Time-complexity of multilayered
DNA strand displacement circuits

(revised 2011-11-09)

Georg Seelig[1] and David Soloveichik[2]

[1] University of Washington, Seattle, WA, USA
[2] California Institute of Technology, Pasadena, CA, USA
`gseelig@u.washington.edu,dsolov@caltech.edu`

Abstract. Recently we have shown how molecular logic circuits with many components arranged in multiple layers can be built using DNA strand displacement reactions. The potential applications of this and similar technologies inspire the study of the computation time of multilayered molecular circuits. Using mass action kinetics to model DNA strand displacement-based circuits, we discuss how computation time scales with the number of layers. We show that depending on circuit architecture, the time-complexity does not necessarily scale linearly with the depth as is assumed in the usual study of circuit complexity. We compare circuits with catalytic and non-catalytic components, showing that catalysis fundamentally alters asymptotic time-complexity. Our results rely on simple asymptotic arguments that should be applicable to a wide class of chemical circuits. These results may help to improve circuit performance and may be useful for the construction of faster, larger and more reliable molecular circuitry.

Circuit depth is the standard measure of time-complexity of feed-forward circuits [8]. While this is well justified in electronic digital circuits, in this paper we ask whether depth is the correct measure of time-complexity for chemical circuits. We provide a quantitative analysis of how computation time is related to circuit size and architecture. We compare two elementary mechanisms for the underlying components: in one case, the underlying chemical reactions are stoichiometric and one input molecule produces one output molecule. In the other case the underlying reactions are catalytic and a single input molecule can trigger an arbitrary number of output molecules. We show that for non-catalytic circuits, the time to half-completion does not always scale linearly with the depth of the circuit. Our analysis shows that for a tree of stoichiometric bimolecular reactions, the time to half-completion scales quadratically with the depth of the circuit — i.e. the additional time due to adding an extra layer increases linearly with the size of the circuit. In contrast, we find that for catalytic systems the time to half-completion is a linear function of the depth independently of the structure of the circuit. The latter results agrees with our intuition from electronics where all gates are amplifying.

In this paper, for the physical model of molecular circuits we focus on DNA-based circuits implemented as cascades of strand displacement reactions. Single-stranded nucleic acids serve as signals that are exchanged between multi-stranded gate complexes. We have previously shown that this technology allows us to build multi-component molecular circuits that incorporate all the main features of digital logic, such as Boolean logic gates

like AND, NOT and OR, signal restoration and modularity [4]. More recent papers have implemented a variety of improvements including gates for fast catalytic amplification [11, 9], and reversible logic gates based on a simple catalytic gate motif [3]. This technology provides a starting point for building large-scale molecular circuitry with quantitatively predictable behavior using standardized off-the-shelf components [6].

In our experiments on nucleic-acid logic circuits we noticed that often the measured time to half-completion did not seem to scale linearly with the depth of the circuit. Instead, every additional layer seemed to add more than a constant offset to the time to half-completion. The observed slowdown may in part be due to non-specific interactions between DNA species that compete with or hinder the desired interactions between DNA gates and their inputs. However, here we will argue that the observed slowdown is at least in part a consequence of the circuit layout used in these experiments and of the underlying reaction kinetics of the DNA components.

In the next section we compare the time-complexity of linear cascades and converging trees for both non-catalytic and catalytic reactions. These circuit architecture represent extreme cases: A linear cascade of length N minimizes total fan-in while the converging tree of equal depth maximizes total fan-in. We describe specific implementations of these circuits with previously developed components and use numerical simulations to investigate how time complexity relates to circuit architecture and reaction mechanism. Then, we derive the asymptotic scaling of time-complexity of circuits using simplified kinetics. The simplified kinetics captures the essential features of the strand displacement circuits, but should be also applicable to alternative implementations of molecular circuits. Indeed the proofs are largely independent of the details of the underlying reaction mechanisms.

1 Comparing DNA strand-displacement base reaction mechanisms and circuit architectures

In this section we consider linear reaction cascades and converging trees based on DNA strand displacement chemistry previously described [4]. In the next section we consider more abstract and more generally applicable models of the same architectures that capture the essential behavior.

