A recording of this talk is available on YouTube at this link.

Natural Computing

Utilizing nature's untapped potential for computation, data storage, and data movement.

V. Hunter Adams

November 2023

Humanity has a long history of using natural processes for the production of **power**, but it has underutilized these processes for **computation**.

Can we unlock the latent computational potential in nature, and change the relationship between nature and machines?

This isn't a research talk.

It's a "should this be researched?" talk.

In this presentation . . .

- What is **computing**, and what is a **computer**?
- **Evolution** and **physics** produce computers, and the components of computers.
- We can interface these **natural computers** with our engineered computers. Doing so will increase total global compute, and improve the relationship between nature and machines.
    - Precedent and context for this claim
    - Near-term natural computing opportunities
    - Long-term natural computing possibilities
    - Technical problems which must be solved
- Getting there from here.

Computation is the *useful transformation* of one quantity (or quantifiable system) into another quantity (or quantifiable system).

A **computer** is anything which does computation.

Utility is in the eye of the beholder. The answer to the question “is this a computer?” is **subjective**.

Computation is the *useful transformation* of one quantity (or quantifiable system) into another quantity (or quantifiable system).

A **computer** is anything which does computation.

What is a computer?

Digital | Analog | Quantum | Natural

- Quantities are represented **symbolically** (by bits, beads, gear rotations, etc.).
- We transform the *representations* for these quantities, rather than the quantities themselves.
  - When you double a quantity in a digital computer, there isn't twice as much of anything in that computer. Instead, the symbol/representation for the quantity is transformed to the symbol/representation for twice that quantity.
- Digital electronic computers symbolically represent these quantities with bits, encoded as **binary voltages in latch circuits**.
  - Transformations of the quantities can thus be performed using logic circuits. Claude Shannon showed that that such circuits can solve any problem that Boolean algebra can solve, and thus any Boolean transformation of the quantity can be performed.
- This enables digital computers of two varieties: special purpose and general purpose

V. Hunter Adams

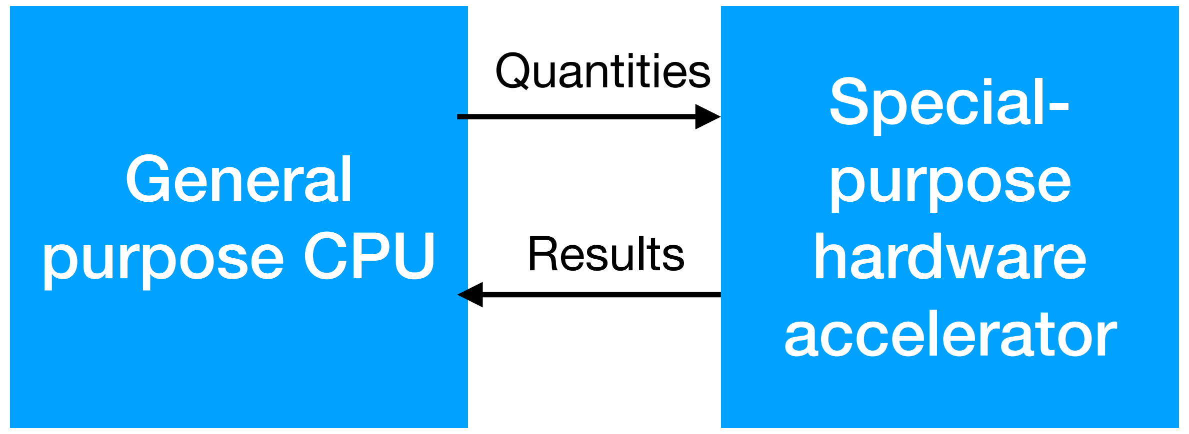
What is a computer?

Digital | Analog | Quantum | Natural

Special-purpose

We design a collection of circuits which implement Boolean algebra. The symbols for some number of quantities are communicated into these circuits as voltages, and the transformed quantities (the result of the operation which the circuits implement) come out as voltages on wires or in registers.

Think ASIC's, and **hardware accelerators**



Take 5760 to learn more!

Lattice-Boltzmann
Mandelbrot
2D Wave equation

General-purpose

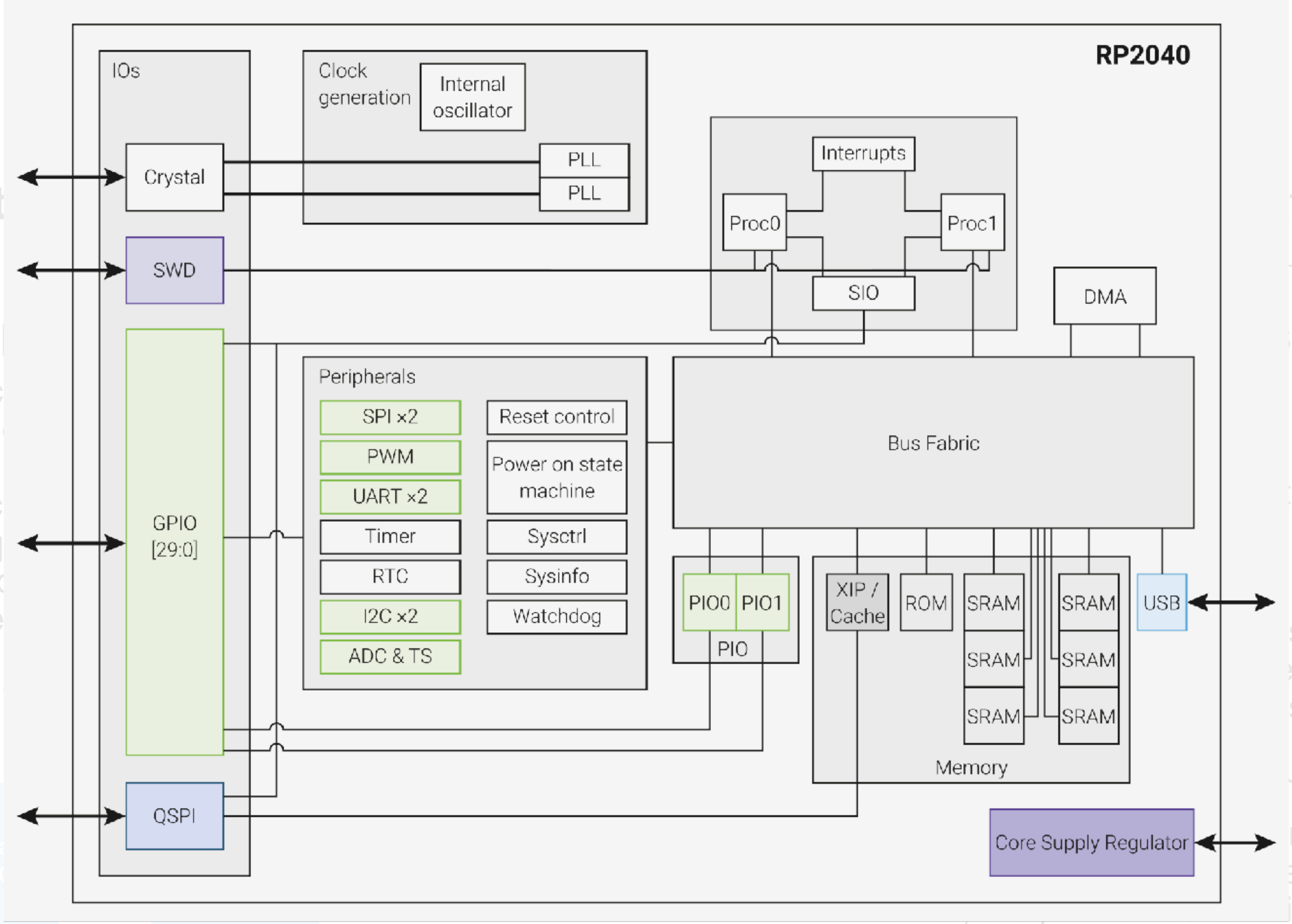
We implement and integrate some subsystems. To oversimplify, these include:

