Less Haste, Less Waste: On Recycling and its Limits in Strand Displacement Systems

Anne Condon, Alan Hu, Ján Maňuch, and Chris Thachuk

The Department of Computer Science,
University of British Columbia, Vancouver, British Columbia, V6T 1Z4

Abstract. We study the potential for molecule recycling in chemical reaction systems and their DNA strand displacement realizations. Recycling happens when a product of one reaction is a reactant in a later reaction. Recycling has the benefits of reducing consumption, or waste, of molecules and of avoiding fuel depletion. We present a binary counter that recycles molecules efficiently while incurring just a moderate slowdown compared to alternative counters that do not recycle strands. This counter is an n-bit binary reflecting Gray code counter that advances through 2^n states. In the strand displacement realization of this counter, the waste—total number of nucleotides of the DNA strands consumed—is $O(n^3)$, while alternative counters have $\Omega(2^n)$ waste. We also show that our n-bit counter fails to work correctly when $\Theta(n)$ copies of the species that represent the state (bits) of the counter are present initially. The proof applies more generally to show that a class of chemical reaction systems, in which all but one reactant of each reaction are catalysts, are not capable of computations longer than $\frac{1}{2}n^2$ steps when there are at least n copies.

1 Introduction

DNA strand displacement systems support simulation of logic circuits and DNA walkers, and can in principle support general purpose computation in an energy-efficient manner [6, 9, 10, 12, 16–19, 21]. The computations typically consume strands at all reaction steps. Catalyst strands are an exception in that they are not consumed during the course of a reaction, but are recycled to perform the same operation multiple times.

Can strand displacement systems recycle strands in more general ways? We show that the answer is yes: we describe chemical reaction system computations, and their strand displacement realizations, where recycling of strands significantly reduces waste and avoids fuel depletion while incurring just a moderate slowdown relative to comparable computations that do not recycle strands. Thus our title: less haste, less waste. Our recycling computations are binary counters—simple and yet fundamental constructs in computation. A new feature of our strand displacement constructions is a mutex synchronization primitive, which ensures that reactions proceed atomically in the sense that all products of one reaction have been released before the next starts. The second contribution of the paper is to demonstrate a limit to recycling: recycling is not possible in certain classes of strand displacement systems that should work correctly even when many copies of the initial state of the system are present in the same environment.

The rest of this introduction illustrates the concept of strand recycling and gives an overview of our results and related work. Sections 2 and 3 then provide the technical details of the strand-recycling counters and the limits of recycling.

1.1 On the potential for strand recycling

We illustrate the concept of recycling using a 3-bit counter that is specified as a chemical reaction system—details of a strand displacement implementation are in Section 2. The counter follows the sequence of bit values shown in the left column of Fig. 1(a). The counter is a binary reflecting Gray code counter [5, 4]. A Gray code counter advances in such a way that exactly one bit changes at each step. A binary reflecting Gray code counter gets its name from the following property: if the states of the counter are written in a column starting from $0_n 0_{n-1} \ldots 0_1$ and a line is drawn just below row 2^{i-1}, where bit i changes from 0_i to 1_i, then in the next 2^{i-1} rows the values of the low order $i-1$ bits are the reflection of those above the line. For example, consider bits b_2 and b_1 of the 3-bit sequence in Fig. 1(a): the last four rows are a reflection of the first four rows. We call the resulting sequence of states the *Gray code sequence*.

Fig. 1(b) gives the chemical reaction system for this counter, which we call GRAY. The state of the 3-bit GRAY counter is determined by three *signal* molecules, one per bit. Presence of a single copy of signal 0_i denotes that the i^{th} bit has value 0 while presence of a single copy of 1_i denotes that the i^{th} bit has value 1. The initial counter state is $0_3 0_2 0_1$ and the reactions ensure that exactly one of 0_i and 1_i is present at any time. The counter advances through application of the three reversible chemical reactions (1-3) of Fig. 1(b). Each row of the table in Fig. 1(a) lists the reaction needed to produce the subsequent row; for example, the counter advances from $0_3 0_2 1_1$ to $0_3 1_2 1_1$ via reaction (2) in the forward direction (2-for).

In realizing these reactions with strand displacement systems (see Section 2), additional strands that are not shown in the chemical reaction system are consumed and produced; we call these additional strands *transformers*. For example, transformers might serve to ensure that all reactants are available before any product is produced, or may be side-products of a reaction that have no further use. Suppose that a set of strands $\mathcal{T}_i^f$ is consumed and a set $\mathcal{T}_i^r$ is produced when reaction (i) takes place in the forward direction; conversely $\mathcal{T}_i^r$ is consumed and $\mathcal{T}_i^f$ is produced when reaction (i) takes place in the reverse direction.

The key point is that in most of the rows, the signal molecules and transformer strands that are consumed were produced by reactions of earlier rows and are thus recycled. For example, in the third step the counter advances from state $0_3 1_2 1_1$ to $0_3 1_2 0_1$, uses reaction (1) in the reverse direction (1-rev) and consumes the signal molecule 0_1 and the set of transformer strands $\mathcal{T}_1^r$ that were produced in the first step, thereby recycling $\mathcal{T}_1^r$. Only in the three rows $0_3 0_2 0_1$, $1_3 1_2 0_1$ and $0_3 1_2 0_1$—precisely those rows when a reaction occurs for the first time—the molecules consumed are not produced in earlier rows. In contrast, a chemical reaction system for a standard binary counter produces waste molecules at every step and these waste molecules are never recycled in subsequent steps.

