

# Building the Infinite Brain

---

COMP 690 (193)

Raghavendra Pradyumna Pothukuchi



THE UNIVERSITY  
of NORTH CAROLINA  
at CHAPEL HILL

✉ [raghav@cs.unc.edu](mailto:raghav@cs.unc.edu)

## What are BCIs?

Broadly, systems that read, process, and modulate neural activity in the brain

## Types of BCIs

Many, both invasive/implanted and wearable

## What are BCIs used for?

Understanding brain function, treat dysfunction, restore lost function, and augment cognition

Intimately tied to an individual's identity

## What are the computational challenges in designing BCIs?

Growing sensor data needs to be processed on the device

Computers today are limited in achieving both efficiency and flexibility, under safety constraints

1-page write-up due on 1/16



# General Suggestions

---

## Take notes

No need to repeat what's in the slides, but note connections, insights

## Write down ideas and follow-ups

Even if you don't plan to revisit them (for now)

## When reading, go down (good) rabbit holes

Trails of papers—builds background knowledge, but don't lose sight of the overall goal: Tree traversal!

## Learn to refine your thoughts into formal and precise summaries, ideas

Even if you don't plan to revisit them (for now)

*“Writing is nature’s way of letting you know how sloppy your thinking is. Math is nature’s way of letting you know how sloppy your writing is.” Leslie Lamport*

# Writing a Paper Review

---



The Task of the Referee

<http://www2.eccs.berkeley.edu/Pubs/TechRpts/1989/CSD-89-511.pdf>

Alan Smith, *Emeritus, UC Berkeley*

Summarize the paper

What did you learn?

What are the novel contributions of the paper?

Positives, strengths etc.

Negatives, limitations

Summary and review of assumptions, methods, results (check correctness too!)

Significance and Impact (past, present or future)

## During paper presentations

1-2 questions for discussion!

Connect the paper to the themes of the course and other papers

# Writing the Project Proposal

---



The Heilmeier Catechism

<https://www.darpa.mil/about/heilmeier-catechism>

George H. Heilmeier, *former DARPA director*

2-page proposals due 1/21

**Goal:** What are you trying to do?

***State of the art and challenges/limitations:*** How is it done today, and what are the limits of current practice?

***Novelty and realizability:*** What is new in your approach, and why do you think it will be successful?

***Significance and impact:*** Who cares? If you are successful, what difference will it make?

***Executional challenges:*** What are the risks?

***Effort:*** How much will it cost?

***Timeline:*** How long will it take?

***Metrics and deliverables:*** What are the midterm and final “exams” to check for success?



# More on The Project

---

## Working project

### Examples

- An FPGA/microcontroller implementation of FFT and analysis of power and performance
- Software to improve/maintain a BCI tool or other software
  - BRAND: <https://github.com/brandbci/brand>
  - Foresee: <https://github.com/bci-foresee/foresee>
- Verification (or Correctness) specification and analysis of a BCI hardware or software kernel
- An EEG BCI game of Pong
- An analysis of algorithms used for seizure detection and their suitability for on-device execution
- A compiler to translate a BCI application to fit microcontrollers
- Python to Verilog translation for DWT in BCI processor design
- Extension of a BCI processing architecture
- Evaluation of a machine learning algorithm for speech decoding and suggested re-design for on-device execution
- Leveraging neuroscientific markers for efficient movement intent decoding on device
- An analysis of cursor movement methods identifying key computational techniques and bottlenecks
  - *(last semester project abstracts will be uploaded for reference)*

You should argue for why your design (hardware, software) is good

You can search online for inspiration, but do not use someone else's code or work—recall the honor code

# For Today

---

- Quick review
- **Introductions**
- **What is computer architecture?**
- **How are computer architectures analyzed and compared?**

