SCIENCE CHINA
Information Sciences

• Supplementary File •

Reconfigurable logic circuit design for stateful Boolean logic computing

Li LUO[1,4], Zhekang DONG[2], Xiaofang HU[3], Lidan WANG[1,4] & Shukai DUAN[3,4*]

[1]*College of Electronics and Information Engineering, Southwest University, Chongqing* 400715, *China;*
[2]*School of Electronics and Information, Hangzhou Dianzi University, Hangzhou* 310018, *China;*
[3]*College of Artificial Intelligence, Southwest University, Chongqing* 400715, *China;*
[4]*Brain-inspired Computing & Intelligent Control of Chongqing Key Lab*

Appendix A Logic function reconfiguration of the circuit for NOR and NAND

Figure A1 illustrates the logic function reconfiguration of the circuit for NOR and NAND operations. Here, assume that the input combination is '00', i.e., $p = 0$ and $q = 0$, then P and Q are initialized to R_{H}, and M and Y are set to R_{L} before logic operation, as demonstrated in Figure A1(a). Then the corresponding voltages are applied to perform the NOR logic operation as shown in Figure A1(b). After the NOR logic operation, the circuit needs to realize a new computing task of NAND logic operation. Before the logic operation, P, Q, M, and Y are set to R_{L} for the input combination '11', as demonstrated in Figure A1(c). Later corresponding voltages are applied to trigger the NAND logic operation as shown in Figure A1(d).

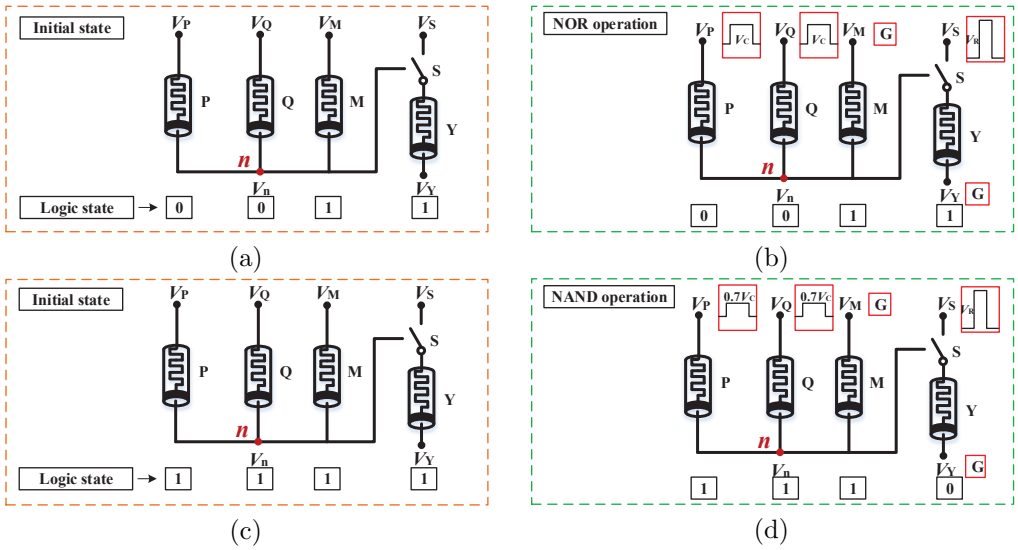

Figure A1 The logic function reconfiguration of the circuit for NOR and NAND operations. (a) Initialization before NOR operation; (b) NOR logic operation; (c) Initialization before NAND operation; (d) NAND logic operation.

Appendix B Overall information for complete 16 Boolean logic operations

For complete 16 Boolean logic operations, the specific applied voltage assignment, the node voltage formula, and the corresponding node voltage and output under different input combinations are summarized in Table B1, where F denotes floated, and the node voltages exceeding $0.4V_{\mathrm{C}}$ and the outputs being logic 0 are marked in bold. The node voltages and the outputs of each gate given in Table B1 correspond to the input combinations '00', '01', '10', and '11'.

