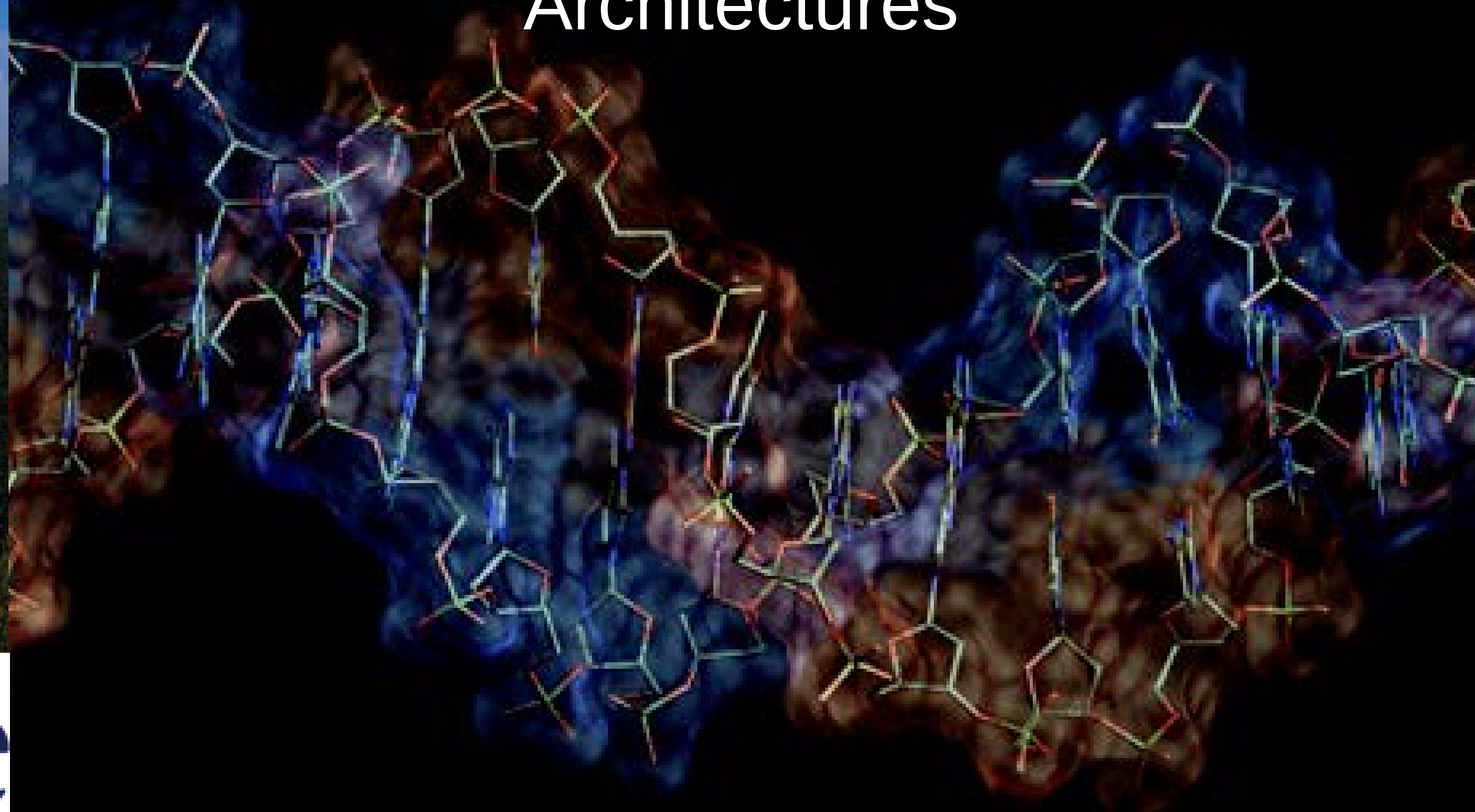
Self-assembled Materials for Computer Architectures

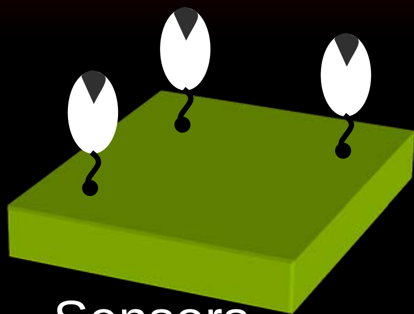


Duke UNIVERSITY

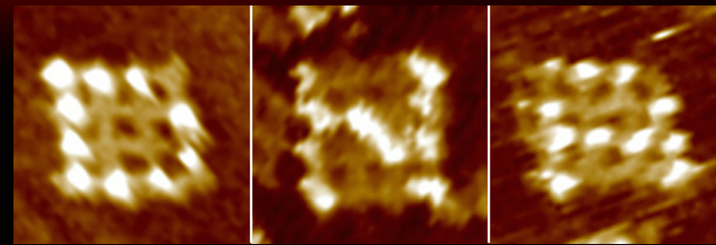
Chris Dwyer
Associate Professor
Department of Electrical and Computer Engineering,
Department of Computer Science

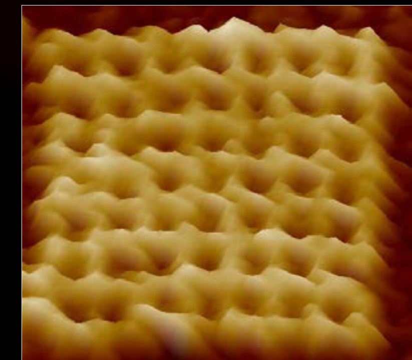
HPEC, 2011

Lab Overview

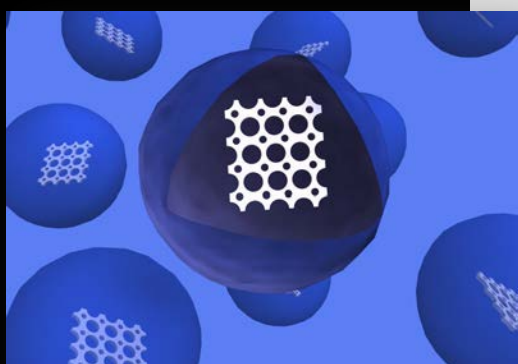


Sensors

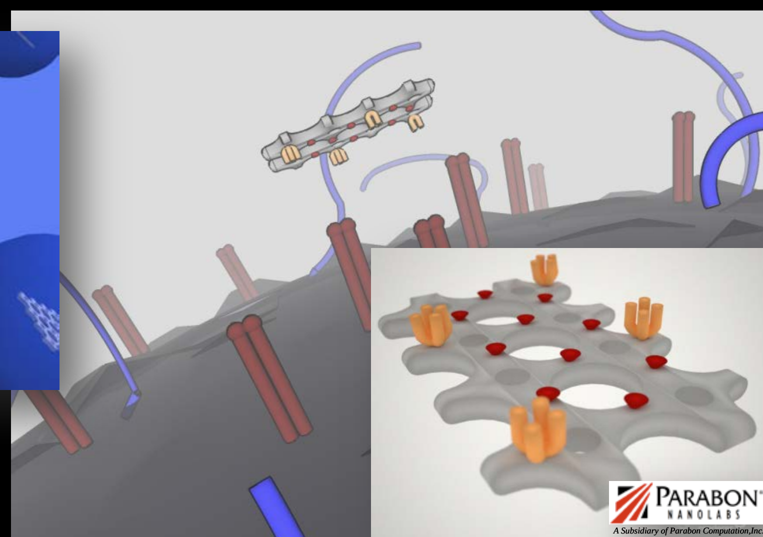




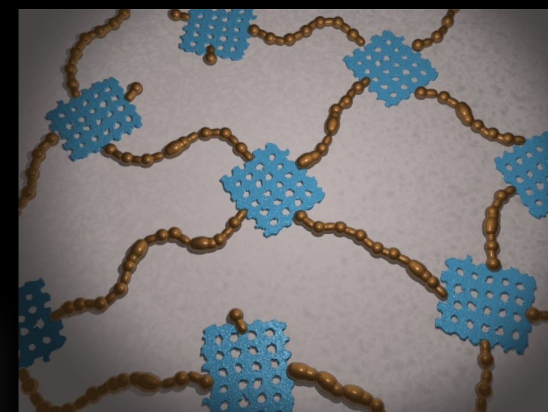
Precise chemical patterning



“Taggants” and Devices



Targeted Chemotherapies



Self-organizing computer architectures

[DNA] self-assembly is a technology that enables next-generation materials, computers, sensors, and systems

Outline

- **Architectural Drivers**
  - Economics of Performance
  - New Computational Domains
- **Disruptive Technology**
  - DNA Self-assembly
  - Devices & Logic Elements
- **Self-organizing Architectures**
  - Multiplexed Sensing
  - Design study: SOSA

Economics of CMOS vs. DNA

	Year	Conven. Fab. Cost*
(180 nm)	2000	~$1B
	2002	Δ=$1.6B
(90 nm)	2004	Δ=$2B
	⋮	⋮
(65 nm)	2006	Δ=$3B
	⋮	⋮
(32 nm)	2009	Δ=$4B
(22 nm)	2010	Δ=$6-8B
	⋮	⋮
(14 nm)	2013	Δ=$5B+?
(<10 nm)	2014+	Δ=?

Value of goods sold†	
Semiconductors	~$75B
(Non-petrol.) Chemical mfg.	~$400B

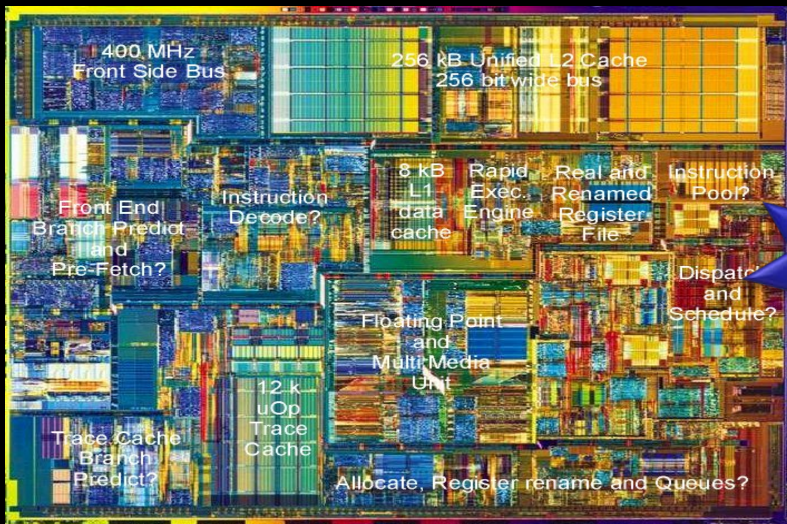
Market Capitalization‡	
Semiconductors	$ 260B
Drug Manufacturing	$1,170B
Specialized Chemical Mfg.	$ 247B
Biotechnology	$ 349B
+	$1,766B

