TOWARDS PROGRAMMING MATTER WITH CHEMICAL COMPUTERS

by
Dominic L. Scalise

A dissertation submitted to Johns Hopkins University
in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
September 2018

Abstract

In this work, we briefly outline a paradigm for programming the growth and function of physical materials, using chemical computers implemented with DNA. We review experimental studies that enable physical stimuli, such as light, heat, temperature, electricity, and chemical concentrations to be converted into signals encoded in strands of DNA. We also review studies that enable strands of DNA to control the growth and reconfiguration of a library of different molecules and materials. This literature review suggests that one way to program materials is to use embedded chemical computers to read in environmental information encoded in strands of DNA, perform information processing algorithms, and output strands of DNA as commands to downstream materials. Next, we discuss a theoretical framework for building DNA computers that can repeatedly respond to changing input signals, using a chemical buffering reaction analogous to a battery or power supply. In these theoretical studies we demonstrate how the power supply motif could enable DNA computers to generate spatiotemporal patterns of chemical concentrations that remain stable for indefinitely long periods of time. We then discuss an experimental implementation of the buffered power supply motif. Using minor variations on this simple motif, we generate some stable one- and two-dimensional spatial chemical gradients *in vitro*, and present temporal circuits that release different chemical signals at different times. Collectively, this work suggests a mechanism for programming elaborate spatiotemporal behavior into synthetic materials, including growth, healing, and replication.

Committee

Dr. Rebecca Schulman (Chemical and Biomolecular Engineering, and Computer Science. *Advisor. Reader.*)

Dr. Marc Ostermeier (Chemical and Biomolecular Engineering. *Reader.*)

Dr. Jeffrey Gray (Chemical and Biomolecular Engineering)

Dr. Karen Fleming (Biophysics. *Reader.*)

Dr. Michael Dinitz (Computer Science)

Acknowledgements

Thank you to my mentor and PhD advisor Rebecca Schulman. I still recall the thrill of reading your paper *Simple Evolution of Complex Crystal Species* in 2010, and meeting you for the first time shortly thereafter in Stanley Hall at UC Berkeley. I am in as much awe of your intellect now as I was then. You painted a vision in my imagination that day, a vision of extending programming outside of the computer using information encoded in DNA. It felt like opening my eyes for the very first time. You took me under your wing, believed in me, and invested immeasurable amounts of your time and energy into my apprenticeship. While I can be stubborn, you have mentored me patiently for the past seven years. I hope someday to inspire other people as much as you have inspired me.

Thank you also to all of the members of Schulman lab, who are my academic family. We have taught and learned from each other, debugged each other's experiments, and co-authored papers together. We have also carved pumpkins, played laser tag, jumped on trampolines, drunk tea, played go, and eaten a lot of food together. Thank you especially to Abdul Mohammed and Ankur Verma, who served as my mentors in the lab, and also to those I have collaborated closely with: Joshua Fern, John Zenk, Angelo Cangialosi, Phillip Dorsey, Samuel Shaffter, Qi Huang, Shivang Sharma, Wenlu Wang, and Misha Rubanov.

An enormous amount of the work contained within this dissertation was performed by the many talented undergraduate and master's students that I have had the privilege to mentor in the laboratory. Henri Berger, Hee Won Park, Kaiyuan Wang, Ariana Cruz, Jiayao Wu, Dylan Howie, Leo Potters, Nisita Dutta, Madeline Noble, Jenna Jacobs, Alex Baca, Jon Millburn, and Xinyu Cui. Thank you for all of the long hours, late nights, and weekends you spent in the laboratory purifying and pipetting DNA. Thank you also for lending your overwhelming creativity and enthusiasm to our projects. Most of these chapters would not exist without you.

John Zenk, Samuel Schaffter, Hannah Zierden, and Courtney Gonzalez, together we founded and perpetuated *Matlab Made Easy*. I have become a better teacher by observing your unique teaching styles, and from witnessing your dedication to our students. You have each served beyond the two-semester teaching requirement needed to obtain our degrees, and taught purely for the joy of teaching, even at times when the department appeared unwilling to compensate your hard work.

Thank you to Karen Fleming, Jeffrey Gray, Jeannine Heynes, Ana Coughlan, Valerie Hartman, Julia Koehler, Kayla Ghantous, Jo Handelsman, Erin Gleeson, Ashley Llorens, Monica Moore, and the many others who have helped open my eyes to the deep gender and racial inequities that persist in science and academia. These barriers continue to divide our community, dilute our creativity, and impede the pace of scientific innovation. Our combined efforts to generate culture change, through projects such as *Achieving Gender Equity in the Sciences* and *The Women of Hopkins,* have equipped me with the experience and resolve to continue struggling for equality.

To all of my beloved extended family, I could not have survived without your love and support. Dad, you showed me how to think in terms of fundamental principles, how to go towards where the puck is going to be, and told me that I could quit even just before the end. No matter what happens, I know there is someone out there who'll never refuse me a bowl of soup. Mom, I remember sitting with you reading Peter and Jane. You haven't stopped teaching and nurturing me since then. Butterfly Valley taught me two of the most important skills I've used in graduate school, programming and 3D modeling. I admire your ambition, your compassion, and your ability to far-transfer. Jessi, I credit you with my imagination and ability to strategize. You've cast me in plays, read *Huck Fin* out loud to me, let me help on your film sets, and sustained me through the end of grad school with installments of your beautiful writing. You are also the most fun spades partner. Karina, you have always protected me and been my confidant. You are the only person to warn me about how difficult graduate school would be before I started. Despite this, watching you pursue

science made me want to pursue it too, just as watching you learn French made me want to learn French. *Comment t'appelles-Nick*. To the Brookes and Gotfrids, I am thankful that the family I married into is so welcoming, warm, creative, and fun to be around. Being with you on holidays always rejuvenates me. Uncle Mike, Aunt Lea Ann, Lauren, Emma, and Sean, the only enjoyable part of being homesick has been getting to spend more time here with you in Maryland. Dominic, Inez, Stan and Nancy, thank you for teaching me derivation, double-deck pinochle, curiosity and grace.

Most importantly, thank you to my darling wife and dearest friend Nicole Scalise. I am not eloquent enough to express just how deeply I love you, so instead I dedicate this dissertation to you. You have edited nearly everything I've written in graduate school, from my application to my dissertation. You made mug after mug of my favorite tea for me, and you aren't upset when I accidentally let it grow cold. You have held me and been with me through everything. I love you more than when I first met you when we were eleven years old, more than when we first kissed, and even more than the day I married you. I can't wait to grow very old and very wrinkly together.

Table of Contents

List of Figures

1 | Introduction

The total mass of carbon contained in all life on earth is estimated to be about 550 gigatons[1]. This immense biomass has been arranged into diverse forms with structures as large as the Great Barrier Reef, which stretches for thousands of kilometers, and *Sequoia sempervirens*, which can surpass 100 meters in height. While these dimensions are vast, biological components routinely exhibit resolution on the order of angstroms, roughly two orders of magnitude smaller than the resolution of most modern electronics. Furthermore, living organisms are ubiquitously capable of regulating their own internal states, detecting and healing damage, harvesting energy, responding and adapting to stimuli, and self-reproducing. Despite this elaborate sophistication, all 550 gigatons of carbon in life forms on earth became incorporated into these current life forms without any significant help from external assembly lines, clean rooms, or top-down manufacturing processes.

If biology can program atoms and molecules to self-assemble into so many different forms without top-down manufacturing, why can't humans do the same with non-biological materials?

Numerous attempts have been made to engineer answers to this question. They generally fall under the umbrella term of *programmable matter*, or synthetic materials that can change their form and function according to programs of instructions encoded within the material. However, no instance of programmable matter has yet been developed past an experimental prototype stage. Electro-mechanical approaches to programmable matter include modular swarm robotics such as Claytronics[2] and Kilobots[3], in which robotic modules rearrange into different shapes, but these modules have not yet been miniaturized down to functional microscopic sizes. Approaches to constructing stimuli responsive materials include 4-dimensional printing[4] and thermoplastics[5], but these materials have relatively limited programmability that is often hardcoded into the structures of the materials themselves.

In this work we focus on a subset of synthetic biology, called DNA nanotechnology, which uses the hybridization of DNA to process information and regulate self-assembly. DNA is a promising medium for implementing chemical programmable matter due to (1) the simple, predictable rules of Watson-Crick base pairing of DNA, and (2) the relatively diverse library of other materials that DNA can interact with and control. In chapter 2 of this dissertation, we review experimental studies that allow DNA-based computing circuits to sense external stimuli in the environment, and to direct the self-assembly and reconfiguration of many other molecules and materials. In this manner, DNA circuits can be viewed as a scaffold for programming other matter.

From this perspective, the purpose of DNA circuits is to read input strands of DNA that encode information from the environment, perform as information-processing algorithm upon this information, and then release output strands of DNA at different points in space and time to program downstream materials.

Chapters 3-4 outline a theoretical framework, based around the simple buffering reaction motif in Eqn. 1.1, which could allow DNA circuits to respond dynamically to changing input signals. This motif acts as a form of chemical battery or power supply, automatically restoring reactants within the circuit when they are depleted. Simulations presented in chapters 3-4 show how this simple motif could theoretically be used to generate stable patterns of chemical concentrations, which could further be used as spatial blueprints to program the assembly of materials. Chapters 5-6 use this same motif to generate spatial patterns of diffusing chemicals that are not only stable, *i.e.* able to remain constant over time, but are also capable of dynamically changing between a series of different stable states at different times. Specifically, a simulation of a one-dimensional elementary cellular automaton is presented, consisting of the solution to a set of partial differential equations that model chemical reactions and diffusion. Interestingly, John von Neumann, who co-invented the concept of cellular automata in the 1950's, recorded that he was in the process of developing cellular

automata that ran on partial differential equations "of the diffusion type" shortly before his death (see section 6.2).

$$\emptyset \underset{k_2}{\overset{k_1}{\rightleftharpoons}} X \qquad (1.1)$$

Chapters 7-8 present an *in vitro* implementation of the chemical reaction motif in equation 1.1. Specifically, this device is a chemical buffering reaction that uses proportional feedback to hold the concentration of a strand "X" of DNA at a target setpoint concentration. We show how to tune the buffer reaction to generate different setpoint concentrations, demonstrate that the system is capable of pushing the concentration of X back towards the setpoint after transient external disturbances, and show that several different sequences of DNA can be buffered independently within the same solution. Chapters 9-10 present a feedback motif that uses mechanical pumps to holding the concentration of strand X constant, which emulates the same form of kinetics as Eqn. 1 but uses mechanical pumps instead of chemical reactions. The chemical buffering reaction can be viewed as analogous to a battery that holds the concentration of X constant for a finite number of disturbances, until the buffer is depleted, while the mechanical device can be viewed as analogous to a power generator capable of recharging the battery indefinitely many times.

Chapters 11-12 use a variation of the chemical buffer motif to generate simple stable 1- and 2-dimensional concentration patterns in experiments, which demonstrates the spatial aspects of this motif as outlined in the theoretical work in chapters 3-6. Chapters 13-14 use a forward-biased form of the buffer motif to release a pulse of X at a target time, which could be used to program a temporal chemical event such as one step in a self-assembly process. Chapter 15 expands on chapters 13-14 to demonstrate how multiple different pulses can be released conditionally at a series of different times.

References:

[1] Bar-On, Y. M., Phillips, R., & Milo, R. (2018). The biomass distribution on Earth. Proceedings of the National Academy of Sciences, 201711842.
[2] Goldstein, S. C., & Mowry, T. C. (2004). Claytronics: An instance of programmable matter. Wild and crazy ideas session of ASPLOS, 12, 1.
[3] Rubenstein, M., Cornejo, A., & Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. Science, 345(6198), 795-799.
[4] Tibbits, S. (2014). 4D printing: multi-material shape change. Architectural Design, 84(1), 116-121.
[5] Lendlein, A., & Kelch, S. (2002). Shape-memory polymers. Angewandte Chemie International Edition, 41(12), 2034-2057.

Contributions:

Chapter 2. A version of this work has been submitted for publication.

Dominic Scalise and Rebecca Schulman co-authored this literature review together.

Chapters 3-4. A version of this work has been published as:

Scalise, D., & Schulman, R. (2014). Designing modular reaction-diffusion programs for complex pattern formation. *Technology*, 2(01), 55-66.

Dominic Scalise and Rebecca Schulman co-authored this theoretical study together. Dominic designed the theoretical architecture. Dominic and Rebecca planned the simulations. Dominic ran the simulations.

Chapters 5-6. A version of this work has been published as:

Scalise, D., & Schulman, R. (2016). Emulating cellular automata in chemical reaction–diffusion networks. *Natural Computing*, 15(2), 197-214.

Dominic Scalise and Rebecca Schulman co-authored this theoretical study together. Dominic designed the architecture and ran the simulations.

Chapters 7-8. A version of this work has been accepted for publication as:

Scalise, D., Dutta, N., & Schulman, R. (2018). DNA Strand Buffers. *J. Am. Chem. Soc.*

Dominic Scalise, Nisita Dutta, and Rebecca Schulman co-authored this experimental study

together. Dominic designed the theoretical background and the DNA strands. Dominic and Rebecca designed the experiments. Dominic and Nisita ran the experiments. Dominic, Nisita, and Rebecca analyzed the data. Dominic ran the simulations.

Chapters 9-10. A version of this work has been submitted for publication.

Xinyu Cui, Dominic Scalise and Rebecca Schulman co-authored this experimental study together. Dominic designed the theoretical background. Xinyu and Dominic designed the flow reactor. Xinyu ran the experiments and simulations. Dominic, Xinyu, and Rebecca analyzed the data.

Chapters 11-12. A version of this work has been published as:

Zenk, J., Scalise, D., Wang, K., Dorsey, P., Fern, J., Cruz, A., & Schulman, R. (2017). Stable DNA-based reaction–diffusion patterns. *RSC Advances,* 7(29), 18032-18040.

John Zenk, Dominic Scalise, Kaiyuan Wang, Phillip Dorsey, Joshua Fern, Ariana Cruz, and Rebecca Schulman co-authored this experimental study together. Dominic designed the theoretical background and DNA strands. John, Dominic and Rebecca designed the experiments. John, Dominic, Kaiyuan, Phillip, and Ariana ran the experiments. John ran the simulations and analyzed the data. Joshua designed the orthogonal sequences. John, Dominic and Rebecca wrote the manuscript.

Chapters 13-14. A version of this work has been published as:

Fern, J., Scalise, D., Cangialosi, A., Howie, D., Potters, L., & Schulman, R. (2016). DNA strand-displacement timer circuits. ACS synthetic biology, 6(2), 190-193.

Joshua Fern, Dominic Scalise, Angelo Cangialosi, Dylan Howie, Leo Potters, and Rebecca Schulman co-authored this experimental study together. Dominic designed the theoretical background. Dominic and Joshua designed the DNA strands. Joshua, Angelo, Dylan, Leo, and Dominic ran the experiments. Joshua and Dominic ran the simulations. Joshua analyzed the data. Dominic, Josh, and Rebecca wrote the manuscript.

Chapter 15. A version of this work is in preparation for submission for publication.

Dominic Scalise, Leo Potters, Madeline Noble and Rebecca Schulman co-authored this experimental study together. Dominic designed the theoretical background and the DNA strands. Dominic, Leo, and Madeline designed and ran the experiments. Dominic ran the simulations.

2 | Manipulating Matter With DNA Circuits

Summary. A diverse set of mechanisms has been developed that allow DNA strands with specific sequences to sense information in their environment and to control material assembly, disassembly and reconfiguration. These sequences could serve as the inputs and outputs for DNA computing circuits, enabling DNA circuits to act as chemical information processors to program complex behavior in chemical and material systems. This chapter describes the range of processes that can be sensed and controlled within such a paradigm. Specifically, there are interfaces that can release strands of DNA in response to chemical signals, wavelengths of light, pH or electrical signals, as well as DNA strands that can direct the self-assembly and dynamic reconfiguration of DNA nanostructures, regulate particle assemblies, control encapsulation, and manipulate materials including DNA crystals, hydrogels, and vesicles. These interfaces have the potential to enable chemical circuits to exert algorithmic control over responsive materials, which may ultimately lead to the development of materials that grow, heal, and interact dynamically with their environments.

2.1 | Introduction

While publishing the world's first computer algorithm, Ada Lovelace remarked that computers "might act on other things besides number[1]." She noted that computer algorithms could manipulate anything that might be represented as digital information, which includes numbers, music, images, and video. Now, nearly two centuries after Ada's insight, chemical circuits are enabling algorithms to

operate outside of the traditional computer and directly manipulate the behavior of physical materials.

Electronic computers operate on information encoded in electronic voltages. Chemical computers[2], in contrast, are reaction networks that operate on information encoded in the concentrations of molecules. The outputs of these chemical computers are molecules, which can be designed to interface with materials. Using chemical algorithms to direct the behavior of materials could eventually lead to the development of programmable matter in which chemical circuits control self-assembly, detect and heal damage, or adapt dynamically to signals sensed in the environment.

DNA circuits are a particularly promising means for computing within chemical systems. They are composed primarily of DNA oligonucleotides (*i.e.* short strands of DNA), but may also contain enzymes such as DNA polymerase or exonuclease[3-5]. DNA circuits can perform the same fundamental operations as electronic transistor circuits, including Boolean logic and arithmetic[6-9], generating oscillations and regulating time[10-13], and executing interactive algorithms such as playing a game of tic-tac-toe[14].

The inputs to DNA circuits are strands of DNA with specific sequences that can convey information about a material or environment to the circuit. Likewise, the outputs from a DNA circuit are strands that can control the states of downstream materials or molecules. The input/output strands for DNA circuits are thus analogous to USB ports for electronic computers, *i.e.* a standard interface that allows the circuit to communicate with peripheral devices, in this case molecules or materials (Fig. 1). The use of modular input/output interfaces is a key design principle that allows the same types of circuits to interact with a diverse set of materials. An input sensor could, in principle, be exchanged for a different type of sensor to allow the same circuit to receive and process information about a different type of environmental stimulus. Similarly, output actuators could be exchanged to allow the same circuit to direct different material responses.

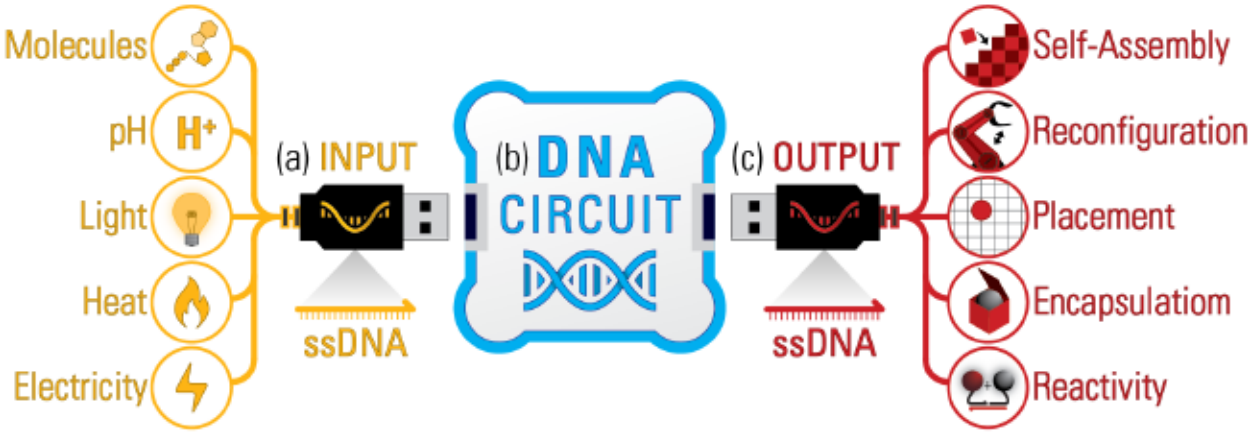

Figure 2.1. DNA circuit interfaces. (a) External stimuli are converted into single-stranded DNA (ssDNA), which serves as a standard interface between stimuli and the DNA circuit in a similar way to how a USB plug serves as a standard interface between peripheral devices and an electronic computer. (b) The DNA circuit reads the input strands, performs a pre-programmed transformation, and releases output strands of ssDNA. (c) The output strands can direct downstream material processes, enabling the DNA circuit to program material behavior.

2.2 | DNA Circuits

DNA circuits can be conceptualized as black boxes that accept DNA strands as inputs, perform a programmed transformation, and release or synthesize strands of DNA as outputs. We collectively term the input and output strands the "interface" strands to distinguish them from the rest of the DNA circuit. The circuit elements within the conceptual black box can include (i) DNA strand-displacement reactions that rely solely on the base-pairing of DNA strands to sequester DNA strand inputs or release DNA outputs (Fig. 2a), (ii) DNAzyme reactions that use the catalytic activity of DNA to make or break the covalent bonds of other strands of DNA (Fig. 2b), and (iii) enzymatic circuits that utilize proteins, such as DNA polymerase or exonuclease, to synthesize, modify, or degrade strands of DNA (Fig. 2c). Circuits composed of these elements can perform many signal

processing functions, including Boolean logic (Fig. 2ab), neural network computation[17,18], signal amplification[19,20], the generation of oscillations and other time dependent signals in the concentration of output molecules[11-13] (Fig. 2c), and the generation of spatially patterned signals where different outputs are released at different spatial locations[11,21,22] (Fig. 2d).

Designing modular interfaces between DNA circuits and materials often requires the circuit to be insulated from the material, and vice versa, so that the circuit does not irreversibly consume the input strands and the material does not irreversibly consume the output strands. In some cases where this irreversible consumption, *i.e.* loading, cannot be avoided, an additional load driver mechanism may need to be inserted to fully insulate the circuit and material[15,16]. A similar challenge can occur if the material imposes conflicting constraints on the exact sequence or secondary structure of the interface strands. In principle, this sequence dependency can be addressed through the use of additional reactions to convert the input strands into entirely independent sequences before they interact with the circuit, and the output strands into different sequences before they interact with the material. This conversion step is generally called "sequence translation". While useful for ensuring modularity, the use of load drivers and sequence translators often comes at the cost of additional circuit complexity.

For the remainder of this review, we will focus on mechanisms that convert environmental stimuli, such as light and heat, to and from interface strands that can interact with DNA circuits. In this scope, we will refrain from further discussing the internal workings of DNA circuit black boxes themselves, which have been covered more extensively in other excellent reviews[3-5].

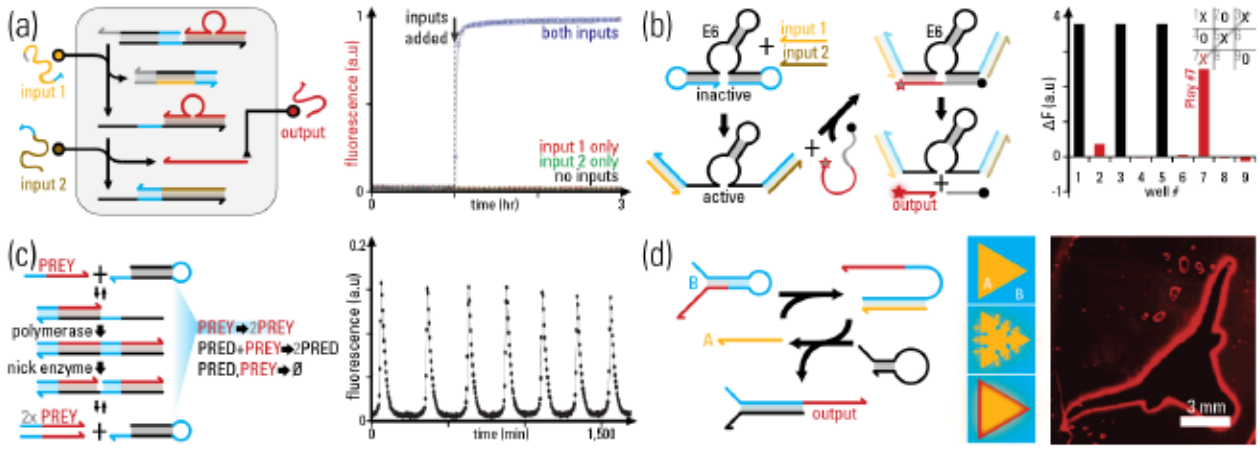

Figure 2.2. Overview of DNA circuits. Input strands of DNA are shown in yellow and output strands are shown in red. (a) A DNA strand-displacement circuit, which processes information *via* competitive DNA base-pairing. Here a Boolean logic AND gate releases an output strand (red) from a complex only when both input strands (yellow, dark yellow) are present[6]. High fluorescence indicates a high concentration of the output strand. (b) A Boolean logic circuit[14] built around the E6 DNAzyme (black) that cleaves a DNA strand, releasing an output strand (red) only when both input strands (yellow, dark yellow) are present. A circuit composed of such gates can "play tic-tac-toe" if it receives input strands that encode information about the presence of X's played by the circuit, and O's played by the human, and produces an output corresponding to the next move to be played. Right, the final step of a game: the large red bar indicates the circuit's direction to play in the 7th cell of the game board to win the round. (c) A dynamic DNA circuit that produces an oscillatory output because of coupled DNA polymerization and nicking reactions, catalyzed by enzymes[11]. This particular circuit is a signal generator and does not have a designated input strand. (d) A DNA edge-detection circuit. Two reactants A and B were initially present in different parts of the pattern. Photocleavage of reactants in certain areas, allowed the reactants to diffuse and encounter one another. The resulting spatial computation process produces output strands only at edges where A is initially present on one side of the edge and B is initially present on the other side[21].

2.3 | Transducing signals from DNA to other forms.

Transduction is the conversion of information, from one signal form to another. Information

encoded in the concentration of a strand of DNA can be transduced from and to many other forms, including the concentration of other molecules, ions or as pH, light, temperature or electrical signals. In this section we review methods for transducing these signals into interface strands of DNA, thus allowing DNA circuits to receive information about stimuli in the surrounding environment.

2.3.1 | Molecules-to-DNA. DNA aptamers are sequences of DNA that can bind specifically to molecules such as ATP[24], cocaine[25], metal ions[26], proteins and peptides[27-31]. Aptaswitches are nucleic acid sequences or complexes that can bind to a target molecule and in doing so can further induce a conformational change of another part of the aptaswitch. In many cases this conformational change can release a strand of DNA[32-35], which can then serve as a DNA circuit input. Aptaswitches can also expose a toehold domain in a DNA complex[36], which can allow a strand-displacement reaction to proceed. Aptaswitches can also displace domains that would otherwise inhibit other strands from binding to a DNA complex[37].

Output strands released by DNA circuits can also direct the capture or release of molecules from an aptamer by binding to the aptamer, thus changing the aptamer's binding affinity for its target. Bahdra and Ellington modified the fluorescent RNA Spinach aptamer to fold into an inactive state in which it did not associate with its target molecule (DFHBI fluorophore). Hybridization with a trigger strand of DNA refolded the inactive aptamer into an active state in which it successfully bound its target[38]. Lloyd *et al* used DNA strands complementary to aptamer sequences, called kleptamers, to bind to and displace aptamers from their targets. This technique was demonstrated with both a Broccoli aptamer, which binds to DFHBI, and an aptamer for a RNA polymerase, which prevents the RNA polymerase from transcribing while bound to the aptamer[39] (Fig. 3b). Aptamers have also been integrated into reconfigurable DNA nanostructures. For example, they have been used as binding sites for target molecules at the end of DNA nanotweezers[40]. When a

trigger strand of DNA opens the tweezers, the two aptamer binding sites at the ends of the tweezers separate. Because binding of the target protein requires interaction with both binding sites, this conformational change releases the protein target. The tweezers nanostructure thus allows a strand of DNA that is not itself an aptamer binding sequence to direct the release of a target molecule.

2.3.2 | Ions and pH-to-DNA. Strand-displacement reactions can be designed to release an interface strand of DNA only in the presence of target ions, which can then serve as an input strand to communicate the presence of the target ions to a DNA circuit. For example, Liu and Mao used Hoogsteen triplex formation, in which CGC triplets form under acidic conditions where $[H^+]$ is high, to co-localize a strand of DNA to a complex and initiate a strand-displacement reaction[41]. The release of the interface strand from this reaction thus signifies a high H^+ concentration. Amodio *et al* implemented a similar pH-responsive strand-displacement reaction that uses CGC triplets in a Hoogsteen triplex to stabilize both a clamp domain, which suppresses a DNA strand-displacement reaction, and a toehold domain, which initiates a strand-displacement reaction[42].

Tang *et al* split the sequences for a pH-dependent i-motif between an invading strand and a strand-displacement complex. At low pH, the formation of the complete i-motif co-localizes the strand to the complex, initiating a strand-displacement reaction that releases an interface strand of DNA[43]. The authors also used a similar mechanism based on the G-quadruplex motif, which formed on the addition of strontium Sr^{++} ions to allow a strand-displacement reaction to occur only in the presence of Sr^{++} ions. Ding *et al* developed a DNA strand-displacement reaction that is triggered by mercury (Hg^{++}) ions[44]. The authors used a thymine-thymine (T-T) mismatch in the toehold domain of a strand-displacement complex to prevent an invading strand of DNA from binding to this toehold. However, when mercury ions are added, a non-Watson-Crick T-T base pair is stabilized between the invading strand and the complex, allowing the toehold domain to fully hybridize and

initiate a strand-displacement reaction. (Fig. 3c).

To the best of our knowledge, no experimental studies have reported the use of a DNA strand to directly trigger the release or sequestration of ions, rather than use of ions to release DNA strands. Such a DNA-to-ion interface might allow DNA circuits to dynamically adjust the pH over time, or change pH suddenly in response to signals detected in the environment.

2.3.3 | Electrical-to-DNA Electronic devices can provide inputs to DNA circuits through electrodes that release or activate interface strands of DNA. Ranallo *et al* developed four different mechanisms for activating DNA strand-displacement reactions by using voltage to release specific ions from gold electrodes. First, a gold electrode was coated with a thin film of mercury. When an oxidizing potential scan from 0.2 to 0.65 V was applied to the electrode, the mercury was released into solution as Hg^{++} ions. The released ions facilitated binding of target partially complementary DNA domains by stabilizing non-Watson-Crick T-T base pairs (Fig. 3d. See also section 3.2 on the ionic activation of DNA). Application of a reductive potential (-0.3 V) reversed the stabilization of the T-T base pairs by re-depositing the mercury back onto the electrode. The authors also electrically activated other DNA circuits by (ii) releasing silver ions to facilitate non-Watson-Crick C-C base pairs, (iii) releasing input strands of DNA initially bound to the electrode by thiol-gold bonds, and (iv) releasing copper ions to activate a copper-sensitive DNAzyme[45]. These strategies might be used together to selectively activate or release different interface strands of DNA within the same solution. Other studies have electrochemically controlled the activity of DNAzymes using ions other than copper[46,47]. Jeong *et al* used an entirely different technique to implement voltage-directed DNA strand activation by fabricating DNA-impregnated multilayer nanofilms on the surfaces of electrodes. Application of voltage to one such electrode reduced the nanofilm and

released the impregnated strands of DNA into solution where they interacted with a fluorescently labeled DNA beacon[48].

DNA strands can produce or regulate voltage or current by regulating the distance between an electrode and an electrically active molecular tag. The close proximity of the electrode and the tag generates a Faradaic current. Fan *et al* and Immoos *et al* both demonstrated this technique using a redox ferrocene-labeled DNA hairpin immobilized on a gold electrode[49, 50]. When no other DNA strands were present, the DNA hairpin's lowest energy conformation kept the ferrocene close to the electrode, generating current. When a DNA strand with a sequence complementary to the hairpin's was added, binding between this interface strand and the immobilized hairpin opened the immobilized hairpin, separating the ferrocene from the electrode and decreasing the measured current. Similar schemes involving electrodes with surface-immobilized DNA have used different labels to generate a current, including methylene blue[51, 52] (Fig. 3e), and conformational switches of DNA motifs other than hairpins[53]. Electrochemical DNA sensors are further reviewed in Drummond *et al*[54].

2.3.4 | Temperature-to-DNA. Most current experimental studies involving DNA circuits have tended to be designed to operate at a particular set temperature, rather than to respond to temperature as an input. However, some DNA hairpin structures have been designed to serve as temperature responsive thermometers by tuning the strength, and thereby the melting temperature, of the stem domains that hold the hairpins closed[55,56]. When the stem melts, the hairpin opens into a single-stranded conformation. Further, Gehrels *et al* modified the temperature dependence of DNA-grafted colloids to program nonlinear aggregation and reentrant melting of the colloids, which suggests an alternative approach to programming temperature sensitive DNA strand-displacement reactions[57,58]. Specifically, the authors started with colloidal assemblies held together by DNA linkers

between the colloids, and added complementary interface strands of DNA to bind to and displace the linkages from each other. At low temperatures, enthalpy favors disruption of the linkages and melting of the colloidal assemblies, because more total bonds can form when the competing strands sequester the linkages. At intermediate temperatures, entropy favors formation of the linkages, release of the interface strands and stabilization of the colloidal assemblies, because two strands are displaced by every linkage formation. At high temperatures complete thermal dissociation of DNA hybridization melts the linkages and the colloidal assemblies (Fig. 3f). This reaction system shows how DNA strands can be released or sequestered at a target temperature, which could be used to communicate information about the temperature of the environment to a downstream DNA circuit.

To the best of our knowledge, no studies have demonstrated that a specific DNA strand can directly alter the temperature of the surrounding solution. This class of DNA-to-temperature interface could be useful for creating programmable three-dimensional spatio-temporal temperature gradients within solutions. However, regulating temperature with DNA strands may prove difficult because the amount of energy that is required to heat water is large compared to the energy stored in DNA hybridization. Such a control system may thus need to harness an external energy supply, such as using light illumination plasmon resonance to generate heat from gold nanoparticles[59]. With such a setup, interface strands of DNA might control where the gold nanoparticles are positioned to direct local heating.

2.3.5 | Light-to-DNA. Light can direct the release of a specific DNA sequence by controlling the degradation or conformational change of reagents that initially block or sequester a DNA domain from participating in downstream reactions. Prokup *et al* created a photosensitive strand-displacement reaction by suppressing the ability of a strand to hybridize to its complement using four NPOM (6-nitropiperonyloxy- methylene)-caged thymidine groups evenly distributed

along the strand. The authors found this photo-caged strand did not react with a downstream strand-displacement circuit until UV irradiation at 365nm removed the caging groups[60]. Huang *et al* synthesized DNA a hairpin gate with a photocleavable linker in the hairpin backbone. On exposure to 365 nm light, the hairpin's backbone was cleaved, exposing a toehold domain within the hairpin loop. This exposed toehold allowed a strand with a complementary toehold to bind and initiate a strand-displacement reaction[61] (Fig. 3g). Kou *et al* inserted *cis*-20,60-dimethylazobenzene (DMazo) into a strand of DNA, which favors duplex association in its *trans* state and dissociation in its *cis* state. By toggling the DMazo into its *trans* state with 475 nm light, the authors initiated a strand-displacement reaction. Toggling the DMazo back into its *cis* state by exposure to 365 nm light induced dissociation of DMazo strands from their complementary sequences, driving the strand-displacement reaction in the reverse direction[62]. Nakamura *et al* used a photo-initiated crosslinker (3-cyanovinylcarbazole, ^{CNV}K) to form a covalent bond between an invading strand of DNA containing ^{CNV}K and a pyrimidine base in a complementary DNA complex, after exposure to light at a wavelength of 366 nm. This covalent bond made a strand-displacement reaction effectively irreversible and sped up its kinetics by a reported factor of 20x by inhibiting the backward branch-migration step[63].

A standard way to transduce a DNA hybridization event into a light signal is to use a fluorescent molecule conjugated to the end of a strand of DNA to emit different intensities of light at a target wavelength, depending on the state of the conjugated DNA strand. Emission from the fluorophore can be a quenched by a nearby quencher molecule or transferred by FRET (fluorescence resonant energy transfer) to a different fluorophore, which effectively changes the wavelength of the fluorescence output signal. An interface strand of DNA can change the distance between these different fluorophore and quencher modifications, by opening a fluorophore-modified hairpin[64] (Fig. 3h), or displacing a fluorophore-modified strand from a complex (6). The kinetics of many of the

devices described in this review are monitored by measuring changes in fluorescence from fluorescently modified DNA complexes.

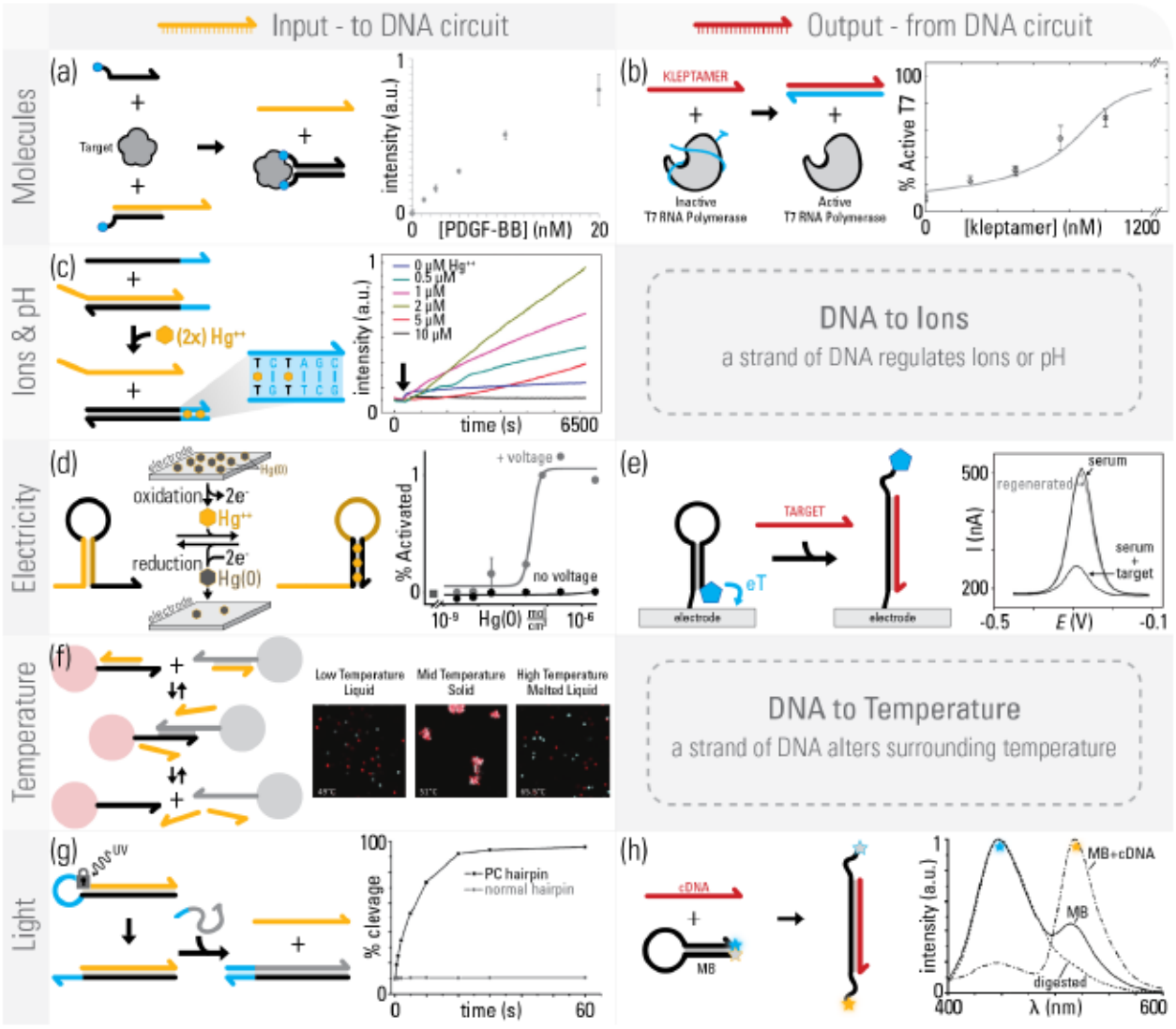

Figure 2.3. Transducing signals to and from DNA sequences. (a) An interface that converts the concentration of a target molecule (gray, platelet derived growth factor PDGF-BB) into the concentration of an interface strand of DNA (yellow). Two aptamer sequences of DNA (cyan) bind to the target molecule and displace the interface strand. A downstream fluorescent reporter complex tracks the concentration of the yellow strand[32]. (b) A kleptamer (red) displaces an aptamer from T7 RNA polymerase. While T7 RNA polymerase is bound by the aptamer it cannot transcribe RNA[39]. (c) An Hg^{++}-dependent DNA strand-

displacement (DSD) reaction that releases the interface strand (yellow). A downstream fluorescent reporter tracks the concentration of the yellow strand over time. Black arrow marks time when Hg^{++} ions were added[44]. (d) A gold electrode is initially coated in mercury Hg(0). When an oxidative potential is applied to the electrode, Hg(0) is oxidized into Hg^{++} which can promote DNA hybridization using the same mechanism as in panel (c). Application of a reductive potential reverses the process[45]. (e) A hairpin (black) modified with a methylene blue redox moiety (MB, cyan) is anchored on an electrode. Proximity of the MB to the electrode generates a Faradic current. A complementary interface strand of DNA (red) opens the hairpin, and prevents the MB from interacting with the electrode, suppressing the current. Data shows high current for the initial state with no interface strand added (*i.e.* serum only), decreased current when the interface is added (*i.e.* serum + target), and high current again when the interface is removed by a washing regeneration step (*i.e.* "regenerated")[52]. (f) A temperature-sensitive DSD reaction that directs DNA-modified (black, gray) particle assembly within a target temperature range. Below the target temperature, strands of DNA (yellow) bind to the linker strands and prevent assembly. At the target temperature, the particles bind to form a solid assembly, and displace the yellow strands into solution. Above the target temperature, all strands are thermally dissociated[57]. (g) A DNA hairpin with a photocleavable linker (gray lock symbol) presents a toehold domain (cyan) after exposure to UV light, which allows a strand-displacement reaction to release the yellow strand[61]. (h) A molecular beacon (MB, black) is a DNA hairpin with two different fluorophores at its ends. In the beacon's initial state, the fluorophores are close to each other, enabling fluorescent resonant energy transfer (FRET) to excite the acceptor fluorophore (FAM) *via* excitation of the donor fluorophore (coumarin). After a complementary strand of DNA is added (red) that binds to and opens the hairpin, the two fluorophores are separated and the intensity of the acceptor fluorophore decreases significantly[64].

2.4 | Controlling DNA nanostructure assembly and reconfiguration.

Strands of DNA that are output from a DNA circuit can interact with DNA nanostructures by serving (i) directly as a building block in a self-assembly process, (ii) as a trigger that activates one or more building blocks, or (iii) as an agent that removes DNA strands or sub-assemblies from an

initially assembled structure. Such interactions effectively use the sequence information encoded in a circuit's output strand to direct how an assembly should form or change shape.

2.4.1 | Self-assembly. DNA can self-assemble into virtually any arbitrary two- or three-dimensional shape[65-68] up to several hundred nanometers in size[69], and into a diverse range of periodic lattices and crystals[70,71]. DNA nanostructures may be broadly classified into either scaffolded DNA origami nanostructures[65], in which short "staple" strands fold a long scaffold strand into a final shape, or unscaffolded assemblies in which DNA bricks[66] or tiles[72-73] self-assemble into ordered lattices *via* a crystal growth process. In both cases, the specific sequences of each strand determine which components may bind to which others and how strongly and, in turn, determine what final structures self-assembled.

In principle, DNA circuits could be used to direct the self-assembly of these structures by presenting outputs that control which strands are available to self-assemble[74]. Dynamic control over the availability of different DNA strands might allow circuits to direct elaborate, hierarchical assembly protocols, or to incorporate feedback from the environment during assembly. One challenge to the design of such systems is that DNA nanostructures are generally formed by annealing: the temperature of the assembly mixture is first raised above the melting temperature of the DNA and then slowly cooled. Assembly occurs after the temperature drops below the nanostructure's melting temperature. Parameters other than temperature, such as denaturant concentration[75]), have also been used for annealing. During an annealing process, most DNA circuits, which are held together by Watson-Crick base pairs, also melt and reform. Such melting and reformation of circuit components would interrupt the function of the many types of DNA circuits that are designed to begin operation in a metastable configuration and could denature enzymes or other co-factors needed for the operation of other DNA circuits. Because circuits are typically

designed to operate isothermally, these circuits would need to be redesigned to operate during an annealing process.

An alternative approach to controlling self-assembly with DNA circuits is to develop mechanisms by which DNA nanostructures assemble isothermally. Isothermal nanostructure assembly processes are generally more difficult to design than assembly processes that use annealing because under isothermal conditions the component strands of DNA assemblies can easily become kinetically trapped in off-target intermediate structures. In principle, these kinetic traps could be avoided by using DNA circuits to control the pathway in which assembly components are made available.

Activation of nanostructure building blocks. One strategy for controlling DNA nanostructure self-assembly is to use DNA strands to activate the building blocks of a target nanostructure. Zhang *et al* used DNA strand-displacement circuits to control the self-assembly of DNA nanotubes composed of DNA tiles[76]. The authors constructed inactive tiles where two of the inter-tile "sticky end" binding sites on each tile were covered by a DNA protector strand, which inhibited nanotube growth. A complementary interface strand, released by a DNA circuit, was shown to successfully remove the protection strand from the inactive tiles, thus activating the tiles and allowing nanotube assembly to occur (Fig. 4a). Franco *et al* demonstrated the reversible assembly and disassembly of similar DNA nanotubes by repeatedly protecting and de-protecting a tile's sticky end strands in response to changes in pH[77]. Padilla *et al* created DNA tiles containing inactive binding domains, in which the binding of a sticky end on one side of these tiles partially displaced a cover strand of DNA. This cover strand could then activate a sticky-end on the opposite end of the tile[78]. A DNA circuit might presumably trigger a cascading assembly process involving such tiles by activating an initial tile, which then uses its newly activated sticky-end to

activate a chain of additional tiles. Zhang *et al* demonstrated an analogous sequential assembly cascade in which droplets, rather than tiles, are assembled into chains. Droplets were functionalized with two DNA attachment strands that were initially hybridized together. Inter-droplet assembly was initiated by adding an interface strand of DNA that binds to one of the attachment strands *via* a single-stranded toehold, displacing the second attachment strand, which could then attach to another droplet[79].

Wang *et al* reversibly activated and deactivated larger DNA origami building blocks that could form different dimers depending on how they were activated[80] (Fig. 4b,). The authors were able to reversibly assemble four different tile types (labeled A, B, C, and D) into different permutations of dimers (AB, CD, AD, BC) by adding different combinations of protector and deprotector strands to regulate different binding interfaces. Four cycles of reversible dimerization were reported, with dimerization yields around 75%. DNA circuits might presumably direct assembly of origami tile with such interfaces into larger structures by releasing sequences of the relevant protector and deprotector strands.

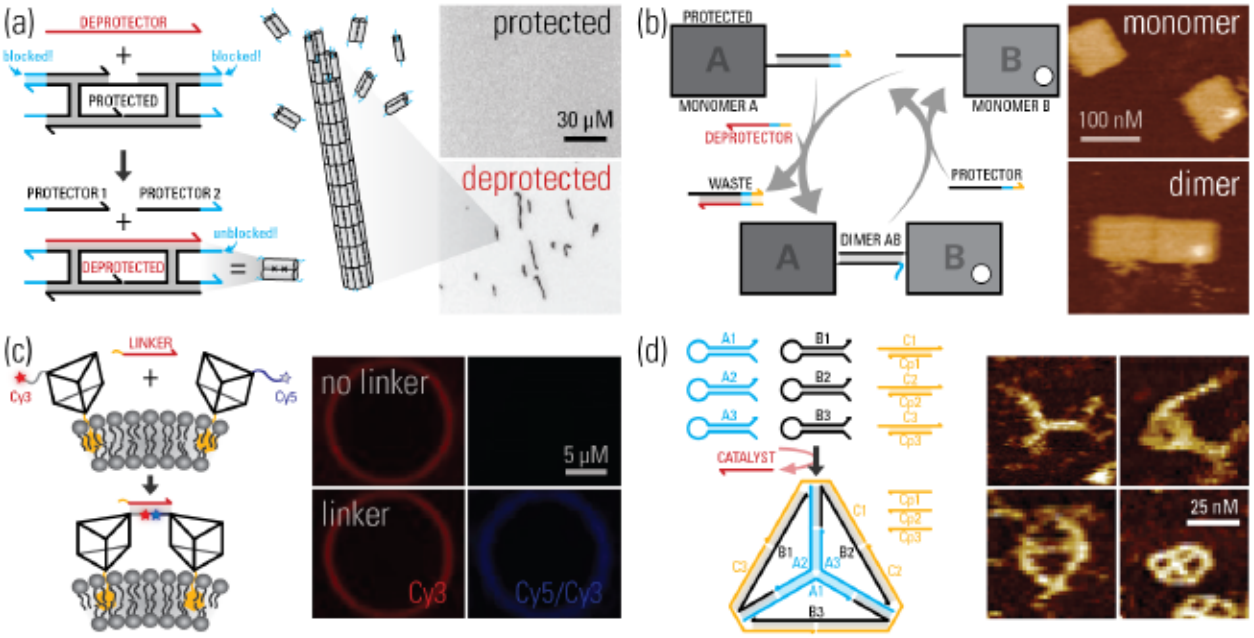

Figure 2.4. DNA strand-directed self-assembly. In each example, the red strand of DNA represents the interface strand, which could be the output of a DNA circuit, that directs the self-assembly of a nanostructure. (a) Activatable DNA tiles[76]. Each tile starts in an inactive state in which two of its four self-complementary sticky ends (cyan) are covered by protector strands of DNA. A deprotector strand (red) binds to a single-stranded toehold region at the center of the tile and displaces the protector strands, resulting in an active or deprotected tile. Deprotected tiles then proceed to self-assemble into nanotube structures. TIRF fluorescence microscopy images show (top) the tiles in their protected state, before deprotector strands are added when no nanotubes are evident, and (bottom) in their deprotected state after deprotector strands were added, where nanotubes have grown. (b) Scaffolded DNA origami tiles whose binding sites can be activated by DNA interface strands. A deprotector strand (red) binds to monomer A and removes its protector strand, creating a waste complex and allowing formation of a dimer AB. Introduction of a protector strand separates the monomers back into their initial state[80]. Additional sequence-specific binding sites (not shown) facilitate different permutations of dimers. AFM images show origami tiles in their monomer and dimer states. (c) Regulating self-assembly through the release of DNA linker strands that serve as bridges between two structures. Here a linker strand (red) connects two DNA triangular prisms on a giant vesicle[81]. (d) Catalytic assembly of a DNA tetrahedron. Initially, a solution was prepared that contained nine metastable hairpins

(cyan, black, yellow), which comprise the parts of the tetrahedron. A catalyst strand (red) opens the A1 hairpin, initiating a cascade in which the tetrahedron forms *via* a series of reactions that follow a designed assembly pathway. AFM images show the intermediate structures: a three-arm junction, a closed triangle, two closed triangles, and the final tetrahedron[88].

Releasing linker strands. DNA sequences can also direct the self-assembly of DNA nanostructures by acting as sequence-specific bridges between two other DNA structures. Unlike the use of interface strands to activate nanostructures described in the previous section, using interface strands as direct linkers helps separate structural building blocks from the sequences that determine connectivity, allowing the same building blocks to bind to different partners depending on which linker is activated.

Peng *et al* used linker strands of DNA to assemble multiple triangular wireframe DNA prisms anchored on vesicles into dimers[81] (Fig. 4c). Displacement strands that were complementary to the linker strands were used to reverse the assembly process. Cao *et al* used different sequences of linker strands of DNA to selectively control the assembly of DNA-functionalized Au and Ag/Au nanoparticles into aggregates, which was used as a two-color mechanism for detection of DNA[82]. Yao *et al* used DNA strand-displacement circuits to control the self-assembly of DNA-functionalized gold nanoparticles by using output strands from a DNA circuit to link particles together[83]. Yao *et al* further explored differences in reaction kinetics for strand-displacement reactions when the reactant strands were conjugated to gold nanoparticles, as opposed to standard strand-displacement kinetics without nanoparticle conjugations[84]. For a more detailed discussion on how DNA strands can control the aggregation of DNA-functionalized nanoparticles, see the review by Thaxton *et al*[85] and the perspective article by Cutler *et al*[86].

Gated catalysts and nucleation sites. Single-stranded outputs from circuits have also been used to activate sites for nanostructure nucleation, or catalysts for downstream self-assembly

processes. This strategy for using DNA strands to control self-assembly effectively amplifies the signal, because each DNA input triggers a multiple assembly steps. Yin *et al* developed a metastable hairpin assembly architecture in which a catalyst initiates a cascade of assembly steps that assemble DNA hairpins into larger structures[87]. Sadowski *et al* used this metastable hairpin architecture to assemble a tetrahedron in response to a catalyst trigger[88] (Fig. 4d). Zhang *et al* used a DNA strand-displacement amplifier circuit to activate a metastable hairpin chain reaction process that triggered the assembly of a branched structure[89]. Kishi *et al* developed isothermal primer exchange (PER) cascades, in which DNA hairpins are templates on which DNA polymerase synthesizes strands of DNA. Each hairpin adds one short sequence domain to a growing strand in a programmed order so that the set of hairpins present controls what DNA strand is synthesized[90]. Using this mechanism, the authors further created a PER hairpin that was initially inactivated by a protector strand blocking the hairpin's toehold. When an interface strand complementary to the protector strand was added, the hairpin's toehold was exposed and DNA synthesis proceeded.

2.4.2 | Actuators. Beyond directing the self-assembly of structures and materials from components, DNA can also be used to change the shapes of nanostructures that have already been assembled. A common mechanism of changing the shape of a DNA nanostructure is to add or remove strands of DNA that constrain the nanostructure in different configurations from a nanostructure. Yurke *et al* used this technique to repeatedly open and close a pair of DNA tweezers: adding a closing strand constrained the tweezers in a closed configuration, while an opening strand displaced the closing strand, thereby re-opening the tweezers[91] (Fig. 5a). The state of the tweezers was monitored by fluorescence energy resonance transfer (FRET) between two fluorophores positioned at the end of the tweezers. A similar nanoactuator was also developed in which an interface strand of DNA constrained the two ends of the actuator in an open state, rather then in a

closed state[92]. Feng *et al* constructed a linear actuator, in which the two ends of the device are pushed away from each other along a straight track when an actuation strand is added. They then incorporated this device into a two-dimensional DNA lattice to actuate expansion of the lattice[93]. Goodman *et al* used a similar linear actuator to reversibly change the length of a side of a DNA polyhedron[94], while Saccà *et al* used linear actuators to extend and contract the edges of a cavity within a DNA origami tile[95]. Chen *et al* developed a scheme in which linker strands bind along the edge of a DNA origami tile to direct the folding of the origami into a cylinder[96].

Several rotary devices have also been developed, in which an interface strand can direct a change in the relative angle between two parts of a nanostructure. Yan *et al* created an axial rotation device that reversibly rotates 180° between a paranemic crossover (PX) motif and its topoisomer (JX_2) in response to two sets of actuation strands[97]. This motif was inserted as a modular component on a DNA array, which provided a fixed frame of reference for integrating multiple rotational actuators together into the same device[98]. Marras *et al* developed a library of different DNA origami joints for both linear and rotational motion, and used interface strands of DNA to control the opening and closing of one of their composite devices[99]. Gerling *et al* used DNA strand-displacement to actuate the rotational movement of a four-armed DNA nanostructure, in which blunt end stacking between shape-complementary components of the device provided the driving force for constraining the actuator with the arms either together or apart[100] (Fig. 5b).

2.4.3 | Subtractive modification. Strands of DNA can also change the shape of a structure by removal or disassembly of targeted components. Han *et al* built a DNA origami Möbius strip and used strand-displacement processes by DNA interface strands to remove staple strands of DNA at target locations along the structure such that the removal of the staples acted akin to cutting[101]. The authors first added DNA strands to "cut" along the centerline of the Möbius strip to create a longer

loop with two full twists, and then added DNA strands to remove the staples along a line one-third of the width into the Möbius strip to create two interlocked loops. Zhang *et al* used strand-displacement to both remove and replace staple strands in an origami structure, transforming a square structure into a quasifractal pattern of squares of decreasing sizes[102]. Wei *et al* created regular square and cuboidal DNA brick "canvases" where any component strand could be removed adding a complementary strand[103] (Fig. 5c). This technique allowed the authors to change the shape of a general-purpose square or cuboidal canvas into many shapes, including all twenty-six letters of the alphabet.

2.4.4 | DNA walkers and molecular transport. DNA walkers are mobile nanostructures that bind to and traverse a larger frame-of-reference surface, such as a DNA origami tile surface. These types of devices can be used to transport cargo molecules to different locations on the origami surface[104,105] (Fig. 6b). Walkers can be thought of as specific types of reconfigurable DNA nanostructures. DNA fuel strands can drive walker motion, and can also act as decision making signals that indicate to the walker which path to take among a set of option branches by selectively removing blocker strands from the target pathway[106] (Fig. 5d). Numerous DNA walkers have been designed, and many walkers whose motion can be controlled by DNA strands are discussed in further detail in the review by Wei *et al*[107].

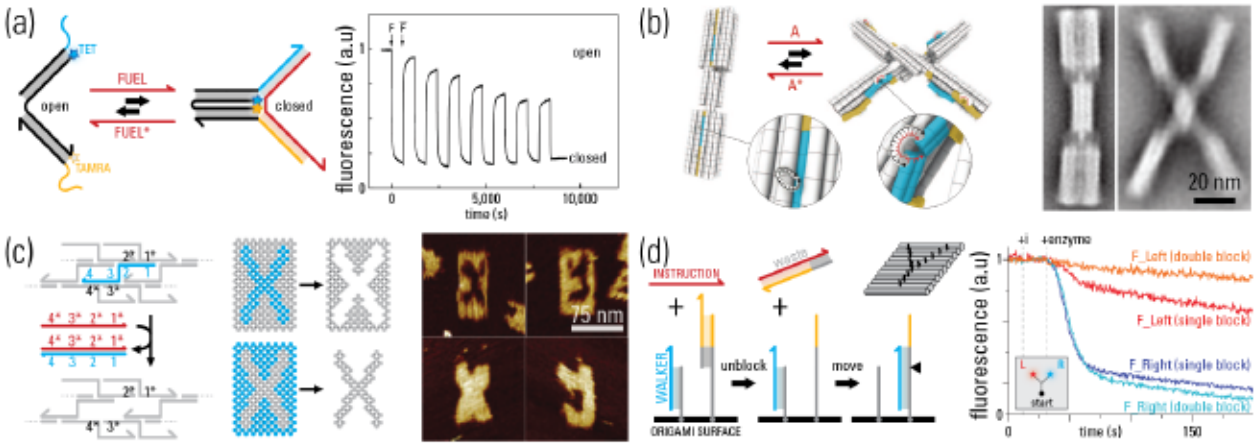

Figure 2.5. Reconfiguration of DNA nanostructures directed by DNA interface strands. In each example, the red strand of DNA is the interface strand that can facilitate communication between a circuit and a reconfiguring structure. (a) A pair of DNA tweezers (black) with two different fluorophores at the ends[91]. The tweezers are initialized in an open state where the TET fluorophore (cyan) can be excited without FRET. A fuel strand of DNA (red) binds to and constrains the tweezers into a closed state, bringing the fluorescent ends of the tweezers close together and facilitating FRET between the TET and TAMRA (yellow) fluorophores. An anti-fuel strand (fuel*) displaces the fuel strand, returning the tweezers to an open state. Fluorescence intensity of TET emission shows multiple cycles of opening and closing the tweezers. (b) A rotary DNA origami actuator[100] driven by shape-complementary blunt end stacking. Interface strand A (red) binds to complementary domains on the origami actuator to disrupt the blunt end stacking and open the actuator. TEM images show the actuator in its closed and open states. (c) Subtractive modification of DNA brick structures. Targeted strands (cyan) are removed from an initial blank canvas by corresponding subtraction interface strands (red). The addition of different sets of removal strands leads to different final shapes. AFM images show letter structures fabricated by this subtractive process[103]. (d) A DNA walker (cyan) on an origami surface (black) is blocked from traversing a target path by a blocker strand (yellow). An instruction strand (red) displaces the blocker strand from the origami, creating a pathway the walker can move across[106]. Fluorescence monitoring shows a walker moving on a track with two gated pathways, after the right-hand pathway is activated by an instruction strand. The walker transports a quencher modification to a fluorophore at the end of the right-hand pathway, decreasing the fluorescence intensity.

2.5 | DNA-directed control of other materials

DNA can also serve as a template for controlling many other molecules besides DNA. In this section we will discuss interfaces that enable strands of DNA to control the placement of nanoparticles on larger structures and surfaces, encapsulation and release of payloads from origami boxes and cages, templating of reactions between other molecules, and manipulating bulk materials such as vesicles and hydrogels.

2.5.1 | Particle placement. In principle, any component that is conjugated to a strand of DNA, or bound by an aptamer sequence, can be positioned at a specific site on other DNA nanostructures *via* hybridization. Loweth *et al* modified gold nanoparticles with DNA tags, and exploited the sequence-specific binding of DNA to assemble these gold particles into dimers and trimers[108]. Hazarika *et al* further demonstrated that nanoparticle assemblies could be reversibly disassembled with DNA strand-displacement reactions[109]. Similar techniques have been used to assemble clusters of other DNA-functionalized components, including platinum-gold nanoparticles[110], quantum dots[111], and carbon nanotubes[112]. Functionalization of cell surfaces by Staudinger ligation to phosphine-conjugated DNA strands has been used to direct the assembly of cell-cell contacts to build three-dimensional microtissues with defined interconnectivities between cells[113]. While in many of these studies the positions of DNA strands on their host particles were not specifically controlled, Kim *et al* developed a method for controlling the precise number and relative position of DNA strands on gold nanoparticles[114]. Le *et al* formed larger two-dimensional arrays of gold nanoparticles using a DNA tile lattice as a template. Arrays were formed *via* the hybridization of DNA strands conjugated to gold nanoparticles to DNA strands on the lattice[115] (Fig. 6a). A similar strategy of DNA tag hybridization to a DNA template has also been used to direct the assembly of proteins into arrays[116]. Erez *et al* used DNA tags immobilized on the surfaces of gold

electrodes to direct the attachment of a long double stranded DNA between two electrodes, separated by 12 μM. The double-stranded DNA was metallized with silver nanoparticles to create a conductive wire with differential resistances on the order of 7-30 MΩ[117]. Hazani *et al* used similar gold electrode immobilized oligonucleotides to direct the attachment of a DNA-modified carbon nanotubes between two electrodes on a gold surface[118]. Maune *et al* assembled two-dimensional cross-shaped junctions of carbon nanotubes on DNA origami, which they found exhibited field-effect transistor-like behavior[119].

Importantly, particles bound to DNA nanostructures can also move when the underlying nanostructure moves. Kuzuyk *et al* demonstrated this by positioning gold nanorods on a rotating DNA origami actuator, and used the actuator to dynamically tune the angle between the two rods. The authors used these devices as reconfigurable plasmonic metamolecules[120]. For a more detailed overview of using DNA to position other nanoparticles, see the recent review by Rogers *et al*[121].

2.5.2 | Encapsulation and drug delivery. DNA nanostructures can be used as containers for other molecules. Andersen *et al* developed a DNA origami box measuring 42 x 36 x 36 nm^3, and used strands of DNA as sequence-specific keys to open the lid[122], although encapsulation of a payload within the box was not specifically demonstrated. Grossi *et al* created a DNA origami "nanovault" that contained an enzyme within a fully enclosed cavity, held closed by double stranded DNA "locks". Addition of "key" strands of DNA complimentary to the lock domains opened the nanovault[123] (Fig. 6c). In the closed state, the enzyme was inhibited by the nanovault from interacting with substrate molecules in solution, with increased enzyme activity measured when the nanovault was in an open state. Sun *et al* used DNA origami boxes as molds for growing metal nanostructures with defined geometries. They began with a hollow DNA origami barrel, added a strand of DNA that was conjugated to a metal nanoparticle to the inside of the box to act as a seed

for growth, and then attached lids to the barrel. The metal seeds within the box were then grown in a mixture of silver nitrate and ascorbic acid until they conformed to the size and shape of their DNA barrel containers[124].

2.5.3 | Templating chemical reactions. By controlling the effective local concentrations of reactive molecules conjugated to their component strands, DNA complexes and nanostructures can regulate the rates of some chemical reactions. Calderone *et al* conjugated different reactive groups, including maleimides, aldehydes, and amines to DNA strands. These "template" strands were then mixed together in the same solution, with complementary strands of DNA binding together and initiating reactions between their attached reagent molecules. This technique allowed the authors to program different chemical reaction pathways within the same solution, which would otherwise have exhibited significant cross-reactivity between the pathways[125]. Kanan *et al* expanded on this technique for DNA-templated reaction discovery, testing 168 different possible reactions between 24 different reactants bound to template strands of DNA in a one-pot mixture[126]. Usanov *et al* further expanded this technique to produce a vast library of 256,000 different small-molecule macrocycles for insulin-degrading enzyme affinity[127].

Wilner *et al* used conjugated strands of DNA to place the enzymes glucose oxidase and horseradish peroxidase close together on a template strand of DNA, forming a biocatalytic cascade that produced ABTS•- (Fig. 6d)[128]. Ke *et al* used a DNA origami tile to assemble two different enzyme reaction pathways on the same nanostructure, using "anchor" strands of DNA on the origami surface to position the enzymes, and then used strands of DNA to reversibly control which of the two pathways was activated at any one time by blocking or unblocking the anchor points[129]. Several reviews exist that cover the topic of DNA templated reactions in further detail[130, 131].

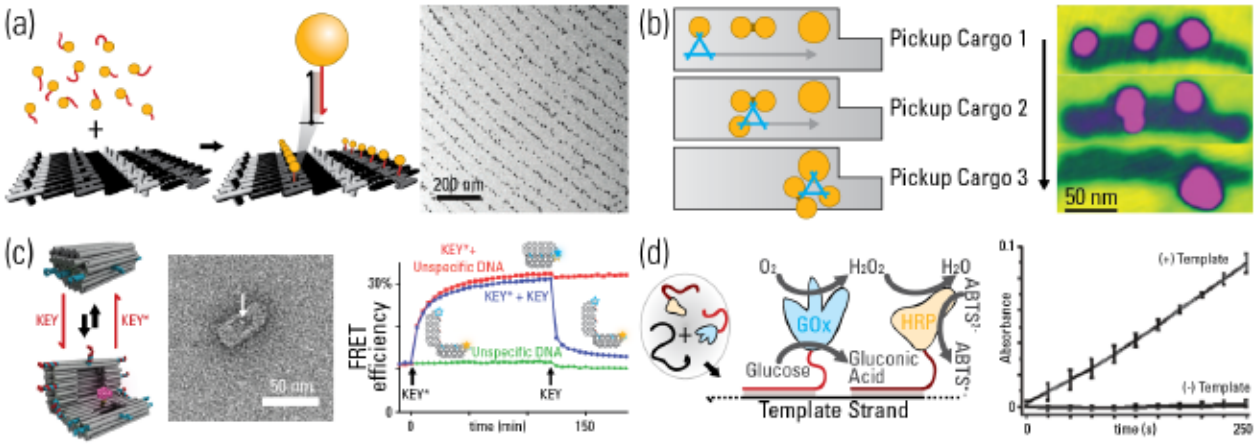

Figure 2.6. Positioning individual components. (a) Gold nanoparticles were positioned in a two-dimensional array by attaching them first to interface strands of DNA (red) that are complementary to attachment sites on a large preformed DNA scaffold (black). TEM image shows the assembled array, where the gold nanoparticles are seen as black dots[115]. (b) A DNA walker (cyan) on an origami surface (gray) picks up and colocalizes several gold nanoparticle cargos[104]. This type of cargo-manipulating walker could potentially be gated by a strand of DNA using a mechanism similar to Fig. 5d. (c) A DNA origami nanovault[123] with an enzyme (bovine alpha-chymotrypsin, pink) attached to its inner cavity. The vault is held closed by lock domains of DNA (cyan) along its outer edges. When complementary key strands of DNA (red) are introduced they displace the locks and open the nanovault, exposing the encapsulated enzyme to substrate in the external medium. FRET transfer between two different fluorophores positioned on the side of the nanovault indicates open and closed configurations of the nanovault. (d) A template strand (black+gray) assembles enzymes with DNA tags (red) into a local reaction cascade[128]. Here the cascade consists of glucose oxidase (GOx, cyan), which oxidized glucose to yield gluconic acid and H_2O_2, and horseradish peroxidase (HRP, yellow), which used the generated H_2O_2 to oxidize $ABTS^{2-}$ into $ABTS^{\bullet-}$. Absorbance readings monitor the accumulation of $ABTS^{\bullet-}$ product over time. A negative control that is missing the template strand shows no significant production.

2.5.4 | Vesicles. Lipid vesicles can enclose small volumes of liquid, and could be used to create artificial cells within which different DNA circuits operate. DNA nanostructures have been used as

templates to control the static size, shape, and dispersity of vesicles[132-136]. DNA origami nanopores have further been added to facilitate transport across membranes[137-139]. More recently, Zhang *et al* demonstrated that interface strands of DNA can change the shape of these template vesicles dynamically in a controlled manner by reconfiguring their templating nanostructures[140] (Fig. 7a). Sato *et al* used DNA interface strands to reversibly regulate the assembly of kinesin motors along the inner surfaces of vesicles to induce local curvature of the vesicle's shape[141] (Fig. 7b).

2.5.5 | DNA hydrogels and crystals. Hydrogels are materials composed of crosslinked hydrophilic polymer chains in water. By incorporating DNA into the hydrogel as crosslinks[142], the material properties of the hydrogel can be manipulated dynamically by adding DNA strands to alter, break, or create crosslinks. Lin *et al* developed DNA crosslinks that could be reversibly dissociated by adding a strand complementary to one of the crosslink strands[143]. Hydrogels with DNA crosslinks can also be stiffened or softened by adding DNA that switches the conformation of crosslinks between a double-stranded state or a partially single-stranded conformation[144] *via* DNA strand-displacement reactions. Lin *et al* reported that these DNA crosslinked hydrogels had bulk elastic moduli ranging from tens of Pascals (Pa) to about ten kPa, and that these hydrogels swelled upon softening, with volumetric swelling ratios up to about 25% between the stiff and soft states. Cangialosi *et al* used photolithographic patterning to form multi-domain DNA-crosslinked hydrogels where each domain could be swollen independently by the addition of different hairpin fuel strands of DNA to the surrounding medium; differential swelling of domains within the hydrogel causes bending or curling of the device, suggesting how such a system could be used for mechanical actuation. The authors employed a hybridization chain reaction[145] to expand the size of the crosslinks by incorporating many fuel strands of DNA into each crosslink to drive substantially higher swelling ratios than in previous work. Linear expansion ($\Delta L/L_0$) due to swelling was reported

on the order of several hundred percent[146] (Fig. 7b). DNA strand-displacement reactions have also been shown to operate within three-dimensional DNA crystals, for instance by using strands of DNA with different fluorescent modifications to change the color of a crystal[147] (Fig. 7d).

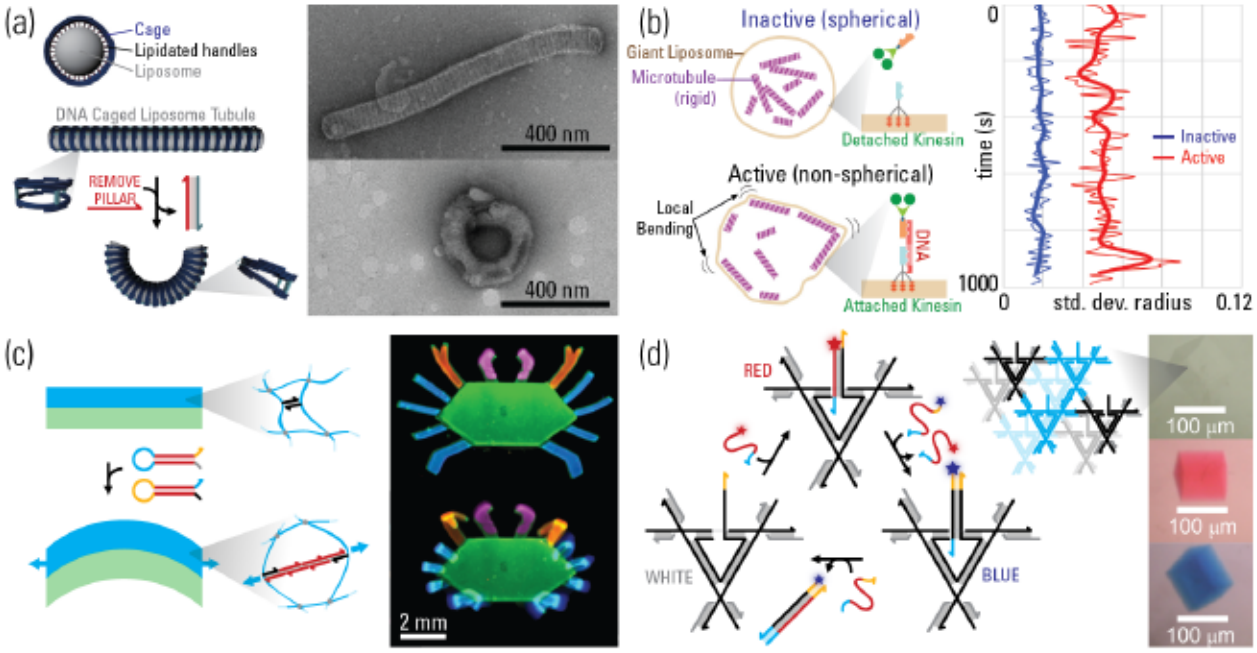

Figure 2.7. DNA-strand directed material change. (a) A DNA origami frame serves as a cage to control the size and shape of a liposome. Lipidated DNA handles immobilized on the inner surface of the origami frame serve as attachment points to position the liposome within the DNA frame. Here the DNA frame directs the formation of a lipid tubule. Reconfiguring the frame structure by removing rigid pillars *via* an interface strand of DNA (red) changes the shape of the caged liposome. Right images show electron micrographs of the caged liposome in the tubule and curled states[139]. (b) A giant liposome in which the attachment of a kinesin motor by an interface strand of DNA (red) induces lipid surface microtubule transport, which in term causes localized dynamic bending of the liposome[138]. Right, the standard deviations of the radius between the center of mass and the periphery of the liposome when the kinesin motor is inactive vs. active. (c) A DNA-crosslinked hydrogel robot with components that swell in response to interface hairpin strands of DNA (red). Selective swelling of domains can drive actuation. Fluorescence microscopy shows a DNA-crosslinked hydrogel crab in initial (top) and actuated (bottom) states[146]. (d) Strand-

displacement reactions within a three-dimensional DNA crystal[147] involving Cy3- and Cy5-modified DNA strands (red) can change what molecules are present within the crystal lattice, switching the color of the crystal between three states.

2.6 | Discussion

In this review we explored mechanisms that allow DNA circuits to sense environmental stimuli and to control the assembly or reconfiguration of downstream materials. We defined DNA circuits as black box information processing devices in which the inputs and outputs take the form of strands of DNA, which we defined as the "interface" strands. In principle, these modular interfaces allow DNA circuits to be developed independently from the material systems they sense and control, analogously to how standardized electrical interfaces allow general-purpose electronic computers to interact with a diverse set of peripheral devices. Although we focused here on DNA, a similar paradigm could apply to circuits composed of other nucleic acids, such as RNA, LNA[148], or PNA[149]. Through standardized interfaces, chemical circuits with different information processing functions could theoretically control a combinatorial variety of other materials and stimuli.

While exploring mechanisms by which interface strands of DNA could facilitate communication between materials and DNA circuits, we identified several interesting unexplored directions for potential future research. First, if mechanisms were developed to allow interface strands of DNA regulate local temperatures, DNA circuits could potentially be used generate elaborate spatiotemporal temperature gradients in three dimensions. Similarly, mechanisms that allow output strands to regulate the concentration of ions could be useful for programming fluctuations in pH in response to events detected in the environment, such as the presence of a virus or bacteria. Regarding bulk materials, mechanisms that allow interface strands to regulate flux across membranes could facilitate communication between networks of isolated DNA circuits within neighboring vesicles, while mechanisms to initiate division of vesicles or hydrogels could be useful for creating

synthetic materials that can replicate and evolve. While there is a growing literature on using DNA interface strands to interact with soft macroscopic materials, interface mechanisms that allow DNA to regulate biomineralization processes could extend this control to harder or stiffer materials that have structural integrity at larger scales and under significant mechanical stresses. Finally, mechanisms that allow DNA circuits to dynamically control the state and position of DNA origami lithography masks[150,151], potentially by reconfiguring origami to fit shape-dependent surface attachment sites[152,153], could assist in the fabrication of solid state electronics and lead to DNA computers that could create electronic computers, which could create additional DNA computers, *et cetera*.

Acknowledgements

The authors thank Phillip Dorsey, Joshua Fern, Naresh Niranjan, Misha Rubanov, Samuel Schaffter, and Wenlu Wang for insightful conversations. This work was supported by grant DE-SC0010595.

References

1 Isaacson, W. (2014). The innovators: How a group of inventors, hackers, geniuses and geeks created the digital revolution. Simon and Schuster.
2 De Silva, A. P., & Uchiyama, S. (2007). Molecular logic and computing. Nature Nanotechnology, 2(7), 399.
3 Zhang, D. Y., & Seelig, G. (2011). Dynamic DNA nanotechnology using strand-displacement reactions. Nature chemistry, 3(2), 103.
4 Baccouche, A., Montagne, K., Padirac, A., Fujii, T., & Rondelez, Y. (2014). Dynamic DNA-toolbox reaction circuits: a walkthrough. Methods, 67(2), 234-249.
5 Willner, I., Shlyahovsky, B., Zayats, M., & Willner, B. (2008). DNAzymes for sensing, nanobiotechnology and logic gate applications. Chemical Society Reviews, 37(6), 1153-1165.
6 Seelig, G., Soloveichik, D., Zhang, D. Y., & Winfree, E. (2006). Enzyme-free nucleic acid logic circuits. science, 314(5805), 1585-1588.
7 Qian, L., & Winfree, E. (2011). Scaling up digital circuit computation with DNA strand displacement cascades. Science, 332(6034), 1196-1201.
8 Genot, A. J., Bath, J., & Turberfield, A. J. (2011). Reversible logic circuits made of DNA. Journal of the American Chemical Society, 133(50), 20080-20083.
9 Zadorin, A. S., Rondelez, Y., Galas, J. C., & Estevez-Torres, A. (2015). Synthesis of programmable reaction-diffusion fronts using DNA catalyzers. Physical review letters, 114(6), 068301.
10 Franco, E., Friedrichs, E., Kim, J., Jungmann, R., Murray, R., Winfree, E., & Simmel, F. C. (2011). Timing molecular motion and production with a synthetic transcriptional clock. Proceedings of the National Academy of Sciences.
11 Fujii, T., & Rondelez, Y. (2012). Predator–prey molecular ecosystems. ACS nano, 7(1), 27-34.

12 Srinivas, N., Parkin, J., Seelig, G., Winfree, E., & Soloveichik, D. (2017). Enzyme-free nucleic acid dynamical systems. Science, 358(6369), eaal2052.
13 Fern, J., Scalise, D., Cangialosi, A., Howie, D., Potters, L., & Schulman, R. (2016). DNA strand-displacement timer circuits. ACS synthetic biology, 6(2), 190-193.
14 Stojanovic, M. N., & Stefanovic, D. (2003). A deoxyribozyme-based molecular automaton. Nature biotechnology, 21(9), 1069.
15 Franco, E., Del Vecchio, D., & Murray, R. M. (2009, December). Design of insulating devices for in vitro synthetic circuits. In Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on (pp. 4584-4589). IEEE.
16 Jayanthi, S., Nilgiriwala, K. S., & Del Vecchio, D. (2013). Retroactivity controls the temporal dynamics of gene transcription. ACS synthetic biology, 2(8), 431-441.
17 Qian, L., Winfree, E., & Bruck, J. (2011). Neural network computation with DNA strand displacement cascades. Nature, 475(7356), 368.
18 Cherry, K. M., & Qian, L. (2018). Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. Nature, 1.
19 Zhang, D. Y., & Seelig, G. (2010, June). DNA-based fixed gain amplifiers and linear classifier circuits. In International Workshop on DNA-Based Computers (pp. 176-186). Springer, Berlin, Heidelberg.
20 Chen, X., Briggs, N., McLain, J. R., & Ellington, A. D. (2013). Stacking nonenzymatic circuits for high signal gain. Proceedings of the National Academy of Sciences, 201222807.
21 Chirieleison, S. M., Allen, P. B., Simpson, Z. B., Ellington, A. D., & Chen, X. (2013). Pattern transformation with DNA circuits. Nature chemistry, 5(12), 1000.
22 Zenk, J., Scalise, D., Wang, K., Dorsey, P., Fern, J., Cruz, A., & Schulman, R. (2017). Stable DNA-based reaction–diffusion patterns. RSC Advances, 7(29), 18032-18040.
23 Thachuk, C., Winfree, E., & Soloveichik, D. (2015, August). Leakless DNA strand displacement systems. In International Workshop on DNA-Based Computers (pp. 133-153). Springer, Cham.
24 Huizenga, D. E., & Szostak, J. W. (1995). A DNA aptamer that binds adenosine and ATP. Biochemistry, 34(2), 656-665.
25 He, J. L., Wu, Z. S., Zhou, H., Wang, H. Q., Jiang, J. H., Shen, G. L., & Yu, R. Q. (2010). Fluorescence aptameric sensor for strand displacement amplification detection of cocaine. Analytical chemistry, 82(4), 1358-1364.
26 Zhou, W., Saran, R., & Liu, J. (2017). Metal sensing by DNA. Chemical reviews, 117(12), 8272-8325.
27 Pendergrast, P. S., Marsh, H. N., Grate, D., Healy, J. M., & Stanton, M. (2005). Nucleic acid aptamers for target validation and therapeutic applications. Journal of biomolecular techniques: JBT, 16(3), 224.
28 Tombelli, S., Minunni, M., & Mascini, M. (2005). Analytical applications of aptamers. Biosensors and Bioelectronics, 20(12), 2424-2434.
29 Bunka, D. H., Platonova, O., & Stockley, P. G. (2010). Development of aptamer therapeutics. Current opinion in pharmacology, 10(5), 557-562.
30 Ellington, A. D., & Szostak, J. W. (1990). In vitro selection of RNA molecules that bind specific ligands. nature, 346(6287), 818.
31 Tuerk, C., & Gold, L. (1990). Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. science, 249(4968), 505-510.
32 Li, F., Zhang, H., Wang, Z., Li, X., Li, X. F., & Le, X. C. (2013). Dynamic DNA assemblies mediated by binding-induced DNA strand displacement. Journal of the American Chemical Society, 135(7), 2443-2446.
33 Picuri, J. M., Frezza, B. M., & Ghadiri, M. R. (2009). Universal translators for nucleic acid diagnosis. Journal of the American Chemical Society, 131(26), 9368-9377.
34 Zhu, J., Zhang, L., Zhou, Z., Dong, S., & Wang, E. (2013). Aptamer-based sensing platform using three-way DNA junction-driven strand displacement and its application in DNA logic circuit. Analytical chemistry, 86(1), 312-316.
35 Niu, S., Qu, L., Zhang, Q., & Lin, J. (2012). Fluorescence detection of thrombin using autocatalytic strand displacement cycle reaction and a dual-aptamer DNA sandwich assay. Analytical biochemistry, 421(2), 362-367.
36 Xing, Y., Yang, Z., & Liu, D. (2011). A responsive hidden toehold to enable controllable DNA strand displacement reactions. Angewandte Chemie International Edition, 50(50), 11934-11936.
37 Wang, H., Zheng, J., Sun, Y., & Li, T. (2018). Cellular environment-responsive intelligent DNA logic circuits for controllable molecular sensing. Biosensors and Bioelectronics.
38 Bhadra, S., & Ellington, A. D. (2014). A Spinach molecular beacon triggered by strand displacement. Rna, 20(8), 1183-1194.
39 Lloyd, J., Tran, C. H., Wadhwani, K., Cuba Samaniego, C., Subramanian, H. K., & Franco, E. (2017). Dynamic Control of Aptamer–Ligand Activity Using Strand Displacement Reactions. ACS synthetic biology, 7(1), 30-37.

