DNA Circuits for Analog Computing

Tianqi Song and John Reif
Duke University

Outline

- Motivation
  - What is analog computing?
  - Why are we interested in DNA-based analog computing?
- DNA-based Analog Gates
- DNA-based Analog Circuits
- Future Work

What is DNA computing?

- DNA is a highly programmable biological material based on Watson-Crick base pairs (A-T, G-C).
- We can control the reaction pathway and kinetics of DNA-based system by programming the sequences.
- DNA computing is using DNA as hardware to perform computing.
- The computing is usually based on DNA hybridization and DNA strand displacement.

Adopted from ***Reif, John H. "Scaling up DNA computation." science 332.6034 (2011): 1156-1157.***

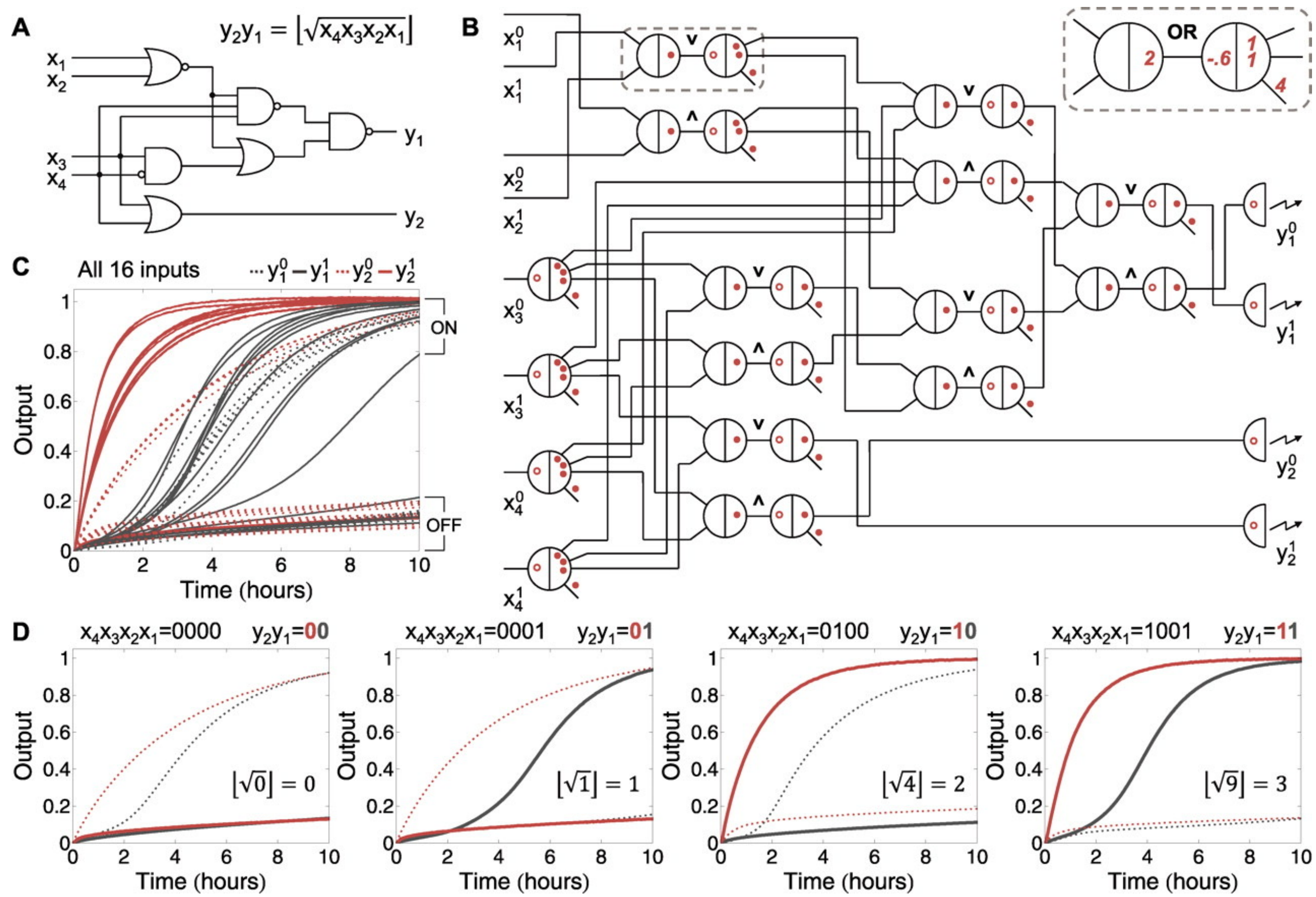
Why are we interested in DNA computing?

- DNA is a highly programmable biological material based on Watson-Crick base pairs (A-T, G-C).
- We can control the reaction pathway and kinetics of DNA-based system by programming the sequences.
- Large scale DNA-based digital circuits have been demonstrated in

Qian, Lulu, and Erik Winfree. "Scaling up digital circuit computation with DNA strand displacement cascades." Science 332.6034 (2011): 1196-1201.

Large scale DNA-based digital circuits have been demonstrated using

- Seesaw Gates and
- Dual-rail logic



Adopted from *Qian, Lulu, and Erik Winfree. "Scaling up digital circuit computation with DNA strand displacement cascades." Science 332.6034 (2011): 1196-1201.*

What is Analog Computing?

- **Input and output are directly encoded by physical quantity (e.g. voltage for electrical analog machines, position for mechanical analog machines or concentration of DNA species for DNA-based analog circuits) without using threshold.**
- **Input Range: the range within which the inputs of an analog device should lie to make sure it works properly. For our DNA-based analog circuits, the input range should be a range of concentration of input DNA species.**
- **Signal degrading and fluctuation: change of physical quantity will directly result in change of signal.**

Related Definitions for DNA-based Analog Computing

- **Analog Values:** **are encoded by the ratio between the concentration of a DNA strand and the unit of relative concentration.**
  - **For example, the concentration of a DNA strand is 20 nM and the unit in relative concentration is 5 nM, then its relative concentration is 20/5 = 4.**
- **Input Range of an analog DNA gate:** **the range of relative concentration within which relative concentrations of its input DNA species of an analog DNA gate should lie.**
- **Valid Output Range:** **the range of relative concentration within which output of an analog DNA gate is considered to be correct. The output is encoded by relative concentration of output DNA strand.**

Why are we interested in DNA-based analog computing?

- Minimal work has been done on systematic construction of DNA circuits for analog computing.

- Biological analog computing including DNA-based analog computing has several potential applications.

Potential Applications of DNA Circuits for Analog Computing

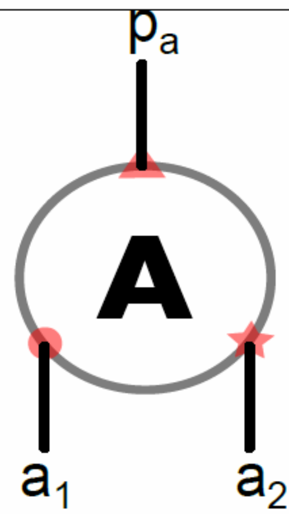
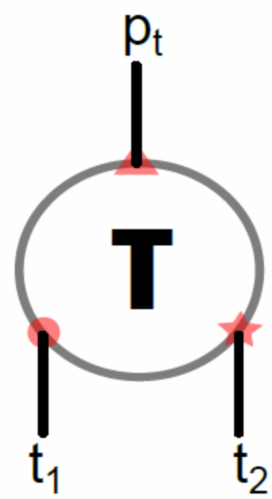
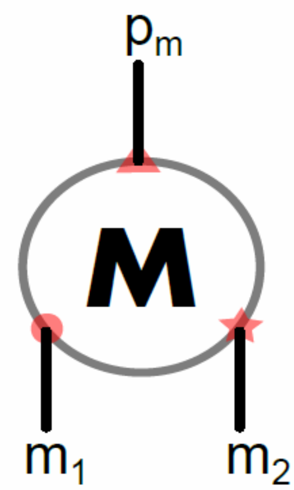
- **Analog circuits need less gates to perform arithmetic operation. For example, we only need one gate for each arithmetic operation (addition, subtraction and multiplication) in analog system.**
- **In digital system, we need several gates.**
- **This property makes analog DNA circuits be useful in resource-limited environment e.g. in living cells**

Daniel, Ramiz, et al. "Synthetic analog computation in living cells." Nature 497.7451 (2013): 619-623.

