Growing Machines

Saul Thomas Griffith

B.Metallurgical Engineering, University of New South Wales, 1997

M.E. University of Sydney, 2000

M.S. MAS, Massachusetts Institute of Technology, 2001.

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences
at the Massachusetts Institute of Technology

September 2004

Signature of Author:.....
Program in Media Arts and Sciences
September 2004

Certified by:.....
Joseph Jacobson
Associate Professor
MIT Program in Media Arts and Sciences

Accepted by:.....
Andrew Lippman
Chairman, Department Committee on Graduate Students
MIT Program in Media Arts and Sciences

Growing Machines

Saul Thomas Griffith

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning,

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy.
at the
Massachusetts Institute of Technology
September 2004

Abstract

Biological systems are replete with examples of high complexity structures that have "self assembled," or more accurately, programmatically assembled from many smaller, simpler components. By comparison, the fabrication systems engineered by humans are typically top down, or subtractive, processes where systems of limited complexity are carved from bulk materials. Self-assembly to date has resembled crystallization more than it has the programmatic assembly of complex or useful structures – these systems are information limited. This thesis explores the programming of self-assembling systems by the introduction of small amounts of state to the sub-units of the assembly.

A six-state, kinematic, conformational latching component is presented that is capable of self-replicating bit strings of two shape differentiated versions of the same component where the two variants represent the 0 and 1 bits. Individual units do not assemble until a string is introduced to the assembly environment to be copied.

Electro-mechanical state machine emulators were constructed. Operating on an air table, the units demonstrated logic limited aggregation, or error-preventing assembly, as well as autonomous self-replication of bit strings.

A new construction was developed that demonstrates that any two dimensional shape composed of square pixels can be deterministically folded from a linear string of vertex-connected square tiles. This non-intersecting series of folds implies a 'resolution' limit of four tiles per pixel. It is shown that four types of tiles, patterned magnetically, is sufficient to construct any shape given sequential folding. The construction was implemented to fold the letters 'M I T' from sequences of the 4 tile types.

An analogous construction is developed in three dimensions. It is similarly shown that right-angled tetrahedrons, when folded from an edge-connected string, can generate any three dimensional structure where the primitive pixel (or voxel) is a rhombic hexahedron. This construction also suggests a concept of 3D completeness for assembly, somewhat analogous to the concept of Turing completeness in computation.

In combination, these pieces of work suggest that a manufacturing system based on four tiles, with seven states per tile, is capable of self-replication of arbitrary 3D structure by copying, then folding, bit strings of those tiles where the desired structure is encoded in the tile sequence.

Thesis Supervisor: Joseph Jacobson.
Associate Professor of Media Arts and Sciences, MIT.

Growing Machines

Saul Thomas Griffith

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning in partial fulfillment of the requirements of the Degree of Doctor of Philosophy.

Massachusetts Institute of Technology.

September 2004.

Joseph Jacobson — **Thesis Advisor**

Associate Professor of Media Arts and Sciences

MIT

Alexander Slocum — **Thesis Reader**

Professor of Mechanical Engineering, MacVicar Faculty Fellow

MIT

Shuguang Zhang — **Thesis Reader**

Associate Director, Centre for Biomedical Engineering,
MIT

Acknowledgements

To Joe Jacobson, I was accepted to come to MIT to work on printed electronics, via excursions in 3D printing at the micron and nanometer scales, work in fluidic self assembly, minimal energy surfaces and adaptive optics, and countless other distractions I was allowed to find my way to lovely discoveries in programmable assembly and the thesis presented here. It is a great mentor who allows a student enough rope to hang themselves. Thanks for the rope and helping me to tie up the loose ends just in time.

To Tim Anderson and Eric Wilhelm, 2 of the best friends a man can find. Whenever I'm flying across a frozen lake at 50mph with my butt strapped to a skateboard with 9 inch knives bolted to the trucks I want you guys there to cut my safety line.

To Dan Goldwater, thankyou for becoming a great friend, sharing the late nights, hauling my ass from the ditch on the Sabbath, and generally for being one of the most capable and humblest of people.

To Alex Slocum. As they would describe you in my country, "you're a no bullshit kind of bloke". Thanks for the guidance, inspiration and just for being you.

To Shuguang Zhang, your conversations are always stimulating and thought provoking. Most of all I appreciate your love of history and perspective in science, something not easily nurtured at MIT.

To Neil Gershenfeld, thanks for the workshop, the tools, the people, and the ethos that you have assembled in CBA.

To MIT, because grounding theory in practice is just plain cool.

To the Media Lab and it's wonderful community, because I never thought my wanderlust and A.D.D. would allow me to spend 6 years anywhere, particularly somewhere with abominable winters, and yet this place is just addictive enough.

To Joost Bonsen and Nick Dragotta, may Howtoons inspire people everywhere in the beauty of Science, Engineering, and the Art of making things.

To Laurie, you are great.

To Axel, your work is a one stop shop for inspiration and new ideas.

To Gemma, Talia, and Meg, stay wonderful and brilliant and all those things.

To Erik and Martin Demaine for some wonderfully playful and always thought provoking conversations on geometry.

To Jim McBride for assistance and idea prodding at just the right moments and getting chapter 4 over the hurdle.

To the men and women of Molecular Machines and PHM, no finer lab mates could one wish for to share the late nights and laugh away the frustrations of research. Well humoured the lot of you.

To Ben, Ben, John, Rhett, Steve, Emma, Danielle, Joseph, Manu, Kelly, Cameron, Kelly, Chris, Ted, Ike, Steve, the Norwhiskan, Limor, and all of the other names I can't squeeze here. You made the environment in Cambridge fantastic.

To Yael, Aggelos and my other room-mates over the years, sorry about the bath-tub, thanks for the parties, and better luck next time you choose a roomie.

To my UROP's Ben, Bobby, Will, Alex, John, and most of all Frank. I hope you all reach your goals.

To Linda and Pat for your patience with my allergy to bureaucracy.

To Mike for getting EVERYTHING just in time and being a bonza mate.

To Arwen, just the right distraction at just the worst and absolutely the best possible moment.

To my friends spread the world over, Jake, Gab, Dave, Mark, and everyone else, thanks!

Most importantly to my family, Pamela, Ross, Selena, and Nana for enduring a prodigal son and providing the background and support that enables me to challenge myself with a safety net of love.

Finally I dedicate this thesis to my unborn Niecephew. May you grow up in a world that matures from it's present state. Let's hope love and reason win out. Together we'll give it a push by working to proliferate both.

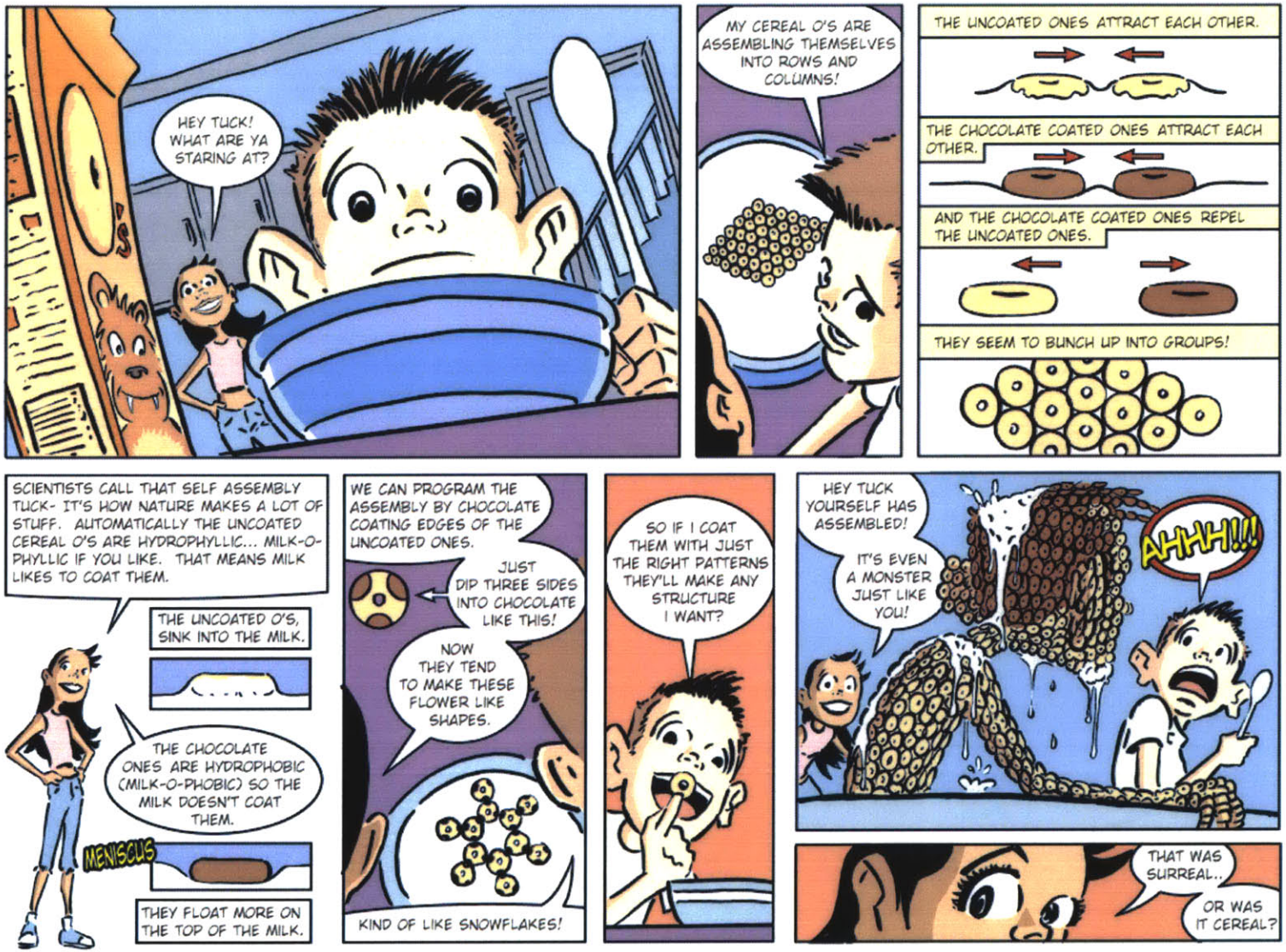


Figure 1-0. **HowToon** of Programmable Assembly.

Growing Machines

Saul Griffith 2004

1 Introduction.

The HowToon of Figure 1-0 perhaps best summarizes the goals and approach to this work. We can see all around us in the natural world compelling examples of self assembly – such as the formation of floating rafts of Cheerio-crystals in our cereal bowl- and would like to exploit similar principles to build machines, mechanisms, and useful objects, similarly autonomously. The problem is one of information and system design. What physical properties of the assembly system can be exploited as the driving forces for the assembly? - in this case menisci formed by hydrophobic and hydrophylic interactions between chocolate and milk. How do we encode enough information to build something interesting? - Celine shows Tucker that by patterning the edges of the cereal O's with alternating bands of chocolate, he can build crystals that resemble snowflakes. What would be really useful technologically though, is to know how to build far more complex objects such as the one we see in the last panel. This thesis explores how we might program and build self-assembling systems to achieve such complexity. Let's not however make monsters.

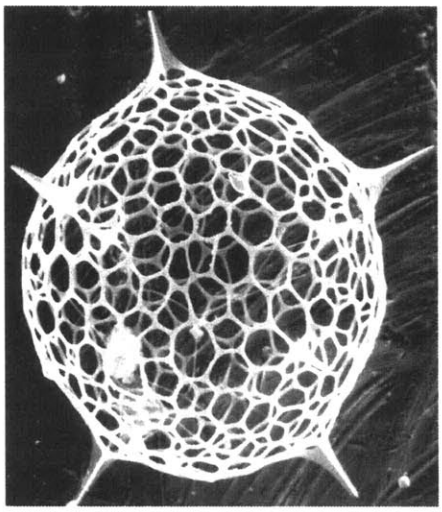


Figure 1-1. The predominantly silica skeleton of a radiolara.

Biology assembles objects of great complexity and effective design, apparently trivially, and most certainly quickly and efficiently. Biological systems exhibit the capacity for self assembly, self repair, logic, reproduction and high complexity[1,2]. Radiolara (Figure 1-1) and diatoms[3,4]. are typically 60-500

---

[1] Thompson, D. A. W. and L. L. Whyte (1942). On growth and form . Cambridge Eng., The University Press.
[2] Bard, J. (1990). Morphogenesis : the cellular and molecular basis of developmental anatomy . Cambridge England ; New York, Cambridge University Press.
[3] Flachentragwerke, I. f. l., Ed. (1984). IL28 Diatoms 1, Shells in Nature and Technics, Universitat Stuttgart.
[4] Flachentragwerke, I. f. l., Ed. (1990). IL33 Radiolaria, Universitat Stuttgart.

microns in largest dimension, with patterning of the silica or $CaCO_3$ features down to 10nm. The organisms that produce these intricate, inorganic, skeletons are amongst the most plentiful vegetable matter on earth and are ubiquitous in fresh and salt water. Figure 1-2 shows artificially colored E.Coli cells dividing and replicating. These two examples from biology exhibit behaviors hitherto unseen in human engineered systems – 3D patterning of complex structure with nm resolution, and self-replication. More impressively still, the processes occur at basically ambient temperatures and in a large range of materials. By comparison, human manufacturing typically requires high temperature or high energy processing steps, and high complexity is achieved either by top-down, computer driven control, or laborious manual assembly.

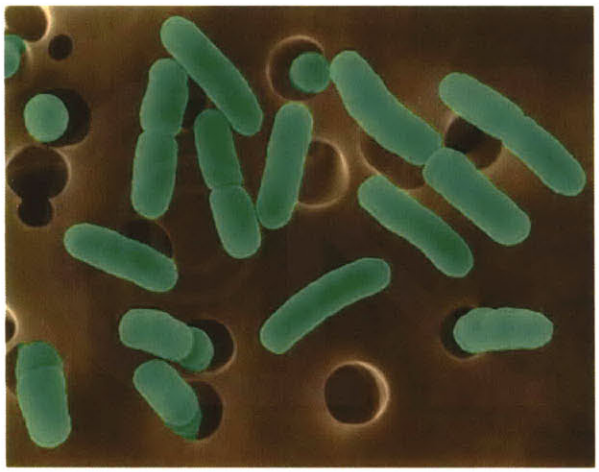


Figure 1-2 E.Coli. cell division and replication

1.1 Growing Machines – Why?

This thesis was born of a frustration with existing construction techniques at the nm-micron scale, and through a fascination with the elegance of the methods for fabrication seen in biology. The work was set out as a design challenge: design a 'minimum' set of components capable of arbitrary 3D structure formation, and of self-replication, by 'self-assembly' – that is assembly without directed human intervention.

Particularly as technologists pursue smaller, more complex, and three-dimensional structures, traditional manufacturing schemes will likely not scale to meet the demands. The ability to have self assembling systems perform error-prevention, error-correction, logic, self replication, and self-repair, is what will enable new modes of manufacture such as exponential manufacture by replication of part assemblies. As the polymerase chain reaction (PCR) has proved of incredibly high value to biologists in amplifying the production of DNA by replication it would be similarly valuable to micro and nano-technologists to be able to copy non-biological objects by replication.

1.2 Serial vs. Parallel assembly

We traditionally associate a serial process with the concept of assembly - a human hand or robot grasping a component or a sub-assembly and affixing it to another component or assembly until an entire device or machine is furnished. Assembly by this mode either requires 1) a highly intelligent robot (or human) to find parts, orient the parts, place them correctly, and fasten them, or 2) highly constrained input components or assemblies where the orientation and position is fixed relative to the assembly co-ordinate system and therefore the control problem of placement and fixing is simplified. The cost of adding more hands, or robots, is often prohibitive. Inherently this type of assembly process is serial, or stepwise, and parallelism is limited to the number of manipulators.

Self assembling processes on the other hand can be seen to have two core properties amenable to parallelisation of the assembly process.

1. Random or pseudo-random interactions between large numbers of components.

2. Asynchronous formation of assemblies in parallel.

These properties imply that parallelism is limited not by the number of manipulators, but by the number of parts in the bin, and by the number of components that act as the nuclei for construction. We will see these properties in the assemblies formed in chapters 4,5 and 6.

1.3 3 Dimensions

At the macro scale, or scales larger than that of human fingers, many techniques have been developed for the fabrication of complex 3 dimensional (3D) machines. Sub-assemblies are ganged. The modern motor vehicle has as many as 100,000 individual components and is an inherently three dimensional machine. Below the scale of millimeters however, very few techniques have been developed for assembling complex 3 dimensional structure, and in the micro and nanometer domain human fabricated structures are almost exclusively 2 dimensional, or at best 3D where the third dimension is achieved by the stacking of 2.5 dimensional layers. This is the domain of micro-lithography processes and the inherently constrained geometries that they produce. Recent work by Clark[5] (2001) has demonstrated 3-dimensional assemblies of thousands of micron scale parts. The complexity of these assemblies is low, in that they are periodic crystals, and the errors are high – as crystal defects - but the number of components is impressive, and it illustrates parallel assembly into ordered structure from random interactions.

---

[5] Clark, T. D., J. Tien, et al. (2001). "Self-assembly of 10-mu m-sized objects into ordered three-dimensional arrays." Journal of the American Chemical Society **123**(31): 7677-7682.

1.4 Replication

We routinely copy, or replicate, digital information. It enables convenient and fast manufacture of computer programs or data. Can this methodology be extended to manufacturing where it would be a convenient method for making complex devices where the first is prototyped and debugged at great length before being copied for manufacture?. There are many philosophical, environmental, ethical, and other debates surrounding both the definition and the implementation of replicating machines which I shall avoid here – merely pointing out that replication as a manufacturing process is already in wide use today. PCR, or the Polymerase Chain Reaction, is a method by which strands of DNA may be artificially replicated by cycling through a denaturing, annealing, and extension cycle.

1.5 A model system for studying biological assembly?

Perhaps in building a replicating, structure forming hierarchy from the ground up we will also learn something about biology and the 'choices' it made to do things the way it does. The work in this thesis is not slavish to biological methods for fabrication, quite the opposite. Rather it looks at how we might achieve similar functionality given the constraints of manufacture that we have in inorganic systems.

1.6 Self Assembly

There is a growing body of work in self assembly[6,7,8]. As the reader contemplates the state of the art presented in that work they will see the extremely limited structures available in existing self-assembly schemes. We can look at biology and the natural world for many inspirational examples of self assembly, and it is precisely this look at biological self assembly that illuminates the limited nature of human made self-assemblies, and of the requirement for more study and invention in this arena to enable higher complexity articles. It is too soon to determine whether self assembly outside of organic or biological systems will ever prove useful. Studying this process however is a natural extension of the human desire to understand the structure and function of the world around us. It will hopefully provide new perspectives on biological self assembly and inspire new directions in manufacturing.

---

[6] Terfort, A., N. Bowden, et al. (1997). "Three-dimensional self-assembly of millimetre-scale components." Nature **386**(6621): 162-164.

[7] Bowden, N., I. S. Choi, et al. (1999). " Mesoscale self-assembly of hexagonal plates using lateral capillary forces: Synthesis using the "capillary bond"." Journal of the American Chemical Society **121**(23): 5373-5391.

[8] Oliver, S. R. J., T. D. Clark, et al. (2001). "Three-dimensional self-assembly of complex, millimeter-scale structures through capillary bonding." Journal of the American Chemical Society **123**(33): 8119-8120.

1.7 The Problem: Information in Structure

The problem which I seek to address may perhaps be best described as the effort to rationally design for symmetry breaking in self-assembly. By rationally I mean apriori design of a fixed, repeatable entity. By symmetry breaking I mean a non-trivial structure, an arbitrary 3D object as opposed to a regular, repeating, crystal structure. This problem might also be described as the following series of questions that seek to outline the relationship between information and structure in physically assembling systems:

> *"How does one encode enough information within an assembly of parts to specify (exactly) the resulting structure?*
>
> *"What is the minimum amount of information, and hence the number of unique part types, and the amount of state, or information within each of those parts, to specify a desired structure or type of structural behaviour?*
>
> *"How does one implement this information, and the logical steps of assembly it implies, in an autonomously proceeding assemblage?*
>
> *"Can interesting assembly protocols be implemented in logic simple enough to be executed without microprocessor control, ie. in systems too small or numerous to consider digital computation or on-board power?"*

1.8 Solution: Programmable Assembly

To date all designed self-assemblies have effectively been crystallizations, or ordered arrays of similar components. These ordered arrays tend also to be imperfect crystals. For self-assembly to be a practical manufacturing device for anything more than arrays (such as photonic crystals) symmetry needs to be broken. A system that relies purely on pre-patterned edge interactions of tiles or components will require a very large number of different components[9] to construct an object of any complexity. This realisation has led to the approach taken in this work: to develop a system based on a minimal part set that can achieve the behaviour that is desired: replication and arbitrary 3D structure. The approach taken is the addition of state to individual tiles in order to implement longer range ordering and patterning. State can also be used to prevent detrimental errors in the assembly.

One issue tantamount to programming assembly is that the instruction set or data must be stored in memory somewhere. One approach is that each component in the assembly carries all of the instructions for the entire assembly in a similar way that every cell contains the complete DNA code. The approach I explore here is storing the code in the structure itself - similar to DNA, or proteins. This implies that to make multiples of a given assembly, one must copy the instructions directly from

[9] Private communications with Erik Winfree and Joe Jacobson.

the structure, not just transfer the information from some digital memory. This leads to questions of assembly analogs to sub-cellular processes.

1.9 Why not purely simulate?

An immediate criticism of this work to the casual reader could be 'why not just simulate programmable assembly?" Indeed this is a good question. There are even software packages such as SWARM, specifically designed to simulate asynchronous agent-based computation. It is my assertion, and a brief reading of the literature cited in chapter 2 should give the reader a similar perspective, that although a good deal of work has been done in simulating systems such as these, they are generally underconstrained as simulations, and therefore are not modeling real, implementable, systems for assembly. I would anticipate that this thesis will inform future simulation work and suggest how the existing simulation and design tools need to change to reflect the ever so harsh constraints of implementation in the physical world.

1.10 Architecture of this thesis.

Chapter 2 of this thesis will introduce enough previous work across fields from DNA computation to meso-scale self assembly and mechanical engineering, to ground the reader in the state of the art and suggest why this piece of work is new and (hopefully) interesting.

In chapter 3 I will define programmed assembly for the purposes of this thesis, and use this chapter to lay out what role small amounts of state could have in programmed assembly. I will touch upon the constraints that make physical assembly different from cellular automaton models, and other constraints relevant to implementation.

In chapter 4 I develop a construction that demonstrates that non-intersecting folding of a 2D string of vertex connected squares can produce any 2D pixilated structure (where a pixel is one of the connected squares). I then show that a similar construction demonstrates that the folding of an edge connected string of right-angled tetrahedra can produce an arbitrary 3D voxelated structure. These constructions imply a resolution limit to the structure, and a concept of '3D completeness' for a manufacturing system.

In chapter 5 I build on the work of Lionel Penrose in the development of kinematic self-replicating machines and demonstrate that a self replicating system capable of copying arbitrary 'bit strings' can be implemented in 6 states. I shall discuss the difficulties in designing such systems to proceed autonomously.

In Chapter 6 I describe the design and building of 'electromechanical emulators' for the study of small program state machines in self assembly. These emulators are assembled on a 2D air-bearing (air-hockey table) environment.

Finally in Chapter 7 I will draw all of these pieces together in conclusion and suggest the further work that may bring some of these ideas closer to reality.

2 Background and previous work

This thesis draws upon work from a number of different fields. As self-assembly is in its infancy as a method of fabrication I will take the liberty of an extended background section to elucidate the different methods and concepts available in self-assembling systems such that it can form a brief literature background for future workers in this area. I have broadly assembled the background literature into the various disciplines that have both informed this work, and will be informed by it.

2.1 Self Assembly

Self-Assembly (SA) is ubiquitous on scales from molecules to galaxies. Two excellent reviews that define the extent to which self-assembly is pervasive in the world around us have been published by G.M.Whitesides[10,11]. An exact definition however is difficult to find as the term self-assembly has been loosely applied across many disciplines. One of the most reasonable definitions might be that of Whitesides:

> *"the autonomous organization into patterns or structures without human intervention"*[10].

Whitesides goes further to say that:

> *"Self-assembly reflects information coded (as shape, surface properties, charge, polarizability, magnetic dipole, mass, etc.) in individual components; these characteristics determine the interactions among them. The design of components that organize themselves into desired patterns and functions is the key to applications of self-assembly."*[10]

This is where this thesis is most concerned. Looking at existing systems for self assembly as will be reviewed below, the reader will come to see that the field of SA is currently information limited, and the resulting structures that are being formed, are simple, error prone, and low in complexity.

Figure 2-1 (from Whitesides[11]) serves to highlight some of the gaps in current thinking about self assembling processes. The ordered macromolecules of D are ordered after minimizing their energy in folding and annealing. Order, however, can also come from state, or program, in the individual components. As chaperone molecules also assist in the folding of macromolecules so may state machines assist in creating complexity and non-equilibrium structures in self assembly processes.

---

[10] Whitesides, G. M. and B. Grzybowski (2002). "Self-assembly at all scales." Science **295**(5564): 2418-2421.

[11] Whitesides, G. M. and M. Boncheva (2002). "Beyond molecules: Self-assembly of mesoscopic and macroscopic components." Proceedings of the National Academy of Sciences of the United States of America **99**(8): 4769-4774.

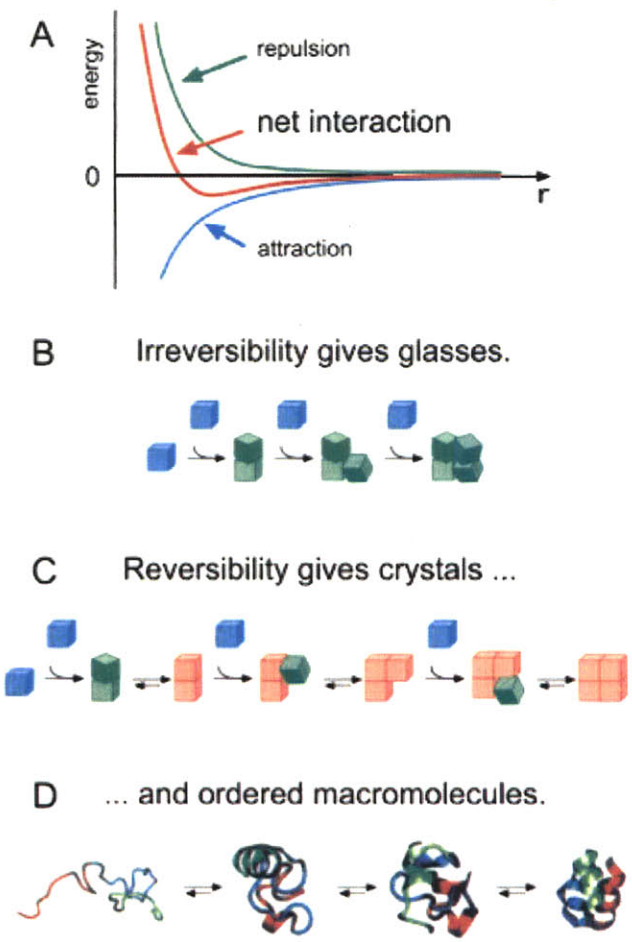


Figure 2-1 From Whitesides[11]. Forces, bonding types, and resulting structures, in self assembling systems.

Broad classification of systems that self assemble have been attempted, the most useful of which once again comes from Whitesides[10]. It is outlined thus:

Static self-assembly: Systems that are at global or local equilibrium and do not dissipate energy. Examples are molecular crystals, and globular folded proteins. Static self assembly may require an energy input such as stirring, but once formed, the system is stable.

Dynamic self-assembly: The patterns or structures by dynamic SA occur if the system is dissipating energy. Examples include the patterns formed by competition between reaction and diffusion in oscillating chemical reactions.

Templated self-assembly. Interactions between the components and features in their assembly environment determine the final formed structures. Crystallization on surfaces that determine the morphology (epitaxy) is one example as is crystallization of colloids in 3D optical fields.

Biological self-assembly. This is a superset of previous categories. In biological systems the defining characteristic is the variety and complexity of the functions produced.

By the end of this work I shall demonstrate that a fifth category is useful, that of logic-regulated self assembly – self assembling systems where the assembly proceeds according to logic embedded in the sub-components.

Interest in self-assembling systems outside of pure chemistry has only occurred quite recently. This was stirred largely in response to a few select papers on micro and meso-scale assembly[12],[13]. Perhaps the most important are those of Bowden et.al. which describe assemblies in systems comprising hexagons patterned with hydrophobic and hydrophilic surfaces assembling at a water / perfluorodecalin (PFD) interface. Assembly by capillary forces on parts of this type at liquid / liquid interfaces has become something of a default system for the study of self assembly. The origins are most likely traced to Hosokawa[12] and have been used by many researchers including Rothemund[14].

The best review of the physics of these systems can be found in Bowden's PhD thesis[15] and related publications[16]. A basic overview of the capillary driving forces is presented in Figure 2-2 (a).

Interestingly this work intones the programming of the final assembly by defining the surface properties of each tile in the assembly. This work highlights the tempting attraction of self assembly, but also its great limitations as it is currently envisioned. Figure 2-2 (b) runs through the combinatorics of possible pre-programmed patterns of hydrophobic and hydrophylic faces of the hexagons and Figure 2-3 demonstrates the resulting structures that self-assemble. It is immediately apparent that these systems always exhibit a high number of assembly errors and that all of the constructions can only be crystalline in their regularity.

---

[12] Hosokawa, K., I. Shimoyama, et al. (1996). "Two-dimensional micro-self-assembly using the surface tension of water." Sensors and Actuators a-Physical **57**(2): 117-125.

[13] Bowden, N., A. Terfort, et al. (1997). "Self-assembly of mesoscale objects into ordered two-dimensional arrays." Science **276**(5310): 233-235.

[14] Rothemund, P. W. K. (2000). "Using lateral capillary forces to compute by self-assembly." Proceedings of the National Academy of Sciences of the United States of America **97**(3): 984-989.

[15] N.B.Bowden, "Order and Disorder: Mesoscale Self Assembly and Waves", Ph.D. in Department of Chemistry. Cambridge, Massachusetts: Harvard University, 1999.

[16] Bowden, N. B., M. Weck, et al. (2001). "Molecule-mimetic chemistry and mesoscale self-assembly." Accounts of Chemical Research **34**(3): 231-238.

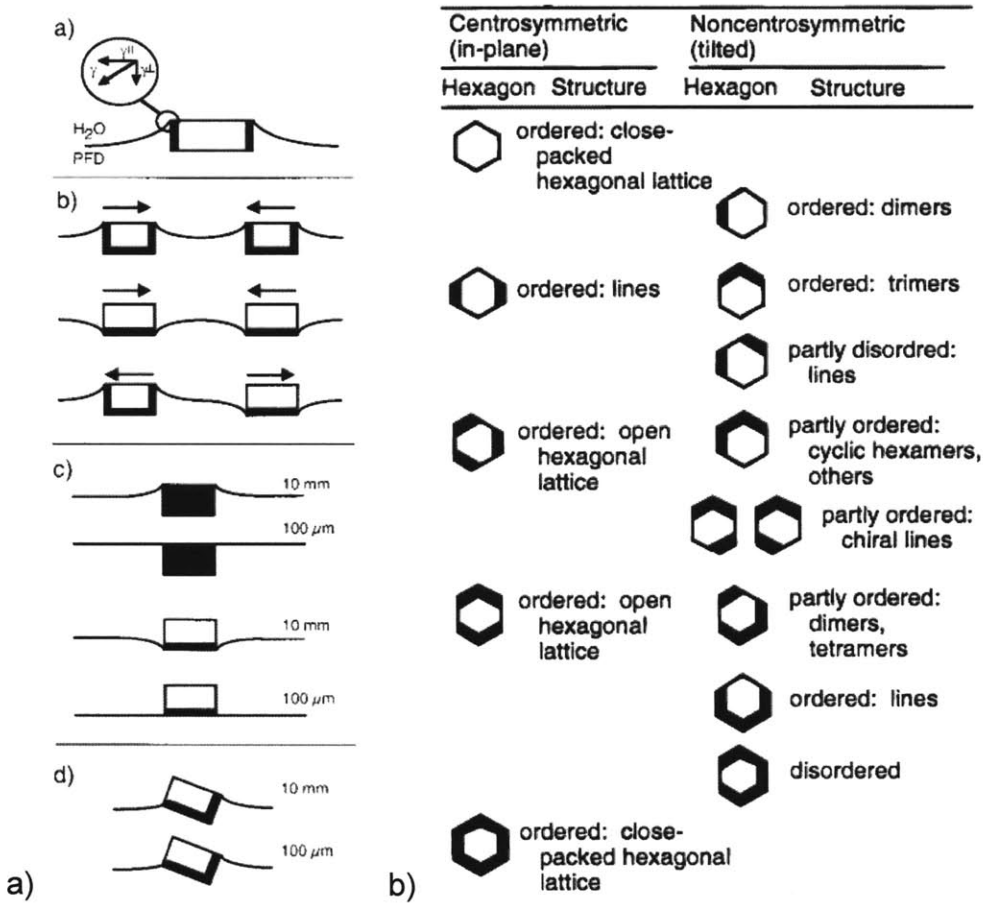


Figure 2-2 (a) Nature of attractive and repulsive forces generated by meniscus surfaces at the interface. (b) Table of possible edge patterns in hydrophobic / phyllic patterned PDMS assembling at the interface of PFD and H2O.

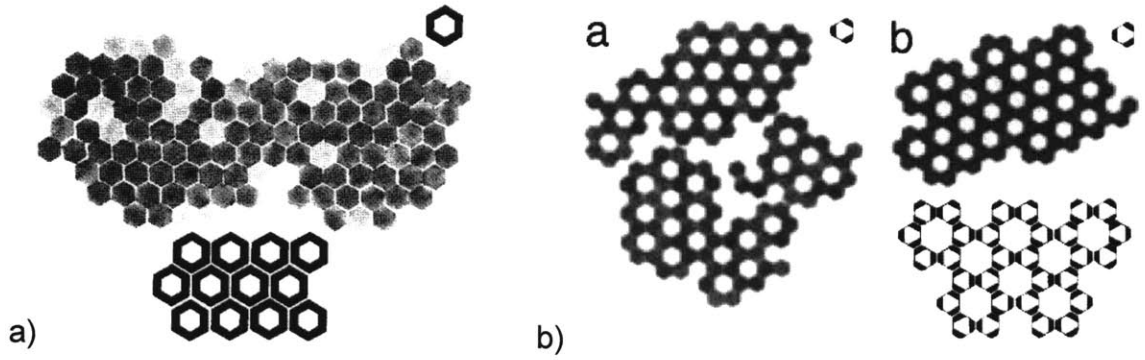


Figure 2-3 Two examples of assemblies of parts as patterned in Figure 2-2. These patterns demonstrate both the crystalline and error prone nature of this type of assembly scheme as well as the limitations in programming any complex structure into the system. From Bowden[15].

Clark[17] and others extended this work to 3D arrays of tiny hexagons assembled as neutrally buoyant components in an agitated 3D environment. This work begins to look very interesting as the assemblies are now of thousands of components and the resulting structures are certainly three dimensional. Careful examination of Figure 2-4 will reveal a number of the realities of this type of SA. The growing arrays do not receive a termination signal (counting in any direction) and so the final structure, although having great regularity, does not have any programmed global shape. Defects, both inclusions and vacancies, can be seen throughout the structures. Annealing time, sub-component manufacturing tolerances, and the proximity of the agitation energy to the correct binding energy determines the defect rate in these systems.

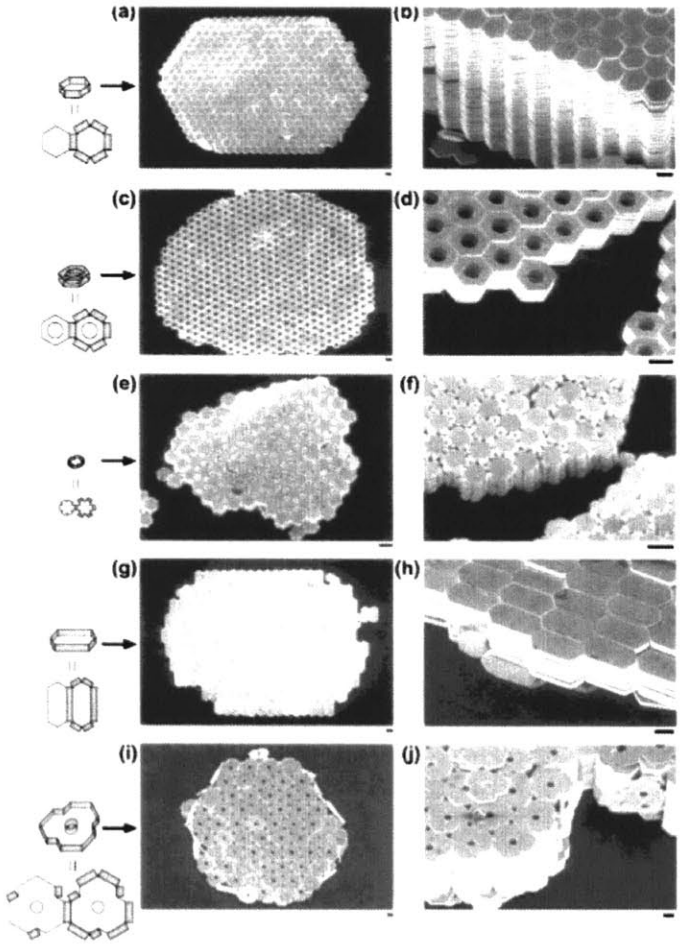


Figure 2-4. Micron scale 3D assembly. Clark[17]

[17] Clark, T. D., J. Tien, et al. (2001). "Self-assembly of 10-mu m-sized objects into ordered three-dimensional arrays." Journal of the American Chemical Society **123**(31): 7677-7682.

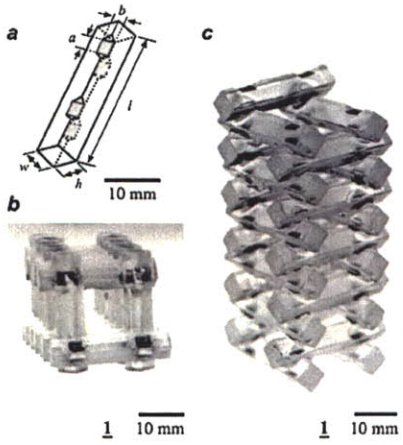


Figure 2-5. Oliver[18] et.al. 3D SA by capillary bonding of low Tm alloy to copper tape on polyurethane rods.

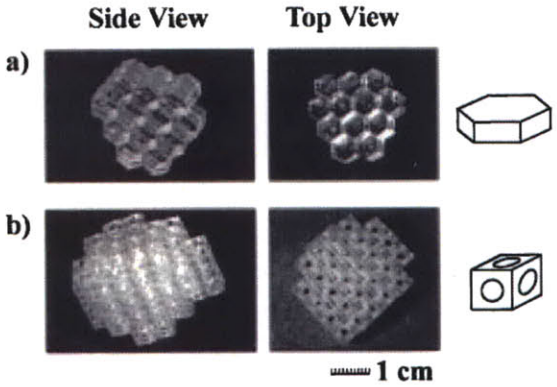


Figure 2-6. Tien[19] et.al. 3D Crystals by SA of polyurethane polyhedra.

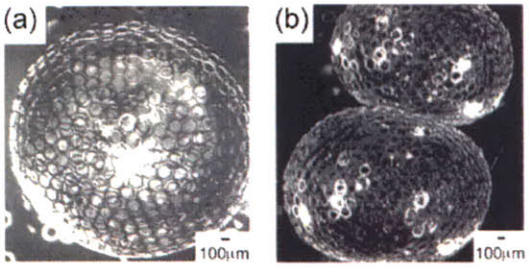


Figure 2-7. 3D structures by the templated assembly of hexagons at the interface between a Perfluorodecalin droplet and water.

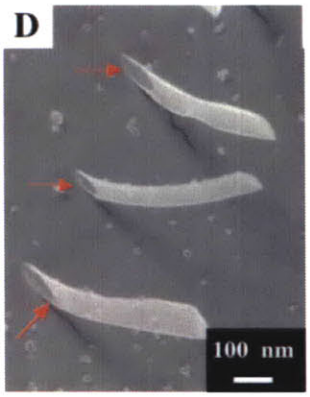


Figure 2-8. 3D nano-tubes by self assembly of surfactant-like peptides. Vauthey[20] et.al. 2002.

The next pages of figures of prior art illustrate the other memes and ideas that have been presented in the self-assembly literature. These processes represent various efforts to extend self assembly into 3D (Figure 2-5,Figure 2-6,Figure 2-7, Figure 2-8); templating (Figure 2-9 & Figure 2-10); functional (

[18] Oliver, S.R. et.al. Three dimensional Self Assembly of Complex, Millimeter-Scale Structures through Capillary Bonding. J.Am.Chem.Soc. 2001,123,8119-8120.

[19] Tien, J., T. L. Breen, et al. (1998). "Crystallization of millimeter-scale objects with use of capillary forces." Journal of the American Chemical Society **120**(48): 12670-12671.

[20] Vauthey, S., Santoso, S., Gong, H., Watson, N., Zhang, S. "Molecular self-assembly of surfactant-like peptides to form nanotubes and nanovesicles." PNAS 99, no.5., April 16, 2002. pp.5355-5360.

Figure 2-13,Figure 2-14), and folding (Figure 2-11,Figure 2-12) memes. All of them rely purely on energy minimization to achieve final structure. I will refer the reader to the detail in the original papers rather than describing them further here as the important thing to be gained is the numerous systems in which self assembly can be performed and studied, and the numerous techniques and resulting structures being explored.

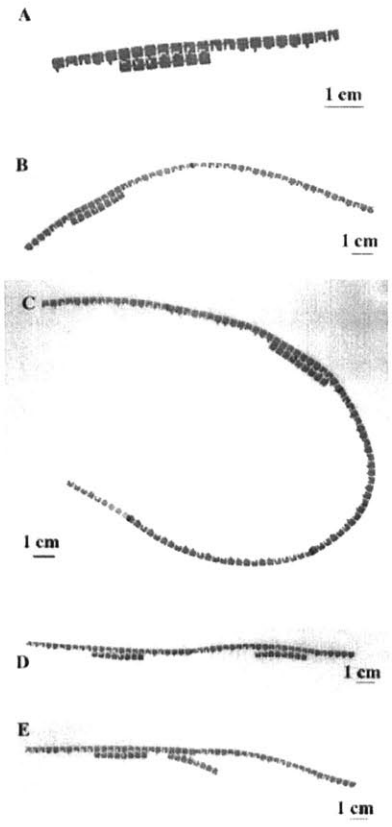


Figure 2-9 Recognition by templating of mm-scale nucleic acid analogues. Weck[21] et.al.

