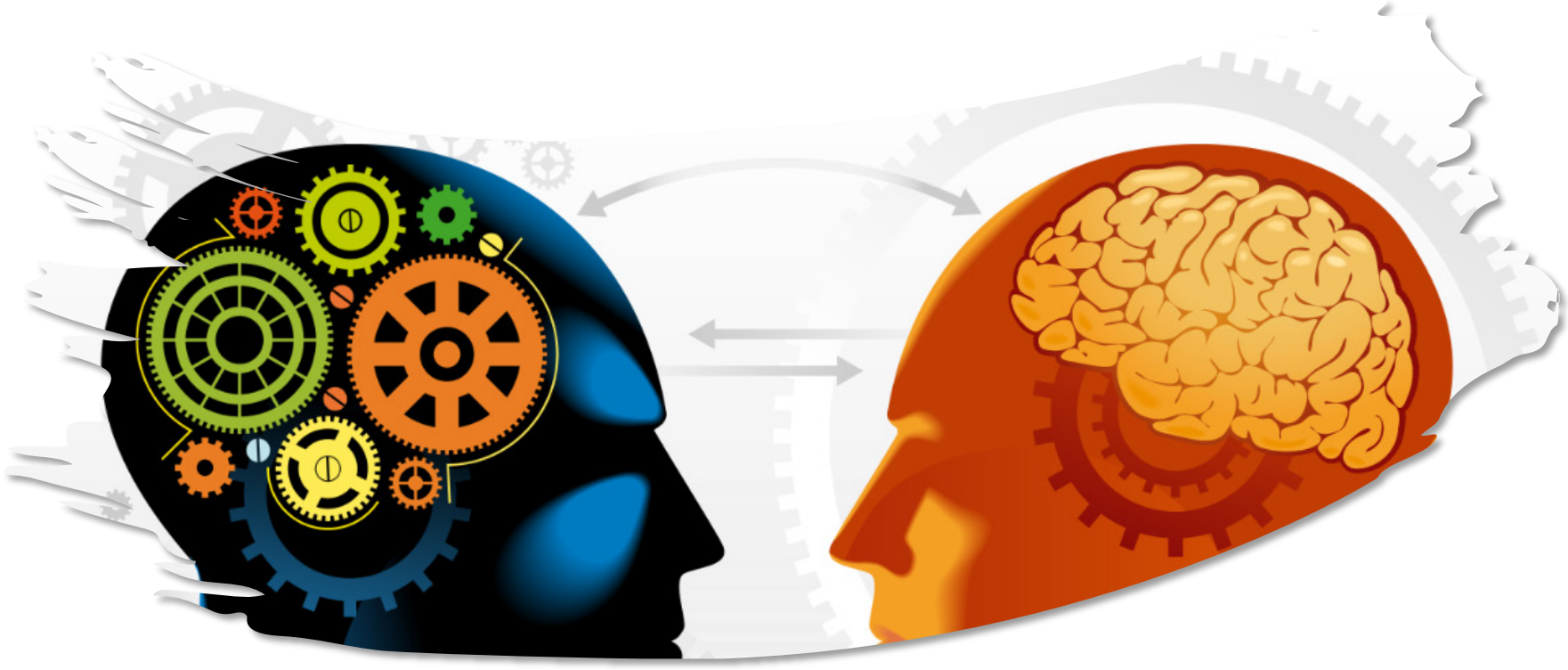


Building the Infinite Brain

COMP 590/790

Raghavendra Pradyumna Pothukuchi

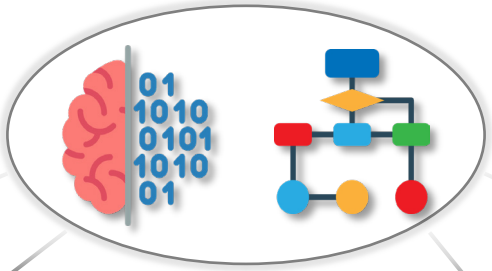
THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

✉ raghav@cs.unc.edu

Building the Infinite Brain

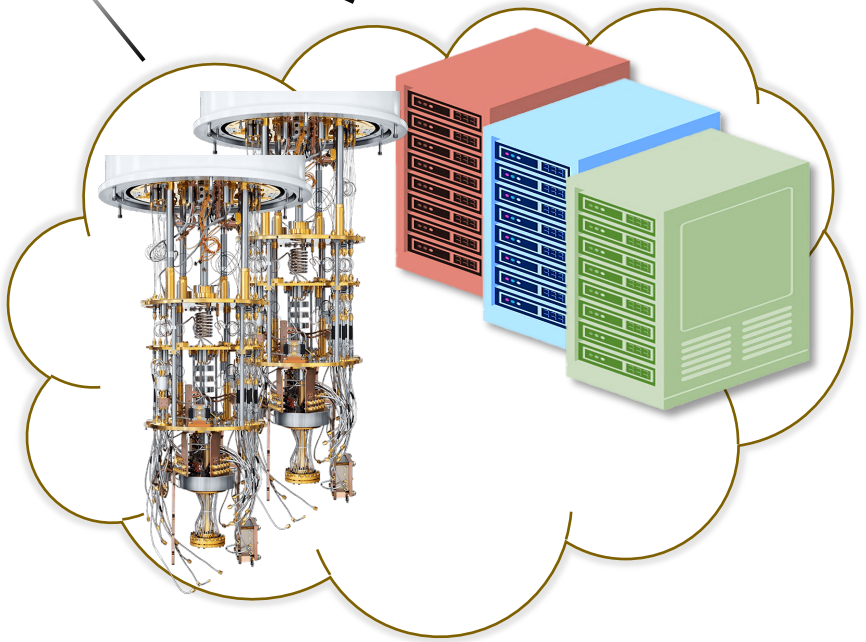
Real-time cognitive frameworks

Tbps, in vivo



Wearables

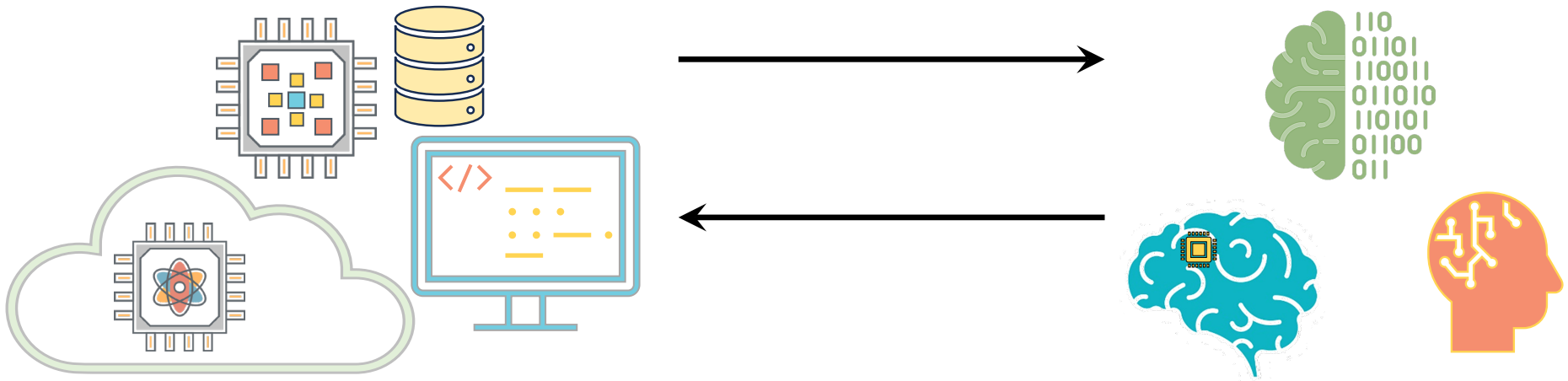
Edge servers



High performance cloud

Central Theme: A Virtuous Cycle of Innovation

Architectures and systems for the brain sciences



Constraints and inspiration from the brain sciences for architectures

BCI Processing Challenge: Efficiency vs Flexibility

Efficiency

High throughput

100-1000 Mbps

Real-time performance

Tens of ms

Safety

≤ 1 °C

≤ few mW

Flexibility

Complex processing

Signal processing, Machine learning

Customization

Personalized treatment,
target multiple conditions, adaptation
to the brain, support evolving methods

Summary: Computer Architecture

What is computer architecture?

