INTERNATIONAL JOURNAL OF CIRCUIT THEORY AND APPLICATIONS
Int. J. Circ. Theor. Appl. 2001; **29**:49–62

Problems in designing with QCAs: Layout = Timing

Michael T. Niemier*,† and Peter M. Kogge

Department of Computer Science and Engineering, University of Notre Dame, 384 Fitzpatrick Hall, Notre Dame, Indiana 46656, U.S.A.

SUMMARY

The quantum cellular automata (QCA) is currently being investigated as an alternative to CMOS VLSI. While some simple logical circuits and devices have been studied, little if any work has been done in considering the architecture for systems of QCA devices. This work discusses the progress of one of the first such efforts. Namely, the design of dataflow components for a simple microprocessor being designed exclusively in QCA are discussed. Problems associated with initial designs and enumerated solutions to these problems (usually stemming from floorplanning techniques) are explained. Finally, areas of future research direction for circuit design in QCA are presented. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: QCA; architecture; processor dataflow, circuit design, quantum cellular automata

1. INTRODUCTION

While there is still much work to be done, early results indicate that QCA could be a very viable alternative to CMOS. QCA cells and a QCA majority gate have been fabricated and tested successfully [1,2]. However, the actual design of many of the circuits and devices required for a QCA microprocessor have not yet even been considered. What is required is a methodology for constructing and designing the QCA circuits that are essential for a design such as a microprocessor. Furthermore, understanding how circuits are built should assist in the actual design of the devices themselves. It will also serve to open a discussion about architectural issues of QCA and other nanotechnologies.

As a means for generating the QCA architecture an obvious first step is to translate existing CMOS designs directly into QCA majority gate logic. However, while such a translation is possible, the nature of QCA devices will require an architecture that is radically different from conventional CMOS.

*Correspondence to: Michael T. Niemier, Department of Computer Science and Engineering, University of Notre Dame, 384 Fitzpatrick Hall, Notre Dame, Indiana 46656, U.S.A.
†E-mail: mniemier@nd.edu

Contract/grant sponsor: National Science Foundation

Received 30 March 2000
Revised 18 July 2000

QCA designs are inherently pipelined and require a device and clocking methodology that differs significantly from CMOS designs. Because of the importance of the QCA clock in creating actual designs it will be briefly discussed in the following subsection.

The next section (Section 2) will provide a short background to what is actually being designed. Section 3 will discuss a first-cut design and Section 4 will discuss problems associated with this design. Section 5 will provide solutions to these problems and Section 6 will discuss a schematic reflecting these solutions. Finally, Section 7 will discuss areas of future work.

1.1. The basics

The clock in QCA is multi-phased. Individual QCA cells are not timed separately. The wiring required to clock each cell individually could easily overwhelm the simplification won by the inherent local interconnectivity of the QCA architecture [3]. However, an array of QCA cells can be divided into subarrays that offer the advantage of multi-phase clocking and pipelining. For each subarray, a single potential modulates the inter-dot barriers in all of the cells in a given array [3].

This clocking scheme allows one subarray to perform a certain calculation, have its state frozen by the raising of its interdot barriers, and have the output of that subarray act as the input to a successor array (i.e. clocking subarray 1 can act as input to clocking subarray 2). During the calculation phase, the successor array is kept in an unpolarized state so it does not influence the calculation. Each of the four clocking subarrays corresponds to one of four different clocking phases. Neighbouring subarrays concurrently receive neighouring clocking phases [3].

Finally, it is important to reiterate and stress what exactly is meant when referring to the QCA ‘clock’. As mentioned above, the QCA clock has more than a high and a low phase. Additionally, it is not a ‘signal’ with four different phases. Rather, it can be said that the clock changes phase when the potential barriers that affect a group of QCA cells (referred to as a *clocking zone*) are raised or lowered or remain raised or lowered (thus accounting for the four clock phases). Furthermore, all of the cells within a clocking zone obviously are in the same phase. It is said that one clock cycle occurs when a given clocking zone cycles through the four different clock phases. The purpose of the ‘clock’ is to trap one set of cells in a specific polarization which in turn allows other cells to make appropriate changes. More will be said about this in the next subsection.