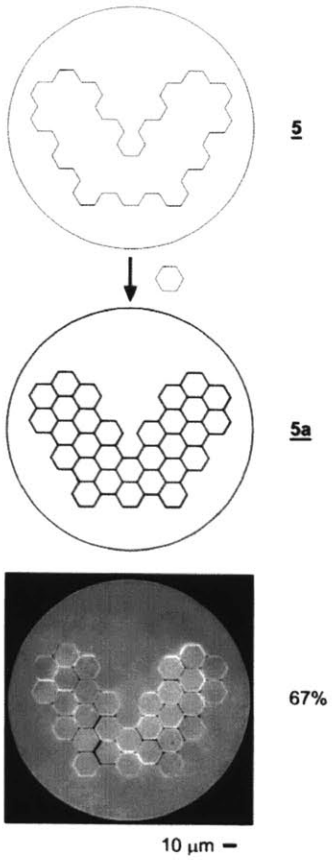


Figure 2-10 Templated assembly of micron scale silicon hexagons. Clark[22] et.al. 2002.

[21] Weck, M., I. S. Choi, et al. (2000). "Assembly of mesoscopic analogues of nucleic acids." Journal of the American Chemical Society **122**(14): 3546-3547.

[22] Clark, T. D., R. Ferrigno, et al. (2002). "Template-directed self-assembly of 10-mu m-sized hexagonal plates." Journal of the American Chemical Society **124**(19): 5419-5426.

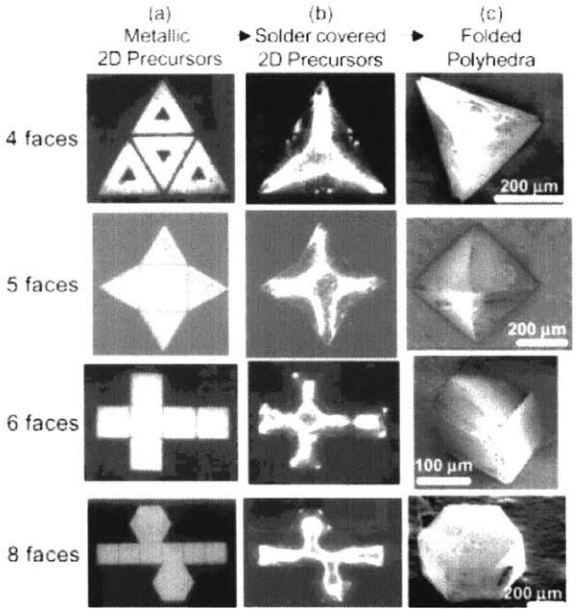


Figure 2-11 Gracias[23] 2002: Folding of polyhedra by surface tension around solder balls.

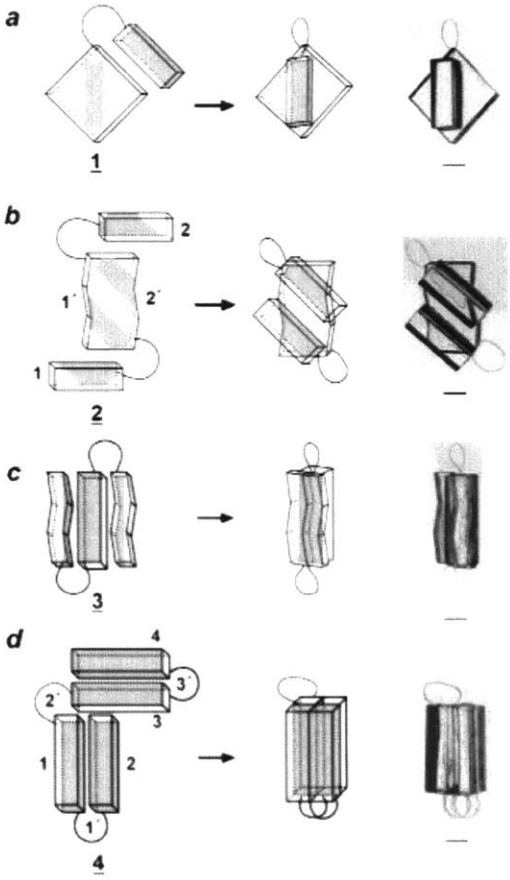


Figure 2-12 Clark[24] 2002: Tethered mm parts of PDMS mimicking molecular scale folding.

[23] Gracias, D. H., V. Kavthekar, et al. (2002). "Fabrication of micrometer-scale, patterned polyhedra by self-assembly." Advanced Materials **14**(3): 235-+.

[24] Clark, T. D., M. Boncheva, et al. (2002). "Design of three-dimensional, millimeter-scale models for molecular folding." Journal of the American Chemical Society **124**(1): 18-19.

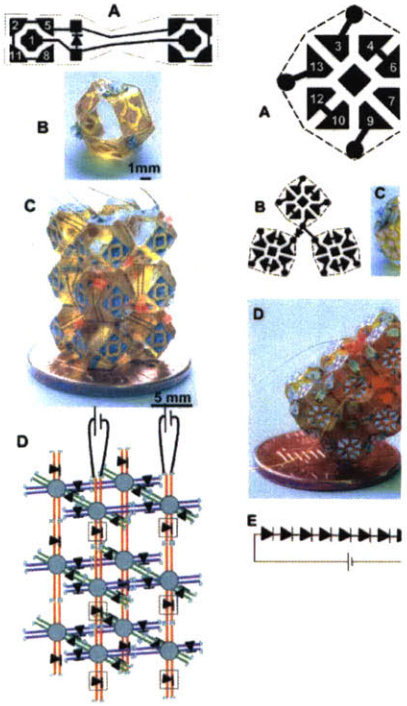


Figure 2-13 Gracias[25] 2000: Electrical networks and LEDs on SA truncated polyhedra.

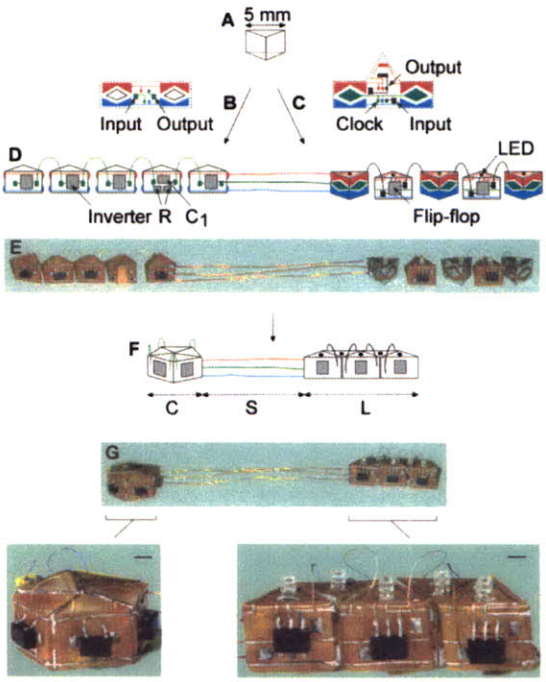


Figure 2-14 Boncheva[26] 2002: functional asymmetric electronic device.

Other ideas in self assembly worth noting are Brittain's[27] origami and tetrahedral, Terfort's [28] functioning SA LED's in solution, Breen's[29] 3D regular open crystals (mm scale) welded with Bi Alloys, Choi's[30] model for protein folding with tethered hexagons and Ng's [31] self assembling gears

---

[25] Gracias, D. H., J. Tien, et al. (2000). "Forming electrical networks in three dimensions by self-assembly." Science **289**(5482): 1170-1172.

[26] Boncheva, M., D. H. Gracias, et al. (2002). " Biomimetic self-assembly of a functional asymmetrical electronic device." Proceedings of the National Academy of Sciences of the United States of America **99**(8): 4937-4940.

[27] Brittain, S. T., O. J. A. Schueller, et al. (2001). " Microorigami: Fabrication of small, three-dimensional, metallic structures." Journal of Physical Chemistry B **105**(2): 347-350.

[28] Terfort, A. and G. M. Whitesides (1998). "Self-assembly of an operating electrical circuit based on shape complementarity and the hydrophobic effect." Advanced Materials **10**(6): 470-+.

[29] Breen, T. L., J. Tien, et al. (1999). "Design and self-assembly of open, regular, 3D mesostructures." Science **284**(5416): 948-951.

[30] Choi, I. S., M. Weck, et al. (2000). " Mesoscale folding: A physical realization of an abstract, 2D lattice model for molecular folding." Journal of the American Chemical Society **122**(48): 11997-11998.

[31] Ng, J. M. K., M. J. Fuerstman, et al. (2003). "Self-assembly of gears at a fluid/air interface." Journal of the American Chemical Society **125**(26): 7948-7958.

including fluid couplings. Ismagilov[32] demonstrated autonomous movement of parts by Pt electrodes bubbling hydrogen rather than agitating the system. Little work has been done in the modeling of the physics of these systems rigorously, though there are a few examples such as Grzybowski[33] modelling capillary interactions with a package called surface evolver.

Mao[34] demonstrated reconfiguring SA with changing environment – an interesting meme where state could be embodied in conformational parts where the conformation is switched by changing environmental conditions such as pH.

2.2 Algorithmic assembly and tile based computation

Early attempts at the theoretical, mathematical aspects and limits of self assembly have been attempted by Rothemund[35], Winfree [36],[37],[38], Adleman [39], and others. Figure 2-15 & Figure 2-17 represent the very preliminary work that exists in linking computation and computational processes to self assembly. The work derives from their interest in performing computation in tilings of double cross-over DNA molecules. Rothemund's experimental work in acrylic parts assembling in water is the most computationally intensive of any to date in self assembly, and elegantly demonstrates the limitations of edge-patterned tiling memes in self assembly. Inclusions, vacancies, and binding errors are prevalent in the assemblies, and it can be seen, especially in the Sierpinski triangle example that errors in the assembly process are critical in allowing an assembly to configure to the desired outcome. The work of these researcher's and others is obviously highly informed by the literature in cellular automata, such as that of Wolfram[42], Winograd, and others. What is not necessarily obvious in these pieces is that for purely pre-patterned or 'stateless' sub components, there will need to be roughly N/10 different tile types for a specific structure of N parts where the structure is not symmetrical or crystalline[40]. As the number of tile types increases, the search time for them in a

---

[32] Ismagilov, R. F., A. Schwartz, et al. (2002). "Autonomous movement and self-assembly." Angewandte Chemie-International Edition **41**(4): 652-+.

[33] Grzybowski, B. A., N. Bowden, et al. (2001). "Modeling of menisci and capillary forces from the millimeter to the micrometer size range." Journal of Physical Chemistry B **105**(2): 404-412.

[34] Mao, C. D., V. R. Thalladi, et al. (2002). "Dissections: Self-assembled aggregates that spontaneously reconfigure their structures when their environment changes." Journal of the American Chemical Society **124**(49): 14508-14509.

[35] Rothemund, P.W.K. (2000) "Using lateral capillary forces to compute by self assembly". PNAS 97(3):984-989.

[36] Winfree, E., Liu, F.R., et.al. (1998) Design and self-assembly of two-dimensional DNA crystals.

[37] Winfree, E. (2000) Algorithmic self-assembly of DNA: Theoretical motivations and 2D assembly experiments." Journal of Biomolecular Structure and Dynamics: 263-270.

[38] Winfree, E., Bekbolatov, R., "Proofreading tile sets: Error correction for algorithmic self-assembly." DNA Computing 2943: 126-144.

[39] Adleman, L. (1999). "Towards a Mathematical Theory of Self-Assembly." Technical Report 00-722, Department of Computer Science, University of Southern California, (2000).

[40] Private communication with Erik Winfree.

random bin also increases and the kinetics is slowed. As each tile type must have a unique edge encoding the problem is exacerbated in having to find larger and larger numbers of non-complimentary codings (eg. nucleotide sequence) which will limit the size of the components. To prevent errors with larger numbers of tiles the thermodynamics of the reaction also need to be slowed and will similarly effect the kinetics of these types of assembly schemes negatively.[40]

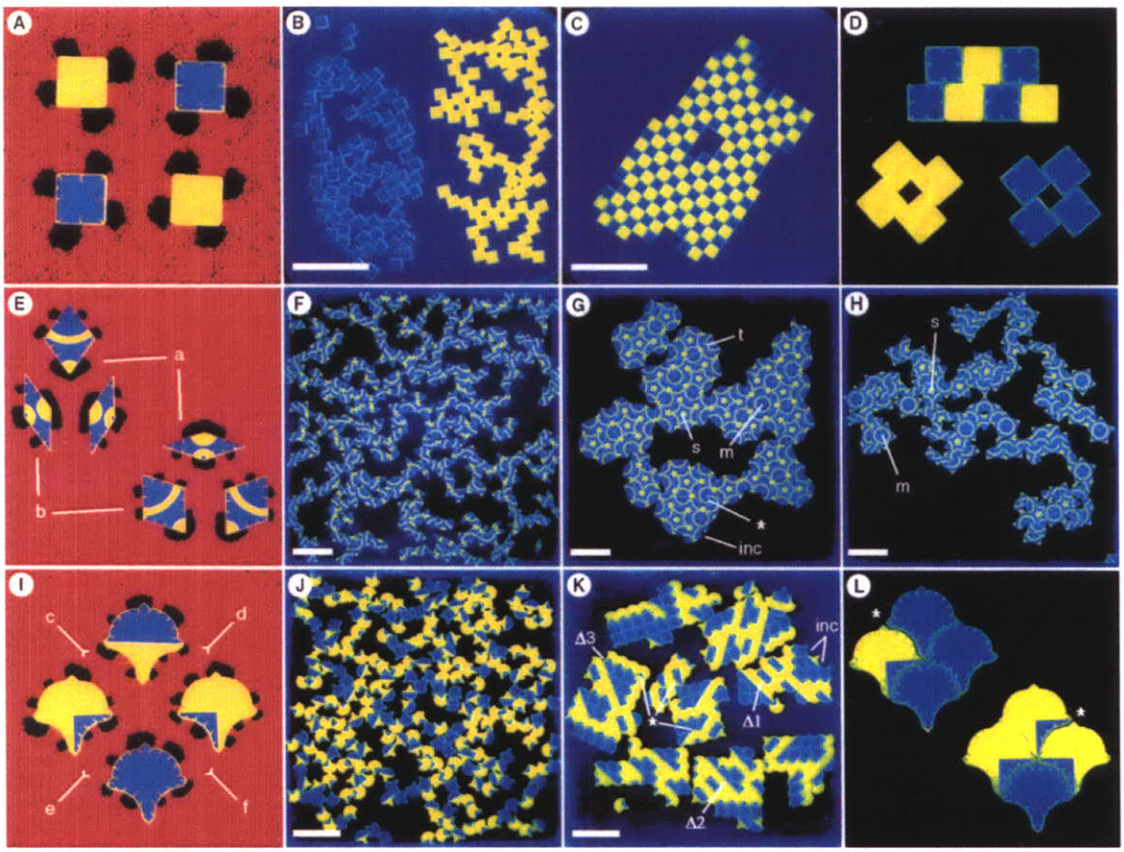


Figure 2-15 Self assembly of higher complexity structures and computational assemblies. Acrylic parts with hydrophobic / hydrophyllic surface treatments at an air/water interface. Note particularly the Penrose tiling in the second row and the Sierpinski triangle in the third. (Rothemund[14]).

It is important and pertinent at this point to differentiate between the programming of cellular automata and simulations, and the implementation in physical systems. In cellular automata the information is assumed to pass between parts in a 'neighbourhood' around each cell which may or may not actually have a physical connection in a real Euclidean space. Each cell uses a program or look-up table to execute the selection of the next part. Cellular Automatons (CA's) are also typically synchronous, clocked to cycles, and exist in a connected, ideal, Euclidean grid or tiling. In a real self-

assembly, the addition of components is asynchronous, and occurs stochastically in time and space. Information can only be passed through physically connected parts, or through an onboard communication channel. In designing the simplest possible parts this leads us to tilings and geometries where the information is passed by direct edge contact. Information can be routed through neighbouring cells, but at the expense of adding complexity and 'memory' requirements to each of those cells. For example, the universal Turing machine (UTM) CA of Wolfram shown in Figure 2-17 can be implemented in a self-assembling tiling, however 3 extra tile types are required at the corners of truncated squares within the CA grid to act as the information transfer by edge contact. This means it is possible to implement a UTM in a SA tile system, however 11 tile types[41] are required and must add sequentially, row by row, proceeding down the figure as seen in the illustration. This is a very inefficient computer and to implement it as a method for creating a specific structure would come at great expense in terms of number of tiles required and resolution.

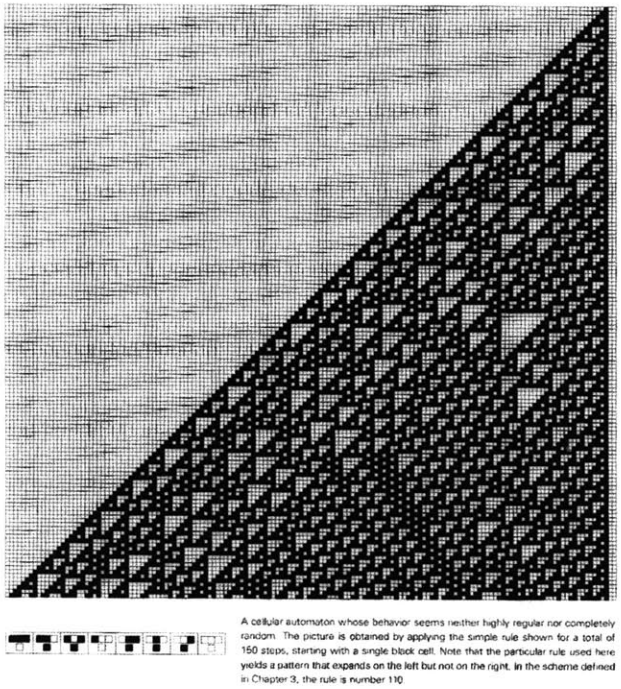


Figure 2-16 Computational assemblies in matrix based Cellular Automata – Wolfram[42], and many others.

[41] Author's own design.
[42] Wolfram. "A New Kind of Science". Wolfram Media Inc., 2002.

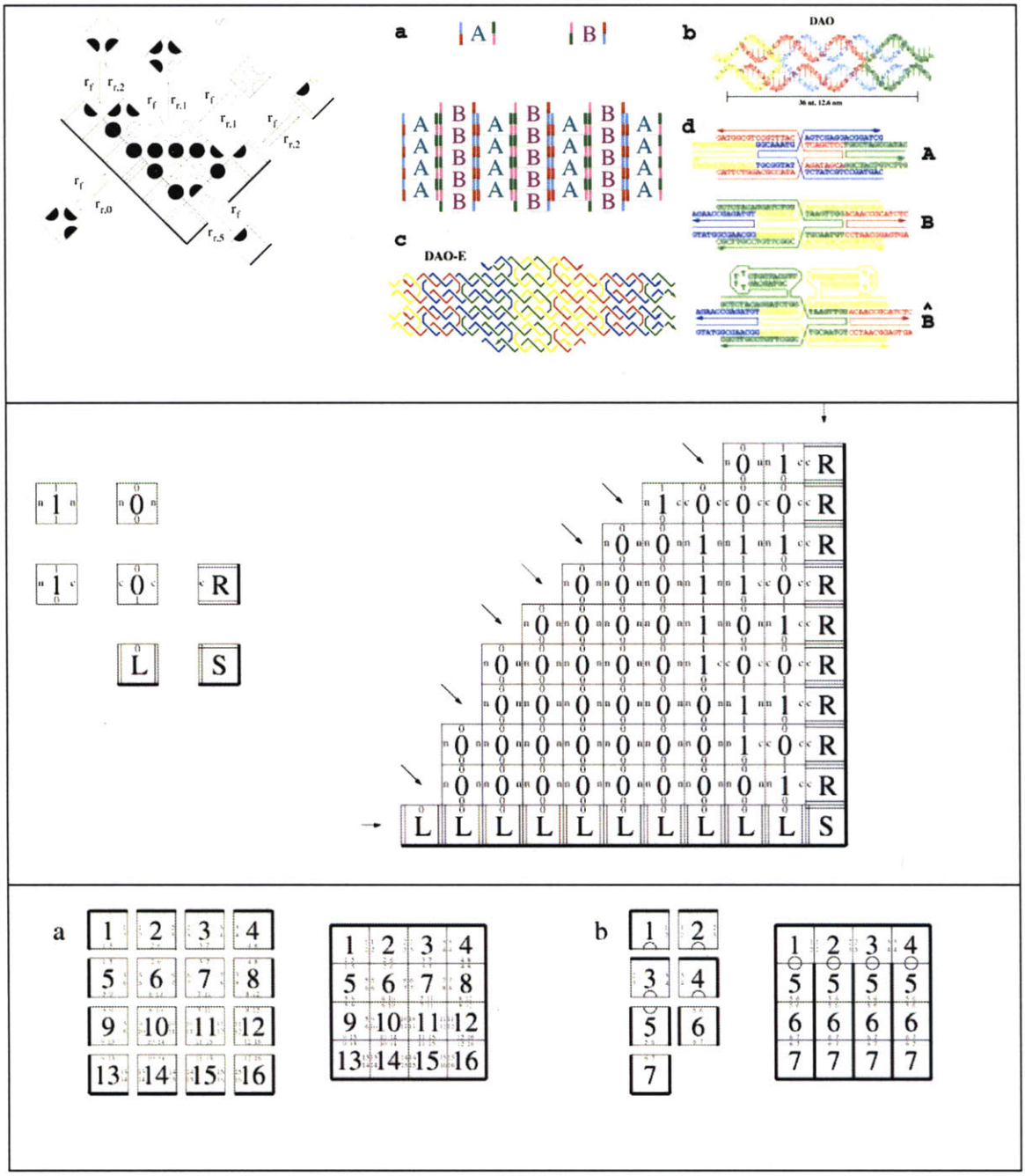


Figure 2-17 Winfree et al. Top left: Mathematical model for self assembly using edge strength and multiple tiles to differentiate structure. Top right: DNA computation by self assembly of double cross over molecules. Centre: Implementing a self-assembling bit counter in a minimal tile set. Bottom: Program Size complexity of an NxN square assembly[43].

[43] Rothemund, P.W.K., Winfree, E., "The Program-Size Complexity of Self-Assembled Squares." STOC'00. Portland, Oregon, USA.

2.3 *Small Programs in Large Numbers*

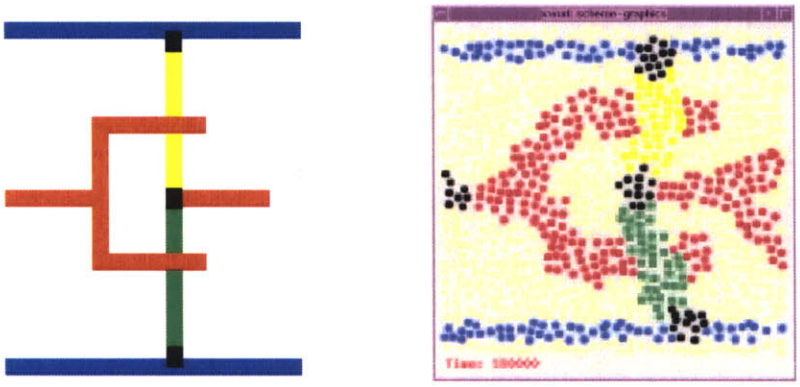


Figure 2-18 Generation of Interconnect topologies by amorphous computing. Coore[44]

There are three other pertinent pieces of work worth mentioning in the context of this thesis for the way they inform the work, and for the potential impact this work will have on them. The amorphous computation group of the MIT AI lab has produced a number of theses addressing the nature of computation in structure formation. The most notable of these is the work of Daniel Coore[44] in generating interconnect topologies on an amorphous computer, and the work of Rhadika Nagpal[45] in generating an origami folding language also implemented in an amorphous computer. Both of these pieces of work were implemented in simulation only. Being simulated systems, the assumptions made about computation about each node or cell in their network influences greatly the relevance of this work to self assembly. In both cases, each cell has computational processing capacities and memory storage capacities of the sort one would find in microprocessors. These are the fundamental units of this type of assembly. I am informed by their work, but the relevance could perhaps be best described in the following biological analogy. Amorphous computing assumes complete and computationally powerful cells, much like a biological cell, and generates complex structure from interactions and communications between these cells. The work I am doing on programmable self assembly is looking at more of a sub-cellular model where the basic units are nucleic and amino acids. The work in this thesis outlines the processes by which we might build the types of complex cells seen in the amorphous computing work from far simpler components, with just a handful of bits of state, the type of components one can imagine manufacturing in molar quantities in a synthetic polymer system, or in the billions of units by MEMS or lithographic processing.

---

[44] Coore, D. (1999). "Botanical computing : a developmental approach to generating interconnect topologies on an amorphous computer": 295. MIT PhD Thesis.

[45] Nagpal, R. MIT PhD Thesis. Dept. of Electrical Engineering and Computer Science. (2001). Programmable self-assembly : constructing global shape using biologically-inspired local interactions and origami mathematics: 105 leaves.

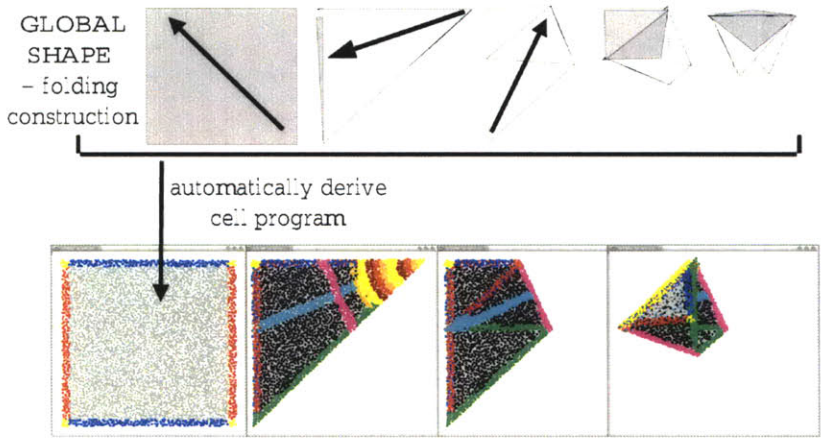


Figure 2-19 Structure generated by origami folds in an amorphous computer. Nagpal[45]

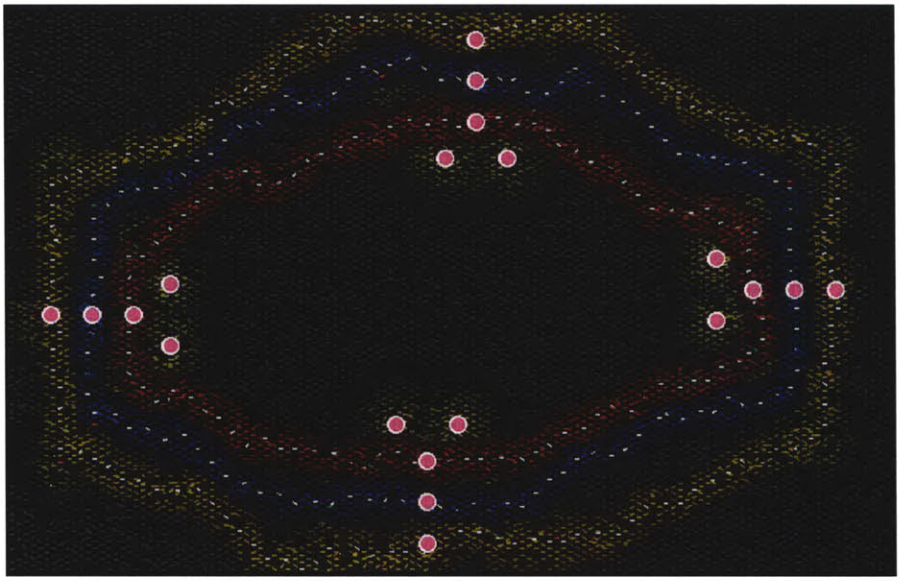


Figure 2-20 Computational Self Assembly. Butera[46]

Structural generation in biology is a multi-scale, tiered progression of assemblies. I expect that self assembly will be divided similarly; using this analogy the current work sits between the molecule type components of traditional self assembly work, and the multi-cellular type work in the amorphous computing community. This work is about the bio-molecules and allosteric molecules that sit between and connect the two. I am most fundamentally interested in the lowest and simplest individual units in this hierarchy, and how one will build reliable and computationally useful components from those, with which to build upon subsequent levels of complex behavior and structure. I should also mention the

work of Bill Butera[46], and his paintable computing project. This too is work that assumes extremely cheap, but powerful individual unit cells that interact amorphously, as particles in paint might be expected to, to produce useful computational behavior that is architecturally robust. Again, I seek to elicit the principles by which his sub-units themselves might be assembled robustly, and in quantity.

2.4 Self Replication and Artificial Life

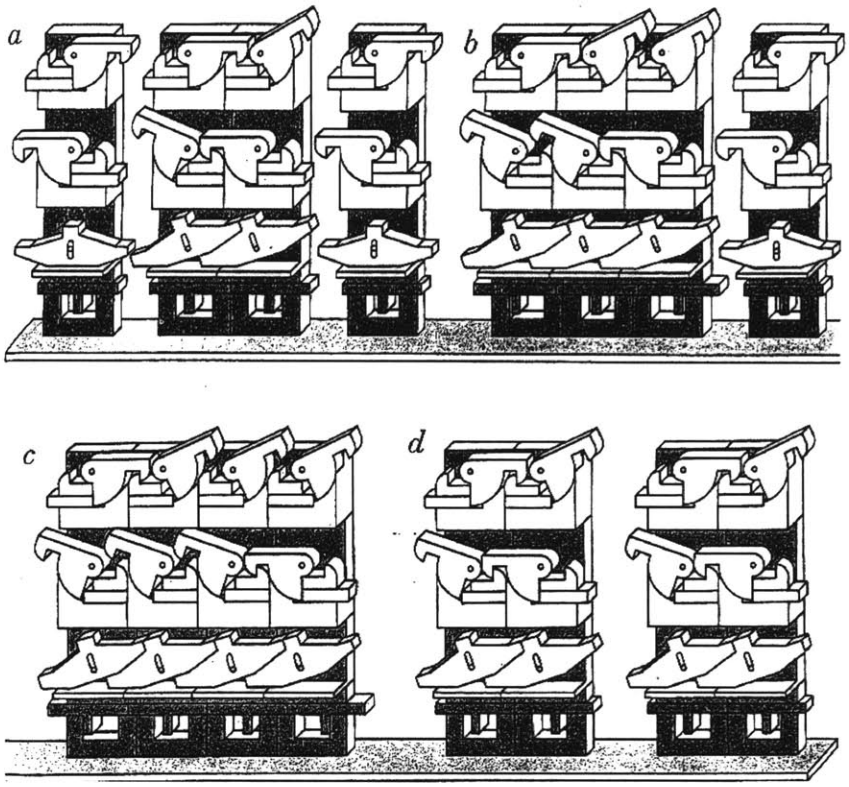


Figure 2-21 Mechanical Self Replicating machine design. Capable of copying di-mers from a pool of mono-mers. Penrose.[48,49]

The earliest and perhaps most famous treatment on self-replicating machines is that of Von-Neumann in his "Theory of Self Reproducing Automata"[47]. His was a rather complicated treatise on the concept that focused on very complicated automata similar to cellular automata.

Penrose and others outlined far simpler methods to achieve self-reproducing machines with varying degrees of generality[48],[49],[50]. I have included diagrams and photos of Penrose's work in Figure 2-21.

---

[46] Butera, W. J. Massachusetts Institute of Technology. Dept. of Architecture. Program in Media Arts and Sciences. (2002). "Programming a paintable computer": 176.
[47] Von Neumann, J. (1966). "Theory of Self-Reproducing Automata." University of Illinois Press.
[48] Penrose, L., S. and R. Penrose (1957). ""A self-reproducing analogue"." Nature **179**: 1183.
[49] Penrose, L. S. (June 1959). "Self Reproducing Machines." Scientific American **vol.200**: pp 105-114.

He was a biologist and mathematician interested in how DNA and RNA might replicate, before such things were understood. He built some wonderful, biologically inspired, models for replication from plywood. These parts were state machines, although he never described them or analysed them as such, possibly because he predated a lot of the state machine literature.

These are perhaps the most important pieces of prior art in describing my own work as they execute very interesting behavior such as replication, and a simplistic version of mutation, in very basic state machines. He implemented autonomously replicating di-mers, really a templating scheme, in 1 dimension, and showed the path to replicating arbitrary strings of information, though never in a system that replicated autonomously. That is one thing I am aiming at achieving in this work, and also in analyzing these systems in terms of their state machines to provide the self – assembly community with blueprints for systems that can achieve higher function such as replication, or logically limited epitaxial growth. Saitou[51], [52] did some interesting design of conformational switches for mechanical SA along similar lines.

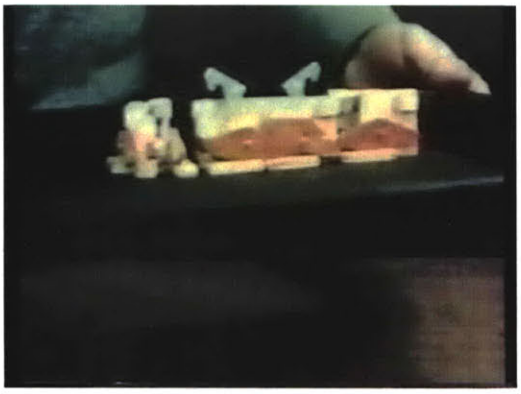
a.  b. 

Figure 2-22 (a) Kinematic self replicating plywood machine. Copies a di-mer by agitation in a 1D channel. (b) Kinematic self-replicating machine in 2D that replicates a bit-string defined by part colours (red/blue). Non-autonomous. Penrose. [48,49]

[50] Jacobson, H. (1958). "On models of reproduction." American Scientist **46**: 255-284.
[51] Saitou, K. (1996). Conformational switching in self-assembling mechnical systems : theory and application, Thesis Ph D --Massachusetts Institute of Technology Dept of Mechanical Engineering 1996: 174.
[52] Saitou, K. (1999). "Conformational switching in self-assembling mechanical systems." Ieee Transactions on Robotics and Automation **15**(3): 510-520.

2.5 Cellular and Reconfigurable Robotics

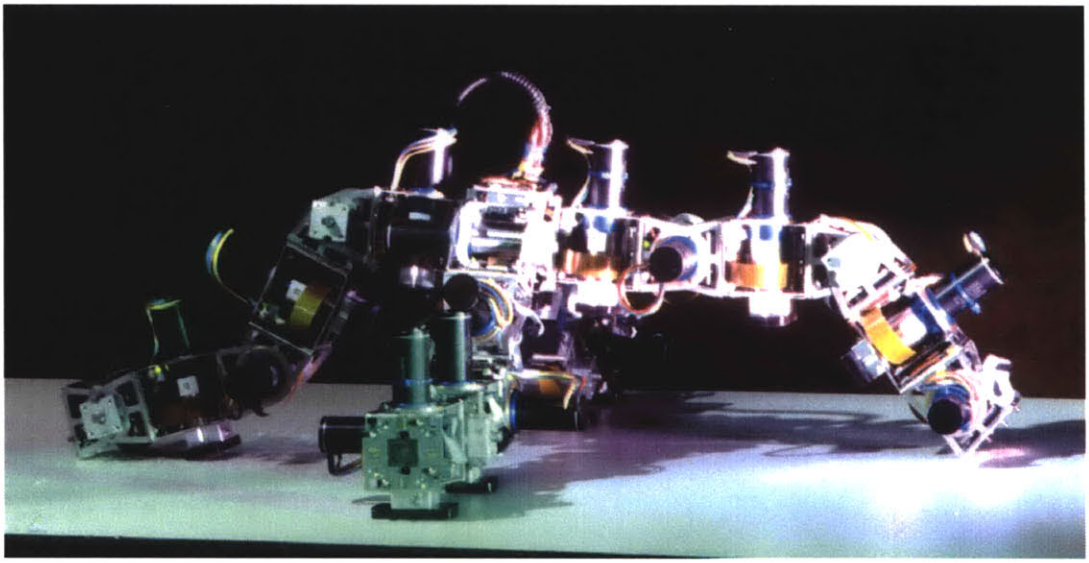


Figure 2-23 Yim's reconfigurable Polybot capable of self-locomotion by co-operation of sub-units.

Modular robotics is a nascent field of robotics where distributed control of reconfigurable entities make mobile robots or sensor networks, for example Hollar's[53] COTS dust.. The work of Yim et.al [54]. and his polybots are perhaps the best known The work of Daniela Rus[55] is also notable. The individual entities that comprise these robots are largely similar, and each individually contains logical elements, actuators, and locking and unlocking mechanisms to enable the re-configurability. At a small enough scale these start to look like the components of self assembly.

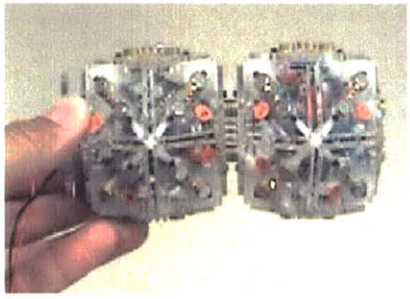


Figure 2-24 Rhombic dodecahedral sub-units of Yim's "Digital Clay"

One important point in the work to date has been the explicit focus on units that have actuation built in. Actuators to date are typically expensive and difficult to fabricate and at small sizes become

[53] Hollar, S. (Fall 2000). "COTS Dust." MSc Thesis, Mechanical Engineering, University of California, Berkeley,.

[54] Yim., M., Zhang, Y., Duff, D. (February 2002). "Modular Robots." IEEE Spectrum.

[55] Rus, D. and M. Vona (2001). "Crystalline robots: Self-reconfiguration with compressible unit modules." Autonomous Robots **10**(1): 107-124.

increasingly difficult. One aspect of the work in this PhD is to draw the link between modular robotics and programmable assembly and to introduce the concept of actuation as the collective behaviour of multiple units where there are no moving parts within each 'molecule' or base unit of the assembly. This concept will enable smaller scaling of modular robots.

Figure 2-25 Polybot Mk.3. unit cell. Each cell is highly complex in terms of degrees of freedom, computation and actuation.

3 Programming Assembly

3.1 A definition of programmed assembly – and a map.

More informed after Chapter 2, let me now define programmed assembly as a self assembling system that repeatably produces a known, designed, and desired output structure. This differs from pure self assembly in that self assembly is often the production of an unknown, or random structure, even if it has local order.

Figure 3-1 is a map of self assembly. The horizontal axis marks the number of states per tile in the system where the tile is the minimal sub-unit - for example the PDMS components of Whitesides, or the peptides of Zhang. The vertical axis is the number of different tile types in the system, for example the number of differently patterned edges on those PDMS parts, or the number of amino acids in a peptide.

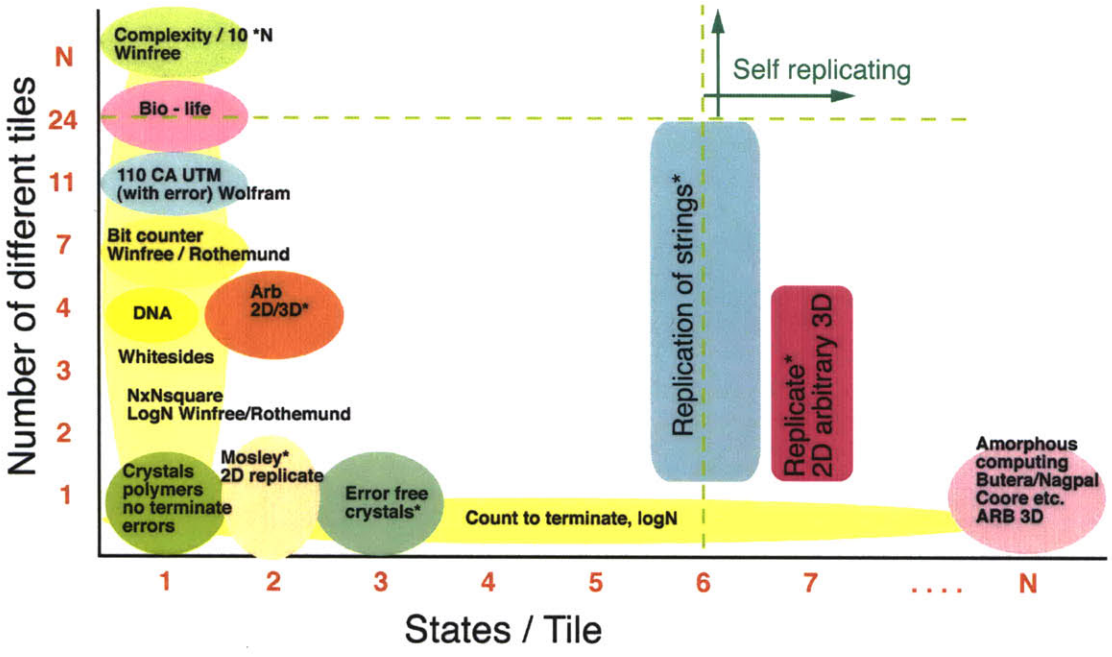


Figure 3-1 Map of states / tile, Number of tiles, and resulting structural capacity or complexity of assembly.

We can use this map for a broad survey of the field. The majority of the work surveyed in Chapter 2 can be seen at the very left, systems with a single pre-programmed state per tile. The simplest possible assemblies, those that look purely crystalline, are in the bottom left corner. Systems of this type are only seen to produce crystals (or regular arrays – the terms shall be used interchangeably) including polymers which can be considered a 1 dimensional crystal. Rothemund and Winfree's NxN squares also fit in this column where logN different tiles are required for the square. Their bit-counter

requires 7 different 1-state tiles to assemble. Note that all assemblies in the 1 state column are prone to errors, the number of which is a function of the 'thermodynamics' of the system where thermodynamics is broadly defined to include mechanically agitated systems such as those of Whitesides et.al. The author's 11-tile UTM by SA is also presented in this column, but will never compute accurately unless the assembly proceeds error free, which observation of chapter 2 assemblies shows is unlikely. At the far right I have defined the amorphous computing systems of Nagpal, Coore, and Butera to have N (close to infinite) number of states, and only one tile type. These systems should be able to build arbitrary 3D structure within the constraints of the individual tile construction.

One interesting point on the graph is a single tile, 2 state per tile system of Mosley[54]. It has the capacity to replicate 2D structures by a reversible plating process. The first 2D shape must be made, but subsequent copies may be replicated by a cyclic plating process. The quality of replicants will degrade with time.

Another interesting point on the graph is 24 tiles of 1 state per tile. If we consider biology to build from 4 nucleic and 20 amino acids, it could be said that this is a self replicating system of arbitrary, or at least very high, complexity.

In orange, blue, and magenta on this graph we have three systems that are introduced in this thesis. ORANGE: In chapter 4 it will be seen that 4 tile types can produce arbitrary (pixellation considered) 2D structure, and potentially also with 4 tile types arbitrary (voxelated) 3D structure.

BLUE: In chapter 5 it will be shown that arbitrary string sequences of shape differentiated 6-state state machines will self replicate. The combination of these two ideas gives us the MAGENTA mark: arbitrary self-replication and 3D structure formation. This was not implemented in this thesis (good luck to people who give it a shot!), but certainly suggests an interesting manufacturing technique.

GREEN: "error free crystals", 3 states per tile, 1 tile type, were implemented in chapter 6. This can be considered "Logic Limited Aggregation" or a first stab at error correction and prevention in fabrication using logic.

For clarification, self-replication within this work means copying of the original structure from the basic sub-units of the assembly environment.

