



# Wire-Crossings Optimization Based on Majority-of-Five and XOR-of-Three Primitives in QCA

Peng Liu<sup>1</sup> · Jianguo Ni<sup>1</sup> · Zhufei Chu<sup>1</sup>

Received: 22 September 2021 / Accepted: 3 January 2022 / Published online: 26 March 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

The difficulty of manufacturing and noise sensitivity of quantum-dot cellular automata (QCA) make wire-crossings optimization increasingly important. Novel logic primitives such as majority-of-five (MAJ<sub>5</sub>) and XOR-of-three (XOR<sub>3</sub>) with efficient QCA realizations can alleviate the wire-crossings issue. In this paper, we aim at 13 standard three-input Boolean functions, the QCA layouts of which are designed by introducing MAJ<sub>5</sub> (Method A) and its combinations with XOR<sub>3</sub> (Method B). The proposed designs are verified by the QCADesigner tool. Experimental results show we can reduce the number of QCA cells and area. In terms of area, we can achieve 30.69% and 48.35% reduction by Method A and B, respectively, compared with the state-of-the-art approach. The proposed designs are further used as building blocks for large-scale circuit designs. In one implementation, the Peres reversible gate is implemented based on Method B to decrease the area by 33.3%.

**Keywords** Quantum-dot cellular automata · Wire-crossings · Boolean function · Majority-of-five · Exclusive-OR

## 1 Introduction

Although complementary metal-oxide semiconductors (CMOS) have been scaling down in feature size, the increased density of chip integration is posing new challenges. The significant factors are leakage power consumption and tunneling effect [1–3]. Against this issue, several new computing paradigms based on novel computing devices are emerging. Quantum-dot cellular automata (QCA) [4] is one of these emerging nanotechnologies. Due to its small feature size, ultra-low power consumption, and high operational frequency, QCA has great potential for next-generation computing [5, 6].

With the development of QCA circuit design systems, the interconnections between electronic devices, i.e., wire-crossings is a problem that should not be underestimated [7]. Wire-crossings in QCA circuits are receiving increasing attention from the literature [8] because it consumes additional resources and increases the complexity of the QCA circuit.

---

✉ Zhufei Chu  
chuzhufei@nbu.edu.cn

<sup>1</sup> EECS, Ningbo University, Fenghua Road, Ningbo, 315211, China

Moreover, a large number of wire-crossings can lead to complex manufacturing problems [9]. This is because wire-crossings in QCA cannot be solved by multiple routing layers and vias as in traditional CMOS. Current automatic layout routing techniques for QCA are still mainly limited to single-layer physical layout implementations [10]. Hence, the design of QCA circuits has gradually shifted from the previous focusing on how to implement functions to how to reduce and further eliminate wire-crossing structures. Minimal the number of wire-crossings could improve circuit design. As a result, the number of wire-crossings has become a significant parameter in the QCA circuit evaluation system.

The main contributions of this paper are as follows.

1. We propose logic expressions of 13 standard three-input Boolean functions based on majority-of-five ( $\text{MAJ}_5$ ) primitive (Method A) and its combinations with XOR-of-three ( $\text{XOR}_3$ ) logical operation (Method B).
2. The QCA layouts of these 13 functions are presented with reduced number of wire-crossings. Comparing with previously published designs [11], our method achieves 30.69% and 48.35% reduction in QCA area by Method A and B, respectively.
3. The proposed QCA layouts are used as building blocks for large scale circuit design. As one example, the area of Peres reversible gate is reduced by 70.0%.

The remainder of the paper is organized as follows. Section 2 introduces the background of QCA and logic operations. Section 3 presents relevant research work. Section 4 shows the redesigned diagrams of the 13 standard three-input Boolean functions with  $\text{MAJ}_5$  and  $\text{XOR}_3$ . Afterwards, we perform experiment result obtained by our method as well as case studies in Section 5. We conclude the paper in Section 6.

## 2 Background

In this section, the backgrounds of QCA and logic primitives are briefly introduced.

### 2.1 QCA

There are several different implementations types of QCA: metal-island QCA, semiconductor-based QCA, atomic-based QCA, molecular QCA, and magnetic QCA [12]. Metal-island QCA cells [13] are based on aluminum islands coupled by aluminum-oxide tunnel junctions and capacitors, fabricated on an oxidized silicon wafer. Here electrons can tunnel between dots using tunnel junctions. The two connecting pairs of islands are coupled to each other using capacitors. Semiconductor-based QCA [14] can operate at significantly higher temperatures than metal-dot devices. in which four GaAs/AlGaAs semiconductor quantum dots are placed at four different corners. Each cell is also capacitively coupled. Atomic-based QCA [15] is prospective for highly complex and room temperature operational QCA circuitry with atomic silicon quantum dots (ASiQDs). Molecular QCA [16] is capable of self-assembling to form larger circuits and operating at room temperature. Molecular QCA offers very high device densities, low power dissipation, and ways to directly integrate sensors with QCA logic and memory elements. Magnetic QCA [17] is a nanomagnetic particle which switching times are slower. However, nanoscale magnets offer the advantages of large energy differences between states, tolerance to the effects of radiation, and the demonstrated room-temperature operation. Hence, Magnetic QCA has the potential integration with other magnetic systems such as magnetic random access memory

(MRAM), make this a potentially useful technology in spite of its relatively low speed compared with conventional electronic devices. Consequently, numerous experiments on QCA physical realization have been made in the past few years including [18–20].

Every type of QCA mentioned before except magnetic QCA all mimic the same behavioral model of a square cell with a length of 18 nm, containing quantum dots at the four corners and two free electrons among the dots [21], which is shown in Fig. 1(a). The four quantum dots are occupied by a tunneling barrier and arranged in a 2 × 2 fashion. The two free electrons in the cell can tunnel between these dots. As the signals pass, the electrons will always be at diagonal positions as a result of Coulomb force repulsion. Therefore, there are two types of energy polarization exist, i.e.,  $P = +1$  and  $P = -1$ . Consequently, logic 1 and 0 can be represented by the polarization of the cell, respectively. Multiple cells are used to form the logic gates and transmission lines. Note that QCA circuits work differently from conventional CMOS circuits. It uses a four-phase clock to control the distribution of electrons, i.e., the transmission of signals, which can be seen in Fig. 2. The four phases are switch, hold, release, and relax. Moreover, QCA’s information transmission is current-free, which results in ultra-low-power consumption [22].

The three most typical primitive of QCA are transmission line, majority-of-three logic gate (MAJ<sub>3</sub>) (Fig. 1(b)), and inverter (INV) (Fig. 1(c)) [21]. The Boolean logic expression for the MAJ<sub>3</sub> is denoted as

$$M_3(A, B, C) = AB + AC + BC. \tag{1}$$

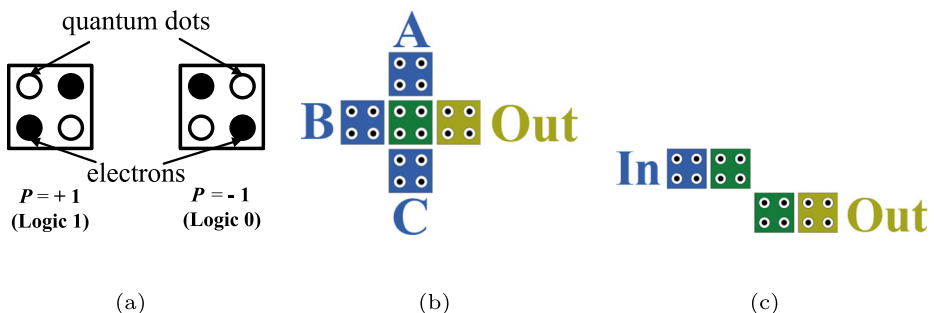
When any one of the three inputs is set to logic 0 or 1, the AND and OR logic gates can be obtained, respectively. For example:  $M_3(A, B, 0) = AB$  and  $M_3(A, B, 1) = A + B$ .