40 Zhou, C., Yang, Z., & Liu, D. (2012). Reversible regulation of protein binding affinity by a DNA machine. Journal of the American Chemical Society, 134(3), 1416-1418.
41 Liu, Z., & Mao, C. (2014). Reporting transient molecular events by DNA strand displacement. Chemical Communications, 50(60), 8239-8241.
42 Amodio, A., Zhao, B., Porchetta, A., Idili, A., Castronovo, M., Fan, C., & Ricci, F. (2014). Rational design of pH-controlled DNA strand displacement. Journal of the American Chemical Society, 136(47), 16469-16472
43 Tang, W., Wang, H., Wang, D., Zhao, Y., Li, N., & Liu, F. (2013). DNA tetraplexes-based toehold activation for controllable DNA strand displacement reactions. Journal of the American Chemical Society, 135(37), 13628-13631.
44 Ding, W., Deng, W., Zhu, H., & Liang, H. (2013). Metallo-toeholds: controlling DNA strand displacement driven by Hg (ii) ions. Chemical Communications, 49(85), 9953-9955.
45 Ranallo, S., Amodio, A., Idili, A., Porchetta, A., & Ricci, F. (2016). Electronic control of DNA-based nanoswitches and nanodevices. Chemical science, 7(1), 66-71.
46 Frasconi, M., Tel-Vered, R., Elbaz, J., & Willner, I. (2010). Electrochemically stimulated pH changes: a route to control chemical reactivity. Journal of the American Chemical Society, 132(6), 2029-2036.
47 Freage, L., Trifonov, A., Tel-Vered, R., Golub, E., Wang, F., McCaskill, J. S., & Willner, I. (2015). Addressing, amplifying and switching DNAzyme functions by electrochemically-triggered release of metal ions. Chemical science, 6(6), 3544-3549.
48 Jeong, H., Ranallo, S., Rossetti, M., Heo, J., Shin, J., Park, K., ... & Hong, J. (2016). Electronic activation of a DNA nanodevice using a multilayer nanofilm. Small, 12(40), 5572-5578.
49 Fan, C., Plaxco, K. W., & Heeger, A. J. (2003). Electrochemical interrogation of conformational changes as a reagentless method for the sequence-specific detection of DNA. Proceedings of the National Academy of Sciences, 100(16), 9134-9137.
50 Immoos, C. E., Lee, S. J., & Grinstaff, M. W. (2004). Conformationally gated electrochemical gene detection. ChemBioChem, 5(8), 1100-1103.
51 Xiao, Y., Lubin, A. A., Baker, B. R., Plaxco, K. W., & Heeger, A. J. (2006). Single-step electronic detection of femtomolar DNA by target-induced strand displacement in an electrode-bound duplex. Proceedings of the National Academy of Sciences, 103(45), 16677-16680.
52 Xiao, Y., Lai, R. Y., & Plaxco, K. W. (2007). Preparation of electrode-immobilized, redox-modified oligonucleotides for electrochemical DNA and aptamer-based sensing. Nature protocols, 2(11), 2875.
53 Immoos, C. E., Lee, S. J., & Grinstaff, M. W. (2004). DNA-PEG-DNA triblock macromolecules for reagentless DNA detection. Journal of the American Chemical Society, 126(35), 10814-10815.
54 Drummond, T. G., Hill, M. G., & Barton, J. K. (2003). Electrochemical DNA sensors. Nature biotechnology, 21(10), 1192.
55 Jonstrup, A. T., Fredsøe, J., & Andersen, A. H. (2013). DNA hairpins as temperature switches, thermometers and ionic detectors. Sensors, 13(5), 5937-5944.
56 Gareau, D., Desrosiers, A., & Vallée-Bélisle, A. (2016). Programmable quantitative DNA nanothermometers. Nano letters, 16(7), 3976-3981.
57 Gehrels, E. W., Rogers, W., & Manoharan, V. N. (2018). Using DNA strand displacement to control interactions in DNA-grafted colloids. Soft matter.
58 Rogers, W. Benjamin, and Vinothan N. Manoharan. "Programming colloidal phase transitions with DNA strand displacement." Science 347.6222 (2015): 639-642.
59 Govorov, A. O., Zhang, W., Skeini, T., Richardson, H., Lee, J., & Kotov, N. A. (2006). Gold nanoparticle ensembles as heaters and actuators: melting and collective plasmon resonances. Nanoscale Research Letters, 1(1), 84.
60 Prokup, A., Hemphill, J., & Deiters, A. (2012). DNA computation: a photochemically controlled AND gate. Journal of the American Chemical Society, 134(8), 3810-3815.
61 Huang, F., You, M., Han, D., Xiong, X., Liang, H., & Tan, W. (2013). DNA branch migration reactions through photocontrollable toehold formation. Journal of the American Chemical Society, 135(21), 7967-7973.
62 Kou, B., Zhang, J., Huai, X., Liang, X., & Xiao, S. J. (2015). Light-driven reversible strand displacement using glycerol azobenzene inserted DNA. Rsc Advances, 5(7), 5055-5058.
63 Nakamura, S., Hashimoto, H., Kobayashi, S., & Fujimoto, K. (2017). Photochemical Acceleration of DNA Strand Displacement by Using Ultrafast DNA Photo-crosslinking. ChemBioChem, 18(20), 1984-1989.
64 Zhang, P., Beck, T., & Tan, W. (2001). Design of a molecular beacon DNA probe with two fluorophores. Angewandte Chemie, 113(2), 416-419.
65 Rothemund, P. W. (2006). Folding DNA to create nanoscale shapes and patterns. Nature, 440(7082), 297.
66 Ke, Y., Ong, L. L., Shih, W. M., & Yin, P. (2012). Three-dimensional structures self-assembled from DNA bricks. science, 338(6111), 1177-1183.

67 Nummelin, S., Kommeri, J., Kostiainen, M. A., & Linko, V. (2018). Evolution of structural DNA nanotechnology. Advanced Materials, 1703721.
68 Tørring, T., Voigt, N. V., Nangreave, J., Yan, H., & Gothelf, K. V. (2011). DNA origami: a quantum leap for self-assembly of complex structures. Chemical Society Reviews, 40(12), 5636-5646.
69 Wagenbauer, K. F., Sigl, C., & Dietz, H. (2017). Gigadalton-scale shape-programmable DNA assemblies. Nature, 552(7683), 78.
70 Zheng, J., Birktoft, J. J., Chen, Y., Wang, T., Sha, R., Constantinou, P. E., ... & Seeman, N. C. (2009). From molecular to macroscopic via the rational design of a self-assembled 3D DNA crystal. Nature, 461(7260), 74.
71 Schulman, R., & Winfree, E. (2007). Synthesis of crystals with a programmable kinetic barrier to nucleation. Proceedings of the National Academy of Sciences, 104(39), 15236-15241.
72 Fu, T. J., & Seeman, N. C. (1993). DNA double-crossover molecules. Biochemistry, 32(13), 3211-3220.
73 Winfree, E., Liu, F., Wenzler, L. A., & Seeman, N. C. (1998). Design and self-assembly of two-dimensional DNA crystals. Nature, 394(6693), 539.
74 Lo, P. K., Karam, P., Aldaye, F. A., McLaughlin, C. K., Hamblin, G. D., Cosa, G., & Sleiman, H. F. (2010). Loading and selective release of cargo in DNA nanotubes with longitudinal variation. Nature chemistry, 2(4), 319.
75 Jungmann, R., Liedl, T., Sobey, T. L., Shih, W., & Simmel, F. C. (2008). Isothermal assembly of DNA origami structures using denaturing agents. Journal of the American Chemical Society, 130(31), 10062-10063.
76 Zhang, D. Y., Hariadi, R. F., Choi, H. M., & Winfree, E. (2013). Integrating DNA strand-displacement circuitry with DNA tile self-assembly. Nature communications, 4, 1965.
77 Amodio, A., Adedeji, A. F., Castronovo, M., Franco, E., & Ricci, F. (2016). pH-controlled assembly of DNA tiles. Journal of the American Chemical Society, 138(39), 12735-12738.
78 Padilla, J. E., Sha, R., Kristiansen, M., Chen, J., Jonoska, N., & Seeman, N. C. (2015). A Signal-Passing DNA-Strand-Exchange Mechanism for Active Self-Assembly of DNA Nanostructures. Angewandte Chemie International Edition, 54(20), 5939-5942.
79 Zhang, Y., McMullen, A., Pontani, L. L., He, X., Sha, R., Seeman, N. C., ... & Chaikin, P. M. (2017). Sequential self-assembly of DNA functionalized droplets. Nature Communications, 8(1), 21.
80 Wang, J., Zhou, Z., Yue, L., Wang, S., & Willner, I. (2018). Switchable Triggered Interconversion and Reconfiguration of DNA Origami Dimers and Their Use for Programmed Catalysis. Nano letters, 18(4), 2718-2724.
81 Peng, R., Wang, H., Lyu, Y., Xu, L., Liu, H., Kuai, H., ... & Tan, W. (2017). Facile Assembly/Disassembly of DNA Nanostructures Anchored on Cell-Mimicking Giant Vesicles. Journal of the American Chemical Society, 139(36), 12410-12413
82 Cao, Y. C., Jin, R., Thaxton, C. S., & Mirkin, C. A. (2005). A two-color-change, nanoparticle-based method for DNA detection. Talanta, 67(3), 449-455.
83 Yao, D., Song, T., Sun, X., Xiao, S., Huang, F., & Liang, H. (2015). Integrating DNA-strand-displacement circuitry with self-assembly of spherical nucleic acids. Journal of the American Chemical Society, 137(44), 14107-14113.
84 Yao, D., Wang, B., Xiao, S., Song, T., Huang, F., & Liang, H. (2015). What Controls the "Off/On Switch" in the Toehold-Mediated Strand Displacement Reaction on DNA Conjugated Gold Nanoparticles?. Langmuir, 31(25), 7055-7061.
85 Thaxton, C. S., Georganopoulou, D. G., & Mirkin, C. A. (2006). Gold nanoparticle probes for the detection of nucleic acid targets. Clinica Chimica Acta, 363(1-2), 120-126.
86 Cutler, J. I., Auyeung, E., & Mirkin, C. A. (2012). Spherical nucleic acids. Journal of the American Chemical Society, 134(3), 1376-1391.
87 Yin, P., Choi, H. M., Calvert, C. R., & Pierce, N. A. (2008). Programming biomolecular self-assembly pathways. Nature, 451(7176), 318.
88 Sadowski, J. P., Calvert, C. R., Zhang, D. Y., Pierce, N. A., & Yin, P. (2014). Developmental self-assembly of a DNA tetrahedron. ACS nano, 8(4), 3251-3259.
89 Zhang, Zhuo, Tsz Wing Fan, and I-Ming Hsing. "Integrating DNA strand displacement circuitry to the nonlinear hybridization chain reaction." Nanoscale 9.8 (2017): 2748-2754.
90 Kishi, J. Y., Schaus, T. E., Gopalkrishnan, N., Xuan, F., & Yin, P. (2018). Programmable autonomous synthesis of single-stranded DNA. Nature chemistry, 10(2), 155.
91 Yurke, B., Turberfield, A. J., Mills Jr, A. P., Simmel, F. C., & Neumann, J. L. (2000). A DNA-fuelled molecular machine made of DNA. Nature, 406(6796), 605.
92 Simmel, Friedrich C., and Bernard Yurke. "Using DNA to construct and power a nanoactuator." Physical Review E 63.4 (2001): 041913.
93 Feng, L., Park, S. H., Reif, J. H., & Yan, H. (2003). A Two-State DNA Lattice Switched by DNA Nanoactuator. Angewandte Chemie, 115(36), 4478-4482.

94 Goodman, R. P., Heilemann, M., Doose, S., Erben, C. M., Kapanidis, A. N., & Turberfield, A. J. (2008). Reconfigurable, braced, three-dimensional DNA nanostructures. Nature nanotechnology, 3(2), 93.
95 Saccà, B., Ishitsuka, Y., Meyer, R., Sprengel, A., Schöneweiß, E. C., Nienhaus, G. U., & Niemeyer, C. M. (2015). Reversible Reconfiguration of DNA Origami Nanochambers Monitored by Single-Molecule FRET. Angewandte Chemie International Edition, 54(12), 3592-3597.
96 Chen, H., Weng, T. W., Riccitelli, M. M., Cui, Y., Irudayaraj, J., & Choi, J. H. (2014). Understanding the mechanical properties of DNA origami tiles and controlling the kinetics of their folding and unfolding reconfiguration. Journal of the American Chemical Society, 136(19), 6995-7005.
97 Yan, H., Zhang, X., Shen, Z., & Seeman, N. C. (2002). A robust DNA mechanical device controlled by hybridization topology. Nature, 415(6867), 62.
98 Ding, B., & Seeman, N. C. (2006). Operation of a DNA robot arm inserted into a 2D DNA crystalline substrate. Science, 314(5805), 1583-1585.
99 Marras, A. E., Zhou, L., Su, H. J., & Castro, C. E. (2015). Programmable motion of DNA origami mechanisms. Proceedings of the National Academy of Sciences, 112(3), 713-718.
100 Gerling, T., Wagenbauer, K. F., Neuner, A. M., & Dietz, H. (2015). Dynamic DNA devices and assemblies formed by shape-complementary, non–base pairing 3D components. Science, 347(6229), 1446-1452.
101 Han, D., Pal, S., Liu, Y., & Yan, H. (2010). Folding and cutting DNA into reconfigurable topological nanostructures. Nature nanotechnology, 5(10), 712.
102 Zhang, F., Nangreave, J., Liu, Y., & Yan, H. (2012). Reconfigurable DNA origami to generate quasifractal patterns. Nano letters, 12(6), 3290-3295.
103 Wei, Bryan, et al. "Complex reconfiguration of DNA nanostructures." Angewandte Chemie 126.29 (2014): 7605-7609.
104 Gu, H., Chao, J., Xiao, S. J., & Seeman, N. C. (2010). A proximity-based programmable DNA nanoscale assembly line. Nature, 465(7295), 202.
105 Thubagere, A. J., Li, W., Johnson, R. F., Chen, Z., Doroudi, S., Lee, Y. L., ... & Winfree, E. (2017). A cargo-sorting DNA robot. Science, 357(6356), eaan6558.
106 Wickham, S. F., Bath, J., Katsuda, Y., Endo, M., Hidaka, K., Sugiyama, H., & Turberfield, A. J. (2012). A DNA-based molecular motor that can navigate a network of tracks. Nature nanotechnology, 7(3), 169.
107 Pan, J., Li, F., Cha, T. G., Chen, H., & Choi, J. H. (2015). Recent progress on DNA based walkers. Current opinion in biotechnology, 34, 56-64.
108 Loweth, C. J., Caldwell, W. B., Peng, X., Alivisatos, A. P., & Schultz, P. G. (1999). DNA-based assembly of gold nanocrystals. Angewandte Chemie International Edition, 38(12), 1808-1812.
109 Hazarika, P., Ceyhan, B., & Niemeyer, C. M. (2004). Reversible switching of DNA–gold nanoparticle aggregation. Angewandte Chemie, 116(47), 6631-6633.
110 Li, Y., Zheng, Y., Gong, M., & Deng, Z. (2012). Pt nanoparticles decorated with a discrete number of DNA molecules for programmable assembly of Au–Pt bimetallic superstructures. Chemical Communications, 48(31), 3727-3729.
111 He, X., Li, Z., Chen, M., & Ma, N. (2014). DNA-Programmed Dynamic Assembly of Quantum Dots for Molecular Computation. Angewandte Chemie International Edition, 53(52), 14447-14450.
112 CNTs Li, Y., Han, X., & Deng, Z. (2007). Grafting Single-Walled Carbon Nanotubes with Highly Hybridizable DNA Sequences: Potential Building Blocks for DNA-Programmed Material Assembly. Angewandte Chemie International Edition, 46(39), 7481-7484.
113 Gartner, Z. J., & Bertozzi, C. R. (2009). Programmed assembly of 3-dimensional microtissues with defined cellular connectivity. Proceedings of the National Academy of Sciences, 106(12), 4606-4610.
114 Kim, J. W., Kim, J. H., & Deaton, R. (2011). DNA-Linked Nanoparticle Building Blocks for Programmable Matter. Angewandte Chemie International Edition, 50(39), 9185-9190.
115 Le, J. D., Pinto, Y., Seeman, N. C., Musier-Forsyth, K., Taton, T. A., & Kiehl, R. A. (2004). DNA-templated self-assembly of metallic nanocomponent arrays on a surface. Nano Letters, 4(12), 2343-2347.
116 Liu, Y., Lin, C., Li, H., & Yan, H. (2005). Aptamer-directed self-assembly of protein arrays on a DNA nanostructure. Angewandte Chemie, 117(28), 4407-4412.
117 Braun, E., Eichen, Y., Sivan, U., & Ben-Yoseph, G. (1998). DNA-templated assembly and electrode attachment of a conducting silver wire. Nature, 391(6669), 775.
118 Maune, H. T., Han, S. P., Barish, R. D., Bockrath, M., Goddard III, W. A., Rothemund, P. W., & Winfree, E. (2010). Self-assembly of carbon nanotubes into two-dimensional geometries using DNA origami templates. Nature nanotechnology, 5(1), 61.
119 Hazani, M., Hennrich, F., Kappes, M., Naaman, R., Peled, D., Sidorov, V., & Shvarts, D. (2004). DNA-mediated self-assembly of carbon nanotube-based electronic devices. Chemical Physics Letters, 391(4-6), 389-392.

120 Kuzyk, A., Schreiber, R., Zhang, H., Govorov, A. O., Liedl, T., & Liu, N. (2014). Reconfigurable 3D plasmonic metamolecules. Nature materials, 13(9), 862.
121 Rogers, W. B., Shih, W. M., & Manoharan, V. N. (2016). Using DNA to program the self-assembly of colloidal nanoparticles and microparticles. Nature Reviews Materials, 1(3), 16008.
122 Andersen, E. S., Dong, M., Nielsen, M. M., Jahn, K., Subramani, R., Mamdouh, W., ... & Pedersen, J. S. (2009). Self-assembly of a nanoscale DNA box with a controllable lid. Nature, 459(7243), 73.
123 Grossi, G., Jepsen, M. D. E., Kjems, J., & Andersen, E. S. (2017). Control of enzyme reactions by a reconfigurable DNA nanovault. Nature communications, 8(1), 992.
124 Sun, W., Boulais, E., Hakobyan, Y., Wang, W. L., Guan, A., Bathe, M., & Yin, P. (2014). Casting inorganic structures with DNA molds. Science, 346(6210), 1258361.
125 Calderone, C. T., Puckett, J. W., Gartner, Z. J., & Liu, D. R. (2002). Directing otherwise incompatible reactions in a single solution by using DNA-templated organic synthesis. Angewandte Chemie International Edition, 41(21), 4104-4108.
126 Kanan, M. W., Rozenman, M. M., Sakurai, K., Snyder, T. M., & Liu, D. R. (2004). Reaction discovery enabled by DNA-templated synthesis and in vitro selection. Nature, 431(7008), 545.
127 Usanov, D. L., Chan, A. I., Maianti, J. P., & Liu, D. R. (2018). Second-generation DNA-templated macrocycle libraries for the discovery of bioactive small molecules. Nature chemistry, 1.
128 Wilner, O. I., Shimron, S., Weizmann, Y., Wang, Z. G., & Willner, I. (2009). Self-assembly of enzymes on DNA scaffolds: en route to biocatalytic cascades and the synthesis of metallic nanowires. Nano letters, 9(5), 2040-2043.
129 Ke, G., Liu, M., Jiang, S., Qi, X., Yang, Y. R., Wootten, S., ... & Yan, H. (2016). Directional regulation of enzyme pathways through the control of substrate channeling on a DNA origami scaffold. Angewandte Chemie International Edition, 55(26), 7483-7486.
130 Li, X., & Liu, D. R. (2004). DNA-templated organic synthesis: Nature's strategy for controlling chemical reactivity applied to synthetic molecules. Angewandte Chemie International Edition, 43(37), 4848-4870.
131 Rajendran, A., Nakata, E., Nakano, S., & Morii, T. (2017). Nucleic-Acid-Templated Enzyme Cascades. ChemBioChem, 18(8), 696-716.
132 Perrault, S. D., & Shih, W. M. (2014). Virus-inspired membrane encapsulation of DNA nanostructures to achieve in vivo stability. ACS nano, 8(5), 5132-5140.
133 Dong, Y., Sun, Y., Wang, L., Wang, D., Zhou, T., Yang, Z., ... & Liu, D. (2014). Frame-guided assembly of vesicles with programmed geometry and dimensions. Angewandte Chemie, 126(10), 2645-2648.
134 Dong, Y., Yang, Z., & Liu, D. (2015). Using Small Molecules to Prepare Vesicles with Designable Shapes and Sizes via Frame-Guided Assembly Strategy. Small, 11(31), 3768-3771.
135 Dong, Y., Yang, Y. R., Zhang, Y., Wang, D., Wei, X., Banerjee, S., ... & Liu, D. (2017). Cuboid Vesicles Formed by Frame-Guided Assembly on DNA Origami Scaffolds. Angewandte Chemie International Edition, 56(6), 1586-1589.
136 Yang, Y., Wang, J., Shigematsu, H., Xu, W., Shih, W. M., Rothman, J. E., & Lin, C. (2016). Self-assembly of size-controlled liposomes on DNA nanotemplates. Nature chemistry, 8(5), 476.
137 Langecker, M., Arnaut, V., Martin, T. G., List, J., Renner, S., Mayer, M., ... & Simmel, F. C. (2012). Synthetic lipid membrane channels formed by designed DNA nanostructures. Science, 338(6109), 932-936.
138 Krishnan, S., Ziegler, D., Arnaut, V., Martin, T. G., Kapsner, K., Henneberg, K., ... & Simmel, F. C. (2016). Molecular transport through large-diameter DNA nanopores. Nature communications, 7, 12787.
139 Langecker, M., Arnaut, V., List, J., & Simmel, F. C. (2014). DNA nanostructures interacting with lipid bilayer membranes. Accounts of chemical research, 47(6), 1807-1815.
140 Zhang, Z., Yang, Y., Pincet, F., Llaguno, M. C., & Lin, C. (2017). Placing and shaping liposomes with reconfigurable DNA nanocages. Nature chemistry, 9(7), 653.
141 Sato, Y., Hiratsuka, Y., Kawamata, I., Murata, S., & Shin-ichiro, M. N. (2017). Micrometer-sized molecular robot changes its shape in response to signal molecules. Science Robotics, 2(4), eaal3735.
142 Nagahara, S., & Matsuda, T. (1996). Hydrogel formation via hybridization of oligonucleotides derivatized in water-soluble vinyl polymers. Polymer Gels and Networks, 4(2), 111-127.
143 Lin, D. C., Yurke, B., & Langrana, N. A. (2004). Mechanical properties of a reversible, DNA-crosslinked polyacrylamide hydrogel. Journal of biomechanical engineering, 126(1), 104-110.
144 Lin, D. C., Yurke, B., & Langrana, N. A. (2005). Inducing reversible stiffness changes in DNA-crosslinked gels. Journal of Materials Research, 20(6), 1456-1464.
145 Venkataraman, S., Dirks, R. M., Rothemund, P. W., Winfree, E., & Pierce, N. A. (2007). An autonomous polymerization motor powered by DNA hybridization. Nature Nanotechnology, 2(8), 490.
146 Cangialosi, A., Yoon, C., Liu, J., Huang, Q., Guo, J., Nguyen, T. D., ... & Schulman, R. (2017). DNA sequence–directed shape change of photopatterned hydrogels via high-degree swelling. Science, 357(6356), 1126-1130.

147 Hao, Y., Kristiansen, M., Sha, R., Birktoft, J. J., Hernandez, C., Mao, C., & Seeman, N. C. (2017). A device that operates within a self-assembled 3D DNA crystal. Nature chemistry, 9(8), 824.
148 Olson, X., Kotani, S., Yurke, B., Graugnard, E., & Hughes, W. L. (2017). Kinetics of DNA Strand Displacement Systems with Locked Nucleic Acids. The Journal of Physical Chemistry B, 121(12), 2594-2602.
149 Kabza, A. M., Young, B. E., & Sczepanski, J. T. (2017). Heterochiral DNA Strand-Displacement Circuits. Journal of the American Chemical Society, 139(49), 17715-17718.
150 Diagne, C. T., Brun, C., Gasparutto, D., Baillin, X., & Tiron, R. (2016). DNA origami mask for sub-ten-nanometer lithography. ACS nano, 10(7), 6458-6463.
151 Surwade, S. P., Zhou, F., Wei, B., Sun, W., Powell, A., O'Donnell, C., ... & Liu, H. (2013). Nanoscale growth and patterning of inorganic oxides using DNA nanostructure templates. Journal of the American Chemical Society, 135(18), 6778-6781.
152 Kershner, R. J., Bozano, L. D., Micheel, C. M., Hung, A. M., Fornof, A. R., Cha, J. N., ... & Wallraff, G. M. (2009). Placement and orientation of individual DNA shapes on lithographically patterned surfaces. Nature Nanotechnology, 4(9), 557.
153 Gopinath, A., Miyazono, E., Faraon, A., & Rothemund, P. W. (2016). Engineering and mapping nanocavity emission via precision placement of DNA origami. Nature, 535(7612), 401.

3 | Designing Modular Reaction-Diffusion Programs for Complex Pattern Formation

Summary. Cells use sophisticated, multiscale spatial patterns of chemical instructions to control cell fate and tissue growth. While some types of synthetic pattern formation have been well studied[1-6], it remains unclear how to design chemical processes that can reproducibly create similar spatial patterns. Here we describe a scalable approach for the design of processes that generate such patterns, which can be implemented using synthetic DNA reaction-diffusion networks[7,8]. In our method, black-box modules are connected together into integrated programs for arbitrarily complex pattern formation. These programs can respond to input stimuli, process information, and ultimately produce stable output patterns that differ in size and concentration from their inputs. To build these programs, we break a target pattern into a set of patterning subtasks, design modules to perform these subtasks independently, and combine the modules into networks. We demonstrate in simulation how programs designed with our methodology can generate complex patterns, including a French flag and a stick figure.

Current methods for controlling spatial variation in the concentration of biomolecules, including microcontact printing[9] and composite hydrogel assembly[10], can produce either patterns of immobilized molecules whose activity is limited, or spatial patterns that are not self-sustaining, dynamic or capable of responding to environmental signals. In vivo, regions of high and low concentrations of diffusing molecules arise because of the controlled interplay of molecular diffusion and chemical reactions. The resulting two- and three-dimensional patterns can remain stable over long time periods or evolve in a controlled fashion, and can form in response to a diverse array of chemical and physical stimuli[11]. Here we suggest a method to design reaction-diffusion systems to produce similarly well-defined, complex and programmable concentration

patterns in vitro. This method could generate spatial concentration patterns that cannot be produced using existing design methods, including stable patterns of diffusing molecules, three-dimensional patterns and dynamic patterns that programmably form in response to chemical signals in the environment.

3.1 | Introduction

During *Drosophila* embryogenesis, initially identical segments of a fruit fly embryo autonomously obtain their identities (*i.e.* head, thoracic, abdominal and terminal) from the different local concentrations of mRNA at different locations within the embryo[11]. Patterns of chemical concentrations are also thought to play important roles in mitotic spindle formation[12] and inter-cellular signaling[13]. Synthetic chemical patterning systems that could produce and manipulate similarly complex gradients might be useful for interaction with cells[14], for exerting dynamic spatial control over chemical systems, and for optimizing reactor throughput[6].

The idea that reaction-diffusion (RD) systems can produce patterns of concentrations was originally proposed by Alan Turing, who in a mathematical study of reaction and diffusion dynamics found mechanisms by which patterns of high and low chemical concentrations could form in an environment where chemical concentrations are initially homogeneous except for occasional local fluctuations in concentration[1]. Subsequent mathematical studies of chemical pattern formation[2,4-6] have suggested a family of reaction-diffusion processes capable of producing a diverse range of patterns including spots, stripes and spiral waves, and such patterns have been observed in chemical systems in two[3,15] and three[16] dimensions. However, because the features of these patterns arise when random fluctuations are amplified, the patterns they produce differ qualitatively from the robust patterns of signaling molecules observed in biological systems. The resulting features are often transient, and their exact shape and position are difficult to control reproducibly.

An alternative approach is to transform a simple pattern of initial chemical species into a more complex output pattern. This separates the task of generating complex patterns from the complementary task of breaking symmetry. Repeatable processes that perform pattern transformations could robustly produce specified patterns with well-controlled features on multiple size scales. Pattern transformation processes have been constructed using biological transcription factors active during development[17-19], and more recently, designed reaction- diffusion networks of synthetic DNA molecules[20,21]. While these systems can generate predictable patterns, the variety and complexity of geometries they can exhibit remains severely limited, and in many cases the patterns that arise are transient, such that the patterned species become well mixed over time.

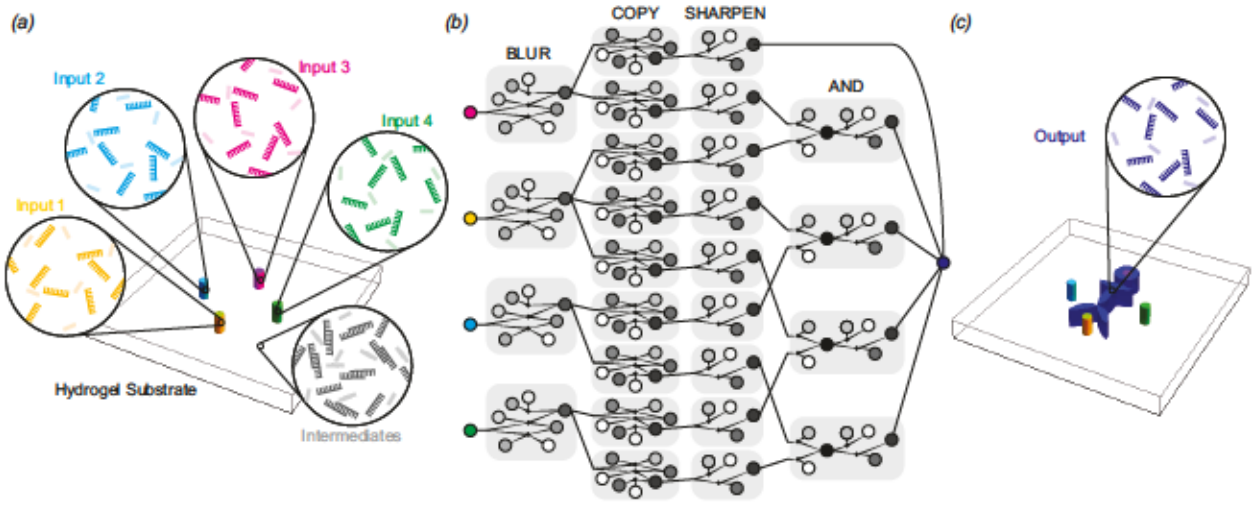

Figure 3.1. Reaction-diffusion programs for pattern generation. (a) Synthetic DNA "input" molecules (yellow, cyan, magenta, and green) are fixed at particular locations inside of a hydrogel substrate. Many other species of DNA "intermediate" molecules (gray) are homogeneously distributed throughout the hydrogel. (b) The intermediate molecules comprise a modular reaction-diffusion "program", where each module (boxed) encodes a specific patterning instruction. The shaded circles represent different chemical species, and the lines connecting the circles represent designed reactions between these species. Intermediate molecules are designed to react only with other intermediate molecules inside of their parent module, facilitating the design

of new patterns simply by rearranging the connections between modules. (c) The colored input molecules interact with the chemical reaction-diffusion network to produce a target pattern of DNA output molecules (blue).

Here we describe a staged pattern formation strategy in which specified patterns are gradually developed from simple initial patterns through a series of modular transformation stages. This strategy builds upon the robust repeatability of pattern transformations, expanding the complexity and diversity of attainable output patterns. Each stage performs a relatively simple transformation upon its input pattern, orchestrated by a reaction-diffusion network "module", producing an incrementally more elaborate output. Modules interact with each other only *via* their input and output patterns, enabling large composite networks to be flexibly assembled by adding modules or modified by rearranging the connections between modules. Viewing each module as a semantic command (*i.e.* COPY, BLUR or SHARPEN), staged networks represent algorithmic lists of patterning instructions, or *reaction-diffusion programs*, that generate complex patterns autonomously (Fig.1).

3.2 | A molecular language for spatial programming

A chemical RD system consists of molecular species that can diffuse and interact through a designed set of coupled chemical reactions. The *de novo* construction of a set of molecules that interact *via* a specific set of a designed reactions can be challenging because of the difficulty in finding or designing chemical species that react as intended without crosstalk and the challenge of independently controlling the diffusion coefficients of different species. The number of technical challenges involved in such a design process increases quickly as the size of the desired network grows.

Recently, it has been proposed that the use of synthetic DNA as molecular species in RD

systems could address many of these design challenges[20]. Advances in the development of DNA-based strand-displacement cascades have enabled the design of large reaction networks similar to biological signal-transduction networks[15,22,23]. The components of these cascades are short, synthetic DNA strands that are easy to synthesize and are biocompatible. DNA strand-displacement cascades have been used to implement robust logical operations and concentration amplification and thresholding. Strand-displacement cascades are programmed to perform these and other specific functions by appropriately designing their nucleotide sequences. Chemical reaction networks involving up to 130 unique species of DNA strands have been demonstrated[7,24]. While networks of this size are somewhat small compared to the size of the reaction networks within a cell, which may consist of thousands of signaling molecules and transcription factors[25], there is no fundamental obstacle to the design of larger synthetic DNA strand-displacement networks. Strand-displacement methods also offer unprecedented flexibility of design: methods summarized in Fig. 2 have been developed with the goal of designing a set of DNA molecules that can emulate any chemical reaction network[8,26]. While quantitative control over reaction rate constants within these networks can still be a challenge[8,27], reaction networks can be designed that only depend qualitatively on the relative rates (*i.e.* fast or slow) between different reactions. The versatility of strand-displacement reactions, along with a growing number of computational tools that are available for the design of these networks[28,29], suggests that the design of large-scale DNA-based reactions networks is feasible. The diffusion coefficients of DNA molecules are similarly programmable. In solution, single- and double- stranded DNA diffusion coefficients vary polynomially with sequence length[30] because their effective Stokes radius is a function of polymer chain length. In principle, therefore, adding additional bases to the strand or complex could slow down the diffusion coefficient of a species. More precise refinement of effective DNA diffusion rates is attainable within a porous substrate such as a hydrogel. By covalently attaching short DNA strand segments to the substrate,

complementary DNA molecules diffusing nearby can transiently bind to the attached segments, slowing down their transport in a sequence-dependent manner[31,32]. The time spent bound is controlled by tuning the density of binding sites and the energy of strand interaction, which can generally be predicted *in silico* using efficient algorithms[28,33]. While these interactions cause diffusion to become anomalous, if the density of binding sites is sufficiently large, the diffusion of DNA species would be expected to be generally uniform and continuous over length scales greater than a few hundreds of nanometers, and thus would obey the standard diffusion equation. Both of these methods for slowing diffusion can be used to independently set the diffusion coefficient of different species in the network. Methods for building complex, arbitrary networks of coupled chemical reactions with synthetic DNA, and for independently controlling the diffusion coefficient of each DNA strand could be combined to design and build complex reaction-diffusion systems *in vitro* with synthetic DNA. The oligonucleotides in these systems might also interact with other molecules in the environment in a sequence- specific fashion through the use of aptamers, nucleic acid sequences that bind specifically to a target ligand. Molecules for which aptamer sequences are known include many growth factors[34] and small molecules[35-37], and such connections have been generated previously as components of DNA strand-displacement networks[38].

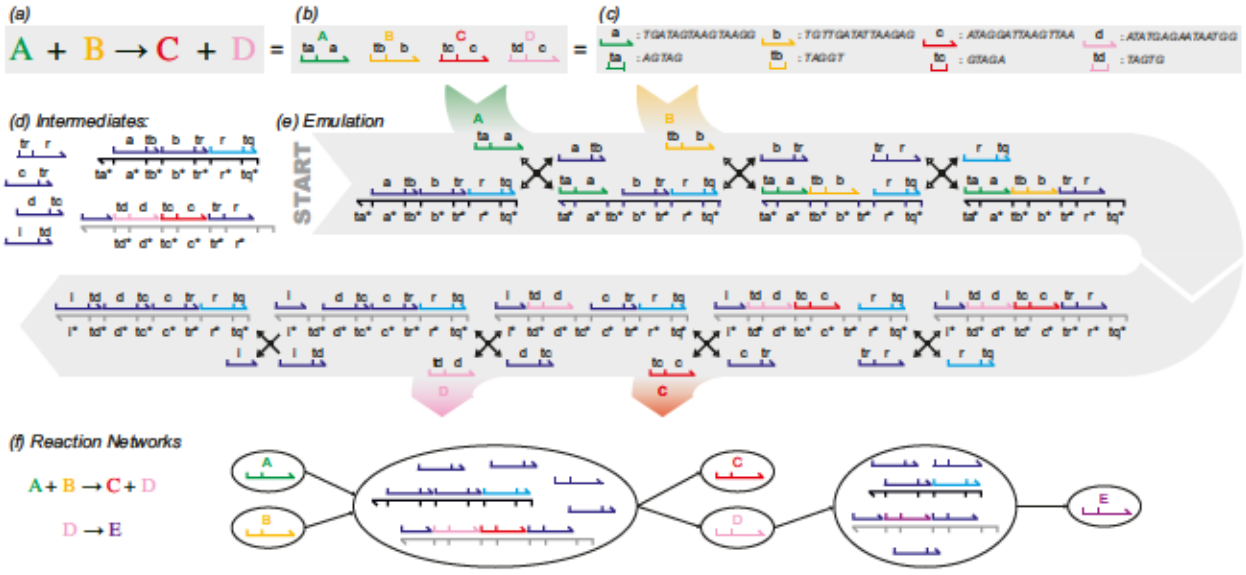

Figure 3.2. Networks of DNA molecules can emulate arbitrary chemical reaction dynamics. (Architecture from[8]). (a) A typical bimolecular chemical reaction. This reaction is emulated in the reaction process shown in b-d. (b) Specific single-stranded DNA molecules represent each reactant and product in the reaction. These molecules consist of a short "toehold" domain and a longer "recognition" domain. (c) Domains are unique sequences of base pairs (A, G, C and T). (d) A set of intermediate species interacts with the reactants. The collective reactions executed by the intermediates, reactants and products emulate the dynamics of the reaction in a. (e) The reactants A and B interact directly with the intermediate species, initiating a series of reactions that ultimately release the product species C and D. A reaction with fewer reactants or products can be emulated by making simple alterations to the intermediate complexes[8]. (f) Multiple such reactions can be chained together to form large reaction networks that emulate the dynamics of arbitrary networks of coupled chemical reactions.

3.3 | Self-sustaining pattern formation processes

Without a continuous supply of energy, heterogeneous distributions of chemical species will become well mixed over time, making the construction of steady-state heterogeneous patterns infeasible. In cells and tissues, proteins and other signaling molecules are constantly produced and

degraded, providing a source of energy with which to maintain heterogeneous patterns of chemical species in spite of the effects of diffusion. Analogous production and decay reactions that continuously produce new species from high-energy precursors and degrade old species into inert waste could likewise produce stable patterns of concentrations *in vitro*. Here we use this technique to produce stable, static patterns of chemical concentrations.

Strand displacement systems can emulate stabilizing production and degradation reactions powered by high concentrations of precursor molecules[26]. Production can occur *via* the slow conversion of an inert precursor present at high concentration into the active species, and degradation can occur *via* the slow conversion of the active species into a lower energy waste species.

3.4 | Modular reaction-diffusion programs

We define a module as a coupled set of chemical reactions that perform a designed transformation of a pattern of input molecules. Modules are connected together such that "upstream" modules produce an output species that "downstream" modules accept as input species. Hierarchical modules, consisting of multiple submodules, can perform more sophisticated pattern transformations. To produce self-sustaining patterns, modules include production and degradation reactions that resupply component species, and maintain heterogeneous patterns at steady state.

Modular engineering makes it possible to design large, complex networks by first designing simple, reusable modules, and then arranging them into networks. Each module implements a semantic patterning instruction (*e.g.* COPY or BLUR a pattern), which can be connected together without worrying about the details of how these instructions are implemented within each module. The modular instructions we use in this paper are detailed in Figure 3.

A reaction-diffusion program is a set of connected modules that orchestrate a pattern formation

process. To verify that the networks we design will form their target patterns, we use a mathematical model of each module's dynamics consisting of a set of coupled partial differential equations that governs their dynamics, assuming mass action kinetics[39]. To evaluate the function of the network, these equations can be integrated numerically using measured reaction rate constants and diffusion coefficients existing literature[27,40]. For instance, given an RD program consisting of the equations:

$$A + B \xrightarrow{k_1} C \qquad (3.1)$$

$$C \xrightarrow{k_2} \text{waste} \qquad (3.2)$$

the corresponding partial differential equation describing the concentration of C over space and time is:

$$\frac{\partial [C](t,x,y)}{\partial t} = D_c \nabla^2 [C](t,x,y) + k_1 [A](t,x,y)[B](t,x,y) - k_2 [C](t,x,y), \qquad (3.3)$$

where D_c is the diffusion coefficient for species C. Similar equations govern the behavior of species A and B. Details concerning our modeling process, including boundary conditions, parameter values and numerical integration techniques, are provided in Supplemental Information Section1.

3.4.1 | A "DRAW FRENCH FLAG" patterning program. A central hypothesis of developmental biology is that spatial patterns of morphogens can instruct groups of cells to differentiate into specialized roles. These morphogen concentrations often encode information in the form of either high or low local concentrations. A concentration can thus be a Boolean value, *i.e.* a variable with only two possible values: *high* or *low*. In the French flag model[19], a one-dimensional linear gradient of an input morphogen is translated into three discrete regions of gene expression with the same geometry as the tri-colored French flag, such that each of three species has either a high or low concentration in each of the distinct regions. In this example, cells in these three regions could then adopt specialized identities depending on which of the Boolean *blue*, *white* or *red* regions they occupy. While this model is broadly accepted as one explanation of pattern formation during

development[25,41], a physical reaction-diffusion network capable of recapitulating this pattern *in vitro* has not been described. Here we design a modular reaction-diffusion program that generates a French-flag pattern (Fig. 4) from a linear input gradient.

We assume that the input to the system is a stable linear gradient of a "morphogen" *I*,

$$[I](x) = x\left(\frac{1\,\mu M}{10\,mm}\right), \qquad (3.4)$$

where x: [0 mm, 10 mm]. The output that results is a stable French-flag pattern in which three different labeled output species, *blue* (B), *white* (W) and *red* (R) are present in the left, middle and right third regions respectively at approximately 1 μM. The concentrations of these species are close to zero elsewhere. To demonstrate our process of RD program design, we construct this patterning program in stages. We first consider how to produce a single red stripe using the reaction networks. We then add modules (Figure 3) to our circuit until the circuit also produces white and blue stripes as outputs.

The challenge in generating a French flag pattern from a linear input gradient is that small differences in the concentration of the input species along the gradient must be transformed into large differences in the concentration of the output species in the final output pattern. For example (see Fig. 4), just to the right of the red-white boundary, *[R]* at steady state should be high and *[W]* should be low. An arbitrarily small distance to the left, on the other side of the boundary, *[R]* should be low and *[W]* should be high. Yet between these points the input concentration *[I]* changes only linearly with distance. As shown in Figure 3a, a SHARPEN module can produce an output species that is *high* everywhere the input species exceeds a threshold concentration and *low* everywhere else. A SHARPEN module with a threshold concentration of $2/3\ \mu M$ operating on the input gradient could thus produce an intermediate species *S* only in the regions where $I > 2/3\ \mu M$, *i.e.* in the right third of the substrate, creating a large change in $[S_1]$ across the red-white boundary (Fig. 4a).

A second SHARPEN module with a threshold value of 1/3 μM could produce a different intermediate species $[S_2]$ in an area that covers two-thirds of the width of the substrate, *i.e.* where both the red and white stripes should exist in the final pattern. Together, S , which is present only in the rightmost third of the space, and S_2 would divide the space into three discrete regions that correspond to the three desired French flag stripes: one where $[S_2]$ and $[S_1]$ are both high, one where $[S_2]$ is high and $[S_1]$ *is low*, and one where the concentrations of both species are low (Fig. 4b).

However, because the SHARPEN module depletes the concentration of its input species, if two different SHARPEN modules both have the same input I, they would compete with each other for I. Because both modules would deplete I during the course of their operation, I would end up being below the threshold value for amplification everywhere. As a result, neither S_1 nor S_2 would be produced anywhere. This problem would be addressed if the two SHARPEN modules used as their respective inputs two different species I_1 and I_2 that each have the same concentration as I in all locations. To produce these species, we designed a COPY module (Fig. 3b) that takes an input I and produces an output O that has the same concentration as I everywhere without depleting the input I. We include a COPY module that produces a copy of I with the same concentrations as I upstream of each SHARPEN module to prevent the two SHARPEN modules from competing for the same input species. The resulting circuit (Fig. 4b) produces a pattern where S_1 is present in the rightmost third of the space and S_2 is present everywhere but the leftmost third of the space.

To produce the French flag pattern, output species R should be produced where $[S_1]$ is *high*, output species W should be produced where $[S_2]$ is *high* but $[S_1]$ is *low*, and output species B should be produced where both $[S_1]$ and $[S_2]$ are *low*. AND and NOT modules (Fig. 3c-d) can direct the

production of each of these species by producing the output molecule only where the corresponding Boolean function of the input concentrations is satisfied (Fig.4c).

The French flag patterning circuit in Fig. 4c is a collection of connected modules, each of which contains a small RD program comprised of abstract chemical species (such as I, A, T and O in the SHARPEN module). This abstract chemical reaction network can be translated into a set of strand-displacement reactions that implement the same dynamics as the reaction-diffusion network we designed. There may be many different ways to translate an abstract chemical reaction network into a set of DNA strand-displacement reactions using a method such as that shown in Fig. 2. In the Supplemental Information, we propose a potential strand-displacement network for each of our modules. One measure of the complexity of the resulting network is the number of species it contains. Our network (shown in Supp. Info. 2) consists of 77 unique initial DNA strand assemblies containing a total of 108 component DNA strands. The size of the network is therefore smaller than strand-displacement networks that have been experimentally demonstrated[7].

To test that the network we designed produces the target pattern, we numerically integrated the partial differential equations for the RD system consisting of the abstract chemical species (*I, A, T,* …). The results (Fig. 4c,f) show that the network is expected to produce the designed pattern, taking on the order of an hour to reach steady state. Close to the boundaries between the blue, white and red regions, there are transition areas where either the blue and white, or the white and the red species are both produced.

These transition regions arise because of the diffusion of the output species, which acts to blur the pattern, and because of what we term "leaks" in our reactions, in which a slow reaction uses a small amount of the product that is otherwise depleted very quickly by a much faster reaction.

The French flag patterning program demonstrates that we can design RD programs that use the geometric information provided by a heterogeneously distributed input species to produce an

incrementally more complex heterogeneous distribution of output species over a single spatial dimension. While microfluidic devices can already be used create a variety of gradient patterns with 1- dimensional features[42] similar to the French flag pattern, producing analogous patterns in 2- and 3-dimensions is considerably more complex.

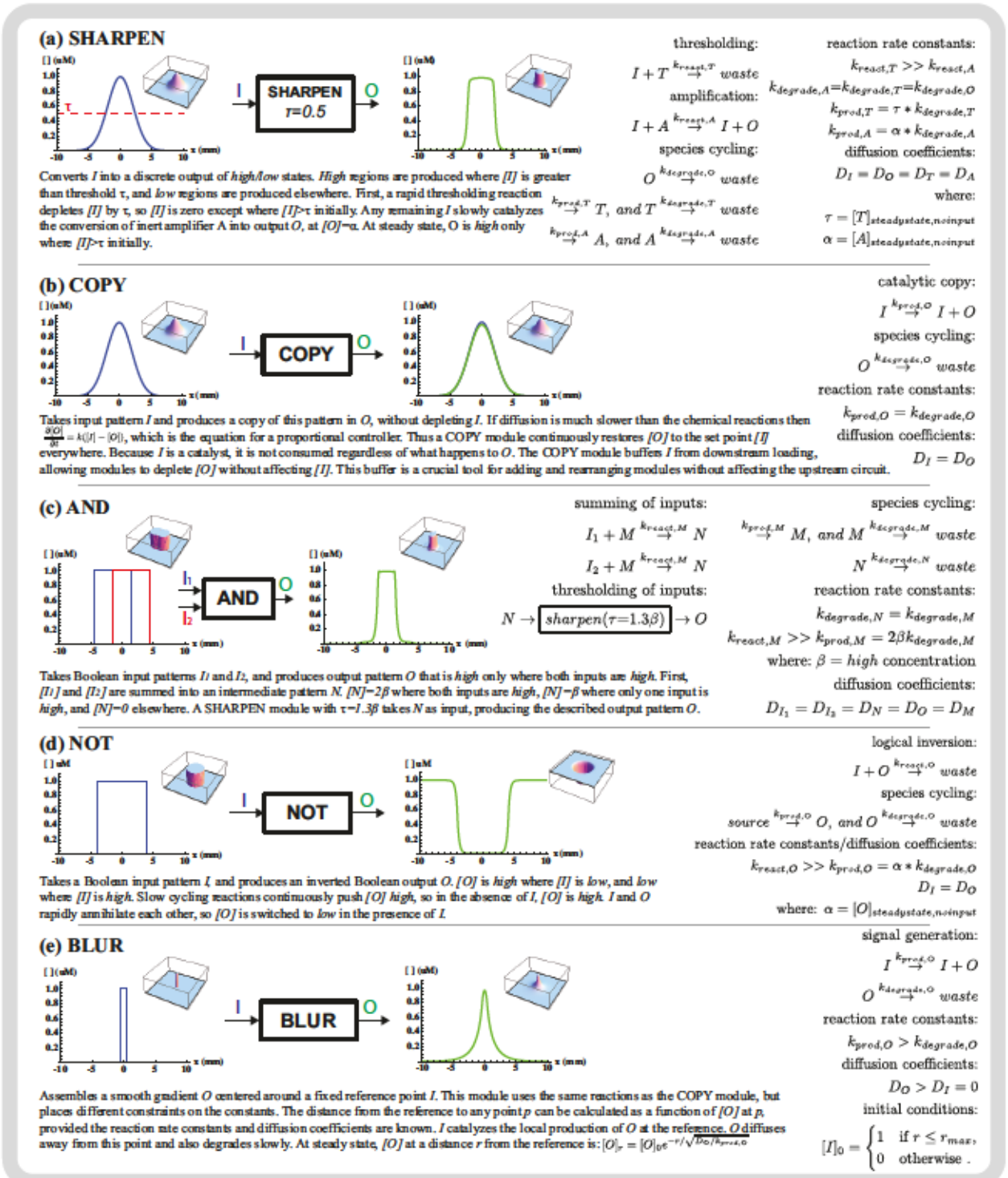

Figure 3.3. A library of reaction-diffusion modules for building pattern-formation programs.

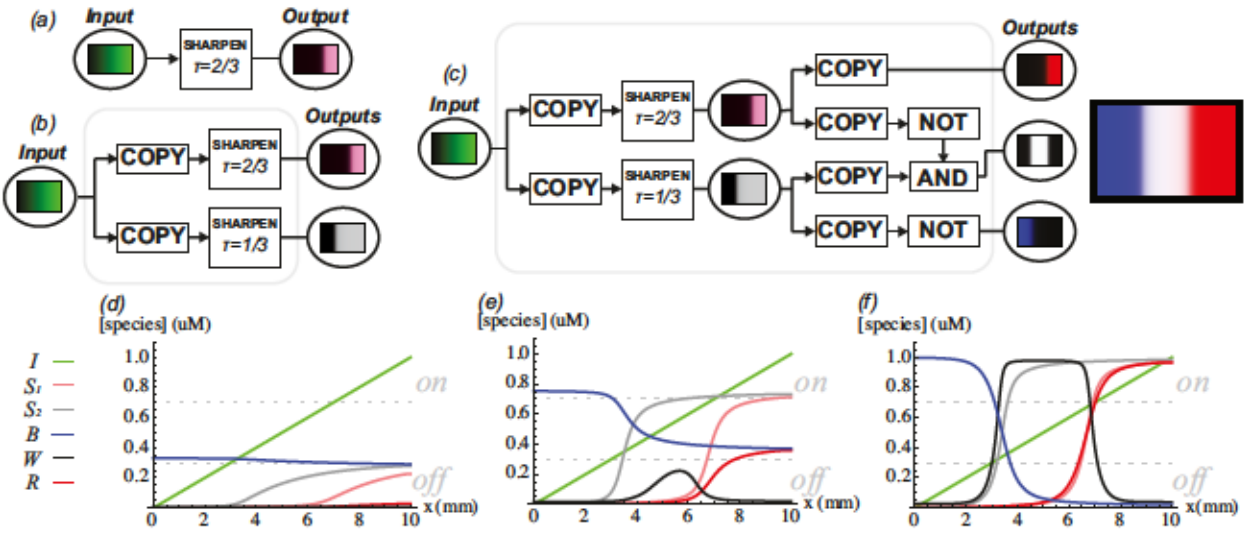

Figure 3.4. An RD program that produces a French flag pattern from a linear input gradient. (a) A simple network that generates a single stripe of intermediate species S_1 (pink) as output when the input is a linear input gradient (green). The stripe appears in regions where $[I] > 2/3\ \mu M$. The simulated steady-state profile of S_1 is shown in the output bubble. (b) A network to produce two stripes. Two copy modules produce two copies I_1 and I_2 of the input species I. I_1 and I_2 serve as inputs to two threshold amplifier modules that produce stripes of S_1 and S_2 (gray). The COPY modules protect the input signal from downstream loading, ensuring that the SHARPEN modules do not affect one another. The simulated steady state profiles of S_1 and S_2 are shown in the top and bottom output bubbles respectively. (c) The complete network to produce a French flag pattern from a linear gradient input. This network is an expanded version of the network in (b). A red stripe of R is produced where $[S_1]$ is *high*. A white stripe of W is produced where $[S_2]$ is *high* but $[S_1]$ is *low*. A blue stripe of B is produced where both $[S_1]$ and $[S_2]$ are *low*. The simulated steady state profiles of B, W and R are shown in the output bubbles, and are superimposed together in the image to the right. (d) Simulated concentrations of species in the French-flag RD program 200 seconds after introducing the input gradient, as they would appear if buffered against a load by a COPY module immediately downstream. Species are color coded to match the bubbles in (c). (e) Simulated concentrations of species after 700 seconds. (f) Simulated concentrations of species close to steady state. Simulation details

including reaction rates, diffusion coefficients, PDEs and initial conditions are provided in the Supplemental Information Section 2.1.

3.4.2 | A "DRAW STICK FIGURE" patterning program. To show that our patterning method can generate 2- dimensional patterns, we use the same step-wise modular design process to construct a program that produce a stick figure pattern that consists of a head, torso, arm and leg segments (Fig. 5a). In a 1-dimensional program like the French-flag generator, a SHARPEN module acting on a linear gradient created a linear *high*/*low* step function. In two dimensions, a radial gradient is generated when a species is produced at a single point and then slowly degrades as it diffuses away from its point of production. When the BLUR module (Fig. 3e) takes an input that is produced at a single point, it produces such a gradient. A SHARPEN module acting on a radial gradient creates a radial step function, *i.e.* a circle. The radius of this circle is specified by the threshold concentration of the SHARPEN module corresponding to the concentration of the radial gradient along the edge of the circle. The SHARPEN module will produce a *high* output signal only in regions where the radial gradient is above the threshold concentration, inside of the radius of the circle. The center

of the circle is set by the location of the input point to the BLUR module. Together, the BLUR and SHARPEN modules encode a DRAW CIRCLE patterning instruction, which itself can be considered a module. Using the DRAW CIRCLE instruction repeatedly along with the AND module, it is possible to produce a complete stick figure pattern from an input consisting of four particular inputs produced only at four particular points (Fig. 5b).

A variety of techniques might be employed to pattern the input points for the stick figuring, including micro-contact printing[9] or lithography[21]. Because the input patterns would be covalently bound to the substrate, they would remain stable over time rather than diffusing away, even though these inputs are not replenished. Alternatively, we could extend existing strand-displacement systems to actively generate and maintain the input pattern from reservoirs attached to the substrate

boundaries[43], analogous to the actively generated input gradient in the French Flag program.

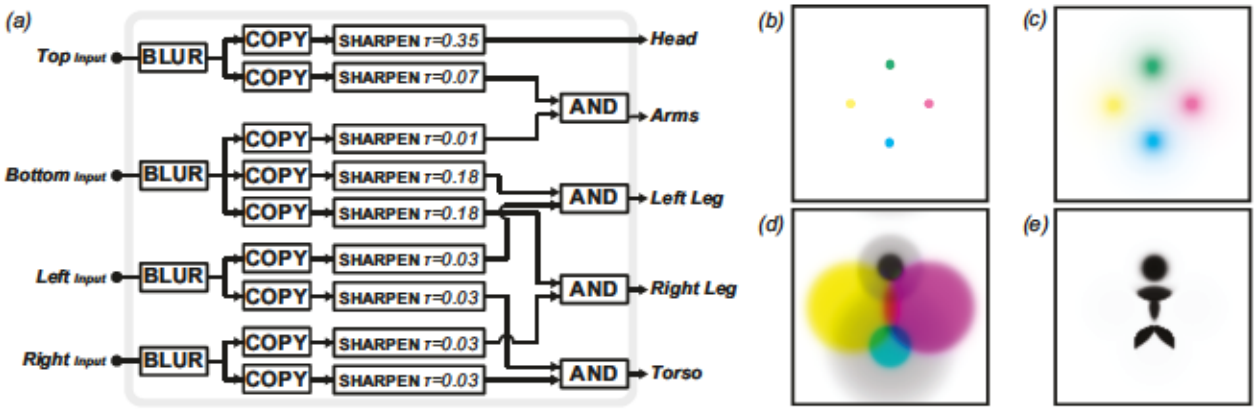

Figure 3.5. An RD program that produces a stick figure when the input pattern is a set of input species fixed at the illustrated locations. (a) The network of RD modules that interprets the signals from the input species and produces the stick figure shape as output. The magnitudes of the threshold concentrations τ for each of the SHARPEN modules set the respective sizes of the circles in d. (b-e) Simulated stick figure formation. (b) The input, four species (shown in different colors) that are each fixed at their corresponding illustrated locations. (c) BLUR modules take the four input species as inputs, producing gradients around them of controlled size. A set of COPY modules produces species with the same spatial extent and concentration as the original inputs. The species produced by the COPY modules are used as buffered inputs to the downstream modules. (d) A set of SHARPEN modules produce circles with defined radii around the input points. The intersections of these circles define the regions where the stick figure segments should appear: the magenta and yellow circles intersect at the stick-figure's torso, the yellow and cyan at the left leg, the magenta and cyan at the right leg, the two light-gray circles at the arms, and the black signal defines the stick-figure's head. (e) AND modules take each pair of species in (d) as inputs and produce the output species (black) where both inputs are present at high concentration. Simulation details, including reaction rates, diffusion coefficients, PDEs and initial conditions are provided in Supplemental Information Section 2.2.

Since the stick figure's head is a circle, we can pattern the head with a single DRAW CIRCLE module. To build the rest of the stick figure, we need a mechanism to produce line segments for the

arms, torso, and legs. One convenient way to produce an approximate line segment is to produce a species at the intersection of two large circles: If two circles are separated by slightly less than then sum of their radii, the intersection is a line-like region. We can repeatedly use the composite DRAW CIRCLE instruction to pattern pairs of circles that intersect at the desired locations for each segment, and an AND module to produce output species where these circles intersect. As with the DRAW FRENCH FLAG program, we include COPY modules where needed to prevent modules from depleting the concentrations of their input patterns and affecting upstream modules.

The combined program produces a stick figure pattern from an input pattern consisting of four points. The pattern is expected to take on the order of a day to reach steady state using the rate parameters estimated for our DNA species, provided in the Supplemental Information. The limiting factor preventing the speedup of formation is the diffusion rate of the DNA.

It is straightforward to rearrange components or add modules to this system to change the geometry of the output pattern. One could imagine using such a program to generate any pattern that consists of a set of line segments. While we were able to re-use some of the input signals in the DRAW STICK FIGURE program by careful use of the COPY module, in general each line-segment feature requires two input signals and seven modules. One interesting goal for future studies is to generate a large number of pattern features using only a fixed number of inputs and modules.

3.5 | Discussion

In this chapter we have described a method for the design of *reaction-diffusion programs*, sets of modular instructions encoded into autonomous chemical pattern generators. Each module takes simple input patterns and generates incrementally more complex output patterns. Elaborate, stable patterns of soluble molecules emerge through the combined effects of each modular stage. While

the reaction-diffusion programs we describe are large, it is plausible to imagine implementing them *in vitro* using existing techniques for designing molecules with prescribed rates of reaction and diffusion: synthetic DNA strand- displacement networks of similar sizes have already been demonstrated. The continued growth in complexity and robustness of devices that can be designed with DNA strand-displacement networks suggests that the range of designed RD programs that are experimentally realizable will expand. Further, we can imagine a "compilation" process that automatically translates arbitrary patterning programs into a network of chemical species that can be implemented using strand-displacement reactions. Specific sets of molecules can implement each module, and compilation can proceed by designing the reactions for each module such that the inputs and outputs for each module interact, but the species within different modules do not react at significant rates. Control over patterning could be extended from two into three dimensions using the existing reaction modules in 3- dimensional substrate by using an intersection of spheres instead of circles in the stick figure example, with the input points printed on, or generated from the boundary surfaces of the 3-dimensional substrate.

Eventually reaction-diffusion programs may be designed that include temporal control of network dynamics, capable of generating patterns that change shape over time, such as Belousov Zhabotinsky-type dynamics[3] or cellular automata[44]. Other potential extensions to our methods include programs that can generate more efficiently produce a wide variety of pattern features, and programs that produce patterns with more graded analog responses instead of only digital *high* or *low* values.

While large DNA strand-displacement networks have been implemented previously, they tend to be "one-time- use" circuits, which calculate a single output before reaching a state in which components no longer react with one another. This strategy is effective for well-mixed solutions, but reaction-diffusion patterns require a constant supply of energy to maintain a stable steady state. To

provide this energy we design networks for which high- energy input species are constantly supplied and low- energy waste species are removed or degraded. Such a supply could be provided by a reservoir containing high concentrations of the necessary raw materials connected to our system directly, or if needed, through a membrane to control which molecules can pass through.

Our modules are designed around abstract chemical reactions that are not specifically constrained to any single implementation medium. In the Supplemental Information, we provide detailed descriptions of DNA strand- displacement species that emulate each proposed reaction, including the production and degradation reactions that supply energy to our system. However, the same abstract reactions could be compatible with a variety of other existing mechanisms[45-47]. Mechanisms that employ enzymatic activity could provide a higher energy density per molecule, consuming energy-source molecules at a lower rate compared to strand displacement networks.

An important feature of our modular design process is that patterns are produced through stages of iterative refinement. Chemical circuits found in well-studied biological development networks, such as the sea urchin network for endodermal-mesodermal differentiation[48,49] and the *Drosophila melanogaster* segment polarity network[12], exhibit many of the same basic design principles that we employ here, including the effective division of the network into modules based on their function, and the use of Boolean on/off signals to rectify concentration fluctuations, uncertainty in reaction rate constants and diffusion coefficients, or imperfectly synthesized component molecules. However, these biological circuits also appear to include feedback loops and redundant "fail-safe" layers of circuitry, in which parallel transcription pathways ensure normal operation even when one pathway fails, making them even more robust than the networks we describe. We expect that if we were to include similar mechanisms in our synthetic systems, our RD programs could become more robust. Like networks that control body plan formation in biology, our pattern formation networks can produce multitudes of pattern variations with a limited set of components, either by changing

the arrangement of modules, or by adding and removing modules. Because our networks are organized into modules with specific, well understood functions, producing a desired new pattern is straightforward. Small portions of the pattern can be changed by changing modules that are downstream of most others in the network, while large changes to the pattern can be made by changing the modules in the network that function at the beginning of the patterning process. By modifying just a few connections or modules in a network, we could make either of these types of changes.

If reaction-diffusion programs can be readily implemented in the laboratory, they could become a flexible platform for delivering target molecules to defined portions of a substrate. We view delivered DNA as an information-carrying device; this information can translated into the form of molecules other than DNA through the use of aptamers[50]. In this context, RD programs represent spatially intelligent processors that could orchestrate complex behaviors in a growing variety of next generation biomaterials.

Acknowledgements

The authors would like to thank Paul Rothemund, Georg Seelig, Josh Fern, Abdul Mohammed, Ankur Verma, John Zenk, and the anonymous referees for insightful reading and comments. This work was supported by NSF-CCF-1161941 and a grant to the Turing Centenary Project by the John Templeton Foundation.

References

1. Turing, A. M. The chemical basis of morphogenesis. *Phil. T. Roy. Soc. B* 237, 37–72 (1952).
2. Pearson, J. E. Complex patterns in a simple system. *Science* 261, 189–192 (1993). 3. Zhabotinsky, A. M. Periodic processes of malonic acid oxidation in a liquid phase. *Biofizika* 9, 306–311 (1964).
4. Meinhardt, H. *The Algorithmic Beauty of Sea Shells* (Springer- Verlag, Berlin Heidelberg, 1995). 5. Epstein, I. R. & Pojman, J. R. *An Introduction to Nonlinear Chemical Dynamics* (Oxford University Press, New York, 1998). 6. Murray, J. D. *Mathematical Biology II: Spatial Models and Biomedical Applications* (Springer, New York, 2003), 3rd ed.
7. Qian, L. & Winfree, E. Scaling up digital circuit computation with DNA strand displacement. *Science* 332, 1196–1201 (2011).

8. Chen, Y.-J., Daichau, N., Srinivas, N., Phillips, A., Cardelli, L, Soloveichik, D., & Seelig, D. "Programmable chemical controllers made from DNA."*Nat. Nano.* 8 755-762 (2013).
9. Ruiza, S. A. & Chen, C. S. Microcontact printing: A tool to pattern. *Soft Matter* 3, 168–177 (2007). 10. Du, Y., Lo, E., Ali, S. & Khademhosseini, A. Directed assembly of cell-laden microgels for fabrication of 3D tissue constructs. *P. Natl. Acad. Sci.* 105, 9522–9527 (2008).
11. Lawrence, P. A. *The making of a fly: the genetics of animal design* (Blackwell Scientific Publications Ltd., Oxford, UK, 1992). 12. Bastiaens, P., Caudron, M., Niethammer, P. & Karsenti, E. Gradients in the self-organization of the mitotic spindle. *Trends Cell Biol.* 16, 125–134 (2006).
13. Lecuit, T. *et al.* Two distinct mechanisms for long-range patterning by Decapentaplegic in the *Drosophila* wing. *Nature* 381, 387–393 (1996). 14. El-Ali, J., Sorger, P. K. & Jensen, K. F. Cells on chips. *Nature* 442, 403–411 (2006).
15. Padirac, A., Fujii, T., Estvez-Torres, A. & Rondelez, Y. Spatial waves in synthetic biochemical networks. *J. Am. Chem. Soc.* (2013). 16. Bnsgi, T., Vanag, V. K. & Epstein, I. R. Tomography of reaction-diffusion microemulsions reveals three-dimensional turing patterns. *Science* 331, 1309–1312 (2011).
17. Isalan, M., Lemerle, C. & Serrano, L. Engineering gene networks to emulate Drosophila embryonic pattern formation. *PLoS Biol.* 3, e64 (2005). 18. Driever, W. & Nasslein-Volhard, C. A gradient of *bicoid* protein in Drosophila embryos. *Cell* 54, 83–93 (1988).
19. Wolpert, L. Positional information and the spatial pattern of cellular differentiation. *J. Theor. Biol.* 25, 1–47 (1969). 20. Allen, P. B., Chen, X., Simpson, Z. B. & Ellington, A. D. Modeling scalable pattern generation in DNA reaction networks. *Artif. Life* 13, 441–448 (2012).
21. Chirieleison, S. M., Allen, P. B., Simpson, Z. B., Ellington, A. D. & Chen, X. Pattern transformation with DNA circuits. *Nature Chem.* (2013). 22. Seelig, G., Soloveichik, D., Zhang, D. Y. & Winfree, E. Enzyme-free nucleic acid logic circuits. *Science* 314, 1585– 1588 (2006).
23. Zhang, D. Y. & Seelig, G. Dynamic DNA nanotechnology using strand displacement reactions. *Nat. Chem.* 3, 103–113 (2011). 24. L. Qian, E. W. & Bruck, J. Neural network computation with
DNA strand displacement cascades. *Nature* 475, 368–372 (2011). 25. Alon, U. *An Introduction to Systems Biology: Design Principles of Biological Circuits* (CRC Press, Boca Raton, Florida, 2006).
26. Soloveichik, D., Seelig, G. & Winfree, E. DNA as a universal substrate for chemical kinetics. *P. Natl. Acad. Sci.* 107, 5393– 5398 (2010).
27. Zhang, D. Y. & Winfree, E. Control of DNA strand displacement kinetics using toehold exchange. *J. Am. Chem. Soc.* 131, 17303-17314 (2009).
28. Zadeh, J. N. *et al.* NUPACK: Analysis and design of nucleic acid systems. *J. Comput. Chem.* 32, 170–173 (2011).
29. Phillips, A. & Cardelli, L. A programming language for composable DNA circuits. *J. R. Soc. Interface* 9, S419S436 (2009).
30. Smith, D. E., Perkins, T. T. & Chu, S. Dynamical scaling of DNA diffusion coefficients. *Macromolecules* 29, 1372–1373 (1996).
31. Kenney, M., Ray, S. & Boles, T. C. Mutation typing using electrophoresis and gel-immobilized acrydite probes. *Biotechniques* 25, 516–521 (1998).
32. Allen, P. B., Chen, X. & Ellington, A. D. Spatial control of DNA reaction networks by DNA sequence. *Molecules* 17, 13390– 13402 (2012).
33. SantaLucia, J., Jr. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *P. Natl. Acad. Sci.* 95, 1460–1465 (1998).
34. Ng, E. W. *et al.* Pegaptanib, a targeted anti-VEGF aptamer for ocular vascular disease. *Nat. Rev. Drug Disc.* 5, 123132 (2006).
35. Famulok, M. Oligonucleotide aptamers that recognize small molecules. *Curr. Op. Struc. Biol.* 9, 324329 (1999).
36. Baker,B.R.*et al.* An electronic, aptamer-based small-molecule sensor for the rapid, label-free detection of cocaine in adulterated samples and biological fluids. *J. Am. Chem. Soc.* 128, 31383139 (2006).
37. An, C.-I., Trinh, V. B. & Yokobayashi, Y. Artificial control of gene expression in mammalian cells by modulating RNA interference through aptamer-small molecule interaction. *RNA* 12, 710716 (2006).
38. Dirks, R. M. & Pierce, N. A. Triggered amplification by hybridization chain reaction. *P. Natl. Acad. Sci.* 101, 15275–15278 (2004).
39. Horn, F. & Jackson, R. General mass action kinetics. *Arch. Ration. Mech. An.* 47, 81–116 (1972).
40. Stellwagen, E. & Stellwagen, N. C. Determining the electrophoretic mobility and translational diffusion coefficients of DNA molecules in free solution. *Electrophoresis* 23, 2794– 2803 (2002).
41. Gierer, A. & Meinhardt, H. A theory of biological pattern formation. *Kybernetik* 12, 3039 (1972).
42. Jeon, N. L. *et al.* Generation of solution and surface gradients using microfluidic systems. *Langmuir* 16, 83118316 (2000).

43. Allen, Peter B., Xi Chen, Zack B. Simpson, and Andrew D. Ellington. "Modeling Scalable Pattern Generation in DNA Reaction Networks." In *Artificial Life*, vol. 13, pp. 441-448. 2012.
44. Burks, Arthur W. *Von Neumann's self-reproducing automata*. University of Michigan, (1969).
45. Nitta, Nao, and Suyama, A. Autonomous biomolecular computer modeled after retroviral replication. *DNA Computing*. Springer Berlin Heidelberg, (2004). 203-212.
46. Kim, Jongmin, White, K. S., and Winfree, E. Construction of an in vitro bistable circuit from synthetic transcriptional switches. *Molecular systems biology* 2, no. 1 (2006).
47. Montagne, K., Plasson, R. Sakai, Y., Fujii, T., and Rondelez, Y. Programming an in vitro DNA oscillator using a molecular networking strategy. *Mol. Sys. Biol.* 7, 1 (2011). 48. Smith, J. & Davidson, E. H. Regulative recovery in the sea urchin embryo and the stabilizing role of fail-safe gene network wiring. *P. Natl. Acad. Sci.* 106, 1829118296 (2009).
49. Levine, M. S. & Davidson, E. H. Gene regulatory networks for development. *P. Natl. Acad. Sci.* 102, 49364942 (2005).
50. Jhaveri, S., Rajendran, M. & Ellington, A. D. In vitro selection of signaling aptamers. *Nat. Biotechnol.* 18, 12931297 (2000).

4 | Supplementary Information: Designing Modular Reaction-Diffusion Programs for Complex Pattern Formation

4.1 | Module Details

4.1.1 | *SHARPEN* Module A homogeneously produced threshold species T rapidly reacts with an input I, reducing $[I]$ to zero where I is produced more slowly than T. A homogeneously produced amplification species A reacts more slowly with excess I leftover from the thresholding reaction, producing O. The concentration of O is stabilized by a degradation reaction.

Below are the *PDEs* that govern how species are affected by this module. These are only the terms that result from the module's interaction with each species. For species on the module's interface (*i.e.* input and output species), additional external terms may affect the behavior of the species, and must be added to these equations in context. Specifically, we do not include production or degradation reactions on the input signal *PDEs*, assuming that these reactions are handled by the upstream circuit.

$$\frac{\partial [I](t,x,y)}{\partial t} = D_I \nabla^2 [I](t,x,y) - k_{react,T}[I](t,x,y)[T](t,x,y)$$

$$\frac{\partial [T](t,x,y)}{\partial t} = D_T \nabla^2 [T](t,x,y) + k_{prod,T} - k_{degrade,T}[T](t,x,y) - k_{react,T}[I](t,x,y)[T](t,x,y)$$

$$\frac{\partial [A](t,x,y)}{\partial t} = D_A \nabla^2 [A](t,x,y) + k_{prod,A} - k_{degrade,A}[A](t,x,y) - k_{react,A}[I](t,x,y)[A](t,x,y)$$

$$\frac{\partial [O](t,x,y)}{\partial t} = D_O \nabla^2 [O](t,x,y) - k_{degrade,O}[O](t,x,y) + k_{react,A}[I](t,x,y)[A](t,x,y)$$

Initial conditions:

$$[T](0,x,y) = [A](0,x,y) = [O](0,x,y) = 0\ \mu M$$

Main Text Fig. 3a uses periodic boundary conditions and a *24 hour* simulation time. In addition

to the *PDEs* defined above, the following equation determines how we modeled the upstream behavior of the input signal *I* for the specific context in Main Text Fig. 3a. This equation must be added together with the terms that describe how the module affects the input signal in order to model the full behavior of the input signal in Main Text Fig. 3a.

$$\frac{\partial[I](t,x,y)}{\partial t} = k_{prod,I}[S](t,x,y) - k_{degrade,I}[I](t,x,y)$$

$$[I](0,x,y) = 0\ \mu M$$

with static source species S governed by

$$\frac{\partial[S](t,x,y)}{\partial t} = 0$$

$$[S](0,x,y) = e^{-\frac{(x^2+y^2)}{8}}\ \mu M$$

Also in this simulation, $k_{prod,A} = \alpha k_{degrade,A}$ and $k_{prod,T} = \tau\ k_{degrade,T}$, where the thresholding set point $\tau=0.5$, and the amplification set point $\alpha=1$. All other cycling (*i.e.* production and degradation) reactions use $k_{cycling}=0.002$ s^{-1}. Thresholding reaction rate constants are $k_{thresholding}=20$ M^{-1} s^{-1}, and all remaining reaction rate constants are $k_{react}=0.2$ M^{-1} s^{-1}. The diffusion coefficient for all diffusing species is $D=0.000150$ mm^2 s^{-1}. We selected all of the diffusion rates used in our simulations to be the same order of magnitude as experimentally derived data from the literature[1, 2].

The DNA strand-displacement network outlined in Fig. 4.1 implements this module. This network uses 11 initial species complexes, consisting of 14 unique strands of DNA (including the input strand).

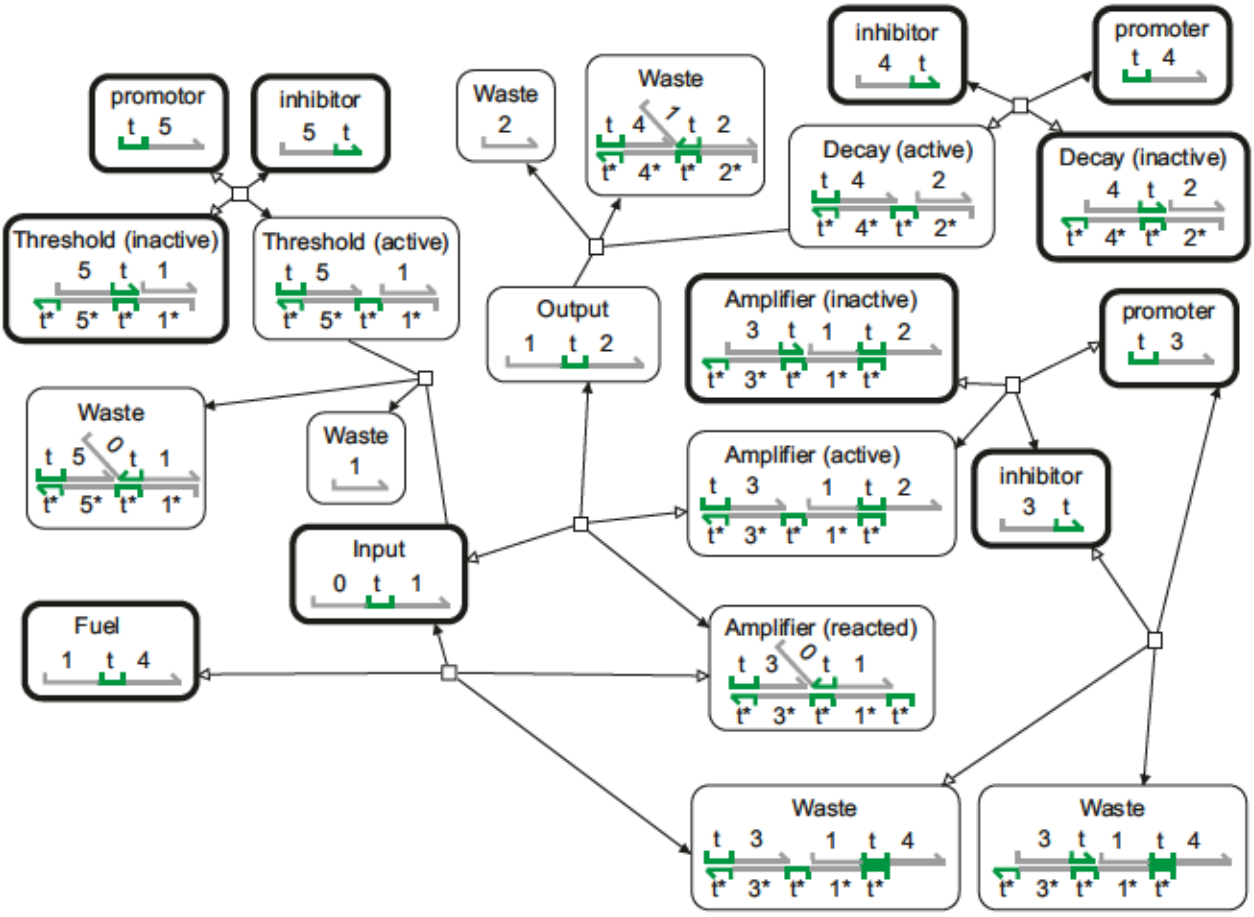

Figure 4.1. SHARPEN Module translated into strand-displacement reactions. Boxes contain domain-level definitions of each DNA species involved in the reaction network. Species that are defined in the abstract chemical reaction definition for this module have bold line weights on their boxes, while all other species are intermediate species necessary to emulate the designed abstract chemical reaction. Arrows connecting the involved species represent reactions between species. Black arrowheads designate forward reactions, and white arrowheads designate reverse reactions.

4.1.2 | *COPY* Module

An input species *I* catalyzes the production of an output species *O*. The input species is not produced, consumed, or significantly affected by this module. To produce a stable steady state, *O* also is degraded.

PDEs that govern how species are affected by this module:

$$\frac{\partial[O](t,x,y)}{\partial t} = D_O \nabla^2[O](t,x,y) + k_{prod,O}[I](t,x,y) - k_{degrade,O}[O](t,x,y)$$

Initial conditions:

$$[O](0,x,y) = 0\ \mu M$$

The production and degradation rates of O are equal, so if the time scale of diffusion is sufficiently slower than the time scale of the reaction kinetics, the equation governing *[O]* is well approximated by

$$\frac{\partial[O](t,x,y)}{\partial t} \approx k_{cycling,O}\left([I](t,x,y) - [O](t,x,y)\right)$$

and at steady state, *[O](t,x,y)=[I](t,x,y)*. Main Text Fig. 3b uses periodic boundary conditions, a *24* hour simulation time, and

$$[I](t,x,y) = e^{\frac{(x^2+y^2)}{8}}\ \mu M$$

with reaction rate constants $k_{prod,O} = 0.002$ s^{-1} and $k_{degrade,O} = 0.002$ s^{-1}, and diffusion coefficients $D_I = D_O = 0.000150$ $mm^2\ s^{-1}$.

The DNA strand-displacement network outlined in Fig. 4.2 implements this module. This network uses 6 initial species complexes, consisting of 8 unique strands of DNA (including the input strand).

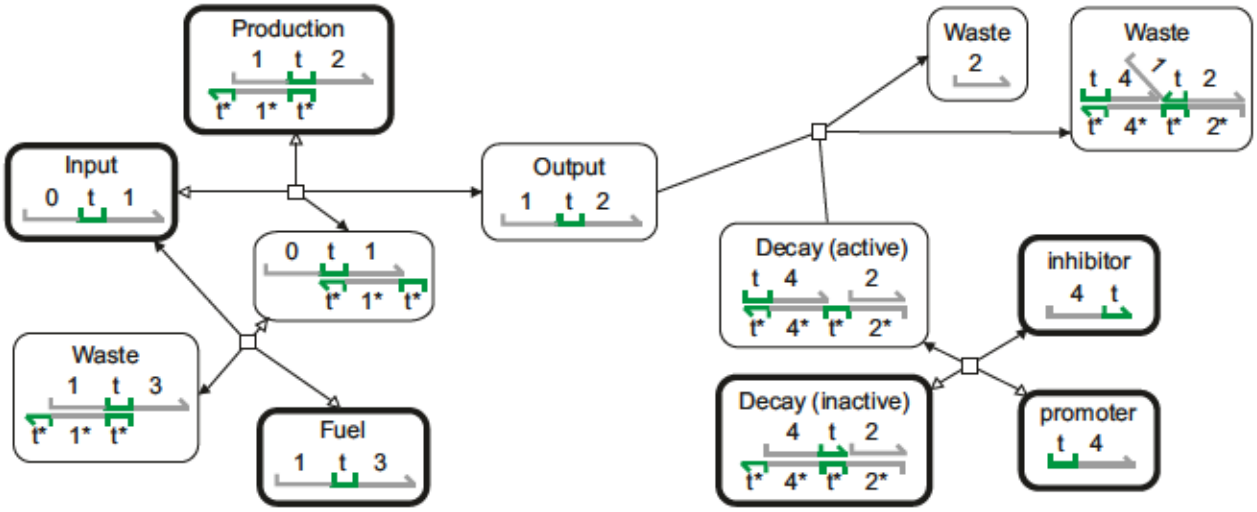

Figure 4.2. COPY Module translated into strand-displacement reactions.

