EE214/PHYS220: Quantum Computing

Hung-Wei Tseng

EE214/PHYS220: Let’s say something!

What’s your name?

What you interested in research

Why’re you interested in EE214?

3:00

How do modern computers compute?

Boolean Algebra

Boolean algebra (disambiguation)

- Boolean algebra — George Boole, 1815—1864
  - Introduced binary variables
  - Introduced the three fundamental logic operations: AND, OR, and NOT
  - Extended to abstract algebra with set operations: intersect, union, complement
- Switching algebra — Claude Shannon, 1916—2001
  - Wrote his thesis demonstrating that electrical applications of **Boolean algebra** could construct any logical numerical relationship
  - Disposal of the abstract mathematical apparatus, casting **switching algebra** as the **two-element Boolean algebra**.
  - We now use switching algebra and boolean algebra interchangeably in EE, but not doing that if you're interacting with a mathematician.

Basic Boolean Algebra Concepts

- {0, 1}: The only two possible values in inputs/outputs
- Basic operators
  - AND (•) — a • b
    - returns 1 only if both a **and** b are 1s
    - otherwise returns 0
  - OR (+) — a + b
    - returns 1 if a **or** b is 1
    - returns 0 if none of them are 1s
  - NOT (’) — a’
    - returns 0 if a is 1
    - returns 1 if a is 0

Truth tables

- A table sets out the functional values of logical expressions on each of their functional arguments, that is, for each combination of values taken by their logical variables

AND

Input		Output
A	B	
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

OR

Input		Output
A	B	
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

NOT

Input	Output
A	
0	1
0	1
1	0
1	0

Derived Boolean operators

- NAND — $(a \cdot b)'$
- NOR — $(a + b)'$
- XOR — $(a + b) \cdot (a' + b')$ or $ab' + a'b$
- XNOR — $(a + b') \cdot (a' + b)$ or $ab + a'b'$

NAND

Input		Output
A	B	
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**

NOR

Input		Output
A	B	
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**0**

XOR

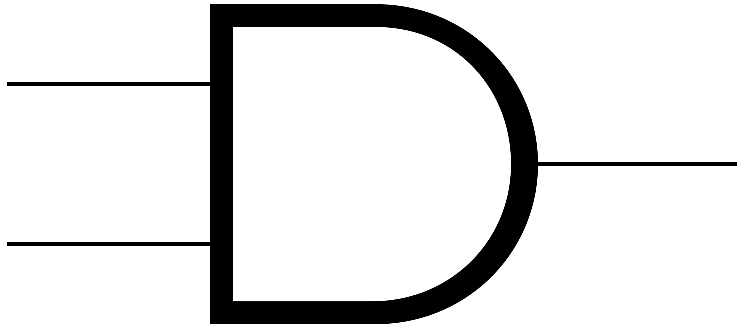
Input		Output
A	B	
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

XNOR

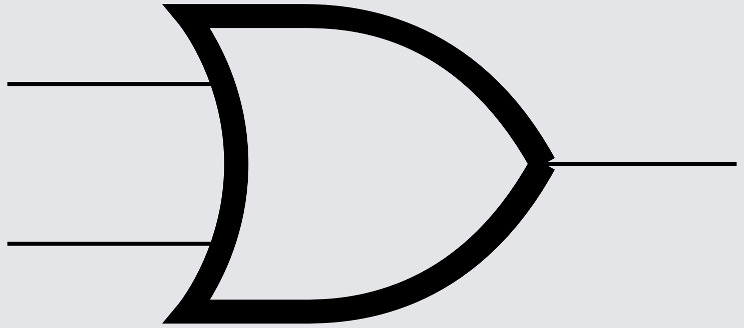
Input		Output
A	B	
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**1**

Boolean operators their circuit "gate" symbols

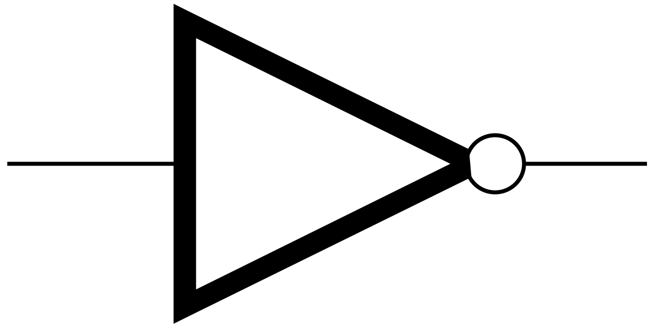
AND

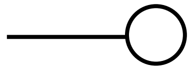


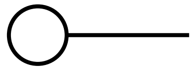
OR



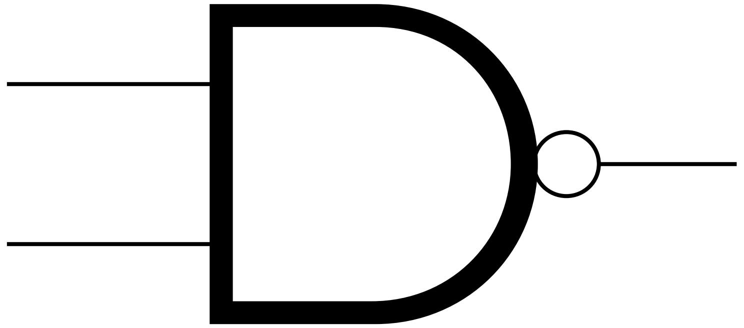
NOT



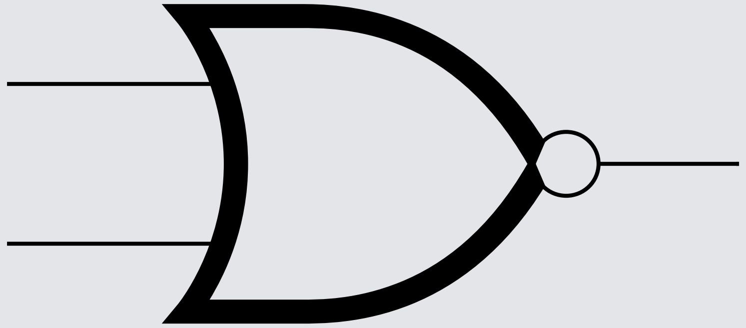
 represents where we take a compliment value on an input

 represents where we take a compliment value on an output

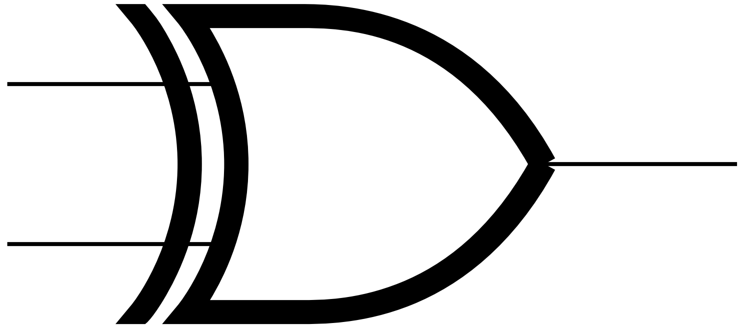
NAND



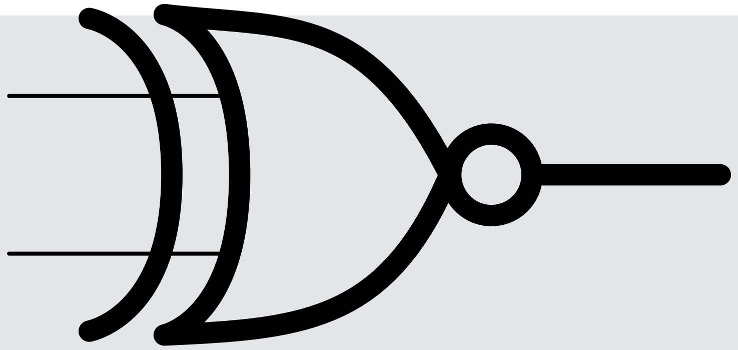
NOR



XOR



NXOR



The basic idea of a number system

- Each position represents a quantity; symbol in position means how many of that quantity
  - Decimal (base 10)
    - Ten symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
    - More than 9: next position
    - Each position is incremented by power of 10
  - Binary (base 2)
    - Two symbols: 0, 1
    - More than 1: next position
    - Each position is incremented by power of 2

$$\begin{array}{ccccc} 10^2 & & 10^1 & & 10^0 \\ \times & & \times & & \times \\ \mathbf{3} & + & \mathbf{2} & + & \mathbf{1} \end{array} \begin{array}{l} =300 \\ +20 \\ +1 \\ =321 \end{array}$$

$$\begin{array}{ccccccc} 2^3 & & 2^2 & & 2^1 & & 2^0 \\ \times & & \times & & \times & & \times \\ \mathbf{1} & + & \mathbf{0} & + & \mathbf{0} & + & \mathbf{1} \end{array} \begin{array}{l} =1 \times 2^3 \\ +1 \times 2^0 \\ =1 \times 8 \\ +1 \times 1 \\ =9 \end{array}$$

Can this work?

