Algorithmic Self-Assembly of Circuits

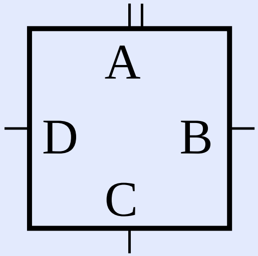
Michael deLorimier
Alexandre Mathy
Dustin Reishus
Rolfe Schmidt
Bilal Shaw
Li Chin Wong

Overview

- Introduction
  - $T = 2$ Model of Self-Assembly
  - Cellular Automata Model of Self-Assembly
  - Self-Assembled Circuits
- Fast Fourier Transform
  - FFT Networks
- Self-Assembly of FFT Networks
  - CA Rules
  - Particles and Collisions
  - Wiring and Logic
- $T = 2$ Tile System for FFT
- Error Correction
- What's next?

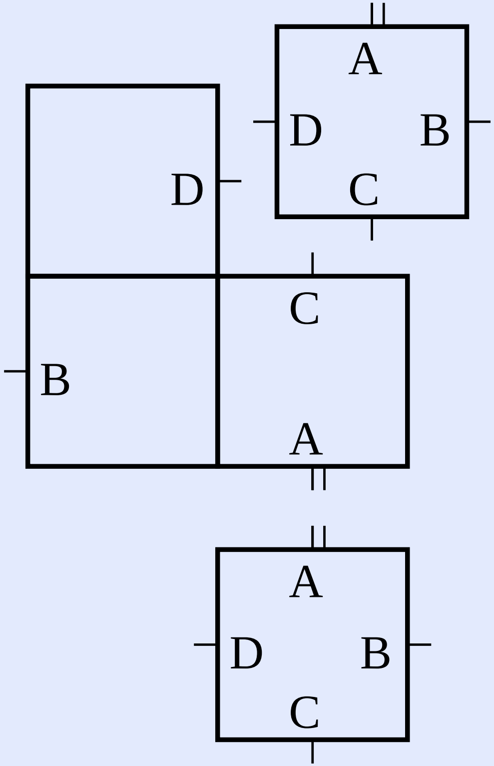
$T = 2$ Model of Self-Assembly

- Tiles are non-rotatable squares with "glues" on each side.



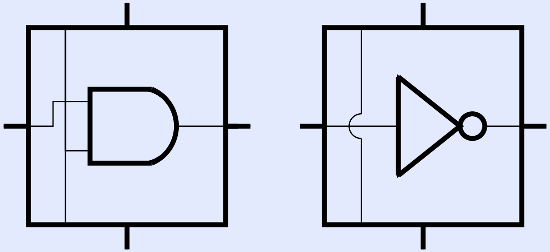
- Each glue has a strength. A tile can stick if it can form one strength 2 bond or two strength 1 bonds.

$T = 2$ Model of Self-Assembly

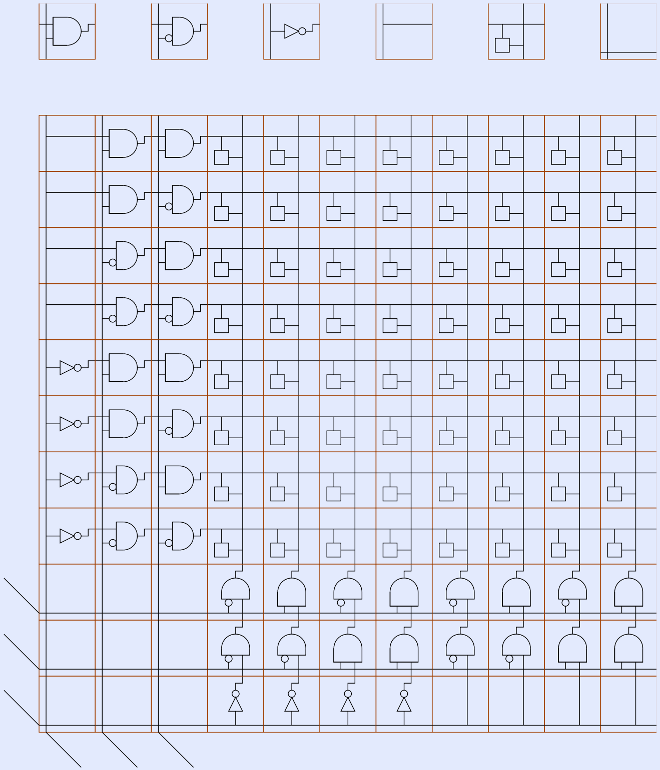


Self-Assembled Circuits

- It may be possible to attach wires and gates directly to the top of the tiles. The tiles then self-assemble to form a circuit.