4.1.3 | *AND* Module. The PDEs that govern how species are affected by this module are:

$$\frac{\partial[I_1](t,x,y)}{\partial t} = D_{I_1}\nabla^2[I_1](t,x,y) - k_{react,M}[I_1](t,x,y)[M](t,x,y)$$

$$\frac{\partial[I_2](t,x,y)}{\partial t} = D_{I_2}\nabla^2[I_2](t,x,y) - k_{react,M}[I_2](t,x,y)[M](t,x,y)$$

$$\frac{\partial[M](t,x,y)}{\partial t} = D_M\nabla^2[M](t,x,y) + k_{prod,M} - k_{degrade,M}[M](t,x,y) - k_{react,M}[I_1](t,x,y)[M](t,x,y) - k_{react,M}[I_2](t,x,y)[M](t,x,y)$$

$$\frac{\partial[N](t,x,y)}{\partial t} = D_N\nabla^2[N](t,x,y) - k_{degrade,N}[N](t,x,y) + k_{react,M}[I_1](t,x,y)[M](t,x,y) + k_{react,M}[I_2](t,x,y)[M](t,x,y)$$

Initial conditions: $[M](0,x,y) = [N](0,x,y) = 0\ \mu M$

Summed species N is then passed through a SHARPEN module with threshold concentration τ = 1.3 and amplify concentration $\alpha = 1$.

Main Text Fig. 3c uses periodic boundary conditions, a 24 hour simulation time, and

$$\frac{\partial[I_1](t,x,y)}{\partial t} = k_{prod,I}[S_1](t,x,y) - k_{degrade,I}[I_1](t,x,y)$$

$$\frac{\partial[I_2](t,x,y)}{\partial t} = k_{prod,I}[S_2](t,x,y) - k_{degrade,I}[I_2](t,x,y)$$

starting from $[I_1](0,x,y) = [I_2](0,x,y) = 0\ \mu M$, with source species S_1 and S_2 governed by

$$\frac{\partial[S_1](t,x,y)}{\partial t} = \frac{\partial[S_2](t,x,y)}{\partial t} = 0$$

$$[S_1](0,x,y) = \begin{cases} 1\ \mu M, & if\ (x+1.5)^2 + y^2 < 3^2 \\ 0\ \mu M, & otherwise. \end{cases}$$

$$[S_2](0,x,y) = \begin{cases} 1\ \mu M, & if\ (x-1.5)^2 + y^2 < 3^2 \\ 0\ \mu M, & otherwise. \end{cases}$$

Also in this simulation $k_{prod,M} = 2\beta k_{degrade,M}$, where the set point $\beta = 1$. All other cycling (i.e. production and degradation) reactions use $k_{cycling}$=0.002 s^{-1}. Thresholding reaction rate

constants are $k_{thresholding}$=20 $M^{-1} s^{-1}$, and all remaining reaction rate constants are k_{react}=0.2 $M^{-1} s^{-1}$. The diffusion coefficient for all diffusing species is D=0.000150 $mm^2 s^{-1}$.

The DNA strand-displacement network outlined in Fig. 4.3 implements this module. This network uses 14 initial species complexes, consisting of 19 unique strands of DNA (including the input strand).

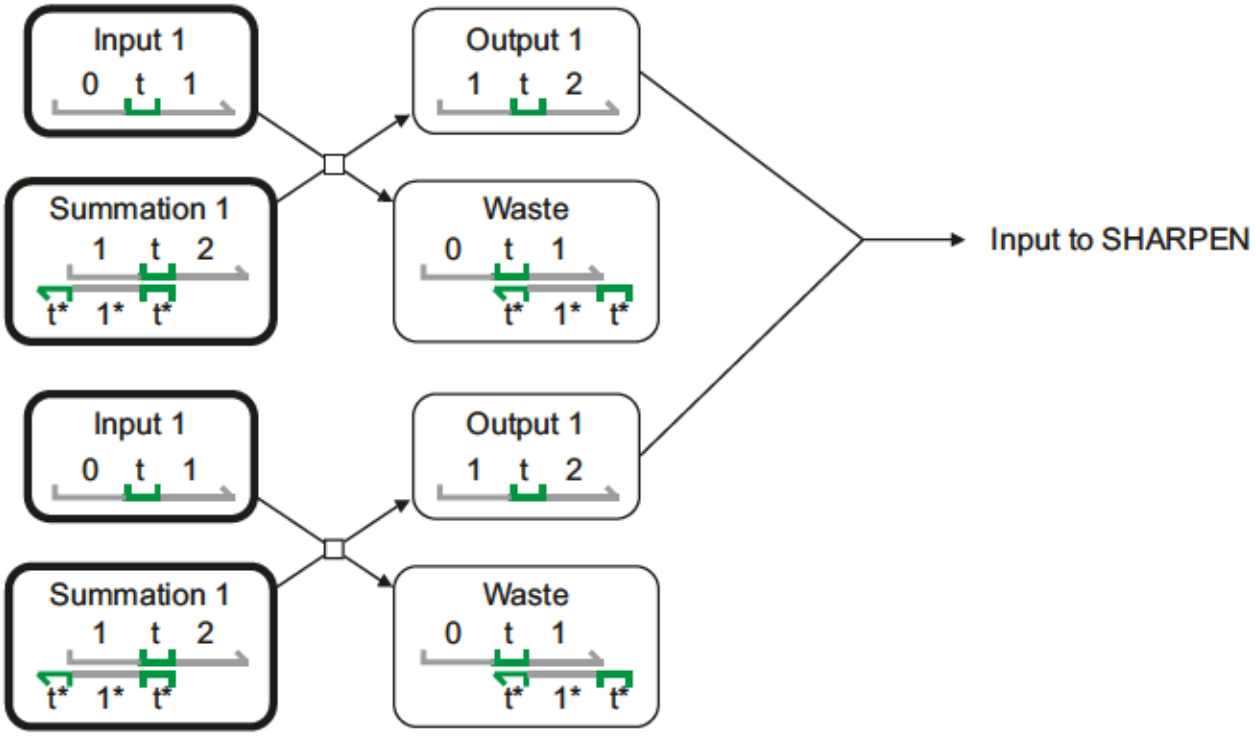

Figure 4.3. AND Module translated into strand-displacement reactions.

4.1.4 | *NOT* Module. The PDEs that govern how species are affected by this module:

$$\frac{\partial [I](t,x,y)}{\partial t} = D_I \nabla^2 [I](t,x,y) - k_{react}[I](t,x,y)[O](t,x,y)$$

$$\frac{\partial [O](t,x,y)}{\partial t} = D_O \nabla^2 [O](t,x,y) + k_{prod,O} - k_{degrade,O}[O](t,x,y) - k_{react}[I](t,x,y)[O](t,x,y)$$

Initial conditions: $[O](0,x,y) = 0\,\mu M$

Main Text Fig. 3d uses periodic boundary conditions, a 24 hour simulation time, and

$$\frac{\partial[I](t,x,y)}{\partial t} = k_{prod,I}[S](t,x,y) - k_{degrade,I}[I](t,x,y)$$

starting from $[I](0,x,y) = 0\ \mu M$ with source species S governed by

$$\frac{\partial[S](t,x,y)}{\partial t} = 0$$

$$[S](0,x,y) = \begin{cases} 1\ \mu M, & if\ x^2 + y^2 < 4^2 \\ 0\ \mu M, & otherwise. \end{cases}$$

Also in this simulation $k_{prod,O} = \alpha\ k_{degrade,O}$, where $\alpha = 1$. All other cycling (i.e. production and degradation) reactions use $k_{cycling}$=0.002 s^{-1}. Thresholding reaction rate constants are $k_{thresholding}$=20 $M^{-1}\ s^{-1}$, and the inversion reaction rate constant between I and O is k_{react}=0.2 $M^{-1} s^{-1}$. The diffusion coefficient for all diffusing species is D=0.000150 $mm^2\ s^{-1}$.

The DNA strand-displacement network outlined in Fig. 4.4 implements this module. This network uses 8 initial species complexes, consisting of 12 unique strands of DNA (including the input strand).

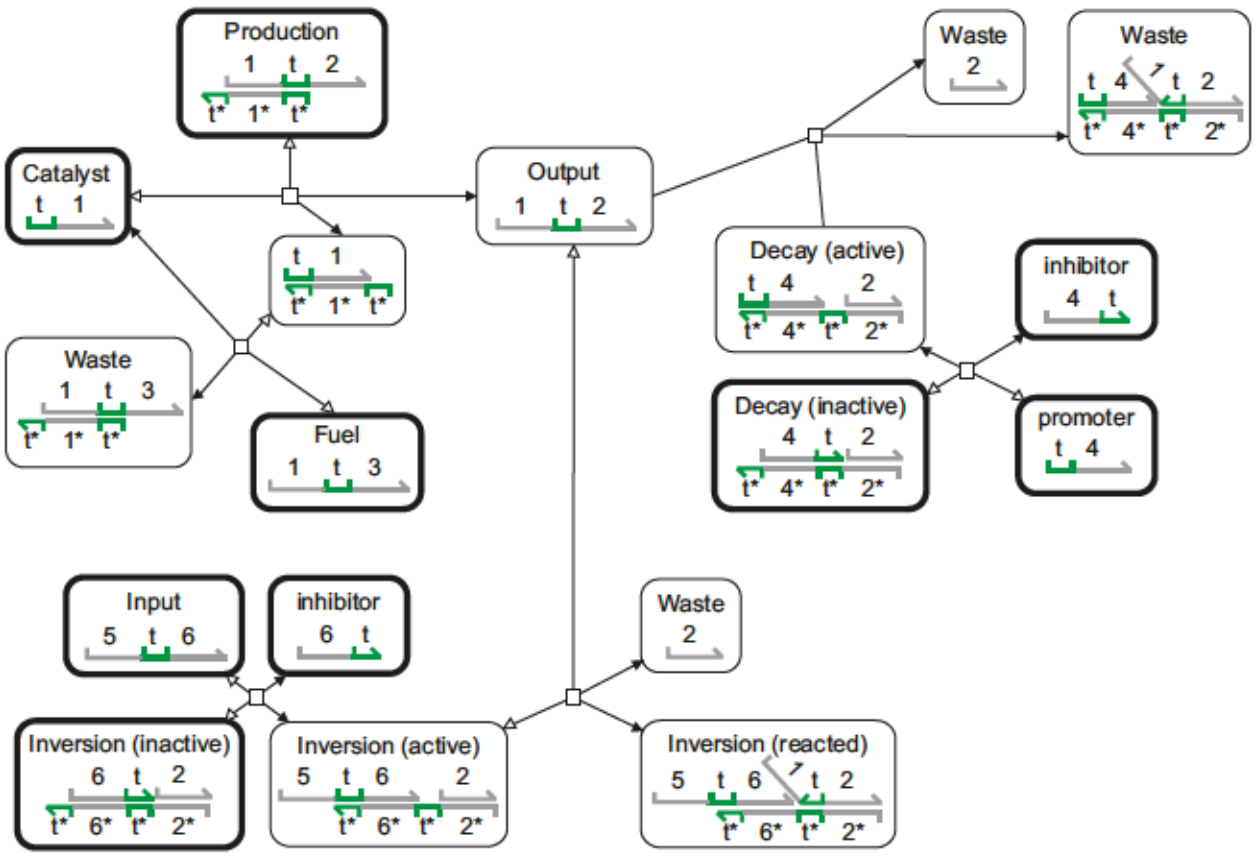

Figure 4.4. NOT Module translated into strand-displacement reactions.

4.1.5 | *BLUR* Module. Input species *I* catalyzes the production of an output species *O*. The input species is not produced, consumed, or otherwise affected by this module. *O* diffuses away from locations where it is produced, and is degraded to produce a blurred copy of the input patter at steady-state concentration.

PDEs that govern how species are affected by this module:

$$\frac{\partial [O](t,x,y)}{\partial t} = D_O \nabla^2 [O](t,x,y) + k_{prod,O}[I](t,x,y) - k_{degrade,O}[O](t,x,y)$$

Initial conditions: $[O](0,x,y) = 0\ \mu M$

If the input pattern is a single reference point of high concentration *I*, centered at (x_0, y_0) with radius *r*, where

$$[I](0,x,y) = \begin{cases} 1\ \mu M, & if\ (x-x_o)^2 + (y-y_0)^2 < r^2 \\ 0\ \mu M, & otherwise. \end{cases}$$

then the resulting output pattern is a gradient where *[O]* drops off monotonically with radial distance from the fixed reference point.

Main Text Fig. 3e uses periodic boundary conditions, a *24* hour simulation time, $r = 0.5$ mm, $x_0 = 0$ mm, $y_0 = 0$ mm, reaction rate constants $k_{prod,O} = 1$ s^{-1} and $k_{degrade,O} = 0.001$ s^{-1}, and diffusion coefficient $D_O = 0.000100$ $mm^2\ s^{-1}$.

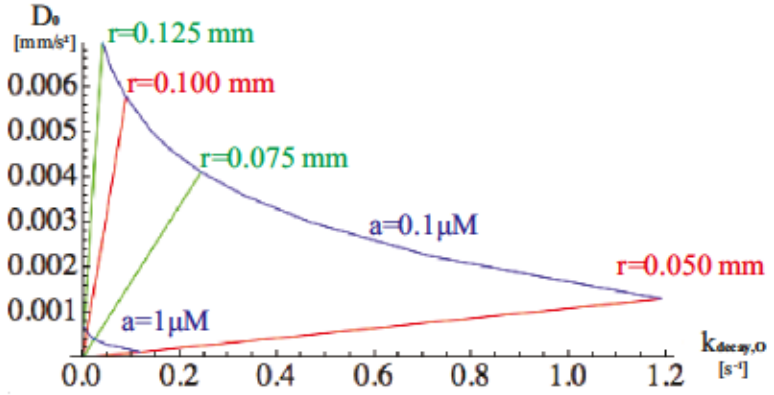

Figure 4.5. BLUR module parameter space. For the BLUR module simulation in the Main Text Fig. 3e: By holding $k_{prod,O}=1$ s^{-1} and tuning the values of $k_{degrad,O}$ and D_O, a diverse range of output signal amplitudes *a* and half-amplitude radii *r* are produced.

The DNA strand-displacement network outlined in Fig. 4.6 implements this module. This network uses 6 initial species complexes, consisting of 8 unique strands of DNA (including the input strand).

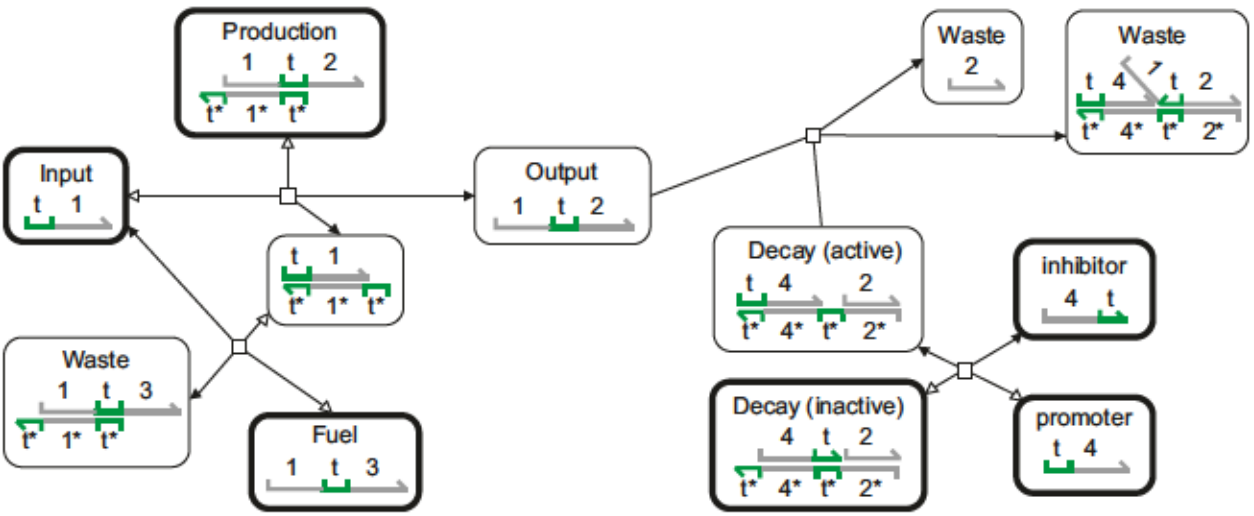

Figure 4.6. BLUR Module translated into strand-displacement reactions.

4.2 | Program Details

4.2.1 | French Flag Program. The simulations for the French flag program (Main Text Section 3.1) were simulated over 24 hours on x: 0, 10 mm with reflective boundary conditions. The input pattern I for this program is a linear gradient,

$$[I](0, x) = \frac{x}{10}\ \mu M$$

Cycling (*i.e.* production and degradation) reactions use $k_{cycling}$=*0.002* s^{-1}. Thresholding reaction rate constants are $k_{thresholding}$=*20* M^{-1} s^{-1}, and all other reaction rate constants are set to k_{react} = 0.2 M^{-1} s^{-1}. The diffusion coefficient for all diffusing species is *D=0.000150* mm^2 s^{-1}.

The threshold concentration τ = *0.667* for the high concentration *SHARPEN* module that produces species S_1. The threshold concentration τ = *0.333* for the medium concentration *SHARPEN* module that produces species S_2. All "on" reference concentration parameters (α, β) are set to *1* M.

One way to measure the complexity of a program is to count the number of DNA strands or

initial complexes it requires for implementation. The French flag program uses one input species (1 strand or 1 complex), six *COPY* modules (6 strands or 8 complexes each), two *SHARPEN* modules (11 strands or 14 complexes each), two *NOT* gates (8 strands or 12 complexes), and an *AND* gate (14 strands or 19 complexes each), however we need to subtract the input species (1 for the *COPY*, 1 for the *SHARPEN*, 1 for the *NOT*, and 2 for the *AND*) since they are already counted as the output of the upstream module. This results in a count of 77 initial complexes or 108 component DNA strands for the French flag program.

4.2.2 | Stick Figure Program. The simulations for the stick figure program (Main Text Section 3.2) were simulated over 24 hours on x: −10, 10 mm, y: −10, 10 mm with reflective boundary conditions. *SHARPEN* modules used production rate constant $k_{prod,SHARPEN}=0.2/300$ s^{-1} and $k_{degrade,SHARPEN}=0.2/5000$ s^{-1}. All other cycling (*i.e.* production and degradation) reactions use $k_{cycling}=0.002$ s^{-1}. Thresholding reaction rate constants are $k_{thresholding}=20$ M^{-1} s^{-1}, and all other reaction rate constants are set to $k_{react}=0.2$ M^{-1} s^{-1}. The diffusion coefficient for all diffusing species is $D=0.000150$ mm^2 s^{-1}.

The inputs for this simulation were four points with radius $r = 0.5$ mm, of top, bottom, left or right input species at $[species] = 1$ M. These points were located 4 mm away from the origin in the positive-x, negative-x, negative-y and positive-y directions, respectively. The reference threshold concentrations for the *SHARPEN* modules are as follows (these are the result of fractional approximations and do not require this level of precision):

Top circle used for head patterning: $\tau = 0.346525$

Top circle used for arms patterning: $\tau = 0.0704974$

Bottom circle used for arms patterning: $\tau = 0.00746513$

Left and right circles used for torso and legs patterning: $\tau = 0.0296572$

Bottom circle used for legs patterning: $\tau = 0.183469$

All "on" reference concentration parameters (α, β) are set to *1* M. The stick figure program uses four input species (1 strand or 1 complex each), four *BLUR* modules (6 strands/8 complexes each), nine *COPY* modules (6 strands or 8 complexes each), nine *SHARPEN* (11 strands or 14 complexes each), and four *AND* gate (14 strands or 19 complexes each). Subtracting the inputs (2 strands/species for the *AND* module and 1 strand/species for all other modules), this results in a count of 207 initial complexes or 280 component DNA strands for the stick figure program.

References

1. Lukacs, G.L., Haggie, P., Seksek, O., Lechardeur, D., Freedman, N., Verkman, AS, Size-dependent DNA mobility in cytoplasm and nucleus, Journal of Biological Chemistry 275, 1625 (2000).
2. Stellwagen, E., Lu, Y., and Stellwagen, N. C., Unified description of electrophoresis and diffusion for DNA and other polyions, Biochemistry, 42, 11745 (2003).

5 | Emulating Cellular Automata in Chemical Reaction-Diffusion Networks

Summary. Chemical reactions and diffusion can produce a wide variety of static or transient spatial patterns in the concentrations of chemical species. Little is known, however, about what dynamical patterns of concentrations can be reliably programmed into such reaction-diffusion systems. Here we show that given simple, periodic inputs, chemical reactions and diffusion can reliably emulate the dynamics of a deterministic cellular automaton, and can therefore be programmed to produce a wide range of complex, discrete dynamics. We describe a modular reaction-diffusion program that orchestrates each of the fundamental operations of a cellular automaton: storage of cell state, communication between neighboring cells, and calculation of cells' subsequent states. Starting from a pattern that encodes an automaton's initial state, the concentration of a "state" species evolves in space and time according to the automaton's specified rules. To show that the reaction-diffusion program we describe produces the target dynamics, we simulate the reaction-diffusion network for two simple 1-dimensional cellular automata using coupled partial differential equations. Reaction-diffusion based cellular automata could potentially be built in vitro using networks of DNA molecules that interact via branch migration processes and could in principle perform universal computation, storing their state as a pattern of molecular concentrations, or deliver spatiotemporal instructions encoded in concentrations to direct the behavior of intelligent materials.

5.1 | Introduction

A fundamental question in materials design is how we might program materials to sense and respond to dynamic signals across time and space. Biological materials routinely exhibit this capacity,

as cells and tissues sense and respond to a complex array of spatial and temporal cues. For example, during chemotaxis, many cells can detect gradients of chemoattractants and move in the direction of increasing chemoattractant concentration. In a mechanism like chemotaxis[16,11,2], cells use spatiotemporal chemical reaction networks to process information collected by distributed chemical sensors to decide on and execute responses to changing environmental conditions. In this paper we discuss the design of analogous synthetic chemical reaction networks that have robust, programmable spatiotemporal dynamics. The ability to engineer such systems could have wide-ranging applications for the design of smart, responsive, programmable materials.

To design a generic set of mechanisms that can process a wide range of input signals and invoke a wide range of responses, we consider a framework for distributed spatial computation that has been studied extensively – the cellular automaton (Fig. 1). A cellular automaton (CA) is a model of computation consisting of a rectangular lattice of domains, or 'cells'. At a given time a cell can be in one of a finite number of states, such as an on or off. In a synchronous CA, cells update their state once per time step based on their current state and the current states of a finite set of nearby cells. Although each cell update is relatively simple, groups of cells can together perform elaborate spatial computation. CA can execute any computable algorithm, a trait known as universal computation[9,10,7,17]. Specific automata also exist that can programmably construct any structure[19,6], self-replicate[19,6,12], mutate and evolve[25].

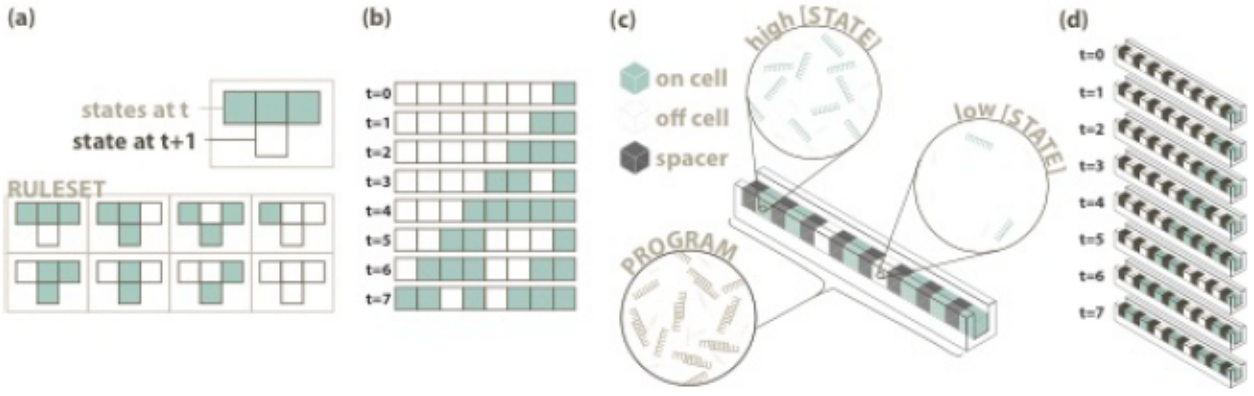

Figure 5.1. A cellular automaton consists of a lattice of cells. At a given time, each cell is in one of a finite number of states, shown by color (blue or white). Cell states change over time as the result of the application of local rules - finite functions that take as inputs the states of the current cell and a finite set of neighbors and produce the cells' new state as output. Here we consider a 1-dimensional automaton where each cell is either on or off, and where update rules take the cell's own state and those of its left and right neighbors as inputs. (a) An example rule set. (b) Example dynamics for the rule set in (a). (c) Schematic of the chemical reaction-diffusion cellular automaton described in this paper. A 1-dimensional channel contains cells separated by spacers. The state in each cell is encoded by either a high or low concentration of a 'state' species within that cell. Spacers between cells, which do not contain any state information, are shown in black. During the computation, the program and state species react and diffuse. This reaction-diffusion process maintains and updates cell state according to the rules of the desired cellular automaton. (d) Target dynamics of the state species for the example cellular automaton rule in (a).

In this paper we propose a strategy for building synchronous CA using chemical reaction-diffusion networks. We begin by breaking down CA into their fundamental operations: storage of cell states, communication between nearby cells, and calculation of new cell states. We demonstrate how existing chemical computing mechanisms could implement these operations. We then combine these chemical mechanisms to emulate two specific automata, known as 'Rule 110' and 'Rule 60'. These chemical CA can be viewed as a proof of concept that synthetic materials could sense signals across space and time and execute a broad class of dynamic programmed responses.

5.2 | Background: Reaction-Diffusion Processes for Computation

Reaction-diffusion (RD) networks are sets of chemically reactive species that diffuse within a continuous substrate. In contrast to a well-mixed chemical reaction system, reaction-diffusion (RD) networks can produce spatial patterns, where some species are more abundant in some parts of the substrate and less abundant in others. The interplay of reactions and diffusion can lead to sustained

spatial patterns and even the emergence of patterns from a homogeneous initial substrate which experiences transient concentration fluctuations[35]. Transient waves within these patterns can emulate Turing machines and perform basic computations[34,31,3].

Recently it has been shown that arbitrary chemical reaction networks can be readily designed and implemented in vitro using short strands of synthetic DNA[30,4]. Because DNA binding under many conditions is largely determined by the Watson-Crick sequence complementarity (A-T, G-C), reactive species can be designed to minimize unintended crosstalk between species that should not interact. These techniques separate the design of new reaction networks from the discovery of naturally occurring chemicals that perform the intended reactions. In support of this idea, large reaction networks involving up to 130 different sequences of DNA have been demonstrated in vitro without substantial crosstalk[22], and have been used to implement Boolean logic[22,27,23]. Further, the rates of the emulated reactions can be controlled[4].

It also appears plausible to extend this mechanism of DNA reaction design to the design of large reaction-diffusion networks, as the diffusion rates of different DNA strands can be programmed. Because the diffusion coefficient of DNA scales polynomially with the length of the strand[28], the diffusion rate of each species in a DNA reaction network can be independently tuned by adding or removing bases from a sequence, and such changes can be done so that the reactive propensity of a species is largely unchanged. Further, within a polymer gel, species attached to the gel substrate do not diffuse, but can continue to react. Together, the capacities to design arbitrary chemical reactions and tune the diffusion coefficient of each species in principle enable us to implement de novo simple RD networks that perform pattern transformations[1,5]. Here we ask how we might design an RD network that could be implemented by DNA molecules, given what is known about designing DNA-based reactions and diffusion processes. To focus on this question, here we ignore experimental nonidealities and the challenges of building large molecular networks, including

unintended crosstalk between species.

By designing RD network modules that perform simple, predictable, repeatable transformations to a pattern of chemical concentrations, circuits of modules can be combined to perform elaborate patterning operations[26]. Pattern transformation modules take specific species as inputs, perform reactions potentially involving some intermediate species within the module, and produce an output species (Fig. 2). Modules can be connected together with the output of upstream modules serving as the input to downstream modules. If these modules are designed such that the intermediate species of one module do not react with the intermediates of other modules, then many modules can operate simultaneously in the same substrate without interfering with each other. Further, by imposing the design requirement that modules must not significantly deplete (or "load") the concentrations of their inputs, it is possible to ensure that a module's reactions affect only other modules that lie downstream within the network. Thus, modules can be added one at a time to a system such that each addition of a module results in a simple, predictable change to the patterning process.

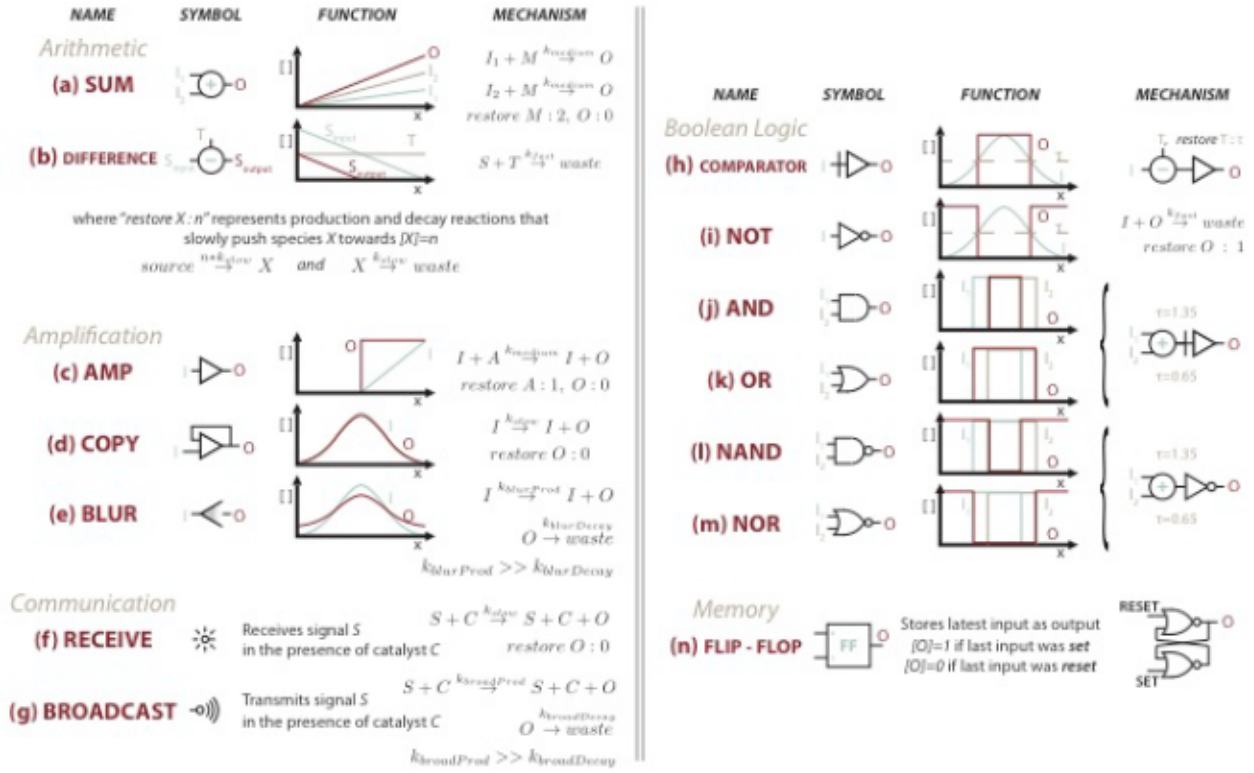

Figure 5.2. Reaction-Diffusion Modules. Each module is a set of simpler modules or of abstract chemical reactions that could in principle be emulated in vitro using a DNA strand-displacement network. In an emulation of these reactions using DNA strand displacement processes, species are represented as DNA strands or hybridized complexes of strands with particular sequences. Other strands or complexes are also required for the emulation, and act as "intermediates" in the reaction process[30,26]. More details on these modules and their operation can be found in[26].

Here we extend existing pattern transformation techniques to emulate a discrete, synchronous, 1-dimensional CA, generating spatial patterns of chemical concentrations with controlled dynamics. We design a network of reaction- diffusion pattern transformation modules (defined in detail in Fig. 2) in combination with a simple, static initial pattern and an external "clock" signal whose concentrations change periodically. This network forms the target CA structure, and controllably transforms the state of that structure over time. In principle, our abstract chemical reaction-diffusion network could be translated into a DNA strand displacement network for in vitro implementation.

One challenge in the design of pattern transformations is that the second law of dynamics implies that without the continual input of energy, purely diffusive patterns are unstable and tend to become well mixed over time. Thus, to prevent spatial patterns of soluble molecules from dissipating, reaction-diffusion networks will require a constant energy flux. One way to achieve this flux is to develop reactions that slowly release and degrade high-energy species. These reactions produce a sustained flux of molecules in the environment, and maintain a pattern such that only sporadic replenishment of some high-energy precursors are required to sustain the pattern formation process. Production reactions take the form *source* → *species*, and continuously produce reactants that are depleted by converting a high-energy precursor into the desired species. Degradation reactions take the form *species* → *waste*, and convert species that are produced into low-energy waste to stabilize the system.

5.3 | Anatomy of Our Reaction-Diffusion Cellular Automaton

Reaction-diffusion systems emulating a CA must be able to store the current state of the system as a tape or grid of cells and execute the update rules as a function of the states of the cell and the cell's left and right neighbors (Fig. 3). For the class of CA we consider here, the state of each cell is either on or off.

In our construction, each cell is a region of the substrate with a static, uniformly high concentration of a particular catalyst molecule. Catalyst molecules are attached to the substrate, so they do not diffuse. We call these molecules 'keys.' In our one-dimensional grid, cells can have one of four different types of keys (KeyA, KeyB, KeyC and KeyD) with key types repeating as one proceeds down the channel so that cells can identify neighbors based on local key type (Fig. 3a). For instance, cells defined by KeyA are always neighbored on the right by KeyB cells, and the left by KeyD cells. Cells are separated by 'spacer' regions that do not contain keys.

Since key molecules only participate in reactions as catalysts, it is assumed that they are not depleted over time. In this paper we assume that this pattern of key catalysts is present at the start of the reaction as a set of initial conditions. Such a grid pattern could either be manually patterned into the substrate by top-down techniques, such as microcontact printing[24] or directed hydrogel assembly[8], or generated by a separate bottom-up RD patterning program[26].

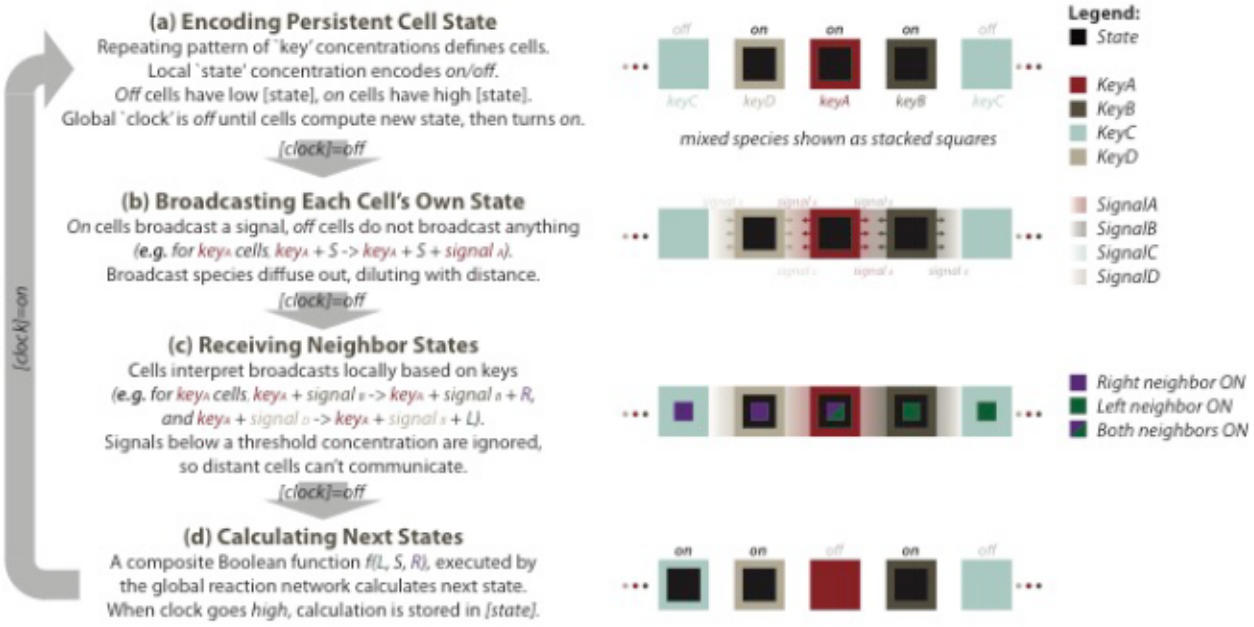

Figure 5.3. Our chemical CA performs three types of operations. (a) The state of each cell is stored locally. (b&c) Cells communicate their state to their immediate neighbor cells. (d) A Boolean logic circuit calculates each cell's next state as a function of the cell's own state and the state of its neighbors, and stores this new state in (a), completing one cycle of automaton dynamics. A global clock signal synchronizes these three operations. The clock is off for the communication and calculation stages, and turns on to allow the calculated new state to be stored in memory for the next cycle.

In addition to the static pattern of key catalysts, a mix of many other freely diffusing "program" species is supplied across the substrate. These program molecules interact with the key molecules to emulate the dynamics of a CA in a cycle of three discrete conceptual stages (Fig. 3b-c). In the first

stage, cells share their states with their neighbors and receive the states of other neighbors. Next, cells use information about their local state and the states of their neighbors to determine their state at the next time step. Finally, the calculated state is stored as the current state. Cycles of communication, calculation of new states, and storage of the new state in a cell's memory emulate the dynamics of a CA. In the construction below, an externally provided clock signal synchronizes the three cycles. While a global clocking mechanism isn't strictly required to implement a CA[18], we chose to construct a CA with a clock, because such a system can compute universally when each cell can take one of only two possible states.

5.3.1 | Communication: sending and receiving addressable signals. We begin the description of a CA cycle in the Communication Stage, right after a set of cell states for the last time step have been stably stored. At this point, each cell's current on/off state is represented, respectively, by the local high or low concentration of a 'state' species. During the communication stage of a computation cycle, cells must communicate their current state to their immediate neighbors.

Communication is managed by Broadcast and Receive modules (Fig. 4). Each on cell broadcasts information about its current state by producing a signal within the cell that can diffuse away from the cell. Broadcast modules (Fig. 2g) execute this function. In order for neighboring cells to interpret these broad- casts as coming from the left or right neighbor cell, these broadcasts must contain information about which cell is sending them. The identity of the cell broadcasting information about its state is encoded using the key types of cells: Cells that are defined by 'key A' species broadcast 'signal A', those defined by 'key B' broad cast 'signal B', and so on. The distance that broadcast signals propagate is controlled by the production and degradation rates of the Broadcast module, such that a cell's broadcasts only reach its neighbors. Each key has its own separate

broadcast module.

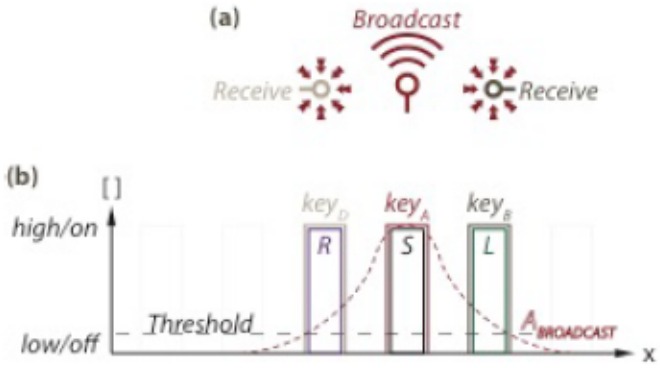

Figure 5.4. Communication Stage. (a) Using a broadcast module, each on cell produces a gradient encoding its current state and key. Receive modules interpret broadcasts based on the identity of their local key. (b) [species] vs. x for a single on A-type cell, with local [S] = high. A broadcast module generates a stable gradient of $A_{Broadcast}$ that decays with distance from KeyA. Receive modules at D-type cells interpret $A_{Broadcast}$ as R, while receive modules at B-type cells interpret the same broadcast as L. Broadcast signals below a threshold are ignored, so cells only communicate with their neighbors.

The counterpart to a Broadcast module, a Receive module (Fig. 2f), receives signals transmitted by a cell's neighbors and translates them into local information about the state of a neighboring cell. This conversion is also done in a cell- specific manner, such that each cell converts particular broadcasts into information about particular types of neighbors. For example, within cells defined by KeyA, the key species catalyzes the conversion of the broadcast signal from KeyB into a species that encodes the right neighbor's state as on and catalyzes the conversion of the broadcast signal from KeyD into a species that encodes the left neighbor's state as on. Key species B through D catalyze a corresponding set of reactions to produce signals that encode whether their right and left neighbors are on. Each conversion reaction of a broadcast to a type of neighbor information is managed by a separate receive module. Because there are four key types and each cell has two types of neighbors, eight Receive modules are required.

Receive modules convert broadcasts into "preliminary neighbor signals." These preliminary neighbor signals are at different concentrations throughout a cell because they are copies of the broadcast signals, which decay in concentration with the distance from the neighbor. To produce uniform on/off neighbor signals throughout a cell, comparators (Fig. 2h) rectify preliminary neighbor signals, producing digital "processed neighbor signals" whose on levels are the same across a cell.

Together, the broadcast and receive modules ensure that after some period of broadcasting, each cell contains species that encode its own state and those of its neighbors.

5.3.2 | Calculation Stage: calculating new state changes. Neighbor broadcasts that are received and processed by each cell are used to calculate the next cell state. Each update rule can be encoded as a Boolean circuit with the neighbors and the cell's own state as inputs. Such circuits can be implemented as a set of reaction- diffusion program modules (Fig. 5). For instance, in a Rule 60 CA, a cell's next generation state is on if its own current state is on OR its left-hand neighbor is on, but NOT if both of these states are on. Because the state of the right-hand neighbor is irrelevant, Rule 60 cells do not need to listen to their right-hand neighbor's broadcast. The logic for a Rule 110 local update is performed by the sub-circuit in Figure 5d. The out- put signal produced by this circuit determines the target state of the cell at the next time step.

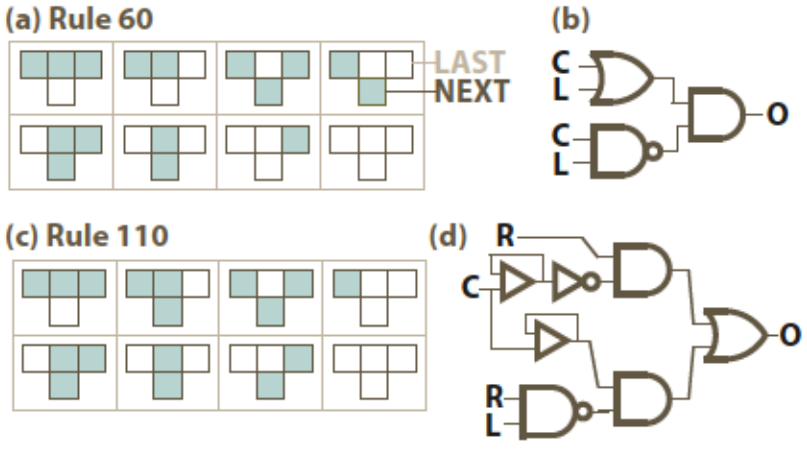

Figure 5.5. Calculation Stage. Every CA update rule has a corresponding Boolean logic implementation. (a) Rule 60. (b) Rule 60 converted into Boolean logic. (c) Rule 110. (d) Boolean logic for Rule 110.

5.3.3 | Memory: storing and stably encoding a new cell state. During the Memory stage the computed next state of the cell is stored using a "flip-flop" module (Fig. 6). Flip-flops have "set" and a "reset" input signal and one output signal. When the set input is on, the output signal also turns on. The output remains on even after the set signal turns off. When the reset signal turns on the output turns off, and remains off until the set signal again turns on. This module encodes the cell's state, providing a persistent state signal used by the communication and calculation stages.

The reactions that communicate a cell's state to its neighbors and compute its next state occur without control over timing. Different cells (or different regions within a cell) may take different amounts of time to compute their new state. To ensure that all cells finish computing their next states before any other cell commits its new state to memory, calculated next states are not stored in memory until a global clock signal turns on. For a given CA, the clock period must be designed so that all cells finish communicating and calculating before the clock signal turns on. The next state must be committed to memory before the clock turns off.

To ensure that calculated next states are not stored in memory unless the clock signal is on, an AND gate takes the clock signal and the calculated next state from the Calculation stage as inputs, and sends a set signal to the flip-flop module only when both the clock and the calculated next state are on. Another AND gate takes the clock signal and the inverse of the calculated next state as inputs, and produces a reset signal when the clock is on but the next state is off. The process of storing the new state in memory ends when the clock signal returns to a low value at the beginning of the next stage of computation.

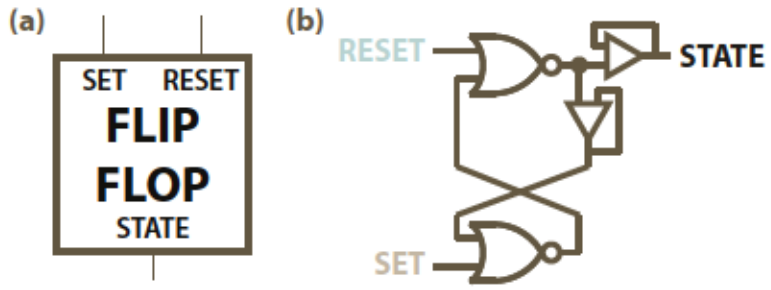

Figure 5.6. The Memory stage stores cell state in a flip-flop. (a) A flip-flop's output does not change when its inputs are off. In our design, these inputs are off when the clock is off. When the clock is on, the set signal is on if the calculation stage outputs on, setting the flip-flop on. The reset signal is on if the calculation stage outputs off, resetting the flip- flop off. Memory is required so the inputs to the other stages are not affected by the calculation of new local or neighbor states. (b) Circuit for our reaction-diffusion flip-flop using modules from Fig. 2. Copy modules ensure output is not depleted by internal feedback or downstream load.

5.4 | Simulation of a Reaction-Diffusion Cellular Automaton

The complete automaton circuit is shown in Fig. 7. We simulated the simultaneous mass action reaction kinetics and diffusion for the entire system, using Rules 60 and 110 in the logic stage, and observed the intended cell updating for both rules (Fig. 8). The complete set of chemical reactions and the corresponding coupled partial differential equations describing these systems are provided in the Appendix B, along with all other simulation details. Concentrations within [0, 0.3] μM were considered off, while concentrations within [0.7, 1] μM were considered on. One irregularity that appears in our system is that the cells have blurred boundaries, an artifact that arises when chemical species produced inside of a cell diffuse across the cell boundary. This blurring effect is the reason that we included short spacer regions to separate adjacent cells, so that the logic inside of one cell does not interfere with the logic inside of its neighbors.

Two important parameters can break the reaction- diffusion program if not tuned carefully: the on time or 'duty cycle' of the clock signal, and the kinetic rates for the broadcast module. If the duty

cycle is too short, then the flip-flop does not have enough time to store the intended next-generation state. In our simulations, this occurs for duty cycles shorter than 15-20 minutes. However, for particularly long duty cycles, some cell states can become desynchronized because cells can erroneously update their state multiple times within a single cell cycle. In our simulations, duty cycles longer than about an hour and a half led cells to become desynchronized.

The second critical parameter is the production rate constant for the broadcast module. When a cell is on, this constant must be high enough to saturate its neighbors with signaling molecule. In the worst case, where a cell is at the minimum on concentration of 0.7 μM, it must maintain a broadcast signal above the receive module's threshold concentration at the farthest edge of its neighboring cell regions. On the other hand, when a cell is off, this constant must be low enough to avoid broadcasting any signal to its neighbors. Specifically, in the worst case where a cell is at the maximum off concentration of 0.3 μM, it must maintain a broadcast signal below the receive module's threshold concentration at the closest edge of its neighboring cell regions. If either of these conditions is not met, then erroneous signals can be sent between cells.

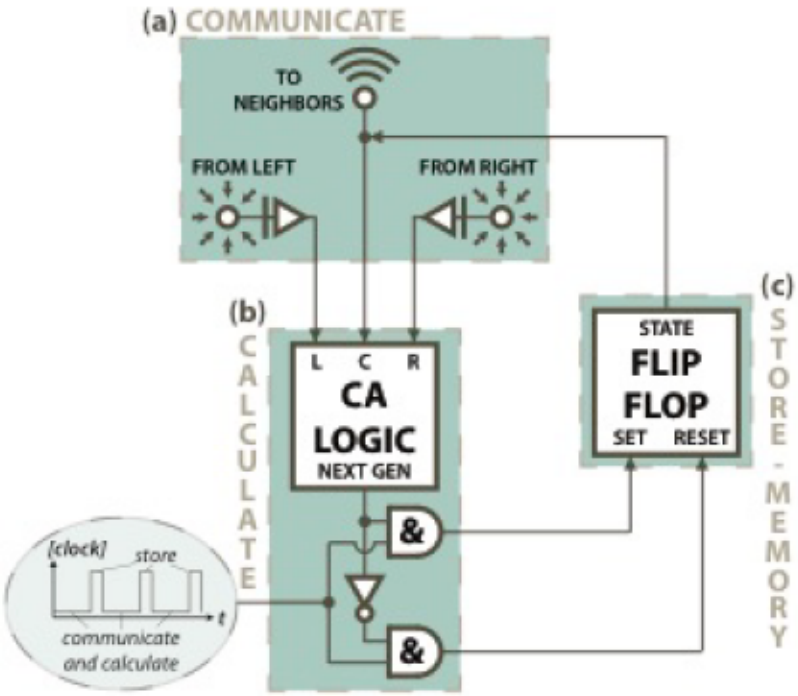

Figure 5.7. CA Circuit Diagram. (a) The 'Communication' stage. Current cell states are broadcast to neighbors, while neighbor states are received. (b) The 'Calculation' stage. The states of a cell and its neighbors are passed through a subcircuit that performs the update logic. The out- put from this subcircuit is on if the cell should change into an on state in the next generation. This next state is prevented from affecting the Memory stage by AND gates when the clock is off. (c) When the clock turns on, the next state is stored in the 'Memory' stage.

5.5 | Discussion

In this work we develop a method for building a CA using a reaction-diffusion program. This program consists of a set of molecules that everywhere can react and diffuse in the same way, along with a small set of molecules that are patterned into a grid of cells. The collective actions of these molecules cause the pattern of molecules that encode an "on" state to change over time to display a series of patterns that are the successive states of a one-dimensional binary CA. While the construction we propose is for a 1-dimensional binary CA, straightforward extensions to the system could be used to produce 2- or 3-dimensional CA, or CA in which cells can be in more than two

states.

This construction thus suggests two important new capabilities for systems that are driven by designed chemical reaction networks. First, this system provides a way to generate dynamic spatial patterns, where the concentrations of species vary continuously over time, by encoding these dynamics within a CA. Second, this system makes it possible to perform computations using molecules in which the same molecular species simultaneously perform different computations within different regions of a substrate.

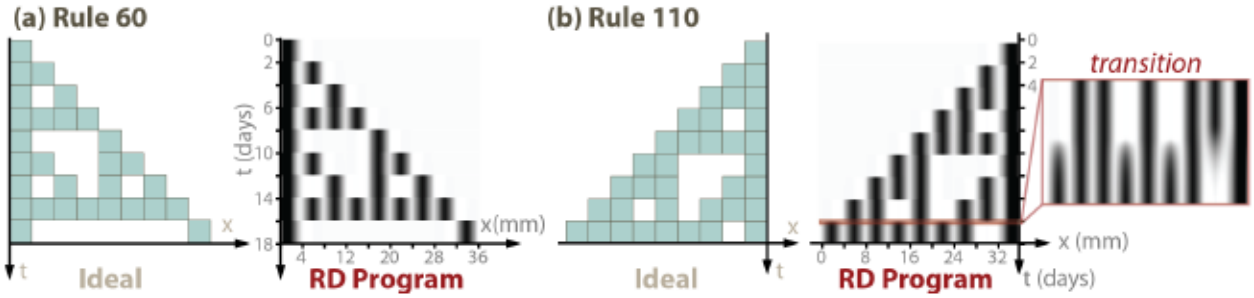

Figure 5.8. Results of Chemical CA Simulations. Ideal CA (left) compared to our simulated reaction-diffusion program (right). Every three-length binary input state is contained in each pattern, demonstrating correct updating for all eight possible local states. (a) Rule 60. (b) Rule 110. The dynamics shown here were computed using the set of coupled partial differential equations in the Appendix B. The detail of the rapid dynamics of a state transition are shown on the far right.

The capacity for this kind of spatial computation is likely to be an important part of scaling the capacity for molecular systems to process information. Because the number of independent molecular interfaces is inherently limited, it is not possible to arbitrarily increase the number of interacting molecules within a well-mixed system without introducing crosstalk. The importance of spatial computation with molecules is underscored by its prevalence in living systems. Reaction-diffusion processes are used for signaling within cells, and across tissues, where different cells (which each share the same genome) collectively coordinate tissue behavior.

While other molecular processes can perform Turing universal computation with only a limited number of molecular species, *i.e.* they are uniform, these constructions require that the state of a computation be encoded either within a single molecular assembly[21] or in the precise number of molecules[29]. As such, these constructions are susceptible to errors that can destroy the computation. In contrast, computation by the CA that we describe involves the collective action of many molecules, so it is not susceptible to errors caused by a small number of microscopic events.

However, the designs presented in this paper require the construction of large chemical reaction networks, a clock signal at regular intervals and a printed grid of different "key" molecules. Our reaction network uses 65 species to emulate a "Rule 60" CA, and 76 species to emulate a "Rule 110" CA. Further emulating these abstract chemical networks using DNA strand-displacement reactions could increase the network size by an order of magnitude, because multiple intermediate DNA strands are generally required when emulating reactions. Likely there are simplifications that could be made to our circuit, as our goal was to demonstrate that such an implementation is theoretically possible instead of designing the smallest possible circuit. For instance, it may be possible to condense some sections or our system into smaller special case circuits for particular CA updating rules. Additionally, our four-key system that provides unique identities to cells in a local group is expensive in terms of number of species, requiring four separate sets of transmitter modules and eight separate sets of receiver modules in 1-dimensional space, and a more clever means for identifying neighboring cells may exist. However, it is unclear how to reduce the number of strands in our system by an order of magnitude.

Generally, the complexity of our circuits suggests that implementing even a simple 1-dimensional automaton would be challenging with current chemical computers. Constructing CA as complex as von Neumann's self-replicating automata is likely to be infeasible for the foreseeable future. It will therefore be important to ask whether there are more efficient models for spatial-

computing in which complex behaviors such as self-replication or healing can be designed as simply as possible. One starting point is to consider computation systems that do not require an explicit regular grid, such as Petri nets[20], Lindenmayer systems[13], or graph automata[36,33], and un-clocked systems such as asynchronous CA[18].

More generally, we might ask not only how to perform molecular computation using space as a medium, but how to construct a scalable architecture for computing appropriate responses of a material to stimuli that are presented across space and time. Patterns generated by CA could act as blueprints, encoding dynamic information spatially. By constructing CA in chemical networks, it may be possible to use this information to coordinate the behavior of intelligent programmable materials. Biological cells, tissues, and synthetic nanostructures could potentially respond to local instructions released by an embedded chemical automaton. CA could endow these physical systems with unique properties, creating artificial structures that heal, self-replicate and evolve.

Acknowledgements

The authors would like to thank Paul Rothemund, Damien Woods, Josh Fern, John Zenk, and the anonymous referees for insightful reading and comments. This work was supported by NSF-CCF-1161941 and a grant to the Turing Centenary Project by the John Templeton Foundation.

References

1. Allen, P.B., Chen, X., Ellington, A. D. "Spatial control of DNA reaction networks by DNA sequence." *Molecules* 17, 13, 390–13, 402 (2012)
2. Baker, M.D., Wolanin, P. M., Stock, J. B. "Signal transduction in bacterial chemotaxis. *Bioessays* 28 (1), 9–22 (2006)
3. Bánsági, T., Vanag, V.K., Epstein, I.R. "Tomography of reaction-diffusion microemulsions reveals three-dimensional Turing patterns." *Science* 331(6022), 1309–1312 (2011)
4. Chen, Y., Dalchau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D., Seelig, G. "Programmable chemical controllers made from DNA." *Nat. Nanotechnol.* 8(10), 755–762 (2013)
5. Chirieleison, S.M., Allen, P.B., Simpson, Z.B., Ellington, A.D., Chen, X. "Pattern transformation with DNA circuits." *Nature Chem.* 5, 1000–1005 (2013)
6. Codd, E. F. "Cellular automata." *Academic Press, Inc.*, SanDiego (1968)
7. Cook, M. "Universality in elementary cellular automata." *Complex Systems* 15(1), 1–40 (2004)
8. Du, Y., Lo, E., Ali, S., Khademhosseini, A. "Directed assembly of cell-laden microgels for fabrication of 3D tissue constructs." *P. Natl. Acad. Sci.* 105(28), 9522–9527 (2008)

9. Gács, P. "Reliable cellular automata with self-organization." *J. Stat. Phys.* 103(1/2), 45–267 (2001)
10. Gács, P., Reif, J. "A simple three-dimensional real-time reliable cellular array." *J. Comput. Syst. Sci.* 36(2), 125–147 (1988)
11. Greenfield, D., McEvoy, A. L., Shroff, H., Crooks, G. E., Wingreen, N. S., Betzig, E., Liphardt, J. "Self-organization of the *Escherichia coli* chemotaxis network imaged with super-resolution light microscopy." *PLoS Biol.* 7(6) (2009)
12. Langton, C. G. "Self-reproduction in cellular automata." *Physica D* 10.1, 135–144 (1984)
13. Lindenmayer, A. "Mathematical models for cellular interactions in development I. filaments with one-sided inputs." *J. Theor. Biol.* 18(3), 280–299 (1968)
14. Lukacs, G., Haggie, P., Seksek, O., Lechardeur, D., Verkman, N.F.A. "Size-dependent DNA mobility in cytoplasm and nucleus." *J. Biol. Chem.* 275(1625) (2000)
15. Montagne, K., Plasson, R., Sakai, Y., Fujii, T., Rondelez, Y. "Programming an in vitro DNA oscillator using a molecular networking strategy." *MOLSYSBIO* 7(1) (2011)
16. Murray, J. D. "Mathematical Biology II: Spatial Models and Biomedical Applications" 3rd edn. Springer, NewYork (2003)
17. Neary, T., Woods, D. "P-completeness of cellular automaton rule 110." *LNCS* 4051(132-143) (2006)
18. Nehaniv, C.L. "Asynchronous automata networks can emulate any synchronous automata network." *International Journal of Algebra and Computation* 14(05), 719–739 (2004)
19. von Neumann, J., A. W. Burks, E. "The Theory of Self-Reproducing Automata." University of Illinois Press, Urbana (1966)
20. Peterson, J. L. "Petri net theory and the modeling of systems." Prentice Hall, Englewood Cliffs, New Jersey (1981)
21. Qian, L., Soloveichik, D., Winfree, E. "Efficient Turing-universal computation with DNA polymers." *DNA computing and molecular programming* pp. 123–140 (2011)
22. Qian, L., Winfree, E. "Scaling up digital circuit computation with DNA strand displacement." *Science* 332 (6034), 1196–1201 (2011)
23. Qian, L., Winfree, E. "A simple DNA gate motif for synthesizing large-scale circuits." *J. R. Soc. Interface* 8(62),1281–1297 (2011)
24. Ruiza, S. A., Chen, C. S. "Microcontact printing: A tool to pattern." *Soft Matter* 3, 168–177 (2007)
25. Sayama, H. "A new structurally dissolvable self-reproducing loop evolving in a simple cellular automata space." *Artificial Life* 5.4, 343–365 (1999)
26. Scalise, D., Schulman, R. "Designing modular reaction-diffusion programs for complex pattern formation." *Technology* 2(01), 55–66 (2014)
27. Seelig, G., Soloveichik, D., Zhang, D. Y., Winfree, E. "Enzyme-free nucleic acid logic circuits." *Science* 314, 1585–1588 (2006)
28. Smith, D. E., Perkins, T. T., Chu, S. "Dynamical scaling of DNA diffusion coefficients." *Macromolecules* 29(4), 1372–1373 (1996)
29. Soloveichik, D., Cook, M., Winfree, E., Bruck, J. "Computation with finite stochastic chemical reaction networks." *Natural Computing* 7(4), 615–633 (2008)
30. Soloveichik, D., Seelig, G., Winfree, E.: DNA as a universal substrate for chemical kinetics. P. Natl. Acad. Sci. 107(12), 5393–5398 (2010)
31. Steinbock, O., Kettunen, P., Showalter, K. "Chemical wave logic gates." *The Journal of Physical Chemistry* 100(49), 18, 970–18, 975(1996)
32. Stellwagen, E., Lu, Y., Stellwagen, N. "Unified description of electrophoresis and diffusion for DNA and other polyions." *Biochemistry* 42(11745) (2003)
33. Tomita, K., Kurokawa, H., Murata, S. "Graph automata: natural expression of self-reproduction." *Physica D: Nonlinear Phenomena* 171(4), 197–210 (2002)
34. Ágota Tóth, Showalter, K. "Logic gates in excitable media." *The Journal of Chemical Physics* 103(6), 2058–2066 (1995)
35. Turing, A.M. "The chemical basis of morphogenesis." *Phil. T. Roy. Soc.* B237, 37–72(1952)
36. Wu, A., Rosenfeld, A. "Cellular graph automata. I. basic concepts, graph property measurement, closure properties. Information and Control" 42(3), 305–329 (1979)
37. Zhang, D. Y., Winfree, E. "Control of DNA strand displacement kinetics using toehold exchange." *J. Am. Chem. Soc.* 131(47), 17, 30317, 314 (2009)

6 | Supplemental Information: Emulating Cellular Automata in Chemical Reaction-Diffusion Networks

6.1 | Rule 110 Chemical Reactions (Synonymous with Section 6.2)

This section describes the complete set of abstract chemical reactions that govern our Rule 110 chemical automaton. These chemical reactions can be converted into a set of coupled partial differential equations and solved to observe how the system behaves over time (see Appendix B). Figure 9 shows a detail of the circuit from Figure 7 for a Rule 110 automaton, with each chemical species labeled.

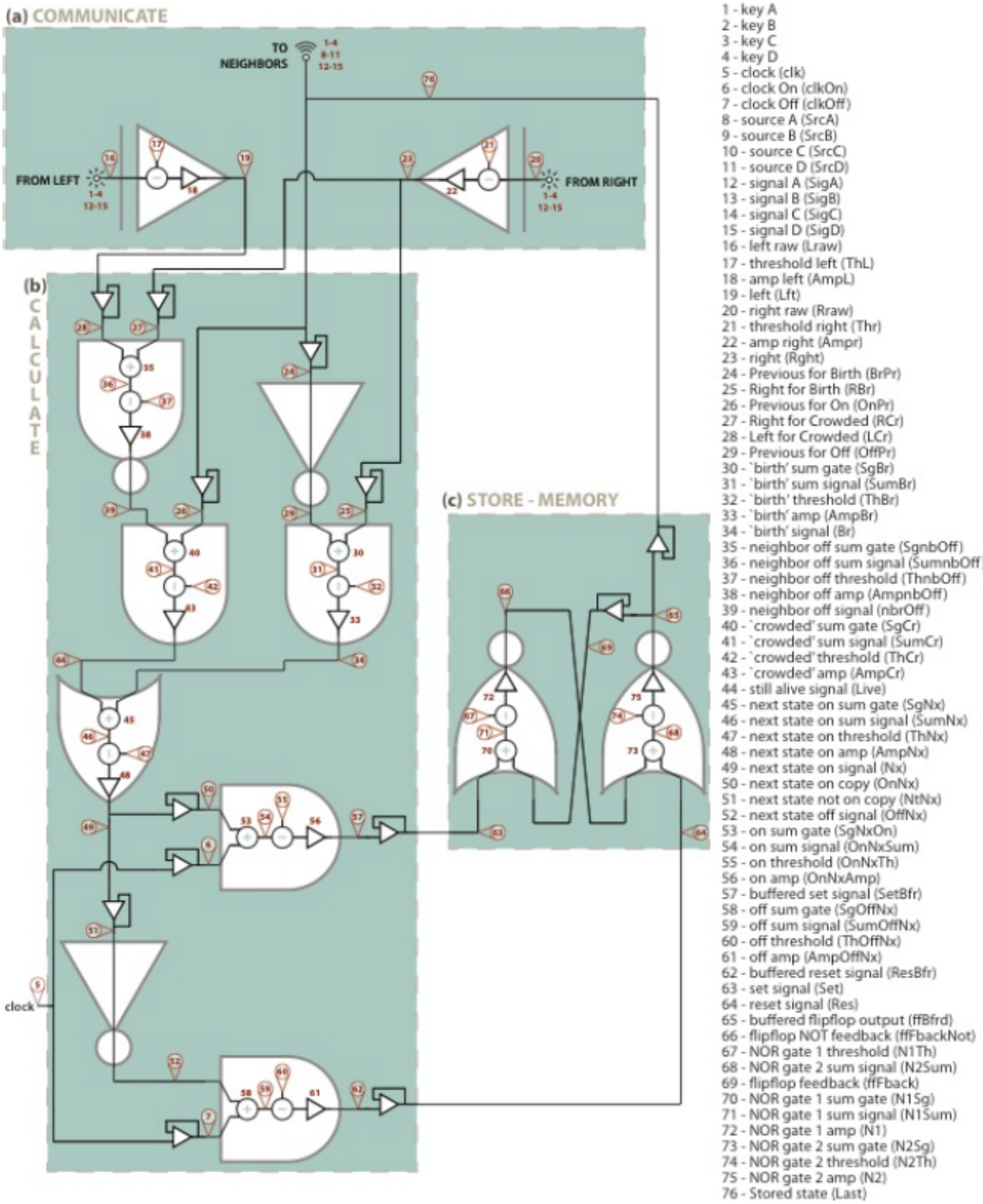

Figure 6.1 Chemical reaction-diffusion circuit for Rule 110. This is a detailed version of the circuit outlined in Fig. 7, using the modules defined in Fig. 2. Species are labeled in red by their equation numbers

from Appendix section B., with species names and abbreviations to the right.

1. Communication Stage
   (a) Broadcasting:

$$SrcA + keyA + Last \xrightarrow{k_B} SrcA + SigA + keyA + Last \quad (A.1)$$
$$SigA \xrightarrow{k_{Bd}} waste \quad (A.2)$$
$$SrcB + keyB + Last \xrightarrow{k_B} SrcB + SigB + keyB + Last \quad (A.3)$$
$$SigB \xrightarrow{k_{Bd}} waste \quad (A.4)$$
$$SrcC + keyC + Last \xrightarrow{k_B} SrcC + SigC + keyC + Last \quad (A.5)$$
$$SigC \xrightarrow{k_{Bd}} waste \quad (A.6)$$
$$SrcD + keyD + Last \xrightarrow{k_B} SrcD + SigD + keyD + Last \quad (A.7)$$
$$SigD \xrightarrow{k_{Bd}} waste \quad (A.8)$$

   (a) Receiving and processing left-hand signal:

$$SigD + keyA \xrightarrow{k_F} SigD + Lraw + keyA \quad (A.9)$$
$$SigA + keyB \xrightarrow{k_F} SigA + Lraw + keyB \quad (A.10)$$
$$SigB + keyC \xrightarrow{k_F} SigB + Lraw + keyC \quad (A.11)$$
$$SigC + keyD \xrightarrow{k_F} SigC + Lraw + keyD \quad (A.12)$$
$$Lraw \xrightarrow{4*k_x} waste \quad (A.13)$$
$$source \xrightarrow{c_{recTh}*k_P} ThL \xrightarrow{k_F} waste \quad (A.14)$$
$$source \xrightarrow{k_P} AmpL \xrightarrow{k_F} waste \quad (A.15)$$
$$Lraw + ThL \xrightarrow{k_T} waste \quad (A.16)$$
$$Lraw + AmpL \xrightarrow{k_L} Lraw + Lft \quad (A.17)$$
$$Lft \xrightarrow{k_F} waste \quad (A.18)$$

   (a) Receiving and processing right-hand signal:

$$SigB + keyA \xrightarrow{k_F} SigB + Rraw + keyA \quad (A.19)$$
$$SigC + keyB \xrightarrow{k_F} SigC + Rraw + keyB \quad (A.20)$$
$$SigD + keyC \xrightarrow{k_F} SigD + Rraw + keyC \quad (A.21)$$
$$SigA + keyD \xrightarrow{k_F} SigA + Rraw + keyD \quad (A.22)$$
$$Rraw \xrightarrow{4*k_x} waste \quad (A.23)$$
$$source \xrightarrow{c_{recTh}*k_P} Thr \xrightarrow{k_F} waste \quad (A.24)$$
$$source \xrightarrow{k_P} Ampr \xrightarrow{k_F} waste \quad (A.25)$$
$$Rraw + Thr \xrightarrow{k_T} waste \quad (A.26)$$
$$Rraw + Ampr \xrightarrow{k_L} Rraw + Rght \quad (A.27)$$
$$Rght \xrightarrow{k_F} waste \quad (A.28)$$

2. Calculation Stage
   (a) Copy left, right and previous time step (pr) signals (multiple gates operate on each):

$$Last \xrightarrow{k_F} Last + BrPr \quad (A.29)$$
$$BrPr \xrightarrow{k_F} waste \quad (A.30)$$
$$Rght \xrightarrow{k_F} Rght + RBr \quad (A.31)$$
$$RBr \xrightarrow{k_F} waste \quad (A.32)$$
$$Last \xrightarrow{k_F} Last + OnPr \quad (A.33)$$
$$OnPr \xrightarrow{k_F} waste \quad (A.34)$$
$$Rght \xrightarrow{k_F} Rght + RCr \quad (A.35)$$
$$RCr \xrightarrow{k_F} waste \quad (A.36)$$
$$Lft \xrightarrow{k_F} Lft + LCr \quad (A.37)$$
$$LCr \xrightarrow{k_F} waste \quad (A.38)$$
$$source \xrightarrow{k_P} OffPr \quad (A.39)$$
$$OffPr \xrightarrow{k_F} waste \quad (A.40)$$
$$BrPr + OffPr \xrightarrow{k_T} waste \quad (A.41)$$

$$source \xrightarrow{2*k_P} SgBr \xrightarrow{k_F} waste \tag{A.42}$$
$$OffPr + SgBr \xrightarrow{k_L} SumBr \xrightarrow{k_F} waste \tag{A.43}$$
$$RBr + SgBr \xrightarrow{k_L} SumBr \tag{A.44}$$
$$source \xrightarrow{1.35*k_P} ThBr \xrightarrow{k_F} waste \tag{A.45}$$
$$source \xrightarrow{k_P} AmpBr \xrightarrow{k_F} waste \tag{A.46}$$
$$SumBr + ThBr \xrightarrow{k_T} waste \tag{A.47}$$
$$SumBr + AmpBr \xrightarrow{k_L} SumBr + Br \tag{A.48}$$
$$Br \xrightarrow{k_F} waste \tag{A.49}$$

(a) Boolean logic for no death condition (*on* and at least one neighbor *off*): stay on:

$$source \xrightarrow{2*k_P} SgnbOff \xrightarrow{k_F} waste \tag{A.50}$$
$$RCr + SgnbOff \xrightarrow{k_L} SumnbOff \xrightarrow{k_F} waste \tag{A.51}$$
$$LCr + SgnbOff \xrightarrow{k_L} SumnbOff \tag{A.52}$$
$$source \xrightarrow{1.35*k_P} ThnbOff \xrightarrow{k_F} waste \tag{A.53}$$
$$source \xrightarrow{k_P} AmpnbOff \xrightarrow{k_F} waste \tag{A.54}$$
$$SumnbOff + ThnbOff \xrightarrow{k_T} waste \tag{A.55}$$
$$ThnbOff + AmpnbOff \xrightarrow{k_L} ThnbOff + nbrOff \tag{A.56}$$
$$nbrOff \xrightarrow{k_F} waste \tag{A.57}$$
$$source \xrightarrow{2*k_P} SgCr \xrightarrow{k_F} waste \tag{A.58}$$
$$OnPr + SgCr \xrightarrow{k_L} SumCr \xrightarrow{k_F} waste \tag{A.59}$$
$$nbrOff + SgCr \xrightarrow{k_L} SumCr \tag{A.60}$$
$$source \xrightarrow{1.35*k_P} ThCr \xrightarrow{k_F} waste \tag{A.61}$$
$$source \xrightarrow{k_P} AmpCr \xrightarrow{k_F} waste \tag{A.62}$$
$$SumCr + ThCr \xrightarrow{k_T} waste \tag{A.63}$$
$$SumCr + AmpCr \xrightarrow{k_L} SumCr + Live \tag{A.64}$$
$$Live \xrightarrow{k_F} waste \tag{A.65}$$

(a) Boolean logic to determine if next state is *on*:

$$source \xrightarrow{2*k_P} SgNx \xrightarrow{k_F} waste \tag{A.66}$$
$$Live + SgNx \xrightarrow{k_L} SumNx \xrightarrow{k_F} waste \tag{A.67}$$
$$Br + SgNx \xrightarrow{k_L} SumNx \tag{A.68}$$
$$source \xrightarrow{0.65*k_P} ThNx \xrightarrow{k_F} waste \tag{A.69}$$
$$source \xrightarrow{k_P} AmpNx \xrightarrow{k_F} waste \tag{A.70}$$
$$SumNx + ThNx \xrightarrow{k_T} waste \tag{A.71}$$
$$SumNx + AmpNx \xrightarrow{k_L} SumNx + Nx \tag{A.72}$$
$$Nx \xrightarrow{k_F} waste \tag{A.73}$$
$$Nx \xrightarrow{k_F} Nx + OnNx \tag{A.74}$$
$$OnNx \xrightarrow{k_F} waste \tag{A.75}$$

(a) Boolean logic to determine if next state is *off*:

$$Nx \xrightarrow{k_F} Nx + NtNx \tag{A.76}$$
$$NtNx \xrightarrow{k_F} waste \tag{A.77}$$
$$source \xrightarrow{k_P} OffNx \xrightarrow{k_F} waste \tag{A.78}$$
$$NtNx + OffNx \xrightarrow{k_T} waste \tag{A.79}$$

$$clk \xrightarrow{k_F} clk + clkOn \tag{A.80}$$
$$clkOn \xrightarrow{k_F} waste \tag{A.81}$$
$$clk \xrightarrow{k_F} clk + clkOff \tag{A.82}$$
$$clkOff \xrightarrow{k_F} waste \tag{A.83}$$
$$source \xrightarrow{2*k_P} SgNxOn \xrightarrow{k_F} waste \tag{A.84}$$
$$OnNx + SgNxOn \xrightarrow{k_L} OnNxSum \xrightarrow{k_F} waste \tag{A.85}$$
$$clkOn + SgNxOn \xrightarrow{k_L} OnNxSum \tag{A.86}$$
$$source \xrightarrow{1.35*k_P} OnNxTh \xrightarrow{k_F} waste \tag{A.87}$$
$$source \xrightarrow{k_P} OnNxAmp \xrightarrow{k_F} waste \tag{A.88}$$
$$OnNxSum + OnNxTh \xrightarrow{k_T} waste \tag{A.89}$$
$$OnNxSum + OnNxAmp \xrightarrow{k_L} OnNxSum + SetBfr \tag{A.90}$$
$$SetBfr \xrightarrow{k_F} waste \tag{A.91}$$
$$source \xrightarrow{2*k_P} SgOffNx \xrightarrow{k_F} waste \tag{A.92}$$
$$OffNx + SgOffNx \xrightarrow{k_L} SumOffNx \xrightarrow{k_F} waste \tag{A.93}$$
$$clkOff + SgOffNx \xrightarrow{k_L} SumOffNx \tag{A.94}$$
$$source \xrightarrow{1.35*k_P} ThOffNx \xrightarrow{k_F} waste \tag{A.95}$$
$$source \xrightarrow{k_P} AmpOffNx \xrightarrow{k_F} waste \tag{A.96}$$
$$SumOffNx + ThOffNx \xrightarrow{k_T} waste \tag{A.97}$$
$$SumOffNx + AmpOffNx \xrightarrow{k_L} SumOffNx + ResBfr \tag{A.98}$$
$$ResBfr \xrightarrow{k_F} waste \tag{A.99}$$

3. Storage Stage

(a) Copies of Set/Res signals:

$$SetBfr \xrightarrow{k_F} SetBfr + Set \tag{A.100}$$
$$Set \xrightarrow{k_F} waste \tag{A.101}$$
$$ResBfr \xrightarrow{k_F} ResBfr + Res \tag{A.102}$$

(a) Flip- op module:

$$Res \xrightarrow{k_F} waste \tag{A.103}$$
$$ffBfrd \xrightarrow{k_F} ffBfrd + ffFback \tag{A.104}$$
$$ffFback \xrightarrow{k_F} waste \tag{A.105}$$
$$source \xrightarrow{2*k_P} N1Sg \xrightarrow{k_F} waste \tag{A.106}$$
$$Set + N1Sg \xrightarrow{k_L} N1Sum \xrightarrow{k_F} waste \tag{A.107}$$
$$ffFback + N1Sg \xrightarrow{k_L} N1Sum \tag{A.108}$$
$$source \xrightarrow{0.65*k_P} N1Th \xrightarrow{k_F} waste \tag{A.109}$$
$$source \xrightarrow{k_P} N1 \xrightarrow{k_F} waste \tag{A.110}$$
$$N1Sum + N1Th \xrightarrow{k_T} waste \tag{A.111}$$
$$N1Th + N1 \xrightarrow{k_L} N1Th + ffFbackNot \tag{A.112}$$
$$ffFbackNot \xrightarrow{k_F} waste \tag{A.113}$$
$$source \xrightarrow{2*k_P} N2Sg \xrightarrow{k_F} waste \tag{A.114}$$
$$Res + N2Sg \xrightarrow{k_L} N2Sum \xrightarrow{k_F} waste \tag{A.115}$$
$$ffFbackNot + N2Sg \xrightarrow{k_L} N2Sum \tag{A.116}$$
$$source \xrightarrow{0.65*k_P} N2Th \xrightarrow{k_F} waste \tag{A.117}$$
$$source \xrightarrow{k_P} N2 \xrightarrow{k_F} waste \tag{A.118}$$
$$N2Sum + N2Th \xrightarrow{k_T} waste \tag{A.119}$$
$$N2Th + N2 \xrightarrow{k_L} N2Th + ffBfrd \tag{A.120}$$
$$ffBfrd \xrightarrow{k_F} waste \tag{A.121}$$
$$ffBfrd \xrightarrow{k_F} ffBfrd + Last \tag{A.122}$$
$$Last \xrightarrow{k_F} waste \tag{A.123}$$

Constants:

$$xMax = 64 \qquad k_p = 0.002\,\mu\text{M}\text{s}^{-1} \qquad k_{Bd} = 0.00002\text{s}^{-1}$$
$$D = 0.00015\,\text{mm}^2\text{s}^{-1} \qquad k_d = 0.002\,\mu\text{s}^{-1} \qquad c_{recTh} = 0.5\,\mu\text{M}$$
$$k_T = 20\,\mu\text{M}^{-1}\text{s}^{-1} \qquad k_x = 0.002\,\mu\text{M}^{-1}\text{s}^{-1} \qquad clkPeriod = 2 * 24 * 3600\,\text{s}$$
$$k_L = 0.2\,\mu\text{M}^{-1}\text{s}^{-1} \qquad k_B = 0.0002\,\mu\text{M}^{-2}\text{s}^{-1} \qquad clkDuty = .5 * 3600\,\text{s}$$

1. External Signals:
   (a) Keys (in μM):

$$Key_A(t,x) = \begin{cases} 1/(1+Exp[-25*(Mod(x,16)-1)]) & : Mod(x,16) \leq 2 \\ 1-1/(1+Exp[-25*(Mod(x,16)-3)]) & : otherwise. \end{cases} \quad \text{(B.1)}$$

$$Key_B(t,x) = \begin{cases} 1/(1+Exp[-25*(Mod(x,16)-5)]) & : Mod(x,16) \leq 6 \\ 1-1/(1+Exp[-25*(Mod(x,16)-7)]) & : otherwise. \end{cases} \quad \text{(B.2)}$$

$$Key_C(t,x) = \begin{cases} 1/(1+Exp[-25(Mod(x,16)-9)]) & : mod(x,16) \leq 10 \\ 1-1/(1+Exp[-25(Mod(x,16)-11)]) & : otherwise. \end{cases} \quad \text{(B.3)}$$

$$Key_D(t,x) = \begin{cases} 1/(1+Exp[-25(Mod(x,16)-13)]) & : mod(x,16) \leq 14 \\ 1-1/(1+Exp[-25(Mod(x,16)-15)]) & : otherwise. \end{cases} \quad \text{(B.4)}$$

   (b) Clock:

$$clk(t,x) = \begin{cases} 1\,\mu\text{M} & : t < 4000 \\ 1\,\mu\text{M} & : Mod(t, clkPeriod) < clkDuty \\ 0 & : otherwise. \end{cases} \quad \text{(B.5)}$$

$$\frac{\partial clk_{On}(t,x)}{\partial t} = D\nabla^2 clk_{On}(t,x) - k_d * clk_{On}(t,x) + k_d * clk(t,x) - k_L * clk_{On}(t,x) * Sg_{nxOn}(t,x) \quad \text{(B.6)}$$

$$\frac{\partial clk_{Off}(t,x)}{\partial t} = D\nabla^2 clk_{Off}(t,x) - k_d * clk_{Off}(t,x) + k_d * clk(t,x) - k_L * clk_{Off}(t,x) * Sg_{OffNx}(t,x) \quad \text{(B.7)}$$

2. Communication Stage:
   (a) Broadcast Modules (in μM):

$$Src_A(t,x) = 1 \quad \text{(B.8)} \qquad Src_B(t,x) = 1 \quad \text{(B.9)} \qquad Src_C(t,x) = 1 \quad \text{(B.10)} \qquad Src_D(t,x) = 1 \quad \text{(B.11)}$$

   (b) Broadcast signals, note initial concentration of Sig_A triggers initial *on* cell:

$$\frac{\partial Sig_A(t,x)}{\partial t} = D\nabla^2 Sig_A(t,x) + k_B Src_A(t,x) Key_A(t,x) Last(t,x) - k_{Bd} Sig_A(t,x) \quad \text{(B.12)}$$

$$Sig_A(t=0,x) = \begin{cases} 2\,\mu\text{M} & : x_{max} - 8 < x < x_{max} - 0.1 \\ 0 & : otherwise. \end{cases}$$

$$\frac{\partial Sig_B(t,x)}{\partial t} = D\nabla^2 Sig_B(t,x) + k_B Src_B(t,x) Key_B(t,x) Last(t,x) - k_{Bd} Sig_B(t,x) \quad \text{(B.13)}$$

$$\frac{\partial Sig_C(t,x)}{\partial t} = D\nabla^2 Sig_C(t,x) + k_B Src_C(t,x) Key_C(t,x) Last(t,x) - k_{Bd} Sig_C(t,x) \quad \text{(B.14)}$$

$$\frac{\partial Sig_D(t,x)}{\partial t} = D\nabla^2 Sig_D(t,x) + k_B Src_D(t,x) Key_D(t,x) Last(t,x) - k_{Bd} Sig_D(t,x) \quad \text{(B.15)}$$

