21 molecular algorithms using reprogrammable DNA self-assembly

Damien Woods
David Doty, Cameron Myhrvold, Joy Hui
Felix Zhou, Peng Yin, Erik Winfree

Maynooth University
National University of Ireland Maynooth

Hamilton Institute

Caltech

Inria

UC Davis

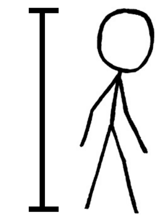
Harvard

DNA26 presentation

Building stuff

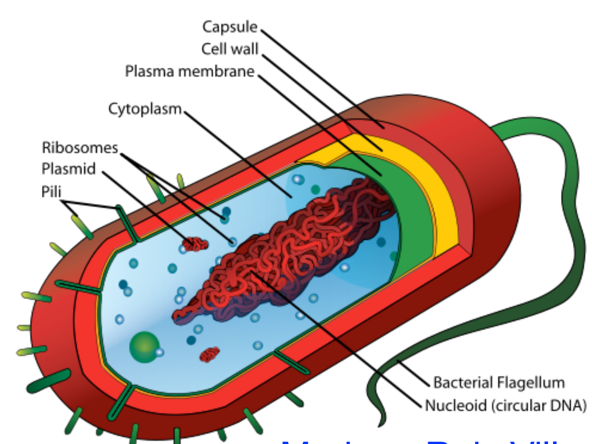
Ljubljana Marshes Wheel. 5k years old

Newgrange, Ireland. 5.2k years old

- **Building stuff by hand**: use tools! Great for scale of $10^{+/-2}$ x 

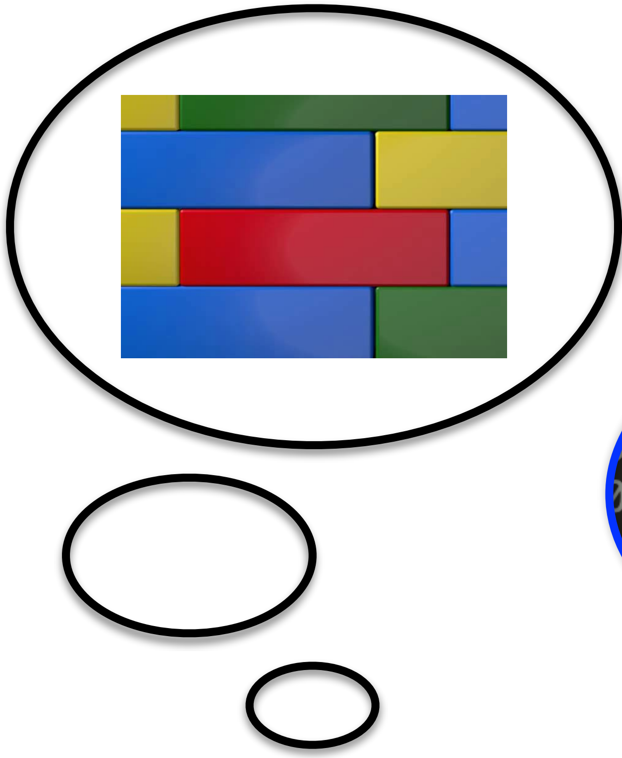
- **Algorithms and tools that build stuff**: specify target object with a computer program that controls the manufacturing process

- Put the algorithm inside: program **stuff to build itself**!

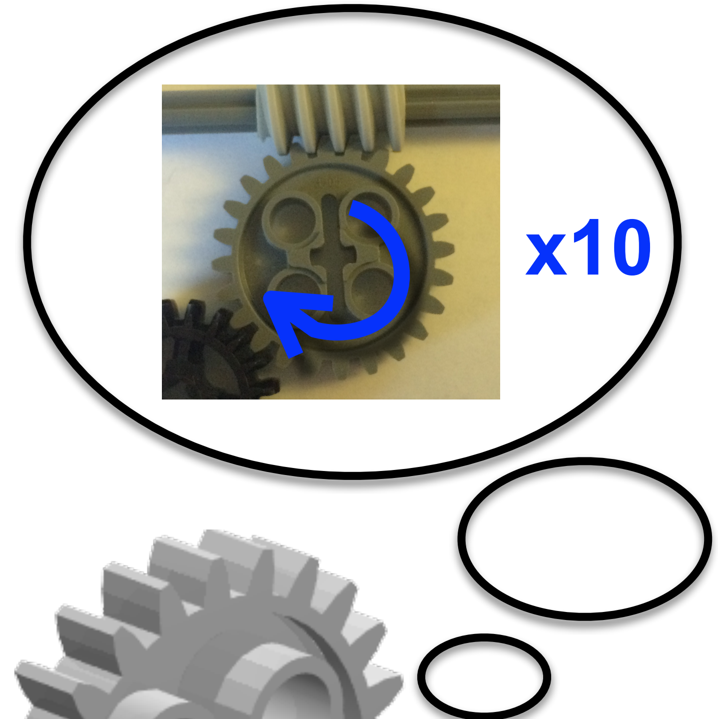


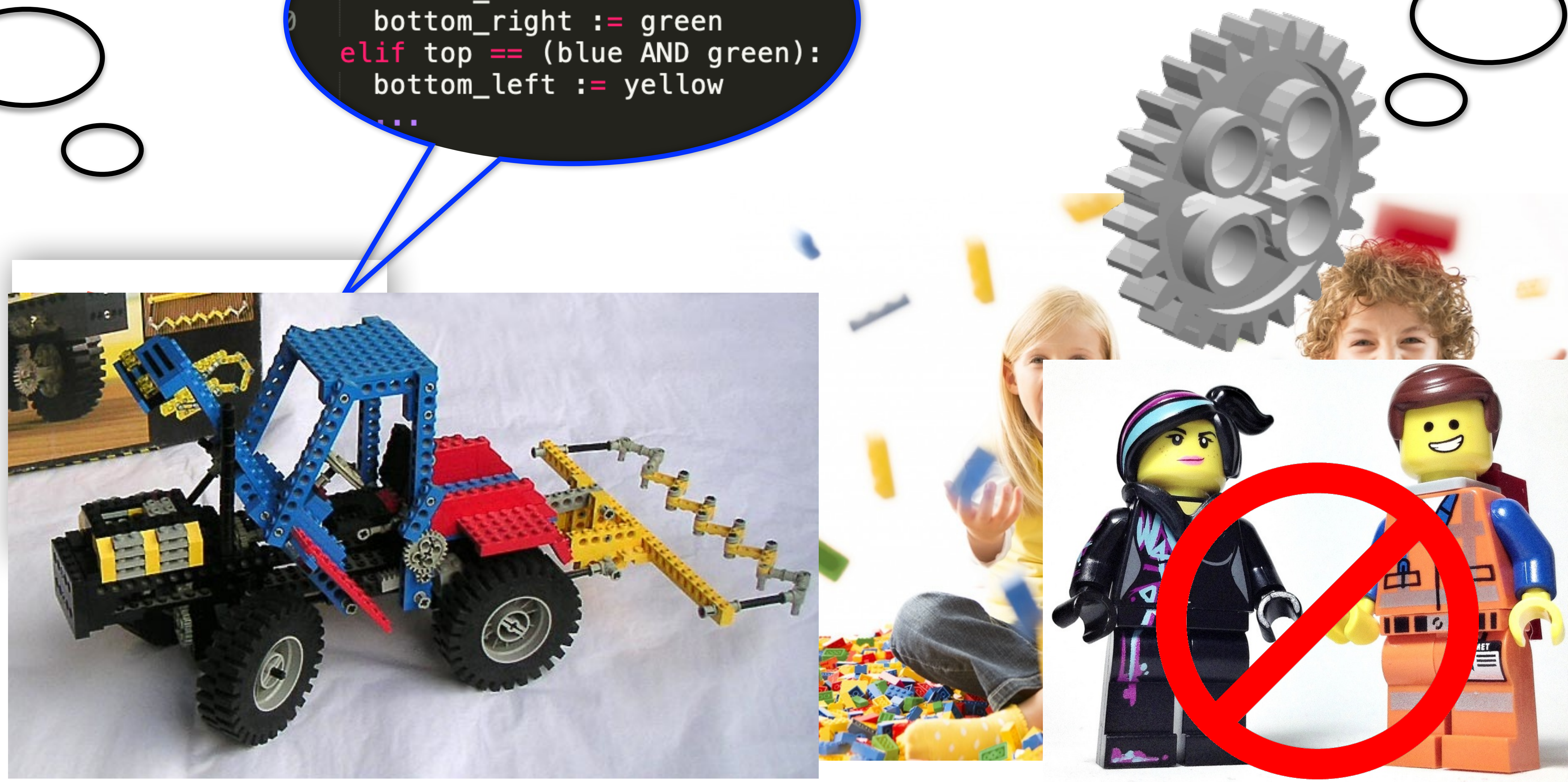
Mariana Ruiz Villarreal

Stuff that builds itself



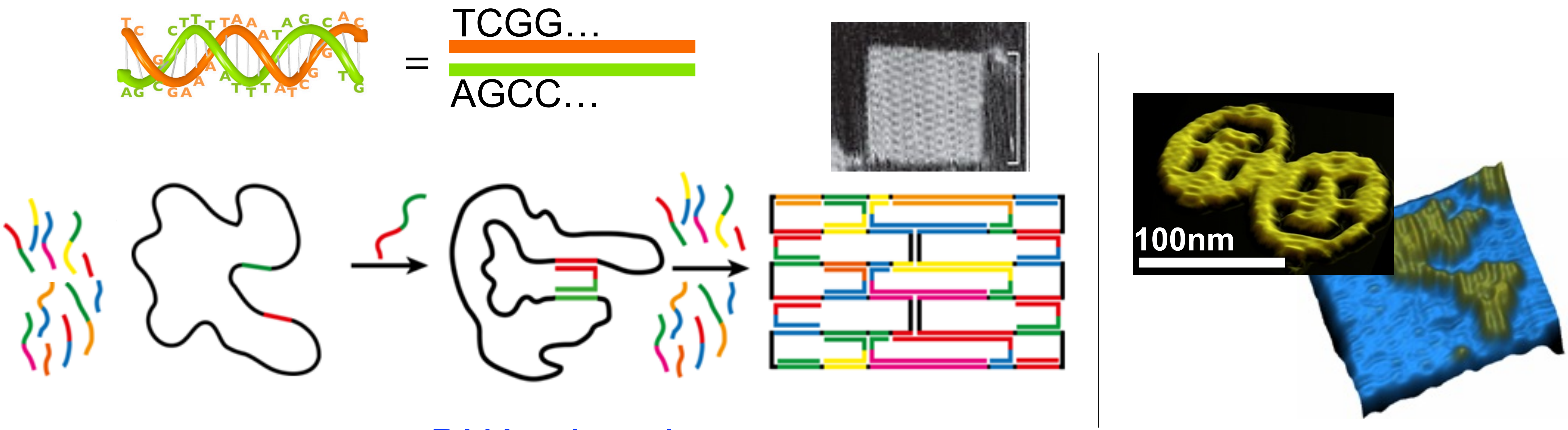
```
if top == (blue AND yellow):
  bottom_left := blue
  bottom_right := green
elif top == (blue AND green):
  bottom_left := yellow
...
```





- Today you'll hear about self-assembling molecules that compute as they build themselves

Background: DNA nanostructures



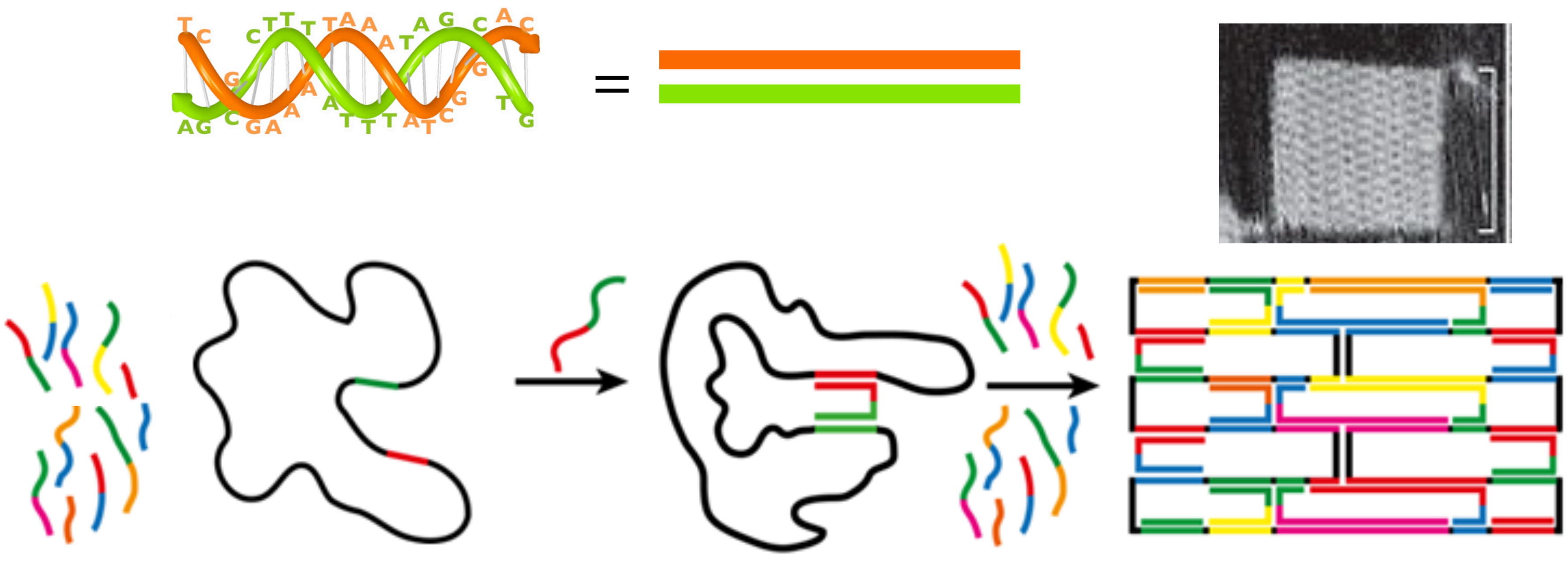
DNA origami

Rothemund. 2006 Nature

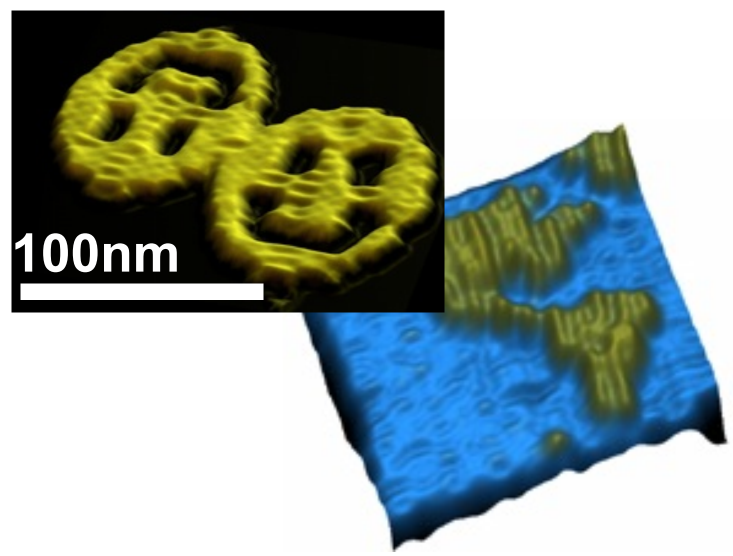
Example DNA nanostructure: DNA origami

Movie by Shawn Douglas

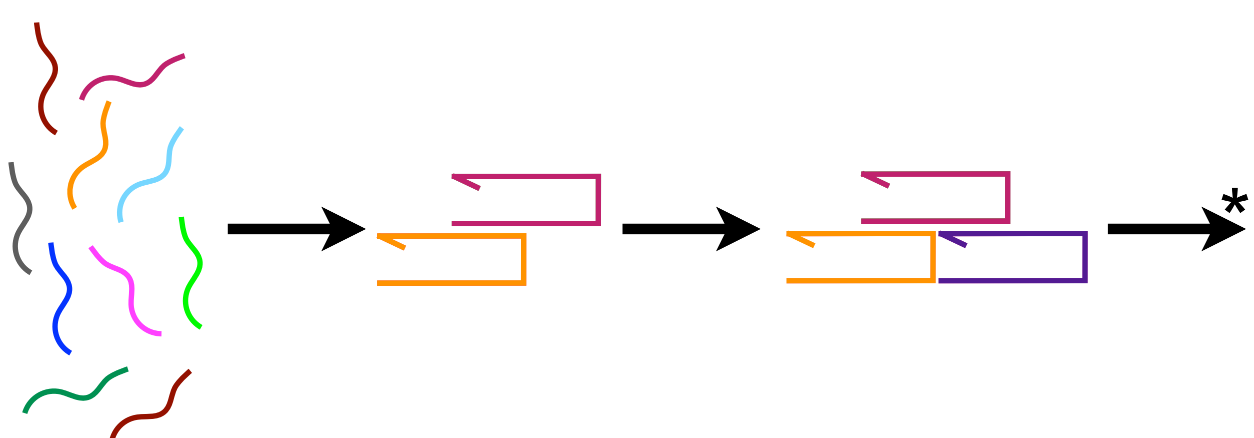
Background: DNA nanostructures

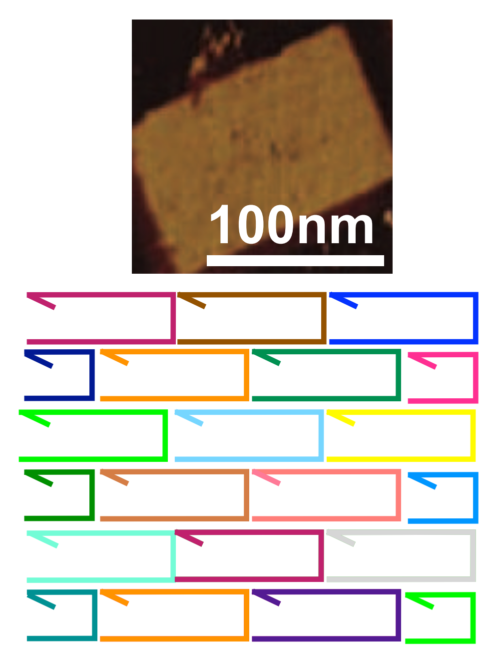


DNA origami



Rothemund. 2006 Nature



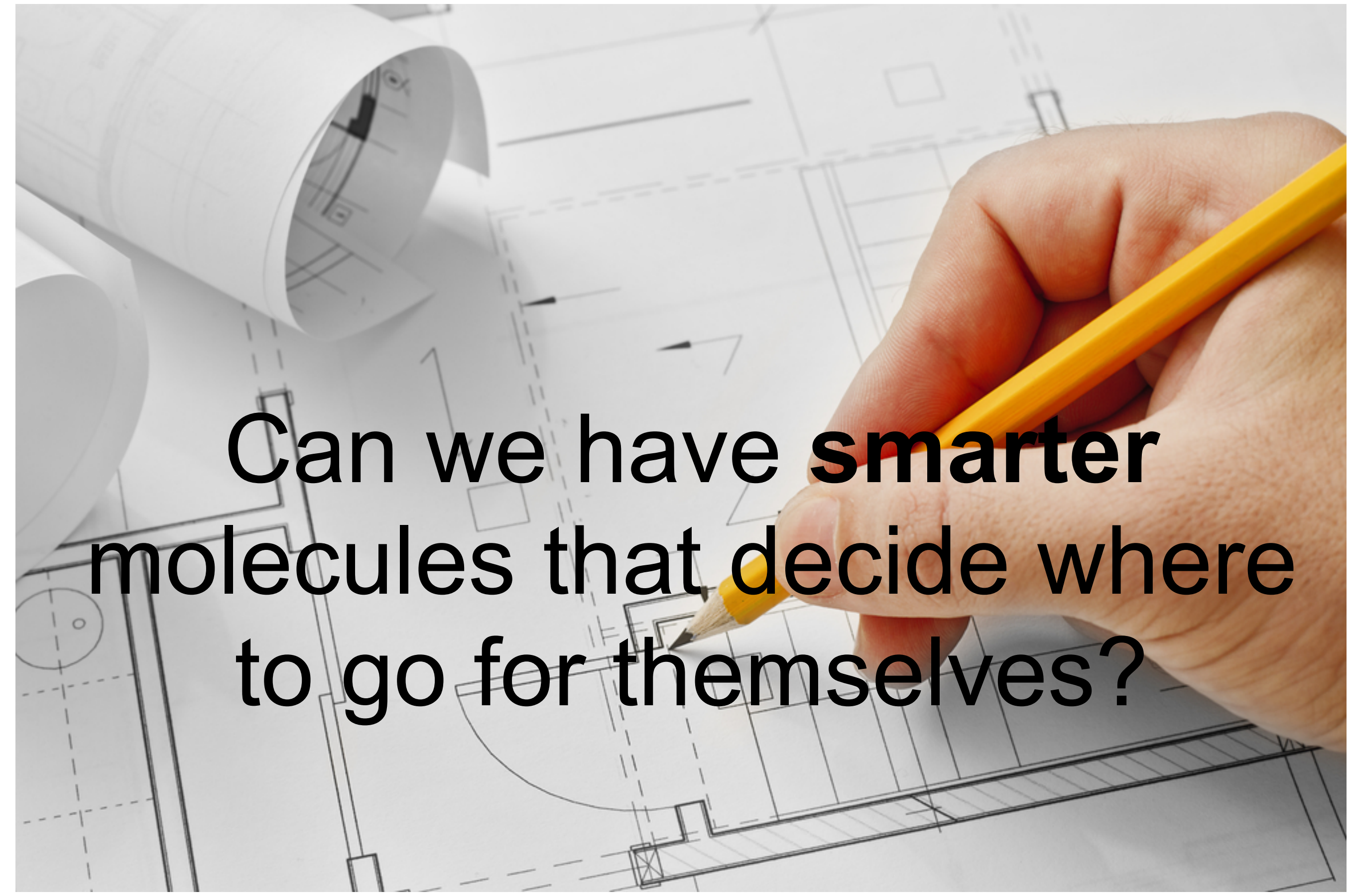


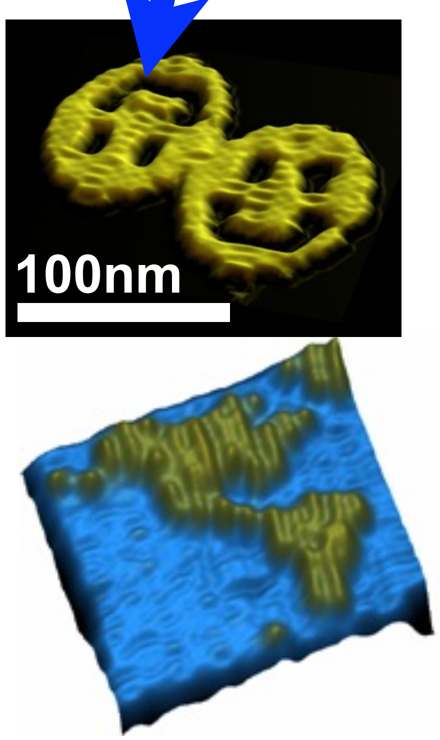
DNA single-stranded tiles

Wei, Dai, Yin. 2012 Nature

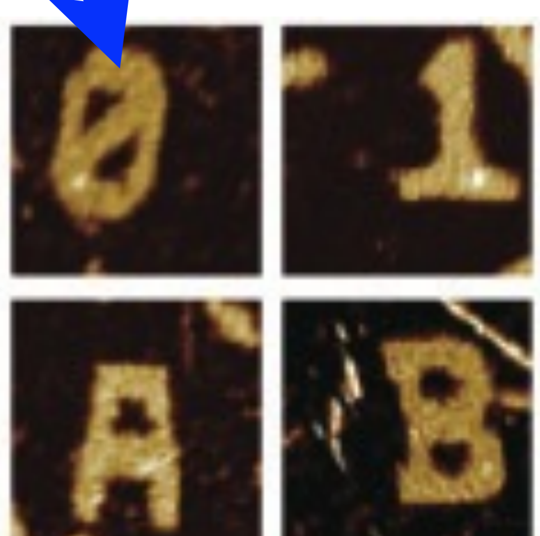
Nanostructure design and self-assembly

Typically, we tell the molecules **exactly** where to go

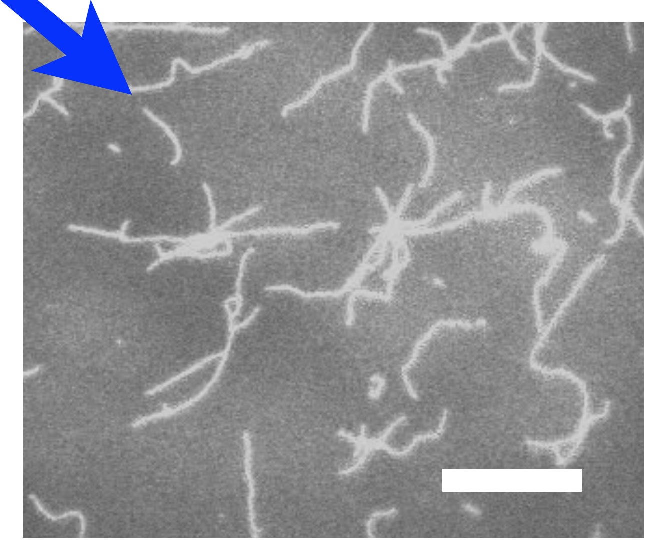


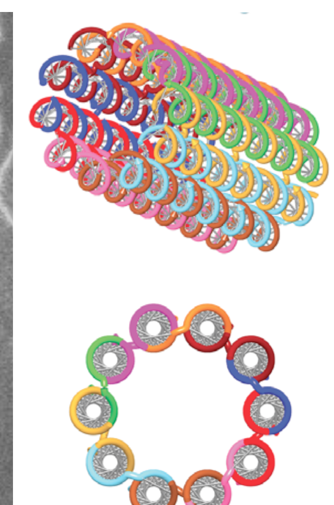


Rothemund
2006 Nature



Wei, Dai, Yin. 2012
Nature



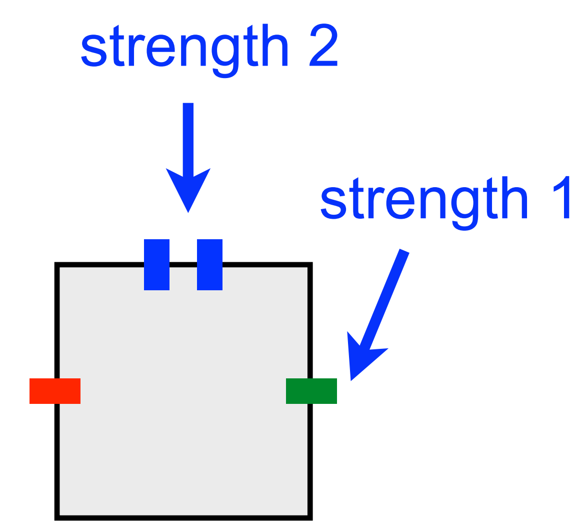


Yin et al 2008 Science

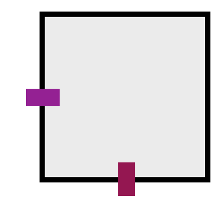
Abstract tile assembly model

An asynchronous cellular automaton model capturing dynamics of molecular binding

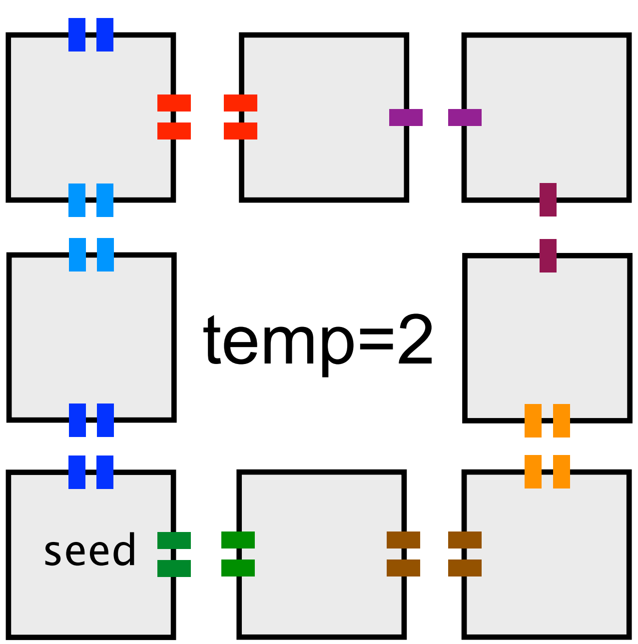
- Square **tiles**
  - finite set of tile types, unlimited supply of each type, non-rotatable
- Each side has a **glue** (colour) and **strength** (0,1,2,3,…)
- System has a **temperature** (e.g. 2)

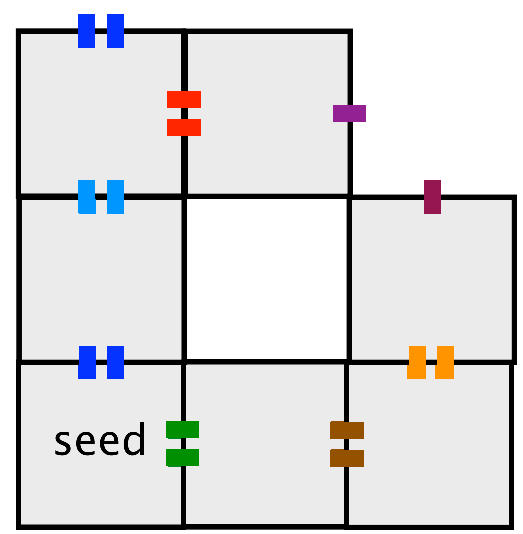


- **Simple local binding rule**: A tile sticks to an assembly if enough of its glues match so that the sum of the strengths of the matching glues is at least the temperature