With the development of QCA, more and more basic devices based on explicit interactions between QCA cells have been designed, such as majority-of-five (MAJ<sub>5</sub>), XOR-of-three (XOR<sub>3</sub>), etc. The introduction of these new devices allows for improvements in both the QCA implementation area and delay.

## 2.2 QCA Wire-Crossing

### 2.2.1 Coplanar

Coplanar crossing refers to the transfer of two crossing singles in the same plane in QCA. By rotating cell as shown in Fig. 3(a). The horizontal transmission line is standard cells and the vertical transmission line is the rotated cells. Because the four quantum dots of the



**Fig. 1** Schematics of QCA basis, (a) QCA cell, (b) majority-of-three (MAJ<sub>3</sub>), and (c) Inverter

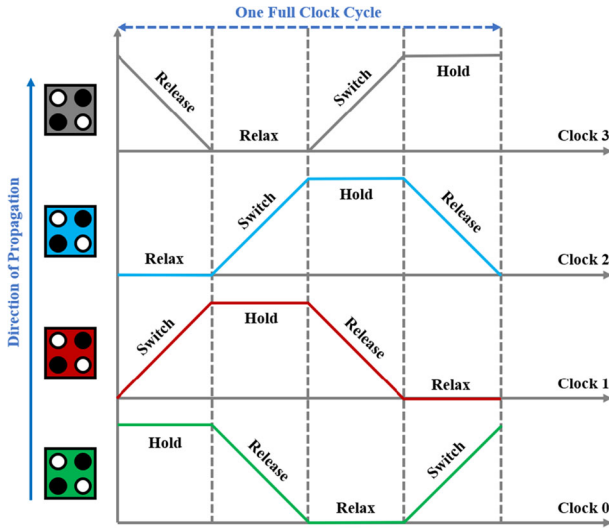


Fig. 2 Schematic of four-phase clock

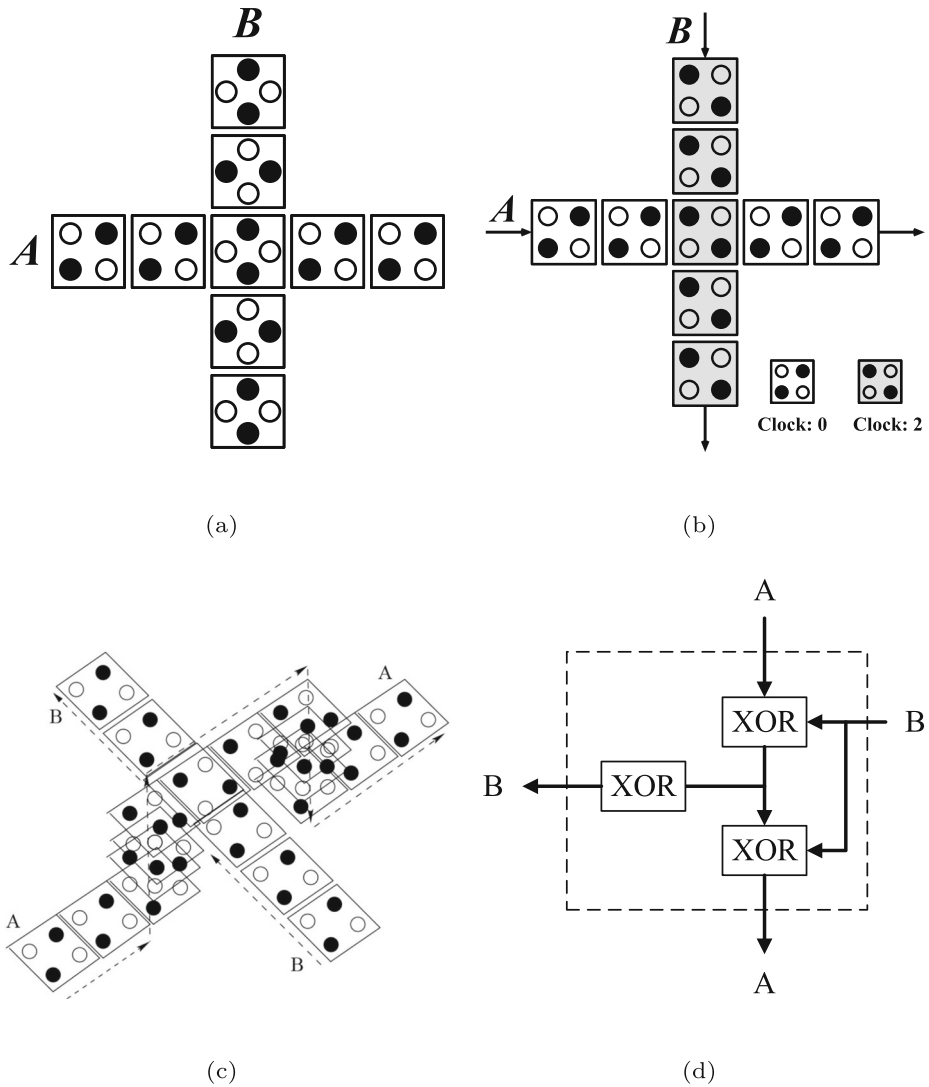
rotating cell and standard cell are in different positions, the influence of Coulomb interaction between different transmission lines in the other direction is not enough to change the polarization value. Thus, the signal coplanar cross-transmission is realized. Although rotating coplanar crossing structure does not bring extra redundancy, the stability of rotating cells is weaker than it could have been in a normal wire [23]. One can refer to [24] for more information on the comparison of various coplanar wire-crossing properties.

### 2.2.2 Robust Coplanar

Robust coplanar wire-crossing is based on clock region. The behavior of cells in the non-adjacent clock region is mutually independent, that is to say, cells separated by 180° phase difference, e.g., Clock phase 0 and Clock phase 2, are independent of each other in the process of information transmission [25]. As shown in Fig. 3(b), the cells at the intersection of horizontal and vertical transmission lines are separated by half a clock cycle. Hence the signal transmission will not cause interference. This is a pretty good way to solve the crossing problem. However, sometimes to match the clock difference, the robust coplanar crossing structure will increase the area and delay of the circuit.

### 2.2.3 Multilayer

It is very straightforward to understand the multilayer crossing, in which one transmission line crosses over the top of the other. Hence, the lines are not on the same plane. The implementation of multilayer crossing requires at least three layers. As shown in Fig. 3(c), it is designed as a crossing bridge where the signals of two transmission lines do not interfere with each other in the process of transmission. Thus, the cross-transmission of signals is implemented. But this technique brings more complexity to the layout implementation and the difficulty of the implementation will increase as well.



**Fig. 3** Schematics of three types of wire-crossings, (a) traditional coplanar, (b) robust coplanar, (c) multilayer, and (d) crossing two wires by using three XORs in [26]

### 2.2.4 Crossing Elimination by Using Three XORs

In addition to addressing the wire-crossings on the level of physical layout, we can also handle wire-crossings on the logical level. As proposed method in [26], one wire-crossing can be resolved by using three XOR gates. However, it should be noted that each XOR consists of four majority gates proposed in [26]. This structure can reduce all wire-crossings to zero at the cost of introducing three additional XOR gates, which is quite costly.