1.2. How it actually works

During the first clock phase, the *switch phase*, QCA cells begin unpolarized and their interdot potential barriers are low. The barriers are then raised during this phase and the QCA cells become polarized according to the state of their driver (i.e. their input cell). It is in this clock phase that the actual computation (or switching) occurs. By the end of this clock phase, barriers are high enough to suppress any electron tunnelling and cell states are fixed. During the second clock phase, the *hold phase*, barriers are held high so the outputs of the subarray can be used as inputs to the next stage. In the third clock phase, the *release phase*, barriers are lowered and cells are allowed to relax to an unpolarized state. Finally, during the fourth clock phase, the *relax phase*, cell barriers remain lowered and cells remain in an unpolarized state [3]. The four clock phases are illustrated in two different ways in Figures 1 and 2.

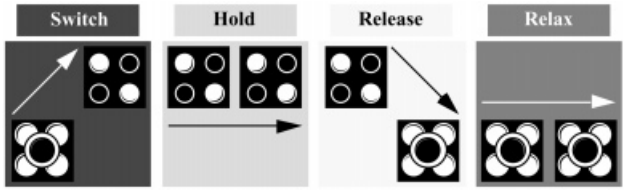

Figure 1. The four phases of the QCA clock.

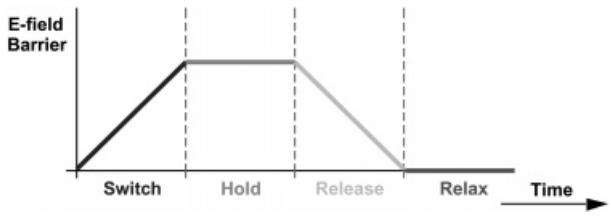

Figure 2. The four phases of the QCA clock (an alternative expression).

2. DESIGN BACKGROUND

To develop a library of design rules and hence the QCA architecture, we are designing and simulating a custom design of a microprocessor called Simple 12 entirely in QCA. The advantages of choosing Simple 12 are three fold. First, the processor IS simple. Simple 12 has 12-bit data words, an 8-bit addressable memory, and uses minimal hardware. Consequently, much of the physical layout can be performed by hand. Second, an actual processor will be designed with an instruction set that includes arithmetic instructions, loads, stores, and jumps. Therefore, solutions to the difficulties encountered in this design will apply to even more complex systems of custom and synthesized logic. Third, we have completed and fabricated a 2 μm CMOS Simple 12. Thus, it will be possible to make comparisons to an existing design in a technology on which we are trying to improve.

2.1. The Simple 12 dataflow

A high-level block diagram of Simple 12 appears in Figure 3. Again, although simple, it exhibits almost all of the major attributes of a more complex design. As mentioned, the design includes three registers, address and data buses, feedback paths, and a memory interface.

2.2. Functions of the Simple 12 ALU

To successfully execute the Simple 12 instruction set, the Simple 12 ALU must be able to generate the following outputs (where A and B are inputs into the ALU): A+B, A−B, A AND B, A OR B, B, B+1, and 0. To generate these outputs several control signals are needed that serve as inputs to intermediate signal generation logic of the ALU [4].

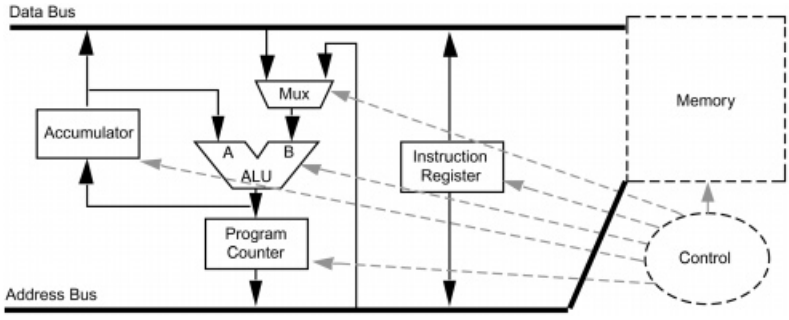

Figure 3. The Simple 12 datapath.

3. A FIRST-CUT OF THE SIMPLE 12 ALU

The QCA ALU was largely designed by translating the logic of the transistor version of the ALU to an equivalent QCA representation. Essentially, equivalent QCA majority gate representations of the transistor logic were determined and implemented. The only difference between the QCA ALU and the transistor ALU is that the QCA ALU did not use a mirror adder, but the full adder designed by Lent *et al.* [3] (the CMOS design used a full adder).

The design of the ALU can essentially be broken down into three blocks. One block represents the adder, another block represents the portion of the ALU that performs operations such as AND and OR (the logic unit), and the last block contains logic for intermediate ALU signal generation [4].