- **Nature operates in a hybrid of analog and digital fashion.**
- **Analog DNA circuits can serve as interface to natural analog systems.**

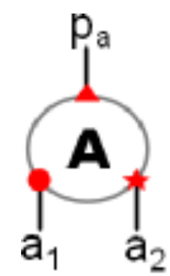
Sarpeshkar, Rahul. "Analog versus digital: extrapolating from electronics to neurobiology." Neural computation 10.7 (1998): 1601-1638.

DNA-based Analog Gates

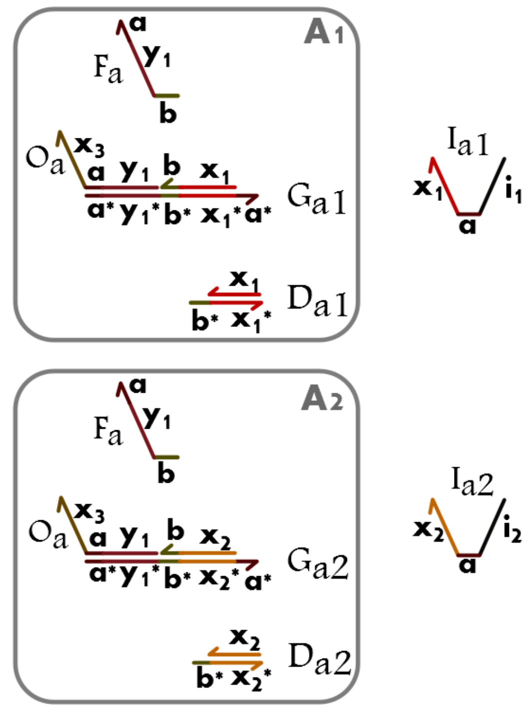
Gates	Addition Gate	Truncation gate	Multiplication Gate
Abstraction			
Function	$p_a = a_1 + a_2$	$p_t = t_1 - t_2 \ (t_1 \geq t_2)$	$p_m = m_1 * m_2$

- **Input of a gate:** is encoded by the initial relative concentrations of its corresponding input DNA species.
- **Output of a gate:** is encoded by relative concentration of the output DNA species.
- **Gate Composition:** A gate is comprised of multiple components (single DNA strand or DNA complex) which are categorized into several groups:
  - **A DNA complex *Gsub*** that takes in input strand or internal signal strand for communication between two groups, and generates output strand or another internal signal strand;
  - **A fuel DNA strand *Fsub*** to help generate the output strand or internal signal strand;
  - **A drain *Dsub*** to consume single strand which will be harmful if stays freely in solution where *sub* is subscript to denote the gate type and group number.

Design of Addition Gate:



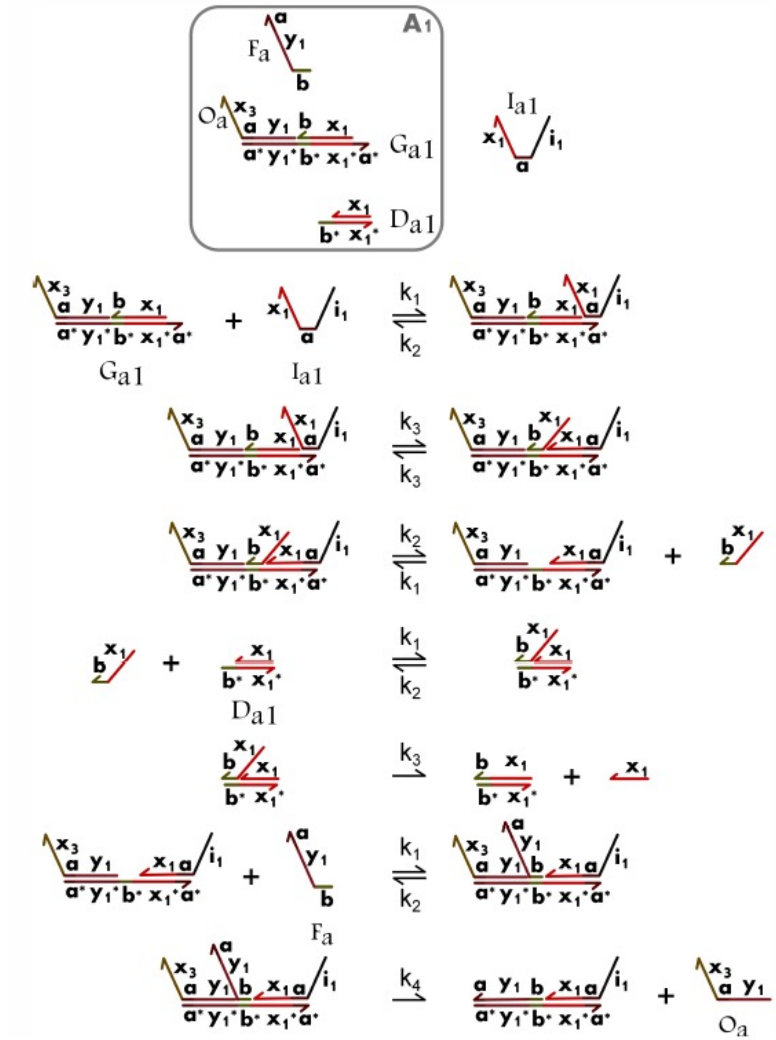
- **Input a_1 is encoded by the initial relative concentration of input strand I_{a1}.**
- **Input a_2 is encoded by the initial relative concentration of input strand I_{a2}.**
- **Output p_a is encoded by the relative concentration of output strand O_a.**
- **To set up input range $(0, r_a)$: the configuration of initial relative concentrations is set as**
  - **$[G_{a1}]_{ini} = [D_{a1}]_{ini} = [G_{a2}]_{ini} = [D_{a2}]_{ini} = r_a$ and**
  - **$[F_a]_{ini} = 2*r_a$,**
  - **where r_a is a positive real number.**



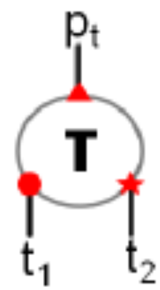
Mechanism of Addition Gate

$$I_{a1} + \mathbf{A}_1 \rightharpoonup O_a$$

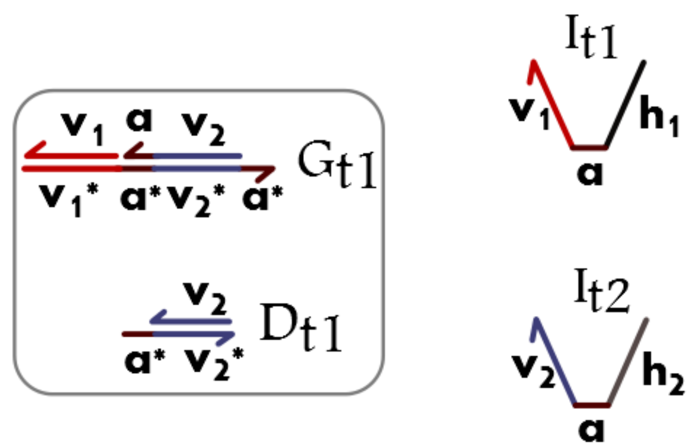
$$I_{a2} + \mathbf{A}_2 \rightharpoonup O_a$$