### 2.3 Majority-of-Five

Efficient MAJ<sub>5</sub> logic gates based on explicit interactions between QCA cells are emerging. Meanwhile, researchers have shown that designs based on MAJ<sub>5</sub> are efficient in terms of the area and delay. Many different kinds of QCA MAJ<sub>5</sub> gate structures were proposed [27–32]. Figure 4(a) gives the schematic diagram of an MAJ<sub>5</sub>, which will be used in the designs of this paper. An MAJ<sub>5</sub> is a Boolean logic gate whose output is true only when it has three or more inputs of logic 1. The logic expression of an MAJ<sub>5</sub> can be expressed as:

$$\begin{aligned} \mathbb{M}_5(A, B, C, D, E) = & ABC + ABD + ABE + ACD \\ & + ACE + ADE + BCD \\ & + BCE + BDE + CDE. \end{aligned} \tag{2}$$

MAJ<sub>5</sub> can be expressed in terms of MAJ<sub>3</sub>, which resulted in an expression (3) of optimal depth using exact synthesis [33].

$$\mathbb{M}_5(A, B, C, D, E) = \mathbb{M}_3(A, \mathbb{M}_3(B, C, D), \mathbb{M}_3(\mathbb{M}_3(A, B, C), D, E)). \tag{3}$$

When any two of MAJ<sub>5</sub> inputs are set to logic 0 or 1, a three-input AND gate and a three-input OR gate can be implemented, respectively, which can be seen in (4):

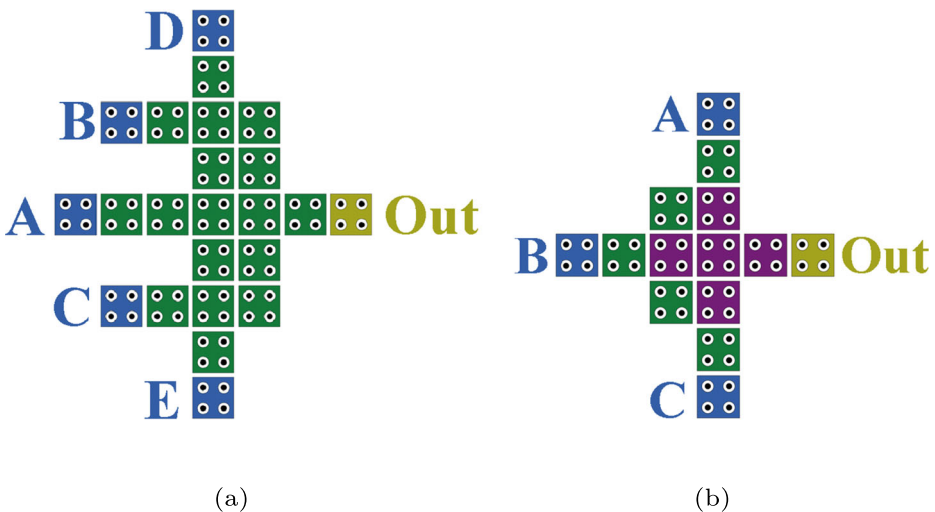
$$\begin{aligned} \mathbb{M}_5(A, B, C, 1, 1) &= A + B + C, \\ \mathbb{M}_5(A, B, C, 0, 0) &= ABC. \end{aligned} \tag{4}$$

Moreover, MAJ<sub>5</sub> can also be reduced to MAJ<sub>3</sub> when either one of the following two conditions is met [34]:

1. There exists two pairs of duplicated inputs.
2. Any two inputs are set to 0 and 1 respectively.

The expressions can be seen in (5)

$$\mathbb{M}_5(A, A, B, B, C) = \mathbb{M}_5(A, B, C, 0, 1) = \mathbb{M}_3(A, B, C). \tag{5}$$



**Fig. 4** Schematic diagrams of majority-of-five and XOR-of-three used in this paper, (a) majority-of-five in [27] and (b) XOR-of-three in [36]

Due to symmetries, more identities can be obtained as follows:

$$\begin{aligned}
 \mathbb{M}_5(A, A, B, B, C) &= \mathbb{M}_5(A, A, B, C, C) \\
 &= \mathbb{M}_5(A, B, B, C, C) \\
 &= \mathbb{M}_5(A, B, C, 0, 1).
 \end{aligned}
 \tag{6}$$

### 2.4 XOR-of-Three

Exclusively-OR (XOR) logic gates have a wide range of applications in digital logic circuits, especially in arithmetic logic circuits. Thus, the introduction of XOR can help to further optimize the circuit area and latency [35]. XOR can also be designed by the explicit interactions between QCA cells. A stable and reliable infrastructure with low energy consumption can be obtained [36–39]. Figure 4(b) shows the schematic diagram of XOR<sub>3</sub> adopted in this paper. Extended from XOR<sub>2</sub>, the Boolean function of a XOR<sub>3</sub> can be expressed as:

$$\mathbb{X}_3(A, B, C) = A \oplus B \oplus C.
 \tag{7}$$

When any one of the inputs is set to logic 0 or 1, the XOR and exclusively-NOR (XNOR) can be obtained, respectively. For example:

$$\begin{aligned}
 \mathbb{X}_3(A, B, 0) &= A \oplus B, \\
 \mathbb{X}_3(A, B, 1) &= A \odot B.
 \end{aligned}
 \tag{8}$$

### 3 Related Work

There are considerable researches on QCA wire-crossings optimization. The most typical way is operated by logic restructuring, in which nodes are duplicated or swapped to generate a functional-equivalent circuit with fewer wire-crossings [40–42]. The logic restructuring method leads to a tradeoff between wire-crossings and the area. The methods for the automatic placement and routing have been introduced in [43]. There are mainly three steps: intermediate function generation, feedback function generation, and inputs rearrangement for solving wire-crossing issues. However, this methodology is limited to three variables propagating to single output functions. As for maintaining a set of small-scale functions of optimized wire-crossings, the methods based on a universal logic gate or standard functions are presented [8, 11, 44, 45]. Although the universal logic gate method reduces wire-crossings, cell count, and operating clock cycles, it does not lead to high efficiency in circuit designs. In [8], 4-variable logic functions are investigated and simplified with the K-Map as sum of product (SOP) form. The heuristic K-Map includes minterms selecting, inputs reordering, inputs postponement, and universal logic gate using. Afterward, it attempts to replace remaining wire-crossings with three XOR gates, as mentioned in Section 2.2.4. In [11], it provides some novel diagrams for the implementation of standard three-input Boolean functions with the aim of reducing the number of wire-crossings, using five-input majorities and robust coplanar wire-crossing [25]. Then, a new technique is used to merge multiplexers when Boolean functions with more than three inputs are implemented. In this paper, we optimized this approach and proposed a better design.

**Table 1** 13 standard three-input Boolean functions proposed in [46]

Func.	Logic expressions	Original diagram (presented in [46])
1	$f = abc$	
2	$f = ab$	
3	$f = abc + a\bar{b}\bar{c}$	
4	$f = abc + \bar{a}\bar{b}\bar{c}$	
5	$f = ab + bc$	
6	$f = ab + \bar{a}\bar{b}\bar{c}$	
7	$f = abc + \bar{a}\bar{b}\bar{c} + a\bar{b}\bar{c}$	
8	$f = a$	
9	$f = ab + bc + ac$	
10	$f = ab + \bar{b}c$	
11	$f = ab + bc + \bar{a}\bar{b}\bar{c}$	
12	$f = ab + \bar{a}\bar{b}$	
13	$f = abc + \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + a\bar{b}\bar{c}$	



## 4 Standard Three-Input Boolean Functions

Logical diagrams are presented in this section for 13 standard three-input Boolean functions introduced in [46] (Table 1). First, MAJ<sub>5</sub> is applied as an implementation primitive, which is shown in the “Method A” column in Table 2. A combination of MAJ<sub>5</sub> with XOR<sub>3</sub> is then applied to the designs as shown in the “Method B” column in Table 2. Moreover, the QCA layouts for all of the 13 standard functions proposed in this paper are illustrated in “Appendix A” (Method A) and “Appendix B” (Method B).

### 4.1 Cubes of Three-Input Boolean Functions

A Boolean function of  $n$  variables can be represented by an  $n$ -dimensional binary hypercube. In this paper,  $n=3$  is taken to represent the three-input Boolean functions, the cube of which is shown in Fig. 5. For a total of  $2^{2^3} = 256$  distinct Boolean functions, they can be classified into 13 standard three-input Boolean functions. The trivial cases of constants 0 and 1 are not considered. The “Original Diagram” column in Table 1 shows the logic expressions and diagrams proposed in [46]. The functions are labeled using index  $i \in [1, 13]$ .