For comparison to Figure 3-1 I have reproduced a table (Figure 3-2) from the thesis of Banks (1971 MIT). This table outlines the minimal systems capable of universal computation and universal construction within a 2D CA. I would suggest that something similar to Figure 3-1. will prove similar to this table, where there will be phase transitions to systems that do arbitrary 3D structure,

[54] Private communications: David Mosley, MIT Media Laboratory.

replication, and both. The dotted lines of Figure 3-1 are early estimates at the self-replicating phase transition.

Neighbours per cell

States per cell	5	9	85
2	Universal Computer Infinite initial configuration req'd (Banks)	Universal Computer Finite initial configuration. (Banks; also Gosper of MIT has shown Conway's Life cells are universal)	Universal Constructor (Codd) The cells were shown capable of simulating Codd's 8-state cells.
3	Universal Computer Finite Initial configuration is sufficient (Banks)		
4	Universal Constructor (Banks)		
8	Universal Constructor This was the first reduction of Von Neumann's cells. (Codd)		
13	Universal Computer These cells required only two rows of the array. (Smith)		
29	Universal Constructor (Von Neumann) This was the first cellular automaton shown to be universal		

**this table is drawn for infinite two-dimensional arrays of square cells. It gives results, discoverer and other information.*

Figure 3-2 Table of Universal Computation and Universal construction, implemented in cellular automata. Banks, 1971[55]

3.2 Information, communication, and memory

Any assembly contains information that determines the structure. This structure is determined by 'communication' between individual components, the information stored within, or 'memory' of those components, and the logical operations performed in the assembly by the components. This can be thought of as the logical operations which define various 'states' within and between the components.

[55] Banks, E. R. (1971). Massachusetts Institute of Technology. PhD thesis. Information processing and transmission in cellular automata: 101 leaves.

Communication can be as simple as the complimentarity of North and South poled magnetic faces of two parts or hydrophobic / hydrophilic interactions at faces, at one end of the spectrum, through to high bandwidth digital communications via optical, radio, capacitance, or induction schemes at the other end. The mode of communication defines much of the process of assembly by determining the 'neighbourhood' (borrowing from the cellular automaton literature). This is the 'communicaton radius' of the components, can parts talk to only those other components through which they share physical contact (as is the case with mechanical systems), or can they talk to distant components over a digital channel, or, as is the case with Butera's / Nagpal / amorphous computing systems do they communicate via gradients, either digitally, chemically, or otherwise?.

3.3 A hierarchy for assembly

Biology uses a hierarchy of assembly techniques to build complex matter, as does modern manufacturing which produces complex objects from assemblies of many sub-assemblies and sub-sub-assemblies etc. Loosely, in biology we could break it into the data assemblies of DNA, the molecular machinery of polymerase, chaperonin, and other functional enzymes comprised of amino acids that produce further functional sub-assemblies from other amino acids. Those sub-assemblies comprise the surprisingly functional cell. Cells produce our bio-materials that self assemble, hair, silk, keratin, and more. Multicellular assemblies make up our organs and assemblies of all these things comprise organisms. Those organisms are further able to build things external to themselves such as ant-hills and coral reefs (and art, architecture and computers!).

One would have to concede that self assembly or programmed self-assembly will similarly need multi-levels in its assembly hierarchy to achieve similar complexity. The work in this thesis suggests how we might approach some of the lowest levels in that hierarchy.

4 Arbitrary structure folded from linear strings.

4.1 Folding Introduction.

Biology has shown that encoding the information for structure in one dimension and folding that to three dimensions is a powerful and general meme – particularly if you wish to replicate. Can it be applied to non-biochemical components, and if so what are the constraints, resolution, and capacities? In particular, is there a notion of '3D completeness' – a set of components that provably can generate any 3D structure? For the purposes of this work I will only consider geometry in the 3D structure, not the placement of particular materials or functions within that geometry. Figure 4-1 is a cartoon of the folding of strings of amino acids into secondary, tertiary, and quartenary structures.

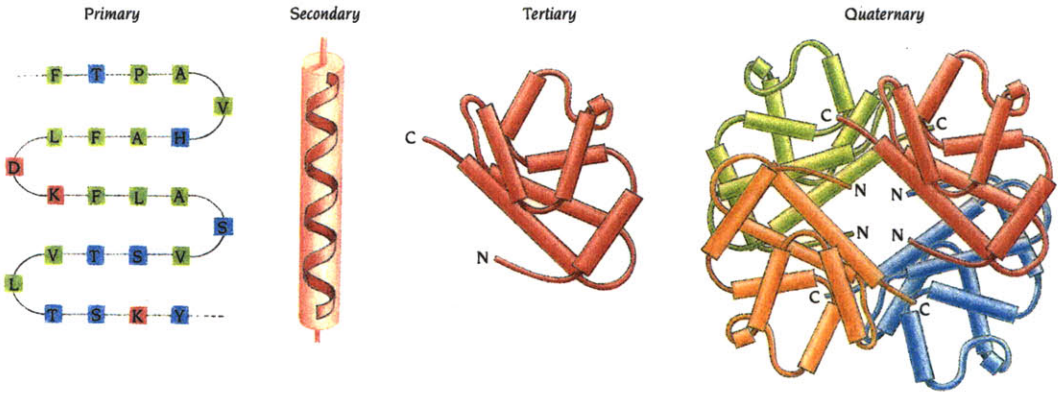


Figure 4-1 Biology's folding meme of linear strings of amino acids (poly-peptides) folding into functional structures.

Figure 4-2 is Rubik's snake, a famous children's toy. It is suggestive of the fact that we can think about constructing physical structure with strings of polyhedra. But is it possible to construct any geometry?

Figure 4-2 Rubik's snake uses a folding meme.

Figure 4-3 is what I will demonstrate is possible with mechanical or geometric analogs to this folding meme: the generation of a specific, and repeatable, final structure (a) from a string of component parts (b) that fold sequentially according to a specific rule set (c).

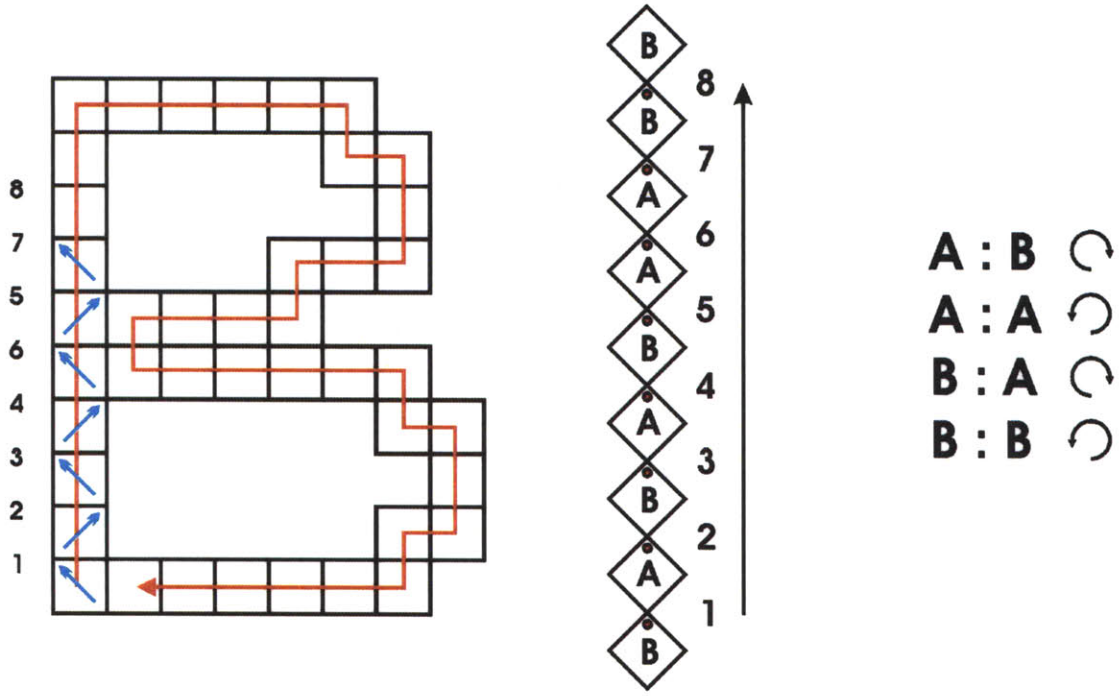


Figure 4-3 (from left t oright, a,b,c) A mechanical tile analog to the biological structures in figure 4.1

I will demonstrate in this chapter that arbitrary 2D or 3D structure can be generated from a linear string of polygons(2D) or polyhedra(3D). Arbitrary structure shall be defined as a space filling (plane tiling in 2D) connected assembly of pixels (2D) or voxels (3D).

In keeping with the ideal of a simplest possible set of components capable of developing a general set of structures this chapter will deal with the concept of building arbitrary 2D or 3D structures from a small tile set. As will be shown in chapters 5 & 6 a linear encoding of parts can be replicated with a small number of states within each component. This implies that a manufacturing system based on replicating in 1D and folding to 3D is possible. It will be pertinent to refer the reader to the work of Erik Demaine et.al. in "Hinged Dissection of Polyominoes and Polyforms"(CiteSeer[1]) and "Hinged Dissection of Polypolyhedra" (unpublished[2]) for some background in similar work within the analytical geometry community – it should be noted that this community does not generally consider manufacturability, in this case a non-intersecting sequence of folds, something important within this work. I will refer often to space filling throughout this thesis. Space-filling in this context means space filling within the boundaries of a defined object – that is the object is solid within it's skin or outer surface. This space will be filled with space-filling (in the traditional sense) polyhedra.

[1] Erik D. Demaine and Martin L. Demaine and David Eppstein and Erich Friedman, "Hinged Dissection of Polyominoes and Polyforms" http://citeseer.ist.psu.edu/demaine99hinged.html
[2] http://www.cs.tufts.edu/r/geometry/hinges/hinges.html

4.2 Constraints

1. Geometry: Simple, regular, space-filling structures, monotonic species (all units are same geometry), which implies equal edge lengths. Table 4-1 shows example polygonal primitives, their plane-filling tilings, and an example vertex-connected string for 2D folding. Table 4-2 shows the analogous polyhedral primitives, crystal stackings, and edge-connected strings for 3D. Of particular advantage are simpler geometries in terms of manufacturing such primitives in large numbers.

2. The folding path should also be realistic, ie, not self-intersecting, such that they can physically be realized.

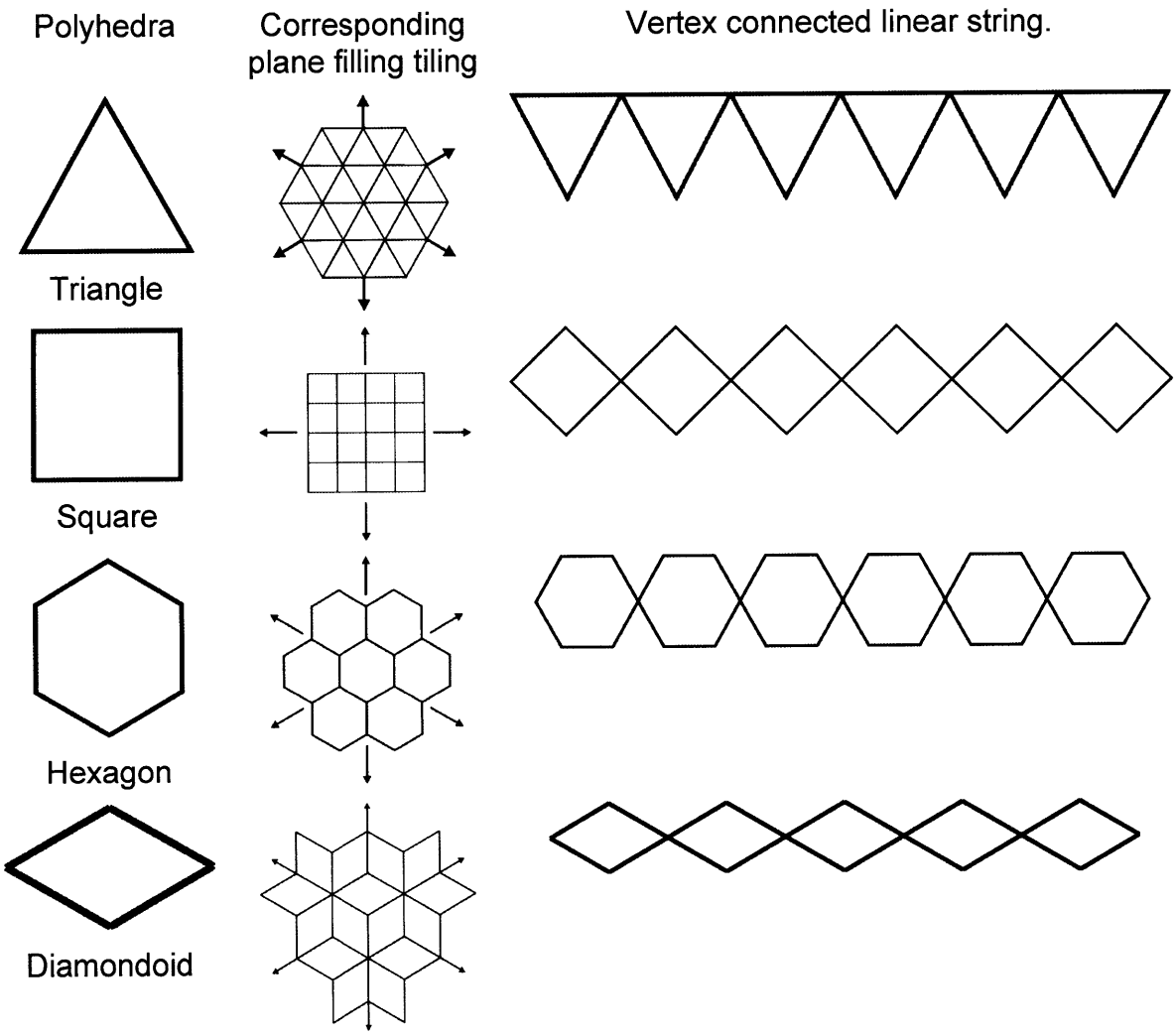


Table 4-1 Polygonal primitives, their plane tilings, and schematic of their vertice connected strings.

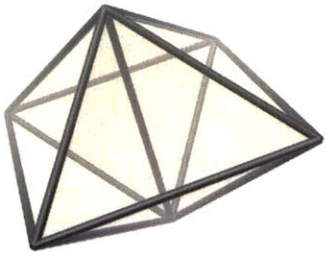
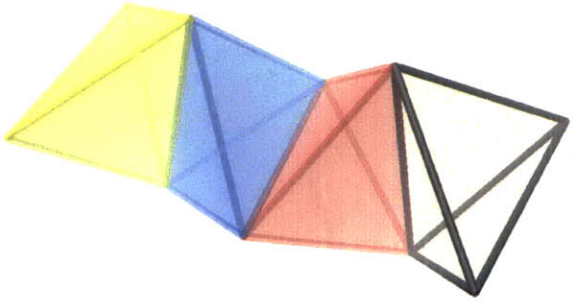
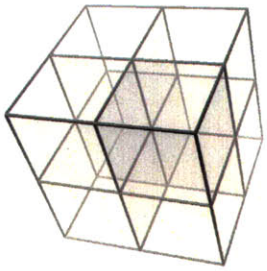
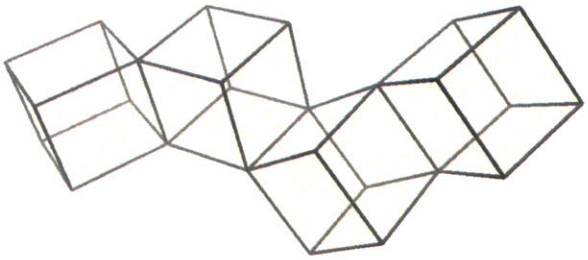
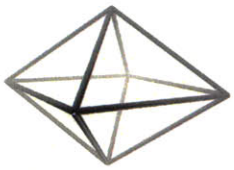
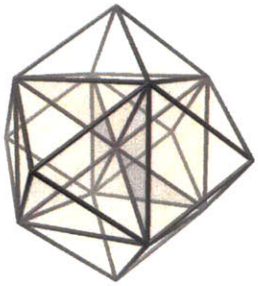
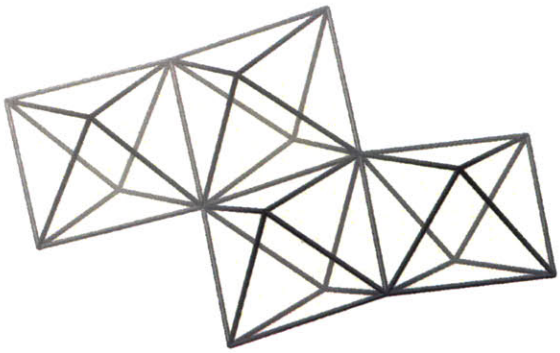
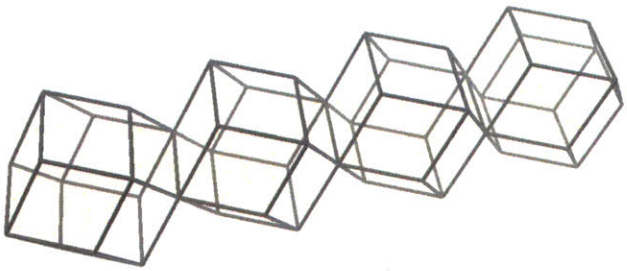
 *Right Tetrahedron*	 *4 Tetrahedron make one octahedron*	
 *Cube*	 *8 cubes in a 2x2x2 cube*	
 *Octahedron*	 *6 Octahedron in a rhombic dodecahedron*	
 *Rhombic Dodecahedron*	 *4 rhombic dodecahedra in closest packing formation*	

Table 4-2 Space filling polyhedra, their crystal stacking, and representation as strings of edge-connected polyhedra as might be useful in folding.

4.3 Problem Statement

Prove that a simple connected series of similar objects (polyhedra in 2D, polyhedron in 3D) connected at vertices (2D) and edges (3D) will fold sequentially – starting from one end and cascading down the string to the other end.

This is demonstrated in the string of vertex-connected squares in Figure 4-4. The vector on each square defines the folding sequence where each subsequent fold occurs at the head of the arrow. Each fold is a binary decision, left or right, about the vertex at the head of that arrow. The folding proceeds sequentially, 1..2..3..4..5..6.....n as illustrated.

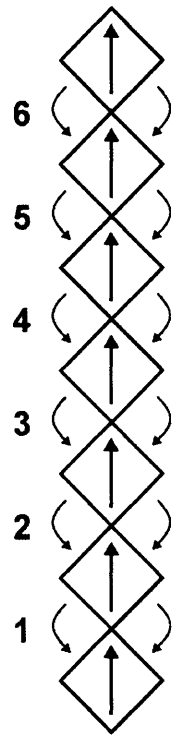


Figure 4-4 Sequential folds occur as a binary decision about the vertice at the head of each arrow from 1 to 6 to..n.

We'll begin with a construction that proves this is possible in 2D, extend that by analogy to 3D and then posit that the right-tetrahedron is 'minimal' in 3D and satisfies all constraints: arbitrary 3D geometry from a simple, linear, sequentially folding, non-intersecting, input string. We will also show an upper bound on the 'resolution' of such procedures, and the minimal tile set required for both 2D and 3D if one were to consider driving the folds using electrostatics, magnetics, surface tensions, actuators, or similar forces.

4.4 *Proof in 2 dimensions.*

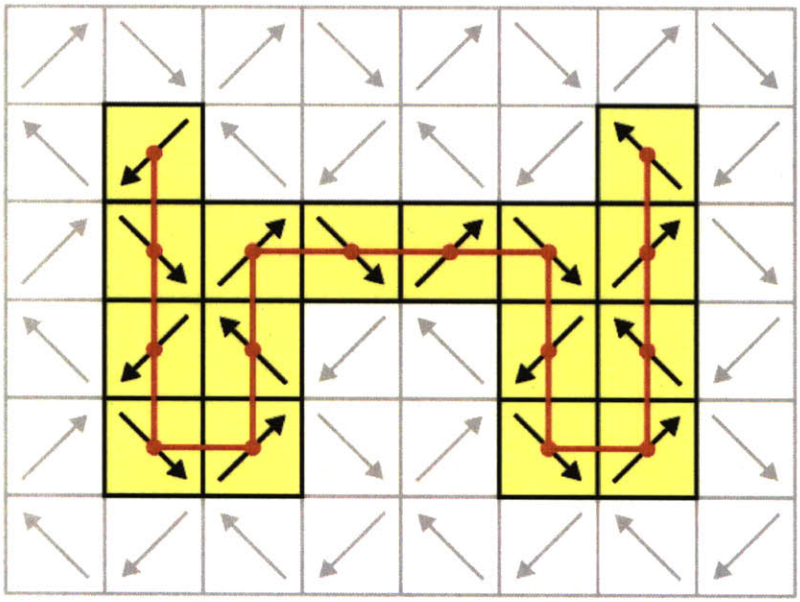


Figure 4-5 A non-branched spanning tree.

Figure 4-5 represents the trivial case of an unbranched spanning tree where the spanning tree is a graph that visits the centroid (node) of all squares (tiles) that are part of the 'object' which is superimposed on a set of square tiles that tile the plane.

A spanning tree is a connected, loop free, sub-graph containing all the original nodes. Such a sub-graph can always be constructed - for example through an exhaustive depth first search on the original graph.

Beginning at one endpoint of this un-branched spanning tree (in this case the top left hand corner), one can fold sequentially about the vertices with arrow heads to produce this (or any similarly un-branched, simple shape). In this example the folding sequence is L:R:L:L:L:R:R:L:R:R:L:L:L:R:L (where L = a left fold about the vertice at the arrow head of the tile vector and R = a right fold). Provided that the spanning tree is not branched, and that the starting vector is along the axis of the beginning of the tree, these trivial structures will always be foldable.

Figure 4-6 is a non-trivial case, that of a branched spanning tree. One can see that if we begin the folding at the snout, or nose, of the dog, at the first branch (roughly the eye of the dog) the next fold completes the ear above the head, but there is no return path to complete the body, legs and tail. Similarly, if one starts at the left leg, the folds proceed upward, to the right along the underbelly, and at the top of the hind leg do not allow a fold down the leg, but rather terminates. For any graph there can be numerous spanning trees, but for any with a branch (eg. the branch at the end of the snout), this folding technique will not suffice.

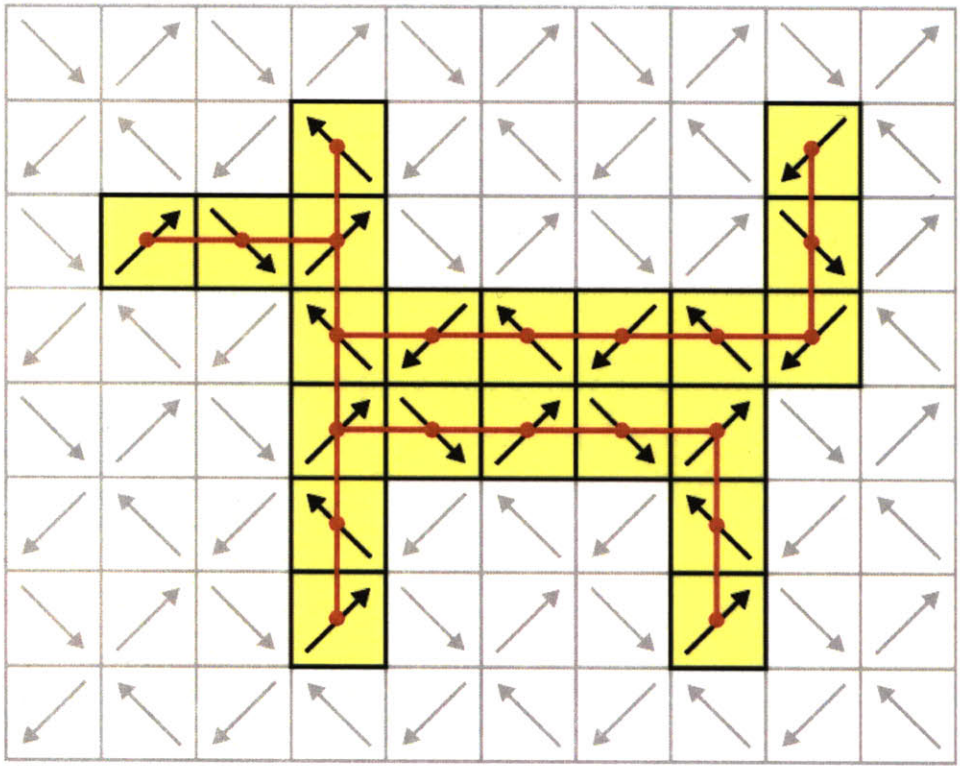


Figure 4-6 A branched spanning tree, in this case a rough approximation of a dog.

Figure 4-7 demonstrates two more of the multiple possible branched spanning trees that could construct this same dog. Nothing is said of the optimality of any of these trees for folding.

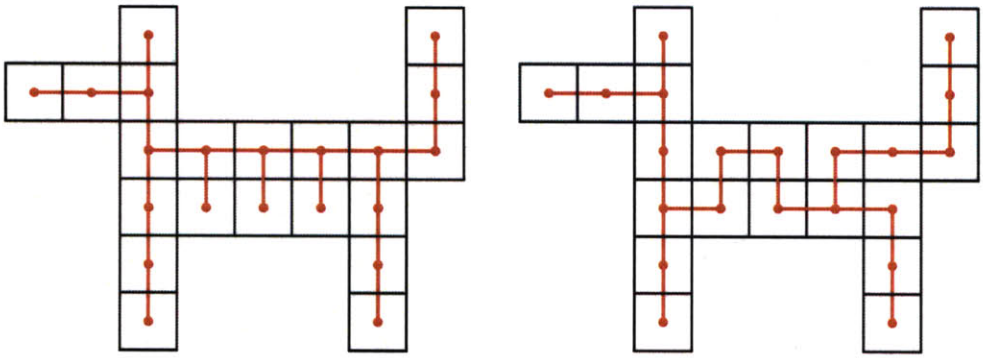


Figure 4-7 Two more branched spanning trees for this dog graph.

Figure 4-8 represents the construction that will allow us to escape this unfoldable dilemma – at a cost of resolution. Each of the original squares (or tiles) in the object is divided into 4 sub-tiles. This gives a perimeter one tile wide around the original spanning tree. By connecting the centroids of the tiles in

the perimeter we are left with an Eulerian path that visits every tile in our new graph. This construction is expensive in terms of resolution, we now need 4n tiles (where n was the number of tiles in the original object) to produce the same object. This green Eulerian loop can be divided at any single connection to produce a single, un-branched spanning tree that is foldable.

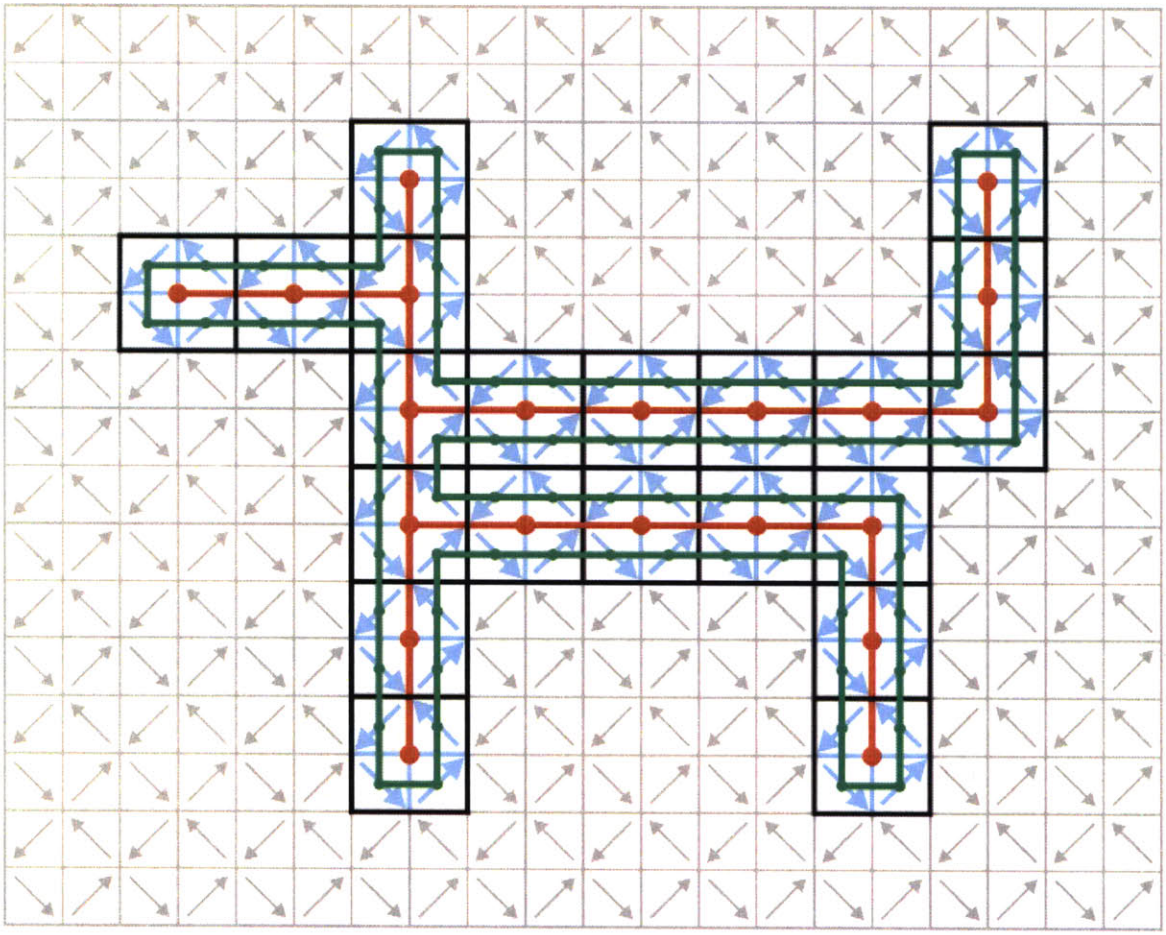


Figure 4-8 Each square (or tile) in the original graph(black) is divided into 4 equal sub-tiles(grey). An Eulerian path (green) that follows the perimeter of the original spanning tree (red) results.

Figure 4-9 visually explains why this construction works, and Figure 4-10 demonstrates how one can construct any object 'additively'. The six examples of Figure 4-9 are the enumerated possibilities for nodes in these graphs where the node has 0 connections (a), 1 connection (b), 2 connections (c,d), or 3 & 4 connections (e,f) to other nodes in the graph. Figure 4-9(f), illustrates that with 8 tile faces for the 2x2 subtiles that make up the original tile, there are 4 entry and 4 exit (total 8) paths, one for each face.

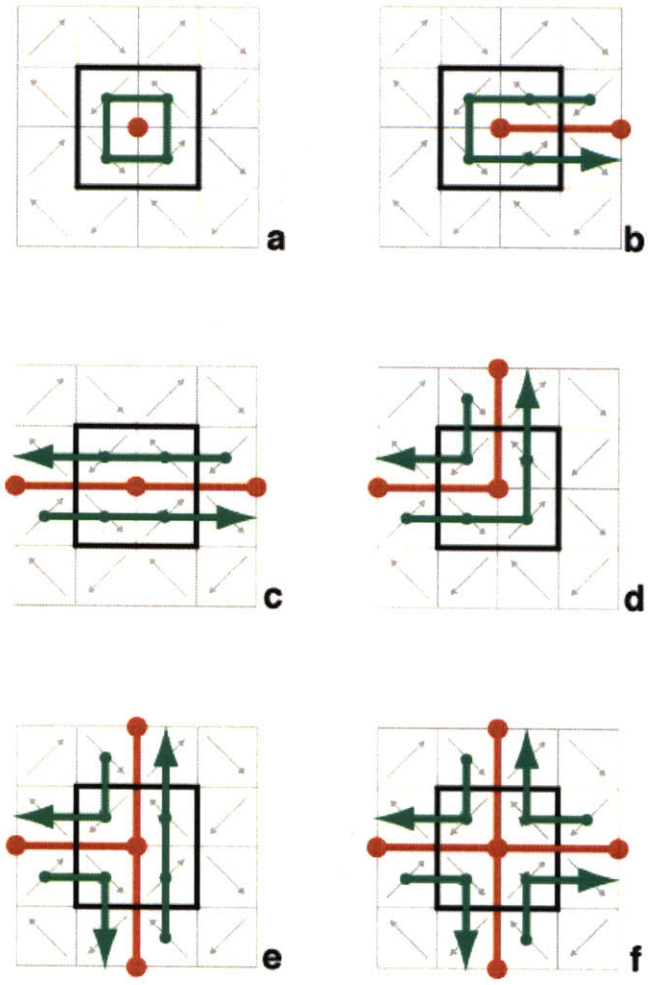


Figure 4-9 Enumerated cases (ignoring symmetrical cases) for paths in and out of each face of the sub-tiles (grey) of original tile (black).

In Figure 4-10(a) a whiskey barrel (yellow tile) is added around the dog's neck. The new node has a 4 sub-tile Hamiltonian around it, similar to the zero path connected node in Figure 4-9(a). This new Hamiltonian will be joined to the original Eulerian loop of the dog in Figure 4-8 by severing both paths around the new connecting branch of the spanning tree shown in Figure 4-10(b). These two paths are highlighted in bold.

The two new connecting paths are shown in Figure 4-10(b) with the removed segments highlighted in yellow. By similar constructions any additions can be made to the dog, or indeed any object built additively from a single starting node.

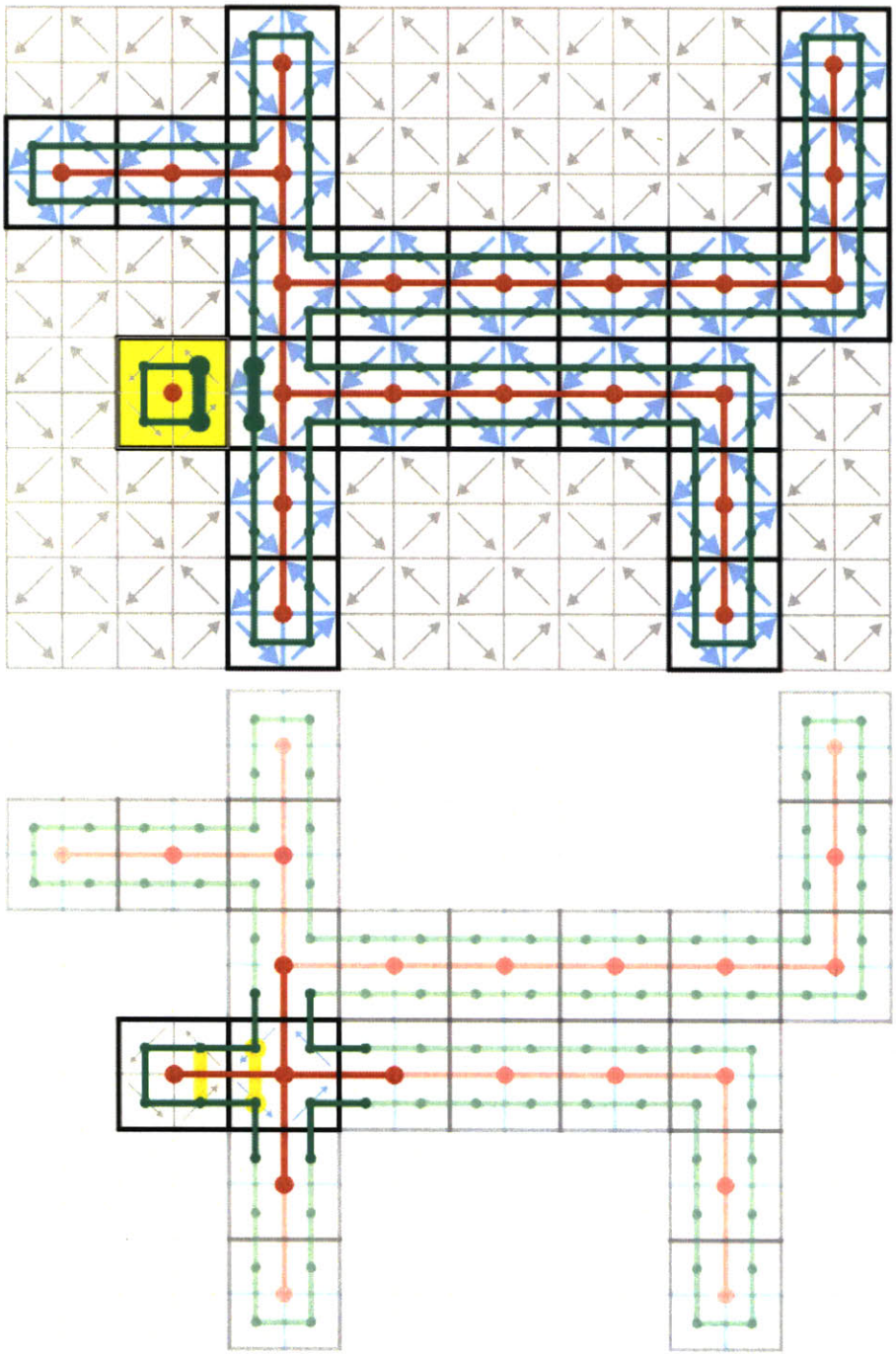


Figure 4-10 a. (top). The addition of a whiskey barrel to the dog (yellow tile). The node of the new tile is surrounded by the 4 sub-tiles Hamiltonian of the original tile. 4.9 b. (bottom) Two new paths are created connecting the whiskey barrel Hamiltonian to the Eulerian loop of the original dog. The two new paths fall on either side of the new branch added to the tree (red). The two yellow paths are removed. By similar constructions, new nodes can be added anywhere to the spanning tree.

4.5 *Tile types for folding in 2 dimensions.*

How many tile types are required to achieve these folds?

Figure 4-11 demonstrates that for a tile set where positive and neutral bonds are available (as might be the case in a set assembling on a liquid surface with hydrophobic and hydrophilic surfaces)only two tile types are required, patterned as illustrated in Figure 4-11a, and folding according to the enumerated sequences. As is seen in figure Figure 4-11b, although this will fold according to these rules, the resulting object will have neutral or weak bonds (+/o), and hence the resulting structure will be a weak crystal.

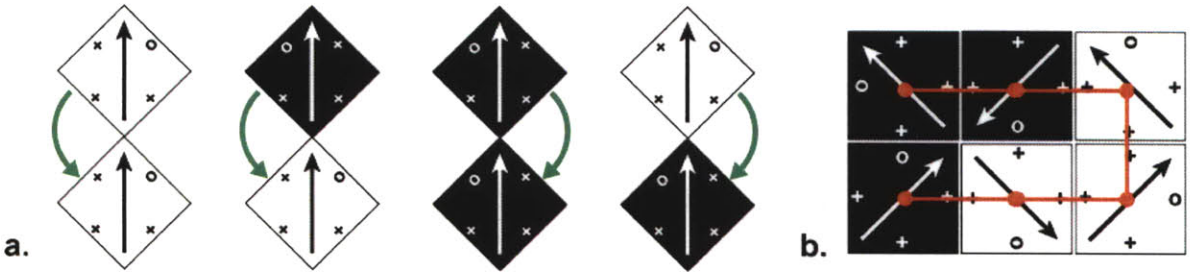


Figure 4-11 a) 4 sequences and resulting folds where (+) faces attract and (o) faces are neutral. b) One can see from this sequence that with only two tile types with a neutral bond (o) many bonds are weak (o/+)

Conversely for a system where attractive and repulsive bonds are possible, as might be the case in a system magnetically or electrostatically actuated there are 4 required tile types and corresponding edge patterns as defined in Figure 4-12a. Like faces in this system repel eg. N/N, S/S and unlike faces attract eg. N/S, S/N. Figure 4-12b. enumerates the 4 clockwise and 4 counter-clockwise folds possible in such a system. Figure 4-12c. shows the resulting plane tiling for such a 2D crystal where all bonds are strong or attractive (N/S or S/N). Any object can be superimposed on the plane tiling in Figure 4-12c. which implies the encoding, or tile sequence that will fold to give that structure. Figure 4-13 is demonstrative of the folding of the letters M.I.T from sequences of the 4 tile types in Figure 4-12a. The necessary tile sequences are shown in figure Figure 4-13 where each tile is denoted by color (B)lue, (Y)ellow, (G)reen, and (R)ed. Note that these sequences will depend upon the starting point, or the point at which the Eulerian loop is broken. In Figure 4-13 the starting point for each of the letters M.I.T is marked with a red circle. Following the arrows on each tile from tail to head will demonstrate the folding path for these structures.

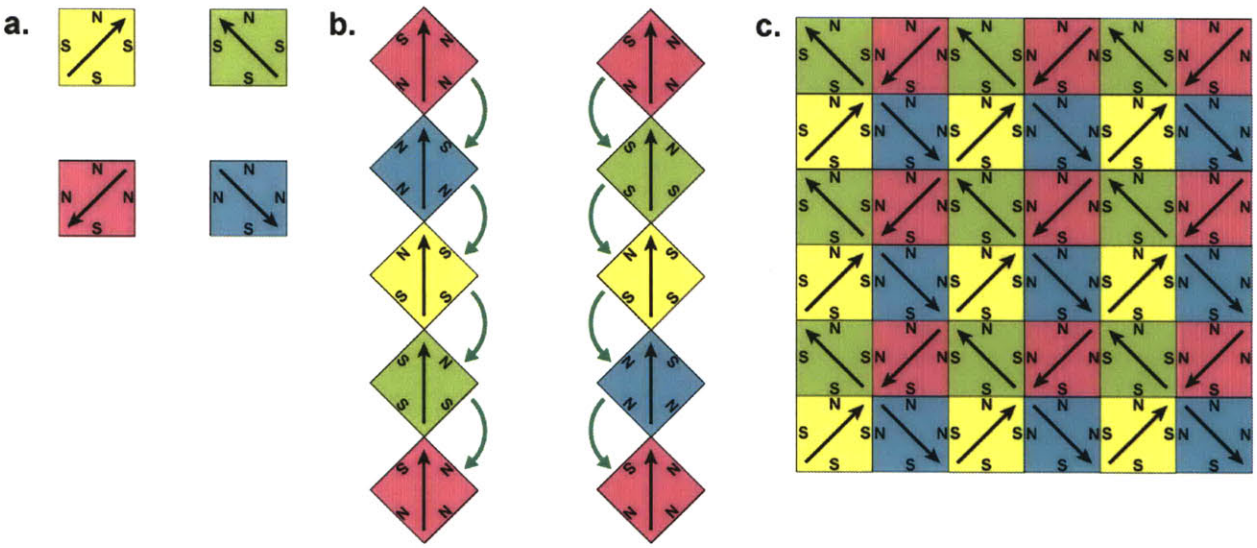


Figure 4-12 a) 4 tile types required for strong bonding. Like faces repel (N/N or S/S), Unlike faces attract (N/S or S/N). b) All possible folding combinations are presented in these two strings, 4 clockwise, 4 counter-clockwise. c) Any object can be mapped onto this tiling of the 4 tile types. The superposition of an object onto this tiling implies the sequence that will give that structure.

