

Received March 24, 2020, accepted April 14, 2020, date of publication April 22, 2020, date of current version May 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989444

DNA Logic Multiplexing Using Toehold-Mediated Strand Displacement

CONGZHOU CHEN¹, HONGYU WANG¹, ENQIANG ZHU², XIAOLONG SHI², AND JIN XU¹

¹School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

²Institute of Computing Science and Technology, Guangzhou University, Guangzhou 510006, China

Corresponding author: Jin Xu (jxu@pku.edu.cn)

This work was supported in part by the National Key R&D Program of China (Grant 2019YFA0706401), and in part by the National Natural Science Foundation of China under Grant 61632002, Grant 61872399, Grant 1872166, Grant 61672264, Grant 61772376, Grant 61872101, and Grant 61702075.

ABSTRACT Since the discovery of the DNA Strand Displacement mechanism, researchers have implemented a lot of applications, such as DNA computing, DNA Circuits, Logic gates, and Chemical Reaction network. To achieve those functions, a well-designed system is essential, among which the toehold domains and migration domains play a vital role. In this paper, we designed three basic logic gates based on the toehold mediated DNA strand displacement mechanism, and utilized them to struct a three-layer multiplexer DNA logic circuit. However, the traditional INHIBIT gate annihilated all the inputs strands which obscure the multiplexer in further applications. Therefore, we improved the INHIBIT gate, so the desired input strand can be selected, and the corresponding output strand can be identified. Lastly, we adjusted the multiplexer and realized a cyclic DNA circuit. The simulation results verified the efficiency and reliability of our multiplexer DNA logic circuits. Our method has the ability in architecting complex DNA integrated circuits.

INDEX TERMS DNA strand displacement, logic circuits, INHIBIT gate, demultiplexer.

I. INTRODUCTION

DNA as a natural material is an excellent tool for many applications. It was used as a novel material for nanostructures, as its length and diameter are in nanometer [1]–[4]. The natural affinity of DNA made it a perfect medium for drug delivery and bio-detection [5]–[7]. Most importantly, the Watson-Crick base-pairing provides the foundation for bio-computing [8]–[9], information storage [10]–[11], synthetic biology [12], etc. One crucial technology for DNA based computing is the DNA combinatorial strand displacement (DSD) [13].

The toehold mediates DNA strand replacement was proposed by Turberfield *et al.* [13]. The DSD principle involves the short single strand domain (toehold) and the replacement of paired double strands. The diagram of DSD mechanism is illustrated in Fig 1. Various DSD based computing systems were verified [14]–[16], the traditional logic gate such as AND, OR, NOR, and INHIBIT gate [17]–[19]. Complex Chemical Reaction Network (CRN) such as polynomial and

logarithms Network [20], Enzymatic reaction Network, and other networks are outstanding works [21]–[24].

In prior works, researchers involved in assembling simple logic gates to complex cascaded circuits or improving the traditional logic gates. To assemble large scale circuits, Qian *et al.* proposed the seesaw gate and verified its abilities, cascade and parallel-connected them into large scale logic circuits [25]. Salehi and coworkers designed four basic formations and computed some advanced mathematic formulas based on the DSD system [20]. While some researchers involved in improving the traditional logic gate to implement novel applications. Song *et al.* designed autocatalytic amplifiers to compute logarithmic and exponential formulas [26]. DelRosso *et al.* designed a DNA molecular Regenerator and used it for regeneration and multiple cycles of computing [27]. Song *et al.* introduced an architecture method to program DSD reaction networks on cell membranes [28]. Additionally, they presented another architecture for fast and compact DNA circuits on Bst2.0 DNA polymerase [29].

In the DNA-based circuits, most gate motifs are use-once only, which are strands consuming. There are some works about constructing renewable circuits [30]–[33], [37]. Song *et al.* used azobenzene isomerization for designing

The associate editor coordinating the review of this manuscript and approving it for publication was Shubhajit Roy Chowdhury¹.

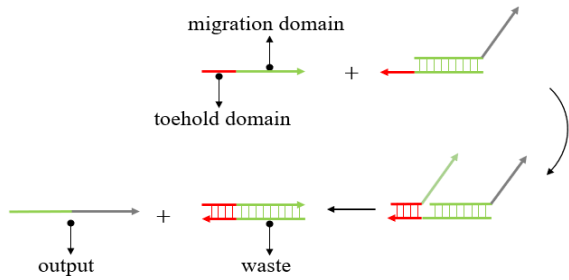


FIGURE 1. The mechanism of the toehold mediates DNA strand displacement.

the renewable DNA seesaw logic circuits [30]. Eshra and El-Sayed utilized the DNA-zyme machine as an odd parity checker, and the enzyme-based machine is reusable [31]. Additionally, Eshra and her coworkers built renewable DNA circuits by reversing the hairpin-gate motif [32]. Garg *et al.* used the dual-rail logic method to realize renewable DNA circuits [33].

