Title: Conway's Game of life in a grid array of *E coli* colonies

David Joy Dec 7th 2015

Introduction: Synthetic biology has allowed for the design of single-cell behaviors in response to external stimuli, creating sensors and actuators to perform many tasks such as toxin detection or cancer treatment. In addition, whole populations of cells have been engineered to perform signal propagation tasks such as oscillations through quorum sensing signals. However, in order to perform more complex tasks, it would be useful to engineer spatially and temporally programmable behavior into cell populations, ideally without generating hundreds of distinct cell lines.

Cellular automata are a class of simple rule driven finite state machines which output a signal in response to an internal state and the state of their immediate neighbors. Despite running an identical program, these systems can exhibit extremely complex, programmable behavior. Implementing this kind of circuit in living cells could be used to create programmable colonies, where multiple cells cooperate to detect and indicate environmental toxins, search for target organs, or coordinate the production of molecules. This project will demonstrate the feasibility of implementing a simple example of such a rule set in *E coli*.

The game of life[1] is a two dimensional cellular automata system where cells are arranged in a grid and change their internal ON/OFF state in response to the state of their eight surrounding neighbors. When less than two neighbors are ON, the cell turns OFF, as if due to starvation. When exactly two neighbors are ON, the cell maintains its current state. When three neighbors are ON, the cell turns ON. If any more than three neighbors are ON, the cell turns OFF, as if due to overcrowding. Despite these simple rules, this system is capable of general purpose computation[2], making it a good demonstration of the power of cellular automata to generate spatially programmable behavior.

Design: The game of life requires a strict arrangement of cells to produce programmable behavior, and the response changes drastically with different types of grids[3]. To ensure a square grid arrangement, colonies of *E. coli* will be grown in a grid of wells. Each well will be permeable to the class of quorum sensing molecules N-acyl homoserine lactone (AHL), commonly used for coordination in gram-negative bacteria[4], but will prevent the colonies from mixing. By measuring the concentration of AHL arriving in the local well, the cell will be able to decide on its next configuration. This arrangement is shown in figure 1.

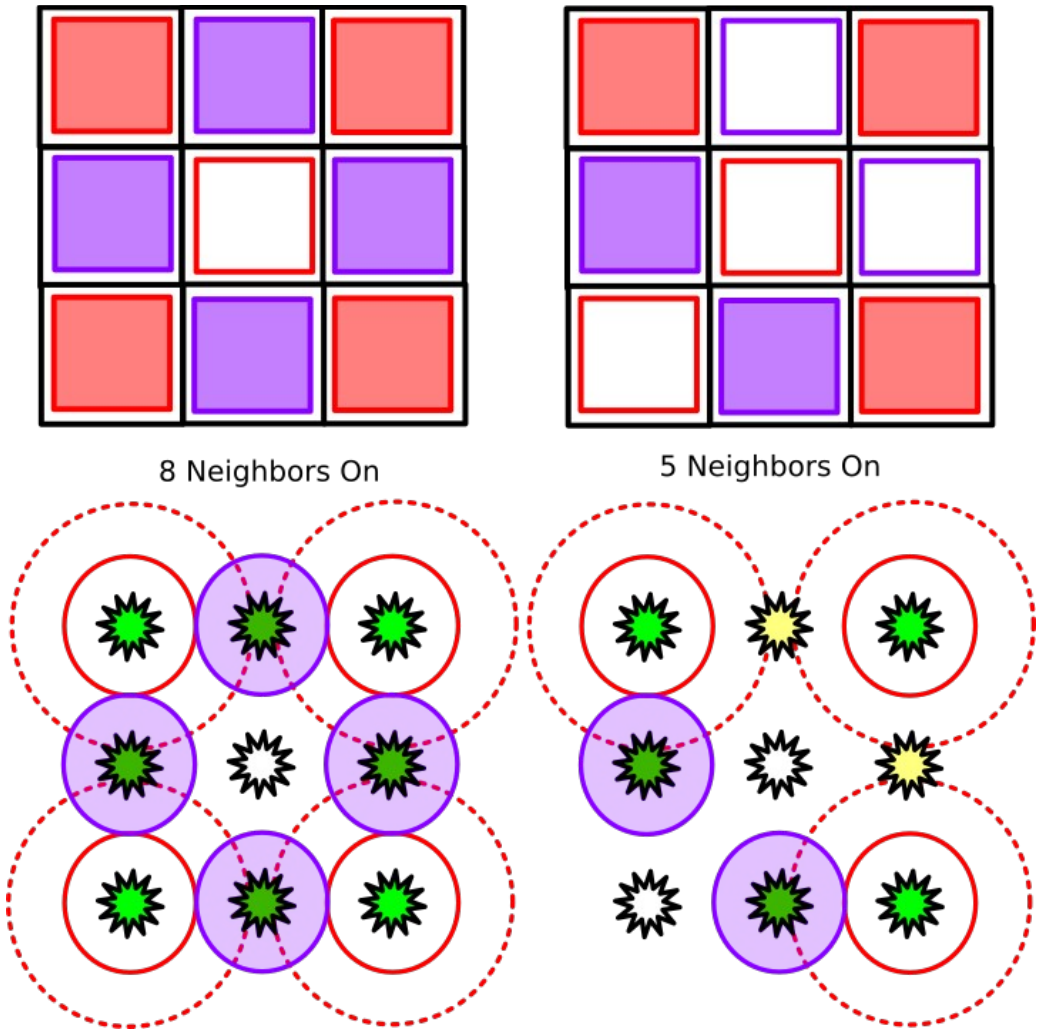


Figure 1: Grid arrangement of E. coli colonies. The center cell and the corners produce AHL2, while the vertical and horizontal cells produce AHL1.

