

Novel memory designs for QCA implementation

Marco Ottavi, Vamsi Vankamamidi and Fabrizio Lombardi

Department of Electrical and Computer Engineering
Northeastern University,
Boston, (MA) 02115, USA
Email: {mottavi,vvankama,lombardi}@ece.neu.edu

Salvatore Pontarelli

Department of Electronic Engineering
University of Rome "Tor Vergata"
Rome, 00133, Italy
Email: pontarelli@ing.uniroma2.it

Abstract—Quantum-dot Cellular Automata (QCA) provides a new functional paradigm for information processing and communication. The main feature of this technology is the so-called processing-in-wire mechanism by which data movement and manipulation are strictly integrated. In this context, the design of memory devices is particularly challenging and interesting because the conventional storage arrangements applicable to CMOS based memories cannot be applied and innovative approaches must be used. This paper analyzes state-of-the-art designs for QCA memories and proposes three new architectures that improve over past approaches different figures of merit.

I. INTRODUCTION

In the past few decades, the exponential scaling in feature size and increase in processing power have been successfully achieved by VLSI using mainly CMOS technology; however there is substantial evidence [9] that emerging technologies (mostly based at nano scale ranges) will be required to supersede the fundamental physical limits of CMOS devices. Among these new technologies, *Quantum-dot Cellular Automata* (QCA) gives a solution at nano scale and also offers a new method of computation and information transformation. Interconnections for signal transfer are used for logic computation and manipulation by which the so-called processing-in-wire and memory-in-motion paradigms are accomplished. Micro-sized QCA devices have been fabricated with metal cells which operate at 50mK [7] (i.e. cryogenic). [7] has reported an experimental demonstration of a metal QCA cell; such device consists of four metal dots, connected with tunnel junctions and capacitors. Basic logic behavior of these cells has been demonstrated in [1] through a majority voter (MV) as primitive block for QCA design. It is also stated that room temperature operation requires QCA cells to be fabricated in the range of 1nm - 5nm size. Some possible realizations of molecular QCA have been proposed.

Clocking is an important feature for QCA. Signal propagation is accomplished along serial timing zones by the one-dimensional technique of [5]. This one-dimensional arrangement results from the four phases required for correctly operating the QCA cells. As shown in [5], QCA has many desirable features for processing; clocking and

timing can be adjusted as function of the cells in a Cartesian layout with low power, high density and regularity. Different devices and circuits have been proposed for QCA implementation. These include a carry look ahead adder, a barrel shifter, microprocessors and FPGAs [6] [10] [3] [13] [4].

Large memory designs in QCA present unique characteristics due to their architectural structure. The objective of this paper is to provide a comprehensive review of QCA memory architectures and propose new solutions. These new QCA memories exploit the features of the memory-in-motion paradigm. The paper is organized as follows: Section 2 provides an overview of QCA memory architectures, while Section 3 introduces a novel loop-based architecture that allows serial write and parallel read access. Section 4 introduces a novel design approach that utilizes a new clocking scheme for compact and efficient design of serial and parallel memories. Finally in section 5 a comparative analysis is pursued; the different architectures are analyzed using various figures of merit as well as the desired application requirements. Conclusions are also drawn.

II. QCA MEMORY ARCHITECTURES OVERVIEW

A straightforward approach to implement a memory by QCA is to maintain a cell (zone) in the Hold phase as long as its value must be retained for storage. The main problem with this rather obvious approach is the requirement of an explicit control of the CMOS clock signal from the decoder (which is implemented in QCA). Also, the transfer of signals from QCA to CMOS requires a complicated sensing process using sophisticated electrometers. For a truly QCA based implementation, memory must be kept in motion, i.e the memory state has to be continuously moved through a set of QCA cells connected in a loop partitioned into 4 clocking zones and at any given time, one of them is in the Hold phase to retain the information. In the technical literature, QCA based memories can be mainly classified into parallel and serial architectures. A parallel architecture offers the advantage of low latency, at each memory cell, only one data bit is stored, so there is no delay in that bit reaching the Read/Write circuitry. In a serial design, multiple bits are stored in each memory