Notably, the introduction of the INHIBIT gate that competes with the input strands helps giving an instant stop, avoided the priming of output and secondary productions. Researchers have designed many INHIBIT gates [34]–[37]. Niu *et al.* designed a triquetrous INHIBIT gate and implemented a 4-to-2 encoder [34]. Zhang *et al.* used the hairpin shape structure as the INHIBIT gate and architected a recycle DNA circuit [37]. However, the traditional INHIBIT gate completely quenched the input signals. Therefore, we intend to improve the traditional INHIBIT gate and use it for further applications.

In this paper, we architected a multiplexer DNA logic circuit based on DNA strand displacement. Besides, we improved the INHIBIT gate and used it as a demultiplexer so that different input signals were selected and different output signals can be identified. Lastly, we adjusted the multiplexer and fulfilled a cyclic circuit, which can be extended for more potential applications.

II. RESULTS

A. BASIC LOGIC GATES

The primary logic gate, such as AND, OR, INHIBIT can be integrated into more complex circuits. Here, we designed these three basic gates by DNA strand displacement, and structured a 3-layer 3-to-2 circuit. The structure of those gates is shown in Table. 1.

As shown in the AND gate, the output strand has two domains that connected with the substrate. Only if Input1 and Input2 coexist and react with this substrate together, the output strand can be completely replaced and released the output signal “1”. While in the OR gate, the substrate is a complex, consist of two segments with toehold t_1^* and t_2^* . Usually, the OR gate has two input strands and one output strand. But in this work, the existence of the output strand or not represents the result of “1”/”0”. If one of the input strands Input1/Input2 exists, it will react with the substrate

TABLE 1. The AND, OR, INHIBIT three basic logic gates.

	Inputs	Substrate	Output
AND	t_1 a b t_2	a t_4 b t_1^* a^* b^* t_2^*	a t_4 b
OR	t_1 a b t_2	a t_4 b t_1^* a^* b^* t_2^*	a b
INHIBIT	t_1 a b t_2	t_2^* b^* a^* t_1^*	

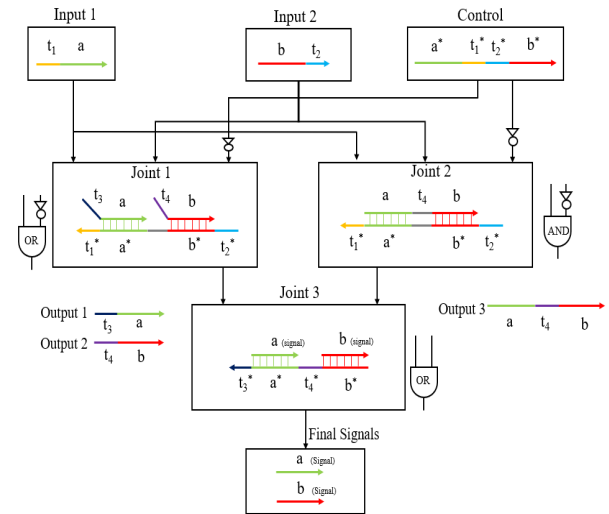


FIGURE 2. Structure of the multiplexer logic circuit.

then release the output signal “1”. In the INHIBIT gate, the control strand (INHIBIT Substrate), complement Input1 and Input2. If there still exist the INHIBIT strands, the input strands will be quenched. The control strand servers as an INHIBIT gate, no matter what the input is, it gives the output signal “0”. Further, the control strand can be used as a threshold, and we can set threshold gate for the input and output signal.

B. LOGIC CASCADED CIRCUITS

We used those three basic logic gates, constructed a three-layer logic circuit. The initial Inputs and Joints are in Table. 2, and the structure of this circuit is shown in Fig. 2.

As shown in Table. 2, the initial input strands are Input1 with toehold t_1 and domain a, Input2 with toehold t_2 and domain b, separately. The initial joint substrates are Joint1, Joint2, and Joint3. When Input1 and Input2 transmit to Joint1, they will react with joint substrate (Joint1), and produce the output strands: Output1 and Output2. The two output strands will transfer to the next joint substrate (Joint3), and react with Joint3, product the final output signals: Signal1 and Signal2. Signal 1&2 are tethered with the same signal (labeled as a-tether(n), b-tether(n)), either exist of Input1 or Input2, this reaction pathway produces the Final output signal “1”. That is the first pathway.