Microelectronics → $ 260B

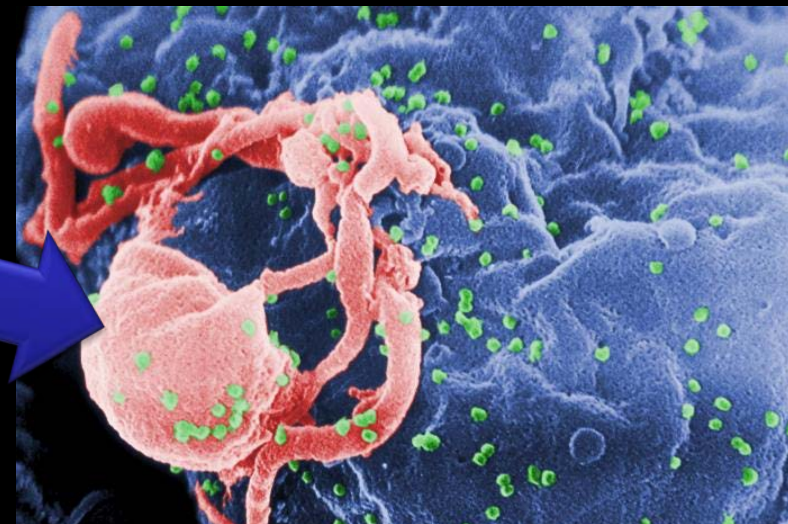
DNA self-assembly (<1 nm) → $1,766B

**EE Times, †2004 U.S. Census, ‡ Yahoo! Finance*

Why? To Enable New Domains for Computation

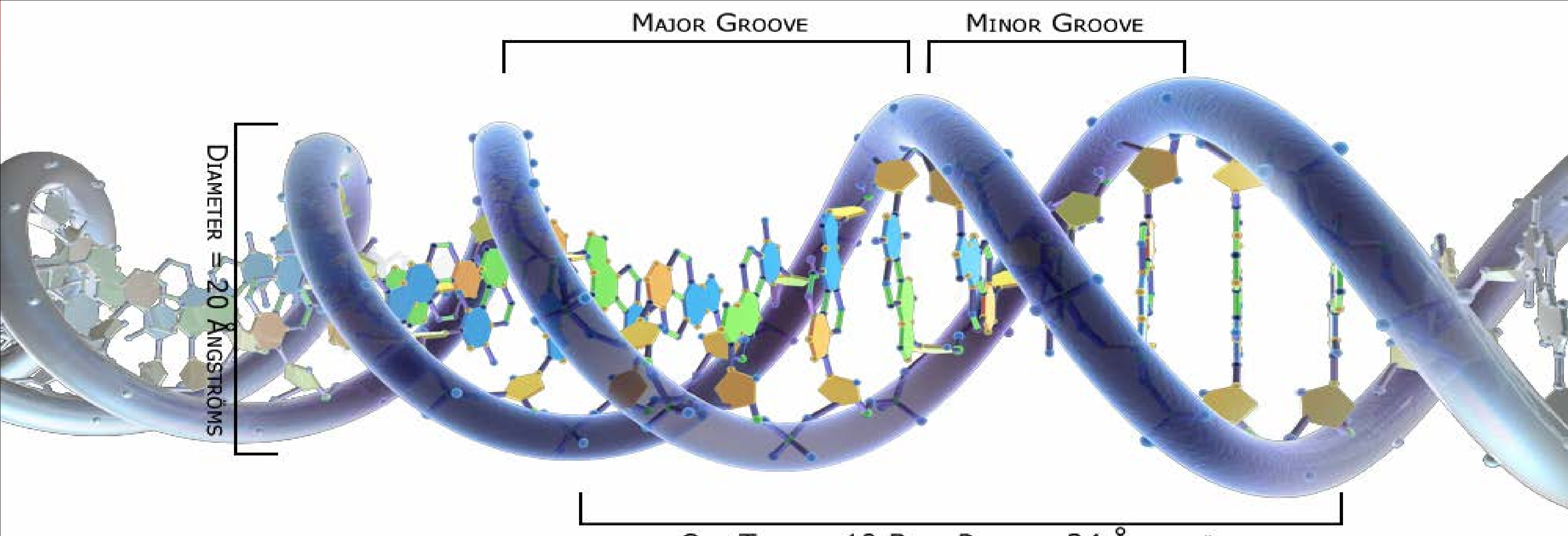


Intel 8088

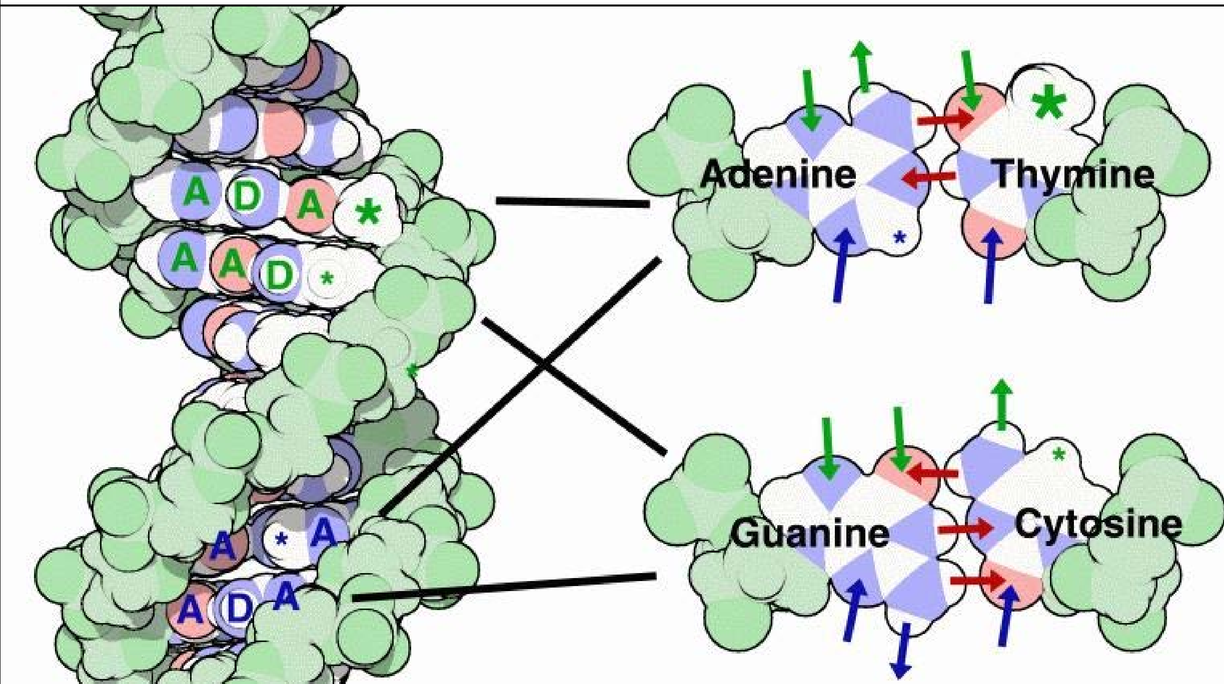


HIV-1 budding from lymphocyte (*CDC*)

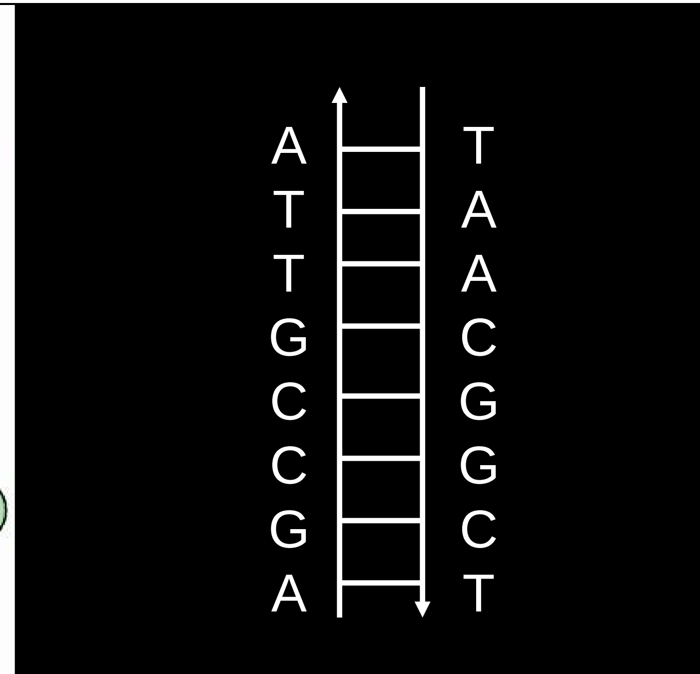
- New Systems
  - Novel materials to introduce computing to entirely new domains
- Hybrid Systems
  - Novel materials to enhance existing CMOS
  - Augmented optical computing on chip (e.g., w/ optical NOCs)



Source credit: Wikipedia



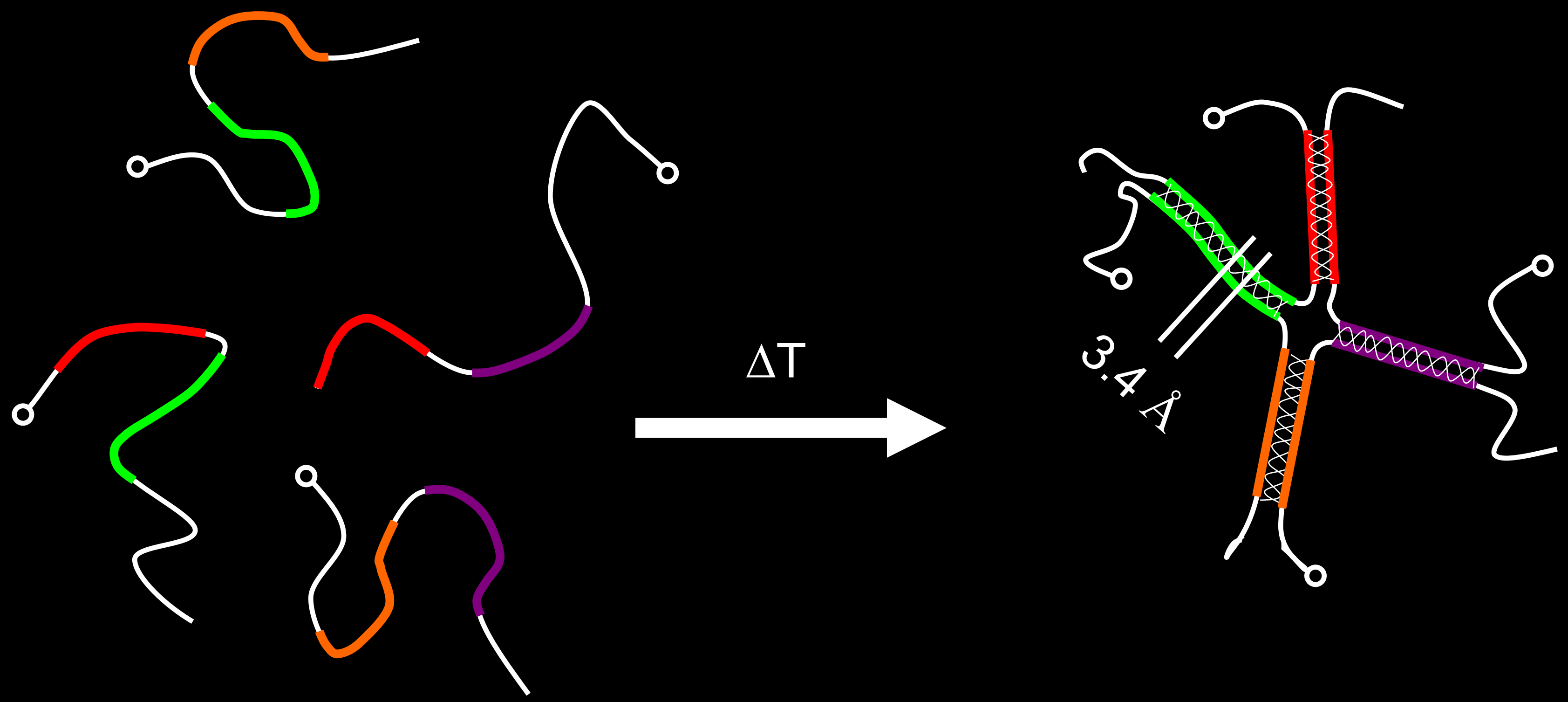
Source credit: Wikipedia



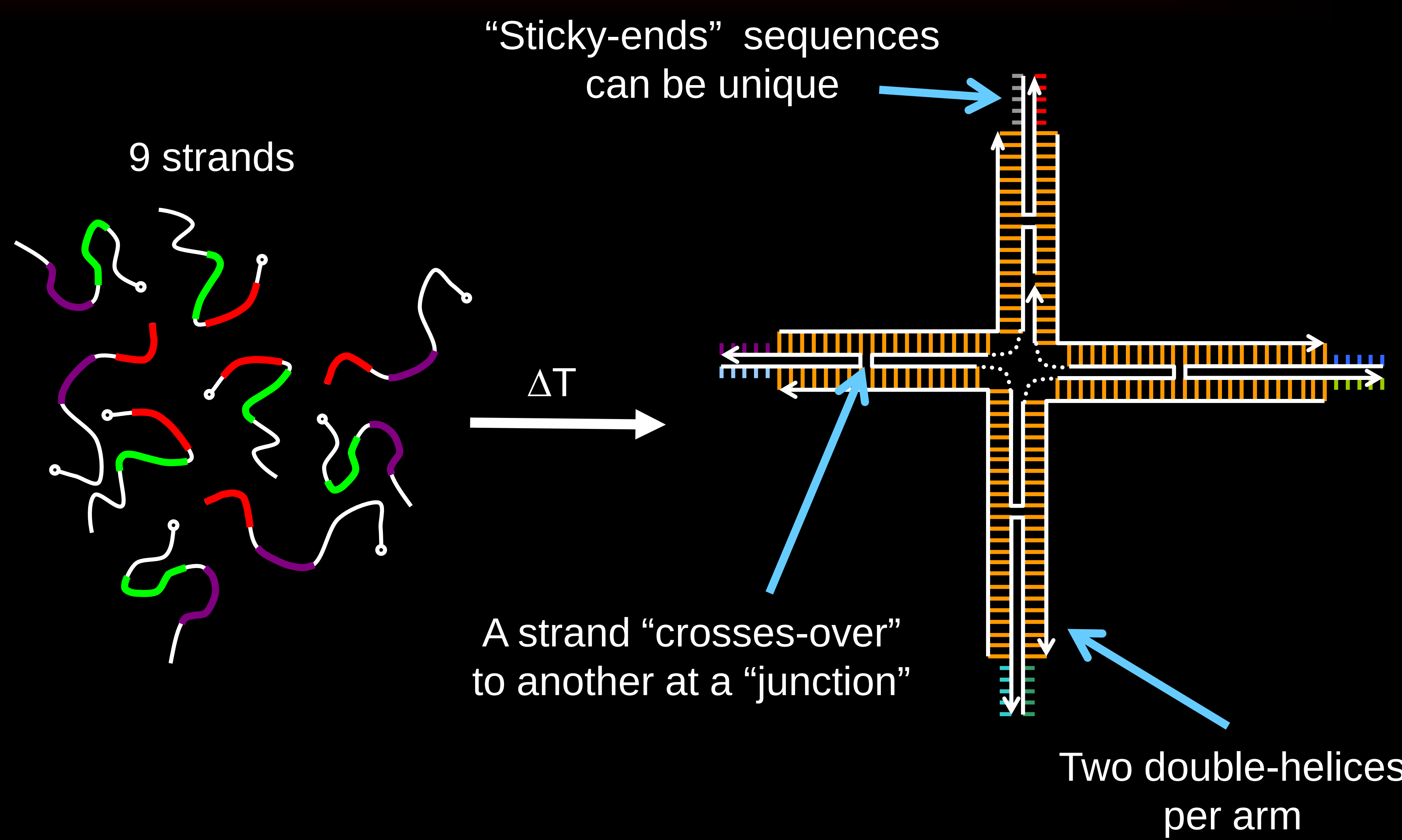
http://nano.ece.duke.edu

DNA Scaffolds - Geometry

- The geometric properties of double strands can form specific, controlled self-assembled nanostructures:

