Massively parallel DNA computing based on domino DNA strand displacement logic gates

(Supplementary Information)

Xin Chen[1], Xinyu Liu[1], Fang Wang[1], Sirui Li[1], Congzhou Chen [2,], Xiaoli Qiang[1,*] and Xiaolong Shi[1,*]*

[1] Institute of Computing Science and Technology, Guangzhou University, Guangzhou 510006, China

[2] Key Laboratory of High Confidence Software Technologies, School of Computer Science, Peking University, Beijing 100871, China

KEYWORDS: DNA computing, DNA strand displacement, Tic-tac-toe, Domino multi-input AND gate

Table S1 DNA sequence design

		sequence
Figure1a	1-logic	GTGGGTAGGAGTGGTTAGGGAGTATTAGGAGTTTG
		TCCCTAACCACTCCTACCCACTAAAC
	2-logic	GAGTATTAGGAGTTTGACGA
		GTGGTAAGGTATAAGGTTAGAAGTTTGAATTGGTG
		TCTAACCTTATACCTTACCACGTCGTCAAACTCCTAATACTCCCTAA
	3-logic	GAAGTTTGAATTGGTGGTGT
		CTCCATTATCAAGCTGAGATGAGGAAGATAGTGGA
		CATCTCAGCTTGATAATGGAGAACACCACCAATTCAAACTTCTAACC
	reporter	TGAGGAAGATAGTGGA-FAM-6`3
		5`BHQ1-TCCACTATCTTCCTCATCTCA
	input a	GTTTAGTGGGTAGGAGTGGTTAGGGA
	input b	ACGACGTGGTAAGGTATAAGGTTAGA
	input c	GTGTTCTCCATTATCAAGCTGAGATG
Figure 1b	input a	ATGTACGAGCCTATTAAAT
	input b	GAGAGGGAAGAGGGTAGAG
	input c	TTGCCGTACCTATTAATTC
	Logic gate	GTACCTATTAATTCGAGAG
		GGAAGAGGGTAGAGATGTA
		CGAGCCTATTAAATGTCCTTGTCACGTC
		ATTTAATAGGCTCGTACATCTCTACCCTCTTCCCTCTCGAATTAATAGGTACGGCAA
	reporter	GTCCTTGTCACGTC-FAM-6`3
		5`BHQ1-GACGTGACAAGGACATTTAATAGGCTCG
Figure 2a	input a	ATGTACGAGCCTATTAAAT
	input b	GAGAGGGAAGAGGGTAGAG

	input c	AATAAAGCACTACAGAACC
	Logic gate	AGCACTACAGAACCGAGAG
		GGAAGAGGGTAGAGATGTA
		CGAGCCTATTAAATGTCCTTGTCACGTC
		ATTTAATAGGCTCGTACATCTCTACCCTCTTCCCTCTCGGTTCTGTAGTGCTTTATT
	reporter	GTCCTTGTCACGTC-FAM-6`3
		5`BHQ1-GACGTGACAAGGACATTTAATAGGCTCG
Figure 2b	input a	ATGTACGAGCCTATTAAAT
	input b	TTGCCGTACCTATTAATTC
	input c	GTGTTCTCCATTATCAAGC
	Logic gate	GTACCTATTAATTC GTGTT
		CTCCATTATCAAGC ATGTA
		CGAGCCTATTAAAT-3'BHQ1
		5'6-FAM-ATTTAATAGGCTCGTACATGCTTGATAATGGAGAACACGAATTAATAGGTACGGCAA