1.1 Cascades of non-catalytic reactions.

A strand displacement reaction (see Fig. 1(A)) can be intuitively described as a hybridization reaction between two complementary strands where one strand is initially fully single stranded (the "input") while the other strand is already partially double stranded (the "gate"). The reaction is driven forward by the formation of extra base pairs in the final compared to the initial state. Even though the underlying reaction mechanism is more complex, the overall reaction kinetics is well described as a bimolecular reaction $I + G \xrightarrow{k} O$ [10]. The rate constant k can be adjusted by changing the length and sequence composition of the single-stranded toe-hold (i.e. the single stranded overhang on the gate complex). If both reactants are present at the same initial concentration, i.e. $I(0) = G(0) = g_0$, the time evolution of G is given by the well-known expression
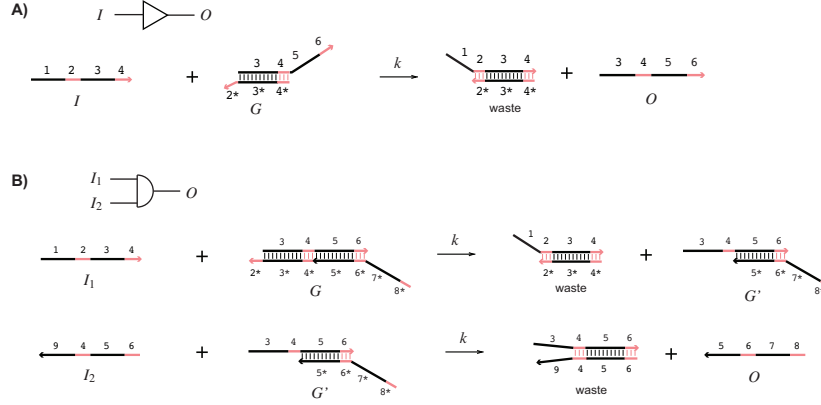
$$G(t) = \frac{g_0}{1 + kg_0 t} \tag{1}$$

Fig. 1: (A) Strand displacement transducer gate, producing signal O if signal I is present. (B) Strand displacement AND gate, producing signal O only if I_1 and I_2 are present. Toehold domains are shown in red. AND gates with different 3'-5' input and output orientations can be constructed similarly.

The time to half-completion $\tau_{1/2}$ for this reaction is $\tau_{1/2} = 1/kg_0$.

We now consider a linear cascade of strand displacement reactions of the type shown in Fig. 1. Fig. 2(A) shows the corresponding circuit diagram. For the n-th layer in the cascade we have the reaction $I_n + G_n \xrightarrow{k} I_{n+1}$. We have already solved the case $n = 1$ and for $n > 1$ conservation of mass requires that $I_n = G_n - G_{n-1}$. The resulting equation of motion is

$$\dot{G}_n = -k\left(G_n - G_{n-1}\right)G_n. \tag{2}$$

This equation has the form of a Bernoulli differential equation and can therefore be integrated formally. If we assume that all gates G_n and the initial input I_1 are present at the same initial concentration $I_1(0) = G_n(0) = g_0$ we can obtain a closed form solution for arbitrary n. In this case the result becomes

$$G_n(t) = g_0 \left[1 - \frac{\frac{1}{n!}\left(kg_0 t\right)^n}{\sum_{m=0}^{n}\frac{1}{m!}\left(kg_0 t\right)^m}\right]. \tag{3}$$

We can now ask at what time τ_f^N a particular fraction fg_0 $(0 < f < 1)$ of the maximum output level has been reached at the final layer N. The output $I_N(t)$ of the last reaction in the cascade is $I_N(t) = g_0 - G_N(t)$, and the time τ_f^N is the solution to $I_N(\tau_f^N) = fg_0$. Solving this equation for τ_f^N exactly would require us to find the zeroes of a polynomial of order N which is generally not possible. However, it can be shown[3] that $\tau_f^N = \frac{1}{1-f}\frac{N}{kg_0}$ is a good solution for large N in the sense that $\lim_{N\to\infty} I_N(\frac{1}{1-f}\frac{N}{kg_0}) = fg_0$. Then the

