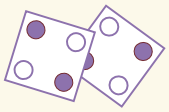


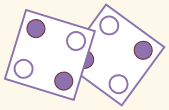
# A Potentially Implementable FPGA for Quantum-Dot Cellular Automata

Michael T. Niemier, University of Notre Dame
Arun Rodrigues, University of Notre Dame
Peter Kogge, University of Notre Dame
Special Thanks to: NSF



Non-Silicon Computing Workshop is Association with HPCA – February 3, 2002

## Motivation

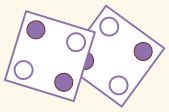


- A Problem: CMOS Limitations
  - Moore's Law: 2x transistor gain every 3 years in CMOS
  - Moore's Law Obstacles: Dopants, quantum effects, $$$
- A Solution: QCA
  - QCA = another way to do computation – but quantumly
  - QCA device = 4 quantum dots positioned in a square
- A Design Target: A QCA FPGA

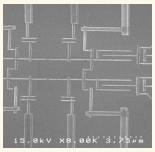
  - Previous designs have been "custom"
  - Near term focus: not the fastest, densest circuit
  - Instead, develop work for implementation experiment.
  - Simple, regular structures are desired (as they might target best for self-assembly

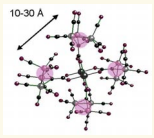
University of Notre Dame

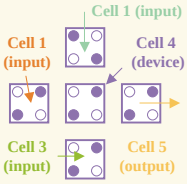
# Where we've been

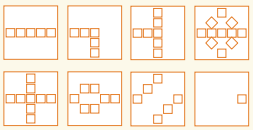


Early QCA work mainly devices

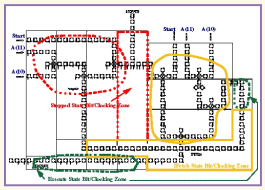




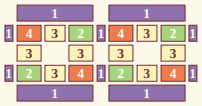



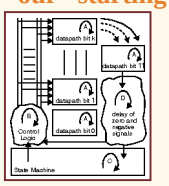



Progressed to simple circuits, architectures – our "starting point"



Simple circuits, µP, and architectures next logical step





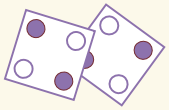

More complex circuits, control logic, and µArchitecture work followed

Now, we move on to something that can be built in near-term

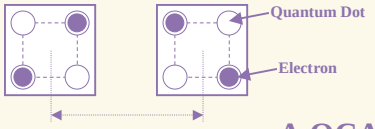


Non-Silicon Computing Workshop is Association with HPCA – February 3, 2002

# The Basics

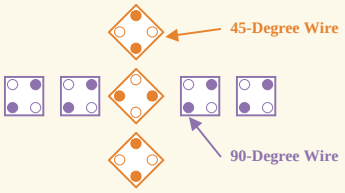


## The Device




Proposed experiment – 42 *nm*
Future molecular – 4.2 *nm*
(and room temperature)

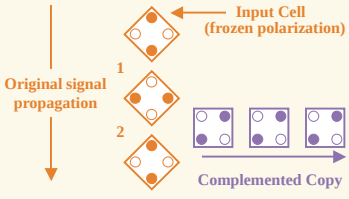
## Wire Cross in the Plane



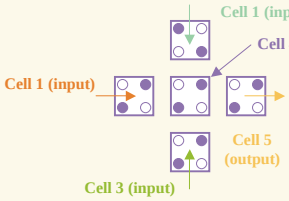

## A QCA Wire

Signal Propagation Direction

## A 45-degree Wire

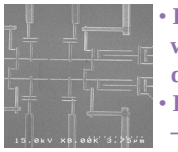



## Majority Gate




University of Notre Dame

# Experimental QCA



- Early and present work uses "metal" dots
- Low temperatures – 70 mK

A. Orlov, I. Amlani, G. Bernstein, C. Lent, G. Snider, "Realization Of a functional cell for quantum-dot cellular automata." Science, 277:928-930, 1997.