The concept of the standard Boolean function is similar to the Negation-Permutation-Negation (NPN) equivalence classes [47]. The NPN-equivalence classes is defined as follows: two functions  $f(x_1, \dots, x_n)$  and  $g(x_1, \dots, x_n)$  are NPN-equivalent if there exists an inversion of the inputs ( $x_i \rightarrow \bar{x}_i$ ) (**N**), a permutation of the inputs ( $x_i x_j \rightarrow x_j x_i$ ) (**P**), or an inversion of the output ( $f \rightarrow \bar{f}$ ) (**N**) so that  $f$  and  $g$  can be made Boolean equivalent. By grouping similar functions into NPN equivalence classes, the representations are compact since  $n$ -input Boolean functions can be classified into 14 and 222 classes, for  $n = \{3, 4\}$  respectively.

### 4.2 Proposed Majority-of-Five Representations of the Standard Functions

We first present the proposed logic expression using MAJ<sub>5</sub> primitives. The logic diagrams are shown in the “Method A (using MAJ<sub>5</sub>)” column in Table 2.

- *Functions 2 and 9* There is nothing that has changed since these functions were already represented by one MAJ<sub>3</sub> gate [11, 46].
- *Functions 1, 5, and 8* The diagrams proposed in [11] improved the original diagrams in [46]. Especially for functions 1 and 5, the logic depth was optimized by introducing MAJ<sub>5</sub>. We also adopt these designs proposed in [11].
- *Functions 3, 6, and 12* These functions can be efficiently implemented using MAJ<sub>5</sub>, which is shown in the “Method A (using MAJ<sub>5</sub>)” column. The robust coplanar wire-crossings in the previous designs [11] are eliminated, which leads to the reduction of the QCA implementation area and the enhancement of the module stability. Especially, function 3 not only eliminates the robust coplanar wire-crossing but also removes the traditional coplanar wire-crossing which is implemented by rotating cells. Functions 3 and 12 are designed by combining one MAJ<sub>3</sub> and one MAJ<sub>5</sub>. In contrast, function 6 is implemented by two MAJ<sub>5</sub>s.
- *Function 10* This function designed by [11] was already with no wire-crossing and a small QCA implementation area. We proposed a structure consisting of one MAJ<sub>5</sub> and one MAJ<sub>3</sub> in this paper.

**Table 2** Schematics of simplified 13 standard three-input Boolean functions given in [46]

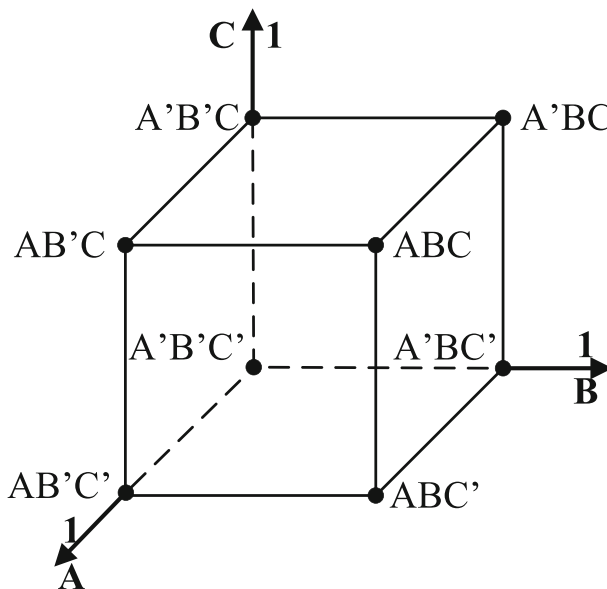
F. Logic expression	Diagram in [11]	Method A (using MAJ <sub>5</sub> )	Method B (using XOR <sub>3</sub> )
1 $f = abc$		The same as [11]	The same as [11]
2 $f = ab$		The same as [11]	The same as [11]
3 $f = abc + a\bar{b}\bar{c}$			
4 $f = abc + \bar{a}\bar{b}\bar{c}$			
5 $f = ab + bc$		The same as [11]	The same as [11]
6 $f = ab + \bar{a}\bar{b}\bar{c}$			
7 $f = abc + \bar{a}\bar{b}\bar{c} + a\bar{b}\bar{c}$			The same as Method A
8 $f = a$		The same as [11]	The same as [11]
9 $f = ab + bc + ac$		The same as [11]	The same as [11]
10 $f = ab + \bar{b}\bar{c}$			
11 $f = ab + bc + \bar{a}\bar{b}\bar{c}$			
12 $f = ab + \bar{a}\bar{b}$			
13 $f = abc + \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}\bar{c} + a\bar{b}\bar{c}$			

- *Functions 4, 7, 11, and 13* The long wire robust wire-crossings in the previous designs [11] in these four functions are removed. The structure of too-long wire in designs is not so freely be routed and connected to other circuits. Although each of these four proposed designs still has one robust wire-crossing at the primitive inputs which increases the stability while bringing an additional 0.5 cycles, the QCA implementation area is more compact.

### 4.3 Proposed Hybrid Representation of Standard Functions

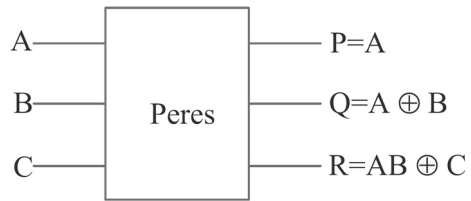
The functions are further studied by using hybrid representation based on MAJ<sub>5</sub> and XOR<sub>3</sub>. The logic diagrams are shown in the “Method B (using XOR<sub>3</sub>)” column in Table 2.

- *Functions 1, 2, 5, 7, 8, and 9* These functions are left unchanged because they are already well-designed shown in the previous discussions.
- *Functions 12 and 13* Function 13 can be represented by one XOR<sub>3</sub> while function 12 is an XNOR. As mentioned earlier in Section 2.4, it can be obtained by setting any of the primitive inputs of XOR<sub>3</sub> to logic 1. Hence, the two functions can be implemented by one single QCA logic gate, which not only simplifies the circuit area but also eliminates all types of wire-crossings.
- *Functions 3, 4, and 11* These functions can be implemented in another efficient way by introducing XOR<sub>3</sub>. All types of the coplanar wire-crossings are eliminated compared to the previous designs [11]. Especially for functions 4 and 11, the hybrid designs remove the wire-crossings that could not be eliminated by introducing MAJ<sub>5</sub> alone.



**Fig. 5** The cube of the three-input Boolean functions

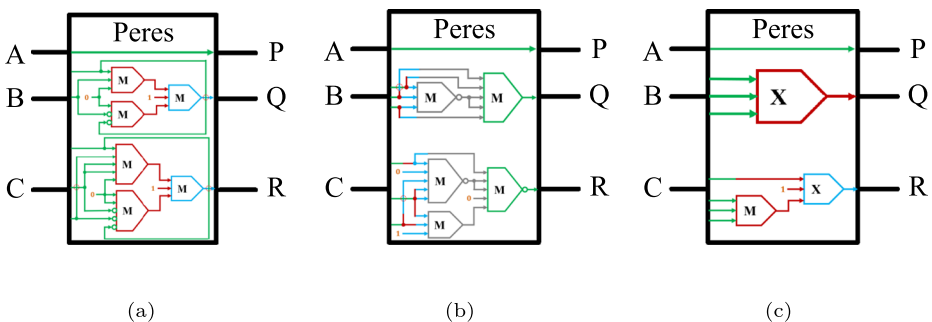
**Fig. 6** Schematic of Peres reversible gate



- *Function 6* The hybrid representation also eliminates the wire-crossing for function 6. Although the number of logic gates and levels are the same, the hybrid designs leads to better area.
- *Function 10* The hybrid design also has no wire-crossings. One can select appropriate implementations proposed by [11] or ours in real designs.

