BioSystems 186 (2019) 103983

Contents lists available at ScienceDirect

BioSystems

journal homepage: www.elsevier.com/locate/biosystems

Robust chemical circuits☆

Samuel J. Ellis[a], Titus H. Klinge[b,*], James I. Lathrop[c]

[a] The Molecular Sciences Software Institute, Blacksburg, VA 24060, USA
[b] Department of Mathematics and Computer Science, Drake University, Des Moines, IA 50311, USA
[c] Department of Computer Science, Iowa State University, Ames, IA 50011, USA

ARTICLE INFO

Keywords:
Nanocomputing
Molecular programming
Circuits
Robustness
Chemical reaction networks

ABSTRACT

We introduce a new motif for constructing robust digital logic circuits using input/output chemical reaction networks. These chemical circuits robustly handle adversarial manipulation to their input signals, initial concentrations, rate constants, and measurements. In particular, we show that all Boolean circuits and several sequential circuits enjoy this robustness. Our results complement existing literature in the following three ways: (1) our logic gates read their inputs catalytically which make fanout gates unnecessary; (2) formal requirements and rigorous proofs of satisfaction are provided for each circuit; and (3) robustness of every circuit is closed under modular composition.

1. Introduction

The development of affordable, fast, and reliable electronic logic circuits has broadly impacted society by accelerating many scientific and technological advancements. Similarly, the development of robust biochemical logic circuits could broadly impact society by introducing new methods for drug therapy, bio-diagnostics, and synthetic biology. Unfortunately, the techniques for implementing electronic logic gates cannot be used in biochemical applications due to the differences in dynamics of electrical and chemical systems (Cardelli et al., 2018). Thus, investigations into general methods for implementing biochemical circuits have progressed independently, and the development of robust, fast, and reliable biochemical computing is still in its infancy.

Research into biochemical circuits dates back at least to 1991 (Hjelmfelt et al., 1991), and since then many theoretical motifs for implementing logic circuits have been proposed (Magnasco, 1997; Hinze et al., 2009; Jiang et al., 2013; Ge et al., 2017; Ellis, 2017; Beiki et al., 2018; Arkin and Ross, 1994; Qian and Winfree, 2011; Garg et al., 2018). Chemical reaction networks (CRNs) are currently the mathematical model of choice for biochemical computing and have been studied for over 50 years (Aris, 1965). This is primarily due to recent results showing they are computationally powerful (Cook et al., 2009; Soloveichik et al., 2008; Fages et al., 2017) and can be implemented using DNA molecules (Soloveichik et al., 2010; Cardelli, 2013; Chen et al., 2013; Srinivas et al., 2017; Badelt et al., 2017) using toehold-mediated strand displacement (Yurke et al., 2000; Zhang and Winfree, 2009; Zhang and Seelig, 2011; Lakin et al., 2012b). Furthermore, high-quality DNA is relatively cheap to synthesize (Hughes and Ellington, 2017), all of which makes the chemical reaction network a promising development tool for biochemical applications.

The aim of this paper is to help further the *reliability* of biochemical circuits. In existing literature, reliability has been primarily investigated in two ways: simulation and experimentation. References (Magnasco, 1997; Jiang et al., 2013; Ge et al., 2017; Beiki et al., 2018) use simulation to analyze their circuits in various contexts, and references (Arkin and Ross, 1994; Qian and Winfree, 2011; Garg et al., 2018) include *in vitro* experiments to verify their designs. Although simulations and experiments demonstrate correctness under certain environmental assumptions and initial conditions, they cannot guarantee the *absence* of failure. Formally stating molecular circuit requirements and rigorously proving their satisfaction in all circumstances satisfying certain conditions gives additional confidence in the design as well as insight into when failure is likely to occur. Model checking is a common approach to formally verify the requirements of chemical reaction networks under stochastic semantics (Lakin et al., 2012a; Ellis, 2014). Unfortunately this approach does not scale to large populations of molecules since the computational resources required to verify the system grows exponentially in the number of molecules.

We introduce a new biochemical circuit motif in the input/output chemical reaction network (I/O CRN) model originally introduced by

☆ This work is supported by National Science Foundation grants 1247051 and 1545028. A preliminary version of a portion of this work was presented at the Sixth International Conference on the Theory and Practice of Natural Computing (TPNC 2017, Prague, Czech Republic, December 18–20, 2017).

* Corresponding author.
E-mail addresses: sjellis@vt.edu (S.J. Ellis), titus.klinge@drake.edu (T.H. Klinge), jil@iastate.edu (J.I. Lathrop).

https://doi.org/10.1016/j.biosystems.2019.103983
Received 29 August 2018; Received in revised form 10 June 2019; Accepted 11 June 2019
Available online 14 June 2019
0303-2647/

Klinge, Lathrop, and Lutz (Klinge et al., 2016). An I/O CRN is an abstraction of the traditional CRN model (Feinberg, 1979; Gunawardena, 2003) making it possible for input signals to be provided externally over time. These inputs can only be used catalytically which makes them *read-only*. We assume *deterministic mass action kinetics*, and therefore an I/O CRN is modeled with a system of polynomial ordinary differential equations (ODEs). Thus, the I/O CRN has strong ties to the general purpose analog computer (GPAC) model (Shannon, 1941; Graça et al., 2008), however, the system of ODEs induced by mass action kinetics is more constrained than a GPAC due to the structural properties of reactions. I/O CRNs also offer a natural notion of robustness with respect to adversarial manipulation of the input signal, initial condition, rate constants, and measurement devices. We use this notion to prove that our circuit designs operate correctly even in hostile environments.

Our circuit design uses *dual-rail* encoding of bits in which two species with opposite operational meaning are used to encode each value. Each bit is designed so that the sum of these two species is constant, ensuring that if one has high (value of 1) concentration, the other is low (value of 0). Dual-rail representation is commonly used in biochemical systems since 0s and 1s are encoded by the *presence* of molecules rather than their absence. (Detecting the absence of a species is challenging since reactions are active only if their reactants are *present*. See Doty (2014), Ellis et al. (2014) for more details on the complexity of absence detection and for a proposed method for overcoming it.) To ensure that only one of the dual-species is high at a time, we also include *signal restoration* reactions for each encoded value. These reactions are key to proving that robustness is preserved under composition and causes the dual-species with majority concentration to consume the minority species. (For a thorough analysis of the behavior of these reactions, see Klinge (2016b). Modular composition of CRNs has been investigated in a stochastic setting by Lakin et al. (2016), Shin et al. (2017) to preserve simulation trace requirements. In contrast, our results concern analog systems that evolve deterministically rather than stochastically. Moreover, we prove that composition of chemical circuits not only preserves correctness but also preserves *robustness*.

The main contributions of this work are: (1) we provide natural and rigorous requirements for what it means for I/O CRNs to simulate Boolean circuits; (2) we give an I/O CRN construction of a NAND gate and formally prove it satisfies its requirement even in the presence of worst-case adversarial manipulation to its input, initial state, rate constants, and measurements; (3) we prove that circuits can be modularly composed to robustly implement any Boolean circuit; and (4) we prove that two commonly used sequential circuits for storing memory can be robustly implemented, namely the SR latch and the D latch. Section 2 reviews the I/O CRN model and the notion of robustly satisfying requirements; Section 3 provides an I/O CRN construction of a NAND gate with a formal proof of its robustness; Section 4 contains our main theorem that all Boolean circuits can be robustly implemented by I/O CRNs; Section 5 provides our I/O CRN constructions for the sequential memory components along with proofs of their robustness; and Section 6 closes with a discussion of the strengths and weaknesses of this method of implementing circuits.

2. Preliminaries

In this section, we review the definition of the input/output chemical reaction network (I/O CRN) and our notion of an I/O CRN robustly satisfying a requirement. These were introduced by Klinge, Lathrop, and Lutz in 2016 and will soon appear in a detailed extension of Klinge et al. (2015). The I/O CRN model integrates concepts from control theory into the traditional model of chemical reaction networks under mass action kinetics. In particular, they include the notion of an *input signal* which is a collection of externally provided time-varying concentrations of input species. Such a notion is especially appropriate for biochemical circuits, since their inputs are usually time-varying signals rather than a static initial condition. For an in-depth overview, see Klinge (2016a).

2.1. Input/output chemical reaction networks

We begin by fixing a countably infinite set $\boldsymbol{S} = \{X_0, X_1, X_2 \ldots\}$ of *species* which are abstract molecule types usually denoted with capital Roman letters such as X, Y, and Z. A *reaction* over a finite set $\mathcal{S} \subseteq \boldsymbol{S}$ of species is a triple $\rho = (\mathbf{r}, \mathbf{p}, k) \in \mathbb{N}^{|\mathcal{S}|} \times \mathbb{N}^{|\mathcal{S}|} \times (0, \infty)$ such that $\mathbf{r} \neq \mathbf{p}$. The elements of a reaction $\rho = (\mathbf{r}, \mathbf{p}, k)$ are called the *reactant vector*, *product vector* and *rate constant*, respectively, and the *net effect* of the reaction is the vector $\Delta\rho = \mathbf{p} - \mathbf{r}$. Given a reaction $\rho = (\mathbf{r}, \mathbf{p}, k)$, we use $\mathbf{r}(\rho) = \mathbf{r}$, $\mathbf{p}(\rho) = \mathbf{p}$, and $k(\rho) = k$ for the individual components of ρ.

We frequently use the more intuitive notation of chemistry to improve the readability of reactions. For example, $A + B \xrightarrow{k} 2B + C$ defines the reaction $\rho = (\mathbf{r}, \mathbf{p}, k)$ over the set $\mathcal{S} = \{A, B, C\}$ where $\mathbf{r} = (1, 1, 0)$ and $\mathbf{p} = (0, 2, 1)$. The net effect of the reaction is $\Delta\rho = (-1, 1, 1)$, meaning it consumes one A and produces one B and one C. For convenience, we treat the vectors $\mathbf{r}$, $\mathbf{p}$, and $\Delta\rho$ as functions from the set $\mathcal{S}$ into the natural numbers. Thus, $\mathbf{r}(A) = 1$, $\mathbf{r}(B) = 1$, and $\mathbf{r}(C) = 0$ for the reaction ρ above. We call a species $Y \in \mathcal{S}$ a *reactant* of $\rho = (\mathbf{r}, \mathbf{p}, k)$ if $\mathbf{r}(Y) > 0$, a *product* of ρ if $\mathbf{p}(Y) > 0$, and a *catalyst* of ρ if $\mathbf{r}(Y) > 0$ and $\Delta\rho(Y) = 0$. Note that a catalyst is simply a species that participates in a reaction but is unaffected by it.

An *input/output chemical reaction network* (*I/O CRN*) is a tuple $\mathcal{N} = (\mathcal{U}, \mathcal{R}, \mathcal{S})$ where $\mathcal{U}, \mathcal{S}, \subseteq \boldsymbol{S}$ are finite sets of species that satisfy $\mathcal{U} \cap \mathcal{S} = \emptyset$ and $\mathcal{R}$ is a finite set of reactions over $\mathcal{U} \cup \mathcal{S}$ such that $\Delta\rho(X) = 0$ for each $\rho \in \mathcal{R}$ and $X \in \mathcal{U}$. We call the elements of $\mathcal{S}$ *state species* and the elements of $\mathcal{U}$ *input species*. The key distinguishing feature of an I/O CRN $\mathcal{N}$ is its explicitly defined input species, which can only be used catalytically in its reactions $\mathcal{R}$. Thus, an I/O CRN effectively has *read-only* access its input species and must process any time-varying input in real time.

We now define the semantics of an I/O CRN, borrowing terminology and notation from control theory. Under *deterministic mass action semantics* (also called *mass action kinetics*), a *state* of an I/O CRN $\mathcal{N} = (\mathcal{U}, \mathcal{R}, \mathcal{S})$ is a vector $\boldsymbol{x} \in [0, \infty)^{|\mathcal{S}|}$ that assigns to each $Y \in \mathcal{S}$ a real-valued *concentration* $\boldsymbol{x}(Y)$. Similarly, an *input state* is a vector $\boldsymbol{u} \in [0, \infty)^{|\mathcal{U}|}$, and a *global state* is a vector $(\boldsymbol{x}, \boldsymbol{u}) \in [0, \infty)^{|\mathcal{S} \cup \mathcal{U}|}$.

