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

Extended Abstract

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. As a result, while the high level of integration has opened the way for massively parallel computation, practical and full exploitation of such a capability in many applications of interest has been hindered by the constraints on interconnection pattern. More precisely, the use of only localized communication significantly simplifies the design of interconnection architecture but at the expense of somewhat restricted class of applications [1]. For example, there are currently commercially available products integrating hundreds of simple processor elements within a single chip. However, the lack of adequate interconnection pattern among these processing elements make them inefficient for exploiting a large degree of parallelism in many applications.

Systolic arrays [1,2] were 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, e.g., FFT, Fast Hartley and Cosine Transforms, etc., are of global communication type, i.e., they require global interconnection between the elements of the computing array and hence they cannot be mapped to another topologically 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].

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. Quantum computing, quantum dot-based computing, DNA based computing, biologically inspired computing, optical computing are examples of such new technologies. In particular, quantum dot-based computing by using Quantum-dot Cellular Automata (QCA) has recently been intensely investigated [4-8] 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.

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 (for a more detailed description see, e.g., [6,7]). 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 polarization, P. The cell polarization is used to encode binary information. 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 [7]. Previous results have shown the feasibility of fabricating quantum dots with single charges [4] and of making large arrays of dots and controlling their occupancy [5]. The design of universal logic gates and binary wire using QCA is also presented in [6-8] (see Fig. 2).

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 (Fig. 3). 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, traditionally, 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 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. For example, as discussed before, the locality and simplicity of interconnection pattern is a major requirement and constraint in the design of systolic arrays. QCA, by offering the possibility of implementing compact and complex interconnection patterns, can potentially provides a *paradigm shift* in analysis and design of parallel algorithms and architectures.

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. As discussed before, systolic computation of FFT by using VLSI has been considered impractical due to its global and complex interconnection requirements. 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 processing elements are driven by the same clock. Each processing element has its own input and output. The QCA modules implement the required interconnection between processing elements of VLSI modules (Fig. 3).

We first consider the design of QCA circuits for a *direct* hardware implementation of three fundamental permutations matrices: the *Downshift* permutation matrix, Q_{2^n}, the *Perfect Shuffle* permutation matrix, Π_{2^n}, and the *Bit Reversal* permutation matrix, P_{2^n}, acting on a 2^n-dimensional vector. These permutation matrices arise in Fourier transforms as well as many other signal and image processing applications [9]. We present detailed design of circuit and its validation through simulation for implementation of these permutations by using QCA. We then consider a reformulation of FFT for its systolic implementation.

The classical Cooley-Tukey Radix-2 FFT for a 2^n-dimensional vector is a sparse matrix factorization of DFT given by [9]

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

where

$$A_i = I_{2^{n-i}} \otimes B_{2^i} \tag{2}$$

($\otimes$ indicates Kronecker Product), $B_{2^i} = \frac{1}{\sqrt{2}} \begin{pmatrix} I_{2^{i-1}} & \Omega_{2^{i-1}} \\ I_{2^{i-1}} & -\Omega_{2^{i-1}} \end{pmatrix}$ and $\Omega_{2^{i-1}} = \mathrm{Diag}\{\omega_{2^i}^j\}$, for $j = 0, 1, \cdots 2^{i-1} - 1$, with $\omega_{2^i} = e^{\frac{-2\iota\pi}{2^i}}$ and $\iota = \sqrt{-1}$. We have that $F_2 = W = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. The operator

$$\underline{F}_{2^n} = A_n A_{n-1} \cdots A_{i+1} A_i \cdots A_2 A_1 \tag{3}$$

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 Π_{2^i}, the matrices B_{2^i} can be reduced to a block diagonal form as

$$\Pi_{2^i} B_{2^i} \Pi_{2^i}^t = R_{2^i} \text{ or } B_{2^i} = \Pi_{2^i}^t R_{2^i} \Pi_{2^i} \tag{4}$$

where t indicates transpose and R_{2^i} is a block diagonal matrix given by $R_{2^i} = \mathrm{Diag}\{r(\omega_{2^i}^j)\}$, for $j = 0, 1, \cdots 2^{i-1} - 1$, 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

$$A_i = I_{2^{n-i}} \otimes (\Pi_{2^i}^t R_{2^i} \Pi_{2^i}) \tag{5}$$

and using the identity

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{6}$$

we then have

$$A_i = (I_{2^{n-i}} \otimes \Pi^t_{2^i})(I_{2^{n-i}} \otimes R_{2^i})(I_{2^{n-i}} \otimes \Pi_{2^i}) \tag{7}$$

