Localized Hybridization Circuits

Harish Chandran[1], Nikhil Gopalkrishnan[1], Andrew Phillips[2], and John Reif[1]

[1] Department of Computer Science, Duke University
{harish,nikhil,reif}@cs.duke.edu
[2] Microsoft Research
Andrew.Phillips@microsoft.com

Abstract. Molecular computing executed via local interactions of spatially contiguous sets of molecules has potential advantages of (i) speed due to increased local concentration of reacting species, (ii) generally sharper switching behavior and higher precision due to single molecule interactions, (iii) parallelism since each circuit operates independently of the others and (iv) modularity and scalability due to the ability to reuse DNA sequences in spatially separated regions. We propose detailed designs for local molecular computations that involve spatially contiguous molecules arranged on addressable substrates. The circuits act via enzyme-free DNA hybridization reaction cascades. Our designs include composable OR, AND and propagation Boolean gates, and techniques to achieve higher degree fan-in and fan-out. A biophysical model of localized hybridization reactions is used to estimate the effect of locality on reaction rates. We also use the Visual DSD simulation software in conjunction with localized reaction rates to simulate a localized circuit for computing the square root of a four bit number.

1 Introduction

Molecular computation (MC) is computation executed at the molecular scale. Since the seminal work of Adleman[1], many DNA based computation schemes have been proposed.

We will be discussing MCs that use strands of DNA to store state and execute various reactions that involve DNA hybridization and DNA strand displacement processes. DNA hybridization is the non-covalent binding of two complementary DNA sequences to form a single duplex structure. DNA strand displacement is the displacement of a single strand of DNA from a double helix by an incoming strand with a longer complementary region to the template strand. The incoming strand has a toehold, an empty single stranded region on the template strand complementary to a subsequence of the incoming strand, to which it binds initially. It eventually displaces the outgoing strand via a kinetic process modeled as a one-dimensional random walk. Strand displacement is a key process in DNA nanoscience and many DNA nanosystems relying on DNA strand displacement have been demonstrated, ranging from DNA walkers[2] to catalytic circuits[3][4] to molecular detectors[5].

L. Cardelli and W. Shih (Eds.): DNA 17, LNCS 6937, pp. 64–83, 2011.

1.1 Local versus Global MC

We distinguish two types of MC:

• A MC is local if its state is encoded by a spatially contiguous set of molecules i.e., the state of each computing element is explicitly determined by the configuration of these molecules, and transitions of state are executed via interactions between local computing elements. An example of a purely local MC is whiplash PCR of Sakamoto et al.[6] in which the computation state is present within a single molecule, and state transitions are executed via polymerization reactions on that molecule. This paper gives a detailed description of a novel local MC using only DNA hybridization and no enzymatic reactions.

• In contrast, a MC is global if the state of the device is spatially distributed i.e., the state is determined by averaging the concentration and configuration of spatially distributed molecules that define state, and transitions of state are executed via multiple distributed interactions. An example of global MC is DNA reaction networks such as seesaw gates of Qian and Winfree[7] where all elementary logical operations (e.g., logical AND and OR) are performed via a host of diffusion based strand displacement interactions.

Note that other MC may combine local and global; for example the tile assembly of Rothemund and Winfree[8] performs computation by incorporating molecules into a growing assembly.

1.2 Motivation for Local MC

We discuss some potential advantages of local MC over global MC by also considering conventional computing devices. Decades of ingenious improvements (e,g., digital behavior, locality of memory, and local parallelism) to the design of conventional computing devices have allowed their performance to be vastly improved, providing optimizations beyond the improvements due purely to fabrication technology. We aim in this paper to exploit key aspects of these designs in our designs of local MCs.

Execution Speedup via Locality of Memory: In conventional computing devices such as digital processors data is stored locally wherever possible and only when the local memory limitations are exhausted is memory repositioned in more distributed locations. Hence conventional computing devices exploit locality as a critical method to improve execution time for both memory access and processing. It is reasonable that locality also be critical to the design of MC devices.

In local MC, because the dominant interactions of a molecule involve a fixed set of neighbors, we can preferentially speed up designed interactions over spurious ones. In contrast, in a global MC, such as a DNA hybridization circuit, the speed of execution is severely limited by diffusion processes, which are stochastic processes whose rates are governed by the mobility and concentration of the molecules. In particular, the rate of hybridization of two complementary freely floating DNA strands (of moderate length) in a dilute buffer solution is limited

by the frequency of collisions between these DNA strands, since these molecules must first find each other by the much slower diffusion reaction prior to the subsequent hybridization reaction that occurs after they contact each other[9], [10]. The rate of diffusion can be adjusted, but is ultimately limited by pragmatic considerations. Note that in a global MC, such as a DNA hybridization circuit, the mobility of molecules can be increased by increasing the temperature, but this is limited to be at most the melting temperature of the hybridization. Alternately, one can attempt in a global MC to increase the rate of collisions by increasing the concentration of the molecules, but this also increases the rate of spurious reactions (e.g., unintended hybridization and/or dehybridizations, etc), by the same factor. Hence, ultimately the computation rate of such global MCs is limited by diffusion rates, not the local reaction rates. This implies that global MCs, such as DNA hybridization networks which rely on diffusion at each step, may be inherently slower than local MCs where the local concentration of the reacting species results in collision rates that are several orders of magnitude larger.

Digital Behavior in Conventional Computing Devices: In conventional digital computing machines the digital behavior of switches is a fundamental building block and it is reasonable that they would also be critical to the design of MC devices. Bi-stable devices that are used to simulate switches exhibit analog behavior as they execute switching transitions. A sharper transition leads to faster switching times. Greater activation energies for the switching transition lead to increased robustness of the device. These two factors together decide the digital behavior of the device. This sharp digital behavior seems likely to be useful in incorporating variants of techniques (such as fault tolerance) perfected for standard silicon based digital devices into the design of MC devices. There are considerable differences in the digital behavior of local and global MC:

- Circuits implemented by local MC using spatially contiguous molecules exhibit sharp switching behavior within the computing unit. In particular, the state of each computing element is explicitly determined by the configuration of these molecules, so is unambiguous and digital in nature.
- In contrast, circuits implemented by global MC such as DNA hybridization reaction networks exhibit analog behavior that can be translated into digital behavior only by applying thresholds. In particular, the state of a global MC is determined by averaging the concentration and configuration of spatially distributed molecules that define state, and so is an analog value.

Local Parallelism in Conventional Computing Devices: Modern-day conventional computing devices exploit parallel processing at many levels, both at the local processor chip as well as in multi-core architectures, to allow computations to progress along different computational paths in parallel. MC can exploit local parallelism to allow multiple distinct local MCs to progress along different computational paths in parallel. For example, a plethora of programmable nanomanufacturing systems have been demonstrated[11][12]. These nanomanufacturing systems require the state information to be stored locally so that different products can be assembled in parallel in different molecules. Even if only a single

product is targeted, different instances of the same product might be at different stages of manufacture at any given time. This requires the state information to be stored locally and is ideally suited for local molecular computing. In contrast, it is not evident how to implement this local control in a global MC.