For a finite set $\mathcal{W} \subseteq \boldsymbol{S}$, we define the *$\mathcal{W}$-signal space* to be the set $C[\mathcal{W}] = C([0, \infty), [0, \infty)^{|\mathcal{W}|})$ where $C(\mathcal{X}, \mathcal{Y})$ is the set of all continuous functions from $\mathcal{X}$ to $\mathcal{Y}$. Thus each element $\boldsymbol{w} \in C[\mathcal{W}]$ is a time-varying signal of concentrations over the species in $\mathcal{W}$. For example, if $Y \in \mathcal{W}$, then $\boldsymbol{w}(t)(Y)$ is the concentration of species Y in $\boldsymbol{w}$ at time t. A *context* of an I/O CRN $\mathcal{N} = (\mathcal{U}, \mathcal{R}, \mathcal{S})$ is a tuple $\boldsymbol{c} = (\boldsymbol{u}, \mathcal{V}, h)$ where $\boldsymbol{u} \in C[\mathcal{U}]$, $\mathcal{V} \subseteq \mathcal{S}$, and h: $[0, \infty)^{|\mathcal{S} \cup \mathcal{U}|} \to [0, \infty)^{|\mathcal{V}|}$. The components of the context $\boldsymbol{c} = (\boldsymbol{u}, \mathcal{V}, h)$ are the *input signal*, the *output species*, and the *measurement function*, respectively. The set of all contexts of an I/O CRN $\mathcal{N}$ is denoted $C_{\mathcal{N}}$. Intuitively, an I/O CRN can be regarded as a chemical machine that when placed in a context $\boldsymbol{c} = (\boldsymbol{u}, \mathcal{V}, h)$, transforms its input signal $\boldsymbol{u} \in C[\mathcal{U}]$ into an observed output signal $\boldsymbol{v} \in C[\mathcal{V}]$. The inclusion of the measurement function h in the definition of a context is to specify which species of the I/O CRN are being observed as well as encapsulate any errors introduced by the measurement equipment. In our main theorems, we assume that the measurement function is perturbed and only approximates the true output concentrations. We also make use of the *zero-error measurement function* h_0 defined by

$$h_0(\boldsymbol{x}, \boldsymbol{u})(Y) = (\boldsymbol{x}, \boldsymbol{u})(Y) \tag{1}$$

for each global state $(\boldsymbol{x}, \boldsymbol{u}) \in [0, \infty)^{|\mathcal{S} \cup \mathcal{U}|}$ and for each state species $Y \in \mathcal{S}$. Note that h_0 is a projection function and corresponds to a *perfect* measurement device.

We now define the mechanism for which the state of an I/O CRN evolves over time within a certain context. According to the law of mass action, given a global state $(\boldsymbol{x}, \boldsymbol{u}) \in [0, \infty)^{|\mathcal{S} \cup \mathcal{U}|}$ and a reaction $\rho \in \mathcal{R}$,

the *rate* of ρ in $(\boldsymbol{x}, \boldsymbol{u})$ is the real-value

$$\text{rate}_{\boldsymbol{x},\boldsymbol{u}}(\rho) = k(\rho) \prod_{Y \in S \cup \mathcal{U}} (\boldsymbol{x}, \boldsymbol{u})(Y)^{\mathbf{r}(\rho)(Y)}. \tag{2}$$

Thus, the rate of a reaction is proportional to each of its reactants and the rate constant. For example, if $\rho = (\mathbf{r}, \mathbf{p}, k)$ is the reaction defined by

$$A + 2B \xrightarrow{k} A + 3C$$

where $\mathcal{U} = \{A\}$ and $S = \{B, C\}$, then its rate in state $(\boldsymbol{x}, \boldsymbol{u}) \in [0, \infty)^{|S \cup \mathcal{U}|}$ is $k\boldsymbol{u}(A)\boldsymbol{x}(B)^2$.

For each species $Y \in S$, the *deterministic mass action function* for Y is

$$F_Y(\boldsymbol{x}, \boldsymbol{u}) = \sum_{\rho \in \mathcal{R}} \Delta\rho(Y) \cdot \text{rate}_{\boldsymbol{x},\boldsymbol{u}}(\rho). \tag{3}$$

Intuitively, the function F_Y specifies the total rate of change imposed on Y in the global state $(\boldsymbol{x}, \boldsymbol{u})$. In the context $\boldsymbol{c} = (\boldsymbol{u}, \mathcal{V}, h)$, the concentrations of all species in S of an I/O CRN evolve according to the system of ordinary differential equations (ODEs) defined by

$$\boldsymbol{x}'(t) = F(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{4}$$

for all $t \in [0, \infty)$ where $F(\boldsymbol{x}, \boldsymbol{u})(Y) = F_Y(\boldsymbol{x}, \boldsymbol{u})$ for each $Y \in S$. (Our occasional use of $\boldsymbol{x}$ and $\boldsymbol{u}$ as single states as well as concentration signals is intentional to reduce obfuscation.)

According to the standard theory of ODEs, if the input $\boldsymbol{u}$ is real analytic, then the system (4) along with an initial state $\boldsymbol{x}_0 \in [0, \infty)^{|S|}$ has a unique solution $\boldsymbol{x}(t)$ satisfying $\boldsymbol{x}(0) = \boldsymbol{x}_0$. For this reason, we assume that all input signals are real analytic.[1] See Krantz and Parks (2002) for a thorough introduction to real analytic functions.

Finally, we define the *output signal* of an I/O CRN $\mathcal{N} = (\mathcal{U}, \mathcal{R}, S)$ with initial state $\boldsymbol{x}_0 \in [0, \infty)^{|S|}$ in context $\boldsymbol{c} = (\boldsymbol{u}, \mathcal{V}, h)$ to be

$$\mathcal{N}_{\boldsymbol{x}_0,\boldsymbol{c}}(t) = h(\boldsymbol{x}(t)), \tag{5}$$

for all $t \in [0, \infty)$ where $\boldsymbol{x}(t)$ is the unique solution to (4) with initial state $\boldsymbol{x}_0$. Intuitively, the output signal of an I/O CRN $\mathcal{N}$ is what is measured by an external machine when $\mathcal{N}$ is placed in a context $\boldsymbol{c}$ and initialized with concentrations $\boldsymbol{x}_0$. This concludes the definition of the semantics of I/O CRNs.

Note that the traditional definition of chemical reaction networks under mass action semantics can be defined as $\mathcal{N} = (\varnothing, S, \mathcal{R})$ with no input species. Without input species, the system of ODEs specified by $\mathcal{N}$ from Eq. (4) simplifies to

$$\boldsymbol{x}(t) = F(x(t))$$

which is an *autonomous* system of *polynomial* ODEs, and only depends on t indirectly through species concentrations. An I/O CRN is neither autonomous nor polynomial in this way since the input signal $\boldsymbol{u}(t)$ can be any real analytic function and does not depend on the concentrations of other species. This also differentiates the model from general purpose analog computers (GPACs) which are equivalent to polynomial systems of ODEs.

We conclude by noting that I/O CRNs offer a natural means of modular design and composition. Given two I/O CRNs $\mathcal{N}_1 = (\mathcal{U}_1, \mathcal{R}_1, S_1)$ and $\mathcal{N}_2 = (\mathcal{U}_2, \mathcal{R}_2, S_2)$, we define the *join of* $\mathcal{N}_1$ and $\mathcal{N}_2$ to be the I/O CRN $\mathcal{N}_1 \sqcup \mathcal{N}_2 = (\mathcal{U}, \mathcal{R}, S)$ where $\mathcal{U} = (\mathcal{U}_1 \cup \mathcal{U}_2) \setminus (S_1 \cup S_2)$, $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$, and $S = S_1 \cup S_2$. If $\mathcal{N}_1$ and $\mathcal{N}_2$ have disjoint sets of state species, we say that $\mathcal{N}_1 \sqcup \mathcal{N}_2$ is *modular*. Our Boolean circuit architecture as well as our SR latch and D latch designs utilize this natural modularity.

[1] All continuous signals produced by natural phenomena are real analytic, including all solutions to systems of polynomial differential equations. Therefore, placing this restriction on our input signals is not only necessary, it is a natural choice.

2.2. Time-dependent I/O CRNs

The rate constant of a reaction depends on a variety of factors, including the temperature and the salinity of the chemical solution as well as the geometry of its reactants. Since rate constants are usually measured experimentally, these rate constants are not arbitrarily precise. Similarly, the temperature of the solution encounters random fluctuations, and therefore the rate constants fluctuate over the course of the computation. Thus, if we hope to use an I/O CRN in practice, it must be robust with respect to fluctuations in its rate constants. Thus, we need a variation of the I/O CRN model that replaces the rate constants of reactions with strictly positive functions of time.

We define a *time-dependent reaction* over the set S to be a tuple $\rho = (\mathbf{r}, \mathbf{p}, \hat{k})$ where $\mathbf{r}, \mathbf{p} \in \mathbb{N}^{|S \cup \mathcal{U}|}$ and $\hat{k}: [0, \infty) \to (0, \infty)$ is a real analytic function. A *time-dependent input/output chemical reaction network* (*I/O tdCRN*) is a tuple $\mathcal{N} = (\mathcal{U}, \hat{\mathcal{R}}, S)$ where $\mathcal{U}, S \subseteq \boldsymbol{S}$ are finite sets of species such that $S \cap \mathcal{U} = \varnothing$ and $\hat{\mathcal{R}}$ is a finite set of time-dependent reactions that only use species in $\mathcal{U}$ as catalysts.

The deterministic mass action semantics of an I/O tdCRN are the same as that of an I/O CRN except that the rate function of (2) changes to

$$\text{rate}_{\boldsymbol{x}(t),\boldsymbol{u}(t)}(\rho) = \hat{k}(\rho)(t) \prod_{Y \in S \cup \mathcal{U}} (\boldsymbol{x}, \boldsymbol{u})(t)(Y)^{\mathbf{r}(\rho)(Y)}, \tag{6}$$

for all time $t \in [0, \infty)$ in order to incorporate the time-dependent reactions. Eqs. (3)–(5) also change using this new rate equation and become

$$F_Y(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \sum_{\rho \in \mathcal{R}} \Delta\rho(Y) \cdot \text{rate}_{\boldsymbol{x}(t),\boldsymbol{u}(t)}(\rho) \tag{7}$$

$$\boldsymbol{x}'(t) = F(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{8}$$

$$\mathcal{N}_{\boldsymbol{x}_0,\boldsymbol{c}}(t) = h(\boldsymbol{x}(t)), \tag{9}$$

respectively.

For an I/O CRN $\mathcal{N} = (\mathcal{U}, \mathcal{R}, S)$ and constant $\delta > 0$, we say that an I/O tdCRN $\hat{\mathcal{N}} = (\mathcal{U}, \hat{\mathcal{R}}, S)$ is *δ-close* to $\mathcal{N}$ if each $\hat{\rho} \in \hat{\mathcal{R}}$ is the time-dependent equivalent of $\rho \in \mathcal{R}$ and satisfies $|k(\rho) - \hat{k}(\hat{\rho})(t)| \leq \delta$ for all $t \in [0, \infty)$. Using this definition, we can require that not only the I/O CRN N behaves as it is designed, but also all I/O tdCRNs that are δ-close to it behave appropriately. This is what we mean by an I/O CRN being robust with respect to its rate constants.