- 3-input majority logic gate demonstrated

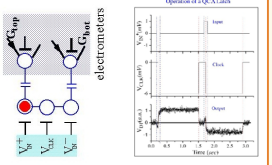



I. Amlani, A. Orlov, G. Toth, G. Bernstein, C. Lent, G. Snider, "Digital logic gate using quantum-dot cellular automata." Science, 284: 289-291, 1999.

- QCA wire demonstrated

A. Orlov, I. Amlani, C. Lent, G. Bernstein, G. Snider, "Experimental demonstration of a binary wire for quantum-dot cellular automata." Applied Physics Letters, 74: 2875-77, 1999.

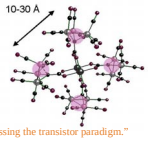
- Single-bit memory demoed




A. Orlov, R. Kummamuru, R. Ramasubramaniam, G. Toth, C. Lent, G. Bernstein, G. Snider, "Experimental demonstration of a latch in Clocked quantum-dot cellular automata: Review and recent Experiments." J. of Appl. Physics, 85: 4283-85, 1999

- Clocked QCA cells demonstrated
- Power gain demoed
- Work underway to raise operating temp.

- QCA with chemical molecules



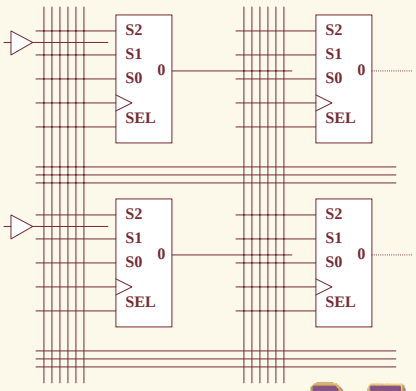

C. Lent, "Molecular electronics: Bypassing the transistor paradigm." Science, 288:1597-1599, 2000.



Non-Silicon Computing Workshop is Association with HPCA – February 3, 2002

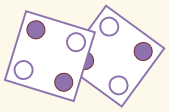
# FPGA Properties

- Generically, an FPGA = a collection of functionally complete logic elements arranged in some interconnection framework
- A common, unique feature is a pattern of horizontal and vertical wires with programmable connections for data routing…
- Logic blocks can be connected directly – direct interconnection…
- …or via long-line interconnect wires that bypass logic blocks to move a signal "far away"




University of Notre Dame

# The QCA Clock



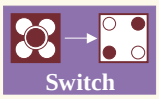
## The CMOS Clock

- ✓ The standard CMOS clock is signal controlling memory transfers
- ✓ The CMOS clock is 2 phases

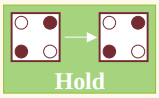
## The QCA Clock

- ✓ QCA clock not a wire or port
- ✓ QCA "clock" an E-field controlling barriers, suppressing $e^-$ tunneling
- ✓ E-field = 4 phase clock

## The QCA Clock Phases




- ✓ Cells begin unpolarized
- ✓ Barriers raised, cells "latched"




