

ALYSON TRINDADE FERNANDES

**UM ALGORITMO DE POSICIONAMENTO E ROTEAMENTO  
DE CÉLULAS QCA NO ESQUEMA DE CLOCK USE**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA  
MINAS GERAIS – BRASIL  
2017

**Ficha catalográfica preparada pela Biblioteca Central da Universidade  
Federal de Viçosa - Câmpus Viçosa**

T

F363a  
2017

Fernandes, Alyson Trindade, 1986-  
Um algoritmo de posicionamento e roteamento de células  
QCA no esquema de clock USE / Alyson Trindade Fernandes. –  
Viçosa, MG, 2017.  
xiii, 58f. : il. (algumas color.) ; 29 cm.

Orientador: Ricardo dos Santos Ferreira.  
Dissertação (mestrado) - Universidade Federal de Viçosa.  
Inclui bibliografia.

1. Circuitos lógicos. 2. Autômato celular. 3. Pontos  
quânticos. I. Universidade Federal de Viçosa. Departamento de  
Informática. Programa de Pós-Graduação em Ciência da  
Computação. II. Título.


CDD 22. ed. 621.395


ALYSON TRINDADE FERNANDES

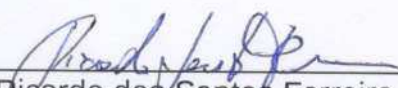
**UM ALGORITMO DE POSICIONAMENTO E ROTEAMENTO DE  
CÉLULAS QCA NO ESQUEMA DE CLOCK USE**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 02 de agosto de 2017.

  
\_\_\_\_\_  
José Augusto Miranda Nacif

  
\_\_\_\_\_  
Omar Paranaíba Vilela Neto

  
\_\_\_\_\_  
Ricardo dos Santos Ferreira  
(Orientador)

*“A vida é e sempre continuará a ser  
uma equação sem solução,  
mas contém alguns fatores conhecidos ...”*

Nikola Tesla

# Agradecimentos

Agradeço primeiramente a Universidade Federal de Viçosa (UFV) pelo conhecimento adquirido nesta grande etapa de minha vida. Uma história é montada com a inserção de pequenas frases, mas que no final seguem o contexto e formam o conto mais belo.

Agradeço também o Instituto Federal do Norte de Minas Gerais (IFNMG) pelo investimento realizado e pelo convívio com amigos companheiros do dia a dia.

Ao meu orientador Ricardo dos Santos Ferreira, por total apoio durante esta etapa tão desgastante para mim. Foram muitos momentos enriquecedores e de construção de conhecimentos que irei levar para toda uma vida.

Ao professor Omar Paranaíba Vilela Neto e seu orientando Douglas Sales pelo apoio no conhecimento da tecnologia QCA e da ferramenta QCADesigner.

Ao professor José Augusto Nacif e seus orientandos Juliana Rezende e Vinícius Reis pelas horas de estudo e auxílio em etapas importantes para a construção desta defesa.

Por fim, meus agradecimentos à Fapemig, CAPES e CNPQ pelo apoio e incentivo financeiro à pesquisa.

# Sumário

	<b>Lista de ilustrações</b>	<b>vi</b>
	<b>Lista de tabelas</b>	<b>x</b>
	<b>Lista de abreviaturas e siglas</b>	<b>xi</b>
	<b>Resumo</b>	<b>xii</b>
	<b>Abstract</b>	<b>xiii</b>
<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	Motivação	1
1.2	Objetivo	3
1.3	Contribuições da dissertação	4
1.4	Organização da dissertação	4
<b>2</b>	<b>NANOTECNOLOGIA QCA</b>	<b>5</b>
2.1	Introdução	5
2.2	Célula QCA	5
2.3	Estruturas básicas	7
2.3.1	Fio	7
2.3.2	Inversor	7
2.3.3	Porta da Maioria ( <i>Majority Gate</i> )	8
2.3.4	Cruzamento de Fios	10
2.4	Clock QCA	12
2.5	USE	15
2.6	Posicionamento e Roteamento em CMOS e QCA	18
2.7	QCADesigner	19
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>21</b>
3.1	Introdução	21
3.2	Posicionamento	21
3.3	Roteamento	23
3.4	Layout	24
3.5	Comentários Finais	26
<b>4</b>	<b>ALGORITMO DE POSICIONAMENTO E ROTEAMENTO</b>	<b>28</b>
4.1	Formulação do Problema	30

<b>4.2</b>	<b>Mapeamento Tecnológico</b>	<b>31</b>
<b>4.3</b>	<b>Algoritmo</b>	<b>34</b>
4.3.1	Entrada	34
4.3.2	Fluxograma	35
4.3.3	Execução Passo a Passo	36
<b>5</b>	<b>ESTUDO DE CASOS</b>	<b>43</b>
<b>5.1</b>	<b>MUX 2x1</b>	<b>44</b>
<b>5.2</b>	<b>Full Adder de 1-bit</b>	<b>45</b>
5.2.1	<i>XNOR</i>	51
5.2.2	<i>Parity Generator</i>	52
5.2.3	<i>Parity Checker</i>	53
<b>6</b>	<b>CONCLUSÃO</b>	<b>55</b>
	<b>REFERÊNCIAS</b>	<b>56</b>

# Lista de ilustrações

Figura 1.1 – Distribuição da quantidade de transistores nos microprocessadores em relação ao tempo, apresentando a inserção de novas tecnologias. (CAVIN; LUGLI; ZHIRNOV, 2012) . . . . .	2
Figura 1.2 – Evolução no desempenho dos processadores (PRESHING, 2012) . . . . .	3
Figura 2.1 – Estrutura básica de uma célula QCA proposta por (LENT et al., 1993) . . . . .	5
Figura 2.2 – Visão tridimensional do modelo da célula QCA . . . . .	6
Figura 2.3 – Representação de fios em QCA. a) Fio reto; b) Fio com curva; c) Fio com <i>fanout</i> (LENT et al., 1993). . . . .	8
Figura 2.4 – Inversor QCA (TOUGAW; LENT, 1994) . . . . .	8
Figura 2.5 – Porta da Maioria . . . . .	9
Figura 2.6 – Tabela verdade da porta da maioria. Entrada A igual a 0 equivale a porta lógica <i>E</i> . Entrada A igual a 1 equivale a porta lógica <i>OU</i> . . . . .	10
Figura 2.7 – Porta da maioria funcionando como <i>E</i> lógico com a entrada A com polaridade induzida. (TOUGAW; LENT, 1994) . . . . .	10
Figura 2.8 – Cruzamento coplanar (SHIN; JEON; YOO, 2013) . . . . .	11
Figura 2.9 – Cruzamento multicamadas . . . . .	11
Figura 2.10–Esquema de clock com quatro estágios. (NIEMIER; KOGGE, 2004) . . . . .	12
Figura 2.11–Exemplo interativo de transmissão de sinal pelas zonas de clock QCA (CAMPOS et al., 2016). . . . .	13
Figura 2.12–A quantidade de células QCA não aumenta a latência do circuito (WALUS; JULLIEN; DIMITROV, 2003) . . . . .	14
Figura 2.13–Exemplo de circuito e forma de onda com a presença de <i>glitch</i> . . . . .	14
Figura 2.14–Circuito da figura 2.13 em QCA . . . . .	15
Figura 2.15–Matriz de células QCA baseada no esquema de clock USE (CAMPOS et al., 2016) . . . . .	16
Figura 2.16–Circuito <i>XOR</i> : a) sem regras do USE, b) com regras do USE . . . . .	17
Figura 2.17–Posicionamento de circuito no esquema de clock USE (CAMPOS et al., 2016) . . . . .	17
Figura 2.18–Modelos de fluxo de processo: a) CMOS, b) QCA. . . . .	18
Figura 2.19–Modelos de fluxo de processo QCA (OTTAVI et al., 2006) . . . . .	19
Figura 2.20–Tela principal do software QCADesigner. . . . .	20

Figura 3.1 – Particionamento, posicionamento e roteamento de um circuito em nível de porta lógica. (a) particionamento do circuito em zonas. (b) posicionamento de células. (c) roteamento global por meio de conexões entre as zonas. (d) roteamento local com conexões dentro das zonas. (LIM; RAVICHANDRAN; NIEMIER, 2005) . . . . .	22
Figura 3.2 – Construção do layout do circuito. Primeiro, o circuito de entrada é particionado em blocos de fios e portas lógicas. Segundo, cada bloco é posicionado em uma matriz bidimensional mantendo as restrições de clock. Terceiro, as células QCA de cada bloco são posicionadas. Quarto, ocorre a interconexão externa e interna de cada bloco. (LIM; RAVICHANDRAN; NIEMIER, 2005) . . . . .	22
Figura 3.3 – Divisão de um circuito lógico em zonas de clock QCA. (ANTONELLI et al., 2004) . . . . .	23
Figura 3.4 – Balanceamento do circuito. (LIM; RAVICHANDRAN; NIEMIER, 2005) . . . . .	23
Figura 3.5 – Posicionamento de blocos de células. (LIM; RAVICHANDRAN; NIEMIER, 2005) . . . . .	24
Figura 3.6 – Expansão do circuito. Vértices cinzas são cópias. (TEODÓSIO; SOUSA, 2007) . . . . .	25
Figura 3.7 – Construção de layout de porta lógica da maioria. (TEODÓSIO; SOUSA, 2007) . . . . .	26
Figura 3.8 – Interconexão de nós duplicados. (TEODÓSIO; SOUSA, 2007) . . . . .	26
Figura 3.9 – Circuito gerado pela ferramenta QCA-LG. (TEODÓSIO; SOUSA, 2007) . . . . .	27
Figura 4.1 – Posicionamento manual. (a) Sem cruzamento. (b) Com cruzamento.	29
Figura 4.2 – Modelos de layout de porta lógica da maioria no esquema de clock USE com tamanho de zona 5x5. . . . .	30
Figura 4.3 – Modelo de grafo DGA . . . . .	31
Figura 4.4 – Representação do fluxo de dados no esquema de clock USE . . . . .	31
Figura 4.5 – Representação em formato de grafo do esquema de clock USE . . . . .	32
Figura 4.6 – Transformação do circuito num grafo AIG de no máximo 2 entradas e 2 saídas. . . . .	32
Figura 4.7 – Numeração de vértices de acordo com profundidade no grafo. . . . .	33
Figura 4.8 – Classificação de vértices em níveis. . . . .	33
Figura 4.9 – Inserção de vértice de balanceamento. . . . .	34
Figura 4.10 – Representação gráfica dos dois arquivos de entrada para a heurística P&R. . . . .	34
Figura 4.11 – Fluxograma para a heurística P&R. . . . .	35

Figura 4.12–Passo a passo da busca pela posição do vértice 1. a) Representação gráfica do circuito lógico à ser posicionado. b) Possível posicionamento para os vértices dos níveis 2, 3, 4 e 5. c) Busca por rotas de tamanho 1 para o posicionamento do vértice 1 com <i>fanout</i> nos vértices 3 e 4. d) Busca por rotas de tamanho 2 para o posicionamento do vértice 1 com <i>fanout</i> nos vértices 3 e 4 e) Busca por rotas de tamanho 3 para o posicionamento do vértice 1 com <i>fanout</i> nos vértices 3 e 4 f) Rotas de tamanho 4 encontradas para o vértice 1 com <i>fanout</i> nos vértices 3 e 4. . . . .	37
Figura 4.13–Busca por rotas de comprimento mínimo. . . . .	38
Figura 4.14–Ordem de posicionamento de vértices. . . . .	39
Figura 4.15–Definição de nomenclatura de vértices. Vértice 6 é pai dos vértice 3 e 4. Vértice 1 é filho dos vértices 3 e 4. . . . .	39
Figura 4.16–Exemplo de busca por rota de comprimento mínimo. . . . .	40
Figura 4.17–(a) Circuito <i>XOR</i> em nível de porta lógica. (b) Grafo do circuito <i>XOR</i> gerado para entrada da heurística. . . . .	41
Figura 4.18–Passo a passo da execução da heurística de P&R para o circuito <i>XOR</i> . . . . .	41
Figura 4.19–Resultado do P&R para o <i>XOR</i> no QCADesigner. . . . .	42
Figura 4.20–Forma de onda do circuito <i>XOR</i> da figura 4.19. . . . .	42
Figura 5.1 – Configuração do mecanismo de simulação no QCADesigner. . . . .	43
Figura 5.2 – (a) Circuito <i>MUX</i> em nível de porta lógica. (b) Grafo do circuito <i>MUX</i> gerado para entrada da heurística. . . . .	44
Figura 5.3 – Passo a passo da execução da heurística de P&R para o circuito <i>MUX 2x1</i> . . . . .	45
Figura 5.4 – Resultado do P&R para o <i>MUX</i> no QCADesigner. . . . .	46
Figura 5.5 – P&R do circuito <i>MUX 2x1</i> na ferramenta QCA-LG (TEODÓSIO; SOUSA, 2007). . . . .	46
Figura 5.6 – Forma de onda do circuito <i>MUX 2x1</i> da figura 5.3. . . . .	46
Figura 5.7 – (a) Circuito <i>Full Adder de 1-bit</i> em nível de porta lógica. (b) Grafo do circuito <i>Full Adder de 1-bit</i> gerado para entrada da heurística. . . . .	47
Figura 5.8 – Passo a passo da execução da heurística de P&R para o circuito <i>Full Adder de 1-bit</i> . . . . .	47
Figura 5.9 – Resultado do P&R para o <i>Full Adder de 1-bit</i> no QCADesigner. . . . .	48
Figura 5.10–P&R do circuito <i>Full Adder de 1-bit</i> na ferramenta QCA-LG (TEODÓSIO; SOUSA, 2007). . . . .	48
Figura 5.11–Forma de onda do circuito <i>Full Adder de 1-bit</i> da figura 5.8. . . . .	49
Figura 5.12–Demonstração de calculo de área do circuito QCA. . . . .	49

Figura 5.13–(a) Circuito <i>XNOR</i> em nível de porta lógica. (b) Grafo do circuito <i>XNOR</i> gerado para entrada da heurística. . . . .	51
Figura 5.14–P&R manual do circuito <i>XNOR</i> (JAGARLAMUDI; SAHA; JAGARLAMUDI, 2011). Resultado do P&R para o <i>XNOR</i> no QCADesigner. . . . .	51
Figura 5.15–Forma de onda do circuito QCA do <i>XNOR</i> . . . . .	52
Figura 5.16–(a) Circuito <i>Parity Generator</i> em nível de porta lógica. (b) Grafo do circuito <i>Parity Generator</i> gerado para entrada da heurística. . . . .	52
Figura 5.17–P&R manual do circuito <i>Parity Generator</i> (AHMAD; BHAT, 2012). Resultado do P&R para o <i>Parity Generator</i> no QCADesigner. . . . .	53
Figura 5.18–Forma de onda do circuito QCA do <i>Parity Generator</i> . . . . .	53
Figura 5.19–(a) Circuito <i>Parity Checker</i> em nível de porta lógica. (b) Grafo do circuito <i>Parity Checker</i> gerado para entrada da heurística. . . . .	54
Figura 5.20–P&R manual do circuito <i>Parity Checker</i> (AHMAD; BHAT, 2012). Resultado do P&R para o <i>Parity Checker</i> no QCADesigner. . . . .	54
Figura 5.21–Forma de onda do circuito QCA do <i>Parity Checker</i> . . . . .	54

# Lista de tabelas

Tabela 1 – Layout Manual versos P&R . . . . . 50

# Lista de abreviaturas e siglas

P&R	Posicionamento e Roteamento
CMOS	<i>Complementary Metal-Oxide-Semiconductor</i>
RAM	<i>Random Access Memory</i>
QCA	<i>Quantum Cellular Automata</i>
CPU	<i>Central Processing Unit</i>
USE	<i>Universal, Scalable and Efficient clocking scheme for QCA</i>
QCA-LG	<i>Quantum Cellular Automata - Layout Generator</i>
RTL	<i>Register transfer level</i>
DGA	Grafo Acíclico Direto
AIG	<i>And-Inverter Graphs</i>
VLSI	<i>Very-large-scale integration</i>
GML	<i>Graph Modelling Language</i>

# Resumo

FERNANDES, Alyson Trindade, M.Sc., Universidade Federal de Viçosa, agosto de 2017. **Um algoritmo de posicionamento e roteamento de células QCA no esquema de clock USE.** Orientador: Ricardo dos Santos Ferreira.

QCA possui um grande potencial para a construção das novas gerações de circuitos integrados, oferecendo baixo consumo de energia, escalabilidade e alta frequência de processamento. Entretanto, faltam ferramentas de projeto para automatizar etapas como o posicionamento e roteamento de circuitos. Este trabalho apresenta a primeira heurística no esquema de Clock USE capaz de efetuar o posicionamento e roteamento de circuitos combinacionais em nível de porta lógica, por meio da alocação de células QCA. O esquema de clock USE é vantajoso por ser regular, escalável e universal. O circuito é mapeado em um grafo direto acíclico e posicionado em uma matriz de células QCA. A heurística proposta busca otimizar a área e gerar resultados de forma automatizada em comparação com outros trabalhos, onde os projetos são feitos manualmente. Os resultados foram validados com o uso da ferramenta QCADesigner.

# Abstract

FERNANDES, Alyson Trindade, M.Sc., Universidade Federal de Viçosa, August, 2017. **An algorithm for positioning and rotating of QCA cells in the scheme of clock USE.** Adviser: Ricardo dos Santos Ferreira.

QCA has great potential for building new generations of integrated circuits, offering low power consumption, scalability and high processing frequency. However, there is a lack of design tools to automate placement and routing of circuits. This work presents the first heuristic in the Clock USE scheme capable of performing the P&R of combinational circuits at the logic gate level, through the allocation of QCA cells. The USE clock scheme is advantageous because it is regular, scalable, and universal. The circuit is mapped to an acyclic direct graph and positioned in an array of QCA cells. The proposed heuristic seeks to optimize the area and generate results in an automated way compared to other works, where the PR are done manually. The results were validated with the use of the QCADesigner tool.

# 1 Introdução

Desde a construção dos primeiros computadores eletrônicos em meados do século XX, o homem vem buscando novas tecnologias para elevar o poder de processamento de suas máquinas.

No final da década de 50, os computadores transistorizados começaram a substituir as máquinas construídas com estanho (KEYES, 2005). Desde então, o processo de fabricação de transistores de silício foi sendo refinado. Posteriormente, na década de 60, começaram a serem produzidos os primeiros circuitos integrados, permitindo a fabricação de centenas de transistores em um mesmo dispositivo.

A evolução tecnológica do processo de fabricação e a redução do tamanho dos transistores permitiu que fosse atingindo no início deste século a marca de um bilhão de transistores em um único circuito integrado.

A principal tecnologia empregada na fabricação dos circuitos atuais é a MOSFET (complementary metal-oxide-semiconductor). Os circuitos MOSFET são largamente utilizados na produção de processadores, controladores e memórias RAM.

Em 1965, Gordon Moore (MOORE et al., 1998) previu que a quantidade de transistores em um chip dobraria anualmente pelos próximos dez anos. Mais tarde, o próprio Moore atualizou este tempo de doze para vinte e quatro meses.

Esta previsão vem se sustentando por várias décadas como apresenta a figura 1.1 (CAVIN; LUGLI; ZHIRNOV, 2012).

## 1.1 Motivação

Trabalhos recentes (KEYES, 2005), (MARKOV, 2014) apontam um limite no aumento da densidade de transistores nos processadores. A miniaturização dos transistores à níveis quânticos gera efeitos que afetam suas características e funcionamento (KEYES, 2005).

O maior problema na miniaturização dos transistores ocorre quando o comprimento do canal, responsável por isolar a fonte do dreno, é pequeno o suficiente para permitir a corrente de fuga quando o transistor se encontra desligado.

Com o objetivo de impedir a corrente de fuga, novos elementos estão sendo estudados a fim de substituir o material utilizado atualmente na fabricação dos transistores. Porém, tem-se convicção de que existe um limite na redução do tamanho de sua litografia (KEYES, 2005), (MARKOV, 2014).

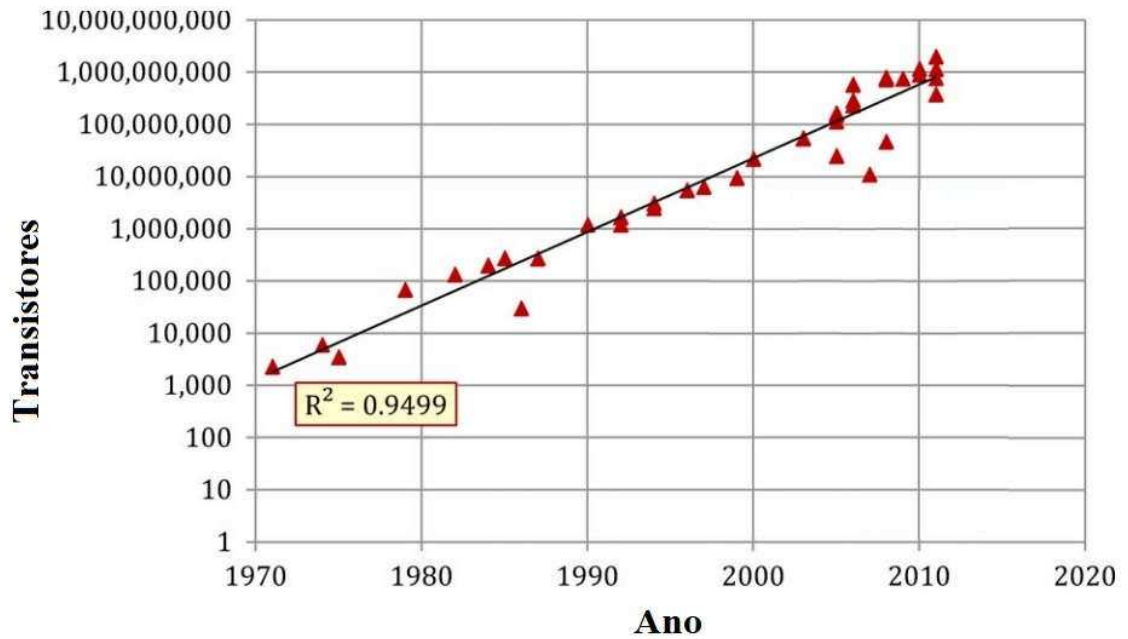


Figura 1.1 – Distribuição da quantidade de transistores nos microprocessadores em relação ao tempo, apresentando a inserção de novas tecnologias. (CAVIN; LUGLI; ZHIRNOV, 2012)

A frequência de chaveamento é outra característica importante ao se avaliar o desempenho dos circuitos. A figura 1.2 demonstra que tal frequência aumentou até se estabilizar por volta de 2005. Diante desta evolução, esperava-se que os processadores para desktops atingissem uma frequência estimada em 10 GHz até o ano de 2005. Todavia, esta previsão não foi confirmada por alguns fatores, entre eles, o acréscimo na geração de calor, a dificuldade em dissipar tal calor, o elevado consumo de energia e problemas de corrente de fuga (SUTTER, 2005). Estes fatores limitaram a frequência de clock além dos 4 GHz.

Nas últimas décadas, diversas tecnologias foram propostas como forma de substituição para a atual técnica de fabricação de circuitos integrados. Entre elas, pode-se citar a computação quântica, nanotubos de carbono, Nanomagnet Logic (NML), Quantum Cellular Automata (QCA), etc.

Cada uma dessas nanotecnologias apresenta um conjunto de vantagens e desvantagens em comparação com a técnica atual para a produção de circuitos integrados. Para que uma dessas tecnologias possa de fato se tornar comercialmente viável, ela deve elevar em grande escala o desempenho dos processadores.

