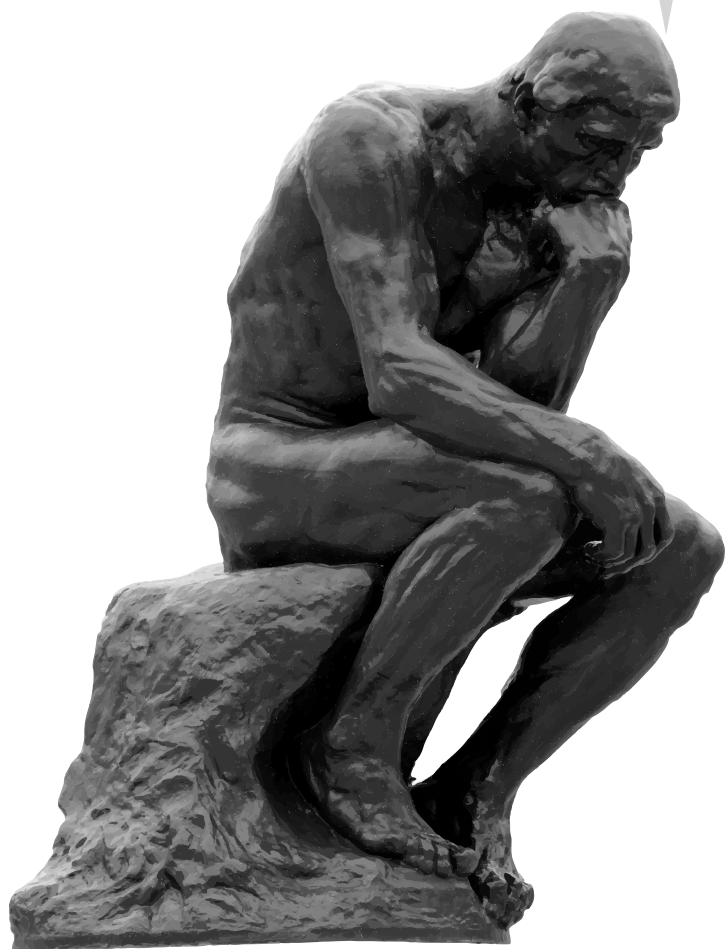
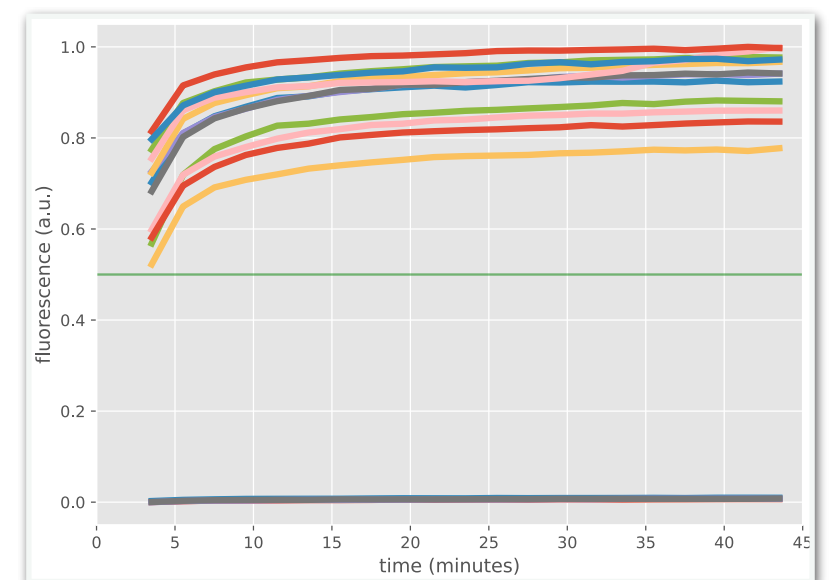
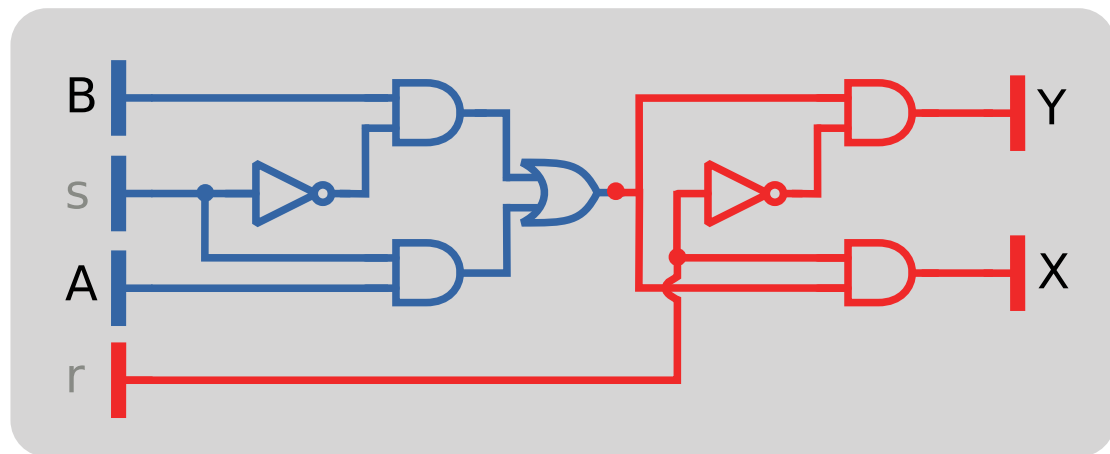


Theory & *Practice* of DNA strand displacement circuits

October 8, 2018 @ DNA 24

Chris Thachuk
Winfree Lab, California Institute of Technology

Today's tutorial in a nutshell



Molecular Circuits

Built upon DNA strand displacement cascades

Input

5'–ACCACGATCACATTAC–3'

5'–GCAACATACAT–3'

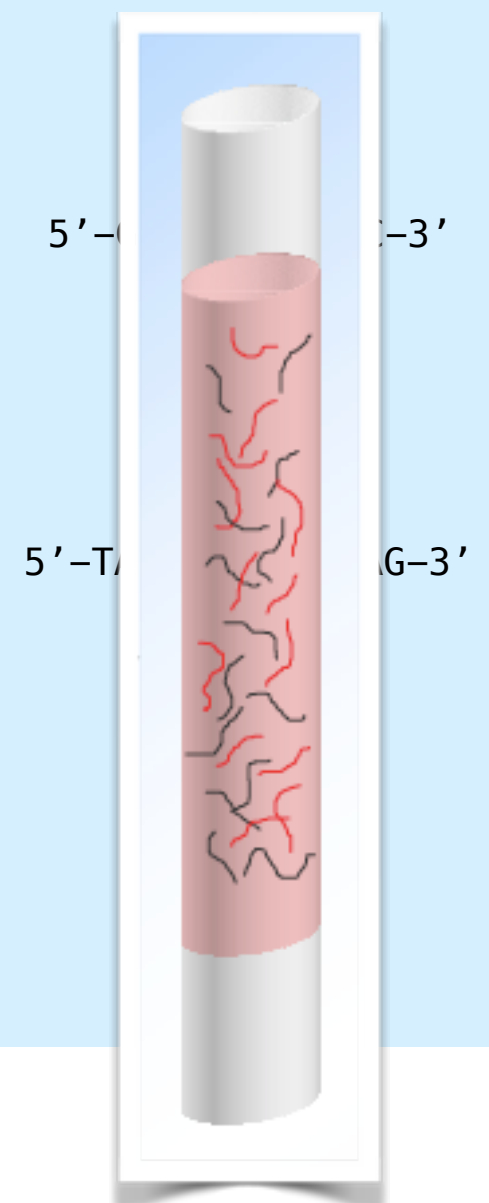
5'–CCCATACATCACCAG–3'

5'–TACCACATGAGCAGCA–3'

Computation

DNA strand displacement cascades

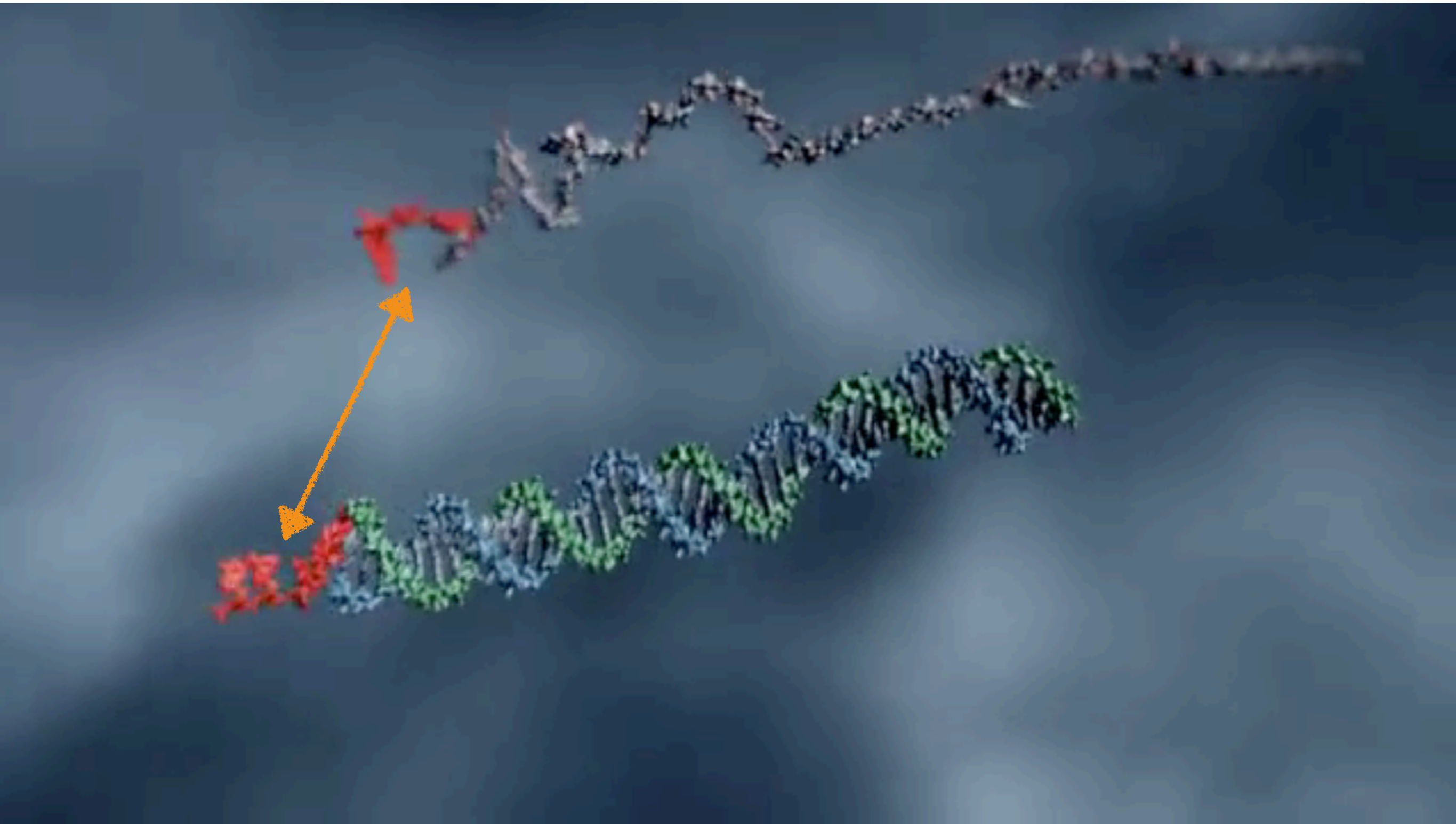
Output



Tutorial Outline

- ▶ **Review of strand displacement**
- ▶ Building and composing logic gates
- ▶ Tools for designing and verifying circuits
- ▶ Robustness of strand displacement

Review of DNA Strand Displacement (DSD)

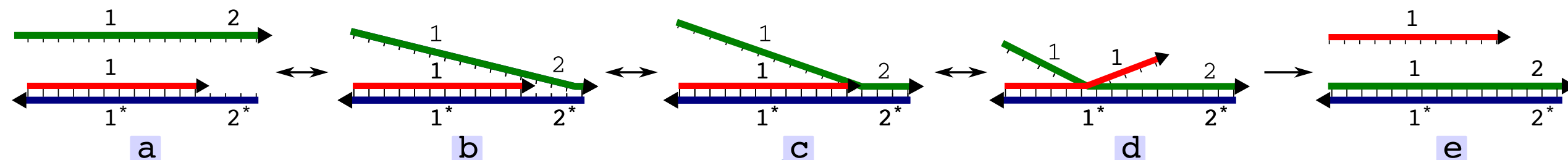


B. Yurke, A. J. Turberfield, A. P. Mills Jr., F. C. Simmel, J. L. Neumann, *Nature* 406, 605 (2000).

A. J. Turberfield et al., *Phys. Rev. Lett.* 90, 118102

Video Courtesy of Biocomputation Group @ Microsoft Research

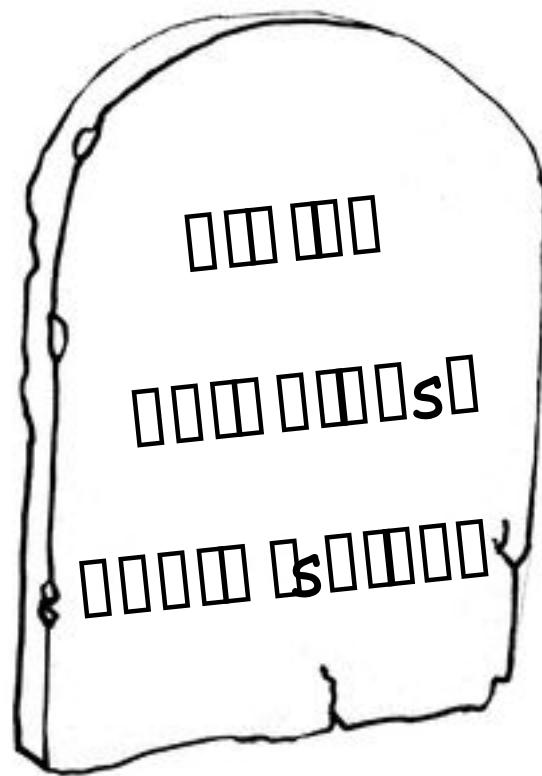
Adapted from Zhang & Seelig 2011



Strand Displacement Cascades

=

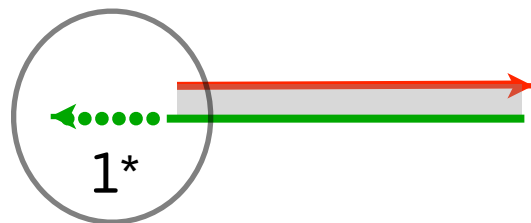
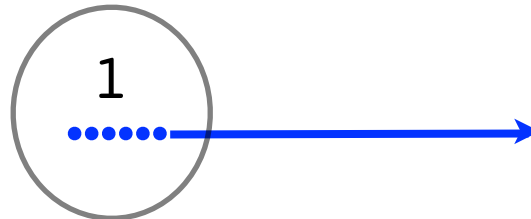
Three Rules



Domain level rules for DSD

Rule 1: *Bind*

Example



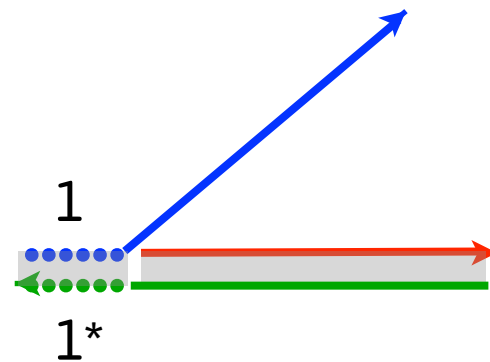
single-stranded
complementary
domains

Domain level rules for DSD

Rule 1: *Bind*

Two single-stranded complementary domains can bind

Example

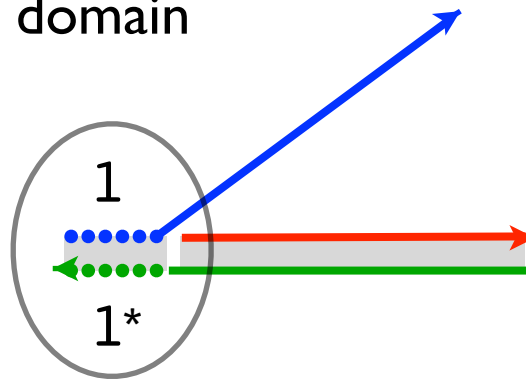


Domain level rules for DSD

Rule 2: *Release*

Example

blue strand bound by only
a short domain



Domain level rules for DSD

Rule 2: ***Release***

Any strand bound by only a short domain can **release**

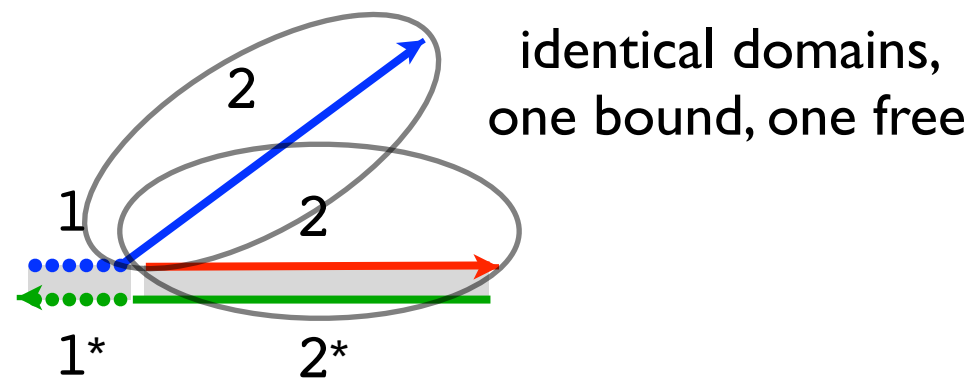
Example



Domain level rules for DSD

Rule 3: *Displace*

Example

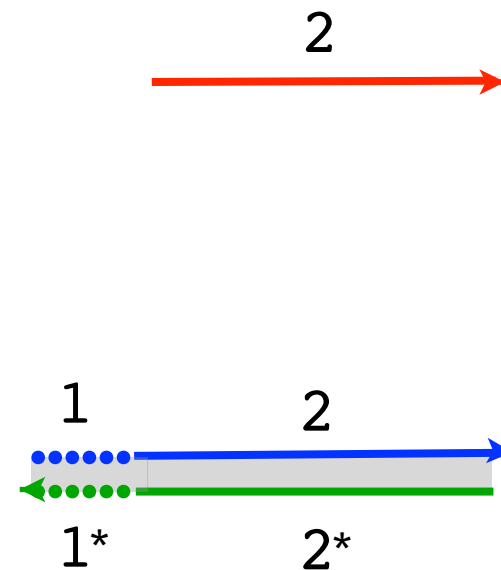


Domain level rules for DSD

Rule 3: ***Displace***

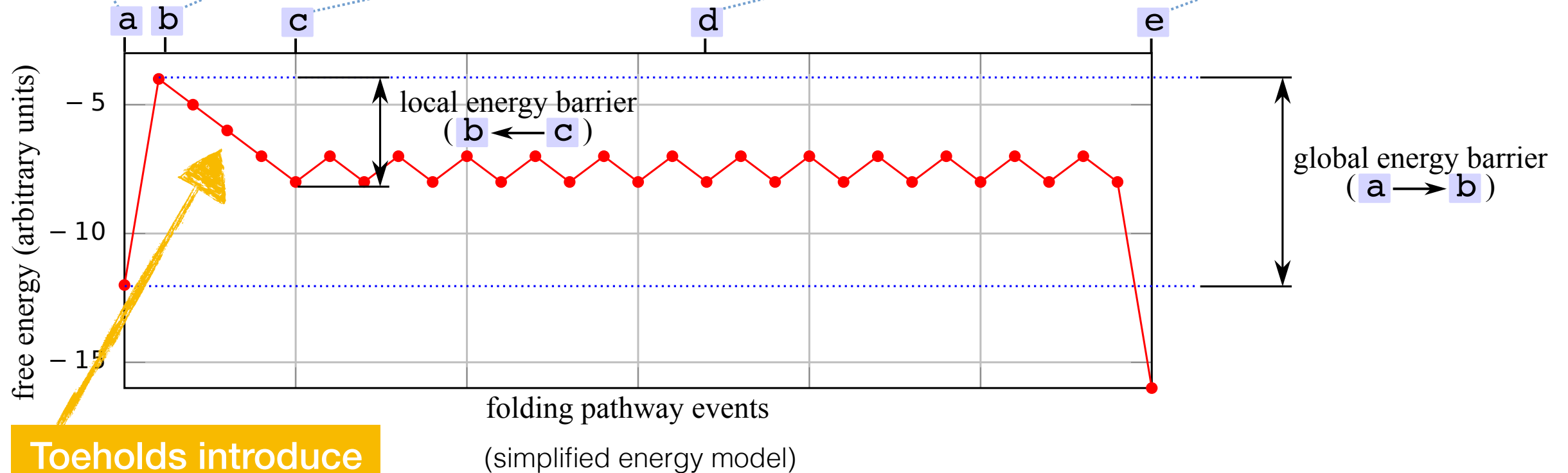
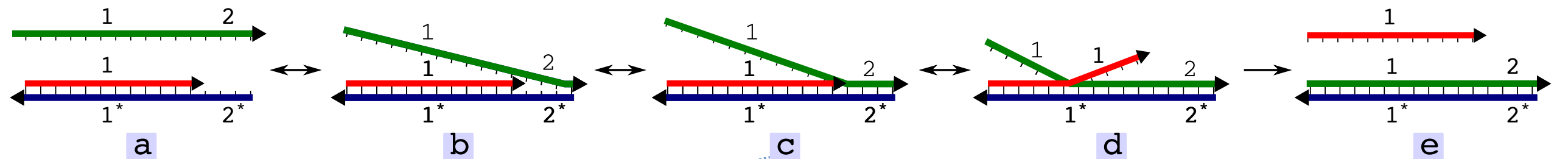
A domain can **displace** an identical domain of another strand, *if neighboring domains are already bound*

Example



Why do we use toeholds?

Adapted from Zhang & Seelig 2011

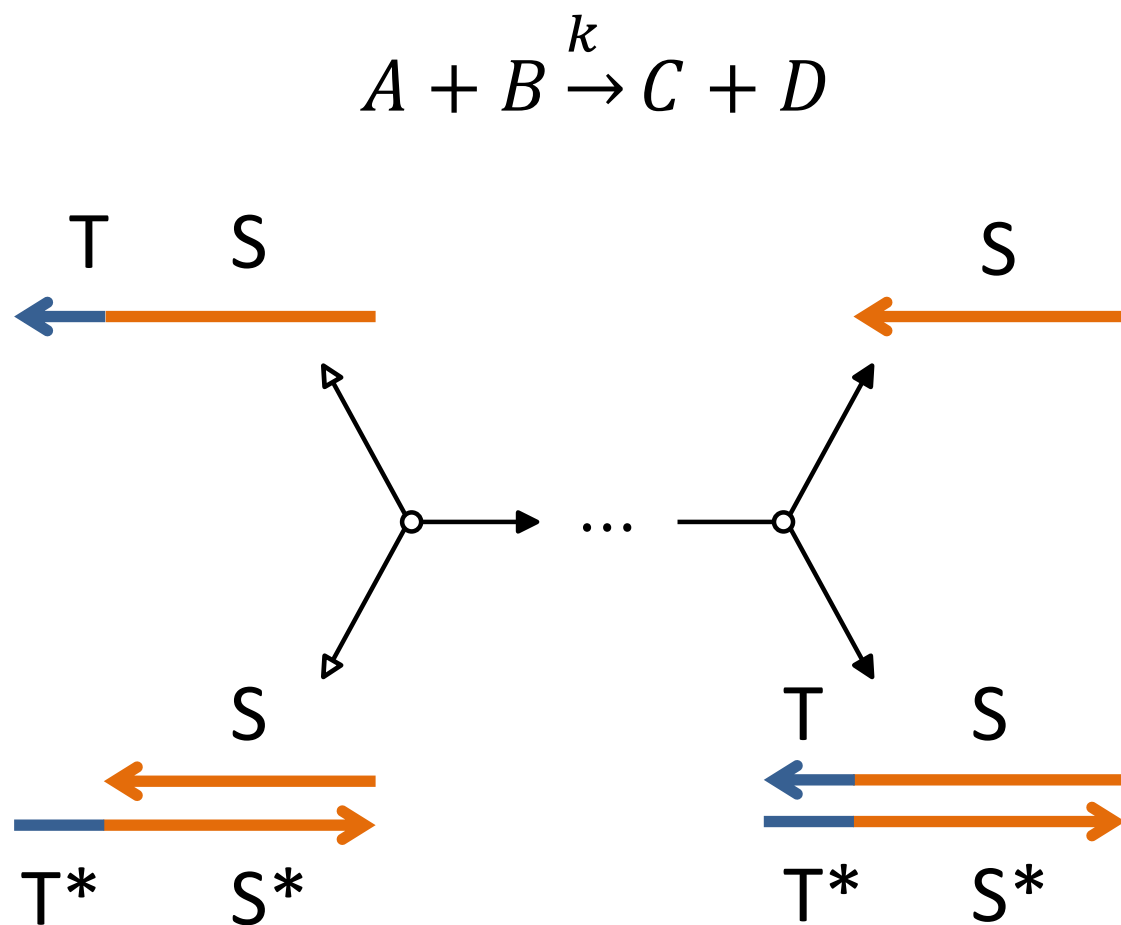


Toeholds introduce energy barriers

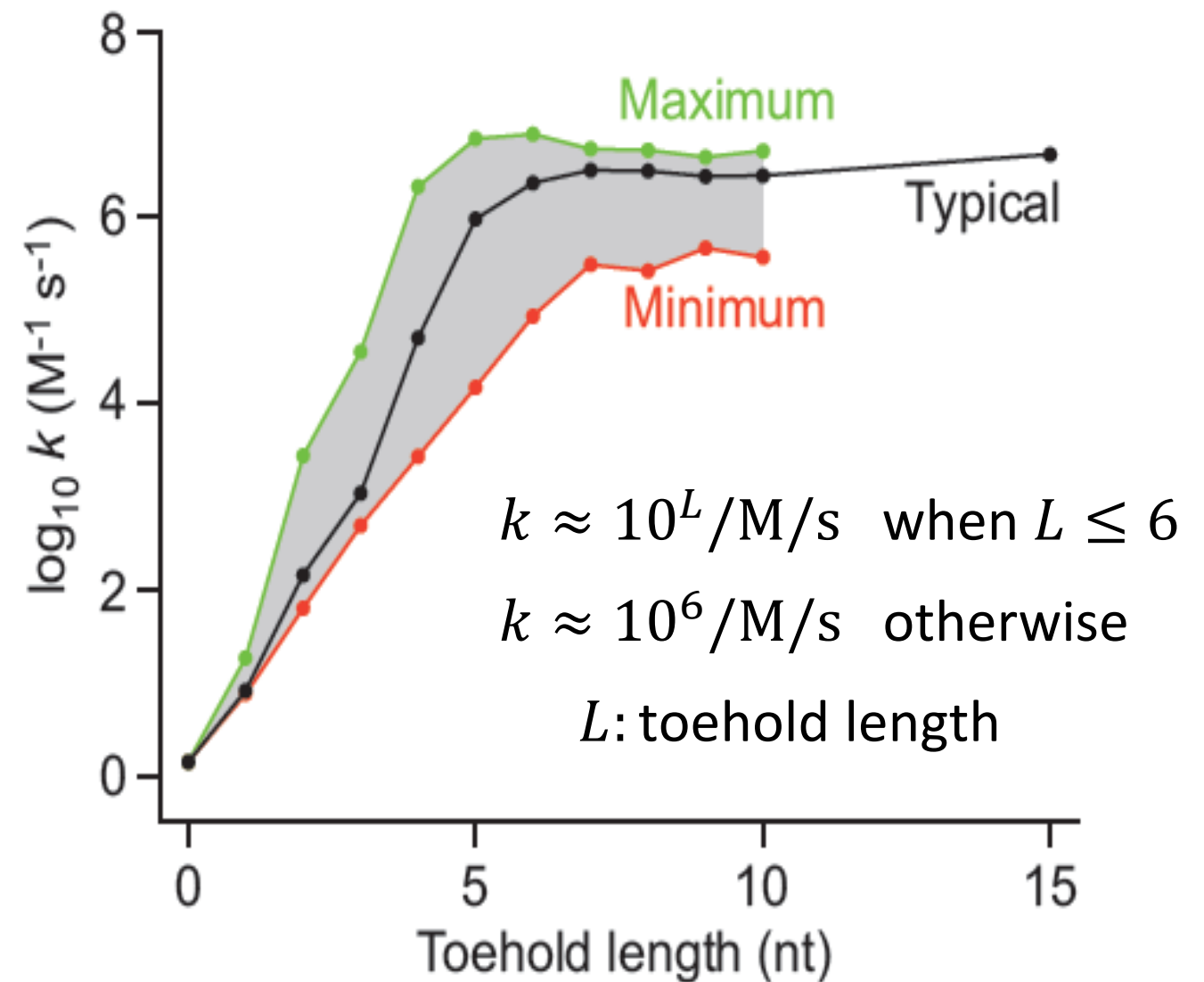
Toehold-mediated DNA strand displacement

T: toehold domain (typically 3-7 nucleotides)

S: branch migration domain (typically 15-20 nucleotides)

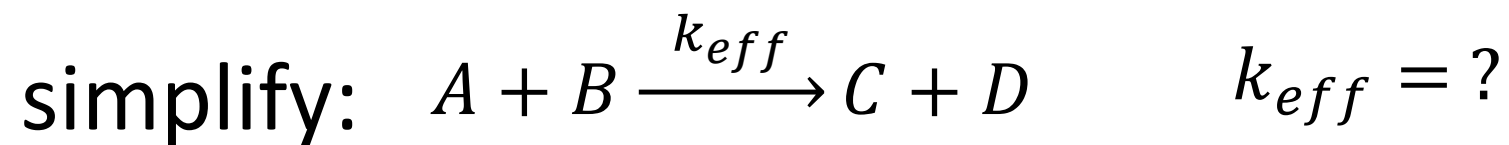
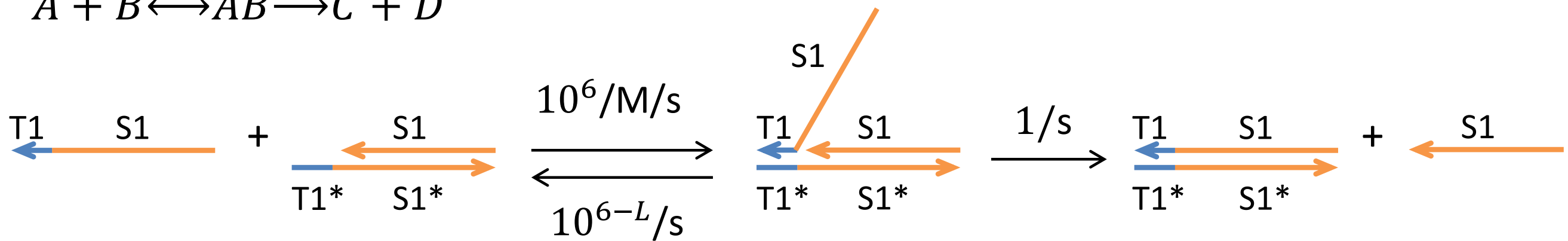


The rate of strand displacement grows exponentially with toehold length for short toeholds.



Zhang and Seelig, *Nature Chemistry* 2011

Kinetics of toehold-mediated strand displacement



collision rate: $10^6[A][B]$

collision success probability: $\frac{1/s}{1/s + 10^{6-L}/s}$

net rate of success: $10^6 \cdot \underbrace{\frac{1}{1 + 10^{6-L}}}_{k_{eff}} [A][B]$

$k_{eff} \approx 10^L/M/s$ when $L \leq 6$
otherwise $k_{eff} \approx 10^6/M/s$
 L : toehold length $|T1|$

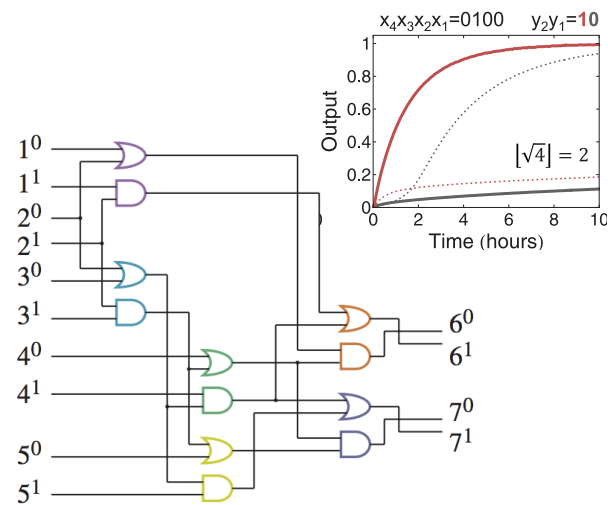
Zhang et al, *JACS* 2009

Srinivas et al, *NAR* 2013

This approximation is valid for low concentrations of A and B (e.g. $[A]=[B]=100\text{nM}$) such that the unimolecular reaction is sufficiently faster than the bimolecular reaction.

Strand displacement in the lab

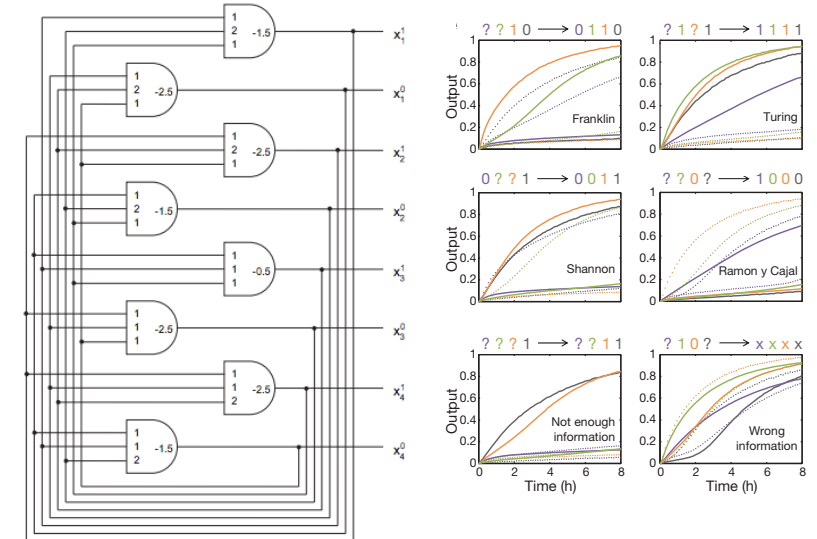
molecular logic circuits



- Large autonomous biochemical networks built from scratch

Qian, Winfree, *Science* 2011

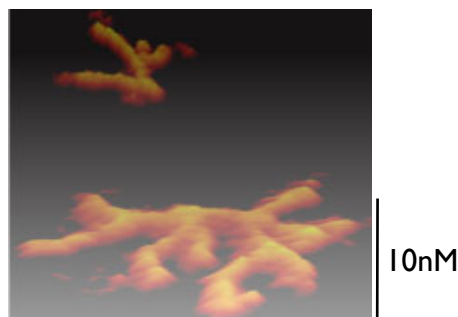
molecular artificial neural networks



- Biochemical system doing inference

Qian, Winfree, Bruck *Nature* 2011

controlling assembly of nanoscale structures

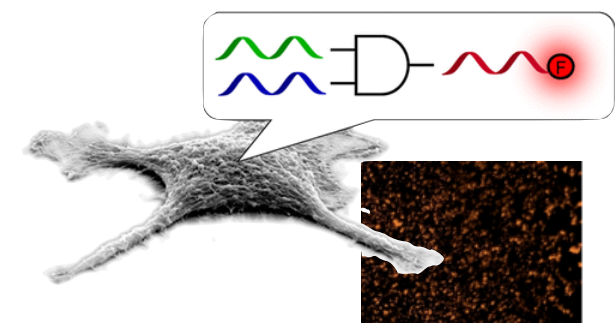


- Prescribed nanoscale structures seen under atomic force microscope

Yin, Choi, Calvert, Yurke, Pierce *Nature* 2008

strand displacement cascades

strand displacement in vivo



- Logic on biological signals

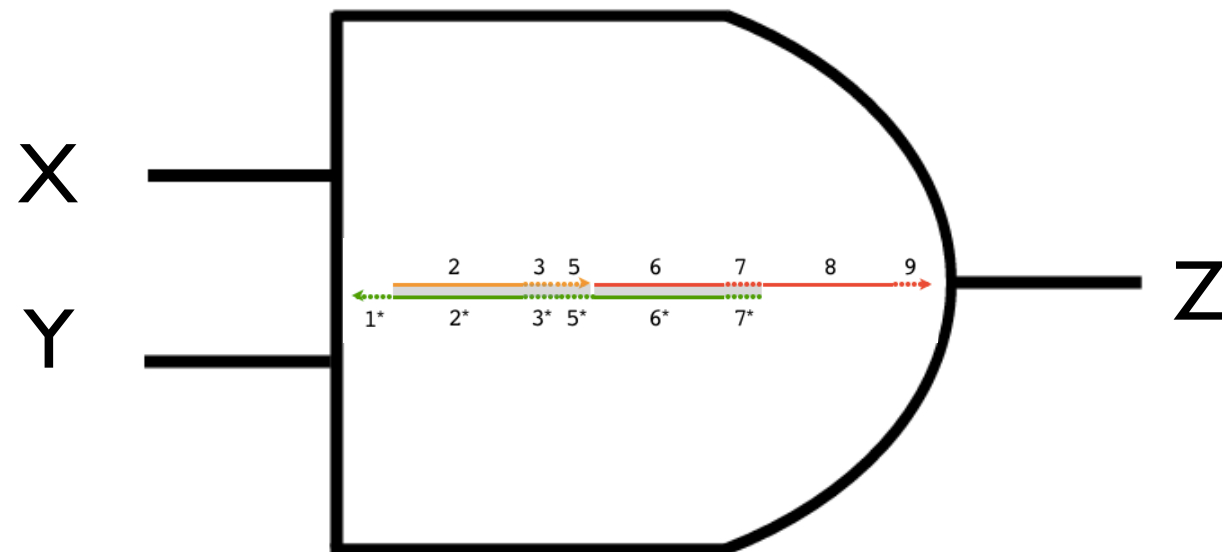
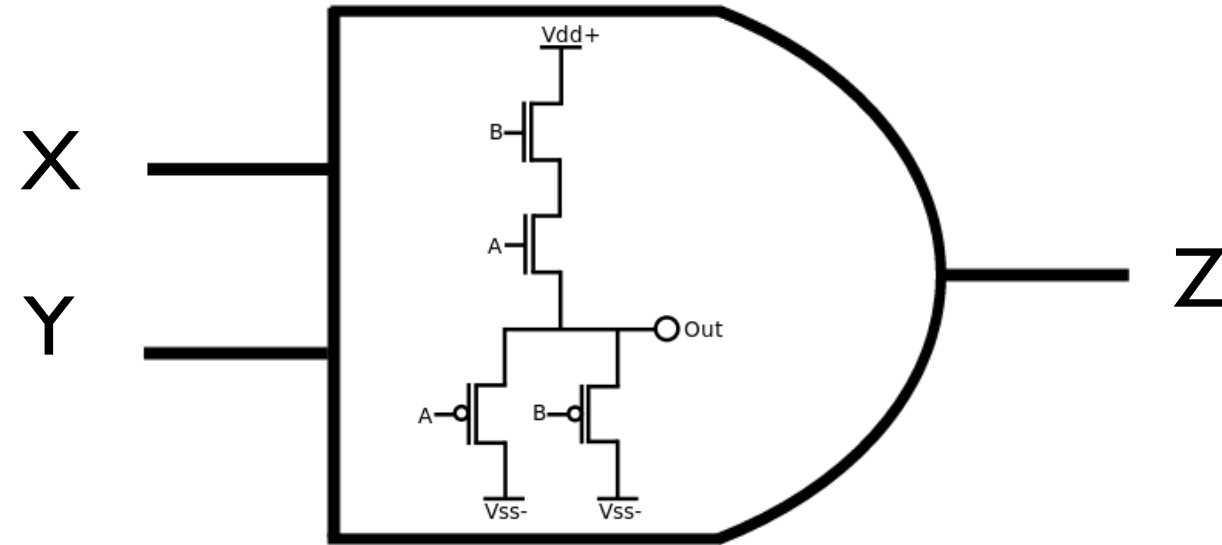
Hemphill, Deiters *J Am Chem Soc* 2013

Tutorial Outline

- ▶ Review of strand displacement
- ▶ **Building and composing logic gates**
- ▶ Tools for designing and verifying circuits
- ▶ Robustness of strand displacement

AND gate

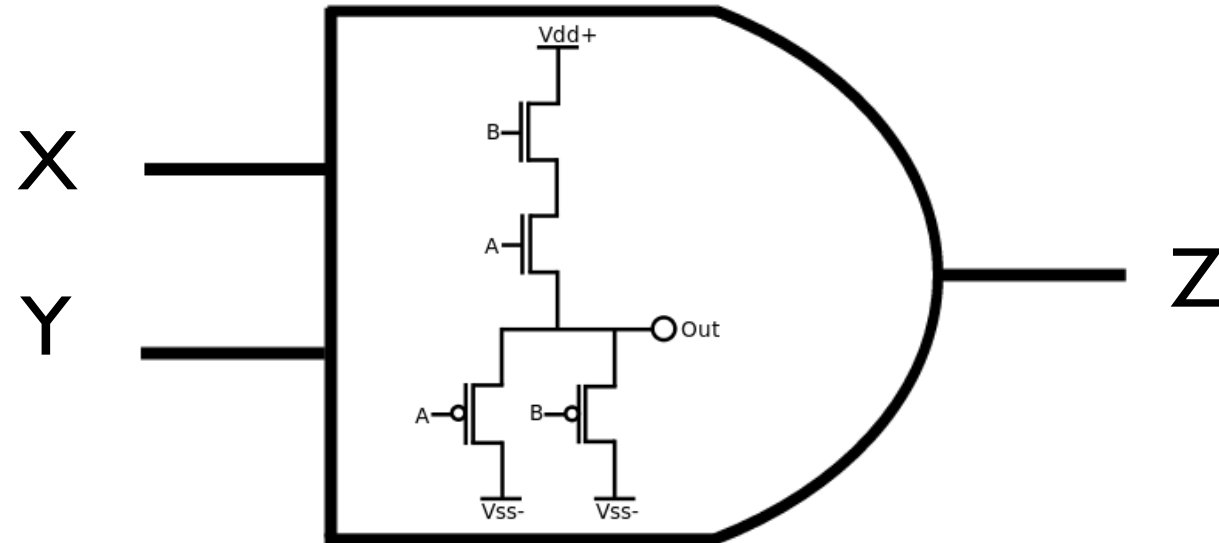
release Z if and only if X and Y are present