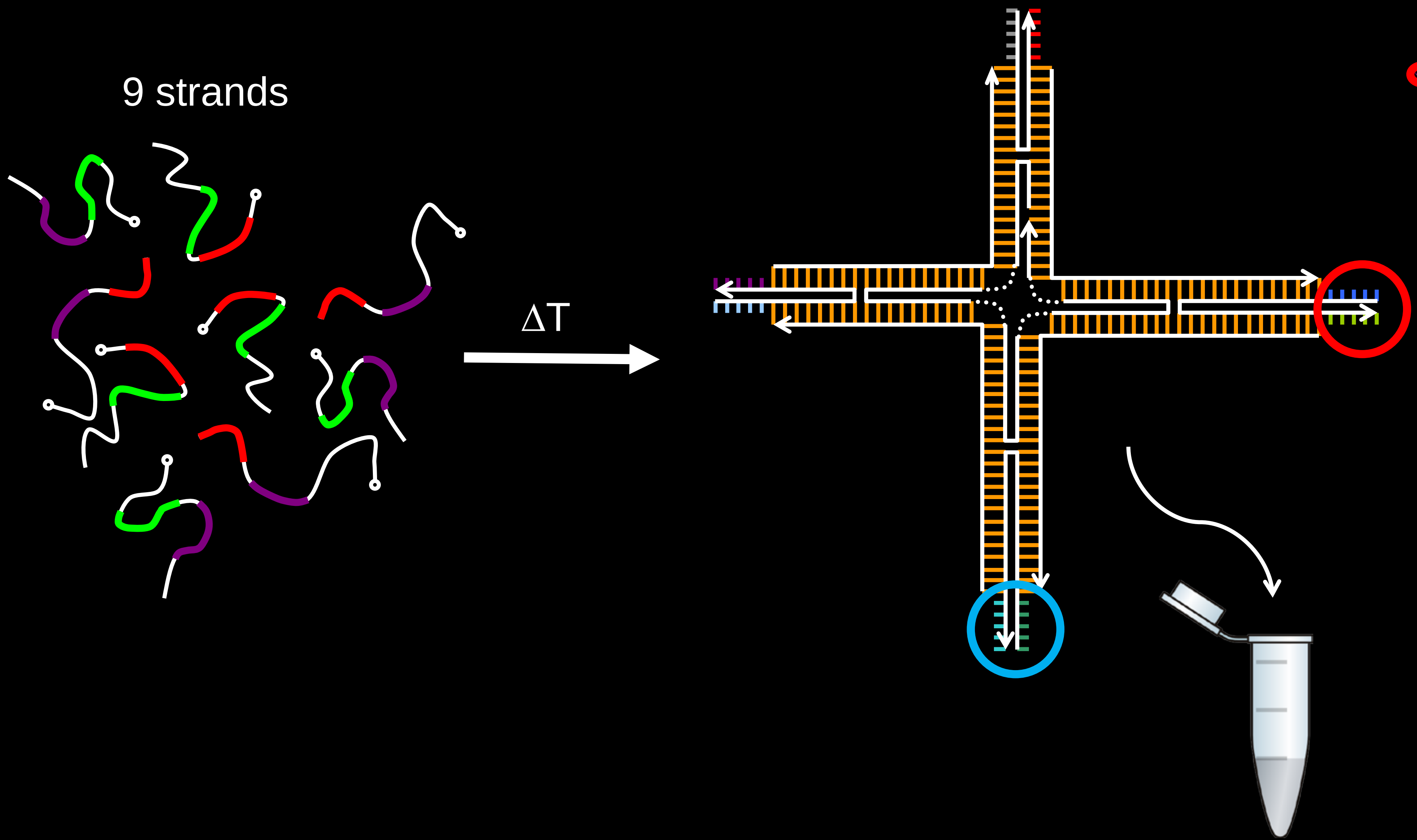


DNA Motifs

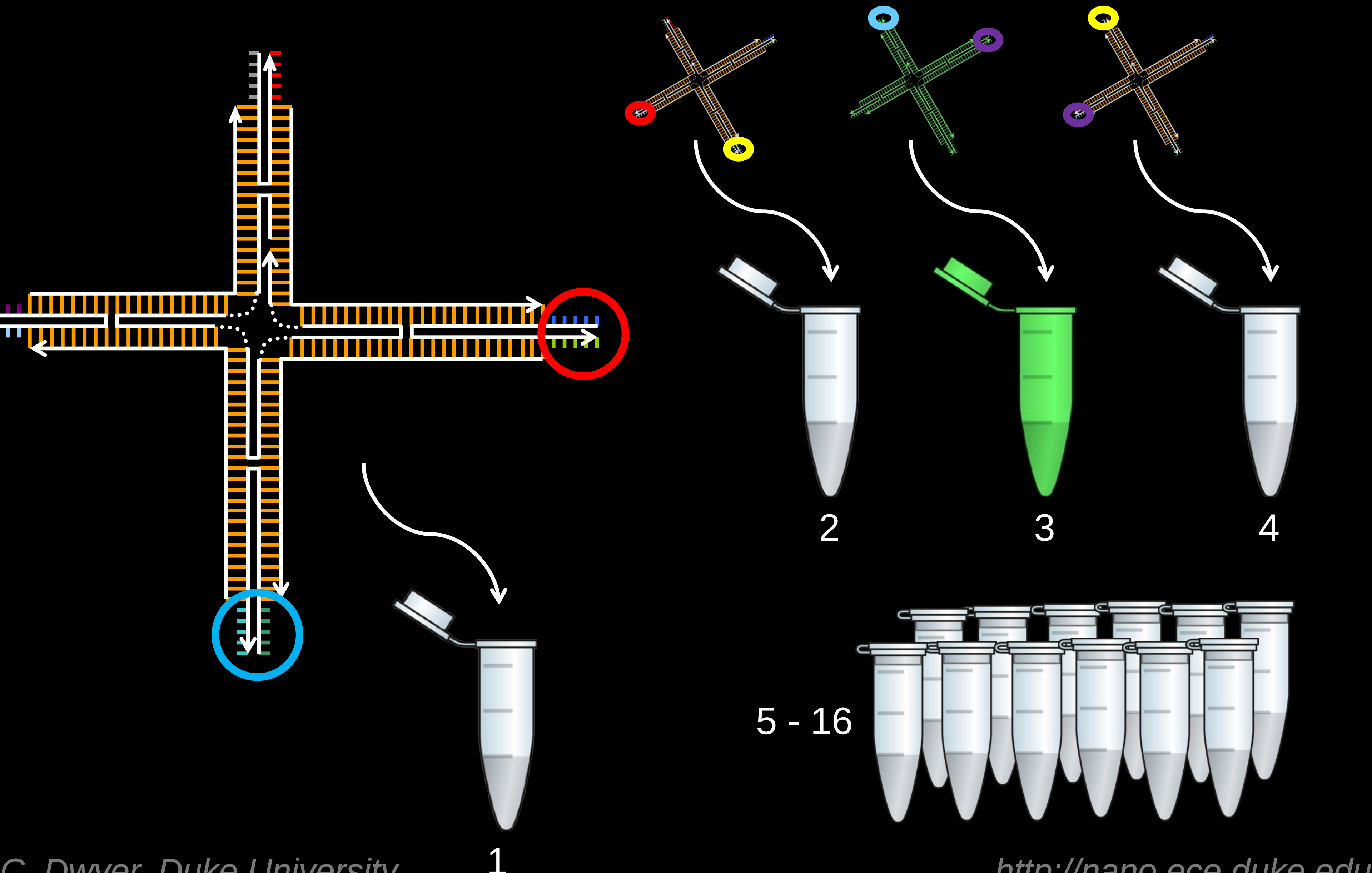


"Sticky-ends" sequences can be unique
9 strands
ΔT
A strand "crosses-over" to another at a "junction"
Two double-helices per arm

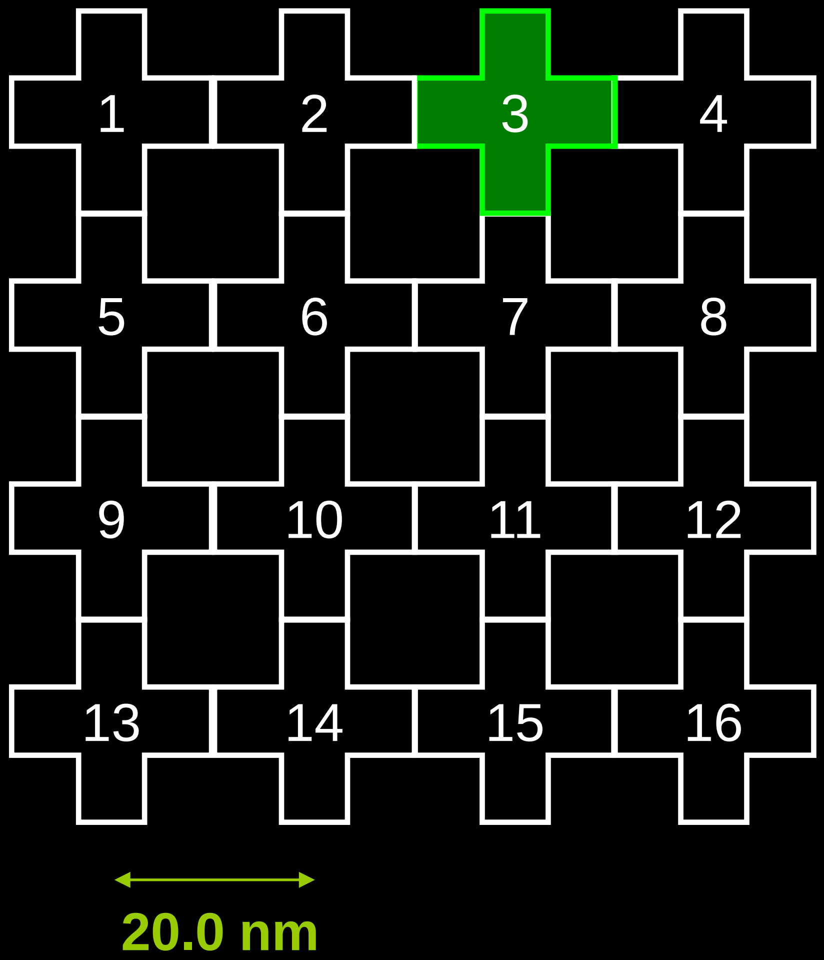
DNA Motifs

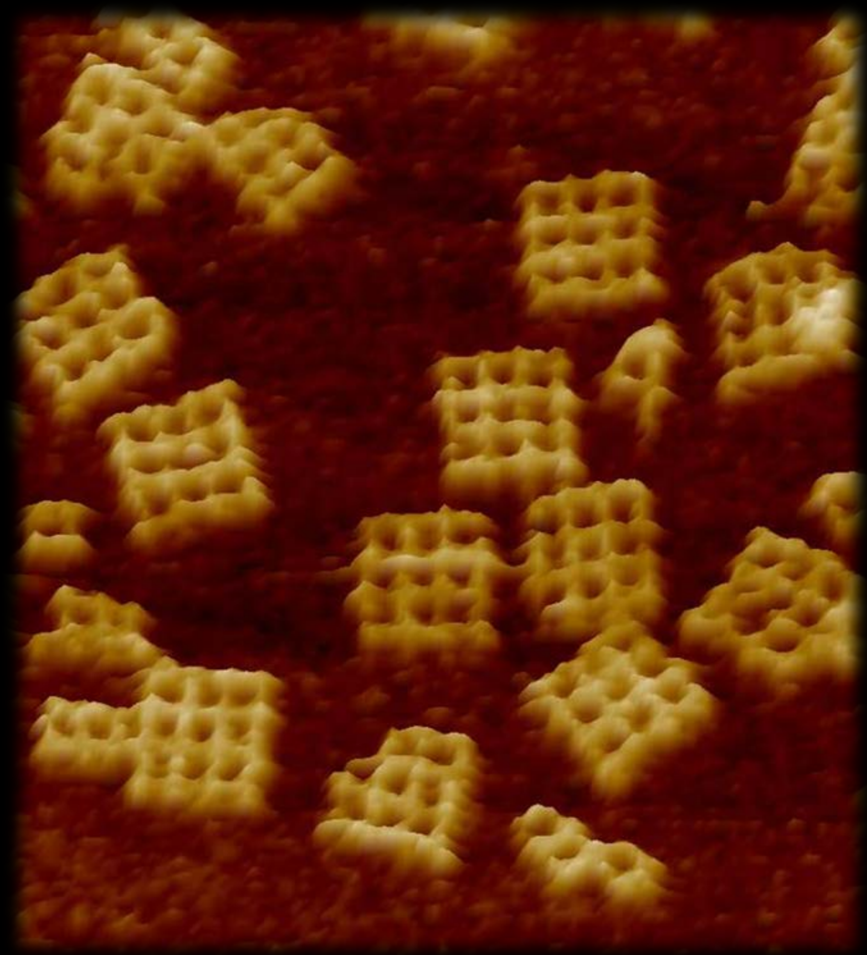


DNA Motifs



DNA Motifs



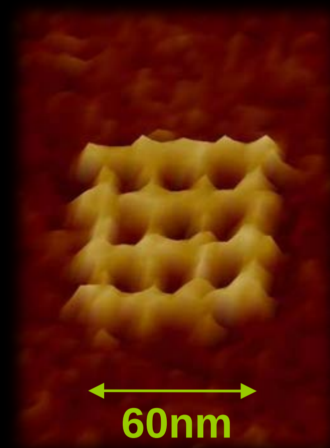


Atomic Force Microscopy (AFM) Image
(360nm X 360nm)