Design of Truncation Gate

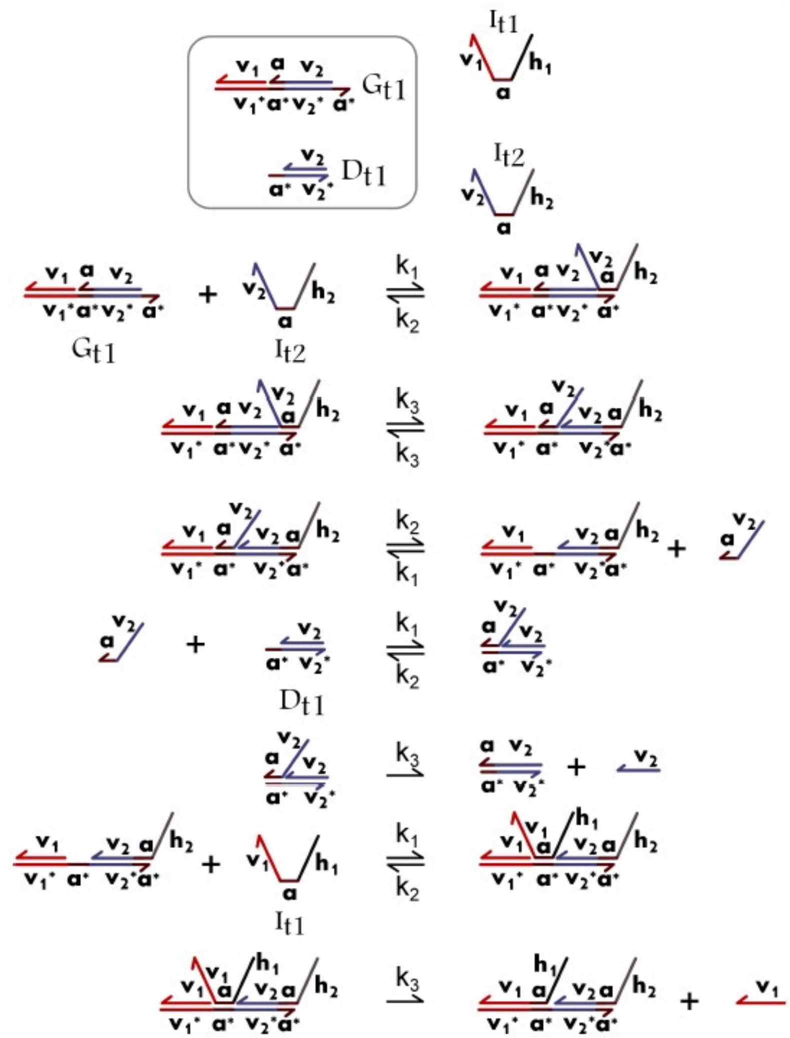


- **Input t_1** is encoded by the initial relative concentration of input strand I_{t1}. Input t_2 is encoded by the initial relative concentration of input strand I_{t2}.
- **Output p_t** is encoded by the relative concentration of remaining I_{t1} after the computing process.
- **To set up input range $(0, r_t)$:** the configuration of initial relative concentrations is set as
  - $[G_{t1}]_{ini} = [D_{t1}]_{ini} = 2*r_t$,
  - where r_t is a positive real number.

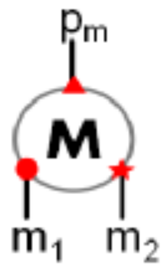


Mechanism of Truncation Gate

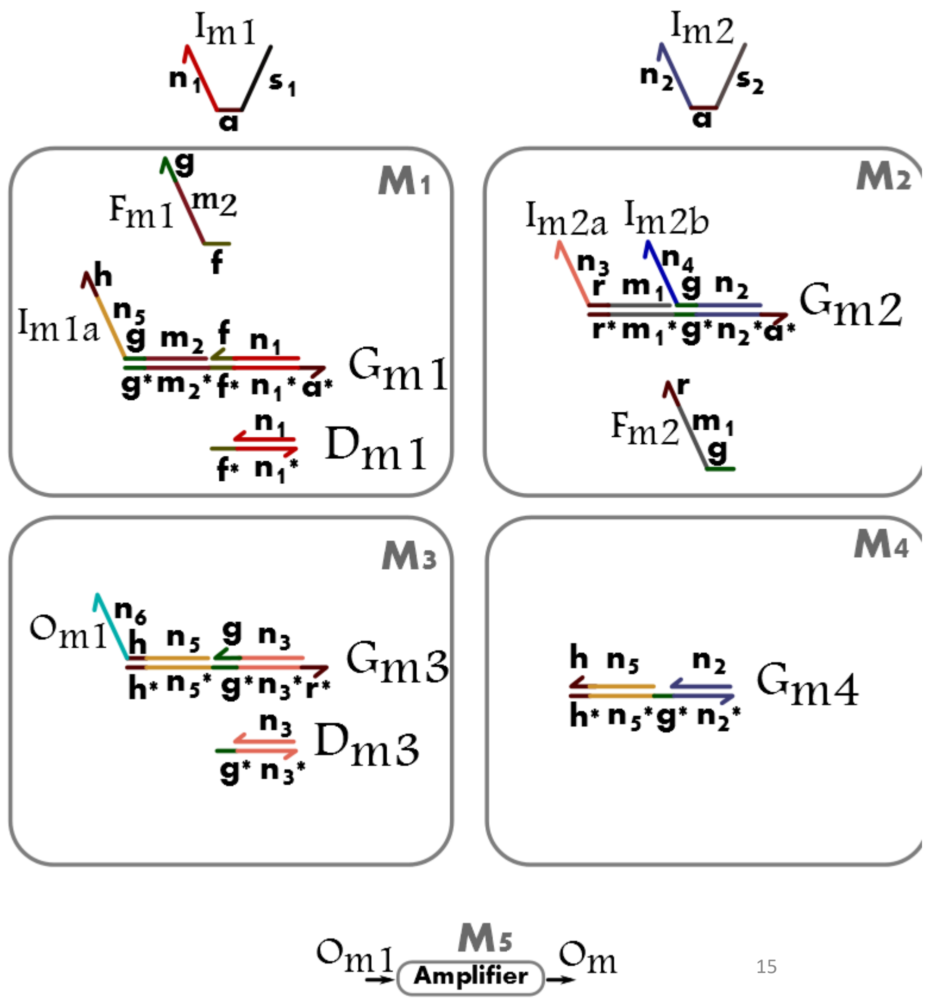
$$I_{t1} + I_{t2} + G_{t1} + D_{t1} \rightharpoonup \text{waste}$$



Design of Multiplication Gate

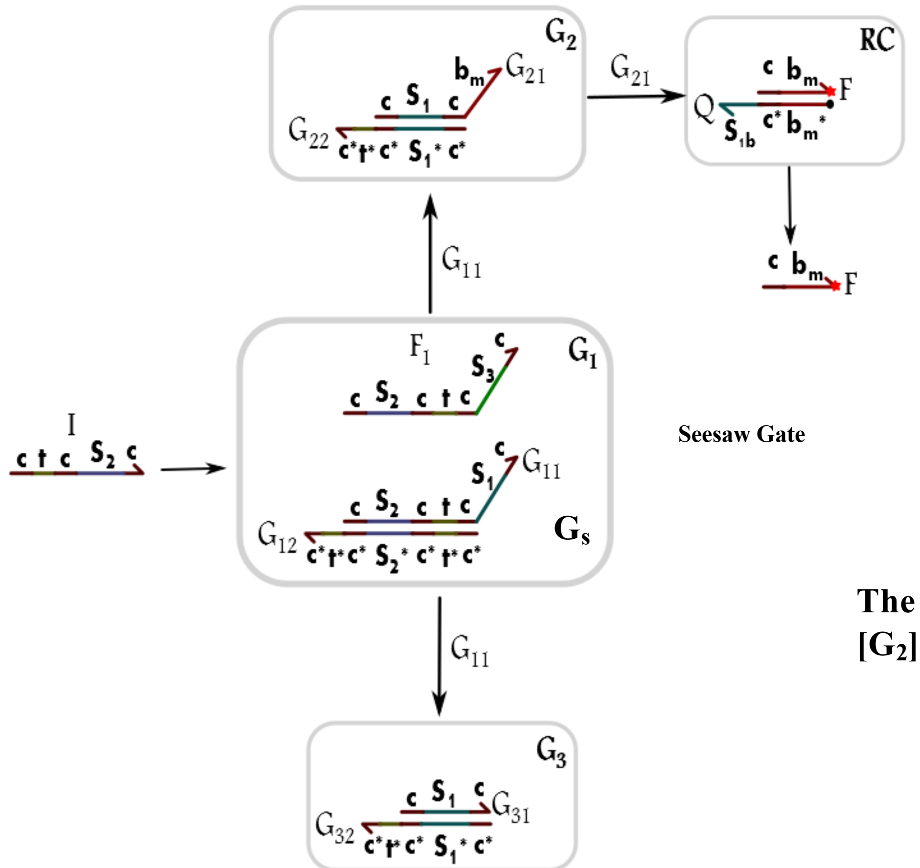


- **Input m_1 is encoded by the initial relative concentration of input strand I_{m1}.**
- **Input m_2 is encoded by the initial relative concentration of input strand I_{m2}.**
- **Output p_m is encoded by the relative concentration of output strand O_m.**
- **To set up input range $(0, r_m)$: the configuration of initial relative concentrations is set as**
- **$[G_{m1}]_{ini} = [D_{m1}]_{ini} = [F_{m1}]_{ini} = [G_{m2}]_{ini} = [F_{m2}]_{ini} = [G_{m3}]_{ini} = [F_{m3}]_{ini} = r_m$,**
- **where r_m is a positive real number.**



