Building Reconfigurable Circuitry in a Biochemical World

Hui-Ju Chiang, Jie-Hong Jiang, François Fages

Building Reconfigurable Circuitry in a Biochemical World

Hui-Ju Katherine Chiang*†, Jie-Hong R. Jiang*, and François Fages†
*Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan
†Inria Paris-Rocquencourt, France
Email: {d01943033@ntu.edu.tw; jhjiang@ntu.edu.tw; Francois.Fages@inria.fr}

***Abstract*—Realizing complex systems within a biochemical environment is a common pursuit in synthetic biology, an emerging technology with promising potential in biomedicine and other applications. Such systems achieve certain computation through properly designed biochemical reactions. Despite fruitful progress being made, most existing reaction designs have fixed target functionality. Their lack of reconfigurability can be disadvantageous, especially when a system has to adapt to a varying biochemical environment. In this paper, we propose an analog approach to economically construct a reconfigurable logic circuit similar to a silicon based field programmable gate array (FPGA). The effective "logic" and "interconnect" of the circuit can be dynamically reconfigured by controlling the concentrations of certain knob species. We study a potential biomedical application of our reconfigurable circuitry to disease diagnosis and therapy at a molecular level.**

I. Introduction

A synthetic approach to biology has been shown useful in biomedicine, energy, environment, and other applications. The advancements of synthetic biology have been broadening the range of realizable systems of increasing complexity both *in vivo* and *in vitro*. Building systems within a biochemical world is not far from reach and has been intensively studied, e.g., in terms of digital logic operations [6], [8], [14], analog computation [4], linear control [2], [10], signal processing [9], program flow control [7], etc. The bio-compatibility of such systems is unique in that they can embed computation tasks, including sensing, information processing, and actuation, inside living cells without physical intrusion. They are thus attractive in biomedical applications in disease diagnosis and therapy at a molecular level.

Most, if not all, of the engineered biochemical systems mentioned above have fixed specific functions or parametric values, and cannot be changed after design. This prespecification of functions or parameters can be disadvantageous when the underlying environment evolves over time with uncertainty or when the intended system behavior cannot be fully determined in the design phase. Even for electronic system design, which is very predictable, it is still not uncommon that a design has to be rectified even after it is manufactured. Likewise in biochemical system design, reconfigurability can be beneficial and crucial especially for biochemical environments, which are intrinsically stochastic. While the reconfigurability of integrated circuits (ICs) can be achieved through embedding firmware or programmable gate arrays into the design, it remains unclear how a similar mechanism can be economically deployed in a biochemical design.

In this paper, we propose a reconfigurable system, comprised of configurable logic units and interconnects, which are built from biochemical reactions. Our construction is advantageous in the following three respects. First, a configurable logic unit is made out of just a few reactions and species based on analog computation. Second, the function of a logic unit can be easily configured by altering the concentrations of certain biochemical species, similar to how organisms adapt their inner functions according to environmental signals received. Third, our construction maintains modularity and composability. The *retroactivity* [13] issue is overcome in the system, that is, composing a system with an extra module cannot invalidate the system's behavior. Finally, we discuss candidate implementation techniques and study the potential use of our reconfigurable circuitry in biomedical applications. Simulation results suggest the feasibility of our methods.

II. Preliminaries

This paper adopts the *classical chemical kinetic (CCK) model* of biochemical reactions. It is assumed that molecules involved in reactions are of *large quantities*, so the spatial non-uniformity of molecule distribution becomes negligible, and the intrinsically stochastic reactions can be safely assumed to happen continuously and deterministically. Under the assumption, the attempt to use a set of ordinary differential equations (ODEs) to approximate the dynamic behavior of a biochemical system is justified.

Consider the following biochemical reaction

$$\sum_{i=1}^{n} \alpha_i r_i \xrightarrow{k} \sum_{j=1}^{m} \beta_j p_j,$$

where species r_i is the i^{th} reactant and p_j the j^{th} product, coefficients α_i's and β_j's specify the stoichiometric amounts, and k is the rate constant. The species' dynamics under the CCK model is:

$$k \prod_{i=1}^{n} [r_i]^{\alpha_i} = -\frac{1}{\alpha_i}\frac{d[r_i]}{dt} = \frac{1}{\beta_j}\frac{d[p_j]}{dt},$$

where $[p_j]$ represents the concentration of species p_j.

In the sequel, to simplify notation, we do not distinguish a species and its concentration when they are clear from the context. Moreover, from a system perspective, we treat a species as a signal and the concentration of a species as the (non-negative) value of the corresponding signal.

