Ricardo Filipe Raimundo Belchior

Licenciado em Ciências da Engenharia Electrotécnica
e de Computadores

Computação Trans-Booleana

Dissertação para obtenção do Grau de Mestre em

Engenharia Eletrotécnica e de Computadores

Orientador: Raul Eduardo Capela Tello Rato, Professor Doutor,
Universidade Nova de Lisboa

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2018

Computação Trans-Booleana

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "novathesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

"Why can't we see
That when we bleed we bleed the same"
– Map of problematique, Muse

Agradecimentos

Quero primeiramente agradecer ao Professor Raul Rato pela oportunidade e confiança prestada ao longo deste ano incrível.

Aos meus pais por toda a dedicação e empenho na minha formação e educação. Sem alicerces de tão elevada qualidade nada disto seria possível. Ao meu irmão por todo o apoio e camaradagem. É, e sempre será, um exemplo claro a seguir.

A todos os meus amigos de faculdade por todo este percurso de 5 anos incrível.

A todos os meus amigos de sempre que, mesmo ausente estão sempre dispostos a recordar o que de melhor se passou nas nossas vidas. São vocês a base de quem sou hoje.

À Ermelinda Mineiro, ao João Pateira e à Sara Baptista por todos os ensinamentos. Se há energia para acreditar e fazer melhor, ela está em vocês. Vou sempre recordar a minha passagem pelo CDUL como um marco na minha vida.

A todos o meu obrigado.

Resumo

A computação e a capacidade de processamento são assuntos cada vez mais na ordem do dia. Hoje em dia qualquer dispositivo incorpora um pequeno processador capaz de realizar milhões de operações por segundo. No entanto, a exigência do processamento tem crescido a um ritmo exponencial. Cada vez mais existe a necessidade de simular sistemas que, para o estado de arte actual, pode demorar anos. A capacidade de processamento tenta acompanhar este crescimento através de arquitecturas altamente paralelas, como as GPUs, capazes de acelerar o processamento e diminuir o tempo até obter um resultado. Uma outra solução, igualmente elegante, será optar por um novo tipo de computação.

A computação quântica surge como uma forma promissora para aumentar a capacidade de processamento face aos computadores clássicos. Nos últimos anos esta área recebe vultuosos investimentos de gigantes tecnológicas, com objectivo de construir o primeiro computador quântico funcional. No entanto, este revela ter alguns problemas de implementação e escalabilidade, uma vez que a sua implementação é baseada nos fenómenos fisicos quânticos.

Ao tentar simplificar a abordagem da computação quântica, surge a computação trans-booleana. Nesta, parte-se do formalismo e dos fundamentos algébricos da mecânica quântica, mas sem os implementar através de um sistema físico quântico. Tenta-se assim, com base em sinais analógicos e o seu processamento, obter os mesmos resultados que a quântica promete. Mostra-se que é possível implementar portas lógicas como a NOT, CNOT e Hadamard, sem considerar ruído na codificação e nas operações. No entanto alguns problemas surgem com o entrelaçamento e interferência.

Palavras-chave: Teoria da Computação, Computação Quântica, Sinais e Sistemas, Computação Trans-Booleana

Abstract

Computational power has become more important over the years. Nowadays, any electronic device, like a small processor, is capable of doing millions of tasks per second. However, the complexity of those tasks is increasing at an exponential rate. This raises a problem of how to simulate such systems, if our current technology, said classical, can not do it in polynomial time. Numerous alternatives, like GPU and others, have been developed to overcome this problems. Besides classical alternatives, a new computation paradigm urges from Quantum Mechanics.

Quantum Computing arrives as an efficient problem solver. In the recent years, research aimed to find a reliable quantum computer has received a lot of attention and investment by major players. The biggest problems in this systems are related to scalability and decoherence.

In order to avoid such problems, a new paradigm arises with the proposed Trans-Boolean Computation. Based on the mathematical formalism of quantum computing, this new type of computation tries to implement such formalism using signals and analog systems, pursuing the same properties as a quantum computer. Here is shown how to implement logic gates as NOT, Hadamard and CNOT with ideal systems. However some problems arises when entanglement and interference are needed.

Keywords: Computation Theory, Quantum Computing, Signals and Systems, Trans-Boolean Computation

Índice

Lista de Figuras

Lista de Tabelas

Símbolos

$\mathbb{F}$	Corpo de escalares genérico.
$\langle\vert$	Vector linha *bra*.
$\otimes$	Produto de Kronecker.
Γ	Valor limite para a detecção.
γ	Ângulo longitudinal da esfera de Bloch.
θ	Ângulo de latitude da esfera de Bloch.
α	Número complexo associado ao valor lógico F.
β	Número complexo associado ao valor lógico V.
θ_α	Ângulo do número complexo associado ao valor lógico F.
φ_β	Ângulo do número complexo associado ao valor lógico V.
$\vert\rangle$	Vector coluna *ket*.
H	Porta Hadamard.
T_s	Tempo de amostragem.
T_{sim}	Tempo de simulação.
U	Porta Genérica.
f	Frequência.
r	Raio da esfera de Bloch.
t	Tempo.

Siglas

CNOT Controlled-NOT.

QFT Quantum Fourier Transform.

Capítulo 1

Introdução

What does the inside of a computer look like?
Crudely, it will be build out of a set of simple basic elements.
Lectures on Computation - Richard Feynman

1.1 Motivação e Objectivos

No início do século XX estava em preparação uma revolução no mundo da ciência e da física. Uma crise instaurou-se quando as teorias (agora clássicas) mais promissoras da altura não conseguiam descrever determinados fenómenos da Natureza. Estas levavam a soluções absurdas que envolviam energia infinita e electrões com trajéctorias impossiveis. Por volta dos anos vinte (1920s) surgiu uma teoria moderna, mas pouco intuitiva, chamada de Mecânica Quântica. Desde aí, esta tem sido uma das ferramentas mais úteis e indispensáveis da ciência, oferecendo explicações para os mais variados fenómenos da Natureza [1].

A Mecânica Quântica pode ser vista como um conjunto de regras matemáticas para a construção de teorias físicas. A electrodinâmica quântica é, por exemplo, uma teoria física que usa a mecânica quântica como base fundamental. [1, 2].

Nos anos que se seguiram, o interesse, mas também o cepticismo, aumentou quantos a novas ideias sobre a mecânica quântica. No inicio dos anos 80, Wooters, Zurek e Dieks propõem o *No-Cloning Theorem* [3]. Essencialmente, este teorema enuncia que não é possível clonar um qualquer sistema quântico. Este resultado teve implicações importantes no desenvolver da mecânica quântica, por exemplo na impossibilidade de existir comunicação mais rápida que a velocidade da luz usando fenómenos quânticos [1, 3].

A implementação e controlo dos sistemas quânticos revela, igualmente, ser um aspecto fundamental. O principal problema reflecte-se no tempo de coerência, ou seja no intervalo

de tempo em que o sistema se encontra num estado estacionário. Sistemas como supercondutores são estudados desde os anos 70 e têm mostrado tempo de coerência e fiabilidade consideráveis [4].

À medida que os algoritmos quânticos foram surgindo, como o de Shor e Groover, muitos cientistas abraçaram a ideia que Richard Feynman apresentou em 1982 [5]. Este sugeriu a implementação de computadores baseados em princípios quânticos, tentando assim resolver o problema da dificuldade de simulação dos sistemas quânticos em computadores clássicos. Neste sentido várias áreas da física apostaram na tentativa de desenvolver fisicamente um computador quântico. Destas destacam-se a Supercondutividade [6] ou Ressonância Magnética Nuclear [7].

Foi assim nascendo um novo paradigma na computação, com base no formalismo da mecânica quântica, que se denomina Computação Quântica. O formalismo que esta acarreta não é mais do que uma generalização da lógica clássica booleana. Às variáveis clássicas 0 e 1, que neste texto se representam por $|0\rangle$ e $|1\rangle$, associam-se quaisquer números complexos criando assim um novo modelo de computação. Desta forma, tal como a figura 1.1 sugere, a computação quântica engloba a clássica uma vez que a generaliza.

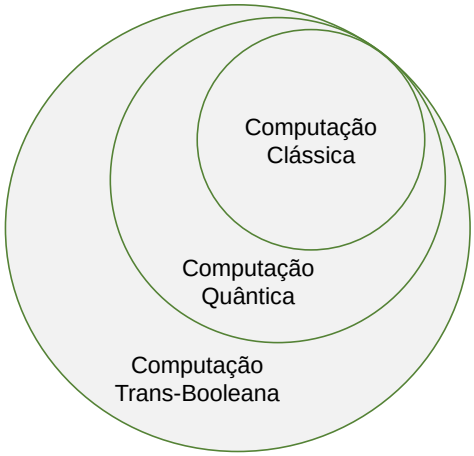


Figura 1.1: Vários modelos de computação. A computação trans-booleana é o modelo mais geral de todos os representados. Segue-se a computação quântica e a computação clássica.

A computação quântica não necessita de um sistema físico quântico que a implemente [8]. Este resultado leva a um outro modelo, denominado de Computação Trans-Boolena. O nome trans-booleano surge do facto de este modelo implementar, da mesma forma que a quântica o faz, um formalismo matemático que consegue realizar operações que na álgebra booleana não são possíveis. Um exemplo é a raiz quadrada de NOT [9]. Este modelo distingue-se do quântico uma vez que a implementação é conforme ao formalsmo, mas independente da microfísica dos efeitos quânticos. Sem estes, prevê-se que o modelo seja mais genérico do que os anteriores, tal como a figura 1.1 sugere [10, 11].

Estes fundamentos algébricos podem ser implementados segundo qualquer um destes três modelos, como a figura 1.2 sugere. Segundo sistemas digitais, vão surgir problemas no tempo e espaço com complexidade exponencial. Estes provêm da exigência do cálculo matricial, que cresce exponencialmente com o aumento polinomial do número de qubits. Optar por um sistema quântico leva, necessariamente, à computação quântica, como já

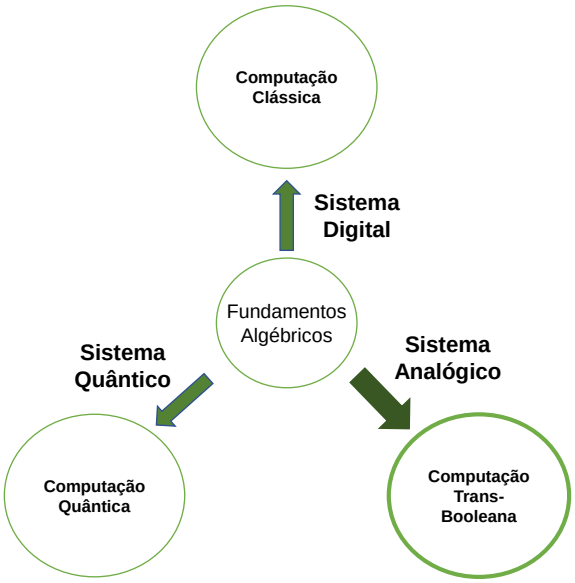


Figura 1.2: Vários tipos de computação em função da sua implementação. Pode-se implementar os mesmos fundamentos algébricos através de computação clássica, quântica ou trans-booleana.

descrito. Uma nova alternativa surge com o uso de sistemas analógicos para implementar o formalismo matemático, levando à computação trans-booleana. Estas três formas de concretização são completamente distintas, convergindo apenas no formalismo que implementam.

Esta dissertação, tal como a figura 1.2 sugere, tem como objectivo estudar uma possível implementação da Computação Trans-Booleana. Esta baseia-se em sistemas analógicos, mais propriamente em sinais sinusoidais e operações sobre os mesmos. Pretende-se mostrar que é possível implementar qubits, portas lógicas e detectores de forma a criar um sistema que emule a computação quântica.

1.2 Contribuição para a investigação

Durante a realização desta dissertação foram criadas novas formas de pensamento em relação a determinados pontos da computação quântica. Estes focaram-se, especialmente, na esfera de Bloch e na forma de construir algoritmos quânticos. No que toca em hardware, foram desenvolvidos protótipos de gates trans-booleanas, que vão ser apresentadas nos próximos capítulos. Em especial, foi implementada a gate de Hadamard com recurso a eletrónica simples. Espera-se que este trabalho tenha um impacto positivo na investigação em novos tipos de computação, contribuindo para a aceleração da potenciação da capacidade de computação.

1.3 Estrutura do Documento

Esta dissertação encontra-se dividida em cinco capítulos. O primeiro apresenta a motivação e objectivos desta dissertação. É também sumarizada a estrutura desta. O segundo introduz os fundamentos algébricos fundamentais para a compreensão da computação quântica. Faz-se também uma pequena introdução à teoria da complexidade. No terceiro, descreve-se o funcionamento base da computação quântica, sem ter em conta o sistema que a implementa. Introduz-se o qubit e os sistemas multi qubit. Por fim apresenta-se o funcionamento das portas lógicas quânticas. No quarto capítulo é proposta uma forma de implementar o sistema quântico descrito no capitulo terceiro. A esta implementação chama-se de computação trans-booleana. Descreve-se, primeiramente, a codificação física do qubit. Apresentam-se, de seguida, esquemáticos de alto nível para a implementação das portas Hadamard e CNOT. Para o sistema de detecção é introduzida uma abordagem com base em níveis de *threshold*. O quinto e último apresenta as conclusões e trabalhos propostos para o futuro.

Capítulo 2

Fundamentos Gerais

O objectivo deste capítulo é compilar algumas definições, conceitos e notações a usar ao longo da dissertação. Com o intuito de explicar as bases da computação quântica é do interesse introduzir, em 2.1, 2.2 e 2.3, breves conceitos de álgebra linear e espaços vectoriais. Estes vão ser introduzidos de forma sucinta mas lata, por forma a abranger o âmago conceptual da área. Por fim, em 2.4, é feita uma pequena introdução à teoria da computação, nomeadamente à teoria da complexidade.