Self-Assembled Memory Array



The Fast Fourier Transform

- The Fourier transform is useful in many applications. For computations we always us a discrete Fourier transform.

$$\hat{f}(\xi) = \sum_{x=0}^{N} f(x) e^{-2\pi i x \xi / N} \qquad \xi = 0, \dots N$$

- A straightforward implementation would require $O(N^2)$ operations, but we can divide and conquer to do much better. Notice that

$$\hat{f}(\xi) = \hat{f}_{\text{even}}(\xi) + e^{-2\pi i \xi / N} \hat{f}_{\text{odd}}(\xi)$$

  when $\xi < N/2$ and

$$\hat{f}(\xi) = \hat{f}_{\text{even}}(\xi - N/2) - e^{-2\pi i \xi / N} \hat{f}_{\text{odd}}(\xi - N/2)$$

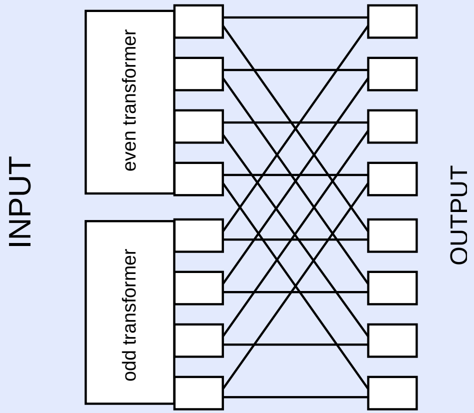
  when $\xi >= N/2$.

- So we can evaluate this recursively. Running time:

$$T(n) = n + 2T(n/2) = \Theta(n \lg n)$$

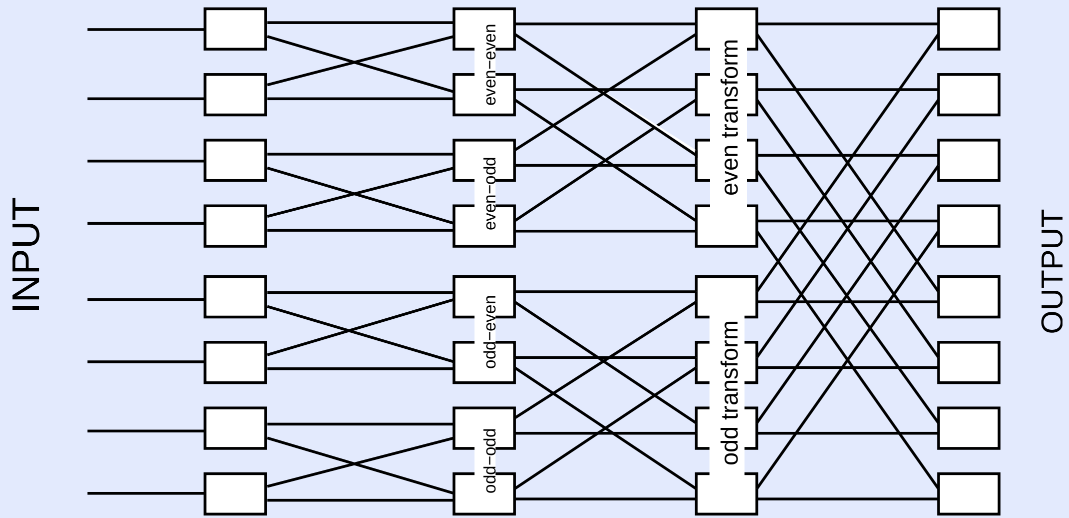
A Fast Fourier Transform Network

- If we had a networks that compute $\hat{f}_{\text{even}}$ and $\hat{f}_{\text{odd}}$, then it is easy to build a network for the full Fourier transform:



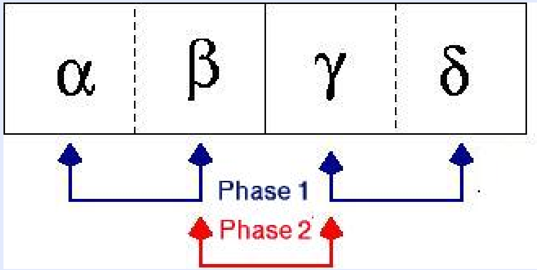
A Fast Fourier Transform Network

So we build the network recursively:



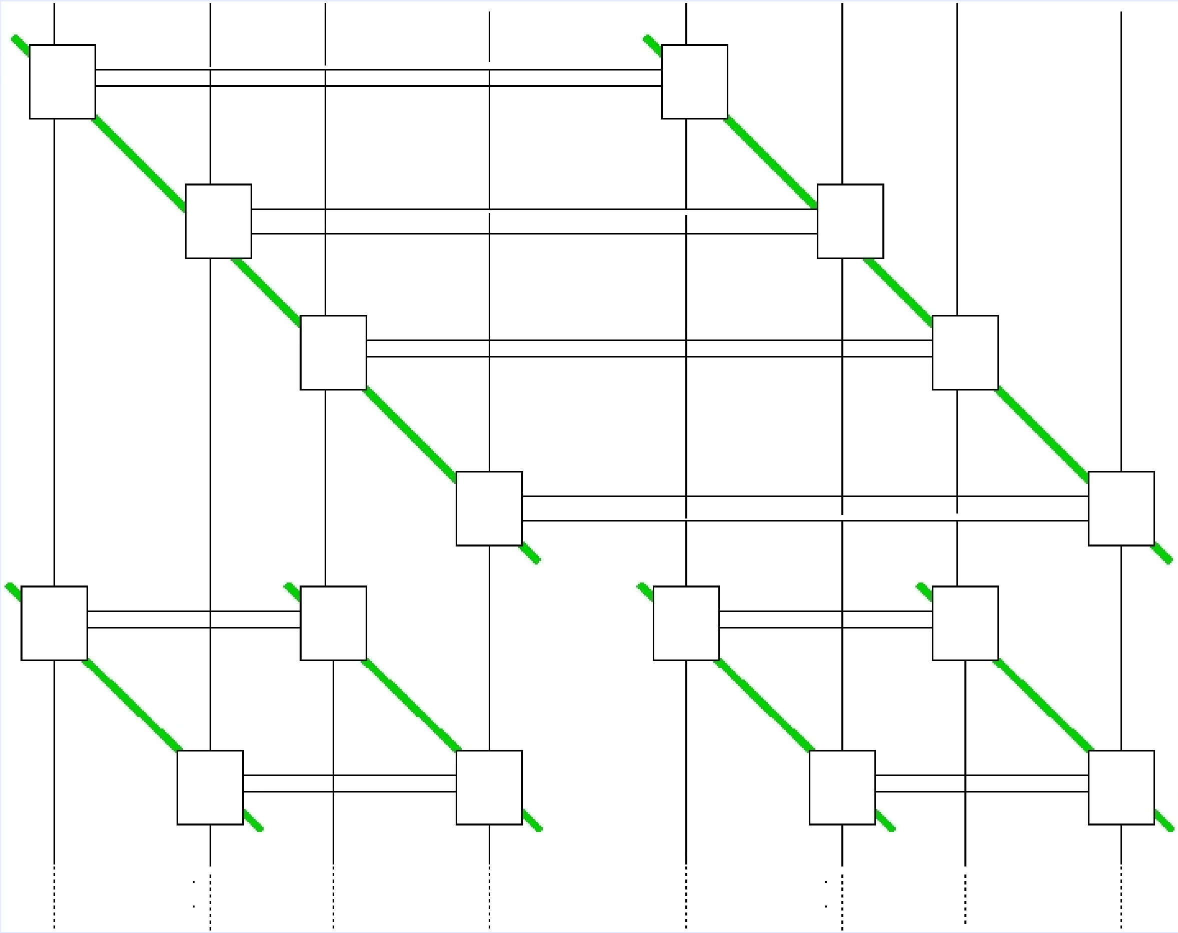
CA rules for building an FFT network

- 1D cellular automaton
- Margolus Neighborhood

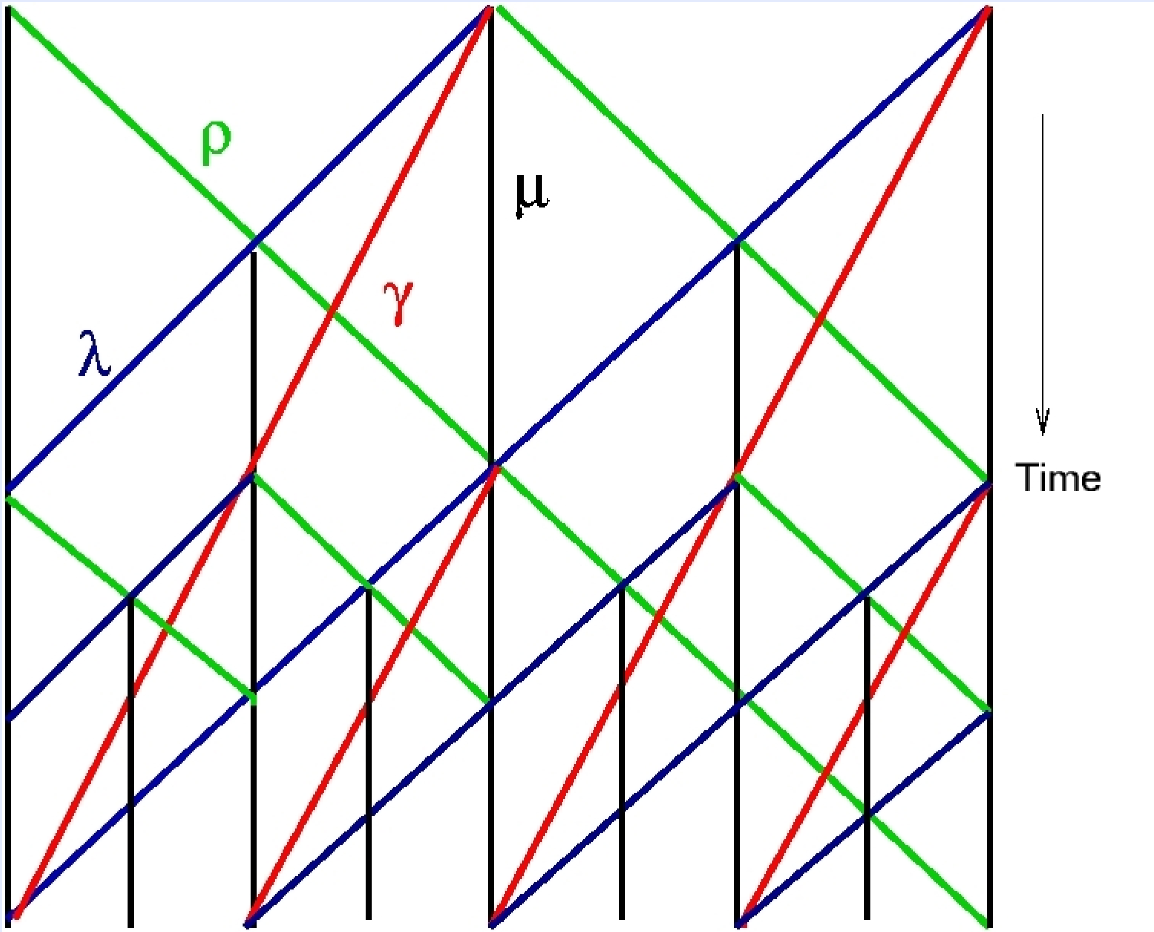


