GENERAL DYNAMICS
Advanced Information Systems

NIAC
NASA Institute for Advanced Concepts

Modeling Kinematic Cellular Automata: An Approach to Self-Replication

NASA Institute for Advanced Concepts

Phase I: CP-02-02

Principal Investigator:
Tihamer Toth-Fejel
Tihamer.Toth-Fejel@gd-ais.com

Consultants:
Robert Freitas
Matt Moses

March 22, 2004 NIAC Fellows Presentation

Modeling Kinematic Cellular Automata

- Rationale
- Benefits
- Applications
- Project Goals
- Strategy
- Accomplishments
- Conclusion and Future Directions
- Additional Material

Rationale

- Why Self-Replication?
- Why not Self-Assembly?
- Why Kinematic Cellular Automata?
- Why both macro and nano scale?

Rationale: Why Self-Replication?

- Revolutionary manufacturing process
- Nanotechnology
- Massive reduction in costs per pound
- Controlled exponential growth

Rationale: Why not Self-Assembly?

Examples have been demonstrated

But…

- Not “Genotype + Ribotype = Phenotype” (GRP)
- No theory
- Against the principles of sound design

However…

Use it for simple input parts

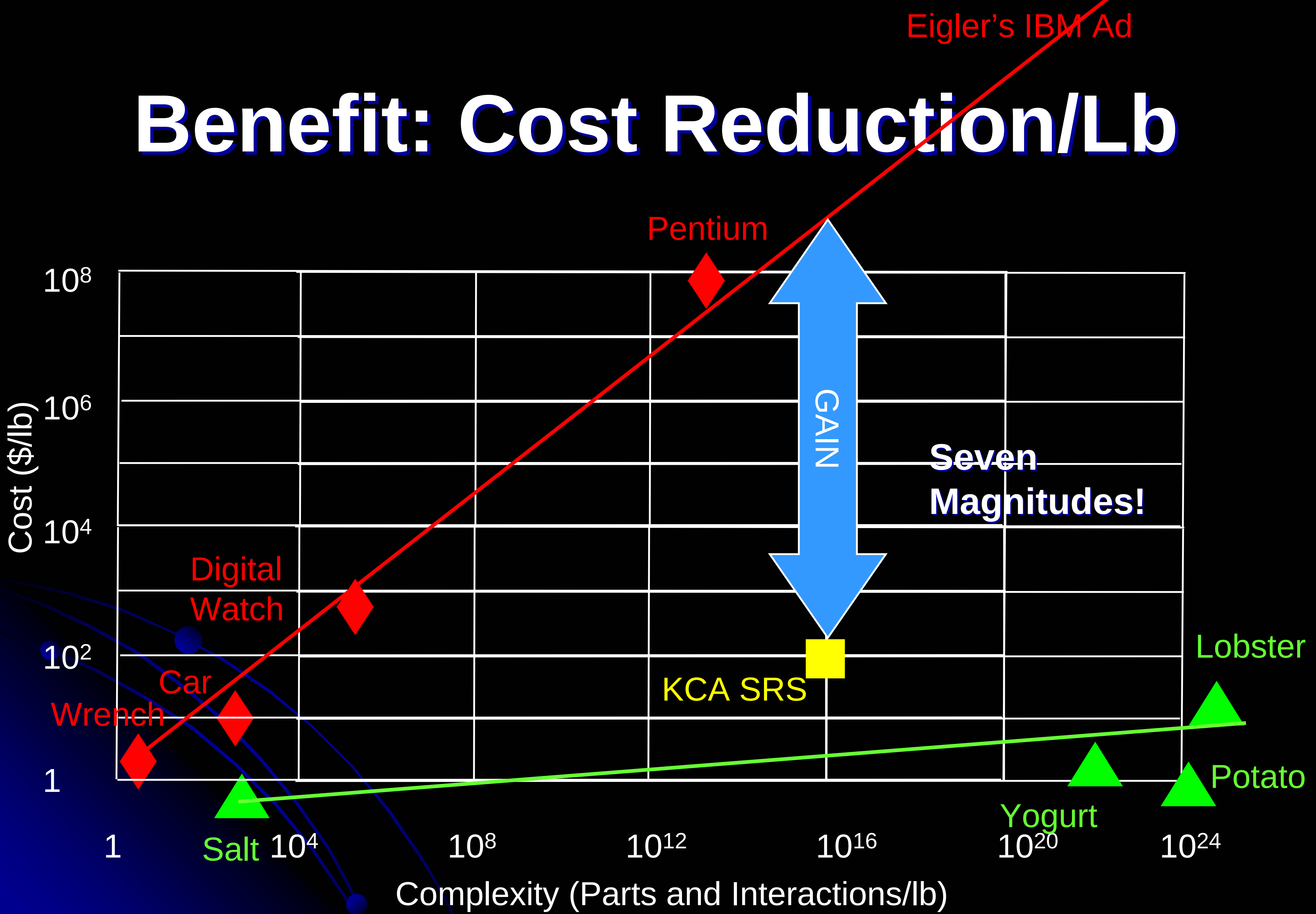
Rationale: Why Kinematic Cellular Automata (KCA)?

- Combines Von Neumann’s two designs
- Increased flexibility
- Decreased complexity
- Large system work envelope
- Sometimes better than smart dust

Rationale:
Why Both Macro and Nano Scale?

- Abstract design
- Macro:
  - Possible with current technology
  - Useful products
  - Proof of concept in short term
- Nano:
  - Quality of atoms (and molecules)
  - Self-assembled input parts possible
  - Significant financial payoff

Benefit: Cost Reduction/Lb

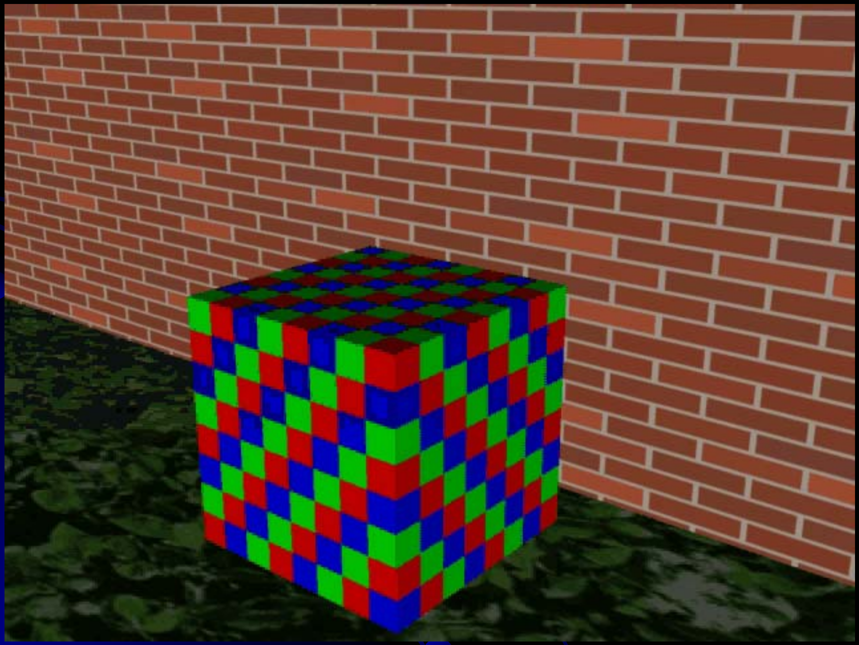


Traditional Top Down Manufacturing vs Bottom-up Molecular Replication

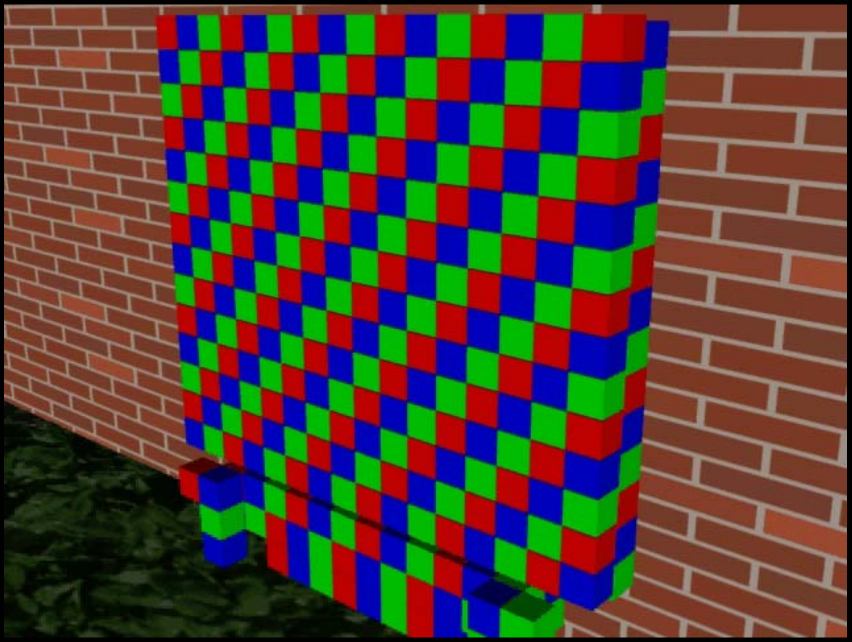
Benefit: Programmable Materials

Simple identical modules

- Flow Mode
- Pixelated Mode
- Logic Processing Mode



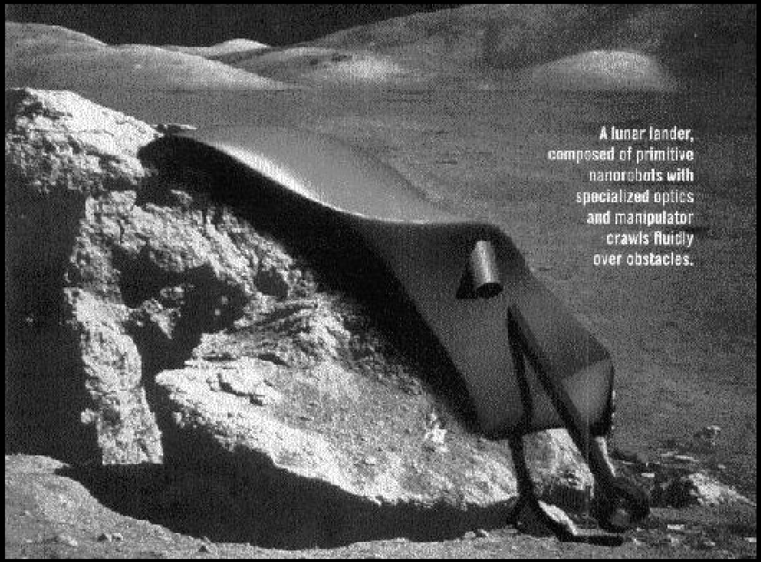
Flow Mode



Pixelated Mode

Application: Space

- Exploration
  - Robust
  - Hyperflexible
- Resource Utilization
  - Lower launch weight
  - Expandable
- Terraforming
  - Politically feasible
  - Opens new frontier



