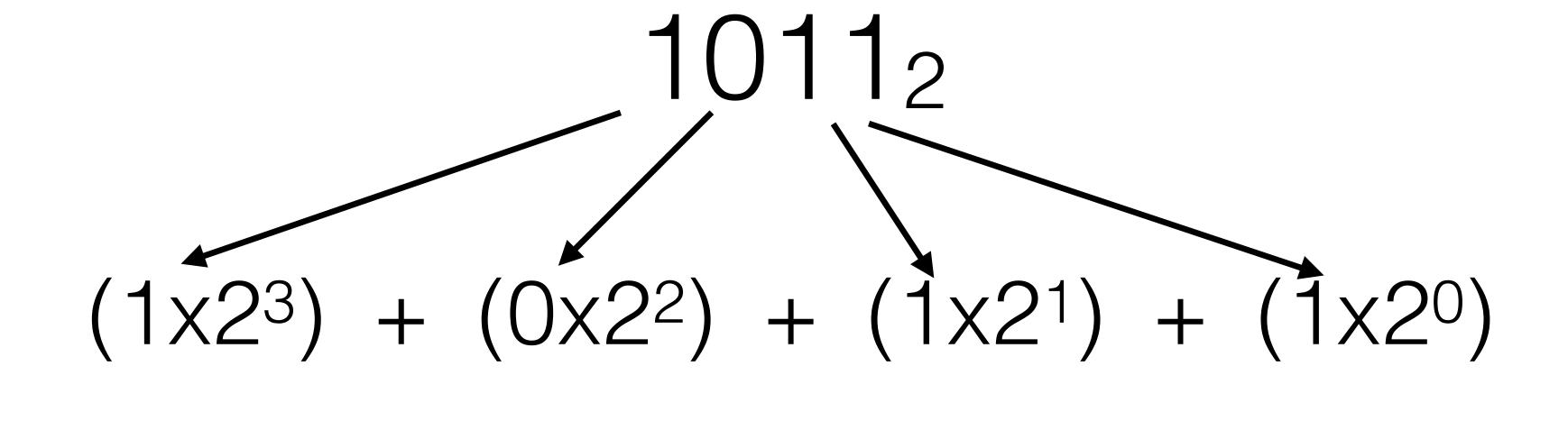
$$5000 + 0 + 70 + 2$$

507210

Binary to Decimal

Computer Science 61C Spring 2022

McMahon and Weaver



$$8 + 0 + 2 + 1$$

11₁₀

Decimal to Binary

Computer Science 61C Spring 2022

- Determine the largest power of 2 that is less than or equal to the current number
- 2. Put a 1 in that position in the final answer
- 3. Subtract that power of 2 from the current number
- 4. Repeat until the current number is 0