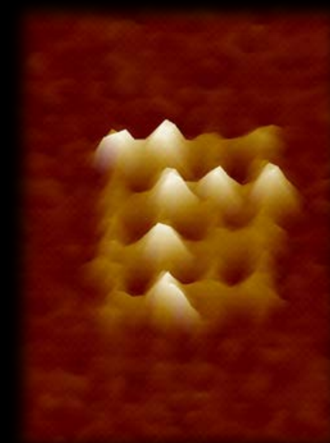
DNA Nano-grids

Self-assembled grids with sub-nm resolution (3.4 Å)

Multiple DNA grids deposited on flat mica plane

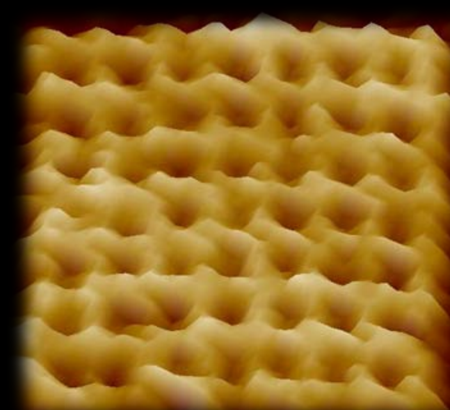


16-tiles



Patterned grid

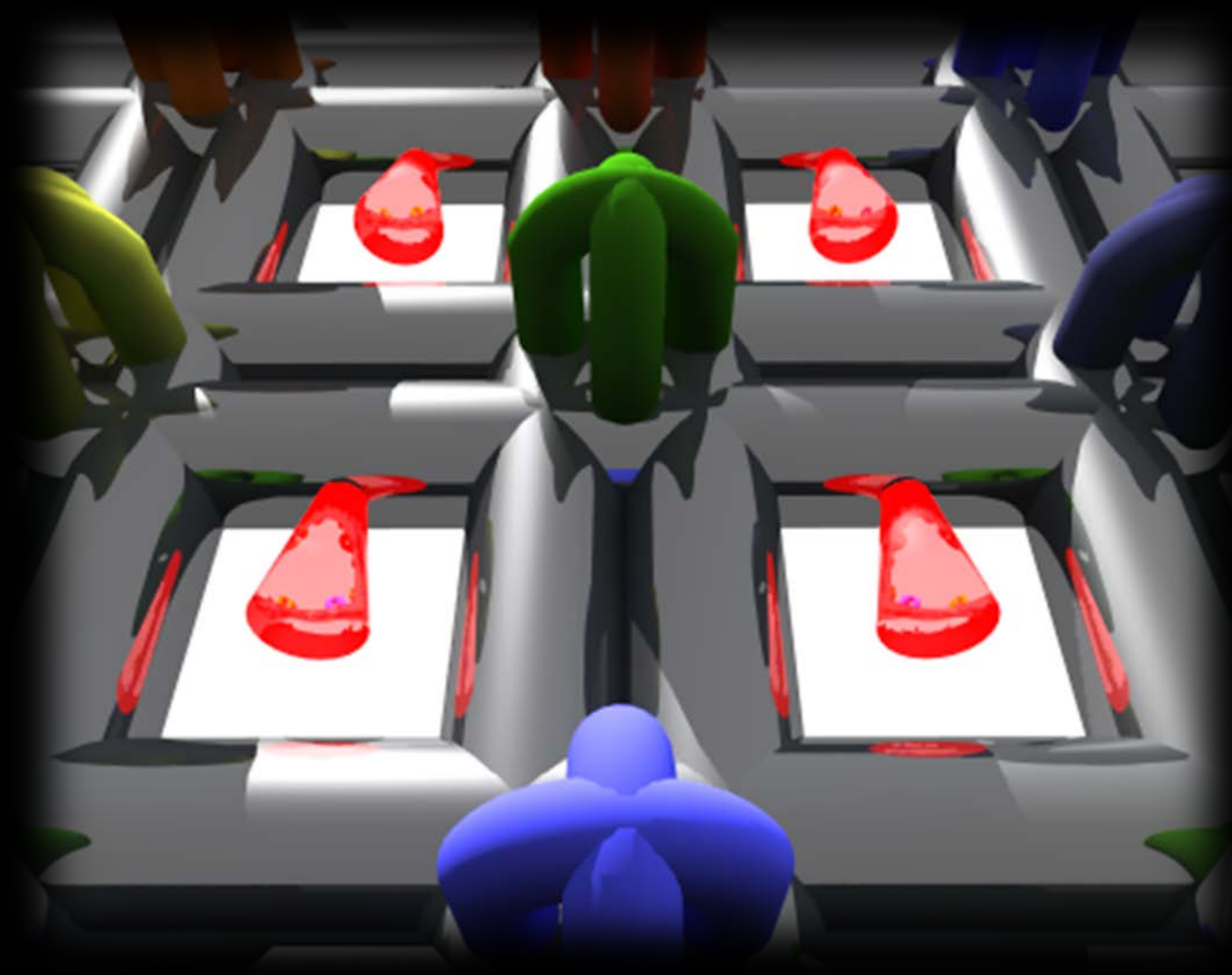
64-tiles



Manufacturing scale: >10^{15} grids/mL

(< 8 hours)

Next-step: Add active, functional components



Outline

Architectural Drivers

- Economics of Performance
- New Computational Domains

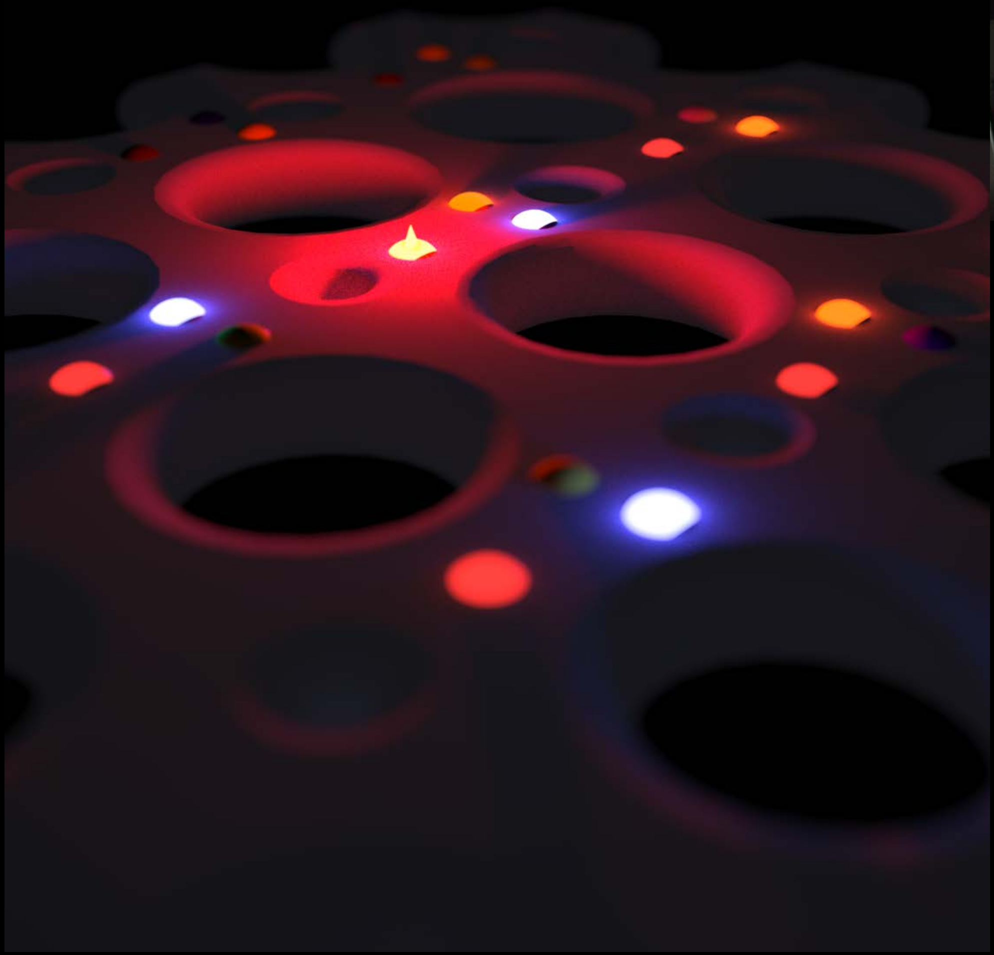
Disruptive Technology

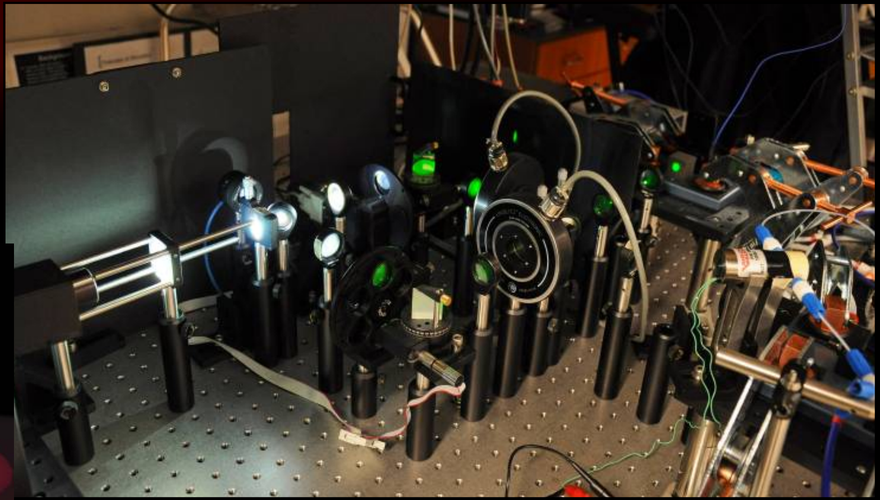
- DNA Self-assembly
- Devices & Logic Elements

