The Hardware/Software Interface

CSE 351 Autumn 2023

Instructor:
Justin Hsia

Teaching Assistants:

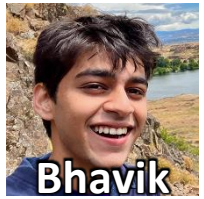
Afifah Kashif	Malak Zaki
Bhavik Soni	Naama Amiel
Cassandra Lam	Nayha Auradkar
Connie Chen	Nikolas McNamee
David Dai	Pedro Amarante
Dawit Hailu	Renee Ruan
Ellis Haker	Simran Bagaria
Eyoel Gebre	Will Robertson
Joshua Tan	

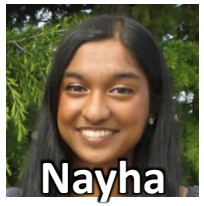
http://xkcd.com/676/

Quarter Specifics

Course Staff

- Instructor: just call me Justin
  - CSE Associate Teaching Professor
  - Raising a toddler takes up energy and dictates my schedule
- TAs:





- More than anything, we want you to feel…
  - ✓ Comfortable and welcome in this space
  - ✓ Able to learn and succeed in this course
  - ✓ Comfortable reaching out if you need help or want change

Bookmarks

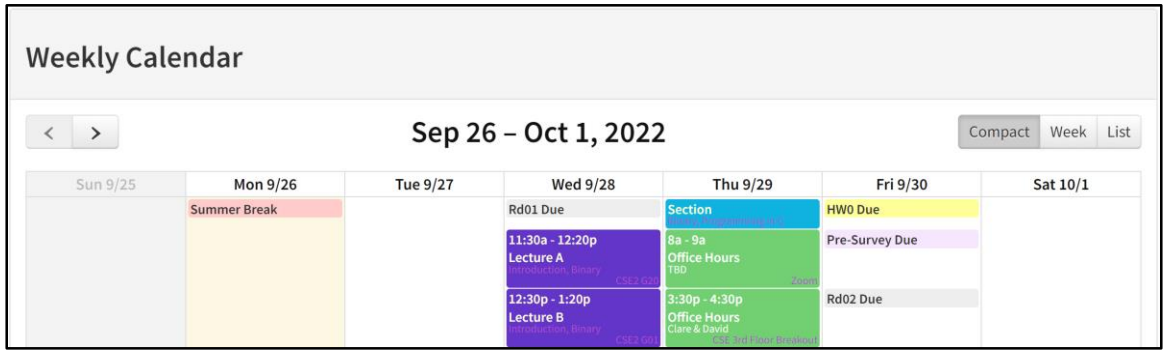
- Website: https://courses.cs.washington.edu/courses/cse351/23au/
  - Schedule, policies, materials, tutorials, assignment specs, etc.
- Ed Course: https://edstem.org/us/courses/41511
  - Discussion: announcements, ask and answer questions
  - Lessons: lessons, practice problems, homework
- Linked from website and Ed
  - Canvas: surveys, grade book, Zoom links
  - Gradescope: lab submissions, take-home exams
  - Panopto: lecture recordings

Grading

- **Lesson Problems:** 6% 
  - Can reveal solution after one attempt (completion)
- **Homework:** 20% total 
  - Unlimited submission attempts (autograded correctness)
- **Labs:** 40% total 
  - Last submission graded (correctness)
- **Exams:** Midterm (16%) and Final (16%) 
  - Take-home; individual, but some discussion permitted
- **EPA:** Effort, Participation, and Altruism (2%)

Support Hours

- Check Weekly Calendar on website for scheduled support hours:
  - In-person or virtual, but NOT hybrid
  - Zoom meeting links found in Zoom tab within Canvas



