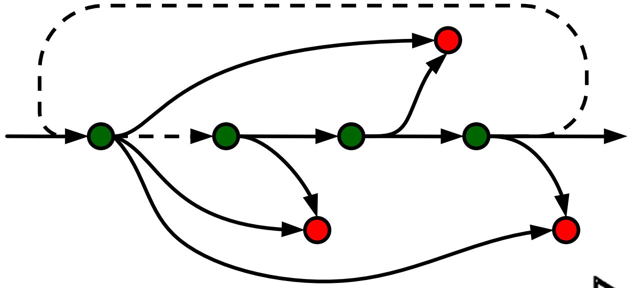
Logic Synthesis for DNA Computing

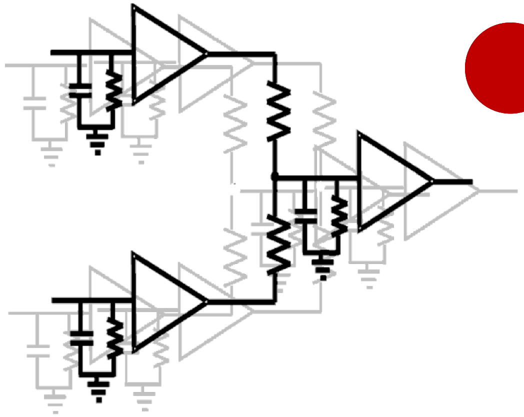
Marc Riedel

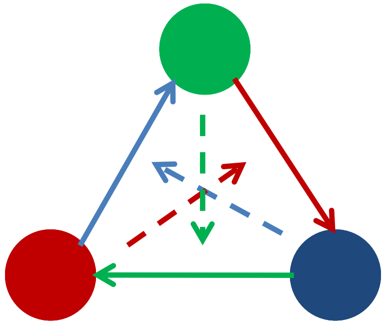
Associate Professor, Electrical and Computer Engineering

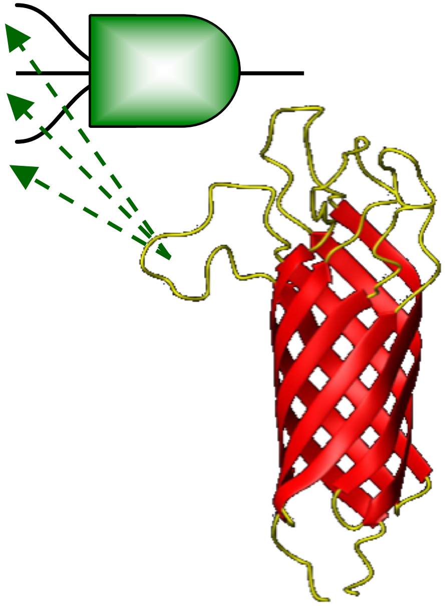
Graduate Faculty, Biomedical Informatics and Computational Biology

University of Minnesota







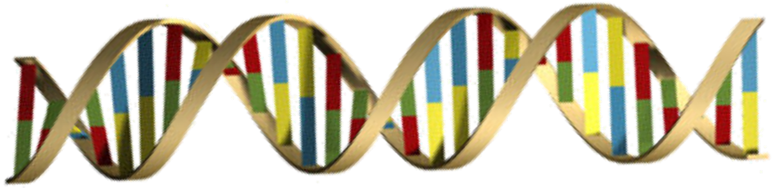


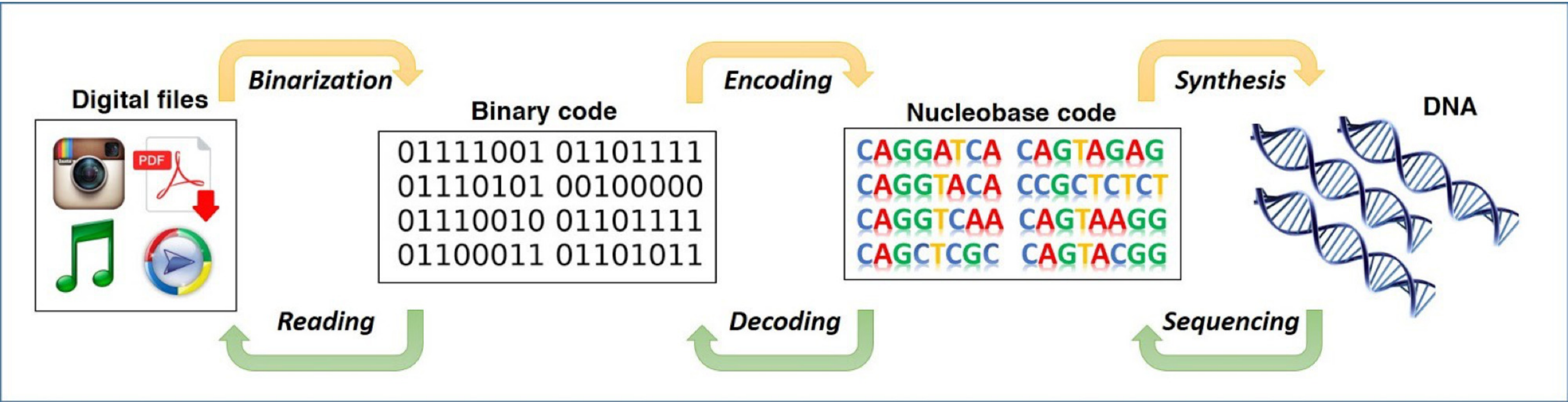
DATE Workshop – Florence, March 29, 2019