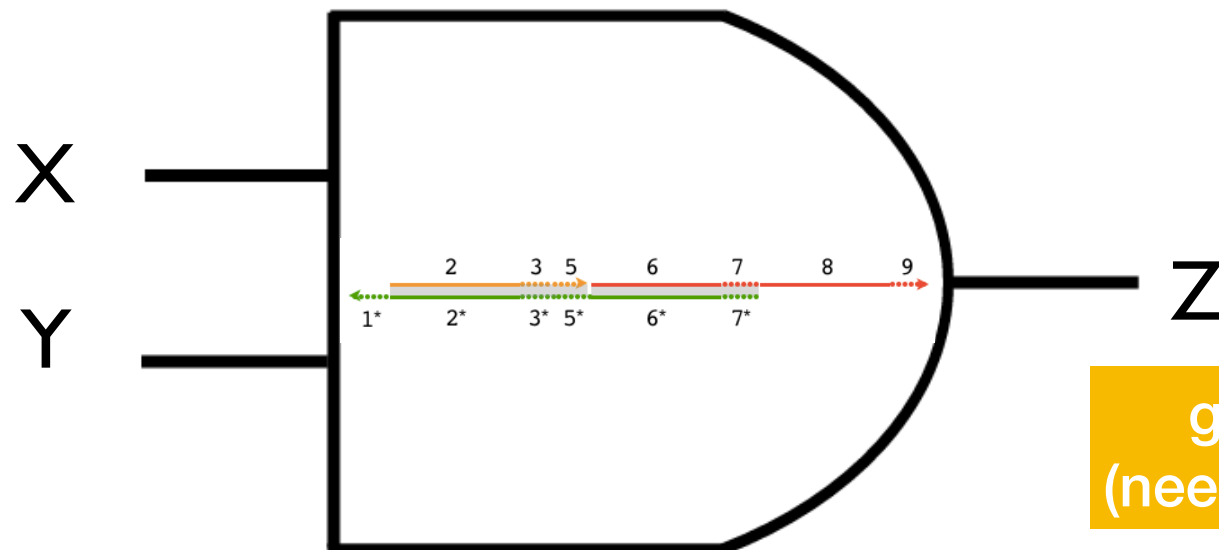
AND gate

release Z if and only if X and Y are present

voltages



strands

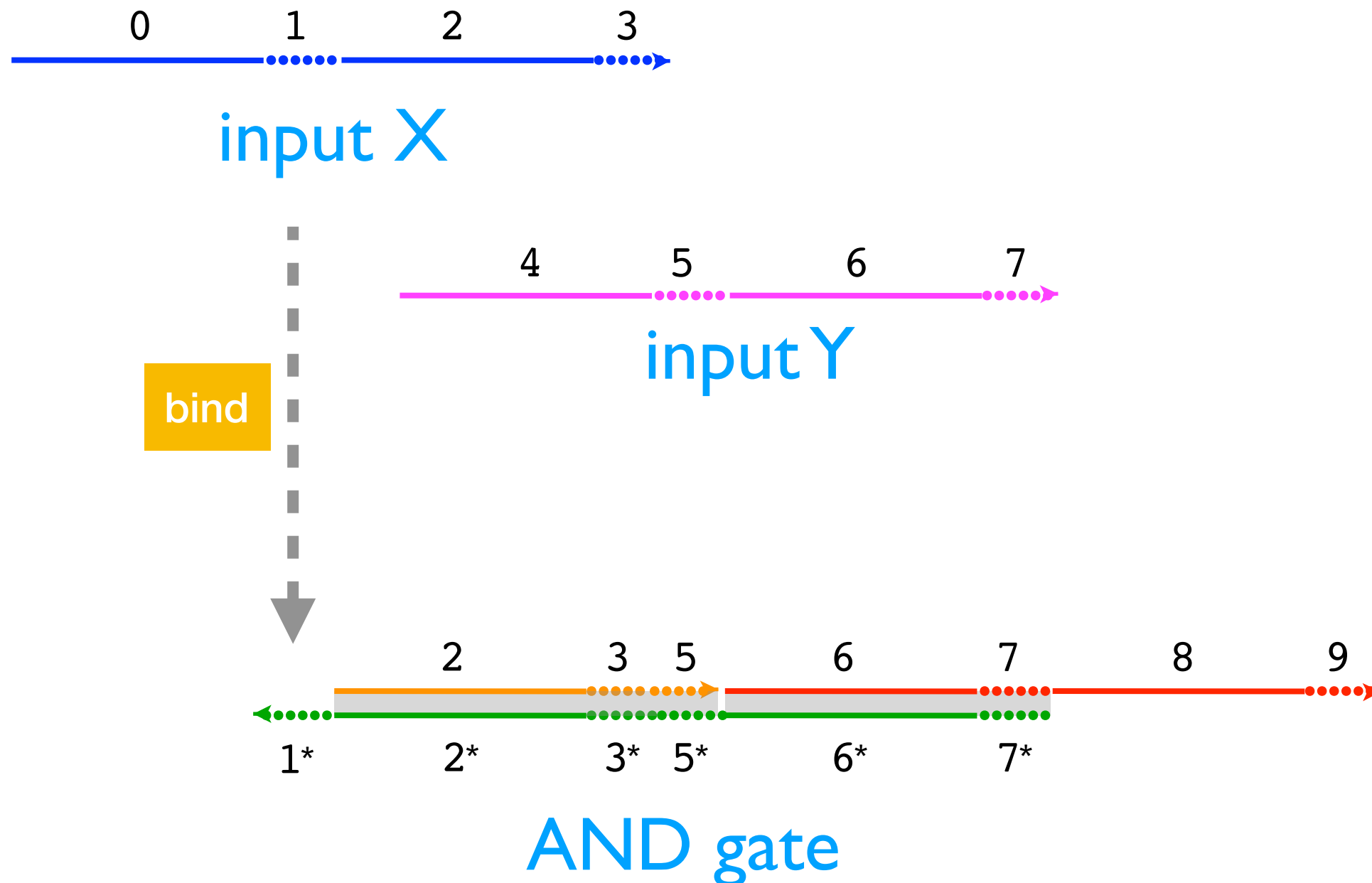


gates get consumed!
(need to have many copies)

gate=complex

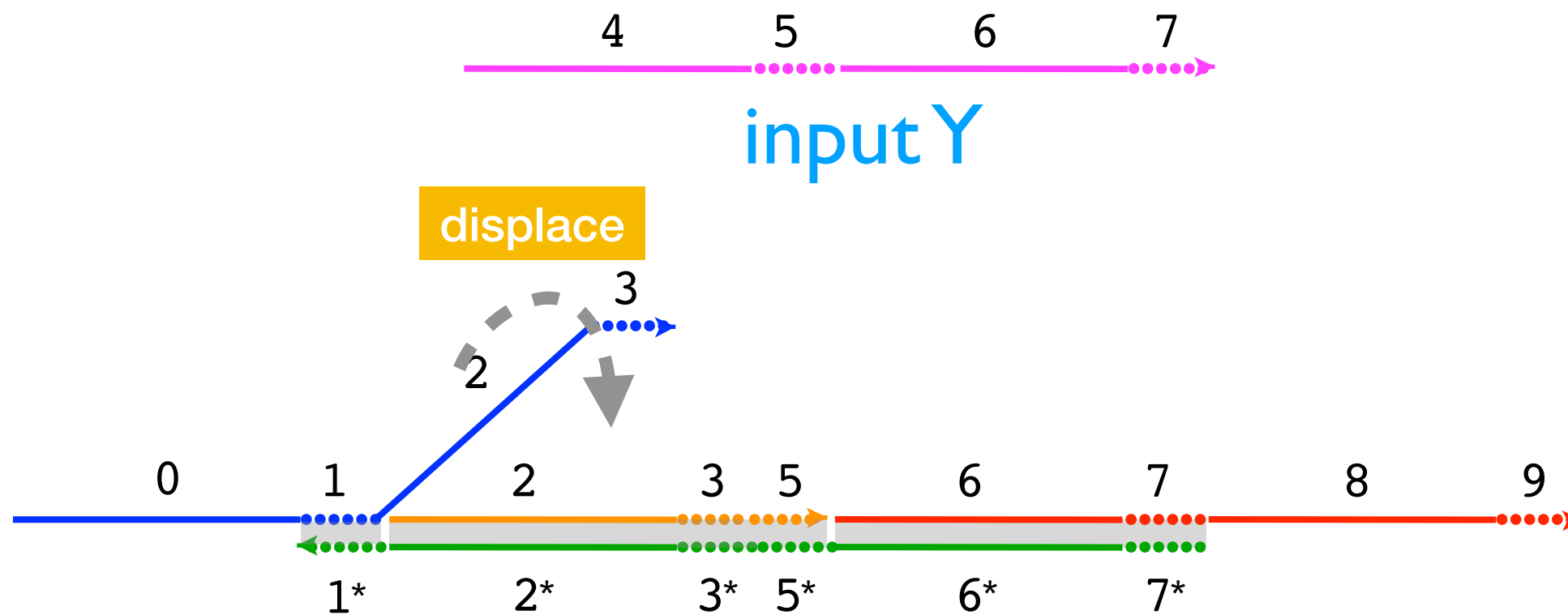
Strand Displacement Cascades Example: AND gate

release Z if and only if X and Y are present



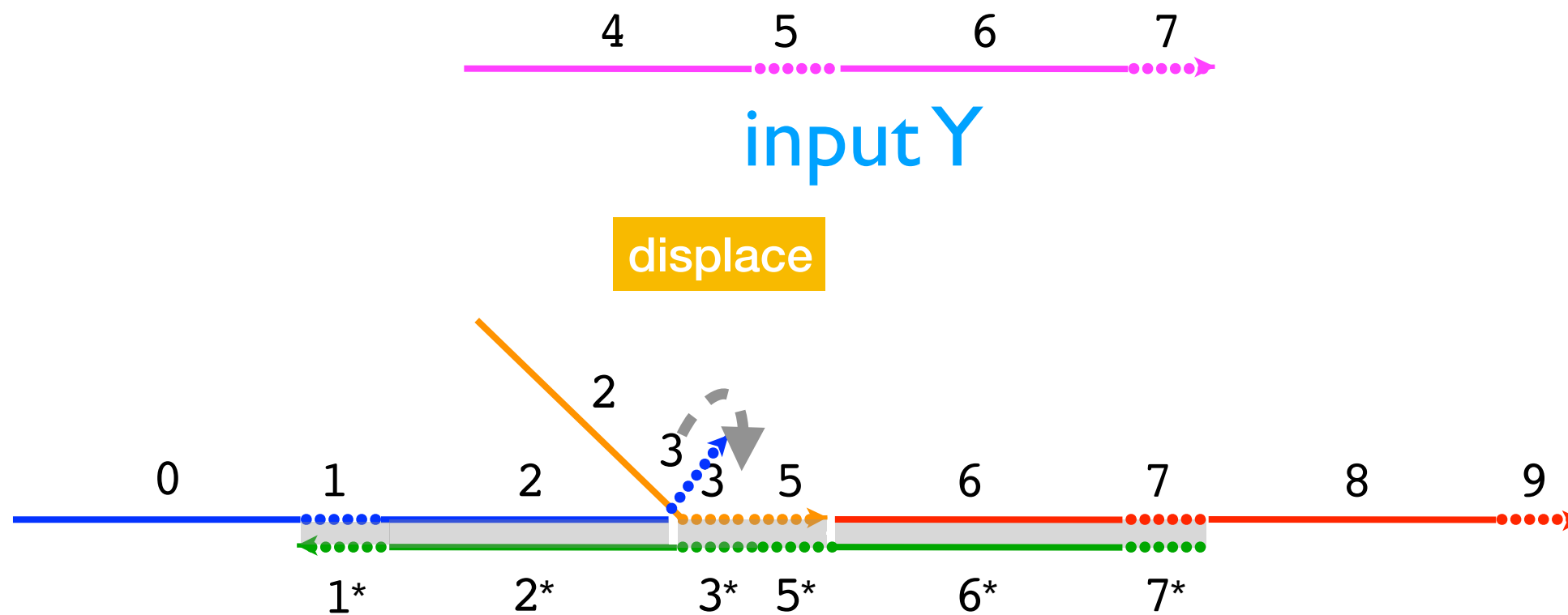
Strand Displacement Cascades Example: AND gate

release Z if and only if X and Y are present



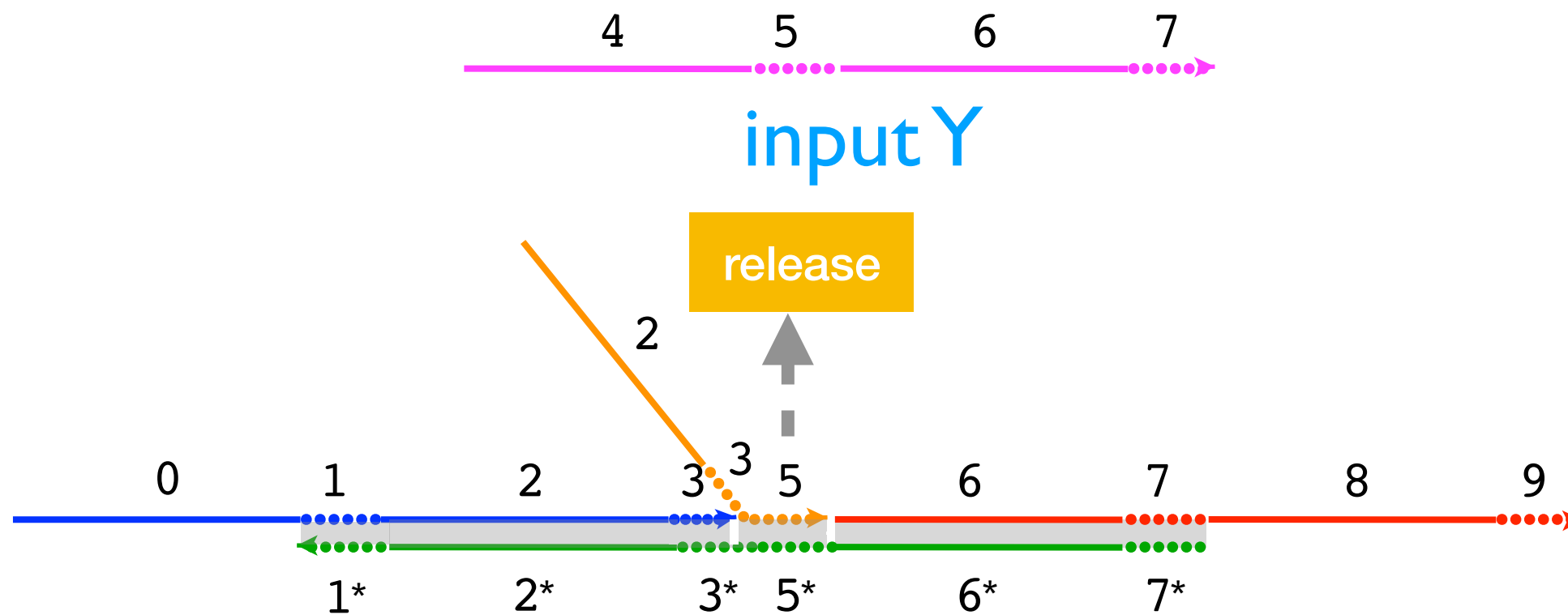
Strand Displacement Cascades Example: AND gate

release Z if and only if X and Y are present



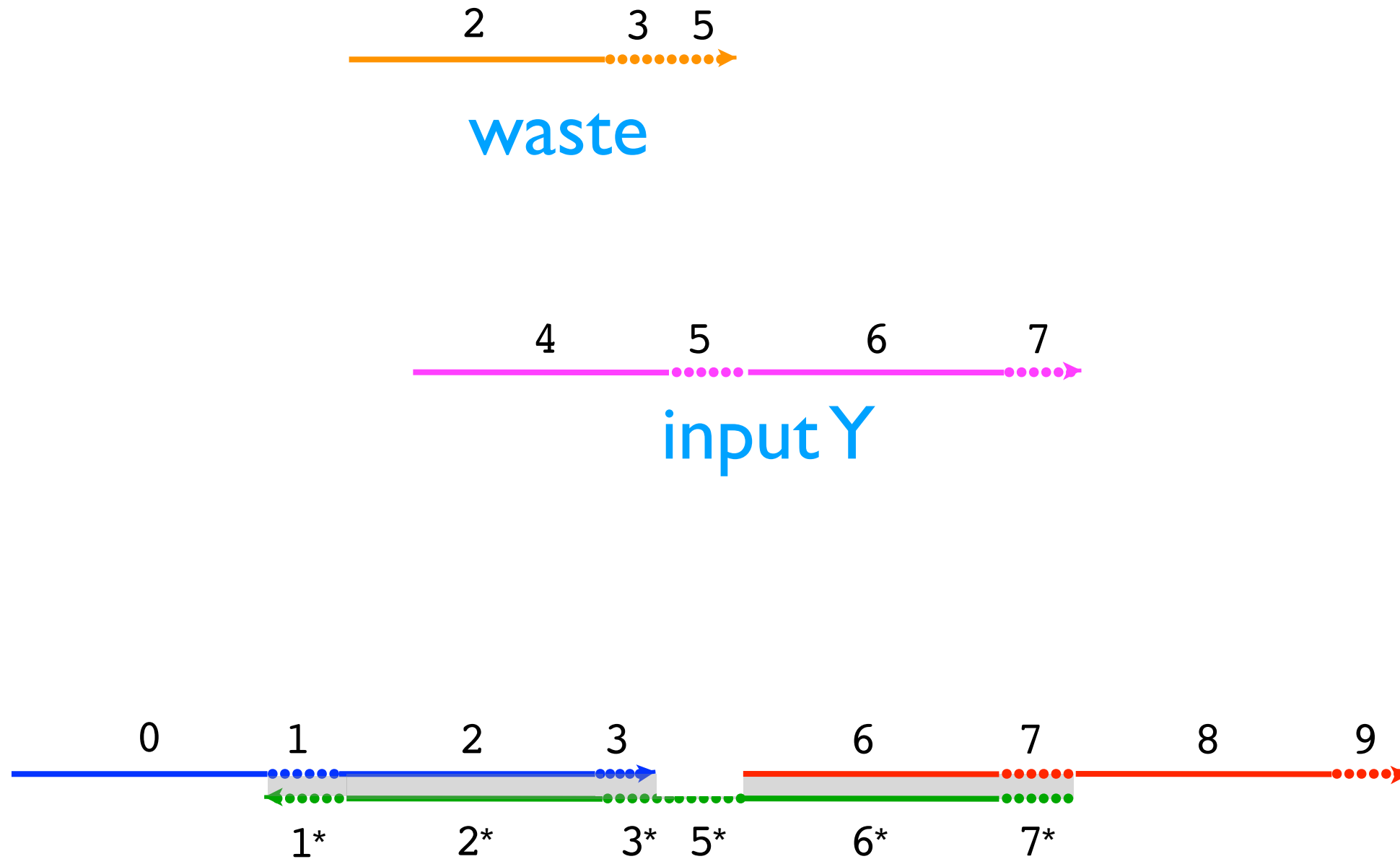
Strand Displacement Cascades Example: AND gate

release Z if and only if X and Y are present



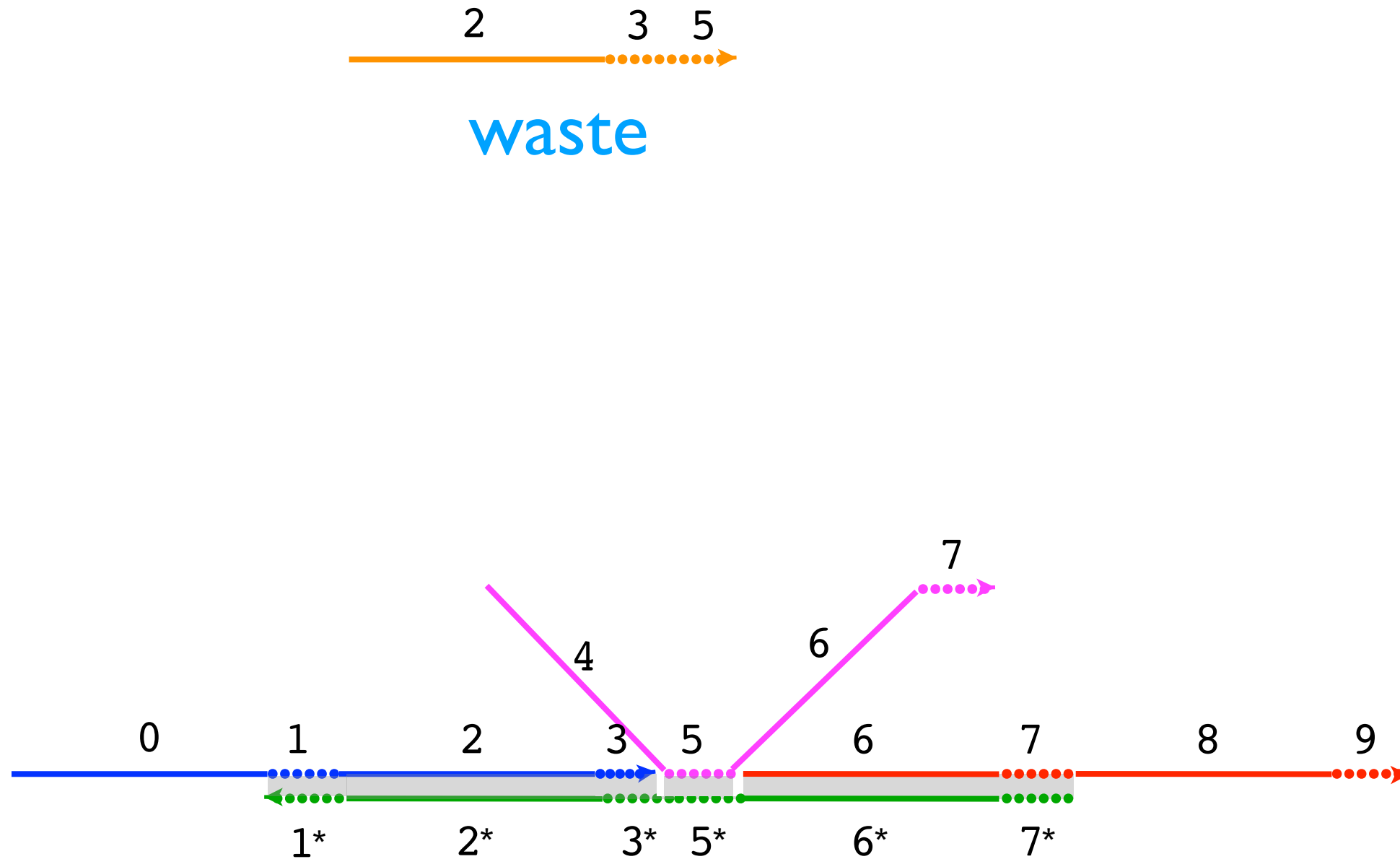
Strand Displacement Cascades Example: AND gate

release Z if and only if X and Y are present



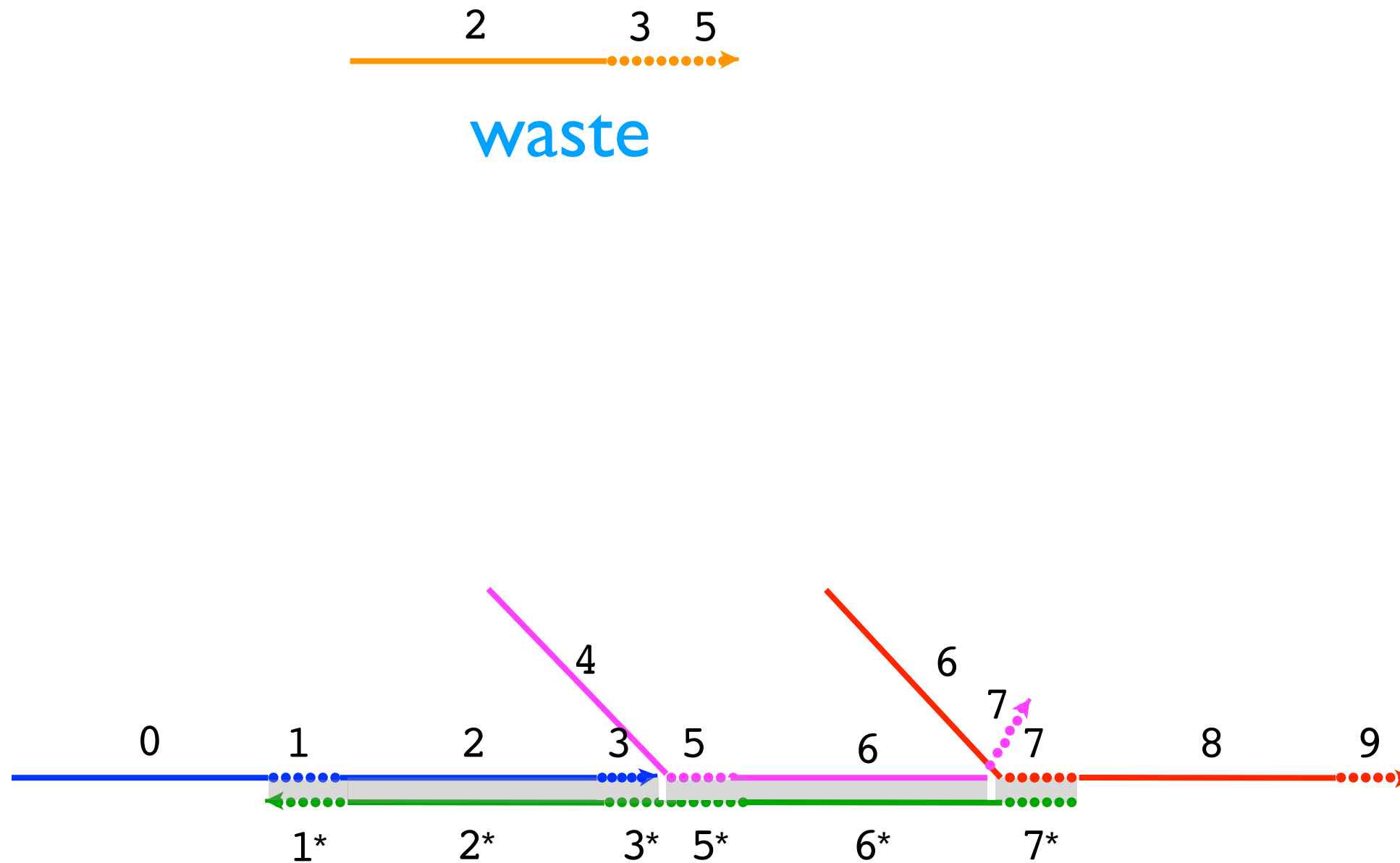
Strand Displacement Cascades Example: AND gate

release Z if and only if X and Y are present



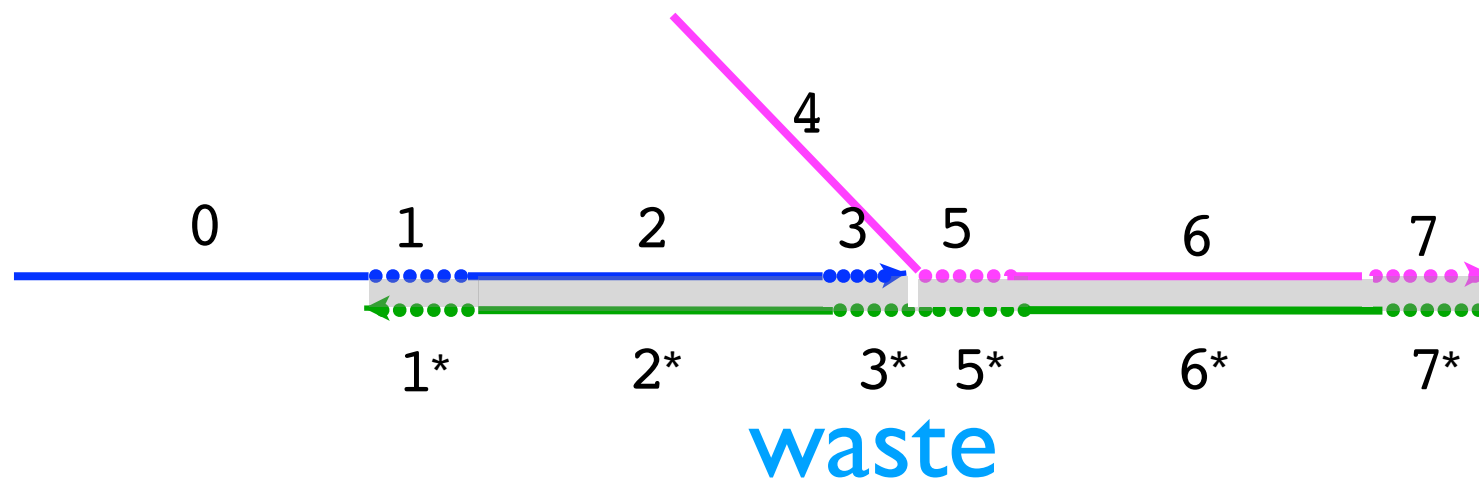
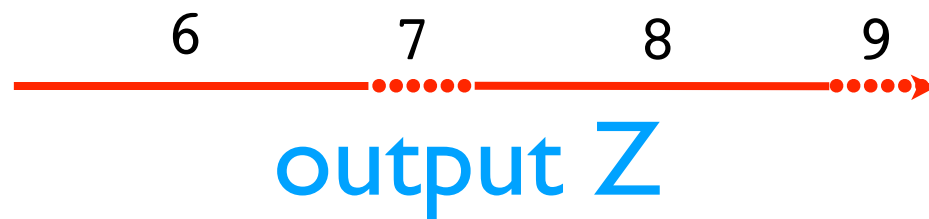
Strand Displacement Cascades Example: AND gate

release Z if and only if X and Y are present



Strand Displacement Cascades Example: AND gate

release Z if and only if X and Y are present



Strand Displacement Cascades Example: AND gate

release Z if and only if X and Y are present

before

after

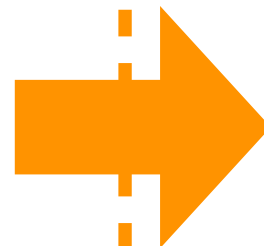


input X



input Y

Composable



gate is "used up"



output Z



waste

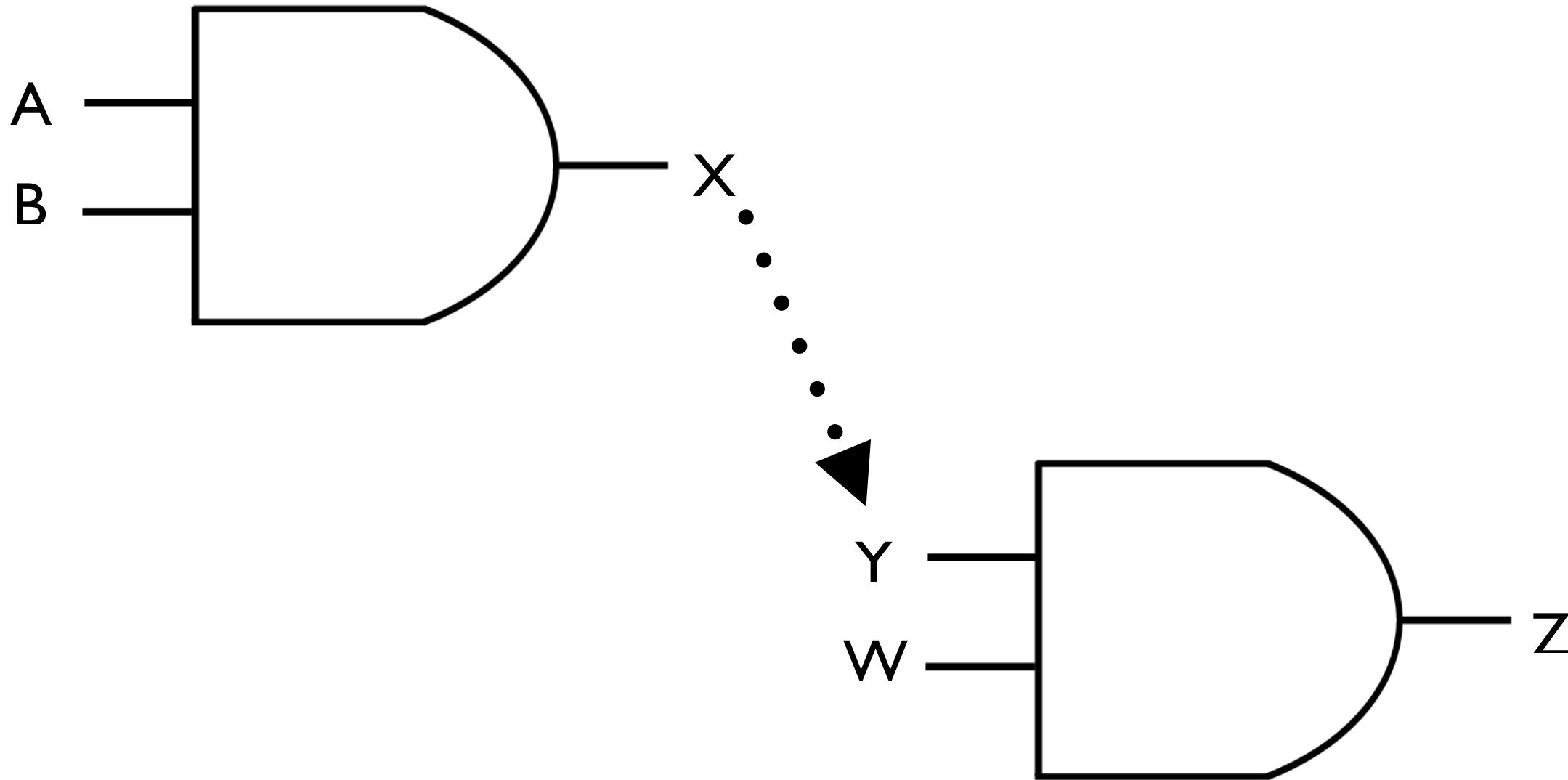


AND gate



waste

Composing AND gates



We need a “wire”

Sequence Independence

Translator (a “wire”): $X \rightarrow Y$

input X



output Y

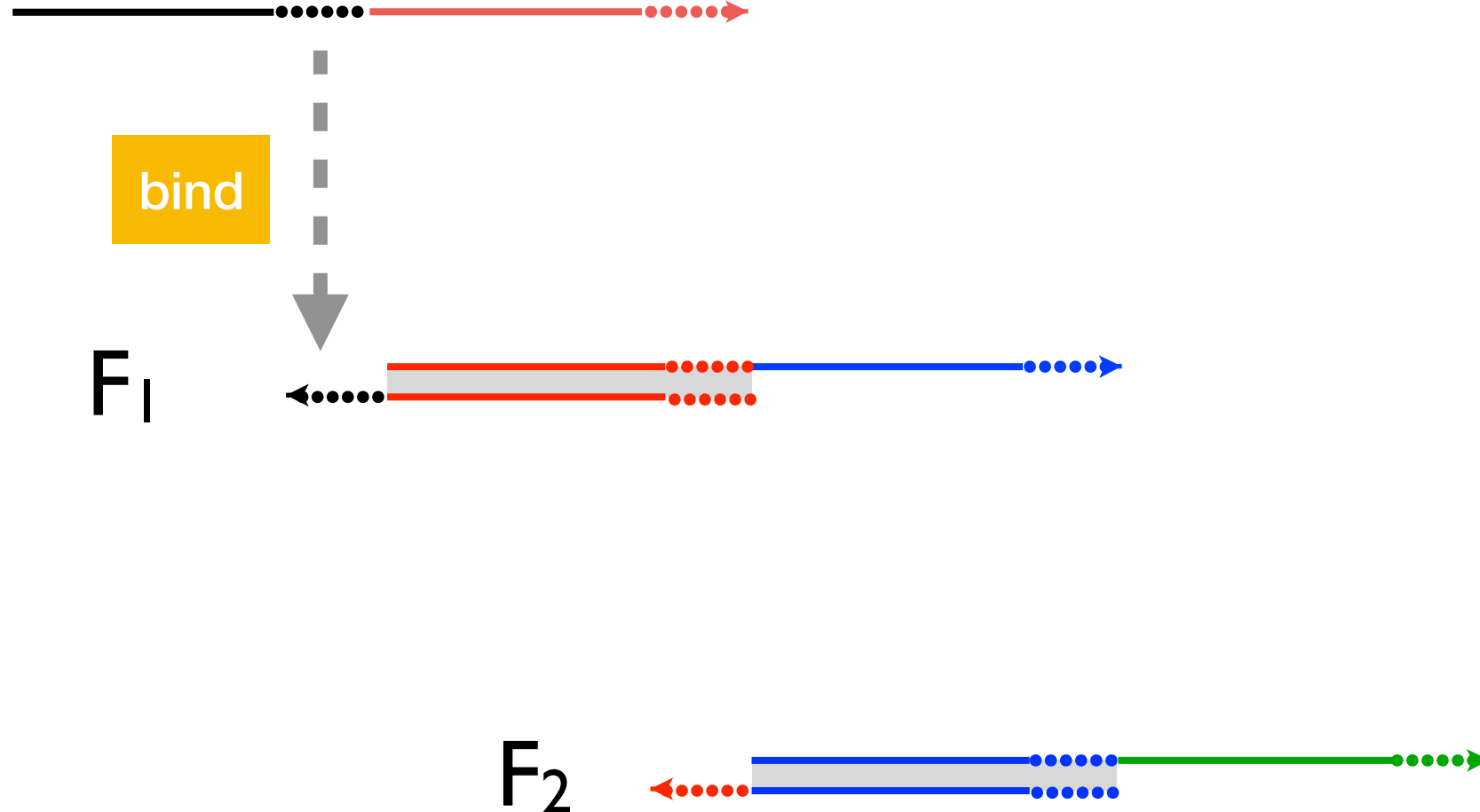


Different coloring scheme to emphasize sequence (in)dependence!

