Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades

Lulu Qian*, et al.*

Science **332**, 1196 (2011);
DOI: 10.1126/science.1200520

This copy is for your personal, non-commercial use only.

If you wish to distribute this article to others, you can order high-quality copies for your colleagues, clients, or customers by clicking here.

Permission to republish or repurpose articles or portions of articles can be obtained by following the guidelines here.

The following resources related to this article are available online at www.sciencemag.org (this infomation is current as of February 1, 2012):

Updated information and services, including high-resolution figures, can be found in the online version of this article at:
http://www.sciencemag.org/content/332/6034/1196.full.html

Supporting Online Material can be found at:
http://www.sciencemag.org/content/suppl/2011/06/02/332.6034.1196.DC1.html

A list of selected additional articles on the Science Web sites **related to this article** can be found at:
http://www.sciencemag.org/content/332/6034/1196.full.html#related

This article **cites 27 articles**, 10 of which can be accessed free:
http://www.sciencemag.org/content/332/6034/1196.full.html#ref-list-1

This article has been **cited by** 8 articles hosted by HighWire Press; see:
http://www.sciencemag.org/content/332/6034/1196.full.html#related-urls

This article appears in the following **subject collections:**
Biochemistry
http://www.sciencemag.org/cgi/collection/biochem
Computers, Mathematics
http://www.sciencemag.org/cgi/collection/comp_math

Science (print ISSN 0036-8075; online ISSN 1095-9203) is published weekly, except the last week in December, by the American Association for the Advancement of Science, 1200 New York Avenue NW, Washington, DC 20005. Copyright 2011 by the American Association for the Advancement of Science; all rights reserved. The title *Science* is a registered trademark of AAAS.

References and Notes

1. A. S. Kondrashov, *Nature* **369**, 99 (1994).
2. H. A. Orr, M. Turelli, *Evolution* **55**, 1085 (2001).
3. D. M. Weinreich, R. A. Watson, L. Chao, *Evolution* **59**, 1165 (2005).
4. S. Kryazhimskiy, G. Tkacik, J. B. Plotkin, *Proc. Natl. Acad. Sci. U.S.A.* **106**, 18638 (2009).
5. J. A. Draghi, T. L. Parsons, G. P. Wagner, J. B. Plotkin, *Nature* **463**, 353 (2010).
6. S. F. Elena, R. E. Lenski, *Nature* **390**, 395 (1997).
7. M. Lunzer, S. P. Miller, R. Felsheim, A. M. Dean, *Science* **310**, 499 (2005).
8. R. Sanjuán, J. M. Cuevas, A. Moya, S. F. Elena, *Genetics* **170**, 1001 (2005).
9. D. M. Weinreich, N. F. Delaney, M. A. Depristo, D. L. Hartl, *Science* **312**, 111 (2006).
10. L. Jasnos, R. Korona, *Nat. Genet.* **39**, 550 (2007).
11. J. A. G. M. de Visser, S. C. Park, J. Krug, *Am. Nat.* **174** (suppl. 1), S15 (2009).
12. J. da Silva, M. Coetzer, R. Nedellec, C. Pastore, D. E. Mosier, *Genetics* **185**, 293 (2010).
13. K. M. Pepin, H. A. Wichman, *Evolution* **61**, 1710 (2007).
14. L. Jasnos, K. Tomala, D. Paczesniak, R. Korona, *Genetics* **178**, 2105 (2008).
15. R. Sanjuán, S. F. Elena, *Proc. Natl. Acad. Sci. U.S.A.* **103**, 14402 (2006).
16. R. Korona, C. H. Nakatsu, L. J. Forney, R. E. Lenski, *Proc. Natl. Acad. Sci. U.S.A.* **91**, 9037 (1994).
17. C. L. Burch, L. Chao, *Nature* **406**, 625 (2000).
18. F. B.-G. Moore, D. E. Rozen, R. E. Lenski, *Proc. Biol. Sci.* **267**, 515 (2000).
19. R. Montville, R. Froissart, S. K. Remold, O. Tenaillon, P. E. Turner, *PLoS Biol.* **3**, e381 (2005).
20. R. Sanjuán, J. M. Cuevas, V. Furió, E. C. Holmes, A. Moya, *PLoS Genet.* **3**, e93 (2007).
21. O. K. Silander, O. Tenaillon, L. Chao, *PLoS Biol.* **5**, e94 (2007).
22. R. E. Lenski, M. R. Rose, S. C. Simpson, S. C. Tadler, *Am. Nat.* **138**, 1315 (1991).
23. M. R. Goddard, H. C. J. Godfray, A. Burt, *Nature* **434**, 636 (2005).
24. J. E. Barrick *et al.*, *Nature* **461**, 1243 (2009).
25. S. E. Schoustra, T. Bataillon, D. R. Gifford, R. Kassen, *PLoS Biol.* **7**, e1000250 (2009).
26. P. J. Gerrish, R. E. Lenski, *Genetica* **102-103**, 127 (1998).
27. V. S. Cooper, R. E. Lenski, *Nature* **407**, 736 (2000).
28. R. E. Lenski, M. Travisano, *Proc. Natl. Acad. Sci. U.S.A.* **91**, 6808 (1994).
29. T. F. Cooper, D. E. Rozen, R. E. Lenski, *Proc. Natl. Acad. Sci. U.S.A.* **100**, 1072 (2003).
30. R. Woods, D. Schneider, C. L. Winkworth, M. A. Riley, R. E. Lenski, *Proc. Natl. Acad. Sci. U.S.A.* **103**, 9107 (2006).
31. M. Travisano, F. Vasi, R. E. Lenski, *Evolution* **49**, 189 (1995).
32. Z. D. Blount, C. Z. Borland, R. E. Lenski, *Proc. Natl. Acad. Sci. U.S.A.* **105**, 7899 (2008).
33. R. J. Woods *et al.*, *Science* **331**, 1433 (2011).
34. J. A. G. M. de Visser, R. E. Lenski, *BMC Evol. Biol.* **2**, 19 (2002).
35. Materials and methods are available as supporting material on *Science* Online.
36. H. A. Orr, *Genetics* **163**, 1519 (2003).
37. H.-H. Chou, H.-C. Chiu, N. F. Delaney, D. Segrè, C. J. Marx, *Science* **332**, 1190 (2011).

