Computational simulation of a gene regulatory network implementing an extendable synchronous single-input delay flip-flop

Imad Hoteit, Nawwaf Kharma, Luc Varin

ECE Department, Concordia University, Montreal, QC, Canada

Biology Department, Concordia University, Montreal, QC, Canada

**Abstract**

We present a detailed and extendable design of the first synchronous single-input delay flip-flop implemented as a gene regulatory network in *Escherichia coli* (*E. coli*). The device, which we call the *BioD*, has one data input (transcribing RNA), one clock input (far-red light) and an output that reports the state of the device using green fluorescent protein (GFP). The proposed design builds on Gardner’s toggle switch, to provide a more sophisticated device that can be synchronized with other devices within the same cell, and which requires only one data input. We provide a mathematical model of the system and simulation results. The results show that the device behaves in line with desired functionality. Further, we discuss the constraints of the design, which pertain to ranges of parameter values. The *BioD* is extended via the addition of an update function and input and output interfaces. The result is the *BioFSM*, which constitutes a synchronous and modular finite state machine, which uses an update function to change its state, stored in the *BioD*. The *BioFSM* uses its input and output interfaces for inter-cellular communications. This opens the door to the design of a circular cellular automata (the *BioCell*), which is envisioned as a number of communicating *E. coli* colonies, each made of clones of one *BioFSM*.

**1. Introduction**

Most of the complex processes that take place in a cell are governed by gene expression, which is regulated at several levels along the pathway leading from DNA to protein. Gene expression may be regulated during transcription and post-transcriptionally, including during protein translation and via post-translational modification of proteins. Notably, much of the control of gene expression is done either by regulatory proteins or by RNAs, which is themselves the products of genes. Hence, the interactions between DNA, RNAs, proteins, and other molecules, form natural gene regulatory networks (or GRNs) of varied complexity.

While studying these networks and their components provides invaluable information, it is essential to: (a) thoroughly investigate these components in different environments, while performing different functions, and (b) integrate this knowledge to build new synthetic gene regulatory networks and other devices. The discipline of Synthetic Biology aims at systematically designing, building, combining and testing new biological functions and systems that do not occur in nature. Indeed, individual parts such as promoters and protein coding sequences can be assembled into GRNs that perform desired functionalities, such as computing machines.

The synthesis of computing machines via the manipulation of DNA within or without living organisms, started in 1994 when Adleman executed an experimental procedure that used DNA, in vitro, to solve an instance of the directed Hamiltonian path problem (Adleman, 1994). In contrast, in vivo cell-based or cellular computing started in 1998 with the modification of the genome of the prokaryote *Escherichia coli*, to realize 1- and 2-input combinational Boolean logic gates (e.g., NOT, AND and IMPLIES) (Knight and Sussman, 1998; Weiss et al., 1998); a similar feat was achieved with eukaryotic cells by Kramer et al. (2004). Along another dimension, time-dependant or sequential Boolean logic devices have also been implemented in living cells, starting with a 2-input toggle switch by Gardner et al. (2000), and a synthetic oscillator by Elowitz and Leibler (2000). In fact, in one decade this field has grown to generate many elementary devices (Drubin et al., 2007; Boyle and Silver, 2009; Tigges et al., 2009; Haynes and Silver, 2009), including band-pass filters (Basu et al., 2005) and counters (Friedland et al., 2009). More complicated devices such as engineered multi-cellular pattern generators (Basu et al., 2004, 2005), single cell biosensors (Levskaya et al., 2005; Tecon et al., 2006), tumor-targeting bacteria (Anderson et al., 2006), cell-based computers (Cox et al., 2007;
Balagadde et al., 2008), and biological memory devices (Chang et al., 2010) have also been synthesized or proposed.

In the particular case of switching devices, there has been a fair number of switches built or theorized, which involve (a) DNA modification (e.g. using invertases); (b) regulation of the process of transcription, (c) post-transcriptional regulation (involving various RNA molecules), as well as (d) post-translational regulation (by changing the state of expressed proteins).

The first example of the use of invertases is Ham et al. (2006), which places the promoter of a gene between two specific elements targeted by the FimE flipase. The flipase inverts the inversion region between these two elements (including them). This completely disables transcription from that promoter, rendering the associated gene silent. This is a unidirectional operation and it does not require qualification by a clock. In 2008, Ham et al. (2008) expanded their initial concept by using both the hin and FimE inversion mechanisms. This allowed them to use the relative positions of the elements marking the inversion regions to propose three- and five-state machines, which rely completely on the two flipases to change state. It is worth noting that this method of defining state is heritable as changes to the DNA are permanent and hence, inherited by the offspring.

The most prominent example of a toggle switch that is transcriptionally controlled is that of Gardner et al. (2000). However, this toggle switch requires two inputs and operates asynchronously (is not controlled by a clock input). Elowitz and Leibler (2000) synthesized a three gene oscillator (plus an additional gene for reporting), dubbed repressilator. The product of each of the three genes represses the next gene in a loop, with the last gene repressing the first one. The repressilator is not a bistable switch but rather a self-maintaining oscillator that proceeds from one state to the next, autonomously and without the need for any clock input. Kobayashi et al. (2004) utilized slightly modified versions of Gardner's toggle switch as memory modules of larger networks that sensed specific events (e.g. DNA damage) and generated particular responses (e.g. biofilm formation). In this case, the toggle switch is, by default, in one specific state, which flips in response to the sensed event. It does not have two inputs, but it does not have two stable states either. And, as is the case with Gardner's switch, it operates asynchronously. Stricker et al. (2008) synthesized a two gene oscillating network, where one gene is responsible for the activation of both genes, and the other gene is responsible for repressing both genes. This network improves on the repressilator in terms of speed, durability of the oscillation and the ability to externally tune its oscillations. Nevertheless, this network is not a switch that can be used as a memory module, such as Gardner's toggle. Lou et al. (2010) propose a single-input toggle switch, made of a Gardner-like two-gene memory module and a single-gene NOR gate module. The memory module is, by default, in a particular stable state. Upon the introduction of a UV input, several proteins degrade, which causes the memory module, with help from the NOR module to switch to a new state and maintain it. This is, in fact, a single-input switch, but it lacks a clock input.

One very significant work of RNA-based switching behavior is that of Bayer and Smolke (2005). They present devices that are regulated post-transcriptionally using RNA riboswitches. A riboswitch is an RNA molecule containing two domains: (i) a ligand-binding aptamer domain and (ii) an antisense regulator domain. The latter is used to block the ribosome binding site (RBS) and prevent translation, while the former binds a ligand that triggers a conformational change in the riboswitch, resulting in either the covering or uncovering of the anti-sense regulator domain. Riboswitches have the advantage that they can be designed and/or evolved to respond to many ligands including proteins and RNA molecules. Riboswitches have been synthesized to respond to one or more inputs (ligands). Although current riboswitches change state uni-directionally, it is possible to imagine riboswitches that respond to inducible small protein ligands. So far, riboswitches act asynchronously.