Modularity: In conventional computing, devices generally contain multiple identically designed computational units that act as modules in the overall architecture and considerably simplify the overall design. In MC, we can expect that modularity will also be a critical aspect of the design and in large part relates to DNA sequence reusability:

- Local MC involves interactions only with a fixed set of neighbors of each molecule and this enables distinct namespaces across molecules and opens up possibilities of sequence reuse in the system. In particular, since our local DNA hybridization circuits are spatially separated and cannot directly interact, we can reuse the same DNA sequences to build the same functionality (e.g., DNA hybridization circuit for a given logical operation) on a different part of the system using very few distinct DNA sequences in the overall design.
- Global MC on the other hand involves interactions among DNA strands that can be present anywhere in the reacting vessel, which implies a single global namespace for the sequences, and hence considerably limits DNA sequence reusability.

1.3 Our Contribution and Paper Organization

We develop detailed designs to implement DNA circuits on fully addressable DNA nanostructures such as a fully addressable lattice developed by Yan et al.[13] or DNA origami developed by Rothemund[14]. In doing so we develop a local molecular computing methodology to compute arbitrary Boolean functions. Our circuits are designed carefully to place downstream gates close enough to upstream gates to implement rapid signal transduction but far enough to minimize leaks. We argue that our circuits will: (i) be faster than chemical reaction networks due to increased local concentration of reacting species, (ii) exhibit generally sharper switching behavior and higher precision due to single molecule interactions, (iii) be highly parallel since each circuit operates independently of the others which finds use in nanomanufacture (iv) be modular and scalable due to the ability to reuse DNA sequences in spatially separated regions. We estimate the effect of locality on reaction rates via a biophysical model of localized hybridization reactions. We then use the Visual DSD simulation software in conjunction with these localized reaction rates to simulate a localized circuit for computing the square root of a four bit number.

1.4 Prior Work

In theory it is possible to perform arbitrary computations using DNA strands, ignoring issues of errors, scale and time. Current experimental demonstrations are far from this ideal. Synthetically designed DNA nanosystems currently cannot be leveraged to perform computationally complex biochemical tasks. The

need is for DNA circuits that are autonomous, error-resilient, scalable, fast and energy efficient. We review progress toward this goal over the past decade.

Catalytic DNA Circuits: Zhang et al.[3] argued that autonomous, scalable circuits require signal amplification and proposed catalysis as a means of achieving amplification. Their entropy-driven catalytic gate achieved an auto-catalytic exponential amplification of a DNA signal, as opposed to linear amplification[5]. The free energy for this reaction is a result of entropic gain due to dissociation of weakly-bound DNA strands. At the same time, Yin et al.[4] also illustrated the power of catalysis by demonstrating catalyzed formation of branched structures, auto-catalytic exponential amplifiers, dendritic growth and bipedal walkers. All these reactions were executed using an elementary hairpin motif with distinct functional sequence domains, allowing the reaction network to be abstracted by a simple visual symbolic language. These papers establish catalysis as a robust paradigm for constructing amplifying DNA gates.

Composable DNA Gates: The next challenge is to integrate catalytic DNA gates into large circuits. Qian and Winfree[15] proposed the simple and elegant *seesaw* gate, a modification of the catalytic gate of Zhang et al.[3] to make it composable. All toehold domains in the seesaw gate are identical, allowing for modular design of the circuit and parallel synthesis of many gates. Using the seesaw gates, Qian and Winfree[7] were able to demonstrate a non-trivial computation using hundreds of gates. The key obstacle toward further scaling up their circuit is the speed of operation. Since the seesaw gates are freely floating in solution, the reaction rates are primarily limited by the time it takes for DNA strands to encounter each other via diffusion. The authors take note of this issue and suggest moving their circuits to origami to speed-up hybridization kinetics, avoid cross talk and re-use sequences in spatially separate locations. This work expands upon those ideas.

Two Domain Fork and Join Gates: Cardelli[16] developed designs for transduction, fork and join gates using DNA strands limited to two domains. He analyzed the power of systems composed of these 3 basic gates, proved that they are equivalent to Petri nets and simulated various systems specified in this two domain language. Our independently derived designs for AND, OR and PROPAGATION gates are similar to the fork and join gates described in this work.

Addressable DNA Substrates: We develop designs that implement DNA hybridization circuits on fully addressable substrates. Much experimental progress has been achieved in the area of addressable DNA substrates. Park et al.[17] demonstrated a fully addressable lattice built out of the cross tile developed by Yan et al.[13]. This was later extended to an 8×8 fully addressable lattice by Pistol and Dwyer[18] using hierarchical assembly techniques. Rothemund[14] developed the technique of DNA origami in which a long scaffold DNA strand obtained from a viral genome is folded into the desired shape by the use of hundreds of short synthetic DNA strands called staples. Each staple strand binds to the scaffold strand at precisely two locations thereby localizing two distinct points on the scaffold strand. Hundred such stapling events fold the scaffold

into the desired shape. Origami has been widely used as a substrate to arrange various molecules[12][19] and has been extended to form three dimensional shapes[20][21].

2 Design of DNA Hybridization Circuits on Addressable DNA Substrates

We begin by designing two composable DNA hybridization driven gates that perform AND and OR Boolean logic. An additional propagation gate serves as a wire and propagates signal. The gates are implemented as a cascade of toehold mediated strand displacement reactions. Each of these reactions is initiated via a *universal toehold binding domain* (labeled $\widetilde{T}$) whose sequence is the same throughout the circuit. The specificity of strand displacement is conferred by a set of *specificity domains* (labeled $\widetilde{S}_i$) that are unique to each gate. We assume that there exists an irreversible downstream drain (not shown in the figures) for each gate.

The gates can be precisely positioned on the fully addressable DNA substrate by designing them as extensions of conventional substrate strands. The actual positioning of the gates depends on the digital circuit being implemented. In particular, gates that are connected to each other are placed close to each other. In this sense, the localized DNA circuit mimics the topology of the actual digital circuit diagram. Each localized circuit structure contains one copy of each gate in the circuit and operates by signaling across gates via single molecules, eliminating the need for signal restoration.

Any boolean function can be implemented using purely AND and OR gates by the use of dual rail logic, which uses two bits each to encode 0 and 1. The propagation gate is used to control the relative positions of the AND and OR gates on the substrate so that the circuit elements can function correctly without steric interference.