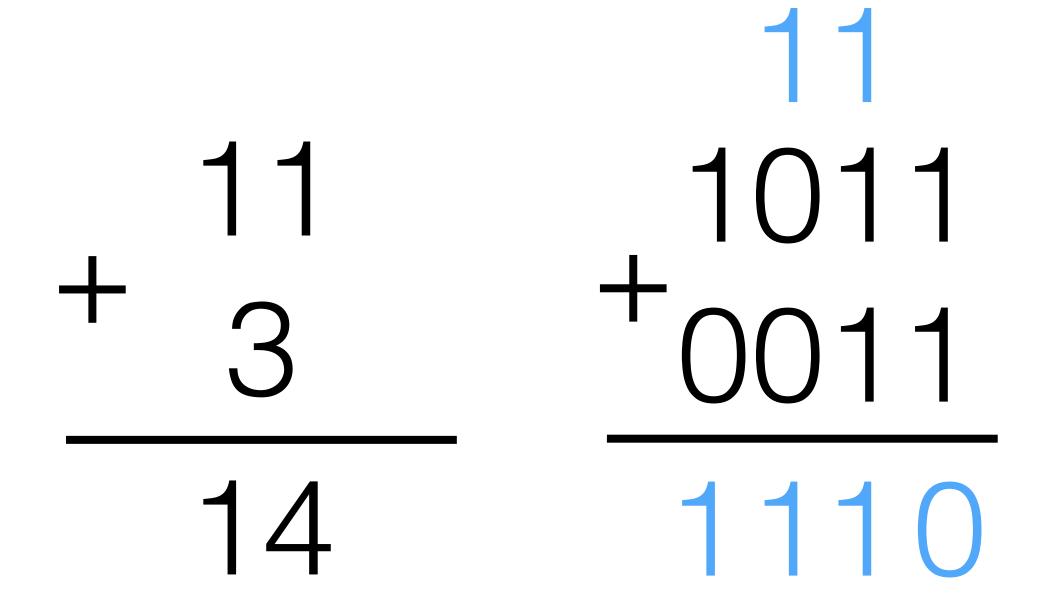
 Berkeley EECS

16

Binary Addition

Computer Science 61C Spring 2022

McMahon and Weaver





Binary Addition - Overflow

Computer Science 61C Spring 2022

- Overflow occurs when you cannot represent the result of the operation in the given number of bits
- When you overflow, you end up with an incorrect result



How many numbers can we represent given n bits?

Computer Science 61C Spring 2022

- Each bit can either be a 0 or 1
- With n bits, we can represent 2ⁿ values
- Ex:
 - How many numbers can we represent with 5 bits?
 - XXXXX
 - 2x2x2x2x2
 - 25
 - 32
- If our range starts at 0, then we can represent [0, 2ⁿ-1] values



Hexadecimal

Computer Science 61C Spring 2022

- Method of representing binary that's easier for humans to read
- Base 16
 - One hex digit can represent 16 numbers
 - One hex digit = 1 nibble
- Denoted by prepending "0x" or appending a subscripted 16
 Berkeley EECS

Decimal	Binary(0b)	Hex(0x)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	В
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Binary to Hex

Computer Science 61C Spring 2022

- 1. Group the digits into groups of 4 from right to left
- 2. Prepend any leading zeros needed to make the leftmost group have four binary digits
- 3. Convert each group to the corresponding hex character

0b111010

0011

1010

McMahon and Weaver

3

A

0x3A



Hex to Binary

Computer Science 61C Spring 2022

McMahon and Weaver

- Convert each hex digit to their corresponding binary digits
- 2. Group the terms together

0x61C

0110

000

1100

0b011000011100



Decimal to Hex Hex to Decimal

Computer Science 61C Spring 2022

McMahon and Weaver

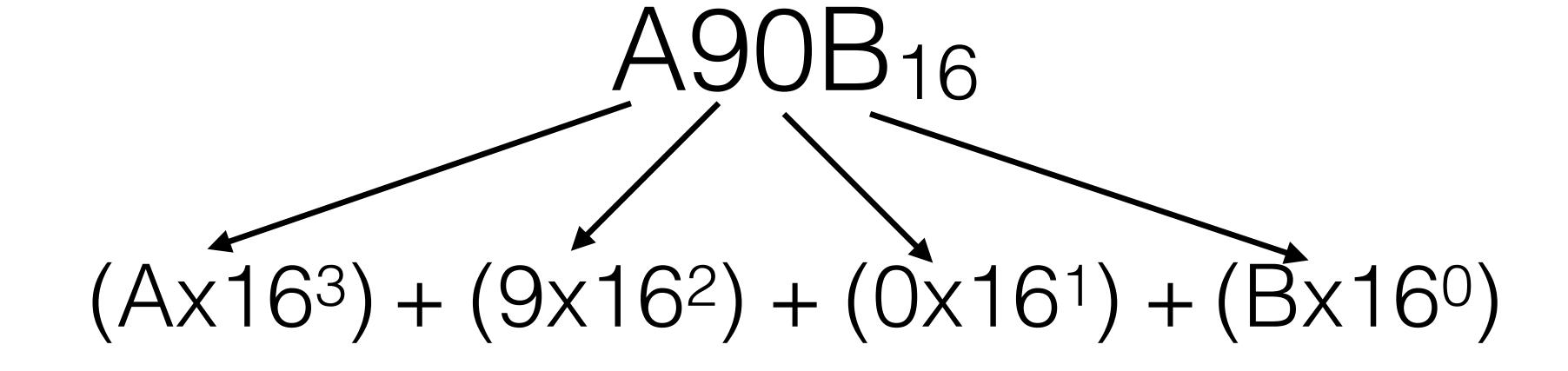
Convert to binary then convert to the desired base