Competitive DNA Strand Displacement:

Using Seesaw gate of *Qian, Lulu, and Erik Winfree. "Scaling up digital circuit computation with DNA strand displacement cascades." Science 332.6034 (2011): 1196-1201.*



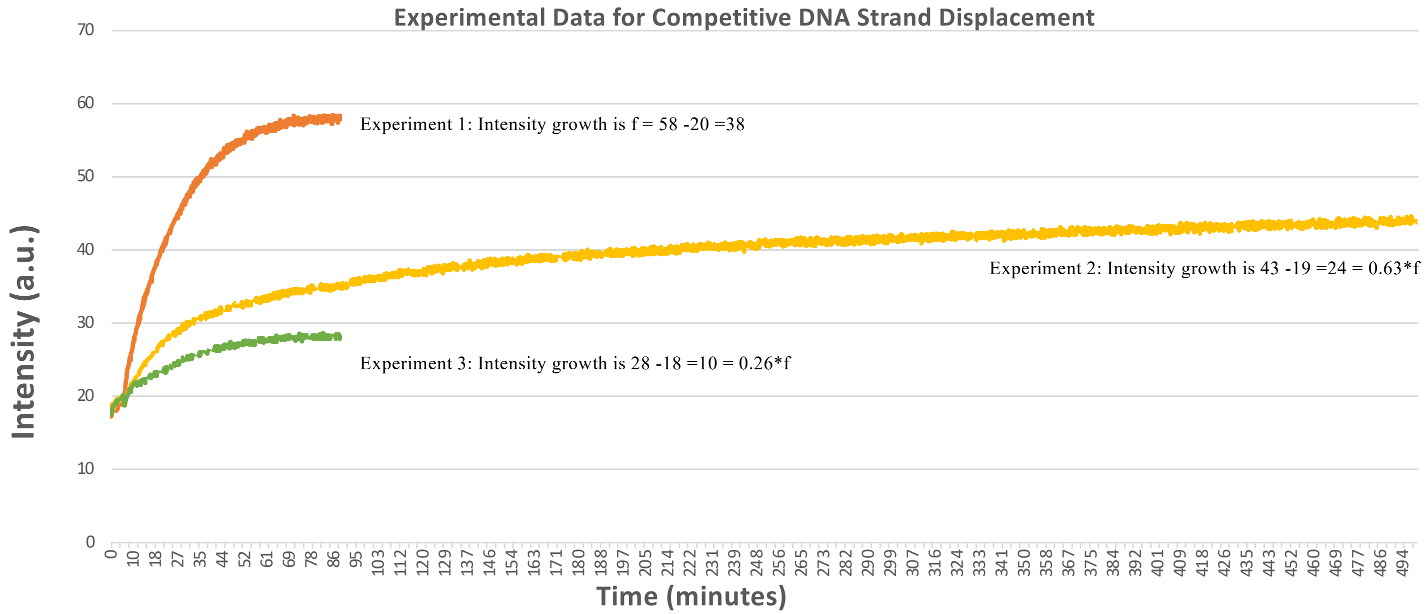
Seesaw Gate

$[G_s]_{ini} < [G_2]_{ini}, [G_3]_{ini}$

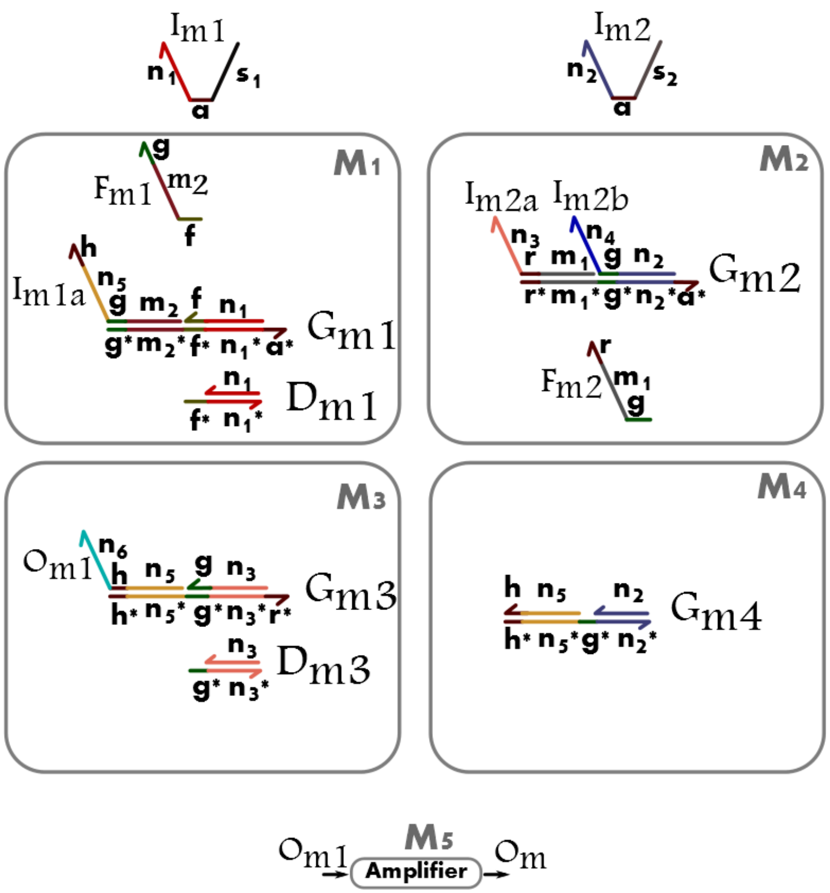
The portion of G_{11} consumed by G_2 is $[G_2]_{ini} / ([G_2]_{ini} + [G_3]_{ini})$

Experiments on Competitive DNA Strand Displacement

Experiment Number	I	G_{1a}	F_1	G_2	G_3	RC	Expected Fluorscence Intensity Growth
1	20 nM	100 nM	200 nM	100 nM	0	150 nM	f
2	20 nM	100 nM	200 nM	100 nM	100 nM	150 nM	$0.5f$ 0.5 = 100/(100+100)
3	20 nM	100 nM	200 nM	100 nM	200 nM	150 nM	$0.33f$ 0.33 = 100/(100+200)