2.1 Design of Logical Gates

Figure 1 illustrates how two hairpin motifs are used to achieve OR logic. For input domains S_{i_1} and S_{i_2} we design motifs with the sequences $S_{i_1}TS_o\widetilde{T}\widetilde{S}_{i_1}\widetilde{T}$ and $S_{i_2}TS_o\widetilde{T}\widetilde{S}_{i_2}\widetilde{T}$ respectively where $\widetilde{S}_{i_1}$ and $\widetilde{S}_{i_2}$ are the input recognition domains and S_o is the output domain. In the presence of either the sequence TS_{i_1} or TS_{i_2}, one of the hairpins opens up to reveal the output S_o as illustrated in Figure 1. Note that the reaction that opens the hairpin is irreversible in the presence of a downstream gate that consumes S_o.

Figure 2 illustrates a two input AND gate complex consisting of a hairpin motif and a protector strand. For input domains S_{i_1} and S_{i_2}, the hairpin motif has sequence $S_{i_2}TS_o\widetilde{T}\widetilde{S}_{i_2}\widetilde{T}\widetilde{S}_{i_1}\widetilde{T}$ and is hybridized to the protector that has sequence $S_{i_1}T$. The sequences $\widetilde{S}_{i_1}$ and $\widetilde{S}_{i_2}$ are the input recognition domains and S_o is the output domain. In the presence of the sequence TS_{i_1} the protector is displaced out of the complex exposing a universal toehold domain $\widetilde{T}$. If the

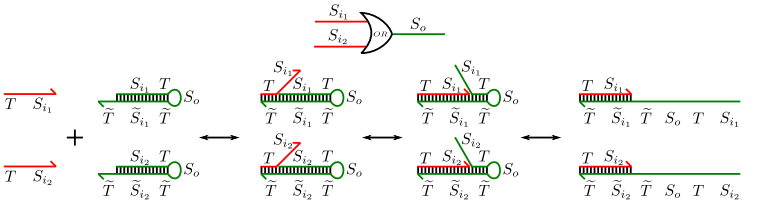


Fig. 1. The OR gate computing the Boolean function $S_{i_1} + S_{i_2}$. It is implemented by two green strands $S_{i_1}TS_o\widetilde{T}\widetilde{S}_{i_1}\widetilde{T}$ and $S_{i_2}TS_o\widetilde{T}\widetilde{S}_{i_2}\widetilde{T}$. The inputs are TS_{i_1} and TS_{i_2}. The presence of either of these input strands triggers the exposure of the output TS_o which is initially sequestered in a hairpin.

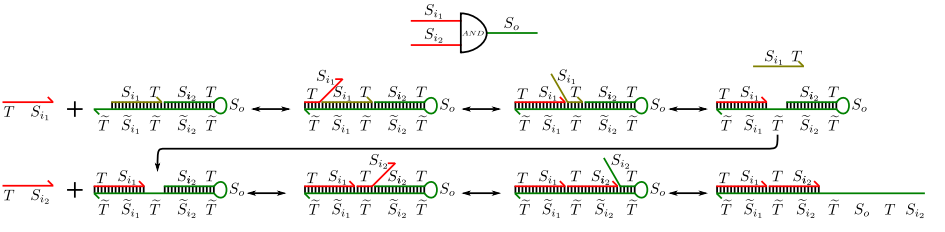


Fig. 2. The AND gate computing the Boolean function $S_{i_1} \cdot S_{i_2}$. It is implemented by a complex consisting of the green strand $S_{i_2}TS_o\widetilde{T}\widetilde{S}_{i_2}\widetilde{T}$ hybridized to the light green strand $S_{i_1}T$. The inputs are TS_{i_1} and TS_{i_2}. The presence of both of these input strands triggers the exposure of the output TS_o which is initially sequestered in a hairpin.

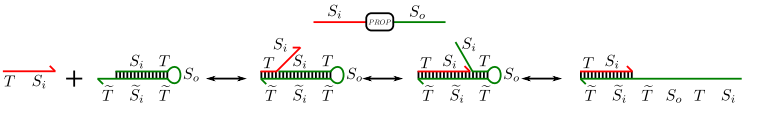


Fig. 3. The Propagation gate enables the signal transduction $S_i \rightarrow S_o$. It is implemented by a green strand $S_iTS_o\widetilde{T}\widetilde{S}_i\widetilde{T}$ The input TS_i triggers the exposure of the output TS_o which is initially sequestered in a hairpin.

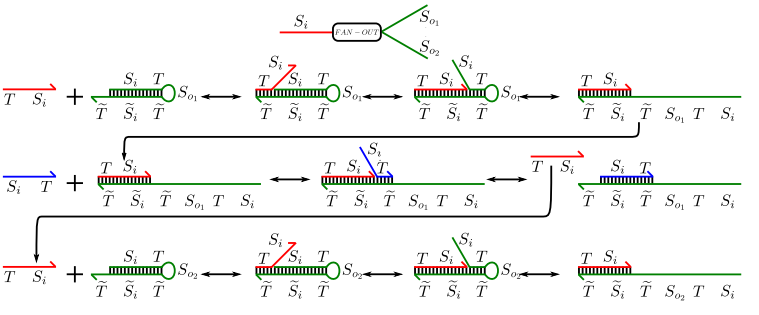


Fig. 4. A degree two fan-out gate transducing input signal S_i to two output signals S_{o_1} and S_{o_2}

sequence TS_{i_2} is also present, it initiates strand displacement via this newly exposed toehold to reveal the output S_o as illustrated in Figure 2. Note that the reaction that opens the hairpin is irreversible in the presence of a downstream gate that consumes S_o.

Figure 3 illustrates a propagation gate, that act as a wire that propagates signal. It can be thought of as an OR gate with one of the inputs hard wired to a Boolean 0. In our implementation this is achieved by simply leaving out one of the two motifs that make up the OR gate. By stringing together a series of propagation gates, we can create signal transduction pathways between gates.

Circuits with gates that support a fan-in of 2 and a fan-out of 1 are capable of computing any boolean function. However, supporting higher fan-in and fan-out reduces the size of the circuit (number of gates) and simplifies the circuit. While unlimited fan-in and fan-out is not practical for real physical systems, we show how one may achieve a fixed small degree of fan-in and fan-out for our DNA gates. A k degree fan-in OR gate can be achieved by using k hairpin motifs in parallel. The degree of fan-in is restricted by the space available on the substrate. Overcrowding or spreading the gate over a larger area may degrade performance. The optimal arrangement would have to be experimentally determined. A k degree fan-in AND gate can be achieved by using a hairpin motif with $k-1$ protectors in serial. The switching speed of the multi-input AND gate is inversely propositional to the degree of fan-in due to the serial nature of the AND gate.

A k degree fan-out from a signal S_i can be implemented using k downstream propagation gates transducing the signal S_i to signals $S_{o_1}, S_{o_2}, \ldots S_{o_k}$ using $k-1$ copies of a fuel strand S_iT. Fuel strands can be tethered to the substrate to achieve localized hybridization kinetics. A degree 2 fan-out gate is illustrated in figure 4. The signal S_i activates one of the k downstream propagation gates whose output region S_{o_i} is consumed by an irreversible downstream drain. The fuel strand now binds to the propagation gate using the newly exposed toehold $\widetilde{T}$ and kicks off the signal S_i which can now activate another propagation gate and so on until all the k distinct propagation gates are serially activated. Once again, transduction speed of the fanout logic is inversely propositional to the degree of fan-out due to its serial nature. This rate might be improved by tethering the k copies of the fuel strands near the propagation gates.