   (c) Receiving and processing left-hand neighbor signal:

$$\frac{\partial L_{raw}(t,x)}{\partial t} = D\nabla^2 L_{raw}(t,x) - 4k_d L_{raw}(t,x) - k_T L_{raw}(t,x) Th_l(t,x) + k_x Sig_D(t,x) Key_A(t,x) + k_x Sig_A(t,x) Key_B(t,x) + k_x Sig_B(t,x) Key_C(t,x) + k_x Sig_C(t,x) Key_D(t,x) \quad \text{(B.16)}$$

$$\frac{\partial Th_l(t,x)}{\partial t} = D\nabla^2 Th_l(t,x) + c_{recTh} k_p - k_d Th_l(t,x) - k_T L_{raw}(t,x) Th_l(t,x) \quad \text{(B.17)}$$

$$\frac{\partial Amp_l(t,x)}{\partial t} = D\nabla^2 Amp_l(t,x) + k_p - k_d Amp_l(t,x) - k_L L_{raw}(t,x) Amp_l(t,x) \quad \text{(B.18)}$$

$$\frac{\partial Lft(t,x)}{\partial t} = D\nabla^2 Lft(t,x) - k_d Lft(t,x) + k_L L_{raw}(t,x) Amp_l(t,x) \quad \text{(B.19)}$$

   (d) Receiving and processing right-hand neighbor signal:

$$\frac{\partial R_{raw}(t,x)}{\partial t} = D\nabla^2 R_{raw}(t,x) - 4k_d R_{raw}(t,x) - k_T R_{raw}(t,x) Th_r(t,x) + k_x Sig_(t,x) Key_A(t,x) + k_x Sig_C(t,x) Key_B(t,x) + k_x Sig_D(t,x) Key_C(t,x) + k_x Sig_A(t,x) Key_D(t,x) \quad \text{(B.20)}$$

$$\frac{\partial Th_r(t,x)}{\partial t} = D\nabla^2 Th_r(t,x) + c_{recTh} k_p - k_d Th_r(t,x) - k_T R_{raw}(t,x) Th_r(t,x) \tag{B.21}$$

$$\frac{\partial Amp_{rt}(t,x)}{\partial t} = D\nabla^2 Amp_{rt}(t,x) + k_p - k_d Amp_{rt}(t,x) - k_L R_{raw}(t,x) Amp_{rt}(t,x) \tag{B.22}$$

$$\frac{\partial Rght(t,x)}{\partial t} = D\nabla^2 Rght(t,x) - k_d Rght(t,x) + k_L R_{raw}(t,x) Amp_{rt}(t,x) \tag{B.23}$$

3. Calculation Stage:

(a) Copy left, right and previous time step (pr) signals (multiple gates operate on each)

$$\frac{\partial Br_{pr}(t,x)}{\partial t} = D\nabla^2 Br_{pr}(t,x) - k_d Br_{pr}(t,x) + k_d Last(t,x) - k_T Br_{pr}(t,x) Off_{pr}(t,x) \tag{B.24}$$

$$\frac{\partial R_{br}(t,x)}{\partial t} = D\nabla^2 R_{br}(t,x) - k_d R_{br}(t,x) + k_d Rght(t,x) - k_L R_{br}(t,x) Sg_{br}(t,x) \tag{B.25}$$

$$\frac{\partial On_{pr}(t,x)}{\partial t} = D\nabla^2 On_{pr}(t,x) - k_d On_{pr}(t,x) + k_d Last(t,x) - k_L On_{pr}(t,x) Sg_{cr}(t,x) \tag{B.26}$$

$$\frac{\partial R_{cr}(t,x)}{\partial t} = D\nabla^2 R_{cr}(t,x) - k_d R_{cr}(t,x) + k_d Rght(t,x) - k_L R_{cr}(t,x) Sg_{nbOff}(t,x) \tag{B.27}$$

$$\frac{\partial L_{cr}(t,x)}{\partial t} = D\nabla^2 L_{cr}(t,x) - k_d L_{cr}(t,x) + k_d Lft(t,x) - k_L L_{cr}(t,x) Sg_{nbOff}(t,x) \tag{B.28}$$

$$\frac{\partial Off_{pr}(t,x)}{\partial t} = D\nabla^2 Off_{pr}(t,x) + k_p - k_d Off_{pr}(t,x) - {}_T Br_{pr}(t,x) Off_{pr}(t,x) - k_L Off_{pr}(t,x) Sg_{br}(t,x) \tag{B.29}$$

(b) Boolean logic for Br (birth) condition: *off* to *on* transition

$$\frac{\partial Sg_{br}(t,x)}{\partial t} = D\nabla^2 Sg_{br}(t,x) + 2k_p - k_d Sg_{br}(t,x) - k_L Off_{pr}(t,x) Sg_{br}(t,x) - k_L R_{br}(t,x) Sg_{br}(t,x) \tag{B.30}$$

$$\begin{aligned}\frac{\partial Sum_{br}(t,x)}{\partial t} &= D\nabla^2 Sum_{br}(t,x) - k_d Sum_{br}(t,x) + k_L Off_{pr}(t,x) Sg_{br}(t,x) \\ &\quad + k_L R_{br}(t,x) Sg_{br}(t,x) - k_T Sum_{br}(t,x) Th_{br}(t,x)\end{aligned} \tag{B.31}$$

$$\frac{\partial Th_{br}(t,x)}{\partial t} = D\nabla^2 Th_{br}(t,x) + 1.35k_p - k_d Th_{br}(t,x) - k_T Sum_{br}(t,x) Th_{br}(t,x) \tag{B.32}$$

$$\frac{\partial Amp_{br}(t,x)}{\partial t} = D\nabla^2 Amp_{br}(t,x) + k_p - k_d Amp_{br}(t,x) - k_L Sum_{br}(t,x) Amp_{br}(t,x) \tag{B.33}$$

$$\frac{\partial Br(t,x)}{\partial t} = D\nabla^2 Br(t,x) - k_d Br(t,x) + k_L Sum_{br}(t,x) Amp_{br}(t,x) - k_L Br(t,x) Sg_{nx}(t,x) \tag{B.34}$$

(c) Boolean logic for no death condition (*on* and at least one neighbor *off*): stay *on*

$$\begin{aligned}\frac{\partial Sg_{nbOff}(t,x)}{\partial t} &= D\nabla^2 Sg_{nbOff}(t,x) + 2k_p - k_d Sg_{nbOff}(t,x) - k_L R_{cr}(t,x) Sg_{nbOff}(t,x) \\ &\quad - k_L L_{cr}(t,x) Sg_{nbOff}(t,x)\end{aligned} \tag{B.35}$$

$$\begin{aligned}\frac{\partial Sum_{nbOff}(t,x)}{\partial t} &= D\nabla^2 Sum_{nbOff}(t,x) - k_d Sum_{nbOff}(t,x) + k_L R_{cr}(t,x) Sg_{nbOff}(t,x) \\ &\quad + k_L L_{cr}(t,x) Sg_{nbOff}(t,x) - k_T Sum_{nbOff}(t,x) Th_{nbOff}(t,x)\end{aligned} \tag{B.36}$$

$$\frac{\partial Th_{nbOff}(t,x)}{\partial t} = D\nabla^2 Th_{nbOff}(t,x) + 1.35k_p - k_d Th_{nbOff}(t,x) - k_T Sum_{nbOff}(t,x) Th_{nbOff}(t,x) \tag{B.37}$$

$$\frac{\partial Amp_{nbOff}(t,x)}{\partial t} = D\nabla^2 Amp_{nbOff}(t,x) + k_p - k_d Amp_{nbOff}(t,x) - k_L Th_{nbOff}(t,x) Amp_{nbOff}(t,x) \tag{B.38}$$

$$\begin{aligned}\frac{\partial nbrOff(t,x)}{\partial t} &= D\nabla^2 nbrOff(t,x) - k_d nbrOff(t,x) + k_L Th_{nbOff}(t,x) Amp_{nbOff} y(t,x) \\ &\quad - k_L nbrOff(t,x) Sg_{cr}(t,x)\end{aligned} \tag{B.39}$$

$$\frac{\partial Sg_{cr}(t,x)}{\partial t} = D\nabla^2 Sg_{cr}(t,x) + 2k_p - k_d Sg_{cr}(t,x) - k_L On_{pr}(t,x) Sg_{cr}(t,x) - k_L nbrOff(t,x) Sg_{cr}(t,x) \tag{B.40}$$

$$\begin{aligned}\frac{\partial Sum_{cr}(t,x)}{\partial t} &= D\nabla^2 Sum_{cr}(t,x) - k_d Sum_{cr}(t,x) + k_L On_{pr}(t,x) Sg_{cr}(t,x) \\ &\quad + k_L nbrOff(t,x) Sg_{cr}(t,x) - k_T Sum_{cr}(t,x) Th_{cr}(t,x)\end{aligned} \tag{B.41}$$

$$\frac{\partial Th_{cr}(t,x)}{\partial t} = D\nabla^2 Th_{cr}(t,x) + 1.35k_p - k_d Th_{cr}(t,x) - k_T Sum_{cr}(t,x) Th_{cr}(t,x) \tag{B.42}$$

$$\frac{\partial Amp_{cr}(t,x)}{\partial t} = D\nabla^2 Amp_{cr}(t,x) + k_p - k_d Amp_{cr}(t,x) - k_L Sum_{cr}(t,x) Amp_{cr}(t,x) \tag{B.43}$$

$$\frac{\partial Live(t,x)}{\partial t} = D\nabla^2 Live(t,x) - k_d Live(t,x) + k_L Sum_{cr}(t,x) Amp_{cr}(t,x) - k_L Live(t,x) Sg_{nx}(t,x) \tag{B.44}$$

(d) Boolean logic to determine if next state is *on*

$$\frac{\partial Sg_{nx}(t,x)}{\partial t} = D\nabla^2 Sg_{nx}(t,x) + 2k_p - k_d Sg_{nx}(t,x) - k_L Live(t,x) Sg_{nx}(t,x) - k_L Br(t,x) Sg_{nx}(t,x) \tag{B.45}$$

$$\begin{aligned}\frac{\partial Sum_{nx}(t,x)}{\partial t} &= D\nabla^2 Sum_{nx}(t,x) - k_d Sum_{nx}(t,x) + k_L Live(t,x) Sg_{nx}(t,x) + k_L Br(t,x) Sg_{nx}(t,x) \\ &\quad - k_T Sum_{nx}(t,x) Th_{nx}(t,x)\end{aligned} \tag{B.46}$$

$$\frac{\partial Th_{nx}(t,x)}{\partial t} = D\nabla^2 Th_{nx}(t,x) + 0.65k_p - k_d Th_{nx}(t,x) - k_T Sum_{nx}(t,x) Th_{nx}(t,x) \tag{B.47}$$

$$\frac{\partial Amp_{nx}(t,x)}{\partial t} = D\nabla^2 Amp_{nx}(t,x) + k_p - k_d Amp_{nx}(t,x) - k_L Sum_{nx}(t,x) Amp_{nx}(t,x) \tag{B.48}$$

$$\frac{\partial Nx(t,x)}{\partial t} = D\nabla^2 Nx(t,x) - k_d Nx(t,x) + k_L Sum_{nx}(t,x) Amp_{nx}(t,x) \tag{B.49}$$

$$\frac{\partial On_{nx}(t,x)}{\partial t} = D\nabla^2 On_{nx}(t,x) - k_d On_{nx}(t,x) + k_d Nx(t,x) - k_L On_{nx}(t,x) Sg_{nxOn}(t,x) \tag{B.50}$$

(e) Boolean logic to determine if next state is *off*

$$\frac{\partial Nt_{nx}(t,x)}{\partial t} = D\nabla^2 Nt_{nx}(t,x) - k_d Nt_{nx}(t,x) + k_d Nx(t,x) - k_T Nt_{nx}(t,x) Off_{nx}(t,x) \tag{B.51}$$

$$\frac{\partial Off_{nx}(t,x)}{\partial t} = D\nabla^2 Off_{nx}(t,x) + k_p - k_d Off_{nx}(t,x) - k_T Nt_{nx}(t,x) Off_{nx}(t,x) - k_L Off_{nx}(t,x) Sg_{OffNx}(t,x) \tag{B.52}$$

(f) Clocked synchronization gates

$$\frac{\partial Sg_{nxOn}(t,x)}{\partial t} = D\nabla^2 Sg_{nxOn}(t,x) + 2k_p - k_d Sg_{nxOn}(t,x) - k_L On_{nx}(t,x) Sg_{nxOn}(t,x) - k_L clk_{On}(t,x) Sg_{nxOn}(t,x) \quad \text{(B.53)}$$

$$\frac{\partial On_{nx}Sum(t,x)}{\partial t} = D\nabla^2 On_{nx}Sum(t,x) - k_d On_{nx}Sum(t,x) + k_L On_{nx}(t,x) Sg_{nxOn}(t,x) + k_L clk_{On}(t,x) Sg_{nxOn}(t,x) - k_T On_{nx}Sum(t,x) On_{nx}Th(t,x) \quad \text{(B.54)}$$

$$\frac{\partial On_{nx}Th(t,x)}{\partial t} = D\nabla^2 On_{nx}Th(t,x) + 1.35k_p - k_d On_{nx}Th(t,x) - k_T On_{nx}Sum(t,x) On_{nx}Th(t,x) \quad \text{(B.55)}$$

$$\frac{\partial On_{nx}Amp(t,x)}{\partial t} = D\nabla^2 On_{nx}Amp(t,x) + k_p - k_d On_{nx}Amp(t,x) - k_L On_{nx}Sum(t,x) On_{nx}Amp(t,x) \quad \text{(B.56)}$$

$$\frac{\partial Set_{Bfr}(t,x)}{\partial t} = D\nabla^2 Set_{Bfr}(t,x) - k_d Set_{Bfr}(t,x) + k_L On_{nx}Sum(t,x) On_{nx}Amp(t,x) \quad \text{(B.57)}$$

$$\frac{\partial Sg_{OffNx}(t,x)}{\partial t} = D\nabla^2 Sg_{OffNx}(t,x) + 2k_p - k_d Sg_{OffNx}(t,x) - k_L Off_{nx}(t,x) Sg_{OffNx}(t,x) - k_L clk_{Off}(t,x) Sg_{OffNx}(t,x) \quad \text{(B.58)}$$

$$\frac{\partial Sum_{OffNx}(t,x)}{\partial t} = D\nabla^2 Sum_{OffNx}(t,x) - k_d Sum_{OffNx}(t,x) + k_L Off_{nx}(t,x) Sg_{OffNx}(t,x) + k_L clk_{Off}(t,x) Sg_{OffNx}(t,x) - k_T Sum_{OffNx}(t,x) Th_{OffNx}(t,x) \quad \text{(B.59)}$$

$$\frac{\partial Th_{OffNx}(t,x)}{\partial t} = D\nabla^2 Th_{OffNx}(t,x) + 1.35k_p - k_d Th_{OffNx}(t,x) - k_T Sum_{OffNx}(t,x) Th_{OffNx}(t,x) \quad \text{(B.60)}$$

$$\frac{\partial Amp_{OffNx}(t,x)}{\partial t} = D\nabla^2 Amp_{OffNx}(t,x) + k_p - k_d Amp_{OffNx}(t,x) - k_L Sum_{OffNx}(t,x) Amp_{OffNx}(t,x) \quad \text{(B.61)}$$

$$\frac{\partial Res_{Bfr}(t,x)}{\partial t} = D\nabla^2 Res_{Bfr}(t,x) - k_d Res_{Bfr}(t,x) + k_L Sum_{OffNx}(t,x) Amp_{OffNx}(t,x) \quad \text{(B.62)}$$

Storage Stage

(a) Copies of Set/Res signals

$$\frac{\partial Set(t,x)}{\partial t} = D\nabla^2 Set(t,x) - k_d Set(t,x) + k_d Set_{Bfr}(t,x) - k_L Set(t,x) N1Sg(t,x) \quad \text{(B.63)}$$

$$\frac{\partial Res(t,x)}{\partial t} = D\nabla^2 Res(t,x) - k_d Res(t,x) + k_d Res_{Bfr}(t,x) - k_L Res(t,x) N2Sg(t,x) \quad \text{(B.64)}$$

(b) Flip- op module

$$\frac{\partial ffBfrd(t,x)}{\partial t} = D\nabla^2 ffBfrd(t,x) - k_d ffBfrd(t,x) + k_L N2Th(t,x) N2(t,x) \quad \text{(B.65)}$$

$$\frac{\partial ffFbackNot(t,x)}{\partial t} = D\nabla^2 ffFbackNot(t,x) - k_d ffFbackNot(t,x) + k_L N1Th(t,x) N1(t,x) - k_L ffFbackNot(t,x) N2Sg(t,x) \quad \text{(B.66)}$$

$$\frac{\partial N1Th(t,x)}{\partial t} = D\nabla^2 N1Th(t,x) + 0.65k_p - k_d N1Th(t,x) - k_T N1Sum(t,x) N1Th(t,x) \quad \text{(B.67)}$$

$$\frac{\partial N2Sum(t,x)}{\partial t} = D\nabla^2 N2Sum(t,x) - k_d N2Sum(t,x) + k_L Res(t,x) N2Sg(t,x) + k_L ffFbackNot(t,x) N2Sg(t,x) - k_T N2Sum(t,x) N2Th(t,x) \quad \text{(B.68)}$$

$$\frac{\partial ffFback(t,x)}{\partial t} = D\nabla^2 ffFback(t,x) - k_d ffFback(t,x) + k_d ffBfrd(t,x) - k_L ffFback(t,x) N1Sg(t,x) \quad \text{(B.69)}$$

$$\frac{\partial N1Sg(t,x)}{\partial t} = D\nabla^2 N1Sg(t,x) + 2k_p - k_d N1Sg(t,x) - k_L Set(t,x) N1Sg(t,x) - k_L ffFback(t,x) N1Sg(t,x) \quad \text{(B.70)}$$

$$\frac{\partial N1Sum(t,x)}{\partial t} = D\nabla^2 N1Sum(t,x) - k_d N1Sum(t,x) + k_L Set(t,x) N1Sg(t,x) + k_L ffFback(t,x) N1Sg(t,x) - k_T N1Sum(t,x) N1Th(t,x) \quad \text{(B.71)}$$

$$\frac{\partial N1(t,x)}{\partial t} = D\nabla^2 N1(t,x) + k_p - k_d N1(t,x) - k_L N1Th(t,x) N1(t,x) \quad \text{(B.72)}$$

$$\frac{\partial N2Sg(t,x)}{\partial t} = D\nabla^2 N2Sg(t,x) + 2k_p - k_d N2Sg(t,x) - k_L Res(t,x) N2Sg(t,x) - k_L ffFbackNot(t,x) N2Sg(t,x) \quad \text{(B.73)}$$

$$\frac{\partial N2Th(t,x)}{\partial t} = D\nabla^2 N2Th(t,x) + 0.65k_p - k_d N2Th(t,x) - k_T N2Sum(t,x) N2Th(t,x) \quad \text{(B.74)}$$

$$\frac{\partial N2(t,x)}{\partial t} = D\nabla^2 N2(t,x) + k_p - k_d N2(t,x) - k_L N2Th(t,x) N2(t,x) \quad \text{(B.75)}$$

(c) Stored State

$$\frac{\partial Last(t,x)}{\partial t} = D\nabla^2 Last(t,x) - k_d Last(t,x) + k_d ffBfrd(t,x) \quad \text{(B.76)}$$

6.2 | Rule 110 Partial Differential Equations

"I hope to say something about a 'continuous' rather than 'crystalline' model [of automata]. There, as far as I can now see, a system of nonlinear partial differential equations, essentially of the

diffusion type, will be used."

- John von Neumann (Oct. 28th, 1952) discussing unfinished Theory of Automata.

von Neumann papers, Library of Congress, Box 28 "Theory of Automata".

This section describes the set of coupled partial differential equations that govern our Rule 110 chemical automaton, derived from the reactions in Appendix A. These equations use standard mass-action equations and the diffusion equation. Figure 8(b) contains a plot of the solution to these equations. One equation is devoted to each of the 76 species in our network. Figure 9 shows a detail of the circuit from Figure 7 for a Rule 110 automaton, with each species labeled.

Unless otherwise specified, all species start with zero initial concentration. Absorbing boundary conditions apply to all species whose concentrations change over time. Our reaction rate constants and diffusion coefficients are selected to be realistically attainable values for DNA-based reaction-diffusion networks, on the same order of magnitude as experimentally derived data in the literature[37,14,32]. The Mathematica code we used to numerically solve these equations is available upon request.

Constants:

$$xMax = 64 \qquad k_p = 0.002\,\mu\text{Ms}^{-1} \qquad k_{Bd} = 0.00002\,\text{s}^{-1}$$
$$D = 0.00015\,\text{mm}^2\text{s}^{-1} \qquad k_d = 0.002\,\mu\text{s}^{-1} \qquad c_{recTh} = 0.5\,\mu\text{M}$$
$$k_T = 20\,\mu\text{M}^{-1}\text{s}^{-1} \qquad k_x = 0.002\,\mu\text{M}^{-1}\text{s}^{-1} \qquad clkPeriod = 2*24*3600\,\text{s}$$
$$k_L = 0.2\,\mu\text{M}^{-1}\text{s}^{-1} \qquad k_B = 0.0002\,\mu\text{M}^{-2}\text{s}^{-1} \qquad clkDuty = .5*3600\,\text{s}$$

1. External Signals:
   (a) Keys (in μM):

$$Key_A(t,x) = \begin{cases} 1/(1+Exp[-25*(Mod(x,16)-1)]) & : Mod(x,16) \le 2 \\ 1-1/(1+Exp[-25*(Mod(x,16)-3)]) & : otherwise. \end{cases} \quad \text{(B.1)}$$

$$Key_B(t,x) = \begin{cases} 1/(1+Exp[-25*(Mod(x,16)-5)]) & : Mod(x,16) \le 6 \\ 1-1/(1+Exp[-25*(Mod(x,16)-7)]) & : otherwise. \end{cases} \quad \text{(B.2)}$$

$$Key_C(t,x) = \begin{cases} 1/(1+Exp[-25(Mod(x,16)-9)]) & : mod(x,16) \le 10 \\ 1-1/(1+Exp[-25(Mod(x,16)-11)]) & : otherwise. \end{cases} \quad \text{(B.3)}$$

$$Key_D(t,x) = \begin{cases} 1/(1+Exp[-25(Mod(x,16)-13)]) & : mod(x,16) \le 14 \\ 1-1/(1+Exp[-25(Mod(x,16)-15)]) & : otherwise. \end{cases} \quad \text{(B.4)}$$

   (b) Clock:

$$clk(t,x) = \begin{cases} 1\,\mu\text{M} & : t < 4000 \\ 1\,\mu\text{M} & : Mod(t, clkPeriod) < clkDuty \\ 0 & : otherwise. \end{cases} \quad \text{(B.5)}$$

$$\frac{\partial clk_{On}(t,x)}{\partial t} = D\nabla^2 clk_{On}(t,x) - k_d * clk_{On}(t,x) + k_d * clk(t,x) - k_L * clk_{On}(t,x) * Sg_{nxOn}(t,x) \quad \text{(B.6)}$$

$$\frac{\partial clk_{Off}(t,x)}{\partial t} = D\nabla^2 clk_{Off}(t,x) - k_d * clk_{Off}(t,x) + k_d * clk(t,x) - k_L * clk_{Off}(t,x) * Sg_{OffNx}(t,x) \quad \text{(B.7)}$$

2. Communication Stage:
   (a) Broadcast Modules (in μM):

$$Src_A(t,x) = 1 \quad \text{(B.8)} \qquad Src_B(t,x) = 1 \quad \text{(B.9)} \qquad Src_C(t,x) = 1 \quad \text{(B.10)} \qquad Src_D(t,x) = 1 \quad \text{(B.11)}$$

   (b) Broadcast signals, note initial concentration of Sig_A triggers initial *on* cell:

$$\frac{\partial Sig_A(t,x)}{\partial t} = D\nabla^2 Sig_A(t,x) + k_B Src_A(t,x) Key_A(t,x) Last(t,x) - k_{Bd} Sig_A(t,x) \quad \text{(B.12)}$$

$$Sig_A(t=0,x) = \begin{cases} 2\,\mu\text{M} & : x_{max} - 8 < x < x_{max} - 0.1 \\ 0 & : otherwise. \end{cases}$$

$$\frac{\partial Sig_B(t,x)}{\partial t} = D\nabla^2 Sig_B(t,x) + k_B Src_B(t,x) Key_B(t,x) Last(t,x) - k_{Bd} Sig_B(t,x) \quad \text{(B.13)}$$

$$\frac{\partial Sig_C(t,x)}{\partial t} = D\nabla^2 Sig_C(t,x) + k_B Src_C(t,x) Key_C(t,x) Last(t,x) - k_{Bd} Sig_C(t,x) \quad \text{(B.14)}$$

$$\frac{\partial Sig_D(t,x)}{\partial t} = D\nabla^2 Sig_D(t,x) + k_B Src_D(t,x) Key_D(t,x) Last(t,x) - k_{Bd} Sig_D(t,x) \quad \text{(B.15)}$$

   (c) Receiving and processing left-hand neighbor signal:

$$\frac{\partial L_{raw}(t,x)}{\partial t} = D\nabla^2 L_{raw}(t,x) - 4k_d L_{raw}(t,x) - k_T L_{raw}(t,x) Th_l(t,x) + k_x Sig_D(t,x) Key_A(t,x) + k_x Sig_A(t,x) Key_B(t,x) + k_x Sig_B(t,x) Key_C(t,x) + k_x Sig_C(t,x) Key_D(t,x) \quad \text{(B.16)}$$

$$\frac{\partial Th_l(t,x)}{\partial t} = D\nabla^2 Th_l(t,x) + c_{recTh} k_p - k_d Th_l(t,x) - k_T L_{raw}(t,x) Th_l(t,x) \quad \text{(B.17)}$$

$$\frac{\partial Ampl(t,x)}{\partial t} = D\nabla^2 Ampl(t,x) + k_p - k_d Ampl(t,x) - k_L L_{raw}(t,x) Ampl(t,x) \quad \text{(B.18)}$$

$$\frac{\partial Lft(t,x)}{\partial t} = D\nabla^2 Lft(t,x) - k_d Lft(t,x) + k_L L_{raw}(t,x) Ampl(t,x) \quad \text{(B.19)}$$

   (d) Receiving and processing right-hand neighbor signal:

$$\frac{\partial R_{raw}(t,x)}{\partial t} = D\nabla^2 R_{raw}(t,x) - 4k_d R_{raw}(t,x) - k_T R_{raw}(t,x) Th_r(t,x) + k_x Sig_(t,x) Key_A(t,x) + k_x Sig_C(t,x) Key_B(t,x) + k_x Sig_D(t,x) Key_C(t,x) + k_x Sig_A(t,x) Key_D(t,x) \quad \text{(B.20)}$$

$$\frac{\partial Th_r(t,x)}{\partial t} = D\nabla^2 Th_r(t,x) + c_{recTh}k_p - k_d Th_r(t,x) - k_T R_{raw}(t,x) Th_r(t,x) \tag{B.21}$$

$$\frac{\partial Amp_{rt}(t,x)}{\partial t} = D\nabla^2 Amp_{rt}(t,x) + k_p - k_d Amp_{rt}(t,x) - k_L R_{raw}(t,x) Amp_{rt}(t,x) \tag{B.22}$$

$$\frac{\partial Rght(t,x)}{\partial t} = D\nabla^2 Rght(t,x) - k_d Rght(t,x) + k_L R_{raw}(t,x) Amp_{rt}(t,x) \tag{B.23}$$

3. Calculation Stage:

(a) Copy left, right and previous time step (pr) signals (multiple gates operate on each)

$$\frac{\partial Br_{pr}(t,x)}{\partial t} = D\nabla^2 Br_{pr}(t,x) - k_d Br_{pr}(t,x) + k_d Last(t,x) - k_T Br_{pr}(t,x) Off_{pr}(t,x) \tag{B.24}$$

$$\frac{\partial R_{br}(t,x)}{\partial t} = D\nabla^2 R_{br}(t,x) - k_d R_{br}(t,x) + k_d Rght(t,x) - k_L R_{br}(t,x) Sg_{br}(t,x) \tag{B.25}$$

$$\frac{\partial On_{pr}(t,x)}{\partial t} = D\nabla^2 On_{pr}(t,x) - k_d On_{pr}(t,x) + k_d Last(t,x) - k_L On_{pr}(t,x) Sg_{cr}(t,x) \tag{B.26}$$

$$\frac{\partial R_{cr}(t,x)}{\partial t} = D\nabla^2 R_{cr}(t,x) - k_d R_{cr}(t,x) + k_d Rght(t,x) - k_L R_{cr}(t,x) Sg_{nbOff}(t,x) \tag{B.27}$$

$$\frac{\partial L_{cr}(t,x)}{\partial t} = D\nabla^2 L_{cr}(t,x) - k_d L_{cr}(t,x) + k_d Lft(t,x) - k_L L_{cr}(t,x) Sg_{nbOff}(t,x) \tag{B.28}$$

$$\frac{\partial Off_{pr}(t,x)}{\partial t} = D\nabla^2 Off_{pr}(t,x) + k_p - k_d Off_{pr}(t,x) - {}_T Br_{pr}(t,x) Off_{pr}(t,x) - k_L Off_{pr}(t,x) Sg_{br}(t,x) \tag{B.29}$$

(b) Boolean logic for Br (birth) condition: *off* to *on* transition

$$\frac{\partial Sg_{br}(t,x)}{\partial t} = D\nabla^2 Sg_{br}(t,x) + 2k_p - k_d Sg_{br}(t,x) - k_L Off_{pr}(t,x) Sg_{br}(t,x) - k_L R_{br}(t,x) Sg_{br}(t,x) \tag{B.30}$$

$$\begin{aligned}\frac{\partial Sum_{br}(t,x)}{\partial t} &= D\nabla^2 Sum_{br}(t,x) - k_d Sum_{br}(t,x) + k_L Off_{pr}(t,x) Sg_{br}(t,x) \\ &\quad + k_L R_{br}(t,x) Sg_{br}(t,x) - k_T Sum_{br}(t,x) Th_{br}(t,x)\end{aligned} \tag{B.31}$$

$$\frac{\partial Th_{br}(t,x)}{\partial t} = D\nabla^2 Th_{br}(t,x) + 1.35k_p - k_d Th_{br}(t,x) - k_T Sum_{br}(t,x) Th_{br}(t,x) \tag{B.32}$$

$$\frac{\partial Amp_{br}(t,x)}{\partial t} = D\nabla^2 Amp_{br}(t,x) + k_p - k_d Amp_{br}(t,x) - k_L Sum_{br}(t,x) Amp_{br}(t,x) \tag{B.33}$$

$$\frac{\partial Br(t,x)}{\partial t} = D\nabla^2 Br(t,x) - k_d Br(t,x) + k_L Sum_{br}(t,x) Amp_{br}(t,x) - k_L Br(t,x) Sg_{nx}(t,x) \tag{B.34}$$

(c) Boolean logic for no death condition (*on* and at least one neighbor *off*): stay *on*

$$\begin{aligned}\frac{\partial Sg_{nbOff}(t,x)}{\partial t} &= D\nabla^2 Sg_{nbOff}(t,x) + 2k_p - k_d Sg_{nbOff}(t,x) - k_L R_{cr}(t,x) Sg_{nbOff}(t,x) \\ &\quad - k_L L_{cr}(t,x) Sg_{nbOff}(t,x)\end{aligned} \tag{B.35}$$

$$\begin{aligned}\frac{\partial Sum_{nbOff}(t,x)}{\partial t} &= D\nabla^2 Sum_{nbOff}(t,x) - k_d Sum_{nbOff}(t,x) + k_L R_{cr}(t,x) Sg_{nbOff}(t,x) \\ &\quad + k_L L_{cr}(t,x) Sg_{nbOff}(t,x) - k_T Sum_{nbOff}(t,x) Th_{nbOff}(t,x)\end{aligned} \tag{B.36}$$

$$\frac{\partial Th_{nbOff}(t,x)}{\partial t} = D\nabla^2 Th_{nbOff}(t,x) + 1.35k_p - k_d Th_{nbOff}(t,x) - k_T Sum_{nbOff}(t,x) Th_{nbOff}(t,x) \tag{B.37}$$

$$\frac{\partial Amp_{nbOff}(t,x)}{\partial t} = D\nabla^2 Amp_{nbOff}(t,x) + k_p - k_d Amp_{nbOff}(t,x) - k_L Th_{nbOff}(t,x) Amp_{nbOff}(t,x) \tag{B.38}$$

$$\begin{aligned}\frac{\partial nbrOff(t,x)}{\partial t} &= D\nabla^2 nbrOff(t,x) - k_d nbrOff(t,x) + k_L Th_{nbOff}(t,x) Amp_{nbOff}y(t,x) \\ &\quad - k_L nbrOff(t,x) Sg_{cr}(t,x)\end{aligned} \tag{B.39}$$

$$\frac{\partial Sg_{cr}(t,x)}{\partial t} = D\nabla^2 Sg_{cr}(t,x) + 2k_p - k_d Sg_{cr}(t,x) - k_L On_{pr}(t,x) Sg_{cr}(t,x) - k_L nbrOff(t,x) Sg_{cr}(t,x) \tag{B.40}$$

$$\begin{aligned}\frac{\partial Sum_{cr}(t,x)}{\partial t} &= D\nabla^2 Sum_{cr}(t,x) - k_d Sum_{cr}(t,x) + k_L On_{pr}(t,x) Sg_{cr}(t,x) \\ &\quad + k_L nbrOff(t,x) Sg_{cr}(t,x) - k_T Sum_{cr}(t,x) Th_{cr}(t,x)\end{aligned} \tag{B.41}$$

$$\frac{\partial Th_{cr}(t,x)}{\partial t} = D\nabla^2 Th_{cr}(t,x) + 1.35k_p - k_d Th_{cr}(t,x) - k_T Sum_{cr}(t,x) Th_{cr}(t,x) \tag{B.42}$$

$$\frac{\partial Amp_{cr}(t,x)}{\partial t} = D\nabla^2 Amp_{cr}(t,x) + k_p - k_d Amp_{cr}(t,x) - k_L Sum_{cr}(t,x) Amp_{cr}(t,x) \tag{B.43}$$

$$\frac{\partial Live(t,x)}{\partial t} = D\nabla^2 Live(t,x) - k_d Live(t,x) + k_L Sum_{cr}(t,x) Amp_{cr}(t,x) - k_L Live(t,x) Sg_{nx}(t,x) \tag{B.44}$$

(d) Boolean logic to determine if next state is *on*

$$\frac{\partial Sg_{nx}(t,x)}{\partial t} = D\nabla^2 Sg_{nx}(t,x) + 2k_p - k_d Sg_{nx}(t,x) - k_L Live(t,x) Sg_{nx}(t,x) - k_L Br(t,x) Sg_{nx}(t,x) \tag{B.45}$$

$$\begin{aligned}\frac{\partial Sum_{nx}(t,x)}{\partial t} &= D\nabla^2 Sum_{nx}(t,x) - k_d Sum_{nx}(t,x) + k_L Live(t,x) Sg_{nx}(t,x) + k_L Br(t,x) Sg_{nx}(t,x) \\ &\quad - k_T Sum_{nx}(t,x) Th_{nx}(t,x)\end{aligned} \tag{B.46}$$

$$\frac{\partial Th_{nx}(t,x)}{\partial t} = D\nabla^2 Th_{nx}(t,x) + 0.65k_p - k_d Th_{nx}(t,x) - k_T Sum_{nx}(t,x) Th_{nx}(t,x) \tag{B.47}$$

$$\frac{\partial Amp_{nx}(t,x)}{\partial t} = D\nabla^2 Amp_{nx}(t,x) + k_p - k_d Amp_{nx}(t,x) - k_L Sum_{nx}(t,x) Amp_{nx}(t,x) \tag{B.48}$$

$$\frac{\partial Nx(t,x)}{\partial t} = D\nabla^2 Nx(t,x) - k_d Nx(t,x) + k_L Sum_{nx}(t,x) Amp_{nx}(t,x) \tag{B.49}$$

$$\frac{\partial On_{nx}(t,x)}{\partial t} = D\nabla^2 On_{nx}(t,x) - k_d On_{nx}(t,x) + k_d Nx(t,x) - k_L On_{nx}(t,x) Sg_{nxOn}(t,x) \tag{B.50}$$

(e) Boolean logic to determine if next state is *off*

$$\frac{\partial Nt_{nx}(t,x)}{\partial t} = D\nabla^2 Nt_{nx}(t,x) - k_d Nt_{nx}(t,x) + k_d Nx(t,x) - k_T Nt_{nx}(t,x) Off_{nx}(t,x) \tag{B.51}$$

$$\frac{\partial Off_{nx}(t,x)}{\partial t} = D\nabla^2 Off_{nx}(t,x) + k_p - k_d Off_{nx}(t,x) - k_T Nt_{nx}(t,x) Off_{nx}(t,x) - k_L Off_{nx}(t,x) Sg_{OffNx}(t,x) \tag{B.52}$$

(f) Clocked synchronization gates

$$\frac{\partial Sg_{nxOn}(t,x)}{\partial t} = D\nabla^2 Sg_{nxOn}(t,x) + 2k_p - k_d Sg_{nxOn}(t,x) - k_L On_{nx}(t,x) Sg_{nxOn}(t,x) - k_L clk_{On}(t,x) Sg_{nxOn}(t,x) \quad \text{(B.53)}$$

$$\frac{\partial On_{nx}Sum(t,x)}{\partial t} = D\nabla^2 On_{nx}Sum(t,x) - k_d On_{nx}Sum(t,x) + k_L On_{nx}(t,x) Sg_{nxOn}(t,x) + k_L clk_{On}(t,x) Sg_{nxOn}(t,x) - k_T On_{nx}Sum(t,x) On_{nx}Th(t,x) \quad \text{(B.54)}$$

$$\frac{\partial On_{nx}Th(t,x)}{\partial t} = D\nabla^2 On_{nx}Th(t,x) + 1.35k_p - k_d On_{nx}Th(t,x) - k_T On_{nx}Sum(t,x) On_{nx}Th(t,x) \quad \text{(B.55)}$$

$$\frac{\partial On_{nx}Amp(t,x)}{\partial t} = D\nabla^2 On_{nx}Amp(t,x) + k_p - k_d On_{nx}Amp(t,x) - k_L On_{nx}Sum(t,x) On_{nx}Amp(t,x) \quad \text{(B.56)}$$

$$\frac{\partial Set_{Bfr}(t,x)}{\partial t} = D\nabla^2 Set_{Bfr}(t,x) - k_d Set_{Bfr}(t,x) + k_L On_{nx}Sum(t,x) On_{nx}Amp(t,x) \quad \text{(B.57)}$$

$$\frac{\partial Sg_{OffNx}(t,x)}{\partial t} = D\nabla^2 Sg_{OffNx}(t,x) + 2k_p - k_d Sg_{OffNx}(t,x) - k_L Off_{nx}(t,x) Sg_{OffNx}(t,x) - k_L clk_{Off}(t,x) Sg_{OffNx}(t,x) \quad \text{(B.58)}$$

$$\frac{\partial Sum_{OffNx}(t,x)}{\partial t} = D\nabla^2 Sum_{OffNx}(t,x) - k_d Sum_{OffNx}(t,x) + k_L Off_{nx}(t,x) Sg_{OffNx}(t,x) + k_L clk_{Off}(t,x) Sg_{OffNx}(t,x) - k_T Sum_{OffNx}(t,x) Th_{OffNx}(t,x) \quad \text{(B.59)}$$

$$\frac{\partial Th_{OffNx}(t,x)}{\partial t} = D\nabla^2 Th_{OffNx}(t,x) + 1.35k_p - k_d Th_{OffNx}(t,x) - k_T Sum_{OffNx}(t,x) Th_{OffNx}(t,x) \quad \text{(B.60)}$$

$$\frac{\partial Amp_{OffNx}(t,x)}{\partial t} = D\nabla^2 Amp_{OffNx}(t,x) + k_p - k_d Amp_{OffNx}(t,x) - k_L Sum_{OffNx}(t,x) Amp_{OffNx}(t,x) \quad \text{(B.61)}$$

$$\frac{\partial Res_{Bfr}(t,x)}{\partial t} = D\nabla^2 Res_{Bfr}(t,x) - k_d Res_{Bfr}(t,x) + k_L Sum_{OffNx}(t,x) Amp_{OffNx}(t,x) \quad \text{(B.62)}$$

4. Storage Stage

(a) Copies of Set/Res signals

$$\frac{\partial Set(t,x)}{\partial t} = D\nabla^2 Set(t,x) - k_d Set(t,x) + k_d Set_{Bfr}(t,x) - k_L Set(t,x) N1Sg(t,x) \quad \text{(B.63)}$$

$$\frac{\partial Res(t,x)}{\partial t} = D\nabla^2 Res(t,x) - k_d Res(t,x) + k_d Res_{Bfr}(t,x) - k_L Res(t,x) N2Sg(t,x) \quad \text{(B.64)}$$

(b) Flip- op module

$$\frac{\partial ffBfrd(t,x)}{\partial t} = D\nabla^2 ffBfrd(t,x) - k_d ffBfrd(t,x) + k_L N2Th(t,x) N2(t,x) \quad \text{(B.65)}$$

$$\frac{\partial ffFbackNot(t,x)}{\partial t} = D\nabla^2 ffFbackNot(t,x) - k_d ffFbackNot(t,x) + k_L N1Th(t,x) N1(t,x) - k_L ffFbackNot(t,x) N2Sg(t,x) \quad \text{(B.66)}$$

$$\frac{\partial N1Th(t,x)}{\partial t} = D\nabla^2 N1Th(t,x) + 0.65k_p - k_d N1Th(t,x) - k_T N1Sum(t,x) N1Th(t,x) \quad \text{(B.67)}$$

$$\frac{\partial N2Sum(t,x)}{\partial t} = D\nabla^2 N2Sum(t,x) - k_d N2Sum(t,x) + k_L Res(t,x) N2Sg(t,x) + k_L ffFbackNot(t,x) N2Sg(t,x) - k_T N2Sum(t,x) N2Th(t,x) \quad \text{(B.68)}$$

$$\frac{\partial ffFback(t,x)}{\partial t} = D\nabla^2 ffFback(t,x) - k_d ffFback(t,x) + k_d ffBfrd(t,x) - k_L ffFback(t,x) N1Sg(t,x) \quad \text{(B.69)}$$

$$\frac{\partial N1Sg(t,x)}{\partial t} = D\nabla^2 N1Sg(t,x) + 2k_p - k_d N1Sg(t,x) - k_L Set(t,x) N1Sg(t,x) - k_L ffFback(t,x) N1Sg(t,x) \quad \text{(B.70)}$$

$$\frac{\partial N1Sum(t,x)}{\partial t} = D\nabla^2 N1Sum(t,x) - k_d N1Sum(t,x) + k_L Set(t,x) N1Sg(t,x) + k_L ffFback(t,x) N1Sg(t,x) - k_T N1Sum(t,x) N1Th(t,x) \quad \text{(B.71)}$$

$$\frac{\partial N1(t,x)}{\partial t} = D\nabla^2 N1(t,x) + k_p - k_d N1(t,x) - k_L N1Th(t,x) N1(t,x) \quad \text{(B.72)}$$

$$\frac{\partial N2Sg(t,x)}{\partial t} = D\nabla^2 N2Sg(t,x) + 2k_p - k_d N2Sg(t,x) - k_L Res(t,x) N2Sg(t,x) - k_L ffFbackNot(t,x) N2Sg(t,x) \quad \text{(B.73)}$$

$$\frac{\partial N2Th(t,x)}{\partial t} = D\nabla^2 N2Th(t,x) + 0.65k_p - k_d N2Th(t,x) - k_T N2Sum(t,x) N2Th(t,x) \quad \text{(B.74)}$$

$$\frac{\partial N2(t,x)}{\partial t} = D\nabla^2 N2(t,x) + k_p - k_d N2(t,x) - k_L N2Th(t,x) N2(t,x) \quad \text{(B.75)}$$

(c) Stored State

$$\frac{\partial Last(t,x)}{\partial t} = D\nabla^2 Last(t,x) - k_d Last(t,x) + k_d ffBfrd(t,x) \quad \text{(B.76)}$$

6.3 | Asynchronous Cellular Automaton Circuit

One limitation to the design for the cellular automaton that we presented is that it requires a global "clock" signal that is continuously supplied such that it continually changes concentration in

an oscillatory pattern. It is likely that providing such a signal externally would be challenging, because existing biomolecular oscillators are generally noisy, and may not provide the required synchronization [15]. An alternative to the design we have presented would be to use an asynchronous cellular automaton, which does not require a clock signal to operate. Instead, for each update round, all cells in the network wait until they have received information indicating that their neighbors are ready to proceed to the next round, thus preventing any single cell from getting out of synchrony with its immediate neighbors.

A design for an asynchronous CA is outlined in Figure 10. The circuit elements (i.e. modules) are the same as those for the synchronous CA, but more elements are required and they are arranged differently. In this asynchronous CA, when both neighbors' broadcasts are received by a cell, the cell disables its own broadcast and proceeds to calculate its own next state. After calculating its own next state, a cell will wait until it is no longer receiving any broadcasts of its neighbor cells' states. When this state is reached, the cell will save its calculated state in memory and restart the broadcasts of its current state. Cells can be at most one step ahead of their neighbors before they are forced to stop and wait for their neighbors to catch up. This strategy guarantees that each cell will pass through the correct series of states, and that local groups can never be out of synchronization. However, because each cell is permitted to proceed up to one step in advance of its neighbors, any two cells that are N neighbors away from each other can also be up to N time steps out of synchronization. So while the overall process will compute the correct dynamics for each individual cell, the state of the system may not represent the particular global state of the ideal cellular automaton for any particular time step. This issue could make it difficult to use such a CA to process incoming spatial information within the environment, because the computation process in different regions of the CA will not see this environmental information at the same stage of computation. Furthermore, the overall circuit size of our asynchronous CA is much larger than the synchronous (clocked) design. However, we

believe that an asynchronous automaton could overcome many of the disadvantages of the synchronous automaton we presented, because the asynchronous version requires less global coordination.

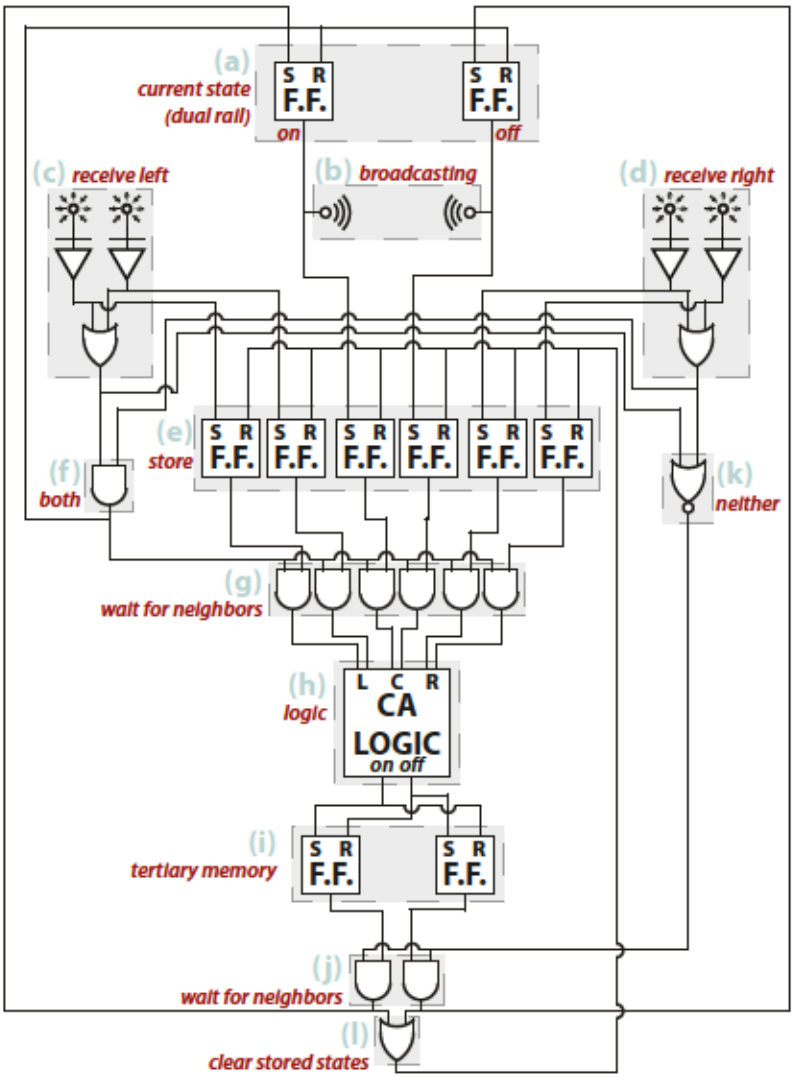

Figure 6.2. An asynchronous cellular automaton. (a)A cell's state is stored in memory. Dual-rail logic is used to differentiate between neighbors that are off and neighbors that have had insufficient time to broadcast their state. (b) Current states are broadcast. (c & d) Neighbor states are received. (e) Dual-rail states of cell and both neighbors are stored in a secondary memory bank. This enables cells to remember neighbors states even after neighbors stop broadcasting. (f) When both neighbor broadcasts are received, a cell resets its primary memory, turns off its broadcast signals, and (g) passes the stored information to (h) the cellular

automaton logic stage. If the time scale for chemical reactions t_R is much faster than the time scale for diffusive propagation between cells t_D, then the logic stage is guaranteed to have sufficient time to use the information stored in (e) to calculate the next state and store the results in (i) a tertiary memory bank before it stops receiving neighbor broadcasts. This calculated next state is blocked by (j) two AND gates until (k) all neighbor broadcasts have had time to turn off. Then, the secondary memory in (e) is reset by (l), the next state is stored in (a), and the cycle repeats.

7 | DNA Strand Buffers

Summary. A buffer reaction actively resists changes to the concentration of a chemical species. Typically, buffering reactions have only been able to regulate the concentration of hydronium (*i.e.* pH) and other ions. Here we develop a new class of buffers that regulate the concentrations of short sequences of DNA (*i.e.* oligonucleotides). A buffer's behavior is determined by its setpoint concentration, capacity to resist disturbances, and response time after a disturbance. We provide simple mathematical formulae for selecting rate constants to tune each of these properties, and show how to design DNA sequences and concentrations to implement the desired rate constants. We demonstrate several oligonucleotide buffers that maintain oligonucleotide setpoint concentrations between 10 and 80 nM in the presence of disturbances of 50 to 500 nM, with response times of less than 10 minutes to 1.5 hours. Multiple buffers can regulate different sequences of DNA in parallel without crosstalk. Oligonucleotide buffers could stabilize and restore reactant concentrations in DNA circuits, or in self-assembly processes, allowing such systems to operate reliably for extended durations. These buffers might also be coupled to other reactions to buffer molecules besides DNA. In general, oligonucleotide buffers can be viewed as a chemical "battery" that maintains the total chemical potential of a buffered species in a closed system.

7.1 | Introduction

Acid-base buffers are used ubiquitously in both nature[1,2] and synthetic biochemistry[3,4] to maintain a constant concentration of hydronium ions, *i.e.* pH, in solution (Fig. 1a). For example, the bicarbonate buffer system[2] neutralizes strong acids and bases in the human bloodstream, thereby maintaining a physiological pH of 7.4. When the bicarbonate buffer system fails, pH fluctuates freely, resulting in ailments collectively known as acidemia and alkalemia. Similarly, acid-base buffers

in polymerase chain reaction (PCR) hold pH constant to stabilize and optimize the activity of polymerase, without which the reaction could not occur. Chemistries exist to buffer some ions besides hydronium, such as metal ions[5], and for specific molecules such as naphthyridine[6,7].

The ability to buffer the concentrations of a larger library of molecules could improve the reliability and robustness of many chemical processes. Changes in reactant concentration can alter chemical reaction rates, which are generally proportional to the products of the concentrations of the reactants. The dependence of reaction rates on concentration can be even more sensitive in large reaction networks[8] or cooperative reactions[9]. During crystallization, for example, small fluctuations in monomer concentration can have dramatic effects on crystal nucleation rates[10,11]. Buffer systems that regulate monomer concentrations could suppress these fluctuations, increasing crystal purity and yield. Buffering reagents could also be used to replenish reagents depleted by downstream loads, for instance to deliver a constant drug dosage[12] over time as a drug is consumed.

In this study we present a class of buffers that regulate the concentrations of oligonucleotides (Fig. 1b), short synthetic sequences of DNA, in an analogous manner to how acid-base buffers regulate pH. Each buffer regulates the concentration of a specific DNA sequence, and multiple different buffers can operate in the same solution, independently controlling the concentrations of their different target sequences. We create our buffers using DNA strand- displacement (DSD) reactions, sequence-specific DNA hybridization processes with tunable kinetics[13-16], which have previously been used to implement information processing reactions including amplifiers[17,18], neural networks[19,20], and Boolean logic circuits[21-25]. While DSD reactions are relatively well understood, here we demonstrate how they can operate in regimes containing high reactant concentrations and low reaction rate constants. Buffering the concentrations of oligonucleotides could allow for the self-assembly of larger DNA structures or DNA-templated structures with fewer defects[26,27] by providing a constant supply of fresh monomers, stabilizing the nucleation of DNA crystal structures[11], and

could enable DNA circuits[24,25] and sensors[28] to operate for extended durations by restoring depleted reactants.

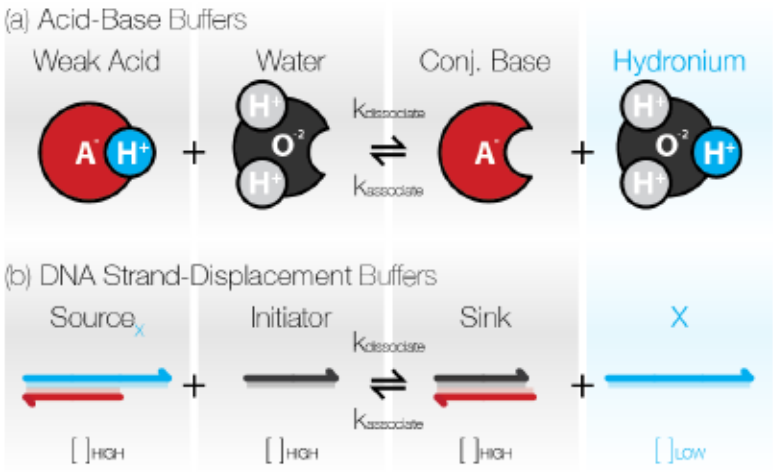

Figure 7.1. Buffer analogy. (a) Acid-base buffers regulate the concentration of hydronium ions. They consist of high concentrations of weak acid that partially dissociates in water to produce hydronium, and a conjugate base that reabsorbs hydronium. (b) DNA oligonucleotide buffers regulate the concentration of a target sequence X of DNA (cyan) that is initially sequestered within a source complex. Source dissociates in the presence of an initiator strand to release X and a conjugate sink complex.

7.2 | Background: acid-base pH buffers

According to the Arrhenius definition, an acid is a chemical that dissociates in water to increase the concentration of hydronium ions (H_3O^+):

$$HA + H_2O \underset{k_2}{\overset{k_1}{\rightleftharpoons}} A^- + H_3O^+ \qquad [1]$$

where HA is the acid and A^- is its conjugate base. In the simple case where the activities of all species are 1, the acid dissociation constant K_a, which determines the ratio of products to reactants at equilibrium, is:

$$K_a \equiv \frac{k_1[H_2O]}{k_2} = \frac{[H_3O^+][A^-]}{[HA]} \qquad [2]$$

where $[H_2O]$ is generally approximated as constant.

Weak acids have low dissociation constants and thus only dissociate partially into products. Acid-base pH buffers take advantage of this partial dissociation to create resistance to pH changes. They consist of a weak acid HA and its conjugate base A^-, each at high concentrations. When mixed together in water, HA and A^- continuously release and recapture a small amount of H_3O^+, forming a dynamic equilibrium. When additional H_3O^+ ions are added to the buffer, the solution changes pH, becoming more acidic. However, according to Le Chatelier's Principle, some of the H_3O^+ ions added to the buffer solution are absorbed into the weak acid state HA to restore equilibrium. In this manner, $[H_3O^+]$ increases by less than it would in the absence of buffer. Similarly, when H_3O^+ is removed from a buffer solution the equilibrium shifts in the opposite direction to replenish $[H_3O^+]$, resisting the change in concentration. The high concentrations of HA and A^- ensure that the concentrations of these species do not change significantly in response to relatively small disturbances.

7.3 | Synthetic buffers for other molecules

The general mechanism underlying pH buffers is not specific to acids and bases. A consequence of Le Chatelier's Principle is that any reversible reaction can serve as a buffer, given appropriate reaction rate constants and provided that all reactants are at a high concentration relative to the regulated species (see SI1). We use this simple idea to regulate the concentration of an arbitrary oligonucleotide in its single-stranded form.

To design a bimolecular buffer that regulates a target species X, we begin with a source complex (S) that is a precursor of X. This source reacts reversibly with an initiator molecule (I), releasing X in an active state, along with a conjugate sink molecule (N) that can recapture X (Eqn. 3). The species

S, I, N & X act as analogs to HA, H_2O, A^- & H_3O^+, respectively, in an acid-base buffer (Fig. 1, Eqn. 1).

$$S + I \underset{k_N}{\overset{k_S}{\rightleftharpoons}} N + X \qquad [3]$$

At high reactant concentrations, this reaction creates a stable equilibrium that resists disturbances to X (Fig. 2a,b). In this generalized form, the initiator is not necessarily H_2O, therefore it may not be possible to provide initiator at so overwhelming a concentration that its depletion can be ignored, as in Eqn. 2. Thus, to describe the operation of this generalized molecular buffer, we revert from the dissociation constant K_a to a more general chemical equilibrium constant:

$$K_{eq} \equiv \frac{k_S}{k_N} = \frac{[X]_{eq}[N]_{eq}}{[S]_{eq}[I]_{eq}}. \qquad [4]$$

A buffer of the form in Eqn. 3 has three important metrics that describe its performance: the setpoint concentration, the relaxation time constant, and the buffering capacity. Here we present equations that give order-of-magnitude estimates for how these values scale with the reactant concentrations and rate constants (see derivations in SI2).

(i) The setpoint $[X]_{set}$ is the equilibrium concentration of X generated by the buffer when $[X]_0 = 0$. For high concentrations of $[S]_0$, $[I]_0$, and $[N]_0$, and low equilibrium constants K_{eq}, the setpoint can be approximated as:

$$[X]_{set} \approx K_{eq}\frac{[S]_0[I]_0}{[N]_0}. \qquad [5]$$

Eqn. 5 indicates two separate types of parameters for specifying $[X]_{set}$: the equilibrium constant K_{eq} and the ratio of the reactant concentrations. Generally, changing K_{eq} requires redesigning the reactant molecules, rather than adjusting their initial concentrations. For simplicity in this paper, we will always keep $[S]_0 = [I]_0$.

(ii) The relaxation time constant τ determines how quickly X relaxes back towards its setpoint after it is disturbed. The relaxation time constant can be approximated by Eqn. 6, where the time it takes for a disturbance to relax to a defined percentage $0<\alpha<1$ of its initial amplitude is given by Eqn. 7.

$$\tau = \frac{1}{k_N [N]_o} \quad [6]$$

$$t_{relax,\alpha} = -\tau \cdot \ln(\alpha) \quad [7]$$

(iii) The buffering capacity β determines how much of an external perturbation the buffer can accommodate while maintaining a final equilibrium concentration $[X]_{eq}$ close to the initial setpoint concentration $[X]_{set}$. Specifically, β^+ is the maximum concentration of X that can be added to a buffer while keeping $[X]_{eq}$ below a specified factor of $[X]_{set}$ (here we use a factor of 1.1):

$$[X]_{eq} \leq 1.1 \cdot [X]_{set}, \quad [8]$$

and β^- is the maximum concentration of X that can be removed from the buffer while keeping $[X]_{eq}$ above an arbitrary factor of $[X]_{set}$ (here we use a factor of 0.9):

$$[X]_{eq} \geq 0.9 \cdot [X]_{set}. \quad [9]$$

The resulting capacities are:

$$\beta^+ = c^+ \cdot ([S]_0 + [I]_0 + [N]_0) \quad [10]$$

$$\beta^- = c^- \cdot ([S]_0 + [I]_0 + [N]_0) \quad [11]$$

where c^+ and c^- are coefficients determined by the relative ratios of S, I, and N (Fig. 2c). From Eqn.'s 10-11 we find that providing our reactants at high concentrations maximizes the capacity to recover from disturbances.

7.4 | Designing buffers using DNA strand-displacement

Next we use DNA strand-displacement (DSD) reactions to implement a buffer for DNA

oligonucleotides. In DSD reactions, an input strand of DNA binds to a multi-stranded DNA complex, and in the process displaces one or more output strands from the complex[29]. Short single-stranded domains, called toeholds, initiate these reactions and determine the forward and reverse rate constants[13,14].

In our DSD implementation (Fig. 3a) of the buffer reaction described in Eqn. 3, we start by designating a target DNA strand as X. Initially, X is bound within a source complex S, such that its toeholds are covered in an inert double-stranded state, preventing downstream reactions. An initiator strand I reversibly displaces X from S, and exposes the toeholds on X. In the process of displacing X, a new sink complex is created that consists of the initiator strand bound to the bottom strand of the source.

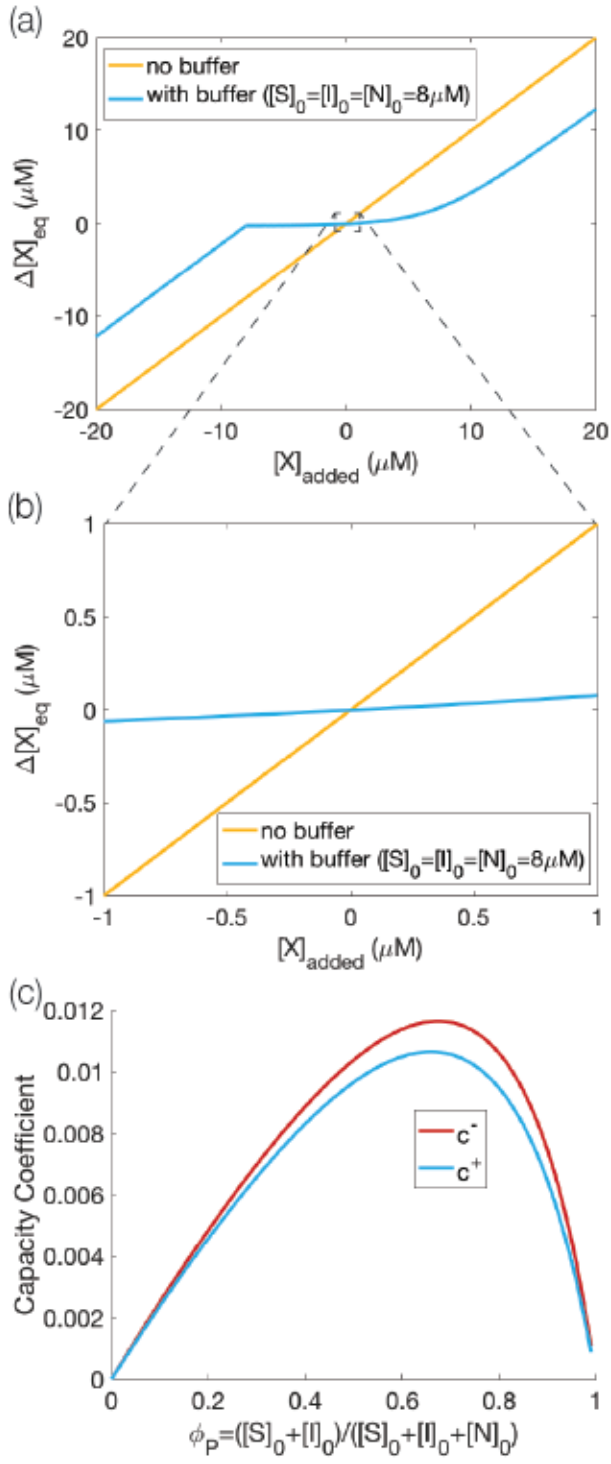

Figure 7.2. Resisting disturbances. (a) When X is added to or removed from an unbuffered solution, its change in concentration is exactly equal to the amount of X added or removed (yellow, slope=1). In contrast, a buffer solution with the form in Eqn. 3 can absorb some of the disturbance to prevent [X] from changing as steeply (cyan, with $K_{eq} = 0.026$). For large enough disturbances the buffer is overwhelmed, at which

point the slope again approaches 1. (b) For small disturbances, the slope is approximately linear and is much less than 1. (c) The buffer capacity is the amount of disturbance that can be absorbed while maintaining [X] within a target range (we use a range of $\pm 10\%$ of $[X]_{eq}$). The capacity is proportional to the total concentration of reactants, and its proportionality coefficients c^+ and c^- are functions of the relative reactant concentrations (from Eqn.'s 10-11).

7.4.1 | Selecting toehold lengths. The forward and reverse reaction rate constants for the DSD buffering reactions are determined by the toehold binding energy for the source and sink complexes, which is loosely correlated with their toehold lengths. We aimed to select toehold lengths that would create a buffer that holds X at setpoint concentrations $[X]_{set}$ on the order of 10-100 nM, with relaxation times on the order of 0.5 hours to relax to 10% of any disturbance (*i.e.* $t_{relax,\alpha} \approx 0.5\ hr$ with α=0.1). These target concentrations and response times are the same order as many existing DSD reactions in the literature[24,25], which facilitates the coupling of our oligonucleotide buffers to existing DSD circuits. To maximize the capacity of the buffer (Eqn.'s 10-11), we selected large concentrations of S, I and N, relative to $[X]_{set}$, up to 8 μM.

First we used Eqn.'s 6-7 to find a target reverse rate constant k_N. We plugged our desired relaxation time into Eqn. 7 to find a time constant of about τ=0.2 hr. Eqn. 6 indicates that to achieve this time constant with $[N]_0 = 8\ \mu M$, we should select a sink rate constant on order $k_N \approx 2 \cdot 10^{-4} \mu M^{-1} s^{-1}$. Similarly, to find a forward rate constant that would result in a setpoint concentration of the desired order with $[S]_0 = [I]_0 = [N]_0 = 8\ \mu M$, we used Eqn. 5 to find that we should aim for an equilibrium constant of roughly $K_{eq} \approx 0.01$. Plugging these target values for K_{eq} and k_N into Eqn. 4, we find that we want a source rate constant on order $k_S \approx 2 \cdot 10^{-6} \mu M^{-1} s^{-1}$.

We next chose toehold lengths whose average rate constants[13] were closest to our desired values. These lengths turned out to be a zero nucleotide (nt) toehold to drive the forward reaction, which

can initiate through fraying at the end of a double stranded complex, and a 2nt toehold to drive the reverse reaction. (Fig. 3, Supp. Section SI3).

7.4.2 | Measuring signal concentrations with a reversible reporting reaction. To monitor the free concentration of X during buffering, we used a DNA strand displacement reporter with quencher and fluorophore labels (Fig. 3b). X reacts reversibly with the reporter, so that when the reaction between X and the reporter is at equilibrium, the magnitude of fluorescence intensity can be used to determine [X] (see SI4). To accurately measure the kinetics of the buffering process, a reporter needs to equilibrate significantly faster than the buffer, such that the reporter remains at pseudo-steady-state as [X] changes. To ensure this condition is met, we selected a reporter toehold length of 5nt to characterize the buffer in Fig. 3a (see SI5).

7.5 | Control over setpoint concentration and relaxation time

To test our prediction (Eqn. 6) that the equilibration concentration of X is controlled by the initial concentrations of source (S), initiator (I), and sink (N), we combined these reactants at different initial concentrations, keeping $[S]_0 = [I]_0$ (see complete methods in SI4). We measured how [X] converged to a stable final value, $[X]_{set}$, for each set of initial reactant concentrations (Fig. 4a). For the values of $[S]_0$, $[I]_0$, and $[N]_0$ tested, $[X]_{set}$ ranged from 7 nM to 83 nM, on the order of the 10-100 nM range of setpoint concentrations we aimed to achieve. From Fig. 4b, the average equilibrium constant from all curves came to $K_{eq} = 0.026 \pm 0.016$, on the same order as our designed value (see section 3.1). Following our predictions, we observed that buffers with higher values of $[S]_0 = [I]_0$ produced higher $[X]_{set}$ values, and higher values of $[N]_0$ produced lower $[X]_{set}$ values (Fig. 4b), also consistent with the design.

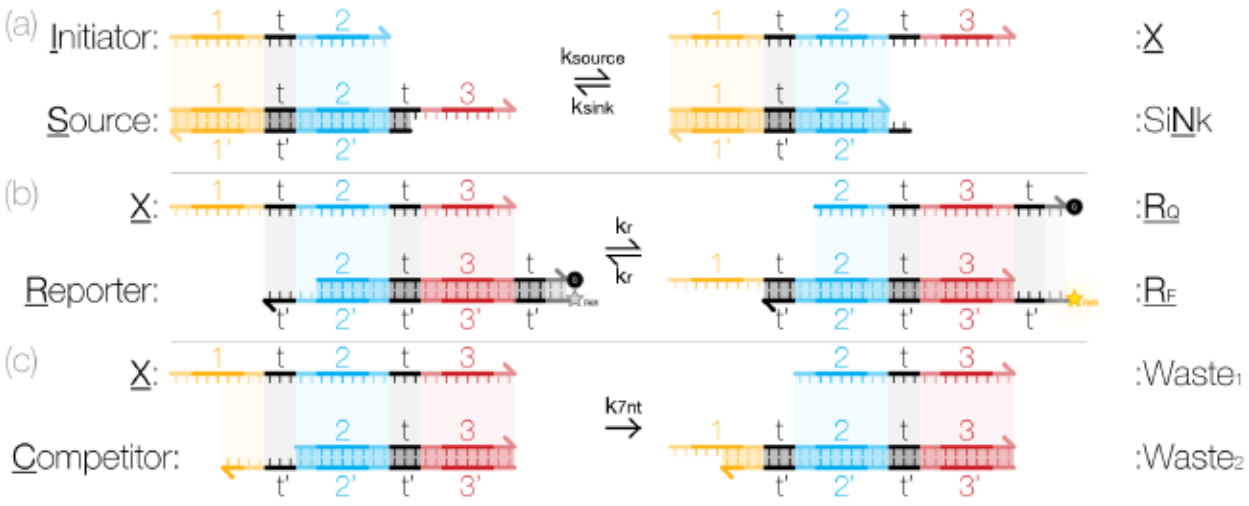

Figure 7.3. A DNA strand-displacement buffer circuit that regulates the concentration of a target DNA strand X. (a) X is initially bound within a source complex. Source reacts reversibly with initiator to release X, also creating a sink molecule, which drives the reverse reaction. (b) The concentration of X is monitored by a reporter complex. X reacts reversibly with reporter to separate a quencher-fluorophore pair, increasing the intensity of fluorescence. When sequestered in the source complex, the first toehold domain (black) on X is not available to initiate reactions with the reporter. (c) A competitor complex (commonly used as a "threshold" in other strand-displacement literature[25]) can irreversibly bind and sequester X via a fast 7nt toehold, reducing its free concentration in solution. We use the "leakless" architecture[30] to suppress reactions between species not designed to react.

We also characterized the relaxation of the buffer by fitting the data (Fig. 4c) to

$$[\mathrm{X}](t) = [\mathrm{X}]_{set}\left(1 + \varepsilon e^{-\frac{t}{\tau}}\right) \qquad [12]$$

$$\varepsilon \equiv \frac{[\mathrm{X}]_0 - [\mathrm{X}]_{set}}{[\mathrm{X}]_{set}}, \qquad [13]$$

which is the curve predicted by a simple bimolecular mass action model of Eqn. 3. For the range of initial concentrations tested, τ ranged from about 0.2 to 0.7 hours. Both the simple bimolecular mass action model of our buffer and a more detailed model[13] (SI6.1) of the buffering and reporting strand-displacement reactions predict that the rise time should be faster as $[\mathrm{N}]_0$ increases, because

$[N]_0$ controls the recapture rate. Consistent with these predictions, we observed that rise times were faster for higher values of $[N]_0$. The simple bimolecular model predicts that rise time should be independent of $[S]_0 = [I]_0$, however, the detailed model suggests a slight increase in rise time with decreasing $[S]_0 = [I]_0$ when the reporting reaction is included (SI6.2). This would suggest that the reporter imposes a small load on the system, causing a delay before equilibrium is reached. In our experiments we observed small increases in rise time for decreasing $[S]_0 = [I]_0$.

From Fig. 4c, the average sink reaction rate constant was $k_N \approx 1.7 \cdot 10^{-4} \pm 4.2 \cdot 10^{-5}\ \mu M^{-1}s^{-1}$. This constant and the measured $K_{eq} = 0.026 \pm 0.016$ give an average source reaction rate constant of $k_N \approx 4.0 \cdot 10^{-6} \pm 7.5 \cdot 10^{-7}\ \mu M^{-1}s^{-1}$. These values are within an order of magnitude of their designed values (Section 3.1).

7.6 | Response to disturbances

We next sought to characterize the relaxation of oligonucleotide buffers as they are perturbed from equilibrium. We used $[S]_0 = [I]_0 = [N]_0 = 8\ \mu M$, which we call the uniform 8 μM buffer.

7.6.1 | Positive disturbances. First we characterized how oligonucleotide buffers resist positive perturbations, *i.e.* sudden increases in [X]. We let the uniform 8 μM buffer equilibrate, and then added a pulse disturbance of 50 nM of X every three hours, for a total of ten pulses (Fig. 5a). After each pulse was added, the concentration of X increased quickly, reflecting the X that was added, and then relaxed back to a new equilibrium state close to $[X]_{set}$. We fit Eqn. 8 to each relaxation and found no significant variation in the time constant τ compared to the initial relaxation from $[X]_0 = 0$ (SI7). These experiments show that over at least ten perturbations of this size, the dynamics of relaxation are largely independent of the history of past perturbations. We would, however, expect that $[X]_{eq}$ should shift slightly over multiple perturbations of the same direction as

the buffer capacity is depleted. $[X]_{eq}$ did increase roughly linearly with the total amount of X added. After 500 nM of X was added, $[X]_{eq}$ increased by 8 nM, which gives a slope of approximately 0.016 (Fig. 5b, compare with Fig. 2b). This corresponds well with Eqn. 10, which predicts that approximately 250 nM of X must be added to increase the equilibrium concentration by 10%.

7.6.2 | Negative disturbances. We then characterized how the buffer responded to decreases in [X] caused by a coupled reaction in which X was consumed. We used a simple, irreversible reaction in which X binds to a competitor complex *via* a 7nt toehold initiated displacement reaction, producing inert waste products (Fig. 3c). We allowed the 8 μM uniform buffer to reach steady state for 3 hours, then added 100 nM of competitor (Fig. 5c). As anticipated, the concentration of X initially dropped, and then recovered to approximately 3 nM less than the initial equilibrium concentration.

The observed magnitude of this negative disturbance was significantly smaller than the concentration of competitor that was added. Our models (SI 5.3) indicate that this effect is due to a partial reaction known as toehold occlusion in which the toehold on the competitor is transiently occupied by the initiator. This interaction reduces the fraction of competitor that is free to react with X, in turn reducing their effective reaction rate. Toehold occlusion is especially significant here due to the long 7nt toehold and because $[I] \gg [X]_{set}$, both of which increase the residency time of toehold binding.

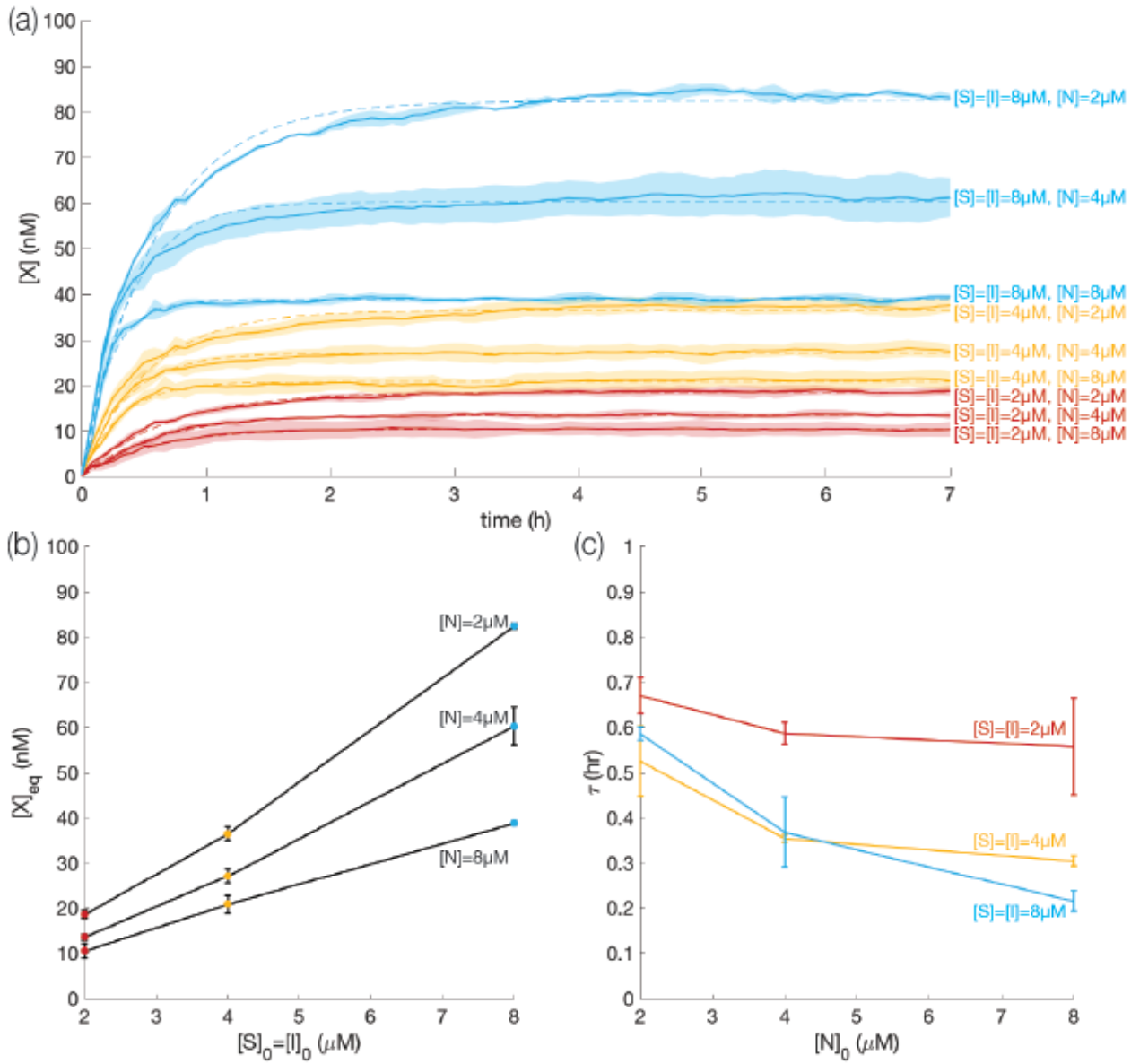

Figure 7.4. Concentration parameter space, showing equilibration of [X] with varied concentrations of $[S]_0$, $[I]_0$, and $[N]_0$. (a) Experimental data (solid lines) showing approach to equilibrium. Exponential fits shown as dashed lines. (b) Equilibrium concentration vs. $[S]_0 = [I]_0$. Cyan points correspond to the $[S]_0 = [I]_0 = 8\,\mu M$ trajectories from panel (a), yellow points to yellow trajectories, and red to red. (c) Relaxation time constants vs. $[N]_0$. Error bars here and elsewhere depict 95% confidence intervals ($1.96\sigma/\sqrt{n}$).

7.6.3 | Large mixed disturbances. Finally, we tested how oligonucleotide buffers respond to very large perturbations. We let a uniform 8 μM buffer approach its steady state for 3 hours and then perturbed it first by adding 250 nM of X, then adding 250 nM of the competitor (Fig. 5d, methods). We then added two more 250 nM pulses of X, followed by two more 250 nM pulses of competitor (Fig 5d). As expected, the changes in $[X]_{eq}$ in the buffered solution were far less in each case than the amount of X that was added or consumed. After the first 250 nM perturbation, $[X]_{eq}$ increased from about 47 to about 49 nM, consistent with the change in $[X]_{eq}$ that occurred after adding five 50 nM pulses of X (see Fig. 5b).

7.7 | Buffering multiple species

In principle, multiple different oligonucleotide sequences can be buffered in the same reaction. We designed a buffering system for a second target sequence, X_2 (see sequences in SI4) and compared how the X and X_2 buffers acted separately vs. together in the same reaction.

The 8 μM uniform buffer for X_2 reached a stable setpoint of around 100nM, and after a 50 nM perturbation $[X_2]$ returned to roughly the original setpoint concentration (Fig. 6a). The corresponding 8 μM uniform buffer for X also reached a setpoint, and responded to a 50 nM disturbance with a similar response time (Fig 6b). When the buffers were combined in the same solution, the setpoints and responses to disturbance for both X and X_2 were similar to their setpoints and responses in isolation (Fig 6c). [X] did not change when X_2 was added and *vice versa*.

7.8 | Faster buffering

The buffers in Figs. 3-5 have relatively slow relaxation times. A slow buffer can allow a system to maintain a memory of recent perturbations that is gradually erased as the concentration of the buffered species returns to equilibrium. This ability to transiently store information about

perturbations and then erase it to receive new perturbations could be used to process streams of chemical inputs[31,32]. However, in many other cases, it is important to maintain a constant concentration in the face of heavy loads. Faster time constants are desirable in such cases.

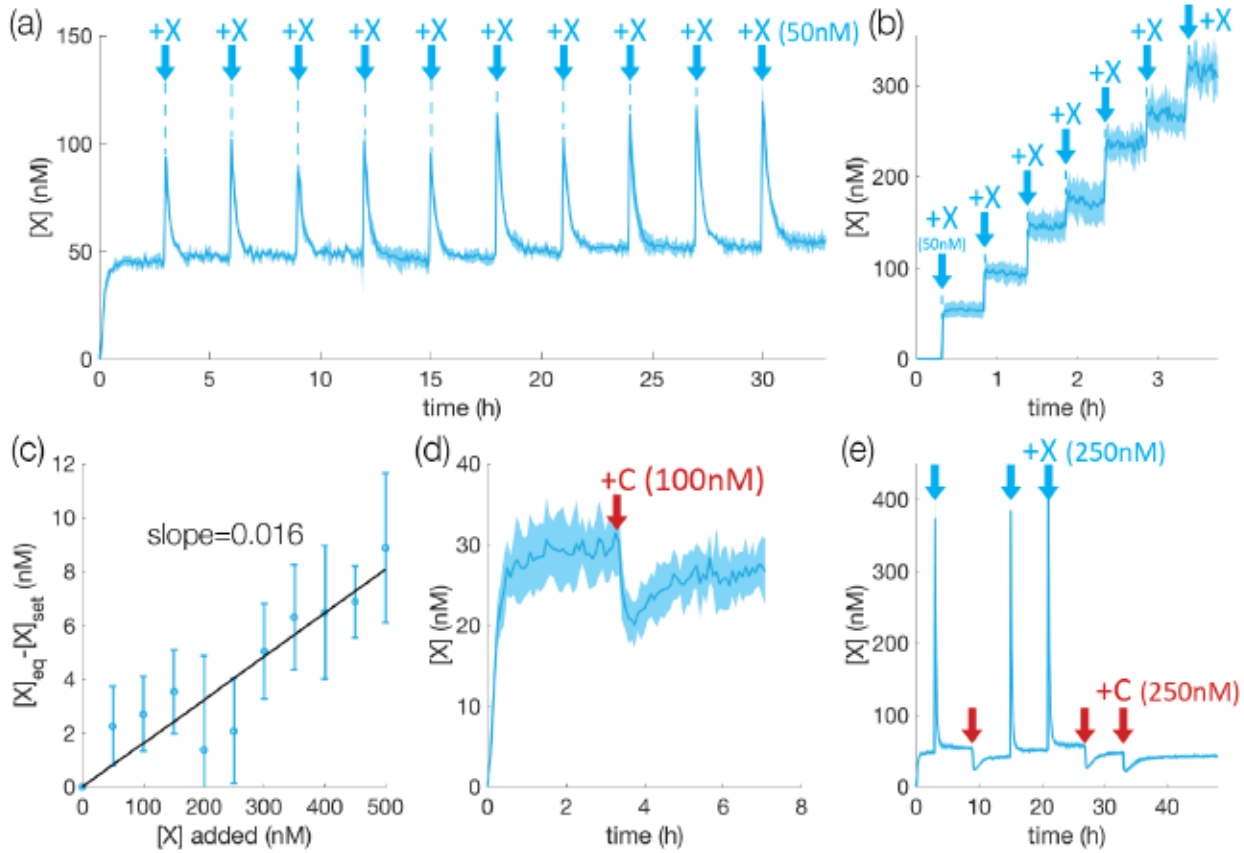

Figure 7.5. Response of the oligonucleotide buffer to disturbances, (a) An 8 μM uniform buffer for target species X disturbed with additions of 50 nM excess X every three hours. Differences in the amount of time required to add and mix the disturbances into the wells (during which no measurements were made) of a 96-well plate cause differences between the peak amplitudes of the disturbances. Since the fastest changes in concentration occur immediately after the disturbance is added, the first measured value of [X] is highly dependent on this delay time. (b) Addition of 50nM X disturbances to a solution containing no buffering reaction, showing cumulative increase in concentration. (c) The change in equilibrium concentration of X vs. the total concentration of X added as disturbance for the buffered data in panel (a). (d) 8 μM buffer disturbed with an addition of 100 nM competitor, C, which consumes X. (e) 8 μM buffer disturbed with large pulses of

250 nM X, followed by 250 nM competitor.

