



BRNO UNIVERSITY OF TECHNOLOGY  
VYSOKÉ UČENÍ TECHNICKÉ



FACULTY OF MECHANICAL ENGINEERING  
FAKULTA STROJNÍHO INŽENÝRSTVÍ  
INSTITUTE OF MATHEMATICS  
ÚSTAV MATEMATIKY

# APPLICATION OF GEOMETRIC ALGEBRAS IN QUANTUM COMPUTING

APLIKACE GEOMETRICKÝCH ALGEBER V KVANTOVÉM POČÍTÁNÍ

BACHELOR'S THESIS  
BAKALÁŘSKÁ PRÁCE

AUTHOR  
AUTOR

JAN MICHÁLEK

SUPERVISOR  
VEDOUCÍ PRÁCE

doc. Mgr. PETR VAŠÍK, Ph.D.

BRNO 2023



# Assignment Bachelor's Thesis

Institut: Institute of Mathematics  
Student: **Jan Michálek**  
Degree program: Mathematical Engineering  
Branch: no specialisation  
Supervisor: **doc. Mgr. Petr Vašík, Ph.D.**  
Academic year: 2022/23

As provided for by the Act No. 111/98 Coll. on higher education institutions and the BUT Study and Examination Regulations, the director of the Institute hereby assigns the following topic of Bachelor's Thesis:

## Application of Geometric Algebras in Quantum Computing

### Brief Description:

When designing the algorithms for quantum computing, it is appropriate to use the notions of geometric algebra. It turns out that it is also effective. The construction of elementary logic operators (gates) is then rather straightforward, especially if a complex Clifford algebra is used.

### Bachelor's Thesis goals:

- The student should acquire the knowledge of quantum computing fundamental notions.
- A simple circuit for quantum computation should be designed.
- Complex (or real) geometric algebras should be applied.

### Recommended bibliography:

DE LIMA MARQUEZINO, F., et al. A Primer on Quantum Computing, SpringerBriefs in Computer Science, 2019.

RUE, J., XAMBO, S. Mathematical essentials of quantum computing, Lecture notes UPC, <https://web.mat.upc.edu/sebastia.xambo/QC/qc.pdf>.

Deadline for submission Bachelor's Thesis is given by the Schedule of the Academic year 2022/23

In Brno,

L. S.

---

doc. Mgr. Petr Vašík, Ph.D.  
Director of the Institute

---

doc. Ing. Jiří Hlinka, Ph.D.  
FME dean

## **Abstract**

This thesis explores the use of Geometric algebras in Quantum computing. It begins by defining the general Clifford algebra and then derives a specific Complex Geometric algebra that is well-suited for representing quantum computing systems. This approach is compared to the traditional method of using a classical matrix representation. By analyzing and comparing these two methods, the thesis aims to provide insights into the potential advantages of using geometric algebras for quantum computing applications.

## **Abstrakt**

Tato práce se zabývá využitím geometrických algeber v oblasti kvantového počítání. Nejprve je definována obecná Cliffordova algebra a následně je odvozena specifická komplexní geometrická algebra, která je vhodná pro reprezentaci kvantových výpočtů. Tento přístup je porovnán s tradiční metodou použití klasické maticové reprezentace. Cílem práce je poskytnout poznatky o potenciálních výhodách použití geometrických algeber pro kvantové výpočty.

## **Keywords**

Clifford algebra, Geometric algebra, Quantum computing

## **Klíčová slova**

Cliffordova algebra, Geometrická algebra, Kvantové počítání

## Rozšířený abstrakt

Tato práce popisuje možné využití geometrických algeber (GA) v problematice kvantového počítání (QC). Je popsáno odvození reprezentace kvantového počítání v GA a následně je tento přístup porovnán s klasickou maticovou reprezentací, která je v současnosti používána.

V první kapitole jsou definovány klíčové pojmy z algebry a jsou zasazeny do kontextu. Zvlášť pozornost je věnována bilineárním a kvadratickým formám a také jednotlivým algebraickým produktům. Je uvedena obecná definice tenzorového součinu a tenzorových prostorů, a následně je ukázána na příkladech. Poté je odvozena Cliffordova algebra. Tato algebra má několik důležitých vlastností, které jsou diskutovány.

Druhá kapitola se věnuje kvantové fyzice, neboť je důležité správně uchopit a popsat koncept qubitu. Je ukázán matematický popis elektromagnetické vlny. Vlna je reprezentována pomocí vektoru napsaného v bázi, která je tvořena složkou elektrického pole a magnetického pole. Následně je uvedena rovnice pro výpočet energie vlny a je popsáno, jak rozložit vlnu do jejích složek a jak měřit množství energie v těchto složkách. V krátkosti je popsán rozpor mezi klasickou vlnovou teorií a pohledem kvantové mechaniky.

Všechny tyto informace jsou využity pro definici qubitu. Význam qubitu a vztah s klasickým bitem je rozebrán. Qubit jako takový si není jednoduché představit, už jen vzhledem k tomu, kolik má stupňů volnosti. Pomocí druhého axiomu pravděpodobnosti a zanedbáním globální fáze qubitu je však možné snížit počet stupňů volnosti na dva. Definice qubitu je následně převedena do sférických souřadnic. Pro účel vizualizace je zavedena tzv. Blochova sféra. Na této sféře se dají jednoduše vizualizovat konkrétní qubity i jejich transformace. Pomocí Kroneckerova součinu je definice qubitu rozšířena na případ multi qubitů. V jednoduchosti je následně ukázáno, jak provést měření qubitu či multi qubitů.

Další kapitola ukazuje jak odvodit obecnou kvantovou bránu a jaké vlastnosti musí splňovat. Definovány jsou základní brány včetně demonstrace na příkladech. Pro lepší ilustraci fungování kvantových bran je použit software Qiskit. Pomocí Kroneckerova součinu je definice kvantové brány pro jeden qubit rozšířena na bránu pro více qubitů. Výpočty na několika paralelních branách jsou popsány tenzorovým součinem. Na konci kapitoly je vytvořen a popsán jednoduchý kvantový obvod se všemi náležitostmi.

V hlavní část práce jsou využity veškeré získané poznatky pro konstrukci aparátu geometrické algebry (GA) vhodného pro kvantové výpočty. Začínáme definicí GA nad reálnými čísly. Poté je detailně popsán geometrický součin, kterým je tato algebra vybavena. Jelikož tento součin hraje klíčovou roli po celou práci je pečlivě odvozen z vnitřního a vnějšího součinu a také jsou zdůrazněny jeho důležité vlastnosti. Pro správnou reprezentaci qubitů v GA je potřeba vnést do naší definice komplexní čísla. Pole nad kterým je algebra zkonstruována je změněno na komplexní čísla, což ovšem není dostatečné, protože komplexní strukturu je třeba dostat přímo do algebry. Proto je zavedena ortogonální lineární transformace. Báze algebry je následně zapsaná s využitím této transformace. Takto definovaná báze se nazývá tzv. Wittova báze a má několik vlastností, které jsou popsány. Tyto vlastnosti jsou poté hojně využívány při zjednodušování výpočtů.

S vhodně zdefinovaným aparatem GA lze přejít k definici qubitu a multi qubitů v GA. Klíčovou myšlenkou je identifikovat jak qubity tak kvantové brány s prvky v GA. To nám umožňuje provádět výpočty v rámci jedné algebraické struktury. Detailně je ukázán vztah mezi maticovou a GA reprezentací a je popsáno, jak se dostat z jedné reprezentace do druhé. S pomocí tenzorového součinu jsme ukázali, jak provádět výpočet na několika paralelních branách. Vše v této kapitole je doprovázeno příklady pro intuitivní pochopení výpočtů v GA. Ve stručnosti je také popsáno jak v GA provádět měření. Zde je využita skalární projekce.

V závěrečné části je ukázáno, jak zapsat jednoduchý obvod v maticové a v GA reprezentaci. První příklad ukazuje kvantové brány v sériovém pořadí. Druhý příklad ukazuje kvantové brány v paralelním nastavení. Správnost GA reprezentace je potvrzena porovnáním výsledků měření s

výsledky měření získanými v maticové reprezentaci. Výsledky jsou vždy stejné a pouze potvrzují správnost GA reprezentace.

Povedlo se nám vytvořit aparát GA, který umožňuje intuitivní a jednoduché kvantové výpočty. Elegantní definice báze umožňuje mnoho zjednodušení, což výpočty zrychluje a potenciálně také může snížit nároky na paměť. Problém maticové reprezentace je zejména s nárůstem dimenze matic. Tento nárůst je exponenciální a i pro malý počet vstupních qubitů se výpočty mohou stát náročné. Dimenze výrazů reprezentujících kvantové brány v GA sice také roste, ale nárůst není tak dramatický a výraz může být často zjednodušen. Největším problémem GA přístupu je jeho implementace. Komplexní čísla jsou obtížná na reprezentaci a v současnosti neexistuje vhodný software pro výpočty v GA. GAALOPWeb nabízí alternativu, při které používá mírně odlišný přístup a pracuje s GA nad reálnými čísly.





I, hereby, declare that I wrote this bachelor's thesis, *Application of Geometric algebras in quantum computing*, all by myself under the direction of my supervisor, doc. Mgr. Petr Vašík, Ph.D. All used sources are listed in references.

Jan Michálek



At this point, I would like to express my sincere gratitude to my supervisor for his invaluable support, guidance and expertise provided. It helped me navigating through all the challenges along the way. I would also like to thank my family for never ending encouragement in my pursuit.

Jan Michálek



# Contents

<b>1</b>	<b>Introduction</b>	<b>12</b>
<b>2</b>	<b>Underlying Algebra</b>	<b>13</b>
2.1	Forms . . . . .	13
2.2	Algebraic products . . . . .	14
2.3	Tensor spaces . . . . .	15
2.4	Tensor algebra . . . . .	16
2.5	Clifford algebra . . . . .	17
2.6	Exterior algebra . . . . .	18
2.7	Bra-Ket notation . . . . .	19
<b>3</b>	<b>Quantum Physics</b>	<b>19</b>
3.1	Electromagnetic waves . . . . .	20
3.2	Mathematical description of electromagnetic waves . . . . .	20
3.3	Energy of the wave . . . . .	21
3.4	Classical versus quantum understanding . . . . .	22
<b>4</b>	<b>Quantum computing</b>	<b>22</b>
4.1	Qubit . . . . .	23
4.2	Bloch sphere . . . . .	24
4.3	Multiple Qubits . . . . .	26
4.4	Measurement . . . . .	26
<b>5</b>	<b>Quantum gates</b>	<b>27</b>
5.1	Single qubit gates . . . . .	27
5.1.1	NOT gates . . . . .	28
5.1.2	Hadamard gate . . . . .	29
5.1.3	Phase gate . . . . .	30
5.1.4	Gates tensor product . . . . .	31
5.2	Multiple qubit gates . . . . .	31
5.2.1	CNOT gate . . . . .	31
5.2.2	SWAP gate . . . . .	32
5.2.3	CZ gate . . . . .	32
5.2.4	Toffoli gate . . . . .	32
5.3	Quantum circuits . . . . .	33
<b>6</b>	<b>Geometric algebras in Quantum computing</b>	<b>34</b>
6.1	Geometric algebra . . . . .	34
6.2	Complexification . . . . .	35
6.3	Qubit in GA . . . . .	36
6.4	Gates in GA . . . . .	37
6.5	Multiple qubits and gates in GA . . . . .	39
6.6	Parallel gates . . . . .	40
6.7	Measurement . . . . .	41
6.8	Circuit example . . . . .	41
<b>7</b>	<b>Conclusion</b>	<b>48</b>

# 1 Introduction

Quantum computing is a relatively new, although very dynamic field in a computer science. Solid mathematical background is essential for further development. Currently the most common way to represent quantum states and operations with them is via Dirac's formalism. It is an elegant tool, which helps us handle all the complications quantum computing is bringing to us. It leads to a matrix representation, where the qubits are seen as vectors and quantum gates as matrices. However, the dimension of the matrices rises with higher number of input qubits and it becomes computationally more demanding.