III. Reconfigurable Circuitry

Our reconfigurable circuitry consists of two kind of components: configurable logic units (Sec. III-A) and configurable interconnects (Sec. III-B). Each logic unit (similar to those in silicon FPGAs) has k input ports/species and one or multiple output ports/species. It can realize a certain set of logic functions up to k inputs. (In our discussion we set $k = 2$ and let the realizable functions be AND, OR, XOR, and NOT.) The logic

978-1-4799-2346-5/14/$31.00 ©2014 IEEE

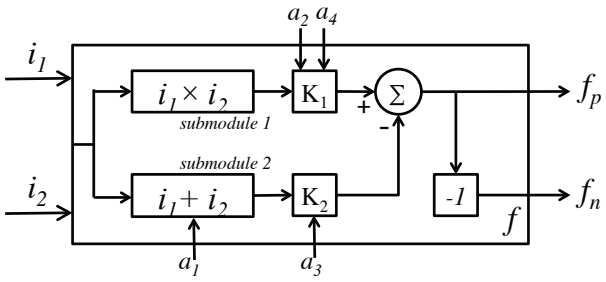

Fig. 1. Block diagram of configurable logic unit.

units can be composed through configurable interconnects. The construction of configurable logic units and interconnects is detailed as follows.

A. Configurable Logic Units

The configurable logic unit that we propose is realized through the following equations of arithmetic over reals.

$$\begin{cases} \text{AND}(i_1, i_2) &= -(0 \times (i_1 + i_2) - 1 \times (i_1 \times i_2)) & (1) \\ \text{OR}(i_1, i_2) &= +(1 \times (i_1 + i_2) - 1 \times (i_1 \times i_2)) & (2) \\ \text{XOR}(i_1, i_2) &= +(1 \times (i_1 + i_2) - 2 \times (i_1 \times i_2)) & (3) \\ \text{NOT}(i_1) &= \text{XOR}(i_1, 1) & (4) \end{cases}$$

The computation is depicted in the block diagram of Fig. 1. Two quantities, $(i_1 + i_2)$ and $(i_1 \times i_2)$, are common to the construction of all four considered logic functions, which differ only in the coefficients combining these two quantities and in the final sign. Assuming that the inputs i_1 and i_2 take on either 0 or 1 unit of concentration (signifying Boolean 0 or 1 logic value, respectively), one can verify that the four equations correspond to the four intended logic interpretations. In essence, the logic operations are achieved through arithmetic over reals, i.e., some form of analog computation, which can be more economical than the digital counterpart [4]. Notice that the definition of unit concentration is relative, and 0 and 1 units of concentration do not need to be exact; slight deviations in concentration from 0 and 1 are immaterial to the correctness of the interpretation.

Below we show how to implement the above four equations in terms of biochemical reactions. Essentially the four equations are *implemented* by the same set of reactions such that the output value of a configured logic unit coincides with the concentration at equilibrium of some designated species in the reactions. According to the block diagram of Fig. 1, the set of biochemical reactions is comprised of four groups:

(a) Reactions implementing Submodule 1 in Fig. 1:

$$\begin{cases} i_1 + i_2 \xrightarrow{k_1} i_1 + i_2 + y_1 & (a.1) \\ y_1 \xrightarrow{k_2} \emptyset & (a.2) \end{cases}$$

(b) Reactions implementing Submodule 2 in Fig. 1:

$$\begin{cases} a_1 + i_1 \xrightarrow{k_3} a_1 + i_1 + y_2 & (b.1) \\ i_2 \xrightarrow{k_4} i_2 + y_2 & (b.2) \\ y_2 \xrightarrow{k_5} \emptyset & (b.3) \end{cases}$$

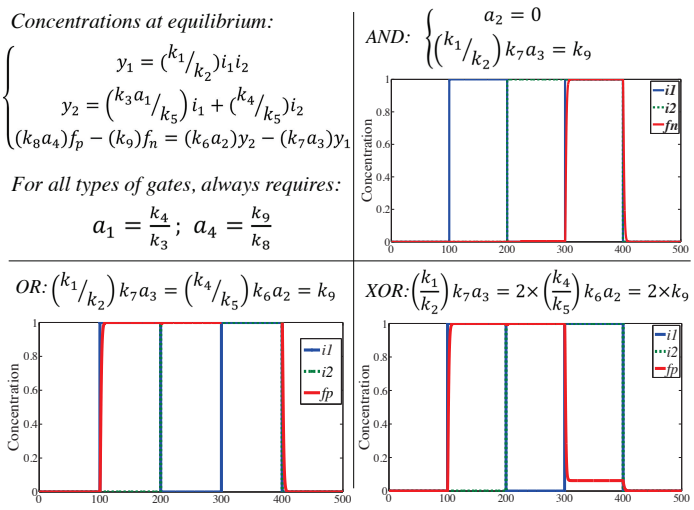

Fig. 2. Concentration settings and simulation results of the reconfigurable logic unit.