2.1 Espaços Vectoriais Complexos

2.1.1 Definições e Propriedades

Relembram-se, por conveniência, algumas definições bem conhecidas. Seja $\mathbf{W}$ um conjunto não vazio, cujos elementos se convencionam chamar **vectores**. Seja $\mathbb{F}$ um corpo cujos elementos são apelidados de **escalares**. Em termos gerais tem-se que $a, b \in \mathbb{F}$ e $\mathbf{u}, \mathbf{w} \in \mathbf{W}$.

Definição 1 (Espaço Vectorial)**.** Diz-se que o conjunto $\mathbf{W}$ é um espaço vectorial sobre um corpo $\mathbb{F}$ se e só se (sse), estiverem definidas as operações de adição vectorial em $\mathbf{W}$ e multiplicação escalar entre $\mathbb{F}$ e $\mathbf{W}$, tal que:

- A adição vectorial é associativa e comutativa.
- A operação de multiplicação escalar, à esquerda, associa a qualquer par de elementos $(a, \mathbf{u})$, um elemento $a\mathbf{u} \in \mathbf{W}$, tal que:
  - $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$
  - $(a + b)\mathbf{u} = a\mathbf{u} + b\mathbf{u}$
  - $a(b\mathbf{u}) = (ab)\mathbf{u}$

■

Se o corpo $\mathbb{F} \equiv \mathbb{R}$ chama-se ao espaço vectorial W **espaço vectorial real**. Se $\mathbb{F} \equiv \mathbb{C}$ chama-se ao espaço W **espaço vectorial complexo**. Nesta dissertação, e daqui para em diante, vai-se considerar que o corpo $\mathbb{F} \equiv \mathbb{C}$, representando-se por $\mathbb{C}$.

Seja $(\mathbf{w}_1, \ldots, \mathbf{w}_n)$ uma lista de vectores do conjunto $\mathbf{W}$ e seja $(w_1, \ldots, w_n)$ uma lista de escalares do corpo $\mathbb{C}$. A **combinação linear** entre estes vectores e escalares é dada por:

$$w_1\mathbf{w}_1 + \cdots + w_n\mathbf{w}_n = \sum_{i=1}^{n} w_i\mathbf{w}_i \tag{2.1}$$

Definição 2 (Independência Linear)**.** Diz-se que uma qualquer lista de vectores, por exemplo $(\mathbf{w}_1, \ldots, \mathbf{w}_n) \in \mathbf{W}$, é linearmente independente sse, a única combinação de escalares $(w_1, \ldots, w_n) \in \mathbb{C}$ que satisfaz a equação 2.2 é $w_1 = \cdots = w_n = 0$.

$$w_1\mathbf{w}_1 + \cdots + w_n\mathbf{w}_n = 0 \tag{2.2}$$

■

Dadas as definições de combinação e independência linear, definições 1 e 2, é possível introduzir o conceito de base vectorial.

Definição 3 (Base Vectorial)**.** Seja W um espaço vectorial sobre o corpo $\mathbb{C}$. Uma **base vectorial**, $\mathscr{B}$, de W é uma lista de vectores linearmente independentes em $\mathbf{W}$, por exemplo $(\mathbf{w}_{e_1}, \ldots, \mathbf{w}_{e_n}) \in \mathbf{W}$, de tal forma que qualquer vector $\mathbf{w} \in \mathbf{W}$ pode ser escrito da forma:

$$\mathbf{w} = \sum_{i=1}^{n} w_i\mathbf{w}_{e_i} \quad w_i \in \mathbb{C} \tag{2.3}$$

■

Em termos notacionais, qualquer vector $\mathbf{w}$ pode ser representado em forma matricial, através de uma matriz coluna, como na equação 2.4, que comporta a lista $(w_1, \ldots, w_n)$.

$$\mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \tag{2.4}$$

Posto isto, $\mathbf{w}^T$ e $\mathbf{w}^\dagger$ representam, respectivamente, o vector transposto e o vector transconjugado, também referido como hermitiano.

$$\mathbf{w}^T = \begin{bmatrix} w_1 & \ldots & w_n \end{bmatrix} \tag{2.5}$$

$$\mathbf{w}^\dagger = \begin{bmatrix} \overline{w_1} & \ldots & \overline{w_n} \end{bmatrix} \tag{2.6}$$

A norma L_P de um qualquer vector, $||\mathbf{u}||$, é dada pela raiz P da soma da potência P dos módulos de cada elemento, tal como na equação 2.7.

$$||\mathbf{w}|| = \sqrt[P]{\sum_n |w_i|^P} \tag{2.7}$$

Diz-se que um vector está normalizado à unidade quando a norma $||\mathbf{w}|| = 1$. Nesta dissertação estuda-se o caso particular de $P = 2$, que corresponde à norma usual do Espaço Euclidiano onde é valido o teorema de Pitágoras.

Definição 4 (Produto Interno)**.** Seja W um espaço vectorial sobre um corpo $\mathbb{C}$. O **produto interno** em W é uma função que transforma pares ordenados de vectores $(\mathbf{u}, \mathbf{w}) \in W$, num número escalar $\langle \mathbf{u}, |\mathbf{w}\rangle \in \mathbb{C}$, tal que:

- $\langle \mathbf{u} + \mathbf{w}|\mathbf{w}\rangle = \langle \mathbf{u}|\mathbf{w}\rangle + \langle \mathbf{v}|\mathbf{w}\rangle$
- $\langle a\mathbf{u}|\mathbf{w}\rangle = a\langle \mathbf{v}|\mathbf{w}\rangle, a \in \mathbb{F}$
- $\langle \mathbf{u}|\mathbf{w}\rangle = \overline{\langle \mathbf{u}|\mathbf{w}\rangle}$
- $\langle \mathbf{u}|\mathbf{u}\rangle > 0$ se $\mathbf{u} \neq 0$

■

Com o produto interno definido, pode introduzir-se a noção de ângulo e, concomitantemente, o conceito de ortogonalidade. Dois vectores dizem-se **ortogonais** se o produto interno for nulo. Diz-se que uma base vectorial é **ortonormal** se todos vectores da mesma forem mutuamente ortogonais e de norma unitária. Um espaço vectorial complexo, equipado com um produto interno, é apelidado de espaço de Hilbert. Assim, o espaço $\mathbf{W}$, que tem sido exemplificado, é um espaço de Hilbert.

2.1.2 Transformações Lineares

Sejam W e V dois espaços vectoriais sobre um mesmo corpo $\mathbb{F}$, com respectivas bases $\mathcal{B}_W = (\mathbf{w}_{e_1}, \ldots, \mathbf{w}_{e_n})$ e $\mathcal{B}_V = (\mathbf{v}_{e_1}, \ldots, \mathbf{v}_{e_n})$.

Definição 5 (Transformação Linear)**.** Uma **transformação linear** de W para V é uma função $T : W \to V$ tal que:

$$T(a\mathbf{u} + b\mathbf{v}) = a(T\mathbf{u}) + b(T\mathbf{v}) \tag{2.8}$$

para todo $a, b \in \mathbb{F}$ e $\mathbf{u}, \mathbf{v} \in W$. ■

O conjunto de todas as transformações lineares de W para V é representado por $\mathcal{L}(W, V)$. Uma transformação linear de W para W é chamada de operação linear em W. Um operador linear num espaço vectorial complexo é chamado de operador complexo [12].

Se W e V são espaços vectoriais sobre $\mathbb{F}$, com bases vectoriais definidas, então uma transformação linear $T : W \to V$, é determinada pelas operações sobre os vectores $\mathbf{u}_j$.

Cada um dos n vectores $T\mathbf{u}_j$ são definidos pela combinação linear entre $\mathbf{w}_i$ e os escalares $a_{1j}, \ldots, a_{mj} \in \mathbb{F}$, tal como na equação 2.9.

$$T\mathbf{u}_j = \sum_{i=1}^{m} a_{ij}\mathbf{w}_i \tag{2.9}$$

Assim a transformação T é determinada através de m escalares a_{ij}. Estes escalares definem uma matriz A, de dimensão $n \times m$ chamada de matriz de T, tal que:

$$A = \begin{bmatrix} a_{1,1} & \ldots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \ldots & a_{n,m} \end{bmatrix} \tag{2.10}$$

A equação 2.9 pode então ser representada em forma matricial, como na equação 2.11.

$$\begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = A \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \tag{2.11}$$

2.2 Notação de Dirac

Ao longo dos próximos capítulos vai ser usada, com frequência, a notação de Dirac. Esta foi introduzida pelo físico inglês Paul Dirac, em [13], de forma a simplificar a representação vectorial associada aos estados dos sistemas na mecânica quântica.

Definição 6 (*ket*)**.** Chama-se de *ket* a um vector de dimensão qualquer (finita ou infinita) que se faz representar pelo símbolo $|\quad\rangle$. Neste pode ainda ser colocada uma etiqueta de identificação, como 'w', ficando $|w\rangle$.

$$\mathbf{w} = |w\rangle = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \qquad w_1, \ldots, w_n \in \mathbb{C}$$

■

Nesta notação estão definidas as operações de adição e multiplicação. Somar vectores *ket* ou multiplicar os mesmos por números complexos tem como resultado um novo *ket*. Na equação 2.12 verifica-se que uma combinação linear entre *kets*, origina um novo *ket*.

$$\alpha\,|w\rangle + \beta\,|u\rangle = |k\rangle \tag{2.12}$$

Onde α e β representam números complexos.

Definição 7 (*bra*)**.** Chama-se *bra*, representado por $\langle\quad|$, ao vector do espaço dual do *ket*. Pode, como no *ket*, ser colocada uma etiqueta identificadora. ■

Existe uma correspondência directa entre um vector *bra* e um *ket*. Cada *bra* é o transposto conjugado do respectivo *ket*, como na representado na equação 2.13.

$$\langle w| = \overline{|w\rangle}^T = |w\rangle^\dagger \tag{2.13}$$

O produto entre *bra* e *ket*, que é representado por *bracket*, $\langle \quad | \quad \rangle$, designa-se por produto interno, como apresentado na equação 2.14.

$$\langle u|w\rangle = |u\rangle^\dagger |w\rangle \tag{2.14}$$

Diz-se que um *bra* está completamente definido sse o produto interno com todos os vectores *ket* estiver definido. Assim, a qualquer *bracket* está associado um número e a qualquer *bra* ou *ket* está associado um vector.

É então equivalente representar um vector segundo um *bra* ou *ket*, desde que se verifique a igualdade apresentada na equação 2.13.

$$|w\rangle = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \quad \equiv \quad \langle w| = \begin{bmatrix} \overline{w_1} & \overline{w_2} & \dots & \overline{w_n} \end{bmatrix}$$

Diz-se que $\langle u|$ e $|w\rangle$ são ortogonais entre si se o produto interno, $\langle u|w\rangle$, for igual a zero. O produto externo é definido por $|w\rangle\langle u|$, segundo a equação 2.15.

$$|w\rangle\langle w| = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \cdot \begin{bmatrix} \overline{u_1} & \dots & \overline{u_n} \end{bmatrix} \tag{2.15}$$

2.3 Produto Tensorial

Sejam dois espaços vectoriais complexos W e V, com respectivas bases $\mathscr{B}_W = (\mathbf{w}_{e_1}, \dots, \mathbf{w}_{e_n})$ e $\mathscr{B}_V = (\mathbf{v}_{e_1}, \dots, \mathbf{v}_{e_n})$.

Definição 8 (Produto Tensorial)**.** Chama-se **produto tensorial** à operação bilinear $V \otimes W$, onde o resultado é também um espaço vectorial complexo. ■

Sejam duas matrizes, A e B, de respectiva dimensão $m \times n$ e $r \times s$, de quaisquer elementos.

Definição 9 (Produto de Kronecker)**.** Chama-se **produto de Kronecker** à operação $A \otimes B$, tal que:

$$A \otimes B = \begin{bmatrix} a_{1,1}B & \dots & a_{1,n}B \\ a_{2,1}B & \dots & a_{2,n}B \\ \vdots & \vdots & \vdots \\ a_{m,1}B & \dots & a_{m,n}B \end{bmatrix} \tag{2.16}$$

■

O resultado apresentado na equação 2.16 pode também ser obtido através da relação entre duas combinações lineares, distintas ou não. A demonstração pode ser encontrada em [14].

Considere-se as duas matrizes A e B e $\alpha, \beta \in \mathbb{C}$. De forma geral, as propriedades do produto de Kronecer estão definidas tal que:

- $A \otimes (\alpha B) = \alpha(A + B)$
- O produto é distributivo em respeito à adição, tal que:
  - $(A + B) \otimes C = A \otimes C + B \otimes C$
  - $A \otimes (B + C) = A \otimes B + A \otimes C$
- $A \otimes (B \otimes C) = (A \otimes B) \otimes C$
- $(A \otimes B)(C \otimes D) = AC \otimes BD$

Ambos os produtos apresentados, embora distintos na definição, podem ser relacionados entre si. O primeiro define a operação sobre os espaços vectoriais e o segundo a operação sobre as bases do mesmo.