In Figure 4, the three parts of the Simple 12 ALU—adder, logic unit, and intermediate signal generation logic—are joined to form the first-cut of the Simple 12 ALU. Problems that exist with this design will be discussed in the next section and solutions to such problems will be discussed in the section after that.

4. PROBLEMS WITH THE FIRST-CUT OF THE SIMPLE 12 ALU

The first-cut of the Simple 12 ALU has five significant problems. First, wire lengths vary from extremely short (i.e. 4 QCA cells in length) to extremely long (i.e. 36 QCA cells in length). Second, the clocking zones in this first-cut have non-uniform widths. Third, some clocking zones contain a very large number of QCA cells while others only contain a few. Fourth, there is no means for generating feedback in this design. Fifth, and finally, there is a large amount of white-space/wasted area in this design. Why the above five characteristics are problems will be discussed in the subsections below. Additionally, the problems are illustrated in Figure 5. These problems (and more importantly solutions to them) must be considered when designing future circuits.

4.1. Wire length

When generating designs in QCA, a significant effort should be made to keep the length of a wire within a given clocking zone to a minimum. There are two very important reasons to

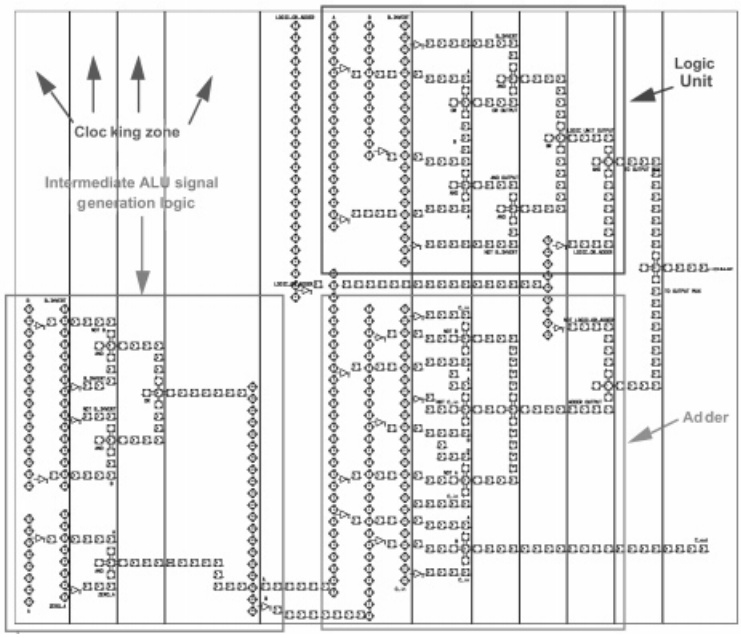

Figure 4. First cut of the QCA Simple 12 ALU.

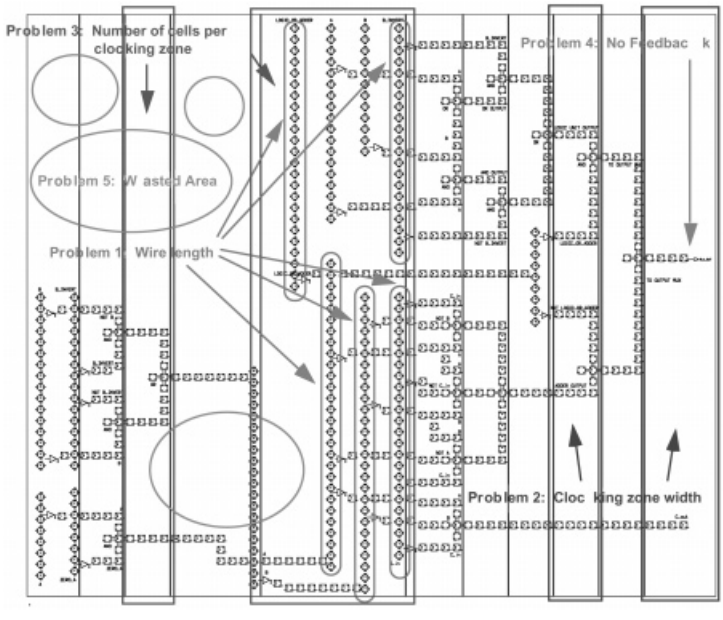

Figure 5. First cut of the QCA Simple 12 ALU.

do this. First, as wire length grows, the probability that a QCA cell will switch successfully decreases in proportion to the distance a particular cell is from a frozen input at the beginning of the 'wire' [3]. Consequently, for shorter wires, there is a higher probability that all cells making up the wire will switch successfully. Additionally, wire length will determine the clock rate—or in other words, the rate at which clocking zones can change clock phases. This is so because, before a given zone can change phase, every cell within the zone must make appropriate polarization changes. Obviously, the longer the wire, the longer the time for a signal to propagate down the length of it.