DNA Storage

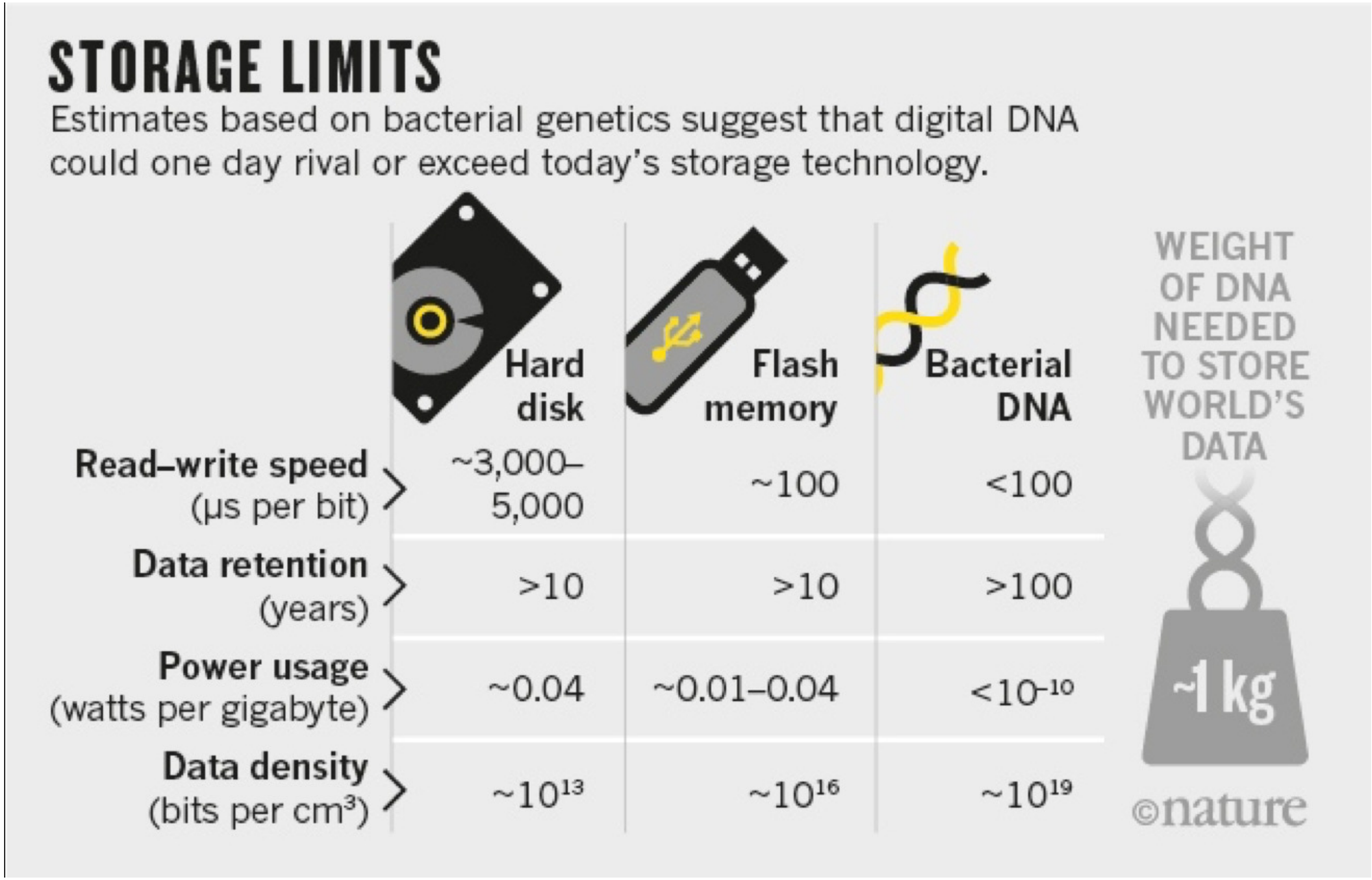
Nucleotides: $\{A, C, T, G\}$

DNA: string of nucleotides





DNA Storage: 200 Petabytes per gram

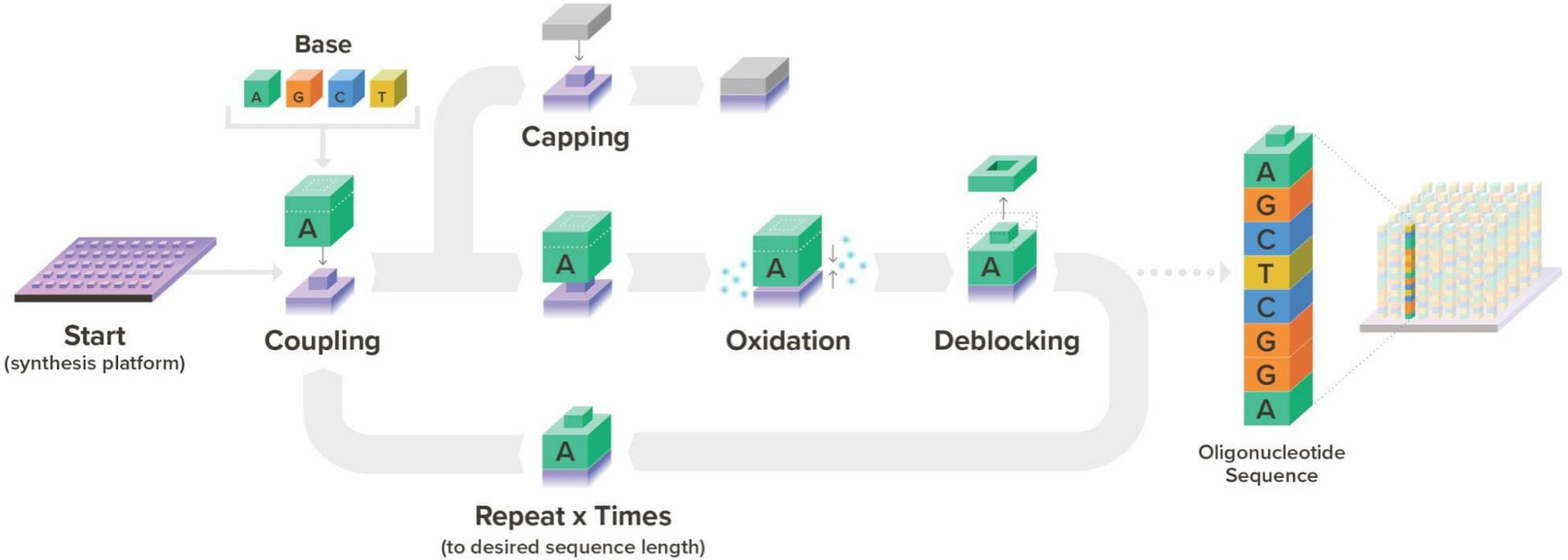


STORAGE LIMITS

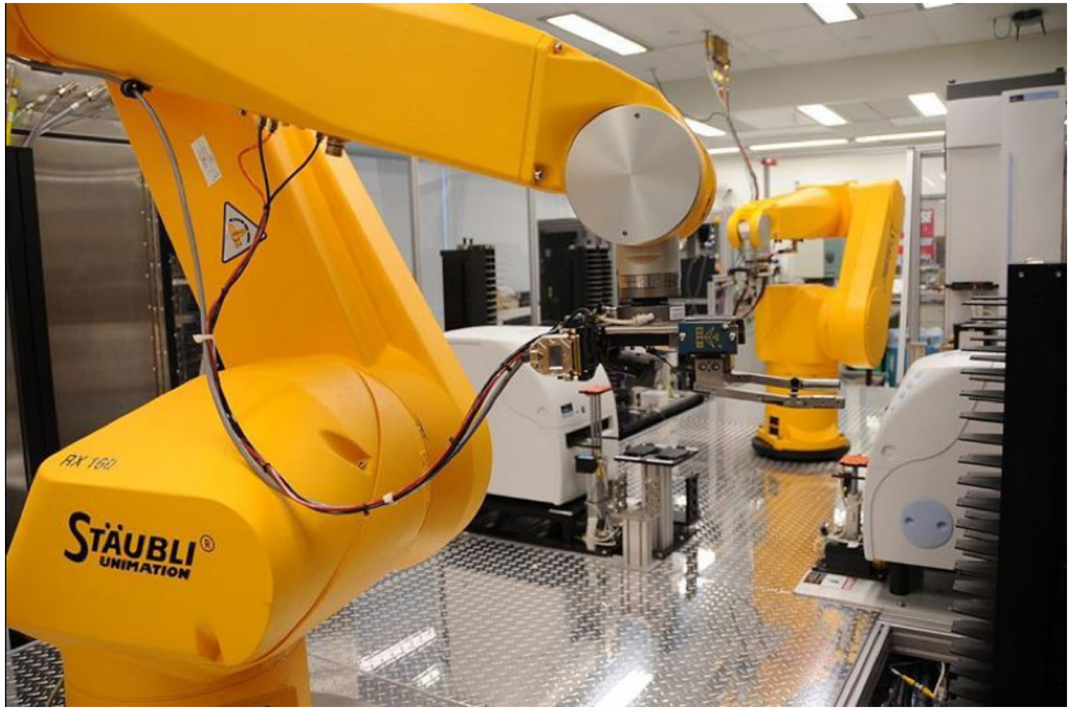
Estimates based on bacterial genetics suggest that digital DNA could one day rival or exceed today's storage technology.

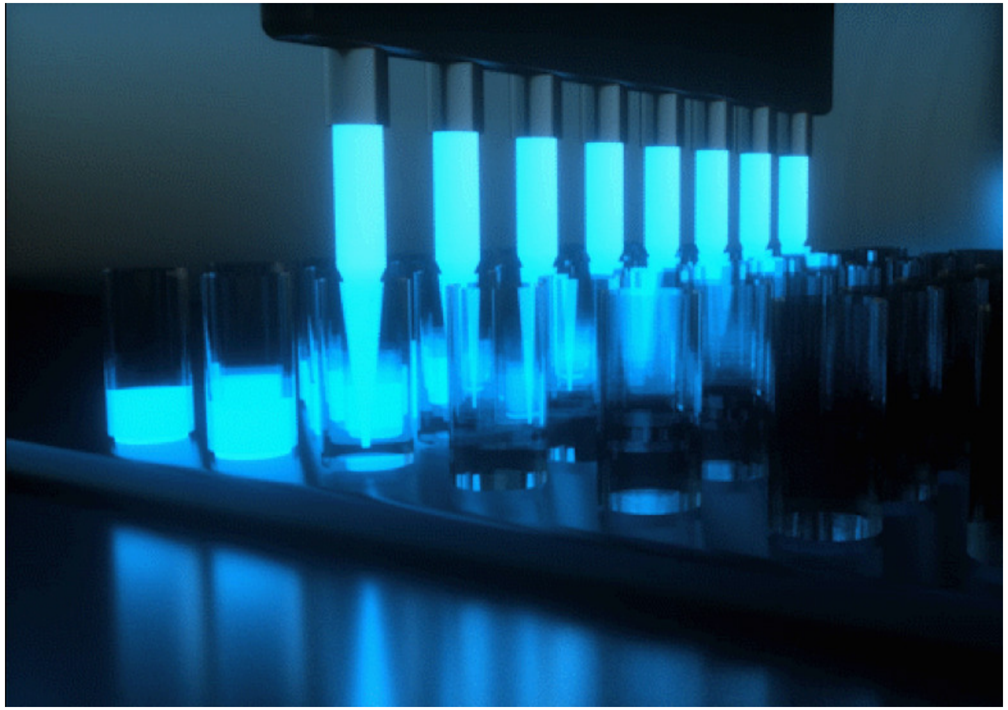
	Hard disk	Flash memory	Bacterial DNA
Read–write speed (μs per bit)	~3,000–5,000	~100	<100
Data retention (years)	>10	>10	>100
Power usage (watts per gigabyte)	~0.04	~0.01–0.04	$<10^{-10}$
Data density (bits per cm³)	$\sim10^{13}$	$\sim10^{16}$	$\sim10^{19}$

DNA Synthesis



DNA Synthesis

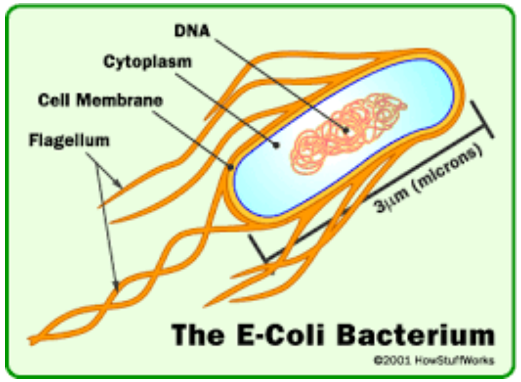


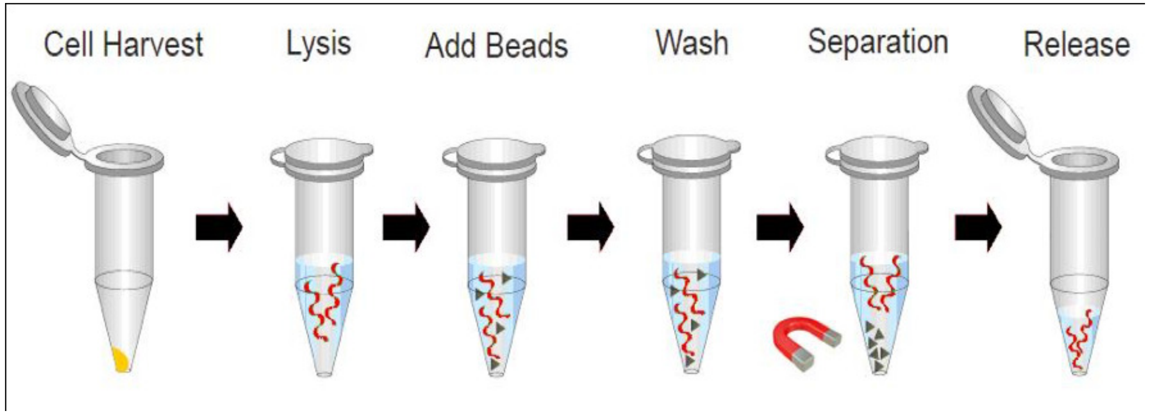


Synthesis rate: few bytes per minute.
Synthesis cost: $1000’s per kilobyte.

Our Approach: Use Existing Native DNA

The storage medium: E. coli K-12



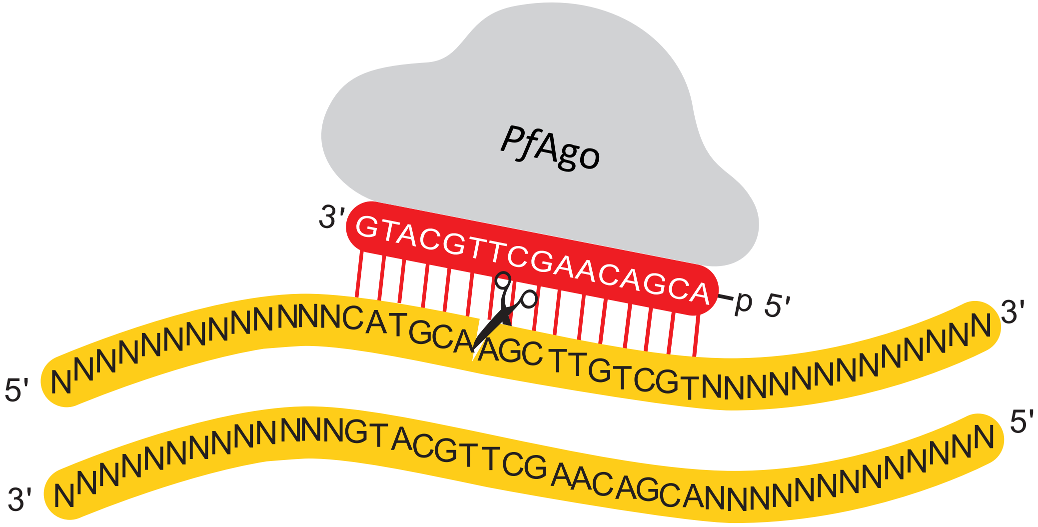


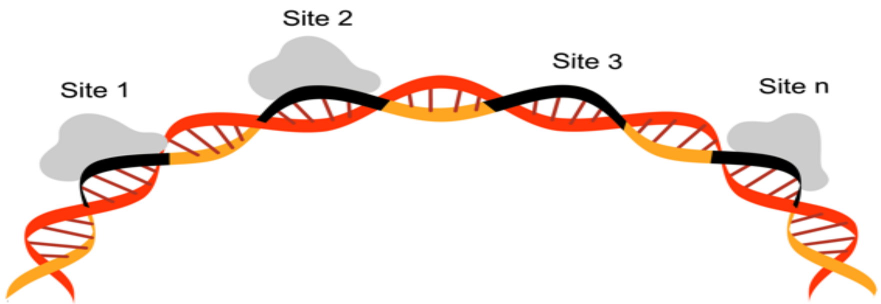
Fixed sequence of A, C, T, G's – so nothing is stored!

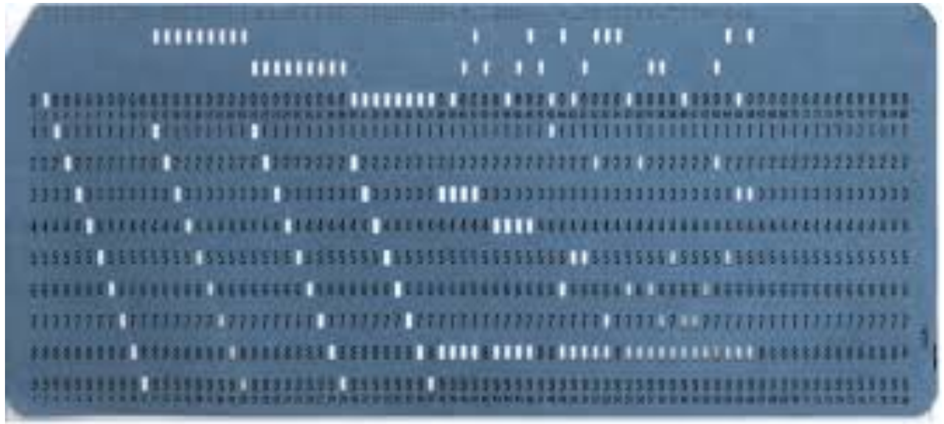
Synthesis rate: Megabytes per second.
Synthesis cost: $1 per megabyte (or less).

Our Storage Modality: “Nicks”

Gene editing with CRISPR/Cas9 or PfAgo







A cut represents a 1; absence of a cut a 0.

Our Storage Modality: “Nicks”

Gene editing with CRISPR/Cas9 or PfAgo

- Use multiple turnover nickase (one molecule can create ~50 nicks)
- Can create multiple nicks in parallel.
- Separate DNA into different wells; nick independently.