This thesis will try to provide an alternative. It will be shown that the apparatus of geometric algebras can be equally suitable for quantum computing and even more powerful in some particular applications. General Clifford algebra will be introduced and a special Geometric algebra will be derived. Quantum states will be then associated with elements of the Geometric algebra. Quantum gates are also associated with elements of this algebra and it allows for a direct application within one algebraic structure. This approach will be compared with matrix representation on simple examples.

## 2 Underlying Algebra

This chapter provides a comprehensive algebraic background that defines key concepts necessary for understanding future material and places them into the context.

This thesis focuses on geometric algebras and in order to properly introduce them, it is essential to define other related concepts such as vector spaces and tensor spaces. Although many of these algebraic concepts may be intuitively understood, they can be abstract and challenging to define. This thesis assumes that the reader has a fundamental understanding of the following concepts: vectors, vector spaces, basis, linear independence, groups, commutative groups, rings or different types of morphism. Other concepts will be defined in detail to ensure a complete understanding of the subject matter.

### 2.1 Forms

An important tool in algebra is called the forms. We will be especially interested in bilinear and quadratic forms.

**Definition 2.1** (Bilinear form). Let  $V$  be a vector space over the ground field  $K$ . Then the map  $B : V \times V \rightarrow K$  is called a *bilinear form* if the linearity in both arguments is satisfied, meaning

$$\begin{aligned} B(\mathbf{u} + \mathbf{v}, \mathbf{w}) &= B(\mathbf{u}, \mathbf{w}) + B(\mathbf{v}, \mathbf{w}) & \text{and} & & B(\alpha\mathbf{u}, \mathbf{w}) &= \alpha B(\mathbf{u}, \mathbf{w}) \\ B(\mathbf{u}, \mathbf{v} + \mathbf{w}) &= B(\mathbf{u}, \mathbf{v}) + B(\mathbf{u}, \mathbf{w}) & \text{and} & & B(\mathbf{u}, \alpha\mathbf{w}) &= \alpha B(\mathbf{u}, \mathbf{w}) \end{aligned}$$

for all  $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$  and all  $\alpha \in K$ .

Once a basis of  $V$  is chosen, every bilinear form is associated with a matrix  $M$  and the form can be then written as

$$B(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T M \mathbf{v}.$$

**Definition 2.2** (Quadratic form). Let  $V$  be a vector space over the ground field  $K$ . Then the map  $Q : V \rightarrow K$  is called a *quadratic form* with associated symmetric matrix  $A$ . Quadratic form of  $\mathbf{u} \in V$  can be then written as

$$Q(\mathbf{u}) = \mathbf{u}^T A \mathbf{u}.$$

Every quadratic form has its associated bilinear form, which is defined as

$$B(\mathbf{u}, \mathbf{v}) = \frac{1}{2}(Q(\mathbf{u} + \mathbf{v}) - Q(\mathbf{u}) - Q(\mathbf{v})). \quad (2.1)$$

Quadratic forms can be seen as a special case of bilinear forms. When the matrix associated to the bilinear form is symmetric, than this bilinear form defines a quadratic form

$$B(\mathbf{u}, \mathbf{u}) = Q(\mathbf{u}). \quad (2.2)$$

**Definition 2.3** (Orthogonal matrix). Matrix  $U$  is called orthogonal if  $U^T U = U U^T = I$ , where  $I$  is the identity matrix. In other words can be said that,  $U^T = U^{-1}$ .

**Definition 2.4** (Diagonal matrix). Square matrix

$$D = \begin{pmatrix} d_{11} & \dots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} & \dots & d_{nn} \end{pmatrix}$$

is called *diagonal* if all the elements except, of the elements on the main diagonal, are zero, ie.,  $\forall i, j \in \{1, 2, \dots, n\}, i \neq j \Rightarrow d_{ij} = 0$ .

**Definition 2.5** (Orthogonally diagonalizable matrix). Matrix  $A$  is called *orthogonally diagonalizable* if there exists an orthogonal matrix  $U$  and diagonal matrix  $D$  such that  $A = UDU^T$ .

**Theorem 2.1.** Every symmetric matrix is orthogonally diagonalizable. The inverse is also true, every orthogonally diagonalizable matrix is symmetric since the transpose of the diagonal matrix is the matrix itself. Then can be written  $(UDU^T)^T = (U^T)^T D^T U^T = UDU^T$ .

## 2.2 Algebraic products

Throughout this thesis, we will encounter several types of algebraic products. Therefore it is important to distinguish properly between them. These products are usually just specific types of the forms, that were defined previously.

**Definition 2.6** (Inner product). Let us have a ground field  $K$  and a vector space  $V$  over this field. Then by *inner product* we understand the map:

$$\langle \cdot, \cdot \rangle : V \times V \rightarrow K$$

which satisfies the following axioms:

(A1) *Conjugate symmetry*:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \overline{\langle \mathbf{v}, \mathbf{u} \rangle}$$

(A2) *Linearity in the first argument*:

$$\begin{aligned} \langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle &= \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle \\ \langle \alpha \mathbf{u}, \mathbf{w} \rangle &= \alpha \langle \mathbf{u}, \mathbf{w} \rangle \end{aligned}$$

(A3) *Positive-definiteness*: if  $\mathbf{u} \neq 0$  than  $\langle \mathbf{u}, \mathbf{u} \rangle > 0$

for all vectors  $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$  and all scalars  $\alpha \in K$ .

The axiom (A3) is not always required and so it will be in this thesis. When talking about inner product, it will be required to hold just the axioms (A1) and (A2).

Notice that the inner product is actually a bilinear form. It takes two vectors and maps them to an element in a ground field. Elements in the ground field are usually called scalars. That is where the name *scalar product* comes from. And although these terms are very similar, they are not to be confused. The *scalar product* is just a special case of inner product.

Next important product to be defined is the *outer product*. As the name *outer product* suggests it creates something out of the original space. It takes two vectors as an input and creates a new object with a higher dimension. This object can be seen as a *tensor*. We will define tensors and tensor products in more details later. For now it will be sufficient to understand the outer product as a special case of the tensor product. They are even denoted in the same way.

**Definition 2.7** (Outer product). Let us have a ground field  $K$  and a vector space  $V$  over this field. Vectors  $\mathbf{u}, \mathbf{v} \in V$  have a form  $\mathbf{u} = (u_1 \ u_2 \ \dots \ u_m)$ ,  $\mathbf{v} = (v_1 \ v_2 \ \dots \ v_n)^T$ . Then their *outer product*,  $\mathbf{u} \otimes \mathbf{v}$ , is a matrix  $A$  of the following form:

$$\mathbf{u} \otimes \mathbf{v} = A = \begin{pmatrix} u_1 v_1 & \dots & u_1 v_n \\ \vdots & \ddots & \vdots \\ u_m v_1 & \dots & u_m v_n \end{pmatrix}.$$

The *outer product* should not be confused with an *exterior product*. The *exterior product* or a *wedge product* will be a crucial element in the study of Geometric algebras, because it enables us to work with planes, volumes and higher-dimensional analogues. This product will be defined later together with the exterior algebra.



## 2.3 Tensor spaces

We have already encountered tensors, when speaking about the outer product. Tensors are abstract objects that can be defined in several ways. We will first introduce the universal property, which gives a constraint, that all tensor products must satisfy. However, this definition is not constructive. Therefore, we will also provide an alternative definition of tensor and show some examples.

**Definition 2.8** (Universal property). Tensor product of two vector spaces  $V$  and  $W$ , which are over the same field  $K$ , is a vector space  $V \otimes W$  together with bilinear map  $\otimes : (\mathbf{v}, \mathbf{w}) \mapsto \mathbf{v} \otimes \mathbf{w}$ ,  $\otimes : V \times W \rightarrow V \otimes W$ , so that for every bilinear map  $h : V \times W \rightarrow Z$  exists unique linear map  $\bar{h} : V \otimes W \rightarrow Z$ , which satisfies  $h = \bar{h} \circ \otimes$  (the following diagram commutes).

$$\begin{array}{ccc} V \times W & \xrightarrow{\otimes} & V \otimes W \\ & \searrow h & \downarrow \bar{h} \\ & & Z \end{array}$$

The universal property should serve us as a check whether a particular construction of a tensor product is valid. As long as the universal property holds, the constructed tensor product is correct.

**Definition 2.9** (Tensor product). Let us have two vector spaces  $V$  and  $W$  over the same field  $K$ . Tensor product  $V \otimes W$  is a vector space together with associated bilinear map  $V \times W \rightarrow V \otimes W$  which takes elements  $\mathbf{v} \in V$  and  $\mathbf{w} \in W$  and maps them to an element of the vector space  $V \otimes W$ ,  $(\mathbf{v}, \mathbf{w}) \mapsto \mathbf{v} \otimes \mathbf{w}$ .

**Example 2.1.** Let us have two vector spaces  $V$  and  $W$  with basis  $\mathbf{B}_V = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ ,  $\mathbf{B}_W = \{\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2\}$  and two vectors  $\mathbf{x} \in V, \mathbf{y} \in W$

$$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} 4 \\ 5 \end{pmatrix} \quad \mathbf{x} \otimes \mathbf{y} = \begin{pmatrix} 1 * 4 \\ 1 * 5 \\ 2 * 4 \\ 2 * 5 \\ 3 * 4 \\ 3 * 5 \end{pmatrix} = \begin{pmatrix} 4 \\ 5 \\ 8 \\ 10 \\ 12 \\ 15 \end{pmatrix},$$

where the  $*$  symbol is just an element multiplication.

The element  $\mathbf{x} \otimes \mathbf{y}$  is an element of a tensor space, a *tensor*. The basis of the tensor space is  $\mathbf{B}_{V \otimes W} = \{\mathbf{e}_1 \otimes \bar{\mathbf{e}}_1, \mathbf{e}_1 \otimes \bar{\mathbf{e}}_2, \mathbf{e}_2 \otimes \bar{\mathbf{e}}_1, \mathbf{e}_2 \otimes \bar{\mathbf{e}}_2, \mathbf{e}_3 \otimes \bar{\mathbf{e}}_1, \mathbf{e}_3 \otimes \bar{\mathbf{e}}_2\}$ . Now let us express this element in basis.

$$\mathbf{x} \otimes \mathbf{y} = 4\mathbf{e}_1 \otimes \bar{\mathbf{e}}_1 + 5\mathbf{e}_1 \otimes \bar{\mathbf{e}}_2 + 8\mathbf{e}_2 \otimes \bar{\mathbf{e}}_1 + 10\mathbf{e}_2 \otimes \bar{\mathbf{e}}_2 + 12\mathbf{e}_3 \otimes \bar{\mathbf{e}}_1 + 15\mathbf{e}_3 \otimes \bar{\mathbf{e}}_2$$

However, this is not the only possible construction of the tensor product. Let us take a look at another example with the same input vectors.