2.3. Robustness

In this paper, we explicitly define the requirements of our I/O CRNs and formally prove their correctness. To this end, we define a *requirement* of an I/O CRN $\mathcal{N} = (\mathcal{U}, \mathcal{R}, S)$ to be an ordered-pair $\Phi = (\alpha, \phi)$ consisting of the two Boolean predicates $\alpha: C_{\mathcal{N}} \to \{\textbf{true}, \textbf{false}\}$ and $\phi: C[\mathcal{U}] \times C[\mathcal{V}] \to \{\textbf{true}, \textbf{false}\}$, called the *context assumption* and the *I/O requirement*, respectively. We say that an I/O CRN $\mathcal{N} = (\mathcal{U}, \mathcal{R}, S)$ *satisfies* the requirement $\Phi = (\alpha, \phi)$, and we write $\mathcal{N} \vDash \Phi$, if there exists an initial state $\boldsymbol{x}_0 \in [0, \infty)^{|S|}$ such that for all $\boldsymbol{c} \in C_{\mathcal{N}}$

$$\alpha(\boldsymbol{c}) \Rightarrow \phi(\boldsymbol{u}, \mathcal{N}_{\boldsymbol{x}_0,\boldsymbol{c}}). \tag{10}$$

Intuitively, the I/O requirement ϕ enforces that there is some desirable relationship between the input signal $\boldsymbol{u}$ and the output signal $\mathcal{N}_{\boldsymbol{x}_0,\boldsymbol{c}}$ of $\mathcal{N}$. The context assumption α is included in case there are certain contexts that are invalid and irrelevant to the requirement.

Since I/O CRNs are an analog model of computation, it will be necessary to define a notion of *approximately* satisfying a requirement. Thus we want to know how close the output signal $\mathcal{N}_{\boldsymbol{x}_0,\boldsymbol{c}}$ is to satisfying ϕ. For a distance measure, we use the *supremum norm* $\|f\| = \sup_{t \in [0,\infty)} |\boldsymbol{w}(t)|$ for all $\boldsymbol{w} \in C[\mathcal{W}]$ where $|\boldsymbol{w}(t)| = \sqrt{\sum_{Y \in \mathcal{W}} \boldsymbol{w}(t)(Y)^2}$ is the Euclidean distance function in $\mathbb{R}^{|\mathcal{W}|}$. Now for $\boldsymbol{w}, \hat{\boldsymbol{w}} \in C[\mathcal{W}]$ and $\epsilon > 0$, we say that $\hat{\boldsymbol{w}}$ is *ϵ-close* to $\boldsymbol{w}$ if $\|\boldsymbol{w} - \hat{\boldsymbol{w}}\| \leq \epsilon$.

An I/O CRN $\mathcal{N} = (\mathcal{U}, \mathcal{R}, \mathcal{S})$ *ϵ-satisfies* a requirement $\Phi = (\alpha, \phi)$, and we write $\mathcal{N} \vDash_{\epsilon} \Phi$, if there exists an initial state $\boldsymbol{x}_0 \in [0, \infty)^{|\mathcal{S}|}$ such that

$$\alpha(\boldsymbol{u}, \mathcal{V}, h) \Rightarrow \exists\, \boldsymbol{v} \in C[\mathcal{W}]\,[\|\mathcal{N}_{\boldsymbol{x}_0,\boldsymbol{c}} - \boldsymbol{v}\| \leq \epsilon \wedge \phi(\boldsymbol{u}, \boldsymbol{v})]. \tag{11}$$

Thus, $\mathcal{N} \vDash_{\epsilon} \Phi$ if the output signal of $\mathcal{N}$ is ϵ-close to some signal $\boldsymbol{v} \in C[\mathcal{V}]$ that satisfies ϕ.

We now define what it means for an I/O CRN $\mathcal{N}$ to robustly satisfy a requirement Φ. Intuitively, we allow an *adversary* to manipulate the input signal, the measurement function, the initial state, and the rate constants of $\mathcal{N}$ and require that it continue to satisfy its requirement Φ during this manipulation. The adversary will only be able to manipulate these parameters in a bounded fashion, which means we need two more definitions. Given a context $\boldsymbol{c} = (\boldsymbol{u}, \mathcal{V}, h)$ and real numbers $\delta_1, \delta_2 > 0$, we say that $\hat{\boldsymbol{c}} = (\hat{\boldsymbol{u}}, \mathcal{V}, \hat{h})$ is (δ_1, δ_2)*-close* to $\boldsymbol{c}$ if $\|\boldsymbol{u} - \hat{\boldsymbol{u}}\| \leq \delta_1$ and $\|h - \hat{h}\| \leq \delta_2$. Given states $\boldsymbol{x}, \hat{\boldsymbol{x}} \in [0, \infty)^{|\mathcal{S}|}$ and $\delta > 0$, we say that $\hat{\boldsymbol{x}}$ is δ*-close* to $\boldsymbol{x}$ if $|\boldsymbol{x} - \hat{\boldsymbol{x}}| \leq \delta$.

Finally, we formally state what it means to robustly satisfy a requirement. Given $\mathcal{N} = (\mathcal{U}, \mathcal{R}, \mathcal{S})$, $\Phi = (\alpha, \phi)$, $\epsilon > 0$, and $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$ such that $\delta_1, \delta_2, \delta_3, \delta_4 > 0$, we say that $\mathcal{N}$ *$\boldsymbol{\delta}$-robustly ϵ-satisfies* Φ, and we write $\mathcal{N} \vDash_{\epsilon}^{\boldsymbol{\delta}} \Phi$, if there exists an initial state $\boldsymbol{x}_0 \in [0, \infty)^{|\mathcal{S}|}$ such that for all contexts $\boldsymbol{c} = (\boldsymbol{u}, \mathcal{V}, h)$ satisfying $\alpha(\boldsymbol{c})$, for each context $\hat{\boldsymbol{c}} = (\hat{\boldsymbol{u}}, \mathcal{V}, \hat{h})$ (δ_1, δ_2)-close to $\boldsymbol{c}$, for each state $\hat{\boldsymbol{x}}_0 \in [0, \infty)^{|\mathcal{S}|}$ δ_3-close to $\boldsymbol{x}_0$, and for each I/O tdCRN $\hat{\mathcal{N}}$ δ_4-close to $\mathcal{N}$, there exists a concentration signal $\boldsymbol{v} \in C[\mathcal{V}]$ that is ϵ-close to the output signal $\hat{\mathcal{N}}_{\hat{\boldsymbol{x}}_0,\hat{\boldsymbol{c}}}$ that satisfies $\phi(\boldsymbol{u}, \boldsymbol{v})$. Thus if $\mathcal{N} \vDash_{\epsilon}^{\boldsymbol{\delta}} \Phi$, then $\mathcal{N}$ is robust to an adversarial modifications to its input signal $\boldsymbol{u}$ by δ_1, its measurement function h by δ_2, its initial state $\boldsymbol{x}$ by δ_3, and its rate constants by δ_4.

We conclude this section with a note on modularly joining I/O CRNs. If $\mathcal{N}_1$ and $\mathcal{N}_2$ are two I/O CRNs satisfying $\mathcal{N}_1 \vDash_{\epsilon_1}^{\delta_1} \Phi_1$ and $\mathcal{N}_2 \vDash_{\epsilon_2}^{\delta_2} \Phi_2$, respectively, and $\mathcal{N} = \mathcal{N}_1 \sqcup \mathcal{N}_2$ is a modular join of $\mathcal{N}_1$ and $\mathcal{N}_2$, then the individual subcomponents of $\mathcal{N}$ still satisfy the requirements Φ_1 and Φ_2. However, if $\mathcal{N}_1$ and $\mathcal{N}_2$ share state species, it is possible for them to interfere with each other, and they may no longer satisfy Φ_1 and Φ_2 after the join. We utilize this modular composition extensively throughout the paper.

3. A robust NAND gate

In this section, we prove that a two-input NAND gate can be robustly implemented by an I/O CRN. First, we formally specify the requirement, then we give our I/O CRN implementation, and finally we prove the construction robustly satisfies the requirement.

Since the inputs and output of the NAND gate are implicit parameters to the requirement, we start by specifying them. Given $X_1, X_2 \in \boldsymbol{S}$, we define the set of input species to be $\mathcal{U} = \{X_1, X_2, \overline{X_1}, \overline{X_2}\} \subseteq \boldsymbol{S}$. The species X_1 and X_2 represent the two inputs of the NAND gate, and $\overline{X_1}$ and $\overline{X_2}$ are their *duals*. A *dual* of a species represents its Boolean complement; thus, if the concentration of X_1 is $b \in \{0, 1\}$, the concentration of $\overline{X_1}$ is $1 - b$. We also use this dual-rail convention for the output, and let $\mathcal{V} = \{Y, \overline{Y}\} \subseteq \boldsymbol{S}$ be the set of output species given $Y \in \boldsymbol{S}$.

Given a positive real number τ, called the *propagation delay*, we define the NAND gate requirement $\Phi_{\text{NAND}}(\tau) = (\alpha, \phi)$ where α is defined by

$$\alpha(\boldsymbol{u}, \mathcal{V}, h) \equiv [\mathcal{V} = \{Y, \overline{Y}\} \text{and } h = h_0], \tag{12}$$

where h_0 from Eq. (1) is the zero-error measurement function. Requiring that $h = h_0$ simply requires it to faithfully measure the output species concentrations. Errors will eventually be introduced into h when we show that $\Phi_{\text{NAND}}(\tau)$ is *robustly* satisfied.

Before we specify the I/O requirement of $\Phi_{\text{NAND}}(\tau)$, we first define some useful notation. Let $\boldsymbol{I}(\tau)$ be the set of all closed intervals at least length τ, defined by

$$\boldsymbol{I}(\tau) = \{I = [t_1, t_2] \subseteq [0, \infty) | t_2 - t_1 \geq \tau\}. \tag{13}$$

Since the I/O requirement ϕ is a predicate that takes parameters $\boldsymbol{u} \in C[\mathcal{U}]$ and $\boldsymbol{v} \in C[\mathcal{V}]$, we use $\boldsymbol{u}$ and $\boldsymbol{v}$ as implicit parameters in the following definitions. Given an interval $I \in \boldsymbol{I}(\tau)$, a species $W \in \mathcal{U} \cup \mathcal{V}$, and a bit $a \in \{0, 1\}$, we define

$$[\![W = a]\!]_I \equiv \begin{cases} (\forall\, t \in I)[\boldsymbol{u}(t)(W) = a = 1 - \boldsymbol{u}(t)(\bar{W})], & \text{if } W \in \mathcal{U} \\ (\forall\, t \in I)[\boldsymbol{v}(t)(W) = a = 1 - \boldsymbol{v}(t)(\bar{W})], & \text{if } W \in \mathcal{V} \end{cases}.$$

Note that $[\![W = a]\!]_I$ simply says that the species W and its dual encode the values a and $1 - a$ for all $t \in I$. To help with our definition of ϕ, we also define the predicates

$$\phi_{11}(I) \equiv [\![X_1 = 1 \wedge X_2 = 1]\!]_I,$$
$$\phi_0(I) \equiv [\![X_1 = 0 \vee X_2 = 0]\!]_I,$$

for all $I \in \boldsymbol{I}(\tau)$. The predicate $\phi_{11}(I)$ says that X_1 and X_2 both encode the value 1 in I and $\phi_0(I)$ says that at least one of X_1 and X_2 must encode 0 in I. Similarly, for $a \in \{0, 1\}$ we define the Boolean predicate

$$\psi_a(I) \equiv [\![Y = a]\!]_{[t_1+\tau, t_2]},$$

for all $I = [t_1, t_2] \in \boldsymbol{I}(\tau)$, which says that Y encodes a for all but the first τ time of the interval I.

We now have sufficient terminology to define the I/O requirement ϕ to be

$$\phi(\boldsymbol{u}, \boldsymbol{v}) \equiv (\forall\, I \in \boldsymbol{I}(\tau))[(\phi_{11}(I) \rightarrow \psi_0(I)) \wedge (\phi_0(I) \rightarrow \psi_1(I))] \tag{14}$$

for all $\boldsymbol{u} \in C[\mathcal{U}]$ and $\boldsymbol{v} \in C[\mathcal{V}]$. Intuitively, ϕ says that if X_1 and X_2 are both 1, then Y must converge to 0 in at most τ time and must remain there as long as both inputs stay 1. Similarly, if either input is 0, then the output must converge to 1 in at most τ time and remain there while the 0 persists. This is visualized in Fig. 1.

We now specify our I/O CRN that robustly simulates a NAND gate.

