Reflections from IWQCA 2009

Timothy J. Dysart
tdysart@cse.nd.edu

Abstract

This informal report will highlight my reflections from the discussions at the First International Workshop on QCA held at the U. of British Columbia on Aug. 7 and 8, 2009. These reflections will mainly be centered on circuit designs, system architectures, and the QCA design tool chain (and related topics) for electrostatic cells. Unless noted otherwise, all opinions are those of the author.

1 Circuit Design/Architecture and Killer Application(s)

Finding the appropriate application may be the most critical challenge facing QCA as a technology. All three panelists (K. Walus, C. Lent, G. Snider) in the discussion session highlighted that finding an appropriate application for QCA is a necessary step. One of the major limiting factors in finding this application is that layout == timing [1]. In other words, the clocking structure and arrangement will dictate what types of circuits can be designed.

Because of the influence that the clocking structure will have on the types of circuits that can be created, it may be worthwhile to focus on floor planning rather than circuit layouts if cells cannot be arbitrarily clocked. By floor planning, I am referring to how the clocking regions are arranged within the circuit. If it is assumed that a QCA circuit will rest upon a CMOS (or similar) stack, then the arrangement of the clocking structure will be dictated by the fabrication process design rules. This is likely to lead to circuit floor plans that show a rigid and highly repeatable structure (each clock region may be a rectangle of equal size). From an architectural perspective, these floor plans lead to circuit designs having deep pipelines with limited feedback. Applications such as memory and systolic processing are likely to be able to take advantage of these limitations, thus they should be investigated in more depth.

Another issue discussed in the workshop was that of crossovers. G. Snider believes it to be a non-issue, since a three dimensional wire crossing should be possible; however, my impression is that this may be limited to the metal-dot cells. For molecular implementations, the use of a third dimension seems unlikely for the time being, thus the logical crossing should be favored in circuit designs. While either option can be used in circuit designs, calculations regarding latency and area will be significantly impacted by the choice of crossover utilized. As such, these calculations should select one of the two implementation options and use it.

Overall, because of the crossover issue and the uncertainty (at least in my mind) of the ability to arbitrarily clock individual cells, I tend to find most of the area and latency calculations in the
circuit design literature to be of limited value. From my point of view, I feel that these calculations need to assume either a metal-dot or molecular implementation and use the factors of each to compute these values. Some of the characteristics of a metal-dot implementation would be larger cells, potentially arbitrarily clocked cells, and the use of a three dimensional crossover; for a molecular implementation the basic characteristics might be smaller cells, non-arbitrarily clocked cells, and the use of logical crossovers. Much of the literature seems to assume the use of molecularly sized cells that are arbitrarily clocked and the use of three dimensional wire crossing. The area and latency calculations based on these assumptions are likely to be best case scenarios, which while useful, may never be achievable. As a result, I would encourage the circuit design community to focus on circuit designs that the experimentalists might be able to achieve.

2 QCA Design Tool Chain

My thoughts on the QCA design tool chain are driven by what I see as the experiences of CMOS based circuit design. These opinions have two major thrusts: divorce circuit design from its implementation and the use of specific tools for specific jobs. My intent is not to degrade QCADesigner as a tool, but rather to offer my thoughts as to the direction(s) it should go in the future.

The first thrust, divorcing circuit design from its implementation, can be viewed as applying the ideas behind CMOS $\lambda$-rules to QCA design. These rules, generally credited to Carver and Mead [2], enabled a CMOS based circuit to be designed once in terms of $\lambda$-units for each polygon, and then the value for $\lambda$ varied as the fabrication process changed. Since a specific device with specific characteristics has yet to be defined for QCA, reusing this method of circuit design should be encouraged. Taking this approach would allow a designer to simulate a circuit structure where (assuming a molecular implementation) the cell size and spacing differs without having to redesign the circuit using a different cell type. I developed an elementary method for doing this a few years back [3,4] that is available at www.nd.edu/~tdysart/ResearchOutput/xml/.

The second major thrust is to have specific tools do specific jobs. In essence, there should be a tool capable of doing the circuit design (gate level), another that can translate the gate level design into cells (I believe one exists in QCA-LG), other tools capable of specific simulation methods (i.e. digital, limited physical), and so on. Having one tool that tries to do all of these things will (most likely) lead to a tool that is unwieldy to use, maintain, and develop. Finding the appropriate divisions between tools may be a challenge, and potentially immature at this point, but investing the effort now may simplify the design process down the road.

Overall, Table 1 shows the following steps could form the basis of a design tool chain (with the logic and physical sides being done in parallel). For this design flow, the cell-level layout would merge the digital logic design with components from a parts library (described below).

In general, I am also of the opinion that conducting physical simulation (e.g. coherence vector) of large circuits (e.g. multi-bit adder) is a waste of computing resources. Physical simulation, particularly computationally intense simulation, should be limited to straight-forward logic structures (e.g. majority gates) that can then be glued together. These logic structures can then be utilized to form the basis of a parts library. There is the possibility that a unique parts library will be necessary for each physical implementation, but that bridge can be crossed at a later time.
<table>
<thead>
<tr>
<th>Logic Side</th>
<th>Physical Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gate-level design (account for clocking)</td>
<td>Cell-level part layout</td>
</tr>
<tr>
<td>Digital logic simulation</td>
<td>Detailed physical simulation</td>
</tr>
<tr>
<td>Repeat until logically correct</td>
<td>Repeat until part reaches desired metric</td>
</tr>
<tr>
<td></td>
<td>Cell-level layout</td>
</tr>
</tbody>
</table>

Table 1: Potential QCA circuit design process

References