- ✓ Barriers are held high
- ✓ Used as input to next zone

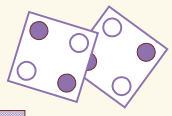

- ✓ Barriers are lowered
- ✓ Cells relax to unpolarized state

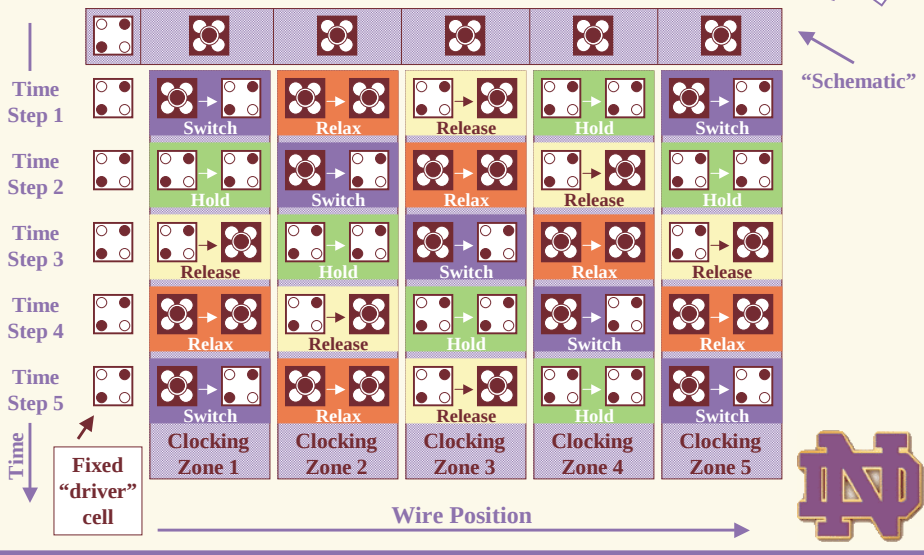

- ✓ Cell barriers remain lowered
- ✓ Unpolarized, neutral state stays



Non-Silicon Computing Workshop is Association with HPCA – February 3, 2002

# A Clocking Example

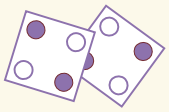




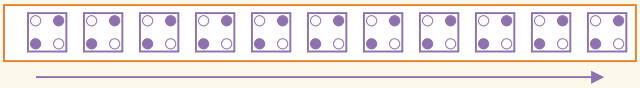



University of Notre Dame

# FPGAs in QCA



- The QCA clock scheme results in "inherent self latching" for data movement (i.e. a shift register)
- Information transmission is "pipelined" and not instantaneous (unlike CMOS electron flow)
- Must coordinate arrival times of signals to logic
- Alternative: One big wire in one big clocking zone

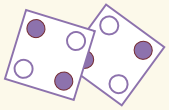


- Why the alternative doesn't work:
  - As a QCA wire grows in length, the probability that all cells will switch successfully decreases

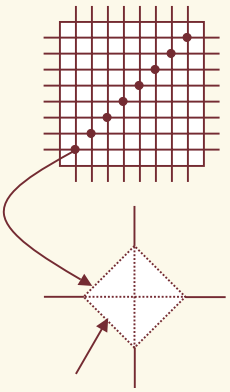


Non-Silicon Computing Workshop is Association with HPCA – February 3, 2002

# Switching Matrices and QCA



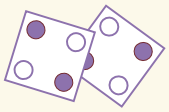
- Most useful feature of CMOS FPGA is an easy means for general purpose interconnect
- Common method:
  - Grid of metal lines (switching matrix)
  - Junctions = network of pass transistors
  - North, South, East, or West movement
- Pass transistors in CMOS allow current (information) to flow between *a* and *b*
- But in QCA information is moved by *nearness*, not $e^-$ flow
- No way for pass transistor of only QCA devices



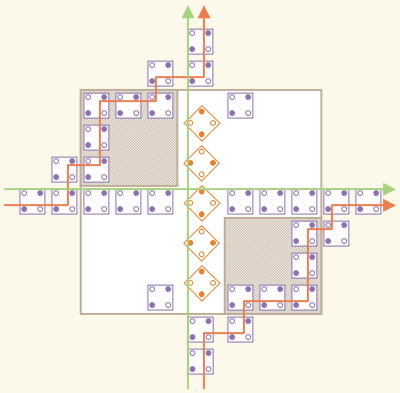
Pass Transistor

University of Notre Dame

# The QCA Clock and Pass Transistors



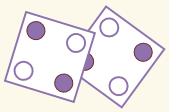
- The *relax* clock phase may be the key to the QCA "pass transistor"
- In the *relax* phase QCA cells remain unpolarized so they don't influence computation
- Pass-transistor-esq routing may be accomplished by using the clock to selectively "turn off" groups of QCA cells to create switches
- Interestingly, a similar technique may be useful to store data
  - However, the *hold* phase would be used to keep data local





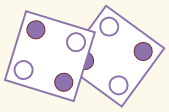


# Logic Block Candidates



- 3 basic logic blocks considered for initial QCA FPGA:
  - A block with a single NAND or NOR gate
  - A block with a programmable majority gate (AND/OR)
  - A block with some form of memory and logic
- Programmable majority gate = sea of AND/OR gates – but no functionally complete logic block!
- Block with memory is nice too – but with it (and option 2) programming signal must be routed/stored
- Basic NAND gate logic block chosen:
  - Its functionally complete
  - Data routing handles all programming
  - This is first cut so let's keep it simple, huh?

