Probabilistic switching circuits in DNA

Daniel Wilhelm[a], Jehoshua Bruck[a,b], and Lulu Qian[c,d,1]

[a]Computation & Neural Systems, California Institute of Technology, Pasadena, CA 91125; [b]Electrical Engineering, California Institute of Technology, Pasadena, CA 91125; [c]Bioengineering, California Institute of Technology, Pasadena, CA 91125; and [d]Computer Science, California Institute of Technology, Pasadena, CA 91125

Edited by David Baker, University of Washington, Seattle, WA, and approved December 22, 2017 (received for review September 10, 2017)

A natural feature of molecular systems is their inherent stochastic behavior. A fundamental challenge related to the programming of molecular information processing systems is to develop a circuit architecture that controls the stochastic states of individual molecular events. Here we present a systematic implementation of probabilistic switching circuits, using DNA strand displacement reactions. Exploiting the intrinsic stochasticity of molecular interactions, we developed a simple, unbiased DNA switch: An input signal strand binds to the switch and releases an output signal strand with probability one-half. Using this unbiased switch as a molecular building block, we designed DNA circuits that convert an input signal to an output signal with any desired probability. Further, this probability can be switched between 2^n different values by simply varying the presence or absence of n distinct DNA molecules. We demonstrated several DNA circuits that have multiple layers and feedback, including a circuit that converts an input strand to an output strand with eight different probabilities, controlled by the combination of three DNA molecules. These circuits combine the advantages of digital and analog computation: They allow a small number of distinct input molecules to control a diverse signal range of output molecules, while keeping the inputs robust to noise and the outputs at precise values. Moreover, arbitrarily complex circuit behaviors can be implemented with just a single type of molecular building block.

molecular programming | DNA strand displacement circuits | stochasticity | digital and analog computation

A simple but fundamental principle underlying the sophistication of life is that individual cells with the same genome can exhibit different types of behaviors in response to stochastic molecular events, and the fraction of cells at a given state can be precisely regulated, giving rise to complex system behaviors for a collection of cells (1, 2). In many cases, the stochastic events are not unbiased choices with equal probabilities. Instead, specific fractions of cell states are maintained in various circumstances (3). Similar to biological systems, stochastic information processing could also give rise to sophisticated behaviors in engineered molecular systems. Here we aim to understand the engineering principles for controlling the output of molecular circuits with arbitrary probabilities. Because it is simple to generate an equal probability between two choices, like flipping a coin, we ask the following question: Does there exist a molecular circuit architecture that generates an output with any desired probability, and thus any desired fraction of a molecular species, from unbiased molecular events controlled by simple building blocks?

Circuits that are capable of processing molecular information have been developed to control complex behaviors in biological (4–6) and biochemical (7–9) systems. In particular, chemical reaction networks implemented using DNA strand displacement reactions (10) have been proposed as a framework for creating arbitrary chemical kinetics and universal computation (11). However, experimental demonstrations of the theoretical proposal have so far been limited to simple systems with specific functions involving no more than three formal reactions (12, 13). Moreover, in many cases, powerful circuit architectures do not necessarily require the full expressibility of chemical reaction networks. Thus, studies on using simpler and more straightforward DNA strand displacement implementations for certain types of information processing, including for both digital (9, 14) and analog (15, 16) computation, continue to play an important role in the development of biochemical circuits.

Digital signals are discrete high or low concentrations of molecular species, which correspond to binary inputs and outputs that are ON or OFF, respectively. Analog signals are continuous concentrations of molecular species, which correspond to real-value inputs and outputs. Here, instead of idealized analog signals with infinite precision, which can be used to compute beyond the Turing limit (17), we are interested in real-world analog signals with finite precision. It has been articulated that digital computation and analog computation each have distinct advantages and thus should be combined in biological and biochemical circuits (18, 19). For example, digital computation is more robust to noise and analog computation is more efficient under certain circumstances. However, an unresolved challenge is to develop a DNA circuit architecture for generating arbitrary analog outputs controlled by digital inputs. This would allow a small number of distinct input molecules to control a diverse signal range of output molecules, while keeping the inputs robust to noise and the outputs at precise values. More importantly, the systematic constructions of DNA strand displacement circuits have so far been focused only on deterministic computation. For example, in DNA-based logic circuits (7, 9), the same input signals will yield the same output signals, whether in the low–copy-number regime or in bulk.