4.2. Clocking zone width

Like wire length, there are also two important reasons for keeping clocking zone width to a minimum. The first reason centres around the desire to keep wire lengths at a minimum. If clocking zone widths are narrow, it will force the designer to keep wire lengths small (at least in one dimension). For example, in most of the clocking zones in Figure 5, a horizontal wire is composed of no more than 5 cells. A second reason centers around uniformity. A conscious effort has been made to make circuits that have been designed in QCA as uniform as possible. This was done to hopefully increase the manufacturability of the circuits and designs if and when that time comes.

4.3. Number of QCA cells per clocking phase

As can easily be seen in Figure 5, there is a large disparity between the number of QCA cells in some clocking zones when compared to the number of QCA cells in other clocking zones (i.e. in the clocking zone at the far right of Figure 5 there are only 8 cells in the zone while in the clocking zone in the middle of Figure 5 there are 211 in the zone).

If too many cells are included in a single clocking zone, the clock rate could deteriorate (simply because the time for all cells to make there required transitions will most likely increase). However, more importantly, for arrays of cells on the order of 10^3 (i.e. 10^3 cells per clocking zone), there will be a tendency for the system to settle into an excited state rather than a ground state—and a cell is in a ground state when it has a definitive polarization, and hence logical value. This occurs because of thermodynamic effects.

It is worthwhile to include a short discussion of what exactly *thermodynamic effects* are. Such effects were first described by Lent *et al*. If thermal fluctuations excite an array of cells above its ground state, i.e. so that the cell does not have a definite polarization, wrong answers can appear at the outputs of a circuit. To be robust, the excitation energy must be well above $k_B T$ (where k_B is Boltzmann's constant and T is the operating temperature). It can be determined from calculations that a maximum operating temperature for cells depends in part on the size of a cell. As cell sizes decrease, the energy separations between states increase and higher temperature operation is possible [3].

Additionally, consider a linear array of N cells acting as a wire transmitting a logical 1. The ground state for such a configuration would be all of the cells obtaining the same polarization as that of the input (or driving cell). The first excited state of this array will consist of the first m cells polarized in a representative binary 1 state and $N–m$ cells in the binary 0 state. This excitation energy of this state (E_k) is the energy of introducing a 'kink' in the polarization. The energy is independent of where the kink occurs (i.e. the exact value of m). As the array N becomes larger, the kink energy E_k remains the same. But, the entropy of this excited state

increases (as there are more ways to make a 'mistake' in a larger array). When the array size reaches a certain size, the free energy of the mistake state becomes lower than the free energy of the correct state. A complete analysis reveals that the maximum number of cells in a single array is given by $e^{E_k/k_B T}$. This again requires the excitation energies to be significantly larger than $k_B T$. Finally, the kink energy increases as the system is scaled to smaller sizes [3].

4.4. Lack of feedback

A significant 'logical' problem with the design appearing in Figure 5 is the complete absence of physical feedback—namely, a value generated as output from the circuit has no means for travelling back to the input. Physical feedback is all but essential in most useful microprocessors and finite state machines. (As a simple example, consider a register. Any processor should be capable of writing to a register and using a register as input. Thus, a path should exist from the output of a dataflow, to a register, and back to the input of that dataflow. Without feedback, this would be impossible.) Furthermore, as seen in Figure 3, the Simple 12 dataflow requires some form of physical feedback. However, in Figure 5 data flows only in one direction. Some method of allowing the output of a given circuit to be used again at the input must be generated.

4.5. Wasted area

A final problem with the design appearing in Figure 5 is that it is not very space efficient (examples of wasted area are illustrated with circles). A significant cause of wasted space in this design comes as a result of the intermediate signal generation logic. Both the logic and adder unit require the A and B inputs to be changed to perform certain operations. Consequently, the intermediate signals must be generated before inputs reach the logic or adder unit. In the first-cut design, data flows only in one direction (to the right). Therefore, the intermediate signal generation logic must precede the logic and adder units. However, despite its importance in precedence, the intermediate signal generation logic is actually two very simple circuits. As a result, not much area is required. Still, because it comes before the other two units, space is wasted.