2.2 Compiling Boolean Circuits into DNA Hybridization Circuits

Converting a Boolean circuit into a DNA circuit involves two stages. First the Boolean circuit is compiled into a dual-rail circuit that computes the same function. There may be several different dual-rail circuits that compute the same function and we may wish to find an optimal one, based on characteristics like small number of gates, uniform depth, balanced signal propagation delay across all pathways (recalling that our DNA AND gates are slower than OR gates) etc. We can compile the dual-rail Boolean circuit into a DNA circuit by using the AND, OR and propagation gate motifs. We optimize the placement of these gates on our substrate, adding or deleting propagation gates as necessary.

The sequence design for the gate motifs is modular and we simply design all binding domains so as to minimize spurious interactions. Techniques for domain level sequence design have been discussed by Zhang[22].

In practice, we may incorporate further optimizations in designing our DNA circuits rather than compiling them directly. For instance, we may wish to target certain leaky portions of the circuit and make them more fault tolerant by, for instance, replicating the circuit module and taking the majority. Such techniques to increase robustness of circuits are well studied in the VLSI community and it is likely that the algorithmic solutions found there may be translated simply into DNA circuits because of the digital gate level abstraction that we provide.

2.3 Assembly of DNA Hybridization Circuits on Addressable DNA Nanostructures

In this section we illustrate two methods to organize DNA circuits on addressable DNA substrates by highly parallel synthesis that is experimentally feasible and scales to large number of gates.

Pistol and Dwyer[18] have demonstrated the assembly of fully addressable DNA lattices of size up to 8×8 using hierarchical assembly techniques. The hierarchical assembly can be parallelized and has the potential to be scaled beyond 8×8 lattices. We use this approach to organize our circuits on DNA lattices. Each DNA gate motif is designed as an extension of one of the strands that is part of the tile that assembles into a lattice. Since each tile in the finally formed lattice is uniquely addressable, the motifs can be precisely and specifically positioned on the lattice according to the circuit being implemented.

Rothemund[14] demonstrated the breakthrough DNA origami technique for manufacturing large, fully addressable DNA nanostructures with high yield in a single pot reaction. Each designed hybridization interaction in a DNA origami structure is between a staple and a region of the scaffold. No staple-staple or scaffold-scaffold interactions are designed. Thus, even if the relative concentrations of the staples are imprecise, the final yield of the origami structure remains high as long as an excess (about $10\times$) of staple strands is used. We use the same approach to organize our circuits on DNA origami. Each DNA gate motif is an extension of a staple strand. Since the surface of the origami is uniquely addressable, the motifs can be precisely and specifically positioned on the origami according to the circuit being simulated.

The key cause for concern is that when annealed, the hairpin strands will interact with each other rather than folding up into the required hairpin motif. However, there is evidence that when annealed, dilute ($\approx$ nM concentrations) interacting strands undergo uni-molecular reactions and fold up into hairpin motifs rather than hybridizing with each other via bi-molecular reactions[23]. This is explained by noting that the hairpin structure is stable at a higher temperature than the intermolecular complex and as the system is cooled, the motifs form hairpins first getting kinetically trapped in the non-optimal thermodynamic state. It is unclear if this assumption holds when the strands are locally concentrated, for instance, by tethering them close to each other. To avoid this problem,

we design our motifs such that their hairpin structure is stable at higher temperatures than both the temperature at which they are stably incorporated into the substrate and the temperature at which the bimolecular complex is stable. When annealed, we expect the hairpin motif to form while the strands are dilute and not yet tethered to the substrate, and then the motifs are incorporated into the substrate.

Since the OR motifs are simply single stranded hairpins, this is easily achieved by making the length of the specificity domain moderately longer than the length by which the motif is tethered to the origami. In practice, the tether length could be 16 bases while the length of the specificity domain could be 20 and the toehold domain could be 5 bases, making the stem of the OR hairpin motif 25 bases long. The AND motif is slightly more problematic since it is a two strand complex - a protector strand hybridized to a hairpin motif. By choosing lengths of 20 and 5 bases for the specificity and toehold domains we can ensure that the protector-hairpin complex is stable at a higher temperature than the temperature at which the origami tether is stable. However, an upstream input to the AND motif would have similar stability with the AND hairpin as the protector-hairpin complex as both are bimolecular reactions. This difficulty can be overcome in one of two ways. We can anneal the protector-hairpin complex separately, purify it and then add it to the origami mix. While annealing the origami mix we take care to not heat the sample above the melting temperature of the protector-hairpin complex. Alternately, we design the AND motif as a single hairpin motif and cleave the motif at the appropriate site after annealing by using a nicking enzyme. For this purpose we can design the toehold sequence as the recognition domain of a nicking enzyme which cleaves one of the strands of a double helix upstream of its recognition site. Note that a single nicking enzyme would be sufficient to prepare all protector-hairpin complexes and this process could be implemented in parallel. Also, the restriction enzyme won't nick the OR hairpin motifs as the corresponding position in these structures is single stranded.

2.4 Reusing Sequences in Spatially Separated Circuits

In the circuits we have discussed thus far, each distinct circuit element (wires and gates) is implemented by a distinct DNA sequence that is never reused. This immediately places an upper bound on the complexity of the circuits since length of DNA sequences constituting each type of circuit element is fixed. Moreover, assigning distinct sequences to different copies of the same circuit element might result in variance in their operational characteristics.

If the gates in a circuit are spatially separated into clusters such gates can interact directly only with other members of the cluster they belong to, then sequences can be reused across different clusters. Information is exchanged between clusters via signal transduction pathways. In the extreme case, one can imagine a cluster to be a single gate and that each such gate is connected to other gates via long signal transduction pathways. These pathways composed of multiple propagation gates can also benefit from domain reuse. For example the k long pathway $W_1 \rightarrow W_2 \rightarrow W_2 \rightarrow \ldots W_k$ can be replaced by an equivalent k

long pathway $W_a \rightarrow W_b \rightarrow W_a \rightarrow \ldots W_b$. Care should be taken when sequences are reused in signal transduction pathways running close to each other or crossing over one another. In such cases, signal flow across one pathway might initiate a spurious signal flow across the other if pathways share same sequences. Such problematic areas in the circuit can use unique sequences to minimize crosstalk.

Reusing DNA sequences to build the same functionality on a different part of the DNA hybridization circuit allows us to build complex circuits with very few distinct DNA sequences. Here we use the key property that our local DNA hybridization circuits (for example a DNA hybridization circuit for a given logical operation) are spatially separated and so cannot directly interact. While the sequence domains that take part in circuit interactions are reused, the tether sequences that position the elements on an addressable substrate can be distinct (eg. in DNA origami) or can be reused (eg. in hierarchical assembly).