Mechanism of Multiplication Gate



$$\textbf{I:}\quad I_{m2} + \mathbf{M}_2 \rightharpoonup I_{m2a} + I_{m2b} \qquad (1)$$

$$I_{m2a} + \mathbf{M}_3 \rightharpoonup G'_{m3} \qquad (2)$$

$$I_{m2b} + G_{m4} \rightharpoonup waste \qquad (3)$$

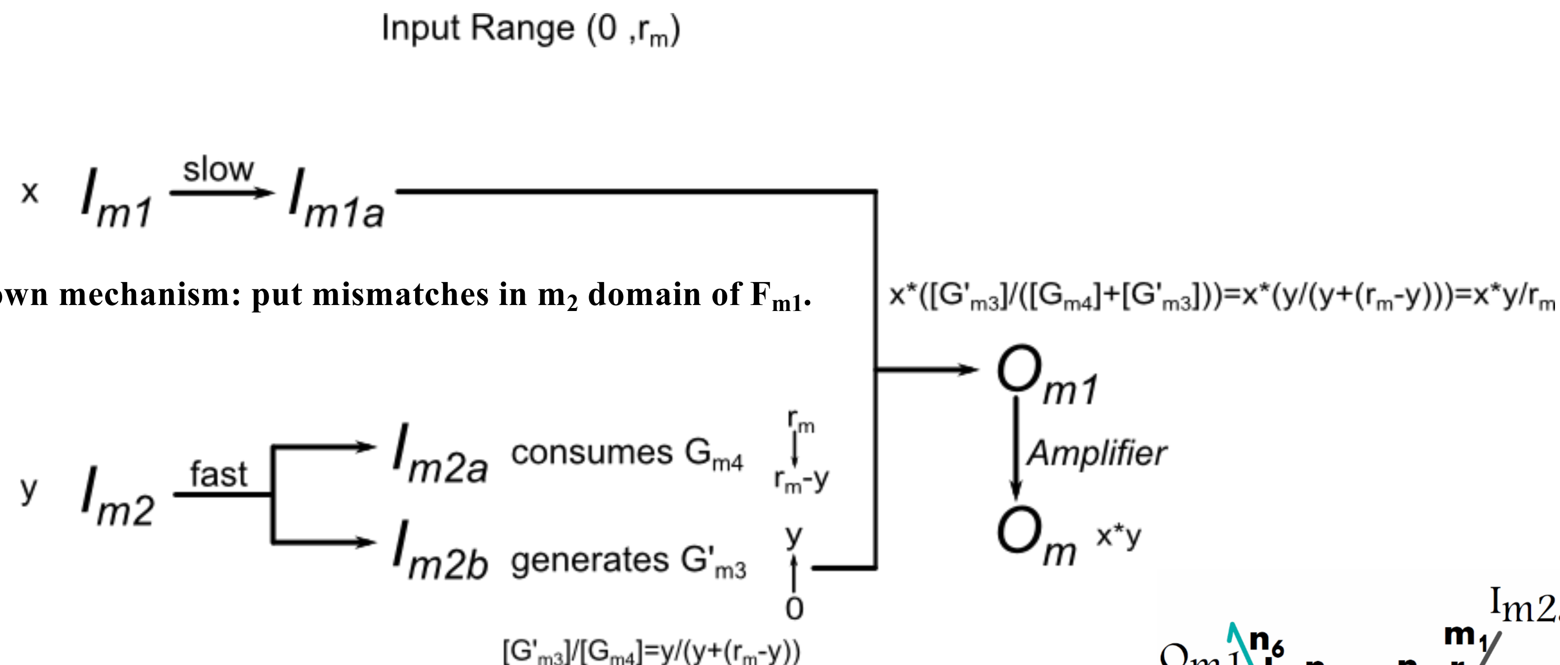
$$\textbf{II:}\quad I_{m1} + \mathbf{M}_1 \overset{\text{slow}}{\rightharpoonup} I_{m1a} \qquad (4)$$

$$\textbf{III:}\quad I_{m1a} + G'_{m3} \rightharpoonup O_{m1} \qquad (5)$$

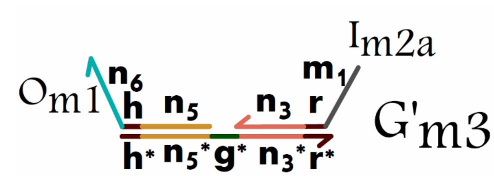
$$I_{m1a} + G_{m4} \rightharpoonup waste \qquad (6)$$

$$\textbf{IV:}\quad O_{m1} + Amplifier \rightharpoonup \mathbf{r}_m O_m \qquad (7)$$

Mechanism of Multiplication Gate



Slow down mechanism: put mismatches in m_2 domain of F_{m1}.

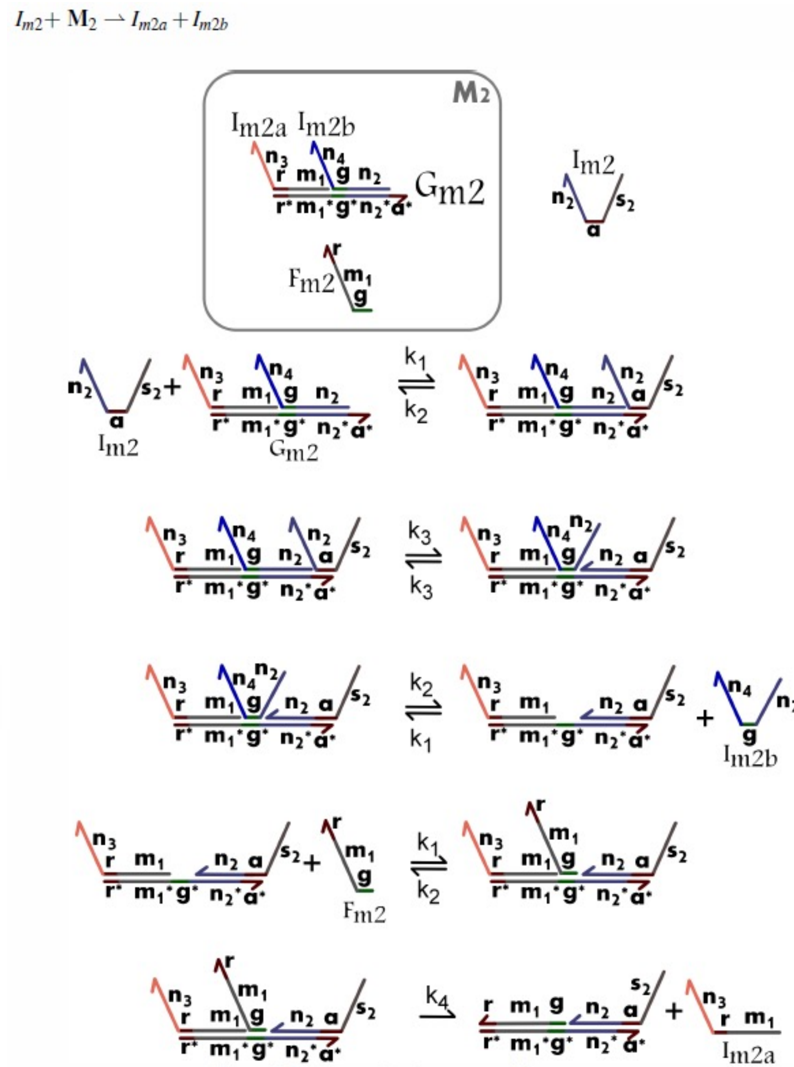


Mechanism of Multiplication Gate: Group I Reactions

$$\textbf{I:}\quad I_{m2}+\mathbf{M}_2 \rightharpoonup I_{m2a}+I_{m2b} \qquad (1)$$