Project Goals

- Characterize self-replication
- Quantify the complexity of Self-Replicating System (SRS) made of Kinematic Cellular Automata (KCA)
- Confirm approach
- Design a KCA SRS
- Simulate designs

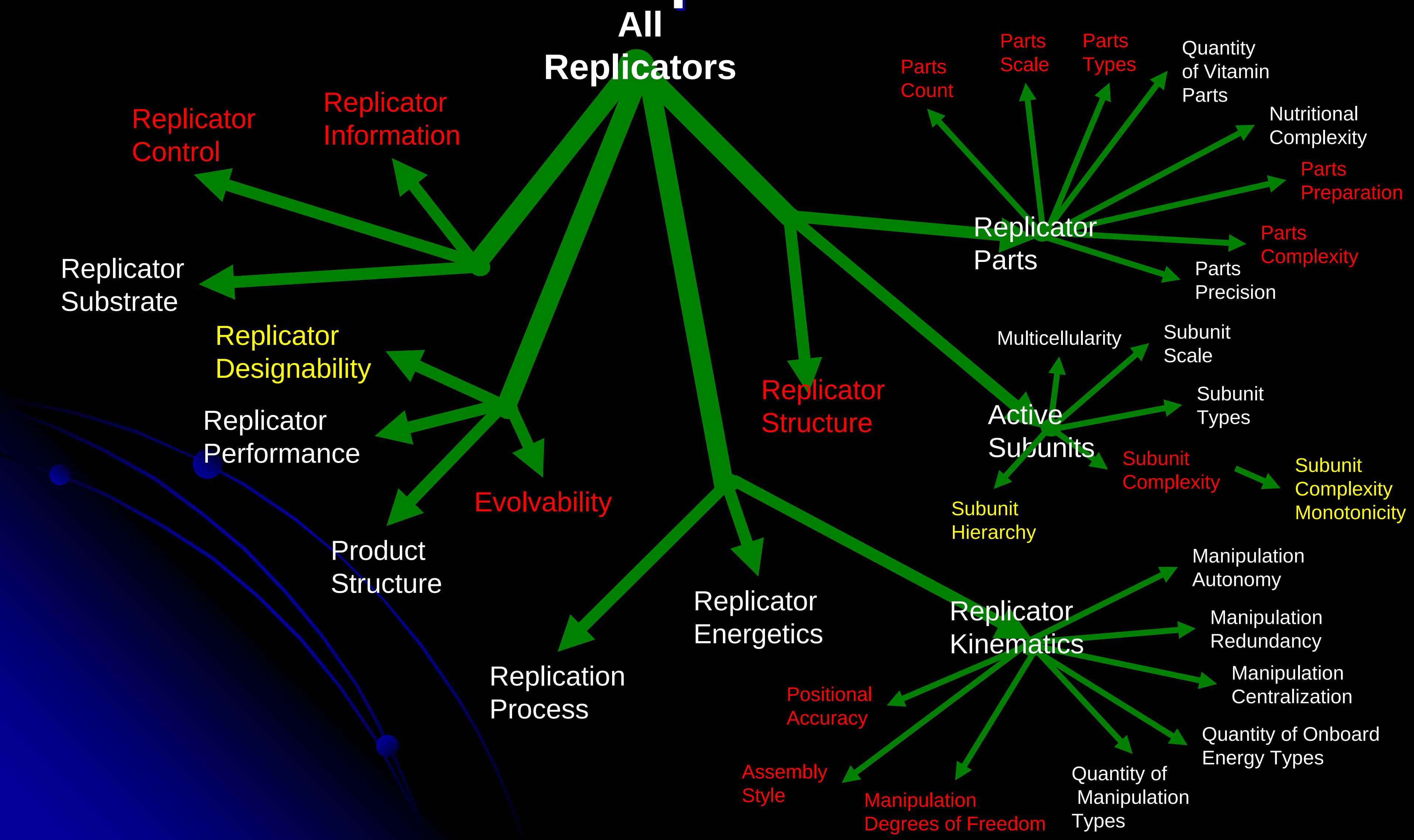
Project Strategy

- Hybridize two self-replication models
- Keep it simple
- Make it complicated
- Refine approach
- Attempt design
- Imitate computers
- Imitate biology

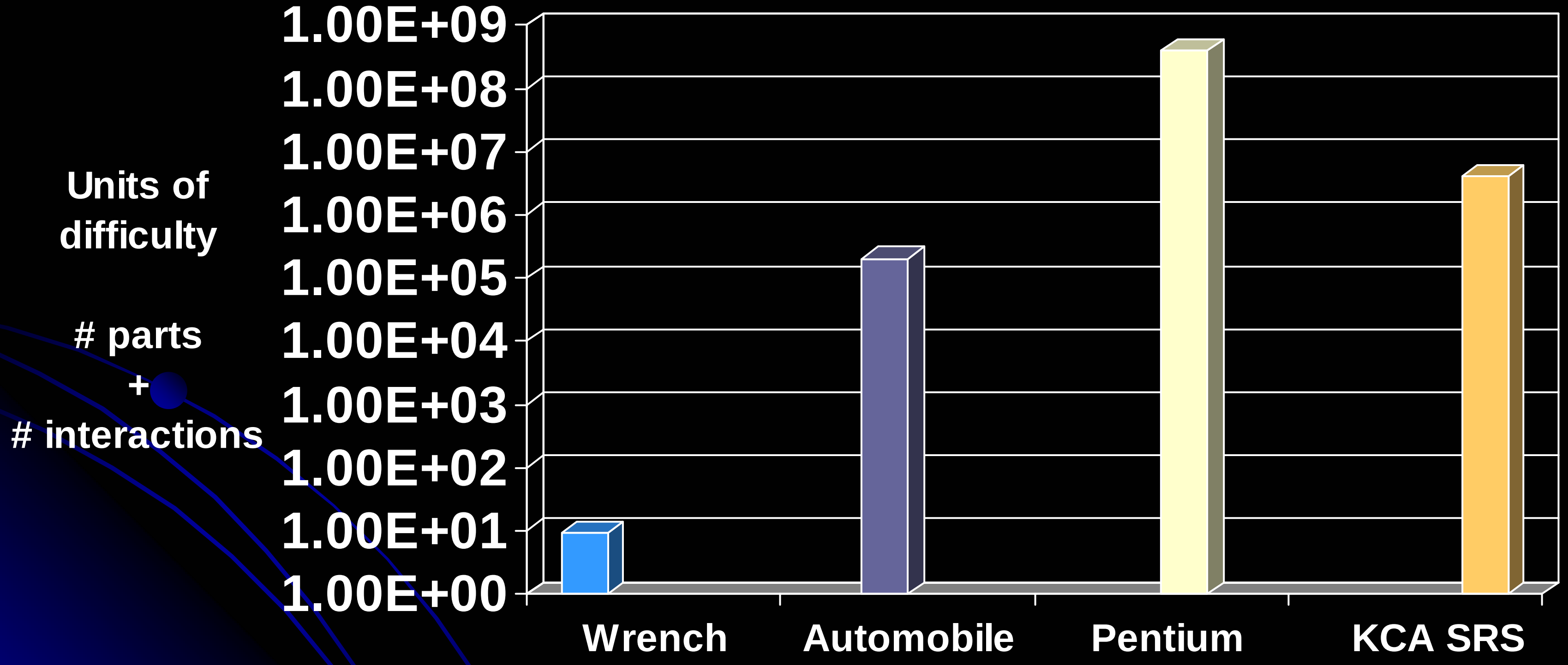
Accomplishments

Goals	Accomplishments
Characterize unexplored area	Explored Multi-Dimensional Space
Quantify the difficulty	Not trivial, but less than a Pentium
Confirm or refute approach	Refined Approach • Useful SRS • Hierarchy of Subsystems, Cells, Facets, & Parts • Transporter, Assembler, & Controller • Low-level simpler than high-level • Top-Down vs Bottom-Up • Self-Assembly for input Parts • Standard concepts • Universal Constructor is approach, not goal
Design a KCA SRS	Developed Requirements Preliminary Design
Simulate designs	Modeled Simulations • Sensor Position • NAND gate and op-amp self-assembly • Facet • Transporter and Assembler

Characterizing Self-Replication: Adjusting the Freitas/Merkle 116-Dimension Design Space



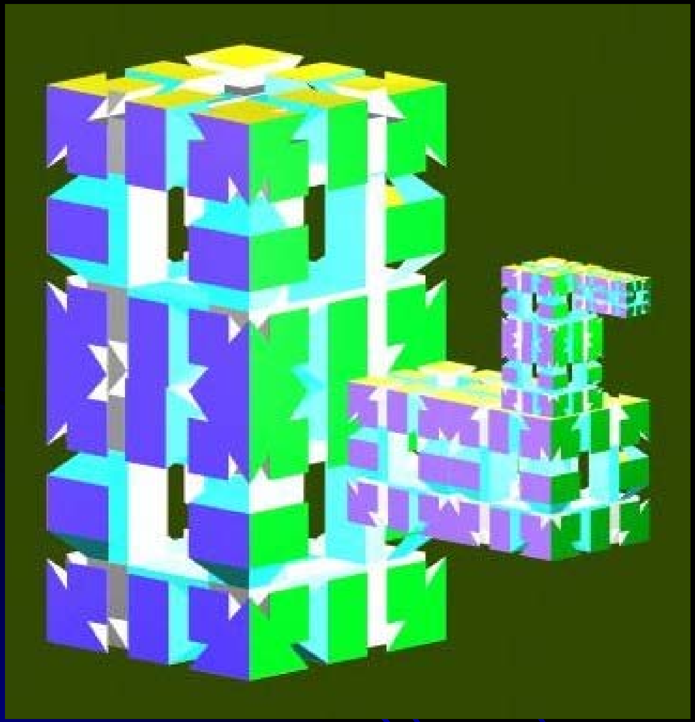
Quantifying Difficulty of SRS Design



Hierarchy

Biology	KCA SRS	Computer
Horse	Self-replicating System: Useful	Processor
Brain and Muscles	Subsystems: Transporter, Assembler, and Controller	Bus/Memory, ALU, and Controller
Cells	Cells: Cubic devices with only three limited degrees of freedom	Finite State Machines, Shift Registers, Adders, and Multiplexers
Organelles	Facets: Symmetrical implementation	
Proteins	Parts: Inert, Simpler than higher levels	NAND gates
Genes	Self-assembling Subparts: Wires, Transistors, actuator components	Transistors, Wires
Molecules	Molecules	Molecules