5. FLOORPLANNING

This section will discuss methods for solving the five problems discussed in the previous subsection. Many of the solutions stem from a specific arrangement of clocking zones onto which a QCA circuit is overlayed [4]. For this reason, this section is entitled 'Floorplanning'.

5.1. Trapezoidal clocking

As discussed in Section 4.5, a significant problem with the first-cut design is wasted area in the design. In particular, this wasted area comes from logic required to generate intermediate ALU data. To remedy this particular problem, the new technique of 'trapezoidal clocking' will be introduced. In Figure 6, QCA logic to generate intermediate ALU data is not placed in front of the computational logic but rather, below it. Instead of leaving large gaps—or areas with no logic (like those appearing in Figure 5), 'trapezoids' containing computational

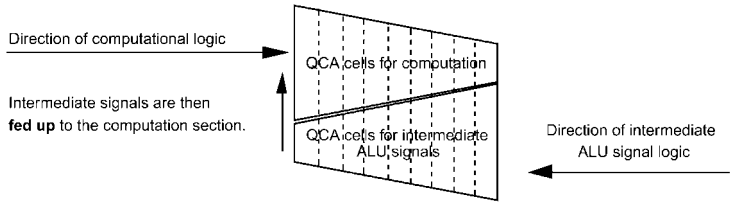

Figure 6. A description of trapezoidal clocking.

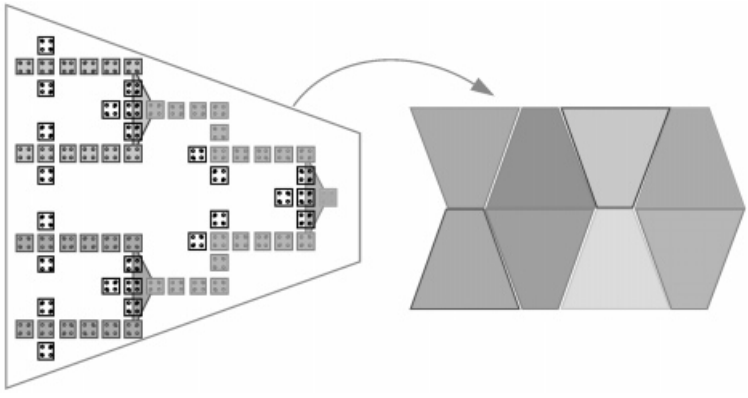

Figure 7. A QCA 'tournament bracket' and potential for very dense circuits.

logic and intermediate signal generation logic can be fit together to minimize wasted area. Thus, data will flow in two different directions. It should be noted that the dotted lines in Figure 6 represent clocking zone boundaries. Thus, the computational and intermediate signal generation logic would still be divided up into clocking zones as depicted in Figure 5.

It is also worth noting that QCA inherently lends itself to such a 'trapezoidal' structure. In QCA circuits, an output is usually generated from a single gate or wire. If it is a logical gate from which the output comes, that gate usually performed a computation by using inputs generated from 2 or 3 other logical gates. This process most likely continues 'backwards' until some inputs are reached. In this way, it is not unlike a 'tournament bracket' in which there are an initial n slots and after a certain number of m stages, a single slot remains. As illustrated in Figure 7 (and in the first cut designs appearing in the figures of Sections 3 and 4), QCA circuits have a very similar form. By allowing data to flow in two directions (as shown in Figure 6) and by carefully fitting 'trapezoids' together it would seem very possible that very dense and compact QCA circuits could be generated.

5.2. Feedback and trapezoidal clocking

Trapezoidal clocking does not only provide a means for minimizing total area. It can also be used to implement a feedback path in QCA circuits. In Figure 8, the four clocking phases are each given a number (1, 2, 3 and 4) and a colour/shade. These correspond to the four different clock phases that were discussed in Sections 1.1 and 1.2 and illustrated in Figures 1

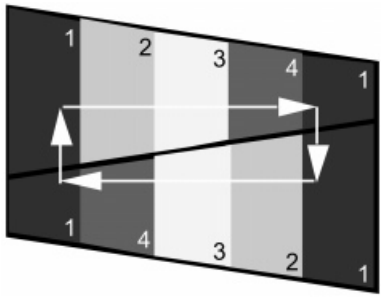

Figure 8. A trapezoidal clocking floorplan with clocking zones.