(c) Reactions implementing linear combination:

$$\begin{cases} a_2 + y_2 \xrightarrow{k_6} a_2 + y_2 + f_p & (c.1) \\ a_3 + y_1 \xrightarrow{k_7} a_3 + y_1 + f_n & (c.2) \\ a_4 + f_p \xrightarrow{k_8} a_4 & (c.3) \\ f_n \xrightarrow{k_9} \emptyset & (c.4) \end{cases}$$

(d) Reaction implementing output aggregation:

$$f_p + f_n \xrightarrow{K} \emptyset \qquad (d.1)$$

Given the above reactions, we are concerned with their equilibriums that we analyze as follows. Reactions $(a.1)$ and $(a.2)$ at equilibrium induce $y_1 = (k_1/k_2)(i_1 \times i_2)$ since

$$\frac{dy_1}{dt} = k_1 i_1 i_2 - k_2 y_1 = 0.$$

Reactions $(b.1)$, $(b.2)$, and $(b.3)$ at equilibrium induce $y_2 = (k_3 a_1/k_5) i_1 + (k_4/k_5) i_2$. Note that, in reaction $(b.1)$, a_1 serves as an auxiliary species, whose purpose is to discharge the stringent rate matching to ensure $k_3 = k_4$. With the presence of species a_1, reaction rates k_3 and k_4 obey $k_3 a_1 = k_4$, which can be easily satisfied since a_1 is a species with its concentration tunable externally. That is, we let $a_1 = k_4/k_3$. Reactions $(c.1)$, $(c.2)$, $(c.3)$, and $(c.4)$ at equilibrium induce $(k_8 a_4) f_p - (k_9) f_n = (k_6 a_2) y_2 - (k_7 a_3) y_1$. Similarly, a_2, a_3, a_4 are auxiliary species whose concentrations can be controlled externally. Specifically, we let $a_4 = k_9/k_8$, and let the concentrations of a_2 and a_3 be determined depending on the intended logic function (to be discussed). Effectively, species a_2 and a_3 serve as control knobs for function configuration. Finally, assuming K much larger than other rate constants k_1, ..., k_9, reaction $(d.1)$ enforces one of output species f_p and f_n to have concentration 0 and the other to have concentration $|f_p - f_n|$.

By the above reactions, the function of a configurable logic unit can be altered by controlling the concentrations of species a_2 and a_3. Specifically, to configure an AND function, we set $a_2 = 0, a_3 = (k_2 k_9)/(k_1 k_7)$ so that at equilibrium the output f_n equals $\text{AND}(i_1, i_2)$. To configure an OR function, we set

$a_2 = (k_5k_9)/(k_4k_6), a_3 = (k_2k_9)/(k_1k_7)$ so that at equilibrium the output f_p equals $\text{OR}(i_1, i_2)$. To configure an XOR function, we set $a_2 = (k_5k_9)/(k_4k_6), a_3 = 2(k_2k_9)/(k_1k_7)$ so that at equilibrium the output f_p equals $\text{XOR}(i_1, i_2)$. On the other hand, NOT function can be built from XOR. Therefore once inputs i_1, i_2 are assigned to their respective 0 or 1 values, the output converges to 0 or 1 automatically when the above reactions reach their equilibriums. Fig. 2 summarizes the concentration requirements and shows the simulation results under input sequence $(i_1, i_2) = (0, 0), (0, 1), (1, 0), (1, 1), (0, 0)$ in a time separation of 100 units.

B. Configurable Interconnects

In addition to the four auxiliary input species, each configurable logic unit has two input i_1, i_2 and two output f_p, f_n ports. These ports allow interconnection among multiple configurable logic units, and thus allow arbitrary composition of logic units to realize any Boolean function. We show how an interconnect can be made configurable as follows.