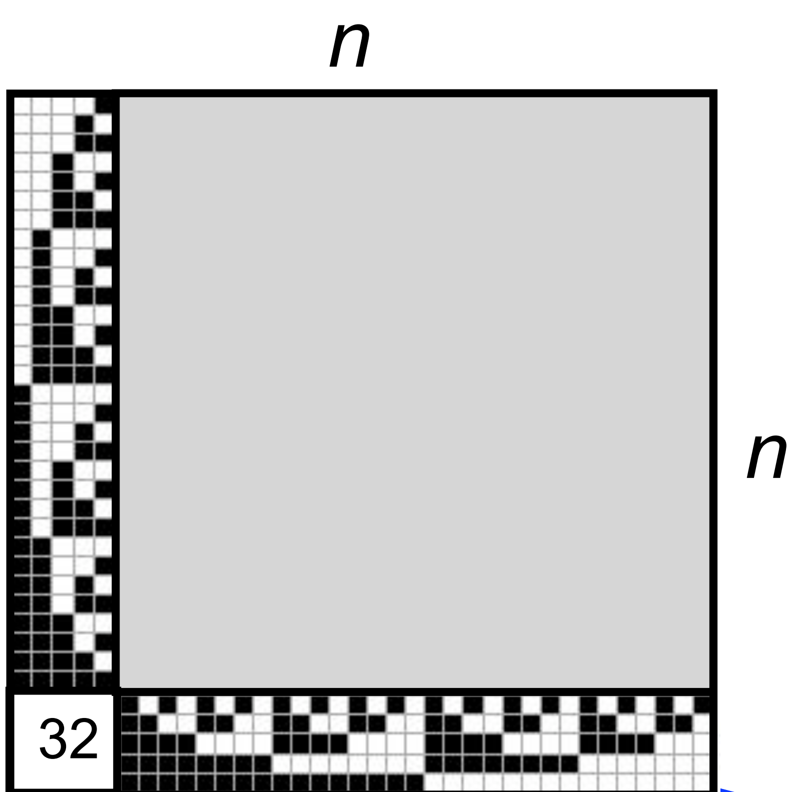
Model by Winfree, 1998





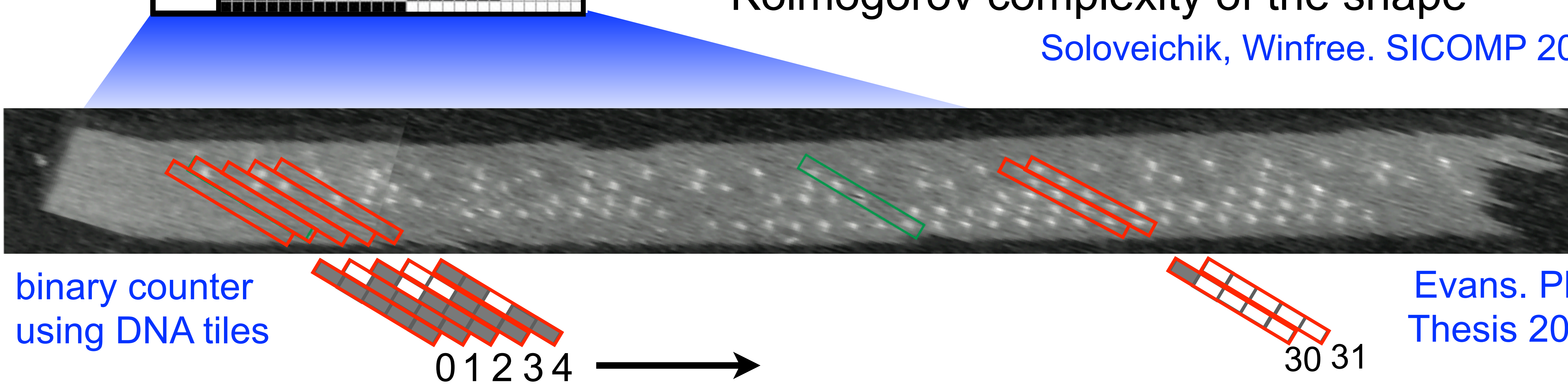
We can make these tiles out of DNA!
Small size, requires us to program in a bottom-up way

Algorithmic self-assembly: some previous work



- Turing universality
  Winfree, PhD Thesis. 1998

- Efficient assembly of simple shapes: *n* x *n* squares using Θ(log *n* / log log *n*) tile types
  Adleman, Cheng, Goel, Huang STOC 2001
  Rothemund, Winfree. STOC 2000
- Efficient assembly of scaled shapes using a number of tile types roughly equal to the Kolmogorov complexity of the shape
  Soloveichik, Winfree. SICOMP 2007

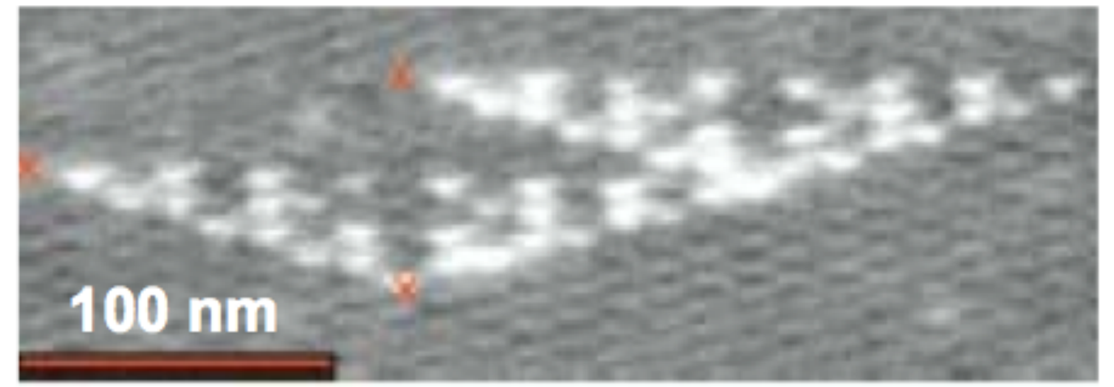


binary counter using DNA tiles

Evans. PhD Thesis 2014

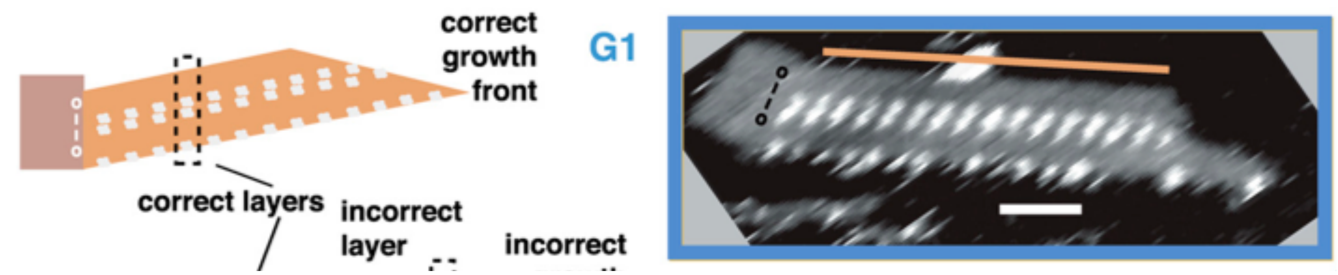
Algorithmic self-assembly experiments: previous work

Bit Copying. Barish et al.2009

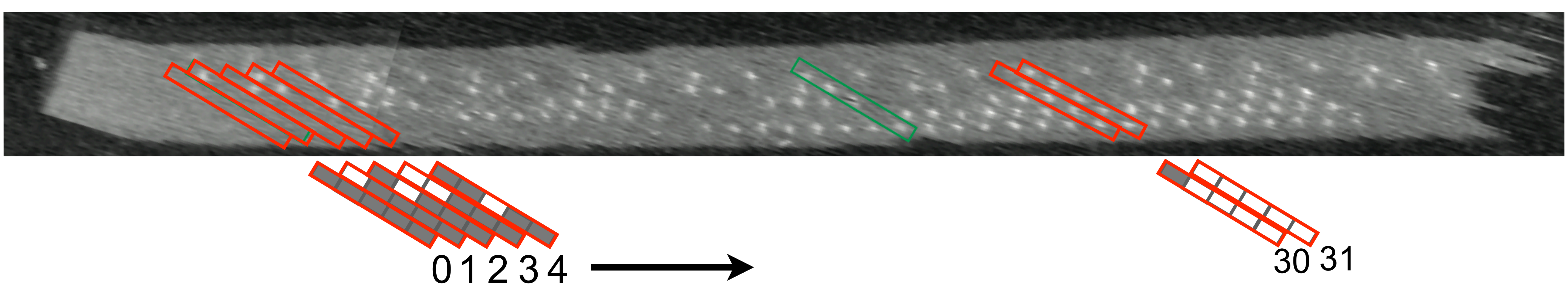


Sierpinski Triangles. Rothemund, Papadakis, Winfree. 2004

Counter. Barish et al. 2009



Copying & replication Schulman, Yurke, Winfree. PNAS. 2012



Evans. PhD. Thesis 2014. Winfree group.

Copying, Sierpinski, binary counting to 31, can we run more self-assembly algorithms?

Structure of talk

Copying, Sierpinski, binary counting to 31, can we run more self-assembly algorithms?