**Example 2.2.**

$$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} 4 \\ 5 \end{pmatrix} \quad \mathbf{x} \otimes \mathbf{y} = \begin{pmatrix} 1 * 4 & 1 * 5 \\ 2 * 4 & 2 * 5 \\ 3 * 4 & 3 * 5 \end{pmatrix} = \begin{pmatrix} 4 & 5 \\ 8 & 10 \\ 12 & 15 \end{pmatrix}$$

This particular tensor product is nothing else than the outer product defined previously, see Definition 2.7. The general definition of the tensor product does not give a constructive definition and we can see on the examples, that many different products can be constructed up to an isomorphism.

It is also worth noticing, that in both cases the dimension of vector  $\mathbf{x}$  and  $\mathbf{y}$  is 2 and 3, respectively, and their tensor product has dimension 6. It is not a coincidence, tensor space truly has the dimension, which is obtained by multiplying dimension of two original vector spaces.

**Definition 2.10** (Tensor space dimension). let us have two vector spaces  $V$  and  $W$  with  $\dim V = n$  and  $\dim W = m$ . Then the tensor space  $V \otimes W$  has a dimension  $\dim V \otimes W = m * n$ .

One particularly interesting example of a tensor product is a *Kronecker product*, sometimes called a *direct sum*. It is widely used in quantum computing. The Kronecker product takes two matrices as an input and produces another matrix with a higher dimension as an output.

**Definition 2.11** (Kronecker product). Take two matrices  $A, B$  of the form

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & \dots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \dots & b_{pq} \end{pmatrix}.$$

Then their Kronecker product, denoted  $C = A \otimes B$  is in the form

$$C = A \otimes B = \begin{pmatrix} a_{11} * B & \dots & a_{1n} * B \\ \vdots & \ddots & \vdots \\ a_{m1} * B & \dots & a_{mn} * B \end{pmatrix}. \quad (2.3)$$

## 2.4 Tensor algebra

Since we have introduced tensor spaces, we can now construct a tensor algebra.

**Definition 2.12** (Tensor power). Let  $V$  be a vector space over a field  $K$ . For every  $k \in \mathbb{N}$  is defined the  $k$ -th tensor power of  $V$  to be the tensor product of  $V$  with itself  $k$ -times:

$$T^k V = V^{\otimes k} = V \otimes V \otimes \dots \otimes V. \quad (2.4)$$

**Definition 2.13** (Tensor algebra). Tensor algebra  $T(V)$  is constructed as a direct sum of  $T^k V$  for  $k = 0, 1, 2, \dots$

$$T(V) = \bigoplus_{k=0}^{\infty} T^k V = K \oplus V \oplus (V \otimes V) \oplus (V \otimes V \otimes V) \dots \quad (2.5)$$

By  $T^0 V$  is understood the ground field  $K$ .

**Definition 2.14** (Graded algebra). An algebra  $A$  over a field  $K$  is said to be graded if it can be written as a direct sum  $A = \bigoplus_{k=0}^{\infty} A^k$  of vector spaces over  $K$  such that the multiplication map satisfies  $A^k \times A^l \rightarrow A^{k+l}$ .

From the definition of tensor algebra can be clearly seen, that this algebra is graded. For example, a scalar ( $T^0 V$ ) is a rank-0 tensor and has grade 0, a vector ( $T^1 V$ ) is a rank-1 tensor and has grade 1 and so on. This property of tensor algebra enables manipulation with objects with different grades within one algebraic structure.

## 2.5 Clifford algebra

A Clifford algebra is denoted  $Cl(V, Q)$ , where  $V$  is a vector space and  $Q$  is a quadratic form defined on the vector space  $V$ . The precise definition can be then written in the following form.

**Definition 2.15** (Clifford algebra). Let  $V$  be an  $n$ -dimensional vector space over a ground field  $K$  and let  $Q$  be the quadratic form on  $V$ . By Clifford algebra is then understood:

$$T(V)/I(Q) = T(V)/(\mathbf{v} \otimes \mathbf{v} - Q(\mathbf{v})\mathbb{1}) \quad (2.6)$$

where  $T(V)$  is a tensor algebra of the vector space  $V$ ,  $I(Q)$  is a two-sided ideal generated by the quadratic form  $Q$  and  $\mathbb{1}$  is the multiplicative identity in  $V$ . [6]

We will be especially interested in the cases, where the ground field  $K$  is  $\mathbb{R}$  or  $\mathbb{C}$ . The Clifford algebra has also some interesting properties. The algebra is free, unitary, associative and also graded.

The alternative for describing Clifford algebra is using the *universal property*. This approach is not constructive but gives a nice context for a deeper understanding.

**Definition 2.16** (Universal property in Clifford algebra). Let  $V$  be a vector space over a field  $K$ , let  $A$  be an unitary associative algebra with operation  $\odot$ . Then let  $Cl(V, Q)$  be a Clifford algebra with operation  $*$ . Then let  $i$  be a linear map  $i : V \rightarrow Cl(V, Q)$  satisfying  $i(\mathbf{v}) * i(\mathbf{v}) = Q(\mathbf{v})\mathbb{1}$  for all  $v \in V$ . It is defined by the *universal property*: given any unitary associative algebra  $A$  over  $K$  and any linear map  $j : V \rightarrow A$  satisfying  $j(\mathbf{v}) \odot j(\mathbf{v}) = Q(\mathbf{v})\mathbb{1}_A$ , where by  $\mathbb{1}_A$  is understood a multiplicative identity in  $A$ , for all  $\mathbf{v} \in V$ , there is a unique algebra homomorphism  $f : Cl(V, Q) \rightarrow A$  such that the following diagram commutes.

$$\begin{array}{ccc} V & \xrightarrow{i} & Cl(V, Q) \\ & \searrow j & \downarrow f \\ & & A \end{array}$$

By commuting is meant that,  $f \circ i = j$ .

The vector space  $V$  is endowed with a quadratic form  $Q$  and therefore an orthogonal basis can be constructed. With use of symmetric bilinear form associated to the quadratic form  $Q$  we can write:

$$\langle \mathbf{e}_i, \mathbf{e}_j \rangle = 0 \text{ for } i \neq j \text{ and } \langle \mathbf{e}_i, \mathbf{e}_i \rangle = Q(\mathbf{e}_i).$$

We are often encountered with Clifford algebras written in the form,  $Cl_{p,q}(V, Q)$ . These are called  $p, q$  Clifford algebras. We will provide a general example explaining the meaning of the coefficients  $p, q$ , properly called a *signature* of the Clifford algebra.

Let  $K = \mathbb{R}$  be the ground field for the vector space  $V = \text{span} \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$  endowed with a quadratic form  $Q(\alpha_1 \mathbf{v}_1 + \dots \alpha_n \mathbf{v}_n)$ , where  $\alpha \in \mathbb{R}$ . The form can be written in the following way:

$$(\alpha_1, \dots, \alpha_n) A \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}$$

where we think about  $\alpha_i$  as coefficients of vector  $\mathbf{v}$  written in basis and by  $A$  is understood a matrix corresponding to the particular quadratic form.

Since we know that  $A$  is a symmetric matrix,  $A = A^T$ , from the definition of the quadratic form, than  $A$  is *orthogonally diagonalizable*:

$$U^T A U = \Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}.$$

Now the base is changed with respect to the diagonalization,  $V = \text{span} \{ \mathbf{x}_1, \dots, \mathbf{x}_n \}$ . The quadratic form is now rewritten in the form:

$$Q(\beta_1 \mathbf{x}_1 + \dots + \beta_n \mathbf{x}_n) = \lambda_1 \beta_1^2 + \dots + \lambda_n \beta_n^2, \quad \beta_i \in \mathbb{R}.$$

Meaning that we can simply write  $Q(\mathbf{x}_i) = \lambda_i$ , whence we can derive  $\mathbf{x}_i^2 = \lambda_i$ .

Now we can further examine what  $\lambda_i$  can be equal to. It actually splits in three non-isomorphic cases:

$$\lambda_i = \begin{cases} 1 \\ 0 \\ -1 \end{cases}$$

It can be shown, that other cases, such as 2 for example, are isomorphic to either 1 or -1.

Let us summarize all information. Let  $A$  be a Clifford algebra  $A \cong Cl(V, Q)$  with  $n$  generators, then the vector space  $V$  and the quadratic form  $Q$  can be written in the following form (quadratic form is already orthogonalized with nonzero elements just on the diagonal):

$$V = \text{span} \{ \mathbf{x}_1, \dots, \mathbf{x}_p, \mathbf{y}_1, \dots, \mathbf{y}_q, \mathbf{z}_1, \dots, \mathbf{z}_r \}, \quad p + q + r = n,$$

$$Q(\alpha_1 \mathbf{x}_1 + \dots + \alpha_p \mathbf{x}_p + \beta_1 \mathbf{y}_1 + \dots + \beta_q \mathbf{y}_q + \gamma_1 \mathbf{z}_1 + \dots + \gamma_r \mathbf{z}_r), \quad \alpha_i, \beta_i, \gamma_i \in \mathbb{R}.$$

For each  $\mathbf{x}, \mathbf{y}, \mathbf{z}$  is now a special case, already described previously:

$$\mathbf{x}_i^2 = 1, \quad \mathbf{y}_i^2 = -1, \quad \mathbf{z}_i^2 = 0.$$

In this thesis we will be interested in algebras where  $r = 0$ . The coefficients  $p, q$  can be now easily explained. Number of basis elements, which square to 1 is coefficient  $p$  and number of basis elements, which square to -1 is coefficient  $q$ . The particular algebra is then written with it's signature:

$$Cl_{p,q}(V, Q).$$

## 2.6 Exterior algebra

The exterior algebra, also known as Grassman algebra, is used especially in geometry to study areas, volumes and higher dimension objects. It uses the exterior product, which was already mentioned.

**Definition 2.17** (Exterior algebra). Let  $V$  be an  $n$ -dimensional vector space over a ground field  $K$  and let  $I$  be the ideal generated by  $(\mathbf{u} \otimes \mathbf{u})$ . By exterior algebra is then understood:

$$T(V)/(\mathbf{u} \otimes \mathbf{u}), \tag{2.7}$$

which is usually denoted as  $\wedge(V)$ .

The product defined on this algebra is the so called exterior product, also known as the wedge product. [6]

**Definition 2.18** (Exterior product). Let  $V$  be a vector space over a ground field  $K$ . Then the map  $\wedge : V \times V \rightarrow V \otimes V$  is called an *exterior product* with the following properties:

(A1) *Antisymmetry*:

$$\mathbf{u} \wedge \mathbf{v} = -\mathbf{v} \wedge \mathbf{u}$$

(A2) *Scaling*:

$$(\alpha \mathbf{u}) \wedge \mathbf{v} = \alpha(\mathbf{u} \wedge \mathbf{v})$$

(A3) *Distributivity*:

$$\mathbf{u} \wedge (\mathbf{v} + \mathbf{w}) = (\mathbf{u} \wedge \mathbf{v}) + (\mathbf{u} \wedge \mathbf{w})$$

for all  $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$  and  $\alpha \in K$ .