- 3 + 2 = 5

$$\begin{array}{r} \,1 \\ 0\,0\,1\,1 \\ +\,0\,0\,1\,0 \\ \hline 0\,1\,0\,1 \end{array}$$

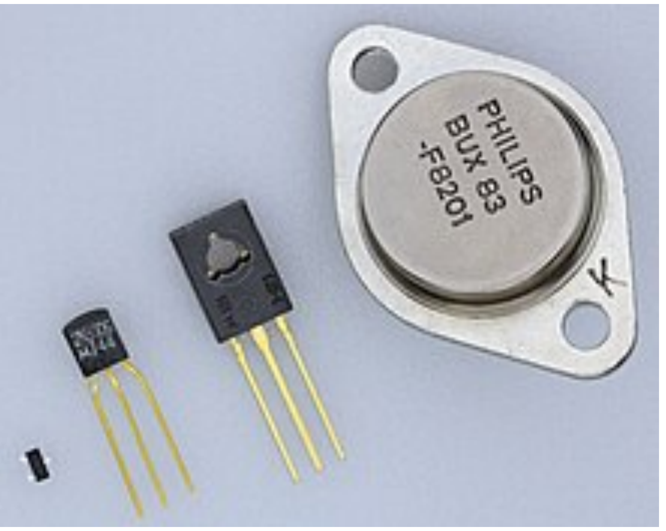
- 3 + (-2) = 1

$$\begin{array}{r} \,1 \\ 0\,0\,1\,1 \\ +\,1\,0\,1\,0 \\ \hline 1\,1\,0\,1 \end{array} = -5 \text{ (Not 1)}$$

Doesn't work well and you need a separate procedure to deal with negative numbers!

Implementing Boolean Functions

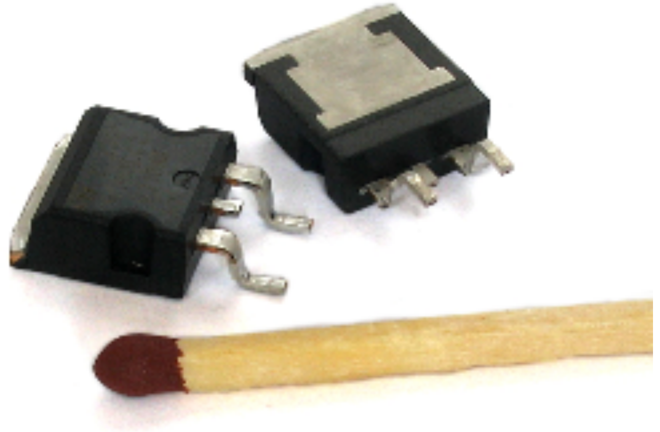
Types of transistors



BJT
(Bipolar junction transistor), 1947

JFET
(junction field-effect transistor), 1953

IGBT
(Insulated-gate bipolar transistor), 1959



MOSFET (the metal–oxide–semiconductor field-effect transistor), 1960

Moore's Law[1]

Present and future

By integrated electronics, I mean technologies which are referred to tronics today as well as any additi result in electronics functions suppli as irreducible units. These technolo i to miniaturize electronics equipment creasingly complex electronic functi space with minimum weight. Sever evolved, including microassembly individual components, thin-film semiconductor integrated circuits.

ICs are increasingly p

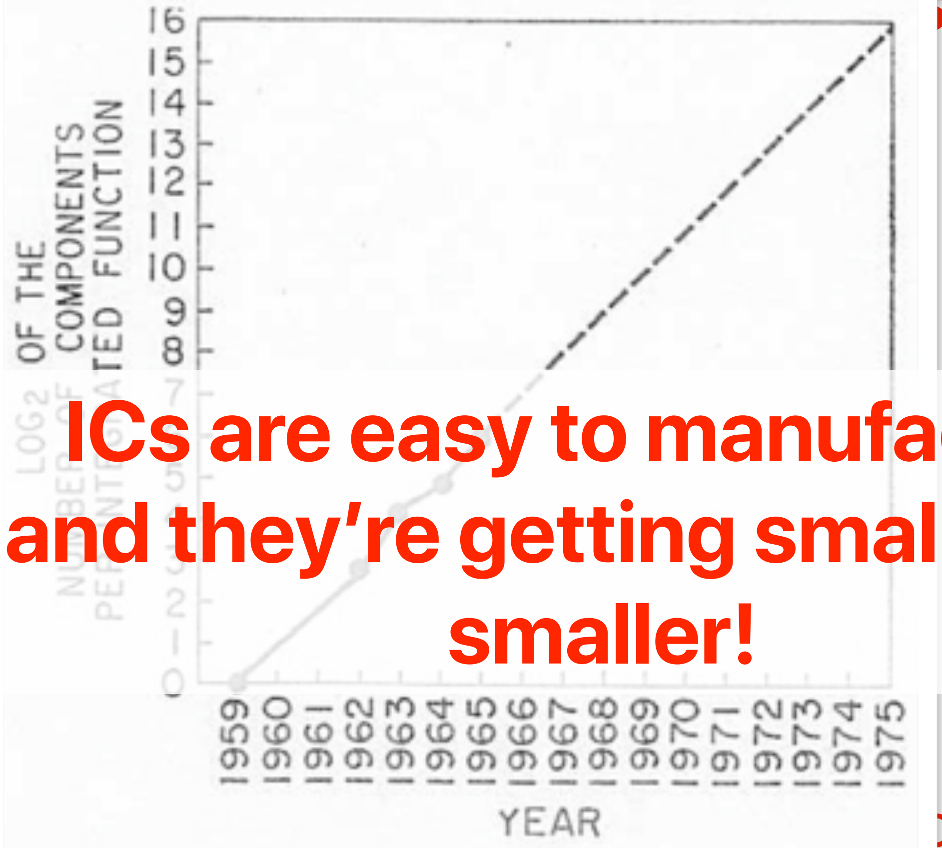
The establishment

Integrated electronics is established or new and v vable llo, fo he reli that ilure as the

stablish

Increasing the yield

There is no fundamental obstacle to achieving device yields of 100%. At present, packaging costs so far exceed the cost of the semiconductor structure itself that there is no incentive to improve yields, but they can be raised as high as is economically justified. No barrier exists comparable to the thermodynamic equilibrium considerations



ICs are easy to manufacture and they're getting smaller and smaller!

Two-mil squares

With the dimensional tolerances already being employed in integrated circuits, isolated high-performance transistors can be built on centers two thousandths of an inch apart. Such a two-mil square can also contain several kilohms of resistance o

ICs are small

Reliability coun

In almost e demonstrated h level of production—low compared to that of discrete components—it offers reduced systems cost, and in many systems improved performance has been realized.

ICs are more reliable

Linear circuitry

Integration will not change linear systems as radically as digital systems. Still, a considerable degree of integration will be achieved with linear circuits. The lack of large-value capacitors and to integrated electronics in the linear area.

ICs are widely applicable

Heat problem

Will it be possible to remove the heat generated by tens of thousands of components in a single silicon chip?