Hex to Decimal

Computer Science 61C Spring 2022

McMahon and Weaver



$$40960 + 2304 + 0 + 11$$

4327510



Decimal to Hex

Computer Science 61C Spring 2022

1. Determine the largest power of 16 that is less than or equal to the current number (n)

- 2. Determine how many of this number will fit into the current number (n*b)
- 3. Subtract (n*b) from the current number
- 4. Repeat until the current number is 0

11082 2(4096)2890 11(256) 74 4(16)

Ox 2 B 4 A 4096 256 16 1

Sign and Magnitude

Computer Science 61C Spring 2022

- We need a way to represent negative values
- The MSB represents the sign of the value
 - 0 stands for positive
 - 1 stands for negative
- The remaining bits represent the magnitude of the number
- Ex: Convert -5 to sign and magnitude using 5 bits:
 - Sign = 1
 - Magnitude = 0101
 - Answer = 10101



Range of Sign and Magnitude

Computer Science 61C Spring 2022

- Recall that the range of unsigned representation is [0, 2ⁿ-1]
- What is the range of negative values that we can represent?
 - XXXXX
 - $[-(2^{n-1}-1), -0]$
- What is the range of positive values that we can represent?
 - XXXXX
 - [+0, 2ⁿ⁻¹-1]
- Putting it together
 - [-(2ⁿ⁻¹-1), 2ⁿ⁻¹-1]



Problems with Sign and Magnitude

Computer Science 61C Spring 2022

McMahon and Weaver

- Two zeros
 - Makes hardware and programming implementations more difficult
- It's difficult to implement in hardware because of the sign bit



One's Complement

Computer Science 61C Spring 2022

- Another way to represent negative numbers
- To form the negative numbers, flip the bits of the positive form of the number
- The MSB represents the sign of the value
 - 0 means its positive
 - 1 means its negative



One's Complement Examples

Computer Science 61C Spring 2022

- Ex1: What is the one's complement representation of -5 using 6 bits?
 - Unsigned: 000101
 - Flip bits: 111010
- Ex2: What's the one's complement representation of 6 using 4 bits?
 - Unsigned 0110
 - No need to flip bits
 - Answer: 0110

- Ex3: Convert the following one's complement number to decimal: 0b11001
 - MSB is 1, so it's negative
 - Flip bits: 0b00110
 - Answer: -6
 - Ex4: Convert the following one's complement number to decimal: 0b01001
 - MSB is 0, so its positive
 - No need to flip bits
 - Answer: 9



