

## 4

## Simulador de QCA

### 4.1.

#### Introdução

O uso de simuladores já se mostrou imprescindível no desenvolvimento e pesquisa de circuitos tradicionais. A importância do seu uso cresce ainda mais quando a implementação física dos circuitos ainda é difícil de ser obtida, como no caso dos circuitos de QCA.

Com o objetivo de desenvolver um sistema para a síntese automática de circuitos de QCA é necessário um simulador eficiente e com uma fácil interface com o algoritmo genético (AG). Os poucos simuladores de circuitos de QCA atualmente existentes não estão disponíveis para o público ou não possuem uma interface que permita a troca de informações com o AG. Portanto, nesta dissertação é proposto um novo simulador de circuitos de QCA, baseado em redes neurais artificiais (RNA), que é rápido e inclui as características desejadas para a execução da função de avaliação de um algoritmo genético.

Além da descrição do simulador desenvolvido neste trabalho, este capítulo fará uma breve apresentação dos simuladores de QCA existentes e uma descrição das considerações feitas para o desenvolvimento do novo simulador.

### 4.2.

#### Simuladores de QCA Já Existentes

O primeiro simulador de QCA foi desenvolvido no Departamento de Engenharia Elétrica da Universidade de Notre Dame nos Estados Unidos e recebeu o nome de AQUINAS [71]. Este simulador foi desenvolvido com o objetivo de simular células de QCA e pequenos circuitos utilizando modelos de mecânica quântica. O AQUINAS certamente fornece resultados corretos quando se deseja verificar a lógica fornecida por um certo arranjo de células. Porém, o simulador se mostrou muito lento, sendo o tempo computacional necessário para a

simulação extremamente elevado, até mesmo para circuitos pequenos. O fato deste simulador não estar disponível para o público é outra desvantagem.

Mais recentemente, o AQUINAS foi recriado e recebeu o nome de MAQUINAS. O objetivo desta versão é focar nos sistemas de QCA molecular. Como na versão anterior, a simulação de grandes sistemas de QCA é bastante lenta.

Com o intuito de desenvolver uma ferramenta mais rápida e que pudesse verificar a lógica produzida por um arranjo de células, pesquisadores do University College of London criaram uma ferramenta simples utilizando o Microsoft Excel [72]. Este simulador também obteve relativo sucesso, mas as restrições do Excel e as dificuldades de criar e alterar a topologia dificultam seu uso. Assim como o AQUINAS, este simulador não é de uso público.

Devido a essa dificuldade de verificar a lógica de circuitos de QCA de tamanho médio ou grande, de forma rápida e precisa, um novo simulador foi proposto pelos pesquisadores da universidade de Notre Dame [73]. O simulador foi chamado de Q-BERT (*Quantum Logic Based Engine Rules Tool*) e permite ao usuário a criação e simulação de circuitos de médio e grande porte, tais como um somador completo e até mesmo um microprocessador simples. Porém, a interface deste simulador é bastante complicada e difícil de ser usada, além de só estar disponível para os pesquisadores da universidade de Notre Dame.

O único simulador de QCA disponível para o público é o QCADesigner [74][75]. Este simulador foi desenvolvido por pesquisadores da universidade de Calgary no Canadá, juntamente com pesquisadores da universidade de Notre Dame. O QCADesigner possui muitas das funções esperadas para a construção das topologias dos circuitos. Além disso, este simulador é capaz de diversos tipos de simulações, desde as baseadas na lógica digital até aquelas envolvendo mecânica quântica. O único empecilho para o uso do QCADesigner na síntese automática de circuitos de QCA, utilizando AG, é o fato do simulador só permitir o desenvolvimento de circuitos através da interface gráfica, obrigando o usuário a posicionar célula por célula do circuito. Esta característica dificulta a comunicação entre o simulador e o AG. Os desenvolvedores do simulador prometem fornecer futuramente outros tipos de interfaces, permitindo esta comunicação com outras ferramentas.

Devido a estas diferenças entre os vários simuladores disponíveis, Dysart propôs recentemente uma representação universal para os circuitos de QCA, a ser usada por todos os simuladores [76]. Esta representação é baseada em XML e possibilitaria que uma topologia de um circuito de QCA desenvolvida em um simulador pudesse ser simulada em um outro simulador de QCA. Além disso, possibilitaria, também, que uma ferramenta de síntese automática de circuitos de QCA pudesse ser usada em conjunto com qualquer simulador.

<b>Simulador</b>	<b>Local</b>	<b>Vantagens</b>	<b>Desvantagens</b>
AQUINAS [71]	Universidade de Notre Dame	Simulação Quântica	Muito Lento
MAQUINAS	Universidade de Notre Dame	Simulação Quântica de QCA molecular	Muito Lento
QCA em Excel [72]	University College of London	Aplicação de Microsoft Excel	Difícil interface e simulação incorreta
Q-BERT	Universidade de Notre Dame	Simulação lógica rápida	Interface difícil e simulação incorreta
QCADesigner	University of Calgary	Vários tipos de simulações e velocidade. Interface amigável	Ainda em desenvolvimento. Difícil interface para o AG

Tabela 2– Simuladores de QCA já desenvolvidos.