2.3.1 Notação de Dirac

Consideram-se os dois espaços vectoriais, W e V, anteriormente introduzidos. Sejam dos vectores, $\mathbf{w} \in \mathbf{W}$ e $\mathbf{v} \in \mathbf{V}$, de dimensão n, tal que:

$$|v\rangle = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \quad |w\rangle = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \tag{2.17}$$

O produto tensorial $W \otimes V$ é dado por:

$$|v\rangle \otimes |w\rangle = \begin{bmatrix} v_1 \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \\ \vdots \\ v_n \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} \end{bmatrix} \tag{2.18}$$

Onde $|v\rangle \otimes |w\rangle$ pode ser também representado por $|v\rangle\,|w\rangle = |v, w\rangle = |vw\rangle$.

2.4 Introdução à teoria da computação

A teoria da computação procura responder à questão: Quais são as capacidades e limitações de um computador? [15]. Esta parece, de certa forma, ter uma resposta directa, dado que é fácil identificar e explicar o funcionamento do mesmo. No entanto, formalmente, a definição é mais abrangente e complexa do que o utilizador comum considera. Os aspectos centrais desta teoria são as tarefas que podem ser executadas (algoritmos, programas) e não a forma como fisicamente o computador é constituído [16].

A necessidade de resolver certos problemas matemáticos e lógicos levou à criação da teoria da computação. As três áreas fundamentais desta são Automatos, Computabilidade e a Complexidade. Cada uma destas áreas estuda propriedades diferentes da teoria da computação [15].

Das três áreas apresentadas em cima, as duas primeiras são as que se ocupam com os fundamentos e propriedades de um computador. São então as mais fundamentais. Na primeira estão definidos, para além dos conceitos elementares, os modelos mais básicos para a computação como máquinas de estado finitas, *pushdown*, entre outras [16].

Embora os modelos sejam abrangentes a vários tipos de aplicações, estes continuam a ser demasiado restritos para um modelo geral da computação. Assim, em 1936, foi proposto por Alan Turing, um modelo mais poderoso chamado Máquina de Turing. Este é muito mais robusto para um modelo genérico da computação [16].

A complexidade, ou teoria da complexidade, é a última grande área presente na teoria da computação. As tarefas que um computador tem que resolver podem ser mais ou menos complexas. A ordenação de números por ordem crescente é um exemplo de uma tarefa que um computador básico consegue realizar. No entanto, existem outras tarefas que requerem maior capacidade de processamento. A teoria da complexidade ocupa-se em perceber como é que se consegue distinguir entre os vários níveis de complexidade. Assim foi desenvolvida uma abordagem elegante e precisa para classificar os vários níveis de complexidade, tal como apresentado de seguida [15].

2.4.1 Teoria da Complexidade

Na ciência da computação um problema é algo que se quer resolver computacionalmente. Tipicamente está estruturado num esquema de entrada/saída. O objectivo será obter na saída os resultados desejados. Mesmo que o problema possa ser, em teoria, computacionalmente solúvel, na prática pode necessitar de um número exponencial de recursos, tanto em tempo como espaço. Estes problemas são atribuídos à classe de complexidade exponencial. No entanto, muitas vezes esta limitação pode ser ultrapassada através de um estudo mais pormenorizado do problema, conseguindo assim transformar a complexidade exponencial em polinomial. Esta abordagem é sempre vantajosa, uma vez que mesmo não sendo possível, ganha-se sempre informação sobre o problema em estudo. [15, 17]

As classes mais básicas da complexidade, [17], são:

- **P** é a classe de todos os problemas possíveis de **resolver** em tempo polinomial.
- **NP** é a abreviatura para *Nondeterministic Polynomial*. Esta classe é uma das mais importantes de toda a teoria. É definida por todos os problemas onde, dado o resultado é possível **provar**, em tempo polinomial, a sua veracidade.
- **PSPACE** é a classe de problemas possíveis de resolver em espaço polinomial. Análogo a **P**, mas tendo em conta o espaço.
- **EXP** é a classe de problemas que são solúveis através de recursos exponenciais.

Uma das classes mais importante é a **NP**. Esta contém os problemas onde existe uma forma de provar, com recursos polinomiais, a veracidade da resposta dada. É fácil perceber que $\mathbf{P} \subseteq \mathbf{NP}$ - se é possível obter a resposta então é fácil convencer que esta é a correta, mesmo sem mais informação. No entanto, a questão $\mathbf{P} = \mathbf{NP}$ continua a ser um dos maiores desafios da ciência da computação. Se a veracidade da resposta é fácil de **verificar**, será o problema fácil de **resolver**? [17]

Capítulo 3

Computação Quântica

Pretende-se com este capítulo introduzir a arquitectura fundamental da computação quântica. Na secção 3.1 e 3.3 são apresentados os conceitos algébricos essenciais para a construção de sistemas com um ou mais qubits, tendo como base os conceitos introduzidos no capitulo 2. A secção 3.2 apresenta uma forma útil de visualizar os qubits no espaço, através da esfera de Bloch. Na secção 3.4 são apresentadas duas propriedades fundamentais dos sistemas quânticos, a interferência e o entrelaçamento. Nas secções 3.5 e 3.6 são apresentadas as várias portas lógicas, clássicas, quânticas e as propriedades das mesmas. Por fim, em 3.7, são introduzidos os diagramas para circuitos quânticos.

3.1 Bits e Qubits

O bit é o elemento fundamental da computação e da informação. Este apresenta apenas dois estados, que podem ser interpretados como variáveis lógicas (verdadeiro/falso), algébricas (+/-) ou qualquer outro atributo que tenha características binárias. Os estados de um bit são normalmente representados por 0 e 1. Um bit pode ser concretizado fisicamente através de dispositivos como, por exemplo, um interruptor, ver figura 3.1, ou um dispositivo mecânico que pode tomar uma de duas posições distintas. Estes sistemas têm de ser suficientemente estáveis e imunes ao ruído de forma a que os estados não permutem espontaneamente entre si [18, 19].

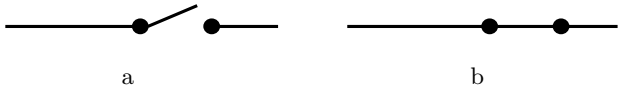


Figura 3.1: Esquemático de um interruptor, fechado em (a) e aberto em (b).

Um bit pode ser representado de forma matricial, recorrendo à notação de Dirac, tal como apresentado nas equações 3.1 e 3.2. Representa-se o estado F por $|0\rangle$, ou seja, por uma matriz 2×1 com 1 na linha dos 0s e 0 na linha dos 1s, como nas equações 3.1.

$$|0\rangle = \begin{matrix}\mathbf{F}\\ \mathbf{V}\end{matrix}\begin{bmatrix}1\\0\end{bmatrix} \tag{3.1}$$

Analogamente, o estado V é representado por $|1\rangle$, como na equação 3.2.

$$|1\rangle = \begin{matrix}\mathbf{F}\\ \mathbf{V}\end{matrix}\begin{bmatrix}0\\1\end{bmatrix} \tag{3.2}$$

Tal como em sistemas clássicos, nos sistemas quânticos é necessário definir a unidade de informação que consiga, em certo sentido, formar uma combinação linear entre os estados $|0\rangle$ e $|1\rangle$. Para isso, introduz-se a definição de qubit.

Definição 10 (Qubit)**.** Um bit quântico, ou **qubit**, é a unidade fundamental da informação de um sistema quântico.

Um qubit pode representar, para além dos estados fundamentais $|0\rangle$ ou $|1\rangle$, uma combinação linear dos mesmos tal que:

$$|\psi\rangle = \begin{matrix}\mathbf{F}\\ \mathbf{V}\end{matrix}\begin{bmatrix}\alpha\\\beta\end{bmatrix} = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \alpha, \beta \in \mathbf{C} \tag{3.3}$$

Com a condição de normalização:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{3.4}$$

Onde $|\psi\rangle$ representa o estado geral do qubit e os coeficientes α e β são números complexos que representam as amplitudes de probabilidade associadas a cada estado. Para garantir que as probabilidades estão normalizadas à unidade, é imposta a condição apresentada na equação 3.4. A normalização garante que, quando se observa um qubit, encontra-se $|0\rangle$ com probabilidade $|\alpha|^2$ ou $|1\rangle$ com probabilidade $|\beta|^2$. Não é possível, no entanto, obter o valor de cada uma das probabilidades individualizada.

3.2 Representação de um Qubit

Uma forma intuitiva, mas aproximada, de visualizar a evolução do estado de um qubit é considerar este como um vector unitário dentro de uma superfície esférica, chamada de esfera de Bloch. Nesta os estados do qubit vão estar representados na sua superfície.

Considera-se a equação 3.3 transcrita em notação polar, tal que

$$|\psi\rangle = |\alpha| e^{j\theta_\alpha}\,|0\rangle + |\beta| e^{j\varphi_\beta}\,|1\rangle \tag{3.5}$$

onde $|\alpha|$, $|\beta|$, θ_α e φ_β representam números reais.

A equação 3.5 indica que é necessário representar quatro números reais. No entanto, é possível simplificar a mesma tendo em conta algumas propriedades dos sistemas quânticos. Segundo a mecânica quântica, um estado quântico não se altera se for multiplicado por um qualquer número complexo, por exemplo $\delta \in \mathbb{C}$, de norma unitária. Considerando esta propriedade pode-se simplificar a equação 3.5 de forma a anular uma das suas componentes de fase.

Considera-se um número complexo $\delta \in \mathbb{C}$ tal que:

$$\delta = e^{-j\theta_\alpha} \tag{3.6}$$

Ao multiplicar δ pela equação 3.5 vem:

$$|\psi\rangle = \delta\,|\psi\rangle = e^{-j\theta_\alpha \alpha}\,|\psi\rangle = |\alpha|\,|0\rangle + |\alpha| e^{j\gamma}\,|1\rangle \tag{3.7}$$

$$\gamma = (\varphi_\beta - \theta_\alpha), \gamma \in \mathbb{R}$$

Desta forma, através da equação 3.7, conclui-se que apenas é necessário representar três números reais $|\alpha|$, $|\beta|$ e γ.

Considerando a equação 3.7 descrita através de coordenadas cartesianas, vem que:

$$|\psi\rangle = |\alpha|\,|0\rangle + (x + jy)\,|1\rangle \tag{3.8}$$

Onde $|\alpha|$, x e z representam números reais. A partir das equações 3.8 e 3.4, vem que:

$$|\alpha|^2 + |(x + jy)|^2 = 1$$

$$x^2 + y^2 + |\alpha|^2 = 1 \tag{3.9}$$

Por simplificação pode-se considerar $z = |\alpha|^2$. Assim a equação 3.9 representa a equação de uma esfera em $\mathbb{R}^3$, tal que:

$$\begin{aligned} x &= r\sin(\theta)\cos(\gamma) \\ y &= r\sin(\theta)\sin(\gamma) \\ z &= r\cos(\theta) \end{aligned} \tag{3.10}$$

Dada a condição de normalização vem $r = 1$.

Assim considera-se uma superfície esférica S, representada na figura 3.3, tal que:

$$S = \{(r, \theta, \gamma) \in \mathbb{R}^3 : r = |\alpha|^2 + |\beta|^2 = 1, \quad 0 < \theta < 2\pi, 0 < \gamma < \pi\} \tag{3.11}$$

A superfície esférica S representa o espaço geométrico onde todos os valores das probabilidades vão estar representados. Nesta superfície é possível observar a evolução das mesmas à medida que se realizam operações sobre o estado geral do sistema.

Tendo em conta a condição de normalização, apresentada na equação 3.4 e a regra trigonométrica $\sin^2\theta + \cos^2\theta = 1$, considera-se que:

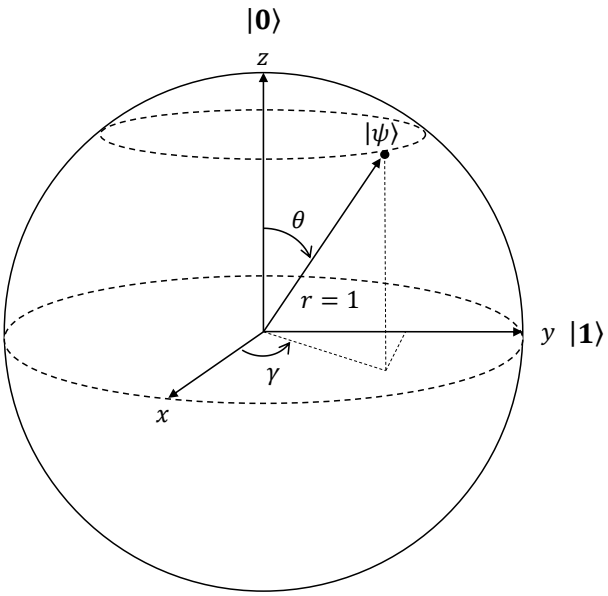


Figura 3.2: Representação da superfície S com raio unitário. Os valores θ e γ representam, respectivamente, os ângulos de latitude e longitude. O qubit $|\psi\rangle$ está definido na superfície da esfera.

$$\begin{aligned} |\alpha| &= \sin\theta \\ |\beta| &= \cos\theta \end{aligned} \tag{3.12}$$

Substituindo na equação 3.7 vem que:

$$|\psi\rangle = \cos\theta\,|0\rangle + e^{j\gamma}\sin\theta\,|1\rangle \tag{3.13}$$

As equações 3.12 e 3.13 mostram que as probabilidades $|\alpha|^2$ e $|\beta|^2$ estão definidas nas projecções do vector $|\psi\rangle$, segundo θ, nos eixos y e z, respectivamente. Ao fazer variar o valor de θ, através da equação 3.13 vem que:

$$|\psi\rangle = \begin{cases} |0\rangle\,, & \theta = 0 \\ |1\rangle\,, & \theta = \frac{\pi}{2} \end{cases} \tag{3.14}$$

Ou seja, o estado $|0\rangle$ ou $|1\rangle$ são observados quando o valor de θ é 0 ou $\pi/2$, respectivamente.