Sequence Independence

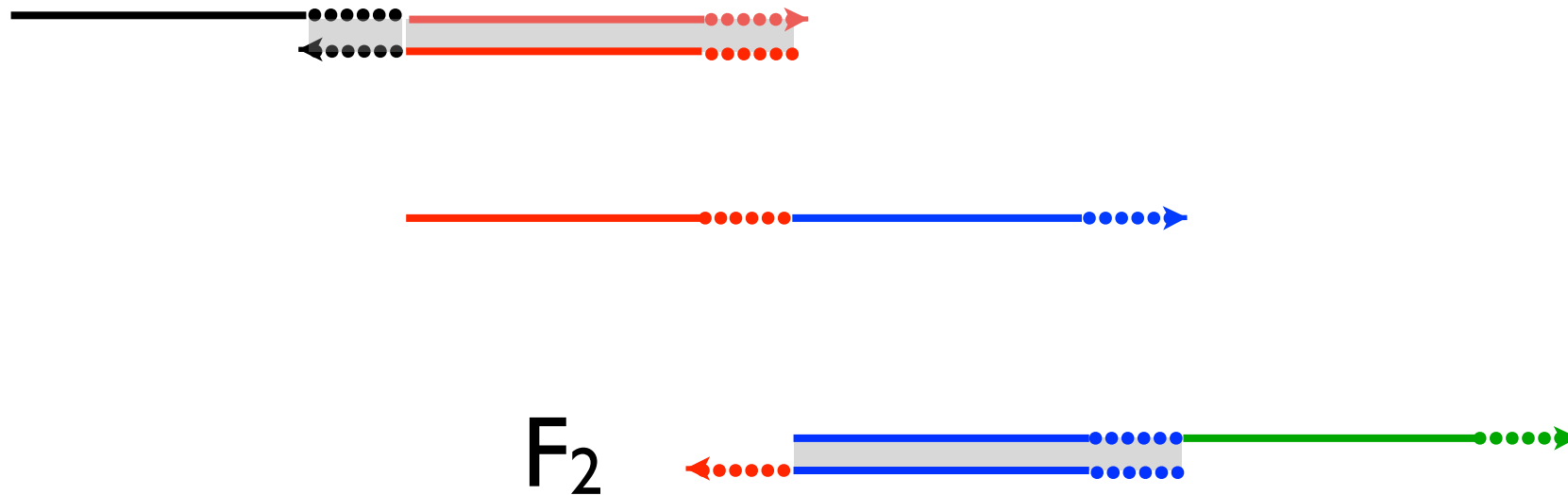
Translator (a “wire”): $X \rightarrow Y$

input X



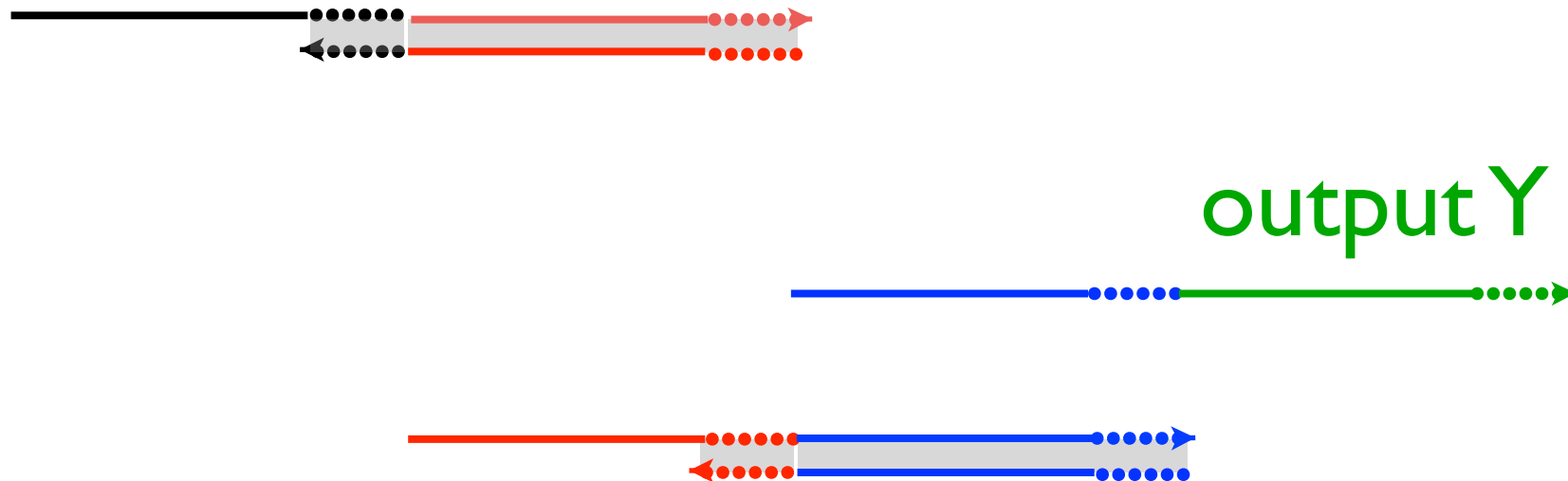
Sequence Independence

Translator (a “wire”): $X \rightarrow Y$

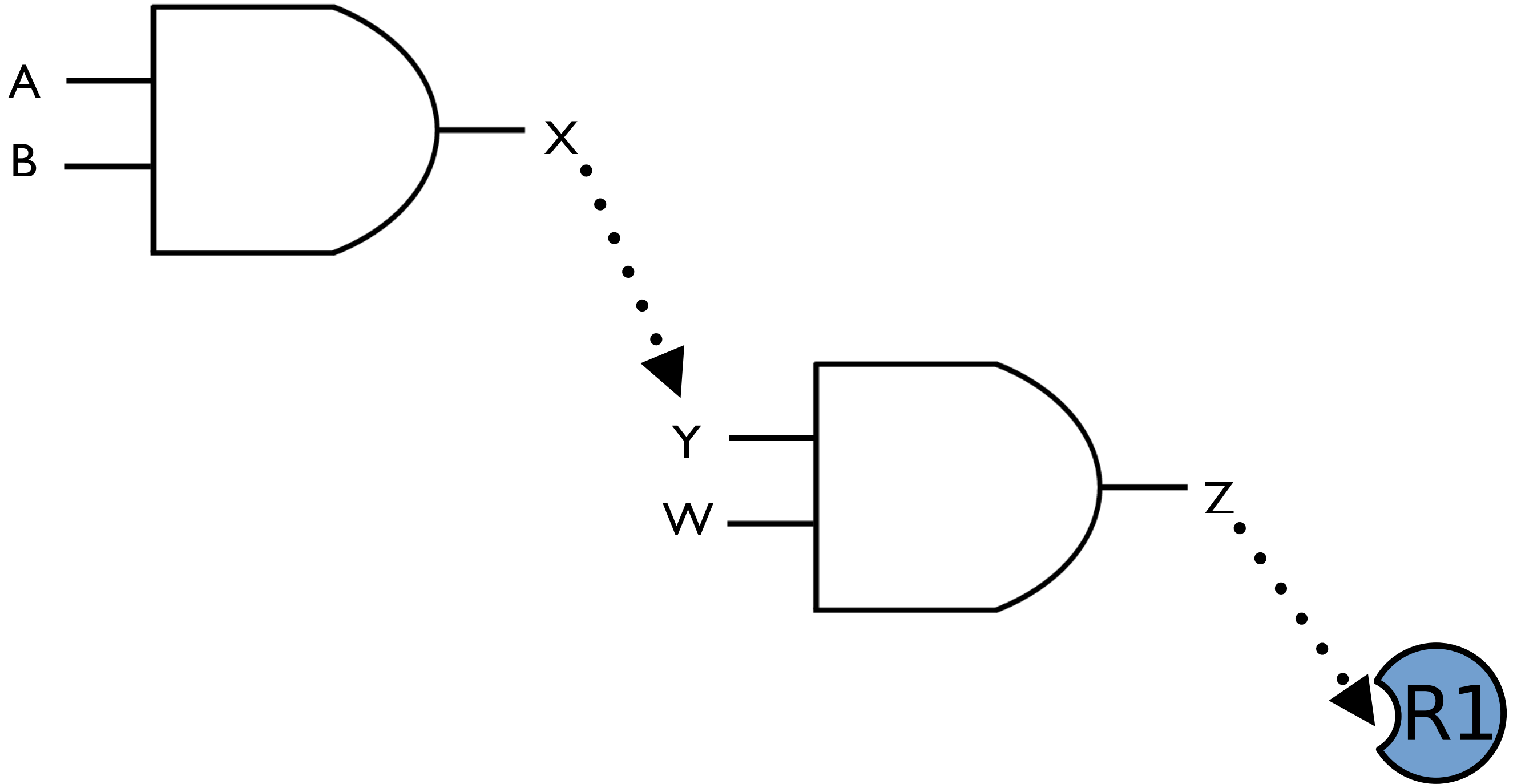


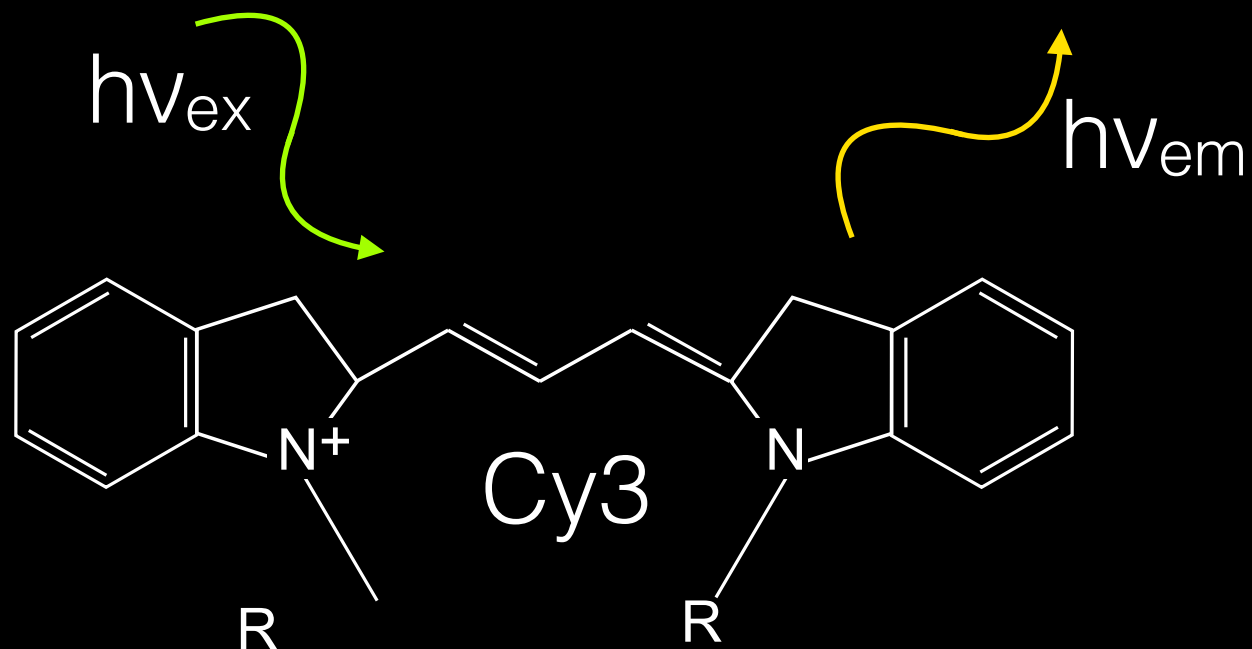
Sequence Independence

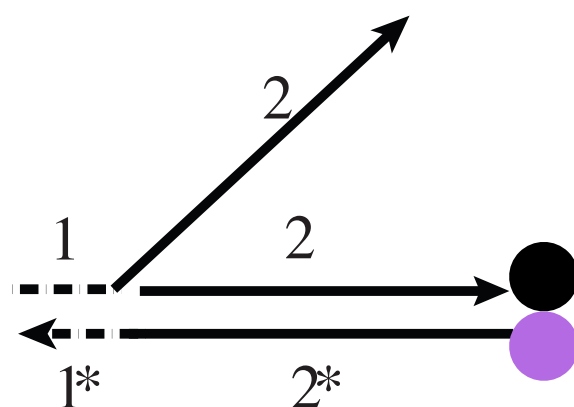
Translator (a “wire”): $X \rightarrow Y$



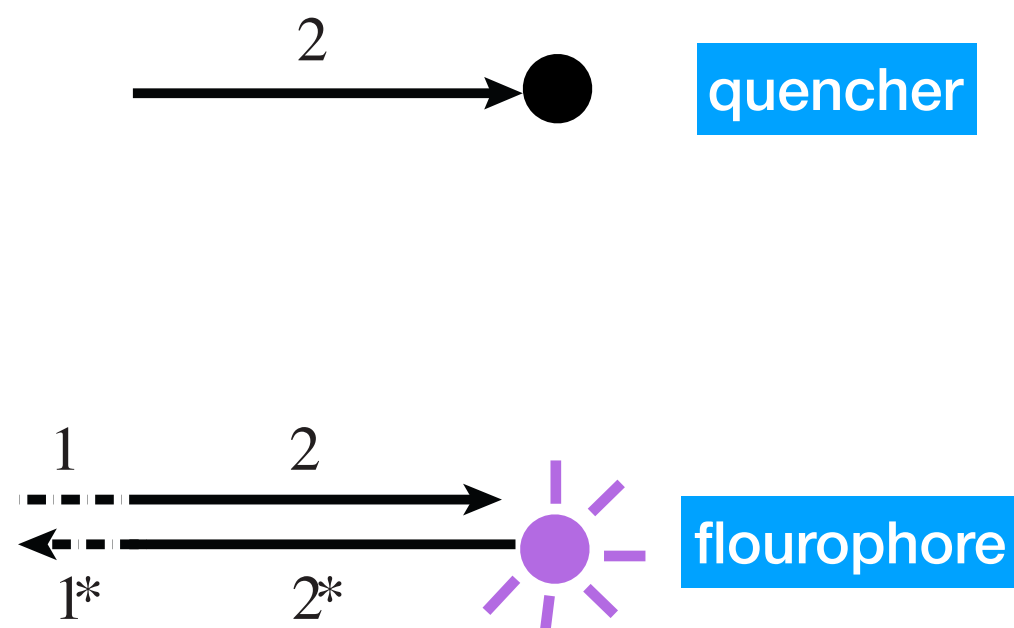
Reading Output



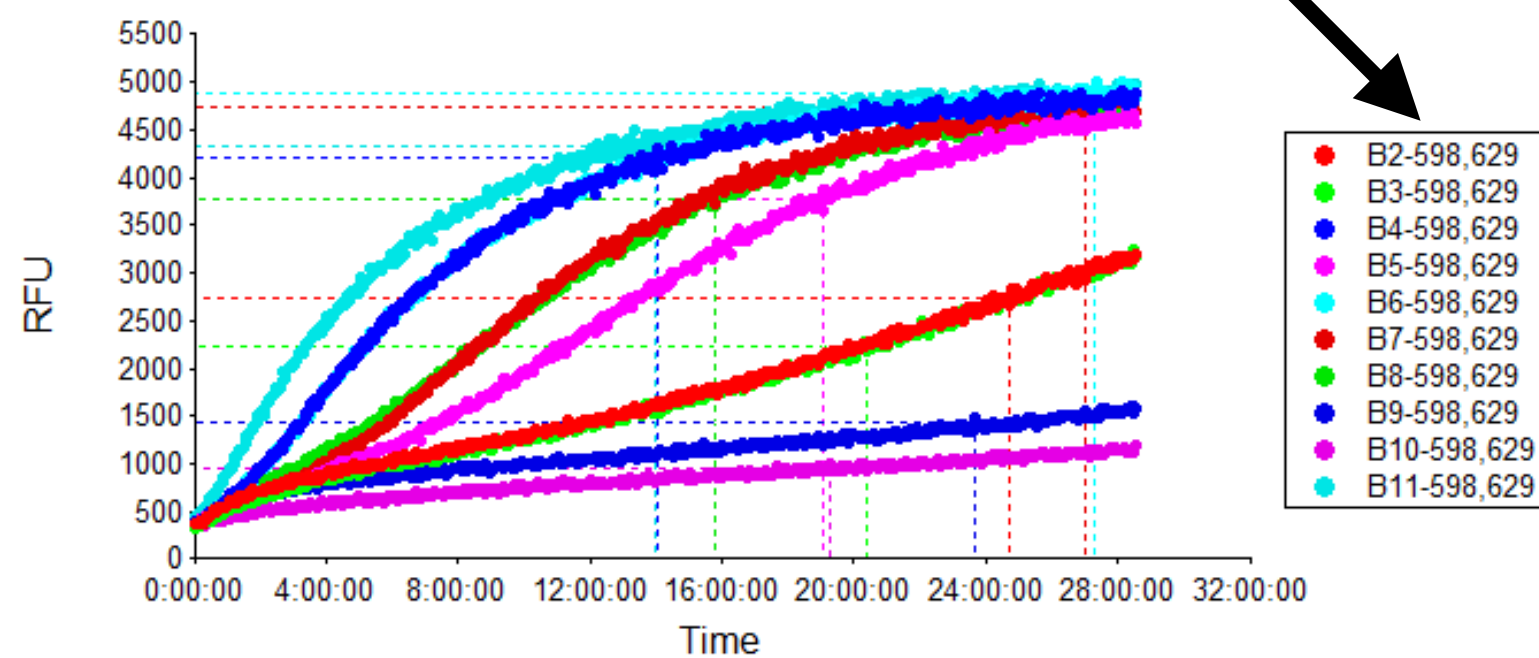




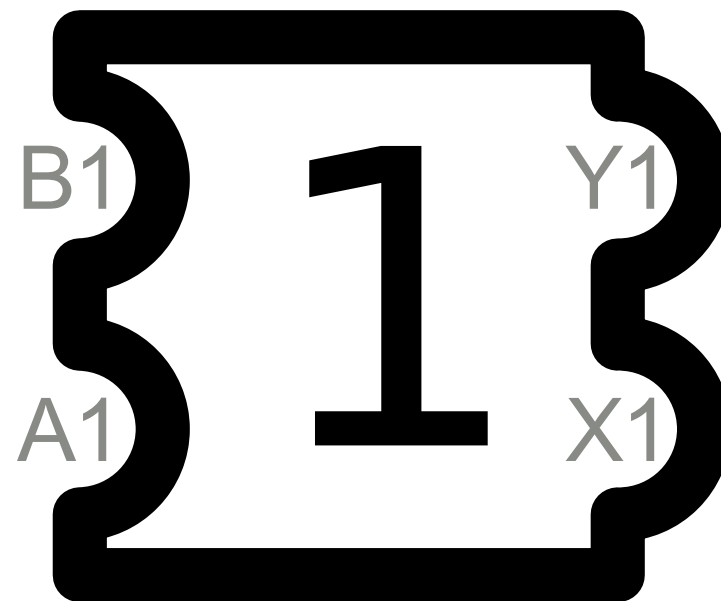
displace



read many different samples

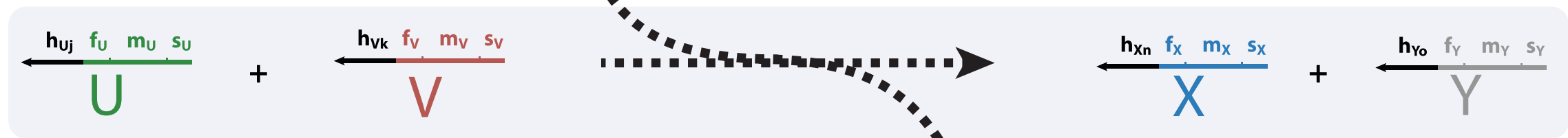
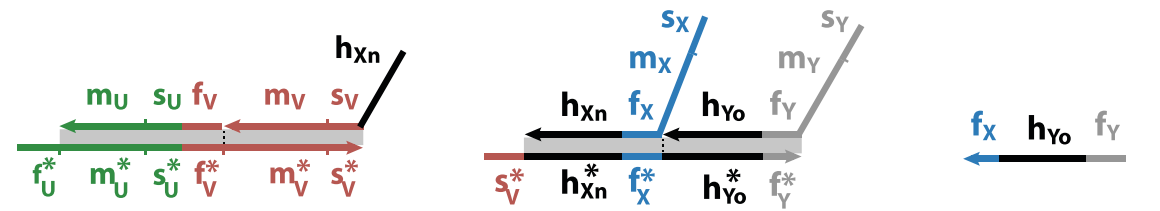


A reaction gate



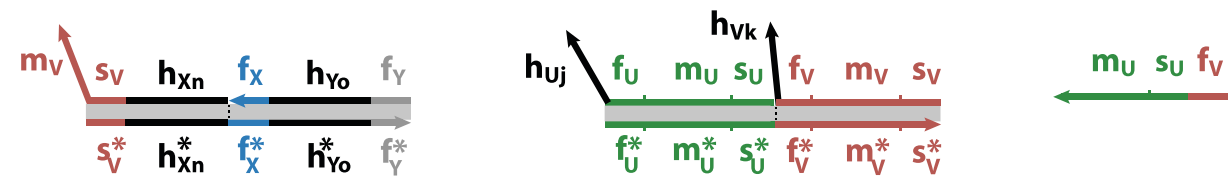
This *universal component*
can realize a number of logic gates

We start with large excess of DNA complexes (fuels) that mediate the reaction:

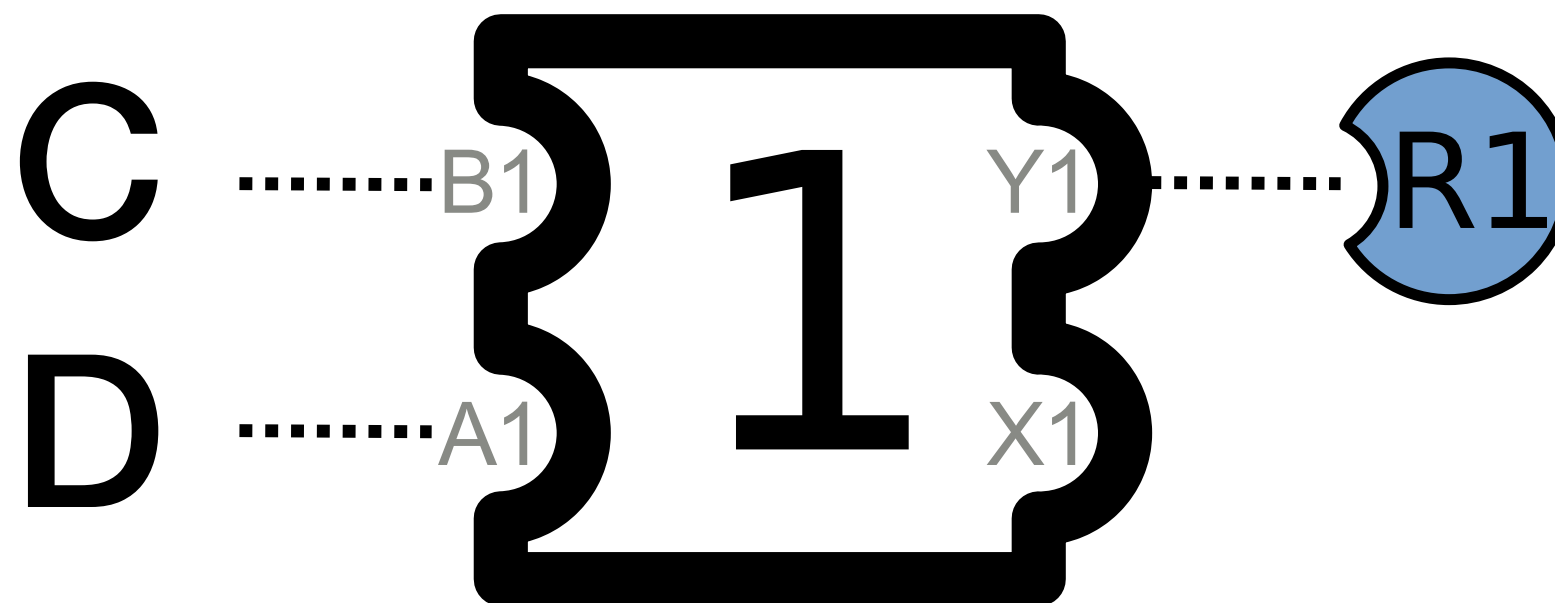


Waste

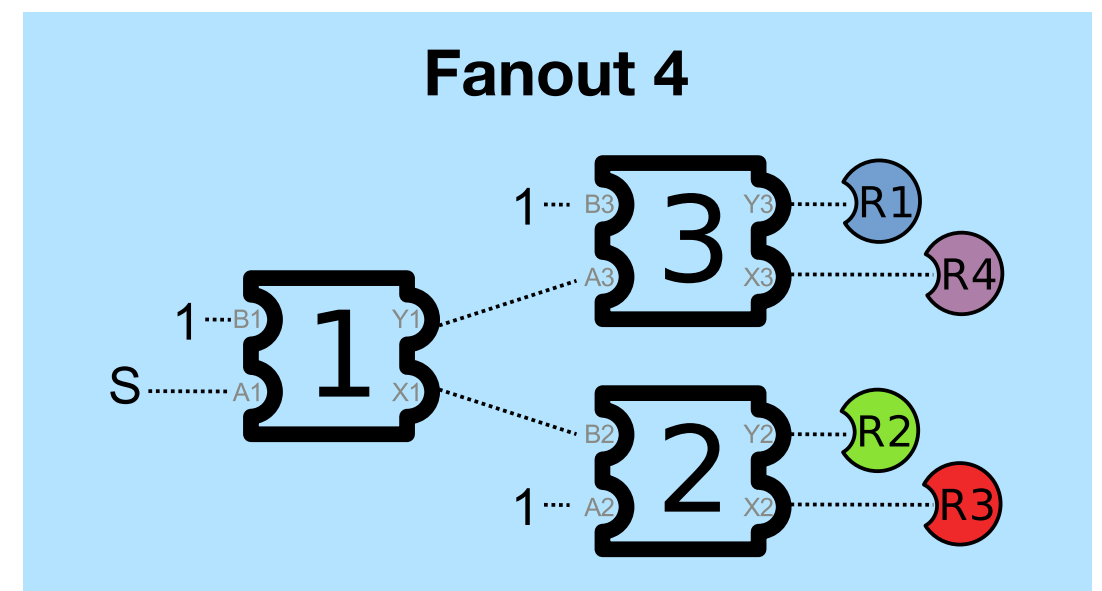
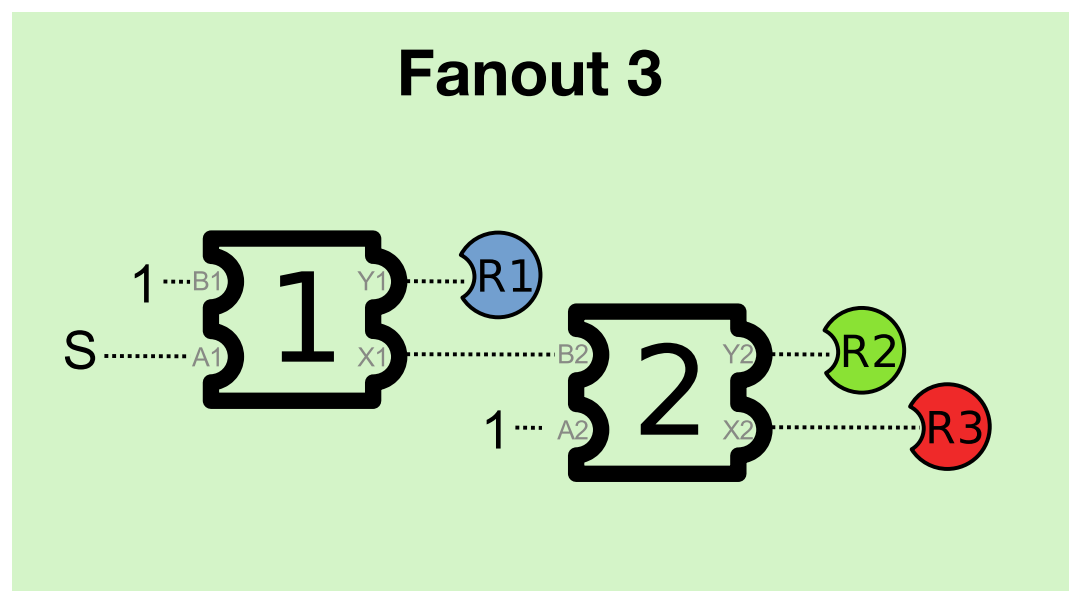
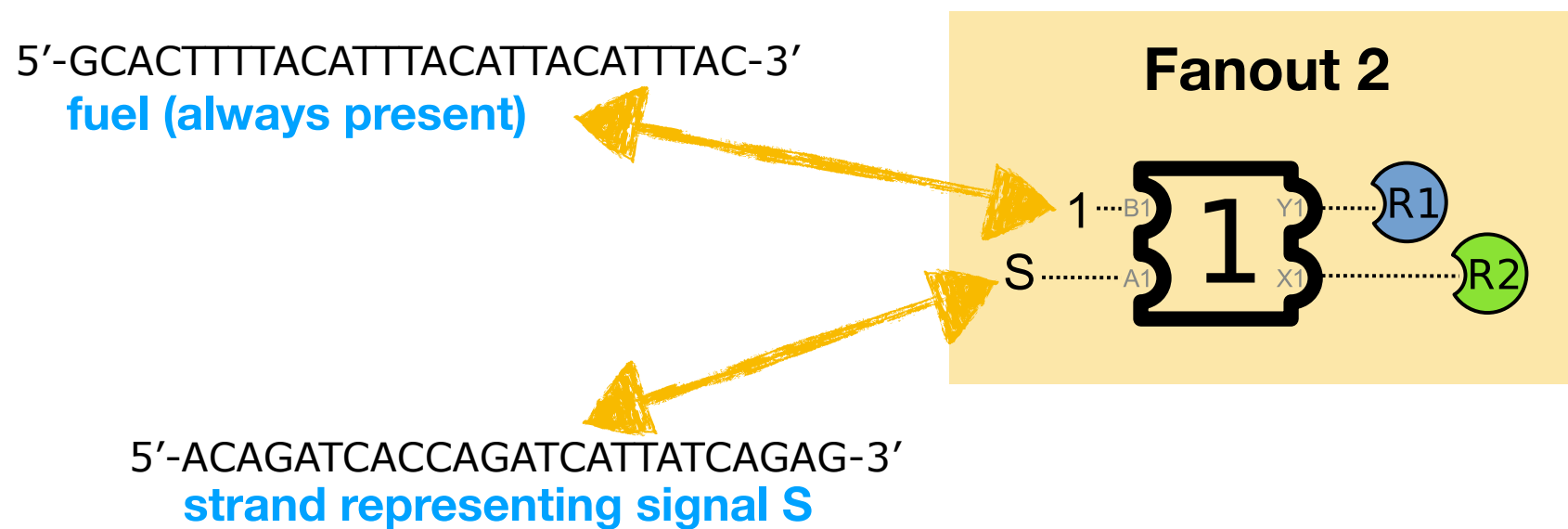
Waste



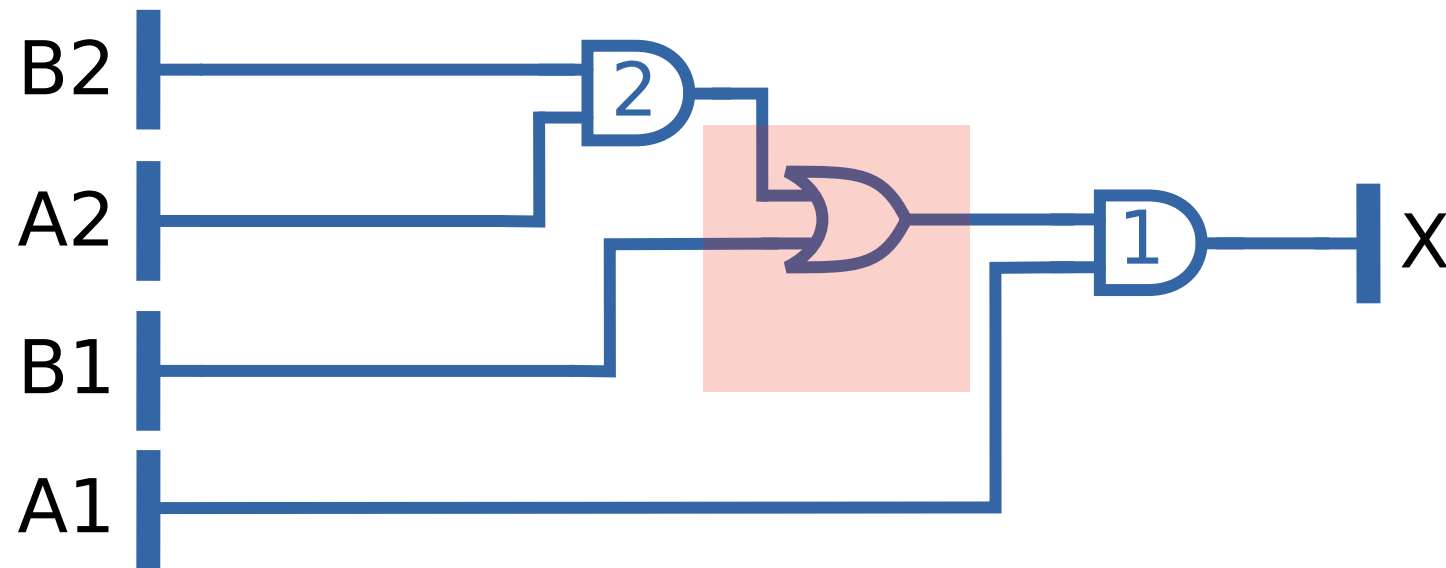
AND gate



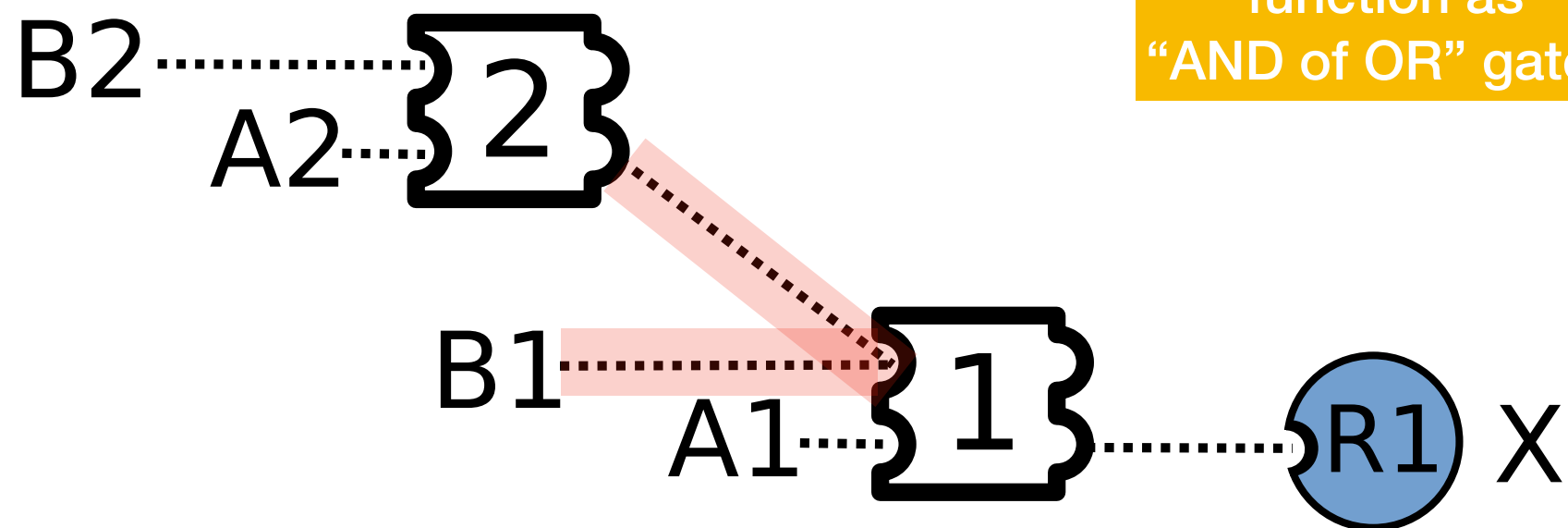
Signal Fanout Gates



Handling OR gates

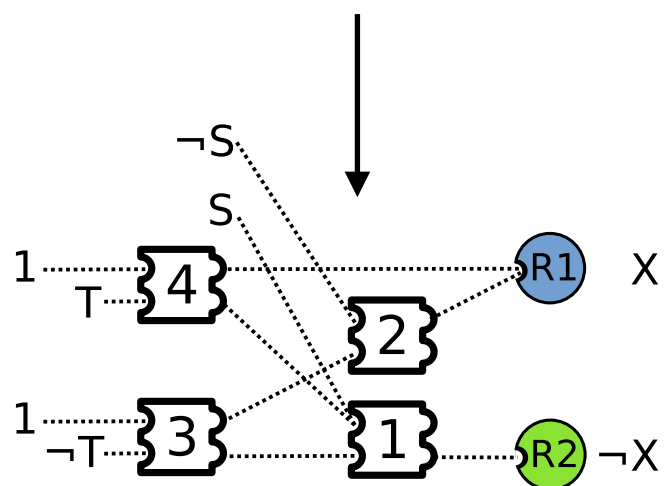
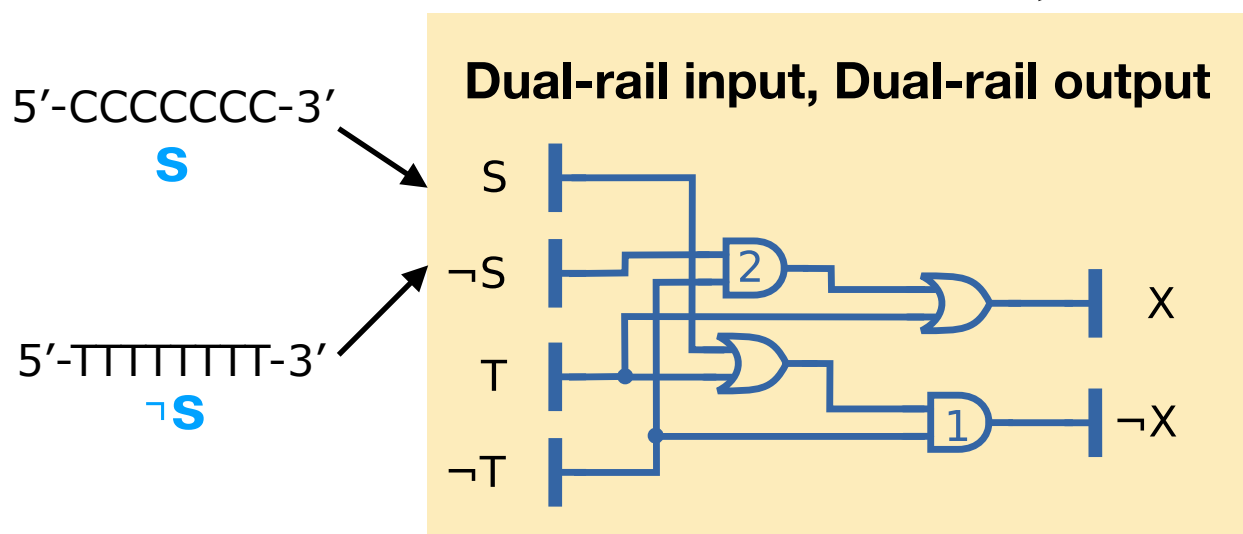
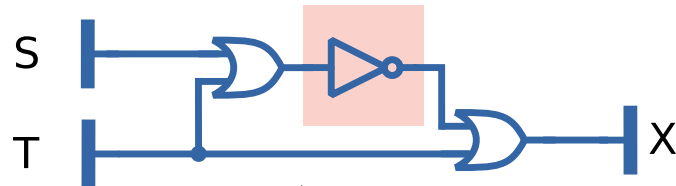


Reaction gates
function as
“AND of OR” gates



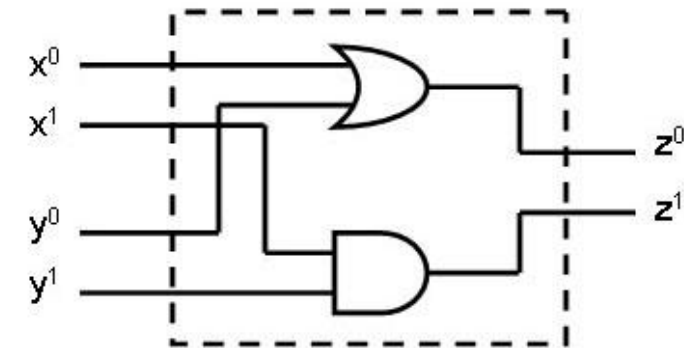
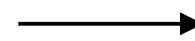
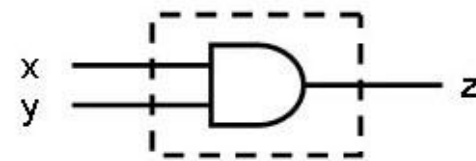
AND(A1, OR(B1, AND(A2, B2))

Handling NOT gates

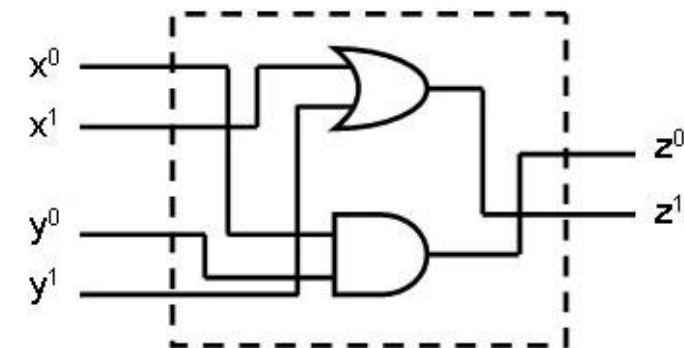
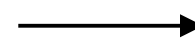
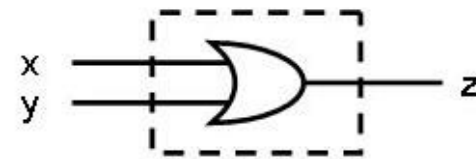


Dual rail logic

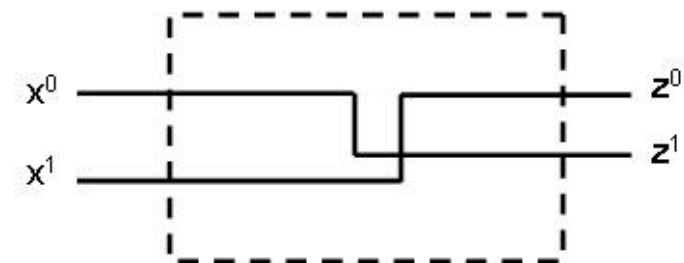
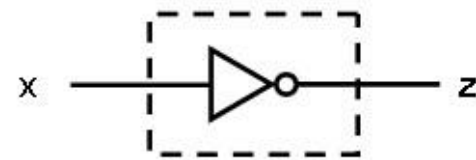
AND



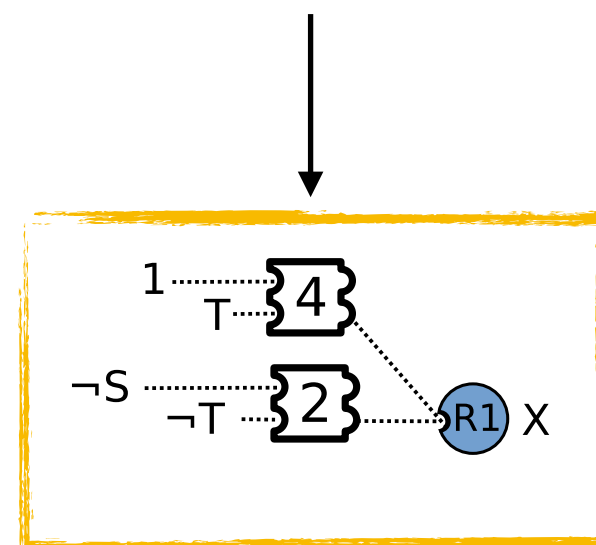
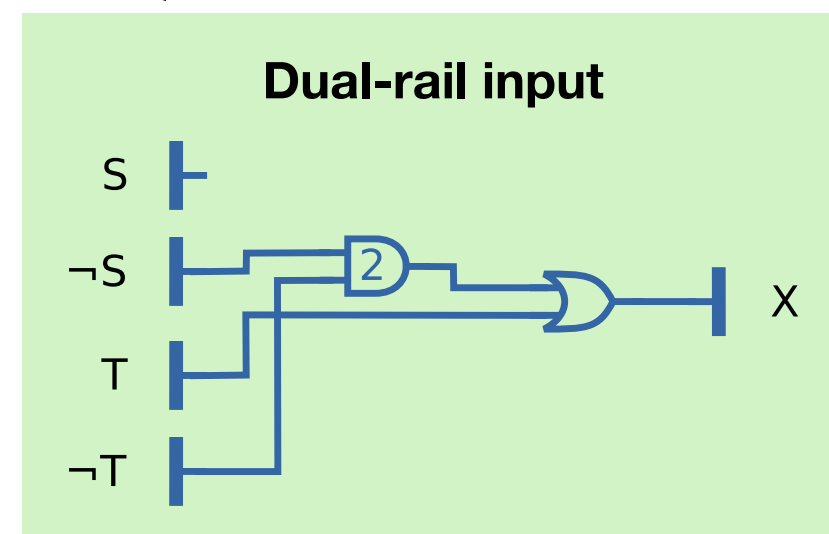
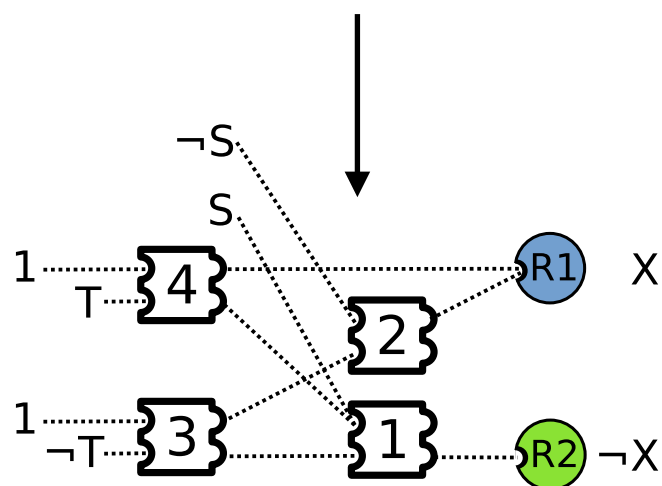
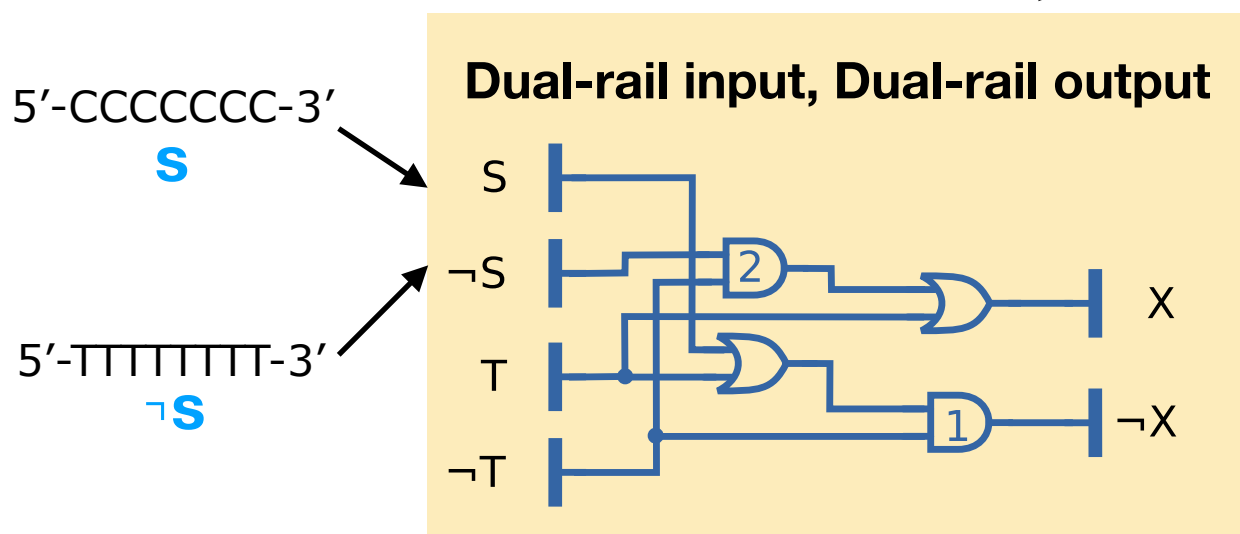
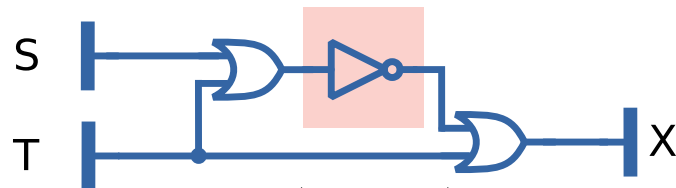
OR



NOT



Handling NOT gates



With reaction gates, wires,
and dual-rail encoding, we can
build any combinatorial circuit

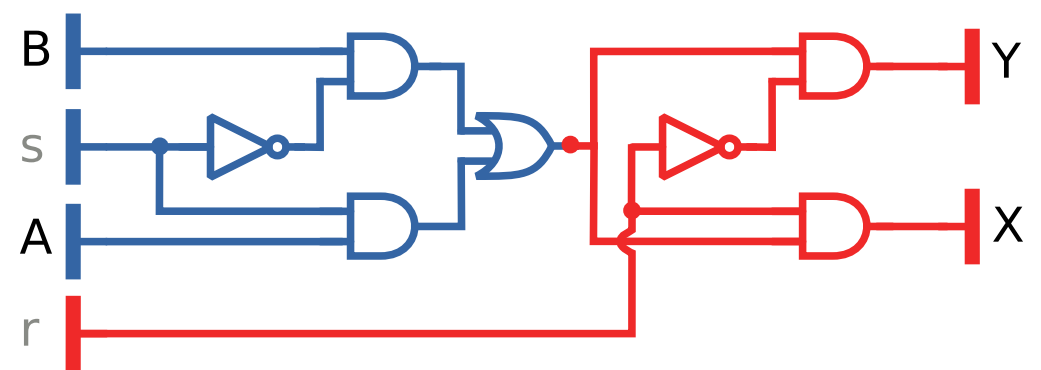
Tutorial Outline

- ▶ Review of strand displacement
- ▶ Building and composing logic gates
- ▶ **Tools for designing and verifying circuits**
- ▶ Robustness of strand displacement

How do you design the circuit?