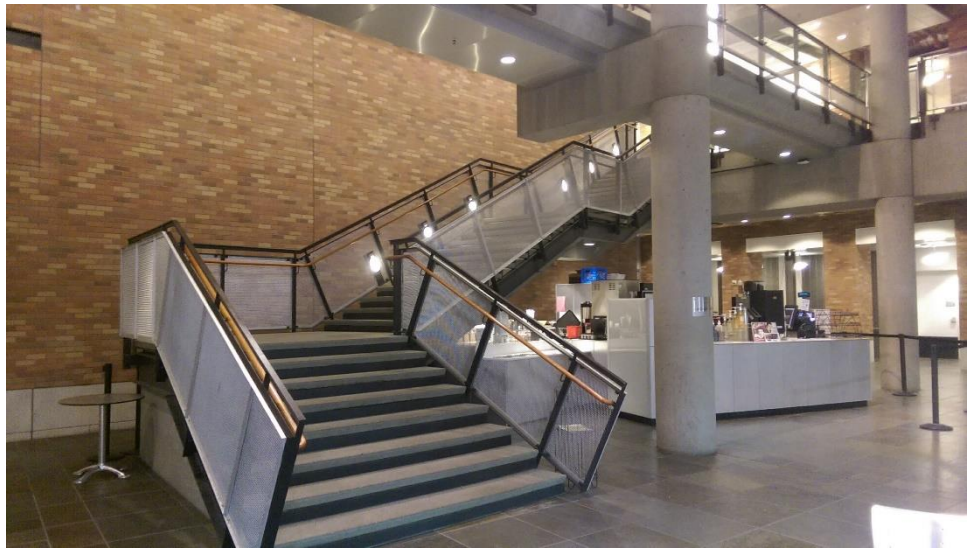
- *All* support hours will use a Google Sheets queue:
  - Fill out first 3 columns to enter queue:

Name(s)	Category	Description	Time Queued	Staff	Status
Example 1	Concept	Question about floating point encoding range.		Justin	Done
Example 2	Debugging	Lab 5: running into a segfault in mm_malloc after reaching end of the heap.		Justin	Done
Example 3	Spec	Lab 1a: confusion over within same block examples		Justin	Done
Example 4	Tools	GDB: how do I examine memory on the stack?		Justin	Done

- We encourage you to chat with other students if the TAs are busy!

In-Person Support Hours

- Allen 3rd floor breakout
  - Up the stairs in the CSE Atrium (Allen Center, not Gates)

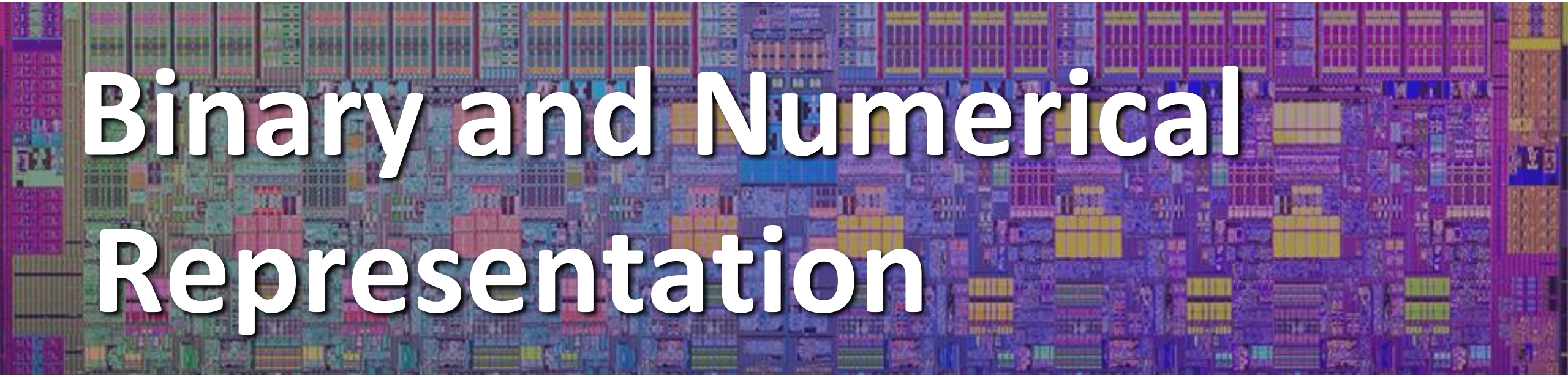


  - At the top of two flights, the open area with the whiteboard wall is the 3rd floor breakout!