Ao analisar a figura 3.2 verifica-se que todos os estados do qubit estão definidos nos intervalos $0 < \theta < \pi/2$ e $0 < \gamma < 2\pi$ isto é, no hemisfério norte da esfera.

Para simplificar a representação considera-se $\theta' = 2\theta$ ou seja, uma revolução completa em θ' equivale a meia revolução em θ. Isto faz com que os estados $|0\rangle$ e $|1\rangle$ sejam representados por pontos antipodais. Na figura 3.3 está representada a esfera de Bloch segundo as características do sistema em estudo.

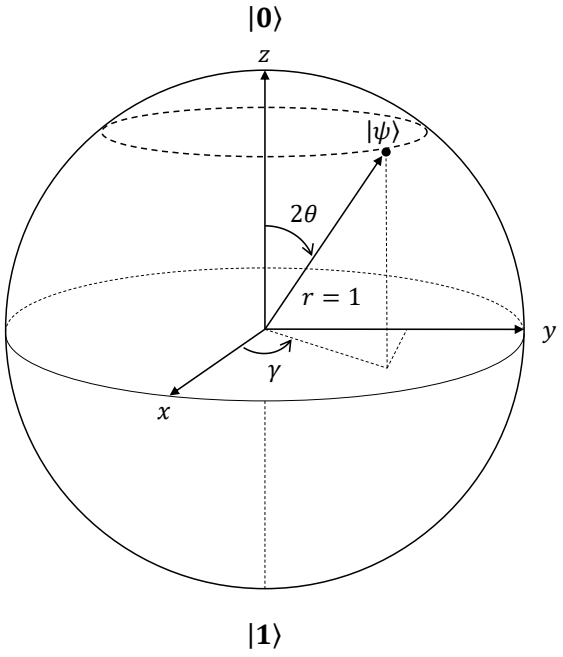


Figura 3.3: Representação da esfera de Bloch. Os valores $|0\rangle$ é antipodal a $|1\rangle$.

O estado geral do qubit, $|\psi\rangle$, é agora dado por:

$$|\psi\rangle = \cos(\theta/2)\,|0\rangle + e^{j\gamma}\sin(\theta/2)\,|1\rangle \tag{3.15}$$

3.3 Sistemas Multi-Qubit

Nas secções anteriores foram apresentados os conceitos base para a construção de sistemas com apenas um qubit. Assim, nesta secção, estes conceitos são generalizados para sistemas com mais do que um qubit (multi-qubit). Estes são elementos estruturantes para a construção de outros mais complexos.

Um sistema multi-qubit, também conhecido por **registo** de qubits, é constituído por n qubits ordenados e indexados, onde $n > 1$. Este registo é construido de forma a que seja possível realizar operações num qubit especifico ou num conjunto dos mesmos [19].

Seja, por exemplo, um registo de dimensão três, onde $\mathscr{H}_A$, $\mathscr{H}_B$ e $\mathscr{H}_C$ são espaços de Hilbert subjacentes a cada qubit, com respectivas bases $\{|0\rangle_A, |1\rangle_A\}$, $\{|0\rangle_B, |1\rangle_B\}$ e $\{|0\rangle_C, |1\rangle_C\}$. O estado de cada qubit do registo é, por exemplo, apresentado na equação 3.16.

$$|\psi\rangle = \begin{cases} |\psi_0\rangle = |0\rangle_A \\ |\psi_1\rangle = |1\rangle_B \\ |\psi_2\rangle = |0\rangle_C \end{cases} \tag{3.16}$$

Onde $|\psi_0\rangle$ e $|\psi_2\rangle$ representam os qubits menos e mais significativos, respectivamente. Assim, os qubits são sempre representados por ordem crescente, do menos para o mais significativo.

O estado geral do registo é obtido através da aplicação do produto tensorial a todos os qubits do registo, como na equação 3.17.

$$|\psi\rangle = |\psi_0\rangle \otimes |\psi_1\rangle \otimes |\psi_2\rangle = |0\rangle \otimes |1\rangle \otimes |0\rangle = |010\rangle = |2\rangle \tag{3.17}$$

As etiquetas presentes nos *kets* da equação 3.17 podem, como mostrado, ser representadas por um número em base decimal, $(010)_2 = (2)_{10}$, ou em outra qualquer notação. Nesta dissertação vai-se fazer uso da base binária, salvo algumas excepções onde será útil considerar a base decimal.

Na equação 3.17 mostra-se como construir o estado geral de um registo de qubits. Este pode, no entanto, ser formado por qubits que se encontram definidos numa combinação linear dos estados fundamentais. Considera-se, por exemplo, que o qubit menos significativo, do registo da equação 3.16, está definido como $|\psi_0\rangle = \alpha_0 |0\rangle + \beta_0 |1\rangle$. O estado geral do registo é agora dado por:

$$|\psi\rangle = |\psi_2\rangle \otimes |\psi_1\rangle \otimes |\psi_0\rangle = \alpha_0 |010\rangle + \beta_0 |110\rangle = \begin{matrix} \mathbf{FFF} \\ \mathbf{FFV} \\ \mathbf{FVF} \\ \mathbf{FVV} \\ \mathbf{VFF} \\ \mathbf{VFV} \\ \mathbf{VVF} \\ \mathbf{VVV} \end{matrix} \begin{bmatrix} 0 \\ 0 \\ \alpha_0 \\ 0 \\ 0 \\ 0 \\ \beta_0 \\ 0 \end{bmatrix} \tag{3.18}$$

De forma geral, para um registo $|\psi\rangle$ com n quaisquer qubits vem:

$$|\psi\rangle = c_0 |00\ldots0\rangle + \cdots + c_{2^n-1} |11\ldots1\rangle = \sum_i c_i |i\rangle \tag{3.19}$$

Onde $i = 0, \ldots, 2^n - 1$ representa a etiqueta do valor decimal correspondente a cada estado e c_i a amplitude complexa associada ao mesmo. A dimensão do vector coluna resultante, ou seja a quantidade de números complexos do registo, cresce exponencialmente com o número de qubits no mesmo, como se pode ver pela equação 3.19. Este crescimento exponencial é um dos problemas que torna inviável a simulação deste tipo de sistemas em computadores clássicos [18].

3.4 Interferência e Entrelaçamento

Quando se questiona quais os fenómenos que permitem tornar a computação quântica mais poderosa, irremediavelmente fala-se em interferência e entrelaçamento. O primeiro, interferência, acontece quando existe mais do que uma forma de obter um determinado resultado computacional. Isto é, as várias formas podem contribuir tanto construtivamente como destrutivamente para a probabilidade desse resultado.

O entrelaçamento, ou *entanglement*, é um dos fenómenos mais curioso e cruciais da computação quântica. Muitos dos algoritmos e resultados quânticos baseiam-se neste fenómeno levando, por exemplo, à aceleração exponencial.

Diz-se que um conjunto de qubits, ou registo, está **entrelaçado**, ou ***entangled***, sse não for possível factorizar o estado geral do registo para os estados individuais dos qubits. Seja $|\psi\rangle_A$ e $|\psi\rangle_B$ dois qubits e $|\psi\rangle$ o estado geral do registo que contêm os mesmos. Diz-se que o registo está entrelaçado sse:

$$|\psi\rangle \neq |\psi_A\rangle \otimes |\psi_B\rangle \tag{3.20}$$

Seja agora o estado $|\psi\rangle$ dado, por exemplo, por:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0_A 0_B\rangle + |1_A 1_B\rangle) \tag{3.21}$$

Segundo a equação 3.21, se o resultado da observação do qubit A for, por exemplo, $|0\rangle$, implica que o qubit B seja também $|0\rangle$, fazendo com que $|\psi\rangle = |00\rangle$. Se o resultado for $|1\rangle$, então $|\psi_B\rangle = |1\rangle$ logo $|\psi\rangle = |11\rangle$. Assim, a observação do qubit A influência o estado do qubit B.

As propriedades apresentadas nesta secção são duas das mais importantes na computação quântica. Para sistemas mais complexos, como circuitos e algoritmos quânticos, estas permitem obter resultados vantajosos face a sistemas clássicos idênticos.

3.5 Portas Lógicas Clássicas

A lógica apresenta-se como uma sub-área da matemática, que se preocupa com a veracidade de argumentos, isto é, determinar se uma dada proposição é verdadeira ou falsa. Esta é uma ferramenta matemática útil para retirar conclusões a partir de um ponto de partida válido [18]. Nesta secção vai-se introduzir os conceitos bem conhecidos de portas lógicas e propriedades como reversibilidade e universalidade.

O conceito de porta lógica foi introduzido pelo matemático Britanico George Boole. Uma porta lógica clássica é, usualmente, considerada como um dispositivo físico com uma ou mais entradas e saídas. Estes dispositivos trabalham com valores booleanos (Verdadeiro ou Falso) tanto na entrada como na saída. Normalmente, estes valores podem ser também representados pelos simbolos 1 e 0, respectivamente [18].

Um bit, ou um conjunto de bits, pode ser representado como um vector coluna, como já enunciado em 3.1. Para descrever a operação das portas lógicas considera-se, como input, um vector de dimensão $2^n \times 1$ e como output $2^m \times 1$. Desta forma, a operação da porta lógica é caracterizada por uma matriz $2^n \times 2^m$, tal que:

$$(2^n \times 2^m)(2^n \times 1) = (2^m \times 1) \tag{3.22}$$

Um dos exemplos mais triviais é a porta NOT. Esta é caracterizada uma matriz de $2^1 \times 1$, ou seja 1 bit na entrada e na saída. A sua operação baseia-se em negar o valor lógico colocado na entrada. Assim, se $|0\rangle$ na entrada, à saída surge $|1\rangle$. Se $|1\rangle$ surge $|0\rangle$.

$$NOT = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{3.23}$$

A matriz NOT satisfaz então as operações:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.24}$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.25}$$

O que corresponde ao funcionamento esperado da operação NOT.

3.5.1 Irreversibilidade

As portas lógicas tem um papel fundamental na construção de circuitos para os computadores modernos. De entre várias portas lógicas podem-se distinguir as irreversíveis. Diz-se que uma porta lógica é **irreversível** se uma configuração de bits na saída corresponder a mais do que uma combinação na entrada [18].

A porta lógica AND é uma das mais fundamentais na computação moderna. Na tabela 3.1 está representada a tabela de verdade correspondente.

Tabela 3.1: Tabela de verdade da porta lógica AND.

A	B	AND(A,B)
0	0	0
0	1	0
1	0	0
1	1	1

Uma vez que a porta AND tem duas entradas ($n = 2$) e uma saída ($m = 1$), a matriz resultante irá ser de dimenção $2^2 \times 2^1$, como apresentado na equação 3.26.

$$AND = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.26}$$

Diz-se assim que a porta lógica AND é irreversível pois o mesmo valor na saída pode corresponder a vários valores na entrada, como se pode verificar na tabela 3.1. Outros exemplos deste tipo de portas lógicas são a OR e XOR.

3.5.2 Universalidade

Existe um outro tipo de portas lógicas designadas por **universais**. Diz-se que uma porta lógica é universal quando, a partir desta, é possível construir qualquer função Booleana. As portas NAND e NOR primam por esta propriedade, uma vez que qualquer função

Tabela 3.2: Tabela de verdade da porta lógica NAND.

A	B	NAND(A,B)
0	0	1
0	1	1
1	0	1
1	1	0

booleana pode ser simplificada até conter apenas os operadores AND e NOT ou OR e NOT [18]. Na tabela 3.2 é apresentada a tabela de verdade para a porta lógica NAND.

A matriz da porta NAND é de dimensão $2^2 \times 2^1$, tal como apresentada na equação 3.27.

$$NAND = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \tag{3.27}$$

No entanto o uso restrito de portas lógicas universais não garante a minimização do circuito, mas facilita o processo de design dos mesmos [18].

3.5.3 Reversibilidade

Uma forma de resolver o problema da energia libertada pelos circuitos, consequência do uso de portas lógicas irreversíveis, é desenhar estes com portas lógicas **reversíveis**. Nestas, para qualquer valor de saída existe um e um só valor de entrada. Assim as portas lógicas reversíveis não existe destruição de informação durante a sua operação [18].

A porta lógica NOT é um exemplo trivial de uma porta lógica reversível. O seu funcionamento baseia-se em inverter o valor lógico na entrada, colocando o resultado na saída. A tabela de verdade é apresentada na tabela 3.3. Na equação 3.23 é apresentada a matriz da porta NOT.

Tabela 3.3: Tabela de verdade da porta lógica NOT.

A	NOT(A)
0	1
1	0

A partir da tabela de verdade 3.3 verifica-se que, se o valor do bit na saída for conhecido, é sempre possível recuperar o seu respectivo valor à entrada, tal como previsto.

Para além da porta NOT, existem outras portas reversíveis de igual ou maior impacto na computação. A porta CNOT é uma das mais importantes na categoria das reversíveis [18]. Nesta porta, o segundo bit (alvo) é negado se, e só se, o primeiro (controlo) estiver definido no estado 1. Através desta descrição é trivial concluir que esta porta é caracterizada por dois valores na entrada ($n = 2$) e na saída ($m = 2$). Assim, a decisão de negar ou não o target bit depende unicamente do valor do bit de controlo. Na tabela 3.4 representa-se a tabela de verdade correspondente a porta CNOT.

Tabela 3.4: Tabela de verdade da porta lógica CNOT. Do lado esquerdo os valores de entrada. Do lado direito os valores de saída.