# Introductions!

---

**Name**

**Grad/undergrad**

**Focus area**

**Why are you taking the course?**

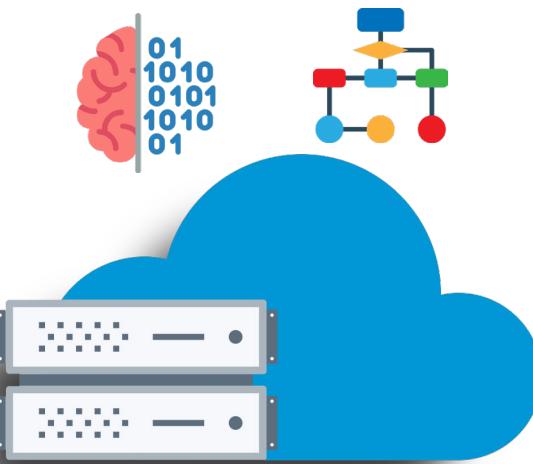
# Example Application

---

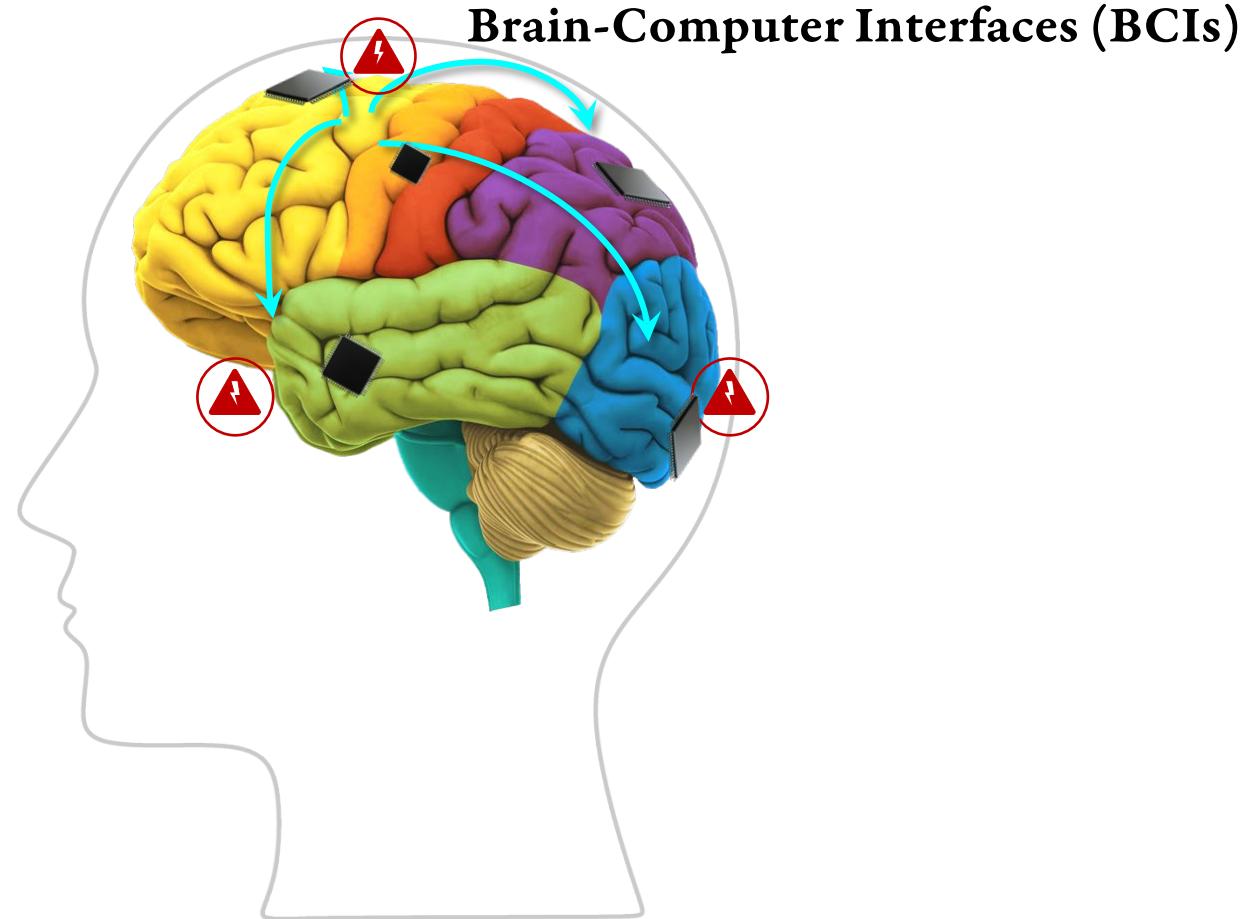
## Restoring function



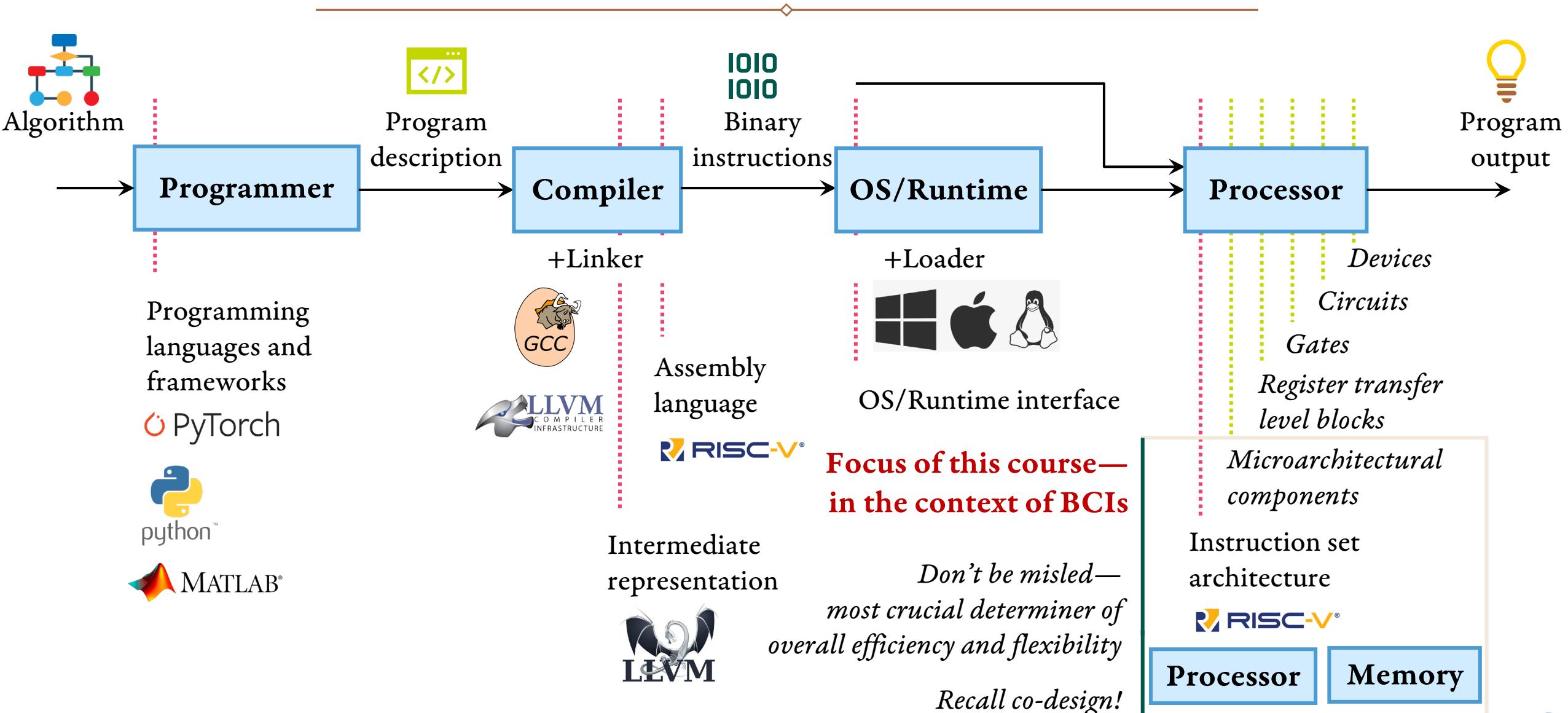
Computational cognitive  
frameworks



## Mitigating seizures



# BCI Computing is a Full Stack Problem



# What is Computer Architecture?

---

**Previously, it meant the ISA**

Most are register based (operands and results in registers), other variants include stack, accumulator

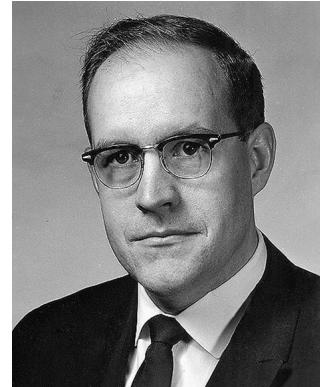
**Now, it could mean everything between the OS and circuit design!**

Distinctions are made based on the context

**What we discuss is also called computer organization**



Gene Amdahl



Fred Brooks Jr

“The term architecture is used here to describe the attributes of a system as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flow and controls, the logical design, and the physical implementation.”