Heat is a solvable issue

Moore's Law is the most important driver for historic CPU performance gains

Day of reckoning

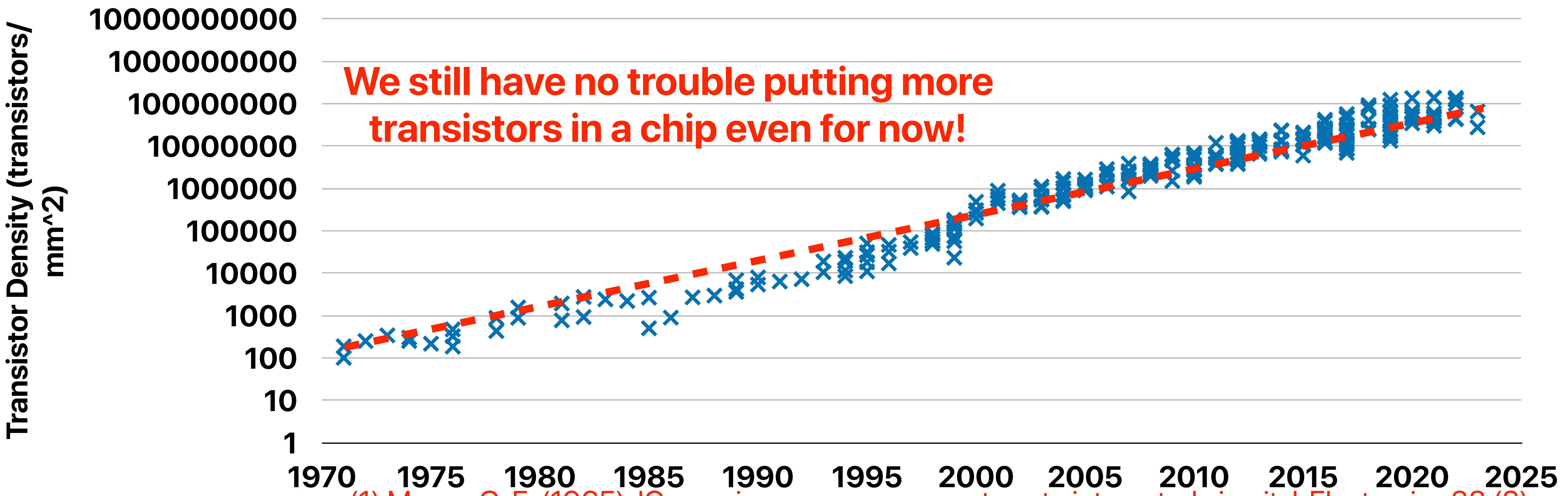
Clearly, we will be able to build such component-crammed equipment. Next, we ask under what circumstances we should do it. The total cost of making a particular system function must be minimized. To do so, we could amortize the engineering over several identical items, or evolve flexible techniques for the engineering of large functions so that no disproportionate expense need be borne by a particular array. Perhaps newly devised design automation procedures could translate from any special engineering.

Designing ICs can be easy

(1) Mo mponents onto integrated circuits', Electronics 38 (8) .

Moore's Law[1]

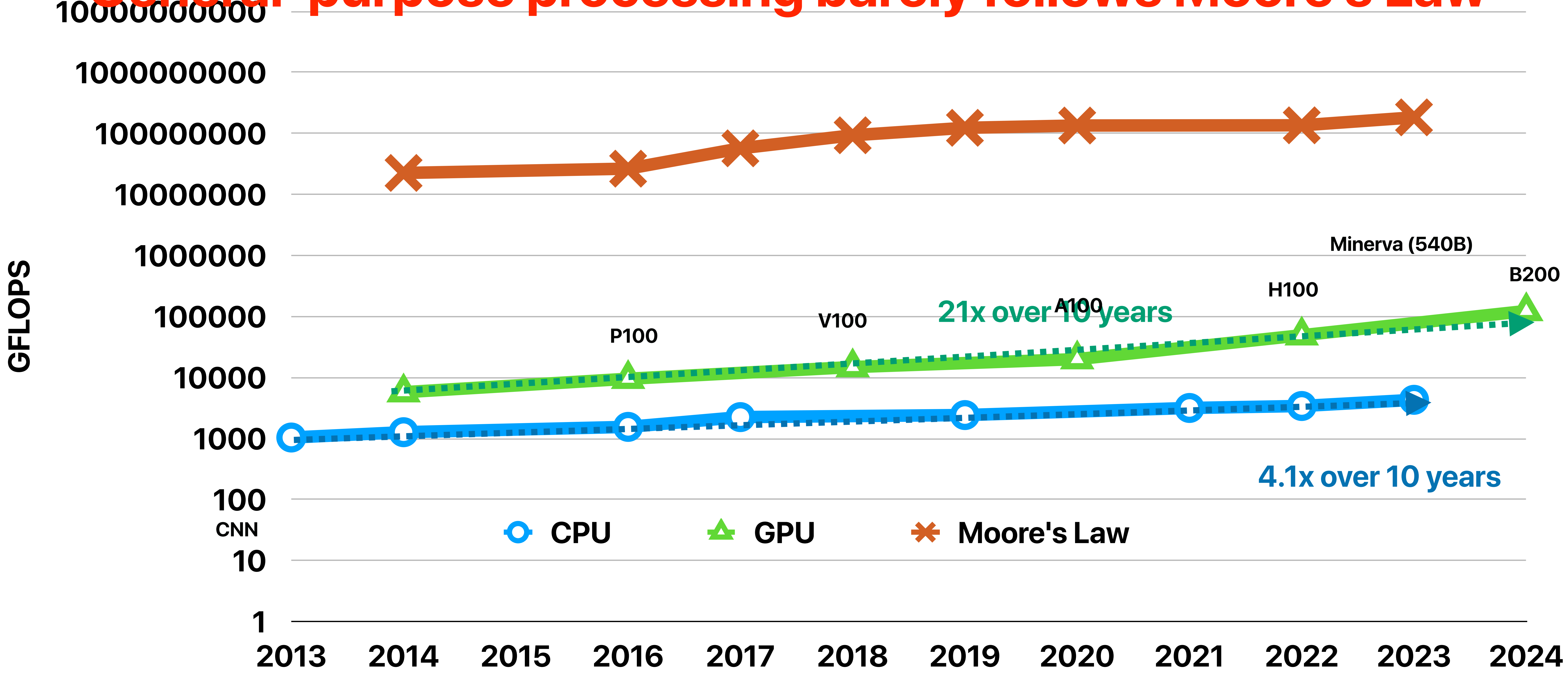
- The number of transistors we can build in a fixed area of silicon doubles every 12 ~ 24 months.
- Moore's Law "was" the most important driver for historic CPU performance gains



(1) Moore, G. E. (1965), 'Cramming more components onto integrated circuits', Electronics 38 (8) .