Let

$$S_i = (I_{2^{n-i}} \otimes \Pi_{2^i})(I_{2^{n-i+1}} \otimes \Pi^t_{2^{i-1}}) \text{ and } K_i = I_{2^{n-i}} \otimes R_{2^i}, \text{ for } i = n, n-1, \cdots i \tag{8}$$

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

$$F_{2^n} = \Pi_{2^n} S_n K_n S_{n-1} K_{n-1} \cdots S_{i+1} K_{i+1} S_i K_i \cdots S_2 K_2 S_1 K_1 P_{2^n} \tag{9}$$

The systolic architecture for implementation of (9) is shown in Fig. (4) where the terms Π_{2^n}, S_i, and P_{2^n} are implemented by using QCA modules and the terms K_i are implemented by using VLSI modules containing a set of bit-serial processing elements.

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] 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.

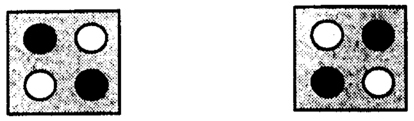
[5] B. Meurer, D. Heitmann, and K. Ploog, "Single electron charging of quantum-dot atoms," *Phys. Rev. Lett.*, vol. 68, p. 1371, 199.

[6] P.D. Tougaw and C.S. Lent,"Logical device implementation using quantum cellular automata", *J. Applied Physics*, 75, p. 1818, 1994.

[7] G.S. Lent and P.D. Tougaw,"A device architecture for computing with quantum dots", *Proc. IEEE*, vol. 85(4), 1997.

[8] G.S. Lent and P.D. Tougaw,"Line of interacting quantum-dot cells: a binary wire", *J. Applied Physics*, vol. 74, p. 6227, 1993.

[9] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform.* SIAM Publications, Philadelphia, 1992.



a) Encoding Binary 1 b) Encoding Binary 0

Figure 1-Cell Polarization

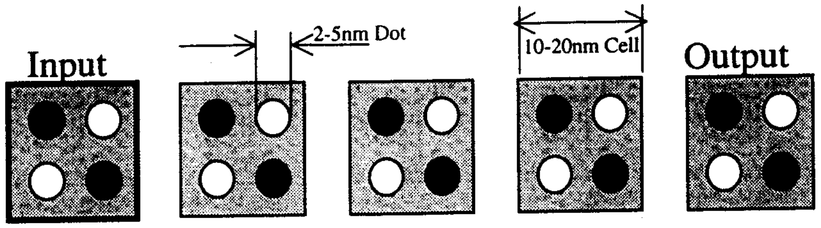


Figure 2-Cell-Cell Interaction to Provide a Binary Wire

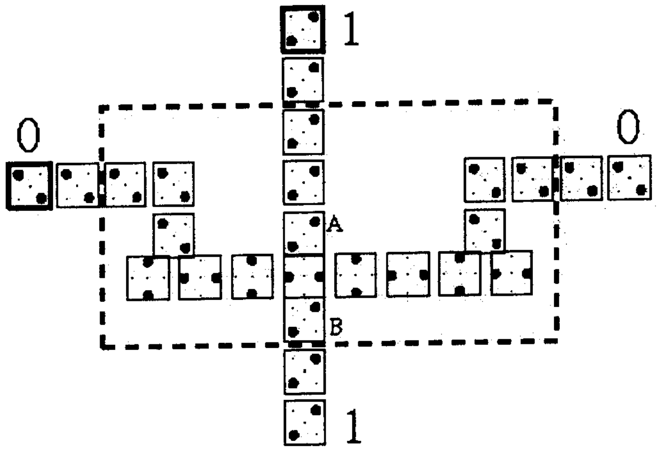


Figure 3-Co-planar Wire Crossing

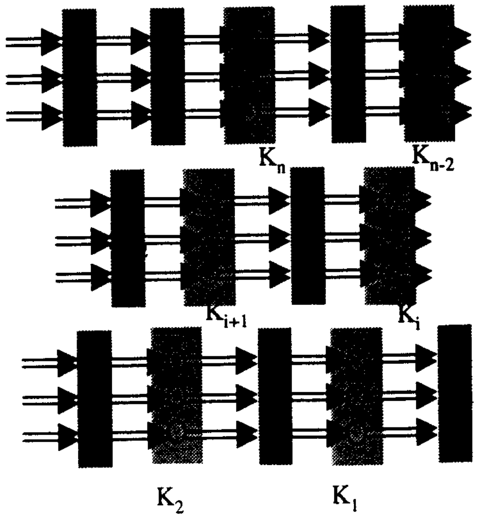


Figure 4-Schematic of a hybrid Architecture (QCA +VLSI) for systolic implementation of FFT