AIMS Bioengineering, 9(4): 400–419.
DOI: 10.3934/bioeng.2022029
Received: 10 November 2022
Revised: 07 December 2022
Accepted: 11 December 2022
Published: 28 December 2022

http://www.aimspress.com/journal/Bioengineering

Research article

Finite state machine and Markovian equivalents of the *lac* Operon in *E. coli* bacterium

Urooj Ainuddin*and Maria Waqas

Department of Computer and Information Systems Engineering, NED University of Engineering and Technology, Karachi, 75270, Pakistan

* **Correspondence:** Email: uroojain@neduet.edu.pk.

Abstract: The *lac* operon in *E. coli* has been extensively studied by computational biologists. The bacterium uses it to survive in the absence of glucose, utilizing lactose for growth. This paper presents a novel modeling mechanism for the *lac* operon, transferring the process of lactose metabolism from the cell to a finite state machine (FSM). This FSM is implemented in field-programmable gate array (FPGA) and simulations are run in random conditions. A Markov chain is also proposed for the *lac* operon, which helps study its behavior in terms of probabilistic variables, validating the finite state machine at the same time. This work is focused towards conversion of biological processes into computing machines.

Keywords: digital modeling; *lac* operon; bistable switch; finite state machine; Markov model

1. Introduction

The main source of nutrition for *E. coli* is glucose. However, bacteria exist and grow even when it is absent, because they use lactose in its place. The genetic region of bacterial DNA that manages lactose metabolism is known as the *lac* operon [22, 47]. An operon is a group of regulatory elements that affect the expression of more than one genes, producing a polycistronic mRNA strand [6]. Figure 1 depicts the genetic regions that constitute the *lac* operon in *E. coli*. Three genes, *lacZ*, *lacY* and *lacA*, are included in the *lac* operon. *lacZ* gene produces $\beta-$galactosidase. *lacY* gene produces Beta−galactoside permease. *lacA* gene produces Galactoside acetyltransferase (GAT). Beta−galactoside permease transports lactose into the bacterium [7]. $\beta-$galactosidase cleaves lactose to give glucose and galactose [33]. The role of GAT in lactose metabolism is yet unclear. Studies of the *lac* operon generally exclude *lacA* and its product from discussion. Upstream of the *lac* operon, the *lacI* gene produces the *lac* repressor [19]. The *lacI* gene is constitutively expressed [6].

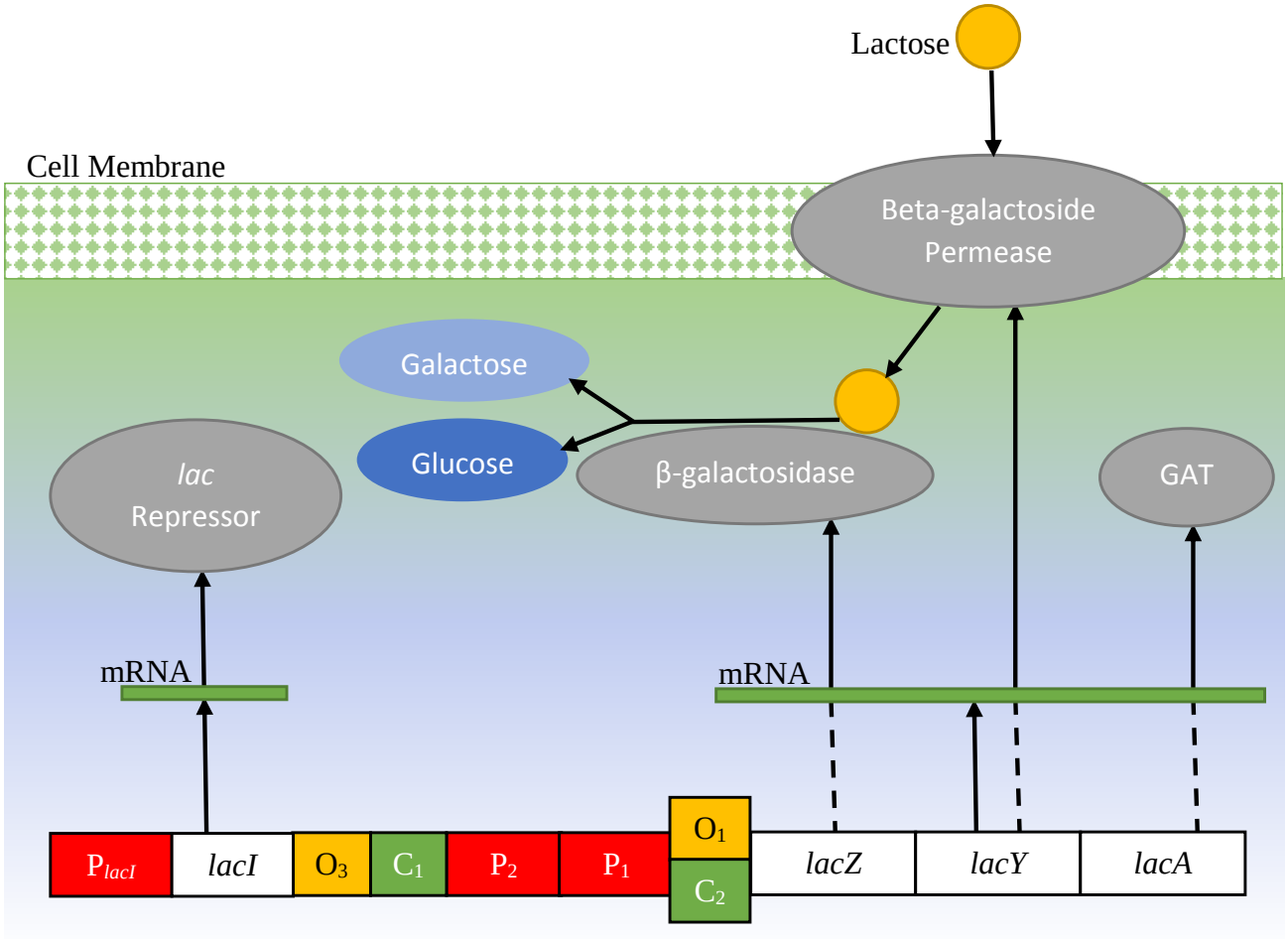

Figure 1. The *lac* operon includes three genes, *lacZ*, *lacY* and *lacA*. A fourth gene, *lacI*, encodes the *lac* repressor. (The figure has been adapted from Shuman and Silvahy [46].)

The *lac* operon is inactive when glucose is present for the bacterium to survive. Additionally, it remains inactive as long as the cellular environment does not have lactose to metabolize. The *lac* repressor (R), binds tightly to the operator O_1 (in Figure 1) and hinders RNA polymerase (RNAP) from transcribing the genes of the operon [6]. R also binds to the other two operators O_2 and O_3. However, binding of R to only O_2 or only O_3 does not deter the transcription of the *lac* genes [43].

The *lac* repressor normally occurs as a tetramer [6] and is an allosteric protein. When it binds with allolactose, it undergoes a conformational change that hinders its DNA binding site from binding to bacterial DNA [11]. Allolactose, a combination of galactose and glucose, is the inducer of the *lac* operon. It is produced at low levels by the few molecules of $\beta-$galactosidase that are present before induction.

The *lac* operon is functional when glucose is absent and lactose is present in the environment. As long as lactose remains present, $\beta-$galactosidase cleaves it into glucose and galactose to give allolactose, which keeps R from turning the operon off [6].

When glucose is absent from the system, the concentration of cyclic Adenosine MonoPhosphate (cAMP), goes up in the bacterium [32]. cAMP binds with the Catabolite Activator Protein (CAP) to create the cAMP-CAP complex. It is this complex that binds to the CAP binding sites, C_1 and C_2.

One bound, the cAMP-CAP complex activates and promotes transcription through interactions with RNA polymerase [4, 25]. The repression of the *lac* operon due to low cAMP levels in the presence of glucose is known as catabolite repression [52]. For brevity, the cAMP-CAP complex will be referred to as CAP only.