Level detection of AHL significantly reduces the complexity of the genetic circuit compared to the implementation of a 3-bit adder circuit. Unfortunately, the AHL level reaching the center from the diagonal corners is expected to be different from the AHL level from a vertically or horizontally adjacent colony due to the ~1.4 increase in travel distance. To mitigate this, two different *E. coli* populations will be created, producing two different variants of AHL, but with otherwise identical circuits. These populations will sense AHL1 and AHL2 with level followers (the LuxI/LuxR and LasI/LasR systems respectively) tuned to remove this difference (as well as self-induction in the ON state), followed by an internal level detector circuit to count the final neighbor level.

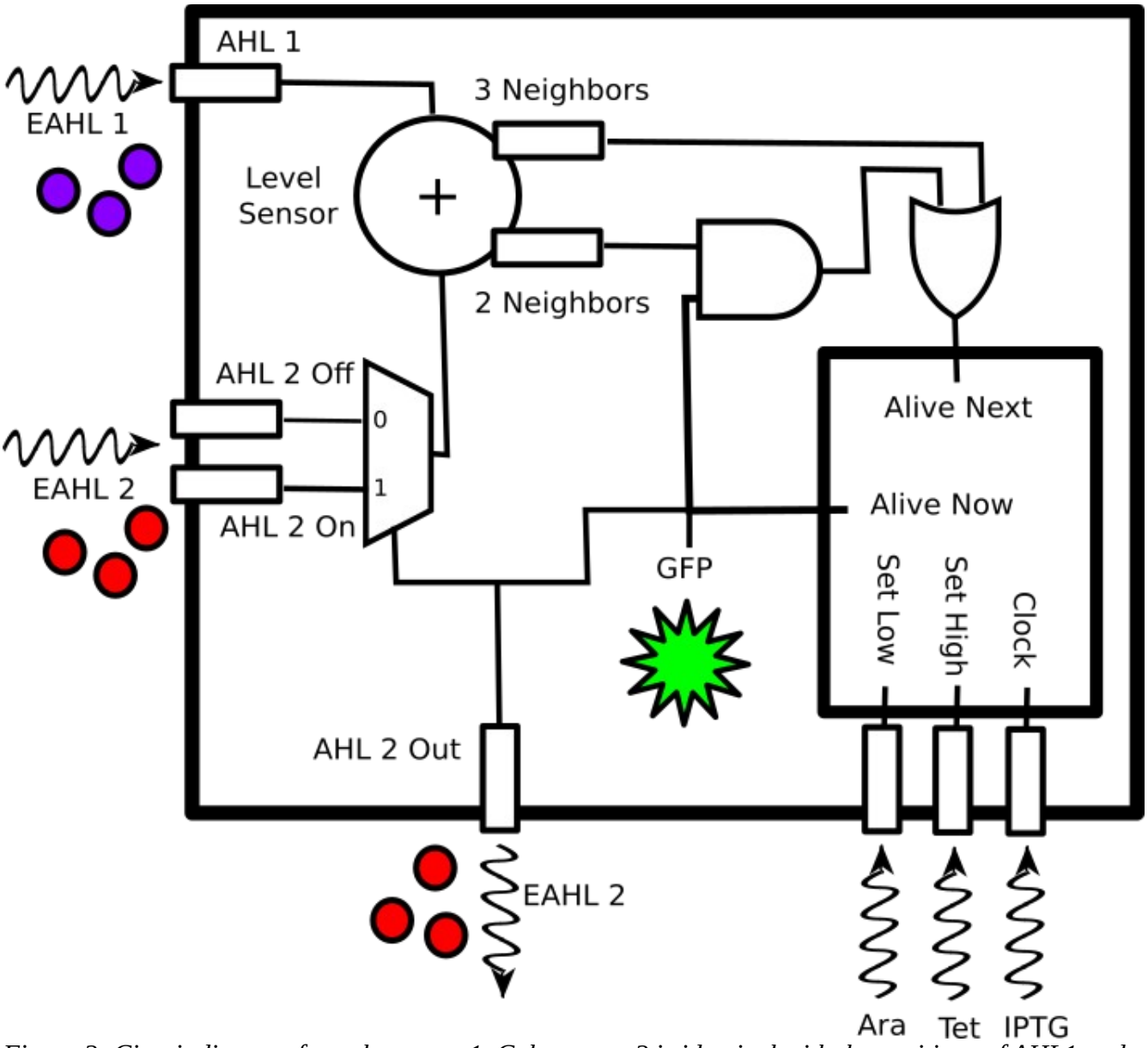


Figure 2: Circuit diagram for colony type 1. Colony type 2 is identical with the positions of AHL1 and AHL2 reversed.

Given a sensor that detects if exactly two neighbors are ON (R_{lvl2}) and a sensor that detects if exactly three neighbors are ON (R_{lvl3}), and the cell's current state (R_{cur}), the cell's next state (R_{next}) is entirely given by the logical expression OR(R_{lvl3},AND(R_{lvl2},R_{cur})). To give the colony memory, this value will be stored in a bistable switch when the IPTG small molecule inducer (R_{clk}) is applied to the system in a pulse. This clock signal could potentially be generated by the colony, but is modeled as an external input for simplicity. To allow the system to be programmed, the additional inducers tetracycline (R_{sl}) and arabinose (R_{sh}) can be used to force the bistable switch to the OFF or ON state regardless of neighbor input signal. This full circuit is given in figure 2.