To-Do List

- Admin
  - Explore/read the course website *thoroughly*, especially the syllabus
  - Check that you can access Ed Discussion & Lessons
  - **Get your machine set up to access the CSE Linux environment (attu or seaside) *as soon as possible***
  - Optionally, sign up for CSE 391: System and Software Tools

- Assignments
  - Pre-Course Survey and hw0 due Friday (9/29)
  - HW1 and Lab 0 due Monday (10/2)
  - Lessons quiz questions due 11:59 pm *after* the associated lecture



Binary and Numerical Representation

Lesson Summary (1/2)

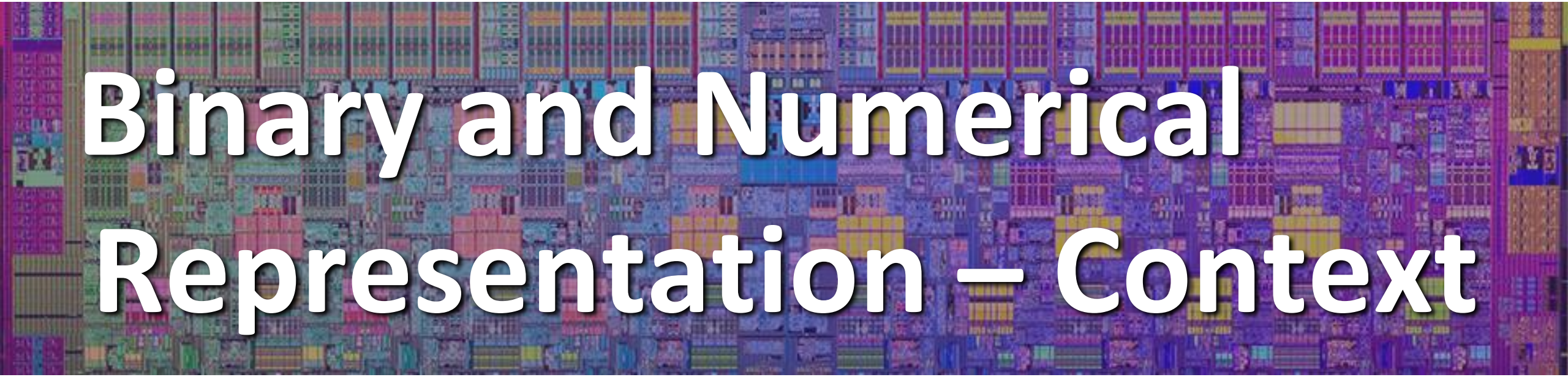
- Humans think about numbers in decimal; computers think about numbers in binary
  - Base conversion to go between them
  - Hexadecimal is more human-readable than binary
- All information on a computer is binary
- Binary encoding can represent *anything*!
  - Computer/program needs to know how to interpret the bits
  - Encodings aren't "neutral"; priorities are baked in

Lesson Summary (2/2)

- Terminology:
  - numeral, digit, base, symbol, digit position, leading zeros
  - binary, bit, nibble, byte, hexadecimal
  - numerical representation, encoding scheme

- Learning Objectives:
  - Convert between binary, decimal, and hexadecimal number representations.
  - Given an encoding scheme, decode and encode binary to/from its intended representation.
  - Identify limitations of given encoding schemes.