Range of One's Complement

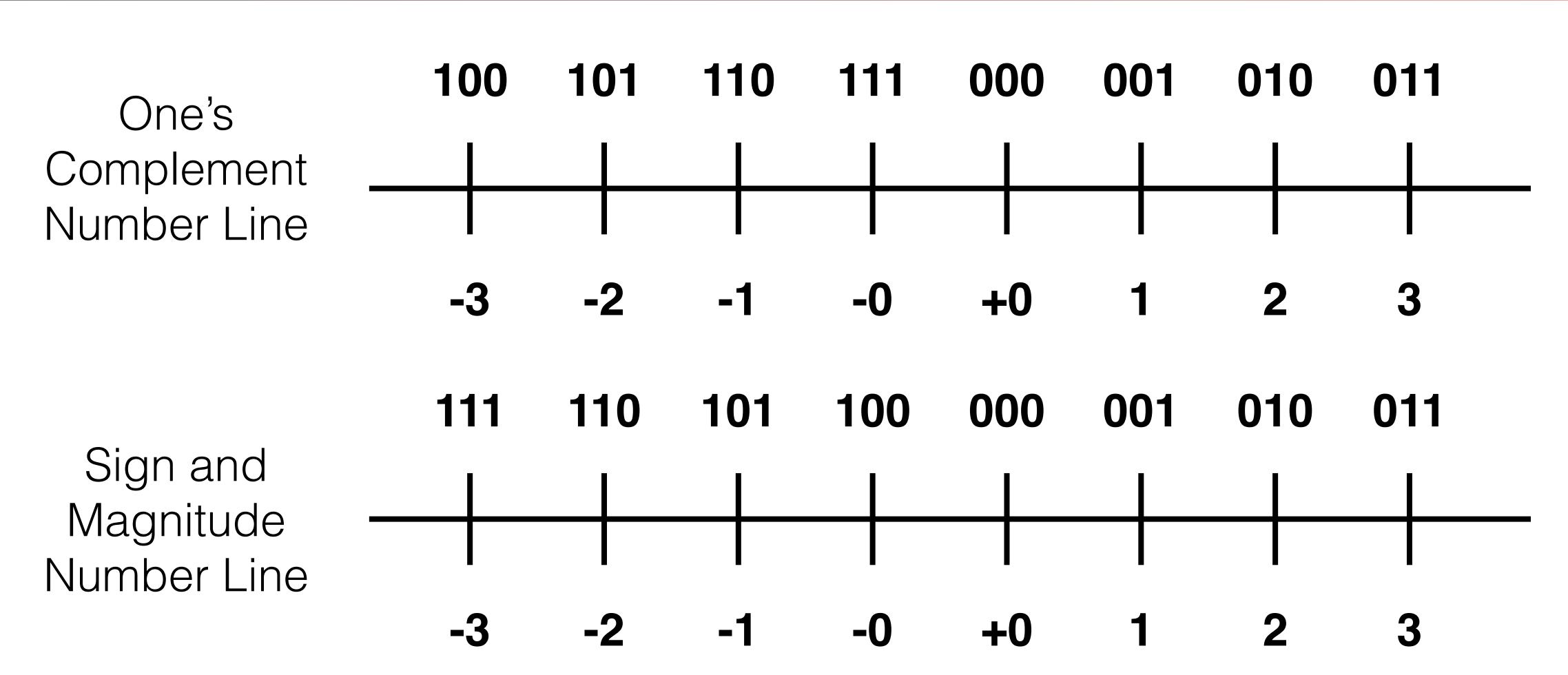
Computer Science 61C Spring 2022

- Recall that the range of sign and magnitude is [-(2ⁿ⁻¹-1), 2ⁿ⁻¹-1]
- What is the range of negative values that we can represent?
 - XXXXX
 - Same as sign and magnitude
 - $-(2^{n-1}-1), -0]$
- What is the range of positive values that we can represent?
 - XXXXX
 - Same as sign and magnitude
 - $[+0, 2^{n-1}-1]$
- Putting it together
- [-(2ⁿ⁻¹-1), 2ⁿ⁻¹-1]
 Berkeley EECS

