The Design of a Genetic Muller C-Element*

Nam-Phuong D. Nguyen, Hiroyuki Kuwahara, Chris J. Myers, James P. Keener
University of Utah
Salt Lake City, UT 84112, USA
{namphuon, kuwahara, myers}@vlsigroup.ece.utah.edu, keener@math.utah.edu

Abstract

Synthetic biology uses engineering principles to design circuits out of genetic materials that are inserted into bacteria to perform various tasks. While synthetic combinational Boolean logic gates have been constructed, there are many open issues in the design of sequential logic gates. One such gate common in most asynchronous circuits is the Muller C-element, which is used to synchronize multiple independent processes. This paper proposes a novel design for a genetic Muller C-element using transcriptional regulatory elements. The design of a genetic Muller C-element enables the construction of virtually any asynchronous circuit from genetic material. There are, however, many issues that complicate designs with genetic materials. These complications result in modifications being required to the normal digital design procedure. This paper presents two designs that are logically equivalent to a Muller C-element. Mathematical analysis and stochastic simulation, however, show that only one functions reliably.

1 Introduction

Synthetic biology is the study of genetically engineering new biological pathways that can potentially change the behavior of organisms in useful ways. For example, bacteria can be induced to synthesize proteins used in creating drugs by inserting the necessary genes into the bacteria's genome. For example, a new metabolic pathway is being inserted in *E. coli* to synthesize *artemisinic acid*, the key component of the antimalarial drug artemisinin [14]. This method may soon lead to the production of cheap and effective drugs to combat malaria. Another example is in environmental cleanup. Scientists have been working on modifying bacteria to metabolize toxic chemicals and break down the toxins into less environmentally damaging forms [3, 4]. Synthetic biology can also help us better understand how microorganisms function by allowing us to examine how a synthetic pathway behaves *in vivo* as compared with *in silico* [16]. To support synthetic biology, MIT has even created a registry of standard biological parts [1].

Synthetic biology requires the construction of new *genetic circuits*. Genetic circuits are biological circuits constructed from DNA. A vital part of synthetic biology is developing methods and tools to assist in the design of these genetic circuits. While a schematic capture tool has been developed [9], there are currently no tools for the synthesis of genetic circuits. Due to the absence of global clock, genetic circuits are inherently asynchronous. Therefore, it may be possible to adapt asynchronous analysis and synthesis methods to the design of genetic circuits. To this end, we have developed one of the most efficient temporal behavior analyzer, `Reb2Sac`, for the analysis of genetic circuits by leveraging ideas inspired by asynchronous circuit analysis [12]. This tool applies automatic abstraction methods to reduce the complexity of the model to facilitate efficient analysis, and it is used to analyze the genetic circuits described in this paper.

Our long-term goal is to adapt asynchronous synthesis and technology mapping tools to target a genetic circuit technology. To achieve this goal, it is essential to first develop a library of gates sufficient to design interesting genetic circuits. While synthetic combinational logic gates have been constructed, there are many open issues in the design of sequential logic gates. One such gate common in most asynchronous circuits is the Muller C-element, which is used to synchronize multiple independent processes. This paper proposes a novel design for a genetic Muller C-element using transcriptional regulatory elements. The design of a genetic Muller C-element enables the construction of virtually any asynchronous circuit from genetic material.

There are, however, many issues that complicate designs with genetic materials. For example, there is no signal isolation in genetic circuits. Any product used as a signal can diffuse throughout the entire cell (and sometimes outside the cell). Gates can never be reused due to the sig-

*This material is based upon work supported by the National Science Foundation under Grant No. 0331270.

13th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'07)
0-7695-2771-X/07 $20.00 © 2007 IEEE

nal isolation problem. This lack of signal isolation can also cause interference problems. If a molecule used in a circuit binds with another molecule in the host cell, then the circuit may behave incorrectly. Another complication is that to connect gates in a genetic circuit, the output protein of one gate must be the same as the input protein of the next gate which further limits gate choices. Finally, genetic gates often behave nondeterministically since fluctuation in transcription/translation rates is too strong to simply neglect, and proteins interact with the gates stochastically and the rates of these interactions are quite variable and usually unknown. These complications result in many modifications being required for the normal digital design procedure. This paper presents two designs that are logically equivalent to a Muller C-element, but mathematical analysis and stochastic simulation show that only one design functions reliably.

2 Genetic Circuit Basics

Figure 1(a) depicts a simple genetic circuit found in the λ phage virus. A genetic circuit is constructed from a strand of DNA. A coding sequence of DNA includes several key regions: *genes*, *promoters*, *operator sites*, *ribosome binding sites (RBS)*, and *terminators*. Genes are regions that code for a protein. In Figure 1(a), the genes are *CI* and *CII*. They code for proteins CI and CII, respectively. Promoters are regions on the DNA that the *RNA polymerase (RNAP)* binds to start transcribing the gene. Operator sites are regions to which proteins can bind to increase or decrease the affinity of the promoter. In Figure 1(a), the transcription of *CI* is *activated* by CII binding to the operator site of the gene *CI*, and transcription of *CII* is *repressed* by the CI *dimer* binding to the operator site of the *CII* gene. RNAP binds to the promoter and starts *transcription*. Transcription is the process in which a strand of *messenger RNA (mRNA)* is created from the DNA. The RBS is the binding site on the mRNA to which the ribosome binds to start *translation*. Translation is the process in which a protein is created from mRNA. Terminator sites signal the RNAP to stop transcription and unbind from the DNA strand. For more a technical review of genetic circuits, please see [10, 15].