This definition can be extended  $\wedge : V \wedge \cdots \wedge V \rightarrow V^{\otimes k}$  and an associativity is added. The linear combination of the products of two vectors  $\mathbf{u} \wedge \mathbf{v}$  is called a *bivector*, similarly the linear combination of the products of three vectors is called a *trivector* and a linear combination of products of  $k$  vectors is called a *k-vector*. A linear combination of mixed  $k$ -vectors is called a *multivector*. We will be also using a term called *blade*. Blade of  $k$ -grade consists of wedge product of  $k$  vectors. Bivector can be understand as an oriented plane and trivector as an oriented volume.

**Definition 2.19** (Exterior power). The  $k$ th exterior power,  $\wedge^k(V)$ , is defined as a vector subspace of  $\wedge(V)$  generated by elements in the form  $\mathbf{x}_1 \wedge \mathbf{x}_2 \wedge \cdots \wedge \mathbf{x}_k$ , where  $\mathbf{x}_i \in V$ .

## 2.7 Bra-Ket notation

When working with vectors, the bra-ket notation will be used. This notation is widely used in quantum computing to describe quantum states. It gives a way to describe a vector  $\mathbf{v}$  in a complex vector space. A *ket* is denoted as  $|a\rangle$ , and a *bra* is denoted as  $\langle a|$ .

Using bra-ket notation is more efficient than writing vectors in columns or rows, and it allows for easier manipulation of vectors. Let's take a closer look at the relationship between bras and kets. A bra represents a column vector, and a ket represents its conjugate transpose:

$$|a\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \quad \langle a| = (a_1^* \quad a_2^* \quad \dots \quad a_n^*). \quad (2.8)$$

Generally can be stated:  $\langle a|^\dagger = |a\rangle$ ,  $|a\rangle^\dagger = \langle a|$ , where the symbol  $\dagger$  is used for marking conjugate transpose.

Inner product of two vectors is then written in the form  $\langle a||b\rangle$ , which is shortened to  $\langle a|b\rangle$ .

## 3 Quantum Physics

Everything comes down to the nature of light. Many physicist tried to understand, whether the light is a particle or a wave. The tricky thing about light is, that it exhibits both particle and wave characteristics. Phenomena such as diffraction, polarization or interference suggest, that it should

be a wave. The famous double-slit experiment also supports this presupposition. On the other hand, in order to explain for example the photoelectric effect, we must suppose that there are particles with given state and momentum. This leads to the conclusion, that light has a particle character. The previous examples are just one of many and try to demonstrate, how complicated the light is. General answer to the basic question, whether the light is a wave or particle, would have been, that it is both and simultaneously neither. Light is much more complex and shouldn't be so simply categorized.

### 3.1 Electromagnetic waves

Electromagnetic waves, just EM waves from now on, are a fundamental aspect of the physical universe and play a crucial role in the behaviour and interaction of matter and energy. They are a type of wave that consists of oscillating electric and magnetic fields, which are perpendicular to each other and to the direction of wave propagation. EM waves travel at the speed of light in a vacuum. When travelling through different materials their properties are impacted. One of the most remarkable aspects of electromagnetic waves is their ability to transport energy through space without requiring a physical medium[10]. This property is what allows us to use cell phones, receive radio waves or watch television.

EM waves can be described using various parameters including frequency, wavelength, amplitude, phase shift or polarization. Based on their frequency, waves can be classified into various regions. Starting with the waves with a very low frequency we can describe radio waves, microwaves, infrared radiation, visible light, ultraviolet radiation, X-rays and come to waves with extremely high frequency, usually called gamma waves [1].

An important phenomenon related to EM waves is polarization. EM waves oscillate in an electromagnetic field and can be described using two orthogonal vectors oscillating in electric and magnetic fields. In an unpolarized wave, the directions of the vectors are random. However, polarized EM waves have only one direction and all others are canceled out. This can be achieved using various filters.

### 3.2 Mathematical description of electromagnetic waves

EM waves oscillate in an electromagnetic field, with the electric field represented by the  $x$ -axis and the magnetic field represented by the  $y$ -axis. The basis for this field consists of unit vectors denoted  $|\rightarrow\rangle$  and  $|\uparrow\rangle$ . These vectors are then multiplied by the respective amplitude and cosine function, which takes into consideration the wave's frequency, time, and phase shift. This representation allows us to describe waves as vectors in the  $x$ - $y$  plane, which is perpendicular to the direction of wave propagation. The wave propagation itself is represented by the  $z$ -axis.

We will start with the example of a polarized wave. Just a quick recall, that by polarization we understand canceling out one particular direction of wave oscillation. By horizontally polarized wave we then understand a wave with a following form:

$$\mathbf{E} = \begin{pmatrix} A_x \cos(2\pi f_x t + \theta) \\ 0 \end{pmatrix} = A_x \cos(2\pi f_x t + \theta) |\rightarrow\rangle + 0 |\uparrow\rangle, \quad (3.1)$$

where  $A_x$  is amplitude,  $f_x$  is frequency,  $t$  is time and  $\theta$  is a phase shift.  $|\rightarrow\rangle$  and  $|\uparrow\rangle$  are unit vectors representing particular direction.

Similarly the vertically polarized wave has just its vertically oriented component

$$\mathbf{E} = \begin{pmatrix} 0 \\ A_y \cos(2\pi f_y t + \theta) \end{pmatrix} = 0 |\rightarrow\rangle + A_y \cos(2\pi f_y t + \theta) |\uparrow\rangle. \quad (3.2)$$

General description of a wave can be easily obtained. It will be just a sum of the examples given above. Sum of these two vectors representing horizontally and vertically polarized waves will create a new vector. This general vector will provide us a way to describe any given wave at any moment in time.

$$\mathbf{E} = \begin{pmatrix} A_x \cos(2\pi f_x t + \theta) \\ A_y \cos(2\pi f_y t + \theta) \end{pmatrix} = A_x \cos(2\pi f_x t + \theta) |\rightarrow\rangle + A_y \cos(2\pi f_y t + \theta) |\uparrow\rangle \quad (3.3)$$

General vector can be then written in this simple form:

$$|\psi\rangle = \alpha |\rightarrow\rangle + \beta |\uparrow\rangle. \quad (3.4)$$

Alternative way to describe a wave is with complex numbers. Following example is showing, how to rewrite the expression using complex numbers:

$$A \cos(2\pi f t + \theta) = A e^{i(2\pi f t + \theta)}, \quad (3.5)$$

$$\mathbf{E} = A_x e^{i(2\pi f t + \theta)} |\rightarrow\rangle + A_y e^{i(2\pi f t + \theta)} |\uparrow\rangle,$$

where  $f$  is the frequency of the wave.

### 3.3 Energy of the wave

When describing waves, the key attribute, which is measured is wave's energy. The following formula describes energy of the wave.

**Definition 3.1** (Energy of the wave). The energy of the wave is given by

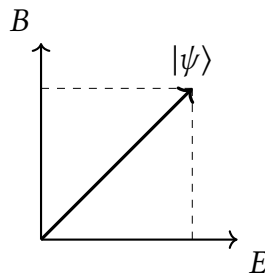
$$E = n\hbar f, \quad (3.6)$$

where  $n \in \mathbb{N}$ ,  $\hbar$  is Planck's constant and  $f$  is a frequency of the wave.

It is important to note, that in the formula is an integer  $n$ , which belongs to the natural numbers. When  $n = 1$  is plugged into the previous equation we get the smallest possible amount of energy a wave can have. It corresponds to the particle called photon. This minimum amount of energy is no further divisible. The fact that  $n \in \mathbb{N}$  also leads to an important insight. Wave's energy is a discrete variable. In other words, changes in energy comes in fixed small quantum and energy of any given wave is then just a multiple of the photon's energy. This indivisibility is where all the quantum magic comes from.

From the mathematical description, which was defined previously, is evident, that, knowing all the parameters, we can decompose a wave into basis vectors and then obtain the value of energy in these directions. Square root of parameter  $\alpha$ , respectively  $\beta$ , really gives us the component of the energy in  $|\rightarrow\rangle$ , respectively  $|\uparrow\rangle$  state.

We will now decompose the following unit vector into basis vectors.



**Figure 3.1:** Unit vector

Knowing that the vector  $|\psi\rangle$  is a unit vector we obtain  $\alpha = \sqrt{1/2}$  and  $\beta = \sqrt{1/2}$ . Now we can use the equation (3.6) and get

$$\begin{aligned} \mathbf{E} &= (\hbar f)(1)^2 \\ &= \hbar f(\sqrt{1/2})^2 |\rightarrow\rangle + \hbar f(\sqrt{1/2})^2 |\uparrow\rangle \\ &= \hbar f 1/2 |\rightarrow\rangle + \hbar f 1/2 |\uparrow\rangle. \end{aligned}$$

We made decomposition into basis vectors. However, what is important to notice is, that energy in these directions are fractions. This can be explained in quantum physics, where the whole paradigm of the waves is shifted.

### 3.4 Classical versus quantum understanding

Classical wave theory teaches us, that EM waves are oscillating in two directions, mutually inducing each other. In every moment we can decompose vector representing our electromagnetic wave into two components and measure them.

Quantum understanding is a bit more complicated [7]. There must be defined the concept of superposition in this context. Proper and precise description of quantum superposition would require a deep dive in quantum mechanics and this thesis is not discussing quantum mechanics as the main topic. So for our purpose superposition state means, that a particle is in more states until the measurement is carried out. In our case is the energy of the wave in both directions representing electric and magnetic fields. But after the measurement is done, all energy is measured just in one direction.

Now come back to the general form of vector in (3.4). We will demonstrate the difference in understanding the nature of waves. Just reminder what  $|\alpha|^2$  means in classical understanding. It is an amount of energy in direction corresponding to  $\alpha$ . However, in quantum understanding  $|\alpha|^2$  is a probability, that all energy will be measured in the direction corresponding to  $\alpha$ .

To further illustrate the difference in understanding, consider wave transformation. In classical theory, Euclidean distances must be preserved during wave transformation, and thus special orthogonal matrices are used. However, in quantum mechanics, the focus is on probabilities, and therefore Hermitian unitary matrices are used to preserve these probabilities, see Definition 5.1 and Definition 5.2 for more details. These matrices are particularly useful for describing the transformation of quantum states, and they play an essential role in quantum computing and information processing. In essence, the use of these matrices reflects the probabilistic nature of quantum mechanics and highlights the difference between classical and quantum understanding of waves.

## 4 Quantum computing

Every computing mechanism is a physical process. One physical state is identified with one information and another physical state with another information. Computing is nothing else than transforming one state to another. Different states can also interact with each other and create new unique states associated with new information [13].

Classical computing has building block in a bit (shortcut from binary digit). It is the smallest unit of data. A bit can acquire two values, usually 0 and 1. Computer science works with bits in a general way and it does not take into consideration the actual physical implementation of bit. The advantage of this approach is, that it can develop the concept of bit in abstract and general way and do not worry about problems with implementation. Mathematical logic behind computing with bits ignore effects of real world, that always influence the process.



Implementation into the real world causes problems and we almost certainly lose some amount of precision in computing. Take for example one of the most common way to store data using a hard drive or other similarly working storage media. Hard drive usually has an aluminium disk covered with a thin magnetic layer. One bit is then stored in a concrete area on a disk platter. Electromagnetic radiation or excessive heat can enter the storing process and partially destroy the information. When observing this mechanism from a microscopic point of view, a huge amount of atoms or molecules are used to store just one bit. Knowing that a bit can acquire just two values, the whole concept seems to be a bit inefficient.

