

Chapter 2

QCA Terminology

In this chapter, we present key aspects of the QCA technology. In particular, we discuss the components of QCA. We also list the logic primitives in the QCA technology and indicate the importance of *clocking* in the QCA paradigm.

2.1 QCA Basics

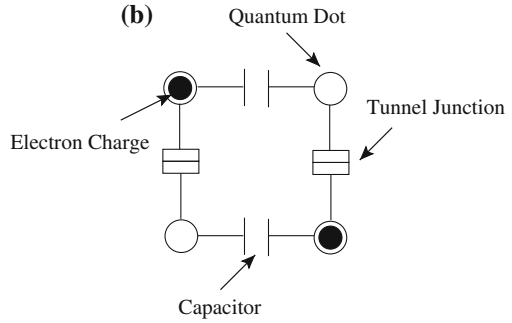
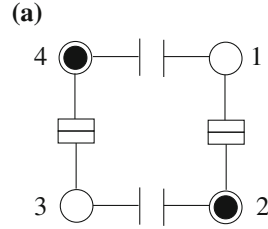
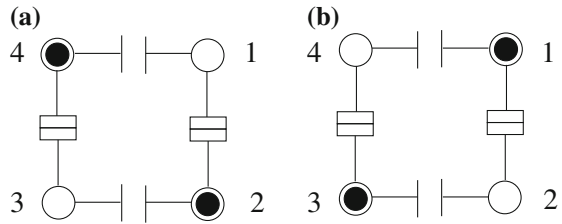
2.1.1 Cell, Quantum Dot and Tunnel Junctions

The basic element in QCA is a cell. Each cell represents a bit through an appropriate configuration of charge. A cell is shown in Fig. 2.1a. It consists of four metal islands (conductors) that constitute what are known as *quantum dots*. Each quantum dot corresponds to a semiconductor nanostructure that exhibits quantum mechanical properties in view of its size. Four dots form a QCA cell with one electron each in two of the four dots occupying “diametrically opposite” locations. Electrons occupy opposite corner (diagonal) quantum-dots because Coulomb repulsion is less compared to the scenario when they are in adjacent quantum dots. Dots are coupled by tunnel junctions. The cellular automata (CA) notion is due to the fact that the state of a given cell at a particular time depends on the state of its neighbors during the previous clock cycle.

Electrons can tunnel between the dots (on application of an appropriate potential). However, electrons are unable to leave the cell. The tunneling happens through *tunnel junctions* shown in Fig. 2.1b.

2.1.2 How Are Different Logic States Obtained?

In a QCA cell, two ground state polarizations (labelled -1 and 1) are possible depending upon the position of (electron) charge. The polarization is calculated via Eq. (2.1).

Fig. 2.1 Basic QCA cell**Fig. 2.2** Representation of logic states in QCA, **a** logic-0, **b** logic-1

$$P = \frac{(p_1 + p_3) - (p_1 + p_3)}{p_1 + p_3 + p_1 + p_3} \quad (2.1)$$

Equation (2.1) shows the calculation of polarization of a QCA cell. Here p_i is the charge of the i th quantum-dot ($p_i = 1$ if an electron is present, otherwise it is 0). These polarizations are represented by logic-0 and logic-1 (in other words, binary 0 and binary 1) as shown in Fig. 2.2.

One can change the logic state of a cell by merely applying a negative potential to a lead near the quantum dot occupied by an electron. This causes the next cell to change the ground states to reduce Coulomb repulsion. Figure 2.3a, b show the propagation of logic-1 and logic-0 respectively from the left end to the right due to Coulomb repulsion. Therefore, in QCA, there is (at most) one position change of electrons in quantum dots (within a cell) to transfer information from one end to the other, whereas in traditional CMOS, there is explicit transfer of charge (from one end to the other) for propagation of information. The arrangement of QCA cells in

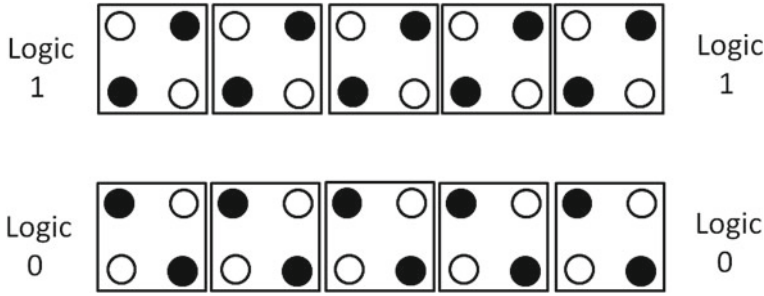


Fig. 2.3 Operation of a QCA wire. **a** QCA wire propagating logic 1. **b** QCA wire propagating logic 0

Fig. 2.3 shows a QCA wire. By application of logic 0 or logic 1 to a QCA cell (at the input), we propagate information to the output ‘side’ taking advantage of the force of repulsion.