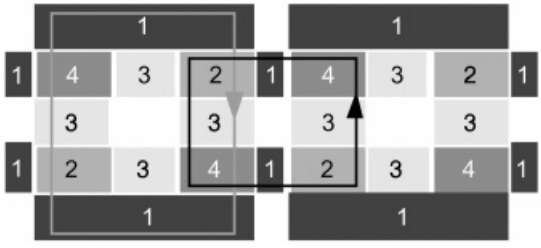

Figure 9. The universal clocking cell.

and 2. If the top 'trapezoid' is computational logic, data can be fed back to the input (assumed to be in clocking zone 1 at the far left) after 'switching' in clocking zone 1 at the far right. White arrows illustrate the feedback path through the numbered clocking zones. It can easily be seen that the clocking phases are traversed in the proper order (i.e. in the order 1, 2, 3 and 4—and so that the required clock phases are always adjacent to one another to allow for correct signal propagation). Furthermore, a signal can start at a given point and a path exists to return to that point—the definition of feedback.

5.3. A universal clocking cell

The next question to be asked is whether or not the clocking zone arrangement illustrated in Figure 8 can be extended to allow efficient and easy wire routing. Thus, can the clocking zones be arranged or tiled so that there are multiple 'wire' loops and 'wire' crossings and still allow feedback? Such a floorplan is necessary because for designs and components (such as the QCA Simple 12 ALU) multiple control and data input and output wires will have to be included in the design. The ALU also requires some feedback mechanism. Furthermore, a standardized clocking floorplan would provide a start for a new design, as a way to run wire and generate feedback would already exist. Such a pattern is possible and is illustrated in Figure 9. As seen in Figure 9, several different loops can be generated that cross (recall that 45 and 90° QCA wires can cross in the plane with no interference of either value being transmitted on either wire) and do not violate the condition that the clocking zones must be traversed in the order 1, 2, 3 and 4.

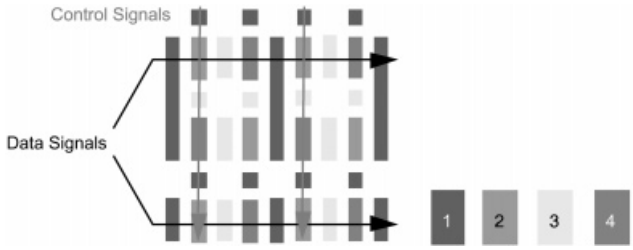

Figure 10. A universal clocking floorplan with data and control signal routing.

5.4. Universal clocking floorplans and data and control routing

The pattern that appears in Figure 9 can be tiled to form a universal clocking floorplan (UCF). The UCF allows large designs to be constructed and allows extremely complicated paths of QCA wires to be constructed without violating the condition that the clocking zones must be traversed in the order 1, 2, 3 and 4.

The UCF solves the problem of physical feedback in QCA circuitry for large designs. Also, it provides a mechanism for tracing complicated paths while still traversing the clocking zones in the proper order. Furthermore, it provides an extremely efficient manner to run data signals and control signals (Figure 10). Data signals can be run horizontally while control signals are run vertically (or vice versa). This is directly analogous to CMOS VLSI where one layer of metal is run in the vertical direction and used to transmit data signals or control signals, while another layer of metal is run perpendicular to the first and used to transmit data or control signals.

6. A SECOND-CUT OF THE SIMPLE 12 ALU

Using floorplanning techniques discussed in Section 5, the first-cut design of the QCA Simple 12 ALU was reworked to solve the problems of long wire length, inconsistent clocking zone width, the large disparity of QCA cells in wires across some clocking zones, the lack of physical feedback, and wasted area in the design. The design first created to address some of these problems appears in Figure 11. As one can see in the figure, this ‘second-cut’ design does not address all of the problems listed above. For instance, there is still a lack of physical feedback and wasted area in the design. Additionally, there are also still a few cases of ‘long’ wires. However, the clocking zone widths have been made more uniform, the number of cells in a given clocking zone have been reduced, and wire lengths have been shortened.

It is worth mentioning that one potential solution employed in this ‘second-cut’ design to help reduce the number of cells per clocking zone really has nothing to do with innovative floorplanning (i.e. arranging/tiling clocking zones in a specific order). In this ‘second-cut’ design, some clocking zones are simply divided in half to reduce the number of cells in the given zone. Adjacent zones can still have the same phase but are reduced in size to reduce the thermodynamic effects of having too many cells in a given zone (see ★ in Figure 11). (It should be noted that future designs do not specifically employ this feature as the number of cells in our design are not numerous enough for thermodynamic effects to be a concern.)