The behavior of the genetic circuit shown in Figure 1(a) is as follows. Initially, there are no molecules of CI or CII in the host cell. In this state, transcription of *CII* is very active while transcription of *CI* is not very active as its transcription must be facilitated by a CII molecule bound to its operator site. As CII molecules begin to build up in the cell, CII activates the transcription of CI. As the CI molecules are produced, pairs of them bind together (i.e., *dimerize*) to form the CI dimer. A CI dimer can bind to the operator site for CII which represses the transcription of CII. Without further production of CII, the level of CII decreases due to degradation. This lowers the amount of CII available to activate CI production, so CI also begins to disappear. Eventually, only small amounts of CII and CI remain returning the circuit to its initial state in which the process begins to repeat. This process is familiar to any digital designer as a simple oscillator as shown in Figure 1(b). It is important to note though that due to the stochastic nature, this circuit does not behave in a perfect digital fashion.

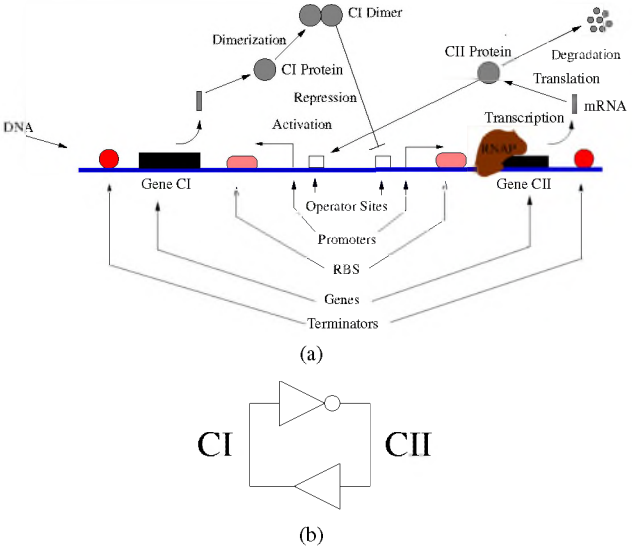


Figure 1. (a) Simple genetic circuit from the λ phage. (b) Digital representation.

3 Transcriptional Regulatory Gates

The inverter and buffer from the genetic circuit shown in Figure 1(a) are created from *transcriptional regulatory gates*. Transcriptional regulatory gates regulate transcription rates of genes by using activation or repression mechanisms at operator sites. In these gates, the proteins serving as *transcription factors* can be thought of as signals and the operator sites can be thought of as gate inputs. This section describes the genetic circuits for an inverter, a NAND gate, an AND gate, and a toggle switch which are used in our genetic Muller C-element design.

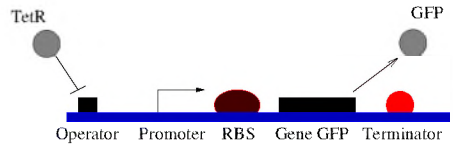
It is important to note that all of the examples shown use specific proteins and promoters, however, any protein-promoter pair with the same regulatory behavior can be interchanged with the example protein-promoter pairs. This allows us to build different versions of the same gates.

3.1 A genetic inverter

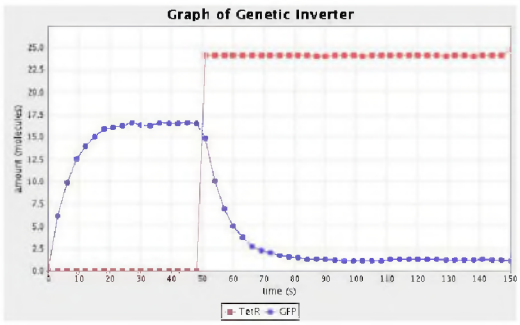
A genetic inverter can be constructed using repression as shown in Figure 2(a). The input of this inverter is a reg-

0-7695-2771-X/07 $20.00 © 2007 IEEE

ulatory protein called *TetR*. The output is a protein called *Green Fluorescent Protein* (GFP). GFP is used as an output because it fluoresces green when exposed to UV light, allowing it to be easily detected when the output signal is high. When the level of TetR is low, the level of GFP goes high, and vice versa. The simulation results for the genetic inverter are shown in Figure 2(b). The results are the average of 500 stochastic simulation runs. Initially, there is no TetR which causes the level of GFP to rise. At a time of 50 seconds, 25 molecules of TetR are added to the simulation which represses further production of GFP and causes the level of GFP to gradually fall due to degradation. This is just one example of a genetic inverter. We can replace TetR with any other protein-promoter pair with inhibitory behavior, and we can replace GFP with any other protein necessary to connect to another gate. This idea can be used with different input and output proteins to construct a variety of different inverter gates.



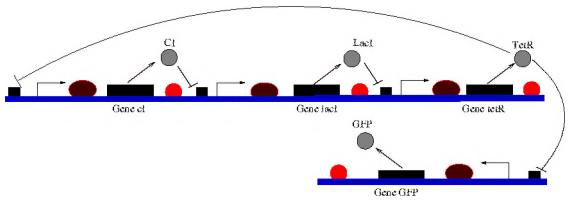
(a)



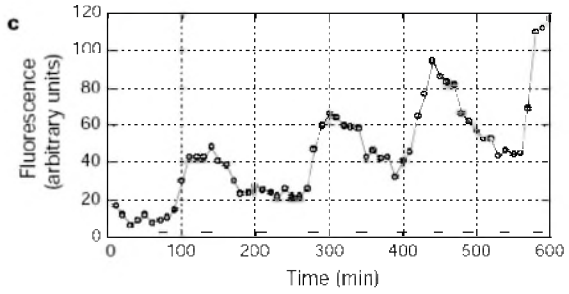
(b)

Figure 2. (a) A genetic inverter. (b) Simulation results.

Elowitz and Leibler used these ideas to create a three inverter ring oscillator as shown in Figure 3(a) [5]. This circuit is created from two sequences. The first sequence is built out of three genes in which the product of each gene represses the production of the next gene in the ring. The second sequence contains one reporter gene that is activated by the absence of TetR, a product from the first sequence. This sequence was inserted into *E. coli*, and the experimental results are shown in Figure 3(b) which clearly oscillate. Note that the gradual increase in fluorescence intensity is due to the increase in the *E. coli* population.