One's Complement vs Sign and Magnitude Number Lines

Computer Science 61C Spring 2022

McMahon and Weaver





One's Complement Shortcomings

Computer Science 61C Spring 2022

McMahon and Weaver

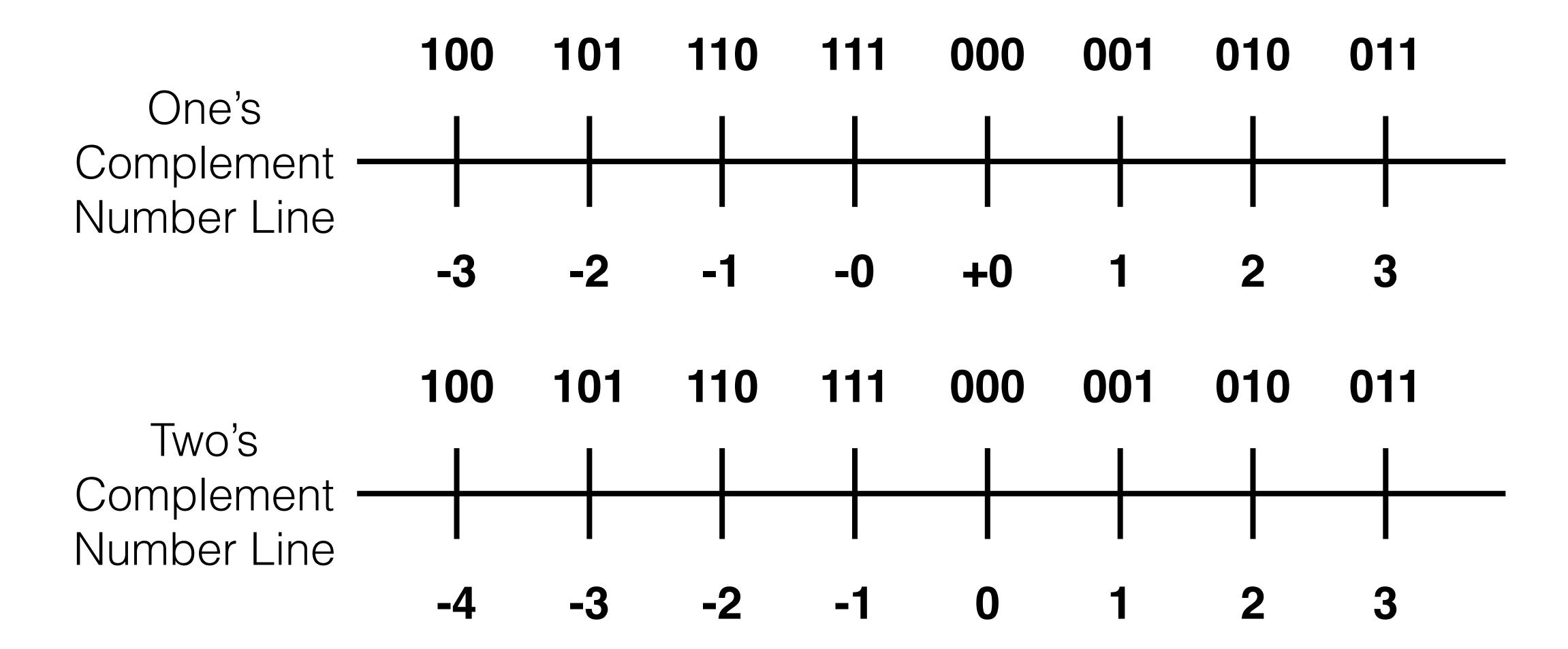
Still have two zeros



Two's Complement

Computer Science 61C Spring 2022

McMahon and Weaver





Two's Complement

Computer Science 61C Spring 2022

- How do we do this arithmetically?
 - Flip the bits and add one
- The MSB represents the sign of the value
 - 0 means its positive
 - 1 means its negative
- We only have one zero now!
- Easier to implement in hardware



Two's Complement Examples

Computer Science 61C Spring 2022

- Ex1: Represent -7 in two's complement with 5 bits
 - Unsigned: 0b00111
 - Flip bits: 0b11000
 - Add one: 0b11001
- Ex2: Represent 10 in two's complement with 6 bits
 - Unsigned: 0b001010
 - Positive, so no need to flip bits
 - Answer: 0b001010

- Ex3: Convert the following two's complement number to decimal: 0b11010
 - MSB is 1, so it's negative
 - Flip bits: 0b00101
 - Add one: 0b00110
 - Answer: -6
- Ex4: Convert the following two's complement number to decimal: 0b01111
 - MSB is 0, so it's positive
 - No need to flip bits
 - Answer: 15



Two's Complement Range

Computer Science 61C Spring 2022

- Recall that the range for one's complement is [-(2ⁿ⁻¹-1), 2ⁿ⁻¹-1]
- What is the range of negative values we can represent?
 - Remember that we shifted the negative values over by one, meaning that we can represent one more negative value
 - $-2^{n-1}, 0$
- What is the range of positive values we can represent?
 - Same as one's complement
 - [0, 2ⁿ⁻¹-1]
- Putting it together
 - [-2ⁿ⁻¹, 2ⁿ⁻¹-1]

Two's Complement Addition

Computer Science 61C Spring 2022

McMahon and Weaver

Arithmetic in two's complement works!



Two's Complement Addition

Computer Science 61C Spring 2022

McMahon and Weaver

0101



Two's Complement Overflow

Computer Science 61C Spring 202

- Overflow: when the result of an operation cannot be represented in the given number of bits
- When adding two positive numbers, overflow occurs when the result is negative
- When adding two negative numbers, overflow occurs when the result is positive
- Overflow will never occur when adding two numbers of opposite signs



Two's Complement Overflow Examples

111 0111 1111 1111 1111 1111 1111 1111 1100 0101 No Overflow 1100 1101 1101 1101 1101 1101 1000 1000 1000 1100	Computer Science 61C Spring 2022			MoMahan and Wasy	IOK-
$ \begin{array}{c} $	-0111 -0101	Overflow	111 1111 +0110 0101	No Overflow	er
Berkeley EECS	1 1000 1001 1001 0001 Berkeley EECS	Overflow	1101 +1011	Overflow	33

Bias Encoding

Computer Science 61C Spring 2022

- A method for storing a range of values where the lowest value is encoded as all zeros
- We'll see an application of this in the floating point lecture