Finally, a good example of how switches can be regulated at the protein level is the work of Dueber et al. (2003), which modified the natural N-WASP allosteric switch to synthesize 1- and 2-input synthetic protein switches. In the 2-input switch, the hybrid protein was engineered to have two A-terminal auto-inhibitory domains that correspond to the output domain and a C-terminal domain on the protein. The way in which the protein responded to the two input ligands (PDZ and Cdc42) relied on the relative positioning of the four domains. They used this to synthesize various switches, whose state (active or not) depended on combinatorial functions of the two inputs. All of their devices are asynchronous and unidirectional.

Despite the many works on genetic switches (also called flip-flops), all published synthesized and proposed designs work asynchronously, usually utilizing more than one external logical input. A notable exception is (Lou et al., 2010) which is a single-input switch, albeit still asynchronous. Lack of synchronization-ability entails that the operation of a flip-flop cannot be synchronized with the operation of other parts of a larger system, using a single global clock. Also, a true delay flip-flop has but one logical input. Though the use of a single input complicates design, it does simplify use and allow for easier expansion of function. We call the proposed GRN embodying a synchronous single-input delay flip-flop the BioD. It is, in summary, a novel GRN that changes states in response to single logical input, and only on the rising edge of a clock signal. Its specification and detailed design, modeling and simulation results follow.

In parallel to advances in GRN design, mathematical modeling and simulation tools have been developed to help make approximate predictions of the behavior of GRNs before significant resources are allotted to their synthesis. These include, but are not limited to, deterministic (Hindmarsh et al., 2005) and stochastic simulation algorithms (Gillespie, 1977), metabolic control analysis (MCA) (Olivier et al., 2005), structural analysis (Olivier et al., 2005) and flux-balance analysis (FBA) (Orth et al., 2010). Deterministic simulation models include differential equations, Boolean networks, logical networks and rule-based formalisms (de Jong et al., 2002). Stochastic models include P systems (Romero-Campero et al., 2009), Bayesian networks and master equations (de Jong et al., 2002). An interesting comparison was offered by Twycross et al. (2010) of the benefits of each of the deterministic and stochastic models and presented as a case study using an auxin-transport example as a common base of comparison. MCA quantifies how variables, such as fluxes and species concentrations, depend on network parameters. Structural analysis is mostly used for genome-scale models to determine reduced stoichiometric matrices. FBA is used for optimizing the growth rate of a modeled organism, while falling within the constraints of its internal metabolites.

There exists a long list of software packages and libraries capable of implementing one or more of the above mentioned simulation methods. A very important clustering of these tools can be found under the SBML.org umbrella. The Systems Biology Markup Language (SBML) was developed by a small team of researchers who identified the need to enable interoperability between the vast arrays of simulation software that became available (Hucka et al., 2003). Although we wrote our own software to simulate our networks, there exist hundreds of very powerful software packages in the SBML repository. Suffice it to say, the scope of this paper does not cover the plethora of tools out there, so we instead highlight a good qualitative modeling tool like the Genetic Network Analyzer (GNA) (Batt et al., 2012), a more complete and quantitative collection of tools like the Systems Biology Workbench (SBW) (Hucka et al., 2002), and a good multi-cell simulation tool, the Infobiotics Workbench (Blakes et al., 2011).
2. Network Design and Modeling

In abstract terms, the BioD is a gene regulatory network acting as a delay flip-flop. By delay flip-flop, we mean a logical device that has an input (D), a clock (CLK), and an output (Q) equal to its state (S); see the logical block diagram in Fig. 11a (Q is the second output and is equal to the logical complement of Q). The state of the delay switch is held constant unless and until its input differs from its state, on the rising edge of the clock. In that case, the next state of the delay switch will copy the value of the input (i.e., Q = D). Hence, a cell that acts as a delay switch is effectively a 1-bit memory device, controlled by an input and a clock. The BioD also exhibits its state by expressing (or not) a fluorescent protein. This was the specification of the BioD; following is its internal design.

2.1. BioD Modeling

The BioD has two (logical and control) inputs: trans-activating RNA or tarRNA as input D, and the presence or absence of far-red (FR) light as the clock (CLK). It has two complementary outputs (Q and \( \overline{Q} \)) defining the state of the flip-flop: the ON state is indicated by the presence, in high concentrations, of green fluorescent protein (GFP), while the OFF state is indicated by its absence. As with its electronic equivalent, the BioD’s output follows the input on the rising edge of the clock. As shown in Fig. 1, the gene regulatory network implementing the BioD is comprised of three major parts: the INPUT genes, SELECTION genes and STATE genes.

Please note that the design involves several operons that include more than one protein coding sequence. To simplify our language without loss of accuracy, we refer to both genes and operons as genes (there are seven of them, numbered 1 to 7). Kindly note that we use italicized courier new for gene names (e.g. TetR) and courier new for proteins (e.g. TetR) as well as protein complexes. We also use italicized courier new for RNAs other than transcripts (e.g. tarR12), while distinguishing transcripts by attaching an “m” prefix to their names (e.g. mTetR).

2.1.1. INPUT Genes

The INPUT genes convey to the SELECTION genes whether an input signal is present or not. They do so by tipping the dynamic balance between the two mutually repressed genes, 4 and 5; this process is detailed in Section 3 below.

In order to sense input $D$, gene 1 is designed to be self-repressed, and this self-repression can only be lifted through the introduction of input $D$. To achieve this, a form of ribo-regulation is used called cis. This cis-regulation or in our case, cis-repression prevents the translation of the transcript of gene 1, as part of the transcript bends over to hybridize with the ribosome binding site (RBS), effectively locking it. The key comes in the form of trans-activating RNA (taRNA), which hybridizes with a particular location on the transcript in a manner that frees the RBS site from its cis-repression. This allows the ribosome to bind at the RBS and start the process of translation (Isaacs et al., 2004). The taRNA chosen for input $D$ is taR12 which is specifically designed to unlock the cis-repression of (the transcript of) gene 1, called crR12.

When input $D$ is present, the transcript of gene 1 gets translated into the $cI$ repressor (originally, from the $\lambda$ phage). $cI$ in turn represses gene 2. In the absence of input $D$, however, the cis-repressed transcript of gene 1 does not get translated into the corresponding repressor protein. This leads to the lifting of repression of gene 2, and hence the expression of its own repressor protein, $cII$ (originally, from the $p22$ phage).

In summary, the presence of input $D$ results in the production of the $cI$ protein, while its absence leads to the production of the $cII$ protein.

### 2.1.2. STATE Genes

The STATE genes have an analogous configuration to that of Gardner's toggle. They consist of two co-repressed genes, and as such define the state of the BioD device. The products of genes 4 and 5 represent complementary outputs $D$ and $Q$, respectively. The presence of a green fluorescent protein (GFP) signals the presence of logical output $Q$, while its absence signals the presence of its logical complement $\bar{Q}$. The co-repressed nature of the toggle switch means that when either gene is active, the toggle enters into a stable steady state. In the context of the BioD, only the SELECTION genes can perturb the stability of the STATE genes.

Two important points need to be made here. First, the SELECTION genes can affect the STATE genes, independently of the current state of the BioD. Second, genes 4 and 5 are mutually exclusive, which makes it impossible for the SELECTION genes to set the state of the STATE genes to both ON and OFF, simultaneously. Which of the two genes (4 or 5) is activated depends on the state of the INPUT genes at the time the CLK signal is turned ON.