(a)



(b)

Figure 3. (a) A genetic oscillator. (b) Experimental results (courtesy of [5]).

3.2 A genetic NAND gate

A genetic NAND gate can be built using two inverters which output the same product as shown in Figure 4(a). The average of 500 stochastic simulation runs for this gate is shown in Figure 4(b). Initially, there is no TetR or LacI which allows the level of GFP to rise. At 100 seconds, 50 molecules of TetR are added causing the level of GFP to fall slightly, but remain high. At 375 seconds, 50 molecules of LacI are added causing GFP to fall to its low level.

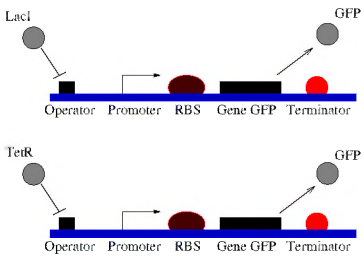
3.3 A genetic AND Gate

A genetic AND gate can be created using activation as shown in Figure 5(a). In this gate, LuxI is converted into $3OC_6HSL$ which binds with LuxR to form a complex. This complex activates the production of GFP. Therefore, both LuxI and LuxR must be present for GFP production to be activated. The average of 500 stochastic simulation runs for the AND gate is shown in Figure 5(b). When LuxR and LuxI levels are low, no GFP is produced. At 50 seconds, LuxR is added, but GFP is still not produced. At 100 seconds, LuxI is added resulting in the production of GFP.

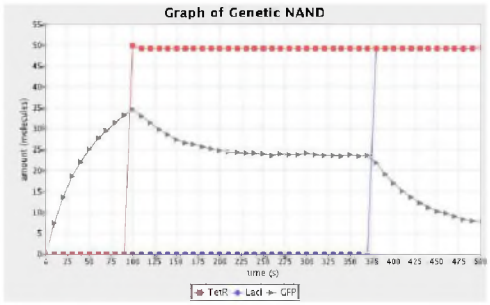
3.4 A toggle switch

In order to build sequential circuits, a state holding gate is necessary. In 2000, Gardner *et al.* created a genetic toggle

0-7695-2771-X/07 $20.00 © 2007 IEEE

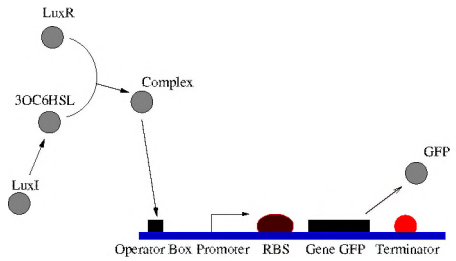


(a)

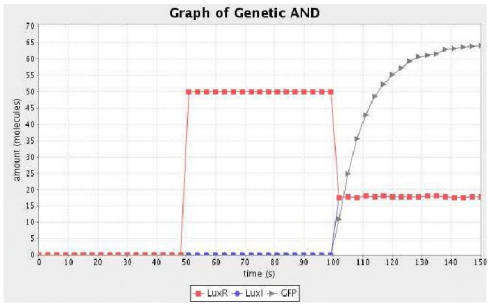


(b)

Figure 4. (a) A genetic NAND gate. (b) Simulation results.



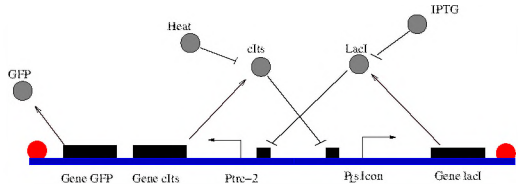
(a)



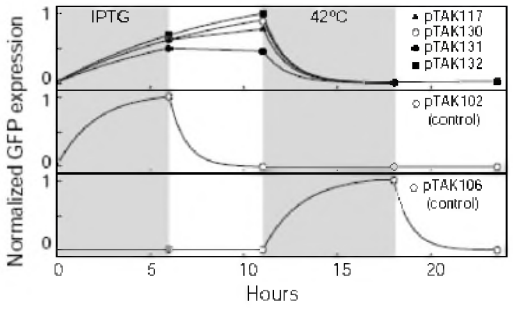
(b)

Figure 5. (a) A genetic AND gate. (b) Simulation results.

switch [1] using two genes that mutually repress each other [6]. Figure 6(a) shows the model for this toggle switch, and Figure 6(b) shows their experimental results. First, the inducer, IPTG, is added to the system which prevents LacI from repressing the production of cIts. This causes the levels of cIts and GFP to rise to their high levels setting the switch into its high state. When cIts is high, cIts blocks the $P_L s1con$ promoter which prevents LacI from being created. Therefore, even after IPTG is removed, the switch stays in its high state. When heat is introduced into the system, it inhibits cIts which allows LacI to be produced. The LacI produced represses the production of cIts and GFP changing the switch to its low state. When heat is removed from the system, LacI remains high so it holds this state.



(a)



(b)

Figure 6. (a) The genetic toggle switch. (b) Experimental results (courtesy of [6]).

4 A Genetic Muller C-Element

A Muller C-element is a state holding gate used to synchronize events in an asynchronous circuit. Its behavior is described by the truth table shown in Table 1. Namely, when both inputs A and B are low the output C should be low. When both inputs are high, the output should be high. Finally, if the inputs are different, the output should retain its old state. The input for both of our designs are heat and

[1] Technically, it is an SR latch, but this paper uses Gardner *et al.*'s original toggle switch name throughout.