Recycling in DNA strand displacement systems offers the potential of supporting energy-efficient DNA computations in which the waste, or number of strands consumed, is logarithmic in the length of the computation. Systems that recycle strands do not use fuel, *i.e.,* large concentrations of certain transformer species that bias reactions in one direction, and so are not prone to problems of fuel depletion or fuel leakage. However, such advantages come at a price: our counter proceeds somewhat more slowly—is less hasty—than comparable fuel-driven strand displacement counters. The

(a)

b_3	b_2	b_1	Reaction	Consumed	Produced
0_3	0_2	0_1	(1-for)	$\mathcal{T}_1^f$	$\mathcal{T}_1^r$
0_3	0_2	1_1	(2-for)	$\mathcal{T}_2^f$	$\mathcal{T}_2^r$
0_3	1_2	1_1	(1-rev)	$\mathcal{T}_1^r$	$\mathcal{T}_1^f$
0_3	1_2	0_1	(3-for)	$\mathcal{T}_3^f$	$\mathcal{T}_3^r$
1_3	1_2	0_1	(1-for)	$\mathcal{T}_1^f$	$\mathcal{T}_1^r$
1_3	1_2	1_1	(2-rev)	$\mathcal{T}_2^r$	$\mathcal{T}_2^f$
1_3	0_2	1_1	(1-rev)	$\mathcal{T}_1^r$	$\mathcal{T}_1^f$
1_3	0_2	0_1			

(b)

(1) $\mathbf{0}_1 \rightleftharpoons \mathbf{1}_1$
(2) $\mathbf{0}_2 + 1_1 \rightleftharpoons \mathbf{1}_2 + 1_1$
(3) $\mathbf{0}_3 + 1_2 + 0_1 \rightleftharpoons \mathbf{1}_3 + 1_2 + 0_1$

Fig. 1. (a) Enumeration of counter states (left three columns), reaction that advances the state from one row to the next and its direction—forward (for) or reverse (rev)—and sets of transformer molecules consumed and produced. (b) Chemical reaction system for a three-bit binary reflecting Gray code counter. Species that appear on just one side of the reaction are shown in boldface and correspond to the bit that changes value as a result of the reaction. To ensure correctness, additional "catalyst" species appear on both sides and the corresponding bits are unchanged. At any step, only one reaction is applicable to advance the counter, although since the reactions are reversible the counter could also retreat to its previous value.

slowdown is due in part to the fact that reactions are used in both directions. Thus, our GRAY counter is not biased to advance towards the final state but rather performs an unbiased random walk, both advancing and retreating, ultimately reaching the final state. We also describe a counter, GRAY-FO that uses reactions in which the reactions are of fixed order, *i.e.,* the maximum number of reactants and products in any reaction are fixed, independent of the number of counter bits.

Table 1 summarizes properties of our counters and compares with another counter, which we call QSW, based on work of Qian *et al.* [11] (see Section 1.3). The properties considered are (i) *order* or max number of reactants or products of chemical reactions that describe the counter, (ii) *waste* or total number of nucleotides needed to implement the counter and (iii) *haste* or expected time for the counter to reach a designated final state from its initial state, when the volume equals the waste. Our n-bit binary reflecting Gray code counter, GRAY, uses reactions of order $\Theta(n)$, generates only $\Theta(n^3)$ waste and uses expected time $\Theta(n^3 2^{2n})$ to reach the final state. Our GRAY-FO counter improves on the GRAY counter in that the reaction order is $\Theta(1)$. The QSW counter also has reaction order $\Theta(1)$ and has expected time $\Theta(2^{2n})$, which is somewhat better than the expected time needed by our counters. However, the QSW counter generates $\Theta(2^n)$ waste, exponentially worse than our counters. All three counters are deterministic in that they advance and retreat through a predetermined linear ordering of states.

1.2 On the limits of strand recycling

The proof that our n-bit GRAY counter advances correctly through 2^n states assumes that only single copies of initial species are present. In Section 3, we show that if $\Theta(n)$ copies of the initial species are present, then the counter does not advance properly in a very strong sense: the final state of the counter can be reached in just $O(n^2)$ chemical reactions, rather than using the intended sequence of 2^n reactions. More generally, un-

Properties	GRAY	GRAY-FO	QSW [11]
Reaction order	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$
Consumption (waste)	$\Theta(n^3)$	$\Theta(n^3)$	$\Theta(2^n)$
Exp. Time (haste)	$\Theta(n^3 2^{2n})$	$\Theta(n^3 2^{2n})$	$\Theta(2^{2n})$

Table 1. Comparison of n-bit counter implementations. The GRAY and GRAY-FO counters are described in Section 2. The QSW counter is based on the simulation of stack machines by strand displacement reactions of Qian *et al.* [11].

der some restrictions on the allowable chemical reaction systems, we prove that in any realization of the system with $\Omega(|S|)$ copies of the initial species, any species of the system can be generated in $O(|S|^2)$ reaction steps, where S is the number of distinct species. In particular, if the waste of such a chemical reaction system is logarithmic in the length of a valid computation, the system does not work correctly when many copies of the initial reactants are present.

1.3 Related work

In related work, Qian *et al.* [11] showed how to simulate a stack machine using strand displacements systems. A binary counter can be implemented via a stack machine; we call such a counter a QSW (Qian-Soloviechik-Winfree) counter and we compare its properties and resources with our counters in Table 1. Details are provided in Section 2.6.

Cardelli [8, 1] has shown how primitives that support concurrent models of computation, such as fork and join gates, can be implemented using strand displacement systems. Many of our techniques are similar to those of Cardelli's constructions: for example, our signal strands share a common toehold while the long domains are distinct, and we do not use branched structures. To effect an abstract chemical reaction with i reactants and i products, we use cascading of strand exchanges whereby the reactants are first absorbed (by transformer molecules) and products are then released by further strand exchanges. This order of events is similar to an i-way join followed by an i-way fork of Cardelli; it is similar also to the strand displacement realizations of i-way molecular reactions of Qian *et al.* [11]. A new feature of our constructions is the use of a *mutex* strand to ensure that the $(k+1)^{\text{th}}$ reaction of a deterministic computation does not proceed until all products of the k^{th} reaction have been produced.

Building on models of Winfree and Rothemund [20, 14], Reif *et al.* [13] studied a tile-based graph assembly model in which tiles may both adhere to and be removed from a tile assembly. In their self-destructible graph assembly model, the removal of tiles allows for the possibility of tile reuse. The authors demonstrate that tile reuse is possible in an abstract tile model, via a PSPACE-hardness result. Doty *et al.* [2] showed a negative result on tile reuse for an irreversible variant of the model of Reif *et al*.

Kharam *et al.* [7] describe a DNA binary counter in which bit values are represented using relative concentrations of pairs of molecules. This is very different than our work in this paper, where the values of bits (0 and 1) are represented by the absence or presence of certain signal molecules.