#### 4.3. Rede Neural do Tipo Hopfield

Conforme já foi informado anteriormente, uma rede neural do tipo Hopfield foi utilizada para o desenvolvimento do simulador de QCA. Nesta seção será feita uma breve introdução sobre esta rede.

As redes neurais do tipo Hopfield [77][78][79], cuja topologia é ilustrada na figura 16, são altamente interconectadas e são utilizadas com sucesso em problemas de otimização e minimização de energia. A estrutura da rede e o peso sináptico entre os neurônios dependem da especificação de cada problema. Cada neurônio simplesmente soma os sinais enviados pelos outros neurônios, multiplicado pelo peso sináptico, e então gera um sinal de saída. A soma dos

sinais de entrada,  $U$ , e o sinal de saída,  $V(U)$ , são chamados de potencial de entrada e potencial de saída, respectivamente.

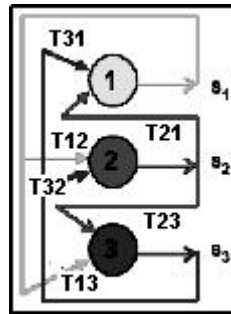


Figura 16– Estrutura de uma rede neural do tipo Hopfield com conexões entre todos os neurônios.

A função de energia da rede pode ser definida como:

$$E = -1/2 \sum_{i=1}^m \sum_{j=1}^m T_{ij} V_i V_j \dots\dots\dots(3)$$

onde  $T_{ij}$  ( $i, j = 1, \dots, m$ ) é o peso sináptico entre os neurônios  $i$  e  $j$ . O conjunto de valores  $T_{ij}$  ( $=T_{ji}$ ) forma uma matriz simétrica contendo os pesos sinápticos. Cada  $T_{ij}$  pode ser positivo ou negativo, e é zero quando não há conexão entre os neurônios  $i$  e  $j$ . O potencial de entrada  $U_i$  de um neurônio  $i$  é definido por

$$U_i = -\partial E / \partial V_i \quad (i = 1, \dots, m) \quad (4)$$

das equações acima, chega-se a

$$U_i = \sum_{j=1}^m T_{ij} V_j \quad (i = 1, \dots, m) \quad (5)$$

As equações acima determinam a evolução da rede para um mínimo local ou global de energia, que é um estado meta-estável ou estável, respectivamente, da rede. Estas equações, juntamente com uma heurística que controla a interação entre os neurônios, são suficientes para a implementação do simulador.

#### 4.4.

#### Considerações Sobre os Dispositivos Simulados

Com o objetivo de desenvolver o simulador de QCA que realize a verificação da lógica digital de circuitos, utilizando uma rede neural do tipo Hopfield, algumas considerações devem ser feitas. Primeiramente, cada neurônio representa uma célula do circuito de QCA e deve ter valores entre -1 e 1. Portanto,

uma rede neural que representa um circuito com 20 células deve possuir 20 neurônios, conforme será exemplificado mais adiante.

No simulador, as células são posicionadas em um *grade*, assim elas não podem assumir qualquer posição no espaço, mas somente aquelas posições já pré-determinadas pela *grade*. A figura 17 mostra um exemplo de *grade* para a criação de um circuito de QCA. O centro da célula deverá ser posicionada em qualquer um dos pontos internos da *grade*, ocupando quatro quadrados. A parte cinza na *grade* representa uma única célula e seu centro está exatamente em um dos pontos da *grade*, como é indicado pela seta.

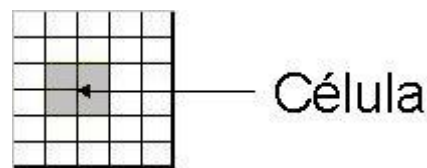


Figura 17– Exemplo de *grade* para posicionamento de células.

O Simulador desenvolvido considera que as células e os circuitos simulados estejam em um ambiente perfeito, ou seja, não há interferência externa, tal como temperatura ou qualquer outro fator que possa alterar o comportamento das interações entre as células de QCA.

A seguir, outras considerações mais detalhadas serão feitas.

#### 4.4.1. Raio de Vizinhança de uma Célula

Na versão do simulador de QCA implementada neste trabalho, a região de vizinhança de uma célula foi definida como um círculo de raio de duas células, conforme ilustrado na figura 18. Logo, uma célula só poderá interagir com outra célula cujo centro estiver dentro deste raio. Esta vizinhança é suficiente para o correto funcionamento do simulador, permitindo o cruzamento da informação em um plano, conforme explicado anteriormente na seção 3.4.