Construction 1:

Given three species $X_1, X_2, Y \in \boldsymbol{S}$, a vector of strictly positive real numbers $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$, and $\tau > 0$, define the I/O CRN $\text{NAND}_{\boldsymbol{\delta},\tau}(X_1, X_2, Y) = (\mathcal{U}, \mathcal{R}, \mathcal{S})$, where $\mathcal{U} = \{X_1, X_2, \overline{X_1}, \overline{X_2}\}$, , $\mathcal{R}$ consists of the reactions

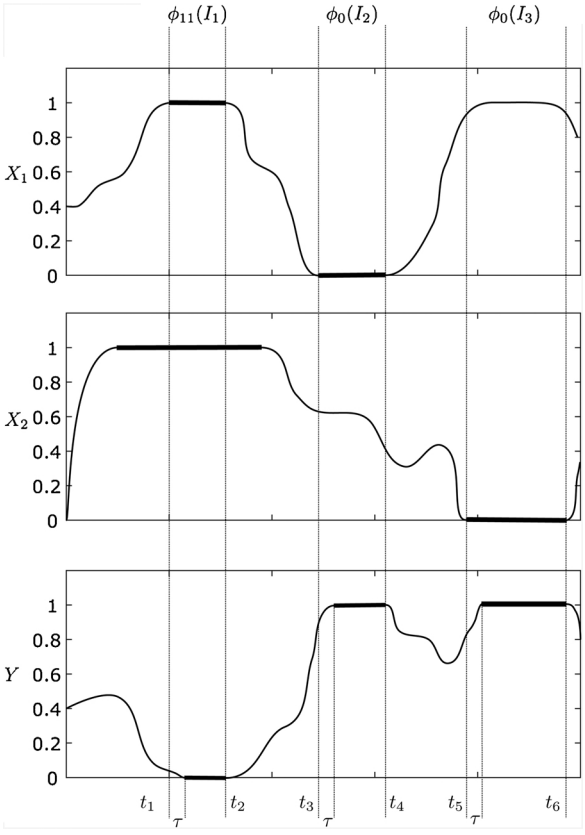

Fig. 1. Visualization of the NAND gate requirement.

$$2Y + \overline{Y} \xrightarrow{3k} 3Y \tag{15}$$

$$2\overline{Y} + Y \xrightarrow{3k} 3\overline{Y}, \tag{16}$$

$$X_1 + X_2 + Y \xrightarrow{k} X_1 + X_2 + \overline{Y} \tag{17}$$

$$\overline{X_1} + \overline{Y} \xrightarrow{k} \overline{X_1} + Y \tag{18}$$

$$\overline{X_2} + \overline{Y} \xrightarrow{k} \overline{X_2} + Y \tag{19}$$

and where $k = 100\delta_4 + \frac{13}{\tau}$.

In the above construction, reaction (17) biases the output toward $\overline{Y}$ when the inputs X_1 and X_2 are both present, reactions (18) and (19) bias the output toward Y in the presence of $\overline{X_1}$ or $\overline{X_2}$ (i.e., in the absence of X_1 or X_2), and reactions (15) and (16) give extra bias to the output species with majority concentration. The latter two reactions are essential for the I/O CRN to produce an output signal that is as clean as its input and was studied extensively in (Klinge, 2016b). The construction also preserves the total concentration of Y and $\overline{Y}$ so that their sum is always constant.

We now state the main theorem of this section.

Theorem 2

If $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4) \in (0, \infty)^4$ and $\tau > 0$ are constants satisfying $\delta_2 + \delta_3 < \delta_1 < \frac{1}{25}$ and $\delta_2 + \delta_3 < \frac{1}{100}$, then $\mathrm{NAND}_{\delta,\tau}(X_1, X_2, Y) \vDash^{\delta}_{\delta_1} \Phi_{\mathrm{NAND}}(\tau)$.

The remainder of this section is devoted to proving this theorem. Since the proof requires examining an arbitrary perturbation of a variety of parameters, we begin the proof by fixing these perturbations.

Assume the hypothesis with $\mathcal{N} = (\mathcal{U}, \mathcal{R}, \mathcal{S}) = \mathrm{NAND}_{\delta,\tau}(X_1, X_2, Y)$. We fix initial state $\boldsymbol{x}_0 \in [0, \infty)^S$ defined by $\boldsymbol{x}_0(Y) = 1$ and $\boldsymbol{x}_0(\overline{Y}) = 0$. (Note that any choice satisfying $\boldsymbol{x}_0(Y) + \boldsymbol{x}_0(\overline{Y}) = 1$ suffices for our argument.) Let $\boldsymbol{c} = (\boldsymbol{u}, \mathcal{V}, h)$ be a context that satisfies the context assumption $\alpha(\boldsymbol{c})$. Let $\hat{\boldsymbol{c}} = (\hat{\boldsymbol{u}}, \mathcal{V}, \hat{h})$ be (δ_1, δ_2)-close to $\boldsymbol{c}$, let $\hat{\boldsymbol{x}}_0$ be δ_3-close to $\boldsymbol{x}_0$, and let $\hat{\mathcal{N}}$ be δ_4-close to $\mathcal{N}$. It now suffices to show that the output function $\hat{\mathcal{N}}_{\hat{\boldsymbol{c}},\hat{\boldsymbol{x}}_0}$ is δ_1-close to a signal $\boldsymbol{v} \in C[\mathcal{V}]$ satisfying $\phi(\boldsymbol{u}, \boldsymbol{v})$ of Φ_{NAND}. Let $\hat{\boldsymbol{x}} \in C[\mathcal{S}]$ as the unique solution generated by $\hat{\mathcal{N}}$ in context $\hat{\boldsymbol{c}}$ on the initial state $\hat{\boldsymbol{x}}_0$. For convenience, we write $y(t)$ and $\bar{y}(t)$ to denote $\hat{\boldsymbol{x}}(t)(Y)$ and $\hat{\boldsymbol{x}}(t)(\overline{Y})$, respectively.

Using the reactions from Construction 1 along with the definition of the deterministic mass action system for an I/O tdCRN from Eqs. (6)–(8), we observe that the ODEs for $y(t)$ and $\bar{y}(t)$ are

$$\frac{\mathrm{dy}}{\mathrm{dt}} = 3\hat{k}_1 y^2 \bar{y} - 3\hat{k}_2 y \bar{y}^2 - \hat{k}_3 x_1 x_2 y + \hat{k}_4 \bar{x}_1 \bar{y} + \hat{k}_5 \bar{x}_2 \bar{y}, \tag{20}$$

$$\frac{d\bar{y}}{\mathrm{dt}} = -\frac{\mathrm{dy}}{\mathrm{dt}}, \tag{21}$$

where $\hat{k}_1$, $\hat{k}_2$, $\hat{k}_3$, $\hat{k}_4$, and $\hat{k}_5$ are all time-varying δ_4-perturbations of the rate constant k and $x_1(t)$, $x_2(t)$, $\bar{x}_1(t)$, and $\bar{x}_2(t)$ are the four components of the δ_1-perturbed input signal $\hat{\boldsymbol{u}}(t)$.

Eq. (21) immediately implies that the total concentration of Y and $\overline{Y}$ is constant, i.e., that $p = y(t) + \bar{y}(t)$ for all $t \in [0, \infty)$ where $p = \hat{\boldsymbol{x}}_0(Y) + \hat{\boldsymbol{x}}_0(\overline{Y})$. It is also useful to note that $|p - 1| < \delta_3$ since $\hat{\boldsymbol{x}}_0$ is a δ_3-perturbation of $\boldsymbol{x}_0$ which satisfies $\boldsymbol{x}_0(Y) + \boldsymbol{x}_0(\overline{Y}) = 1$.

The I/O requirement $\phi(\boldsymbol{u}, \boldsymbol{v})$ is the conjunction of two statements, and we prove each statement holds individually in Lemmas 5 and 6. Before proving these lemmas, we show that the solution $\hat{\boldsymbol{x}}(t)$ is bounded by the solution of much simpler systems of ODEs, and the analyses of these simpler ODEs are given in Lemmas 3 and 4. For convenience, we define the constant $d = \frac{\delta_4}{k}$.

Lemma 3

If $x(t)$ is the solution to the initial value problem defined by $x(0) = 0$ and

$$\frac{\mathrm{dx}}{\mathrm{dt}} = k(-a + b(p - x) - \mathrm{cx}), \tag{22}$$

where $a = \frac{p^3}{18}((3 + d)^{3/2} + 9d)$, $b = (1 - d)(1 - \delta_1)^2$, and $c = 2\delta_1(1 + d)$, then $x\left(\frac{\tau}{2}\right) > \frac{3}{5}$.

Proof

The single variable ODE (22) can be solved by separation of variables and integrating which yields

$$x(t) = \frac{\mathrm{b\,p} - a}{b + c}(1 - e^{-k(b+c)t}).$$

Using the facts that $\delta_1 < \frac{1}{25}$, $d < \frac{1}{100}$, $\delta_3 < \frac{1}{100}$, $|p - 1| < \delta_3$ and $k > \frac{13}{\tau}$, it is easy to verify via substitution that $x\left(\frac{\tau}{2}\right) > \frac{3}{5}$. □

Lemma 4

If $x(t)$ is the solution to the initial value problem defined by $x(0) = \frac{3}{5}$ and

$$\frac{\mathrm{dx}}{\mathrm{dt}} = \mathrm{ax}^2(p - x) - \mathrm{bx}(p - x)^2 - \mathrm{cx}, \tag{23}$$

where $a = 3k(1 - d)$, $b = 3k(1 + d)$, and $c = 2k\delta_1(1 + d)$, then $x(t) > p - \gamma$ for all $t \geq \frac{\tau}{2}$ where $\gamma = \delta_1 - \delta_2 - \delta_3$.

Proof

The ODE (23) has been studied extensively and is sometimes referred to as a *signal restoration algorithm*. According to two theorems proved in Klinge (2016b), if the inequalities

$$c < \frac{p^2 a^2}{4(a + b)} \text{and} \tag{24}$$

$$x(0) > E_1, \tag{25}$$

hold where $E_1 = p\left(\frac{b}{a+b}\right) + A$ such that $A = \frac{p}{2}\left(\frac{a}{a+b}\right)(1 - \sqrt{1 - c^*})$ and $c^* = \frac{4c(a+b)}{p^2 a^2}$, then $x(t)$ exponentially quickly converges to the value $E_2 = p - A$. Using the facts that $d < \frac{1}{100}$, $\delta_1 < \frac{1}{25}$, $\delta_3 < \frac{1}{100}$ and $x(0) = \frac{3}{5}$, it is easy to verify that both of the above inequalities hold.

Corollary 4.5 of Klinge (2016b) shows that under these conditions $x(t)$ will converge to the quantity $p - \gamma$ and remain above it indefinitely in at most time

$$T = \frac{a + b}{\mathrm{abp}^2(1 - c^*)} \log u,$$

where $u = \frac{(p - \gamma - E_1)(E_2 - \frac{3}{5})}{\left(\frac{3}{5} - E_1\right)(E_2 - p + \gamma)}$. Using the bounds of d, δ_1, and δ_3 and the fact that $k > \frac{13}{\tau}$, it is easy to verify that $T \leq \frac{\tau}{2}$. Thus, $x(t) > p - \gamma$ for $t \geq \frac{\tau}{2}$. □

Lemma 5

If $I \in \boldsymbol{I}(\tau)$ such that $\phi_{11}(I)$ holds, then $\psi_0(I)$ holds.

Proof

Assume the hypothesis for $I = [t_1, t_2] \in \boldsymbol{I}(\tau)$. To show that $\psi_0(I)$ holds, we need to show that $1 - \delta_2 < \bar{y}(t) < 1 + \delta_2$ and $y(t) < \delta_2$ holds for all $t \in [t_1 + \tau, t_2]$. Since $y(t) + \bar{y}(t) = p$, it suffices to show that $\bar{y}(t) > p - \gamma$ where $\gamma = \delta_1 - \delta_2 - \delta_3$ for all $t \in [t_1 + \tau, t_2]$. We will show this by bounding the ODE of $\bar{Y}$ from Eq. (21).