It seems obvious to try diminishing the amount of atoms needed for storing one bit. The ideal would be to represent one bit with one atom or photon. The solution occurs almost instantly. We can polarize photons in horizontal or vertical direction and identify this direction with 1 or 0 value. However, as it was shown previously, our universe behaves differently on atomic and subatomic level. Quantum mechanics steps into the game and the whole paradigm must be shifted. A completely new area of computing arises.

Quantum computing tries to exploit quantum mechanics phenomena and use it for advantage rather than stressing out about uncertainties. It acknowledges the fact that a physical system usually is not in just a state A or a state B and lays somewhere between. Bits are no more useful and a new concept must be defined to represent data.

## 4.1 Qubit

A qubit is the basic unit of quantum information and it differs from a classical bit which has only two possible states - 0 or 1. A qubit, on the other hand, can exist in a superposition of both states. This means that a qubit can store not just 0 or 1, but all possible combinations of these states at the same time. This opens up a vast number of possibilities for processing and storing information in ways that are impossible with classical bits. However, this increased flexibility comes at a cost. The main challenge of quantum computing is to find ways, how to represent and manipulate these superposition states mathematically, as well as designing and building a real world hardware that can effectively implement these computations.

For representing the qubit, two orthonormal basis states will be defined. They will be denoted in the following way:  $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  and  $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ . These two states are often called the computational basis.

General quantum state is identified with a normalized vector, generated from the orthonormal computational basis, in two dimensional space. As we have mentioned earlier, qubit is a combination, in our case linear combination, of the basis state and therefore could be written in the following form:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad (4.1)$$

where the coefficients  $\alpha$  and  $\beta$  are complex numbers. Notice that the equation is almost identical to the equation (3.4). The only difference is, that in the first case the basis are directions of polarized light written in a form of arrows. In the second case the basis are two orthonormal vectors written in the form of bra-ket notation. This is the first example, how to implement our abstract concept of qubit into the real world.

The coefficients  $\alpha$  and  $\beta$  are correctly called the *probability amplitudes*. If we measure any given qubit, than  $|\alpha|^2$  and  $|\beta|^2$  is the probability of  $|0\rangle$  and  $|1\rangle$ , respectively, outcome. The second axiom of probability gives us the following constraint  $|\alpha|^2 + |\beta|^2 = 1$ , which always has to be satisfied. The coefficients  $\alpha$  and  $\beta$  are complex numbers, therefore the qubit has four degrees of

freedom. There are many ways, how to handle the fact, that we are working with something with four degrees of freedom and how to easily visualise any given qubit.

## 4.2 Bloch sphere

An elegant way to represent qubit is via a Bloch sphere.

Coming back to the Definition 4.1 we can see, that to describe a single qubit, a combination of two basis states is needed. We also know that the coefficients  $\alpha$  and  $\beta$  are complex numbers. So any given state can be described by four real numbers. It would be very difficult to visualise a system with four degrees of freedom.

However, since we identified quantum states with complex vectors, where the coefficients are probability amplitudes, we can now introduce a *phase*. A phase is an angle between the real and imaginary components of the complex vector. There are actually two types of phases. A *relative phase* refers to the phase difference between two probability amplitudes. It is a measure how the amplitudes interfere with each other. A *global phase* refers to the overall phase in a quantum state. It is a common phase factor that affects all the probability amplitudes in the state. It is important to emphasize, that the global phase does not impact probabilities of measuring a particular outcome. And what is more, the global phase has no physical significance, because it cancels out, when the probabilities are calculated. It means, that one degree of freedom can be removed. When taking into the consideration the constraint given by the second probability axiom, another degree of freedom is taken away. This leads to the system with two degrees of freedom.

As the name suggests, a sphere will be used to represent a qubit. And since we have two degrees of freedom, spherical coordinates can be efficiently used. Then the general qubit can be written in the following form

$$|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi} \sin(\theta/2)|1\rangle = \cos(\theta/2)|0\rangle + (\cos \phi + i \sin \phi) \sin(\phi/2)|1\rangle, \quad (4.2)$$

where  $0 \leq \theta \leq \pi$  and  $0 \leq \phi \leq 2\pi$ .

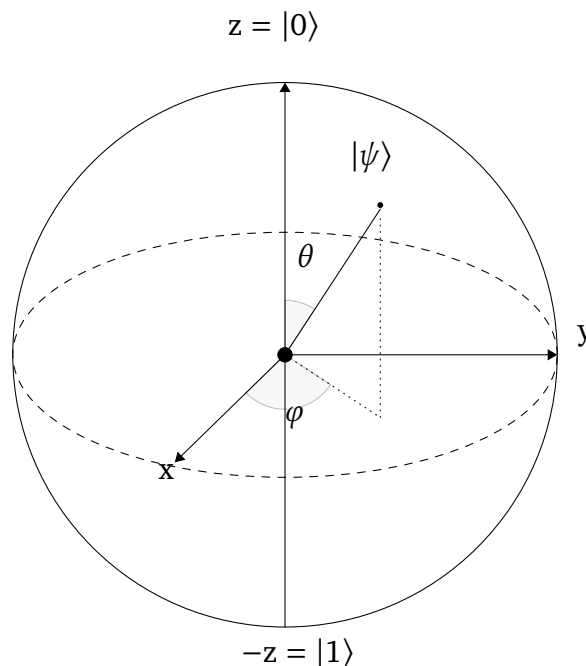
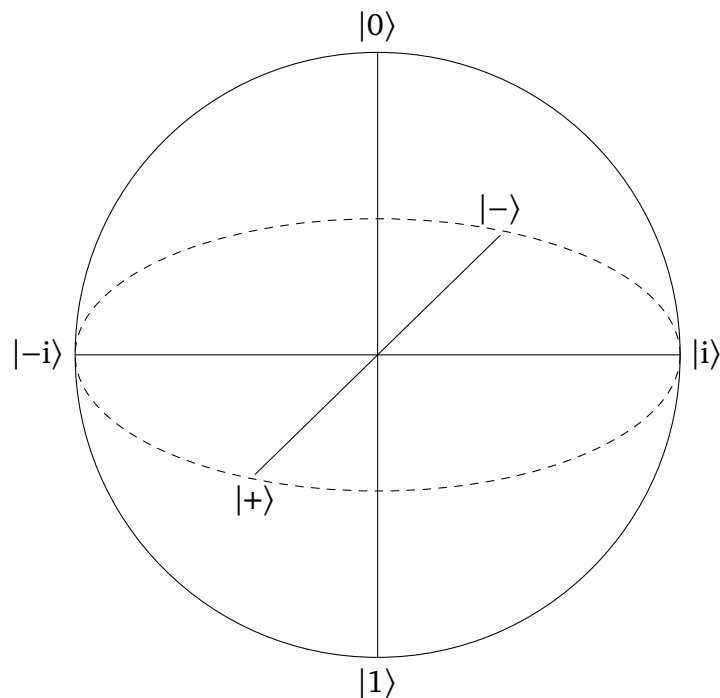


Figure 4.1: Bloch sphere

With the appropriate choice of parameters  $\theta$  and  $\phi$  any state can be uniquely described, see Figure 4.1. More precisely, there will be a one to one correspondence between quantum states (represented by qubits) and points on the sphere. It is important to notice that the qubit has a form of a two dimensional vector. The choice of the basis is arbitrary. From the beginning we have worked with  $|0\rangle$  and  $|1\rangle$ . These states can be identified with the north and the south pole of our sphere. However, every pair of two antipodal points is orthogonal, its inner product is zero. They can therefore serve us as the orthogonal base. For practical reasons we will stick to the  $|0\rangle$  and  $|1\rangle$  choice.

States  $|0\rangle$  and  $|1\rangle$  are put on  $z$ -axis. Let us now look closely on  $x$ -axis and  $y$ -axis. They have their own pair of antipodal points. On  $x$ -axis we usually talk about  $|+\rangle$ ,  $|-\rangle$  and on  $y$ -axis about  $|i\rangle$ ,  $|-i\rangle$ . These pairs can also form a basis, as we have stated earlier. Since the  $|0\rangle$ ,  $|1\rangle$  basis was chosen, now express these pairs of points in the basis vectors. These points are then visualised on the sphere, see Figure 4.2.

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) & |-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ |i\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) & |-i\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \end{aligned} \quad (4.3)$$



**Figure 4.2:** Bloch sphere axes

To put the classical bit into this new perspective. It was stated, that a bit can acquire just two values, 0 and 1. So it can be either on the north pole or on the south pole on our sphere and nowhere else. Quantum bit can acquire any position on the sphere, classical bit must choose between two antipodal points chosen as the basis. There should be clearly seen an immense variability, which is provided by qubit.

### 4.3 Multiple Qubits

Qubits were associated with state vectors and a state vector of dimension 2 was used to represent a single qubit. To represent multiple qubits we just need to increase the vector's dimension. One qubit is represented with 2-dimensional vector, two qubits are represented with 4-dimensional vector and  $n$ -qubit is represented with  $2^n$ -dimensional vector. The dimension also determines, how many basis vectors are needed. Let us see on an example of two qubits, how the state vector is constructed.

$$|a\rangle = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle = \begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix}. \quad (4.4)$$

Let us just remind, that the coefficients  $a_{ij}$  are the *probabilistic amplitudes*. The square of the coefficient  $|a_{ij}|^2$  is the probability of measuring the state  $|ij\rangle$ .

The second axiom of probability must hold, therefore we come to the following normalisation condition:

$$|a_{00}|^2 + |a_{01}|^2 + |a_{10}|^2 + |a_{11}|^2 = 1. \quad (4.5)$$

Same construction will be used for  $n$ -qubits. The normalisation condition must always hold.

For two separated qubits we define a *collective state* using the Kronecker product, see Definition 2.3. For representing the collective state we will also use the state vector. The dimension of the state vector is  $2^n$ , where  $n$  is the number of qubits we are describing. Let us see on an example how the collective state of two qubits is obtained

$$|a\rangle = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}, |b\rangle = \begin{pmatrix} b_0 \\ a_1 \end{pmatrix},$$

$$|ba\rangle = |b\rangle \otimes |a\rangle = \begin{pmatrix} b_0 * \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \\ b_1 * \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} b_0 a_0 \\ b_0 a_1 \\ b_1 a_0 \\ b_1 a_1 \end{pmatrix}. \quad (4.6)$$

We took two qubits as an input and used the Kronecker product to construct the state vector representing the collective state. The final statevector has the dimension 4. It was stated, that a state vector describing a collective state has a dimension  $2^n$ . We took two qubits, so  $2^2$ , which is truly 4.

### 4.4 Measurement

It was said, that a qubit can acquire not just 0 and 1 values, but all their combinations. However, when the measurement is carried out, the qubit is measured either in a position corresponding to  $|0\rangle$  or in a position corresponding to  $|1\rangle$ . It is important to clarify in which basis the measurement is performed. We can measure the same qubit in different basis and get different results. Measuring in the  $|0\rangle, |1\rangle$  basis, gives the possible outcomes  $|0\rangle$  and  $|1\rangle$ . But if the basis is changed to  $|+\rangle, |-\rangle$  for example, then the possible outcomes are  $|+\rangle$  and  $|-\rangle$ . It is clear, that the choice of the basis is crucial for the right interpretation of the measurement. We will stick to the  $|0\rangle, |1\rangle$  basis.