The *lac* operon acts as a bi-stable switch, by existing in one of two possible states, the operon being on and off [42]. Figure 2 depicts the regulatory elements that constitute the *lac* operon in *E. coli*.

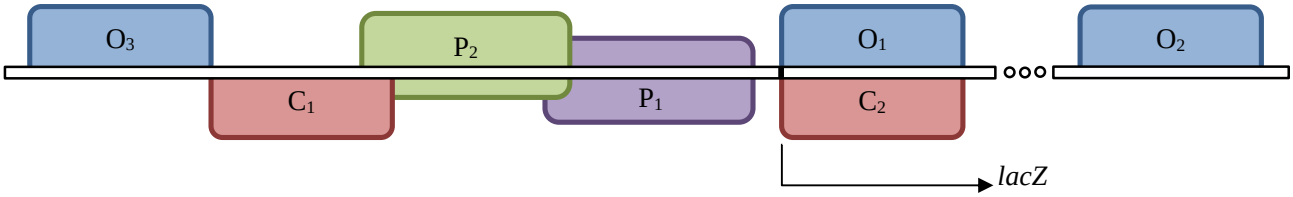

Figure 2. The *lac* operon includes three operators, O_1, O_2 and O_3, two promoter sites, P_1 and P_2, and two CAP binding sites, C_1 and C_2. (The figure has been adapted from Santillán and Mackey [43].)

O_3 lies at the end of *lacI* gene and O_2 lies in the early part of *lacZ* gene [35]. P_2 is considered a secondary promoter, as it does not play a significant role in transcription initiation [12, 39]. Mathematical models including that presented by [43] only consider the role of P_1 in the activation of the operon. O_3 and C_1 are adjacent to each other. R bound to O_3 bends DNA in the same direction as CAP bound to C_1 does [50]. If R binds to O_3, it is not possible for CAP to bind to C_1, and vice versa. O_1 and C_2 are at the same position but on opposite sides of the DNA chain. Hence, if R binds to O_1, it is not possible for CAP to bind to C_2, and vice versa [43].

The operators O_2 and O_3 play a major role in bringing about DNA folds. This happens as a result of a *lac* repressor tetramer binding to more than one operators [34, 35, 43]. R may bind simultaneously to O_1 and O_2. No transcription can take place because of the attachment of the repressor to O_1. R may bind simultaneously to O_1 and O_3. This leaves no space for RNAP to bind to the promoter. R may bind simultaneously to O_2 and O_3. Although the DNA strand is folded, RNAP can bind to the promoters. As O_1 is unoccupied, transcription can take place too.

A list of possible configurations of regulatory elements of the *lac* operon is drawn in Santillán and Mackey [43]. The authors developed a set of states $L = \{l_0, l_1, ..., l_{18}\}$ from the list given by [43] which is shown in Table 1. States are classified on the basis of the occupancy of O_1, O_3, C_1, C_2 and P_1. Each state name is composed of three characters. An unbound entity (operator, CAP binding site or promoter) is represented by **e**. A bound operator is represented by **r**. A bound CAP binding site is represented by **c**. A bound promoter is represented by **p**. In each state name, the first character represents O_3 or C_1, the second character represents P_1, and the last character represents O_1 or C_2. **fef** signifies that O_1 and O_3 are bound to the same tetramer of R, **f** representing a fold. It must be noted that R_F signifies the repressor tetramer bound simultaneously to two operators, bringing about a DNA fold. States in which RNAP is bound to the promoter are transcriptionally active states.

Table 1. Regulatory configurations of the *lac* operon.

S. No.	State Name	O_3	C_1	P_1	O_1	C_2	O_2
1	l_0: eee						
2							R
3	l_1: eer				R		
4					R		R
5					R_F		R_F
6	l_2: eec					C	
7						C	R
8	l_3: epe			RNAP			
9				RNAP			R
10	l_4: epr			RNAP	R		
11				RNAP	R		R
12				RNAP	R_F		R_F
13	l_5: epc			RNAP		C	
14				RNAP		C	R
15	l_6: ree	R					
16		R					R
17		R_F					R_F
18	l_7: rer	R			R		
19		R			R		R
20		R_F			R		R_F
21		R			R_F		R_F
22	l_8: rec	R				C	
23		R				C	R
24		R_F				C	R_F
25	l_9: rpe	R		RNAP			
26		R		RNAP			R
27		R_F		RNAP			R_F
28	l_{10}: rpr	R		RNAP	R		
29		R		RNAP	R		R
30		R_F		RNAP	R		R_F
31		R		RNAP	R_F		R_F
32	l_{11}: rpc	R		RNAP		C	
33		R		RNAP		C	R
34		R_F		RNAP		C	R_F
35	l_{12}: cee		C				
36			C				R
37	l_{13}: cer		C		R		
38			C		R		R
39			C		R_F		R_F

Table 1. Regulatory configurations (continued).

S. No.	State Name	O_3	C_1	P_1	O_1	C_2	O_2
40	l_{14}: cec		C			C	
41			C			C	R
42	l_{15}: cpe		C	RNAP			
43			C	RNAP			R
44	l_{16}: cpr		C	RNAP	R		
45			C	RNAP	R		R
46			C	RNAP	R_F		R_F
47	l_{17}: cpc		C	RNAP		C	
48			C	RNAP		C	R
49	l_{18}: fef	R_F			R_F		
50		R_F			R_F		R

2. Existing models of the *lac* Operon

The *lac* operon has been the focus of mathematical modeling ever since the conception of computational biology. Wong et al. developed a mathematical model of the *lac* operon which included catabolite repression, inducer exclusion, lactose hydrolysis to glucose and galactose, and synthesis and degradation of allolactose [52]. Mahaffy and Savev proposed a mathematical model for induction of the *lac* operon using biochemical kinetics [27]. Suen and Jacob presented a symbolic, grammar-based model for the operon in *Mathematica* [47]. Yildirim and Mackey proposed a mathematical model for the regulation of induction in the lac operon [55]. Yildirim et al. derived a reduced model from an existing model to explore the bi-stability of the operon [56]. Santillán and Mackey used a mathematical model to investigate the influence of catabolite repression and inducer exclusion on the bi-stable behavior of the operon [43]. Jacob and Burleigh developed a swarm-based, 3-dimensional model of the lactose operon gene regulatory system [21]. Santillán investigated bi-stability of the operon using a mathematical model [42]. Veliz-Cuba and Stigler presented a Boolean network as a discrete model for the *lac* operon [49]. Angelova and Ben-Halim proposed a deterministic model of the *lac* operon with a noise term, representing the stochastic nature of regulation [3]. Yildirim and Kazanci revisited the Yildirim-Mackey model [55] to show how deterministic and stochastic methods can be used to investigate various aspects of the operon [54]. Esmaeili et al. built a 3-dimensional, interactive computer model of the lactose operon using PROKARYO, an agent-based cell simulator [14]. Choudhary and Narang solved a detailed model of lac regulation using singular perturbation theory in [10]. Li et al. investigated the controllability of logical control networks and applied the obtained results to the reachability of the *lac* operon in *E. coli* [26].

3. Digital and Markov models of genetic circuits

FSMs are machines with memory [48]. A finite state machine is composed of two finite sets, S and E. S is the set of all possible states of the machine. E is the set of all possible events for the machine. At any time instant, the machine persists in one of the states of S, s_i, which is considered the current

state. When an event occurs, the machine transits from s_i to the next state, s_j. The transition from s_i to s_j depends on both s_i and the event that triggered the transition [29]. FSMs are immensely useful in modeling a closed environment which accepts external stimuli and reacts accordingly, changing itself in alignment to a set of rules. They can be programmed on a field-programmable gate array (FPGA) using hardware description language (HDL).

A stochastic process has value X_n ($n = 0, 1, 2...$) for each time instant n. $X_n = i$ represents that the process is in state i at time instant n. There is a fixed probability P_{ij}, which signifies transition of the process from state i to j. This process is called a Markov chain if the next state depends only on the current state, and never on states attained in the past [40].

$$P_{ij} = P\{X_{n+1} = j | X_n = i\} \tag{3.1}$$

Eq 3.1 signifies that the probability of any future state depends only on the present state. Markov chains are used to simulate randomly changing events.