Since the perturbed rate constants are within δ_4 of k, we know that

$$\frac{d\bar{y}}{\mathrm{dt}} \geq 3(k - \delta_4)\bar{y}^2 y - 3(k + \delta_4)\bar{y} y^2 + (k - \delta_4)\hat{k}_3 x_1 x_2 y - (k + \delta_4)\hat{k}_4 \bar{x}_1 \bar{y} - (k + \delta_4)\bar{x}_2 \bar{y}.$$

Thus if we let $d = \frac{\delta_4}{k}$, we can rewrite this equation as

$$\frac{d\bar{y}}{\mathrm{dt}} \geq k[3(1 - d)\bar{y}^2 y - 3(1 + d)\bar{y} y^2 + (1 - d)x_1 x_2 y - (1 + d)(\bar{x}_1 + \bar{x}_2)\bar{y}]. \tag{26}$$

It is also not difficult to show that the expression $3(1 - d)\bar{y}^2 y - 3(1 + d)\bar{y} y^2$ is minimized by letting $\bar{y} = \frac{p}{6}(d + 3 - \sqrt{d^2 + 3})$. By substituting this into the expression, we obtain

$$\begin{aligned} 3(1 - d)\bar{y}^2 y - 3(1 + d)\bar{y} y^2 &\geq -\frac{p^3}{18}(3\sqrt{d^2 + 3} + d(d(\sqrt{d^2 + 3} - d) + 9)) \\ &\geq -\frac{p^3}{18}((3 + d)^{3/2} + 9d). \end{aligned}$$

After substituting this into (26) we obtain the bound

$$\frac{d\bar{y}}{dt} \geq k[-\frac{p^3}{18}((3+d)^{3/2}+9d)+(1-d)x_1x_2y-(1+d)(\bar{x}_1+\bar{x}_2)\bar{y}].$$

Since $\phi_{11}(I)$ holds, we know that within the interval I that x_1, x_2, $\bar{x}_1$, and $\bar{x}_2$ are encoding 1, 1, 0, and 0, respectively. However, these are only δ_1-approximating these because of the input perturbation. Thus, for all $t \in I$ we have

$$\frac{d\bar{y}}{dt} \geq k[-a+b(p-\bar{y})-c\bar{y}],$$

where $a = \frac{p^3}{18}((3+d)^{3/2}+9d)$, $b = (1-d)(1-\delta_1)^2$, and $c = 2\delta_1(1+d)$. By Lemma 3, we know $\bar{y}(t_1+\frac{\tau}{2}) \geq \frac{3}{5}$.

To bound the behavior of $\bar{Y}$ after time $t_1+\frac{\tau}{2}$, we take another look at (26) and see that

$$\begin{aligned}\frac{d\bar{y}}{dt} &\geq k[3(1-d)\bar{y}^2y-3(1+d)\bar{y}y^2-2\delta_1(1+d)\bar{y}]\\ &\geq a\bar{y}^2(p-\bar{y})-b\bar{y}(p-\bar{y})^2-c\bar{y},\end{aligned}$$

where $a = 3k(1-d)$, $b = 3k(1+d)$, and $c = 2k\delta_1(1+d)$. By Lemma 4, we see that $\bar{y}(t) > p-\gamma$ for all $t \in [t_1+\tau, t_2]$ which also means that $y(t) < \gamma$ during that interval since $y(t)+\bar{y}(t) = p$.

Finally, since $p > 1-\delta_3$, $\gamma = \delta_1-\delta_2-\delta_3$, and the measurement function can only introduce δ_2 amount of error, $\hat{N}_{\hat{x}_0,\hat{\epsilon}}(t)(\overline{Y}) > 1-\delta_1$ and $\hat{N}_{\hat{x}_0,\hat{\epsilon}}(t)(Y) < \delta_1$. Therefore $\hat{N}_{\hat{x}_0,\hat{\epsilon}}$ is δ_1-close to encoding an output of $Y = 0$ and $\overline{Y} = 1$ in the interval $[t_1+\tau, t_2]$. □

Lemma 6

If $I \in \boldsymbol{I}(\tau)$ such that $\phi_0(I)$ holds, then $\psi_1(I)$ holds.

Proof

During an interval $I = [t_1, t_2]$ satisfying $\phi_0(I)$, it is easy to show by a similar argument to Lemma 5 that the inequalities

$$\frac{dy}{dt} \geq k[-\frac{p^3}{18}((3+d)^{3/2}+9d)+(1-d)(1-\delta_1)(p-y)-2(1+d)\delta_1\bar{y}]$$

and

$$\frac{dy}{dt} \geq k[3(1-d)y^2(p-y)-3(1+d)y(p-y)^2-2\delta_1(1+d)y]$$

hold for all $t \in I$. Thus by Lemmas 3 and 4, we see that $y(t) > p-\gamma$ and $\bar{y}(t) < \gamma$ for all $t \in [t_1+\tau, t_2]$, and thus $\phi_1(I)$ holds. □

4. Robust Boolean circuits

In this section, we state and prove our main theorem, namely, that every Boolean circuit can be implemented with an I/O CRN. For each Boolean circuit, we define its requirement, give an I/O CRN construction for it, and prove the construction robustly satisfies its corresponding requirement.

Given positive integers n, $m > 0$, we define an *n-input m-output Boolean circuit* $C_{n,m}$ to be a directed acyclic graph where each node is a two-input one-output NAND gate. The circuit $C_{n,m}$ has n incoming edges called *inputs* and m outgoing edges called *outputs*. The *depth* of a circuit $C_{n,m}$ is the longest path from an input to an output. Each circuit $C_{n,m}$ can be regarded as a function $C_{n,m}: \{0,1\}^n \to \{0,1\}^m$ defined in the obvious way by computing the values of the outputs by propagating the input values through each of the NAND gates of the circuit. Since NAND gates are universal for Boolean circuits, this definition includes all possible functions for this class. Furthermore, our dual-rail scheme gives access to the negation of each signal without any additional gates. This substantially reduces the size of many circuits.

For a circuit $C_{n,m}$, we define the set of input species to be

$$\mathcal{U} = \{X_i, \overline{X_i} | 0 \leq i < n\} \subseteq \boldsymbol{S},$$

and define the requirement $\Phi(C_{n,m}, \tau) = (\phi, \alpha)$ where α is defined by

$$\alpha(\boldsymbol{u}, \mathcal{V}, h) \equiv [\mathcal{V} = \{Y_i, \overline{Y_i} | 0 \leq i < m\} \text{ and } h = h_0]. \quad (27)$$

To state the I/O requirement ϕ, we need a bit more terminology. For a string $w \in \{0,1\}^n$ and input $\boldsymbol{u} \in C[\mathcal{U}]$, we use the notation $\boldsymbol{u}(t) = w$ to denote that $\boldsymbol{u}(t)(X_i) = w[i]$ and $\boldsymbol{u}(t)(\overline{X_i}) = 1-w[i]$ for each $0 \leq i < n$. We also define the predicates

$$\phi_w(\mathrm{I}) \equiv \ (\forall\, \mathrm{t} \in \mathrm{I})[\boldsymbol{u}(\mathrm{t}) = \mathrm{w}],$$
$$\psi_w(\mathrm{I}) \equiv \ (\forall\, \mathrm{t} \in [t_1+\tau, t_2])[\boldsymbol{v}(\mathrm{t}) = \mathrm{w}],$$

for all $I = [t_1, t_2] \in \boldsymbol{I}(\tau)$. The I/O requirement ϕ can then be defined by

$$\phi(\boldsymbol{u}, \boldsymbol{v}) \equiv (\forall\, I \in \boldsymbol{I}(\tau))(\forall\, w \in \{0,1\}^n)[\phi_w(I) \to \psi_{C_{n,m}(w)}(I)]. \quad (28)$$

Thus, $\Phi(C_{n,m}, \tau)$ simply requires that an I/O CRN generates the output $C_{n,m}(w)$ within τ time whenever the inputs encode $w \in \{0,1\}^n$. We now give the I/O CRN construction for an arbitrary Boolean circuit.

Construction 7

Given a Boolean circuit $C_{n,m}$ with G gates and depth d along with constants $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$, and $\tau > 0$, define the CRN $\mathcal{N}(C_{n,m}, \boldsymbol{\delta}, \tau)$ by joining G copies of the I/O CRN $\mathrm{NAND}_{\delta,\tau/d}$ from Construction 1 according to the circuit $C_{n,m}$.

As an example, consider a two-input one-output exclusive or (XOR) circuit. Since negations are free in our motif, the XOR circuit can be constructed using three NAND gates, depicted in Fig. 2.

According to Construction 7, the I/O CRN defined by this circuit is

$$\mathcal{N} = \mathcal{N}(C_{n,m}, \delta, \tau) = \mathcal{N}_1 \sqcup \mathcal{N}_2 \sqcup \mathcal{N}_3,$$

where $\mathcal{N}_1 = \mathrm{NAND}_{\delta,\tau/2}(\overline{X_1}, X_2, Z_1)$, $\mathcal{N}_2 = \mathrm{NAND}_{\delta,\tau/2}(X_1, \overline{X_2}, Z_2)$, and $\mathcal{N}_3 = \mathrm{NAND}_{\delta,\tau/2}(Z_1, Z_2, Y)$. For convenience, we assume that the dual of $\overline{X}$ is X so that negations are handled intuitively. The unlabeled intermediate wires correspond to the state species Z_1 and Z_2 of $\mathcal{N}$ and are neither inputs nor outputs of the XOR circuit. The I/O CRN $\mathcal{N}$ is *modular* since $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_3$ do not share any state species. In fact, every I/O CRN produced by Construction 7 is a modular join of NAND gates since Boolean circuits are acyclic. We now state the main theorem of the paper.

Theorem 8

If $C_{n,m}$ is a Boolean circuit, the constants $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4) \in (0, \infty)^4$ and $\tau > 0$ satisfy $\delta_2+\delta_3 < \delta_1 < \frac{1}{25}$, $\delta_2+\delta_3 < \frac{1}{100}$, and $\mathcal{N} = \mathcal{N}(C_{n,m}, \delta, \tau)$ is constructed according to Construction 7, then $\mathcal{N} \vDash^{\delta}_{\delta_1} \Phi(C_{n,m}, \tau)$.

Proof

This theorem immediately follows from the fact that $\mathcal{N}$ consists of a modular family of NAND gates and by Theorem 2 each individual

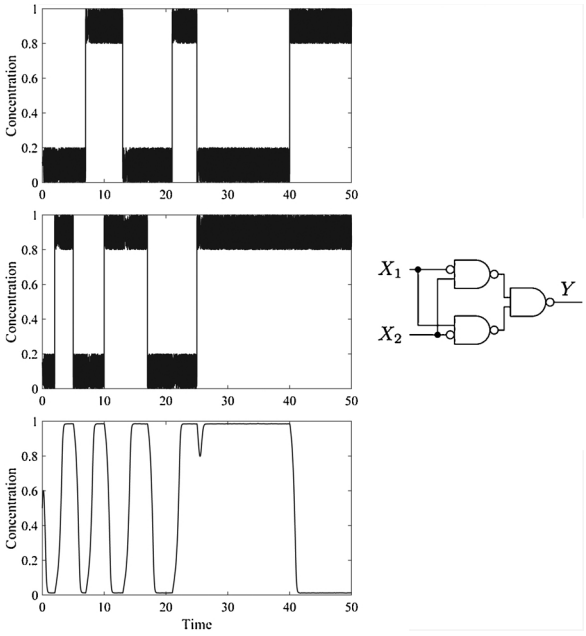

Fig. 2. XOR circuit with random noise.

NAND gate is robust. Intuitively, each of these NAND gates uses their input signals catalytically which makes them truly independent, making fanout gates unnecessary. Thus, we know that each NAND gate produces an output signal that is δ_1-close to its appropriate binary value within $\frac{\tau}{d}$ time. Since d is the depth of the circuit, the total propagation delay for the circuit is at most τ. □

To demonstrate the robustness of these circuits, Fig. 2 also visualizes the output of the XOR circuit on a noisy input signal.

5. Robust memory components

Memory is essential to compute most algorithms, so limiting ourselves only to Boolean circuits is too restricting. The basic memory components of modern circuits are latches and flip flops, but these circuits are *sequential* and depend on cyclic feedback to store data. As a result, the techniques from the previous section do not apply, since joining our NAND gates together in a cyclic environment may cause them to send and receive signals that are not binary. This can cause failure since the behavior of our NAND gate is undefined on non-binary inputs.