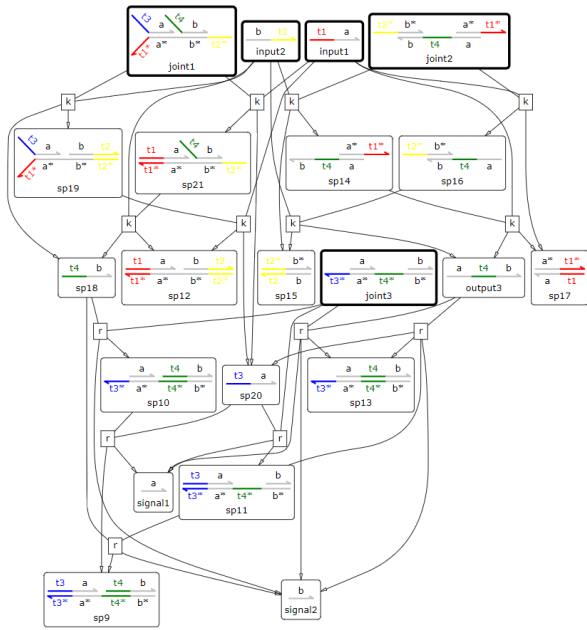


FIGURE 3. The reaction network of the proposed logic circuit, automatically formed by the DSD simulation software. The “sp” is the abbreviation of “species”, and those species are the in-process produces.

TABLE 2. The initial Inputs and substrate (joints).

Joint1		Input1	
Joint2		Input2	
Joint3		Inhibit	

In the second reaction pathway, Input1 and Input2 transmit to the joint substrate (Joint2). They will work together to replace the original domain and release the output strand, Output3 (the AND gate reaction). Then Output3 transmits to the joint substrate (Joint3), and produce the final signals: Signal1 and Signal2. If either loss of the Input strands, this pathway will be obscured, with no final signal, giving logic signal “0”.

The two pathways are integrated into the multiplexer. The truth table is shown in table3, and the reaction network is illustrated in Fig. 3.

Simulation results (Fig. 4) and the truth table (Table. 3) show the output signal according to different input situations. We set the concentrations of Input1 and Input2 are all 70nM, the Control strand is 75nM. If Input1 and Input2 both exist in the system, then the Control strand will be 150nM. The Control strand is just a little more than input strands. Hence, while existing INHIBIT gate, the Final output will be “0”.

We also found that the trajectory of Final output Signal1&2 has a difference. The output signal trajectory in “(1,0,0;1)” and “(0,1,0;1)” are the same. While in

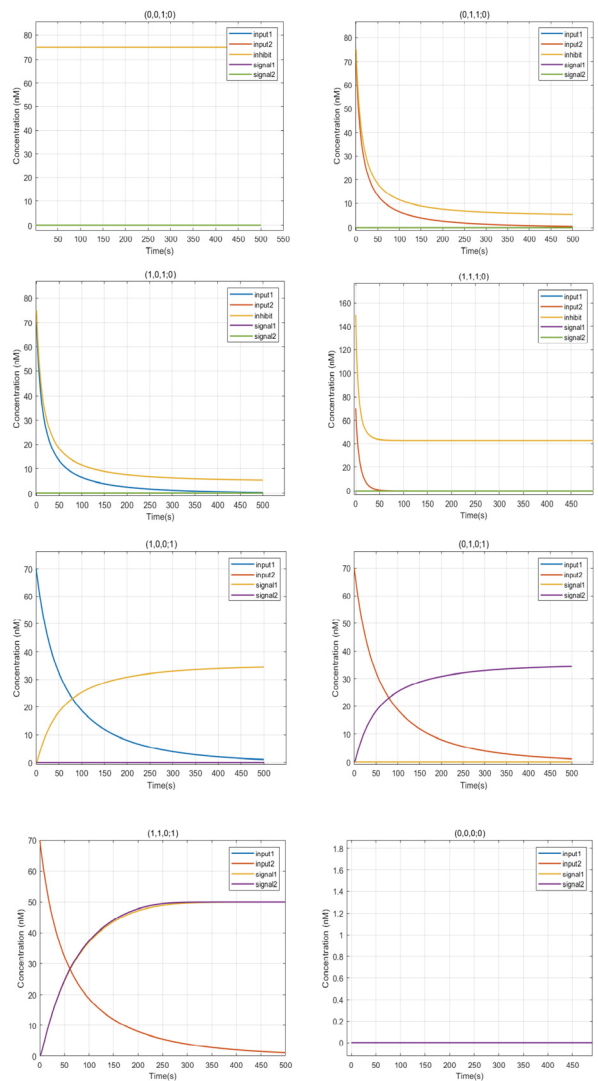


FIGURE 4. The DSD simulation results of the logic circuits. The numbers in the bracket are the truth logic according to each situation. (Input1, Input2, control; Final output).