Memory, to hold data and instructions.

A processing unit, to execute the instructions (transformations) that quantities in the instruction memory represent. These transformations act on data from elsewhere in memory, or from an input mechanism.

A bus, for moving data between subsystems.

Peripherals. These are special-purpose subsystems (a timer, a serializer/deserializer, a DMA channel, etc.) which do a limited number of things efficiently, and communicate with the CPU over the bus.



System architecture for RP2040 microcontroller. This is one level of abstraction up from the computer architecture.

What is a computer?

Digital | Analog | Quantum | Natural

Special-purpose

We design a collection of circuits which implement Boolean algebra. The symbols for some number of quantities are communicated into these circuits as voltages, and the transformed quantities (the result of the operation which the circuits implement) come out as voltages on wires or in registers.

Think ASIC's, and **hardware accelerators**.

Put a pin in these ideas.

General-purpose

We implement and integrate some subsystems. To oversimplify, these include:

Memory, to hold data and instructions.

A processing unit, to execute the instructions (transformations) that quantities in the instruction memory represent. These transformations act on data from elsewhere in memory, or from an input mechanism.

A bus, for moving data between subsystems.

Peripherals. These are special-purpose subsystems (a timer, a serializer/deserializer, a DMA channel, etc.) which do a limited number of things efficiently, and communicate with the CPU over the bus.

What is a computer?

Digital | **Analog** | Quantum | Natural

- Quantities are represented **physically** (i.e., with physics) by position, analog voltage, etc..
- If we'd like to model/predict the degrees of freedom of some system (tides, movement of planets, etc.), we build an ***analog****ous* analog computer which contains the same number of degrees of freedom, and the same equations for describing how those degrees of freedom change with time.
  - When you double a quantity in an analog computer, something inside the computer doubles!
- Analog computers use physics to understand one system with another system. They do so by **changing the units of the quantities being transformed**, but keeping those transformations identical between systems.
  - The position of a planet may be modeled as the voltage from an operational amplifier, or the angular position of a rotating gear.
- Analog computers tend to be **special purpose**, but they can be extremely fast and power efficient as compared to digital electronic computers.

What is a computer?

Digital | **Analog** | Quantum | Natural

How do you design an analog computer?

First:

Generate an understanding of the system which you'd like to model (what are the degrees of freedom, and what are the equations which describe how those degrees of freedom change with time?).

→

Then:

Build a separate system which contains the same number of degrees of freedom, and the same equations which describe how those degrees of freedom change with time.

You have control over this system! You can make it run faster, slower, forward, backward, and you can change parameter values.

What is a computer?

Digital | **Analog** | Quantum | Natural

How do you design an analog computer?

First:

Generate an understanding of the system which you'd like to model (what are the degrees of freedom, and what are the equations which describe how those degrees of freedom change with time?).

→

Then:

Build a separate system which contains the same number of degrees of freedom, and the same equations which describe how those degrees of freedom change with time.

You have control over this system! You can make it run faster, slower, forward, backward, and you can change parameter values.

Put a pin in this idea too.

What is a computer?

Digital | Analog | **Quantum** | Natural

- Quantities are represented **symbolically** by the probability amplitudes of finding a *qubit* in one of two possible quantum states. A qubit can exist in a superposition of these two states, but chooses one (based on those probability amplitudes) upon observation. We never observe the superposition, only the classical bit which results from the observation.

- Because all of our qubits are entangled (their probability amplitudes are correlated), each additional qubit **doubles** the dimension of the state space. The size of the state space for a quantum computer with 300 qubits is 2^{300}, greater than the number of atoms in the universe.

- Quantum computers transform the representations for these quantities (the probability amplitudes) by means of **quantum logic gates**, which are analogous to classical logic gates in digital computers.
  - These gates logically modify the probability amplitudes for the qubits, potentially conditioned on the probability amplitudes of other qubits.
  - Because an observation collapses these probability amplitudes, we defer observation to the *end* of the quantum computation. This observation collapses the system to classical bits, which we interpret as the output of our calculation.

- The power of these computers is **parallelism**. The transformations encoded by the quantum logic gates occur for all values in the (potentially massive) superposition state space in parallel. The resulting output encodes the transformation as applied to all values in the superposition. This collapses upon observation, suggesting that quantum computers do not *replace* other computers, but accelerate particular kinds of algorithms.

What is a computer?

Digital | Analog | **Quantum** | Natural

How do you use a quantum computer?

1. Initialize the probability amplitudes for all of your qubits to describe an initial superposition.
2. Setup your quantum logic gates such that they implement your algorithm of interest.
3. Allow for these quantum logic gates to transform your qubit probability amplitudes.
4. Measure the system, which collapses the qubits to classical bits according to their updated probability amplitudes.
5. Treat the resulting collection of classical bits as the result of your computation.

For example . . .

What is a computer?

Digital | Analog | **Quantum** | Natural

RSA public key N (product of two big prime numbers) → **Quantum computer** Find a number, r, which is likely to share a prime factor with N → Run Euclid's algorithm to find the greatest common factor between N and r, then use that factor (likely to be one of the prime factors of N) to find the other. → Now you have the RSA private key.

What is a computer?

Digital | Analog | **Quantum** | Natural

RSA public key N (product of two big prime numbers) → **Quantum computer**

Find a number, r, which is likely to share a prime factor with N → Run Euclid's algorithm to find the greatest common factor between N and r, then use that factor (likely to be one of the prime factors of N) to find the other. → uh-oh! Now you have the RSA private key.

What is a computer?