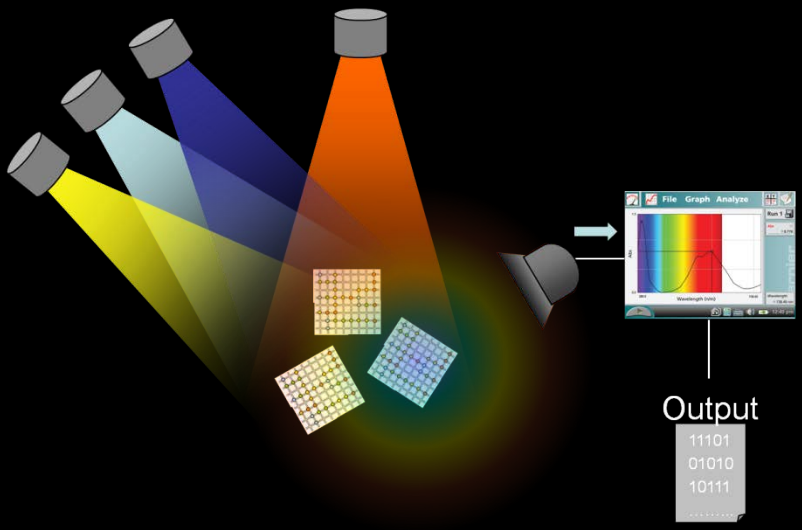
Self-organizing Architectures

- Multiplexed Sensing
- Design study: SOSA

Operational Overview for RET Logic

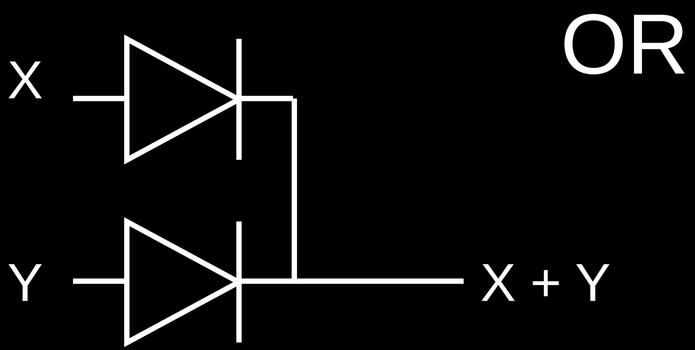


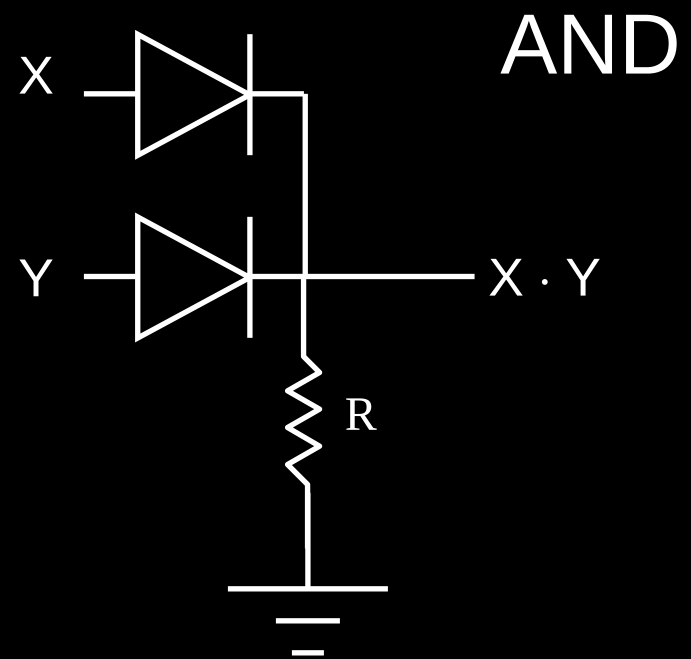




C. Pistol, et al., *Small,* 2010.

RET Circuit Theory





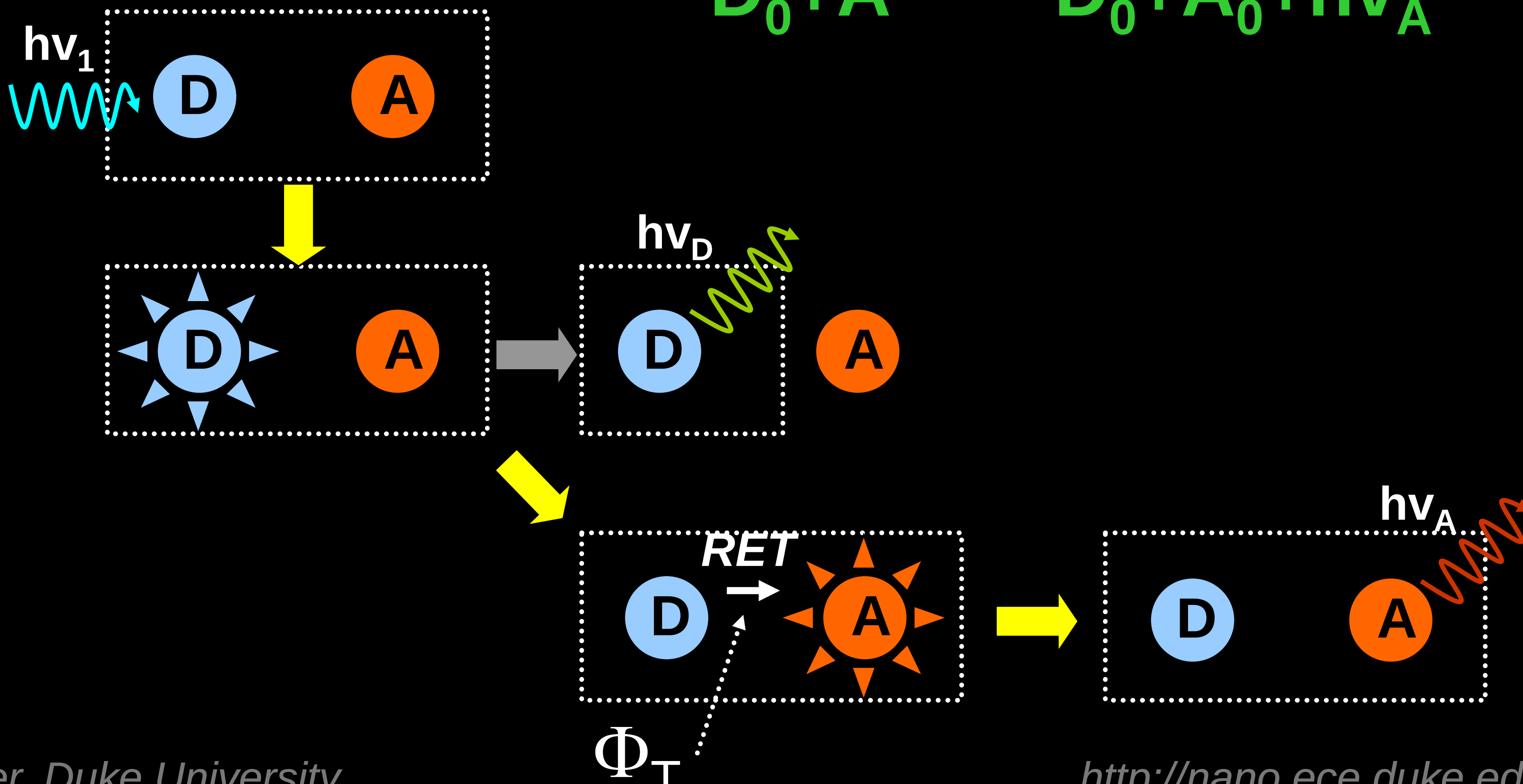
- OR / AND primitives, good but…
- Inversion by dual-rail encoding OR new devices

- New opportunity to integrate logic into *molecular* scale processes

RET-Logic: Resonance Energy Transfer

$$D_0+A_0 \xrightarrow{hv_1} D^*+A_0 \longrightarrow D_0+A_0+hv_D$$

$$D^*+A_0 \xrightarrow{RET} D_0+A^* \longrightarrow D_0+A_0+hv_A$$



RET Efficiency

$$D_0 + A_0 \xrightarrow{hv_1} D^* + A_0 \xrightarrow{RET} D_0 + A^* \longrightarrow D_0 + A_0 + hv_A$$

- **Energy transfer efficiency**

$$\Phi_{T_{D \to A}} = \frac{R_0^{\,6}}{r^6 + R_0^{\,6}}$$

$$R_0 = 0.211(\kappa^2 \eta^{-4} Q_D J(\lambda))^{1/6}$$

QM + EM derivation

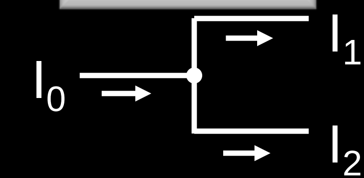
R_0	: Förster Radius
r	: D – A distance
κ	: D – A alignment
η	: index of refraction
Q_D	: Donor quant. yield
J(λ)	: Overlap integral

T. Förster, "Intermolecular energy migration and fluorescence," Annals of Physics, vol. 2, pp. 55-75, 1948.

RET Circuit Theory

(1)
$$V_F = I_E \cdot R_T$$
$$R_T \equiv (1 - \Phi_T)$$

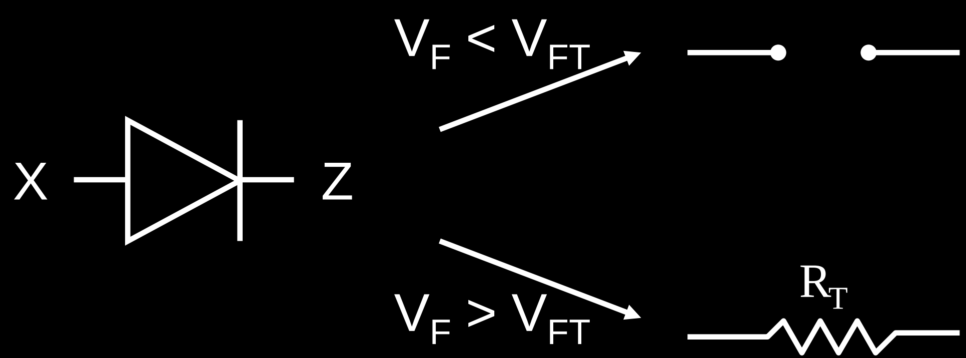
(2)
$$\int_N I_E = 0$$



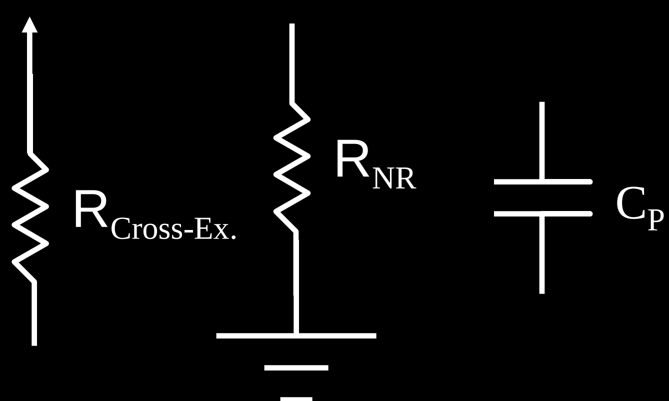
(3)
$$\oint_P V_F = 0$$

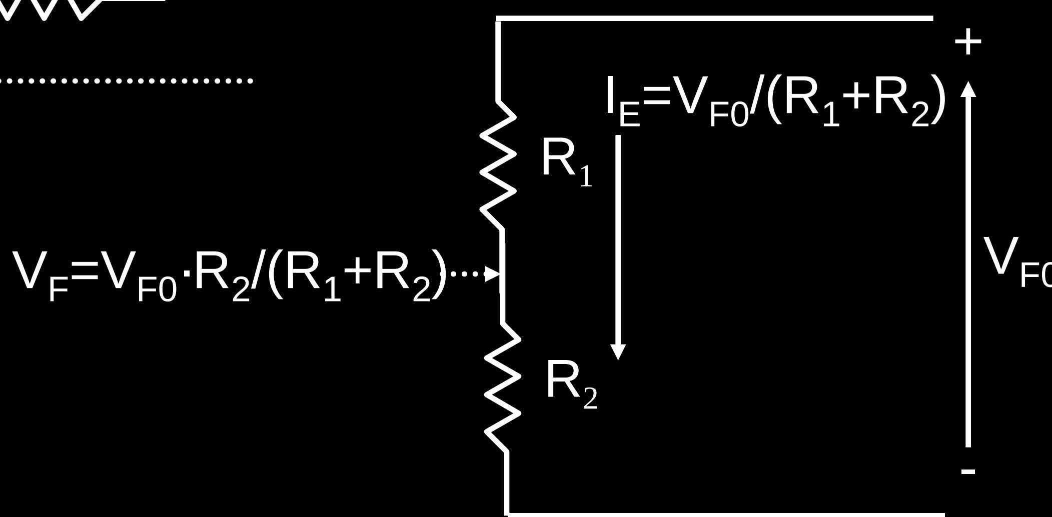
(energy migration)