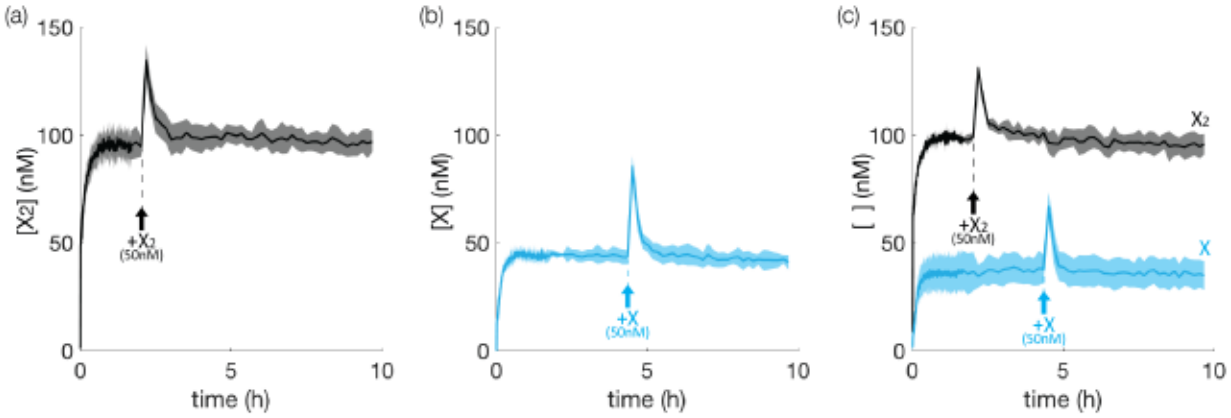

Figure 7.6. Buffers for different oligonucleotides operate in tandem without crosstalk. (a) Uniform 8 μM buffer for a second sequence X_2 before and after a 50 nM addition of X_2. The reporter for X_2 uses a HEX fluorophore. (b) Uniform 8 μM buffer for X before and after a 50 nM addition of X. (Using the reporter in Fig. 2c with FAM fluorophore). (c) The concentrations of X and X_2 in a solution containing 8 μM uniform buffers (and associated reporters) for both sequences, with 50 nM disturbances added at the same times as (a) and (b).

We therefore next investigated whether we could build buffers that responded more quickly to perturbations. To do so we created a buffer for X with longer toeholds on the source and sink complexes (+1 and +2 nucleotides respectively) that were designed to increase k_S and k_N (see SI4).

We characterized [X] over time using 8 μM of each of the new faster source, initiator and sink species, *i.e.* the fast 8 μM uniform buffer (Fig. 7), which produced a setpoint concentration of [X] of about 13 nM. When the system was perturbed by adding X, the concentration of X had returned close to its setpoint by the time we had returned the sample to the fluorescence reader after adding the disturbances. This delay in measurement was no more than 10 minutes, and generally less, suggesting that the fast buffer had a response time of at most 10 minutes, significantly faster than

the original buffer. As with the slower buffer, we observed a slower response time of the fast buffer to the addition of competitor than to the addition of X, consistent with the effects of toehold occlusion (SI8).

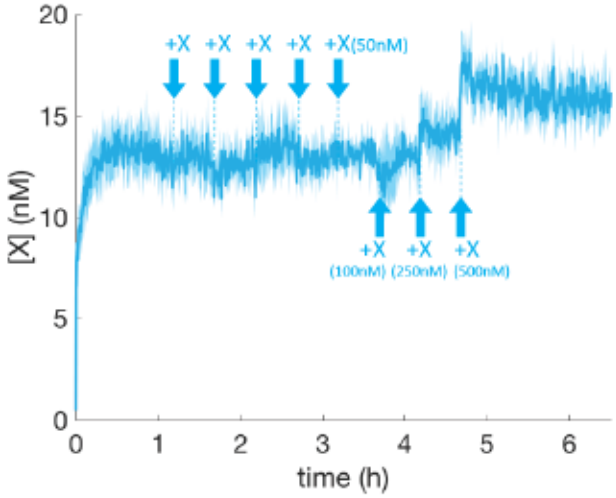

Figure 7.7. Faster responses to disturbances by a buffer with longer toeholds. We used a 1nt source toehold and a 4nt sink toehold to increase the rate of response of a uniform 8 μM fast buffer. X was added to disturb the system at times noted.

7.9 | Discussion

In this paper we demonstrate a DNA strand-displacement reaction for buffering the concentration of oligonucleotides. High concentrations of reactants continuously release and recapture a target strand, forming an equilibrium that resists perturbations to the concentration of the target. Using this architecture, buffers could be designed for arbitrary sequences with a wide range of setpoints, response times and capacities. Several buffers can operate in parallel within the same solution, to independently regulate the concentrations of multiple target strands.

Oligonucleotide buffers could be incorporated into a wide variety of existing reactions in which oligonucleotides play a key role, including self-assembly[11,26,27,33-36], sensing[28], photochemistry[37,38], and molecular release[39]. Oligonucleotide buffers could also regulate molecules besides DNA if coupled

to reactions that interface with other species, such as enzymes[40-42] and small molecules[43].

The ability to maintain the concentrations of molecular species is essential for building large reaction networks that operate reliably for extended times. Biological reaction networks often resist changes in concentration using mechanisms that are mathematically quite similar to buffers. For example, competing processes of synthesis and degradation in gene networks regulate the concentrations of most RNA and proteins[8]. Similar competing reactions on faster time scales also regulate the concentrations of actin[44] and active membrane receptors[45]. The ability to incorporate molecular buffering for a variety of species into synthetic chemical systems could therefore facilitate the design of robust and scalable synthetic chemical reaction networks.

Acknowledgments

The authors thank Chris Thachuk, Oleg Gang, Karen Fleming, Joshua Fern, Samuel Schaffter and Shivang Sharma for insightful conversations. This work was supported by NSF-SHF-1527377, DE-SC0010595 for some equipment and reagents, NSF-CCF-1161941 and a grant to the Turing Centenary Project by the John Templeton Foundation to R.S.

References

[1] Harris, D. C. (2010). Quantitative chemical analysis. Macmillan.
[2] Meldrum, N. U., & Roughton, F. J. W. (1933). Carbonic anhydrase. Its preparation and properties. The Journal of physiology, 80(2), 113-142.
[3] Saiki, R. K. (1989). The design and optimization of the PCR. PCR technology (pp. 7-16). Palgrave Macmillan, London.
[4] O'Farrell, P. H. (1975). High resolution two-dimensional electrophoresis of proteins. Journal of biological chemistry, 250(10), 4007-4021.
[5] Perrin, D. D. (2012). Buffers for pH and metal ion control. Springer Science & Business Media.
[6] Paffen, T. F., Ercolani, G., de Greef, T. F., & Meijer, E. W. (2015). Supramolecular Buffering by Ring–Chain Competition. Journal of the American Chemical Society, 137(4), 1501-1509.
[7] Paffen, T. F., Teunissen, A. J., de Greef, T. F., & Meijer, E. W. (2017). Model-driven engineering of supramolecular buffering by multivalency. Proceedings of the National Academy of Sciences, 114(49), 12882-12887.
[8] Alon, U. (2006). An introduction to systems biology: design principles of biological circuits. CRC press.
[9] Ferrell, J. E., & Machleder, E. M. (1998). The biochemical basis of an all-or-none cell fate switch in Xenopus oocytes. Science, 280(5365), 895-898.
[10] Coquerel, G. (2014). Crystallization of molecular systems from solution: phase diagrams, supersaturation and other basic concepts. Chemical Society Reviews, 43(7), 2286-2300.

[11] Schulman, R., & Winfree, E. (2007). Synthesis of crystals with a programmable kinetic barrier to nucleation. Proceedings of the National Academy of Sciences, 104(39), 15236-15241.
[12] Blume, G., & Cevc, G. (1990). Liposomes for the sustained drug release in vivo. Biochimica et Biophysica Acta (BBA)-Biomembranes, 1029(1), 91-97.
[13] Zhang, D. Y., & Winfree, E. (2009). Control of DNA strand displacement kinetics using toehold exchange. Journal of the American Chemical Society, 131(47), 17303-17314.
[14] Srinivas, N., Ouldridge, T.E., Šulc, P., Schaeffer, J.M., Yurke, B., Louis, A.A., Doye, J.P. and Winfree, E. (2013). On the biophysics and kinetics of toehold-mediated DNA strand displacement. Nucleic acids research, 41(22), 10641-10658.
[15] Soloveichik, D., Seelig, G., & Winfree, E. (2010). DNA as a universal substrate for chemical kinetics. Proceedings of the National Academy of Sciences, 107(12), 5393-5398.
[16] Chen, Y. J., Dalchau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D. and Seelig, G. (2013). Programmable chemical controllers made from DNA. Nature nanotechnology, 8(10), 755.
[17] Zhang, D. Y., & Seelig, G. (2010, June). DNA-based fixed gain amplifiers and linear classifier circuits. In International Workshop on DNA-Based Computers (pp. 176-186). Springer, Berlin, Heidelberg.
[18] Chen, X., Briggs, N., McLain, J. R., & Ellington, A. D. (2013). Stacking nonenzymatic circuits for high signal gain. Proceedings of the National Academy of Sciences, 201222807.
[19] Qian, L., Winfree, E., & Bruck, J. (2011). Neural network computation with DNA strand displacement cascades. Nature, 475(7356), 368.
[20] Cherry, K. M., & Qian, L. (2018). Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. Nature, 1.
[21] Takahashi, K., Yaegashi, S., Kameda, A., & Hagiya, M. (2005). Chain reaction systems based on loop dissociation of DNA. In International Workshop on DNA-Based Computers (pp. 347-358). Springer, Berlin, Heidelberg.
[22] Zhang, C., Yang, J., & Xu, J. (2009). Circular DNA logic gates with strand displacement. Langmuir, 26(3), 1416-1419.
[23] Genot, A. J., Bath, J., & Turberfield, A. J. (2011). Reversible logic circuits made of DNA. Journal of the American Chemical Society, 133(50), 20080-20083.
[24] Seelig, G., Soloveichik, D., Zhang, D. Y., & Winfree, E. (2006). Enzyme-free nucleic acid logic circuits. science, 314(5805), 1585-1588.
[25] Qian, L., & Winfree, E. (2011). Scaling up digital circuit computation with DNA strand displacement cascades. Science, 332(6034), 1196-1201.
[26] Tikhomirov, G., Petersen, P., & Qian, L. (2017). Fractal assembly of micrometre-scale DNA origami arrays with arbitrary patterns. Nature, 552(7683), 67.
[27] Mohammed, A. M., & Schulman, R. (2013). Directing self-assembly of DNA nanotubes using programmable seeds. Nano letters, 13(9), 4006-4013.
[28] He, J. L., Wu, Z. S., Zhou, H., Wang, H. Q., Jiang, J. H., Shen, G. L., & Yu, R. Q. (2010). Fluorescence aptameric sensor for strand displacement amplification detection of cocaine. Analytical chemistry, 82(4), 1358-1364.
[29] Zhang, D. Y., & Seelig, G. (2011). Dynamic DNA nanotechnology using strand-displacement reactions. Nature chemistry, 3(2), 103.
[30] Thachuk, C., Winfree, E., & Soloveichik, D. (2015, August). Leakless DNA strand displacement systems. In International Workshop on DNA-Based Computers (pp. 133-153). Springer, Cham.
[31] Scalise, D., & Schulman, R. (2016). Emulating cellular automata in chemical reaction–diffusion networks. Natural Computing, 15(2), 197-214.
[32] Scalise, D., & Schulman, R. (2014). Designing modular reaction-diffusion programs for complex pattern formation. Technology, 2(01), 55-66.
[33] Rothemund, P. W. (2006). Folding DNA to create nanoscale shapes and patterns. Nature, 440(7082), 297.
[34] Ke, Y., Ong, L. L., Shih, W. M., & Yin, P. (2012). Three-dimensional structures self-assembled from DNA bricks. science, 338(6111), 1177-1183.
[35] Marras, A. E., Zhou, L., Su, H. J., & Castro, C. E. (2015). Programmable motion of DNA origami mechanisms. Proceedings of the National Academy of Sciences, 112(3), 713-718.
[36] Zhang, David Yu, Hariadi, R.F., Choi, H.M. and Winfree, E. "Integrating DNA strand-displacement circuitry with DNA tile self-assembly." Nature communications, 4 (2013), 1965.
[37] Chirieleison, S. M., Allen, P. B., Simpson, Z. B., Ellington, A. D., & Chen, X. (2013). Pattern transformation with DNA circuits. Nature chemistry, 5(12), 1000.
[38] Cangialosi, Angelo, Yoon, C., Liu, J., Huang, Q., Guo, J., Nguyen, T.D., Gracias, D.H. and Schulman, R. "DNA sequence–directed shape change of photopatterned hydrogels via high-degree swelling." Science 357.6356 (2017): 1126-1130.

[39] Douglas, S. M., Bachelet, I., & Church, G. M. (2012). A logic-gated nanorobot for targeted transport of molecular payloads. Science, 335(6070), 831-834.
[40] Gatto, B., M. Palumbo, and C. Sissi. "Nucleic acid aptamers based on the G-quadruplex structure: therapeutic and diagnostic potential." Current medicinal chemistry 16.10 (2009): 1248-1265.
[41] Liu, Yan, Lin, C., Li, H. and Yan, H. "Aptamer-directed self-assembly of protein arrays on a DNA nanostructure." Angewandte Chemie 117.28 (2005): 4407-4412.
[42] Wilner, O. I., Weizmann, Y., Gill, R., Lioubashevski, O., Freeman, R., & Willner, I. (2009). Enzyme cascades activated on topologically programmed DNA scaffolds. Nature nanotechnology, 4(4), 249.
[43] Huizenga, David E., and Jack W. Szostak. "A DNA aptamer that binds adenosine and ATP." Biochemistry 34 (1995): 656-656.
[44] Carlier, M. F., & Shekhar, S. (2017). Global treadmilling coordinates actin turnover and controls the size of actin networks. Nature Reviews Molecular Cell Biology, 18(6), 389.
[45] Sourjik, V. (2004). Receptor clustering and signal processing in E. coli chemotaxis. Trends in microbiology, 12(12), 569-576.

8 | Supplemental Information: DNA Strand Buffers:

8.1 | Buffer reactions of orders 0 through 2

In principle, a reversible reaction of any order can act as a buffer, provided (1) that one of the products is the molecule X, whose concentration we wish to regulate or buffer, (2) all of the other species besides X are present at high concentration relative to the equilibrium concentration of X, (3) we have sufficient control over the forward and reverse rate constants (*e.g.* for a pH buffer we require a weak acid with a low dissociation constant) to tune the buffer. We decided to implement a buffer using two reactants and two products because this reaction form provides the ability to finely tune both the forward and reverse reaction rates by adjusting the appropriate reactant concentrations, and because the buffering reaction occurs only when the reactants are mixed, making it straightforward to characterize buffering kinetics. Below is a table describing the generalized buffer reactions up to the second order, with a brief summary of the benefits of each form.

SI Table 1: Buffer reaction orders

Reaction Order		Buffering Reaction	Equilibrium $[X]_{eq}$	Concentration Requirements	Notes
Left side	Right side				

0th	1st	$\overset{k_f}{\underset{k_r}{\rightleftharpoons}} X$	$\frac{k_f}{k_r}$	-	Infinite capacity. Equivalent to a proportional controller ($gain = \frac{1}{k_r}$, reference=$\frac{k_f}{k_r}$).
1st	1st	$A \overset{k_f}{\underset{k_r}{\rightleftharpoons}} X$	$\frac{k_f[A]}{k_r}$	$[A]_0 \gg [X]_{eq}$	Allows the forward rate to be tuned by changing $[A]_0$
2nd	1st	$A + B \overset{k_f}{\underset{k_r}{\rightleftharpoons}} X$	$\frac{k_f[A][B]}{k_r}$	$[A]_0, [B]_0 \gg [X]_{eq}$	Prevents the reaction from occurring until A and B are mixed.
0th	2nd	$\overset{k_f}{\underset{k_r}{\rightleftharpoons}} X + C$	$\frac{k_f}{k_r[C]}$	$[C]_0 \gg [X]_{eq}$	Allows the reverse rate to be tuned by changing $[C]_0$
1st	2nd	$A \overset{k_f}{\underset{k_r}{\rightleftharpoons}} X + C$	$\frac{k_f[A]}{k_r[C]}$	$[A]_0, [C]_0 \gg [X]_{eq}$	Standard form for an acid-base pH buffer

2nd	2nd	$A + B \underset{k_r}{\overset{k_f}{\rightleftharpoons}} X + C$	$\frac{k_f[A][B]}{k_r[C]}$	$[A]_0, [B]_0, [C]_0 \gg [X]_{eq}$	Bimolecular form explored in the main text.

8.2 | Derivations of setpoint concentration, relaxation time, and capacity

The governing equation for our bimolecular buffer reaction is:

$$S + I \underset{k_N}{\overset{k_S}{\rightleftharpoons}} N + X \qquad \text{(SI.Eqn 1)}$$

where:

X is the buffered molecule,

S is the source that contains X in an inactive state,

I is the initiator that releases X from S, and

N is the sink that recaptures X.

The mass action ordinary differential equation that governs the kinetics of this reaction is

$$\frac{d[X]}{dt} = k_S[S][I] - k_N[N][X]. \qquad \text{(SI.Eqn 2)}$$

The reaction in SI.Eqn 1 approaches an equilibrium concentration of

$$[X]_{eq} = \frac{k_S}{k_N}\frac{[S]_{eq}[I]_{eq}}{[N]_{eq}} = K_{eq}\frac{[S]_{eq}[I]_{eq}}{[N]_{eq}} \qquad \text{(SI.Eqn 3)}$$

where the equilibrium constant K_{eq} is defined as:

$$K_{eq} \equiv \frac{k_S}{k_N} = \frac{[N]_{eq}[X]_{eq}}{[S]_{eq}[I]_{eq}}. \quad \text{(SI.Eqn 4)}$$

8.2.1 | Setpoint concentration

By stoichiometry, we have the following conservation equations:

$$[S]_{eq} = [S]_0 - ([X]_{eq} - [X]_0) \quad \text{(SI.Eqn 5)}$$

$$[I]_{eq} = [I]_0 - ([X]_{eq} - [X]_0) \quad \text{(SI.Eqn 6)}$$

$$[N]_{eq} = [N]_0 + ([X]_{eq} - [X]_0) \quad \text{(SI.Eqn 7)}$$

For very large initial reactant concentrations $([S]_0, [I]_0, [N]_0 \gg |[X]_{eq} - [X]_0|)$. In this case the concentrations of S, I and N can be approximated as roughly constant $(i.e. [S]_{eq} \approx [S]_0, [I]_{eq} \approx [I]_0, [N]_{eq} \approx [N]_0)$. This reduces Eqn. SI.Eqn 3 to

$$[X]_{eq} \approx K_{eq} \frac{[S]_0[I]_0}{[N]_0} \quad \text{(SI.Eqn 8)}$$

8.2.2 | Relaxation time

If we assume that the initial reactant concentrations are sufficiently high, such that they do not change significantly as the system approaches equilibrium (*i.e.* $[S]_0, [I]_0, [N]_0 \gg [X]_{eq}$), then we can treat [S], [I] and [N] as approximately constant, which reduces Eqn. SI.Eqn 2 to

$$\frac{d[X]}{dt} = k_p - k_d[X] \qquad \text{(SI.Eqn 9)}$$

where

$$k_p \equiv Production\ Rate \equiv k_S[S][I] \qquad \text{(SI.Eqn 10)}$$

$$k_d \equiv Degradation\ Rate \equiv k_N[N] \qquad \text{(SI.Eqn 11)}$$

The solution to SI.Eqn 9 is

$$[X](t) = [X]_{set}\left(1 + \varepsilon e^{-\frac{t}{\tau}}\right) \qquad \text{(SI.Eqn 12)}$$

where

$$[X]_{set} \equiv equilibrium\ setpoint \equiv \frac{k_p}{k_d} \qquad \text{(SI.Eqn 13)}$$

$$\varepsilon \equiv relative\ offset\ from\ setpoint \equiv \frac{[X]_0 - [X]_{set}}{[X]_{set}} \qquad \text{(SI.Eqn 14)}$$

$$\tau \equiv time\ constant \equiv \frac{1}{k_d} \quad \text{(SI.Eqn 15)}$$

We can use SI.Eqn 12 to solve for the time it takes to relax to an arbitrary factor $\alpha < 1$ of the initial offset:

$$(1 + \alpha\varepsilon)[\mathrm{X}]_{set} = [\mathrm{X}]_{set}\left(1 + \varepsilon e^{-\frac{t}{\tau}}\right) \quad \text{(SI.Eqn 16)}$$

which simplifies to

$$\alpha = e^{-\frac{t}{\tau}} \quad \text{(SI.Eqn 17)}$$

which can be solved to find

$$t_{relax,\alpha} = -\tau ln(\alpha) \quad \text{(SI.Eqn 18)}$$

Alternatively, we can set $t = \tau$ and solve SI.Eqn 18 to find that each time constant τ is the amount of time to relax to a factor of

$$\alpha_{t=\tau} = \frac{1}{e} \quad \text{(SI.Eqn 19)}$$

8.2.3 | Capacity

To calculate the capacity of the buffer, we first render Eqn. SI.Eqn 3 dimensionless

$$[X]_{eq} \equiv [X^*]_{eq}[]_c \qquad \text{(SI.Eqn 20)}$$

which gives

$$[X^*]_{eq} = K_{eq}\frac{[S]_{eq}[I]_{eq}}{[N]_{eq}[]_c} \qquad \text{(SI.Eqn 21)}$$

where $[]_c$ is a concentration scale. We can now define the following useful quantities, analogous to pH and pKa in an acid base buffer. Note that this operation is not permissible without rendering X dimensionless, as logarithms can only operate on dimensionless numbers.

$$pX \equiv -log_{10}([X^*]_{eq}) \qquad \text{(SI.Eqn 22)}$$

$$pK_{eq} \equiv -log_{10}(K_{eq}) \qquad \text{(SI.Eqn 23)}$$

Taking the $-\log_{10}$ of both sides of SI.Eqn 21, and rearranging, we find

$$pX = pK_{eq} + log_{10}\left(\frac{[N]_{eq}[]_c}{[S]_{eq}[I]_{eq}}\right) \qquad \text{(SI.Eqn 24)}$$

which is analogous to the Henderson-Hasselbalch equation for acid-base buffers. We define the capacity β as the concentration of X that must be added to the system to change the pX by a given offset γ. Any γ could be selected to define capacity, depending on the magnitude of offset that is relevant to an experiment. In acid base buffers, $\gamma = \pm 1$ pH units is often used to define the buffer capacity, but here we are generally interested in a smaller range. There are two directions to change

pX, and therefore two capacities, specifically:

Positive capacity $\beta^+ \equiv [X]$ added to decrease pX_{final} to $pX_{initial} + \gamma$, for $\gamma < 0$ (SI.Eqn 25)

Negative capacity $\beta^- \equiv [X]$ removed to increase pX_{final} to $pX_{initial} + \gamma$, for $\gamma > 0$ (SI.Eqn 26)

Note that the only way to change pX is through the second term on the right hand side of SI.Eqn 24, which we define as:

$$\delta_{eq} \equiv log_{10}\left(\frac{[]_c[N]_{eq}}{[S]_{eq}[I]_{eq}}\right) \quad \text{(SI.Eqn 27)}$$

Therefore the positive and negative capacity definitions from SI.Eqn's 25-26 can be restated as the concentrations of X that must be added or removed to change $\delta_f = \delta_{eq} + \gamma$. For clarity, we will use the following subscripts:

0: initial state of buffer before reaching equilibration,

eq: at initial equilibrium,

f: post-disturbance equilibrium

Due to the high concentrations of S, I and N relative to the equilibrium concentration of X, we assume that the concentrations of S, I and N do not change significantly during the initial equilibration, *i.e.* $[S, I, N]_0 \approx [S, I, N]_{eq}$. Furthermore, we will assume that if a disturbance of $\propto \equiv [X]_{added} = -[Competitor]_{added}$ is applied to the buffer, then the concentrations of S, I and

N change linearly as follows: $[S,I]_f = [S,I]_0 + \propto$, and $[N]_f = [N]_0 - \propto$. With these assumptions:

$$\delta_f = \delta_{eq} + \gamma = log_{10}\left(\frac{[]_c([N]_0 - \propto)}{([S]_0 + \propto)([I]_0 + \propto)}\right) \qquad \text{(SI.Eqn 28)}$$

Exponentiating both sides of SI.Eqn 28 with base 10, and rearranging into quadratic form, we have:

$$a \propto^2 + b \propto + c = 0 \qquad \text{(SI.Eqn 29)}$$

where:

$$a = 1$$

$$b = [P]_0 + \frac{[]_c}{10^{\delta_{eq}+\gamma}}$$

$$[P]_0 \equiv [S]_0 + [I]_0 \qquad \text{(SI.Eqn 30)}$$

$$c = [S]_0[I]_0 - \frac{[]_c[N]_0}{10^{\delta_{eq}+\gamma}}$$

With a =1, the quadratic formula gives the solution to SI.Eqn. 29 as:

$$\propto = \frac{- \quad \pm\sqrt{b^2 - 4c}}{2} \qquad \text{(SI.Eqn. 31)}$$

which we can substitute the values in SI.Eqn. 30 back in to find:

$$\propto = \frac{-[P]_0 - \frac{[]_c}{10^{\delta_{eq}+\gamma}} \pm \sqrt{[P]_0{}^2 + 2[P]_0\frac{[]_c}{10^{\delta_{eq}+\gamma}} + \frac{[]_c^2}{\left(10^{\delta_{eq}+\gamma}\right)^2} - 4[S]_0[I]_0 + 4\frac{[]_c[N]_0}{10^{\delta_{eq}+\gamma}}}}{2}$$

notice $$10^{\delta_{eq}+\gamma} = 10^{\gamma} \cdot \frac{[]_c[N]_0}{[S]_0[I]_0} \quad \text{(SI.Eqn. 32)}$$

therefore $$\left(10^{\delta_{eq}+\gamma}\right)^2 = 10^{2\gamma} \cdot \frac{[]_c{}^2[N]_0^2}{[S]_0^2[I]_0^2} \quad \text{(SI.Eqn. 33)}$$

which we can substitute back in to find:

$$\propto = -\frac{[P]_0}{2} - \frac{[S]_0[I]_0}{2 \cdot 10^{\gamma} \cdot [N]_0} \pm \sqrt{\frac{[P]_0{}^2}{4} + \left(\frac{[P]_0 + 2[N]_0}{2 \cdot 10^{\gamma} \cdot [N]_0} - 1\right)[S]_0[I]_0 + \frac{[S]_0^2[I]_0^2}{4 \cdot 10^{2\gamma} \cdot [N]_0^2}}$$

Applying the following definitions:

$$[Total]_0 = [S]_0 + [I]_0 + [N]_0$$

$$\phi_P \equiv \frac{[P]_0}{[P]_0 + [N]_0} = \frac{[P]_0}{[Total]_0}$$

$$\phi_S \equiv \frac{[S]_0}{[S]_0 + [I]_0}$$

$$[P]_0 = \phi_P[Total]_0$$

$$[N]_0 = (1 - \phi_P)[Total]_0$$

$$[S]_0 = \phi_S[P]_0 = \phi_S\phi_P[Total]_0$$

$$[I]_0 = (1 - \phi_S)[\ \ ]_0 = (1 - \phi_S)\phi_P[Total]_0$$

we have:

$$\frac{\propto}{\phi_P[Total]_0} = -\frac{1}{2} - \frac{\phi_S\phi_P - {\phi_s}^2\phi_P}{2 \cdot 10^{\gamma} \cdot (1-\phi_P)}$$

$$\pm\sqrt{\frac{1}{4} + \left(\frac{\phi_P + 2(1-\phi_P)}{2 \cdot 10^{\gamma} \cdot (1-\phi_P)} - 1\right)(\phi_S - {\phi_S}^2) + \frac{{\phi_S}^2{\phi_P}^2(1-\phi_S)^2}{4 \cdot 10^{2\gamma} \cdot (1-\phi_P)^2}}$$

which is a general form of the capacity equation, normalized by the initial concentration of S+N. In this paper we primarily explored the case where $\phi_S = 0.5$, *i.e.* $[S]_0 = [I]_0$, in which case we can simplify to:

$$\frac{2 \propto}{\phi_P[Total]_0} = -1 - \frac{0.5\phi_P}{2 \cdot 10^{\gamma} \cdot (1-\phi_P)}$$

$$\pm\sqrt{\frac{1}{10^{\gamma} \cdot (1-\phi_P)} - \frac{\phi_P}{2 \cdot 10^{\gamma} \cdot (1-\phi_P)} + \frac{{\phi_P}^2}{16 \cdot 10^{2\gamma} \cdot (1-\phi_P)^2}}$$

If we select $\gamma \pm 1$, *i.e.* the capacity to change the equilibrium by a factor of ten from the setpoint concentration, this gives a negative capacity ($\gamma = +1, \beta^- \equiv -\infty$)) of

$$\beta^- = \frac{\phi_P}{2}\left(1 + \frac{\phi_P}{40 \cdot (1-\phi_P)} \pm \sqrt{\frac{1}{10 \cdot (1-\phi_P)} - \frac{\phi_P}{20 \cdot (1-\phi_P)} + \frac{{\phi_P}^2}{1600 \cdot (1-\phi_P)^2}}\right)[Total]_0$$

and a positive capacity ($\gamma = -1, \beta^+ \equiv \infty$) of

$$\beta^+ = \frac{\phi_P}{2}\left(-1 - \frac{\phi_P}{0.4 \cdot (1-\phi_P)} \pm \sqrt{\frac{10}{(1-\phi_P)} - \frac{5\phi_P}{(1-\phi_P)} + \frac{6.25{\phi_P}^2}{(1-\phi_P)^2}}\right)[Total]_0$$

If we are interested in the capacity for smaller changes to the equilibrium concentration, we select a

different γ value. For instance for the negative capacity to shift the equilbrium by a factor of 0.1 in either direction (*i.e.* $[X]_{eq}=0.9[X]_{set}$ and . $[X]_{eq}=1.1[X]_{set}$) we find a negative capacity ($\gamma = \log_{10}\left(\frac{1}{0.9}\right), \beta^- \equiv -\infty$) of:

$$\frac{-2\beta^-}{\phi_P[\ \ otal]_0} = -1 - \frac{0.5\phi_P}{2\cdot(1/0.9)\cdot(1-\phi_P)} \pm \sqrt{\frac{1}{(\frac{1}{0.9})\cdot(1-\phi_P)} - \frac{\phi_P}{2\cdot(\frac{1}{0.9})\cdot(1-\phi_P)} + \frac{{\phi_P}^2}{16\cdot 10^{2\log(\frac{1}{0.9})}\cdot(1-\phi_P)^2}}$$

and a positive capacity ($\gamma = \log_{10}\left(\frac{1}{1.1}\right), \beta^+ \equiv \infty$) of:

$$\frac{2\beta^+}{\phi_P[Tota\ \ ]_0} = -1 - \frac{0.5\phi_P}{2\cdot(1/1.1)\cdot(1-\phi_P)} \pm \sqrt{\frac{1}{(\frac{1}{1.1})\cdot(1-\phi_P)} - \frac{\phi_P}{2\cdot(\frac{1}{1.1})\cdot(1-\phi_P)} + \frac{{\phi_P}^2}{16\cdot 10^{2\log(\frac{1}{1.1})}\cdot(1-\phi_P)^2}}$$

These capacities can be rewritten as:

$$\beta^- = c^-\cdot[Total]_0$$

$$\beta^+ = c^+\cdot[Tota\ \ ]_0$$

where for large 100% offsets ($\gamma = \pm 1$):

$$c^- \equiv \frac{\phi_P}{2}\left(1 + \frac{\phi_P}{40\cdot(1-\phi_P)} \pm \sqrt{\frac{1}{10\cdot(1-\phi_P)} - \frac{\phi_P}{20\cdot(1-\phi_P)} + \frac{{\phi_P}^2}{1600\cdot(1-\phi_P)^2}}\right)$$

$$c^{+} \equiv \frac{\phi_P}{2}\left(-1-\frac{\phi_P}{0.4\cdot(1-\phi_P)} \pm \sqrt{\frac{10}{(1-\phi_P)}-\frac{5\phi_P}{(1-\phi_P)}+\frac{6.25{\phi_P}^2}{(1-\phi_P)^2}}\right)$$

The $\pm$ term in the expression for c^{+} must be +, to ensure that $\beta^{+} > 0$. Similarly, the $\pm$ term in the expression for c^{-} must be -, to ensure that $\beta^{-} < \min([S]_0, [I]_0)$. For smaller 10% offsets ($\gamma = \log_{10}\left(\frac{1}{0.9}\right), or\ \log_{10}\left(\frac{1}{1.1}\right)$)

$$c^{-} \equiv \frac{\phi_P}{2}\left(\frac{1+\frac{0.5\phi_P}{2\cdot\left(\frac{1}{0.9}\right)\cdot(1-\phi_P)}}{\pm\sqrt{\frac{1}{(\frac{1}{0.9})\cdot(1-\phi_P)}-\frac{\phi_P}{2\cdot(\frac{1}{0.9})\cdot(1-\phi_P)}+\frac{{\phi_P}^2}{16\cdot 10^{2\log(\frac{1}{0.9})}\cdot(1-\phi_P)^2}}}\right)$$

$$c^{+} \equiv \frac{\phi_P}{2}\left(\frac{-1-\frac{0.5\phi_P}{2\cdot\left(\frac{1}{1.1}\right)\cdot(1-\phi_P)}}{\pm\sqrt{\frac{1}{(\frac{1}{1.1})\cdot(1-\phi_P)}-\frac{\phi_P}{2\cdot(\frac{1}{1.1})\cdot(1-\phi_P)}+\frac{{\phi_P}^2}{16\cdot 10^{2\log(\frac{1}{1.1})}\cdot(1-\phi_P)^2}}}\right)$$

8.3 | Detailed DNA Strand-Displacement Diagram for the Zero Nucleotide Toehold Reaction

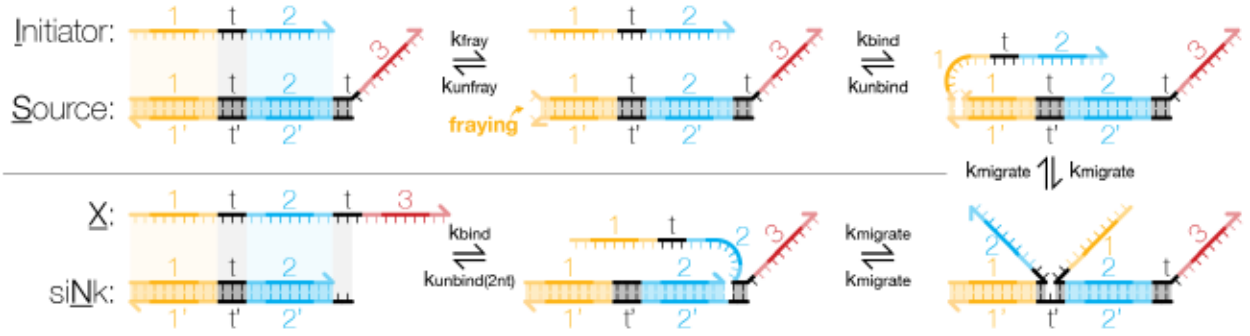

Figure 8.1. Detailed diagram for 0-nucleotide reaction between the slow Source and slow Initiator, with a 0 nucleotide toehold (*i.e.* no toehold) on the forward reaction. The reaction is initiated when the end base pair on the Source complex frays open, effectively creating a transient 1-nucleotide "toehold" for the Initiator to bind. Branch migration and displacement of the signal strand X then proceeds as usual for an ordinary toehold-mediated strand-displacement reaction. Signal X and sink N are produced, and the reaction is reversible.

8.4 | Sequences, Methods, and Calibrations

8.4.1 | Sequences

The sequences of our domains were drawn from Table S1 of the Supporting Online Material for reference SI-R1. Secondary structure of was verified for each complex using NuPack[SI-R2]. Sequences for each strand used in the study are listed in SI Table 2. The domain level structure of the fast buffer (with toeholds extended from 0nt to 1nt for the Source, and 2nt to 4nt for the Sink) is illustrated in Fig. SI2.

SI Table 2: Sequence Data

Strand	Sequences	IDT Purification
Signal (X)	CA TAACA CA TCT CA CAATC CA TCT CA	PAGE

	CCACC CA	
Source Bottom	GATG GATTG TG AGA TG TGTTA TG	PAGE
Initiator	CA TAACA CA TCT CA CAATC CA	PAGE
Reporter Top (R_Q)	CAATC CA TCT CA CCACC CA TCT CA/3IABkFQ/	HPLC
Reporter Bottom (R_{bF})	/56-FAM/TG AGA TG GGTGG TG AGA TG GATTG TG AGA	HPLC
Reporter Full Complement	TCT CA CAATC CA TCT CA CCACC CA TCT CA	Standard desalting
Competitor Top	CA CAATC CA TCT CA CCACC CA CT	Standard desalting
Competitor Bottom	AG TG GGTGG TG AGA TG GATTG TG AGA TG TG	Standard desalting
$Signal_2$ (X_2)	CAA TCT ACA TCT CAA CAC TCA TCT CAT TCC TCA	PAGE
$Source_2$ Bottom	GAT GAG TGT TGA GAT GTA GAT TG	PAGE

Initiator$_2$	CAA TCT ACA TCT CAA CAC TCA	PAGE
Reporter$_2$ Top	CA ACACT CA TCT CA TTCCT CA TCT CA/3IABkFQ/	HPLC
Reporter$_2$ Bottom	/5HEX/TG AGA TG AGGAA TG AGA TG AGTGT TG AGA TG	HPLC
Reporter$_2$Full Complement	CA TCT CA ACACT CA TCT CA TTCCT CA TCT CA	Standard desalting
Initiator (Fast)	C CA TAACA CA TCT CA CAATC	PAGE
Source Bottom (Fast)	GA TG GATTG TG AGA TG TGTTA TG GT	PAGE

Figure 8.2. Fast buffer reaction diagram, showing the 1-nucleotide toehold on Source(FAST) that drives the forward reaction shown here faster than the forward reaction in main text Figure 3a, and the 4-nucleotide Sink(FAST) toehold that drives the reverse reaction shown here faster than the comparable reverse reaction in main text Figure 3a. Making k_{sink} faster decreases the response time of the buffer.

8.4.2 | Experimental Methods

All DNA strands were obtained from Integrated DNA Technologies (IDT), using the

purification options listed in SI Table 1. On arrival all strands were suspended in Millipore purified water at a concentration of about 1mM and stored at -20°C. Stock concentrations were determined by measuring the absorbance of light at a wavelength of 260nm (OD260), together with the extinction coefficient for each strand provided by IDT (EXT), using the Beer-Lambert law: [ssDNA]=OD260/EXT.

We prepared each of the double-stranded complexes (Source, Sink, Reporter, and Threshold) separately at 100 μM in Tris-acetate-EDTA buffer with 12.5 mM Mg++ (1x TAE/Mg++). Source and Sink complexes were prepared with a 1.2x excess of top strand (*i.e.* 120 μM Signal X for the Source, and 120 μM Initiator for the Sink) to ensure all bottom strands were occupied by a top strand. The Reporter complex was prepared with a 2x excess of top strand (*i.e.* 200 μM R_Q), which helped reverse biased the reporting reaction to report on higher concentrations of Signal X. We then annealed them in an Eppendorf Mastercycler PCR, first heating the solutions to 90°C, holding the temperature constant for 5 minutes, and then cooling at -0.1°C per every 6 seconds down to 20°C. Source and Sink complexes were purified by polyacrylamide gel electrophoresis (PAGE). Reporter and Threshold were not gel purified.

For gel purification, we cast 15% polyacrylamide gels by mixing 3.25mL of 19:1 40% acrylamide/bis solution (Bio-Rad) with 1.3mL 10x TAE/Mg++ and 8.45mL Millipore-purified H2O, and initiated polymerization with 75μL 10% ammonium persulfate (APS, Sigma Aldrich) and 7.5μL tetramethylethylenediamine (TEMED, Sigma Aldrich). We mixed 200 μL of dsDNA complex with 6x loading dye (New England Biolabs, product #B7021S) and loaded into a Scie Plas TV100K cooled vertical electrophoresis chamber. We ran our gels at 150V and 4°C for 3 hours and then cut out the purified bands using UV-shadowing at 254nm[SI-R1]. The gel bands were chopped into small pieces, mixed with 300μL of 1x TAE/Mg++ buffer, and then left on a lab bench overnight to allow

the DNA to diffuse out of the gel into the buffer. The next day, the buffer was transferred by pipet to a fresh tube, leaving behind as much of the gel as possible. These fresh tubes were centrifuged for 5 minutes to draw any remaining gel pieces to the bottom of the tube, and then transferred to yet another fresh tube, leaving behind ~50μL of gel/solution at the bottom. The concentrations of these purified complexes were then measured with an Eppendorf Biophotometer, using the approximate extinction coefficient $EXT=EXT_{top\_strand}+EXT_{bottom\_strand}-3200N_{AT}-2000N_{GC}$, where N_{AT} and N_{GC} are the number of hybridized A-T and G-C pairs in each complex, respectively.[SI-R3] Purified complexes were stored at 4°C.

Reaction kinetics were measured on quantitative PCR (qPCR) machines (Agilent Stratagene Mx3000 and Mx3005 series) at 25°C. Fluorescence was typically measured every 30 seconds for baseline measurements and for the first 1-2 hours after a reaction was triggered by adding Initiator, to accurately capture the early kinetics of a reaction, and then every 5 minutes for the remainder of the experiment to avoid photobleaching the fluorophore. Reactions were prepared in 96-well plates using 50μL/well volume. Each well contained 1x TAE/Mg++ and 1 μM of 20-mer PolyT strands to help displace reactant species from the pipet tips used to add them to the well. In a typical experiment, Millipore-purified H_2O, TAE/Mg++ and $PolyT_{20}$ strands were first mixed together. Reporter was then added and a measurement of the baseline reporter fluorescence was taken to determine what fluorescence corresponded to the state of the system with zero output signal concentration added. We then any other DNA reactant species, in the amounts specified for each experiment, and tracked the resulting kinetics.

8.4.3 | Reporter Calibrations and Data Processing

Concentrations of X reported in Figures 3-4 were determined using a reporter complex that reacts reversibly with X, resulting in a change in fluorescence. Rhe unreacted reporter has a

quenched fluorophore, while the reacted form R_F has an unquenched fluorophore. The change in fluorescence thus reflects the concentration of X.

To determine [X] from raw fluorescence values we used a set of empirical calibration curves that we measured. To create these curves, we first measured a time-dependent scaling factor $\propto(t)$ that relates the measured fluorescence intensity, in counts per second (CPS), to the concentration of unquenched reporter complex [R_F] in solution (see R_F in main text Fig. 3b) at a given time during an experiment. Time dependence of this factor results from fluctuations in the light or detector that affect all samples. To measure $\propto(t)$, we mixed 100nM of Reporter with different concentrations of the Reporter's full complement (see SI Table 1). This full complement (FC) reacts irreversibly with the Reporter, producing an unquenched product we call R_{F2}. Thus, the concentration of R_{F2} should be equal to the concentration of full complement that was added (Fig. SI3a). We assume the fluorescence of R_F is equal to the fluorescence if R_{F2}.

For each trajectory, we then calculate $\propto$ as a function of time, which accounts for fluctuations in lamp intensity.

$$\propto(t) \equiv \frac{[FullComplement]}{\Delta RawCPS(t)}$$

$\Delta RawCPS(t)$ is the difference between the fluorescence intensity at time t and the fluorescence intensity before the Full Complement strand is added. We take the average $\overline{\propto(t)}$ for five different full complement trajectories in an experiment. We then use this factor to calculate the concentration of R_F in all other experiments as follows:

$$[R_F](t) \equiv \Delta RawCPS(t) \cdot \overline{\propto(t)}$$

We calculated $\propto(t)$ separately every time we ran an experiment, to take into account variations in

lamp intensity that are specific to each experiment.

Next, we determined the time-independent relationship between [R_F] and [X]. We mixed 100nM of Reporter with different concentrations of X, which reacts reversibly with Reporter to separate the R_F and R_Q species. We then fit this data to a calibration curve, of the form $[X] = c_1 \cdot ([R_F])^{c_2}$, where c_1 and c_2 are the fit parameters (Fig. SI3b). Separate curves were fit for each set of experiments run with different batches of reporter, to account for batch-to-batch variation.

With these transformations, we can calculate [X] in our subsequent experiments as:

$$[X](t) = c_1 \cdot (\Delta RawCP \quad (t) \cdot \propto (t))^{c_2}$$

for the $\propto (t)$, c_1 and c_2 determined for the given experiment.

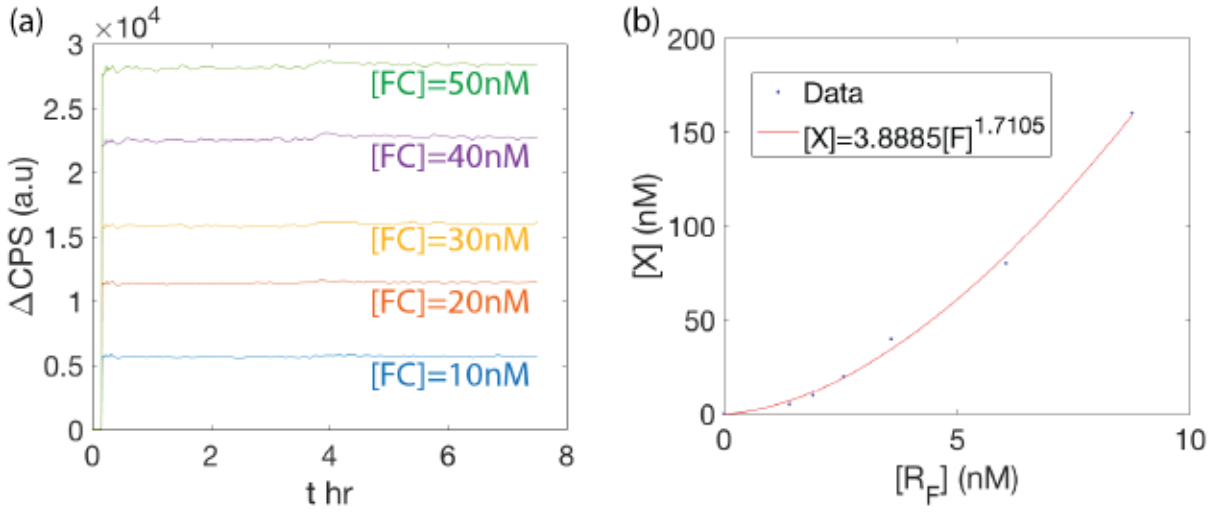

Figure 8.3. Reporter Calibrations. (a) Full complement calibration to convert from raw fluorescence in counts per second to the concentration of unquenched fluorophore [R_F] in solution, (b) Reverse calibration to convert from [R_F] to [X].

8.5 | Selecting a Reporter Toehold Length

To use a reporting reaction to measure the kinetics of buffering, we need to ensure that the

reporter equilibrates at least as fast as the buffering reaction. We verified that we chose a reporter whose reaction rate was fast enough to measure our buffer by modeling both the buffer and reporter as simple bimolecular reaction using estimates for bimolecular reaction rate constants as a function of toehold length from reference SI-R4. We used SI Eqn. 1 for the buffer, and SI Eqn. 34 for the reporter:

$$X + R \underset{k_r}{\overset{k_f}{\rightleftharpoons}} R_F + R_Q \qquad \text{(SI Eqn. 34)}$$

We then compared the kinetics of the buffering reaction to the kinetics of the reporting reaction when coupled to the buffering reaction with the goal of seeing that the rates of the two reactions were very close to one other. To further ensure that the reporting reaction would not place a significant load on the reporting reaction, we also checked that our simulation predicted that the kinetics of the buffering reaction in the presence of the reporter would be similar to the kinetics of the same reaction without the reporting reaction.

The results of our simulation are shown in Figure SI4. For an 8μM uniform buffer with a 0nt source toehold and a 2nt sink toehold, we found that a reporter with a 5nt toehold was sufficient to equilibrate on the same timescale as the buffer shown in Figure 3a of the main text. (Fig. SI4a). In contrast, a 2nt reporter toehold equilibrates much slower than the buffer (Fig. SI4b). For the faster buffer with a 2nt source and 4nt sink toehold, the 5nt reporter complex did appear to marginally slow the equilibration process somewhat, indicating that the fast buffer could be equilibrating even faster than the reporter's change in fluorescence would indicate (Fig. SI4c), In none of these idealized bimolecular models did the concentration of the reporter used in our experiments place a significant load on the equilibrium state of the buffering reaction.

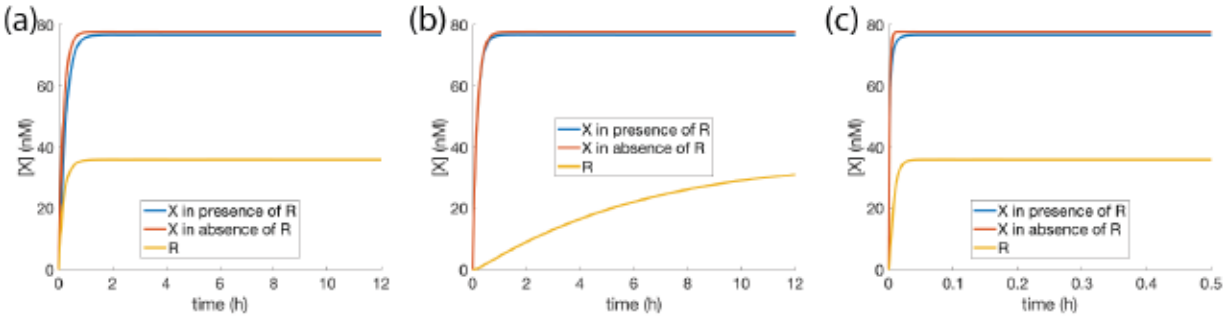

Figure 8.4. Comparison between reporters with different toehold lengths, using a simple bimolecular model to simulate the kinetics. For both simulations we used an 8μM uniform buffer with a 0nt source toehold and a 2nt sink toehold, with $[\text{Reporter}]_0=[R_Q]_0=100\text{nM}$. (a) A reporter with 5nt toeholds correctly recapitulates the kinetics of the buffering reaction. This is the reporter selected for the experiments in the main text. (b) A slower reporter with 2nt toeholds is not fast enough to recapitulate the kinetics of the buffering reaction. (c) The fast buffer (with 0nt source toehold, and 2nt sink toehold) equilibrates faster than the 5nt reporter indicates it does. We can therefore bound the maximum relaxation time, but did not measure it directly by using a reporter with 5nt toeholds.

8.6 | Using the detailed three-step model to predict setpoints, relaxation times, and the effects of reporter loading and toehold occlusion.

The three-step model[SI-R4] of DNA strand-displacement (DSD) offers a means to quantitatively estimate the kinetics of DSD reactions, to within an order of magnitude. It approximates each strand-displacement event as a series of three steps (i) toehold binding, (ii) branch migration, and (iii) toehold dissociation (Figure SI5).

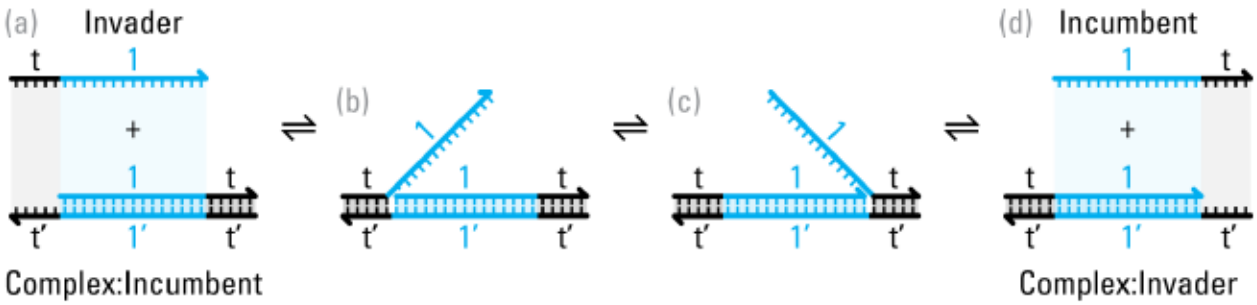

Figure 8.5. Three-step model of a DNA strand-displacement reaction, in which an input or invader strand displaces an incumbent or output strand from a complex [SI-R4]. (a) Invader is single-stranded, while the incumbent strand is initially hybridized to the complex. (b) Invader binds to the complex *via* a short unstable "toehold" domain (black, t), which initiates the displacement reaction. (c) In a random walk process, the invader strand competes with the incumbent strand to occupy the longer "recognition" domain (cyan, 1). (d) The incumbent strand, now bound only by the short unstable toehold domain, dissociates from the complex. All steps in the reaction are reversible. The lengths of the toehold domains determine the rate constants with which strands dissociate from the complex when bound only by the toehold. These rate constants determine the effective rate constant of the entire strand-displacement reaction.

8.6.1 | Reaching different concentration set points by altering the concentration of Source, Initiator and Sink

To better under the experiments in Fig. 4 in the main text where we characterized the buffering reactions for different concentrations of either Source and Initiator or Sink, we simulated these same reactions using the three-state model. These simulations matched the experimental data in main text Fig. 4 to within an order of magnitude, which given the model's lack of incorporation of sequence effects in determined reaction rate constants is about what might be expected. The simulations predicted slightly lower steady-state concentrations than were observed in experiments and slightly higher rise times for low concentrations of Sink than were observed in experiments (Fig 8.6).

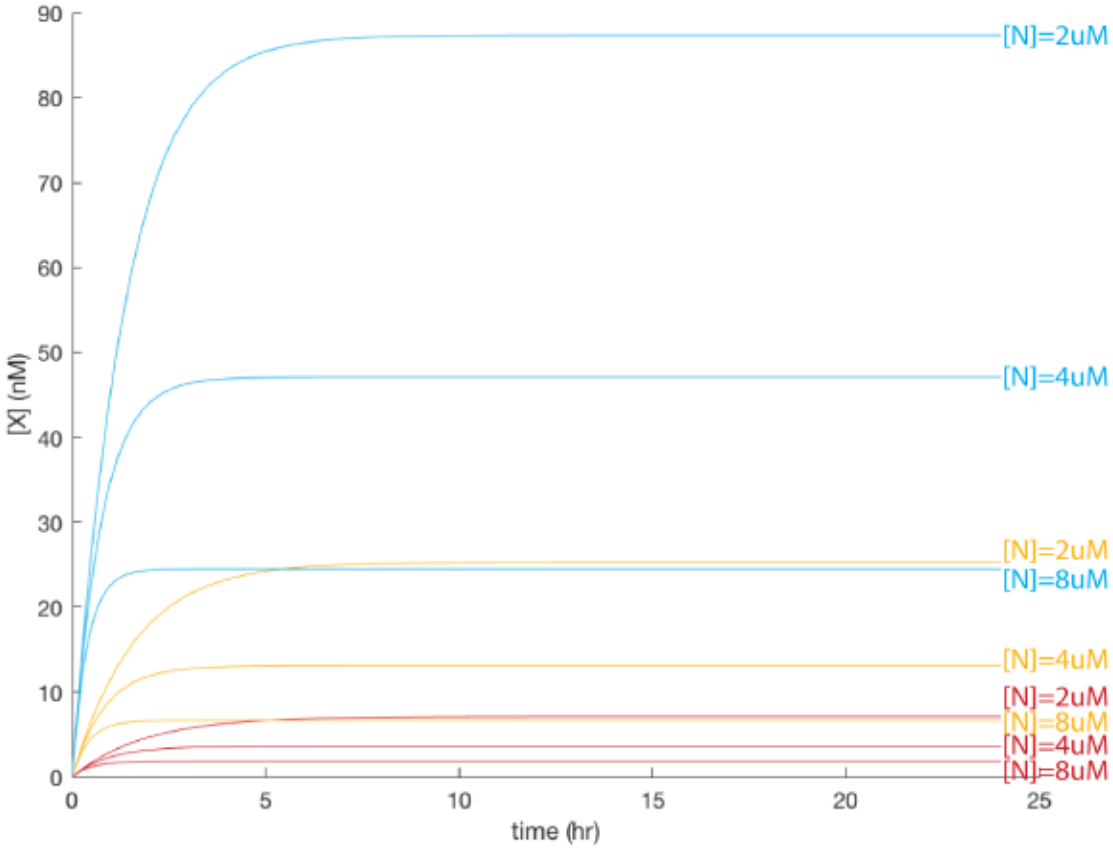

Figure 8.6. Three-step model predictions of buffer kinetics with different concentrations of Source, Initiator, and Sink. Blue lines correspond to buffers with $[S]_0=[I]_0=8uM$, yellow to $[S]_0=[I]_0=4uM$, and red to $[S]_0=[I]_0=2uM$. [Reporter]=100nM, with $[R_Q]=100nM$.

The Matlab script used to run the Three-Step Model simulations of buffering is printed below:

```
runTime=24*3600;%s

displacementBP = 15;%bp

kf = 3.5;%uM^-1 s^-1

kb = 400/(displacementBP^2);%s^-1
```

```
krFactor = 10^6;

RT=298*1.987*10^-3;

krNEG1bp = kf*krFactor*(2/displacementBP)*exp(3.5/RT);%s^-1

kr0bp = kf*krFactor*(2/displacementBP)*exp(1.9/RT);%s^-1

kr1bp = kf*krFactor*(2/displacementBP)*exp(0.2/RT);%s^-1

kr2bp = kf*krFactor*(2/displacementBP)*exp(-1.7/RT);%s^-1

kr3bp = kf*krFactor*(2/displacementBP)*exp(-3.0/RT);%s^-1

kr4bp = kf*krFactor*(2/displacementBP)*exp(-4.5/RT);%s^-1

kr5bp = kf*krFactor*(2/displacementBP)*exp(-6.9/RT);%s^-1

kr6bp = kf*krFactor*(2/displacementBP)*exp(-8.3/RT);%s^-1

kr7bp = kf*krFactor*(2/displacementBP)*exp(-9.2/RT);%s^-1



krRelease=kr0bp;

krRecapture=kr2bp;

initR=0.1;

initRiQ=initR;



[Si,id]=getID(1);[Ii,id]=getID(id);[SIi1,id]=getID(id);[SIi2,id]=getID(id);[Ni,id]=getID(id);[in,id]=get
```

```
ID(id);
[Ri,id]=getID(id);[Ri1,id]=getID(id);[Ri2,id]=getID(id);[RiF,id]=getID(id);[RiQ,id]=getID(id);
[IR1,id]=getID(id);[IR2,id]=getID(id);[IRf1,id]=getID(id);[IRf2,id]=getID(id);[IS1,id]=getID(id);[IS
2,id]=getID(id);


figure
hold on
ylabel('[X] (nM)');
xlabel('time (hr)');
for initN=[2,4,8]
    for initS=[2,4,8]
        disp(['Buffer Run: [I]=[S]=',num2str(initS),', [N]=',num2str(initN)]);
        dydt=@(t,y) rxn(y,[Si,Ii,SIi1,SIi2,Ni,in],21,krRelease,krRecapture)+...%Input Buffering
            rxn(y,[in,Ri,Ri1,Ri2,RiF,RiQ],19,kr5bp*15/19,kr5bp*15/19)+...%Input Reporting
            rxn(y,[Ii,Ri,IR1,IR2],7,kr5bp*15/7)+...%Initiator:Reporter toehold occlusion (forward)
            rxn(y,[Ii,RiF,IRf1,IRf2],2,kr5bp*15/2)+...%Initiator:RiF toehold occlusion (reverse)
            rxn(y,[Ii,Si,IS1,IS2],2,krRecapture*15/2)...%Initiator:Sink toehold occlusion
            ;
```

```
initC=[...
    initS,initS,0,0,initN,0,...
    initR,0,0,0,initRiQ,...
    0,0,0,0,0,0,...
    ]';

[time,y] = ode15s(dydt,[0,runTime],initC);

color=[0,174,239]/255;
if initS==4
    color=[252,180,21]/255;
elseif initS==2
    color=[203,32,39]/255;
end
concentrations=(y(:,in)+y(:,Ri1)+y(:,Ri2)+y(:,RiF)+y(:,IRf1)+y(:,IRf2))*1000;
plot(time/3600,concentrations,'Color',color);
disp('Steady State (nM)= ');
disp(concentrations(end));
```

```
        riseTime=time(find(concentrations>=0.8*concentrations(end),1))/3600;
        disp('rise time (hr)= ');
        disp(riseTime);


        disp(' ');
    end
end


function [var,id]=getID(id)
var=id;
id=id+1;
end


function dydt=rxn(y,species,displacementBP,krForward,krReverse)
%species=[reactant1,reactant2,intermediateReactantComplex,intermediateProductComplex,product
1,product2];
```

```
%units in uM and s

kf = 3.5;%uM^-1 s^-1

kb = 400/(displacementBP^2);%s^-1



dydt=zeros(length(y),1);

dydt(species(1))=-kf*y(species(1))*y(species(2))+krForward*y(species(3));

dydt(species(2))=dydt(species(1));

dydt(species(3))=kf*y(species(1))*y(species(2))-krForward*y(species(3))-
kb*y(species(3))+kb*y(species(4));

dydt(species(4))=kb*y(species(3))-kb*y(species(4));

if length(species)>4

    dydt(species(4))=dydt(species(4))-krReverse*y(species(4))+kf*y(species(5))*y(species(6));

    dydt(species(5))=krReverse*y(species(4))-kf*y(species(5))*y(species(6));

    dydt(species(6))=dydt(species(5));

end



end
```

8.6.2 | Relaxation Time Slows Down when Reporter is Included

We next used three-step model simulations (following the same script as above) to examine the effect of the reporter on the buffer's relaxation time. By comparing simulations of the buffer's equilibration with and without the reporter present, we observed the presence of the reporter may increase relaxation times (*i.e.* increase τ), and create a small dependence of the relaxation time constant τ on the initial concentrations of Source and Initiator (Fig. 8.7). These effects arise because it takes time for the buffer to generate or "charge up" the reporter and its intermediate products in the three-step model. Higher concentrations of Source and Initiator can generate X at a faster rate when the reporter is far from equilibrium, and thus can charge the reporter faster.

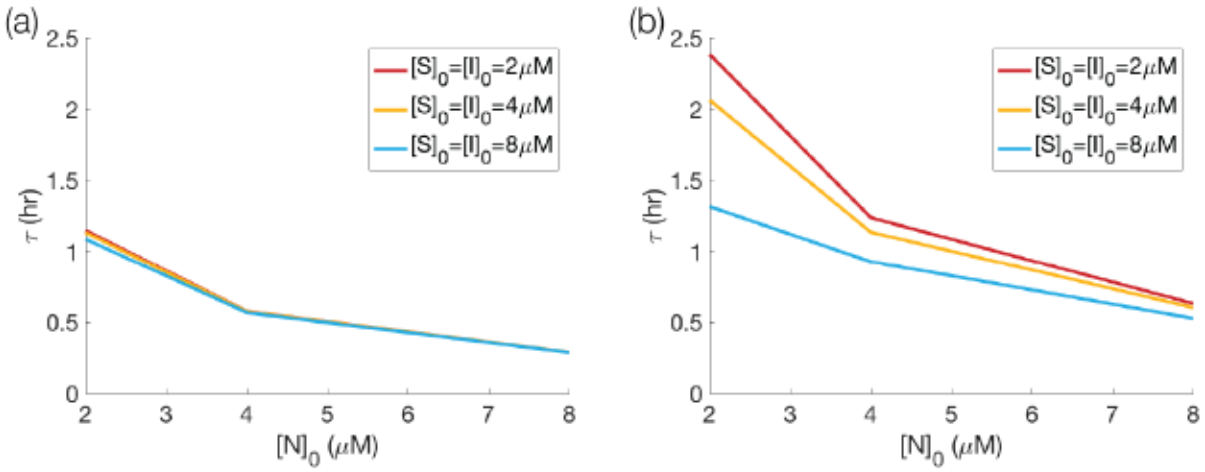

Figure 8.7. Three-Step Model Predictions of the relaxation time constant with and without the reporter present. (a) Time constants with no reporter, showing no significant dependence of the time constant on the concentrations of Source and Initiator. (b) Time constants with reporter included ($[Reporter]_0$=100nM, with $[R_Q]_0$=100nM), showing dependence of time constant on the initial concentrations of Source and Initiator. Note: toehold occlusion effects are not included in these simulations.

8.6.3 | Toehold Occlusion

DNA strand displacement (DSD) reactions can slow down significantly due to toehold

occlusion, the phenomenon in which one species temporarily binds to a complementary toehold on another species but cannot then fully displace the adjacent recognition domain. The rate at which toehold occlusion occurs increases with reactant concentration and toehold length, so it is particularly noticeable when there are high reactant concentrations involving long toeholds. In our DSD buffer design, the Initiator is present at high concentration and shares a long complementary 7nt toehold with the Competitor complex (Fig. SI8a) creating a potential for significant toehold occlusion involving the Initiator and Competitor. Initiator+Reporter and Initiator+R_F also have the potential for toehold occlusion, with a 5nt toehold. Initiator+Sink may also have some occlusion interactions, as both species are present at high concentration, although they only share a short 2nt toehold. All other possible occlusion interactions involve significantly lower reactant concentrations and/or smaller toeholds, and are not expected to play as significant a role in determining the buffer system's kinetics.

To estimate the degree to which toehold occlusion affects the magnitude and timing of the negative disturbances, we ran three Three-Step Model simulations of the 8uM uniform buffer being disturbed by 100nM Competitor. In the first simulation, we did not include any occlusion effects (Fig. SI8b). This allowed us to observe how the system should behave ideally. In the second simulation, we included the expected relatively minor occlusion effects of the Initiator+Reporter and Initiator+R_F and Initiator+Sink (Fig. SI8c). We saw that with these three occlusion effects turned on, the setpoint concentration was shifted down, but the disturbance was still able to deplete the concentration of X, after which the buffer restored the concentration of X close to the setpoint concentration. The effect of these types of occlusion should be able to be addressed by tuning the buffer with higher concentrations of source or initiator, or by increasing the forward rate constant to counteract the reduced setpoint, but shouldn't cause other problems. In the third simulation we included the same toehold occlusions as in simulation 2, but also added the expected effect of the

occlusion between initiator and competitor (Fig. SI8d). We observed that the 100nM of competitor was unable to fully deplete the concentration of X, similar to what was observed in experiments (Fig. 5a). We also observed that the recovery to the setpoint concentration was significantly slower than when this form of toehold occlusion was not present.

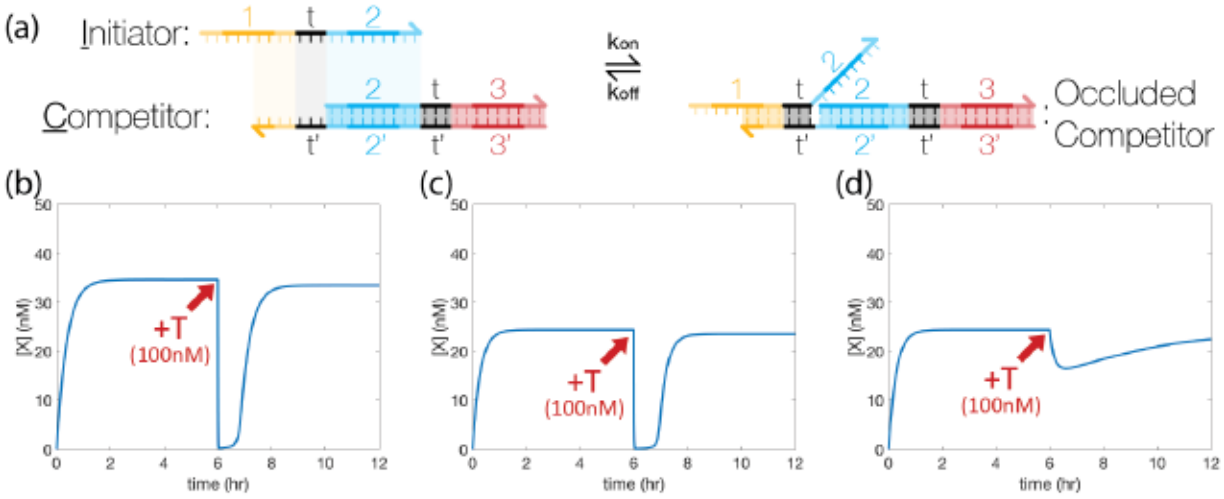

Figure 8.8. The effect of toehold occlusion on buffers with negative disturbances. (a) DNA strand-displacement diagram illustrating the Initiator occluding the Competitor's toehold. (b) Three-step model simulation of a buffer responding to a negative disturbance of [C]=100nM without any occlusion (initial concentrations at [S]=[I]=[N]=8uM, [Reporter]=100nM, [R_Q]=100nM). (c) The same simulation as in b, with occlusion effects included between I+Reporter, I+R_F, and I+N. The equilibrium concentration is reduced. (d) The same simulation as in c, with the occlusion of I+C included. The negative disturbance is damped, as observed experimentally in main text Fig. 5a. This occlusion effectively reduces the reaction rate constant between X and the competitor C, which prevents the disturbance species from reacting quickly to consume X.

The Matlab script used to run our occluded Three-Step Model predictions is printed below:

```
clear all %#ok<CLALL>

close all

clc



runTime=6*3600;%s

fontSize=21;

lineWidth=3;



displacementBP = 15;%bp

kf = 3.5;%uM^-1 s^-1

kb = 400/(displacementBP^2);%s^-1

krFactor = 10^6;

RT=298*1.987*10^-3;

krNEG1bp = kf*krFactor*(2/displacementBP)*exp(3.5/RT);%s^-1

kr0bp = kf*krFactor*(2/displacementBP)*exp(1.9/RT);%s^-1

kr1bp = kf*krFactor*(2/displacementBP)*exp(0.2/RT);%s^-1

kr2wbp = kf*krFactor*(2/displacementBP)*exp(-0.8/RT);%s^-1

kr2bp = kf*krFactor*(2/displacementBP)*exp(-1.7/RT);%s^-1
```

```
kr2NPbp = kf*krFactor*(2/displacementBP)*exp(-2.37481/RT);%s^-1

kr2NPNNbp = kf*krFactor*(2/displacementBP)*exp(-2.50781/RT);%s^-1

kr2sbp = kf*krFactor*(2/displacementBP)*exp(-3.2/RT);%s^-1

kr3bp = kf*krFactor*(2/displacementBP)*exp(-3.0/RT);%s^-1

kr4bp = kf*krFactor*(2/displacementBP)*exp(-4.5/RT);%s^-1

kr5bp = kf*krFactor*(2/displacementBP)*exp(-6.9/RT);%s^-1

kr6bp = kf*krFactor*(2/displacementBP)*exp(-8.3/RT);%s^-1

kr7bp = kf*krFactor*(2/displacementBP)*exp(-9.2/RT);%s^-1



prodOn=1;

sinkOn=1;

disturb=-0.1;

inputOn=0;

occlusionOn=1;



krRelease=kr0bp*15/21;

krRecapture=kr2bp*15/21;

initS=prodOn*8;%uM
```

```
initD=sinkOn*8;%uM

initR=0.1;

initRiQ=initR*0.5;

initInput=inputOn*0.160;


[Sx,id]=getID(1);[Ix,id]=getID(id);[SIx1,id]=getID(id);[SIx2,id]=getID(id);[Dx,id]=getID(id);[X,id]=
getID(id);

[Si,id]=getID(id);[Ii,id]=getID(id);[SIi1,id]=getID(id);[SIi2,id]=getID(id);[Di,id]=getID(id);[in,id]=ge
tID(id);

[Ri,id]=getID(id);[Ri1,id]=getID(id);[Ri2,id]=getID(id);[RiF,id]=getID(id);[RiQ,id]=getID(id);

[Ti,id]=getID(id);[Ti1,id]=getID(id);[Ti2,id]=getID(id);[TiW1,id]=getID(id);[TiW2,id]=getID(id);

[TRiQ1,id]=getID(id);[TRiQ2,id]=getID(id);[TRiQW1,id]=getID(id);[TRiQW2,id]=getID(id);

[IR1,id]=getID(id);[IR2,id]=getID(id);%Initiator:Reporter toehold occlusion (forward)

[IRf1,id]=getID(id);[IRf2,id]=getID(id);%Initiator:RiF toehold occlusion (reverse)

[IT1,id]=getID(id);[IT2,id]=getID(id);%Initiator:Threshold toehold occlusion

[IS1,id]=getID(id);[IS2,id]=getID(id);%Initiator:Sink toehold occlusion


dydt=@(t,y) rxn(y,[Sx,Ix,SIx1,SIx2,Dx,X],21,krRelease,krRecapture)+...%Control Signal X Buffering
```

```
    rxn(y,[Si,Ii,SIi1,SIi2,Di,in],21,krRelease,krRecapture)+...%Input Buffering
    rxn(y,[in,Ri,Ri1,Ri2,RiF,RiQ],19,kr5bp*15/19,kr5bp*15/19)+...%Input Reporting
    rxn(y,[in,Ti,Ti1,Ti2,TiW1,TiW2],21,kr7bp*15/21,kr0bp*15/21)+...%Input Thresholding
    rxn(y,[RiQ,Ti,TRiQ1,TRiQ2,TRiQW1,TRiQW2],19,kr0bp*15/19,kr2bp*15/19)+...%Rq:Thresh
leak reaction
    occlusionOn*rxn(y,[Ii,Ri,IR1,IR2],7,kr5bp*15/7)+...%Initiator:Reporter toehold occlusion
(forward)
    occlusionOn*rxn(y,[Ii,RiF,IRf1,IRf2],2,kr5bp*15/2)+...%Initiator:RiF toehold occlusion (reverse)
    occlusionOn*rxn(y,[Ii,Ti,IT1,IT2],9,kr7bp*15/9)+...%Initiator:Threshold toehold occlusion
    occlusionOn*rxn(y,[Ii,Si,IS1,IS2],2,kr2bp*15/2)...%Initiator:Sink toehold occlusion
    ;
initC=[...
    initS,initS,0,0,initD,initInput,...
    initS,initS,0,0,initD,initInput,...
    initR,0,0,0,initRiQ,...
    0,0,0,0,0,...%Input Thresholding
    0,0,0,0,...%Thresh/Reporter Waste Reaction
    0,0,...%Initiator:Reporter toehold occlusion (forward)
```

```
    0,0,...%Initiator:RiF toehold occlusion (reverse)
    0,0,...%Initiator:Threshold toehold occlusion
    0,0,...%Initiator:Sink toehold occlusion
    ]';

[time,y] = ode15s(dydt,[0,runTime],initC);
if disturb~=0
    if disturb<0
        y(end,Ti)=y(end,Ti)-disturb;
    else
        y(end,in)=y(end,in)+disturb;
    end
    [time2,y2] = ode15s(dydt,[0,runTime],y(end,:));
    time=[time;time2+time(end)];
    y=[y;y2];
end

figure
```

```
plot(time/3600,(y(:,in)+y(:,Ri1)+y(:,Ri2)+y(:,RiF)+y(:,IRf1)+y(:,IRf2))*1000,'LineWidth',lineWidth);

ylabel('[X] (nM)','FontSize',fontSize);

xlabel('time (hr)','FontSize',fontSize);

set(gca, 'FontSize', fontSize);

axis([0,2*runTime/3600,0,50]);
```

8.7 | Relaxation constant remains the same after multiple buffer disturbances.

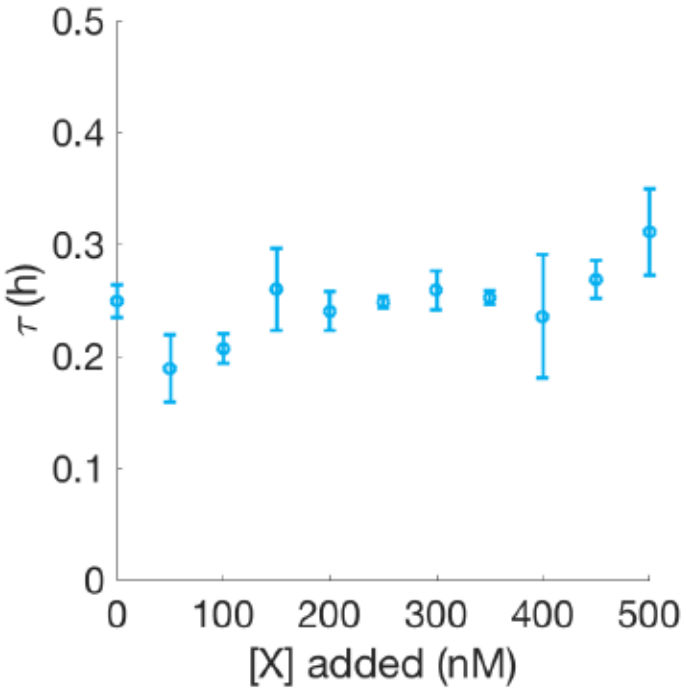

Figure 8.9. Time constants as a function of total disturbance of X added to an 8uM uniform buffer, fit to data from main text Fig. 5a, in which a 50nM disturbance is repeatedly added to the system and then allowed to equilibrate. The time constant to relax to equilibrium does not vary significantly for the range of disturbances tested.

8.8 | Negatively disturbing the 8μM uniform fast buffer

We tested the 8μM uniform fast buffer to negative disturbances by adding pulses of competitor (Fig. SI10). We observed a gradual relaxation back to equilibrium. The slower speed of relaxation in response to the competitor than in response to X is consistent with the hypothesis that toehold occlusion prevents the competitor from reacting instantaneously with X. Effectively, toehold occlusion creates a transient load on the system until all of the competitor is consumed.

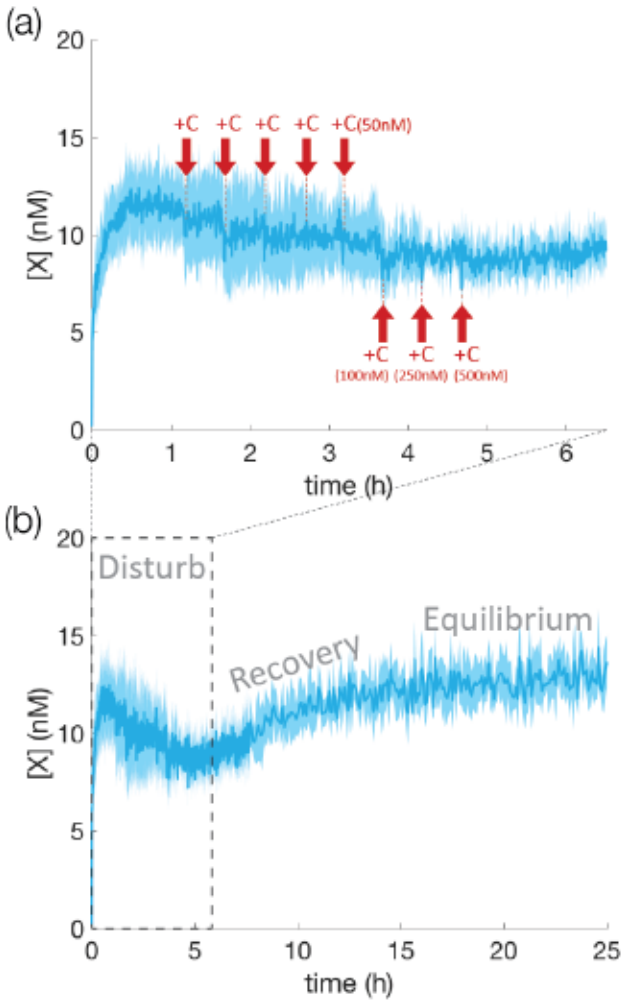

Figure 8.10. Negative disturbances to the 8μM uniform fast buffer, with additions of competitor as noted in the figure. (a) a zoomed in plot showing the rapid response to the disturbances, (b) a zoomed out plot showing the gradual consumption of competitor, after which the system recovers to equilibrium.

References

[SI-R1] Qian, L.; Winfree, E. "Scaling up digital circuit computation with DNA strand displacement cascades." *Science* 2011, *332.6034*, 1196-1201.

[SI-R2] J. N. Zadeh, C. D. Steenberg, J. S. Bois, B. R. Wolfe, M. B. Pierce, A. R. Khan, R. M. Dirks, N. A. Pierce. NUPACK: analysis and design of nucleic acid systems. *J Comput Chem*, 32:170–173, 2011.

[SI-R3] Seelig, G.; Soloveichik, D.; Zhang, D. Y.; Winfree, E. (2006) Enzyme-free nucleic acid logic circuits. *Science*, *314.5805*, 1585-1588.

[SI-R4] Zhang, David Yu, and Erik Winfree. "Control of DNA strand displacement kinetics using toehold exchange." *Journal of the American Chemical Society* 131.47 (2009): 17303-17314.

9 | Powering DNA strand-displacement circuits with a continuous flow reactor

Summary. Living systems require a sustained supply of energy and nutrients to survive. These nutrients are ingested, transformed into low-energy waste products, and excreted. In contrast, synthetic DNA strand-displacement circuits typically run within closed systems provided with a finite initial supply of reactants. Once the reactants are consumed, all net reactions halt and the circuit ceases to function. Here we run DNA strand-displacement circuits in a continuous flow reactor, infusing fresh reactants and withdrawing waste, enabling circuits to dynamically update their outputs in response to changing inputs. Running DNA strand-displacement circuits inside of continuous flow reactors allows circuits to be re-used for multiple rounds of computation, which could enable these circuits to execute more elaborate information processing tasks, including single-rail negation and sequential logic.

9.1 | Introduction

DNA strand displacement (DSD) reactions are a useful mechanism for processing molecular information. They have previously been used to implement Boolean logic gates[1-5], signal amplifiers[6-9], neural networks[10], and other circuits[11-15], as well as for sensing[16] and directed self assembly[17,18]. However, these reactions circuits are typically run inside of a closed reactor, without additional reactants provided after initialization. When the circuit consumes its initial reactant stock and reaches equilibrium, it ceases processing information and can no longer update its output in response to changing input conditions. Providing DSD circuits with a sustained source of reactants could enable them to operate for longer times, extending the lifespan of sensors, increasing the classes of dynamic circuits that can be implemented, and enabling the directed self-assembly of more

elaborate structures.