↔

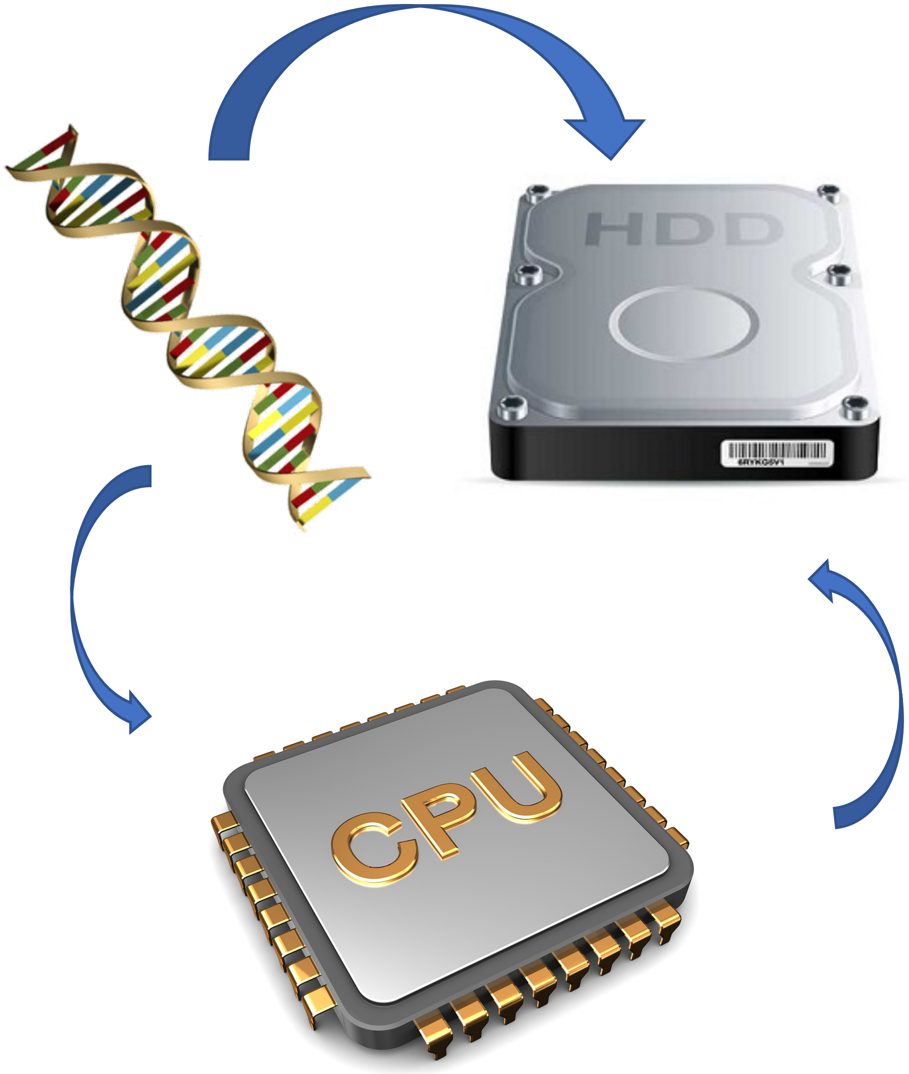
Computation

Objectives:

- Leverage the high-density of storage with effective computation.
- Perform "computation in memory" to reduce I/O operations.
- Integrate storage with data-intensive algorithms, such as machine learning.

Motivation:

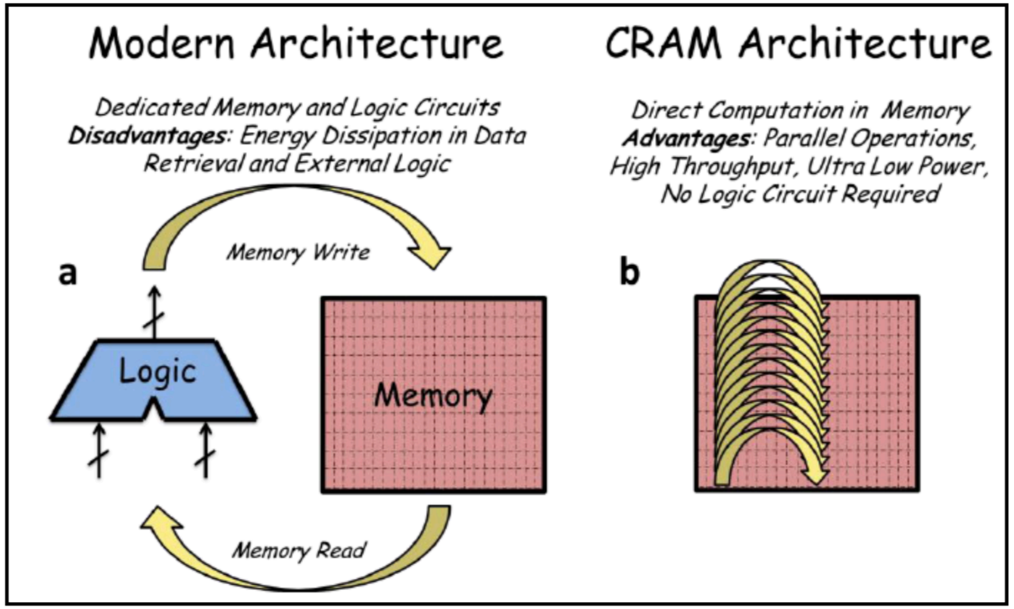
- While DNA storage might achieve densities of 100's of petabytes/gram, the I/O operations are slow.
- Techniques such as data aggregation and "computation-in-memory" could reduce the I/O requirements.
- The paradigm might be most effective for applications that generate large volumes of static data.



In-Memory Computing

"In-memory computing" or "computational memory" is an emerging paradigm that exploits the physical properties of memory devices for both storing and processing information. (Contrast with von Neumann systems which shuttle data back and forth between memory and the computing unit.)

- Instead of viewing memory as a place where we merely store information, can exploit the physics of DNA storage to implement high-level computational primitives.
- The result of the computation is also stored in the memory devices.
- Concept is loosely analogous to by how the brain computes.



Concepts Needed

1. How to compute functions with stochastic logic:
2. How to implement stochastic logic with DNA strands:
   encode as fractional concentrations.
3. How to obtain DNA strands from DNA complexes with "nicks":
   concept of probes.
4. How to transform the DNA strands:
   with strand displacement cascades.
5. How to scale the concentration of DNA strands:
   with *competitive* strand displacement.

Objectives

Demonstrate "in-memory" computation of non-trivial, interesting functions.

1. First exhibit simple computational primitives: multiplication and inversion.
2. Next, develop a methodology to implement polynomials.
3. Finally, develop a method to implement non-polynomial functions via polynomial approximations.

$$x\,x = x^2,\;\; 1 - x$$

$$P_1(x) = a_0 + a_1 x = 1 - (1 - a_0)(1 - \frac{a_1}{(1-a_0)}x).$$

Using Maclaurin Series Expansion

$$e^{-x} = \sum_{n=0}^{\infty} \frac{(-x)^n}{n!} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \frac{x^5}{5!}$$

Applying Horner's Rule

$$e^{-x} = 1 - x(1 - \frac{x}{2}(1 - \frac{x}{3}(1 - \frac{x}{4}\left(1 - \frac{x}{5}\right))))$$

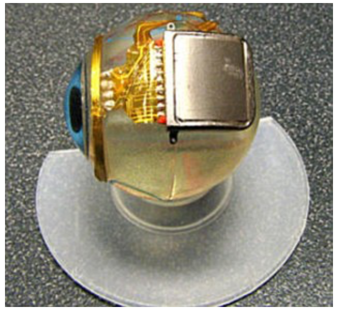
Concept 1:
Stochastic Logic

Stochastic Logic

A real value x in [0, 1] is represented by a sequence of random bits, each of which has probability x of being one and probability of $1 - x$ of being zero.

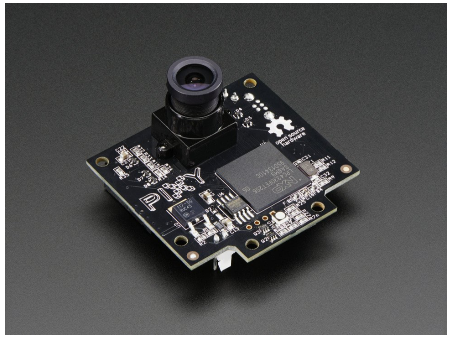
$$\overbrace{\ldots,0,1,0,1,1,0,0,1,0,0,0,1,1,1,0,0,\ldots}^{x = 7/16}$$

Permits complex mathematical functions to be implemented with very few transistors: compared to conventional design methodologies, reduces area by 95% to 98%.



Implantable Biomedical Devices, Harvesting Energy from Movement

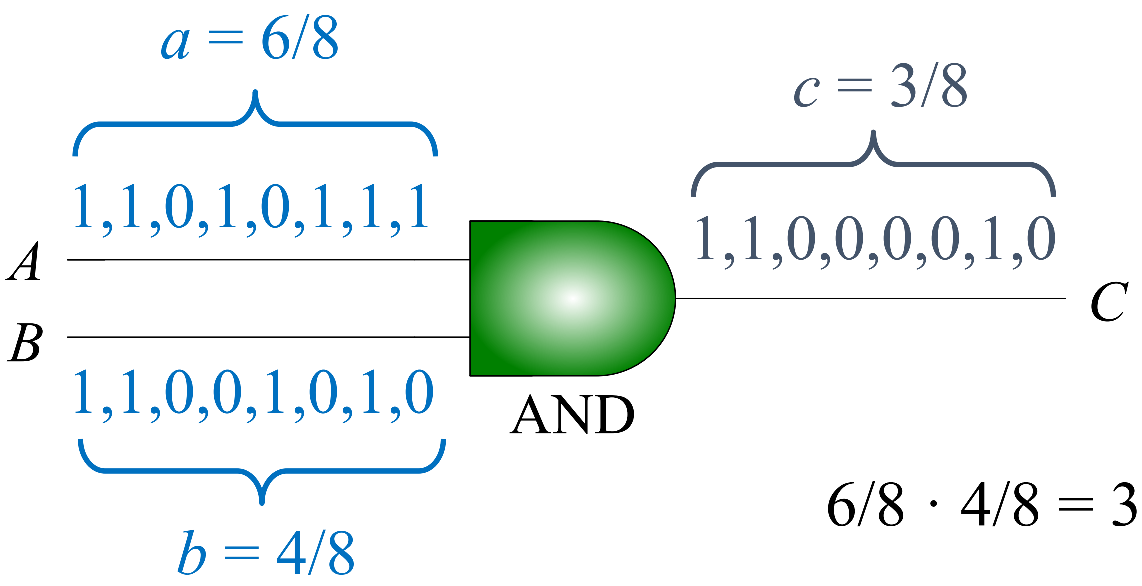
Insect-sized UAVs, Harvesting Energy from Small Solar Cells



Ultra Low Power Digital Circuitry for Communications and Image/Video Processing

Fractional Encodings

Computes on probabilities, or equivalently, fractions.