Table S2 Sequence design used in tic-tac-toe games

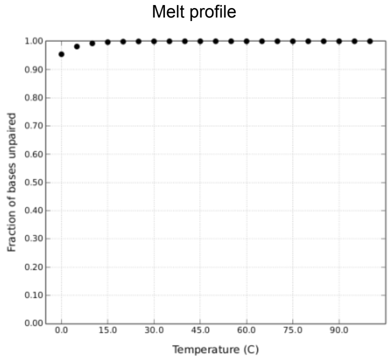
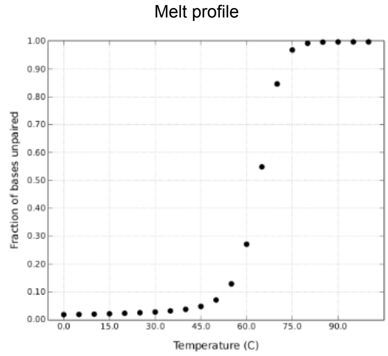
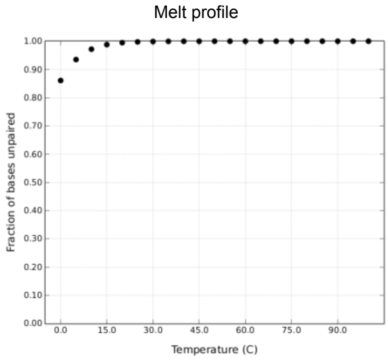
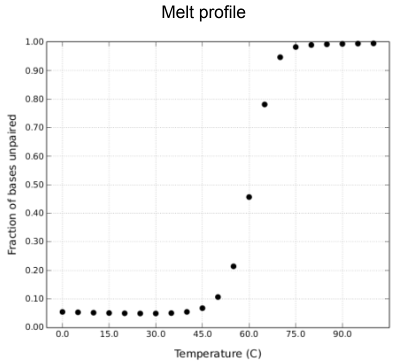
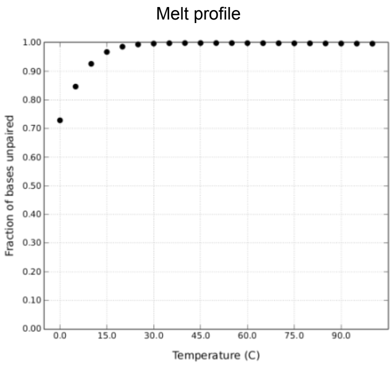
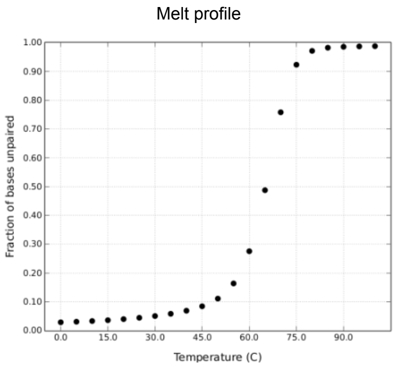
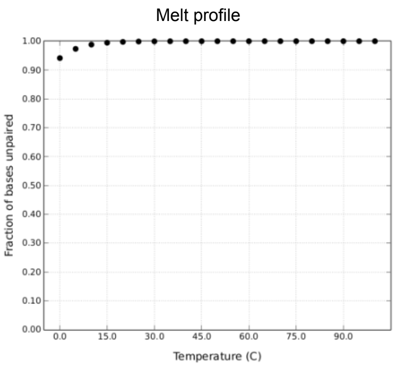
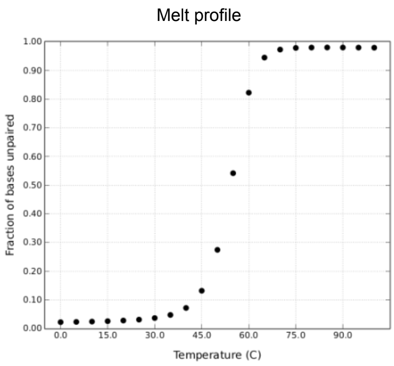
position	Logic gate	sequence
1	gate24	CGAGCCTATTAAAT-3'BHQ1
		5'6-FAM-ATTTAATAGGCTCGTACAT
112	gate23	GTGGTAAGGTATAA GTGTT
		CTCCATTATCAAGC GTTTA
		GTGGGTAGGAGTGG-3'BHQ1
		5'6-FAM-CCACTCCTACCCACTAAACGCTTGATAATGGAGAACACTTATACCTTACCACGTCGT
	gate18	GTACCTATTAATTCGAGAG
		GGAAGAGGGTAGAGATGTA
		CGAGCCTATTAAAT-3'BHQ1
		5'6-FAM-ATTTAATAGGCTCGTACATCTCTACCCTCTTCCCTCTCGAATTAATAGGTACGGCAA
	gate17	GTACCTATTAATTC GTGTT
		CTCCATTATCAAGC ATGTA
		CGAGCCTATTAAAT-3'BHQ1
		5'6-FAM-ATTTAATAGGCTCGTACATGCTTGATAATGGAGAACACGAATTAATAGGTACGGCAA
3	gate22	GTGGTAAGGTATAAGTTTA
		GTGGGTAGGAGTGG-3'BHQ1
		5'6-FAM-CCACTCCTACCCACTAAACTTATACCTTACCACGTCGT
	gate25	GTACCTATTAATTCATGTA
		CGAGCCTATTAAAT-3'BHQ1
		5'6-FAM-ATTTAATAGGCTCGTACATGAATTAATAGGTACGGCAA