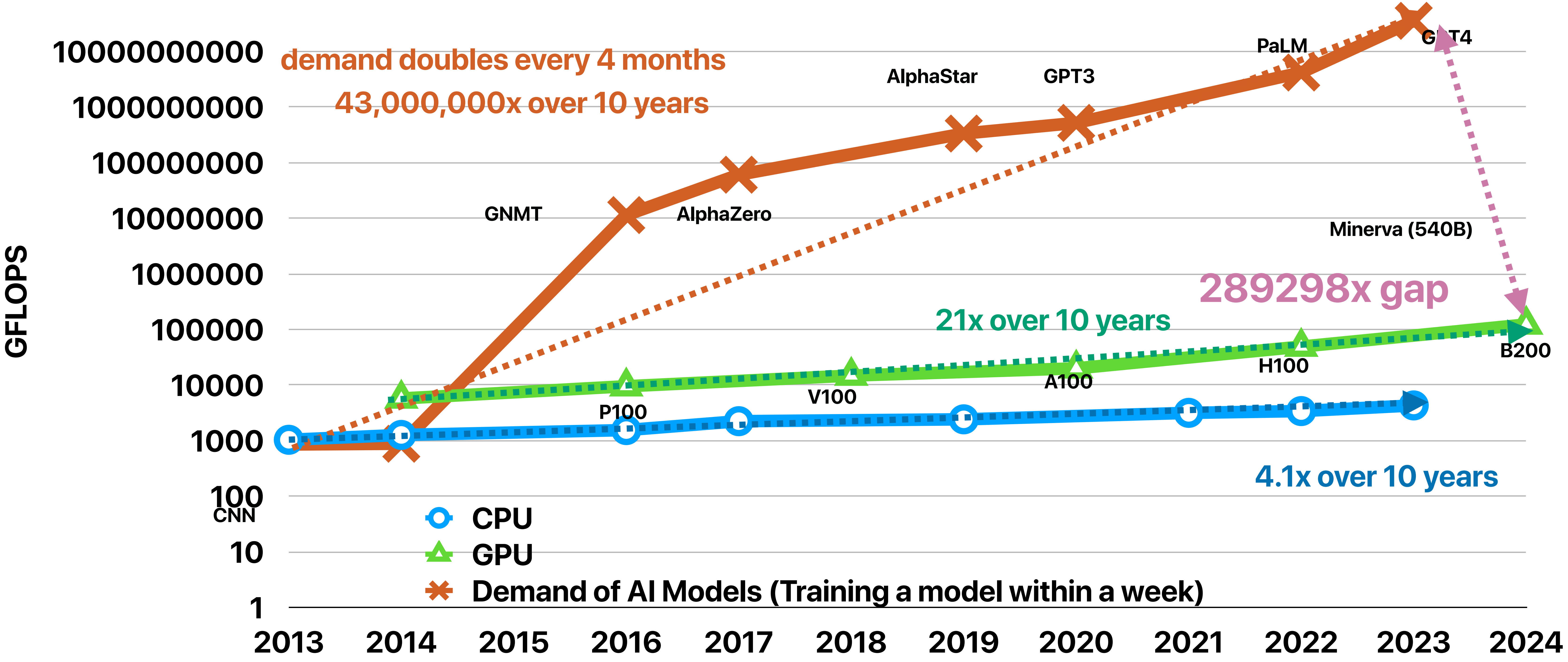
Plot based on https://en.wikipedia.org/wiki/Transistor_count by Hung-Wei Tseng

General-purpose processing barely follows Moore's Law



https://ourworldindata.org/grapher/artificial-intelligence-training-computation

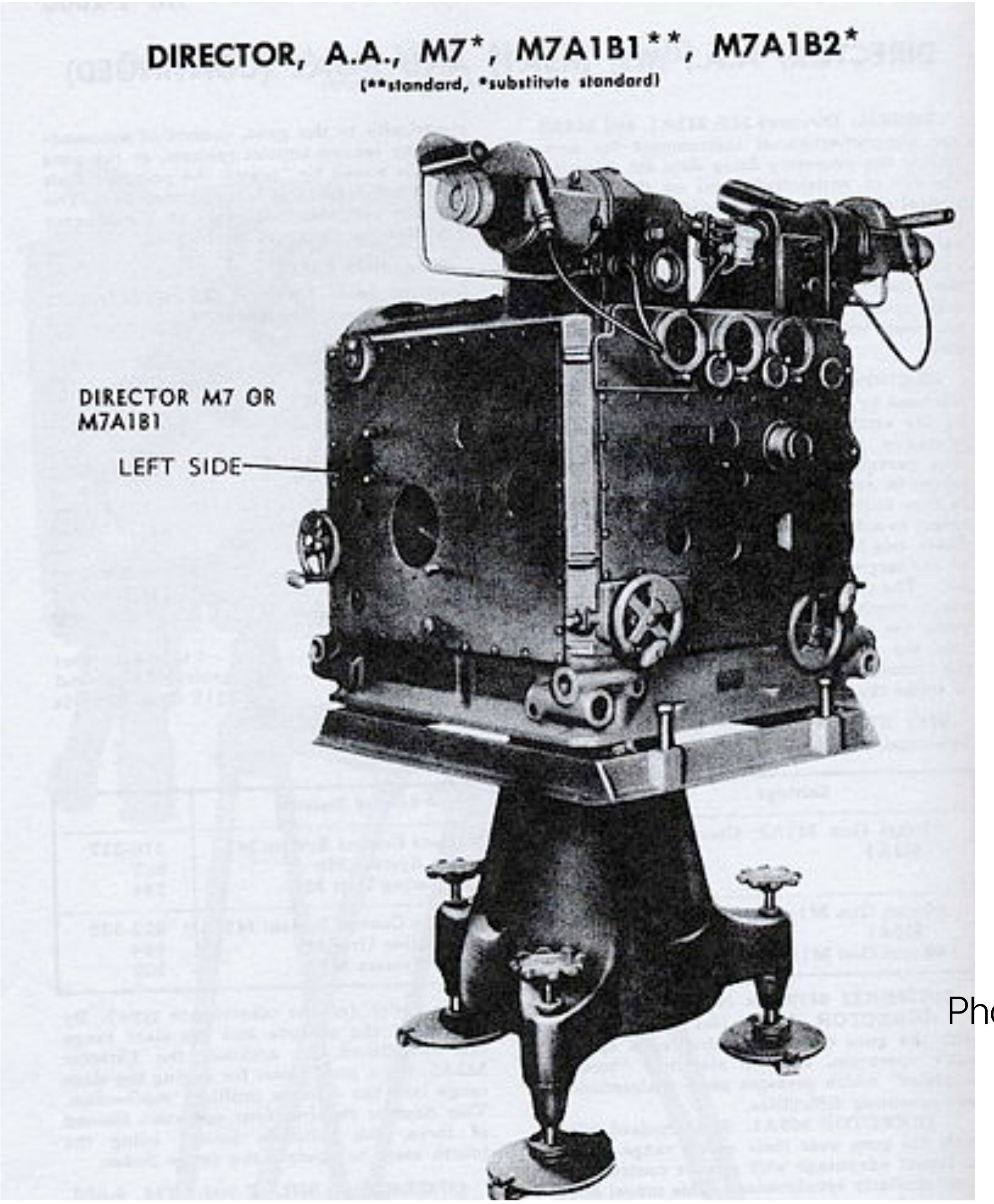
But digital computers cannot address the demand of modern applications



https://ourworldindata.org/grapher/artificial-intelligence-training-computation

What if we rethink the way of computing?

Computers were not “digital”