Um grande desafio para as tecnologias emergentes vem da necessidade de adaptação nas diversas fases de fabricação de um circuito. Manter a maior compatibilidade possível em seu processo de fabricação com as linhas de produção de hoje irá possibilitar a reutilização de processos, equipamentos e ferramentas atuais.

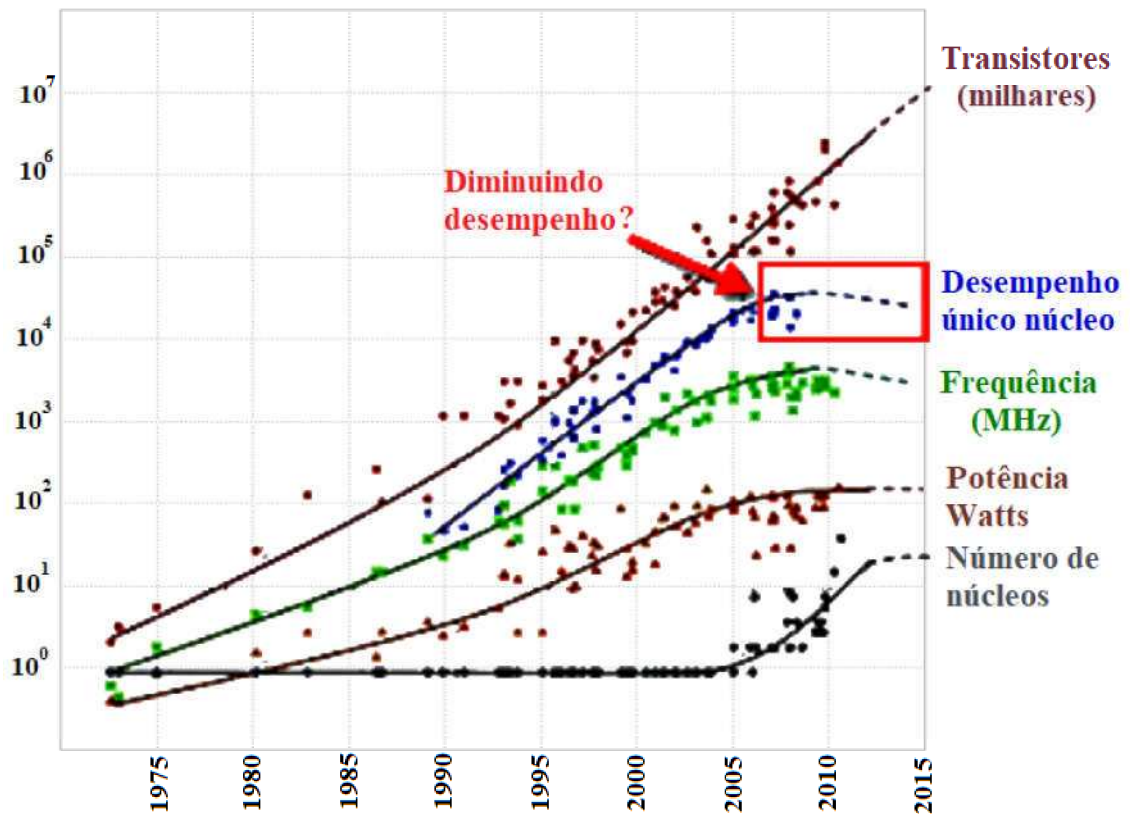


Figura 1.2 – Evolução no desempenho dos processadores (PRESHING, 2012)

Neste contexto, a nanotecnologia QCA se apresenta com grande potencial para a construção de circuitos integrados. Dentre suas principais características, percebe-se, uma menor perda de energia elétrica, portas lógicas sendo criados com menor escala do que a atual e o aumento na frequência de processamento (AMIRI; MAHDAVI; MIRZAKUCHAKI, 2008; LANTZ; PESKIN, 2006).

Entretanto, existem poucas ferramentas para se trabalhar na produção de circuitos para QCA. Tarefas como construção e validação de circuitos QCA são realizadas manualmente e demandam grande quantidade de tempo. Portanto mais ferramentas devem ser desenvolvidas com o propósito de facilitar e acelerar o projeto de circuitos para QCA.

## 1.2 Objetivo

Com intenção de auxiliar no processo de aperfeiçoamento do QCA, este trabalho propõe a construção de uma heurística para realizar as tarefas de posicionamento e roteamento (P&R) de circuitos lógicos em circuitos QCA.

As etapas de P&R são executadas na fase final do projeto para a construção de um novo circuito. Estas etapas consistem primeiro em definir a posição das portas

lógicas e gerar conexões entre estas.

Este trabalho considera o esquema de clock USE proposto em (CAMPOS et al., 2016). As principais contribuições do USE (CAMPOS et al., 2016) são a possibilidade de construção de estruturas como fios e portas com formatos distintos, a garantia do mapeamento de circuitos lógicos de tamanhos e formas variadas e o posicionamento de portas lógicas e fios no mesmo plano e sem qualquer restrição.

### 1.3 Contribuições da dissertação

Este trabalho apresenta a primeira heurística capaz de realizar o posicionamento e roteamento automáticos de funções lógicas em QCA com esquema de clock USE.

O algoritmo desenvolvido recebe como entrada uma descrição em nível lógico de um circuito. Posteriormente são realizadas algumas transformações com objetivo de prepará-lo para o P&R. Por fim, o circuito é posicionado e roteado em uma matriz de células QCA.

### 1.4 Organização da dissertação

Este trabalho está dividido da seguinte forma. O capítulo 2 apresenta as definições para os principais conceitos envolvendo a nanotecnologia QCA e o esquema de clock USE.

O capítulo 3, descreve os principais trabalhos desenvolvidos tendo como objetivo principal a construção de algoritmos e/ou ferramentas para o P&R de circuitos para QCA.

A ferramenta desenvolvida neste trabalho é apresentada no capítulo 4, assim como, suas principais características e seus resultados.

Por fim, o capítulo 5 traz as conclusões deste trabalho e sugere temas importantes para futuras pesquisas.

## 2 Tecnologia QCA

### 2.1 Introdução

QCA foi proposta inicialmente em (LENT et al., 1993). A ideia inicial do QCA é apresentada na figura 2.1. Lent propôs a construção de um conjunto de células compostas por quatro pontos quânticos posicionados nos cantos de cada célula e interligados a um ponto central por meio de túneis. Posteriormente, o ponto central foi eliminado da proposta inicial.

Um ponto quântico, também conhecido como dot, é definido por uma localização da célula QCA onde é permitido o repouso de cargas elétricas.

Para efeito de cálculo, dois elétrons livres são posicionados dentro da célula. A interação coulombiana entre estes elétrons gera a repulsão eletrostática capaz de organizá-los sempre em posições antipodais.

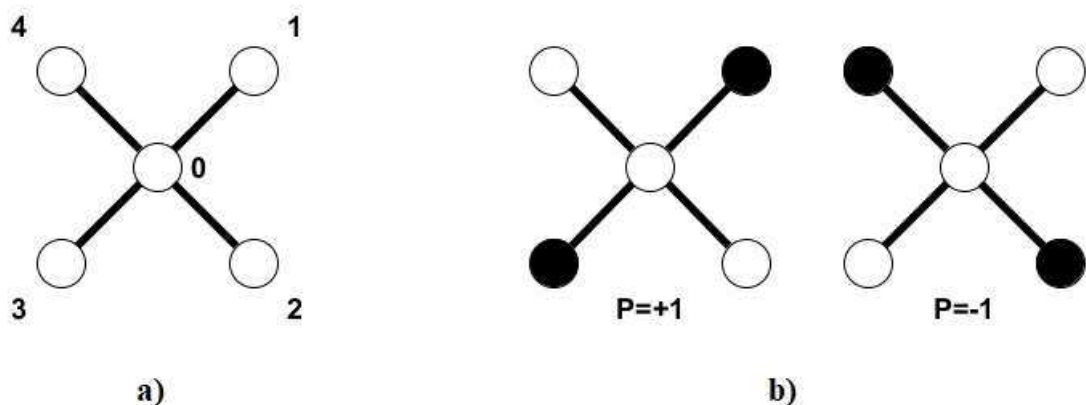


Figura 2.1 – Estrutura básica de uma célula QCA proposta por (LENT et al., 1993)

Uma modelagem que sustenta a lógica QCA foi proposta em (LENT et al., 1993). Ademais foram apresentadas formas de construir estruturas básicas como fios, inversores e portas lógicas *E* e *OU*, por meio de conjuntos regulares de células QCA.

### 2.2 Célula QCA

A célula QCA é a estrutura básica para a definição da tecnologia. A figura 2.2 a) apresenta uma figura tridimensional baseada na estrutura desta célula. Esta figura demonstra uma célula QCA hipotética.

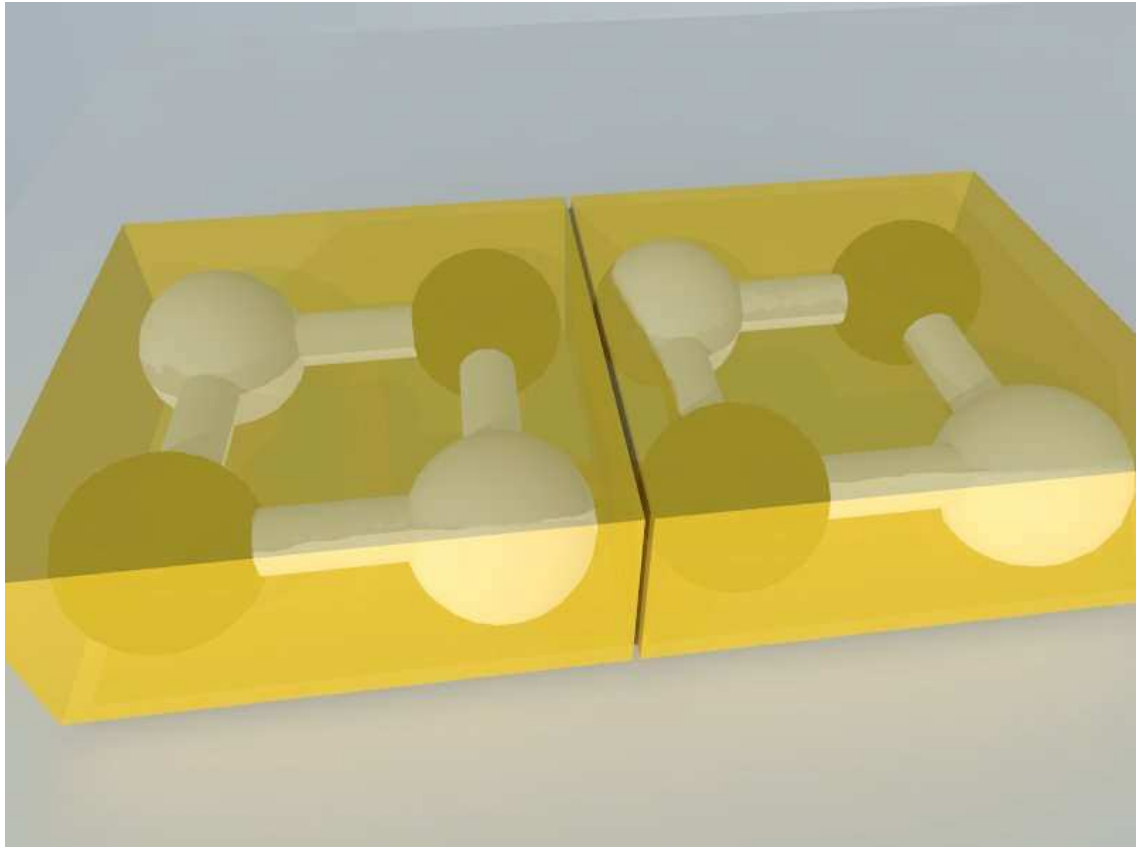


Figura 2.2 – Visão tridimensional do modelo da célula QCA

Internamente à célula, percebe-se a disposição de quatro dots posicionados nos cantos da estrutura. Estes dots podem ser entendidos como áreas de repouso para os elétrons livres que se encontram dentro da célula.

Entre os quatro pontos quânticos devem haver barreiras fortes o suficiente para suprimir o tunelamento dentro da célula QCA (LENT *et al.*, 1993). Em determinado momento deve ser possível diminuir as barreiras de contenção da célula para que cargas elétricas estejam livres em seu interior.

Há ainda, dois elétrons livres dentro de cada célula QCA. A repulsão eletrostática natural entre partículas de mesma carga levará os elétrons a ocuparem posições antipodais em relação à célula. Neste cenário haverá sempre duas possibilidades de configurações distintas para as posições dos elétrons livres. Assim, pode-se utilizar estas configurações na representação de informações binárias. Esta estrutura foi experimentalmente validada em (ORLOV *et al.*, 1997).

A figura 2.1 b) apresenta células polarizadas. (TOUGAW; LENT, 1994) define a polarização de uma célula como uma variável  $P$  que mede a extensão da distribuição de cargas pelos dois eixos diagonais da célula QCA.

A polarização de uma célula é medida pela fórmula a seguir:

$$P \equiv \frac{(p_1+p_3)-(p_2+p_4)}{(p_1+p_2+p_3+p_4)}$$

Onde  $p_i$  equivale à carga elétrica do dot  $i$  sendo  $i$  o índice do mesmo. Sendo assim, quando os elétrons se encontram posicionados nos dots um e três a polarização interna da célula é igual à +1 e quando os elétrons ocupam os dots dois e quatro a polarização final da célula é equivalente à -1. Por convenção, definiu-se que células com cargas +1 representam a informação binária 1, enquanto células com carga -1 armazenam o dígito binário 0.

## 2.3 Estruturas básicas

### 2.3.1 Fio

QCA utiliza a interação coulombiana para transmitir informação entre células vizinhas. Desde que seja viável manter a polarização de uma célula QCA, é possível transmitir a informação desta célula para outra vizinha se elas estiverem posicionadas a certa distância.

Uma célula QCA com polarização induzida possui um campo de força eletromagnética de raio  $R$ . Toda célula despolarizada que estiver dentro deste raio é atingida por uma força magnética denominada energia de acoplamento capaz de induzir a polarização desta célula.

Baseado neste conceito, é possível construir um fio QCA posicionando uma cadeia de células em fila e próximas o bastante para que as células vizinhas interajam entre si. Ao se fixar a polaridade da célula em uma das duas bordas da fila, é possível transmitir este sinal para a outra extremidade da fila relaxando a polarização de todas as outras células.

A figura 2.3 apresenta a estrutura do fio utilizando células QCA alinhadas. Nesta figura percebe-se sempre a célula mais à esquerda com polarização induzida. Todas as outras células da estrutura receberão a informação a partir desta célula. Também é possível perceber nesta figura a construção de fios com curva e bifurcações (*fanout*).

### 2.3.2 Inversor

Existem vários formatos propostos na literatura para a construção de um inversor (LENT et al., 1993), (TOUGAW; LENT, 1994), (FARAZKISH et al., 2010), (HUANG et al., 2004), (LANTZ; PESKIN, 2006), (PATITZ, 2006). A estrutura apresentada na figura 2.4 representa o inversor construído em (TOUGAW; LENT, 1994). Nota-se nesta figura que o sinal da saída (direita) está invertido em relação ao sinal da entrada (esquerda). A troca do sinal ocorre no posicionamento perpendicular

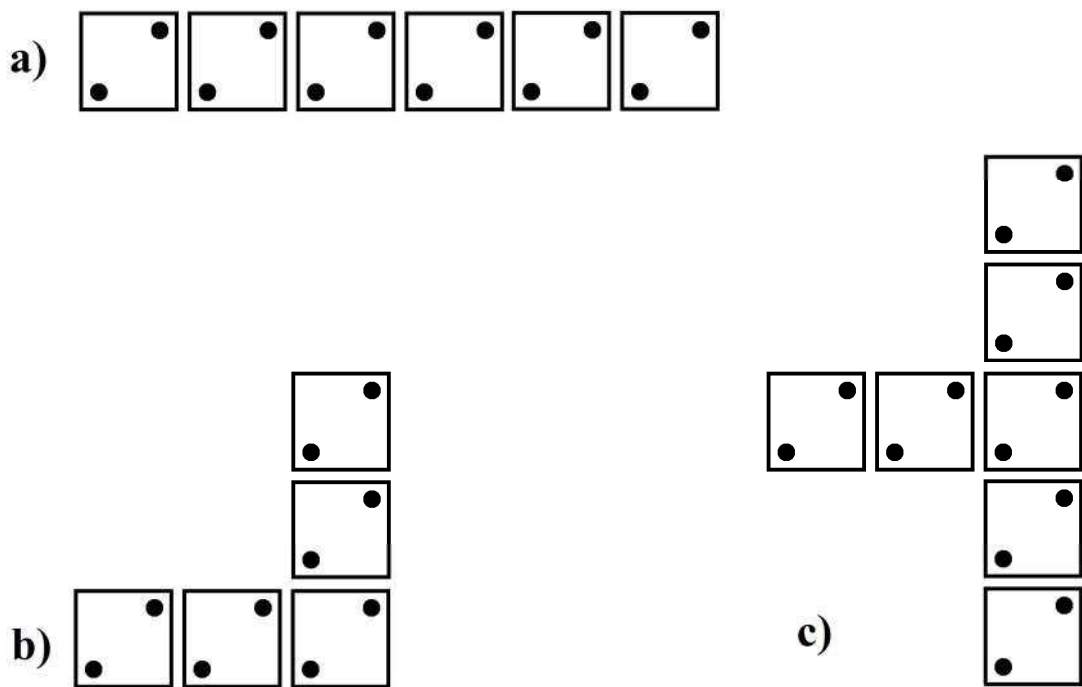


Figura 2.3 – Representação de fios em QCA. a) Fio reto; b) Fio com curva; c) Fio com *fanout* (LENT et al., 1993).

entre as células (A, B) e C. Sempre que duas células vizinhas forem posicionadas desta forma ocorrerá a inversão de sinal (polarização).

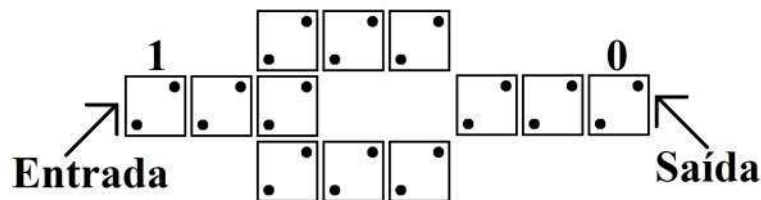


Figura 2.4 – Inversor QCA (TOUGAW; LENT, 1994)

### 2.3.3 Porta da Maioria (*Majority Gate*)

O padrão mais utilizado para se construir portas lógicas em QCA tem como base a porta lógica da maioria. A porta da maioria, pode se construída com apenas cinco células QCA dispostas em formato de cruz. O formato desta porta foi criado por (TOUGAW; LENT, 1994). A figura 2.5 apresenta tal formato.

A porta da maioria possui três células configuradas como entradas e uma célula de saída. Como ilustrado na figura, a célula central é responsável por executar a função lógica. A saída receberá o sinal mais presente nas três entradas da porta lógica. No exemplo da figura 2.5, as entradas B e C possuem sinal lógico 1, enquanto

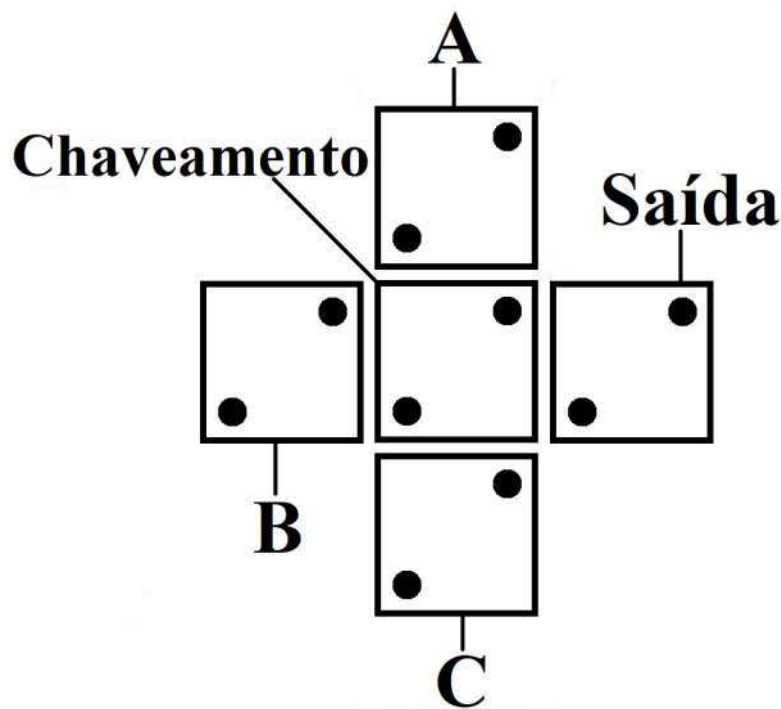


Figura 2.5 – Porta da Maioria

a entrada A possui sinal 0. Neste caso a força de acoplamento resultante do somatório das células B e C é maior que a força da célula A. Isto faz com que os elétrons encontrados na célula central sejam pressionados a manter-se na polarização +1 (sinal 1). Posteriormente, a célula de saída também é induzida a manter-se na polarização +1. Esta porta pode ser representada pela seguinte função booleana:

$$M(A,B,C) = AB+AC+BC$$

Utilizando a estrutura de uma porta da maioria é possível construir portas *E* e *OU* por meio da fixação da polaridade de uma das células de entrada. A figura 2.6 apresenta a tabela verdade da porta da maioria. Verifica-se que ao dividir a tabela pela célula de entrada A, a porta da maioria se comporta hora como uma função lógica *E*, hora como uma função lógica *OU*. Quando esta, fixa o sinal de entrada na célula A em 0, o restante do sistema se comporta como uma porta *E*. Porém, quando a célula A tem seu sinal fixado em 1, a saída mantém o padrão de uma porta *OU*. As imagens da figura 2.7 expõem este cenário por meio de imagens da porta da maioria que possui uma entrada com polaridade fixa.

As funções a seguir demonstram a derivação da função geral da porta da maioria com uma entrada com polaridade fixa:

$$M(1,B,C) = 1B+1C+BC = B+C$$

-	A	B	C	Maioria	
E	0	0	0	0	E) A=0 MAIORIA = 0B + 0C + BC MAIORIA = BC
	0	0	1	0	
	0	1	0	0	
	0	1	1	1	
OU	1	0	0	0	OU) A=1 MAIORIA = 1B + 1C + BC MAIORIA = B+C
	1	0	1	1	
	1	1	0	1	
	1	1	1	1	

Figura 2.6 – Tabela verdade da porta da maioria. Entrada A igual a 0 equivale a porta lógica E. Entrada A igual a 1 equivale a porta lógica OU

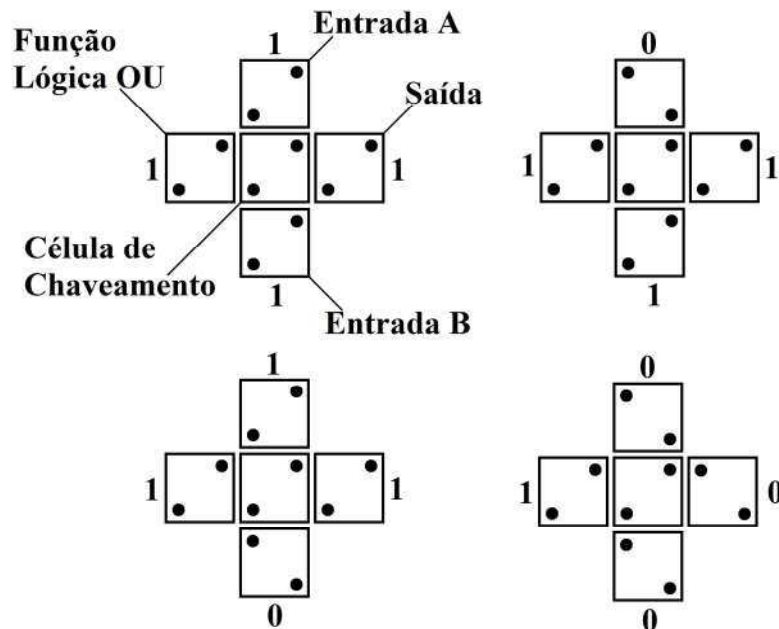


Figura 2.7 – Porta da maioria funcionando como E lógico com a entrada A com polaridade induzida. (TOUGAW; LENT, 1994)

$$M(0,B,C) = 0B+0C+BC = BC$$

### 2.3.4 Cruzamento de Fios

Em qualquer tecnologia é importante possibilitar o cruzamento de fios para a construção de circuitos que em geral possuem uma topologia que não pode ser implementada por um grafo planar.