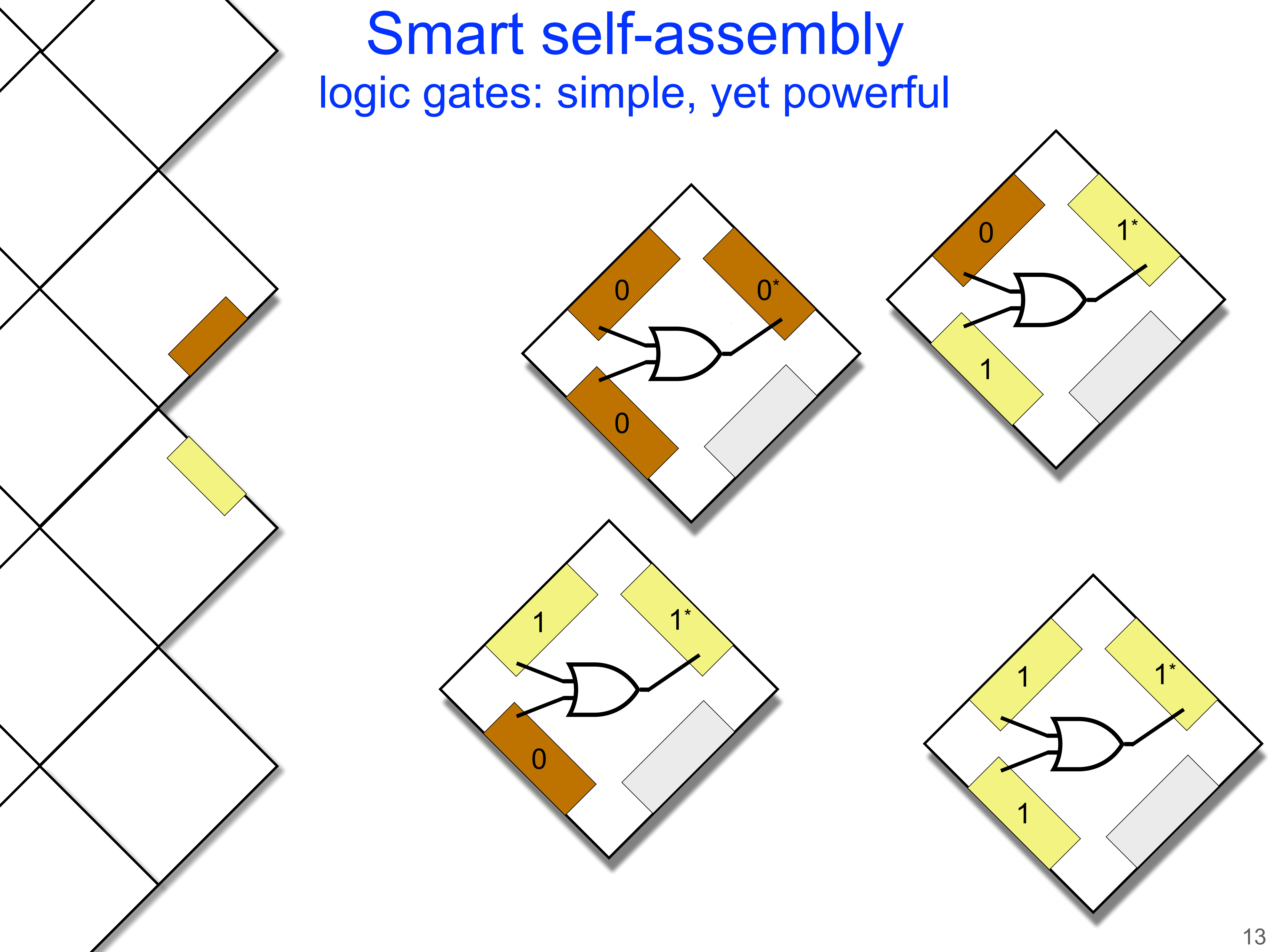
Theoretical circuit model

How it works: design and implementation

Experimental results

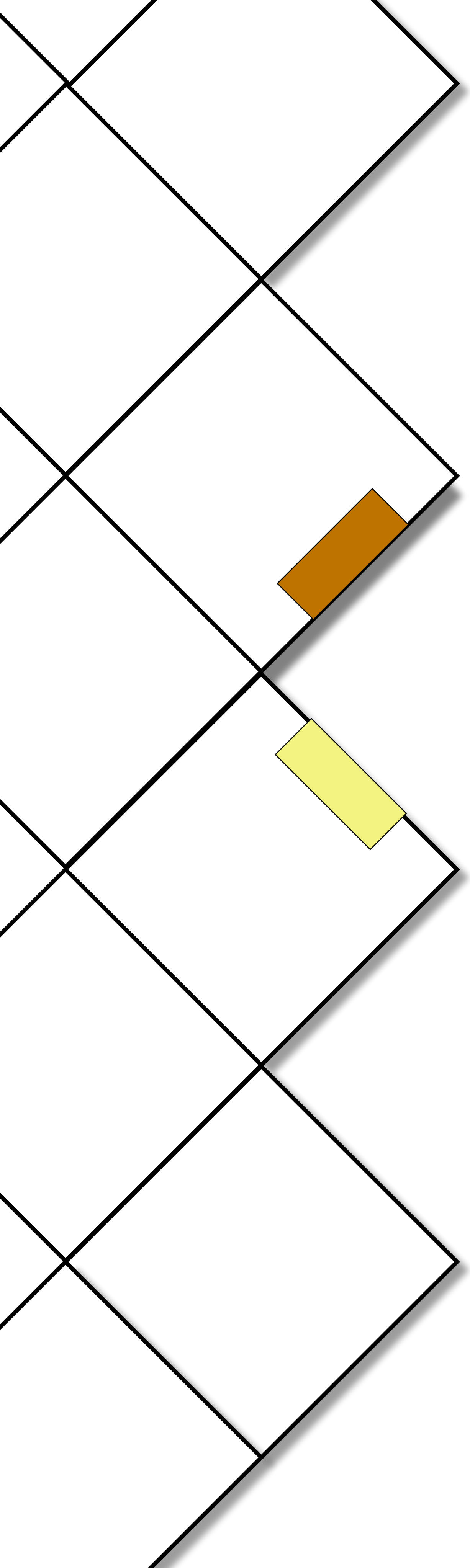
Smart self-assembly

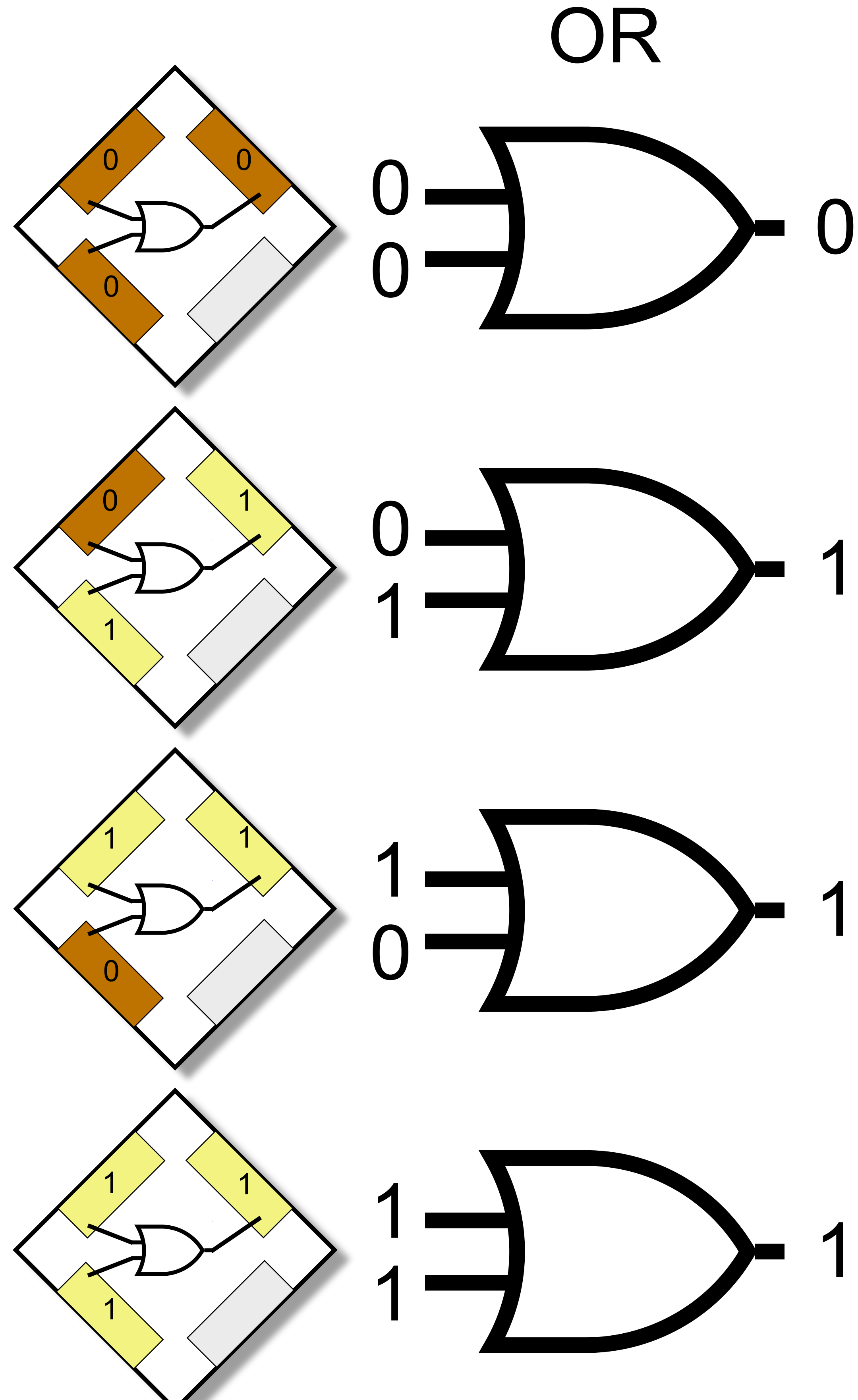
logic gates: simple, yet powerful



Smart self-assembly

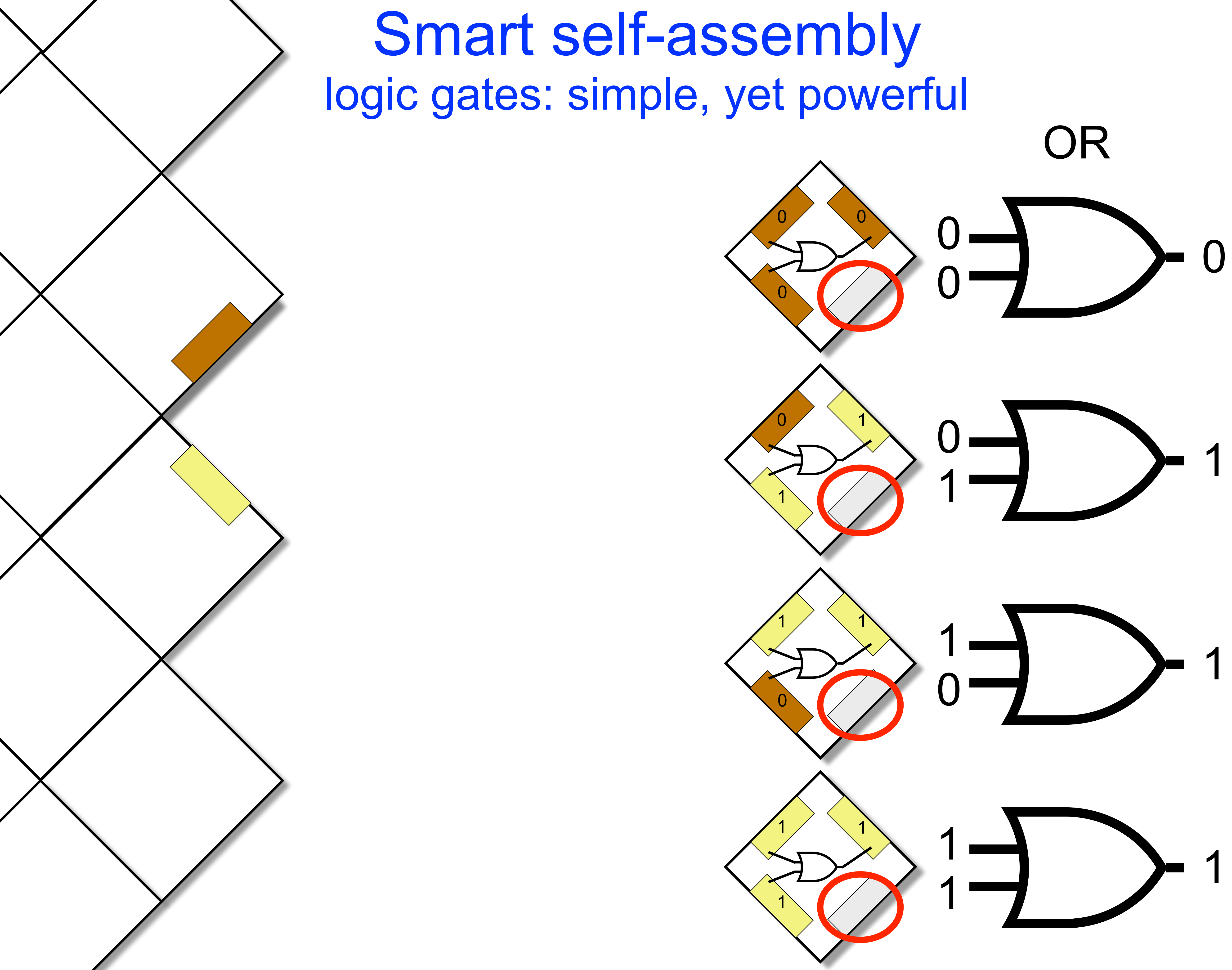
logic gates: simple, yet powerful





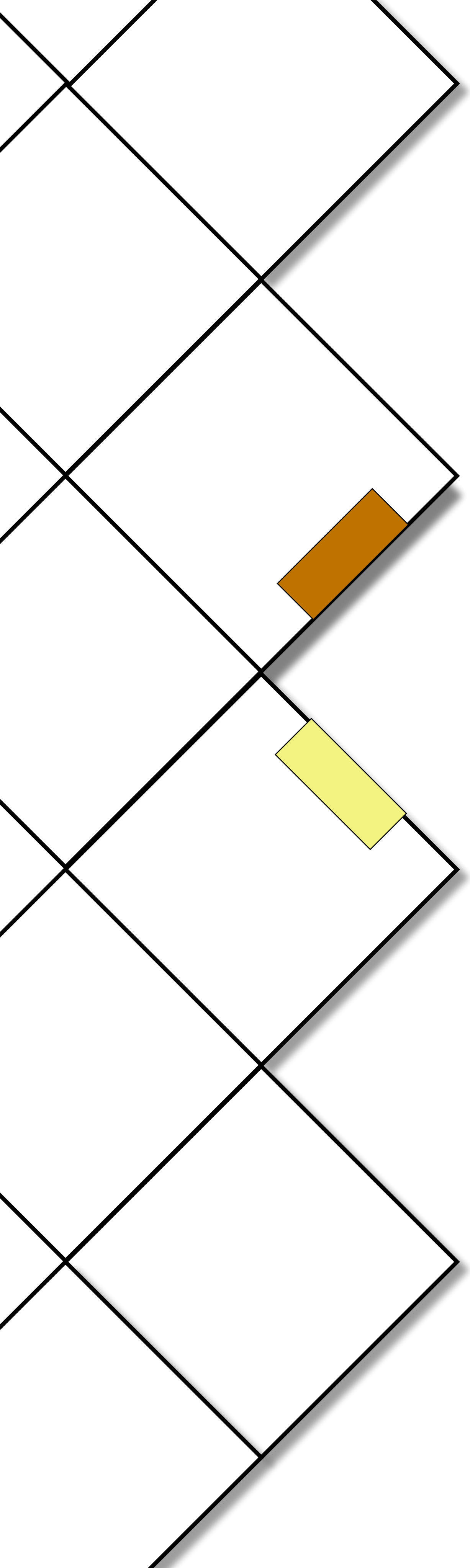
Smart self-assembly

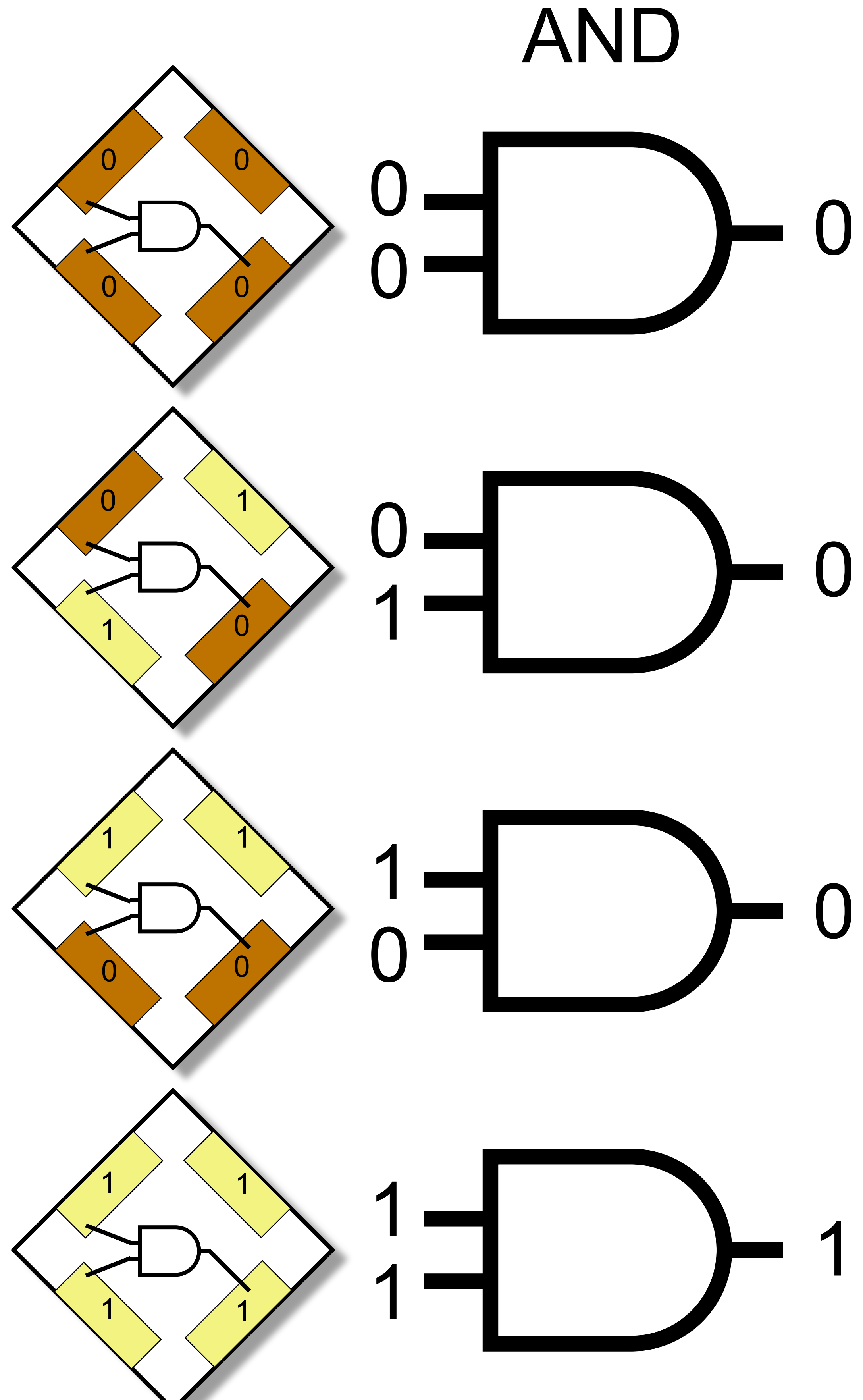
logic gates: simple, yet powerful



Smart self-assembly

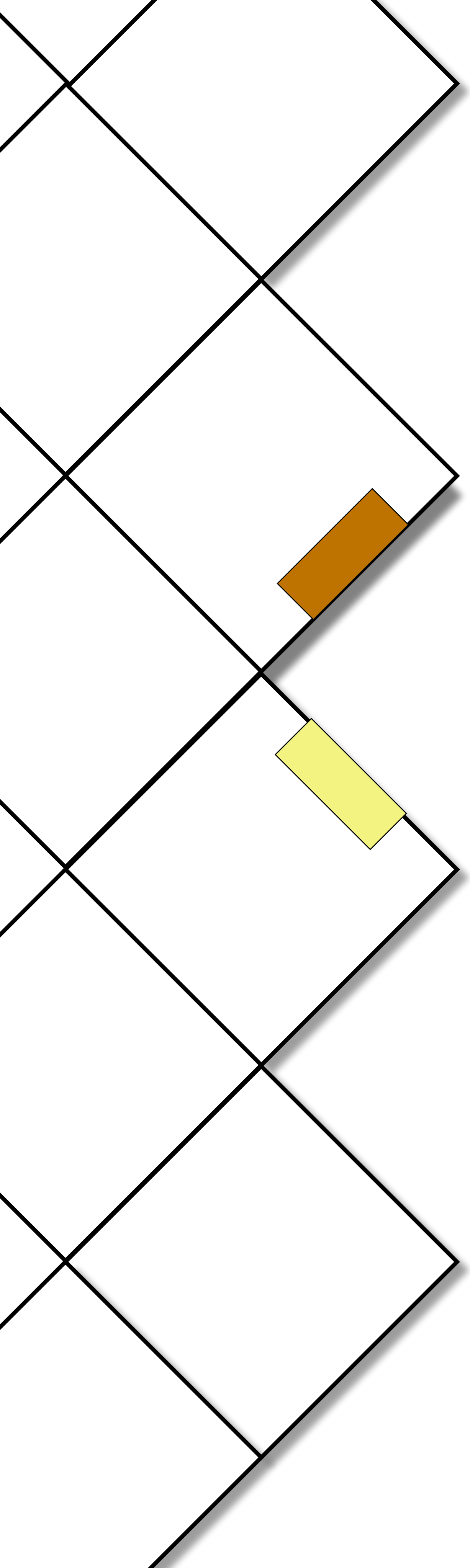
logic gates: simple, yet powerful

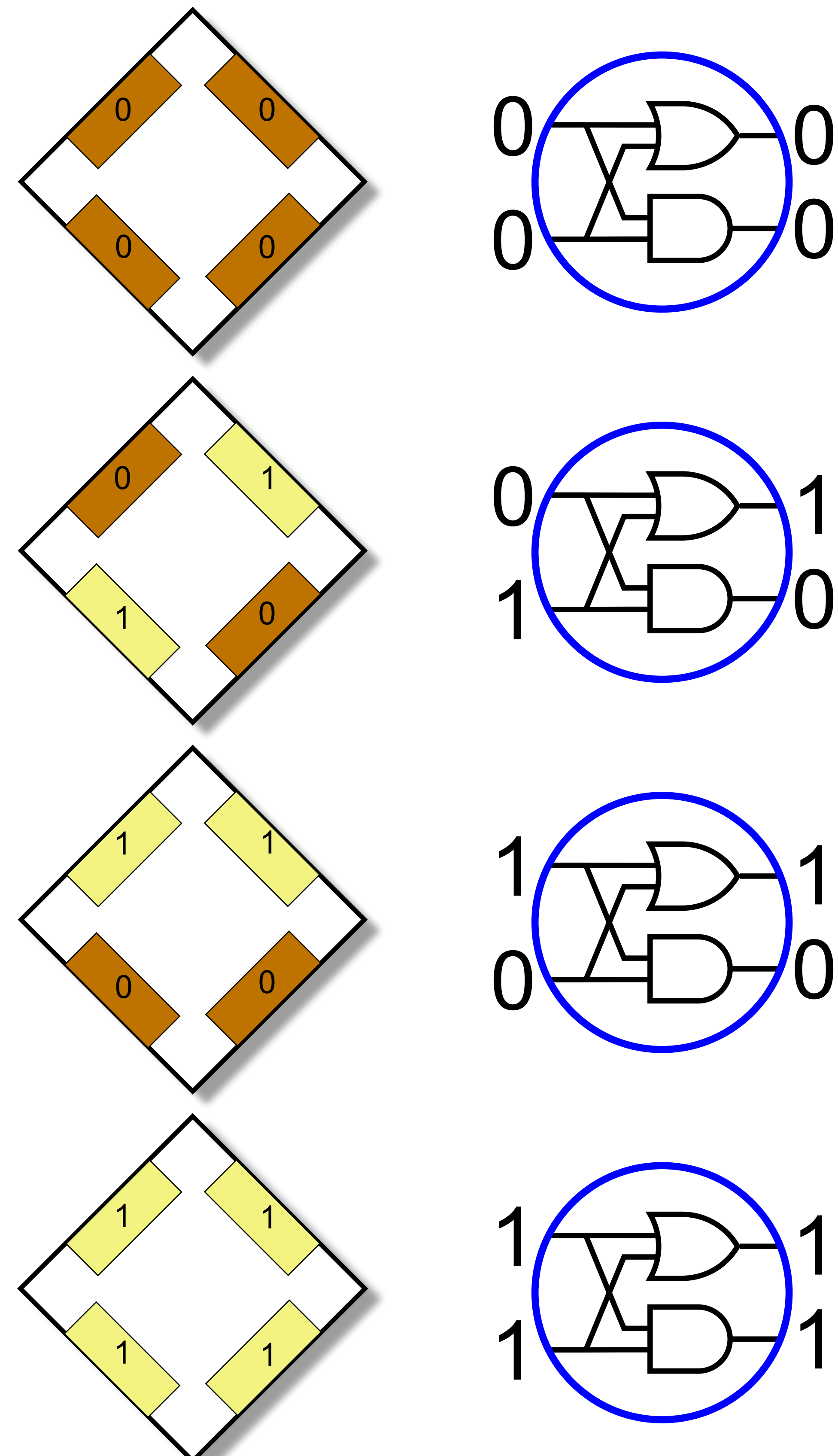




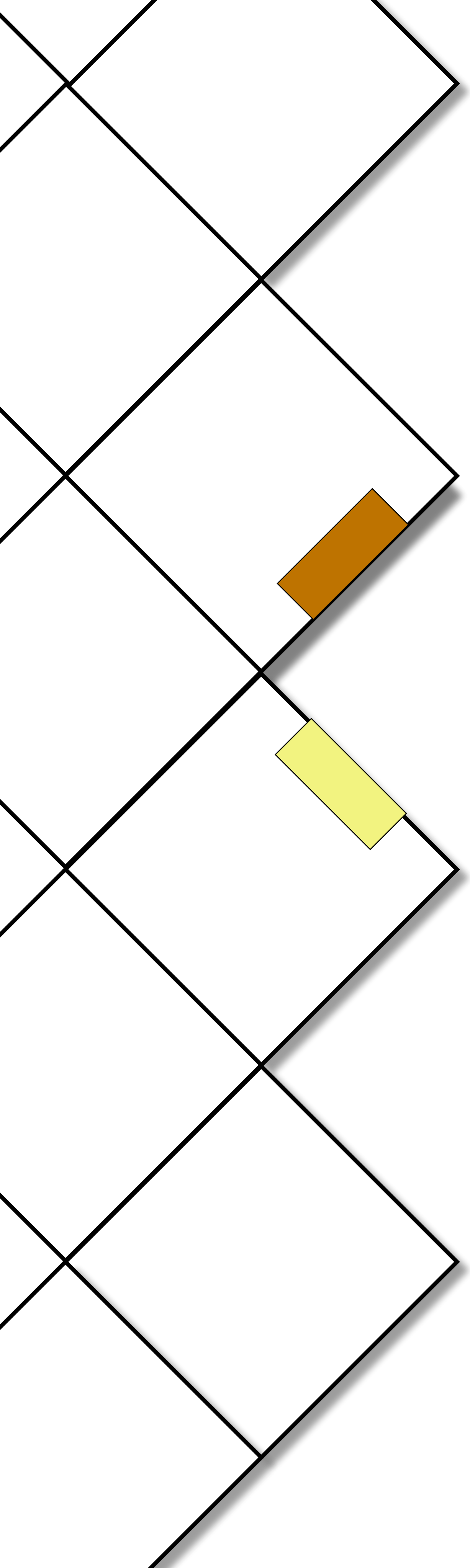
Smart self-assembly

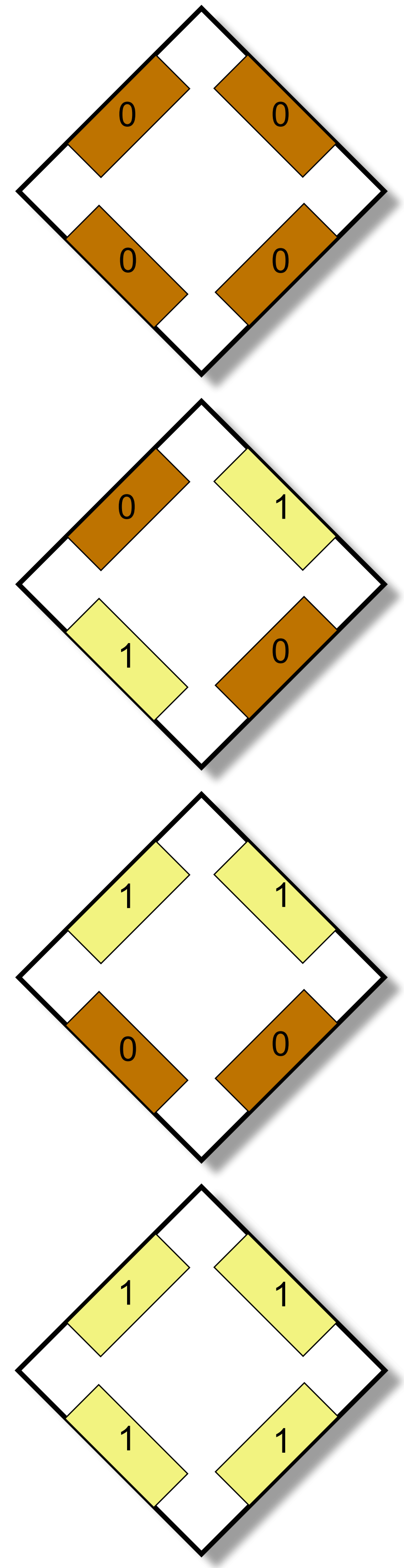
logic gates: simple, yet powerful

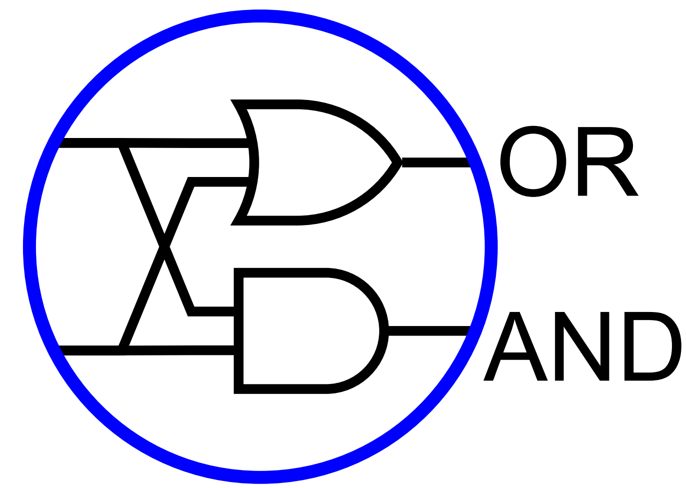




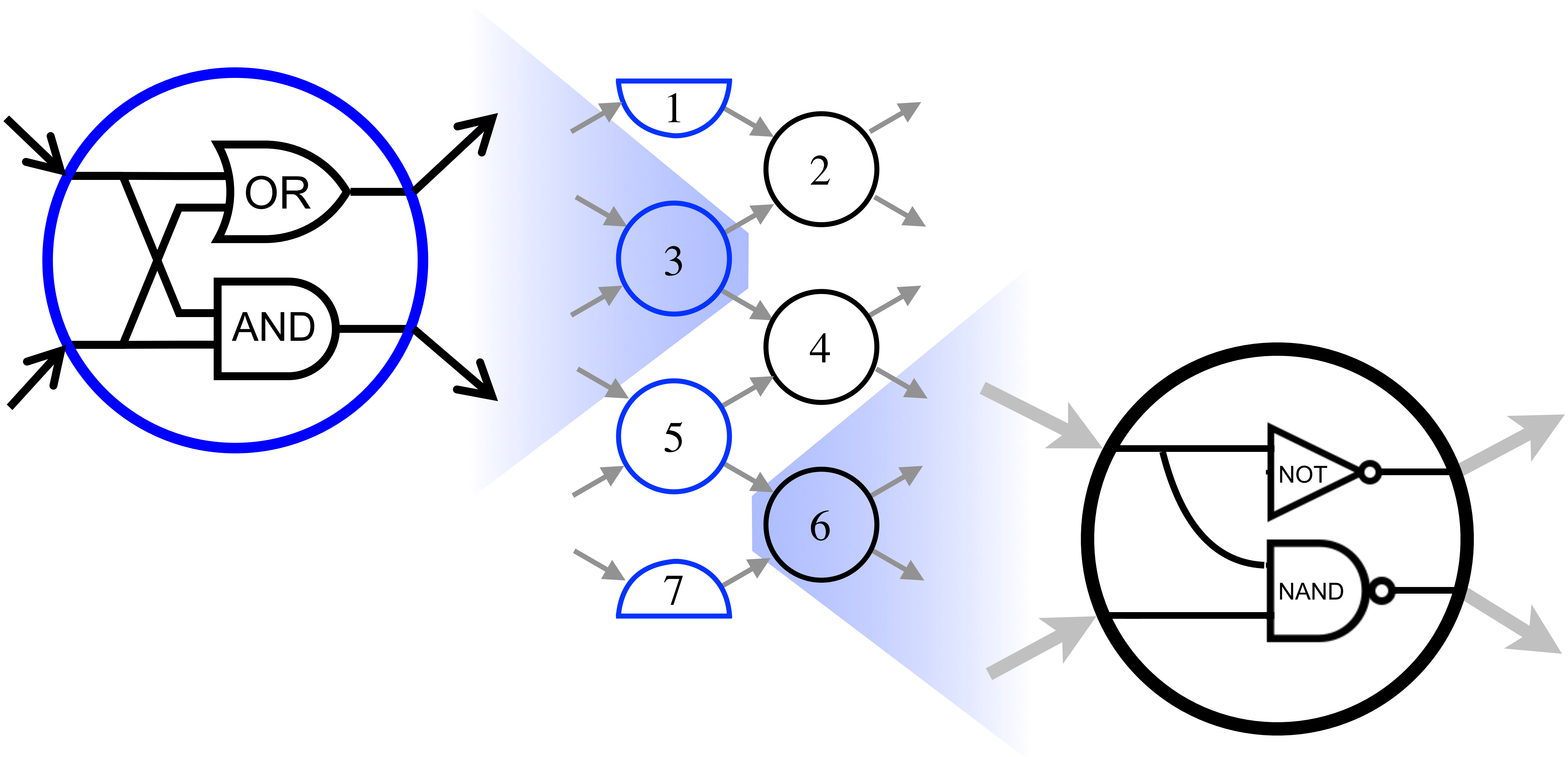
Smart self-assembly
logic gates: simple, yet powerful