Entrada		Saída	
A	B	A'	B'
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

A matriz, de dimensão $2^2 \times 2^2$, correspondente à porta CNOT é apresentada na equação 3.28.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.28}$$

3.6 Portas Lógicas Quânticas

Tal como nos sistemas clássicos, qualquer operação na computação quântica pode ser decomposta numa sequência de portas lógicas. A diferença, para o clássico, está no facto de estas operarem sobre qubits. Estes podem representar combinações lineares de estados fundamentais, $|0\rangle$ e $|1\rangle$, tal como descrito em 3.1.

Uma porta lógica quântica, segundo os elementos da computação quântica, ao operar sobre um sistema quântico isolado, deverá realizar operações reversíveis e unitárias, sendo estas descritas por matrizes unitárias.

Dado que a unitariedade e a reversibilidade são características obrigatórias, existem algumas portas clássicas, como a NOT, CNOT ou TOFFOLI, que vão ser de novo consideradas. Estas mantém as mesmas funções, logo também a mesma representação matricial. Nesta secção introduzem-se novas portas lógicas, como Hadamard, e revêm-se algumas portas clássicas, mas agora aplicadas aos sistemas quânticos.

3.6.1 NOT

A porta lógica NOT, também conhecida, na computação quântica, por Pauli X, é uma porta lógica quântica, reversível, tal que:

$$NOT \equiv X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{3.29}$$

A operação desta porta é idêntica ao apresentado para o caso clássico. Quando aplicado o estado $|0\rangle$ ou $|1\rangle$ à entrada, na saída obtém-se $|1\rangle$ ou $|0\rangle$, respectivamente, como descrito nas equações 3.24 e 3.25.

Esta operação, na esfera de Bloch, traduz-se na deslocação do estado inicial (fundamental) para o ponto antipodal, norte para sul ou vice-versa. No entanto, se o estado inicial do qubit for definido por uma qualquer combinação linear de estados, a operação da porta lógica não garante a transformação deste para o ponto antipodal na esfera. Assim, segundo a esfera de Bloch, o NOT não opera de forma idêntica para qualquer estado do qubit, ao contrário do que se passa no caso clássico [18]. Para não se confundir os dois casos, nesta dissertação, usa-se a designação de X sempre que se referir a porta quântica NOT.

3.6.2 Hadamard

A porta lógica Hadamard é uma das mais fundamentais na computação quântica. A sua principal operação é a de colocar o estado do qubit numa combinação linear igualmente distribuída. A porta lógica Hadamard, ou H, é definida pela matriz:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3.30}$$

A sua operação sobre os vários estados possíveis do qubit são resumidos nas tabela 3.5.

Tabela 3.5: Tabela com valores de entrada e saída *standard* para a porta lógica H.

Hadamard	
Entrada	Saída
$\|0\rangle$	$\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$
$\|1\rangle$	$\frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$
$\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$	$\|0\rangle$
$\frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$	$\|1\rangle$

Uma das grandes vantagens desta porta lógica surge quando esta é aplicada a um registo de qubits. Tem-se, por exemplo, um registo com n qubits no estado fundamental $|0\rangle$. Para colocar o registo numa combinação linear de todos os 2^n estados possíveis, é apenas necessário utilizar n portas lógicas H. Uma combinação exponencial de valores pode ser preparada, fisicamente, através de um número polinomial de recursos. Esta operação está esquematizada na figura 3.4.

Para obter o estado geral do registo é aplicado, no fim da operação das portas H, o produto tensorial entre todos os estados dos qubits, tal que:

$$H|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i}^{2^n-1} |i\rangle \tag{3.31}$$

Um caso particular interessante acontece quando se juntam em série duas portas H. Considere-se um qubit $|\psi\rangle$ definido no estado fundamental $|0\rangle$, tal que:

$$|\psi\rangle = |0\rangle \tag{3.32}$$

Ao aplicar duas portas H em série vem:

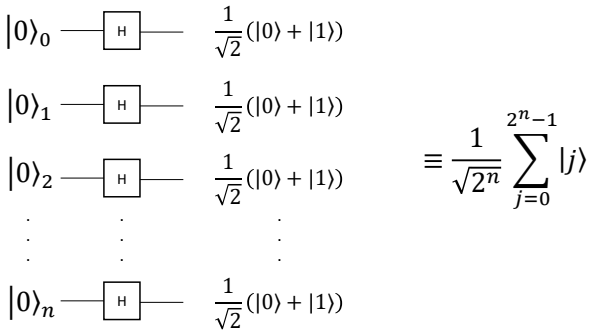


Figura 3.4: Ao aplicar n portas lógicas H em n qubits distintos, é possível colocar o registo de qubits numa combinação linear de $2^n - 1$ valores.

$$HH|0\rangle = H \times \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |0\rangle \tag{3.33}$$

Através da equação 3.33 é possível verificar que, ao aplicar a primeira porta H, o qubit fica definido segundo uma combinação linear dos seus estados com as amplitudes de probabilidade idênticas. Após ser submetido à segunda porta H, o estado resultante é igual ao inicial. Verifica-se assim que é possível obter a configuração inicial do qubit, através de um sistema de duas portas H em série. Este resultado é válido para qualquer configuração do qubit na entrada.

3.6.3 Controled-NOT

Na comutação quântica, para realizar operações não triviais é necessário recorrer a portas lógicas mais complexas, isto é com mais do que um qubit na entrada e saída. Um exemplo de uma porta lógica que com estas características é a porta CNOT. Em sistemas quânticos a operação desta é a generalização do sistema clássico. Assim, em vez de operar unicamente sobre qubits no estado fundamental, $|0\rangle$ ou $|1\rangle$, deve-se também considerar qubits definidos numa combinação linear de ambos os estados. No caso do bit de controlo estar definido numa combinação linear de estados, pela linearidade da mecânica quântica, garante-se que todas as operações são realizadas proporcionalmente em todas as componentes do segundo qubit. [18]

As portas lógicas com bits de controlo podem ser generalizadas para qualquer tipo de operação. Considera-se uma porta genérica U, de um qubit, tal que:

$$U = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{3.34}$$

Onde A, B, C e D são números complexos.

De forma a controlar, através de um qubit, a porta lógica U, defini-se a porta C-U tal que:

$$Controlled - U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & C & D \end{bmatrix} \tag{3.35}$$

A porta CNOT é então um caso particular da equação 3.35, onde a matriz U é a NOT, tal como descrito na equação 3.36.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.36}$$

Considera-se agora um sistema de multi-qubit, como na equação 3.19, com $N = 2$ tal que:

$$|\psi\rangle = \alpha_0\alpha_1 |00\rangle + \alpha_0\beta_1 |01\rangle + \beta_0\alpha_1 |10\rangle + \beta_0\beta_1 |11\rangle \tag{3.37}$$

Onde α_0, α_1, β_0 e β_1 representam números complexos. Ao aplicar a porta CNOT vem:

$$CNOT\,|\psi\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_0\alpha_1 \\ \alpha_0\beta_1 \\ \beta_0\alpha_1 \\ \beta_0\beta_1 \end{bmatrix} = \begin{bmatrix} \alpha_0\alpha_1 \\ \alpha_0\beta_1 \\ \beta_0\beta_1 \\ \beta_0\alpha_1 \end{bmatrix} \tag{3.38}$$

A equação 3.38 representa o funcionamento da porta CNOT para quaisquer dois qubits à entrada. O número de qubits, tanto de controlo como de *target*, podem aumentar, implementando portas como a TOFFOLI ou FREDKIN.

Existem, no entanto, casos singulares de configurações dos qubits à entrada. Considera-se a equação 3.38 com as amplitudes de probabilidade definidas tal que:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{3.39}$$

Ao aplicar a porta CNOT ao qubit $|\psi\rangle$ vem:

$$CNOT\,|\psi\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.40}$$

O resultado da equação 3.40 representa o estado entrelaçado dos dois qubits. Não é assim possível separar este estado em dois qubits distintos.

3.7 Diagramas de Circuitos Quânticos

Da mesma forma que os circuitos booleanos representam a base da computação actual, os circuitos quânticos também o são para a computação quântica, embora com um nível de complexidade superior. Através destes obtém-se uma representação visual de como um sistema complexo pode ser decomposto em operações (portas lógicas) de um ou mais qubits [18].

Os diagramas de circuitos quânticos esquematizam a sequência de portas lógicas a implementar para um determinado objectivo. Este é, normalmente, um algoritmo complexo. Na figura 3.5 é possível observar um exemplo, no caso uma implementação do algoritmo de Shor.

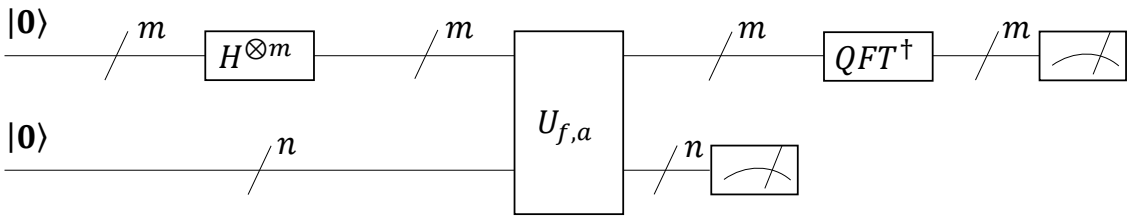


Figura 3.5: Circuito Quântico para a implementação do algoritmo de Shor. Adaptado de [19].

Um circuito quântico contêm n linhas horizontais, cada uma correspondente a um qubit. Convenciona-se, neste documento, que o qubit menos significativo está representado no topo e o mais na base. O tempo flui da esquerda para a direita, sendo as portas lógicas aplicadas no mesmo sentido pela sequência esquematizada. O detector é, por norma, o último elemento do circuito quântico. O símbolo, que se assemelha a um medidor comum, pode ser observado na figura 3.5. A função deste dispositivo é medir o valor do qubit [18]. De notar também que estes circuitos podem ser vistos como três grandes módulos, tais como Geração, Processamento e Detecção.

Capítulo 4

Computação Trans-Booleana

A necessidade de aumentar a capacidade de processamento dos dispositivos de computação tem crescido de forma acentuada nos últimos anos. Algumas tentativas de aumentar esta capacidade tem-se mostrado eficientes, mas não suficientes, como a paralelização do processamento através de, por exemplo, GPUs. A computação quântica mostra ser uma alternativa viável para a exponenciação da capacidade de processamento. Este tipo de computação exige uma mudança de paradigma do pensamento clássico, uma vez que trás consigo uma álgebra mais rica e complexa. No entanto, esta pressupõem elementos quânticos na sua implementação, o que aumenta a complexidade do sistema e, consequentemente, da escalabilidade do hardware em comparação aos sistemas clássicos hoje existentes. Numa tentativa de colmatar o problema introduz-se, neste capitulo, a Computação Trans-Booleana. Na secção 4.1 apresenta-se uma forma de codificar qubits e em 4.2 a arquitectura para implementar um sistema de Computação Trans-Booleana.

4.1 Codificação do Qubit

Considera-se um qubit, tal como definido em 3.1, de tal forma que o seu estado é descrito pela equação 4.1.

$$|\psi\rangle = \begin{matrix}\mathbf{F}\\\mathbf{V}\end{matrix}\begin{bmatrix}\alpha\\\beta\end{bmatrix} = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \alpha,\beta \in \mathbf{C} \tag{4.1}$$

Com a condição de normalização:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{4.2}$$

Onde $|\psi\rangle$ representa o estado geral do qubit e α e β são números complexos correspondentes às amplitudes de probabilidade de $|0\rangle$ e $|1\rangle$, respectivamente. Estes números complexos podem também ser representados em notação polar, tal como na equação 4.3.

$$|\psi\rangle = |\alpha|e^{j\theta_\alpha}|0\rangle + |\beta|e^{j\varphi_\beta}|1\rangle \tag{4.3}$$

Onde $|\alpha|$, $|\beta|$, θ_α e φ_β representam números reais que descrevem o qubit na sua totalidade.

A partir da equação 4.3, propõem-se de seguida uma codificação para os parâmetros de interesse. Esta é baseada na computação analógica e no processamento de sinal. Considere-se o sinal $x(t)$ tal que:

$$x(t) = Acos(wt + \theta) \tag{4.4}$$

Considerando que $A = |\alpha|$, $\theta = \theta_\alpha$ e $w > 0$ vem,

$$x(t) = |\alpha|cos(wt + \theta_\alpha) \tag{4.5}$$

A amplitude máxima do sinal na equação 4.5 representa o valor de $|\alpha|$ e o desvio de fase do sinal, θ, codifica o valor de θ_α. Como um qubit é definido, no máximo, por dois números complexos consideram-se dois sinais, $x_0(t)$ e $x_1(t)$, e um qubit $\psi(t)$, tal que:

$$\psi(t) = \begin{cases} x_0(t) = |\alpha|cos(wt + \theta_\alpha) \\ x_1(t) = |\beta|cos(wt + \varphi_\beta) \end{cases} \tag{4.6}$$

De forma geral, para representar N qubits são necessários $2N$ sinais, tal que:

$$\psi^n(t) = \begin{cases} x_0^n(t) = |\alpha_n|cos(wt + \theta_\alpha^n) \\ x_1^n(t) = |\beta_n|cos(wt + \varphi_\beta^n) \end{cases} \tag{4.7}$$

Onde $n \in \{0, 1, \ldots, N-1\}$.

4.2 Arquitectura

Os circuitos quânticos, tal como apresentados em 3.7, podem ser divididos nos módulos de Geração, Processamento e Detecção tal como esquematizado na figura 4.1. Neste caso, ao invés da codificação tradicional em função de elementos quânticos, implementa-se uma com base em sinais analógicos, tal como apresentado na secção anterior. O primeiro módulo, Geração, tem como objectivo gerar qubits e colocá-los na entrada do sistema. Consequentemente, o módulo de processamento vai, como se prevê, processar a informação dos qubits da forma desejada. Por último surge a detecção, onde se observa o resultado. As linhas que ligam os módulos transportam a informação de cada qubit. Nas secções seguintes vão ser apresentadas descrições detalhadas de cada módulo.