cell and share the Read/Write circuitry, thus resulting in a delay proportional to the word size. [2] has made an early attempt to design a serial QCA memory using the so-called SQUARES formalism. The basic principle of this technique is to define a set of equally sized blocks, each performing a basic function in QCA. These blocks can then be tiled together to design more complex QCA circuits. The obvious advantage of this technique is the ease in the geometric layout; However, as the blocks are of standard size, a substantial unutilized area appears in each block, thus causing spatial redundancy and lower density in the overall design. Clocking each SQUARE requires a large number of clocking zones even for a modest memory size, thus also requiring a considerable amount of CMOS circuitry to generate the clocking signals. [4] has introduced a H-Memory architecture with high density and uniform access time. The H-Memory has a complete binary tree structure with control circuitry at each node; as the memory spirals are at the leaf nodes, an integration of logic and memory is accomplished in the layout, but the control circuitry and memory are logically separate (similarly to CMOS design). However unlike conventional designs, control and data bits are serialized. The bit stream enters the memory structure at the root node and traverses down the tree by utilizing one control bit for routing at every node in the path. The architectural choice of dealing with serial bit streams results also in rather complex control logic for QCA. The memory cell at each leaf node is a spiral allowing storage of several bits, while sharing clocking zones between multiple loops. In this design, the memory size at each spiral and the cell count do not have a linear relationship; each outer loop has an increasing diameter, thus requiring more QCA cells for its implementation (although its storage capacity remains constant). [13] has proposed a conventional parallel memory architecture (such as encountered in CMOS-based RAM design) for QCA, i.e. by storing one bit at each memory cell. The single-bit memory cells allow the design of a simple Read/Write circuitry; each memory cell is implemented using 170 QCA cells and the Select signals are separately generated using decoders. The main disadvantage of this approach is the same as the one encountered in [2] namely, data in each memory cell is stored using a closed QCA wire loop (which is partitioned into four clocking zones). Also, clocking zones cannot be shared between memory loops and their dimensions are very small. Therefore, the memory design requires a large number of clocking zones, thus complicating the routing of underlying clock lines.

III. HYBRID MEMORY

The hybrid memory architecture [8] can be considered as an evolution of the serial memory presented in [2]. It is referred to as “hybrid” because it has serial write and parallel read capabilities. This characteristic permits to combine

the low latency advantage of a parallel architecture with the low area requirement (and therefore high density) of a serial architecture. As a serial memory still incurs in slow access for both the write and read operations, this architecture uses a parallel read approach.

A block diagram of the proposed memory architecture is shown in Figure 1. In this Figure, m loops of $2^n = N$

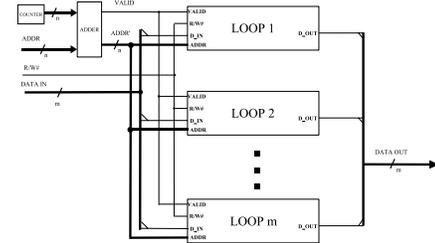


Fig. 1. Block diagram of the hybrid QCA memory

bits are arranged to form a m bit word of $2^n = N$ locations which can be accessed in parallel. Each loop has as inputs the n bit address of the accessed bit and the following additional signals: (1) the $R/W\#$ control signal which specifies if the loop is accessed in a write or read operation; (2) the serial data input D_{in} ; (3) a $VALID$ control signal. The last signal is provided to each loop by the adder and allows the synchronization of the write operation. The write operation must be performed serially on the loops and thus, the correct bit must be addressed. For both the read and write operations, addressing the same bit independently of the configuration of the shift register requires the input address to be added to an offset (which is stored in a 2^n counter). The operation of the hybrid

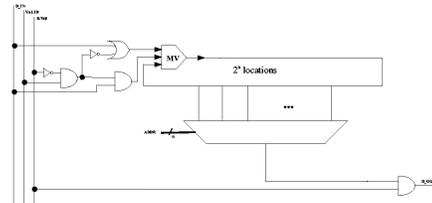


Fig. 2. Loop implementation of the hybrid memory

memory can be described as follows: when a write to the location $ADDR$ is requested, the operation is performed by setting the $VALID$ signal, when the value of the “biased” address $ADDR'$ is zero. Therefore, the write operation can have at most $(2^n - 1)$ clock cycle delay. If a read must be performed, the value of $ADDR'$ is directly provided to the 2^n -to-1 demultiplexers of every loop, thus incurring into an immediate (virtually zero delay) read operation for the addressed m bits word. The logic structure inside each loop is shown in Figure 2. The inputs of the loops are the m bits $ADDR'$, D_{in} , $VALID$ and the $R/W\#$ signals, while the output is the D_{out} signal. Note that the write logic

circuitry provides the inputs to the majority voter (MV) to either change the value of the stored information (placing the same new data at two of the three inputs), or leave it unchanged (placing a 0 and 1 at two inputs).

IV. LINE BASED MEMORY

In this section, a fundamentally different design of a QCA memory is introduced. This architecture is based on a novel logic arrangement for the MV, namely the wires to an MV can behave differently (either as input or output) in time depending on the clock phase in which they are operative. This arrangement combined with a new clocking strategy, overcomes the limitation of a traditional unidirectional flow of logic signals in QCA.