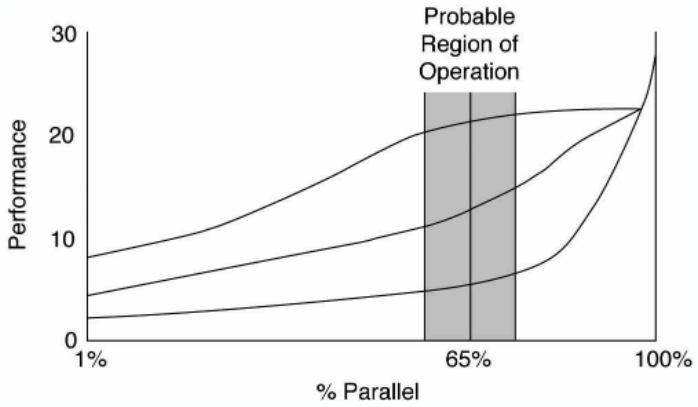
Historically, the ISA; but now encompasses organization

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

Pipelining Summary

What is pipelining?

Splitting work into many components executed independently, and passed in a chain

Why pipelining?

Increases efficiency via parallelism

What are the challenges in pipelining?

Hazards: structural, data, control

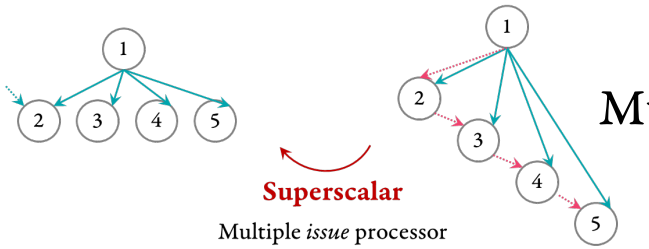
How to fix hazards?

Concurrency (structural), eager (forwarding data), eager or speculative (branch delays, prediction)

What systems design principles does pipelining touch?

Tradeoff simplicity for efficiency, leveraging parallelism through fragmenting and regrouping

Superscalar and Dynamic Scheduling Summary



What are superscalar processors?

Multiple issue processors (send multiple instructions to execute)

Why dynamic scheduling?

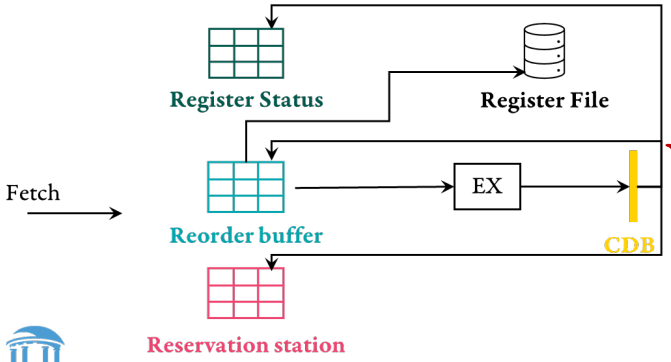
To be resource efficient in exploiting instruction level parallelism

What are hardware methods for dynamic scheduling?

Scoreboarding, Tomasulo's algorithm, speculation

What systems principles does dynamic scheduling involve?

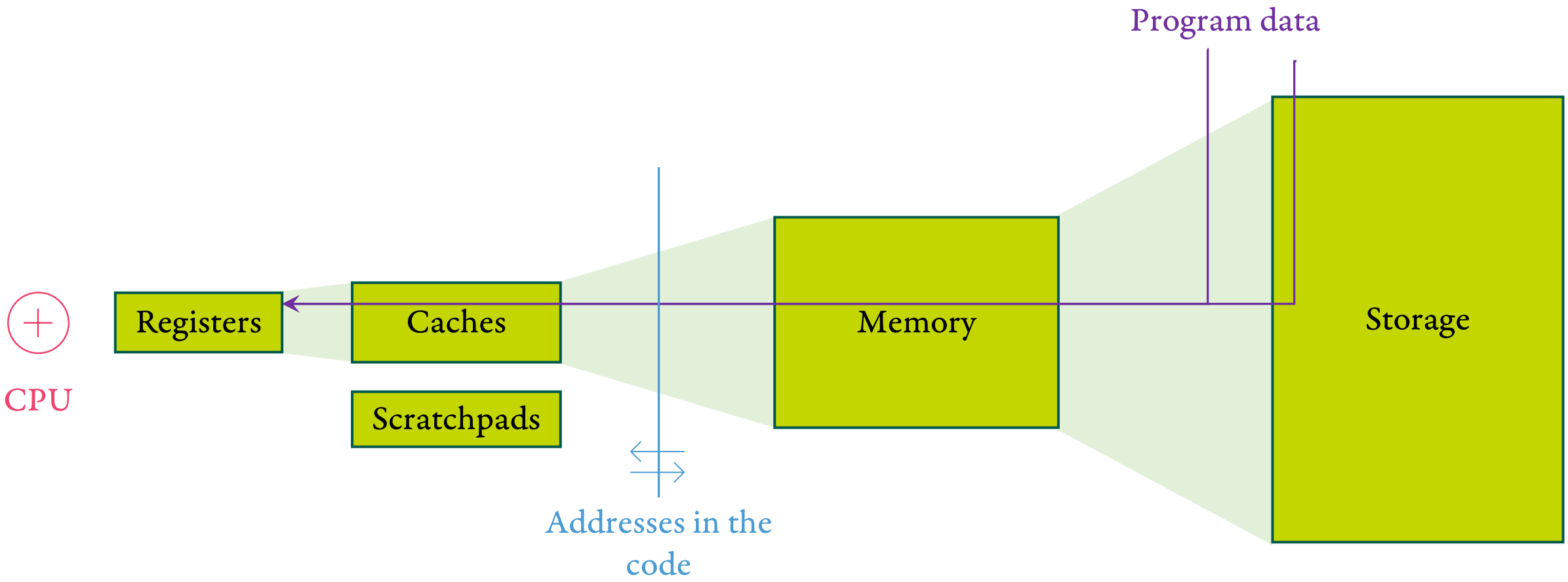
Eager, speculative, and concurrency



What are some other methods to exploit ILP?

Software pipelining, unrolling, which target static ILP

Memory Hierarchy



Cache Summary

What is caching?

A technique to minimize the impact of long memory latencies

What are the basic cache parameters?

Associativity (placement), block size, capacity, write through/back, write-allocate or no

What are the types of cache misses?

Compulsory, capacity, conflict

What are some ways in which cache performance can be improved?

Non-blocking, banking, software or hardware prefetching etc.

How to choose the right memory hierarchy design?

Tailor it to the access patterns and layout: general purpose to domain specific

Memory Summary

What are some goals in designing the memory subsystem?

Capacity, program abstraction, protection, and sharing

What is virtual memory?

Abstraction to deliver the goals

What is a bottleneck in virtual memory systems?

Address translation

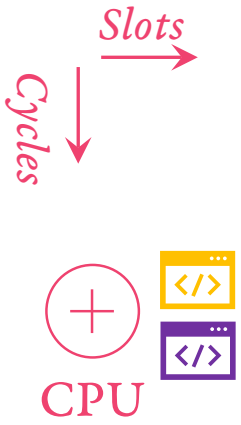
How can addressing bottlenecks be tackled?

Through fast paths: TLBs

How to choose the right memory hierarchy design?

Tailor it to the access patterns and layout: general purpose to domain specific

Multithreading

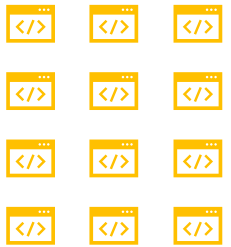


Scalar

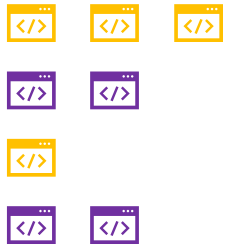
Fine (interleaved)

Coarse (block interleaved)