Acknowledgments: This work was supported by grants from the NSF (DEB-1019989 to R.E.L. and DEB-0844355 to T.F.C.), the James S. McDonnell Foundation (220020174 to T.F.C.), the Agence Nationale de la Recherche (Program Génomique, Grant ANR-08-GENM-023-001 to D.S.) and the Defense Advanced Research Projects Agency "Fun Bio" Program (HR0011-09-1-0055 to R.E.L. and T.F.C.). We thank R. Azevedo, T. Paixão, and D. Stoebel for valuable discussions and helpful comments on the manuscript. R.E.L. will make the ancestral and evolved strains used in this study available to qualified recipients, subject to completion of a material transfer agreement that can be found at http://technologies.msu.edu/forms.html. T.F.C. will make the strains constructed in this study available to qualified recipients. Competition experiment counts, summary input data, and analysis scripts that pertain to the experiments and analyses reported in this paper have been deposited at http://dx.doi.org/10.5061/dryad.5rv40.

Supporting Online Material

www.sciencemag.org/cgi/content/full/332/6034/1193/DC1
Materials and Methods
Figs. S1 to S6
Tables S1 to S4
References and Notes

3 February 2011; accepted 25 April 2011
10.1126/science.1203801

Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades

Lulu Qian[1] and Erik Winfree[1,2,3]*

To construct sophisticated biochemical circuits from scratch, one needs to understand how simple the building blocks can be and how robustly such circuits can scale up. Using a simple DNA reaction mechanism based on a reversible strand displacement process, we experimentally demonstrated several digital logic circuits, culminating in a four-bit square-root circuit that comprises 130 DNA strands. These multilayer circuits include thresholding and catalysis within every logical operation to perform digital signal restoration, which enables fast and reliable function in large circuits with roughly constant switching time and linear signal propagation delays. The design naturally incorporates other crucial elements for large-scale circuitry, such as general debugging tools, parallel circuit preparation, and an abstraction hierarchy supported by an automated circuit compiler.

The power and mystery of life is entangled within the information processing at the heart of all cellular machinery. Engineering molecular information processing systems may allow us to tap into that power and elucidate principles that will help us to understand and appreciate the mystery. DNA is an excellent engineering material for biochemical circuits because its biological nature supports technological applications in vivo, its easy chemical synthesis facilitates practical experiments in vitro, its combinatorial structure provides sufficient sequence design space, and the Watson-Crick complementarity principle enables predictable molecular behavior.

DNA has been used as a computing substrate since the first demonstration of solving a seven-city Hamiltonian path problem in 1994 (*1*) and has evolved away from competing with silicon to embedding control within molecular systems. Although DNA automata can be built with deoxyribozymes (*2*, *3*) or with restriction enzymes (*4*), the introduction of toehold-mediated DNA strand displacement enabled enzyme-free DNA machinery that is automated by hybridization alone (*5*–*8*). A DNA strand can serve as a signal when it is free, but is inhibited when it is bound to a complementary strand. A single-stranded DNA signal can first bind to a partially double-stranded complex by a single-stranded domain called a toehold, then release the originally bound strand after branch migration has occurred. Thus, an output signal can be activated upon the arrival of an input signal, and the reaction rate can be controlled by the length of the toehold. This principle has inspired the development of a rich theory (*9*, *10*) and practice (*11*–*13*) of DNA strand displacement circuits, resulting in a wide range of applications such as medical therapeutics in vivo (*14*), molecular instruments in situ (*15*), and biomedical diagnostics in vitro (*16*). To date, the largest digital circuit built with DNA strand displacement cascades involved 12 initial DNA species (*11*). However, their logic gates were constructed with multistranded DNA complexes, challenging sequence design constraints were required, and signal restoration occurred only at the circuit output, perhaps explaining why the performance decayed surprisingly with scale.

To create a scalable DNA circuit architecture, we proposed (*17*) a simple DNA gate motif—a "seesaw" gate—that makes use of a reversible strand displacement reaction based on the principle of toehold exchange (*8*, *12*). In this context, seesawing is the reversible reaction that exchanges

---

[1] Bioengineering, California Institute of Technology, Pasadena, CA 91125, USA. [2] Computer Science, California Institute of Technology, Pasadena, CA 91125, USA. [3] Computation and Neural Systems, California Institute of Technology, Pasadena, CA 91125, USA.

*To whom correspondence should be addressed. E-mail: winfree@caltech.edu

the activity of DNA signals; a pair of seesawing steps completes a catalytic cycle, allowing signal amplification and signal isolation. A pair of seesaw gates can perform AND or OR operation, sufficient for universal Boolean function evaluation using dual-rail logic (*18*). A robust digital abstraction is maintained by embedding thresholding and catalysis into every logic operation to clean up signal degradation. With the use of plug-and-play molecular components, gates can be easily wired into circuits with arbitrary numbers of inputs (fan-in) and outputs (fan-out) at each gate, and they can be reconfigured to perform an AND or OR logic function through simple concentration adjustments. DNA sequence design is straightforward because of the independence of strand domains. The simplicity of gate structures makes parallel DNA synthesis and circuit preparation plausible. We present all circuits using a formal abstraction that concisely defines the DNA species and their initial states, thus determining the circuit wiring, logical function, and temporal behavior. The size of the circuits implemented here with the seesaw architecture is larger than any previous strand displacement circuit, as measured by the number of initial DNA species in the circuit, by at least a factor of 5.