# An Introduction to Interconnect

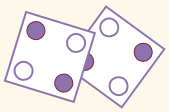


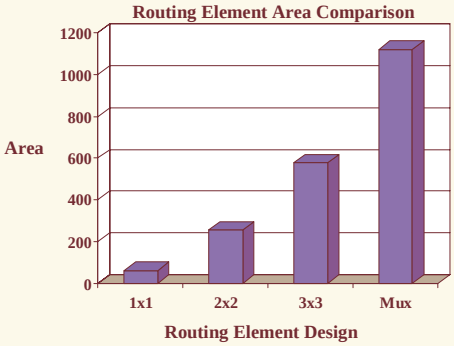
- Again, in CMOS we usually use SRAM memory to configure pass transistors
  - But in QCA there are no pass transistors…at least that can be made exclusively of QCA devices…
  - …and we're a long way away from a physical SRAM…
  - …and we want design simplicity
- What about multiplexors?
  - They might offer the most direct translation of CMOS routing techniques to QCA
  - Simple 2x1 multiplexor/1x2selector could be a pass transistor

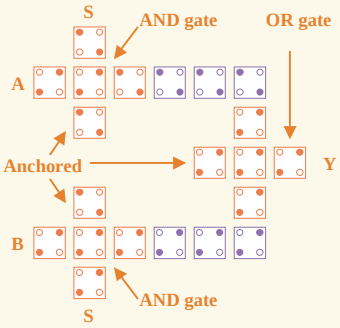




# Multiplexors – and why they're bad

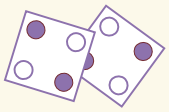




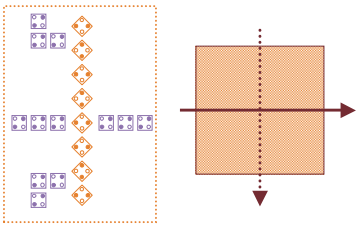




- Multiplexors require would take at least 6 majority gates!
  - This is much bigger than the NAND gate logic block!!!
- Also, programming signals would have to be stored/routed

# Routing Element Choices

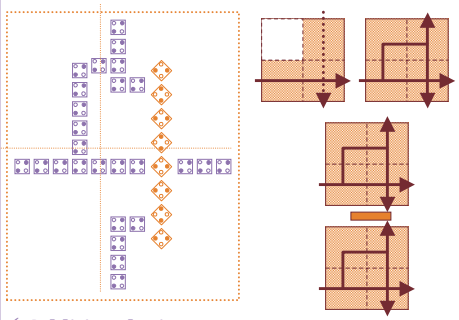


## 1 clock zone routing element



- ✓ 1 clock zone element allows 2 signals to cross…
- ✓ …but is ineffective as a routing element as signals cannot be conditionally directed

## A 2x2 zone routing element

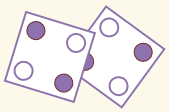


- ✓ Additional wire segment can be clocked or unclocked
- ✓ Allows for perpendicular crossing or a "corner turn"
- ✓ Problem: signal collision

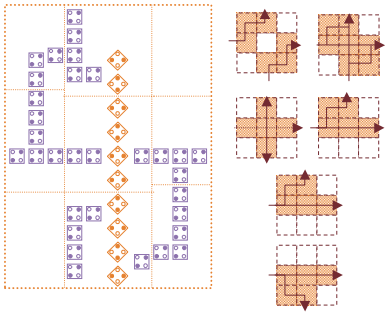


Non-Silicon Computing Workshop is Association with HPCA – February 3, 2002

# Routing Element Choices

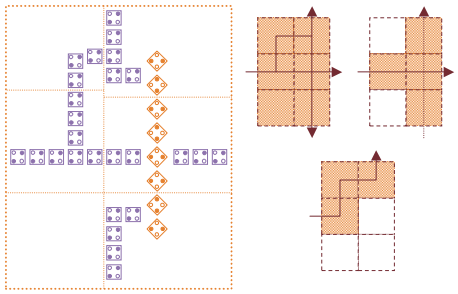


## A 3x3 zone routing element



- ✓ Can alleviate "signal collision" with unclocked zone/buffer
- ✓ Allows increased routing flexibility
- ✓ Problem: area increases by 125%