Here we show a molecular circuit architecture that controls the stochastic states of individual molecular events with any

Significance

Biological organisms exhibit sophisticated control over the stochastic states of individual cells, but the understanding of underlying molecular mechanisms remains incomplete. It has been argued that unbiased choices are easy to achieve, but choices biased with specific probabilities are much harder. These natural phenomena raise an engineering challenge: Does there exist a simple method to program molecular systems that control arbitrary probabilities for individual molecular events? Here we show a molecular circuit architecture, using just a simple DNA strand displacement building block that functions as an unbiased switch, for creating a circuit output with any desired probability. We constructed several DNA circuits with multiple layers and feedback, demonstrating complex molecular information processing that exploits the inherent stochasticity of molecular interactions.

Author contributions: J.B. and L.Q. designed research; D.W. and L.Q. performed research; D.W. and L.Q. analyzed data; and D.W., J.B., and L.Q. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

[1]To whom correspondence should be addressed. Email: luluqian@caltech.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1715926115/-/DCSupplemental.

desired probability: With the same input signals and at the single-molecule level, a circuit will yield either no output or a desired output. Alternatively, a circuit will yield one specific output among a few possible outputs. At the bulk level (i.e., for a collection of molecules), the circuit architecture controls an arbitrary fraction of the input molecules to yield a desired output. Functionally, these circuits enable output signals with analog concentration $I \times p$ generated from a single input with analog concentration I, for arbitrary binary and rational fraction p, which can be controlled by a set of digital signals that are either present or absent.

Our approach is a systematic implementation of probabilistic switching circuits, using DNA strand displacement reactions. Unlike the theory of the original switching circuits proposed by Shannon (20), wherein signals arriving at the input terminal of a deterministic switch always reach the output terminal if the switch is ON and always stop flowing if the switch is OFF, the theory of probabilistic switching circuits allows the signals to flow through a switch with a specified probability (21, 22). Exploiting the intrinsic stochasticity of molecular interactions, in our implementation, each input DNA signal is designed to bind to a DNA switch and release an output signal with a one-half probability. Composing these switches together, arbitrary probabilities for any input signal to yield a circuit output can be implemented with just a single type of DNA building block.

Results

Circuit Design. In a probabilistic switching circuit, any circuit components can be combined in series or parallel (Fig. 1*A*). Each probabilistic switch, or pswitch, is associated with a Bernoulli variable defining the probability that the switch is closed. When it is closed, the input and output terminals are connected and a signal can be propagated; otherwise it is open and the signal cannot flow through. When two pswitches with probabilities p and q are combined in series, a signal can flow through only if both pswitches are closed, and thus the probability is pq. When they are combined in parallel, a signal can flow through if either pswitch is closed, and thus the probability is $1-(1-p)(1-q)=p+q-pq$. An extension of the pswitch is a probabilistic splitter wherein an input terminal is connected to one of two output terminals with probabilities p and $1-p$. Equivalently, the fan-out wires in the parallel construction can be replaced by a splitter and the pswitch by a wire, which enables a simple molecular implementation for splitting signal flow at a junction.