TABLE 3. The truth table of the logic circuit.

Control	Input1	Input2	Final output
1	0	0	0
1	1	0	0
1	0	1	0
1	1	1	0
0	1	0	1
0	0	1	1
0	1	1	1
0	0	0	0

“(1,1,0;1)”, trajectories of those two Signals have a small gap. We think it is because the AND gate reaction rate is slower than the OR gate, as Input1 and Input2 must work

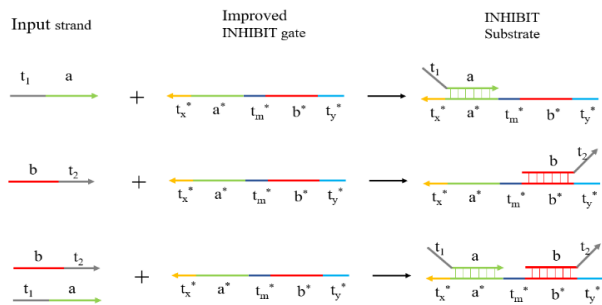


FIGURE 5. The improved INHIBIT gate and the inhibit reactions.

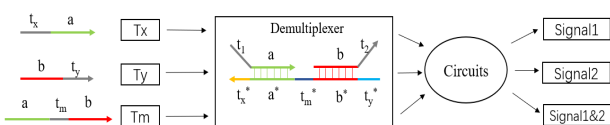


FIGURE 6. The inhibit substrate functions as Demultiplexer, Tx/Ty/Tm can replace different Inputs and produce corresponding Final output signal.

together to replace the original strand. However, the final concentrations of the two Signals 1&2 are the same. Additionally, in real experiments, the signal in the final output strand can be fluorescent dyes, nanoparticles, etc.

C. DEMULTIPLXER

In this logic circuit, the INHIBIT gate completely quenched the input signals. Those input strands are wasted and cannot be reused. Hence, we improved the INHIBIT gate that input strands can be integrated and co-worked in further reactions. We redesigned the INHIBIT gate, both input strands can be inhibited but have the change for further branch immigration. The improved INHIBIT gate is shown in Fig. 5.

After reacting, Input1 and Input2 DNA strands are integrated into this INHIBIT substrate. The improved INHIBIT gate can restrain the input signals, but toehold domain t_1 and t_2 aren't annihilated. That is, the unpaired toehold t_x , t_y , and t_m are the remained domains where input strands can be replaced. In this improved inhibit gate, Input1 can be immigrated by Tx strand (toehold t_x with domain a, from 5' to 3' direction); Input2 can be immigrated by Ty strand (toehold t_y with domain b, from 5' to 3' direction); Input 1&2 can be immigrated by Tm strand from this substrate, simultaneously. The reactions are shown in Fig. 6, reaction network of Demultiplexer is shown in supplementary Fig. S1.

With the extra strands Tx, Ty, and Tm, the improved INHIBIT gate can be functioned as a demultiplexer. Tx strand selects the final Signal1, Ty selects the final Signal2, and Tm produces the final Signal1&2, gives output signal "1". In this way, the different Final output signals are identified. The simulation results are in Fig. 7.

Fig. 7 shows the simulation results of adding Tx, Ty, Tm, separately. If no extra strand is added into this system, there won't be a Final output signal. The improved INHIBIT gate will consume all the input strands (we set the Inhibit strands 75nM, which is more than the input strands). If we added the

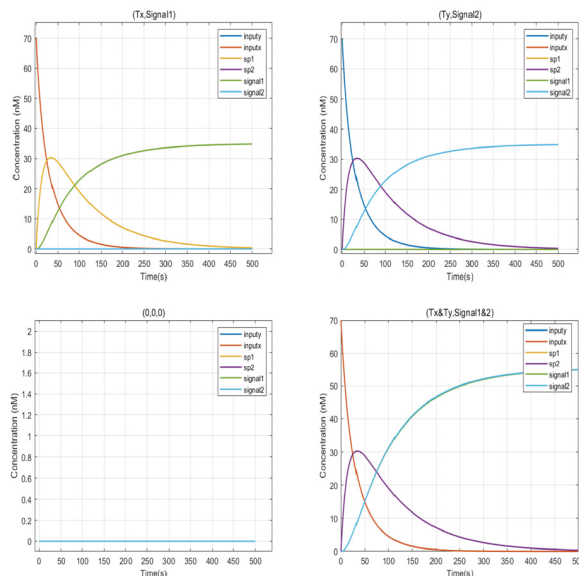


FIGURE 7. The simulation results, the improved INHIBIT gate serves as a demultiplexer.