In this section, we show that I/O CRNs are capable of robustly simulating two common memory components. In Section 5.1, we show that an SR latch can be robustly simulated by two NAND gates, and in Section 5.2, we introduce a new I/O CRN design that robustly simulates a D latch. A D latch is traditionally implemented using two SR latches; however, our I/O CRN construction uses fewer reactions than a single NAND gate.

5.1. SR latch

The set-reset latch (SR latch) is a simple and commonly used memory element in digital circuits. Composed of two NAND gates, the latch operates with two inputs, usually named $\overline{S}$ and $\overline{R}$, and has three stable states. First, if $\overline{S}$ is 0 and $\overline{R}$ is 1, then the output Q will be 1, i.e., Q is *set*. Similarly, if $\overline{R}$ is 0 and $\overline{S}$ is 1, then the output Q is 0, i.e., Q is *reset*. If both $\overline{S}$ and $\overline{R}$ are 1, the output Q maintains its previous value, i.e., Q is *held*. A schematic diagram of the SR latch is shown in Fig. 3.

To show that this SR latch is robust, we begin by specifying its requirement. We first define the set of input species, set of output species, and some useful predicates. Given $\overline{S}, \overline{R} \in \boldsymbol{S}$, we define the set of input species to be $\mathcal{U} = \{S, \overline{S}, R, \overline{R}\} \subseteq \boldsymbol{S}$, and given $Q_1, \overline{Q}_2 \in \boldsymbol{S}$, we let the set of output species be $V = \{Q_1, \overline{Q}_1, Q_2, \overline{Q}_2\} \subseteq \boldsymbol{S}$. Given $\tau > 0$, we also define the predicates

$$\phi_{\text{set}}(I) \equiv [\![\overline{S} = 0]\!]_{[t_1, t_1+\tau]} \wedge [\![\overline{R} = 1]\!]_I \tag{29}$$

$$\phi_{\text{reset}}(I) \equiv [\![\overline{R} = 0]\!]_{[t_1, t_1+\tau]} \wedge [\![\overline{S} = 1]\!]_I \tag{30}$$

$$\psi_a(I) \equiv [\![Q_1 = a]\!]_{[t_1+\tau, t_2]} \wedge [\![Q_2 = a]\!]_{[t_1+\tau, t_2]}, \tag{31}$$

for all intervals $I = [t_1, t_2] \in \boldsymbol{I}(\tau)$. Note that ϕ_{set} and ϕ_{reset} only require that $\overline{S} = 0$ and $\overline{R} = 0$ for the first τ time of I, but they require $\overline{R} = 1$ and $\overline{S} = 1$ for the entire interval I, respectively. This allows inputs to transition between the set/reset state to the hold state while satisfying $\phi_{\text{set}}/\phi_{\text{reset}}$.

Given a $\tau > 0$, we then define the SR latch requirement to be $\Phi_{\text{SR}}(\tau) = (\alpha, \phi)$ where the context assumption α is defined by

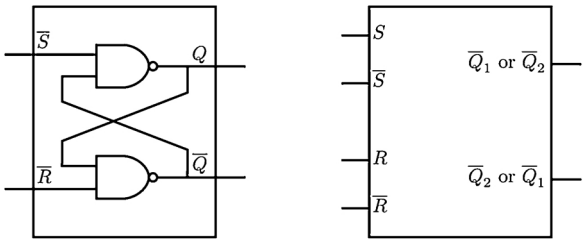

Fig. 3. SR latch implemented with two NAND gates, and SR block diagram with labeled species inputs and outputs.

$$\alpha(\boldsymbol{u}, \mathcal{V}, h) \equiv [\mathcal{V} = \{Q_1, \overline{Q}_1, Q_2, \overline{Q}_2\} \text{ and } h = h_0], \tag{32}$$

and the I/O requirement ϕ is defined by

$$\phi(\boldsymbol{u}, \boldsymbol{v}) \equiv (\forall I \in \boldsymbol{I}(\tau))[(\phi_{\text{set}}(I) \to \psi_1(I)) \wedge (\phi_{\text{reset}}(I) \to \psi_0(I))]. \tag{33}$$

Intuitively, the requirement Φ_{SR} requires that whenever $\overline{S} = 0$ and $\overline{R} = 1$ for at least τ time, then $Q = 1$ within that time and remains there until $\overline{R} \neq 1$. It also requires that if $\overline{S} = 1$ and $\overline{R} = 0$ for at least τ time, then $Q = 0$ until $\overline{S}$ is no longer 1. A visualization of the input/output relationship is included in the timing diagram of Fig. 4.

We now state the construction of the SR latch.

Construction 9

Given four species $\overline{S}$, $\overline{R}$, Q_1, $\overline{Q}_2$, a vector of strictly positive real numbers $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$, and $\tau > 0$, define the CRN

$$\text{SR}_{\delta,\tau}(\overline{S}, \overline{R}, Q_1, \overline{Q}_2) = \mathcal{N}_1 \sqcup \mathcal{N}_2,$$

where $\mathcal{N}_1 = \text{NAND}_{\delta,\tau/2}(\overline{S}, \overline{Q}_2, Q_1)$ and $\mathcal{N}_2 = \text{NAND}_{\delta,\tau/2}(\overline{R}, Q_1, \overline{Q}_2)$.

We now prove that our construction robustly satisfies Φ_{SR}. Our proof shows that the requirements of the two subcomponents suffice to prove the high-level requirement of the SR latch.

Theorem 10

If $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4) \in (0, \infty)^4$ *and* $\tau > 0$ *are constants satisfying* $\delta_2 + \delta_3 < \delta_1 < \frac{1}{25}$ *and* $\delta_2 + \delta_3 < \frac{1}{100}$, *then* $\text{SR}_{\delta,\tau}(\bar{S}, \bar{R}, Q_1, \bar{Q}_2) \vDash^{\delta}_{\delta_1} \Phi_{\text{SR}}(\tau)$.

Proof

Assume the hypothesis, let $\mathcal{N} = \text{SR}_{\delta,\tau}(\overline{S}, \overline{R}, Q_1, \overline{Q}_2)$, and let $\mathcal{N}_1 = \text{NAND}_{\delta,\tau/2}(\overline{S}, \overline{Q}_2, Q_1)$ and $\mathcal{N}_2 = \text{NAND}_{\delta,\tau/2}(\overline{R}, Q_1, \overline{Q}_2)$ be the I/O CRNs used to construct $\mathcal{N}$ from Construction 9. By Theorem 2, we know that

$$\mathcal{N}_1 \vDash^{\delta}_{\delta_1} \Phi_{\text{NAND}}\left(\frac{\tau}{2}\right) \quad \text{and} \tag{34}$$

$$\mathcal{N}_2 \vDash^{\delta}_{\delta_1} \Phi_{\text{NAND}}\left(\frac{\tau}{2}\right) \tag{35}$$

hold. We complete the proof by showing that these imply that $\mathcal{N} \vDash^{\delta}_{\delta_1} \Phi_{\text{SR}}(\tau)$. Note that Φ_{SR} can be easily split up into two parts. We first show that $\phi_{\text{set}}(I) \to \psi_1(I)$ holds, and then show that $\phi_{\text{reset}}(I) \to \psi_0(I)$ holds.

Let $I = [t_1, t_2] \in \boldsymbol{I}(\tau)$ be an interval such that $\phi_{\text{set}}(I)$ holds. Since $[\![\bar{S} = 0]\!]$ holds for all $t \in [t_1, t_1 + \tau]$, (34) tells us that $[\![Q_1 = 1]\!]$ for all $t \in [t_1 + \frac{\tau}{2}, t_1 + \tau]$. Since $[\![\overline{R} = 1]\!]$ and $[\![Q_1 = 1]\!]$ for all $t \in [t_1 + \frac{\tau}{2}, t_1 + \tau]$, (35) tells us that $[\![\overline{Q}_2 = 0]\!]$ starting at time $t_1 + \tau$. As a result, the output of $[\![Q_1 = 1]\!]$ and $[\![\overline{Q}_2 = 0]\!]$ is stable since the output of $\mathcal{N}_1$ will be held constant at 1 while one of its inputs is 0 and $\mathcal{N}_2$ will continue to output 0 while both its inputs are 1 which will be true until time t_2. Thus $\phi_1(I)$ holds for all $t \in [t_1 + \tau, t_2]$.

It remains to be shown that for all $I \in \boldsymbol{I}(\tau)$, $\phi_{\text{reset}}(I) \to \psi_0(I)$ holds. Let $I = [t_1, t_2] \in \boldsymbol{I}(\tau)$ be an interval such that $\phi_{\text{reset}}(I)$ holds. Since $[\![\overline{R} = 0]\!]$ holds for all $t \in [t_1, t_1 + \tau]$, (35) tells us that $[\![\overline{Q}_2 = 1]\!]$ for all $t \in [t_1 + \frac{\tau}{2}, t_1 + \tau]$. Since $[\![\bar{S} = 1]\!]$ and $[\![\bar{Q}_2 = 1]\!]$ for all $t \in [t_1 + \frac{\tau}{2}, t_1 + \tau]$, (34) tells us that $[\![Q_1 = 0]\!]$ starting at time $t_1 + \tau$. As a result, the output of $[\![Q_1 = 0]\!]$] and $[\![\overline{Q}_2 = 1]\!]$ is stable since the output of $\mathcal{N}_2$ will be held constant at 1 while one of its inputs is 0 and $\mathcal{N}_1$ will continue to output 0 while both its inputs are 1 which will be true until time t_2. Thus $\phi_0(I)$ holds for all $t \in [t_1 + \tau, t_2]$. □

Simulations show that the SR latch works even better than the theorem predicts. Fig. 4 shows its output with minor random noise and Fig. 5 demonstrates how it handles significant random and sinusoidal noise.

5.2. D latch

Another commonly used memory element is the D latch. Instead of using the traditional D latch design using four NAND gates, we provide a simpler construction using only four reactions. The design is modeled closely after our NAND gate and uses the signal restoration algorithm of Klinge (2016b) to maintain the signals. Before we give the construction, we first formally specify the requirement for a D latch.

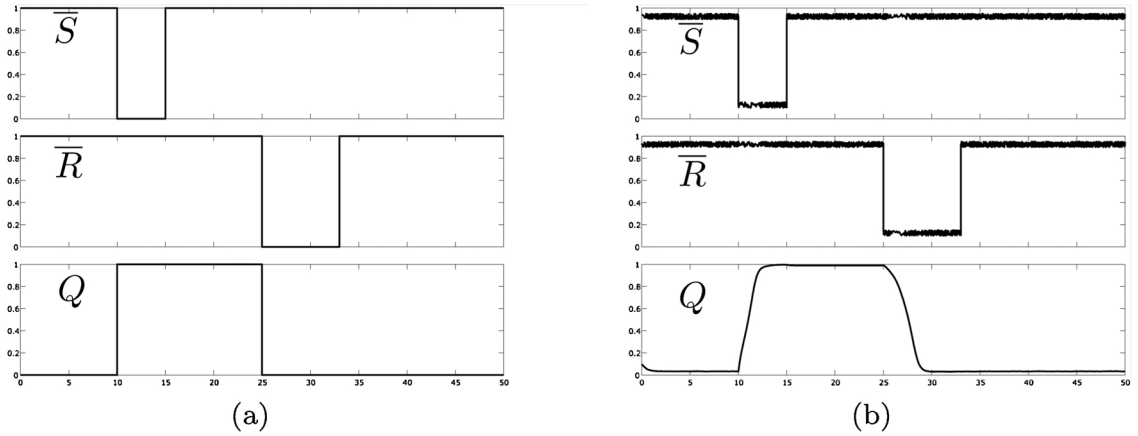

Fig. 4. Ideal SR latch timing diagram along with an I/O CRN simulation of our SR latch with random noise.

