QCA Layout Generator

Computação Quântica: arquitecturas e simulação de operação de dispositivos

Geração automática de Layout QCA para circuitos combinatórios

Tiago Teresa Teodósio

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Júri
Presidente: Doutor José António Beltran Gerald
Orientador: Doutor Leonel Augusto Pires Seabra de Sousa
Vogal: Doutor Paulo Ferreira Godinho Flores

Dezembro de 2007

QCA Layout Generator

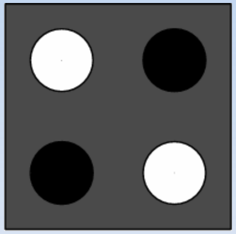
Summary

- QCA Theory
- QCA Logic Gates
- QCA Synchronization
- Layout Generation
  - Expansion
  - Placement
  - Shaping
  - Routing
  - Synchronization
  - Signal Distribution
- Implementation
- Generated Layouts
- Conclusions

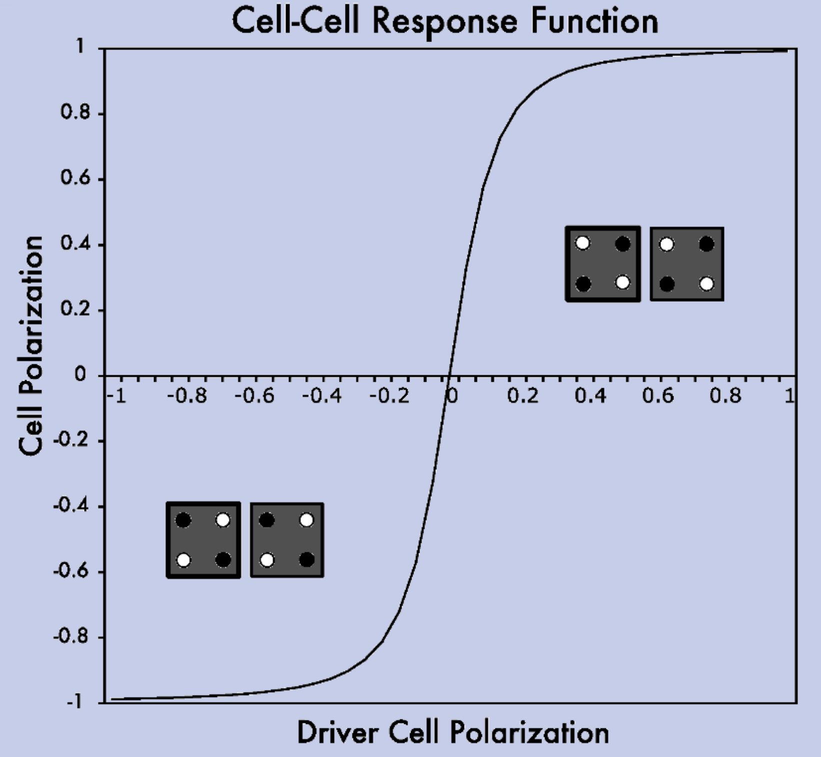
QCA Layout Generator

QCA Theory * QCA Logic Gates * QCA Synchronization
Layout Generation (Expansion, Placement, Shaping, Routing, Synchronization, Signal Distribution)
Implementation * Generated Layouts * Conclusions

The basic four-dot QCA cell is a square charge container with four quantum dot locations at the corners. Each cell has two electrons in excess, localized on antipodal quantum dots due to the Coulomb repulsion. White circles represent unoccupied dots while a black circle denotes that the dot is occupied by an electron. These electrons are only allowed to tunnel between quantum dots inside a cell, and therefore, only two different charge distributions are possible in a cell. Although these two states of an isolated cell are equal in energy, they are observable and can be used to encode the logical levels ’0’ and ’1’ of a bit.



QCA Cell



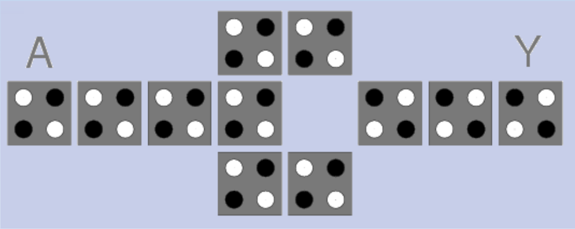
Electrical energy minimization drives the free cell to the polarization of the neighbor cells.

This is the working principle of the three input majority gate, where the sum of the electrical field from all three inputs is used to drive the central cell.

The signal inversion is based on the same principle, but with a different geometric arrangement of the driver cells.

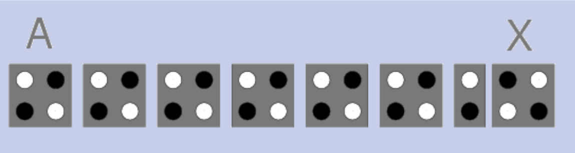
QCA Layout Generator

QCA Theory * **QCA Logic Gates** * QCA Synchronization
Layout Generation (Expansion, Placement, Shaping, Routing, Synchronization, Signal Distribution)
Implementation * Generated Layouts * Conclusions



QCA Inverter Gate

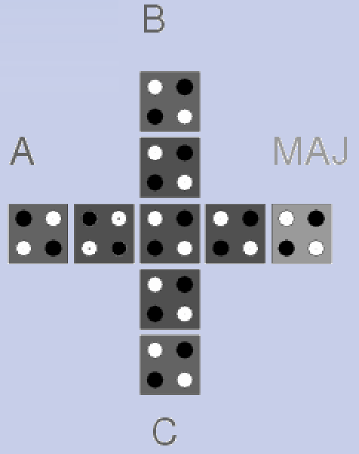
All logic functions can be implemented using only majority gates and inverters, once a majority gate can be used to perform AND and OR logic operations.
MAJ(A, B, C) = A.B + B.C + A.C
MAJ(A, B, ’0’) = AND(A, B)
MAJ(A, B, ’1’) = OR(A, B)



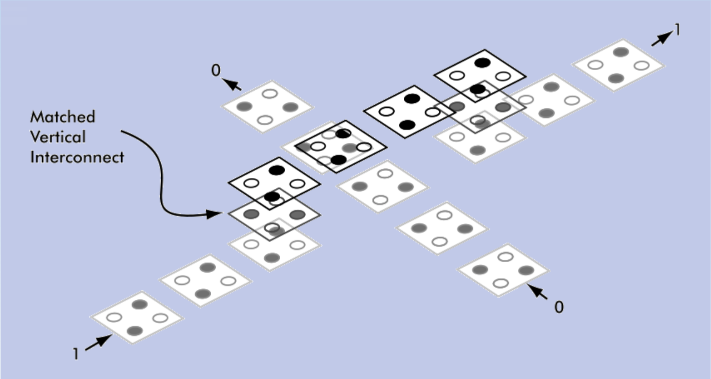
QCA “in wire” Inverter using one QCA half-cell.

Both logic gates and interconnection wires are made of QCA cells.