Muri et al. used hidden Markov models to study bacterial genomes [30]. Krogh et al. described a membrane protein topology prediction method based on a hidden Markov model [24]. Markov chains were used to ponder the effect of individual genes on global dynamical network behavior [45]. Calder et al. analyzed signal transduction networks using continuous time Markov chains [8]. Julius et al. used Markov chains to model the lactose regulation system of *E. coli* [23]. Wang et al. studied the dynamic properties of the regulatory network governing lytic and lysogenic growths of coliphage lambda using a Markov chain stochastic model [51]. Vergne presented a drifting Markov model for the genetic content of lambda phage [31]. Yang et al. modeled proteins with the help of molecular finite automata (MFA) [53]. A Markov chain was presented for the process of target finding by lambda phage on the surface of *E. coli* [9]. Gao and Hu used FSMs to study gene expression [16]. Gao et al. defined a finite state machine and a hidden Markov model for genetic mutation [17]. FSMs were created using DNA [13]. Synthetic biologists created a framework to create FSMs using gene regulatory networks [36]. Shenker and Lin proposed Markov chains for cooperative binding in *E. coli*'s infection with lambda phage [44]. Fang et al. created a hidden Markov model for the λ switch [15]. A finite state machine and a Markov chain were created by Ainuddin et al. [1] for the lambda switch. This paper extends the same modeling technique to the *lac* operon.

4. Finite state machine design

The set L in Table 1 is used to create a finite state machine which transits in response to presence and absence of glucose and lactose. Six transitions have been defined for the finite state machine, shown in Table 2. Figure 3 depicts the binding transitions of the equivalent FSM. Figure 4 depicts the unbinding transitions of the equivalent FSM. The FSM is translated to a field-programmable gate array (FPGA) with the help of Verilog hardware description language (HDL). The implementation is named Digital *lac* Operon (**DLO**). The input signals of **DLO** are listed in Table 3. The output signals of **DLO** are listed in Table 4.

Table 2. Transitions of **DLO**.

Label	Transition
p	RNA polymerase binds to promoter
p′	RNA polymerase unbinds from promoter
r	R binds to one of the three operators
r′	R unbinds from one of the three operators
c	CAP binds to one of the two CAP binding sites
c′	CAP binds to one of the two CAP binding sites

Table 3. Inputs of **DLO**.

Name	Bits	Significance
enable	1	Enables the functioning of the machine
clock	1	An alternating sequence for synchronization
reset	1	Resets the machine to initial state
seed	8	Serves as seed for pseudo-random number generation
glu	1	Represents the presence of glucose in cellular environment
lac	1	Represents the presence of lactose in cellular environment

Table 4. Outputs of **DLO**.

Name	Bits	Significance
mem	3	Represents the contents of memory internal to the **Lac** module
state	5	Represents one of the elements of L from Table 1
out	2	Indicates transitional output of the *lac* operon

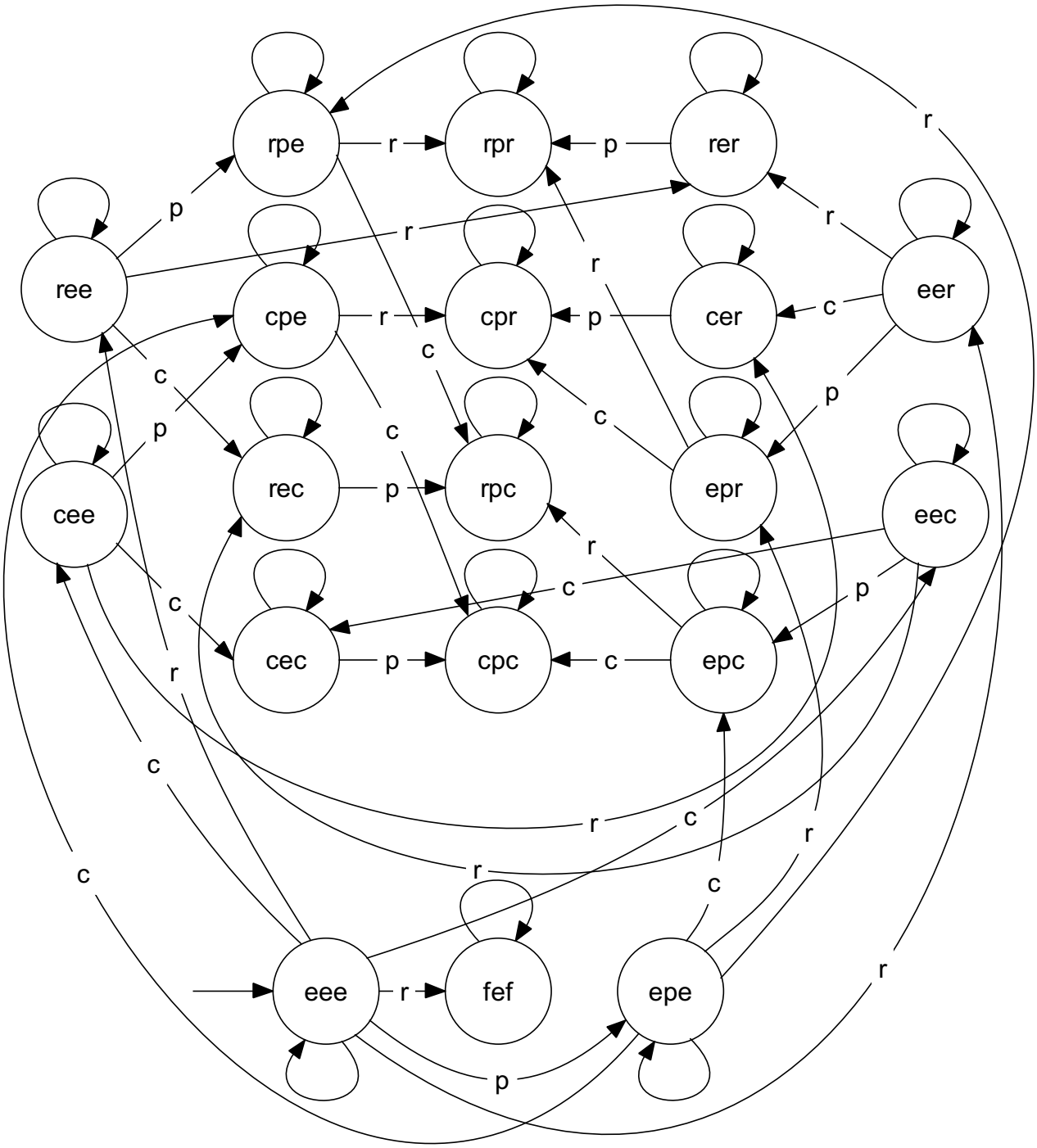

Figure 3. This state diagram only depicts the transitions in which R, CAP or RNAP binds to bacterial DNA. The self-loops signify binding of R to O_2.