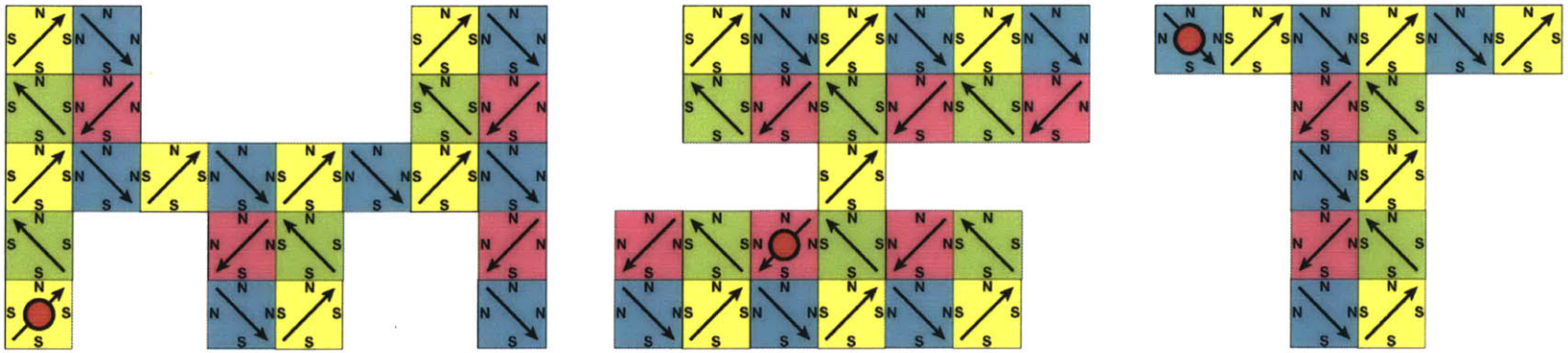


Figure 4-13 The letters MIT folded from linear strings of 4 tile types. The strings begin at the red circle and proceed from the head of each arrow to the tail of the subsequent arrow. The resulting tile sequences for these structures are shown below each letter.

4.6 Proof in 3 dimensions.

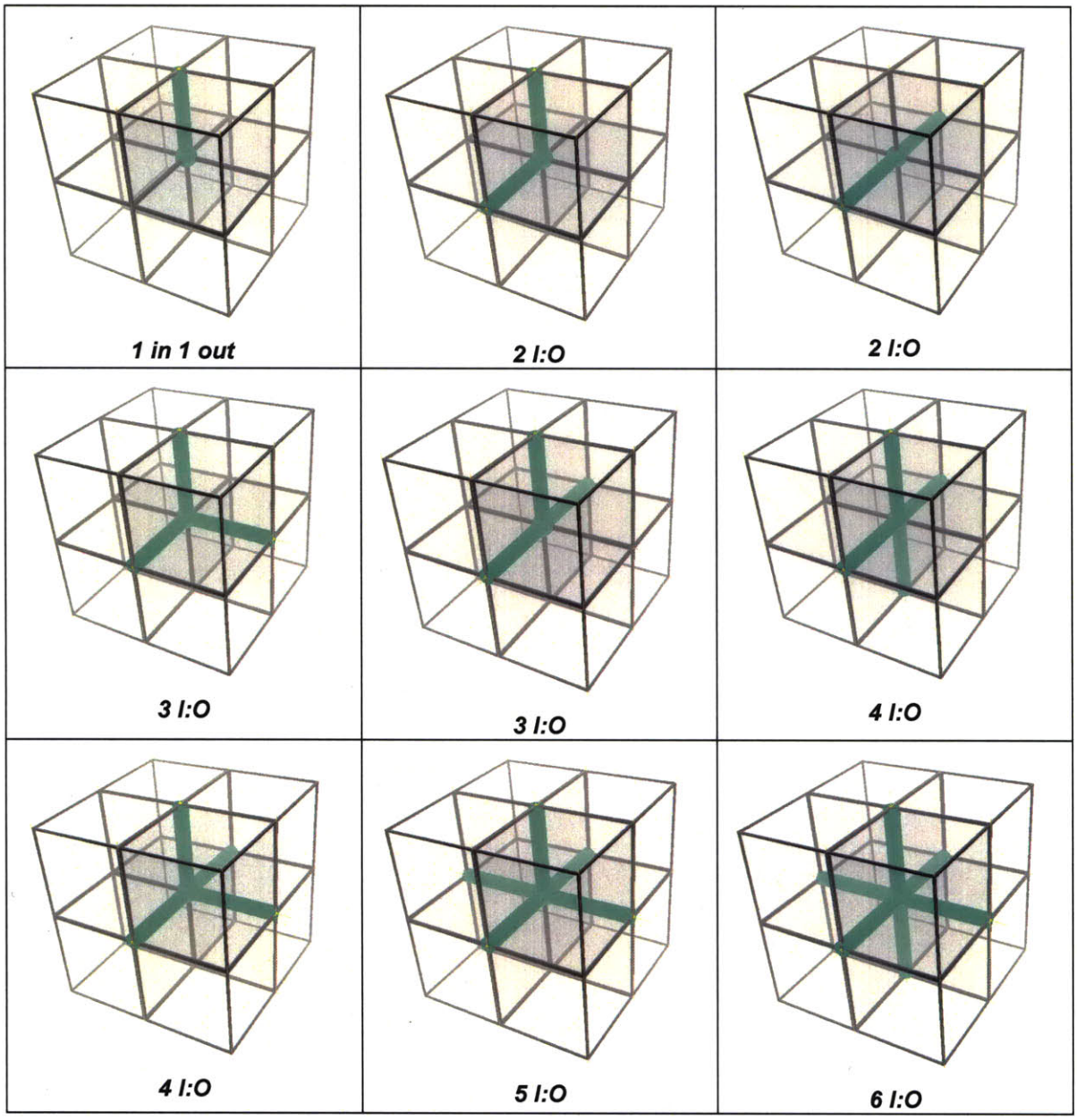


Figure 4-14 The 9 enumerated cases for a 2x2x2 cube composed of 8 sub-cubes, analogous to the 6 cases presented in Figure 4.8. Symmetrical cases are again ignored. The Number of faces requiring an entry and an exit path is listed as eg. 6 I:O (6 faces, in/out).

A similar construction to the one presented above for 2 dimensions can be developed by analogy to 3 dimensions. First I will disclose the case for a cube with 2x2x2, or 8, sub-cubes. This will be seen to be awkward in terms of manufacturability, and not minimal in terms of resolution, and hence a

construction will then be developed for a rhombic hexahedron composed of 6 sub – right-angled tetrahedra.

Figure 4-14 demonstrates the 9 analogous cases to those presented for the 2D case in figure 4.8. Where 1 entry and 1 exit path was required for each side of the square tiles in the 2D case, the 3D case requires 1 entry and 1 exit path on each face of the cube. There are 4 sub-cube faces per face. The center of each cube in Figure 4-14 can be considered a node in a 3D graph. The green lines extending from this central node in each of the 9 cases is a branch in the spanning tree for the 3D graph, analogous to the red lines in Figure 4-9.

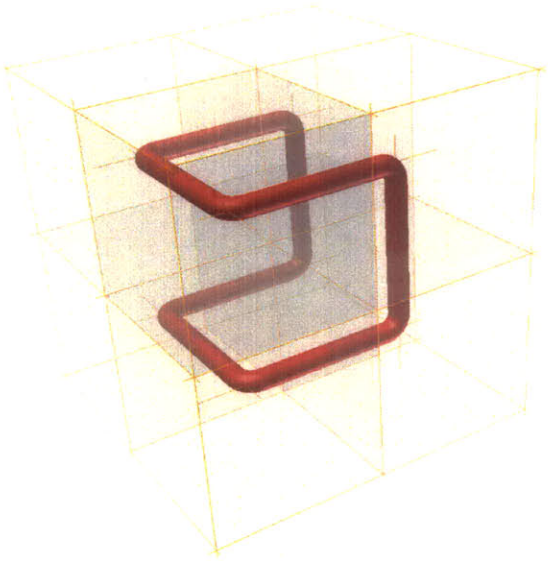


Figure 4-15 Connected Hamiltonian path that visits all sub-cubes of the original cube – represented as a red pipe. The node of a graph is again at the center of this cube.

Figure 4-15 is the 3D Hamiltonian that visits all sub-cubes of the original cube, analogous to the hamilttonian about the central node for tile and 4 sub tiles of Figure 4-9(a). Similar to the additive construction of Figure 4-10, the Hamiltonian of the 3D cube can be split between any 2 sub-cubes to allow one entry and one exit path on any given face of the cube. Figure 4-16 demonstrates this construction for a single face and green arrows represent the entry and exit paths for a single face.

Figure 4-17 demonstrates the additive construction of (a) 2 joined cubes about a single face, and (b) 3 joined cubes around a central cube which has had it's Hamiltonian cut twice to allow an entry and an exit path on two faces.

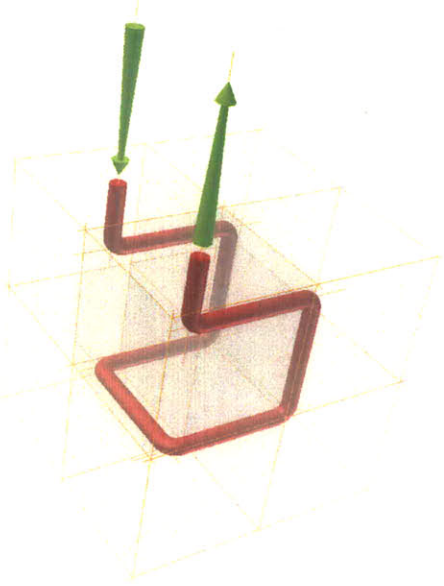
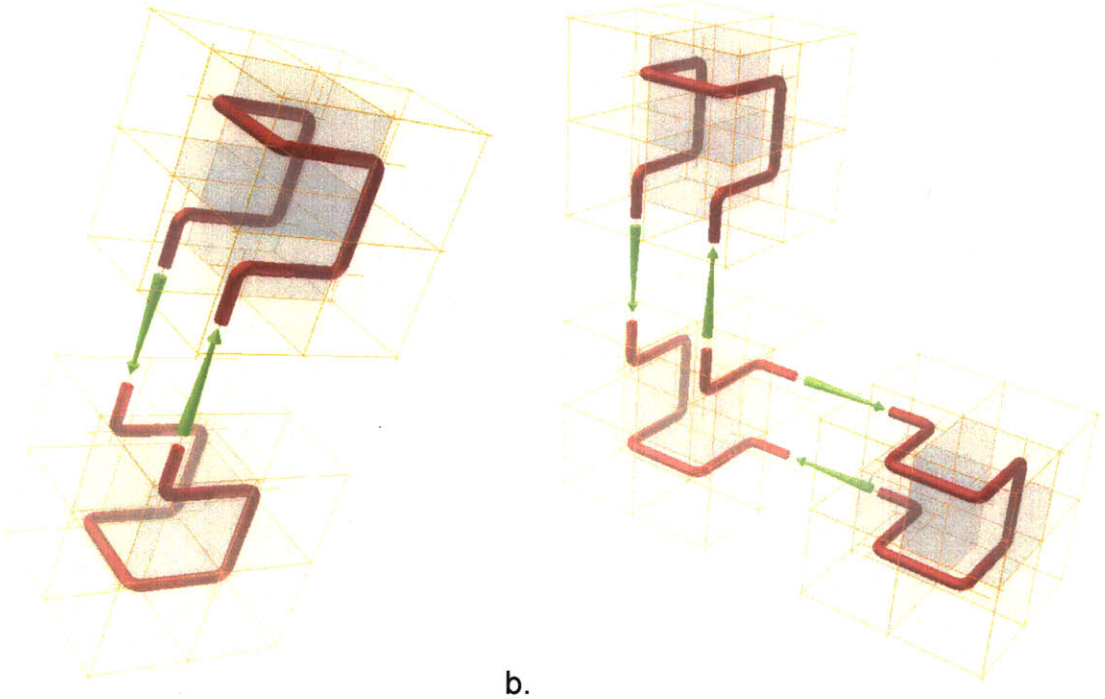


Figure 4-16 The Hamiltonian for a single 2x2x2 is split resulting in one entry and one exit path via single faces of 2 sub-cubes of the original cube.



a. b.

Figure 4-17 a) Two cubes are joined additively via a single face. b) similarly three cubes are connected about a single central cube which has had it's Hamiltonian cut on two faces.

Figure 4-18 demonstrates the completely populated case of 6 cubes attached to all 6 faces of a central cube. The Hamiltonian of the central cube has been cut 6 times to accommodate the six entry and exit paths via sub cubes to the 6 attaching cubes. This figure represents a single Eulerian loop that connects all 56 sub-cubes of the 7 original cubes. This loop can be traced between cubes by following the green arrows that represent connected entry and exit paths. It will be seen tracing this loop that the paths do not intersect in any single sub-cube. This is analogous to the 2D case in Figure 4-9(f).

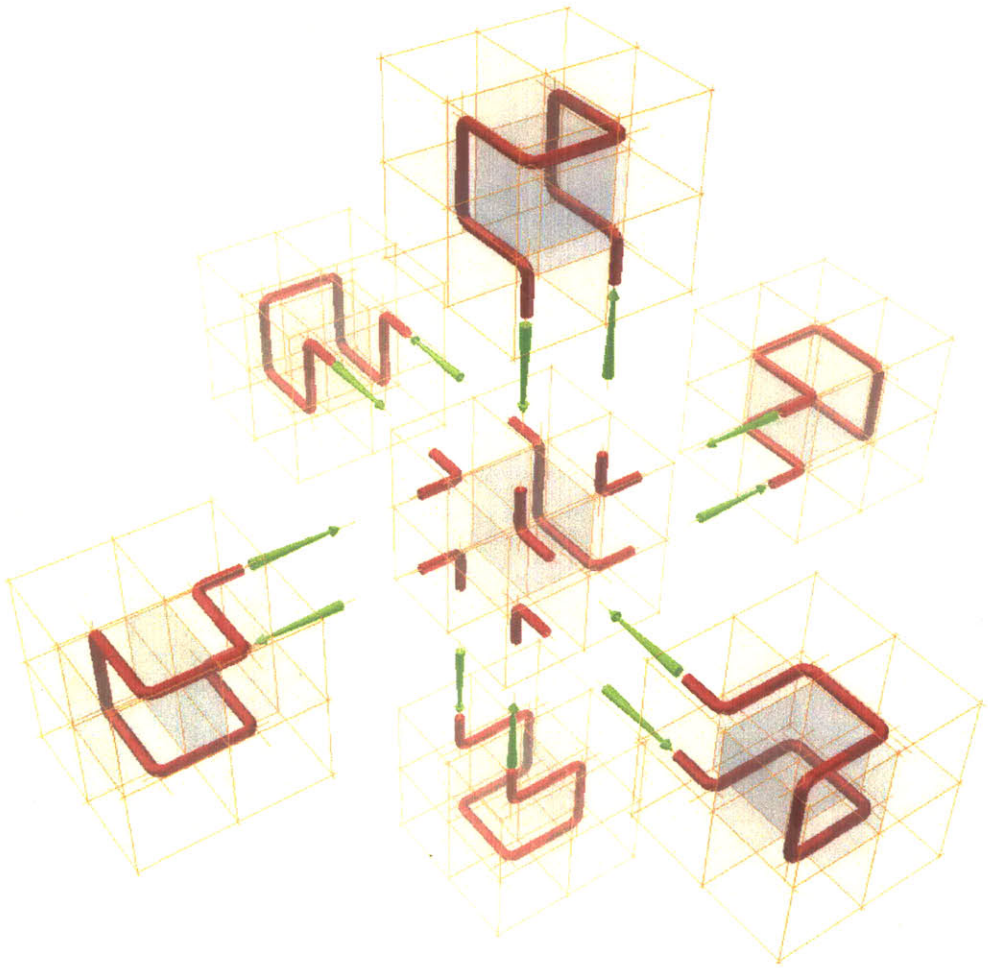


Figure 4-18 The 6 face case for a central cube connected on all possible faces to 6 other cubes. The Hamiltonian for the central cube has been cut 6 times to allow this construction and following the hamiltonian's for each cube, and the connecting paths (green arrows) one can see that this is a single, non-intersecting, Eulerian loop that visits all 56 sub-cubes of the 7 original cubes.

Figure 4-19 suggests that there is a simpler unit and set of sub-units that will achieve the same goal. Two redundant connecting paths between sub-cubes are left remaining after severing six of these

paths to reconnect them to the 6 cubes attaching to each of the 6 faces of the central cube. This would seem apparent from the fact that to fill 3 space we need only be able to translate positively or negatively in the 3 cartesian co-ordinates X,Y,Z. ie.+X,-X,+Y,-Y,+Z,-Z.

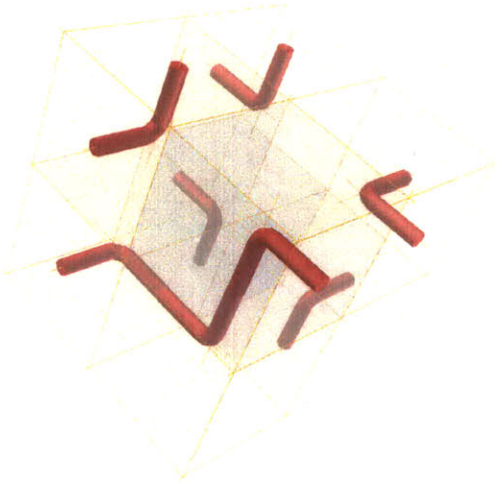


Figure 4-19 There are 8 connecting paths between the 8 sub-cubes of the original cube. When fully connected by severing six of these paths, this leaves 2 redundant connections.

Figure 4-20 demonstrates the 4 sub-cube types required to achieve all of the folds and translations required to visit all 8 sub-cubes of a cube. Each fold is about an edge, represented by a vector arrow. The input edge is denoted by a red vector, and the exiting edge by a green vector. The head of the input edge vector must align to the head of the outgoing edge vector. These 4 translations are believed to be the minimal set required to make all of the folds required in the constructions above.

Figure 4-21 Illustrates the connections that can be made between sub-cubes about edge vectors. Similar to the 2D case in Figure 4-11 and Figure 4-12, the folding process proceeds sequentially down the linear chain, with a binary (clockwise or counter-clockwise) folding instruction about each edge. It can be seen for all but the simplest sequences that the linear string of sub-cubes in Figure 4-21does not necessarily fold out flat into a linear chain where a face of each cube would sit flat on that plane. This non-planarity would make the string difficult to construct by existing micro or nano-manufacturing techniques.

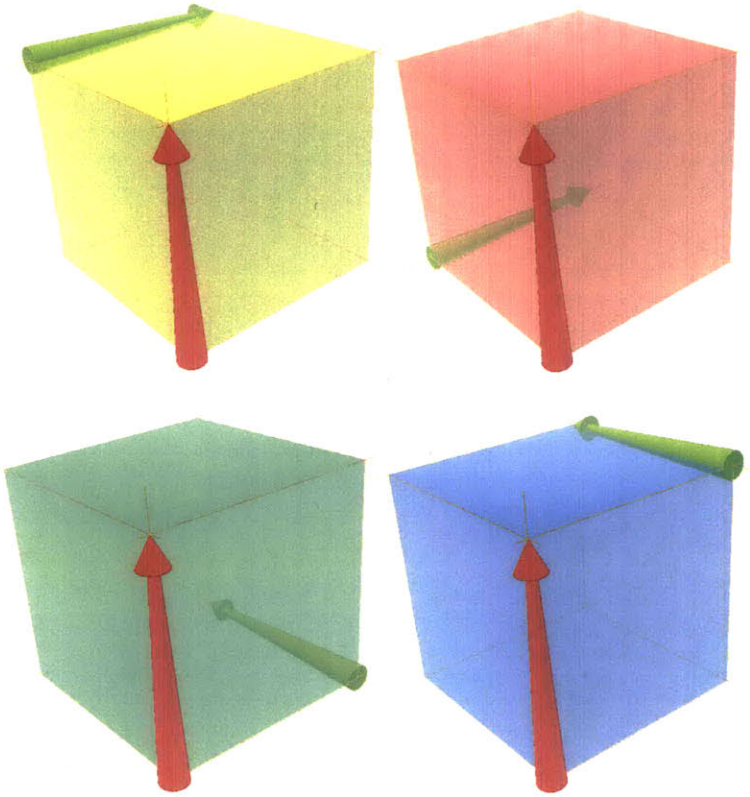


Figure 4-20 The 4 sub-cube types required to achieve all folds necessary to make a 2x2x2 cube from sub-cubes, and all folds necessary to achieve the construction of figure 4.17 and analogous Eulerian paths for any set of connected cubes and sub-cubes.

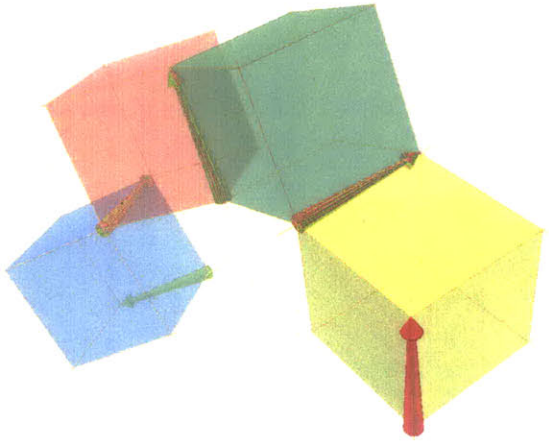


Figure 4-21. Example of connected sub-cubes in a linear chain pre-folding. Similar to the 2D case there is a sequential binary folding instruction about each connecting edge (clockwise or counter clockwise relative to the edge vector.

4.7 *A 'minimal', and 'manufacturable' polyhedron for folding arbitrary 3D structure.*

Section 4.6 presented a proof by analogy that a string of edge connected cubes can fold sequentially about the connected edges to make an arbitrary 3 dimensional shape (or voxelated approximation of that shape). Similar to the 4n cost in resolution of tiles to the 2D case due to this construction there is an 8n cost in resolution of the voxels given that 2x2x2, or 8 sub-cubes are required in the original division of the cubes connected in a branched spanning tree of a 3D object. It was also seen in Figure 4-21 that the connected string of sub-cubes will not necessarily (or even probably) lie flat in a plane for a given object, and hence would be difficult to manufacture by the principally 2D systems available at the small scales where such a system might be desirable for manufacturing complex 3D structures.

Figure 4-19 suggested that there might be a simpler 'voxel' (or cube in the constructions of section 1.6) that could be sub-divided into another space-filling polyhedra that still enables arbitrary 3D folding. The requirements on these primitives are:

To reach all 3 dimensions by translation from each voxel positive and negative translations in all 3 dimensional axes are required.

One entry and exit sub-face per face is required for the Hamiltonian construction used thus far.

The desirable 'voxel' has 6 faces - one for positive, and one for negative translation in each axis.

The voxel will have 12 sub-faces derived from the sub units that comprise it, two on each of the 6 faces.

The rhombic hexahedron of Figure 4-22 satisfies all of these constraints. The 6 faces of the rhombic hexahedron are similar rhombi. This polyhedron can be sub-divided into 6 right-angled tetrahedra, described in Figure 4-23. The right-angled tetrahedron has two edges with a 90 degree angle between faces and 4 edges with a 60 degree angle between faces, as illustrated.

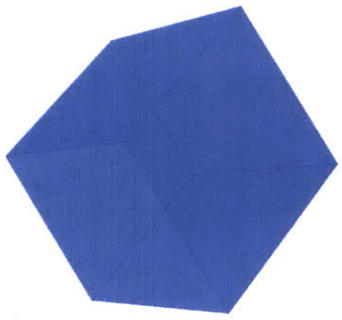


Figure 4-22. The Rhombic Hexahedron. 6 similar Rhombic faces.

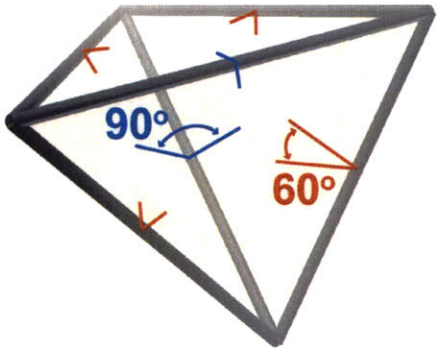


Figure 4-23. The right-angled tetrahedron. Two pairs of faces which are subtended by a 90 degree angle. These two pairs of faces are orthogonal to eachother.

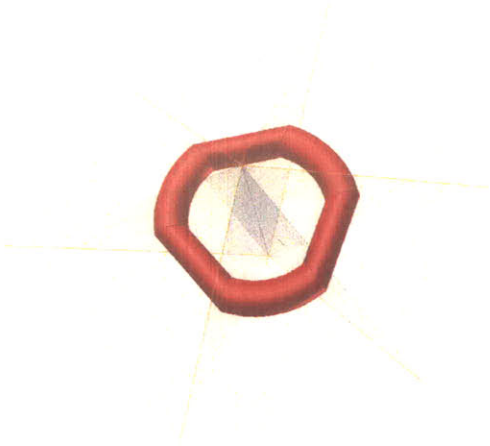


Figure 4-24. A rhombic hexahedron comprised of 6 right-angled tetrahedral, connected by a Hamiltonian loop that visits the centroid of all 6 sub-tetrahedra of the rhombic hexahedron.

Figure 4-24 demonstrates a rhombic hexahedron sub-divided into 6 right-angled tetrahedron. The Hamiltonian path analogous to that of Figure 4-15 is illustrated. The rhombic hexahedron so presented is space filling, 4 of which can be stacked to produce a rhombic dodecahedron which is also a space filling solid.

Now by analogy to the case presented in section 1.6 I will demonstrate that the right-angled tetrahedron (right-tetra) and the rhombic hexahedron (hexahedron) is similarly capable of folding sequentially into an arbitrary 3D space filling structure.

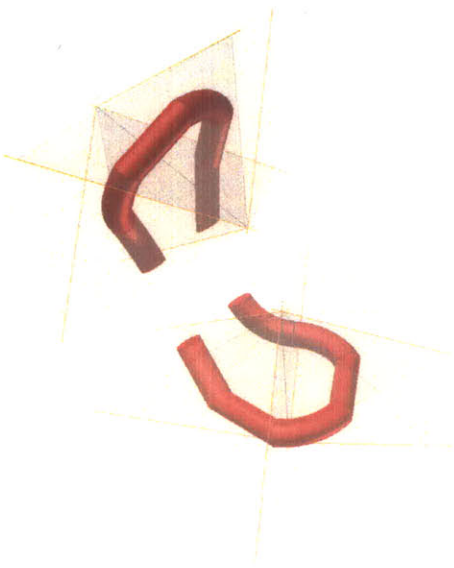


Figure 4-25. Analogous to Figure 4-16Figure 4-17, two Hamiltonian loops may be split and reconnected additively to eachother leaving a connected Eulerian loop.

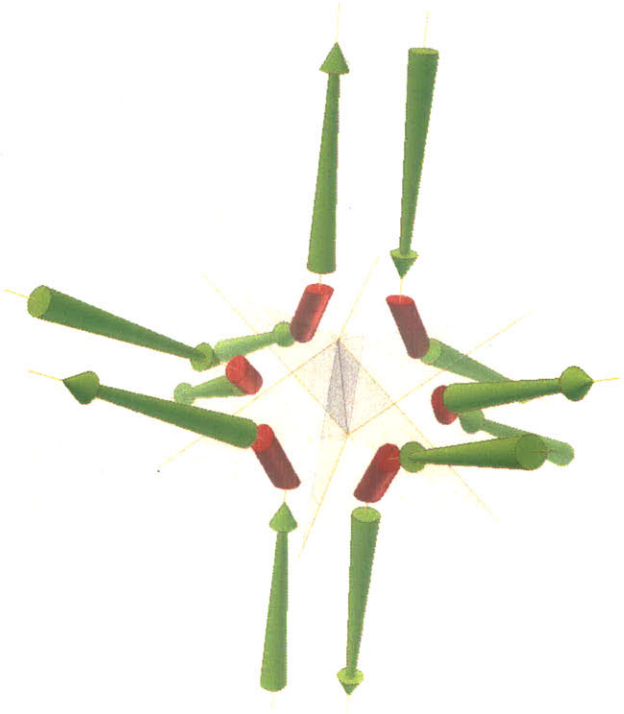


Figure 4-26. one entry and one exit path via faces of sub right-tetra for each of the 6 faces of the hexahedron.

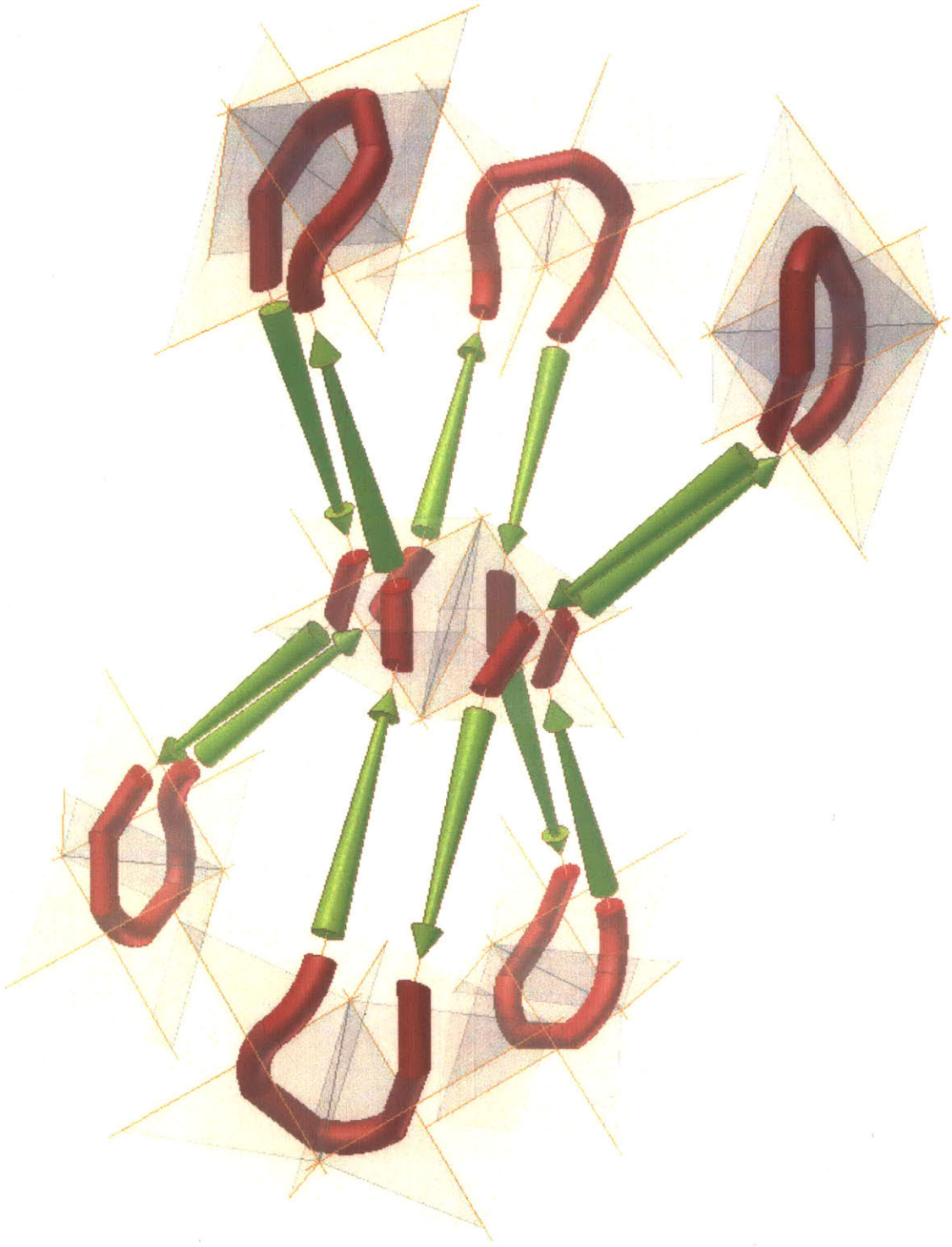


Figure 4-27. 7 connected rhombic hexahedra about a single central hexahedron via a single Eulerian loop that visits all 42 (7x6) sub-right-angled tetrahedra. The resulting loop is both foldable and non-intersecting.

By the additive construction method it is necessary to show that sufficient entry and exit paths are available - a pair for each of the 6 faces of the hexahedron. Figure 4-25 demonstrates the severing of one connection between sub right-tetra in each of two hexahedron primitives. They can be seen to rejoin with a connected, non intersecting Eulerian loop. Figure 4-26shows that a pair of entry and exit paths for each face is possible.

Figure 4-27 shows the analogous case to the 7 cubes in Figure 4-18. All 6 faces are connected to 6 other hexahedron via one entry and exit path per (sub) right-tetra face. Once again, the path can be followed and be seen to be a non-intersection Eulerian loop. The construction in Figure 4-27 has been manually reproduced in laser-cut acrylic model right-tetrahedron, and indeed unfolds to a linear string of edge connected right-tetras.

Figure 4-28 illustrates that right-tetrahedron will unfold to a linear chain that is non-intersecting when constrained to a plane by alternating vertices. Many alternatives for constructing very large numbers of these primitive units and methods for stringing them together might be imagined.

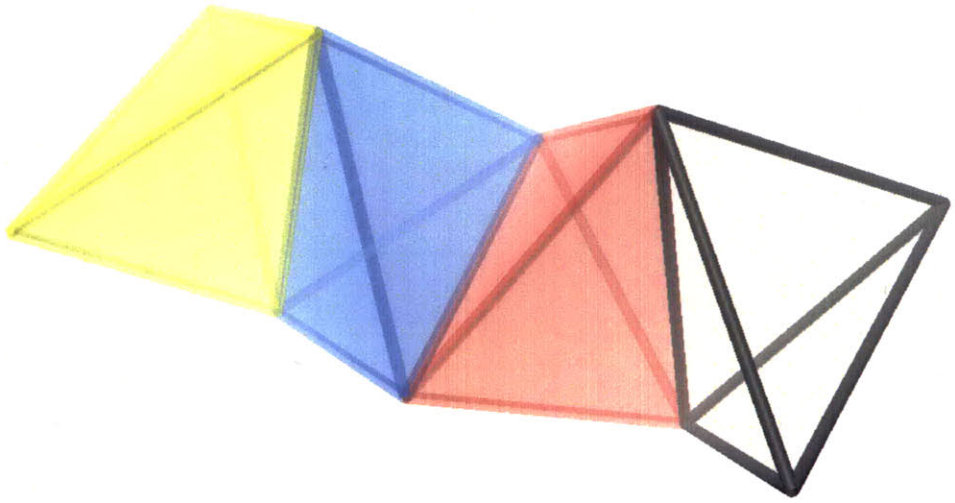


Figure 4-28. 4 Edge connected Right-Tetrahedron. Connected by alternating opposing long edges a string of right-angled tetrahedron will always unfold to a linear chain that sits flat on a plane upon alternating vertices.

4.8 A physical implementation of the 2 dimensional case.

Laser cut tiles were constructed with the geometry described in Figure 4-29. 3 x1.5mm NdFeB magnets (amazingmagnets.com) were fitted into flexure based snap-fits on each tile according to the polarities described in Figure 4-12. To ensure predicated folding (sequential folds around each

vertice before the next tile), the strings of tiles were fed through a constraining channel, folding as they exited the channel.

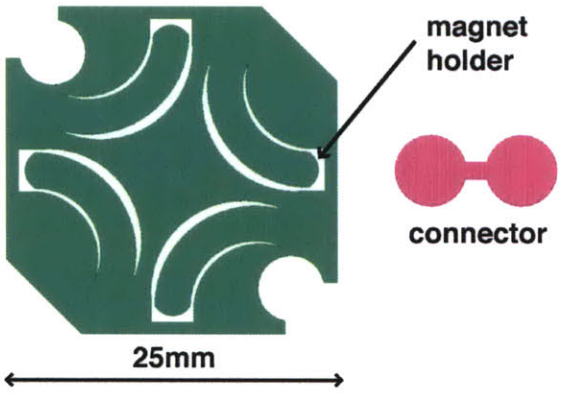


Figure 4-29. Laser-cut acrylic tiles for 2D predicated folding on air table.

Figure 4-31Figure 4-32, & Figure 4-33 respectively demonstrate the folding of the letter's M.I.T from linear strings of tiles. The structure of each letter is specified by the sequence of tiles, determined as per the technique shown in Figure 4-12 &Figure 4-13. Figure 4-33 is the final output of the complete set of letters, M.I.T. All images are frame captured by video taken of the folding. Each tile is made neutrally buoyant and close to frictionless by virtue of the air table supplying a flow of air underneath each tile. The holes in the air table were placed on a hexagonal lattice at 4mm intervals. Airflow was supplied via a garden blower attached to a sealed box with the perforated plate on top. The yellow grid in background also serves as an air baffle to equalize air-flow from the garden blower as it enters the air-box. Each letter took approximately 30 seconds to fold, but the speed is obviously scale dependent.

Figure 4-30. M.I.T. as folded by the above described technique.

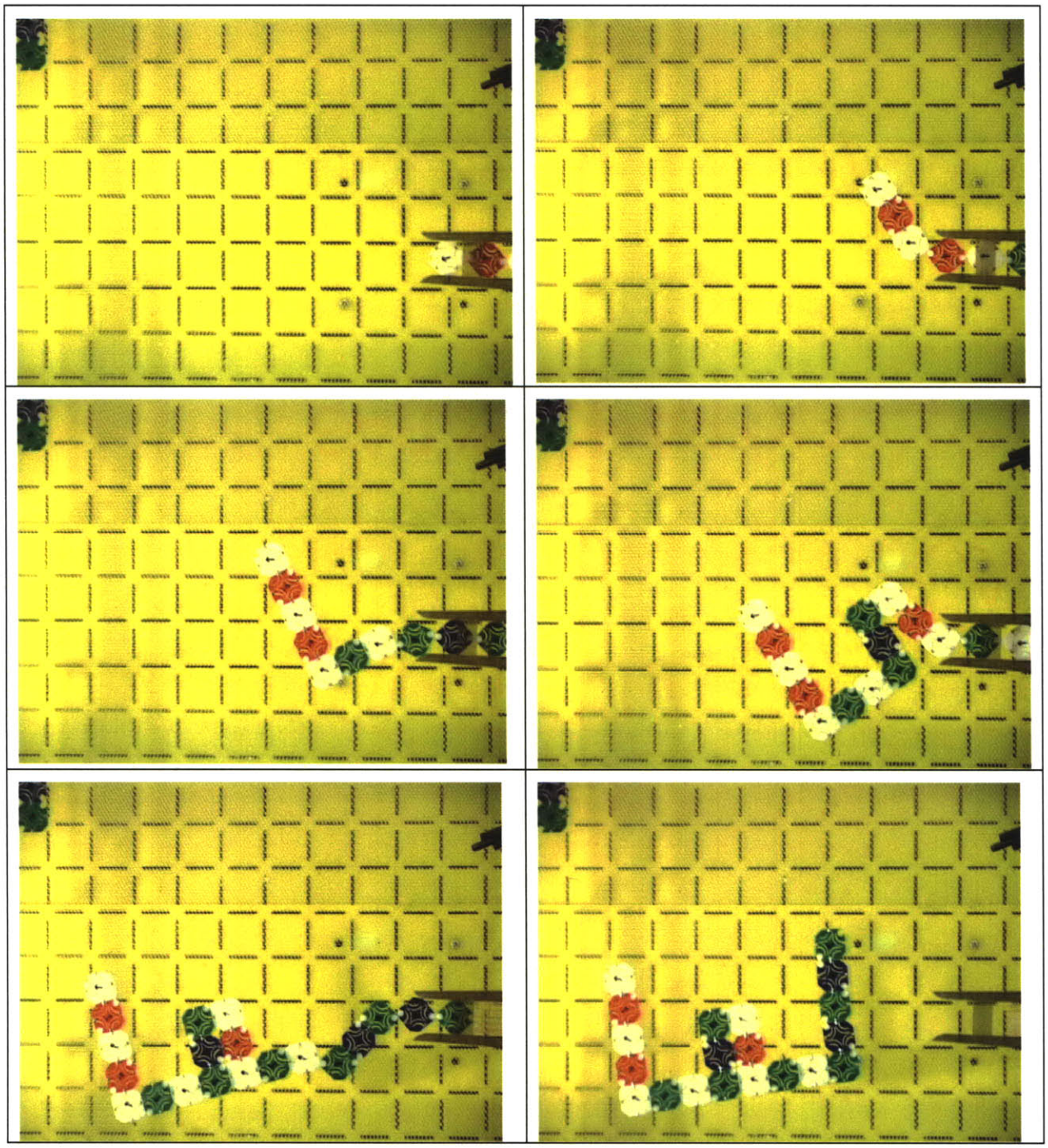


Figure 4-31. Sequential folding of the letter M as it exits a constraining channel. The tile sequence specifies the exact structure.

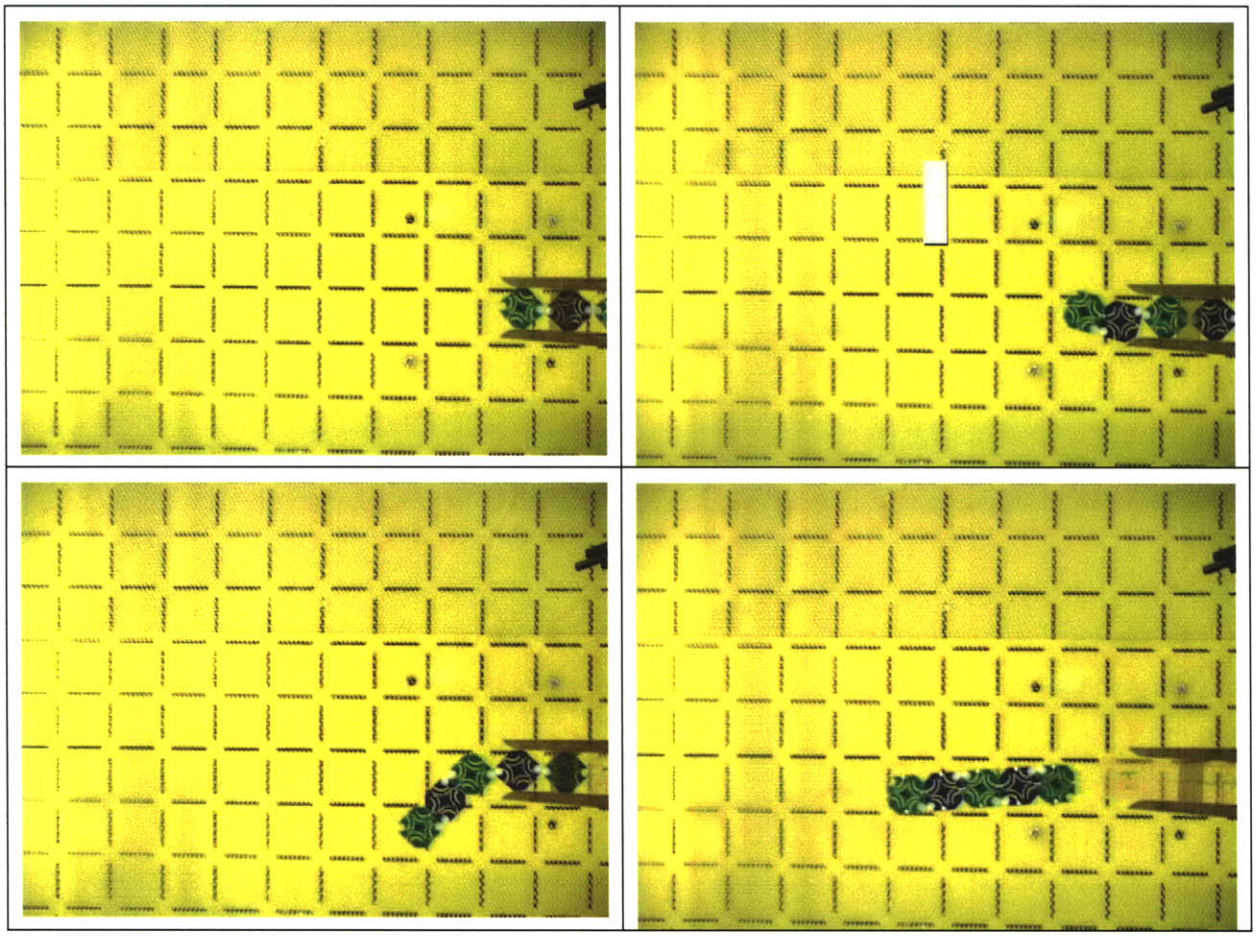


Figure 4-32. Sequential folding of the letter I as it exits a constraining channel.