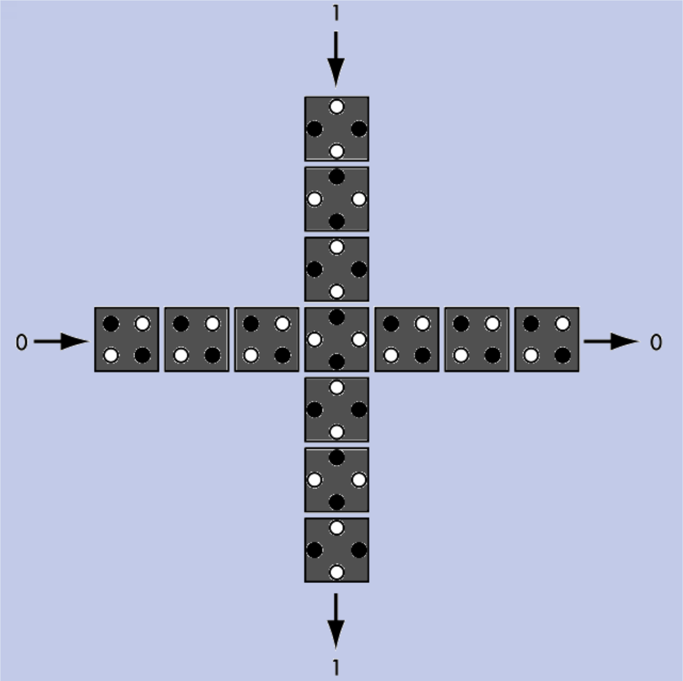
Two alternatives exist to perform wire crossover: multi layer and planar.



QCA Majority Gate



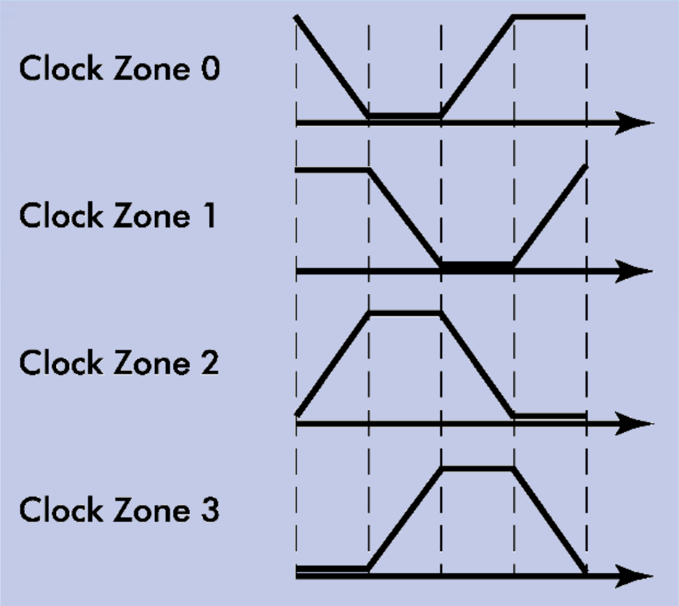
Multi layer crossover



Planar crossover

QCA Layout Generator

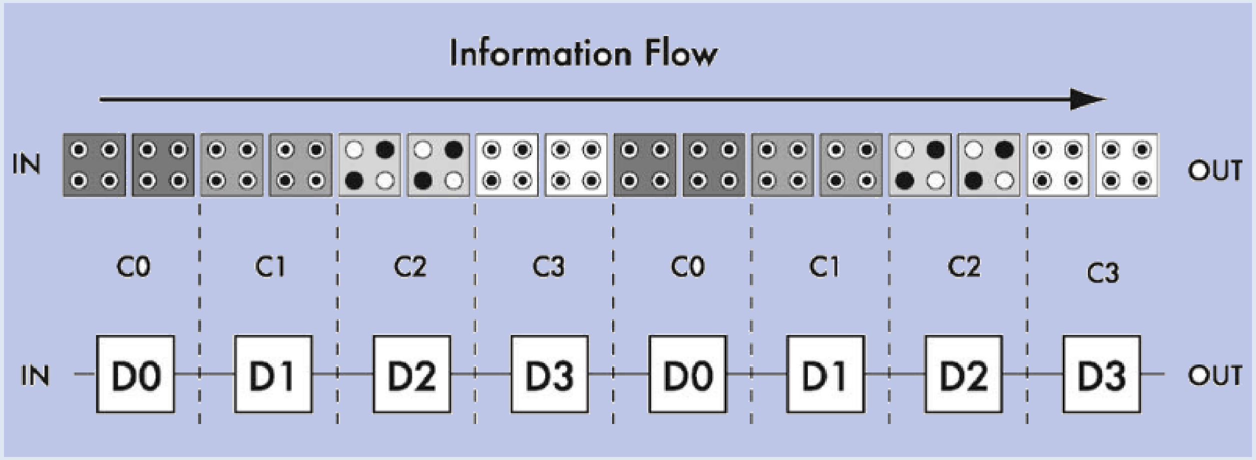
QCA Theory * QCA Logic Gates * **QCA Synchronization**
Layout Generation (Expansion, Placement, Shaping, Routing, Synchronization, Signal Distribution)
Implementation * Generated Layouts * Conclusions



QCA clock signals.

The QCA cell is controlled by a clock signal that allows/inhibits electron tunneling between quantum dots, thus freezing the cell by moments to strongly drive other cells, and then letting it be driven by other cells.

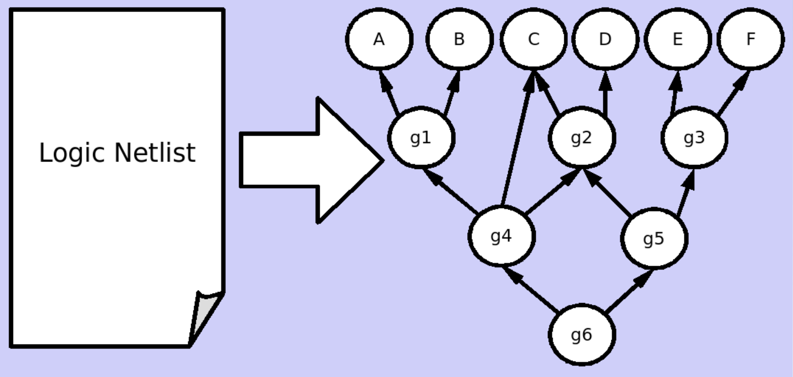
QCA synchronization system is based in four clock signal of the same frequency but dephased one quarter period between them. Therefore, signals can be pumped through QCA wires.



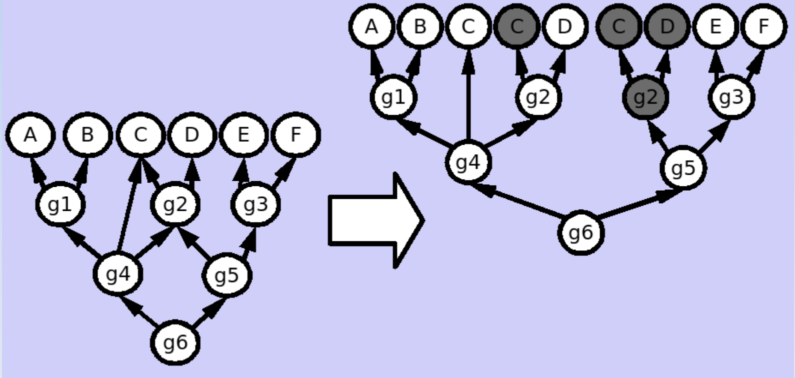
QCA wire split into clock zones.