$$I_{m2a}+\mathbf{M}_3 \rightharpoonup G'_{m3} \qquad (2)$$

$$I_{m2b}+G_{m4} \rightharpoonup waste \qquad (3)$$



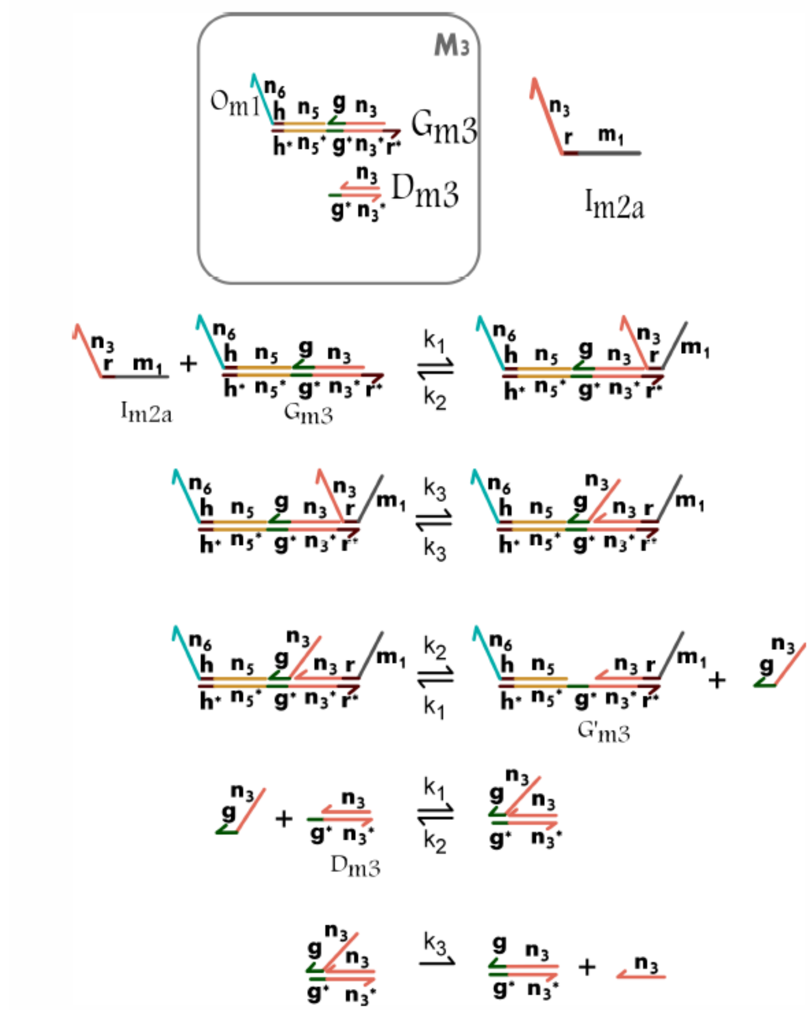
Mechanism of Multiplication Gate: Group I Reactions

$$\textbf{I:}\quad I_{m2} + \mathbf{M}_2 \rightharpoonup I_{m2a} + I_{m2b} \qquad (1)$$

$$I_{m2a} + \mathbf{M}_3 \rightharpoonup G'_{m3} \qquad (2)$$

$$I_{m2b} + G_{m4} \rightharpoonup waste \qquad (3)$$

$I_{m2a} + \mathbf{M}_3 \rightharpoonup G'_{m3}$



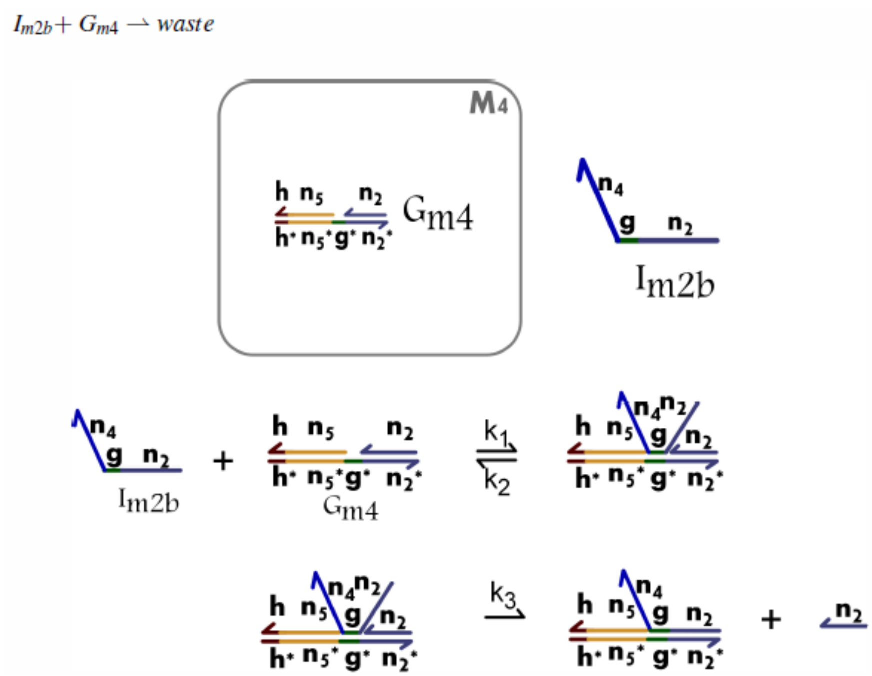
Mechanism of Multiplication Gate: Group I Reactions

I:

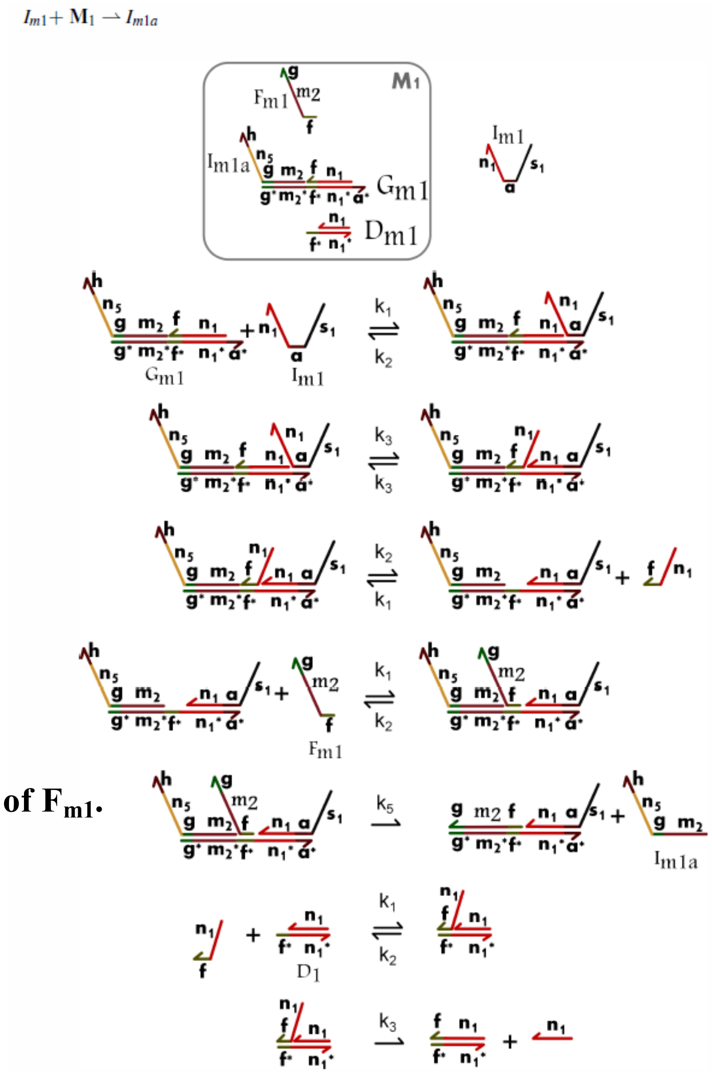
$$I_{m2} + \mathbf{M}_2 \rightharpoonup I_{m2a} + I_{m2b} \qquad (1)$$

$$I_{m2a} + \mathbf{M}_3 \rightharpoonup G'_{m3} \qquad (2)$$

$$I_{m2b} + G_{m4} \rightharpoonup waste \qquad (3)$$



Mechanism of Multiplication Gate: Group II Reactions



$$\mathbf{II}: \quad I_{m1} + \mathbf{M}_1 \overset{\text{slow}}{\rightharpoonup} I_{m1a} \qquad (4)$$

Slow down mechanism: put mismatches in m_2 domain of F_{m1}.

Mechanism of Multiplication Gate: Group III Reactions

III: $I_{m1a} + G'_{m3} \rightharpoonup O_{m1}$ (5)

$I_{m1a} + G_{m4} \rightharpoonup waste$ (6)

$I_{m1a} + G'_{m3} \rightharpoonup O_{m1}$

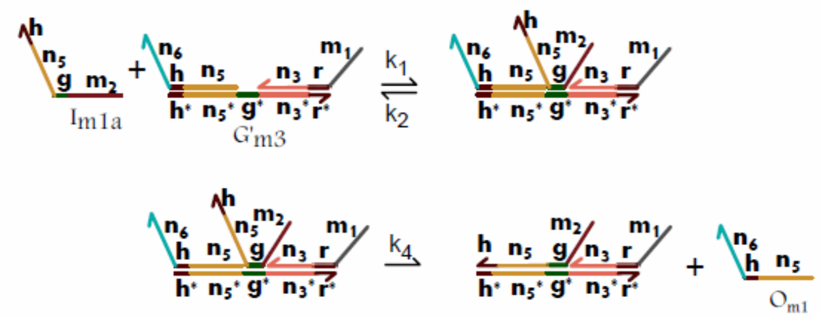


Figure 8: Reactions to implement $I_{m1a} + G'_{m3} \rightharpoonup O_{m1}$.

