Building blocks of a biochemical CPU based on DNA transcription logic

Extended Abstract

Mario Lauria[1], Kaustubh Bhalerao[2], Muthu M Pugalanthiran[3], and Bo Yuan[4]

[1] Dept of Computer Science and Engineering, lauria@cis.ohio-state.edu
[2] Dept. of Food, Agricultural and Biological Engineering, bhalerao.1@osu.edu
[3] Dept. of Electrical Engineering, muthu@ee.eng.ohio-state.edu
[4] Departments of Biomedical Informatics, and Pharmacology, yuan.33@osu.edu
The Ohio State University, Columbus, OH

Abstract. In this paper we study the design of transcriptional logic based on quantitative models of cis-regulatory networks. Recent efforts in the area of synthetic biology have shown that logic gates can be implemented using the DNA transcriptional machinery of the cell. We show how to extend these previous results to the design of combinational and sequential circuits. The extension of our method to the design of sequential circuits is particularly attractive because they represent the most general class of circuits. As representative examples here we demonstrate the construction of a memory element and of a 1-bit ALU, two basic building blocks of a transcription-based biochemical CPU.

1 Introduction

In recent years, a growing interest has emerged toward the construction of biochemical circuits based on cellular logic gates [5, 7]. A number of model systems have been constructed to perform oscillations [3], switching [4], and simple logic computations [2]. In this paper we describe an approach to the design of synthetic cis-regulatory networks based on their analogy with logic circuits. Our scheme is based on the fact that the expression of a gene is regulated by the mutual interaction of DNA-binding proteins (transcription factors) which bind to specific subsequences in the upstream region of the gene (promoter sites).

Previous work has shown the feasibility of designing simple gene regulation networks that implement logic gates and simple boolean circuits [2]. Building upon these results, we show how circuits of higher complexity can be implemented according to the basic principles of digital circuit design. In particular, simple blocks can be put together to implement more complex combinational circuits. Then the addition of feedback paths enables the construction of sequential circuits. The extension of our method to the design of sequential circuits is particularly attractive because they represent the most general class of circuits. For example, we demonstrate the construction of two basic building blocks of a CPU, a memory element and a 1-bit Arithmetic-Logic Unit (ALU).

The applications of our results are twofold. First, the development of the systems biology approach has recently sparked a growing interest in studying biological processes in organisms using a systems framework [6]. Several instances of biological control have been demonstrated computationally and experimentally in organisms like E. coli. [3, 6] These instances show that the implementation of control in biological organisms can be conveniently modeled using conventional digital logic schemes. We hope that improving our understanding of a quantitative model that captures the essential "computational" aspects of the transcriptional machinery while abstracting some of its biochemical details will provide a powerful tool for representing and analyzing regulatory networks.

Second, it is conceivable and feasible that transcriptional networks can be manipulated in order to 'program' the biological organism to perform a function that previously did not exist in it. Such organisms could compute complex functions of their inputs (stimuli) such as counting, integration, correlation, etc. Practical bioengineering applications include bacteria engineered for environmental monitoring and drug delivery.

2 Transcription logic

For our transcriptional circuit design we use the model proposed in [2] without modifications. Here we give a brief description of the model; for a more detailed description the reader is referred to

the cited paper. Any gene that is coded into the DNA of prokaryotes is preceded upstream by the so-called promoter region, which represents the binding site for the RNA polymerase (RNAP). For the RNAP to be able to bind and start the gene transcription, one or more DNA-binding proteins called transcription factors (TF) need to bind to the DNA further upstream at specific binding sites. These binding sites are highly specific for each TF, and the binding affinity (i.e. the propensity of the TF to bind) can be tuned over several orders of magnitude by changing the nucleotide sequence of the site.

Depending on the relative placement of binding sites, the interaction among TFs, and between TFs and RNAP, can be positive (i.e. the presence of one increases the probability of the other binding to the DNA) or negative (mutually exclusive binding). The cooperative interaction between TFs and RNAP, and between TFs, can be modeled as a weak, glue-like interaction of measurable intensity.

The process of transcription involves the probabilistic attachment of the transcription factors as well as the RNAP, followed by the polymerase reaction that leads to the formation of the mRNA. Although the gene regulation is typically quantified by the expression levels of the mRNA or the protein, it is represented compactly in terms of a binary expression level, where 1 means 'high' and 0 means 'low' or expressed at a basal level. According to this model, the design of a transcriptional logic gate is reduced to the selection and careful placement of binding sequences in the cis-regulatory region of a gene.