- Design in terms of *particles* and *collisions*

FFT Layout

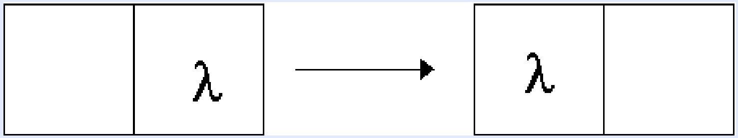


CA collision map

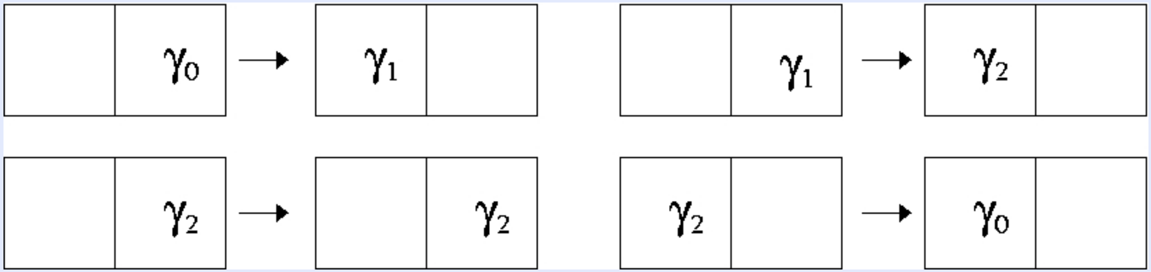


Particles

- A left moving particle with unit speed.