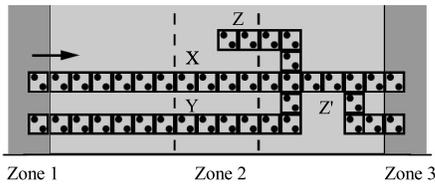


Fig. 3. Majority Voter as a Memory Element

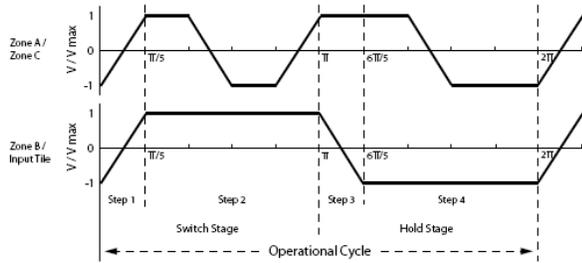


Fig. 4. Clock signals for required switching mechanism

Figure 3 depicts the main principle on which the serial and parallel architectures are based. Differently from all previous architectures (based on a loop structure), the line based approach exploits the Majority Voter gate as a memory element that stores the value of a bit and propagates it as in a shift register. By suitably modifying (as shown in Figure 4) the clock signal the execution of the four different states of the cells (Hold, Switch, Release, Relax) can be controlled; the MV gate acts as a bidirectional element and holds a logic value. In particular, three zones are defined, each clocked with a different signal to provide the execution of four operational steps, i.e. allowing terminal Z of the MV to operate alternatively as an input or as an output (for details see [12], [11]). The main advantage of the line-based approach is that the clocking distribution circuitry is dramatically simplified because the same clocking zones can be shared by different memory cells as per the following implementations.

A. Parallel Memory

Figure 5 shows the schematic diagram of a parallel memory arrangement. The schematic is general and applies

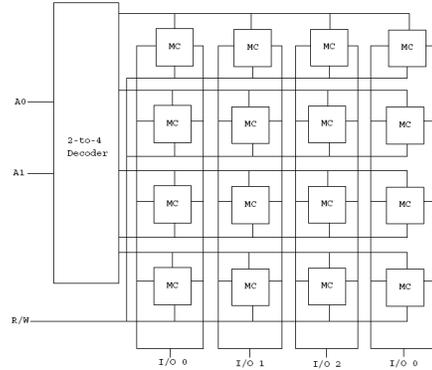


Fig. 5. Parallel Memory Arrangement

also to CMOS based architectures. In the specific parallel memory implementation [11] the memory cells are based on the approach shown in Fig. 3, thus the single memory cells are actually majority voters embedded in the novel clocking scheme. With the given clocking approach all bit lines (columns) share the same clock signal, thus allowing to have a straightforward and therefore easily implementable column-wise distribution of the clock signal.

B. Serial Memory

The memory cell of the line based serial architecture [12] consists of two long horizontal wires connected together at both ends by two short vertical wires, thus creating a loop for the memory-in-motion paradigm. Three types of tiles are utilized for a QCA implementation of this architecture, input and output tiles (which close the loop as in typical serial QCA memory implementations) and the internal memory tile, which is based on the dual-way Majority Voter operational principle (as discussed previously). The significant difference, which also confirms the validity of the proposed approach, is that a mirrored circuit appears above the MV to propagate the information in the opposite direction (see Fig. 6). Remarkably, the mirrored circuit is still built on the same underlying clocking scheme

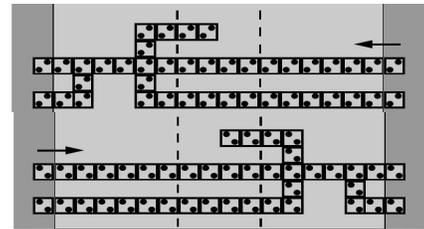


Fig. 6. Serial Memory Tile

thus allowing the stacking of loops and sharing the same clock. The number of tiles into which the memory loop is partitioned, determines the word size, while the number of stacked memory loops determines the memory size.

V. DISCUSSION AND CONCLUSIONS

For evaluating the performance of QCA based memories some of the figures of merit used in CMOS based designs can be utilized. However, some evaluation parameters are closely related to the QCA technology and related paradigms. Specific issues of QCA based memory architectures are the clock distribution and the timing of the clock schemes. In this paper the following figures of merit are considered: (1) the read and write access latency defined as the number of clock cycles for accessing a memory bit in the read or write mode (assuming that the clock frequency is the same for all considered architectures); (2) memory density defined as the number of QCA cells per stored bit; (3) the clocking distribution complexity. Table I provides a comparison of different memories ([4] is not considered because the H-Memory utilizes an addressing technique based on interleaved packets of data and address using a customized design, which would introduce a different set of figures of merit).