2 GRAY: A binary reflecting Gray code counter

Here we describe the chemical reaction network and strand displacement implementation of our GRAY counter, provide a proof of its correctness, and analyze its expected time (haste) and space usage (waste). We show how it can be modified to use only bi-molecular reactions, resulting in our fixed-order GRAY counter: GRAY-FO. We also compare our counters to the QSW counter.

2.1 Chemical reaction system for the GRAY counter

We generalize the 3-bit GRAY counter in Section 1.1 in the obvious manner to n-bits. The counter state is represented by n *signal* molecules, one per bit. Presence of signal molecule b_i denotes that the i^{th} bit has value b_i, for $b = 0$ or $b = 1$. Initially, the state is $0_n \ldots 0_2 0_1$. Each possible state of the counter represents a value in the GRAY code sequence. The counter is described abstractly by the following chemical reactions:

$$\text{(gc-1)}\ \mathbf{0}_1 \rightleftharpoons \mathbf{1}_1$$

$$\text{(gc-}i\text{)}\ \mathbf{0}_i + 1_{i-1} + 0_{i-2} + \ldots + 0_1 \rightleftharpoons \mathbf{1}_i + 1_{i-1} + 0_{i-2} + \ldots + 0_1,\ 2 \leq i \leq n$$

Lemma 1. *The above chemical reaction system ensures the GRAY counter, when in state v, can only advance to state v_{next}, or retreat to state v_{prev}, corresponding to the next or previous value in the GRAY code sequence, respectively, if each reaction is atomic, and all initial signal molecules exist as single copies.*

Proof. First, observe that at any state of the system, for each i, exactly one of the signals 0_i and 1_i is present (in an unbound state). Hence, at any state of the system only two reactions can be applied: (gc-1) and (gc-i), where i is the smallest index such that signal 1_{i-1} is present. Indeed, the reactions (gc-j), where $2 \leq j < i$, cannot be applied as, by the definition of i, signal 0_{j-1} is present, and hence, 1_{j-1} is not present. Similarly, the reactions (gc-j), where $j > i$, cannot be applied as signal 1_{i-1} is present, and hence, 0_{i-1} is not present. It follows that at any state of the system, the system can only progress in the forward or the backward directions. □

2.2 Strand displacement implementation of the GRAY counter

A strand displacement implementation of the GRAY counter requires a means to simulate the chemical reaction equations. Furthermore, the correctness of the counter is predicated on the assumption that each chemical reaction is *atomic*. Qian *et al.* [11] proposed a construction—hereafter called the QSW construction—that is capable of simulating bi-molecular, and higher-order, chemical reactions. Specifically, the construction can exchange a set of signals (the reactants) for another set of signals (the products) through a sequence of strand displacement events. Unfortunately, the construction is not atomic, since some product signals can start initiating other reactions before before all product signals are produced. However, the strand displacements do occur in a fixed order and all reactant signals are consumed before any product signal is produced. We exploit this fact to simulate atomicity.

In particular, we borrow the concept of *transactions* from digital computation—a group of operations either completes or does not complete in its entirety. We achieve this

with the use of a simple synchronization primitive: a *mutex*. The state of our counter is only defined when the mutex is available. This is analogous to processes blocking when attempting to read a memory location currently being written to by another. Let μ denote a single copy of a signal molecule representing the mutex. In any sequence of strand displacements representing a chemical reaction, μ is the first reactant to be consumed and the last product to be produced. Therefore only one reaction (transaction) can be in progress at any given time. When μ is next available, either all strand displacements in the sequence took place and the counter is in a new state—the transaction succeeded—or the counter is in the same state and the configuration of all molecules is exactly the same prior to the reaction beginning—the transaction failed. Since each reaction is implemented as a transaction, they *appear atomic*.

We will use only one type of toehold, and therefore we will not label toeholds in the figures below. All signals in the QSW construction are of the same form: a negative recognition domain ${}^{-}d$, followed by a toehold t, followed by a positive recognition domain ${}^{+}d$. The construction also uses auxiliary strands consisting of a single domain and a single toehold, and one (saturated) template strand initially bound to signal and auxiliary strands. We refer to the saturated template complex and associated auxiliary strands, collectively, as a transformer. An example of the signal molecules and the transformer associated with the forward direction of the reaction $\mathbf{0_1} \rightleftharpoons \mathbf{1_1}$ is given in Fig. 2.

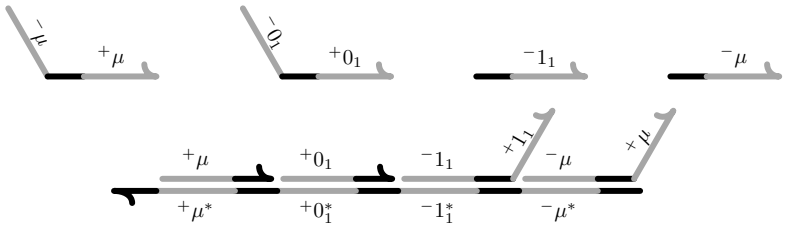

Fig. 2. An example of signal molecules (top two left strands) and the transformer, consisting of auxiliary strands (top two right strands) and a saturated template strand (bottom complex) associated with the forward direction of $\mathbf{0_1} \rightleftharpoons \mathbf{1_1}$. In this and later figures, the Watson-Crick complement of a domain x is denoted by x^*. The state of the system shown is $\mathbf{0_1}$.

As previously discussed, the reaction can only initiate if the signal molecule μ is present, and can only complete if all other reactants—in this case 0_1, assuming a forward reaction—are available. An example of the sequence of strand displacements for the reaction $\mathbf{0_1} \rightleftharpoons \mathbf{1_1}$ is given in Fig. 3. The reaction proceeds from top to bottom in the forward direction and from bottom to top in the backwards direction.

The transformers that implement the i^{th} reaction (gc-i) are a straightforward generalization of the first reaction. As before, the μ signal must initiate the first strand displacement, and is not produced until the last strand displacement. The number of required intermediate strand displacement reactions is dependent on the number of reactants and products. Specifically, the i^{th} reaction requires $2i + 2$ strand displacements to complete. An example of the transformer for the i^{th} reaction is given in Fig. 4.