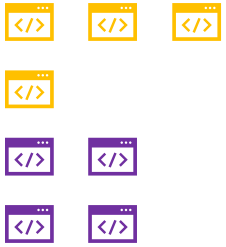
Superscalar



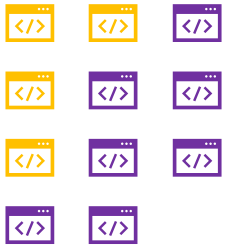
Fine

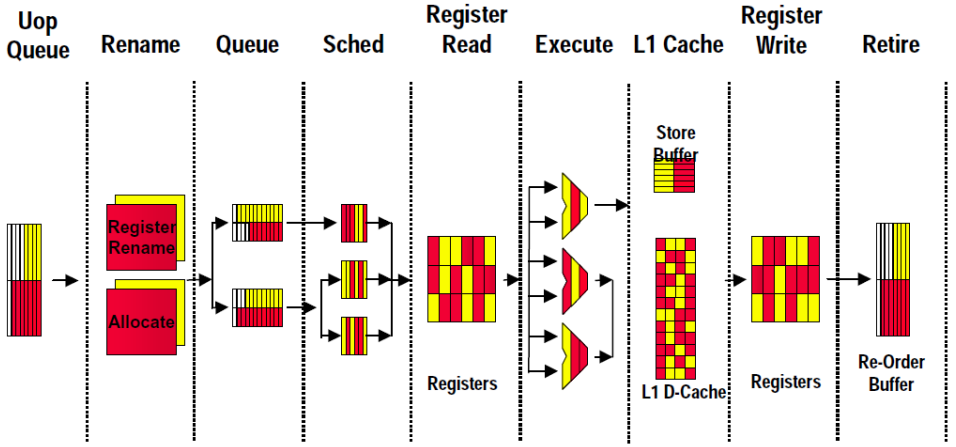


Coarse

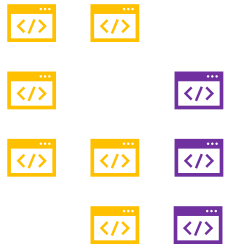


Simultaneous





Very Large Instruction Word (VLIW)
Static

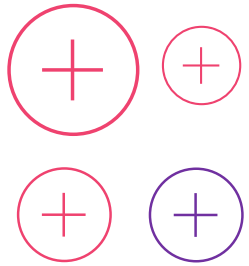


"Hyper-Threading Technology Architecture and Microarchitecture", Intel Technology Journal Q1, 2002

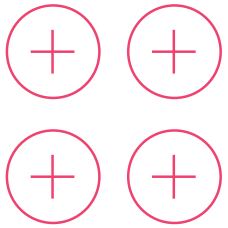
Multicore Organization

Compute

Asymmetric

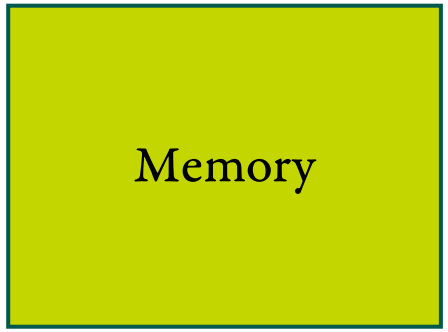


Symmetric

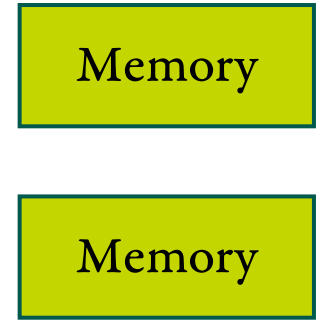


Memory

Centralized



Distributed



Uniform or Non-uniform memory access

Multicore and Multiprocessor Summary

What are the ways to improve throughput beyond single instruction stream?

Multithreading, chip multiprocessing, and combinations

What is the difference between multithreading and chip multiprocessing?

Multithreading: multiple instructions on the same core/processor/CPU
Chip multiprocessing: using multiple cores on the same chip

What are the aspects of multicore organization?

Compute (symmetric or asymmetric) and memory (centralized or distributed)

How are hardware and software parallelism related?

Abstractions are different—software (threads, processes), hardware (threads/contexts)
Hardware: implicit (single instruction stream) or explicit (multiple instruction streams)
Software: sequential (via automatic parallelism) or parallel (shared memory or message passing)

Coordinating Memory Accesses

Define correct behavior!

"Write the spec"

Shared data block

Cache coherence

Also called cache consistency

All accesses

Memory consistency

Applies to sequential programs too!

Multicore Memory Summary

What is the challenge with (multicore/distributed) system design and programming?

Implementing and reasoning *correctly* about it

What are two specifications that help the above?

Cache coherence: about writes to the same location across caches
Memory consistency: about ordering of memory accesses

What is cache coherence?

Each write is eventually visible and writes to the same location are serialized

What is memory consistency?

Specification of what values a read can return and when

What are the principles at play in defining and implementing coherence and consistency?

Abstraction, Efficiency, “Yummy”, Concurrency, Approximate

Parallel Architectures

Low-level parallelism

Pipelining, superscalar, CPU+I/O separation

Parallel architecture

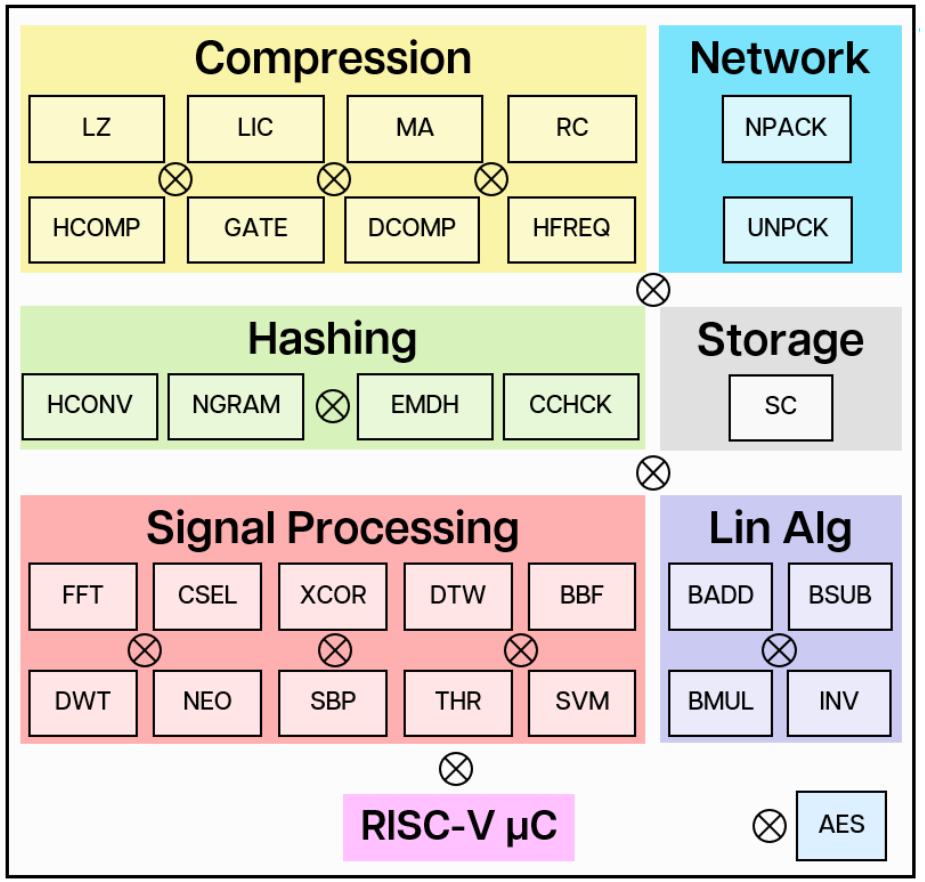
Multiple processors executing concurrently to solve a problem, with an explicit high-level framework