### 2.1.3. SELECTION Genes

The SELECTION genes are always OFF until turned ON by FR light (the CLK input). In the absence of FR light, genes 4 and 5 are always repressed by the phosphorylated version of Ompr, i.e., OmprP. Gene 3 is constitutively expressed and produces Ompr. OmprP is phosphorylated in the presence of the EnvZ enzyme. EnvZ is connected to Cphi, which in the presence of FR light, induces a conformational change in EnvZ, preventing the phosphorylation of OmprP. The genes that produce EnvZ and Cphi (and others needed for the light response system) are not shown in Fig. 1. See reference Levskaya et al. (2005) for a fully detailed explanation.

The phosphorylation of OmprP is dominant in the absence of FR light and negligible in its presence. Therefore, the FR light signal causes a drop in OmprP levels and a corresponding rise in Ompr levels. This drop results in partial lifting of the repression of both genes 4 and 5, as their promoter omprP, is both repressed by OmprP and activated by Ompr. Both the functionality of omprP and the complementary levels of Ompr and OmprP result in a system that is quick to start or stop transcription of both genes 4 and 5.

The SELECTION genes also respond to and affect the INPUT genes. As previously stated, the BioD is an edge-triggered device, i.e., it responds to the input when the CLK signal turns ON, but not when the CLK signal is ON. If the CLK signal is ON and either gene 4 (or 5) is ON, then gene 4 (or 5) would be repressing the genes that could potentially repress it. Namely, gene 4 would repress genes 2 and 5, and gene 5 would repress genes 1 and 4. As a result, any change due to input $D$, when the CLK signal is already ON, does not propagate to the SELECTION genes. For a toggle (ON) input signal to affect the current state of the SELECTION genes, the CLK signal must first turn OFF for a period then ON again.

Given that the dynamics of such a gene network are non-trivial, we provide a summary of its operation using a state transition table (Table 1) plus a single fully detailed scenario, tracing through one typical sequence of transitions. The scenario is that of a change of state, from OFF to ON, in response to a turned ON input $D$, whose level must stabilize, prior to the introduction of the CLK signal (FR light).

<p>| Table 1 |
|-----------------|-----------------|
| <strong>State transition table:</strong> “X” is do not care; “-” is no change. |</p>
<table>
<thead>
<tr>
<th>CLK</th>
<th>D</th>
<th>Current state</th>
<th>Next state</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$S_2$</td>
<td>$S_4$</td>
<td>$S_5$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

As long as the CLK signal is ON, the new state will be maintained. If a significant change in the input level occurs while the clock is ON, the repressions of genes 2 and 5 would not disappear, since gene 4 is ON and produces $cI$. Indeed, as long as gene 4 is ON, it has the ability to keep itself from being repressed by other genes, that is, by repressing them. It is only when the CLK signal is removed and both genes 4 and 5 are OFF that the system is again free to respond to input $(D)$, upon the re-introduction of the CLK signal.

### 2.1.3.1. Model

The gene regulatory network of Fig. 1 is simulated deterministically and stochastically, following a mathematical model. The model is shown below as (a) a system of ordinary differential equations (ODEs) modeling the production of mRNA
transcripts, and (b) a system of ODEs modeling the translation of these transcripts into their respective proteins.

We define the following terms and chemical species: mClcr is the cis-repressed mRNA transcript of gene 1; mX is the mRNA transcript for the protein X; prodGeneX is the amount of transcripts produced by gene X at any given time; ρX is the maximum transcription rate of the promoter of gene X; while ωX, nX, KX and [X] are, respectively, the degradation constant, the Hill cooperativity coefficient, the dissociation constant and the concentration of substrate X.

2.1.3.2. Transcription ODEs.

\[
\frac{d [mClcr]}{dt} = \text{prodGene1} - \omega_{mRNA} \cdot [mClcr] \tag{1}
\]

\[
\frac{d [mCl]}{dt} = \text{prodGene4} - \omega_{mRNA} \cdot [mCl] \tag{2}
\]

\[
\frac{d [mCl]}{dt} = \text{prodGene2} + \text{prodGene5} - \omega_{mRNA} \cdot [mCl] \tag{3}
\]

\[
\frac{d [mOmpR]}{dt} = \text{prodGene3} - \omega_{mRNA} \cdot [mOmpR] \tag{4}
\]

\[
\frac{d [mGal4]}{dt} = \text{prodGene5} - \omega_{mRNA} \cdot [mGal4] \tag{5}
\]

\[
\frac{d [mTetR]}{dt} = \text{prodGene5} + \text{prodGene6} - \omega_{mRNA} \cdot [mTetR] \tag{6}
\]

\[
\frac{d [mLacI]}{dt} = \text{prodGene4} - \text{prodGene7} - \omega_{mRNA} \cdot [mLacI] \tag{7}
\]

where gene 1 is repressed by Gal4,

\[
\text{prodGene1} = \frac{ρ1}{1 + ([Gal4]/K_{Gal4})^{1/κ_{Gal4}}} \tag{8}
\]

gene 2 is repressed by cl, 

\[
\text{prodGene2} = \frac{ρ2}{1 + ([Cl]/K_{Cl})^{1/κ_{Cl}}} \tag{9}
\]

gene 3 is constitutively expressed,

\[
\text{prodGene3} = ρ3 \tag{10}
\]

gene 4 is repressed by both cl and OmpR, while being activated by OmpR,

\[
\text{prodGene4} = \frac{ρ4}{1 + ([Cl]/K_{Cl})^{1/κ_{Cl}}} × \frac{1}{1 + ([OmpR]/K_{OmpR})^{1/κ_{OmpR}}} \\
× \frac{([OmpR]/K_{OmpR})^{κ_{OmpR}}}{1 + ([OmpR]/K_{OmpR})^{κ_{OmpR}}} \tag{11}
\]

gene 5 is repressed by both cl and OmpR, while being activated by OmpR,

\[
\text{prodGene5} = \frac{ρ5}{1 + ([Cl]/K_{Cl})^{1/κ_{Cl}}} × \frac{1}{1 + ([OmpR]/K_{OmpR})^{1/κ_{OmpR}}} \\
× \frac{([OmpR]/K_{OmpR})^{κ_{OmpR}}}{1 + ([OmpR]/K_{OmpR})^{κ_{OmpR}}} \tag{12}
\]

gene 6 is repressed by LacI,

\[
\text{prodGene6} = \frac{ρ6}{1 + ([LacI]/K_{LacI})^{1/κ_{LacI}}} \tag{13}
\]

gene 7 is repressed by TetR,

\[
\text{prodGene7} = \frac{ρ7}{1 + ([TetR]/K_{TetR})^{1/κ_{TetR}}} \tag{14}
\]

2.1.3.3. Translation ODEs.

\[
\frac{d [Cl]}{dt} = γ_{Cl} \cdot K_{nCl} \cdot [nCl] \cdot [mClcr] + γ_{Cl} \cdot [mCl] - ω_{Cl} \cdot [Cl] \tag{15}
\]

\[
\frac{d [Cl]}{dt} = γ_{Cl} \cdot [mCl] - ω_{Cl} \cdot [Cl] \tag{16}
\]

The (1−L) term inserted in the Michaelis–Menten expressions of Eqs. (17) and (18) enables phosphorylation in the absence of FR light, i.e., when L=0.