#### 4.4 An Example of Using Proposed Diagrams

We use the reversible Peres logic gate [48] as an example to demonstrate the proposed wire-crossing optimization. The schematic and the logic expressions of Peres logic gate over Boolean variables A, B, and C are shown in Fig. 6, in which P, Q, and R are outputs. For a circuit module, the inputs and outputs are often fixed. In this case, we have simplified the inputs layout to highlight the idea of using 13 standard three-input Boolean functions for substitution in order to clarify the usage. However, it is important to note that the inputs layout can also have a significant impact on reducing wire-crossings, which is not explored in this paper. One can refer to [43] for further information. The gate designed with diagrams proposed in [11] is shown in Fig. 7(a). Meanwhile, the gate designed using MAJ<sub>5</sub> is shown in Fig. 7(b). The internal modules of output Q and output R are replaced with the standard functions 11 and 12 designed by MAJ<sub>5</sub>. It can be seen that the robust wire-crossings existing in the previous circuit have been reduced by using XOR<sub>3</sub>, which is shown in Fig. 7(c). The use of XOR<sub>3</sub> allows for a more efficient reduction of wire-crossings and circuit area.



**Fig. 7** Schematics of Peres reversible gate using different diagrams, (a) diagrams proposed in [11], (b) diagrams using MAJ<sub>5</sub>, and (c) diagrams using XOR<sub>3</sub>

**Table 3** Comparison of different diagrams implementing 13 standard three-input Boolean functions

Func.	Latency (clock)			Complexity (number of cells)			Area (nm <sup>2</sup> )		
	[11]	A	B	[11]	A	B	[11]	A	B
1	0.5	0.5	0.5	30	30	30	27,738	27,738	27,738
2	0.5	0.5	0.5	13	13	13	11,858	11,858	11,858
3	0.75	0.75	0.75	92	50	39	89,358	43,164	43,524
4	0.75	1.25	0.75	144	99	49	143558	97,524	60,604
5	0.5	0.5	0.5	29	29	29	27,738	27,738	27,738
6	0.75	0.75	0.75	91	68	47	89,698	56,355	59,386
7	0.75	1.25	1.25	133	92	92	136,398	73,684	73,684
8	0.25	0.25	0.25	5	5	5	4,018	4,018	4,018
9	0.5	0.5	0.5	13	13	13	11,858	11,858	11,858
10	0.75	0.75	0.75	38	49	40	39,798	43,164	43,590
11	0.75	1.25	0.75	150	94	30	151,578	97,676	35,443
12	0.75	0.75	0.5	56	51	19	52,598	47,421	21,080
13	0.75	1.25	0.5	78	73	24	77,778	56,604	25,724
Total	8.25	10.25	8.25	872	666	430	863,974	598,802	446,245
Improv.		- 24.24%	0		23.62%	50.69%		30.69%	48.35%
Ratio	1	1.24	1	1	0.76	0.49	1	0.69	0.52
		1	0.80		1	0.65		1	0.75

**Table 4** Simulation engine parameters of QCADesigner

Parameters	Bistable approximation	Coherence vector
Number of samples	16000	-
Convergence tolerance	0.001	-
Radius of effect	65 nm	80 nm
Relative permittivity	12.9	12.9
Clock high	$9.80 \times 10^{-22}$ J	$9.80 \times 10^{-22}$ J
Clock low	$3.80 \times 10^{-23}$ J	$3.80 \times 10^{-23}$ J
Clock shif	0	0
Clock amplitude factor	2	2
Layer separation	11.5 nm	11.5 nm
Maximum Iterations per sample	100	-
Temperature	-	1K
Relaxation time	-	$1 \times 10^{-15}$ s
Time step	-	$1 \times 10^{-16}$ s
Total simulation time	-	$7 \times 10^{-11}$ s



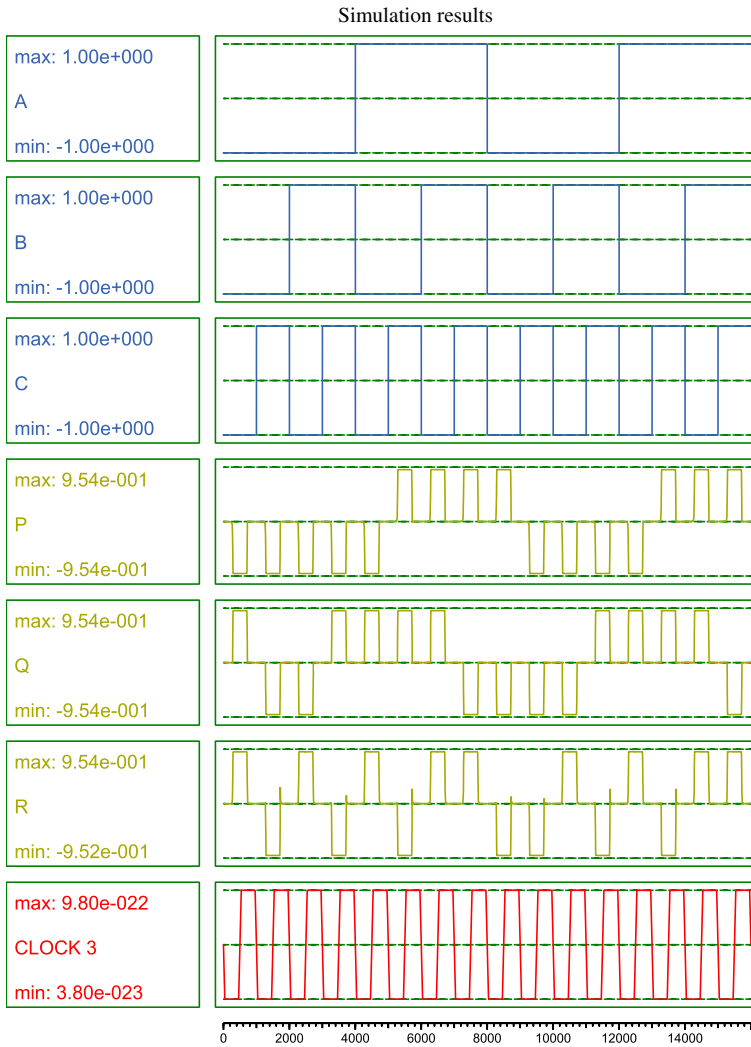


Fig. 9 QCA simulation result of the Peres

Table 5 shows the comparison results of different diagrams (proposed in [11], using MAJ<sub>5</sub>, and using XOR<sub>3</sub>) implementing Peres gate when realized in QCA. It should be noted that for the sake of visualising the advantages of our design we have only counted the area

Table 5 Comparison of different diagrams implementing Peres gate

Design	Latency (clock)			Number of wire-crossings			Area ( $\mu\text{m}^2$ )		
	[11]	A	B	[11]	A	B	[11]	A	B
Peres	1.25	1.25	0.75	4	1	0	0.20	0.19	0.06
Improv.		0%	40%		75%	100%		5.00%	70.0%

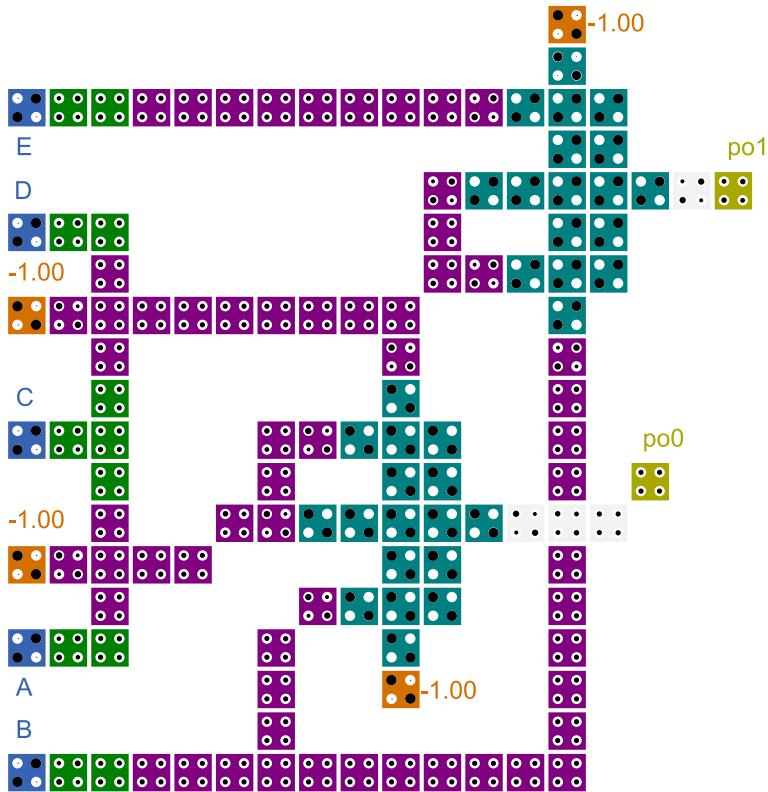


Fig. 10 QCA layout of the c17 using proposed diagrams

and the number of crossed lines in the red dotted box in Fig. 8. By comparing, Fig. 8(c) is the best design, i.e., using XOR<sub>3</sub>, which occupies an area of 0.06 μm<sup>2</sup> and 0.75 clock cycles with no wire-crossings.