In the seesaw abstraction, each DNA gate is represented by a two-sided node (Fig. 1A and fig. S1, A and B). Each DNA signal is

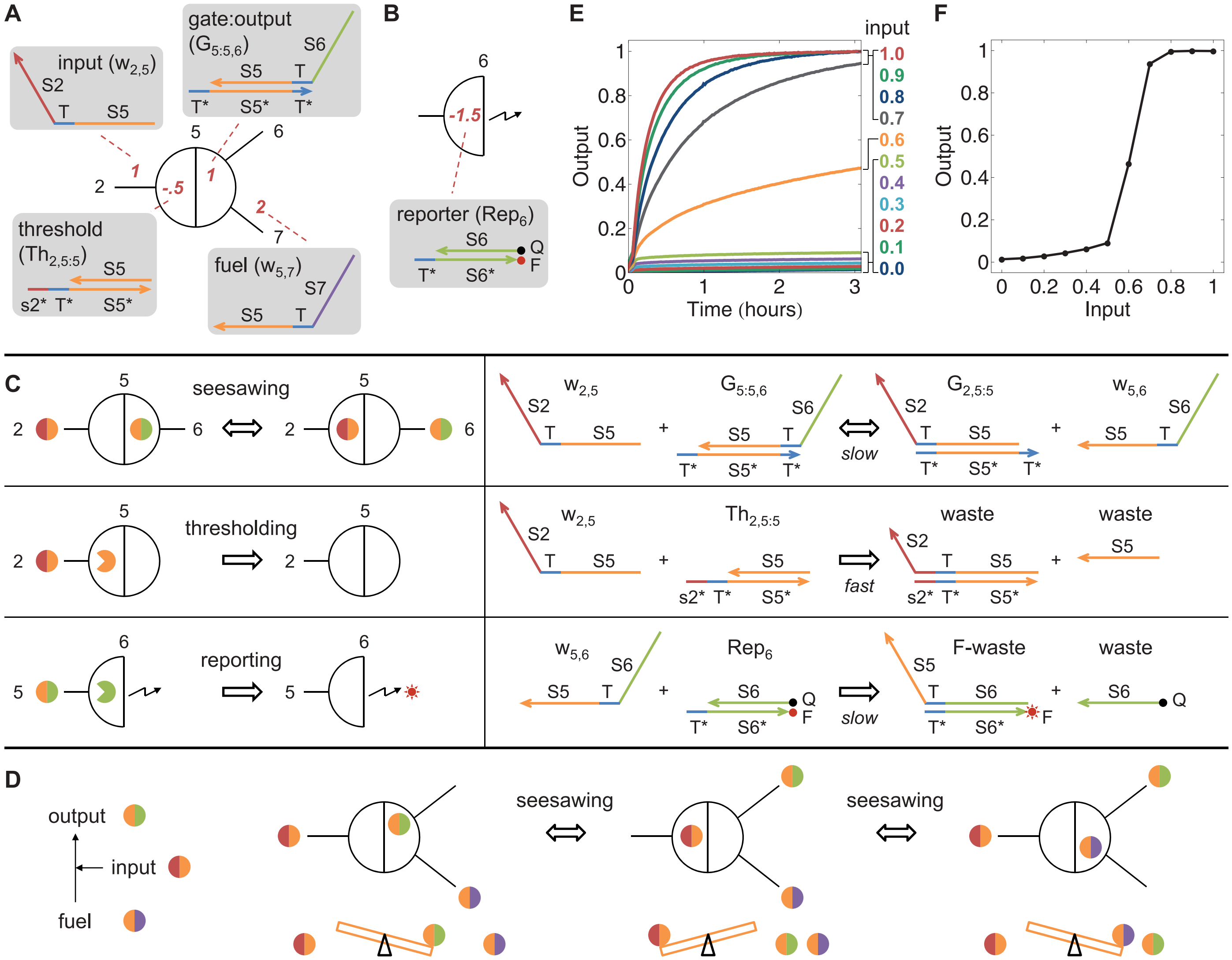


Fig. 1. The seesaw gate motif and its DNA implementation. (**A**) Abstract diagram for a gate. Black numbers indicate identities of nodes (or interfaces to those nodes in a network). Red numbers within the nodes or on the wires indicate relative concentrations of different initial DNA species. Each species plays a specific role (e.g., input) within a gate and has a unique name (e.g., $w_{2,5}$) within a network. Colored lines represent DNA strands at the domain level, with arrowheads marking their 3′ ends and colors indicating distinct DNA sequences. *S2*, *S5*, and *S6* are long (15-nucleotide) recognition domains corresponding to nodes 2, 5, and 6; *S7* does not interact with other nodes in the network but preserves the uniform format of a signal strand. *T* is a short (5-nucleotide) toehold domain; *T** is the Watson-Crick complement of *T*, etc.; *s2** is the first few nucleotides of *S2** from the 3′ end. (**B**) Abstract diagram for a reporter; F and Q denote fluorophore and quencher, respectively. (**C**) Three basic reaction mechanisms involved in a seesaw network: seesawing, thresholding, and reporting. Solid circles with two colors indicate signal strands that have two sides. Colored pac-men indicate threshold or reporter complexes. $w_{2,5}$ is the signal strand that connects gates 2 and 5; $G_{5:5,6}$ is signal strand $w_{5,6}$ bound to gate 5; $Th_{2,5:5}$ is the threshold that absorbs $w_{2,5}$ when it arrives at gate 5; and *Rep6* is the reporter that absorbs $w_{i,6}$ and generates fluorescence signal for any *i*. (**D**) One cycle of a seesaw catalytic reaction. (**E**) Kinetics experiments of the seesaw DNA catalyst with a threshold. Threshold complex, gate:output complex, fuel strand, and reporter complex were mixed in solution with relative concentrations of 0.5×, 1×, 2×, and 1.5×, respectively (standard concentration 1× = 100 nM). Input strands were then added at 0.0× to 1.0× in increments of 0.1×. Sequences of strands are listed in tables S2 and S3, circuit 2. Experiments were performed at 20°C in Tris-acetate-EDTA buffer containing 12.5 mM Mg^{2+}. Output signals were inferred by fluorescence signals normalized to the maximum completion level. (**F**) Input versus output plot of (E). The output at ~3 hours is replotted against the initial input.

represented by a wire. Each side of the node can be connected to any number of wires. Each wire connects two different sides of two nodes. Each red number indicates one DNA species with its initial relative concentration: Each number on a wire corresponds to a free signal strand; each number within a node at the end of a wire corresponds to a bound signal strand (positive number) or a threshold that absorbs a signal when it arrives at the gate (negative number). A reporter that transforms a DNA signal into a fluorescence signal is represented by half a node with a zigzag arrow (Fig. 1B), with its initial relative concentration written similar to a threshold.

Each signal is a single-stranded DNA molecule that has two recognition domains identifying the two gates it connects, one on either side of a central toehold domain. Each gate is associated with a gate base strand that has (the complement of) one recognition domain flanked by two toehold domains. When a signal strand is bound to a gate, it forms a gate:signal complex with the gate's base strand. At any given moment (not counting the transient states during reactions shown in fig. S1C), a gate base strand always has a signal strand bound to one side, leaving the toehold on the other side uncovered.