QCA Layout Generator

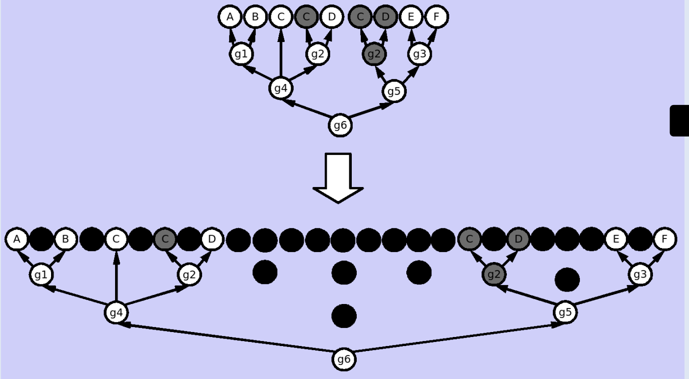
QCA Theory * QCA Logic Gates * QCA Synchronization
Layout Generation (Expansion, Placement, Shaping, Routing, Synchronization, Signal Distribution)
Implementation * Generated Layouts * Conclusions



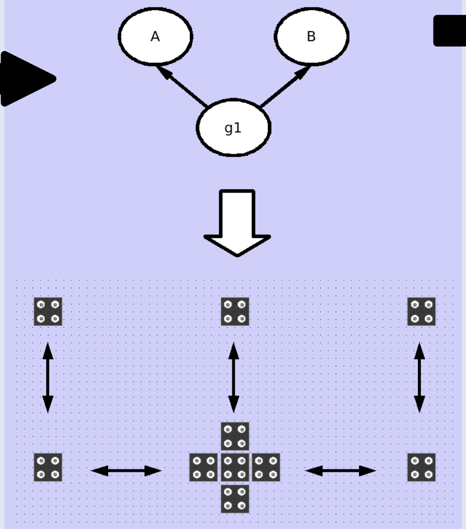
Read the input logic netlist and create a directed graph.



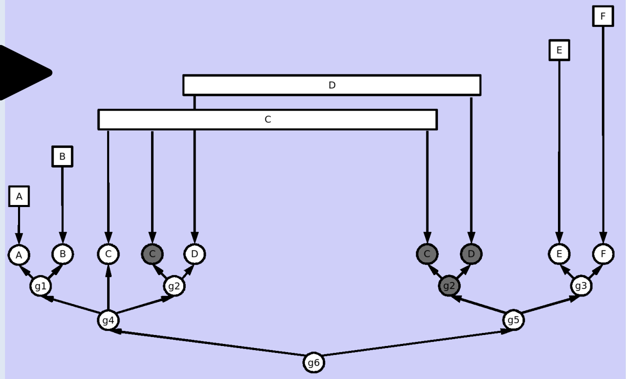
Expand the circuit by replication of the shared nodes.



Determine the (x,y) coordinates of each gate, (black circles represent wasted space).

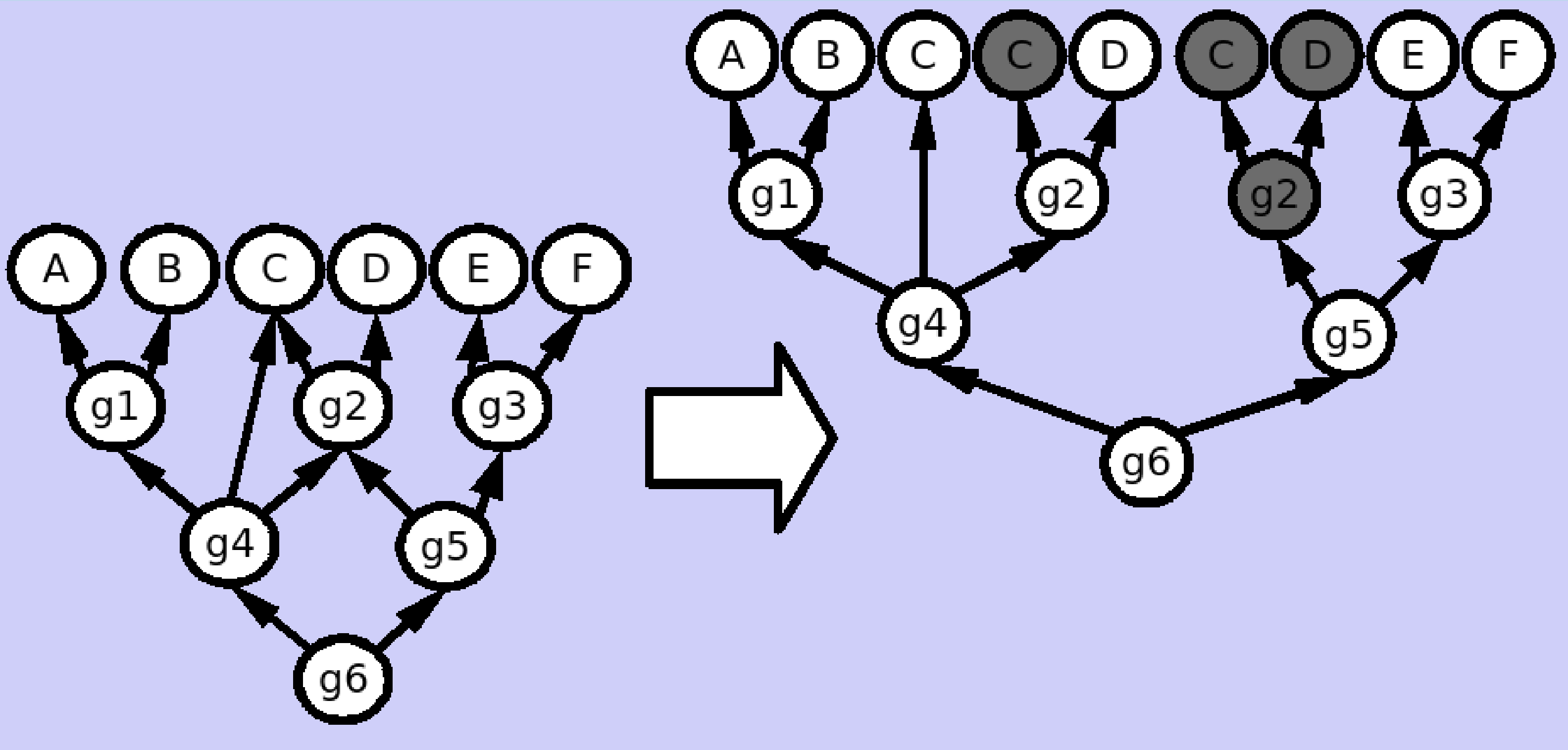


Determine the shape of each gate and implicitly route signals from inputs.



Create distribution wires for the shared inputs and perform delay equalization.

QCA Layout Generator



This operation makes the place and route task easier, avoiding wire crossing at the expense of an increase in the circuit area.

The method applied for replication consists on performing breadth-first exploration of the circuit, starting from each primary output towards the inputs.

Every time a node is revisited it is duplicated, as well as all the nodes included in the sub tree rooted at it.