Flow and diffusion reactors have previously been used to supply reactants to several other types of chemical reactions, including protein synthesis in artificial cells on biochips[19], sustained oscillations in microfluidic flow reactors[20-22], and enzymatic reactions in cascading flow cells[23-26]. In general, these systems operate by allowing material exchange between the reaction chamber(s) and an external flow of reactants. This continuous flow of reactants allows circuits to continue to react with inputs as long as the flow is maintained, analogous to the exchange of matter between living cells and their external environments.

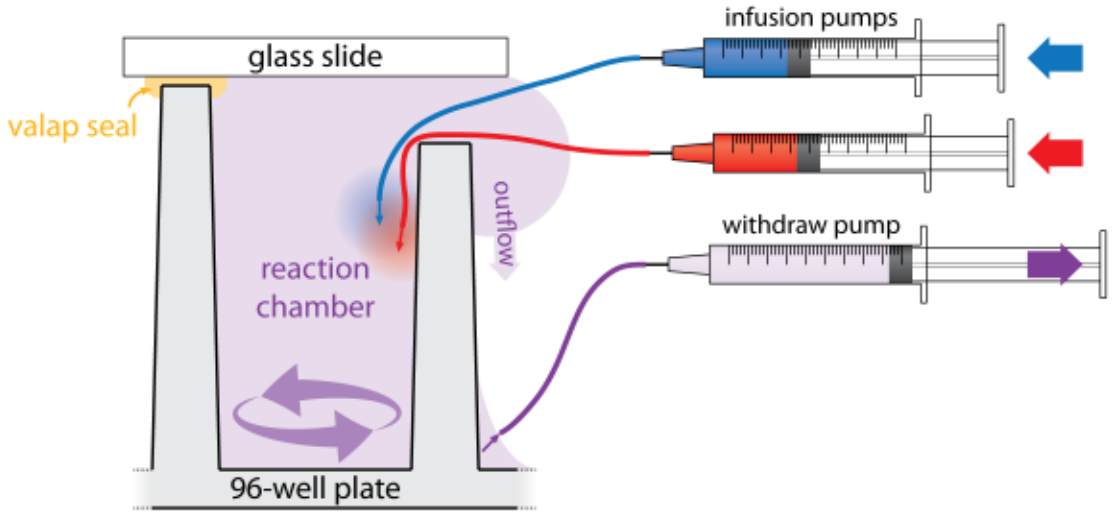

Figure 9.1. Continuous-flow, stirred-tank reactor design. Infusion pumps flow reactants into one well in a 96-well plate. The solution in the well is mixed by a magnetic stir bar. A hole in the side of the chamber allows waste solution to exit the chamber, maintaining a constant reaction volume. Waste is collected from the adjacent well by a withdrawal pump. Fluorescence in the wells is monitored through the glass slide on top using a gel imager.

In this study, we ran DNA strand displacement (DSD) circuits inside a continuous-flow stirred-tank reactor that pumps in reactants while removing waste products (Fig. 1). To do so, we

developed a simple stirred-tank flow reactor design and monitoring protocol that can be implemented without access to specialized equipment or micro-fabrication facilities. This setup can maintain reactions far from equilibrium for as long as the pumps are on, allowing chemical circuits to respond dynamically to changing inputs. In principle, any DSD circuit can run dynamically inside of a flow reactor. We specifically demonstrate how an existing one-time-use DSD Boolean logic gate1 can dynamically respond to changing input concentrations, without modification, within our reactor.

9.2 | Reactor Construction

To construct the reactor used in this study (Fig. 1), we began with a 96-well plate. One well inside the plate was designated as the reaction chamber. A hole of about 2-3mm in diameter was cut into the top of one side of the reaction chamber to allow fluid to flow in and out of the chamber. A stir bar was placed in the reactor to ensure the solution is well-mixed during the reaction. Tubing (ID 0.28mm, Warner Instruments) attached to syringes controlled by syringe pumps (New Era NE-300) was placed through the hole to flow reactants into the chamber. The top of the well is sealed with valap (equal parts Vaseline, lanolin and paraffin wax) and a glass slide to prevent evaporation or outflow over the top of the well. The reactor was designed so that excess liquid flowed out of the cut hole into the adjacent well, where it was collected with a withdraw pump operating at approximately 2x the volumetric inflow rate to prevent fluid buildup. The volume of the reaction chamber was measured at 380uL. For reactions involving multiple reactants we supplied reactants that could react with one another using different infusion lines to prevent reactions from occurring before the species reached the reaction chamber.

The rate of change of the concentration of a molecule X being pumped into the reaction chamber is:

$$\frac{d[X]}{dt} = \frac{\dot{V}_{in}}{V}[X]_{in} - \frac{\dot{V}_{out}}{V}[X] + R \qquad (1)$$

where:

$\dot{V}_{in}$=volumetric flow rate in (units: volume/time)

$\dot{V}_{out}$=volumetric flow rate out (units: volume/time)

V=volume of the reaction chamber

$[X]_{in}$=concentration of X being pumped in

R=net rate X is generated by chemical reactions in the reactor.

When the chamber is full, the escape hole cut in the side of the reaction chamber allows the same volume to escape as is pumped in. Thus in this case the volumetric flow rates are constrained to be equal (i.e. V˙_out=V˙_in). Substituting, we can obtain:

$$\frac{d[X]}{dt} = f_{in} - f_{out}[X] + R \qquad (2)$$

where:

$$f_{in} = concentration\ flux\ in \equiv \frac{\dot{V}_{in}}{V}[X]_{in} \qquad (3)$$

$$f_{out} = flux\ constant\ out \equiv \frac{\dot{V}_{out}}{V} = \frac{\dot{V}_{in}}{V} \qquad (4)$$

Solving Eqn. 2 for $[X]_0$=0 in the absence of any chemical reactions gives that when f_{in} and f_{out} are held constant, [X] approaches a constant value equal to the ratio of the flux in, to the flux constant out, *i.e.*:

$$[X](t) = \frac{f_{in}}{f_{out}} + \left([X]_0 - \frac{f_{in}}{f_{out}}\right) \cdot e^{(-f_{out} \cdot t)} \qquad (5)$$

$$lim_{t\to\infty}[X](t) = \frac{f_{in}}{f_{out}} \qquad (6)$$

The time that it takes for [X] to relax from an initial concentration $[X]_0$ to a given percentage τ of the offset from steady state concentration $\frac{f_{in}}{f_{out}} + \tau\left([X]_0 - \frac{f_{in}}{f_{out}}\right)$ is then:

$$t_{rise,\tau} = \frac{-\ln(\tau)}{f_{out}} \qquad (7)$$

9.3 | Control of reactant concentrations using flow alone

To test that the flow reactor operates as designed, we used a single-infusion syringe to flow in solution containing a strand of DNA labeled, which we termed X, with Cy3 fluorophore (Sequence in SI Table 1) in Tris-acetate-EDTA buffer containing 12.5 mM magnesium acetate (1x TAE/Mg^{2+}), and tracked the resulting fluorescence intensity of the solution over time (Fig. 2a). The infusion line here, and in all subsequent experiments in this paper, also contained 1 μM of an inert, single-stranded DNA, consisting of twenty thiamine nucleotides to prevent the gate and input DNA from sticking to the tubes or the reactor walls1 We tested three different volumetric flow rates (V˙_in=10 μL/min, 20 μL /min and 40 μL /min) with two different infused concentrations of X ([X]in = 50nM and 100nM). We observed the fluorescence (which is assumed to be proportional to [X]) approaching a stable steady state controlled by the ratio f_in/f_out , with rise times controlled by f_out, as predicted by the model. However, when we infused a 50nM solution of a X at a rate of 10 μL /min, we observed a transient delay in the increase in fluorescence. In experiments that followed, we chose concentrations and flow rates that produced good agreement with the model.

We next asked how the concentration of X in the reactor changed over time and how these changes compared with the changes predicted by the model in Eqn. 2 as $[X]_{in}$ is repeatedly switched between high and low concentrations. We first pumped in plain buffer from t=0 to 1hr. We then allowed [X] to equilibrate in the reaction chamber by pumping in a solution containing $[X]_{in}$ = 50 nM at 6 μl/min from t=1 to 6 hr. We then infused a buffer containing no X into the reactor from t=6 to 12 hr, then infused the solution with [X]in = 50 nM into the reactor from t=12 to 17 hr, and finally infused plain buffer into the reactor for the remainder of the experiment (Fig.2b). As predicted, the fluorescence rose when X was pumped in, and decreased dramatically when buffer

alone was pumped in.

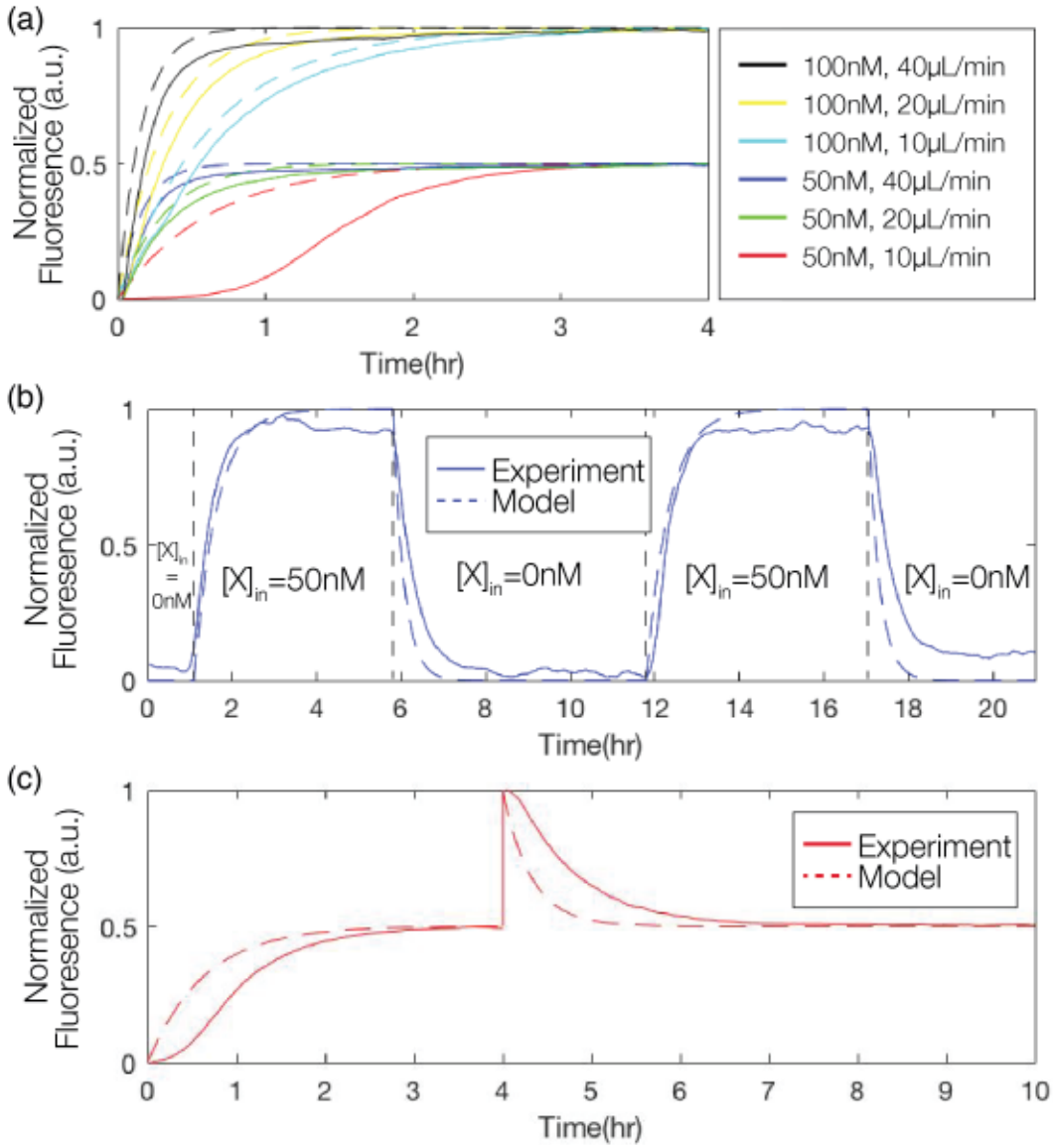

Figure 9.2. Pumped control of the concentration of a fluorescently modified DNA strand X (47 bases long, labeled with Cy3, see SI 1.3 for sequence). (a) Steady state normalized fluorescence for different values of [X]in being infused into the reactor and V ̇_in as shown in legend. (b) Normalized fluorescence as the infused solution is changed over time. After reaching steady state, the concentration of X decays to zero after the infusion pump is switched to plain buffer with no fluorescent DNA, then increases when fluorescent DNA is again pumped into the system. Two on/off cycles are shown, with V ̇_in=6 μL/min and $[X]_{in}$=50 nM during the on cycles. (c) Experiment showing how the normalized fluorescence in the reactor changes

during disturbances. [X] first approaches a steady state while $\dot{V}_{in}$=10 μL/min for a solution in which $[X]_{in}$=50 nM. At t=4 hr, the system is disturbed by pipetting in enough of a concentrated solution of X to increase its concentration by 50nM inside the reaction chamber. After this addition, the steady state fluorescence is eventually restored. Dashed lines show kinetics predicted by Eqn. 2 (see SI3). Data is normalized to the maximum fluorescence intensity in each experiment.

Lastly, we demonstrated the capacity of the flow reactor to resist transient disturbances to the concentration of X. We set up an experiment with $\dot{V}_{in}$=6 μL/min and $[X]_{in}$=50 nM. After allowing the system to reach steady state, we manually pipetted in a 50 nM disturbance to the reaction chamber. We observed that the fluorescence jumped when the disturbance was added, and then returned to the same concentration as before the disturbance (Fig. 2c).

9.4 Dynamic Boolean strand-displacement logic in a flow reactor

Next we used the flow reactor to perform dynamic molecular logic by introducing a Boolean logic AND circuit into our continuous flow reactor. The molecules of the circuit were designed to fluoresce only when the two input strands are both present (Fig. 3a-b)[1]. We can thus characterize the true/false result of the logic at a given time by measuring the high/low value of fluorescence.

First we tested that the Boolean logic gate gave the correct output response to all four possible combinations of its two binary inputs (Fig. 3c). One problem that can occur if the volumetric flow rates are set too high, relative to the reaction rate constants, is that reactant can enter and leave the reaction chamber quickly enough that there is insufficient time for it to react while in the chamber. To avoid this issue, we imposed bounds on the maximum pumping rate (Eqn. 9, see SI4 for derivation).

$$\dot{V}_{in} \leq \alpha \cdot V \cdot k_{logic}[ON], \qquad (9)$$

where:

$\dot{V}_{in}$=volumetric flow rate in

k_{logic}= reaction rate constant for the rate limiting step in the logic reactions

$\alpha = f_{in}/k_{logic} \ll 1$

(we found empirically that values of α=0.01 worked well)

V=volume of the reaction chamber

[ON]=maximum output concentration for the logic circuit.

We separated the inputs and the logic gate molecules into two separate infusion lines with equal volumetric flow rates (5 μl/min in each line), to prevent them from reacting until they reached the reaction chamber. To compensate for the resulting dilution of these reactants once they were mixed in the reactor, the concentration of each reactant in its respective infusion line was $N \cdot [X]_{in}$, where N=2 was the total number of infusion lines, and [X]in was the desired effective infusion concentration.

In these experiments, one infusion line contained 400nM AND gate and remained on continuously for the duration of the experiment. The other line contained the two input molecules at concentrations that denoted either ON, when the infusion line contained 300 nM of an input, or OFF if the infusion line contained none of that input.

To test the dynamic operation of the logic circuit (Fig. 3c), we first initialized the system with both inputs in the OFF state, allowing it to run for 2 hours to ensure it was at steady state. At t=2 hr, we turned Input 1 ON by increasing its concentration in the input infusion line, while keeping Input 2 OFF. We observed a small increase in fluorescence, but the steady state fluorescence was relatively low, indicating an OFF output. At t=6 hr, we again changed the infusion line to contain both inputs in the ON state. The fluorescence jumped to a high intensity, indicating a correct ON output for the Boolean AND gate. At t=10 hr, we turned Input 1 OFF, while keeping Input 2 ON, and observed a drop in fluorescence, indicating that the circuit had switched back to its correct

output state OFF for these inputs. Finally, at t=14 hr, we set both Inputs to OFF, and observed that the fluorescence remained in a low OFF output state. Thus the circuit responded correctly to all possible combinations of inputs when they were provided dynamically in the flow reactor.

Finally, we tested the same circuit with a more complex set of randomized inputs (Fig. 3d). We supplied Input 1 and Input 2 in the following combinations (t=0-2hr: I1=OFF, I2=OFF), (t=2-6hr: I1=ON, I2=ON), (t=6-10hr: I1=OFF, I2=ON) , (t=10-14hr: I1=OFF, I2=OFF), (t=14-26hr: I1=ON, I2=ON) , (t=26-30hr: I1=ON, I2=OFF) , (t=30-34hr: I1=ON, I2=ON) , (t=34-38hr: I1=OFF, I2=ON) , (t=38-42hr: I1=OFF, I2=OFF). The output fluorescence dynamically tracked the changing inputs to report the correct Boolean AND output state as the inputs changed over time. In addition to demonstrating that the flow reactor enables the logic gate to operate for many cycles, this experiment also showed how the ON state can be sustained for multiple cycles (t=14 hr to 26 hr).

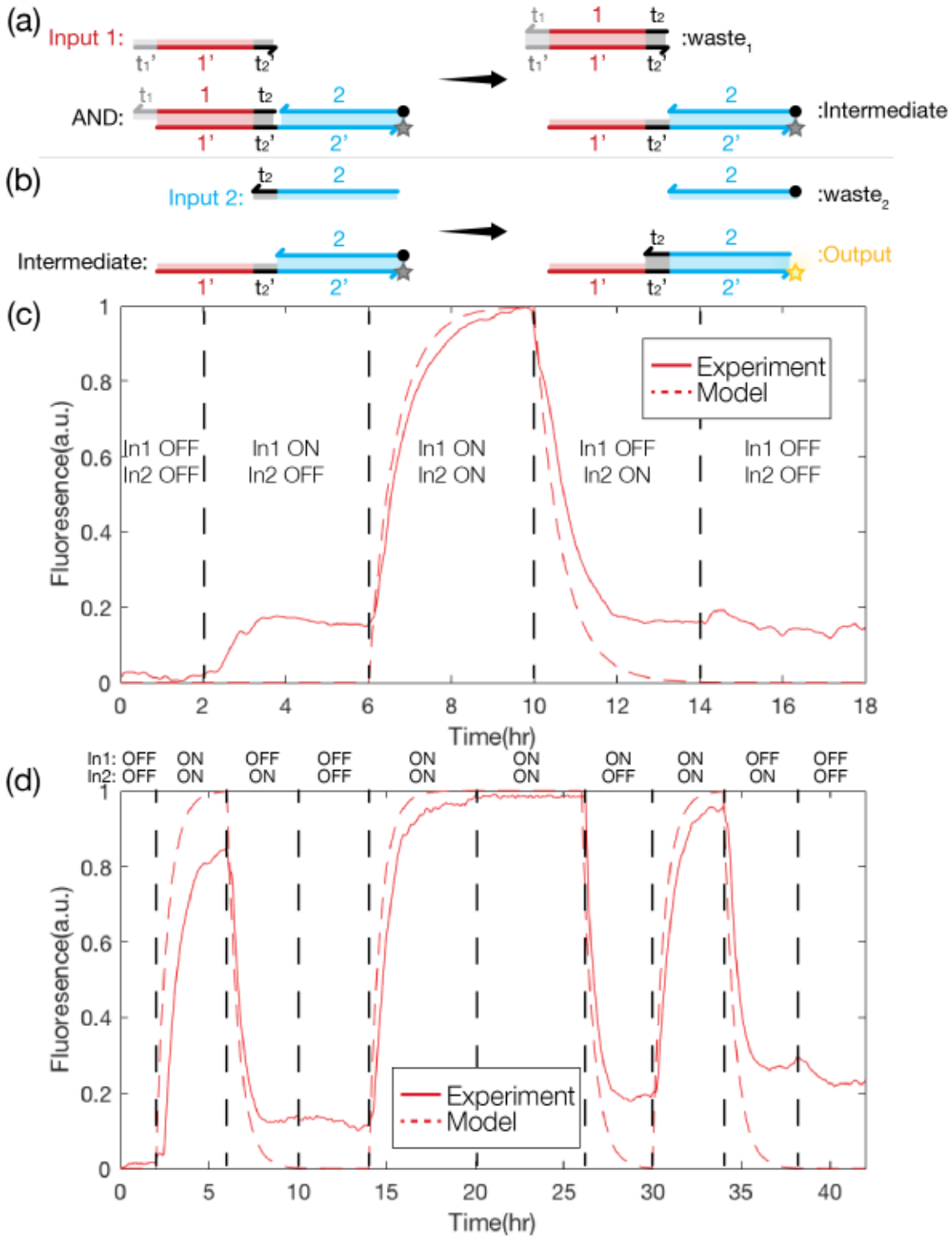

Figure 9.3. Operation of a DNA logic gate in a flow reactor. (a-b) DNA strand-displacement diagram for the Boolean AND gate[1]. DNA strands are represented by lines, and contain short toehold domains (gray, black), and long branch-migration domains (red, cyan). Complementary domains are shown in the same

color. The 3' end of each strand is indicated by an arrowhead. Logical evaluation occurs in two reaction steps when both inputs are present. In the first step, (a), Input 1 binds to the gray toehold on the AND gate, displacing the red domain and exposing a second toehold (black) in the gate. In the second step, (b), Input 2 binds to this newly exposed toehold, displacing the output payload. The output is only released (and thus the fluorescence only changes) when both inputs are present. (c) Experiments showing the AND gate responding to inflow solutions containing all possible ON/OFF combinations of its two inputs (listed in figure) with a volumetric flow rate of V_{in}=5 μL/min. In the ON state, inputs were infused at a concentration of $[X]_{in}$=150 nM, while in the OFF state $[X]_{in}$=0 nM . The dashed line shows the kinetics predicted by Eqn. 2 (see SI 3). (d) Experiments showing the circuit operating for ten different inputs provided in series (separated by dashed vertical lines). The combination of inputs provided during each interval are specified at the top of the plot. The dashed red line shows the kinetics predicted by Eqn. 2 (see SI 3).

Conclusions

In this study, we demonstrated that DNA strand displacement reactions can respond dynamically to changing input conditions within a continuous flow reactor. We also developed a protocol for building a relatively simple, easy to manufacture, and affordable flow reactor. Future studies could explore the use of microfluidics to create reactors with smaller volumes, which would reduce the total cost of reagents and could improve robustness.

Although flow reactors are a convenient means to sustain dynamic reactions, they have several fundamental limitations that closed reactors do not have. First, pumped reactions are not closed systems, in the sense that they depend on flow from outside of the reaction chamber. Second, by definition, reactions in well-stirred reactors cannot generate stable spatial patterns, so they cannot participate in a broad class of interesting spatially heterogeneous phenomena. Third, since the withdraw pump continuously extracts liquid from the well mixed reactor, anything that we wish to keep inside of the reactor without infusion must be anchored inside of the reactor, or filtered to

prevent it from exiting into the withdrawing line. Analogous dynamics to Eqn. 2 might be accomplished chemically instead of mechanically, allowing restoration in a closed solution for a finite number of cycles[27,28], at the cost of increased chemical complexity.

Acknowledgements

The authors thank Qi Huang, Yi Li, and Samuel Schaffter for insightful conversations. This work was supported by NSF-SHF-1527377, DE-SC0010595 for some equipment and reagents, NSF-CCF-1161941 and a grant to the Turing Centenary Project by the John Templeton Foundation to R.S.

References

[1] G. Seelig, D. Soloveichik, D. Y. Zhang and E. Winfree, Science, 2006, 314, 1585–1588.
[2] K. Takahashi, S. Yaegashi, A. Kameda and M. Hagiya, DNA Computing Lecture Notes in Computer Science, 2006, 347–358.
[3] C. Zhang, J. Yang and J. Xu, Langmuir, 2010, 26, 1416–1419.
[4] L. Qian and E. Winfree, Science, 2011, 332, 1196–1201.
[5] A. J. Genot, J. Bath and A. J. Turberfield, Journal of the American Chemical Society, 2011, 133, 20080–20083.
[6] D. Y. Zhang and G. Seelig, Lecture Notes in Computer Science DNA Computing and Molecular Programming, 2011, 176–186.
[7] D. Y. Zhang, A. J. Turberfield, B. Yurke and E. Winfree, Science, 2007, 318, 1121–1125.
[8] X. Chen, N. Briggs, J. R. Mclain and A. D. Ellington, Proceedings of the National Academy of Sciences, 2013, 110, 5386–5391.
[9] G. Seelig, B. Yurke and E. Winfree, Journal of the American Chemical Society, 2006, 128, 12211–12220.
[10] L. Qian, E. Winfree and J. Bruck, Nature, 2011, 475, 368–372.
[11] Y.-J. Chen, S. D. Rao and G. Seelig, Journal of Visualized Experiments, 2015.
[12] D. Soloveichik, G. Seelig and E. Winfree, Proceedings of the National Academy of Sciences, 2010, 107, 5393–5398.
[13] Y.-J. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik and G. Seelig, Nature Nanotechnology, 2013, 8, 755–762.
[14] J. Fern, D. Scalise, A. Cangialosi, D. Howie, L. Potters and R. Schulman, ACS Synthetic Biology, 2016, 6, 190–193.
[15] N. Srinivas et al., Science, 2017
[16] J. L. He, Z. S. Wu, H. Zhou, H. Q. Wang, J. H. Jiang, G. L. Shen & R. Q. Yu, Analytical chemistry, 2010, 82(4), 1358-1364.
[17] D. Y. Zhang, R. F. Hariadi, H. M. Choi, & E. Winfree, Nature communications, 2013, 4, 1965.
[18] W. B. Rogers & V. N. Manoharan, Science, 2015, 347(6222), 639-642.
[19] E. Karzbrun, A. M. Tayar, V. Noireaux & R. H. Bar-Ziv Science, 2014 345(6198), 829-832.
[20] H. Niederholtmeyer, V. Stepanova & S. J. Maerkl, Proceedings of the National Academy of Sciences, 2013 110(40), 15985-15990.
[21] S. N. Semenov, et al. Nature chemistry, 2015 7(2), 160.
[22] H. W. van Roekel et al. Chemical Society Reviews, 2015 44(21), 7465-7483.
[23] F. Moseley et al. Analyst, 2014 139(8), 1839-1842.
[24] B. E. Fratto & E. Katz, ChemPhysChem, 2015, 16(7), 1405-1415.
[25] B. E. Fratto & E. Katz, ChemPhysChem, 2016 17(7), 1046-1053.
[26] B. E. Fratto & E. Katz, Advances in Unconventional Computing, 2017, 29-59.

[27] D. Scalise & R. Schulman, Technology, 2014, 02, 55–66.
[28] D. Scalise & R. Schulman. International Workshop on DNA-Based Computers. Springer, Cham, 2014.

10 | Supplemental Information: Powering DNA circuits with continuous flow reactors

10.1 | Materials and methods

10.1.1 | Circuit Preparation

All DNA oligonucleotides were purchased from Integrated DNA Technologies (IDT). All unmodified oligonucleotide strands were ordered unpurified (standard desalting); all strands with fluorophore/quencher modifications were ordered purified by high-performance liquid chromatography (HPLC).

After being shipped as powder, DNA oligonucleotides were suspended in Milli-Q water at a target concentration of 1000μM, based on the yield provided by IDT, and stored at 4°C. To obtain concentrations more accurately, we used a BioPhotometer (Eppendorf) to measure the absorbance of each strand at 260nm (*OD260*). Typically, we take the average of three measurements where a 5 μl sample was diluted with 195 μl Milli-Q water. Using the extinction coefficient EXT (Unit: L/(mole-cm)) provided by IDT, the concentration (μM) of each single strand is calculated with the formula:

$$Concentration = DF \times \frac{OD260}{EXT} \times 10^6 \qquad \text{(SI1)}$$

Where the dilution factor calculated as:

$$Dilution\ factor(DF) = \frac{Final\ Volume}{Solute\ Volume} \qquad \text{(SI2)}$$

Multi-stranded DNA complexes were then prepared at 100 μM suspended in Tris-acetate-EDTA

buffer containing 12.5mM magnesium acetate (1x TAE/Mg^{2+}), with a 1.2x excess of the top strands (*i.e.* 120 μM) relative to the bottom strand to ensure all bottom strands were bound. We used a slow anneal where the reaction mixture is heated up to 90°C and then cooled down to room temperature (20°C) slowly (1°C/min) with a PCR machine (Mastercycler eppendorf). Each sample was annealed in a separate tube.

10.1.2 | Gate Purification

We used polyacrylamide gel electrophoresis (PAGE) to remove excess single strands and poorly-formed double-stranded complexes from the complex solution before use in experiments. 15% non-denaturing PAGE gels with 1.0mm thickness and a single large well were made by mixing 13mL pre-mix with 78uL 10%APS (Sigma Aldrich) and 7.8uL TEMED (Sigma Aldrich), where pre-mix was made by adding 13.0 mL 10X TAE/Mg^{2+} and 48.75mL 40% Bis-Acrylamide (Bio-Rad) to 68.25mL MilliQ-water.

40 µL 6X Gel Loading Dye blue (New England Biolabs Inc.) was added to each 100 µL of annealed DNA sample before loading the sample to cassettes (Life Technologies). We usually loaded 280uL sample (200uL of annealed DNA with 80uL of loading dye) per gel and ran at 150V for 3 hours at 4 C°. Typically we refreshed the 1XTAE Mg^{2+} buffer after 1.5 hours. After electrophoresis, the band within the gel containing the desired complex was cut out using UV shadowing to visualize the bands. To do so, the gels were placed on top of a bench top transilluminator (3UV™ Transilluminator 254nm), with a Kimwipe placed on top of the gel to act as a screen for visualizing the UV shadow. The target band appeared as a dark line and was cut from the gel with a razor blade. The removed gel slice was then cut into small pieces and soaked in a 1.7mL test tube with 400uL 1X TAE Mg^{2+} buffer for 12 to 24 hours at room temperature to allow most of the DNA complexes to diffuse out of the gel pieces into the surrounding solution. We then pipetted the solution into a fresh

test tube, attempting to leave the gel chunks behind, and centrifuged the solution in the new test tube at 10,000 rcf for 10 minutes to concentrate any remaining gel pieces at the bottom of the tube (Centrifuge 5415D Eppendorf). The majority of the supernatant was then pipetted off of the top of the tube into another fresh test tube, leaving behind approximately 50uL at the bottom that contained most of the polyacrylamide residue.

For purified double-stranded complexes, we used Eqn. (SI1) to calculate the final concentration, with the extinction coefficient calculated as follows[SI1]:

$$EXT = EXT_{top\ strand} + EXT_{bottom\ strand} - 3200 \cdot N_{AT} - 2000 \cdot N_{GC} \qquad \text{(SI3)}$$

where N_{AT} and N_{GC} are the numbers of AT base pairs and GC base pairs in the complex, respectively.

10.1.3 Sequences

Table S.1 DNA Strands

X=/5Cy3/TCTACGGAAATGTGGCAGAATCAATCATAAGACACCAGTCGGTTTTT/3AmMO/
Input 1 = TATGGTTGTTTATGTGTTCCCTGATGCCTTAA
Input 2 = ATGTTAGTTTCACGAAGACAATGATTAAGGC
AND gate cover strand = TAAGGCATCAGGGAACACATAAACAACCATA
AND gate bottom strand=TGTTTATGTGTTCCCTGATGCCTTAATCATTGTCTTCGTGAAACTAACAT/3Cy3sP/
AND gate top quencher strand=/5IABkFQ/ATGTTAGTTTCACGAAGACAATGAT

The codes /5Cy3/, /3Cy3sP/, /5IABkFQ/ and /3AmMO/ used in Table S.1 represent the IDT end modifications for a 5' Cy3 fluorophore, a 3' Cy3 fluorophore, an Iowa Black quencher, and an amino modifier, respectively.

10.2 | Data Processing

Data was recorded as images (Fig.1) from gel imager (Syngene GBox EF2) and converted to concentrations through MATLAB. Figure S1 shows an example image that was used for quantification. The lower left corner well in the image is the reaction well. We selected a square area inside this well to measure the fluorescence intensity in the well. To determine how fluorescence intensity corresponded to fluorescence strand or complex concentration, then we set up four calibration wells on top of reaction well, which is same solution volume with reaction well, but contains 300nM reporter and 150nM Input1 with 0nM, 50nM, 100nM, 150nM input2 from left to right, respectively.

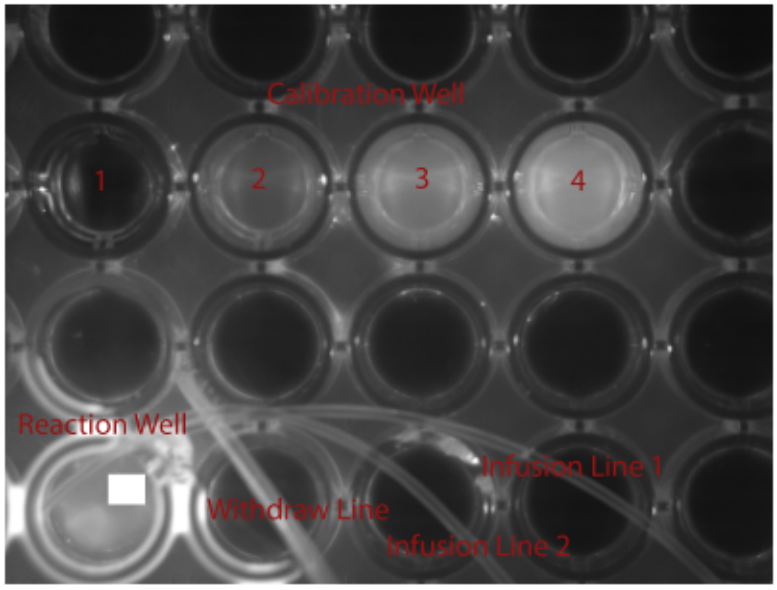

Figure 10.1. Photograph of the reaction well and surrounding wells. The white square over the reaction well indicates the area of integration from which data was collected, with the escape hole to the top right of

the reaction well. The blurry spot in the reaction well, below the area of integration, is the magnetic stir bar.

The concentration of reacted Gates is calculated as:

$$[A] = \frac{R_{final} - R_{initial}}{B} \qquad \text{(SI4)}$$

$$B = \frac{C_2 - C_1}{50} + \frac{C_3 - C_1}{100} + \frac{C_4 - C_1}{150} \qquad \text{(SI5)}$$

where [A] is the concentration of reacted gates, $R_{current}$ and $R_{initial}$ are the current and initial fluorescence intensity of the reaction well. B is the coefficient to convert light intensity into molar concentration. C_1, C_2, C_3, C_4 are the fluorescence intensities of solutions where Input 1= 0nM,50nM,100nM and 150nM respectively.

10.3 | Model

The dashed model lines in Fig. 2a of the main text are the solution to Eqn. 2 (reproduced as Eqn. SI6 below).

$$\frac{d[X]}{dt} = f_{in} - f_{out}[X] + R \qquad \text{(SI6)}$$

where:

$$f_{in} = concentration\ flux\ in \equiv \frac{\dot{V}_{in}}{V}[X]_{in} \qquad \text{(SI7)}$$

$$f_{out} = flux\ const\ \ nt\ out \equiv \frac{\dot{V}_{out}}{V} = \frac{\dot{V}_{in}}{V} \qquad \text{(SI8)}$$

The volumetric flow rate ($\dot{V}_{in}$) and infusion concentrations ($[X]_{in}$) for those experiments are listed

in the legend for Fig. 2a, with the volume of the reaction chamber set to V=380uL, and the reaction term set to zero (*i.e.* R=0). For Fig. 2b, the same model is used, alternating between $[X]_{in}$=0nM and 50nM for the different times labeled in the figure. For Fig. 2c, the two segments before and after the disturbance are modeled separately, with $[X]_0$=0 for the first pre-disturbance segment, and $[X]_0$=100nM for the second segment.

For the Boolean AND gate experiments in Fig. 3, the dashed model lines are the solution to a series of coupled ODE's of the same form as Eqn. 2, one for each chemical species, as follows:

$$\frac{d[Input_1]}{dt} = \frac{\dot{V}_{in}}{V} \cdot [Input_1]_{in} - \frac{\dot{V}_{out}}{V} \cdot [Input_1] - k_{logic} \cdot [Input_1][AND_1] \quad \text{(SI9)}$$

$$\frac{d[Input_2]}{dt} = \frac{\dot{V}_{in}}{V} \cdot [Input_2]_{in} - \frac{\dot{V}_{out}}{V} \cdot [Input_2] - k_{logic} \cdot [Input_2][AND_2] \quad \text{(SI10)}$$

$$\frac{d[AND_1]}{dt} = \frac{\dot{V}_{in}}{V} \cdot [AND_1]_{in} - \frac{\dot{V}_{out}}{V} \cdot [AND_1] - k_{logic} \cdot [Input_1][AND_1] \quad \text{(SI11)}$$

$$\frac{d[AND_2]}{dt} = -\frac{\dot{V}_{out}}{V} \cdot [AND_2] + k_{logic} \cdot ([Input_1][AND_1] - [Input_2][AND_2]) \quad \text{(SI12)}$$

$$\frac{d[Output]}{dt} = -\frac{\dot{V}_{out}}{V} \cdot [Output] + k_{logic} \cdot [Input_2][AND_2] \quad \text{(SI13)}$$

where:

$$V = 380\mu L$$

$$\dot{V}_{in} = 300\,\mu L/hr$$

$$\dot{V}_{out} = \dot{V}_{in}$$

$$k_{logic} = 7.2\,nM/hr$$

$$[Input_1]_{in} = 150nM$$

$$[Input_2]_{in} = 150nM$$

$$[AND_1]_{in} = 200nM$$

For the segments labeled "OFF" in Fig. 3, the corresponding input infusion concentrations

$[Input_1]_{in}$ and $[Input_2]_{in}$ are switched to 0nM.

References

[SI1] V. Bloomfield & D. M. Crothers, D. M. *Nucleic acids: structures, properties and functions*, 2000 (No. 574.192 B52).

11 | Stable DNA-based Reaction-Diffusion Patterns

Summary. We demonstrate reaction-diffusion systems that generate stable patterns of DNA oligonucleotide concentrations within agarose gels, including linear and "hill" (i.e. increasing then decreasing) shapes in one and two dimensions. The reaction networks that produce these patterns are driven by enzyme-free DNA strand-displacement reactions, in which reactant DNA complexes continuously release and recapture target strands of DNA in the gel; a balance of these reactions produces stable patterns. The reactant complexes are maintained at high concentrations by liquid reservoirs along the gel boundary. We monitor our patterns using time-lapse fluorescence microscopy and show that the shape of our patterns can be easily tuned by manipulating the boundary reservoirs. Finally, we show that two overlapping, stable gradients can be generated by designing two sets of non-interacting release and recapture reactions with DNA strand-displacement systems. This paper represents a step toward the generation of scalable, complex reaction-diffusion patterns for programming the spatiotemporal behavior of synthetic materials.

11.1 | Introduction

Gradients are ubiquitous drivers of spatially differentiated behavior and communication in biological systems. For example, stripes of mRNA concentrations generated by reaction-diffusion (RD) processes in the embryo of the fruit fly Drosophila act as chemical blueprints to direct the growth of the embryo[1,2]. A variety of spatial concentration patterns also arise during intercellular signaling processes[3,4]. Synthetic patterns have been generated in vitro to study and control chemotaxis[5], angiogenesis[6], stem cell proliferation and differentiation[7], axon growth[8], cell culture and cell behavior in hydrogels[9,10], and protein expression[11]. Chemical gradients have also been used to control reactivity, direct mechanical actuation, pattern synthetic materials and orchestrate self-

regeneration[12-14].

Chemical gradients are often produced using lithographic or light-driven patterning methods that encode variations of density of a molecule along a surface or within a 3-dimensional material[15-17], or by diffusion and flow across membranes in microfluidic devices[18-21]. While these top-down processes can generate patterns of molecules of high complexity[16, 17], the resulting patterns cannot easily evolve or regenerate over time as materials are consumed or diffuse away.

Synthetic reaction-diffusion systems can also produce spatiotemporal chemical patterns from the bottom up, using inorganic systems such as the Belousov-Zhabotinsky reaction[22,23], or enzymatic networks[24-29] both with transcriptional circuits[30] and with the Polymerase, Exonuclease, Nickase (PEN) toolbox[31,32], including gradients, traveling waves and spatial patterns of spots or stripes[33,34]. However, the kinetics of many enzyme-based RD systems are sensitive to temperature variations of a few degrees C and to buffer conditions, limiting when they can be applied. Further, it is generally difficult to scale the number of components that can be combined in the same solution in many inorganic reaction-diffusion processes, limiting the complexity of patterns that may be formed with the components[35].

An alternative bottom-up approach for generating chemical patterns is to use enzyme-free DNA strand-displacement reactions[36], which can be used to program large, coupled chemical reaction networks[37-39]. Because they are controlled by forward rates of reaction between DNA species, strand-displacement reactions have relatively consistent rate constants[36,37,39-44] across a temperature range of tens of degrees C. Strand-displacement reactions have also been demonstrated in a variety of buffers, cell media[45], and within living cells[46]. While DNA reaction-diffusion systems have been built using strand-displacement reactions[47,48], the patterns that form tend to be temporally unstable, as diffusion eventually drives soluble output patterns into homogeneity.

We have previously suggested a design for strand-displacement reaction networks in which the molecules that form a pattern are continuously released and recaptured faster than diffusion can mix them together, generating stable patterns[49,50]. This process enables chemical patterns formed by reaction-diffusion processes to regenerate when perturbed, and could serve as a building block for the modular design of reaction-diffusion processes that form more complex patterns such as a stick figure.

In this paper, we use a continuous release-and-recapture motif to generate stable patterns of soluble DNA molecules within a hydrogel substrate using enzyme-free DNA strand displacement reactions. These patterns are maintained by reactant molecules diffusing in from liquid reservoirs along the substrate boundary where these molecules are present at high concentrations. We show that our system can produce millimeter-scale heterogeneous patterns in one and two dimensions, and that these patterns can be made to either grow continuously or to remain stable over time. In principle, stable patterns should remain stable as long as the reservoirs have a high concentration of reactant molecules. To underscore this point, we show that these patterns remain stable for over 30 hours. We also create multiplexed patterns involving multiple species of DNA with orthogonal nucleotide sequences.

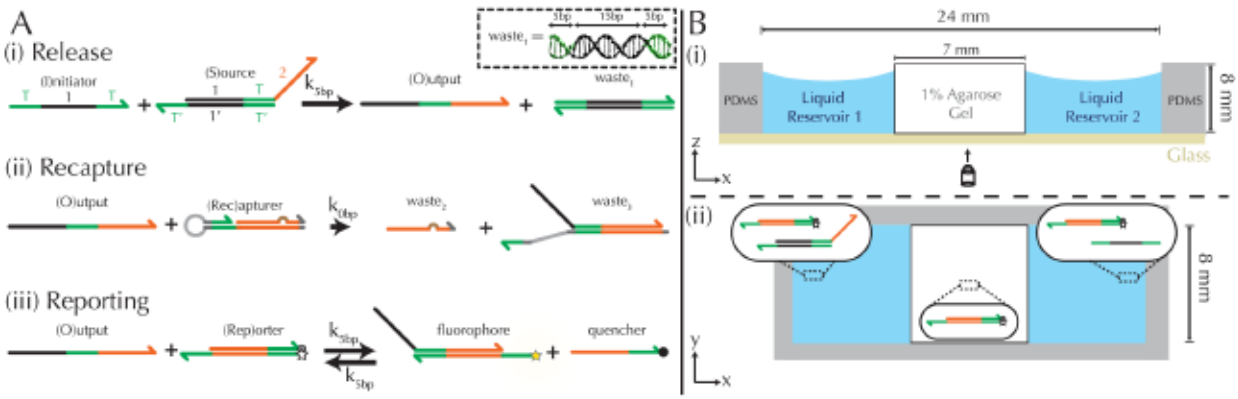

Figure 11.1. Schematic of DNA strand displacement reactions and the setup of reaction diffusion systems. (A) Three strand displacement reactions generate and monitor gradients. (i) Release: The Output

strand is initially bound within a Source complex with its toehold domain (green) sequestered in an inactive, double-stranded state. The Output is released by an Initiator strand via a relatively fast 5 base-pair (bp) toehold mediated reaction. (ii) Recapture: The Output strand is recaptured by a Recapturer complex in a relatively slow reaction. (iii) Reporting: The concentration of the Output is "read" by reversibly reacting with a reporter complex whose strands have an attached fluorophore and quencher. The inset shows the structure of a waste1 complex. Green domains indicate toeholds (5bp) and the black domain indicates the "1" domain (15bp). Complementary sequences are denoted by an apostrophe (e.g., sequence 1 is fully complementary to 1') and share the same color. Three-prime ends of the DNA strands are labeled with an arrow. The brown bump in the Recapturer complex indicates a single base mismatch. (B) (i) Side-view schematic of the RD cell (x-z plane), whose exterior is formed from PDMS cast around a negative mold. The RD system inside consists of a 1% agarose hydrogel between two liquid reservoirs, each containing a solution of DNA species. An optical microscope images the cell through a glass coverslip bound to the PDMS (Methods). (ii) Top-view schematic of an RD cell (x-y plane). Inset diagrams depict initial conditions for an experiment in which a growing gradient forms: Liquid Reservoir 1 contains of Source and Reporter species, Liquid Reservoir 2 contains Initiator and Reporter species, and the hydrogel contains Reporter species.

11.2 | Results and Discussion

11.2.1 | System design and mechanism. Our goal was to develop a reaction-diffusion (RD) system that would lead to the formation of a spatiotemporally stable (i.e. unchanging) gradient using DNA-based strand displacement systems. RD processes can be described by a set of partial differential equations of the form:

$$\frac{\partial C_j}{\partial t} = R_j + D_j \nabla^2 C_j, \qquad (1)$$

where C_j, D_j and R_j are the concentration, diffusion coefficient and the total rate of the reactions involving species j, respectively, with one such equation for each of the species in the system.

Our model for the generation of stable gradients is a coupled set of reactions that (1) release the

species forming the gradient and (2) recapture that species. These reactions together induce the dynamics

$$R_O = k_{rel} - k_{cap}[O], \qquad (2)$$

where k_{rel} is the rate of release of the output species O, and k_{cap} is the reaction rate constant of recapture. At steady state in a well-mixed solution, [O] is stable at the concentration k_{rel}/k_{cap}.

To emulate this abstract release and recapture process, we devised a set of DNA strand-displacement reactions that release and recapture an output species O, respectively:

$$S + I \xrightarrow{k_1} O + W_1, \qquad (3)$$

$$Rec + O \xrightarrow{k_2} W_2 + W_3, \qquad (4)$$

where S, I and Rec are reactant species and W_1, W_2 and W_3 are waste products. We call S, I, Rec, and O the Source, Initiator, Recapturer and Output, respectively. When the concentrations of S, I and Rec are held constant, the change in [O] over time follows Equation 2 where $k_{rel}=k_1[S][I]$ and $k_{rec}=k_2[Rec]$.

These opposing release and recapture reactions balance each other to form a stable point that can be used to generate stable concentration gradients. By controlling the spatial variation of the reactant concentrations and choosing the rate constants for the release and recapture reactions, a variety of gradient profiles may be created. For example, if S and I meet and release O at a single location, and Rec is present throughout the substrate, a stable gradient of O will form around the release location, which will remain stable as long as S, I and Rec are supplied to the system.

11.2.2 | DNA strand displacement release and recapture reactions.

To implement the release reaction described in Eqn. 3, we initially sequester the Output molecule in an inert form within the Source complex. O is rapidly released from S when it is displaced by an Initiator

molecule. The Initiator binds to the Source complex via a 5 base pair toehold, which has a standard rate constant[40] $k_{5bp}=5\times10^4$ $(M\ s)^{-1}$. The recapture reaction in Eqn. 4 occurs when a Recapturer complex binds to O, sequestering its toehold and rendering it unable to react further (Fig. 1Aii). We designed the recapture reaction to occur much more slowly, via a strand displacement process that is initiated at a nick in the Rec dsDNA backbone, which we model with a rate constant of $k_2=k_{nick}=500$ $(M\ s)^{-1}$. Together, release and recapture processes cycle molecules of O between their released (free) and recaptured (inert) state, and can form a spatial gradient of the output strand faster than diffusion mixes the components into homogeneity. To experimentally monitor the concentration of released O, we also designed a 5bp reversible reporting reaction (Eqn. 5). In this reaction, a Reporter complex, Rep, which consists of a fluorescent strand f, and a quencher strand q, rapidly and reversibly reacts with Output O through 5bp toeholds (Fig. 1Aiii).

$$\text{Reporting:} \qquad Rep + O \underset{k_{r,off}}{\overset{k_{r,on}}{\rightleftharpoons}} f + q, \qquad (5)$$

$$\text{where: } k_{r,on} = k_{r,off} = k_{\mathbf{5bp}}.$$

We next designed the DNA sequences, for the Source, Output, Recapturer and Reporter by starting with a set of sequences used in other strand displacement circuits[37]. We added clamps to these sequences to reduce the rates of unintended "leak" reactions between the complexes[51]. Such leak reactions are likely a result of fraying ends of complexes[52,53], synthesis errors[54] or imperfections in complex purification, which limits the ability to design reactions to exact specifications. We include rates of these leak reactions, based on measurements and estimates[55], in our models (see section 12.1).

11.2.3 | Construction of a reaction-diffusion reaction cell and reaction monitoring.

We built a reaction cell composed of three portions of approximately equal volume. The outer two

portions are liquid reservoirs, with a 1% agarose hydrogel in the middle (Fig. 1B and section 12.5). This set up allows us to maintain constant concentration boundary conditions at either end of the hydrogel substrate for the reactant molecules (S, I, Rec, and Rep) by manually exchanging the liquid within the reservoirs. The continuous diffusion of fresh reactants from the reservoir into the substrate, and the diffusion of waste products out of the substrate, drives the release and recapture reactions that hold the pattern of O stable in spite of diffusion. Because the agarose substrate resists non-diffusive flows, we can exchange the reaction buffer in these reservoirs repeatedly without perturbing the pattern. We measure the intensity of fluorescence using time-lapse microscopy (see Methods) and convert intensity to concentration of Output using a calibration curve (section 12.2, Methods).

To design gradient patterns we built a simple, 0-parameter fit computational model that used diffusion rates for single- and double-stranded DNA and reaction rate constants for the designed reactions. We assumed that the reaction rate constants for the strand displacement were the same as those measured in free solution. Because diffusion rates can be strongly affected by the surrounding medium, we first measured the diffusion rates of single- and double-stranded DNA oligonucleotides of sizes approximately equal to those used in our reactions. We built a two-chamber agarose system where approximately 1/3 of the total length of the cell contained a 1% agarose gel, which initially contained 100 nM of fluorescently labeled DNA species whose diffusion constant was to be measured.

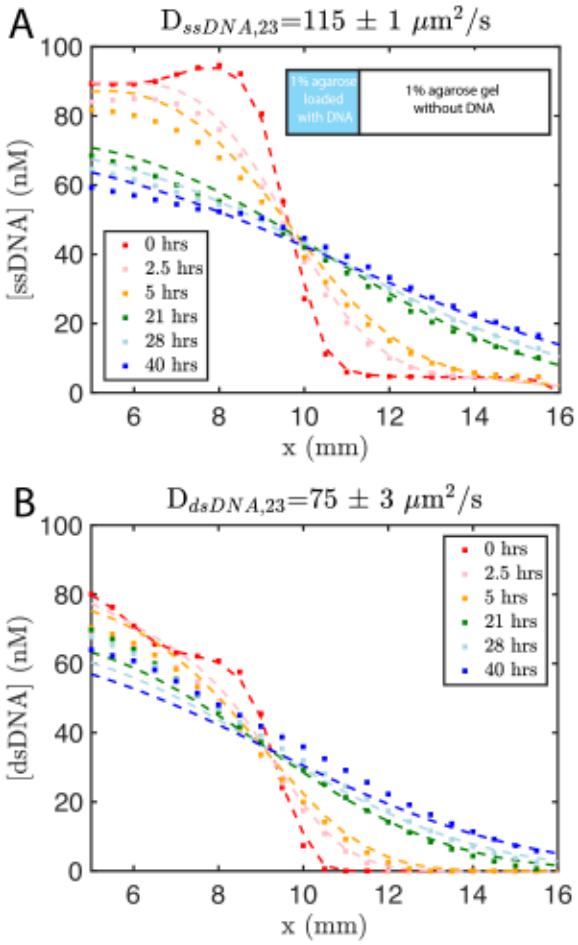

Figure 11.2. Diffusion coefficient measurement for ssDNA and dsDNA in 1% agarose. Experimental data (square markers) and simulation of the best fit to the diffusion coefficient (dashed line) for (A) ssDNA and (B) dsDNA. The initial conditions for both experiments are shown in the inset in (A). Diffusion coefficients were fit to experimental values using a least squares fitting method; simulations show the predictions of diffusion with the best fit. 0 hours indicates the time at which imaging started, which was about 30 minutes after the fluorescently labeled DNA in agarose was added to the reaction channel. Both DNA species are 23 base pairs long. Sequences are listed in section 12.6.

The remaining 2/3 was composed of 1% agarose gel without DNA (Methods). We followed the spread of the species using time-lapse fluorescence microscopy (see Methods). Using standard diffusion equations, we fit a diffusion rate constant of D_(ssDNA,23)=115±1 μm2/s, for single-

stranded DNA or ~75% the reported value for DNA of similar size in solution56,57, and D_(dsDNA,23)=75 ±3 μm2/s (Fig. 2). Using these values, and those assumed for the reaction rate constants, we modeled the growing gradient (Fig. 12.3) and the stable gradient (Fig. 12.4) to ensure the profiles would form as intended.

11.2.4 | Gradient generation by diffusion alone. We next characterized how diffusion gradients form in the absence of any release or recapture reactions. We placed a buffer solution containing 300 nM of the Output in the right-hand liquid reservoir and a buffer solution with no Output in the opposing reservoir. For this reactionless configuration, Eqn. 1 becomes

$$\frac{\partial o}{\partial t} = D_O \nabla^2 [O]. \qquad (6)$$

Setting Eqn. 6 to zero gives a steady state profile as a function of position x $[O](x)=[O]_{reservoir}*(x/L)$, a linear concentration gradient where $[O]_{reservoir}$ is the concentration of O in the high concentration reservoir and L is the length of the hydrogel.

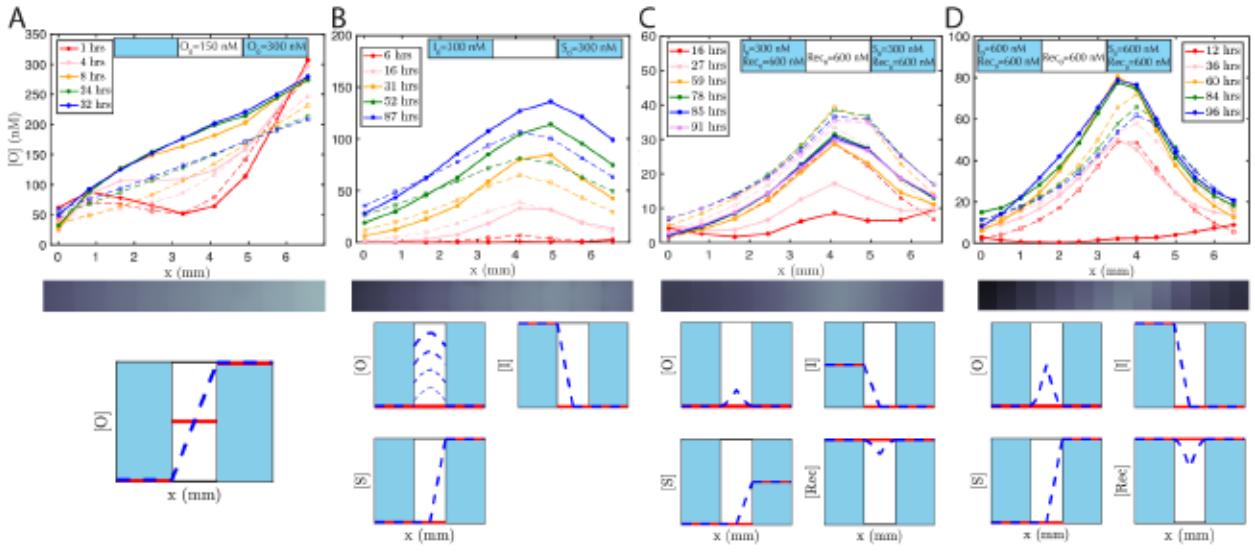

Figure 11.3. Pattern formation using DNA strand displacement reactions within an agarose hydrogel. The boundary conditions are controlled by liquid reservoirs on opposing sides of the agarose hydrogel, which

contain defined concentrations of reactants and which were periodically replenished manually. Initial conditions established through loading the liquid reservoirs and the agarose hydrogel with reactants. Solid lines show the concentrations of the output species determined from micrographs (see Methods). Dotted lines are the predictions of zero parameter-fit simulations (see Methods). (A) Linear gradients, (B) "hill" gradients that grow over time and (C-D) stable hill gradients. Each gradient forms as predicted by a partial-differential equation model of reaction-diffusion. The initial concentration of the reporter complex is $[Rep]_0$=200 nM in both the liquid reservoirs and the agarose hydrogels for all systems. The y-axis scale varies between plots. Exchange of the buffer in the reservoirs occurred in the experiment after (A) 22 hours, (B) 15 and 65 hours, (C) 24 and 52 hours, and (D) 48, 68 and 89 hours. Fluorescence micrograph depicted underneath each plot shows the state of the gradient after the longest time listed in the legend. Qualitative representation of initial conditions (red line) and concentration profile at steady state (blue dashed line) are depicted beneath plots, where the two light blue regions indicate liquid reservoirs and the white region indicates hydrogel. In the growing hill gradient, increasing time is shown as increasing line width. [O], [I], [S], [Rec], indicate the concentrations of Output, Initiator, Source and Recapturer, respectively.

To accelerate gradient development the hydrogel was initially loaded with 150 nM of Output. 200 nM of Reporter was added to the reservoirs and the hydrogel. The expected linear gradient formed by about 8 hours and remained stable over at least 24 hours (Fig. 3A). The dynamics of formation were consistent with simulations that used the measured diffusion rates. We also verified that the expected linear gradient formed in response to different initial concentrations of Output in the reservoirs and hydrogel, and where the total amount of DNA in the hydrogel would need to change to reach steady state (Fig. 12.5).

11.2.5 | Coupled release and diffusion form growing gradients. To demonstrate that designed gradient patterns could also be formed using coupled reaction and diffusion, we loaded buffer containing 300 nM of Initiator species into one reservoir and buffer containing 300 nM of

Source species into the opposite reservoir. Source and Initiator can diffuse into the gel and react to release the Output species at the point where their respective diffusion gradients meet. The height and width of the Output gradient should increase over time as Output continues to be released and diffuses outward (Fig. 12.3). Consistent with these predictions, a hill-shaped Output gradient formed (Fig. 3B). The gradient of Output that formed grew and broadened for at least 87 hours, with Output concentrations reaching 150 nM at the central peak. To maintain the boundary conditions, the buffers in the reservoirs were replaced with fresh solution with the stated concentrations of Reporter, Source, and Initiator about once per day. Exact times when the reservoirs were refreshed are stated in the figure captions.

11.2.6 | Stable gradients form through balanced release and recapture. To build stable gradients, we next combined the release and recapture reactions within the hydrogel. We added 300 nM of Initiator species into one reservoir, 300 nM of Source species into the second reservoir and 600 nM of Recapturer species into both reservoirs and the hydrogel portion of the RD cell (see Methods). The high concentration of Recapturer ensured that it would not be depleted significantly through interaction with the Output, and that its concentration would therefore remain stable across the substrate given that the solutions in the reservoirs would be replaced about once every 24 hours. A gradient of Output species emerged over 30-60 hours with higher concentrations of Output near the middle of the hydrogel and lower concentrations of the edges (Fig. 3C, Fig. 12.6). After 60 hours, the shape of the gradient reached a shape that remained stable for an additional 30 hours. Zero parameter-fit simulations matched experiments closely: in both the simulations and experiments, gradients formed the same stable shape, and the approach to stability and the time scale at which a stable shape is achieved were also similar.

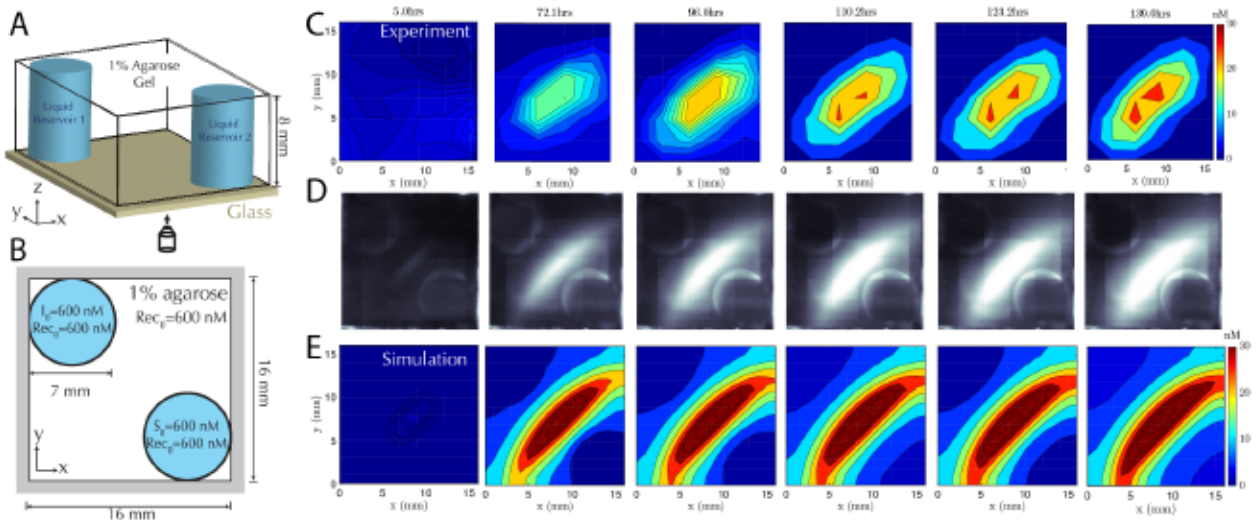

Figure 11.4. Stable two-dimensional gradient. (A) Schematic of the two-dimensional RD cell. Two cylindrical liquid reservoirs are at opposing corners of the RD cell. (B) Schematic of the top view of the 2D RD cell with initial and boundary conditions for the reactants. Reporter concentration is R_0=200 nM in both cylindrical liquid reservoirs and the agarose gel. (C) Filled contour plots depicting the concentrations of the Output at times ranging from 5 hours (left) to 139 hours (right). The stable gradient within the hydrogel takes approximately 110 hours to develop and is stable until the experiment was terminated at 139 hours. Exchange of the buffer in the reservoirs occurred in the experiment after 22, 48, 71, 93 and 116 hours. (D) Corresponding fluorescence micrographs (see Methods). (E) Filled contour plots depicting the simulated values of Output concentration profile from 5 to 139 hours.

11.2.7 | Stable gradient height is controlled by boundary conditions. Higher concentrations of Source and Initiator on the boundary should increase the rate of Output release within the hydrogel and thus increase gradient height. To test this prediction, we assembled a stable gradient using the Source, Initiator and Recapturer species where Source and Initiator concentrations in their two reservoirs were 600 nM, double the concentrations used in the first experiment (Fig. 3D). The resulting gradient shape was higher as expected, and stabilized at least as quickly as the gradient formed using lower concentrations of the Source and Initiator, consistent with predictions, and remained stable for 30 hours.

11.2.8 | Stable reaction-diffusion in two dimensions. Having characterized the formation of 1-dimensional patterns of DNA species, our next goal was to characterize the formation of 2-dimensional patterns using a similar process. To enable control over the boundary conditions in such a way that they could be maintained through the replenishment of external reservoirs, we fabricated a square RD cell (Methods) containing two cylindrical liquid reservoirs positioned in opposite corners of the hydrogel (Fig. 4A, B). We first added Source to the buffer within one reservoir and Initiator molecules the liquid reservoir in the other for cells of two different sizes. The dynamics of gradient growth followed the predictions of simulations in both cases, and as expected, gradients arose more quickly in the smaller system, where the Source and Initiator needed to diffuse a smaller distance from their reservoir to react and release the output (SI Figs. S7, S8).

To test that stable, two-dimensional gradients of Output concentration could form, we loaded one reservoir with 600 nM of Source complex and the other with 600 nM of Initiator species, with 600 nM of Recapturer in both reservoirs and in the hydrogel. The gradient produced a peak concentration of about 30 nM in 110 hours, which was stable for a further 29 hours (Fig. 4C-E). Simulation generally captured the behavior of the stable two-dimensional gradient, although slightly higher peak amplitudes were predicted. We expect this difference could result from non-uniform hydrogel composition along the reservoir interface, which can reduce local diffusion rates and thus reduce the flux of reactants into the hydrogel, while the model assumes a homogeneous gel. These variations could not be captured in our 0-parameter fit simulations.

11.2.9 | DNA systems with different sequences form multiplexed gradient patterns. DNA strand-displacement systems are of interest as a substrate for programming RD processes not only because the reaction rates of the components can be controlled, but also because multiple reaction processes involving different sequences can operate together with minimal crosstalk or as

coupled reaction processes to produce more complex patterns. To verify that such scaling is feasible, we developed two non-interacting (i.e., orthogonal) sets of reactions for the Release, Recapture and Reporting of Output species and tested whether they could be executed in parallel. We refer to the initial set of complexes as the 1-2 system and the new set of complexes as the 3-4 system, after the numbered domains within each system. We used NUPACK58 to determine the secondary structures that could form between the strands of the complexes and ensured that no more than 1% of the complexes were predicted to have structures other than those that were designed at equilibrium. For the Reporter in the 1-2 system, a 3' Iowa Black FQ quenches 5' fluorescein (FAM) fluorophore (Figure 1A(ii)) and for the 3-4 system, a 3' Iowa Black RQ quenches 5' Texas Red® fluorophore (see sequence information in section 12.6).

To form stable, multiplexed gradients, we loaded one reservoir in a 1-dimensional RD cell with 300 nM of each Initiator, the other reservoir with 300 nM of each Source and in each liquid reservoir and hydrogel loaded 300 nM Recapturer for each system (Fig. 5A). The reporter complexes for each system were presented at 200 nM in both the hydrogel and reservoirs. Both gradients achieved the expected shape and approached a final stable state, consistent with the predictions of simulations (Fig. 5B-C). Each gradient took approximately the same amount of time to form, but the 1-2 gradient had a maximum concentration of about 60 nM, whereas the 3-4 gradient had a maximum concentration of about 40 nM. The 3-4 gradient was not yet stable after about 85-95 hours with increases in output concentration on the order of a few percent per hour over 88-94 hours, whereas the 1-2 gradient appeared to have achieved a stable configuration by this time. Sequence-specific differences in reaction rate constants can vary by up to an order of magnitude[40] in strand displacement processes. The differences in the precise maximum concentrations and time constants between the two systems are consistent with such differences in reaction rate constants. In the future, modeling software that takes sequence data into consideration

could be used to control this variation[59].

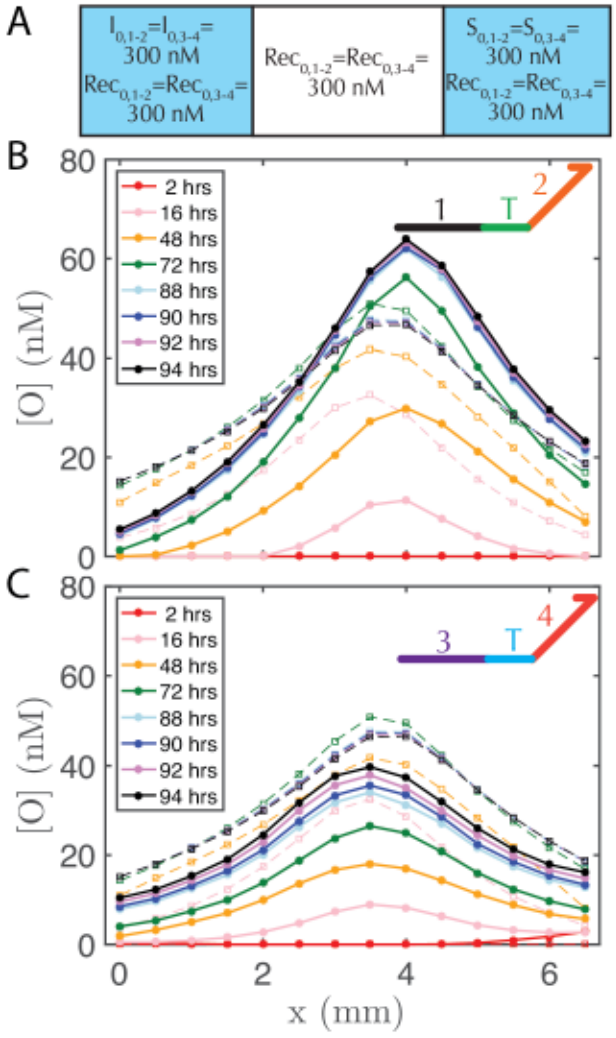

Figure 11.5. Two stable, overlapping gradients of Output species can be formed in one hydrogel by two non-interacting (orthogonal) sets of reactants (denoted 1-2 and 3-4 for their recognition domains, see inset cartoon diagrams of Output species in B-C).). (A) Initial conditions of reactants in the 1-2 and 3-4 systems in the liquid reservoirs and hydrogel. Initial reporter concentrations are $R_{(0,1-2)}$=$R_{(0,3-4)}$=200 nM in both liquid reservoirs and the agarose hydrogel and are not depicted for clarity. Concentration profiles for the Output strands from the (B) 1-2 system and (C) 3-4 system. The Reporter in the 1-2 system has a FAM fluorophore, whereas the Reporter complex in the 3-4 system has a Texas-Red fluorophore, so the respective fluorescence profiles (which are then converted to concentration profiles) were measured using non-

overlapping filters for FAM and Texas Red channel (see Methods). Buffer in the reservoirs was exchanged after 48 hours.

11.3 | Discussion

In this work we have shown how DNA strand displacement reactions operating far from equilibrium can be used to engineer stable reaction-diffusion gradients. Such patterns evolve over time in a predictable fashion. Much of the vast potential parameter space for such systems remains to be explored, including altering the reaction rate constants (e.g., changing the length of the toehold domain[39,40]) and/or the diffusion rates (e.g., by introducing hydrodynamic drag[56] or by altering length of the DNA components[57]), which could also serve to sharpen the spatial resolution. External reservoirs allow us to refuel the system, enabling far-from-equilibrium patterns to be sustained for at least tens of hours, and no fundamental time limit for such stability was observed. It may be possible to speed gradient formation by increasing the hydrogel pore size by using a lower percentage agarose, or miniaturizing the size scale of the RD cell to a microfluidic device. Device miniaturization could also improve the spatial resolution, which is currently on the order of hundreds of μm. To build more sophisticated systems, it will also be important to reduce undesirable leak reactions between reactants[59-62].

The stable release and recapture reaction mechanism could enable the implementation of self-regenerating patterns, which return to a target stable structure after they are perturbed by some external stimulus (e.g., a high concentration of a reactant in a strand displacement system, or a transient flow). To characterize the capacity for regeneration, methods for reliable perturbation are required, such as through the use of light-driven release. Finally, downstream Boolean logic operations could facilitate the generation of patterns with arbitrarily complex shape[49], dynamics[50], and functional responses to changing environmental conditions. The use of biocompatible reactants,

hydrogels and strand-displacement reactions, which can operate under a variety of conditions, mean that one could envision interacting with biological systems[63] and other downstream processes[64,65].

Materials and Methods

DNA complex preparation. All DNA strands were ordered from Integrated DNA Technologies (IDT) with standard desalting except fluorescently modified strands, which were HPLC purified. Complexes were formed by mixing the component strands at equimolar ratio in TAE Mg^{2+} (40 mM Tris-Acetate, 1 mM EDTA buffer containing 12.5 mM Magnesium Acetate) and then placed in an Eppendorf Mastercycler PCR, where the strands were annealed. Annealing consisted of holding the temperature at 95 °C for 5 minutes and then cooling the solution to 25 °C at a rate of -1 °C per minute. After annealing, each complex was purified by polyacrylamide gel electrophoresis (PAGE) using a 10% polyacrylamide run at 120 V for 90 minutes at 4 °C (see Ref. 55 for more details). The bands corresponding to the complexes were identified using UV-shadowing at a wavelength of 254 nm. The band was then diced, combined with TAE Mg2+ buffer into a tube and shaken on a vortexer for about 12 hours, to promote complex migration into the aqueous solution. The solution was then centrifuged for 5 minutes at 3000x g and the supernatant removed, which was repeated twice to ensure separation of gel from solution. Concentration measurements were obtained using an Eppendorf Biophotometer. The extinction coefficient, ε, of a complex was approximated by the formula: ε=ε_top+ε_bottom-3200N_AT-2000N_GC, where ε_top and ε_bottom are the extinction coefficients of the two strands that comprise the complex and N_AT and N_GC are the number of hybridized A-T and G-C base pairs in the complex[66].

Hydrogel preparation. Agarose gels with DNA complexes were prepared by mixing liquid agarose and complexes and then cooling the gels in devices to set. We prepared 1% agarose hydrogel (1 g / 100 mL) in TAE Mg^{2+} and left it to cool to 40 °C, after which we transferred the agarose

solution to a glove box with PID fan temperature control (Coy Labs) set to 40 °C. Buffers, DNA complex solutions, pipettes and pipette tips were left in the glove box at least 30 minutes prior to sample preparation to achieve thermal equilibrium. The agarose was mixed with the DNA solution (typically with Reporter and any other complex that was required for the experiment) and the resulting mixed was transferred to the device in the desired well(s). After all wells were patterned, a piece of Scotch tape was adhered to the PDMS to seal the wells and the device was transferred to the refrigerator at 4 °C for 15 minutes to set the gels. We found that 40 °C was hot enough for the agarose to remain a liquid, but NuPack's[58] compute melt function predicts 40 °C is well below the melting temperature of our DNA complexes. The reservoir solutions were added after the gel had been cooled to room temperature and all results were collected at room temperature. Glass coverslips were placed on top of the device to mitigate evaporation from the calibration and reaction wells during the reaction-diffusion process (see more discussion on evaporation in Fig. 12.9).

Reaction Monitoring. The reaction was monitored using time-lapse fluorescence microscopy on either an IX73 or IX71 (Olympus) optical microscope. Image sets were obtained every ~30 minutes with an exposure of 50-150 ms using a 4X, 10X or 20X objective (Olympus) and were captured by an Infinity 3 CCD camera (Lumenera Corporation) in non-overlapping FAM and Texas-Red channels (Chroma) on the IX73, or an iXon3 cooled EMCCD camera (Andor) using a FAM channel (Chroma) on the IX71. The center of the focal plane for each experiment was approximately 1 to 4 mm in the z direction above the glass slide, as determined by the minimum width of the light beam in the direction perpendicular to the optical axis. We post-processed the images via binning and dark frame correction to compress the data and eliminate some of the optical artifacts, respectively (Fig. 12.10).

Diffusion coefficient measurement. Diffusion coefficients were fit to experimental values using a least squares fitting method; simulations show the predictions of diffusion with the best fit. The system consisted of a 1% agarose gel that initially contained no DNA as shown and was cast and left at room temperature to gel for approximately 30 minutes prior to adding the gel with DNA. After this set time, the remaining ~1/3 of the reaction cell was loaded with reaction buffer that contained ~75 to 100 nM of fluorescently labeled DNA. 0 hours indicates the time at which imaging started, which was about 30 minutes after the fluorescently labeled DNA in agarose was added to the reaction channel. Starting concentration profile used in the simulation was taken from the initial experimental concentration profile (*i.e.*, at 0 hours).

Liquid Reservoir Exchange. Contents of the liquid reservoirs are removed with a transfer pipette and replaced with fresh reaction buffer (prepared <10 minutes prior to exchange). Reservoir solution exchanges were performed approximately every 24 hours and are stated in the figure captions.

Simulations. To perform the simulations, we used COMSOL Multiphysics® Version 4.4 and LiveLink™ for MATLAB. Models of reaction-diffusion channels were built using COMSOL with the "Transport of Diluted Species" physics. All hydrogel-PDMS and liquid-PDMS boundaries were simulated with no flux boundary conditions. Simulations were run using scripts written in MATLAB using COMSOL Java API commands, which is how we defined simulation parameters such as reactions and their rate constants, diffusion constants, initial conditions and boundary conditions, mesh size and buffer exchange times. Buffer exchange of the liquid reservoirs occurred in simulations at the same time points as those in our experiments. Buffer exchange consisted of replacing each liquid reservoir with their initial contents (unreacted species) while the concentration profiles of species in the hydrogel remained unchanged and to simulate a particular process, the

specific timing of buffer exchange from that experiment was included in the simulation. Diffusion coefficients for the hydrogel domain were set to measured values. Diffusion coefficients in the liquid reservoirs were all equal and had a value of 150 μm2/s[57,67]. Reaction rate constants for intended reactions were set according to estimated values in solution (see Ref. 40) and reaction rate constants for leak reactions were estimated (see Fig. 12.1). COMSOL models and MATLAB scripts are available upon request.

Acknowledgements

The authors would like to thank Deepak Agrawal, Angelo Cangialosi, Qi Huang, Abdul M. Mohammed, and Samuel Schaffter for helpful discussions, and Markela Ibo for her assistance with fabrication techniques for reaction-diffusion devices. This work was supported by DOE BES grant 90068952 (for support of J.F. and J.Z. and some materials and supplies) and NSF CCF grant 1161941 to R.S.

References

1. M. Akam, Development, 1987, 101, 1-&.
2. J. B. A. Green and J. Sharpe, Development, 2015, 142, 1203-1211.
3. P. Bastiaens, M. Caudron, P. Niethammer and E. Karsenti, Trends Cell Biol, 2006, 16, 125-134.
4. B. N. Kholodenko, Febs Lett, 2009, 583, 4006-4012.
5. N. L. Jeon, H. Baskaran, S. K. W. Dertinger, G. M. Whitesides, L. Van de Water and M. Toner, Nat Biotechnol, 2002, 20, 826-830.
6. G. S. Jeong, S. Han, Y. Shin, G. H. Kwon, R. D. Kamm, S. H. Lee and S. Chung, Anal Chem, 2011, 83, 8454-8459.
7. B. G. Chung, L. A. Flanagan, S. W. Rhee, P. H. Schwartz, A. P. Lee, E. S. Monuki and N. L. Jeon, Lab Chip, 2005, 5, 401-406.
8. S. K. W. Dertinger, X. Y. Jiang, Z. Y. Li, V. N. Murthy and G. M. Whitesides, P Natl Acad Sci USA, 2002, 99, 12542-12547.
9. Y. Shin, S. Han, J. S. Jeon, K. Yamamoto, I. K. Zervantonakis, R. Sudo, R. D. Kamm and S. Chung, Nat Protoc, 2012, 7, 1247-1259.
10. C. A. DeForest, B. D. Polizzotti and K. S. Anseth, Nat Mater, 2009, 8, 659-664.
11. E. Karzbrun, A. M. Tayar, V. Noireaux and R. H. Bar-Ziv, Science, 2014, 345, 829-832.
12. J. K. He, Y. A. Du, J. L. Villa-Uribe, C. M. Hwang, D. C. Li and A. Khademhosseini, Adv Funct Mater, 2010, 20, 131-137.
13. V. Sourjik and N. S. Wingreen, Curr Opin Cell Biol, 2012, 24, 262-268.
14. L. D. Zarzar, Q. H. Liu, X. M. He, Y. H. Hu, Z. G. Suo and J. Aizenberg, Soft Matter, 2012, 8, 8289-8293.
15. M. Mayer, J. Yang, I. Gitlin, D. H. Gracias and G. M. Whitesides, Proteomics, 2004, 4, 2366-2376.
16. S. H. Lee, J. J. Moon and J. L. West, Biomaterials, 2008, 29, 2962-2968.
17. M. S. Hahn, J. S. Miller and J. L. West, Adv Mater, 2006, 18, 2679-+.

18. S. K. W. Dertinger, D. T. Chiu, N. L. Jeon and G. M. Whitesides, Anal Chem, 2001, 73, 1240-1246.
19. J. Atencia, J. Morrow and L. E. Locascio, Lab Chip, 2009, 9, 2707-2714.
20. F. Lin, W. Saadi, S. W. Rhee, S. J. Wang, S. Mittal and N. L. Jeon, Lab Chip, 2004, 4, 164-167.
21. J. P. Diao, L. Young, S. Kim, E. A. Fogarty, S. M. Heilman, P. Zhou, M. L. Shuler, M. M. Wu and M. P. DeLisa, Lab Chip, 2006, 6, 381-388.
22. F. Sagues and I. R. Epstein, Dalton T, 2003, DOI: 10.1039/b210932h, 1201-1217.
23. V. K. Vanag and I. R. Epstein, Chaos, 2008, 18.
24. S. N. Semenov, A. J. Markvoort, W. B. L. Gevers, A. Piruska, T. F. A. de Greef and W. T. S. Huck, Biophys J, 2013, 105, 1057-1066.
25. M. Loose, E. Fischer-Friedrich, J. Ries, K. Kruse and P. Schwille, Science, 2008, 320, 789-792.
26. D. G. Miguez, V. K. Vanag and I. R. Epstein, P Natl Acad Sci USA, 2007, 104, 6992-6997.
27. E. Jee, T. Bansagi, A. F. Taylor and J. A. Pojman, Angew Chem Int Edit, 2016, 55, 2127-2131.
28. M. M. Wrobel, T. Bansagi, S. K. Scott, A. F. Taylor, C. O. Bounds, A. Carranza and J. A. Pojman, Biophys J, 2014, 106, 1548-1548.
29. S. N. Semenov, A. J. Markvoort, T. F. A. de Greef and W. T. S. Huck, Angew Chem Int Edit, 2014, 53, 8066-8069.
30. M. Isalan, C. Lemerle and L. Serrano, Plos Biol, 2005, 3, 488-496.
31. A. Padirac, T. Fujii, A. Estevez-Torres and Y. Rondelez, J Am Chem Soc, 2013, 135, 14586-14592.
32. A. S. Zadorin, Y. Rondelez, J. C. Galas and A. Estevez-Torres, Phys Rev Lett, 2015, 114.
33. T. Bansagi, V. K. Vanag and I. R. Epstein, Science, 2011, 331, 1309-1312.
34. I. R. Epstein and B. Xu, Nat Nanotechnol, 2016, 11, 312-319.
35. H. W. H. van Roekel, B. J. H. M. Rosier, L. H. H. Meijer, P. A. J. Hilbers, A. J. Markvoort, W. T. S. Huck and T. F. A. de Greef, Chem Soc Rev, 2015, 44, 7465-7483.
36. G. Seelig, D. Soloveichik, D. Y. Zhang and E. Winfree, Science, 2006, 314, 1585-1588.
37. L. Qian and E. Winfree, Science, 2011, 332, 1196-1201.
38. Y. J. Chen, B. Groves, R. A. Muscat and G. Seelig, Nat Nanotechnol, 2015, 10, 748-760.
39. D. Y. Zhang and G. Seelig, Nat Chem, 2011, 3, 103-113.
40. D. Y. Zhang and E. Winfree, J Am Chem Soc, 2009, 131, 17303-17314.
41. D. Y. Zhang, A. J. Turberfield, B. Yurke and E. Winfree, Science, 2007, 318, 1121-1125.
42. Y. J. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik and G. Seelig, Nat Nanotechnol, 2013, 8, 755-762.
43. D. Soloveichik, G. Seelig and E. Winfree, Proc Natl Acad Sci U S A, 2010, 107, 5393-5398.
44. L. Qian and E. Winfree, J R Soc Interface, 2011, 8, 1281-1297.
45. D. Y. Duose, R. M. Schweller, J. Zimak, A. R. Rogers, W. N. Hittelman and M. R. Diehl, Nucleic Acids Res, 2012, 40, 3289-3298.
46. J. Hemphill and A. Deiters, J Am Chem Soc, 2013, 135, 10512-10518.
47. S. M. Chirieleison, P. B. Allen, Z. B. Simpson, A. D. Ellington and X. Chen, Nat Chem, 2013, 5, 1000-1005.
48. P. B. Allen, X. Chen and A. D. Ellington, Molecules, 2012, 17, 13390-13402.
49. D. Scalise and R. Schulman, TECHNOLOGY, 2014, 02, 55-66.
50. D. Scalise and R. Schulman, Natural Computing, 2016, 15.2, 197-214
51. P. Yin, H. M. T. Choi, C. R. Calvert and N. A. Pierce, Nature, 2008, 451, 318-U314.
52. J. L. Leroy, M. Kochoyan, T. Huynhdinh and M. Gueron, J Mol Biol, 1988, 200, 223-238.
53. S. Nonin, J. L. Leroy and M. Gueron, Biochemistry-Us, 1995, 34, 10652-10659.
54. K. H. Hecker and R. L. Rill, Biotechniques, 1998, 24, 256-260.
55. J. Fern, D. Scalise, A. Cangialosi, D. Howie, L. Potters and R. Schulman, ACS Synthetic Biology, 2016, DOI: 10.1021/acssynbio.6b00170.
56. S. Doose, H. Barsch and M. Sauer, Biophys J, 2007, 93, 1224-1234.
57. G. L. Lukacs, P. Haggie, O. Seksek, D. Lechardeur, N. Freedman and A. S. Verkman, J Biol Chem, 2000, 275, 1625-1629.
58. J. N. Zadeh, C. D. Steenberg, J. S. Bois, B. R. Wolfe, M. B. Pierce, A. R. Khan, R. M. Dirks and N. A. Pierce, J Comput Chem, 2011, 32, 170-173.
59. J. M. Schaeffer, C. Thachuk and E. Winfree, in DNA Computing and Molecular Programming: 21st International Conference, DNA 21, Boston and Cambridge, MA, USA, August 17-21, 2015. Proceedings, eds. A. Phillips and P. Yin, Springer International Publishing, Cham, 2015, DOI: 10.1007/978-3-319-21999-8_13, pp. 194-211.
60. C. Thachuk, E. Winfree and D. Soloveichik, International Workshop on DNA-Based Computers, 2015
61. X. Olson, S. Kotani, J. E. Padilla, N. Hallstrom, S. Goltry, J. Lee, B. Yurke, W. L. Hughes and E. Graugnard, ACS Synth Biol, 2016, DOI: 10.1021/acssynbio.5b00231.
62. C. Grun, J. Werfel, D. Y. Zhang and P. Yin, J R Soc Interface, 2015, 12.