The operating ways of remaining 12 Boolean functions except XOR and XNOR are the same as that of NOR and NAND, so they are not described in detail. In addition to the initialization, however, it takes two steps to implement XOR and XNOR logic operations. For XOR, in the first logic operation step, V_{P}, V_{Q}, and V_{M} are set to $0.7V_{\mathrm{C}}$, $0.7V_{\mathrm{C}}$, and G, respectively, which is equivalent to NAND logic operation, and the operation result stored in Y is regarded as an initial state of Y participated in the next logic operation step. In the second logic operation step, V_{P}, V_{Q}, and V_{M} are altered to G, G, and $0.7V_{\mathrm{C}}$, respectively, which accomplishes the XOR logic operation. The operating way of XNOR operation is the same as that of XOR operation, and the assignment of the applied voltages for XNOR is listed in Table B1.

* Corresponding author (email: duansk@swu.edu.cn)

Table B1 Overall information for complete 16 Boolean logic operations

Function	Voltage			Node voltage formula	Specific node voltage				Output			
	V_P	V_Q	V_M	V_n	V_n				y			
FALSE	V_C	G	V_C	$\frac{R_Q(R_P+R_M)}{R_Q(R_P+R_M)+R_P\cdot R_M}\cdot V_C$	$\mathbf{0.95V_C}$	$\mathbf{0.51V_C}$	$\mathbf{0.97V_C}$	$\mathbf{0.66V_C}$	**0**	**0**	**0**	**0**
p NOR q	V_C	V_C	G	$\frac{R_M(R_P+R_Q)}{R_M(R_P+R_Q)+R_P\cdot R_Q}\cdot V_C$	$0.09V_C$	$\mathbf{0.51V_C}$	$\mathbf{0.51V_C}$	$\mathbf{0.67V_C}$	1	**0**	**0**	**0**
q NIMP p	V_C	G	F	$\frac{R_Q}{R_P+R_Q}\cdot V_C$	$\mathbf{0.50V_C}$	$0.04V_C$	$\mathbf{0.95V_C}$	$\mathbf{0.50V_C}$	**0**	1	**0**	**0**
NOT p	V_C	F	G	$\frac{R_M}{R_P+R_M}\cdot V_C$	$0.04V_C$	$0.04V_C$	$\mathbf{0.50V_C}$	$\mathbf{0.50V_C}$	1	1	**0**	**0**
p NIMP q	G	V_C	F	$\frac{R_P}{R_P+R_Q}\cdot V_C$	$\mathbf{0.50V_C}$	$\mathbf{0.95V_C}$	$0.04V_C$	$\mathbf{0.50V_C}$	**0**	**0**	1	**0**