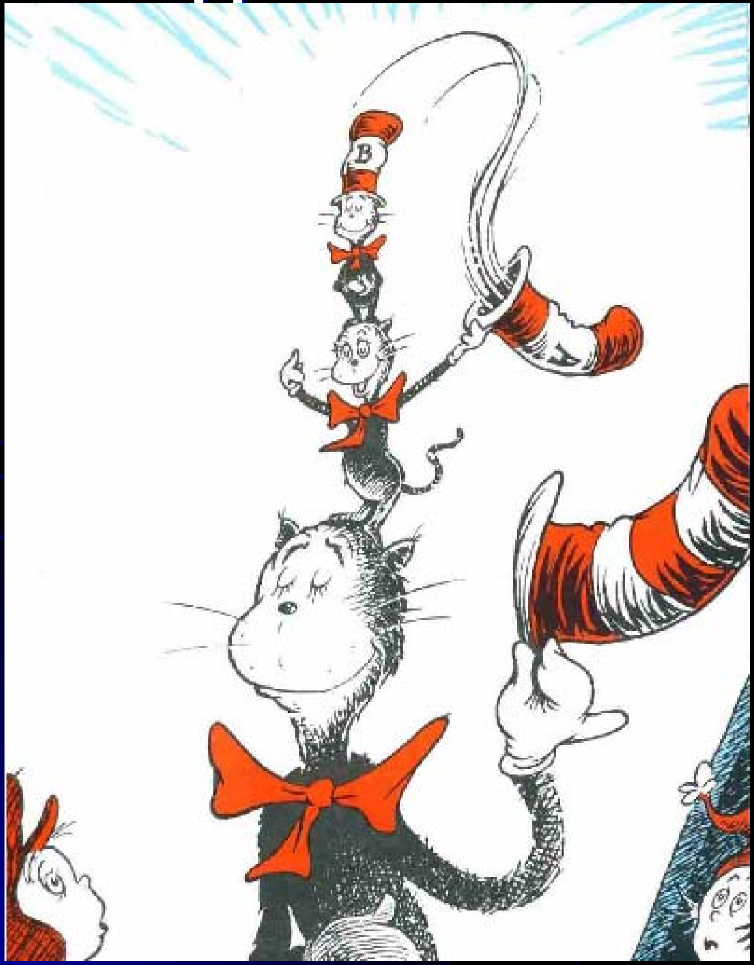
Original Approach: Feynman method

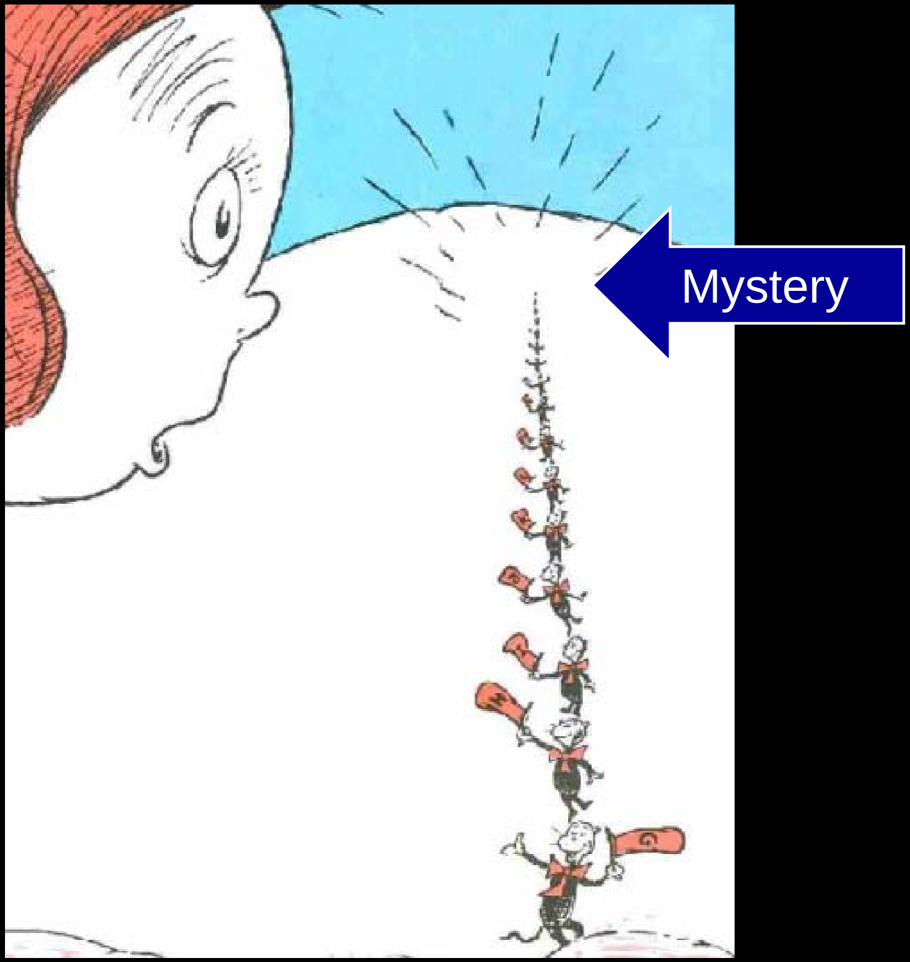


1. Start with trivial self-replication
2. Move the complexity out of the environment and into the SRS by doubling parts count of the component (Trivial^{+1} case)
3. Reiterate

Original Approach: Feynman method

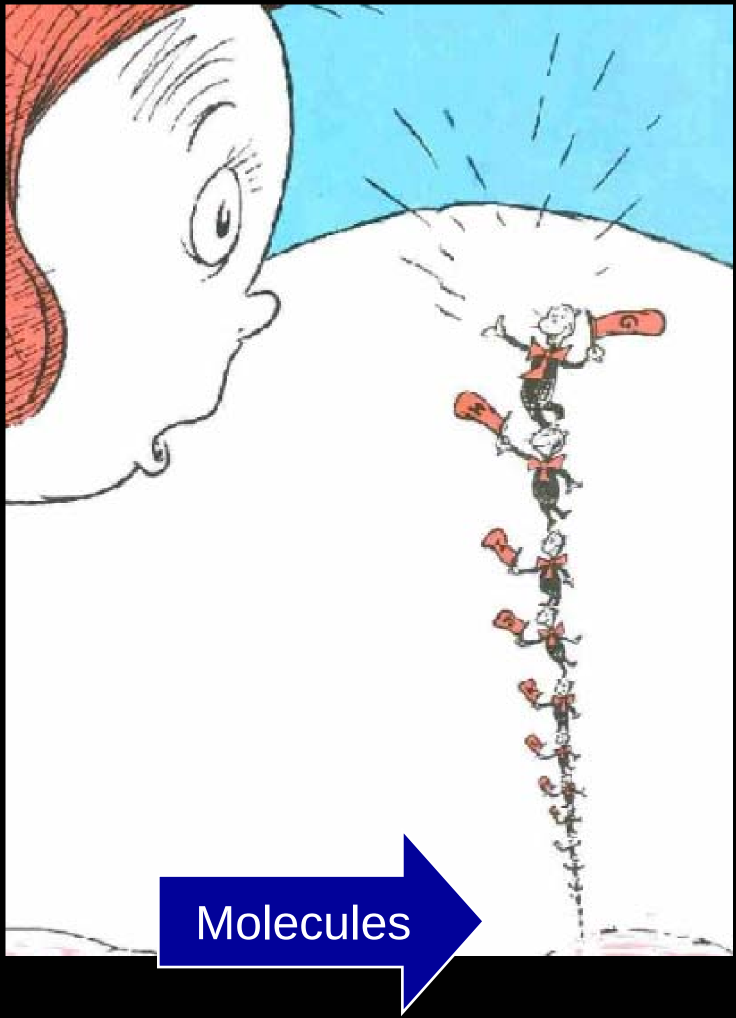
- **"Plenty of room at the bottom", top-down, fractal approach**





Refine Approach (by 180°)

- We should start at the bottom level and work <u>up</u>
- Imitate Mother Nature
- The Trivial+2 case has already been done



The Bottom-up Approach

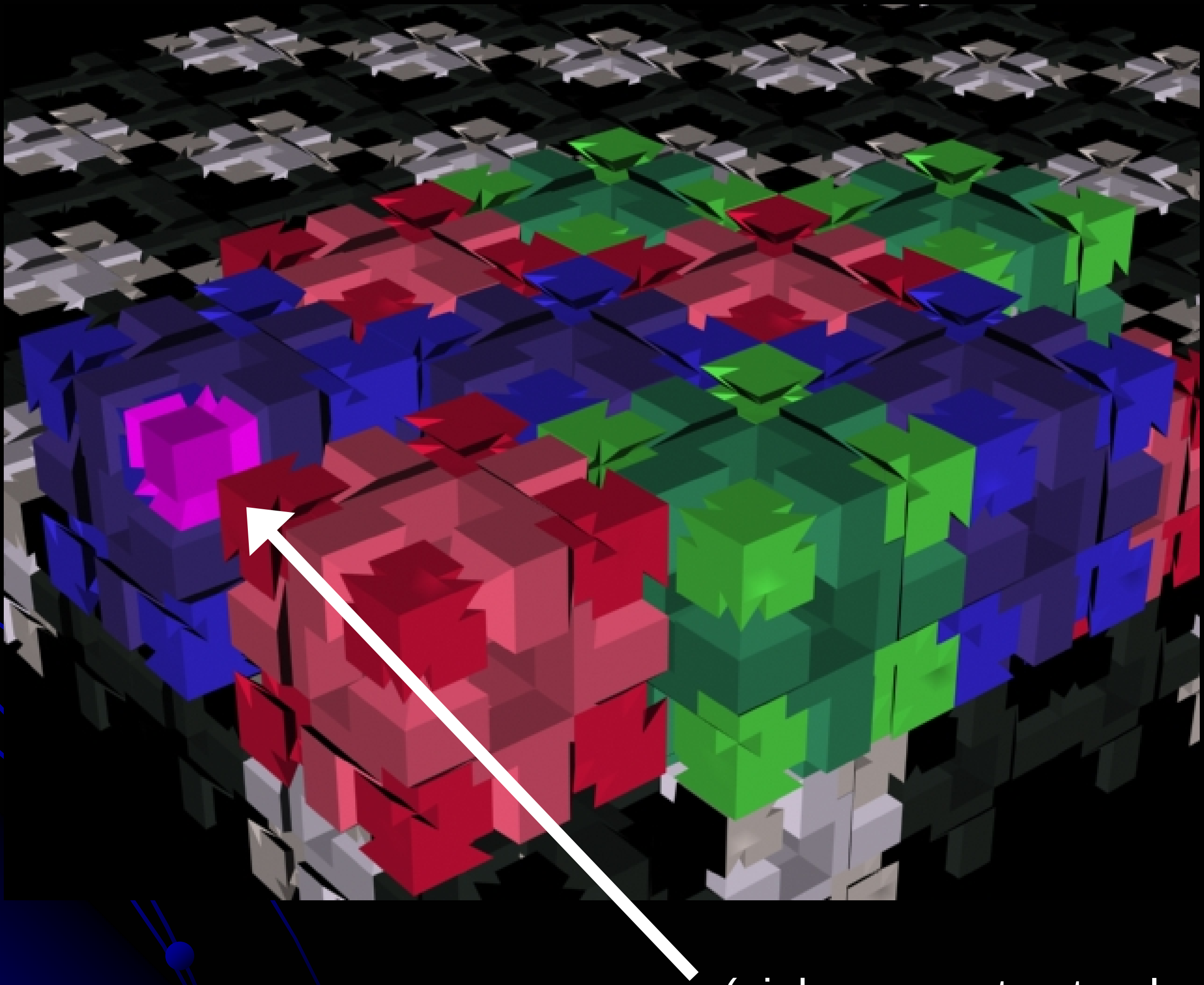
- Well-ordered environment,
- Simple inert *parts*
- Symmetric *facets*
- Modular *cells*
- Assembler, Transporter, and Controller *subsystems*
- Self-Replicating *System*