0-7695-2771-X/07 $20.00

red light, and the output is GFP. Heat is defined as an increase from 30° C to 37° C. This inhibits the temperature dependent promoter [1] used in the design. Using heat and red light allow us to easily change the inputs, while using GFP allows us to easily detect to see if our design correctly works. Again, these are just potential instance of genetic Muller C-Element designs. Heat and red light can be replaced with any proteins-promoter pair with inhibitory behavior, and GFP can be replaced with any protein that may be necessary to connect to the next gate in the circuit.

Table 1. Truth table for a Muller C-element.

A	B	C
0	0	0
0	1	C
1	0	C
1	1	1

This section presents two designs of a genetic Muller C-element. The first is based upon a majority gate which is logically equivalent to a C-element, but it is shown not to work reliably. The second is based upon the toggle switch, and it is shown to perform much more reliably.

4.1 A majority gate design

The schematic for the majority gate C-element is shown in Figure 7(a), and its genetic circuit is shown in Figure 7(b). This gate uses heat and red light as inputs and GFP as output. The average of 500 stochastic simulation runs for this design is shown in Figure 7(c). Initially, no heat or red light is applied, and this results in a GFP level close to zero molecules. When heat is applied, the level of GFP remains unchanged near zero molecules. Next, red light is also applied, and the level of GFP rises to about 22 molecules. After heat is removed, however, the level of GFP drops to about two to three molecules. This intermediate state is so close to the low state that even small fluctuations can knock the gate into an incorrect state. Since this gate cannot reliably hold its state, it is a poor design for a Muller C-element.

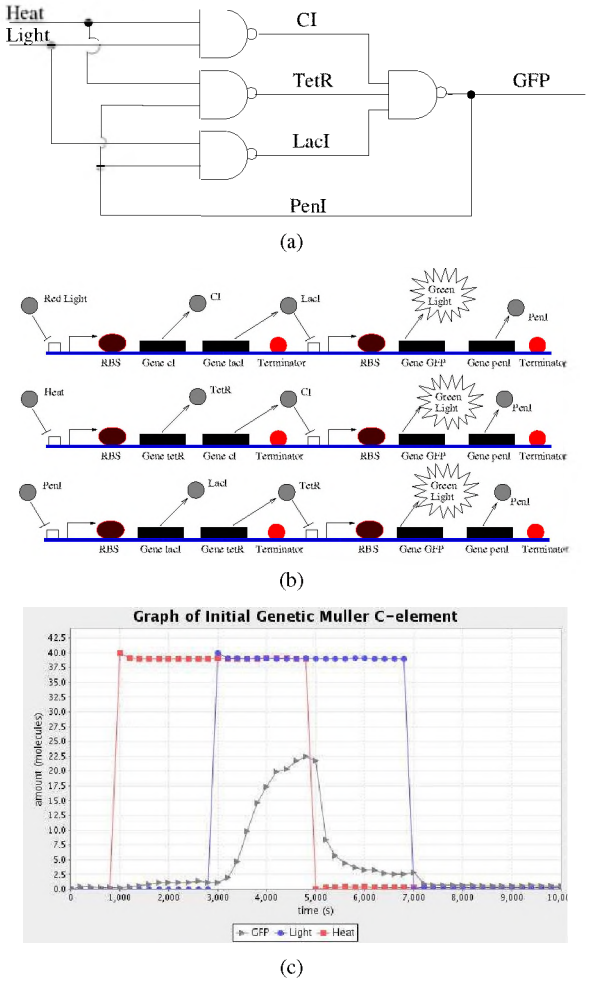


Figure 7. (a) Schematic, (b) genetic circuit, and (c) simulation results for a majority gate implementation of a genetic Muller C-element.

4.2 A toggle switch design

Our second C-element design utilizes Gardner *et al.*'s toggle switch. The logical model for this design is shown in Figure 8(a). In addition to the toggle switch, it uses inverters, a NAND gate, and an AND gate. The genetic circuit for this design is shown in Figure 8(b). All of the regulatory gates used in this design are built using parts found in MIT's Parts Library [1]. Figure 9 shows the simulation results for our redesigned genetic Muller C-element. Initially, no heat or red light is applied, and this results in a GFP level close to zero molecules. When heat is applied, the level of GFP remains very near zero molecules. Next, red light is also applied, and the level of GFP rises to about 25 molecules. After heat is removed, the level of GFP remains near 25 molecules. When red light is removed, the level of GFP falls back to zero molecules. With a separation value of about 25 molecules between the high and low states, this circuit is much more robust than the majority gate design which has a separation of only about two molecules. While not shown, the symmetric case where red light is removed first also holds state correctly.

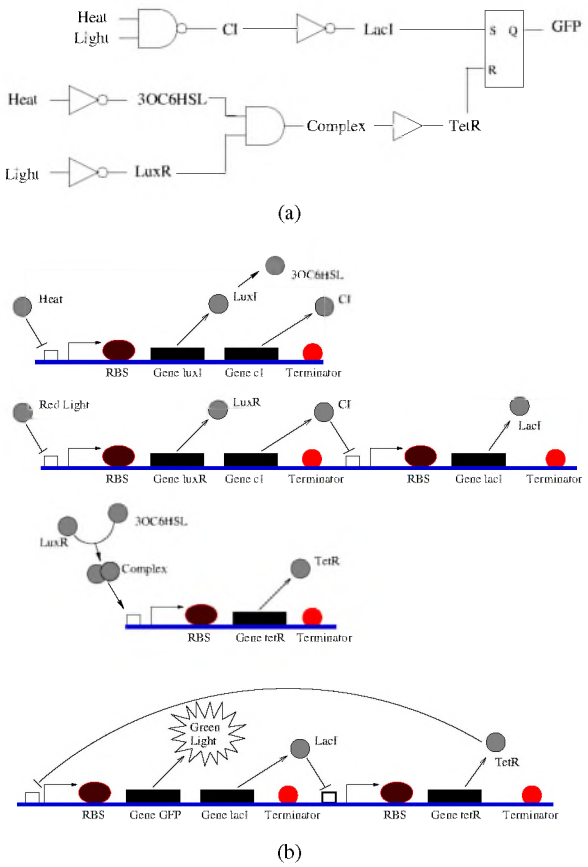


Figure 8. (a) Logical model for a toggle switch genetic Muller C-element. (b) Genetic circuit for the toggle switch C-element.