4.2.1 Gerador de Qubits

O módulo de geração é o mais elementar de toda a arquitectura. Relembrando a definição de um qubit, este pode ser inicializado num dos seus estados fundamentais $|0\rangle$ ou $|1\rangle$. Dada

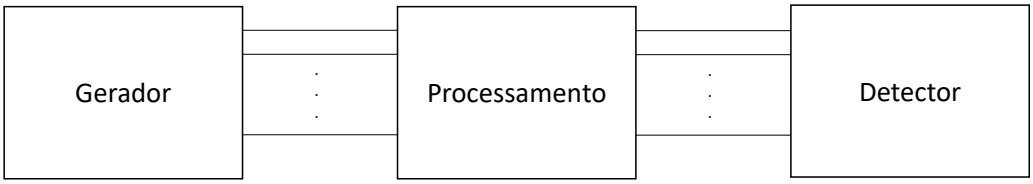


Figura 4.1: Arquitectura geral de um circuito quântico. Cada linha de ligação entre módulos representa um qubit.

a codificação da equação 4.6 basta, para definir um estado fundamental, gerar apenas um dos sinais $x_0(t)$ ou $x_1(t)$. Na figura 4.2 está esquematizado os módulos de geração para cada estado fundamental.

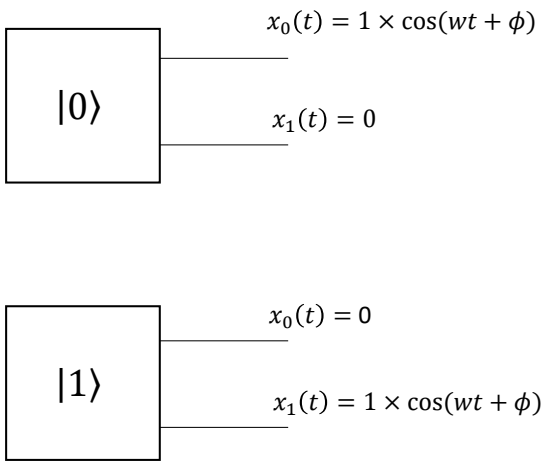


Figura 4.2: Gerador de qubits para os estados $|0\rangle$ e $|1\rangle$.

4.2.2 Processamento

O módulo de processamento baseia-se na aplicação de um conjunto de portas lógicas trans-booleanas sobre os qubits. Nesta secção vão ser apresentadas sugestões de implementações para algumas destas, com base na codificação apresentada.

4.2.2.1 NOT

A operação da porta NOT, tal como apresentada em 3.6.1, é definida pela matriz na equação 3.29. Ao aplicar a porta NOT ao qubit $|\psi\rangle$, equação 4.1, vem:

$$NOT\,|\psi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \tag{4.8}$$

Os números complexos α e β são codificados nos sinais sinusoidais $x'_0(t)$ e $x'_1(t)$. Assim para implementar a porta basta trocar os fios na saída, tal como a figura 4.3 esquematiza. No anexo I, figura I.1, está representado o esquemático simulink da porta NOT.

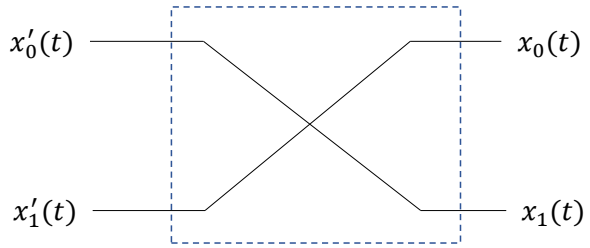


Figura 4.3: Implementação da porta NOT. $x'_0(t)$ e $x'_1(t)$ representam os sinais de entrada e $x_0(t)$ e $x_1(t)$ os de saída.

4.2.2.2 Hadamard

A operação da porta H, tal como apresentada na secção 3.6.2, é definida pela equação 4.9.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4.9}$$

Considerando o qubit e a respectiva condição de normalização, dadas respectivamente nas equações 4.1 e 4.2, vem que:

$$H\,|\psi\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha+\beta \\ \alpha-\beta \end{bmatrix} \tag{4.10}$$

$$H\,|\psi\rangle = \frac{1}{\sqrt{2}}((\alpha+\beta)\,|0\rangle + (\alpha-\beta)\,|1\rangle) \tag{4.11}$$

Onde, nas equações 4.10 e 4.11, α e β representam números complexos que são codificados em sinais sinusoidais $x'_0(t)$ e $x'_1(t)$, respectivamente. Os números complexos resultantes da operação da porta H são codificados nos sinais $x_0(t)$ e $x_1(t)$, tal como apresentado nas equações 4.12 e 4.13, respectivamente.

$$x_0(t) = (x'_0(t) + x'_1(t))\frac{1}{\sqrt{2}} \tag{4.12}$$

$$x_1(t) = (x'_0(t) - x'_1(t))\frac{1}{\sqrt{2}} \tag{4.13}$$

Assim para implementar a porta H basta considerar um bloco somador e um subtractor, tal como representado na figura 4.4.

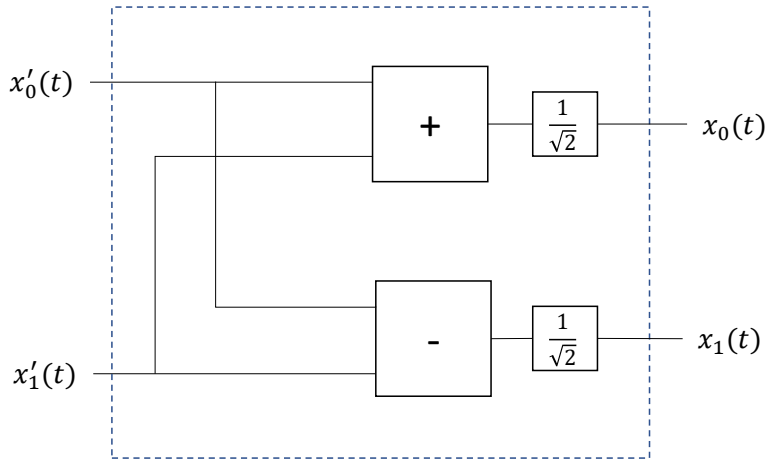


Figura 4.4: Implementação da porta H. $x_0'(t)$ e $x_1'(t)$ representam os sinais de entrada e $x_0(t)$ e $x_1(t)$ os de saída.

Para simular o funcionamento do sistema foi usada a ferramenta de simulação Simulink. O esquema que modela a porta H está representado no anexo I, figura I.2. Considera-se, daqui em diante, que os parâmetros de simulação correspondentes à frequência dos sinais, tempo de amostragem e tempo de simulação são:

$$f = 60Hz$$
$$T_s = \frac{1}{8000}s$$
$$T_{sim} = 0.5s$$

Tome-se agora o primeiro caso da tabela 3.5 onde o qubit é definido no seu estado fundamental, tal que:

$$\psi(t) = \begin{cases} x_0(t) = 1cos(wt) \\ x_1(t) = 0 \end{cases} \tag{4.14}$$

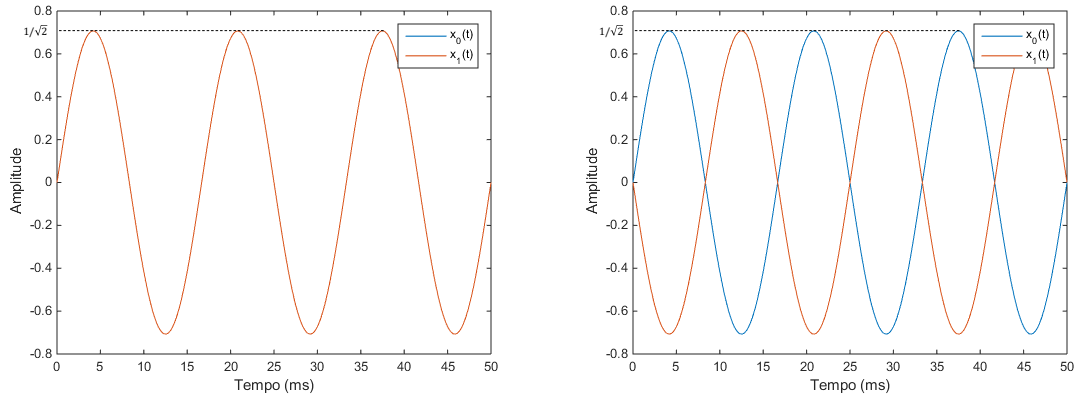
Ao aplicar a porta H no qubit da equação 4.14 vem:

$$\psi(t) = \begin{cases} x_0(t) = \frac{1}{\sqrt{2}}cos(wt) \\ x_1(t) = \frac{1}{\sqrt{2}}cos(wt) \end{cases} \tag{4.15}$$

Espera-se então que os sinais fiquem idênticos, tal como apresentado na figura 4.5a. Para o caso do qubit definido no estado $|1\rangle$ o raciocínio é análogo existindo apenas uma diferença de fase entre os dois sinais, como se pode verificar na figura 4.5b.

Os restantes casos da tabela 3.5 consideram que o qubit de entrada está numa combinação linear dos dois estados fundamentais, como na equação 4.16.

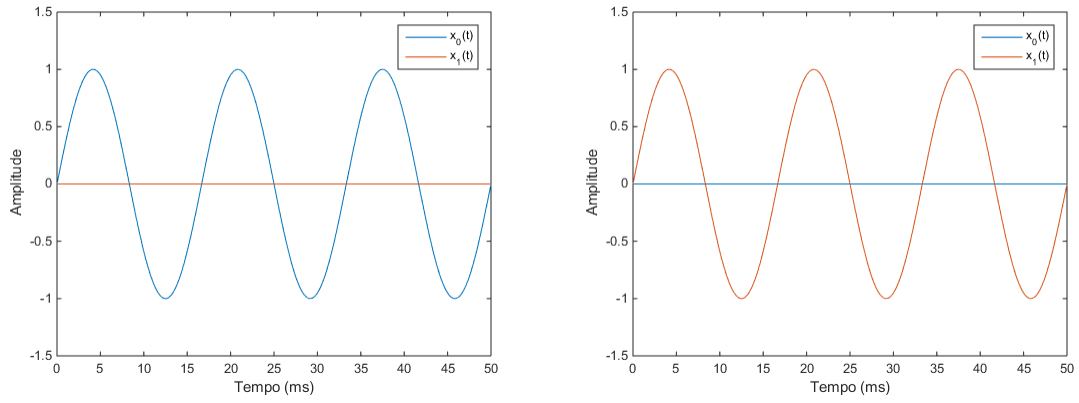
$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle) \tag{4.16}$$



a b

Figura 4.5: Sinais de saída da porta H para qubits no estado fundamental à entrada. Na figura a) representa-se o resultado para o qubit definido como $|0\rangle$, onde os sinais $x_0(t)$ e $x_1(t)$ estão sobrepostos. Na figura b) representa-se o resultado para o qubit definido como $|1\rangle$ com os sinais desfasados de π.

O resultado da operação vai ser um dos estados fundamentais, $|0\rangle$ ou $|1\rangle$, dependendo da fase do qubit à entrada. Se ambos os sinais estiverem definidos com amplitudes idênticas e fase nula, ao aplicar a porta H obtém-se o estado $|0\rangle$, tal como na figura 4.6a. Caso o sinal associado ao estado $|1\rangle$ esteja definido com um atraso de fase de 180 graus, ao aplicar a porta H obtém-se o estado $|1\rangle$, como na figura 4.6b.



a b

Figura 4.6: Sinais de saída da porta H para um qubit numa sobreposição de estados, segundo a equação 4.16. Na figura a) representa-se o resultado para o qubit definido como $|0\rangle + |1\rangle$. Na figura b) representa-se o resultado para o qubit definido como $|0\rangle - |1\rangle$.

Verifica-se assim que é possível obter todos os estados descritos na tabela 3.5. Uma vez que a porta H é implementada na sua forma mais genérica é possível usar esta porta com qualquer configuração do qubit na entrada.

Um caso interessante surge quando se colocam duas portas H em série, tal como descrito em 3.6.2 e esquematizado na figura 4.7.

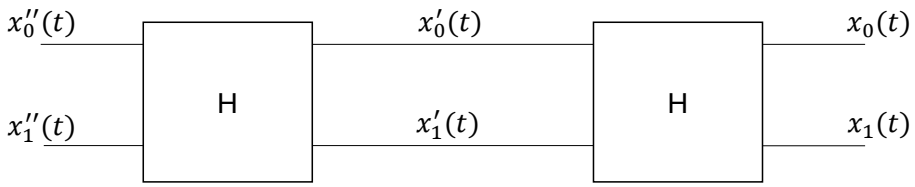


Figura 4.7: Esquema de duas portas H em série. Os sinais $x_0''(t)$ e $x_1''(t)$ representam os sinais de entrada e $x_0(t)$ e $x_0(t)$ os de saída.

Neste caso espera-se que o valor na saída corresponda ao que foi inicialmente colocado na entrada da primeira porta. No caso da figura 4.7 espera-se que:

$$\begin{aligned} x_0(t) &= x_0''(t) \\ x_1(t) &= x_1''(t) \end{aligned} \tag{4.17}$$

Na figura 4.8 estão representados os resultados da modelação do sistema em simulink, segundo o esquema apresentado no anexo I. Na figura 4.8a observam-se os sinais de entrada, para um qubit definido no estado fundamental $|0\rangle$. Após a primeira porta H verifica-se, na figura 4.8b que ambos os sinais ficam idênticos. Por último, após a segunda porta H verifica-se, figura 4.8c, que o sinal é idêntico ao que foi inicialmente introduzido na primeira porta H.