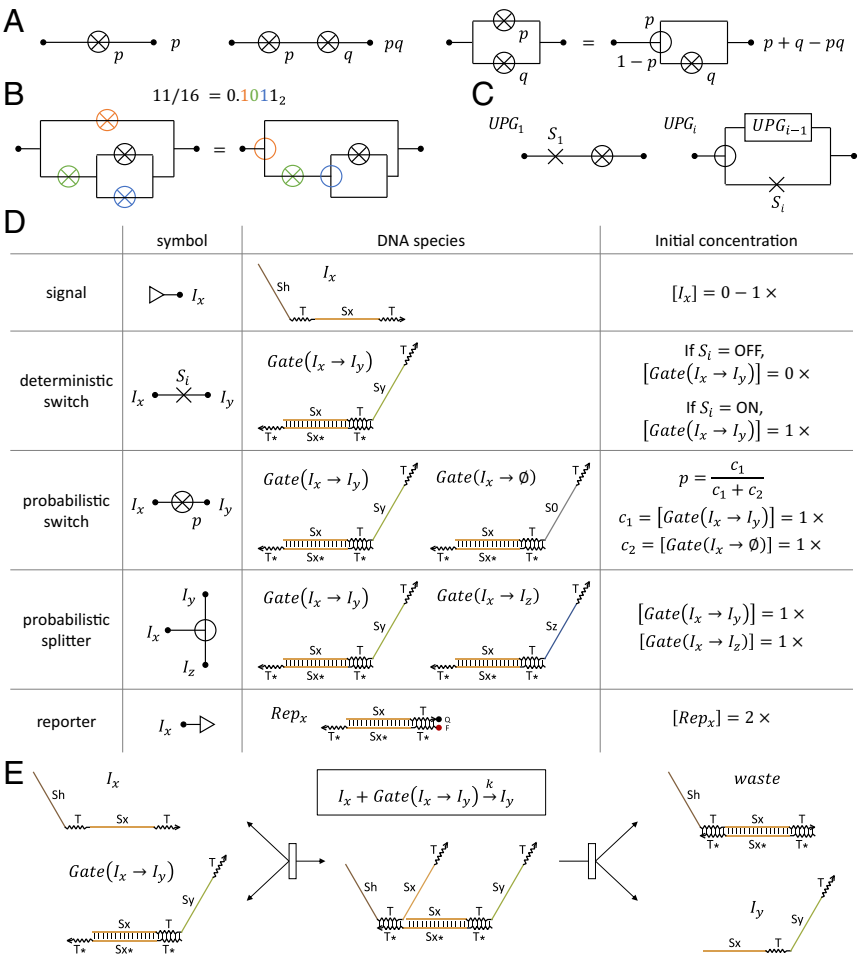

Fig. 1. DNA implementation of probabilistic switching circuits. (*A*) Series and parallel circuit constructions. To the right of each circuit is the probability its terminals are connected, given the probability for each pswitch and splitter. (*B*) An example circuit closed with probability 11/16. Here, $p=1/2$ for all pswitches and splitters. (*C*) A universal probability generator. Binary fractions $0/2^i$ to $(2^i-1)/2^i$ are realized with $S_i \cdots S_1 = 0 \cdots 0$ to $1 \cdots 1$. (*D*) DNA implementation for each circuit component. Squiggled lines indicate short toehold domains and straight lines indicate long branch migration domains in DNA strands, with arrowheads marking their 3′ ends. Asterisks indicate domain complementarity. F indicates a fluorophore and Q indicates a quencher. The term $1\times$ is a standard concentration of, for example, 50 nM. The default probability is $p=1/2$, which is implemented with $c_1=c_2$. (*E*) Reaction mechanism for a signal species interacting with a gate species.

In theory, it has been shown that using pswitches with just probability $1/2$, arbitrary n-bit binary fractions can be realized with n pswitches (21). The construction is quite simple: Read from the least to the most significant bit, add a $1/2$ pswitch in parallel if the bit is 1, and add it in series if the bit is 0 (Fig. 1*B*). Allowing feedback in the circuits, arbitrary rational fraction a/b with $a \leq b \leq 2^n$ can be realized with n splitters (22).

Using $2n$ switches, including deterministic switches, $1/2$ pswitches, and splitters, a circuit that maps n digital inputs to all 2^n n-bit binary fractions can be systematically constructed (21). The circuit is referred to as a universal probability generator (UPG). A one-bit UPG consists of a deterministic switch controlled by signal S_1 and a $1/2$ pswitch (Fig. 1*C*). When S_1 is OFF, the circuit is open and the output is 0. When S_1 is ON, the circuit is closed with $1/2$ probability. An i-bit UPG is recursively constructed by adding a $1/2$ splitter and a deterministic switch to an $(i-1)$-bit UPG. A UPG is functionally equivalent to a digital-to-analog converter, except that the output value $I \times (S_i \cdots S_1)/2^i$ is controlled not only by a set of digital signals S_1–S_i but also by an analog circuit input I.

The three types of switches can be implemented with a single type of DNA molecule (Fig. 1*D*). To test the circuit function, an analog signal and a reporter will be placed at the input and the output terminal, respectively. An arbitrary signal I_x is implemented with a single-stranded DNA species that has a 15-nt history domain (Sh) and two 6-nt toehold domains (T) flanking a 15-nt branch migration domain (Sx). The concentration of I_x corresponds to its analog value.

A deterministic switch is implemented with a partially double-stranded gate species $Gate(I_x \to I_y)$ that has a signal strand I_y with its $5'$ end half bound to a complementary strand, which we refer to as a gate bottom strand. It has an uncovered toehold domain at the $3'$ end. The gate species will be present or absent, depending on whether the switching signal S_i is ON or OFF. If the switch is ON, input signal I_x will be converted to output signal I_y through an irreversible strand displacement reaction (23) (Fig. 1*E*): The input strand first binds to the gate via the uncovered toehold domain. Branch migration occurs when the two Sx domains in the input and output strands compete for binding to the complementary domain on the gate bottom strand. When branch migration proceeds to the $3'$ end of the input, the output strand will be released from the gate and become an active signal.