To have a configurable connection between a source port/species s and a destination port/species d, we introduce a unique wiring species w_{sd} for the pair such that s and d are connected (i.e., $[d]$ stabilizes to $[s]$ with negligible delay) if w_{sd} is of value 1 (one unit concentration) and disconnected (i.e., $[d]$ resets to 0 regardless of $[s]$) if w_{sd} is of value 0 (zero concentration). The reactions that fulfill this connection are:

$$\begin{cases} s + a_{sd} + w_{sd} \xrightarrow{k_1} d + s + a_{sd} + w_{sd} & (e.1) \\ d \xrightarrow{k_2} \emptyset & (e.2) \end{cases}$$

where a_{sd} is an auxiliary species making $a_{sd} \times k_1 = k_2$ to discharge the need of rate matching of $k_1 = k_2$, and the second reaction resets the destination species d to 0.

Notice that, unlike the well isolation of a signal in electronic circuits, a signal/species in a biochemical circuit without compartmental isolation is globally seen by all reactions. It is therefore necessary for each signal to be realized by a unique species. Note also that the retroactivity issue, similar to the *loading* effect in electronic circuits, is overcome in our construction by two means. First, the amount of an up-stream species is not affected by composing it with a down-stream species. For example, in reaction $(e.1)$, up-stream species s appears both as a reactant and a product with the same stoichiometric amount. Hence the amount of s remains intact under the presence of reaction $(e.1)$ for the creation of down-stream species d. The same principle is applied to retain the amounts of species $i_1, i_2, a_1, a_2, a_3, a_4$ in the reactions $(a.1)$, $(b.1)$, $(c.1)$, $(c.2)$. Second, we sustain the concentration of a species that can be consumed or produced by some reactions at its intended value based on equilibrium. For example, the concentrations of f_p and f_n remain at their equilibrium values due to the fact that the equilibriums of y_1 and y_2 are ensured by reaction groups (a) and (b) since no other reaction involves y_1 and y_2. Hence in the equilibrium equation $(k_8a_4)f_p - (k_9)f_n = (k_6a_2)y_2 - (k_7a_3)y_1$, the right-hand side is a constant and so are the values of f_p and f_n on the left hand side. (Species a_2, a_3, a_4 have determined constant concentrations.) Thereby our established modularity and composability ensure robust system construction.

C. Logic Synthesis

Given an arbitrary Boolean function, it can be realized with biochemical reactions by mapping it into our proposed reconfigurable architecture, similar to the conventional FPGA technology mapping in electronic design. Moreover, our approach well supports *reconfigurable computing* [3], [12] in biochemical systems. It is possible to conduct multiple computation tasks on the same circuitry in a time-multiplexed fashion and may have unique application in biochemical systems.

IV. Case Study

A microRNA (miRNA) is a small, highly conserved non-coding RNA that involves in almost every cellular process and down-regulates gene expressions through partial base-pairing with its (multiple) messenger RNA (mRNA) targets. Inappropriate miRNA expressions have been linked to the regulation and progression of a wide range of diseases [1], such as numerous cancers, cardiovascular, neurological, immunological, and metabolic diseases. Early onset of those diseases can be detected by monitoring changes in miRNA expression levels. Due to the *partial* base-pairing during target recognition, the regulation relation between miRNAs and mRNAs is *many-to-many*. As a result, diagnosis of certain diseases may involve multiple miRNAs and complex decision conditions, which may be expressible in Boolean formulae.

For potential implementation of our proposed biochemical reactions, there is recent demonstration of oligonucleotide AND-gates that can respond to specific miRNA inputs in live mammalian cells [6]. Moreover, *DNA strand displacement* [11], [15] has been successful in implementing various chemical reaction networks. These techniques may bring promise to the feasibility of conducting Boolean operations on miRNA inputs, recognizing endogeneuos miRNA expression patterns, and generating different oligonucleotide outputs correspondingly to manipulate miRNA levels for therapeutic purposes.

As reconfigurable circuitry may conduct different computation tasks utilizing the same set of reactions, it may realize different diagnostic and therapeutic strategies whichever one is needed. As a thought example, we consider function switching between two diagnostic-therapeutic specifications expressed in two Boolean expressions f_1 and f_2:

$$f_1 = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (x_3 \vee x_4)$$
$$f_2 = (x_1 \wedge x_2 \wedge x_3) \vee (\neg x_2) \vee (\neg x_1 \wedge x_4)$$

where $\wedge$, $\vee$, $\neg$ are Boolean connectives conjunction (and), disjunction (or), and negation (not), respectively. Imagine that each variable x_i represents a distinct type of miRNA related to the diagnostic tasks at hand. Let f_1 and f_2 encode the therapeutic actions corresponding to the diagnostic tests of diseases A and B, respectively. When disease A (resp. B) is in consideration, the reconfigurable circuitry implements f_1 (resp. f_2) function. The function output may be coupled with some miRNA whose expression level is to be raised for disease treatment.