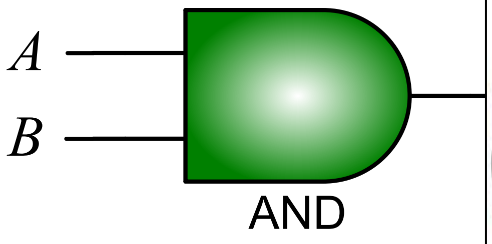


$$6/8 \cdot 4/8 = 3/8$$

Assume two input bit streams are independent

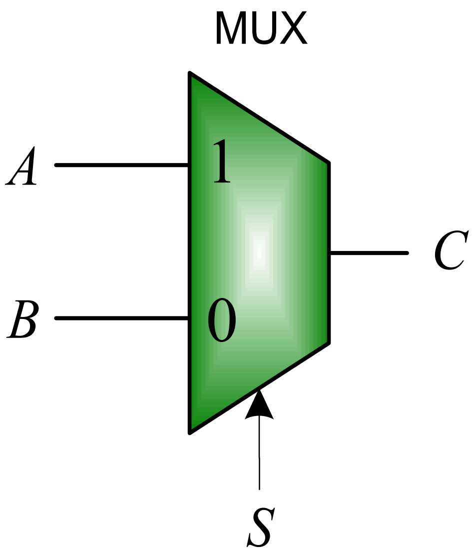
Arithmetic Operations

Multiplication



$$c = P(C) = P(A)P(B) = ab$$

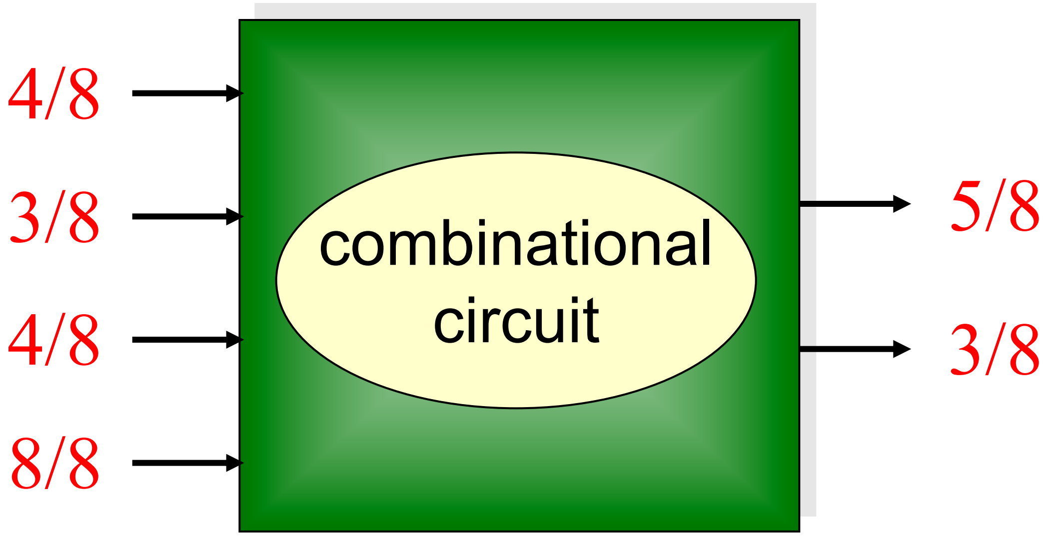
(Scaled) Addition



$$(C) = P(S)P(A) + [1 - P(S)]P(B) = s\,a + (1-s)b$$

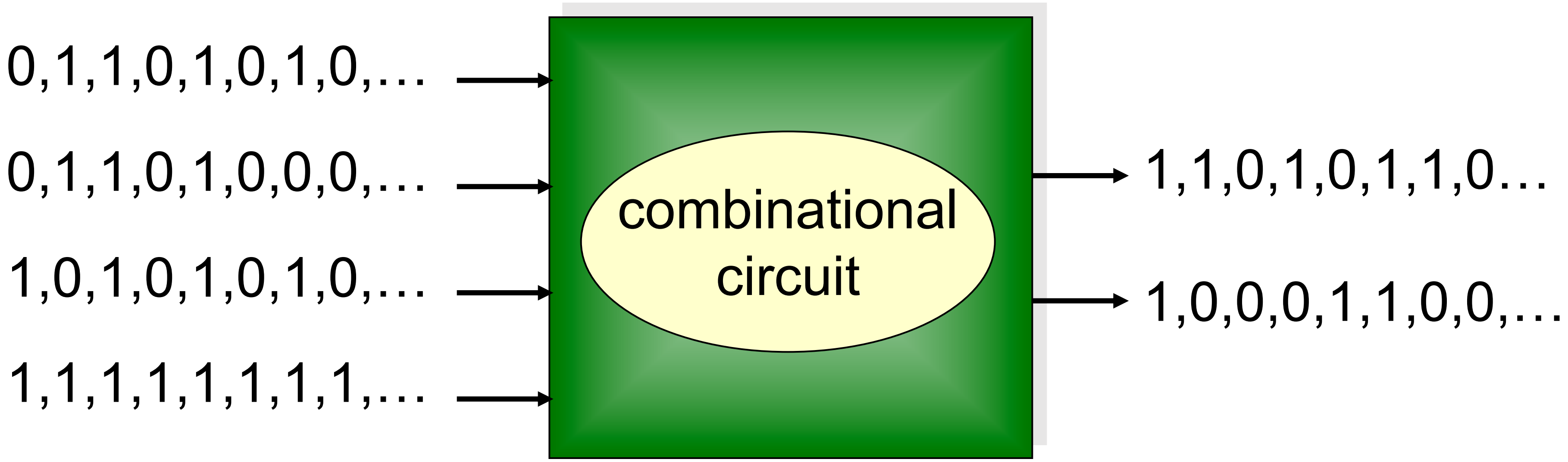
Stochastic Logic

Probability values are the input and output signals.



Stochastic Logic

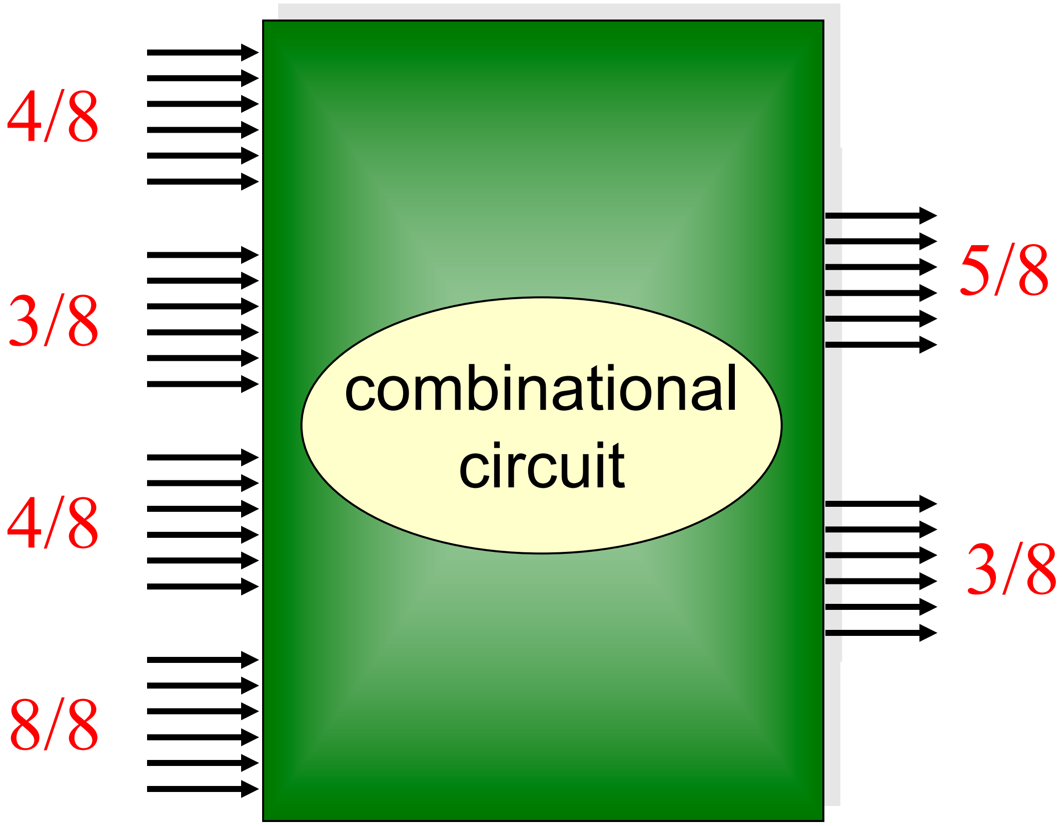
Probability values are the input and output signals.



serial bit streams

Stochastic Logic

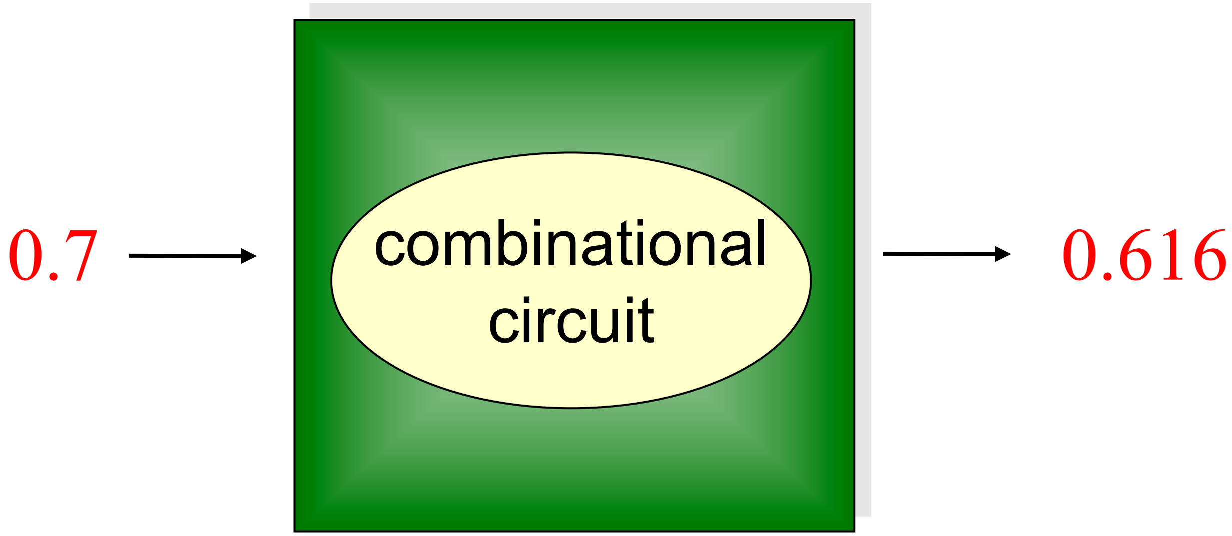
Probability values are the input and output signals.



parallel bit streams

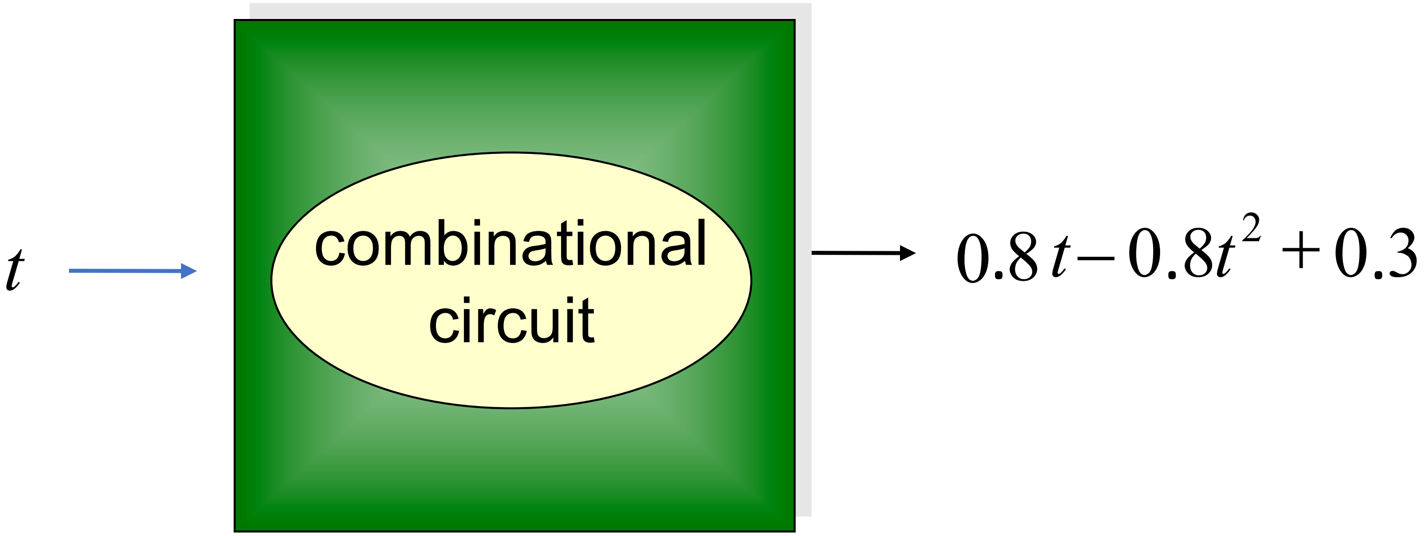
Stochastic Logic

Probability values are the input and output signals.



Stochastic Logic

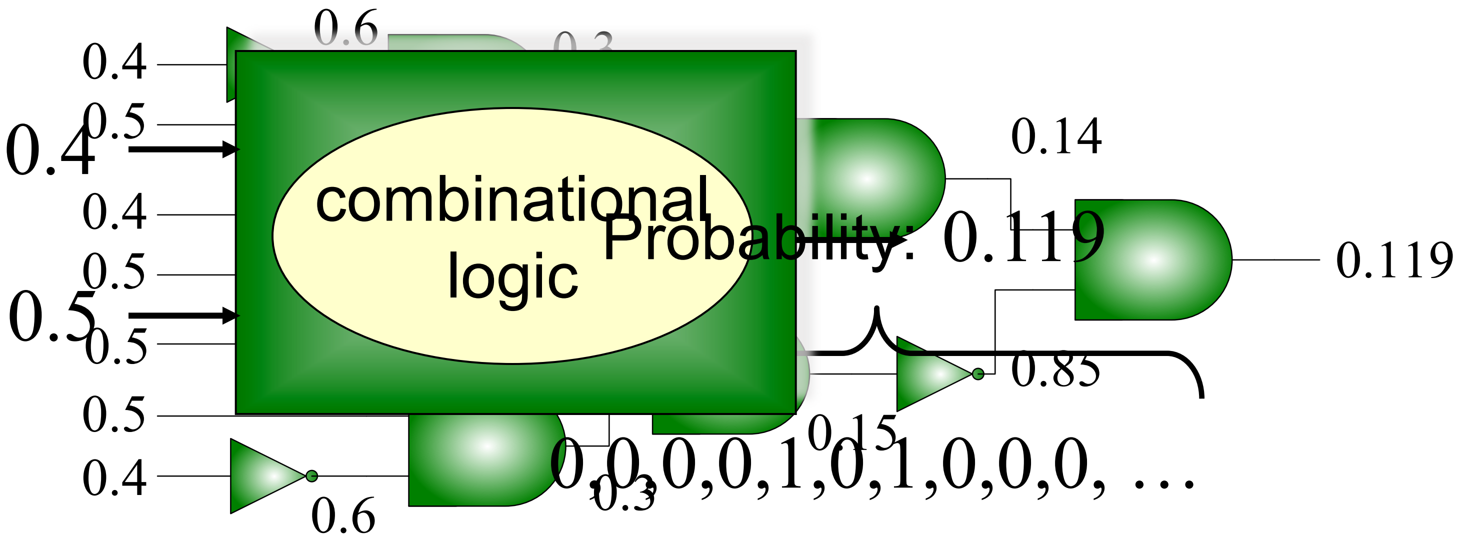
Probability values are the input and output signals.