Bias Encoding (unbiased -> biased steps)

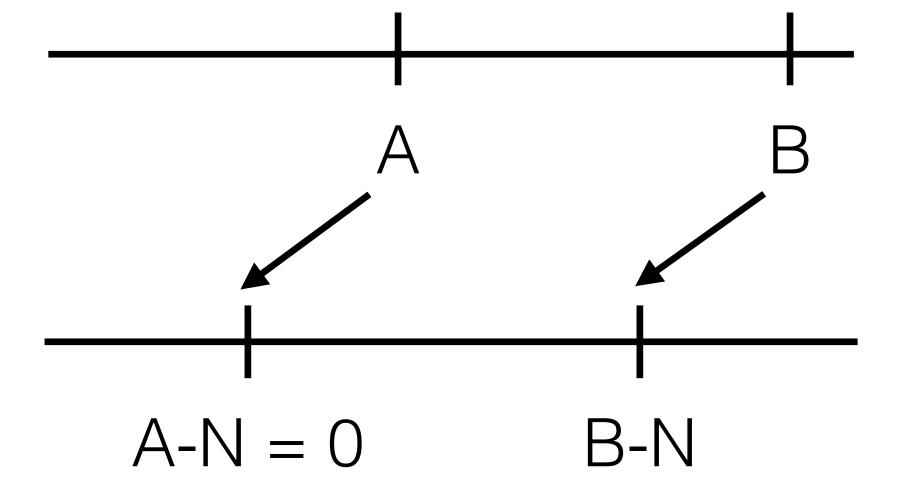
Computer Science 61C Spring 2022

McMahon and Weave

- 1. Translate the number so that the lowest value is at zero
 - · The amount that we translate by is called the bias (N)
- 2. Convert the number to binary

Original Number Line

> Biased Number Line



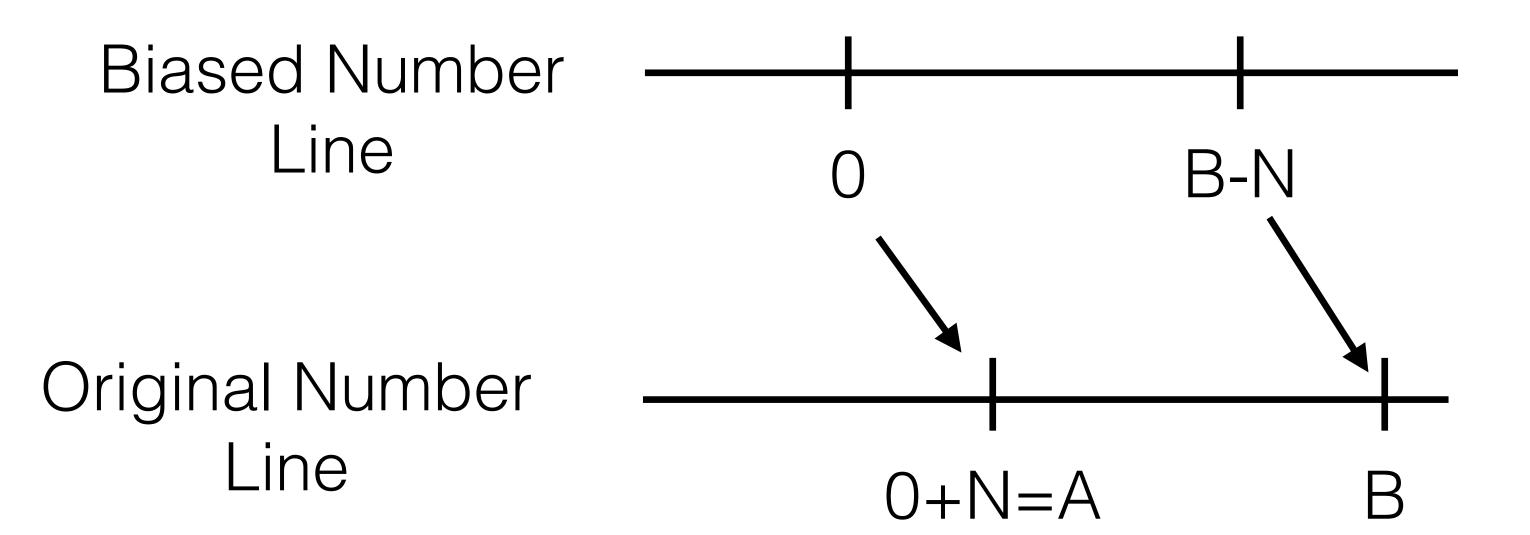


Bias Encoding (biased -> unbiased steps)