A schematic diagram implementing the above two functions is shown in Fig. 3, where the gates correspond to the configurable logic units introduced in Sec. III-A, the four side-inputs to a gate indicate the auxiliary inputs, and the dashed boxes correspond to the configurable interconnects. For simplicity, here the configurability of interconnects is only limited to certain port to port connections. To implement functions f_1 and f_2 on the circuit shown, the inputs $(l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9)$ are assigned $(x_1, x_2, 0, \neg x_1, x_2, \neg x_3, x_3, x_4, 0)$ for f_1, and

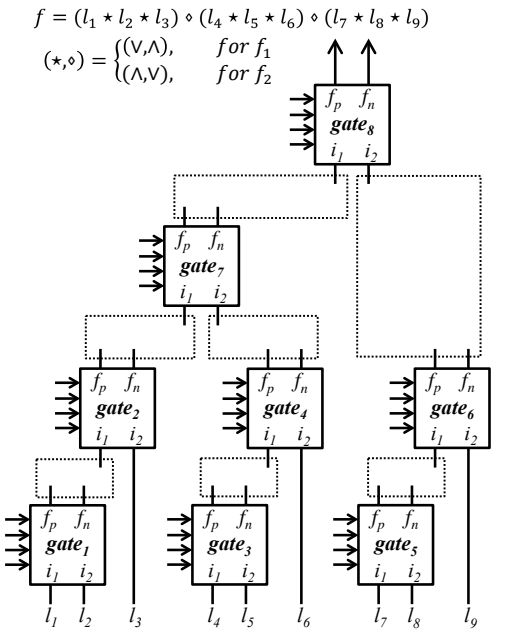

Fig. 3. A circuit diagram implementing functions f_1 and f_2.

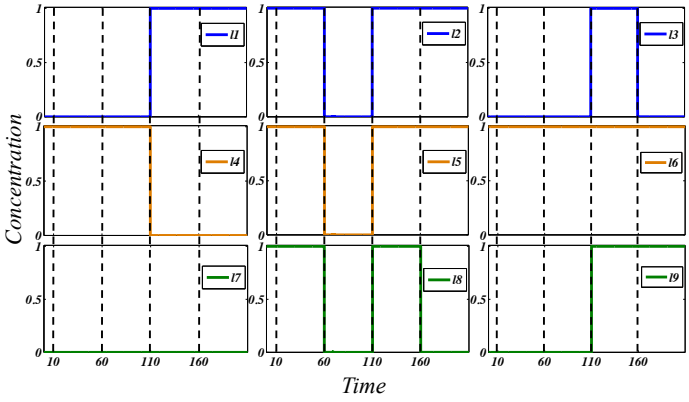

Fig. 4. Waveforms of inputs $l_1, l_2, \ldots, l_9$.

$(x_1, x_2, x_3, \neg x_2, 1, 1, \neg x_1, x_4, 1)$ for f_2. Gates 1 to 6 implement the part of logic *inside* the parentheses in the formulae of f_1 and f_2, while gates 7 and 8 implement the logic operations that connect between parentheses. So for function f_1, the f_p ports of gates 1 to 6 (which are of type OR) hold the evaluation results of formulae $(x_1 \vee x_2)$, $(x_1 \vee x_2)$, $(\neg x_1 \vee x_2)$, $(\neg x_1 \vee x_2 \vee \neg x_3)$, $(x_3 \vee x_4)$, and $(x_3 \vee x_4)$, respectively; the f_n ports of gates 7 and 8 (which are of type AND) hold the evaluation results of $(x_1 \vee x_2) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$ and $(x_1 \vee x_2) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (x_3 \vee x_4)$, i.e., function f_1, respectively. On the other hand, for function f_2, the f_n ports of gates 1 to 6 (which are of type AND) hold the evaluation results of $(x_1 \wedge x_2)$, $(x_1 \wedge x_2 \wedge x_3)$, $(\neg x_2)$, $(\neg x_2)$, $(\neg x_1 \wedge x_4)$, and $(\neg x_1 \wedge x_4)$, respectively; the f_p ports of gates 7 and 8 (which are of type OR) hold the evaluation results of $(x_1 \wedge x_2 \wedge x_3) \vee (\neg x_2)$ and $(x_1 \wedge x_2 \wedge x_3) \vee (\neg x_2) \vee (\neg x_1 \wedge x_4)$, i.e., function f_2, respectively.