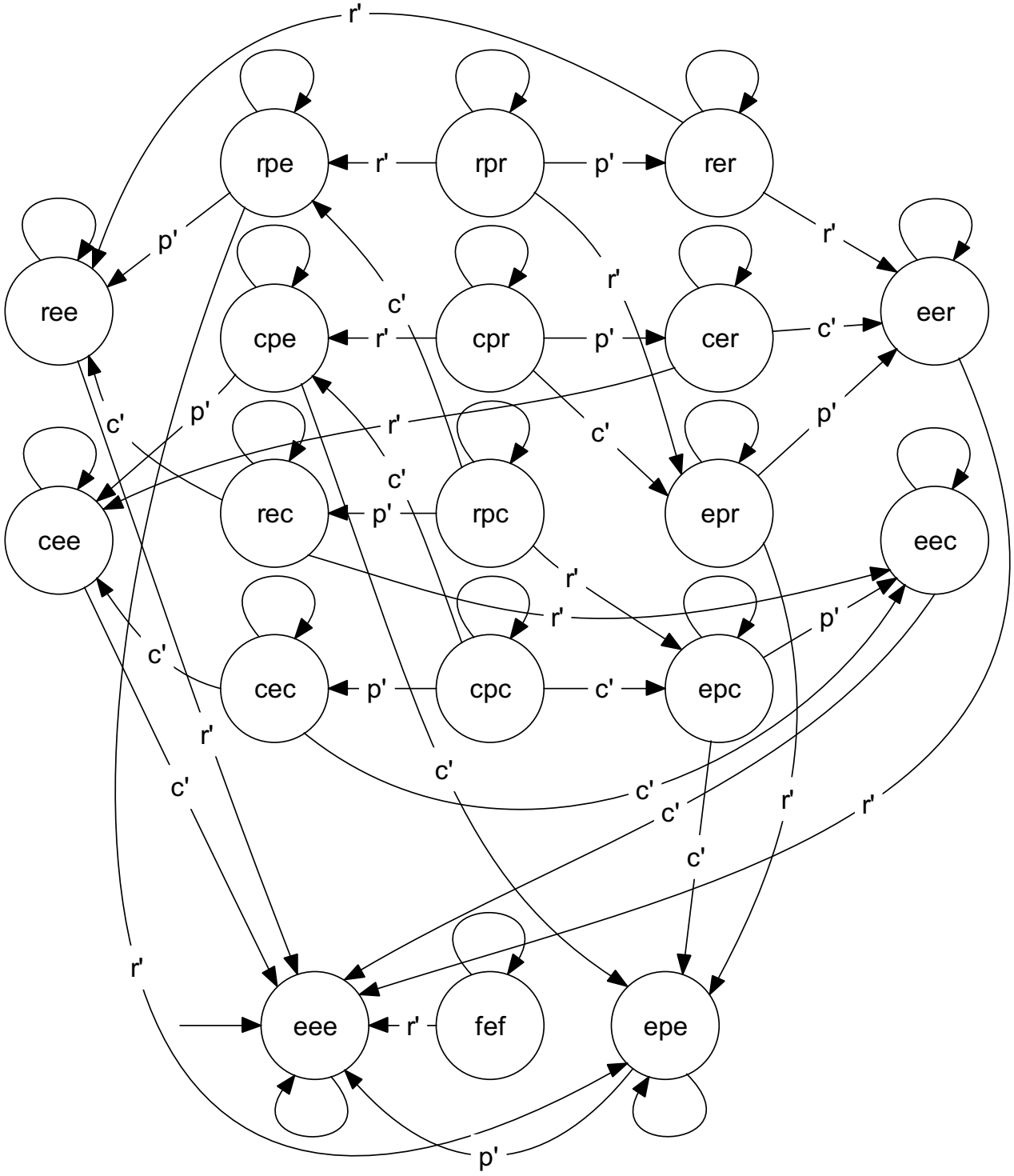

Figure 4. This state diagram only depicts the transitions in which R, CAP or RNAP unbinds from bacterial DNA. The self-loops signify unbinding of R from O_2.

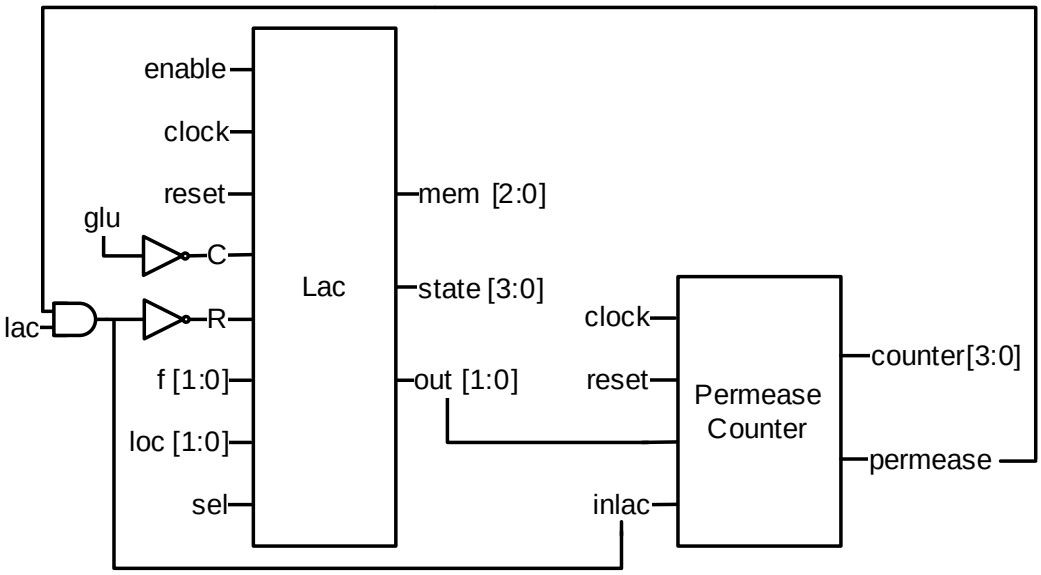

Figure 5. Internal circuit design of **DLO** incorporates a counter to keep count of the permease produced via transcription of *lacY* gene. The circuit shows how **Lac** module is interfaced with the counter.

Table 5. Additional inputs of the **Lac** module.

Name	Bits	Significance
C	1	Signifies the availability of CAP for binding
R	1	Signifies the availability of R for binding
f	2	Considered during repressor binding when multiple operators are available and DNA folding is possible
loc	2	Considered when multiple operators are available for binding or unbinding, or when both CAP binding sites are available for binding or unbinding
sel	1	Considered during unbinding when both R and CAP are bound to the DNA chain

When the initial state is either l_3 or l_9, and the machine transits with RNAP unbinding from the operon, $out = 01$. When the initial state is any one of l_5, l_{11} and l_{15}, and the machine transits with RNAP unbinding from the operon, $out = 10$. When the initial state is l_{17} and the machine transits with RNAP unbinding from the operon, $out = 11$. In all remaining transitions, $out = 00$. A module called **Lac**, and a permease counter are brought together in **DLO**. Figure 5 depicts the internal circuitry of **DLO**.

An 8-bit linear feedback shift register (LFSR) is used for pseudo-random number generation [2,37]. The feedback polynomial of this LFSR is $f(x) = x^8 + x^4 + 1$. The output byte is called **rand**. $sel = rand[0]$, $loc_1 = rand[1]$, $loc_0 = rand[2]$, $f_1 = rand[3]$ and $f_0 = rand[4]$.

All outputs of **DLO** are delivered by the **Lac** module. All inputs to **DLO** are fed on to the **Lac** module. Additional inputs to the **Lac** module are listed in Table 5.

Consider states l_0 and l_3, when $R = 1$, $C = 0$ and O_2 is unbound. If the **f** bit pair is 00, R binds to any one of the available repressors. If the **f** bit pair is 01, R binds to both O_2 and O_3. If the **f** bit pair is 10, R binds to both O_1 and O_2. If the **f** bit pair is 11, R binds to both O_1 and O_3. Consider states l_1,

l_2, $l_4 - l_6$, l_9, l_{12} and l_{15}, when $R = 1$, $C = 0$ and O_2 is unbound. If $f_0 = 0$, R binds to any one of the available operators. Otherwise, R binds to two operators, one of which is O_2, creating a DNA fold.

Consider states l_0 and l_3, when $R = 1$, $C = 0$, O_2 is unbound and $f = 00$. If the **loc** bit pair is 00, R binds to O_3. If the **loc** bit pair is 01, R binds to O_1. If $loc_1 = 1$, R binds to O_2. Consider states l_7 and l_{10}, when $R = 0$, O_2 is bound and no DNA fold exists. If the **loc** bit pair is 00, R unbinds from O_3. If the **loc** bit pair is 01, R unbinds from O_1. If $loc_1 = 1$, R unbinds from O_2. loc_0 is used in all states of L, when two operators are available for binding, or two repressor tetramers are bound to the operon, or two CAP binding sites are available for binding, or two cAMP-CAP complexes are bound to the operon. If $loc_0 = 0$, the available operator closest to the *lacI* gene is selected for binding or unbinding. Otherwise, the available operator farthest from the *lacI* gene is selected. If $loc_0 = 0$, C_1 is selected, otherwise C_2.