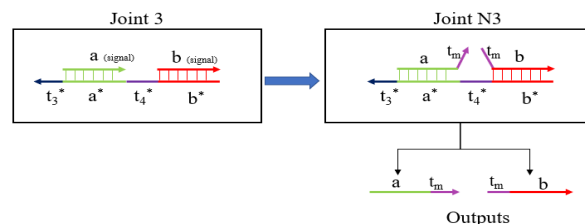


FIGURE 8. The modified Joint3, new Joint 3. Extra toehold domain tm is attached. Then the outputs can cyclicly participate in the cyclic reaction.

Tx strands (70nM), then the Input1 strand will be produced from the inhibit substrate, and Signal1 will be the output. The same situation, we added Ty, Signal2 is the Final output.

In this method, Input1 and Input2 are selected, and different Output Signal1 and Signal2 are identified. Additionally, if we add the Tm strands, it will produce the Input1 and Input2, simultaneously.

In the demultiplex simulation results, Input1 and Input2 become the in-process produces, hence they are labeled as sp1 and sp2 by the DSD software, separately. The yellow curve (Input1) and purple curve (Input2) rise rapidly when adding the extra strands Tx (or Ty), then they are consumed by the downstream reactions (multiplexer circuits). Therefore, the trajectory is risen at the beginning and then slowly falls to zero.

D. CYCLIC MOLECULAR CIRCUIT

In this multiplexer circuits, the concentrations of Input1 and Input2 are 70nM, and the concentration of the Improved INHIBIT substrate is 150nM. The INHIBIT gate just annihilates all the inputs strands. And we set the concentrations of Tx, Ty, Tm are all 70nM. Hence, the concentration of Final output stands is decided by Tx, Ty, Tm. (Downstream Joints

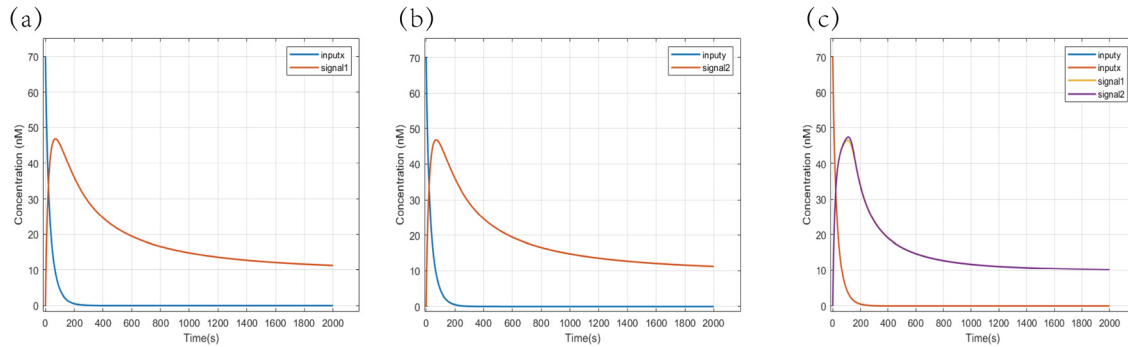


FIGURE 9. The simulation results of the cyclic reaction. (a) Tx strand produces the Signal1, the Signal1 cycle. (b) Ty strand produces Signal2, the Signal2 cycle. (c) Tm strand produces the Signal1&2, the Signal1&2 cycle.

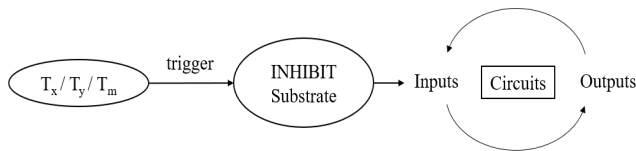


FIGURE 10. The whole reaction flow chart. Tx/Ty/Tm trigger the substrate, then outputs in Fig. 8 will be recycled.

are all oversupplied) We wondered whether this system could be adjusted when adding the extra input strand (Tx, Ty, Tm). The system can continuously produce Input1 and Input2 so that the circuits cyclically reacted.

To fulfill this assumption, we adjusted the Final output signal strand1&2, added extra toehold domain t_m on them, as shown in Fig. 8. Therefore, the Output Signal 1&2 can participate in the cyclic reactions to produce the input strands. We added 70nM of Tx (Ty/Tm) in this cyclic reaction system. Fig. 9 is the simulation results, and the Reaction cycle is illustrated in Fig. 10.