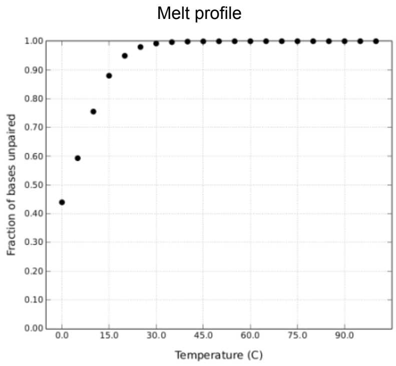
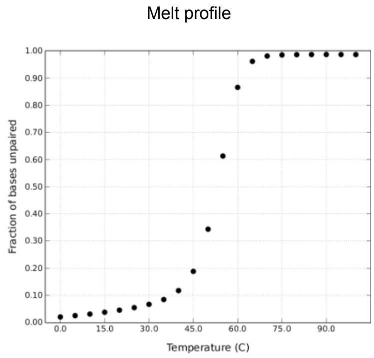
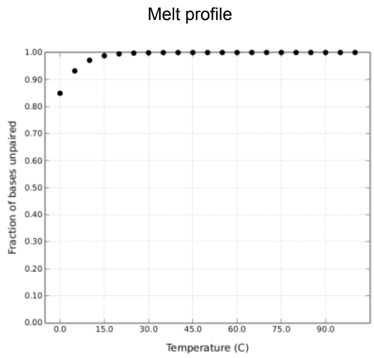
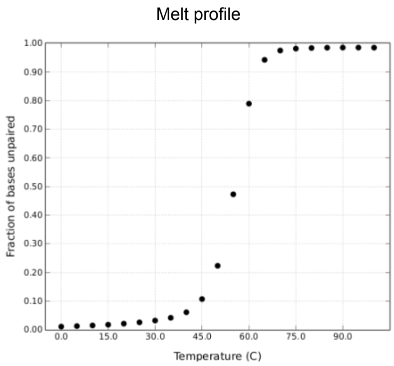
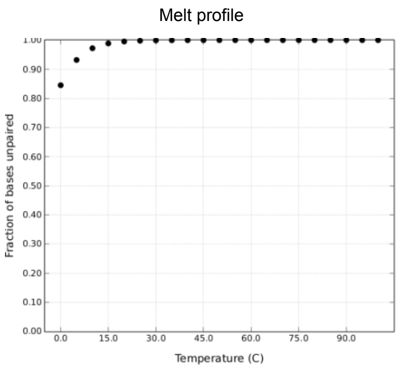
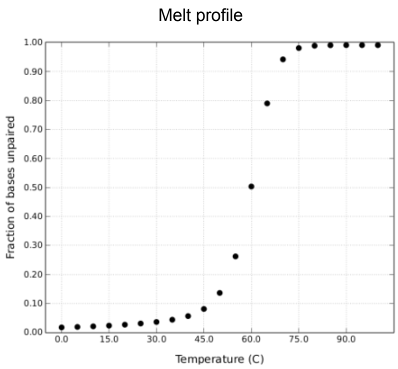
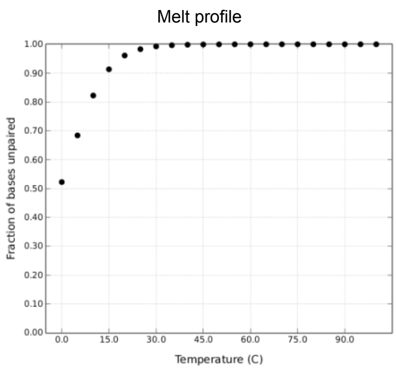
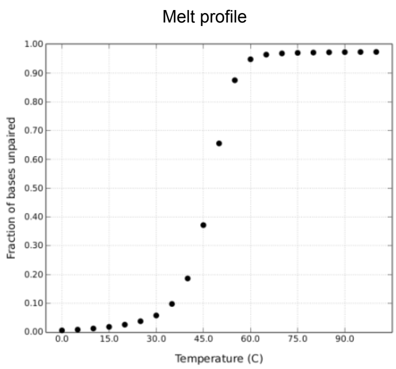
4	gate21	GTGGGTAGGAGTGG-3'BHQ1
		5'6-FAM-CCACTCCTACCCACTAAAC
6	gate1	AGCACTACAGAACCGTTTA
		GTGGGTAGGAGTGG-3'BHQ1
		5'6-FAM-CCACTCCTACCCACTAAACGGTTCTGTAGTGCTTACTG
	gate2	ATTGGACCCGTGAGGTTTA
		GTGGGTAGGAGTGG-3'BHQ1
		5'6-FAM-CCACTCCTACCCACTAAACCTCACGGGTCCAATTCTCA
	gate3	CTCCATTATCAAGCGTTTA
		GTGGGTAGGAGTGG-3'BHQ1
		5'6-FAM-CCACTCCTACCCACTAAACGCTTGATAATGGAGAACAC
	gate4	GGAAGAGGGTAGAGGTTTA
		GTGGGTAGGAGTGG-3'BHQ1
		5'6-FAM-CCACTCCTACCCACTAAACCTCTACCCTCTTCCCTCTC
	gate5	GTACCTATTAATTCGTTTA
		GTGGGTAGGAGTGG-3'BHQ1
		5'6-FAM-CCACTCCTACCCACTAAACGAATTAATAGGTACGGCAA
7	gate6	GTGGTAAGGTATAACAGTA
		AGCACTACAGAACCGTTTA
		GTGGGTAGGAGTGG-3'BHQ1
		5'6-FAM-CCACTCCTACCCACTAAACGGTTCTGTAGTGCTTACTGTTATACCTTACCACGTCGT
	gate7	GTGGTAAGGTATAAGAGAG
		GGAAGAGGGTAGAGGTTTA
		GTGGGTAGGAGTGG-3'BHQ1