\[
\frac{d [OmpR]}{dt} = γ_{OmpR} \cdot [mOmpR] - \frac{ν_{phos} \cdot (1 - L) \cdot [OmpR]}{κ_{phos} + [OmpR]} + ν_{dephos} \cdot [OmpRP] - ω_{OmpR} \cdot [OmpR] \tag{17}
\]

\[
\frac{d [OmpR]}{dt} = \frac{ν_{phos} \cdot (1 - L) \cdot [OmpR]}{κ_{phos} + [OmpR]} - ν_{dephos} \cdot [OmpRP] - ω_{OmpRP} \cdot [OmpRP] \tag{18}
\]

\[
\frac{d [Gal4]}{dt} = γ_{Gal4} \cdot [mGal4] - ω_{Gal4} \cdot [Gal4] \tag{19}
\]

\[
\frac{d [TetR]}{dt} = γ_{TetR} \cdot [mTetR] - ω_{TetR} \cdot [TetR] \tag{20}
\]

\[
\frac{d [LacI]}{dt} = γ_{LacI} \cdot [mLacI] - ω_{LacI} \cdot [LacI] \tag{21}
\]

Parameters values are as shown in Table 2. The degradation rates of various molecules are not known, so we use the rates arising from dilution by cell-growth. Somewhat elevated rates are used for ωCl and ωTet in order to avoid lingering production of cl and TetR, when the state is not favorable. That is feasible because protein-degradation rates can be artificially increased by adding to the protein coding sequence an SsrA tag, making the modified protein a target of various proteases in the cell (Elowitz and Leibler, 2000).

2.2. BioFSM modeling

From a computational point of view, a logical next step to the BioD is the design of a GRN embodying a finite state machine, which uses the BioD as a 1-bit memory module. We call this design a BioFSM, which is also a stand-alone module that can be modified to carry out different logical functions and/or to communicate with other modules via inter-cellular signaling.

The BioFSM has the following specification, characterized by its inputs, clock and current state. When the clock is OFF, there is no change in the state of the device. However, when the clock turns ON, the next state of the BioFSM is determined by a state update function (the UF), which is a function of its external inputs and its own current state.

The design of the BioFSM is shown in Fig. 11b. It consists of 3 modules: (a) a BioD, which holds the state of the BioFSM; (b) the UpdateFunction/InputInterface (or UF/UI) module. The genes implementing the UF/UI implicitly include the input interface, as changing any of the two external inputs requires a change to the promoter side of the genes (see Fig. 3b); (c) the OutputInterface (OI) module, which is used to enable a chosen acyl-homoserine lactone (AHL) molecule as the output of the BioFSM (Fig. 3c). AHLS are a class of small molecules capable of inter-cellular signaling in E. coli and other bacteria (Fuqua et al., 2001). In fact, the two external inputs to the BioFSM are also AHLS. The modular design of the BioFSM allows us to alter its logic/inputs or output only by changing only its UF/UI or OI, respectively.

The example shown in Fig. 3 illustrates a particular UF/UI and OI. The update function is A + BC. Inputs A and C are the two originating
Table 2
Nominal values of the parameters of transcription and translation equations.

<table>
<thead>
<tr>
<th>Parameter Description</th>
<th>Value</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_1$ Max. transcription rate of promoter of gene 1</td>
<td>0.680 [nM/s]</td>
<td>Estimate</td>
</tr>
<tr>
<td>$\rho_2$ Max. transcription rate of promoter of gene 2</td>
<td>0.595 [nM/s]</td>
<td>Estimate</td>
</tr>
<tr>
<td>$\rho_3$ Max. transcription rate of constitutive promoter of gene 3</td>
<td>0.085 [nM/s]</td>
<td>Estimate</td>
</tr>
<tr>
<td>$\rho_4$ Max. transcription rate of promoter of gene 4</td>
<td>0.255 [nM/s]</td>
<td>Estimate</td>
</tr>
<tr>
<td>$\rho_5$ Max. transcription rate of promoter of gene 5</td>
<td>0.255 [nM/s]</td>
<td>Estimate</td>
</tr>
<tr>
<td>$\rho_6$ Max. transcription rate of constitutive promoter of gene 6</td>
<td>0.765 [nM/s]</td>
<td>Estimate</td>
</tr>
<tr>
<td>$\rho_{1,8,10}$ Max. transcription rate of promoters of genes 7, 8, 9 and 10</td>
<td>0.850 [nM/s]</td>
<td>Estimate</td>
</tr>
<tr>
<td>$\gamma X$ Translation rate of gene X (any gene)</td>
<td>0.1</td>
<td>Estimate</td>
</tr>
<tr>
<td>$V_{\text{pro}}$ Rate of cspG phosphorylation</td>
<td>20.0</td>
<td>Estimate</td>
</tr>
<tr>
<td>$V_{\text{deppro}}$ Rate of cspG de-phosphorylation</td>
<td>0.01</td>
<td>Estimate</td>
</tr>
<tr>
<td>$k_{\text{phos}}$ Kinetic phosphorylation constant</td>
<td>1.0</td>
<td>Estimate</td>
</tr>
<tr>
<td><strong>Degradation constants</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\omega_{\text{LacI}}$ Degradation of LacI</td>
<td>$2.31\times 10^{-3}$ [1/s]</td>
<td>Estimate</td>
</tr>
<tr>
<td>$\omega_{\text{TetR}}$ Degradation of TetR</td>
<td>$2.3\times 10^{-2}$ [1/s]</td>
<td>Baumeister et al. (1991)</td>
</tr>
<tr>
<td>$\omega_{\text{cI}}$ Degradation of cI</td>
<td>$7.6\times 10^{-3}$ [1/s]</td>
<td>Rennitz and Vainsyns (1990)</td>
</tr>
<tr>
<td>$\omega_{\text{CL}}$ Degradation of CI</td>
<td>$6.9\times 10^{-3}$ [1/s]</td>
<td>Vohradsky (2001)</td>
</tr>
<tr>
<td>$\omega_{\text{cR}}$ Degradation of cR</td>
<td>$1.3\times 10^{-2}$ [1/s]</td>
<td>Zhu et al. (2000)</td>
</tr>
<tr>
<td>$\omega_{\text{Gal}}$ Degradation of Gal4</td>
<td>$2.88\times 10^{-2}$ [1/s]</td>
<td>Estimate</td>
</tr>
<tr>
<td>$\omega_{\text{LexA}}$ Degradation of LexA</td>
<td>$2.01\times 10^{-5}$ [1/min]</td>
<td>Carnas et al. (2006) (half-life of ~60 min)</td>
</tr>
<tr>
<td>$\omega_{\text{tauR12}}$ Degradation of luxR12</td>
<td>$1.96\times 10^{-3}$ [1/s]</td>
<td>Goryachev et al. (2006)</td>
</tr>
<tr>
<td>$\omega_{\text{deg}}$ Degradation of an mRNA transcript</td>
<td>$2.88\times 10^{-3}$ [1/s]</td>
<td>Friedland et al. (2009)</td>
</tr>
<tr>
<td><strong>Dissociation constants</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$K_{\text{bas}}$ LacI repressor dissociation constant</td>
<td>10 [nM]</td>
<td>Wang et al. (2005)</td>
</tr>
<tr>
<td>$K_{\text{tet}}$ TetR repressor dissociation constant</td>
<td>5.6 [nM]</td>
<td>Stekel and Jenkins (2008)</td>
</tr>
<tr>
<td>$K_c$ cI repressor dissociation constant</td>
<td>8 [nM]</td>
<td>Basu et al. (2005)</td>
</tr>
<tr>
<td>$K_{CI}$ cI2 repressor dissociation constant</td>
<td>50 [nM]</td>
<td>Estimate</td>
</tr>
<tr>
<td>$K_{CIC}$ cII repressor dissociation constant</td>
<td>151 [nM]</td>
<td>Head et al. (1998)</td>
</tr>
<tr>
<td>$K_{CP}$ cIpr repressor dissociation constant</td>
<td>6 [nM]</td>
<td>Head et al. (1998)</td>
</tr>
<tr>
<td>$K_{Gal}$ Gal4 repressor dissociation constant</td>
<td>24 [nM]</td>
<td>Hong et al. (2008)</td>
</tr>
<tr>
<td>$K_{\text{tauR12}}$ luxR12 repressor dissociation constant</td>
<td>10 [nM]</td>
<td>Basu et al. (2005)</td>
</tr>
<tr>
<td>$K_{\text{tauR12}}$ luxR12 repressor dissociation constant</td>
<td>10 [nM]</td>
<td>Basu et al. (2005)</td>
</tr>
<tr>
<td>$K_{\text{tauR12}}$ luxR12 repressor dissociation constant</td>
<td>80 [nM]</td>
<td>Isaacs et al. (2004)</td>
</tr>
<tr>
<td><strong>Hill cooperativity</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n_{\text{bas}}$ LacI repressor Hill cooperativity</td>
<td>2</td>
<td>Basu et al. (2005)</td>
</tr>
<tr>
<td>$n_{\text{tet}}$ TetR repressor Hill cooperativity</td>
<td>2</td>
<td>Estimate</td>
</tr>
<tr>
<td>$n_{\text{cI}}$ cI repressor Hill cooperativity</td>
<td>2</td>
<td>Basu et al. (2005)</td>
</tr>
<tr>
<td>$n_{\text{cII}}$ cII repressor Hill cooperativity</td>
<td>2</td>
<td>Shih and Gussin (1984)</td>
</tr>
<tr>
<td>$n_{\text{cIpr}}$ cIpr repressor Hill cooperativity</td>
<td>2</td>
<td>Estimate</td>
</tr>
<tr>
<td>$n_{\text{Gal}}$ Gal4 repressor Hill cooperativity</td>
<td>2</td>
<td>Estimate</td>
</tr>
<tr>
<td>$n_{\text{LexA}}$ LexA repressor Hill cooperativity</td>
<td>2</td>
<td>Aksenov (1999)</td>
</tr>
<tr>
<td>$n_{\text{LuxA}}$ luxR12 repressor Hill cooperativity</td>
<td>4</td>
<td>Basu et al. (2005)</td>
</tr>
<tr>
<td>$n_{\text{tauR12}}$ luxR12 repressor Hill cooperativity</td>
<td>2</td>
<td>Estimate</td>
</tr>
</tbody>
</table>