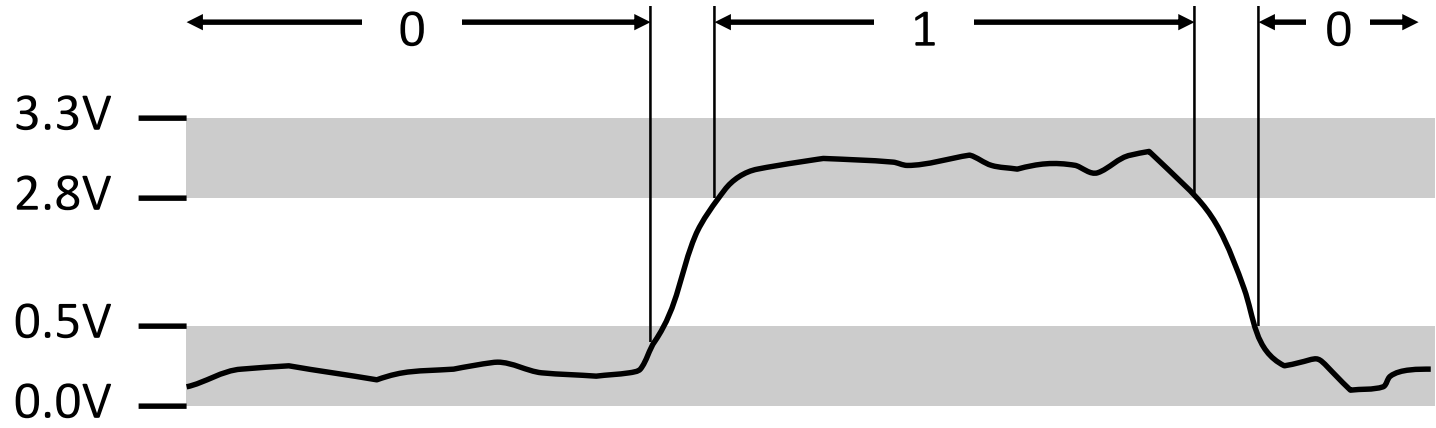
- What lingering questions do you have from the lesson?



Binary and Numerical Representation – Context

Why Base 2?

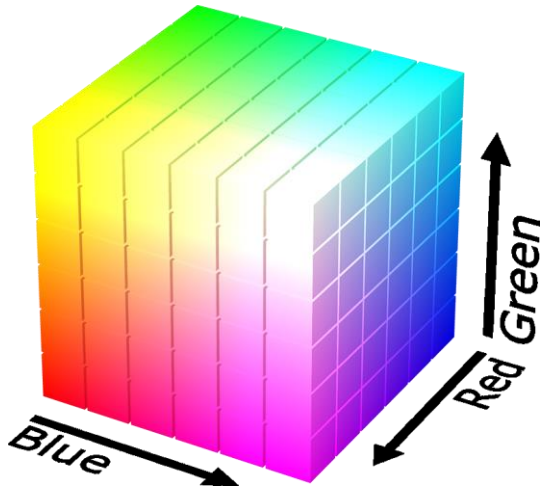
- Electronic implementation
  - Easy to store with bi-stable elements
  - Reliably transmitted on noisy and inaccurate wires

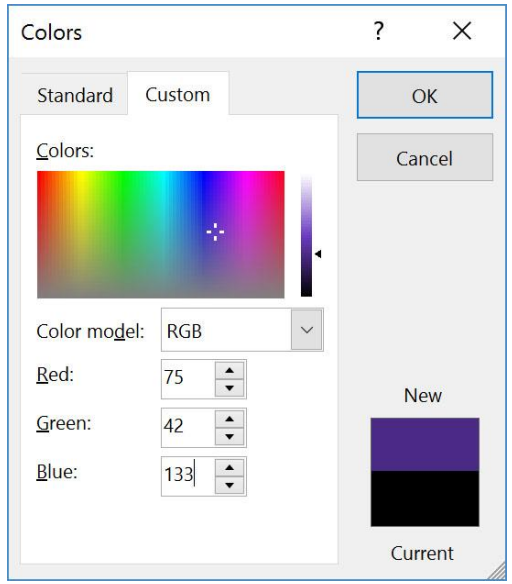


- Other bases possible, but not yet viable:
  - DNA data storage (base 4: A, C, G, T) is hot @UW
  - Quantum computing

Binary Encoding – Colors

- RGB – Red, Green, Blue
  - Additive color model (light): byte (8 bits) for each color
  - Commonly seen in hex (in HTML, photo editing, etc.)
  - Examples: **Blue**→0x0000FF, **Gold**→0xFFD700, **White**→0xFFFFFF, **Deep Pink**→0xFF1493





Binary Encoding – Characters/Text

- ASCII Encoding (www.asciitable.com)
  - American Standard Code for Information Interchange

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	14	100;	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	1	01;	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	46	102;	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	10	67	47	103;	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	72		4	68	150	04;	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	11	I		1	69	1	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4	12	J		1	6	1	j	j
11	B	013	VT (vertical tab)	43	2B	053	&#43		7	4B	1			107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	0	#44		76	4	1	#76;	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	0	15	-	7	4	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	05		.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	#4	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escap			060	#48;	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (d ice co	4		061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	1	24	ev t trol 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	5	ega ve acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	0	N nchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	02	TB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

What's Missing?