2.2 Logic Primitives in QCA

In the previous section, we discussed about operation of a QCA cell and QCA wire. Here we discuss the elements of interest in digital design, namely logic gates. By arranging QCA cells as shown in Fig. 2.4a, b, we can implement an inverter (or a NOT gate) and a majority gate respectively. Figure 2.4a shows that upon applying input “0”, we get output of “1” which amounts to flipping of the input. This corresponds to realization of an inverter. The output of the gate shown in Fig. 2.4b is the majority of its inputs: for example in Fig. 2.4b, “001” is applied to obtain an output of “0” which

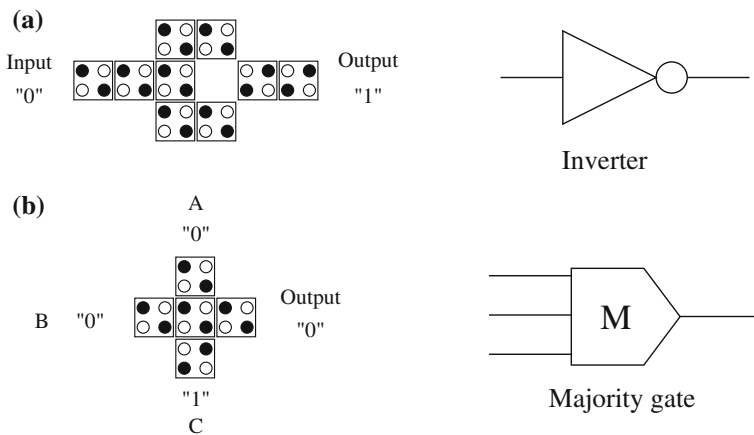


Fig. 2.4 Basic gates in QCA. **a** Inverter. **b** Majority gate $M(A, B, C) = AB + BC + AC$

represents the majority of the inputs. Suppose A , B and C are inputs to the majority gate, then the output is given by $M(A, B, C) = AB + BC + AC$. By setting one input (say C) to 0, the majority gate acts as an AND gate. In other words, $M(A, B, 0) = AB$. Similarly, by setting one of the three inputs to 1, the majority gate functions as an OR gate ($M(A, B, 1) = A + B$). A majority gate has advantages compared to an AND or OR gate, since three inputs enable realization of certain arithmetic functions (such as a carry in a one-bit full-adder) with just one majority gate. Further, a majority gate requires only the same number of cells in QCA as an AND gate and OR gate. This can be contrasted with MOSFET technology (where the number of transistors required to realize a 3-input majority gate is much more than that required for some other gates for example). Note that a majority gate along with an inverter is a universal set and can be used to realize any Boolean function just like NAND or NOR. Further, we will observe that a rich set of properties for majority gates (described in the subsequent chapters) enables very low-complexity QCA design of various arithmetic circuits.

Remark 2.1 In principle, the majority gate permits an odd number of inputs. However, for a 2-D QCA cell, we can have only three inputs (and 1 output) given that there are only four ‘directions’. Extensions to 3-D would permit additional inputs. Further, for arithmetic, three inputs seem to be appropriate. For example, the carry function in a one-bit full-adder takes three inputs.