- RET Cascades

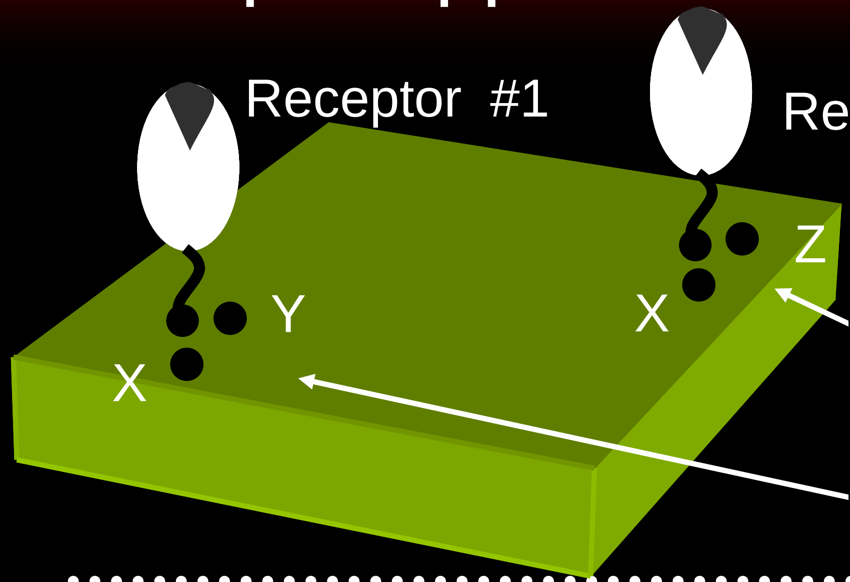


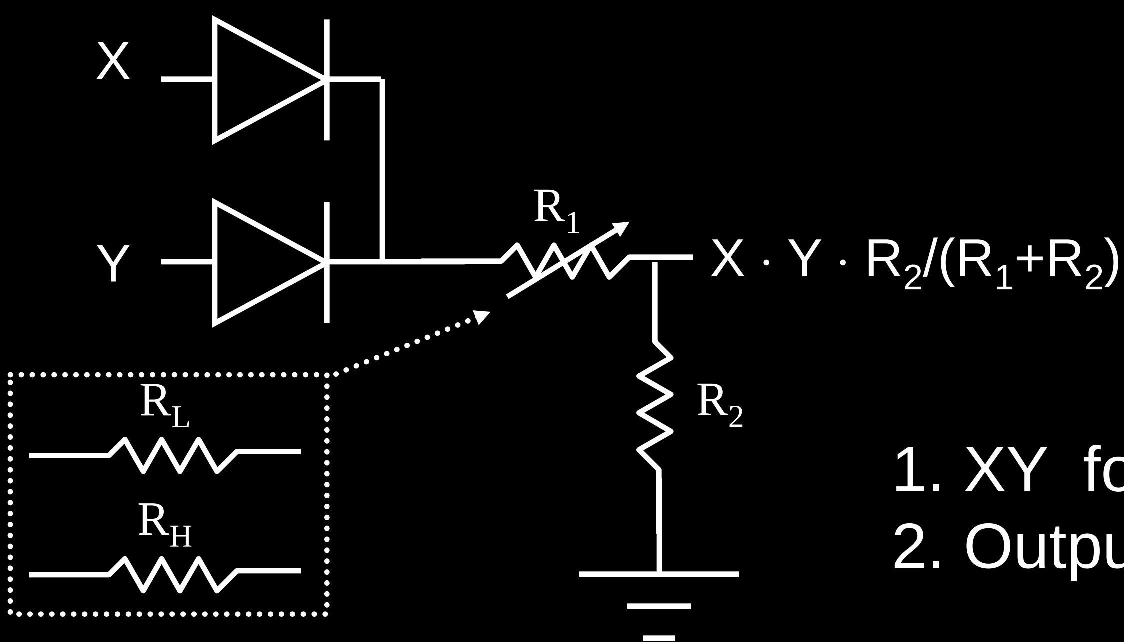
- Other elements





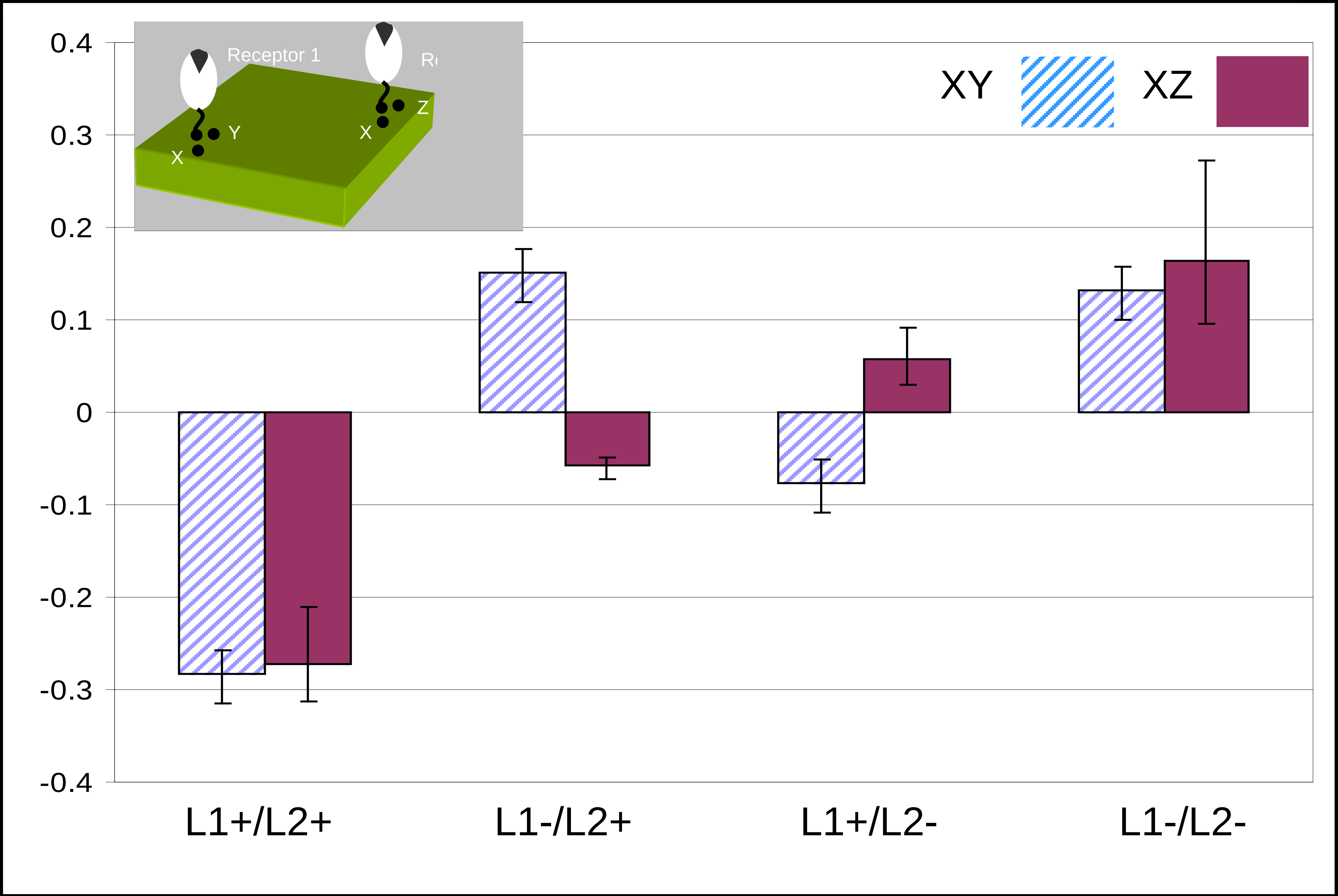
Simple Application: Logic + Sensors





1. XY forms a distinct "address"
2. Output depends on $R_2/(R_1+R_2)$

Logic and Sensors



Design Study: SOSA and New Constraints

- Self-assembly introduces chaos at the microscale:
  - ($>$1-10 μm)
- DNA hybridization restores order at the nanoscale:
  - ($<$ 1 μm) sub-nm resolution

- Three fundamental assumptions must change:
  - Availability of long-range interconnect *(toward none!)*
    - No large-scale interconnect networks / limited local comm.
  - Area vs. cost tradeoff *(toward an even worse exponential!)*
    - Large ($>$ ~5 μm on a side) circuit footprints are impractical
  - Reliable devices *(toward error prone!)*
    - The substrate can have defects
    - The devices can have defects

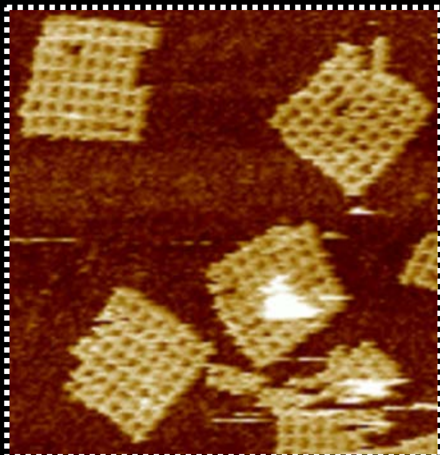
Self-Organizing Architectures

- Conventional architectures do not self-assemble

- Several new architectures can exploit self-assembly:
  - *Oracles*: DNA computing with a device twist that enables rapid (electrical) re-use of a DNA computation
  - *DAMP*: Distributed Array Multi-processor, SIMD without an interconnection network for embarrassingly parallel codes.
  - *NANA*: Nanoscale Active Network Architecture, general purpose but imbalanced due to a large communication / execution ratio- under utilized resources.
  - *SOSA*: **Self-organizing SIMD Architecture**, a SIMD model on a reconfigurable network topology. Utilization is high due to a depth first network traversal.

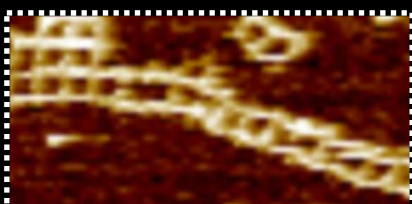
Tangent: Interconnect, even if limited…

1

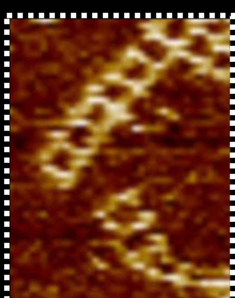


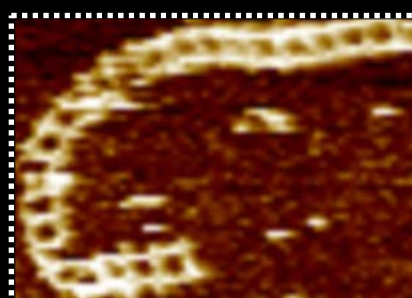
• Self-assemble grids, ... , randomly deposit on smooth surface.

2



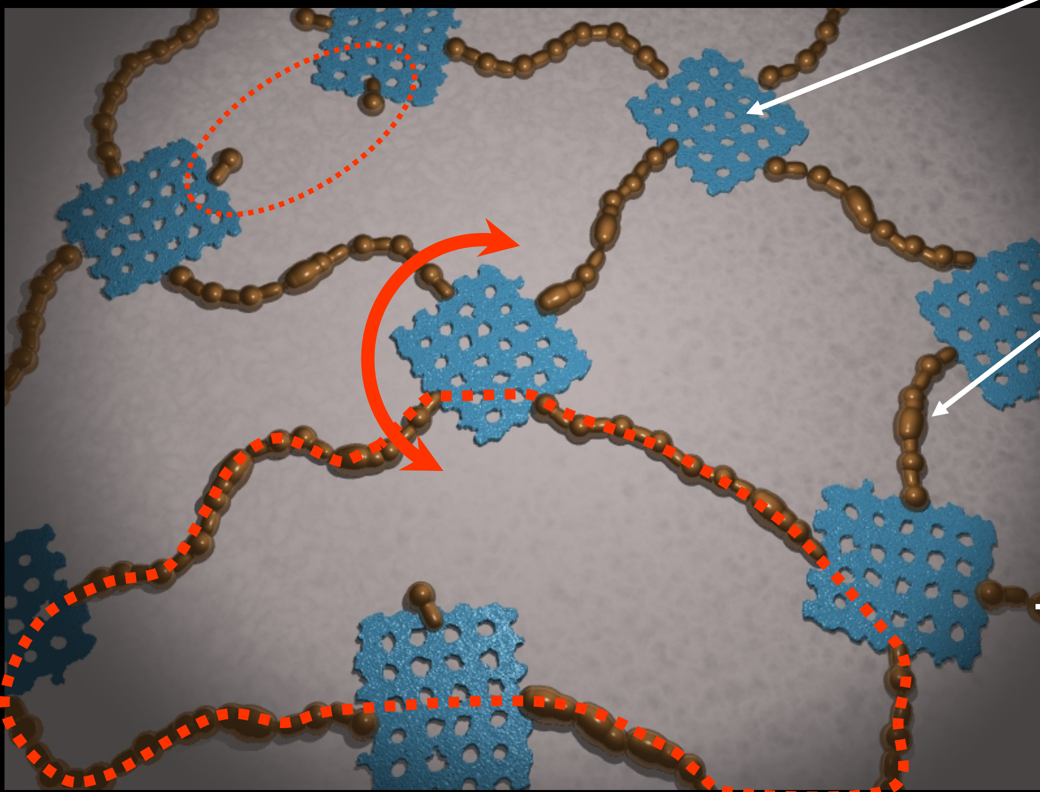
• Self-assemble nanotracks between grids on surface.