NOT q	F	V_C	G	$\frac{R_M}{R_Q+R_M}\cdot V_C$	$0.04V_C$	$\mathbf{0.50V_C}$	$0.04V_C$	$\mathbf{0.50V_C}$	1	**0**	1	**0**
p XOR q	$0.7V_C$	$0.7V_C$	G	$\frac{0.7R_M(R_P+R_Q)}{R_M(R_P+R_Q)+R_P\cdot R_Q}\cdot V_C$	$0.06V_C$	$0.35V_C$	$0.35V_C$	$\mathbf{0.46V_C}$	1	1	1	**0**
	G	G	$0.7V_C$	$\frac{0.7R_P\cdot R_Q}{R_M(R_P+R_Q)+R_P\cdot R_Q}\cdot V_C$	$\mathbf{0.63V_C}$	$0.33V_C$	$0.33V_C$	$0.23V_C$	**0**	1	1	**0**
p NAND q	$0.7V_C$	$0.7V_C$	G	$\frac{0.7R_M(R_P+R_Q)}{R_M(R_P+R_Q)+R_P\cdot R_Q}\cdot V_C$	$0.06V_C$	$0.35V_C$	$0.35V_C$	$\mathbf{0.46V_C}$	1	1	1	**0**
p AND q	G	G	V_C	$\frac{R_P\cdot R_Q}{R_M(R_P+R_Q)+R_P\cdot R_Q}\cdot V_C$	$\mathbf{0.90V_C}$	$\mathbf{0.48V_C}$	$\mathbf{0.48V_C}$	$0.33V_C$	**0**	**0**	**0**	1
p XNOR q	$0.7V_C$	G	F	$\frac{0.7R_Q}{R_P+R_Q}\cdot V_C$	$0.35V_C$	$0.03V_C$	$\mathbf{0.66V_C}$	$0.35V_C$	1	1	**0**	1
	G	$0.7V_C$	F	$\frac{0.7R_P}{R_P+R_Q}\cdot V_C$	$0.35V_C$	$\mathbf{0.66V_C}$	$0.03V_C$	$0.35V_C$	1	**0**	**0**	1
COPY q	F	G	$0.7V_C$	$\frac{0.7R_Q}{R_Q+R_M}\cdot V_C$	$\mathbf{0.66V_C}$	$0.35V_C$	$\mathbf{0.66V_C}$	$0.35V_C$	**0**	1	**0**	1
p IMP q	$0.7V_C$	G	F	$\frac{0.7R_Q}{R_P+R_Q}\cdot V_C$	$0.35V_C$	$0.03V_C$	$\mathbf{0.66V_C}$	$0.35V_C$	1	1	**0**	1
COPY p	G	F	$0.7V_C$	$\frac{0.7R_P}{R_P+R_M}\cdot V_C$	$\mathbf{0.66V_C}$	$\mathbf{0.66V_C}$	$0.35V_C$	$0.35V_C$	**0**	**0**	1	1
q IMP p	G	$0.7V_C$	F	$\frac{0.7R_P}{R_P+R_Q}\cdot V_C$	$0.35V_C$	$\mathbf{0.66V_C}$	$0.03V_C$	$0.35V_C$	1	**0**	1	1
p OR q	G	G	$0.7V_C$	$\frac{0.7R_P\cdot R_Q}{R_M(R_P+R_Q)+R_P\cdot R_Q}\cdot V_C$	$\mathbf{0.63V_C}$	$0.33V_C$	$0.33V_C$	$0.23V_C$	**0**	1	1	1
TRUE	G	$0.7V_C$	G	$\frac{0.7R_P\cdot R_M}{R_Q(R_P+R_M)+R_P\cdot R_M}\cdot V_C$	$0.03V_C$	$0.33V_C$	$0.01V_C$	$0.23V_C$	1	1	1	1

