

# Novel Highly Parallel and Systolic Architectures using Quantum Dot-Based Hardware

Amir Fijany, Benny N. Toomarian, and Matthew Spotnitz

*Jet Propulsion Laboratory, California Institute of Technology*  
4800 Oak Grove Drive, Pasadena, CA 91109

## Abstract

In this paper, we present novel parallel architectures based on Quantum-dot Cellular Automata (QCA) hardware. We show that the QCA, by allowing co-planar line crossing, overcomes a major limitation of VLSI and, in this sense, it can potentially open a new direction in the design of parallel algorithms and architectures. In addition, the QCA is inherently suitable for pipeline and systolic computation. Exploiting these two unique features of QCA and as representative applications, we present systolic architectures for co-planar implementation of complex permutation matrices and computation of FFT by using QCA-based hardware.

## 1 Introduction

VLSI technology has made possible the integration of massive number of components (processors, memory, etc.) into a single chip. In VLSI design, memory and processing power are relatively cheap and the main emphasis of the design is on reducing the overall interconnection complexity since data routing costs dominate the power, time, and area required to implement a computation. Communication is costly because wires occupy the most space on a circuit and it can also degrade clock time [1]. In fact, much of the complexity (and hence the cost) of VLSI design results from minimization of data routing. The main difficulty in VLSI routing is due to the fact that crossing of the lines carrying data, instruction, control, etc. is not possible in a plane. Thus, in order to meet this constraint, the VLSI design aims at keeping the architecture highly regular with local and short interconnection.

Systolic arrays [1,2] are perhaps the best representative example of strengths and limitations of VLSI. In fact, systolic arrays were originally devised as a novel paradigm for massively parallel computation to take advantage of and conform to the features of VLSI. Systolic arrays exploit massive parallelism in the computation by integrating a large number of simple processor elements interconnected with simple, recursive, and regular pattern. However, due to the inherent limitation of VLSI, many applications of interest are not amenable to systolic processing. There are two types of algorithms: the *local communication* type and the *global communication* type [1]. A large class of algorithms for signal/image processing, matrix operations, etc., belong to the class of local communication type. These algorithms can be classified based on their *planar graph*, that is, their graph can be mapped to another topologically equivalent graph with no *crossover* of wires. As a result, they require only

local interconnection between the elements of the computing array. Such algorithms are highly suitable for systolic processing and consequently various systolic arrays have been proposed for their implementation [1,3]. However, a very important class of algorithms, the so called Fast Transforms including FFT, Hartley and Cosine Transforms, etc., are of global communication type, that is, they require global interconnection between the elements of the computing array and hence they cannot be mapped to another topologically equivalent graph with no crossover. Consequently, there has not been any proposal for systolic computation of this class of problems. In fact, this class of problem is considered as not suitable for systolic processing [1,3].

*It should be emphasized that the main obstacle in a systolic (and highly parallel) computation of this class of problems is the need for complex data permutations that arise in their implementation.* In fact, the so called Fast Transforms achieve their efficiency by exploiting the structure of their underlying operators through the use of various permutation matrices. In a sequential implementation, the permutation matrices are implemented in an implicit fashion, that is, their effect on a given vector is implemented by performing the corresponding permutation of the elements of the vector. Therefore, such an implementation only involves data movement. However, even in a sequential environment and depending on the underlying complexity of a given permutation matrix, this implementation might represent a significant part of the overall computation time. A salient example is the so called Bit-Reversal Permutation that arises in FFT (see for example [4]). In a parallel computation, the implementation of permutation matrices is considerably more complex since the elements of the target vector are distributed among a set of processor/memory modules. And, in fact, depending on the underlying complexity of a permutation matrix and the interconnection network among the processor/memory modules, the parallel implementation of the data permutation may represent the most intensive part of the overall parallel computation. As a result, the complexity of parallel implementation of various data permutations is a major obstacle in exploiting a high degree of parallelism in computation of the Fast Transforms.