Digital Analog Quantum Natural

The quantum computer behaves like a **physics-based accelerator** for a digital electronic computer. By interfacing the digital electronic computer with a separate computing device which is based on *completely different* underlying mechanisms for computation, we improve the speed with which that digital electronic computer can compute certain algorithms.

This idea has been extended into . . .

- **Reservoir computing**: Use the intrinsic properties of a material (e.g. twisted magnets, ferroelectrics, memresistors, or a bucket of water) to do computation. This removes the separation between processing and memory units, and the computation being performed can be adjusted by adjusting the properties of the material.
    - Lee, Oscar, et al. "Task-adaptive physical reservoir computing." Nature Materials (2023): 1-9.
    - Duport, F., Schneider, B., Smerieri, A., Haelterman, M. & Massar, S. All-optical reservoir computing. Opt. Express 20, 22783–22795 (2012).
    - Grollier, J. et al. Neuromorphic spintronics. Nat. Electron. 3, 360–370 (2020).
    - Fernando, C. & Sojakka, S. Pattern recognition in a bucket. In Proc. ECAL 2003: Advances in Artificial Life (eds Banzhaf, W. et al.) 588–597 (Springer, 2003).
- **Physical computing:** Implement deep-learning accelerators that use physics (optics, mechanics, etc.) to generate neural networks.
    - Wright, Logan G., et al. "Deep physical neural networks trained with backpropagation." Nature 601.7894 (2022): 549-555. **Peter McMahon**
    - Lee, Ryan H., Erwin AB Mulder, and Jonathan B. Hopkins. "Mechanical neural networks: Architected materials that learn behaviors." Science Robotics 7.71 (2022): eabq7278.

But can we **find** these computers rather than **engineer** them?

What is a computer?

Digital | Analog | Quantum | **Natural**

In the near-term . . .

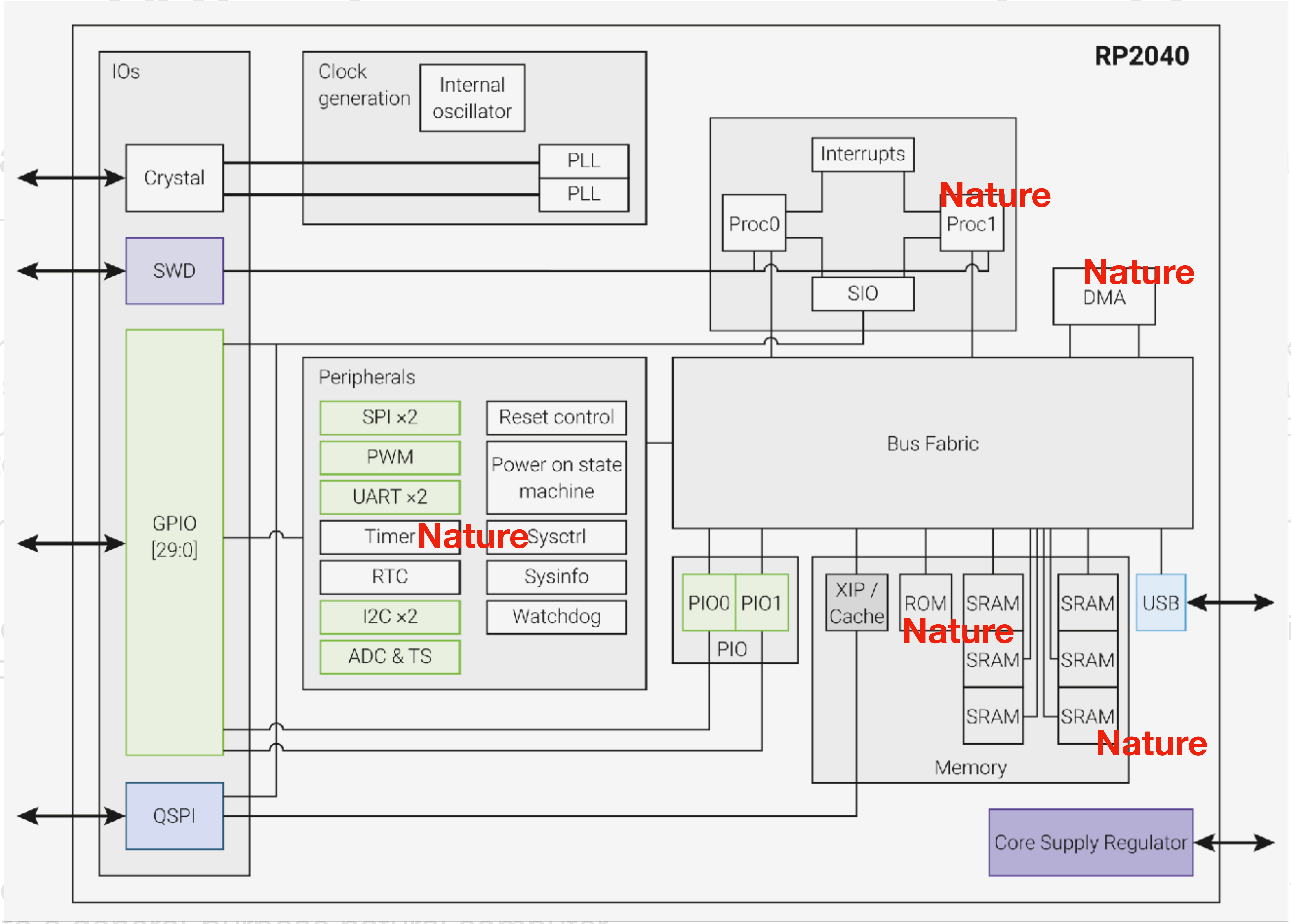
- Nature is replete with processes which transform one set of quantities into another. To the extent that these transformations are **useful**, then these are examples of **natural computation**. If we can interface these natural sources of computation with our digital electronic computers, they can be used as **natural accelerators** (as opposed to hardware accelerators).
- Nature is replete with **repositories for data**. If we can interface these natural repositories for data with our digital electronic computers, they can be used as **natural memory**.
- Nature offers mechanisms for **moving data**, *en masse*, from one place to another. If we can interface this natural movement of data with our digital electronic computers, they can be used for **natural data transfer.**

In the limit . . .

- We can piece together these natural accelerators, natural memory, and natural mechanisms for data transfer into a general-purpose natural computer.

Digit tural

- Nature is r xtent that these tran n interface these natu d as **natural accelerat**
- Nature is r data with our digital
- Nature offe interface this natural mo **l data transfer.**



- We can pi for data transfer into a general-purpose natural computer.

V. Hunter Adams

What is a computer?

Digital | Analog | Quantum | **Natural**

In the near-term . . .