QCA Layout Generator

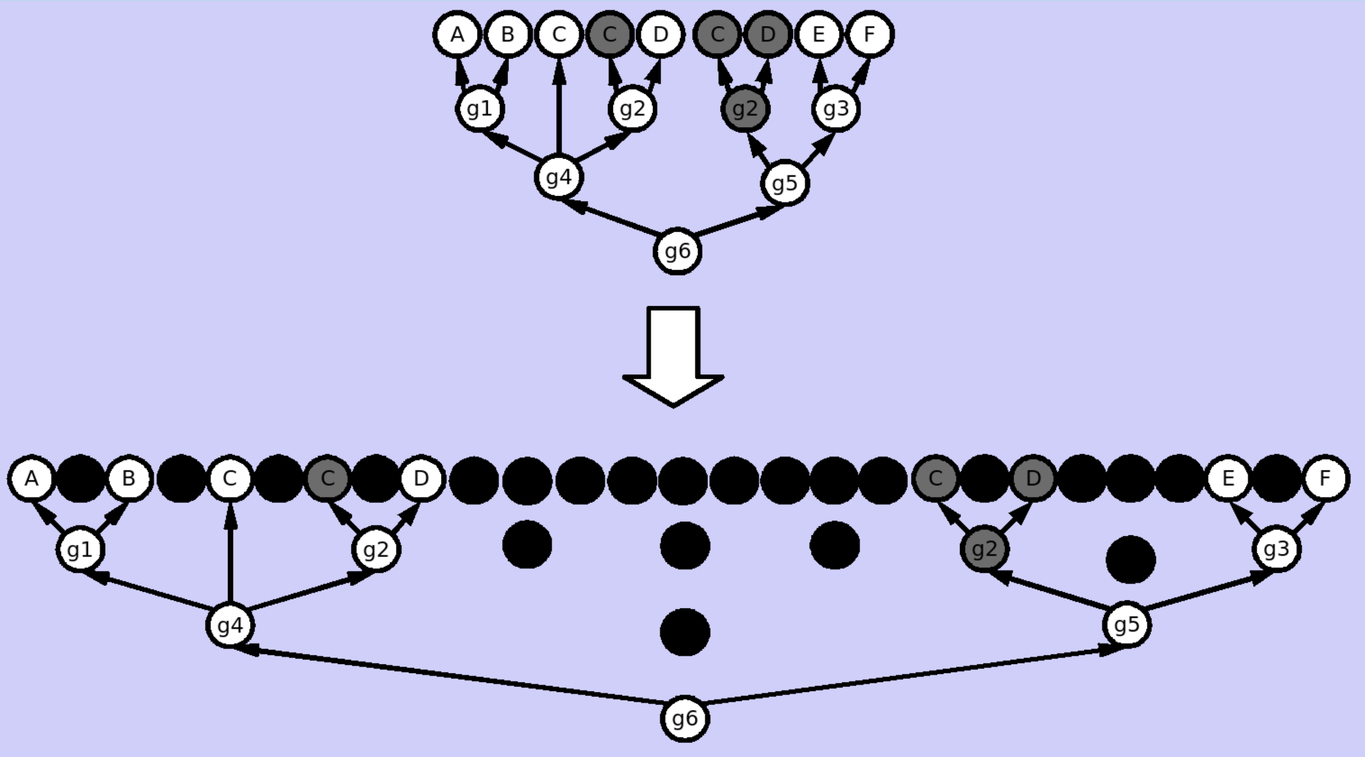
QCA Theory * QCA Logic Gates * QCA Synchronization

Layout Generation (Expansion, **Placement**, Shaping, Routing, Synchronization, Signal Distribution)

Implementation * Generated Layouts * Conclusions

Three different referentials are considered:

- gate level coordinates,
- cell level coordinates,
- physical coordinates.



The determination of the coordinates where each gate will be placed is done in three phases.

1 – Determine the level in the graph for each gate. The graph level is the y coordinate in gate level coordinates, and the maximum level found indicates the maximum y coordinate, called ymax .

2 – Assign different numbers to the gates within each level. When visiting a given gate, with ngate as number, its inputs receive a number determined as follows:

- n = 3 ∗ ngate, for the left input;
- n = 3 ∗ ngate + 1, for the central input;
- n = 3 ∗ ngate + 2, for the right input.

3 – Calculate the x coordinate of the gate level coordinates.

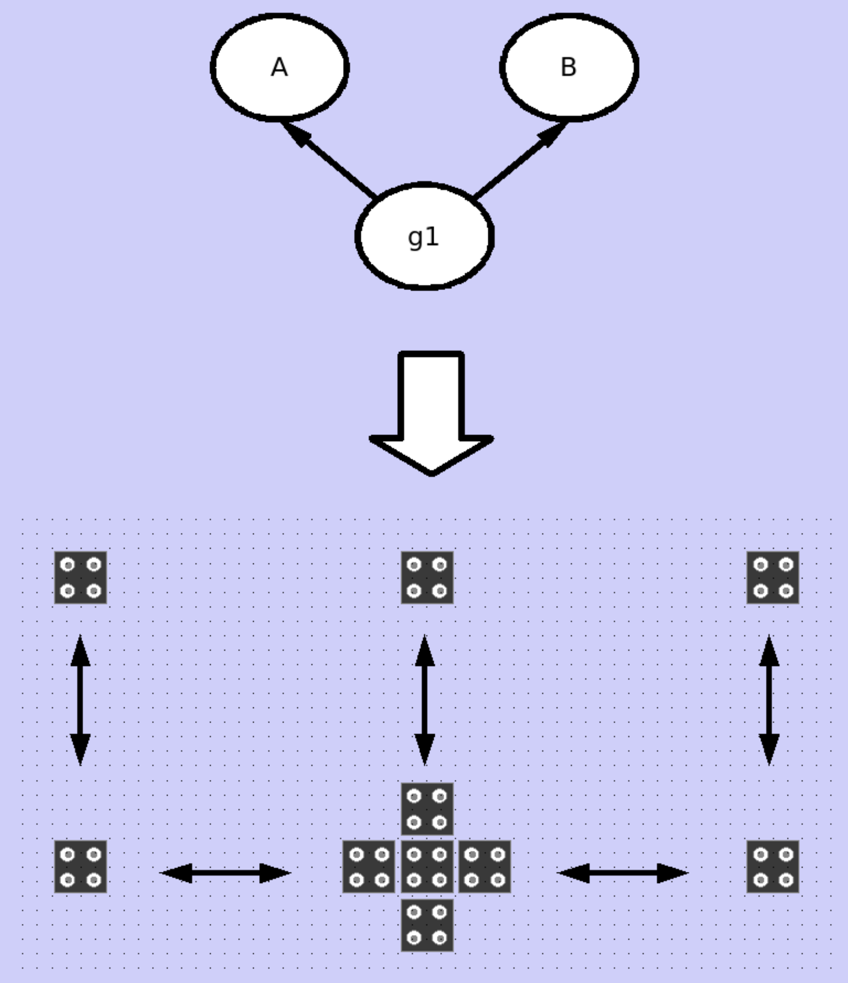
$$x = (1+2n)\frac{(3^{y_{max}-y}+1)}{2} - n$$