Computer Science 61C Spring 2022

1. Convert the number to decimal

2. Add the bias





Bias Encoding Convention

Computer Science 61C Spring 2022

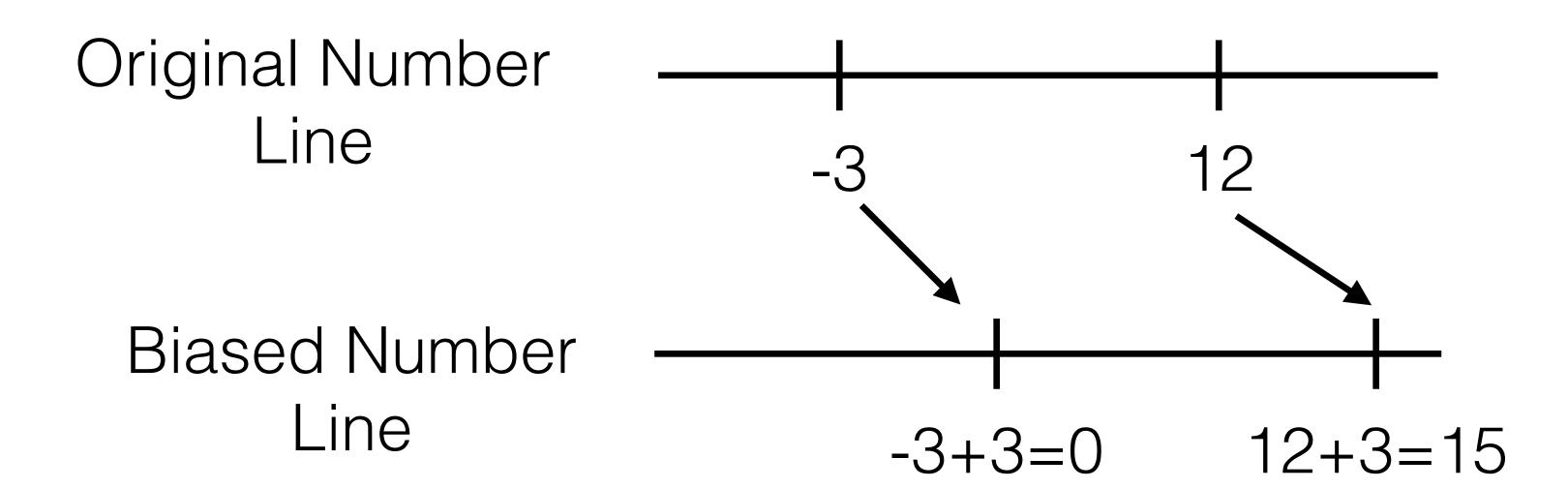
- Non-biased encoding -> bias encoding
 - Subtract the bias
- Biased encoding -> non-biased encoding
 - Add the bias



Bias Encoding

Computer Science 61C Spring 2022

- You can set the bias to be anything you want
- If you want to represent the [-3, 12], you can set the bias to
 -3 in order to make the range be [0, 15]
 - This enables you to represent the range [-3, 12] with only 4 bits!





Calculating the Bias (for 2's complement)

Computer Science 61C Spring 202

- Recall that the range of a 2's complement number with n bits is [-2ⁿ⁻¹, 2ⁿ⁻¹-1]
- Bias Formula for a 2's complement number with n bits
 - $N = -(2^{n-1}-1)$
- Hold on, $-2^{n-1} -(2^{n-1}-1) = -1$. That's not zero
 - In floating point representation, all ones is used for a special case

Bias Examples (2's Complement)

Computer Science 61C Spring 2022

- Covert -9 to bias notation with 5 bits
 - Bias = $-(2^{n-1}-1)$
 - Bias = $-(2^{5-1}-1)$
 - Bias = -15
 - -9 + 15 = 6
 - Bias Encoding = 00110