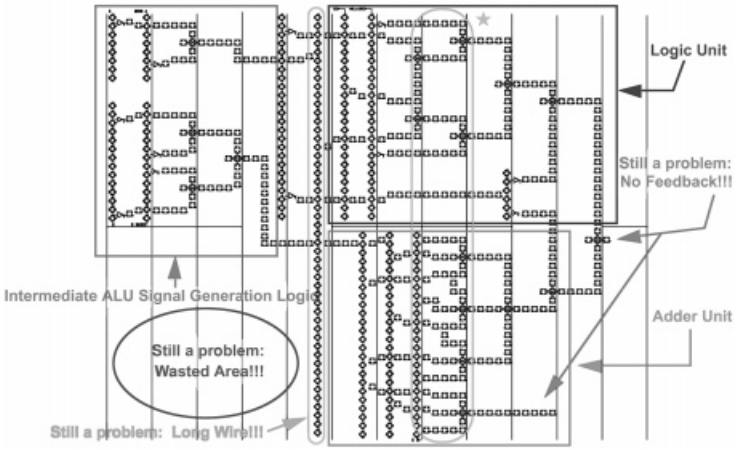

Figure 11. A 'second-cut' design of the QCA Simple 12 ALU.

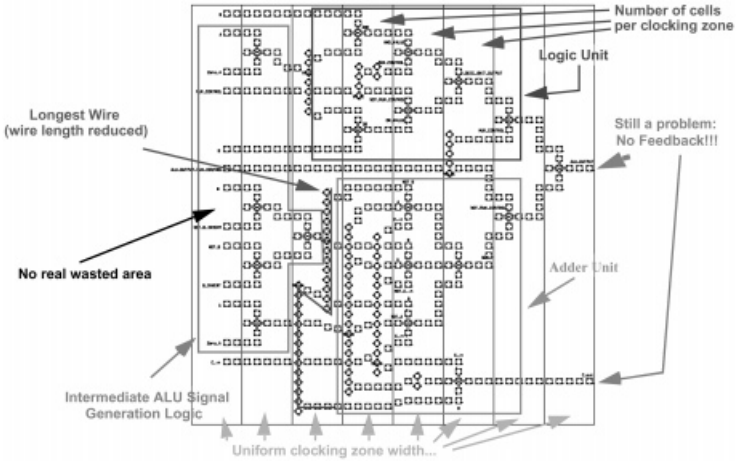

Figure 12. A 'second-cut' design of the QCA Simple 12 ALU.

As illustrated in Figure 11, three significant problems still exist. In an attempt to solve the remaining problems, another design was generated and appears in Figure 12. As seen in Figure 12, the problems of wasted area and long wire lengths have all but been eliminated. The method used to eliminate wasted area was to simply duplicate portions of the intermediate signal generation logic—in particular, the logic needed to zero the A input. This logic was simply placed in front of the adder unit and the logic unit. As this logic only involves an AND gate, the additional QCA cells required were minimal. Interestingly, this duplication of logic actually solves two problems at the same time. The duplication of logic eliminates the need for long 'routing wires' to carry signals to various portions of the ALU. This not only saves area, but also reduces wire length!

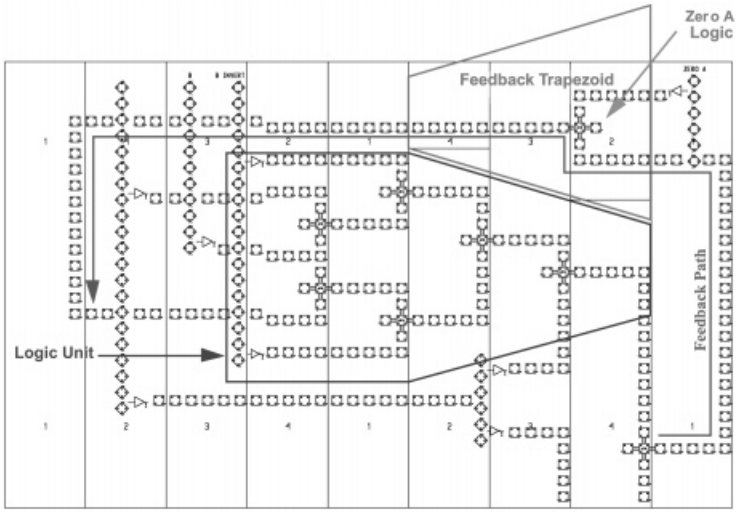

Figure 13. A simple example of feedback in a QCA circuit schematic.