Comparing the simulation results in Fig. 9 and Fig.7, several differences illustrate the cyclic reaction. First, the final concentration of Signal 1&2 is 10nM, while in Fig.7, the final concentration of them is 35nM. Therefore, the signal degradation is about 28.6% (10/35). Second, the reaction time is different. In Fig. 7, Signal1&2 were produced in 300s. In Fig. 10, the Signal 1&2 took 1500s to fall to 10nM. Therefore, three cyclic reactions occurred in the system. Lastly, the concentrations of Joints are all constant variables in this simulation (we set constant order for Joints when simulation), therefore the cyclic reaction won't stop because of exhausting of the Joints. Judging by those facts, we think the cyclic reaction circuit works.

III. DISCUSSION

The mechanism of DSD is the toehold mediates strands immigration. The toehold domain is short sequences that can fast bound to its base-pair domain and displace the other part by its migration domain. To make sure all the reactions are toehold-mediated. We only design the necessary domains in the whole system. This is the reason why all the Inputs, Outputs, and Joints are so short. There are only two migration

domains in this system (a&b); the key exchanges are in the toehold domains. As we added extra toehold domain in the final output strands, and formed a cyclic reaction network. Researchers can add extra migration domains on our system to fulfill some functions, such as feedback circuits, mathematical formulas, DNA-based computing, etc.

In this paper, we architected a 3-layer logic cascaded circuit based on the DNA combinatorial strand displacement, and verified its logic truth through DSD simulation. Besides, we improved the INHIBIT gate so that it can be used as a demultiplexer. In many multilayer DNA circuits [20], [34], [35], the outputs are mixed, and the individual strand can't be selected. The proposed demultiplexer has significant usages in signal selection. Lastly, we modified the toehold domain on the output strands, and formed a cyclic reaction network. The cyclic system is different from the above-mentioned works [30]–[32]. We do not add extra strands to pull back motifs [32] or use extra mechanisms for reverse [30], [31], the key point in our cyclic circuits is the cyclic replacement of the toehold domains. (Adding additional strand to restore the circuits is also viable in the proposed system as shown in supplementary Fig. S2) The system demonstrated that the proposed circuit has powerful potentials for large scale computing and other functions.

In the future, we intend to use this logic circuit to form more complex structures and fulfill more functions. The system can be ejected into the living cell, and implement the in-cell computing. We can also apply our circuits onto surface substrates and complete the surface aid computing, such as on membrane, on DNA origami, etc. The proposed circuits can be adapted and utilized on DNA computing, nanocircuits, and other fields.

REFERENCES

- [1] P. W. K. Rothemund, "Folding DNA to create nanoscale shapes and patterns," *Nature*, vol. 440, no. 7082, pp. 297–302, Mar. 2006.
- [2] X. Shi, W. Lu, Z. Wang, L. Pan, G. Cui, J. Xu, and T. H. LaBean, "Programmable DNA tile self-assembly using a hierarchical sub-tile strategy," *Nanotechnology*, vol. 25, no. 7, Jan. 2014, Art. no. 075602.
- [3] X. Shi, C. Chen, X. Li, T. Song, Z. Chen, Z. Zhang, and Y. Wang, "Size-controllable DNA nanoribbons assembled from three types of reusable brick single-strand DNA tiles," *Soft Matter*, vol. 11, no. 43, pp. 8484–8492, Nov. 2015.