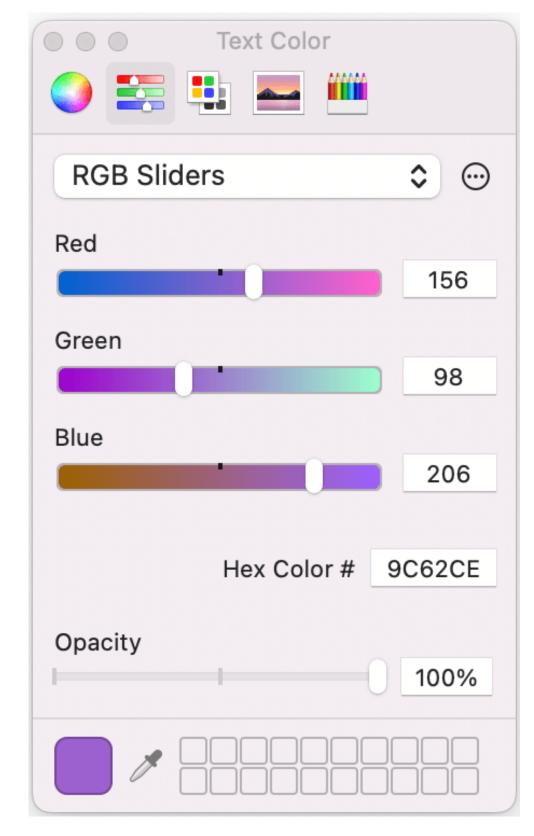
- What number is being represented in this biased encoding? 0b1010
 - Bias = $-(2^{n-1}-1)$
 - Bias = $-(2^{4-1}-1)$
 - Bias = -7
 - 10 7 = 3

Big Idea

Computer Science 61C Spring 2022

McMahon and Weaver

 Bits can represent anything, you can make them mean anything you want





Powers of 2

Computer Science 61C Spring 2022 McMahon and Weaver

Result
1
2
4
8
16
32
64
128
256
512
1024
2048
4096



SI Prefixes

Computer Science 61C Spring 2022

McMahon and Weaver

Name	Abbreviation	Power of 2	Power of 10
Kilo	K	210	10 ³
Mega	M	220	106
Giga	G	230	109
Tera	T	240	1012
Peta	Р	2 50	10 ¹⁵
Exa	E	260	10 ¹⁸
Zetta	Z	270	1021
Yotta	Y	280	1024



IEC Standard Prefixes

Computer Science 61C Spring 2022

McMahon and Weaver

Name	Abbreviation	Power of 2
kibi	Ki	210
mebi	Mi	220
gibi	Gi	230
tebi	Ti	240
pebi	Pi	250
exbi	Ei	260
zebi	Zi	270
yobi	Yi	280



Converting powers of 2 to IEC notation

Computer Science 61C Spring 2022

- 2XY
 - X tells us the prefix
 - Y tells us the number that precedes the prefix (ex: 2 Ti)

$$X = 0 => no prefix$$

$$X = 1 => kibi$$

$$X = 2 => mebi$$

$$X = 3 => gibi$$

$$X = 4 =$$
 tebi

$$X = 5 => pebi$$

$$X = 6 => exbi$$

$$X = 7 => zebi$$

$$X = 8 => yobi$$

$$Y = 0 = 1$$

$$Y = 1 => 2$$

$$Y = 2 => 4$$

$$Y = 3 = 8$$

$$Y = 4 = > 16$$

$$Y = 5 => 32$$

$$Y = 6 = > 64$$

$$Y = 7 => 128$$

$$Y = 8 = 256$$



IEC Examples

Computer Science 61C Spring 2022

- Write the following in IEC notation: 2⁷⁵
 - X = 7 = zebi
 - Y = 5 => 32
 - Answer: 32 Zi

- Write the following as a power of 2: 64 Ti
 - Ti => X = 4
 - $Y = log_2(64) = 6$
 - Answer: 2⁴⁶

Additional Problems

Computer Science 61C Spring 2022

- Unfortunately, we cannot go over all of the examples needed to fully cover this topic due to the large amount of material that we are required to teach in the course
- Please attend discussion or review the discussion worksheet for more examples including
 - Two's complement subtraction
 - Hex arithmetic



Summary

Computer Science 61C Spring 2022

- Binary, Decimal, Hex
- Sign and Magnitude
- One's Complement
- Two's Complement
- Bias
- IEC Prefixes