2.2.1 Clocking in QCA-Role and Types

Several approaches have been suggested for computation with an array of QCA cells. One approach is based on transferring the array to an excited state from a ground state by merely applying input data (without explicit clocking). The array is expected to settle to a new ground state. However, sometimes the transition may result in a metastable intermediate state. To facilitate transfer to a new ground state, another approach based on clocking has been suggested [1]. Clocking (by application of an appropriate voltage to a cell) leads to adjustment of tunneling barriers between quantum dots for transfer of electrons between the dots. Figure 2.5 depicts clocking of a QCA cell. Here V_c corresponds to the clock voltage. Clocking is performed in one of two ways: zone clocking and continuous clocking. In zone clocking [2], each QCA cell is clocked using a four-phase clocking scheme as shown in Fig. 2.6. The four phases correspond to *switch*, *hold*, *release* and *relax*. In the switch phase, cells begin unpolarized and with low potential barriers but the barriers are raised during this phase. In the hold phase, the barriers are held high while in the release phase, the barriers are lowered. In the last phase, namely relax, the barriers remain lowered and keep the cells in an unpolarized state. An alternative to zone clocking, called continuous clocking, involves generation of a potential field by a system of submerged electrodes. The former clocking scheme is adopted in this book since the CAD tool used for simulation supports zone clocking and further, prior works on adders are only based on this clocking scheme. Figure 2.7 shows the interdot barriers

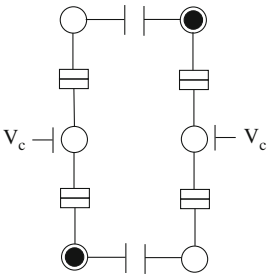


Fig. 2.5 QCA cell with clocking

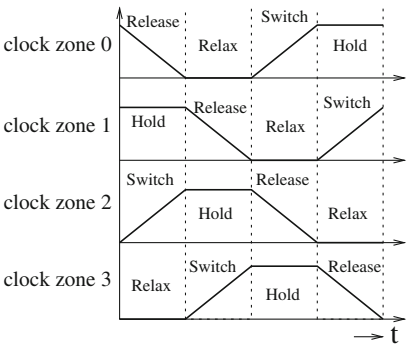


Fig. 2.6 QCA clock zones

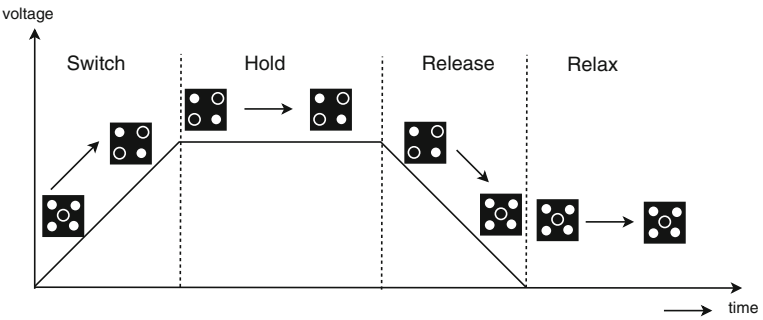


Fig. 2.7 Interdot barriers in a clocking zone

in a clocking zone while Fig. 2.8 shows the operation of a wire in different clock zones.

A key difference between circuit design in QCA and that in conventional CMOS technology is that a circuit in QCA has no control over the clocks [3] unlike in CMOS. Hence, information is merely *transmitted* through each cell and not retained. Each cell “erases” its own state through every clock cycle. Further, every logic element in a QCA circuit is clocked.