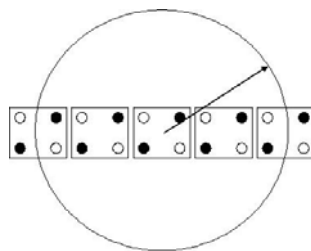


Figura 18– O raio de vizinhança de uma célula de QCA do simulador.

#### 4.4.2. Interação entre as Células

Nesta seção serão apresentadas as configurações das células utilizadas no simulador, definindo a distância entre os pontos quânticos, o valor das cargas negativas e positivas e o cálculo das interações entre as células.

Todas as células deste trabalho possuem um tamanho único e, como assume-se a simulação de um circuito ideal, nenhuma unidade de medida específica é assumida. Logo, os valores que indicam as distâncias entre os pontos quânticos, apresentados nas figuras abaixo, representam uma unidade arbitrária e não uma determinada unidade de medida. Estas distâncias, juntamente com os valores das cargas positivas e negativas, são utilizadas para os cálculos das interações coulombianas entre as células.

Segundo a teoria sobre as células de QCA, apresentada anteriormente no item 3.1, cada célula possui dois elétrons livres e as cargas positivas ( $0,5\text{ C}$  cada), que compensam a carga negativa dos elétrons, estão fixadas nos quatro pontos quânticos da célula. Portanto, quando um elétron está posicionado em um ponto quântico, este ponto possui carga igual a  $-0,5\text{ C}$ . De maneira oposta, o ponto possui carga de  $+0,5\text{ C}$  quando não há elétron. Porém, com o intuito de simplificar o cálculo, já que o interesse é desenvolver um simulador que verifique a lógica de um circuito de QCA ideal, assume-se que a carga no ponto quântico é igual a  $-1\text{ C}$  na presença de um elétron e  $+1\text{ C}$  caso contrário.

Para a implementação do simulador é necessário calcular a “*kink energy*” entre duas células para todas as possíveis posições que respeitem o raio de vizinhança apresentado. Os cálculos para cada uma destas configurações serão apresentados a seguir, acompanhados das figuras com as indicações das distâncias entre os pontos quânticos.

Inicialmente, a figura 19 mostra duas células posicionada lado a lado. A figura à esquerda mostra as células com polarizações opostas, enquanto a imagem da direita mostra as células com a mesma polarização.



Figura 19– Duas células lado a lado com a polarizações diferentes e polarizações semelhantes.

O cálculo da interação entre células com a polarização diferente é apresentado a seguir, onde  $k$  é uma constante que depende do material utilizado.

$$E_1 = k \left[ \frac{-1}{2} + \frac{+1}{3} + \frac{+1}{2,23} + \frac{-1}{3,16} + 1 + \frac{-1}{2} + \frac{-1}{1,41} + \frac{+1}{2,23} + \frac{-1}{2} + \frac{+1}{3} + \frac{+1}{2,23} + \frac{-1}{3,16} + 1 + \frac{-1}{2} + \frac{-1}{1,41} + \frac{+1}{2,23} \right]$$

$$\text{Logo, } E_1 = 0,41k .$$

Já o cálculo da interação entre as células com polarizações iguais é apresentado abaixo:

$$E_2 = k \left[ \frac{+1}{2} + \frac{-1}{3} + \frac{-1}{2,23} + \frac{+1}{3,16} - 1 + \frac{+1}{2} + \frac{+1}{1,41} + \frac{-1}{2,23} + \frac{+1}{2} + \frac{-1}{3} + \frac{-1}{2,23} + \frac{+1}{3,16} - 1 + \frac{+1}{2} + \frac{+1}{1,41} + \frac{-1}{2,23} \right]$$

$$\text{Assim, } E_2 = -0,41k \text{ e a "kink energy" é igual a: } E_k = E_1 - E_2 = 0,82k .$$

É fácil perceber que a interação entre as células com polarizações iguais e polarizações diferentes é igual em módulo, mas com sinais diferentes.

Para as demais posições, que respeitem o raio de vizinhança determinado, o valor da "kink energy" será mostrado na tabela 3 e o cálculo detalhado pode ser visto no apêndice 1.

Figura	Kink Energy	Comentários
	$E_k = 0,82k$	Interação entre células convencionais lado a lado.
	$E_k = -1,25k$	Interação entre células rotacionadas lado a lado.
	$E_k = 0k$	Interação entre uma célula convencional e uma rotacionada lado a lado.