A probabilistic switch is implemented with two gate species: One is the same as the deterministic switch and the other is $Gate(I_x \to \emptyset)$ that consumes the input signal without generating any active output signals (Fig. 1*D*). The two gate species will compete with each other for interacting with the input strand, and the outcome of the competition depends on the rates of the two reactions, which in turn depend on the concentrations of the gate species and the rate constants. The rate constant of a strand displacement reaction is primarily determined by the standard free energy of the toehold (24), and thus using the same toehold for both reactions will result in roughly the same rate constant, allowing the competition to be simply controlled by the concentrations of the gates. Similarly, a probabilistic splitter is also implemented with two gate species at equal concentration, each generating a distinct output signal with $1/2$ probability (Fig. 1*D*).

A reporter converts an output strand to a fluorescent signal, which can then be measured by a spectrofluorometer. The reporter molecule has two DNA strands, one modified with a fluorophore and the other with a quencher (Fig. 1*D*). It interacts with a signal strand just like the gate does, but upon completion of the irreversible strand displacement reaction the fluorophore will be separated from the quencher, resulting in increased fluorescence.

Simple Circuits. We start the experimental demonstration with a single $1/2$ pswitch and splitter (*SI Appendix*, Fig. S1 *A* and *B*). At first, we designed the second gate species $Gate(I_x \to \emptyset)$ in the pswitch to have no tail. The tail in other gate species contains a branch migration domain and a toehold domain—once the signal is released from the gate, these domains can then participate in reactions with downstream gates. Since no active signal should be produced by the second gate species in the pswitch, having no tail is the simplest way to satisfy that. Unlike the pswitch, the second gate species in the splitter must have a tail to generate another active signal, which was the only difference between the two circuits that we tested. For simplicity, we left the second output of the splitter unconnected to any downstream gate or reporter. With this setup, we expected the two circuits to produce the same amount of output signal, given the same amount of input. However, experimental data showed that the output signal of the pswitch was noticeably less than that of the splitter (*SI Appendix*, Fig. S1*C*). We hypothesized that the uncovered toehold in the tail of the gate species might have reversibly bound to the complementary toehold in the gate bottom strand, forming a loop structure (*SI Appendix*, Fig. S1*D*). In that case, at any given time, only a fraction of the gate species will be able to interact with the input signal as designed, resulting in a slower reaction rate compared with that of the gate species without a tail. Thus, the actual probability for the pswitch was smaller than intended.

To solve this toehold problem, we made two design changes: First, a tail is added to the second gate species in all pswitches, but with a poly-A domain (S0) instead of an active branch migration domain. This way, the two competing gate species are now structurally the same. Second, three distinct toeholds are used instead of one universal toehold (*SI Appendix*, Fig. S1*E*). With a simple rule for toehold assignment, the toehold in the tail of any gate species will be different from that in the bottom strand: Choose the longest pathway from the circuit input to the output and assign two toeholds to each signal species along the pathway, following the order shown in *SI Appendix*, Fig. S1*E*. Trace all other pathways from the circuit output back to the input and assign the remaining toeholds based on the existing ones.

With these changes, we constructed a one-bit UPG (Fig. 2 *A* and *B*). To convert raw fluorescence signal to concentration of output signal, we introduced a postexperiment triggering step that directly generates a reference output signal change which was then used to normalize the data (*SI Appendix*, Fig. S2). To compare data with expected circuit behavior, we simulated the set of strand displacement reactions using mass-action kinetics. Since we designed the three toeholds to have similar binding energy, to simplify the model, we used a single rate constant for all reactions. In agreement with simulations, the circuit produced approximately no output and $1/2$ output when the deterministic switch was OFF and ON, respectively (Fig. 2*C*).

Adding a splitter and a second deterministic switch, we then constructed a two-bit UPG (Fig. 2 *D* and *E*). The circuit correctly produced the expected output for all four combinations of the two digital switching signals (Fig. 2*F*), suggesting that the three types of switches compose well in a multilayer circuit with branches.