---

[3] The proof technically reduces to showing

$$\lim_{N\to\infty} \frac{(\alpha N)^N}{N!} \Big/ \sum_{m=0}^{N} \frac{(\alpha m)^m}{m!} = \frac{\alpha - 1}{\alpha} \tag{4}$$
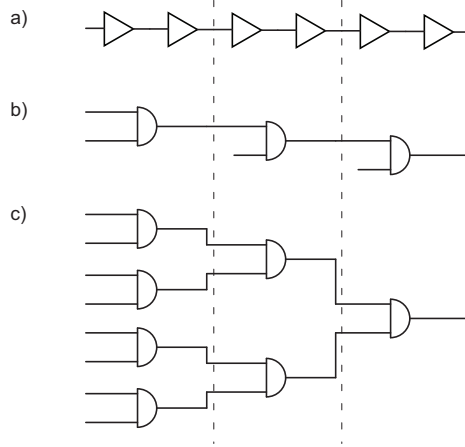
Fig. 2: Circuit architectures: (A) Cascade of tranducer gates. (B) Linear cascade of AND gates. (C) Maximally converging tree of AND gates. The alignment indicates that activation of an AND gate requires two sequential strand displacement reactions while a activation of a transducer gate requires one reaction.

cost in terms of additional run-time of adding an extra layer to a cascade of length N, for large N, can be well approximated as $\tau_f^{N+1} - \tau_f^N = \frac{1}{1-f}\frac{1}{kg_0}$ which means that adding an additional layer only leads to a constant (independent of N) increase in computation time. This agrees with the results from our numerical simulations (see Fig. 3).

1.2 Converging trees of non-catalytic reactions.

When two strand displacement reactions are cascaded on the same DNA complex, we obtain the second most simple gate motif: the AND gate of Ref. [4] shown in Fig. 3(B). The two inputs to the AND gate react sequentially and in an order specified by the gate design. The strand displacement reactions between the AND gate its two inputs are

$$I_1 + G \xrightarrow{k} G', \tag{5}$$

$$I_2 + G' \xrightarrow{k} O. \tag{6}$$

A linear chain of N AND gates where the output of a gate in layer n is the first inputs to the gate in layer $n + 1$ while the second input to each gate is provided initially is an alternative implementation of a strand-displacement reaction cascade of depth $2N$ (see Fig. 2 (B)). It can be analyzed using the same reaction equations, but with the roles of activated gate complex G' and single stranded input switched in the even layers.

---

for $\alpha \geq 1$. We can prove that the limit is at least $\frac{\alpha-1}{\alpha}$ by substituting $N^N/N! > m^m/m!$ in the denominator. We upper-bound the limit by discarding some initial terms in the denominator. Specifically, for any ϵ we can choose a large enough k that $\alpha^k/\sum_{m=0}^{k}\alpha^m$ is within $\epsilon/2$ of $\frac{\alpha-1}{\alpha}$, and for large enough N, the left hand side of eq. 4, keeping only the last $k+1$ terms, is within $\epsilon/2$ of $\alpha^k/\sum_{m=0}^{k}\alpha^m$.

As a contrasting architecture to a linear cascade, we consider a converging tree of AND gates that is N gates deep and $2N$ reactions deep as indicated in Fig. 2 (C). In this case, both inputs to an AND gate in any layer $n > 1$ must be the output of an AND gate in the previous layer. While the linear cascade has uniform fan-in 1, this circuit has uniform fan-in 2.

Simulation results for converging trees with up to 50 layers are shown in Fig. 3 (B). Interestingly, and unlike in the previous examples the time-complexity is not a linear function of the number of layers but in fact appears to increase quadratically with the depth of the circuit.