Functions of a probability value t.

Synthesizing Logic that Generates Probabilities

Transform a source set of probabilities to a target set entirely through combinational logic



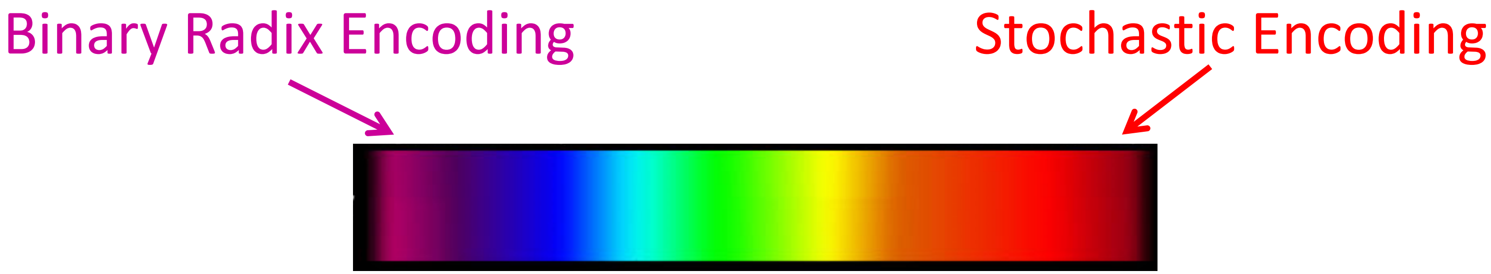
History

- Ideas first proposed by Gains and Poppelbaum in the late 1960's.
- Revisited by Brown and Card the Neural Networks Community in the 1990's.
- Work by my group (W. Qian's Ph.D.) in 2008 reignited interest:

  Proposed the first general synthesis methodology.
  270 Google Scholar Citations

Comparison of Encoding

	Binary Radix Encoding	Stochastic Encoding
Circuit Area	Large (Positional, Weighted)	Small (Uniform)
Fault Tolerance	Bad (Positional)	Good (Uniform, Long Stream)
Delay	Short (Compact, Efficient)	Long (Not compact, Long Stream)

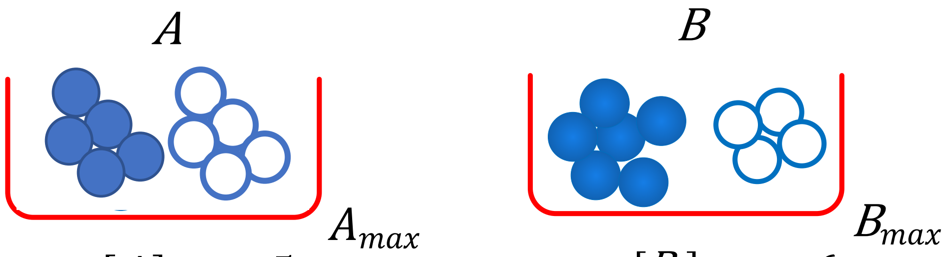


Spectrum of Encoding

Concept 2:
Encoding as Fractional Concentrations

Fractional Encodings as Concentrations

- A variable is associated with nicking or not nicking DNA at a given position.
- Its value is a fraction between 0 or 1 relative to a maximum concentration.
- To set the value, separate a solution of DNA strands into two different wells; nick the strands in one well at a 100% success rate; do not nick the strands in the other well; then mix the contents of the wells together at the desired proportion.
- "Multiplication" is achieved by concatenating these operations.



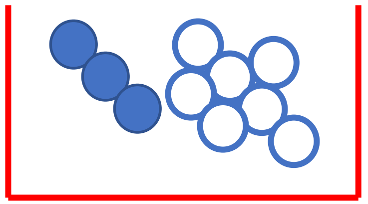
$$a = \frac{[A]}{[Amax]} = \frac{5}{10}$$

$$b = \frac{[B]}{[Bmax]} = \frac{6}{10}$$

Multiplying using multiple nicks

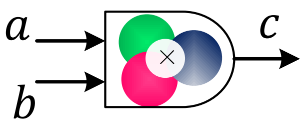
- Suppose there are two nickable locations A and B on a DNA strand. Separate; nick at A; mix with proportion a; separate; nick at B; mix with proportion b. Then use a probe to release the strand AB. The concentration of AB should be $a \times b$.
- This method can be extended to an arbitrary cascade of multiplication operations.

$$c = a \times b = \frac{5}{10} \times \frac{6}{10} = \frac{3}{10}$$



AB_{max}

$$c = a \times b \equiv$$

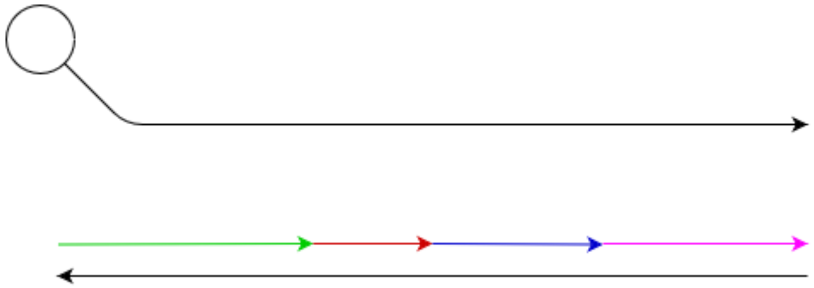


$$a = \frac{[AB\ ]}{[ABmax]} = \frac{3}{10}$$

Concept 3:
From Nicks to Strands

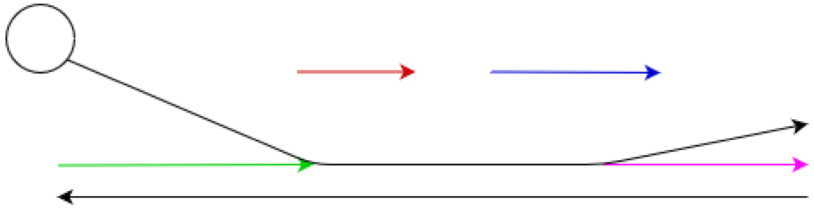
From Nicks to Strands

Suppose there are multiple nicks on a DNA strand. A probe that is complimentary to the strand causes the nicked strand to release substrands (shown in red and blue).



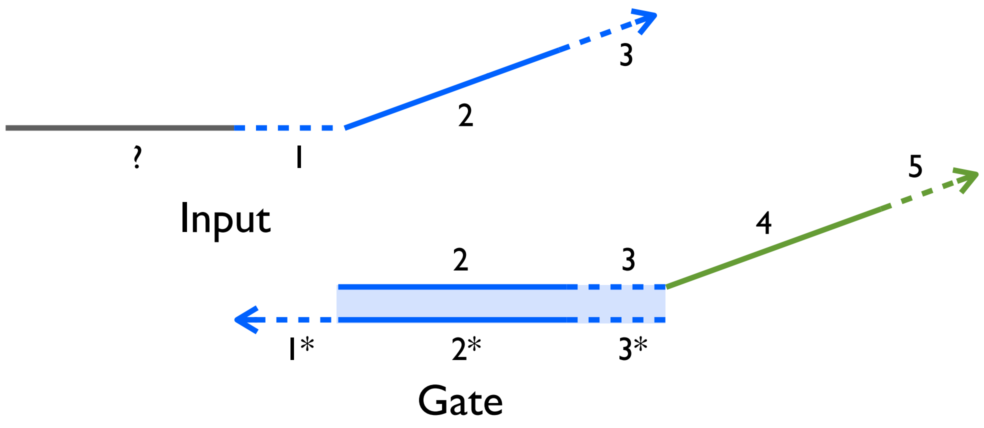
From Nicks to Strands

Suppose there are multiple nicks on a DNA strand. A probe that is complimentary to the strand causes the nicked strand to release substrands (shown in red and blue).

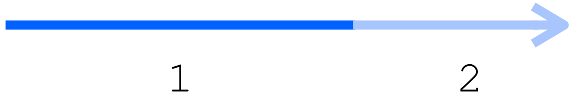


Concept 4:
Transforming DNA Strands

DNA strand displacement mechanism

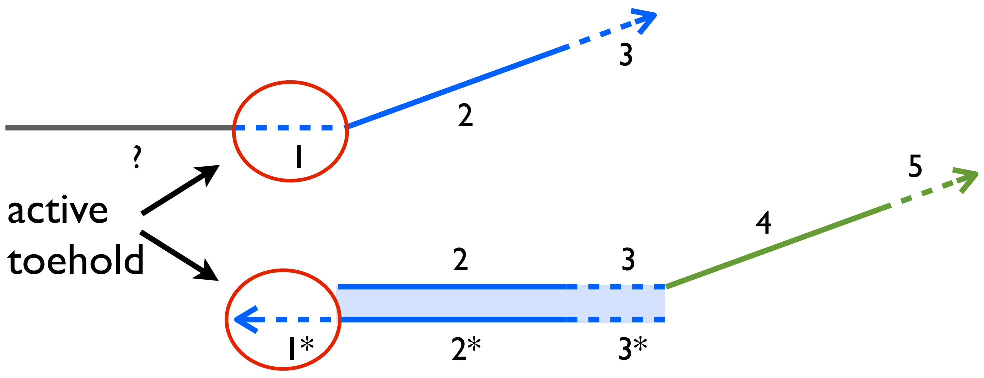


RNA sequence:
5'-AAUUCAGAUCCACCCAAAGAG-3'



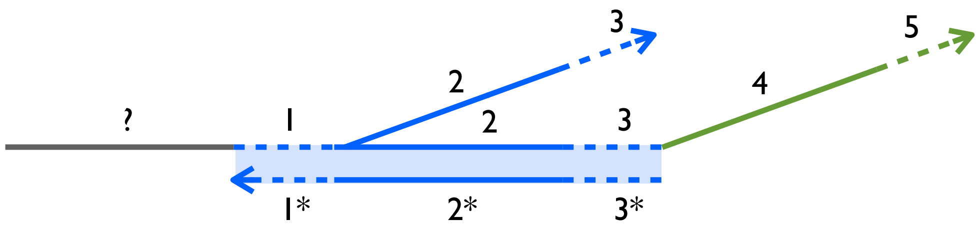
For a review see D.Y. Zhang and G. Seelig, Nature Chemistry (2011)

DNA strand displacement mechanism



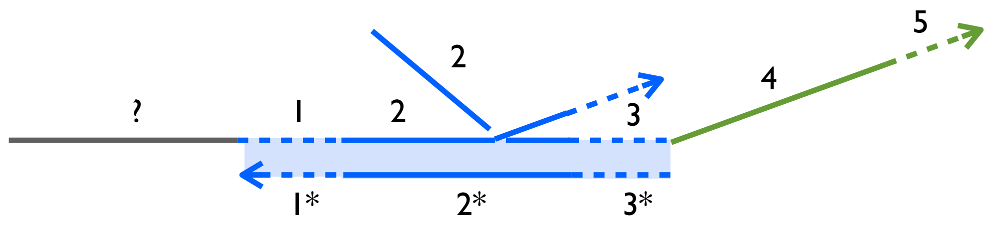
For a review see D.Y. Zhang and G. Seelig, Nature Chemistry (2011)

DNA strand displacement mechanism



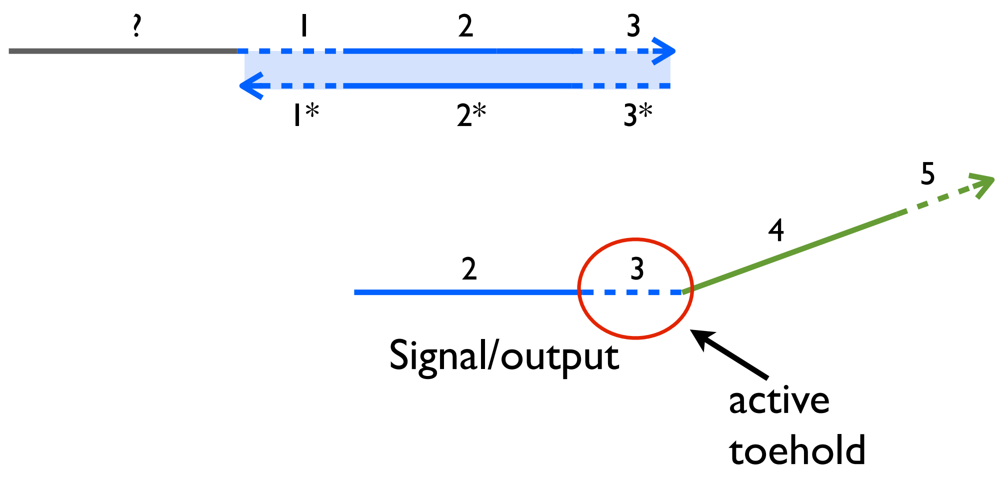
Strand displacement is initiated at the single-stranded toeholds. Toehold binding is a reversible process.