Appendix C Impact of resistance variation

Device variations such as resistance variation may impact the correctness of the logic operation. One of the key points to realize the logic operations proposed in this paper is the node voltage. The node voltage, however, could be influenced by the resistance variation of memristor. Furthermore, NAND is a universal logic gate, and based on Table B1, the distinction between its node voltages $0.46V_C$ (i.e., $p=1$, $q=1$) and $0.35V_C$ (i.e., $p=0$, $q=1$ or $p=1$, $q=0$) making S in two different states is the smallest. Therefore, taking NAND logic gate as an example, the impact of resistance variation on the node voltage V_n is discussed.

Assuming that the variation ratios of R_H and R_L are f and g ($f, g \in$(-1,1]), respectively, the changed high and low resistances are $R'_H=(1+f)\cdot R_H$ and $R'_L=(1+g)\cdot R_L$, and $R'_H/R'_L=(1+f)/(1+g)\cdot(R_H/R_L)$. Let the variation ratio $h=(1+f)/(1+g)$, then, $f=h(1+g)-1$. According to (3) and different input combinations, the changed node voltages can be concluded as follows.

Case 1: When $p=q=0$, i.e., $R_P=R_Q=(1+f)\cdot R_H$, the changed node voltage V'_n is

$$V'_n=\frac{0.7(1+g)R_L[(1+f)R_H+(1+f)R_H]}{(1+g)R_L[(1+f)R_H+(1+f)R_H]+(1+f)R_H\cdot(1+f)R_H}\cdot V_C. \tag{C1}$$

By substituting $f=h(1+g)-1$ into (C1), we can get

$$\begin{aligned}V'_n&=\frac{0.7(1+g)R_L[h(1+g)R_H+h(1+g)R_H]}{(1+g)R_L[h(1+g)R_H+h(1+g)R_H]+h(1+g)R_H\cdot h(1+g)R_H}\cdot V_C\\&=\frac{1.4R_L}{2R_L+h\cdot R_H}\cdot V_C.\end{aligned} \tag{C2}$$

Case 2: When $p=0$ and $q=1$, or $p=1$ and $q=0$, i.e., $R_P=(1+f)\cdot R_H$ and $R_Q=(1+g)\cdot R_L$, or $R_P=(1+g)\cdot R_L$ and $R_Q=(1+f)\cdot R_H$, the changed node voltage V'_n become

$$\begin{aligned}V'_n&=\frac{0.7(1+g)R_L[(1+f)R_H+(1+g)R_L]}{(1+g)R_L[(1+f)R_H+(1+g)R_L]+(1+f)R_H\cdot(1+g)R_L}\cdot V_C\\&=\frac{0.7(h\cdot R_H+R_L)}{2h\cdot R_H+R_L}\cdot V_C.\end{aligned} \tag{C3}$$

Case 3: When $p=q=1$, i.e., $R_P=R_Q=(1+g)\cdot R_L$, the changed node voltage V'_n is given by

$$\begin{aligned}V'_n&=\frac{0.7(1+g)R_L[(1+g)R_L+(1+g)R_L]}{(1+g)R_L[(1+g)R_L+(1+g)R_L]+(1+g)R_L\cdot(1+g)R_L}\cdot V_C\\&=\frac{7}{15}V_C.\end{aligned} \tag{C4}$$

Figure C1 shows that the node voltage of NAND logic gate varies with the variation ratio h. The blue solid line represents the change of node voltage for the input combination '01' or '10', and the node voltage gradually decreases and tends to $0.35V_C$ as h varies from 1/8 (such as $g=1$, $f=-0.75$) to 50 (such as $g=-0.97$, $f=0.5$). The magenta solid line indicates that the node voltage changes under the input combination '00', which gradually decreases and is always lower than $0.4V_C$. The black solid line represents the change of node voltage when the input combination is '11', and no matter how resistances change, the node voltage in this case is constant and always greater than $0.4V_C$. Based on the above analysis, the proposed logic gate shows strong robustness to resistance variation as h varies from 1/8 (i.e., $R'_H/R'_L=25$) to 50 (i.e., $R'_H/R'_L=10000$).

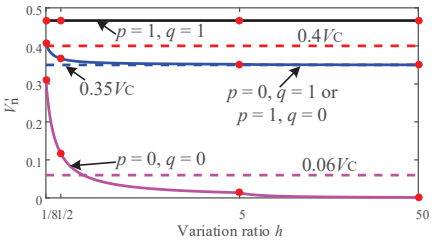

Figure C1 Changed node voltage of NAND logic gate with the variation ratio h.

Appendix D Logic operation steps and corresponding applied voltage levels for half adder and full adder

Figure D1 schematically illustrates the unidirectional crossbar arrays for the half adder, including two input memristors A and B, one load memristor M, and two output memristors C and D. For the one-bit half adder, there are two inputs (i.e., addend a and summand b) and two outputs (i.e., summary d and carry c). The logic functions are given by $d = a \oplus b$ and $c = a \cdot b$.

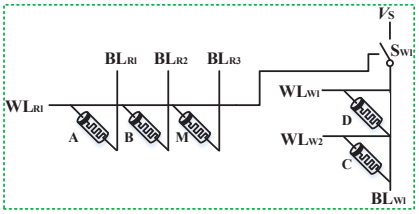

Figure D1 Half adder.