	$E_k = 0,1k$	Interação entre células convencionais distanciadas de meia célula.
	$E_k = -0,12k$	Interação entre células rotacionadas distanciadas de meia célula.
	$E_k = 0k$	Interação entre uma célula convencional e uma rotacionada distanciadas de meia célula.
	$E_k = 0,025k$	Interação entre células convencionais distanciadas de uma célula.
	$E_k = -0,026k$	Interação entre células rotacionadas distanciadas de uma célula.
	$E_k = 0k$	Interação entre uma célula convencional e uma rotacionada distanciadas de uma célula.
	$E_k = -0,18k$	Interação entre células convencionais em diagonal.
	$E_k = 0,14k$	Interação entre células rotacionadas em diagonal.



	$E_k = 0k$	<p>Interação entre uma célula convencional e uma rotacionada em diagonal.</p>
	$E_k = -0,04k$	<p>Interação entre células convencionais em diagonal e distanciadas.</p>
	$E_k = 0,024k$	<p>Interação entre células rotacionadas em diagonal e distanciadas.</p>
	$E_k = 0,026k$	<p>Interação entre uma célula convencional e uma rotacionada em diagonal e distanciadas. O valor da “<i>kink energy</i>” pode inverter o sinal, dependendo da posição. Observe explicação a seguir e apêndice 1.</p>
	$E_k = -0,18k$	<p>Interação entre células convencionais em meia diagonal.</p>
	$E_k = 0,046k$	<p>Interação entre células rotacionadas em meia diagonal.</p>

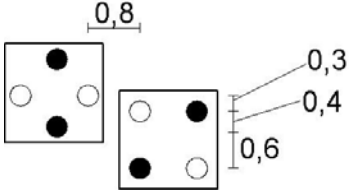
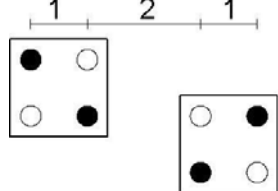
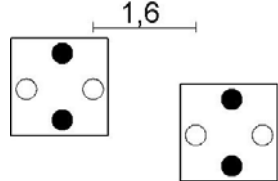
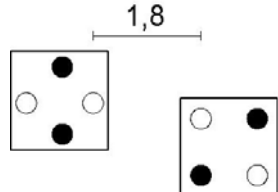
	$E_k = 0,68k$	<p>Interação entre uma célula convencional e uma rotacionada em meia diagonal. O valor da “<i>kink energy</i>” pode inverter o sinal, dependendo da posição. Observe explicação a seguir e apêndice 1.</p>
	$E_k = 0,02k$	<p>Interação entre células convencionais em meia diagonal e distanciadas.</p>
	$E_k = -0,046k$	<p>Interação entre células rotacionadas em meia diagonal e distanciadas.</p>
	$E_k = 0,086k$	<p>Interação entre uma célula convencional e uma rotacionada em meia diagonal e distanciadas. O valor da “<i>kink energy</i>” pode inverter o sinal, dependendo da posição. Observe explicação a seguir e apêndice 1.</p>

Tabela 3– “*Kink energy*” para todas as possíveis posições que respeitem o raio de vizinhança definido.

Quando existem duas células vizinhas, sendo uma convencional e outra rotacionada, o cálculo da energia deve ser tratado com cuidado. Conforme descrito nas observações da tabela 1, quando o valor da energia entre estas células não for nulo, o valor da energia é positivo ou negativo, dependendo da posição da célula com relação à célula vizinha. A figura 20 abaixo mostra duas possíveis configurações diferentes de células vizinhas. Note, pela marcação utilizada na figura, que os dois pontos quânticos mais próximos possuem uma interação de

repulsão na figura mais à esquerda e uma interação de atração na figura mais à direita, apesar das células possuírem a mesma polarização nos dois casos. Isto deve ser tratado com cuidado pelo simulador, a fim de evitar que o circuito simulado apresente interações incorretas.

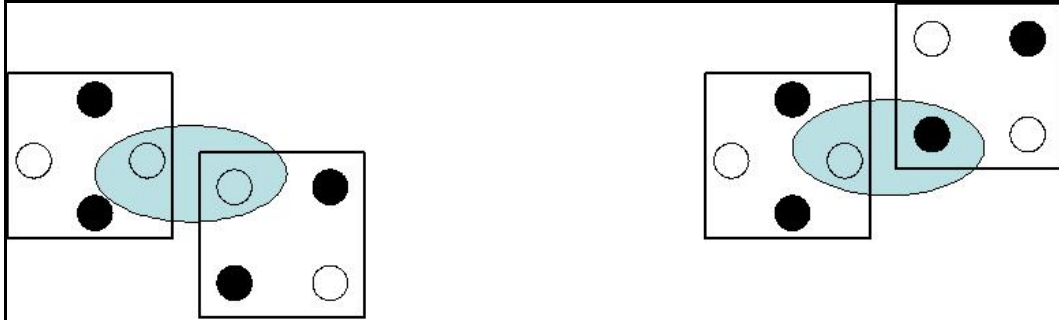


Figura 20– Como a posição da célula inverte o sinal da “*kink energy*” quando uma célula convencional é vizinha de uma célula rotacionada.