- Nature is replete with processes which transform one set of quantities into another. To the extent that these transformations are **useful**, then these are examples of **natural computation**. If we can interface these natural sources of computation with our digital electronic computers, they can be used as **natural accelerators** (as opposed to hardware accelerators).
- Nature is replete with **repositories for data**. If we can interface these natural repositories for data with our digital electronic computers, they can be used as **natural memory**.
- Nature offers mechanisms for **moving data**, *en masse*, from one place to another. If we can interface this natural movement of data with our digital electronic computers, they can be used for **natural data transfer.**

Let us consider **natural computation**, **natural memory**, and **natural data transfer** in turn

In the limit . . .

- We can piece together these natural accelerators, natural memory, and natural mechanisms for data transfer into a general-purpose natural computer.

Nature and computing

Natural computation | Natural memory | Natural data transfer

How does one implement a natural computational accelerator?

1. Engage in **computational naturalism**, through which one searches in nature for examples of **algorithmic processes** through which quantities are being transformed into other quantities.
2. Develop a model for this system which describes that transformation of inputs to outputs, in detail or in a blackbox fashion.
3. Develop a system by which a digital electronic computer can affect the inputs to the system and observe the outputs, and then use that system as an **accelerator** for that particular algorithm.

Rather than **building** analog computers, we are **finding** analog computers, understanding the transformations that they perform, and plugging them into our digital electronic computers as accelerators. There is evidence that this works . . .

Nature and computing

Natural computation | Natural memory | Natural data transfer

How does one implement a natural computational accelerator?

1. Engage in **computational naturalism**, through which one searches in nature for examples of **algorithmic processes** through which quantities are being transformed into other quantities.

This relates to Wolfram's Principle of Computational Equivalence. Every process (from simple cellular automata, to physics, to brains) can be thought of as *computational* (transforming inputs to outputs). The Principle of Computational Equivalence states that, above a low threshold, all these processes correspond to computations of equivalent sophistication.

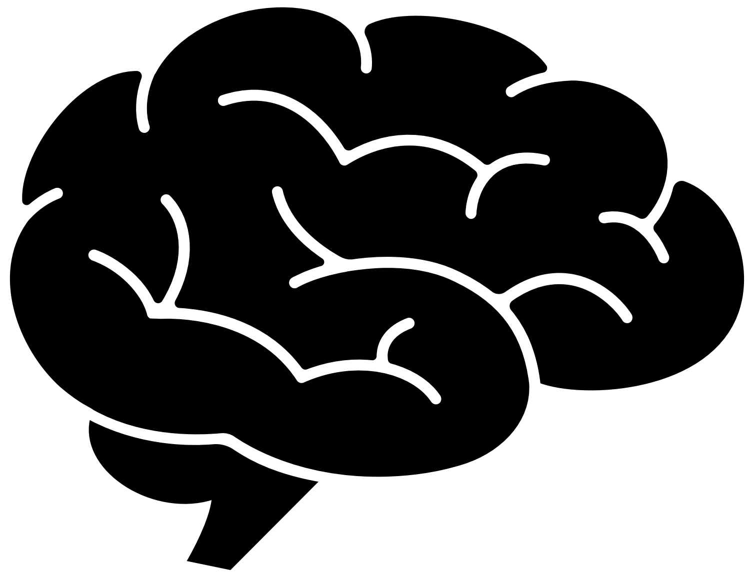
But can these computations be *used*?

Nature and computing

Natural computation | Natural memory | Natural data transfer

How do we know that this works, in principle?

1. Our **brains** are natural accelerators for our **digital electronic computers**.
2. Our brains are better than our computers at certain algorithms.
3. We interface our brains with digital electronic computers by means of keyboards, mice, screens, microphones, and speakers. Our computers can thereby use our brains as accelerators, offloading work to our brains by providing an input to the brain, and then prompting for the output from the brain.



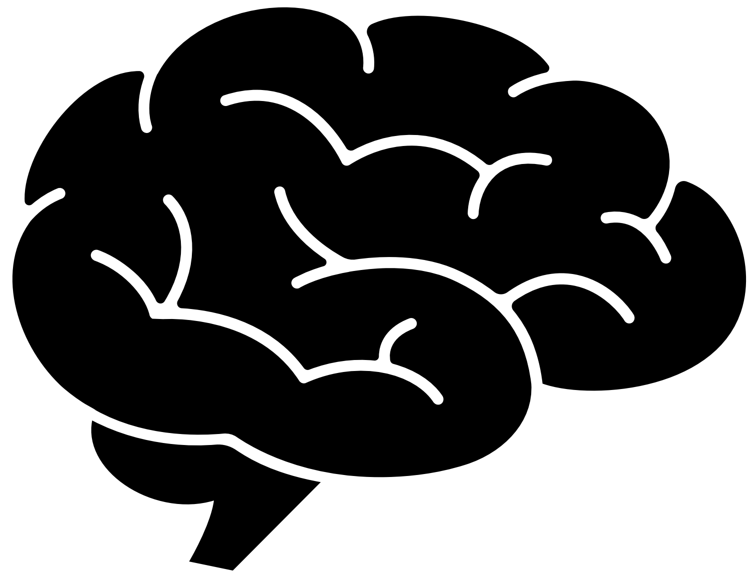
An example of a natural accelerator

Nature and computing

Natural computation | Natural memory | Natural data transfer

How do we know that this works, in principle?

1. Our **brains** are natural accelerators for our **digital electronic computers**.
2. Our brains are better than our computers at certain algorithms.
3. We interface our brains with digital electronic computers by means of keyboards, mice, screens, microphones, and speakers. Our computers can thereby use our brains as accelerators, offloading work to our brains by providing an input to the brain, and then prompting for the output from the brain.



An example of a natural accelerator

But there are more examples!

Nature and computing

Natural computation | Natural memory | Natural data transfer

Low-hanging fruit

1. Nature excels at **generating randomness**. We might develop a natural random number generator:
    - Use galactic cosmic rays (or high-energy particles from a small piece of radioactive material) as bus-masters. Build radiation-softened memory to increase the rate of single-event upsets in a section of memory, and use those single-event upsets as a source of entropy.
2. Nature offers **periodic processes**, which might be used as natural timer peripherals in event-driven systems.
    - Synchronization occurs all over the place in nature! Can we sync our computers with these natural oscillators?
    - Extend battery life in your IoT device by turning it **off** with a latching circuit (consumes ~0W), and using a stochastic natural process (wind blowing on a piezo, a bird pecking a vibration sensor, etc.) to wake the system at some rate.

Strogatz, Steven H., and Ian Stewart. "Coupled oscillators and biological synchronization." Scientific american 269.6 (1993): 102-109.