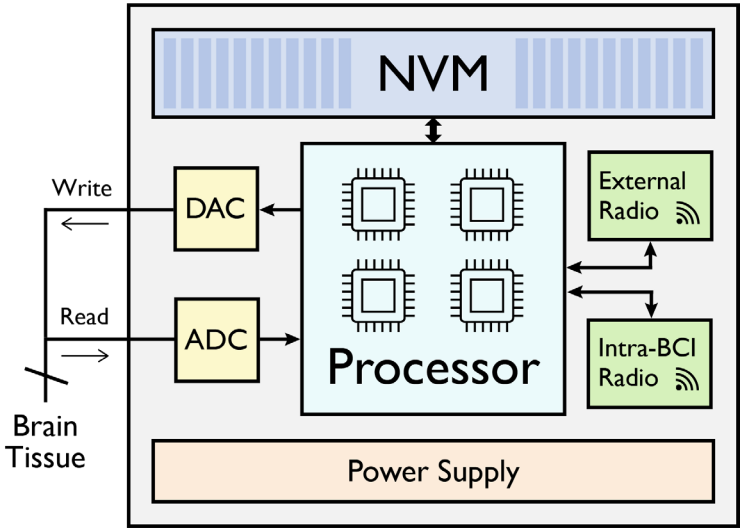
Various classifications and taxonomies

Flynn: Single/Multiple Instruction and Single/Multiple Data

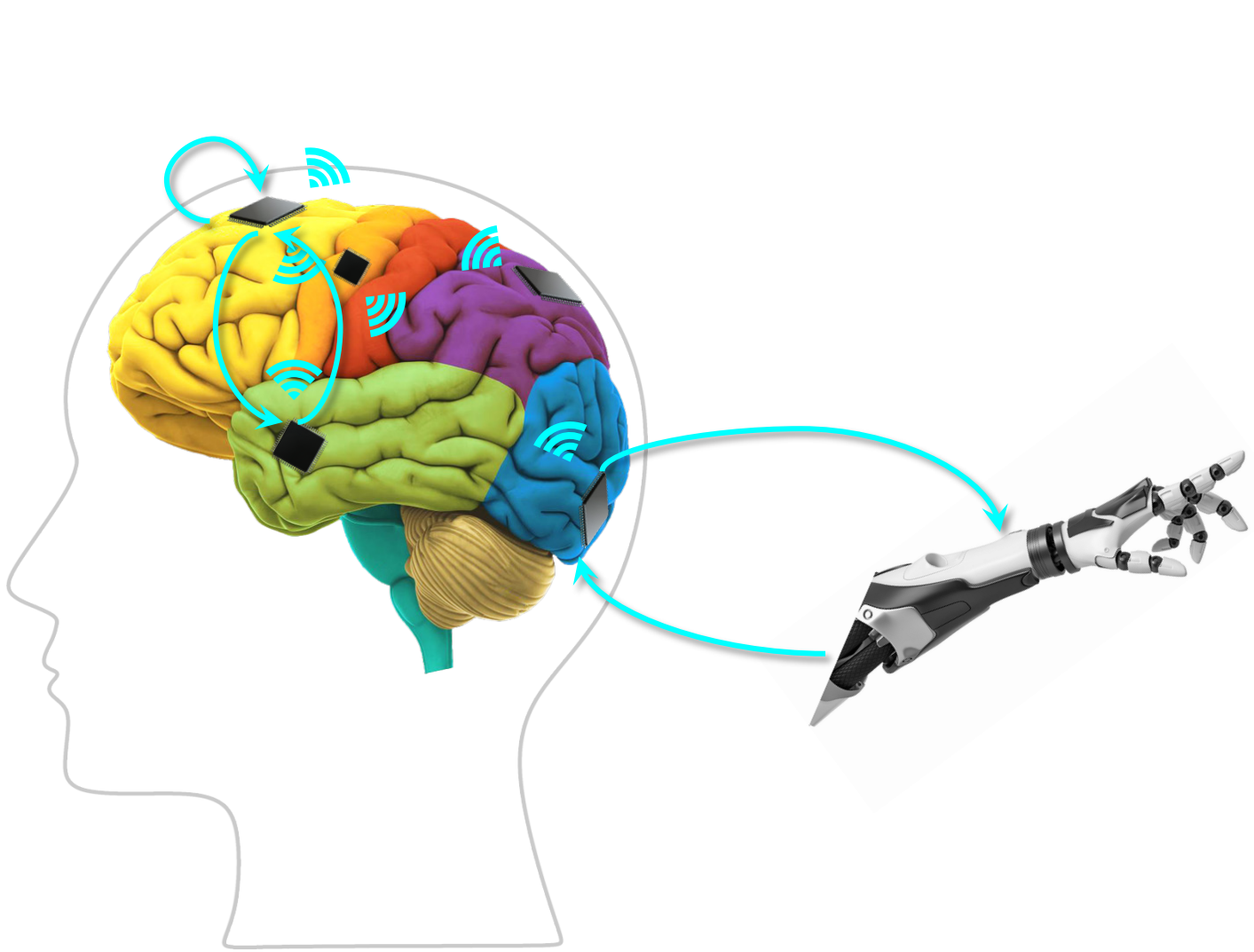
Vector processors, SIMD units, Associative processors, Systolic arrays,
MIMD (Shared and distributed memory)
Dataflow processors

Example with a BCI Processor: SCALO





Example with a BCI Processor: SCALO



Dataflow

Static resource sharing and ordering

Software PE scheduling

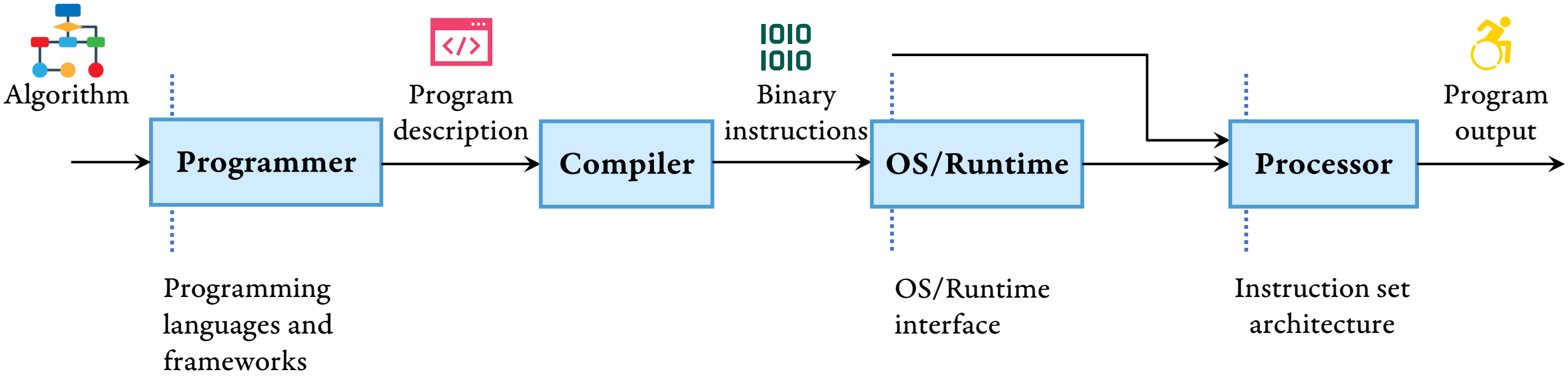
Co-designed fast paths

Explicit communication

No caching

Non-blocking writes with merging

BCI Computing is a Full Stack Problem



Emerging user trends (multimodal, wireless, portability)

Emerging algorithms (DNNs, Online learning, RNNs)

Emerging hardware trends (Beyond ASIC)

Co-design (Infinimind, Marple, Noema, KalmMind)

Abstractions, Programming or Development Platforms (BRAND, xDev, OpenVIBE)

Full stack constraints (Neuralink)

What You Should Have Learned

Computer architecture concepts

Recognizing and understanding them

BCI applications and algorithms

Understand what BCIs can offer, and how they work

BCI computational needs

Understanding them and the sources of these needs and constraints

BCI state of the art

Understand latest developments in the broad BCI space across the stack, limitations, future directions

Apply computer system design to BCIs

Demonstrate understanding of BCIs and computer system design through a working project