QCA Layout Generator

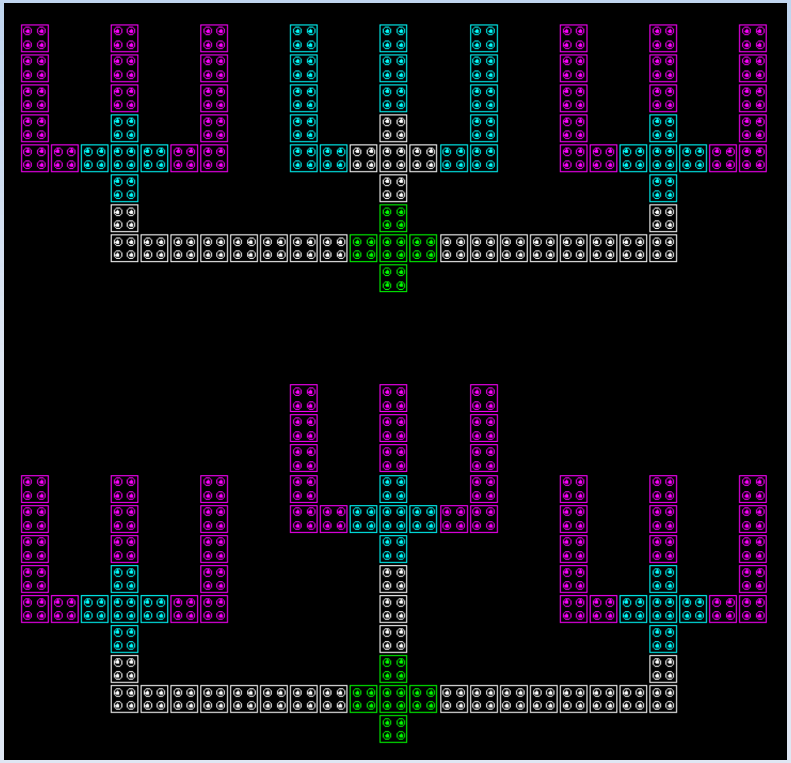
QCA Theory * QCA Logic Gates * QCA Synchronization
Layout Generation (Expansion, Placement, **Shaping**, Routing, Synchronization, Signal Distribution)
Implementation * Generated Layouts * Conclusions

Mapping from the gate level referential to the cell level referential.

x_cell = (x_gate + x_gate_offset)*x_factor, x_factor = 3 cells
y_cell = (y_gate)*y_factor + y_cell_offset, y_factor = 6 cells



Build QCA logic gates with QCA cells



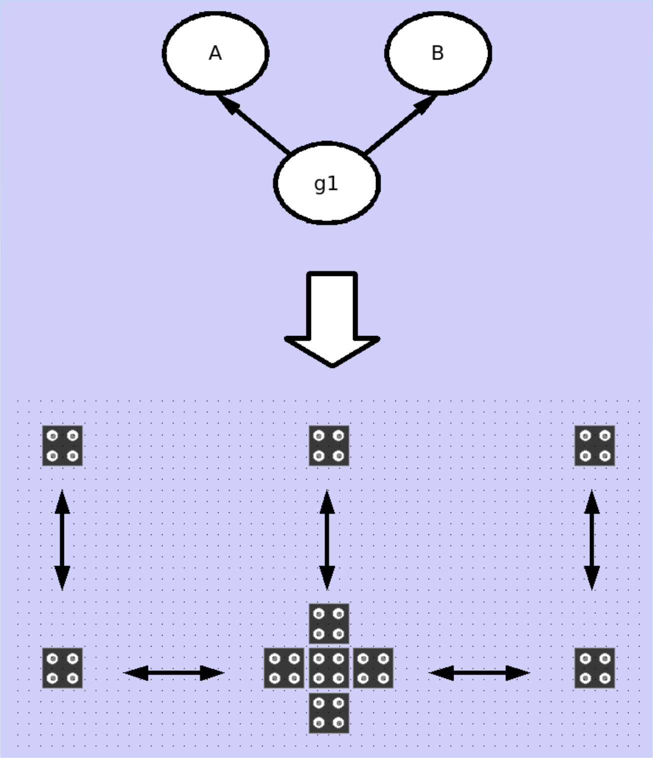
Alternative topologies

Inverter gates will be place in wires. NAND and NOR gates will be treated as AND and OR gates, respectively, where an additional Inverter gate is considered at the gate’s output.

QCA Layout Generator

QCA Theory * QCA Logic Gates * QCA Synchronization
Layout Generation (Expansion, Placement, Shaping, **Routing**, Synchronization, Signal Distribution)
Implementation * Generated Layouts * Conclusions

The input wires of each gate connect directly to the output cells of the gates at its inputs. This way the routing is done implicitly and as a part of gate representation. Input wire length is calculated from both the coordinates of each gate and the coordinates of its inputs.



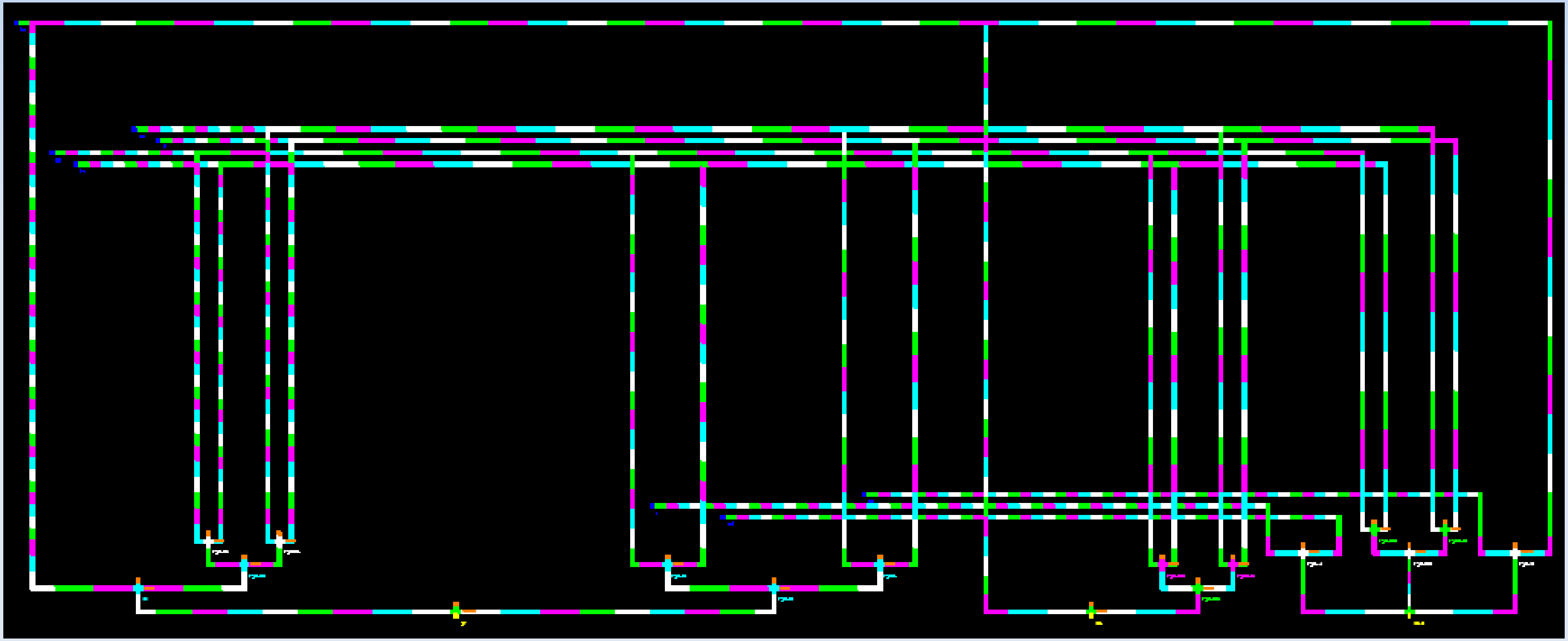
A half QCA cell replaces a full QCA cell when a given input wire is connected to an inverting gate, which means an Inverter, NAND or NOR gate; and this replacement implies some extra space between cells to keep the original wire length.