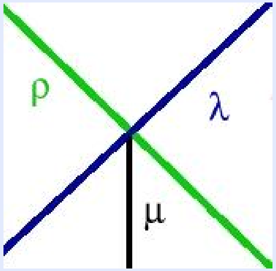


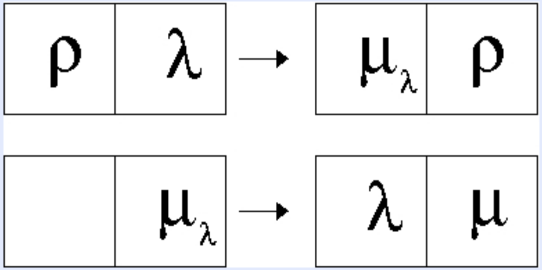
- A left moving particle with half speed.



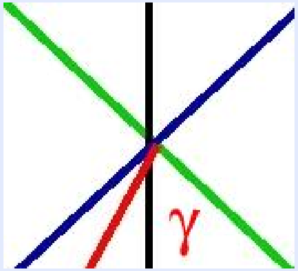
Collisions

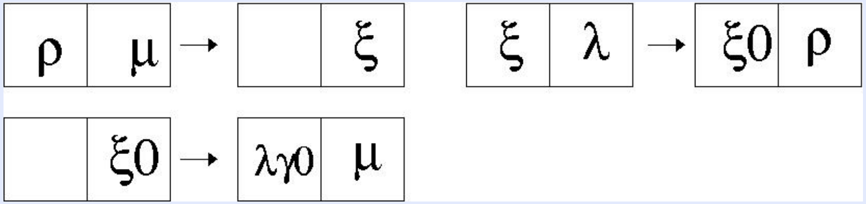
- λ and ρ





- λ and ρ hit a μ





Home Page
Title Page
Page 14 of 25
Go Back
Full Screen
Close
Quit

Other Issues

- Phase between λ and ρ
- Termination
- Number of symbols can grow as a power of logical particles.
- Number of explicit rules can grow as a power of symbols.