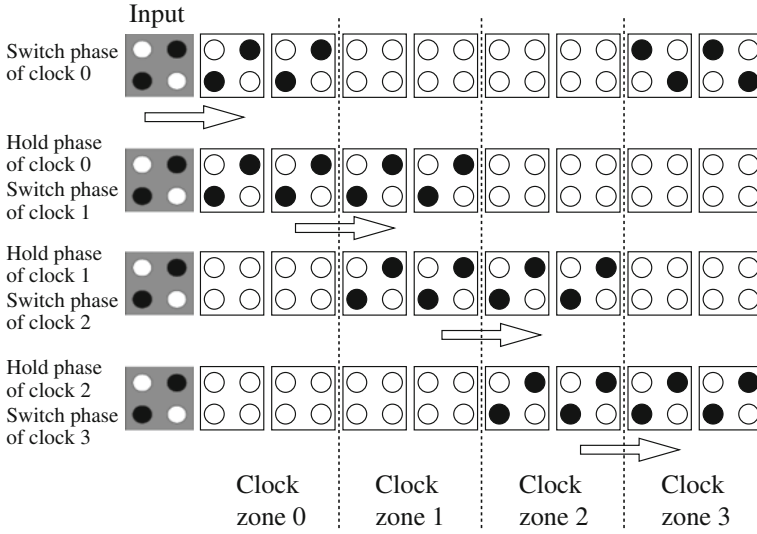


Fig. 2.8 Operation of a wire in different clock zones

2.2.2 Crossovers in QCA

A QCA design permits two options for crossover, termed coplanar crossover and multilayer crossover. While the coplanar crossover uses only one layer but involves usage of two cell types (termed regular and rotated), the multilayer crossover uses more than one layer of cells (analogous to multiple metal layers in a conventional IC). Multilayer crossover is used in this book for wire crossings since we can effectively cross signals over on another layer and the extra layers of QCA can be used as active components of the circuit [2]. Further, multilayer QCA circuits can potentially consume much less area as compared to planar circuits [2].

2.3 Tool for QCA Simulation

In this book, a tool for designing layouts and performing simulations called QCADesigner [3] is used. The tool allows us to describe different components of a digital design. The tool also permits specification of clock zones for a design. Four clocking zones in QCA are represented by four different colours in QCADesigner. In particular, clock zone 0 is represented in green, clock zone 1 in magenta, clock zone 2 in blue and clock zone 3 in white. When cells are first input to a design, they are in clock zone 0. To apply a different clock (zone) to a cell or an array of cells, we need to select the cells and specify the clock. A snapshot of QCADesigner (with an inverter on the left and a majority gate on the right) is shown in Fig. 2.9.

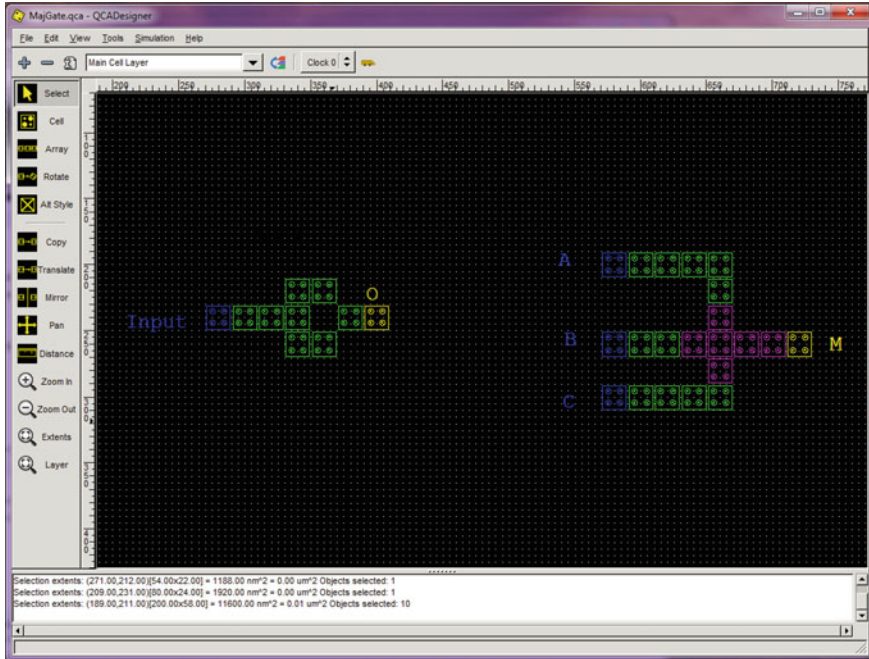


Fig. 2.9 Illustration for QCADesigner

2.4 Summary

In this chapter, we have presented QCA terminology that will be used in the remainder of this book. In the next chapter, we present design of several basic logic elements in QCA.

References

1. C.S. Lent, P.D. Tougaw, A device architecture for computing with quantum dots. *Proc. IEEE* **85**(4), 541–557 (1997)
2. K. Walus, G.A. Jullien, Design tools for an emerging SoC technology: quantum-dot cellular automata. *Proc. IEEE* **94**(6), 1225–1244 (2006)
3. K. Walus, T. Dysart, G. Jullien, R. Budiman, QCADesigner: a rapid design and simulation tool for quantum-dot cellular automata. *IEEE Trans. Nanotechnol.* **3**(1), 26–29 (2004)

Design of Arithmetic Circuits in Quantum Dot Cellular
Automata Nanotechnology

Sridharan, K.; Pudi, V.

2015, XV, 116 p. 100 illus., 66 illus. in color., Hardcover

ISBN: 978-3-319-16687-2