QCA Layout Generator

QCA Theory * QCA Logic Gates * QCA Synchronization

Layout Generation (Expansion, Placement, Shaping, Routing, **Synchronization**, Signal Distribution)

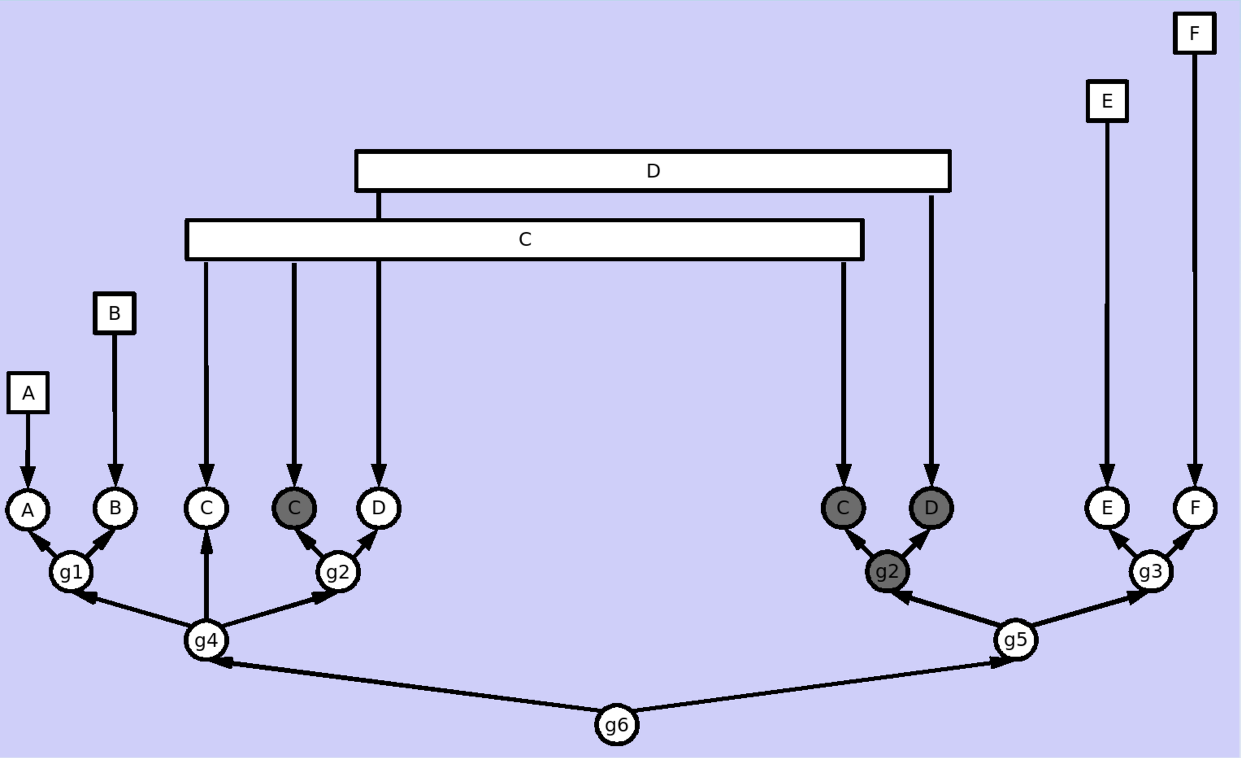
Implementation * Generated Layouts * Conclusions



In order to ensure the proper operation of the circuit it is necessary to set the clock zone of each gate’s output cell as the preceding clock zone of the input cell of the next gate towards the primary outputs.

QCA Layout Generator

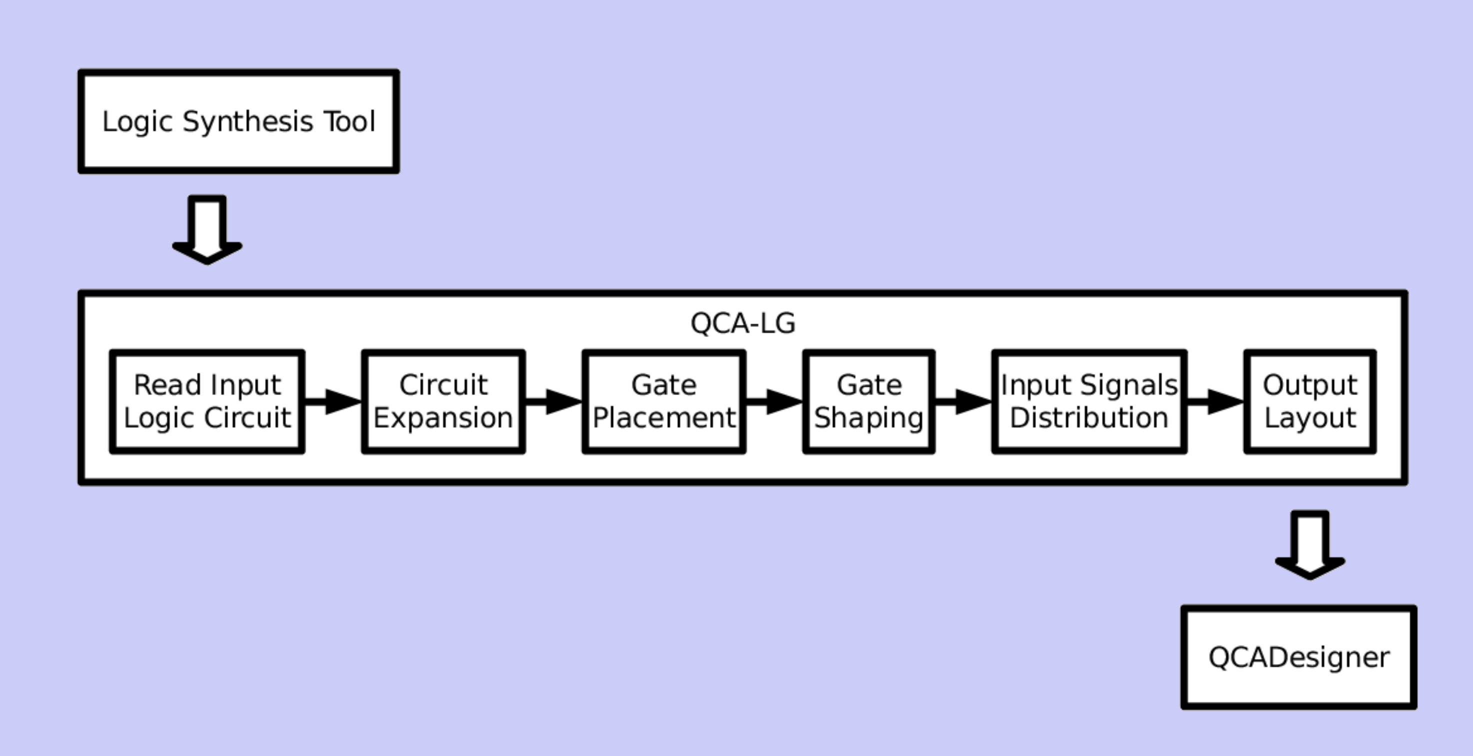
The expansion of the circuit generates copies of some logic gates, which means some input signals must be delivered to some extra points.