#### 4.5. A Simulação

O simulador desenvolvido neste trabalho recebe como entrada uma matriz que define a topologia do circuito, isto é, esta matriz define onde existe uma célula e qual o tipo desta célula (convencional ou rotacionada). Além disso, o simulador também recebe como entrada uma outra matriz que informa a qual zona de *clock* cada célula pertence. Por último, é necessário informar a posição das células de entrada e suas polarizações.

A “*kink energy*” entre duas células, conforme calculado no item anterior, é utilizada como o peso sináptico  $T_{ij}$  entre dois neurônios, que representam duas células de um circuito. Portanto, como um exemplo, o peso sináptico entre o neurônio 1 e o neurônio 2 ( $T_{12} = T_{21}$ ), ilustrado na figura 21, é o valor da “*kink energy*” entre a célula 1 e a célula 2, também ilustrado na figura 21. Após todos os pesos sinápticos serem calculados, a função de energia da rede,  $E$ , pode ser determinada. A derivada desta função é o potencial de entrada do neurônio  $i$ ,  $U_i$ , como definido anteriormente no item 4.3.

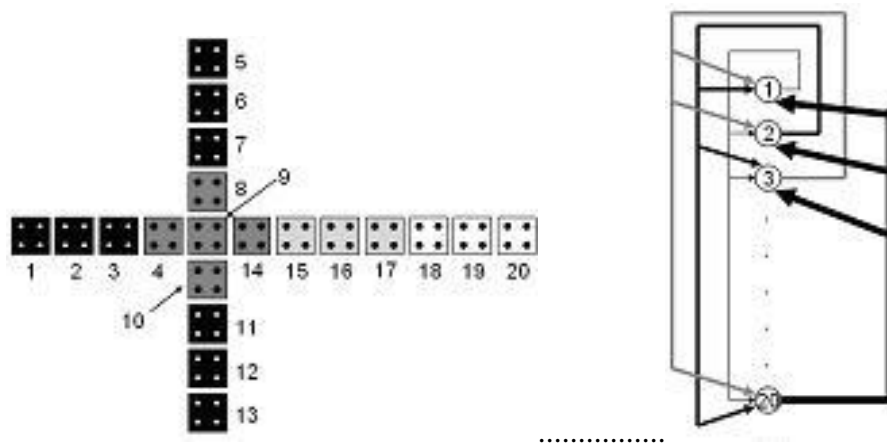


Figura 21- O circuito do Majority Gate (os tons de cinza representam as diferentes zonas de clock). E a topologia da rede neural de Hopfield que o representa.

Um aspecto muito importante para o correto funcionamento do circuito é a escolha da função que representa o potencial de saída do neurônio,  $V(U)$ . Conforme foi discutido no capítulo 3, a interação entre as células de QCA possui uma forte característica não-linear. Logo, o potencial de saída deve também seguir esta característica. A função a seguir foi escolhida devido a sua semelhança com o comportamento não linear entre as células de QCA, apresentado na figura 4.

$$V(U) = \tanh(4.5 * U) \dots\dots\dots(6)$$

Com o objetivo de simular circuitos de QCA, a rede neural é executada diversas vezes, de acordo com as quatro fases de *clock* possíveis. Somente os neurônios que representam células na fase de *switch* podem ter seus valores alterados. Os neurônios que correspondem às células que estão na fase de *clock* anterior (*hold*) têm seus valores congelados e são usados como entradas para a zona de *clock* corrente. Os neurônios que representam as células pertencentes às zonas de *clock release* ou *relax* têm seus valores iguais a zero.

Com o intuito de fornecer uma polarização inicial às células despolarizadas, uma função diferente, mostrada na eq. (7), é usada para representar o potencial de saída de um neurônio,  $V(U)$ , retornando um valor menor do que aquele retornado pela eq. (5). Isto é feito para evitar erros nos passos seguintes da simulação, conforme foi observado nos primeiros testes.

$$V(U) = \tanh(U) \dots\dots\dots(7)$$

A figura 22 mostra o algoritmo simplificado utilizado para executar a simulação.

**procedure:** Simula um circuito

**input:** matriz com a topologia do circuito

Cria a matriz de pesos sinápticos;

Inicia *clock*;

**while** (condição de parada é falsa)

    Limpa zona de *clock* deslocada de 180°;

**while** (circuito é alterado)

**for** (cada neurônio despolarizado e na zona de *clock* atual)

            Fornece uma polarização inicial às células  
            despolarizadas;

**end for**

**while** (polarização não converge ou não atinge o número  
    máximo de interações)

**for** (Para cada neurônio polarizado e na zona de  
        *clock* atual)

            Calcula o estado de polarização das células  
            polarizadas;

**end for**

**end while**

**end while**

        Passa para a próxima zona de *clock*;

**end while**

**end procedure**

Figura 22– Algoritmo simplificado utilizado na simulação.