MNIAC — 1949

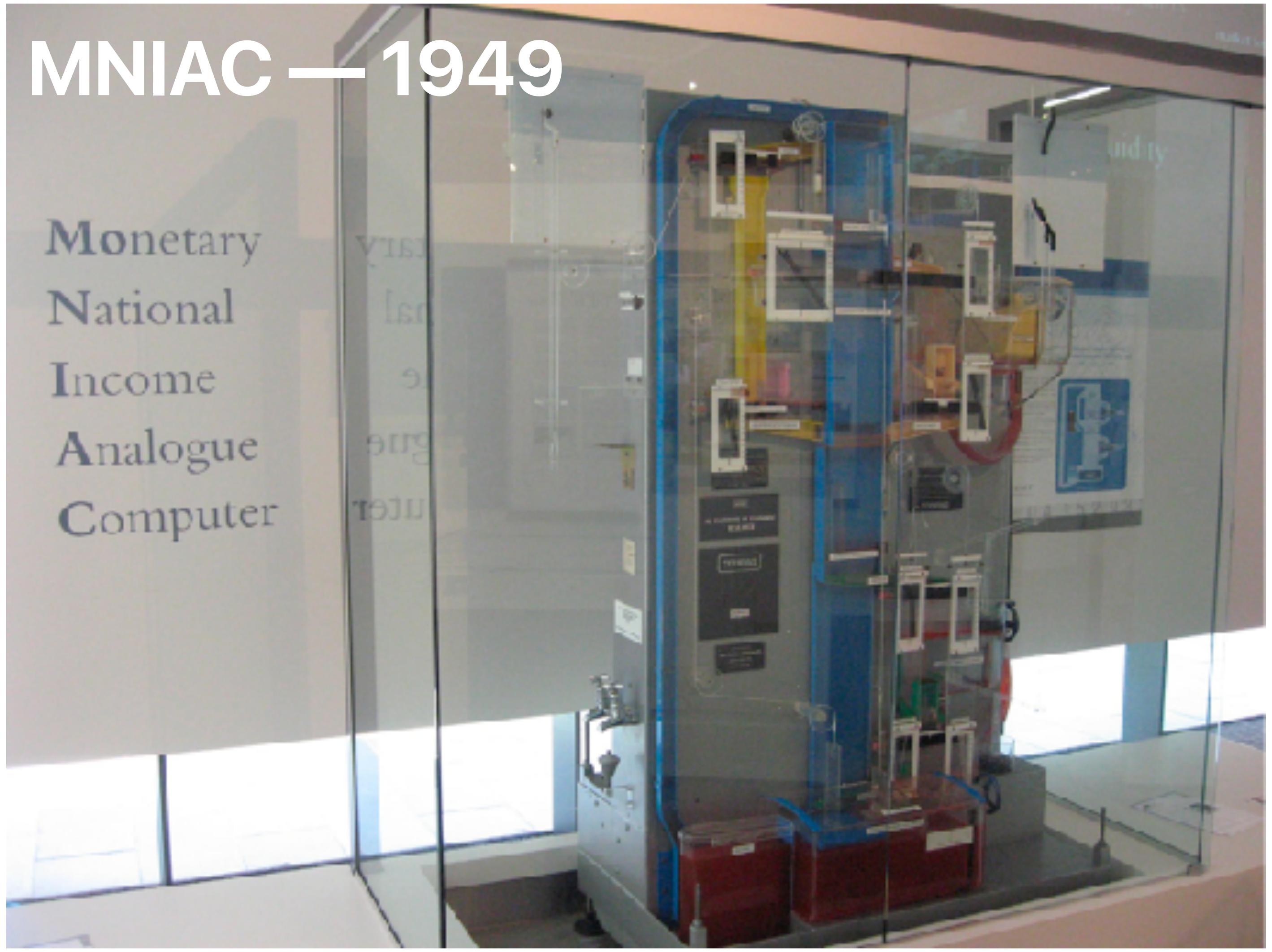


Photo Credit By Kaihsu Tai, https://commons.wikimedia.org/w/index.php?curid=3956307

Fermiac — 1947

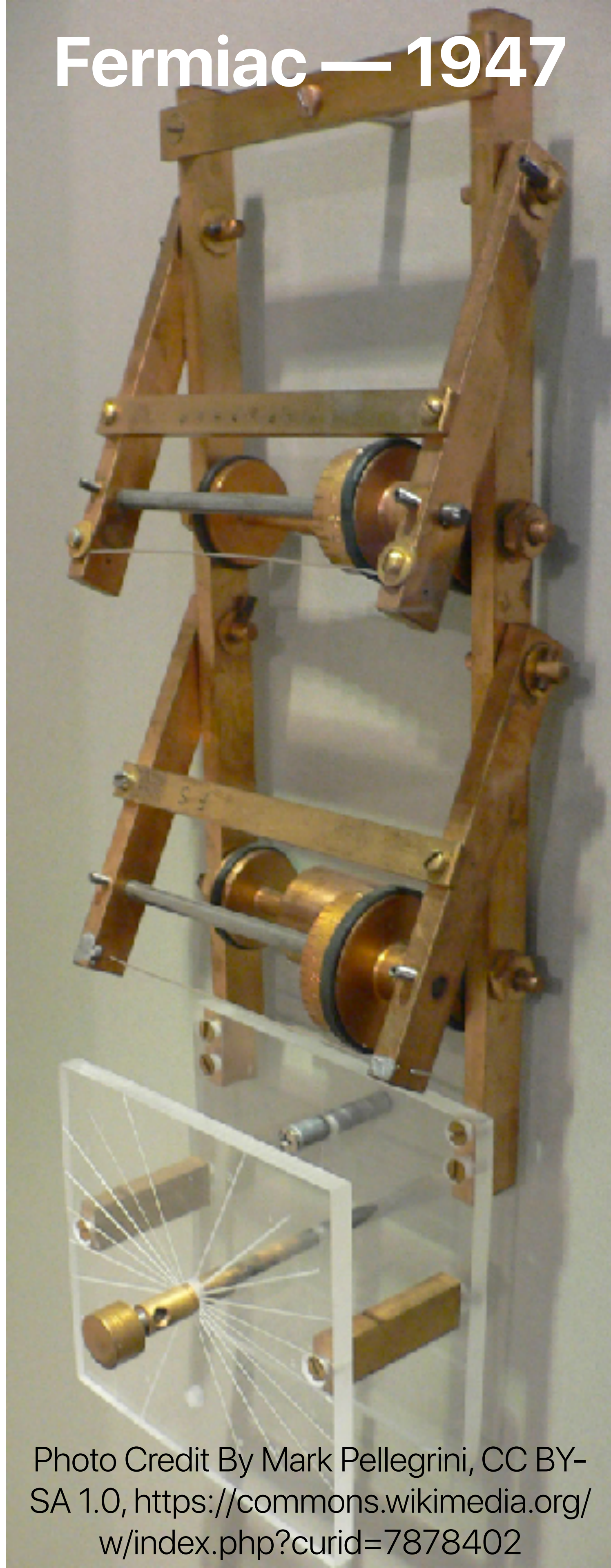


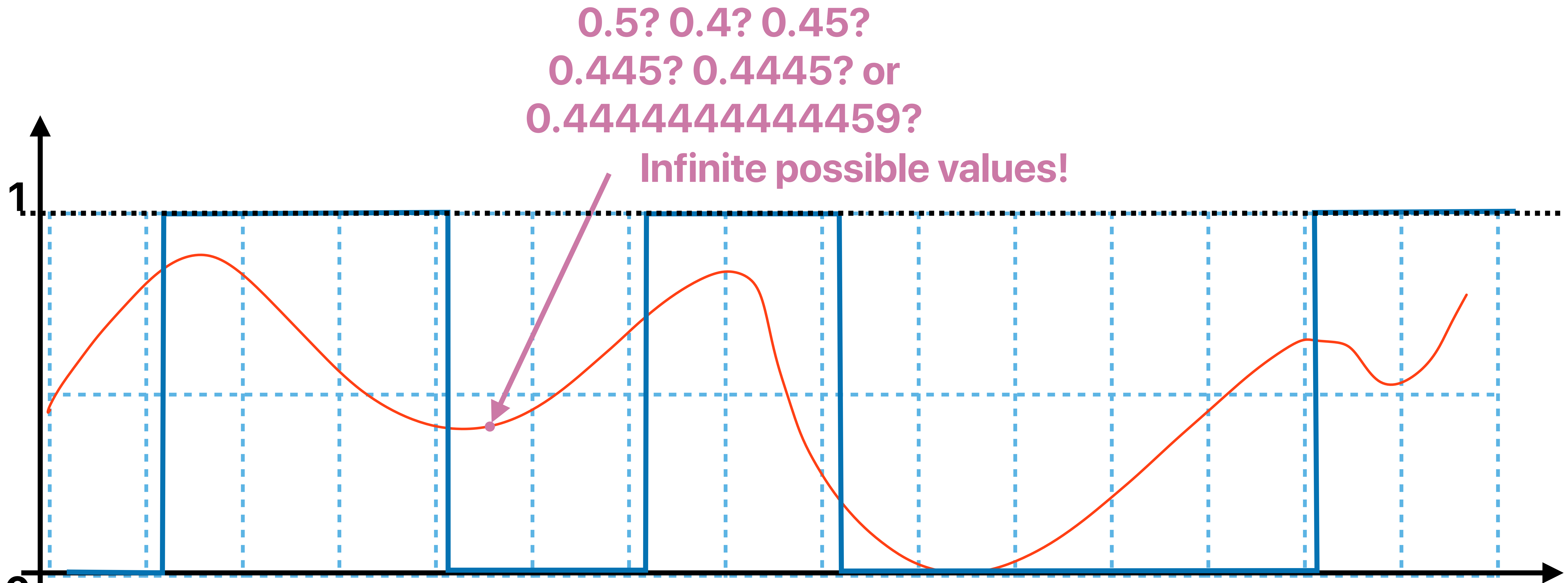
Photo Credit By Mark Pellegrini, CC BY-SA 1.0, https://commons.wikimedia.org/w/index.php?curid=7878402