*Architecture of the IBM System/360, Apr 1964*

# Goals of A Computer Architect



Reading task: “Hints and Principles for Computer System Design (2021)”  
<https://arxiv.org/abs/2011.02455>

## STEADY by AID with ART

Butler Lampson, *Pioneer in computer system design*

Applicable for BCI computers too!

**Simple**

**Timely**

**Efficient**

**Adaptable**

**Dependable**

**Yummy**

Is it clean?

Is it ready?

Is it fast?

Can it evolve?

Does it work?

Will it sell?

Goals

**Approximate**

Speculate,  
compress...

**Incremental**

Independent,  
iterate...

**Divide and conquer**

Abstractions,  
concurrent...

Techniques

**Architecture**

**Automate**

**Review**

**Techniques**

**Test**

Process

What?

**Goals**

STEADY

— Simple, Timely, Efficient, Adaptable, Dependable, Yummy

How?

**Techniques** by AID

— Approximate, Incremental, Divide & Conquer

When, who? **Process**

with ART

— Architecture, Automate, Review, Techniques, Test

# Measuring Goals

---

**Simple**

**Timely**

**Efficient**

**Adaptable**

**Dependable**

**Yummy**

Time-to-market

Latency

Uptime

Throughput

Downtime

Energy

Mean time to failure

Throughput per watt

Mean time between failures

Energy per bit

Power

Energy×Delay (ED or EDP)

Quality of service

Tail latency

# Measuring Execution Time

---

$$\text{Execution time} = \frac{\text{instructions}}{\text{program}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{time}}{\text{cycle}}$$

*Path length*      *CPI*  $\left( = \frac{1}{IPC} \right)$       *period*  $\left( = \frac{1}{f} \right)$

# Measuring Power

---

$$\text{Power} = \text{Power}_{\text{static}} + \text{Power}_{\text{dynamic}}$$

$$\text{Power}_{\text{dynamic}} \propto \text{Activity} \times \text{Capacitance} \times \text{Voltage}^2 \times \text{frequency}$$

$$\text{Power}_{\text{static}} \propto \text{Voltage} \times \text{Current}$$

# Improving Efficiency

---

**First get it right, and then, get it fast**

**Efficiency of resources—what are they?**

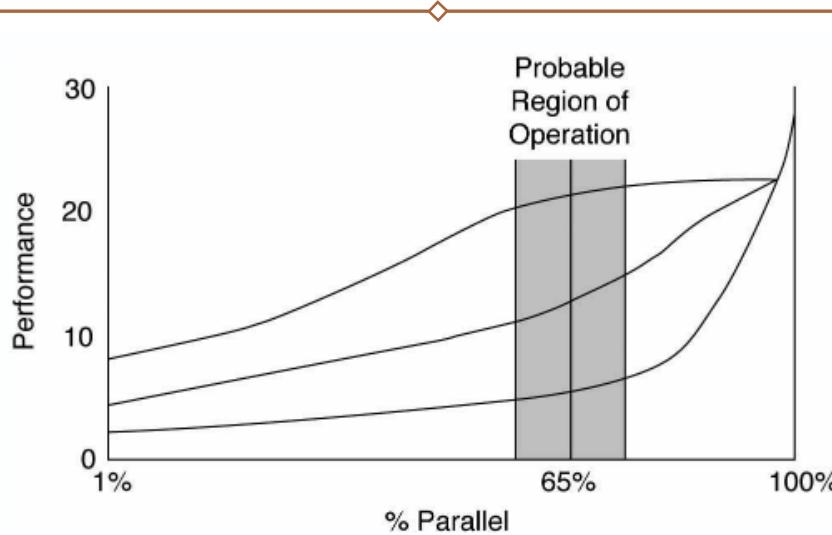
Computing, storage, communication, power, energy

