Review of:

A Potentially Implementable FPGA for Quantum Dot Cellular Automata

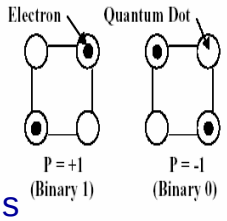
- Paper by:
  - Michael Thaddeus Niemier, Arun Francis Rodrigues, Peter M.Kogge
- Published in:
  - Workshop on Non-Silicon Computation (NSC), in conjunction with Int. Symp. On High Performance Computer Architecture (HPCA)
  - February 3, 2003
- Review by:
  - Dave Lim

General Outline

- Introduction
  - What is Quantum-dot Cellular Automata (QCA)?
- What's already available in QCA?
- Clocking in QCA.
- What's in an FPGA?
- What should be used as a logic block for QCA FPGA?
- What should be used as interconnects for QCA FPGA?
- What would QCA FPGA look like?

Introduction

- Falls under a more general title of nanotechnology
- A QCA cell can be viewed as a set of four change containers or “dots”, positioned at the corners of a square.
- Each cell has two extra mobile electrons
  - Electrons like to be as far away from each other as possible
  - Electrons can only move by tunneling
- Can use QCA as an alternative to CMOS technology
  - First step, figure out a target design that allows for ‘early’ implementation, i.e. FPGAs

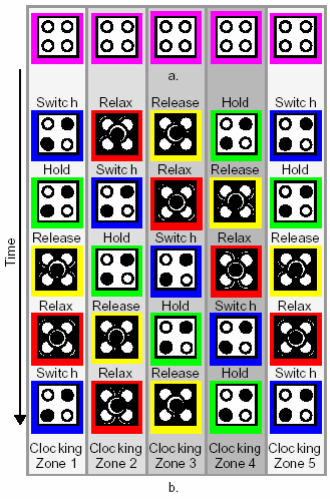


What’s already available in QCA?

- Functional complete logic set exist and has been experimentally tested:
  - researchers at Notre Dame have experimentally demonstrated a ‘majority gate’ which can be used to implement AND/OR gates
  - inverter can also be easily implemented
- QCA binary wire has also been fabricated
  - an array of QCA cells (inherently pipelined)
- Clocked QCA cells have been seen
- Addressable memory cell arrays have been designed and simulated
  - single QCA memory bit has been experimentally demonstrated
- Work now is on how to put everything together to

Clocking in QCA

- 4 clock phases (instead of 2):
  - switch
  - hold
  - release
  - relax



What's in an FPGA?

- Collection of functionally complete logic elements (logic blocks)
- Interconnection framework that connects logic blocks
  - Horizontal and vertical wires with programmable connections (GRM)
  - Direct connection between logic blocks (carry chain)

What should be used as a logic block for QCA FPGA?

- 3 things to consider:
  - what functionality is absolutely needed?
  - how would logic elements in block be programmed?
  - is it simple enough to potentially fabricate?
- Designs considered:
  - a single NAND or NOR gate
  - a single majority gate (programmable AND/OR gate)
  - some form of memory (LUT)

What should be used as a logic block for QCA FPGA? (continued)

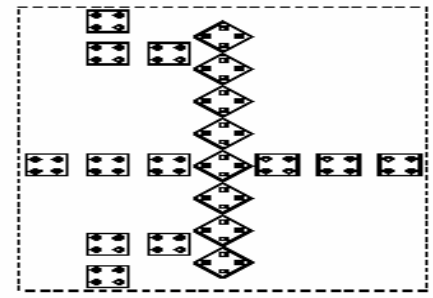
- Majority gate (AND/OR):
  - no inverter
  - needs to be conditionally programmed, makes things difficult
- Memory (LUT):
  - functionally complete
  - needs to be conditionally programmed, makes things difficult
- Verdict: choose single NAND gate
  - don't need to worry about programming logic blocks, FPGA is configured by programming interconnects

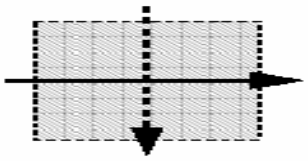
What should be used as interconnects for QCA FPGA?