DNA strand displacement mechanism



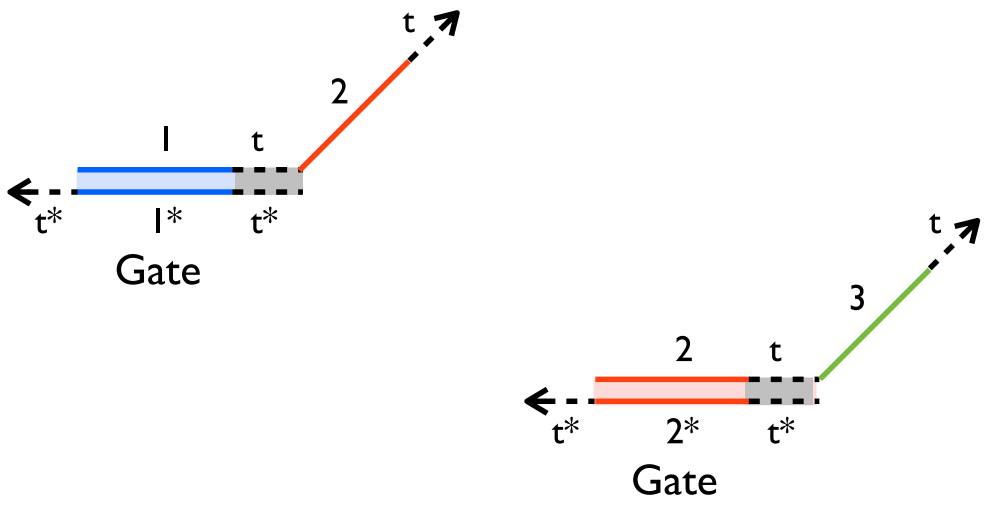
Strand displacement proceeds through a branch migration. Branch migration is a random walk.

DNA strand displacement mechanism



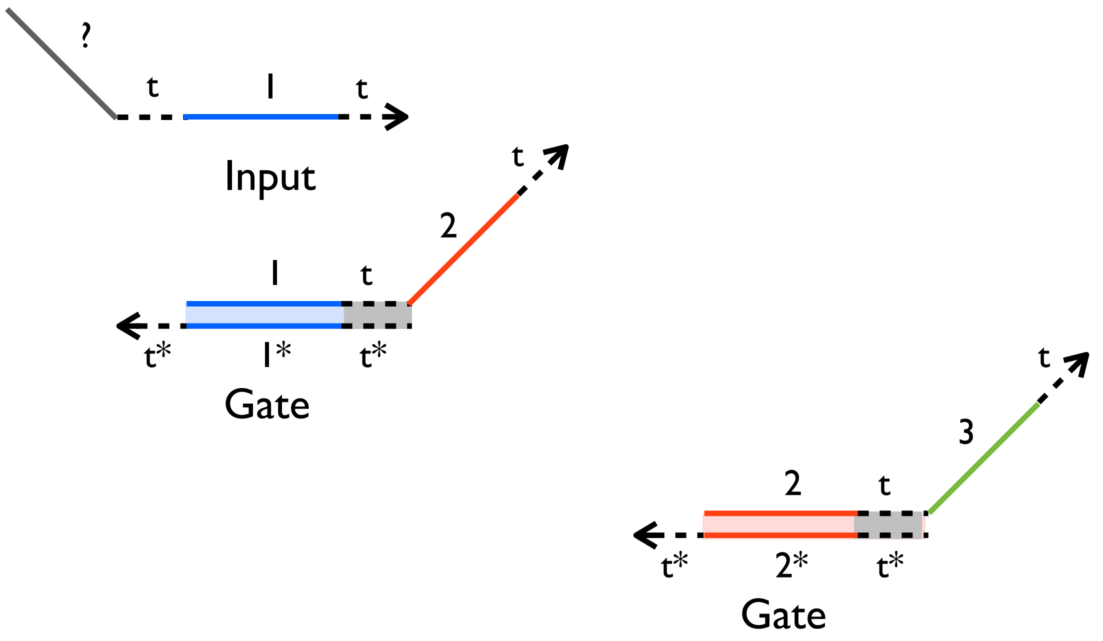
Release of the output strand is (almost) irreversible in the absence of a toehold for the reverse reaction.

Signals can propagate through multiple layers



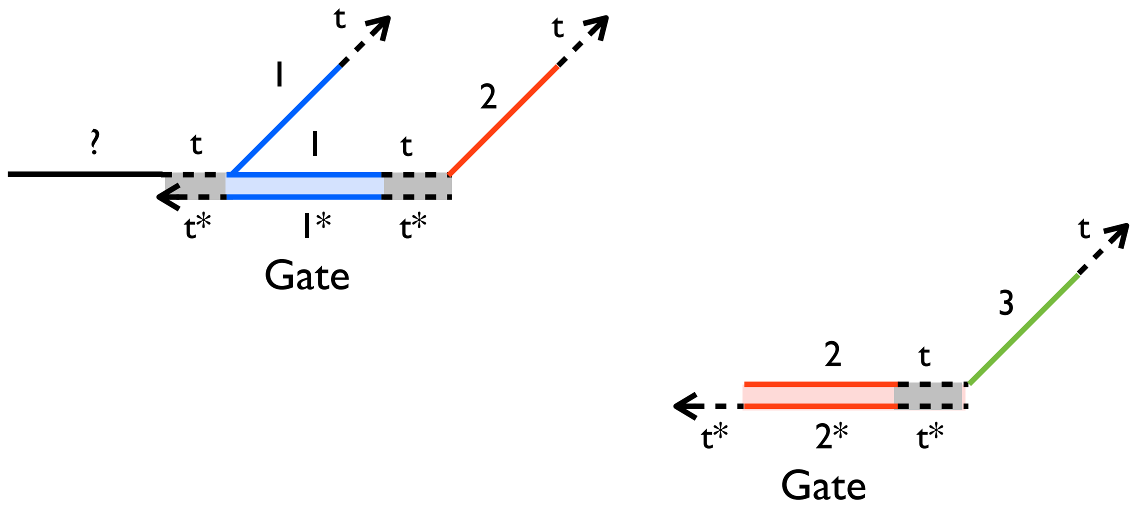
The sequences of inputs and outputs can be completely independent.

Signals can propagate through multiple layers



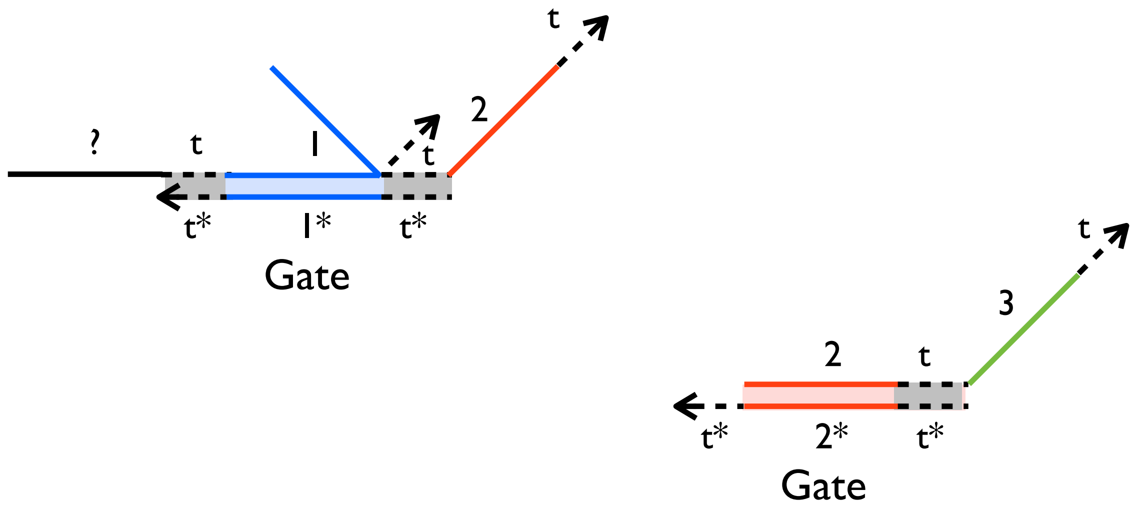
The sequences of inputs and outputs can be completely independent.

Signals can propagate through multiple layers



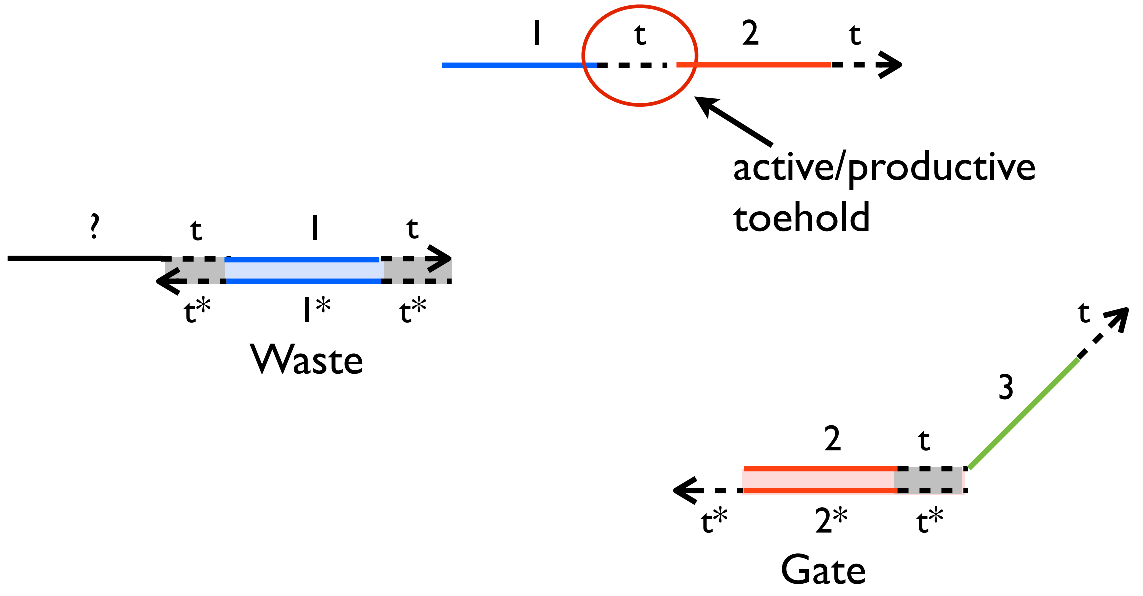
The sequences of inputs and outputs can be completely independent.

Signals can propagate through multiple layers



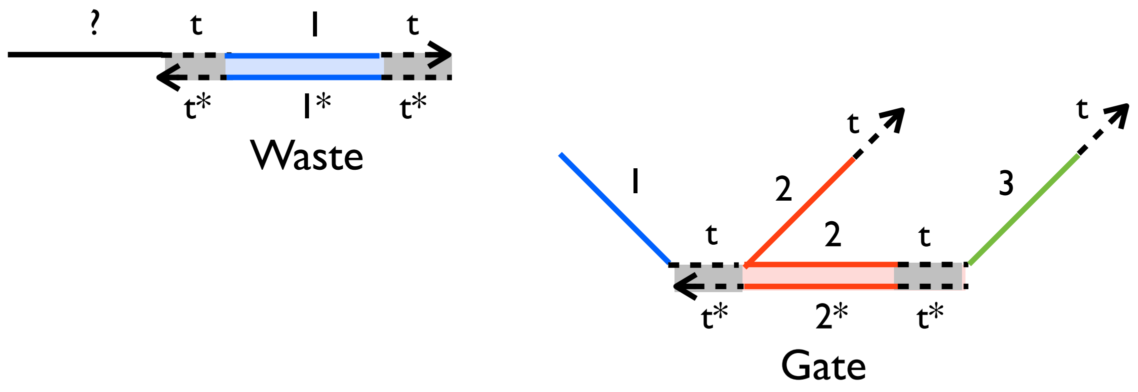
The sequences of inputs and outputs can be completely independent.