2-input MUX				2-output DEMUX			
A	B	s	out	in	r	X	Y
0	0	0	0	0	0	0	0
0	1	0	1	1	0	0	1
1	0	0	0	0	1	0	0
1	1	0	1	1	1	1	0
0	0	1	0				
0	1	1	0				
1	0	1	1				
1	1	1	1				

Compile from Verilog, or truth table, into AND-OR-NOT circuit

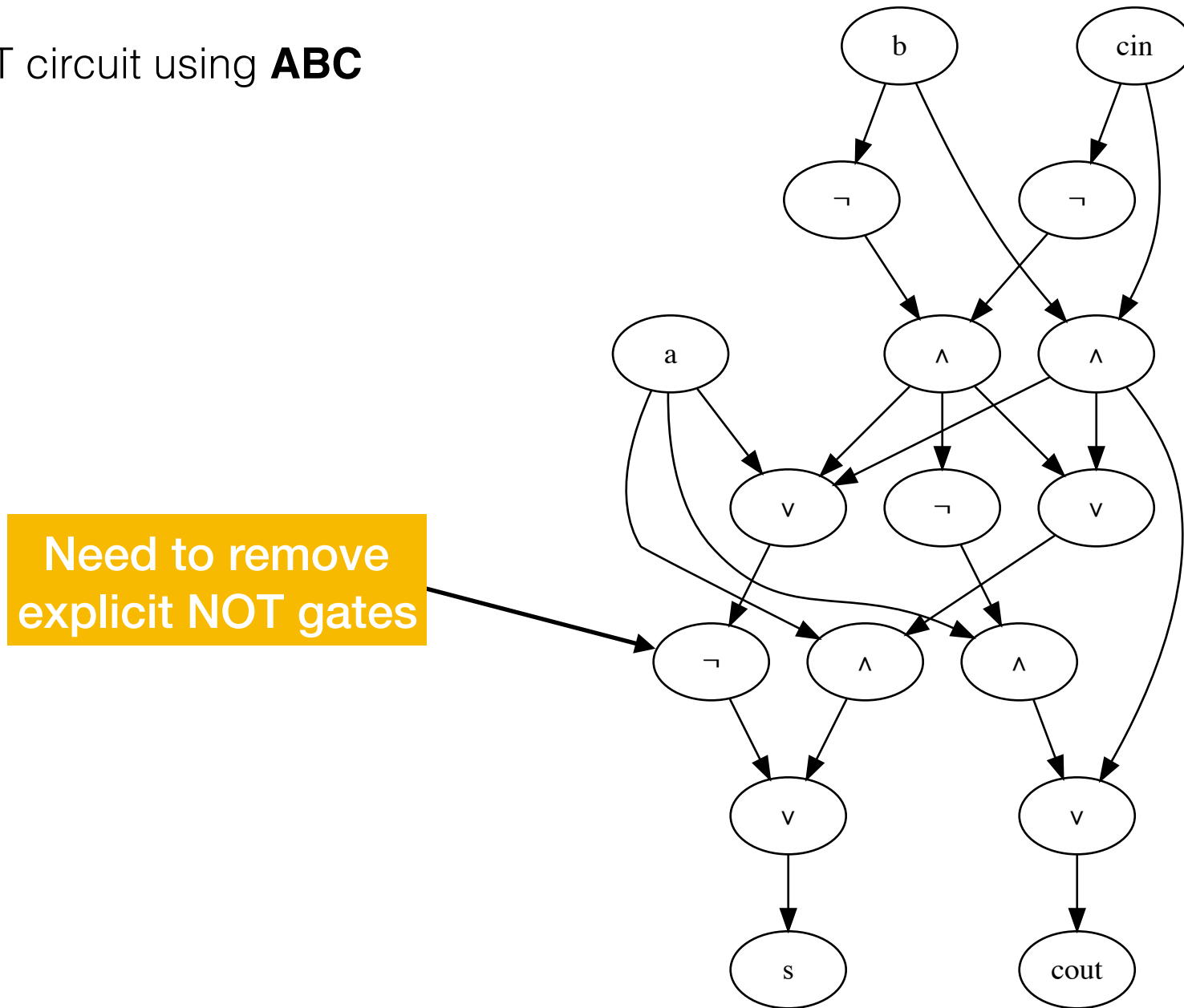


How do you design the circuit?

Inputs			Outputs	
<i>A</i>	<i>B</i>	<i>C_{in}</i>	<i>C_{out}</i>	<i>S</i>
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

How do you design the circuit?

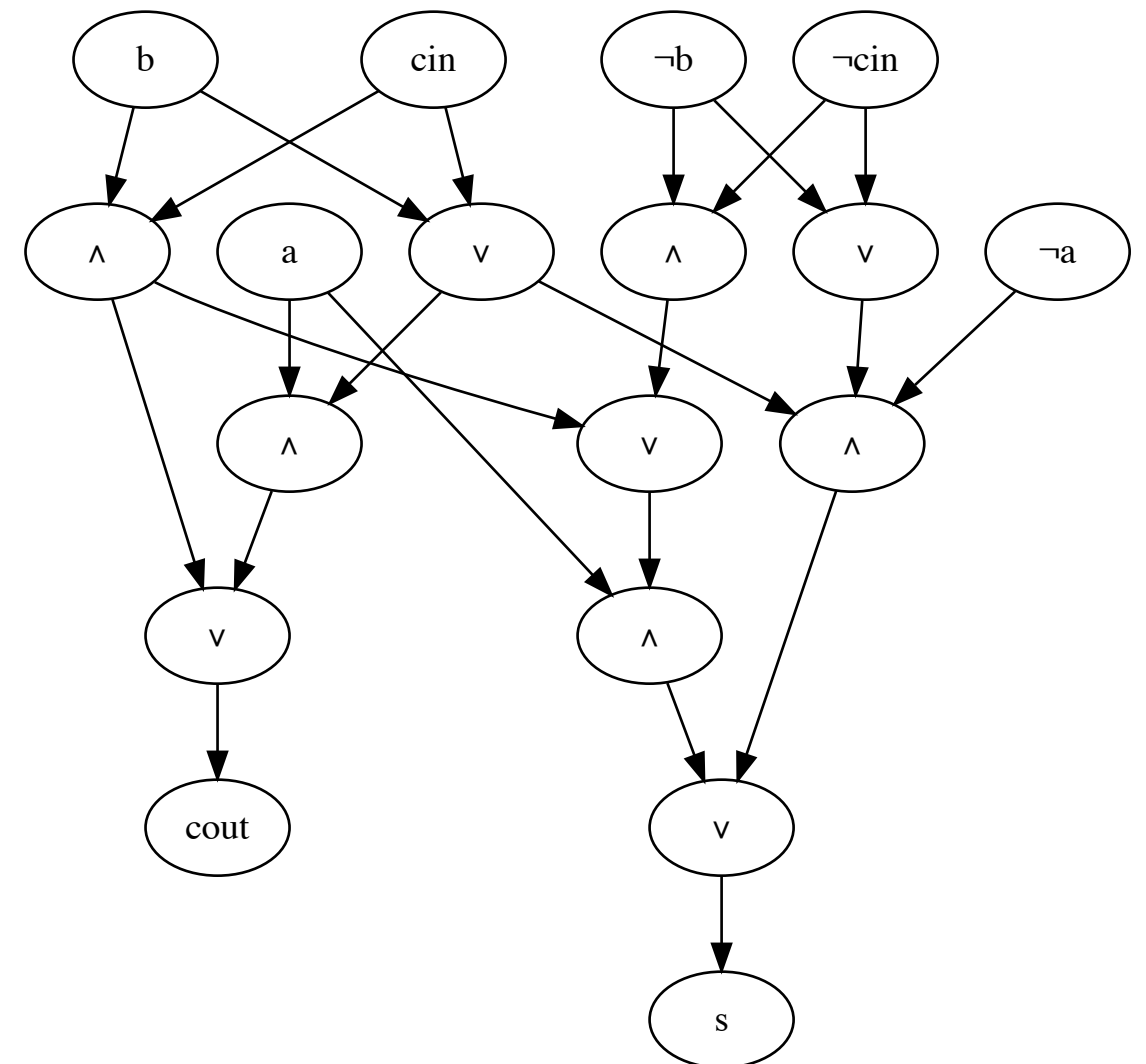
- ☒ Find minimized AND-OR-NOT circuit using **ABC**



How do you design the circuit?

- ✓ Find minimized AND-OR-NOT circuit using ABC
- ✓ “Tree-ify” circuit
- ✓ Push negations to literal level (dual-rail inputs)
- ✓ Compress circuit

Circuit now using
dual-rail input

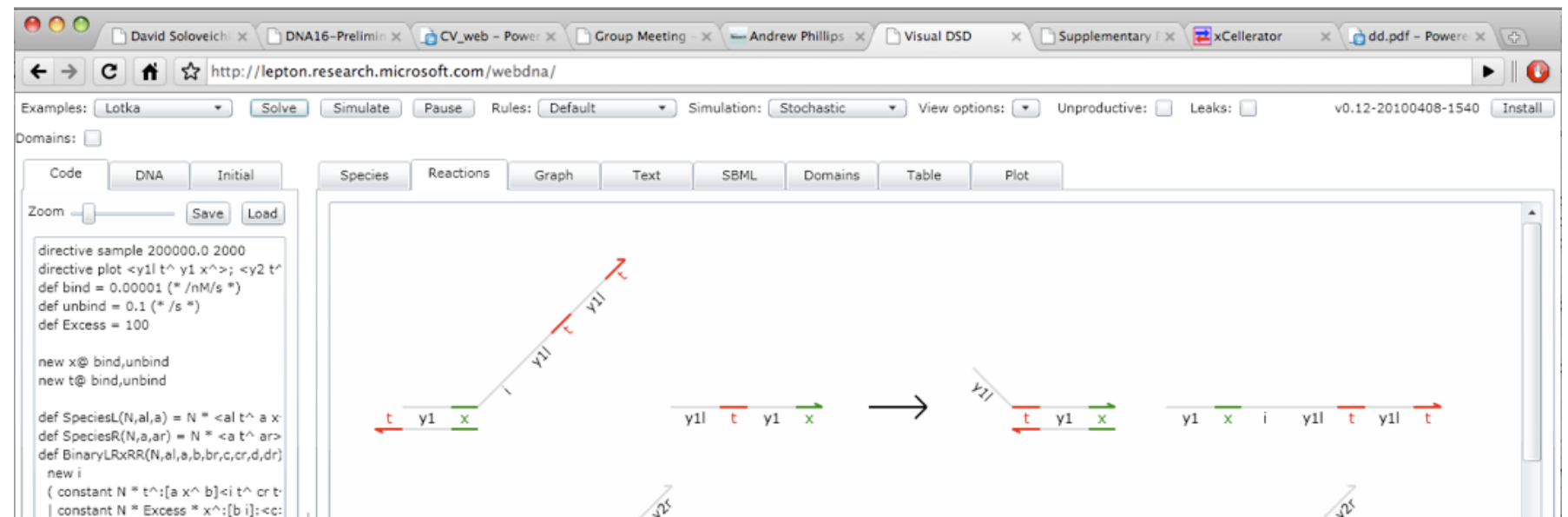
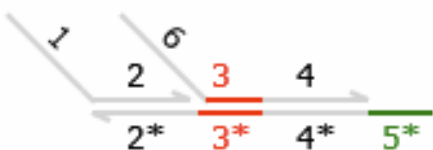


From circuit to DSD system

DSD: formal language for describing and modeling strand displacement cascades

<http://lepton.research.microsoft.com/webdna/>

$\langle 1 \rangle [2] : \langle 6 \rangle [3^{\wedge} 4] : 5^{\wedge *} =$



formal semantics

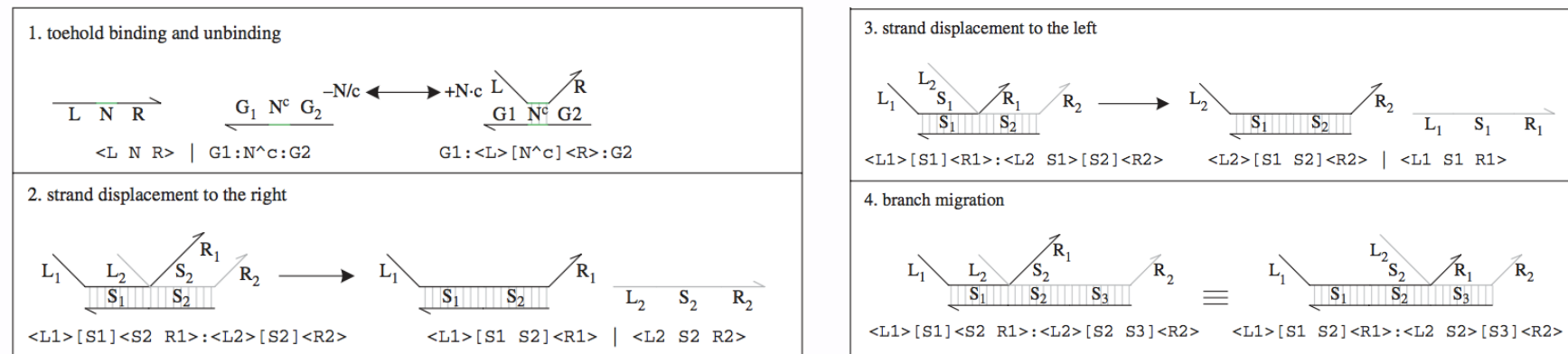
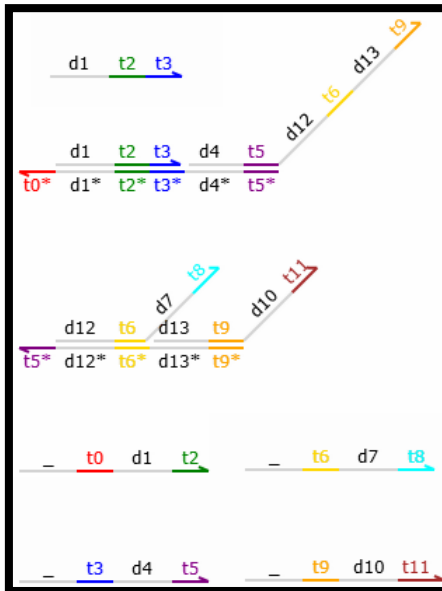


Figure 2. Reduction and branch migration rules of the strand displacement language. For each rule, the graphical representation at the top is equivalent to the program code at the bottom.

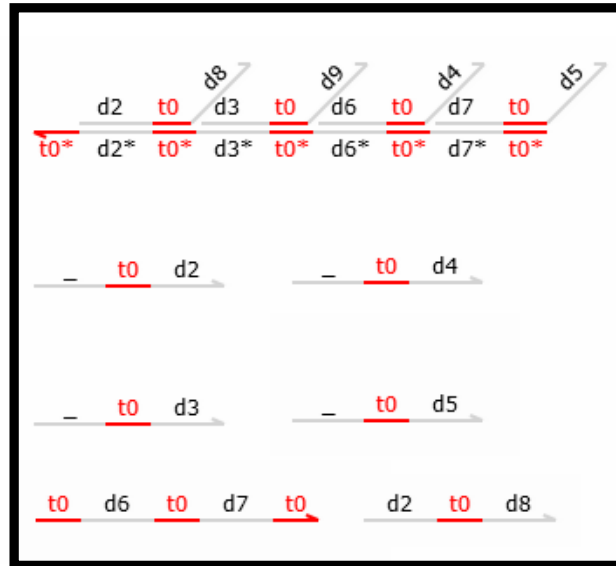
From circuit to DSD system

$$A + B \rightarrow C + D$$

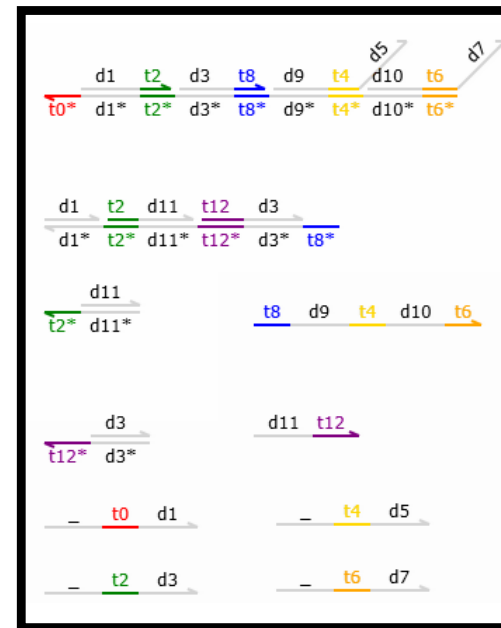
Soloveichik
et al. (2010)



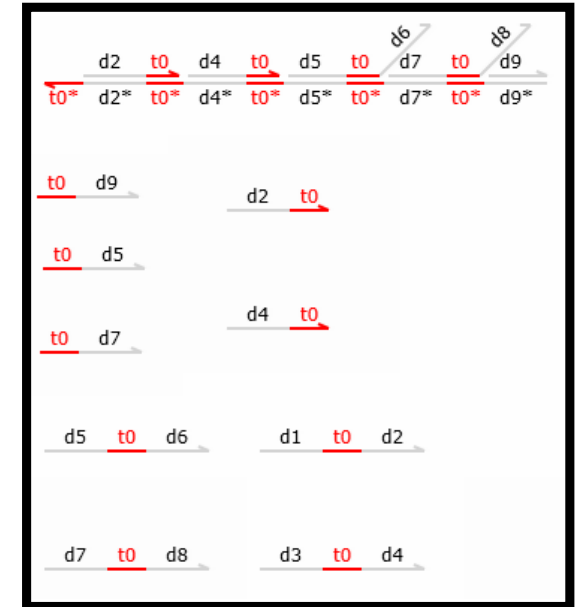
Lakin
et al. (2012)



Cardelli (2011)



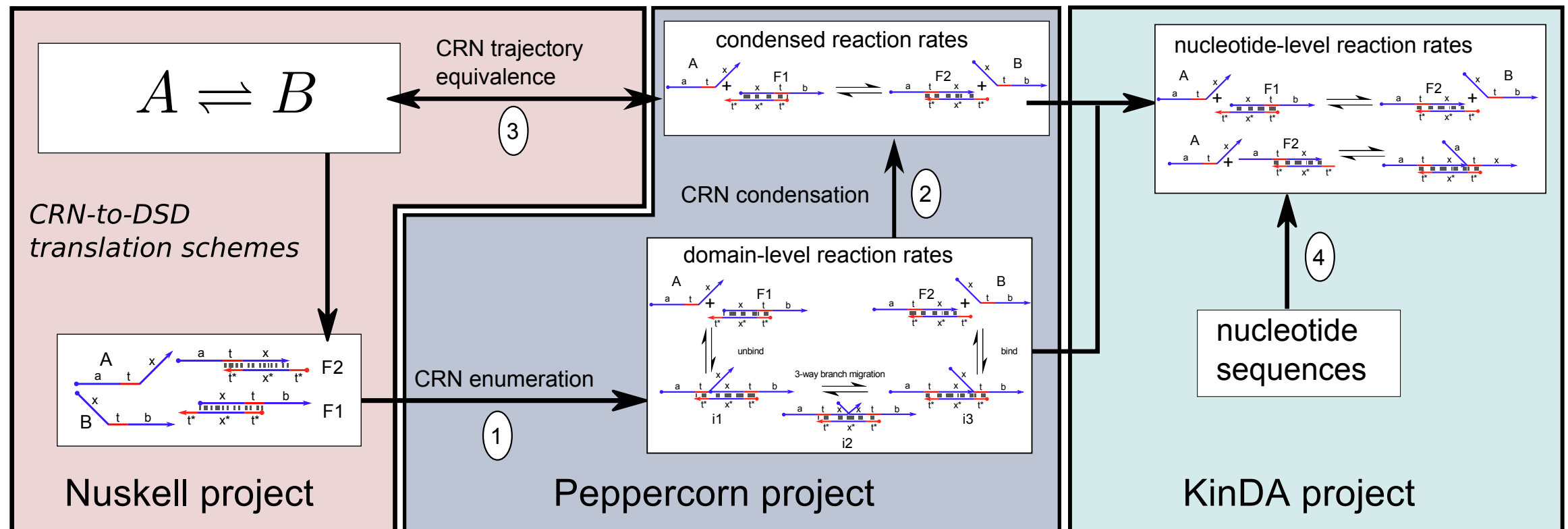
Qian et al. (2011)



Chen et al. (2012), Cardelli (2013), Srinivas (2015), Lakin et al. (2016), ...

Images drawn using VisualDSD, Lakin et al. (2012)

The Nuskell compiler framework



Badelt et al. (2017) - Nuskell

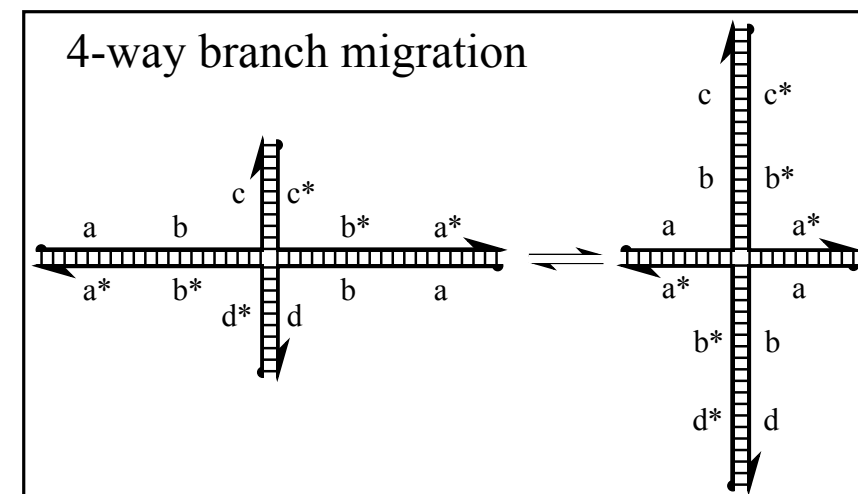
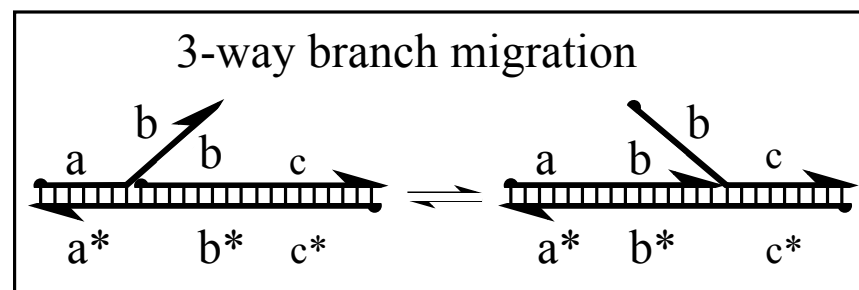
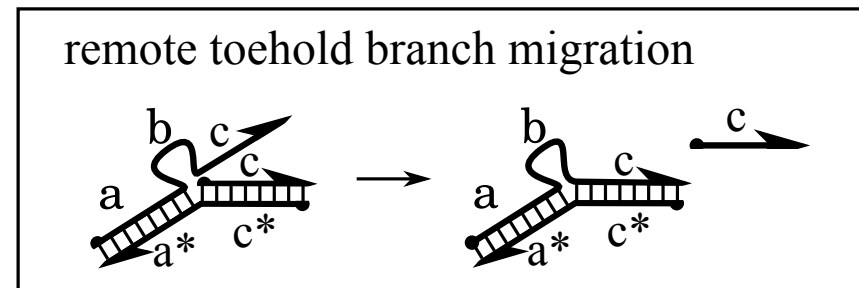
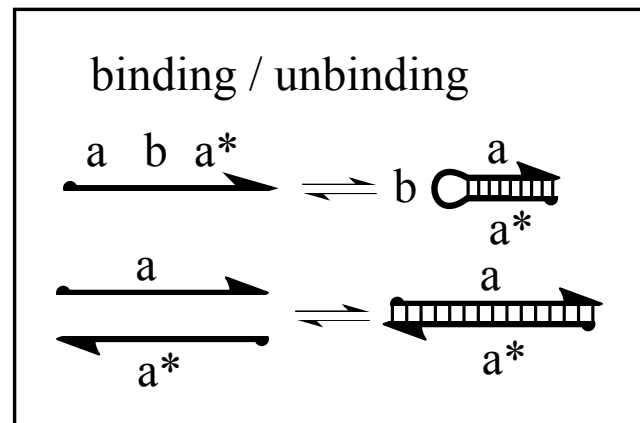
Grun et al. (2014) - Peppercorn

Shin et al. (2017) - CRN pathway decomposition equivalence

Johnson et al. (2018) - CRN bisimulation equivalence

Berleant et al. (submitted) - KinDA

Reaction Enumeration



CRN equivalence verification

formal input CRN

3 species

7 reactions

```
A      -> A + A
A + A  -> A
A + B  -> B + B
B      ->
A + C  ->
C      -> C + C
C + C  -> C
```

condensed CRN

42 species

32 reactions

```
f14 + C -> e1428 + f15
e853 + f12 -> C + f13
A + f4 -> f3 + e71
f2 + e25 -> A + f1
A + e25 -> f2 + e7
e996 + f3 -> A + f10
e1428 + f15 -> f14 + C
f3 + e71 -> A + f4
e465 + B -> e418 + f6
e614 + f9 -> e611 + e730
e996 + C -> e1040 + f12
e465 + f5 -> e514 + e368
e308 + f7 -> f8 + B
e418 + B -> e371 + f6
C + f13 -> e853 + f12
A + f1 -> f2 + e25
B + e71 -> e319 + f7
f2 + e7 -> A + e25
e1040 + f11 -> e1162 + e1163 + e1158
e319 + f7 -> B + e71
e308 + f9 -> e614 + e615 + e611
e371 + f6 -> e418 + B
e1040 + f12 -> e996 + C
f8 + B -> e308 + f7
e319 + f5 -> e372 + e371 + e368
f3 + e7 -> A + f0
e853 + f15 -> e1428 + C
e1428 + C -> e853 + f15
A + f10 -> e996 + f3
e1163 + f11 -> e1158 + e1246
e418 + f6 -> e465 + B
A + f0 -> f3 + e7
```

enumerated CRN

360 species

668 reactions

Are these networks equivalent?

translation scheme: qian2011_3D_var1.ts

CRN equivalence verification

formal input CRN	interpreted CRN	condensed CRN	enumerated CRN
3 species	3 species	42 species	360 species
7 reactions	7 non-trivial reactions	32 reactions	668 reactions
$A \rightarrow A + A$ $A + A \rightarrow A$ $A + B \rightarrow B + B$ $B \rightarrow$ $A + C \rightarrow$ $C \rightarrow C + C$ $C + C \rightarrow C$	$C \rightarrow C$ $C + C \rightarrow C$ $A \rightarrow A$ $A \rightarrow A$ $A + A \rightarrow A + A$ $A \rightarrow A$ $C \rightarrow C$ $A \rightarrow A$ $B \rightarrow B$ $\rightarrow$ $A + C \rightarrow A + C$ $\rightarrow$ $B \rightarrow B$ $B + B \rightarrow B + B$ $C \rightarrow C + C$ $A \rightarrow A$ $B + A \rightarrow A + B$ $A + A \rightarrow A + A$ $A + C \rightarrow$ $A + B \rightarrow B + A$ $B \rightarrow$ $B + B \rightarrow B + B$ $A + C \rightarrow A + C$ $B \rightarrow B$ $A + B \rightarrow B + B$ $A + A \rightarrow A$ $C + C \rightarrow C + C$ $C + C \rightarrow C + C$ $A \rightarrow A$ $\rightarrow$ $B \rightarrow B$ $A \rightarrow A + A$	$f14 + C \rightarrow e1428 + f15$ $e853 + f12 \rightarrow C + f13$ $A + f4 \rightarrow f3 + e71$ $f2 + e25 \rightarrow A + f1$ $A + e25 \rightarrow f2 + e7$ $e996 + f3 \rightarrow A + f10$ $e1428 + f15 \rightarrow f14 + C$ $f3 + e71 \rightarrow A + f4$ $e465 + B \rightarrow e418 + f6$ $e614 + f9 \rightarrow e611 + e730$ $e996 + C \rightarrow e1040 + f12$ $e465 + f5 \rightarrow e514 + e368$ $e308 + f7 \rightarrow f8 + B$ $e418 + B \rightarrow e371 + f6$ $C + f13 \rightarrow e853 + f12$ $A + f1 \rightarrow f2 + e25$ $B + e71 \rightarrow e319 + f7$ $f2 + e7 \rightarrow A + e25$ $e1040 + f11 \rightarrow e1162 + e1163 + e1158$ $e319 + f7 \rightarrow B + e71$ $e308 + f9 \rightarrow e614 + e615 + e611$ $e371 + f6 \rightarrow e418 + B$ $e1040 + f12 \rightarrow e996 + C$ $f8 + B \rightarrow e308 + f7$ $e319 + f5 \rightarrow e372 + e371 + e368$ $f3 + e7 \rightarrow A + f0$ $e853 + f15 \rightarrow e1428 + C$ $e1428 + C \rightarrow e853 + f15$ $A + f10 \rightarrow e996 + f3$ $e1163 + f11 \rightarrow e1158 + e1246$ $e418 + f6 \rightarrow e465 + B$ $A + f0 \rightarrow f3 + e7$	

Johnson et al. (2016) - CRN bisimulation equivalence
translation scheme: qian2011_3D_var1.ts

CRN equivalence verification

formal input CRN

3 species

7 reactions

```
A      -> A + A
A + A -> A
A + B -> B + B
B      ->
A + C ->
C      -> C + C
C + C -> C
```

interpreted CRN

3 species

7 non-trivial reactions

```
C      -> C
C + C -> C
A      -> A
A      -> A
A + A -> A + A
A      -> A
C      -> C
A      -> A
B      -> B
      ->
A + C -> A + C
      ->
B      -> B
B + B -> B + B
C      -> C + C
A      -> A
B + A -> A + B
A + A -> A + A
A + C ->
A + B -> B + A
B      ->
B + B -> B + B
A + C -> A + C
B      -> B
A + B -> B + B
A + A -> A
C + C -> C + C
C + C -> C + C
A      -> A
      ->
B      -> B
A      -> A + A
```

condensed CRN

42 species

32 reactions

```
f14 + C -> e1428 + f15
e853 + f12 -> C + f13
A + f4 -> f3 + e71
f2 + e25 -> A + f1
A + e25 -> f2 + e7
e996 + f3 -> A + f10
e1428 + f15 -> f14 + C
f3 + e71 -> A + f4
e465 + B -> e418 + f6
e614 + f9 -> e611 + e730
e996 + C -> e1040 + f12
e465 + f5 -> e514 + e368
e308 + f7 -> f8 + B
e418 + B -> e371 + f6
C + f13 -> e853 + f12
A + f1 -> f2 + e25
B + e71 -> e319 + f7
f2 + e7 -> A + e25
e1040 + f11 -> e1162 + e1163 + e1158
e319 + f7 -> B + e71
e308 + f9 -> e614 + e615 + e611
e371 + f6 -> e418 + B
e1040 + f12 -> e996 + C
f8 + B -> e308 + f7
e319 + f5 -> e372 + e371 + e368
f3 + e7 -> A + f0
e853 + f15 -> e1428 + C
e1428 + C -> e853 + f15
A + f10 -> e996 + f3
e1163 + f11 -> e1158 + e1246
e418 + f6 -> e465 + B
A + f0 -> f3 + e7
```

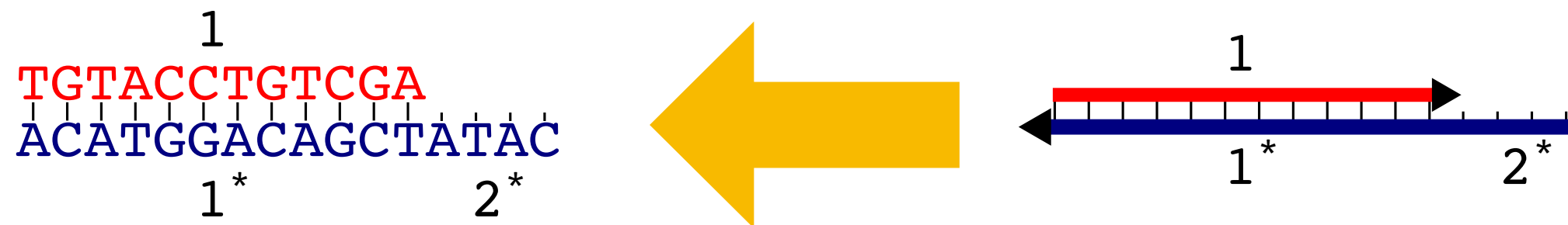
enumerated CRN

360 species

668 reactions

Johnson et al. (2016) - CRN bisimulation equivalence
translation scheme: qian2011_3D_var1.ts

Designing Sequences



Designing Sequences

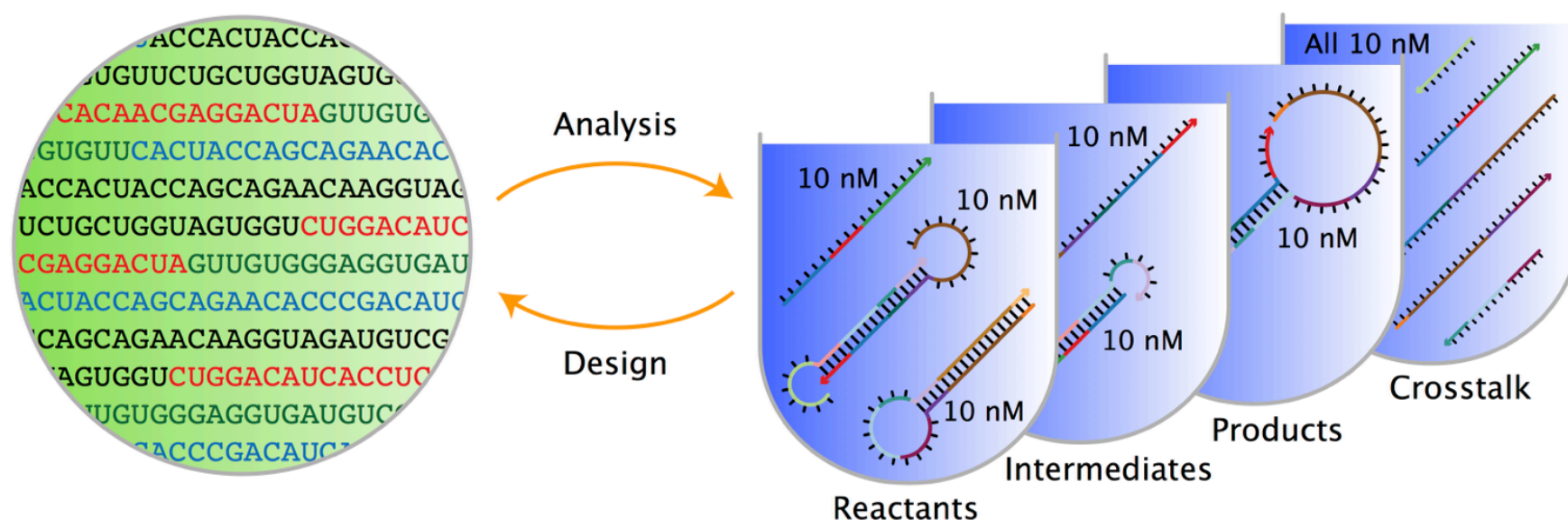


NUPACK

[Analysis](#)[Design](#)[Utilities](#)[Downloads](#)

NUPACK is a growing software suite for the analysis and design of nucleic acid structures, devices, and systems.

The NUPACK web application enables analysis and design of the equilibrium base-pairing properties of one or more test tubes of interacting nucleic acid strands:



Please [cite](#) the web application and algorithms appropriately; usage statistics are an important component in helping to secure funding for NUPACK development. We are happy to provide advice and [technical support](#).

— The NUPACK Team

News: Constrained multistate test tube design for reaction pathway engineering is now published! ([pdf](#), [supp info](#), [source code](#), [user guide](#))

Designing Sequences

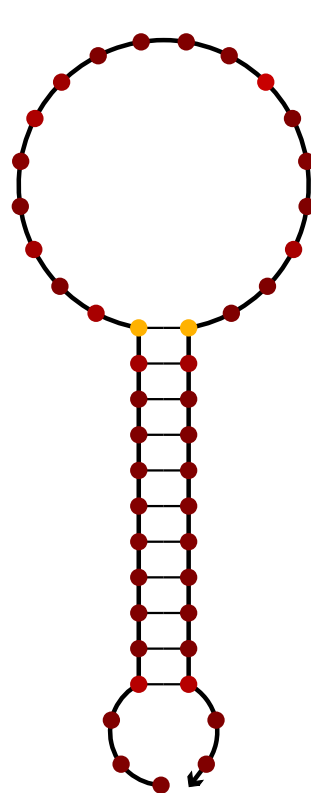


input X

poor sequence
for a signal strand

good sequence
for a signal strand

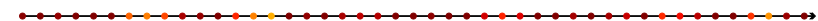
MFE structure at 25.0 C



Equilibrium probability
1.0
0.8
0.6
0.4
0.2
0.0

Free energy of secondary structure: -10.70 kcal/mol

MFE structure at 25.0 C



Equilibrium probability
1.0
0.8
0.6
0.4
0.2
0.0

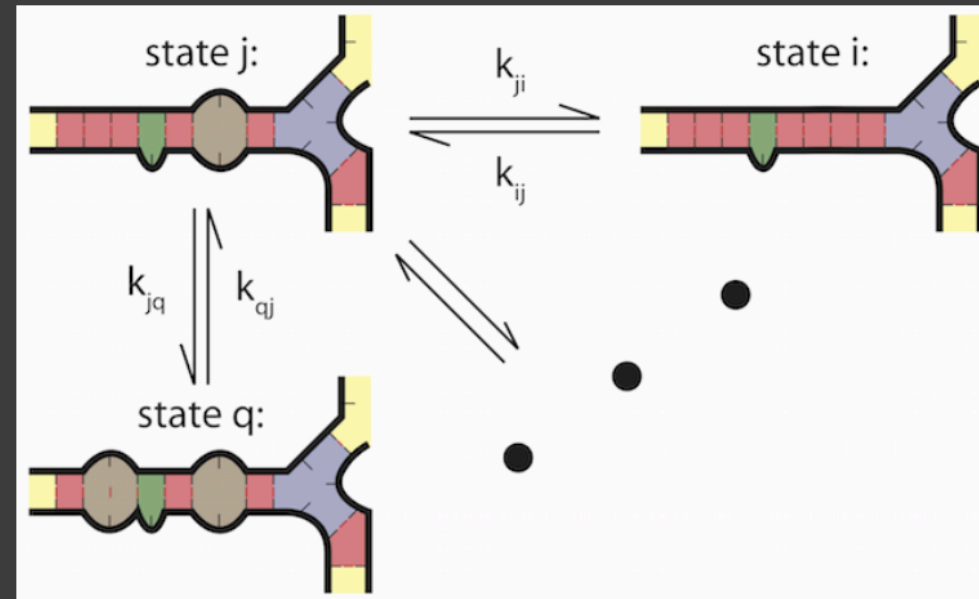
Free energy of secondary structure: 0.00 kcal/mol

Multistrand.org to determine reaction rates

Multistrand is a software package for simulating the kinetics of multiple interacting nucleic acid strands. It is developed at the Winfree lab at the California Institute of Technology.