## 4.6. Resultados

Esta seção apresentará detalhadamente algumas simulações de pequenos circuitos. No apêndice 2 mais simulações serão apresentadas com detalhes, mostrando as ilustrações dos circuitos a cada zona de *clock*.

### 4.6.1. Majority Gate

A seguir, a simulação do *majority gate*, mostrado na figura 23, será apresentada em detalhes. Nesta simulação, as três células de entrada são representadas pelas células #1, #5 e #13 e suas polarizações iniciais são -1, 1 e 1, respectivamente. A célula #20 representa a célula de saída.

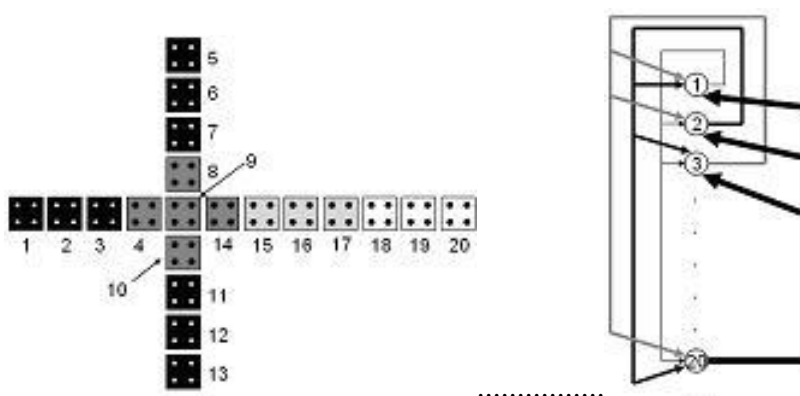


Figura 23- O circuito do Majority Gate (os tons de cinza representam as diferentes zonas de *clock*). E a topologia da rede neural de Hopfield que o representa.

A tabela 4 apresenta a polarização de cada célula, passo a passo. Note que a coluna mais à esquerda indica a zona de *clock* e os valores das células pertencente a zona de *clock* estão em negrito. A primeira linha mostra o índice da célula.

Como era esperado, o valor do neurônio #20 no fim da execução da zona de *clock* 4 é bem próximo de 1. Isto significa que o valor de saída da célula #20, neste caso, é o binário 1. Logo, a simulação está correta, já que na entrada há dois neurônios com o binário 1 e um com o binário 0.

CK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	-1	-0,67	-0,025	0	1	0,67	0,025	0	0	0	0,25	0,67	1	0	0	0	0	0	0	0
1	-1	-0,99	-0,99	0	1	0,99	0,99	0	0	0	0,99	0,99	1	0	0	0	0	0	0	0
1	-1	-1	-0,99	0	1	1	0,99	0	0	0	0,99	1	1	0	0	0	0	0	0	0
1	-1	-1	-0,99	0	1	1	0,99	0	0	0	0,99	1	1	0	0	0	0	0	0	0
1	-1	-1	-0,99	0	1	1	0,99	0	0	0	0,99	1	1	0	0	0	0	0	0	0
1	-1	-1	-0,99	0	1	1	0,99	0	0	0	0,99	1	1	0	0	0	0	0	0	0
2	-1	-1	-1	-0,68	1	1	1	0,68	0,025	0,68	1	1	1	0	0	0	0	0	0	0
2	-1	-1	-1	-1	1	1	1	0,99	0,99	0,99	1	1	1	0	0	0	0	0	0	0
2	-1	-1	-1	-0,93	1	1	1	1	0,99	1	1	1	1	0	0	0	0	0	0	0
2	-1	-1	-1	-0,93	1	1	1	1	0,99	1	1	1	1	0	0	0	0	0	0	0
2	-1	-1	-1	-0,93	1	1	1	1	0,99	1	1	1	1	0,41	0	0	0	0	0	0
2	-1	-1	-1	-0,93	1	1	1	1	1	1	1	1	1	0,96	0	0	0	0	0	0
2	-1	-1	-1	-0,92	1	1	1	1	1	1	1	1	1	0,96	0	0	0	0	0	0
2	-1	-1	-1	-0,92	1	1	1	1	1	1	1	1	1	0,96	0	0	0	0	0	0
2	-1	-1	-1	-0,92	1	1	1	1	1	1	1	1	1	0,96	0	0	0	0	0	0
3	0	0	0	-1	0	0	0	1	1	1	0	0	0	1	0,68	0,025	0	0	0	0
3	0	0	0	-1	0	0	0	1	1	1	0	0	0	1	0,99	0,99	0	0	0	0
3	0	0	0	-1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0
3	0	0	0	-1	0	0	0	1	1	1	0	0	0	1	1	1	0,68	0	0	0
3	0	0	0	-1	0	0	0	1	1	1	0	0	0	1	1	1	0,99	0	0	0
3	0	0	0	-1	0	0	0	1	1	1	0	0	0	1	1	1	0,99	0	0	0
3	0	0	0	-1	0	0	0	1	1	1	0	0	0	1	1	1	0,99	0	0	0
3	0	0	0	-1	0	0	0	1	1	1	0	0	0	1	1	1	0,99	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0,68	0,025	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0,99	0,99	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0,99	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0,99	0,68
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0,99
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0,99
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0,99
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0,99

Tabela 4– O estado de polarização de cada célula, a cada passo da rede neural.