- [4] C. Chen, J. Xu, and X. Shi, "Adjusting linking strands to form size-controllable DNA origami rings," *IEEE Trans. Nanobiosci.*, vol. 19, no. 2, pp. 167–172, Apr. 2020.
- [5] Q. Zhang, Q. Jiang, N. Li, L. Dai, Q. Liu, L. Song, J. Wang, Y. Li, J. Tian, B. Ding, and Y. Du, "DNA origami as an *in vivo* drug delivery vehicle for cancer therapy," *ACS Nano*, vol. 8, no. 7, pp. 6633–6643, Jun. 2014.
- [6] R. Chhabra, J. Sharma, Y. Liu, S. Rinker, and H. Yan, "DNA self-assembly for nanomedicine," *Adv. drug Del. Rev.*, vol. 62, no. 6, pp. 617–625, 2010.
- [7] X. Li, T. Song, Z. Chen, X. Shi, C. Chen, and Z. Zhang, "A universal fast colorimetric method for DNA signal detection with DNA strand displacement and gold nanoparticles," *J. Nanomaterials*, vol. 2015, pp. 1–9, 2015.
- [8] J. Xu, "Probe machine," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1405–1416, Jul. 2016.
- [9] P. W. K. Rothmund, N. Papadakis, and E. Winfree, "Algorithmic self-assembly of DNA sierpinski triangles," *PLoS Biol.*, vol. 2, no. 12, p. e424, Dec. 2004.
- [10] Y. Erlich and D. Zielinski, "DNA fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950–954, Mar. 2017.
- [11] M. Blawat, K. Gaedke, I. Hütter, X.-M. Chen, B. Turczyk, S. Inverso, B. W. Pruitt, and G. M. Church, "Forward error correction for DNA data storage," *Procedia Comput. Sci.*, vol. 80, pp. 1011–1022, Jan. 2016.
- [12] A. Hernandez-García, N. A. Estrich, M. W. T. Werten, J. R. C. Van Der Maarel, T. H. LaBean, F. A. de Wolf, M. A. C. Stuart, and R. de Vries, "Precise coating of a wide range of DNA templates by a protein polymer with a DNA binding domain," *ACS Nano*, vol. 11, no. 1, pp. 144–152, Dec. 2016.
- [13] A. J. Turberfield, J. C. Mitchell, B. Yurke, A. P. Mills, M. I. Blakey, and F. C. Simmel, "DNA fuel for free-running nanomachines," *Phys. Rev. Lett.*, vol. 90, no. 11, Mar. 2003, Art. no. 118102.
- [14] G. Chatterjee, N. Dalchau, R. A. Muscat, A. Phillips, and G. Seelig, "A spatially localized architecture for fast and modular DNA computing," *Nature Nanotechnol.*, vol. 12, no. 9, pp. 920–927, Jul. 2017.
- [15] Z. Yin, J. Cui, and C. Zhen, "Molecular beacon computing model for maximum weight clique problem," *Math. Comput. Simul.*, vol. 151, pp. 147–155, Sep. 2018.
- [16] J. Yang, Z. Yin, Z. Tang, K. Huang, J. Cui, and X. Yang, "Search computing model for the knapsack problem based on DNA origami," *Mater. Express*, vol. 9, no. 6, pp. 553–562, Sep. 2019.
- [17] Z. Tang, Z.-X. Yin, X. Sun, J.-Z. Cui, J. Yang, and R.-S. Wang, "Dynamically NAND gate system on DNA origami template," *Comput. Biol. Med.*, vol. 109, pp. 112–120, Jun. 2019.
- [18] S. M. Douglas, I. Bachelet, and G. M. Church, "A logic-gated nanorobot for targeted transport of molecular payloads," *Science*, vol. 335, no. 6070, pp. 831–834, Feb. 2012.
- [19] C. Zhang, J. Yang, S. Jiang, Y. Liu, and H. Yan, "DNAzyme-based logic gate-mediated DNA self-assembly," *Nano Lett.*, vol. 16, no. 1, pp. 736–741, Dec. 2015.
- [20] S. A. Salehi, X. Liu, M. D. Riedel, and K. K. Parhi, "Computing mathematical functions using DNA via fractional coding," *Sci. Rep.*, vol. 8, no. 1, p. 8312, May 2018.
- [21] J. Yang, R. Wu, Y. Li, Z. Wang, L. Pan, Q. Zhang, Z. Lu, and C. Zhang, "Entropy-driven DNA logic circuits regulated by DNAzyme," *Nucleic Acids Res.*, vol. 46, no. 16, pp. 8532–8541, Jul. 2018.
- [22] A. A. Green, J. Kim, D. Ma, P. A. Silver, J. J. Collins, and P. Yin, "Complex cellular logic computation using ribocomputing devices," *Nature*, vol. 548, no. 7665, pp. 117–121, Jul. 2017.
- [23] A. J. Genot, J. Bath, and A. J. Turberfield, "Combinatorial displacement of DNA strands: Application to matrix multiplication and weighted sums," *Angew. Chem. Int. Ed.*, vol. 52, no. 4, pp. 1189–1192, Dec. 2012.
- [24] C. Liu, E. Zhu, Q. Zhang, and X. Wei, "Modeling of agent cognition in extensive games via artificial neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4857–4868, Oct. 2018.
- [25] 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.
- [26] T. Song, S. Garg, R. Mokhtar, H. Bui, and J. Reif, "Design and analysis of compact DNA strand displacement circuits for analog computation using autocatalytic amplifiers," *ACS Synth. Biol.*, vol. 7, no. 1, pp. 46–53, Dec. 2017.
- [27] N. V. DeRossa, S. Hews, L. Spector, and N. D. Derr, "A molecular circuit regenerator to implement iterative strand displacement operations," *Angew. Chem. Int. Ed.*, vol. 56, no. 16, pp. 4443–4446, Mar. 2017.
- [28] T. Song, S. Shah, H. Bui, S. Garg, A. Eshra, D. Fu, M. Yang, R. Mokhtar, and J. Reif, "Programming DNA-based biomolecular reaction networks on cancer cell membranes," *J. Amer. Chem. Soc.*, vol. 141, no. 42, pp. 16539–16543, Oct. 2019.
- [29] T. Song, A. Eshra, S. Shah, H. Bui, D. Fu, M. Yang, R. Mokhtar, and J. Reif, "Fast and compact DNA logic circuits based on single-stranded gates using strand-displacing polymerase," *Nature Nanotechnol.*, vol. 14, no. 11, pp. 1075–1081, Sep. 2019.
- [30] X. Song, A. Eshra, C. Dwyer, and J. Reif, "Renewable DNA seesaw logic circuits enabled by photoregulation of toehold-mediated strand displacement," *RSC Adv.*, vol. 7, no. 45, pp. 28130–28144, 2017.
- [31] A. Eshra and A. El-Sayed, "An odd parity checker prototype using DNAzyme finite state machine," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 2, pp. 316–324, Mar. 2014.
- [32] A. Eshra, S. Shah, T. Song, and J. Reif, "Renewable DNA hairpin-based logic circuits," *IEEE Trans. Nanotechnol.*, vol. 18, pp. 252–259, 2019.
- [33] S. Garg, S. Shah, H. Bui, T. Song, R. Mokhtar, and J. Reif, "Renewable time-responsive DNA circuits," *Small*, vol. 14, no. 33, Jul. 2018, Art. no. 1801470.
- [34] Y. Niu, F. Han, X. Zhang, and Z. Zhou, "A circuit simplification mechanism based on DNA combinatorial strands displacement," *Fundamenta Informaticae*, vol. 164, nos. 2–3, pp. 243–257, Jan. 2019.
- [35] X. Zhang, W. Zhang, T. Zhao, Y. Wang, and G. Cui, "Design of logic circuits based on combinatorial displacement of DNA strands," *J. Comput. Theor. Nanosci.*, vol. 12, no. 7, pp. 1161–1164, Jul. 2015.
- [36] E. Zhu, C. Chen, Y. Rao, and W. Xiong, "Biochemical logic circuits based on DNA combinatorial displacement," *IEEE Access*, vol. 8, pp. 34096–34103, 2020, doi: 10.1109/ACCESS.2020.2974024.
- [37] C. Zhang, Z. Wang, Y. Liu, J. Yang, X. Zhang, Y. Li, L. Pan, Y. Ke, and H. Yan, "Nicking-assisted reactant recycle to implement entropy-driven DNA circuit," *J. Amer. Chem. Soc.*, vol. 141, no. 43, pp. 17189–17197, Sep. 2019.