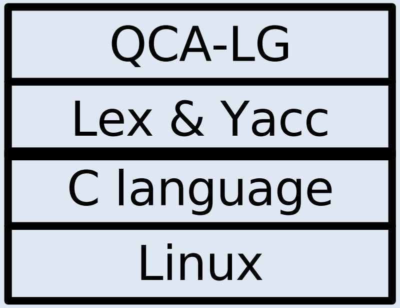


The expression used to evaluate the “urgency” to arrive at a given destination(x,y) at a given moment (clk) is:

$$pri(x, y, clk) = (x - xmin) + (y - ymax) - (clk - clkmin) * Zone$$

QCA Layout Generator

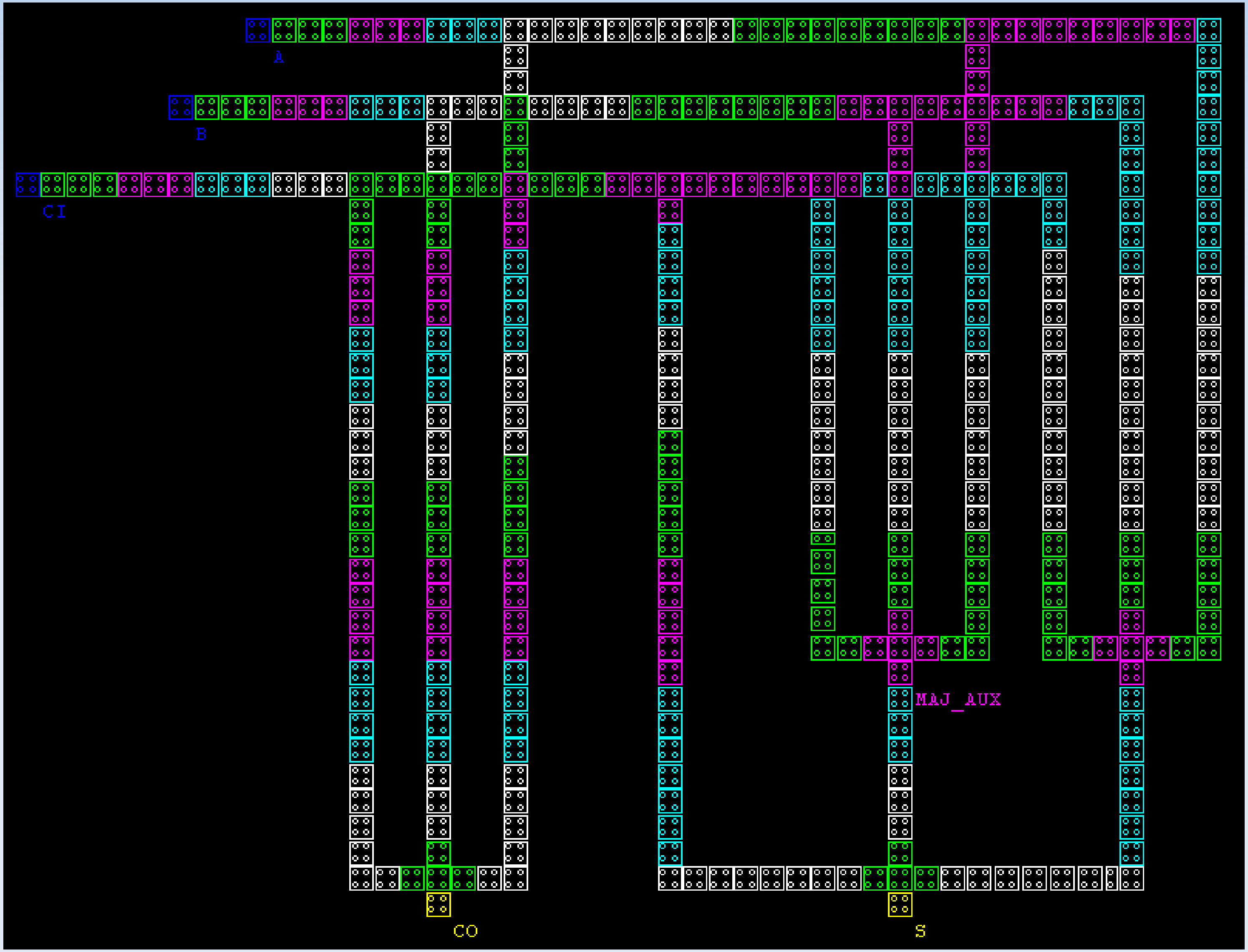




QCA Layout Generator

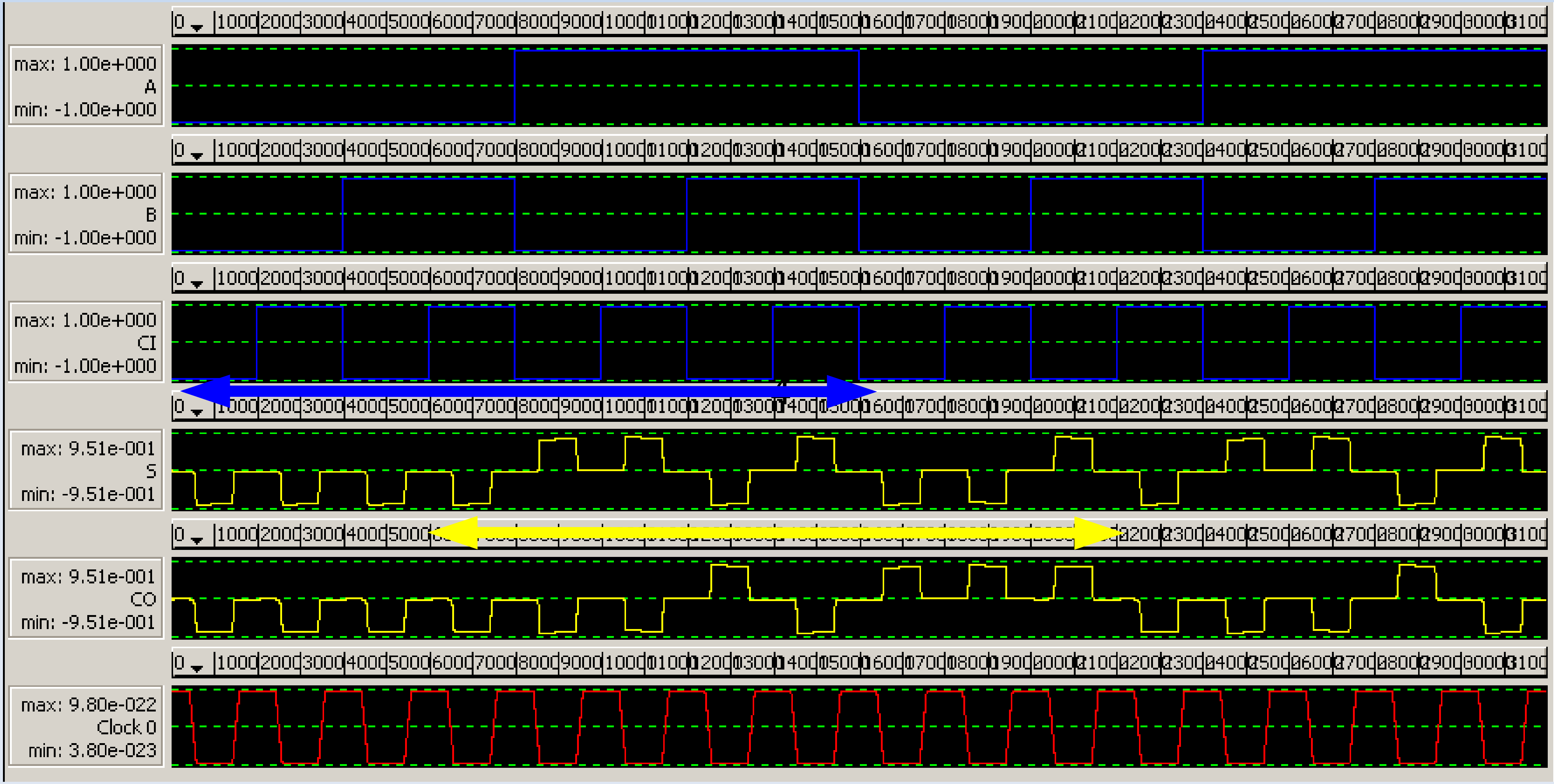
QCA Theory * QCA Logic Gates * QCA Synchronization
Layout Generation (Expansion, Placement, Shaping, Routing, Synchronization, Signal Distribution)
Implementation * **Generated Layouts 1/3** * Conclusions

```
/* LSI code */
COMPILE;
DIRECTORY MASTER;
MODULE ADDER4;
        INPUTS  A,B,CI;
        OUTPUTS  S,CO;
LEVEL FUNCTION;
DEFINE

A = (A);
B = (B);
CI = (CI);
S = (S);
CO = (CO);

i0( N_CI = Z ) =
        INV(CI=A);
m1( CO = Z ) =
        MAJ3(A=A,B=B,CI=C);
i1( N_CO = Z ) =
        INV(CO=A);
m2( MAJ_AUX = Z ) =
        MAJ3(A=A,
                B=B,
                N_CI=C);
m3( S = Z ) =
        MAJ3(N_CO=A,
                MAJ_AUX=B,
                CI=C);

END MODULE;
END COMPILE;
END;
```