There are three basic reactions involved in a seesaw network (Fig. 1C and fig. S1C). The first one is seesawing: A free signal on one side of a gate can release a signal bound on the other side of the gate by toehold-mediated strand displacement. The process starts with the free signal strand (e.g., $w_{2,5}$) hybridizing to the gate:signal complex (e.g., $G_{5:5,6}$) at the uncovered toehold domain (e.g., T^*) and then undergoing branch migration through the recognition domain (e.g., $S5$). The previously bound signal will fall off when it is attached to the gate base strand only by the short toehold. The resulting gate:signal complex (e.g., $G_{2,5:5}$) will have an uncovered toehold on the other side, and therefore the now-free signal (e.g., $w_{5,6}$) can reverse the process symmetrically. The second reaction is thresholding: A threshold species associated with a gate and an impinging signal can react with the signal by means of a longer toehold (e.g., $s2^*T^*$), producing only inert waste species that have no exposed toehold. Thresholding is much faster than seesawing because the toehold-mediated strand displacement rate grows exponentially with toehold length for short toeholds (*7*, *8*). As a result, seesawing effectively only happens when the input signal exceeds the threshold. The third reaction is reporting: A reporter species similar to a threshold, but modified with a fluorophore and quencher pair, can absorb an impinging signal while generating a fluorescence signal. Unlike thresholding, reporting does not compete with seesawing, and it therefore does not require a longer toehold.

DNA signals can play different roles such as input (signals that arrive at a gate), output (signals that are produced by a gate), and fuel (signals that help to catalytically produce the output). One seesaw gate with a few wires can create a catalytic cycle in which input transforms free fuel into free output without being consumed in the process (Fig. 1D and fig. S1, B and C). Initially, the output signal is bound to the right side of the gate; the input and fuel signals are free (in our analogy, the output is riding on the right side of the seesaw board; the input and fuel are wandering around). The input signal first releases the output signal and binds to the gate instead (the input jumps onto the left side of the board and makes the output jump off). The fuel signal then displaces the input signal by binding to the gate in the same way (the fuel pushes off the input). A catalytic cycle has been completed. In general, a free signal on one side of a seesaw gate can catalyze the exchange of signals on the other side, and this exchange will not happen without the catalyst. These reactions are driven forward by the entropy of equilibration for the seesawing reactions. A small amount of free input can catalyze the release of a large amount of free output (fig. S2).

Thresholding can be directly combined with a seesaw catalyst to support a digital abstraction—which is the basic principle underlying digital logic in electronics—by pushing the intrinsically analog signal toward either the ideal ON or OFF value. Fluorescence kinetics experiments (Fig. 1E) demonstrated the circuit in Fig. 1A connected to the reporter in Fig. 1B. The input-versus-output relationship (plotted in Fig. 1F) reveals a sharp threshold, ideal for signal restoration.

A cascade of two seesaw gates can compute the logic function OR or AND. To explain this, we introduce two composable seesaw components for digital circuits. We first define the gross production of signal X as the total amount eventually released from the gate:

$$\langle X\rangle = \int_0^{+\infty} X^{\mathrm{prod}}(t)dt \qquad (1)$$

Motivated by sequence design constraints (figs. S3 and S4), we then define two types of feedforward seesaw gates, each assuming an irreversible downstream drain. The first type is called an amplifying gate. It has a threshold and fuel. If the gross production of its input is greater than the initial amount of threshold, the output will keep being released catalytically until it reaches the maximum, which is the initial amount of bound

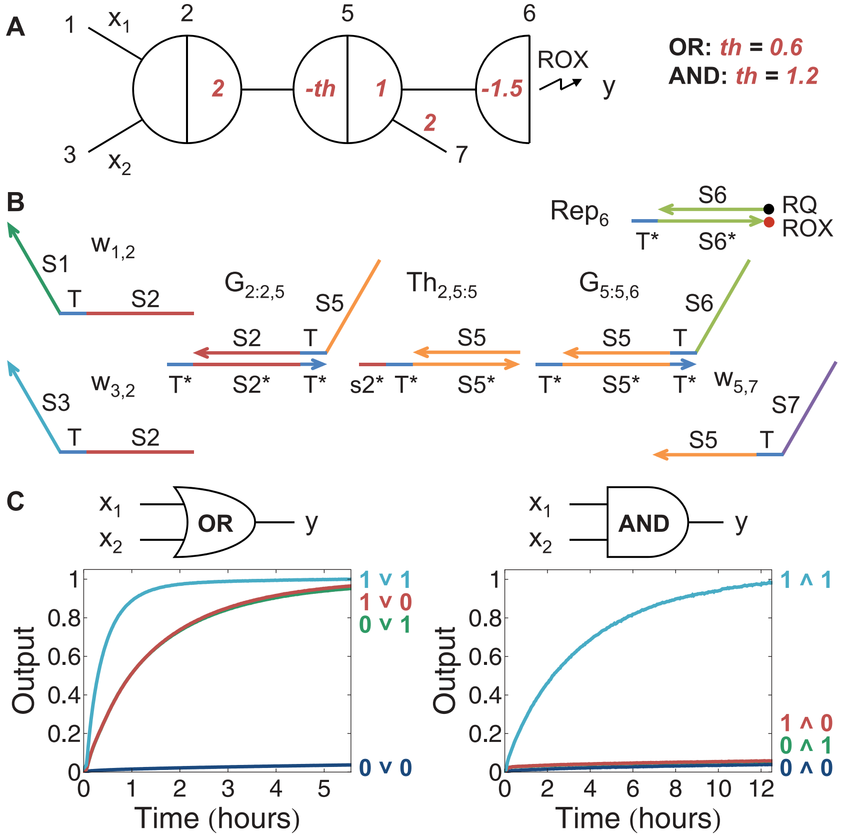
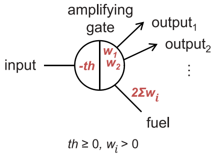