The **sel** bit is used to select R or CAP in states l_2, l_5, l_8, and $l_{11} - l_{17}$, when at least one instance of both is attached to the operon. If $sel = 0$, R is removed, otherwise CAP is removed.

A 3−bit memory register called **mem** is defined inside the **Lac** module. O_2 is represented by the most significant bit. If O_2 is unbound, its relevant bit is 0, otherwise, 1. If DNA is not folded, the rightmost bits are 00. If O_2 and O_3 are bound to the same R tetramer, the rightmost bits are 01. If O_1 and O_2 are bound to the same R tetramer, the rightmost bits are 10. If O_1 and O_3 are bound to the same R tetramer, this is not represented by **mem**, but by the state itself (l_{18}).

DLO employs an up-down counter to keep track of the permease transcribed. It is a hexadecimal counter of 4 bits. Table 6 depicts the inputs of the permease counter. Table 7 depicts the outputs of the permease counter.

Table 6. Inputs of the permease counter.

Name	Bits	Significance
clock	1	An alternating sequence for synchronization
reset	1	Resets the machine to initial state
out	2	Indicates transitional output of the *lac* operon
inlac	1	Represents the presence of lactose inside the cell

Table 7. Outputs of the permease counter.

Name	Bits	Significance
counter	4	Indicates the current count of permease
permease	1	Represents the presence of permease inside the cell

The bits of **out** are ORed to create a signal which advances the counter when HIGH. The **inlac** bit decreases the counter when HIGH. If the 4−bit output **counter** value is greater than $0H$, the output bit **permease** is HIGH.

The memory **mem** is reset at initialization. The **Lac** module is initialized to state l_0. The permease counter is initialized to $0H$. The FSM was synthesized on Xilinx Virtex-4 FPGA using ISE Design Suite from Xilinx Design Tools. The machine runs for a simulation time of $1\mu s$. Time resolution is $1ps$. Simulation results for $seed = FEH$ are summarized in Table 8.

The complete truth table of the **DLO** appears at the end of this paper as Appendix A.

Table 8. Simulation summary for **DLO** with $seed = FEH$.

Time (ps)	Glucose	Lactose	Observations
0 – 190	Absent	Absent	The operon is on. RNAP repeatedly transcribes the operon's genes. As a result, both $\beta-$galactosidase and Beta−galactoside permease are produced.
215 – 410	Present	Absent	The operon is off. R binds to O_2 and O_3, causing a DNA fold. Another R tetramer binds to O_1. Even if RNAP binds to the DNA, transcription initiation is not possible.
415 – 610	Absent	Absent	The operon is off. The repressor-DNA bindings remain intact. Even if RNAP binds to the DNA, transcription initiation is not possible.
615 – 810	Absent	Present	The operon is on. R unbinds from O_1 and CAP binds to C_2. RNAP binds to the operon repeatedly, and transcription takes place at an elevated rate.
815 – 990	Present	Present	The operon is off. CAP unbinds from C_2. R binds to O_2 and O_3, causing a DNA fold. Another R tetramer binds to O_1. Even if RNAP binds to the DNA, transcription initiation is not possible.

5. Markov chain for the *lac* operon

We define α as the probability of R binding to the operon, β as the probability of CAP binding to the operon and γ as the probability of RNAP binding to the operon. We also define the following probabilities:

$$\alpha' = 1 - \alpha \tag{5.1}$$

$$\beta' = 1 - \beta \tag{5.2}$$

$$\gamma' = 1 - \gamma \tag{5.3}$$

Figure 6 depicts the transition matrix M for the *lac* operon. This is a left stochastic square matrix.

5.1. Steady state analysis of M

A Markov model attains steady state when the probability of the system to exist in any state does not change with t or n, as $n \to \infty$, as given by Eqs 5.4,5.5 [18]. For a scalar value ψ, eigenvector e for a matrix T satisfies Eq 5.6 [41]. We can deduce that $v_{ss} = e$ when $M = T$ and $\psi = 1$. The normalized eigenvector corresponding to unity eigenvalue is shown in Eq 5.7.

$$M\, v_{ss} = v_{ss} \tag{5.4}$$

$$v_{ss} = \{v_0, v_1, ..., v_{k-1}\}^t \tag{5.5}$$

$$T\, e = \psi\, e \tag{5.6}$$

$$v_{ss} = \frac{e}{\sum_{i=0}^{k-1} e_i} \tag{5.7}$$

Table 9 details configurations of regulatory elements that lead to operon induction [43]. We refer to the steady state probability of activation as p_{acss}.

$$p_{acss} = v_3 + v_5 + v_9 + v_{11} + v_{15} + v_{17} \tag{5.8}$$

$$\begin{bmatrix}
\alpha'\beta'\gamma' & \frac{\alpha'\beta'\gamma'}{2} & \frac{\alpha'\beta'\gamma'}{2} & \gamma' & 0 & 0 & \frac{\alpha'\beta'\gamma'}{2} & 0 & 0 & 0 & 0 & 0 & \frac{\alpha'\beta'\gamma'}{2} & 0 & 0 & 0 & 0 & 0 & \alpha' \\
\frac{\alpha\gamma'}{3} & \frac{\gamma'}{2} & 0 & 0 & \gamma' & 0 & 0 & \frac{\alpha'\gamma'}{2} & 0 & 0 & 0 & 0 & 0 & \alpha\beta'\gamma' & 0 & 0 & 0 & 0 & 0 \\
\frac{\alpha'\beta\gamma'}{2} & 0 & \frac{\gamma'}{2} & 0 & 0 & \gamma' & 0 & 0 & \alpha'\gamma' & 0 & 0 & 0 & 0 & 0 & \frac{\beta'\gamma'}{2} & 0 & 0 & 0 & 0 \\
\gamma & 0 & 0 & \alpha'\beta'\gamma & \frac{\alpha'\beta'\gamma}{2} & \frac{\alpha'\beta'\gamma}{2} & 0 & 0 & 0 & \frac{\alpha'\beta'\gamma}{2} & 0 & 0 & 0 & 0 & 0 & \frac{\alpha'\beta'\gamma}{2} & 0 & 0 & 0 \\
0 & \gamma & 0 & \frac{\alpha\gamma}{2} & \frac{\gamma}{2} & 0 & 0 & 0 & 0 & 0 & \frac{\alpha'\gamma}{2} & 0 & 0 & 0 & 0 & 0 & \alpha\beta'\gamma & 0 & 0 \\
0 & 0 & \gamma & \frac{\alpha'\beta\gamma}{2} & 0 & \frac{\gamma}{2} & 0 & 0 & 0 & 0 & 0 & \alpha'\gamma & 0 & 0 & 0 & 0 & 0 & \frac{\beta'\gamma}{2} & 0 \\
\frac{\alpha\gamma'}{3} & 0 & 0 & 0 & 0 & 0 & \frac{\gamma'}{2} & \frac{\alpha'\gamma'}{2} & \alpha\beta'\gamma' & \gamma' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\alpha\gamma'}{2} & 0 & 0 & 0 & 0 & \frac{\alpha\gamma'}{2} & \alpha\gamma' & 0 & 0 & \gamma' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{\alpha\gamma'}{2} & 0 & 0 & 0 & \frac{\alpha'\beta\gamma'}{2} & 0 & \alpha\beta\gamma' & 0 & 0 & \gamma' & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\alpha\gamma}{2} & 0 & 0 & \gamma & 0 & 0 & \frac{\gamma}{2} & \frac{\alpha'\gamma}{2} & \alpha\beta'\gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{\alpha\gamma}{2} & 0 & 0 & \gamma & 0 & \frac{\alpha\gamma}{2} & \alpha\gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{\alpha\gamma}{2} & 0 & 0 & \gamma & \frac{\alpha'\beta\gamma}{2} & 0 & \alpha\beta\gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\alpha'\beta\gamma'}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\gamma'}{2} & \alpha'\gamma' & \frac{\beta'\gamma'}{2} & \gamma' & 0 & 0 & 0 \\
0 & \frac{\alpha'\beta\gamma'}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\alpha\gamma'}{2} & \alpha\beta\gamma' & 0 & 0 & \gamma' & 0 & 0 \\
0 & 0 & \frac{\alpha'\beta\gamma'}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\alpha'\beta\gamma'}{2} & 0 & \beta\gamma' & 0 & 0 & \gamma' & 0 \\
0 & 0 & 0 & \frac{\alpha'\beta\gamma}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & \frac{\gamma}{2} & \alpha'\gamma & \frac{\beta'\gamma}{2} & 0 \\
0 & 0 & 0 & 0 & \frac{\alpha'\beta\gamma}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & \frac{\alpha\gamma}{2} & \alpha\beta\gamma & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{\alpha'\beta\gamma}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & \frac{\alpha'\beta\gamma}{2} & 0 & \beta\gamma & 0 \\
\frac{\alpha\gamma'}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha
\end{bmatrix}$$

Figure 6. The transition matrix M for Markov chain of the *lac* operon is a 19×19 left stochastic matrix.