Subsystem Requirements

If atoms are analogous to bits, then:

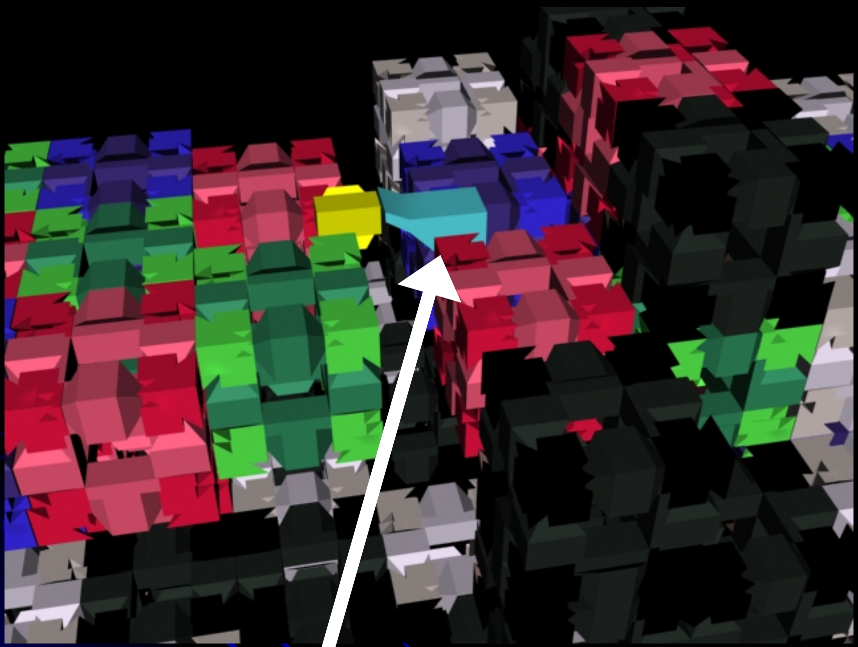
- Memory/Bus --> Transporter
  - Moves Parts
- ALU --> Assembler
  - Connects Parts
- Control --> Controller
  - Decides which Parts go where
  - Standardized

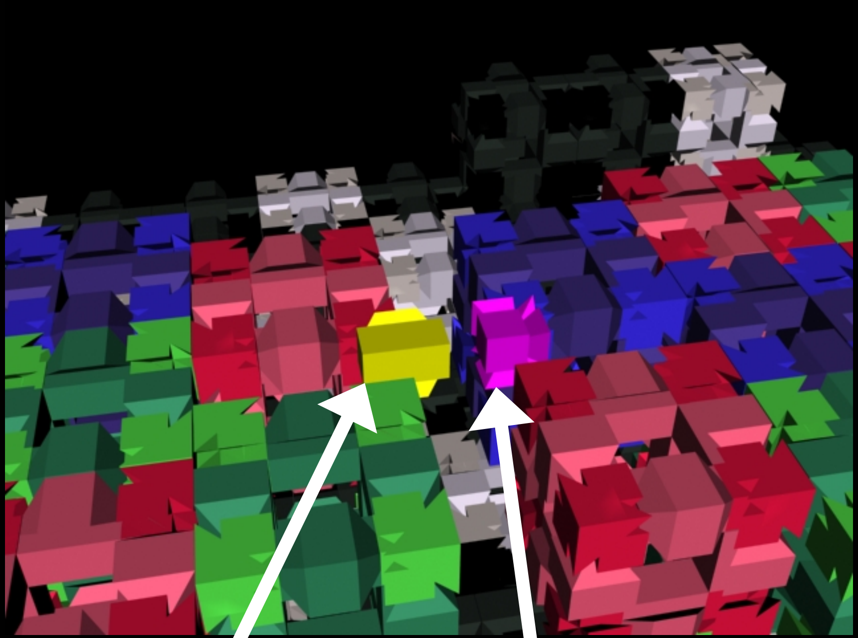
Transporter Subsystem



(pink corner structural part)

Assembler Subsystem



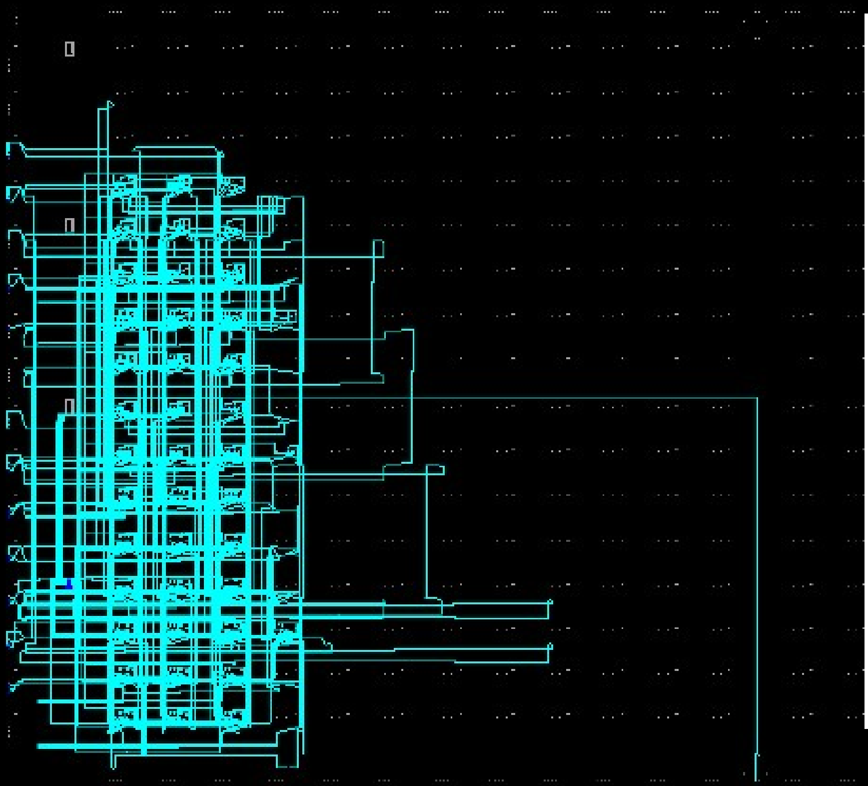


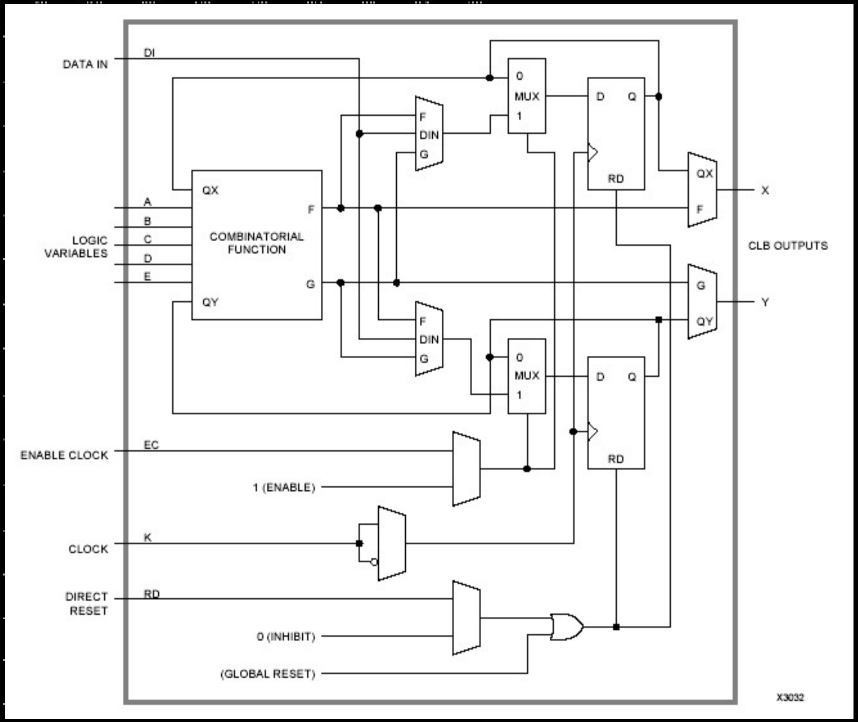
(light blue preparation tool)

(yellow edge structural part)

(pink corner structural part)