Methods: The system is neatly divided into two parts: an analog level follower, and a digital state machine. To limit the scope of the analysis, the level follower was modeled separately, its analog output fed into the state machine, and then the dynamic output of the state machine was used to generate the spatial response. To avoid modeling RNA dynamics, which significantly increased simulation time, the

dynamics of transcription and translation were assumed to be fast compared to the dynamics of promoter binding and protein degradation. Further, promoter dynamics were assumed to be well characterized by Hill functions with normalized K_D of 1 and cooperativity coefficients n of 2. Finally, all proteins were assumed to decay at the same rate gamma.

With these assumptions, the level follower is modeled as a pair of activators that respond to the external signals EAHL1 and EAHL2 and both output AHL2. The gain required is modeled by assuming radial spatial diffusion with the output of the current cell normalized to $1^{5,}$, giving a required Kg for 1.513. Because an ON cell also also produces EAHL2, there are two EAHL2 sensors, one which is repressed by the ON state of the cell which is tuned for EAHL2 without autoinduction, and one which is activated by the ON state of the cell and is tuned for EAHL2 concentrations with autoinduction. Since both followers output the same protein, the combined concentration AHL2 measures the ON state of all eight neighbors (figure 3).

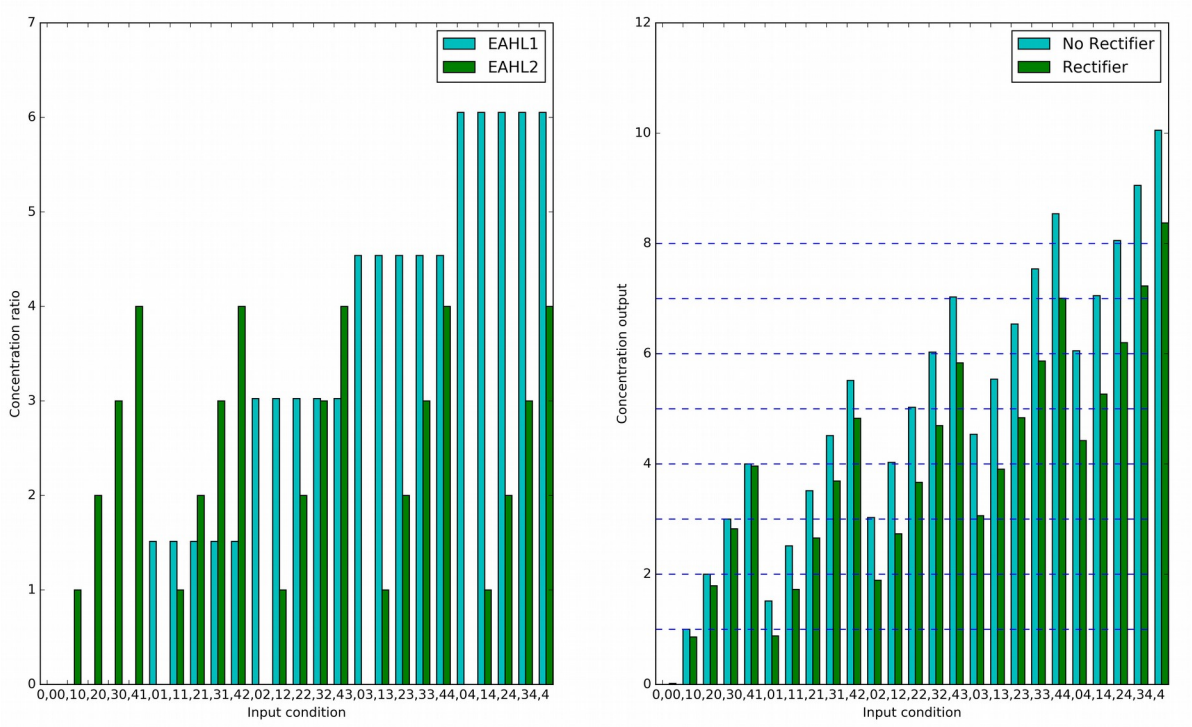


Figure 3: Input signal level follower/rectifier for EAHL1 and EAHL2

Since this system has no dynamic behavior other than as a buffer, and to reduce the number of variables in the equation solver, the EAHL1 and EAHL2 levels were translated to an internal AHL2 level. This level was treated as an input to the state machine level sensors. Using the same fast RNA dynamic assumptions as described above, equations 1-5 give the following models for the four generic components in the digital state machine.

NOT Gate

$$\frac{dY}{dt} = -Y\gamma + Kg_a \frac{Kr_a^{N_a}}{Kr_a^{N_a} + A^{N_a}} \tag{1}$$

NOR Gate

$$\frac{dY}{dt} = -Y\gamma + Kg_a Kg_b Kg_c \frac{Kr_a^{N_a}}{Kr_a^{N_a} + A^{N_a}} \frac{Kr_b^{N_b}}{Kr_b^{N_b} + B^{N_b}} \frac{Kr_c^{N_c}}{Kr_c^{N_c} + C^{N_c}} \tag{2}$$

Bistable Latch

$$\frac{dY}{dt} = -Y\gamma + Kg_x \frac{Kr_x^{N_x}}{Kr_x^{N_x} + X^{N_x}} \tag{3}$$

$$\frac{dX}{dt} = -X\gamma + Kg_y \frac{Kr_y^{N_y}}{Kr_y^{N_y} + Y^{N_y}} \tag{4}$$

Level Sensor

$$\frac{dY}{dt} = -Y\gamma + Kg_x \frac{Kr_x^{N_x}}{Kr_x^{N_x} + X^{N_x}} \frac{X^{N_x}}{Ka_x^{N_x} + X^{N_x}} \tag{5}$$

The state machine shown in figure 2 was modeled as a truth table (appendix 1), reduced to a product-of-sums equation and then further reduced to a form containing only NOT and NOR gates for implementation as a repressor network (figure 4). Each of these elementary components was modeled as a generic activator or repressor binding to a promoter (figure 5). Unrolling this network and applying equations 1-5 to translate the logic gates to Hill functions gives the system of 12 differential equations in appendix 2.