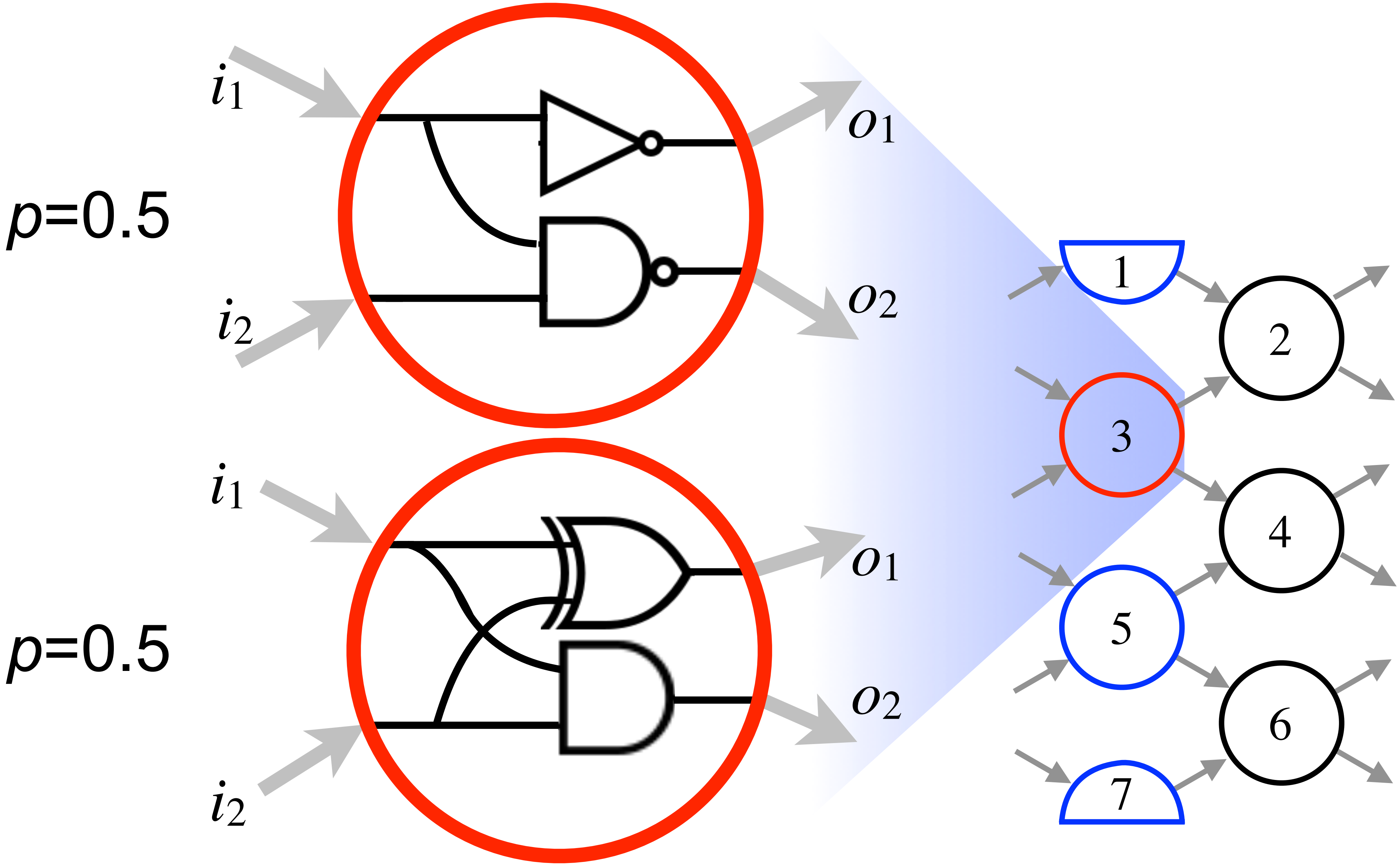




Iterated Boolean Circuit model



Iterated Boolean Circuit model: randomised gates

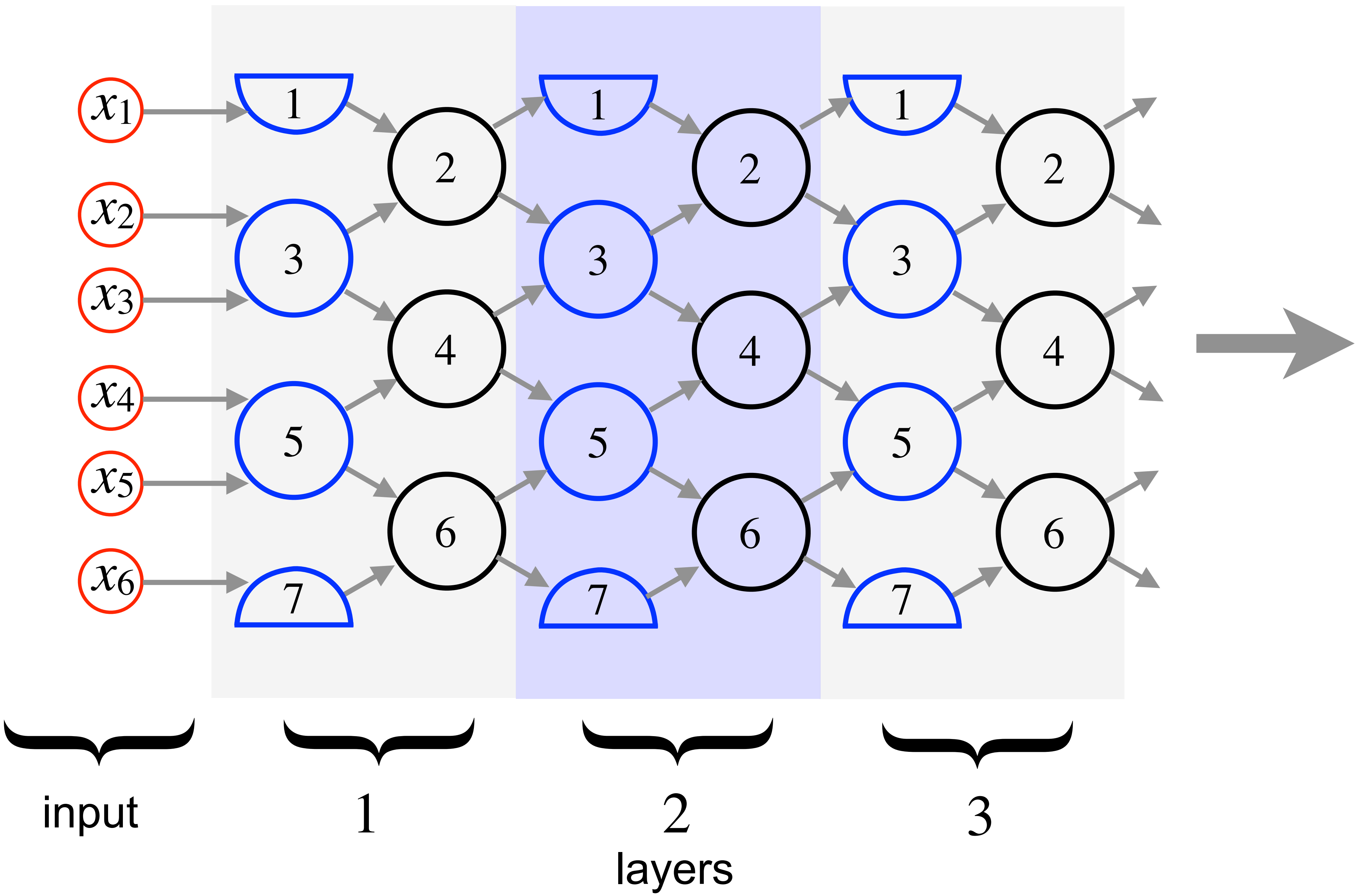


Iterated Boolean Circuit model

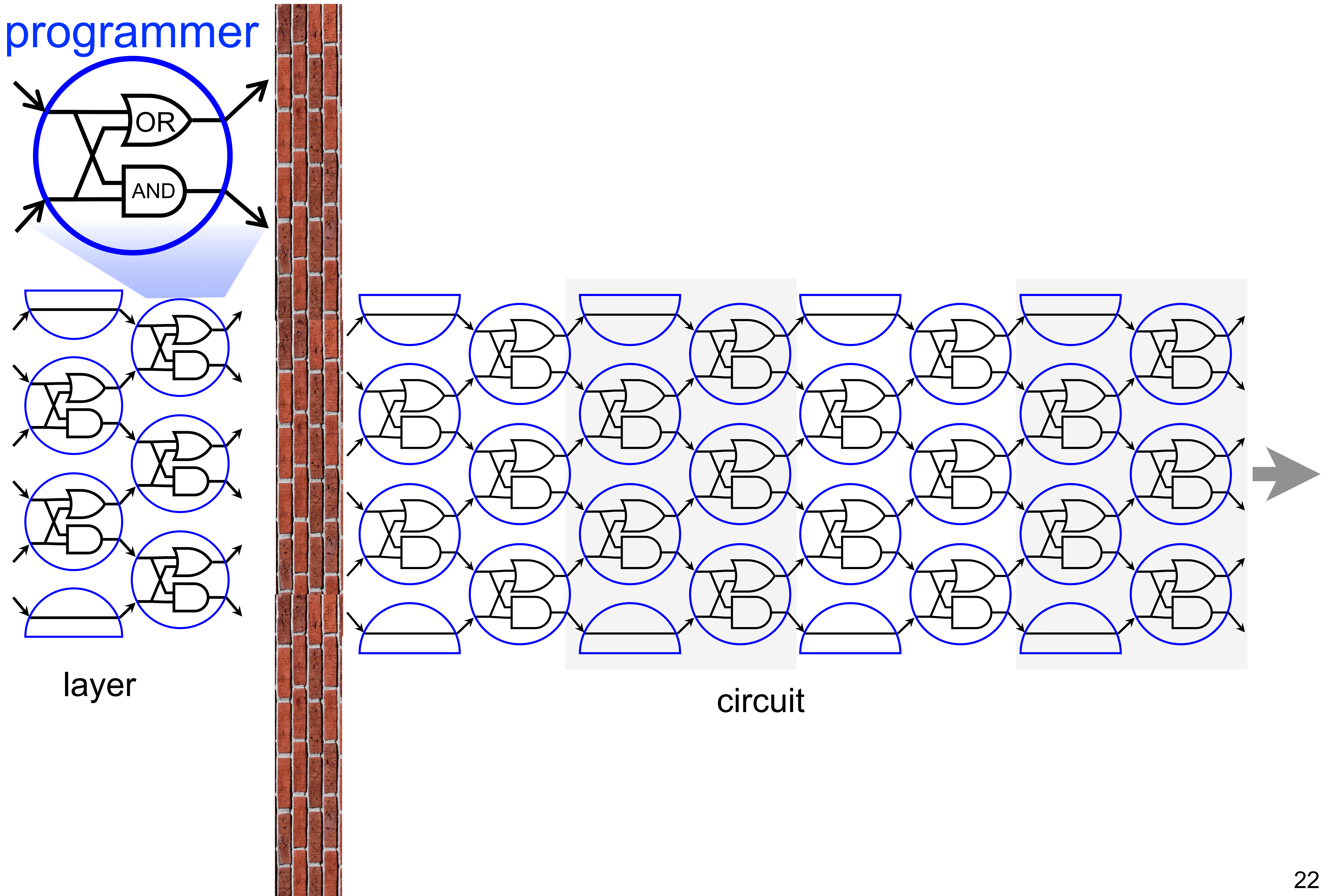
Programmer specifies a layer

User gives *n* input bits $x_k \in \{0,1\}$

Computation flows from input gates to layer 1, layer 2, layer 3 …



Example circuit

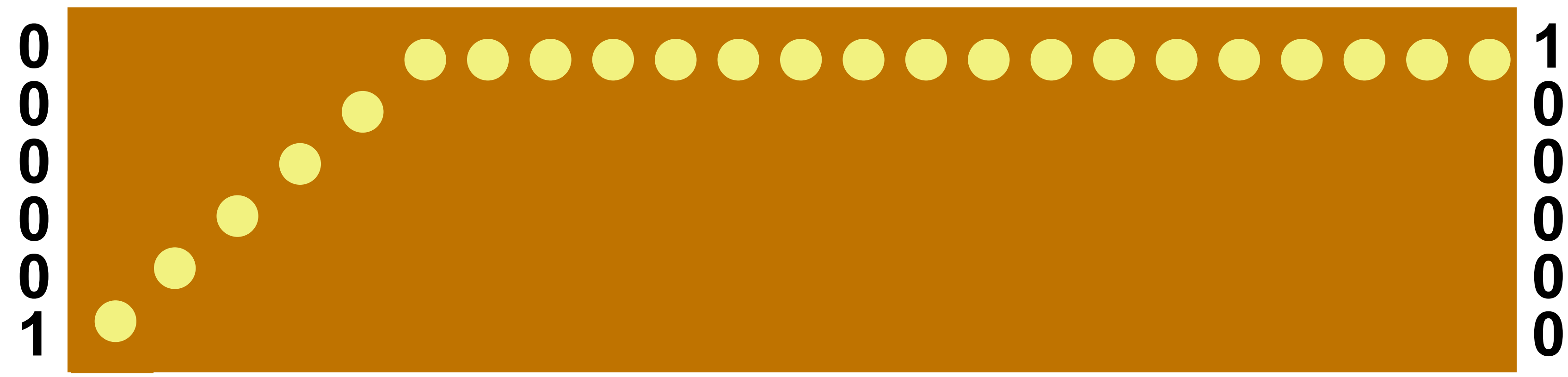


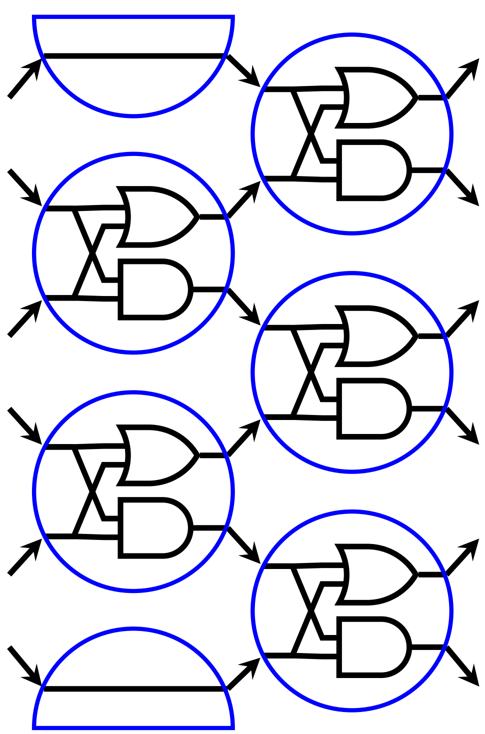
Example circuit

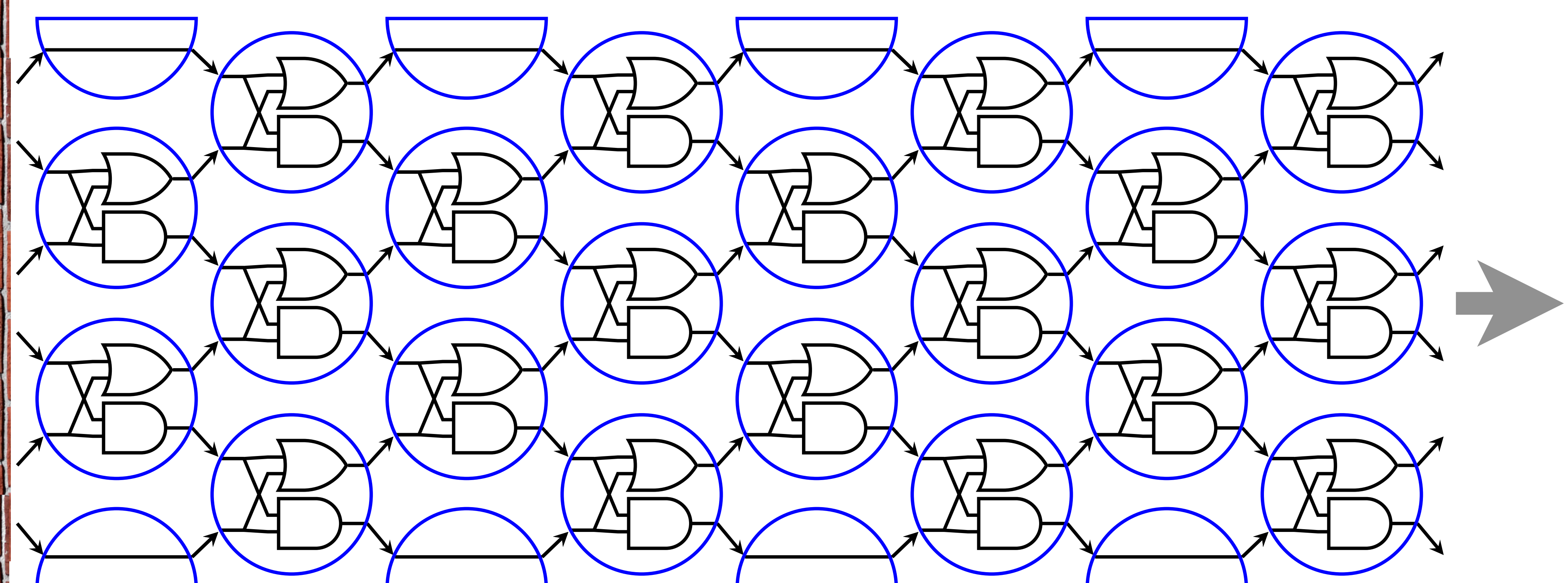
programmer

user

computation



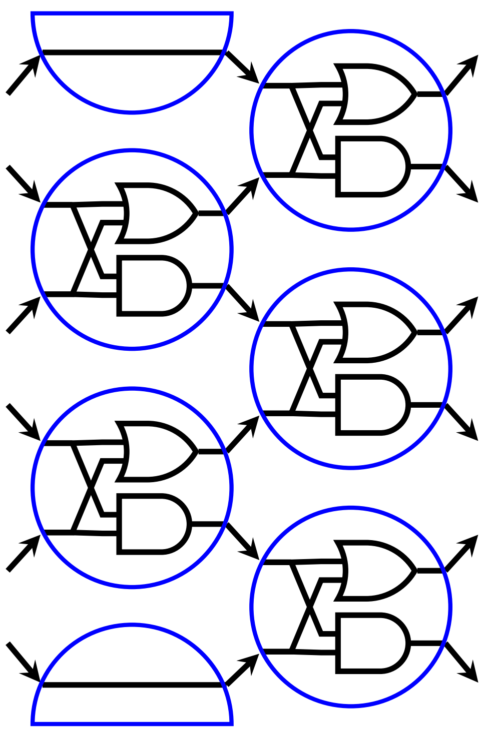




wall credit: Eric Klavins

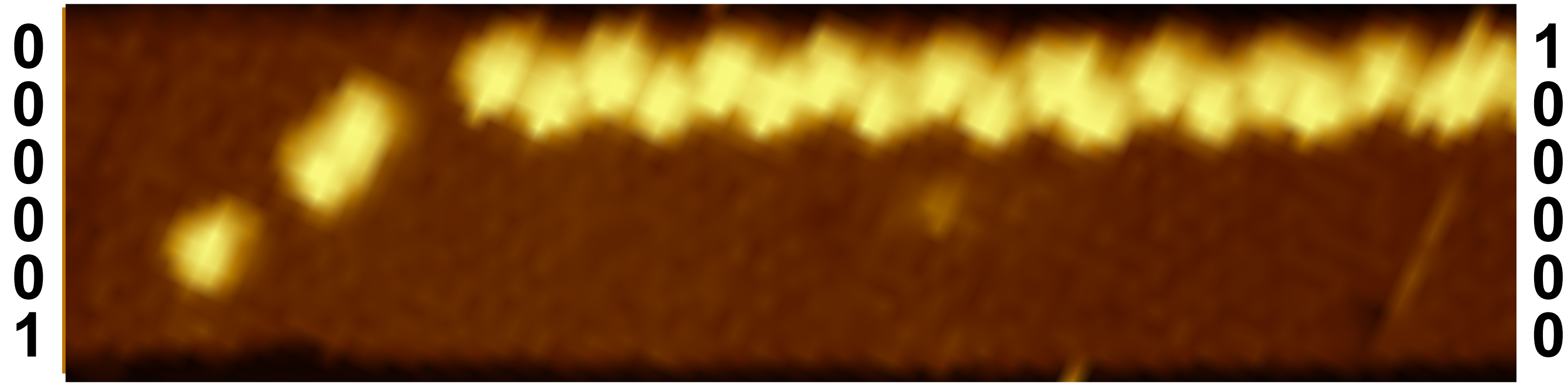
Example circuit: "SORTING"

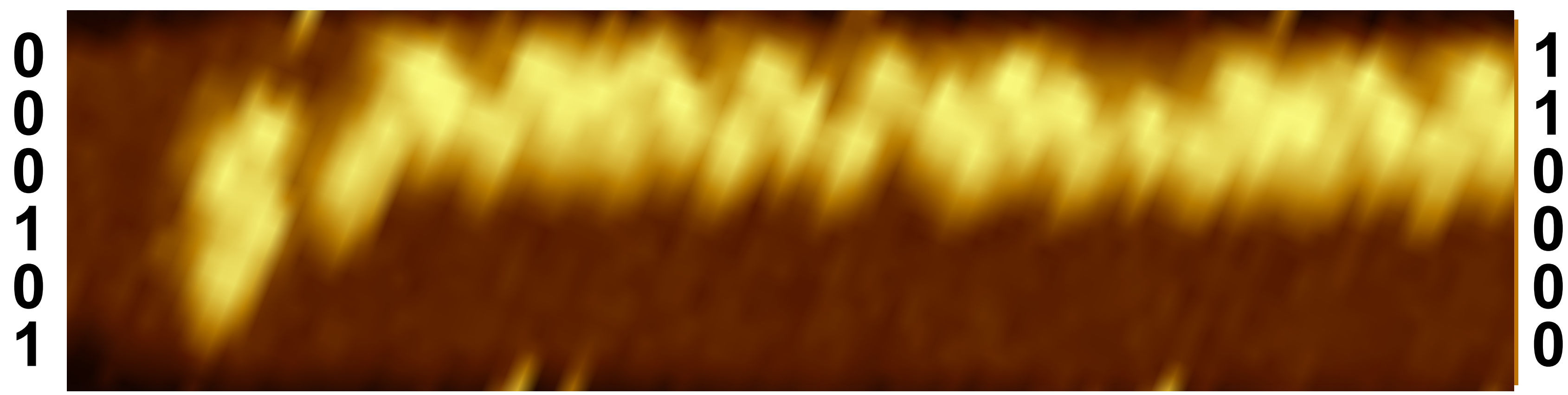
programmer

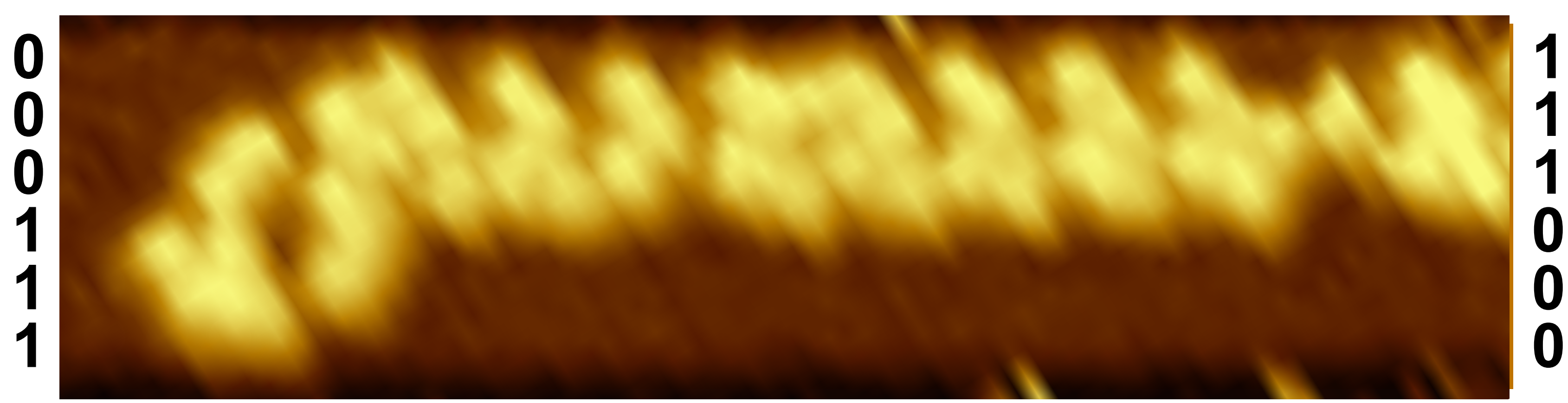


user

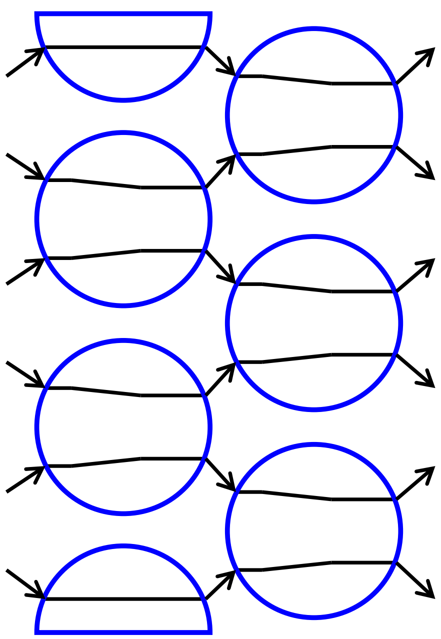
computation

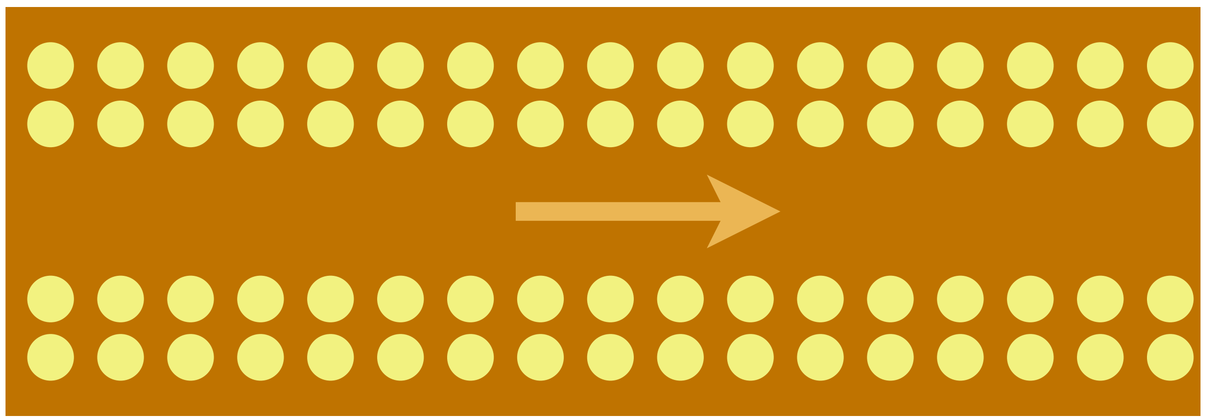






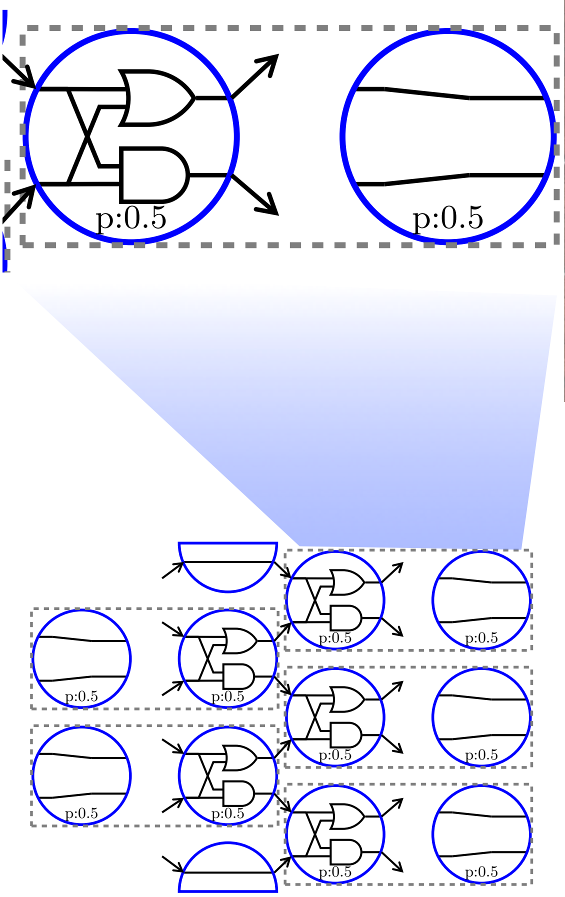
Example circuits: COPY bits to the right





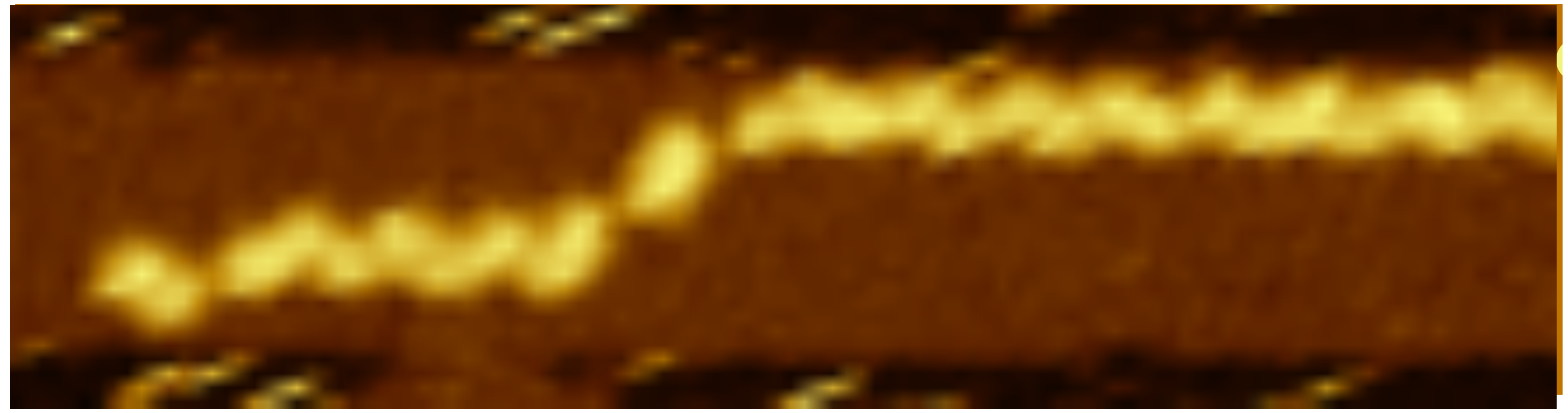
Example circuit: LAZYSORTING

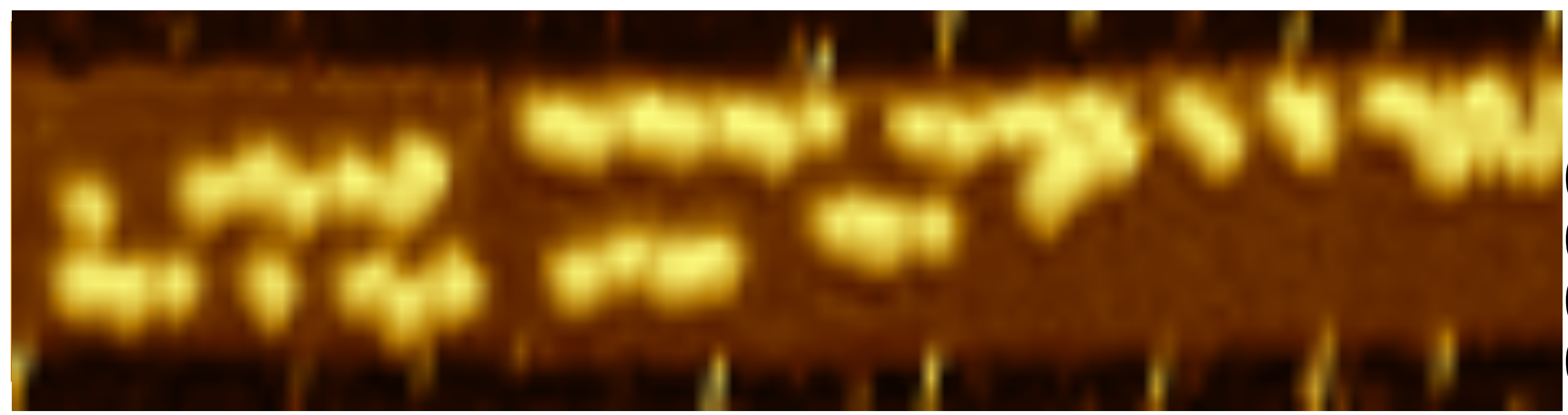
programmer



user

computation

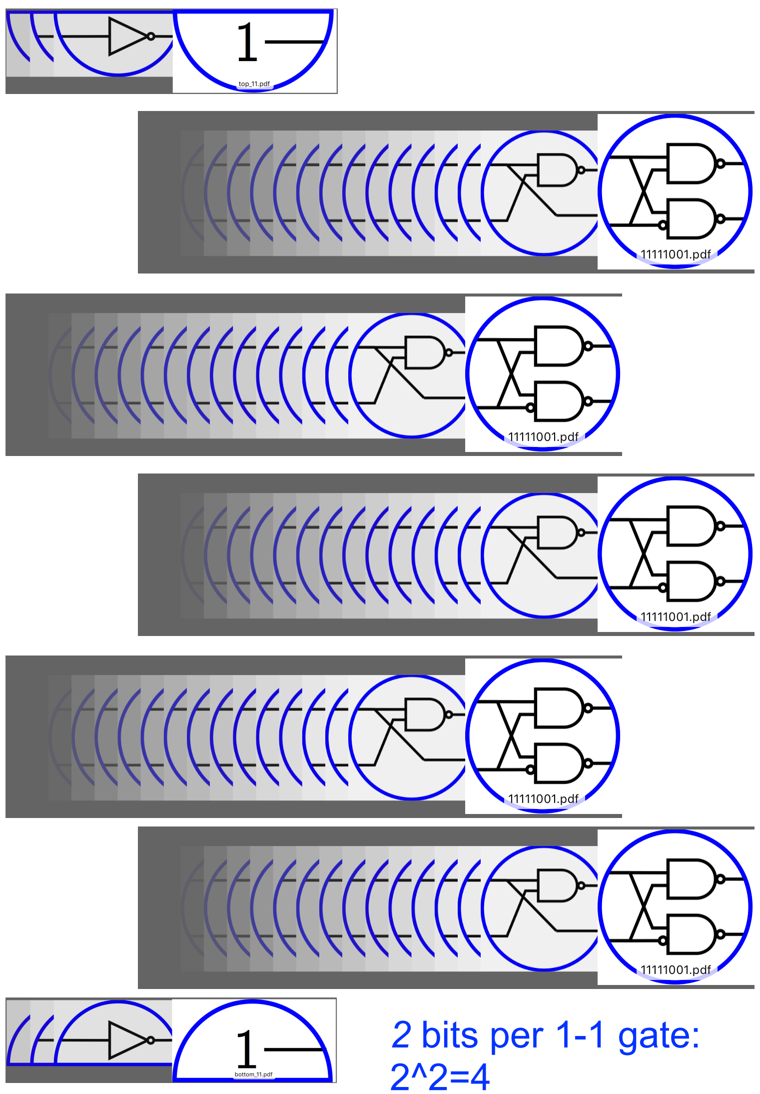
0 0 0 0 0 1

1 0 0 0 0 0

0 0 0 1 0 1

1 1 0 0 0 0

0 0 0 1 1 1

1 1 1 0 0 0

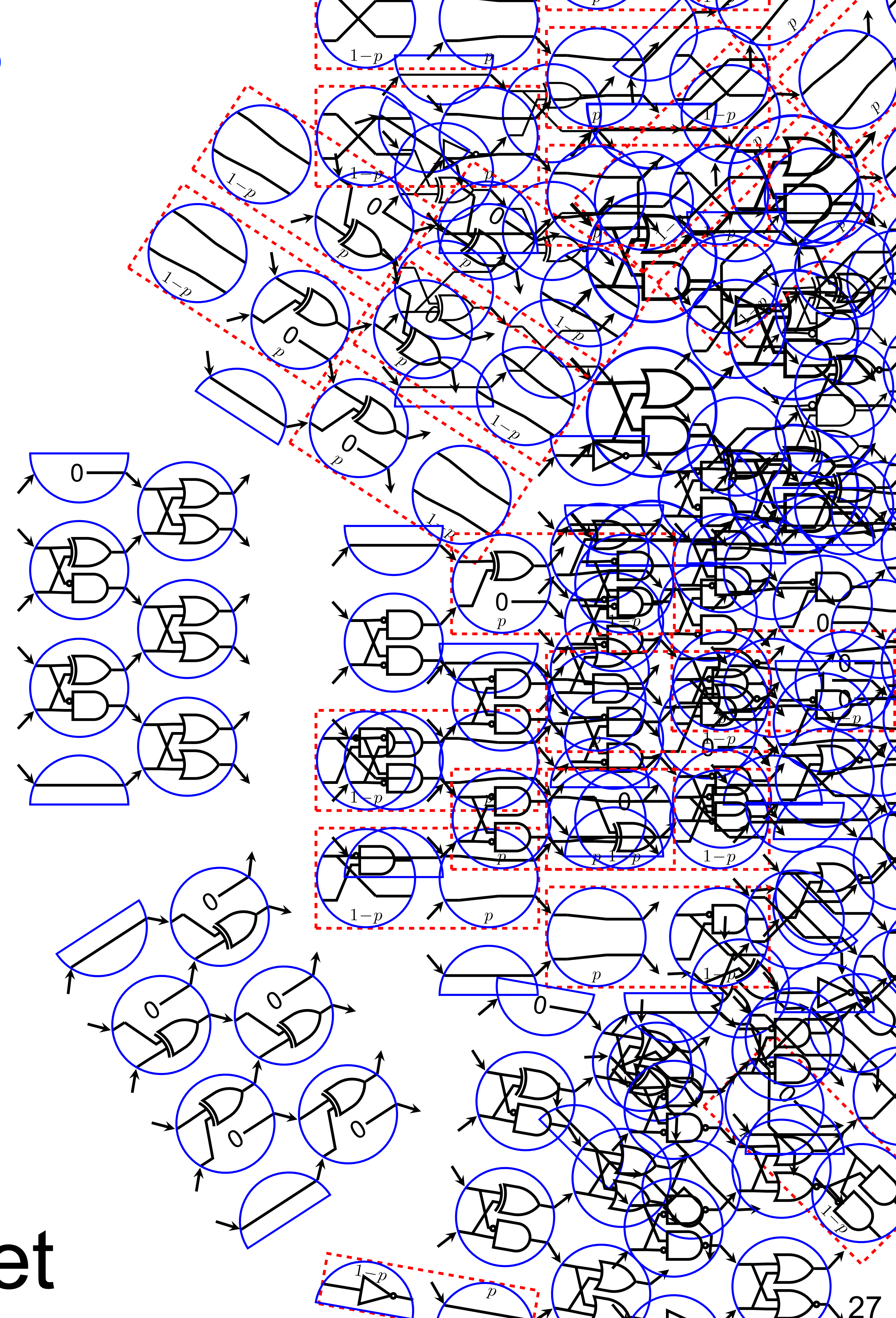
Which circuits to build?