Feedback on Paper Reading

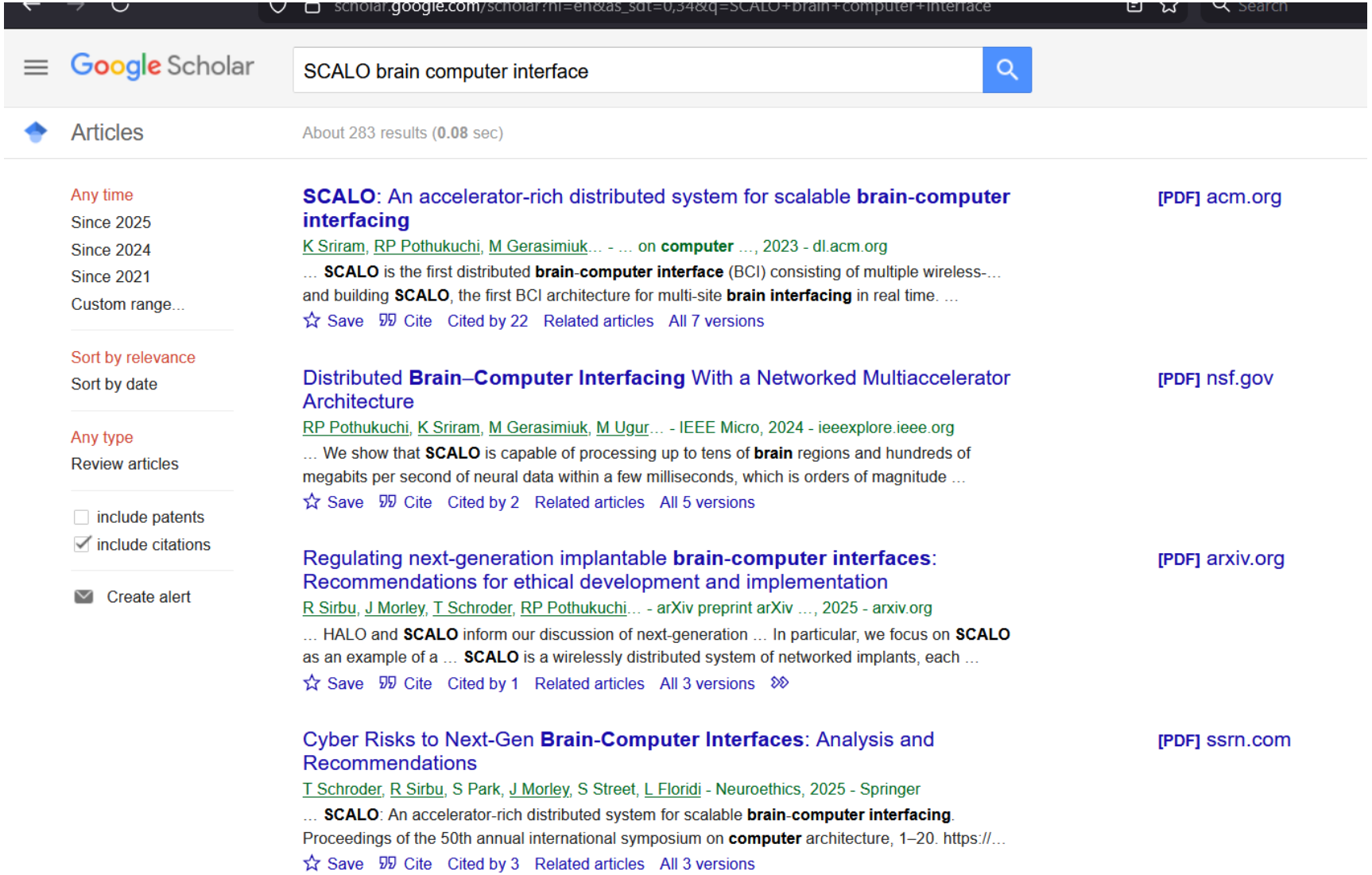
Read in detail

Multiple passes

Use manual search first

You'll discover many things—helps with breadth and depth

Feedback on Paper Reading



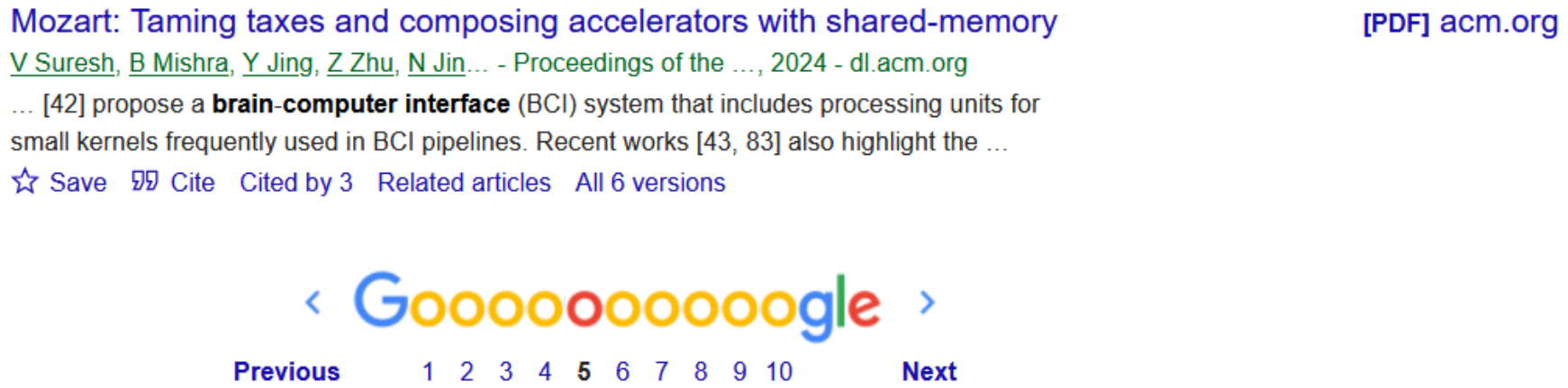
Feedback on Paper Reading

Read in detail

Multiple passes

Use manual search first

You'll discover many things—helps with breadth and depth



Stimulates broad domain learning, abstract reasoning, and multi-task learning

Feedback on Presentations

Different types of presentations

Tease them (motivate the interaction), Show them (storytelling), Throw at them (detailed review)

Presentation is an aid to an immersive experience

Draw attention where you want through color, font, objects, graphics, and *absence*
Reading notes or long text stimulates reading or hearing—not listening, and not engaging

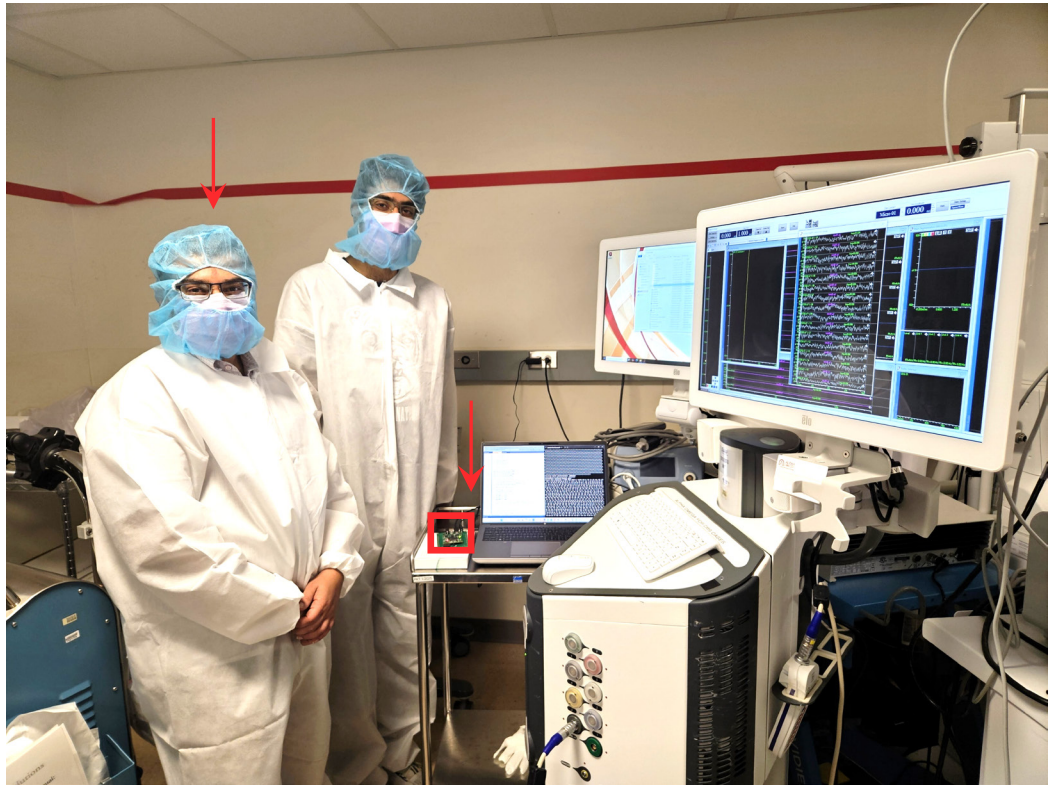
Feedback on Presentations

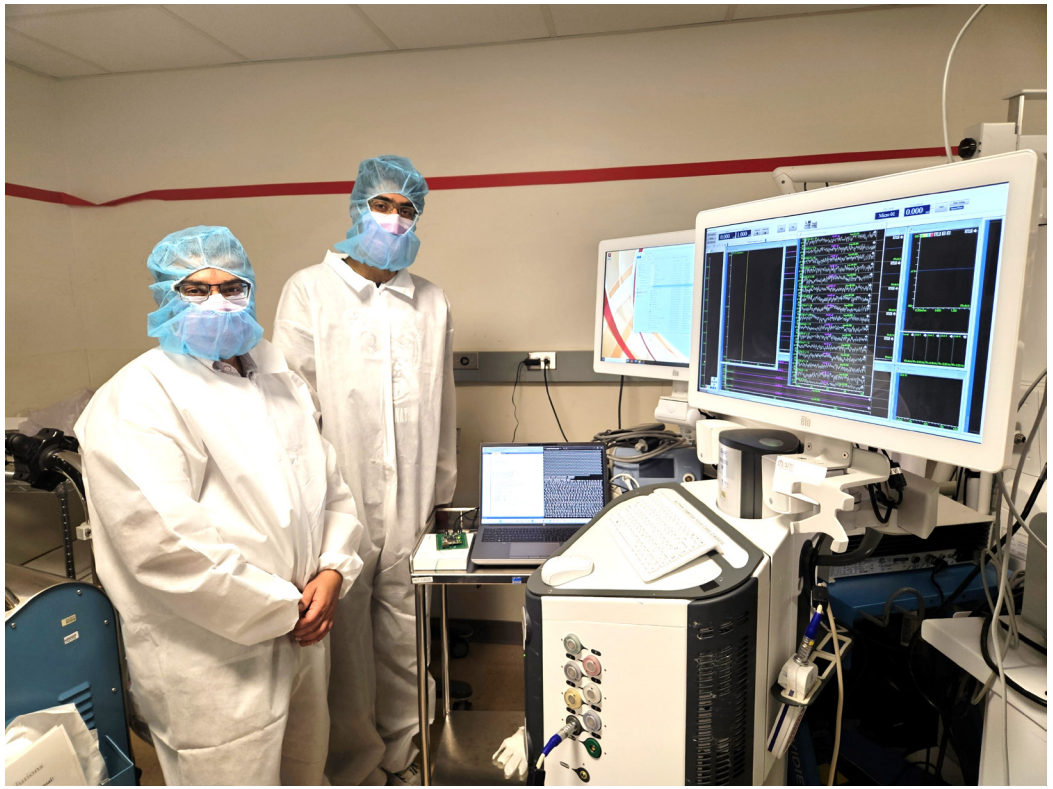
Different types of presentations

Tease them (motivate the interaction), Show them (storytelling), Throw at them (detailed review)