Signals can propagate through multiple layers



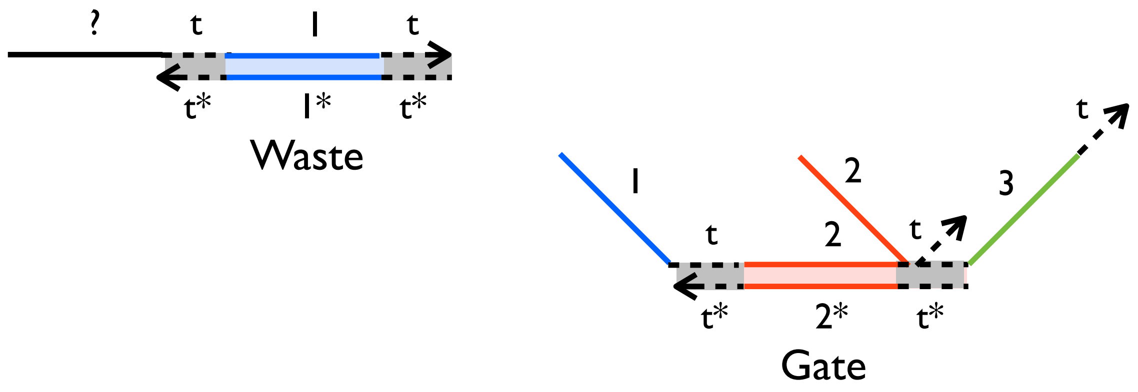
The sequences of inputs and outputs can be completely independent.

Signals can propagate through multiple layers



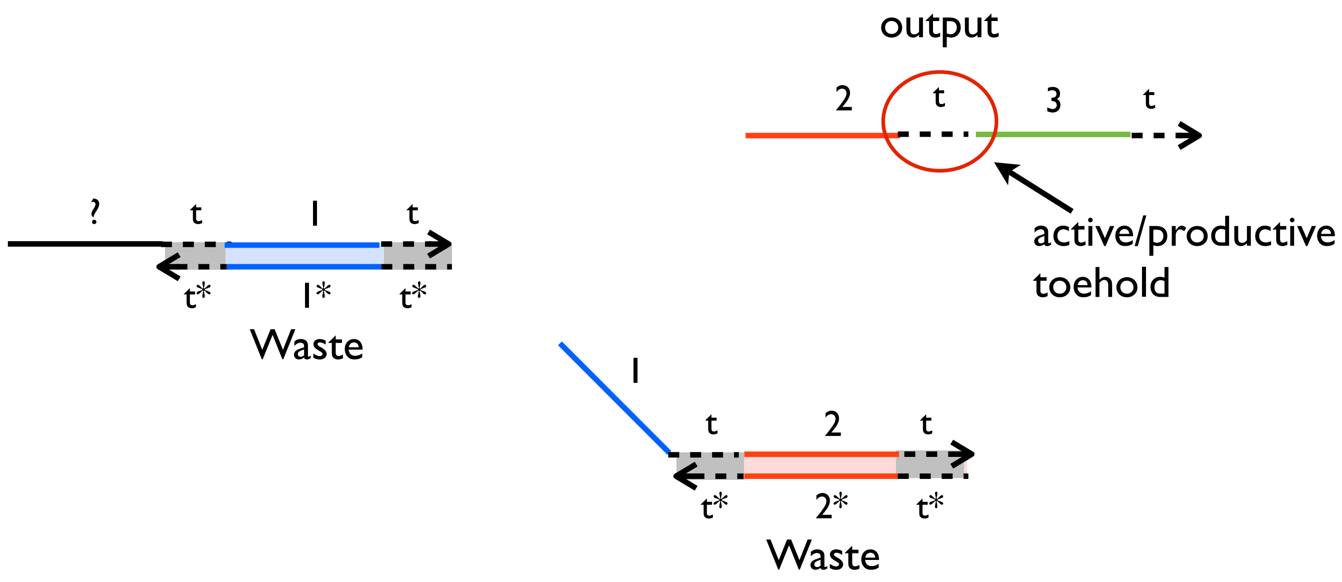
The sequences of inputs and outputs can be completely independent.

Signals can propagate through multiple layers



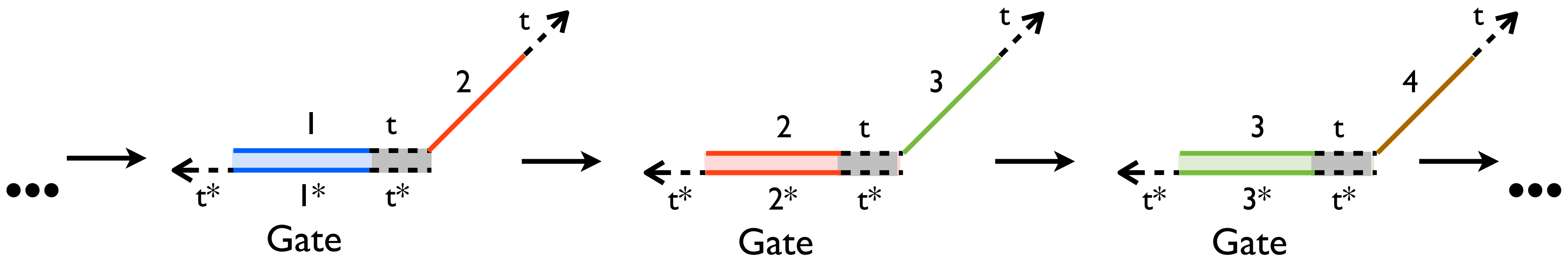
The sequences of inputs and outputs can be completely independent.

Signals can propagate through multiple layers



The sequences of inputs and outputs can be completely independent.

Signals can propagate through multiple layers

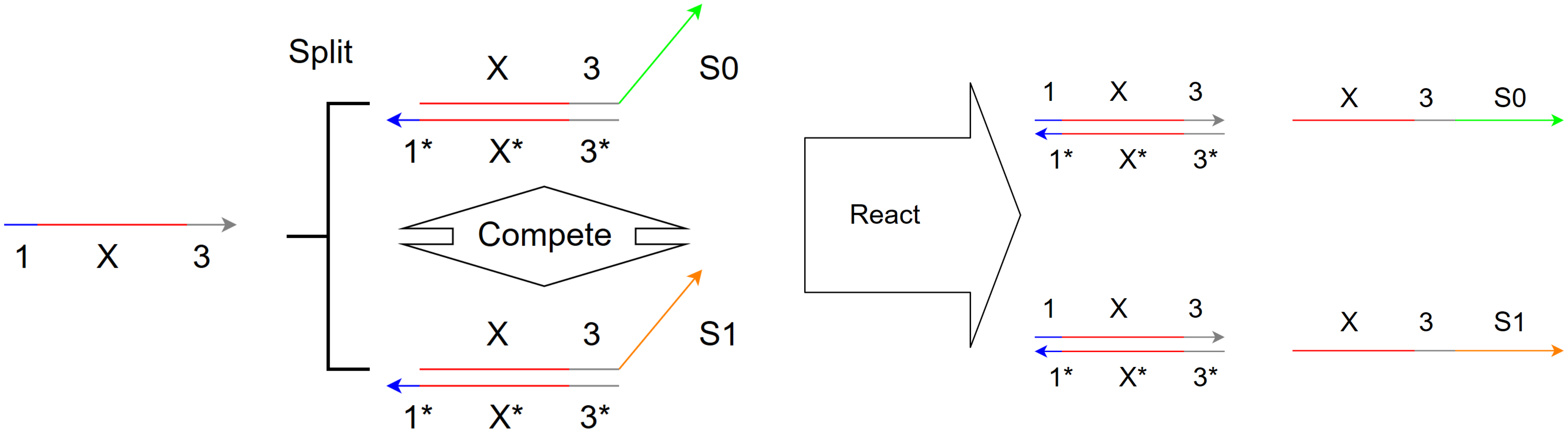


The sequences of inputs and outputs can be completely independent.

Concept 5:
Scaling DNA Strands

Scaling: Using Probabilistic Switch

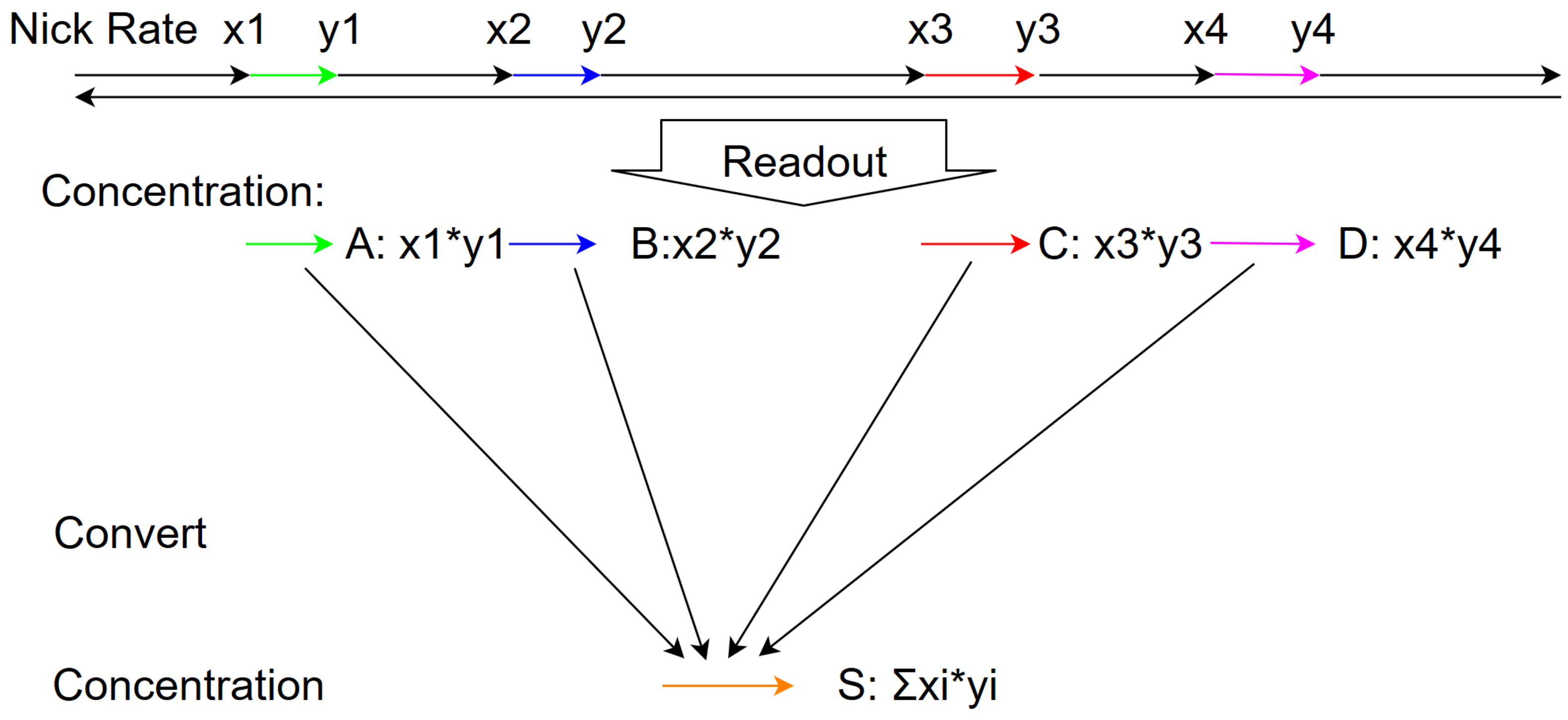
Competitive DNA Strand Displacement
(Wilhelm, Bruck, Qian, 2018)