There has been significant improvement in the performance (size, power consumption, and speed) of VLSI devices in recent years and this trend may also continue for some near future. However, it is a well known fact that there are major obstacles, i.e., physical limitation of feature size reduction and ever increasing cost of foundry, that would prevent the long term continuation of this trend. This has motivated the exploration of some fundamentally new technologies that are not dependent on the conventional feature size approach. Such technologies are expected to enable scaling to continue to the ultimate level, i.e., molecular and atomistic size. In particular, quantum dot-based computing by using Quantum-dot Cellular Automata (QCA) has recently been intensely investigated [8-13] as a promising new technology capable of offering significant improvement over conventional VLSI in terms of reduction of feature size (and hence increase in integration level), reduction of power consumption, and increase of switching speed.

However, we strongly believe that the main advantage of QCA over VLSI is not in offering quantitative (and though significant) improvement in performance, i.e., feature size, integration, and power consumption. Rather, QCA offers a unique capability which overcomes the

major limitation of VLSI, i.e., the data routing constraint. In fact, due to their cellular nature, it is possible to cross QCA wires in a plane (see §2). Such a capability then allows compact implementation of complex interconnection networks in a plane by using QCA wires, which has not been possible in VLSI. In this sense, QCA opens a new direction in designing novel and highly parallel algorithms and architectures. Note that, the communication requirement has been considered as the key factor in evaluating practical efficiency of parallel algorithms. In fact, many known efficient (in terms of computational complexity) parallel algorithms are not suitable for practical implementation on available parallel architectures, due to their complex communication requirements. In parallel computing, communication is a key factor because implementation of arbitrary and complex interconnection among a large number of processors is either impossible or very expensive. QCA, by offering the possibility of implementing compact and complex interconnection patterns, can potentially provide a *paradigm shift* in analysis and design of parallel algorithms and architectures. One major requirement for (and to some degree a limitation of) QCA is the need for an appropriate clocking mechanism (see §2). However, this requirement makes QCA inherently suitable for pipeline and systolic computation.

In this paper, in order to show the potential of QCA for designing novel parallel algorithms and architectures, we propose a hybrid VLSI/QCA architecture for systolic computation of FFT as a representative application. We first present QCA circuits for a *direct* hardware implementation of two fundamental permutation matrices: the *Perfect Shuffle* and the *Bit Reversal* permutation matrices which arise in FFT and many other signal and image processing applications [5]. We then consider a reformulation of FFT and a hybrid architecture for its systolic implementation. The hybrid architecture considered in this paper consists of a set of VLSI and QCA modules (chips). The VLSI modules contain a set of simple bit-serial processing elements capable of performing multiply and add operations. The QCA modules implement the required interconnection between processing elements of VLSI modules.

## 2 A Brief Overview of Quantum-dot Cellular Automata

In this section, we briefly review some pertinent features of basic functioning of and computation with QCA. More detailed descriptions can be found in [8-13]. The basic computational element in QCA is a quantum-dot cell (or molecule). A QCA cell consists of four quantum dots positioned at the corner of a square (Fig. 1). The cell contains two extra mobile electrons, which are allowed to tunnel between neighboring sites. Tunneling out of the cell is assumed to be completely suppressed by the potential barriers between cells. Indeed, if the barriers between cells are sufficiently high, the electron will be well localized on individual dots. The Coulomb repulsion between the electrons will tend to make them occupy antipodal sites in the square. For an isolated cell, there are two energetically equivalent arrangements of the extra electrons which are denoted as cell state or polarization,  $P$ . The cell polarization is used to encode binary information (Fig. 1). The polarization of a non-isolated cell is determined based on interaction with neighboring cells. The interaction between cells is Coulombic and provides the basis for computing with QCA. No current flows between cells

and no power or information is delivered to individual internal cells. Local interconnection between cells are provided by the physics of cell-cell interaction [9]. Previous results have shown the feasibility of fabricating quantum dots with single charges [6] and of making large arrays of dots and controlling their occupancy [7]. The design of universal logic gates and binary wire using QCA is presented in [8-10] (see Fig. 2). The first experimental demonstration of a functioning QCA cell is presented in [11]. Experimental demonstrations of a functioning QCA binary wire and a majority gate are presented in [12,13]. More interestingly, however, QCA offers a unique capability which overcomes the major limitation of VLSI, i.e., the data routing constraint. In fact, due to their cellular nature and Coulombic interaction, it is possible to cross QCA wires in a plane (Fig. 3). We exploit such a capability of QCA (which is not possible in VLSI) for designing compact and complex interconnection networks in a plane.