2.5 Functional Units and Architectures

We have discussed how to build digital circuits using DNA sequences on an addressable substrate. We can build complex circuits via a hierarchical method. Small substrates can implement functional units that can be connected in a precise manner to synthesize computing architectures. The hierarchical assembly process developed by Park et al.[24] can be directly applied to build such circuits using tile based assemblies. If origami is used as a substrate, then different origami can be connected to each other via sticky ends to form larger assemblies. One could also think of using a secondary scaffold to organize different origami in a precise manner to enable information flow between them. One simple layout for such architecture would be to have the computing elements in the middle of the origami and connect them up to neighboring origami via long signal transduction pathways that terminate at the edge of the origami. An advantage of using such architectures is the ability to "plug and play" various functional units. For example, if we have designed and experimentally tested a set of functional units, say an adder, subtracter and square rooter, then we can build circuits that are composed of these functions by plugging these units into precise positions on the assembly. These functional units could be designed to ensure that they can communicate via the same signal transduction pathways for each input/output bit so that they can be composed seamlessly.

3 Modeling and Simulations of Tethered Systems

In this section we investigate the speedups obtained in localized hybridization circuits as follows: (i) We develop a simple biophysical model of tethered hybridization and estimate values for a toehold binding speedup factor λ which depends on parameters of the design of the tethers. Our biophysical model of tethered hybridization closely follows the work of Genot et al.[25] and we use data reported by Qian and Winfree[7] as a starting point for computing reaction rates. (ii) Then we use the Visual DSD system[26] to model and simulate a four bit square root circuit for various values of λ. We discuss some of the shortcomings of this simple model in section 4.1.

3.1 A Biophysical Model of Tethered Hybridization

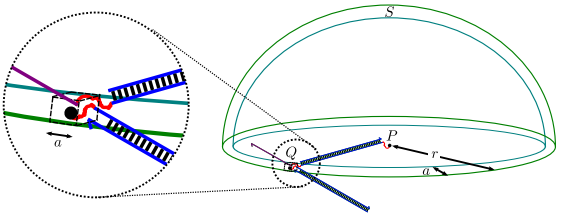


Fig. 5. A biophysical model of substrate tethered hybridization

Toehold exchange by strand displacement is modeled by Zhang and Winfree[27] as a three stage process: toehold binding, branch migration and toehold unbinding. We study the effect of tethering on a toehold exchange reaction by considering the effect of tethering on each of these stages. Branch migration and toehold unbinding rates are unlikely to be affected by tethering since they are local processes internal to the molecule. The rate and order of a toehold binding reaction changes due to tethering and essentially becomes a unimolecular reaction, taking place within a single macromolecule, with a larger rate constant. We make the biophysics motivated assumption that the speedup in the toehold binding rate constant is purely due to the effective concentration of the incoming strand. Our analysis and assumptions closely follow the work of Genot et al.[25]. We first calculate an approximate effective concentration c for the incoming toehold strand under certain biophysical assumptions. The new toehold binding rate constant $\tilde{k}_s$ is calculated as the product of the original (diffusion-based) toehold binding rate k_s and the effective concentration c of the incoming strand.

We assume that the gates are tethered to their substrate via a short single strand of DNA that has negligible persistence length and hence acts as a completely flexible hinge. The double stranded portions of the gates are much longer than these tethers and are assumed to behave like stiff rods. Figure 5 illustrates the interaction between an upstream gate tethered at P and a downstream gate tethered at Q, a distance r away. The cube of side a illustrates the reaction volume, the volume within which the incoming toehold region must lie for toehold binding reaction to occur. We assume that a is much smaller than r. We also ignore the single stranded region at the end of the double stranded region of the upstream gate and assume that the effective concentration for toehold binding is approximately the concentration of the end of the double stranded region in the reaction volume. The probability that the end of the double stranded region lies within the reaction volume is given by: $p = V_a/V_s$ where $V_a = a^3$ is the reaction volume and $V_s = 2\pi r^2 a$ is the volume of the shell S of thickness a that the end of the double stranded region can explore. Thus, $p = a^2/2\pi r^2$. The effective concentration in particles/m^3 is $c = p/a^3 = 1/2\pi r^2 a$. We convert this into molarity (moles per liter) by dividing by $1000N_a$ where $N_a = 6.023 \times 10^{23}$ is Avogadro's number. Thus $c = 1/2000\pi r^2 a N_a$ M. The rate constant for short toehold binding in diffusion based systems is $k_s = 5 \times 10^4$ /M/s at 25°C (see [7] supplementary information). This gives a tethered toehold rate constant of $\tilde{k}_s = k_s \times c$ /s.

r	a	λ
20 nm	5 nm	$\lambda_1 = 1.32 \times 10^3 \approx 10^3$
10 nm	2 nm	$\lambda_2 = 1.32 \times 10^4 \approx 10^4$
5 nm	1 nm	$\lambda_3 = 1.06 \times 10^5 \approx 10^5$

The overall rate of any toehold binding reaction is the product of the scaled rate constant $\tilde{k}_s$ and the operating concentration c_0 of the assembled circuits in solution according to mass action kinetics. We approximate the kinetics of toehold binding as a bimolecular reaction and for the purposes of simulation generate a pseduo-second order rate constant scaled by the operating concentration c_0, $\hat{k}_s = \tilde{k}_s/c_0 = k_s \times c/c_0 = k_s \times \lambda$ where λ is the pseudo-bimolecular rate constant speedup. This effectively translates to a tethered toehold binding reaction rate of $\tilde{k}_s \times c_0$. For typical operating concentration of $c_0 = 100$ nM, $\lambda = 1/2000\pi r^2 N_a c_o = 2.64 \times 10^6/ar^2$ when a and r are expressed in nanometers. The above table illustrates three values for λ based on different physically feasible values for a and for the length of the double stranded region r.

3.2 Simulations of Tethered Hybridization Circuits using Visual DSD

Visual DSD [26][28] is a tool for specification, compilation and simulation of a wide class of toehold mediated DNA strand displacement reactions. Various hybridization circuits based on toehold exchange can be specified as a program in the DSD programming language along with rate constants for binding and unbinding of toeholds and for branch migrations. These programs are compiled into a set of reactions under one of four different semantics that allow modeling of the reaction at different granularities. This set of reactions can then be simulated by either a deterministic ODE solver or a stochastic Gillespie molecular simulation.