CONGZHOU CHEN received the B.S. degree in automation from the Wuhan University of Science and Technology, Wuhan, China, in 2014, and the M.S. degree in automation from the Huazhong University of Science and Technology, Wuhan, in 2016. He is currently pursuing the Ph.D. degree with Peking University, Beijing, China. His interests include DNA nanotechnology and DNA biocomputing.



HONGYU WANG received the B.S. degree in mathematics and applied mathematics from Heilongjiang University, the M.S. degree in operational research and cybernetics from Northwest Normal University, and the Ph.D. degree in computer science from Peking University, Beijing, China.

She currently holds a postdoctoral position with the School of Electronics Engineering and Computer Science, Peking University. Her main research interests include DNA Computing, graph and combinatorial optimization, and network and information security.



ENQIANG ZHU received the B.S. and M.S. degrees in computer science from Lanzhou Jiaotong University and the Ph.D. degree in computer science from Peking University, Beijing, China. He is currently a Professor with the Institute of Computing Science and Technology, Guangzhou University. His main research interests include DNA computing, combinatorics, and computer networks.



XIAOLONG SHI received the Ph.D. degree in systems engineering from the Huazhong University of Science and Technology. He was a Tenured Professor with the Huazhong University of Science and Technology, and he is currently the Dean/Professor of the Institute of Computer Science and Technology, Guangzhou University.



JIN XU received the first Ph.D. degree from Xi'an Jiaotong University in 1993, and received the second Ph.D. degree from the Beijing Institute of Technology in 1994. He is currently a Tenured Professor at Peking University.

He has published more than 200 SCI papers as the first or corresponding author. He has led 10 national projects, including NSFC project, “973” Project, “863” Project, etc. His main research areas include graph theory, biological computing, and information security.

• • •