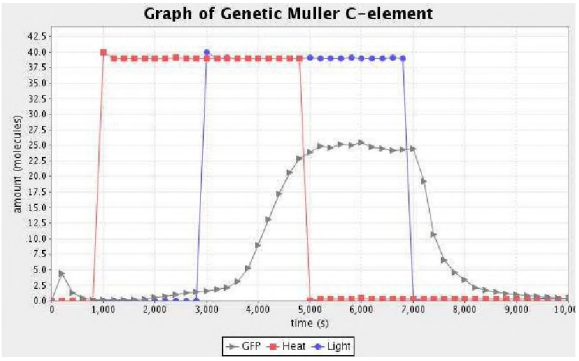


Figure 9. Simulation results for the toggle switch genetic Muller C-element.

5 Mathematical Analysis

While our simulations indicate that the toggle switch C-element seems to work correctly for a particular set of parameter values, a more rigorous mathematical analysis is necessary to determine if the results are robust to variations in the model parameters. In other words, we would like to determine whether the state holding behavior is a mathematical property of our circuit. Using the *Law of Mass Action*, we can derive a differential equation model of the toggle switch C-element which is shown in Equations 1-6. Note that, in this mathematical analysis, molecular populations are assumed to evolve continuously and deterministically. The variables used in the equations are shown in Table 2, and the parameters of the equations are shown in Table 3. For the heat input, we assume that the activity of the promoter hybB behaves as a linear decrease with increased temperature from 32°C [1]. Light inactivates EnvZ, a molecule that works in conjunction with OmpR-P and the promoter ompR [1]. Since light inactivates EnvZ which leads to the inactivation of the promoter ompR, we can model light as a repressor, even though physically, light does not have a molecule count.

$$\frac{da}{dt} = \frac{k_a K_{eqH}}{K_{eqH} + (H - H_0)} - d_a\, a + k_{cr}\, c - k_c\, ab \quad (1)$$

$$\frac{db}{dt} = \frac{k_b K_{eqL}}{K_{eqL} + L} - d_b\, b + k_{cr}\, c - k_c\, ab \quad (2)$$

$$\frac{dc}{dt} = k_c\, ab - k_{cr}\, c - d_c\, c \quad (3)$$

$$\frac{dx}{dt} = k_a\, e^{H_0 - H} + \frac{k_b\, K_{eqL}}{K_{eqL} + L} - d_x\, x \quad (4)$$

$$\frac{dy}{dt} = \frac{k_{y1}\, K_{eqX}}{K_{eqX} + x} + \frac{k_{y2} K^m_{eqZ}}{K^m_{eqZ} + z^m} - d_y\, y \quad (5)$$

$$\frac{dz}{dt} = k_{z1} \left(\frac{c}{K_{eqC} + c} \right) + \frac{k_{z2}\, K_{eqY}}{K^n_{eqY} + y^n} - d_z\, z \quad (6)$$

Table 2. Variables.

Variable	Name	Units
a	LuxI	*molecule*
b	LuxR	*molecule*
c	Complex	*molecule*
x	CI	*molecule*
y	LacI	*molecule*
z	TetR	*molecule*

Our analysis considers the system at *steady state* (i.e., a system state in which the rates of net change of all state variables are equal to zero). Equations 1-4 are not affected by Equations 5-6, so we can solve for the steady state solutions

0-7695-2771-X/07 $20.00

Table 3. Parameters.

Parameter	Biological Interpretation	Units
$k_a, k_b, k_{y1},$ k_{y2}, k_{z1}, k_{z2}	Rate of production.	$\frac{molecule}{sec}$
k_c	Rate of binding.	$\frac{1}{molecule \cdot sec}$
k_{cr}	Rate of disassociation.	$\frac{1}{molecule}$
$d_a, d_b, d_c,$ d_x, d_y, d_z	Rate of degradation.	$\frac{1}{sec}$
$K_{eqL}, K_{eqH},$ K_{eqY}, K_{eqZ} K_{eqX}	Equilibrium constant.	$molecule$
H_0	Optimal temperature.	$^\circ C$
H	Current temperature.	$^\circ C$
L	Light amount molecule.	$molecule$
m, n	Cooperativity factor.	–

of Equations 1-4 separately from the rest of the system. The steady state value, x^*, can be found by setting $\frac{dx}{dt}$ to 0 and solving Equation 4 for x as follows:

$$x^* = \frac{\lambda_1 + \lambda_2}{d_x} \tag{7}$$

where

$$\lambda_1 = \frac{k_a K_{eqH}}{K_{eqH} + (H - H_0)}$$
$$\lambda_2 = \frac{k_b K_{eqL}}{K_{eqL} + L}.$$