		5'6-FAM-CCACTCCTACCCACTAAACCTCTACCCTCTTCCCTCTCTTATACCTTACCACGTCGT
	gate8	GTACCTATTAATTCACGAC
		GTGGTAAGGTATAAGTTTA
		GTGGGTAGGAGTGG-3'BHQ1
		5'6-FAM-CCACTCCTACCCACTAAACTTATACCTTACCACGTCGTGAATTAATAGGTACGGCAA
	gate16	GTACCTATTAATTCACGAC
		GTGGTAAGGTATAAATGTA
		CGAGCCTATTAAAT-3'BHQ1
		5'6-FAM-ATTTAATAGGCTCGTACATTTATACCTTACCACGTCGTGAATTAATAGGTACGGCAA
	gate19	GTACCTATTAATTCCAGTA
		AGCACTACAGAACCATGTA
		CGAGCCTATTAAAT-3'BHQ1
		5'6-FAM-ATTTAATAGGCTCGTACATGGTTCTGTAGTGCTTACTGGAATTAATAGGTACGGCAA
8	gate9	GTACCTATTAATTCACGAC
		GTGGTAAGGTATAAGTGTT
		CTCCATTATCAAGCGTTTA
		GTGGGTAGGAGTGG-3'BHQ1
		5'6-FAM-CCACTCCTACCCACTAAACGCTTGATAATGGAGAACACTTATACCTTACCACGTCGTGAATTAATAGGTACGGCAA
9	gate10	GTGGTAAGGTATAAGAGAG
		GGAAGAGGGTAGAGGTGTT

		CTCCATTATCAAGCGTTTA
		GTGGGTAGGAGTGG-3'BHQ1
		5'6-FAM-CCACTCCTACCCACTAAACGCTTGATAATGGAGAACACCTCTACCCTCTTCCCTCTCTTATACCTTACCACGTCGT
	gate11	AGCACTACAGAACCATGTA
		CGAGCCTATTAAAT-3'BHQ1
		5'6-FAM-ATTTAATAGGCTCGTACATGGTTCTGTAGTGCTTACTG
	gate12	ATTGGACCCGTGAGATGTA
		CGAGCCTATTAAAT-3'BHQ1
		5'6-FAM-ATTTAATAGGCTCGTACATCTCACGGGTCCAATTCTCA
	gate13	GTGGTAAGGTATAAATGTA
		CGAGCCTATTAAAT-3'BHQ1
		5'6-FAM-ATTTAATAGGCTCGTACATTTATACCTTACCACGTCGT
	gate14	CTCCATTATCAAGCATGTA
		CGAGCCTATTAAAT-3'BHQ1
		5'6-FAM-ATTTAATAGGCTCGTACATGCTTGATAATGGAGAACAC
	gate15	GGAAGAGGGTAGAGATGTA
		CGAGCCTATTAAAT-3'BHQ1
		5'6-FAM-ATTTAATAGGCTCGTACATCTCTACCCTCTTCCCTCTC
Input	input 1	GTTTAGTGGGTAGGAGTGG
	input 2	CAGTAAGCACTACAGAACC
	input 3	TGAGAATTGGACCCGTGAG
	input 4	ATGTACGAGCCTATTAAAT
	input 6	ACGACGTGGTAAGGTATAA
	input 7	GTGTTCTCCATTATCAAGC
	input 8	GAGAGGGAAGAGGGTAGAG
	input 9	TTGCCGTACCTATTAATTC

table S3 the Melt profile of the sequence used in the tic-tac-toe game. The Melt profile is measured by NUPACK.