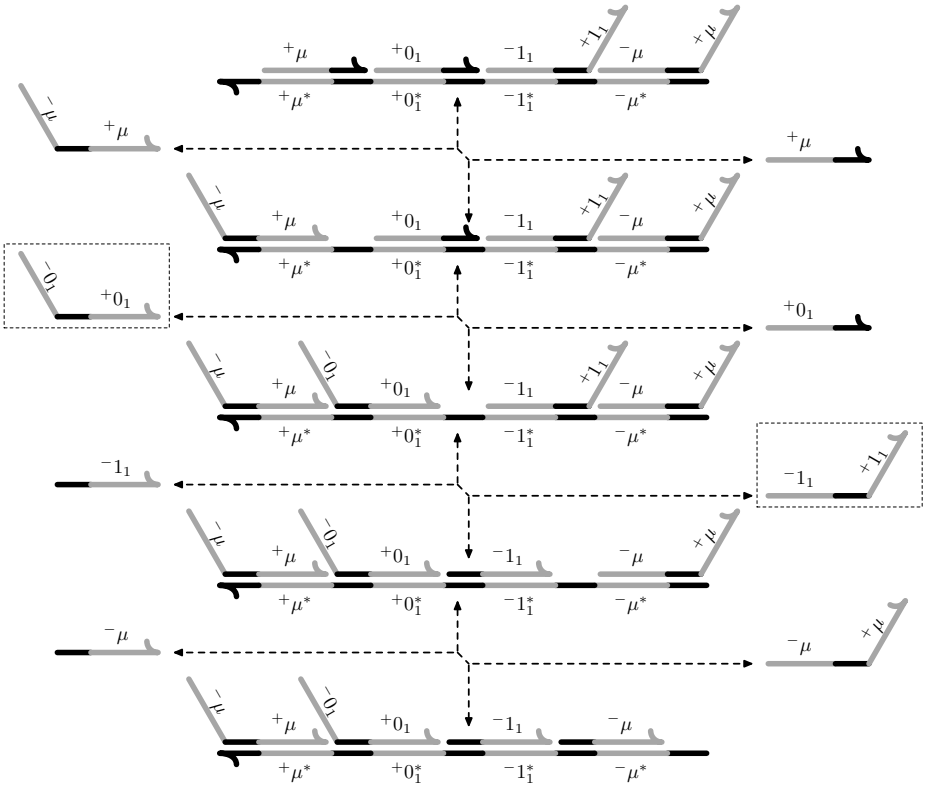

Fig. 3. The sequence of strand displacement events for the reaction $\mathbf{0_1} \rightleftharpoons \mathbf{1_1}$

2.3 Correctness

In the reactions of our counter, strand displacement should only happen when the toehold of the invading strand first binds to a free toehold, following which a domain of the invading strand displaces the bound domain of the strand being released. The invading and released domains should be identical. We say that such a strand displacement is *legal*. Illegal strand displacements can arise when the invading domain is different from the released domain; we call such displacements *mismatched domain displacements*. Illegal strand displacements can also arise due to blunt-end displacement, *i.e.,* displacements where invading and released domains are identical but domain displacement is not preceded by the binding of a free toehold, or when more than one invading domain strand displaces the strand being released. The next lemma shows correctness of the GRAY counter, assuming that only legal strand displacements can occur.

Lemma 2. *The above strand displacement implementation of the GRAY counter ensures all chemical reactions occur as transactions, and therefore appear atomic, assuming all initial signal molecules exist as single copies and all strand displacements are legal.*

Proof. We argue by induction on the sequence of chemical reactions. Prior to any chemical reaction beginning, we require the following invariant to hold: (i) for each digit,

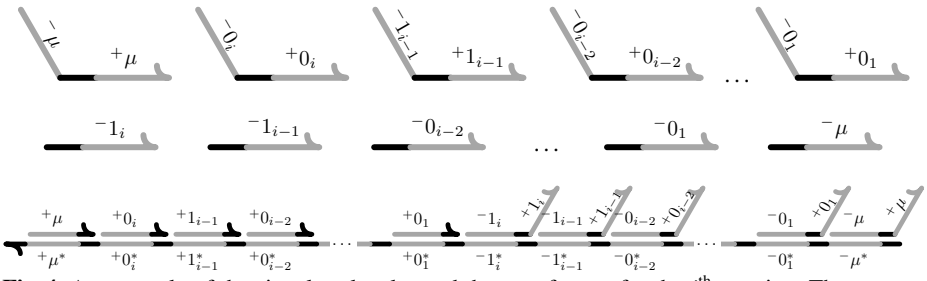

Fig. 4. An example of the signal molecules and the transformer for the i^{th} reaction. The counter is in state $b_n \dots b_{i+1} 0_i 1_{i-1} 0_{i-2} \dots 0_1$.

there is exactly one available signal which denotes its value, (ii) all template strands of all transformers are saturated, and require the mutex signal μ to initiate the first strand displacement, and (iii) there is exactly one available copy of μ. The invariant is trivially satisfied for the base case, when no reaction has yet occurred. Suppose the first $i-1$ reactions appear atomic, and the invariant is satisfied. Without loss of generality, suppose the next attempted reaction involves the k^{th} transformer.

Because we assume that all strand displacements are legal, no auxiliary strand or signal molecule representing the value of a digit can displace any strand in any transformer. Since there is exactly one available copy of the mutex signal, μ, that strand alone can initiate a reaction. Suppose the reaction is in the forward direction, as the reverse direction is symmetric. The signal μ must initiate the first strand displacement by binding to the left end of the k^{th} transformer's template strand. This begins the transaction. Note that there is another copy of μ sequestered at the right end of the template. When the signal μ is once again produced, there are two cases to consider.

Case 1. If the copy on the right end of the transformer is released, then the transaction succeeded and the counter is in a new state. Furthermore, the invariant is preserved as (i) exactly k signals that represent k different digits were consumed and exactly k signals corresponding to the same k digits were produced; (ii) the k^{th} transformer is saturated, and only a μ signal strand can initiate a new reaction on the right end of the template, and (iii) exactly one signal μ was produced as the final strand displacement.

Case 2. Otherwise, the original copy of μ was released, the transaction failed, and the counter is in the same state, satisfying the invariant, as any intermediate strand displacements must have been reversed prior to the original μ signal molecule being released.