Algumas formas foram propostas para definir o cruzamento de fios em QCA. Predominam dois formatos nesta solução. Cruzamento coplanar (TOUGAW; LENT,

1994), (PATITZ, 2006), (BHANJA et al., 2008) e (BHANJA et al., 2006) e cruzamento multicamadas (DINDIGUL, ) e (SHIN; JEON; YOO, 2013).

A figura 2.8 ilustra o cruzamento coplanar. Na figura 2.8 a) percebe-se o uso de células QCA com seus dots rotacionados em  $45^\circ$  em relação à célula padrão. A figura 2.8 b) utiliza o clock para impedir a interferência de sinal no cruzamento (SHIN; JEON; YOO, 2013).

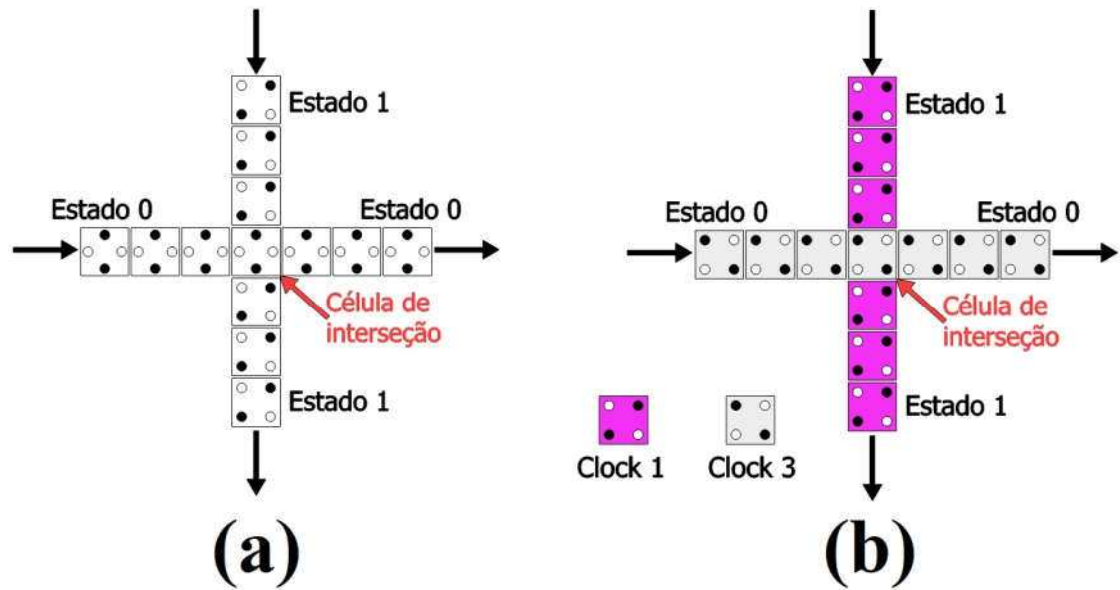


Figura 2.8 – Cruzamento coplanar (SHIN; JEON; YOO, 2013)

O segundo formato de cruzamento pode ser visto na figura 2.9. Ele é denominado de cruzamento multicamadas por empilhar células QCA em camadas verticais (SHIN; JEON; YOO, 2013).

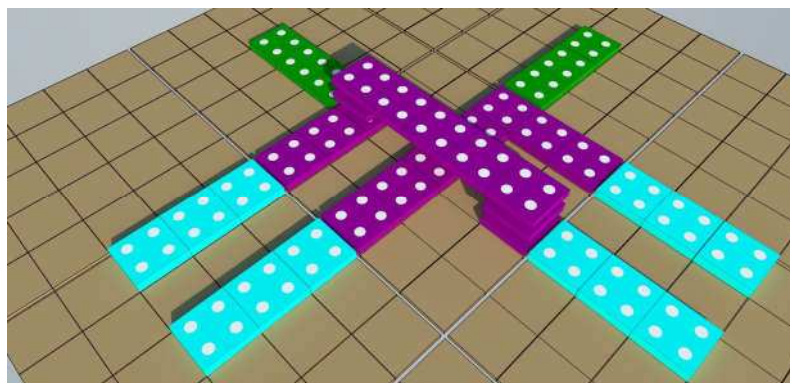


Figura 2.9 – Cruzamento multicamadas

Ambas as técnicas possuem vantagens e desvantagens. O cruzamento multicamadas apresenta o problema de ruído entre células que se cruzam na região de interseção (JANEZ; PECAR; MRAZ, 2012). O cruzamento com células rotacionadas

(coplanar) reduz o desempenho do circuito por diminuir a temperatura máxima de operação e o tempo de chaveamento mínimo (GRAUNKE et al., 2005).

Devido a falta de ferramentas para QCA, ainda não foi possível indicar qual destas técnicas prevalecerá na evolução da tecnologia.

## 2.4 Clock QCA

Em CMOS, os elétrons transitam pelo circuito sempre num mesmo sentido. Em QCA os sinais podem percorrer em direções incontroláveis. Deste modo, o fluxo de informação pode assumir direções indeterminadas (WALUS; JULLIEN; DIMITROV, 2003). Como forma de proporcionar um comportamento determinístico, um esquema de clock e pipeline para QCA foi demonstrado em (LENT; TOUGAW, 1997).

A proposta consiste em controlar a barreira que impede o fluxo de elétrons dentro da célula QCA. Este controle deve ser capaz de alterar o potencial de restrição do fluxo de elétrons nos canais que interligam os dots. Desta forma é possível transformar este canal hora em uma via de tráfego de elétrons, hora em uma barreira que impeça o fluxo dos mesmos. Sendo assim, é possível por meio deste controle armazenar o sinal lógico contido em uma célula QCA por um determinado tempo.

O esquema de clock proposto em (LENT; TOUGAW, 1997) é baseado em quatro fases. São utilizadas barreiras entre os dots para controlar a transição entre cada estágio. A figura 2.10 exhibe a configuração dos quatro estágios distintos propostos neste esquema (LENT; TOUGAW, 1997).

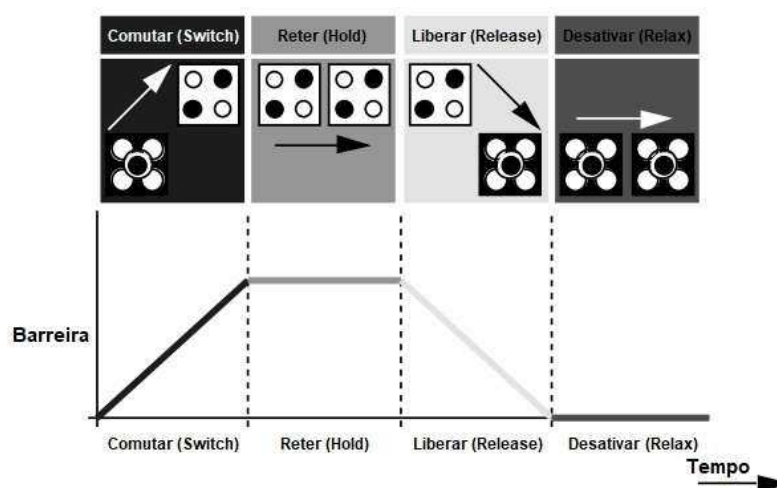


Figura 2.10 – Esquema de clock com quatro estágios. (NIEMIER; KOGGE, 2004)

A figura 2.11 apresenta a transição entre os quatro estágios do clock QCA. O primeiro estágio é nomeado de *comutar* (*switch*). Nesta fase, inicialmente as barreiras se encontram num estado baixo, onde os elétrons podem trafegar livremente dentro

da célula. Ainda na fase de *comutar*, num segundo momento, as barreiras são erguidas até polarizarem a célula por completo. Portanto, a célula que se encontra neste estágio será polarizada de acordo com a força de acoplamento de suas células vizinhas. Neste momento os elétrons se encontram confinados dentro dos dots. Ao final desta fase cada célula carregou o sinal lógico de sua vizinha.

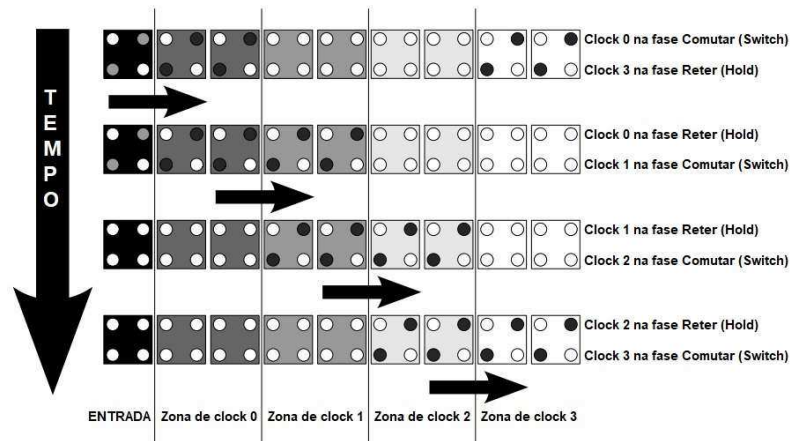


Figura 2.11 – Exemplo interativo de transmissão de sinal pelas zonas de clock QCA (CAMPOS et al., 2016).

A segunda etapa é responsável por manter as barreiras das células QCA polarizadas e permitir que o sinal armazenado na célula sirva como entrada para suas células vizinhas. Dá-se o nome de *reter (hold)* para esta etapa. Deve-se garantir que os elétrons confinados não escapem para qualquer outro dot durante esta etapa.

Posteriormente, a etapa *liberar (release)* é acionada. Neste instante, as barreiras de contenção de elétrons vão sendo reduzidas de modo a despolarizar a célula. Ao final desta etapa, a célula não possui nenhum sinal válido armazenado.

A etapa *desativar (relax)* é o quarto e último estágio. Ela Consiste em manter as células sem polarização.

Independente da quantidade de células que ocupam uma zona de clock, o atraso do sinal será sempre o mesmo (WALUS; JULLIEN; DIMITROV, 2003). A figura 2.12 exibe uma porta da maioria e um fio projetados com quatro zonas de clock cada. As zonas são representados pelos diferentes tons de cinza. Neste cenário, o atraso que o sinal leva para transitar da entrada até a saída será o mesmo para ambos. Portanto, o atraso total de um circuito é medido não pelo tamanho do circuito e sim pela quantidade de zonas de clock que compõe seu caminho crítico.

O circuito combinatório da figura 2.13 demonstra como os sinais são processados em CMOS. Suponha que cada porta lógica apresenta um atraso de uma unidade de tempo e os fios não gerem atrasos. A partir do tempo 2, momento em que as três entradas têm seus valores alterados a entrada *C* leva o intervalo de uma unidade de

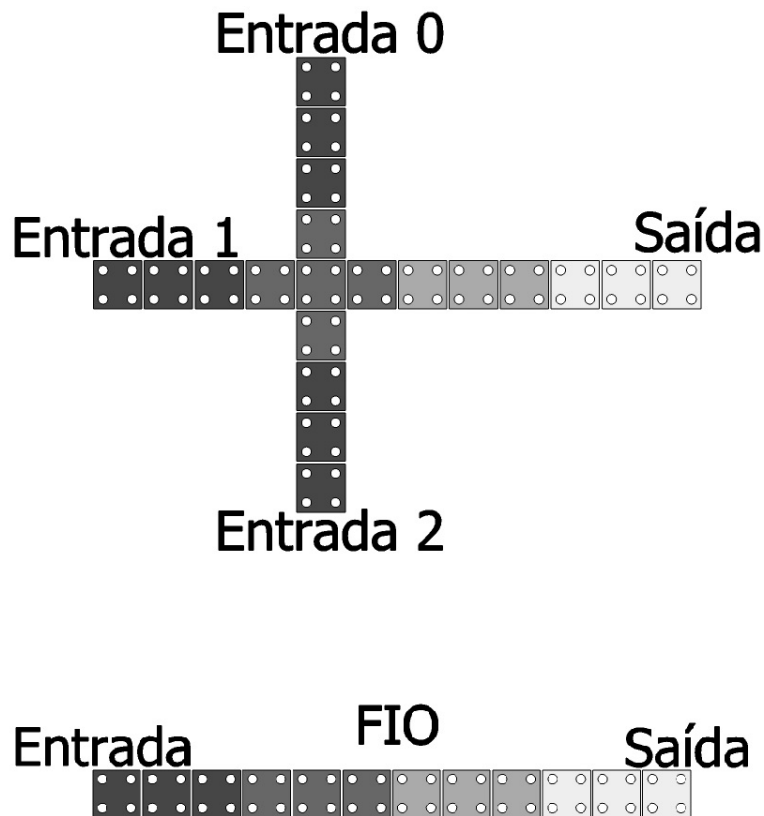


Figura 2.12 – A quantidade de células QCA não aumenta a latência do circuito (WALUS; JULLIEN; DIMITROV, 2003)

tempo para alcançar a saída, enquanto que as entradas *A* e *B* necessitam de duas unidades de tempo para atingir a saída. Durante o período de tempo entre a chegada da entrada *C* e a chegada das entradas *A* e *B* na saída, o circuito gera uma variação de curta duração (*glitch*). Portanto, o circuito precisa esperar um intervalo de duas unidades de tempo (atraso do circuito) a partir da mudança de sinal nas entradas para obter um saída válida.

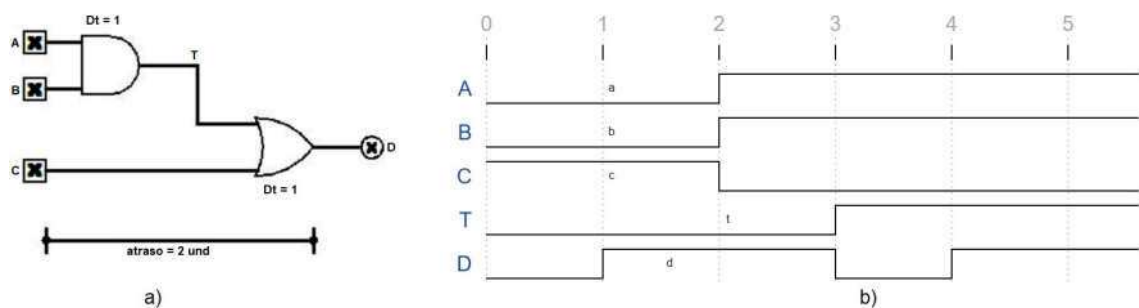


Figura 2.13 – Exemplo de circuito e forma de onda com a presença de *glitch*

No entanto, os circuitos QCA são mais complexos, pois todos os sinais devem estar balanceados. O atraso para todas as entradas deve ser o mesmo. Conseqüentemente, é necessário um esquema de clock para sincronizar as entradas mesmo em

circuitos combinatórios.

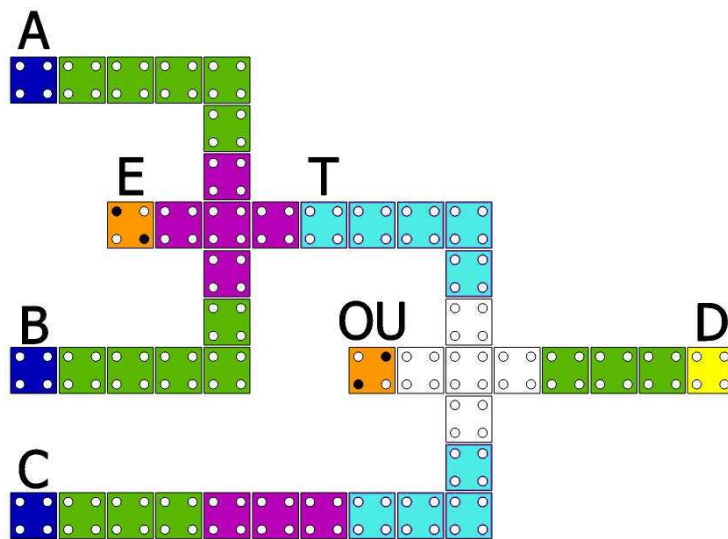


Figura 2.14 – Circuito da figura 2.13 em QCA

## 2.5 USE

Um desafio na evolução da nanotecnologia QCA é a definição do esquema de clock capaz de sincronizar o fluxo de informação que trafega pelo circuito. Porém, vários problemas foram apontados por (NIEMIER, 2000) no alcance desta tarefa. Fios muitos longos, zonas de clock não uniformes e de tamanhos muito distintos, realimentações e regiões não utilizadas no meio do circuito são os principais desafios indicado por (NIEMIER, 2000).

Um esquema capaz de minimizar o tamanho dos fios e criar zonas de clock de tamanho regular é apresentado em (VANKAMAMIDI; OTTAVI; LOMBARDI, 2008), porém as realimentações e regiões não utilizadas ainda se apresentaram como dificuldades não alcançadas.

O esquema proposto por (CAMPOS et al., 2016) possibilita que todos os desafios apontados em (VANKAMAMIDI; OTTAVI; LOMBARDI, 2008) sejam ultrapassados. Este esquema se destaca por ser universal, escalável e eficiente. Tal esquema é denominado de USE (*Universal, Scalable and Efficient*).

Entre as principais contribuições deste esquema observa-se o fato dele:

- Possuir zonas de clock bem delimitadas, uniformes e regulares;
- Dar flexibilidade para criar realimentações (*feedback paths*) de qualquer tamanho;
- Permitir o posicionamento e roteamento de qualquer tipo de circuito;

- Possibilitar que haja fluxo de informação em todas as direções.

A figura 2.15 demonstra como este esquema é proposto. Observa-se a definição de uma matriz de células QCA que são agrupadas em submatrizes menores. Cada submatriz é representada por um quadrado na figura. O número dentro de cada quadrado representa a zona de clock em que esta região se encontra em um determinado momento. As setas demonstram a direção do fluxo de informação.

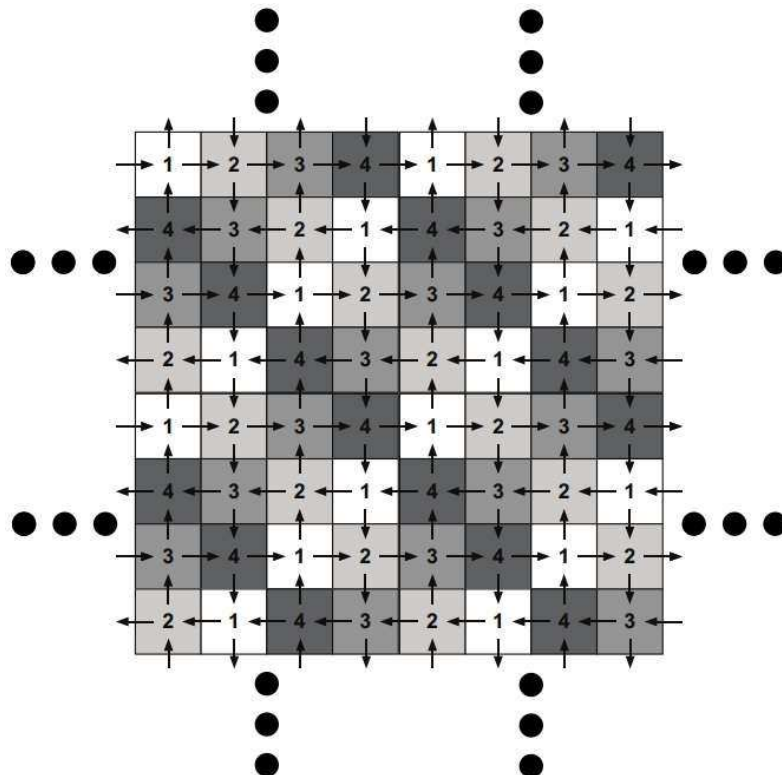


Figura 2.15 – Matriz de células QCA baseada no esquema de clock USE (CAMPOS et al., 2016)

Este esquema é escalável porque sua matriz pode ser ampliada indeterminadamente para receber circuitos de qualquer tamanho.

O dado trafega da região de clock 1 para a região de clock 4, passando pelas regiões 2 e 3. Neste sentido, toda região possui duas zonas de clock vizinhas que agem como fornecedora de dados e duas zonas de clock vizinhas que agem como receptoras de dados. Esta organização é quem garante a flexibilidade na direção do fluxo de dados pelo circuito.

A figura 2.16 apresenta dois circuitos lógicos que representam a função lógica *XOR* modelados na nanotecnologia QCA. A figura da esquerda representa o posicionamento manual do circuito, sendo o projeto mais observado na literatura para a porta *XOR*. Nota-se aqui que as zonas de clock possuem quantidade variada de células, portanto, não há regularidade. O circuito da direita foi posicionado com



## 2.6 Posicionamento e Roteamento em CMOS e QCA

A nanotecnologia QCA ainda se encontra em amadurecimento. Devem haver adaptações para que ela se assemelhe ao fluxo de construção de circuitos CMOS. Inúmeros investimentos já foram feitos nesta área de circuitos. É essencial que novas tecnologias se aproveitem de processos e ferramentas já reconhecidas por todos os profissionais da área.

Um modelo de fluxo de processo para a construção de circuitos QCA, baseado no processo de fabricação de circuitos CMOS é apresentado por (HENDERSON et al., 2004). A figura 2.6 expõe lado-a-lado os fluxos baseados nas duas tecnologias. Nota-se a semelhança entre os dois processos.

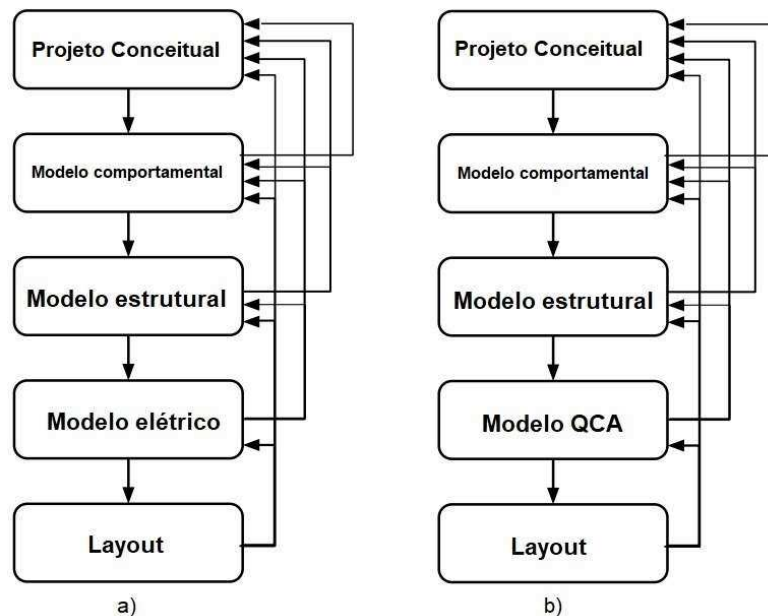


Figura 2.18 – Modelos de fluxo de processo: a) CMOS, b) QCA. (HENDERSON et al., 2004)

As adaptações começam a surgir a partir da terceira fase destes fluxos. As fases de projeto conceitual e modelo comportamental utilizam linguagens de programação e descrição de hardware para definir detalhes estruturais e a modelagem do circuito em alto nível.