Table 9. Machine states leading to operon activation.

State No.	State name	Configurations mapped (S. No. from Table 1)
l_3	epe	19,20
l_5	epc	31,32
l_9	rpe	21,22,45
l_{11}	rpc	33,34,50
l_{15}	cpe	23,24
l_{17}	cpc	35,36

6. Discussion

The inverter at input **C** of the **Lac** module is the digital manifestation of catabolite repression.

In the first 200 epochs ($0 - 190\ ps$) of simulation (as shown in Table 8), the **DLO** is functional even in the absence of both glucose and lactose. The cell is starving, and some $\beta-$galactosidase, still present from the operon's previous activity, produces a small amount of allolactose, which inactivates the *lac* repressor. This cause the operon to be induced.

The top-left subplot of Figure 7 depicts the condition where $\gamma = 0$. This implies that RNAP does not bind to the promoter. When $\gamma \neq 0$, the probability of activation always remains less than or equal to γ, as can be seen from the Figure 7. Figure 7 clearly displays that bacterial cultures with higher growth rates have their operons transitioning swiftly to full induction. This is a positive feedback.

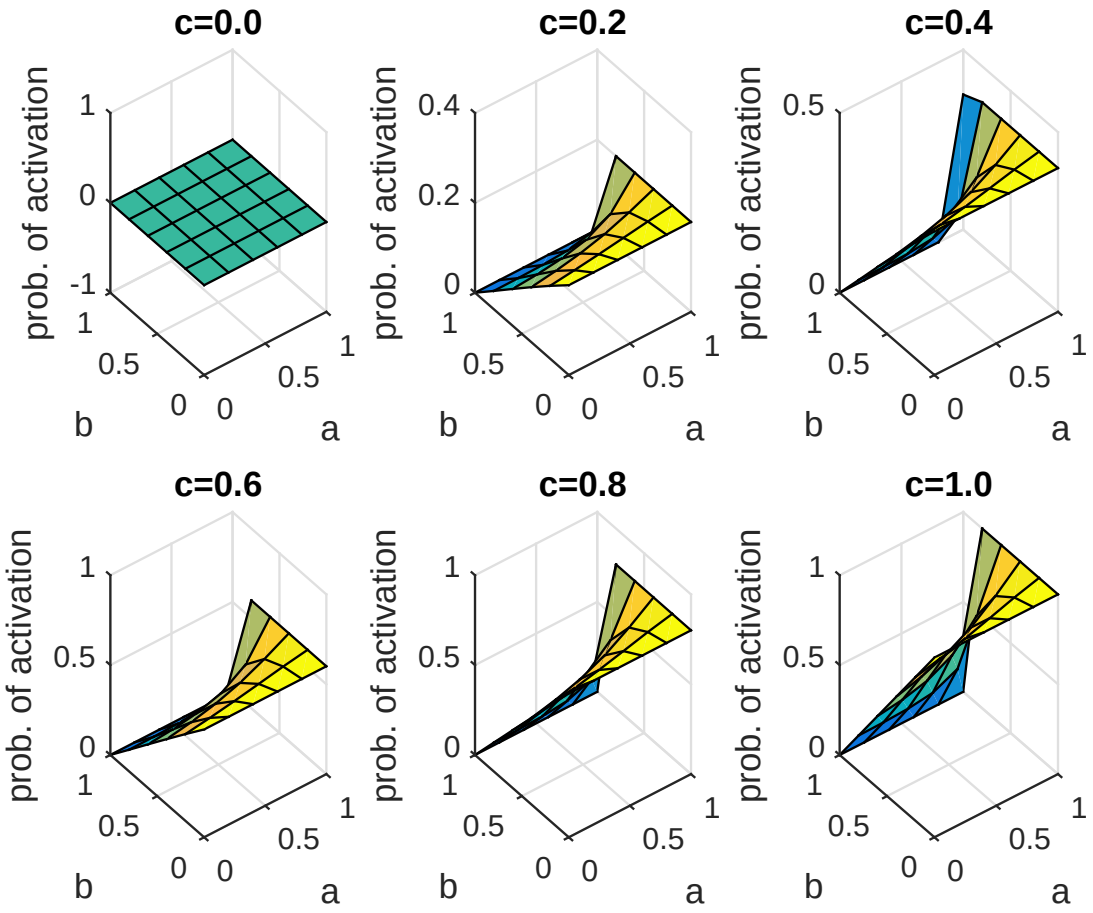

Figure 7. Surface plots are created for steady state probability of activation, p_{acss}. **a** represents α, **b** represents β, and **c** represents γ.

In Figure 8, p_{acss} decreases with α. This means that as the concentration of the *lac* repressor increases, the operon transitions to the off state.

In Figure 9, p_{acss} increases with β. This means that as the concentration of the cAMP-CAP complex increases, the operon transitions to full induction. Figure 9 shows a highly activated operon for lower values of α, or lower concentration of R.

6.1. Comparable results from existing literature

Wong et al. have reported similar findings regarding concentrations of the *lac* repressor, $\beta-$galactosidase and Beta−galactoside permease [52]. Yildirim and Mackey have reported similar findings regarding concentrations of lactose and $\beta-$galactosidase [55]. Santillán and Mackey furnish graphs conveying the same information as Figures 8,9 [43]. Esmaeili et al. produce similar results from their model of the operon [14].

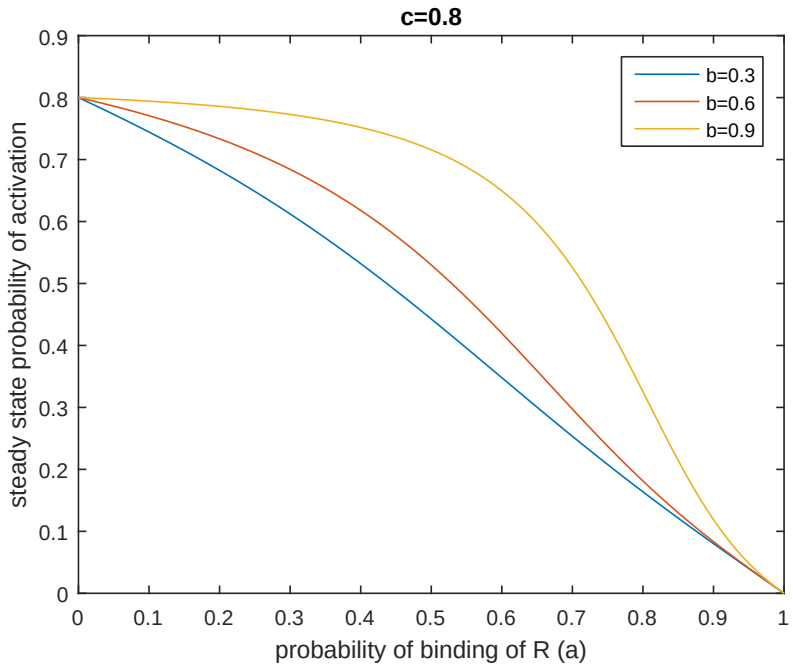

Figure 8. Steady state probability of activation, p_{acss}, is plotted for $\gamma = 0.8$. Three values of β are used, namely 0.3, 0.6 and 0.9. For each of these values, p_{acss} is plotted against α. **a** represents α, **b** represents β, and **c** represents γ.