As for c17, the QCA layout and simulation results are shown in Figs. 10 and 11, respectively. Our design occupies an area of 0.14 μm<sup>2</sup> and 1 clock cycle with only one robust wire-crossings. Table 6 shows the comparison results of c17 between the XMG presented in [50] and method proposed in this paper. The comparison result indicates that our method can achieve 42.86%, 88.89%, and 46.15% reduction in latency, the number of wire-crossings, and area, respectively.

## 6 Conclusions

Wire-crossing is a very important issue in QCA, which imposes significant complexity, area dissipation, noise sensitivity, and signal weakness. Aiming at reducing the number of wire-crossings, the majority-of-five gate and the exclusively-OR gate are introduced to address this issue. The diagrams and logic expressions of the simplified 13 standard three-input



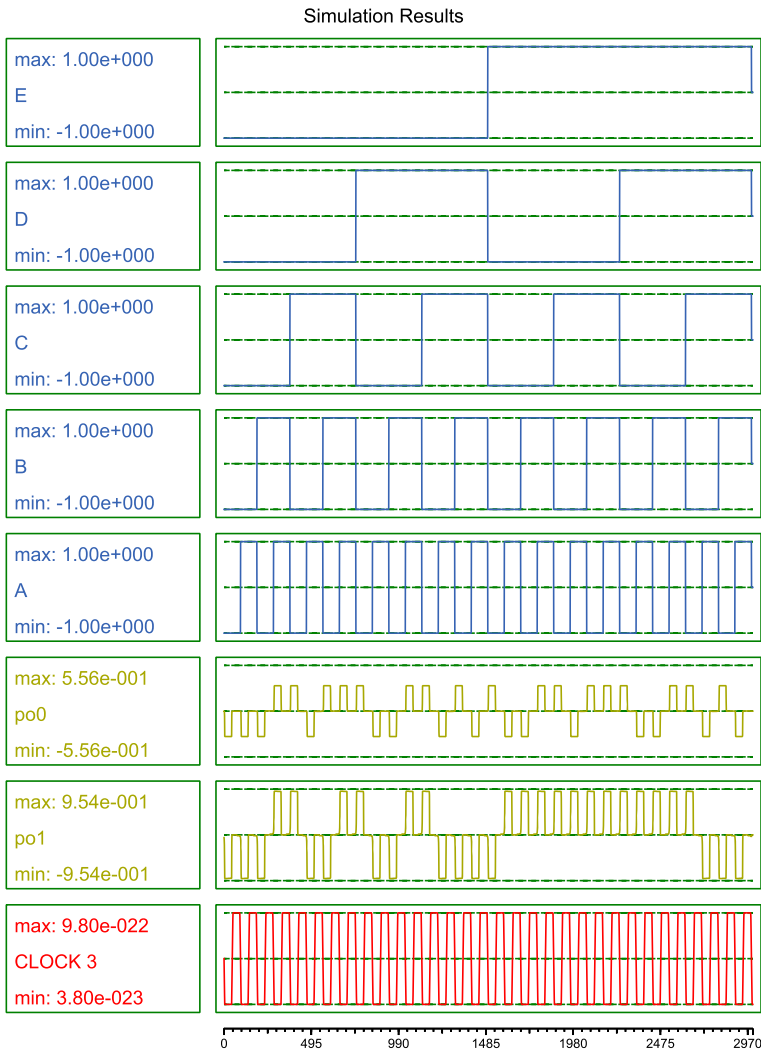


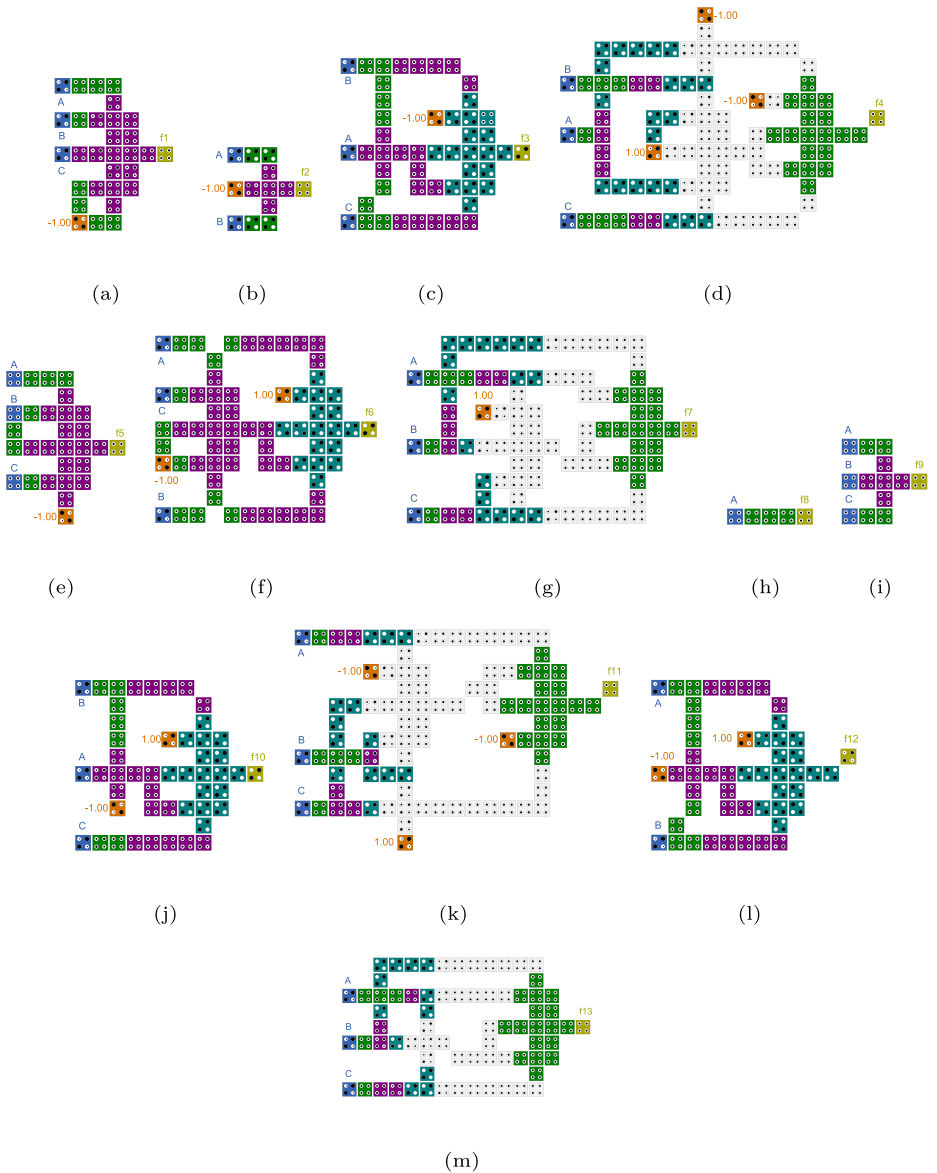
Fig. 11 QCA simulation result of the c17

Boolean functions are proposed, which results in fewer wire-crossings. The experimental results show that we get improvement in size by 30.69% and 48.35% when introducing MAJ<sub>5</sub> and XOR<sub>3</sub>, respectively.