<table>
<tr><td rowspan="2">Position1</td><td colspan="2">
</td><td colspan="2">
</td></tr>
<tr><td>toehold</td><td>GTTTA</td><td>information domains</td><td>GTGGGTAGGAGTGG</td></tr>
<tr><td rowspan="2">Position2</td><td colspan="2">
</td><td colspan="2">
</td></tr>
<tr><td>toehold</td><td>CAGTA</td><td>information domains</td><td>AGCACTACAGAACC</td></tr>
<tr><td rowspan="2">Position3</td><td colspan="2">
</td><td colspan="2">
</td></tr>
<tr><td>toehold</td><td>TGAGA</td><td>information domains</td><td>ATTGGACCCGTGAG</td></tr>
<tr><td>Position4</td><td colspan="2">
</td><td colspan="2">
</td></tr>
</table>

	toehold	ATGTA	information domains	CGAGCCTATTAAAT
Position6				
	toehold	ACGAC	information domains	GTGGTAAGGTATAA
Position7				
	toehold	GTGTT	information domains	CTCCATTATCAAGC
Position8				
	toehold	GAGAG	information domains	GGAAGAGGGTAGAG
Position9				
	toehold	TTGCC	information domains	GTACCTATTAATTC

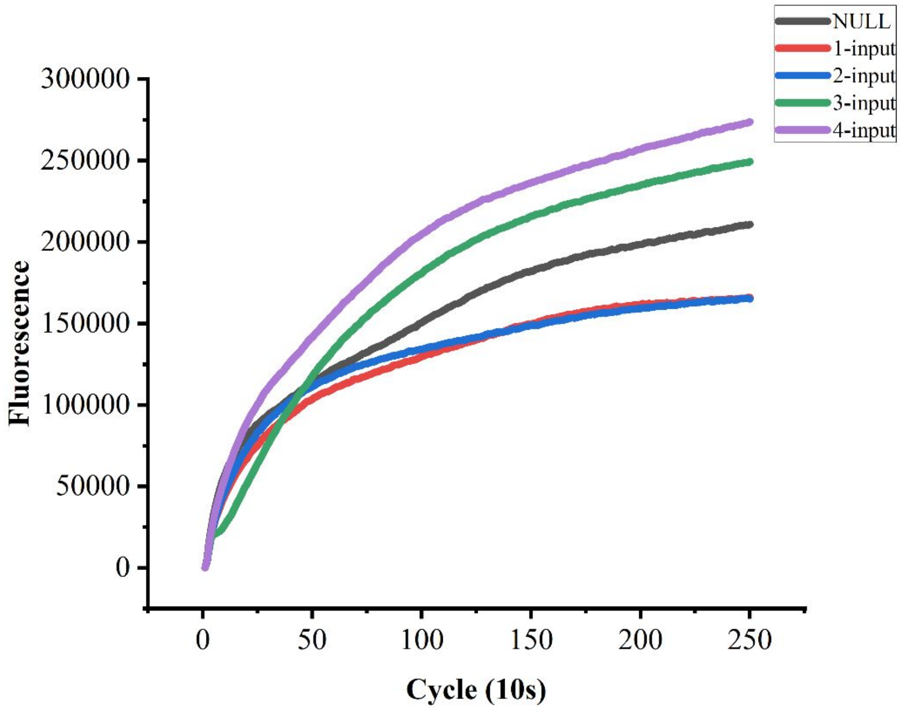


Figure S1 Fluorescence Reporter Results of a Four-Conditional AND-Gate Computational System Implemented in a Multilevel Cascade Approach. A larger number of cascades leads to more significant leakage problems. The total value of the fluorescence response for the input of the three signals was approximately 95% of the value when all four signals were present.