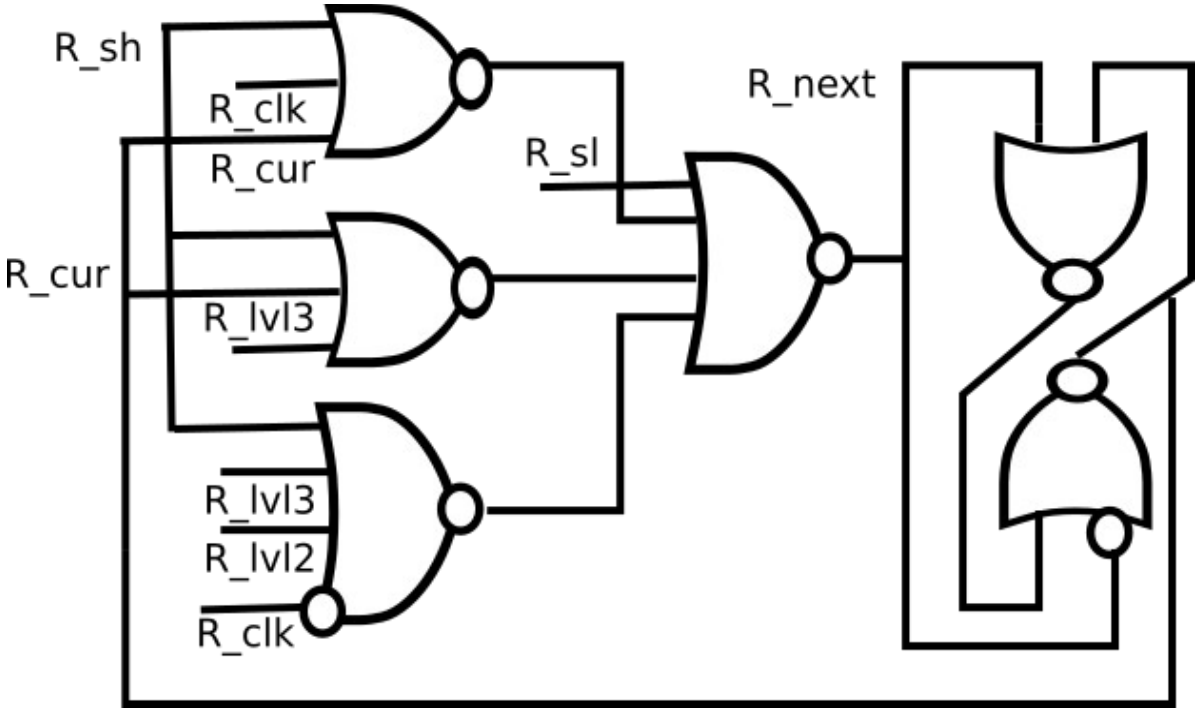


Figure 4: Gate network for the clock triggered latch

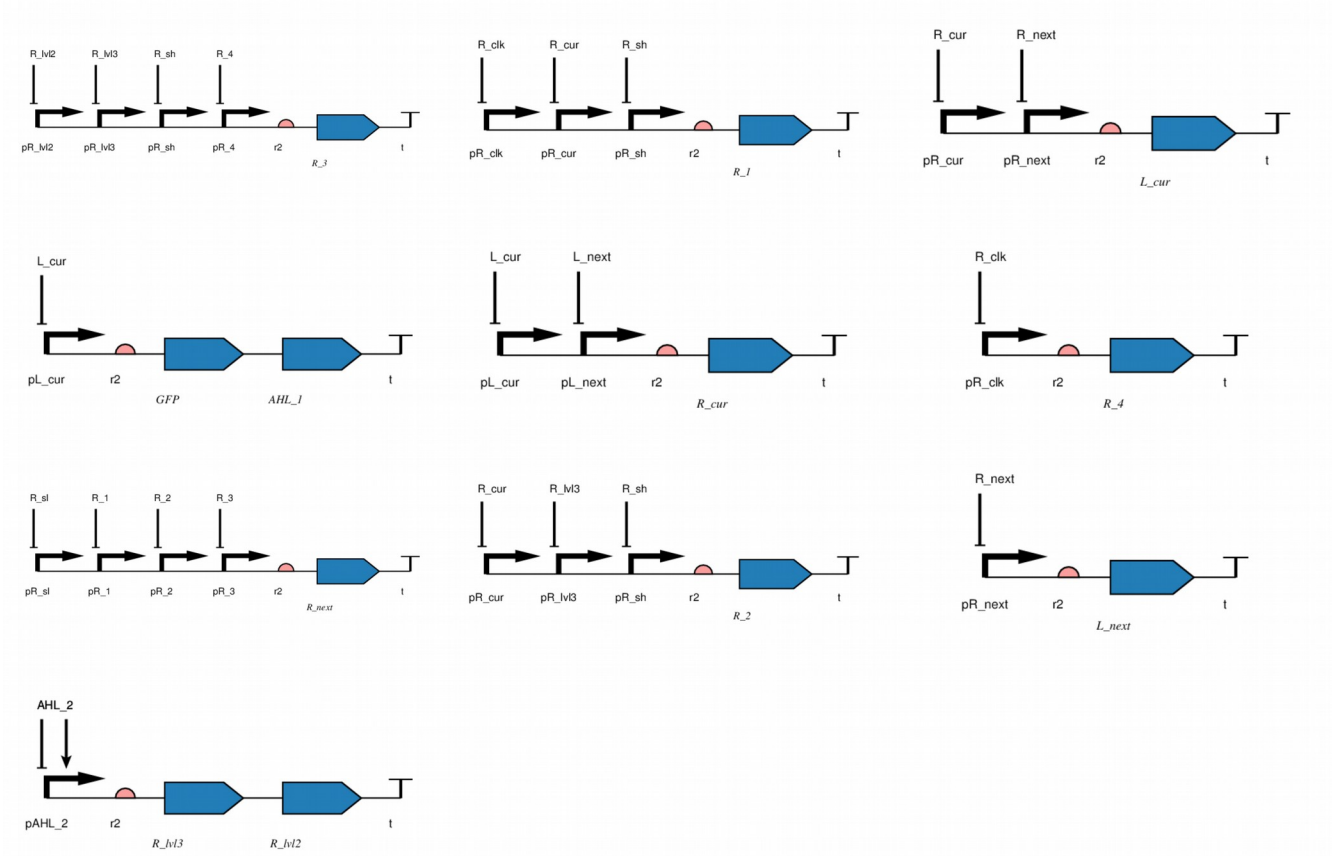
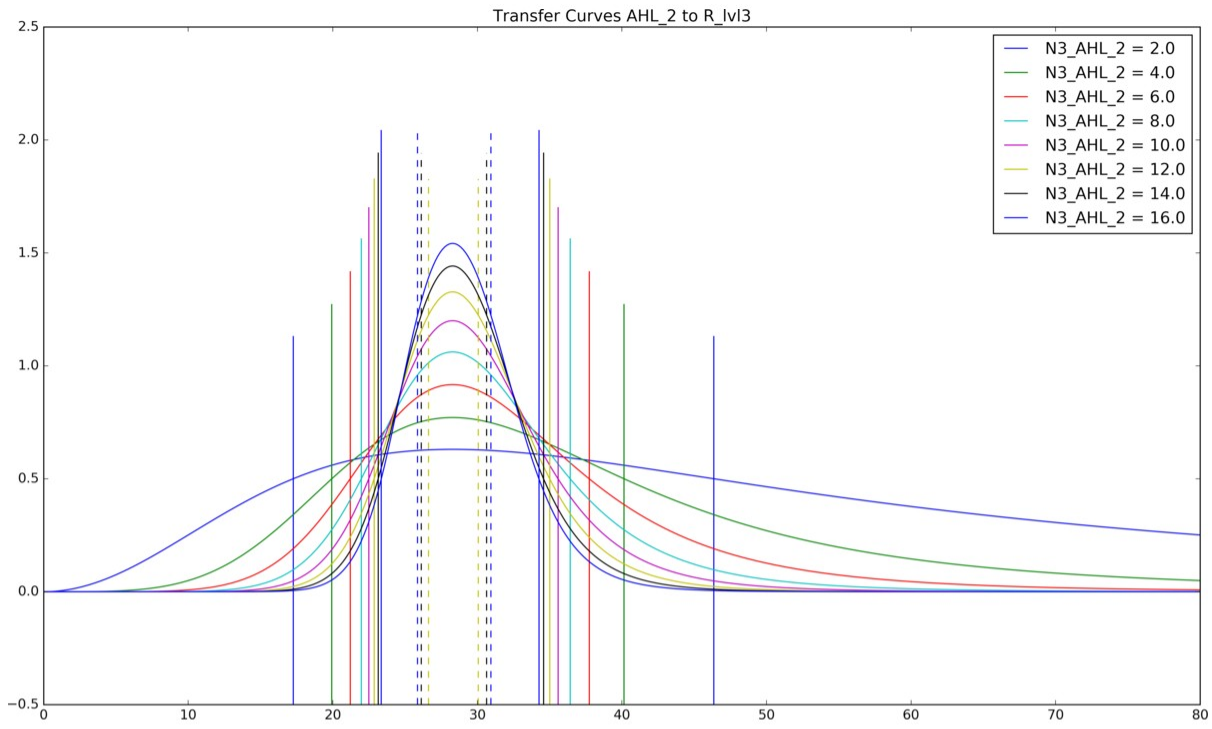


Figure 5: Gene cassettes for the game of life created in PigeonCAD[6]

Results: The system was shown to be able to simulate the game of life if the level detectors were capable of differentiating between the one neighbor, two neighbor, three neighbor, and four neighbor states. To calibrate the level sensors, the response curves were plotted for R_{lvl2} and R_{lvl3} to attempt to minimize the cooperativity needed to detect the correct level and attenuate all other levels. Due to the proximity of levels 2 and 3, a very sharp cooperativity of n=12 was required to prevent state transitions in the one neighbor and four neighbor conditions. In addition, the level sensor gain Kg was critical to ensuring a strong enough response at the peak input concentration. The responses for the level sensor at different n and Kg are shown in figure 6.