Simulating 1D cellular automata with the T=2 tile model

A simple 1D CA

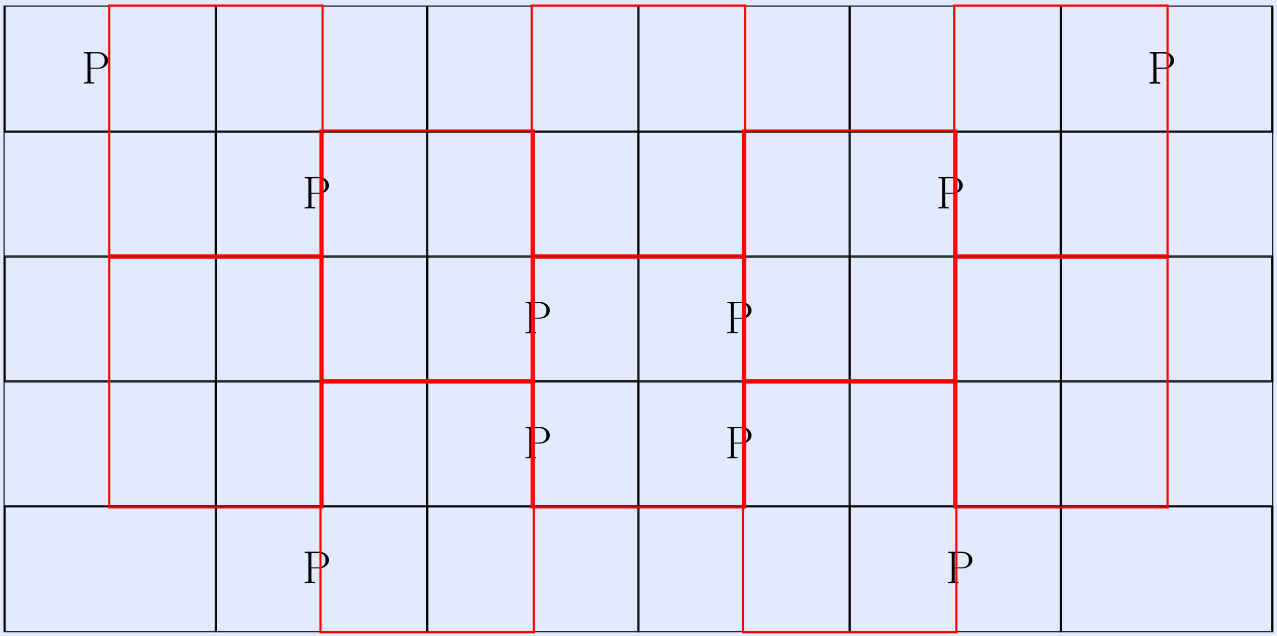
P					P
	P			P	
		P	P		
		P	P		
	P			P	

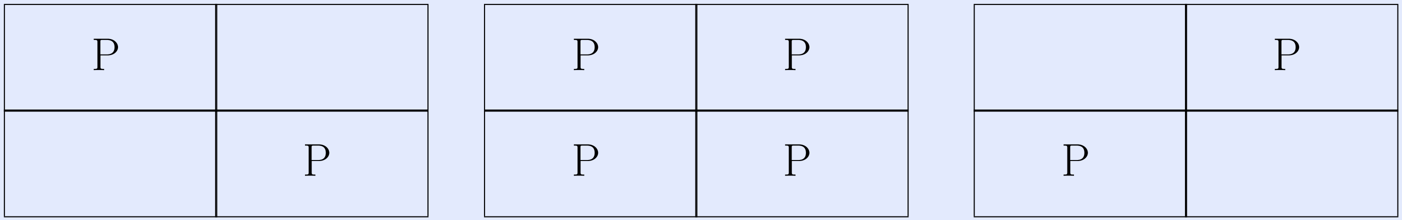
P	
	P

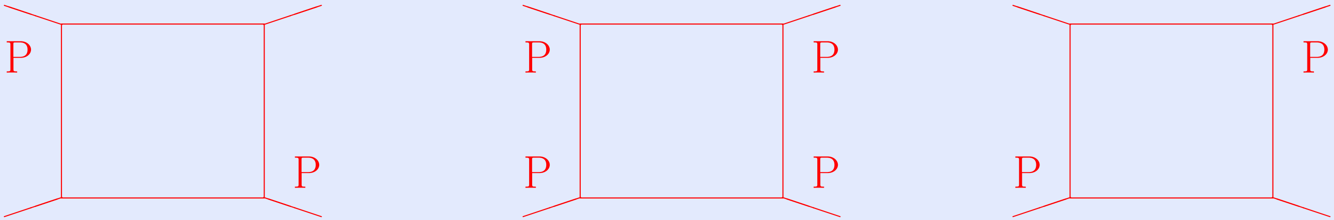
P	P
P	P

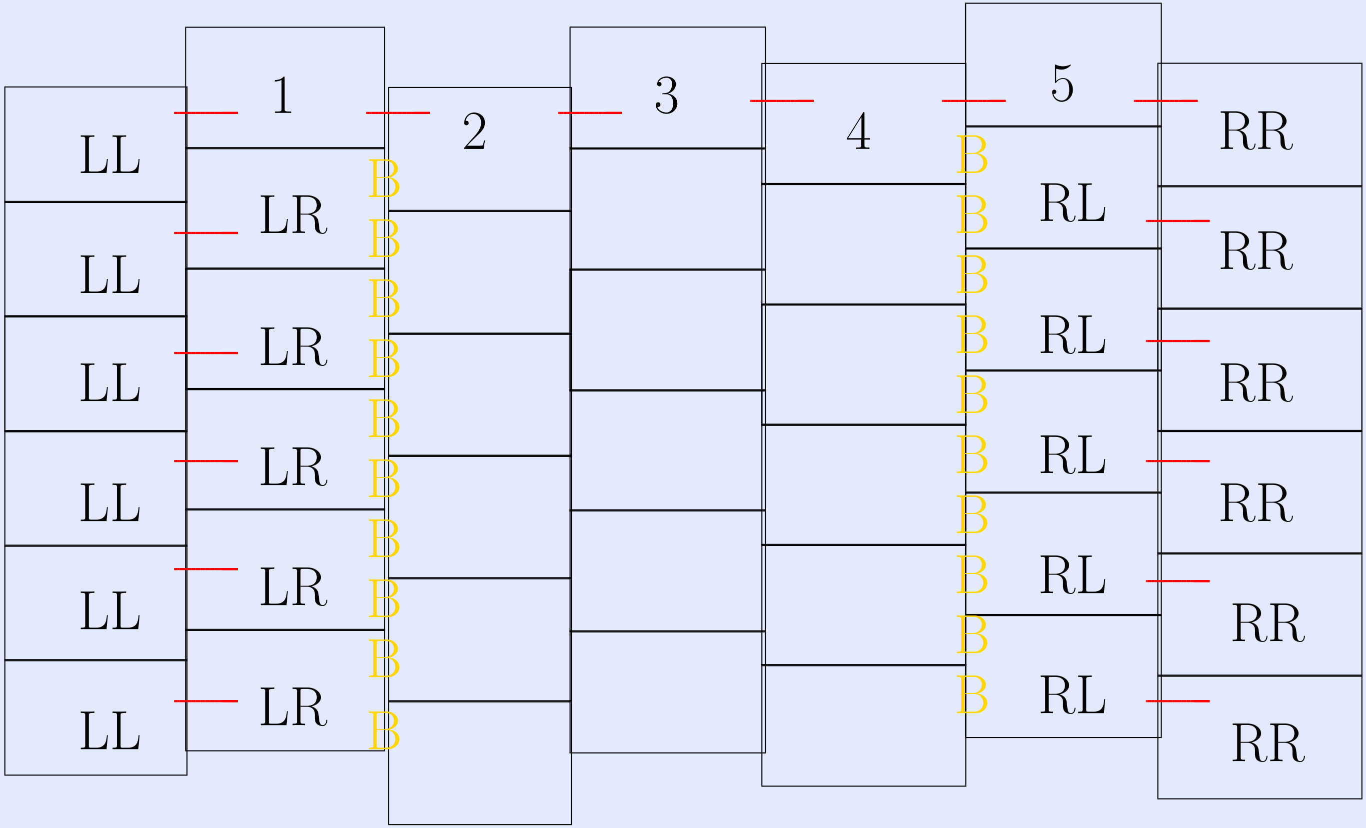
	P
P	

Each tile corresponds to a CA rule:
Each bond corresponds to a CA tile



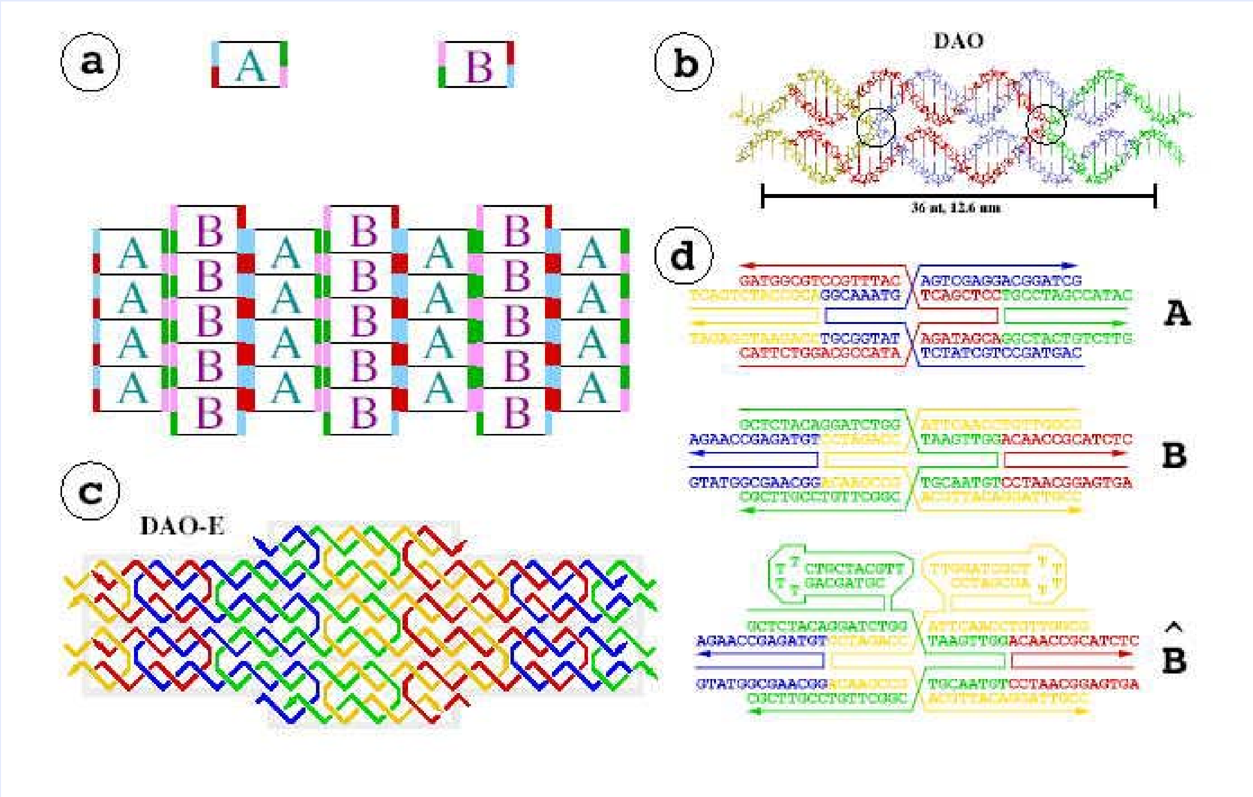






A bounded CA, red lines indicate strength 2 bonds
All other bonds are strength 1
b's indicate boundary bonds
If the BCA has c columns + 2 edge columns then
the tiles have c+1 columns + 4 edge columns.

DNA Tiles



DNA Tile Design

- # of tiles = 73
- # of sticky ends with optimization = 45 + 24 = 69
- # of complementary sticky ends = 69
- # of DNA to synthesize = 2 + 12*4 + 45*2 = 140
  - ⋆ If it occurs in either N/S, then need one sequence and complementary
  - ⋆ Total of 2 sequences
  - ⋆ If it occurs in both N/S and E/W then require total of 4 sequences