Note que, para cada zona de *clock*, os neurônios previamente polarizados pelas zonas de *clock* anterior têm seus valores congelados e iguais a 1 ou -1, representando a fase de *clock hold*, e sendo usados como entradas para a próxima zona de *clock*.

A figura 24, a seguir, mostra o gráfico de saída do simulador para a simulação descrita acima. As cores das células indicam as polarizações e as zonas de *clock* são destacadas, facilitando a visualização. Deve-se lembrar que cada quatro quadrados do *grade* representam uma célula.

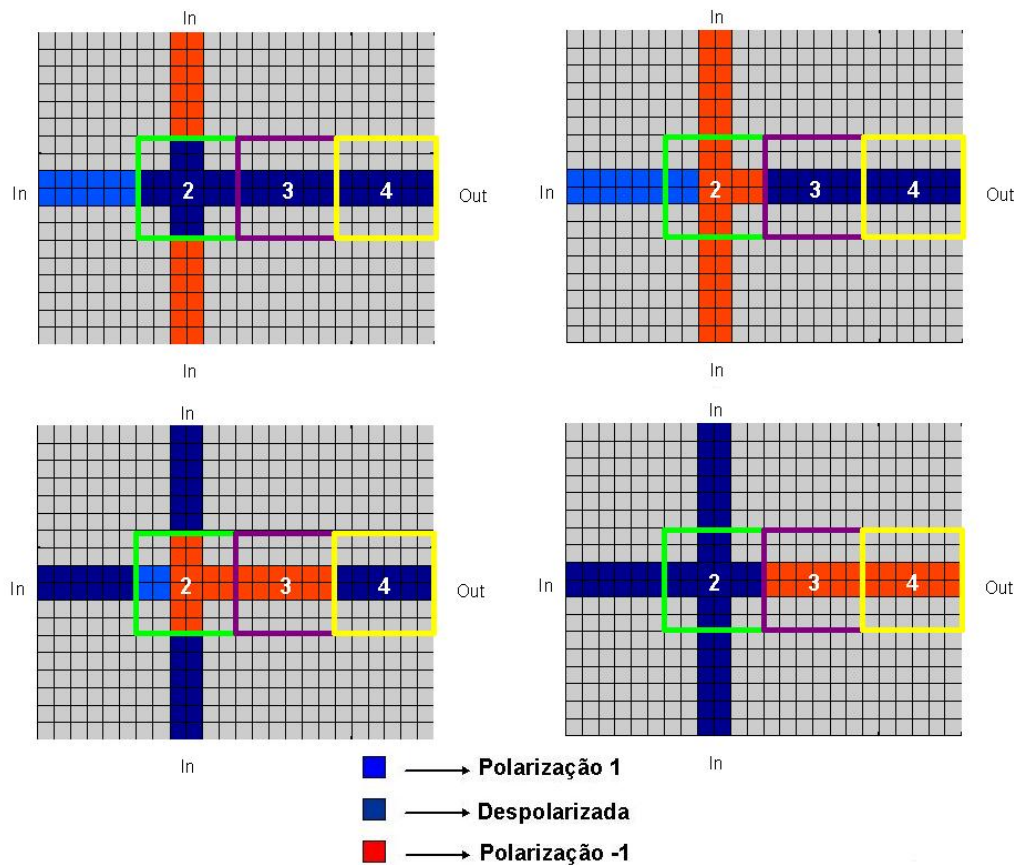


Figura 24– Resultado da simulação do majority gate para um conjunto de entradas.

#### 4.6.2. Circuito com ruído

A seguir, outro pequeno exemplo é utilizado para demonstrar o funcionamento do simulador. Apesar de simples, o exemplo utiliza o conceito de ruído de uma célula dentro do circuito, o que é um bom teste para a interação entre os neurônios da rede neural que representam as células do circuito.

O circuito utilizado é uma simplificação daquele utilizado por Kim [54]. A simplificação se deve ao fato do simulador utilizado neste trabalho utilizar um raio de vizinhança menor do que aquele utilizado por Kim em seu trabalho. Apesar da simplificação o resultado encontrado foi o mesmo e é um ótimo exemplo para testar o simulador.