Nature and computing

Natural computation | Natural memory | Natural data transfer

Low-hanging fruit

1. Nature excels at **generating randomness**. We might develop a natural random number generator:
   - Use galactic cosmic rays (or high-energy particles from a small piece of radioactive material) as bus-masters. Build radiation-softened memory to increase the rate of single-event upsets in a section of memory, and use those single-event upsets as a source of entropy. Chris Batten, José Martinez
2. Nature offers **periodic processes**, which might be used as natural timer peripherals in event-driven systems.
   - Synchronization occurs all over the place in nature! Can we sync our computers with these natural oscillators? Steve Strogatz
   - Extend battery life in your IoT device by turning it **off** with a latching circuit (consumes ~0W), and using a stochastic natural process (wind blowing on a piezo, a bird pecking a vibration sensor, etc.) to wake the system at some rate.

Strogatz, Steven H., and Ian Stewart. "Coupled oscillators and biological synchronization." Scientific american 269.6 (1993): 102-109.

Nature and computing

Natural computation | Natural memory | Natural data transfer

Low-hanging does not mean low-impact!

Low-hanging fruit:

What fraction of total global compute is devoted to Monte Carlo analysis? What fraction of all of the instructions being executed each second by all the CPU’s on the planet is devoted to generating pseudorandom numbers? How much energy would be saved by utilizing natural random number generators?

1. Nature excels at **generating randomness**. We might develop a natural random number generator:
   - Use galactic cosmic rays (or high-energy particles from a small piece of radioactive material) as bus-masters. Build radiation-softened memory to increase the rate of single event upsets in a region of memory, and use those single event upsets as a source of entropy.
2. Nature offers **periodic processes**, which might be used as natural timer peripherals in event-driven systems.
   - Extend battery life in your IoT device by turning it **off** with a latching circuit (consumes ~0W), and using a stochastic natural process (wind blowing on a piezo, a bird pecking a vibration sensor, etc.) to wake the system at some rate.

Nature and computing

Natural computation | Natural memory | Natural data transfer

Low-hanging does not mean low-impact!

Low-hanging fruit:

What fraction of total global compute is devoted to Monte Carlo analysis? What fraction of all of the instructions being executed each second by all the CPU’s on the planet is devoted to generating pseudorandom numbers? How much energy would be saved by utilizing natural random number generators?

1. Nature excels at **generating randomness**. We might develop a natural random number generator:
   - Use galactic cosmic rays (or high-energy particles from a small piece of radioactive material) as bus-masters. Build radiation-softened memory to increase the rate of single event upsets in a region of memory, and use those single event upsets as a source of entropy.
2. Nature offers **periodic processes**, which might be used as natural timer peripherals in event-driven systems.
   - Extend battery life in your IoT device by turning it **off** with a latching circuit (consumes ~0W), and using a stochastic natural process (wind blowing on a piezo, a bird pecking a vibration sensor, etc.) to wake the system at some rate.

Eilyan Bitar

Nature and computing

Natural computation | Natural memory | Natural data transfer

Long-term possibilities . . .

1. Nature excels at **parallel computation**:
   - Though it take tremendous computational effort to do things like integrate the Navier-Stokes equations on an FPGA, nature **just does it**. For some computational physics experiments, we may be able to substitute sensors/actuators for parallelized computers. This resurrects a version of analog computation in which the analog computer is **already built.**
   - We aren't building an engineered analog for a process of interest. We're instead identifying natural computational processes, and looking for analogous computational problems which it might be used to solve.

2. Nature excels at **optimization**.
   - Can we find any examples of natural systems which are solving the Knapsack, or Traveling Salesman, or other NP hard problems? Can those natural systems be influenced to solve a **version** of that problem which is of interest to us?
   - A cliché example is that of a slime mold re-creating the Japanese rail system.

3. We've yet to fully realize the potential of the **brain as natural accelerator**.
   - With better input/output between brain and digital electronic computer, or better understanding of the brain's API, we may better realize the computational potential of the brain.

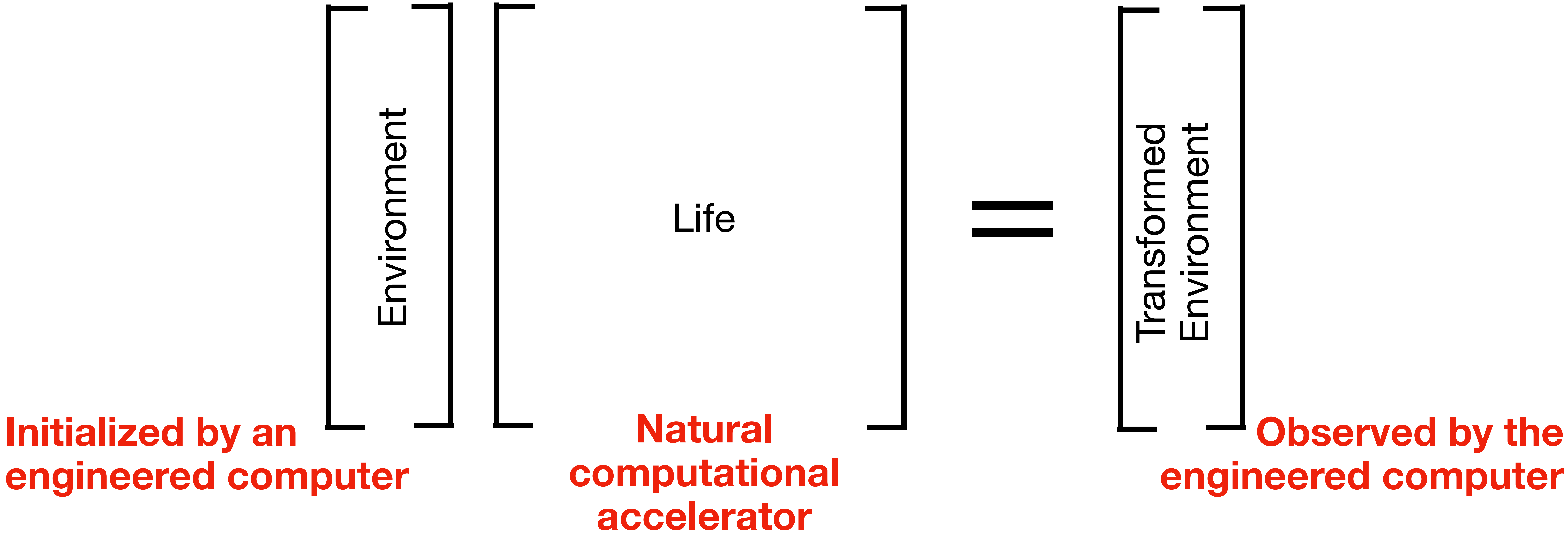
Nature and computing

Natural computation | Natural memory | Natural data transfer

More long-term possibilities . . .

The environment is a quantifiable system, which **life** transforms.

Linearize life about a moment in time, and it is a matrix which acts upon a long vector containing all of its environment's degrees of freedom. If those transformation are useful, then that life is a natural computational accelerator.