A and B are used to store inputs a and b, respectively. The sum output d and the carry output c are stored into D and C, respectively. Before executing the logic operation, assume that all memristors are preloaded into desired states. During the logic operation, the voltage V_S is set to V_R. The logic operation process of the one-bit half adder is described as follows.

Step 1: WL_{R1}, WL_{W2}, and BL_{W1} are all set to F, WL_{W1} and BL_{R3} are both G, and $0.7V_C$ is applied to BL_{R1} and BL_{R2} to execute the first step of XOR logic operation, which is equivalent to NAND logic operation. Hence, $d = \overline{a \cdot b}$ is performed and stored in D, whose resistance state is the initial state of D in the next step.

Step 2: F is applied to WL_{R1}, WL_{W2}, and BL_{W1}, G is applied to WL_{W1}, BL_{R1} and BL_{R2}, and $0.7V_C$ is applied to BL_{R3} to accomplish the XOR logic operation. Therefore, data in D is changed to $a \oplus b$, which is the final result of d.

Step 3: F is applied to WL_{R1}, WL_{W1}, and BL_{W1}, G is applied to WL_{W2}, BL_{R1} and BL_{R2}, and V_C is applied to BL_{R3} to execute the AND logic operation, i.e., $c = a \cdot b$. The carry of half adder is gained and stored in C.

The logic operation steps and corresponding applied voltage levels for half adder and full adder is listed in Table D1 and Table D2, respectively.

Table D1 The logic operation steps and corresponding applied voltage levels for the half adder

Step	Logic operation	WL_{R1}	WL_{W1}	WL_{W2}	BL_{R1}	BL_{R2}	BL_{R3}	BL_{W1}
1	① XOR(A, B) → D	F	G	F	$0.7V_C$	$0.7V_C$	G	F
2	② XOR(A, B) → D	F	G	F	G	G	$0.7V_C$	F
3	AND(A, B) → C	F	F	G	G	G	V_C	F

Appendix E Simulation results

A series of SPICE simulations of the proposed reconfigurable stateful logic design are conducted. Because of its feasibility, simplicity, accuracy, and generality, the VTEAM memristor model is utilized for simulation [1], and the SPICE model of VTEAM is given in [2]. In these simulation figures, only the output results of logic operations are shown.

Figure E1 shows the simulation results of NOR and NAND logic operations. The curves corresponding to U4 in Figure E1(a) and (b) represent the operation results of NOR and NAND, respectively. For the NOR logic operation, the operating voltages V_P, V_Q, V_M, V_Y, and V_S are assigned as V_C, V_C, G, G, and V_R, respectively. Figure E1(a) shows the simulation result of the NOR operation for inputs $p = 0$ and $q = 0$, and the resistance of the output memristor Y is R_L corresponding to logic 1. By setting the applied voltages V_P, V_Q, V_M, V_Y, and V_S to $0.7V_C$, $0.7V_C$, G, G, and V_R, respectively, the logic function of the circuit can be reconfigured as the NAND operation, whose simulation result for inputs $p = 1$ and $q = 1$ is shown in Figure E1(b). As can be seen in Figure E1(b), the output of the NAND operation for $p = 1$ and $q = 1$ is logic 0, and the delay of the logic operation is determined by the time of switching from R_L to R_H, which is about 4 nanoseconds (ns).