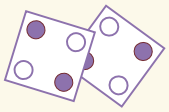
## A 3x2 zone routing element



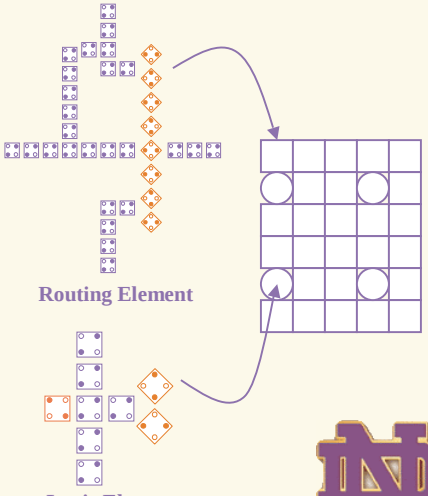
- ✓ A compromise b/t the 2x2 and 3x3 routing element
- ✓ Allows flexible routing options, a reasonable size, and no signal collisions

University of Notre Dame

# A Complete QCA FPGA



- Routing and logic elements previously discussed are modular – can easily be connected to form complete FPGAs
- CMOS place and routing techniques should still be applicable
- For a physical device:
  - Logic element would probably be scaled to the routing element size
  - Routing element might be scaled to minimum clocking zone width size

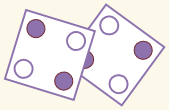




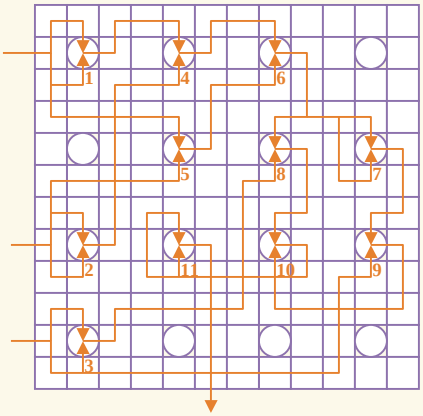


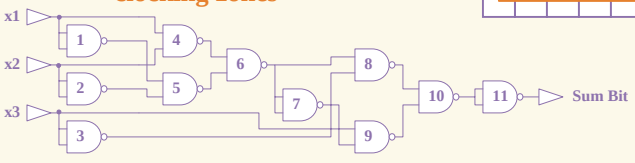



# A NAND Adder

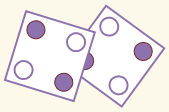


- Most processors need an ALU (and an adder is part of that)
- Circles represent logic blocks, squares – routing elements
- Routing paths always = multiple of 4 squares (or clocking zones)
- Some routing paths not equal
  - But, can program by the clock and create "larger" clocking zones

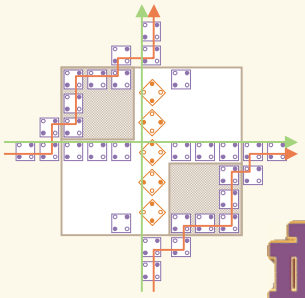



# Conclusions



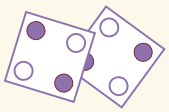
- 1st QCA-based FPGA designed
- Circuit components are simple, regular
  - Good candidates for future experimental work
- Potential schemes for implementing QCA switches have been devised
- While our FPGA is simple…
  - Placing and routing is possible
  - Programming sophisticated circuits is possible





Non-Silicon Computing Workshop is Association with HPCA – February 3, 2002

# Future Work



- Logical completeness not the only requirement for a "useful" FPGA
  - The ability to store state would also help
- Lack of direct QCA flip-flop equivalent makes this difficult
  - But, data could be stored by creating/programming a QCA wire loop
- Work should continue with the technologists to move to the implementable
- Continue work with "QCA pass transistor"
  - Is there another way to create a similar element?
  - Might it be useful in other applications?

University of Notre Dame