Analog data storage

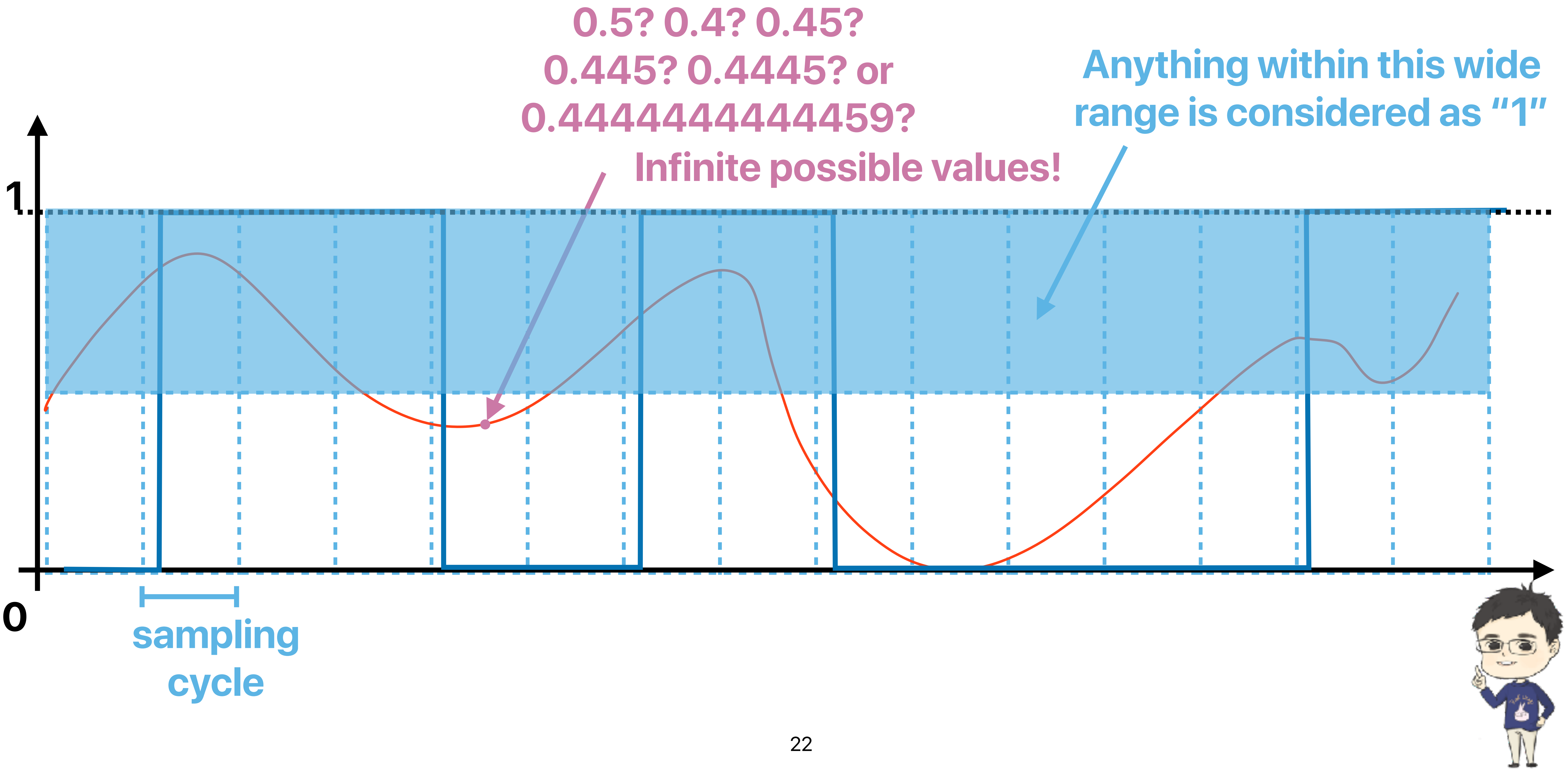
Mike
@Doranimated

The CDC has developed a simple test to determine if you are at risk of developing complications from coronavirus. Please examine the following two items. Do you understand the connection between them? If your answer is yes, you are in the at-risk category. Please self-isolate.

Analog v.s. digital signals



Analog v.s. digital signals



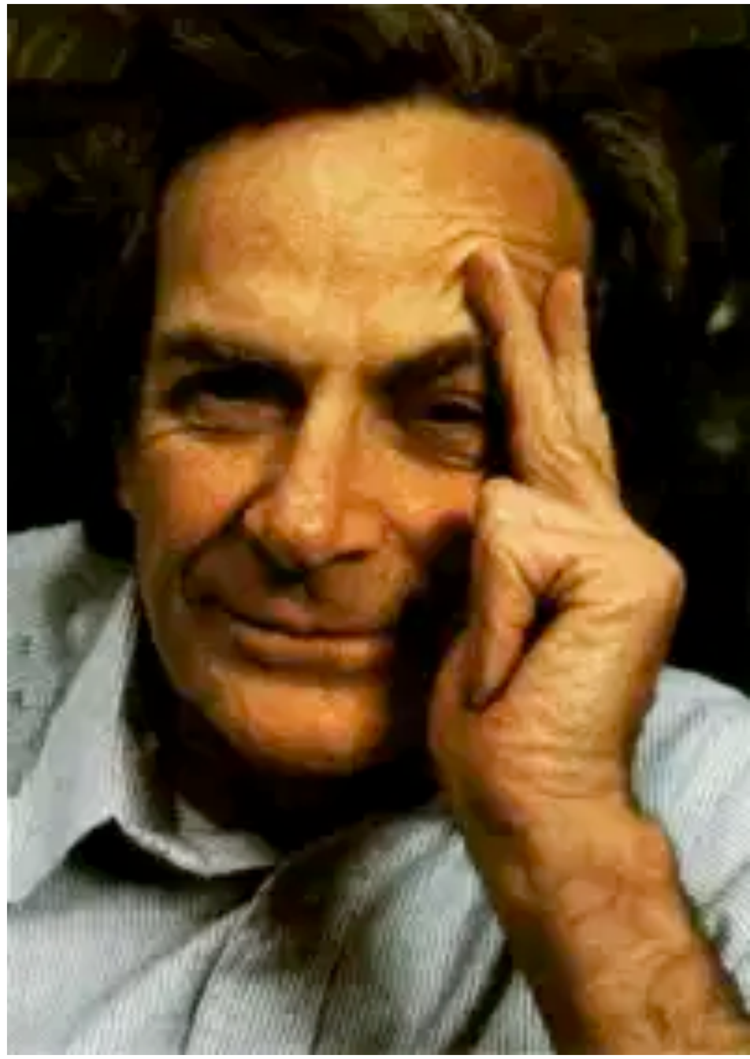
Why are we going digital?

- The cost of building systems with the same functionality is lower by using digital computers.
- Digital signals are less fragile to noise and defective/low-quality components.
- Digital data are easier to store.

Quantum Computing

Brief History of Quantum Computing

- 1959 — Richard Feynman states the possibility of using quantum effects for computation
- 1980 — Paul Benioff published paper on quantum Turing machines
- 1981 — Feynman urged the world to build a quantum computer. "Nature isn't classical...and if you want to make a simulation of nature, you'd better make it quantum mechanical…"
- 1993 — an international group of six scientists, showed that perfect quantum teleportation is possible
- 1994
  — Dan Simon showed another computational problem that a quantum computer has an exponential advantage over any classical computers.
  —Peter Shor discovered a quantum algorithm, which allows a quantum computer to factor large integers exponentially much faster than the best known classical algorithm. Shor's can theoretically break many of the Public-key cryptography systems in use today