Equations 1-3 are a system of three equations and three variables. Thus, the state steady solution of Equation 3 can be expressed in terms of a and b as follows:

$$c = k_c \frac{a\, b}{k_{cr} + d_c}. \tag{8}$$

Substituting c in Equations 1-2 at steady state results in the following two equations:

$$0 = \lambda_1 - d_a\, a + \gamma a\, b \tag{9}$$
$$0 = \lambda_2 - d_b\, b + \gamma a\, b \tag{10}$$

where $\gamma = \frac{k_c k_{cr}}{k_{cr} + d_c} - k_c$. By separating the variables in Equation 9 and substituting into Equation 10, we can solve for the fixed points of a, b, and c as follows:

$$a^* = \frac{\lambda_1}{d_a - \gamma\, b^*} \tag{11}$$

$$b^* = \frac{-\alpha \pm \sqrt{\alpha^2 - 4\, d_b\, \gamma\, \lambda_2\, d_a}}{2\, d_b\, \gamma} \tag{12}$$

$$c^* = k_c \frac{a^*\, b^*}{k_{cr} + d_c} \tag{13}$$

where $\alpha = \gamma\lambda_1 - \gamma\lambda_2 - d_a d_b$. The fixed points a^*, b^*, c^*, and x^* are dependent on the input parameters H and L, so these fixed point values are functions of H and L. Using this observation, Equations 5-6 can be rewritten as follows:

$$0 = f(H, L) + \frac{k_{y2}\, K_{eqZ}^m}{K_{eqZ}^m + z^m} - k_y\, y \tag{14}$$

$$0 = g(H, L) + \frac{k_{z2}\, K_{eqY}^n}{K_{eqY}^n + y^n} - k_z\, z \tag{15}$$

where

$$f(H, L) = \begin{cases} 0, & \text{if } H, L = (0, 0) \\ f_M, & \text{if } H, L = (0, 1)\ or\ (1, 0) \\ f_H, & \text{if } H, L = (1, 1) \end{cases}$$

and

$$g(H, L) = \begin{cases} g_H, & \text{if } H, L = (0, 0) \\ 0, & \text{if } otherwise \end{cases}$$

where f_M, f_H, and g_H are dependent on the values of the parameters. Note that the relationship $f_M < f_H$ is always true. If we ignore the functions $f(H, L)$ and $g(H, L)$ in Equations 14 and 15, the nullclines of Equations 14 and 15 are two sigmoid functions. A change in the values of $f(H, L)$ and $g(H, L)$ shifts the nullclines, causing the creation or annihilation of fixed points. Figure 10 shows the behavior of the phase plane in each of the different states.

Suppose the levels of H and L are low, then $f(H, L) = 0$ and $g(H, L) = g_H$. This shifts the nullcline of z up, and causes one fixed point at a high z value and a low y value as shown in Figure 10(a). Physically, this means that the system stabilizes in a state where the TetR level is high and the LacI level is low. Small perturbations to the system such as a spike in LacI levels dies out and returns to the stable equilibrium. When both H and L are high, $f(H, L) = f_H$ and $g(H, L) = 0$. Thus, this shifts the nullcline of x significantly to the right and causes one fixed point where y is high and z is low as shown in Figure 10(c). Physically, this means that the system stabilizes in a state where the TetR level is low and the LacI level is low. When H and L have different values, $f(H, L) = f_M$ and $g(H, L) = 0$. This slightly shifts the y nullcline to the right. In this situation, three fixed points appear as shown in Figure 10(b). The system goes to equilibrium at either the point λ_1 or λ_2, depending on the initial conditions of TetR and LacI. If the system is originally in the low LacI state, then the system is closer to λ_1, and stabilizes at λ_1. If the system is originally in the high LacI state, the system is closer to λ_2, and it stabilize at λ_2. This property of having two stable equilibrium points separated by an unstable fixed point allows the system to hold state when the input signals are mixed. Since $f(H, L)$ and $g(H, L)$ can be controlled by changing the *tetR* and *lacI* gene counts, the state holding behavior of our genetic Muller C-element can also be controlled outside of the toggle switch.

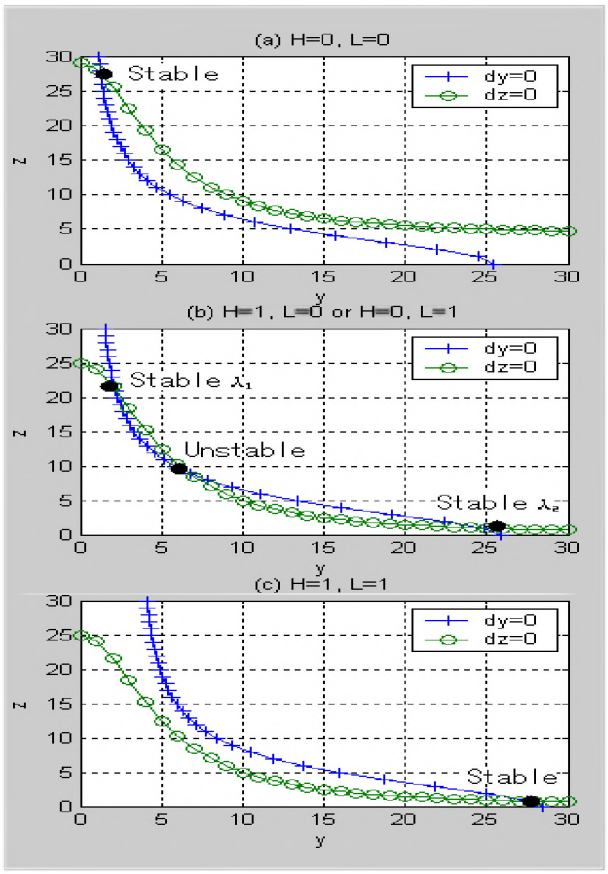


Figure 10. Phase plane diagram. (a) Heat and light are low. (b) Heat is high or light is high. (c) Heat and light are high.