**Concept of a bottleneck**

The part of the system that consumes the most resources

E.g., single server under contention, sequential execution, administration

# Amdahl's Law



“Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities”, AFIPS 1967

$$T_{orig} = T_{seq} + T_{par}$$

$$s = \frac{T_{seq}}{T_{orig}}, f = \frac{T_{par}}{T_{orig}}, p \text{ is speedup of } T_{par}$$

$$T_{new} = T_{seq} + \frac{T_{par}}{p} = T_{orig} - T_{par} + \frac{T_{par}}{p}$$

There may be more!

$$\text{Overall speedup} = \frac{T_{orig}}{T_{new}} = \frac{T_{orig}}{T_{orig} - T_{par} + \frac{T_{par}}{p}} = \frac{1}{(1-f) + \frac{f}{p}} = \frac{1}{s + \frac{f}{p}}$$

Bottleneck!

# Amdahl's Law Class Exercise

---

# How to Fix a Bottleneck?

---

## First find it!

### Change the algorithm

The previous bottleneck no longer applies—e.g., concurrency

### Find a fast path

Make the bottleneck rare—normal case is fast, and some progress is made for the worst case

Slow path time,  $T_s$  and Fast path time,  $T_f$

Slow path probability,  $p \ll 1$

Mean time  $\langle T \rangle \approx pT_s + T_f$

How slow is this over the fast path (the ceiling)?

$$\frac{(pT_s + T_f)}{T_f} = 1 + p \left( \frac{T_s}{T_f} \right)$$

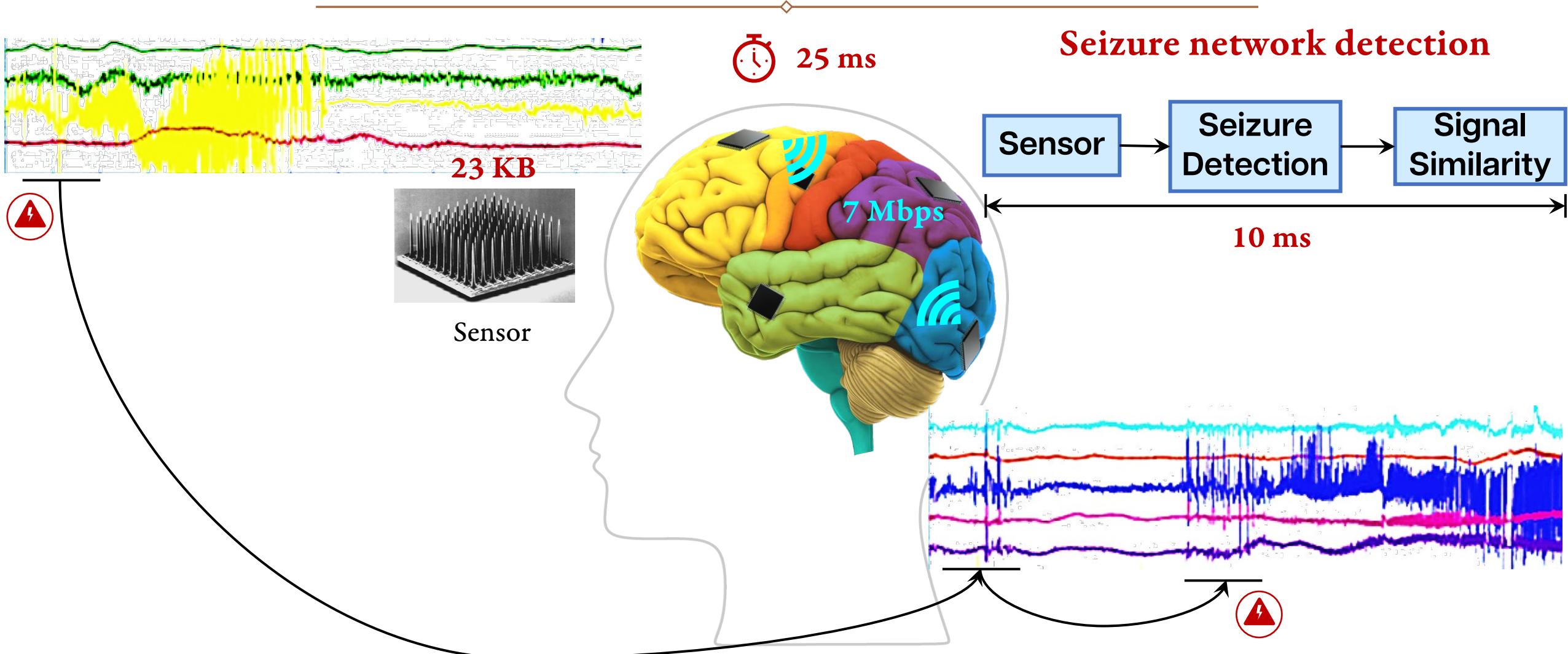
How fast is this over the slow path (the floor)?