Brief History of Quantum Computing

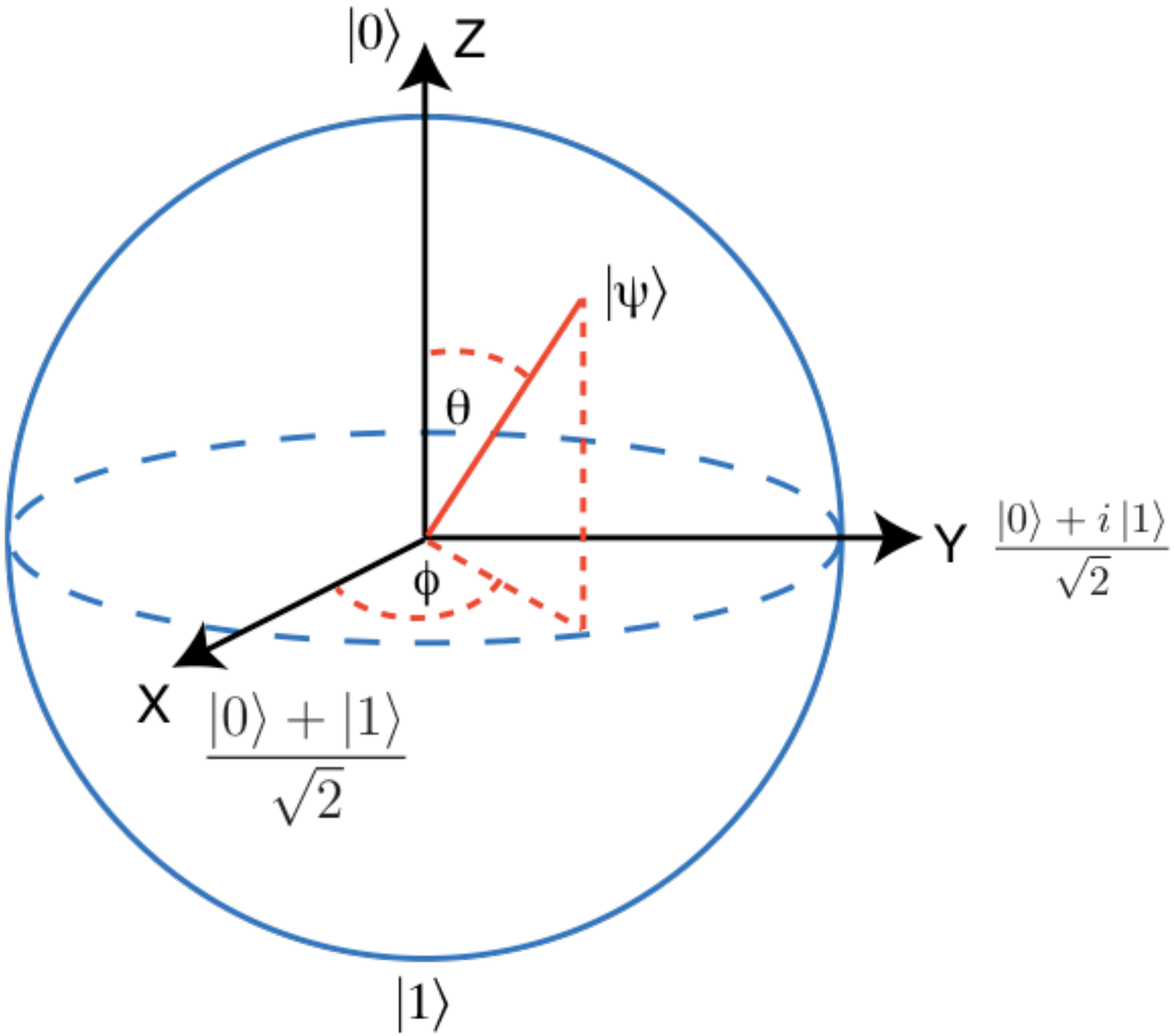
- 2009 — Yale created first solid-state quantum processor, a 2-qubit superconducting chip
- 2011
  — scientists from Australia and Japan made a breakthrough in quantum teleportation, successfully transferring quantum data with full transmission integrity
  — D-Wave announced first commercial quantum annealer
- 2012 — the first quantum teleportation from one macroscopic object to another was reported by scientists at the University of Science and Technology of China
- 2013 — Google announced that it was launching the Quantum AI Lab
- 2014 — Edward Snowden showed the NSA is running a $79.7 million research program titled "Penetrating Hard Targets", to develop a quantum computer capable of breaking vulnerable encryption
- 2015 — NASA publicly displayed the world's first fully operational quantum computer, D-Wave Systems
- 2017 — IBM Research scientists successfully "broke the 49-qubit simulation barrier”
- 2018 — IBM, Intel, and Google each reported testing quantum processors containing 50, 49, and 72 qubits
- 2019 — Google announced it had achieved quantum supremacy - marking a huge milestone in the advancement of practical quantum computing

Quantum Computer

- A computer exploits its internal states through special quantum mechanical properties such as superposition and entanglement

A qubit state

θ and ϕ can be arbitrarily small! — some flavor of analog computing



An example of quantum vs conventional computing

Factorization

$$60 = 2^2 \times 3 \times 5$$

Classical computation model

- The state-of-the-art General Number Field Sieve (GNFS) algorithm will require $O(e^{c(logn)^{\frac{1}{3}}(log(logn))^{\frac{2}{3}}})$ — exponential in the length of the given number n.
- Factor a 232-digit (768-bit) RSA key using hundreds of machines in the span of two years in 2010.

Quantum computing

- Shor's algorithm can solve the problem with $O(\sqrt{N})$ steps

Will quantum computing release the conventional computing?

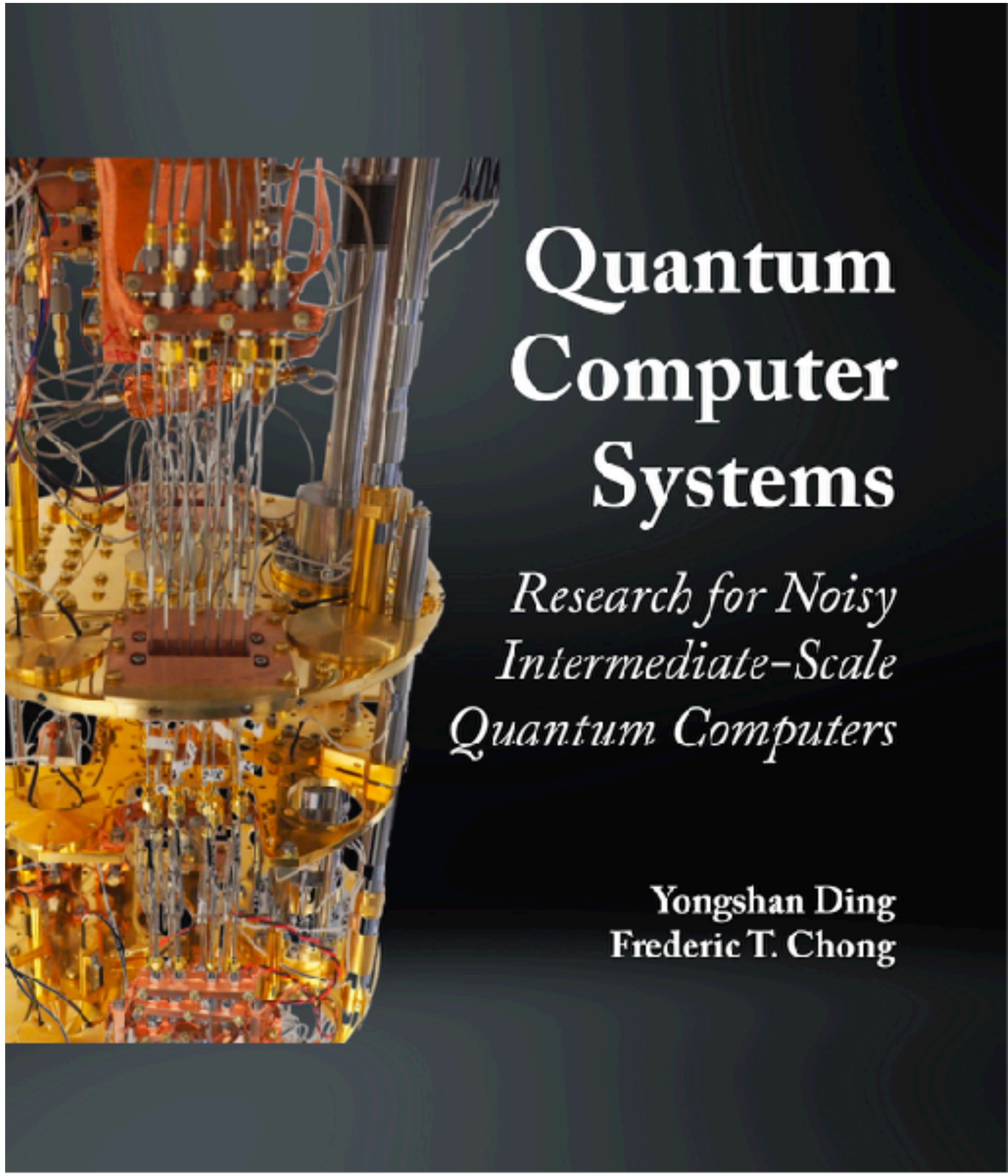
To just a certain degree because...