Given species $D, E, Q \in \boldsymbol{S}$ and $\tau > 0$, define the set of input species be $\mathcal{U} = \{D, \overline{D}, E, \overline{E}\} \subseteq \boldsymbol{S}$, let $V = \{Q, \overline{Q}\} \subseteq \boldsymbol{S}$ be the set of output species, and for $a \in \{0, 1\}$ let ϕ_a and ψ_a be the predicates

$$\phi_a(I) \equiv [\![D = a \wedge E = 1]\!]_{[t_1, t_1+\tau]} \wedge [\![D = a \vee E = 0]\!]_{[t_1+\tau, t_2]} \tag{36}$$

$$\psi_a(I) \equiv [\![Q = a]\!]_{[t_1+\tau, t_2]} \tag{37}$$

for all $I = [t_1, t_2] \in \boldsymbol{I}(\tau)$. Then let $\Phi_{\mathrm{DL}}(\tau) = (\alpha, \phi)$ be the requirement where the context assumption α is defined by

$$\alpha(\boldsymbol{u}, \mathcal{V}, h) \equiv [\mathcal{V} = \{Q, \overline{Q}\} \text{ and } h = h_0], \tag{38}$$

and the I/O requirement ϕ is defined by

$$\phi(\boldsymbol{u}, \boldsymbol{v}) \equiv (\forall I \in \boldsymbol{I}(\tau))[(\phi_0(I) \Rightarrow \psi_0(I)) \wedge (\phi_1(I) \Rightarrow \psi_1(I))]. \tag{39}$$

Intuitively, the requirement Φ_{DL} says that whenever a set event occurs, i.e., when $D = a$ and $E = 1$ for at least time τ, then within τ time Q converges to a and remains there as long as either $D = a$ or $E = 0$. This is visualized in the timing diagram of Fig. 6.

We now give the I/O CRN construction that satisfies the above requirement.

Construction 11

Given three species D, E, Q, a vector of strictly positive real numbers $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)$, and $\tau > 0$, define the CRN

$$\mathrm{DL}_{\delta,\tau}(D, E, Q) = (\mathcal{U}, \mathcal{R}, \mathcal{S}),$$

where $\mathcal{U} = \{D, \overline{D}, E, \overline{E}\}$, $\mathcal{S} = \{Q, \overline{Q}\}$, and $\mathcal{R}$ consists of the four reactions

$$D + E + \overline{Q} \xrightarrow{k} D + E + Q \tag{40}$$

$$\overline{D} + E + Q \xrightarrow{k} \overline{D} + E + \overline{Q} \tag{41}$$

$$2Q + \overline{Q} \xrightarrow{3k} 3Q \tag{42}$$

$$2\overline{Q} + Q \xrightarrow{3k} 3\overline{Q} \tag{43}$$

where $k = 100\delta_4 + \frac{13}{\tau}$.

Below is the final theorem of this paper showing that the above construction robustly satisfies its requirement.

Theorem 12

If $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4) \in (0, \infty)^4$ *and* $\tau > 0$ *are constants satisfying* $\delta_2 + \delta_3 < \delta_1 < \frac{1}{25}$ *and* $\delta_2 + \delta_3 < \frac{1}{100}$, *then* $\mathrm{DL}_{\delta,\tau}(\overline{S}, \overline{R}, Q_1, \overline{Q}_2) \vDash^{\delta}_{\delta_1} \Phi_{\mathrm{DL}}(\tau)$.

Proof. Assume the hypothesis and let $\mathcal{N} = (\mathcal{U}, \mathcal{R}, \mathcal{S}) = \mathrm{DL}_{\delta,\tau}(D, E, Q)$. We fix initial state $\boldsymbol{x}_0 \in [0, \infty)^S$ defined by $\boldsymbol{x}_0(Q) = 1$ and $\boldsymbol{x}_0(\overline{Q}) = 0$. Let $\boldsymbol{c} = (\boldsymbol{u}, \mathcal{V}, h)$ be a context that satisfies the context assumption $\alpha(\boldsymbol{c})$. Let $\hat{\boldsymbol{c}} = (\hat{\boldsymbol{u}}, \mathcal{V}, \hat{h})$ be (δ_1, δ_2)-close to $\boldsymbol{c}$, let $\hat{\boldsymbol{x}}_0$ be δ_3-close to $\boldsymbol{x}_0$, and let $\hat{\mathcal{N}}$ be δ_4-close to $\mathcal{N}$. We fix $\hat{\boldsymbol{x}} \in C[\mathcal{S}]$ as the unique solution generated by $\hat{\mathcal{N}}$ in context $\hat{\boldsymbol{c}}$ on the initial state $\hat{\boldsymbol{x}}_0$, and for convenience, we write $q(t)$ and $\bar{q}(t)$ to denote $\hat{\boldsymbol{x}}(t)(Q)$ and $\hat{\boldsymbol{x}}(t)(\overline{Q})$, respectively. Now let $p = \hat{\boldsymbol{x}}_0(Q) + \hat{\boldsymbol{x}}_0(\overline{Q})$. Since $\frac{\mathrm{dq}}{\mathrm{dt}} = -\frac{d\bar{q}}{\mathrm{dt}}$, we know that $q(t) + \bar{q}(t) = p$ for all $t \in [0, \infty)$.

Let $I = [t_1, t_2]$ be an interval that satisfies $\phi_1(I)$. It is easy to show by bounding arguments similar to Theorem 2 that the inequality

$$\frac{\mathrm{dq}}{\mathrm{dt}} \geq k\left[-\frac{p^3}{18}((3+d)^{3/2} + 9d) + (1-d)(1-\delta_1)^2(p-q) - 2(1+d)\delta_1\bar{q}\right]$$

holds for all $t \in [t_1, t_1 + \tau]$ where $d = \frac{\delta_4}{k}$. Similarly, we can easily show

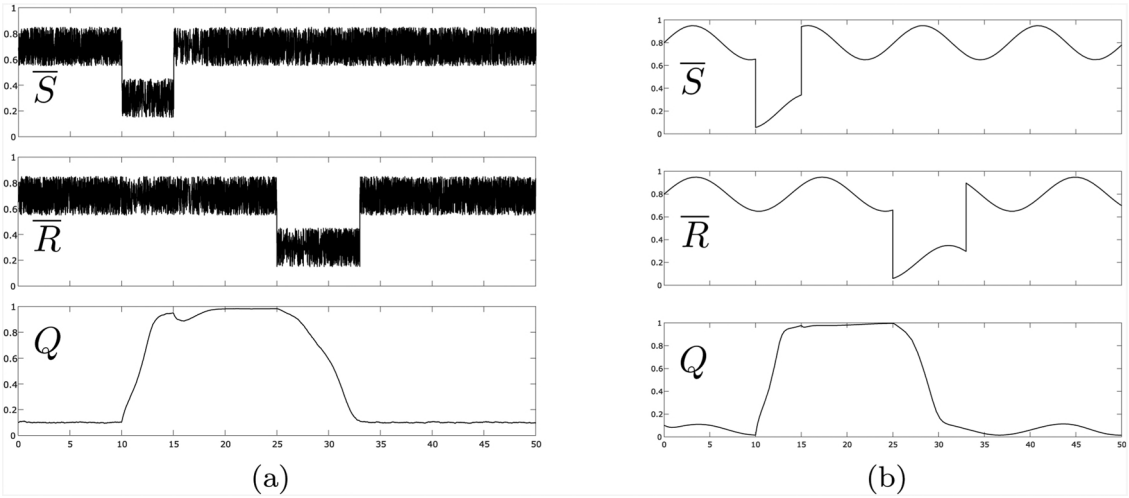

Fig. 5. Simulations of the SR latch design with significant random and sinusoidal noise.

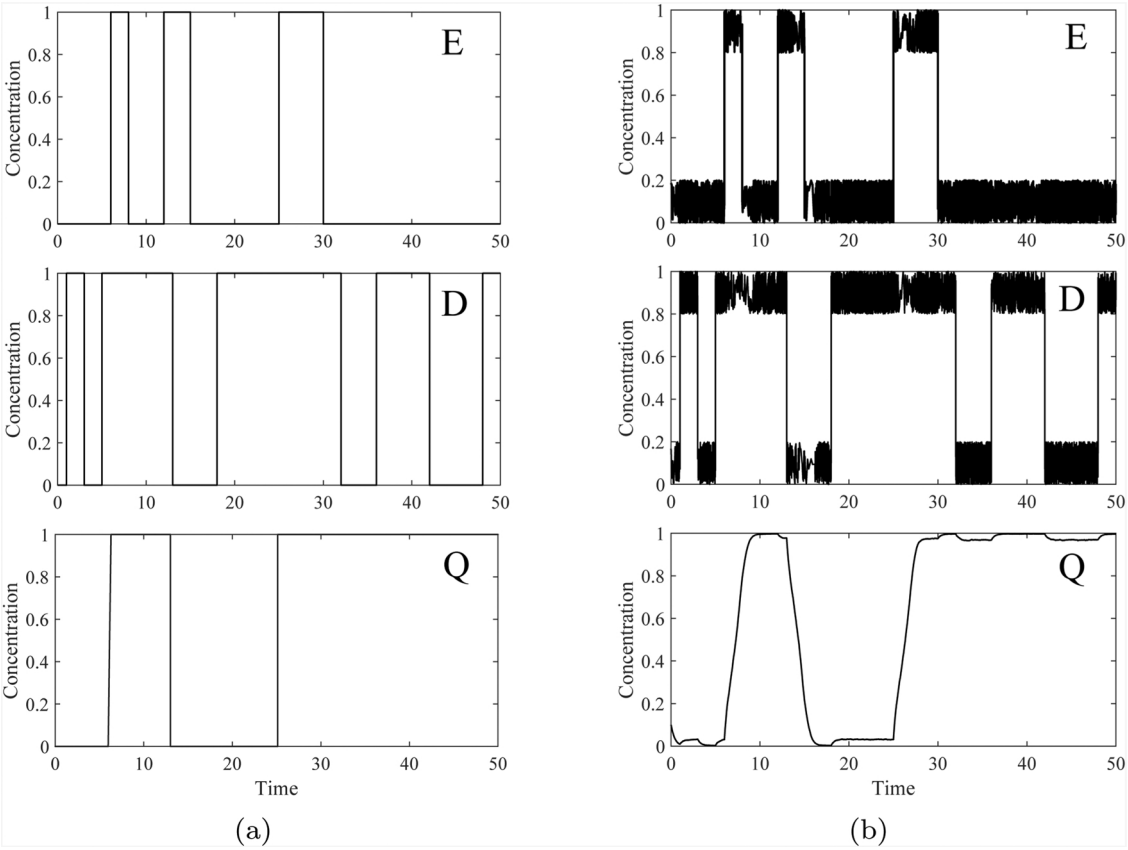

Fig. 6. CRN D latch timing diagram with random noise.

that

$$\frac{dq}{dt} \geq k[3(1-d)q^2(p-q) - 3(1+d)q(p-q)^2 - 2\delta_1(1+d)q]$$

holds for all $t \in I$. Thus by Lemmas 3 and 4, we see that $q(t) > p - \gamma$ and $\bar{q}(t) < \gamma$ for all $t \in [t_1 + \tau, t_2]$ where $\gamma = \delta_1 - \delta_2 - \delta_3$. Thus $\hat{\boldsymbol{x}}$ is δ_1-close to satisfying $\psi_1(I)$.

By symmetry, if I is an interval that satisfies $\phi_0(I)$, then $\hat{\boldsymbol{x}}$ is δ_1-close to satisfying $\psi_1(I)$. Therefore $\mathcal{N} \vDash^{\delta}_{\delta_1} \Phi_{\mathrm{DL}}(\tau)$. □

A simulation of the D latch operating on an input is visualized in Fig. 6. Again, random noise is added to demonstrate the robustness of the construction.

6. Discussion

We have shown that any Boolean circuit can be implemented by a robust input/output chemical reaction network. By "robust" we mean that it tolerates bounded *adversarial* manipulation to the input signals, initial concentrations, reaction rate constants, and output measurements. Thus, not only do we account for random noise, but also the *worst case* scenario due to an adversary. A key feature of our construction is that it preserves robustness under composition. Furthermore, each circuit uses its input signals catalytically, making fanout gates unnecessary. Thus, adding gates to a Boolean circuit is easy and does not affect its robustness, however, it does increase the propagation delay if the new gates increase the depth of the circuit. Preservation of robustness in this way allows designers to construct more complex circuits without needing to prove additional robustness theorems.