A não linearidade e a saturação bi-estável da interação célula-célula funciona tal como um ganho em circuitos digitais convencionais [80]. Um pequena polarização de uma célula induz uma grande polarização em seus



vizinhos, conforme já foi mostrado na figura 4. O vizinho também induz uma polarização de volta para a célula até mesmo antes da polarização desta célula vizinha saturar. Esta sinergia amplifica não somente a polarização de um sinal, mas também o ruído propagado através de um fraco sinal ruidoso.

Considere o arranjo de células mostrado na figura 25. Duas entradas são aplicadas em 1 e 8. A entrada 1 atua como sinal enquanto a entrada 8 atua como ruído. Apesar da “*kink energy*” entre as células 7 e 10 ser aproximadamente 30 vezes mais forte do que a “*kink energy*” entre as células 9 e 10, o ruído da entrada 8 chega na célula 10 antes do sinal, devido ao caminho mais curto, e então se propaga para as células da linha vertical. A polarização que atua entre estas células amplifica o ruído e este fica estagnado na célula 10.

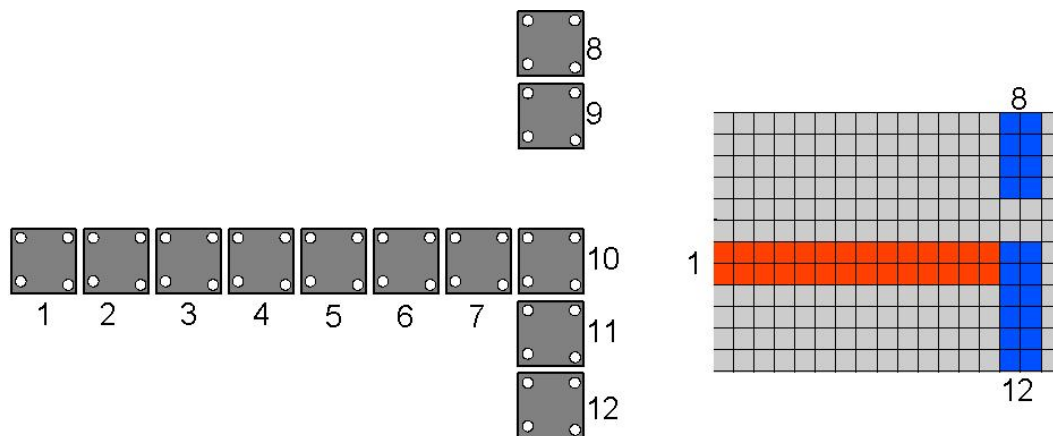


Figura 25– Arranjo de células com ruído. À direita o resultado da simulação, onde a cor vermelho representa a polarização -1 e a cor azul a polarização 1.

Porém, se a célula 12 for removida do final da linha vertical, o ruído induzido pela polarização não é amplificado o suficiente, e o ruído desaparece, como ilustrado na figura 26. Este experimento demonstra o bom funcionamento do simulador e mostra que a amplificação do ruído só ocorre quando duas condições são encontradas: o ruído chega antes do sinal e a linha onde é inserido o ruído é longa o suficiente.

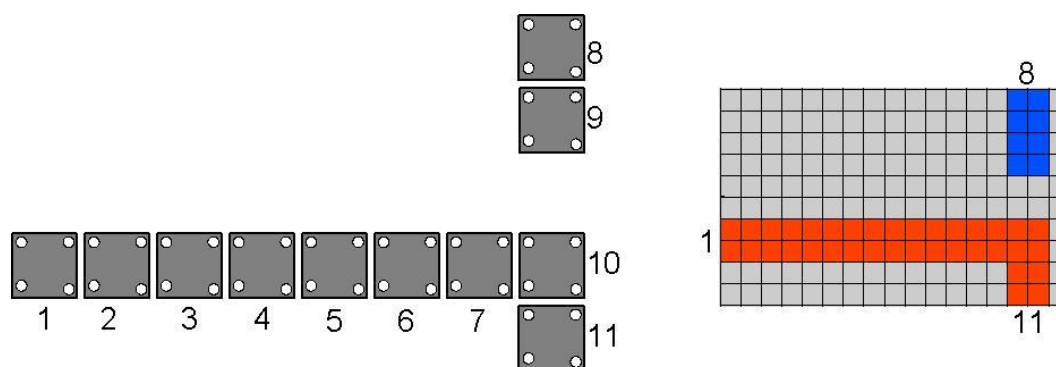


Figura 26-- Arranjo de células sem ruído. À direita o resultado da simulação, onde a cor vermelho representa a polarização -1 e a cor azul a polarização 1.

Além destes circuitos apresentados, muitos outros foram simulados. Alguns desses podem ser encontrados no apêndice 2, juntamente com todas as figuras da simulação.

O próximo capítulo apresenta o modelo evolucionário utilizado para a síntese automática de circuitos de QCA.