It is actually quite simple to calculate the probability of the outcomes for any given qubit. Computing the inner product of a basis state and a state vector, representing the qubit, gives the probability of a qubit ending in the particular basis state. To be precise, the square of this

value is the desired probability. Take a general vector  $|\psi\rangle$  for example and see how to obtain the probabilities of ending in the basis states  $|0\rangle, |1\rangle$ ,

$$p(|0\rangle) = |\langle 0|\psi\rangle|^2 \quad p(|1\rangle) = |\langle 1|\psi\rangle|^2. \quad (4.7)$$

## 5 Quantum gates

Having associated qubit with a state vector we can now transform it. The transformation of a qubit is nothing more than just rotating around the Bloch sphere. Classical computation works with logical gates. These gates take as an input single or multiple bit, perform an operation defined by mathematical logic and give another bit or multiple bit as an output.

For meaningful quantum computing, logical gates must be also introduced. Bloch sphere is very useful for visualising, how single qubit gates work. For multiple qubit gates a tensor product will be used.

For visualisation will be used Qiskit [11].

### 5.1 Single qubit gates

Single qubit gates take as input one qubit represented via state vector, apply operator and give another state vector as output

$$M|\psi\rangle \rightarrow |\psi'\rangle. \quad (5.1)$$

We can see, that the operator  $M$  changes our state vector to another. From linear algebra is well known, that matrices are used to transform vectors. So matrices will represent qubit gates.

These matrices must satisfy two important properties. They must be Hermitian and unitary.

**Definition 5.1** (Hermitian matrix). A matrix is called Hermitian when the matrix is equal to its conjugate transpose,  $A = \overline{A^T} = A^*$ . Or equivalently, a matrix is Hermitian when the element in the  $i$ -th row and  $j$ -th column is equal to the complex conjugate of the element in the  $j$ -th row and  $i$ -th column,  $a_{ij} = \overline{a_{ji}}$ .

**Definition 5.2** (Unitary matrix). An invertible square matrix is called unitary when its conjugate transpose  $A^*$  is also its inverse,  $AA^* = A^*A = I$ , where  $I$  is an identity matrix.

These properties are crucial, because we need state vectors to stay on the Bloch sphere. The only way to provide this, is to keep the norm of the vector, even after multiplying it with the matrix. The great advantage of unitary matrices is, that they preserve the norm of the vector.

There are indefinitely many ways, how to write a general form of operator  $M$ . One of possible is following:

$$M(\phi, \theta, \lambda) = \begin{pmatrix} \cos(\theta/2) & e^{-i\lambda} \sin(\theta/2) \\ e^{i\phi} \sin(\theta/2) & e^{i(\phi-\lambda)} \cos(\theta/2) \end{pmatrix}. \quad (5.2)$$

This form is however really impractical and not needed. Basic matrices will be defined. The demanded operator is then obtained by combining these matrices. With just small set of matrices, any requested matrix can be obtained. Plus, the product of two unitary matrices is also an unitary matrix. Chosen matrices will be multiplied, create a new matrix, which will represent the operator. The state vector is then multiplied by the final matrix and the transformation is performed.

Since we have introduced the Bloch sphere for representing qubits, we can imagine the transformations as rotations on the Bloch sphere. If unitarity of the matrix is satisfied, the state vector stays on the sphere and just rotates around. It will be demonstrated how different gates perform different rotations.

### 5.1.1 NOT gates

Basic logic gate in classical computing is the NOT gate. It flips bit's value to other possible. Bit with value 0 resp. 1 is flipped to 1 resp. 0. The situation with qubit is a bit more complicated. Since qubit can acquire any value on the Bloch sphere, NOT gate cannot be so easily defined.

We start with the simplest example. If a qubit is on the north pole of the Bloch sphere, it has value  $|0\rangle$ , than the NOT gate should change its value to  $|1\rangle$ . That could be achieved by rotating the state vector around  $x$ -axis or around  $y$ -axis. When NOT gate is applied to a state vector, its value is changed to the value associated with the antipodal state vector. Matrices which provide this transformation are not always easy to find. For now it will be enough to define just rotations around  $x,y,z$ -axes, respectively:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (5.3)$$

**Example 5.1.** Let us take  $|0\rangle$  and apply the  $X$  matrix. It should change to  $|1\rangle$ .

$$X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle.$$

Here is a visualisation using Qiskit [11]. The first Figure 5.1 is the base state  $|0\rangle$ . Than the  $X$ -gate acts on it and we can see the result on the Figure 5.2.

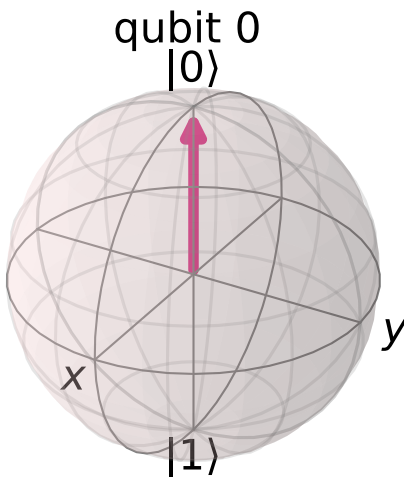


Figure 5.1: State vector in  $|0\rangle$  state

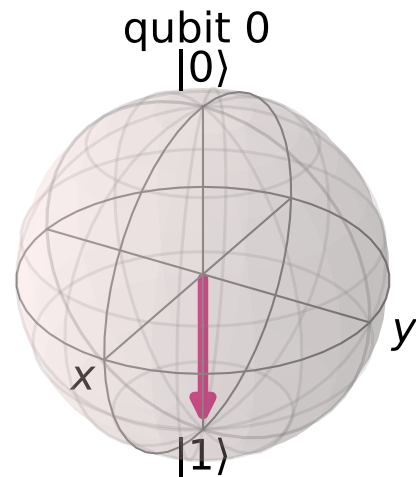


Figure 5.2: After applying X-gate

The effect of the  $Y$ -gate would be the same on this particular state vector. Let us see on example how  $Z$ -gate would work. On the Figure 5.3 we can see a state vector  $|+\rangle$  and on Figure 5.4 we can see that after the  $Z$ -gate acts on the qubit, the state is truly  $|-\rangle$ . This corresponds to the rotation around the  $z$ -axis.

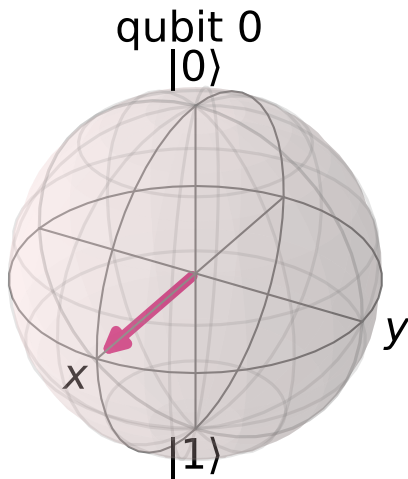
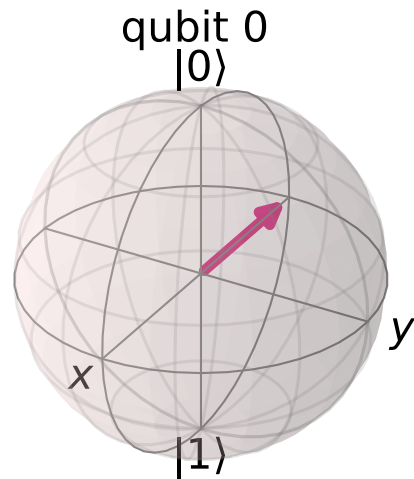
Figure 5.3: State vector in  $|+\rangle$  state

Figure 5.4: After applying Z-gate

### 5.1.2 Hadamard gate

Maybe the most important gate in quantum computing is called the Hadamard gate. It occupies special position in the set of elementary matrices. It may not be obvious at first glance, why is this matrix so essential. The importance will be explained later, now just the definition will be provided. When the Hadamard gate acts on a state vector, it rotates the vector around the  $(x - z)$ -axis. The form of the matrix representing the Hadamard gate is following:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (5.4)$$

It is obvious, that this matrix is Hermitian. Unitarity is satisfied by the  $\frac{1}{\sqrt{2}}$  in front of the matrix. Let us demonstrate, how this matrix acts on basis state vectors

$$H|0\rangle = |+\rangle, \quad H|1\rangle = |-\rangle.$$

This can be easily visualised on the Bloch sphere. We will start with the qubit in state  $|0\rangle$ , Figure 5.5. Then the Hadamard gate is applied and the result is truly the qubit in  $|+\rangle$  state, Figure 5.6.

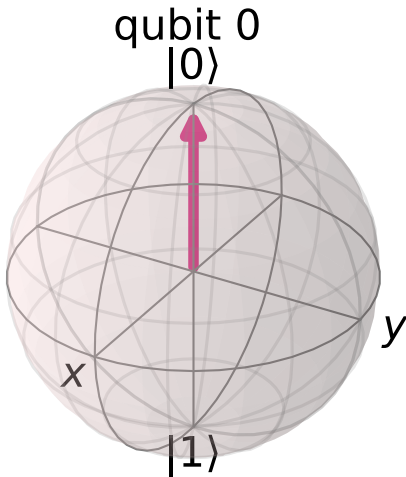
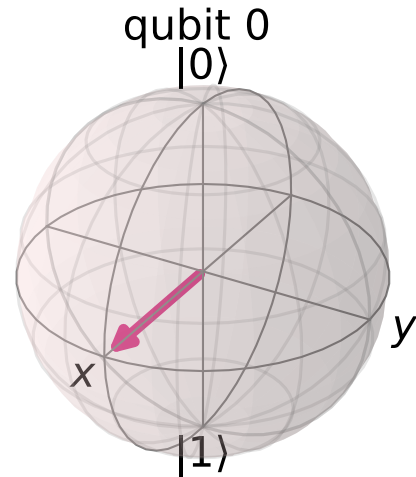
Figure 5.5: State vector in  $|0\rangle$  state

Figure 5.6: After applying Hadamard gate

### 5.1.3 Phase gate

Phase gate is used to rotate the state vector around the  $z$ -axis. It takes a real number  $\phi$  as a parameter, which specifies the rotation. The form of the matrix is following:

$$P = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}. \quad (5.5)$$

We now explore this gate more closely. We can change the parameter  $\phi$  and we obtain a so called phase gate family tree. Most important members of this family are these matrices:

$$\begin{aligned} \phi = \pm\pi, & \quad P = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z, \\ \phi = \frac{\pi}{2}, & \quad P = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix} = S, \\ \phi = \frac{\pi}{4}, & \quad P = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} = T, \\ \phi = 0, & \quad P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I. \end{aligned}$$

When specific input of parameter  $\phi$  is plugged in, special gates arise. The first gate turns out to be a known  $Z$ -gate.

The second one is called  $S$ -gate or  $S^\dagger$ . It performs a quarter turn around the  $z$ -axis. Two consecutive  $S$ -gates has same effect as the  $Z$ -gate

$$SS = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z.$$

The third gate, called  $T$ -gate, performs the eighth turn around the  $z$ -axis. It can be similarly shown that four consecutive  $T$ -gates have the same effect as  $Z$ -gate. The last mentioned gate, called  $I$ -gate, is in the form of an identity matrix. Therefore it has no effect on any given state vector.