We used DSD to program a four bit square root circuit constructed out of AND, OR, FAN-OUT and PROPAGATION gates described in the previous section. As the current implementation of DSD does not allow for specification of hairpins, we modeled our systems without hairpins but enforced necessary locality by private name-spaces for different toeholds. This ensures that the output strand that physically detaches from the gate (modeling the opening of hairpins) attaches only to a very specific set of downstream gates via a private toehold. This models locality of reactions at the logical level. To model the kinetics of localized hybridization, we used rate constants reported by Qian and Winfree[7] but multiplied the toehold binding rate constant by a factor λ derived in the previous section. As noted earlier, localized hybridization circuits can be thought of as series of uni-molecular reactions within a macromolecule that changes state every time a reaction occurs. Our modifications to the toehold binding rate constant and its modeling as pseudo second order rate constant are consistent with this understanding. It is important to note that our simulation methodology is based on an approximation using mass action kinetics. Refer to section 4.1 for a discussion on alternate modeling approaches.

In particular, we set our toehold binding rate constant to $\lambda \times 5 \times 10^{-5}$/nM/s, toehold unbinding rate constant to 26/s and branch migration rate constant to 1/s. To validate the chosen rate constants, the solution based, non-localized, four bit square root circuit experimentally demonstrated by Qian and Winfree[7] was simulated with these rate constants. We also set $\lambda = 1$ and used reported rate constants for threshold toehold binding (2×10^{-3}/nM/s) and threshold toehold unbinding (1.3/s). DSD simulations with these rate constants agreed well with experimental data reported by Qian and Winfree[7][1].

As an initial test we implemented the circuit illustrated in figure 6 in DSD using our simulation methodology[1]. The reaction graph for this system is shown in figure 7. In addition to using private toeholds to model locality as described previously, we also used a common toehold with unmodified binding rate constant for the input strands, to model the diffusion of inputs to the substrate.

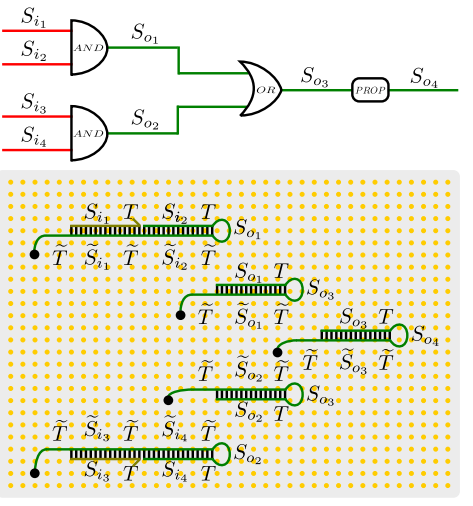


Fig. 6. Example computation of $(S_{i_1} \cdot S_{i_2} + S_{i_3} \cdot S_{i_4}) \rightarrow S_{o_4}$ implemented by two AND gates, an OR gate and a Propagation gate indicated by green strands and complexes. The addressable substrate is illustrated by the gray rectangle and the points of addressability are illustrated by the yellow dots. The gates are placed at specific locations, indicated by big black dots, such that down stream gates are next to the their corresponding inputs from the upstream gates.

Figure 8a shows the stochastic simulation of the circuit compiled with finite semantics with unproductive reactions switched on, for different values of λ. Finite semantics mode in DSD models the toehold exchange reactions as a three step process of toehold binding, branch migration and toehold unbinding and assigns specific rates to each process. Unproductive reactions occur when branch migration does not follow after successful toehold binding. This happens when the specificity domain of the incoming strand is different from the specificity domain of the gate which is caused when multiple specificity domains share the same toehold. For more details about the Visual DSD tool and the options it offers see [28].

The simulations show marked improvement in overall reaction rates with increasing λ but diminishing gains as λ increases beyond 1000. This can be attributed to the fact that the overall reaction kinetics is governed by the rate limiting step of input diffusion at higher λ.

[1] Data and code available at
`http://research.microsoft.com/dna/dna17localized.zip`

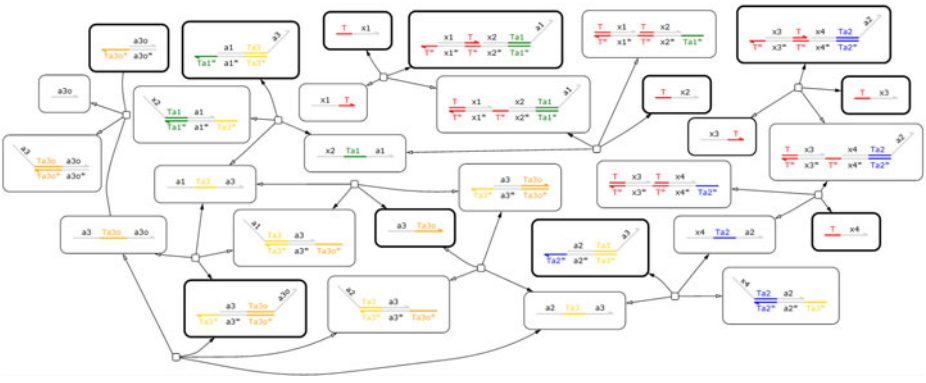
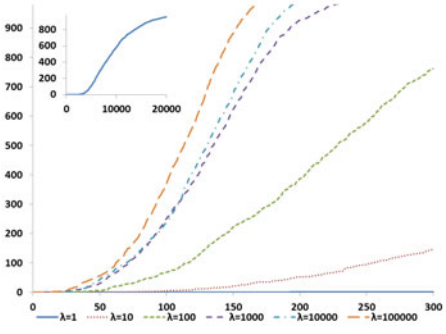
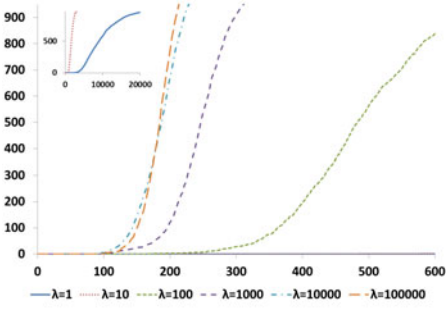


Fig. 7. Reaction graph for the example circuit illustrated in figure 6 implementing the Boolean function $x_1 \cdot x_2 + x_3 \cdot x_4$. Initial species are highlighted with darker borders. Note that the input strands $Tx1, Tx2, Tx3, Tx4$ have a common universal toehold T whose binding rate constant is set to 5×10^{-5}/nM/s while the other toeholds are private (for example $Ta2$ is private to domain $a2$) and their binding rate constants are set to $\lambda \times 5 \times 10^{-5}$/nM/s.



(a) Simulation data for circuit implementing $x_1 \cdot x_2 + x_3 \cdot x_4$ with inputs set to $x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 0$ for different λ. Simulation time: 300 seconds. Simulation carried out in finite mode with unproductive options on. Inset shows longer run of system for $\lambda = 1$.



(b) Simulation data for output LSB from the square root circuit with input 1100 for different λ. Simulation time: 600 seconds. Simulation carried out in finite mode with unproductive options on. Inset shows longer run of system for $\lambda = 1, 10$.