Importantly, whether or not a transaction succeeds, while one is in progress no other reaction can be initiated since no μ signal is available. Thus, all reactions are implemented as transactions and appear atomic. □

We assume that blunt end displacement and displacement of a single strand by multiple strands do not occur; we do not know how to design strands so as to prevent such illegal displacements. However, we can ensure that the probability of mismatched domain errors is low by ensuring that the energy barrier of a mismatched strand displacement is high. Briefly, the rate at which one domain d displaces another domain d' is $2^{-\Omega(\mathsf{eb}(d,d'))}$, where $\mathrm{eb}(d, d')$ is the energy barrier required for d to displace d'. More concretely, suppose that all domains have the same length. We consider a simple energy model in which $\mathrm{eb}(d, d') = |d| - l$, where l is the length of the longest subsequence that

is common to both d and d'. Intuitively, if the bases in the longest common subsequence of d and d' form base pairs, then the energy barrier is the total number of base pairs lost when d displaces d'.

To ensure that the energy barrier between any pair of distinct domains is sufficiently high, we can use an error correcting code of Schulman and Zukerman [15]. They show how to construct a set of $2^{\Theta(n)}$ domains (*i.e.,* binary strings in their code) of equal length $\Theta(n)$ such that the energy barrier between any pair of domains is at least cn, for any given constant c. For our n-bit counter, we use a polynomial number of words in such a code for our signal domains (specifically, as explained in Lemma 3 the GRAY counter uses $\Theta(n)$ domains, and as explained in section 2.5, the GRAY-FO counter uses $\Theta(n^2)$ domains). We choose a sufficiently large constant c, so that the rate of mismatched domain displacements is at most $2^{-c'n}$, where c' is a constant that depends on c but is independent of n.

Since the expected time of the unbiased random walk that simulates our counter is $\Theta(n^3 2^{2n})$ (see the end of Section 2.4), the walk completes within time 2^{3n} with probability $1 - 2^{\Theta(n)}$. By choosing c so that $2^{-c'n} < 2^{-4n}$, we can conclude the probability of mismatched displacements occurring before the unbiased random walk that simulates our counter is completed is an exponentially small function of n.

2.4 Waste and haste analysis of the GRAY counter

Here we analyze the waste—the total number of nucleotide bases of all species consumed and haste—expected time—of the GRAY counter as it advances from initial to final states. We assume single copies of the initial signal, transformer, and mutex species. To analyze waste we first count the number of bases required for all initial signal, transformer, and mutex molecules.

Lemma 3. *The total number of nucleotide bases needed for a single copy of each initial signal, transformer, and mutex molecule of the n-bit GRAY counter is $\Theta(n^3)$.*

Proof. Each signal species 0_i and the initial mutex signal μ is composed of a toehold and two long domains. The same is true of the molecules for states 1_i and the sequestered μ signals that are part of the initial transformer species. There is an auxiliary transformer strand species consisting of one toehold and one long domain for each type of signal species. As noted in Section 2.3, to avoid mismatched strand displacement, we choose the $\Theta(n)$ domains of the signal species according to the Schulman and Zukerman code [15], and so they have length $\Theta(n)$. We choose the toehold length to be $\Theta(1)$. Since the domain length dominates the toehold length, the total number of bases in all signal species and auxiliary strands is $\Theta(n^2)$.

The template molecules in the sets $\mathcal{T}_i^f$ and $\mathcal{T}_i^r$ have $\Theta(i)$ domains, which dominate their length, and thus the template molecules have length $\Theta(in)$. Thus, the total number of bases in all transformer molecules in the system is $\sum_{i=1}^{n} \Theta(in) = \Theta(n^3)$. □

The next lemma shows that just one copy of each signal, mutex, and transformer species is sufficient for the GRAY counter to advance from its initial to final states. The proof is omitted, but is straightforward.

Lemma 4. *Advancement of the GRAY counter from its initial to final states requires just one initial copy of the mutex μ, one initial copy of each signal species 0_i, and one initial copy of each transformer species $\mathcal{T}_i^f$, for each $i, 1 \leq i \leq n$.*

In summary, just one copy of each signal, mutex, and transformer molecule is needed for the counter. Moreover, the total number of bases of these molecules is dominated by those in the transformers and is thus $\Theta(n^3)$. Hence the waste is $\Theta(n^3)$.

Next consider the expected time (haste) for the counter to progress from its initial to final states. We assume that reactions occur in a volume of size $\Theta(n^3)$, since this is the total number of bases of species in the system. Each strand displacement step involves interaction between two species and thus the rate of each strand displacement step is $1/\Theta(n^3)$.

First, consider the *shortest path* from the initial state to the final state. On this path, each order-i reaction is applied 2^{n-i} times and involves $\Theta(i)$ strand displacements. Thus the total number of strand displacement steps along the shortest path is $\sum_{i=1}^{n} \Theta(i)2^{n-i} = \Theta(2^n)$.

Because each reaction is reversible, the system does not strictly follow the shortest path but rather proceeds as an unbiased random walk along this path. The expected number of steps for a random walk to reach one end of a length-$\Theta(2^n)$ path from the other is $\Theta((2^n)^2) = \Theta(2^{2n})$ [3]. Therefore, the expected number of strand displacement steps is $\Theta(2^{2n})$. Since each strand displacement step occurs at a rate of $1/\Theta(n^3)$, the overall expected time—the haste—is $\Theta(n^3 2^{2n})$.

2.5 A fixed order implementation of the GRAY counter

An n-digit GRAY counter can perform a computation having length exponential in n, while only generating waste polynomial in n. However, it relied on template strands containing $O(n)$ domains, each of length $O(n)$, resulting in an overall length of $O(n^2)$ bases. Synthesis of long nucleic acid strands is challenging, and the fidelity of synthesized strands generally decreases as sequence length increases. For this reason, it is desirable to bound the length of all strands in the system to $O(n)$ bases. We now briefly describe how a template strand from the GRAY counter consisting of $2i + 2$ domains, can be split into $i + 1$ template strands requiring 4 domains each, for any $i > 1$. The overall waste will only be increased by a constant, resulting in the same volume, and thus the same haste.