Putting it all together:
Performing Computation

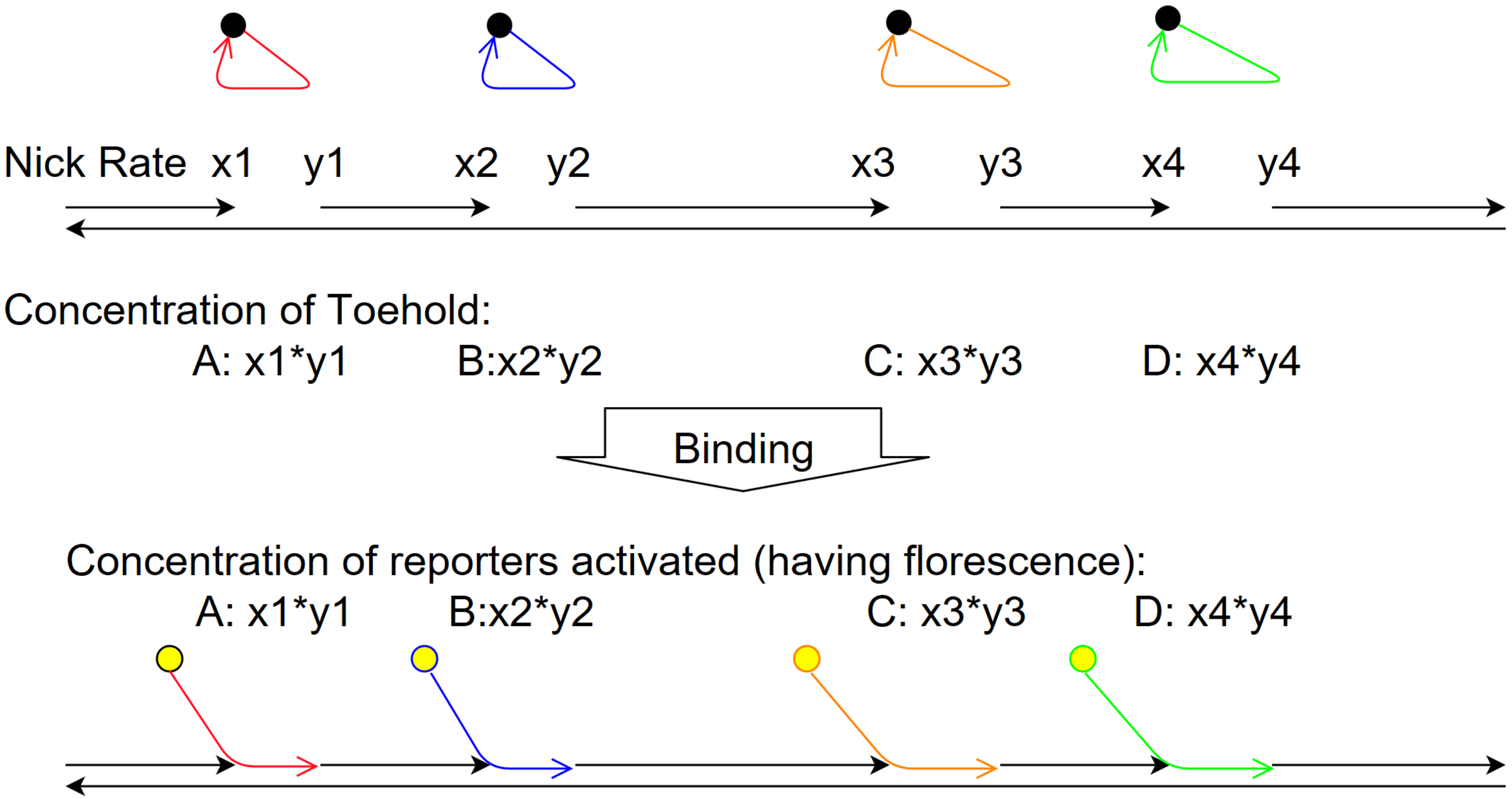
Performing Dot Product $c = \sum a_k * b_k$

- Build up each term $a_k * b_k$ through successive multiplication operations.
- Use a probe to release the required strands.
- Use buffer gates to convert each strand to a common output strand; its concentration is the result.



Dot Product: Using florescence

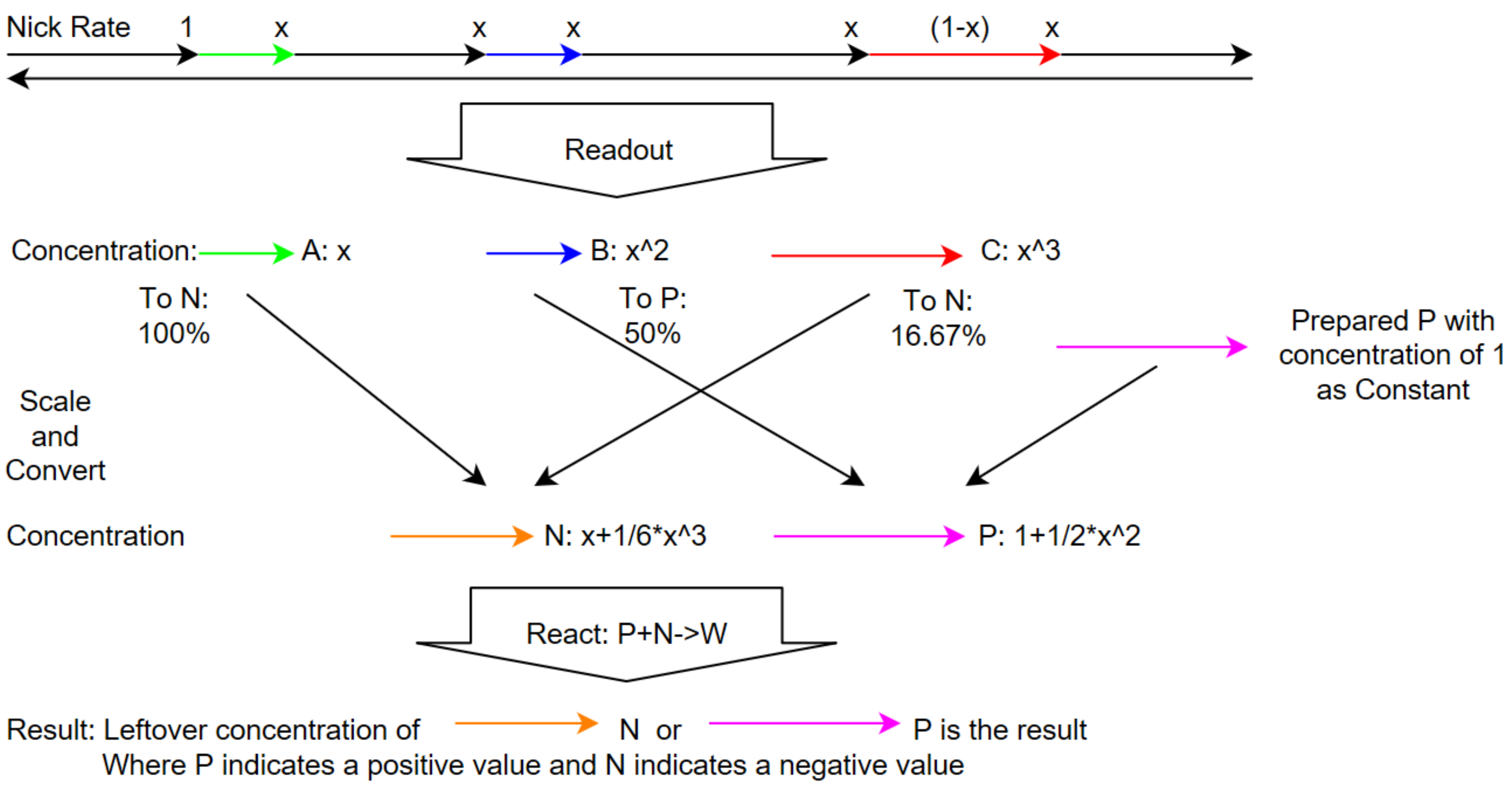
- Build up each term $a_k * b_k$ through multiplication; leave toeholds as the result.
- For each potential toehold, prepare a reporter.
- Report result through florescence, which measures the sum of the concentration of the reporters.



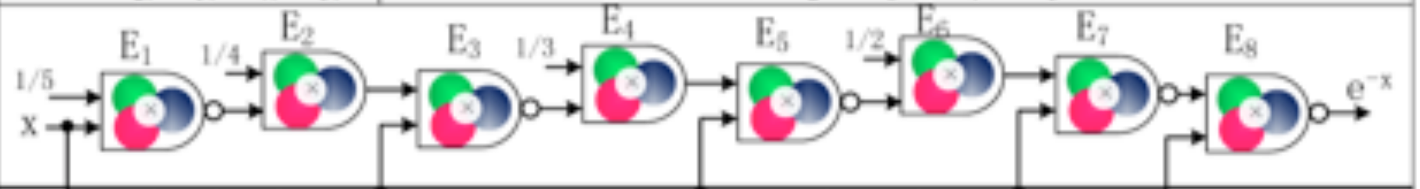
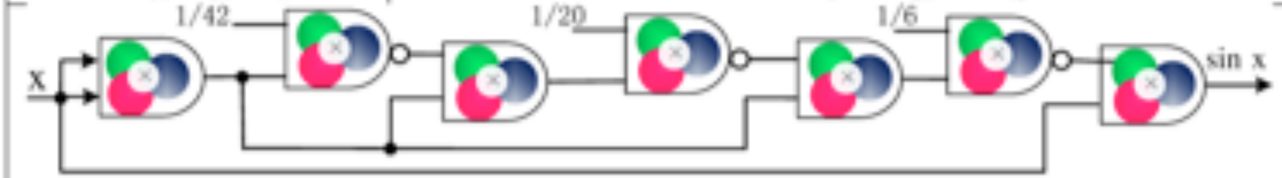
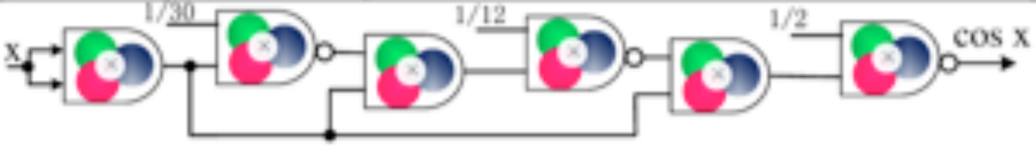
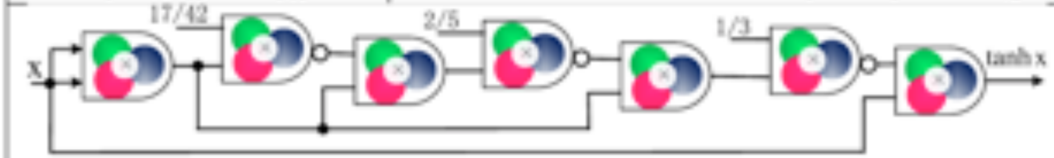
Building a polynomial function

- Build up each term through successive multiplication operations.
- Use a probe to release the required strands.
- Scale each term using competitive DNA strand displacement (into "positive" values P and "negative" values N.)
- Using strand displacement, execute the reaction $P + N \rightarrow Waste$. The concentration of the leftover is the evaluated value of the polynomial function.

Example: $f(x) = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!}$



More Examples

Function	Truncated Maclaurin series	Reformatted using Equation (7)
e^{-x}	$1-x+\frac{x^2}{2!}-\frac{x^3}{3!}+\frac{x^4}{4!}-\frac{x^5}{5!}$	$1-x(1-\frac{x}{2}(1-\frac{x}{3}(1-\frac{x}{4}(1-\frac{x}{5}))))$
		
$\sin(x)$	$x-\frac{x^3}{3!}+\frac{x^5}{5!}-\frac{x^7}{7!}$	$x(1-\frac{x^2}{6}(1-\frac{x^2}{20}(1-\frac{x^2}{42})))$
		
$\cos(x)$	$1-\frac{x^2}{2!}+\frac{x^4}{4!}-\frac{x^6}{6!}$	$1-\frac{x^2}{2}(1-\frac{x^2}{12}(1-\frac{x^2}{30}))$
		
$\log(1+x)$	$x-\frac{x^2}{2}+\frac{x^3}{3}-\frac{x^4}{4}$	$x(1-\frac{x}{2}(1-\frac{2}{3}x(1-\frac{3}{4}x)))$
		
$\tanh(x)$	$x-\frac{1}{3}x^3+\frac{2}{15}x^5-\frac{17}{315}x^7$	$x(1-\frac{x^2}{3}(1-\frac{2}{5}x^2(1-\frac{17}{42}x^2)))$
		
sigmoid(x)	$\frac{1}{2}+\frac{x}{4}-\frac{x^3}{48}+\frac{x^5}{480}$	$1-\frac{1}{2}(1-\frac{x}{2}(1-\frac{x^2}{12}(1-\frac{x^2}{10})))$
		

More Examples

Function	Truncated Maclaurin Series	Total nicks needed	Parallel Read outs	Gates used in Stochastic computing
e^{-x}	$1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \frac{x^5}{5!}$	16	5	8
$\sin(x)$	$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}$	17	4	7
$\log(1 + x)$	$x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4}$	11	4	6
$sigmoid(x)$	$\frac{1}{2} + \frac{x}{4} - \frac{x^3}{48} + \frac{x^5}{480}$	10	3	7

Challenges

- Reading without destroying data: how to translate data encoded with nicks into displacement strands without destroying the original nicked structure.
- Performing the requisite DNA strand-displacement operations: “leakage” and experimental artifacts present challenges for computations with more than 3 levels.
- Performing the readout.
- Alternatively, re-encoding the results of “in-memory” computation.

Long-Term Goals

- Demonstrate solutions to machine learning problems:
  core operations are matrix multiply and thresholding, i.e.,
  $c_{ij} = \sum_k a_{ik} * b_{kj}$ followed by $sigmoid(cij)$
- Develop "in-memory" computation for "big data": leverage the high density of storage with DNA for applications with large volumes of data, but limited I/O requirements.