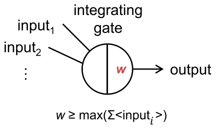
Fig. 2. Digital logic gates implemented with the seesaw DNA motif. (**A**) Abstract diagram of a seesaw circuit that computes either OR or AND, depending on the initial concentration of the threshold. Input signals x_1 ($w_{1,2}$) and x_2 ($w_{3,2}$) are summed together at gate 2 and, if they exceed the threshold, are amplified by gate 5 to generate output signal y ($w_{5,6}$), which is reported by the ROX fluorophore in reporter 6. (**B**) Domain-level DNA implementation of the two-input AND or OR gate. (**C**) Kinetics experiments. Input strands were at 0.1× (0, logic OFF) or 0.9× (1, logic ON), where 1× = 100 nM. Sequences of strands are listed in tables S2 and S3, circuit 3. Experiments were performed at 20°C.

output; otherwise, the output remains at zero. An amplifying gate can support multiple outputs. This is simply done by adding one bound output signal for each output wire. These gate:output complexes will have the same gate base strand bound by signal strands with different right-side recognition domains to connect to different downstream gates. To sufficiently drive the release of all outputs, the initial amount of free fuel will be twice the sum of all initially bound outputs:



$$\langle \text{output}_1 \rangle = \begin{cases} w_1 & \text{if } \langle \text{input} \rangle > th \\ 0 & \text{if } \langle \text{input} \rangle \leq th \end{cases}$$
$$\langle \text{output}_2 \rangle = \begin{cases} w_2 & \text{if } \langle \text{input} \rangle > th \\ 0 & \text{if } \langle \text{input} \rangle \leq th \end{cases}$$
$$\vdots \qquad (2)$$

The second type is called an integrating gate. It has no threshold or fuel. The output is released stoichiometrically with the input. An integrating gate can support multiple inputs. This is simply done by adding one input signal for each input wire. These input strands will have the same right-side recognition domain but different left-side recognition domains to connect to different upstream gates. With multiple inputs, the output will be the sum of all inputs. To ensure that all free inputs can be transformed into free output, the initial amount of bound output must be at least the maximum sum of all inputs that can possibly arrive:



$$\langle \text{output} \rangle = \langle \text{input}_1 \rangle + \langle \text{input}_2 \rangle + \cdots \qquad (3)$$

An integrating gate followed by an amplifying gate can compute either OR or AND (Fig. 2A). A two-input OR gate will have an integrating gate that outputs the sum of the two inputs. The downstream amplifying gate will output 1 when the sum is greater than 0.6 and will output 0 otherwise. In practice, the outputs will not be exactly 0 or 1 because of spurious or incomplete reactions, so we must ensure that logic gates will function correctly even with imperfect inputs. Assuming a digital abstraction where OFF signals may be in the range 0 to 0.2 and ON signals in the range 0.8 to 1, we see that only when both inputs are OFF can the output remain OFF. Changing the threshold from 0.6 to 1.2 computes AND. In this case, only when both inputs are ON can the sum exceed the threshold and catalyze the output to be ON.

In kinetics experiments, a reporter gate was used to provide an irreversible drain and to transform the output into a fluorescence signal. With exactly the same set of molecules (Fig. 2B) but different initial concentrations of the threshold, OR and AND computations were demonstrated (Fig. 2C). The AND gate behaved slower than the OR gate because the initial concentration of the threshold was higher, so it took longer for the upstream signal to exceed the threshold. Thanks to the thresholding and catalysis, even when the inputs were imperfect (0.1 was used for OFF inputs and 0.9 for ON inputs), the outputs still achieved ideal OFF and ON signal levels, preserving the digital abstraction.