The transcription process is modeled analytically using a thermodynamic approach. The degree of transcription of the gene is expressed as the equilibrium binding probability P of the RNAP to its DNA binding site (promoter region), given the cellular concentrations of all the TFs [2]. We have implemented this model in Mathematica for each of the circuits we have designed, and for each of them we provide a graph of P in terms of the concentrations of the TFs used as inputs.

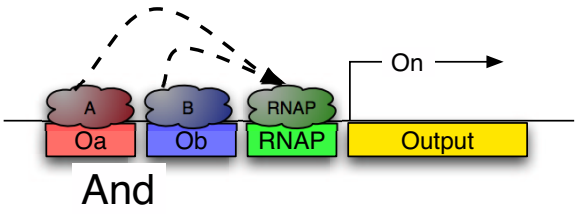


Fig. 1. The transcriptional implementation of an AND gate.

As an example, the implementation of the AND gate as reported in [2] is described. The analogous biological circuit of an AND gate is shown in Figure 1. Dotted arrows indicate the co-operativity factor between transcription factors A, B and RNAP. Weak binding sites (low affinity) are chosen for both TFs, requiring both of them to be present in order to bind to the DNA. When both binding sites Oa and Ob are occupied, conditions for the binding of the RNA polymerase to the DNA become highly favorable.

3 Complex Circuits Based on Transcriptional Logic

Two of the basic building blocks of a CPU are registers and the Arithmetic Logic Unit (ALU). In this abstract we describe the implementation of a 1-bit register, implemented as a D-latch. In the full paper we will also describe the 1-bit ALU.

3.1 D-Latch

Electrical D-Latch Implementation A D-Latch is a simple memory element that preserves the state when the control signal 'ctl' is low and follows the input signal when 'ctl' is high. When ctl is '0', Out = In and the D does not have any impact on the output. When ctl is '1', Out = D and the In does not have any impact on the output. The Block diagram of the D-Latch and its truth table are shown. The Boolean expression for the D-Latch can also be expressed in a sum of products form.

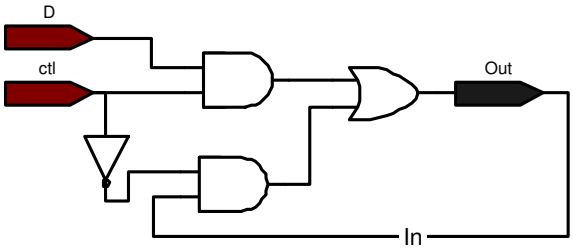


Fig. 2. Electrical circuit of DLatch

$$Out = ctl \cdot D + Inverter(ctl) \cdot In \tag{1}$$

The boolean expression for this circuit shows a simple multiplexer implementation.