1.3 Cascades of catalytic reactions.

Cascades of catalytic reactions occur in many biological networks such as metabolic or signaling pathways. A variety of rationally designed DNA catalytic gates and even cascades of catalytic gates have been experimentally implemented [7, 5, 11, 2, 1]. In the following we use the minimal model $I_1 + G_1 \xrightarrow{k} I_1 + I_2$ to describe the catalytic systems. The reaction mechanisms proposed for hybridization-based catalytic systems in the literature [7, 5, 11] typically involve multiple strand displacement steps, but this simple model should capture the key differences between catalytic and non-catalytic strand displacement.

In contrast to the non-catalytic case, each step in a catalytic reaction cascade is intrinsically amplifying. A single input or catalyst can react sequentially with multiple gates or substrates and one would therefore expect that such cascades transmit signals faster. Similarly, since a sub-stoichiometric amount of input is enough to catalytically turn over a large number of substrates, we would expect the approach to equilibrium to be exponentially fast. A numerical simulation of a cascade of catalytic reactions confirms this intuition. The simulation provides strong evidence that the time-complexity grows linearly with the number of layers (see Fig. 3).

1.4 Converging trees of catalytic reactions

Finally, we want to consider a converging tree of catalytic reactions. We consider a tree using an ordered AND gate where each input acts catalytically, i.e.

$$I_1 + G \xrightarrow{k} I_1 + G', \tag{7}$$

$$I_2 + G' \xrightarrow{k} I_2 + O. \tag{8}$$

As in the case of a non-catalytic AND gate, G' is a partially double-stranded reaction intermediate. A converging tree of catalytic AND gates can be based on the same composition rules we used for the non-catalytic case (see Fig. 2(C)). However, unlike for the non-catalytic case, the time-complexity of a converging tree is linear in the depth of the tree as shown by numerical simulations (see Fig. 3). The systematic use of catalysis can therefore lead to a considerable speed-up of signal propagation.
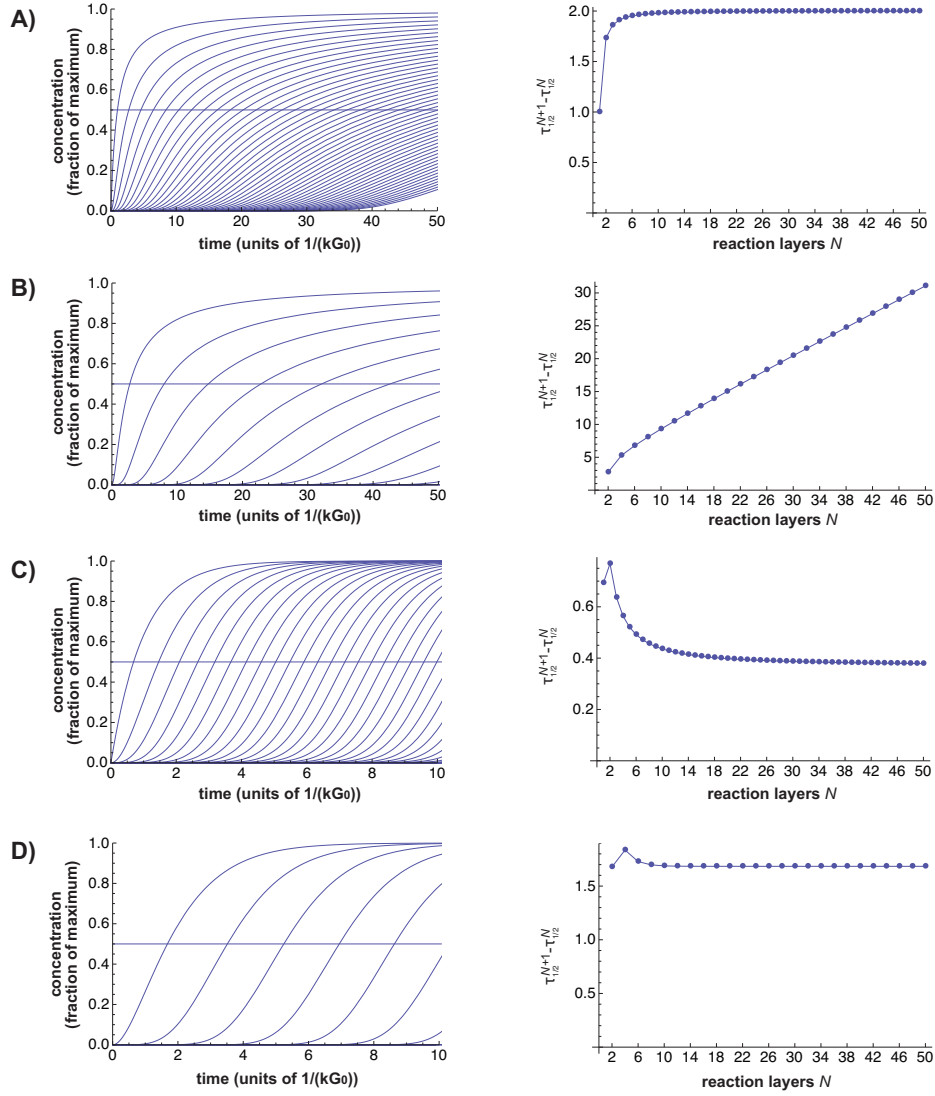
Fig. 3: Numerical simulations of different network topologies: (A) linear cascade, (B) converging exponential tree, (C) catalytic linear cascade, (D) catalytic converging exponential tree. For each case we simulated circuits of reaction depth $N = 1$–50. Note that in the converging exponential tree, a single circuit layer corresponds to two reactions. (Left column) Time-evolution of the output O_N for all circuits ($N = 1$–50). Note the vastly different timescales of (A)-(D). (Right column) $\tau_{1/2}^{N+1} - \tau_{1/2}^{N}$ as a function of N. Note that unlike in the case of a linear cascade (A) or catalytic systems (C)-(D), adding each new layer to the converging exponential tree without catalysis increases the half-time to completion by a larger and larger interval.