6 Stochastic Analysis

According to our differential equation model, our genetic C-element design works correctly. However, the molecule counts in genetic circuits are extremely small, making the continuous-deterministic assumption inadequate. Genetic circuits are never going to be perfect as there is always a possibility that stochastic effects could accumulate to cause the circuit to fail. Failure for a C-element is the inability for it to retain its state when the inputs have different values. To analyze the probability of failure, we performed 5,000 stochastic simulation runs using `Reb2Sac` [12] under several conditions for both the original majority gate C-element and the toggle switch C-element. This tool uses an optimized form of Gillespie's stochastic simulation algorithm [8]. First, the C-element is put into its low state and either heat or light is applied. If the C-element changes into its high state, then a failure is recorded. Figure 11 shows the probability of this type failure at various times from immediately after the input is applied to up to one cell cycle later (i.e., about 35 minutes or 2100 seconds). The result is that both circuits perform reasonably well with the toggle switch design having a probability of failing within one cell cycle of only about 4 percent. Next,the C-element is put into its high state and either heat or light is removed. If the C-element changes into its low state, then a failure is recorded. Figure 12 shows the probability of this type failure at various times. In this case, the majority gate circuit performs very poorly with about 80 percent changing state before half a cell cycle. The toggle C-element, however, fails less than 10 percent of the time during one cell cycle.

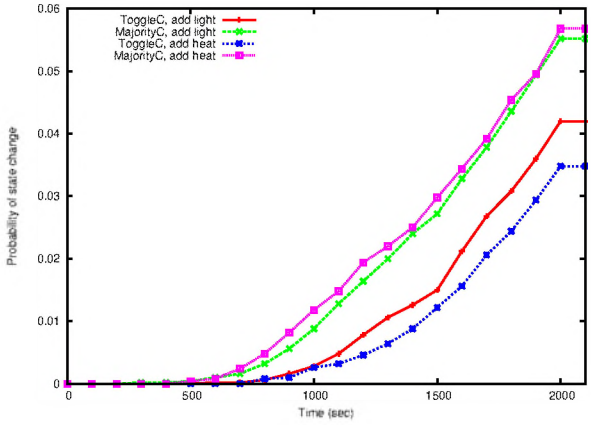


Figure 11. Probability of the C-element state to change from low to high after only one of the inputs has been applied.

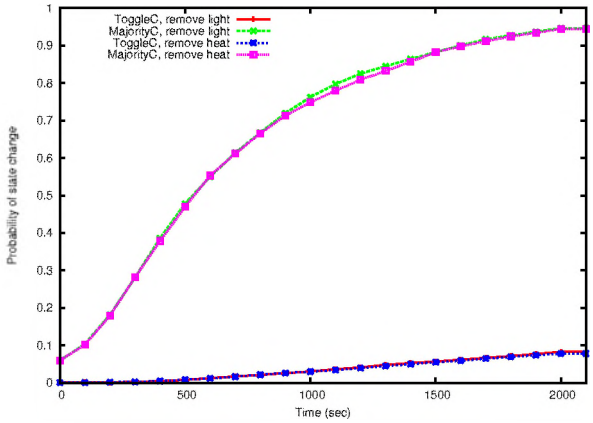


Figure 12. Probability of the C-element state to change from high to low after only one of the inputs has been removed.

7 Discussion

This paper presents the design of a Muller C-element that is constructed from genetic logic gates and a genetic toggle switch. Analytical analysis of a differential equation model of this circuit shows that its state-holding behavior is fairly robust to parameter variation. This analysis also yields insight as to how modifications of rates affect the robustness of the design. Finally, genetic circuits are inherently stochastic in nature, so this paper also shows that the state can be held fairly reliably for more than a cell cycle.

Another goal of this paper is to show the challenges faced in the design of genetic circuits. In particular, this paper describes a majority gate implementation of a genetic C-element, and it shows that this circuit is not robust. This result indicates that design methods for genetic circuits face numerous challenges since a purely logical abstraction is not always valid.

This construction of a genetic Muller C-element coupled with prior results with combinational logic gates allows for the potential to design any asynchronous circuit using genetic material. Therefore, the long-term goal of this research is to adapt asynchronous methods and tools for the design of genetic circuits. Such a potential design flow is shown in Figure 13. In this flow, logic equations for a genetic circuit would be produced by the timed asynchronous circuit synthesis tool `ATACS` [13]. These equations would then be mapped to genetic gates from a library of gates including those from MIT's parts database and other sources. The technology mapper would likely need to make use of stochastic simulation using the `Reb2Sac` tool. Finally, the DNA sequence for the genetic circuit would be produced. There are currently several companies which will produce DNA given a sequence. This DNA can then be inserted into an organism such as *E. coli* for testing.

Once we can build a reliable genetic Muller C-element, we can use the gate to synchronize multiple independent inputs. One potential use is to combine the genetic Muller C-element and *quorum sensing*, the mechanism used by bacteria to communicate among each other. This combination can allow a bacteria cell to determine when all its inputs have arrived, and signal all other cells that the cell is ready. Anderson *et al.* are using a similar idea to engineer bacteria to attack cancerous cells [2].

Figure 13. Design flow.

While this paper represents a promising first step, there are several open issues that must further explored. First, using heat as an input may affect other elements in the cell in ways that may conflict with the operation of the C-element. Second, due to the stochastic nature of the behavior of the gates, we will need to examine how to build reliable circuits out of unreliable parts. One possible way to handle failure is to have multiple copies of the circuit in a cell. Another issue is that LuxR and LuxI can diffuse outside a bacteria which may affect the behavior of other bacteria in unpredictable ways. This may require us to use different proteins to build our AND gate. Recent research in different biological gates [7, 17, 11] may allow us to construct even better Muller C-elements in the future. Finally, many issues cannot be fully explored until we synthesize the DNA sequence for this genetic circuit and test it *in-vivo* (i.e., within a living organism) which we do plan to do in the future.

References

[1] Registry of standard biological parts. `http://parts.mit.edu/`.

[2] J. C. Anderson, E. J. Clarke, and A. P. Arkin. Environmentally controlled invasion of cancer cells by engineering bacteria. *J. Mol. Biol.*, 355:619–627, 2006.