Two-layer cascading was demonstrated with OR-OR, AND-OR, OR-AND, and AND-AND

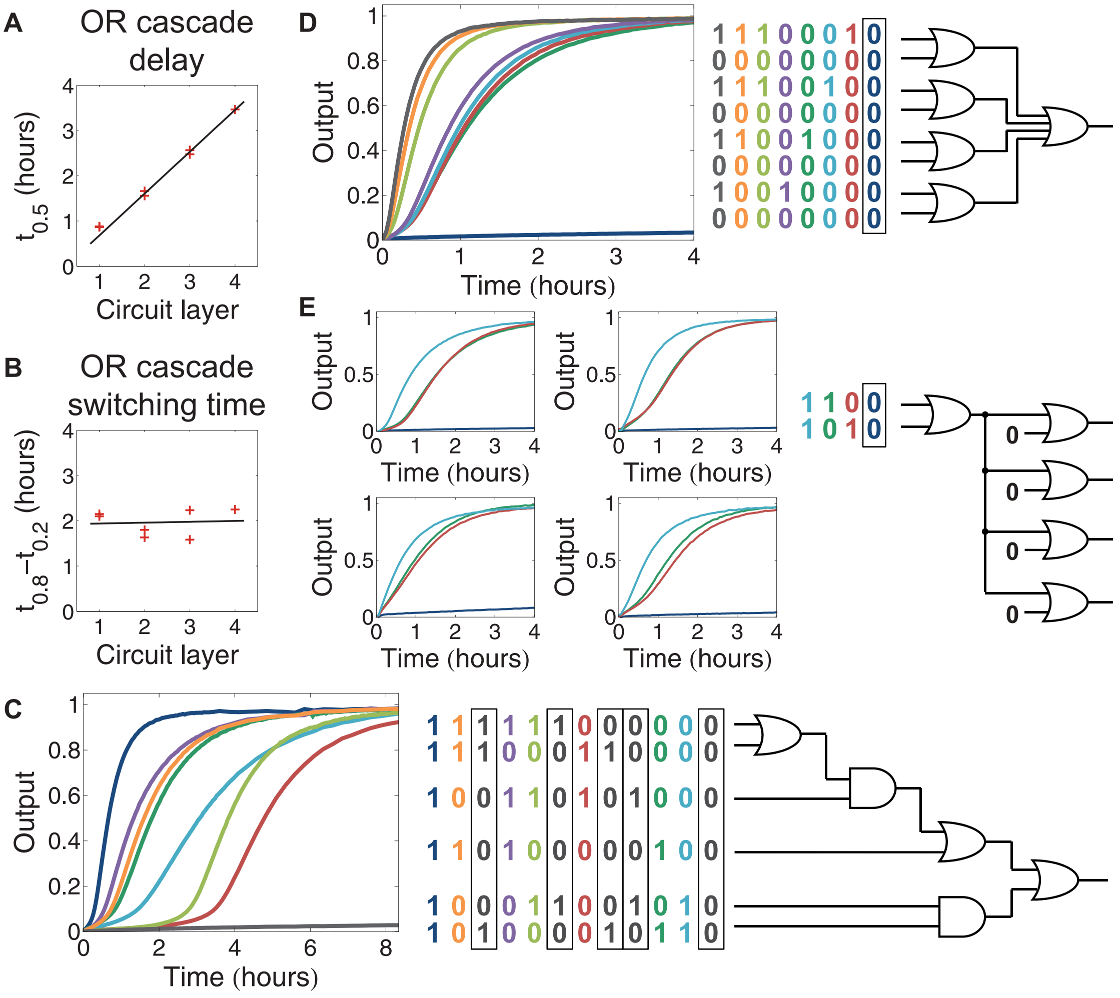


Fig. 3. Digital logic composition implemented with the seesaw DNA motif. (**A**) Circuit layer versus delay in OR cascades. Half completion times of seven selected experiments with a single input being ON in OR cascade circuits (Fig. 2 and figs. S5 and S6) are plotted against the depth of the activated input. (**B**) Circuit layer versus switching time in OR cascades. The time intervals between 20% and 80% completion of the above seven experiments are plotted against the depth of the activated input. (**C**) A circuit with four layers and five AND or OR gates. Numbers aligned with six input wires are logic values of respective inputs from 12 different experiments. Rectangles indicate the experiments where the output stayed OFF. Trajectories and their corresponding inputs have matching colors. (**D**) A circuit with a four-input OR gate. (**E**) A circuit with a four-output OR gate. Outputs from top to bottom in the circuit diagram correspond to plotted data of left top, right top, left bottom, and right bottom. Abstract diagrams of seesaw circuits in (C), (D), and (E) are included in figs. S7, S8, and S9, respectively. Sequences of strands are listed in tables S2 and S3, circuits 7, 8, and 9, respectively. Experiments were performed at 20°C, 1× = 100 nM, and 0.1× was used for OFF and 0.9× for ON inputs.

(fig. S5). In all tested cases, the output went to the correct ON or OFF state. A three-OR cascade (fig. S6, A and B) and a four-OR cascade (fig. S6, C and D) also worked. The delay time required for circuit computation increased linearly with the number of layers (Fig. 3A). However, once the threshold for the output gate was exceeded, the signal increased at roughly the same rate as in the smaller circuit (Fig. 3B). In a circuit with four layers, two AND gates, and three OR gates, with 12 different combinations of inputs, the output went to clear and correct ON or OFF states in 8 hours (Fig. 3C).

Because integrating gates support multiple inputs and amplifying gates support multiple outputs, logic gates built from a pair of them can easily support fan-in and fan-out. In a circuit with a four-input OR gate, only when all inputs from the upstream OR gates were OFF did the output stay OFF (Fig. 3D). In a circuit with a four-output OR gate, each output copied the correct logic from the upstream OR gate (Fig. 3E). Circuits with a four-input AND gate and a four-output AND gate are shown in fig. S8C and fig. S9C, respectively.

To demonstrate a digital circuit with an interesting function, we built a circuit that computes the floor of the square root of a four-bit binary number (Fig. 4A). It is not an optimized digital logic circuit; it is designed to showcase AND, OR, NOT, NAND, NOR, fan-in, and fan-out of logic gates, as well as fan-out of input signals. NOT gates are difficult to implement directly using representations where the ON or OFF state of an input is determined by the presence or absence of a single DNA species: A circuit might compute a false output before all input strands are added, because NOT gates already produce ON signals in the absence of their inputs, and for use-once circuits (such as seesaw circuits), computations cannot be undone. Therefore, we use dual-rail logic (fig. S10B). Each input is replaced by a pair of inputs, representing logic ON and OFF separately. Each logic gate is replaced by a pair of AND or OR gates. (Taking the NOR gate as an example, output being OFF is the OR of both inputs being ON; output being ON is the AND of both inputs being OFF.) Initially, the pair of inputs is absent, indicating that the logic value of this signal is unknown. At the beginning of computation, one input of the pair will be added, indicating either logic ON or OFF. In this way, no computation will take place before the input signals arrive. With dual-rail logic, any AND-OR-NOT circuit can be transformed into an equivalent circuit with AND or OR gates only. Then, any AND-OR circuit can be further transformed into an equivalent seesaw