The transformation is straightforward and to simplify the description, we introduce some notation. Consider the (gc-i) reaction of the GRAY counter which has i reactants and i products:

$$\mathbf{0}_i + 1_{i-1} + 0_{i-2} + ... + 0_1 \rightleftharpoons \mathbf{1}_i + 1_{i-1} + 0_{i-2} + ... + 0_1$$

The previous implementation demonstrated that by using the QSW construction and introducing a mutex molecule μ—thus creating an order $i + 1$ reaction—chemical reactions occur as transactions and therefore appear atomic. Specifically, μ is first consumed, then 0_i, then, 1_{i-1}, and so on. Likewise, after all reactants are consumed, 1_i is first produced, then 1_{i-1}, and so on, until finally μ is produced. We denote a strand displacement implementation supporting a transaction of this type, which is initiated by consuming a mutex α, and terminated when producing a mutex β, by:

$$[\alpha + 0_i + 1_{i-1} + 0_{i-2} + ... + 0_1 \rightleftharpoons 1_i + 1_{i-1} + 0_{i-2} + ... + 0_1 + \beta]$$

In the case of the GRAY counter, $\alpha = \beta = \mu$. Our goal is to convert this order $i+1$ reaction into a cascade of $i+1$ bi-molecular reactions, while preserving the appearance of atomicity. Using the above notation, we implement the following reaction cascade:

$$\left.\begin{array}{llll} [\mu & + 1_{i-1} \rightleftharpoons 1_{i-1} & + \alpha_i^1] \\ [\alpha_i^1 & + 0_{i-2} \rightleftharpoons 0_{i-2} & + \alpha_i^2] \\ & \vdots & \\ [\alpha_i^{i-2} & + 0_1 \rightleftharpoons 0_1 & + \alpha_i^{i-1}] \end{array}\right\} \text{ catalysts checked in sequence}$$

$$\left.\begin{array}{l} [\alpha_i^{i-1} + 0_i \rightleftharpoons 1_i + \alpha_i^{i-1}] \\ [\alpha_i^{i-1} + 1_i \rightleftharpoons 1_i + \mu] \end{array}\right\} \text{ the } i\text{-th bit flipped and the mutex signal released}$$

The overall transaction has been split into a cascade of sub-transactions. Each sub-transaction is implemented as a bi-molecular reaction using the QSW construction (see Fig. 3). The first $i-1$ sub-transactions check, in sequence, that all $i-1$ catalysts are present. The mutex signal μ is consumed during the first check. The last two transactions will first perform the bit flip and then releases the mutex signal. Every sub-transaction, except the $(i-1)$-st and the i-th, produces a unique mutex that is required to initiate the next sub-transaction in the cascade. Upon successful completion of the first i sub-transactions in the cascade, the final sub-transactions occurs, producing the original mutex signal μ, and thus finalizing the overall transaction. The implementation works in the reverse direction in a similar way with the exception that the bit is flipped first and the mutex signal μ is released only after the presence of all catalysts have been verified. Using the above transformation for all higher-order reactions in the original GRAY counter implementation results in a new, fixed order counter, GRAY-FO.

2.6 The QSW counter

The stack machine construction of Qian *et al.* [11], which is based on strand displacements systems, can be used to implement a binary counter. We call such a counter a QSW (Qian-Soloviechik-Winfree) counter. An n-bit implementation advances deterministically through 2^n states and uses reactions of order 2 (some of which involve polymer extension reactions that realize the stack). The transformer molecules used in the strand displacement realizations of these reactions can serve as fuel, biasing the reaction so that the counter advances. We analyze the biased version of the counter; the unbiased version is slower. The expected number of reactions for the biased counter to advance to its final state is $\Theta(2^n)$. Each reaction consumes a constant number of molecules and so the overall expected consumption, or waste, is $\Theta(2^n)$. The expected time depends on the volume in which the reaction takes place. If all strands consumed are initially present in the reaction volume, then the volume is $\Theta(2^n)$ and thus each step takes expected time $\Theta(2^n)$, leading to an overall expected time of $\Theta(2^{2n})$. Alternatively, it may be possible that fuel concentrations could be replenished over the course of the reaction and waste molecules removed, in which case the volume could be as low as $\Theta(n)$ and the overall reaction time would be $\Theta(n2^n)$. How to replenish fuel or remove waste is not addressed by Qian *et al.*

3 Limits on strand recycling for multiple-copy systems

In this section, we show that certain classes of chemical reaction networks that efficiently recycle strands, or that can perform useful computations for time that significantly exceeds the number of signals, cannot work properly when multiple copies of the initial species are present. In particular, our GRAY counters do not work in a multi-copy setting.

The underlying problem is the representation of the state of the system as specific *combinations* of signals. If there are multiple copies of the system in the same reaction vessel—as would typically occur in a laboratory setting—then the states of the different copies may interfere with one another. To illustrate this point, we again consider the 3-bit GRAY counter. Initially, in a single copy of the construction, the signals $\{0_3, 0_2, 0_1\}$ denote the state $0_3 0_2 0_1$. Consider a two-copy system where the initial set of present signals is duplicated, yielding the multiset $\{0_3, 0_3, 0_2, 0_2, 0_1, 0_1\}$. As in the single copy case, assume reaction (1) occurs in the forward direction, followed by reaction (2) in the forward direction. The resulting multiset of signal molecules is $\{0_3, 0_3, 0_2, 1_2, 0_1, 1_1\}$. In the single copy case, we intend that reaction (1) in the reverse direction will occur next; however, given the current set of present signals in the two-copy case, reaction (3) in the forward direction could instead occur, resulting in the multiset $\{0_3, 1_3, 0_2, 1_2, 0_1, 1_1\}$. At this point, a copy of every signal species is present, and any reaction can occur, in either direction. Furthermore, the single copy case required *at least* seven reactions to produce the final state $1_3 0_2 0_1$, whereas the two-copy case can reach it in three. Crosstalk between the copies has broken the counter.

In the remainder of this section, we treat this problem formally. We define a *chemical reaction system* to be a tuple $\mathbf{C} = \langle S, \mathcal{R}, S_0, s_{\text{end}} \rangle$, where

- S is a set of (signal) molecule species.
- $\mathcal{R}$ is a set of *reaction equations*, where each $R \in \mathcal{R}$ is an ordered pair of sets of signal molecules. Intuitively, a reaction equation $R = (I, P)$ consumes the signal molecules in I as *inputs* and produces the signal molecules in P as *products*. Our formalism is directional to allow modeling non-reversible reactions; a reversible chemical reaction is modeled as two separate elements of $\mathcal{R}$, i.e., (I, P) and (P, I).
- S_0 is a multiset of signal molecules initially present.
- $s_{\text{end}} \in S$ is a signal molecule denoting the end of computation[1].