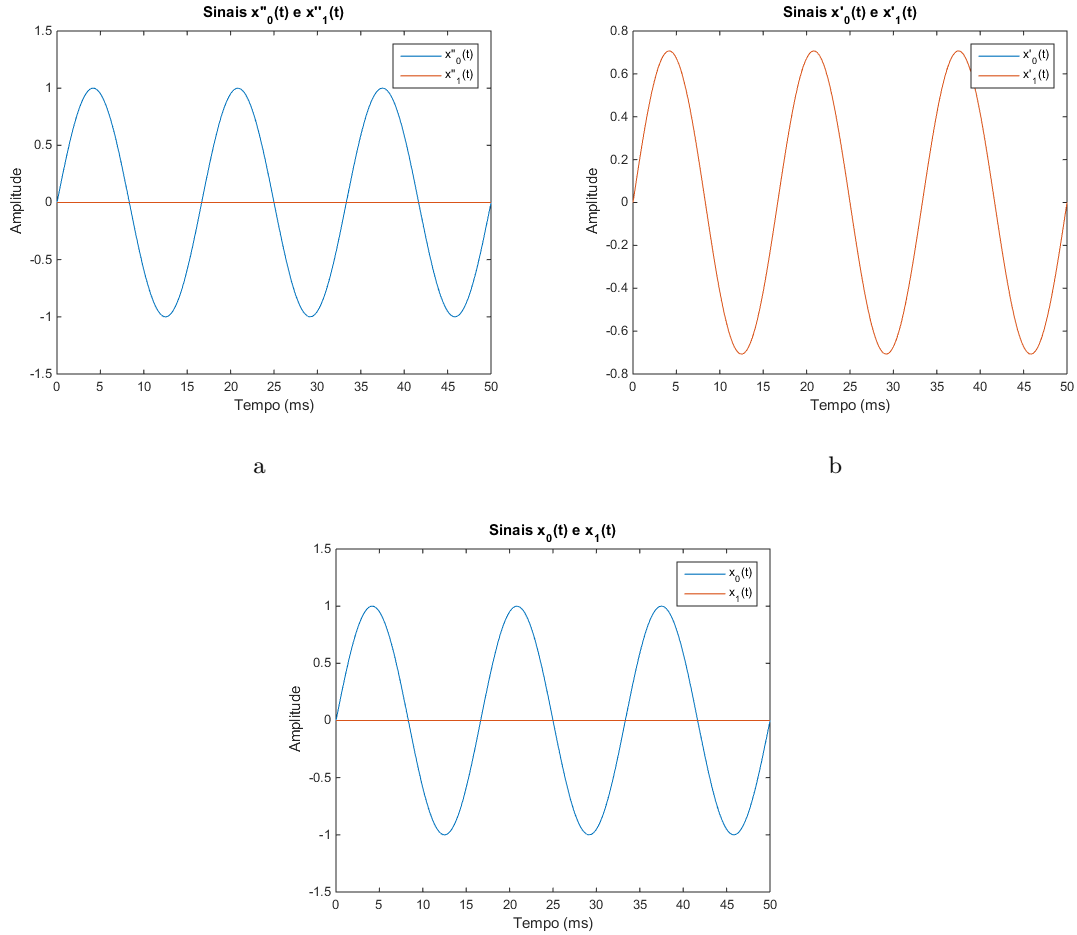


Figura 4.8: Sinais de entrada e saída para duas portas H em série. em a) e c) o sinal de entrada que representa o estado fundamental $|0\rangle$. Em b) o sinal de saída da primeira porta H e entrada da segunda. Neste os sinais $x_0(t)$ e $x_1(t)$ estão sobrepostos.

4.2.2.3 Controled-NOT

A porta Controled-NOT, tal como apresentada em 3.6.3, é uma porta de $N > 2$ qubits, descrita pela matriz na equação 4.18. Uma descrição mais detalhada pode ser encontrada na secção 3.6.3.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.18}$$

Pretende-se agora implementar uma porta CNOT na sua formulação mais genérica. Considera-se as equações 4.19 e 4.19 onde se codifica dois quaisquer qubits $|\psi\rangle_0$ e $|\psi\rangle_1$.

$$\psi^0(t) = \begin{cases} x_0^0(t) = |\alpha_0| cos(wt + \theta_\alpha^0) \\ x_1^0(t) = |\beta_0| cos(wt + \varphi_\beta^0) \end{cases} \tag{4.19}$$

$$\psi^1(t) = \begin{cases} x_0^1(t) = |\alpha_1| cos(wt + \theta_\alpha^1) \\ x_1^1(t) = |\beta_1| cos(wt + \varphi_\beta^1) \end{cases} \tag{4.20}$$

Para implementar a porta CNOT é necessário, em primeiro lugar, obter todos os estados do sistema através do produto tensorial. De acordo com a equação 3.38, definem-se sinais $y(t)$, tal que:

$$\begin{aligned} y^{00}(t) &= x_0^0(t)x_0^1(t) \\ y^{01}(t) &= x_0^0(t)x_1^1(t) \\ y'^{10}(t) &= x_1^0(t)x_0^1(t) \\ y'^{11}(t) &= x_1^0(t)x_1^1(t) \end{aligned} \tag{4.21}$$

O próximo passo é trocar os sinais $y'^{10}(t)$ e $y'^{11}(t)$ entre si, tal que:

$$\begin{aligned} y'^{10}(t) &= y^{11}(t) \\ y'^{11}(t) &= y^{10}(t) \end{aligned} \tag{4.22}$$

Um possível esquemático para a implementação da porta CNOT é apresentado na figura 4.9. Inicialmente são feitas as multiplicações de acordo com a equação 4.21. Por fim os sinais $y'^{10}(t)$ e $y'^{11}(t)$ são trocados e colocados na saída.

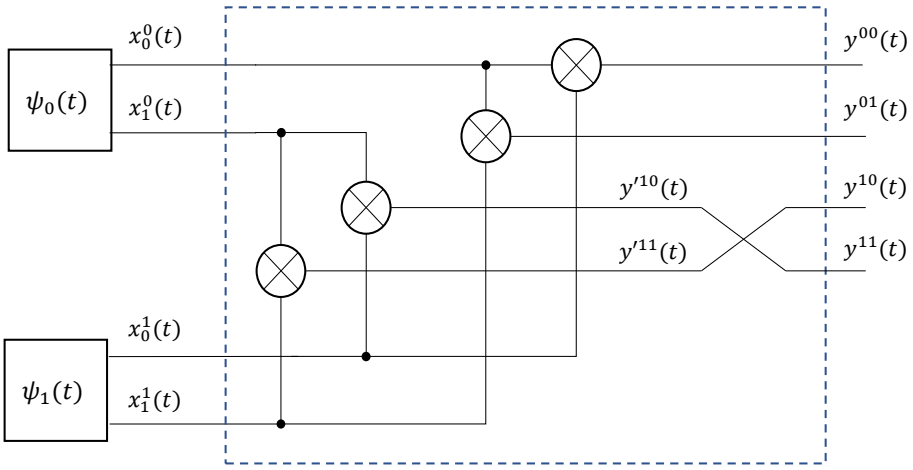


Figura 4.9: Implementação da porta CNOT. Primeiramente os sinais de entrada são multiplicados entre si, de acordo com o produto tensorial. O bloco multiplicador é representado através de uma circunferência com o símbolo 'X' inscrito na mesma. De seguida os sinais $y'^{10}(t)$ e $y'^{11}(t)$ são trocados entre si e colocados à saída.

Através da implementação esquematizada na figura 4.9 é possível descrever todos os estados do sistema. Este esquema permite assim representar estados entrelaçados, como o da equação 3.40. Neste caso os sinais y^{00} e y^{11} teriam uma amplitude de $\frac{1}{\sqrt{2}}$ e os restantes zero. O resultado da simulação, do esquemático da figura I.4, deste caso particular está representado na figura 4.10.

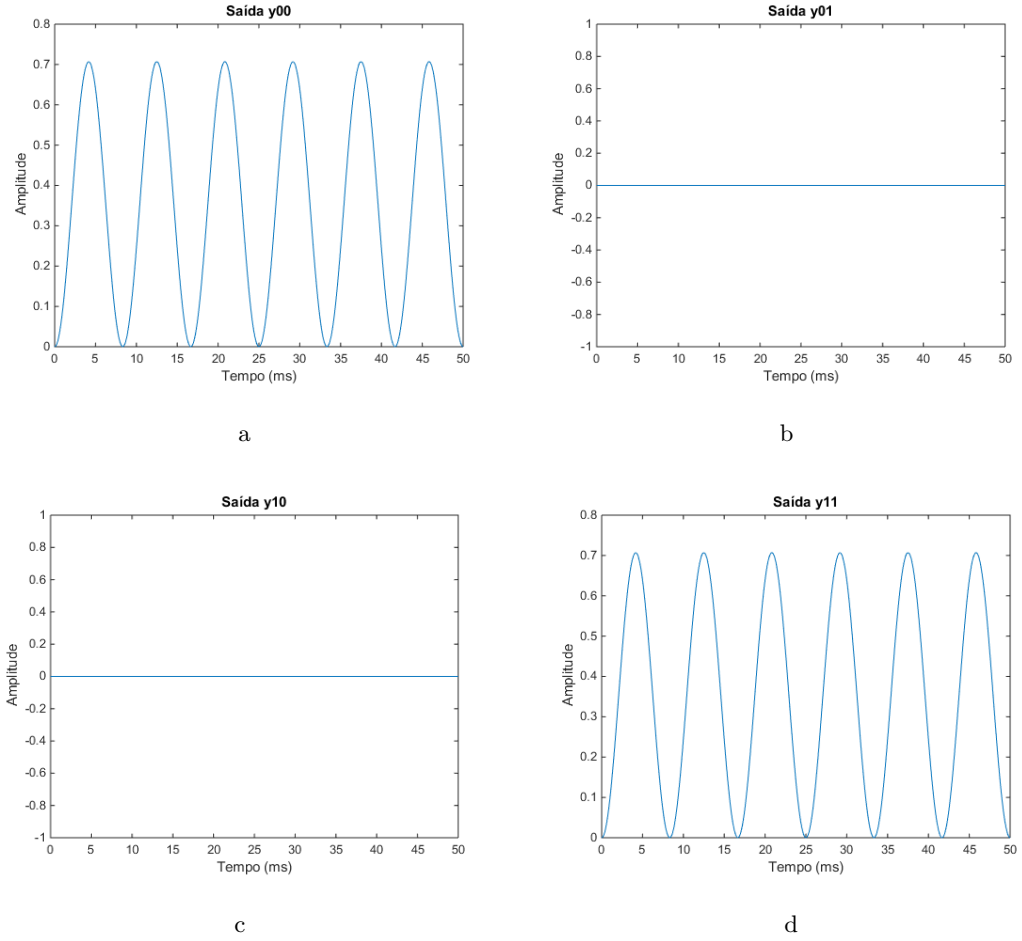


Figura 4.10: Sinais de saída da porta CNOT com o qubit de controlo numa combinação linear e o alvo no estado fundamental $|0\rangle$. Os sinais y^{00}, y^{01}, y^{10} e y^{11} estão representados nas figuras a), b), c) e d), respectivamente.

É necessário, ao contrario do que a computação quântica prevê, um número de saídas que cresce exponencialmente com o número de qubits (N). Para além das saídas, os recursos para implementar a porta também crescem, em número, de forma exponencial.

4.2.3 Detecção

Seguindo a arquitectura apresentada na figura 4.1, o último bloco implementa o módulo para obter o resultado da computação. A implementação deste depende da codificação dos qubits isto é, de que forma é que o sistema físico está implementado. Ao considerar

um sistema analógico como base da computação, introduz-se o trabalho de La Cour, [20], onde apresenta uma forma de detecção com características semelhantes aos quânticos.

O modelo de detector apresentado em [20] considera uma detecção se um dado valor é maior do que um *threshold*. Existem, no entanto, casos onde pode não acontecer detecção ou existir mais do que uma (detecção múltipla). Estes casos não são considerados detecções válidas. A motivação física para este modelo surgiu, segundo [20], na Electrodinâmica e Óptica estocástica.

Segundo [20], considera-se um registo de qubits $|\psi\rangle$ tal que,

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \cdots + \alpha_N |N\rangle, \quad \alpha_0, \ldots, \alpha_N \in \mathbf{C}. \tag{4.23}$$

Represente-se agora todos os números complexos associados a cada estado num único vector complexo, ao qual se chama $\boldsymbol{\alpha}$, tal que

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_0 & \ldots & \alpha_N \end{bmatrix}^T. \tag{4.24}$$

O modelo proposto em [20] considera um vector complexo aleatório $\mathbf{a}$ tal que

$$\mathbf{a} = s\boldsymbol{\alpha} + \mathbf{w} \tag{4.25}$$

onde $\boldsymbol{\alpha}$ representa o sinal normalizado, s a amplitude do sinal ($s \geq 0$). O vector $\mathbf{w} = \begin{bmatrix} w_0 & \ldots & w_N \end{bmatrix}^T$ representa um vector de ruído aleatório, onde cada componente w_i é uma variavél aleatória com distribuição gaussiana normalizada, independente e identicamente distribuída.

Dado um valor de *threshold* Γ diz-se que ocorreu uma detecção se, e só se,

$$|a_n| \geq \Gamma, \quad |a_{n'}| \leq \Gamma$$

para qualquer valor de $n \neq n'$. De acordo com [20] ocorrem os seguintes três casos.