$I_{m1a} + G_{m4} \rightharpoonup waste$

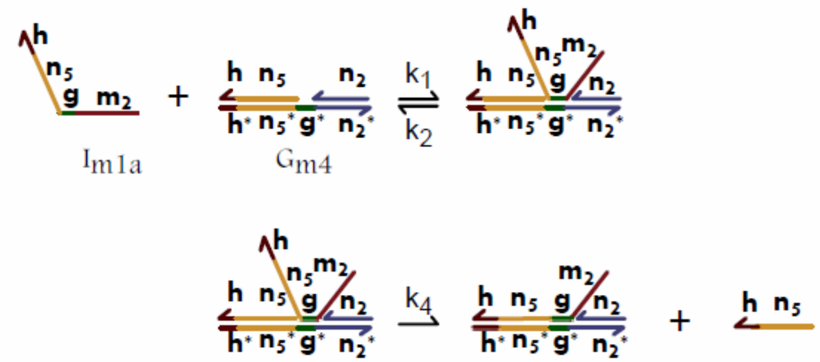
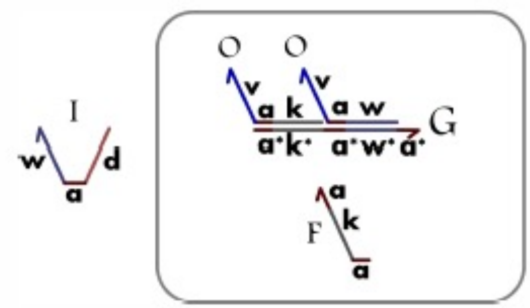


Figure 9: Reactions to implement $I_{m1a} + G_{m4} \rightharpoonup waste$.

Mechanism of Multiplication Gate: Group IV Reactions

$$\mathbf{IV}: O_{m1} + Amplifier \rightharpoonup r_m O_m \quad (7)$$

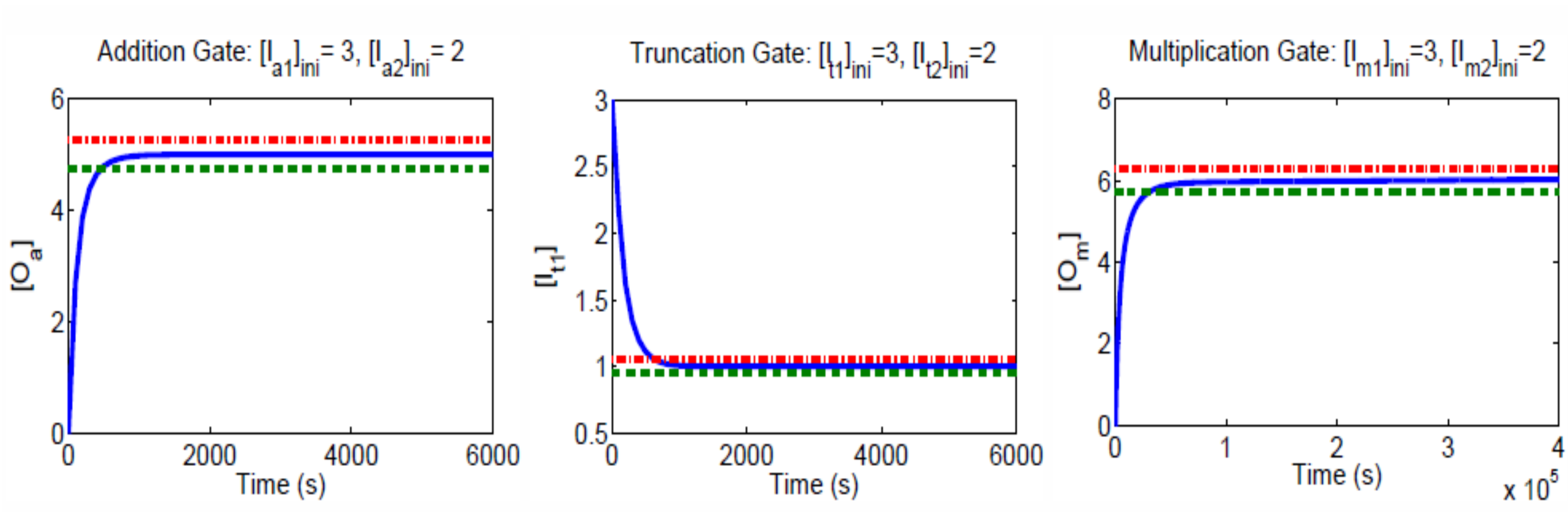
Design of 2x Amplifier



Software Simulations of Analog DNA Circuits

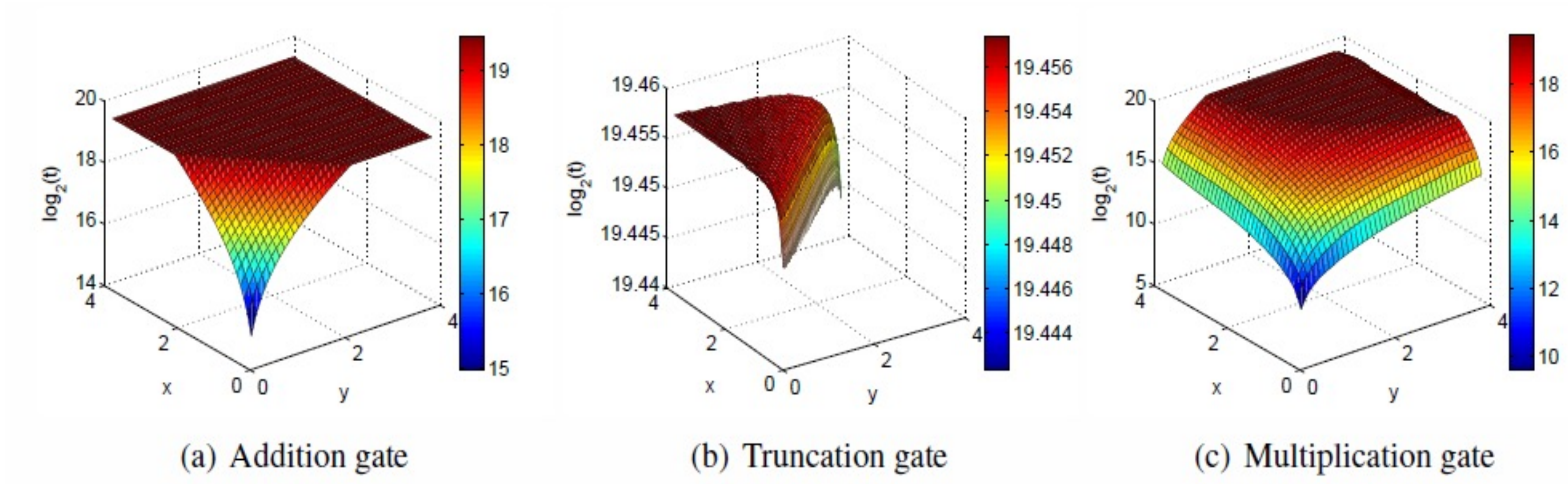
- **Software:** Visual GEC and Matlab
- **Rate Constants:**
  - toehold binding $2*10^{-3}$ nM^{-1} s^{-1},
  - toehold unbinding 10 s^{-1},
  - branch migration $8000/x^2$ s^{-1},
  - branch migration with mismatch $0.01*8000/x^2$ s^{-1},
  - where x is the length (number of nucleotides) of branch migration domain.
  - Leak rate constant 10^{-9} nM^{-1} s^{-1}.
  - From *[Qian et al, Science, 2011] and [Zhang et al, JACS, 2009]*
- **Input Ranges:** (0, 1), (0, 2), (0, 4), later we will use gates with these input ranges to construct circuits
- **Valid Output Range:** [0.95*r, 1.05*r] where r is the exactly correct result.
- **Unit of Relative Concentration:** 5 nM
- **Simulated time:** 720000 seconds (200 hours)
- **Benchmark:** the time that output stays in valid output range during simulated time.