► DNA and Natural Algorithms Group @ Caltech

[Live demo](#)



Key Features

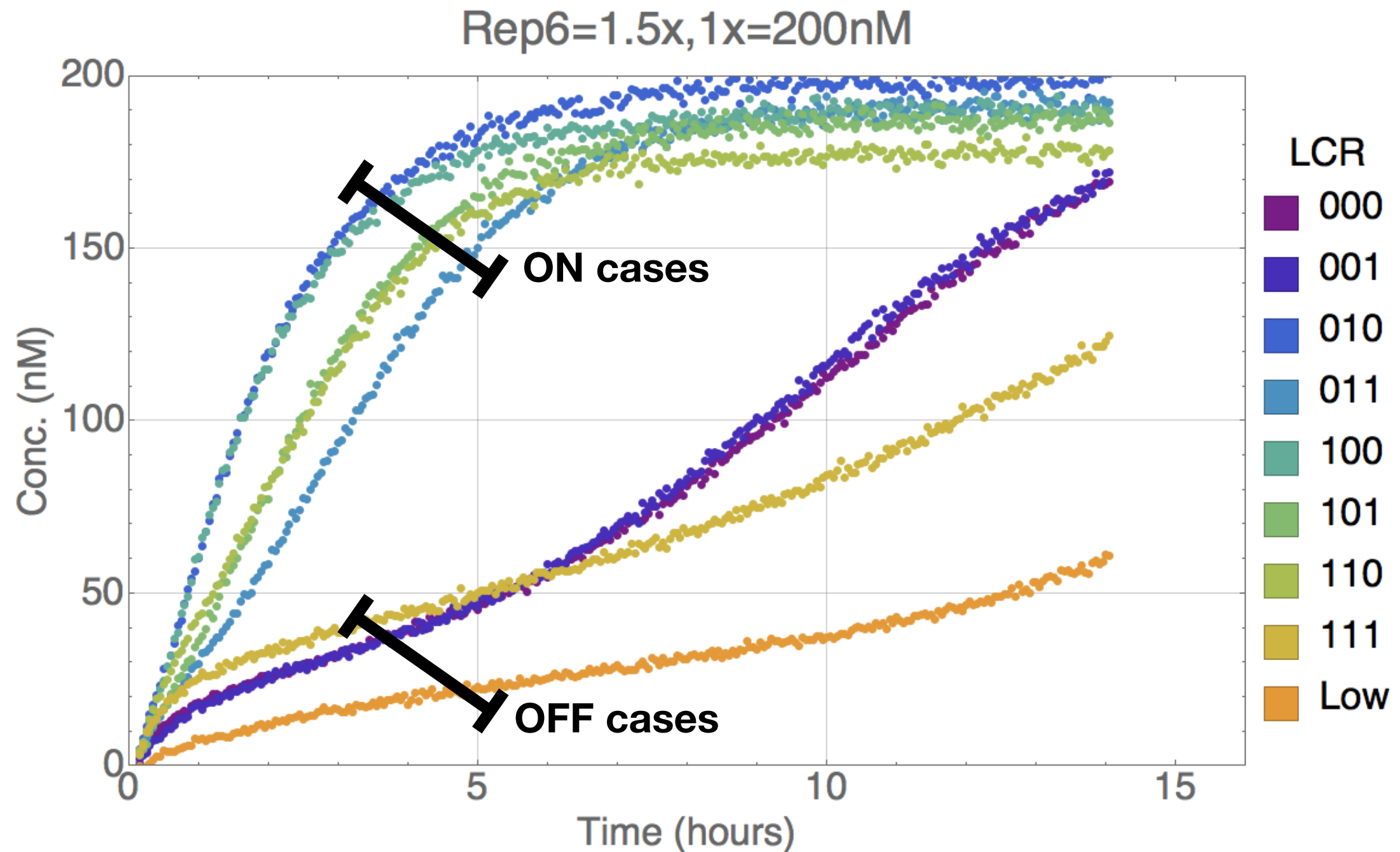
- Kinetic simulations of nucleic acids as random walk on thermodynamic energy model
- Supports multiple interacting strands
- Equilibrium consistent with [NUPACK](#)
- Various usage modes to study kinetic trajectories
- Distributed as a Python package
- MIT License



Tutorial Outline

- ▶ Review of strand displacement
- ▶ Building and composing logic gates
- ▶ Tools for designing and verifying circuits
- ▶ **Robustness of strand displacement**

Why is this circuit not *robust*?

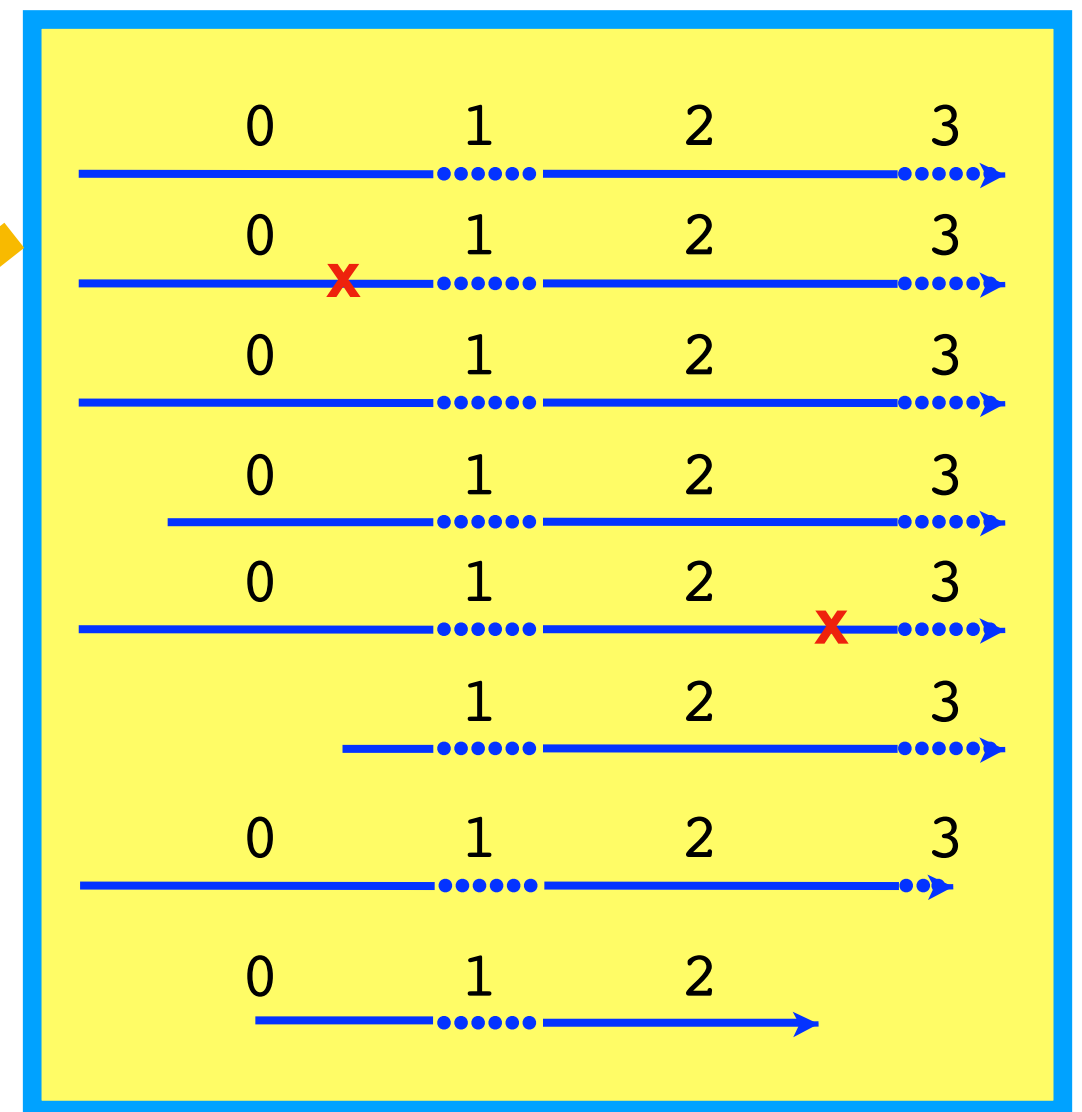


What causes signal *leak*?



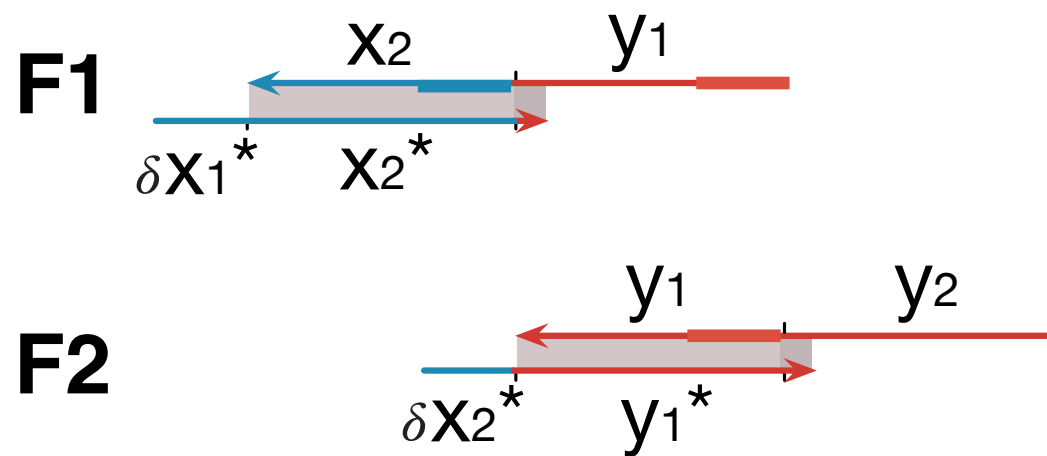
Problem 1: Molecules are not perfect

Imperfect strands from imperfect synthesis

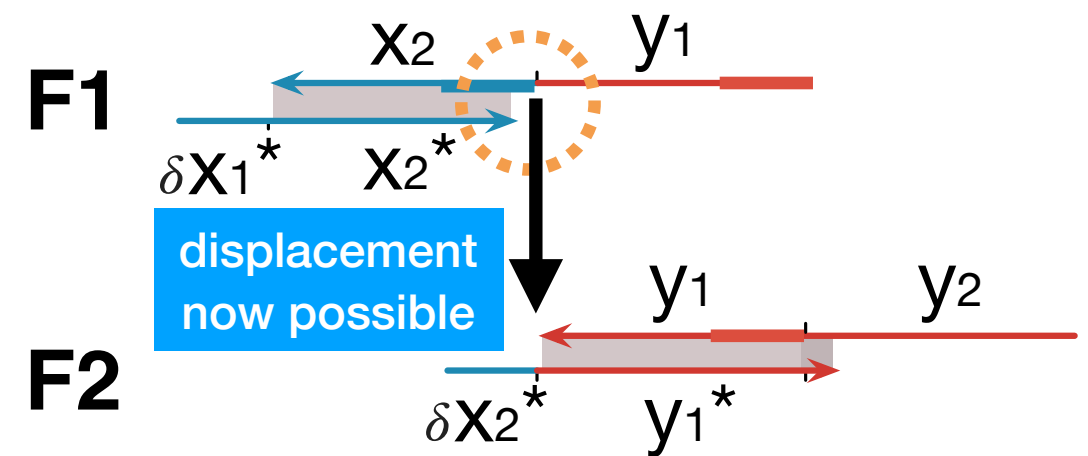


Problem 1: Molecules are not perfect

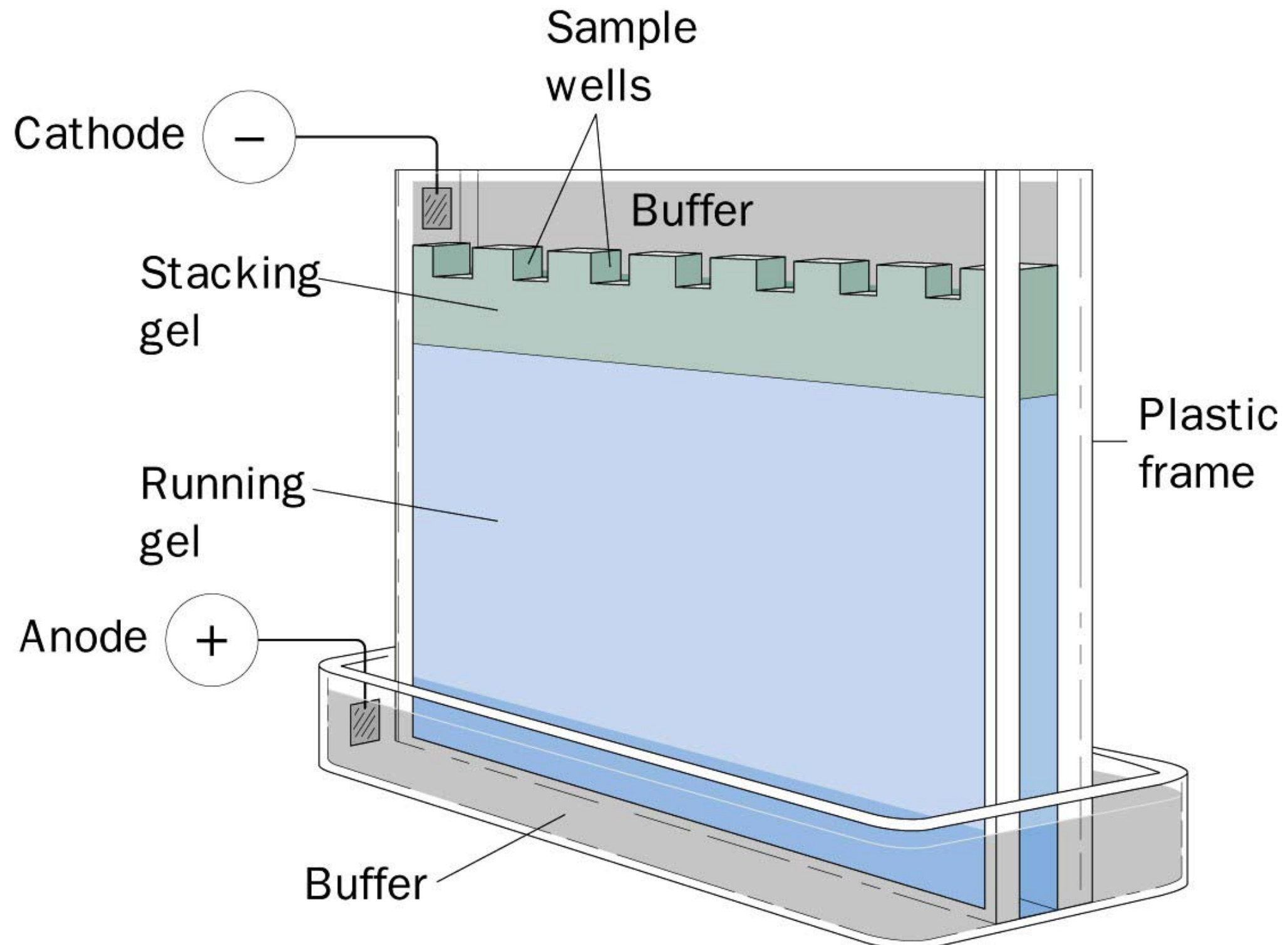
translator cascade
with perfect molecules



translator cascade
with imperfect molecules

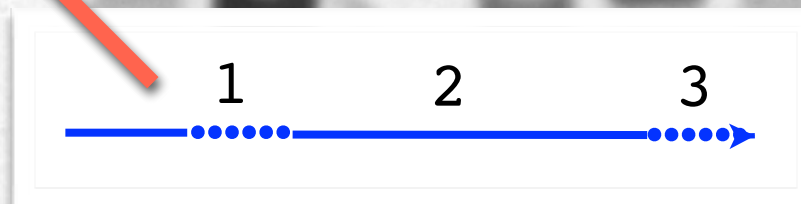
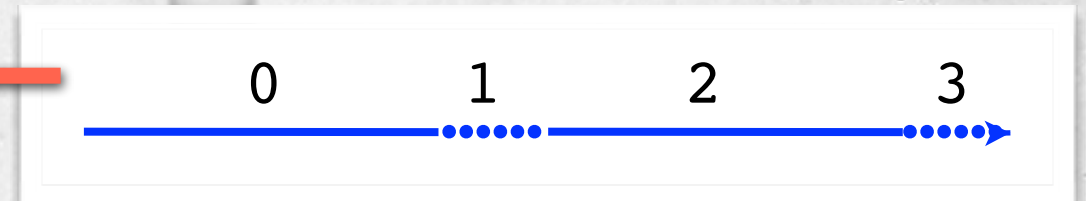
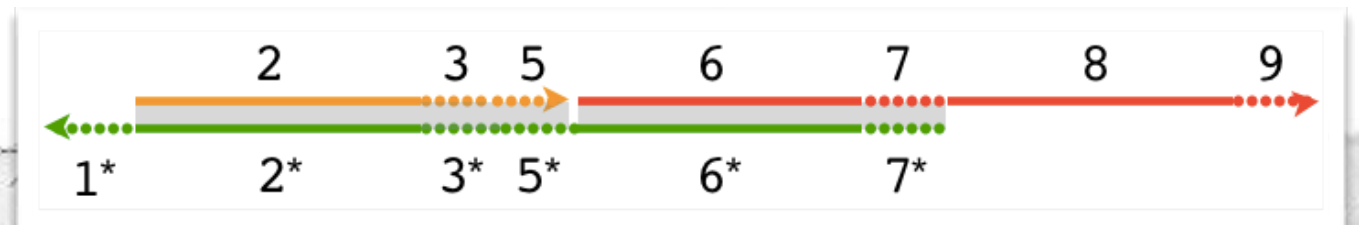


(Partial) solution to Problem 1

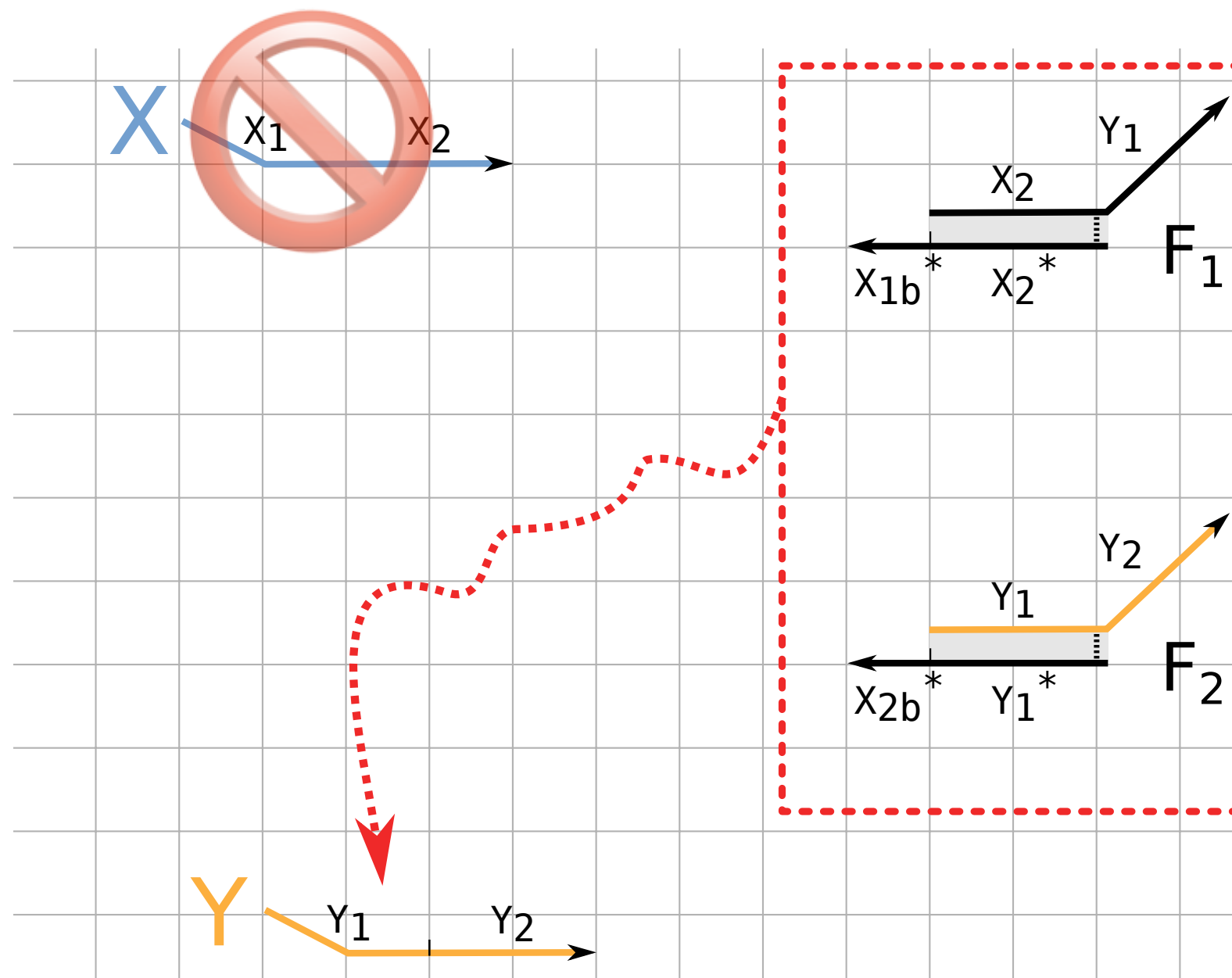
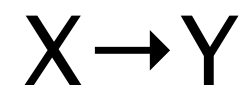


(Partial) solution to Problem 1

Complexes can also be purified by gel

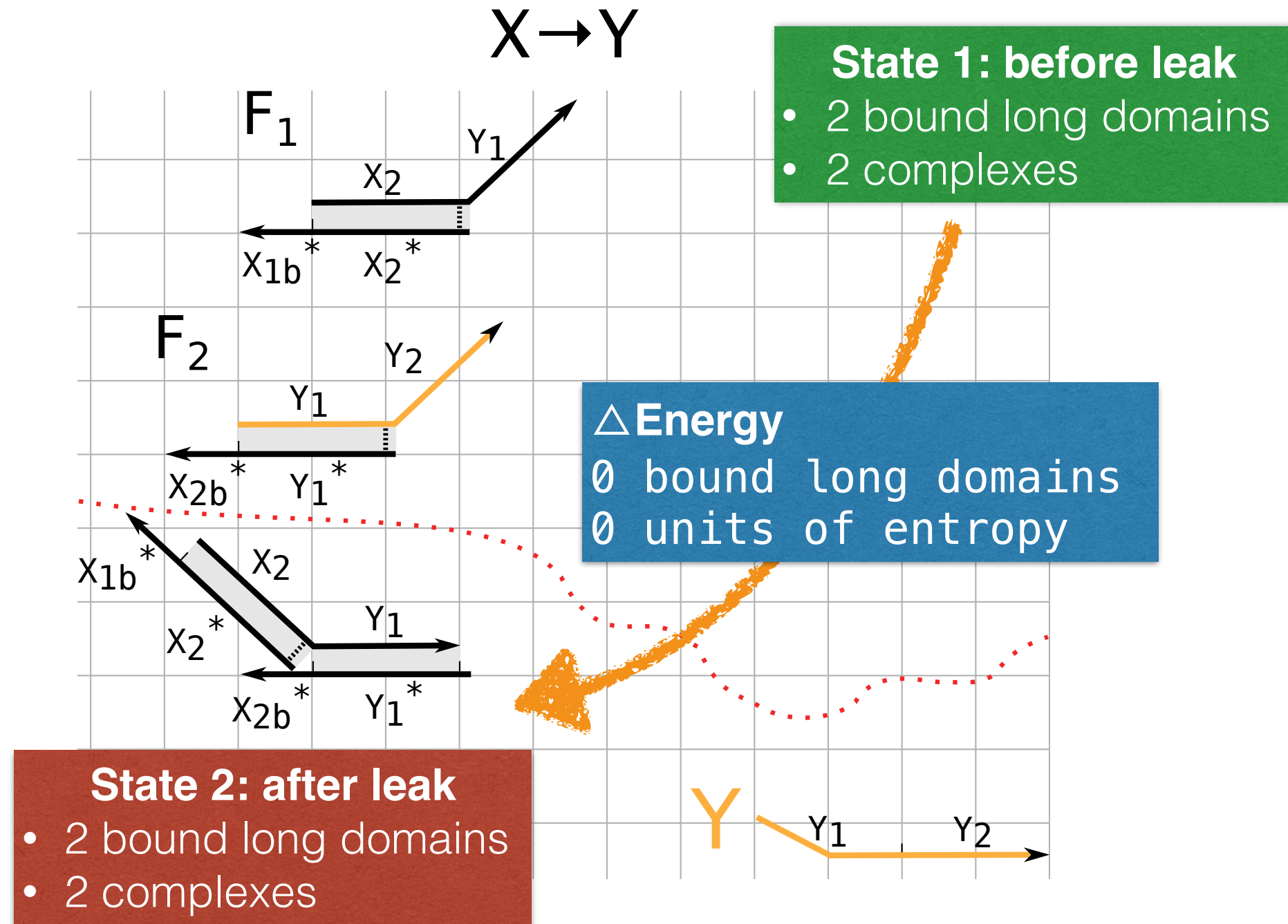


Problem 2: Spurious reactions occur (even with perfect molecules)



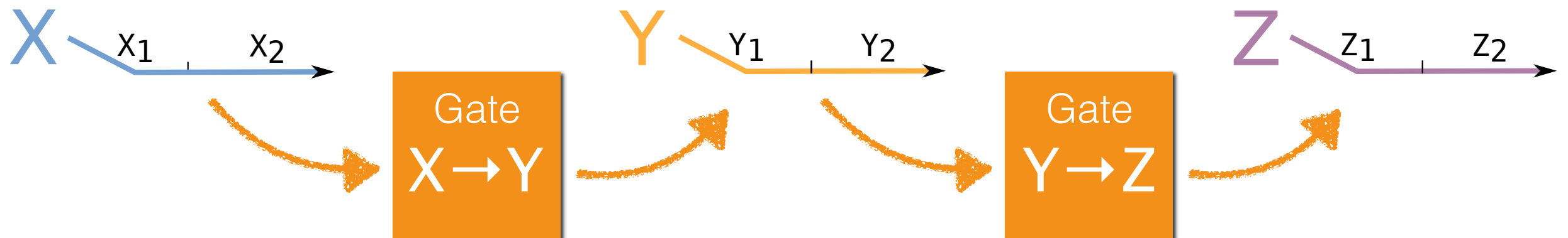
Y has been spuriously “produced”

Some rough energy accounting

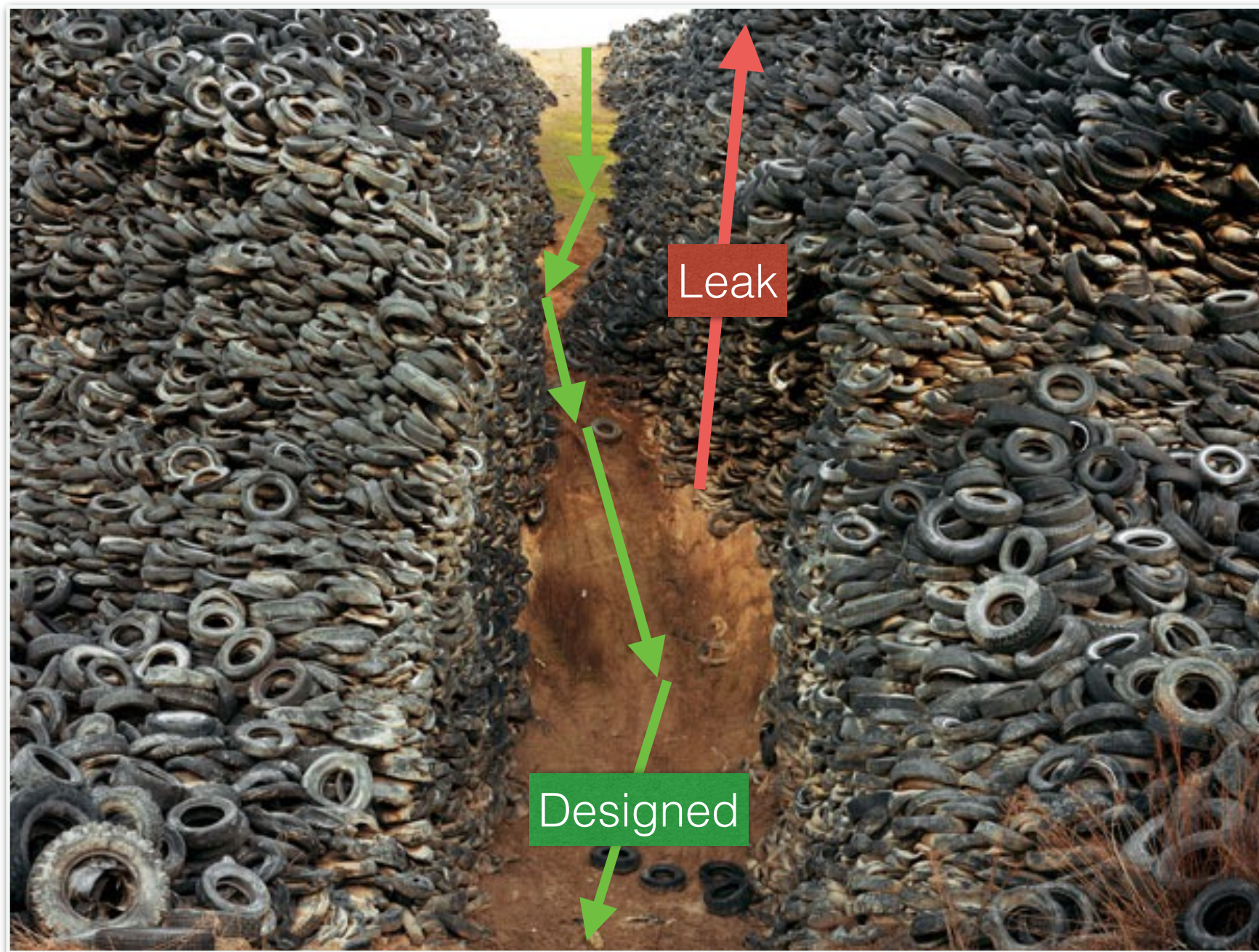


A Motivating Question

Can we rationally design
composable, leakless
DSD gates?

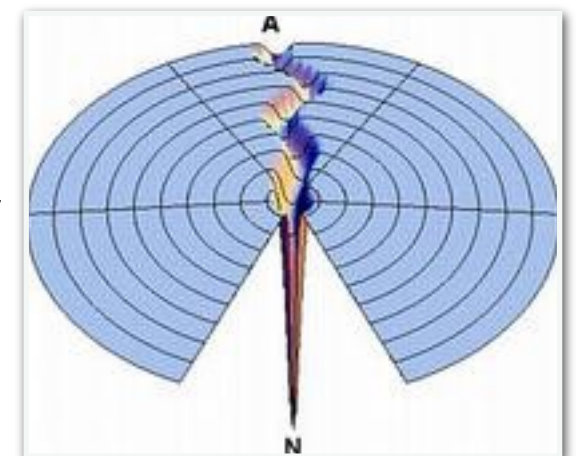


What do we mean by *leakless*?



“Golf funnel with deep groove” pathway

K. Dill & Bromberg (2002). Molecular Driving Forces.



(Partial) solution to Problem 2

For a redundancy parameter **N**, there exist **translator** and **AND** gates using **N** long domains that have the following property:

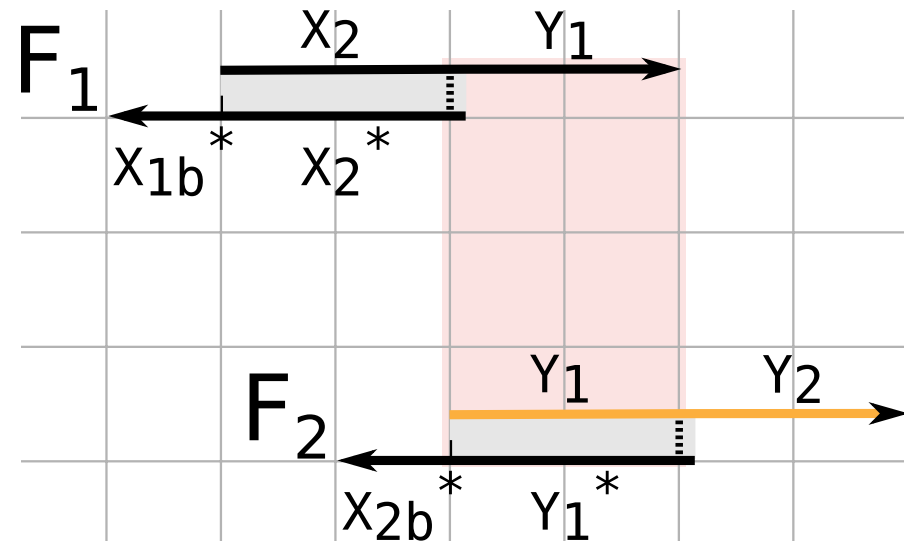
even at thermodynamic equilibrium,

the net leak decreases exponentially with **N**.

Thachuk, Winfree, David Soloveichik. (2015)
Leakless DNA strand displacement. DNA 21.

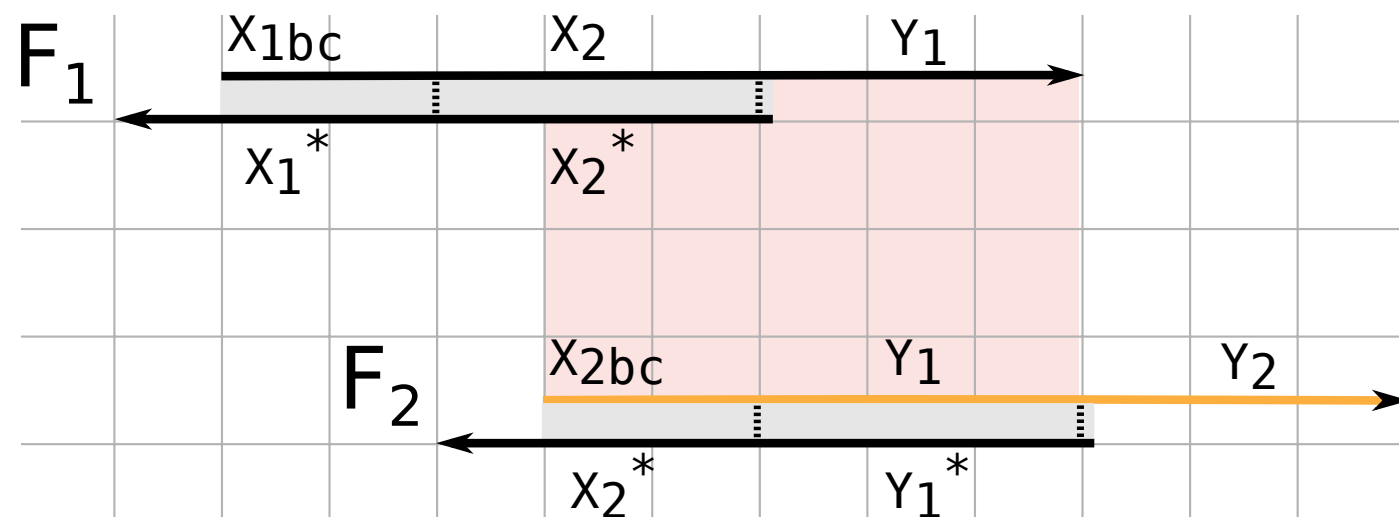


Typical translator using “Single Long Domain” (SLD)



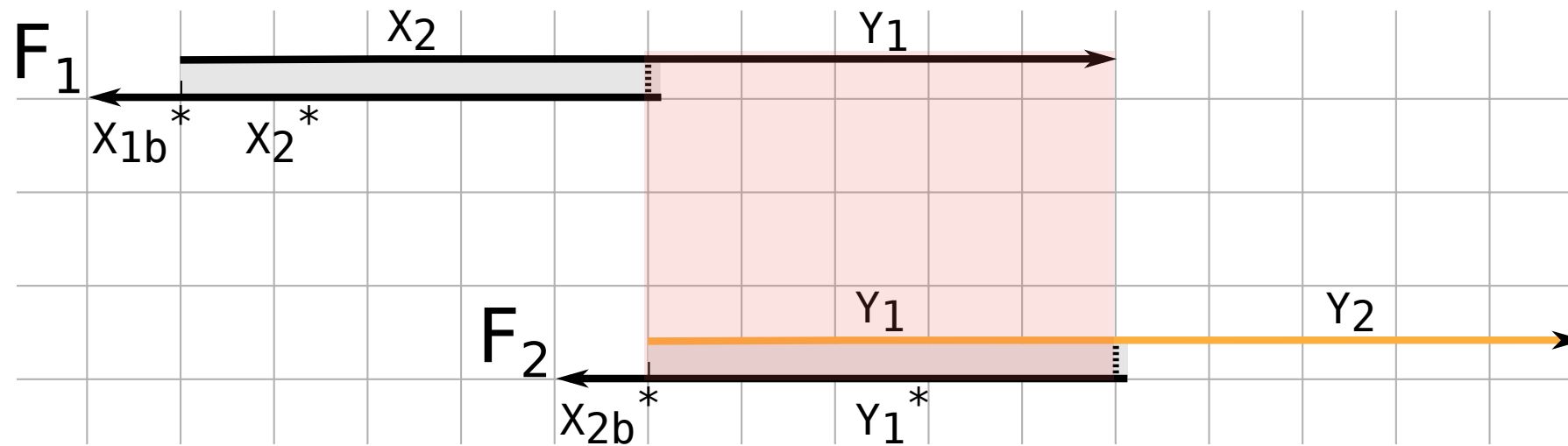
- Designed pathways: bimolecular
- Leak pathways: bimolecular

DLD translator using “Double Long Domain” (DLD)



- Designed pathways: bimolecular
- Leak pathways: **trimolecular**

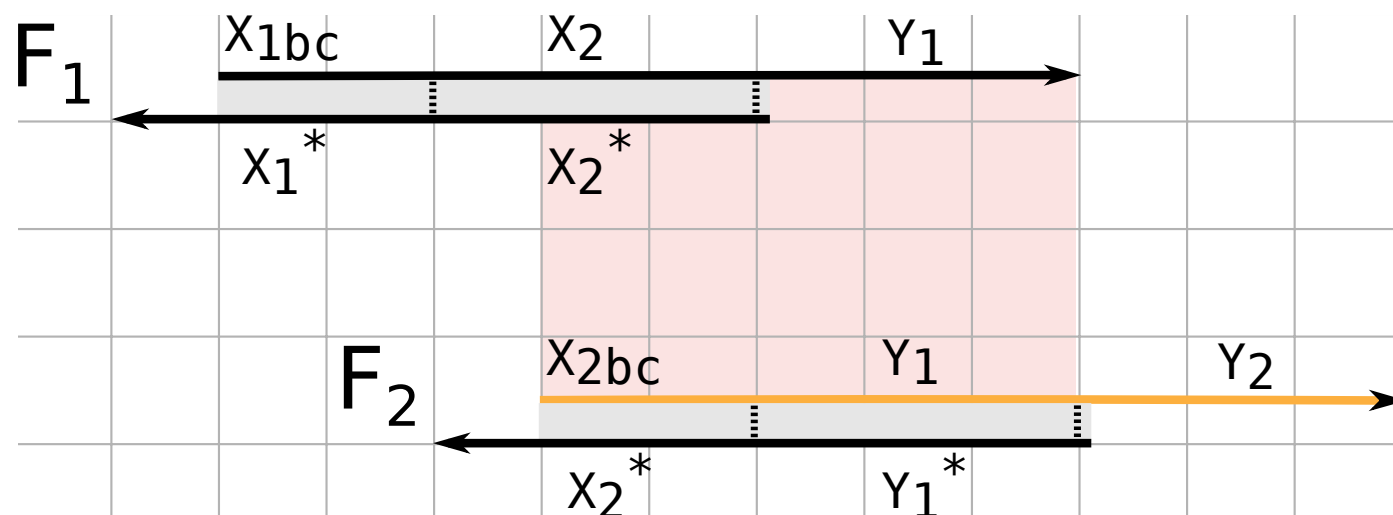
Typical translator using “Single Long Domain” (SLD)



- Designed pathways: bimolecular
- Leak pathways: bimolecular

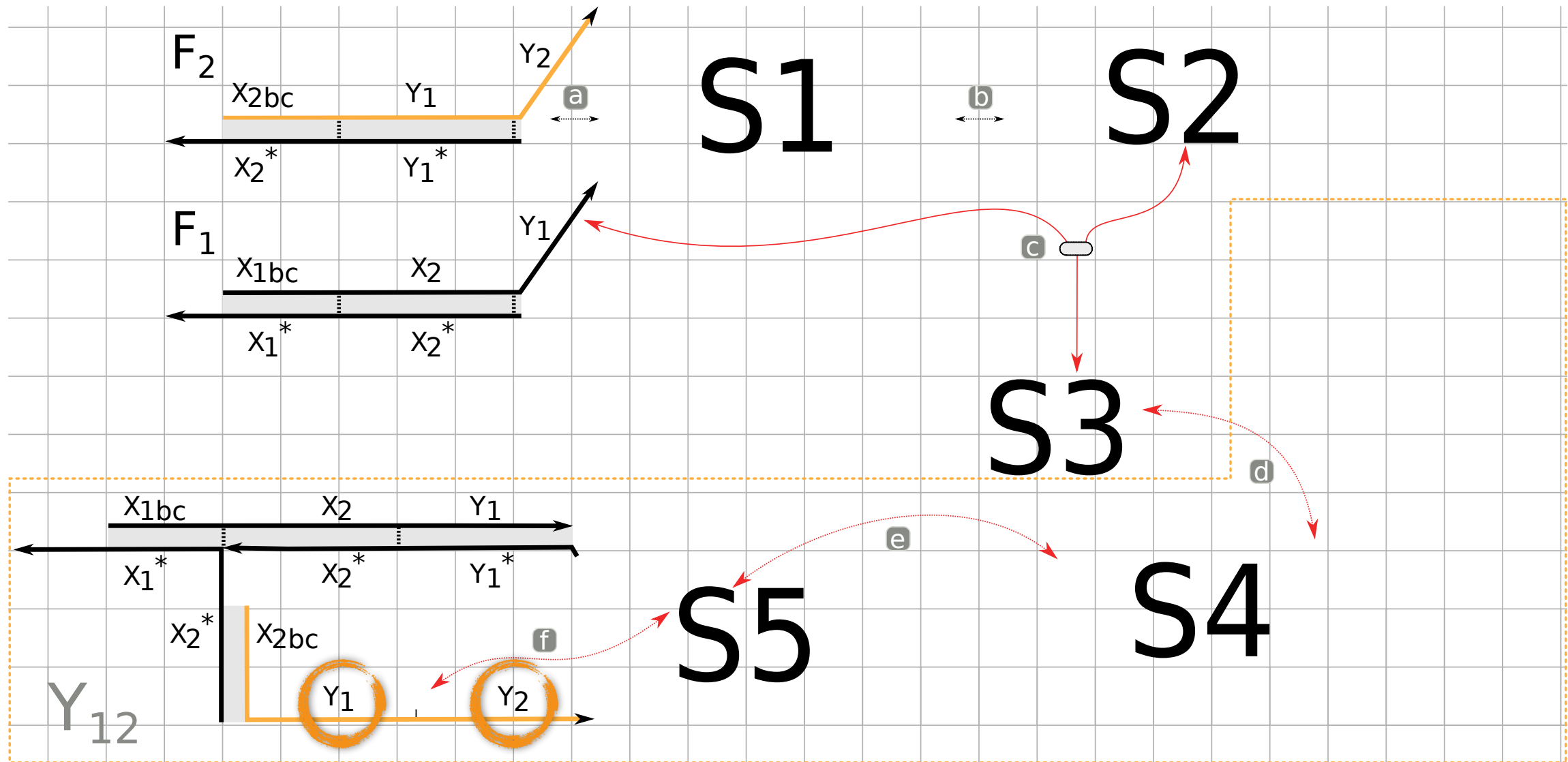
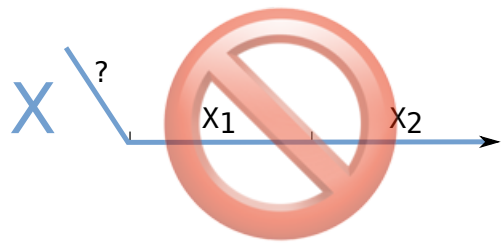
Lengthening recognition domains
does not help