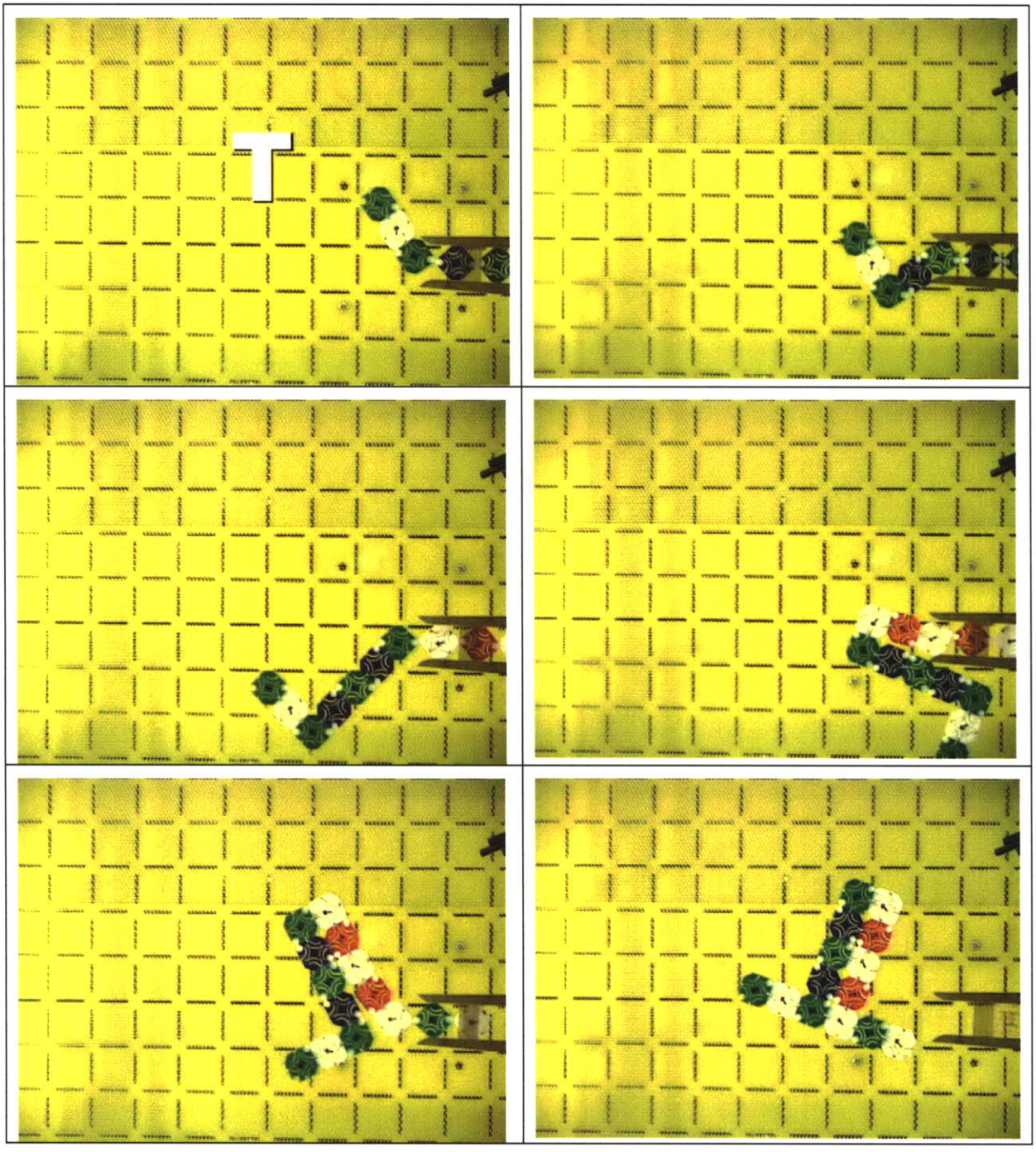


Figure 4-33. Sequential folding of the letter T as it exits a constraining channel.

4.9 Schematic for reconfigurable folding chain.

With the assistance of Dan Goldwater a schematic was designed for building an electronically reconfigurable folding chain as per the above system. Each cell in the schematic represents one tile or polyhedron in the chain. It was desirable to design a system which would fold sequentially from the far end of a linear chain with the instructions coming from the near end. All power, and logic for folding instructions can be implemented with an 8 wire bus. Each cell is a 3 bit state machine. The cells are identical through the chain with no need for a termination cell. This would be amenable to mass production as a continuous chain. The assembly scales linearly with the number of cells. The schematic and it's operation is described in more detail in Figure 4-34. No claim is made that this is a minimum implementation in terms of bus lines or any other metric, merely that it would be possible to implement a reconfigurable folding string to fold arbitrary 3D structures programmatically.

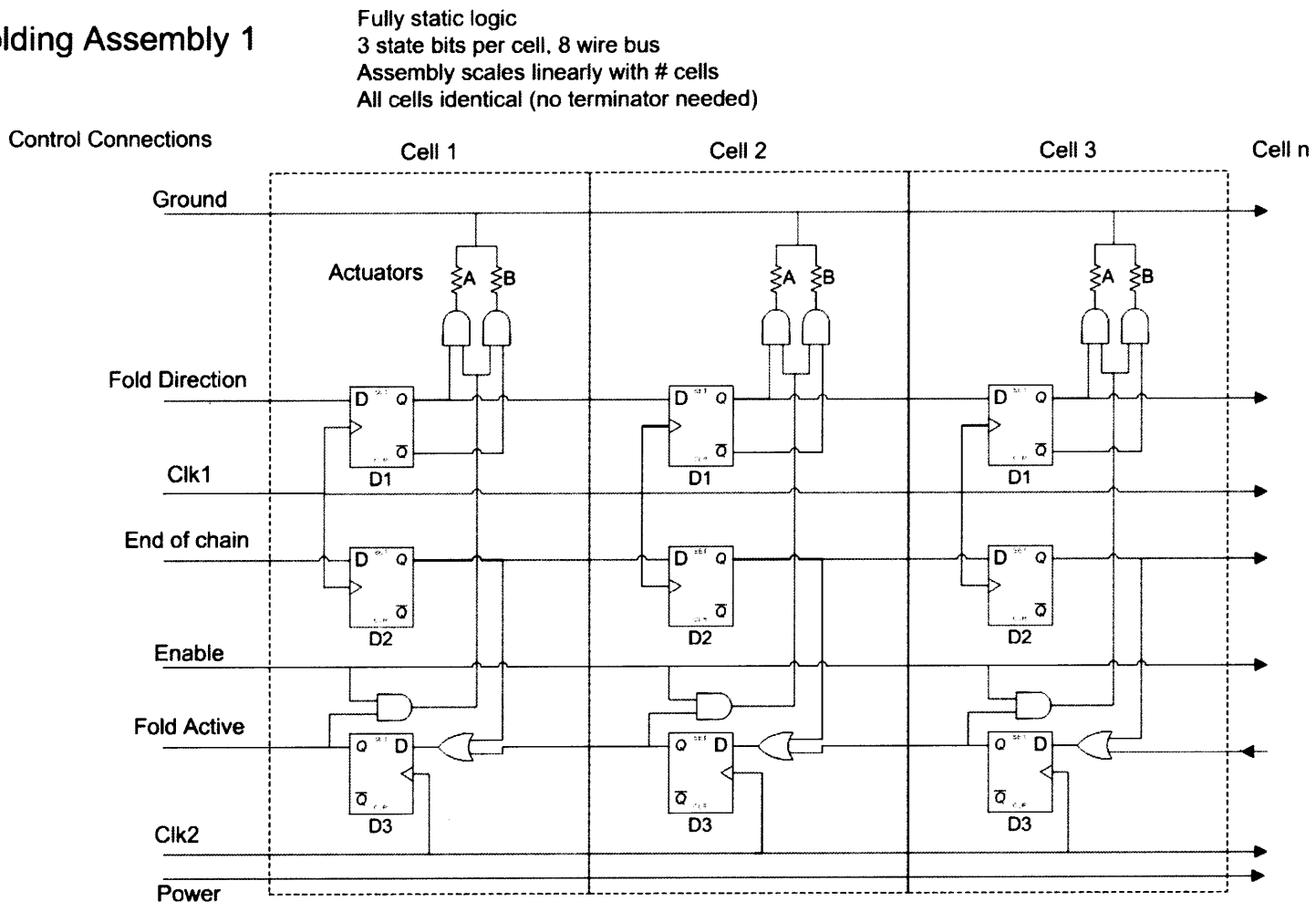


Figure 4-34. Schematic diagram for electronic implementation of a folding chain that folds sequentially from the far end with instructions and power supplied from the near end.

4.10 Manufacturability and foldability

It was emphasized throughout the design process that the string or linear chain of polyhedra / polygons should be easily manufactured with existing or imaginable techniques (including that described in section 1.9). One method that may be interesting is an 'object sequencer' where the individual components are fed into a folding channel from pools of parts such as is illustrated in figure 4.34.

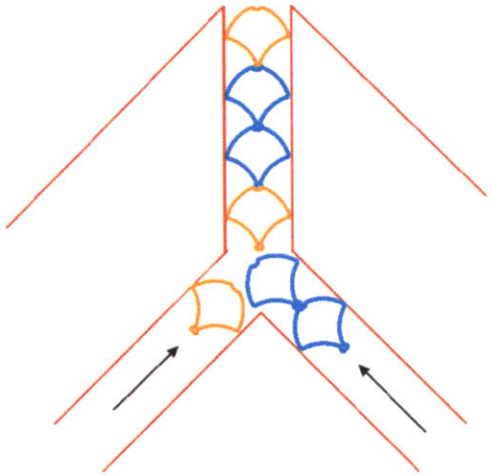


Figure 4-35. Object Sequencer – edge or face programmed polygons or polyhedra are fed sequentially into a folding channel whereupon they fold upon exit similar to section 1.8.

But the manufacturability does not say anything about the foldability. Although the constructions presented herein give a non-intersecting folding path, there will need to be compliance of the entire structure as it folds to allow the 'swinging' of each subsequent part (and the loose, unfolded, string of parts that follows it) into place. This will be especially difficult where the remaining string must fill holes or channels left in the previously folded path. There will be some optimality in terms of the 'foldability' in the choice of the branched spanning tree that is used to connect the graph that defines the pixels or voxels of any given object. I will not discuss that optimality here except to say that it will also depend upon the compliance of all previous 'bonds' in the structure and the exact geometry in terms of truncated corners etc. of the folding parts. Finally, only a few of the cases presented in Table 4-1Table 4-2 have been presented. It is expected that many or all will prove foldable by a similar treatment.

5 Conformational Programming of state in Self Assembling Systems.

5.1 Programmable Assembly at Liquid / Liquid Interfaces

My initial approach to this work in programmable assembly was to extend the well documented area of assembly at liquid/liquid interfaces by the addition of state to the individual parts.

The physical system employed borrows from the work of Whitesides etal and comprises PDMS components at fluid/fluid interfaces and the minimization of surface energies to converge on an assembled state. To date this type of self-assembling system has dominated the literature. For the reasons discussed in chapters 2 and 3 this approach is limited in some very important ways. Firstly all such systems have significant 'program' limitations in that the individual units have a set state determined prior to assembly. In essence the only thing being programmed is the basic attractive and repulsive forces between given faces. These systems all build periodic crystalline type structures which have inherently limited complexity. In essence all of these systems are seeking out a global energy minima and hence a static equilibrium.

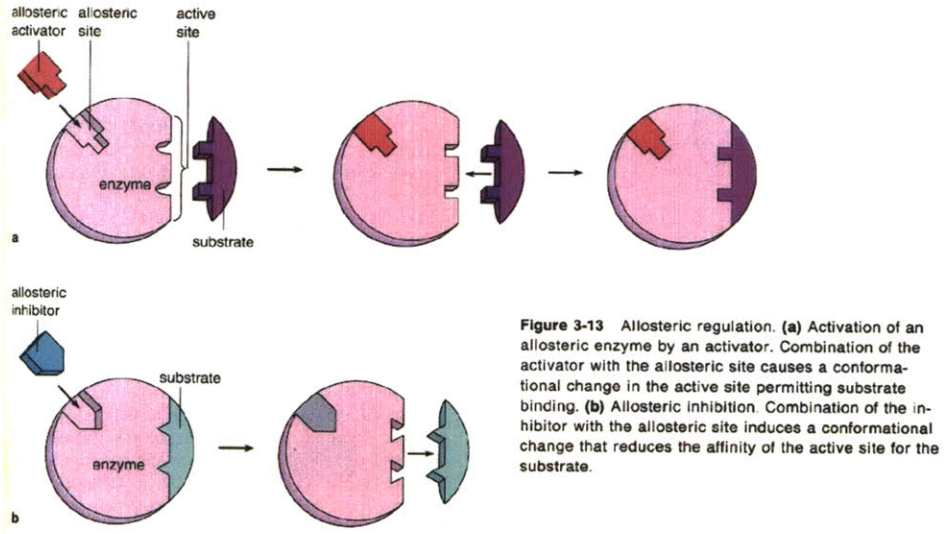


Figure 3-13 Allosteric regulation. **(a)** Activation of an allosteric enzyme by an activator. Combination of the activator with the allosteric site causes a conformational change in the active site permitting substrate binding. **(b)** Allosteric inhibition. Combination of the inhibitor with the allosteric site induces a conformational change that reduces the affinity of the active site for the substrate.

Figure 5-1 Schematic of allosteric 'state' in biochemical systems. Ptashne[1]

Biological systems do much more than this simple type of self-assembly, though crystallization and periodic arrays are also within it's toolkit. Perhaps the most important difference is that biological systems are capable of being non-equilibrium or dynamic systems, and that some biological components may also be thought of to have more than one pre-programmed state. Biology employs the allosteric function heavily in the control of self-assembly, particularly in sub-cellular (bio-macro-

molecule) systems Ptashne[1]. A schematic diagram of allostery can be seen in Figure 5-1. The various allosteric configurations of a part can in some respects be considered states of a simple state machine. Biology also uses the concept of co-operative binding heavily, which I will not discuss here, but which indicates another important area of further study in programming self assembly. The co-operative binding method is more similar to the work being pursued by Winfree.

This poses the question of individual component design: How does one build the required number of states and inter-component interactions into a single unit, and particularly, in the types of units we have described that assemble at liquid interfaces.

I have chosen to focus on limited state machines as every unit can conceivably be manufactured in very high numbers. This is more akin to sub-cellular assembly of macro-molecules than to multi-cellular assembly.

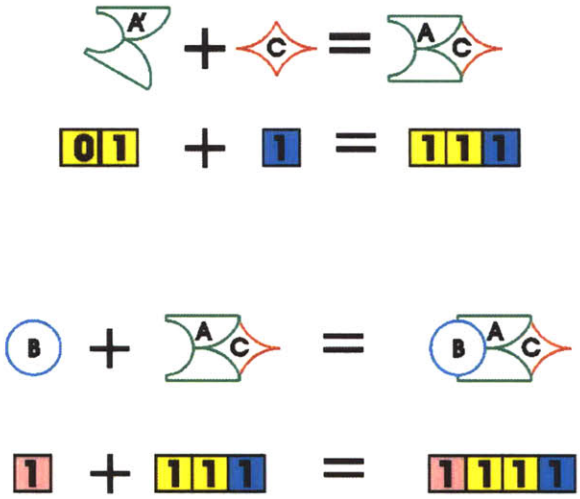


Figure 5-2 Schematic of allosteric, conformational state, implemented in meso-scale SA parts.

Figure 5-2 demonstrates the approach that I took and represents it as a state machine. The conformational shape change of part A to A' is the two states of component A. Figure 5-3 demonstrates how these components were manufactured by a double mold to PDMS process. Figure 5-4 demonstrates the assembly proceeding at the interface between water and perfluorodecalin (PFD). The allosteric part is the horse-shoe shaped red component that is 'opened', or activated when the catalyst (4-lobed blue part) binds it's functional site. The bound blue circular part can only fit into the horseshoe once the catalyst is bound.

[1] Ptashne, M., Gann, A., "Genes and Signals" Cold Spring Harbor Laboratory Press, Woodbury, NY, USA. 2001.

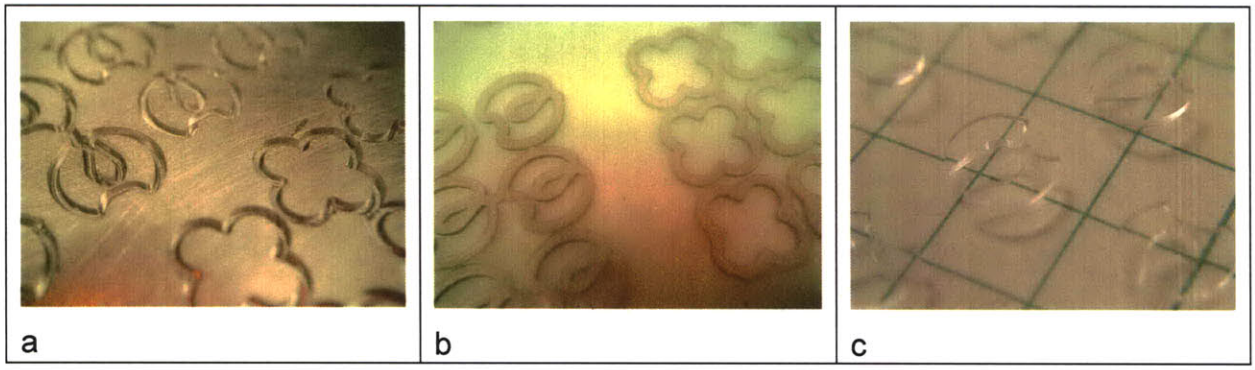


Figure 5-3 (a) Aluminum positive mold. (b) Polyurethane negative mold cast from (a). (c) PDMS SA components cast from (b) components were 4-8mm across.

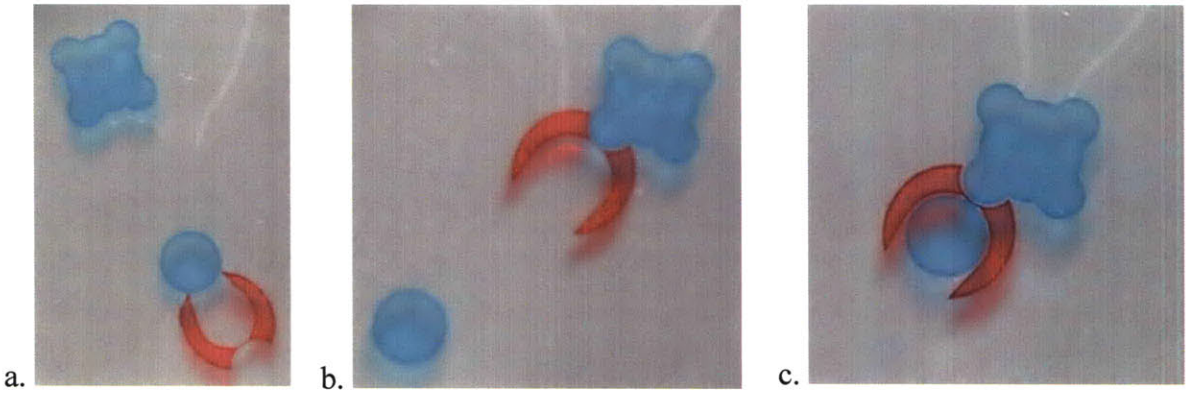


Figure 5-4 Conformational programmatic assembly. (a) neutral state, allosteric component (dyed red) is closed to circular component. (b) upon binding of activator component, the allosteric mouth 'opens' allowing the binding of the circular component (c).

I have quickly discovered that the extra complexity of designing state and flexible parts into the self assembling systems dramatically increases the complexity of their design. You will observe in Figure 5-4. that there are no straight edges on the components to avoid local energy minima on their collision. The essence of designing mechanisms for assembly in this system is designing an energy diagram of the different components' geometric alignment with respect to one-another, such that the desired states sit in the deepest possible local energy minima. I have successfully done this for the 2 state system above and made some progress on a 3 state system, however it is a problem of increasing difficulty, and as more component types are added to a system the challenge is to avoid any undesirable local energy minima.

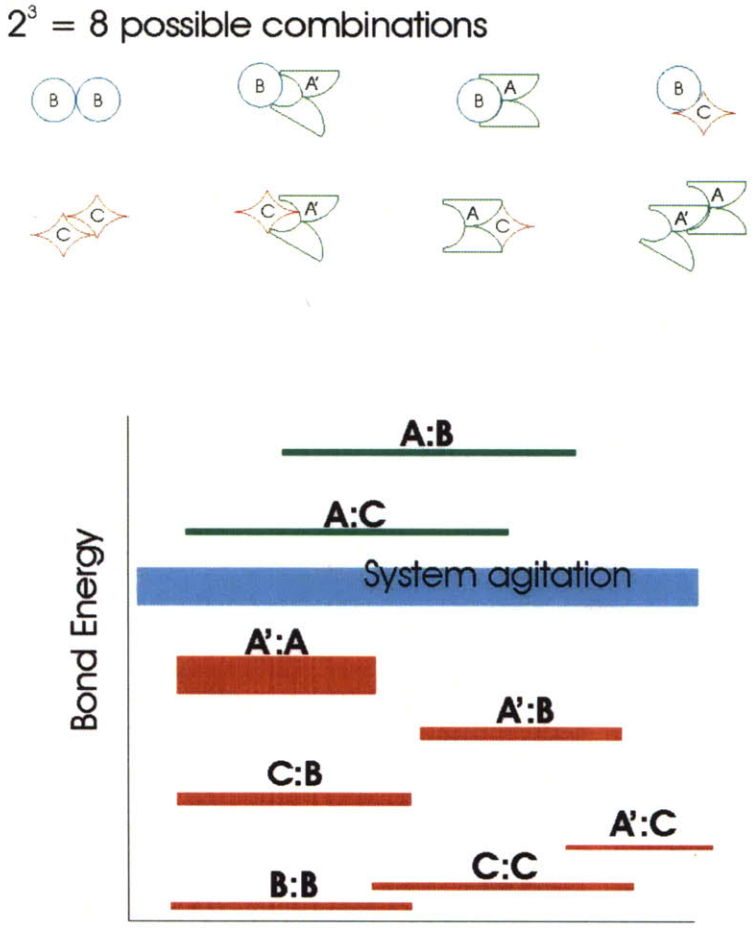


Figure 5-5 *System energy design and component interactions*

A cartoon of the energetics of the system and the challenge of system design is given in Figure 5-5. For the example of a 3 component system where one component has two states, A & A', and the other two components are single state, fixed parts (B & C) there are 8 possible combinations of parts. For each combination of parts in a 2D surface interaction there is a continuum of rotational orientations, with a subsequent bond energy relative to orientation curve. The key is to design the components such that desirable interactions A:B and A:C have stable conformations whilst the other bond energies are below the level of the system agitation energy (akin to KT in chemical systems). Figure 5-6 illustrates the design challenge – one is engineering not for the interactions they want (that is the easy part), but is engineering to avoid all possible failing interactions. This means considering all orientational collisions that might occur between each part within a SA system, and even more difficult to design for, is randomly oriented multiple component collisions illustrated in (c).

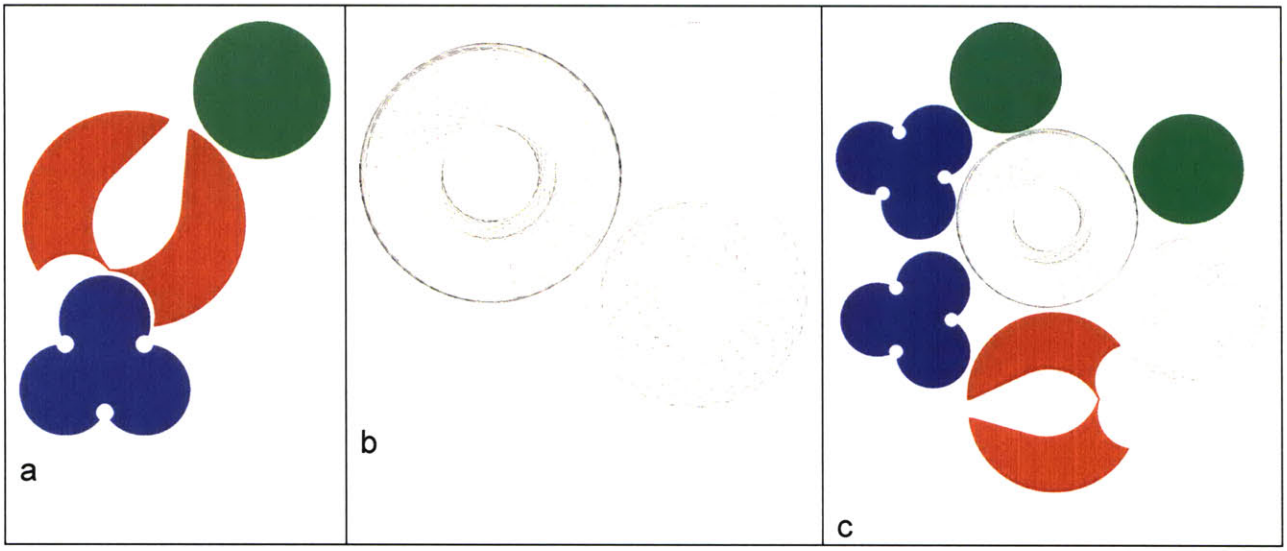


Figure 5-6. In designing for this type of SA system one must consider not only the desired interactions (a), but also all possible rotational combinations of them with eachother (b), and even more diffult, with a random number of many of themselves as illustrated in (c).

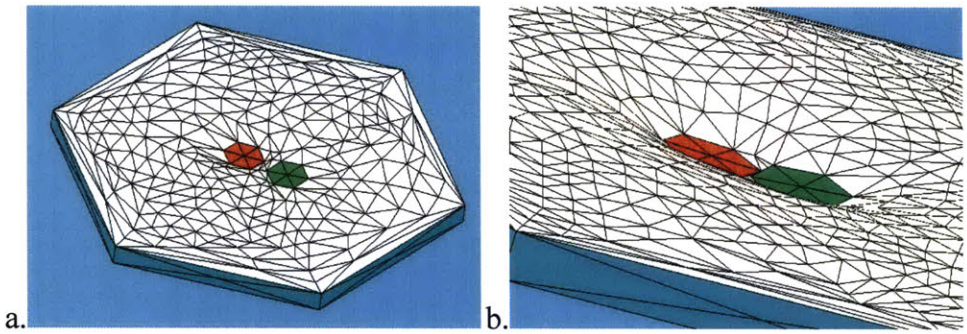


Figure 5-7. Surface Evolver was used to model component-component interactions and the resulting menisci by surface energy minimization. 2 hexagons floating on an hexagonal petri dish with mass acting at their center of gravity, with one wetting face (in contact with fluid) with a contact angle of 35 degrees. Hexagons are given an initial orientation, position, and tilt angle relative to the absolute so-ordinate system. Sides and top of hexagon are not shown.

There would be utility in having modeling tools that can facilitate this design process by automating the generation of the rotational energy diagram for any pair of components. From here it is possible to imagine using genetic algorithms or some other virtual evolutionary mechanism to evolve sets of parts with more optimal interactions and more complex interactions by performing a mechanical mutation on the part shapes at each generation – perhaps similar to the evolution of locomotive mechanisms by Sims[2]. To this effect I began by building a 'Surface Evolver' minimal energy surface

[2] Sims, K., "Locomotion of jointed figures over complex terrain" MIT Thesis, 1987.

modeler for multi-component systems. This work is described in more detail elsewhere - screenshots of the evolved minimal energy surfaces can be seen in Figure 5-7.

5.2 Kinematic Conformational State Machines

Figure 5-8. Two Kinematic state machines (Penrose) – the more impressive for he built these plywood models when the concept of state machines was far barely being developed.

The reduction to practice of any idea or concept is inherently a design task. Furnishing a design for a physical object implies meeting a number of constraints based on materials, manufacturing methods, geometry, and the environment for the final 'product'. Traditionally engineering design is focused on tasks where the machine being designed has a very specific set of interactions, with a specific environment, and randomness is avoided in the design stage. Inherent in the design task of building complex assemblies of randomly interacting parts, is randomness. Also inherent in this particular design task is the incorporation of logic into the mechanical elements. This is an engineering design task I have not previously encountered in the literature, and hence I present this case study as a blue print for engineers attempting to design such complex systems. It is an incomplete blueprint, no doubt, but hopefully the main challenges in this design task, and the differences between this and designing a specific machine, will be elucidated and there will be more work for following workers to build upon.

Shown in Figure 5-8 are two examples of plywood state-machines that Lionel Penrose (father of Sir Roger) was building in the late 50's to try and elucidate possible replicating methods for the recently discovered structure DNA. The first of these models is capable of replicating the Red / Blue bit string by templating with reversible bonds, using purely kinematic interactions. This process was hand assisted, not autonomous. Penrose completed his work at a time when state machines and the language to describe them were just being developed. It is not surprising then that his description of his mechanisms offers no easily discernible blueprint for the logical design. The challenge I had here was to try and reproduce these designs as simply as possible with the fewest number of states and

greatest robustness such that they would assemble autonomously. Figure 5-9 is a model of a reversible conformational switch by Saitou to assist in his work on a 'random bin picking' model of assembly.

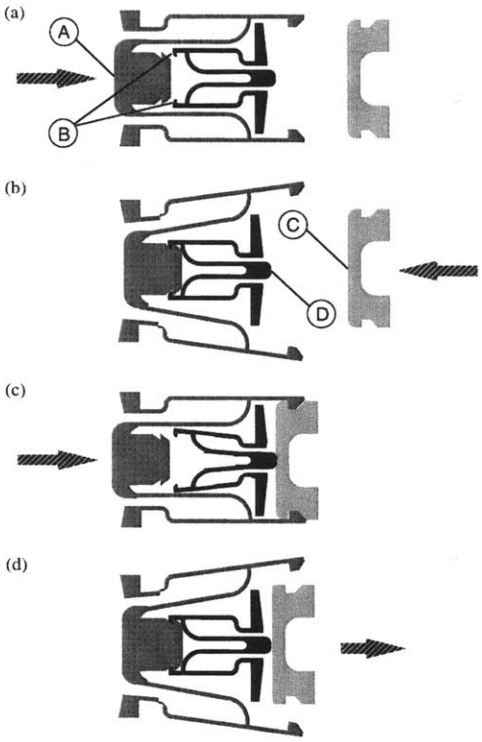


Figure 5-9 Reversible onformational clamps designed by Saitou.

5.3 Case Study: Design of a kinematic self replicating mechanism on an air bearing surface.

I am interested in the question of what a minimal self-replicating system looks like. This interest in the minimal logic required to exhibit self-replicating behaviour stems from an assertion that if the logical overhead is low enough self-replication has potential in a manufacturing sense.

There seem two ready approaches to basic replication (and undoubtedly more), one where the individual units of the assembly are state machines capable of selecting and copying similar units in a similar assembly. The biological analogy can be thought of as an RNAse-less RNA replication where the logical function of the RNAse selecting and adding base pairs is performed by state machines within each base. The second approach is more similar to biological systems in that the logical machinery is the RNAse which assembles more basic units into a second RNAse and so forth. We might describe these as distributed logic replication and centralized logic replication. In the first the logic requirements are performed by CA type nearest neighbour interactions between the units of assembly. In the latter, the copying assembly (or RNAse equivalent) is the central processor reading

and copying the data tape. I shall be pursuing the distributed logic replication model schematically outlined in Figure 5-10.

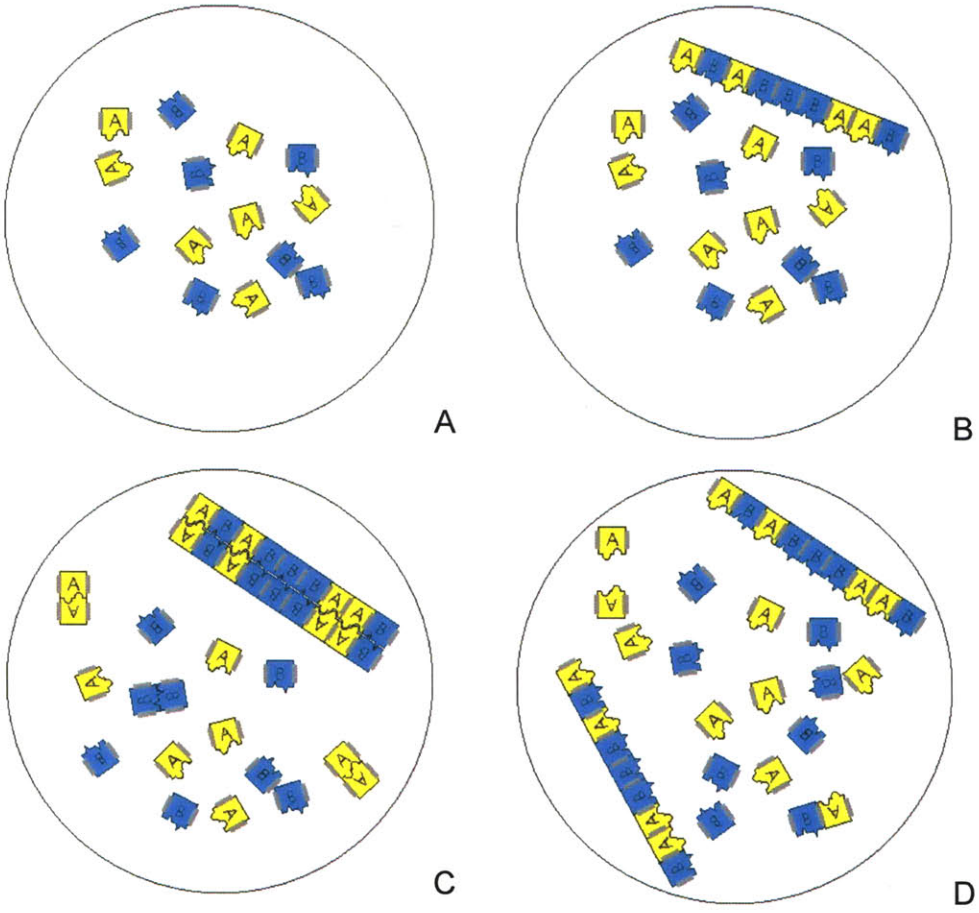


Figure 5-10 Distributed logic replication model by templated copying of an input 'string' built of similar components to the input string.

An important observation in this work is that of the memory requirements for replication. If each part of an assembly must keep the entire code for the assembly, then each part must have a memory capacity capable of that description. Each and every cell in a biological entity contains the entire code for the assembly, but this is a coarser scale of assembly in the multi-tiered assembly design of biological entities. At the sub-cellular level a very interesting memory management scheme occurs where the memory is contained within the structure itself. Ie. The code within the base pairs of DNA.

This recognition, or localized copying / complimentarity method can similarly be used in non-biological systems. In fact we can imagine using it in either 1D (linear) systems or in 2D (planar) systems. A 1D system will look very similar to DNA except that complimentarity is not necessary and we can actually template an exact copy, not a compliment. A 4 base system is not strictly necessary, and in fact you will see a 2 base system is sufficient (binary). A 2D, or planar, system would look like a

programmed or templated 'plating' type system. The difficulty in a plating replication is in the termination signal, where the copy is to separate from the original to make way for a subsequent round of replication. A 3D system is not amenable to this replication treatment as not all of the structure is available at the surface of the assembly and therefore some of the information within the structure would need to be transferred (and stored in memory) through outer layers of the assembly. It may be stating the obvious, but this is probably why biological replicating systems use linear assemblies for replication that can subsequently fold into 3D structures (or unfold for replication).

So with the above considerations in mind, how do we design the minimal logical units for a replicating system capable of replication of an arbitrarily complex structure? Ie a structure that contains an arbitrarily complex string of assembly instructions?.

We will need two types of components that represent the data string. They can be similar logically yet different in some key way (eg. shape selectivity) to enable subsequent processing (addressed later in this piece).

With such low costs for digital logic why concern ourselves with minimal state machines for replication?

Concerted engineering efforts over the last 50 years have reduced the cost of silicon based digital logic to extremely low levels. Basic microcontrollers with a lot of memory (by replication standards) are available for under $1 each and if packaging costs didn't dominate as the size is shrunk they could lower in cost. However even at any finite cost of pennies, or fractions of pennies, objects comprising 10^6 components or more start to have significant cost. By reducing the complexity of the sub-units to an absolute minimum one can reconsider logic implemented by mechanical, chemical, or similar means. Babbage's revenge!. At the level of just 5-6 states per sub-unit one can start to imagine mechanical or chemical finite state machines that make up the micro (or potentially nano-) scale sub-units for replicating more complex machines.

The nascent field of self-assembly has progressed quickly to demonstrate assemblies of very large numbers of components, at nm, um, and mm scales in 1D, 2D, and 3D. To date these components have been strictly passive, the only logic they employ is selectivity of the edges 'programmed' with magnetics, electrostatics, or surface chemistry (hydrophobic / phyllic). With minimal logical requirements we can imagine developing the basic units of self-assembly with executable logic built in. Indeed this is a requirement to make self-assembly that is programmably asymmetric.

Outlining a minimal logic structure capable of self-replication will give other researchers a blueprint for designing self replicating systems in other biological or non-biological 'environments'.

1.1.1 Design 1: Logic design and LEGO

Figure 5-11 is a cellular automata representation of the logical requirement for a linear templating replication. This diagram shows the physical process of the replication. Each individual sub-unit or 'cell' is represented as a box. A 1 represents an edge in an attractive state (attracts and binds random untethered cells). A 0 represents a neutral edge state where no binding will occur. A line between two cells represents a tethered bond (note that these occur at the junction between any two edges in the 1, attractive state eg 1-1). A free floating unit cell is always in the (0,0,0,0) state. A 1 edge will attract a 0 edge but not bind it. A 1 edge will bind to another 1 edge. 0 edges neither attract nor bond. The steps in the replication are numbered from 1- 10. Note that this is replicating an original input string of 5 cells (in the upper row), but in fact scales to strings of arbitrary length.

This mathematical representation uses (or perhaps misuses!) the concept of cellular automata. The wire is first filled with the initial values of the cells. The wire has 4 rows: the first two rows represent the cells of the original programmed string, and the third and fourth rows the replicating string. The orientation of the states of the cells is different for the programmed string and the replication string. This is mainly because we want to arrange communication intensive edges of neighboring cells close to each other in order to keep the bits of each rule to a minimum. We will see later that this is convenient when actually mechanically implementing this design where the geometry of the cell tiling is important in minimizing the number of states (and components). Figure 5-12 shows the actual orientation of the states of each cell where N is the north edge, S the south edge, and so on.

Notice that the division at the end of the replicating steps can be shown easily by simply dividing the wire horizontally in the middle: the top two rows will be the original programmed string; the bottom two will be the copied string. The quiescent space outside the wire is filled with 1's. The updating of the wire itself is shown in Figure 5-13. Each update is of one cycle or binding event, starting from the initial state of an unbound input string.

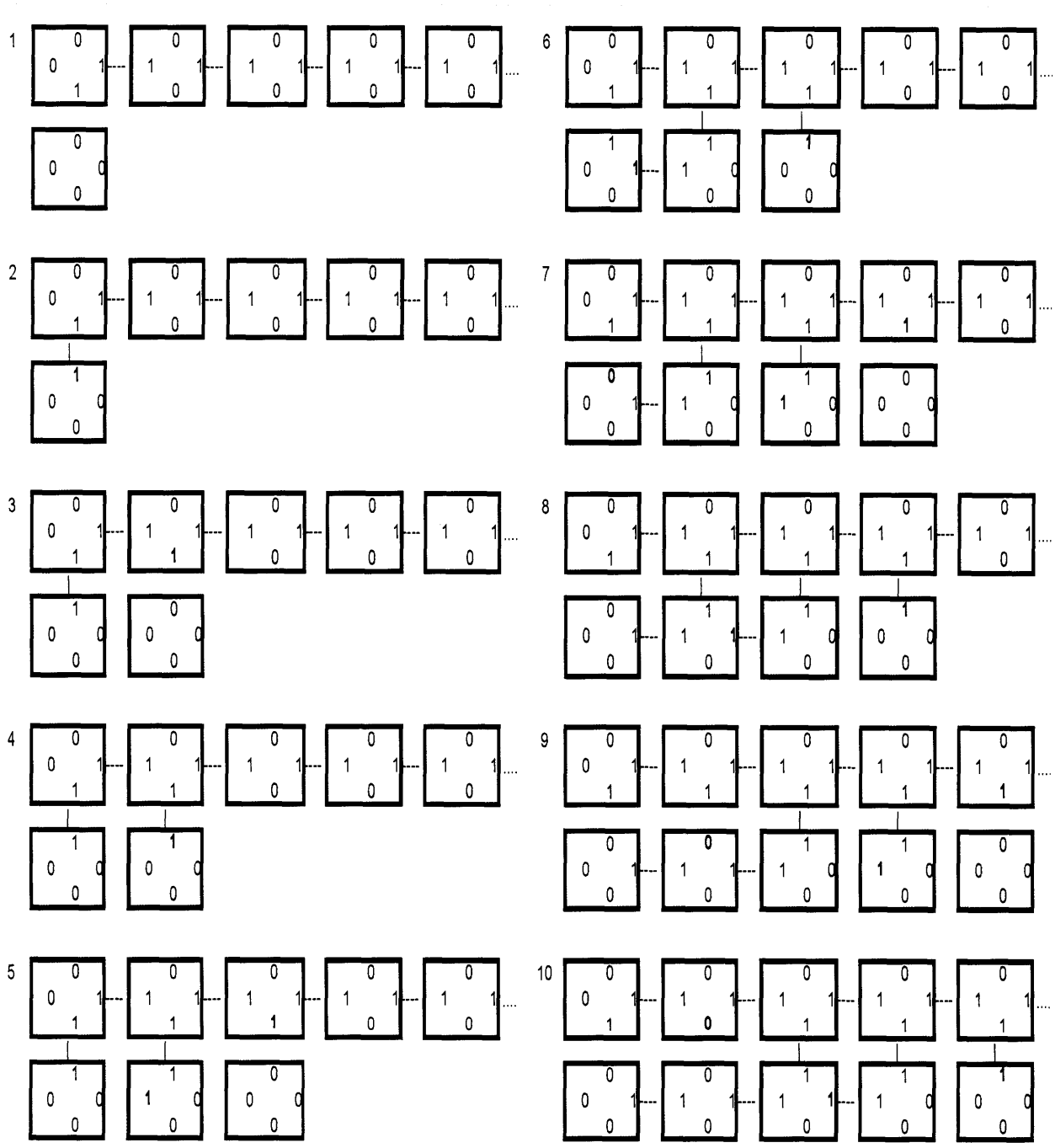


Figure 5-11 Cellular automaton representation of self-replicating state machine.