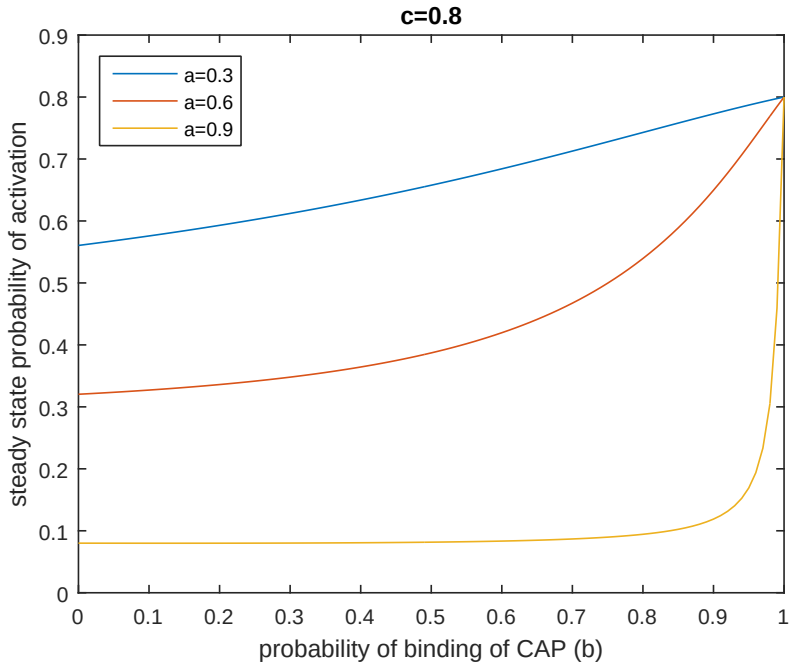

Figure 9. Steady state probability of activation, p_{acss}, is plotted for $\gamma = 0.8$. Three values of α are used, namely 0.3, 0.6 and 0.9. For each of these values, p_{acss} is plotted against β. **a** represents α, **b** represents β, and **c** represents γ.

7. Conclusion

This work focuses on digital and stochastic counterparts to the *lac* operon and translates the process from the cytoplasm to silicon. FPGA implementation of the developed FSM is simulated and documented, creating a silicon mimetic [20, 38]. The FSM is found to bear integrity with the modeled genetic switch. A Markov chain with the same set of states as the FSM, undergoing the same set of transitions, furnishes stochastic results that are in line with experimental data.

Although the *lac* operon has been the subject of mathematical modeling in numerous other works, this paper sets itself apart because it converts it to a digital machine. This is an emerging approach to modeling biological circuits. Other ventures in digital modeling of the *lac* operon have been listed in Section 2, but the techniques used are all together different from this research. It should be noted that in [28], the circuit diagram proposed is genetic (and not electronic) in nature. In [5], the modeling approach is not Boolean; however it is logical and uses discrete variables and functions. Table 10 clearly shows the novelty of modeling presented in this work. This work helps to bring the genetic switch to electronic hardware, where it can be tested in the same ways as an electronic circuit. Although this work restricts itself to a genetic switch, the modeling strategy can be adapted to genetic processes with more than two stable behaviors.

Table 10. Analysis of models proposed for the *lac* operon.

Work	Type of model developed						
	Boolean	Mathematical	Stochastic	Computer simulation	State transition graphs	Circuit diagram	Digitial
[52]	No	Yes	No	No	No	No	No
[27]	No	Yes	No	No	No	No	No
[47]	No	No	No	Yes	No	No	No
[55]	No	Yes	No	No	No	No	No
[56]	No	Yes	No	No	No	No	No
[43]	No	Yes	No	No	No	No	No
[21]	No	No	No	Yes	No	No	No
[42]	No	Yes	No	No	No	No	No
[49]	Yes	No	No	No	Yes	No	No
[3]	No	Yes	No	No	No	No	No
[54]	No	Yes	Yes	No	No	No	No
[14]	No	No	No	Yes	No	No	No
[10]	No	No	Yes	No	No	No	No
This work	Yes	No	Yes	No	Yes	Yes	Yes

Conflict of interest

The Authors declare that there is no conflict of interest.

References

1. Ainuddin U, Khurram M, Hasan SMR (2019) Cloning the λ Switch: digital and Markov representations. *IEEE Trans NanoBiosci* 18: 428–436. https://doi.org/10.1109/TNB.2019.2908669
2. Alspector J, Gannett J W, Haber S, et al. (1990) Generating multiple analog noise sources from a single linear feedback shift register with neural network application. *ISCAS* 1990: 1058–1061.
3. Angelova M and Ben-Halim A (2011) Dynamic model of gene regulation for the lac operon. *J Phys Conf Ser* 286: 012007. https://doi.org/10.1088/1742-6596/286/1/012007
4. Balaeff A, Mahadevan L, Schulten K (2004) Structural basis for cooperative DNA binding by CAP and lac repressor. Structure *Structure* 12: 123–132. https://doi.org/10.1016/j.str.2003.12.004
5. Bérenguier D, Chaouiya C, Monteiro P T, et al. (2013) Dynamical modeling and analysis of large cellular regulatory networks *Chaos* 23: 025114. https://doi.org/10.1063/1.4809783
6. Berg JM, Tymoczko JL, Stryer L, et al. (2002) *Biochemistry*, 5th edition, New York: W. H. Freeman.
7. Büchel DE, Gronenborn B, Müller-Hill B (1980) Sequence of the lactose permease gene. *Nature* 283: 541–545. https://doi.org/10.1038/283541a0
8. Calder M, Vyshemirsky V, Gilbert D, et al. (2006) Analysis of signalling pathways using continuous time Markov chains, In: Corrado Priami and Gordon Plotkin, *Transactions on Computational Systems Biology VI*, Heidelberg: Springer, 44–67.
9. Chatterjee S and Rothenberg E (2012) Interaction of bacteriophage λ with its E. coli receptor, lamB. *Viruses* 4: 3162–3178. https://doi.org/10.3390/v4113162
10. Choudhary K and Narang A (2019) Analytical expressions and physics for single-cell mRNA distributions of the lac Operon of E. coli. *Biophys J* 117: 572–586. https://doi.org/10.1016/j.bpj.2019.06.029
11. Daber R, Stayrook S, Rosenberg A, et al. (2007) Structural analysis of lac repressor bound to allosteric effectors. *J Mol Biol* 370: 609–619. https://doi.org/10.1016/j.jmb.2007.04.028
12. Donnelly CE and Reznikoff WS (1987) Mutations in the lac P2 promoter. *J Bacteriol* 169: 1812–1817. https://doi.org/10.1128/jb.169.5.1812-1817.1987
13. Eshra A and El-Sayed A (2014) An odd parity checker prototype using DNAzyme finite state machine. *IEEE/ACM Trans Comput Biol Bioinform* 11: 316–324. https://doi.org/10.1109/TCBB.2013.2295803
14. Esmaeili A, Davison T, Wu A, et al. (2015) PROKARYO: an illustrative and interactive computational model of the lactose operon in the bacterium Escherichia coli. *BMC Bioinformatics* 16: 1–23. https://doi.org/10.1186/s12859-015-0720-z
15. Fang X, Liu Q, Bohrer C, et al. (2018) Cell fate potentials and switching kinetics uncovered in a classic bistable genetic switch. *Nat Commun* 9: 2787. https://doi.org/10.1038/s41467-018-05071-1
16. Gao R and Hu W (2012) Study on some modeling problems in the process of gene expression with finite state machine. *10th World Congress on Intelligent Control and Automation (WCICA)*, IEEE, 2012: 5066–5070. https://doi.org/10.1109/WCICA.2012.6359438

17.Gao R, Hu W, Tarn TJ (2013) The application of finite state machine in modeling and control of gene mutation process. *IEEE Trans NanoBiosci* 12: 265–274. https://doi.org/10.1109/TNB.2013.2260866

18.Gebali F (2008) *Analysis of Computer and Communication Networks*, US: Springer, 2008. https://doi.org/10.1007/978-0-387-74437-7

19.Griffiths AJ, Gelbart WM, Miller JH, et al. (1999) *Darwin's Revolution. Modern Genetic Analysis.* New York: W. H. Freeman, 1999.