To evaluate the predictive power of the simple model, we used the same rate constant estimated from the one-bit UPG experiment to simulate the two-bit UPG: The data and simulations semiquantitatively agreed with each other (Fig. 2*F*). Adjusting the rate constant in the simulations led to a better fit to the data (*SI Appendix*, Fig. S3*B*), which is unsurprising given that an additional toehold sequence (T1) was used in the two-bit UPG. Leak reactions between an upstream and a downstream gate species (or between an upstream gate and a downstream reporter) can be included in simulations to better explain the gradual signal

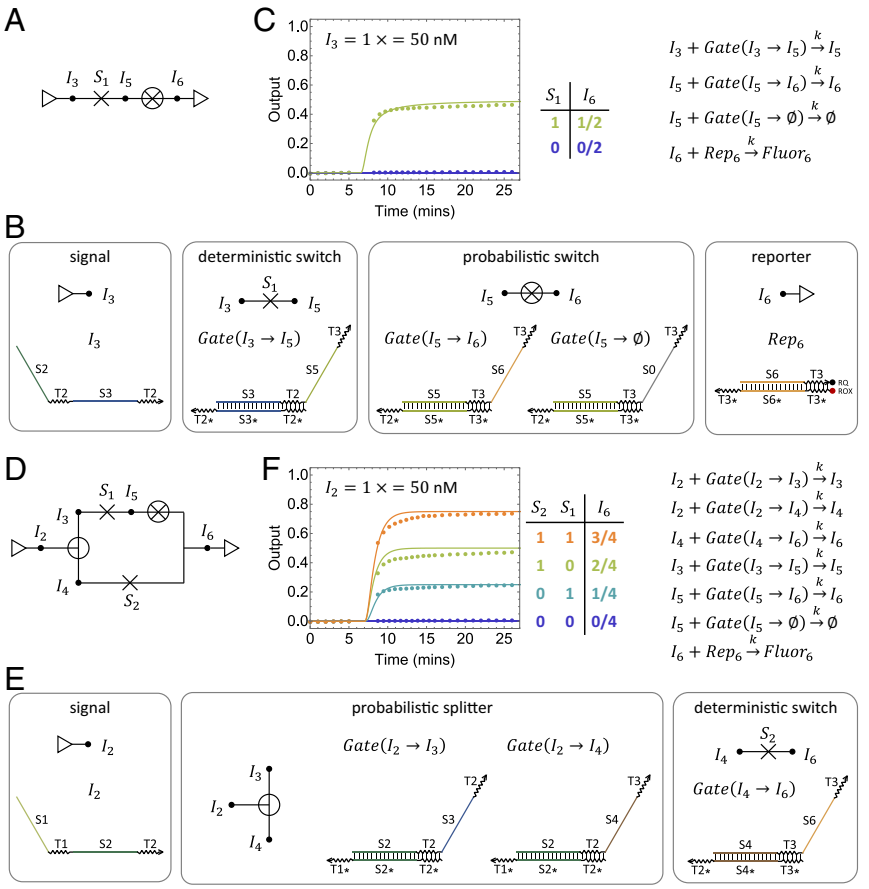

Fig. 2. Simple circuits. (*A*–*C*) Circuit diagram (*A*), DNA species (*B*), and simulations and fluorescence kinetics experiments (*C*) of a one-bit universal probability generator. The error of the DNA circuit is 0.031 ± 0.024, comparing the last data point with the expected circuit output. ROX is the name of the fluorophore and RQ is the name of the quencher used in Rep_6. (*D*–*F*) Circuit diagram (*D*), additional DNA species (*E*), and simulations and fluorescence kinetics experiments (*F*) of a two-bit universal probability generator. The error of the DNA circuit is 0.027 ± 0.010, comparing the last data point with the expected circuit output. Dotted lines are experimental data and solid lines are simulations. Truth tables show expected values of the analog output based on the digital switching signals, and the values are the same as the reaction completion levels shown in simulations at roughly 25 min. Simulations were performed by solving a set of ordinary differential equations derived from the listed reactions, using mass-action kinetics. $k = 6.5 \times 10^5 \cdot \mathrm{M} \cdot \mathrm{s}^{-1}$ was used in all simulations.

increase in the output trajectories (*SI Appendix*, Fig. S3*C*). It is also reasonable to assume that an up to 10% inaccuracy of the input concentration could occur in the experiments, because of pipetting errors as well as the signal loss due to synthesis errors in the DNA strands. With these two modifications in the model, the simulations quantitatively agreed with the data (*SI Appendix*, Fig. S3*C*).

More Complex Circuits. Next, we wanted to understand whether the circuit architecture is robust enough for an increasing circuit size. To explore that, we constructed a three-bit UPG (Fig. 3*A*). This was when we encountered a problem with one of the splitters: It yielded roughly 0.4 instead of the desired 0.5 output (*SI Appendix*, Fig. S4). We hypothesized that either the effective concentration of one gate species in this splitter was 50% higher than that of the other or the rate constant for the input signal interacting with one gate was larger than that with the other, both resulting in one reaction pathway being faster. In either case, the desired circuit behavior should be restored by reducing the concentration of the gate involved in the faster pathway. Indeed, with a $2/3\times$ concentration of the gate in the faster pathway, the circuit produced the desired output for all possible three-bit switching signals (Fig. 3*B*). Because synthesis, concentration, and pipetting errors all could affect the desired molecular behavior (14), it is important that the circuit architecture allows for a simple method to tune individual components and restore the overall circuit function.