It should be mentioned that there is no need to duplicate the logic that will change the B input signal for various ALU operations. The B signal only needs to be altered if the output from the adder unit is desired. Consequently, the 'normal' B input can be fed to the logic unit.

6.1. Feedback and its applications—the remaining problem

The two second-cut designs have addressed four of the five design issues associated with the first-cut design of Section 4. Wire lengths have been reduced, clocking zone widths have been made uniform, the number of cells in a given QCA clocking zone have been reduced to a reasonable number, and wasted area has been eliminated. However, physical feedback does still not exist in any of the first or second-cut designs. This is not to say that it is impossible to generate feedback in QCA circuits. Floorplanning techniques discussed in Section 6 clearly indicate that feedback is possible. However, it has not yet been incorporated into any of the designs. A simple example of a QCA feedback circuit will be provided.

6.2. A simple feedback example

Figure 13 shows a portion of the QCA Simple 12 ALU (part of the logic unit and intermediate signal generation logic) that uses the Section 4 floorplanning technique illustrated in Figure 8 to feed a value from the output of the ALU back to the input. While no registers or latches are implemented, this figure serves to demonstrate that feedback is possible in an actual circuit.

There are two other interesting things about this design that are also worthy of mention. First, a portion of this design provides the first illustration of trapezoidal clocking in an actual QCA circuit schematic (it is highlighted in Figure 13). Second, in this schematic, part of the intermediate signal generation logic is placed 'directly after' the output of the ALU—or more specifically, above it in the 'feedback trapezoid'. Thus, the A input to the ALU is zeroed on

the way back to the input of the ALU. Essentially, useful computation is being performed 'in wire'! This is a VERY valuable technique that can be used to save area and perform useful computation during signal transport.

7. THE FUTURE

Based upon the results discussed above, several topics have been identified as sources for future work and research. First and foremost, Simple 12 must be completed. Potential problems in accomplishing this task stem from the existence of 'long' wires in designs and from signal routing. Some technology issues must also be considered. One of the most important areas for investigation might be how a clocking zone would physically be implemented. Potential areas of future work are discussed below.

7.1. Technology issues

There are many opportunities for research when considering the 'lowest levels' of QCA design. For example, means for actually implementing the clocking zones that are an integral part of all of the designs in this thesis must be developed and pursued. In particular, a method must be found that assures that there is a sharp 'edge' between two clocking zones. Also, research efforts should be directed into determining how a specific logical value (i.e. a binary 1 or 0) is actually 'read-in' by the design or 'read-out' by somebody or something in the outside world.

Additionally, specific methods and tools for developing power, area, and speed models for a given design should be researched and developed. Along with this, the effects of devices with 'defects' and how they affect an overall design should be considered. Finally, while a rather thorough set of design rules for QCA circuits has been developed and presented here, work should continue to form a complete set of design rules that will apply to all possible QCA circuits (i.e. design rules for memory devices and control and state machine logic should be developed and enhanced).

7.2. Logic design

Based on the discussions of Section 4 (problems with the first-cut design of the Simple 12 ALU) work in the immediate future will have an extensive focus on methods for efficient interconnect and routing. Obviously, floorplans are required to properly route and time signals from the control logic block (as well as from memory). However, more importantly, floorplanning techniques must be studied further in an effort to eliminate long wires and crowded clocking zones that currently result from the interconnect between two bit slices of the Simple 12 dataflow.

7.3. Architecture

Because of the inherent pipelining created by the clocking zones of a QCA design, QCA is an ideal candidate for multithreading. In an ideal situation (i.e. assuming that all control signal and data routing timing and delay problems can be worked out successfully), new data bits for a separate instruction stream could enter the dataflow every time a clocking zone

bordering the inputs recycles back to the switch phase (or in other words it goes through each of its four phases). This would result in a remarkable throughput.

REFERENCES

1. Amlani I. Digital logic gate using quantum-dot cellular automata. *Science* 1999; **284**:289.
2. Amlani I, Orlov AO, Snider GL, Lent CS. Experimental demonstration of a binary wire for quantum-dot cellular automata. *Applied Physics Letters* 1999; **72**:2179.
3. Lent CS, Douglas TP. A device architecture for computing with quantum dots. *Proceedings of the IEEE* 1997; **85**:541.
4. Niemier MT, Kogge PM. Logic in wire: Using quantum dots to implement a microprocessor. *Proceedings of the 6th International Conference on Electronics, Circuits and Systems*, 1999.