Table 6 Comparison of c17 between the XMG presented in [50] and ours

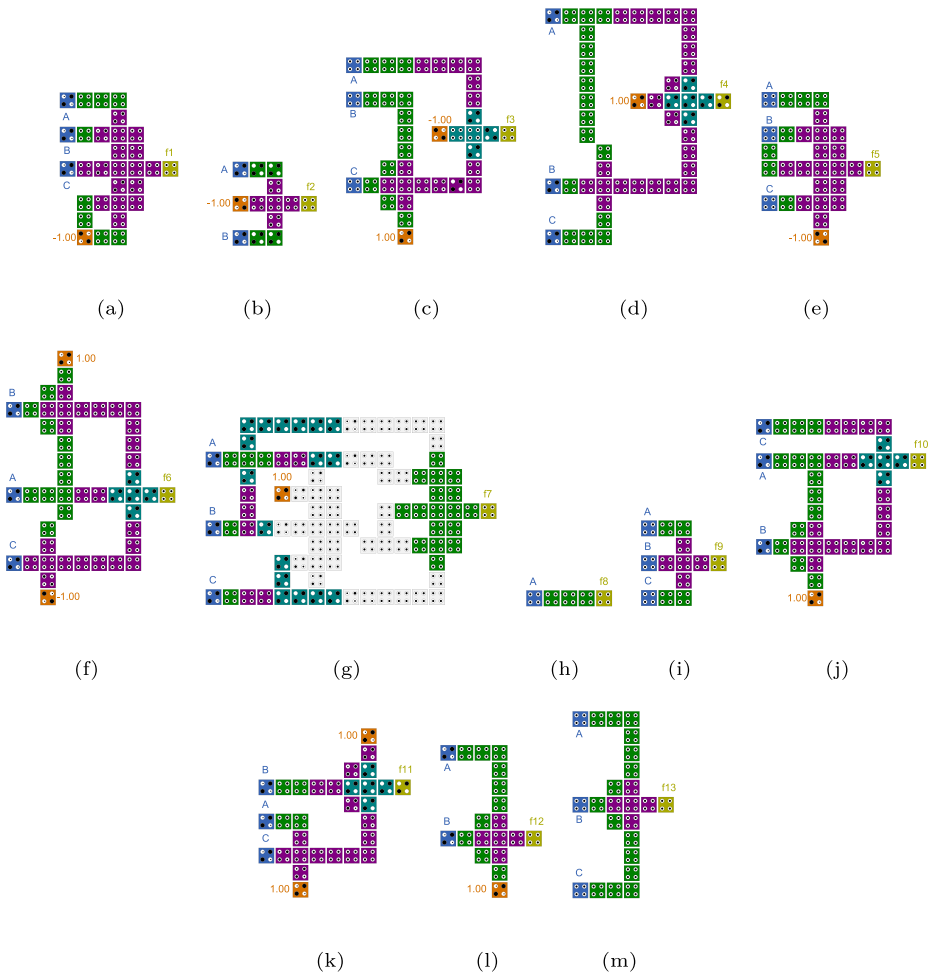
Design	Latency (clock)	Number of wire-crossings	Area ( $\mu\text{m}^2$ )
[50]	1.75	9	0.26
Our method	1.00	1	0.14
Improv.	42.86%	88.89%	46.15%

### Appendix A: QCA Layouts of of 13 Standard with Method A



**Fig. 12** The proposed QCA layouts of 13 functions based on Method A, (a) Function 1, (b) Function 2, (c) Function 3, (d) Function 4, (e) Function 5, (f) Function 6, (g) Function 7, (h) Function 8, (i) Function 9, (j) Function 10, (k) Function 11, (l) Function 12, (m) Function 13

## Appendix B: QCA Layouts of of 13 Standard with Method B



**Fig. 13** The proposed QCA layouts of 13 functions based on Method B, (a) Function 1, (b) Function 2, (c) Function 3, (d) Function 4, (e) Function 5, (f) Function 6, (g) Function 7, (h) Function 8, (i) Function 9, (j) Function 10, (k) Function 11, (l) Function 12, (m) Function 13

**Author Contributions** P. Liu proposed the logic expressions for 13 standard three-input Boolean functions, and J. Ni simulated and verified the proposed designs using QCADesigner. P. Liu and Z. Chu contribute to the writing. The final manuscript was read and approved by all authors.

**Funding** This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61871242; in part by the State Key Laboratory of ASIC & System under Grant 2021KF008; and in part by the Scientific Research Project of the Department of Education of Zhejiang under Grant Y202045495.

**Availability of data and materials** All data generated or analyzed during this study are included in this published article.

## Declarations

**Conflict of Interests** The authors declare that they have no competing interests.

## References

1. Khouri, K.S., Jha, N.K.: Leakage power analysis and reduction during behavioral synthesis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **10**(6), 876–885 (2002)
2. Chiang, M.H., Kim, K., Chuang, C.T., Tretz, C.: High-density reduced-stack logic circuit techniques using independent-gate controlled double-gate devices. *IEEE Transactions on Electron Devices* **53**(9), 2370–2377 (2006)
3. Heydari, M., Xiaohu, Z., Lai, K.K., Afro, S.: A cost-aware efficient ram structure based on quantum-dot cellular automata nanotechnology. *Int. J. Theor. Phys.* **58**(12), 3961–3972 (2019)
4. Lent, C.S., Tougaw, P.D., Porod, W., Bernstein, G.H.: Quantum cellular automata. *Nanotechnology* **4**(1), 49 (1993)
5. Timler, J., Lent, C.S.: Power gain and dissipation in quantum-dot cellular automata. *Journal of Applied Physics* **91**(2), 823–831 (2002)
6. Nejad, S.M., Kakhki, F.A., Rahimi, E.: A simple mathematical model for clocked QCA cells. In: *IEEE 2010 7th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP 2010)*, pp. 351–354 (2010)
7. Chaudhary, A., Chen, D.Z., Hu, X.S., Niemier, M.T., Ravichandran, R., Whitton, K.: Fabricatable interconnect and molecular QCA circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **26**(11), 1978–1991 (2007)
8. Taherifard, M., Fathy, M.: Improving logic function synthesis, through wire crossing reduction in quantum-dot cellular automata layout. *IET Circuits, Devices & Systems* **9**(4), 265–274 (2015)
9. Chaudhary, A., Chen, D.Z., Hu, X.S., Niemier, M.T., Ravichandran, R., Whitton, K.: Fabricatable interconnect and molecular QCA circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **26**(11), 1978–1991 (2007)
10. Abedi, D., Jaberipur, G., Sangsefidi, M.: Coplanar full adder in quantum-dot cellular automata via clock-zone-based crossover. *IEEE Trans. Nanotechnol.* **14**(3), 497–504 (2015)
11. Tahmasebi, M., Mirzaee, R.F., Komleh, S.H.P.: On the design methodology of boolean functions with quantum-dot cellular automata for reducing delay and number of wire crossings. *J. Comput. Electron.* **17**(4), 1756–1770 (2018)
12. Liu, W., Swartzlander, E.E., O’neill, M.: *Design of Semiconductor QCA systems*. Artech House (2013)
13. Snider, G., Orlov, A., Amlani, I., Bernstein, G., Lent, C., Merz, J., Porod, W.: Quantum-dot cellular automata: line and majority logic gate. *Jpn. J. Appl. Phys.* **38**(12S), 7227–7229 (1999)
14. Mitic, M., Cassidy, M.C., Petersson, K.D.: Demonstration of a silicon-based quantum cellular automata cell. *Appl. Phys. Lett.* **89**(1), 013503 (2006)
15. Wolkow, R.A., Livadaru, L., Pitters, J.: Silicon atomic quantum dots enable beyond-CMOS electronics. In: *Field-Coupled Nanocomputing*, pp. 33–58. Springer, Berlin (2014)
16. Lieberman, M., Chellamma, S., Varughese, B.: Quantum-dot cellular automata at a molecular scale. *Ann. N. Y. Acad. Sci.* **960**(1), 225–239 (2002)
17. Bernstein, G.H., Imre, A., Metlushko, V.: Magnetic QCA systems. *Microelectron. J.* **36**(7), 619–624 (2005)
18. Hu, X.K., Dey, H., Liebing, N.: Edge-mode resonance-assisted switching of nanomagnet logic elements. *IEEE Trans. Magn.* **51**(11), 1–4 (2015)
19. Lent, C.S., Henderson, K.W.: Molecular cellular networks: a non von Neumann architecture for molecular electronics. In: *IEEE 2016 International Conference on Rebooting Computing*, pp. 1–7 (2016)
20. Bohloul, S., Shi, Q., Wolkow, R.A.: Quantum transport in gated dangling-bond atomic wires. *Nano Lett.* **17**(1), 322–327 (2017)
21. Amlani, I., Orlov, A.O., Toth, G., Bernstein, G.H., Lent, C.S., Snider, G.L.: Digital logic gate using quantum-dot cellular automata. *Science* **284**(5412), 289–291 (1999)
22. Liu, W., Lu, L., O’Neill, M., Swartzlander, E.E.: A first step toward cost functions for quantum-dot cellular automata designs. *IEEE Trans. Nanotechnol.* **13**(3), 476–487 (2014)