from external sources, and are both AHLs, while $B$ represents the state of the BioD, and is a repressor. It is worth noting here that AHLs can be activators or repressors by forming complexes with larger proteins called R-proteins. The resulting R-protein/AHL complex can activate or repress production of genes based on the positioning of their specific binding sites within the promoter region of the AHL-regulated gene (Anderson et al., 1999; Medina et al., 2003). Hence, the logical complements of the external inputs, $\bar{A}$ and $\bar{C}$, are readily available, while the state $B = \text{LacI}$ and its logical complement $B = \text{TetR}$ are made available by the BioD. This flexibility often allows for the reduction in the number of genes required for the implementation of the UpdateFunction. As to the OutputInterface, all possible realizations are driven by the BioD’s LexA output, but would have different (AHL) products, depending on the application.

There exists many different types of AHLs (Fuqua et al., 2001; Shrouf and Parsek, 2006; Steinberg and Venturi, 2007) and each AHL has a particular R-protein that it activates. A very important property of the AHL signaling system is its high specificity. In nature, this system is used mainly for quorum sensing, and a high specificity insures no inter-species crosstalk. Both the very specific production process of an AHL molecule, and the specificity in binding of the R-protein to its cognate AHL, are responsible for the reliability of this system (Taga and Bassler, 2003). In the design of the BioSM, we have two external AHL input signals and one AHL output signal. Specifically, we use the LuxI/LuxR and LasI/LasR pairs for input, and the LuxX/LuxRe for output.

2.2.1. Model

The model used for the U/I and the OI simulations for $F = A + BC$ is presented below. The production of the R-proteins is not considered here because they are constitutively produced proteins, generated without regulation. The protein translation ODEs are not shown because there is no transcriptional regulation.

### 2.2.2. Transcription ODEs

$$\frac{d[t\text{al}2]}{dt} = prod_{\text{Gene8}} + prod_{\text{Gene9}} - \omega_{\text{mRNA}} \cdot [t\text{al}2] \quad (22)$$

$$\frac{d[m\text{Las}]}{dt} = prod_{\text{Gene10}} - \omega_{\text{mRNA}} \cdot [m\text{Las}] \quad (23)$$
The above systems of ODEs are the base of the stochastic simulation used to generate Figs. 4, 8 and 9. We used the tau-leaping algorithm (Cao et al., 2007), which achieves fast and accurate stochastic simulation by taking large time steps that leap over individual reactions. During a leap interval \((t, t + \tau)\), each reaction channel operates as a Poisson process with a constant intensity. The values of \(\tau\) that were used in our simulations varied between \(\tau = 5,\ \tau = 10\ and \ \tau = 20\). The figures were generated using GnuPlot, as previously described for the deterministic runs.

3. Simulation Results and Discussion

In the sequel, we present the results of simulating the device using a system of rate equations. The results confirm our expectation that the device will toggle when and only when required – though its speed can still be improved.

3.1. BioD

The core functionality of our BioD device is illustrated in Fig. 4. The highlighted areas indicate the presence of an input. The reddish hue reflects the presence of the clock input (CLK), while the grey diagonal pattern reflects the presence of the data input (D). The examples provided show different data cycles intersecting (or not) with four different clock cycles. This setting allows us to show that the device can indeed go from one state to the other in response to nothing more than the introduction of the inputs it was designed to respond to. Furthermore, this setting also goes through the various permutations of the inputs shown in Table 1.

Ideally, with four separate CLK inputs, the state of the device should follow the D input four times. In this case, the state should turn ON, then OFF, and then OFF again and finally ON. Fig. 4a displays those exact state changes in a stochastic run with initial condition is an OFF state. The normalized GFP expression output follows the input only at the rising edge of the clock. However, when the clock is ON or is OFF, any changes in the input do not propagate to the output. Fig. 4b shows the changes in the concentrations of the mRNA transcripts of the various substances involved. Please note that the concentration level of mOmpR is not displayed because this transcript is constitutively expressed. Fig. 4c shows the changes in the protein levels; the levels of LexA and GFP were not displayed because they do not affect the behavior of the device. Changes in protein concentrations follow changes in corresponding mRNA concentrations, except in situations where post-transcriptional regulation is in effect. In particular, when mCIcr is expressed in the absence of input D, the level of the cI repressor does not subsequently increase. Because of this highly correlated relationship between transcript and protein, the protein levels are not shown for the rest of the examples. Rather, the GFP figure is used to demonstrate the overall input/output relationship.