An *x-copy* version of $\mathbf{C}$, denoted $\mathbf{C}^{(x)}$, is obtained by duplicating the initial multiset S_0 x times, i.e., $\mathbf{C}^{(x)} = \langle S, \mathcal{R}, S_0^{(x)}, s_{\text{end}} \rangle$ where $S_0^{(x)}$ is a multiset consisting of x copies of S_0.

We formalize computations in $\mathbf{C}$ in the natural manner: Let ρ be a sequence of reactions $R_1, R_2, \ldots, R_m$ from $\mathcal{R}$, where each $R_i = (I_i, P_i)$. We define ρ to be a *trace* of $\mathbf{C}$ if ρ induces a corresponding sequence of multisets $S_0, S_1, \ldots, S_m$, with S_0 being the multiset of initial signal molecules in $\mathbf{C}$, and for all $1 \leq i \leq m$, we have both $I_i \subseteq S_{i-1}$ and $S_i = S_{i-1} - I_i + P_i$. (We use "$-$" and "$+$" to denote multiset subtraction and union.)

[1] A computation may have multiple final states. To model this situation, we can let s_{end} be produced in all final reactions, in addition to any other signals that may indicate the result of the computation.

The next definitions help delineate the class of chemical reaction systems for the main result of this section. For a reaction equation $R = (I, P)$, consider the signal molecules in $I - P$. We dub these input signals *proper*. (The other signal molecules, in $I \cap P$, function as catalysts—they are necessary for the reaction, but not consumed.) We define a set of reactions to be k-*proper* if k is the maximum number of proper inputs of all reactions in the set. Note that the GRAY counter system is 1-proper. Let $|S_0|$ be the number of distinct elements in the multiset S_0.

Theorem 1. *Let $\mathbf{C} = \langle S, \mathcal{R}, S_0, s_{end} \rangle$ be a 1-proper chemical reaction system. If there exists a trace that produces s_{end} in $\mathbf{C}$, then for the x-copy chemical reaction system $\mathbf{C}^{(\mathbf{x})}$ with $x \geq |S| - |S_0|$, there exists a trace that produces s_{end} in at most $(|S| - |S_0| + 1)(|S| - |S_0|)/2$ steps.*

Proof. Let $\rho = R_1, \ldots, R_m$ be a sequence of reactions that produces s_{end} in the (single-copy) system $\mathbf{C}$, and $S_0, \ldots, S_m$ be the corresponding sequence of multisets of signals. Let $S' = \bigcup_{0 \leq j \leq m} S_j$ denote the set of all signals that occur in the computation. Obviously, $S' \subseteq S$. Let $k = |S'| - |S_0| \leq |S| - |S_0|$ denote the number of molecule species produced by ρ that were not present initially.

The proof is by construction, constructing a trace of the appropriate length for the multi-copy system from the trace ρ for the single-copy system. The high-level structure of the proof is as follows: First, we project out from ρ the k reactions, in order, that first produce each of the new molecule species. From that sequence, we build a trace of the multi-copy system that is the concatenation of k phases. Each phase adds one more signal molecule to the multiset of signals molecules present, preserves the presence of all signal molecules previously produced, and "consumes" one copy of the initial signal molecules in S_0. The ith phase is at most i reactions long, so the total length of the trace is bounded by $\sum_{i=1}^{k} i = (k+1)k/2 \leq (|S| - |S_0| + 1)(|S| - |S_0|)/2$.

We now formalize the construction of the k phases. Let $s_1, \ldots, s_k$ be the sequence of signal molecule species from $S' - S_0$ in order of their first appearances in $S_1, \ldots, S_m$, and let index(s_j) be the position in ρ where s_j was first produced. In other words, $R_{\text{index}(s_j)}$ is the reaction that produced s_j for the first time.

The k phases are constructed to maintain two invariants:

1. After the jth phase, the multiset of signal molecules contains at least one copy of each signal molecule in $\{s_1, \ldots, s_j\}$.
2. The trace constructed so far has not relied on the existence of more than j copies of the initial signal molecules S_0.

The invariants are vacuously true initially (before any phases). Assuming they are true after $j-1$ phases, we construct the jth phase as follows: the first reaction in the phase is $R_{\text{index}(s_j)}$, the reaction that produced s_j for the first time. We know this reaction can be applied because all of $\{s_1, \ldots, s_{j-1}\}$ are available, as well as the jth copy of S_0. This guarantees that the multiset now contains s_j, and we have relied on only j copies of S_0. However, the reaction may have consumed its inputs. In particular, since the system is 1-proper, the reaction consumed at most 1 input signal molecule. If the reaction consumed 0 molecules, or if the 1 molecule is in S_0, the invariant is maintained and the phase ends. Otherwise, the reaction consumed some $s_{j'}$, where $j' < j$. To restore $s_{j'}$ to the multiset, we append the reaction $R_{\text{index}(s_{j'})}$ to the phase, which is guaranteed to be applicable by

the same reasoning. In turn, this reaction might consume an earlier molecule $s_{j''}$, with $j'' < j'$, necessitating appending $R_{\text{index}(s_{j''})}$ to the phase, etc. The sequence $j, j', j'', \ldots$ is strictly decreasing, so it can have length at most j, which bounds the length of the phase to be at most j reactions long. At the end of the phase, the invariants are preserved.

Concatenating the k phases produces a trace for the k-copy chemical reaction system $\mathbf{C}^{(k)}$, which produces all of $\{s_1, \ldots, s_k\}$ within $(k+1)k/2$ reactions. Since $k \leq |S| - |S_0|$, the result follows. □

Note that Theorem 1 is much stronger than our intuitive notion of crosstalk short-circuiting a computation. It states that with only a linear number of copies, any signal molecule can be produced in at most a quadratic length computation.

We can formalize the intuitive notion of short-circuiting. A system $\mathbf{C}$ is *x-copy-tolerant* if, for all $s \in S$, the length of the shortest trace to produce s in $\mathbf{C}$ and in $\mathbf{C}^{(x)}$ is the same. A system is *copy-tolerant* if it is *x-copy-tolerant* for all x.