Figure E2 illustrates the behavior of the half adder for the addend $a = 1$ and the summand $b = 1$. The curve of each subfigure represents the result of a certain logic operation, and the specific logic operation has been marked below the subfigure. The logic operation time of each step is set to 6 ns. In the first logic operation step, the first step of XOR logic operation is executed, which is equivalent to NAND logic operation. Hence, the sum output $d = \overline{a \cdot b} = 0$ is performed, and the operation result is in situ stored as the resistance of the output memristor D, as shown in Step 1 of Figure E2. In the second operation step, the operation result stored into D in the first operation step is regarded as the initial state of D to execute the second step of XOR logic operation,

Table D2 The logic operation steps and corresponding applied voltage levels for the full adder

Step	Logic operation	WL_{R1}	WL_{R2}	WL_{R3}	BL_{R1}	BL_{R2}	BL_{R3}	WL_{W1}	WL_{W2}	WL_{W3}	BL_{W1}	BL_{W2}	BL_{W3}
1	① XOR(R_{11}, R_{12}) → W_{11}	F	G	G	$0.7V_C$	$0.7V_C$	G	G	F	F	F	F	F
2	② XOR(R_{11}, R_{12}) → W_{11}	F	G	G	G	G	$0.7V_C$	G	F	F	F	F	F
3	① XOR(W_{11}, W_{12}) → R_{21}	F	G	F	F	F	F	F	G	G	$0.7V_C$	$0.7V_C$	G
4	② XOR(W_{11}, W_{12}) → R_{21}	F	G	F	F	F	F	F	G	G	G	G	$0.7V_C$
5	AND(W_{11}, W_{12}) → R_{31}	F	F	G	F	F	F	F	G	G	G	G	V_C
6	AND(W_{21}, W_{22}) → R_{32}	F	F	G	F	F	F	G	F	G	G	G	V_C
7	OR(R_{31}, R_{32}) → W_{33}	G	G	F	G	G	$0.7V_C$	F	F	G	F	F	F

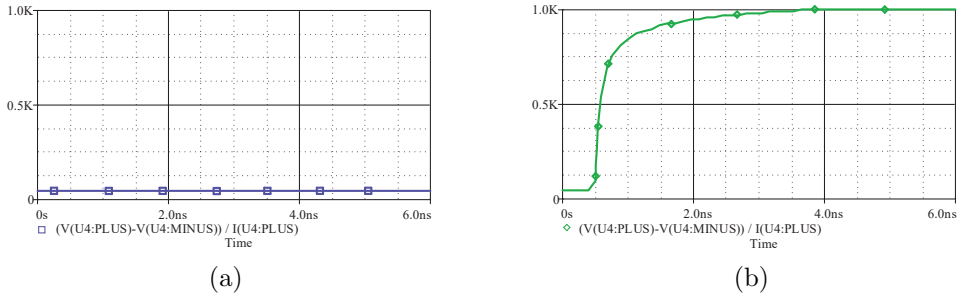

Figure E1 Simulation of the reconfigurable stateful logic operations. (a) NOR logic operation for inputs $p = 0$ and $q = 0$; (b) NAND logic operation for inputs $p = 1$ and $q = 1$.

accomplishing the XOR operation, i.e., $d = a \oplus b = 0$, which is the final result of d. As demonstrated in Step 2 of Figure E2, the sum output of the half adder for $a = 1$ and $b = 1$ is logic 0. In the third operation step, the AND logic operation between a and b is executed, that is, the carry $c = a \cdot b = 1$ is gained, as shown in Step 3 of Figure E2. Therefore, from Figure E2, the sum output d and the carry output c are logic 0 and 1, respectively, which are anastomotic with the theoretical values of the one-bit half adder for $a = 1$ and $b = 1$.

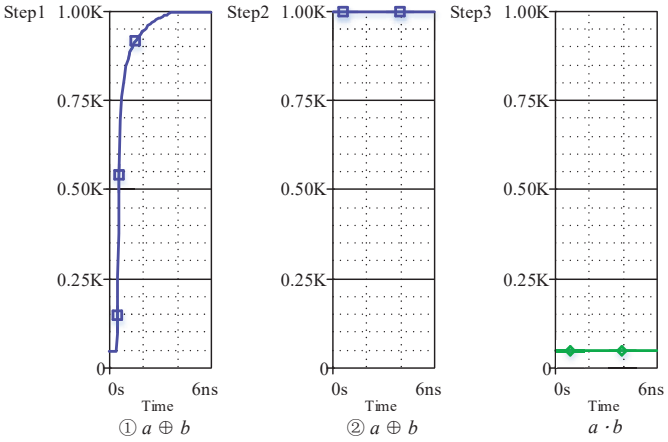

Figure E2 Simulation of the half adder for addend $a = 1$ and summand $b = 1$.