DLD translator using “Double Long Domain” (DLD)

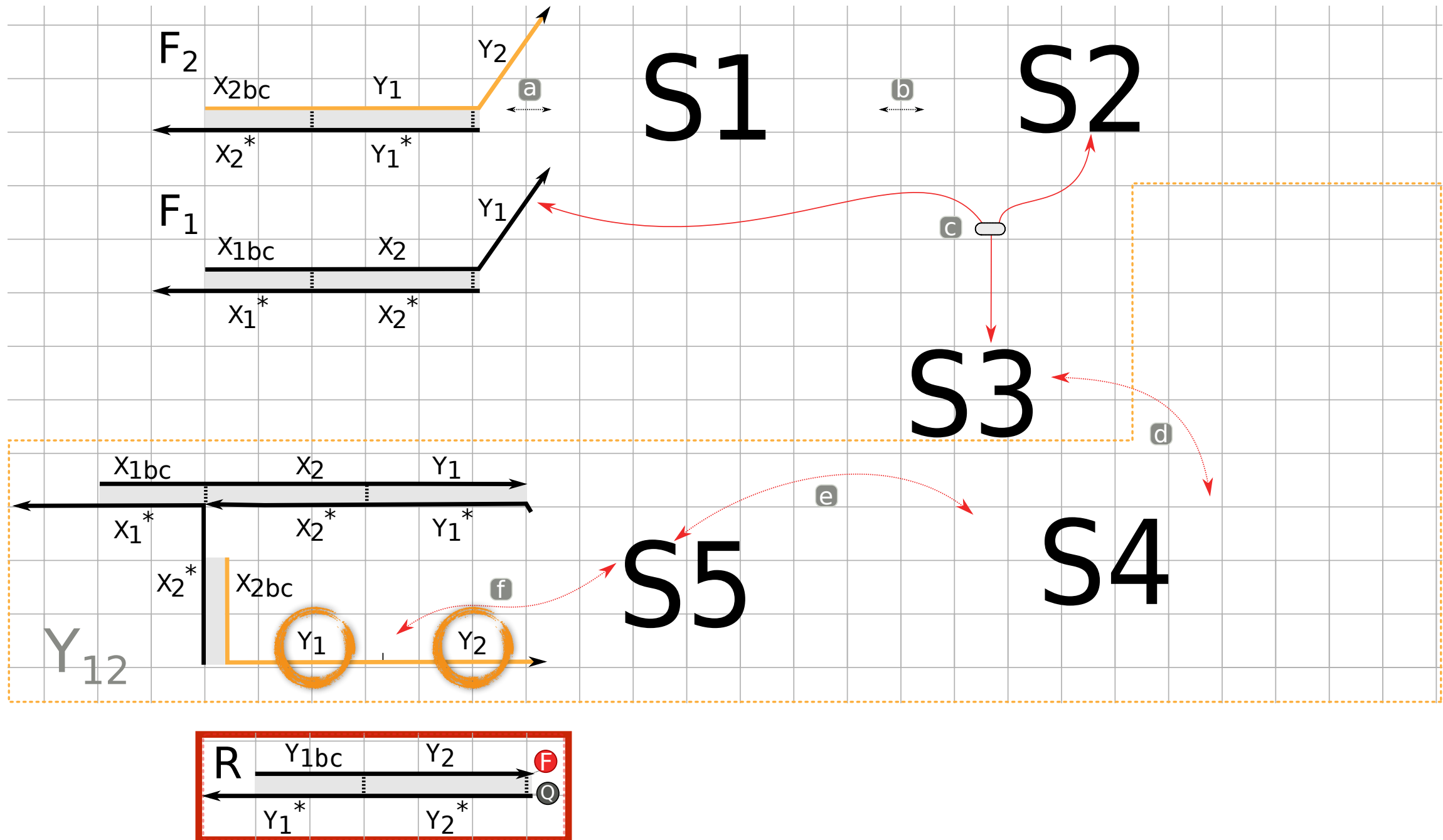
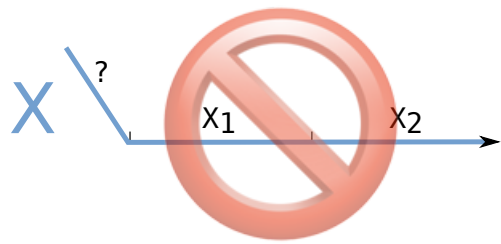


- Designed pathways: bimolecular
- Leak pathways: **trimolecular**

DLD translators are intrinsically less “leaky”

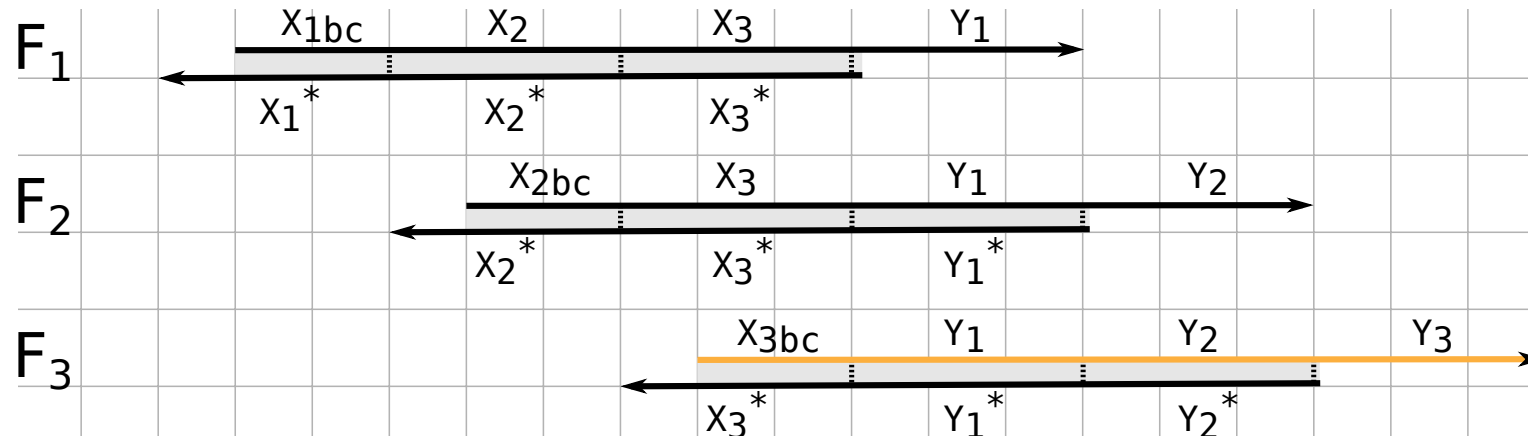


DLD translators are intrinsically less “leaky”



Can we generalize the
DLD motif?

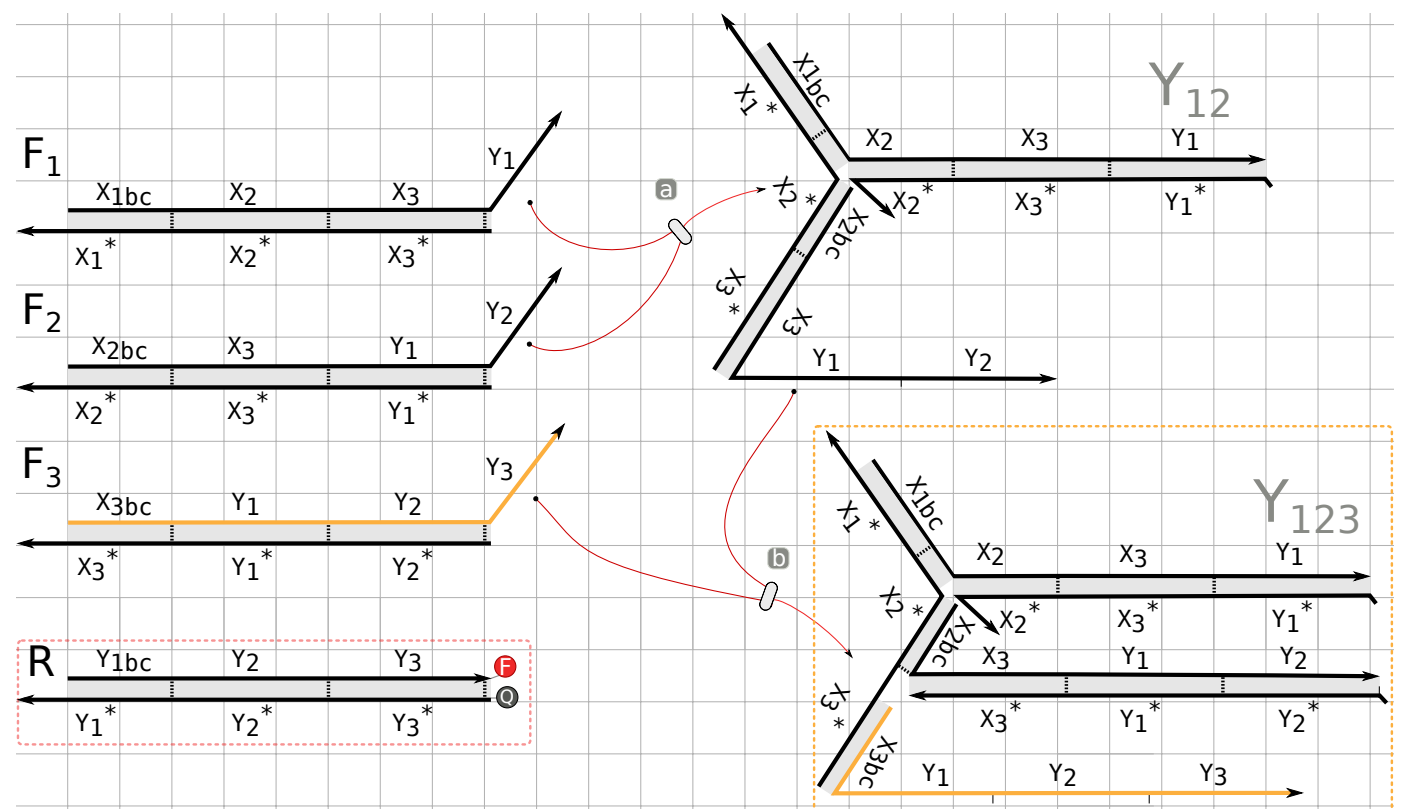
Translator using Triple Long Domain (TLD) motif



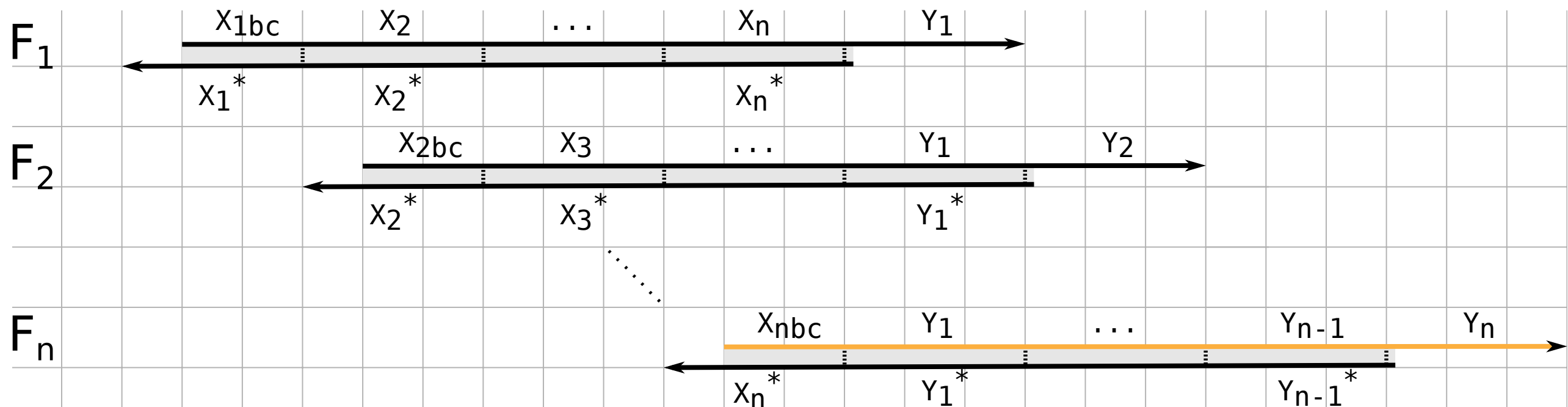
Three fuel complexes must combine to activate output signal.

Δ Energy

0 bound long domains
-2 units of entropy



Translator using N Long Domain (NLD) motif

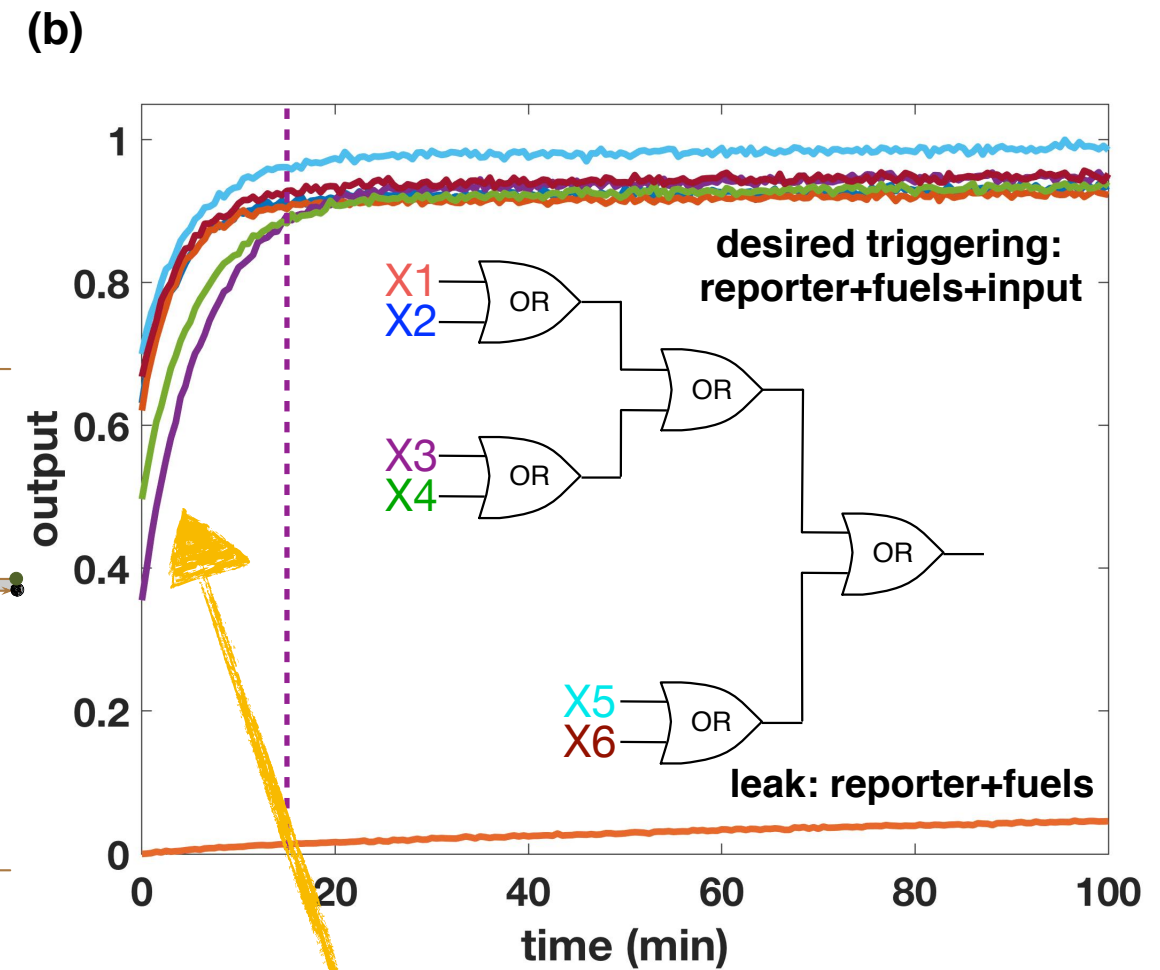
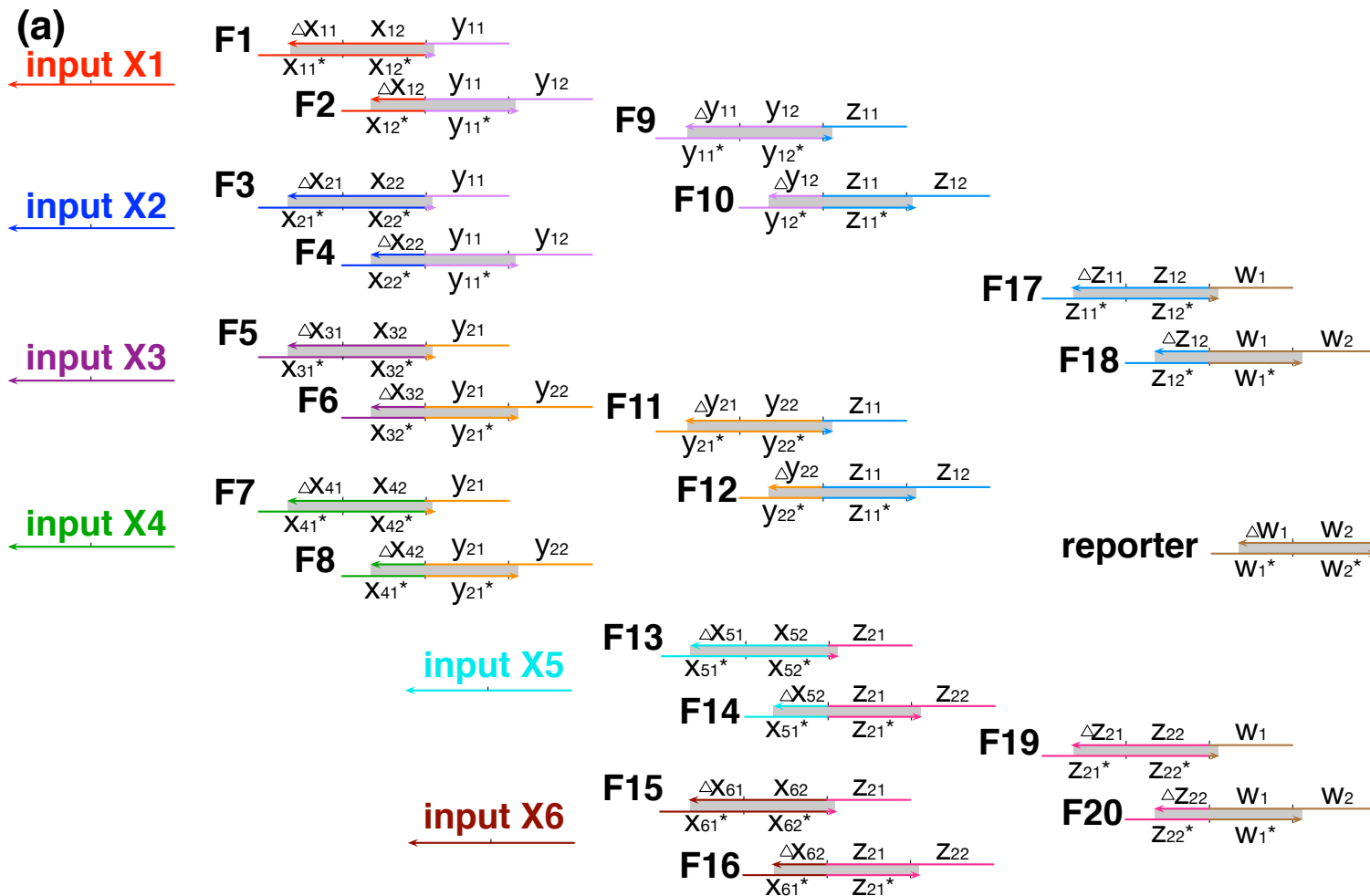


N fuel complexes must combine to activate output signal.

Δ Energy to leak state

0 bound long domains
 $-(N-1)$ units of entropy

Building OR circuits from DLD translators



Half-time completion ~6 minutes
No signal restoration used.

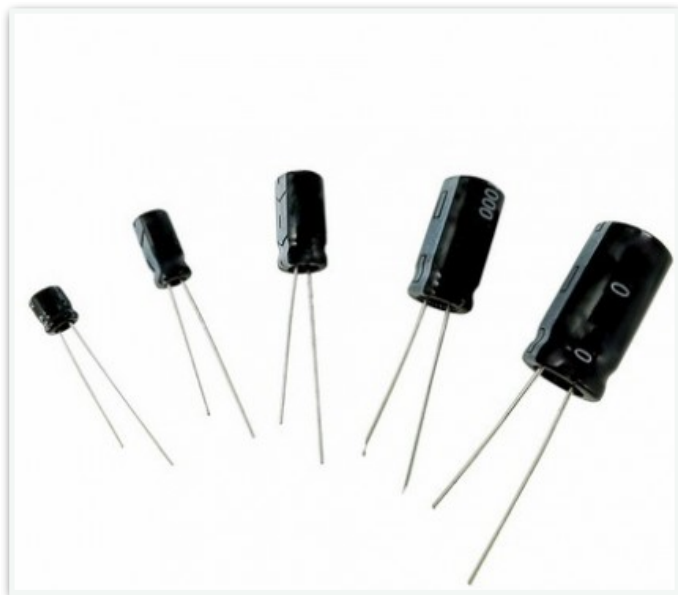
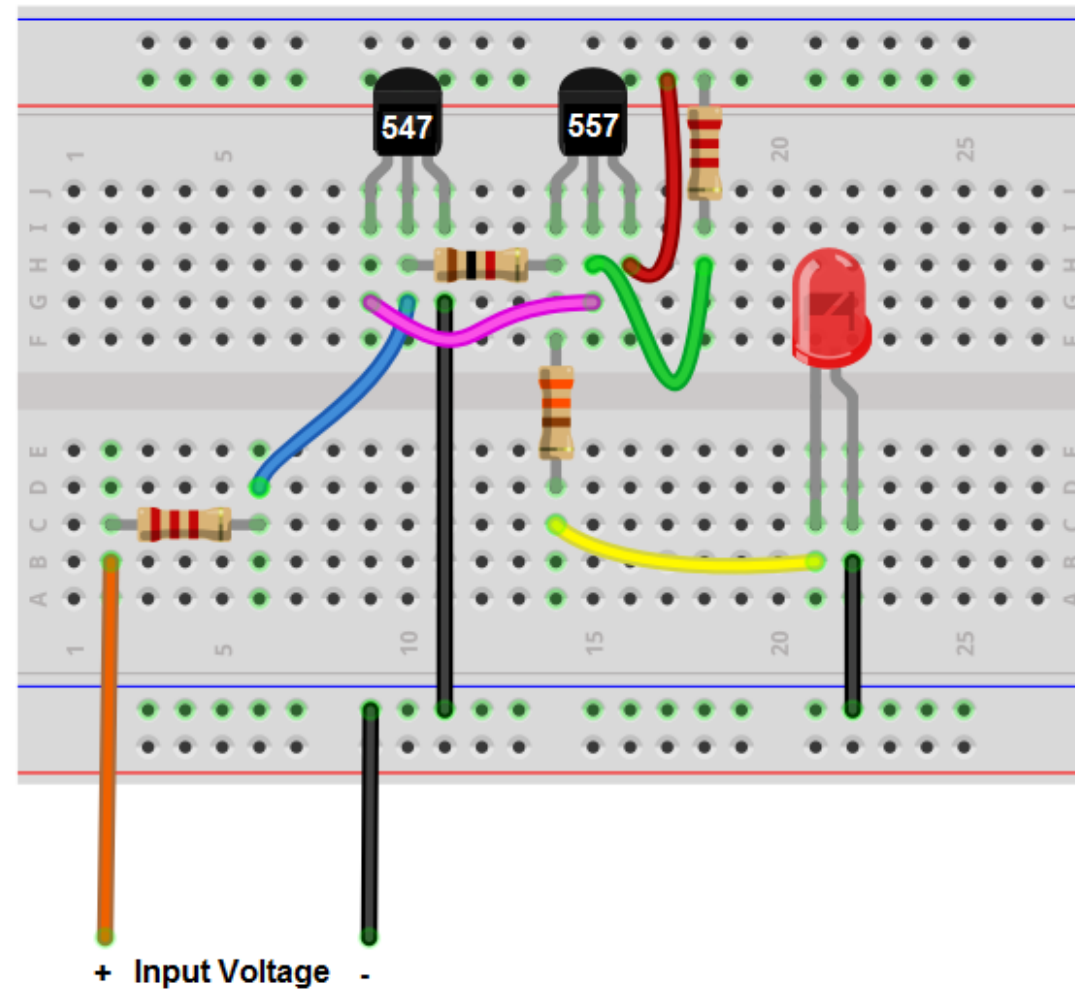
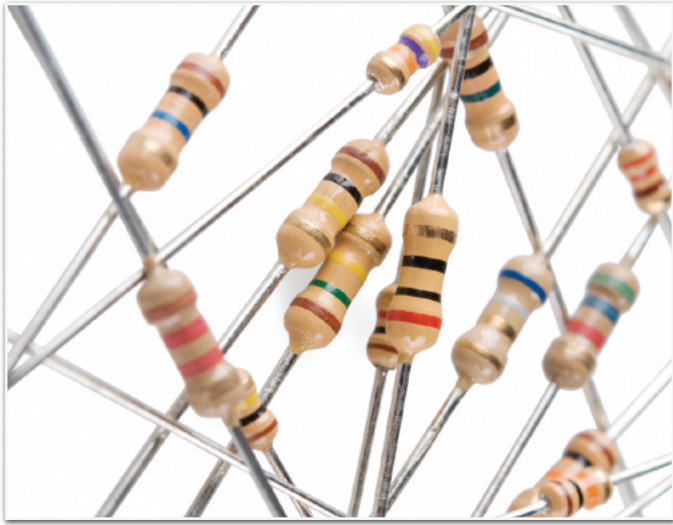
[fuel]=[input]=1000nM
[reporter]=500nM

Boya Wang, Thachuk, Ellington, Winfree, **David Soloveichik**. (In Review)
Effective Design Principles for Leakless Strand Displacement Systems

Tutorial Outline

- ▶ Review of strand displacement
- ▶ Building and composing logic gates
- ▶ Tools for designing and verifying circuits
- ▶ Robustness of strand displacement
- ▶ **(Bonus) DSD circuits the easy way**

Does it need to be this
difficult to build a circuit?

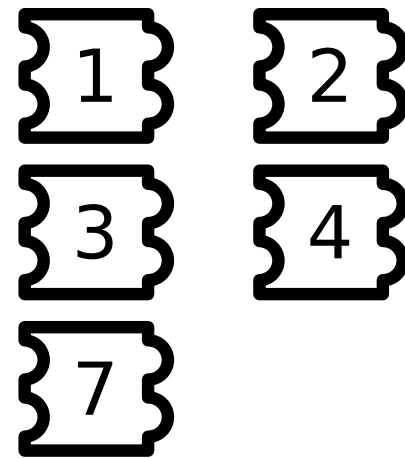


Molecular breadboard 1.0

input signals

A1 B1
A2 B2
A3 B3
A4 B4
A7 B7

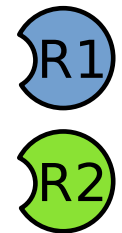
reaction gates



wires

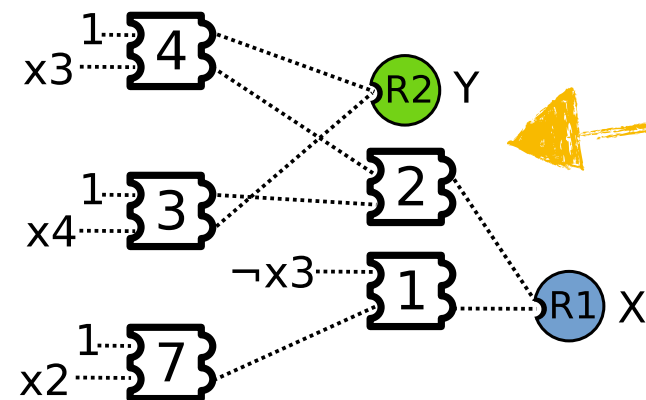
X1.....R1 Y3.....A2
Y1.....R2 Y3.....B2
X2.....B1 X4.....B1
X2.....R2 X4.....B2
Y2.....R1 Y4.....A2
X3.....B1 Y4.....R2
X3.....B4 X7.....A1
X3.....R2

reporters



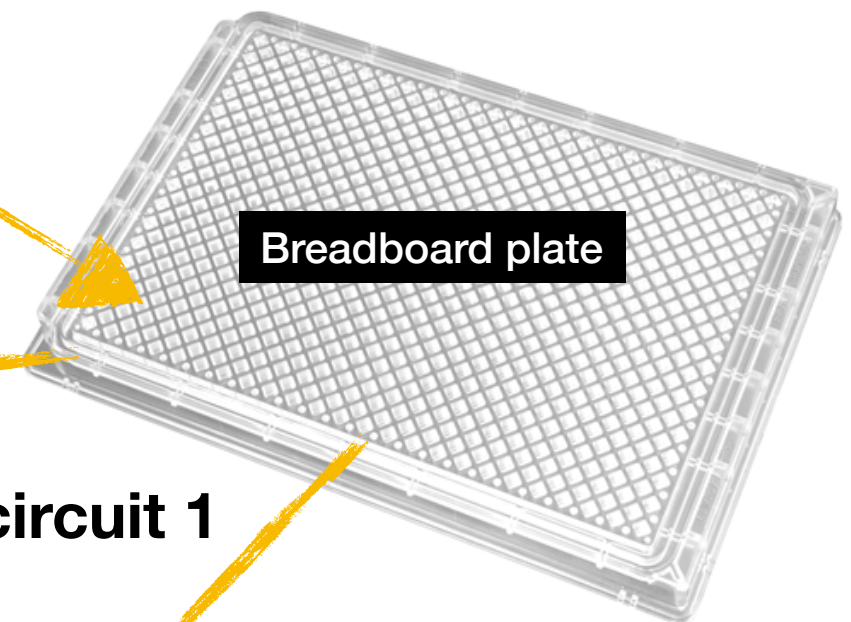
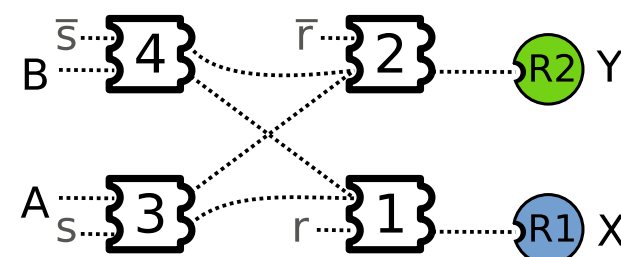
Load breadboard components
onto 384-well plate

Built using leakless motif



circuit 2

circuit 1

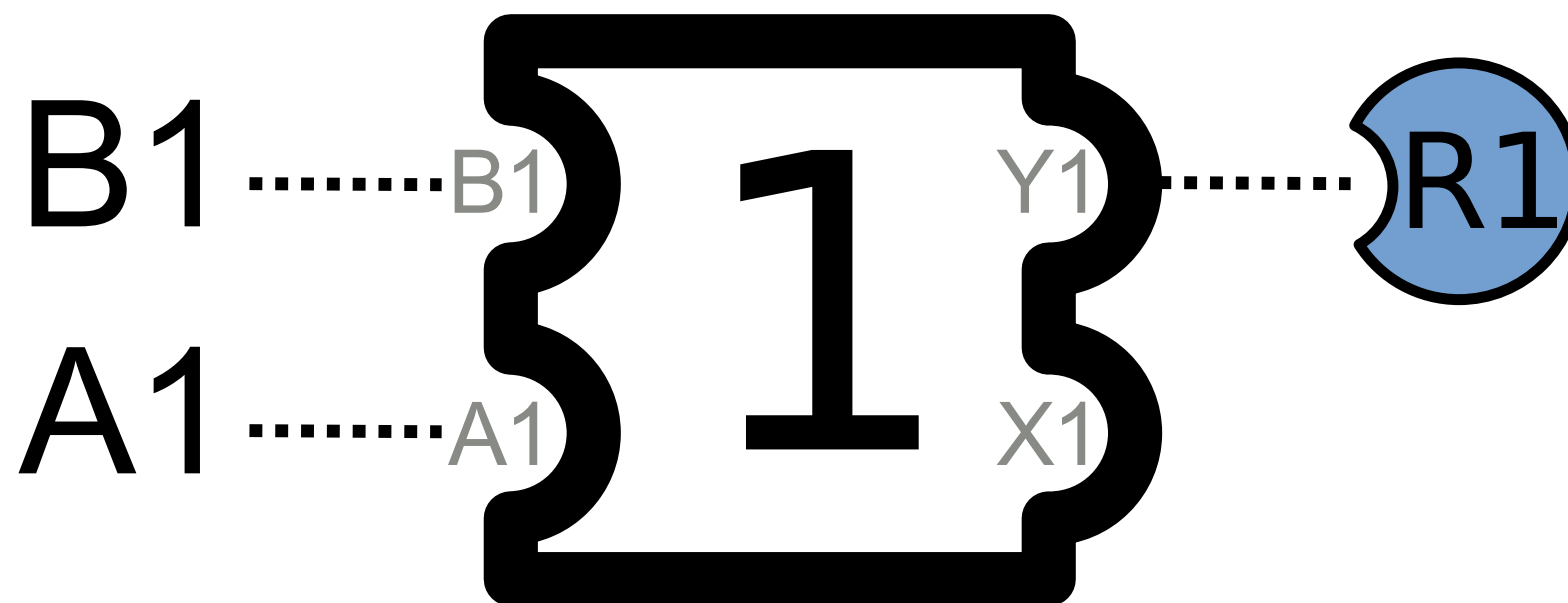


Breadboard plate

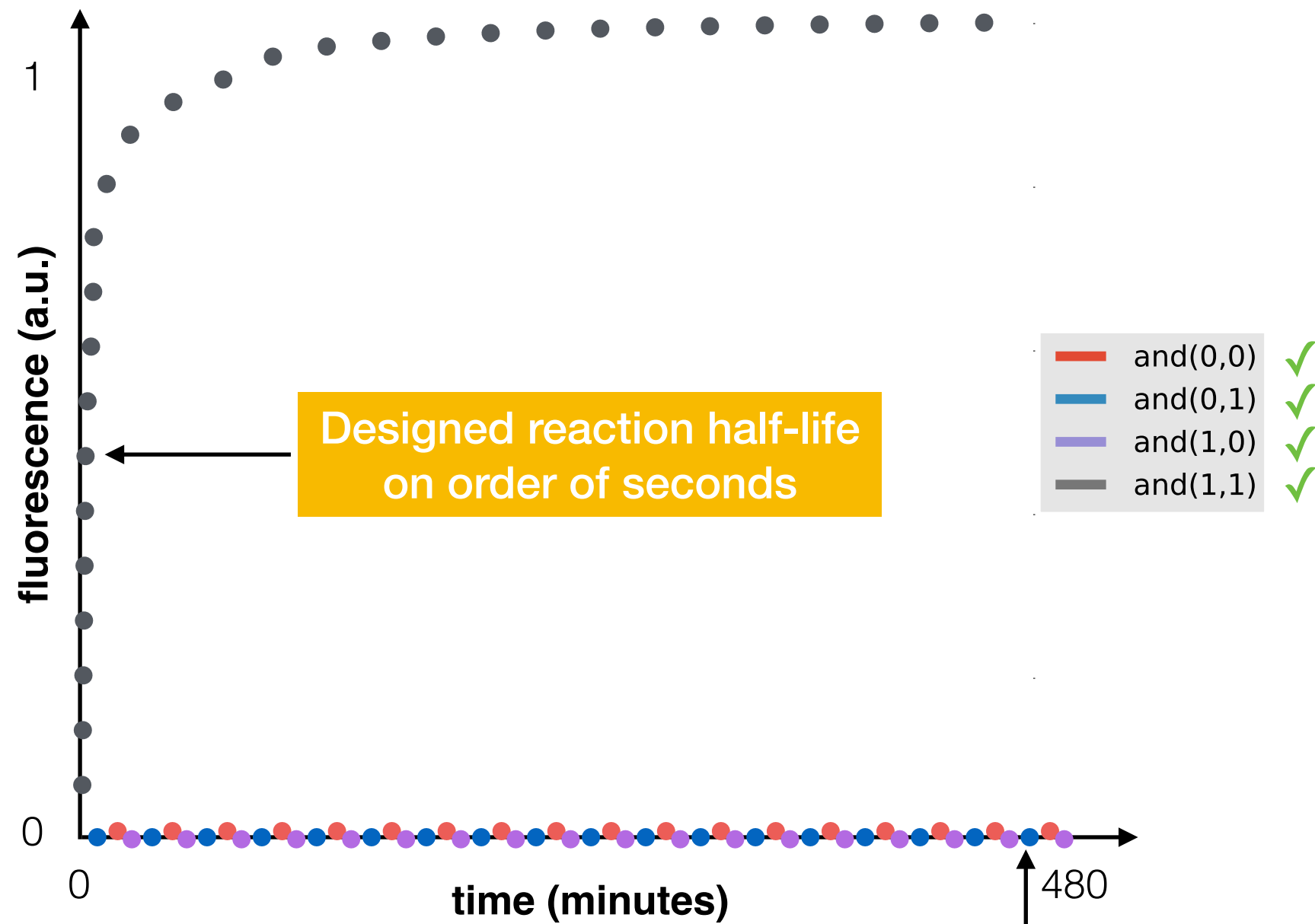
Testing breadboard components

- ▶ Typical DSD circuits are 50nM - 200nM concentration
(our circuits can operate at these concentrations)
- ▶ To demonstrate *robustness*, all experiments will be at 2uM
(~20x higher than typical concentrations)

AND gate

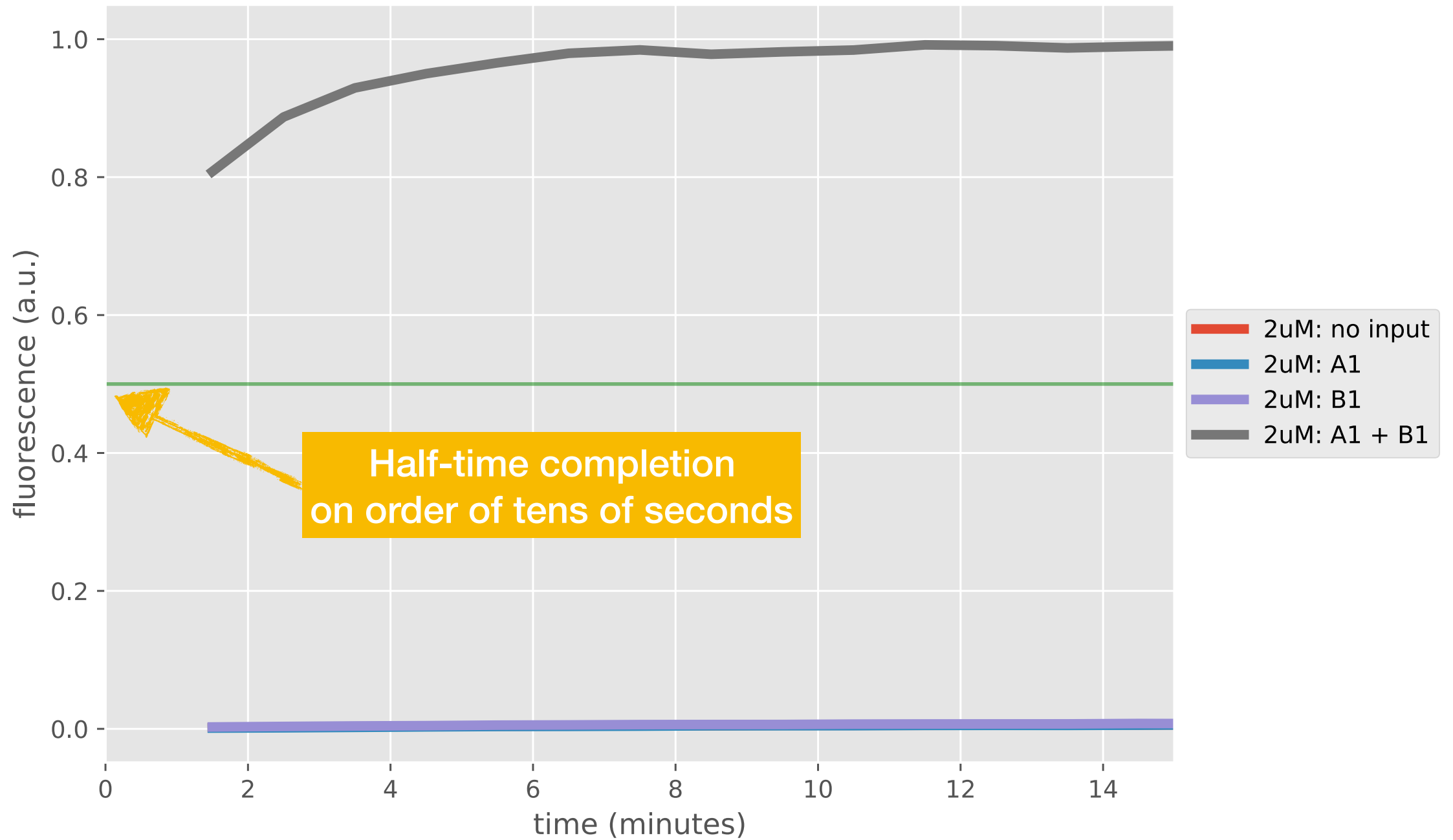
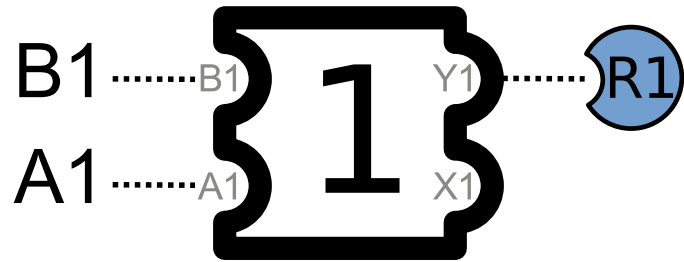