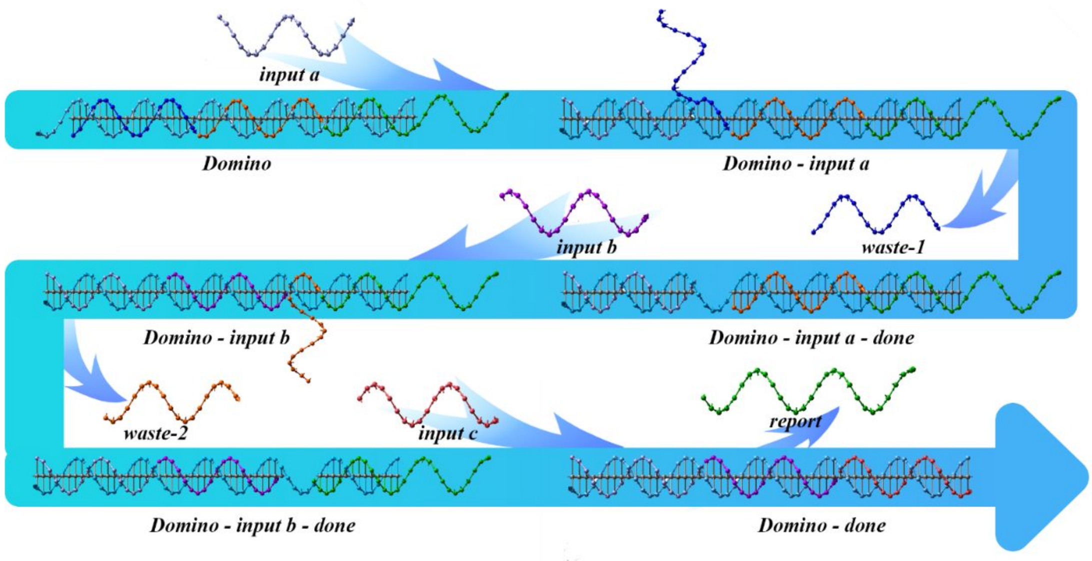


Figure S2. Three-condition domino AND gate replacement process. The three-condition replacement process can be divided into five stages: *"Domino"* is the initial state of the domino logic gate, input *"input a"* into it, and *"input a"* is first combined with the exposed toehold field t_a*, and then replaced out d_a, namely This is the *"Domino-input a"* stage. The t_b falls off due to the molecule's free energy, *"waste-1"* is completely replaced, and the toehold domain t_b is exposed, which is the *"Domino-input a-done"* phase. After input *"input b"*, *"Domino-input b"* is similar to *"Domino-input a"*, and *"Domino-input b-done"* is similar to *"Domino-input a-done"*. *"waste-2"* is completely replaced, and the toehold domain t_c* is exposed. When inputting *"input c", "input c"* is first combined with toehold domain t_c, and then the report is completely replaced. It is the *"Domino-done"* stage, which is the waste in Figure 1b.

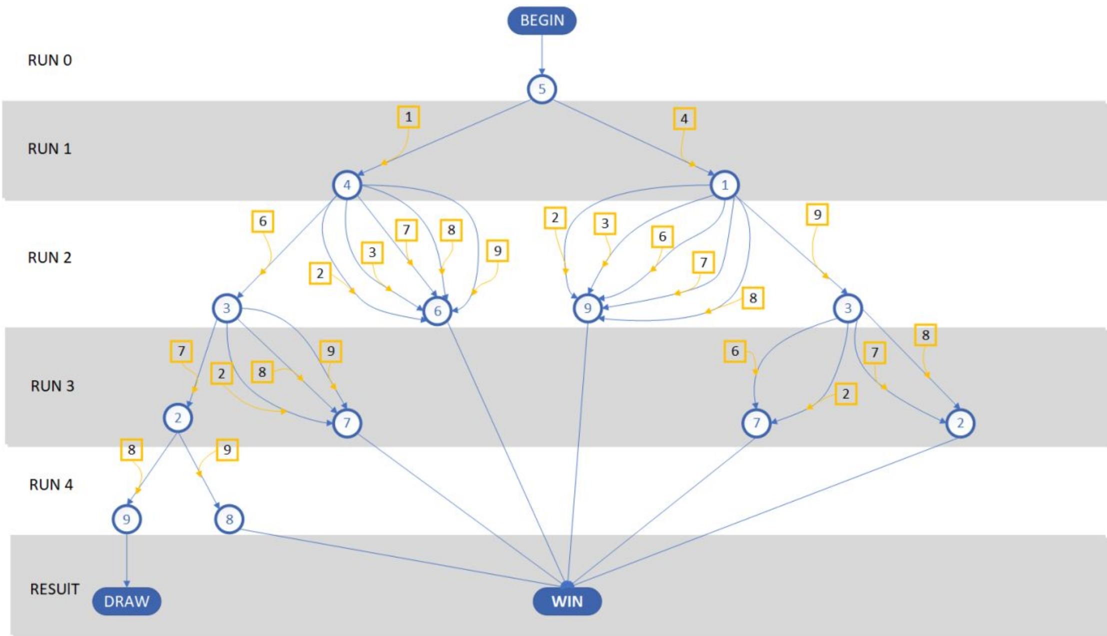


Figure S3 The game tree for tic-tac-toe. The blue circles represent bots, and the yellow boxes represent players. According to the strategy of MAYA-1, the first step of the robot is in the middle, and the default is RUN 0. In each round after that, each step of the robot can be determined by all the points that the player has placed on the path.