Similar to the two-bit UPG, simulations using the simpler model semiquantitatively agreed with the experimental data (Fig. 3*B*), and simulations using the more complex model including leak reactions resulted in a better agreement (*SI Appendix*, Fig. S5). Allowing different rate constants for reactions involving different toeholds and branch migration domains provided an even more ideal fit to the data (*SI Appendix*, Fig. S6).

Finally, we demonstrate the full power of probabilistic switching circuits by constructing a feedback circuit that realizes two rational fractions: $1/3$ and $2/3$ (Fig. 3*C* and *SI Appendix*, Fig. S7*A*). The circuit consists of two splitters in a cascade, with one output of the downstream splitter connected to the input of the upstream splitter. Without feedback, the probability for an input molecule to reach each of the two output terminals is $1/2$ and $1/4$, respectively. With feedback, an input molecule always has $1/4$ probability to follow the loop, and thus the overall probability for it to reach one of the two output terminals is simply adding the probabilities together: $\sum_{n=1}^{\infty} 1/2 \times (1/4)^{n-1} = 2/3$ and $\sum_{n=1}^{\infty} 1/4 \times (1/4)^{n-1} = 1/3$, respectively. Similar to the three-bit UPG, we had to tune down the concentration of one gate species. But after this simple tuning, the circuit produced the expected output both without and with feedback. Again, simulations using the simple model (Fig. 3 *D* and *E*) and the complex

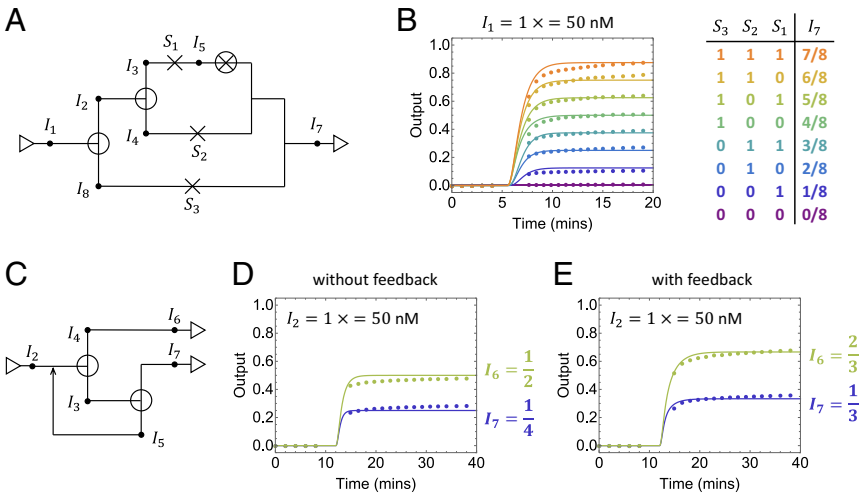

Fig. 3. More complex circuits. (*A* and *B*) Circuit diagram (*A*) and simulations and fluorescence kinetics experiments (*B*) of a three-bit universal probability generator. The error of the DNA circuit is 0.017 ± 0.004, comparing the last data point with the expected circuit output. (*C*–*E*) Circuit diagram (*C*) and simulations and fluorescence kinetics experiments of a circuit without feedback (*D*) and with feedback (*E*) for generating rational fractions. The error of the DNA circuit without and with feedback is 0.027 ± 0.007 and 0.020 ± 0.007, respectively, comparing the last data point with the expected circuit output. Dotted lines are experimental data and solid lines are simulations. To compensate for the observed difference between the two reaction pathways in a splitter (*SI Appendix*, Fig. S4), $Gate(I_1 \rightarrow I_2) = 2/3\times$ instead of $1\times$ was used in the experiments shown in *B*. Similarly, $Gate(I_3 \rightarrow I_7) = 2/3\times$ was used in the experiments shown in *D* and *E*.

model (*SI Appendix*, Fig. S7*B*) semiquantitatively and quantitatively reproduced the experimental data, respectively.

In theory, many classes of probabilistic switching circuits—including all UPGs and some feedback circuits—are surprisingly robust with imperfect building blocks: If the error of each pswitch is bounded by ϵ, the total error of a circuit is bounded by a constant multiple of ϵ, regardless of the circuit size (25). In our experiments, the error of the smallest circuit was 3.1% and that of the largest circuit was 1.7%, and the errors of all other circuits were between these two values (Figs. 2 and 3 legends).