8 bits per 2-2 gate:
2^8=256

2 bits per 1-1 gate:
2^2=4

1,288 gates that implement **any** 6-bit circuit



“Complete” 6-bit gate set

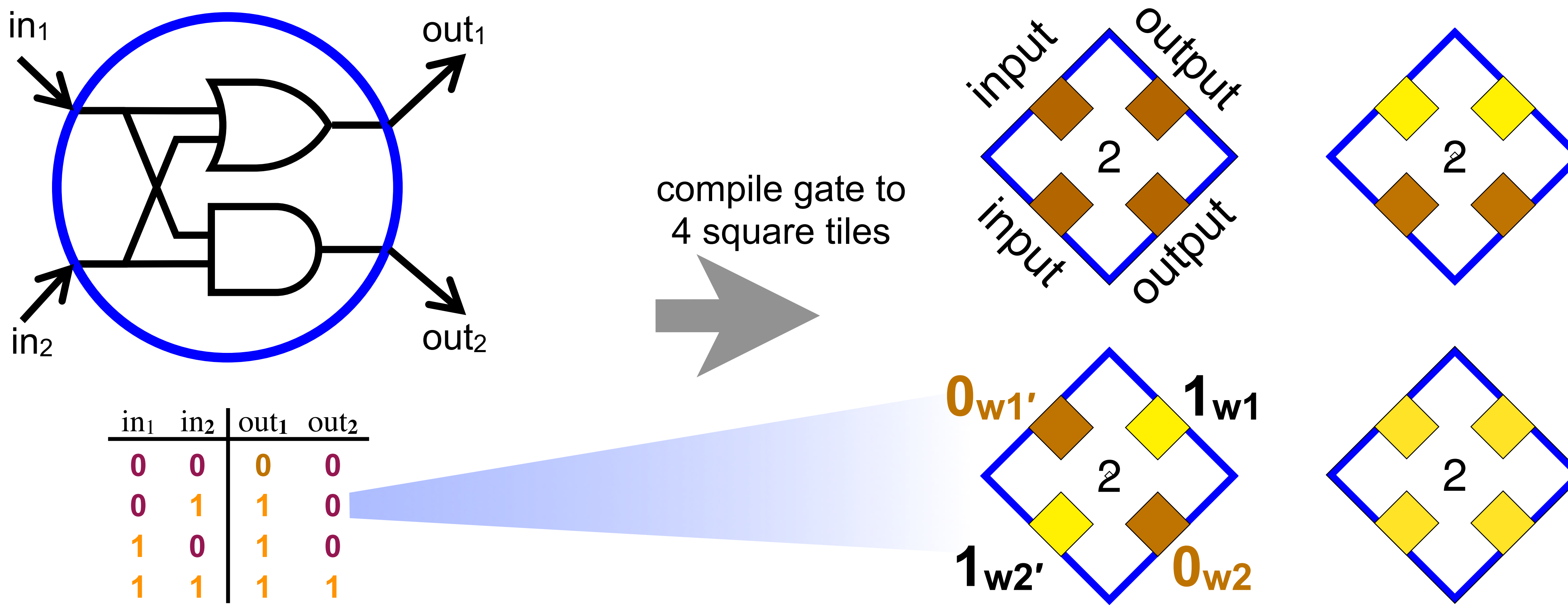
Structure

Theoretical circuit model

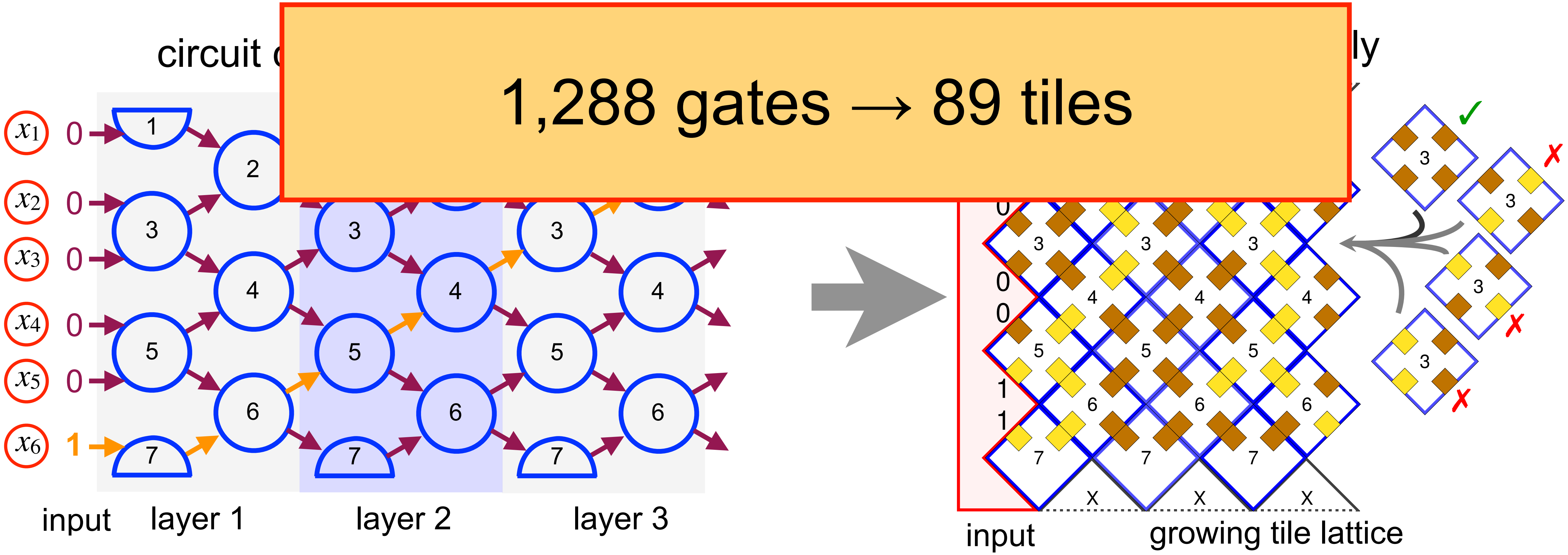
How it works: design and implementation

Experimental results

From circuits to square tiles

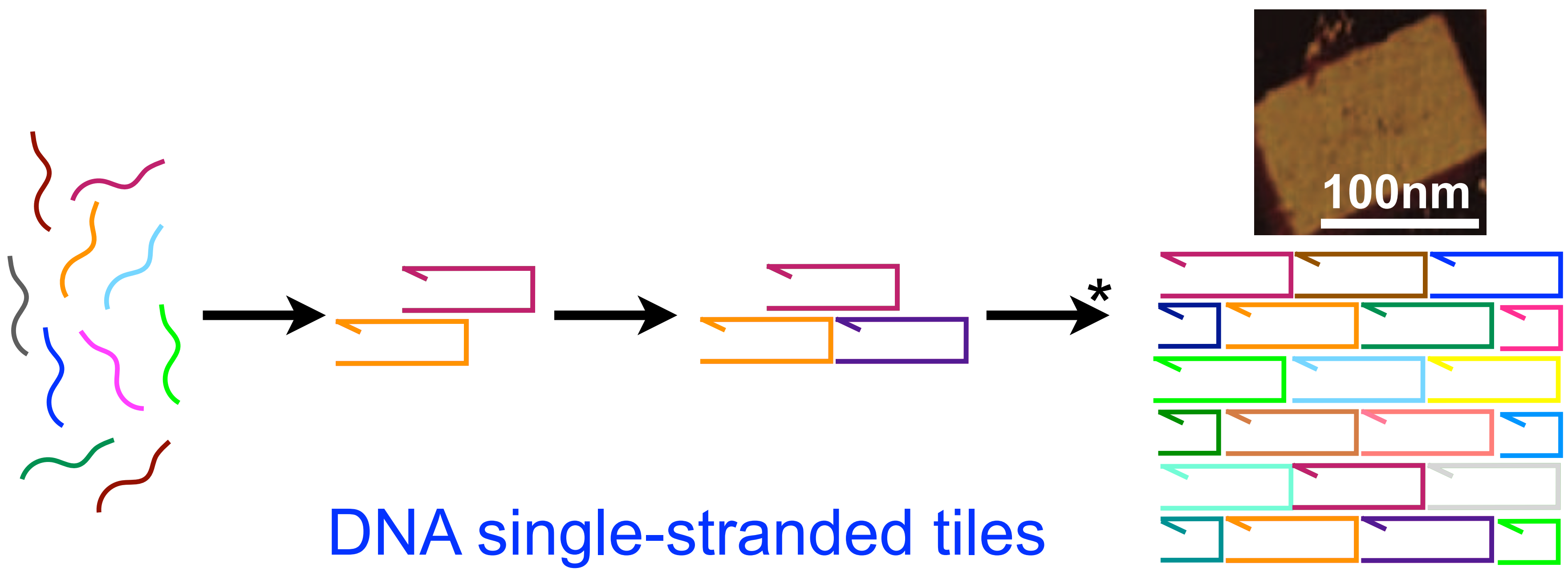


in_1	in_2	out_1	out_2
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1



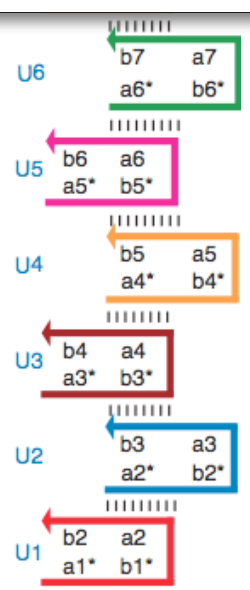
1,288 gates → 89 tiles

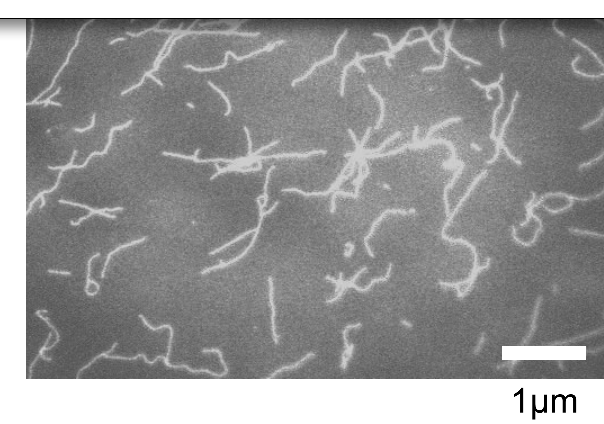
Single-stranded tile motif



DNA single-stranded tiles

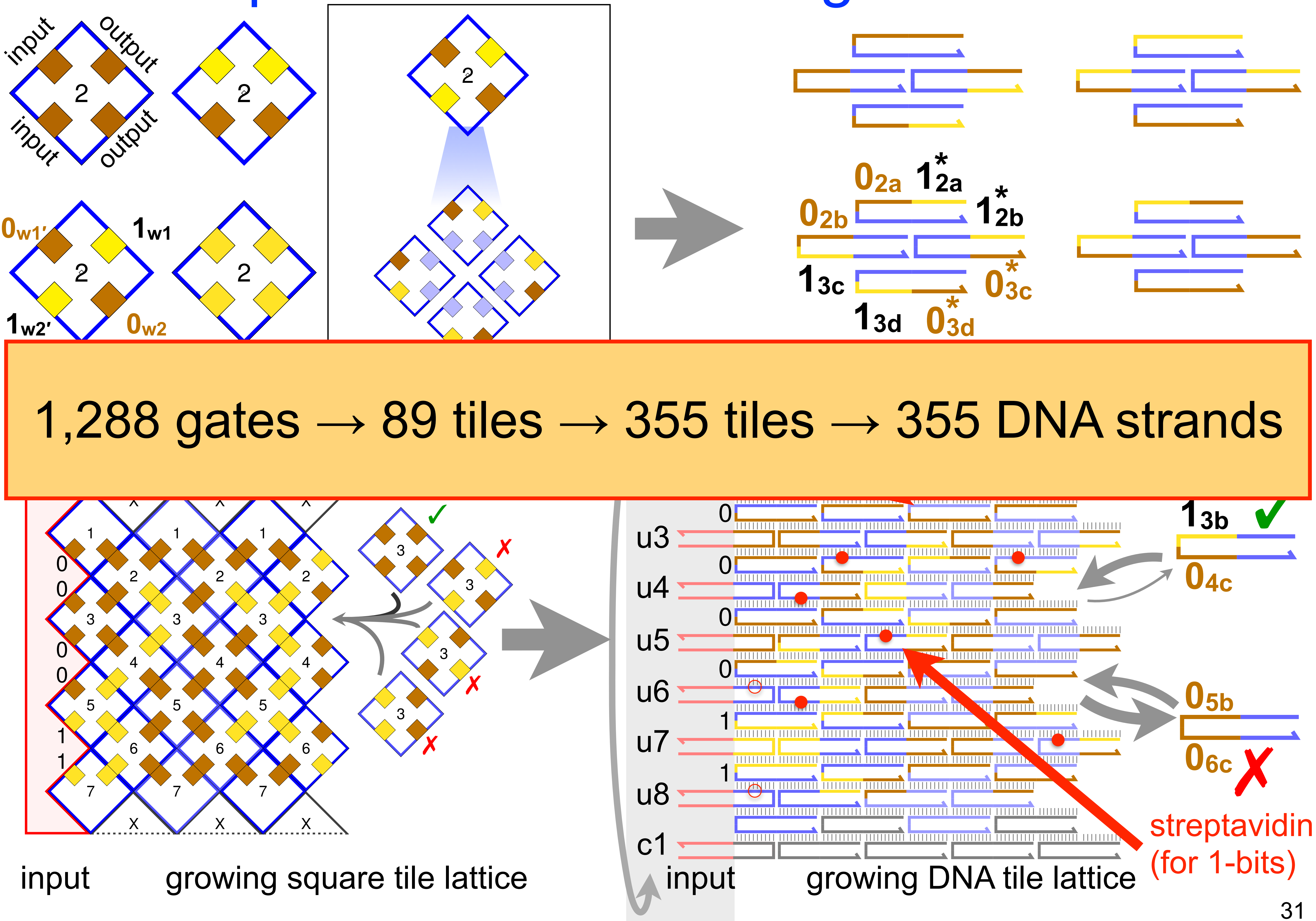
Wei, Dai, Yin. 2012 Nature





Yin, Hariadi, Sahu, Choi, Park, LaBean, Reif. Science. 2008

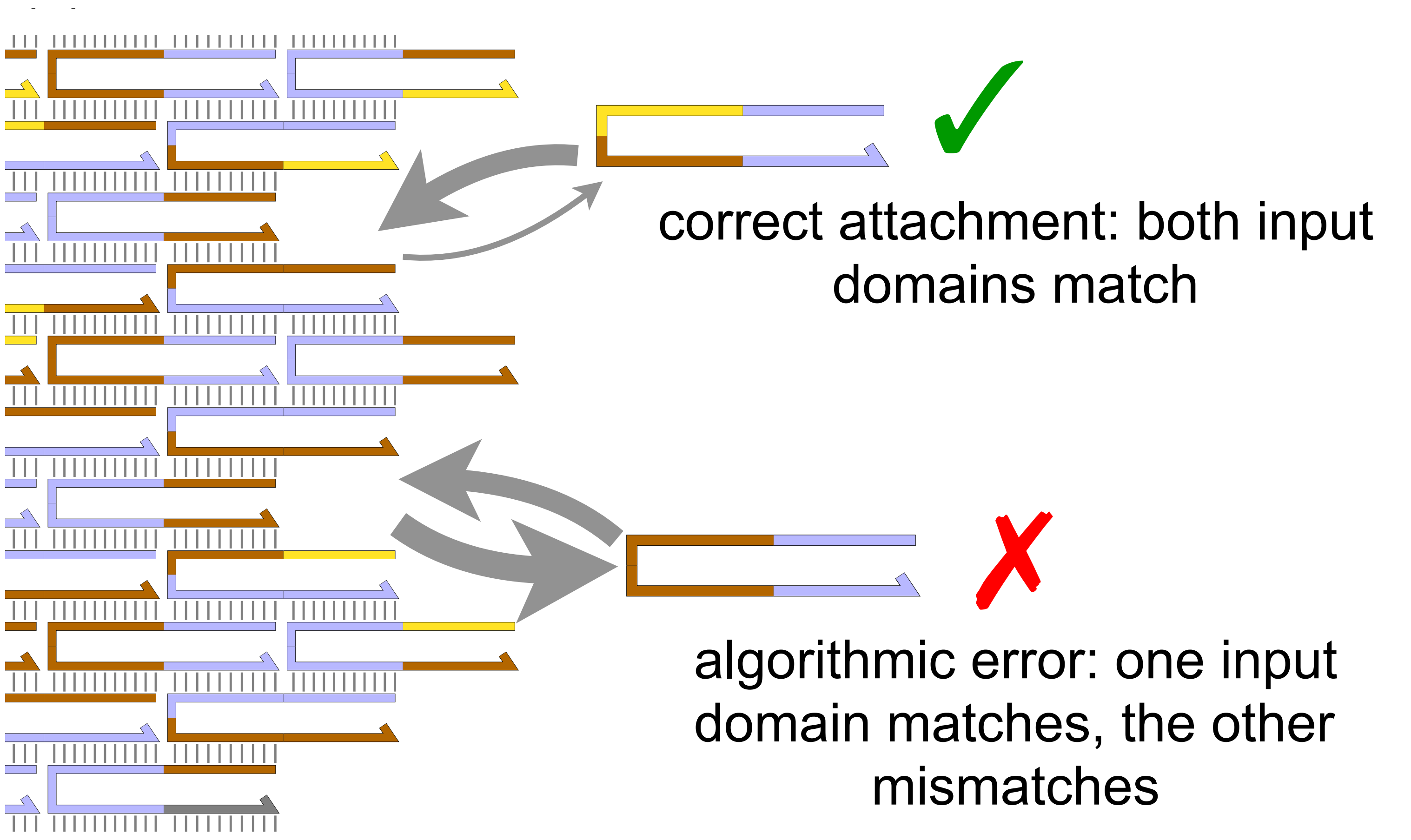
From square tiles to DNA single-stranded tiles



1,288 gates → 89 tiles → 355 tiles → 355 DNA strands

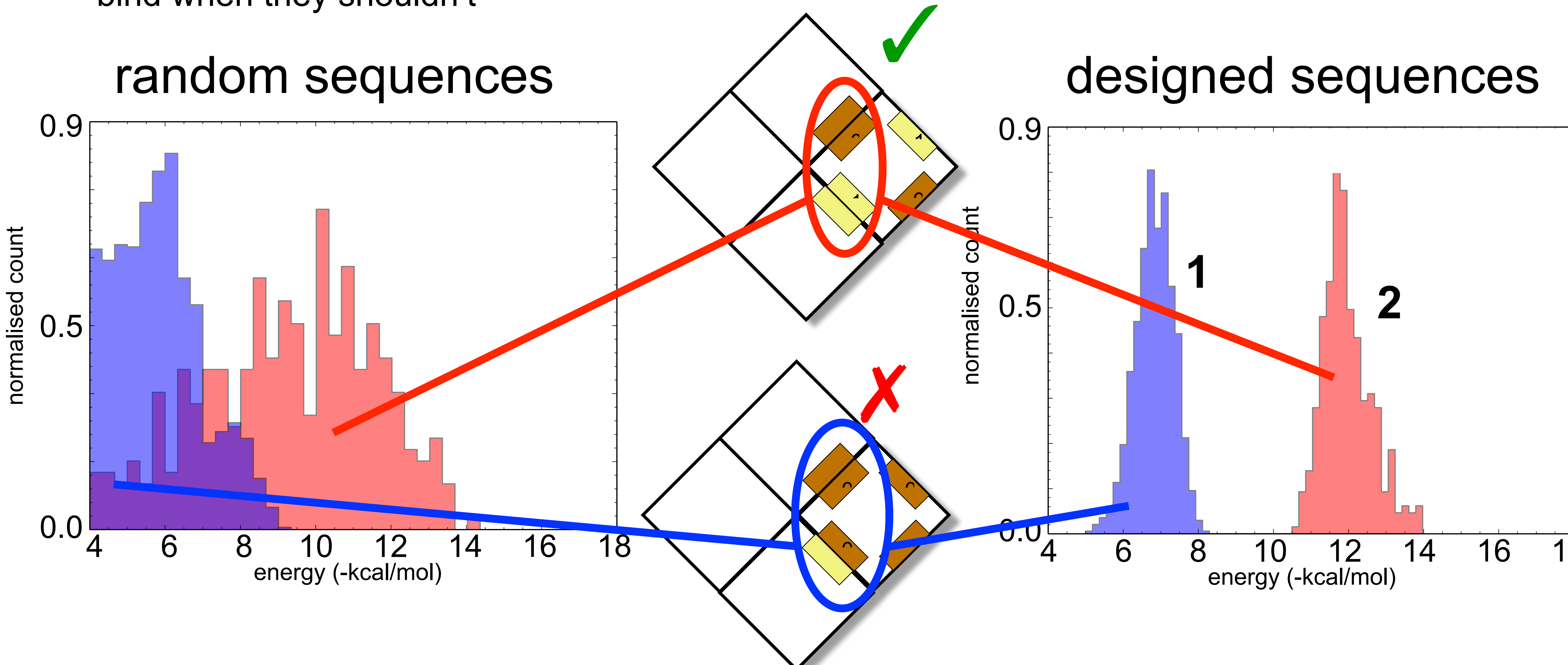
DNA sequence design

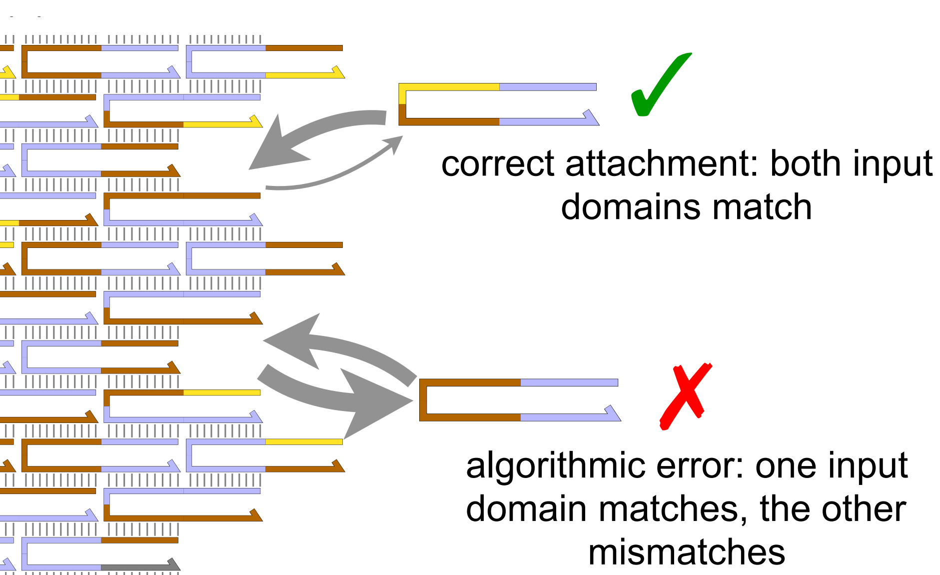
- Major challenge: We need to design DNA sequences that bind when they should, and to not bind when they shouldn't



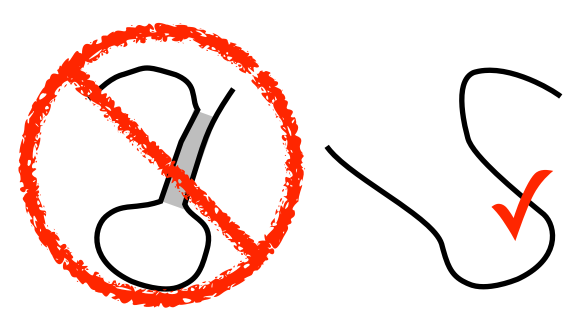
DNA sequence design

- Major challenge: We need to design DNA sequences that bind when they should, and to not bind when they shouldn't



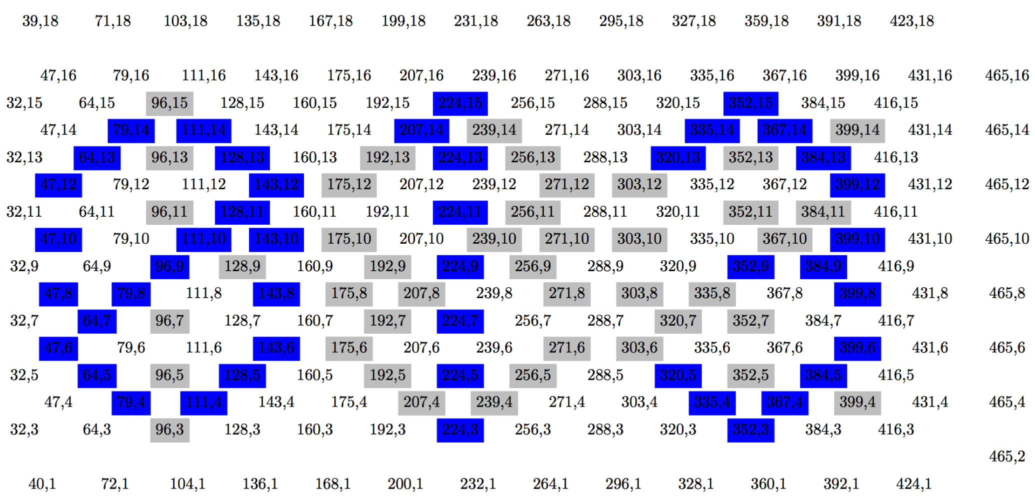


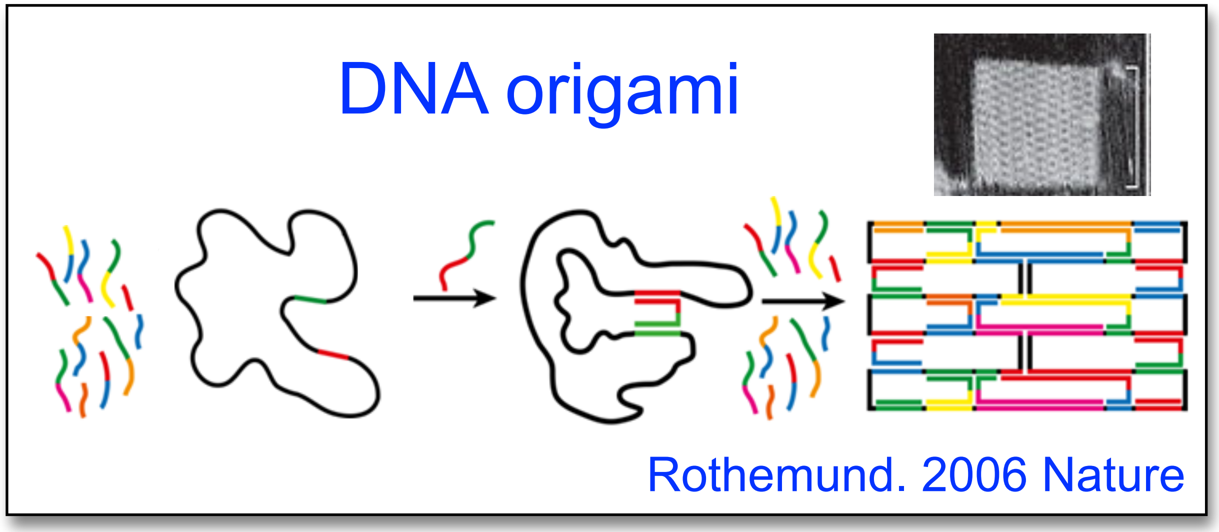
- Sequences designed via a stochastic local search algorithm
- Some design problems decomposed into smaller problems for non-pseudoknotted structures
- Calls NUPACK and ViennaRNA to evaluate energetics



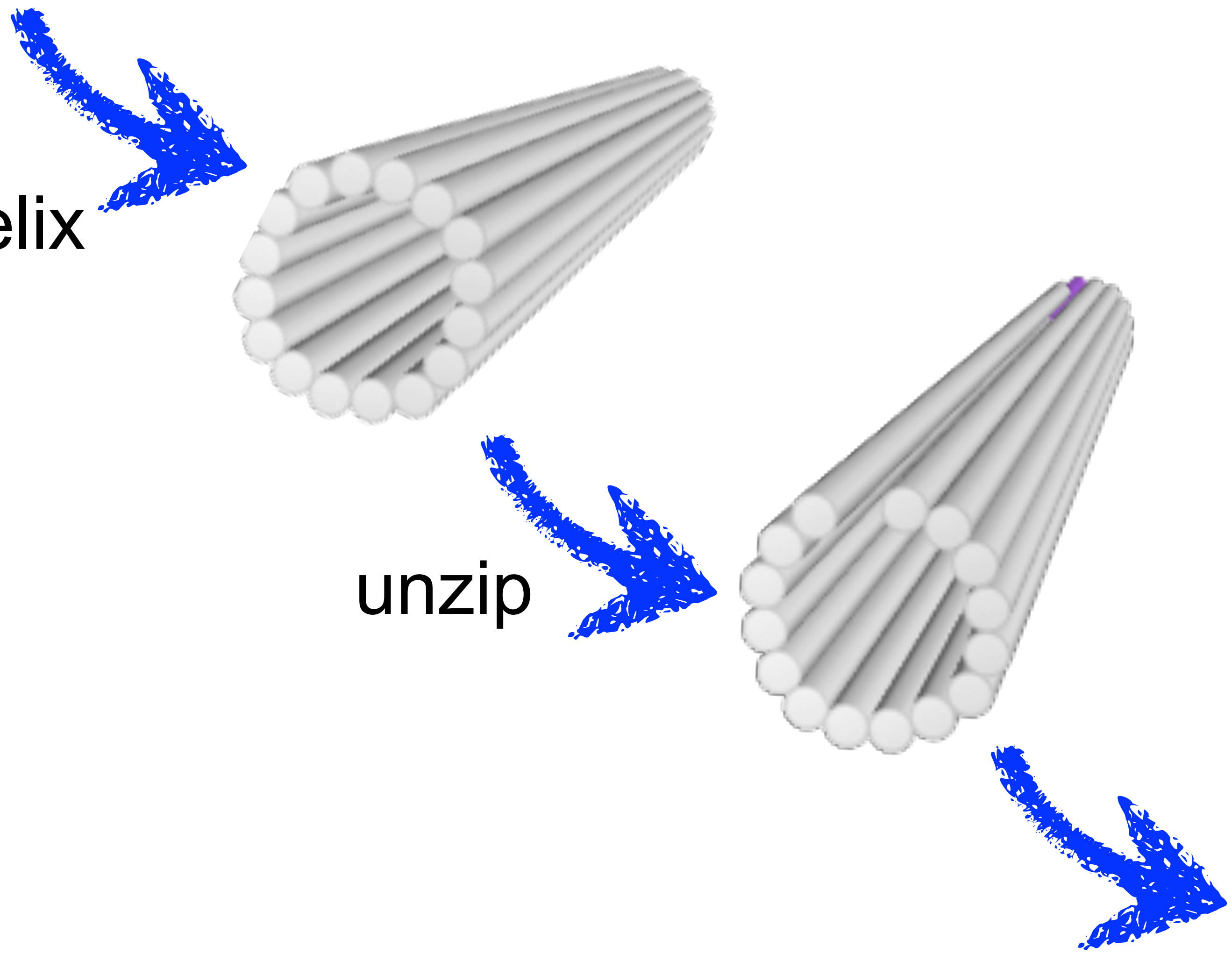
- Isoenergetic binding
- Strand sec struct
- Clean lattice
- Strand pairs