$$\frac{T_s}{pT_s + T_f} = \frac{1}{p + \frac{T_f}{T_s}}$$

What can we say about the maximum speedup based on Amdahl's law?



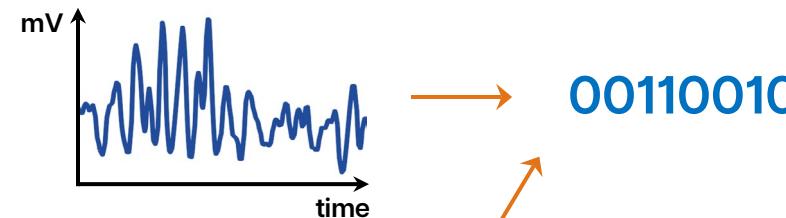
# Example from a BCI Processor (SCALO)



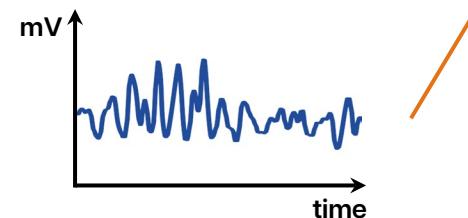
# Rewriting Applications to Filter Communication

---

## Locality sensitive hashing



Equality check!



240 B

1 B

Euclidean Distance:  $O(N)$

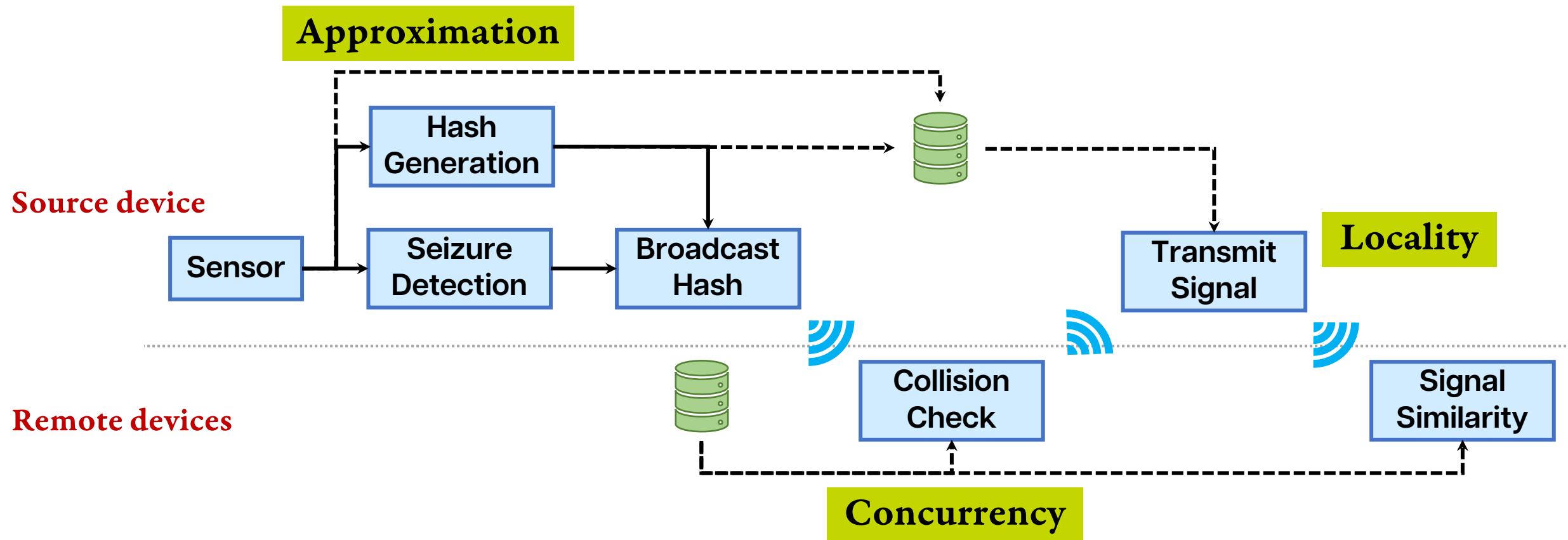
Cross correlation:  $O(kN)$

Dynamic Time Warping:  $O(N^2)$

Earth mover's distance:  $O(N^3 \log N)$

# Rewriting Applications to Filter Communication

Created a fast path!



# More ways to Improve Efficiency

---

## Exploit technology

e.g., at 12 nm, area not a key constraint for BCIs.  
NVMs are a feasibility

## Approximate

Filter, compress, sample, precision, stale data

Predict

Relax synchronization, “correctness” (expectations of correctness)—e.g., HogWild! NeurIPS 2011—but, know how correct you are!

## Locality

Keep data small, keep close

*Batch*, when one-time costs are high and non-recurring—e.g., cache data block access

*Fragment*, when it enables concurrency—  
e.g., pipelining, streaming, sharding.

Independent components

Centralized vs distributed

# Takeaways

1-page write-up due on 1/16  
2-page project proposals due 1/21

---

## What is computer architecture?

Historically, the ISA; but now encompasses organization

## What are the goals?

Mnemonic: Simple Timely Efficient Adaptable Dependable Yummy

## How to estimate impact of fixing bottlenecks?

Amdahl's law

## How to fix bottlenecks?

Algorithms, adding a fast path

## How to improve efficiency?

Technology, approximation, locality, regroup

# Image Credits (Educational, Fair Use)

---

- Title image: VLADGRIN, <https://www.istockphoto.com/vector/human-machine-gm147409511-16840728> (Educational fair use)