Software Simulations of Analog DNA Circuits



- **Examples to show execution of our gates when the input range is (0;4).**
- **The vertical axes represent relative concentrations of output species.**
- **The ranges between the red and green dotted lines are valid output ranges.**
- **We did not show the curves for the whole simulated period (7.2*10^5 seconds) for the convenience to see the shape of the curves at early stage.**

Software Simulations of Analog DNA Circuits



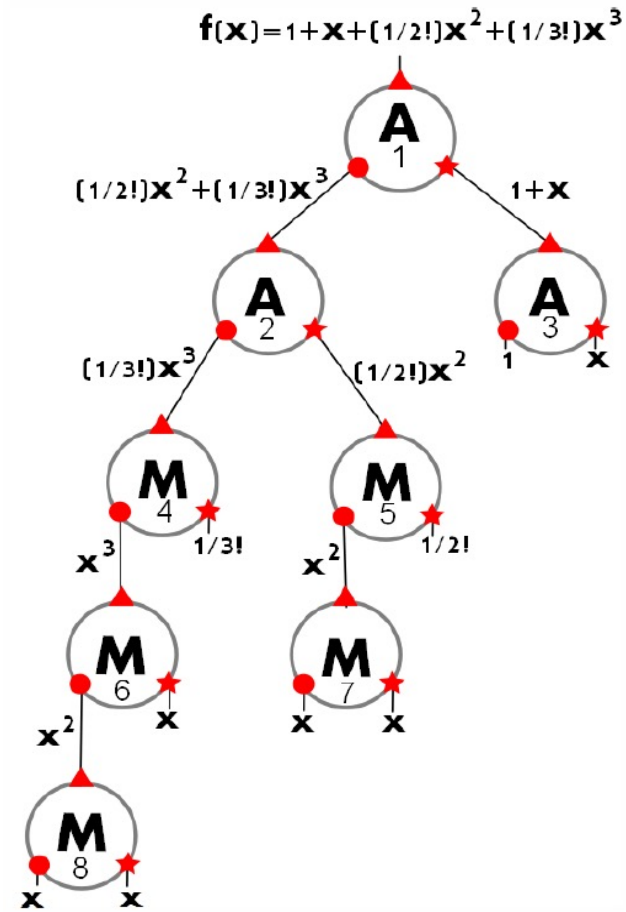
- Performance of the gates when input range is (0;4). x axis represents I_{a1}, I_{t1} or I_{m1}. y axis represents I_{a2}, I_{t2} or I_{m2}. Vertical axis represents $\log_2(t)$ where t is the time (seconds) that the output signal of corresponding inputs stays within the valid output range. $\log_2(t)$ is used in place of t simply for convenience in plotting. We can see the output signals stay longer in their valid output ranges when the input combinations produce larger outputs because the leak is relatively smaller.

Analog DNA Circuits

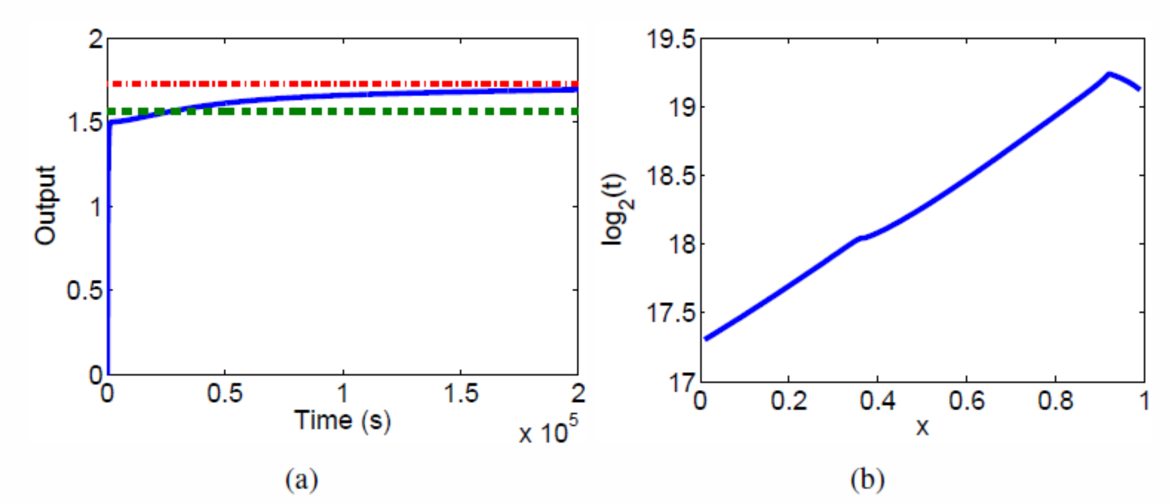
- **Circuits to Compute Polynomial Functions:**
  - **Strategy:** connect gates together by simply programming the sequence of output strand of a gate.
  - **Potential Problem:** static input vs. dynamic input
  - **Tricks:** use static input for I_{m2} input of all multiplication gates to set up the concentration ratio between G'_{m3} and G_{m4} as soon.
- **Beyond Polynomial Functions:**
  - **Taylor Series and Newton Iteration**

DNA Circuit to Compute $f(x) = 1 + x + x^2/2! + x^3/3!$, $0 < x < 1$

- The formula assigned to each wire describes the signal that goes through it.
- Each gate is assigned a number for the convenience to describe the circuit design.
- The input range:
  - The input range of gate-2, gate-4, gate-5, gate-6, gate-7 and gate-8 is 1.
  - The input range of gate-1 is 4.
  - The input range of gate-3 is 2.
  - The input ranges: are determined by the upper bound of input signals of a gate and we use the gates with the input ranges that we tested by simulation.



Software Simulation Results for a DNA Circuit to Compute $f(x) = 1 + x + x^2/2! + x^3/3!$, $0 < x < 1$



(a) Execution of the circuit to compute f (x) when x = 0.5.

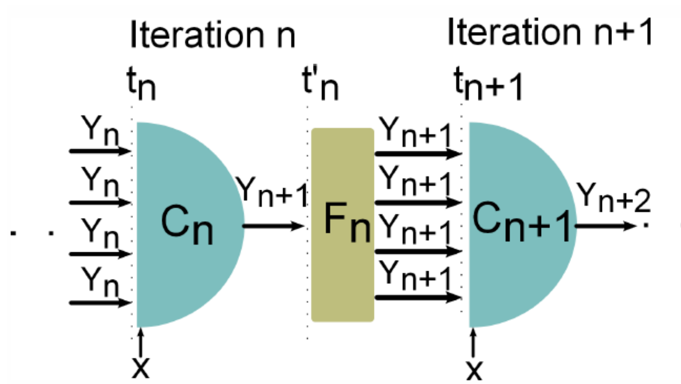
- We did not show the curve for the whole simulated period ($7.2*10^5$ seconds) for the convenience to see the shape of the curve at early stage.

(b) Performance of the circuit to compute f (x) where 0 < x < 1.

- t is the time (seconds) that the output signal stays in the valid output range.
- The valid output range is [0.95*f(x), 1.05*f(x)].
- Model of simulation is the same for simulating single gates.

DNA Circuits for Functions beyond Polynomials

- **Taylor Series:**
  - **$f(x) = 1 + x + x^2/2! + x^3/3!$ is a good approximation of e^x, when $0 < x < 1$.**
- **Newton Iteration:**
  - **polynomial function $F(Y_n) = Y_{n+1} = 2Y_n - (Y_n)^2 x$ is the formula of Newton Iteration to compute reciprocal (1/x), where $0 < x < 0.5$.**
  - **We can construct a circuit for each iteration and let iterations happen sequentially by techniques like optical activation.**



Future Work on Analog DNA Circuits

- **Experimental demonstration of current design**
- **Speed up the computation**
- **Design with feedback loop**
- **More compact design for analytic functions like exponentiation and logarithm**