Presentation is an aid to an immersive experience

Draw attention where you want through color, font, objects, graphics, and *absence*





Feedback on Presentations

Different types of presentations

Tease them (motivate the interaction), Show them (storytelling), Throw at them (detailed review)

Presentation is an aid to an immersive experience

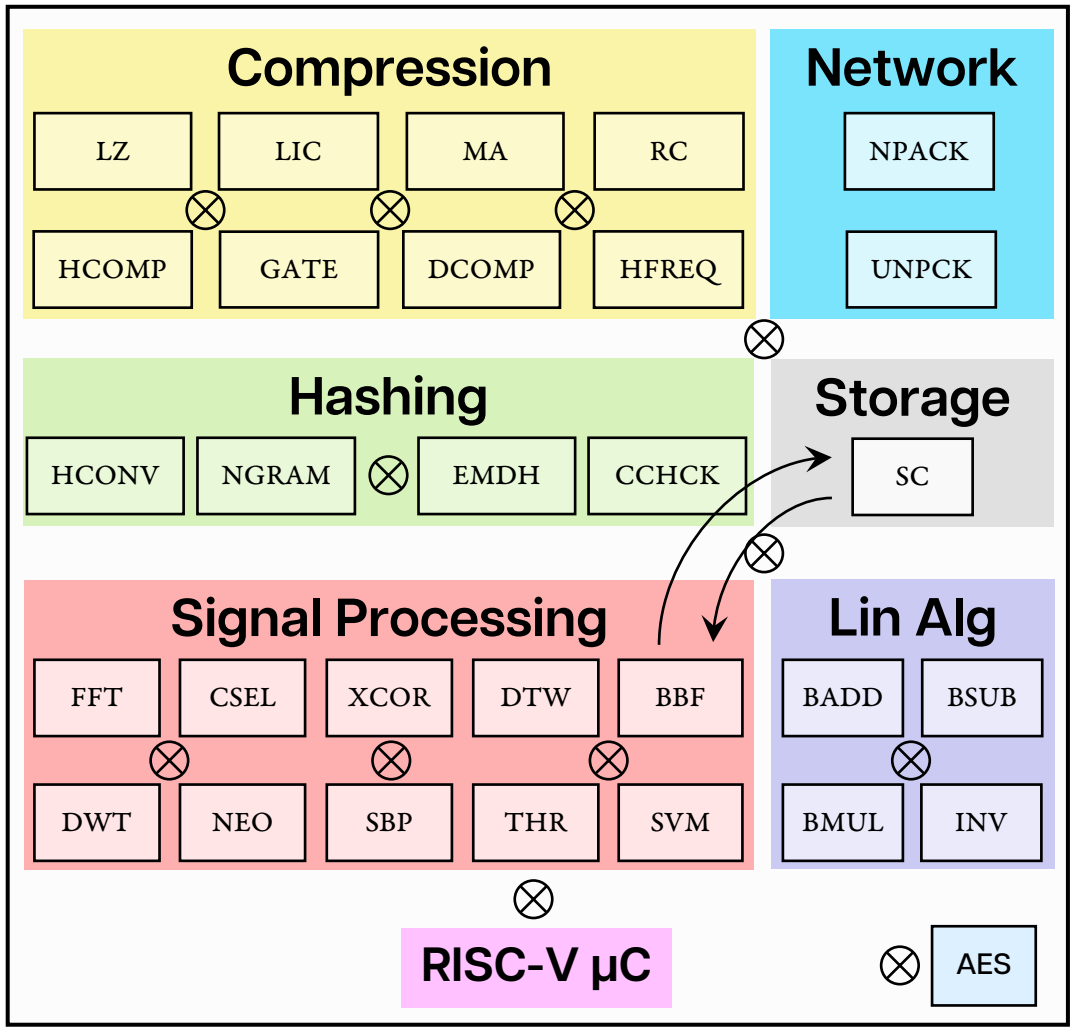
Draw attention where you want through color, font, objects, graphics, and *absence*
Reading notes or long text stimulates reading or hearing—not listening, and not engaging

Lead the audience

Move from a high-level introduction to the nitty gritty and go back up with the wisdom