A QCA array performs computation through Coulombic interaction among neighboring cells which causes them to influence each other's polarization. The computation in QCA array is *edge driven*, that is, both energy and information flow in from the edges of the array only. This also provides the directionality in the computation by the array. In this sense, the difference between input and output cells is simply that inputs are fixed while outputs are free to change [9]. The QCA array then performs the desired computation by reacting to the change in the boundary conditions, i.e., the input to the array. The fact that the computation in edge-driven implies that no contact to interior cells are made directly and thus eliminating the interconnection problem. It also implies that the paradigm involves *computing with ground state*, that is, the QCA array reacts to the input and settles to a new ground state which represents solution of the desired computational problem for which the array is specifically designed. However, computing with ground state implies that the computation is temperature sensitive. In fact, if the thermal fluctuations excite the array above its ground state then the array can produce wrong answer. There are two other major problems with computing with ground state. First, the dynamics of the array, i.e., its evolution to the ground state, is hard to control [9]. Consequently, the settling time to the ground state cannot be controlled or predicted and it would vary depending on the complexity of the array. Second, the array might settle to a *metastable* state, producing wrong result or leading to a significant delay in reaching to the true ground state.

In order to overcome these limitations of computing with ground state, an *adiabatic switching scheme* has been developed [9]. In this scheme, a QCA array is divided into subarrays and each subarray is controlled by a different clock. The proposed clock in QCA is *multi-phased*. This clocking scheme allows a given subarray to perform its computation, have its state frozen by raising of its interdot barriers, and then have its output as the input to the successor subarray. Due to the multi-phase nature of this clocking scheme, the successor subarray is kept in an unpolarized state so it does not influence the calculation of preceding subarray. Such a clocking scheme implies a pipeline computation since different subarray can perform different parts of computation. In other words, QCA arrays are inherently suitable for pipeline and moreover systolic computations. Indeed, the architectures presented in this paper exploit this feature of QCA to enable a systolic implementation of FFT.

### 3 QCA Circuits for Implementation of Perfect-Shuffle and Bit-Reversal Permutation Matrices

In this section, we present QCA circuits for implementation of the perfect shuffle permutation matrix,  $\Pi_{2^n}$ , and the Bit-Reversal permutation matrix,  $P_{2^n}$ , which arise in Fourier transforms as well as many other signal and image processing applications [9]. These circuits allow a co-planar (i.e., in a single layer), compact, and direct (i.e., hardwired) implementation of these permutation matrices. The QCA circuits in this section have been validated through extensive simulation by considering all possible combinations of the input vector. The simulations are performed by using AQUINAS (A QUantum Interconnected Network Array Simulator) which encapsulates the physics of Hartree-Fock model for simulation of QCA array and is provided by the University of Notre Dame.

#### 3.1 Perfect Shuffle Permutation Matrix

A description of permutation matrix  $\Pi_{2^n}$  can be given by describing its effect on a given vector. If  $Z$  is an  $2^n$ -dimensional vector, then the vector  $Y = \Pi_{2^n} Z$  is obtained by splitting  $Z$  in half and then shuffling the top and bottom halves of the deck. Alternatively, a description of the matrix  $\Pi_{2^n}$ , in terms of its elements  $\Pi_{2^n}(i, j)$ , for  $i$  and  $j = 0, 1, \dots, 2^n - 1$ , can be given as

$$\Pi_{2^n}(i, j) = \begin{cases} 1 & \text{if } j = i/2 \text{ and } i \text{ is even, or if } j = (i - 1)/2 + 2^{n-1} \text{ and } i \text{ is odd} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Figure 4 shows the schematic and the designed QCA circuit for implementation of  $\Pi_8$ . The circuit in Fig. 4.b has been validated through extensive simulation by considering all possible combinations of the input vector.