2 Asymptotic arguments

In this section we derive the asymptotic time complexity of linear cascades and converging trees, using simplified reaction kinetics. The simplified kinetics is sufficient to explain the scaling of the time complexity with the increasing number of layers and desired completion fraction of the strand displacement circuits considered above, but should be more generally applicable.

2.1 Non-catalytic reactions

For the non-catalytic case, we only consider a converging reaction tree. (As shown in Sec. 2.1 the differential equations describing a linear cascade can be solved exactly and the asymptotic time complexity is linear in the depth of the cascade.) As the simplest and most general non-catalytic AND gates we use the reaction schema $I_1 + I_2 \rightarrow O$. We prove that for maximally converging trees composed of these gates the time to f-fraction completion scales inversely as a function of $1 - f$ and quadratically as a function of N, establishing both lower and upper bounds that are within a factor of 4 of each other. For simplicity we assume that the initial concentration of inputs at the first layer is 1 and all rate constants are 1.

Lemma 1. *The time to f-fraction completion for N layers is at least $N^2/(4(1 - f))$ assuming $f \geq 1/2$.*

Proof. Let $f_n(t)$ be the amount of *each* signal in layer n at time t, and let $F_n(t)$ be the total amount of *each* signal that has been produced in layer $n+1$ by time t (i.e. has crossed from layer n to $n + 1$). First note that if we instantaneously force some amount of signals from layer n (i.e. decreasing $f_n(t)$) to layer $n + 1$ (i.e. increasing $f_{n+1}(t)$ and $F_n(t)$ by this same amount), we can only make $F_n(t)$ larger at any future time. This is because while $f_n(t)$ is lower, and thus $dF_n(t)/dt$ smaller, by conservation of signals across the layers, $f_n(t) + F_n(t)$ at any future time does not change whether we had pumped or not. Thus, since $f_n(t)$ is lower after pumping, $F_n(t)$ must be higher. Clearly, this also implies that this forward forcing of signals can only make $F_{n'}(t)$ for $n' > n$ larger at any future time.