### 5.1.4 Gates tensor product

One additional thing, that needs to be defined, in order to be able to perform a meaningful computation, is how a single qubit gate acts on a multiple qubit vector. Or more specifically, how will two single qubit gates act simultaneously on a multiple qubit vector. The Kronecker product will become a handy tool. Take a look on the example. Let us consider two single qubit gates  $A, B$  and a vector  $q$  representing the state of two qubits  $q_1, q_2$ .

$$A|q_2\rangle \otimes B|q_1\rangle = (A \otimes B)|q_2q_1\rangle, \quad (5.6)$$

$$\begin{aligned} A \otimes B &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \otimes \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11} * \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} & a_{12} * \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \\ a_{21} * \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} & a_{22} * \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \end{pmatrix} \\ &= \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \end{pmatrix}. \end{aligned} \quad (5.7)$$

## 5.2 Multiple qubit gates

Multiple qubit gates are a critical component of quantum computing, just QC from now on, as they enable operations on multiple qubits simultaneously. Just like with single qubit gates, multiple qubit gates also rely on matrix operations to transform the state vectors that describe the quantum system. However, the matrices used for multiple qubit gates are much larger, with dimensions that depend on the number of input qubits. The dimension of a multiple qubit gate matrix equals to  $2^n \times 2^n$ , where  $n$  is the number of input qubits. These matrices must satisfy the same properties as single qubit gates, and their construction requires careful consideration of how the gate acts on basis states.

To illustrate this concept, we can consider some common examples of multi-qubit gates. For instance, there are 2-qubit gates such as the CNOT (Controlled NOT) gate, the SWAP gate, and the CZ gate (controlled phase gate). Additionally, there are 3-qubit gates like the Toffoli gate (also known as the CCNOT gate). It's important to note that these gates are just a few examples of many possible multi-qubit gates, and the selection of a specific gate will depend on the desired computational outcome. However, understanding the basics of how these gates operate and their effect on state vectors is a crucial step in designing effective quantum circuits.

### 5.2.1 CNOT gate

Often called Controlled-NOT gate is the most used multi-qubit gate. It takes two qubits as an input, checks the value of the first qubit, called *controlled qubit*, and if it is 1, then it flips the second qubit, called *target qubit*. Let us see how the gate acts on basis states.

$$\text{CNOT}|00\rangle = |00\rangle \quad \text{CNOT}|01\rangle = |01\rangle \quad \text{CNOT}|10\rangle = |11\rangle \quad \text{CNOT}|11\rangle = |10\rangle$$

The matrix representing the CNOT gate has the following form:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (5.8)$$

The first two rows correspond to the cases, where the controlled qubit is in a state  $|0\rangle$  and therefore nothing is changed. The third and fourth row corresponds to the situation, when the controlled qubit is in a state  $|1\rangle$  and therefore the value of the target qubit is swapped.

**Example 5.2.**

$$\text{CNOT} \cdot |10\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle$$

### 5.2.2 SWAP gate

The SWAP gate is really simple. It just swaps the state of the two qubits. It has no effects on states  $|00\rangle, |11\rangle$ , but it affects the states  $|10\rangle, |01\rangle$ . The matrix representation is following:

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.9)$$

**Example 5.3.**

$$\text{SWAP}|01\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle$$

### 5.2.3 CZ gate

Also called Controlled-Z gate, performs a rotation around the z-axis, if the first qubit (controlled qubit) is 1. It has effect for example on  $|11\rangle$  state, which is then changed to  $-|11\rangle$ . General form is following:

$$\text{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (5.10)$$

**Example 5.4.**

$$\text{CZ}|11\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix} = -|11\rangle$$

### 5.2.4 Toffoli gate

Also known as Controlled-Controlled-NOT gate. It works similarly as the CNOT gate, however it checks two controlled qubits and only if both are 1, then it flips the third one. It can be

implemented using single qubit gates and CNOT gate. The matrix representation is following:

$$\text{Toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (5.11)$$

### 5.3 Quantum circuits

We can now start drawing circuits. Qiskit [11] will be used for that. let us take a look on an example, Figure 5.7.

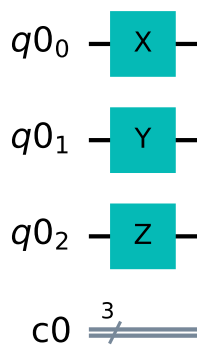


Figure 5.7: Basic single qubit gates

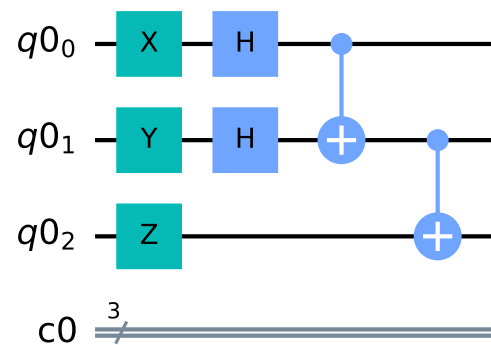


Figure 5.8: Hadamard gate and C-NOT gate

This circuit takes three qubits as an input. Each of them is stored in the separate wire. These wires will be noted with  $q0_i$  and will be used to store qubits. We are interested just in the lower case index  $i$  and the 0 in the qubit expressions has no special meaning for us. Classical bits are also used in circuits. We use them to store the value acquired from the measurement of the qubit. They are denoted  $c0$  and merged to one wire in order to save space and to make circuits more understandable. We can see, that on the quantum wires are already the quantum gates, namely X-gate, Y-gate and Z-gate. These gates act on particular qubits. Other gates can be added now, such as a Hadamard gate and C-NOT gate, see Figure 5.8. Notice, that the C-NOT gate is a 2-qubit gate, meaning it takes two qubit as an input. The last piece to a meaningful and functional circuit is a measurement. That is where the classical wires come in use. Measurement is put on a quantum wire and the result is stored into a classical bit, because since there are only two options, how the measurement can end up, we can store this value into a classical bit. On the Figure 5.9 can be seen, how the measurement is performed and where the results are stored.

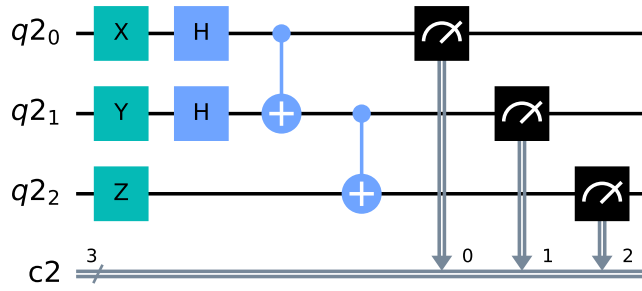


Figure 5.9: Measurement

This circuit has no purpose, but can serve us as an example. In the next chapter will be shown, how we can represent these circuits using GA.

## 6 Geometric algebras in Quantum computing

In more advanced circuits the amount of input qubits increase rapidly and so does the matrices dimension. The computation becomes more and more complicated, since the dimension of the matrix increase exponentially. The idea of implementing GA in QC lays in the special representation of qubits. Every quantum state of a qubit is associated with an element of the GA and even quantum gates are associated with elements of GA. It allows us for direct application. Action of a quantum gate on a qubit is just a product of the elements associated with the particular gate and quantum state.

### 6.1 Geometric algebra

We will now define an appropriate apparatus of GA for the purpose of QC. In the first chapter were discussed and defined basic algebraic concepts in a general way. These definitions will be now modified to desired forms. We will start with a real GA. In order to define GA a new concept called *geometric product* needs to be introduced [3] [4]. The geometric product is a combination of the inner and exterior product. Let us have a real vector space  $\mathbb{R}^n$ . The vector space has an orthonormal basis  $(\mathbf{e}_1, \dots, \mathbf{e}_n)$ . It is also equipped with a bilinear form  $B$  with signature  $(p, q)$ .

$$B(\mathbf{e}_i, \mathbf{e}_j) = \begin{cases} 1 & \text{if } i = j = 1, \dots, p \\ -1 & \text{if } i = j = p + 1, \dots, m \\ 0 & \text{if } i \neq j \end{cases} \quad (6.1)$$

Notice that the bilinear form  $B$  is symmetric. Therefore the quadratic form  $Q$ , which is associated to this bilinear form can be defined. The relation is following:

$$B(\mathbf{a}, \mathbf{b}) = \frac{1}{2}(Q(\mathbf{a} + \mathbf{b}) - Q(\mathbf{a}) - Q(\mathbf{b})). \quad (6.2)$$

**Definition 6.1.** In the case of vector space  $\mathbb{R}^n$  with bilinear form  $B$  the inner product is defined as

$$\mathbf{e}_i \cdot \mathbf{e}_j = B(\mathbf{e}_i, \mathbf{e}_j). \quad (6.3)$$

For the definition of exterior product we will use the definition from exterior algebra, Definition 2.18. Adding these two products together gives rise to the so called geometric product [8].

**Definition 6.2** (Geometric product). Geometric product, denoted  $\mathbf{e}_i \mathbf{e}_j$ , of two basis vectors is a combination of inner and outer product

$$\mathbf{e}_i \mathbf{e}_j = \mathbf{e}_i \cdot \mathbf{e}_j + \mathbf{e}_i \wedge \mathbf{e}_j. \quad (6.4)$$

It is important to notice all the useful properties, that the geometric product has. These properties will be widely used, because they allow us to simplify the calculation.

**Remark 6.1.** Since the inner product of two orthonormal elements is equal to zero, see (6.1), than the geometric product of two orthonormal elements reduces to their exterior product,

$$\mathbf{e}_i \mathbf{e}_j = \mathbf{e}_i \wedge \mathbf{e}_j. \quad (6.5)$$

Similar behaviour has also the wedge product. When acting on a basis elements we know, that the wedge product of the element with itself is equal to zero. Therefore can be derived, that the geometric product of two orthonormal basis elements reduces to the inner product which is either positive or negative one,

$$\mathbf{e}_i \mathbf{e}_i = \mathbf{e}_i \cdot \mathbf{e}_i = \begin{cases} 1 & \text{if } i = j = 1, \dots, p \\ -1 & \text{if } i = j = p + 1, \dots, m \end{cases} \quad (6.6)$$

These definitions are now extended to the general grade  $r$  blade. For the inner product we get the form

$$\mathbf{e}_j \cdot \mathbf{e}_A = \sum_{k=1}^r (-1)^k B(\mathbf{e}_j, \mathbf{e}_{i_k}) \mathbf{e}_{A \setminus \{i_k\}}. \quad (6.7)$$

For the outer product we obtain

$$\mathbf{e}_j \wedge \mathbf{e}_A = \begin{cases} \mathbf{e}_j \wedge \mathbf{e}_{i_1} \wedge \dots \wedge \mathbf{e}_{i_r} & \text{if } j \notin A \\ 0 & \text{if } j \in A. \end{cases} \quad (6.8)$$

And for the geometric product we simply define

$$\mathbf{e}_i \mathbf{e}_A = \mathbf{e}_j \cdot \mathbf{e}_A + \mathbf{e}_j \wedge \mathbf{e}_A. \quad (6.9)$$

The algebra created is called real geometric algebra with associated geometric product, also called a Clifford product. It is usually denoted as  $\mathbb{G}_{p,q}$ . This algebra is actually a well known Clifford algebra, Definition 2.15. The quadratic form generating the ideal is the one associated with the bilinear form, (6.2). Knowing, that the constructed Geometric algebra is in fact a Clifford algebra gives an access to all the properties the Clifford algebra has. This algebra is associative and also free, meaning it can be generated from the basis elements using the geometric product.