Discussion

We used exactly one type of gate species to construct all three types of switches required for arbitrary probabilistic switching circuits. The structure of the gates is similar to that of seesaw gates (9, 26), except that an additional toehold is included in the signals to make the reactions irreversible. Unlike the seesaw circuit architecture wherein thresholding was required for the overall circuit function and was implemented by competition between a fast and a slower pathway, all reactions in the pswitch circuits require just one rate, which can be close to the maximum rate of DNA strand displacement reactions. As a result, all DNA circuits that we demonstrated yielded the desired output signals in just a few minutes, which were one to two orders of magnitude faster than the seesaw circuits.

The unbiased DNA switch that we developed simply exploits the inherent stochasticity in molecular interactions: If one molecule can react with two different molecules, it will react with the one that it bumps into first through random diffusion. One might ask, Since the equal probability can be realized by equal concentration of the two reactive molecules, why not use different concentrations of different reactive molecules to create a biased probability? Yes, that is possible, but that would not be robust—if any environmental changes resulted in fluctuation of the concentrations, the circuit function would break down. However, if only unbiased choices are involved, the exact concentrations of the reactive molecules may vary, as long as their concentrations remain equal to each other. For example, it should be possible to make the $1/2$ pswitch or splitter in the form of a dimer: Two gate species could be linked together in multiple ways (*SI Appendix*, Fig. S8), and complexes that include both gates could then be gel purified. This way, the concentration of the gates would have to be equal. The fact that arbitrary binary and rational fractions can be realized using just the $1/2$ pswitch and splitter is critical for the possibility of perfecting the building block and enabling even more robust construction of increasingly complex circuits. However, to truly demonstrate a perfectly unbiased switch, further study will be required to explore the different designs and understand their trade-offs.

There are at least two other aspects of the DNA-based probabilistic switching circuits that merit further study: First, we used bulk fluorescence kinetics experiments to observe the circuit behavior, but in principle, the stochastic states of individual molecular events could be observed in droplets (27, 28), on microparticle surfaces (29), or on DNA origami surfaces (30, 31). Second, we used a simple clamp design to reduce undesired leak reactions between circuit components (*Materials and Methods*), which was not very effective. Scaling up the complexity of these circuits will require a more advanced design to eliminate undesired reactions, for example by using leakless mechanisms (32).

DNA-based probabilistic switching circuits could be directly composed together with previously developed DNA-based analog circuits (11, 13). With a revised design of deterministic switches (*SI Appendix*, Fig. S9), the circuits could also be composed together with previously developed DNA-based logic circuits (9) and neural networks (26). Integrating multiple circuit architectures would enable more powerful molecular information processing in complex biochemical environments, while each part of the circuit could be optimized for a specific task, combining robustness and efficiency. Moreover, what we have shown here could have implications for natural molecular systems in biology and chemistry, as well as for engineered molecular systems in material science and medicine: Any desired probability of an individual molecular event, leading to any desired fraction of molecular species at a specific state, could be generated from just one type of molecular building block that generates an equal probability between two choices, as simple as flipping a coin. Finally, it is now conceivable to create engineered molecular systems with programmable stochastic behavior in simple and compartmentalized

environments, and the communication among these simple systems could give rise to complex global behavior.

Materials and Methods

All DNA sequences are listed in *SI Appendix*, Tables S1 and S2. A 1-nt clamp was used in all gate bottom strands to reduce undesired gate–gate interactions. DNA oligonucleotides were purchased from Integrated DNA Technologies. All gate and reporter species were annealed at 20 μM in 1× TE buffer with 12.5 mM Mg^{2+}. After annealing, the gate species were purified using 15% PAGE. Fluorescence kinetics experiments were performed at 25 °C.

ACKNOWLEDGMENTS. We thank D. Y. Zhang and E. Winfree for discussions. D.W., J.B., and L.Q. were supported by an NSF Expedition in Computing grant (0832824). L.Q. was also supported by a Career Award at the Scientific Interface from the Burroughs Wellcome Fund (1010684) and a Faculty Early Career Development Award from the NSF (1351081).