	Read Latency	Write Latency	Cells per bit	Clocking Distribution
[13] Parallel	n	n	170	Complex
SQ Serial	n	n	$70+100/n$	Complex
Line b. Serial	n	n	$83+150/n$	Simple
Line b. Parallel	1	1	233	Simple
Hybrid	1	n	$95+100/n$	Complex

TABLE I
COMPARISON OF FIGURES OF MERIT

In Table I, n represents the number of bits stored in the cell and *SQ Serial* refers to the solution proposed by [2]. The following considerations can be drawn from these results and architectures. (1) Serial memory implementations provide a very high density, but they have an operational latency that is proportional to the number of stored bits. These architectures are best suitable for applications in which data is accessed in a serial mode e.g. the memory organized as in a file system. (2) The hybrid memory presents a slightly smaller density than the serial organization, but it improves over latency during the read operation. This architecture is best suited in applications in which data is written rarely or in a burst mode, but the read operation is performed often (e.g. like the program memory of a microprocessor). (3) The parallel memory resolves most of the problems related to latency, but the density of this solution is low compared with other arrangements.

For comparing the clocking distribution circuitry the clocking scheme of [13], [2] serial and the proposed hybrid require that for each stored bit the loop must be divided into four different zones which must be clocked

with four different phases; this could require a complex clock distribution circuitry. For the line-based serial and parallel memories this could be a difficult requirement if the maximum and minimum hold and switch times are tight because the modified clocking signals could have a maximum hold time four times longer than the minimum one. In conclusion, the applicability of a memory architecture depends on the target application, and the presented architectures provide a comprehensive set of excellent alternatives.

REFERENCES

- [1] I. Amlani, A. Orlov, G. Toth, C. Lent, G. Bernstein, and G.L.Snyder. Digital logic gate using quantum-dot cellular automata. In *Science*, volume 284, pages 289–291.
- [2] D. Berzon and T. Fountain. A memory design in qcacs using the squares formalism. In *Proceedings Ninth Great Lakes Symposium on VLSI*, pages 168–172, 1999.
- [3] V. S. Dimitrov, G. A. Jullien, and K. Walus. Quantum-dot cellular automata carry-look-ahead adder and barrel shifter. In *IEEE Emerging Telecommunications Technologies Conference*, 2002.
- [4] S. Frost, A. Rodrigues, A. Janiszewski, R. Rausch, and P. Kogge. Memory in motion: A study of storage structures in qca. In *First Workshop on Non-Silicon Computing*, 2002.
- [5] C. Lent and P. Tougaw. A device architecture for computing with quantum dots. In *Proc. of the IEEE*, volume 85, pages 541–557, Mar 1997.
- [6] M. Niemier, A. Rodrigues, and P. Kogge. A potentially implementable fpga for quantum dot cellular automata. In *1st Workshop on Non-Silicon Computation (NSC-1), held in conjunction with 8th Int. Symp. on High Performance Computer Architecture (HPCA-8)*, Boston, MA, 2002.
- [7] A. Orlov, I. Amlani, G. Bernstein, C. Lent, and G. Snyder. Realization of a functional cell for quantum-dot cellular automata. In *Science*, volume 277, pages 928–931, 1997.
- [8] M. Ottavi, S. Pontarelli, V. Vankamamidi, and F. Lombardi. Design of a qca memory with parallel read/serial write. In *Proceedings of 2005 IEEE Computer Society Annual Symposium on VLSI*.
- [9] R. Compano, L. Molenkamp, and D.J. Paul. Technology roadmap for nanoelectronics. In *European Commission IST programme, Future and Emerging Technologies*.
- [10] P. Tougaw and C. Lent. Logical devices implemented using quantum cellular automata. In *Journal of Applied Physics*, volume 75(3), pages 1818–1825, 1994.
- [11] V. Vankamamidi, M. Ottavi, and F. Lombardi. A line-based parallel memory for qca implementation. In *Internal report, available upon request*.
- [12] V. Vankamamidi, M. Ottavi, and F. Lombardi. Tile based design of a serial memory in qca. In *Proceedings of ACM Great Lakes Symposium on VLSI Chicago, Illinois, Apr. 2005*, pages 201–206.
- [13] K. Walus, A. Vetteth, G. Jullien, and V. Dimitrov. Ram design using quantum-dot cellular automata. In *Technical Proceedings of the 2003 Nanotechnology Conference and Trade Show*, volume 2, pages 160–163, 2003.