The concentration of a molecule is decided, mainly, by its rates of synthesis and degradation. Some transcripts have multiple stable levels of expression. Since cI, cII, LacI and TetR are not only produced by the SELECTION genes (but can also be produced by some of the INPUT or STATE genes) the production of their transcripts is significantly increased in the presence of the CLK signal. mTetR has four levels of expression: (i) all the genes that can produce it are OFF, (ii) gene 6 is ON, (iii) gene 5 is ON, and (iv) genes 5 and 6 are ON. mLacI has similar multiple levels of expression, using genes 4 and 7. In the case of mCI, however, since gene 4 can only turn ON when gene 1 is ON, it only has three levels of expression. The case of mCII is analogous to that of mCI.

Tracing the various signals in Fig. 4b shows that, the simulation starts with three active transcripts, mTetR (the state of the device is
OFF), mCII (unrepressed since the CLK and therefore Gal4 are OFF) and mCI (unrepressed since input D is ON). Following, is a step-by-step explanation of the changes shown in the timing diagram (Fig. 4b).

First, input D is introduced, causing the repression of gene 2 (or mCI). Since the transcript of gene 1 is translated and gene 2 is OFF, gene 4 is on a hair-trigger to be turned ON, while gene 5 is doubly repressed by OmpRP and cI. The CLK signal is introduced, stopping the phosphorylation of OmpRP and activating gene 4. This raises the level of mCI and mLacI. The latter represses gene 6 and starts turning the state of the device ON. As mTetR degrades, GFP levels increase. Then, the CLK signal is turned OFF followed by input D. These two actions turn OFF gene 4 and disable gene 1, respectively. With both inputs OFF, the cI repressor produced by genes 1 and 4 degrades without replacement, allowing mCI to return to its previous level. mLacI, which is now produced by gene 7, reaches its unpressed (ON) state equilibrium.

The second state change occurs when the CLK signal is turned ON again. Since mCI is expressed at that time (no input D), gene 5 turns ON, causing the repression of gene 1 (through mLacI), the repression of gene 7 (through mTetR), and an increase in the level of mCI (as it is produced by both genes 2 and 5). When the CLK is removed, gene 5 is turned OFF, but mCI and mTetR remain high, while mLacI is repressed. This allows the production of mCII to start again (after Gal4 degrades). Note, however, that mTetR is now produced by gene 6, and not by gene 5.

The third CLK signal starts now. Gene 5 is again turned ON; the levels of mCI, mGal4 and mTetR climb; the level of mCII drops (repressed by Gal4). In the middle of the CLK pulse, input D is introduced. This causes no change in the network. Since input D only affects gene 1, its effects are muted because the clock has already turned on gene 5 which repressed gene 1. It is only after the clock is turned OFF that the repression of gene 1 is lifted. At this point, even though the CLK signal is removed, input D is still present, and since gene 1 is no longer repressed by gene 5 (or Gal4), cI is synthesized, which proceeds to repress gene 2. The state of the device, however, does not change since the STATE genes are not directly affected by the INPUT genes.

The fourth CLK signal turns the state of the device back ON. In the presence of input D, the CLK turns gene 4 ON causing a similar sequence of events to the one witnessed following the first CLK signal.

3.1.1. Model Constraints

An important factor in the design of any gene network is the choice of regulatory sequences, promoters and coding sequences, which make up the various genes. The specific genes used for the realization of the BioD are just an example, meaning that other genes can be used to realize the logical design of the BioD (shown in Fig. 2), though they will likely have a different set of model parameters. The variation of these parameters changes the behavior of the network, possibly making it faster or slower in responding to the inputs or in reaching a steady state.

The model parameters are explained as follows: the dissociation constant $K_d$ reflects the affinity of a repressor binding to its operator site; the Hill coefficient $n_H$ reflects the cooperativity of repression of the constituent molecules of a multimer; the degradation rate $o_{ij}$ depends on the chemical and spatial properties of the substance but can be modified using certain well-studied methods (such as the addition of an SsrA tag to speed-up degradation).

In a network where two genes repress each other (such as the two STATE genes), a small increase in the dissociation constant ($K_d$) of one of the two repressors, affects the network’s response time to the input in two separate ways: (i) it significantly reduces the response time of the target gene, and (ii) it increases the response time of the gene that produces it. The state genes are used to illustrate this issue.

We chose to record the effect of separately varying $K_{lac}$ on the dynamic behavior of mLacI and mTetR. Fig. 5a illustrates the effect of changing $K_{lac}$ from 0.5 nM to 14.0 nM on mLacI, leading – or not – to a change of state from OFF to ON. Similarly, Fig. 5b illustrates...
the effect of changing the value of $K_{\text{Lac}}$ on $[\text{mTetR}]$, leading to a change of state from ON to OFF.

Generally speaking, the $K_d$ value is not the only parameter defining a repressor, nor can this value be changed at will, because it is dependent on the chemical and conformational properties of both the repressor and its corresponding binding site. Therefore, any change in one gene's parameters might have effects beyond those intended. This must always be taken into consideration during design or optimization of gene regulatory networks.

It is noteworthy that in Fig. 5a, there is one $K_d$. This occurs because the CLK signal becomes too short for the state change to occur at this $K_d$ value. A more detailed discussion of the relationship between $K_{\text{TetR}}$, $K_{\text{Lac}}$ and the CLK signal is provided in Section 3.1.4.

3.1.2. Clock Input (CLK)

When the input and output states are at opposite levels, the length of the CLK signal must be large enough to allow a change of state to occur. As an example, when input = ON and output = OFF, the CLK signal must be sustained for a time greater than the minimum time needed for the cell concentration of $[\text{mTetR}]$ (or $[\text{mTetR}]$) and for $[\text{TetR}]$ to degrade below $[\text{mLacI}]$ and $[\text{LacI}]$, respectively. If the CLK signal is removed too soon, the production of $\text{mLacI}$ from the SELECTION gene 4 is cut too quickly. The output responds to its short presence and reduces the production of $\text{mTetR}$, seemingly heading towards a state change. However, when the CLK signal is removed, the $\text{mTetR}$ production is simply reasserted, because gene 7 has not yet begun the production of $\text{mLacI}$, and the state of the device fails to toggle. As seen in Fig. 6a, the GFP levels do not rise even though input D was present at the rising edge of the clock. In point of fact, the CLK signal enabled the transcription of $\text{mLacI}$ from gene 4 (which is not repressed by $\text{TetR}$). This causes the levels of $\text{TetR}$ to fall rapidly. However, the CLK signal is removed before they could fall low enough to turn gene 7 ON. Gene 4 is then turned OFF on the CLK, and gene 6 is reasserted. This situation explains the need for the CLK signal to remain active until the target STATE gene is activated.