- Need to take into consideration the same things.
- Most intuitively similar to CMOS pass transistor: an array of programmable multiplexors and selectors
- Issues with this scheme:
  - Complex
  - Takes up a ton of room
- Need to device a more efficient routing mechanism.
  - Take advantage of QCA clocking scheme, only apply clocks to wanted connection

What should be used as interconnects for QCA FPGA? (continued)

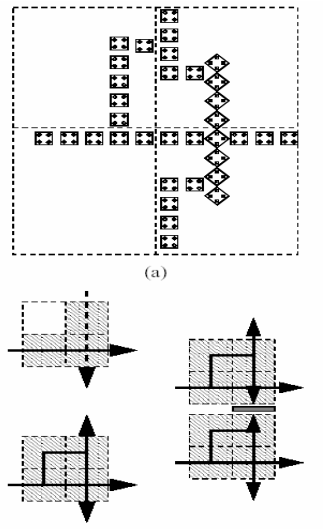
- Base-line routing element – cross two QCA wires in a single clocking region:
  - not flexible, either on or off
  - can't turn corners, no fanout





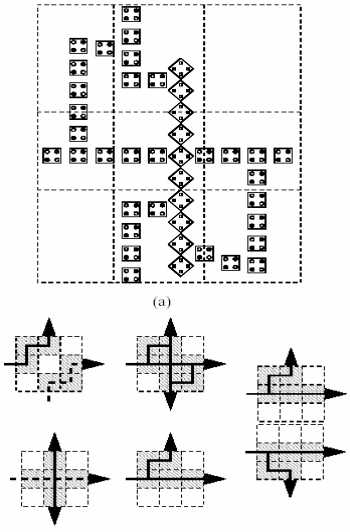
What should be used as interconnects for QCA FPGA? (continued)

- 2x2 routing element – 4 clocking regions in a 2x2 pattern:
  - more flexible
  - supports fanout
  - switch signal is not terminated



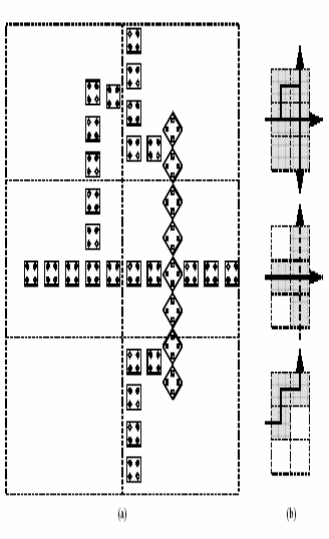
What should be used as interconnects for QCA FPGA? (continued)

- 3x3 routing element – 9 clocking regions in a 3x3 pattern:
  - very flexible
  - allows termination of switched signals
  - 125% expansion in area



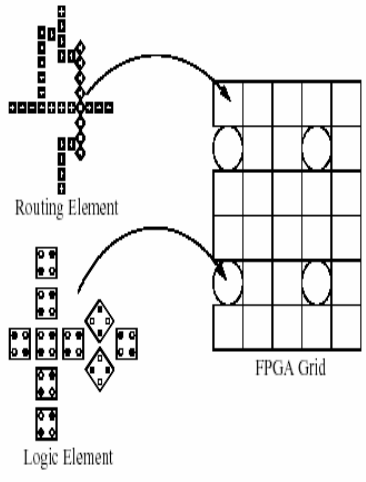
What should be used as interconnects for QCA FPGA? (continued)

- 3x2 routing element – 6 clocking regions in 3x2 pattern:
  - Compromise between flexibility of 3x3 element and size of 2x2 element



What would QCA FPGA look like?

- Place and route FPGA by applying clocks to desired regions



Conclusion

- Construct QCA wire loops from routing elements storage (i.e. flip-flops)
- Succeeded in designing the first QCA-based FPGA

Observations:

- Storage devices will take burn a big part of the chip
- Doesn't actually talk about clock distribution
- Wires are uni-directional
- Wires that are pipelined screws everything up
  - Mentions placing cells in a single clocking zone to eliminate pipelining but this brings up a bunch of other issues, does not address those issues, then uses it same technique to create adder