The above reconfigurable circuit is simulated using BIOCHAM [5]. The input waveforms and resultant output waveforms are shown in Fig. 4 and Fig. 5, respectively. The configuration switches from function f_1 to function f_2 at time $t = 110$. After connection configuration is established at $t = 10$, the input values change every 50 time units, at $t = 60, 110, 160$, with input sequence $(x_1, x_2, x_3, x_4) = (0,1,0,1), (0,0,0,0), (1,1,1,1), (1,1,0,0)$, which imitates the change of miRNA expression patterns. The waveforms of $l_1, \ldots, l_9$ in response to the input sequence is shown in Fig. 4. The waveforms of f_1 and f_2 are shown in Fig. 5, which imitate the therapeutic responses to diseases A and B, respectively.

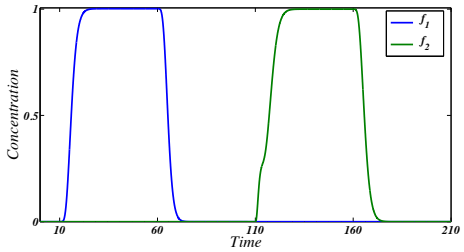

Fig. 5. Waveforms of outputs f_1 and f_2.

V. CONCLUSION

We presented a framework for building reconfigurable logic circuits using biochemical reactions. The reconfiguration can be easily done by controlling the concentrations of certain auxiliary species. In our construction, special attentions were paid to ensure modularity and composability. We studied potential biomedical application of our method in disease diagnosis and therapy at a molecular level. Our reconfigurable architecture may benefit system construction in synthetic biology for task switching in unpredictable environments.

REFERENCES

[1] J. A. Broderick and P. D. Zamore, "MicroRNA therapeutics," *Gene Therapy*, Vol. 18, No. 12, pp. 1104–1110, Apr. 2011.

[2] Y.-J. Chen et al., "Programmable chemical controllers made from DNA," *Nature Nanotechnology*, Vol. 8, pp. 755–762, Oct. 2013.

[3] K. Compton and S. Hauck, "An introduction to reconfigurable computing," *IEEE Computer*, Apr. 2000.

[4] R. Daniel et al., "Synthetic analog computation in living cells," *Nature*, Vol. 497, pp. 619–623, May 2013.

[5] François Fages et al., *BIOCHAM 3.5 Reference Manual*, Nov. 2013.

[6] J. Hemphill and A. Deiters, "DNA computation in mammalian cells: microRNA logic operations," *J. Am. Chem. Soc*, Vol. 135, No. 28, pp. 10512–10518, 2013.

[7] D.-A. Huang et al., "Compiling program control flows into biochemical reactions," in *Proc. ICCAD*, pp. 361–368, Nov. 2012.

[8] H. Jiang et al., "Digital logic with molecular reactions," in *Proc. ICCAD*, pp. 721–727, Nov. 2013.

[9] H. Jiang et al., "Digital signal processing with molecular reactions," *IEEE Design & Test of Computers*, Vol. 29, No. 3, pp. 21–31, 2012.

[10] K. Oishi and E. Klavins, "Biomolecular implementation of linear I/O systems," *IET Syst. Biol.*, Vol. 5, No. 4, pp. 252–260, 2011.

[11] D. Soloveichik et al., "DNA as a universal substrate for chemical kinetics," *PNAS*, Vol. 107, No. 12, pp. 5393–5398, 2010.

[12] T. J. Todman et al., "Reconfigurable computing: architectures and design methods," *Computer and Digital Techniques, IEE Proceedings*, Vol. 152, No. 2, pp. 193–207, Mar. 2005.

[13] D. Del Vecchio et al., "Modular cell biology: retroactivity and insulation," *Molecular Systems Biology*, Vol. 4, No. 161, 2008.

[14] L. Qian and E. Winfree, "Scaling up digital circuit computation with DNA strand displacement cascades," *Science*, Vol. 332, No. 6034, pp. 1196–1201, Jun. 2011.

[15] D. Y. Zhang and G. Seelig, "Dynamic DNA nanotechnology using strand-displacement reactions," *Nature Chem.*, Vol. 3, No. 2, pp. 103–113, 2011.