Controller Subsystem



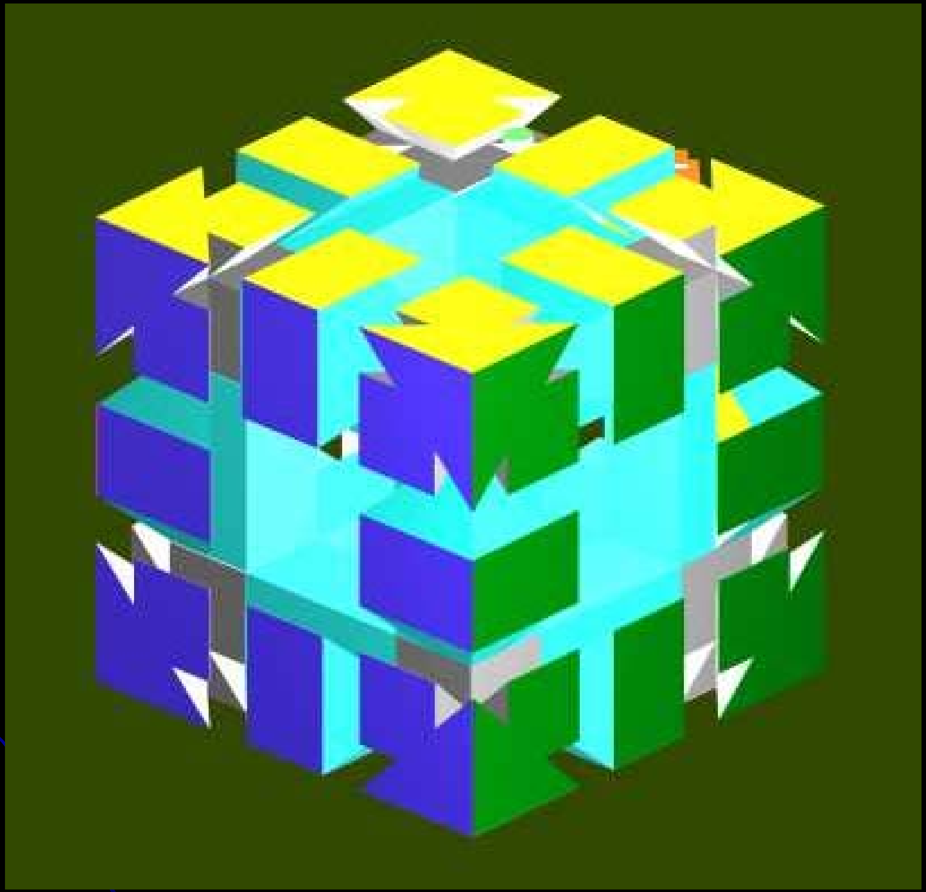


FPGA Editor View of a PicoBlaze Macro in an XC2S50E Spartan-IIE Device

Cell Design Requirements

- Structure:
  - Lock, 1-D slide, disconnect
- Actuators:
  - Transform
  - Move
- Sensors:
  - Detect Position
  - Transmit messages
- Logic:
  - Decode messages
  - Accept, store, forward messages
  - Activate commands

Unit Cell

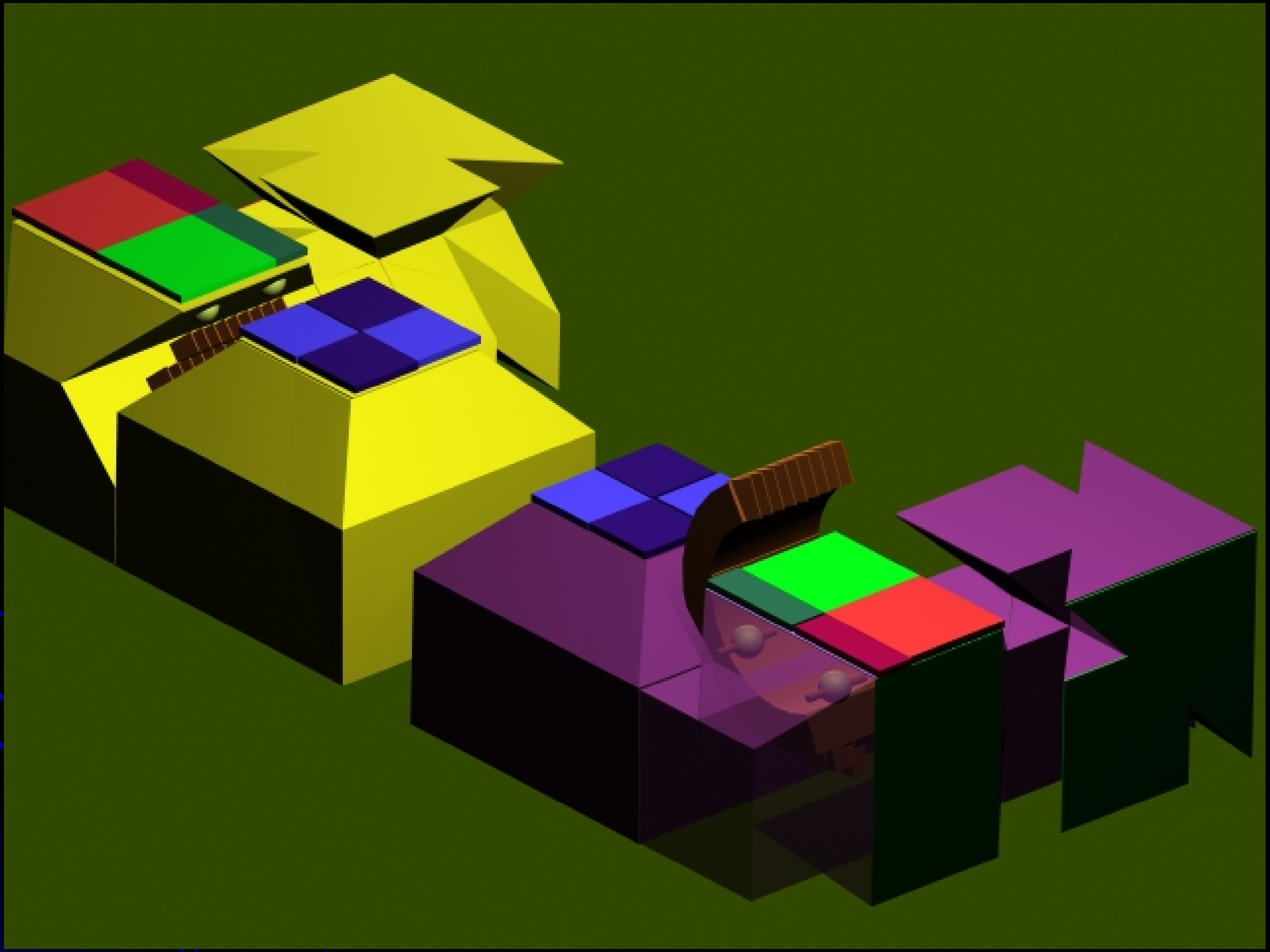


(center structure, motors, sensors, and tabs omitted)

Facet Design Requirements

- Structure:
  - Insert or retract
- Actuators:
  - Transform
  - Move tabs
- Sensors:
  - Transform
- Logic:
  - Decode

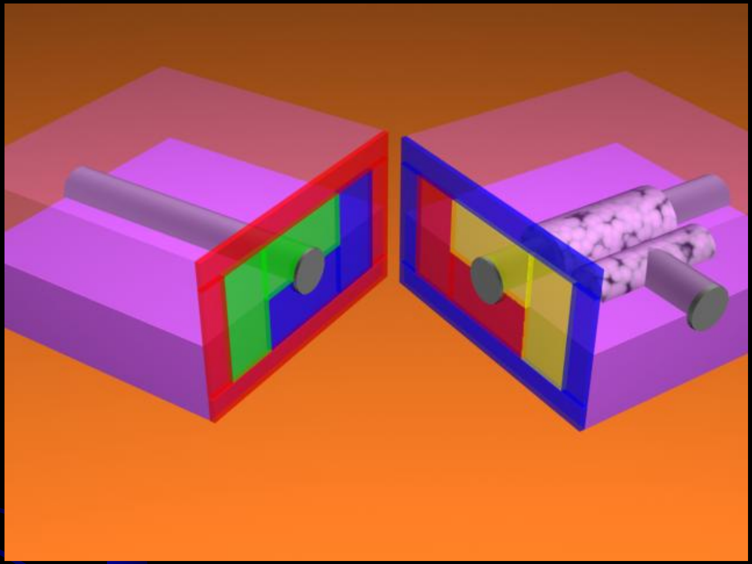
Unit Facets

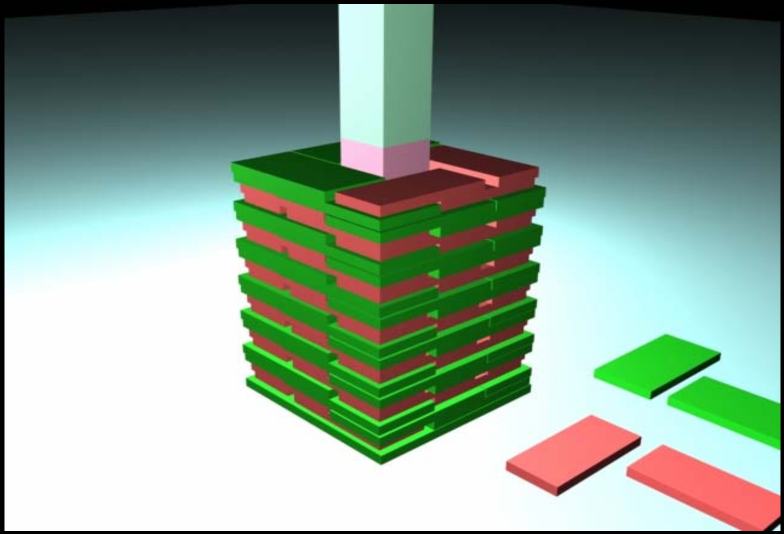


Parts Design Requirements

- **Structure:**
  - Solid
- **Motors:**
  - Rotary
  - Linear
  - IMPC
- **Sensors:**
  - Translate signals
  - Detect parts position
- **Logic**
  - Activate messages to motors
  - Aggregate digital logic