Fig. 8. Simulation data

Encouraged by these results we programmed a four bit square root circuit using AND, OR and FAN-OUT gates described earlier. As before, we used private toeholds with modified binding rate constants to model locality and used a universal toehold with standard rate constants to model diffusion of inputs. Figure 8b shows data from simulations using DSD in finite mode with unproductive reactions switched on. We first simulated the four bit square root circuit for all

input values with $\lambda = 1$ and found that the computation of the output LSB for input 1100 was the slowest to 95% completion. Due to space consideration and in the spirit of worst case analysis, we show the behavior of this signal at different values of λ. More data from our simulations is available at this URL[1]. As before, completion rates show dramatic improvements with increasing λ and the overall kinetics is rate limited at higher λ by input diffusion (which roughly takes 150 seconds at 100 nM).

4 Discussion

4.1 Refined Modeling and Simulations of Tethered Systems

All localized toehold binding interactions are assigned the same rate constant in our simulations. However, there are two types of localized toehold binding, one between regions on the same strand (resulting in hairpin formation) and another between distinct strands. Our biophysical model only applies to the latter case. A more careful analysis would include different binding rates in both cases. The challenge of modeling and simulation of tethered systems is to model uni-molecular reactions rather than the bimolecular reactions found in most other conventional hybridizations reaction systems. Our biophysical model is preliminary and ignores the worm-like chain behavior of single strands of DNA. A more detailed model may give a better estimation of local concentrations.

Molecular circuits tend to leak: downstream hybridization cascades are sometimes set off by thermal noise even in the absence of upstream signals. Spurious hybridization interactions, either due to unintended sequence complementarity or unproductive interactions between complementary domains (e.g. universal toehold binding), is another common problem. Sequence reuse can mitigate unintended sequence complementarity while careful positioning of interfering strands on the substrate can inhibit unproductive reactions. Nevertheless we expect leaks and unproductive reactions to exist, and hence the need for a model of leak reactions in localized circuits. A fruitful approach may be to model the time to failure of each gate as a random variable and use this to estimate the overall leak rate. We discuss some error-tolerance mechanisms in section 4.4.

We simulate tethered systems using mass action kinetics. An alternate approach would be to model the entire network as a continuous time Markov chain and estimate (or simulate) its expected time to completion where the times to completion of toehold binding, branch migration and toehold unbinding could be estimated via DNA molecular dynamics. Our preliminary simulations (data not shown) under this regime indicates higher speedups in tethered systems. Addressing these issues is beyond the scope of this paper and will be a part of future work in tethered hybridization systems.

4.2 Optimizations

The circuits discussed thus far have only utilized one side of the addressable surface. Suppose the substrate is addressable on both sides and the substrate is

stiff and dense enough to ensure that strands on one side cannot interact with the strands on the other (note that these assumptions are true for certain DNA origami). Then it is possible to use both sides of the substrate for implementing different circuits. These circuits can still interact at the edges of the substrate for signal transduction.

The output of simple circuits could be a Boolean value requiring just one bit. The output of such computation can be detected via standard flurophore/quencher protocols. But complex circuits might output multiple bits or compute an integer requiring the detection of multiple bits. Though one can use multiple flurophore/quencher pairs operating at various frequencies, it is quite clear that this solution does not scale. One possible way to overcome this issue is to implement the circuit for each bit of the output in different test tubes. Once again, we can exploit the spatial separation of circuits to reuse the same flurophore/quencher pair for every output bit.

To assemble large circuits on origami requires longer scaffolds. Naturally occurring long scaffolds might be problematic since they might exhibit strong secondary structure or might interfere with the DNA sequences used for various gates. If we use spatial separation to restrict ourselves to a small set of DNA sequences, we might mitigate the latter problem. The former problem might be solved by careful design of a synthetic scaffold.

4.3 Synchronous Computation and Nanomanufacture

The circuits described in this paper were asynchronous and used dual rail logic. It is possible to achieve synchronous lock-step computation using AND gates. For example, if we want one part of a circuit (say part B) to be activated only after another part of the circuit (say part A) has finished its computation, then each signal transduction pathway entering into B can be changed into the output of the AND of that original pathway and a specific completion signal from part A. Thus part B is locked unless part A is complete. Alternating this strategy across two circuits allows them to proceed in a lock-step fashion.

This technique can be extended to nanomanufacture applications. We can think of the entire process having two components, a fabricating nanomachine like a DNA walker and a computing logic that governs the action of the fabricating device. The fabricating device and the computing logic can be operated in the lock step fashion described earlier. The computation leading to this product formation can be governed by the actual inputs to the circuit.

4.4 Possible Errors and Techniques to Mitigate Them

Errors with localized DNA circuits can broadly be classified into two types: errors in organizing the gate motifs on the substrate and errors in operation. The chief possible errors in organizing the motifs on the substrate are: (i) missing motifs and (ii) damaged motifs due to incorrect folding, sequence truncation or formation of spurious bimolecular complexes. Techniques to deal with these kinds of errors are discussed in section 2.3.

Errors in operation are chiefly due to: (i) leaks via spontaneous opening of the hairpin motifs (ii) leaks via stacking induced strand displacement and (iii) spurious toehold binding. Spontaneous opening of the hairpin motifs are likely to be rare at our operation conditions, since the stem of the hairpin is 25 bases. We refer to the end of the stem at the loop region as the head of the motif and the other end as its tail. Stacking induced strand displacement is likely to occur via head to tail stacking of motifs. The loop region is likely to sterically hinder such stacking, destabilizing it. We will also experiment with carefully orienting the motifs on the origami surface such that these stackings strain the motif tether region and are hence sterically hindered. For instance the motifs likely to undergo stacking could be oriented alongside each other. Since each motif has the same toehold binding region, the output of one motif may bind to the toehold region of a neighboring motif even if they are not designed to interact. This spurious interaction is prevented from setting off downstream reactions by the mismatch in the specificity domains. However, such reactions may block the toehold region and slow down the operation of the circuit. This problem is present even with the seesaw circuit of Qian and Winfree[7] but does not seem to significantly affect their correct operation for moderate circuit sizes. The spurious toehold interactions in our designs are restricted to the diameter of motifs reachable by the tethered motif, in contrast to the seesaw circuits where it is a global problem.

In spite of these techniques, a certain level of leaks is unavoidable. If we assume that every copy of the circuit on origami has a fixed independent failure probability of ϵ, then we expect that out of N targeted copies of the circuit, $N(1-\epsilon)$ of them will function correctly. The final output of the circuit is the consensus of the outputs across all copies of the circuit, with appropriately set thresholds based on the failure rate. Alternately, we can implement standard techniques in fault tolerance into our circuits. This correction of errors at the logical level has been used with great success in building of semiconductor based circuits and this provides inspiration in dealing with errors due to leaks.