Programmed-->	N	E	N	E	N	E	N	E	N	E	
	S	W	S	W	S	W	S	W	S	W	...
Replicate-->	W	N	W	N	W	N	W	N	W	N	...
	S	E	S	E	S	E	S	E	S	E	

Figure 5-12 Orientation of states in CA representation of self replication.

Figure 5-14 is the effective rule table for the automaton of Figure 5-15. Note that cycle refers to each numbered image in the previous diagram. There can be multiple logical steps at each cycle. In this rule table the central value (second row, third column) is the one that's changing in relation to its 9 neighbors (8 nearest neighbours and 1, 2-distance neighbour to the left.). This central value changes to the value to the right of the arrow.

Somewhat differently to how we typically conceive of a cellular automaton as a clocked updating of all cells at a regular interval, this is a self clocking (or asynchronous) cellular automaton. Physically this means that in a replicating system such as the original input string floating in a bath of loose unbound cells, the clocking and logical operations occur at the binding of a cell to the replicating string. It is important to note that the copying proceeds from left to right and that at any given moment in the replication the replicating string is tethered to the input string by just a single bond at the site of most recent cell addition.

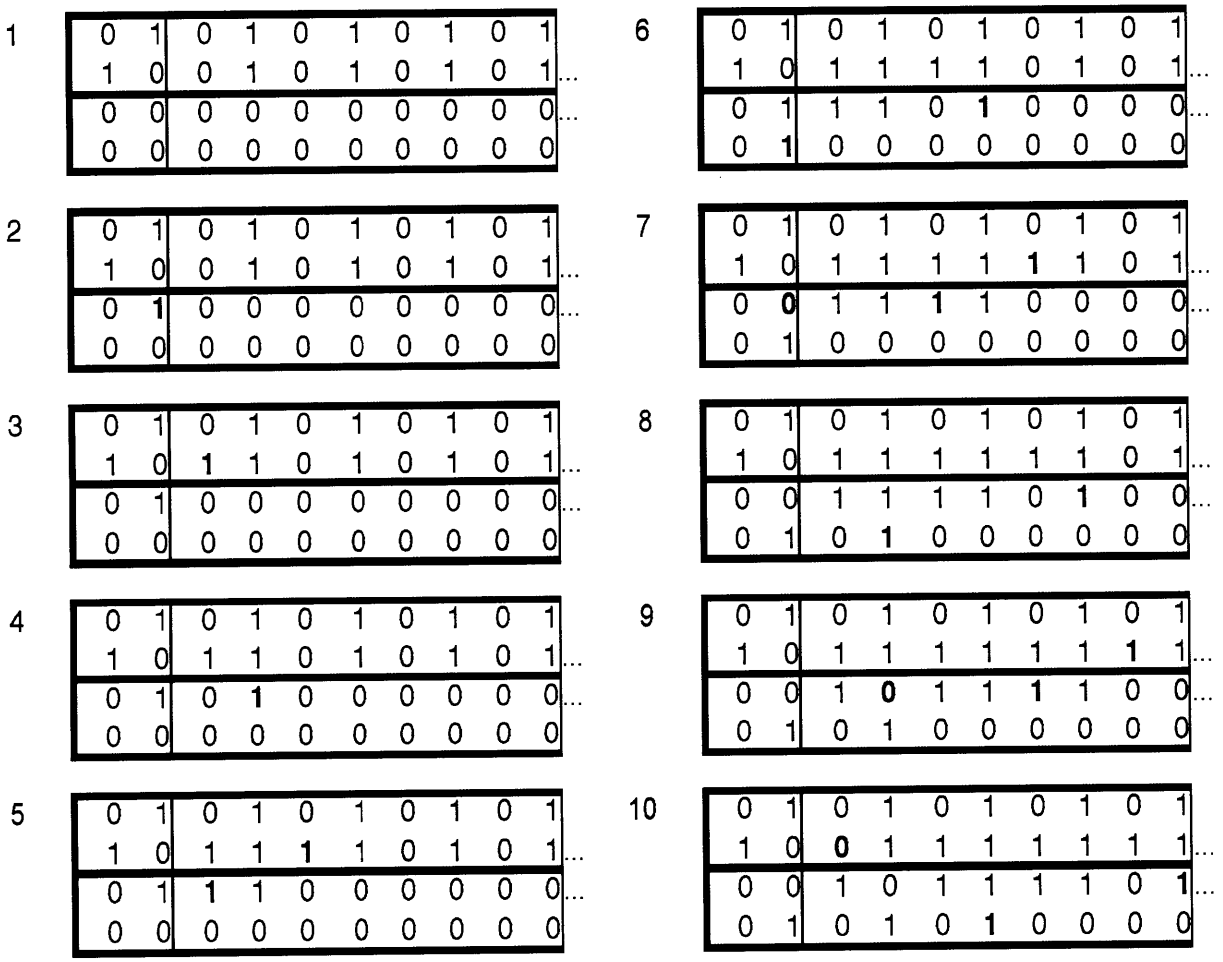


Figure 5-13 Time (or binding steps) in CA representation of self replicating state machine.

Note that in cycle 8 the rules start to repeat; this is due to the fact that the replication process can be arbitrarily long, but the general algorithm for updating the wire remains the same.

Cycles	Steps	Rules
1	1	1 0 0 / 1 0 0 0 ---> 1 / 0 0 0
2	2	1 0 1 / 1 0 0 1 ---> 1 / 1 0 0
3	3	1 1 0 / 1 0 0 0 ---> 1 / 0 0 0
4	4	0 1 1 / 0 1 0 1 ---> 1 / 0 0 0
	5	1 0 1 / 1 1 0 1 ---> 1 / 1 0 0
5	6	0 1 1 / 1 0 0 0 ---> 1 / 1 1 1
	7	Same as 3
6	8	1 0 1 / 1 0 1 1 ---> 0 / 0 1 0
	9	1 1 1 / 1 1 0 1 ---> 1 / 0 0 0
	10	Same as 5
7	11	1 1 1 / 1 0 0 0 ---> 1 / 1 1 1
	12	Same as 3
8	13	1 1 1 / 0 1 1 1 ---> 0 / 0 1 0
	14	Same as 9
	15	Same as 5
9	16	Same as 8
	17	Same as 11
	18	Same as 3
10	19	Same as 13
	20	Same as 9

Figure 5-14 Automaton rule table.

It is somewhat difficult to interpret what these rules mean physically by merely looking at the tables above. Let us look at a mechanism that can execute this logic. One of the first choices in the design of a mechanism was the tiling scheme. The main constraint is that any logical operation should not be passed through another tile as that would require vias, memory,or communications, or extra logic within that cell. As the logic designed above requires communication with 4 neighbours a tiling design that enables 4 contact face neighbours was desirable. Furthermore 4 neighbour contacts that are all in one axis was desirable such that a kinematic collision (billiard ball type) can supply the energy for all logical operations.

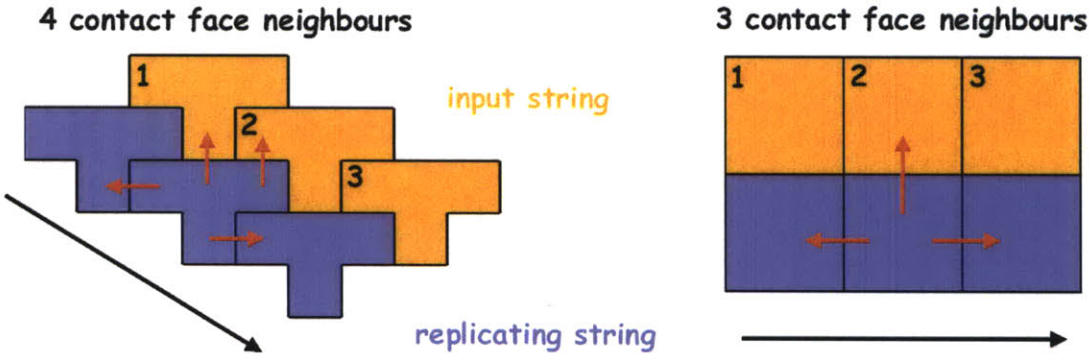


Figure 5-15 Four nearest neighbour faces with contact faces is desirable leading to the T shaped tiles at left which also allow all communicating faces to contact on same axis. This is in contrast to 3 neighbour tiling of a traditional CA model.

The T shaped tiles of Figure 5-15 achieve both goals of 4 contact face neighbours and all of those contacts can be executed on faces lying in one axis. Figure 5-16 shows the logical routing. Arrow heads represent the opening of a gate for binding. The origin of the line with the arrowhead represents the input connection that forces this change of state. The input of a connection can be thought of as routing a switch from a 0 to a 1 state (open for binding) at the arrowhead.

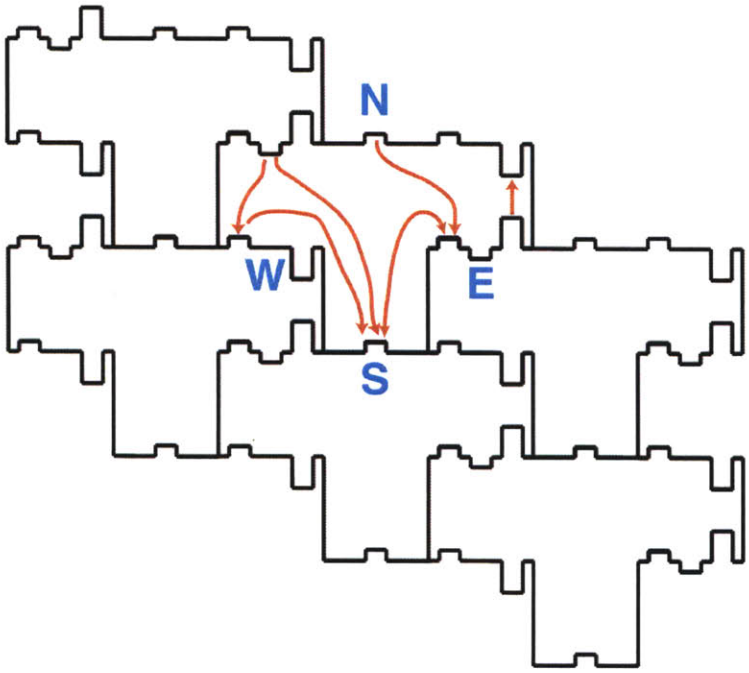


Figure 5-16 Logical routing in T-shaped tiles. Protrusions represent input and output routings of logical connections.

The first iteration of this design was implemented in that ever so useful rapid prototyping tool: LEGO. Custom parts were laser cut in acrylic. The scale is roughly 3x4 inches for the whole cell. The

images in Figure 5-17 demonstrate the replication of a dimer. The blue and pink units have exactly the same logic as represented above. The only difference is a mechanical shape selectivity that prevents pink binding to blue and vise versa. One can now see how a bit string of data can be replicated by such a system. Any arbitrary input string of pink and blue tiles can be replicated in the lower string. One should note that at the completion of the final cell in the replication the two joined strings automatically divide into two complete strings ready for the next round of replication.

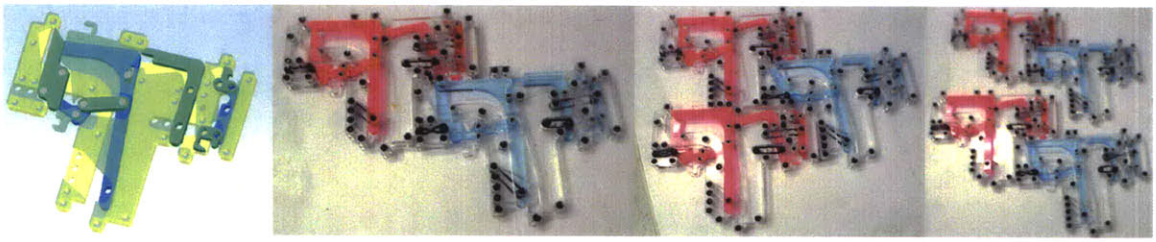


Figure 5-17 LEGO / Acrylic implementation (in two layers of laser-cut acrylic) and di-mer replication. Shape differentiation of pink and blue cells is present.

When operating the above mechanism, only 6 discrete states are used. These are enumerated in the state diagram of Figure 5-18. Similar to the plywood machines of Penrose these units only replicate when collisions are directed by hand.

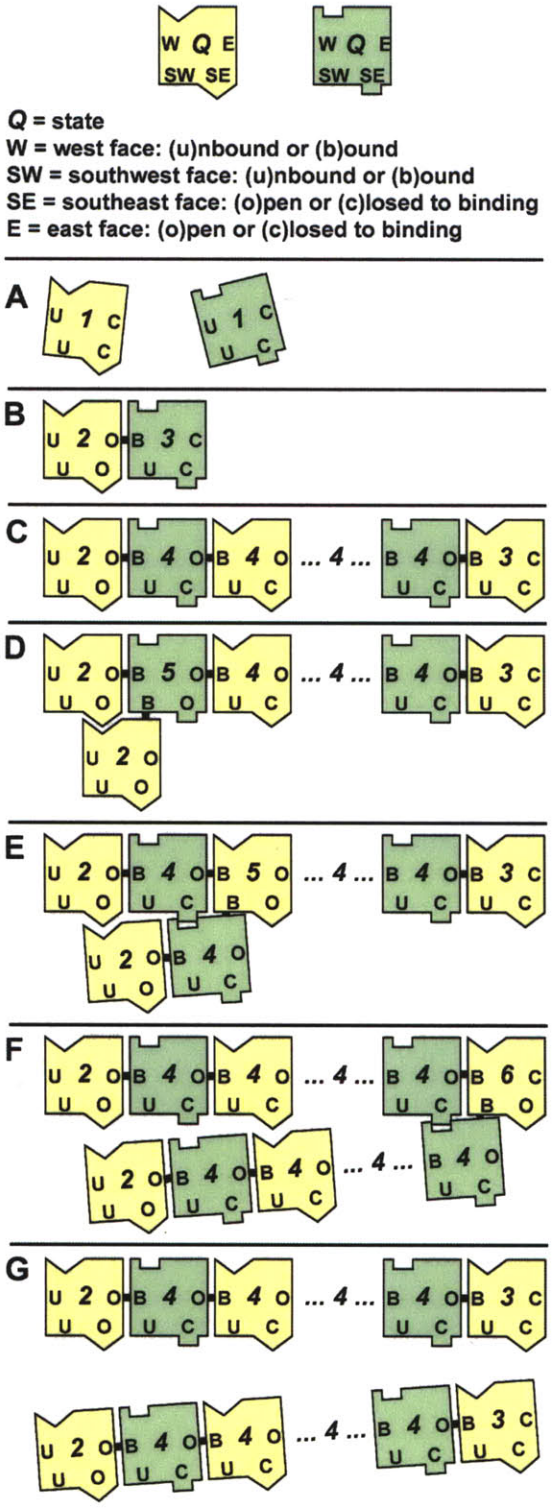
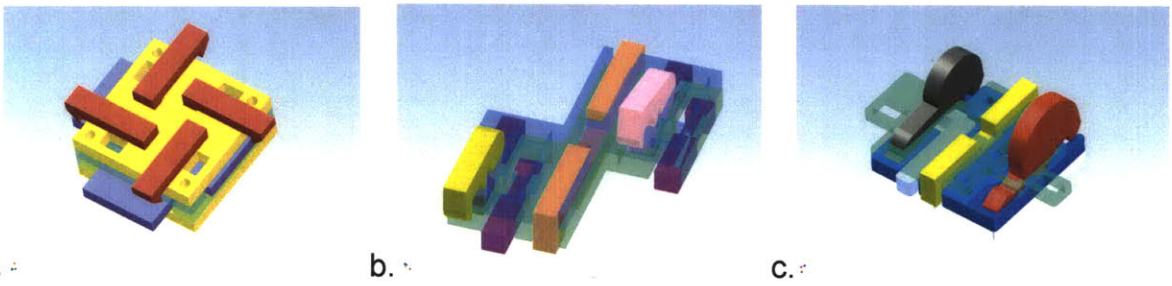


Figure 5-18 6 discrete states in an enumerated state diagram of the self-replicating algorithm.

5.4 *Autonomous self replication on a 2D air bearing.*



a. b. c.

Figure 5-19 3 Iterations in the design of autonomous self-replicating mechanisms.

The 1st unit iteration described in the previous section is not autonomous, large, complicated, and difficult to manufacture. I will briefly describe here further design efforts to reduce the complexity of the component to something conceivably mass implemented. This is a similarly difficult process of iterative design evolution to that described in section 5.1. Three early iterations are described in Figure 5-19. These used pivoted latches, cams, and slides, with gravity as the return force and kinetic interactions by collision with other components in 2D the driving force for logic execution. It was quickly determined that the tribology of laser cutting in acrylic was too inaccurate for reliable operation of the components and cams would regularly jam and require human intervention to reset. The convenience of laser-cutting acrylic was useful in fast iteration of the design cycle so flexure based levers were chosen as a conformation latch where friction and tribology were no longer constraints and that the logical operation was now purely a matter of the geometry of the flexures, and the driving forces required to execute the logical operations could be tailored by changing the geometry of the flexures. After many iterations to balance other constraints such as surface area and mass that were floatable on the air-bearing assembly environment (see chapter 6 for details) the design shown in Figure 5-20 was settled upon for testing. The 6 components at left (4, red, flexure based latches and 2, blue, cam-push arms) press fit into the bases at right which are identical save for a small shape differentiation which defines the two bits. The lowermost flexure arm is the reversible latch which is unlocked by a collision with the flexure arm and pushrod in the lower centre of the bases. Figure 5-21 illustrates the reason for the strange base tiling. The T shaped tiling of the previous section allowed for rotational alignment errors deleterious to the assembly. The tiling of (b) kinematically aligns and constrains the tiling which is still plane filliing, and allows all logic levers and latches to be actuated along one axis. The assembled units can be seen in Figure 5-22.

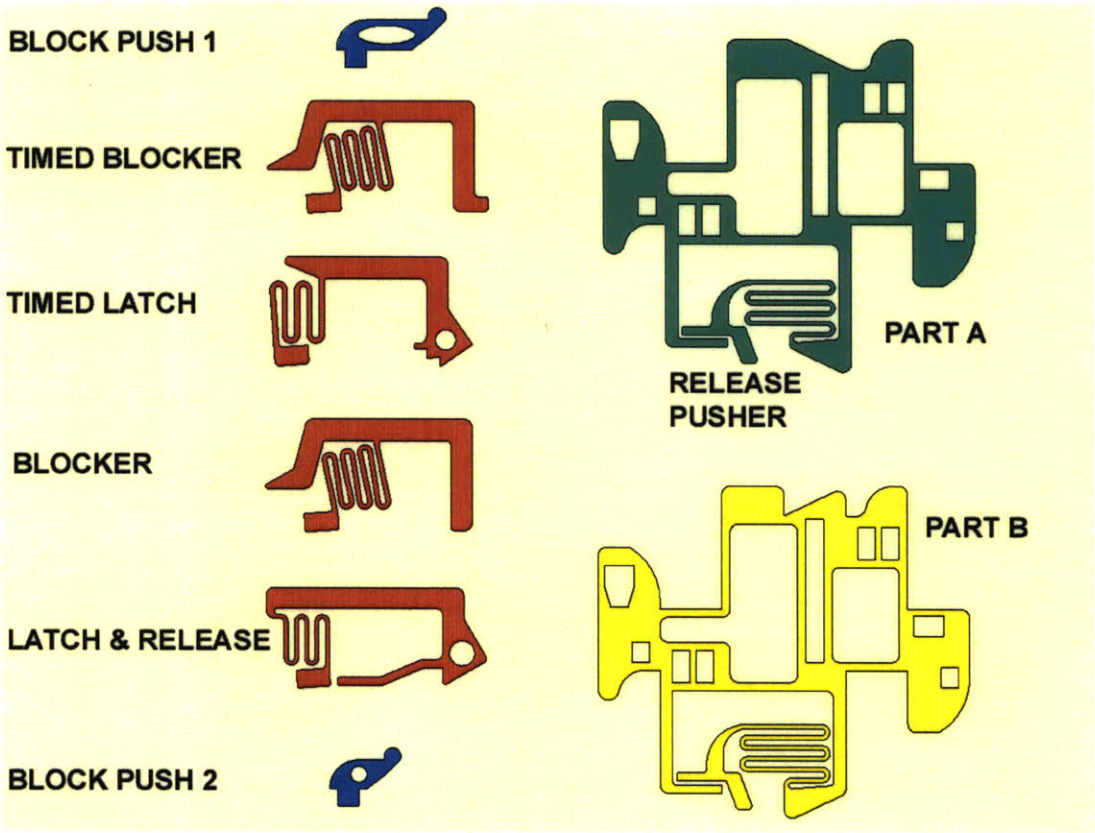


Figure 5-20 Flexure based, Conformational latching, 6 state, self replicating mechanisms. The 6 components at left press fit into the two bases at right which are shape selective but otherwise identical.

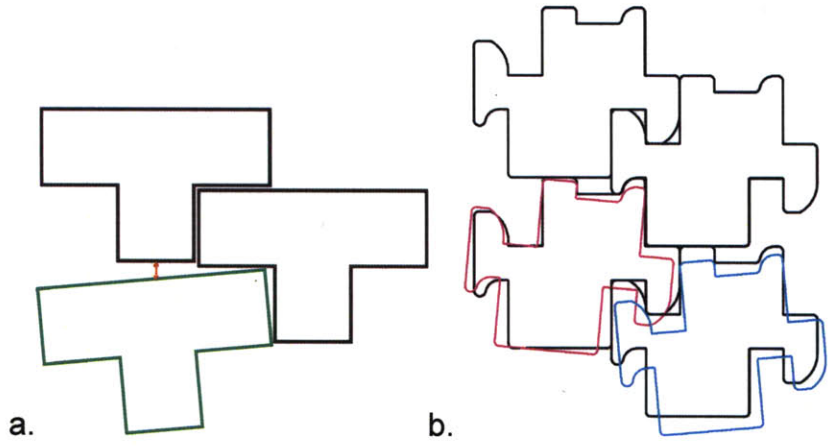


Figure 5-21 (a) The tiling of the T shaped tile from Figure 5-17 and the consequent difficulties with alignment errors. (b) The tiling chosen to alleviate tiling errors by kinematic constraint.

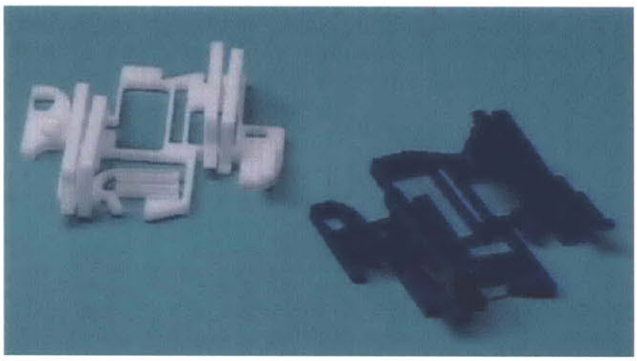


Figure 5-22 Acrylic replicating units assembled by snap fit.

6 of each of the components in Figure 5-22 were constructed to run autonomous replication tests. The experimental setup was similar to that described in chapter 6.

The same experimental setup described in chapter 6 was used to attempt autonomous replication. Screenshots from the experiment were taken at 15 minute intervals throughout the 10 hours the experiment was run. A selection can be see in Figure 5-23.

Two di-mers were preset manually (one of each orientation B:W, and W:B) and introduced to the table as seen in (a). This was the initial condition of the experiment. Somewhere between 15 and 30 minutes into the experiment the W:B di-mer broke apart. At 3hrs 15 mins. A blue unit appears to bind to the di-mer (c). At 4hrs 30mins there are two W:B di-mers indicating successful autonomous replication. Alternatively it could indicate that the 3 mer of (c) spontaneously broke up and another dimmer spontaneously formed by a high energy collision or low tolerance flexure in the unit. At 5 hours once again it appears that a di-mer has broken apart leaving only 1 remaining di-mer (e). At 8hrs 15mins another trimer appears. This is how the experiment was found running when it was terminated at 10 hours.

Was this successful autonomous replication? Perhaps. If so it does indicate an interesting aspect of these systems – that one only need exceed the failure rate with the replication rate to have net gain. The kinetics of this assembly are slow due to the highly orientation specific requirement on the collisions dictated by the geometry of the tiles. There is much to learn in the design of such systems.

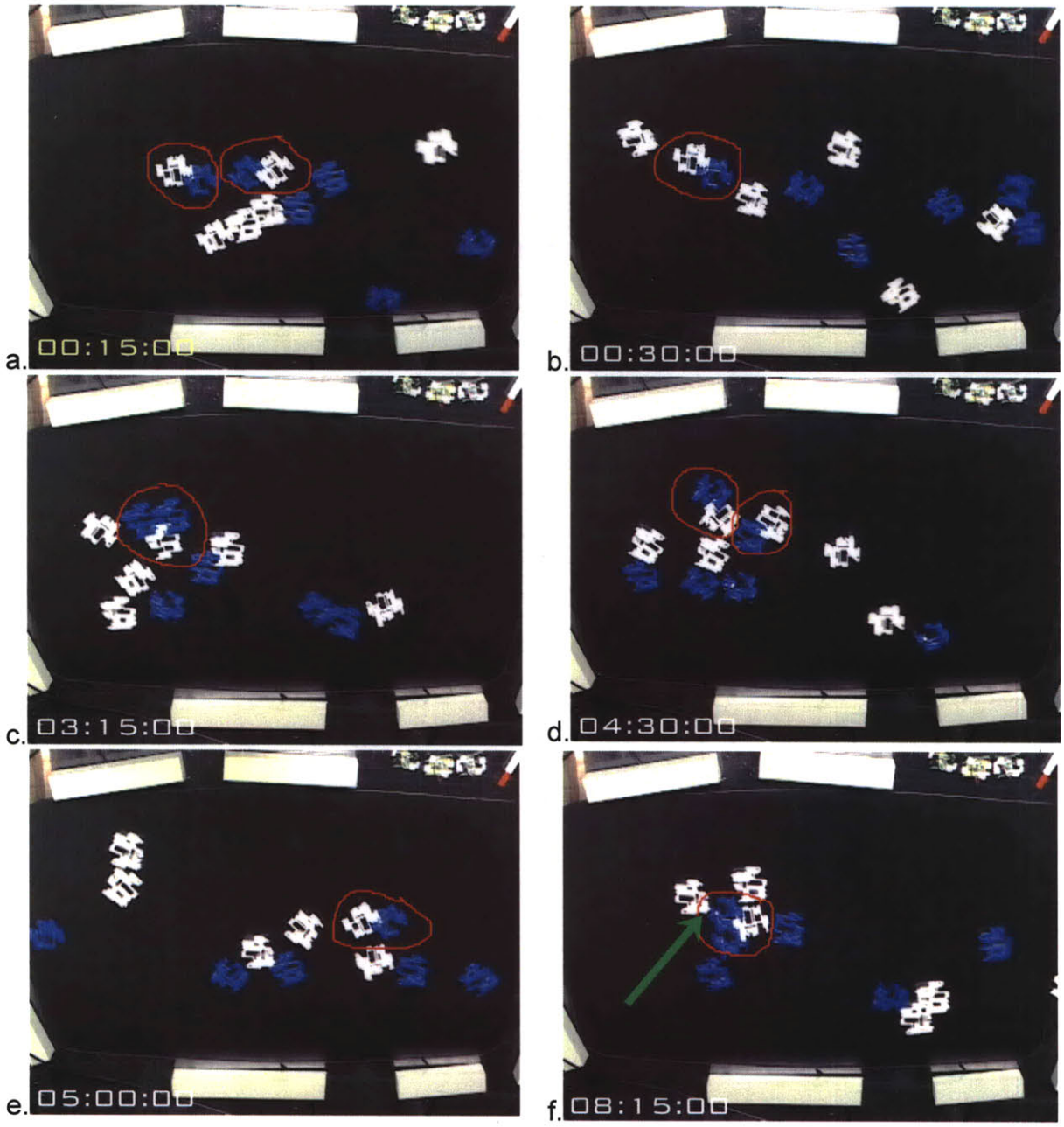


Figure 5-23 Screen shots from self-replication experiment.

6 Electromechanical Assemblers

As might have been suggested by the difficulty of the work in Chapter 5 designing a single example of a kinematic state machine for programmable assembly, a more general tool for studying the programs & the kinetics was desired. This was deemed sensible rather than designing a specific mechanical implementation for any particular state machine or algorithm that was desirable to test.. To that goal reprogrammable electromechanical 'units' were designed as a physical 'simulator' or 'emulator' for studying the properties of programmably assembling systems. It will be seen that these units also suggest the utility of reprogrammable assembly. I mention here with gratitude the assistance of Dan Goldwater with the development of the electronics in these devices[1].

6.1 Design

Of most interest to me is the study of self-assembling (or programmably assembling) systems where the amount of logic within each component is minimal, in fact so small that the logical operations can be implemented without digital logic. Without this constraint I do not believe these types of components will ever scale down to nm and um components, nor will they be cheap enough to be implemented in sufficiently large number to be interesting. This dictated many of the design choices in building these particular components such that they would emulate similar systems. For example only nearest neighbour communications were implemented to emulate systems where communication between parts can only be achieved by contact, as in for example a kinematic system such as that discussed in chapter 5.

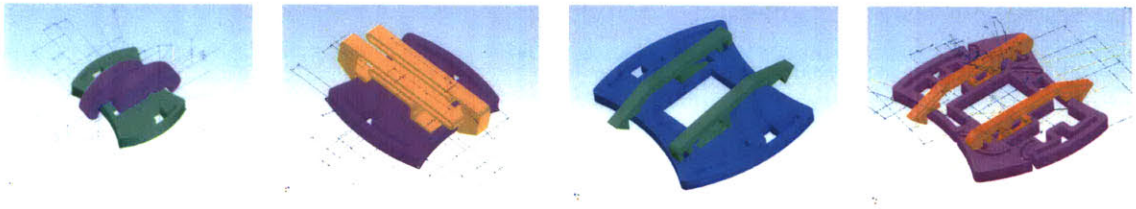


Figure 6-1Early prototypes of magnetically actuated reversible latches.

A tiling had to be chosen for the base of these units. An offset tiling such as that used in the parts in chapter 5 would have been desirable for emulating such a self replicating system, but limiting in studying or building other types of assemblies. Rather a basic square lattice was chosen for it's generality and with the redundant computational cycles of the Electromechanical Assemblers (EMA's) the communication neighbourhood can be increased artificially by message passing. The tiling sits

[1] Dan, as well as being a fabulous friend, knows his way around circuit design, surface mount stuffing, communications protocols, and microprocessor programming. I owe him a debt of gratitude for his assistance in the design, debugging and operation of these units. It had its exciting moments, I hope that can be part of the reward.

on a plane square lattice, but individual tiles are radiused on all edges, 2 convex, 2 concave, as this was shown in early mechanical experiments to increase the likelihood of successful binding upon collision.

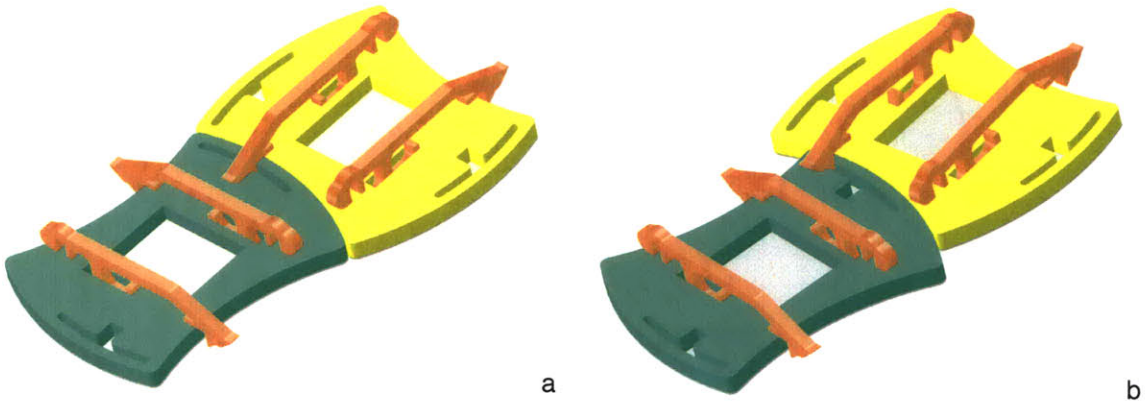


Figure 6-2 a) Two components correctly aligned. The latch has seated in the deeper extension of the latching channel. b) A 'metastable' bond between two components. To increase the chances of successfully binding collisions the latching channel allows a temporary bond that upon further agitation allows the latch arm to sit into the deeper latching position.

To further increase the number of successful binding interactions a 'metastable' binding channel was employed as can be seen in Figure 6-2. Latches can catch in this channel giving an effective approach angle of around 30 degrees. Once caught in the channel the parts are free to settle into a single successful bound position whereupon the communications magnets align. This position can be seen as the deeper cut section in the centre of the binding channel. Figure 6-3 shows the interlocking of 4 successfully bound components. The components in Figure 6-3 are the final design used throughout the experiments in this work.

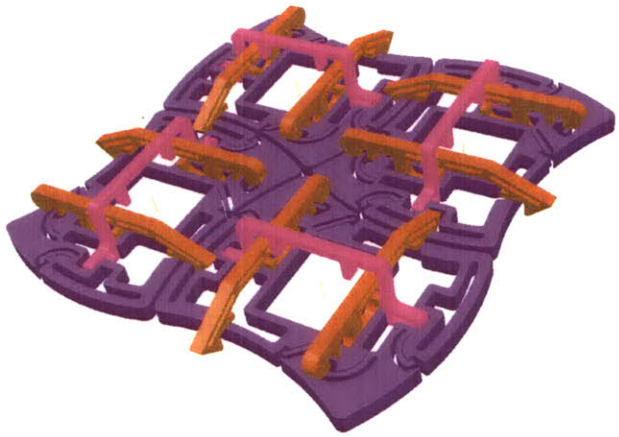


Figure 6-3 Four successfully bound units demonstrating the square lattice that the parts conform to.

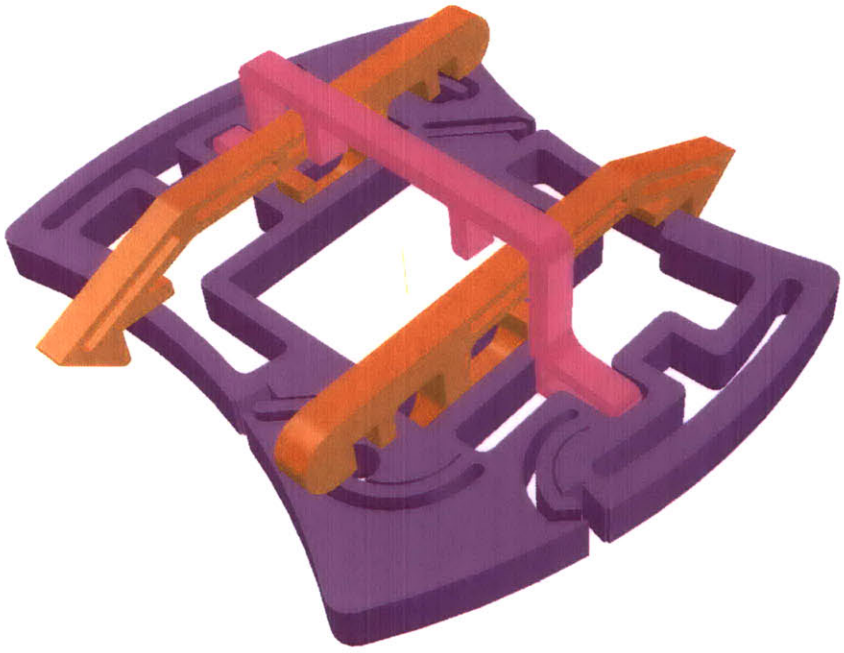


Figure 6-4 Detail of unit base, two latching arms, and a roll cage to prevent arm loss upon heavy collision.

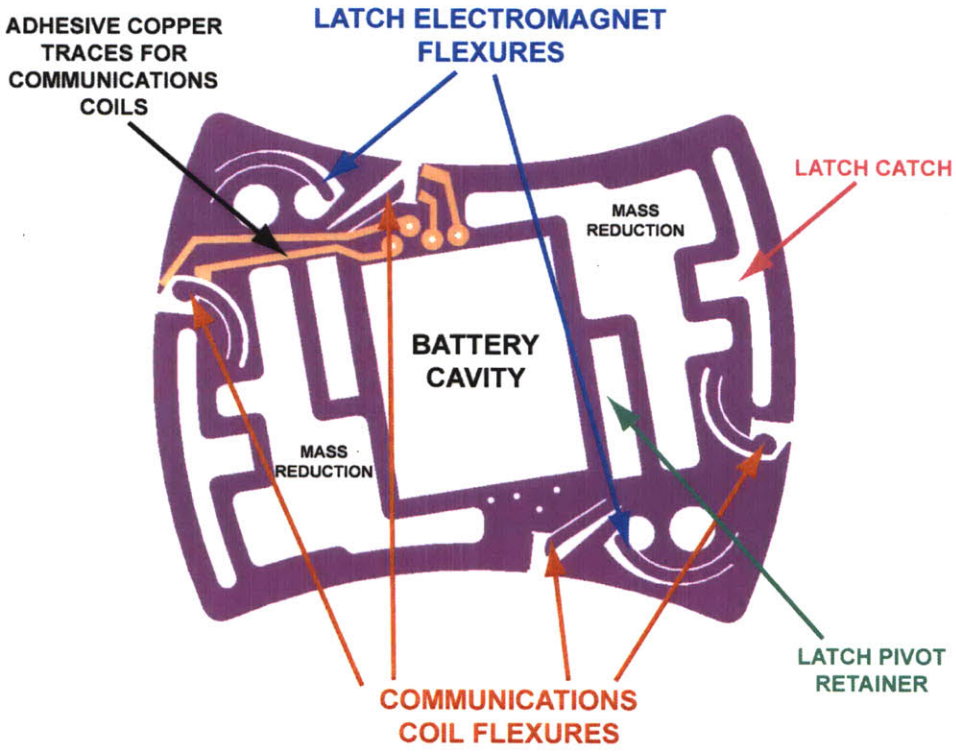


Figure 6-5 Projected Detail of unit base.

Figure 6-4 details the acrylic components of each unit, 4 in total. The latches sit in recesses labeled latch pivot retainer in Figure 6-5. The center of mass is on the hook end of the latch pivot such that by default it sits down. The roll cage, or retainer, sits over the two latches to prevent latches from falling out upon heavy collision. Design was done with assembly in mind such that all parts were designed for press fit or seating by gravity without bearings or axles. Figure 6-5 clearly illustrates the flexures which were used for alignment and retainment of both the communication coils and the latching electromagnets. With magnets installed the flexures acted as retaining springs. The battery cavity is also illustrated and was a snug press fit. The circuit board sat atop the battery held in place with double sided tape. The rechargeable lithium batteries with sufficient energy capacity dictated much of the size of the units as nothing smaller than 20x25mm batteries had enough capacity to run the units for the estimated 4 hours required for larger/ more complex assemblies.

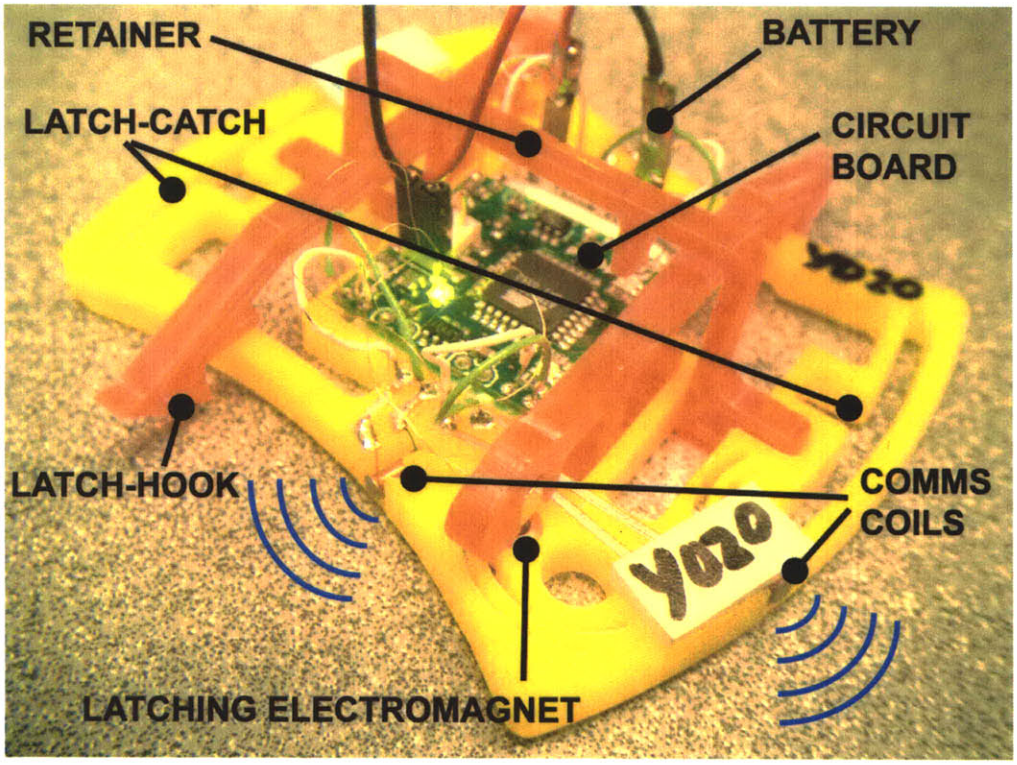


Figure 6-6 Detail of fully assembled unit.

Self adhesive copper traces were cut on a vinyl cutter and adhered between the communications magnets and the circuit board. Wherever possible mass was removed from the units by cutting holes and recesses as can be seen in Figure 6-5 & Figure 6-6. The circuit board was designed at 20x20mm to sit atop the battery. Wires for traces and electromagnets were routed as short as possible to prevent excess wire from being snagged during the random collisions of the assembly environment. Also shown in Figure 6-6 is the unit numbering to aid in debugging, and the

removeable battery connector to allow for battery recharging and programming – in retrospect I would have designed for IR or other wireless programming to save the laborious and time consuming process of re-programming each by hand and for a less fragile recharging method – units were often damaged when removing the battery connector.

6.2 Latching

A reversible latching mechanism was required to implement the logic. This latching had to be designed with minimal power consumption in mind. Purely electromagnetic latching was experimentally determined not to be robust enough not to break apart upon random collisions. An electromagnetically activated physical latch was therefore used. A permanent magnet (NdFeB) cantilevered in the latch is suspended over an electromagnet in the base. Current applied to the electromagnet pulls the permanent magnet down, lifting the hooked end of the latch on the opposite side of the fulcrum of the latch. In this manner the default position for all latches is down, but ready to lock, and only upon application of current may components disconnect. The advantage of this is that for virtually all algorithms this is the most energy efficient – the electromagnet is by far the largest current draw of each device. The disadvantage is that because the default state is locking, in the event of a unit failing (power loss, logic loss etc) it cannot release from surrounding components – it is not a failsafe design, but was consciously chosen as such as power was the most limiting factor at this scale.