Parts: Structure, Sensors & Solenoids

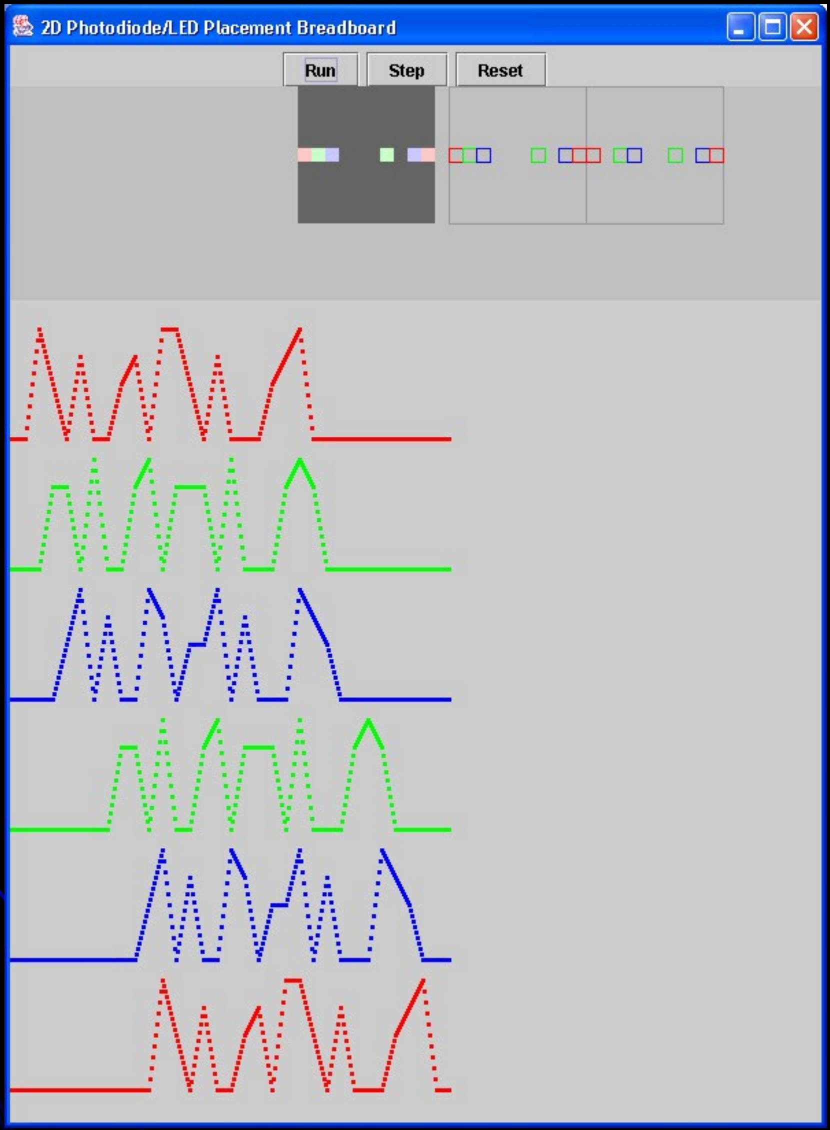




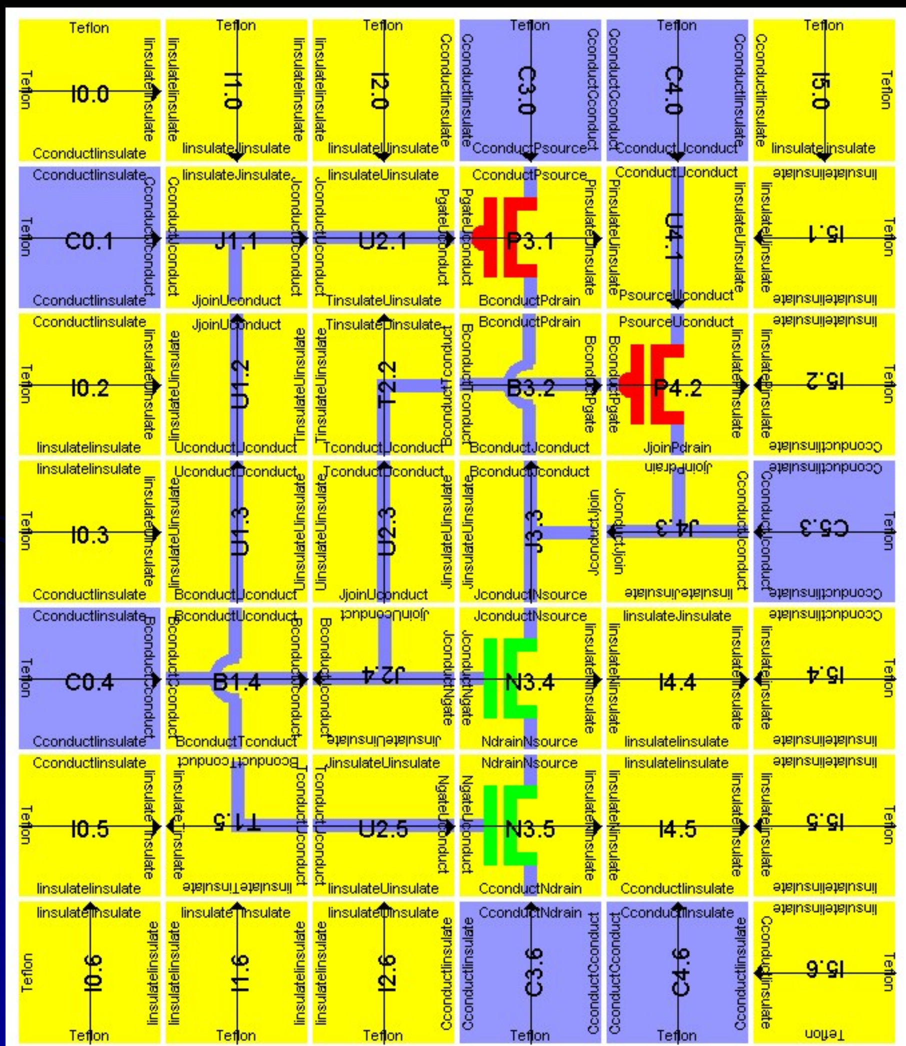
Software Simulation

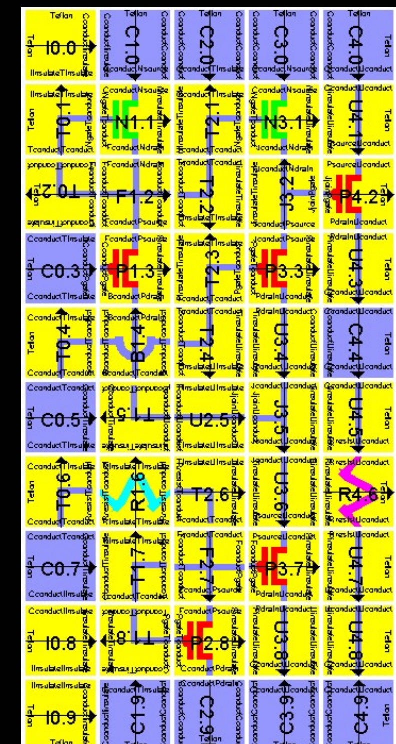
- Sensor Position Simulation Tool
- NAND gate & op-amp Self-Assembly Tool
- Facet Animation
- Transporter and Assembler Simulation