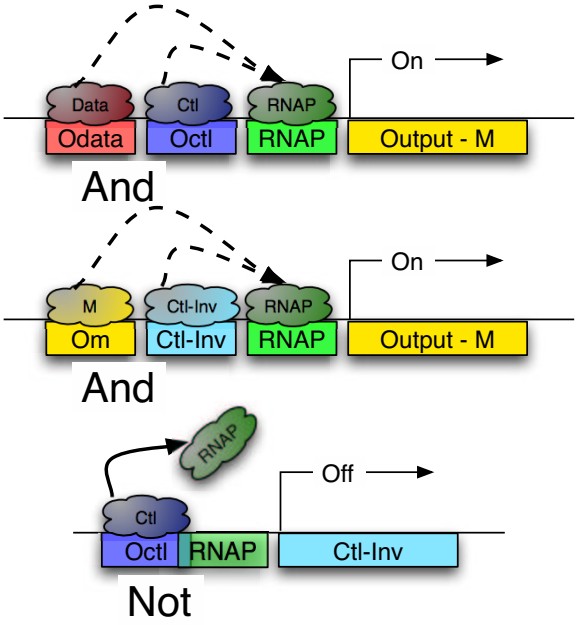


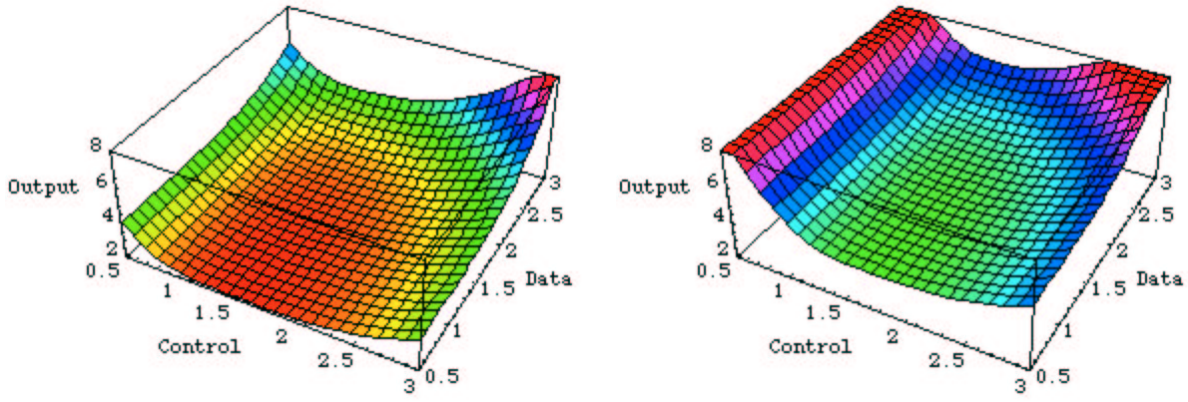
Fig. 3. The DNA equivalent of the D-Latch.

Equivalent Biological circuit The biological D-latch construct consists of two AND gates in parallel forming a distributed OR gate, and one inverter. The 'ctl' protein controls the whole circuit. When it is low(ctl = '0'), output 'M' gets a positive feedback and its concentration is maintained. But when the ctl protein is high(ctl = '1'), the output is dependent on the input data protein 'Data'. If 'Data' protein is high, then output 'M' is produced and if 'Data' protein is low, then output 'M' is not produced.

The three dimensional graphs indicate the performance characteristics of the biological circuit. There are two graphs depending upon the initial state of the circuit(if In is '1' or '0'). The graphs are given with explanation.

3.2 Limitations

Following the example of the adder and the D-Latch, in principle any state machine can be implemented as a transcriptional circuit using the principles of logic circuit design.



(a) (b)

Fig. 4. (a) The D-latch with initial state '0'. When the data is initially 'low', the output is maintained at a low '0'. Only when the control concentration is high the output follows the 'Data' values. (b) The D-latch with initial state '1'. When the data is initially 'high', the positive feedback maintains the value of the output at a high '1'. Only when the control concentration is high the output follows the 'Data' values.

However there are practical considerations that limit the range of circuits that can be realistically implemented. In a transcriptional circuit the data paths are not physically isolated and therefore the input and output signals must use different proteins (TFs). Another concern is the lack of signal regeneration, which represents an obstacle to the construction of deep networks in favor of wide ones. We are currently exploring the design of regenerative elements based on recent results on the analysis and design of positive feedback systems [1]. Finally, the model we used has not been validated yet with *in vitro* experiments.

References

1. D. Angeli, J. Ferrell, James E., and E. D. Sontag. Detection of multistability, bifurcations, and hysteresis in a large class of biological positive-feedback systems. *PNAS*, 101(7):1822–1827, 2004.
2. N. E. Buchler, U. Gerland, and T. Hwa. On schemes of combinatorial transcription logic. *Proc Natl Acad Sci U S A.*, 100(9):5136–5141, April 2003.
3. M. B. Elowitz and S. Liebler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335–338, January 2000.
4. T. S. Garder, C. R. Cantor, and J. J. Collins. Construction of a genetic toggle switch in escherichia coli. *Nature*, 403:339–342, January 2000.
5. T. F. K. Jr. and G. J. Sussman. Cellular gate technology. In *First International Conference on Unconventional Models of Computation (UCM98)*, pages 257–272, Auckland, NZ, January 1998.
6. H. Kitano. Systems biology: A brief overview. *Science*, 295(5560):1662–1664, 2002.
7. R. Weiss and S. Basu. The device physics of cellular logic gates. In *1st Workshop on Non-Silicon Computing (NSC-1)*, Cambridge, Mass, February 2002.