6.3 Specifications:

Figure 6-7 Mutliple units ready for experiment. Each is individually numbered.

Electromechanical Autonomous Assemblers: Components.

logic	• Atmel Mega8 microcontroller: 8MHz operation • 10kHz 10bit A/D • 8K program storage + 1K RAM
battery	3.7V, 150mAh Lithium-Polymer rechargeable: 4.2gm
Electronics	A few LED's, FET's and discretes
Latch Actuation	• 2x(4x4x4mm) electromagnets, 350mW. Custom wound, 700 winds, 42 gauge wire. • 2x(3x3x3mm) rare-earth permanent magnets. NdFeB, www.amazingmagnets.com
Communications	• 4 Full duplex concurrent channels • Wireless, software adjustable, 1 – 10mm range • Send and receive directly with microcontroller • Small size: 2x2x4.5mm electromagnets, 2-wire connection • Low power: 2mW transmit, passive reception
Base	Corners sit on 50x50mm square. Laser cut 4mm acrylic with features (6.9gm) over .9mm acrylic base (4.7gm). • 4 x flexures for squeeze fit of comms magnets • 2 x flexures for squeeze fit of latch magnet • Total mass: 11.6gm
Latches	Laser cut 3.2mm acrylic Press fit permanent magnet hold
Roll-cage (prevent latch bounce-out	Laser cut 3mm acrylic Press fit to base
Electronics package (logic battery actuators)	Total size: 25x20x6mm, 6g
Total package	Size: 50x50x15mm, ~26gm.

Table 6-1 Specifications for electromechanical units.

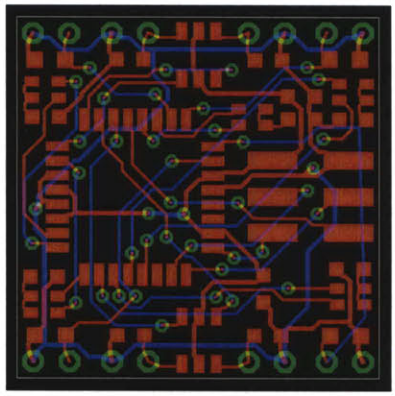


Figure 6-8. Circuit board layout. Circuit board is 20mm on an edge.[2]

[2] Design and implementation of circuit board with assistance from Dan Goldwater.

6.4 Debugging and maintenance.

50-100 of anything is probably the worst number to make. Below 10 hand assembly and debugging is simple. Above 1000 outsourcing the manufacturing is economic. Unfortunately this project sat in the middle. As such tools for debugging and maintenance were also built to simplify the process of keeping sufficient units running correctly.

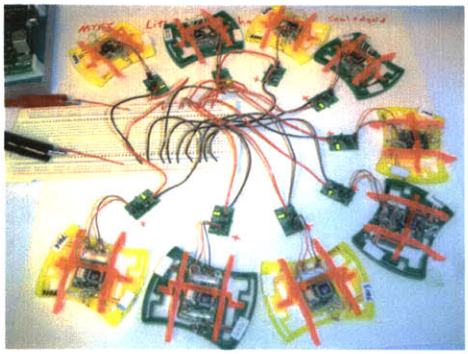


Figure 6-9 Multi-unit recharging bay with a simple circuit board driving each.

Figure 6-9 illustrates a multi-unit recharging bay, each battery served by a simple charging circuit board specifically designed for the Lithium Polymer batteries.

Figure 6-10 shows the communications testing unit. The communications coils are tested by placing a unit in the centre or this piece. LED's on the circuit board at left in this photo indicate whether each communications coil is capable of both send and receive.

Figure 6-11 shows 4 units each used in debugging and programming. Units labeled D1 and D2, at left, were designed to probe the internal state of other units, and to provide signal-level breakouts for probing with an oscilloscope. The two units on the right in figure Figure 6-11 are for reprogramming the units without needing to connect them to an external computer. There was enough space in memory on each unit to store 4 programs at any given time as well as a colour identifier. The top right unit would program any given unit to be either green, or yellow for algorithms where colour was sensitive, and the lower right unit would switch the units between 4 programs (numbered 1-4).

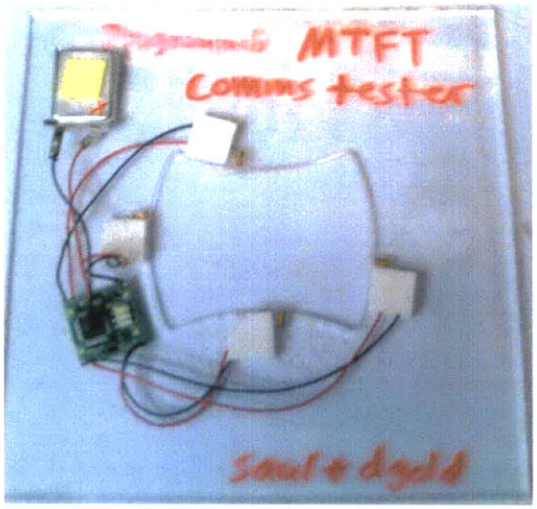


Figure 6-10 Communications testing unit. This unit tests the communication coils on all 4 sides of a unit placed in the centre.

Further debugging functions were programmed into the units such as an auto latch test upon powering up (toggle both latches up and down) and state indicator sequences in the LED's on each circuit board. Similarly each unit would identify whether it was transmitting or receiving at any given time via two colour (red/green) LED's for each of the 4 communications coils. A similar LED indicator was associated with each latch such that faulty latches could be identified during experiments. Low battery charge, for example, could prevent latches lifting during experiments, similarly in the occasionally violent environment of a random-collision air table, electromagnet wires would occasionally fail.

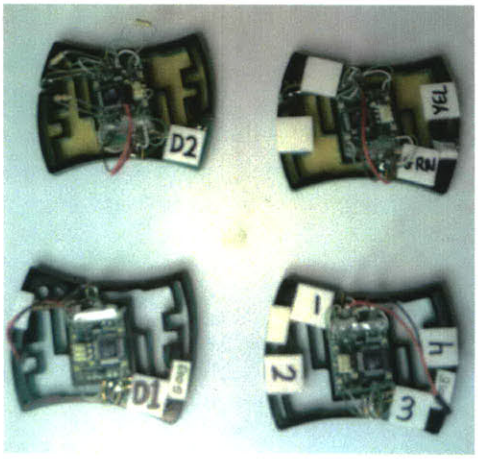


Figure 6-11 Debugging and reprogramming units.

6.5 Communications

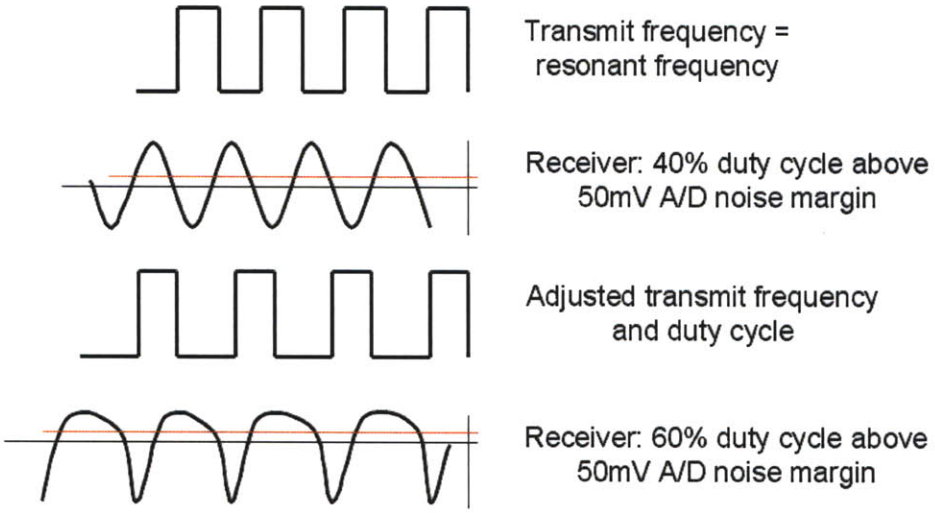


Figure 6-12 Schematic of transmit and receive implementation in communications channel.

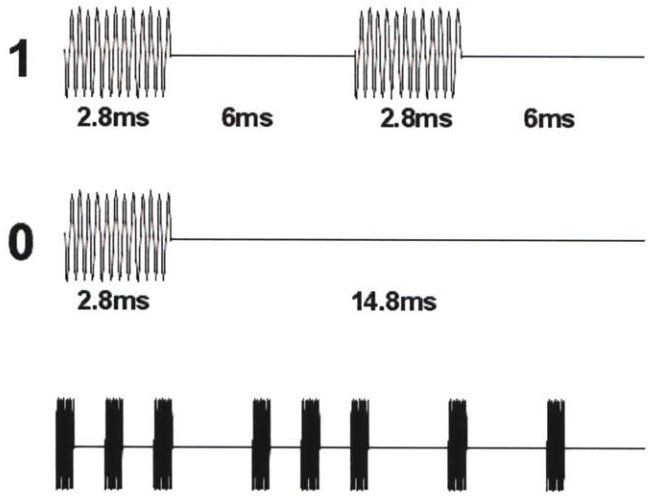


Figure 6-13 Schematic of collision avoidance implementation on communication channels. Coils receive when not transmitting & randomize transmit to avoid collisions. Packets are 12 bits (7 data bits) 210ms transmit time.

Figure 6-12 & Figure 6-13 [3]demonstrate the method by which communications was achieved via the coils on each of the 4 sides of the units. Whenever the coils were not transmitting their state they

[3] The communications protocols shown here were designed and implemented with the assistance of Dan Goldwater.

were listening for incoming tiles. The transmit times were randomized to avoid collisions. In such a communications environment many error sources exist: crosstalk, transmit collisions and communication distance variation. Multiple methods of error detection were implemented to ensure high communications integrity they can be simply described as:

- Packet size: discrete 12 bit packet must be received
- Fixed bits: 4 bits of the data packet are constant, these are spread through the data packet
- Parity bit: parity bit is appended to the data packet
- Redundant transmission: two identical transmissions must be received in sequence to be considered valid

Experimentally it appeared that this communications protocol was sufficient for the scale of these experiments (both temporal and spatial)

6.6 Assembly Environment

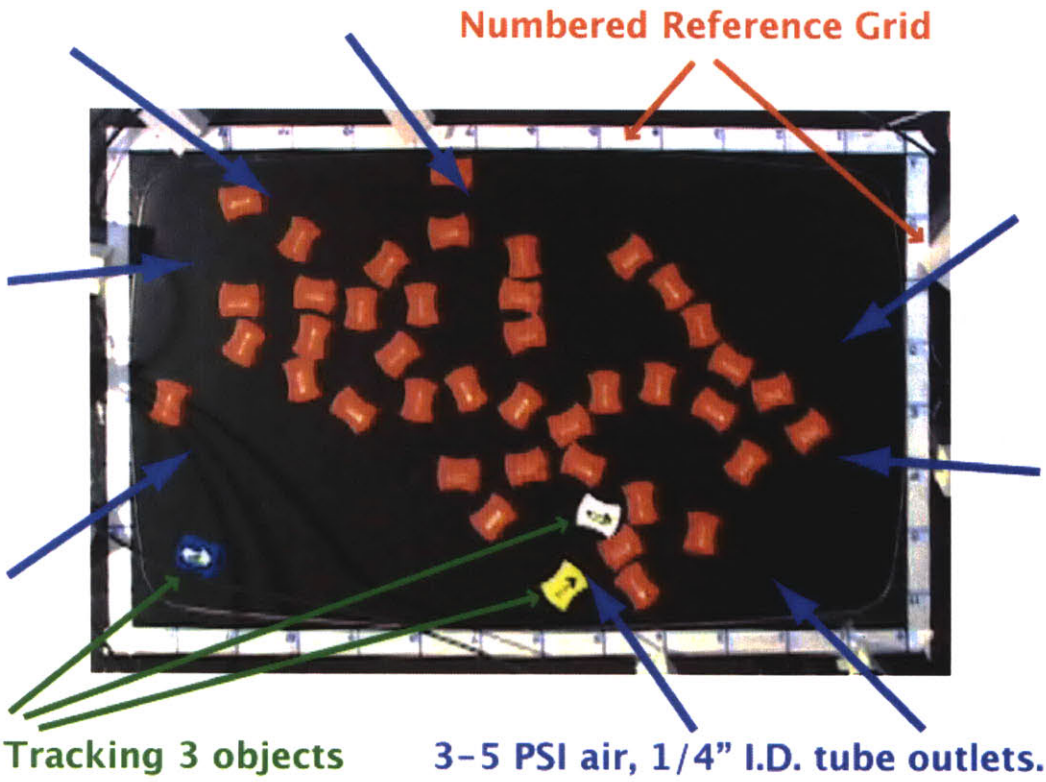


Figure 6-14 Experimental verification of system dynamics with dummy parts. The direction of incoming air streams is indicated by the blue arrows. Three uniquely coloured dummy tiles were tracked on video.

Figure 6-14 illustrates the assembly environment. A custom-built air-hockey table was used to constrain assemblies to 2 dimensions and simulate a frictionless environment. 0.8mm holes were drilled throughout the air-table at 1inch intervals on a square grid. A 130 x 75mm square constraining frame was cut in a 3mm polycarbonate sheet that sat over the air table. The lip of this polycarbonate sheet acted as a retaining barrier constraining the parts within those bounds. The maximum floating weight for a tile of the given geometry was determined and units were designed and lightened to be under this weight. This maximum weight was 30+/- 2 grams.

To introduce randomness to the system 8 air jets were introduced from the sides of the table. These air jets were run through 1/4" I.D. polypropylene tubing that was anchored by weights at the sides of the table. Unregulated house-air was run through the tubes and a valve and pressure gauge was used to keep the exit pressure between 3.5 and 4 PSI. Natural variation in the house-air pressure meant occasional variations above and below these values during the course of lengthy experiments. These values were experimentally determined to maximize reaction speed and minimize random unit damage due to high energy collisions. Qualitative observation of collisions in this environment gave confidence in them being largely elastic collisions.

1.1.1 Data recording

All experiments were recorded with a Sony miniDV camcorder equipped with time stamp. Video was processed and analysed with a combination of I-Movie, Final Cut Pro, Adobe After Effects, and Quicktime Professional.

6.7 System Dynamics

System dynamics were experimentally verified using the set-up in Figure 6-14. Fourty-two dummy units of equivalent mass and 'sail' area (the vertical area presented to the side air jets) were run on the table for 5 minutes at a time. 3 different coloured (blue, yellow, white – marked with a direction vector) dummy tiles (similar in mass and sail area) were tracked. At 1 second intervals during the course of the experiment the X and Y coordinates and orientation (divided into 22.5 degree slices of the 360 degrees of rotation) were recorded.

Figure 6-15 shows screenshots at 15s intervals during one run of the control experiments. The orientation of the parts and their position is clearly visible.

Fifty seconds worth of the data is shown in Figure 6-16. More data points did not illuminate anything extra and only confused the graph. As might be expected the components tended to clump in the centre of the board at furthest distance from the incoming 'randomising jets'. Overall system dynamics could be observed qualitatively with trends in clumping for different positions and orientations of the incoming jets.

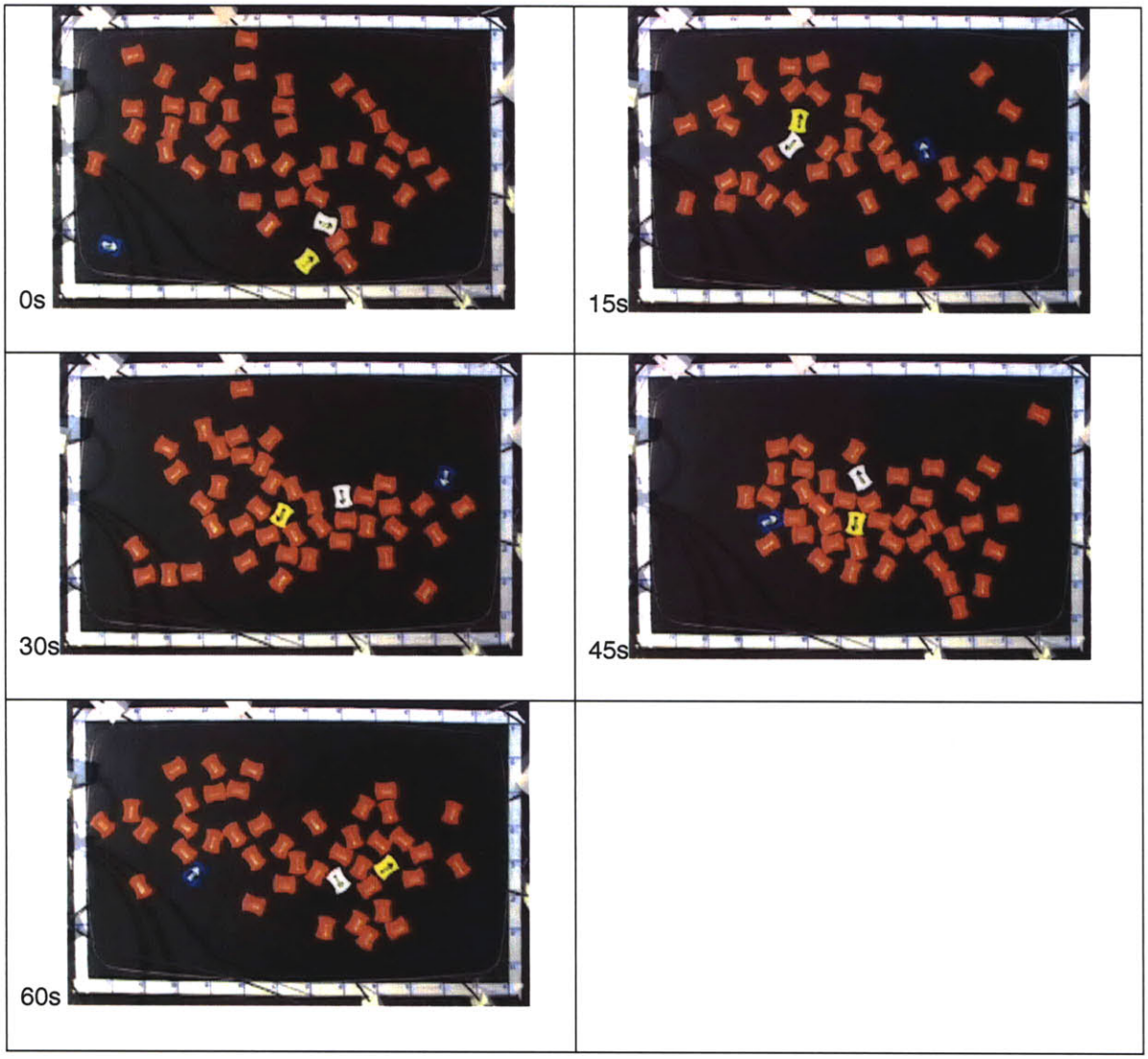


Figure 6-15 Screenshots from system dynamics control experiment taken at 15 s intervals. Position and rotational orientation are recorded for the blue, yellow, and white tiles.

A set of orientations for all 8 jets that was least violent in terms of high speed individual collisions, yet maximized number of collisions was decided upon qualitatively. The data in Figure 6-16 was taken with those positions and orientations which are illustrated as blue arrows in Figure 6-14. The jet positions were retained for all experiments. This dynamics data convinced the author of sufficient local randomness as seen by each tile to assume the entire system provided random collisions over the typically 30-60 minute time intervals of the programmed tile experiments.

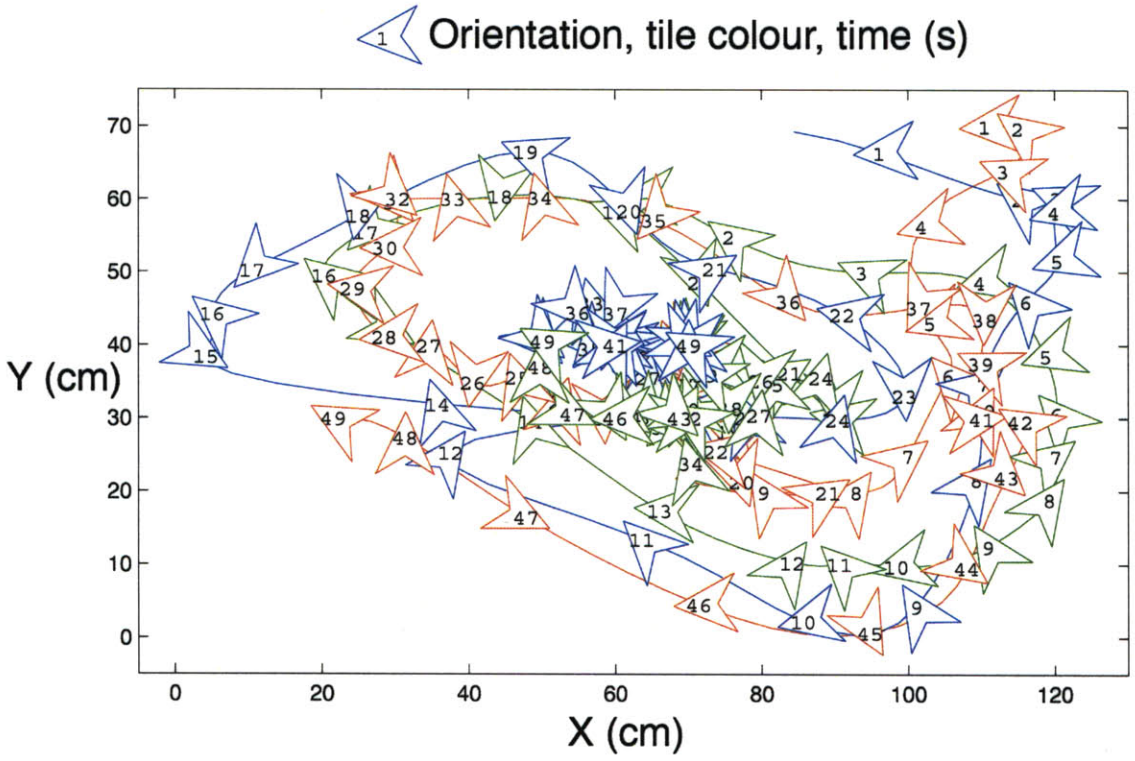


Figure 6-16. Graphed data for a random 50 second interval in dynamics control experiment. Three tiles are tracked at 1s intervals. Data is shown as an arrow indicating orientation, within which is a time stamp (seconds) and a loosely fitted trajectory line. Blue tile shown as blue, Yellow tile as green, White tile as red.[4]

6.8 Programming

The state machines were implemented in C code compiled for the Atmel chip. The code will not be presented here as the details of any particular implementation are not important to the thrust of this work. Rather the programming model for implementation of the finite state machine active on each tile shall be presented. The state machines were implemented as schematically outlined in Figure 6-17. Transmitted and received codes (including unit color) are represented for each of the 4 faces. The received code and color-match that result in a state transition are labeled alongside the state transition arrow which shows the transition from one tile state to another.

Because of the square lattice tiling rather than the offset square lattice tiling used in chapter 5, more than the 6 states required for the mechanical replicating state machines were used. These were essentially memory states such that data could be communicated to non-contact tiles. Extra states

[4] Matlab generation of these dynamics with kind assistance from James McBride.

were also used to handle the time delays in communications that are not seen in the kinematic model presented in Chapter 5.

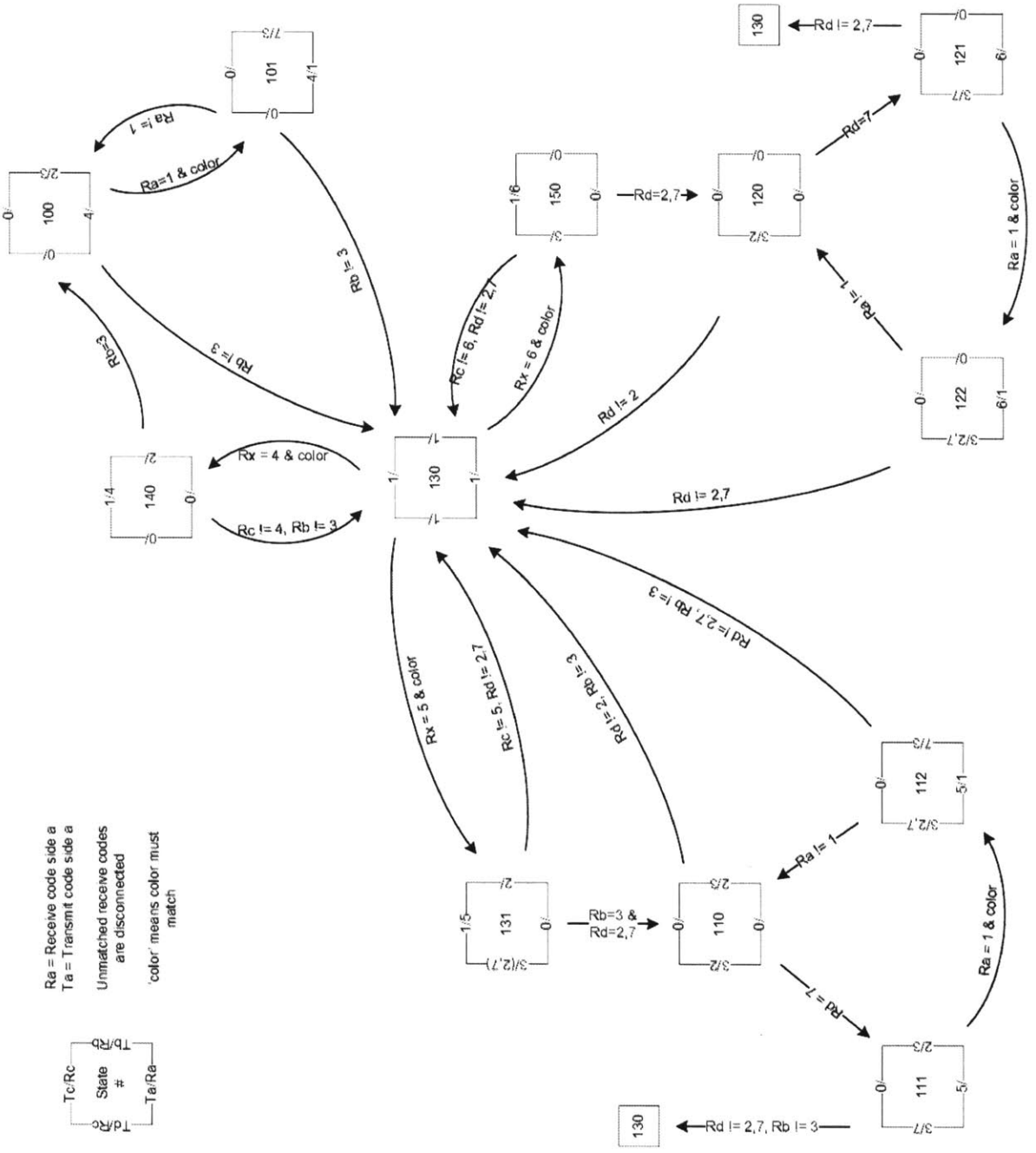


Figure 6-17. Schematic of programming model for implementing finite state machines on individual units. The most complex of the state machines, the self-replicating code, is presented in this diagram. Diagram with assistance of Dan Goldwater.

6.9 Experimental

3 principal algorithms were run with the electromechanical units. Line formation, error-preventing crystal formation, bit-string replication, and logic-less control crystal formation. There were two principal tile colors, green and yellow, with a few specially appended colours to denote (red) seed tile for crystalisation and (black) infectious tile for reprogramming virally. Color dependent, and color independent experiments were run for both line formation and error-preventing crystal formation. In color dependent experiments the extra state of tile color was considered in the state machine giving single colored lines (line formation) or chess board patterns (error-preventing crystal formation). In colour independent experiments tile color was ignored giving multicolored lines or randomly colored crystals. System dynamics for all experiments were controlled according to section 6.7.

The state diagrams are presented slightly differently here to more accurately reflect their implementation in this system. Rather than represent the state of the entire tile, the state of each face is shown. This state describes the broadcast and receive signals and binding condition for each face.

1.1.2 Logic-less control crystal.

This is the control experiment which demonstrates the value of logic in self assembly. Obviously these are non-annealing structures as the bonds are irreversible, similar to the glasses described by whitesides in chapter 2. The experiment was run multiple times by not connecting the batteries, thus all latches bound and did not release. The state diagram can hence be drawn very simply as in Figure 6-18. The results from two runs of this experiment are shown in the still frame excerpts of Figure 6-19 and Figure 6-20. The final frame (bottom-most at left in both series) shows the final state of the system at experiment's end. The two structures are both notably different, as one would expect, and will likely be different for every assembly. Both structures show growth errors where vacancies are entrapped by growth faces. As might be expected these experiments had the fastest kinetics of all assemblies as no time-delays for logic to occur were present, nor searching for colour matches, nor waiting for parts to bind to specific growth faces. These kinetics can be seen most vividly as compared to logic limited crystallization in Figure 6-28.

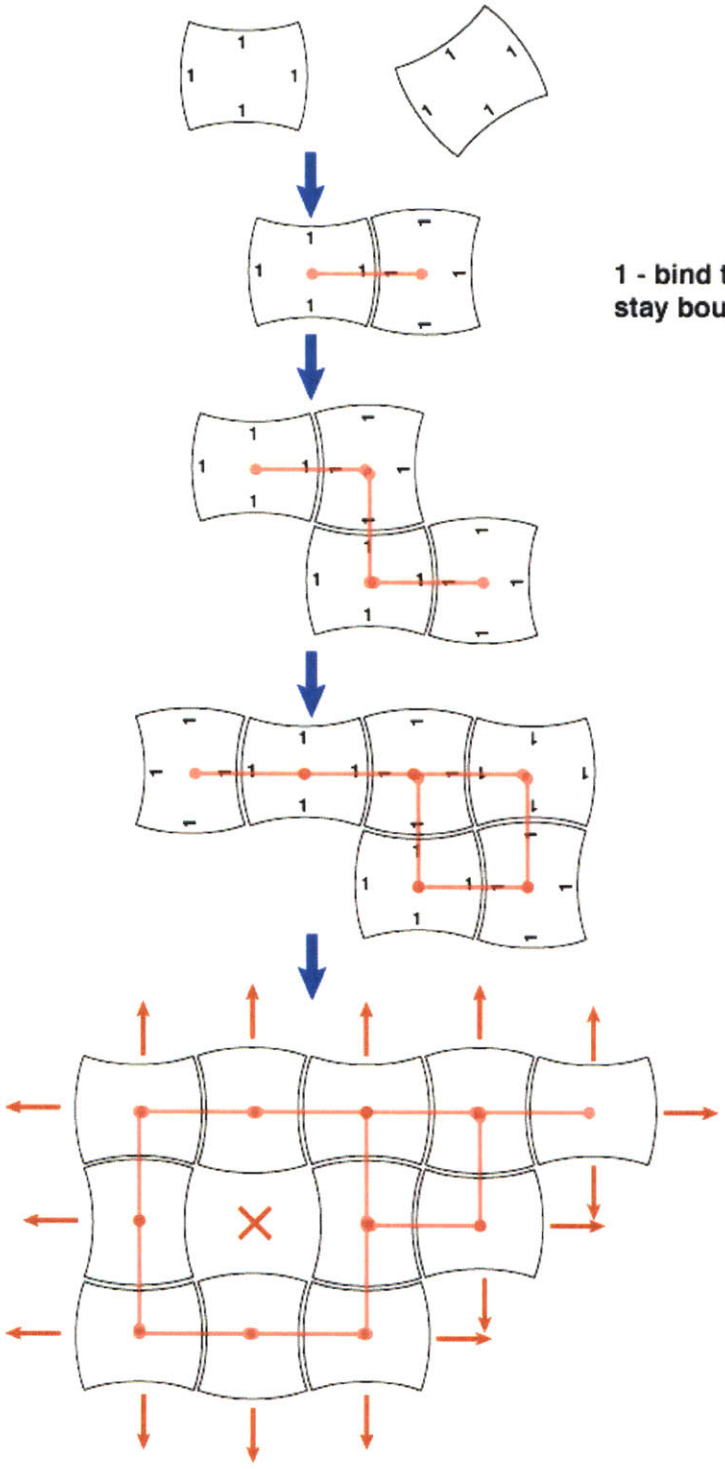


Figure 6-18 Schematic of state machine implementation for logical epitaxy of error free crystals.

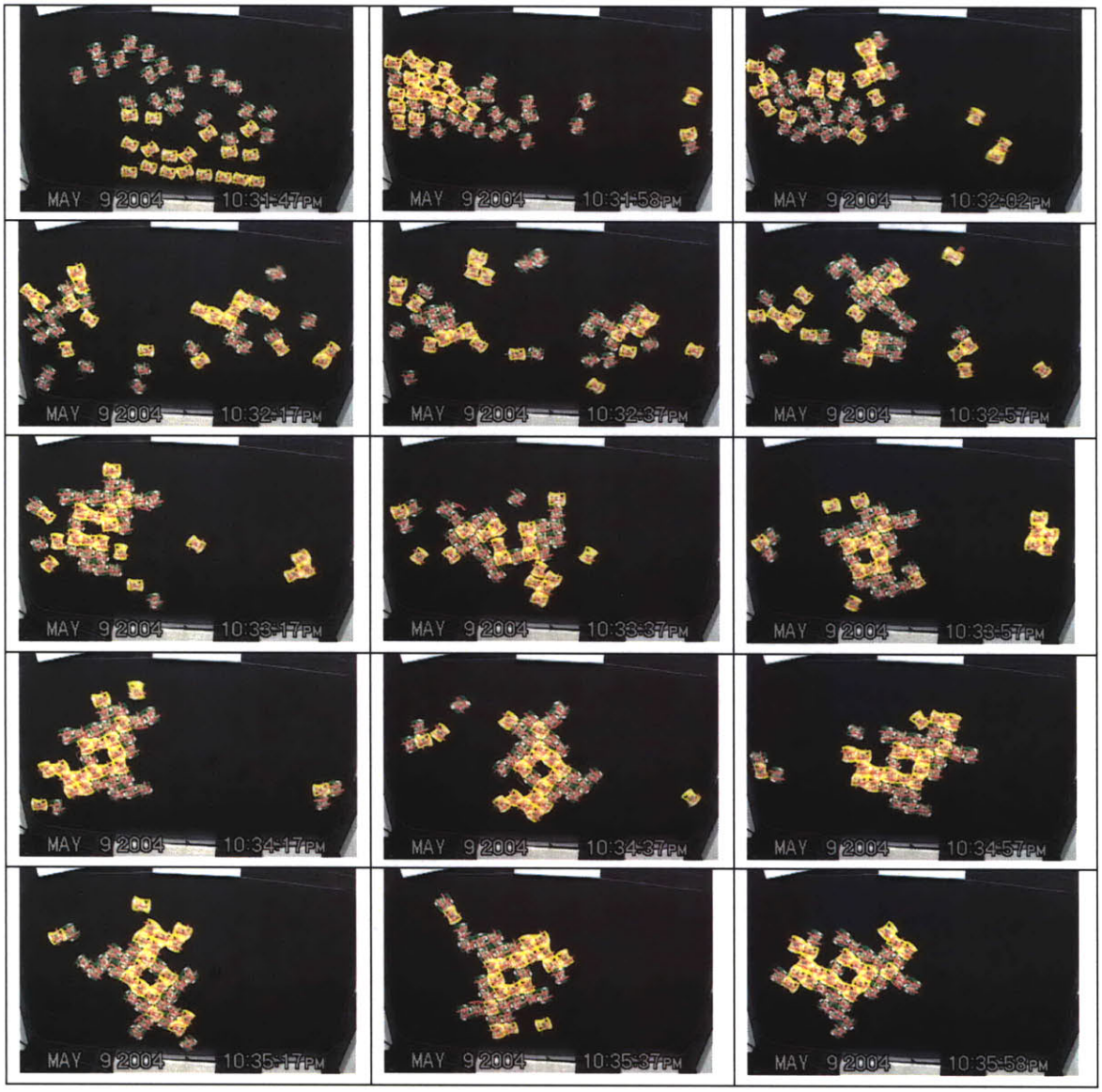


Figure 6-19 Logicless crystal control 1. Series proceeds from left to right, top to bottom. The final state is seen in the bottom-most right hand picture. All units are bound.

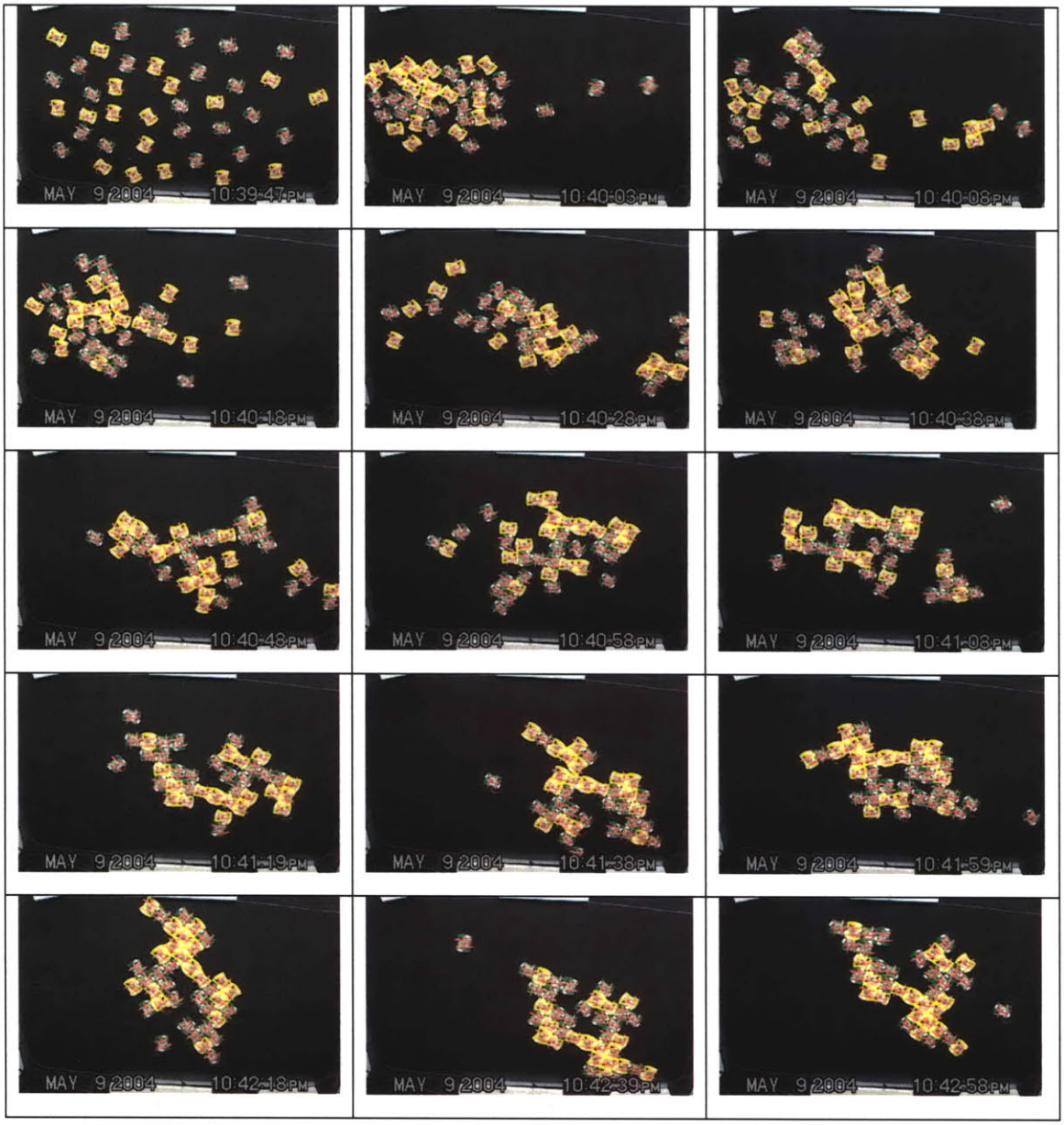


Figure 6-20. Logic-less crystal control 2. Series proceeds from left to right, top to bottom. The final state is seen in the bottom-most right hand picture. All but one units are bound.

1.1.3 Line formation

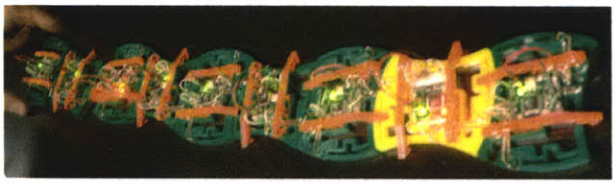


Figure 6-21 Lines are linear, end-end, string formations of tiles, either color specific or mixed (as is shown above).

The simplest program to be implemented was a polymerization, or line forming algorithm. This algorithm produces linear strings such as that in Figure 6-21. This algorithm was tested both for lines of a single color and lines of mixed colors. The line formation state diagram is illustrated in Figure 6-22. Screenshots from the implementation of this algorithm for color specific lines can be seen in Figure 6-23. The difficulty of implementing this algorithm on the experimental set-up as described is that the small 'reaction frame' defined by the polycarbonate fence limited quickly the lengh that the polymer could grow to, and indeed it would quickly jam between sides of the frame, occasionally with enough force to break the bonds.

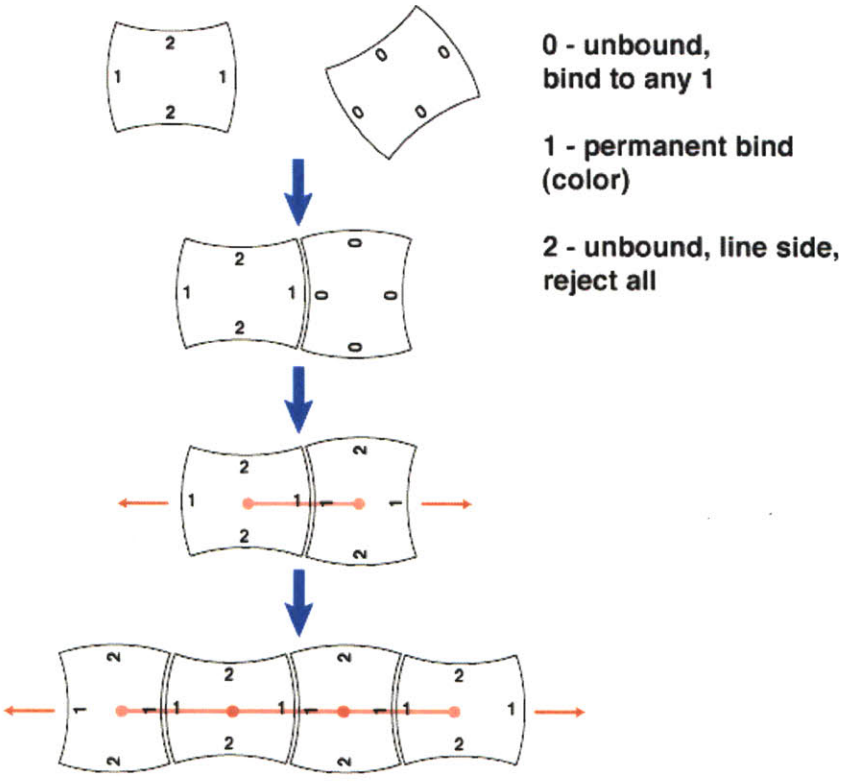


Figure 6-22 Schematic of state machine implementation for polymerization or line formation.