3 Metallize tracks...

Self-Organizing Architectures



Self-assembled
Computational nodes
(minimal computational power & defect-prone)

Self-assembled
Interconnect
(defect-prone)

Distant micro-scale
I/O contact
(low bandwidth)

Defect model includes:

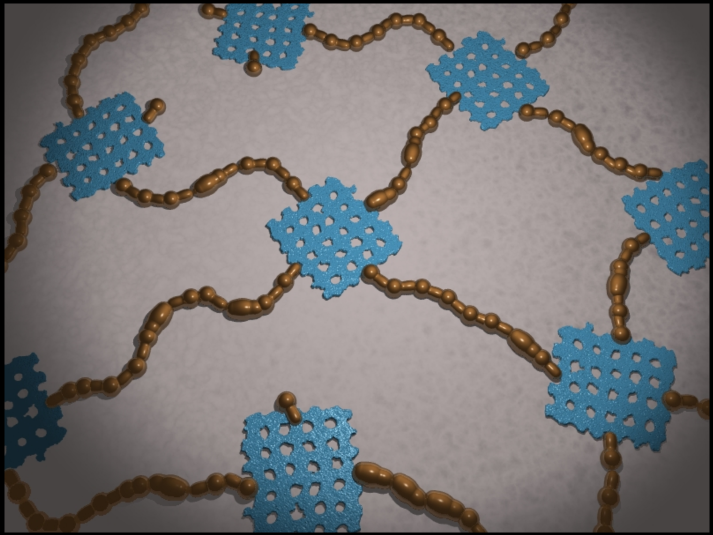
- Rotation, position
- Connectivity
- Defective devices on nodes

Strategy

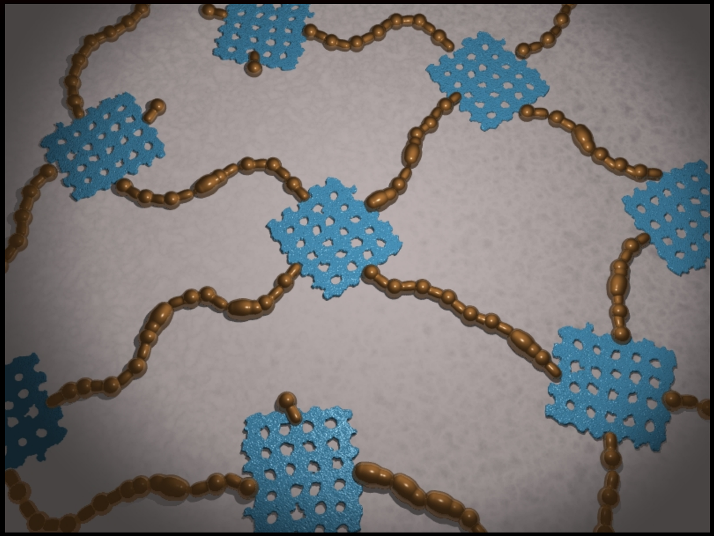
1. Map out both topology and defective links and nodes in the network (i.e., *configure*)

2. Group nodes together to form more powerful computational elements

3. Provide a convenient programming abstraction



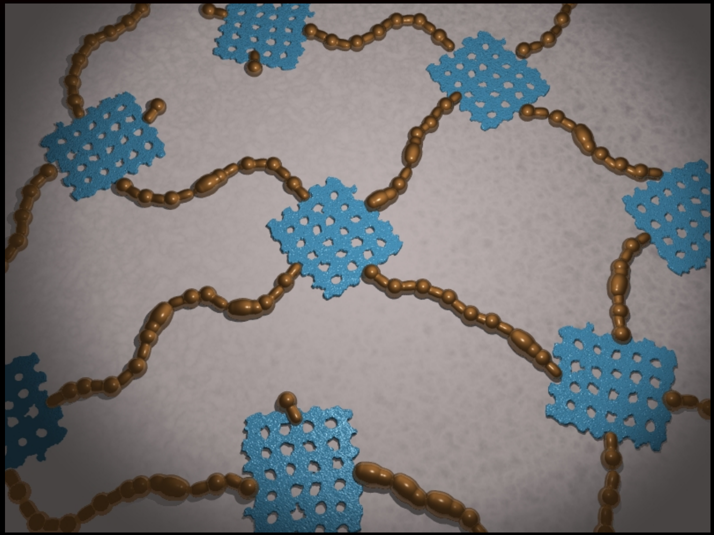
Major Design Challenges (w/ Self-assembly)



- Conventional approaches to defect tolerance fail
  - Global defect maps are impractical if:
    - System is large (>10^{12} nodes),
    - Nodes are weak
    - I/O bandwidth is limited
- The network topology is unknown and random
  - Every system instance is different!
- Nodes *will* have defective pieces
  - Functional and performance binning impractical

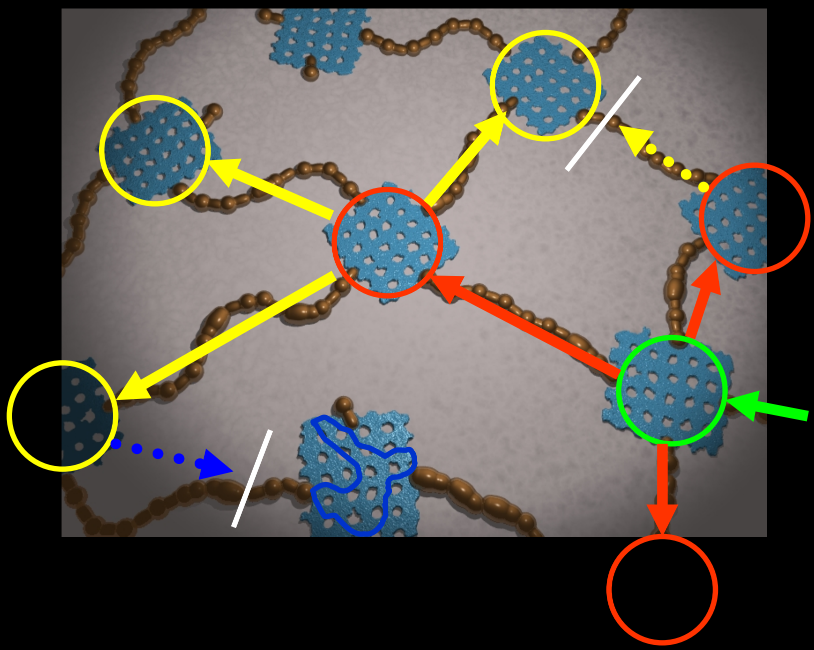
Design Solution

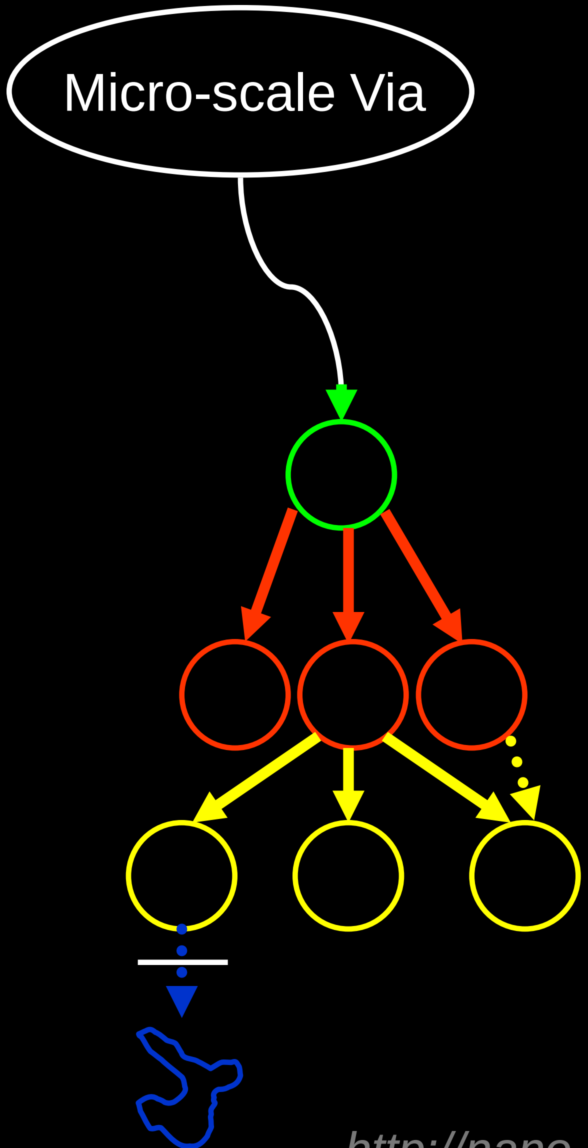
- Use the reverse path forwarding (RPF) algorithm to create a tree
- Use Euler paths (EP) to create an abstract ring



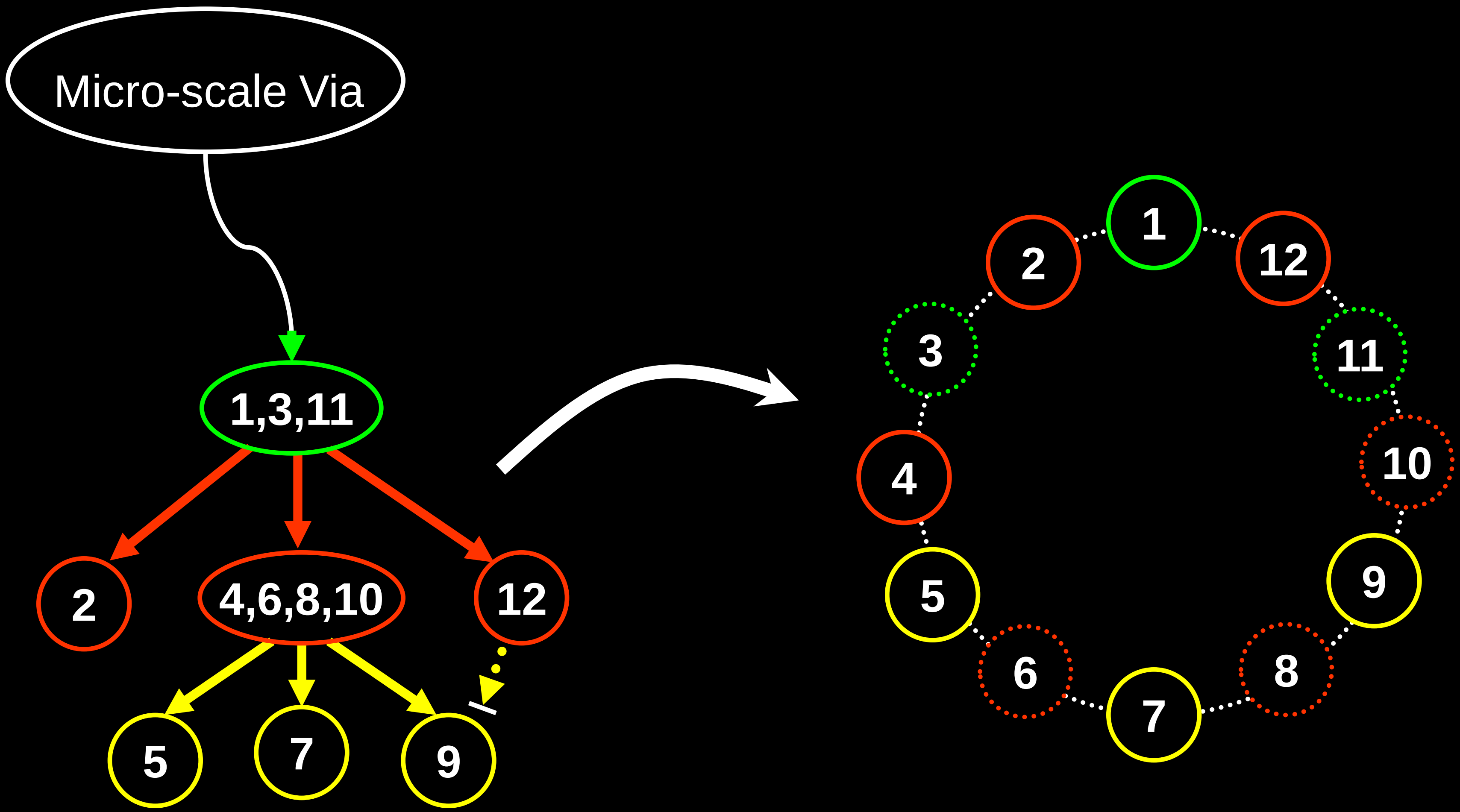
- Use lightweight built-in self-test per node to protect connectivity
- Asynchronous! (no global clock)
- Instructions and data are broadcast to all nodes in *network-order*, down the RPF tree.