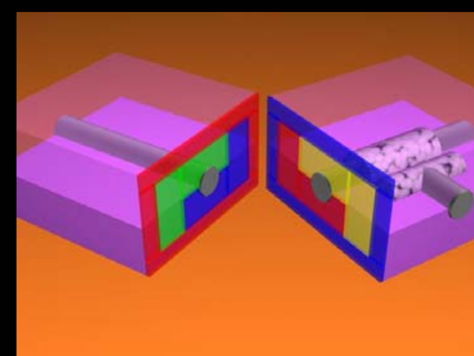
Position Sensor Simulation

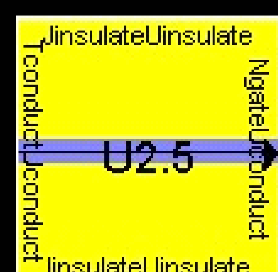


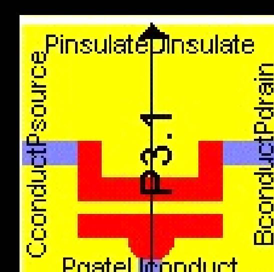
Self-assembly of NAND gate and op-amp



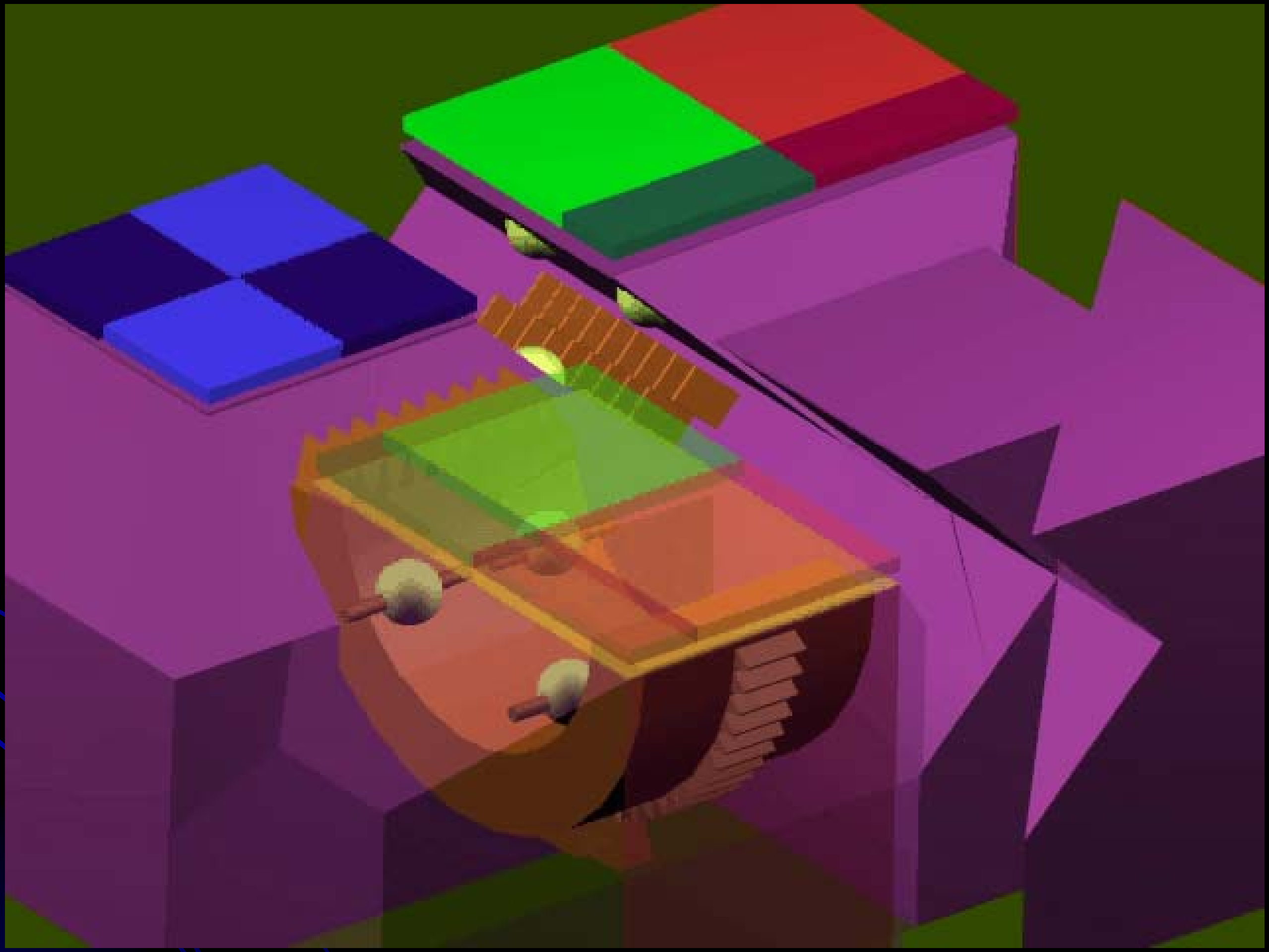




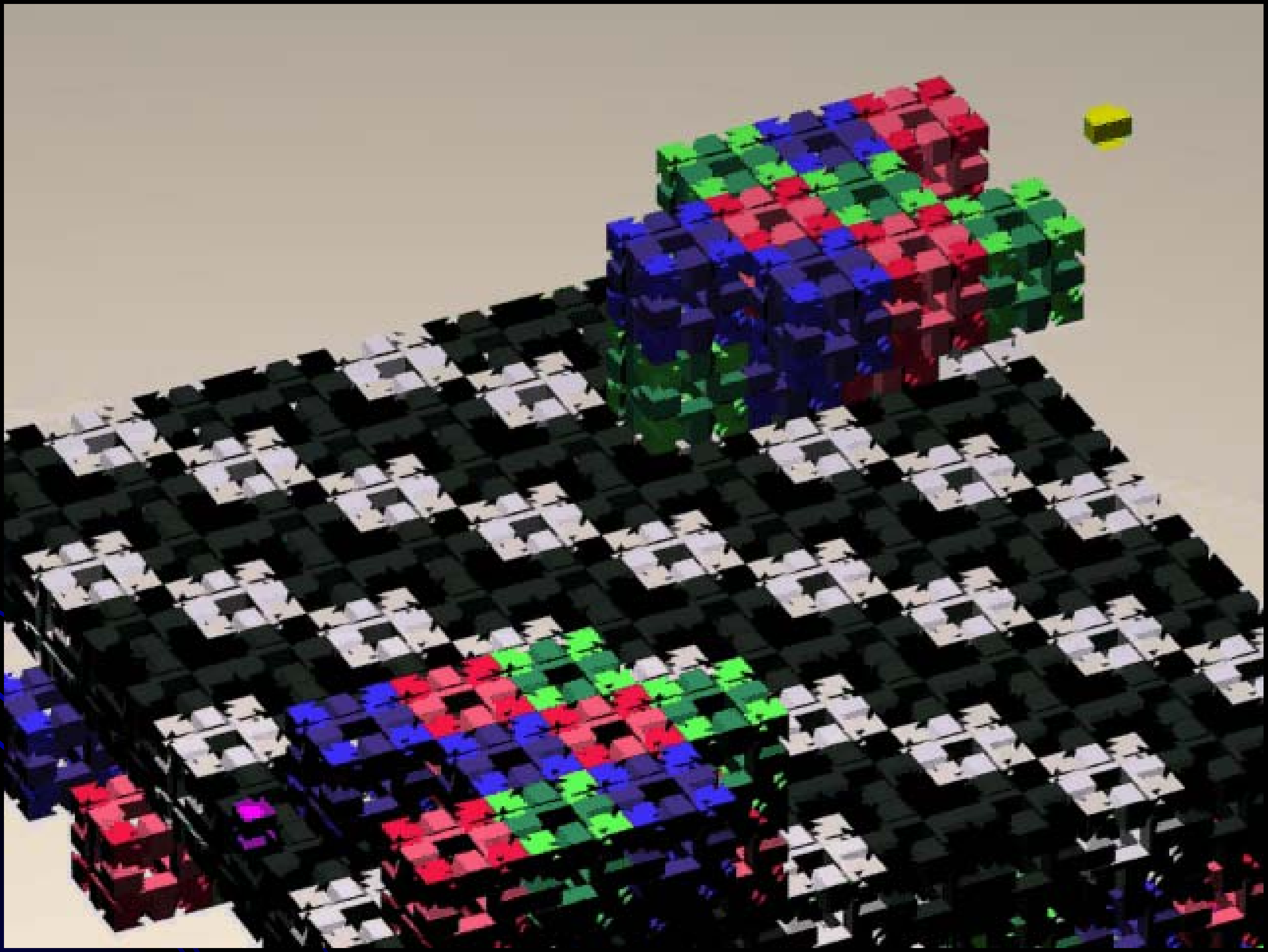




Facet Animation



Simulation of Transporter and Assembler



Conclusion and Future Directions

No roadblocks!

- Final Design for macro physical prototypes
- Build physical prototypes
- Build and run small cell collections
- Build and run subsystems
- Build macro scale SRS
- Write Place and Route software
- Refine concept at nano scale

Acknowledgements

- NASA Institute for Advanced Concepts
- John Sauter – Altarum
- Rick Berthiaume, Ed Waltz, Ken Augustyn, and Sherwood Spring – General Dynamics AIS
- John McMillan and Teresa Macaulay – Wise Solutions
- Forrest Bishop – Institute of Atomic-Scale Engineering
- Joseph Michael – Fractal Robots, Ltd.

Additional Material

- Assumptions
- Previous and Related Work

KCA SRS Assumptions

- Parts supplied as automated cartridges
- Low rate of errors detected in code

Previous and Related Work

- Freitas and Long - NASA Summer Study: Advanced Automation for Space Missions (1980)
- Michael - Fractal Robots
- Chirikjian and Suthakorn - Autonomous Robots
- Moses - Universal Constructor Prototype
- Zyvex - Exponential Assemblers
- Freitas and Merkle - Kinematic Self-Replicating Machines (2004)

Previous Work:NASA Summer Study

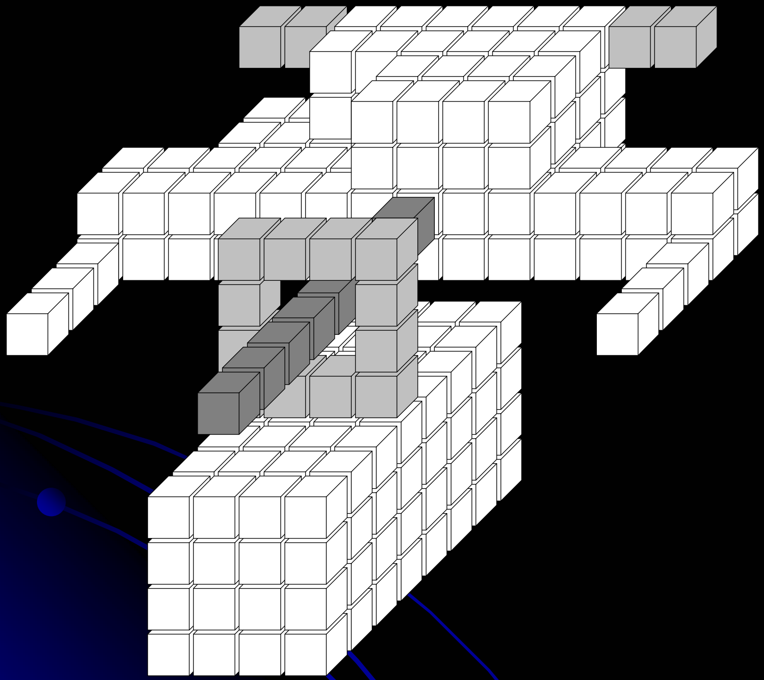
Advanced Automation for Space Missions - Freitas and Long - (1980)

- **Strengths**
  - **First major work since 1950s**
  - **Cooperation of many visionaries**
- **Weaknesses**
  - **short, no follow-up**
  - **paper study only**
  - **pre-PC technology**

FOR MORE INFO...

http://www.islandone.org/MMSG/aasm/

Previous Work: Joseph Michael



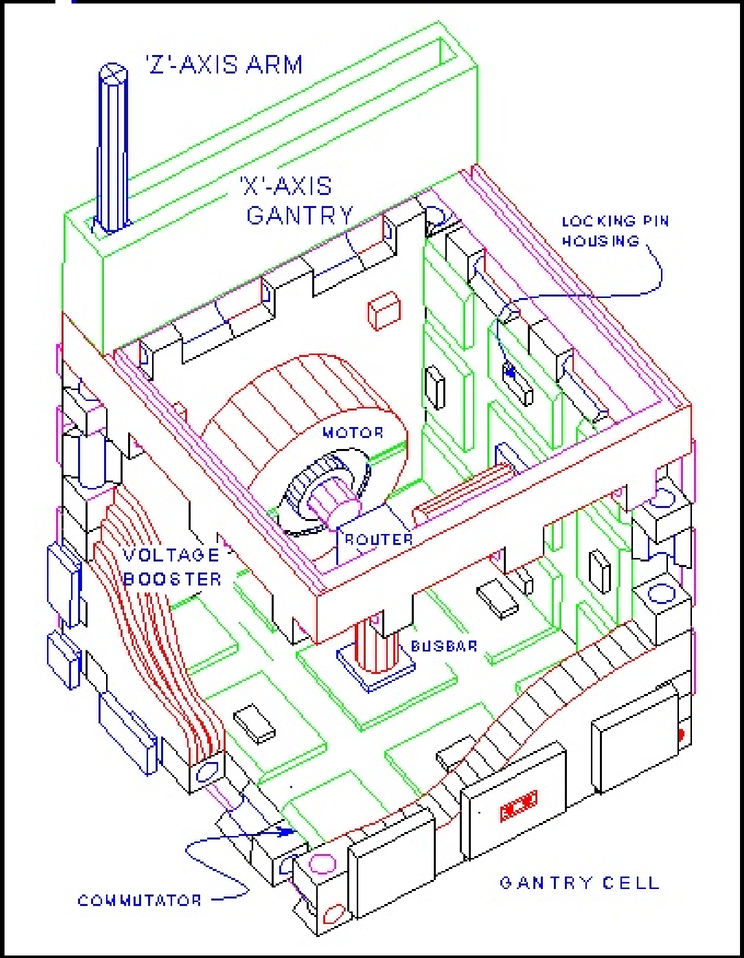
FOR MORE INFO...

http://www.fractal-robots.com/

- Strengths
  - "The DOS of Utility Fog"
  - Working macro modular robots
  - Limited DOF = better structure
- Weaknesses
  - Fractals just push problem to lower, less-accessible level
  - no detailed methodology for self-replication

Previous Work: Forrest Bishop

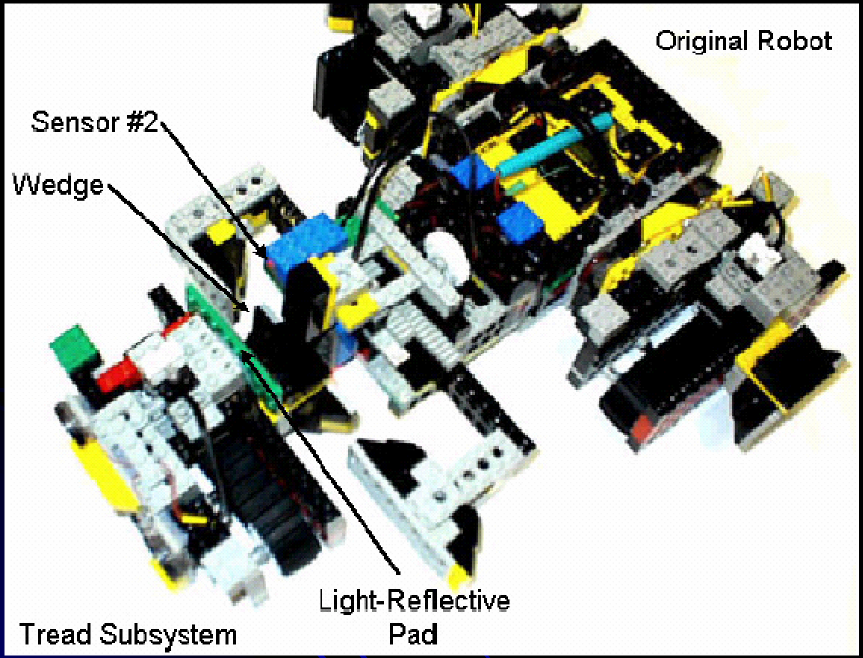
- Strengths
  - Very Limited DOF
  - Clear macro design
- Weaknesses
  - Nanoscale implementation clearly implied, but not clearly designed
  - no detailed methodology for self-replication