Automatically generated 1-bit full adder layout.

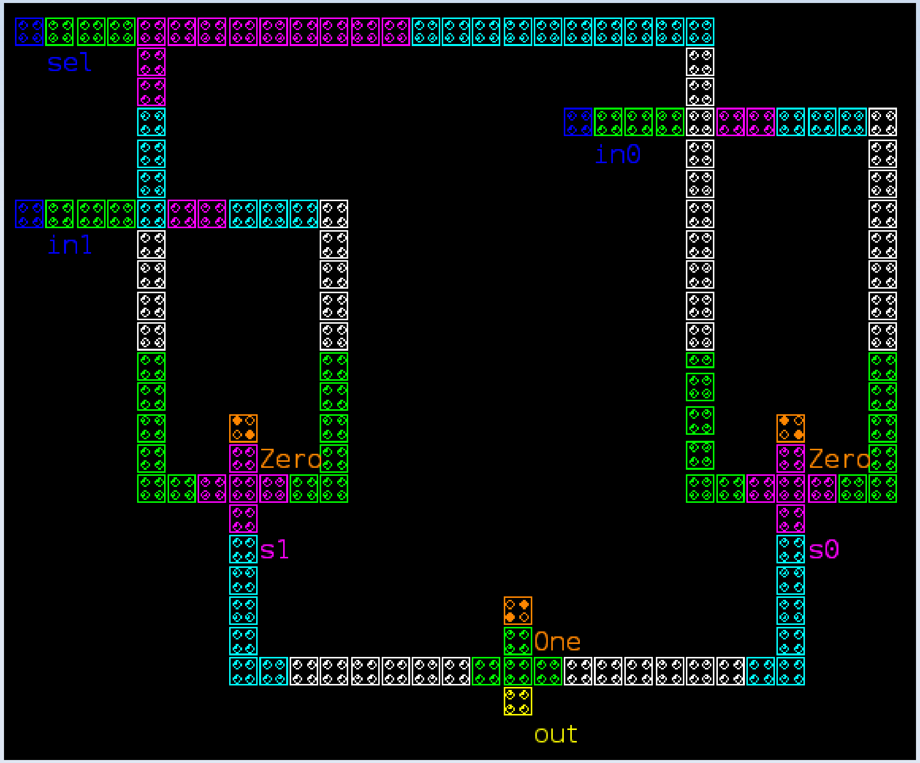
QCA Layout Generator



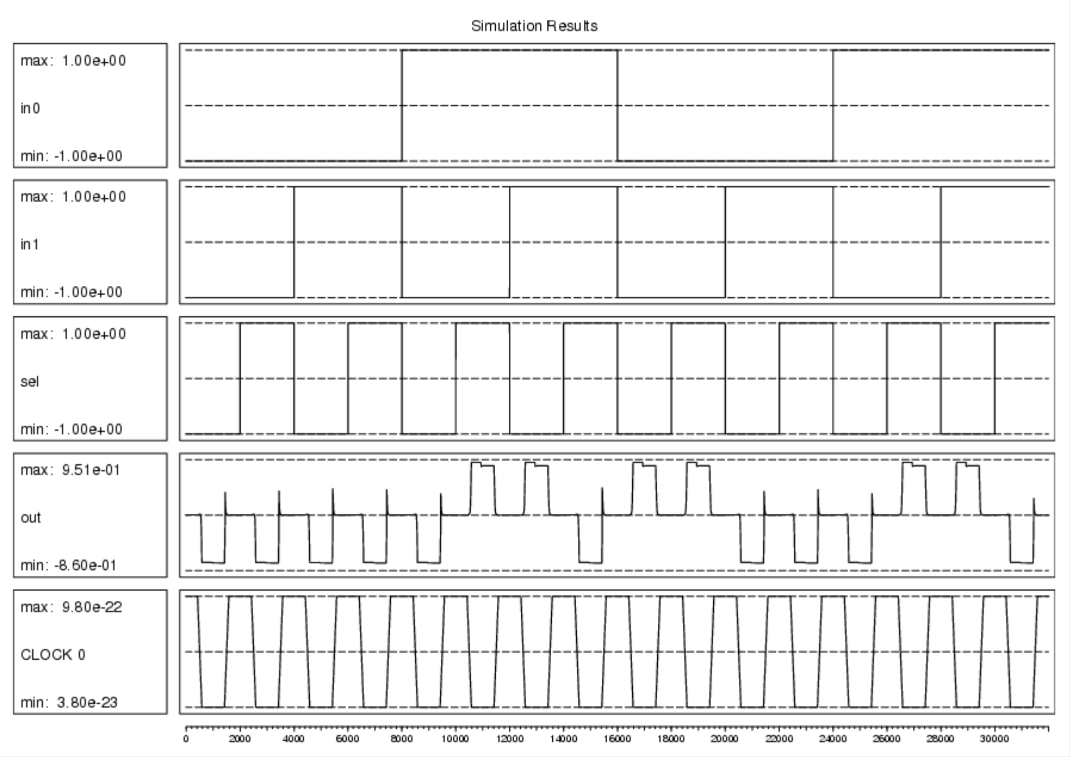
Automatically generated 1-bit full adder simulation results.

QCA Layout Generator

QCA Theory * QCA Logic Gates * QCA Synchronization
Layout Generation (Expansion, Placement, Shaping, Routing, Synchronization, Signal Distribution)
Implementation * **Generated Layouts 3/3** * Conclusions



Automatically generated 1-bit 2:1 multiplexer.



Simulation results for the 1-bit 2:1 multiplexer.

QCA Layout Generator

- QCA allows high throughput and deep pipeline.
- CMOS technology will be needed to bound the real analog world to QCA.
- QCA-LG tool is able to automatically generate layouts for small sized circuits.
- The optimization effort should be focused on the gate’s placement.
- With QCA-LG the design flow for QCA technology is now almost complete.

QCA Layout Generator

INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

The End

inesc id
lisboa