Nature and computing

Natural computation | Natural memory | Natural data transfer

More long-term possibilities . . .

The environment is a quantifiable system, which **life** transforms.

Linearize life about a moment in time, and it is a matrix which acts upon a long vector containing all of its environment's degrees of freedom. If those transformation are useful, then that life is a natural computational accelerator.

$$\begin{bmatrix} \text{Environment} \end{bmatrix} \begin{bmatrix} \text{Life} \end{bmatrix} = \begin{bmatrix} \text{Transformed Environment} \end{bmatrix}$$

Kirstin Petersen, Nils Napp

Initialized by an engineered computer

Natural computational accelerator

Observed by the engineered computer

V. Hunter Adams

Nature and computing

Natural computation | Natural memory | Natural data transfer

Long-term possibilities . . .

This takes advantage of the fact that nature reuses her mathematical models. One natural system can be used to model another, or to model a non-natural system of human interest.

1. Nature excels at **parallel computation**.
   - Though it take tremendous computational effort to do things like integrate the Navier-Stokes equations on an FPGA, nature **just does it**. For some computational physics experiments, we may be able to substitute natural accelerators for parallelized computers. This resurrects a version of analog computation in which the analog computer is **already built.**
   - We aren't building an engineered analog for a process of interest. We're instead identifying natural computational processes, and looking for analogous computational problems which it might be used to solve.

2. Nature excels at **optimization**.
   - Can we find any examples of natural systems which are solving the Knapsack, or Traveling Salesman, or other NP hard problems? Can those natural systems be influenced to solve a **version** of that problem which is of interest to us?
   - A cliché example is that of a slime mold re-creating the Japanese rail system.

3. We've yet to fully realize the potential of the **brain as natural accelerator**.
   - With better input/output between brain and digital electronic computer, or better understanding of the brain's API, we may better realize the computational potential of the brain.

Nature and computing

Natural computation | Natural memory | Natural data transfer

Problems to solve

1. Input/output between natural and engineered computers:
   - For a natural accelerator to be useful, the total time required to communicate inputs to the accelerator, compute outputs, and communicate those outputs back to the engineered computer with which it interfaces must be **less** than the time to compute the same algorithm on the engineered computer. We need high-speed I/O between nature and machine. **Amit Lal**
   - We must be able to affect the inputs to the natural system in a controlled fashion. This is easy for some systems, and very hard for others. **Amal El-Ghazaly**
2. Generating models for natural computers:
   - In order for a natural computer to be of use, we must understand the relationship between its inputs and outputs. For some systems, this is easy. For others, it's extremely difficult.

Nature and computing

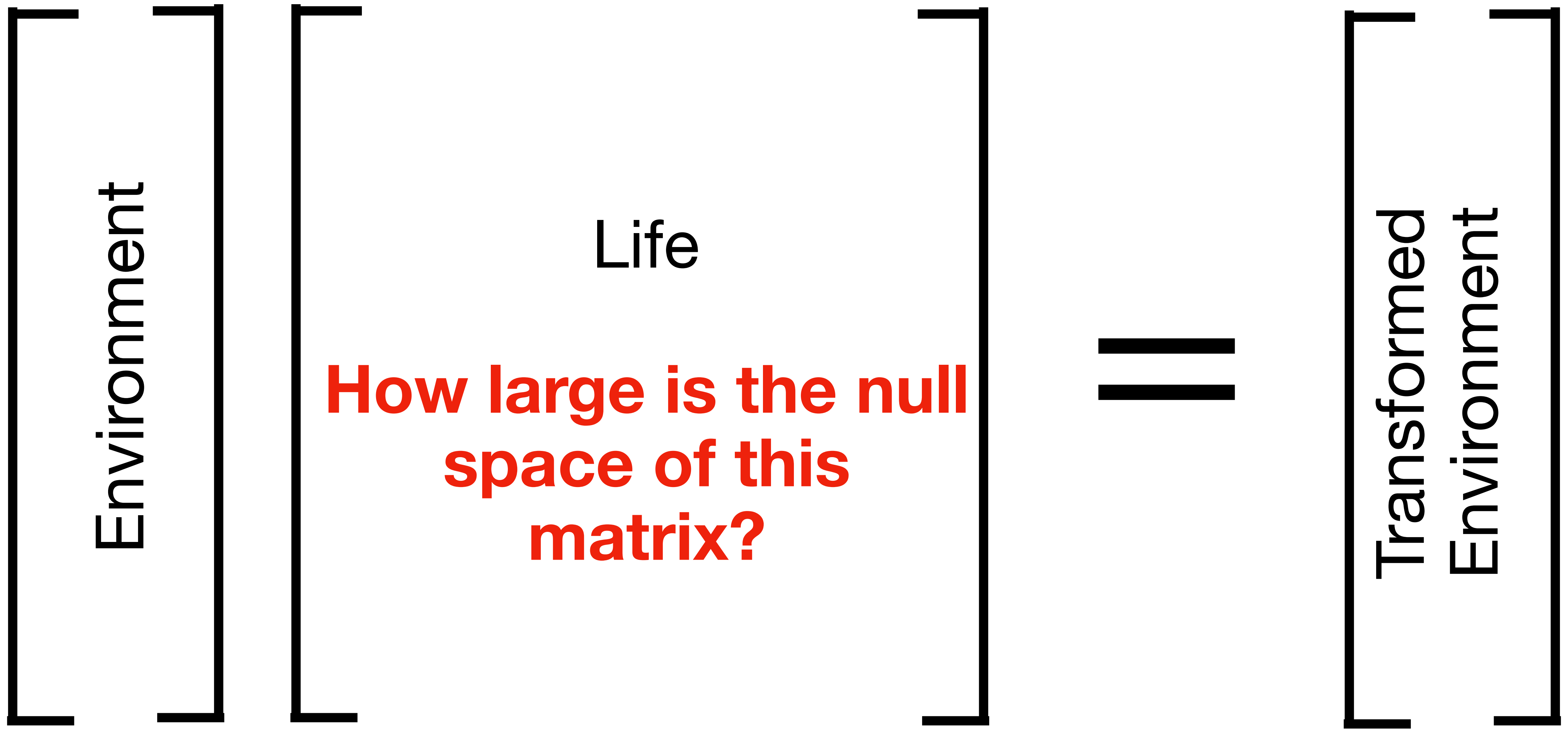
Natural computation | **Natural memory** | Natural data transfer

Nature offers repositories for data in *natural null spaces.*

These are environments with degrees of freedom which are effectively invisible to the organisms which inhabit those places. These invisible degrees of freedom can be used for information storage without adversely affecting any local life.

If interfaced with an engineered computer, these repositories for data can be used as **memory**. Some of these natural repositories have attractive features, including extreme non-volatility or extreme capacity.

Let us consider some specific examples of three varieties: **proof of concept**, **of niche application**, and **far-out**.