Example slide follows

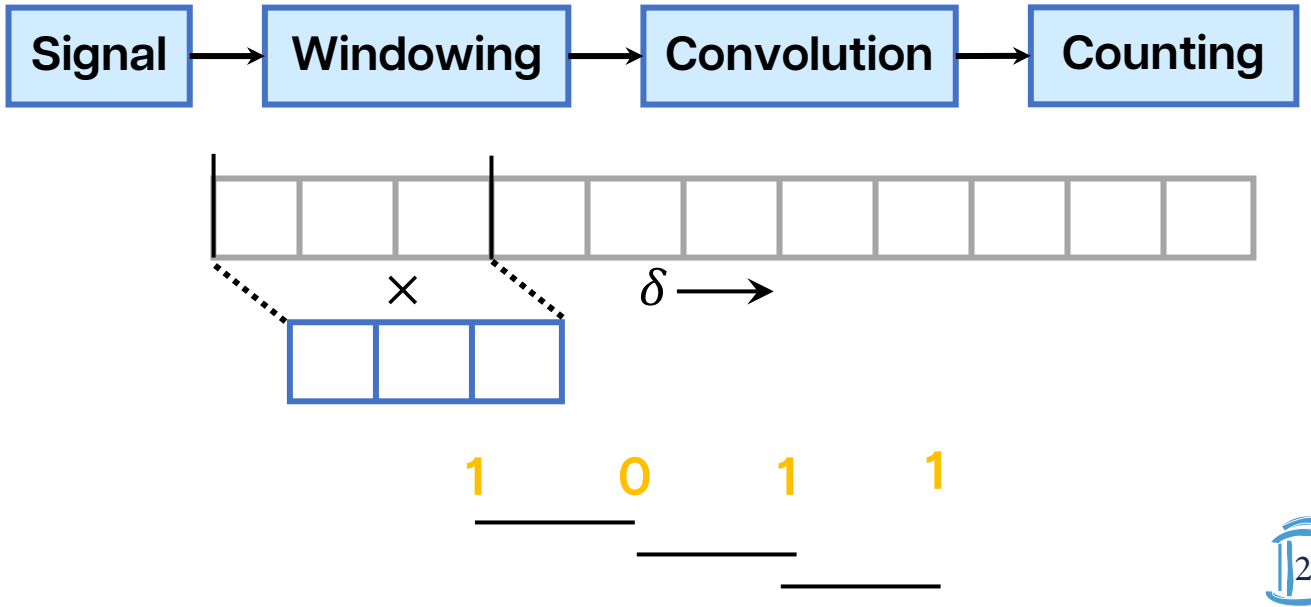
Accelerator-Rich Architecture



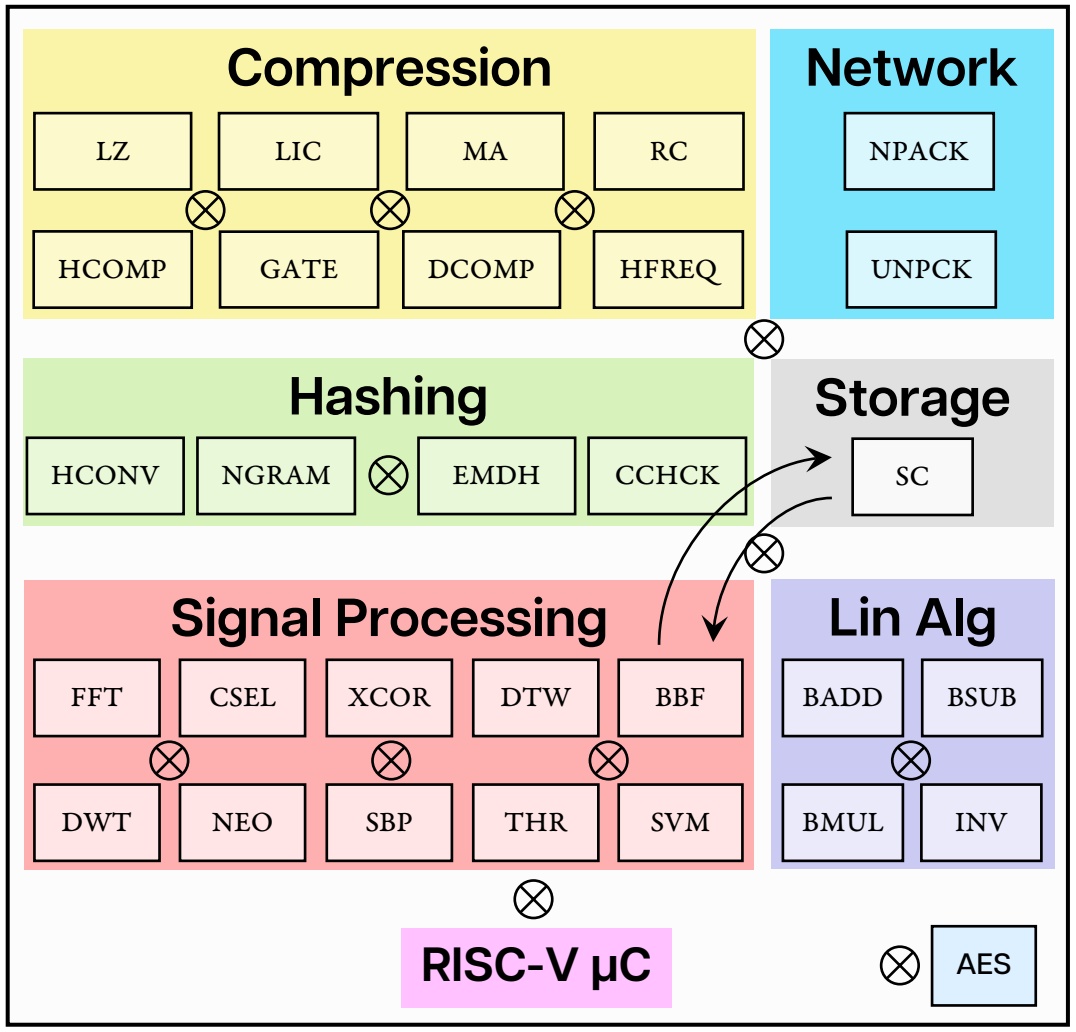
Optimize processing elements (PEs) for reuse

Locality Sensitive Hashing

Dynamic Time Warping, Euclidean Distance, Correlation, Earth-Mover's Distance



Accelerator-Rich Architecture



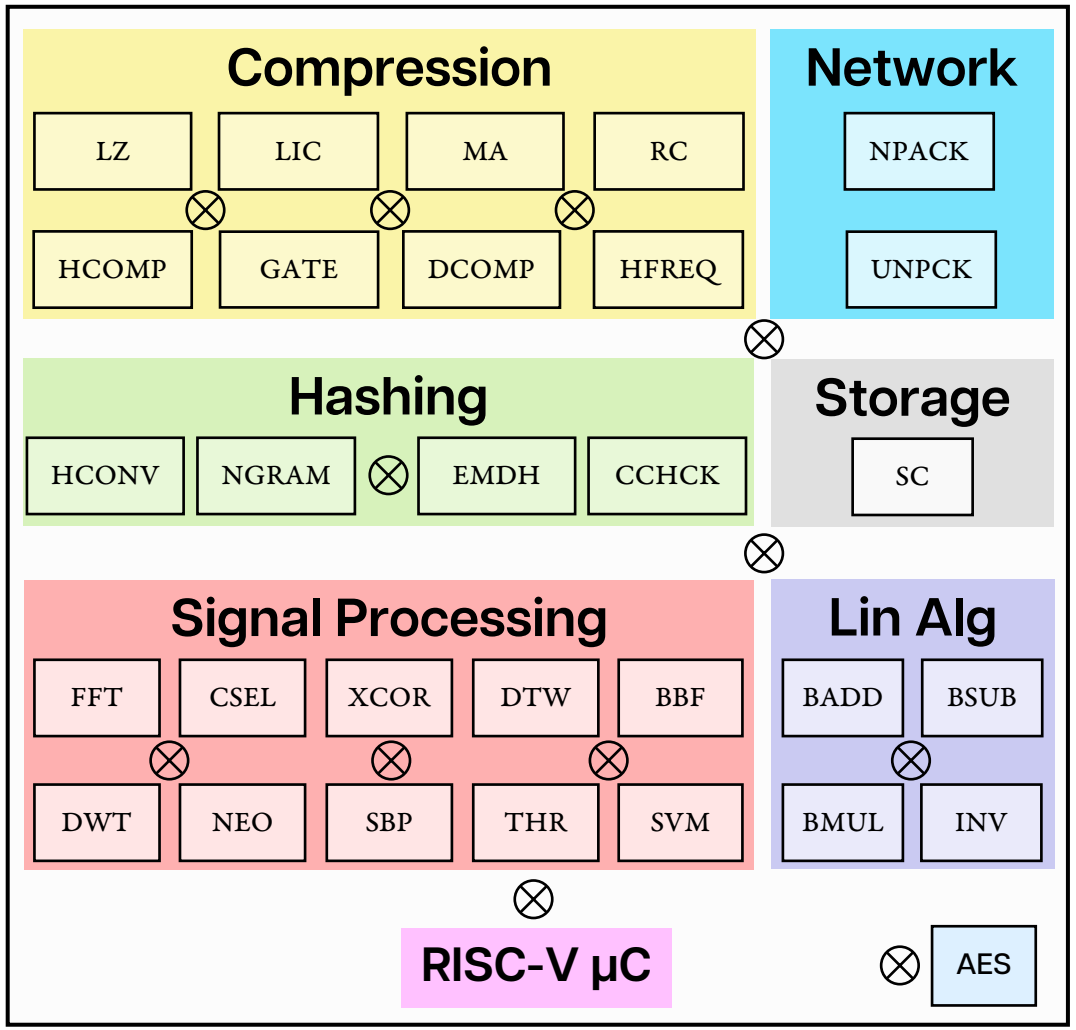
Optimize processing elements (PEs) for reuse