SOSA Configuration: RPF



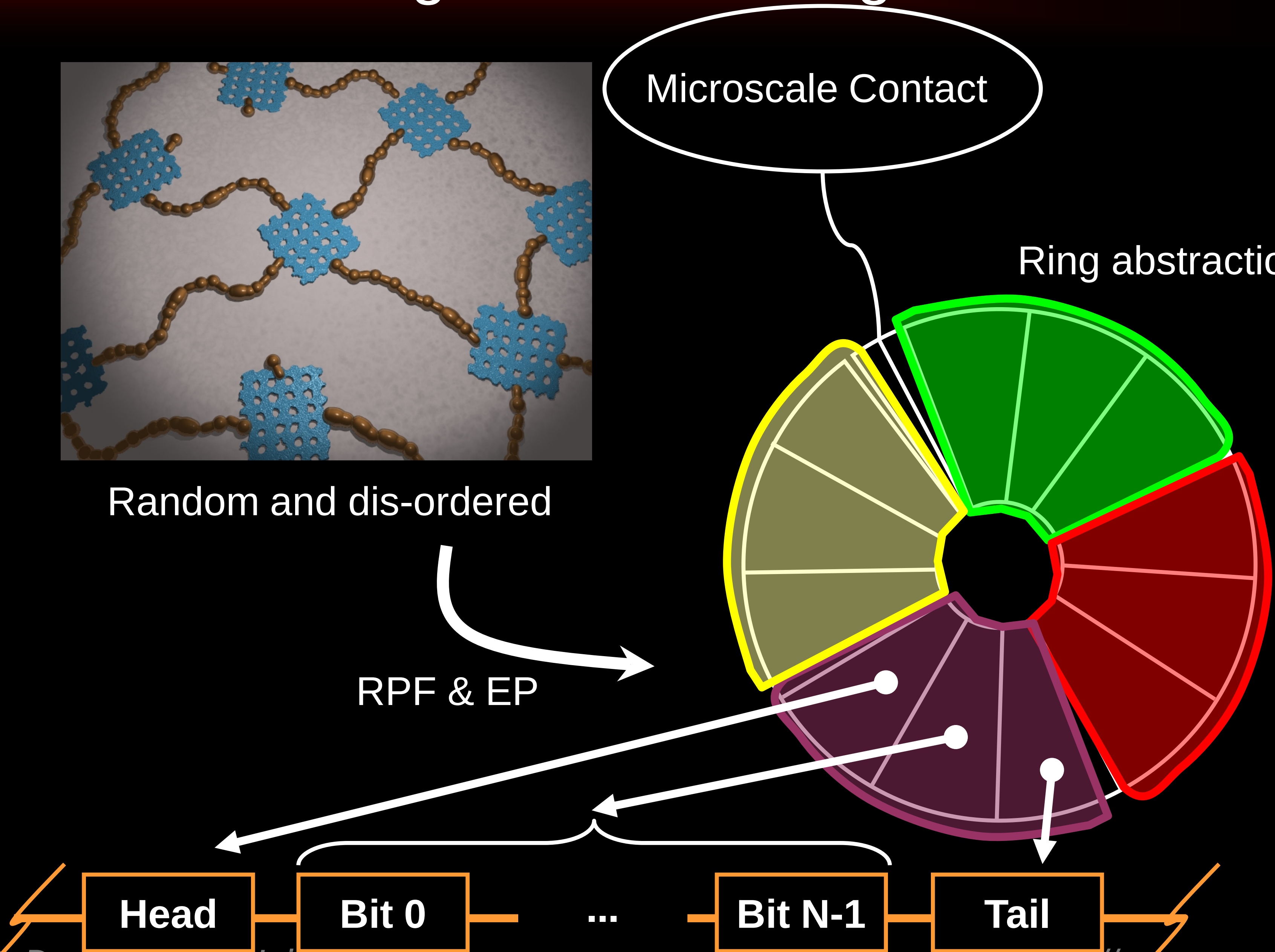


SOSA Configuration: EP

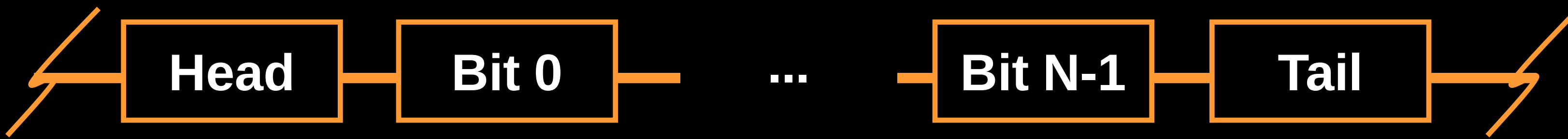


(DFS visit time)

SOSA Configuration: the ring



SOSA Execution Model Overview



- Instructions are broadcast to all PEs from the microscale I/O contact
- Partial instruction re-use is permitted to reduce instruction broadcast overhead
- Predicated instructions serialize through a synchronization signal sent from the head node of the PE (the head node holds predicate flags)
  – *Why? out-of-order inst. arrival at nodes within a PE*
- PESHIFT operations (left, right) enable PE communications on the ring

Performance Comparison

We compare SOSA against:

- P4 at 3GHz
- Ideal SOSA
- Ideal Superscalar
- Ideal 16-way CMP

SOSA Configuration

Parameter	Value
Register File	16 entry, 2-bits per node (32 bits wide)
Time Quantum	1 ns (Synchronous equivalent: 1GHz)
ALU Latency	1 time quanta
Inst. buffer size	1 entry
Link type	Full duplex with 3 virtual channels

Ideal SOSA (I-SOSA):

- 1 time quanta instruction execution latency
- No communication overhead

Performance Comparison

Ideal Superscalar (I-SS) Configuration

Parameter	Value	Parameter	Value
Width	128 (Fetch/Dec./Iss./Commit)	Branch prediction	Perfect
Inst. Fetch Queue	1024 Entries	Memory latency	1 cycle
ROB/LSQ	8192 entries, 1 cycle access	Memory ports	128
Integer ALU	128 Add, 128 Mult.	Frequency	10 GHz
FP ALU	128 Add, 128 Mult.		

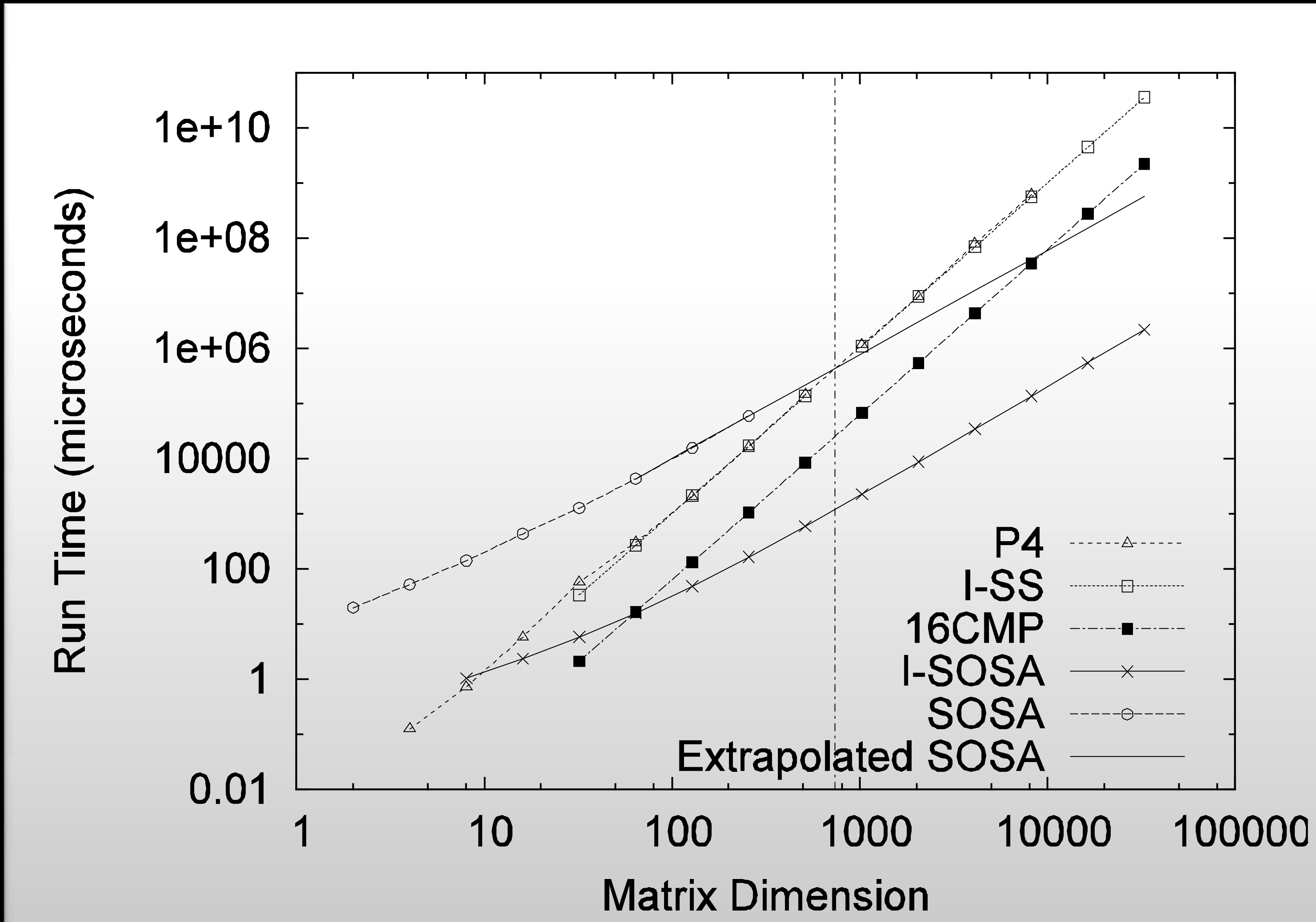
Ideal 16-way CMP (16CMP):

- Linear scaling of I-SS performance

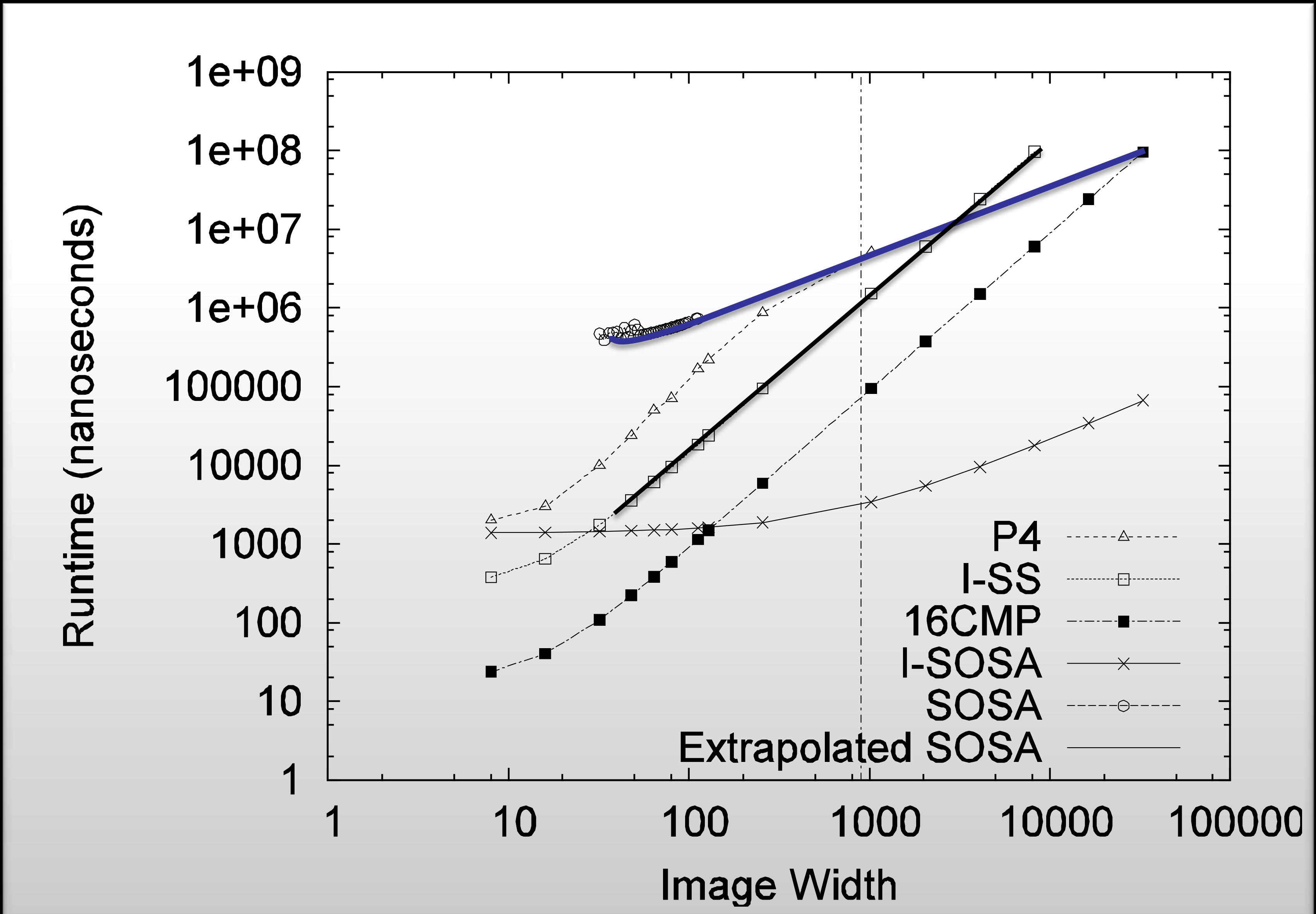
Benchmark Highlights

Application Class	Description
Scientific	Multiply integer NxN matrices (N^2 PEs)
Image Processing (Filters)	1. Generic 3x3 filter 2. Separable Gaussian filter 3. Median noise reduction filter Each over an NxN image (N^2 PEs)
General Purpose	Odd-Even transposition sort. Parallel sort with nearest neighbor communication (N PEs for sorting N keys)
Cryptography	Tiny Encryption Algorithm (TEA) used in XBox IP security.
Search	Search a database for a match with an input 32 bit string. O(N) PEs for N strings.
Bin-Packing	Pipelined version of bin-packing with first-fit heuristic (N PEs for N bins)

Highlight 1 of 2 – *Matrix Multiply*



Highlight 2 of 2– *Gaussian Filter*



Conclusions

- Early days of self-assembled computing and the devices are nascent
  - However, we have demonstrated the fundamental building blocks and are moving toward more powerful devices
- Self-assembly disrupts the conventional paradigm for computer architecture
  - Fortunately, self-organization as a principle can be applied in a defect-prone, distributed setting
- Despite significantly higher defect rates high*er*-performance self-organizing architectures are possible

Students and Sponsors

Thank you!

Jaidev Patwardhan
Costantin Pistol
Vincent Mao
Viresh Thusu
Mohammad Mottaghi
Jun Pang
Vishwa Nellore
Siyang Wang
Heather Duschl
Arjun Rallipalli
Craig LaBoda

Dr. Perumal Ramasamy
Dr. Wanqiu Shen