1. Balaban NQ, Merrin J, Chait R, Kowalik L, Leibler S (2004) Bacterial persistence as a phenotypic switch. *Science* 305:1622–1625.
2. Bintu L, et al. (2016) Dynamics of epigenetic regulation at the single-cell level. *Science* 351:720–724.
3. Losick R, Desplan C (2008) Stochasticity and cell fate. *Science* 320:65–68.
4. Daniel R, Rubens JR, Sarpeshkar R, Lu TK (2013) Synthetic analog computation in living cells. *Nature* 497:619–623.
5. Nielsen AA, et al. (2016) Genetic circuit design automation. *Science* 352:aac7341.
6. Gander MW, Vrana JD, Voje WE, Carothers JM, Klavins E (2017) Digital logic circuits in yeast with CRISPR-dCas9 NOR gates. *Nat Commun* 8:15459.
7. Seelig G, Soloveichik D, Zhang DY, Winfree E (2006) Enzyme-free nucleic acid logic circuits. *Science* 314:1585–1588.
8. Zhang DY, Turberfield AJ, Yurke B, Winfree E (2007) Engineering entropy-driven reactions and networks catalyzed by DNA. *Science* 318:1121–1125.
9. Qian L, Winfree E (2011) Scaling up digital circuit computation with DNA strand displacement cascades. *Science* 332:1196–1201.
10. Zhang DY, Seelig G (2011) Dynamic DNA nanotechnology using strand-displacement reactions. *Nat Chem* 3:103–113.
11. Soloveichik D, Seelig G, Winfree E (2010) DNA as a universal substrate for chemical kinetics. *Proc Natl Acad Sci USA* 107:5393–5398.
12. Chen YJ, et al. (2013) Programmable chemical controllers made from DNA. *Nat Nanotechnol* 8:755–762.
13. Srinivas N, Parkin J, Seelig G, Winfree E, Soloveichik D (2017) Enzyme-free nucleic acid dynamical systems. *Science* 358:eaal2052.
14. Thubagere AJ, et al. (2017) Compiler-aided systematic construction of large-scale DNA strand displacement circuits using unpurified components. *Nat Commun* 8: 14373.
15. Oishi K, Klavins E (2011) Biomolecular implementation of linear I/O systems. *IET Syst Biol* 5:252–260.
16. Song T, Garg S, Mokhtar R, Bui H, Reif J (2016) Analog computation by DNA strand displacement circuits. *ACS Synth Biol* 5:898–912.
17. Siegelmann HT (1995) Computation beyond the Turing limit. *Science* 268:545–548.
18. Lu TK, Khalil AS, Collins JJ (2009) Next-generation synthetic gene networks. *Nat Biotechnol* 27:1139–1150.
19. Sarpeshkar R (2014) Analog synthetic biology. *Philos Trans R Soc A* 372:20130110.
20. Shannon CE (1938) A symbolic analysis of relay and switching circuits. *Trans Am Inst Electr Eng* 57:713–723.
21. Wilhelm D, Bruck J (2008) *Stochastic Switching Circuit Synthesis* (IEEE International Symposium on Information Theory, Piscataway, NJ), pp 1388–1392.
22. Zhou H, Chen HL, Bruck J (2014) Synthesis of stochastic flow networks. *IEEE Trans Comput* 63:1234–1247.
23. Yurke B, Turberfield AJ, Mills AP, Simmel FC, Neumann JL (2000) A DNA-fuelled molecular machine made of DNA. *Nature* 406:605–608.
24. Turberfield AJ, et al. (2003) DNA fuel for free-running nanomachines. *Phys Rev Lett* 90:118102.
25. Loh PL, Zhou H, Bruck J (2009) The robustness of stochastic switching networks. *IEEE International Symposium on Information Theory* 3:2066–2070.
26. Qian L, Winfree E, Bruck J (2011) Neural network computation with DNA strand displacement cascades. *Nature* 475:368–372.
27. Weitz M, et al. (2014) Diversity in the dynamical behaviour of a compartmentalized programmable biochemical oscillator. *Nat Chem* 6:295–302.
28. Genot AJ, et al. (2016) High-resolution mapping of bifurcations in nonlinear biochemical circuits. *Nat Chem* 8:760–767.
29. Gines G, et al. (2017) Microscopic agents programmed by DNA circuits. *Nat Nanotechnol* 12:351–359.
30. Rothemund PW (2006) Folding DNA to create nanoscale shapes and patterns. *Nature* 440:297–302.
31. Scheible MB, Pardatscher G, Kuzyk A, Simmel FC (2014) Single molecule characterization of DNA binding and strand displacement reactions on lithographic DNA origami microarrays. *Nano Lett* 14:1627–1633.
32. Thachuk C, Winfree E, Soloveichik D (2015) Leakless DNA strand displacement systems. *Lecture Notes Comput Sci* 9211:133–153.