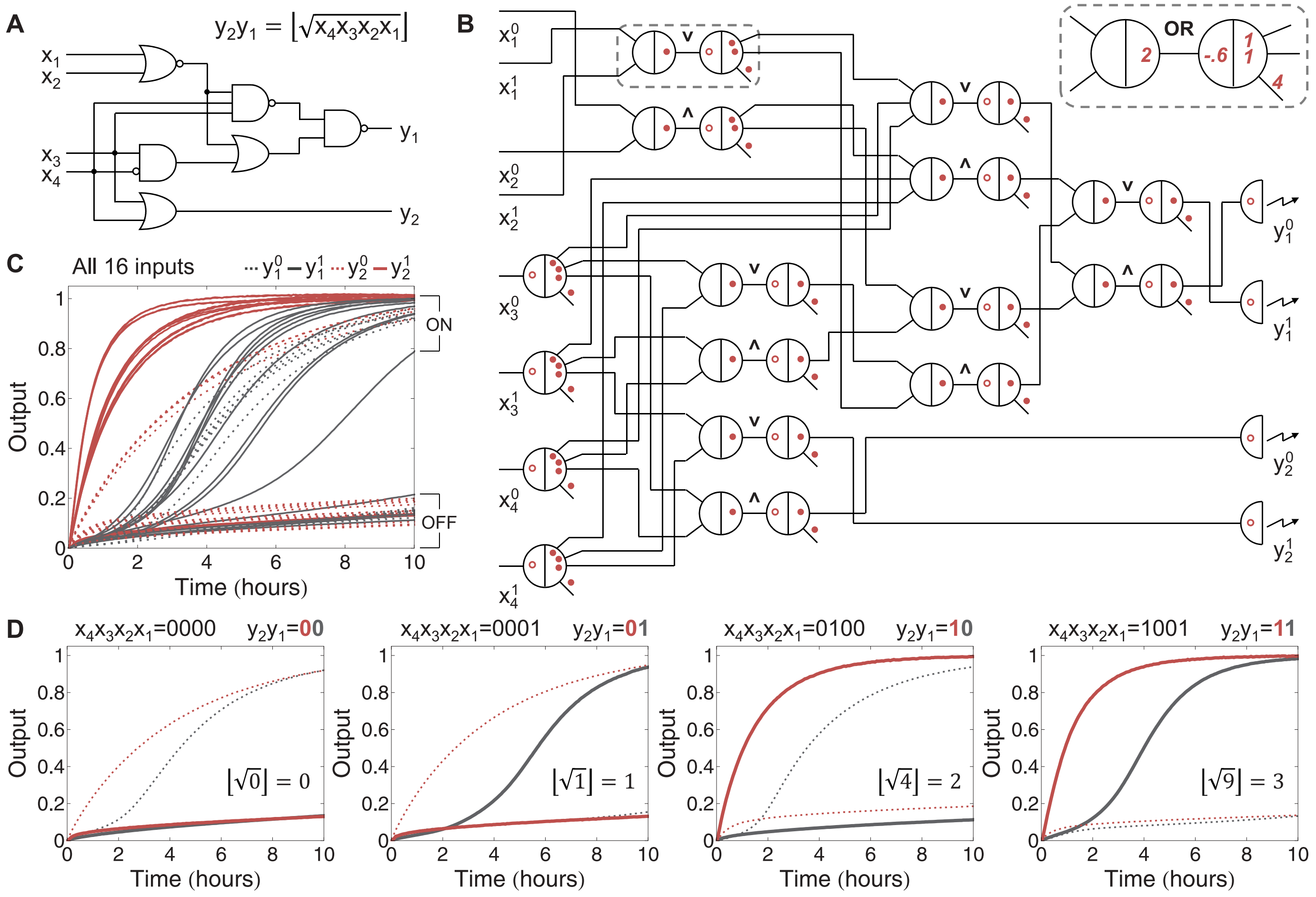


Fig. 4. A square-root circuit implemented with the seesaw DNA motif. (**A**) A digital logic circuit that computes the floor of the square root of four-bit binary numbers. (**B**) Abstract diagram of the seesaw circuit that is equivalent to the square-root digital logic circuit. x_i^0 and x_i^1 are dual-rail inputs of x_i, and they represent logic OFF and ON, respectively (the same rule applies to the outputs). Each pair of seesaw gates implements an AND (∧) or OR (∨) gate. Each pair of dual-rail AND or OR gates implements one ANDNOT, OR, NAND, or NOR gate. Red dots indicate positive red numbers, specifying initial relative concentrations of free or bound signals; red circles indicate negative red numbers, specifying initial relative concentrations of thresholds or reporters. An example of a two-input, two-output OR gate is highlighted; full details are provided in fig. S10. (**C**) Kinetics experiments of the square-root circuit with all combinations of inputs from 0000 to 1111. All 16 plots are shown separately in fig. S11. (**D**) Kinetics experiments that compute the square roots of 0, 1, 4, and 9. Trajectories and their corresponding outputs have matching colors. Dotted and solid lines indicate dual-rail outputs that represent logic OFF and ON, respectively. Sequences of strands are listed in tables S4 to S7. Experiments were performed at 25°C, 1× = 50 nM, and 0.1× was used for OFF and 0.9× for ON inputs.

circuit, with the construction we described above, and translated into DNA. The seesaw circuit that is equivalent to the square-root circuit in Fig. 4A is shown in Fig. 4B. This circuit has 74 initial DNA species, excluding inputs. When it runs, there are 130 different DNA strands—consisting of 15 to 33 nucleotides each—interacting within one test tube. With all possible inputs from 0000 to 1111, outputs went to the correct ON state or OFF state. The 16 plots are superimposed in Fig. 4C, and four selected examples are shown separately in Fig. 4D.

As circuits grow in size, a general debugging tool would be useful. In addition to the four output reporters for the square-root circuit, we had a fifth reporter to read one arbitrary internal output at a time (fig. S12). To do so, only one extra wire that connects the target gate to the fifth reporter needed to be added to the circuit. This corresponds to a single bound signal strand as an additional output from the target gate. Internal outputs from an OR gate, an AND gate, and an ANDNOT gate were observed without disturbing the functioning of the square-root circuit (fig. S13).

In our experiments, especially as the circuit size increased, we encountered issues related to sequence design and experimental conditions. For example, we added "clamps" to eliminate one type of leaky reaction (figs. S14 to S16), chose an optimal toehold length (fig. S17), selected the synthesis procedure for DNA strands (fig. S18), and adjusted the running temperature (fig. S19).

All components in seesaw circuits can be easily obtained from single-stranded DNA precursors (fig. S20), which facilitates parallel synthesis using DNA microarrays and parallel circuit preparation (*19*). During annealing, intramolecular hairpins form first and become kinetically trapped before other intermolecular reactions occur (*20*). Then, the undesired stem-loop fragments can be removed by restriction enzyme digestion or photocleavage. Kinetics experiments showed that thresholding and catalysis by a single seesaw gate worked well with DNA complexes prepared from hairpins (fig. S21).

The simple and systematic architecture for seesaw circuits made it possible to build quantitative models and to compile digital logic networks all the way to their DNA implementations. With just five rate constant parameters, our models fit single-gate data well (figs. S22 and S23) and semiquantitatively reproduced all other experimental data (figs. S24 to S31). The compiler [figs. S32 and S33; see also (*21*)] automatically translates any feedforward logic circuit into its equivalent seesaw circuit and DNA sequences, generates Mathematica and Systems Biology Markup Language (SBML) (*22*) code for simulations at the chemical reaction level, and generates DNA strand displacement calculus (DSD) (*10*) code for visualization and simulation at the domain level.

Three general principles guided us in successfully scaling up the complexity of DNA strand displacement circuitry. *Simplicity:* With just three basic reactions (seesawing, thresholding, and reporting) involving just four types of active species (free or bound signal, threshold, and reporter) comprising no more than two short strands, it is possible to develop a detailed understanding that generalizes to all units in a complex network. *Abstraction:* With five levels of hierarchical abstraction (DNA sequence, DNA domain, seesaw circuit, dual-rail logic, and AND-OR-NOT logic), design and analysis can take place at a higher level while neglecting irrelevant details. *Tolerance:* Signal restoration within every logic operation ensures that the digital abstraction is maintained even when synthesis and operational defects are inevitable.