Binary Encoding – Characters/Text

- ASCII Encoding (www.asciitable.com)
  - *American* Standard Code for Information Interchange
- Created in 1963
  - Memory was expensive, 32KB in brand new machines
  - *Economic incentive* to use fewer bits for encoding
- **Design Goals:**
  - Represent everything on an *American* typewriter as *efficiently* as possible
  - Organize similar characters together
    - Numbers, uppercase, lowercase, then other stuff

Binary Encoding – Unicode & Emoji

- Unicode Standard is managed by the Unicode Consortium
  - "Universal language" that uses 1-4 bytes to represent a much larger range of characters/languages, including emoji
  - Adds new emojis every year, though adoption often lags: 🫨
    - https://emojipedia.org/new/

- Emojipedia demo: http://www.emojipedia.org
  - Desktop Computer: 🖥
  - Code points: U+1F5A5, U+FE0F
  - Display: 

Binary Encoding – Files and Programs

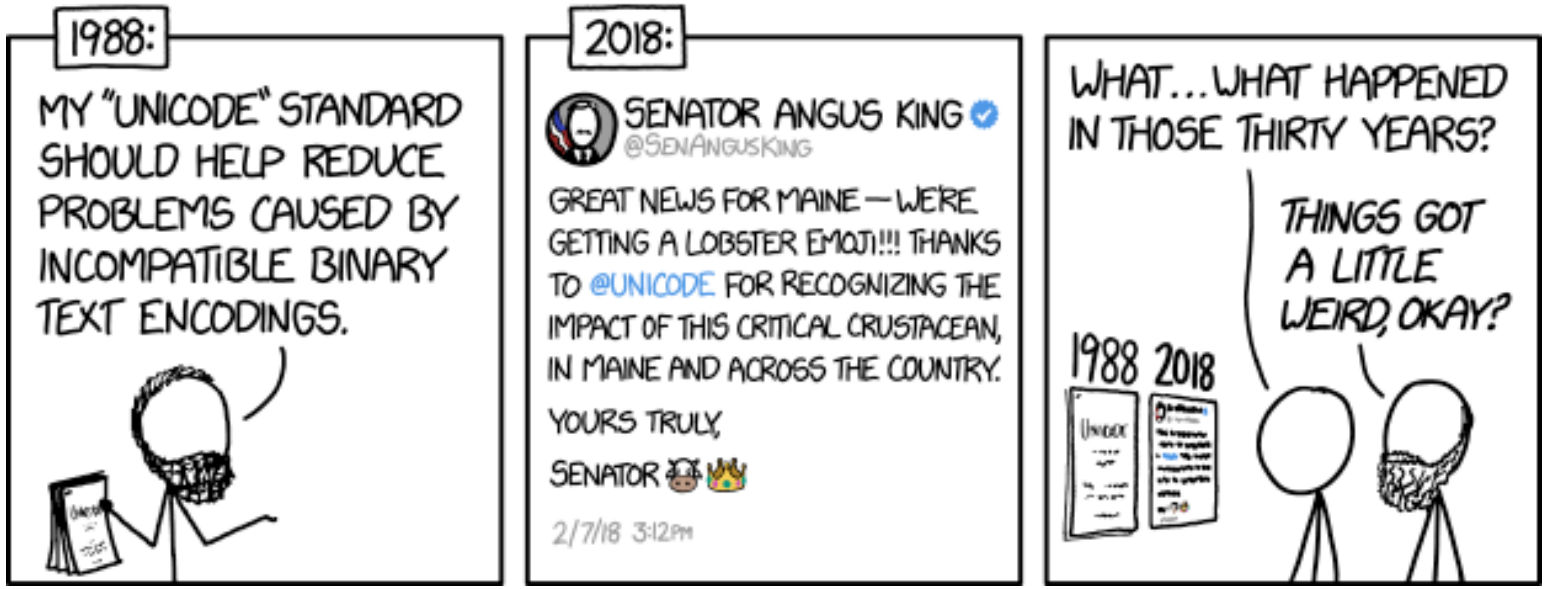
- At the lowest level, all digital data is stored as bits!