V. Hunter Adams

Nature and computing

Natural computation | **Natural memory** | Natural data transfer

Proof-of-concept examples of natural memory

Tree rings store information about past environmental conditions that the tree experienced. If that tree existed in a greenhouse, and if the environmental conditions of that greenhouse were modulated by some other data source (e.g. temperature/humidity controlled by GDP of the USA), then those tree rings instead encode that other data source (low-passed to ~1 datapoint/year).

Or if all environmental conditions are held constant *except* for water, then the tree rings encode the level of responsibility of the person responsible for watering the tree.

Ice cores encode information about past atmospheric conditions and composition. If the atmospheric condition/composition above a section of ice were modulated by some other data source, then an ice core at that place would similarly encode that other data source.

Very non-volatile! But not much data.

We might store data in these tree rings . . .

Tree ring image from Wikipedia

Nature and computing

Natural computation | **Natural memory** | Natural data transfer

An example of natural memory with niche utility

Paleomagnetism is the study of the history of the Earth's magnetic field by means of the magnetic moments of volcanic rock. Lava contains ferromagnetic crystals that align themselves with the direction of the magnetic field as the lava cools. Once the lava gets below a certain temperature, the magnetic moments become locked into the rock, recording the direction and intensity of the local magnetic field at the time that the lava cooled.

If we built a device which quickly actuated the magnetic field of cooling lava as it moved under the device, we could record data in the magnetic moment of the rock. Potentially, a lot of data. These data could then be retrieved by means of a magnetometer which traversed that section of rock.

This is **super non-volatile**, and offers a bit more data storage!


Rapidly actuated
electromagnet
Data encoded as
magnetic moment

Nature and computing

Natural computation | **Natural memory** | Natural data transfer

Far-out examples of natural memory

Natural systems contain an unbelievable amount of **state**. How long would the vector be which fully specifies the state of a cubic meter of beach? Or the molecular state of a mineral? Or which specifies all degrees of freedom in a coral reef?

Life is a matrix which acts on these vectors. We are free to store data (a lot of data) in the null space of this matrix without having any effect on local life. It is worth noting that every degree of freedom is available for information storage in environments that contain no life, like the Moon and asteroids. Tantalizingly, some of these repositories for data *move*.

For some of these, entropy may present a problem.

If only we can figure out how to read/write it, we can use nature's huge state space to store information.

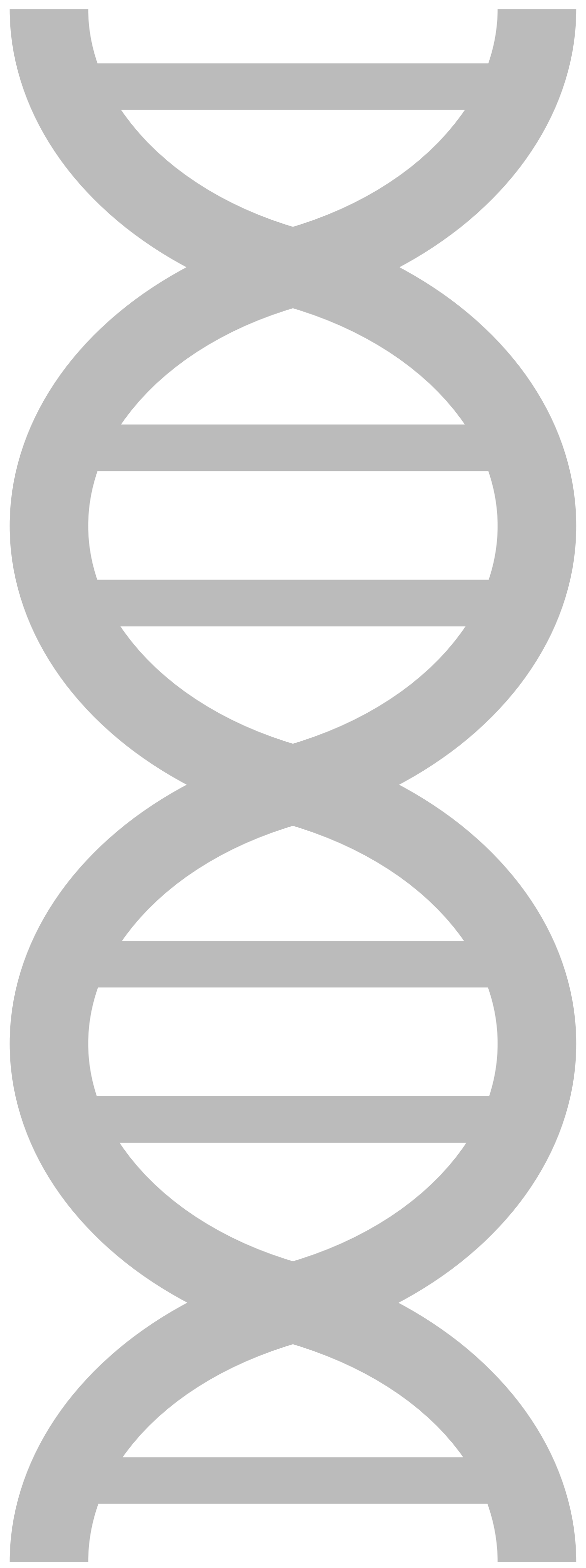
Nature and computing

Natural computation | **Natural memory** | Natural data transfer

There's reason to believe this is possible!

Data *can* be written and read to/from DNA . . .

- Goldman, N., Bertone, P., Chen, S., Dessimoz, C., LeProust, E. M., Sipos, B., ... & Birney, E. (2013). Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. Nature, 494(7435), 77-80.
- Church, George M., Yuan Gao, and Sriram Kosuri. "Next-generation digital information storage in DNA." Science 337.6102 (2012): 1628-1628.
- Shipman, Seth L., et al. "CRISPR–Cas encoding of a digital movie into the genomes of a population of living bacteria." Nature 547.7663 (2017): 345-349.
- Erlich, Yaniv, and Dina Zielinski. "DNA Fountain enables a robust and efficient storage architecture." science 355.6328 (2017): 950-954.



Nature and computing

Natural computation | Natural memory | **Natural data transfer**

Some observations . . .

1. Nature is **really good** at moving huge amounts of matter over huge distances. Rivers, ocean currents, the jet stream, the water/carbon cycles, planets and moons, etc.
2. These are tremendously **energetic** processes, and humanity has a long history of extracting small amounts of energy from some of these processes for use on other work. Water wheels, hydroelectric generators, wind farms, etc.
3. Humanity has **underutilized** these processes for data transfer. By moving huge amounts of matter, **nature is also moving huge amounts of data.**

Back of the envelope calculations . . .

Nature and computing

Natural computation | Natural memory | **Natural data transfer**

Thinking about moving **matter** as moving **information** . . .