63. M. X. You, G. Z. Zhu, T. Chen, M. J. Donovan and W. H. Tan, J Am Chem Soc, 2015, 137, 667-674.
64. E. Franco, E. Friedrichs, J. Kim, R. Jungmann, R. Murray, E. Winfree and F. C. Simmel, P Natl Acad Sci USA, 2011, 108, E784-E793.
65. I. R. Epstein and B. Xu, Nat Nano, 2016, 11, 312-319.
66. J. D. Puglisi and I. Tinoco, Jr., Methods Enzymol, 1989, 180, 304-325.
67. E. Stellwagen, Y. J. Lu and N. C. Stellwagen, Biochemistry-Us, 2003, 42, 11745-11750.

12 | Supplemental Information: Stable DNA-based Reaction-Diffusion Patterns

12.1 | Leak Reactions and Recapture Kinetics

Leak reactions, or unintended reactions between DNA species which result in Output release or recapture, are a ubiquitous and well-documented phenomenon in DNA-based strand displacement systems. Strands and complexes that are less prone to leak can be designed but require more complexity[2]. Previous work [3] documented several types of leak reactions based on molecules similar to those used in this study and are the basis for reactions and leak parameters used in our models here. Such reactions are enumerated in Fig. S10.

Truncated complexes are a possible source of leak reactions. Such complexes can be created when one full-length strand and one truncated strand hybridize to form a complex that has a smaller molecular weight than the full-length complex. Such leaky complexes are difficult to distinguish from full-length complexes in a PAGE purification process. Thus, leaky complexes are unintentionally introduced into all experiments. For example, leaky Source complexes occur when the bottom strand of a Source complex is missing bases on its 5' end (Fig. S10(i)), thus providing another reaction mechanism through which a Reporter complex can interact. Further examples of such leaky complexes and their reaction mechanisms can be seen in Fig. S10(ii-iii). Optimizing PAGE protocols to reduce the leak reactions is an ongoing effort, with some changes to protocols described here in Ref. [3].

The quantity of the leaky complexes has been measured in Ref [3], where the authors measured Source strands, similar to those used in this study, with leaky toehold domains to be anywhere between 0.5 and 4.7% of $[S]_0$, and react with rate constant ~50 $M^{-1}s^{-1}$, which is the approximate rate constant for a strand displacement reaction mediated through a two base pair toehold, k_{2bp}. For the

simulations here we assume that all leak reactions through a truncated strand occur at a reaction rate constant of $k_{2bp} = 50$ $M^{-1}s^{-1}$ and that $[S]_{leak,0} = 0.047[S]_0$, as the bottom strand of the Source species in this paper is longer than in Ref. [3] and therefore we expect the truncation error to be on larger end of the measured range. We also assume that a leaky Recapturer complexes (Fig. S9(iii)) also have an initial concentration of $0.047[Rec]_0$ and react with rate constants of k_{0bp}. Although leaky (i.e. unintended), the leak pathway in a recapture reaction is slower and almost insignificant when compared to the primary mechanism of recapture.

While a strand displacement reaction mediated by a nick in a backbone of a DNA duplex, as seen in the Recapturer complex, would typically be expected to produce strand displacement kinetics with forward rates of 0.5 to 5 $M^{-1}s^{-1}$, (which approximately correspond to 0 and 1 bp toehold mediated reaction kinetics, respectively) we found that the Recapture reaction was best modeled with a reaction rate constant of 500 $M^{-1}s^{-1}$ more in line with a 3 bp toehold mediated reaction [1]. Significant fraying of ends can explain the increase in kinetic rates. When analyzed with NUPCK at room temperature, the 1-2 and 3-4 Recapturer complexes have ~50% and ~60% probability, respectively, of having an unpaired base at the nick. To mitigate this fraying in a next generation design, a GC rich region that occludes fraying, for example, can be designed.

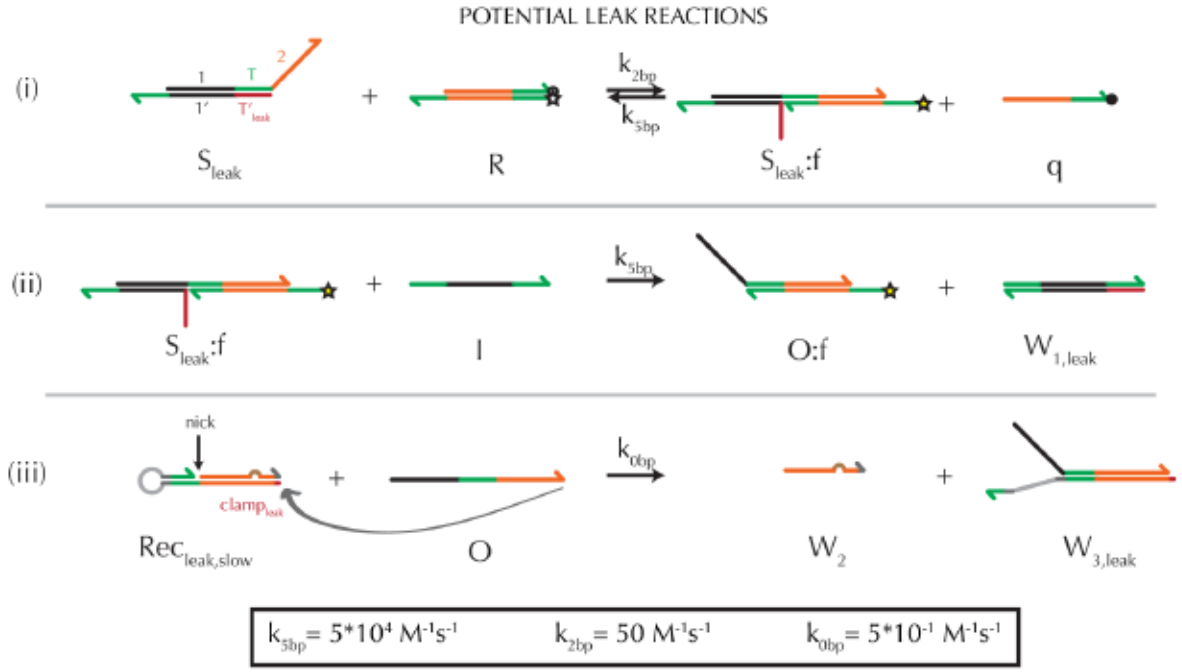

Figure 12.1. Potential leak reactions considered in the model. (i) Reversible reaction between a leaky Source complex ($\boldsymbol{S_{leak}}$) and the Reporter complex, where the leaky Source complex has a bottom strand that is missing some bases (shown in red and labeled $\mathbf{T'_{leak}}$). Reaction rate constants shown above and below arrows and their values listed at the bottom of the figure and are estimates from Ref. [1]. We assume a leaky toehold has a -2bp truncation [3] at the 5' end (IDT synthesizes DNA from 3' to 5', thus a truncation error will occur at the 5' end of a strand) and is shown in red. Although we PAGE purify the complexes, we found it challenging to distinguish between complexes with lengths that differ by only a few base pairs in our gels; the primary reason for PAGE purification is to ensure proper stoichiometric ratios of all strands in a complex. Such a source of error could be likely eliminated by purification methods which were able to differentiate between small changes in number of base pairs in a complex, or by ordering purified ssDNA strands. (ii) Irreversible reaction between a leaky Source complex with a fluorophore strand ($\boldsymbol{S_{leak}:f}$) and an Initiator through a 5bp toehold-mediated strand displacement reaction to produce an Output:fluorophore and waste complex. (iii) Irreversible reaction between a leaky Recapturer with slow kinetics, where the 5' end of the bottom strand in the Recapturer has two missing bases, therefore the non-reactive clamp would be

eliminated, exposing a potential 0bp strand displacement reaction between an Output and the Recapturer. Gray arrow indicates the trajectory of the strand displacement reaction.

Table 12.1: List of reactions modeled in RD system.

Release	$\boldsymbol{S + I \xrightarrow{k_{5bp}} O + W_1}$	(1)
Recapture	$O + Rec \xrightarrow{k_{nick}} W_2 + W_3$	(2)
Reporting	$O + R \underset{k_{5bp}}{\overset{k_{5bp}}{\rightleftharpoons}} f + q$	(3)
LEAK (i)*	$S_{leak} + R \underset{k_{5bp}}{\overset{k_{2bp}}{\rightleftharpoons}} S_{leak}{:}f + q$	(4)
LEAK (ii)*	$S_{leak}{:}f + I \xrightarrow{k_{5bp}} O{:}f + W_{1,leak}$	(5)
Leak (iii)*	$Rec_{leak,slow} + O \xrightarrow{k_{0bp}} W_2 + W_{3,leak}$	(6)

*Leak reactions correspond to those listed in Fig. S10.

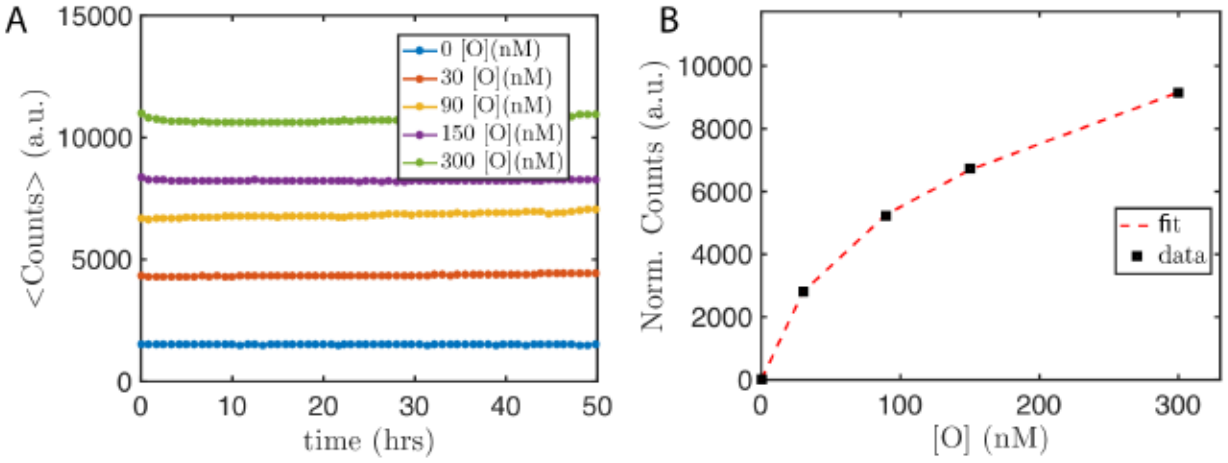

Figure 12.2. Calibrating fluorescence. (A) A typical calibration plot depicting counts from the calibration wells as a function of time. Counts are averaged across entire image. Wells have varying concentration of Output but have a constant amount of Reporter $[\boldsymbol{R}]_{\boldsymbol{0}} = \boldsymbol{200}$ nM in each well. Images are typically taken with an exposure ranging from 50 to 150 ms. (B) The counts are averaged over a period of time, typically 20 hours, and then normalized to zero (such that the average counts at $[\boldsymbol{O}] = \boldsymbol{0}$ nM are 0). The data is fit to the curve $\boldsymbol{C_N} = \frac{\boldsymbol{a}[\boldsymbol{O}]^2+\boldsymbol{b}[\boldsymbol{O}]+\boldsymbol{c}}{[\boldsymbol{O}]+\boldsymbol{d}}$ where $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ and $\boldsymbol{d}$ are fitted parameters, $\boldsymbol{NC}$ is the normalized counts and $[\boldsymbol{O}]$ is the Output concentration. This curve is then used to calculate the concentration of the total Output species (unbound Output species + Output species bound to the Reporter), where:

$$[\boldsymbol{O}] = \frac{\sqrt{-4\boldsymbol{ac}+4\boldsymbol{ad}\boldsymbol{C_N}+\boldsymbol{b}^2-2\boldsymbol{b}\boldsymbol{C_N}+\boldsymbol{C_N}^2}-\boldsymbol{b}+\boldsymbol{C_N}}{2\boldsymbol{a}}.$$

Where the calibration wells were only fluorescent ssDNA or dsDNA, such as in the diffusion only experiments (see Figure 2 in the main text), the calibration curve was linear and thus fit to a linear equation, $\boldsymbol{C_N} = \boldsymbol{m}[\boldsymbol{O}] + \boldsymbol{b}$ (the fit largely depends on the gain setting used). All calibration wells are comprised of 1% agarose. Calibration wells measured 8x8x8 mm (512 µL).

12.3 | One-Dimensional Gradient Simulations and Experiments

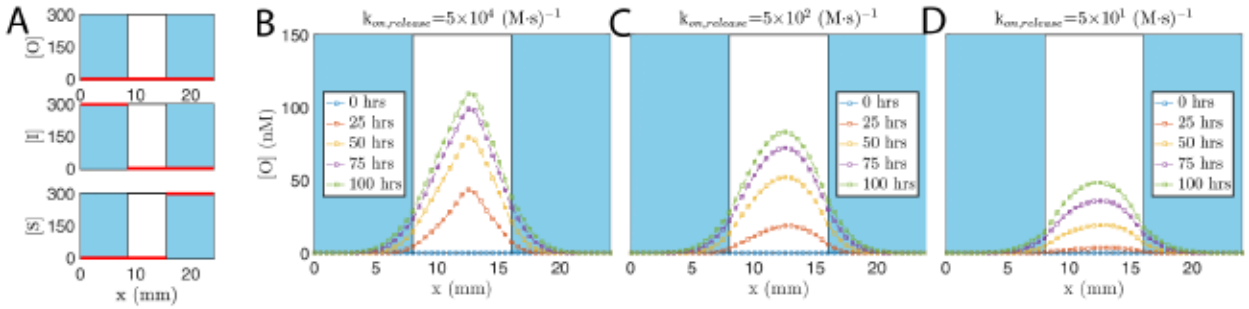

Figure 12.3. One-Dimensional Growing Gradient Simulations. To design a gradient formed by DNA strand displacement processes, we needed to both choose reactions and set the rates for these reactions by choosing the length of the toeholds that initiate the reaction process. To understand how the reaction rate constant for the release reaction would affect the shape of the gradient, we simulated the reaction and diffusion of Source and Initiator species in 1D with measured diffusion coefficients. (A) Initial conditions for the various species in the reservoirs and the agarose gel. The buffer in the liquid reservoirs was exchanged every 24 hours so that these conditions were maintained over time. Blue regions denote liquid reservoirs whereas the white region denotes hydrogel. We simulation the RD using various forward reaction rate constants for the release reaction: (B) $k_{on,release} = 5\times10^{4}$ (M s)$^{-1}$, (B) 5×10^{2} (M s)$^{-1}$ and (C) 50 (M s)$^{-1}$, which correspond to reactions mediated by approximately 5bp, 3bp, 2bp toeholds respectively[1]. (E) Schematic of the strand displacement process designed to occur at each of the rates considered in simulation. Leak reactions are not included in the simulations shown here.

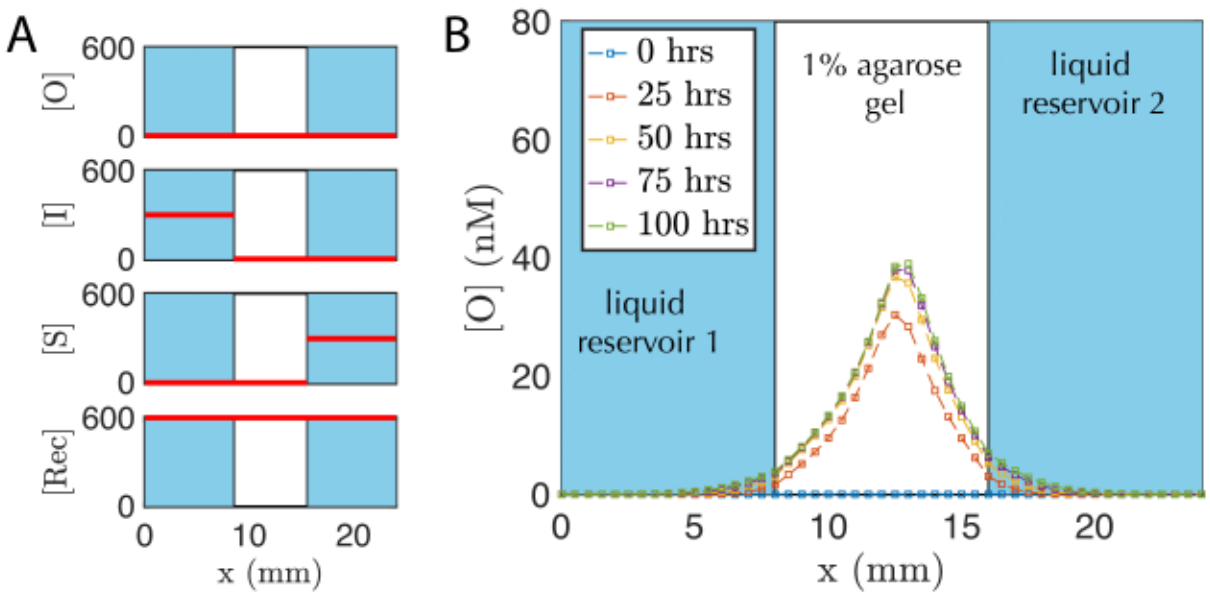

Figure 12.4. One-Dimensional Stable Gradient Simulation. To determine whether 1D gradients formed by release, recapture and reporting reactions in a 1% agarose hydrogel could be stable over time when the liquid reservoirs were periodically refreshed, we simulated the RD system with measured diffusion coefficients and assumed reaction rate constants based on measurements made in solution (see Simulation section in Methods of main text). (A) Initial conditions for the species in the reservoirs and the agarose gel assumed in simulation. Reporter concentration is not shown but is initially 200 nM in both liquid reservoirs and in the hydrogel. (B) Simulations showed that gradients could remain stable once formed. Buffer exchange was simulated in the liquid reservoirs every 24 hours. Blue regions denote liquid reservoirs whereas the white region denotes hydrogel. Leak reactions are not included in the simulations shown here. While the initial conditions of this simulation are identical to those in the experiment shown in Fig. 3C in the main text, the buffer refresh times are slightly different, thus leading to different dynamics.

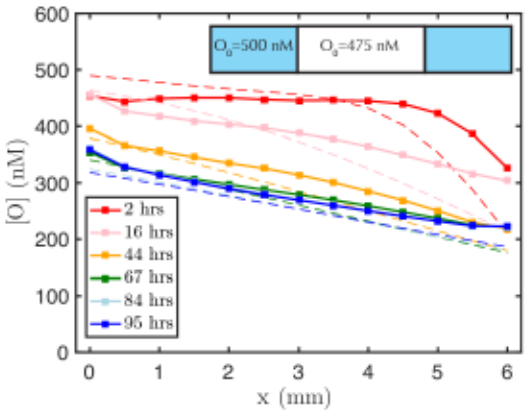

Figure 12.5. One-Dimensional Linear Diffusion Gradient. The inset shows the initial concentrations of the Output in the left reservoir and hydrogel (there is no Output initially in the right reservoir). Solid lines on the plot show the Output concentration as a function of position within the hydrogel and time. Dashed lines are results from a zero parameter-fit simulation, whereas bold lines and squares indicate measured values from experiment (see Methods). The experimentally measured Output concentration at the initial time point were used as initial conditions for the simulation. Initial Reporter concentrations are not depicted for clarity and are $R_0 = 200$ nM in both reservoirs and the hydrogel. The buffers in the reservoirs were exchanged at 24, 51 and 76 hours for fresh buffer containing the initial concentrations of the Output and Reporter.

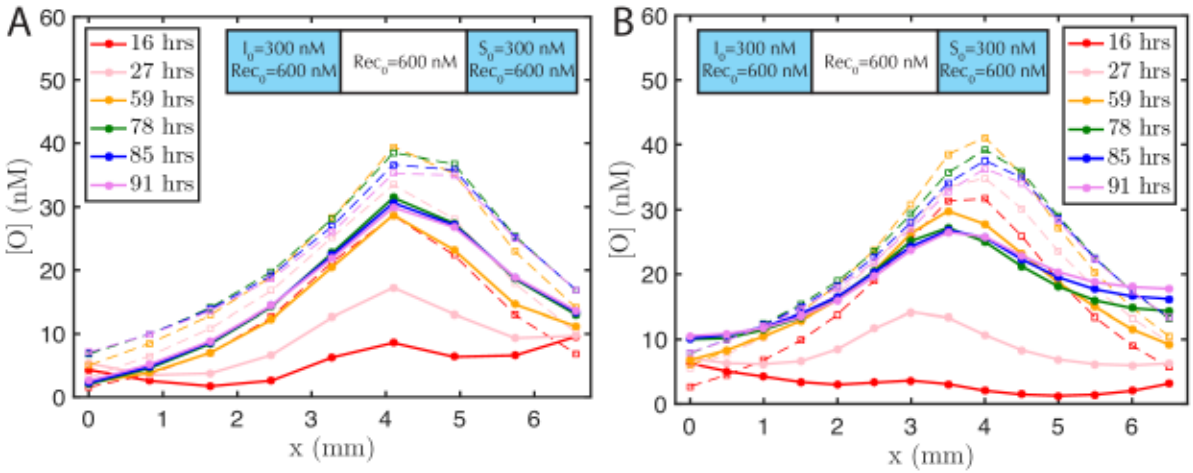

Figure 12.6. Stable gradients form consistently. To characterize the inherent variations expected in the formation of gradient patterns within RD strand displacement systems, we performed two separate experiments in which we formed two stable hill gradients with identical initial and boundary conditions from separate Source, Initiator, Reporter and Recapturer stocks. The two gradients both stabilize in similar periods of time and have similar shapes but differ in peak height and shape. Such differences could be the result in differences in gels, reservoir heights and the purity and effective concentration of the component complexes. Initial conditions of the RD system (the same in both experiments) are depicted in the insets. Initial Reporter concentration is not depicted for clarity and is $R_0 = 200$ nM in the liquid reservoirs and agarose hydrogels for all systems. Inset is a cartoon schematic of the Output species (left) and is the same for both experiments. Buffer exchange occurred after (A) 24 and 52 hours and (B) 48 hours. Differences in simulations between the two figures reflect the different buffer exchange times. (A) 10X objective (IX71 microscope) and (B) 20X objective (IX73 microscope) were used to image the systems.

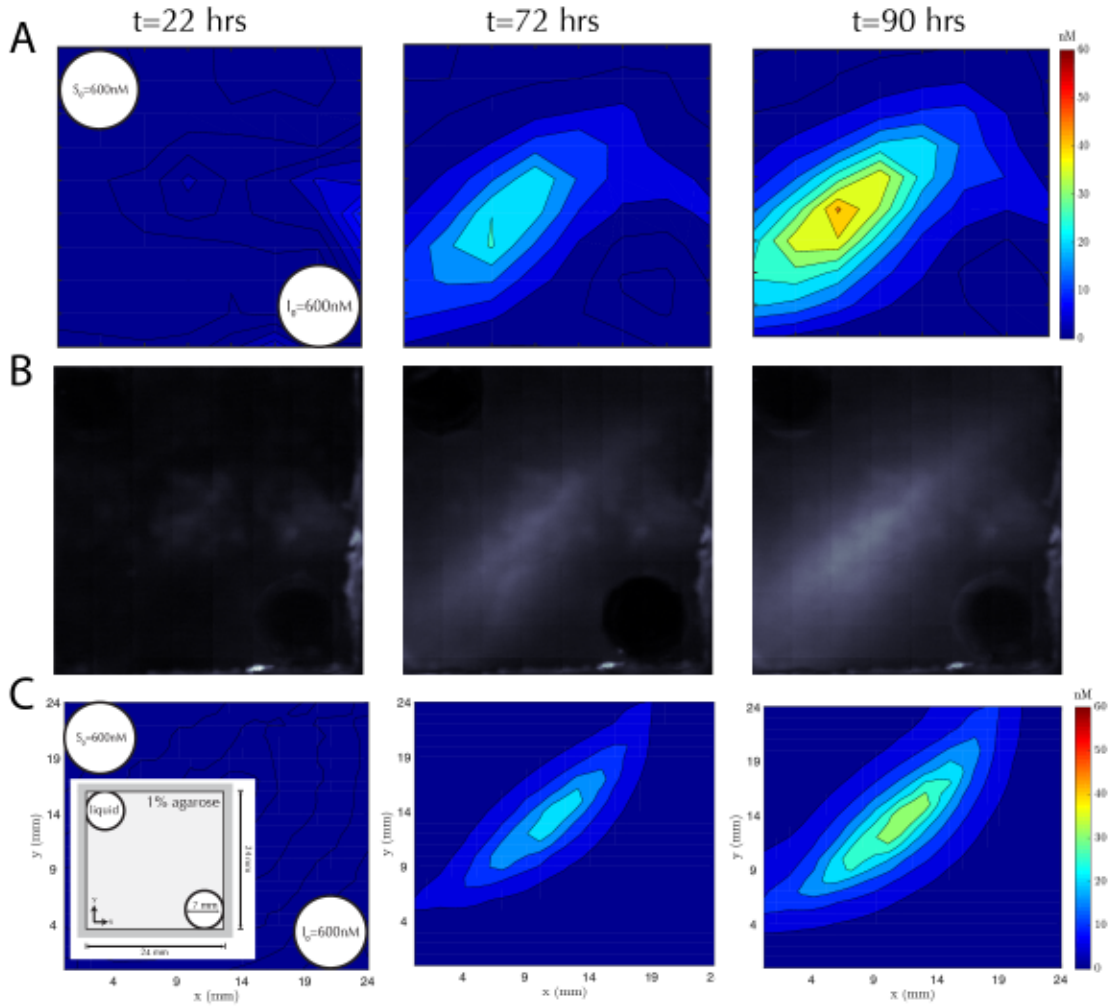

Figure 12.7. A large (24 x 24 x 8 mm), growing two-dimensional gradient. (A) Contour map of the l concentration of Output species in experiment and corresponding (B) optical images used to determine concentrations and (C) simulations at three time points: 22, 72 and 90 hours. The concentrations of source and initiator in the reservoirs at reaction start and after refresh are shown in A and C at 22 hours. Reporter concentration is $R_0 = 200$ nM in both liquid reservoirs and in the agarose gel and is not depicted for clarity. Inset diagram in simulation contour map at 22 hours depicts the schematic for the reaction cell. Fluorescent micrographs are obtained in a raster fashion and stitched together as a mosaic, as the field of view of the 4X objective used is smaller than the RD cell. Dark frame correction was performed for the individual images

(see SI Note S2). Buffer was exchanged after 23, 49 and 75 hours.

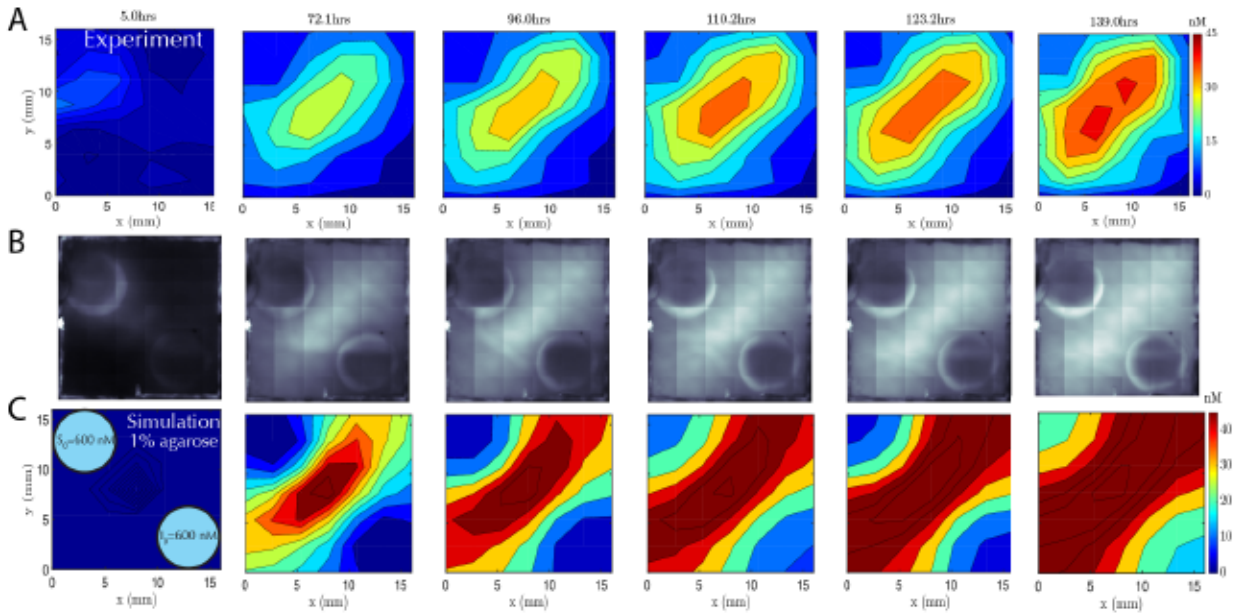

Figure 12.8. A medium (16 x 16 x 8 mm), growing two-dimensional gradient. (A) Filled contour plots depicting the experimental values of Output concentration profile in from 5 hours (left) to 139 hours (right). The gradient continues to grow for at least 139 hours, when the experiment was terminated. Buffer exchange occurred in the experiment after 22, 48, 71, 93 and 116 hours. (B) Corresponding fluorescent micrographs, which are obtained in a raster fashion and stitched together as a mosaic, as the field of view of the 4X objective used is smaller than the RD cell. Dark frame correction was performed for the individual images (see SI Note S2). (C) Filled contour plots depicting the simulated values of Output concentration profile from 5 to 139 hours. Leftmost plot in (C) depicts initial and boundary conditions of species. Reporter concentration is $R_0 = 200$ nM in both liquid reservoirs and in the agarose gel and is not depicted for clarity. See Figure 3 in main text for RD cell cartoon diagram.

12.5 | Reaction-Diffusion Device Fabrication

To fabricate the PDMS molds we used Sylgard 184 Silicone Elastomer Kit (Dow Corning) and mixed 10:1 of polymer:curing agent. We epoxied 8x8x8 mm wooden cubes (Amazon.com) to the bottom of a weigh boat to create the negative for the mold. Each calibration well was comprised of a single wooden cube and a reaction well was comprised of a linear chain of three cubes so that the 1D RD cell measured 24x8x8 mm (LxWxH) and the calibration wells $(8\ mm)^3$. Two-dimensional molds were made from 4 or 9 cubes (dimensions of 16x16x8 or 24x24x8 mm, respectively) and the diameter of the cylindrical wells was 7 mm. The polymer and curing agent were well-mixed, poured into weigh boats and then placed in the desiccator for 90 minutes to eliminate entrapped air bubbles. The device was then cured for 2 hours at 65 °C. The PDMS mold was then extracted from the weigh boat. Dust particles were removed from the PDMS mold using Scotch Tape and the glass slide (48x65 mm, Ted Pella) was cleaned using 70% EtOH and dried with N_2. To attach the glass slide to the PDMS, the coverslip and PDMS were treated for ~45 seconds each using a corona surface treater (BD-20, Electro-Technic Products). The device was set at 1.5 hours at 85°C to help promote bonding of the glass to the PDMS.

12.6 | DNA Sequences

12.6.1 | Diffusion Measurement Experiment

ssDNA used in Diffusion Experiment (shown Figure 2A):

Rb28f: /5TEX615/GTATTGTTGAATTGTAGAGTATT

dsDNA used in Diffusion Experiment (shown Figure 2B):

Rb28f: /5TEX615/GTATTGTTGAATTGTAGAGTATT

Rb28f_full_comp: AATACTCTACAATTCAACAATAC

12.6.2 | System (1-2)

Reversible Reporter 5

Rv5q: CCACCAAACTTCATCTCA/3IABkFQ/

Rb5f: /56-FAM/TGAGATGAAGTTTGGTGGTGAGA

Source 6_5

W6_5: CATAACACAATCACATCTCACCACCAAACTTCA

Gb6(5bp): TGAGATGTGATTGTGTTATGAGATG

Initiator 6

W_6_: CATCTCATAACACAATCACATCTCA

Recapturer 5

Dv5: CACCACCAATCTTCACT

Db5: AGTGAAGTTTGGTGGTGAGATGTTTTTACATCT

(base pair mismatch)

12.6.3 | System (3-4)

Reversible Reporter 28

Rv28q: TCTACAATTCAACAATAC/3IAbRQSp/

Rb28f: /5TEX615/GTATTGTTGAATTGTAGAGTATT

Source 27_28

W27_28: ACAACACTCTATTACAATACTCTACAATTCAAC

Gb6(5bp): TGAGATGTGATTGTGTTATGAGATG

Initiator 27

W_27_: ACAATACAACACTCTATTACAATAC

Recapturer 28

Dv28: ACTCTACAAATCAACAG

Db28: CTGTTGAATTGTAGAGTATTGTATTTTTACAAT

(base pair mismatch)

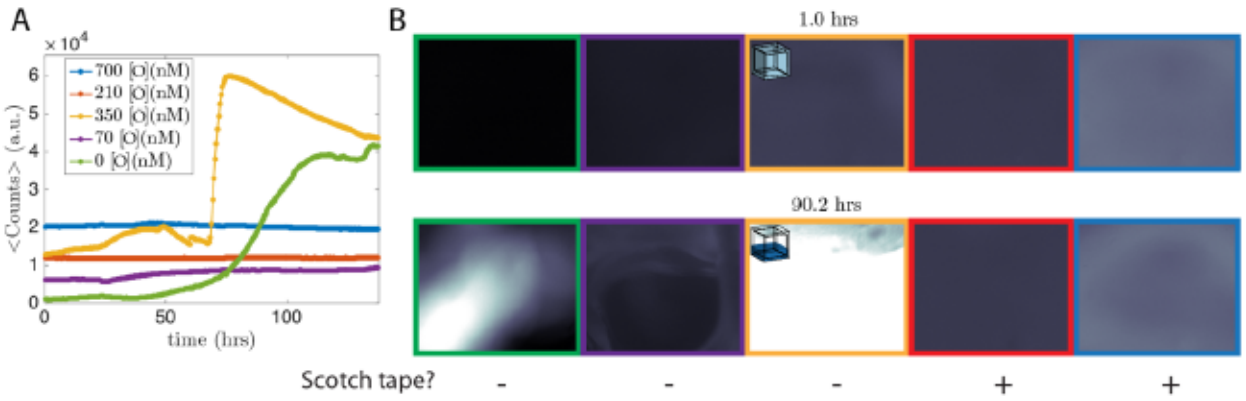

Figure 12.9. Preventing Evaporation. One significant factor in accurately converting fluorescence values to DNA concentrations was ensuring that material did not evaporate during time lapse imaging. Specifically, when covered only with a glass coverslip, almost all of the moisture in the calibration wells would evaporate in ~24-100 hours, leaving the agarose hydrogel at 1-5% of its original volume. This would, in turn increase the fluorescence of the sample as the evaporation process concentrates the DNA species, seen in (A) mean fluorescence vs time of the calibration wells (yellow, green and purple lines) and in the fluorescent images in (B). However, the two calibration wells sealed with Scotch tape did not significantly change fluorescence over the course of 130 hours (red and blue lines). Thus, we sealed all calibration and reaction channels with Scotch tape to mitigate evaporation. Border color in optical images (right side) corresponds with line color in plot. Optical images are shown after 1 hour (above) and 90.2 hours (below). Inset cartoon cube diagrams show (above) a calibration well full of agarose solution with DNA and (below) a well after some evaporation resulting in a solution that is more concentrated in DNA species.

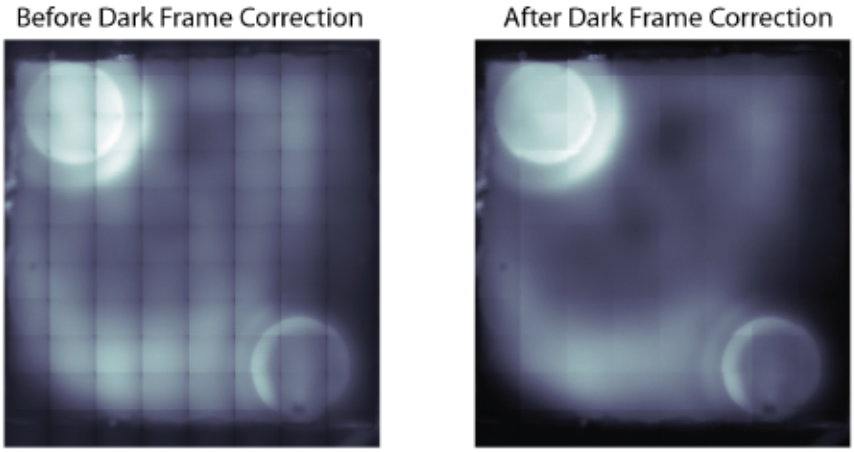

Figure 12.10. Dark frame correction to reduce edge effects in image montages. For the sake of continuity in fluorescence images, we eliminated the majority of the edge effects (typically a darker ring around the outer pixels) by using standard dark frame correction algorithms. While still not perfect, the dark frame correction eliminates the majority of the imaging artifacts. A more sophisticated algorithm like flat fielding would likely eliminate more edge effects, however, it seemed infeasible to obtain the necessary uniformly lit images for every image in the set. Immediately after image acquisition, we use a binning algorithm to compress the images, as a typical experiment captures ~20,000 images and an image captured on the 16-bit Infinity 3 CCD camera has 2752 x 2192 pixels (which would amount to ~240 Gb of data per experiment). The binning algorithm takes the mean intensity value of the nearest 4x4 pixels and stores the resulting value as a new pixel (resulting in a 16X compression).

References

[1] D. Y. Zhang and E. Winfree, *J Am Chem Soc*, 2009, **131**, 17303-17314.
[2] C. Thachuk, E. Winfree and D. Soloveichik, in *DNA Computing and Molecular Programming: 21st International Conference, DNA 21, Boston and Cambridge, MA, USA, August 17-21, 2015. Proceedings*, eds. A. Phillips and P. Yin, Springer International Publishing, Cham, 2015, DOI: 10.1007/978-3-319-21999-8_9, pp. 133-153.
[3] J. Fern, D. Scalise, A. Cangialosi, D. Howie, L. Potters and R. Schulman, *ACS Synthetic Biology*, 2016, DOI: 10.1021/acssynbio.6b00170.

13 | DNA Strand-Displacement Timer Circuits

Summary. Chemical circuits can coordinate elaborate sequences of events in cells and tissues, from the self-assembly of biological complexes to the sequence of embryonic development. However, autonomously directing the timing of events in synthetic systems using chemical signals remains challenging. Here we demonstrate that a simple synthetic DNA strand-displacement circuit can release target sequences of DNA into solution at a constant rate after a tunable delay that can range from hours to days. The rates of DNA release can be tuned to the order of 1-100 nM per day. Multiple timer circuits can release different DNA strands at different rates and times in the same solution. This circuit can thus facilitate precise coordination of chemical events *in vitro* without external stimulation.

13.1 | Introduction

While gene networks in cells can orchestrate intricate processes by modulating gene expression to release a series of target molecules at specified times,[1,2] synthetic *in vitro* biochemical protocols commonly involve manual steps performed by an experimenter, in which reagents are added, filtered, heated or otherwise altered. Artificial mechanisms to automate the temporal release of trigger molecules would make it possible to direct sequential events without the need for external stimulation. Further, the timed release of molecules using such a process could act as a trigger to control self-assembly processes,[3-5] multistep reaction cascades,[6] or to time the release of signaling molecules or other reagents in cell culture.[7-9]

In this paper, we build a chemical *timer circuit* that releases a target sequence of DNA at a constant rate from DNA complexes after a tunable delay period. In contrast to previously designed synthetic *in vitro* transcriptional timer circuits,[10] our timer is controlled solely by DNA strand-

displacement processes, which have previously been used to perform diverse information processing tasks including Boolean logic,[11-15] signal amplification,[16-19] neural network computation[20] and oscillatory signal generation.[21] The timer circuit is designed such that the strand that is released can be coupled to many of these systems in their present form, suggesting that timer circuits will make it possible to activate elaborate information processing tasks at specified times. Further, a design based on strand displacement reactions alone should allow the circuit to operate in a variety of buffers and at a variety of temperatures without redesign.[17]

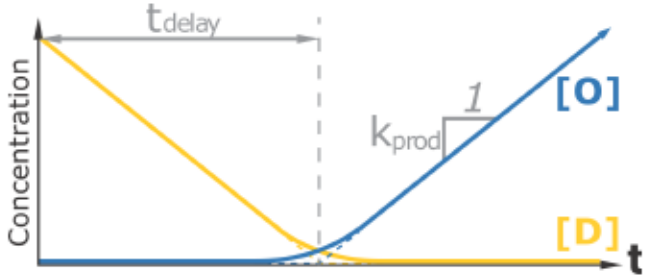

Figure 13.1. Schematic for the operation of a timer circuit. The output species (blue) is constantly produced at rate k_{prod}, but is rapidly consumed by the delay species (yellow). This rapid consumption prevents the accumulation of output until time t_{delay}, when the delay species is depleted.

The timer circuit works by suppressing the release of a single-stranded DNA molecule for a delay period, after which the molecule is allowed to increase in concentration at a constant rate. We show how to design a timer circuit within an abstract chemical reaction network, and then describe an implementation of the abstract network using a simple set of DNA molecules that interact through strand-displacement reactions. Next, we investigate the range of delay periods and release rates that are possible using our circuit, and finally demonstrate that multiple timer circuits can operate within the same solution.

The timer circuit consists of two simultaneous abstract chemical processes: *production* (Eqn. 1) and *delay* (Eqn. 2)

$$\emptyset_{Source} \xrightarrow{k_{prod}} \mathbf{O} \qquad (1)$$

$$\mathbf{O} + \mathbf{D} \xrightarrow{k_{delay}} \emptyset_{Waste} \qquad (2)$$

In the production process, the output **O** is released by a zero-order reaction at a constant rate k_{prod}. In the delay reaction, **O** is rapidly converted into inert waste when it reacts with a delay species **D**. If $k_{delay}[\mathbf{O}][\mathbf{D}] >> k_{prod}$, **O** cannot accumulate until all of **D** has been depleted (Fig. 1).

We call the time during which **O** cannot accumulate the delay time, t_{delay}, which is the time needed to produce enough **O** to consume all of the **D** that is initially present:

$$t_{delay} = \frac{[\mathbf{D}]_0}{k_{prod}} \qquad (3)$$

This time can be easily tuned by changing the initial concentration of **D**.

During the delay period, [**O**] remains very small because any molecules of **O** that are produced are rapidly removed. After **D** is depleted, however, [**O**] increases linearly with time. The approximate concentration of free **O** is therefore

$$[\mathbf{O}] \approx \begin{cases} k_{prod}/(k_{delay}[\mathbf{D}]) \approx 0, & \text{if } t < t_{delay} \\ k_{prod}(t - t_{delay}), & \text{otherwise} \end{cases} \qquad (4)$$

where [**O**] and [**D**] are functions of time.

To construct a timer circuit that controls the release of a DNA strand, we built a set of DNA strand-displacement reactions that emulate the abstract reactions in Equations 1 and 2 (Fig. 2).[22,23] The domain level structure of our complexes follows a DNA architecture previously used for Boolean logic circuit evaluation.[15] Within this implementation, strand **O** is initially partially bound within a complex. The production process frees **O** from this complex, making **O** available in its full single-stranded form (Fig. 2a). The delay process likewise sequesters **O** in a waste complex in which the toehold domain of **O** is covered (Fig. 2b). Because an exposed toehold domain is generally required to initiate downstream strand-displacement reactions, the delay circuit will control when **O**

is available in a functional form.

13.2 | Production

The production reaction releases an output molecule **O** when a source complex **S** and an initiator strand **I** react (Fig. 2a). This strand-displacement process is facilitated by the spontaneous pairing and un-pairing of the bases on the ends of the source complex, *i.e.* fraying. During the short time periods when these bases are frayed, **I** can bind and compete with **O** until one or the other is displaced.

Reactions initiated by fraying alone have an exceptionally small reaction rate constant, on the order of 0.5 $M^{-1}s^{-1}$, which we denote by k_{obp}.[24,25] On the order of days, very little **S** and **I** react, allowing us to assume [**S**] and [**I**] remain effectively constant when considering shorter time scales. To release an appreciable concentration of **O**, we use a large amount of **S** and **I** in a reaction. The rate at which **O** is released into solution can therefore be approximated as a constant we term k_{prod}:

$$\frac{d[\mathbf{O}]}{dt} = k_{obp}[\mathbf{S}][\mathbf{I}] \approx k_{prod} \qquad (5)$$

Equation 5 shows that k_{prod} can be tuned by changing the initial concentrations of **S** and **I**. For simplicity, and to maximize the time during which the approximation of constant concentrations is reasonable, we keep their initial concentrations equal, *i.e.* $[\mathbf{S}]_0=[\mathbf{I}]_0$. At longer time scales, the approximation of a constant k_{prod} is violated and [**O**] increases according to 2nd order reaction kinetics.

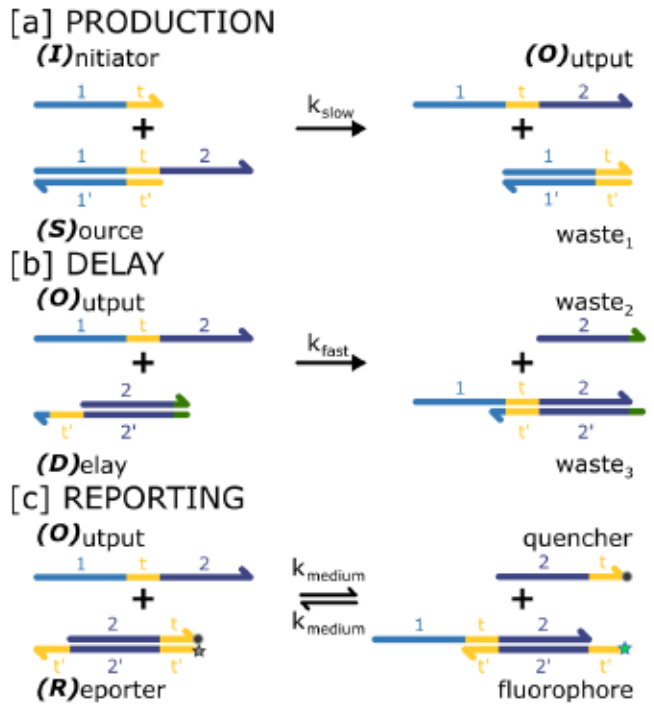

Figure 13.2. Strand-displacement reactions for a timer circuit. *(a) PRODUCTION:* Output is slowly released from source in the presence of initiator. *(b) DELAY:* Output is rapidly consumed by the delay complex. *(c) REPORTING:* Free output binds reversibly to a reporter complex, separating quencher and fluorophore modifiers. FAM and TexasRed paired with appropriate quenchers were used to report on two different output sequences (SI 0).

13.3 | Delay

To keep [O] low while the delay species is present, the delay reaction must sequester O at a rate much faster than k_{prod}. The delay complex D (Fig. 2b), which has a 7 base pair (7bp) single-stranded toehold domain that binds to O and co-localizes it with D, acts as a concentration thresholding device.[14] Reactions mediated by 7bp toeholds proceed at approximately six orders of magnitude faster than reactions without a mediating toehold.[24,25] A 2bp clamp (green in Fig. 2b) inhibits some undesired interactions between S and D while ensuring that the reaction between O and D remains strongly forward-biased. Clamps with only 1bp may not reliably prevent interactions at that end and

clamps with greater numbers of bases (*e.g.* 5bp) are expected to strongly decrease the sequestering ability of the thresholding device due to reaction reversibility.[11,17]

To monitor the reaction's progress, we also include a reporter complex modified with a fluorophore and an associated quencher to track the concentration of free O over time. This complex reacts reversibly with the output strand on a time scale much faster than the production reaction, but slower than the delay process, and produces fluorescence as a function of [O] at a given time (Fig. 2c). The concentration of **O** is related to the fluorescence levels using a calibration curve (SI 1).[26,27] To build a timer circuit, the source complex **S** and the delay complex **D** are initially combined and the timer is triggered upon the addition of the initiator strand **I**.

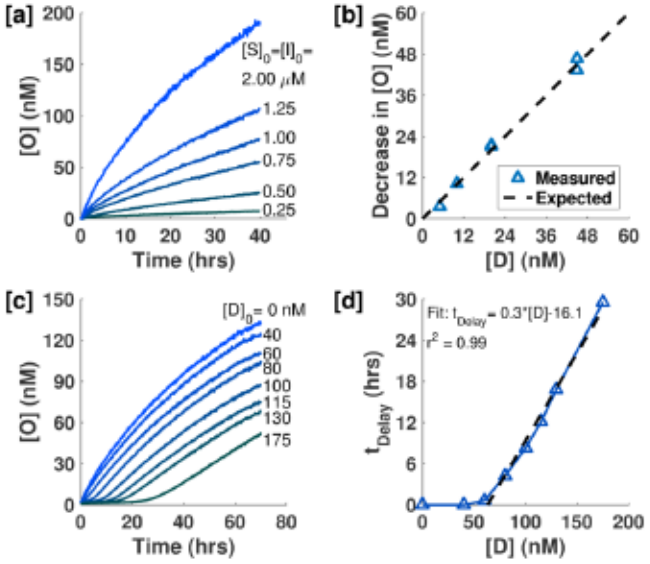

Figure 13.3. Production, Delay and Timer circuit reactions. *(a)* Production kinetics in the absence of D. Release rate is dependent upon initial [S] and [I]. *(b)* [O] decreases proportionally to added [D]. *(c)* Delay time was tuned by changing the initial concentration of the delay species D. $[S]_0=[I]_0=1$ μM. *(d)* Plot of delay time *vs.* initial concentration of D, showing an approximately linear relationship after an initial offset.

To demonstrate that the individual reactions performed as desired and determine how the production rate varied with initial concentrations of S and I, we tested each reaction in isolation at

25°C (Fig. 3a-b, SI 2). To test the production reaction, we varied $[S]_0=[I]_0$ from 0.25 μM to 2 μM (Fig. 3a) and determined the average k_{olp} to be 0.49±0.13 $M^{-1}s^{-1}$ (Supp. Table 2) which is in good agreement with previous estimates.[24,25] We calculated, using $[S]_0$ and $[I]_0$, that k_{prod} varied from 0.15 to 4 nM/hour over the range of concentrations tested (Supp. Table 2). When the delay reaction was tested in isolation, the delay complex sequestered free O, resulting in a sudden decrease in output detected by the reporter (Supp. Fig. 7). The decrease in [O] matched the concentration of D added to the reaction solution (Fig. 3b).

To characterize the delay time before S begins accumulating as a function of $[D]_0$ (Eqns. 3 and 4), we varied $[D]_0$ while keeping the production rate (determined by $[S]_0$ and $[I]_0$) constant. O remained low for a delay period that increased with $[D]_0$ (Fig. 3c). For each trial, we used linear least squares fitting to identify the portion of the production regime with the steepest slope (SI 3) and used the slope and y-intercept of this fit to calculate the delay time. This method allowed us to measure the delay time without being affected by the ramp up in release that occurs because small amounts of D are still present when release noticeably begins. We observed that the delay time varied linearly with respect to the initial concentration of D (Fig. 3d). We also tested two other timer circuits with lower initial concentrations of S, I and D, and obtained systems with similar delay times but slower rates of output release (SI 3).

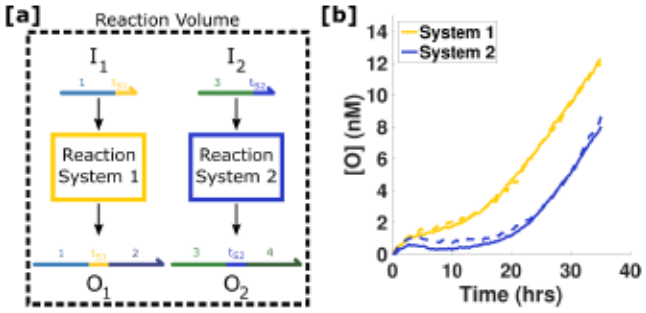

Figure 13.4. Multiplexed timers. *(a)* Schematic of multiple timer circuits operating within the same solution, releasing independent output strands at different times. *(b)* Two timers release output with the same rate but

at different times (9 and 17 hours). Here, $[S]_0=[I]_0=0.5$ μM and $\{[D]_{Syn1}, [D]_{Syn2}\}=\{46$ nM, 63 nM$\}$. Dashed lines indicate the same reactions except with each system in isolation.

13.4 | Multiplexing

Because the delay circuit is based on DNA strand-displacement events involving a particular DNA sequence, it is possible to create multiple circuits that use different sequences and can trigger the delayed release of two different DNA strands (Fig. 4). To characterize the operation of two timer circuits in a single solution, we prepared a second timer circuit and reporter complex with different sequences and fluorophore/quencher than our original system. The second system had the same qualitative behavior, and it was possible to programmatically tune both delay times and release rates. Differences in rates and delay times were observed between the two systems, possibly due to differences in toehold sequence that affect reaction rates (SI 4-5). The two systems were able to operate together in the same solution with virtually identical kinetics to those observed when the systems were operated apart (Fig. 4).

13.5 | Discussion

The timer circuit developed here successfully releases target strands of DNA into solution at a constant rate after a delay period. The sequence, delay time and production rate were easily tuned without needing to redesign the release system. We demonstrated delay times on the order of hours to days and production rates from a few nM/day to a hundred nM/day, which for volumes of 100 μL are approximately 0.1-10 nmol/day.

While the release of O observed (Fig. 3, SI Figures 8-10) qualitatively followed our simple model described in Equations 1-5, there were differences between experiments and reaction curves predicted by the model. Notably we found that the release rate of O decreased faster and to a greater degree than the predicted reaction curves, suggesting the existence of uncharacterized reactions (*i.e.*

leak reactions) between S and D or the reporter. Based on the experiments, we hypothesize that DNA synthesis errors (*e.g.* base-mismatches, truncations, additions or deletions) in the bottom strand of the S complex led to leak reactions and pathways that explained most of the deviation from the simple model. By accounting for these and other more minor leak reactions, a model was developed that provided a close fit with the experimental observations (SI 6), indicating that models can be used to program the rate and timing of output release.

By designing a circuit in which the output DNA sequence is an aptamer[3,28] – *i.e.* sequences of DNA that bind specifically to non-DNA species such as proteins[29] and other small molecules,[30] the circuit described here could also be used to control the dynamics of a wide range of other chemical systems beyond DNA strand-displacement.

Timer devices that can be programmed to release a particular species with a pre-specified delay could be important for designing cascades for therapeutics or for self-assembly[5,32] in which different species are activated at different times. The constant low-rate of production we have shown could be used to design therapeutic hydrogels with novel, linear release profiles or within a reliable pulsed delivery system by combining it with a threshold amplifier system such as those used for signal restoration in molecular logic circuits.[14,31]

Acknowledgments

The authors would like to thank Elisa Franco, Deepak Agrawal, Abdul Mohammed and John Zenk for insightful conversations. This work was supported by NSF-CCF-1161941, NSF-SHF-1527377, award SC0010595 from the Department of Energy for some materials and supply costs, and a grant to the Turing Centenary Project by the John Templeton Foundation.

References

(1) Legewie, S, Dienst, D, Wilde, A, Herzel, H, and Axmann, IM. (2008) Small RNAs establish delays and temporal thresholds in gene expression. *Biophys. J.*, *95*, 3232-3238.
(2) Zaslaver, A, Mayo, AE, Rosenberg, R, and Bashkin, P. (2004) Just-in-time transcription program in metabolic pathways. *Nat. Genet.*, *36*, 486-491.
(3) Dirks, R. M.; Pierce, N. A. (2004) Triggered amplification by hybridization chain reaction. *Proc. Natl. Acad. Sci. U.S.A.*, *101*(43), 15275-15278.
(4) Yin, P.; Choi, H. M.; Calvert, C. R.; Pierce, N. A. (2008) Programming biomolecular self-assembly pathways. *Nature*, *451*(7176), 318-322.
(5) Zhang, D. Y.; Hariadi, R. F.; Choi, H. M.; Winfree, E. (2013) Integrating DNA strand-displacement circuitry with DNA tile self-assembly. *Nat. Commun.*, *4*.
(6) Fu, J.; Yang, Y. R.; Johnson-Buck, A.; Liu, M.; Liu, Y.; Walter, N. G.; Woodbury, N. W.; Yan, H. (2014) Multi-enzyme complexes on DNA scaffolds capable of substrate channeling with an artificial swinging arm. *Nat. Nanotechnol.*, *9.7*, 531-536.
(7) Caldorera-Moore, M.; Peppas, N. A. (2009) Micro-and nanotechnologies for intelligent and responsive biomaterial based medical systems. *Adv. Drug Delivery Rev.*, *61.15*, 1391-1401.
(8) Santini Jr, J. T.; Richards, A. C.; Scheidt, R.; Cima, M. J.; Langer, R. (2000) Microchips as controlled drug-delivery devices. *Angew. Chem., Int. Ed.*, *39*(14), 2396-2407.
(9) Staples, M.; Daniel, K.; Cima, M. J.; Langer, R. (2006) Application of micro- and nanoelectromechanical devices to drug delivery. *Pharm. Res.*, *23*(5), 847-863.
(10) Kapsner, K., and Simmel, F. C. (2015) Partitioning Variability of a Compartmentalized In Vitro Transcriptional Thresholding Circuit. *ACS Synth. Biol.*, *4*, 1136–43.
(11) Seelig, G.; Soloveichik, D.; Zhang, D. Y.; Winfree, E. (2006) Enzyme-free nucleic acid logic circuits. *Science*, *314.5805*, 1585-1588.
(12) Takahashi, K.; Yaegashi, S.; Kameda, A.; Hagiya, M. (2006) Chain reaction systems based on loop dissociation of DNA. *DNA computing*. 347-358.
(13) Zhang, C.; Yang, J.; Xu, J. (2009) Circular DNA logic gates with strand displacement. *Langmuir*, *26.3*, 1416-1419.
(14) Qian, L.; Winfree, E. (2011) Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, *332.6034*, 1196-1201.
(15) Genot, A. J.; Bath, J.; Turberfield, A. J. (2011) Reversible logic circuits made of DNA. *J. Am. Chem. Soc.*, *133.50*, 20080-20083.
(16) Zhang, D. Y.; Seelig, G. (2011) DNA-based fixed gain amplifiers and linear classifier circuits. *DNA Computing and Molecular Programming*, 176-186.
(17) Zhang, D. Y.; Turberfield, A. J.; Yurke, B.; Winfree, E. (2007) Engineering entropy-driven reactions and networks catalyzed by DNA. *Science*, *318*(5853), 1121-1125.
(18) Chen, X.; Briggs, N.; McLain, J. R.; Ellington, A. D. (2013) Stacking nonenzymatic circuits for high signal gain. *Proc. Natl. Acad. Sci. U.S.A.*, *110*(14), 5386-5391.
(19) Seelig, G.; Yurke, B.; Winfree, E. (2006) Catalyzed relaxation of a metastable fuel. *J. Am. Chem. Soc.*, *128.37*, 12211-12220.
(20) Qian, L.; Winfree, E.; Bruck, J. (2011) Neural network computation with DNA strand displacement cascades. *Nature*, *475.7356*, 368-372.
(21) Srinivas, N. (2015) Programming chemical kinetics: engineering dynamic reaction networks with DNA strand displacement. *Diss. California Institute of Technology*.
(22) Soloveichik, D.; Seelig, G.; Winfree, E. (2010) DNA as a universal substrate for chemical kinetics. *Proc. Natl. Acad. Sci. U.S.A.*, *107.12*, 5393-5398.
(23) 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.
(24) Zhang, D. Y.; Winfree, E. (2009) Control of DNA strand displacement kinetics using toehold exchange. *J. Am. Chem. Soc.*, *131.47*, 17303-17314.
(25) Srinivas, N.; Ouldridge, T. E.; Šulc, P.; Schaeffer, J. M.; Yurke, B.; Louis, A. A.; Doye, J. P. K.; Winfree, E. (2013) On the biophysics and kinetics of toehold-mediated DNA strand displacement. *Nucleic Acids Res.*, *41.22*, 10641-10658.
(26) Qian, L.; Winfree, E. (2014) Parallel and scalable computation and spatial dynamics with DNA-based chemical reaction networks on a surface. *DNA Computing and Molecular Programming*, 114-131.
(27) Wang, J. S.; Zhang, D, Y. (2015) Simulation-guided DNA probe design for consistently ultraspecific hybridization. *Nat. Chem.*
(28) Zhang, Z.; Chen, N.; Li, S.; Battig, M. R.; Wang, Y. (2012) Programmable hydrogels for controlled cell catch and release using hybridized aptamers and complementary sequences. *J. Am. Chem. Soc.*, *134.38*, 15716-15719.

(29) Gatto, B.; Palumbo, M.; Sissi, C. (2009) Nucleic acid aptamers based on the G-quadruplex structure: therapeutic and diagnostic potential. *Curr. Med. Chem.*, *16.10*, 1248-1265.
(30) Huizenga, D. E.; Szostak, J. W. (1995) A DNA aptamer that binds adenosine and ATP. *Biochemistry*, *34.2*, 656-665.
(31) Scalise, D.; Schulman, R. (2014) Designing modular reaction-diffusion programs for complex pattern formation. *Technology*, *2.01*, 55-66.
(32) Yao, D., Song, T., Sun, X., Xiao, S., Huang, F., and Liang, H. (2015) Integrating DNA-Strand-Displacement Circuitry with Self-Assembly of Spherical Nucleic Acids. *J. Am. Chem. Soc.*, *137*.

14 | Supplemental Information: DNA Strand-Displacement Timer Circuits

14.1 | Materials and Methods

The timer circuit was designed following the principles outlined in Supp. Reference 1. Sequences for each domain were drawn from Table S1 of the Supporting Online Material for Supp. Reference 1 and are listed here in Table 1. Domains *S6* and *S5* listed below correspond to Domains *1* and *2* depicted in Figure 2 of the Main Text and Supp. Figure 0. The toehold (*t*) domain of System 2 was designed to have minimal non-specific interactions with the sequences of both systems using NUPACK.[2] Domain Names and Sequences are listed 5' to 3'. Additional schematics of the DNA complexes and the reactions are shown in SI 2.

Table 1: Sequence Data

Strand Names	**Domain Names**	**Sequences**
Source 1 Top	*S6 t S5*	CA TAACACAATCA CA TCT CA CCACCAAACTT CA
Source 1 Bottom	*t' S6'*	TG AGA TG TGATTGTGTTA TG
Initiator 1	*S6 t*	CA TAACACAATCA CA TCT CA
Delay 1 Top	*S5*	CA CCACCAAACTT CA CT
Delay 1 Bottom	*S5' t' S6' (2nt)*	AG TG AAGTTTGGTGG TG AGA TG TG
Reporter 1 Top	*S5 t Quencher*	CA CCACCAAACTT CA TCT CA/3IABkFQ/
Reporter 1 Bottom	*FAM t' S5' t'*	/56-FAM/TG AGA TG AAGTTTGGTGG TG

		AGA TG
Reporter 1 Full Complement	*t S5 t*	CA TCT CA CCACCAAACTT CA TCT CA
Source 2 Top	*S27 t S28*	AC AACACTCTATT AC AAT AC TCTACAATTCA AC
Source 2 Bottom	*t' S27'*	GT ATT GT AATAGAGTGTT GT
Initiator 2	*S27 t*	AC AACACTCTATT AC AAT AC
Delay 2 Top	*S28*	AC TCTACAATTCA AC CA
Delay 2 Bottom	*S28' t' S27' (2nt)*	TG GT TGAATTGTAGA GT ATT GT AA
Reporter 2 Top	*S28 t Quencher*	AC TCTACAATTCA AC AAT AC/3IABRQSp/
Reporter 2 Bottom	*TexasRed t' S28' t'*	/5TexRd-XN/GT ATT GT TGAATTGTAGA GT ATT GT
Reporter 2 Full Complement	*t S28 t*	AC AAT AC TCTACAATTCA AC AAT AC

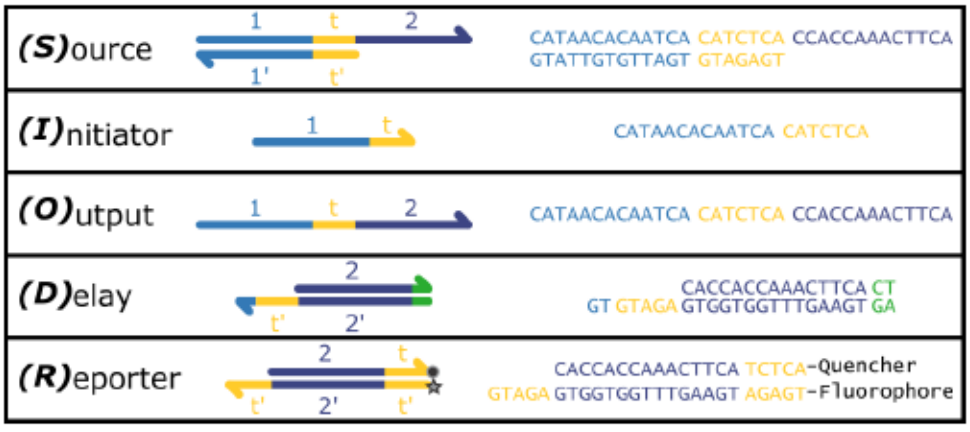

Figure 14.1. Schematic of reaction species with their sequences as listed in Table 1. Reactions between

species for the timer circuit are shown in Supp. Figures 3 and 6.

Sequences were ordered as lyophilized powder from Integrated DNA Technologies (IDT). The **Reporter Top** and **Reporter Bottom** strands were ordered purified by high-performance liquid chromatography (HPLC) and all other strands were ordered impure with standard desalting. Strands were suspended in Millipore purified water to a concentration of ~1 mM and stored at -20°C. Empirical oligonucleotide stock concentrations were determined by assaying the absorbance (OD260) of 1000x diluted samples of each stock solution at 260 nm. The extinction coefficient provided by IDT was used to calculate stock concentrations using the Beer-Lambert law.

Source, **Delay** and **Reporter** complexes were prepared at a concentration of 100 μM in Tris-acetate-EDTA buffer with 12.5 mM Mg^{++} (1x TAE/Mg^{++}). Each complex was annealed in an Eppendorf Mastercycler PCR by holding the solutions at 90°C for 5 minutes followed by cooling at -1°C per minute down to 20°C. After annealing, the **Source** complexes were incubated with 100 μM of their complementary **Initiator** strand overnight at room temperature to react with any poorly formed **Source** complexes. The **Source** complex was not incubated with any other strands or complexes due to the increased complexity of gel purification.

After annealing each complex and incubating the **Source** complexes overnight, all complexes were purified by polyacrylamide gel electrophoreses (PAGE). Ten percent polyacrylamide gels were cast by mixing 3.25 mL of 19:1 40% acrylamide/bis solution (Bio-Rad) with 1.3 mL 10x TAE/Mg^{++} and 8.45 mL Millipore-purified H_2O. This solution was polymerized by the addition of 78 μL 10% ammonium persulfate (APS) and 5.4 μL tetramethylenediamine (TEMED) in a gel cassette with a large single well comb at the top of each gel. Fifteen percent polyacrylamide gels were prepared in a similar fashion except with a corresponding higher fractional volume of 40% acrylamide/bis stock solution. Two hundred microliters of annealed DNA complexes were mixed with 6x loading dye (New England Biolabs, product #B7021S) and loaded

into the wells of the gels in a Scie Plas TV100K cooled vertical electrophoresis chamber. The gels were run at 150V and 4°C for 1.5 or 3 hours for 10% and 15% polyacrylamide gels. **Reporter** and **Delay** complexes were purified using 10% gels and the **Source** complex was purified using 15% gels. After running for the appropriate time, the bands were cut out using UV-shadowing at 254 nm for visualization. Bands were diced into ~1 mm^3 pieces, mixed with 500 μL of 1x TAE/Mg^{++} buffer and were shaken on a vortexer overnight at room temperature. The DNA solutions were then transferred by pipet to a fresh tube leaving behind the gel pieces. The solutions were centrifuged for 5 minutes at 3000x*g* to precipitate any remaining gel pieces. The DNA solutions were transferred to a new tube and stored at 4°C until use. The concentrations of these purified complexes were then measured with an Eppendorf Biophotometer with a dilution factor of 30x using the approximate extinction coefficient (ε):

$$\varepsilon_{\mathbf{Final}} = \varepsilon_{Top-strand} + \varepsilon_{Bottom-strand} - 3200N_{AT} - 2000N_{GC}$$

where N indicates the number of hybridized A-T or G-C pairs in each complex.[3]

Reaction kinetics were measured on quantitative PCR (qPCR) machines (Agilent Stratagene Mx3000 and Mx3005 series) at 25°C. Reactions were prepared in 96-well plates using 150 μL/well volumes. Each well contained 1x TAE/Mg^{++} and 2 μM PolyT_{20} strands to help displace reactant species from the pipet tips used to add species and potentially from the well walls. In a typical experiment, Millipore-purified H_2O, TAE/Mg^{++} and PolyT_{20} strands were first mixed together. **Reporter** complexes were then added at 100 nM for System 1 or 200 nM for System 2. Baseline fluorescent measurements of the **Reporter** complex alone was conducted for each experiment for 0.5 to 1 hour with measurements every 1 to 10 minutes. This baseline was taken to be where the **[Output]** is equal to zero as detected by the **Reporter** and was subtracted from all subsequent data. After measuring this baseline, DNA strands or complexes were added to each well, depending on

the experiment (see SI 1-4). Fluorescence measurements were taken every 1-5 minutes for Delay characterization or every 5-10 minutes for Production characterization and Timer experiments.

14.2 | Reporter Calibration

The **Reporter** complex (Figure 2c in Main Text) was used to indirectly measure the concentration of the single-stranded **Output** strands in solution as a function of time. A **Reporter** that reacts reversibly with the **Output** species was used as an irreversible reporter could compete with the **Delay** species. The reporter follows the reaction:

$$Reporter + Output \underset{k_r}{\overset{k_f}{\rightleftharpoons}} \text{Fluorophore} + \text{Quencher} \qquad \textbf{(SI-Eqn. 1)}$$

where the quencher-modified top strand of the Reporter complex is displaced by an invading strand causing in increase of fluorescence. The forward reaction rate constant, kf, is expected to be around 5x104 M-1s-1.4 Two calibrations were conducted to translate measured fluorescence intensities to levels of free Output concentration for each experiment. It was assumed that the measured fluorescence was proportional to the concentration of unquenched fluorophore, [Fluoro.] (e.g. Fluorophore in SI-Eqn. 1), through a proportionality constant α. To determine α, we measured the fluorescence of the Reporter complex with known concentrations of the full complement (FC) to the bottom strand of the Reporter (see Table 1) and measured the change in fluorescence before and after addition of the complementary strand (Supp. Figure 1). In general, we used the equation

$$[FC] = \alpha * \Delta\text{Fluorescence} + \beta \qquad \text{(SI-Eqn. 2)}$$

for **[FC]** equal to 0, 25, 50, 75 and 100 nM. In the ideal case, β is equal to zero. Alpha was determined by calculating the slope after fitting a line to **[FC]** vs. ΔFluorescence (Supp. Figure 1). This calibration enables the normalization of all fluorescence data into **[Fluoro.]**. Additionally, Supp.

Figure 1 shows that photobleaching of the fluorophores are not a significant factor in measuring fluorescence as seen by the stable, non-decreasing intensity values.

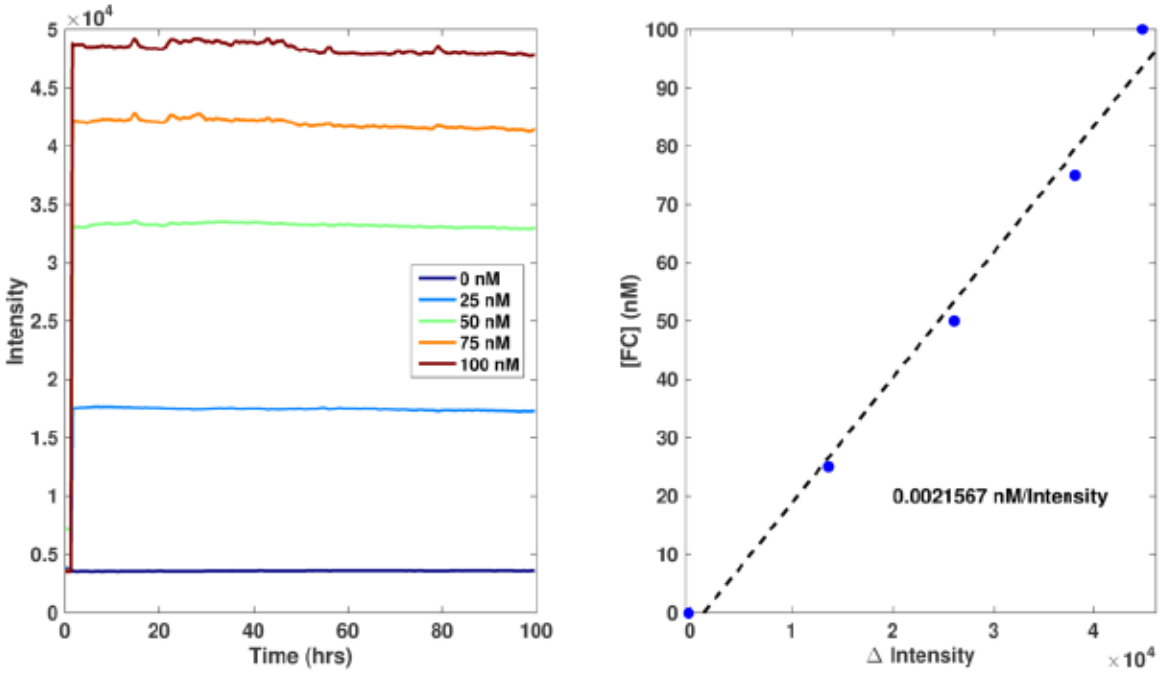

Figure 14.2. Example calibration plots for the **Reporter** complex with its full complement to convert raw intensity values into **[Fluoro.]**. **FC** was added to 100 nM **Reporter** at concentrations ranging from 0 – 100 nM as noted in the legend.

To convert the **[Fluoro.]** into **[Output]**, the K_{eq} for the reporter reaction shown in SI-Eqn. 1 was calculated by mixing the **Reporter** complex with known concentrations of **Output** strand and using the equation:

$$K_{eq} = \frac{[Fluoro.]\text{^}2}{([O]_0 - [Fluoro.])([R]_0 - [Fluoro.])} \qquad \text{(SI-Eqn. 3)}$$

where $[O]_o$ is the concentration of **Output** (*e.g.* 25, 50, 75 or 100 nM) added to the **Reporter** solution and $[R]_o$ is the initial **Reporter** complex concentration (*e.g.* 100 nM). In general, we found that the intensities measured with this calibration method decreased over the duration of the experiment (Supp. Fig. 2a), possibly due to **Reporter** complexes becoming stuck in the "off" state.

Due to this decrease, the K_{eq} was calculated as a function of time. The data was segmented into 75 bins (~1.3 hours each), with each bin having a K_{eq} calculated as the average K_{eq} over the **[Output]** tested (Supp. Fig. 2b,c).

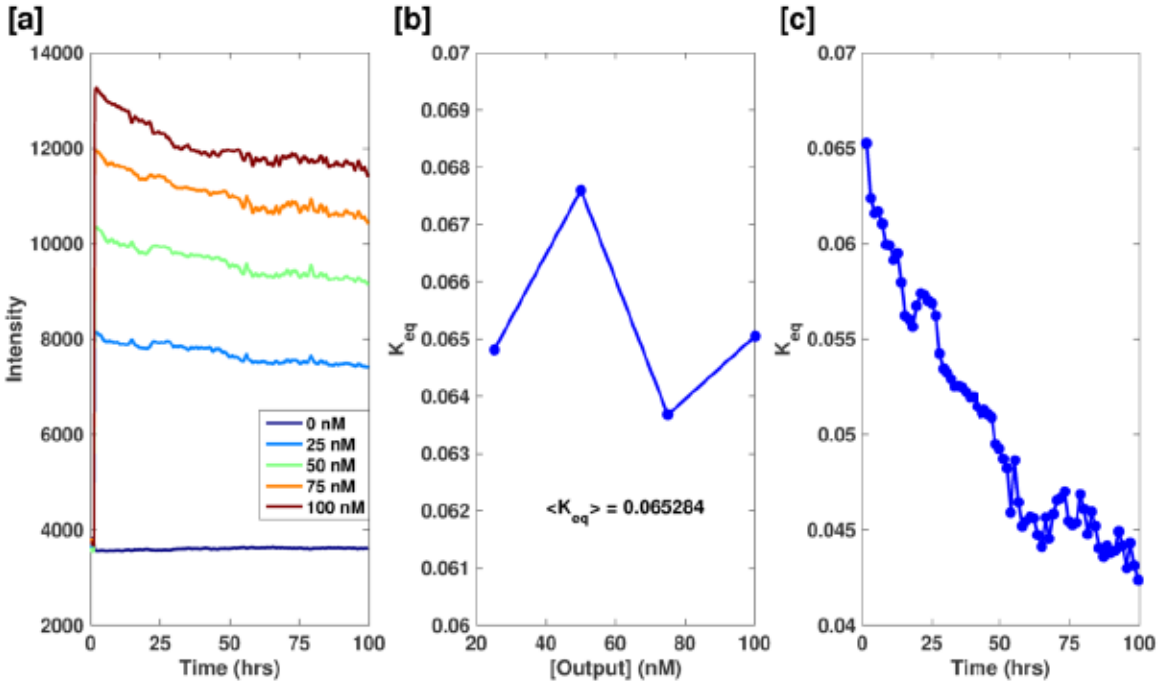

Figure 14.3. Example calibration plots for the **Reporter** complex with **Output** to convert **[Fluoro.]** values into **[Output]**. (a) **Output** was added to 100 nM **Reporter** at concentrations ranging from 0 – 100 nM as noted in the legend. (b) For each time segment, the K_{eq} was calculated as the average value over the **[Output]**'s tested. Little variance was seen between the K_{eq}'s calculated at each **[O]** within a given time segment. The initial segment is shown (t = 1 hr). (c) The K_{eq} decreases as a function of time. Each segment usually contained 1.3-1.5 hours of data.

The concentration of **Fluoro.** was then converted into **[Output]** through the equation

$$[O] = \frac{[Fluoro.]^2}{K_{eq}([R]_0 - [Fluoro.])} + [Fluoro.] \qquad \text{(SI-Eqn. 4)}$$

where *[Fluoro.]*, *[O]* and K_{eq} are functions of time. This equation reports the total concentration of **Output** as the sum of free **O** in solution and **O** that is transiently bound to the **Reporter** complex.

14.3 | Production and Delay Reaction Characterization

To understand the timer system, we initially characterized the production reaction (between the **Source** and **Initiator** molecules) without a concurrent delay reaction. Multiple production reactions using various concentrations of **Initiator** and **Source** were conducted to calculate the forward and reverse reaction rate constants for the set of reactions shown in Supp. Figure 3. Two sets of reactions were conducted: [S]=[I] and [S]x[I]=1. In the first case, the production rate (k_{prod}) varies because the product of the initial concentrations of **Initiator** and **Source** is changing. In the second case, the production rate is theoretically constant between experiments on "short" time scales.

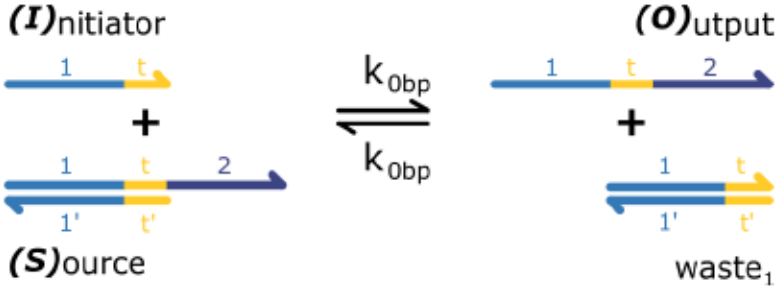

Figure 14.4. Schematic for the Production circuit. The **Initiator** reacts with the **Source** complex through a fraying mechanism at the ends of the double-stranded regions to produce **Output** and **waste$_1$**. The forward and reverse reaction rate constant depend on DNA sequence and the point of strand-displacement initialization and thus could be different values, but for simplicity a single rate constant, k_{0bp}, was chosen.

The **Source** and **Reporter** were incubated for about 6 hours until the measured intensity reached a steady state prior to the addition of **I**. This steady state intensity is thought to be a small population of free **Output** left over from the purification process, although interactions between the **Source** and **Reporter** complexes may exist. The concentration of detected **Output** by the **Reporter** corresponded to 0.01-0.02x[**Source**] used in each experiment (Supp. Figure 4).