A partir da terceira etapa, questões da nanotecnologia QCA começam a ser consideradas. Novas estruturas básicas podem ser desenvolvidas, devido à diferença na criação de portas lógicas básicas entre CMOS e QCA.

O modelo QCA é uma nova etapa no fluxo proposto em (HENDERSON et al., 2004). Embora esta seja uma etapa análoga ao CMOS, as maiores diferenças entre os processos construtivos dos circuitos são evidenciadas neste momento. Dentro do modelo QCA deve ocorrer o posicionamento e roteamento de portas lógicas com

balanceamento dos caminhos (CAMPOS et al., 2016).

Uma modelo expandido do trabalho de (HENDERSON et al., 2004) onde foram adicionadas características como simulação de novos esquemas de clock e injeção de falhas em tempo de execução foi apresentado em (OTTAVI et al., 2006). A figura 2.6, exibe o novo fluxo proposto. A principal alteração vem da inserção de uma base de dados que fornece um conjunto de defeitos e uma biblioteca de dispositivos QCA. Esta base serve de alimentação para rotinas de validação de circuitos em tempo de execução.

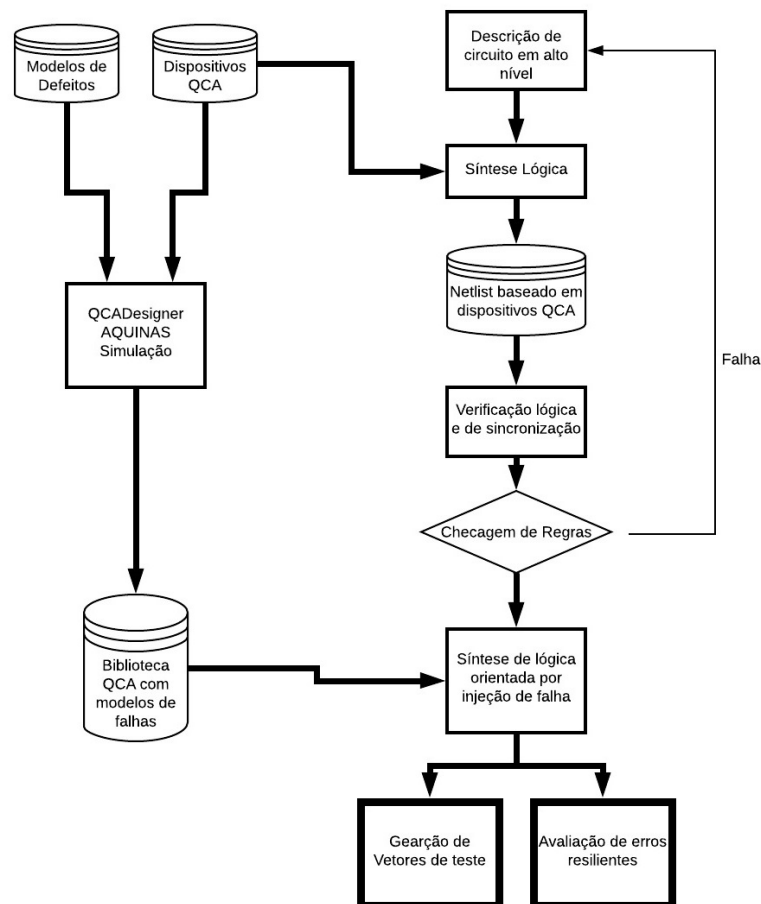


Figura 2.19 – Modelos de fluxo de processo QCA (OTTAVI et al., 2006) (HENDERSON et al., 2004)

## 2.7 QCADesigner

Hoje, a principal ferramenta utilizada no projeto e simulação de circuitos QCA é a QCADesigner (WALUS et al., 2004). QCADesigner apresenta um amplo conjunto de ferramentas CAD para a rápida construção de layouts de circuitos. Sua distribuição é gratuita.

A figura 2.7 apresenta a tela principal do QCADesigner com uma porta *XOR* projetada.

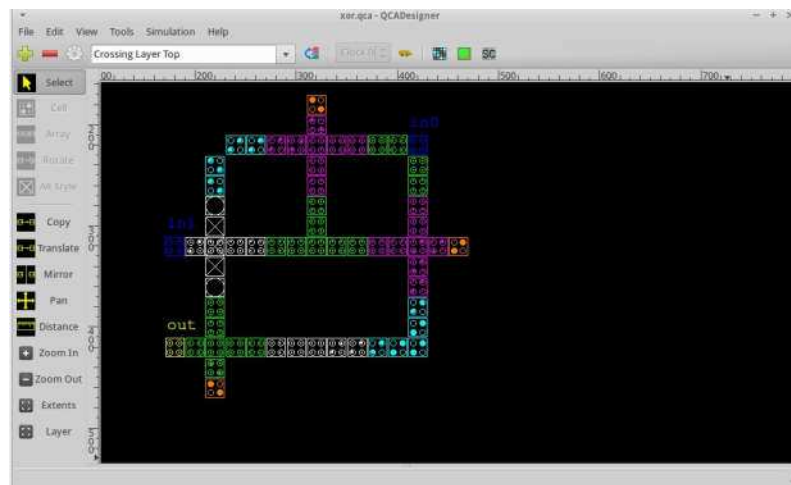


Figura 2.20 – Tela principal do software QCADesigner.  
([HENDERSON et al., 2004](#))

Nesta dissertação os circuitos gerados foram balanceados no QCADesigner. Esta ferramenta está disponível em ([COLUMBIA, 2012](#)). A figura 2.7 a tela principal da ferramenta.

## 3 Trabalhos Relacionados

### 3.1 Introdução

A tecnologia QCA surge como uma das candidatas a substituir o método atual de fabricação de circuitos. Entretanto, deve haver uma adaptação ao processo tradicional de projeto de circuitos CMOS. Como parte de amadurecimento do QCA, será necessário a evolução em duas etapas distintas. O desenvolvimento e validação de conceitos que sustentem a implementação física da tecnologia, e a elaboração de ferramentas que auxiliem no projeto de circuitos.

Este trabalho se concentra em algoritmos capazes de auxiliar tarefas relacionadas ao projeto de circuitos QCA. Duas sub-tarefas são importantes neste contexto: o posicionamento e o roteamento de portas lógicas. Estas tarefas devem buscar diminuir o consumo de recursos, além de, proporcionar o melhor desempenho possível para o circuito projetado.

Dada uma matriz de células QCA e um circuito no nível estrutural de portas lógicas, a tarefa de posicionamento consiste em alocar cada elemento do circuito (entrada, porta lógica, saída) em uma ou mais células USE. O roteamento deve gerar os fios com as células QCA para interligar os elementos.

### 3.2 Posicionamento

Seguindo esta linha, uma heurística capaz de realizar o particionamento e posicionamento de um circuito em nível de portas lógicas em zonas, o posicionamento de células e posteriormente o roteamento foi formulada por (LIM; RAVICHANDRAN; NIEMIER, 2005).

A figura 3.1 expõe graficamente todas as etapas apresentadas pela heurística demonstrada em (LIM; RAVICHANDRAN; NIEMIER, 2005). Inicialmente, várias iterações são executadas em busca dos melhores tamanhos de partições. O objetivo desta etapa é diminuir o número de fios nas bordas entre as diversas partições mantendo o grafo acíclico e conseqüentemente minimizando a quantidade total de fios do circuito. Ainda nesta etapa, são inseridos blocos de fios para balancear o clock do circuito e manter a coesão do mesmo.

A segunda etapa é responsável por alocar cada zona do grafo em uma matriz. Cada zona do primeiro nível do grafo ocupa uma célula da primeira linha da matriz da esquerda para a direita. As zonas do segundo nível são posicionadas na segunda

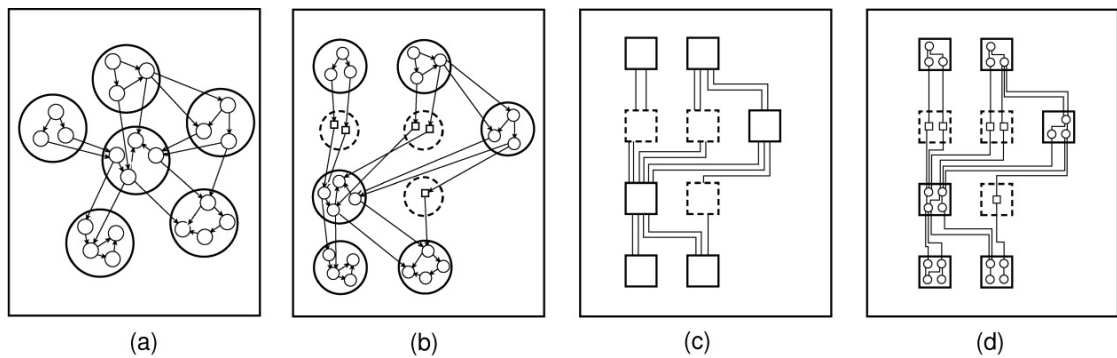


Figura 3.1 – Particionamento, posicionamento e roteamento de um circuito em nível de porta lógica. (a) particionamento do circuito em zonas. (b) posicionamento de células. (c) roteamento global por meio de conexões entre as zonas. (d) roteamento local com conexões dentro das zonas. (LIM; RAVICHANDRAN; NIEMIER, 2005)

linha da matriz e assim por diante, nível a nível. Por fim, as células da matriz são reordenadas para diminuir o cruzamento de fios.

Finalmente, todas as zonas são visitadas e individualmente cada célula QCA é posicionada, a fim de construir as portas lógicas e conexões internas e externas de cada zona dentro do circuito. Esta etapa é exposta pela figura 3.2.

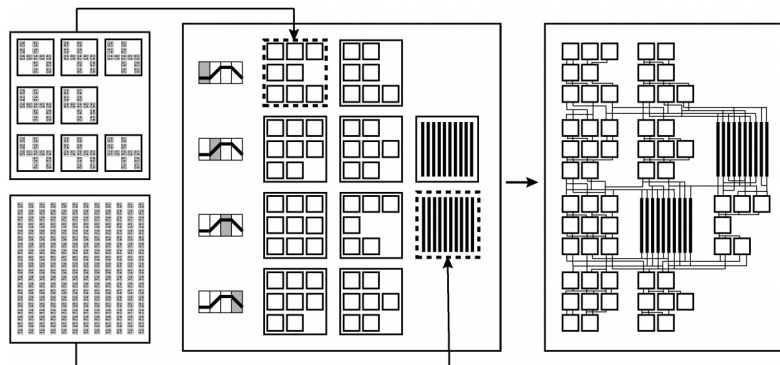


Figura 3.2 – Construção do layout do circuito. Primeiro, o circuito de entrada é particionado em blocos de fios e portas lógicas. Segundo, cada bloco é posicionado em uma matriz bidimensional mantendo as restrições de clock. Terceiro, as células QCA de cada bloco são posicionadas. Quarto, ocorre a interconexão externa e interna de cada bloco. (LIM; RAVICHANDRAN; NIEMIER, 2005)

Além dos objetivos principais já citados, são objetivos secundários de (LIM; RAVICHANDRAN; NIEMIER, 2005):

- Identificar portas lógicas e blocos que podem ser duplicados para reduzir o cruzamento de fios;
- Reordenar portas lógicas para diminuir o cruzamento de fios;

- Criar caminhos de roteamentos mais curtos;
- Reduzir a área de um circuito.

Referente aos objetivos secundários, a heurística apresentada em (LIM; RAVICHANDRAN; NIEMIER, 2005) organizou o circuito QCA em forma de um grafo dirigido acíclico (DGA) como demonstra a figura 3.3. Foram inseridas linhas tracejadas para definir os limites de cada zona de clock. Os nós são posicionados em diferentes níveis de acordo com sua profundidade no grafo (LIM; RAVICHANDRAN; NIEMIER, 2005).

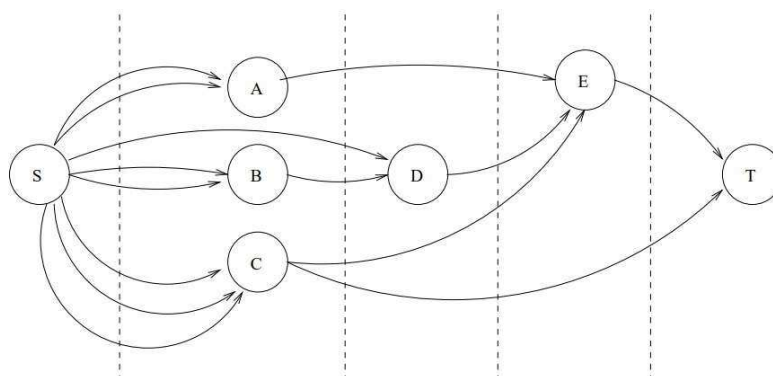


Figura 3.3 – Divisão de um circuito lógico em zonas de clock QCA. (ANTONELLI et al., 2004)

### 3.3 Roteamento

A tarefa de roteamento para a tecnologia QCA é dificultada pelo esquema de clock. Deve-se garantir que as rotas de duas entradas de uma mesma porta lógica possuam a mesma quantidade de zonas de clock. A figura 3.4 representa uma função lógica e demonstra o funcionamento de um algoritmo capaz de balancear a quantidade de zonas de clock em um circuito (LIM; RAVICHANDRAN; NIEMIER, 2005).

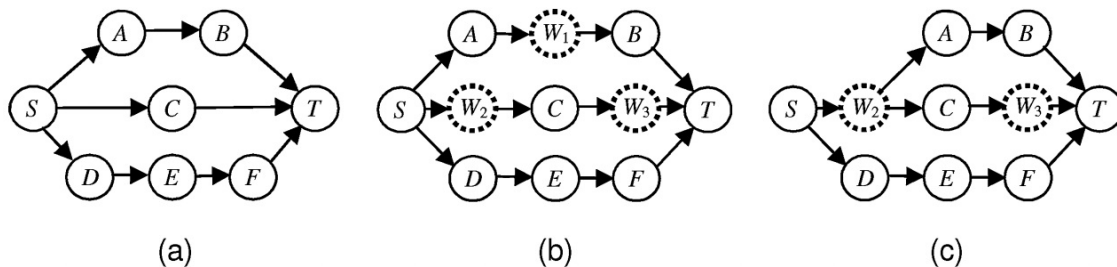


Figura 3.4 – Balanceamento do circuito. (LIM; RAVICHANDRAN; NIEMIER, 2005)

Inicialmente o grafo da figura 3.4 encontra-se desbalanceado por apresentar três caminhos reconvergentes e com tamanhos diferentes. Dois caminhos são ditos

reconvergentes quando possuem rotas distintas, porém a mesma fonte e destino. Num momento inicial, são inseridos vértices de fios para balancear o circuito. Agora, todos os caminhos reconvergentes transitam pelo mesmo número de vértices, portanto, estão balanceados. Posteriormente ocorre uma reordenação alterando as posições de (A) e (W1). Por fim (W1) e (W2) são convertidos em apenas um vértice. Esta supressão foi possível por si tratar de vértices de fios inseridos na etapa anterior.

A minimização de cruzamento de fios em QCA diminui a complexidade do roteamento. Uma heurística capaz de minimizar o cruzamento de fios foi implementada por (LIM; RAVICHANDRAN; NIEMIER, 2005). Seu funcionamento pode ser visto na figura 3.5. Um DGA é utilizado como estrutura de representação do circuito lógico. Inicialmente cada vértice é carimbado com um índice que indica o nível dentro do grafo. O nível de um vértice é definido pela profundidade do caminho entre o nó e as entradas do circuito. Cada nó será posicionado em uma zona de clock de acordo com seu nível. Então, são criados nós de fios para balancear as zonas de clock e evitar que zonas não sequenciais sejam vizinhas. Os vértices (d) e (c) são adjacentes, porém com zonas não sequenciais 0 e 2, assim foi criado o vértice (x) com zona de clock 1. Os nós são posicionados em quatro níveis que representam as respectivas zonas de clock. Finalmente ocorre uma reordenação de nós (a) e (d) como forma de evitar o cruzamento de fios.

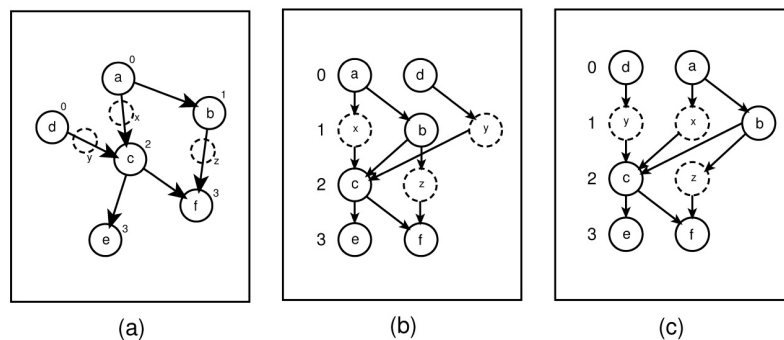


Figura 3.5 – Posicionamento de blocos de células. (LIM; RAVICHANDRAN; NIEMIER, 2005)

O trabalho apresentado em (LIM; RAVICHANDRAN; NIEMIER, 2005) não realiza a validação dos circuitos gerados na tecnologia QCA. É necessário realizar a análise das formas de onda dos circuitos gerados utilizando um simulador.

### 3.4 Layout

Um ponto a se considerar nos algoritmos de P&R para circuitos é a estrutura do dispositivo que receberá a impressão do circuito final. Aspectos relacionados à disposição dos elementos e ao sistema de clock precisam ser observados. Neste

âmbito, uma ferramenta denominada QCA-LG capaz de gerar o layout de circuitos combinacionais para a tecnologia QCA com formato compatível com o QCA Designer foi proposto em (TEODÓSIO; SOUSA, 2007).

Esta ferramenta executa seis tarefas principais:

- Leitura do circuito lógico: Foi construído um *parser* para carregar o circuito lógico a ser posicionado pela ferramenta. O *parser* identifica dois formatos distintos: a) LSI, reconhecido pela ferramenta de síntese RTL da Synopsys. b) Gate, formato usado pela ferramenta de síntese de circuito sequencial SIS (SENTOVICH et al., 1992).
- Expansão do circuito: Como forma de evitar a ocorrência de cruzamentos de fios, todo nó compartilhado é duplicado. Ao final desta etapa o *fanout* de cada nó é 1. A figura 3.6 apresenta o resultado desta etapa. Os nós (c) e (d) foram duplicados.

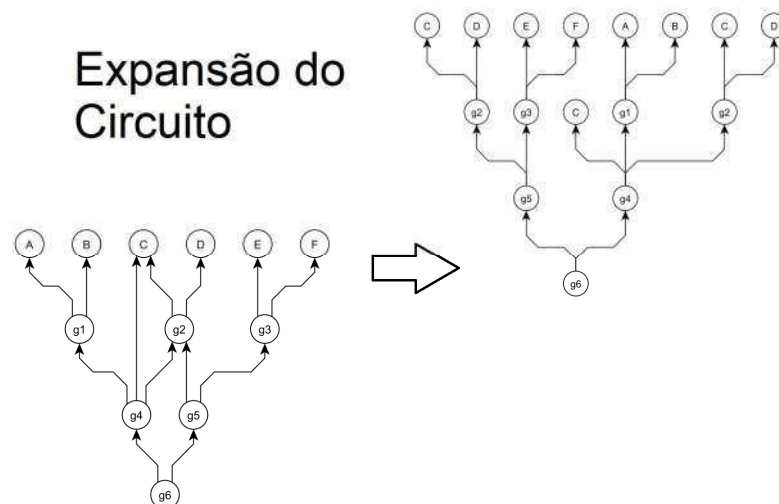


Figura 3.6 – Expansão do circuito. Vértices cinza são cópias. (TEODÓSIO; SOUSA, 2007)

- Posicionamento de portas: As portas do circuito são posicionadas de acordo com sua posição no grafo gerado pela etapa anterior. Cada porta vai receber uma coordenada física na matriz de células QCA.
- Modelagem de portas lógicas: De acordo com o tipo de porta lógica é gerado um esquema para simular seu comportamento. Portas lógicas *NOT* serão posicionadas diretamente na rota que interliga os nós. As portas *NAND* e *NOR* são geradas diretamente nas portas *AND* e *OR*. A figura 3.7 apresenta a formação de uma porta lógica em (TEODÓSIO; SOUSA, 2007). Nesta etapa há ainda uma sub-tarefa capaz de gerar as rotas que interligam os vértices do grafo.

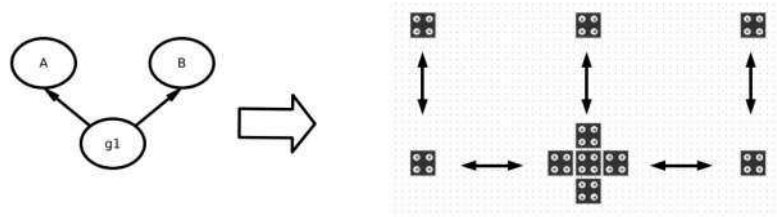


Figura 3.7 – Construção de layout de porta lógica da maioria. (TEODÓSIO; SOUSA, 2007)

- Distribuição de sinais de entrada: A segunda etapa é responsável por duplicar todos os nós que sejam compartilhados. Nesta etapa estes nós serão novamente interligados para encaminhar o sinal de entrada por todas as suas dependências. Na figura 3.8 pode-se visualizar o acréscimo de barramentos para interligar os nós duplicados (c) e (d).

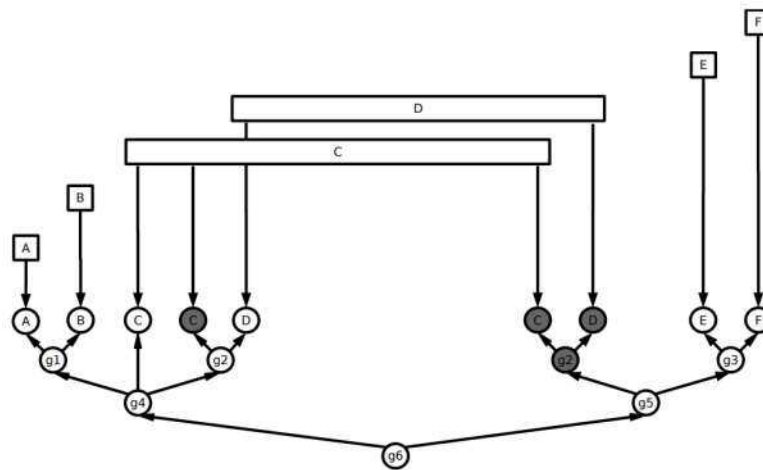


Figura 3.8 – Interconexão de nós duplicados. (TEODÓSIO; SOUSA, 2007)

- Layout de saída: Por fim, é gerado um arquivo seguindo o padrão da ferramenta QCADesigner com o layout do circuito.

A ferramenta QCA-LG não otimiza o espaço ocupado e gera layouts que ocupam muito recurso. Verifica-se na figura 3.9 a quantidade de espaço consumido pelas rotas do circuito gerado por tal ferramenta no layout de um multiplexador de duas entradas. Este problema deve ser contornado ao se considerar a construção de uma ferramenta profissional de P&R. O exemplo de um circuito *MUX 2x1* gerado pela ferramenta QCA-LG é exposto pela figura 3.9.