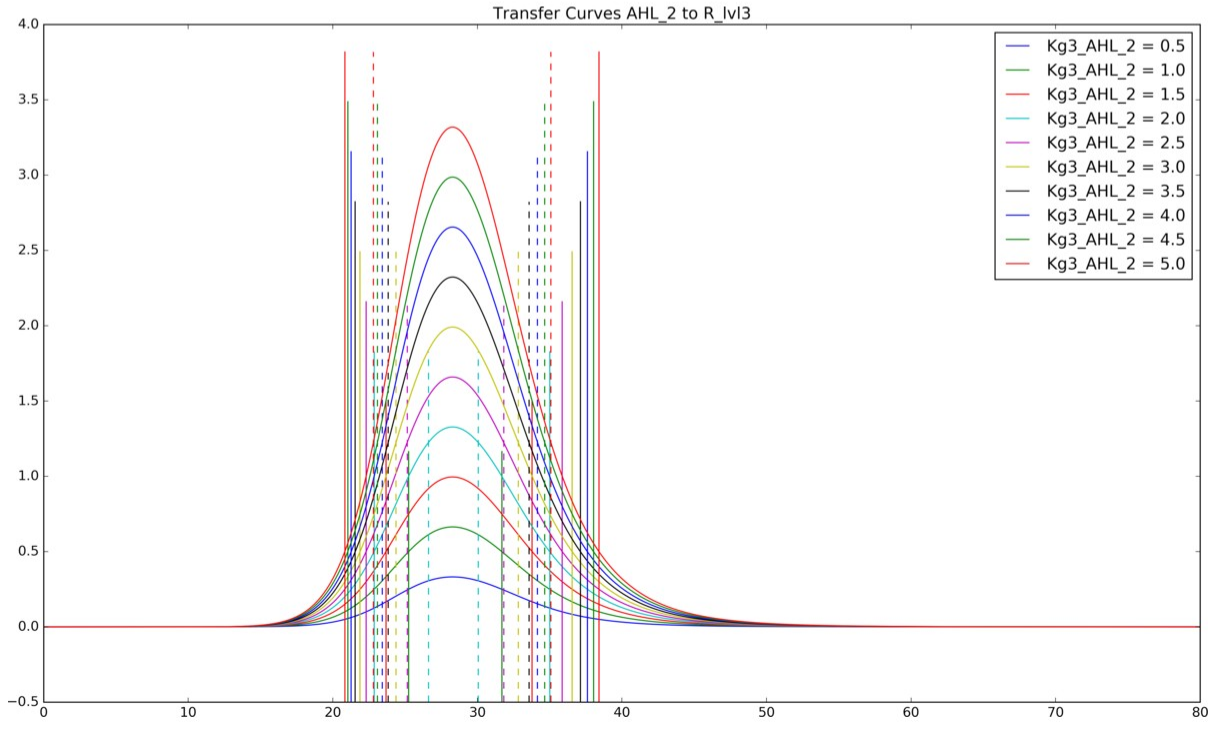


Figure 6: Sweeping input parameters Kg3 and N3. When N3 < 12, the peak is not sharp enough to trigger a response. When Kg3 < 2 or > 3, the transfer lines between OFF and ON begin to overlap.

To allow the analysis to proceed, it was assumed that suitable level sensors could be constructed. The gate network was simulated starting from both the OFF and ON state under all input conditions, with the input applied for 100 time steps, followed by 100 time steps of no input. The final state of the system was compared to the expected state for that stimulus. All transitions were found to be viable with the default parameters of Kd = 1 and n = 2. The time courses resulting from these simulations were used to produce the output plots in figure 7.

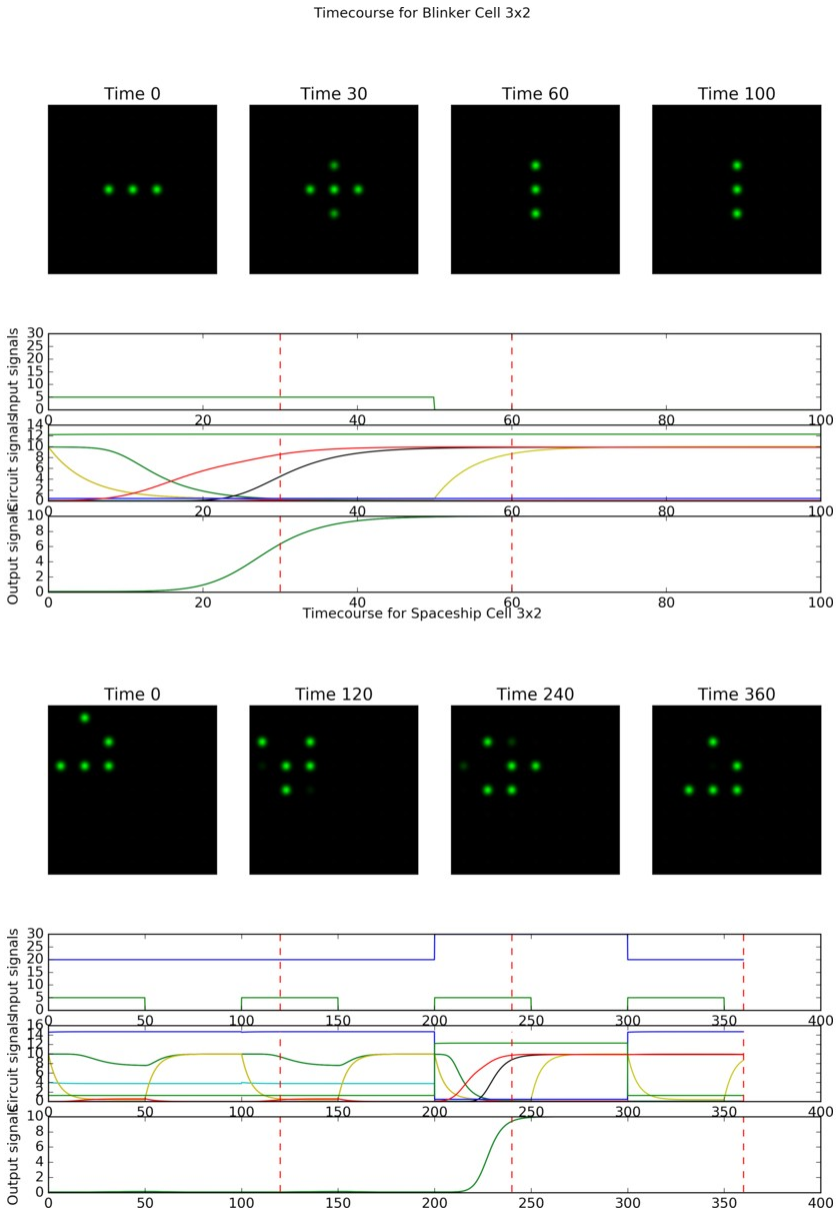


Figure 7: Simulated time courses for a blinker (top) and a spaceship (bottom)