Barcoded DNA origami seed





form
16-helix
tube



unzip

add streptadividin
& image on mica

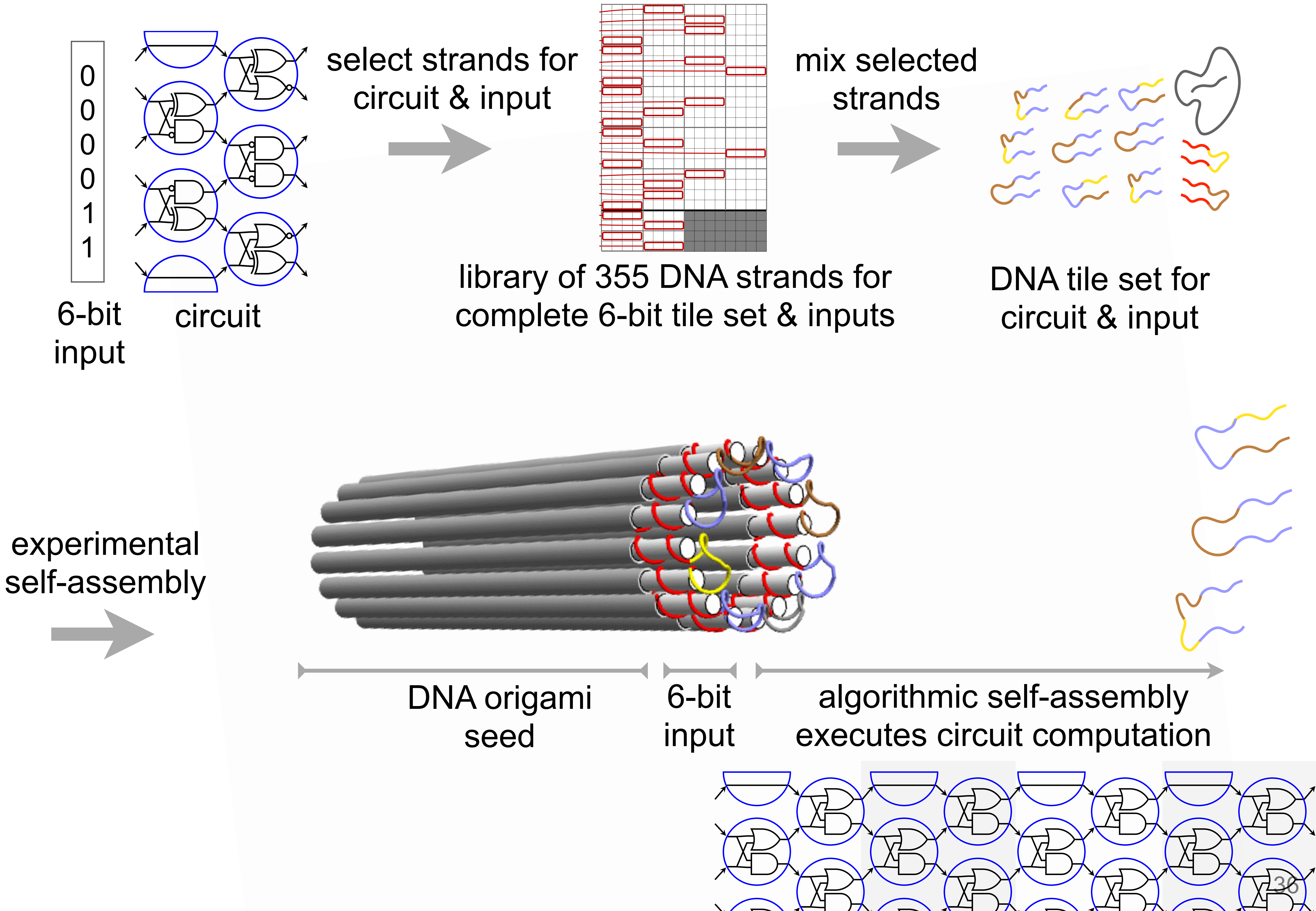
Structure

Theoretical circuit model

How it works: design and implementation

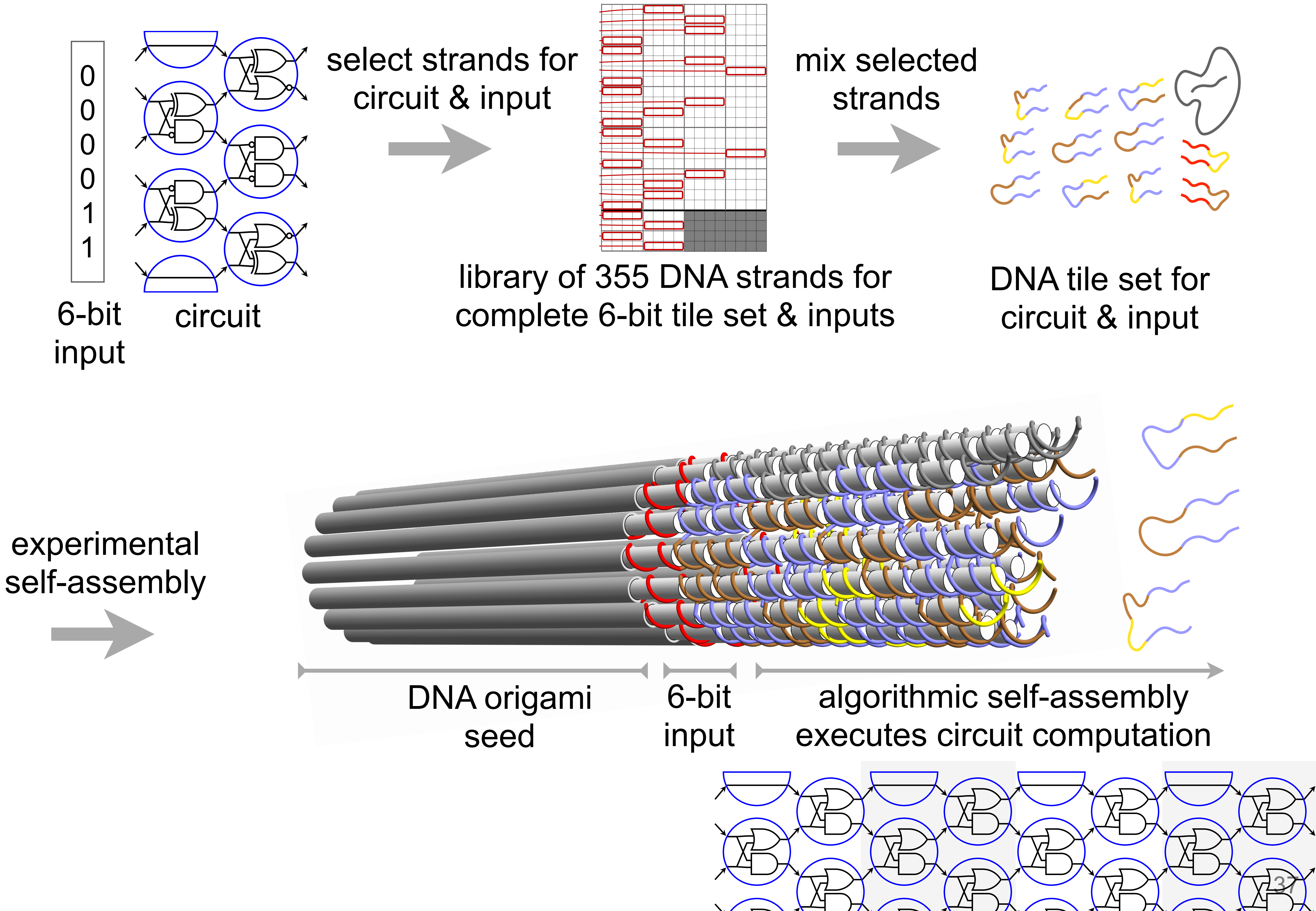
Experimental results

Schematic



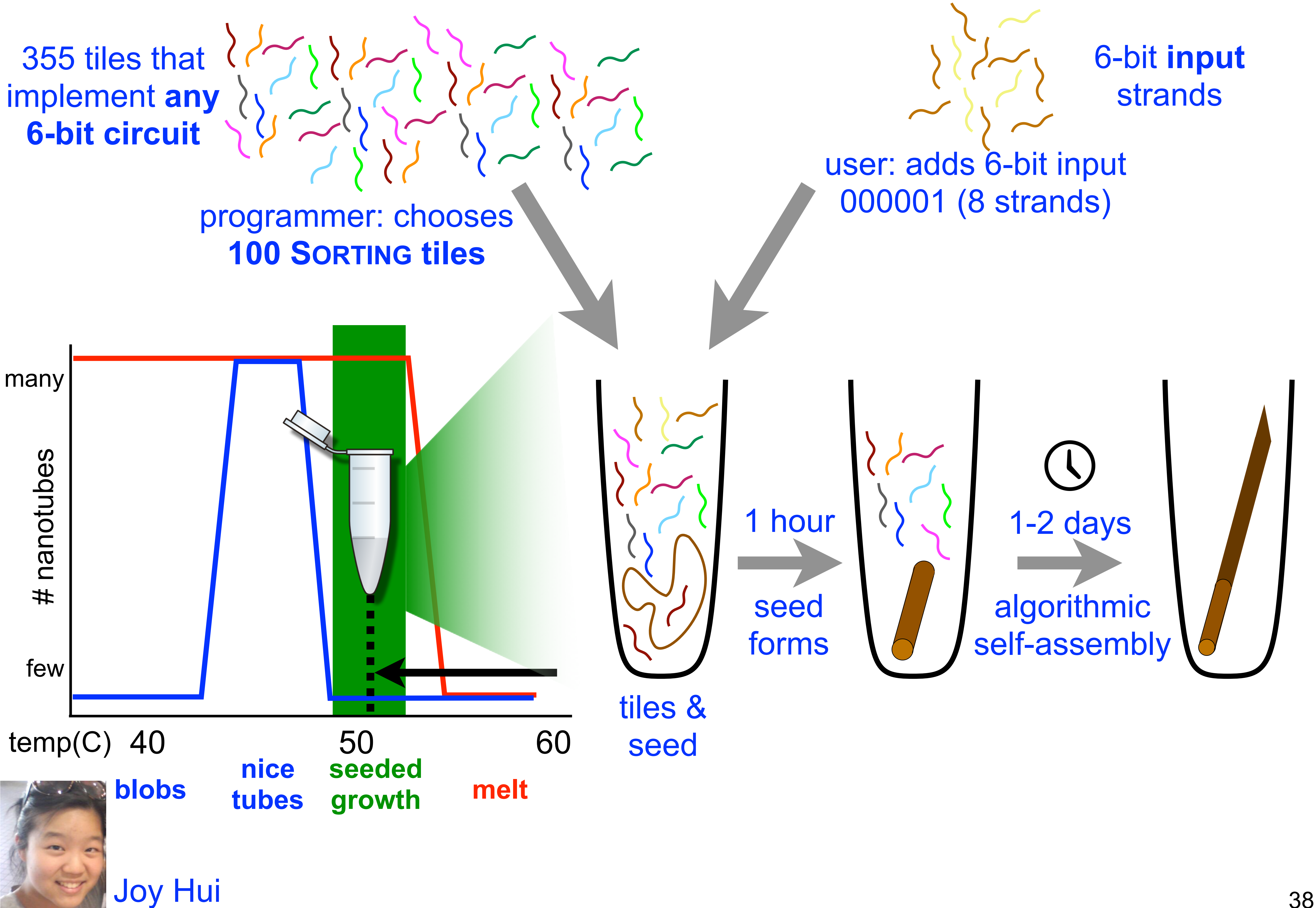
0
0
0
0
1
1
6-bit input
circuit
select strands for circuit & input
library of 355 DNA strands for complete 6-bit tile set & inputs
mix selected strands
DNA tile set for circuit & input
experimental self-assembly
DNA origami seed
6-bit input
algorithmic self-assembly executes circuit computation

Schematic

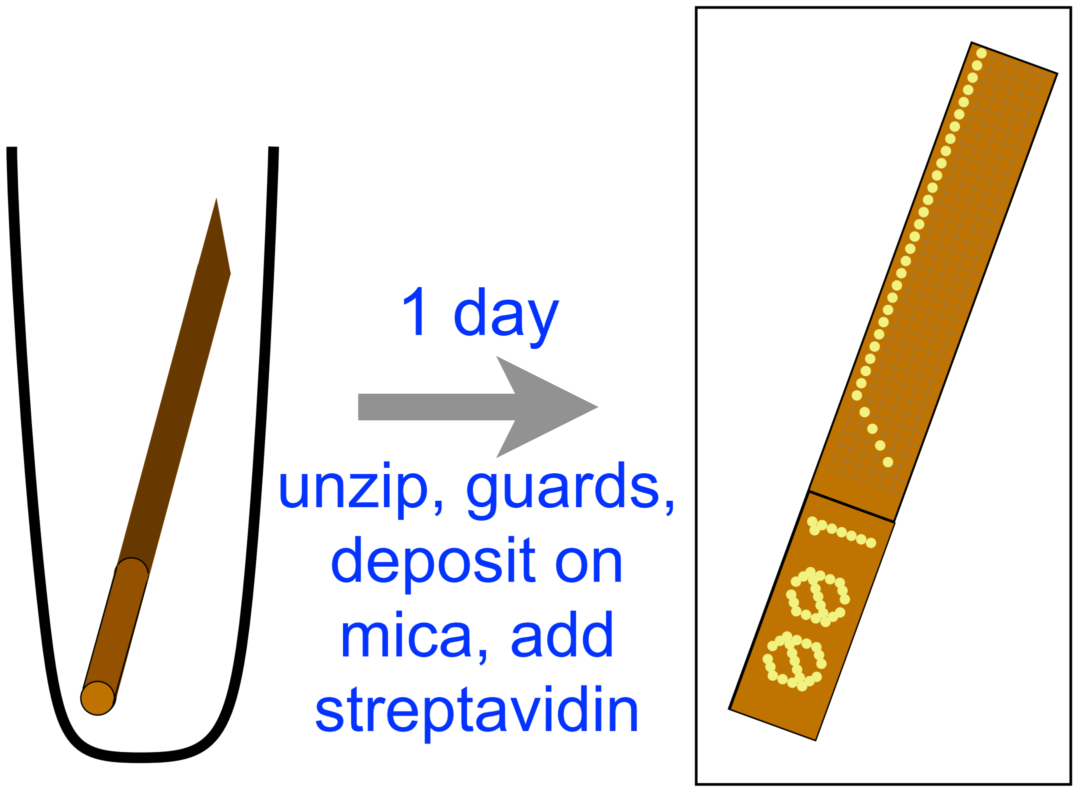


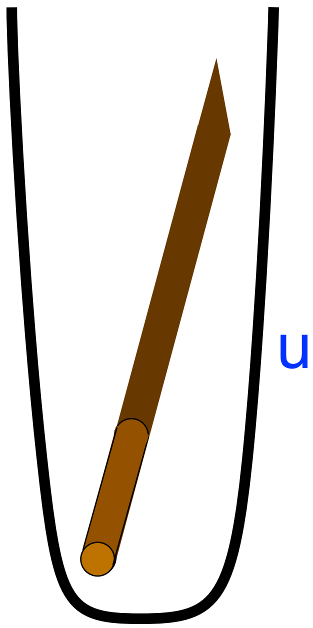
0
0
0
0
1
1
6-bit input
circuit
select strands for circuit & input
library of 355 DNA strands for complete 6-bit tile set & inputs
mix selected strands
DNA tile set for circuit & input
experimental self-assembly
DNA origami seed
6-bit input
algorithmic self-assembly executes circuit computation

An example experiment: SORTING



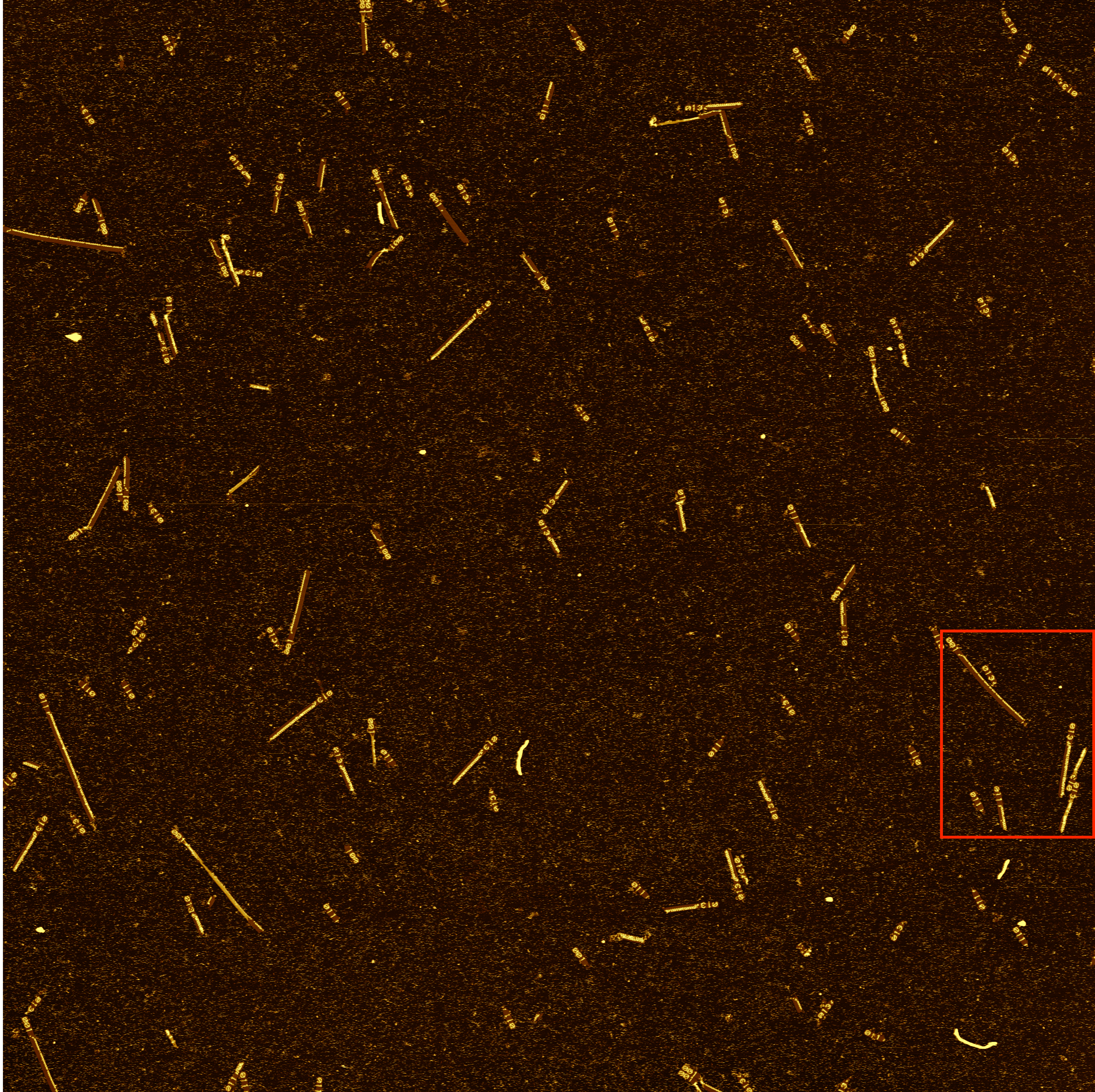
An example experiment: Sorting





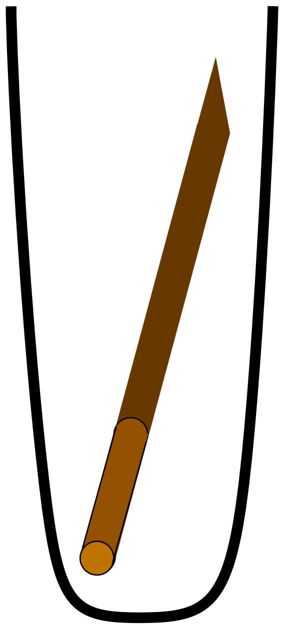
1 day

unzip, guards, deposit on mica, add streptavidin



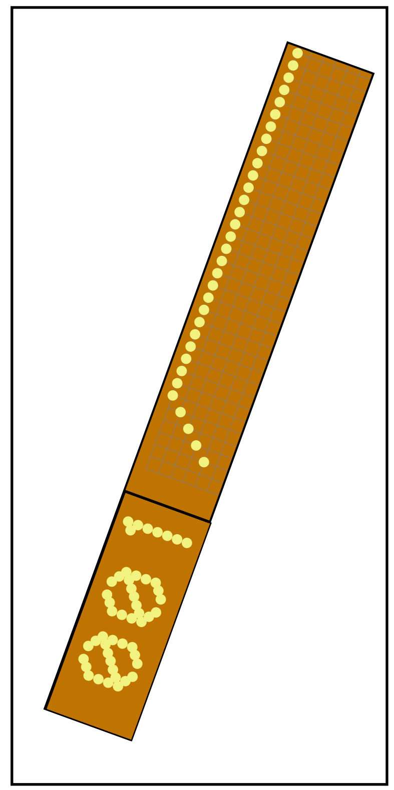
8 μm x 8 μm

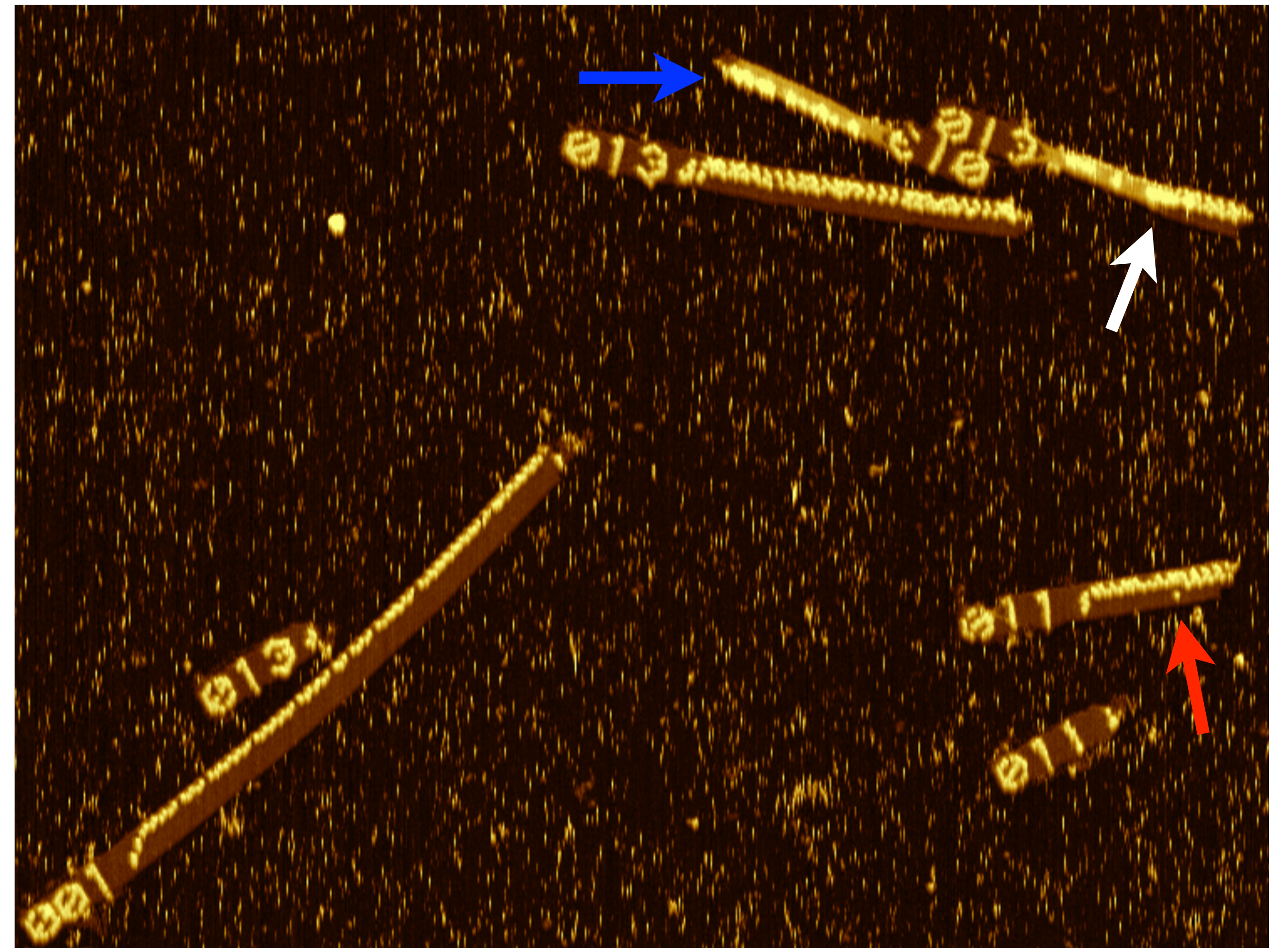
An example experiment: Sorting



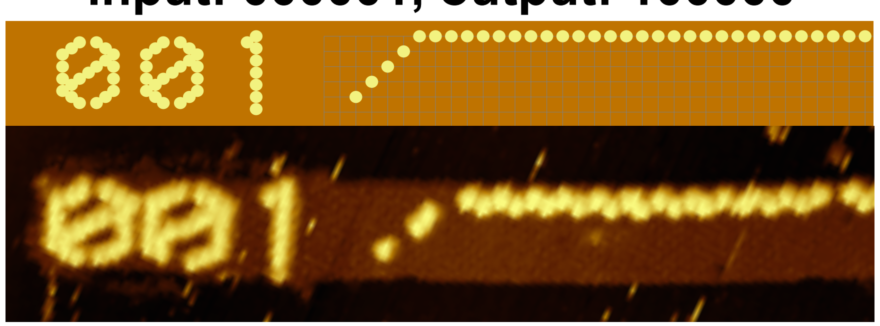
1 day

unzip, guards, deposit on mica, add streptavidin

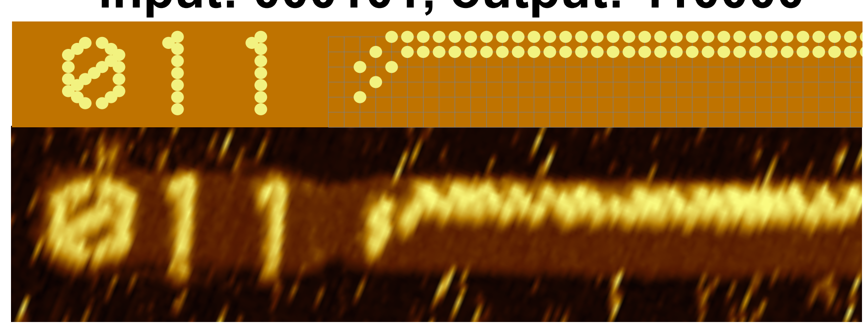




input: 000001, output: 100000

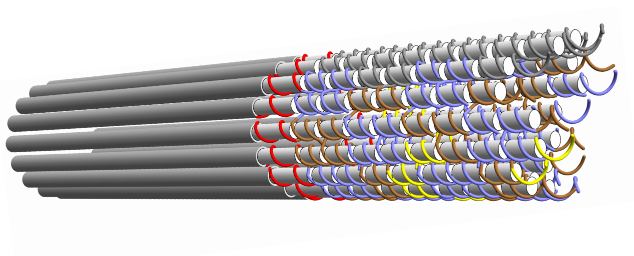


input: 000101, output: 110000

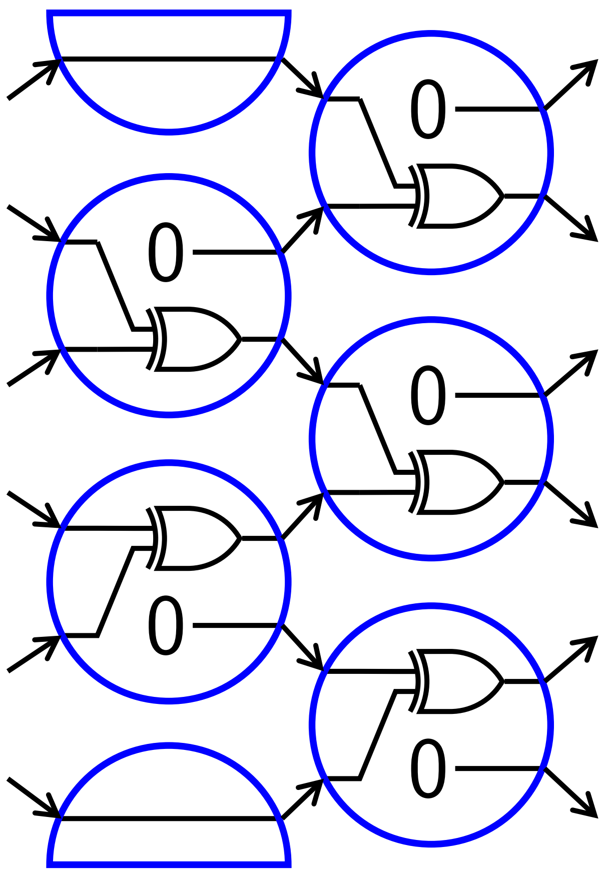


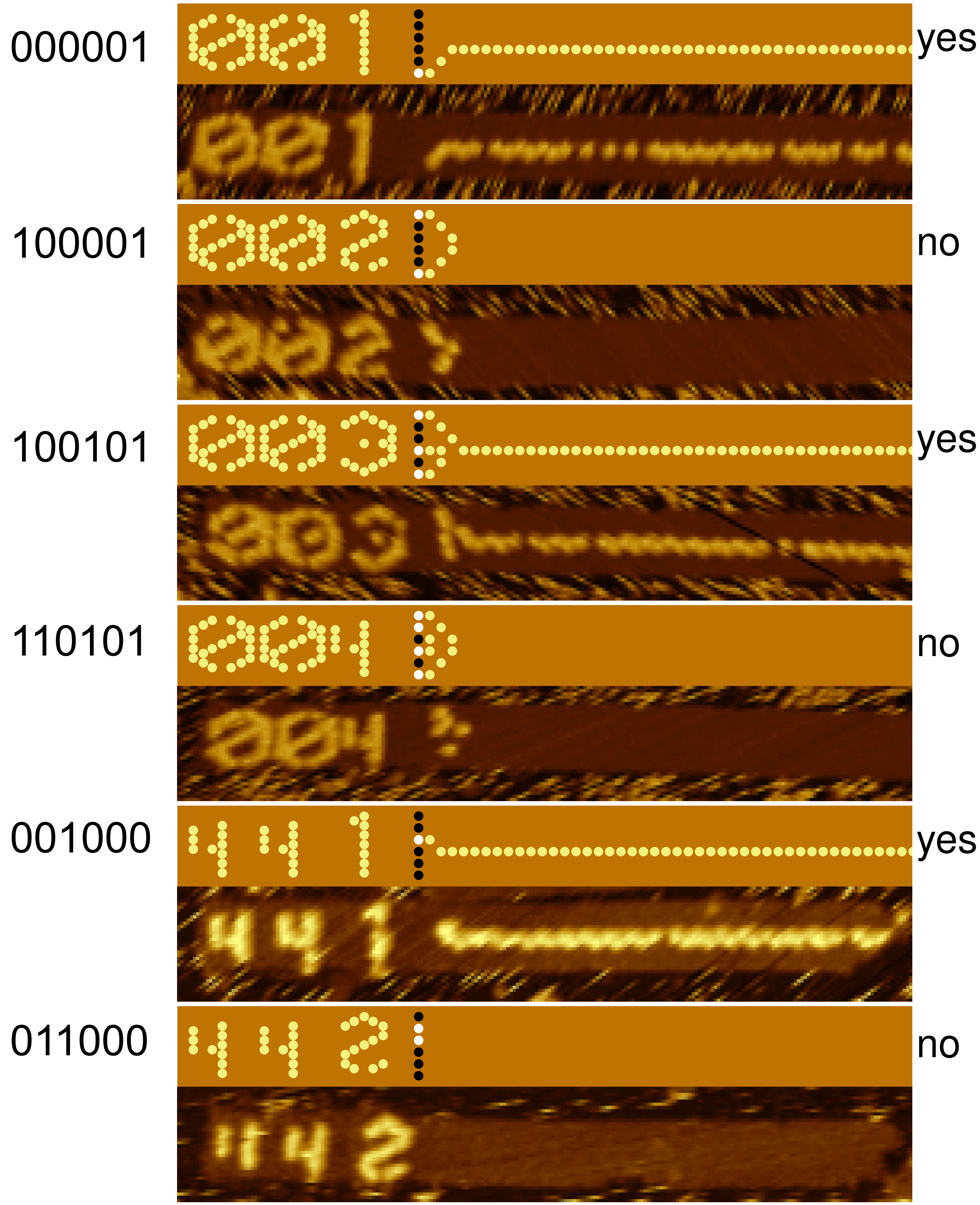
input: 000111, output: 111000

100nm

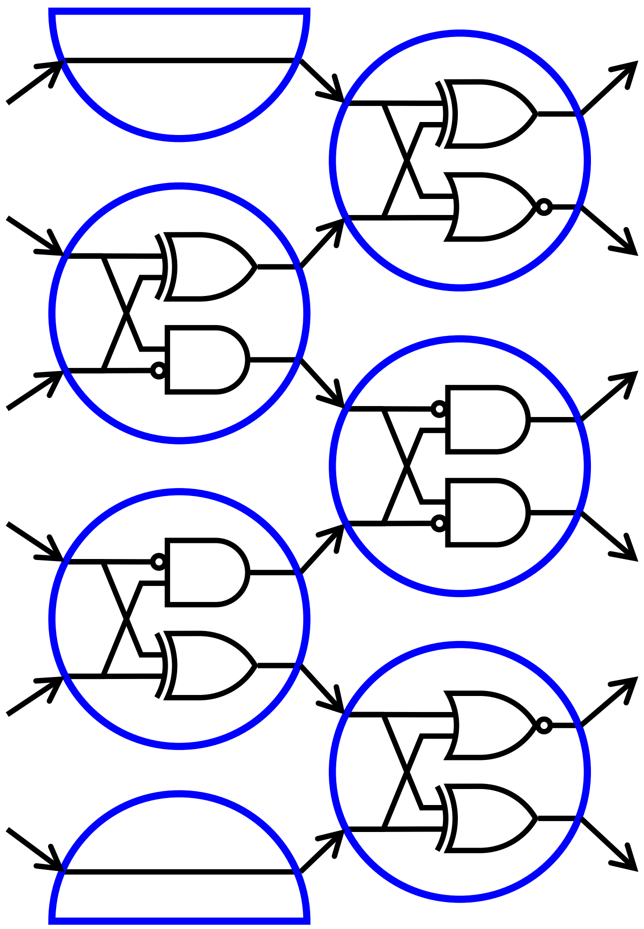


Parity: is the number of 1s odd?