- Infinite brain: Science wonder stories, May 1930, Illustrator: Frank R Paul, Editor: Hugo Gernsback
- Brain color, ICs, cloud server, black rat: No attribution required (Hiclipart)
- Hand with spoon: public domain freepng
- Signals: <https://www.nature.com/articles/nrn3724>
- Thought clouds: F. Willett et al./*Nature* 2021/Erika Woodrum, <https://med.stanford.edu/neurosurgery/news/2022/bci-award>. <https://www.thescientist.com/news-opinion/brain-computer-interface-user-types-90-characters-per-minute-with-mind-68762>
- Picture of scientists: <https://www.cs.auckland.ac.nz/~brian/rutherford8.html> (original: Pierre de Latil), Bush (Carnegie Science), Others (Wikipedia)
- Flowchart: Pause08 – flaticon.com
- Digital brain: Smashicons – flaticon.com
- Server rack: upklyak – freepik.com
- Quantum processor icons created by Paul J. - Flaticon
- Arm, Lotus: Adobe stock
- Quantum processor: Rigetti computing
- Images of implanted users: Top: Case Western Reserve University (<https://thecase.case.edu/man-quadriplegia-employs-injury-bridging-technologies-move-just-thinking/>), Bottom: Jan Scheuermann (University of Pittsburgh/UPMC; <https://www.upmc.com/media/news/bci-press-release-chocolate>)
- Images of wearable BCIs: Cognixion, NextMind
- Types of BCIs: “Brain–computer interfaces for communication and rehabilitation,
- Illustrative BCI: Neuralink
- Electrodes: “Electrochemical and electrophysiological considerations for clinical high channel count neural interfaces”, Vatsyayan et al.
- Form factors: Neuropace, Medtronic, Bloomberg, “Fully Implanted Brain–Computer Interface in a Locked-In Patient with ALS” by Vansteensel et al., Blackrock Neurotech
- Jose Delgado’s video: Online, various sources (CNN, Youtube)
- Video of Kennedy and Ramsey: Online, various sources (Youtube, Neural signals)

Logos, trademarks are all properties of respective owners

Not to be shared outside the course