5 Conclusions

Local bimolecular reactions have multiple advantages over global molecular computation. In this paper we have developed detailed designs to implement DNA circuits on fully addressable DNA nanostructures such as a fully addressable lattice developed by Yan et al.[13] or DNA origami developed by Rothemund[14]. In doing so we developed a local molecular computing methodology to compute arbitrary Boolean functions. Our circuits are designed carefully to place downstream gates close enough to upstream gates to implement rapid signal transduction but far enough to minimize leaks. We argued that our circuits will: (i) be faster than chemical reaction networks due to increased local concentration of reacting species, (ii) exhibit generally sharper switching behavior and higher precision due to single molecule interactions, (iii) be highly parallel since each circuit operates independently of the others which finds use in nanomanufacture

(iv) be modular and scalable due to ability to reuse DNA sequences in spatially separated regions. A biophysical model of localized hybridization reactions was used to estimate the effect of locality on reaction rates. We used the Visual DSD simulation software in conjunction with these localized reaction rates to simulate a localized circuit for computing the square root of a four bit number.

This effort is a first attempt at realizing enzyme-free localized hybridization circuits. In light of the rapid growth of DNA nanotechnology, it is our hope that the principles expounded in this paper will serve as a starting point for the eventual realization of localized hybridization circuits in the laboratory.

Acknowledgments. We wish thank to Erik Winfree, David Zhang and Bernard Yurke for useful discussions on the localized strand displacement process. We thank anonymous referees for pointing out relevant prior work and critical suggestions on modeling of tethered systems. We would also like to thank Sudhanshu Garg for assisting in creating figures for a preliminary draft and Archana Ramamoorthy for proof-reading. This work was supported by NSF EMT Grants CCF-0829797 and CCF-0829798.

References

1. Adleman, L.: Molecular Computation of Solutions to Combinatorial Problems. Science 266(5178), 1021–1024 (1994)
2. Sherman, W., Seeman, N.: A Precisely Controlled DNA Biped Walking Device. Nano Letters 4, 1203–1207 (2004)
3. Zhang, D., Turberfield, A., Yurke, B., Winfree, E.: Engineering Entropy-Driven Reactions and Networks Catalyzed by DNA. Science 318, 1121–1125 (2007)
4. Yin, P., Choi, H., Calvert, C., Pierce, N.: Programming Biomolecular Self-assembly Pathways. Nature 451(7176), 318–322 (2008)
5. Dirks, R., Pierce, N.: Triggered Amplification by Hybridization Chain Reaction. Proceedings of the National Academy of Sciences of the United States of America 101(43), 15275–15278 (2004)
6. Sakamoto, K., Kiga, D., Momiya, K., Gouzu, H., Yokoyama, S., Ikeda, S., Sugiyama, H., Hagiya, M.: State Transitions by Molecules. Biosystems, 81–91 (1999)
7. Qian, L., Winfree, E.: Scaling up Digital Circuit Computation with DNA Strand Displacement Cascades. Science 332(6034), 1196–1201 (2011)
8. Rothemund, P., Winfree, E.: The Program-Size Complexity of Self-Assembled Squares. In: Symposium on Theory of Computing, pp. 459–468 (2000)
9. Turberfield, A., Mitchell, J., Yurke, B., Mills, A., Blakey, M., Simmel, F.: DNA Fuel for Free-Running Nanomachines. Physical Review Letters 90(11) (2003)
10. Seelig, G., Yurke, B., Winfree, E.: Catalyzed Relaxation of a Metastable DNA Fuel. Journal of the American Chemical Society 128(37), 12211–12220 (2006)
11. He, Y., Liu, D.: Autonomous Multistep Organic Synthesis in a Single Isothermal Solution Mediated by a DNA Walker. Nature Nanotechnology 5(11), 778–782 (2010)
12. Gu, H., Chao, J., Xiao, S.-J., Seeman, N.: A Proximity-based Programmable DNA Nanoscale Assembly Line. Nature 465(7295), 202–205 (2010)

13. Yan, H., Park, S.H., Finkelstein, G., Reif, J., LaBean, T.: DNA-Templated Self-Assembly of Protein Arrays and Highly Conductive Nanowires. Science 301(5641), 1882–1884 (2003)
14. Rothemund, P.: Folding DNA to Create Nanoscale Shapes and Patterns. Nature 440, 297–302 (2006)
15. Qian, L., Winfree, E.: A Simple DNA Gate Motif for Synthesizing Large-scale Circuits. DNA Computing, 70–89 (2009)
16. Cardelli, L.: Two-Domain DNA Strand Displacement. DCM, 47–61 (2010)
17. Park, S.-H., Yin, P., Liu, Y., Reif, J., LaBean, T., Yan, H.: Programmable DNA Self-assemblies for Nanoscale Organization of Ligands and Proteins. Nano Letters 5, 729–733 (2005)
18. Pistol, C., Dwyer, C.: Scalable, Low-cost, Hierarchical Assembly of Programmable DNA Nanostructures. Nanotechnology 18, 125305–125309 (2007)
19. Lin, C., Liu, Y., Yan, H.: Self-Assembled Combinatorial Encoding Nanoarrays for Multiplexed Biosensing. Nano Letters 7(2), 507–512 (2007)
20. Douglas, S., Dietz, H., Liedl, T., Hogberg, B., Graf, F., Shih, W.: Self-assembly of DNA into Nanoscale Three-dimensional Shapes. Nature 459(7245), 414–418 (2009)
21. Dietz, H., Douglas, S., Shih, W.: Folding DNA into Twisted and Curved Nanoscale Shapes. Science 325(5941), 725–730 (2009)
22. Zhang, D.: Towards Domain-Based Sequence Design for DNA Strand Displacement Reactions. DNA 16, 162–175 (2010)
23. Dirks, R., Bois, J., Schaeffer, J., Winfree, E., Pierce, N.: Thermodynamic Analysis of Interacting Nucleic Acid Strands. SIAM Review 49, 65–88 (2007)
24. Park, S.H., Pistol, C., Ahn, S.J., Reif, J., Lebeck, A., LaBean, C.D.T.: Finite-Size, Fully Addressable DNA Tile Lattices Formed by Hierarchical Assembly Procedures. Angewandte Chemie International Edition 45(5), 735–739 (2006)
25. Genot, A., Zhang, D., Bath, J., Turberfield, A.: Remote Toehold: A Mechanism for Flexible Control of DNA Hybridization Kinetics. Journal of American Chemical Society 133(7), 2177–2182 (2011)
26. Phillips, A., Cardelli, L.: A Programming Language for Composable DNA Circuits. Journal of The Royal Society Interface 6(11), 419–436 (2009)
27. Zhang, D.Y., Winfree, E.: Control of DNA Strand Displacement Kinetics Using Toehold Exchange. Journal of the American Chemical Society 131(48), 17303–17314 (2009)
28. Lakin, M., Youssef, S., Cardelli, L., Phillips, A.: Abstractions for DNA Circuit Design. Journal of The Royal Society Interface (in press, 2011)