3.1.3. Data Input (D)

The data input (D) introduces two more timing constraints. The first prohibits the introduction of the input too soon after the start of the CLK signal. While this might seem odd, it is in fact consistent with network behavior. Since gene 1 is only repressed by Gal4, it can only be repressed when the clock is ON. Therefore, when the clock is OFF, gene 1 is not prevented from continuously transcribing $\text{mCICr}$. Since translation on its own is faster than transcription followed by translation, when input D is introduced, it quickly induces the translation of $\text{mCICr}$, now unlocked. During that time, the CLK signal selects gene 5, but before

![Fig. 5](image_url) The BioD network described above is left unchanged except for one variable, $K_{\text{Lac}}$. Its effect on the input response time is highlighted for two complementary genes, $\text{LacI}$ and $\text{TetR}$. (a) Increasing the $K_{\text{Lac}}$ value from 0.5 nM to 14.0 nM increases the time it takes to start production of the $\text{mLacI}$ transcript in response to the proper input sequence. (b) As expected, increasing the $K_{\text{Lac}}$ value has the opposite effect on the production of the $\text{mTetR}$ transcript.

GAL4 has had a chance to be transcribed and then translated, the direct translation of the transcript of gene 1 into its corresponding (repressor) protein causes the repression of cII (by way of genes 2 and 5) and hence, the activation of gene 4. This ultimately results in an erroneous change of state as illustrated in Fig. 6b.

The second timing constraint occurs when input D is turned OFF. Indeed the level of expression of protein cII does not climb immediately. Time is needed to allow for the degradation of the cII protein, the cII2 molecule, and the unlocked mRNA molecule that are still in the system, in order to stop the production of more cII and allow the production (transcription and translation) of cII. Fig. 6c shows the CLK signal being activated too soon after input D is turned OFF. Since the system has not had enough time to reach equilibrium, it reacts to the clock as if its input was still ON.

The clock pulses must be sufficiently apart to allow the system to go to equilibrium (steady state) before the next pulse. Which SELECTION gene gets enabled depends heavily on that. Essentially, the input signal must stabilize (as ON or OFF), then the levels of cII and cII must stabilize as well, allowing the selection of one of the SELECTION genes, which must occur prior to the start of the clock pulse.

3.1.4. Bi-stability

A necessary feature of the BioD system is its bi-stability. Bi-stability means that the network is capable of being in any one of two steady states for as long as the inputs remain unchanged. This is a crucial feature because we do not want a BioD that is in (say) an ON state to autonomously switch to the OFF state, without any prompting from its input. Furthermore, we want these two steady states to be stable. Dynamically, a stable steady state is a basin of attraction with all nearby trajectories leading into it. In other words, the effect of small, non-sustained and/or noisy perturbations in the inputs are absorbed and do not prevent a return to the original stable steady state. This does not only apply to the STATE genes, but also to the SELECTION genes (which also form a toggle switch).

The conditions for toggle switch bi-stability have been discussed by Gardner et al. (2000), asserting that (i) the gene products must have a cooperative repression of transcription (Hill cooperativity) that is greater than 1; (ii) the rates of synthesis of the two repressors must be balanced (approximately equal). According to Gardner et al., these two conditions decide the size of the bi-stability region; where larger cooperative repressions and larger synthesis rates result in larger bi-stability areas. We add to these findings by including the effect of our CLK signal in relation to the genes used in the network. The results of our investigation resulted in a delineation of the region of bi-stability identical in general shape to the one discovered by Gardner et al., but having different exact boundaries.

In more detail, we varied the two $K_d$ values of the two STATE genes as well as the length the CLK pulse, while keeping all other parameter values constant. For every pair of $K_d$ values, we sought a minimum CLK pulse width that would result in a bi-stable network. In some cases, we found it, such as the green, yellow and red regions of Fig. 7, but in others – the black area – we did not. In other words, for all of these regions – except for the black one – a clock pulse whose length is equal or greater than the noted values would ensure a bi-stable behavior.

As can be seen, a smaller CLK pulse significantly reduces the range of $K_d$ values (and hence potential genes) that can be used to construct a bi-stable BioD. Extending the length of the CLK pulse too much, however, would not only be highly impractical, but would also mean that the state change is occurring across multiple reproductive cycles of an E. coli cell. It is therefore important to balance speed, practicality and the absolute need for bi-stability.

Fig. 7. Varying the $K_d$ values of the toggle switch genes while keeping all other parameters constant results in the above functional plot of the BioD. The BioD is said to be bi-stable (or functional) when it is able to toggle from one state to the other on the right inputs and is able to hold at that state indefinitely if unperturbed. The green zone, which is included in the yellow zone, which itself is included in the red zone all define the bi-stability regions of the BioD at CLK pulse widths of 25, 34 and 42 min, respectively. The black region denotes results of simulations that did not lead to a bi-stable network.

3.1.5. BioFSM

As previously described, the BioFSM is built by connecting the BioD to the U/FII and OI. Of these, the OI is the simplest module. It is in effect just an inverter that uses Q from the BioD to generate an AHL version of Q that is meant for inter-cellular signaling.

When LexA is ON (Q is ON) the AHL production is stopped; while when LexA is OFF (Q is OFF) AHL production is resumed. The U/FII is a variable module whose complexity depends on the desired functionality of the BioFSM. It can be as simple as the OI inverter or it can be an elaborate network that handles numerous inputs and performs complex combinatorial logic.

Figs. 8 and 9 display the stochastic simulations of all eight possible inputs to two U/FII's implementing $F = A + BC$ and $F = AB + AC$, respectively. The output $F$ of the U/FII is the input $D$ (or $\text{tar}12$) to the BioD. The inputs $A$, $B$ (or $\bar{B}$) and $C$ of the U/FII are the AHL, LacI (or TetR), and AHR, respectively. The core functionality of our BioFSM hinges on the manipulation of the input to the BioD incorporated within the BioFSM. We therefore highlight the proper functionality of the U/FII's that provide these inputs.

Here, we have one internal and two external inputs. The highlighted areas indicate the presence of an external input, while $m\text{Tet}R$ reflects the internal input $\bar{B} = Q$. The two opposing diagonal patterns reflect the presence of the left and right inputs A and C.

These simulations are presented as a table of diagrams, sorted by input presence, top to bottom, starting with the left column. We say $A = 0$ when AHL is not present, while $A = 1$ means that AHL is present in high quantities. Similarly, the values of $B$ and $C$ denote the presence and absence of LacI and AHLr respectively. As previously described in BioD, in our design LacI and TetR are complementary signals, which is why we consider $\bar{B} = \text{Tet}R$, and why we used it in the diagrams below.

The top left diagram displays the U/FII level at input $A B C = 0 0 0$ while the bottom right diagram displays that level at input $A B C = 1 1 1$. The top left diagram has no highlighted areas (i.e., no diagonal patterns) denoting the absence of the external inputs (AHL and AHLr). $m\text{Tet}R$ is present however, meaning $\bar{B} = 1$ (or $B = 0$), denoting the absence of LacI. Hence, this diagram displays the value of the U/FII, namely $F = A + BC$, with zero inputs, which is zero itself. The bottom right diagram has two areas of diagonal patterns (overlapping) denoting the presence of the external inputs. $m\text{Tet}R$ is absent meaning $\bar{B} = 0$ (or $B = 1$), denoting the presence of...
**4. Proposed Extension and Future Work**

An interesting extension to our designs, and purely as proposed future work, would be using multiple strains of BioFSMs connected in sequence to build circular cellular automata (CA), or BioCell.