[3] G. M. Brazil, L. Kenefick, M. Callanan, A. Haro, V. de Lorenzo, D. N. Dowling, and F. O'Gara. Construction of a rhizosphere pseudomonad with potential to degrade polychlorinated biphenyls and detection of *bph* gene expression in the rhizosphere. *Applied and Environmental Microbiology*, 61(5):1946–1952, 1995.

[4] I. Cases and V. de Lorenzo. Genetically modified organisms for the environment: stories of success and failure and what we have learned from them. *International Microbiology*, 8:213–222, 2005.

[5] M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335–338, 2000.

[6] T. S. Gardner, C. R. Cantor, and J. J. Collins. Construction of a genetic toggle switch in *escherichia coli*. *Nature*, 403:339–342, 2000.

[7] E. Gerhart, H. Wagner, and K. Flärdh. Antisense rnas everywhere? *TRENDS in Genetics*, 18(5):223–226, 2002.

[8] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, 1977.

[9] J. A. Goler. Biojade: A design and simulation tool for synthetic biological systems. Technical report, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 2004. AI Technical Report 2004-003.

[10] J. Hasty, D. McMillen, and J. J. Collins. Engineered gene circuits. *Nature*, 420:224–230, November 2002.

[11] F. J. Isaacs, D. J. Dwyer, C. Ding, D. D. Pervouchine, C. R. Cantor, and J. J. Collins. Engineered riboregulators enable post-transcriptional control of gene expression. *Nature Biotechnology*, 22(7):841–847, 2004.

[12] H. Kuwahara, C. Myers, N. Barker, M. Samoilov, and A. Arkin. Automated abstraction methodology for genetic regulatory networks. *Trans. on Comput. Syst. Biol. IV*, pages 150–175, 2006.

0-7695-2771-X/07 $20.00 © 2007 IEEE

[13] C. J. Myers, W. Belluomini, K. Killpack, E. Mercer, E. Peskin, and H. Zheng. Timed circuits: A new paradigm for high-speed design. pages 335–340, Feb. 2001.
[14] D.-K. Ro, E. M. Paradise, M. Ouellet, K. J. Fisher, K. L. Newman, J. M. Ndungu, K. A. Ho, R. A. Eachus, T. S. Ham, J. Kirby, M. C. Y. Chang, S. T. Withers, Y. Shiba, R. Sarpong, and J. D. Keasling. Production of the antimalarial drug percursor artemisinic acid in engineered yeast. *Nature*, 440, 2006.
[15] M. L. Simpson, C. D. Cox, G. D. Peterson, and G. S. Sayler. Engineering in the biological substrate: Information processing in genetic circuits. 92(5):848–863, 2004.
[16] D. Sprinzak and M. B. Elowitz. Reconstruction of genetic circuits. *Nature*, 438:443–448, 2005.
[17] L. Yen, J. Svendsen, J.-S. Lee, J. T. Gray, M. Magnier, T. Baba, R. J. D'Amato, and R. C. Mulligan. Exogenous control of mammalian gene expression through modulation of rna self-cleavage. *Nature*, 431:471–476, 2004.

8 Appendix A: Parameter Values

Selecting parameter values is a difficult task since many parameter values are unknown [16]. For our simulation and analysis, we selected reasonable values for our parameters. For many unknown parameters, we used ratios to describe the affinity for rates for binding and unbinding. Table 4 lists the parameter values used in the creation of the phase plane diagrams shown in Figure 10. Our effective rates of protein production parameters are on the same order of magnitude as those used in Gardner *et al.*'s model [6].

Table 4. Parameters used in the phase plane.

Parameter	Value	Units
k_{y1}, k_{z1}	5	$\frac{molecule}{sec}$
k_a, k_b, k_{y2}, k_{z2}	25	$\frac{molecule}{sec}$
All k_{eq}	5	$\frac{1}{molecule}$
All decay rates	.75	$\frac{1}{sec}$
k_c	2	$\frac{1}{sec \cdot molecule}$
$k_c r$	10	$\frac{1}{sec}$
m, n	2	-

Table 5 shows the parameters used in simulating the majority gate design of the Muller C-element. Table 6 shows the parameters used in simulating the toggle switch design of the Muller C-element. Activated basal rate refers to the rate of production after the LuxR-LuxI complex has bound to the promoter. Complex dissociation rate refers the the rate at which the LuxR and LuxI complex degenerates back into LuxR and LuxI. The initial dissociation rate refers to the rate at which the first LacI or TetR molecule unbinds with the promoter. The second dissociation rate refers to the rate at which the second molecule of LacI or TetR unbinds with the promoter. We model negative cooperation by having a high rate of unbinding if only one molecule is bound to the promoter, but a low rate of unbinding if two molecules are bound to the promoter.

Table 5. Majority gate design parameters.

Parameter	Value	Units
GFP basal rate	.2	$\frac{molecule}{sec}$
All other basal rates	.1	$\frac{molecule}{sec}$
All decay rates	.0075	$\frac{1}{sec}$
Heat, Light dissociation rate	.5	$\frac{1}{sec}$
PenI, LacI, TetR dissociation rate	.2	$\frac{1}{sec}$

Table 6. Toggle switch design parameters.

Parameter	Value	Units
GFP and LacI basal rate	.2	$\frac{molecule}{sec}$
TetR basal rate	.15	$\frac{molecule}{sec}$
Activated basal rate	.025	$\frac{molecule}{sec}$
All other basal rates	.1	$\frac{molecule}{sec}$
Heat, Light dissociation rate	.5	$\frac{1}{sec}$
LuxR, LuxI basal rate	.1	$\frac{molecule}{sec}$
Complex dissociation rate	4	$\frac{1}{sec}$
LuxR, LuxI, Complex decay rate	.0125	$\frac{1}{sec}$
All other decay rates	.0075	$\frac{1}{sec}$
Initial TetR and LacI dissociation rate	1	$\frac{1}{sec}$
Second TetR and LacI dissociation rate	.1	$\frac{1}{sec}$