### 3.5 Comentários Finais

Em comparação com a ferramenta QCA-LG, esta dissertação apresenta uma heurística capaz de efetuar o posicionamento e roteamento de circuitos lógicos

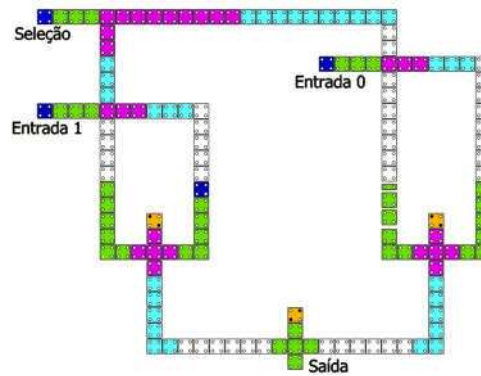


Figura 3.9 – Circuito gerado pela ferramenta QCA-LG. (TEODÓSIO; SOUSA, 2007)

com maior otimização de área. Esta economia é alcançada com a adoção de um esquema que constrói zonas de clock com formato regular e que permitem o fluxo de informações em todas as direções de uma matriz de células QCA bidimensional. Outra vantagem apresentada aqui é o posicionamento de portas lógicas e rotas de forma dinâmica, possibilitando a busca pela melhor posição de cada componente do circuito lógico em tempo de execução.

Os resultados experimentais demonstrados em (LIM; RAVICHANDRAN; NIEMIER, 2005) não apresentam a geração de circuitos QCA, dificultando sua comparação.

Esta dissertação traz uma série de circuitos QCA validados na ferramenta QCADesigner.

## 4 Algoritmo de Posicionamento e Roteamento

De acordo com discussões anteriores, a falta de ferramentas automáticas para o projeto de circuitos lógicos para QCA cria dificuldades para que pesquisadores possam gerar e validar novos conceitos para esta nanotecnologia.

Neste capítulo, será abordada uma heurística capaz de efetuar o posicionamento e roteamento de um circuito combinacional, em nível de porta lógica, por meio da alocação de células QCA seguindo o esquema de clock USE. Embora o foco deste trabalho seja o P&R de circuitos combinacionais, foi demonstrado em (CAMPOS et al., 2016) que o USE também possibilita o P&R de circuitos sequenciais.

Para garantir o mesmo comportamento funcional do nível lógico, o mapeamento na tecnologia QCA deve sincronizar todos os sinais usando um esquema de clock. Conforme já mencionado, é de suma importância que todas as restrições impostas pelo esquema de clock empregado sejam obedecidas. Desta forma, o esquema proposto em (CAMPOS et al., 2016) foi adotado por ser universal, escalável e regular. O posicionamento e roteamento de um circuito lógico proposto aqui, busca:

- Minimização de cruzamentos de fios;
- Minimização de células QCA empregadas na solução proposta;
- Posicionamento de entradas e saídas;
- Construção e validação de portas lógicas, E, OU e INVERSOR;
- Construção de uma árvore de soluções parciais para permitir a continuidade de execução mesmo que o algoritmo de P&R tenha encontrado uma solução inviável, sem que seja necessário reiniciar todo processo de P&R.

Todos estes pontos serviram de base para os algoritmos implementados neste trabalho.

A única ferramenta encontrada na literatura para efetuar o P&R no nível do layout para a tecnologia QCA (TEODÓSIO; SOUSA, 2007) gera circuitos com rotas que consomem muitas células. Durante o posicionamento e roteamento das células deve-se evitar cruzamento de fios e ao mesmo tempo minimizar a área total. A figura 4.1, expõe dois posicionamentos possíveis para o circuito apresentado. O posicionamento da figura (a) não faz uso do cruzamento de fios, porém utiliza duas

células QCA a mais em relação ao posicionamento da figura (b). Num primeiro momento, este acréscimo pode parecer superficial, porém, de acordo com o aumento do circuito ele será cada vez maior. Quando é identificado a presença de duas rotas distintas de mesmo comprimento, uma com cruzamento e outra sem, a heurística apresentada adota a rota sem cruzamento. Porém, quando esta situação não ocorrer e forem identificadas apenas rotas com cruzamento, estas são imediatamente adotadas.

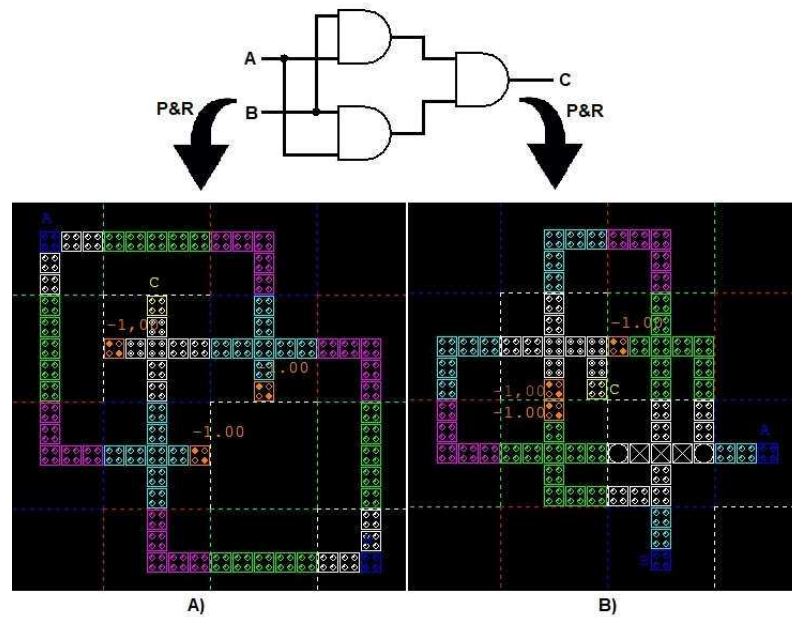


Figura 4.1 – Posicionamento manual. (a) Sem cruzamento. (b) Com cruzamento.

Este trabalho utilizou um esquema padrão de clock USE que faz uso de um espaço de 5 linhas e 5 colunas que poderá comportar até 25 células QCA organizadas em forma de uma matriz quadrada para cada zona de clock. Este padrão pode ser alterado ao escolher matrizes quadradas de qualquer ordem. Assim sendo, existem algumas formas diferentes para se projetar uma porta da maioria ou uma negação no espaço escolhido de 5x5. Porém, alguns destes layouts podem não responder como esperado. Então, foram testados manualmente alguns dos layouts possíveis para estas portas. A figura 4.2, demonstra algumas formas distintas de organização da porta da maioria, porém os três layouts da parte inferior (b) da figura não executam o comportamento funcional do nível lógico corretamente. Isto se deve pelo fato de haver entradas que possuem quantidade distinta de células em sua rota. Quando uma entrada possui menos células no caminho para a célula de chaveamento (central) da porta da maioria, seu sinal leva menos tempo para alcançar esta célula e a computação não ocorre como se deve, assim sendo, o sinal de saída da porta da maioria é estabelecido unicamente por esta entrada.

A solução proposta é uma heurística gulosa que percorre o circuito das saídas em direção as entradas. O posicionamento e roteamento é realizado por nível, que

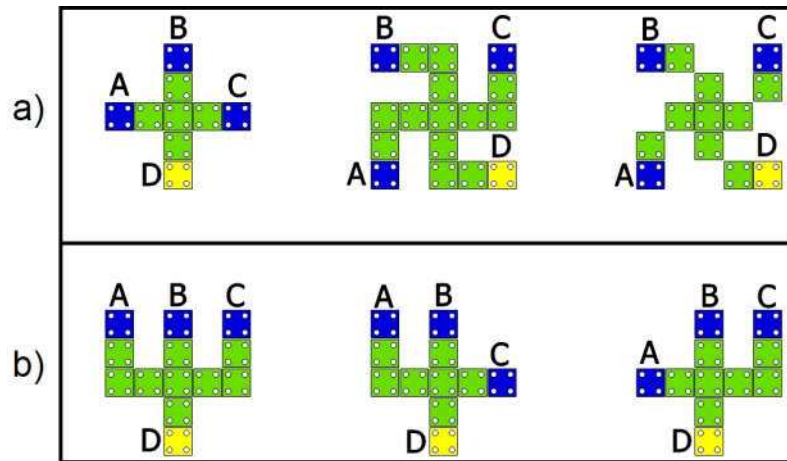


Figura 4.2 – Modelos de layout de porta lógica da maioria no esquema de clock USE com tamanho de zona 5x5.

é a profundidade a partir das saídas. Se durante o P&R de todas as portas de um determinado nível não existir um posicionamento ou um roteamento possível, o algoritmo retorna ao nível anterior e refaz o posicionamento do nível acima, para depois buscar novamente no nível atual. Portanto, é uma heurística gulosa local que faz uma exploração parcial da árvore de solução.

## 4.1 Formulação do Problema

A etapa de posicionamento consiste em definir o local de cada componente de um circuito em nível de porta lógica em uma matriz de células USE. O roteamento é executado por meio da construção de rotas que interligam cada componente do circuito lógico.

O circuito é representado por meio de um grafo direto acíclico (DGA)  $G(V,E)$ . O conjunto de vértices é representado por  $V$ . Cada vértice é equivalente a uma entrada, saída, fio ou porta lógica do circuito. O símbolo  $E$  retrata o conjunto de arestas. Cada aresta no grafo denota uma conexão entre os diversos elementos do circuito. Duas rotas  $r_1$  e  $r_2$  são consideradas reconvergentes caso elas compartilham os mesmos vértices em suas extremidades e possuem conjuntos de vértices intermediários distintos. A figura 4.3 demonstra um modelo de grafo DGA que representa um dado circuito lógico.

A natureza do esquema USE faz com que as zonas de clock sejam organizadas em forma de uma matriz  $M \times N$ . Todas as zonas têm formato regular e sua distribuição pelas células da matriz dá a possibilidade de transmissão de sinal em todas as direções. A figura 4.4 apresenta um modelo do USE em forma de matriz. De forma intercalada, cada linha desta matriz possibilita o envio de um sinal da esquerda para a direita e vice-versa. As linhas tracejadas na figura representam este fluxo de dados. Da mesma

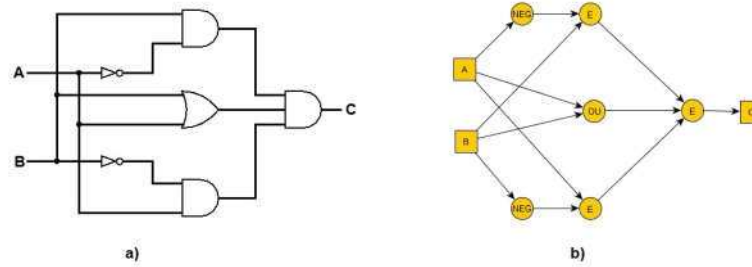


Figura 4.3 – Modelo de grafo DGA

maneira, as colunas permitem a transmissão de sinal de baixo para cima e vice e versa, alternadamente.

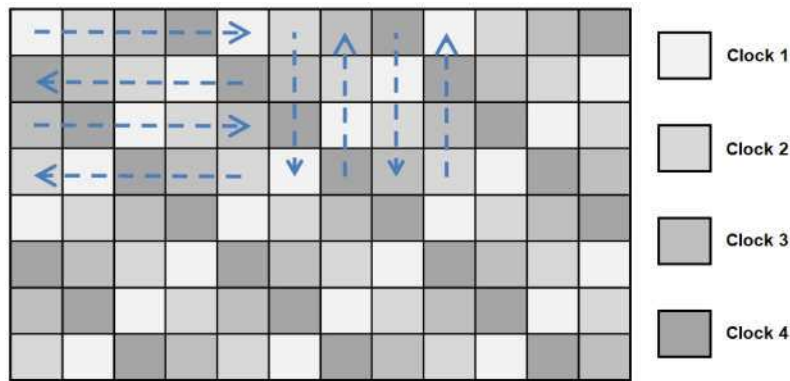


Figura 4.4 – Representação do fluxo de dados no esquema de clock USE

Este esquema faz com que cada célula da matriz seja vizinha de outras quatro células. A organização intercalada do fluxo de sinal na horizontal e vertical permite que cada célula possa transmitir e receber e receba uma informação por meio de duas de suas células vizinhas. A figura 4.5 é uma representação em formato de grafo da figura 4.4. Cada vértice representa uma zona da matriz QCA. As setas indicam o sentido do fluxo de informação de acordo com o clock de cada vértice. O número de cada vértice é a configuração inicial de cada zona de clock. Nota-se que todo vértice que não está na borda do grafo possui duas arestas que chegam de seus vizinhos e duas arestas que saem em direção a seus vizinhos.

## 4.2 Mapeamento Tecnológico

A síntese gera um circuito lógico que pode ter portas com N entradas e N saídas. O primeiro passo, ilustrado na figura 4.6, é o mapeamento tecnológico que consiste em adequar as portas do circuito a portas que estejam disponíveis na tecnologia alvo. No caso desta dissertação temos a restrição de mapear para módulos QCA vértices com duas entradas e duas saídas. Portanto, o mapeamento tecnológico irá transformar o circuito, fazendo a decomposição das portas que tenham mais

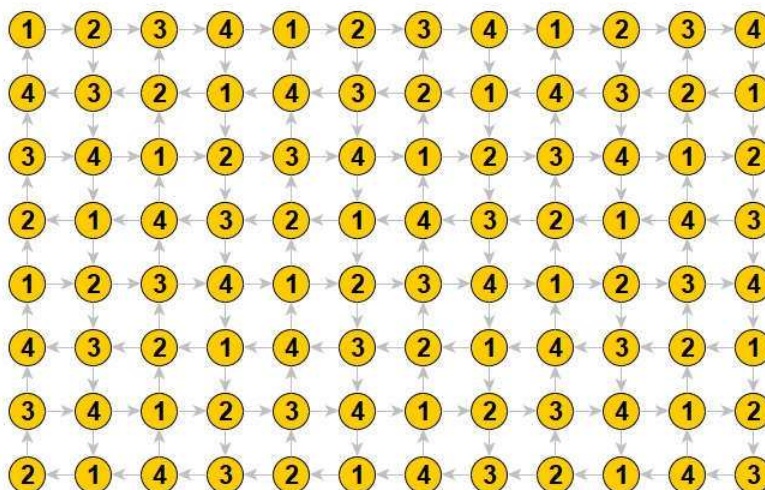


Figura 4.5 – Representação em formato de grafo do esquema de clock USE

entradas e saídas para respeitar esta restrição. Usamos portas *E*, *OU* e *Inversor* como as primitivas básicas. No formato *And-Inverter Graphs* (AIG) as portas contém no máximo duas entradas e duas saídas (SYNTHESIS, 2007). Seguindo este exemplo, o grafo da figura 4.3 deve ter o vértice *E* mais a direita (3 entradas) convertido em dois vértices com duas entradas cada. A figura 4.6 (b) demonstra tal processo. As linhas tracejadas indicam a alteração ocorrida no grafo para alcançar o formato desejado. Outra alteração ocorre nas duas entradas que apresentam três saídas. Um vértice de *fanout* é criado para concentrar duas saídas da entrada.

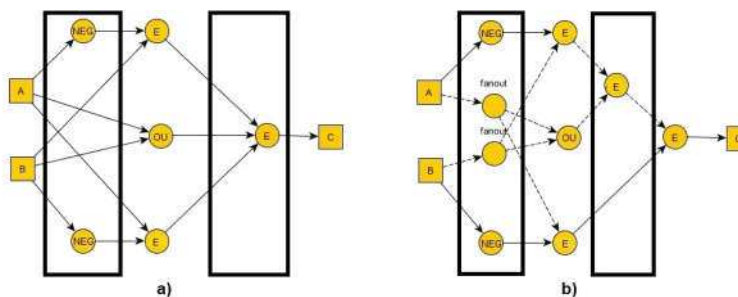


Figura 4.6 – Transformação do circuito num grafo AIG de no máximo 2 entradas e 2 saídas.

Após o mapeamento tecnológico, o segundo passo é o balanceamento por níveis que é o ponto de partida para o algoritmo de posicionamento e roteamento. Seguindo como exemplo o grafo da figura 4.6 (b), esta tarefa consiste em enumerar sequencialmente todos os membros que se encontram no caminho crítico do circuito lógico. Levando em consideração que o atraso total do circuito QCA vem da quantidade de zonas de clock que este circuito apresenta, o caminho crítico de cada circuito é inicialmente representado pela rota que conecta uma entrada de uma saída com mais vértices intermediários. A figura 4.7 demonstra tal tarefa. O número que cada vértice do caminho crítico recebe representa o nível deste vértice no grafo.

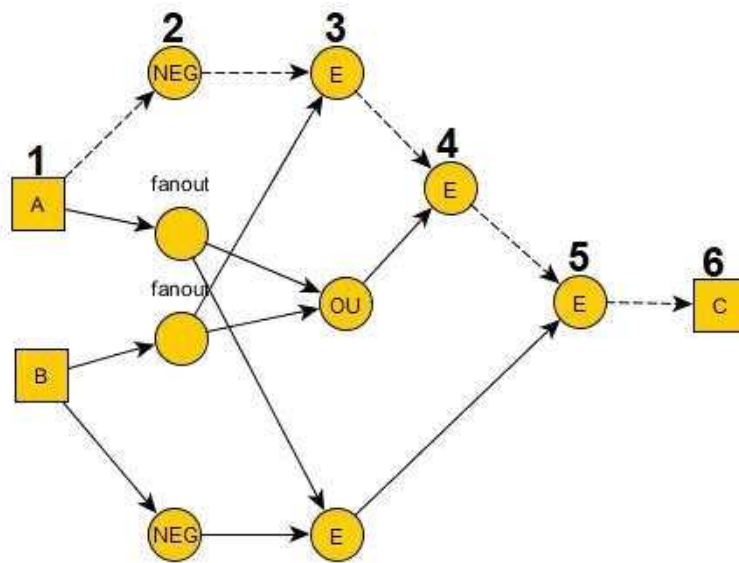


Figura 4.7 – Numeração de vértices de acordo com profundidade no grafo.

O restante do circuito deve ser enumerado de acordo com o caminho crítico. Todos os vértices são visitados da(s) entrada(s) para a(s) saída(s) e cada vértice que não possui nível definido será posicionado no próximo nível do seu vértice antecessor. A figura 4.8 traz o resultado deste processo.

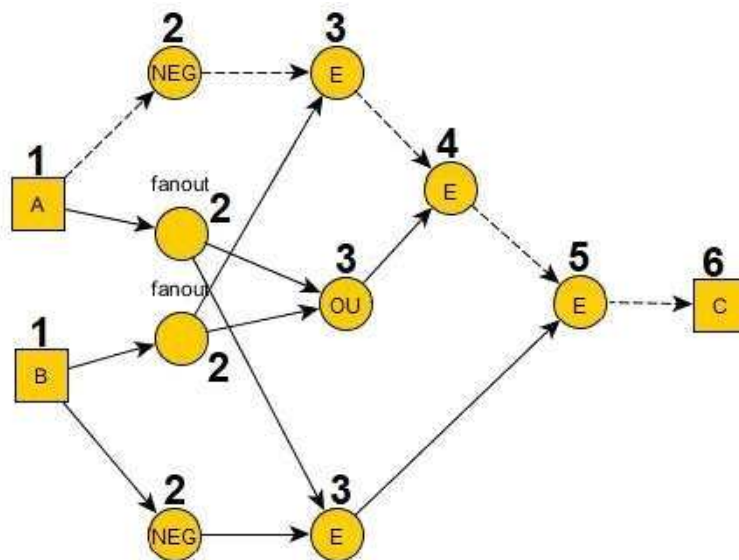


Figura 4.8 – Classificação de vértices em níveis.

Depois que todos os vértices são rotulados com um nível no grafo, auxiliando a etapa de posicionamento, são criados vértices de fio para balancear vértices diretamente conectados. Esta alteração auxilia a etapa de roteamento fazendo com que todas as rotas tenham o mesmo tamanho. A figura 4.9 apresenta a criação de mais um vértice no nível quatro, dado que, um vértice do nível três estava diretamente ligado à um vértice do nível cinco.

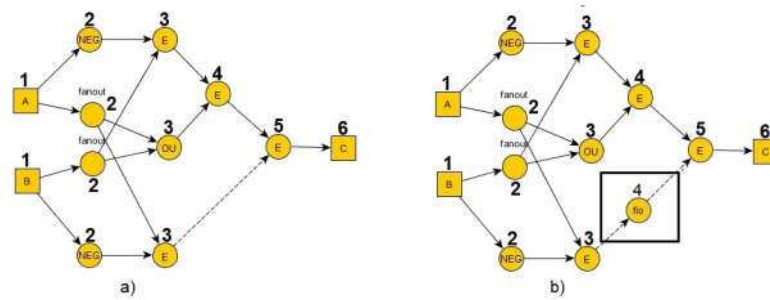


Figura 4.9 – Inserção de vértice de balanceamento.

### 4.3 Algoritmo

Esta seção expõe a implementação em fluxograma dos algoritmos de posicionamento e roteamento de circuitos lógicos para a tecnologia QCA.

#### 4.3.1 Entrada

A entrada é o circuito em nível de porta lógica em forma de um grafo AIG. O formato GML (*e Graph Modelling Language* (HIMSOLT, 1997)) é utilizado para modelar o grafo que representa o circuito. A função de cada vértice (entrada, saída, fio ou porta lógica *Inversor* e *E*) é estabelecida a partir de seu rótulo.

Um segundo arquivo em formato de texto simples (ASCII) é construído para definir o nível em que cada vértice se encontra no grafo. A primeira linha deste arquivo indica o número de vértices  $v$  e níveis  $n$ . As  $n$  linhas seguintes são iniciadas com um valor representando a quantidade de vértices daquele nível seguido pelo índice de cada vértice. A figura 4.10 é uma representação gráfica dos dois arquivos de entrada.

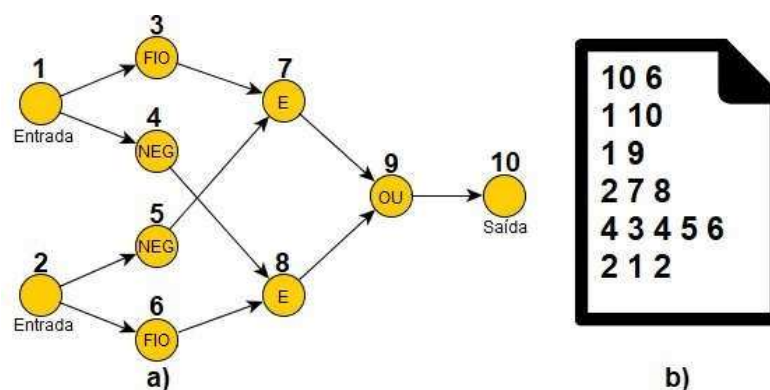


Figura 4.10 – Representação gráfica dos dois arquivos de entrada para a heurística P&R.

O formato de entrada pode ser facilmente adaptado para receber a descrição do circuito em Verilog ou formato BLIF ou outro equivalente, sendo a etapa de

mapeamento tecnológico responsável pelas tarefas de decomposição do circuito em portas de 2 entradas e 2 saídas, bem como a geração dos níveis iniciais para o P&R com o balanceamento.