Ideal AND gate simulation @ 2 μ M



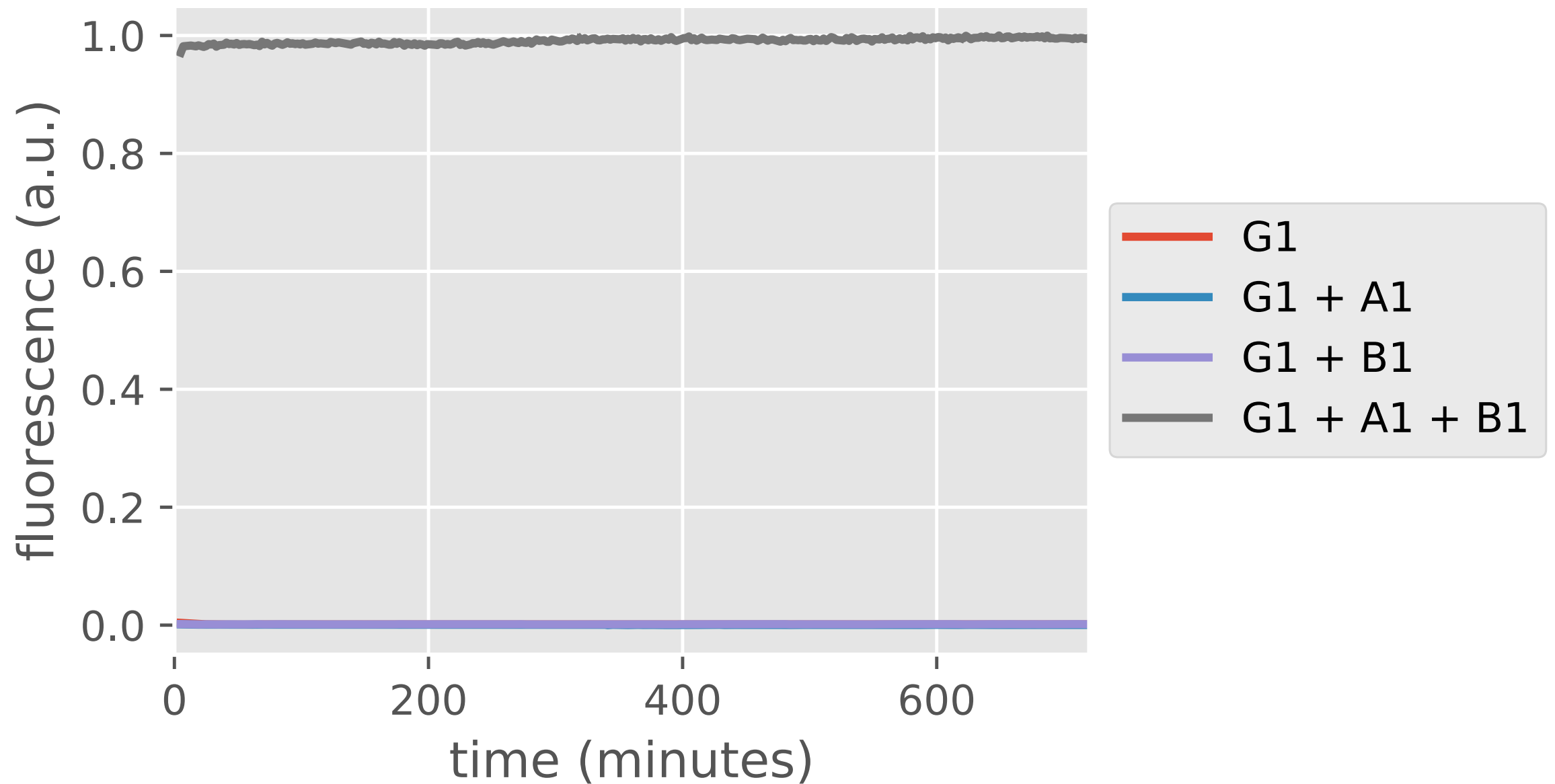
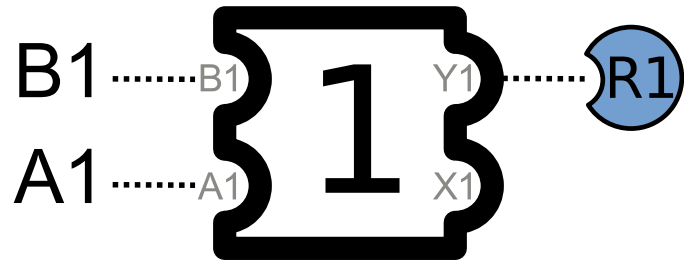
Spurious reactions not observable
on order of hours

AND gate @ 2 μ M



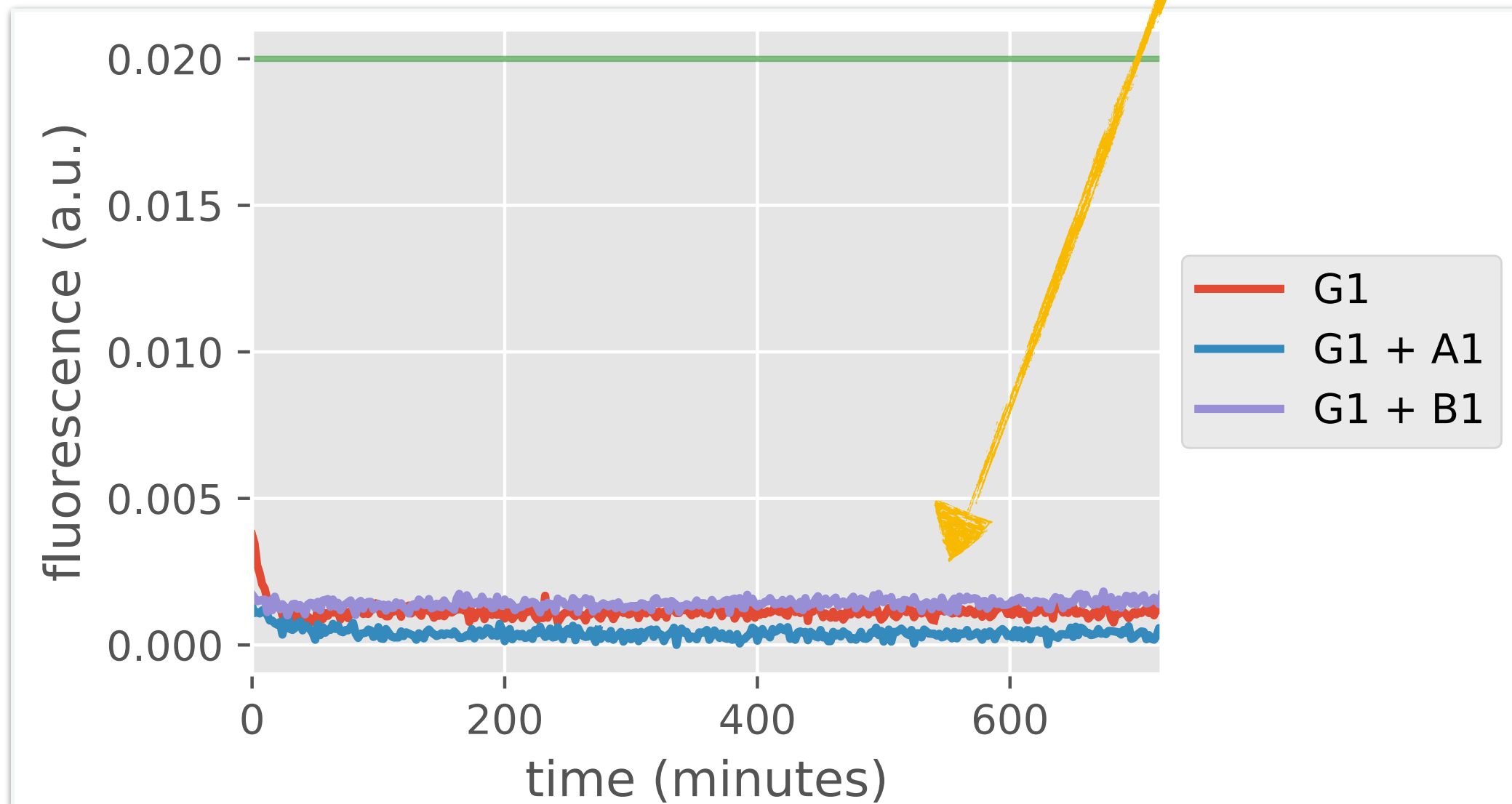
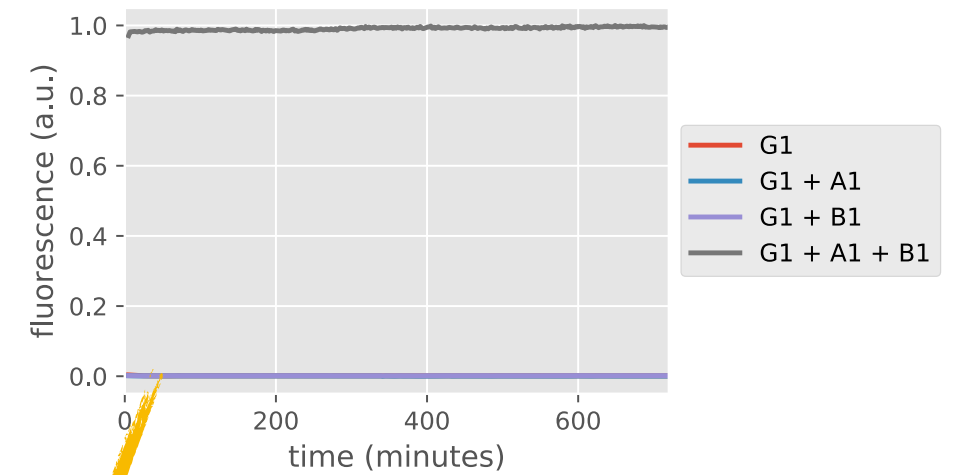
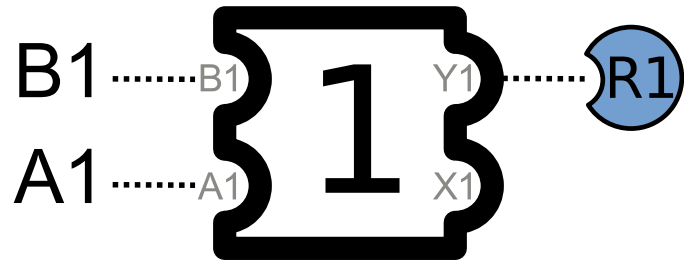
[fuel]=[input]=2 μ M, [reporter]=1 μ M

AND gate @ 2 μ M (12 hours)



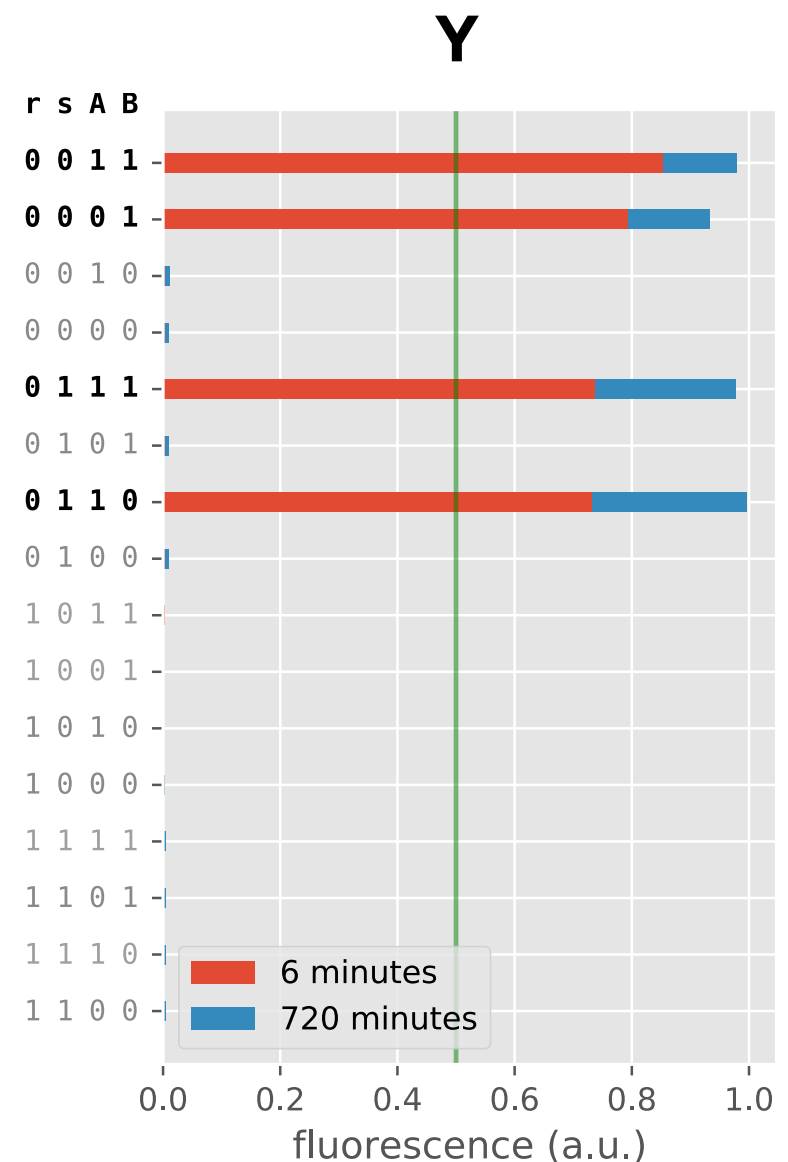
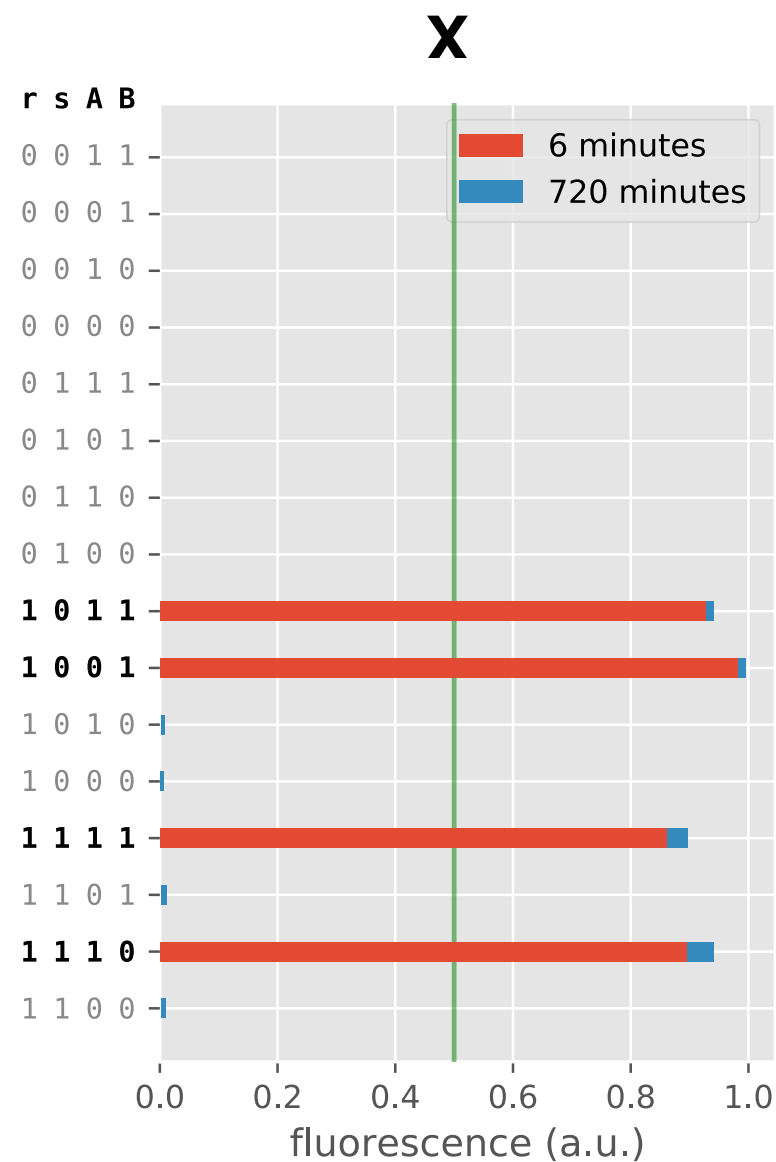
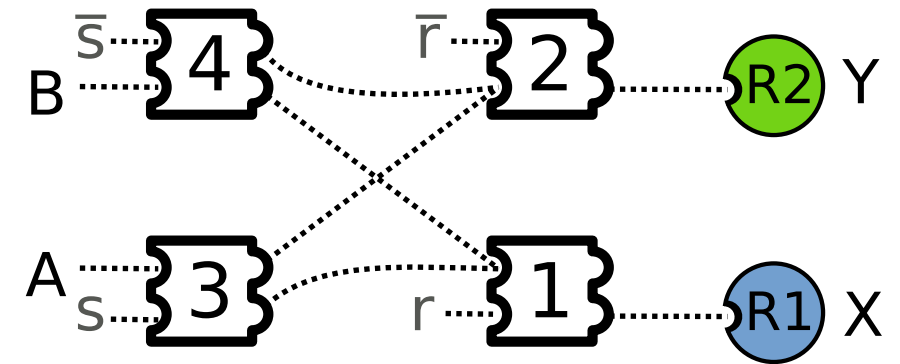
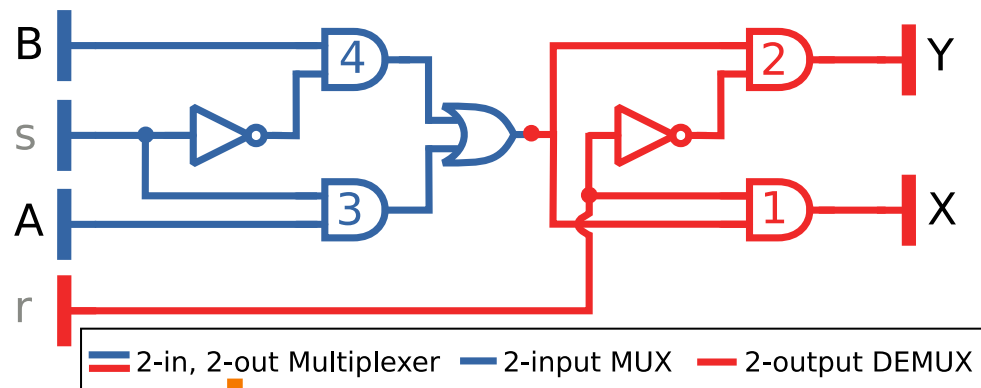
[fuel]=[input]=2uM, [reporter]=2.5uM

AND gate @ 2 μ M (12 hours)

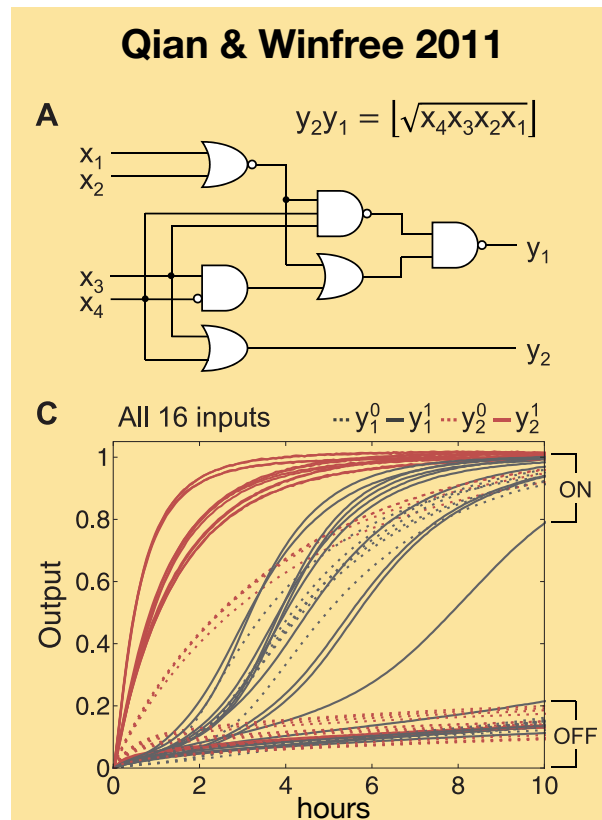


[fuel]=[input]=2 μ M, [reporter]=2.5 μ M

Multiplexer-Demultiplexer

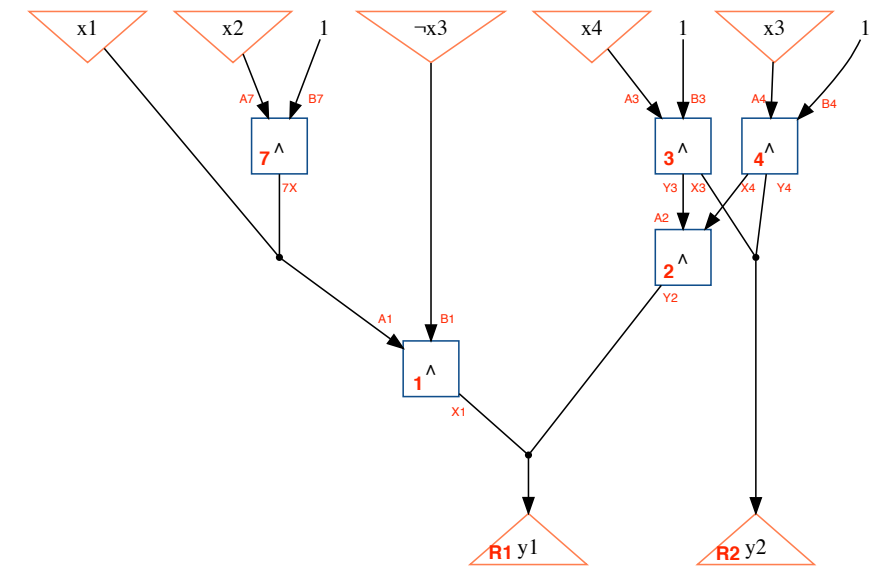
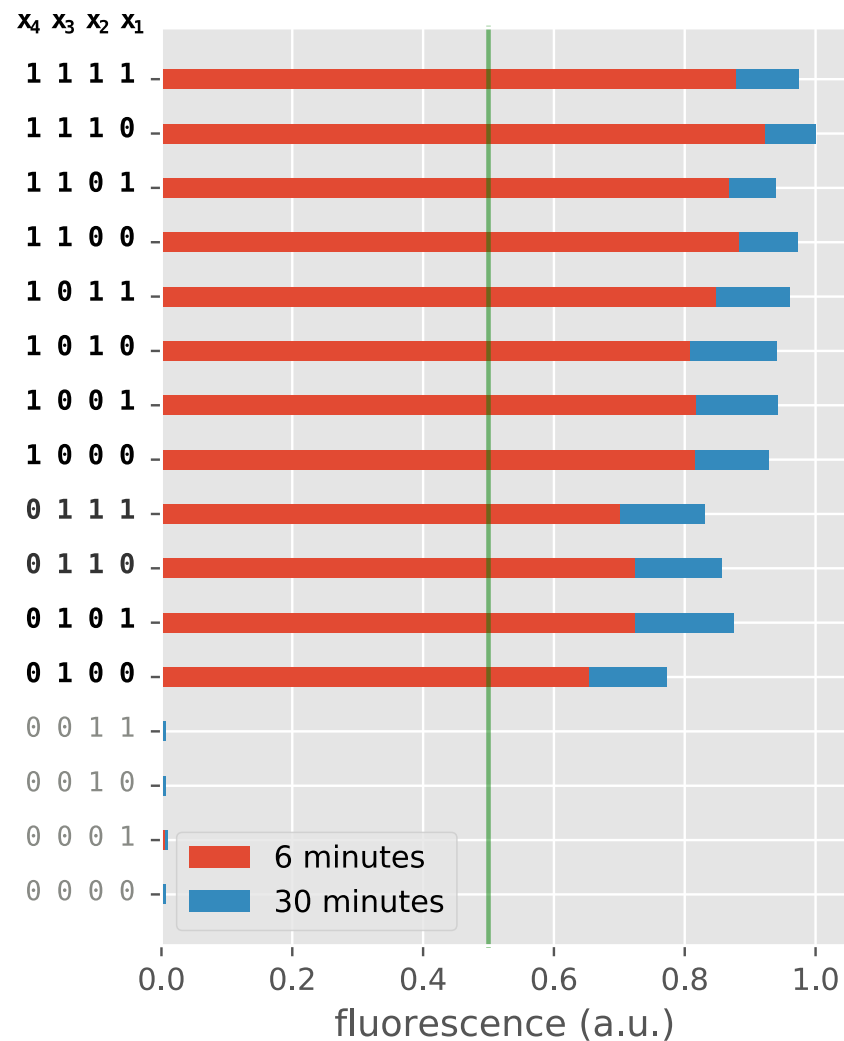


Large circuits that are fast

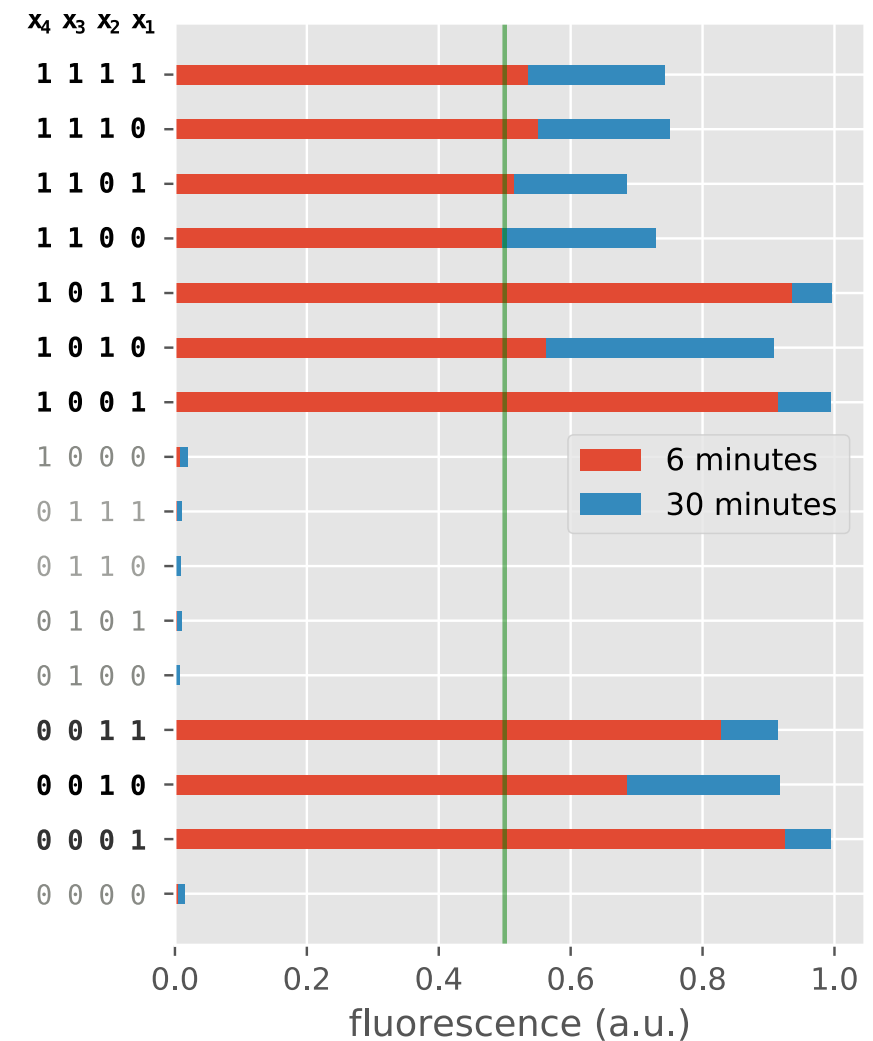


Reaction half-time improved
from ~6 hours to < 6 minutes

Y2



Y1

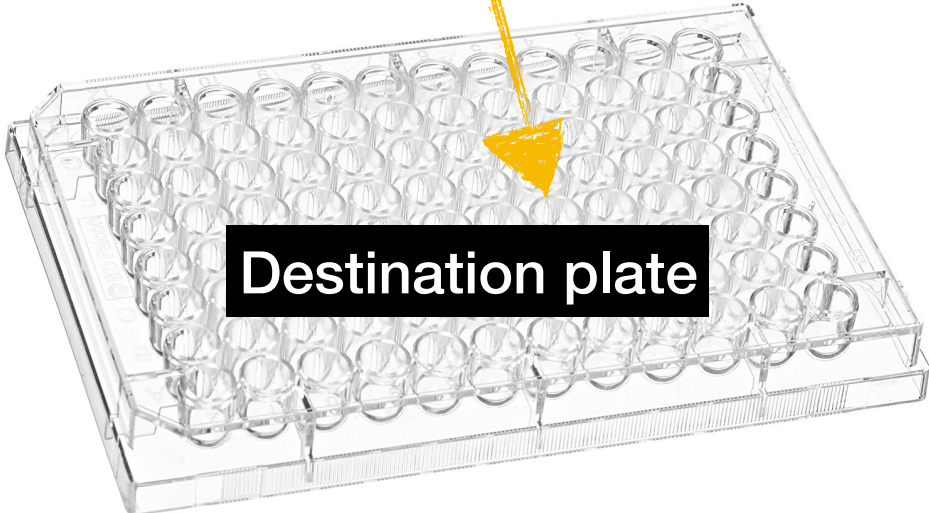
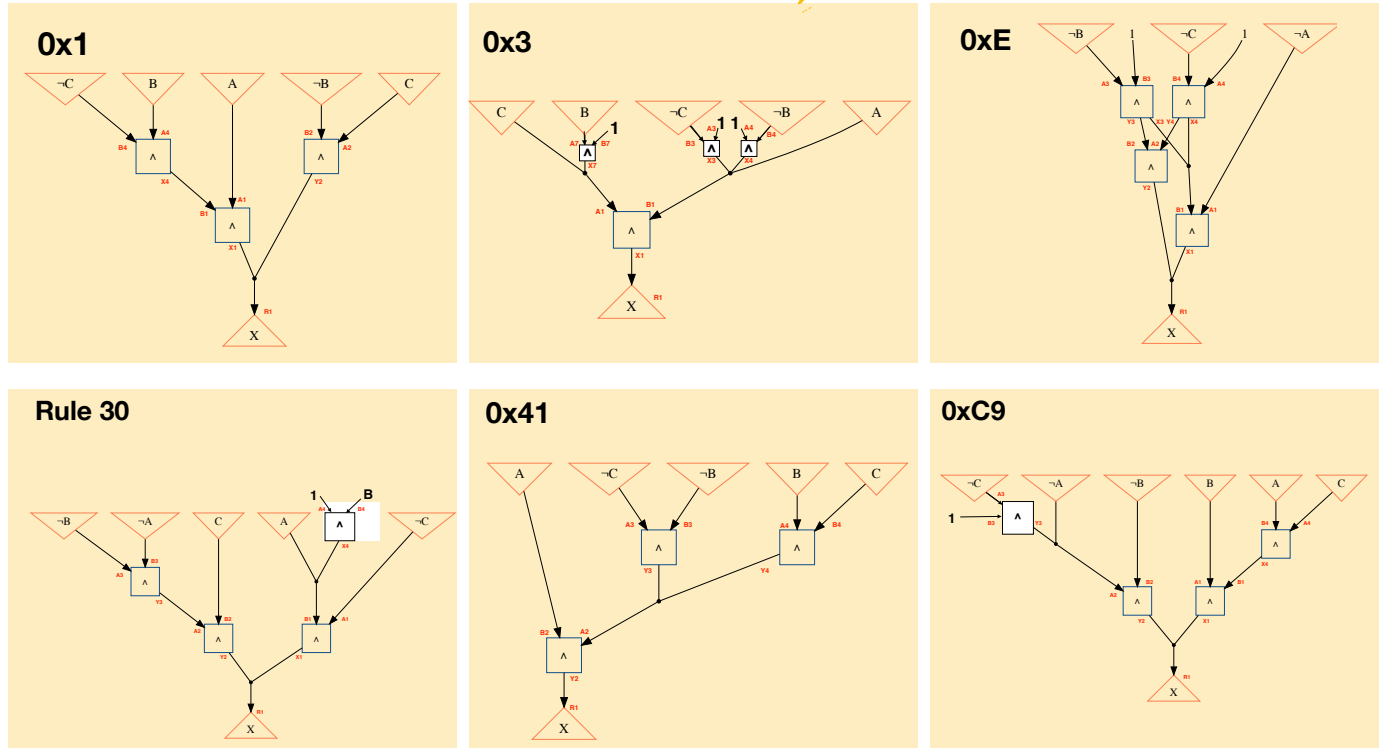


Breadboard plate

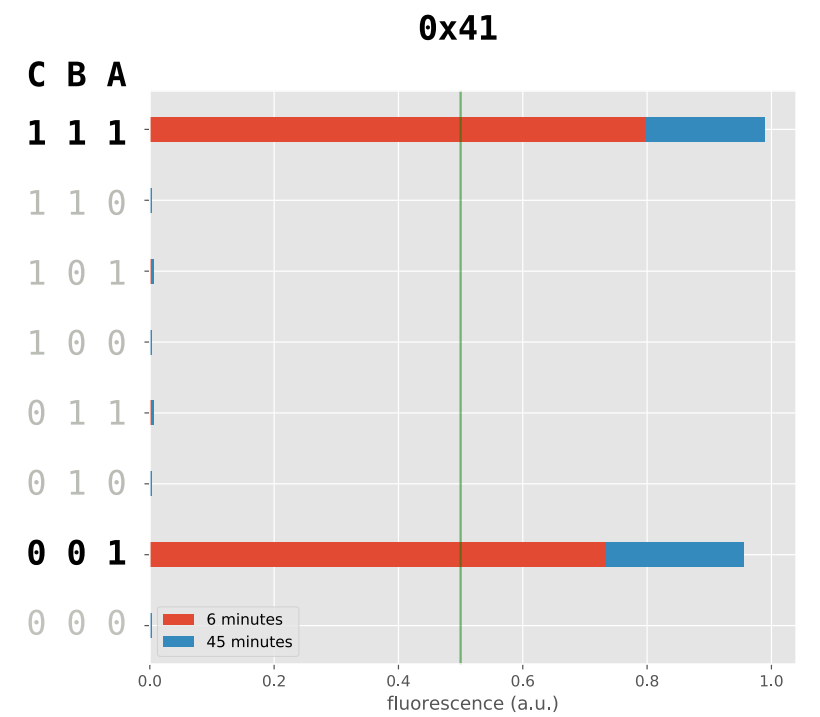
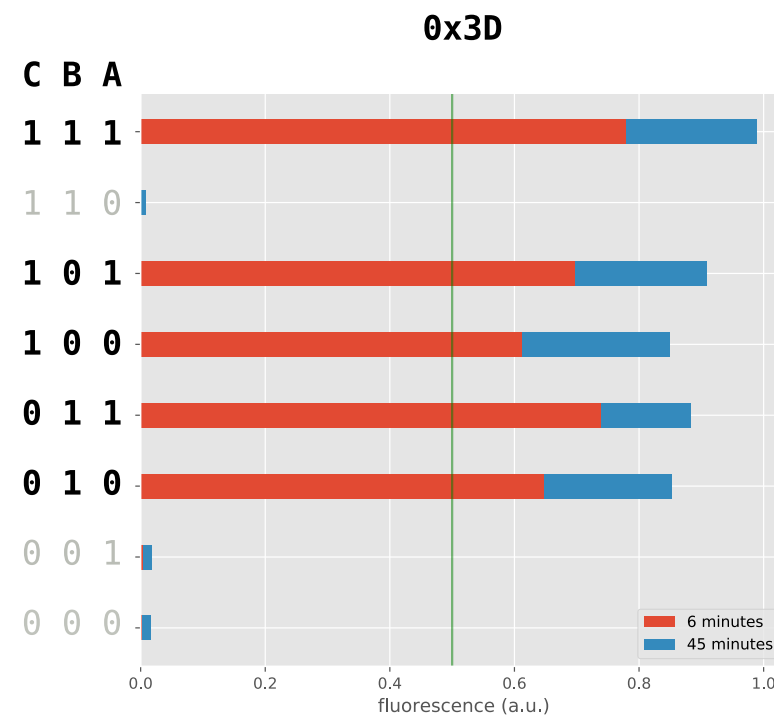
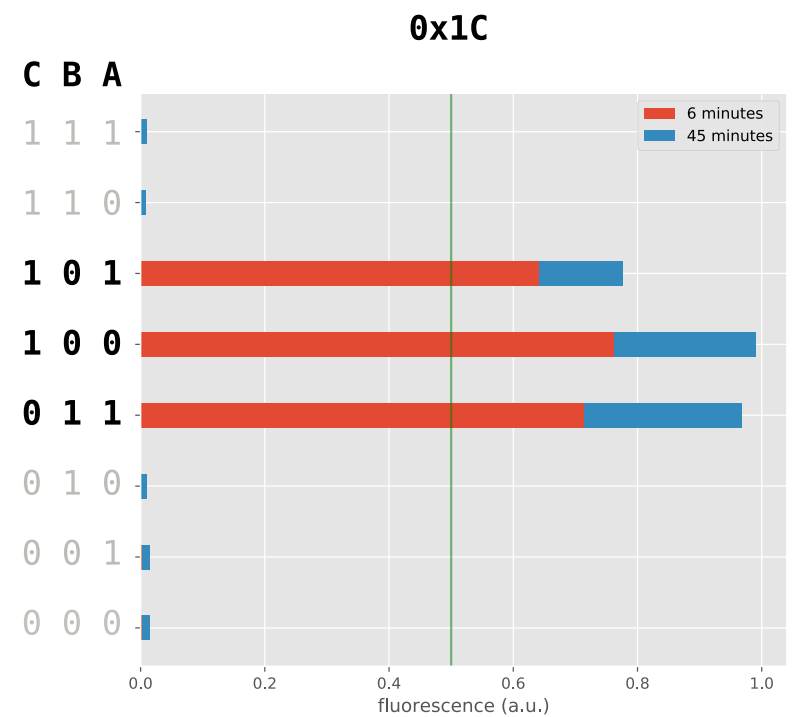
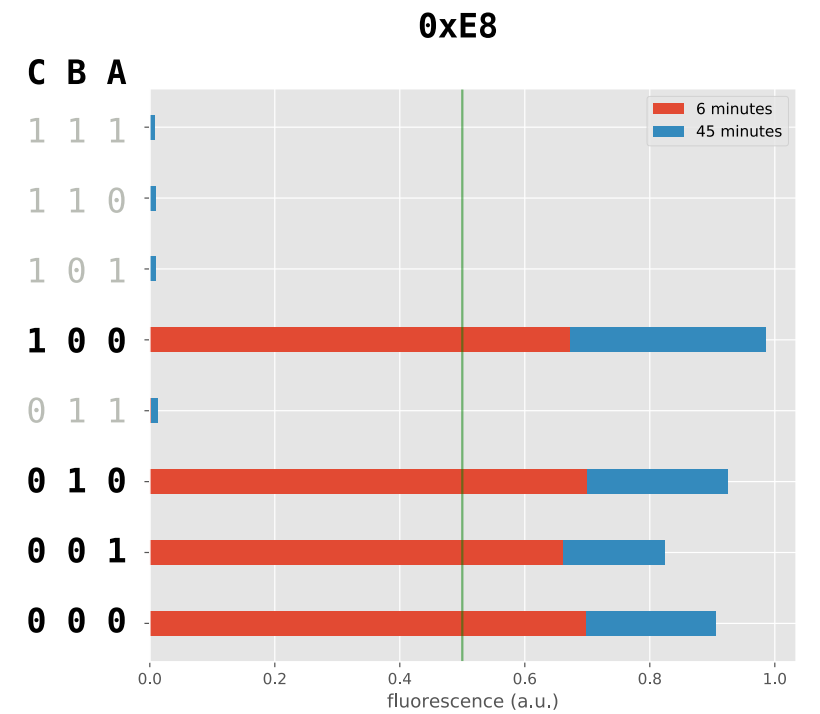
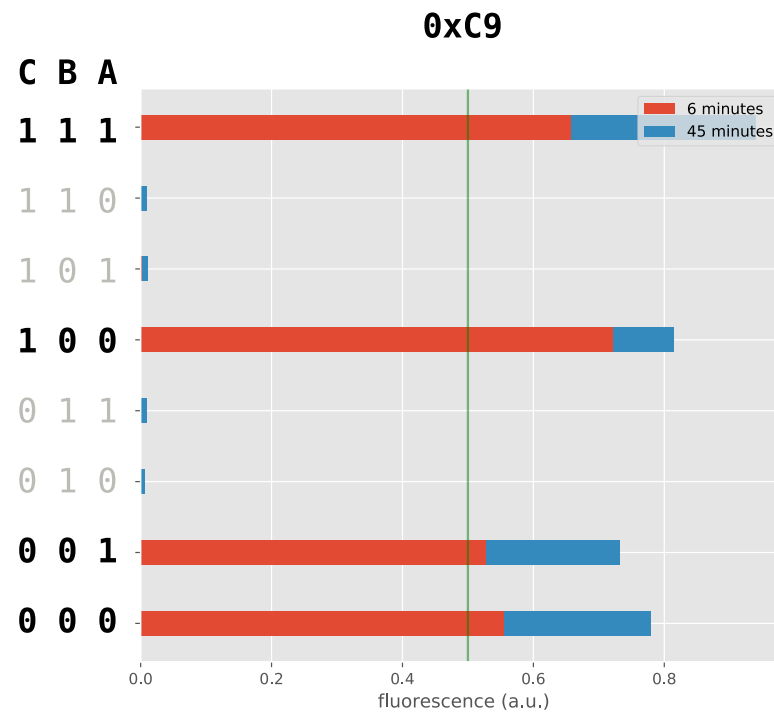
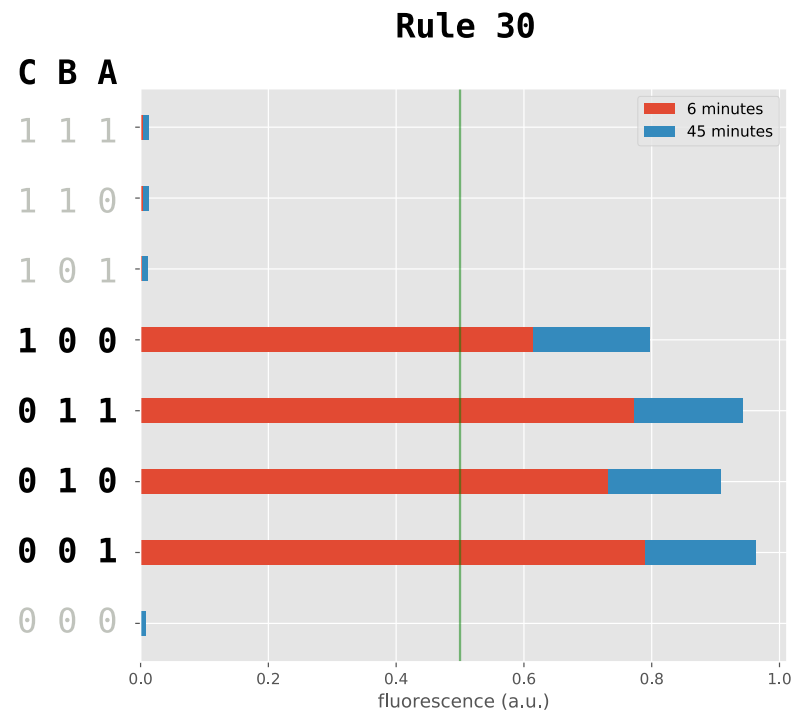
Acoustic
Liquid
Handler