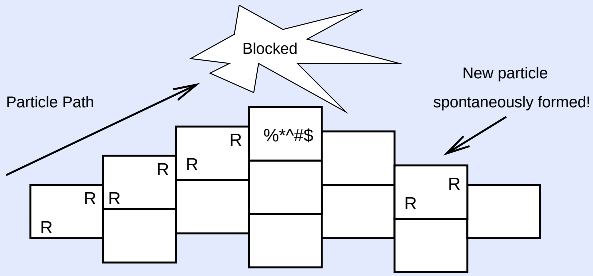
Sequence Design For DNA Tiles

- # of nucleotides for sticky end = 6
- Sequence space for
$$6bp = 4^6 = 4096$$
- Sequence space for 6bp, with 50% GC content = 1280
- Sequence complexity considerations:
  - ⋆ No 6bp words that anneal to sticky ends should occur in the tile core
- Potential Technical Problems:
  - ⋆ Stoichiometry: change of concentration as assembly occurs
  - ⋆ Temperature of hybridization: standardized temperature of hybridization may not be optimal for self-assembly

Error Correction

- Our solution depends on CA rule “moving particles”.
- These are very sensitive to misincorporations-
  - ★ If a particle tile gets misincorporated it will propogate and be locked in quickly.
  - ★ If a misincorporation occurs on a particle path it destroys the particle.

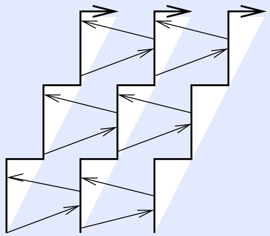
Example: Consider a tile system with one moving particle and misincorporation rate ϵ.



$$\text{Expected spontaneous wires} = \epsilon A + o(\epsilon)$$
$$\text{Expected path length} = \frac{1-\epsilon}{\epsilon}$$

Error Correction

- We can improve this by using “teams” of particles



- A team of $2k+1$ particles corrects k errors and requires $k+1$ errors for spontaneous creation, but the errors can appear anywhere throughout the three “check and move” steps.

  Analysis:

$$\text{Expected \# spontaneous particles} = O(A\tilde{\epsilon}^k)$$
$$\text{Expected path length} = \Omega(\exp\left[\tfrac{k}{6\epsilon(1-\epsilon)}\right])$$
$$\text{Number of tiles increase by a factor of } k$$

TO DO: handle collisions and scattering with linear tile growth.

Conclusions

- We think $T = 2$ circuit assembly is relatively easy.
- The same sort of tricks work to build other shapes, including
  - ⋆ power-law crossbar
  - ⋆ 2-hot decoder
  - ⋆ sorting network
  - ⋆ fat tree?
- What shapes can we build with $T = 2$ tiles but not CA rule tiles?
- What do we need to do to make this work in the lab (or a factory)? Error correction? A 'weaker' assembly model?