### 4.3.2 Fluxograma

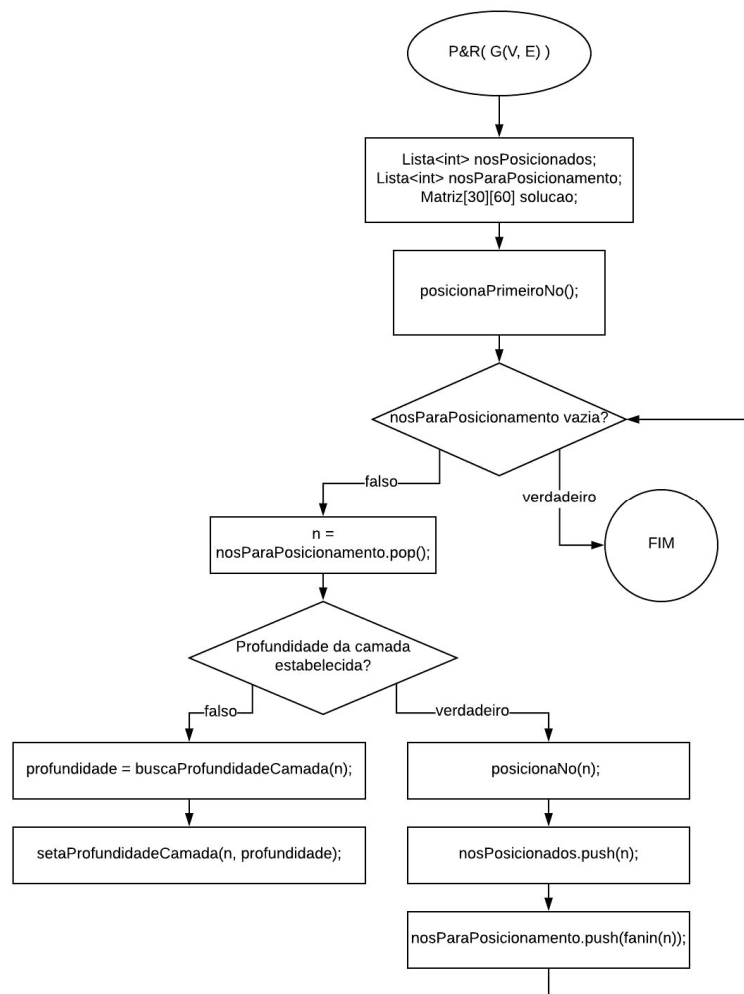


Figura 4.11 – Fluxograma para a heurística P&R.

Nesta seção será apresentado o algoritmo responsável em mapear um circuito em nível de porta lógica dentro de uma matriz de células USE. A representação em fluxograma do algoritmo de P&R é exposta pela figura 4.11. A tarefa consiste em posicionar célula por célula e estabelecer rotas respeitando as restrições do QCA com relação ao clock. Os principais objetivos deste algoritmo são: minimizar consumo de área, minimizar quantidade de células QCA e apresentar respostas em tempo hábil. O posicionamento e roteamento são executados simultaneamente.

O algoritmo recebe um grafo de vértices e arestas que representa o circuito a ser posicionado.

São utilizadas três estruturas de dados para chegar à solução final. A lista *nosPosicionados* irá armazenar os vértices que já foram posicionados, a lista *nosParaPosicionamento* é utilizada para ordenar os próximos vértices à posicionar e a matriz de células USE irá receber a solução do layout proposto de acordo com a ferramenta QCADesigner.

Inicialmente a função *posicionaPrimeiroNo()* é executada com a responsabilidade de inicializar as estruturas de dados e posicionar o primeiro vértice no centro da matriz USE.

Neste momento o algoritmo entra num *looping* encarregado de posicionar todos os vértices restantes. A cada novo vértice à posicionar é indicado o tamanho de sua rota (profundidade) no layout. A principal restrição a ser respeitada deve ser a uniformidade na profundidade das rotas dos vértices de mesmo nível.

A função *buscaProfundidadeCamada(n)* deve ser capaz de encontrar a melhor profundidade de rota para todos os vértices de um nível. Esta tarefa retorna como profundidade do nível o pior caso no posicionamento entre todos os vértices do nível atual. Este passo garante a possibilidade de posicionamento de todos os vértices do nível com mesma profundidade.

A função *posicionaNo(n)* efetua o posicionamento do vértice na matriz de células USE respeitando a profundidade atribuída pela etapa anterior. Se o posicionamento for possível de ser realizado, todas as estruturas do algoritmo são atualizadas. Caso não seja possível realizar o posicionamento de um vértice dada a profundidade do nível, todos os vértices daquele nível são retirados das estruturas de dados e o tamanho de rota daquele nível é incrementado em uma unidade.

Este processo segue até que todos os vértices do grafo tenham sido posicionados.

### 4.3.3 Execução Passo a Passo

O algoritmo percorre o grafo do circuito em largura (por nível) da saída em direção as entradas, realizando por nível o P&R. Aqui, a busca parte da posição dos vértices do nível superior em direção à um local livre na matriz de células QCA. A figura 4.12 apresenta o grafo do circuito *XOR* (a) e os passos da busca por uma posição válida para o vértice 1. Este vértice é ligado à outros dois vértices (3 e 4) do próximo nível e deve ter rotas que os conectem com o mesmo comprimento. A figura 4.12 (b) apresenta um estado válido para o posicionamento dos vértices dos níveis 2 à 5 do grafo da figura 4.12 (a).

Deve-se encontrar uma posição para o vértice 1 com rotas de mesmo comprimento em direção aos vértices 3 e 4. Como a busca parte das saídas para as entradas,

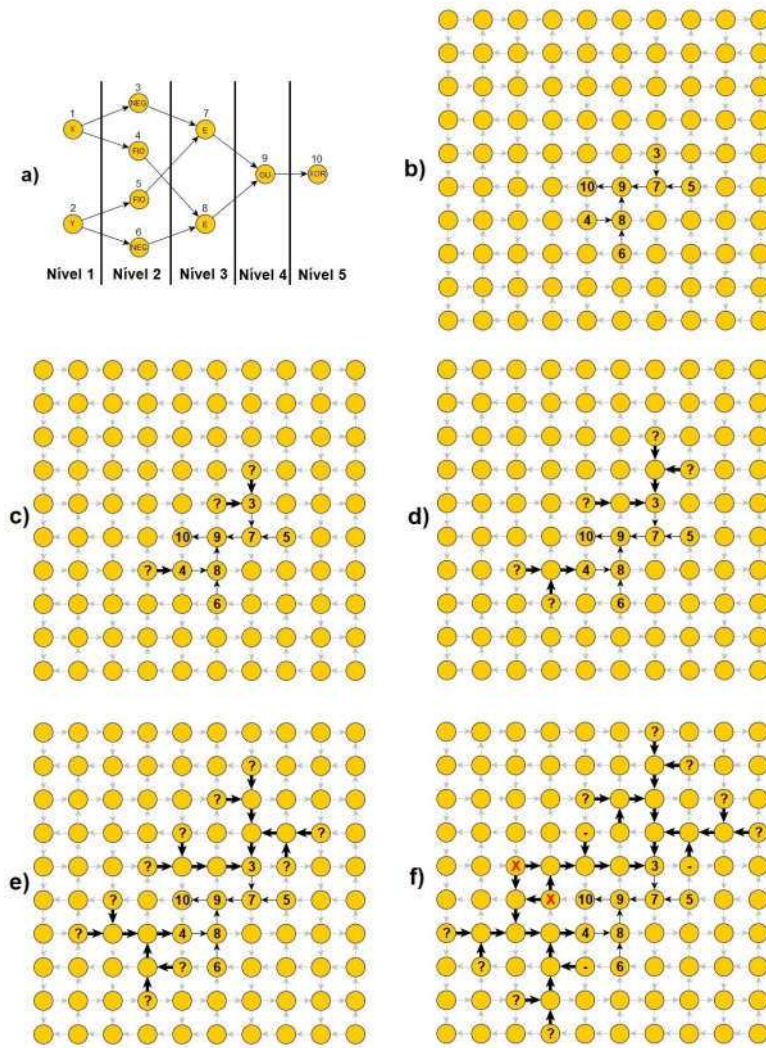


Figura 4.12 – Passo a passo da busca pela posição do vértice 1. a) Representação gráfica do circuito lógico à ser posicionado. b) Possível posicionamento para os vértices dos níveis 2, 3, 4 e 5. c) Busca por rotas de tamanho 1 para o posicionamento do vértice 1 com *fanout* nos vértices 3 e 4. d) Busca por rotas de tamanho 2 para o posicionamento do vértice 1 com *fanout* nos vértices 3 e 4 e) Busca por rotas de tamanho 3 para o posicionamento do vértice 1 com *fanout* nos vértices 3 e 4 f) Rotas de tamanho 4 encontradas para o vértice 1 com *fanout* nos vértices 3 e 4.

o algoritmo inicia das posições dos vértices 3 e 4. A figura 4.12 (c) expõe três vértices rotulados com o carácter interrogação indicando posições livres que se conectam aos vértices 3 e 4 com comprimento um. Porém, nenhuma das três rotas encontradas se interceptam. Nas figuras 4.12 (d) e (e) a busca é ampliada incrementando o comprimento das rotas gradualmente. A busca termina quando existe uma interseção entre o conjunto de soluções de posicionamento com distância  $D$  (no caso 4) para o vértice 1 que conecta no vértice 3 com o conjunto de soluções para a ligação do vértice 1 como entrada do vértice 4. Podemos observar estas posições marcadas com um X na Figura 4.12 (f).



A figura 4.14 apresenta a ordem de posicionamento de vértices de um grafo. O posicionamento inicia da saída (vértice 9) e vai se desenvolvendo em cada nível do grafo até que o último vértice do primeiro nível (vértice 2) tenha sido posicionado.

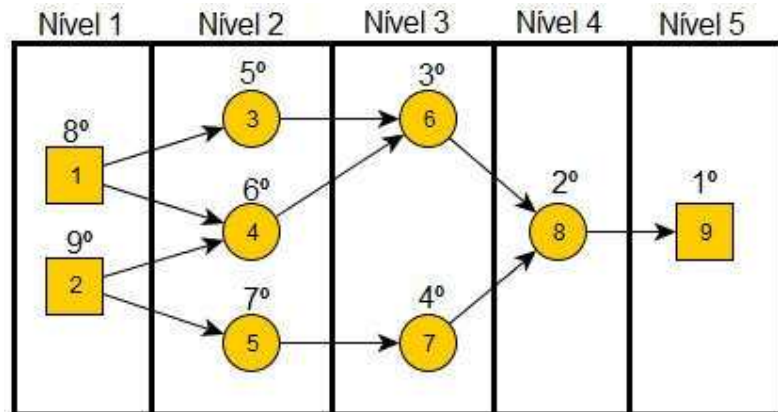


Figura 4.14 – Ordem de posicionamento de vértices.

Uma função importante no algoritmo de P&R consiste em encontrar o posicionamento de um vértice que gere o caminho mais curto. Dentro desta função é necessário avaliar a quantidade de rotas que incidem diretamente no vértice a ser posicionado. Cada vértice pode estar ligado a um ou dois outros vértices. O vértice que será posicionado é denominado de filho e os vértices do nível superior com rotas que incidem diretamente nele são considerados seus pais. Na figura 4.15, o vértice 6 é pai dos vértice 3 e 4, assim como o vértice 1 é considerado filho dos vértices 3 e 4.

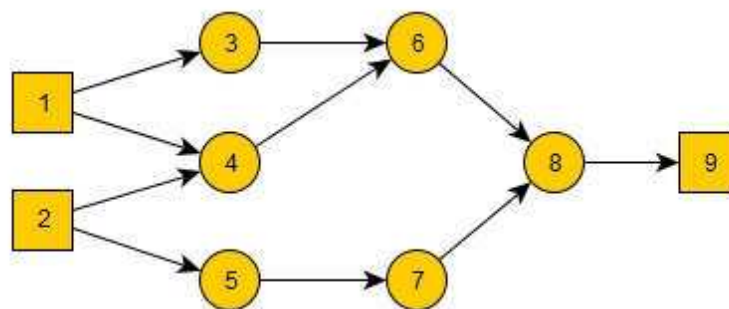


Figura 4.15 – Definição de nomeclatura de vértices. Vértice 6 é pai dos vértice 3 e 4. Vértice 1 é filho dos vértices 3 e 4.

A função de busca de mínima rota começa identificando quantos pais o vértice possui. Será necessário encontrar tantas rotas distintas quantos pais o vértice tiver. Inicialmente, rotas de comprimento 1 começam a ser mapeadas. Se não existir nenhuma rota válida o parâmetro de busca é incrementado e rotas com comprimento 2 são pesquisadas. Caso a quantidade necessária de rotas válidas seja encontrada a função retorna o comprimento encontrado.

A figura 4.16 demonstra o exemplo de posicionamento de um circuito. Em certo momento todos os vértices do nível 2 ao 5 já foram posicionados. Restam dois vértices do nível 1 à posicionar. Esta função inicia uma busca pelas rotas de menor tamanho para o posicionamento dos vértices 1 e 2. É encontrado um local para o posicionamento do vértice 1 que gera rotas de comprimento mínimo 3 e que incidem nos vértices 3 e 4. Outros dois locais possíveis com rotas de comprimento mínimo 3 são encontrados para posicionar o vértice 2. Neste caso, o nível será marcado com o comprimento da maior rota.

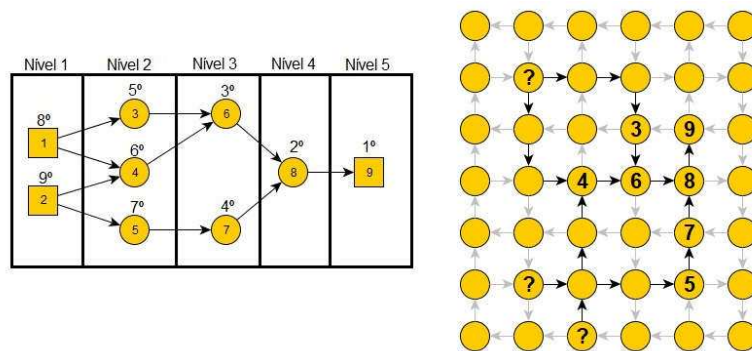


Figura 4.16 – Exemplo de busca por rota de comprimento mínimo.

O fato da posição de um vértice afetar diretamente sua rota justifica a definição imediata de sua rota no momento de seu posicionamento. Estas tarefas são exploradas na função a seguir.

As próximas figuras trazem um exemplo completo do posicionamento e roteamento do circuito lógico XOR. Seu esquema é apresentado pela figura 4.17 (a). O grafo do circuito já balanceado com indicação de função e numeração de cada vértice é trazido pela figura 4.17 (b).

A figura 4.18 apresenta um conjunto de imagens que demonstram passo a passo a execução do algoritmo de P&R. Na figura (a) o vértice 10 (raiz) já foi posicionado e existem duas possibilidades de posicionamento do vértice 9 com comprimento de rota 1. Os vértices 7 e 8 devem ser posicionados com rotas em direção ao vértice 9. São encontrados dois locais que geram comprimento de rota 1 para posicionar os vértices 7 e 8. A figura (c) apresenta os locais de posicionamento dos vértices 7 e 8. Neste momento devem ser posicionados os vértices (3, 4, 5 e 6) do nível 2. Também existem quatro locais possíveis para receber quatro vértices, logo não há grande dificuldade em seus posicionamentos. A figura (d) mostra os locais escolhidos para posicionar os vértices 3, 4, 5 e 6.

O posicionamento dos vértices do último nível é apresentado na figura (e), porém, anteriormente na figura (d), pode-se notar que o melhor posicionamento para o vértice 1 gera rotas de comprimento 4, enquanto que o posicionamento do vértice

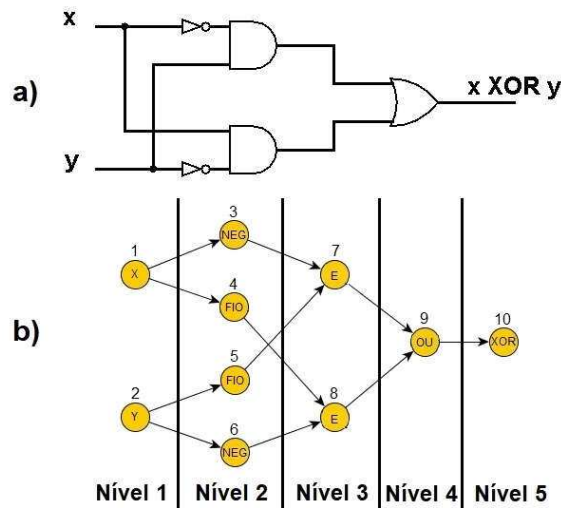


Figura 4.17 – (a) Circuito  $XOR$  em nível de porta lógica. (b) Grafo do circuito  $XOR$  gerado para entrada da heurística.

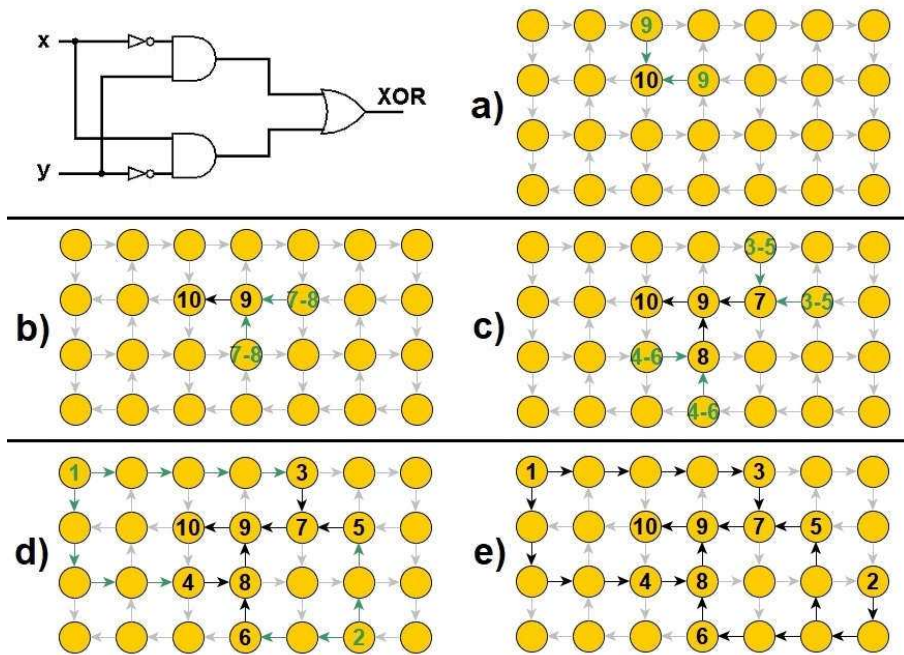


Figura 4.18 – Passo a passo da execução da heurística de P&R para o circuito  $XOR$ .

2 gerou rotas de comprimento 2. Como deve-se garantir que todos os vértices do mesmo nível tenham rotas de tamanho idêntico, o vértice 2 foi posicionado em um local que também gerou rota de tamanho 4. Este posicionamento é visto na figura (e).

Algo a se analisar é que se os vértices 3 e 5 tiverem suas posições invertidas, certamente a posição dos vértices do próximo nível seriam afetadas. Isto se deve pelo fato de que é necessário buscar rotas saindo do vértice 1 e em direção aos vértices 3 e 4. Como o vértice 3 teve sua posição alterada, logicamente poderá haver modificação no posicionamento do vértice 1. Esta alteração pode levar à uma melhora ou piora na

solução. Tal situação pode ser explorada por meio da árvore de soluções do algoritmo que armazena todas as possibilidades de P&R.

A figura 4.19 apresenta o projeto da porta XOR em QCA seguindo a resultado gerado pela heurística de trabalho. Os vértices do grafo da figura 4.19 foram demarcados no circuito. As setas indicam a direção do fluxo de dados pelo circuito.

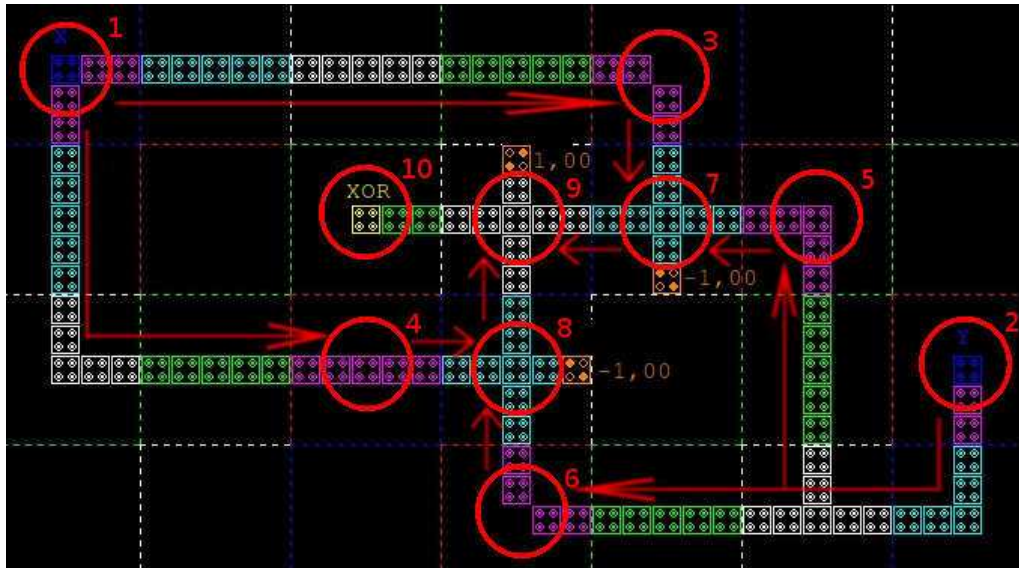


Figura 4.19 – Resultado do P&R para o XOR no QCADesigner.

A figura 4.20 expõe a forma de onda apresentada pelo QCADesigner para o circuito da figura 4.17. Devido ao atraso gerado pelo circuito, a saída só pode ser coletada depois de dois pulsos de clock em relação às entradas. Este atraso é indicado pelas setas que informam as entradas relativas para cada saída.

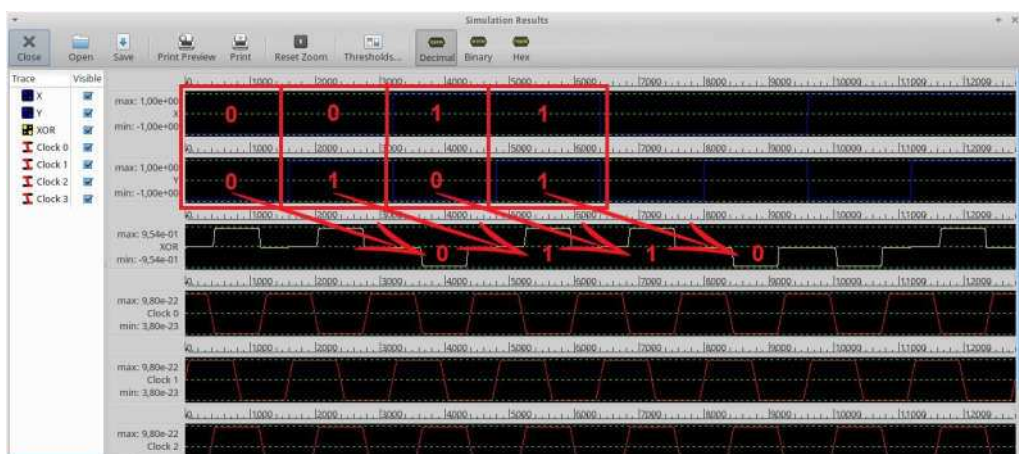


Figura 4.20 – Forma de onda do circuito XOR da figura 4.19.

## 5 Estudo de Casos

Os resultados alcançados neste trabalho são apresentados nesta seção. Para avaliar o algoritmo e comparar com o trabalho relacionado (TEODÓSIO; SOUSA, 2007), foram selecionados os seguintes circuitos disponibilizados também nas referências: MUX 2x1 (CAMPOS et al., 2016) e Full-adder 1-bit (CAMPOS et al., 2016). Ainda foram gerados *layouts* para os circuitos *XOR*, *XNOR*, *Parity Generator* e *Parity Checker*.