Is the input a multiple of 3?



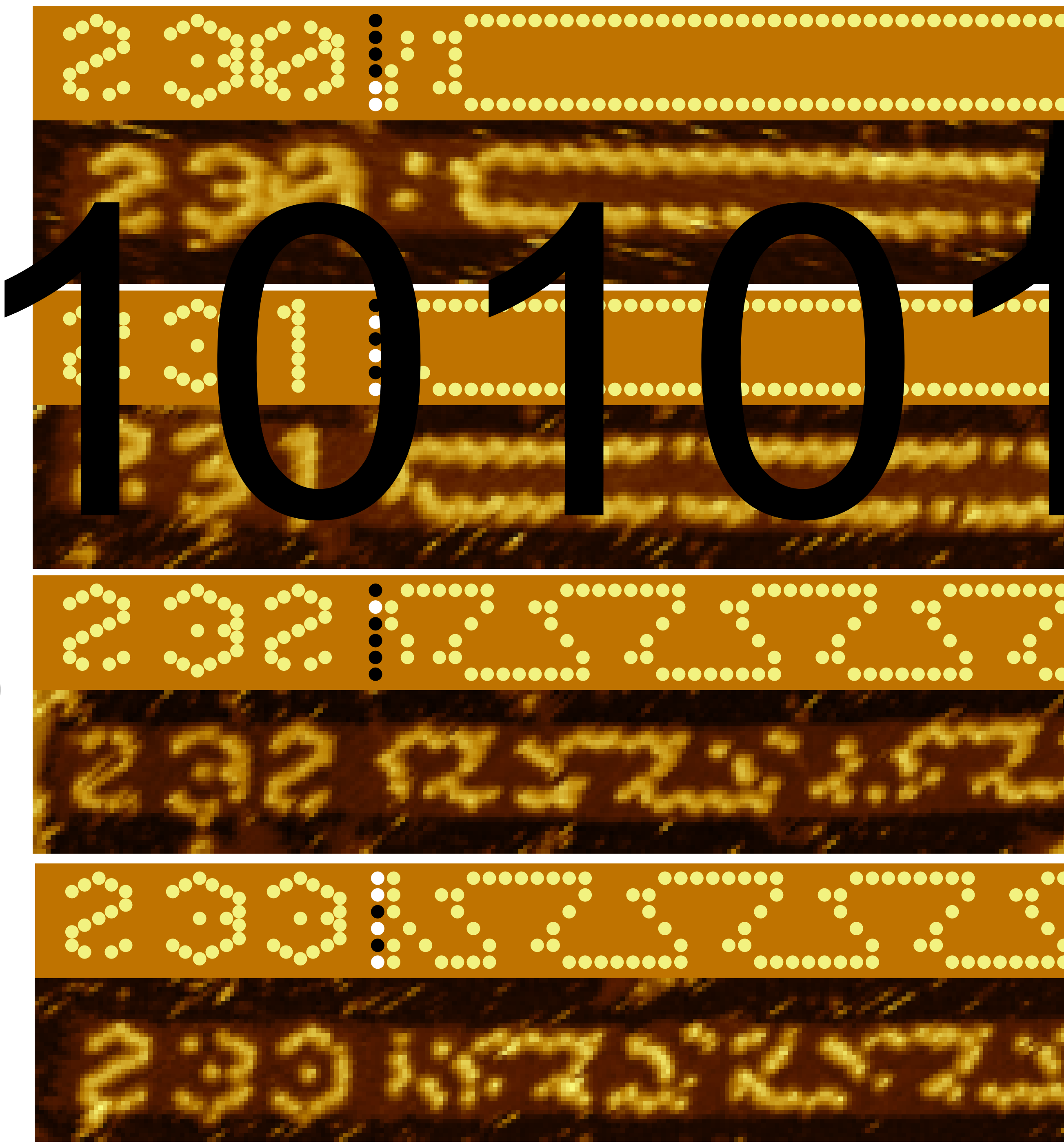
000011
=3

010101
=21

010000
=16

=53

110101



yes

yes

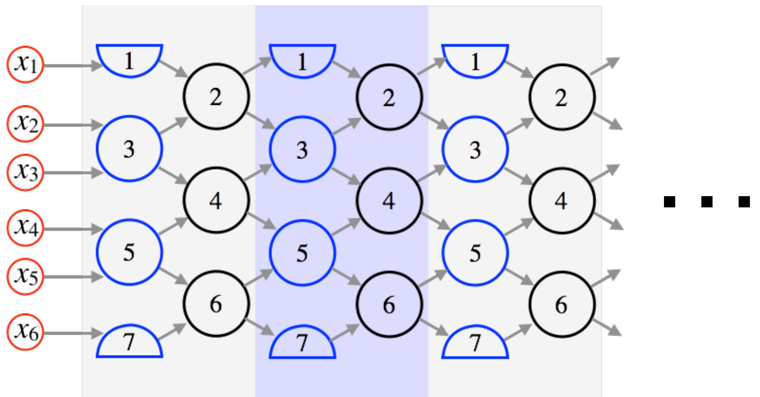
no

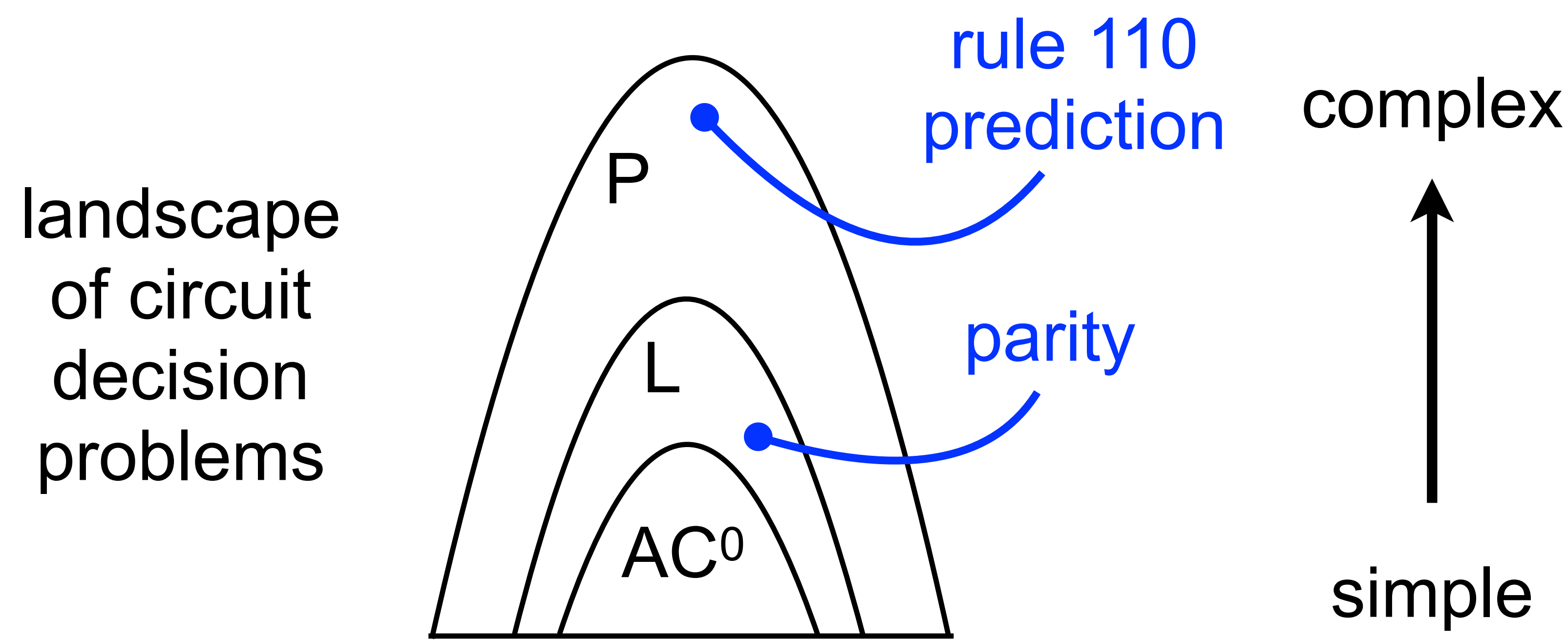
no

Erik Winfree

Computational power of this model?

The model is a rather restricted circuit model: “depth 2 layer”, restricted wiring within layer, repeated-layer, 0/1 signals on the wires. What can it compute?





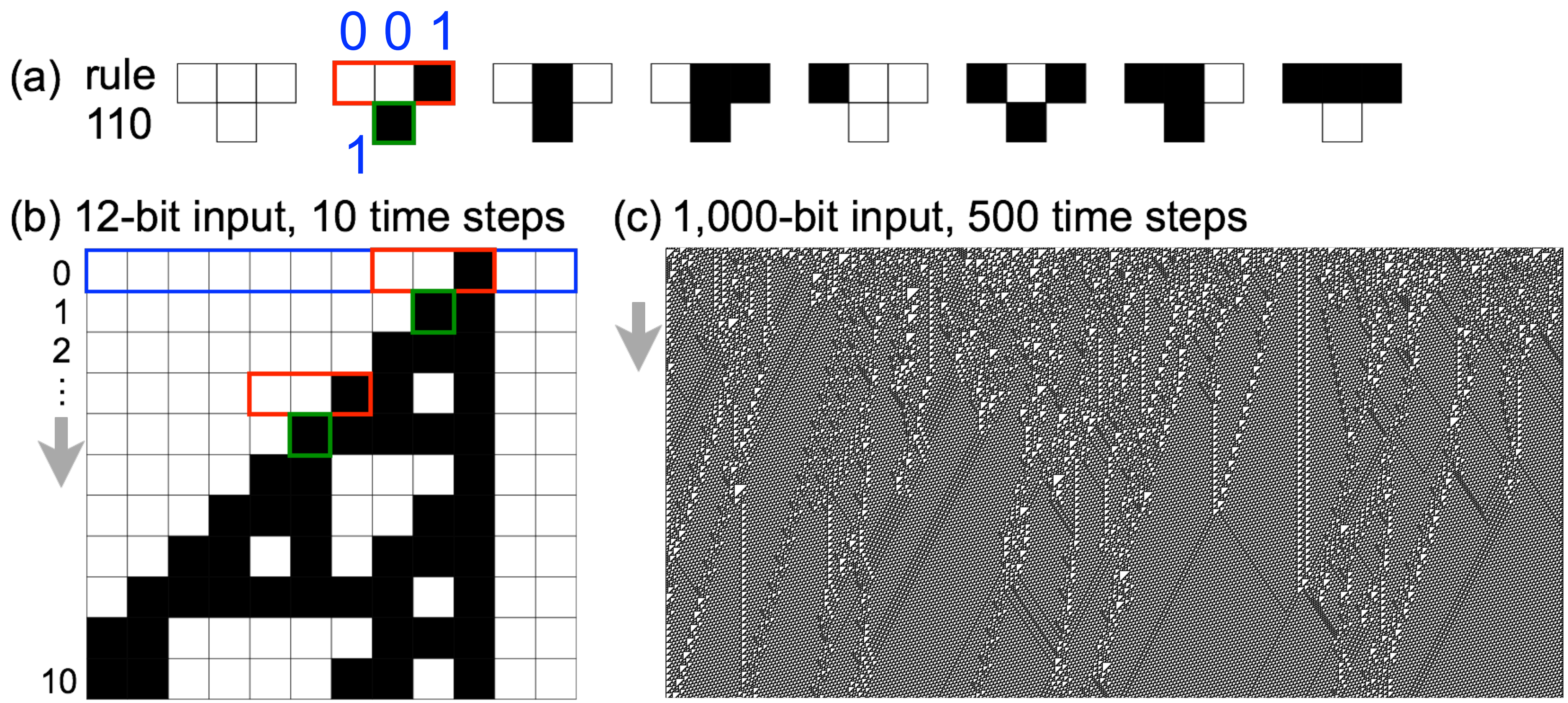
IBCs can do something outside AC^0 (via parity)

All of P (via simulation of rule 110)

Just as powerful as arbitrary Boolean circuits

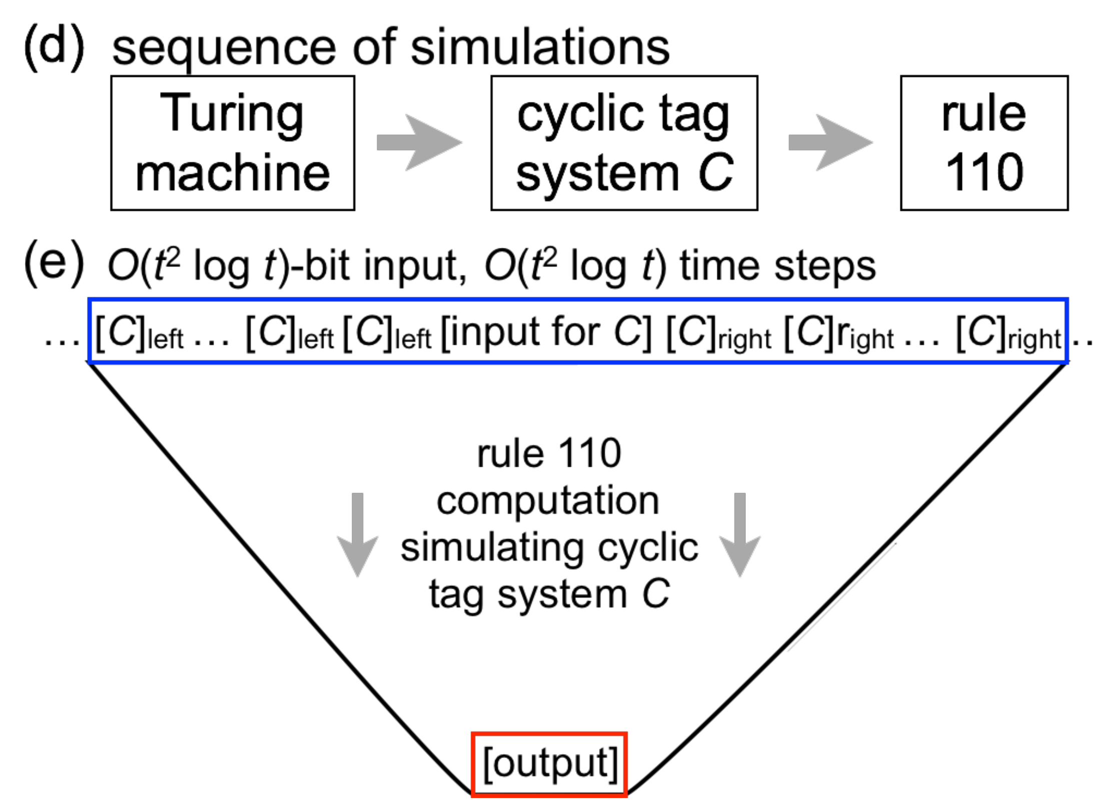
Classes of problems, solved by:
- AC^0: constant depth, poly size, Boolean circuits with arbitrary fanin gates
- L: deterministic log space Turing machines
- P: deterministic polynomial time Turing machines

Rule 110

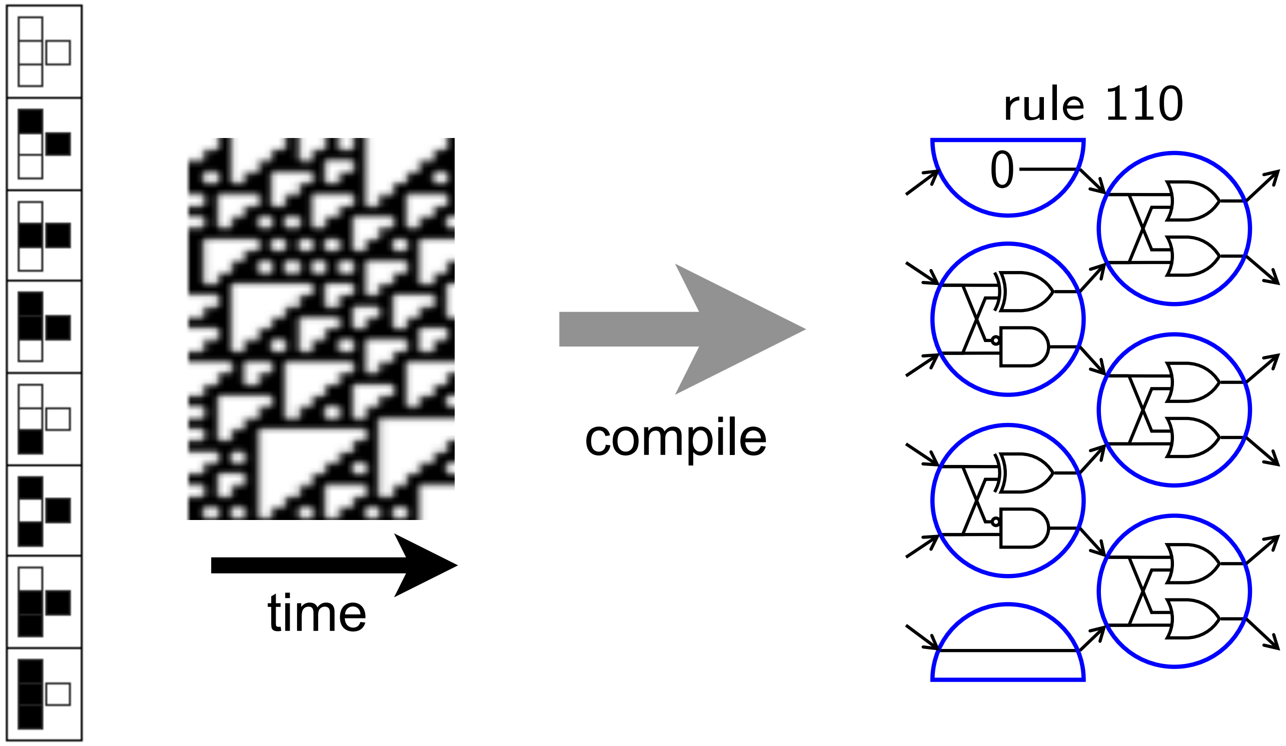


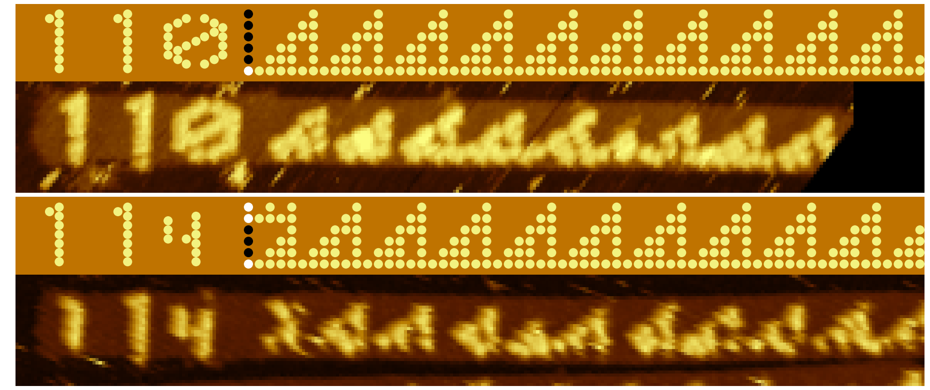
Theorem: Let M be a Turing machine that runs in time t, rule 110 simulates M in $O(t^2 \log t)$ steps

[Cook 2004]
[Neary, Woods, 2006]
[Neary, PhD thesis]

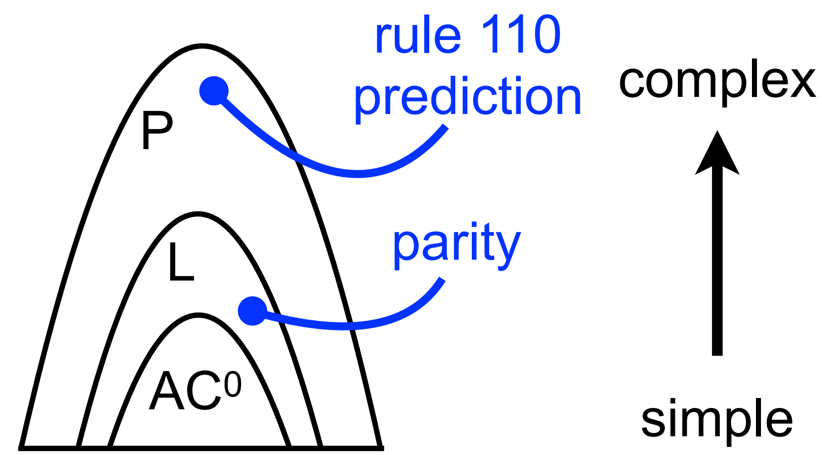


Rule110 circuit: simulation of cellular automata





landscape of circuit decision problems

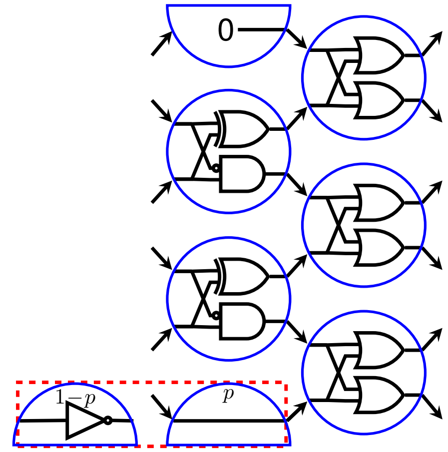


Theorem: Let M be a single-tape Turing machine that runs in time t, then $O(t^2 \log t)$-bit 1-layer circuits (IBCs) simulate M

IBCs efficiently simulate any algorithm

[Cook 2004]
[Neary, Woods, 2006]
[Neary, PhD thesis]

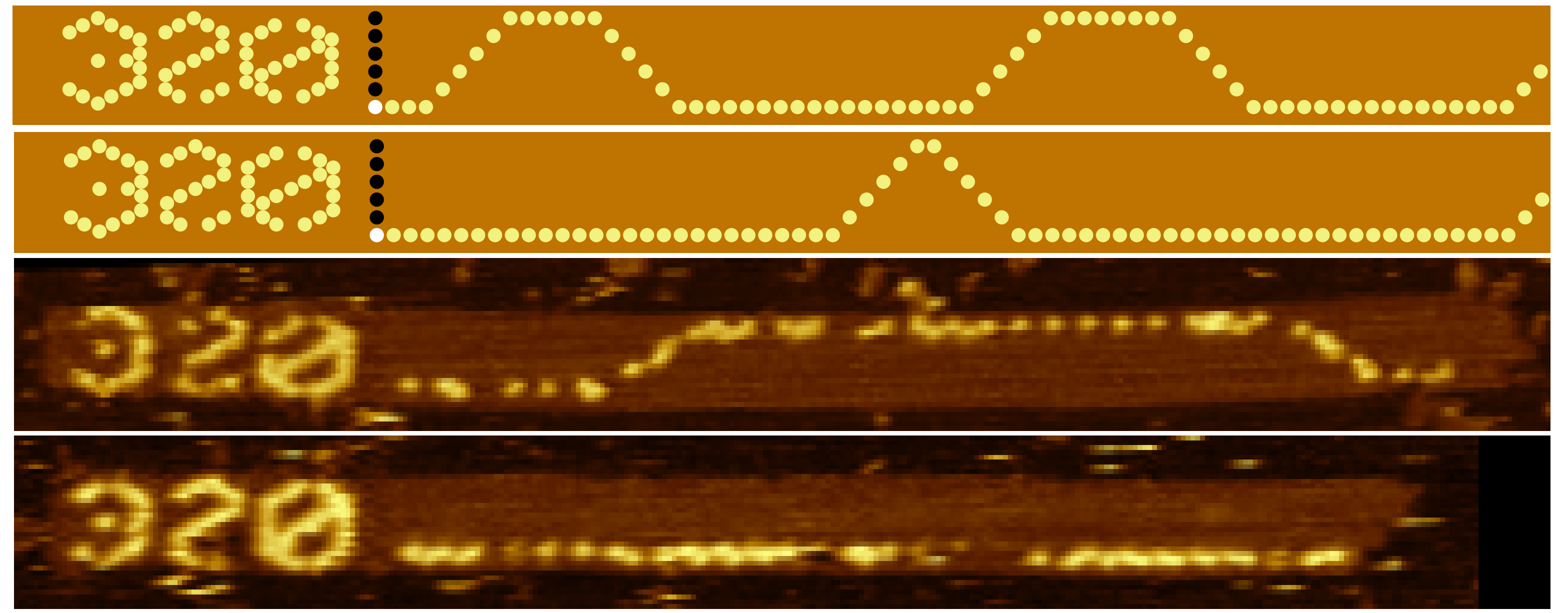
rule 110, randomised



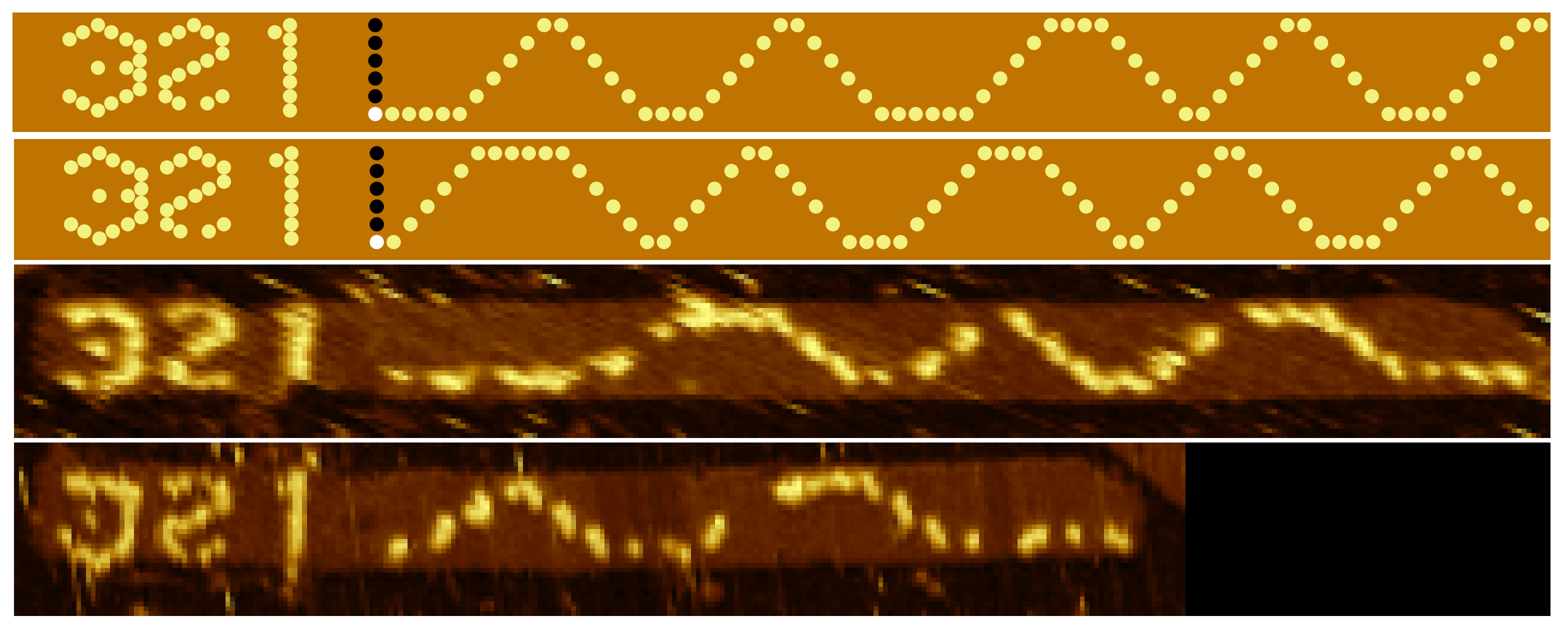
Open: characterise power of randomised model