Conclusions: This project demonstrated the plausibility of constructing a game of life state machine in *E. coli* using a quorum sensing system. Although real binding coefficients were not used in the simulation, the cooperativity required to process the logical aspects of the game of life seems within the bounds of commonly used repressor/promoter pairs. Further optimization could be made by using a large library of repressors and an automated gate matching technique[7], so it is expected that this circuit could be synthesized for real and would likely function similarly to the presented model.

This project also demonstrated a potential failure point in the design of a level detector using an activator/repressor bandpass filter. Especially when discriminating between close concentration levels, these systems have poor response characteristics and are very sensitive to Kg, Kd, and most problematically n. It is unlikely that in-vivo circuits would have such a narrow range of variability, or that activator/promotor pairs could be found with the precise character required by the level sensors in the model. Although it was avoided for reasons of complexity, it is possible that a more traditional analog to digital converter circuit could be implemented with more forgiving parameters.

References:

[1] Gardner, M. Mathematical games: The fantastic combinations of John Conway's new solitaire game 'life'. *Scientific American* **223,** 120–123 (1970).
[2] Rendell, P. A Turing machine in Conway's game Life. *A Turing machine in Conway's game Life* (2015). at <http://rendell-attic.org/gol/tm.htm>
[3] Bays, C. A note on the Game of Life in hexagonal and pentagonal tessellations. *Complex Systems* **15,** 245–252 (2005).
[4] Telford, G. *et al.* The Pseudomonas aeruginosa Quorum-Sensing Signal MoleculeN-(3-Oxododecanoyl)-l-Homoserine Lactone Has Immunomodulatory Activity. *Infection and Immunity* **66,** 36–42 (1998).
[5] Danino, T., Mondragón-Palomino, O., Tsimring, L. & Hasty, J. A synchronized quorum of genetic clocks. *Nature* **463,** 326–330 (2010).
[6] Bhatia, S. & Densmore, D. Pigeon: A Design Visualizer for Synthetic Biology. *ACS Synthetic Biology* **2,** 348–350 (2013).
[7] Huynh, L., Tsoukalas, A., Köppe, M. & Tagkopoulos, I. SBROME: a scalable optimization and module matching framework for automated biosystems design. *ACS synthetic biology* **2,** 263–273 (2013).

Appendix 1: Truth table for the game of life

R_lvl2	R_lvl3	R_clk	R_sl	R_sh	R_cur	R_next
0	0	0	0	0	0	0
0	0	0	0	0	1	1
0	0	0	0	1	0	1
0	0	0	0	1	1	1
0	0	0	1	0	0	0
0	0	0	1	0	1	0
0	0	0	1	1	0	0
0	0	0	1	1	1	0
0	0	1	0	0	0	0
0	0	1	0	0	1	0
0	0	1	0	1	0	1
0	0	1	0	1	1	1
0	0	1	1	0	0	0
0	0	1	1	0	1	0
0	0	1	1	1	0	0
0	0	1	1	1	1	0
0	1	0	0	0	0	0
0	1	0	0	0	1	1
0	1	0	0	1	0	1
0	1	0	0	1	1	1
0	1	0	1	0	0	0
0	1	0	1	0	1	0
0	1	0	1	1	0	0
0	1	0	1	1	1	0
0	1	1	0	0	0	1
0	1	1	0	0	1	1
0	1	1	0	1	0	1
0	1	1	0	1	1	1
0	1	1	1	0	0	0
0	1	1	1	0	1	0
0	1	1	1	1	0	0
0	1	1	1	1	1	0
1	0	0	0	0	0	0
1	0	0	0	0	1	1
1	0	0	0	1	0	1
1	0	0	0	1	1	1
1	0	0	1	0	0	0
1	0	0	1	0	1	0
1	0	0	1	1	0	0
1	0	0	1	1	1	0
1	0	1	0	0	0	0
1	0	1	0	0	1	1
1	0	1	0	1	0	1
1	0	1	0	1	1	1
1	0	1	1	0	0	0
1	0	1	1	0	1	0
1	0	1	1	1	0	0
1	0	1	1	1	1	0
1	1	0	0	0	0	0
1	1	0	0	0	1	1
1	1	0	0	1	0	1
1	1	0	0	1	1	1
1	1	0	1	0	0	0
1	1	0	1	0	1	0
1	1	0	1	1	0	0
1	1	0	1	1	1	0
1	1	1	0	0	0	1
1	1	1	0	0	1	1
1	1	1	0	1	0	1
1	1	1	0	1	1	1
1	1	1	1	0	0	0
1	1	1	1	0	1	0
1	1	1	1	1	0	0
1	1	1	1	1	1	0