There are 1.26×10^{26} molecules of water in one gallon. Each of these molecules has a position and an orientation (6 degrees of freedom).

700,000 gallons of water flow over Niagara Falls every second. That's 5.3×10^{32} degrees of freedom. That's over one zettabyte (10^{23}) of information.

700,000 gallons of water flow over Niagara Falls every second. That's 5.3×10^{32} degrees of freedom. That's over one **zettabyte** (10^{23}) of information.

What is the information channel capacity of the Nile? Or the Jet Stream?

Nature and computing

Natural computation | Natural memory | **Natural data transfer**

It is just as ridiculous to imagine that one could transfer this much data by waterfall as it is to imagine that one could extract all the energy from a waterfall. But perhaps it's not ridiculous to imagine that we might *sip* the total channel capacity of these systems, much like we *sip* the total energy capacity of these systems.

And there are some nearer-term opportunities that can precede these longer-term possibilities . . .

Nature and computing

Natural computation | Natural memory | **Natural data transfer**

Near-term opportunities

- We can add matter, in the form of conventional mass storage devices, to these systems such that they are “swept along” from origin to destination
- This makes the communication channel “bursty” in the sense that the information arrives all at once rather than bit-by-bit. But it enables very large **average data transfer rates**.
- As a silly example, the data rate of a single pigeon carrying 1TB SD cards from NYC to Boston is ~3GB/s. Other systems (ocean currents, trade winds, etc.) possess *way* more capacity for excess mass, and thus allow even larger average data transfer rates.

Long-term possibilities

- We encode the data *in the matter itself*. How do we read/write this matter efficiently? I’m not sure yet.

How does this start?

Where does it lead?

Nature and computing

How this starts | Where it goes

Near-term opportunities

- Developing natural computers will occur in phases, the first of which resurrects the naturalist of previous centuries. Like those naturalists, the natural computing researcher will go into the wilderness to look for new things. Rather than searching for new plants and animals, this person seeks natural computational processes, natural repositories for data, and natural migrations of matter.

- These processes will be studied and modeled to gain an understanding of their algorithmic qualities, and then the researcher will design and build devices which make these evolved systems components of a larger machine.

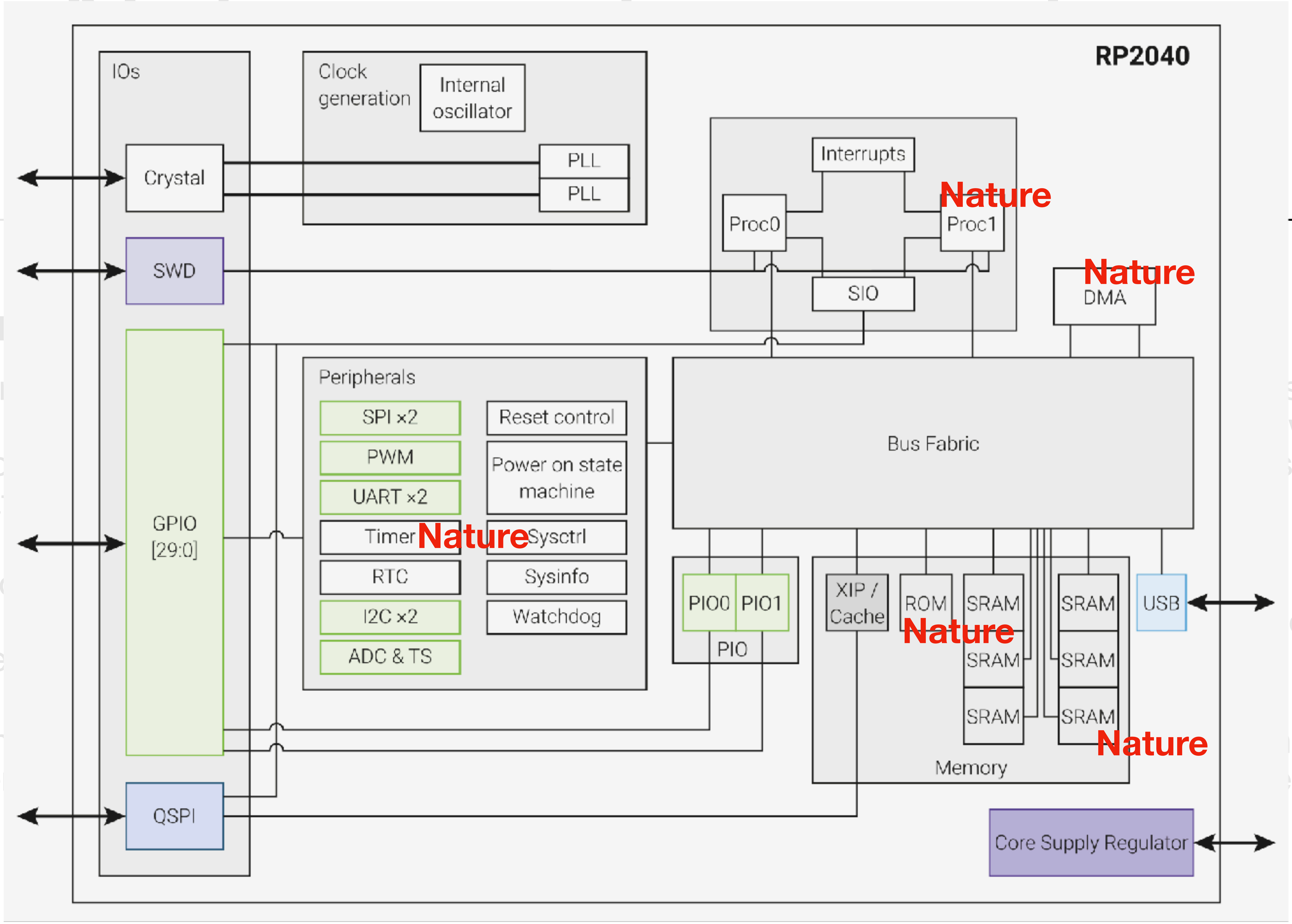
- Development for natural computers will proceed similarly to the development of conventional computers. Like the original ENIAC and its predecessors, the first natural computers will be special-purpose.

Near-ter

- Developi ... st of
previous ... wilderness
to look fo ... natural
computat

- These pr ... qualities,
and then ... omponents
of a large

- Developr ... al
compute ... special-
purpose.



Nature and computing

How this starts | **Where it goes**

To what world does this lead?

- The incorporation of these natural processes into computing machine will **incentivize their preservation.** Natural places will be preserved for their computational utility and potential.

- Unlocking the latent computational potential of nature will **change the world** in much the same way that unlocking nature's latent potential for power production changed the world. Total global compute will radically increase.


Photo by Vlad Hilitanu on Unsplash
Preserved for natural computation