$$\underbrace{i_4}_{24} \Rightarrow position\,1$$

$$\underbrace{\left(i_1 \bigwedge i_6 \bigwedge i_7\right)}_{23} || \underbrace{\left(i_4 \bigwedge i_9 \bigwedge i_7\right)}_{17} || \underbrace{\left(i_4 \bigwedge i_9 \bigwedge i_8\right)}_{18} \Rightarrow position\,2$$

$$\underbrace{\left(i_1 \bigwedge i_6\right)}_{22} || \underbrace{\left(i_4 \bigwedge i_9\right)}_{25} \Rightarrow position\,3$$

$$\underbrace{i_1}_{21} \Rightarrow position\,4$$

$$\underbrace{1}_{20} \Rightarrow position\,5$$

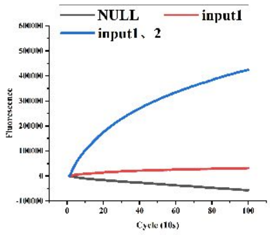
$$\underbrace{\left(i_1 \bigwedge i_2\right)}_{1} || \underbrace{\left(i_1 \bigwedge i_3\right)}_{2} || \underbrace{\left(i_1 \bigwedge i_7\right)}_{3} || \underbrace{\left(i_1 \bigwedge i_8\right)}_{4} || \underbrace{\left(i_1 \bigwedge i_9\right)}_{5} \Rightarrow position\,6$$

$$\underbrace{\left(i_1 \bigwedge i_6 \bigwedge i_2\right)}_{6} || \underbrace{\left(i_1 \bigwedge i_6 \bigwedge i_8\right)}_{7} || \underbrace{\left(i_1 \bigwedge i_6 \bigwedge i_9\right)}_{8} || \underbrace{\left(i_4 \bigwedge i_9 \bigwedge i_6\right)}_{16} || \underbrace{\left(i_4 \bigwedge i_9 \bigwedge i_2\right)}_{19} \Rightarrow position\,7$$

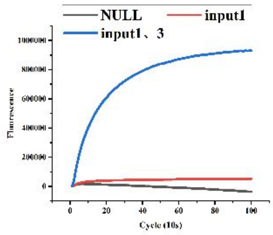
$$\underbrace{i_1 \bigwedge i_6 \bigwedge i_7 \bigwedge i_9}_{9} \Rightarrow position\,8$$

$$\underbrace{\left(i_1 \bigwedge i_6 \bigwedge i_7 \bigwedge i_8\right)}_{10} \underbrace{\left(i_4 \bigwedge i_2\right)}_{11} || \underbrace{\left(i_4 \bigwedge i_3\right)}_{12} || \underbrace{\left(i_4 \bigwedge i_6\right)}_{13} || \underbrace{\left(i_4 \bigwedge i_7\right)}_{14} || \underbrace{\left(i_4 \bigwedge i_8\right)}_{15} \Rightarrow position\,9$$

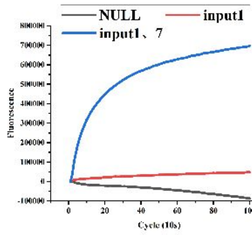
Figure S4 Implementation strategy of the tic-tac-toe game. According to the implementation strategy of the MAYA-I robot, the robot can make judgments based on the player's chess position, and the judgment basis is shown in the figure. The robot's judgment for each position is determined by a disjunctive paradigm. Each parenthesis is a logic AND gate, and the number below is the logic gate number. For example, when the player's chess position is 1, 6, and 7, the robot makes a judgment on the position 2.