Despite the speed, robustness, and straightforward sequence design of seesaw circuits, further scaling up will encounter challenges such as increased spurious binding that slows down the desired reaction rates and decreases the effectiveness of the thresholds. We expect that these challenges can be partially addressed by improved sequence design and by running reactions at lower concentrations (see supporting online material). However, a better solution would be to transition from solution-phase circuitry to circuitry organized on a surface, such as DNA origami (*23–25*), where adjacent DNA gates can interact without diffusion, spurious interactions are limited to immediate neighbors, and sequences can be safely reused in spatially separated locations.

A picture is now emerging for the future of DNA strand displacement circuitry. Proposals for systematic design of circuits with analog behavior (*9*) and systems that exploit spatial structures (*26*) make use of DNA complexes only moderately more complex than those used here. The problems of making reusable DNA gate components (*27*) powered by a single universal fuel (*28*) are being tackled. It is not yet clear what limits the amount of intelligence (*29*) that purely nucleic acid systems can exhibit.

As with other DNA strand displacement cascades, seesaw circuits can be applied to the embedded sensing and control of various molecular events by adapting biological signals such as microRNAs (*11*), small molecules (*30*), and proteins (*31*) as inputs or outputs. Moreover, the seesaw motif is structurally similar to microRNAs and small interfering RNAs (*32*, *33*)—all are short duplex nucleic acids with single-stranded overhangs processed from hairpins—hinting at the possibility that strand displacement circuitry may play an important, although still obscure, regulatory role within biological cells.

References and Notes

1. L. M. Adleman, *Science* **266**, 1021 (1994).
2. M. N. Stojanovic, D. Stefanovic, *Nat. Biotechnol.* **21**, 1069 (2003).
3. J. Elbaz *et al.*, *Nat. Nanotechnol.* **5**, 417 (2010).
4. Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, E. Shapiro, *Nature* **429**, 423 (2004).
5. B. Yurke, A. J. Turberfield, A. P. Mills Jr., F. C. Simmel, J. L. Neumann, *Nature* **406**, 605 (2000).
6. A. J. Turberfield *et al.*, *Phys. Rev. Lett.* **90**, 118102 (2003).
7. B. Yurke, A. P. Mills Jr., *Genet. Program. Evolvable Mach.* **4**, 111 (2003).
8. D. Y. Zhang, E. Winfree, *J. Am. Chem. Soc.* **131**, 17303 (2009).
9. D. Soloveichik, G. Seelig, E. Winfree, *Proc. Natl. Acad. Sci. U.S.A.* **107**, 5393 (2010).
10. A. Phillips, L. Cardelli, *J. R. Soc. Interface* **6** (suppl. 4), S419 (2009).
11. G. Seelig, D. Soloveichik, D. Y. Zhang, E. Winfree, *Science* **314**, 1585 (2006).
12. D. Y. Zhang, A. J. Turberfield, B. Yurke, E. Winfree, *Science* **318**, 1121 (2007).
13. P. Yin, H. M. T. Choi, C. R. Calvert, N. A. Pierce, *Nature* **451**, 318 (2008).
14. S. Venkataraman, R. M. Dirks, C. T. Ueda, N. A. Pierce, *Proc. Natl. Acad. Sci. U.S.A.* **107**, 16777 (2010).
15. H. M. T. Choi *et al.*, *Nat. Biotechnol.* **28**, 1208 (2010).
16. G. Eckhoff, V. Codrea, A. D. Ellington, X. Chen, *J. Systems Chem.* **1**, 13 (2010).
17. L. Qian, E. Winfree, *J. R. Soc. Interface* 10.1098/rsif.2010.0729 (2011).
18. D. E. Muller, in *Symposium on the Application of Switching Theory to Space Technology* (Stanford Univ. Press, Stanford, CA, 1963), pp. 289–297.
19. J. Tian, K. Ma, I. Saaem, *Mol. Biosyst.* **5**, 714 (2009).
20. J. S. Bois, thesis, California Institute of Technology (2007).
21. www.dna.caltech.edu/SeesawCompiler.
22. M. Hucka *et al.*, *Bioinformatics* **19**, 524 (2003).
23. P. W. K. Rothemund, *Nature* **440**, 297 (2006).
24. H. Gu, J. Chao, S. J. Xiao, N. C. Seeman, *Nature* **465**, 202 (2010).
25. K. Lund *et al.*, *Nature* **465**, 206 (2010).
26. L. Qian, D. Soloveichik, E. Winfree, in *DNA Computing and Molecular Programming, Lecture Notes in Computer Science, Vol. 6518*, Y. Sakakibara, Y. Mi, Eds. (Springer, New York, 2011), pp. 123–140.
27. E. Chiniforooshan, D. Doty, L. Kari, S. Seki, in *DNA Computing and Molecular Programming, Lecture Notes in Computer Science, Vol. 6518*, Y. Sakakibara, Y. Mi, Eds. (Springer, New York, 2011), pp. 25–36.
28. A. Goel, M. Ibrahimi, in *DNA Computing and Molecular Programming, Lecture Notes in Computer Science, Vol. 5877*, R. Deaton, A. Suyama, Eds. (Springer, New York, 2009), pp. 67–77.
29. E. B. Baum, *Science* **268**, 583 (1995).
30. R. M. Dirks, N. A. Pierce, *Proc. Natl. Acad. Sci. U.S.A.* **101**, 15275 (2004).
31. W. U. Dittmer, A. Reuter, F. C. Simmel, *Angew. Chem. Int. Ed.* **43**, 3550 (2004).
32. V. N. Kim, *Nat. Rev. Mol. Cell Biol.* **6**, 376 (2005).
33. R. W. Carthew, E. J. Sontheimer, *Cell* **136**, 642 (2009).

Acknowledgments: We thank D. Y. Zhang for providing useful comments on the manuscript. Supported by NSF grants 0728703 and 0832824 (Molecular Programming Project) and Human Frontier Science Program award RGY0074/2006-C.

Supporting Online Material

www.sciencemag.org/cgi/content/full/332/6034/1196/DC1
Materials and Methods
Figs. S1 to S34
Tables S1 to S9
References (*34–43*)

16 November 2010; accepted 15 April 2011
10.1126/science.1200520