#### 3.2 Bit-Reversal Permutation Matrix

A description of  $P_{2^n}$  can be given by describing its effect on a given vector. If  $Z$  is an  $n$ -dimensional vector and  $Y = P_{2^n} Z$ , then  $Y_i = Z_j$ , for  $i = 0, 1, \dots, 2^n - 1$ , wherein  $j$  is obtained by reversing the bits in the binary representation of index  $i$ . Therefore, a description of the matrix  $P_n$ , in terms of its elements  $P_{2^n}(ij)$ , for  $i$  and  $j = 0, 1, \dots, 2^n - 1$ , is given as

$$P_{2^n}(ij) = \begin{cases} 1 & \text{if } j \text{ is bit reversal of } i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

A factorization of  $P_{2^n}$  in terms of  $\Pi_{2^i}$  is given as [5]

$$P_{2^n} = \Pi_{2^n}(I_2 \otimes \Pi_{2^{n-1}}) \cdots (I_{2^i} \otimes \Pi_{2^{n-i}}) \cdots (I_{2^{n-3}} \otimes \Pi_8)(I_{2^{n-2}} \otimes \Pi_4) \quad (3)$$

where  $\otimes$  indicates Kronecker Product. Figure 5 shows the schematic and the designed QCA circuit for implementation of  $P_8$ . Again, the circuit in Fig. 5.b has been validated through

extensive simulation by considering all possible combinations of the input vector. Note, however, that the implementation of  $P_{2^n}$  is significantly more complex than that of  $\Pi_{2^n}$ , due to its more complex permutation pattern. In fact, we have been unable to simulate a direct implementation of even  $P_{16}$  since its complexity has easily overwhelmed the capability of our currently available simulation tools. An alternative technique to overcome such a limitation is to decompose (factorize) the permutation matrix  $P_{2^n}$ . Indeed, Eq. (5) represents such a factorization wherein  $P_{2^n}$  can be implemented by using the circuits for implementation of the same and smaller size permutation matrix  $\Pi$ . However, we are currently investigating a more efficient technique by exploiting the structure of matrix  $P_{2^n}$  which has allowed us to design a series of simple circuits for its implementation.

## 4 A Hybrid VLSI/QCA Systolic Array for FFT

The classical Cooley-Tukey Radix-2 FFT for a  $2^n$ -dimensional vector is a sparse matrix factorization of DFT given by [5]

$$F_{2^n} = A_n A_{n-1} \cdots A_{i+1} A_i \cdots A_2 A_1 P_{2^n} = \underline{F}_{2^n} P_{2^n} \quad (4)$$

where

$$A_i = I_{2^{n-i}} \otimes B_{2^i} \quad (5)$$

$$B_{2^i} = \frac{1}{\sqrt{2}} \begin{pmatrix} I_{2^{i-1}} & \Omega_{2^{i-1}} \\ I_{2^{i-1}} & -\Omega_{2^{i-1}} \end{pmatrix} \text{ and } \Omega_{2^{i-1}} = \text{Diag}\{\omega_{2^i}^j\}, \text{ for } j = 0, 1, \dots, 2^{i-1} - 1, \text{ with } \omega_{2^i} = e^{\frac{-2i\pi}{2^i}}$$

and  $\iota = \sqrt{-1}$ . We have that  $F_2 = W = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ . The operator

$$\underline{F}_{2^n} = A_n A_{n-1} \cdots A_{i+1} A_i \cdots A_2 A_1 \quad (6)$$

represents the computational kernel of Cooley-Tukey FFT while  $P_{2^n}$  represents the bit-reversal permutation which needs to be performed on the elements of the input vector before feeding that vector into the computational kernel.

The Cooley-Tukey FFT as given by (1), though optimal for a sequential computation, is not suitable for a systolic implementation. A suitable variant for systolic implementation is developed as follows. Using the permutation matrix  $\Pi_{2^i}$ , the matrices  $B_{2^i}$  can be reduced to a block diagonal form as

$$\Pi_{2^i} B_{2^i} \Pi_{2^i}^t = R_{2^i} \text{ or } B_{2^i} = \Pi_{2^i}^t R_{2^i} \Pi_{2^i} \quad (7)$$

where t indicates transpose and  $R_{2^i}$  is a block diagonal matrix given by  $R_{2^i} = \text{Diag}\{r(\omega_{2^i}^j)\}$ , for  $j = 0, 1, \dots, 2^{i-1} - 1$ , with  $r(\omega_{2^i}^j) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & \omega_{2^i}^j \\ 1 & -\omega_{2^i}^j \end{pmatrix}$ . Using (4), the matrices  $A_i$  given by (2) can be written as

$$A_i = I_{2^{n-i}} \otimes (\Pi_{2^i}^t R_{2^i} \Pi_{2^i}) \quad (8)$$

and using the identity

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \quad (9)$$

we then have

$$A_i = (I_{2^{n-i}} \otimes \Pi_{2^i}^t)(I_{2^{n-i}} \otimes R_{2^i})(I_{2^{n-i}} \otimes \Pi_{2^i}) \quad (10)$$