Appendix 2: System of equations for the game of life simulation

$$\frac{dAHL_1}{dt} = -AHL_1\gamma_{deg} + \frac{Kg_{Lcur}Kr_{Lcur}^{N_{Lcur}}}{Kr_{Lcur}^{N_{Lcur}} + L_{cur}^{N_{Lcur}}} \tag{6}$$

$$\frac{dGFP}{dt} = -GFP\gamma_{deg} + \frac{Kg_{Lcur}Kr_{Lcur}^{N_{Lcur}}}{Kr_{Lcur}^{N_{Lcur}} + L_{cur}^{N_{Lcur}}} \tag{7}$$

$$\frac{dL_{cur}}{dt} = \frac{Kg_{Rcur}Kg_{Rnext}Kr_{Rcur}^{N_{Rcur}}Kr_{Rnext}^{N_{Rnext}}}{\left(Kr_{Rcur}^{N_{Rcur}} + R_{cur}^{N_{Rcur}}\right)\left(Kr_{Rnext}^{N_{Rnext}} + R_{next}^{N_{Rnext}}\right)} - L_{cur}\gamma_{deg} \tag{8}$$

$$\frac{dL_{next}}{dt} = \frac{Kg_{Rnext}Kr_{Rnext}^{N_{Rnext}}}{Kr_{Rnext}^{N_{Rnext}} + R_{next}^{N_{Rnext}}} - L_{next}\gamma_{deg} \tag{9}$$

$$\frac{dR_1}{dt} = \frac{Kg_{Rclk}Kg_{Rcur}Kg_{Rsh}Kr_{Rclk}^{N_{Rclk}}Kr_{Rcur}^{N_{Rcur}}Kr_{Rsh}^{N_{Rsh}}}{\left(Kr_{Rclk}^{N_{Rclk}} + R_{clk}^{N_{Rclk}}\right)\left(Kr_{Rcur}^{N_{Rcur}} + R_{cur}^{N_{Rcur}}\right)\left(Kr_{Rsh}^{N_{Rsh}} + R_{sh}^{N_{Rsh}}\right)} - R_1\gamma_{deg} \tag{10}$$

$$\frac{dR_2}{dt} = \frac{Kg_{Rcur}Kg_{Rlvl3}Kg_{Rsh}Kr_{Rcur}^{N_{Rcur}}Kr_{Rlvl3}^{N_{Rlvl3}}Kr_{Rsh}^{N_{Rsh}}}{\left(Kr_{Rcur}^{N_{Rcur}} + R_{cur}^{N_{Rcur}}\right)\left(Kr_{Rlvl3}^{N_{Rlvl3}} + R_{lvl3}^{N_{Rlvl3}}\right)\left(Kr_{Rsh}^{N_{Rsh}} + R_{sh}^{N_{Rsh}}\right)} - R_2\gamma_{deg} \tag{11}$$

$$\frac{dR_3}{dt} = \frac{Kg_{R4}Kg_{Rlvl2}Kg_{Rlvl3}Kg_{Rsh}Kr_{R4}^{N_{R4}}Kr_{Rlvl2}^{N_{Rlvl2}}Kr_{Rlvl3}^{N_{Rlvl3}}Kr_{Rsh}^{N_{Rsh}}}{\left(Kr_{R4}^{N_{R4}} + R_4^{N_{R4}}\right)\left(Kr_{Rlvl2}^{N_{Rlvl2}} + R_{lvl2}^{N_{Rlvl2}}\right)\left(Kr_{Rlvl3}^{N_{Rlvl3}} + R_{lvl3}^{N_{Rlvl3}}\right)\left(Kr_{Rsh}^{N_{Rsh}} + R_{sh}^{N_{Rsh}}\right)} - \tag{12}$$

$$\frac{dR_4}{dt} = \frac{Kg_{Rclk}Kr_{Rclk}^{N_{Rclk}}}{Kr_{Rclk}^{N_{Rclk}} + R_{clk}^{N_{Rclk}}} - R_4\gamma_{deg} \tag{13}$$

$$\frac{dR_{cur}}{dt} = \frac{Kg_{Lcur}Kg_{Lnext}Kr_{Lcur}^{N_{Lcur}}Kr_{Lnext}^{N_{Lnext}}}{\left(Kr_{Lcur}^{N_{Lcur}} + L_{cur}^{N_{Lcur}}\right)\left(Kr_{Lnext}^{N_{Lnext}} + L_{next}^{N_{Lnext}}\right)} - R_{cur}\gamma_{deg} \tag{14}$$

$$\frac{dR_{lvl2}}{dt} = \frac{AHL_2^{N_{2AHL2}}Kg_{2AHL2}Kr_{2AHL2}^{N_{2AHL2}}}{\left(AHL_2^{N_{2AHL2}} + Ka_{2AHL2}^{N_{2AHL2}}\right)\left(AHL_2^{N_{2AHL2}} + Kr_{2AHL2}^{N_{2AHL2}}\right)} - R_{lvl2}\gamma_{deg} \tag{15}$$

$$\frac{dR_{lvl3}}{dt} = \frac{AHL_2^{N_{3AHL2}}Kg_{3AHL2}Kr_{3AHL2}^{N_{3AHL2}}}{\left(AHL_2^{N_{3AHL2}} + Ka_{3AHL2}^{N_{3AHL2}}\right)\left(AHL_2^{N_{3AHL2}} + Kr_{3AHL2}^{N_{3AHL2}}\right)} - R_{lvl3}\gamma_{deg} \tag{16}$$

$$\frac{dR_{next}}{dt} = \frac{Kg_{R1}Kg_{R2}Kg_{R3}Kg_{Rsl}Kr_{R1}^{N_{R1}}Kr_{R2}^{N_{R2}}Kr_{R3}^{N_{R3}}Kr_{Rsl}^{N_{Rsl}}}{\left(Kr_{R1}^{N_{R1}} + R_1^{N_{R1}}\right)\left(Kr_{R2}^{N_{R2}} + R_2^{N_{R2}}\right)\left(Kr_{R3}^{N_{R3}} + R_3^{N_{R3}}\right)\left(Kr_{Rsl}^{N_{Rsl}} + R_{sl}^{N_{Rsl}}\right)} - R_{next} \tag{17}$$