A probabilidade de detectar um único componente de n acima do *threshold* é dada por

$$P_n(\boldsymbol{\alpha}, \Gamma) = P[|a_n| \geq \Gamma, \quad |a_{n'}| \leq \Gamma \quad \forall n \neq n']. \tag{4.26}$$

A probabilidade de nenhum dos valores ser superior ao *threshold*, representado por $P_0(\boldsymbol{\alpha}, \Gamma)$, é

$$P_0(\boldsymbol{\alpha}, \Gamma) = P[|a_n| \leq \Gamma, \quad \ldots, \quad |a_N| \leq \Gamma]. \tag{4.27}$$

Por fim, como último caso, a probabilidade de existir mais do que uma detecção é dada por $P_\infty(\boldsymbol{\alpha}, \gamma)$ tal que

$$P_\infty(\boldsymbol{\alpha}, \Gamma) = 1 - P_0(\boldsymbol{\alpha}, \Gamma) - \sum_{n=1}^{N} P_n(\boldsymbol{\alpha}, \Gamma). \tag{4.28}$$

Além disso, é também possível reproduzir várias características quânticas, como o entrelaçamento ou verificar a violação da desigualdade de Bell [20].

Embora a abordagem apresentada seja eficiente, fica em falta grande parte das configurações de estados possíveis para os qubits, uma vez que este sistema só funciona em estados ditos *standard* e os restantes são tratados como aproximações [20]. É assim necessário mais investigação para se obterem modelos mais robustos e capazes.

Capítulo 5

Conclusões

Neste capítulo são discutidos alguns resultados, considerações finais e trabalhos futuros. Na secção 5.1 são apresentadas as conclusões gerais do trabalho. São revistos alguns pontos essenciais e discutidos os resultados mais fundamentais. Na secção 5.2 são apresentadas algumas linhas de investigação para seguir, com base no trabalho desenvolvido durante a dissertação.

5.1 Considerações Finais

Nos últimos anos tem-se verificado um aumento significativo do investimento na computação quântica. Grandes empresas, como a Google ou a IBM, tem apostado no desenvolvimento desta, na esperança de encontrar o ouro da supremacia quântica. Paralelamente a estes desenvolvimentos, alguns grupos de investigação tem encontrado algumas alternativas à computação quântica. Estas tendem em ser mais baratas e acessíveis através do estado da arte tecnológico actual. Uma tecnologia baseada em, por exemplo, sistemas analógicos funciona à temperatura ambiente, em tempos de coerência elevados (horas, dias) e não sofre restrições provenientes da delicadeza da microfísica.

O modelo considerado nesta dissertação mostrou que, com base em sistemas analógicos, é possível implementar uma arquitectura de forma a emular um sistema microfísico. Partindo da codificação dos qubits em sinais analógicos, foi possivel mostrar o funcionamento, implementação e limitações das portas lógicas X, Hadamard e CNOT. Considerou-se, no entanto, um sistema ideal sem ruído. As duas primeiras portas são implementadas sem crescimento exponencial de recursos. No entanto, as portas de N-qubits, com $N \geq 2$, vão, na sua descrição mais genérica, necessitar de um número de recursos que cresce exponencialmente (2^N). Este crescimento pode ser controlado, considerando que na entrada apenas vão estar configurações de qubits pré-definidas.

Um outro desafio, tal como referido nas linhas finais do capitulo 4, está na detecção. Isto é, na forma como se transformam os valores de probabilísticos em resultados de detecção. A abordagem apresentada, baseada em métodos de telecomunicações clássicos, tem algumas limitações uma vez que só detecta, com baixo erro, certas configurações probabilísticas dos qubits. É pois necessário investigar estes pontos considerando talvez outras formas de onda e outras técnicas de processamento de sinal.

Conseguíu-se assim mostrar que é possivel implementar um sistema analógico com base no formalismo matemático proposto. Este resultado abre portas para uma nova linha de pensamento e investigação, onde se prevê atingir mais e melhor resultados computacionais.

5.2 Trabalho Futuro

O modelo apresentado pode ser melhorado em vários aspectos. A forma de onda que codifica o qubit deverá ser repensada, por forma a melhorar o seu desempenho face à introdução de ruído. Posteriormente poderá ser melhorada a implementação física da mesma, diminuindo o número de recursos físicos necessários. Na detecção será interessante procurar desenvolver um outro modelo mais abranjente.

Após estas melhorias, seguidamente será a altura de estudar uma forma de implementar e validar os algoritmos quânticos já existentes, como o algoritmo de Groover ou Shor, segundo este modelo. Este passo será um dos mais importantes para aplicar o funcionamento do modelo. O grande objectivo final é atingir uma aceleração exponencial da computação sem fazer crescer o número de recursos necessários de uma forma exponencial.

Bibliografia

[1] M. A. Nielsen e I. Chuang. *Quantum computation and quantum information*. 2002.

[2] R. P. Feynman. *QED: The strange theory of light and matter*. 2006.

[3] W. K. Wootters e W. H. Zurek. *A single quantum cannot be cloned*. 1982.

[4] J. M. Gambetta, J. M. Chow e M. Steffen. *Building logical qubits in a superconducting quantum computing system*. 2017.

[5] R. P. Feynman. *Simulating physics with computers*. 1982.

[6] J. M. Gambetta, J. M. Chow e M. Steffen. *Building logical qubits in a superconducting quantum computing system*. 2017.

[7] L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood e I. L. Chuang. *Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance*. 2001.

[8] L. B. Kish. *Quantum computing with analog circuits: Hilbert space computing*. International Society for Optics e Photonics, 2003.

[9] V. Kreinovich, L. Kohout e E. Kim. *Square root of 'not': a major difference between fuzzy and quantum logics*. 2011.

[10] B. R. La Cour e G. E. Ott. *Signal-based classical emulation of a universal quantum computer*. 2015.

[11] C. Dyson. *Implementing quantum algorithms using classical electrical circuits: Deutsch, Deutsch-Jozsa and Grover*. 2011.

[12] S. Roman. *Advanced linear algebra*. 2008.

[13] P. A. M. Dirac. *The principles of quantum mechanics*. 1981.

[14] W.-H. Steeb e T. K. Shi. *Matrix calculus and Kronecker product with applications and C++ programs*. 1997.

[15] M. Sipser. *Introduction to the Theory of Computation*. 2006.

[16] D. I. Cohen. *Introduction to computer theory*. 1991.

[17] S. Aaronson. *Quantum computing since Democritus*. 2013.

[18] C. P. Williams. *Explorations in quantum computing*. 2010.

[19] N. S. Yanofsky, M. A. Mannucci e M. A. Mannucci. *Quantum computing for computer scientists.* 2008.

[20] B. R. La Cour. *A locally deterministic, detector-based model of quantum measurement.* 2014.

ANEXO I

Anexo 1 Simulink

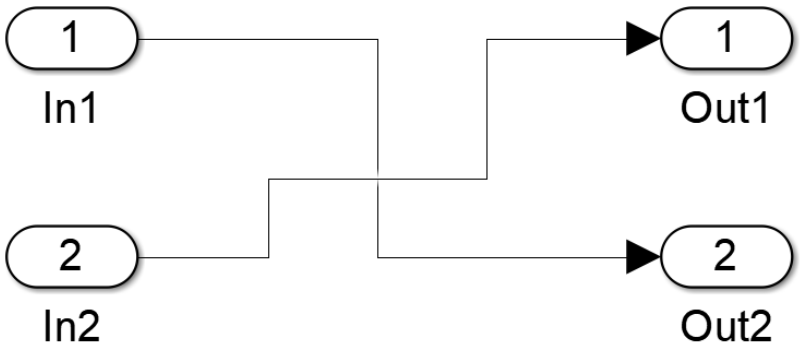


Figura I.1: Esquemático simulink da porta NOT.

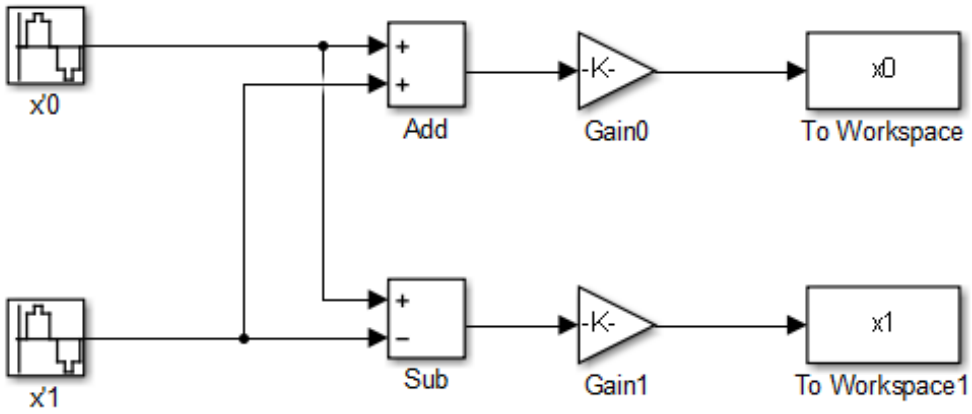


Figura I.2: Esquemático simulink da porta H. O ganho é dado por $k = 1/\sqrt{2}$.

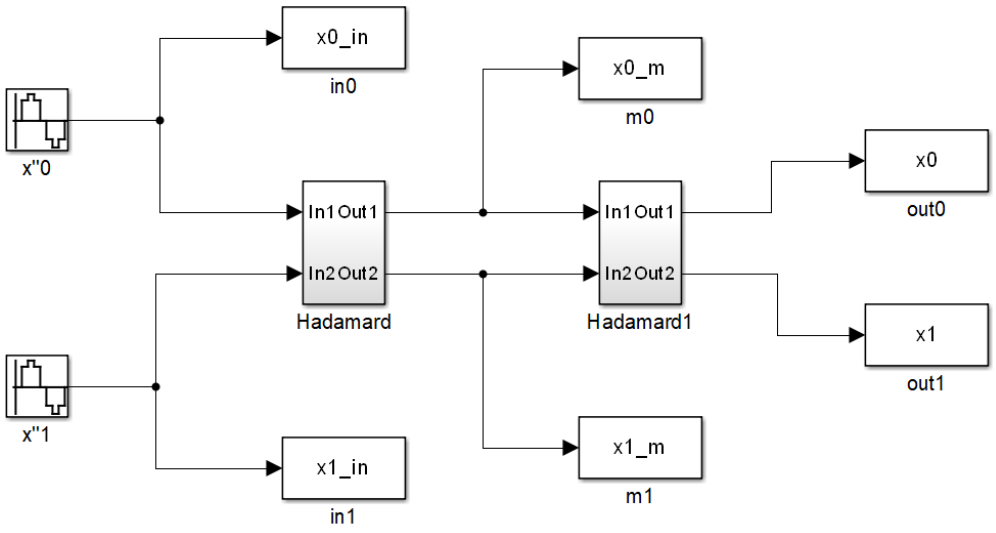


Figura I.3: Esquemático simulink de duas portas H em série.Os blocos hadamard e hadamard1 representam duas portas H. O esquemático destes blocos encontra-se na figura I.2.

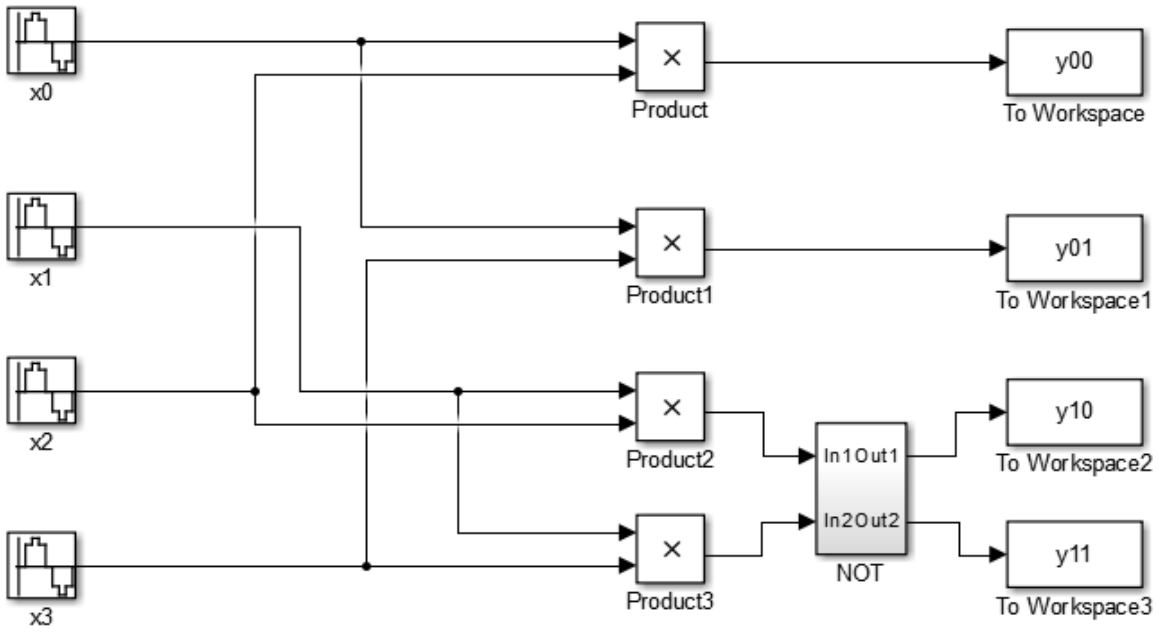


Figura I.4: Esquemático simulink da porta lógica CNOT.