Now, rather than starting with concentration 1 of each of the inputs, consider a sequence of such instantaneous forcings resulting in an an initial state where each $f_n(0) = (1 - g)/N$, with some g-fraction already produced at the output. By the above argument, we have only decreased the time to f-fraction completion.

Now note that in order for the amount of signal species in a given layer to increase, the amount of each signal species in the previous layer must exceed the amount of each signal species in the given layer; in other words, $df_n(t)/dt > 0 \Rightarrow f_{n-1}(t) > f_n(t)$. Thus starting in our start state, no signal species can exceed $(1 - g)/N$ concentration. Then, looking at the last layer, the rate of the production of the final output is never more than $(1 - g)^2/N^2$ and it takes at least $(f - g)N^2/(1 - g)^2$ time to produce the $f - g$ output to reach f-fraction completion. Maximizing this time with respect to g yields a maximum at $g = 2f - 1$. Assuming $f \geq 1/2$, we can use this worst-case amount of g to get completion time of at least $N^2/(4(1 - f))$. $\square$

Lemma 2. *The time to f-fraction completion for N layers is no more than $N^2/(1 - f)$.*

Proof. Using reasoning analogous to that in the previous lemma, if we artificially delay signal propagation, we can only delay completion. Consider the process in which we wait, sequentially, for each layer n to convert $1 - (1 - f)n/N$ amount of each input signal species to each output species. This process reaches f-fraction completion at the final layer no faster than the original process. Now the time for a layer starting with amount a of each input species to produce amount $b < a$ of each output species is $(b/a)/(a-b) < 1/(a-b)$. Thus, in our new process each layer takes less than $N/(1-f)$ time to convert the desired amount of input to output, for a total f-fraction completion time of the whole circuit of less than $N^2/(1-f)$. $\qquad\square$

2.2 Catalytic reactions

Next we consider our two examples of catalytic circuits: catalytic linear chains and catalytic converging trees using the simplest models conserving mass. The linear catalytic chain is composed of reactions $I + G \rightarrow I + O$ where I is the input, O is the output, and G is the source of mass for the output. The catalytic converging tree is composed of AND modules consisting of reaction $I_1 + I_2 + G \rightarrow I_1 + I_2 + O$, where I_1 and I_2 are the two inputs, O is the output, and G is the source of mass. We assume all G start at 1 so that the maximum output level is equal to the input level of 1. Note that the following results can be easily applied to different implementations since they do not rely on the details of the reaction mechanism.

We now show that the time for f-fraction completion of the catalytic linear chain and the catalytic converging tree scales linearly with the number of layers N and logarithmically in $1/(1-f)$.

Lemma 3. *The time to f-fraction completion is $O(N + \ln(1/(1-f)))$.*

Proof. In catalytic circuits, the amount of signal for any layer can never decrease. First of all, the time for a layer, assuming some fixed amount a of each input species, to produce amount b of each output species, is $\ln(1/(1-b))/a$ for the linear catalytic chain, and $\ln(1/(1-b))/a^2$ for the tree. Now consider the process in which we wait for the previous layer to produce $1/2$ of each output species, stop it, and start the next layer; this process is slower than the original process. In the new process, the total $1/2$-fraction completion time for the final layer is $O(N)$. Then in extra $O(\ln(1/(1-f)))$ time f-fraction completion can be reached. $\qquad\square$

Lemma 4. *The time to f-fraction completion is at least $\ln(1/(1-f)))$. Fixing $0 < f < 1$, the time to f-fraction completion is $\Omega(N)$ as $N \rightarrow \infty$.*

Proof. Starting with concentration 1 of inputs to the circuit, the time to produce f in the next layer is $\ln(1/(1-f)))$. The time to f-fraction completion for the output of the whole circuit is at least that, proving the first statement.