Per-PE clock, and frequency scaling

Configurable interconnect

Predictable power and performance

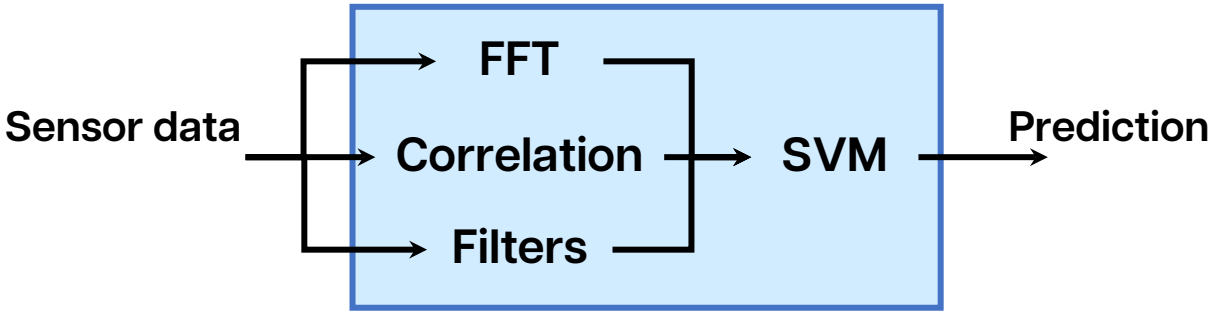
Accelerator-Rich Architecture



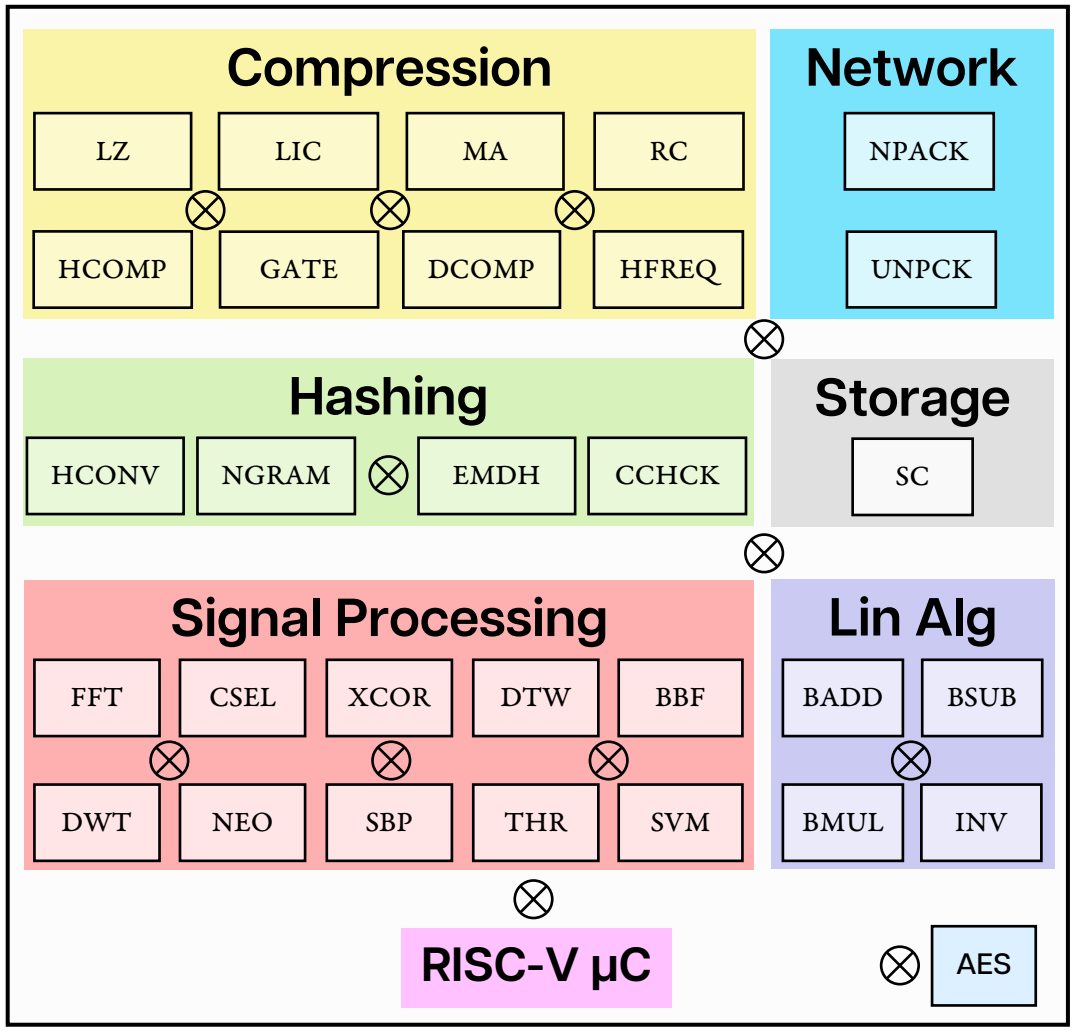
ISA, but with accelerators, for BCI applications

Accelerator as the instruction

Efficiency with flexibility



Accelerator-Rich Architecture

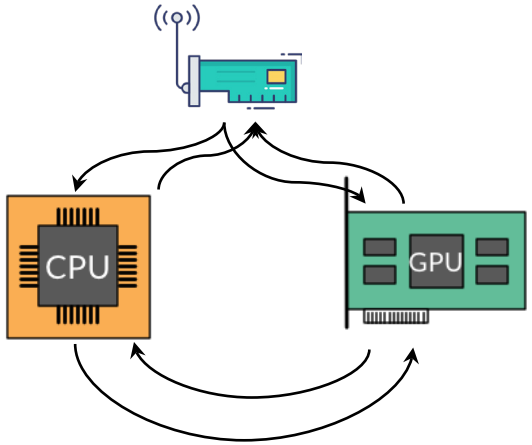


ISA, but with accelerators, for BCI applications

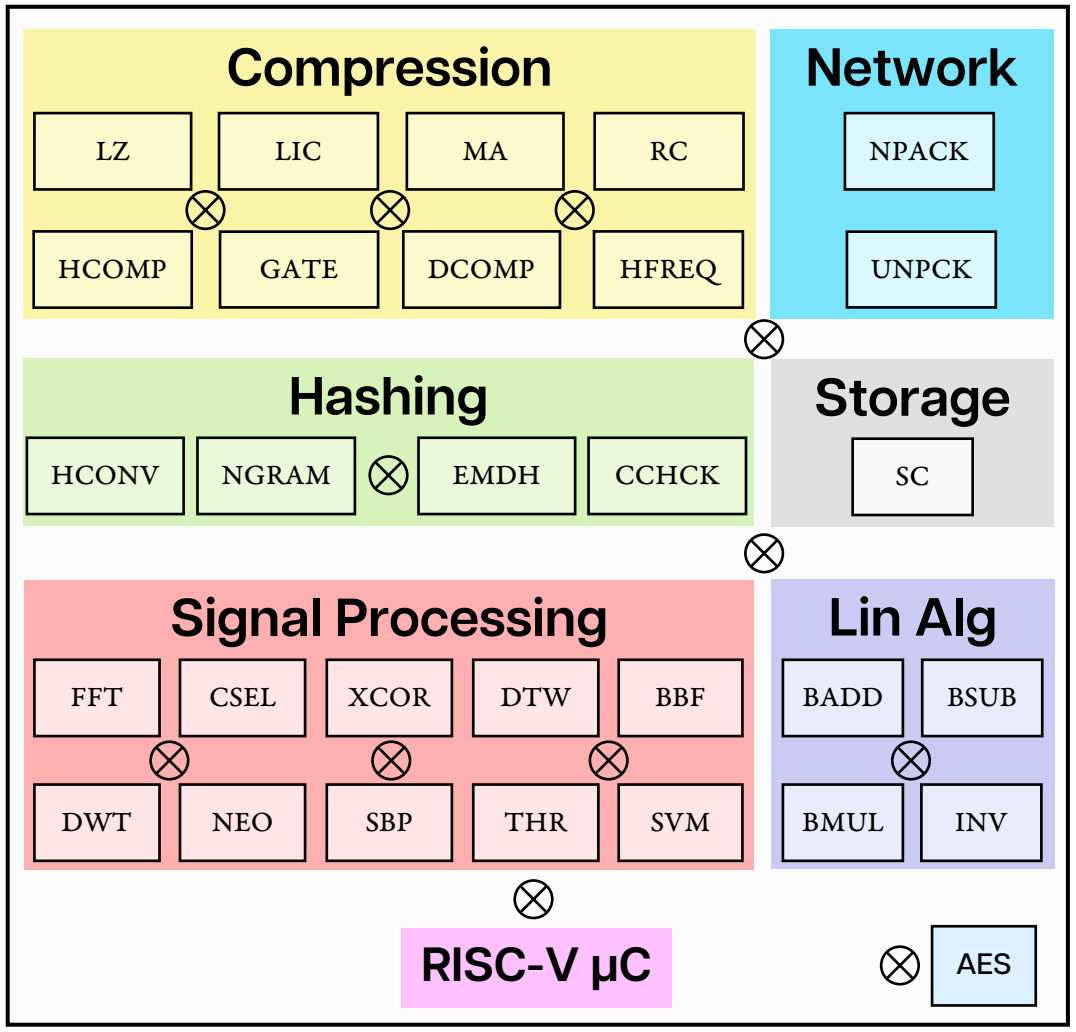
Accelerator as the instruction

Efficiency with flexibility

Minimizing CPU-led overheads



Accelerator-Rich Architecture



ISA, but with accelerators, for BCI applications

Accelerator as the instruction

Efficiency with flexibility

Minimizing CPU-led overheads

Hash-based communication for scalability

Feedback on Presentations

Different types of presentations

Tease'em (motivate the interaction), Show'em (storytelling), Throw at'em (detailed review)

Presentation is an aid to an immersive experience

Draw attention where you want through color, font, objects, graphics, and *absence*
Reading notes or long text stimulates reading or hearing—not listening, and not engaging

Lead the audience

Move from a high-level introduction to the nitty gritty and go back up with the wisdom

Make the audience think, and be excited

Show connections that weren't obvious, discuss significance, tie to real examples that matter

Practice!

Perfect practice makes perfect—to deliver the best experience

Feedback?

Content

Delivery

Organization

Alignment with expectations

Knowledge and skill building

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.the-scientist.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, National Academies, IEEE, and university profile images)
- Flowchart: Pause08 – flaticon.com; Digital brain: Smashicons – flaticon.com; Quantum processor icons created by Paul J. - Flaticon
- Server rack: upklyak – freepik.com
- Arm, Lotus: Adobe stock
- Quantum processor: Rigetti computing
- Images of implanted users: Top: Case Western Reserve University (https://thedaily.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)
- Code snippet inspiration: ECE 252 slides at Duke (Dan Sorin et al.)
- Apple processor pipeline: https://dougallj.github.io/applecpu/firestorm.html