We have also shown that two sequential memory circuits can be implemented with robust I/O CRNs. First, we showed that an SR latch can be constructed by composing two NAND gates together. The proof of correctness relies solely on the proven requirements of the NAND gate subcomponents without any additional bounding arguments. We also constructed a robust D latch which uses half the number of species and one-third the number of reactions of the SR latch construction. This was a surprising reduction in complexity since traditional D latch designs use two SR latches (four NAND gates).

Our results are also related to the general purpose analog computer (GPAC). Recently, the GPAC has been proven to be capable of simulating a Turing machine in the presence of bounded perturbations in the initial condition and in the ODEs of all variables (Graça et al., 2008). Our results differ from theirs in the following ways. First, our definition of robustness includes perturbations to the *rate constants* which can cause *unbounded* deviation to the ODEs induced by the CRN. For example, the CRN consisting of the reaction $X \xrightarrow{k} \varnothing$ is modeled with the ODE $\frac{dx}{dt} = -kx(t)$. Any perturbation to the rate constant k can have an arbitrarily large effect on $\frac{dx}{dt}$, since it also depends on the concentration $x(t)$. Second, our I/O CRNs receive their input over time via an *input signal* rather than having all input provided in the initial condition. Furthermore, our input signals are also manipulated by an adversary which can also lead to unbounded error in the ODEs of the species. Third, our definitions include the notion of a *measurement function* from control theory which is used to observe the concentrations of the output species. Finally, we also assume the measurement function is manipulated by an adversary.

There are several drawbacks to our circuit design which we hope will be resolved in future research. First, our NAND gate construction requires three termolecular reactions which are implemented in practice as a sequence of bimolecular reactions. This change to the NAND gate affects its dynamics, and we do not know if it will continue to satisfy its requirement in this case. Previous investigations show that some termolecular algorithms can be reduced to bimolecular ones and maintain robstuness properties. For example, some of our termolecular reactions implement the *signal restoration* algorithm which has been shown to be exactly related to an equivalent bimolecular system (Klinge, 2016b), but it remains an open question whether our circuits provably satisfy their requirements in a

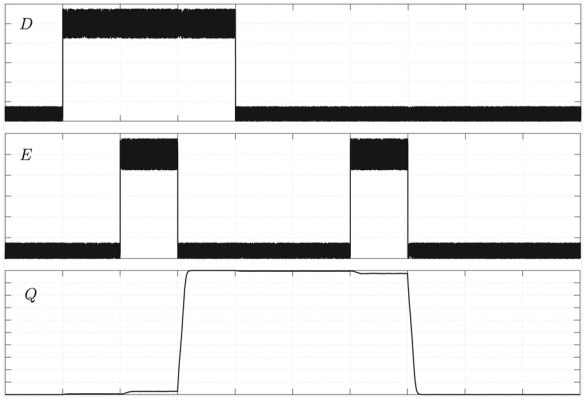

Fig. 7. CRN negative edge-triggered D flip-flop timing diagram with random noise.

bimolecular setting. Several techniques for proving equivalence of stochastic CRNs have been investigated (Lakin et al., 2016; Shin et al., 2017); it may be possible to apply similar approaches to provably guarantee correctness of bimolecular circuits. Second, our circuits depend on reactions that are not reversible. This means that in practice, our circuits will consume *fuel* molecules that provide energy to the system. Further research into reversible I/O CRNs could lead to a construction that does not require fuel, but instead uses the energy introduced by the input signal. Third, our circuit design does not inherently support hysteresis, and therefore circuits instantaneously react to changes in their input. As a result, our construction fails on many common sequential circuits. For example, a ring oscillator circuit constructed by connecting the output of a NAND gate to its own inputs ought to rapidly oscillate between 0 and 1. However, the lack of hysteresis prevents the inputs to the NAND gate from being held at a 0 or 1 value long enough for the output to update accordingly. As a result, the circuit coverages to an equilibrium state rather than rapidly oscillate. This is easy to verify by examining the resulting ODE generated from Construction 1 and examining the stability of its equilibria.

Although some sequential circuits obviously fail, others can be constructed without issue. For example, a negative edge-triggered D flip flop can be constructed using two D latches connected in a master-slave configuration. In Fig. 7, we show a MATLAB Simbiology simulation of an I/O CRN design of this circuit composed of two D latches from Construction 11. The simulations suggest that it works appropriately, and we suspect that techniques similar to those in Section 5 can be used to show it is robust. However, such proofs depend on properly stating the requirements of an edge-triggered flip flop, which is a natural next step to our research.

Conflict of interest

None declared.

Acknowledgments

We thank Jack Lutz and the Laboratory of Molecular Programming at Iowa State University for useful discussions. We also thank our anonymous reviewers for their insightful comments and suggestions.

References

Aris, R., 1965. Prolegomena to the rational analysis of systems of chemical reactions. Arch. Rational Mech. Anal. 19 (2), 81–99.
Arkin, A., Ross, J., 1994. Computational functions in biochemical reaction networks. Biophys. J. 67 (2), 560–578.
Badelt, S., Shin, S.W., Johnson, R.F., Dong, Q., Thachuk, C., Winfree, E., 2017. A general-purpose CRN-to-DSD compiler with formal verification, optimization, and simulation capabilities. Proceedings of the 23rd International Conference on DNA Computing and Molecular Programming, Lecture Notes in Computer Science 232–248.
Beiki, Z., Dorabi, Z.Z., Jahanian, A., 2018. Real parallel and constant delay logic circuit design methodology based on the DNA model-of-computation. Microprocess. Microsyst. 61, 217–226.
Cardelli, L., Tribastone, M., Tschaikowski, M., 2018. From Electric Circuits to Chemical Networks. CoRR abs/1812.03308, http://arxiv.org/abs/1812.03308.
Cardelli, L., 2013. Two-domain DNA strand displacement. Math. Struct. Comput. Sci. 23 (2), 247–271.
Chen, Y.-J., Dalchau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D., Seelig, G., 2013. Programmable chemical controllers made from DNA. Nat. Nanotechnol. 8 (10), 755–762.
Cook, M., Soloveichik, D., Winfree, E., Bruck, J., 2009. Programmability of chemical reaction networks. In: Condon, A., Harel, D., Kok, J.N., Salomaa, A., Winfree, E. (Eds.), Algorithmic Bioprocesses, Natural Computing Series. Springer, pp. 543–584.
Doty, D., 2014. Timing in chemical reaction networks. Proceedings of the 25th Symposium on Discrete Algorithms 772–784.
Ellis, S.J., Henderson, E.R., Klinge, T.H., Lathrop, J.I., Lutz, J.H., Lutz, R.R., Mathur, D., Miner, A.S., 2014. Automated requirements analysis for a molecular watchdog timer. In: Proceedings of the 29th International Conference on Automated Software Engineering. ACM. pp. 767–778.
Ellis, S.J., 2014. Designing a Molecular Watchdog Timer for Safety Critical Systems, Master's Thesis. Iowa State University.
Ellis, S.J., 2017. Devices for Safety-Critical Molecular Programmed Systems, Ph.D. Thesis. Iowa State University.
Fages, F., Le Guludec, G., Bournez, O., Pouly, A., 2017. Strong Turing completeness of continuous chemical reaction networks and compilation of mixed analog-digital programs. In: Proceedings of the 15th International Conference on Computational Methods in Systems Biology. Springer International Publishing. pp. 108–127.
Feinberg, M., 1979. Lectures on Chemical Reaction Networks. http://www.crnt.osu.edu/LecturesOnReactionNetworks.
Garg, S., Shah, S., Bui, H., Song, T., Mokhtar, R., Reif, J., 2018. Renewable time-responsive DNA circuits. Small 1801470.
Ge, L., Zhong, Z., Wen, D., You, X., Zhang, C., 2017. A formal combinational logic synthesis with chemical reaction networks. IEEE Trans. Mol. Biol. Multi-Scale Commun. 3 (1), 33–47.
Graça, D.S., Campagnolo, M.L., Buescu, J., 2008. Computability with polynomial differential equations. Adv. Appl. Math. 40 (3), 330–349.
Gunawardena, J., 2003. Chemical Reaction Network Theory for In-Silico Biologists. http://www.jeremy-gunawardena.com/papers/crnt.pdf.
Hinze, T., Fassler, R., Lenser, T., Dittrich, P., 2009. Register machine computations on binary numbers by oscillating and catalytic chemical reactions modelled using mass-action kinetics. Int. J. Found. Comput. Sci. 20 (3), 411–426.
Hjelmfelt, A., Weinberger, E.D., Ross, J., 1991. Chemical implementation of neural networks and Turing machines. Proc. Natl. Acad. Sci. USA 88 (24), 10983–10987.
Hughes, A., Ellington, A.D., 2017. Synthetic DNA synthesis and assembly: putting the synthetic in synthetic biology. Cold Spring Harbor Perspect. Biol. 9 (1).
Jiang, H., Riedel, M.D., Parhi, K.K., 2013. Digital logic with molecular reactions. In: Proceedings of the 32nd International Conference on Computer-Aided Design. IEEE. pp. 721–727.
Klinge, T.H., Lathrop, J.I., Lutz, J.H., 2016. Work Initially Introduced in (Klinge 2016a) and Will Appear in a Forthcoming Extension of Klinge et al. 2015.
Klinge, T.H., Lathrop, J.I., Lutz, J.H., 2015. Robust Biomolecular Finite Automata. Tech. Rep. 1505.03931. arXiv.org e-Print archive.
Klinge, T.H., 2016a. Modular and Robust Computation With Chemical Reaction Networks, Ph.D. Thesis. Iowa State University.
Klinge, T.H., 2016b. Robust signal restoration in chemical reaction networks. In: Proceedings of the 3rd International Conference on Nanoscale Computing and Communication. ACM. 6:1–6:6.
Krantz, S.G., Parks, H.R., 2002. A Primer of Real Analytic Functions. Springer Science +Business Media.
Lakin, M.R., Parker, D., Cardelli, L., Kwiatkowska, M., Phillips, A., 2012a. Design and analysis of DNA strand displacement devices using probabilistic model checking. J. R. Soc. Interface 9 (72), 1470–1485.
Lakin, M.R., Youssef, S., Cardelli, L., Phillips, A., 2012b. Abstractions for DNA circuit design. J. R. Soc. Interface 9 (68), 470–486.
Lakin, M.R., Stefanovic, D., Phillips, A., 2016. Modular verification of chemical reaction network encodings via serializability analysis. Theor. Comput. Sci. 632, 21–42.
Magnasco, M.O., 1997. Chemical kinetics is Turing universal. Phys. Rev. Lett. 78 (6), 1190–1193.
Qian, L., Winfree, E., 2011. Scaling up digital circuit computation with DNA strand displacement cascades. Science 332 (6034), 1196–1201.
Shannon, C.E., 1941. Mathematical theory of the differential analyzer. Stud. Appl. Math. 20 (1–4), 337–354.
Shin, S.W., Thachuk, C., Winfree, E., 2017. Verifying chemical reaction network implementations: a pathway decomposition approach. Theor. Comput. Sci.
Soloveichik, D., Cook, M., Winfree, E., Bruck, J., 2008. Computation with finite stochastic chemical reaction networks. Nat. Comput. 7 (4), 615–633.
Soloveichik, D., Seelig, G., Winfree, E., 2010. DNA as a universal substrate for chemical kinetics. Proc. Natl. Acad. Sci. 107 (12), 5393–5398.
Srinivas, N., Parkin, J., Seelig, G., Winfree, E., Soloveichik, D., 2017. Enzyme-free nucleic acid dynamical systems. Science 358 (6369).
Yurke, B., Turberfield, A.J., Mills, A.P., Simmel, F.C., Neumann, J.L., 2000. A DNA-fuelled molecular machine made of DNA. Nature 406 (6796), 605–608.
Zhang, D.Y., Seelig, G., 2011. Dynamic DNA nanotechnology using strand-displacement reactions. Nat. Chem. 3 (2), 103–113.
Zhang, D.Y., Winfree, E., 2009. Control of DNA strand displacement kinetics using toehold exchange. J. Am. Chem. Soc. 131 (47), 17303–17314.