Abstract: Novel Architectures and Synthesis Methods for Quantum-dot Cellular Automata

Quantum-dot Cellular Automata (QCA) is an emerging nano-scale low-power fabric that is being explored as an alternative to traditional CMOS devices. The two most interesting features of the QCA paradigm are: a) both wires and gates are created using the same basic element (the QCA cell), and the required functionality is achieved by directing data flow using the QCA clocking scheme. b) the 3-input Majority gate, the primary logic element of QCA, is computationally more powerful than the AND/OR gates of traditional fabrics. In this work, we exploit these features to enhance the usability and efficiency of QCA devices through novel architectures and synthesis methods :

Clocked Coplanar Wire Crossings Traditional wire crossings in QCA use the weak coupling between distant cells and are unreliable. We propose a robust crossing by extending the QCA clock to an 8-phase scheme. This exploits the ability of the QCA clock to direct data flow to eliminate all need for complicated cell positioning and orientations.

p-QCA: A Programmable Architecture We develop a programmable device architecture for QCA that exploits the fabric, rather than import concepts from traditional FPGAs. This architecture eliminates the difference between routing and logic elements on a programmable device, and achieves programmability while retaining near-ASIC performances.

***n*-input Majority Algebra and Minimization** Logic synthesis methods for QCA have been focused on searching for patterns that fit well into 3-input Majority gate networks, and do not consider the fundamental mathematics of Majority. We generalize the 3-input Majority gate into the generic *n*-input Majority gate, and develop the fundamental algebra for *n*-input Majority. This new Boolean algebra for the Majority operator is a generalization of the traditional Boolean algebra for sum and product operators. Using this, we propose a specialized logic minimization method for Majority gates.

Technology Mapping for QCA We bridge the gap between technology-independent *n*-input Majority minimization and the needs of QCA, which uses 3-input Majority (and at most 5-input Majority) gates. We propose an effective method to map unrestricted *n*-input Major-

ity expressions to those that use only 3-input Majority (or at most k-input Majority) terms, as required by the target technology.

Exploiting Symmetry Majority functions form a special class of symmetric functions that can be used to efficiently realize all types of symmetry in a Boolean graph. We develop a direct method of expressing any symmetric function in terms of n-input Majority-terms. Boolean graph decomposition techniques targeted for large scale Majority synthesis need to be aware of the various types and levels of symmetry seen in Boolean functions, and to be able to exploit such symmetry using Majority-terms. We extend the concept of Binary Decision Diagrams to Symmetry Decision Diagrams (SDD), and take the representation of generic Boolean functions closer to Majorities and symmetric forms. We develop the basic framework for building and manipulating SDDs to perform Majority-friendly Boolean graph decomposition.

The *MajSynth* tool We have built *MajSynth*, a logic synthesis tool that implements our two-step strategy of technology-independent logic minimization to n-input Majority expressions followed by technology mapping to 3-input Majority (or 5-input Majority) expressions. We have included symmetry optimizations and SDD-based symmetry-aware graph decomposition techniques in this tool, making it a comprehensive framework for logic synthesis for Majority gates. *MajSynth* is the first comprehensive tool that addresses generic n-input Majority, and handles the core issues of large scale Majority synthesis using the fundamental mathematics of Majority.