- Layers of abstraction keep everything comprehensible
  - Data/files are groups of bits interpreted by program
  - Program is actually groups of bits being interpreted by your CPU

- Computer Memory Demo
  - Linux tool: `xxd`

Discussion Question

- Discuss the following question(s) in groups of 3-4 students
  - I will call on a few groups afterwards so please be prepared to share out
  - Be respectful of others' opinions and experiences

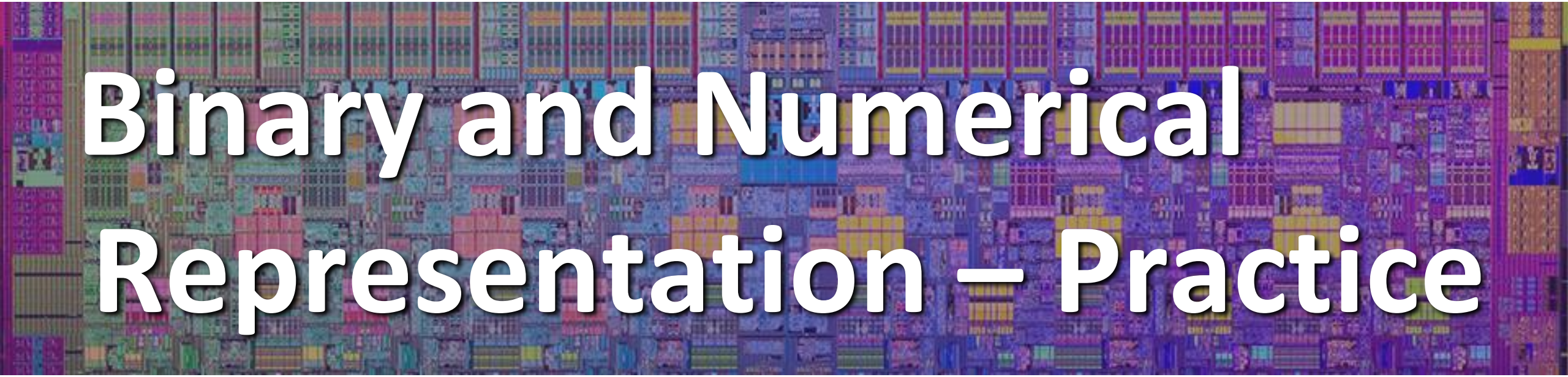


http://xkcd.com/1953/

Discussion Question

- Discuss the following question(s) in groups of 3-4 students
  - I will call on a few groups afterwards so please be prepared to share out
  - Be respectful of others' opinions and experiences

- The Unicode Consortium publicly solicits proposals from the public for new emoji to add to future standards
  - What do you think some of the decision factors are (or should be) in how many and which ones to add?
  - Voting is done by a combination of paid members consisting of companies, institutions, and individuals – how do you feel about who has control and how they gained that control?
    - https://home.unicode.org/membership/members/



Binary and Numerical Representation – Practice

Group Work Time

- During this time, you are encouraged to work on the following:
  1) If desired, continue your discussion
  2) Work on the lesson problems (solutions at the end of class)
  3) Work on the homework problems

- Resources:
  - You can revisit the lesson material
  - Work together in groups and help each other out
  - Course staff will circle around to provide support

Practice Problems

- What is the *decimal value* of the numeral 107_8?
  - A. 71
  - B. 87
  - C. 107
  - D. 568
- Represent 0b100110110101101 in hex.
- What is the decimal number 108 in hex?
  - A. 0x6C
  - B. 0xA8
  - C. 0x108
  - D. 0x612
- Represent 0x3C9 in binary.