20.Hasan SMR (2010) A digital cmos sequential circuit model for bio-cellular adaptive immune response pathway using phagolysosomic digestion: a digital phagocytosis engine. *J Biomed Sci Eng* 3: 470–475. https://doi.org/10.4236/jbise.2010.35065

21.Jacob C and Burleigh I (2004) Biomolecular swarms–an agent-based model of the lactose operon. *Nat Comput* 3: 361–376. https://doi.org/10.1007/s11047-004-2638-7

22.Jacob F and Monod J (1961) Genetic regulatory mechanisms in the synthesis of proteins. *J Mole Biol* 3: 318–356. https://doi.org/10.1016/S0022-2836(61)80072-7

23.Julius AA, Halasz A, Kumar V, et al. (2006) Finite state abstraction of a stochastic model of the lactose regulation system of Escherichia coli. *Proceedings of the 45th IEEE Conference on Decision and Control*, IEEE, 2006: 19–24. https://doi.org/10.1109/CDC.2006.376739

24.Krogh A, Larsson B, Von Heijne G, et al. (2001) Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J Mole Biol* 305: 567–580. https://doi.org/10.1006/jmbi.2000.4315

25.Lawson CL, Swigon D, Murakami KS, et al. (2004) Catabolite activator protein: DNA binding and transcription activation. *Curr Opin Struct Biol* 14: 10–20. https://doi.org/10.1016/j.sbi.2004.01.012

26.Li H, Wang S, Li X, et al. (2020) Perturbation analysis for controllability of logical control networks. *SIAM J Control Optim* 58: 3632–3657. https://doi.org/10.1137/19M1281332

27.Mahaffy JM and Savev ES (1999) Stability analysis for a mathematical model of the lac operon. *Quart Appl Math* 57: 37–53. https://doi.org/10.1090/qam/1672171

28.McAdams HH and Shapiro L (1995) Circuit simulation of genetic networks. *Science* 269: 650–656. https://doi.org/10.1126/science.762479

29.Moore EF (1956) Gedanken-experiments on sequential machines. *Automata Studies* 34: 129–153. https://doi.org/10.1515/9781400882618-006

30.Muri F (1998) Modelling bacterial genomes using hidden Markov models. *COMPSTAT* 1998: 89–100. https://doi.org/10.1007/978-3-662-01131-7-8

31.Vergne N (2008) Drifting Markov models with polynomial drift and applications to DNA sequences. *Stat Appl Genet Mol Biol* 7: 6. https://doi.org/10.2202/1544-6115.1326

32.Notley-McRobb L, Death A, Ferenci T (1997) The relationship between external glucose concentration and cAMP levels inside Escherichia coli: implications for models of phosphotransferase-mediated regulation of adenylate cyclase. *Microbiology* 143: 1909–1918. https://doi.org/10.1099/00221287-143-6-1909

33.Novick A and Weiner M (1957) Enzyme induction as an all-or-none phenomenon. *Proc Natl Acad Sci USA* 43: 553–566. https://doi.org/10.1073/pnas.43.7.553

34.Oehler S, Amouyal M, Kolkhof P, et al. (1994) Quality and position of the three lac operators of E. coli define efficiency of repression. *EMBO J* 13: 3348–3355. https://doi.org/10.1002/j.1460-2075.1994.tb06637.x

35.Oehler S, Eismann ER, Krämer H, et al. (1990) The three operators of the lac operon cooperate in repression. *EMBO J* 9: 973–979. https://doi.org/10.1002/j.1460-2075.1990.tb08199.x

36.Oishi K and Klavins E (2014) Framework for engineering finite state machines in gene regulatory networks. *ACS Synth Biol* 3: 652–665. https://doi.org/10.1021/sb4001799

37.Rani MJ and Malarkkan S (2012) Design and analysis of a linear feedback shift register with reduced leakage power. *Int J Comput Appl* 56: 9–13. https://doi.org/10.5120/8957-3159

38.Hasan SMR (2008) A novel mixed-signal integrated circuit model for DNA-protein regulatory genetic circuits and genetic state machines. *IEEE Trans Circuits Syst Regul Pap* 55: 1185–1196. https://doi.org/10.1109/TCSI.2008.925632

39.Reznikoff WS (1992) Catabolite gene activator protein activation of lac transcription. *J Bacteriol* 174: 655–658. https://doi.org/10.1128/jb.174.3.655-658.1992

40.Ross SM (2010) *Introduction to Probability Models*, 10th edition, Boston: Academic Press, ocn444116127. https://doi.org/10.1016/B978-0-12-375686-2.00007-8

41.Saad Y (2011) *Numerical methods for large eigenvalue problems: revised edition,* USA: SIAM. https://doi.org/10.1137/1.9781611970739

42.Santillán M (2008) Bistable behavior in a model of the lac operon in Escherichia coli with variable growth rate. *Biophys J* 94: 2065–2081. https://doi.org/10.1529/biophysj.107.118026

43.Santillán M and Mackey MC (2004) Influence of catabolite repression and inducer exclusion on the bistable behavior of the lac operon. *Biophys J* 86: 1282–1292. https://doi.org/10.1016/S0006-3495(04)74202-2

44.Shenker JQ and Lin MM (2015) Cooperativity leads to temporally-correlated fluctuations in the bacteriophage lambda genetic switch. *Front Plant Sci* 6: 214. https://doi.org/10.3389/fpls.2015.00214

45.Shmulevich I, Dougherty ER, Zhang W (2002) From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proc IEEE* 90: 1778–1792. https://doi.org/10.1109/JPROC.2002.804686

46.Shuman HA and Silhavy TJ (2003) The art and design of genetic screens: Escherichia coli. *Nat Rev Genet* 4: 419–431. https://doi.org/10.1038/nrg1087

47.Suen G and Jacob C (2003) A symbolic and graphical gene regulation model of the lac operon. In: MiticP, Carne J, Ramsden P, *Challenging the Boundaries of Symbolic Computation*, London: Imperial College, 73–80. https://doi.org/10.1142/9781848161313-0010

48.Floyd TL (2015) *Digital Fundamentals*, 11th edition, England: Pearson Education.

49.Veliz-Cuba A and Stigler B (2011) Boolean models can explain bistability in the lac operon. *J Comput Biol* 18: 783–794. https://doi.org/10.1089/cmb.2011.0031

50.Vossen KM, Stickle DF, Fried MG (1996) The Mechanism of CAP-lacRepressor Binding Cooperativity at the E. coliLactose promoter. *J Mol Biol* 255: 44–54. https://doi.org/10.1006/jmbi.1996.0005

51. Wang S, Zhang Y, Ouyang Q (2006) Stochastic model of coliphage lambda regulatory network. *Phys Rev E* 73: 041922. https://doi.org/10.1103/PhysRevE.73.041922

52. Wong P, Gladney S, Keasling JD (1997) Mathematical model of the lac operon: inducer exclusion, catabolite repression, and diauxic growth on glucose and lactose. *Biotechnol Prog* 13: 132–143. https://doi.org/10.1021/bp970003o

53. Yang J, Meng X, Hlavacek WS (2010) Rule-based modelling and simulation of biochemical systems with molecular finite automata. *IET Syst Biol* 4: 453–466. https://doi.org/10.1049/iet-syb.2010.0015

54. Yildirim N and Kazanci C (2011) Deterministic and stochastic simulation and analysis of biochemical reaction networks: The lactose operon example, In: Johnson ML and Brand L, *Methods in Enzymology*, New York: Academic Press, 371–395. https://doi.org/10.1016/B978-0-12-381270-4.00012-3

55. Yildirim N and Mackey MC (2003) Feedback regulation in the lactose operon: a mathematical modeling study and comparison with experimental data. *Biophys J* 84: 2841–2851. https://doi.org/10.1016/S0006-3495(03)70013-7

56. Yildirim N, Santillan M, Horike D, et al. (2004) Dynamics and bistability in a reduced model of the lac operon. *Chaos* 14: 279–292. https://doi.org/10.1063/1.1689451

AIMS Press