For the second part, consider the catalytic linear chain since it is faster than the catalytic tree. Now consider the process in which the G species are never used up and always stay at their initial value 1. Clearly, the f-fraction completion time for this process is faster than that for the original. Now for the modified process the concentration of the output of layer $n - 1$ at time t is the derivative of the concentration of the output of layer

n at time t. Thus the concentration of output at layer n at time t is $t^n/n!$. By Sterling's formula, $\lim_{n\to\infty}(n/e)^n/n! = 0$, and thus we would not be able to reach the desired f-fraction for some sufficiently large number of layers N in time N/e. $\qquad\square$

3   Discussion

Motivated by the recent advances in DNA nanotechnology which allowed for the construction of complex biochemical circuits, we investigated the time-complexity of multilayered molecular circuits. We considered two different circuit architectures, namely linear reaction cascades and converging trees of chemical reactions of increasing depth. In each case, we compared catalytic and non-catalytic mechanisms for the underlying reactions.

First, we investigated time complexity through numerical simulations of multilayered circuits. We used a model of circuit components based on cascaded strand displacement reactions. To obtain analytic confirmation of the observed behavior, we derived asymptotic scaling relationships for a simplified kinetics model. The simplified model captures the essential behavior of the strand displacement cascades, but is general enough to apply to a wide class of circuits. Simulation and analytic results show that the time to (f-fraction) completion scales linearly with the number of layers for all considered architectures except non-catalytic converging trees. In the latter case, the corresponding completion time scales quadratically with the number of layers.

The approach to completion is increasingly slow with increasing depth of the circuit but the asymptotic approach to equilibrium behaves differently for catalytic and non-catalytic circuits: The time to f-fraction completion scales as $1/(1-f)$ for non-catalytic systems considered, but as $\ln(1/(1-f))$ when catalysis is introduced.

From these observation we can conclude that the systematic use of catalysis allows one to build faster circuits with considerably improved asymptotic behaviors. However, care has to be taken since in a catalytic circuit any small leak at the input level will get exponentially amplified leading to a false positive output. A functional catalytic circuit will therefore require some degree of signal restoration which for example can be achieved through competitive inhibition (note that the additional components necessary for signal restoration can potentially slow down circuit operation). In contrast since there is no amplification in a cascade of bimolecular reactions the level of the output will not exceed the level of input.

References

1. J. Bois, S. Venkataraman, H. Choi, A. Spakowitz, Z. Wang, and N. Pierce. Topological constraints in nucleic acid hybridization kinetics. *Nucleic acids research*, 33(13):4090, 2005.
2. S. Green, D. Lubrich, and A. Turberfield. DNA hairpins: fuel for autonomous DNA devices. *Biophysical journal*, 91(8):2966–2975, 2006.
3. L. Qian and E. Winfree. A simple DNA gate motif for synthesizing large-scale circuits. In *Proceedings of the 14th International Conference on DNA Computing*, 2008.
4. G. Seelig, D. Soloveichik, D. Zhang, and E. Winfree. Enzyme-free nucleic acid logic circuits, 2006.
5. G. Seelig, B. Yurke, and E. Winfree. Catalyzed relaxation of a metastable DNA fuel. *Journal of the American Chemical Society*, 128(37):12211–12220, 2006.
6. D. Soloveichik, G. Seelig, and E. Winfree. DNA as a Universal Substrate for Chemical Kinetics (Extended Abstract). In *Proceedings of the 14th International Conference on DNA Computing*, 2008.
7. A. Turberfield, J. Mitchell, B. Yurke, A. Mills Jr, M. Blakey, and F. Simmel. DNA Fuel for Free-Running Nanomachines. *Physical Review Letters*, 90(11):118102, 2003.
8. H. Vollmer. *Introduction to Circuit Complexity: A Uniform Approach*. Springer, 1999.
9. P. Yin, H. Choi, C. Calvert, and N. Pierce. Programming biomolecular self-assembly pathways. *Nature*, 451(7176):318–322, 2008.
10. B. Yurke and A. Mills. Using DNA to Power Nanostructures. *Genetic Programming and Evolvable Machines*, 4(2):111–122, 2003.
11. D. Zhang, A. Turberfield, B. Yurke, and E. Winfree. Engineering entropy-driven reactions and networks catalyzed by DNA. *Science*, 318(5853):1121, 2007.