With that definition, we have the following two corollaries based on the fact that if a 1-proper chemical reaction system is $|S|$-copy-tolerant, then s_{end} can be computed in $\mathbf{C}$ in the same number of steps as in $\mathbf{C}^{(|S|)}$, which is polynomial in $|S|$ by Theorem 1.

Corollary 1. *For any 1-proper chemical reaction system $\mathbf{C} = \langle S, \mathcal{R}, S_0, s_{end} \rangle$ that is $|S|$-copy-tolerant, if there is a computation that produces a given signal species s_{end} in $\mathbf{C}$, then there is a computation that produces s_{end} in $\mathbf{C}$ in $O(|S|^2)$ steps.*

Corollary 2. *Let $\mathbf{C} = \langle S, \mathcal{R}, S_0, s_{end} \rangle$ be a 1-proper, $|S|$-copy-tolerant chemical reaction system. Then the haste of any computation of $\mathbf{C}$ is bounded by a polynomial function of the waste.*

4 Conclusions

In this paper we have introduced the concept of recycling, or molecule reuse, in strand displacement systems and chemical reaction networks. Our n-bit GRAY counters effectively use recycling to step through 2^n states while consuming, or wasting, molecules whose total number of bases is $O(n^3)$. Our GRAY counter strand displacement constructions also introduce the use of a *mutex* strand to ensure that higher-level chemical reactions are executed atomically. Finally, we show limits to recycling: for example, signals representing the final state of our n-bit counter can be generated using just $O(n^2)$ reactions when $\Theta(n)$ copies of the initial species share the same volume.

One weakness of our counter construction is that the number of distinct domains needed is polynomial in n, the number of bits of the counter. In contrast, a QSW binary counter that is implemented via the stack machine of Qian *et al.* [11] uses a just a constant number of domains independent of n. Is it possible to construct an n-bit counter that combines the best of the GRAY and QSW counters, *i.e.,* generates waste that is polynomial in n and uses $O(1)$ distinct domains? More generally, can *all* computation be realized by strand displacement systems whose waste and haste are within a (small) polynomial factor of the space and time of the computation? Our negative result raises the question as to whether there are alternative strand displacement realizations of certain chemical reaction network classes that generate little waste, say logarithmic in the computation length, and that also behave correctly in the multi-copy setting. We will investigate these questions in future work.

Acknowledgements. Thanks to Bonnie Kirkpatrick and to the anonymous reviewers for their very helpful suggestions.

References

1. L. Cardelli. Strand algebras for DNA computing. *Natural Computing*, 10(1):407–428, 2001.
2. D. Doty, L. Kari, and B. Masson. Negative interactions in irreversible self-assembly. In *Proceedings of the Sixteenth Annual Conference on DNA Computing and Molecular Programming, Springer-Verlag Lecture Notes in Computer Science 6518*, pages 37–48, 2011.
3. W. Feller. *An Introduction to Probability Theory and Its Applications, Vol. 1*. Wiley, 1971.
4. E.N. Gilbert. Gray codes and paths on the n-cube. *Bell Systems Technical Journal*, 37:815–826, 1958.
5. F. Gray. Pulse code communications. U.S. Patent 2632058, 1953.
6. G. Hongzhou, J. Chao, S.-J. Xiao, and N.C. Seeman. A proximity-based programmable DNA nanoscale assembly line. *Nature*, 465:202–205, 2010.
7. A. Kharam, H. Jiang, M. Riedel, and K. Parhi. Binary counting with chemical reactions. In *Proceedings of the 2011 Pacific Symposium on Biocomputing*, pages 302–313. World Scientific Publishing, 2011.
8. L.Cardelli. Two-domain DNA strand displacement. In *Proc. of Developments in Computational Models (DCM 2010)*, volume 26 of *Electronic Proceedings in Theoretical Computer Science*, pages 47–61, 2010.
9. K. Lund, A.T. Manzo, N. Dabby, N. Michelotti, A. Johnson-Buck, J. Nangreave, N. Taylor, R. Pei, M.N. Stojanovic, N.G. Walter, E. Winfree, and H. Yan. Molecular robots guided by prescriptive landscapes. *Nature*, 465:206–210, 2010.
10. T. Omabegho, R. Sha, and N.C. Seeman. A bipedal DNA brownian motor with coordinated legs. *Science*, 324(5923):67–71, 2009.
11. L. Qian, D. Soloveichik, and E. Winfree. Efficient turing-universal computation with DNA polymers. In *Proceedings of the Sixteenth Annual Conference on DNA Computing and Molecular Programming, Springer-Verlag Lecture Notes in Computer Science 6518*, pages 123–140, 2011.
12. L. Qian and E. Winfree. A simple DNA gate motif for synthesizing large-scale circuits. *J. R. Soc. Interface*, 2011.
13. J.H. Reif, S. Sahu, and P. Yin. Complexity of graph self-assembly in accretive systems and self-destructible systems. In *Proceedings of the Eleventh Annual Conference on DNA-Based Computing, Springer-Verlag Lecture Notes in Computer Science 3892*, pages 257–275, 2006.
14. P.W.K. Rothemund and E. Winfree. The program-size complexity of self-assembled squares. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 459–468, 2000.
15. L.J. Schulman and D. Zuckerman. Asymptotically good codes correcting insertions, deletions, and transpositions. *IEEE Transactions on Information Theory*, 45:2552–2557, 1999.
16. G. Seelig, D. Soloveichik, D.Y. Zhang, and E. Winfree. Enzyme-free nucleic acid logic circuits. *Science*, 314(5805):1585–1588, 2006.
17. J.-S. Shin and N.A. Pierce. A synthetic DNA walker for molecular transport. *J Am Chem Soc*, 126:10834–10835, 2004.
18. D. Soloveichik, G. Seelig, and E. Winfree. DNA as a universal substrate for chemical kinetics. *Proc. Nat. Acad. Sci. USA*, 107(12):5393–5398, 2010.
19. S. Venkataraman, R.M. Dirks, P.W.K. Rothemund, E. Winfree, and N.A. Pierce. An autonomous polymerization motor powered by DNA hybridization. *Nature Nanotech*, 2(8):490–494, 2007.
20. E. Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, Caltech, 1998.

21. D.Y. Zhang and G. Seelig. Dynamic DNA nanotechnology using strand displacement reactions. *Nature Chemistry*, 3:103–113, 2011.