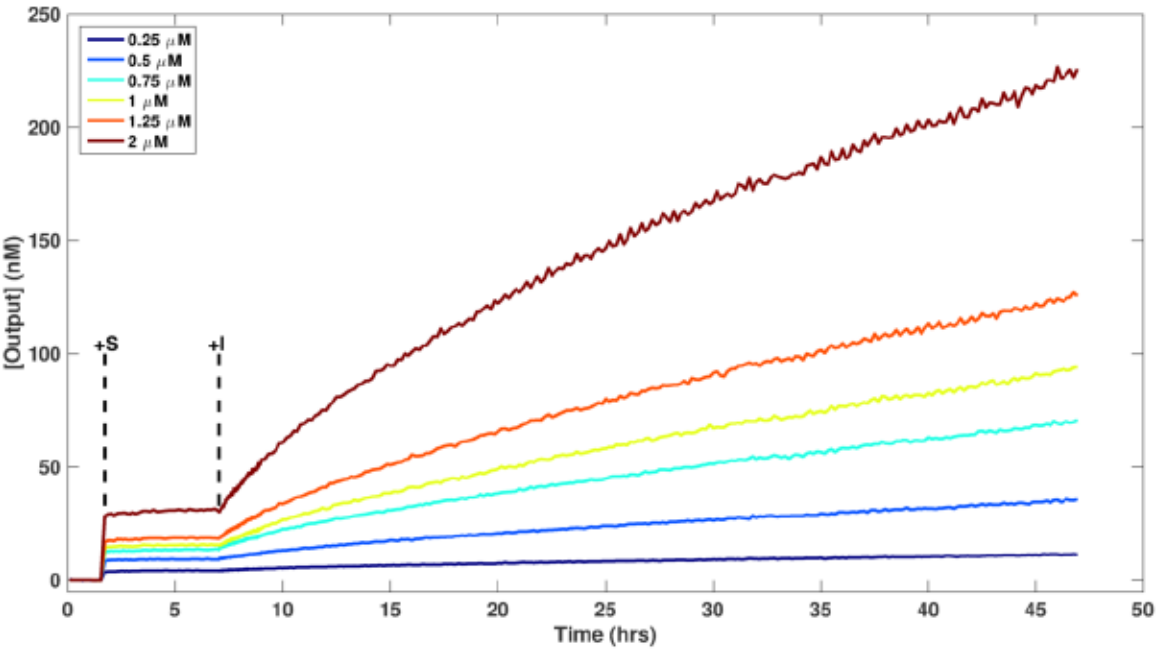

Figure 14.5. Characterization of the effect of [Source] and [Initiator] on the observed release of Output. The **Source** is mixed with the **Reporter** after 1.5 hours and incubated for about 6 hours, when the **Initiator** is added (denoted by black dashed lines). For the curves shown, [S] is equal to **[I]** and is shown in the legend. The concentration of initial **Output** detected prior to initiation scales with the amount of [S] added. Data is identical to that shown in Supp. Figure 5a and Main Text Figure 3a.

The post-initiation reaction curves were fit using MATLAB to initially calculate k_{0bp} for this set of reactions using the second-order reaction kinetics equation:

$$\frac{d[O]}{dt} = k_{0bp}[I][S] - k_{0bp}[O][waste_1] \qquad \text{(SI-Eqn. 5)}$$

The **Reporter** reaction was not included in the fitting procedure because its equilibration kinetics were assumed to be much faster (~10^5x faster reaction rate constant) than that of the Production reaction. However, the reaction rate constants calculated from this model did not capture the dynamics seen in Supp. Fig 5a-b - *i.e.* a quick release of **O** followed by a slower, more linear region. We hypothesized that this was due to a small concentration of **Source** that reacted quickly and irreversibly with the **Initiator** present in the reaction volumes, perhaps because of errors in

sequence produced during solid state DNA synthesis. We call this small population $[S]_{Leak}$ which reacts with reaction rate constant k_{Leak}. To account for this possibility, we used the following equation to generate a better fit to the experimental data:

$$\frac{d[O]}{dt} = k_{0bp}[I][S] - k_{0bp}[O][waste_1] + k_{Leak}[S]_{Leak}[I]$$

and to calculate the model parameters k_{0bp}, k_{Leak} and $[S]_{Leak}$. Additional information regarding S_{Leak} can be found in SI 6. An example of such a fit is shown in Supp. Figure 5c and the fit parameters are compiled in Table 2. The average k_{0bp} was 0.49±0.13 1/M-sec which is in good agreement with the reported value of 0.5 1/M-sec.[4] $[S]_{Leak,0}$ varied from 0.5 to 4.7% of $[S]_0$.

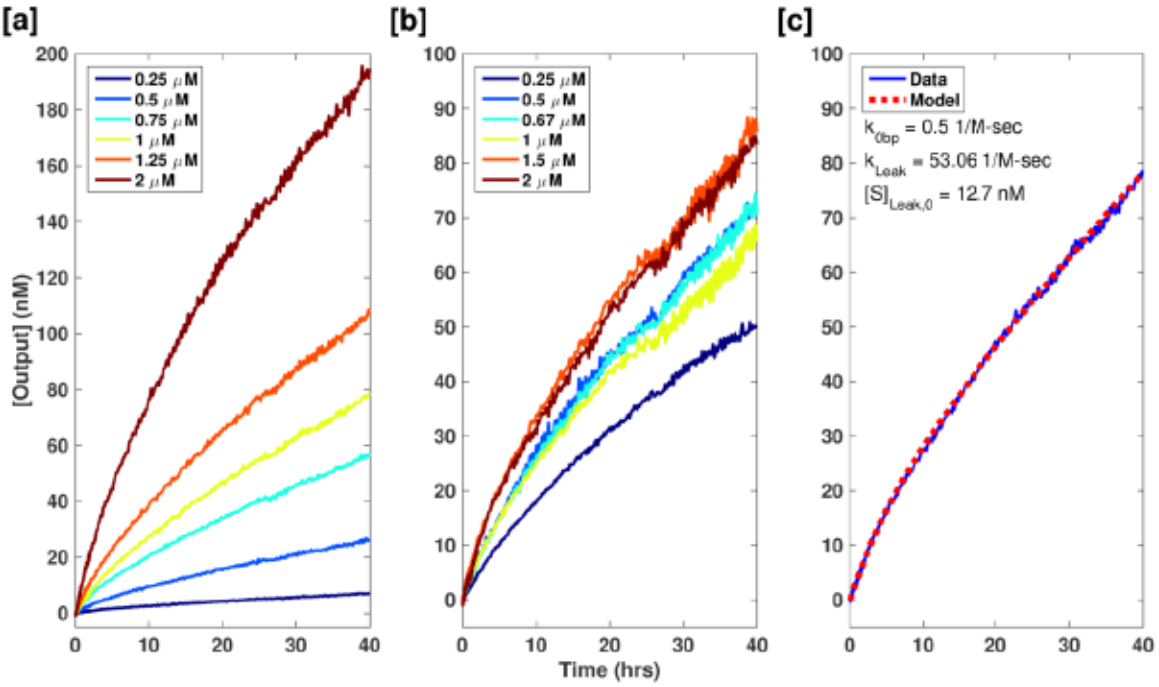

Figure 14.6. Characterization of Production reactions. Production reactions were measured for **[Source]** = **[Initiator]** in the range of 0.25-2 μM (a) and for **[S]** x **[I]** = 1 (b). The legend in (b) shows the concentration of **Source** in the reaction mixture. (c) Example comparison between experimental measurements and the model resulting from the fit of reaction rate constants for **[S]**=**[I]**=1 μM. Reaction curves were fit using a bimolecular reaction kinetics model to calculate a second-order reaction rate constant. Reaction rate constants

for each curve are listed below in Table 2.

As shown in Supp. Figure 5a, the production rate varied with the concentrations of **Source** and **Initiator**. From Equation 5 in the Main Text, the production rate is expected to follow a power law ($k_{Prod} \sim [S]_0^2$) when initial concentrations of **S** and **I** are equal. The experimentally determined production rate increased to the exponential of 1.6 with increasing **S** and **I** instead of 2. There was also a small variation in the production rate for the case [S]x[I]=1, where the initial k_{prod} was expected to be constant among the reaction conditions tested (see Table 2). The deviation from the expected result in both cases could be due to other undesired reactions present in between reaction species.

Table 2: Reaction rate constants for the Production reactions shown in Supp. Figure 5. The production rate constant (k_{prod}) was calculated using Equation 5 listed in the Main Text using the initial concentrations of **Source** and **Initiator**. Values listed are result of fit with 95% confidence interval bounds. Overall values are the average and standard deviation of all rows (k_{Obp}) or rows in the [S]x[I]=1 set of experiments (k_{prod}).

Experiment Type	[S] (μM)	[I] (μM)	k_{Obp} (1/M-sec)	k_{prod} (nM/hr)	k_{Leak} (1/M-sec)	$[S]_{Leak}$ (nM)
[S]=[I]	0.25	0.25	0.676±0.008	0.152±0.002	336±20	1.3±0.1
	0.5	0.5	0.645±0.006	0.58±0.005	129±7	4.6±0.2
	0.75	0.75	0.647±0.004	1.311±0.009	91±4	8.8±0.2
	1	1	0.501±0.004	1.803±0.014	53±2	12.7±0.4
	1.25	1.25	0.439±0.004	2.47±0.021	43±2	19.0±0.6
	2	2	0.279±0.005	4.016±0.066	20±1	53.8±2.0
[S]x[I]=1	0.25	4	0.32±0.01	1.14±0.02	8±1	11.7±0.8

	0.5	2	0.46±0.01	1.67±0.02	27±2	13.2±0.6
	0.67	1.5	0.46±0.01	1.65±0.02	35±3	12.6±0.6
	1	1	0.39±0.01	1.41±0.03	52±4	14.1±0.7
	1.5	0.67	0.53±0.01	1.91±0.03	96±6	18.2±0.8
	2	0.5	0.55±0.01	1.97±0.02	129±7	15.9±0.6
Overall			0.5±0.1	1.6±0.3		

The Delay circuit module was characterized by the degree to which the **Delay** complex was able to sequester free **Output**. The kinetics of the reaction between **Delay** and **Output** was too fast to capture using the concentration ranges tested in order to fit reaction rate constants for the set of reactions show in Supp. Figure 6. The Delay reaction is slightly reversible due to 2 extra bases on the **Delay** complex that are not complementary to the **Output** (shown in green). These bases are important for decreasing undesired reactions between the **Source** and **Delay** complexes, whose reaction rate constant is on the same order as the Production circuit (initialized *via* fraying). However, this reaction set is expected to be very forward reaction dominated since $k_{forward}$ ~$2x10^6$ 1/M-sec and $k_{reverse}$ ~10^2 1/M-sec.[4]

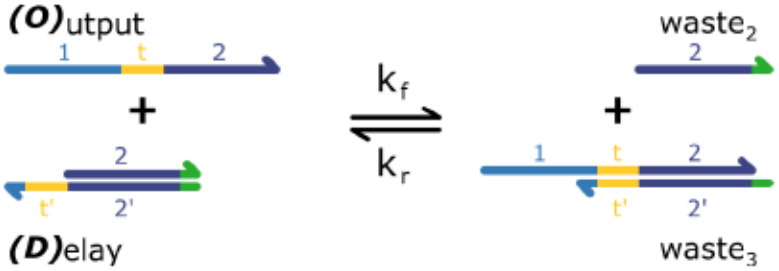

Figure 14.7. Schematic for the Delay circuit. The **Output** reacts with the **Delay** complex through a 7 base-pair toehold to produce two waste species. Two extra bases on the **Delay** complex (shown in green) inhibit a leak reaction between **Source** and **Delay** complexes.

Two sets of experiments were run to investigate whether the **Delay** complex efficiently sequestered **Output** in a reaction solution. The first set was run by adding **D** at various concentrations to the **Reporter** followed by the addition of **O**. As shown in Supp. Fig. 7a, the concentration of **O** detected closely matched the expected result: $[\mathbf{O}] = [\mathbf{O}]_0 - [\mathbf{D}]$. The second set of experiments was run by first adding **O** to the **Reporter** followed by the addition of **D**, leading to a sudden decrease in fluorescence intensity and detected free **O** (Supp. Fig. 7b). Again the remaining [**O**] matched the expected concentrations determined by the amount of **D** added. The concentration of **O** remaining when the **D** exceeded the initial concentration of **O** added did not fully decrease to zero due to the reversibility of the Delay reaction (Supp. Fig. 6).

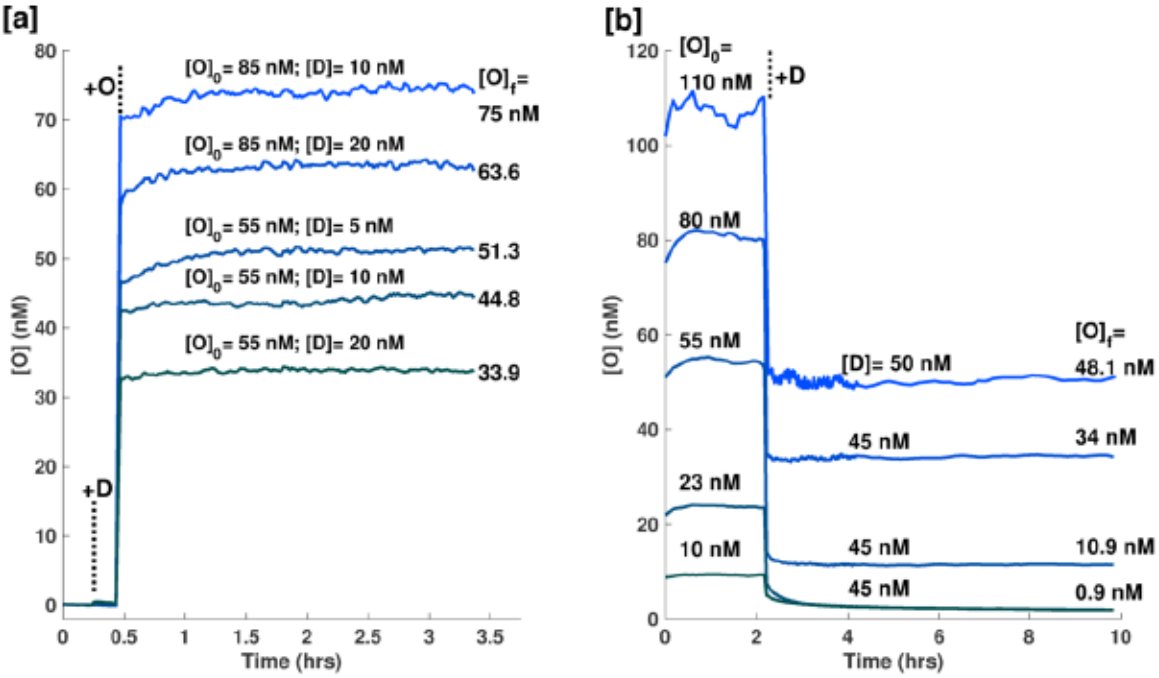

Figure 14.8. Characterization of the Delay circuit. (a) **O** is added to **R** and **D** leading to a rise in detected **O** by the **Reporter**. Black dashed lines indicate the times **D** or **O** is added to the **Reporter** solution. The concentration of **O** and **D** added to each reaction mixture is annotated above each trace. $[\mathbf{O}]_f$ is the average [**O**] over the last 30 data points. (b) **D** is added to **Reporter** and **O** leading to a decrease in fluorescence as **O** is being sequestered by **D**. The data in Fig. 3b of the Main Text is calculated from the data shown here in (a)

and (b).

14.4 | Timer Experiments with System 1

Timer experiments were conducted similar to Production experiments, except both **Source** and **Delay** were mixed with the **Reporter** after the initial baseline was measured (~1 hour). **Initiator** was mixed into the wells after the intensity reached a steady-state, at about 22-24 hours. Both **S** and **I** were kept equal in these experiments. From Table 2, [S]=[I]=1 μM produces **Output** at ~1.8 nM/hr, [S]=[I]=0.5 μM at ~0.6 nM/hr and [S]=[I]=0.25 μM at ~0.15 nM/hr (Fig. 3c-d in the Main Text and Supp. Figs. 8-10).

The delay time (t_{Delay}) was determined by calculating a moving linear fit of each curve post-**Initiator** addition. The span of each fit included 100 or 60 data points, depending on the reaction conditions. The step size was ½ of the span in each case (50 and 30 points). The x-intercept from the fit with the largest slope was chosen as the delay time. Other algorithms (*e.g.* the time [O] or d[O]/dt surpasses a specified value) gave similar time delays, but were more sensitive to noise/bias.

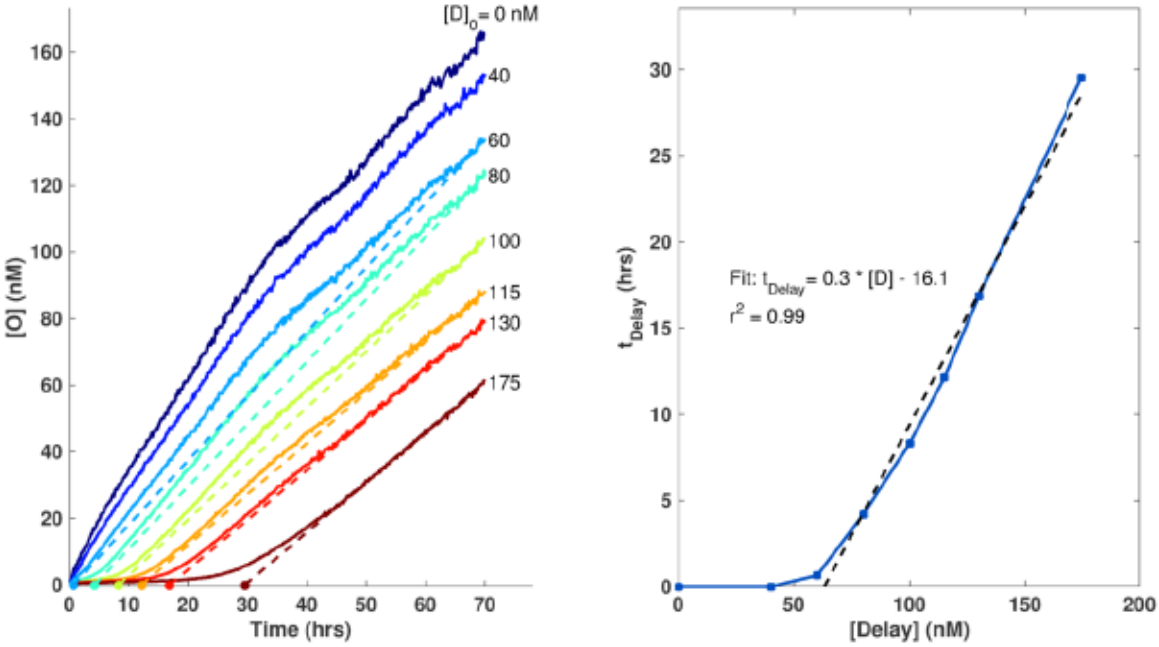

Figure 14.9. [Output] *vs.* time and t_{delay} *vs.* [Delay]

for reactions using [S]=[I]=1 μM production conditions for System 1. Dashed lines in the left plot are guides showing the calculated time delay. Dashed line in the right plot shows a linear fit for points with t_{Delay} greater than zero. Data is the same as in Figure 3c,d of the Main Text.

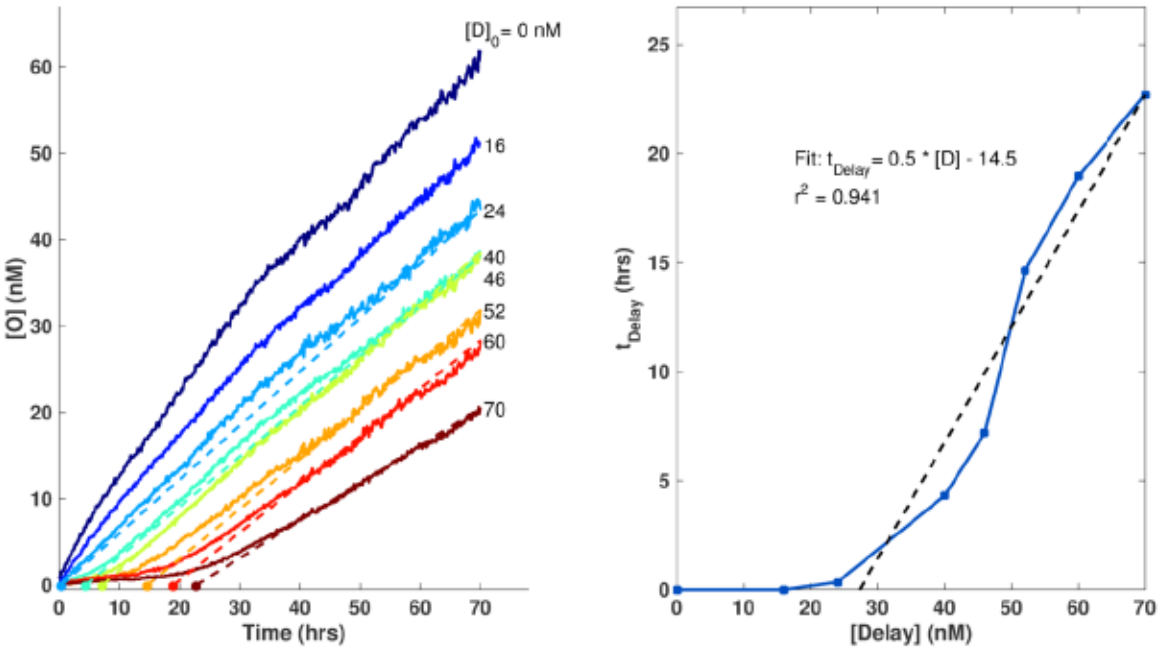

Figure 14.10. [Output] *vs.* time and t_{delay} *vs.* [Delay] for reactions using [S]=[I]=0.5 μM production conditions for System 1. Dashed lines in the left plot are guides showing the calculated time delay. Dashed line in the right plot shows a linear fit for points with t_{Delay} greater than zero.

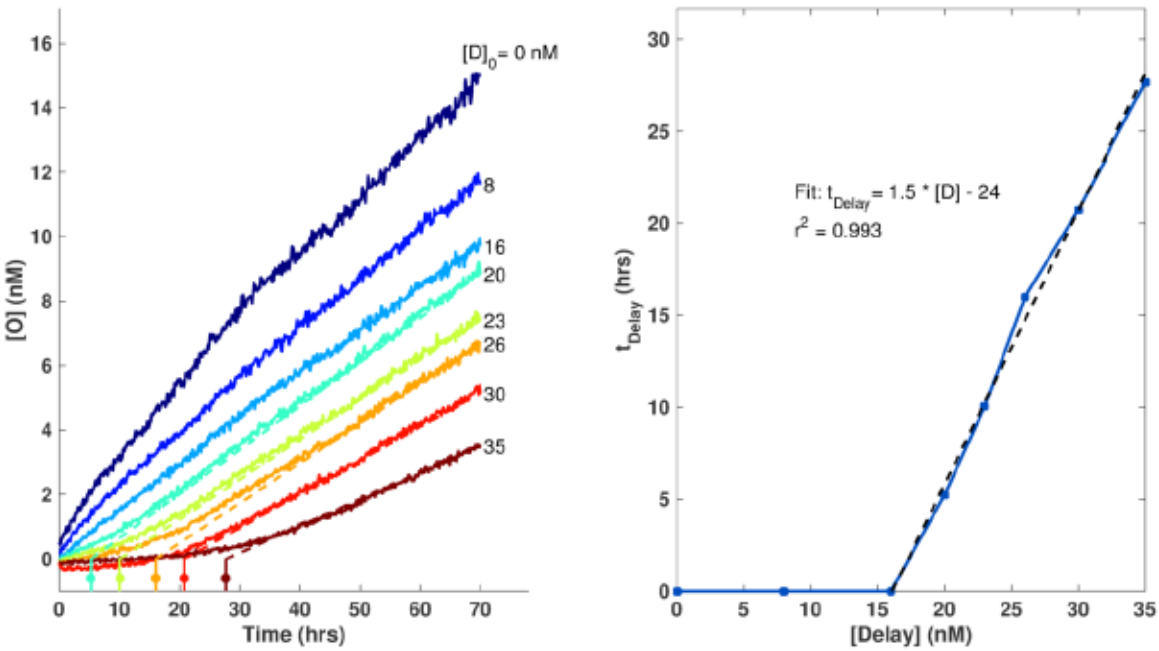

Figure 14.11. [Output] *vs.* time and t_{delay} *vs.* [Delay] for reactions using [S]=[I]=0.25 μM production conditions for System 1. Dashed lines in the left plot are guides showing the calculated time delay. Dashed line in the right plot shows a linear fit for points with t_{Delay} greater than zero.

The rate of production (d[**O**]/dt) was calculated for each reaction condition. The slope of each curve was calculated as the average over the last 90 data points (15 hours). We found that the production rate at that time point decreased slightly as a function of [**Delay**], possibly due to the uncharacterized reactions mentioned elsewhere. Despite the **Delay** dependent effects observed, an appropriate [**Delay**] can be chosen from a desired production rate and time delay using Table 2 and Supp. Figure 11.

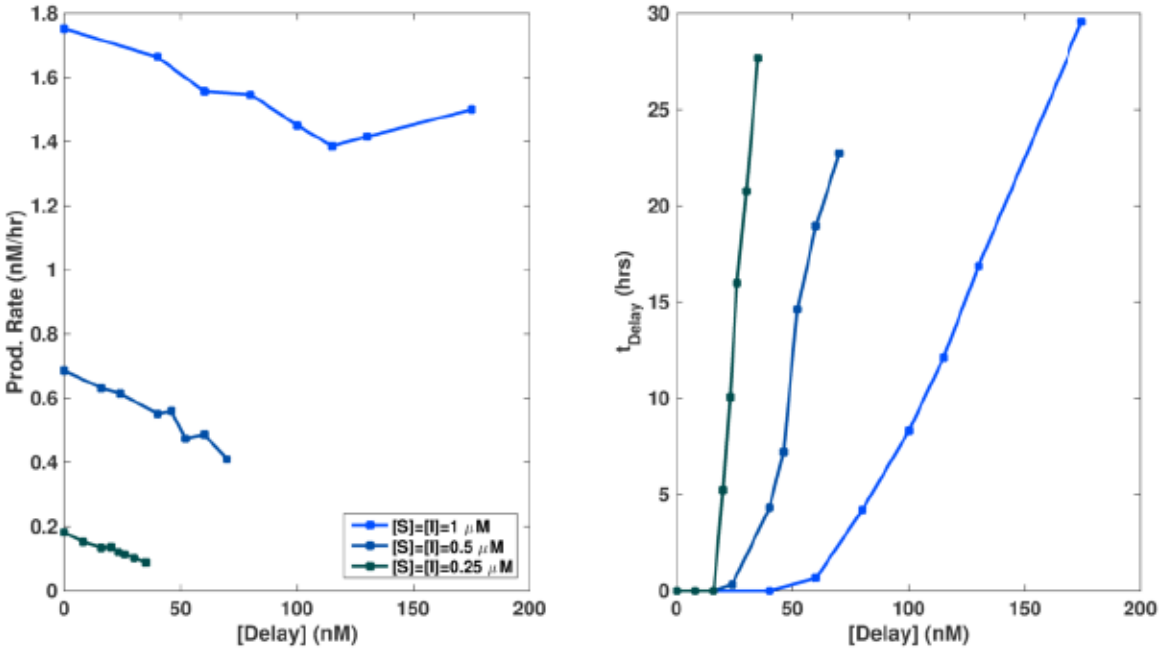

Figure 14.12. Production rate *vs.* [Delay] and t_{delay} *vs.* [Delay] for System 1. Production rate was calculated as the average over the last 90 data points (15 hours). Production rate decreases with **[Delay]** due to possible undesired reactions between circuit components.

14.5 | Timer Experiments with System 2

Experiments for System 2 were conducted the same as with System 1 except with a **[Reporter]** of 200 nM. **Output** production from System 2 was found to be slower than with System 1 (Supp. Figures 12-14). This could be due to DNA sequence differences between the systems; notably the toehold domain of System 2 is expected to have weaker binding than that of System 1 because it has less G-C base content. The 7bp toehold of the **Delay** complex is weaker as well. Additionally, if there are significant interactions between the **Source** or **Delay** and the **Reporter** complex, the increased **Reporter** concentration could be an attributing factor. Finally, while System 1 had a decrease in production rate with increasing **[Delay]**, System 2 showed an increase in production rate (Supp. Figure 15).

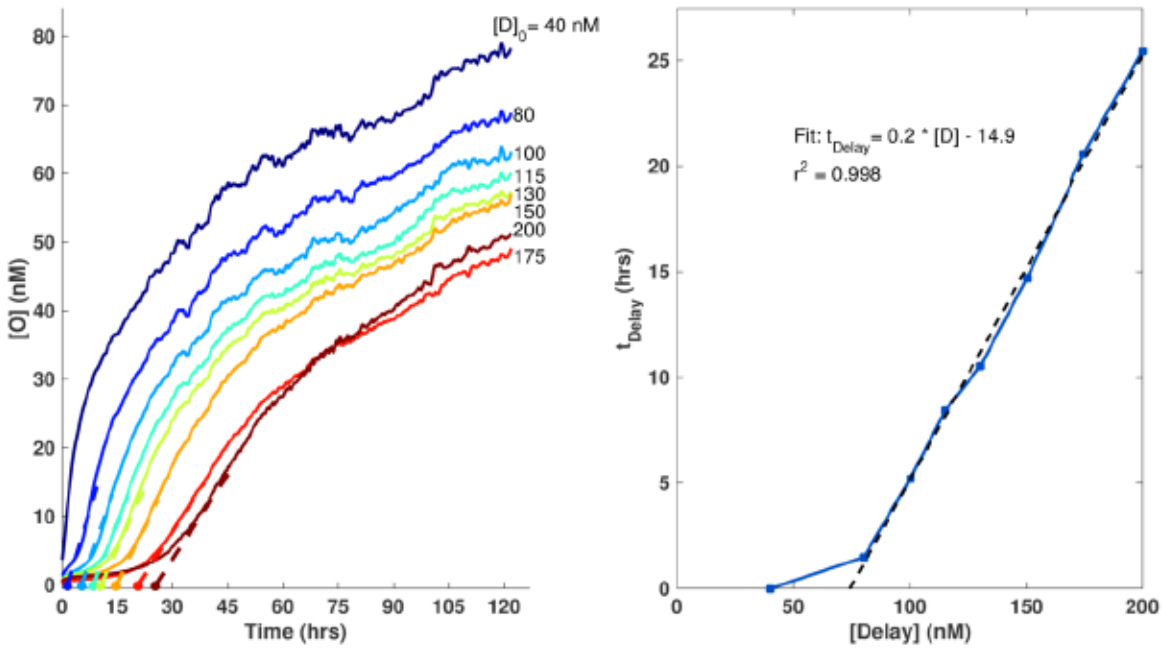

Figure 14.13 [Output] *vs.* time and t_{delay} *vs.* [Delay] for reactions using [S]=[I]=1 μM production conditions for System 2. Dashed lines in the left plot are guides showing the calculated time delay. Dashed line in the right plot shows a linear fit for points with t_{delay} greater than zero.

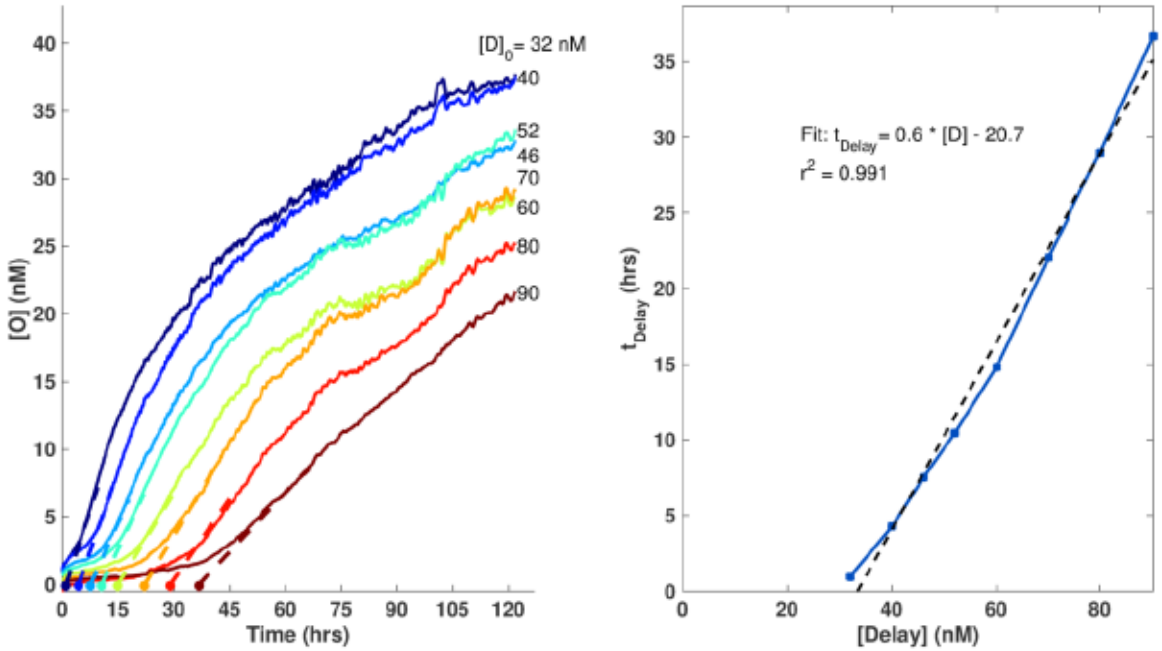

Figure 14.14. [Output] *vs.* time and t_{delay} *vs.* [Delay] for reactions using [S]=[I]=0.5 μM production conditions for System 2. Dashed lines in the left plot are guides showing the calculated time delay. Dashed

line in the right plot shows a linear fit for points with t_{delay} greater than zero.

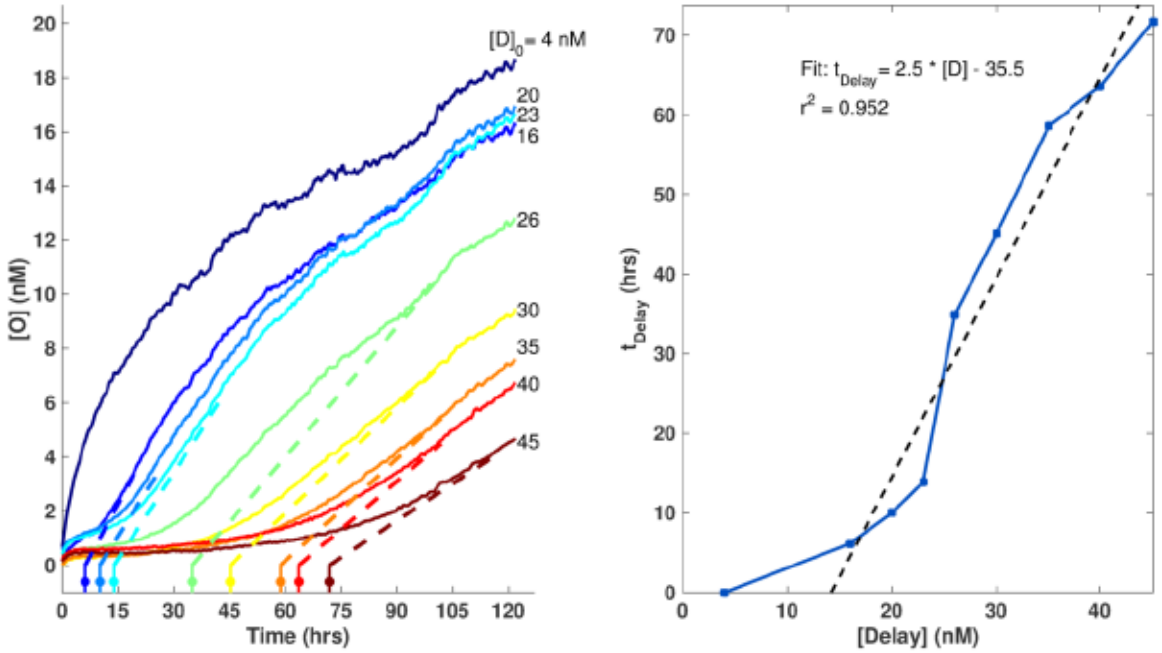

Figure 14.15 [Output] *vs.* time and t_{delay} *vs.* [Delay] for reactions using [S]=[I]=0.25 μM production conditions for System 2. Dashed lines in the left plot are guides showing the calculated time delay. Dashed line in the right plot shows a linear fit for points with t_{delay} greater than zero.

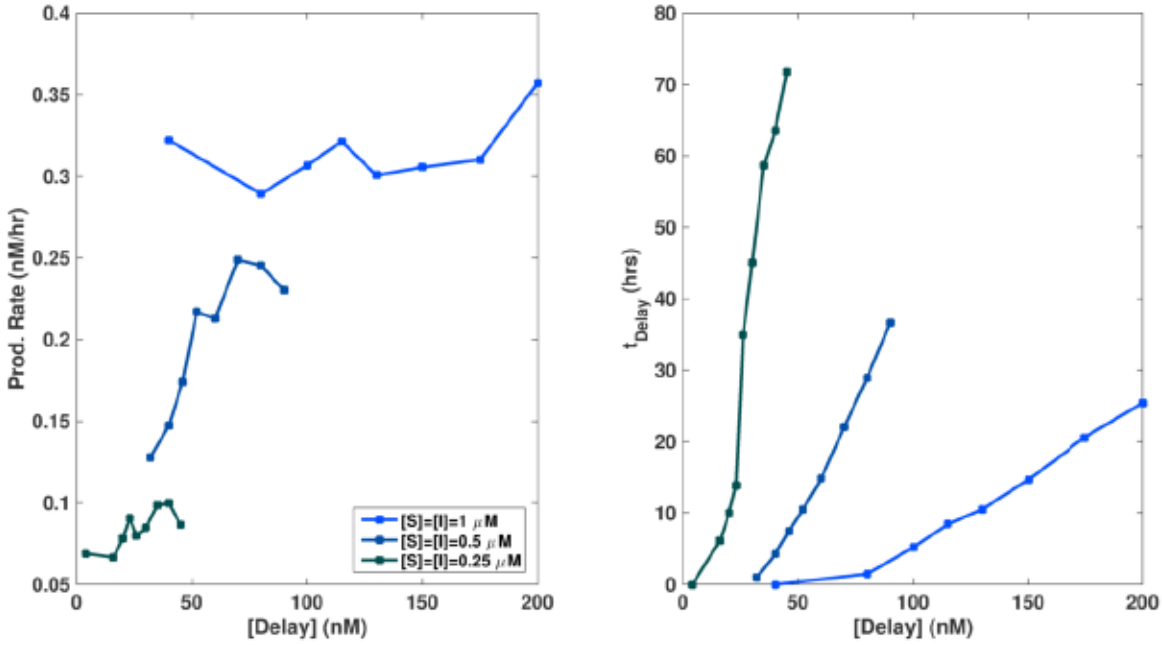

Figure 14.16. Production rate *vs.* [Delay] and t_{delay} *vs.* [Delay] for System 2. Production rate was

calculated as the average over the last 250 or 100 data points (about 40 or 16 hours). Production rate increases with [Delay] due to possible undesired reactions between circuit components.

14.6 | Multiplex Timer Experiments (Systems 1 and 2)

Multiplexing experiments were conducted the same as the experiments described in SI 3-4. Briefly, the **Reporters** of each system were mixed and a baseline was taken followed by the addition of **Delay** and **Source** complexes. **Initiator** was added after 22-24 hours. For experiments comparing data of each system in isolation *vs.* together, the reaction solution contained **Reporters** from both systems, but only the **D**, **S** and **I** from the system being studied (dashed lines in Supp. Fig. 16). System 1 was tracked using FAM and System 2 with TexasRed fluorophores using two different filters on the qPCR. Fluorescence from one fluorophore was not observed when measuring the fluorescence of the other fluorophore.

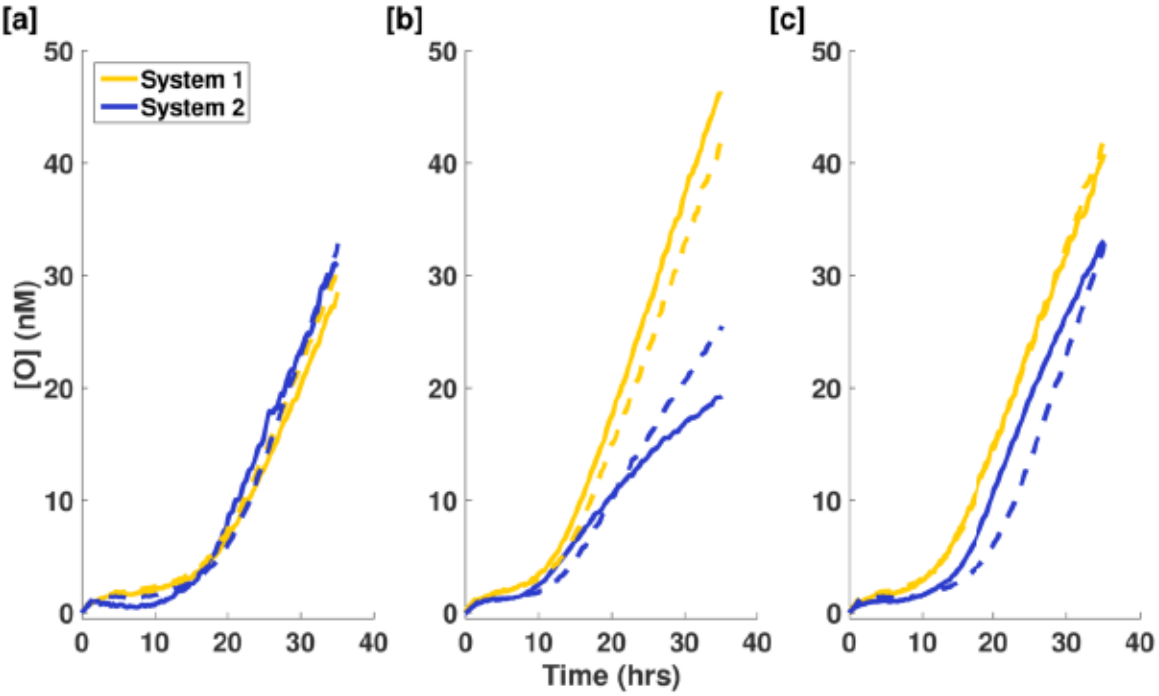

Figure 14.17. Additional examples of multiplexing two timer circuits. In each case, dashed lines indicate a reaction with the system in isolation. (a) Reaction mixture conditions were chosen such that both systems would release their respective O at the same time (19 hours) and rate. $[S]_0=[I]_0=1$ μM; $\{[D]_{Sys1}, [D]_{Sys2}\}=\{130$ nM, 165 nM$\}$. (b) The release rate of each system can be independently controlled while keeping the delay time constant (9 hours). $[S]_0=[I]_0=1$ μM for Sys1 and 0.5 μM for Sys2; $\{[D]_{Sys1}, [D]_{Sys2}\}=\{100$ nM, 50 nM$\}$.

(c) The time of release (9 and 19 hours) of each system can be independently controlled while keeping the release rate constant. $[S]_0=[I]_0=1$ μM; $\{[D]_{Sys1}, [D]_{Sys2}\}=\{100$ nM, 165 nM$\}$.

14.7 | Timer Circuit Simulations and Characterization of Leak Reactions

Since the DNA strand-displacement circuit can be represented by a series of mathematical equations (SI-Equations 1 and 5 and SI Figure 6), a model that matches the experimental behavior of the system could be built to further tune the circuit for future applications. However, we found that a simple model derived from those equations failed to quantitatively capture the delay times observed in experiments (Supp. Figure 17). Based on that mismatch and the observation of an increase in fluorescence (or detected **O**) by the **Reporter** when **S** is added in the absence of **I** or **D** (Supp. Figure 4), we postulated that a series of interactions might exist between **S**, **D**, **I** and **Reporter** beyond what is predicted by the simple model.

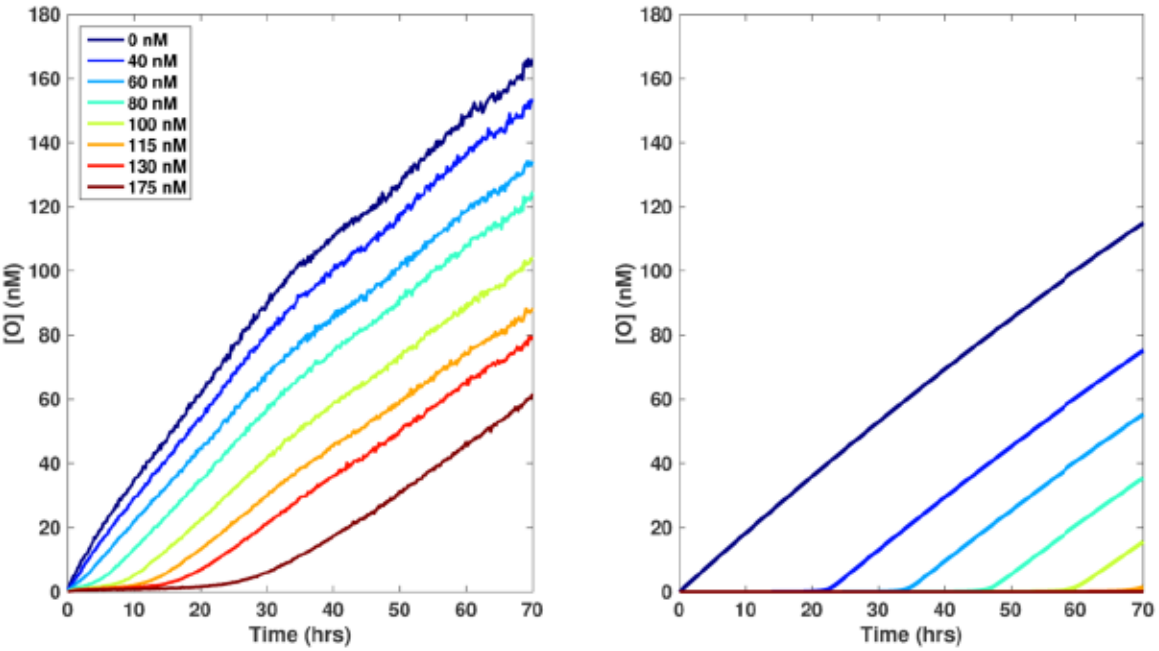

Figure 14.18. Comparison of data (left) and a model considering only abstract reactions described in

SI-Eqn. 5 and Supp. Fig. 6 (right) for System 1 using 1 μM **S** and **I**. **Delay** concentrations for both plots are shown in the legend. Parameters for the model were as described in SI 2 (Supp. Figs. 3,6) and in Zhang and Winfree.[4] Note the decrease in overall production and increase in time delay of the model prediction compared to experimental results.

In control experiments, we found that mixing the **Reporter** and **Source** quickly produced an observable fluorescence signal in the absence of **Initiator** (Supp. Fig. 4), suggesting that the **Source** complex and the **Reporter** interacted in an undesired or "leak" reaction (Supp. Fig. 18a). While by design these species could interact through a 4-way 0bp pair branch migration interaction, such a pathway would not explain the fast rate of reaction we observed. We therefore postulated that this reaction could be due to truncations or base mismatches within the toehold region of the bottom strand of some **S** complexes. We designated **S** complexes with these variations as the subspecies $\mathbf{S}_{\mathbf{Leak}}$. These complexes would not have been separated from pure **S** complex during the purification process because their electrophoretic mobility is very similar to that of **S**.

We also observed that some leftover **O** remained after the gel purification process due to the proximity of the bands in the gel. Purifying **S** using a 15% polyacrylamide gel instead of a 10% gel significantly reduced the level of pre-initiation **O** detected, but some may still remain in the purified **S** complex solution. The leftover **O** and $\mathbf{S}_{\mathbf{Leak}}$ complex are also expected to interact with the **Delay** complex for timer circuit reactions. We also considered a leak reaction between the **Initiator** and the **Delay** complex since there are 7 complementary nucleotides for a transient hybridization. The schematic shown in Supp. Figures 18-20 shows the possible leak reactions considered here. Unless specified in the figure captions, reaction rate constants for these reactions were taken from Supplemental Reference 4 or fit using the bimolecular rate equation shown in SI 2 and Supp. Figure 5. This approach of choosing reaction rate constants supports a physical representation of the postulated reactions and provides consistency with studies of other DNA strand-displacement

reactions.

While the reactions described above may account for the unintended reactions that occurred within the timer system, including these reactions in a model still predicted significantly different time delay values than what were observed experimentally. To account for the decreased time delay observed in experiments, we added a small leak reaction between pure **Source** complexes and **Delay** complexes. While such a reaction would be expected to occur with a rate constant smaller than k_{0bp}, a reaction rate constant of k_{1bp} was needed to account for the large decrease in delay time. As noted in SI 2, we found that a reaction between $\mathbf{S_{Leak}}$ and **I** produced a better fit to the production dynamics. However, we would not expect this reaction to occur since **S** and **I** are incubated prior to PAGE purification of the **S** complex, any **S** species that would quickly react with **I** would be removed. Supp. Figure 19 shows an example comparison between experimental data (System 1, [S]=[I]=1 μM) and the resulting model prediction. While these reactions are only a possible description of the interactions between the DNA species, they show that an understanding of the reaction behavior is possible through the incorporation of leak pathways. We found that incorporation of each of these leak pathways into our model, using previously published rates and the fitted parameters of $[\mathbf{S_{Leak}}]$, $k_{S,Leak}$ and k_{0bp} as described in SI 2 (no additional fitting parameters were required) produced quantitative agreement between our model and the experimental results that we observed (Supp. Fig. 21). Thus, we expect that this model can be used to tune the system's performance, including release rates and delay times.

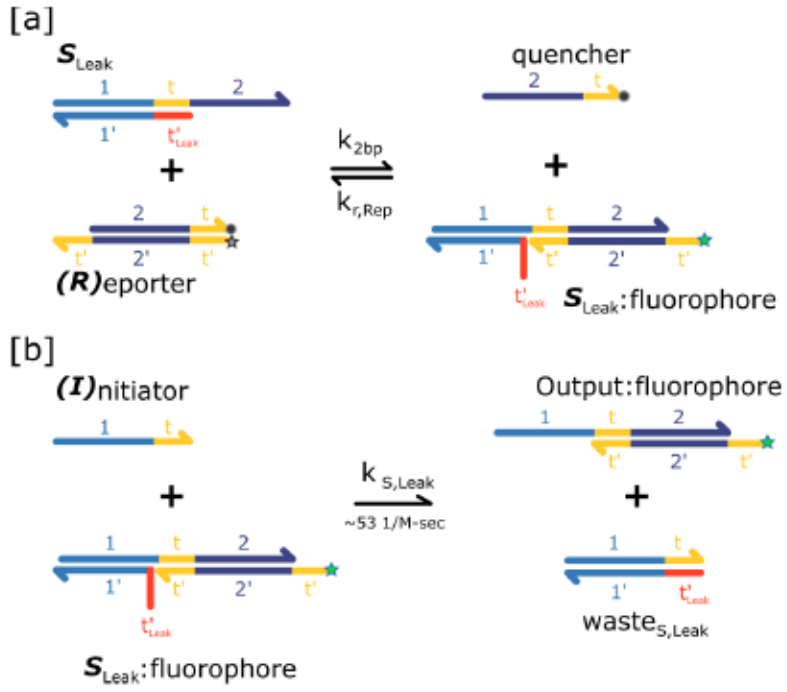

Figure 14.19. Leak reactions between a **Source** complex with mismatches or base truncations in the toehold region and the **Reporter** complex lead to the detection of a fluorescent signal prior to the addition of the **Initiator**. (a) S_{Leak} complex interacts reversibly with the **Reporter** to produce a fluorescent complex that reacts irreversibly with the **Initiator** (b). $k_{r,rep}$ was calculated from k_{5bp} (ref. 4) and the experimentally measured K_{eq} of the **Reporter-Output** reaction (Supp. Fig. 2).

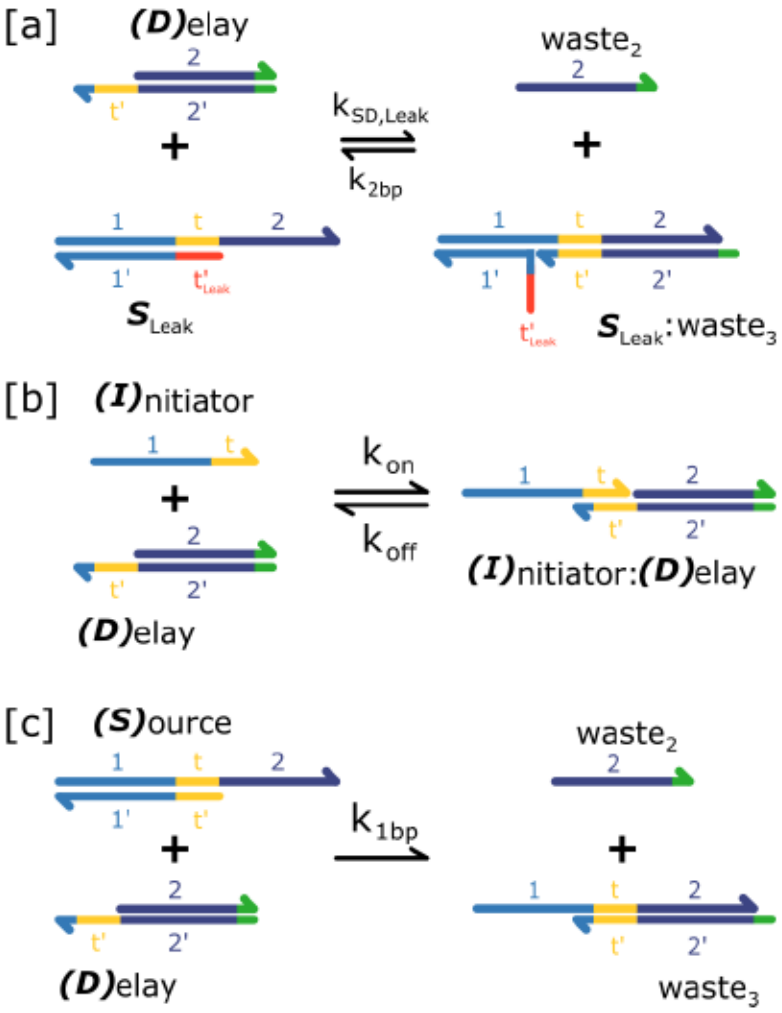

Figure 14.20. Reactions between the Delay complex and other reaction species. (a) **Delay** and S_{Leak} react reversibly with a forward rate constant estimated to be $k_{SD,Leak}$ ~250 1/M-sec. (b) **Initiator** hybridizes and de-hybridizes with the toehold of the **Delay** complex. From Zhang and Winfree,[4] k_{on} ~ 3.5×10^6 1/M-sec and k_{off} ~ 0.08 1/sec. (c) Leak reaction between pure **Source** complex and the **Delay** complex.

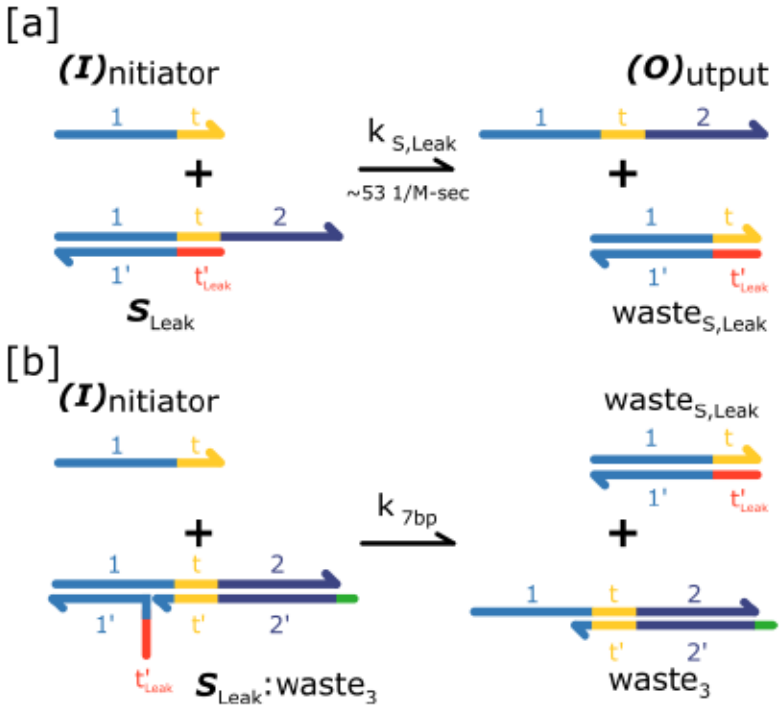

Figure 14.21. Leak reactions with Initiator. (a) Reaction between I and S_{Leak}. (b) Reaction between I and the complex produced from S_{Leak}+D.

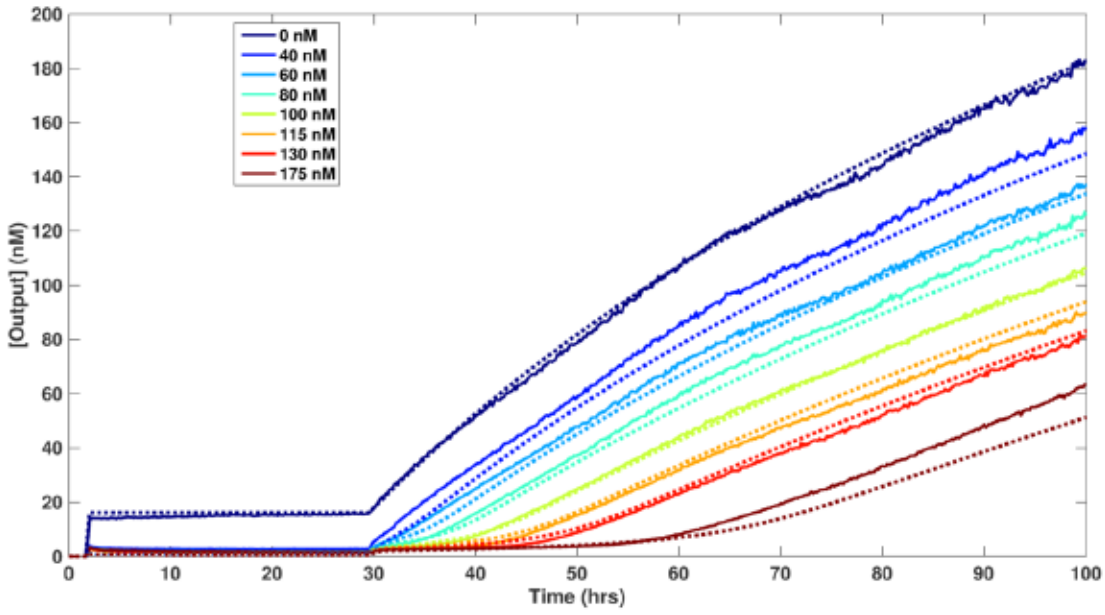

Figure 14.22. Comparison of data and the model prediction for System 1 using 1 μM S and I. Delay concentrations are listed in the legend. Reporter-only baseline was initially measured followed by Source and

Delay complex addition at 1 hour. **Initiator** was added after 30 hours. Parameters and reactions included in the model were as described above in Supp. Figs. 18-20 and in Zhang and Winfree.[4]

References

[1] Qian, L.; Winfree, E. (2011) "Scaling up digital circuit computation with DNA strand displacement cascades." *Science*, *332.6034*, 1196-1201.

[2] Zadeh J., Steenberg C., Bois J., Wolfe B., Pierce M., Khan A., Dirks R., Pierce N. (2011) "NUPACK: analysis and design of nucleic acid systems." *Journal of Computational Chemistry*, 32(1):170-173.

[3] Puglisi, J. D.; Tinoco, I. Jr. (1989) "Absorbance Melting Curves of RNA." *Methods in Enzymol.*, *180*, 304–325.

[4] Zhang, D. Y.; Winfree, E. (2009) "Control of DNA strand displacement kinetics using toehold exchange." *J. Am. Chem. Soc.*, *131.47*, 17303-17314

15 | Programming the Sequential Release of DNA

Summary. This study presents a mechanism for releasing a series of molecules into solution, one after another, with control over release timing. The process relies on stages of coupled DNA strand-displacement reactions that first release an output molecule, and then trigger the initiation of the next release stage. We demonstrate the sequential release of four different strands of DNA, using both asynchronous and clocked control over the release timing. We then demonstrate branched control over release, where the presence or absence of an input molecule determines which release pathway is active. This sequential release circuit offers a means to schedule downstream chemical events, such as steps in the assembly of a nanostructure, or stages in the stimulus response of a material.

15.1 | Introduction

At their most basic level, computers are machines that execute a programmed series of instructions. In many types of chemical computing, each instruction consists of the release of a molecule into solution, changing that molecule's concentration. One simple chemical program is, "Release output molecule A, and then release output molecule B, and then release output molecule C" (Fig. 1). This type of simple sequential program could, for example, coordinate processive behavior in physical materials if the released molecules coordinated downstream chemical reactions or assembly processeses[1].

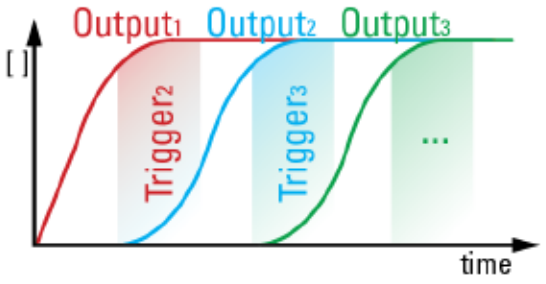

Figure 15.1. An idealized sequential release circuit. At each stage an output molecule is released, and then the next stage is triggered. The red stage triggers the cyan stage, which triggers the green stage, etc.

In this study, we outline a method to release an ordered series of output molecules in solution, from a sequestered state. The mechanism of release consists of stages of paired reactions that first release an output molecule quickly (Eqn. 1), and then slowly trigger the next reaction stage (Eqn. 2). The large difference in the rate constants of these reactions ensures that each current stage will be essentially complete before the next stage is triggered.

Release Output: $$Trigger_i + Payload_{i+1} \xrightarrow{fast} Output_i \quad (1)$$

Convert Trigger: $$Trigger_i + Convert_{i,i+1} \xrightarrow{slow} Trigger_{i+1} \quad (2)$$

This system can either run asynchronously, or can be coupled with a central clocking mechanism that slowly generates $Trigger_1$ to control the pace of execution (Eqn. 3, see details in section 2).

Clock control: $$\emptyset \xrightarrow{k_{prod}} Trigger_1 \quad (3)$$

Here we implement sequential release programs that can release different sequences of DNA using DNA strand-displacement (DSD) reactions. DSD reactions are designed interactions between short synthetic strands of DNA, in which an input strand binds to a partially double-stranded complex and displaces an incumbent output strand into solution[2]. The reaction rate constants for DSD reactions are mediated by a short single-stranded DNA 'toehold' domain. By varying the length of the toehold domain from 0 to 7 nucleotides, the rate constant can be tuned across six

orders of magnitude at room temperature[3]. Cascades of DSD reactions, in which the output of one reaction can serve as a reactant to a downstream process, have been used to implement a growing library of signal processing circuits, including amplifiers[4-7], Boolean logic gates[8-12], a neural network[13], an oscillator[14], a timer[15], and a feedback controller[16]. Further, DSD circuits can control molecules other than DNA by designing their output to be aptamers, or sequences that can bind to proteins[17] and small molecules[18].

15.2 | A sequential release cascade

To implement the reactions outlined in Eqn.'s 1-2, we start by specifying a set of target Output strands of DNA that we wish to release in series. Each Output is initially sequestered in an inert state within a complex called the Payload. A set of Trigger strands is then designed to release the Outputs from the Payloads, following Eqn. 1. We select a 7nt toehold for the Trigger complex, which has the fastest rate constant ($\sim 4 \mu M^{-1} s^{-1}$) available to DSD reactions at room temperature[3]. Next, we designed a set of Convert reactions to translate each Trigger_i molecule into the next Trigger_{i+1} molecule in the series, following Eqn. 2, mediated by a slower 4bp toehold ($\sim 2 \cdot 10^{-2} \mu M^{-1} s^{-1}$). To track the Outputs, we append a fluorescent modifier to each of the Outputs, and a quencher to the bottom strands of the Payloads (Fig. 2a). Our design uses the "leakless" DNA strand-displacement architecture[19] to suppress unintended leak reactions directly between the Convert and Trigger complexes. This architecture imposes some sequence overlap between the Outputs from each stage, however the sequence overlap can be eliminated with an additional translator reaction inserted after each stage (see SI1).

To test whether the sequential release cascade releases Outputs in the sequential order, we prepared a four-stage cascade by mixing $25nM$ of each Payload complex, together with $37.5 \cdot (4 - i) nM$ of the $\text{Convert}_{i,i+1}$ complexes for i=1, 2 or 3 to translate each Trigger_i to Trigger_{i+1}.

Earlier stages require high concentrations of their convert complex to drive all of the remaining downstream stages. $112.5nM$ of the Trigger_1 strand was added to this mix to trigger the reaction cascade, and the resulting kinetics were tracked by measuring changes in fluorescence over time (See full materials and methods in SI2). The fluorescent Outputs were released in the designed order (Fig. 2b). The release rate decreased at every stage, as additional convert steps (of the form in Eqn. 2) are required to translate Trigger_1 from the 1st stage into Trigger_i on increasing i^{th} stages.

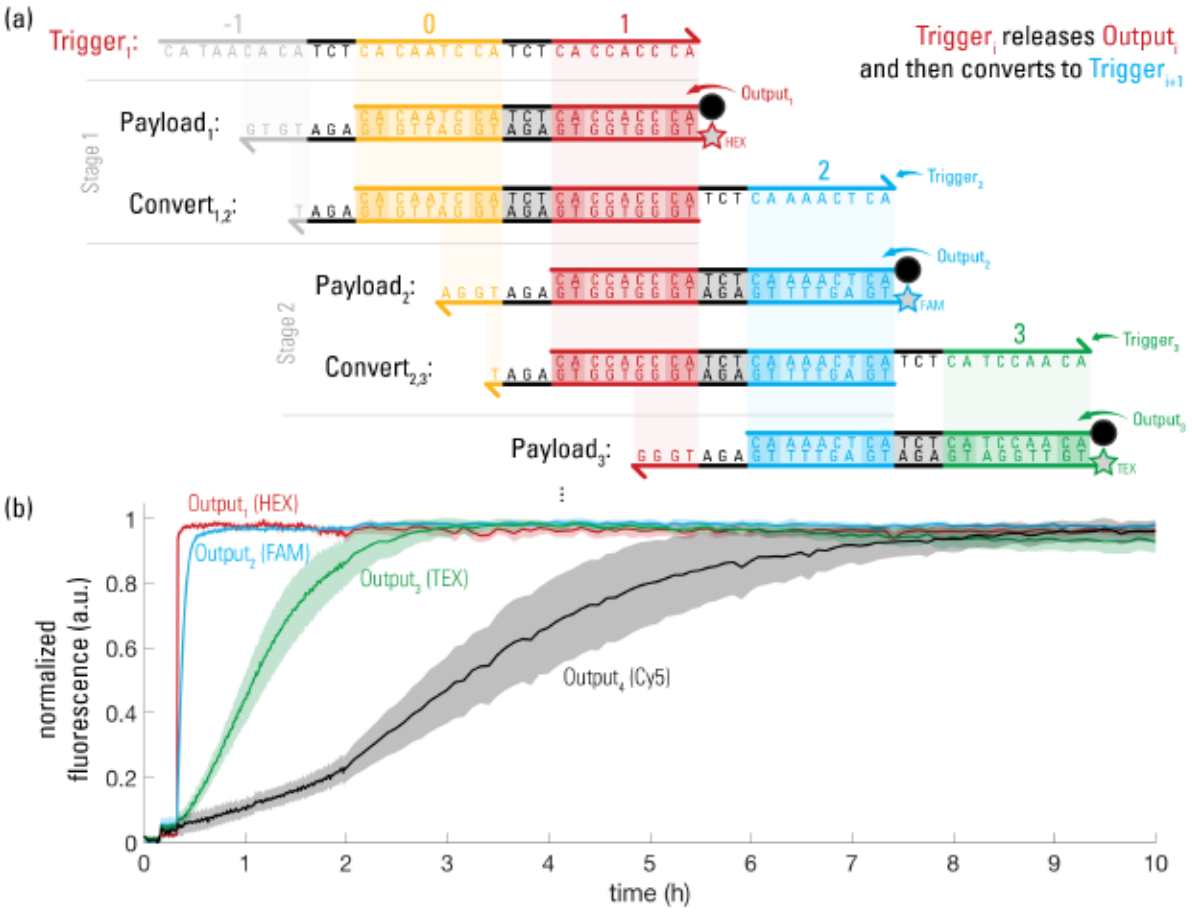

Figure 15.2. Asynchronous Sequential Release. (a) The reaction cascade consists of stages of Payload and Convert complexes (Stages 1-3 shown, see SI3 for sequences to all stages). At each stage, a Trigger molecule first reacts quickly with the Payload to release a fluorescent Output into solution. Any remaining Trigger then reacts slowly with the Convert complex, which converts it into the Trigger molecule for the next stage. (b) Experimental data showing the fluorescent Outputs being released in order, with $25nM$ Payloads and

$37.5 \cdot (4 - i)nM$ $\text{Convert}_{i,i+1}$. We used the leakless architecture[19] to prevent some unintended leak reactions between $\text{Convert}_{i-1,i+1}$ and Payload_i complexes. See SI4 for data processing procedure.

15.3 | Clocking

To make the timing of the release events more uniform, we add an upstream clocking reaction that continuously produces Trigger_1 at a constant rate, replacing the large initial concentration of Trigger_1. In this manner, we can make the clocking production reaction into the rate-limiting step and thus control the pace of the reaction. The production reaction is performed by initially sequestering Trigger_1 within a high concentration Source complex. An Initiator strand reacts with the Source to displace Trigger_1 into solution (Fig. 3a), where it can then react with the sequential release cascade. By designing the reaction between Source and Initiator to have a very small rate constant, we can ensure very little Source and Initiator is consumed on timescales of several days, and thus can approximate their concentrations as constant. This allows us to treat the net production rate as roughly a constant k_{prod} for the duration of the experiment (Eqn. 1)[15,16].

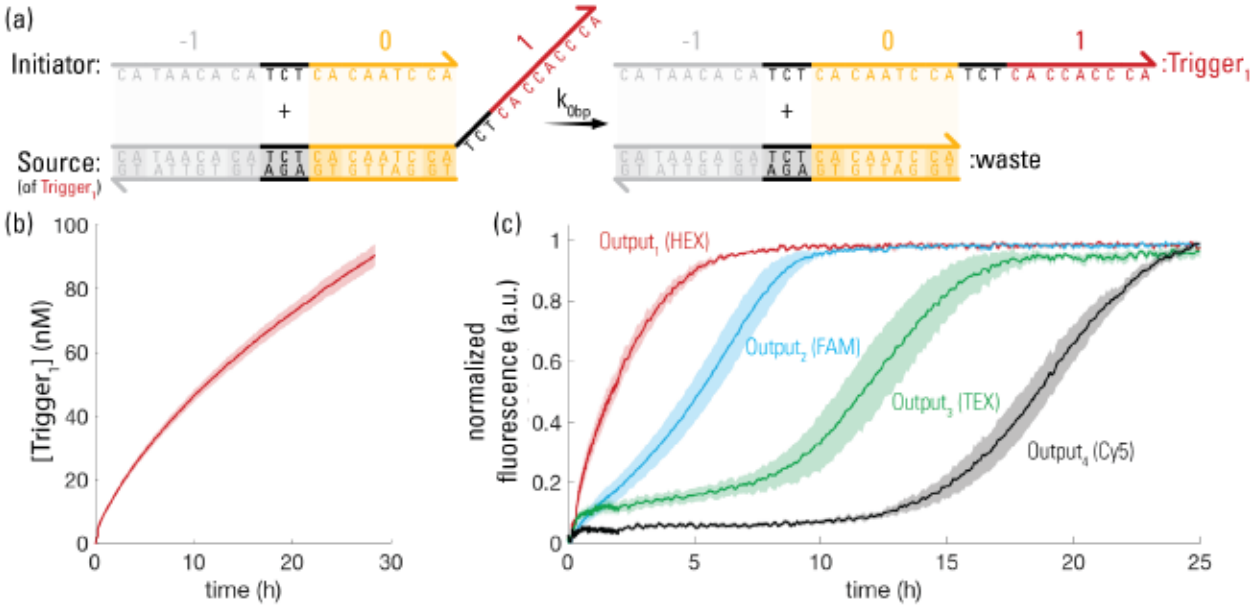

Figure 15.3. Clocking. (a) Clock production DSD circuit[15]. (b) Experimental data showing the production

circuit releasing $Trigger_1$, with $1\mu M$ Source and Initiator, without a downstream sequential release cascade. (c) The sequential release cascade connected to the production circuit with $1\mu M$ Source and Initiator, $25nM$ Payloads, and $37.5 \cdot (4-i)nM$ $Convert_{i,i+1}$. The timing of release events is now rate limited by the production circuit, making the delay times between stages roughly linear.

$$\frac{d[Trigger_1]}{dt} = k_{release}[Source][Initiator] \approx k_{prod} \qquad (1)$$

To verify that the clock circuit can sustain approximately linear release rates of the $Trigger_1$ molecule, we mixed Source and Initiator together at $1\mu M$ in the presence of $300nM$ $Payload_1$. By tracking increase in fluorescence, we could indirectly infer the rate at which $Trigger_1$ was released (Fig. 3b).

Next, we coupled the clock reaction with the sequential release cascade to release the Outputs at regular intervals. We mixed $1\mu M$ Source and Initiator with $25nM$ of each Payload complex, together with $37.5 \cdot (4-i)nM$ of each $Convert_{i,i+1}$ complex. As expected, we still observed the same order of Output release events, but now with a slower and more regular interval between the stages (Fig 3c).

15.4 | Branched Pathways and Conditional Statements

The circuits in the previous section execute a linear release program with no branching, which is analogous to a computer program that does not have any conditional "if" statements. Every time they are run, they release the same molecules in the same hard-coded order. Including the capacity for conditional statements would allow for release programs to make decisions based on external factors sensed in solution, such as the presence of other signal strands of DNA. Therefore, we next updated the design of the Convert complexes to create a conditional Convert complex, iConvert, which is initially inactive and can be conditionally activated through a reaction with a Deprotect strand (Eqn. 4).

Branching: $$Deprotect_{i,j} + iConvert_{i,j} \xrightarrow{k_{deprotect}} Convert_{i,j} \quad (4)$$

The toeholds on iConvert complexes are covered to prevent reactions with the trigger signals (Fig. 4a). A Deprotect strand is designed to expose the toehold on the inactive Convert complex, allowing it to react with the trigger. For simplicity we approximate the deprotection reaction as irreversible, because the product activated convert complex is irreversibly consumed by Eqn. 2. This serves as a conditional statement of the form:

if(**Deprotect**$_{i,j}$ is present){

convert(**Trigger**$_i$ to **Trigger**$_j$);

}

Multiple inactive Convert complexes can be combined in the same solution to create conditional statements with more than one branching cases.

We tested this conditional circuit by preparing a two-stage sequential release circuit with a single Payload for the first stage and two different Payloads for the second stage. We refer to these species as Payload$_1$, Payload$_{2A}$ and Payload$_{2B}$. Two inactive Convert complexes iConvert$_{1,2A}$ and iConvert$_{1,2B}$ form separate branches to release the Outputs from their respective Payloads. The inactive Convert complexes can be activated by their Deprotect strands (see SI2 for sequences). We mixed together two batches of the clock circuit ($1\mu M$ of Source and Initiator), 37.5 nM of iConvert$_{1,2A}$ and iConvert$_{1,2B}$, and 25 nM of Payload$_{2A}$ and Payload$_{2B}$. To one batch we added 50 nM of Deprotect$_{1,2A}$ (Fig. 4b), and to the other batch we added 50 nM of Deprotect$_{1,2B}$ (Fig. 4c). In both cases, the Output for the activated branch was released, while the Output for the inactive branch was not significantly released.

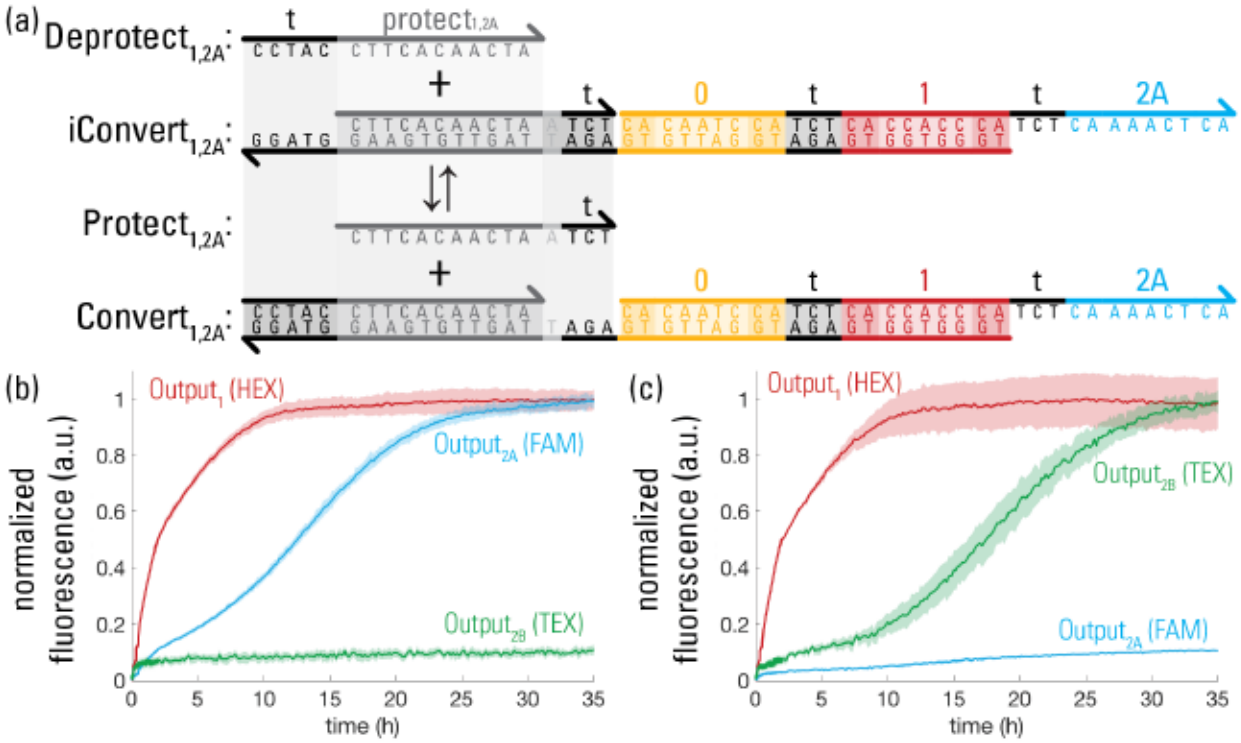

Figure 15.4. Branching. (a) DSD diagram for a conditional $Convert_{1,2A}$ complex, which is only active in the presence of an associated Deprotect strand (see sequences for parallel $Convert_{1,2B}$ system in SI2). (b) Experimental results for a branched two-stage sequential release program, with branch 2A deprotected. (c) Experimental results for a branched two-stage sequential release program, with branch Bb activated. For both (b) and (c) we used $1\mu M$ Source and Initiator, $25nM$ Payloads, $37.5nM$ Convert complexes, and $50nM$ of the stated Deprotect strands.

15.5 | Discussion

In this paper we developed a DNA strand-displacement circuit that releases a series of different Output strands of DNA, one after another. This circuit serves as a simple scheduling program to trigger molecular events at discrete times. We specifically demonstrated a four-stage circuit with 25 nM Payload concentrations per stage. We showed that the circuit can run in an asynchronous or clocked configuration. In the asynchronous mode, the time delay between stages is non-uniform and

slows down dramatically between stages, while the clocked mode enforces more uniform temporal spacing between stages. Lastly, we demonstrated that the circuit can be modified to enable conditional logic, where different branches of the release program can be activated depending on the presence of activating signal strands in the solution.

The four-stage release circuit demonstrated here is relatively primitive, and can only coordinate four events. Larger circuits with additional stages, or looped circuits in which feedback from the final stage stimulates an additional round of release from the initial stage could significantly increase the complexity of the coordinated events. Further, the conditional branching mechanism could be re-imagined as a "pause" feature, in which the release program halts until receiving a "continue" signal from the environment. Finally, coupling between the Output strands and other systems, such as aptamers[17,18] or nanostructures[1] could enable these sequential release circuits to control a wider variety of downstream outputs.

References

[1] Zhang, David Yu, et al. "Integrating DNA strand-displacement circuitry with DNA tile self-assembly." *Nature communications* 4 (2013).
[2] Zhang, David Yu, and Georg Seelig. "Dynamic DNA nanotechnology using strand-displacement reactions." *Nature chemistry* 3.2 (2011): 103-113.
[3] Zhang, David Yu, and Erik Winfree. "Control of DNA strand displacement kinetics using toehold exchange." *Journal of the American Chemical Society* 131.47 (2009): 17303-17314.
[4] Seelig, Georg, Bernard Yurke, and Erik Winfree. "Catalyzed relaxation of a metastable DNA fuel." *Journal of the American Chemical Society* 128.37 (2006): 12211-12220.
[5] Chen, Xi, et al. "Stacking nonenzymatic circuits for high signal gain." *Proceedings of the National Academy of Sciences* 110.14 (2013): 5386-5391.
[6] Zhang, David Yu, and Georg Seelig. "DNA-based fixed gain amplifiers and linear classifier circuits." *International Workshop on DNA-Based Computers*. Springer Berlin Heidelberg, 2010.
[7] Zhang, David Yu, et al. "Engineering entropy-driven reactions and networks catalyzed by DNA." *Science* 318.5853 (2007): 1121-1125.
[8] Seelig, Georg, et al. "Enzyme-free nucleic acid logic circuits." *science* 314.5805 (2006): 1585-1588.
[9] Takahashi, Keiichiro, et al. "Chain reaction systems based on loop dissociation of DNA." *International Workshop on DNA-Based Computers*. Springer Berlin Heidelberg, 2005.
[10] Zhang, Cheng, Jing Yang, and Jin Xu. "Circular DNA logic gates with strand displacement." *Langmuir* 26.3 (2009): 1416-1419.
[11] Qian, Lulu, and Erik Winfree. "Scaling up digital circuit computation with DNA strand displacement cascades." *Science* 332.6034 (2011): 1196-1201.
[12] Genot, Anthony J., Jonathan Bath, and Andrew J. Turberfield. "Reversible logic circuits made of DNA." *Journal of the American Chemical Society* 133.50 (2011): 20080-20083.

[13] Qian, Lulu, Erik Winfree, and Jehoshua Bruck. "Neural network computation with DNA strand displacement cascades." *Nature* 475.7356 (2011): 368-372.
[14] Srinivas, Niranjan, et al. "Enzyme-free nucleic acid dynamical systems." *bioRxiv* (2017): 138420.
[15] Fern, Joshua, et al. "DNA Strand-Displacement Timer Circuits." *ACS Synthetic Biology* (2016).
[16] Scalise, Dominic, Nisita Dutta and Rebecca Schulman. "DNA Strand Buffers." *Journal of the American Chemical Society* (2018)
[17] Gatto, B., M. Palumbo, and C. Sissi. "Nucleic acid aptamers based on the G-quadruplex structure: therapeutic and diagnostic potential." *Current medicinal chemistry* 16.10 (2009): 1248-1265.
[18] Huizenga, David E., and Jack W. Szostak. "A DNA aptamer that binds adenosine and ATP." *Biochemistry* 34.2 (1995): 656-665.
[19] Thachuk, C., Winfree, E., & Soloveichik, D. (2015, August). Leakless DNA strand displacement systems. In *International Workshop on DNA-Based Computers* (pp. 133-153). Springer, Cham.

Biographical Sketch

Dominic Scalise was born on October 22nd, 1988 in Oakland California. He earned a B.S. in Mechanical Engineering from the University of California Berkeley in 2011. He successfully defended his Ph.D. dissertation in Chemical and Biomolecular Engineering at Johns Hopkins University on September 17th, 2018. Dominic considers himself a programmer first, and has been working to find ways to allow computer programs to run outside of traditional electronic computers, in the real physical world.