FOR MORE INFO...

http://www.iase.cc/html/overtool.htm

Related Work: Chirikjian/Suthakorn

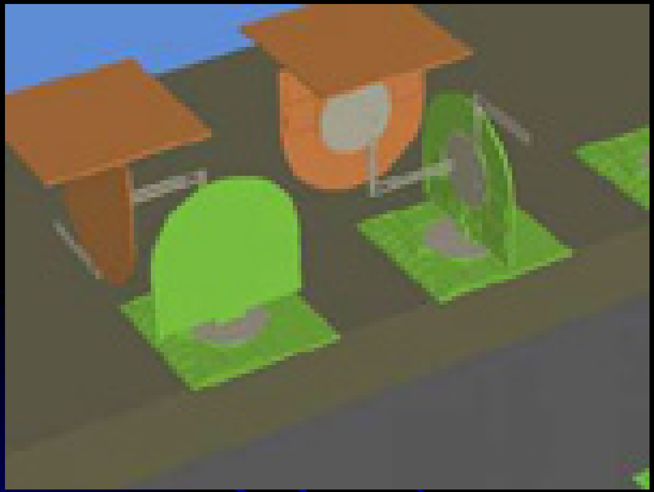


- Strengths
  - Autonomous implementation of Trivial^{+2} case (4 parts)
  - Directed towards extraterrestrial applications
  - Lego isomorphic with molecules
- Weaknesses
  - Small UC envelope
  - Depends on non-replicating jigs
  - High entropy environment limits extension to Trivial^{+3} case

FOR MORE INFO...

http://caesar.me.jhu.edu/research/self_replicating.html

Related Work: Zyvex

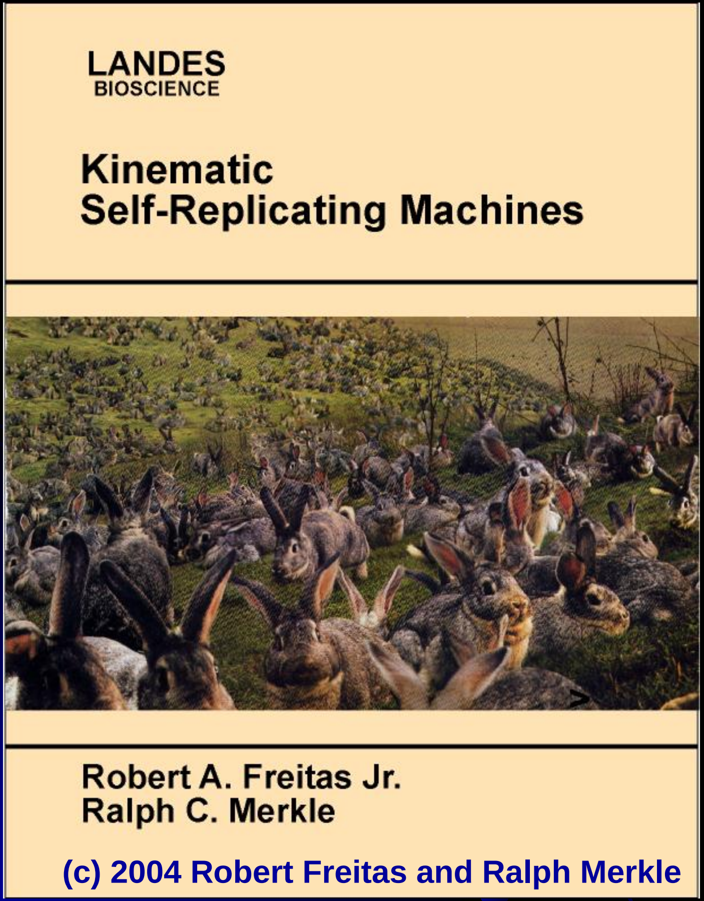


- Projects
  - Applying MEMS and nanotubes
  - Parallel Micro and Exponential Assembly
- Strengths
  - First and only funded company trying to build a Drexlerian assembler
- Weaknesses
  - MEMS is 1000X too big
  - surfaces too rough
  - Exponential Assembly is machine self-assembly (not Universal Constructor; not GRP paradigm; not Utility Fog)

FOR MORE INFO...

http://www.zyvex.com/

Related Work: Freitas/Merkle



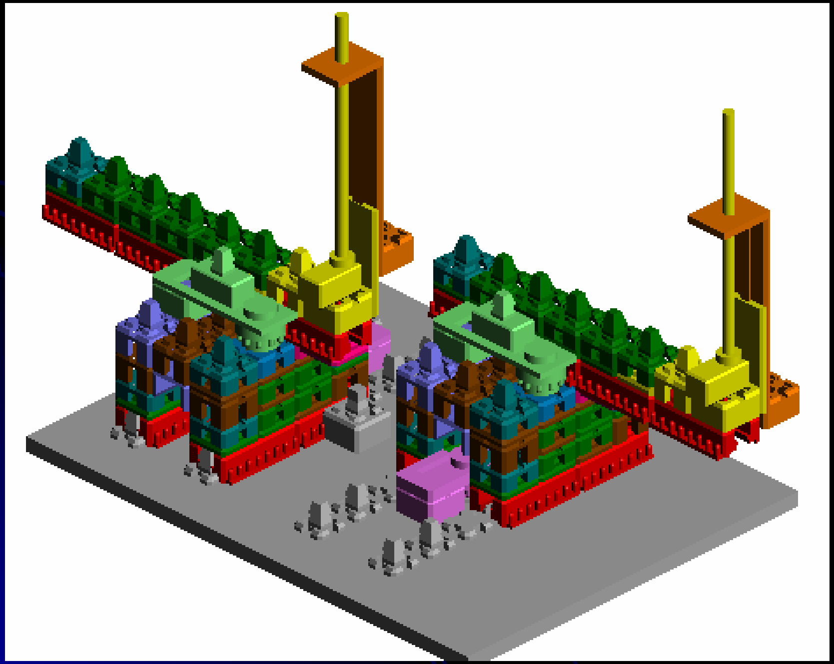
Kinematic Self-Replicating Machines
(Landes Bioscience, 2004)
First comprehensive review of field

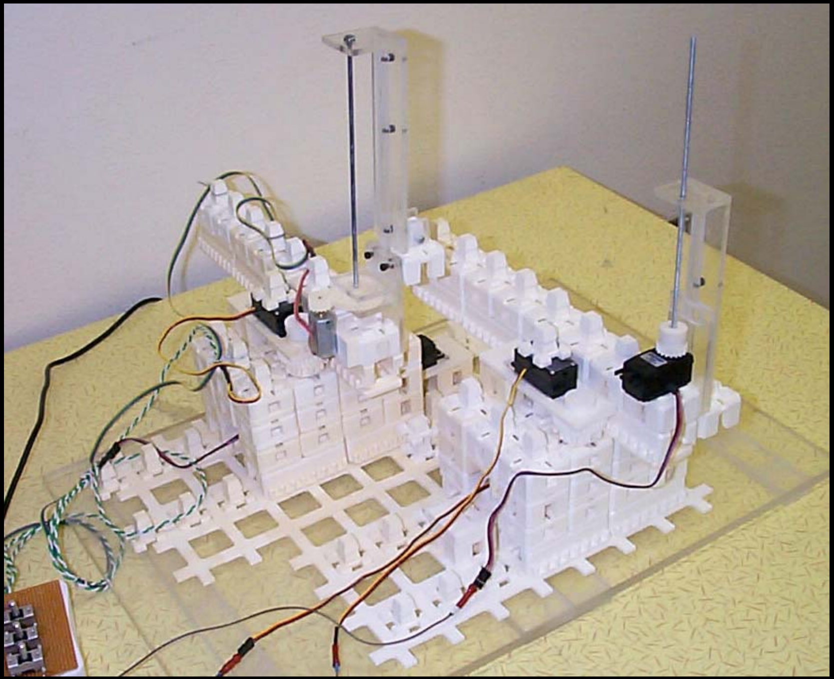
1. **The Concept of Self-Replicating Machines**
2. **Classical Theory of Machine Replication**
3. **Macroscale Kinematic Machine Replicators**
4. **Microscale and Molecular Kinematic Machine Replicators**
5. **Issues in Kinematic Machine Replication Engineering**
6. **Motivations for Molecular-Scale Machine Replicator Design**

Freitas is a technical consultant for this project

Related Work: Matt Moses

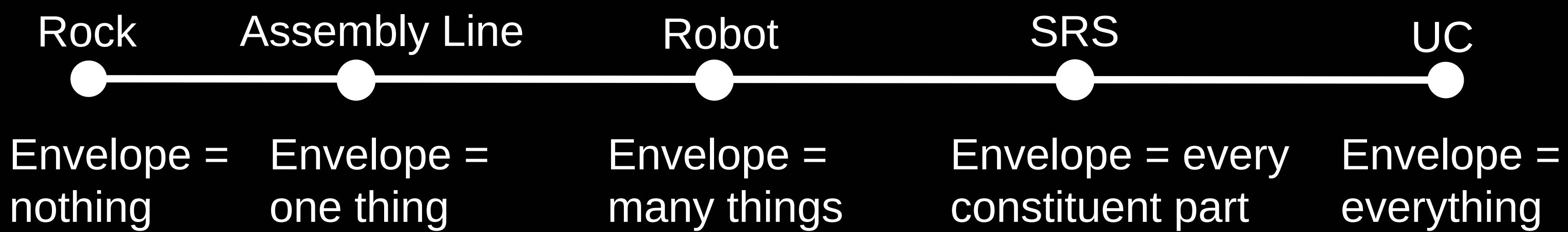
- Strengths:
  - CAD to physical implementation
  - Large envelope UC
- Weaknesses:
  - Strain bending under load
  - Manual control





Moses is a technical consultant for this project

Why Universal Constructors?



- **UC is the ability to build anything**
- **Uses “Genotype+Ribotype = Phenotype”**
- **Construction envelope includes itself**
- **Atoms equivalent to bits**
- **SRS only needs limited UC capability**