Let

$$S_i = (I_{2^{n-i}} \otimes \Pi_{2^i})(I_{2^{n-i+1}} \otimes \Pi_{2^{i-1}}^t) \text{ and } K_i = I_{2^{n-i}} \otimes R_{2^i}, \text{ for } i = n, n-1, \dots, 1 \quad (11)$$

Substituting (7) and (8) into (1), we then get

$$F_{2^n} = \Pi_{2^n} S_n K_n S_{n-1} K_{n-1} \cdots S_{i+1} K_{i+1} S_i K_i \cdots S_2 K_2 S_1 K_1 P_{2^n} \quad (12)$$

A hybrid VLSI/QCA architecture for a systolic implementation of (9) is shown in Fig. (6). The terms  $\Pi_{2^n}$ ,  $S_i$ , and  $P_{2^n}$ , which represent data permutation operators, are implemented by using QCA modules. The terms  $K_i$  are implemented by using VLSI modules containing a set of simple bit-serial processing elements. Each processing element has two inputs and two outputs. It reads data from its two inputs and produces two outputs by performing simple multiply and add operations. Aside being driven by the same clock, the processing elements are totally independent from each other. Due to this feature, the processing modules are highly suitable for a large-scale implementation with CMOS VLSI. In order to achieve the global synchronization, the VLSI and the QCA modules are driven by the same clock.

## 5 Conclusion

In this paper, we presented novel circuits for a compact, co-planar, and direct implementation of two fundamental permutation matrices by using QCA-based hardware. Using these circuits, we then presented a hybrid VLSI/QCA architecture for systolic computation of FFT. The architecture of this paper underlines the unique advantage of QCA. Although QCA offers significant quantitative advantages over CMOS VLSI, in terms of feature size (and hence integration level), switching speed, and power consumption, we strongly believe that the unique advantage of QCA is the capability of co-planar line crossing. This capability can potentially overcome a major limitation of VLSI, i.e., the data routing constraint. It can also open a new direction in designing massively parallel algorithms and architectures. As it was shown for FFT, it can enable the design of novel systolic arrays for applications which have previously been considered not amenable to a systolic computation by VLSI.

However, It should be emphasized that much more work remains to be done in developing a more systematic approach for the design of QCA-based circuits. Note that, in this paper and for the sake of proof of concept, we considered bit-serial circuits for computation and communication. The bit-serial computation fully exploits the pipeline nature of QCA (through deep pipelining at the bit level). However, many applications of interest might require a bit-parallel format resulting in more complex circuits which cannot be simulated and analyzed by the currently available simulators.

## Acknowledgment

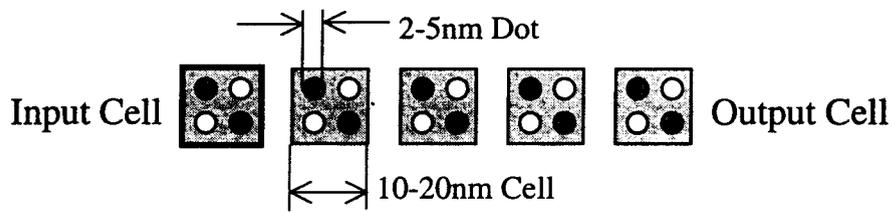
The research described in this paper was performed at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under contract with National Aeronautics and Space Administration (NASA). This work was supported by the Revolutionary Computing Technologies Program Element of the JPL Center for Integrated Space Microsystems (CISM). The authors gratefully acknowledge Professors C.S. Lent and P.M. Kogge from Notre Dame University for insightful discussions as well as for providing us with AQUINAS.