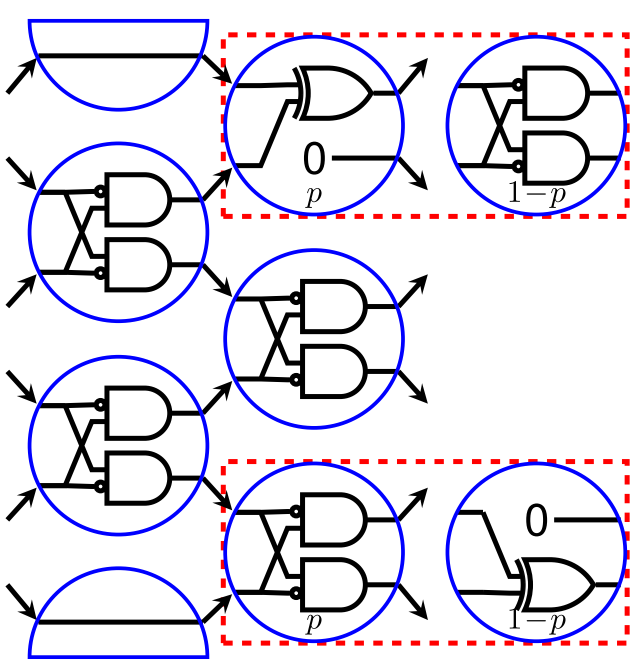
California surf: Waves

Pr[create wave]=0.1
Pr[crash wave] =0.5



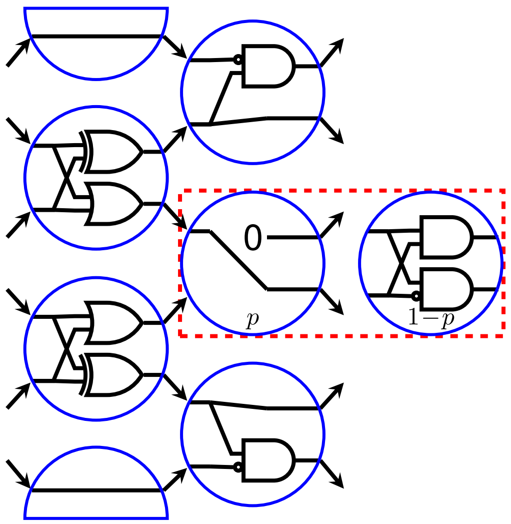
Pr[create wave]=0.5
Pr[crash wave] =0.5

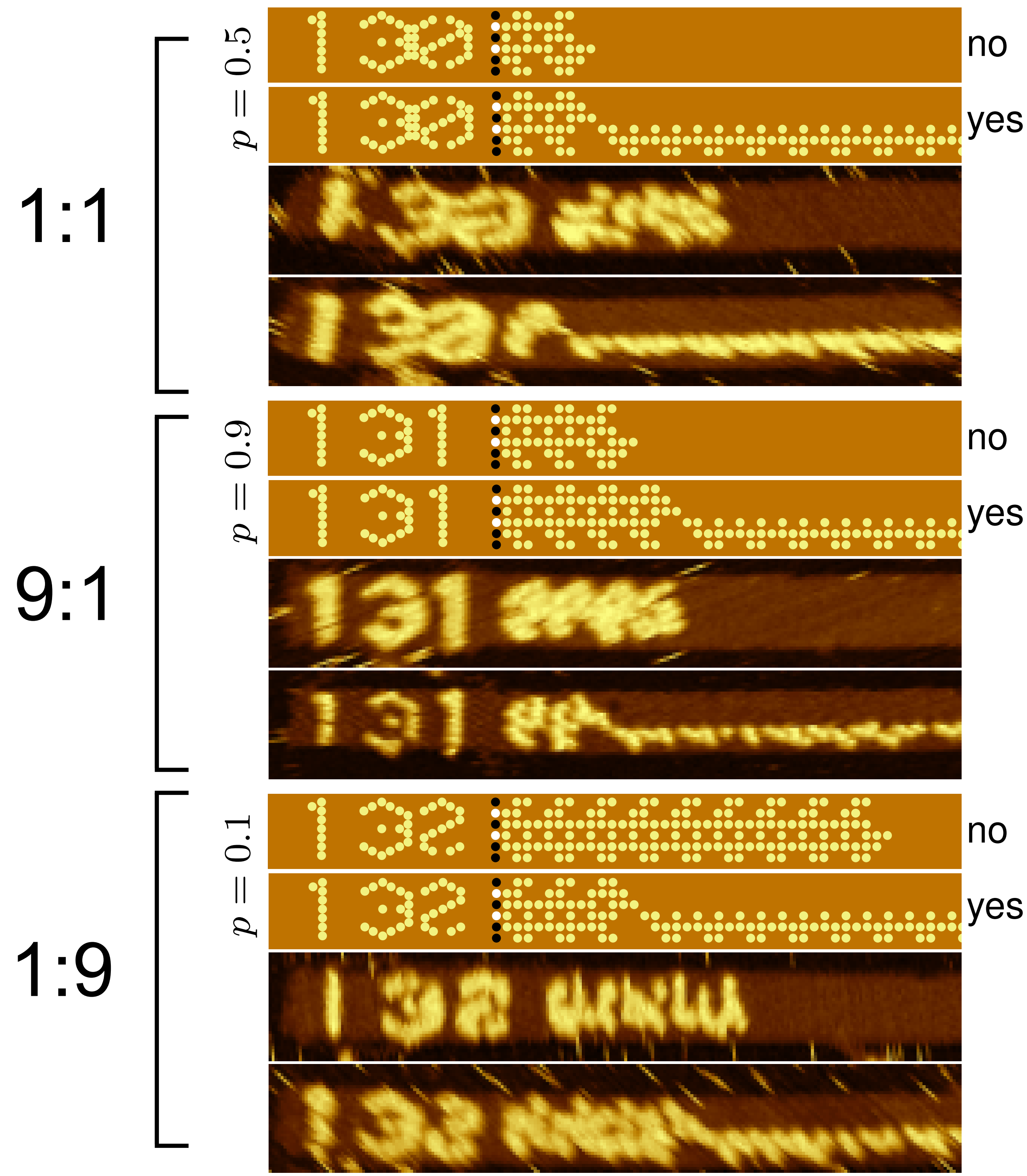




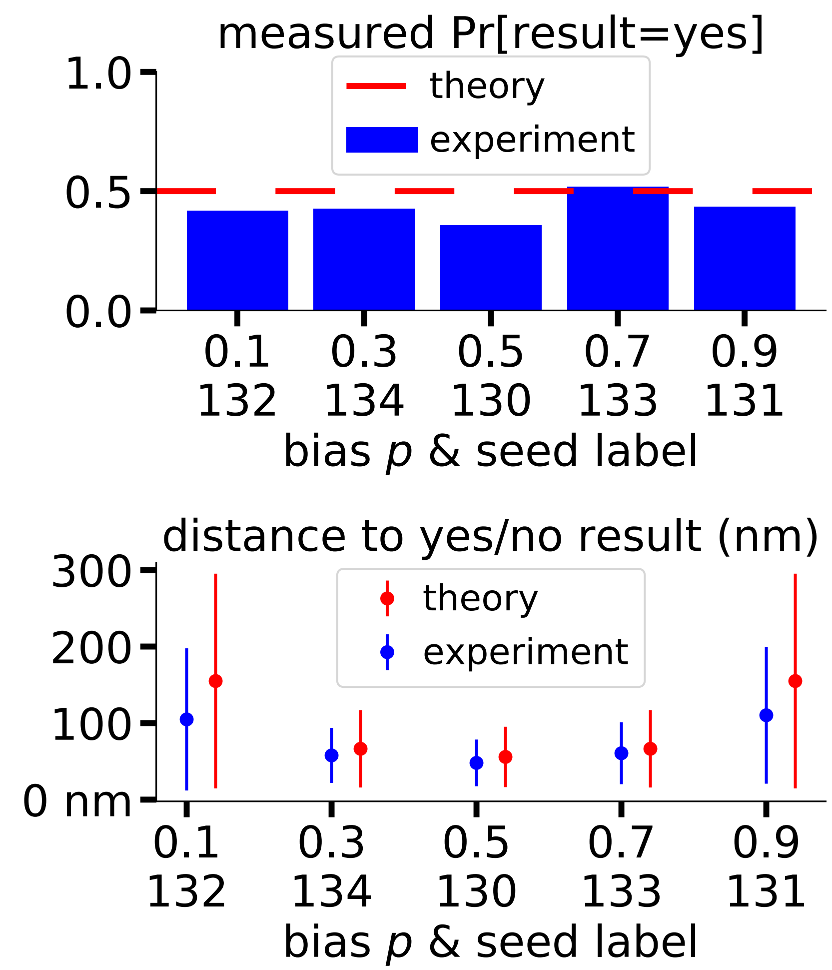
FairCoin: Unbiased bit from biased coin

fair coin





heads:tails

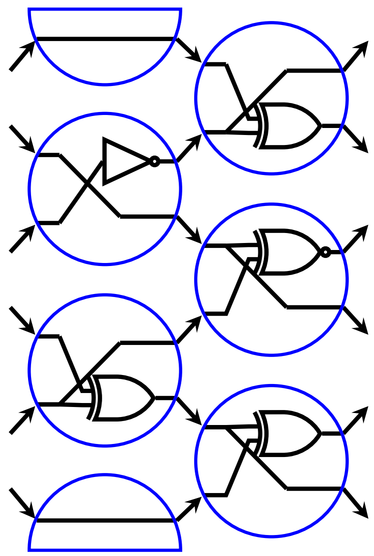


Dave Doty

von Neumann 1951; Chalk, Fu, Martinez, Schweller, Wylie. 2017

Counting to 63

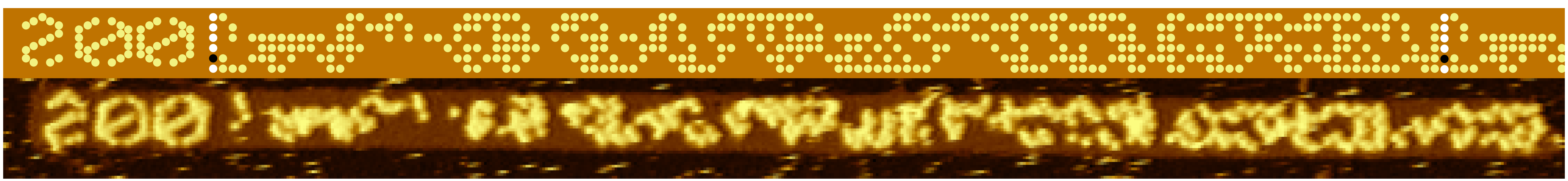
cycle 63



Circuit with 63 distinct strings

Erik Winfree

1 2 3 …   …62 63 1 2 …

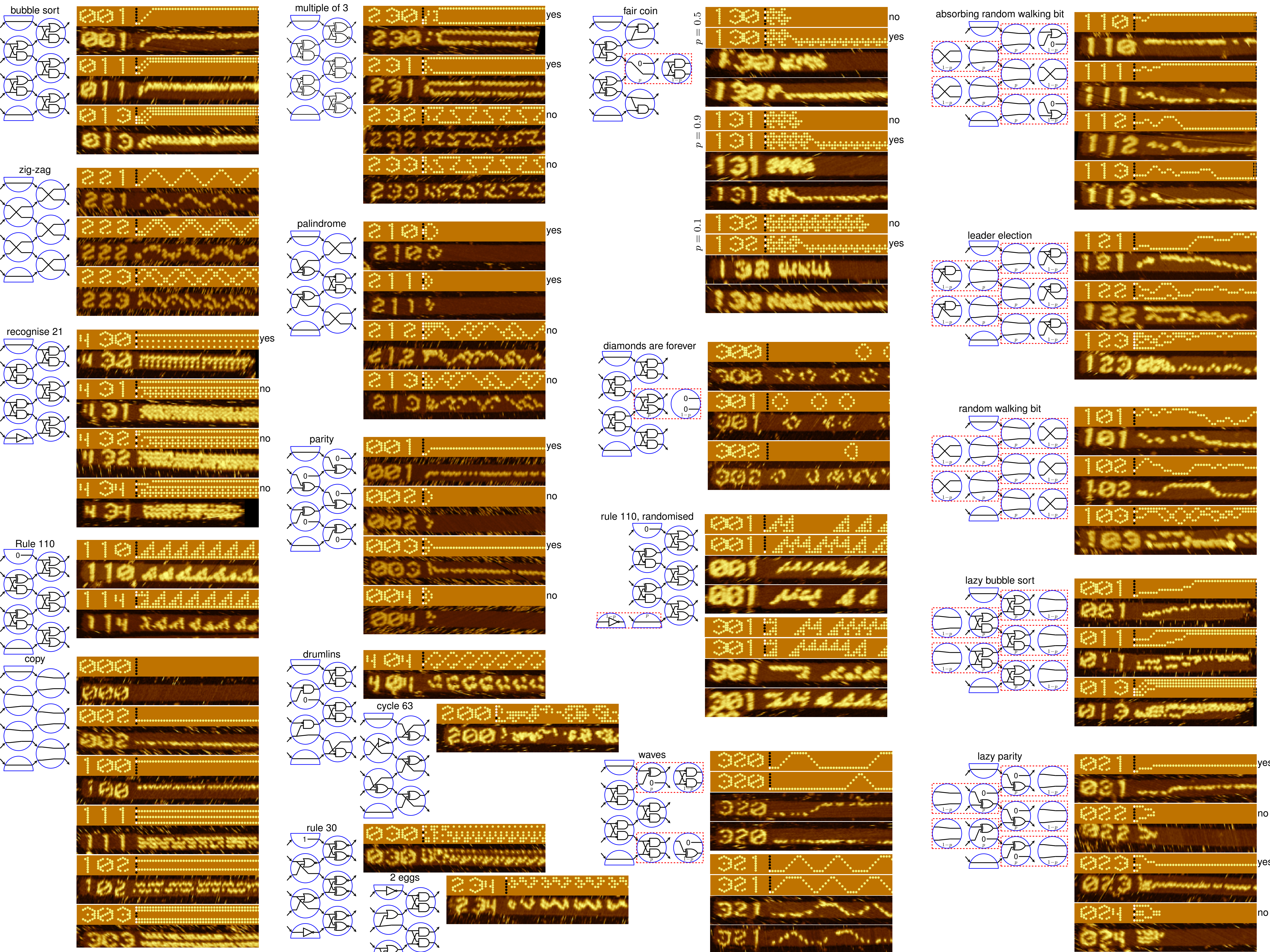


Is there a 64-counter?

No!
Proof by Tristan Stérin

Tristan Stérin

Stérin, Woods
Limitations on counting in Boolean circuits and self-assembly
arXiv:2005.13581

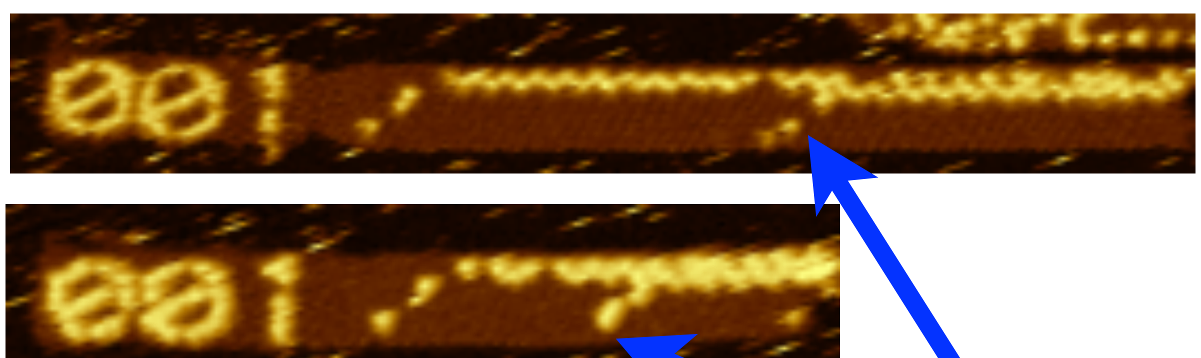


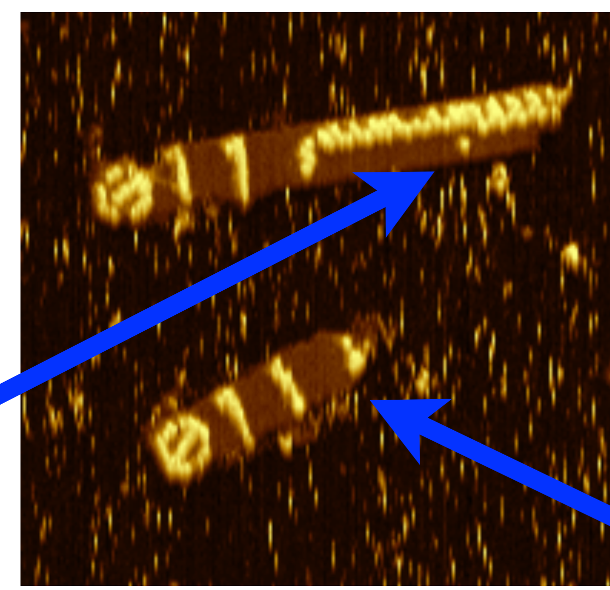
bubble sort
zig-zag
recognise 21
Rule 110
copy
multiple of 3
palindrome
parity
drumlins
cycle 63
rule 30
2 eggs
fair coin
$p = 0.5$
$p = 0.9$
$p = 0.1$
diamonds are forever
rule 110, randomised
waves
absorbing random walking bit
leader election
random walking bit
lazy bubble sort
lazy parity

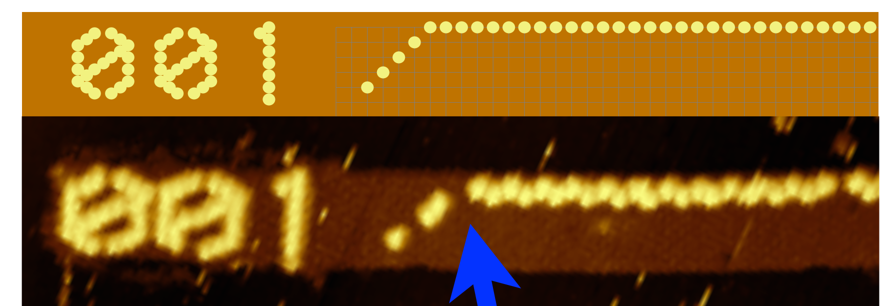
How well did the 21 circuits work?

Extensive testing of all 355 tiles:
- **every tile type** was used in some circuit
- for many circuits **tested all tile types for that circuit**
- ran one circuit on **all 64 inputs**

Analysed ~12k nanotubes with ~5M tile attachments:





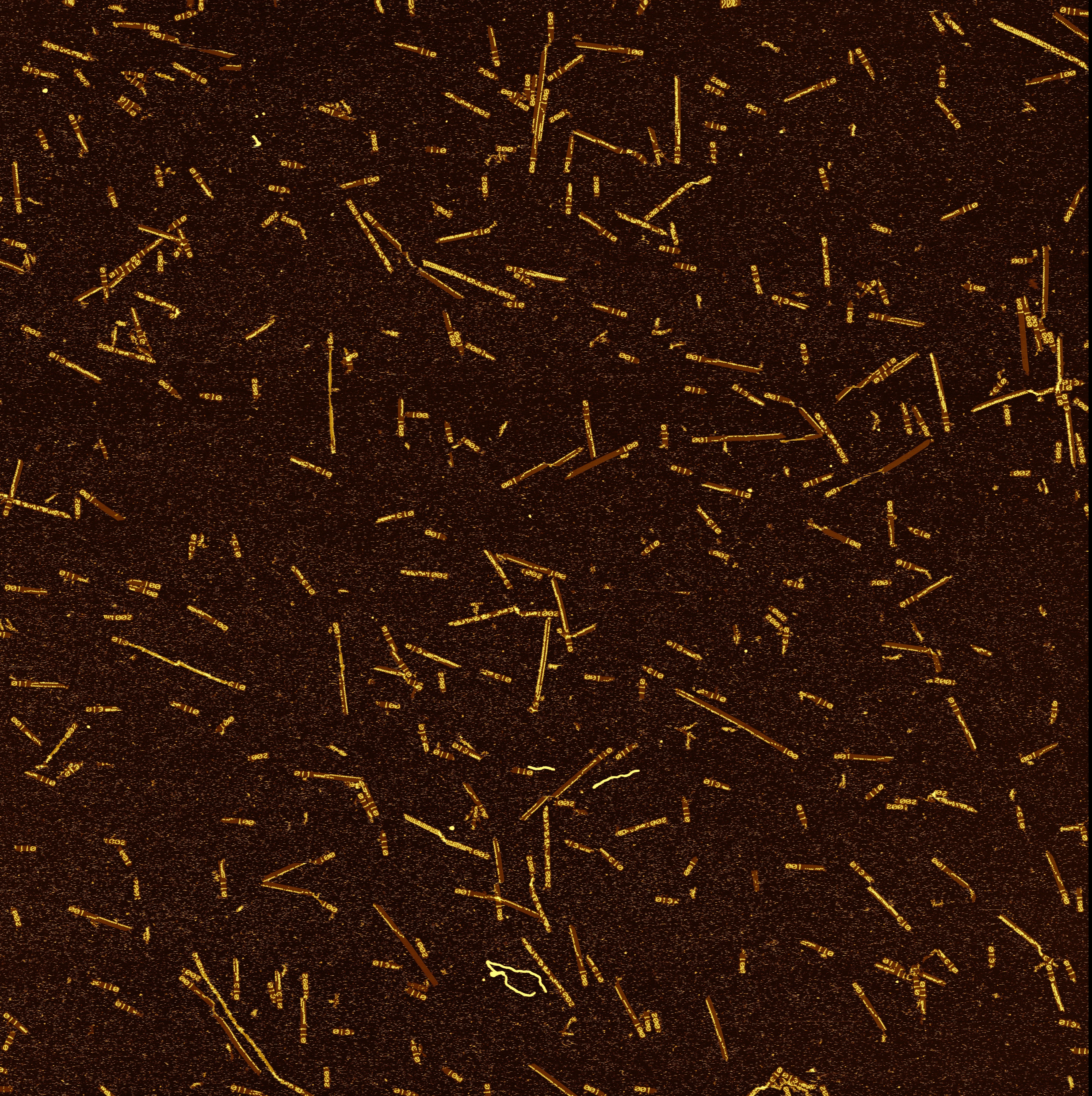


Errors: 0.03% of tile attachments were incorrect (i.e. error rate = 1/3000; 1,419 errors out of 4.6M attachments)

38.9% of seeds had no growth

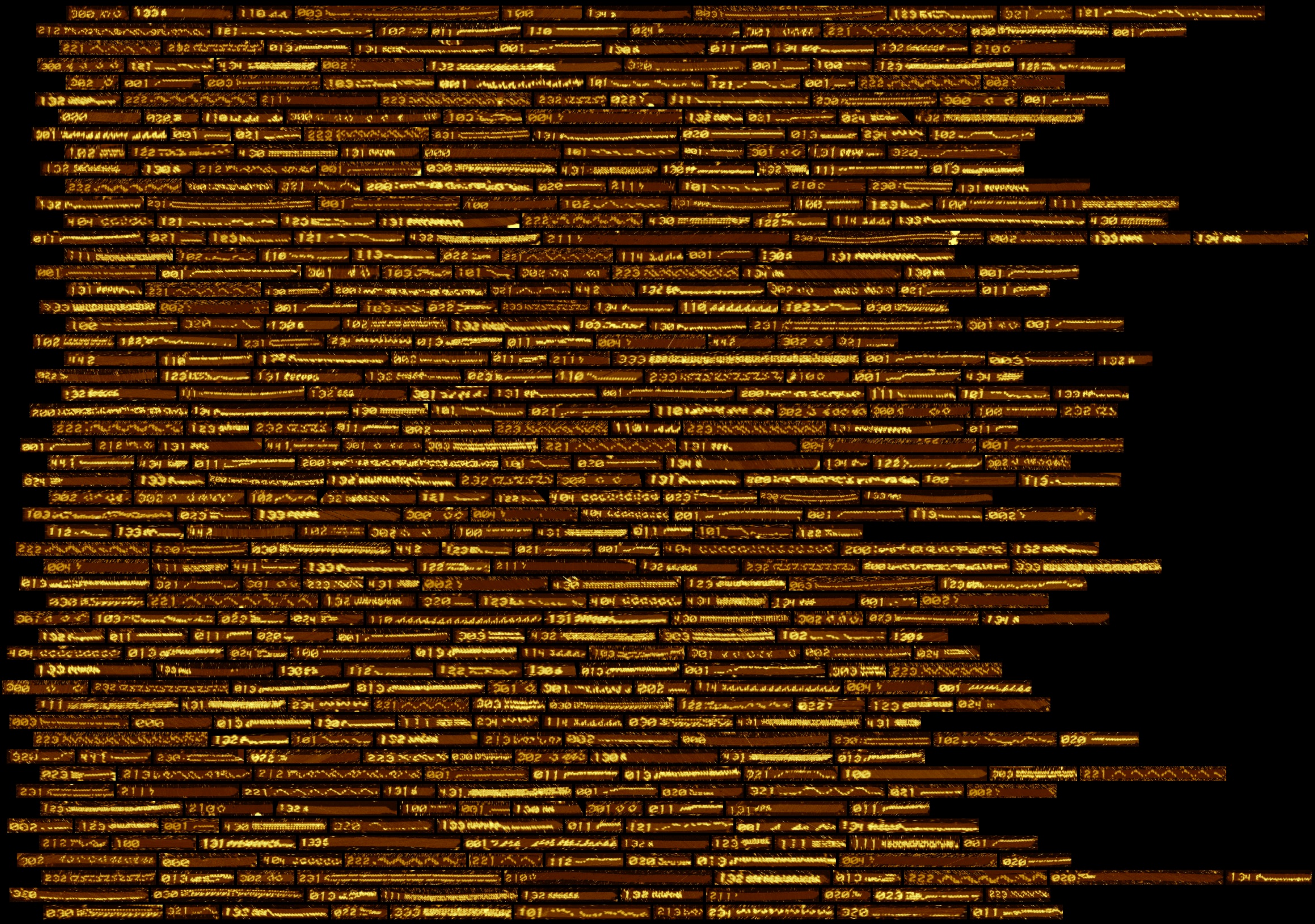
imperfect streptavidin labelling

Reprogrammable: demonstrated many new self-assembly programs
Scaling up: 15x more tile types than previous algorithmic self-assembly systems
Low error: Careful sequence design; Proofreading
Good structure: Nanotube lattice & hardcoded rows
Lots of tile types: Long SST domains

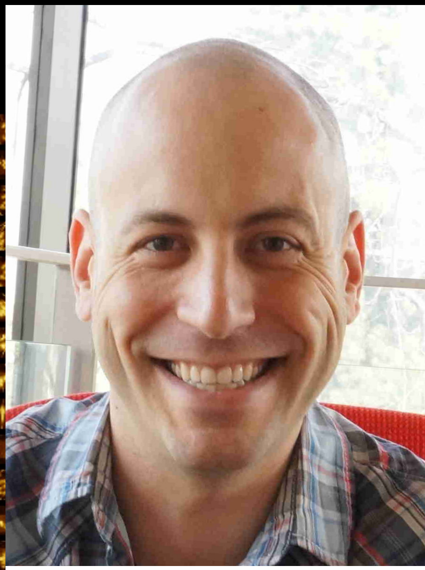


raw data
8μm x 8μm

A flying carpet of algorithms



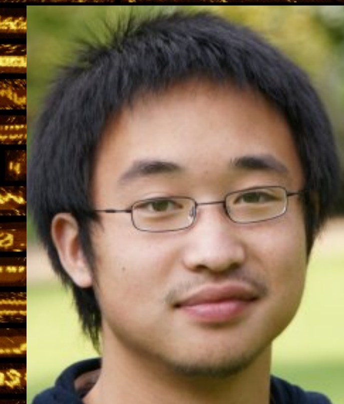
Acknowledgements

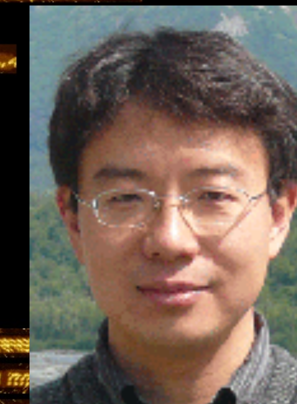

Dave Doty
UC Davis


Erik Winfree
Caltech


C Myhrvold
Harvard


Joy Hui
Harvard


Felix Zhou
Oxford


Peng Yin
Harvard

Caltech

Inria

UC Davis

Harvard

Maynooth University
National University of Ireland Maynooth

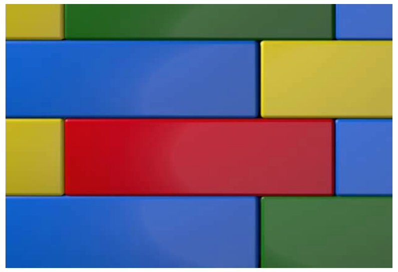
Woods*, Doty*, Myhrvold, Hui, Zhou, Yin, Winfree. *Diverse and robust molecular algorithms using reprogrammable DNA self-assembly* **Nature**. 567:366-372. 2019 *Joint lead co-authors

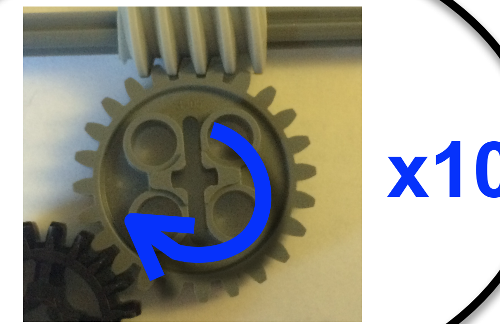
Special thanks to: Constantine Evans, Ashwin Gopinath, Bryan Wei, Cody Geary, Rebecca Schulman, Sungwook Woo, Paul Rothemund, Yannick Rondelez, Rob Barish, Rizal Hariadi, Cris Moore, Tristan Stérin, Chris Thachuk, P.-É. Meunier, Lulu Qian, Greg Tikhomirov, Philip Petersen.

NSF

NASA

Whats next?






```
if top == (blue AND yellow):
  bottom_left := blue
  bottom_right := green
elif top == (blue AND green):
  bottom_left := yellow
  ...
```

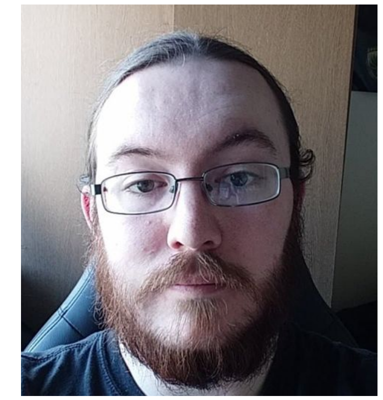
Molecular computing at Maynooth University

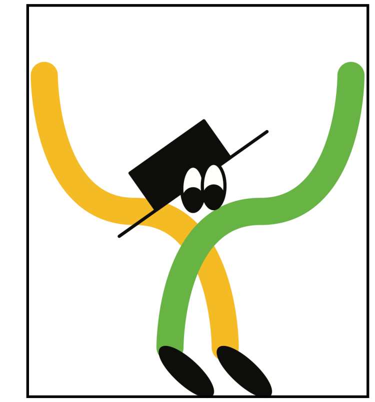

Damien Woods


Constantine Evans


Trent Rogers


Tristan Stérin


Cai Wood


you?

Maynooth University
National University of Ireland Maynooth

Hamilton Institute

We're hiring!

Postdoc, PhD, Professor!

See: dna.hamilton.ie

erc
Active-DNA

sfi Science Foundation Ireland

Fin