Entretanto, deve-se levar em consideração que a maioria dos resultados gerados nos outros trabalhos da literatura [(CAMPOS et al., 2016), (LIM; RAVICHANDRAN; NIEMIER, 2005), (JAGARLAMUDI; SAHA; JAGARLAMUDI, 2011), (AHMAD; BHAT, 2012)] foram obtidos por meio do posicionamento e roteamento manual com ajustes ad-hoc das zonas de clock. Ou seja, as outras abordagens não são escaláveis e/ou não são triviais para automatizar.

Os resultados apresentados foram validados por meio da ferramenta QCA-Designer utilizando o mecanismo de simulação por aproximação de dois estados. A figura 5.1 exibe as opções configuradas no mecanismo de simulação da ferramenta. A célula QCA foi configurada com largura e altura igual a 18,0 nanômetros e diâmetro de dot medindo 5,0 nanômetros.

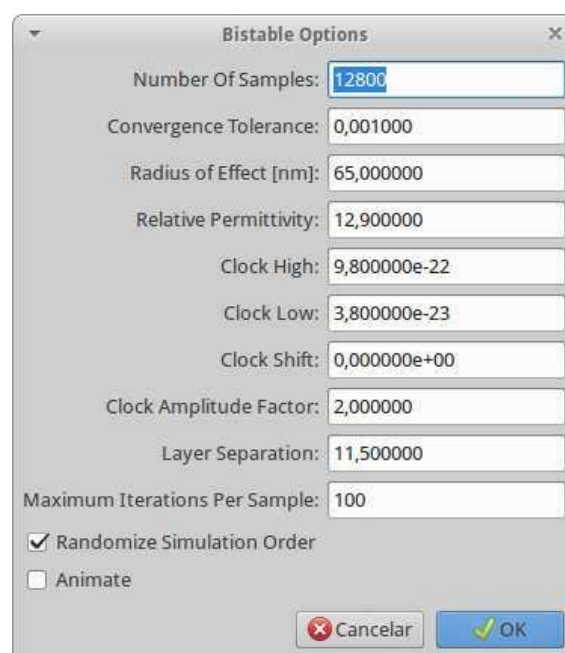


Figura 5.1 – Configuração do mecanismo de simulação no QCADesigner.

Inicialmente, dois exemplos serão apresentados detalhadamente para demonstrar os resultados alcançados. Um *MUX 2x1* e um *Full Adder de 1-bit*.

## 5.1 MUX 2x1

O primeiro exemplo que este trabalho traz é o posicionamento e roteamento de um circuito *MUX 2x1* para a tecnologia QCA. A figura 5.2 a) demonstra o esquema do circuito *MUX 2x1* em nível de porta lógica. O grafo da figura 5.2 b) apresenta em seu nível-2, quatro vértices. Três destes vértices foram criados para tornar o grafo regular a facilitar a etapa de roteamento. Estes vértices recebem o nome de *WIRE* por serem apenas um ponto intermediário do fio que interliga os componentes do circuito lógico.

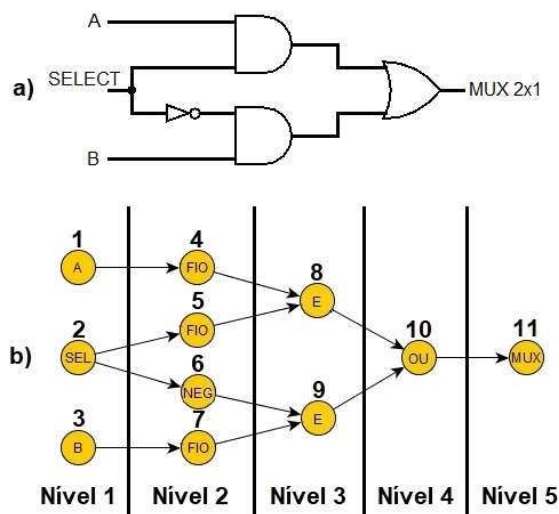


Figura 5.2 – (a) Circuito *MUX* em nível de porta lógica. (b) Grafo do circuito *MUX* gerado para entrada da heurística.

A figura 5.3 mostra o processo de P&R passo a passo. Importante perceber no posicionamento dos vértices 1, 2 e 3 é que existiam rotas de comprimento distintos no momento da busca pelo melhor posicionamento de cada vértice. Como todos estes vértices compartilham o mesmo nível, é necessário tomar como parâmetro na construção de rotas para todos os elementos deste nível o comprimento estabelecido pelo pior caso, ou aquele que gerou maior rota. Neste caso, o vértice 2 teve rota de comprimento dois e os vértices 1 e 3 tiveram rota de comprimento um. Logo, os posicionamentos dos vértices 1 e 3 foram relaxados, a fim de manter o sincronismo no circuito.

A figura 5.4 apresenta o resultado do posicionamento e roteamento de células QCA trazidas pela heurística exposta neste trabalho. Como no exemplo anterior, os vértices e as conexões do grafo da figura 5.2 b) foram indicados na figura.

A figura 5.5 representa o posicionamento do circuito *MUX 2x1* na ferramenta implementado em (TEODÓSIO; SOUSA, 2007).

As formas de onda que indicam o funcionamento do circuito são vistas na

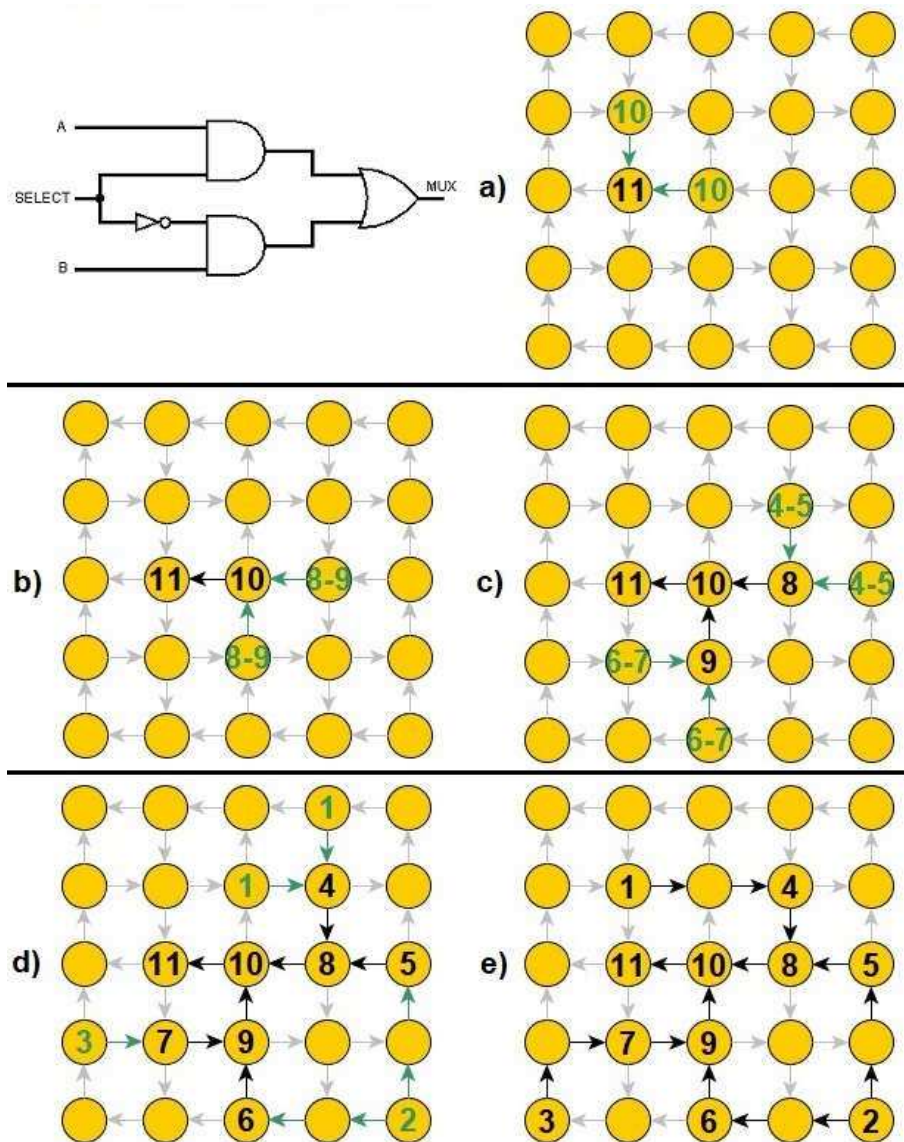


Figura 5.3 – Passo a passo da execução da heurística de P&R para o circuito *MUX 2x1*.

figura 5.6.

## 5.2 Full Adder de 1-bit

O circuito lógico que representa um *Full Adder de 1-bit* é apresentado como segundo exemplo. Na figura 5.7, nota-se a existência de dois cruzamentos de fios. Os cruzamentos dificultam o processo do P&R de circuitos por requerem, em diversas vezes, que dois fios caminhem paralelamente por uma zona de clock do USE.

As tomadas de decisão feitas pela heurística que com maior importância no P&R do circuito são apresentadas na figura 5.8. A figura a) ilustra o posicionamento dos vértices de nível-4. Neste momento é roteado o primeiro cruzamento de fios do

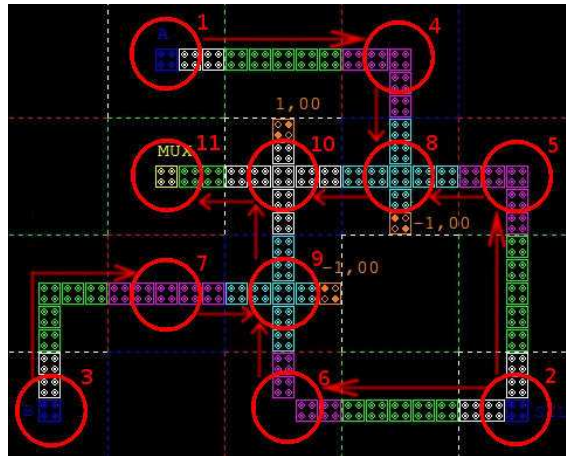


Figura 5.4 – Resultado do P&R para o *MUX* no QCADesigner.

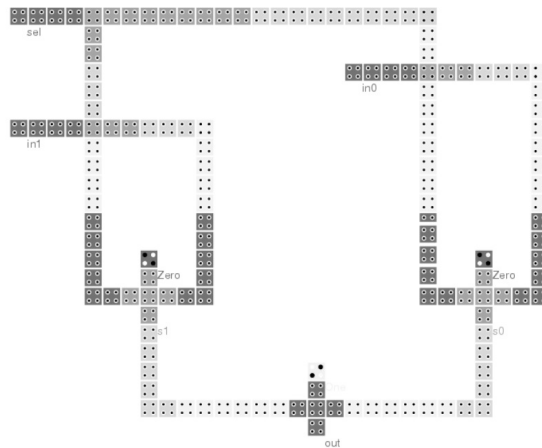


Figura 5.5 – P&R do circuito *MUX*  $2 \times 1$  na ferramenta QCA-LG (TEODÓSIO; SOUSA, 2007).

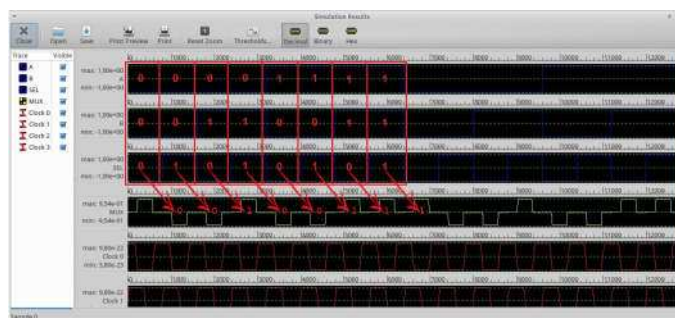


Figura 5.6 – Forma de onda do circuito *MUX*  $2 \times 1$  da figura 5.3.

circuito. Ao buscar pelo melhor posicionamento dos vértices o algoritmo encontra duas rotas de comprimento dois. Entretanto, caso o vértice 11 seja posicionado no local indicado pela figura a) ele não poderá receber informação de nenhum outro vértice, pois os dois caminhos de entrada de dados deste local estão ocupados pelos vértices 14 e 15. Este situação levará a uma barreira no construção do restante do circuito. Desta forma, a heurística identifica este falha e acresce o comprimento da

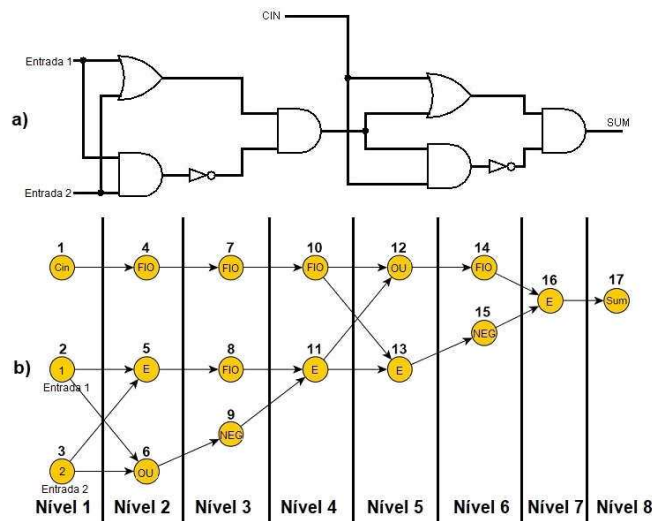


Figura 5.7 – (a) Circuito *Full Adder de 1-bit* em nível de porta lógica. (b) Grafo do circuito *Full Adder de 1-bit* gerado para entrada da heurística.

rota dos vértices 10 e 11 em uma unidade. A figura b) apresenta o posicionamento e roteamento deste vértices com comprimento de rota maior. O posicionamento do segundo cruzamento de fios do circuito é demonstrado na figura c). Neste caso o vértice 2 estará impedido pelos vértices 8 e 9 de receber dados. Porém, como o vértice 2 é uma entrada ele não precisará receber nenhuma informação de nenhum outro local. Desta forma, não há necessidade de relaxamento no comprimento de sua rota. A figura d) apresenta o resultado final no posicionamento do circuito.

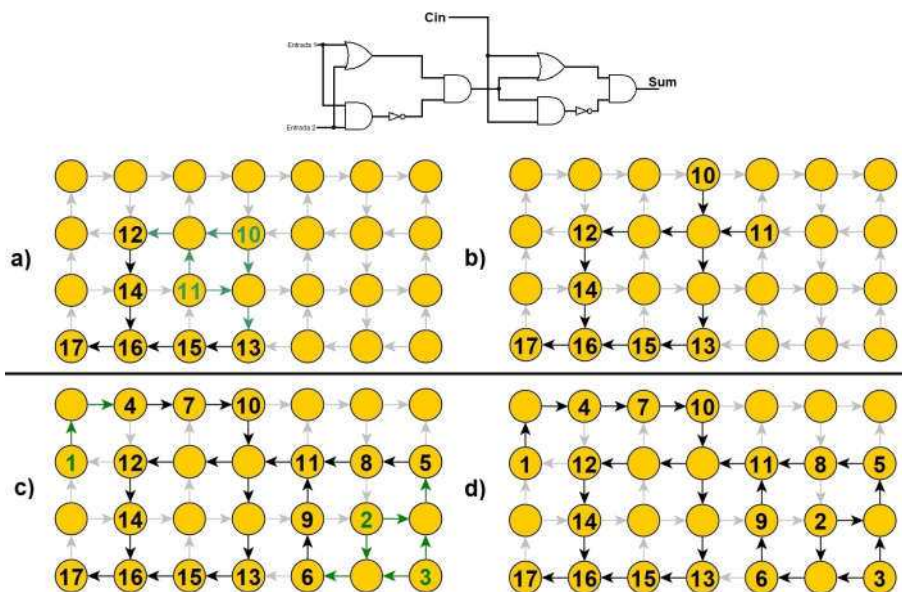


Figura 5.8 – Passo a passo da execução da heurística de P&R para o circuito *Full Adder de 1-bit*.

A próxima figura expõe o posicionamento das células QCA na construção do circuito lógico. Percebe-se a construção de portas da maioria com *layouts* distintos

dependendo do sentido de suas entradas. Além disso, nota-se a possibilidade de dois fluxos de dados compartilharem a mesma zona de clock. Isto diminui o consumo de área e consequentemente células QCA do projeto.

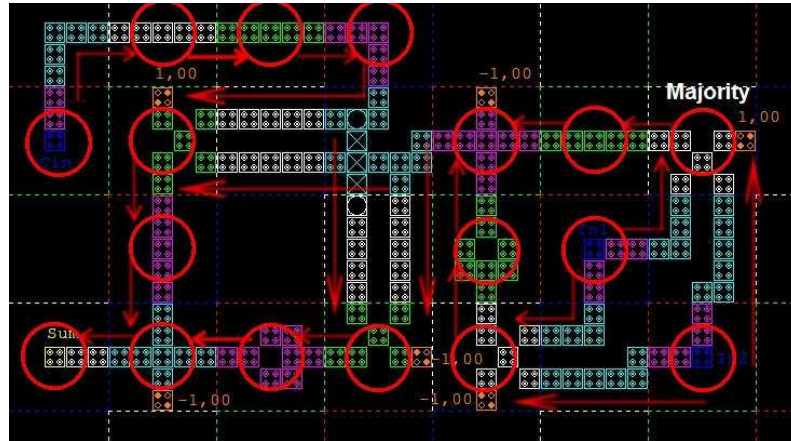


Figura 5.9 – Resultado do P&R para o *Full Adder de 1-bit* no QCADesigner.

A figura 5.10 representa o posicionamento do circuito *Full Adder de 1-bit* na ferramenta implementado em (TEODÓSIO; SOUSA, 2007).

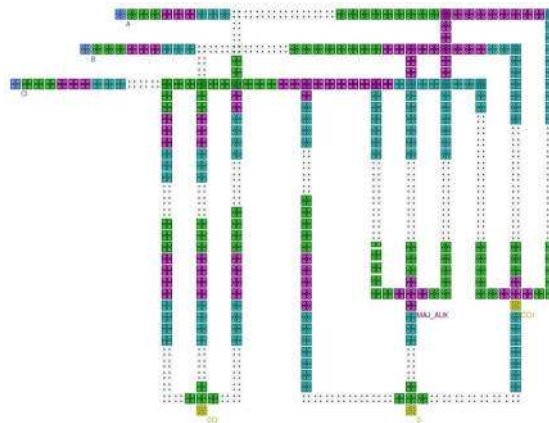


Figura 5.10 – P&R do circuito *Full Adder de 1-bit* na ferramenta QCA-LG (TEODÓSIO; SOUSA, 2007).

A figura 5.11 apresenta a forma de onda do teste do circuito. Neste caso, o atraso entre a inserção das entradas e a coleta da saída é maior, pois o circuito tem um caminho crítico com mais zonas de clock. Dito que em QCA cada zona de clock do caminho crítico de um circuito impõe um tempo de atraso constante, quanto mais zonas de clock em seu caminho crítico mais tempo levará para o circuito processar os dados.

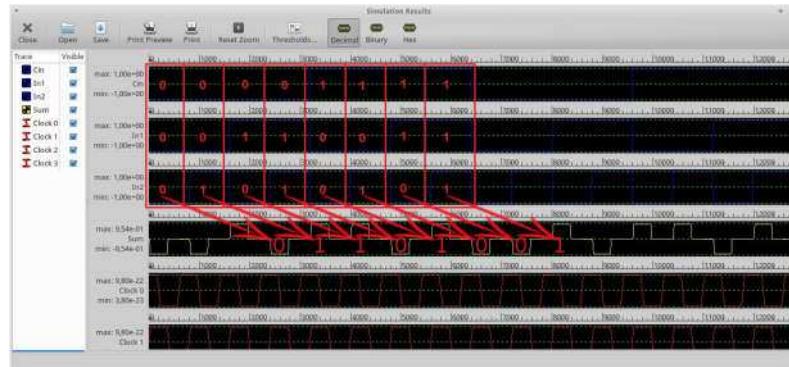


Figura 5.11 – Forma de onda do circuito *Full Adder de 1-bit* da figura 5.8.

A tabela a seguir demonstra um comparativo entre alguns dos circuitos automaticamente implementados pela heurísticas deste trabalho e circuitos de terceiros.

Foram calculados cinco atributos de seis circuitos distintos encontrados. Os circuitos são os seguintes: *MUX 2x1*, *XOR*, *Full-adder 1-bit*, *XNOR*, *Parity Generator*, *Parity Checker*. Os atributos comparados foram os seguintes: Área, caminho crítico, cruzamentos de fios, Potas da maioria, *Células QCA*.

A área foi calculada a partir da quantidade de linhas e colunas necessárias para posicionar todas as células QCA do circuito. A figura 5.12 demonstra como é feito o cálculo da área consumida pelo circuito QCA.

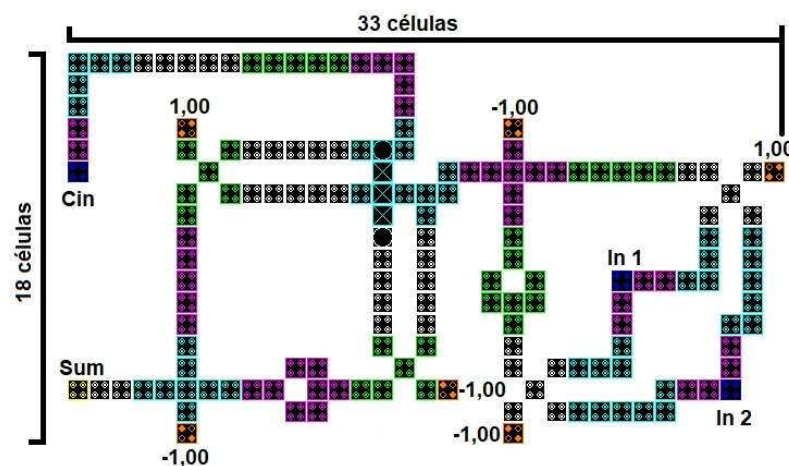


Figura 5.12 – Demonstração de cálculo de área do circuito QCA.

O caminho crítico foi computado pela pior caso na contagem do número de células QCA entre a(s) entrada(s) e a(s) saída(s). O pior caso é definido pelo cenário com maior quantidade de células. Os atributos cruzamento de fios, portas da maioria e células QCA são alcançados pela simples contagem de cada um dentro do circuito.

Cabe ressaltar que todos os circuitos de outros autores foram projetados sem adotar qualquer esquema de clock regular e de forma manual, conseqüentemente a área do circuito pode ser otimizada consideravelmente devido a implementação de

zonas de clock com formato e posição irregulares.

Os posicionamentos manuais empregam zonas de clock com quantidade mínima de células QCA e utilizam mais cruzamento de fios para ocupar menos espaço na área do circuito, diferentemente, do esquema de clock adotado aqui, que utiliza um tamanho padrão de zonas de clock de 5x5 células QCA e permite maior escalabilidade por ser regular.

Nossa proposta pode construir rotas em todas as direções de uma matriz bidimensional de células QCA em comparação aos outros trabalhos que organizam o fluxo de informação de forma unidirecional. Além disso, a maioria dos autores não validam seus resultados em nível de layout. Nossos circuitos são validados por meio da ferramenta de simulação QCADesigner.