A BioCell is presented as a ring of N colonies of E. coli. Each colony consists of clones of one of three strains, genetically modified to realize a BioFSM. The three strains implement the same logical functionality (same BioD and UF rules) but have different input and output interfaces (for inter-colony communications). We chose to connect these BioFSMs as ring cellular automata, i.e., each BioFSM is connected to its left and right BioFSM neighbors only (see Fig. 11c).

In effect, each colony will implement one type of BioFSM, and will communicate with its neighbor colonies via AHLs. Therefore the UF must have three inputs (two from its immediate neighbors, and one from itself). Each strain (BioFSM) needs to be able to recognize the origin of its inputs (to the UF/UI), and to broadcast a recognizable output (from its UI). The left-hand strain produces AHL and responds to AHLc and AHLr, while the centre strain produces AHLc and responds to AHL and AHLr, and so on... in order to function as expected. A colony processes its inputs to decide whether to alter its state, upon the application of a global clock pulse (FR light). The decision to change the state is made following the rules implemented by the UF. Those rules are the same for all strains, though with variations merely reflecting the chemical nature of the inputs with which each strain is confronted. A colony exhibits its state by expressing (or not) a florescent protein.

The BioCell will have the following dynamic behavior, determined by its inputs states of its N colonies (collectively making up the BioCell’s state). When there is no FR light (i.e., CLK = 0), there is no change in the state of the BioCell. In contrast, when the device receives a FR light pulse (i.e., CLK = 1) applied to all the colonies simultaneously, the next states of the colonies follow the outputs of their UF/UI (by processing its own and the neighbors’ states). The next clock pulse has to wait until the BioCell is back in equilibrium. Equilibrium, after an OFF to ON state change, comes after

---

Fig. 8. Stochastic simulation of all 8 possible inputs to function $F = A + BC$ (or Rule 248 from Wolfram, 2002). The external inputs A and C are AHL_left (AHLl) and AHL_right (AHLr), respectively. Their presence is highlighted by the grey diagonal patterns. The internal input B comes from the BioD. In this case, B was needed for the implementation of the UF (see Fig. 3b), so TetR was used and its respective mRNA level is displayed.

LacI. Hence, this diagram displays the value of the UF/UI, namely $F = A + BC$, with all inputs present. It is in fact doubly asserted by both A and BC and results in a higher production of tetR (representing F) than the other cases where it is asserted, at $ABC = 011$, 100, 101 and 110. The value of the input is highlighted in each diagram by three little squares in the bottom left area. The UF/UI module is designed as a non-synchronous module, but the BioFSM still functions synchronously using the embedded BioD clock.
the colonies have had a chance to produce enough AHLs and after those AHLs have diffused to the neighbors. Equilibrium, after an ON to OFF state change, comes after the AHLs produced by the colonies have had a chance to degrade. This is critical because the AHLs are the only indicator of the neighbors’ states. A clock pulse that comes before equilibrium might cause an erroneous change of state of the BioCell.

From a computational point of view, the BioCell device is a synchronous ring of cellular automata implemented as a ring of \( N \) communicating colonies of three new strains of \( E. coli \). Given this setup (three binary inputs and one binary output) for every BioSM in the BioCell, there exists \( 2^3 = 8 \) possible functions (or rules) that can be implemented by the UF. We chose two such rules to implement: (a) rule 248, as defined by Wolfram (2002), allows us to demonstrate signal propagation and counting behaviors, depending on the initial state of the ring; (b) rule 30, can be used as a pseudo random number generator or to exhibit cyclical behavior, depending on the initial state of the ring.

Some of the power of cellular automata is emphasized when rules exhibit different dynamic behaviors, i.e., chaotic, cyclical or fixed, by merely varying the initial conditions of the cellular automata. The particular UF used when introducing BioSM above, implements rule 248. Fig. 10(a) and (b) displays runs of this rule on a BioCell of 12 colonies. The change in the initial state results in two different behaviors, namely signal propagation in (a) and counting in (b). Rule 30, whose runs are displayed in Fig. 10(c)–(e), is an interesting rule that can result in either chaotic behavior as in (c), various cyclical behaviors such as (d) or simply lead the ring to a fixed state, as in (e).

This is a device that can be configured to perform many different functions using simple or no modifications (via change in initial conditions). Many cellular automata are capable of universal computation (Wolfram, 2002).

As this is only a proposed extension to the work presented in this paper, Fig. 10 is not the result of a simulation of the BioCell system, but merely a visualization of its possible results. For more detailed research on this proposed device, a simulation tool capable of handling multi-cell environments is needed. A good example is the Infobiotics Workbench (Blakes et al., 2011).

5. Conclusion and Critique

In this paper, we present a mathematical model and simulation results of a synchronous single-input delay flip-flop, realized as a gene regulatory network for implementation in *E. coli*. The simulation we present provides evidence that the device can toggle from the ON state to the OFF state and back, according to its intended functionality. The inherent symmetry of the design reduces the number of genes used, but introduces some complexity, which is palpable when tracing the various changes the device goes through when toggling.

The BioD is effectively a 1-bit memory element that can operate synchronously with any number of other elements. As such, it can be used to hold the state of a finite state machine, as it does in the BioFSM. It could also be used to build a memory bank, an event sequence detector/effectector, a decision-making system, and numerous other memory-requiring devices. The BioFSM is made of three modules: the BioD, the Update Function/Input Interface (UF/II) and the Output Interface (OI). The modular design of the BioFSM allows us to hold the BioD constant while changing the UF/II or/and OI, if and when the time-dependant behavior of the BioFSM, or its input/output interfaces require alteration. Then, there is the BioCell, which is made of a number of BioFSM colonies, and is capable of exhibiting a large number of computational, communicational and pattern formation behaviors depending on the particular UF and/or initial states of its constituent BioFSMs.

Speed is a main area of improvement. Indeed, the slowest reactions in a cell are the ones involving regulated transcription and translation. The time it takes to execute these operations depends on many factors, including various binding affinities, generation and degradation rates. For example, the impact of a repressor is delayed until a mature protein is formed and manages to interact with its corresponding operator site on the DNA. Using post-transcriptional regulation like tPOL or RNA interference (RNAi) – where possible – to affect regulation in the BioD will make the system significantly faster. One possible location for such a change would be where the SELECTION genes interact with the STATE genes. Instead of producing repressors for genes 6 or 7, the use of RNAi molecules to prevent the translation of repressor proteins would make the entire system significantly faster. However, since we already make use of *taR12* for input sensing, we would have to use two more riboregulators that do not interfere with *taR12* or with each other.

Another notable property of genetic networks is that the building blocks tend to vary significantly from one another, whether they be promoters, operators, or coding sequences – to name a few. That is to say, when designing a gene regulatory network, the choice of the building blocks is not easily exchangeable. In fact, the literature does not provide much in the way of “acceptable ranges” because most networks are presented as they are. In the case of dynamic and extendable circuits like BioD or BioFSM, that need is reasserted. Gene networks constitute highly interconnected graphs such that, for example, a repressor contributes to the functioning of the designed network by means of its dissociation constant (for a given operator), its rates of synthesis, diffusion and degradation, as well as the possibility of unintended (and often unexpected) cross-talk with the native DNA and constitutively generated molecules. We attempted to provide such “ranges” for our