- Classical computers are better at storing data, while quantum computers are better at identifying the most likely answers in large data sets
- Quantum computers are expensive and require a lot of maintenance, making them less practical for tasks that classical computers can do more efficiently
- Quantum computers need to be kept at very low temperatures, close to absolute zero, which would be impractical for most households and businesses
- Modern quantum computers are noisy and inaccurate.

Learning experience

What the class is going to cover?

- Quantum Computing for Computer Architects (2nd Edition). Tzvetan S. Metodi, Arvin I.Faruque and Frederic T. Chong. Synthesis Lectures on Computer Architecture, 2006. (https://doi.org/10.1007/978-3-031-01718-6)
- Quantum Computer Systems: Research for Noisy Intermediate-Scale Quantum Computers. Yongshan Ding and Frederic T. Chong. Synthesis Lectures on Computer Architecture, 2020. (https://doi.org/10.2200/S01014ED1V01Y202005CAC051)
- Qiskit https://learning.quantum.ibm.com/

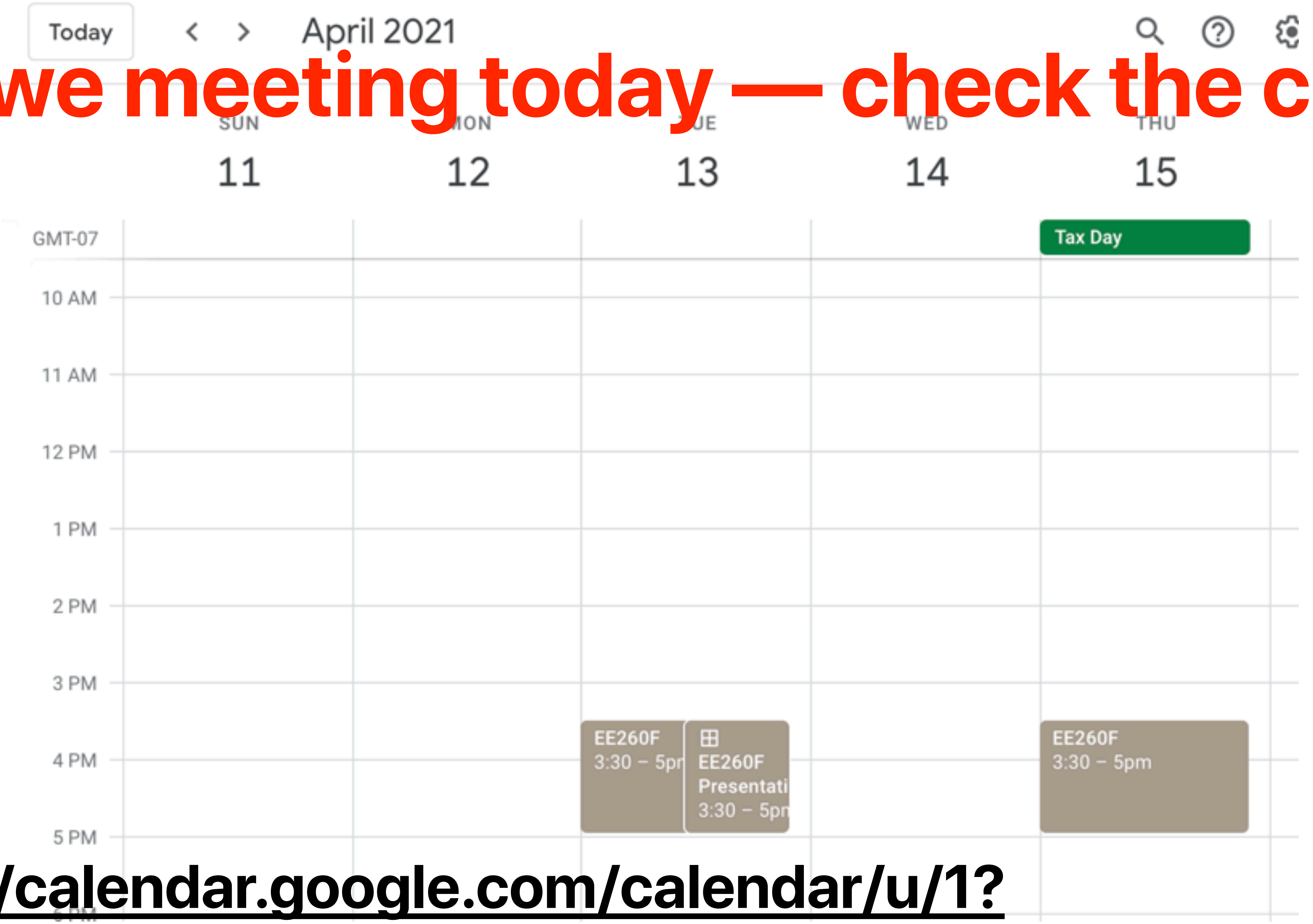


Schedule — on the website

Date	Topics	Readings	Due
01/07/2025	Introduction	Ding/Chong Chapter 1	
01/09/2025	Quantum Computing Basics & Quantum	Metodi/Faruque/Chong	
01/14/2025	Quantum Computing Basics & Quantum		
01/16/2025	Quantum Algorithms	Metodi/Faruque/Chong	Assignment #1
01/21/2025	Quantum Algorithms		
01/23/2025	NISQ Quantum Algorithms	Ding/Chong Chapter 3	
01/28/2025			
01/30/2025	Quantum Systems	Ding/Chong Chapter 4	Assignment #2
02/04/2025			
02/06/2025			
02/11/2025	Circuit Synthesis and Compilation	Ding/Chong Chapter 6	
02/13/2025			
02/18/2025	Microarchitecture	Ding/Chong Chapter 7	Assignment #3
02/20/2025			
02/25/2025	Noise Mitigation & Error Correction	Ding/Chong Chapter 8	
02/27/2025			
03/04/2025	No lectures		Assignment #4
03/06/2025	No lectures		
03/11/2025	Paper Presentations		
03/13/2025	Paper Presentations		

https://www.escalab.org/classes/ee214-2025wi/

Are we meeting today — check the calendar



- https://calendar.google.com/calendar/u/1?cid=dWNyLmVkdV82N2FianBmNXFya3Nvb25uYmQyamlkcWZuNEBn91cC5jYWxlbmRhci5nb29nbGUuY29t

Logistics

- Check https://www.escalab.org/classes/ee214-2025wi/ for slides and the topic of the week
- Subscribe to the calendar for the most up-to-date meeting schedule
https://calendar.google.com/calendar/u/1?cid=dWNyLmVkdV82N2FianBmNXFya3Nvb25uYmQyamlkcWZuNEBncm91cC5jYWxlbmRhci5nb29nbGUuY29t
- You should try to ask questions

Instructor — Hung-Wei Tseng

- Associate Professor @ UC Riverside, 05/2019—
- Website: https://intra.engr.ucr.edu/~htseng/
- E-mail: htseng @ ucr.edu
- Office Hours: W 2p-4p WCH 406
- Visiting Researcher @ Google, 01/2023—03/2023
  - Working for TensorFlow Lite
- PhD in **Computer Science**, University of California, San Diego, 2014
- Research Interests
  - General-purpose computing on AI/ML/NN/RayTracing accelerators
  - Or anything else fun — we have an OpenUVR project recently
- Fun fact: Hung-Wei was once considering a career path as a singer but went back to academia due to the unsuccessful trial

Grading

- Assignments (70%)
  - Assignments will be assigned throughout the course. We will drop your lowest one.
  - Our assignments will be based on Jupyterhub/Jupyter notebooks and programming.
  - There is no group submission or collaboration allowed.
  - There is no regrading on assignments except for obvious grading errors.
  - Please make sure your answers are clearly comprehensible.
- Paper Presentations (20%) — you will need to present one paper this quarter
- Class participation and discussion (10%)

Questions?