Estruturas QCA	Layouts Anteriores					Layouts Propostos				
	Área	Caminho crítico	Cruzamentos	Portas da maioria	Células QCA	Área	Caminho crítico	Cruzamentos	Portas da maioria	Células QCA
MUX 2x1 (CAMPOS et al., 2016)	13x15	23	0	3	55	18x21	30	0	3	86
XOR (CAMPOS et al., 2016)	20x20	35	1	3	79	16x31	34	0	3	108
Full-adder 1-bit (CAMPOS et al., 2016)	40x41	75	2	7	226	51x35	98	3	7	339
XNOR										
(JAGARLAMUDI; SAHA; JAGARLAMUDI, 2011)	13x12	24	0	4	65	26x21	42	0	3	116
Parity Generator (AHMAD; BHAT, 2012)	27x08	58	2	6	99	41x46	71	1	6	270
Parity Checker (AHMAD; BHAT, 2012)	18x15	40	3	9	145	28x76	71	2	9	355

Tabela 1 – Layout Manual versus P&R

Em comparação com a única ferramenta capaz de realizar a geração do *layout* de circuitos combinacionais para a nanotecnologia QCA encontrada em (TEODÓSIO; SOUSA, 2007), nossos resultados apresentam um consumo de área menor. As figuras 4.23 e 4.24 expõe a disparidade de área consumida no P&R do circuito *MUX* entre (TEODÓSIO; SOUSA, 2007) e os resultados desta dissertação. Já as figuras 4.28 e 4.29 apresentam o P&R dos dois trabalhos para o circuito *Full-adder 1-bit*.

As figuras a seguir representam o circuito lógico, grafo em nível de circuito, P&R manual, circuito QCA e forma de onda dos seguintes circuitos: *XNOR*, *Parity Generator* e *Parity Checker*.

### 5.2.1 XNOR

As figuras a seguir apresentam os resultados do P&R para o circuito *XNOR* seguindo o algoritmo proposto neste trabalho. A figura 5.13 (b) traz o grafo do circuito da figura 5.13 (a).

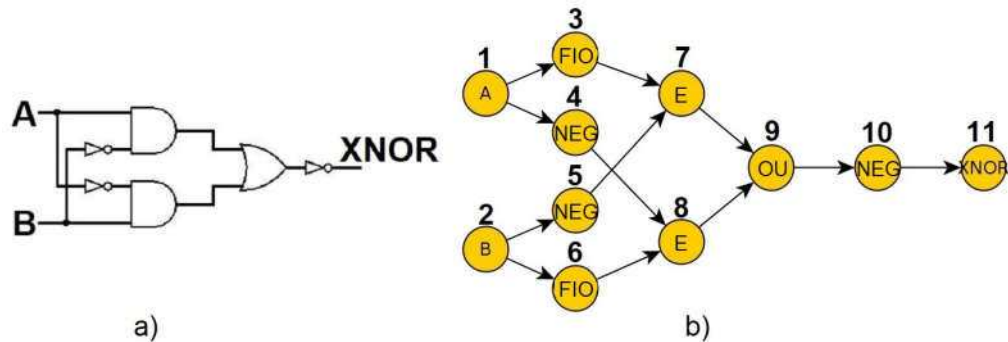


Figura 5.13 – (a) Circuito *XNOR* em nível de porta lógica. (b) Grafo do circuito *XNOR* gerado para entrada da heurística.

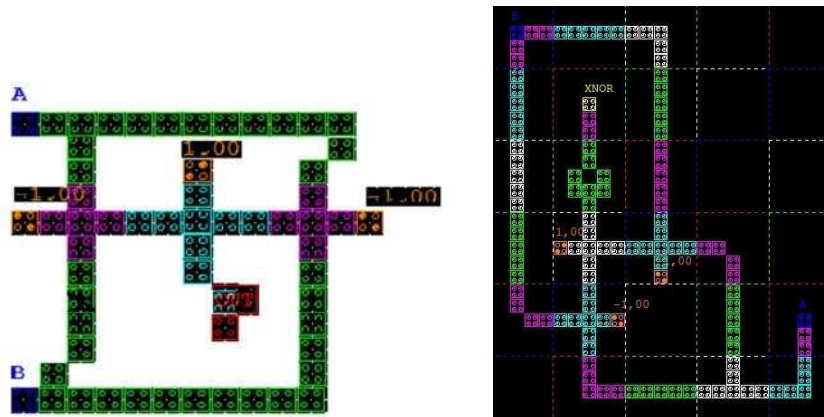


Figura 5.14 – P&R manual do circuito *XNOR* (JAGARLAMUDI; SAHA; JAGARLAMUDI, 2011). Resultado do P&R para o *XNOR* no QCADesigner.

A figura 5.14 expõe o P&R manual do circuito *XNOR* e o P&R automático encontrado pelo algoritmo deste trabalho, respectivamente.

A figura 5.15 demonstra o resultado da forma de onda para o P&R automático do circuito *XNOR*.

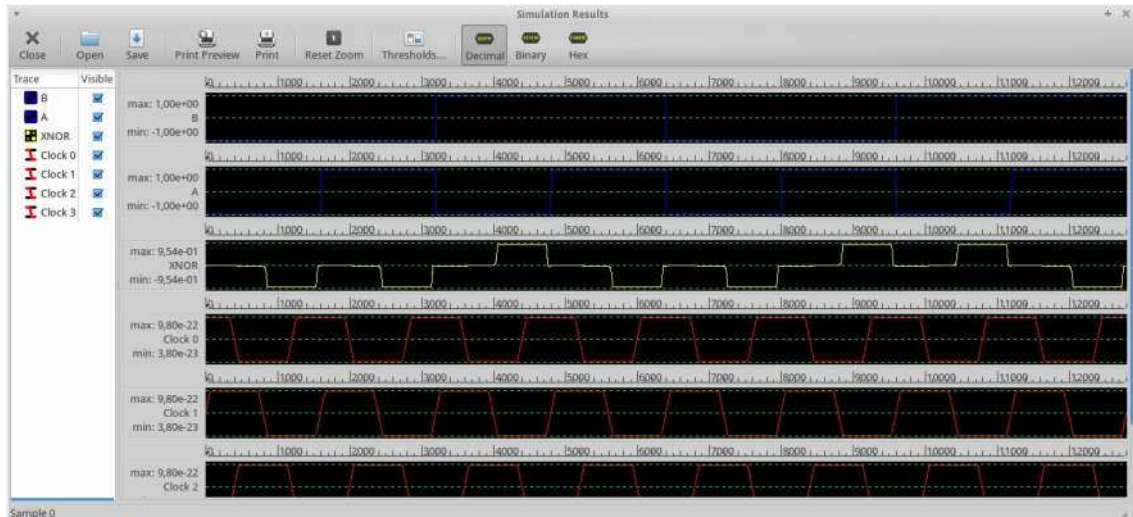


Figura 5.15 – Forma de onda do circuito QCA do *XNOR*.

### 5.2.2 Parity Generator

As figuras a seguir apresentam os resultados do P&R para o circuito *Parity Generator* seguindo o algoritmo proposto neste trabalho. A figura 5.16 (b) traz o grafo do circuito da figura 5.16 (a).

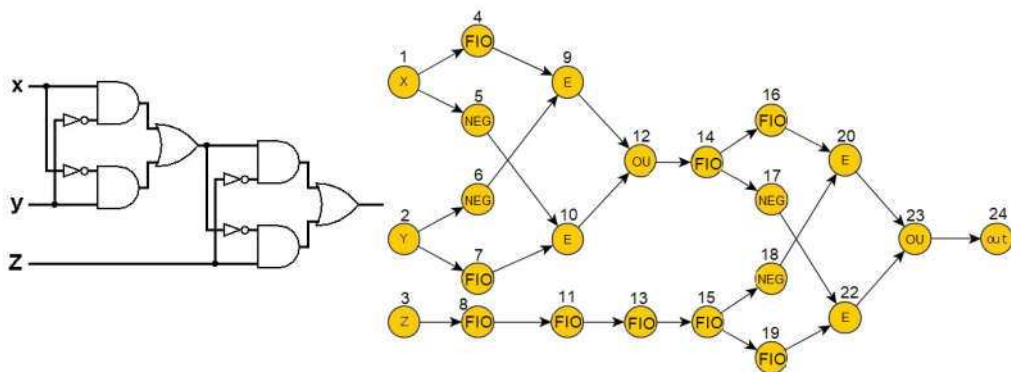


Figura 5.16 – (a) Circuito *Parity Generator* em nível de porta lógica. (b) Grafo do circuito *Parity Generator* gerado para entrada da heurística.

A figura 5.17 expõe o P&R manual do circuito *Parity Generator* e o P&R automático encontrado pelo algoritmo deste trabalho, respectivamente.

A figura 5.18 demonstra o resultado da forma de onda para o P&R automático do circuito *Parity Generator*.

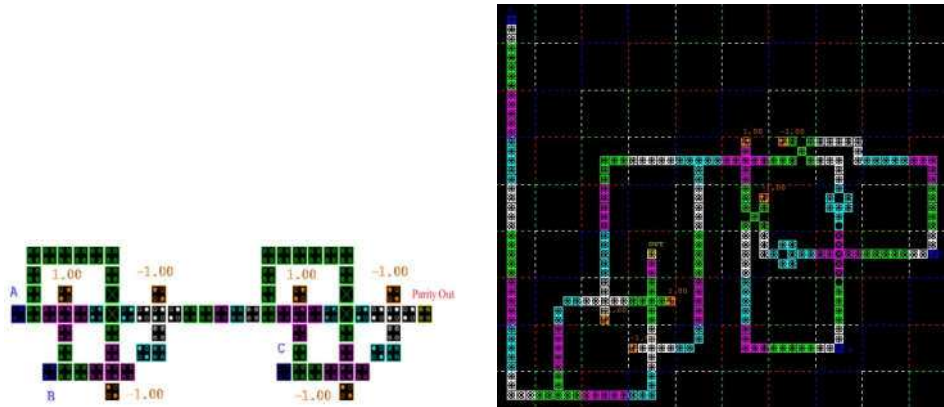


Figura 5.17 – P&R manual do circuito *Parity Generator* (AHMAD; BHAT, 2012). Resultado do P&R para o *Parity Generator* no QCADesigner.



Figura 5.18 – Forma de onda do circuito QCA do *Parity Generator*.

### 5.2.3 *Parity Checker*

As figuras a seguir apresentam os resultados do P&R para o circuito *Parity Checker* seguindo o algoritmo proposto neste trabalho. A figura 5.16 (b) traz o grafo do circuito da figura 5.16 (a).

A figura 5.20 expõe o P&R manual do circuito *Parity Checker* e o P&R automático encontrado pelo algoritmo deste trabalho, respectivamente.

A figura 5.21 demonstra o resultado da forma de onda para o P&R automático do circuito *Parity Checker*.

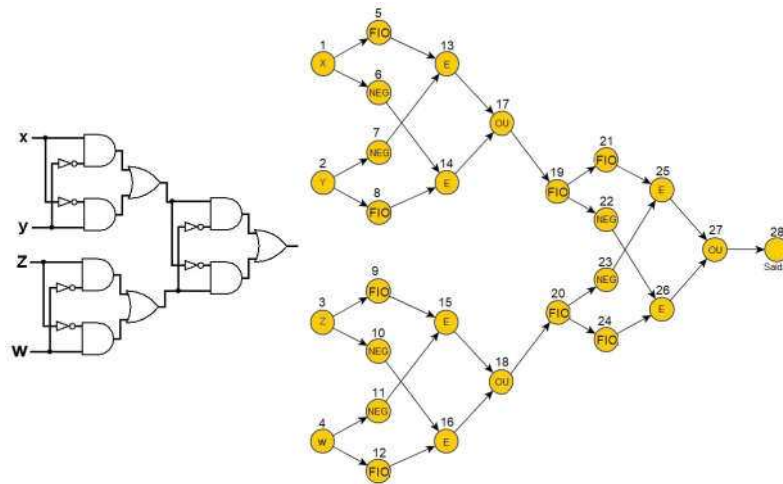


Figura 5.19 – (a) Circuito *Parity Checker* em nível de porta lógica. (b) Grafo do circuito *Parity Checker* gerado para entrada da heurística.



Figura 5.20 – P&R manual do circuito *Parity Checker* (AHMAD; BHAT, 2012). Resultado do P&R para o *Parity Checker* no QCADesigner.



Figura 5.21 – Forma de onda do circuito QCA do *Parity Checker*.

## 6 Conclusão

A nanotecnologia QCA necessita da construção de ferramentas que possam viabilizar a automatização na construção de projetos de circuitos. Atualmente, não existe nenhuma ferramenta capaz de permitir que circuitos lógicos sejam automaticamente sintetizados na tecnologia QCA. A ferramenta mais utilizada em testes de circuitos QCA é a QCADesigner, que permite o projeto manual e simulação de circuitos lógicos.

Nesta direção, é importante desenvolver algoritmos e ferramentas de projeto circuitos QCA. Características como escalabilidade, regularidade, baixo consumo de recursos e alto desempenho são necessárias à estes algoritmos. A camada a nível de clock (USE) apresentada em (CAMPOS et al., 2016) dá a possibilidade de alcançar tais características, uma vez que resolve um dos grandes desafios da tecnologia QCA, que é a sincronização.

A escalabilidade de USE traz a possibilidade de implementação de várias heurísticas de partição de grafos para implementar o posicionamento e roteamento de circuitos. Outro fator importante em USE é a presença de fluxo de informação bidirecional em comparação com outras abordagens unidirecionais (VANKAMAMIDI; OTTAVI; LOMBARDI, 2008).

A sincronização de circuitos QCA introduz novas restrições no posicionamento e roteamento de circuitos e questões como balanceamento de comprimento de fios e minimização de cruzamento de fios têm prioridade elevada.

Dando continuidade ao esquema de clock USE (CAMPOS et al., 2016) este trabalho apresenta a primeira heurística gulosa de posicionamento e roteamento que gera automaticamente o layout de circuitos combinacionais para a nanotecnologia QCA. O circuito é mapeado em um grafo acíclico direto e posicionado em uma matriz de células QCA. Os circuitos gerados pela heurística apresentam área similar aos outros trabalhos da literatura (TEODÓSIO; SOUSA, 2007), (LIM; RAVICHANDRAN; NIEMIER, 2005), porém é a primeira a usar um esquema regular e escalável de clock.

Trabalhos futuros devem ir em direção das seguintes linhas: inserir a heurística apresentada em uma ferramenta de simulação de circuitos como QCADesigner, adaptação da heurística proposta para realizar o P&R de circuitos sequencias, P&R de grandes circuitos por meio de particionamento, melhorar a exploração do espaço de solução realizando o posicionamento dos níveis intermediários.

## Referências

- AHMAD, F.; BHAT, G. Novel code converters based on quantum-dot cellular automata (qca). *International Journal of Science and Research*, v. 3, n. 5, p. 364–371, 2012. Citado 5 vezes nas páginas [ix](#), [43](#), [50](#), [53](#) e [54](#).
- AMIRI, M. A.; MAHDAVI, M.; MIRZAKUCHAKI, S. Qca implementation of a mux-based fpga clb. In: IEEE. *Nanoscience and Nanotechnology, 2008. ICONN 2008. International Conference on*. [S.l.], 2008. p. 141–144. Citado na página [3](#).
- ANTONELLI, D. A. et al. Quantum-dot cellular automata (qca) circuit partitioning: problem modeling and solutions. In: ACM. *Proceedings of the 41st annual Design Automation Conference*. [S.l.], 2004. p. 363–368. Citado 2 vezes nas páginas [vii](#) e [23](#).
- BHANJA, S. et al. Novel designs for thermally robust coplanar crossing in qca. In: EUROPEAN DESIGN AND AUTOMATION ASSOCIATION. *Proceedings of the conference on Design, automation and test in Europe: Proceedings*. [S.l.], 2006. p. 786–791. Citado na página [11](#).
- BHANJA, S. et al. Qca circuits for robust coplanar crossing. In: *Emerging Nanotechnologies*. [S.l.]: Springer, 2008. p. 227–249. Citado na página [11](#).
- CAMPOS, C. A. T. et al. Use: A universal, scalable, and efficient clocking scheme for qca. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, IEEE, v. 35, n. 3, p. 513–517, 2016. Citado 11 vezes nas páginas [vi](#), [4](#), [13](#), [15](#), [16](#), [17](#), [19](#), [28](#), [43](#), [50](#) e [55](#).
- CAVIN, R. K.; LUGLI, P.; ZHIRNOV, V. V. Science and engineering beyond moore’s law. *Proceedings of the IEEE*, IEEE, v. 100, n. Special Centennial Issue, p. 1720–1749, 2012. Citado 3 vezes nas páginas [vi](#), [1](#) e [2](#).
- COLUMBIA, T. U. of B. *QCADesigner / Microsystems and Nanotechnology Group (MiNa)*. 2012. Disponível em: <http://www.mina.ubc.ca/qcadesigner>. Citado na página [20](#).
- DINDIGUL, S. Efficient design of logical structures and functions using nanotechnology based quantum dot cellular automata design. *vectors*, v. 10, p. 11. Citado na página [11](#).
- FARAZKISH, R. et al. Design and characterization of a novel inverter for nanoelectronic circuits. In: *International Conference on Nanotechnology: Fundamentals and Applications*. [S.l.: s.n.], 2010. v. 219. Citado na página [7](#).
- GRAUNKE, C. R. et al. Implementation of a crossbar network using quantum-dot cellular automata. *IEEE Transactions on Nanotechnology*, IEEE, v. 4, n. 4, p. 435–440, 2005. Citado na página [12](#).
- HENDERSON, S. C. et al. Incorporating standard cmos design process methodologies into the qca logic design process. *IEEE Transactions on nanotechnology*, IEEE, v. 3, n. 1, p. 2–9, 2004. Citado 3 vezes nas páginas [18](#), [19](#) e [20](#).

- HIMSOLT, M. Gml: A portable graph file format. *Html page under <http://www.fmi.uni-passau.de/graphlet/gml/gml-tr.html>*, Universität Passau, 1997. Citado na página [34](#).
- HUANG, J. et al. Design and characterization of an and-or-inverter (aoi) gate for qca implementation. In: ACM. *Proceedings of the 14th ACM Great Lakes symposium on VLSI*. [S.l.], 2004. p. 426–429. Citado na página [7](#).
- JAGARLAMUDI, H. S.; SAHA, M.; JAGARLAMUDI, P. K. Quantum dot cellular automata based effective design of combinational and sequential logical structures. *World Academy of Science, Engineering and Technology*, v. 60, p. 671–675, 2011. Citado 4 vezes nas páginas [ix](#), [43](#), [50](#) e [51](#).
- JANEZ, M.; PECAR, P.; MRAZ, M. Layout design of manufacturable quantum-dot cellular automata. *Microelectronics Journal*, Elsevier, v. 43, n. 7, p. 501–513, 2012. Citado na página [11](#).
- KEYES, R. W. Physical limits of silicon transistors and circuits. *Reports on Progress in Physics*, IOP Publishing, v. 68, n. 12, p. 2701, 2005. Citado na página [1](#).
- LANTZ, T.; PESKIN, E. A qca implementation of a configurable logic block for an fpga. In: IEEE. *Reconfigurable Computing and FPGA 's, 2006. ReConFig 2006. IEEE International Conference on*. [S.l.], 2006. p. 1–10. Citado 2 vezes nas páginas [3](#) e [7](#).
- LENT, C. S.; TOUGAW, P. D. A device architecture for computing with quantum dots. *Proceedings of the IEEE*, IEEE, v. 85, n. 4, p. 541–557, 1997. Citado na página [12](#).
- LENT, C. S. et al. Quantum cellular automata. *Nanotechnology*, IOP Publishing, v. 4, n. 1, p. 49, 1993. Citado 5 vezes nas páginas [vi](#), [5](#), [6](#), [7](#) e [8](#).
- LIM, S. K.; RAVICHANDRAN, R.; NIEMIER, M. Partitioning and placement for buildable qca circuits. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, ACM, v. 1, n. 1, p. 50–72, 2005. Citado 8 vezes nas páginas [vii](#), [21](#), [22](#), [23](#), [24](#), [27](#), [43](#) e [55](#).
- MARKOV, I. L. Limits on fundamental limits to computation. *Nature*, Nature Research, v. 512, n. 7513, p. 147–154, 2014. Citado na página [1](#).
- MOORE, G. E. et al. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, v. 86, n. 1, p. 82–85, 1998. Citado na página [1](#).
- NIEMIER, M.; KOGGE, P. Origins and motivations for design rules in qca. *Nano, quantum and molecular computing*, Springer, p. 267–293, 2004. Citado 2 vezes nas páginas [vi](#) e [12](#).
- NIEMIER, M. T. *Designing digital systems in quantum cellular automata*. Tese (Doutorado) — University of Notre Dame, 2000. Citado na página [15](#).
- ORLOV, A. et al. Realization of a functional cell for quantum-dot cellular automata. *Science*, American Association for the Advancement of Science, v. 277, n. 5328, p. 928–930, 1997. Citado na página [6](#).

- OTTAVI, M. et al. Hdlq: a hdl environment for qca design. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, ACM, v. 2, n. 4, p. 243–261, 2006. Citado 2 vezes nas páginas [vi](#) e [19](#).
- PATITZ, Z. D. *Fault Tolerant Quantum-dot Cellular Automata Majority Gate Design*. Tese (Doutorado) — Oklahoma State University, 2006. Citado 2 vezes nas páginas [7](#) e [11](#).
- PRESHING, J. *A Look Back at Single-Threaded CPU Performance*. 2012. Disponível em: <http://preshing.com/20120208/a-look-back-at-single-threaded-cpu-performance/>. Citado 2 vezes nas páginas [vi](#) e [3](#).
- SENTOVICH, E. M. et al. Sis: A system for sequential circuit synthesis. Citeseer, 1992. Citado na página [25](#).
- SHIN, S.-H.; JEON, J.-C.; YOO, K.-Y. Wire-crossing technique on quantum-dot cellular automata. In: *NGCIT2013, the 2nd International Conference on Next Generation Computer and Information Technology*. [S.l.: s.n.], 2013. v. 27, p. 52–57. Citado 2 vezes nas páginas [vi](#) e [11](#).
- SUTTER, H. The free lunch is over: A fundamental turn toward concurrency in software. *Dr. Dobbs's journal*, v. 30, n. 3, p. 202–210, 2005. Citado na página [2](#).
- SYNTHESIS, B. L. *ABC: a system for sequential synthesis and verification, release 70930*. 2007. Citado na página [32](#).
- TEODÓSIO, T.; SOUSA, L. Qca-1g: A tool for the automatic layout generation of qca combinational circuits. In: IEEE. *Norchip, 2007*. [S.l.], 2007. p. 1–5. Citado 12 vezes nas páginas [vii](#), [viii](#), [25](#), [26](#), [27](#), [28](#), [43](#), [44](#), [46](#), [48](#), [50](#) e [55](#).
- TOUGAW, P. D.; LENT, C. S. Logical devices implemented using quantum cellular automata. *Journal of Applied physics*, AIP, v. 75, n. 3, p. 1818–1825, 1994. Citado 6 vezes nas páginas [vi](#), [6](#), [7](#), [8](#), [10](#) e [11](#).
- VANKAMAMIDI, V.; OTTAVI, M.; LOMBARDI, F. Two-dimensional schemes for clocking/timing of qca circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, IEEE, v. 27, n. 1, p. 34–44, 2008. Citado 2 vezes nas páginas [15](#) e [55](#).
- WALUS, K. et al. Qcadesigner: A rapid design and simulation tool for quantum-dot cellular automata. *IEEE transactions on nanotechnology*, IEEE, v. 3, n. 1, p. 26–31, 2004. Citado na página [19](#).
- WALUS, K.; JULLIEN, G.; DIMITROV, V. Computer arithmetic structures for quantum cellular automata. In: IEEE. *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*. [S.l.], 2003. v. 2, p. 1435–1439. Citado 4 vezes nas páginas [vi](#), [12](#), [13](#) e [14](#).