23. Rajeswari, D., Paul, K., Balakrishnan, M.: Clocking-based coplanar wire crossing scheme for QCA. In: IEEE 2010 23rd International Conference on VLSI Design, pp. 339–344 (2010)
24. Bhanja, S., Ottavi, M., Lombardi, F., Pontarelli, S.: QCA circuits for robust coplanar crossing. *J. Electron. Test.* **23**(2), 193–210 (2007)
25. Labrado, C., Thapliyal, H.: Design of adder and subtractor circuits in majority logic-based field-coupled QCA nanocomputing. *Electron. Lett.* **52**(6), 464–466 (2016)
26. Dysart, T.J., Kogge, P.M.: Probabilistic analysis of a molecular quantum-dot cellular automata adder. In: IEEE 2007 22nd International Symposium on Defect and Fault-Tolerance in VLSI Systems, pp. 478–486 (2007)
27. Angizi, S., Sarmadi, S., Sayedsalehi, S., Navi, K.: Design and evaluation of new majority gate-based RAM cell in quantum-dot cellular automata. *Microelectron. J.* **46**(1), 43–51 (2015)
28. Kassa, S.R., Nagaria, R.: A novel design of quantum dot cellular automata 5-input majority gate with some physical proofs. *J. Comput. Electron.* **15**(1), 324–334 (2016)
29. Sasamal, T.N., Singh, A.K., Mohan, A.: An optimal design of full adder based on 5-input majority gate in coplanar quantum-dot cellular automata. *Optik* **127**(20), 8576–8591 (2016)
30. Sasamal, T.N., Singh, A.K., Mohan, A.: An efficient design of quantum-dot cellular automata based 5-input majority gate with power analysis. *Microprocess. Microsyst.* **59**, 103–117 (2018)
31. Majeed, A.H., AlKaldy, E., Zainal, M., Nor, D.B.: A new 5-input majority gate without adjacent inputs crosstalk effect in QCA technology. *Indonesian Journal of Electrical Engineering and Computer Science* **14**(3), 1159–1164 (2019)
32. Deng, F., Xie, G., Wang, S., Cheng, X., Zhang, Y.: An ultra-low-power five-input majority gate in quantum-dot cellular automata. *Journal of Circuits, Systems and Computers* **29**(11), 2050176 (2020)
33. Soeken, M., Amaru, L.G., Gaillardon, P.E., De Micheli, G.: Exact synthesis of majority-inverter graphs and its applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **36**(11), 1842–1855 (2017)
34. Chu, Z., Haaswijk, W., Soeken, M., Xia, Y., Wang, L., De Micheli, G.: Exact synthesis of boolean functions in majority-of-five forms. In: IEEE 2019 International Symposium on Circuits and Systems (ISCAS), pp. 1–5 (2019)
35. Háleček, I., Fišer, P., Schmidt, J.: Are XORs in logic synthesis really necessary? In: 2017 IEEE 20th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), pp. 134–139 (2017)
36. Ahmad, F., Bhat, G.M., Khademolhosseini, H., Azimi, S., Angizi, S., Navi, K.: Towards single layer quantum-dot cellular automata adders based on explicit interaction of cells. *J. Comput. Sci.* **16**, 8–15 (2016)
37. Bahar, A.N., Waheed, S., Hossain, N., Asaduzzaman, M.: A novel 3-input XOR function implementation in quantum dot-cellular automata with energy dissipation analysis. *Alexandria Engineering Journal* **57**(2), 729–738 (2018)
38. Balali, M., Rezai, A., Balali, H., Rabiei, F., Emadi, S.: Towards coplanar quantum-dot cellular automata adders based on efficient three-input XOR gate. *Results in Physics* **7**, 1389–1395 (2017)
39. Kalpana, K., Paulchamy, B., Chinnapparaj, S., Mahendran, K., AbdulHayum, A.: A novel design of nano scale tieo based single layer full adder and full subtractor in QCA paradigm. In: IEEE 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 575–582 (2021)
40. Chung, W.J., Smith, B., Lim, S.K.: Node duplication and routing algorithms for quantum-dot cellular automata circuits. *IEEE Proceedings-Circuits Devices and Systems* **153**(5), 497–505 (2006)
41. Smith, B.S., Lim, S.K.: QCA channel routing with wire crossing minimization. In: Proceedings of the 15th ACM Great Lakes Symposium on VLSI, pp. 217–220 (2005)
42. Bubna, M., Roy, S., Shenoy, N., Mazumdar, S.: A layout-aware physical design method for constructing feasible QCA circuits. In: Proceedings of the 18th ACM Great Lakes Symposium on VLSI, pp. 243–248 (2008)
43. Nath, R.K., Sen, B., Sikdar, B.K.: Optimal synthesis of QCA logic circuit eliminating wire-crossings. *IET Circuits, Devices & Systems* **11**(3), 201–208 (2017)
44. Sen, B., Sengupta, A., Dalui, M., Sikdar, B.K.: Design of testable universal logic gate targeting minimum wire-crossings in QCA logic circuit. In: IEEE 2010 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, pp. 613–620 (2010)
45. Sen, B., Dalui, M., Sikdar, B.K.: Introducing universal QCA logic gate for synthesizing symmetric functions with minimum wire-crossings. In: Proceedings of the International Conference and Workshop on Emerging Trends in Technology, pp. 828–833 (2010)
46. Zhang, R., Walus, K., Wang, W., Jullien, G.A.: A method of majority logic reduction for quantum cellular automata. *IEEE Trans. Nanotechnol.* **3**(4), 443–450 (2004)

47. Benini, L., De Micheli, G.: A survey of boolean matching techniques for library binding. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* **2**(3), 193–226 (1997)
48. Peres, A.: Reversible logic and quantum computers. *Phys. Rev. A* **32**(6), 3266 (1985)
49. Walus, K., Dysart, T.J., Jullien, G.A., Budiman, R.A.: QCADEsigner: a rapid design and simulation tool for quantum-dot cellular automata. *IEEE Trans. Nanotechnol.* **3**(1), 26–31 (2004)
50. Tian, H.M., Chu, Z.F.: Inversion Optimization Strategy based on Primitives with Complement Attribute. *J. Comput. Sci. Technol.* **36**(5), 1145–1154 (2021)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.