Programming Biological Cells

Ron Weiss, George Homsy, Radhika Nagpal *

{rweiss,ghomsy,radhi}@ai.mit.edu

MIT Laboratory for Computer Science and Artificial Intelligence

It appears that biological organisms can be harnessed as substrates for computation. Biological cells possess important characteristics, such as energy efficiency, self-reproduction, and miniature scale, that make them attractive for many applications. Examples include embedded intelligence in materials, sensing, smart medicine, and nanoscale fabrication. Vast numbers of programmed cells executing in parallel will enable cheap computation. This abstract argues that individual cells are programmable, and presents a programming paradigm for colonies of cells.

Cellular Gates

A fundamental chemical process in the cell is the production of proteins from genes encoded in the DNA. The cell performs important regulatory activities through DNA-binding proteins that repress the production of specific proteins. [2] proposes using this regulatory mechanism to implement digital logic inverters. This idea can be extended to construct complex digital logic, making the cell a self-contained computational unit.

Microbial circuit design creates a biological digital circuit using a small set of basic gates and a database of protein kinetic rates. To prevent interference between the gates, a different protein is used for each unique signal. Therefore, the number of proteins needed to implement a circuit is proportional to the complexity of the circuit. The design process requires searching the database and assigning suitable proteins to each gate, where the dynamic characteristics of the gate may differ depending on the proteins chosen. The basic gates must be robust enough to function with a wide variety of kinetically different proteins.

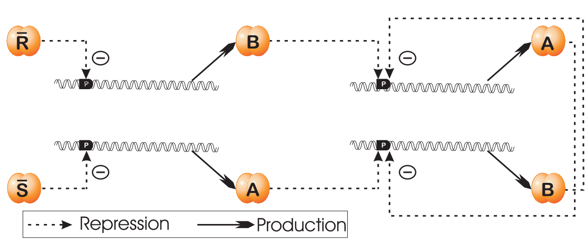

Figure 1: Genes and Proteins of a Biological $\overline{R}\cdot\overline{S}$ Latch

We are currently implementing a digital abstraction in *E. coli* bacteria. A model of the chemical dynamics of digital logic was developed using characteristics of known proteins encoded by the lambda-phage virus. Several basic gates have been designed, such as a biological RS-latch for storing state. The latch uses two input proteins for $\overline{\texttt{set}}$ and $\overline{\texttt{reset}}$, and cross couples the output proteins A and B to maintain state (Figure 1). The simulated dynamic behavior of this gate shows that a biological RS-latch is plausible (Figure 2). In parallel with the modeling effort we are building a ring oscillator in the lab, creating the first bacterial digital circuit.

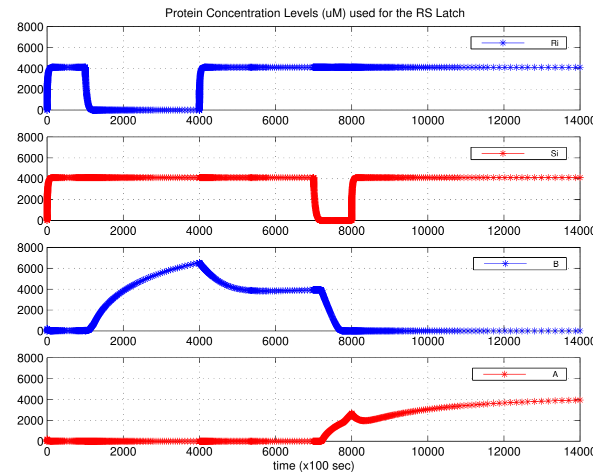

Figure 2: Dynamic Behavior of a Biological $\overline{R}\cdot\overline{S}$ Latch

Programming a Colony of Cells

Individual biological cells have limited computational capacity and frequently fail. However, by programming large colonies of cells that execute in parallel and coordinating their actions through intercellular communication, significant computational power can be achieved. The goal of the Amorphous Computing project [1] is to develop novel paradigms for programming such substrates.

The *Microbial Colony Language* is a programming paradigm simple enough for biological cells, yet expressive enough to implement interesting applications. The language exposes programming mechanisms that biological cells can perform reliably. The program for a single cell comprises event-triggered rules, boolean state, boolean operations, and limited range chemical diffusion for communication. We have implemented protein-level and language-level simulators that model cell colonies executing microbial programs. Simulations show that these programs can produce large scale pattern generation and coordinated group behavior.

The endeavor to create programmable biological systems is only a first step toward expanding the role and nature of computation beyond traditional applications.

References

[1] Abelson, Knight, and Sussman. Amorphous computing. *White paper*, October 1995.

[2] Knight and Sussman. Cellular gate technology. In *Unconventional Models of Computation*, 1997.

* Authors are advised by Gerry Sussman, Hal Abelson, and Tom Knight, and are supported by DARPA under contract number N00014-96-1-1228 administered by ONR.