Breadboard compiler
produces
a mixing protocol

All 8 input combinations for 6 circuits



First measurement 6 minutes after mixing start time

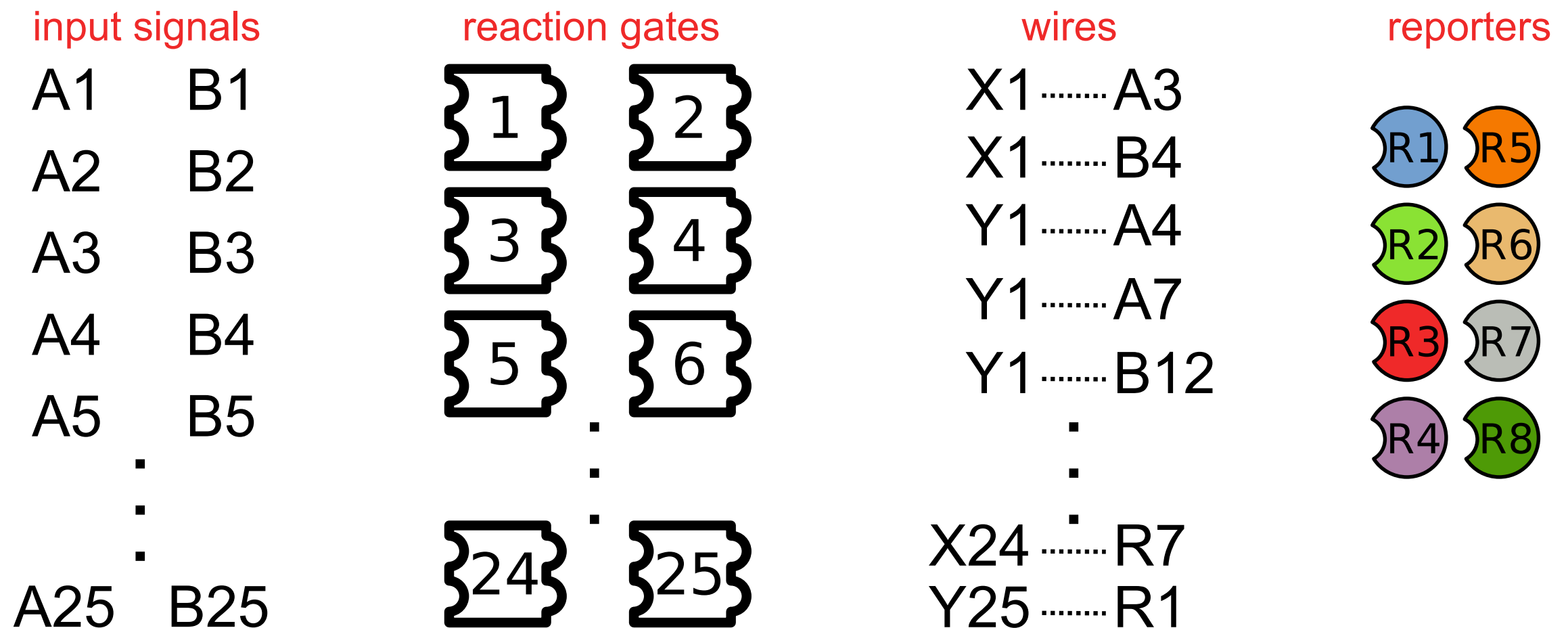


Molecular Circuit Breadboard

Roadmap

Molecular Breadboard 2.0:

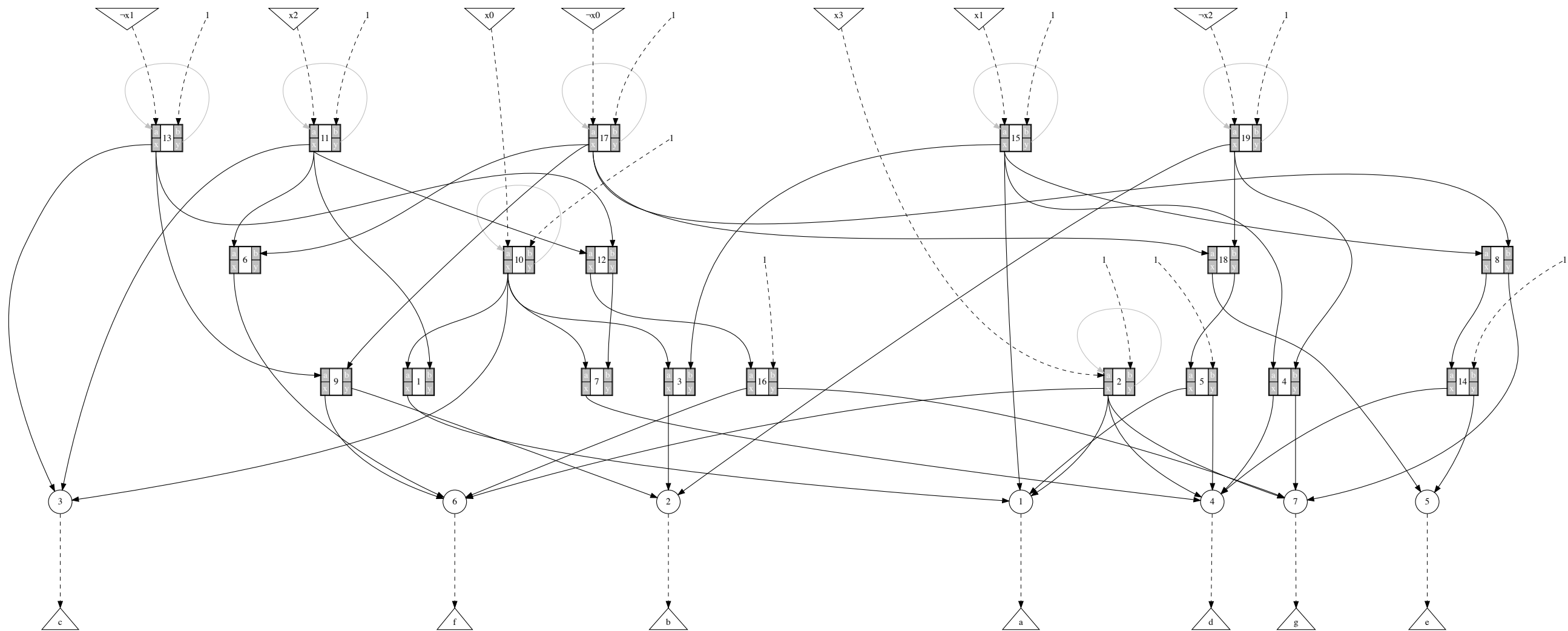
More circuits



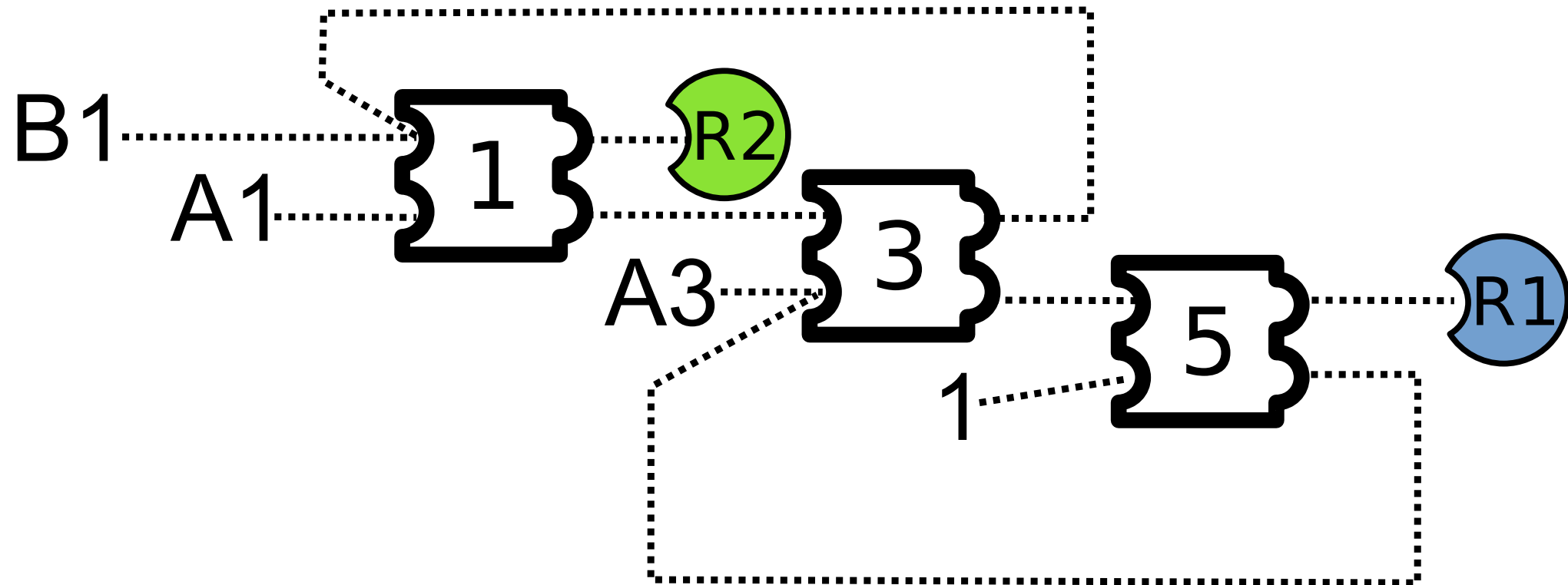
Breadboard 2.0 can
realize > 130 K circuits

Molecular Breadboard 2.0:

Larger circuits



Building circuits with feedback loops

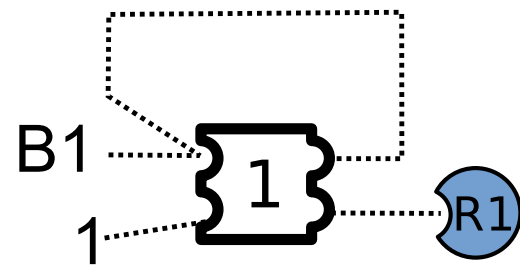


Chemical Reaction Networks

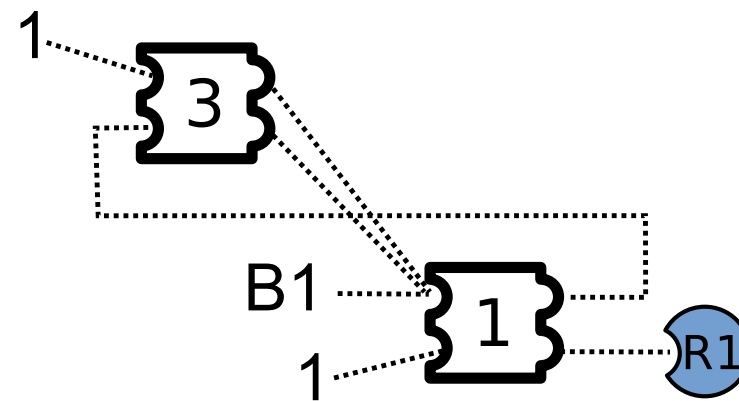
Asynchronous Sequential Logic Circuits

└─ **Finite state machines**

Providing input amplifiers & output signal restoration



Linear input amplifier



Exponential input amplifier



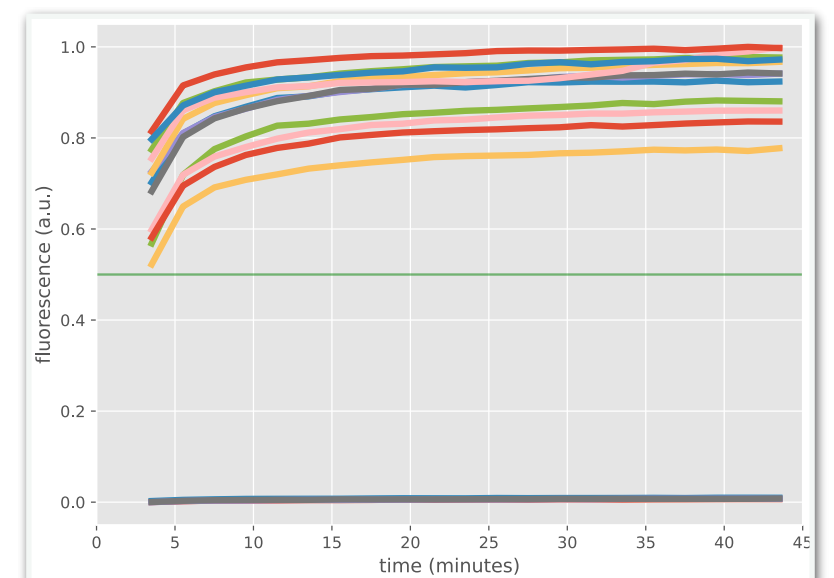
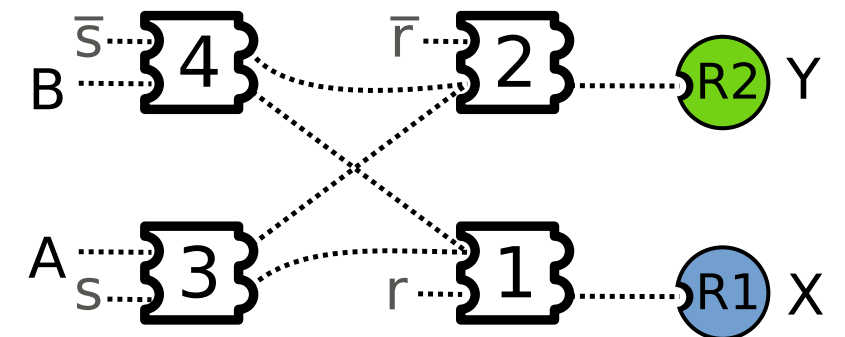
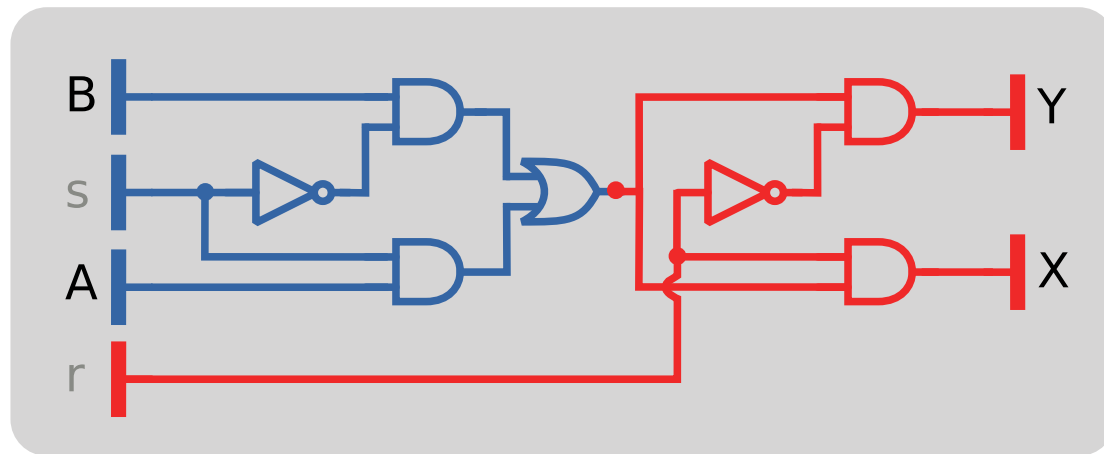
Output signal restoration

<http://DSDbreadboard.org>

Increased speed

Robustness to error

Automation



Related talks & posters @ DNA 24

Dominic Scalise, Nisita Dutta and [Rebecca Schulman](#)
DNA strand-displacement buffers

Si-Ping Han, Lisa Scherer, Matt Gethers, Marwa Ben Hadj Salah, Rebecca Mancusi, Sahil Sagar, Robin Hu, Julia Derogatis, Ya-Huei Kuo, Guido Marcucci, John Rossi and William A. Goddard III
Development and optimization of strand displacement based conditional small interfering RNAs for operation inside mammalian cells

[Eyal Nir](#), Yaron Berger and Miran Liber
Computer Controlled DNA Bipedal Walker that Perform Several Steps a Minute

Abhinav Singh and [Manoj Gopalkrishnan](#)
EM Algorithm with DNA Molecules

Wooli Bae, [Thomas Ouldridge](#) and [Guy-Bart Stan](#)
Autonomous generation of multi-stranded RNA complexes for synthetic molecular circuits

Yan Shan Ang and Lin-Yue Lanry Yung
Design of Split Proximity Circuit as a Plug-and-Play Translator for Discriminating Single Nucleotide Mutation

Yan Shan Ang and Lin-Yue Lanry Yung
Dynamically Elongated Association Toehold for Tuning Circuit Kinetics and Thermodynamics

Patrick Irmisch and Ralf Seidel
Modelling DNA-strand displacement reactions in the presence of base-pair mismatches

Boya Wang and [David Soloveichik](#)
Experimentally characterizing the design space of strand displacement translators with toehold-size clamps

Allison Tai and Anne Condon
Error-free stable computation with stack-supplemented chemical reaction networks

Kevin Cherry, Gokul Gowri and Lulu Qian
DNA-based neural networks that learn from their molecular environment

Robert F. Johnson and [Erik Winfree](#)
Using Bisimulation for Verification of Polymer Reaction Networks

Acknowledgments

- **Winfree lab** (Caltech)
- **Soloveichik lab** (University of Texas at Austin)
- **Qian lab** (Caltech)
- **Murray lab** (Caltech)
- Thanks to DNA 24 organizers for the invitation



Tools discussed in tutorial

ABC: logic synthesis and verification

<https://people.eecs.berkeley.edu/~alanmi/abc>

VisualDSD

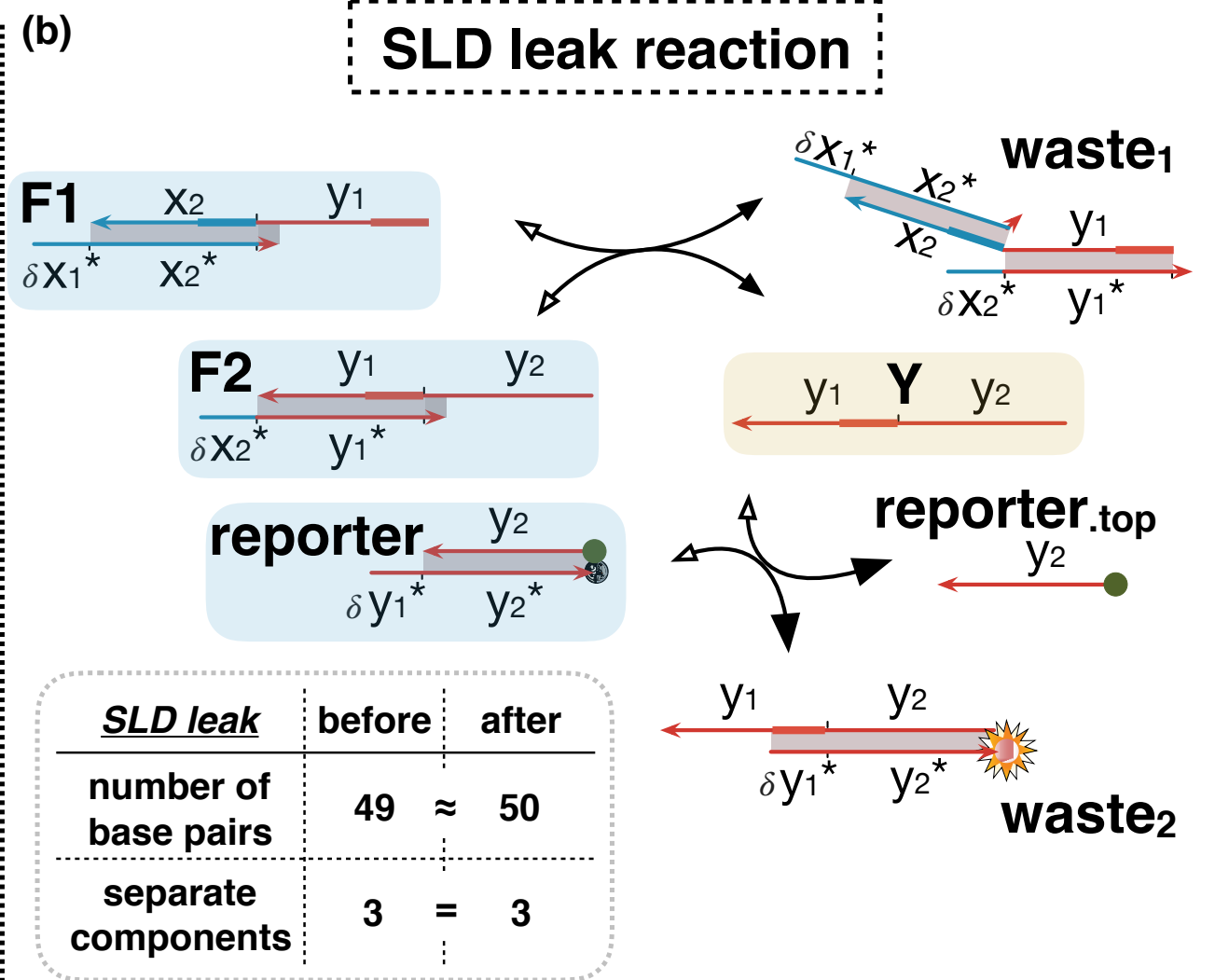
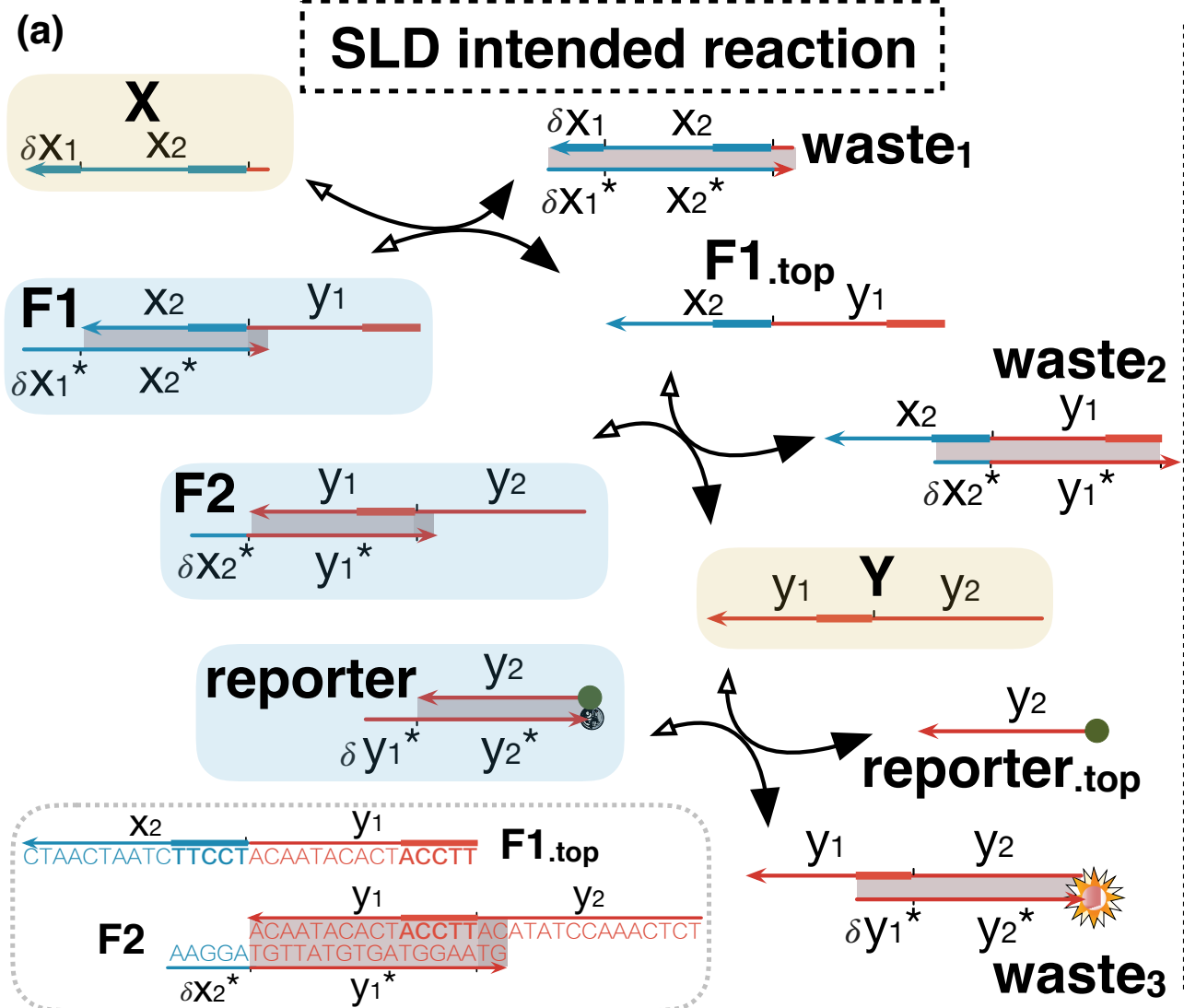
<https://lepton.research.microsoft.com/webdna>

Nuskell compiler framework

<https://github.com/DNA-and-Natural-Algorithms-Group>

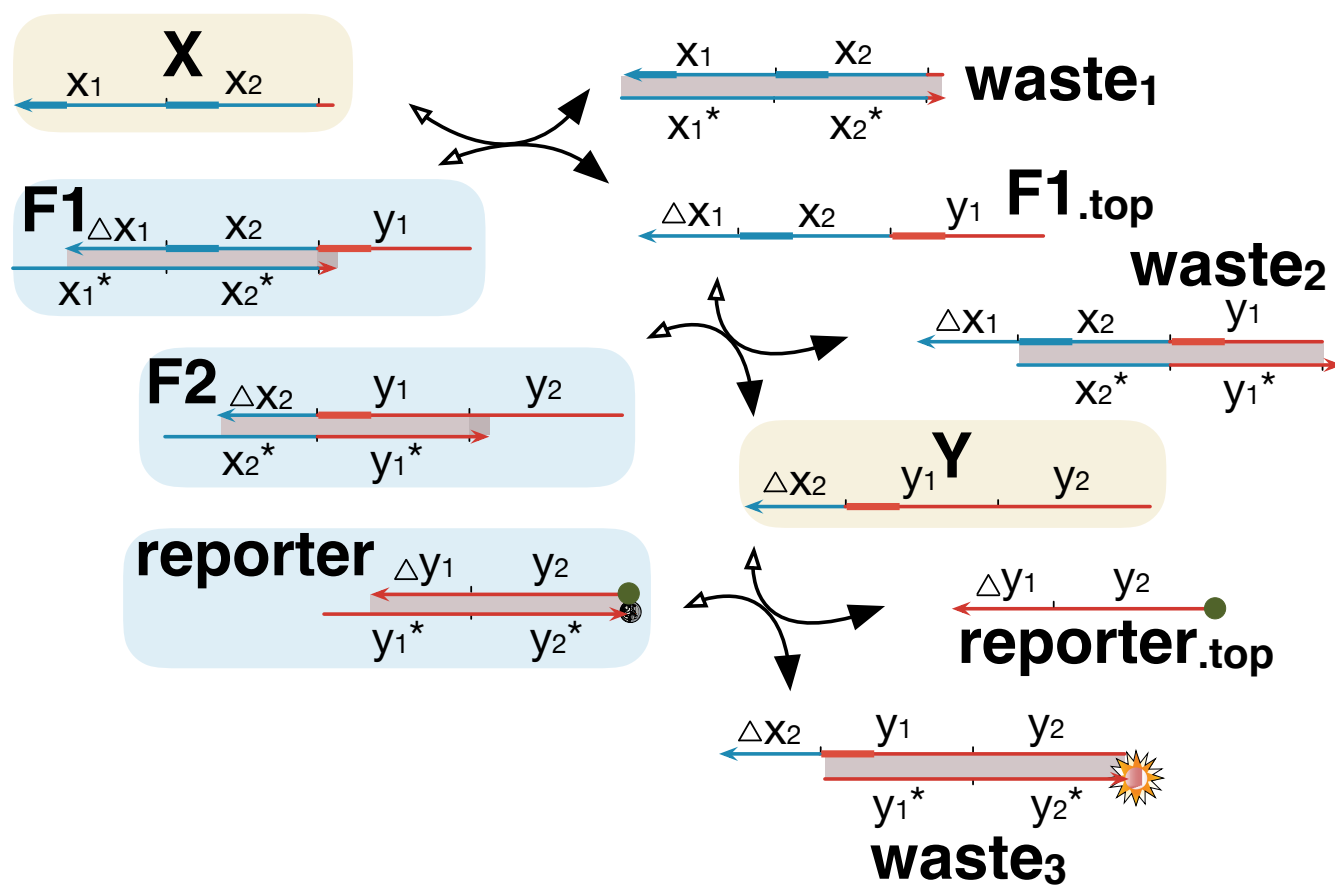
DSD breadboard

<http://dsdbreadboard.org> *(online later this year)*



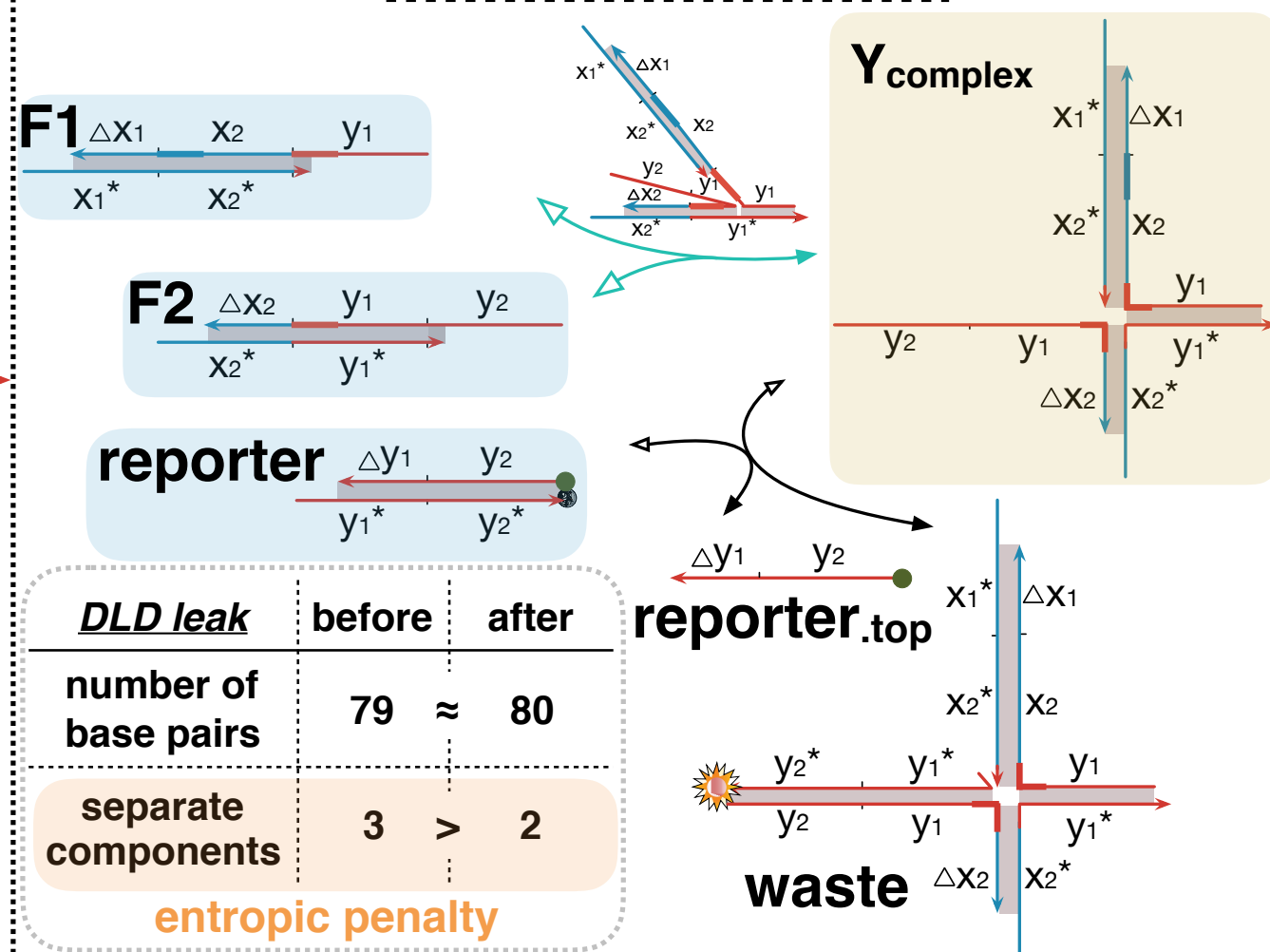
(a)

DLD intended reaction

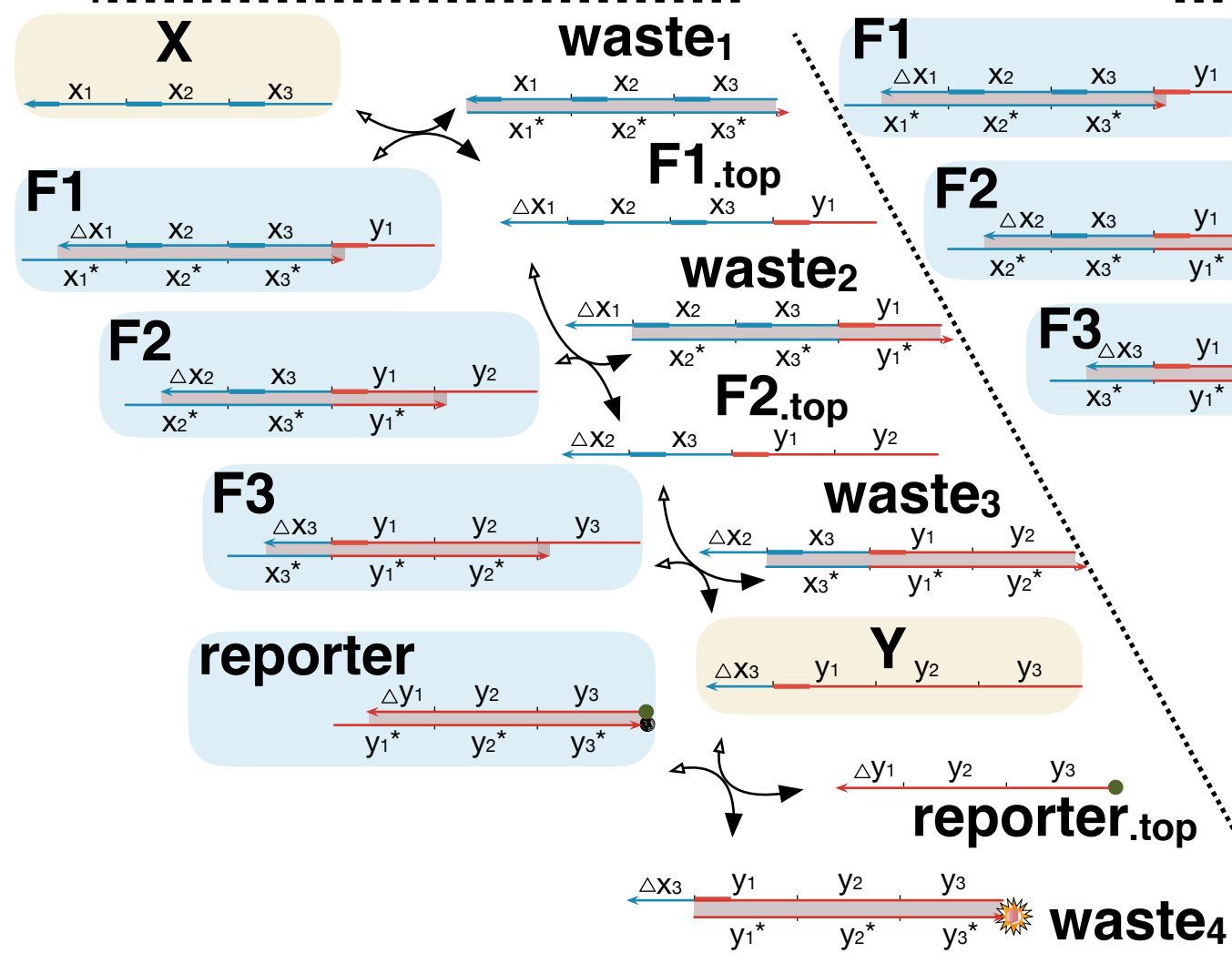


(b)

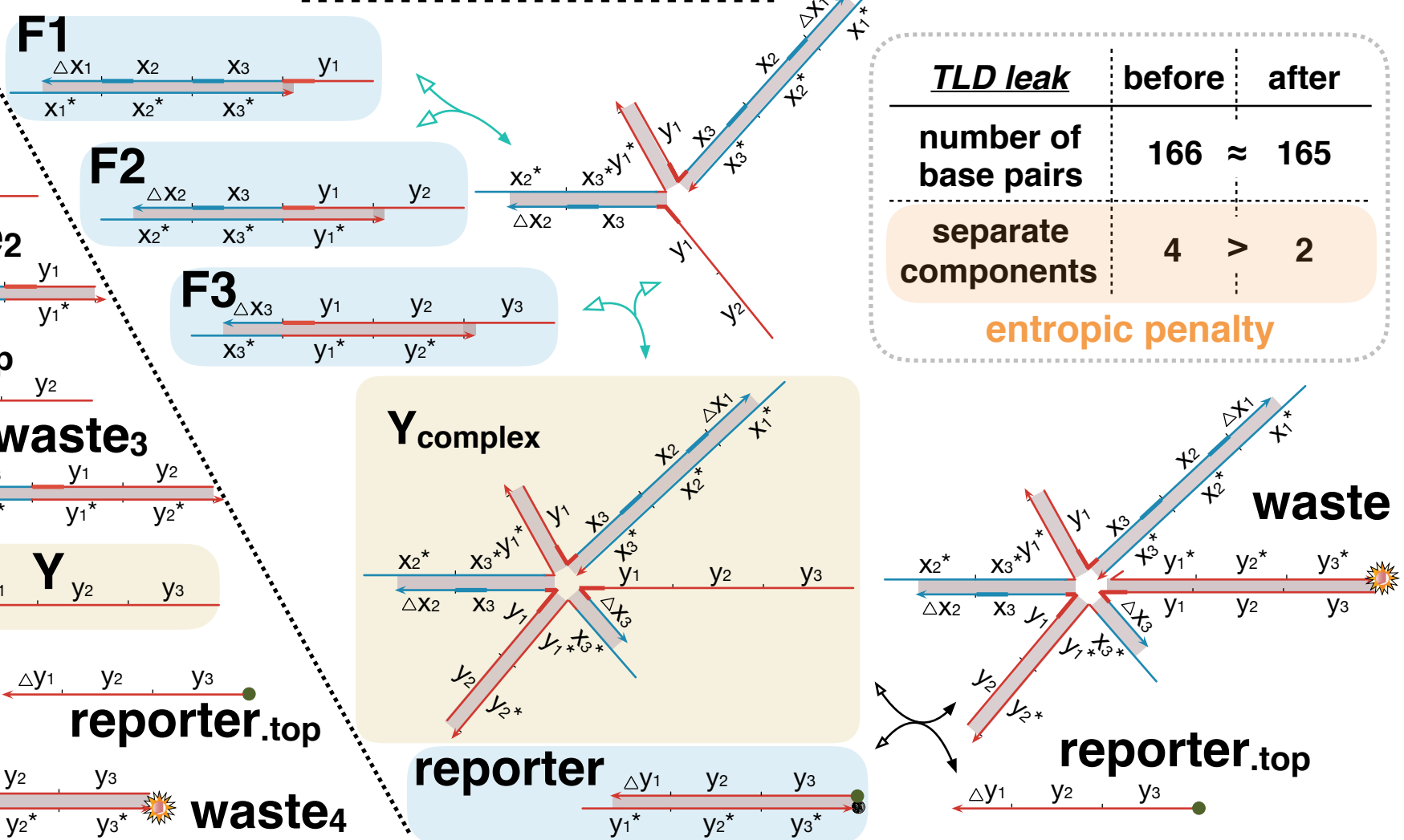
DLD leak reaction



(a) **TLD intended reaction**



(b) **TLD leak reaction**



<i>TLD leak</i>	before	after
number of base pairs	166	≈ 165
separate components	4	> 2

entropic penalty