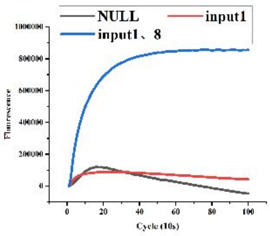
gate 1



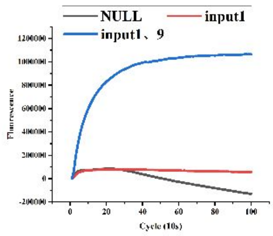
gate 2



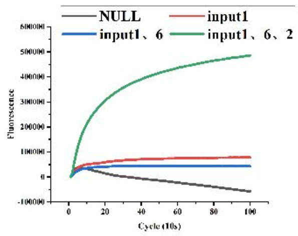
gate 3



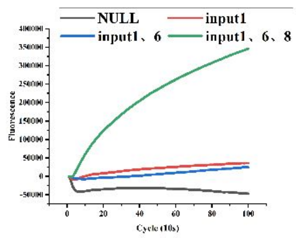
gate 4



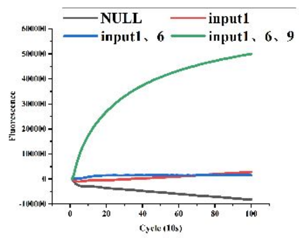
gate 5



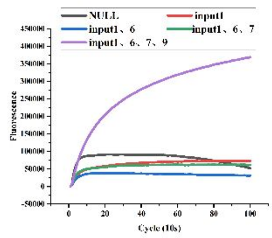
gate 6



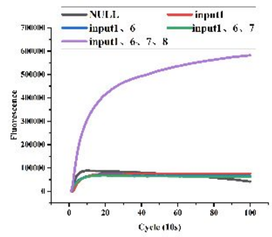
gate 7



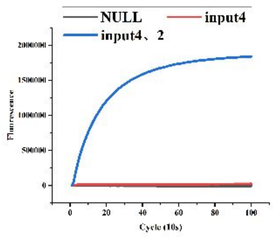
gate 8



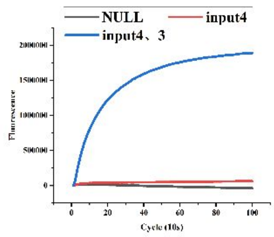
gate 9



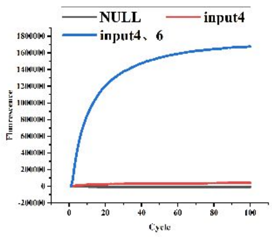
gate 10



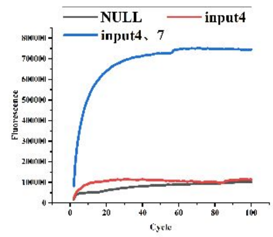
gate 11



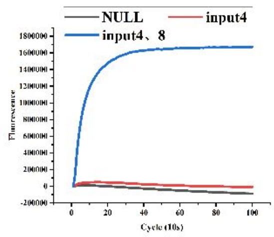
gate 12



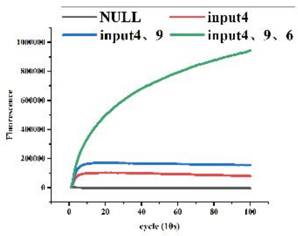
gate 13



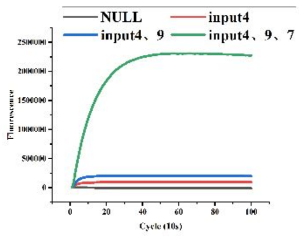
gate 14



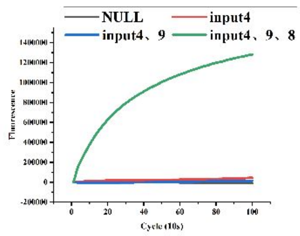
gate 15



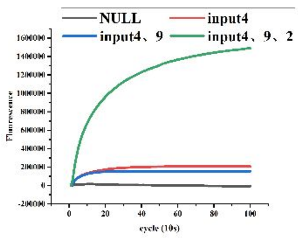
gate 16



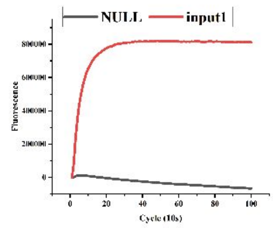
gate 17



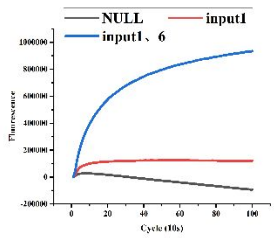
gate 18



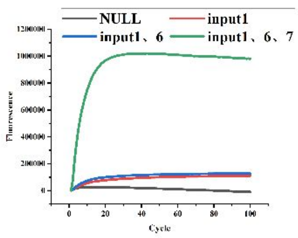
gate 19

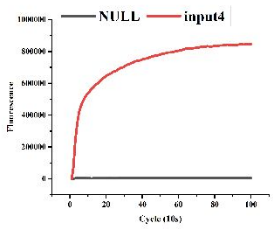


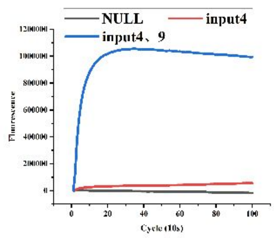
gate 21



gate 22







gate 23 gate 24 gate 25

Figure S5 Fluorescence detection results of Tic-tac-toe game logic gates. We tested the logic gates used in the tic-tac-toe game in separate test tubes. The results show that the logic gates all work properly. And the difference in the fluorescence signal of domino logic gates containing positions 4 and 9 was generally higher than that of domino logic gates containing positions 1 and 6.