The simulation results of the one-bit full adder for the input combination $a = 1$, $b = 0$, and $c_{IN} = 1$ are shown in Figure E3. Based on Table D2, seven logic operation steps are required to realize the logic function of the one-bit full adder. Similarly, the logic operation time of each step is set to 6 ns. In the first two steps, the XOR logic operation between a and b is executed, and $a \oplus b = 1$, as demonstrated in Step 1 and Step 2 of Figure E3. As shown in Step 3 and Step 4 of Figure E3, in these two steps, the XOR logic operation between $a \oplus b$ and c_{IN} is achieved, that is, $a \oplus b \oplus c_{IN} = 0$, which is the sum output d of the full adder. In the Step 5 of Figure E3, the AND logic operation, $a \oplus b \cdot c_{IN}$, is executed, whose result is logic 1. The AND logic operation between a and b is performed, and its output is logic 0, as given in the Step 6 of Figure E3. In the final operation step, the results obtained in Step 5 and 6 are utilized as inputs for the OR gate, and after OR operation, the carry-out c_O is gained. Based on Figure E3, $d = 0$ and $c_O = 1$ are obtained.

References

1 Kvatinsky S, Ramadan M, Friedman E G, et al. VTEAM: A general model for voltage-controlled memristors. IEEE Trans Circuits Syst II, Exp Briefs, 2015, 62: 786-790
2 Memristor Models, https://asic2.group/tools/memrist-or-models/, accessed 16 September 2019

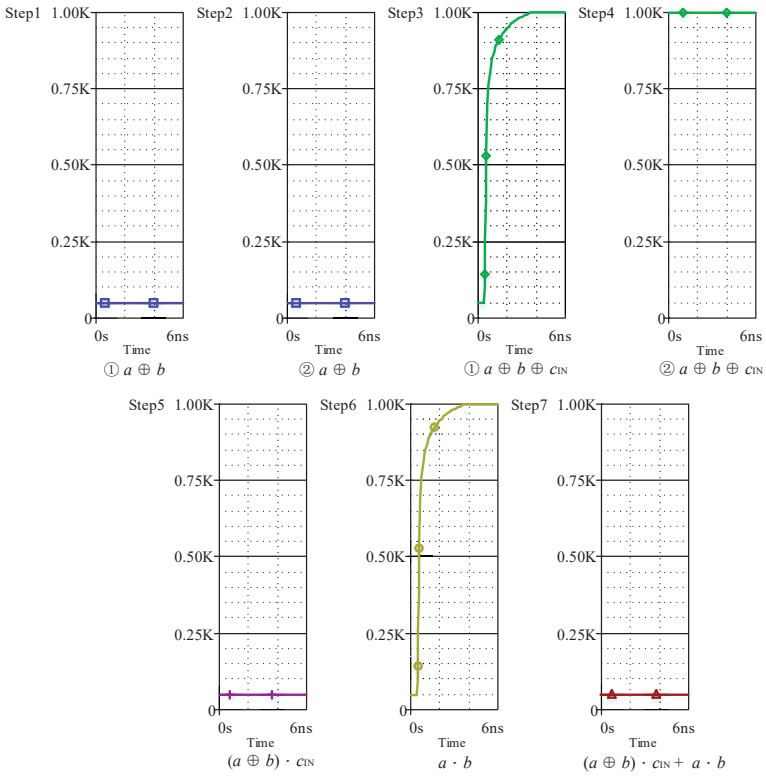

Figure E3 Simulation results of the one-bit full adder for addend $a = 1$, summand $b = 0$, and carry-in $c_{\mathrm{IN}} = 1$.