## References

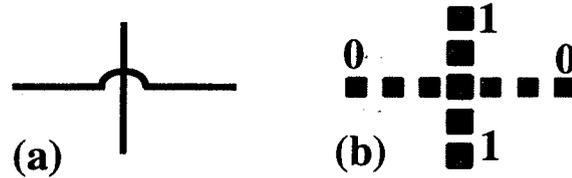
- [1] S.Y. Kung, *VLSI Array Processors*, Prentice Hall, 1988.
- [2] H.T. Kung, "Why systolic architectures?," *Computer*, vol. 15, p. 37, 1982.
- [3] D.I. Moldovan, *Parallel Processing: From Applications to Systems*. Morgan Kaufmann Publishers, 1993.
- [4] A.H. Karp, "Bit Reversal on Uniprocessor", *SIAM Review*, vol. 18(1), pp. 1-26, 1996.
- [5] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*. SIAM Publications, Philadelphia, 1992.
- [6] R.C. Ashoori, H.L. Stormer, J.S. Weiner, L.N. Pfeiffer, K.W. Baldwin, and K.W. West, "N-electron ground state energies of a quantum dot in a magnetic field," *Phys. Rev. Lett.*, vol. 71, p. 613, 1993.
- [7] B. Meurer, D. Heitmann, and K. Ploog, "Single electron charging of quantum-dot atoms," *Phys. Rev. Lett.*, vol. 68, p. 1371, 1999.
- [8] P.D. Tougaw and C.S. Lent, "Logical device implementation using quantum cellular automata", *J. Applied Physics*, 75, p. 1818, 1994.
- [9] C.S. Lent and P.D. Tougaw, "A device architecture for computing with quantum dots", *Proc. IEEE*, vol. 85(4), 1997.
- [10] C.S. Lent and P.D. Tougaw, "Line of interacting quantum-dot cells: a binary wire", *J. Applied Physics*, vol. 74, p. 6227, 1993.
- [11] I. Amlani, A.O. Orlov, G.H. Bernstein, C.S. Lent, and G.L. Snider, "Realization of a Functional Cell for Quantum-Dot Cellular Automata", *Science.*, vol. 277, p. 289, 1997.
- [12] A.O. Orlov, I. Amlani, G. Toth, C.S. Lent, G.H. Bernstein, and G.L. Snider, "Experimental demonstration of a binary wire for quantum-dot cellular automata", *Applied Physics Lett.*, vol. 74, p. 928, 1997.
- [13] I. Amlani, A.O. Orlov, G. Toth, C.S. Lent, G.H. Bernstein, and G.L. Snider, "Digital logic gate using quantum-dot cellular automata", *Applied Physics Lett.*, vol. 74, p. 2875, 1999.



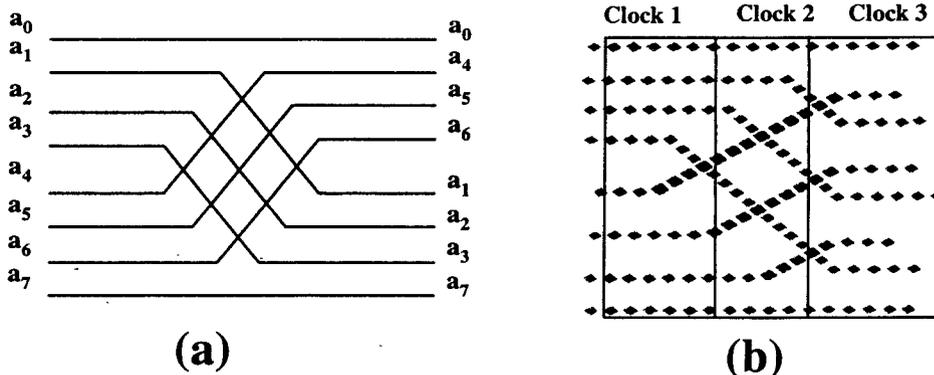
**Figure 1. Cell Polarization and Binary Information Encoding**



**Figure 2. Binary Wire and Cell-Cell Interaction**



**Figure 3. Co-planar Wire Crossing**  
a) Schematic      b) QCA Circuit



**Figure 4. Implementation of Permutation Matrix  $\Pi_8$**   
a) Schematic      b) QCA Circuit