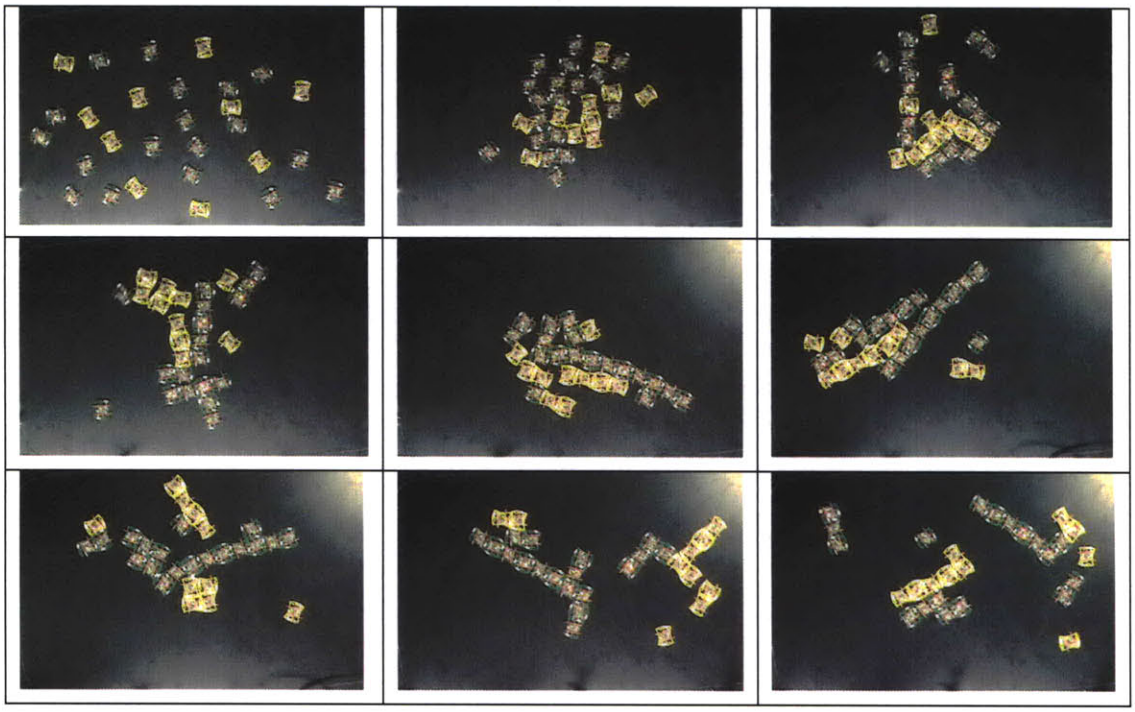


Figure 6-23 Screenshots from a line formation experiment. Series proceeds from left to right, top to bottom. The final state is seen in the bottom-most right hand picture. A number of different polymer chains can be seen to have grown.

1.1.4 Error-preventing Crystal Growth.

I will also call the error-preventing crystal growth algorithm the logical epitaxy algorithm. A non colour specific example may be seen in Figure 6-24. Figure 25 is the non color specific example. Similarly this algorithm could be implemented to produce checkerboard patterns as will be seen in Figure 6-26.

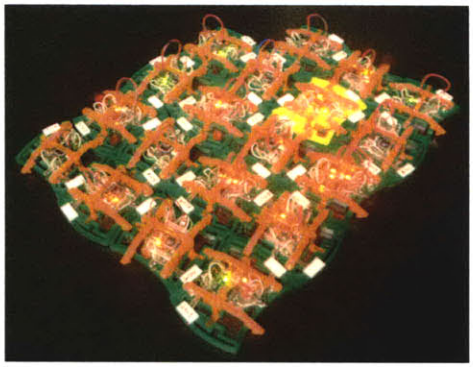


Figure 6-24 Error preventing crystal growth by spiral growth face. Also could be described as logical epitaxy.

The algorithm for logical epitaxy is outlined in Figure 6-25. The crystal can be seen to grow from a spirally defined growth face that only allows a single unit growth face. Screenshots from this

experiment can be seen in Figure 6-26, and the kinetics are analysed in comparison to the kinetics of the crystal control experiment in Figure 6-28. It can be seen that the logical overhead of this algorithm makes the growth kinetics significantly slower than random aggregation due to the single, as opposed to many, growth front.

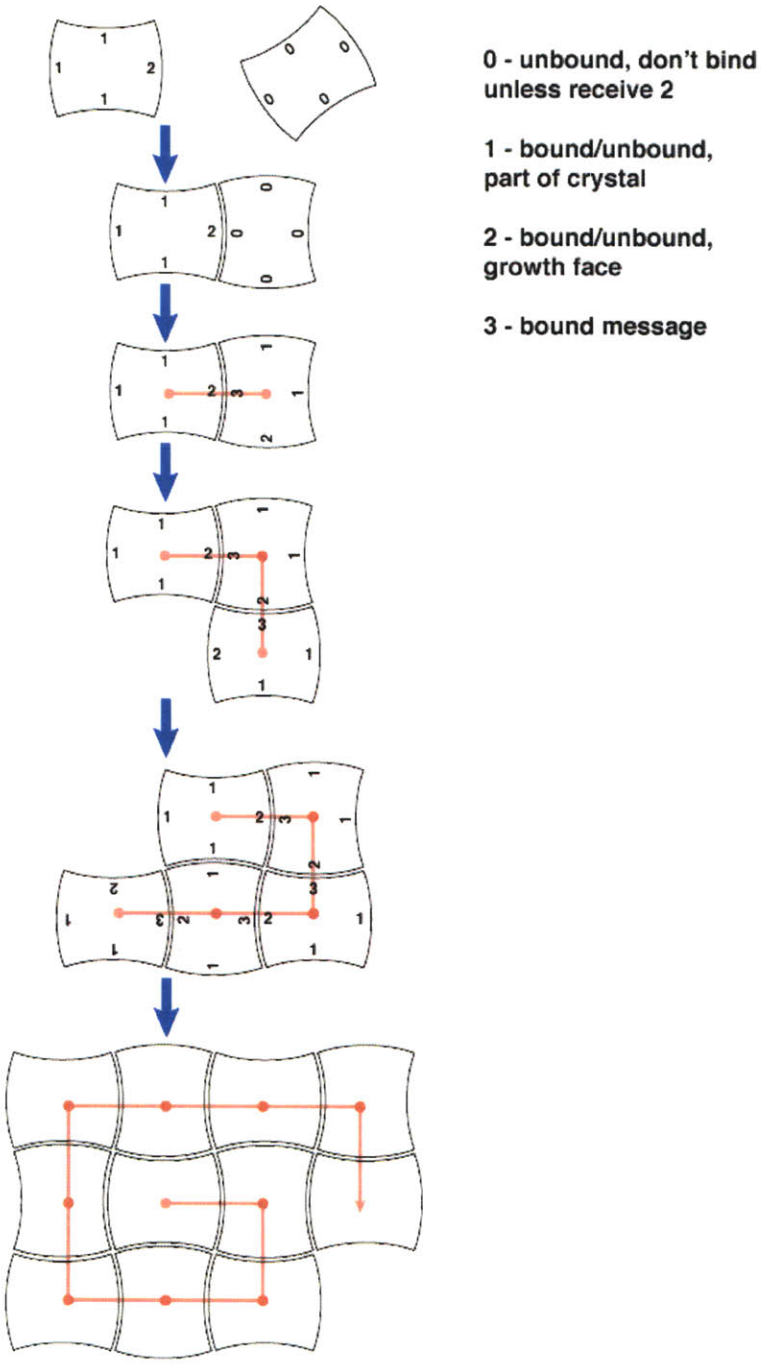


Figure 6-25 Logic Limited Aggregation algorithm.

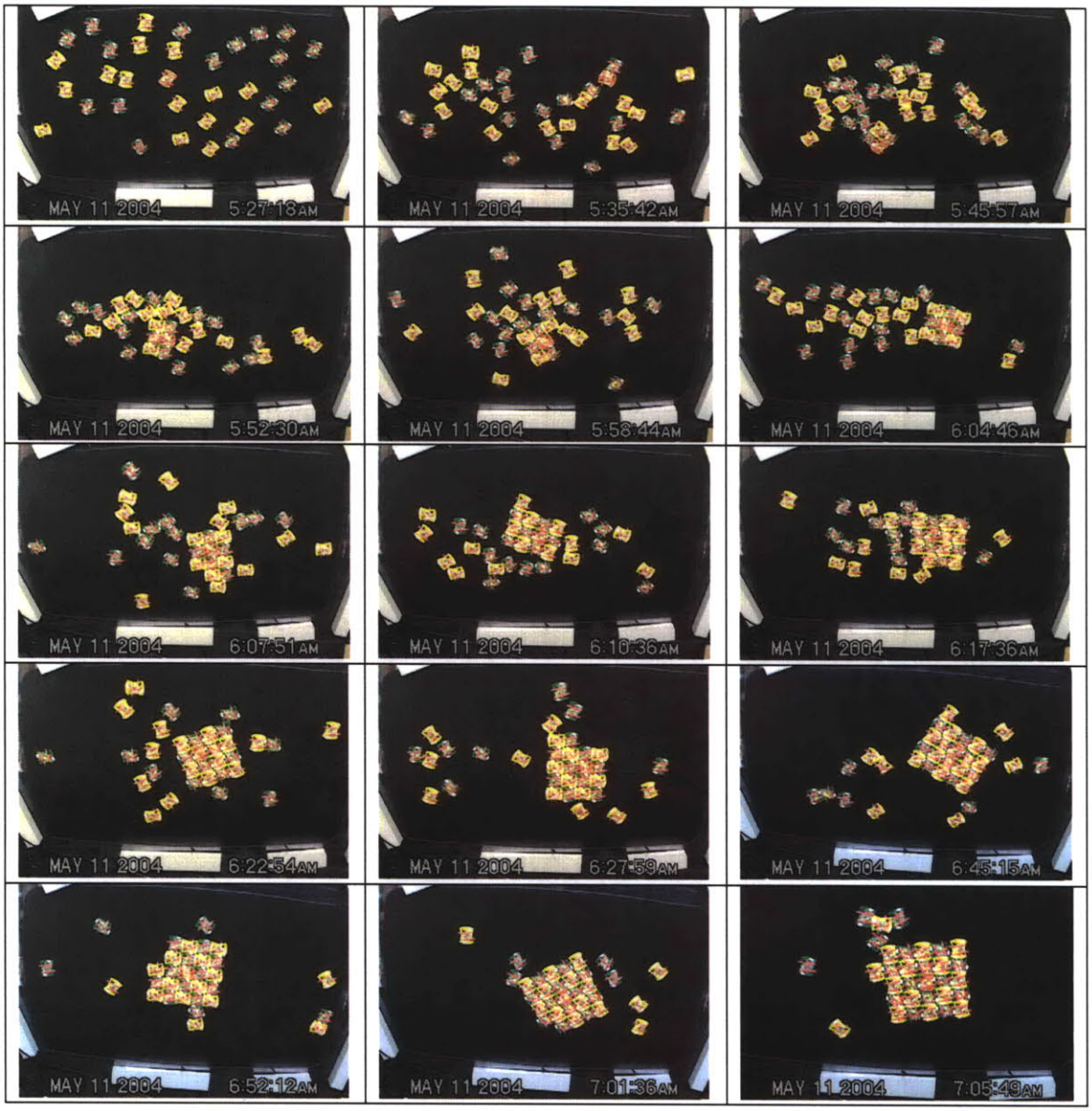


Figure 6-26. Screenshots from a logic limited aggregation experiment. Series proceeds from left to right, top to bottom. The final state is seen in the bottom-most right hand picture.

1.1.5 An autonomous, self replicating machine.

If we define a self replicating machine as a collection of sub-components, capable of replicating itself from a sea of similar sub-components, then the following experiment successfully demonstrates a self-replicating machine. The algorithm for this experiment implements the state diagram of figure 5-18 that was designed for the mechanical system of chapter 5. The implementation of this algorithm was not amenable to the face-state treatment of Figure 6-18 & Figure 6-22. The implementation was done according to Figure 6-17. A length 5 colored polymer, sequence Green, Green, Yellow, Yellow,

Green, was introduced into the assembly environment as shown in the 1st of the screenshots of Figure 6-27, all other units were unbound. From this initial condition the experiment was allowed to proceed until 3 copies had replicated. To aid in the visualization of the experiment the 5-mer's were tracked by software (Adobe After Effects) by using contrast tracking of the beginning and end tiles of the 5-mer, from the video track. At the conclusion of the video in the last screen shot the 4 5-mers, 3 replicated and original, can be seen as well as the remaining individual units, one of which is being pointed to. Each replicant is numbered in order of production. Similar experiments were run for 3 and 4-mers.

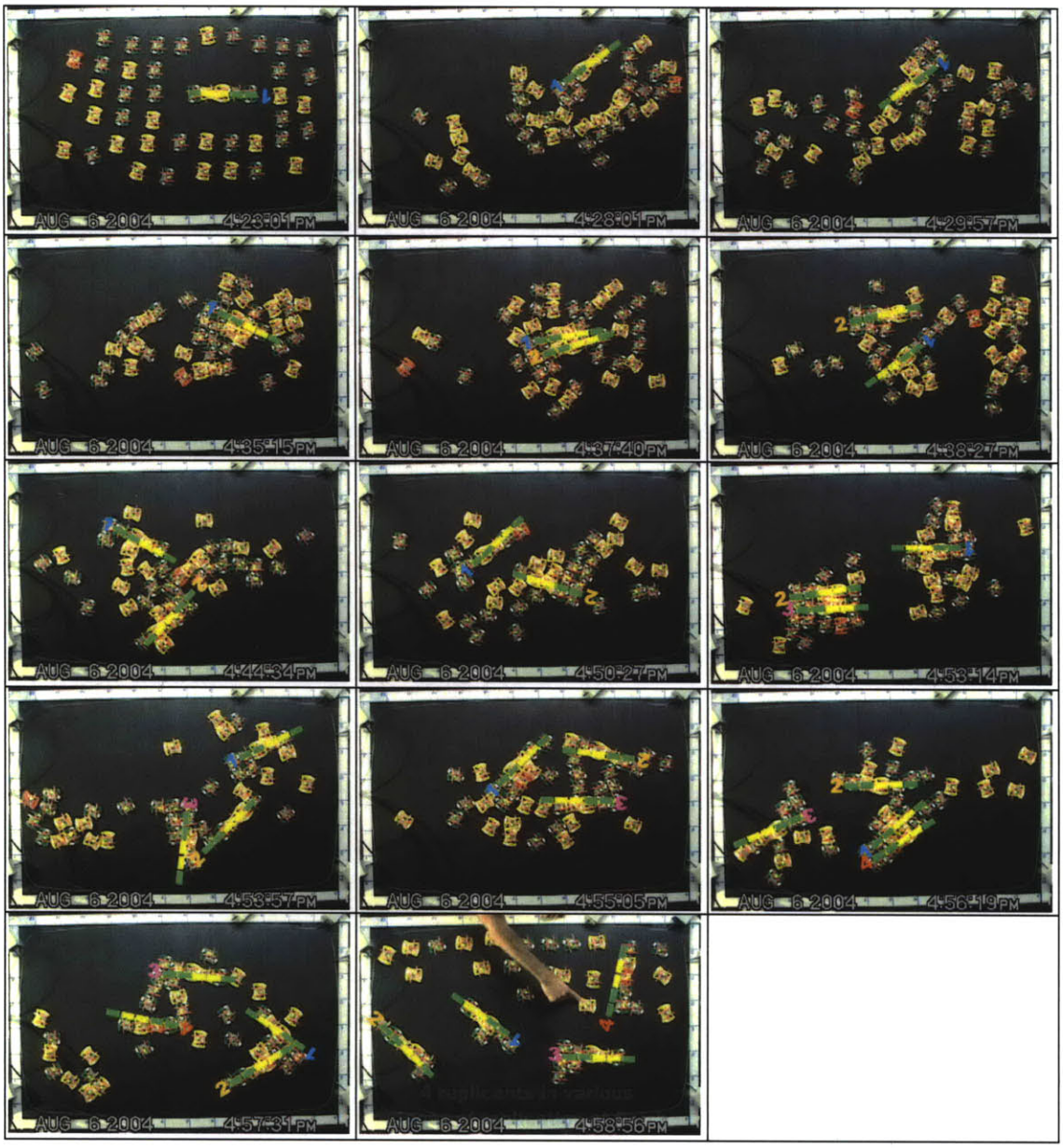


Figure 6-27 Screenshots from a self-replicating bit-string experiment.

1.1.6 Reaction kinetics.

Data collection for the kinetics of these assemblies (also referred to as reactions) is laborious. Video data is analysed manually. The number of bound tiles and other information is recorded along with the video time stamp at periods throughout the assembly by observation of still frames of the video. Some small error is assumed for the counting of bound tiles as video resolution is not high enough to always see whether each tile is firmly bound or about to be rejected. The small number of units in the system also limits the data collection. For example, in the replication experiments where one would expect the process to consume parts exponentially faster with time, the reactants are mostly consumed after a small number of replicants. For all experiments a similar number of units (around 40) were used, although almost 60 units were made, at any given time there were a number being recharged, repaired, reprogrammed, or debugged.

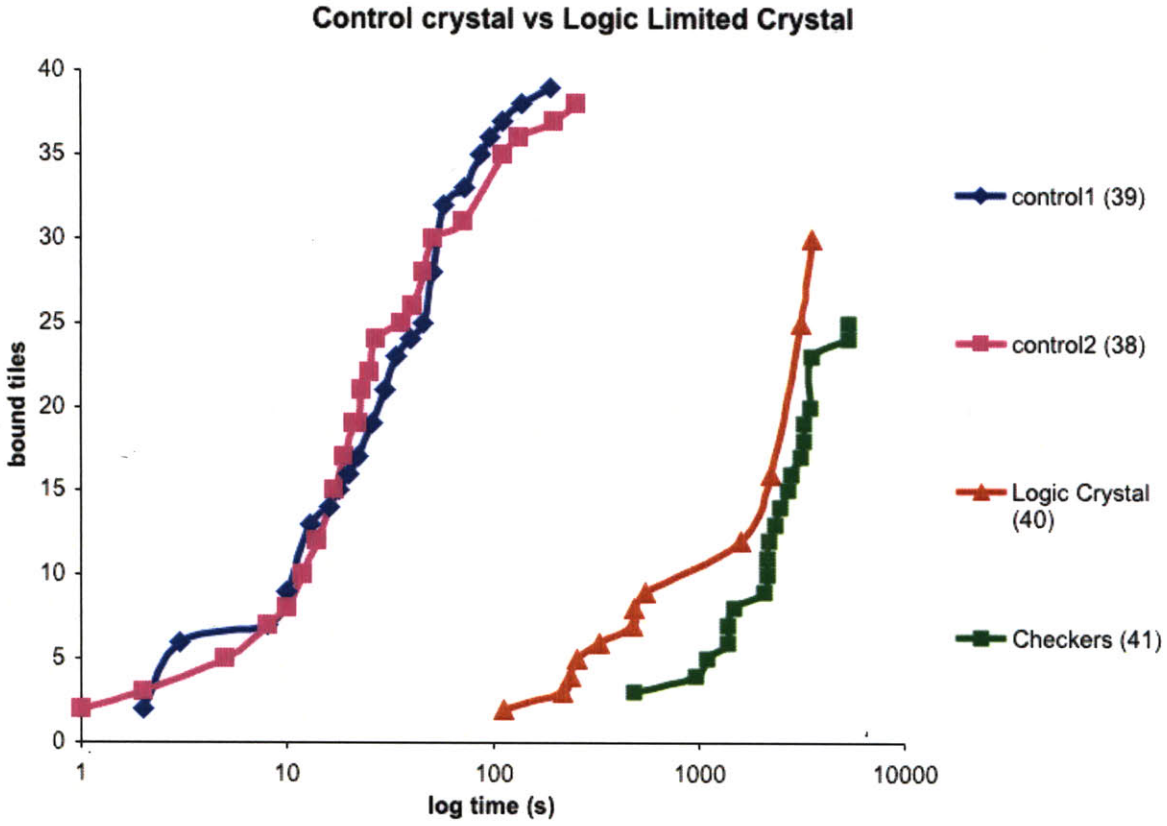


Figure 6-28 Comparing the kinetics of logicless (and error ridden) logicless, irreversible binding, to logic limited aggregation on a spiraling growth face which grows without vacancies or errors.

Figure 6-28 compares the kinetics of the logic limited crystal to the random control crystal. Both the color dependent and color independent data is shown for the logic limited crystal. As might be expected the logicless crystal follows roughly sigmoidal kinetics as might be expected in a small

reaction vessel. Slow start with few groth faces, followed by a fast growth period that slows as the reactants are consumed ultimately to stop growth when there are no remaining unbound units. The logic limited crystals might be expected to be basically linear, trailing off as the reactants are consumed as there is always only a single growth face accepting only a single unit. The slow start here is due to the mechanics of the components. The small crystal at first has low mass and therefore low momentum and the kinematic collisions are more successful between heavier components as the latches are more likely to be lifted from their neutral state. As the crystal grows it's mass is higher and the kinetics start to become linear before trailing off as expected.

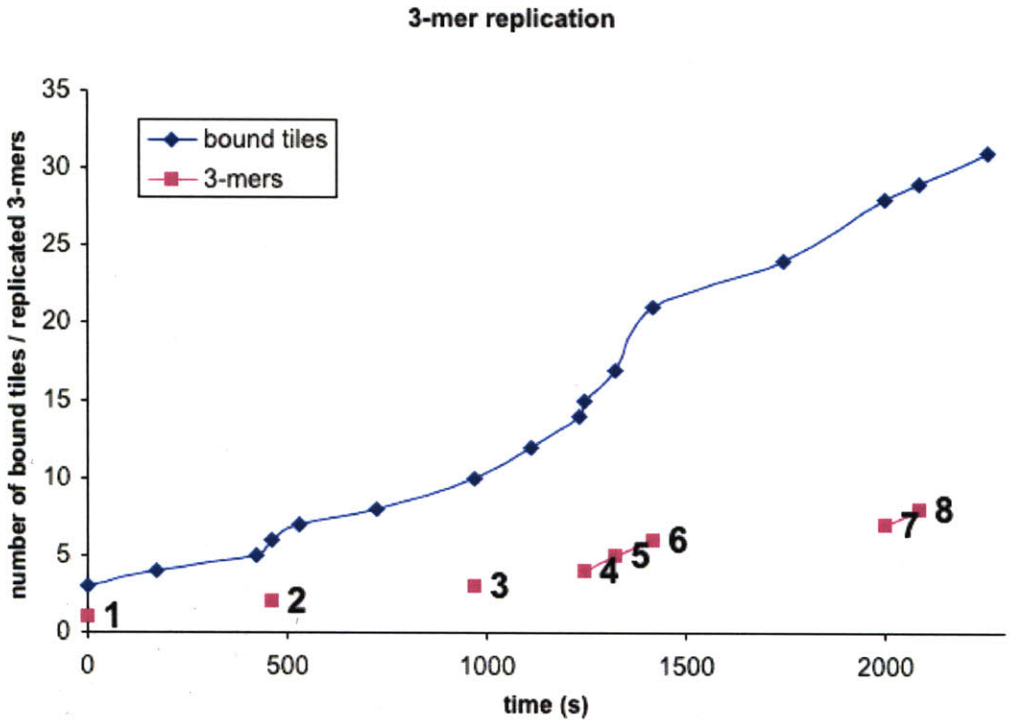


Figure 6-29 Number of bound tiles with time, and number of replicants for a 3-mer replication experiment. Bound individual units shown in blue, complete, bound replicants and original string numbered in pink.

Figure 6-29 looks at the replication of 3-mer's. One should expect exponential reaction rates as the number of growth faces increases exponentially with the number of copies. Unfortunately the tragedy of small component count systems is demonstrated here. The kinetics seems to begin going exponential at the 4th, 5th, and 6th copies, however at this stage roughly half or more of the units are consumed in replicants or partial replicants. The reaction slows. Figure 6-30 shows a similar trend. At the 3rd and 4th copies the bind rate seems to begin increasing, but again, at this stage too many unbound units are consumed. It will be exciting indeed to miniaturise similar components and test for the kinetics in a less bounded reaction vessel by higher reactant number.

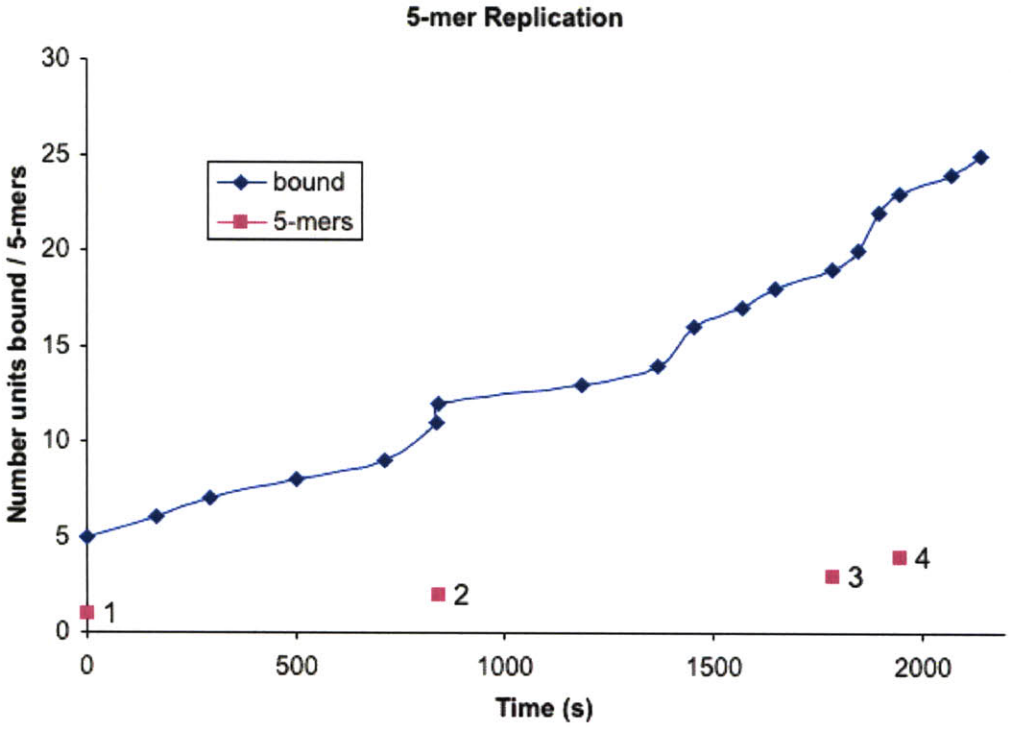


Figure 6-30 Replication of 5-mers. Bound individual units shown in blue, complete, bound replicants and original string numbered in pink.

The reaction rates for all experiments are plotted together in Figure 6-31 (log time) and Figure 6-32. The time cost of logic in the kinetics is clearly shown. The fewer growth faces dictated by the algorithm the slower the reactions. Extra search time is clearly indicated by the fact that color selective algorithms are almost exactly twice as slow as color independent algorithms. This is seen most dramatically in the logic crystal kinetics compared to the kinetics of the checkerboard crystal growth. It would similarly be interesting to run these algorithms in much larger reaction vessels where the stoichiometry of the colored tiles could be varied in color selective algorithm experiments. This particular aspect of this work is highly amenable to computer simulation, and I hope this work informs and to a degree calibrates that type of work.

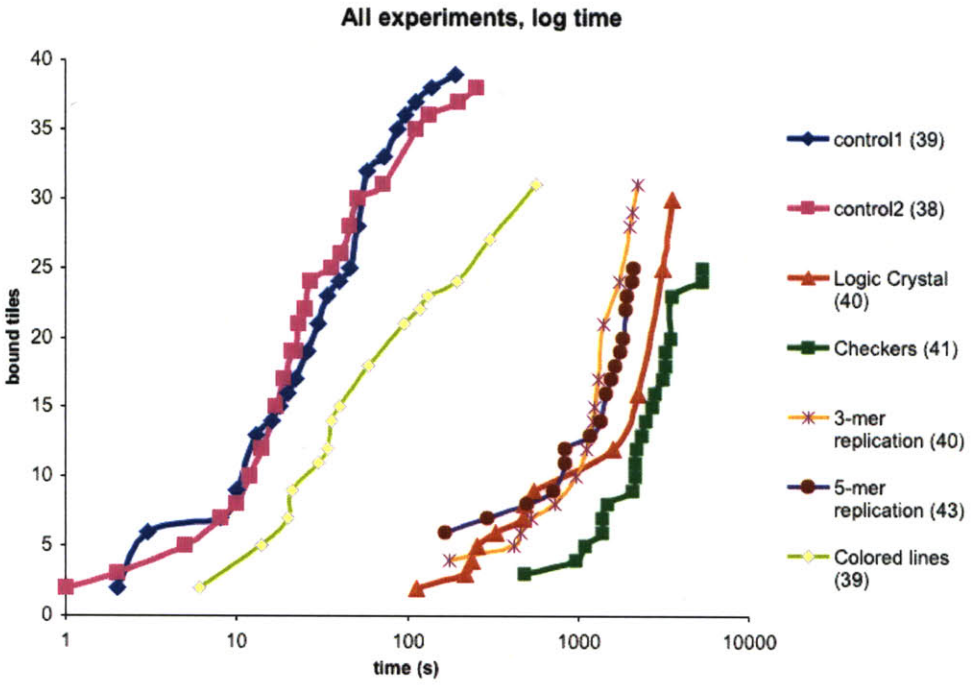


Figure 6-31 All logic limited self assembly experiments graphed together against log time. Number of tiles in reaction vessel shown in (brackets)

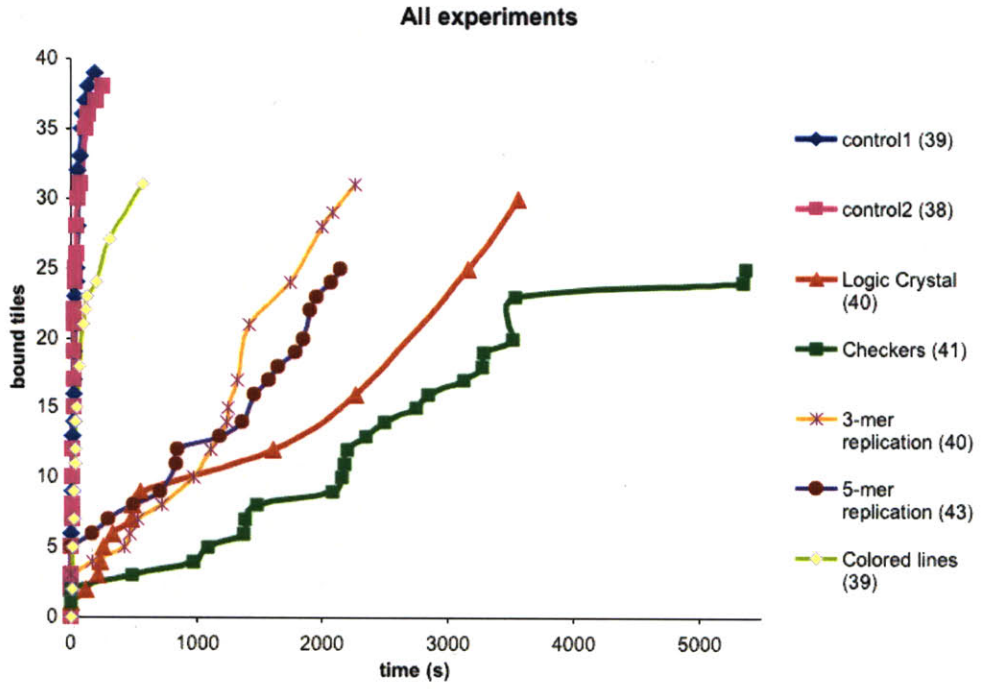


Figure 6-32 All logic limited self assembly experiments graphed together against time.

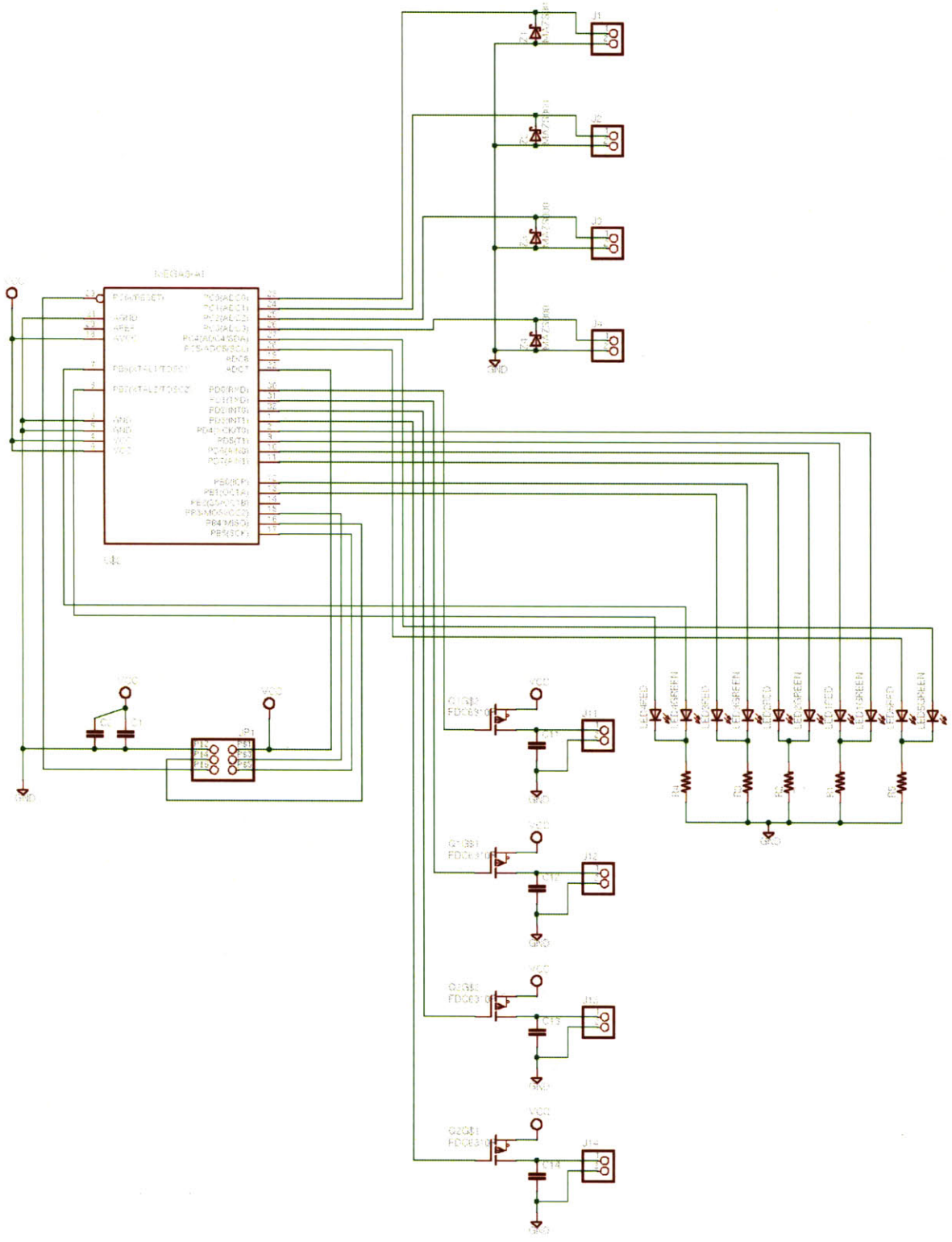


Figure 6-33 Schematic of circuit board used for electromechanical units. (With assistance of Dan Goldwater)

7 Summary and Future Work

Chapters 1,2, & 3, introduced the fields related to programmatic assembly and developed the motivations for studying the addition of state to the components of a self assembling system. Ideas for a hierarchy of assembly that will allow us to build complex 3D objects, reliably and repeatably, in self assembling systems were presented.

In Chapter 4 a geometric construction was developed showing that any 2D shape ccomposed of square pixels can be folded without intersection by a linear string of vertex connected squares provided that the vertex folds are made sequentially proceeding from one end of the string to the other. The geometric construction implies the correct sequence of folds. The cost in resolution of this technique is 4n tiles over any original pixelisation of a 2D object with n tiles (pixels).

Similarly it was shown in Chapter 4 that any voxelated 3D structure can be folded by an edge connected string of right-angled tetrahedra folding sequentially. The resolution cost was 6n voxels in 3D. It is postulated that this is the minimal polyhedral primitive that will fold to any space filling voxelated object. I think this is one of the most fascinating pieces of this work and I am keen to continue examining it. For one, it suggests the role of chaperonin and other molecules that assist in protein folding may be more important than previously realized in creating repeatable (non-prion!) enzymes and proteins.

In 2D it was also demonstrated that the structure could be strongly bound with 4 tile types, edge patterned with attractive and repulsive forces. This was implemented with a magnetically 'programmed' set of tiles from which the letters 'M' ,'I' and 'T' were constructed along with some other structures. It was not determined exactly how many 3D tile types there needs to be to have a similarly strongly bound 3D crystal. I suspect it is also 4 (4 edges for 2D vs 4 sides for 3D) and believe this to be an interesting future question. It would be also interesting to see whether a flat string of faces could be edge folded to fill all of the faces of a 3D object composed of face-only (empty) polyhedra. I suspect this is also possible.

Two designs for pursuing this as a fabrication method were presented, both of which I believe would make interesting further work. The first is a 'machine sequencer' where the tile types are fed into a channel or tube in sequence and fold according to edge or side patterned attractors as they emerge. The second method is a circuit design to implement the 3D case where the power for actuation and the logic for folding is supplied from one end of the string and the resulting structure is reconfigurable. This has interesting implications for manufacture as well as entertainment, education, and reconfigurable robotics.

In Chapter 5 a conformational PDMS based self assembling component for assembly at liquid / liquid interfaces was presented. A 6 state mechanically self-replicating unit was also presented that utilized flexure based conformational logic. The reasoning and difficulties with the design of these classes of components was presented. I believe this to be a fascinating area of future work, in miniaturization, in the development of the design tools capable of facilitating the design of components that assemble in random environments, and in the promise of making truly autonomous and reliable self-replicating mechanisms.

In Chapter 6 electromechanical emulators of state machine assemblers were presented. These were shown capable of replication, logic limited aggregation, and other pattern formation. I believe this to be the first implemented, autonomous self-replicating machine. Again I believe these to be a fascinating area of future research. One new direction is to also build the capacity of actuation into the components by interaction between the components. This is schematically presented in Figure 7-1.

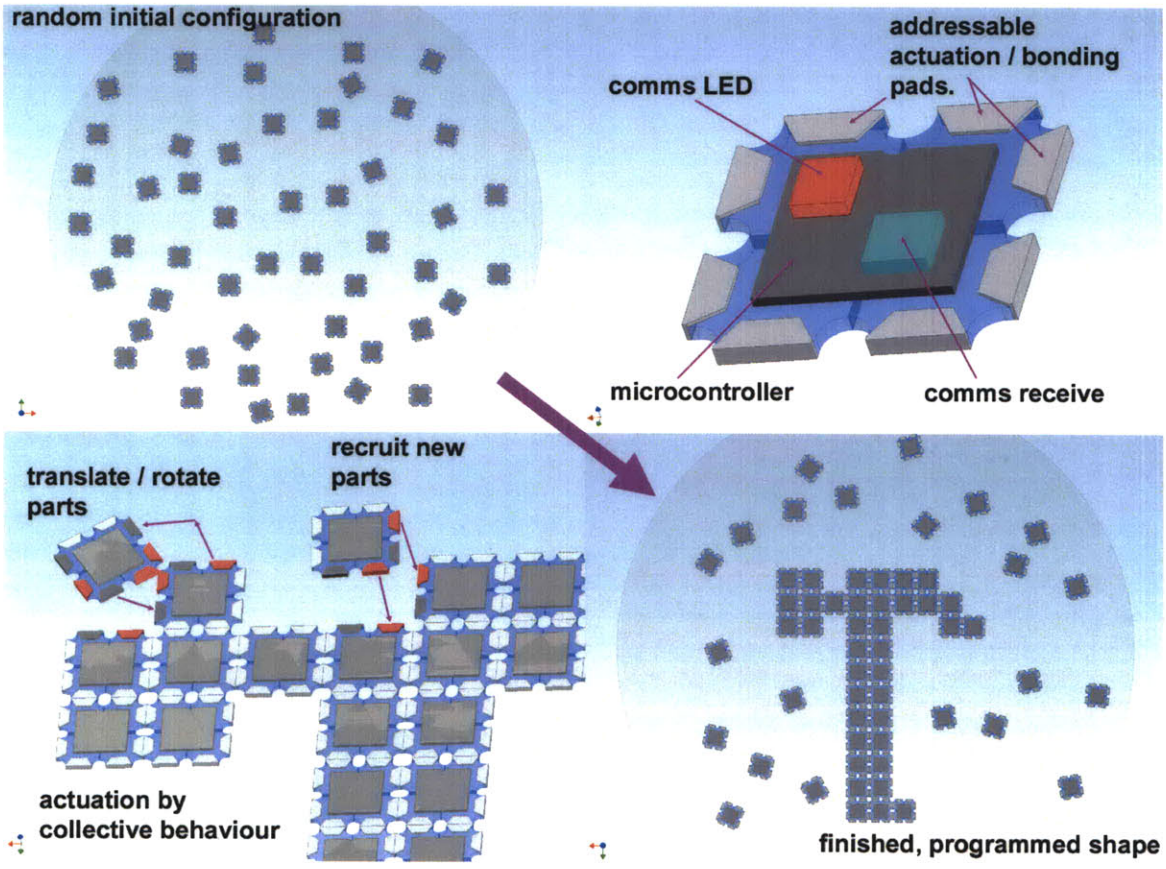


Figure 7-1 Alternative to cellular robotics where actuation is by collective action of components as opposed to fully functioning actuators within each component.

Another area of future work I would like to see explored is hierarchies of assembly. To build truly complex assemblies a hierarchy similar to that used in biology is almost certainly going to be necessary. Just how that hierarchy is designed for synthetic systems should prove quite interesting.

Throughout this whole thesis, structure was abstracted to be purely the assembly of some vague 'sub-unit' or 'tile'. Useful and functional structure is comprised of many different elements and materials to achieve functionality. I am also interested in seeing at what expense adding function to the components of these assemblies would have on the number of tiles required. There may seem to be a lot of redundancy in having 20 amino acids when many have similar function, but as well as being partly due to error prevention / correction I suspect that they differentiate the function of the final structures formed. I would like to explore how adding a material property to the tiles of chapter 4 such as semiconducting, metallic, insulating, would affect the number of tile types required, and the folding sequence, such that the final structure has functional placement of the materials as well as a specified structure.

I feel fortunate at the end of my thesis to still be excited by the work and by the future of that work. I believe this is just the tip of the iceberg of logic based assembly.

Intentionally Blank Page