The inner and exterior product can be viewed as a symmetric and antisymmetric product,  $\mathbf{a} \cdot \mathbf{b} = \frac{1}{2}(\mathbf{ab} + \mathbf{ba})$  and  $\mathbf{a} \wedge \mathbf{b} = \frac{1}{2}(\mathbf{ab} - \mathbf{ba})$ . These properties can be derived directly from the definition of these products, and then are just written in the form of a geometric product.

## 6.2 Complexification

This chapter uses many definitions from the [5]. It expands the concepts, relates them with the ones already defined and gives many examples for a good understanding.

In order to represent the complex nature of qubits it is necessary to introduce complex numbers. We set the ground field to  $\mathbb{C}$ , but that is not enough because we need to bring the complex behaviour directly into the algebra. That can be achieved by defining a specific orthogonal linear

transformation and also setting the basis of the vector space to the even number. Consider the linear orthogonal transformation  $J : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$  such that  $J^2 = -1$ , where 1 is the identity map. The action of  $J$  on the orthonormal basis  $(\mathbf{e}_1, \dots, \mathbf{e}_{2n})$  is following [5]:

$$J(\mathbf{e}_j) = -\mathbf{e}_{j+n}, \quad J(\mathbf{e}_{j+n}) = \mathbf{e}_j, \quad (6.10)$$

where  $j = 1, \dots, n$ . With the use of  $J$ , we will now introduce new transformed basis, which will become very elegant in the context of QC [5].

**Definition 6.3.** Witt basis

$$\begin{aligned} f_j &= \frac{1}{2}(1 + iJ)(\mathbf{e}_j) = \frac{1}{2}(\mathbf{e}_j - i\mathbf{e}_{j+n}), \quad j = 1, \dots, n \\ f_j^\dagger &= \frac{1}{2}(1 - iJ)(\mathbf{e}_j) = \frac{1}{2}(\mathbf{e}_j + i\mathbf{e}_{j+n}), \quad j = 1, \dots, n \end{aligned}$$

This definition is especially elegant because we can verify that for each  $j = 1, \dots, n$  is satisfied  $f_j^2 = 0$  and  $f_j^{\dagger 2} = 0$ . Indeed,

$$\begin{aligned} f_j^2 &= \frac{1}{2}(\mathbf{e}_j - i\mathbf{e}_{j+n})\frac{1}{2}(\mathbf{e}_j - i\mathbf{e}_{j+n}) \\ &= \frac{1}{4}(\mathbf{e}_j^2 - i\mathbf{e}_j\mathbf{e}_{j+n} - i\mathbf{e}_{j+n}\mathbf{e}_j + i^2\mathbf{e}_{j+n}^2) \\ &= \frac{1}{4}(1 - i\mathbf{e}_j\mathbf{e}_{j+n} + i\mathbf{e}_j\mathbf{e}_{j+n} - 1) = \frac{1}{4}(0) = 0. \end{aligned} \quad (6.11)$$

The choice of this particular  $f, f^\dagger$  is also motivated by the fact, that it satisfies the Grassman and duality identities:

$$\begin{aligned} ff^\dagger + f^\dagger f &= 1, \\ f_j f_k + f_k f_j &= f_j^\dagger f_k^\dagger + f_k^\dagger f_j^\dagger, \quad j, k = 1, \dots, n, \\ f_j f_k^\dagger + f_k^\dagger f_j &= \delta_{jk}, \quad j, k = 1, \dots, n, \end{aligned} \quad (6.12)$$

where the symbol  $\delta_{jk}$  is the Kronecker delta.

An easy way, how to construct the basis of the whole Clifford algebra  $\mathbb{C}_{2n}$  is with the use of geometric product. we can use the Grassmann blades of the Witt elements. The construction is following;

$$\begin{aligned} f_j f_k &= f_j \cdot f_k + f_j \wedge f_k = f_j \wedge f_k \\ f_j^\dagger f_k^\dagger &= f_j^\dagger \cdot f_k^\dagger + f_j^\dagger \wedge f_k^\dagger = f_j^\dagger \wedge f_k^\dagger \\ f_j f_k^\dagger &= f_j \cdot f_k^\dagger + f_j \wedge f_k^\dagger = \frac{1}{2}\delta_{jk} + f_j \wedge f_k^\dagger \end{aligned} \quad (6.13)$$

To extend the Witt basis to the whole algebra  $\mathbb{C}_{2n}$  we just take  $2^{2n}$  possible geometric products of

$$(f_1)^{i_1} (f_1^\dagger)^{j_1} \dots (f_n)^{i_n} (f_n^\dagger)^{j_n}, \quad i_k, j_k \in \{0, 1\} \text{ for } k = 1, \dots, n. \quad (6.14)$$

### 6.3 Qubit in GA

As we have mentioned earlier, a qubit will be represented by an element lying in the complex Clifford algebra  $\mathbb{C}_2$ . We can use the Witt basis vectors  $(f, f^\dagger)$  to induce a basis for the complex Clifford algebra. The basis can be then written in the form  $(1, f, f^\dagger, ff^\dagger)$ . Before associating qubit

with a particular element we will discuss the properties of the algebra in more details. The choice of the basis leads us to two primitive idempotents in this algebra, namely  $I = ff^\dagger$  and  $K = f^\dagger f$ . Conventions in physics, which won't be discussed in details, tells us to use the idempotent  $I$  for the construction. With this choice we come to the following representation of  $|0\rangle$  and  $|1\rangle$  state of qubit.

$$\begin{aligned} |0\rangle &= I = ff^\dagger \\ |1\rangle &= f^\dagger I = f^\dagger ff^\dagger = (1 - ff^\dagger)f^\dagger = f^\dagger \end{aligned} \quad (6.15)$$

We can now come to our definition of qubit Definition 4.1 and plug in our representation of  $|0\rangle$  and  $|1\rangle$  states:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \alpha ff^\dagger + \beta f^\dagger = (\alpha + \beta f^\dagger)I. \quad (6.16)$$

## 6.4 Gates in GA

For the rest of this thesis we will stick to this representation of  $|0\rangle$  and  $|1\rangle$  states. We can now consider the construction of gates. The basic gates can be obtained easily by taking into consideration their action on basis states. To distinguish between matrix and GA representation we will use different notation [5].

$$\begin{aligned} X - gate : \quad \lambda_X &= f^\dagger + f \\ Y - gate : \quad \lambda_Y &= if^\dagger - if \\ Z - gate : \quad \lambda_Z &= ff^\dagger - f^\dagger f \end{aligned} \quad (6.17)$$

Following example should show, that the construction is correct and when acting on basis states the results are correct.

### Example 6.1.

$$\begin{aligned} \lambda_X |0\rangle &= (f^\dagger + f)ff^\dagger = f^\dagger ff^\dagger = (1 - ff^\dagger)f^\dagger = f^\dagger = |1\rangle \\ \lambda_X |1\rangle &= (f^\dagger + f)f^\dagger = ff^\dagger = |0\rangle \end{aligned}$$

$$\begin{aligned} \lambda_Y |0\rangle &= (if^\dagger - if)f^\dagger f = -if^\dagger ff^\dagger = -i(1 - ff^\dagger)f^\dagger = -if^\dagger = |1\rangle \\ \lambda_Y |1\rangle &= (if^\dagger - if)f^\dagger = iff^\dagger = |0\rangle \end{aligned}$$

The form of the  $Z$ -gate in GA was given. But we know that, the  $Z$ -gate is just a special case of the more general phase gate. In the language of GA, this gate has the following form [5]:

$$Phase\ gate : \quad \lambda_P = ff^\dagger - e^{i\phi} f^\dagger f. \quad (6.18)$$

Concrete phase gates, that we have discussed earlier are obtained similarly. Parameter  $\phi$  is changed and different gates arise. And for  $\phi = \pi$  we get the  $Z$ -gate. The last important gate, we have not discussed yet is the Hadamard gate. The construction is not intuitive, that is why we go straight to the final form [5].

$$\lambda_H = \frac{1}{\sqrt{2}}(ff^\dagger - f^\dagger f + f + f^\dagger) \quad (6.19)$$

All the basic gates are defined and we can come to the conclusion, that every gate can be written as

$$\lambda = aff^\dagger + bf + cf^\dagger + df^\dagger f, \quad (6.20)$$

where  $a, b, c, d \in \mathbb{C}$  such that  $a^2 + c^2 = b^2 + d^2 = 1$ ,  $\bar{b}a + \bar{d}c = 0$  is satisfied.

These constraints noticeably resemble the conditions we specified, when defining general matrix for gate representation. It is no coincidence and it also shows the equivalence between these two types of representation. We can even express the relation

$$aff^\dagger + bf + cf^\dagger + df^\dagger f \longleftrightarrow \begin{pmatrix} a & b \\ c & d \end{pmatrix}. \quad (6.21)$$

This relation is now examined on examples. The matrix representation of all NOT gates is written down. The coefficients are obtained and then the GA representation derived.

**Example 6.2.**

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad a = 0, b = 1, c = 1, d = 0$$

Now using the formula (6.21) we can write the GA representation

$$\lambda_X = 0ff^\dagger + 1f + 1f^\dagger + 0f^\dagger f = f^\dagger + f$$

which is truly the form we have obtained previously. For other NOT gates the process is analogical.

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad a = 0, b = -i, c = i, d = 0$$

$$\lambda_Y = 0ff^\dagger - if + if^\dagger + 0f^\dagger f = if^\dagger - if$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad a = 1, b = 0, c = 0, d = -1$$

$$\lambda_Z = 1ff^\dagger + 0f^\dagger + 0f - 1f^\dagger f = ff^\dagger - f^\dagger f$$

The general phase gate will be constructed in the same way. The last important single qubit gate is the Hadamard gate. The construction will not differ, but we just have to be careful with the  $\frac{1}{\sqrt{2}}$  factor.

**Example 6.3.**

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad a = \frac{1}{\sqrt{2}}, b = \frac{1}{\sqrt{2}}, c = \frac{1}{\sqrt{2}}, d = -\frac{1}{\sqrt{2}}$$

$$\lambda_H = \frac{1}{\sqrt{2}}ff^\dagger + \frac{1}{\sqrt{2}}f + \frac{1}{\sqrt{2}}f^\dagger - \frac{1}{\sqrt{2}}f^\dagger f = \frac{1}{\sqrt{2}}(ff^\dagger + f + f^\dagger - f^\dagger f)$$

Which is again the form we have already obtained. This mechanism for transforming matrix representation into GA representation can be used for any given gate. And it of course works both ways. We have simplified matrix multiplication to normal multiplication, where we can use Grassman or duality identities to even further simplify the computation.



## 6.5 Multiple qubits and gates in GA

Let us just recall, that for the basis of  $\mathbb{C}_{2^n}$  complex algebra we use the elements  $(f_1, f_1^\dagger, \dots, f_n, f_n^\dagger)$  and construct  $2^{2^n}$  combinations of their geometric product. This leads to two primitive idempotents  $I$  and  $K$ . The idempotent  $I$  will be again used

$$I = I_1 \dots I_n = f_1 f_1^\dagger \dots f_n f_n^\dagger, \quad (6.22)$$

where  $I_i = f_i f_i^\dagger$ .

The definition of  $n$ -qubit state will be now introduced. The realisation is done in the space of dimension  $N = 2^{2^n}$  with the orthonormal basis

$$|i_1 \dots i_n\rangle = (f_1^\dagger)^{i_1} \dots (f_n^\dagger)^{i_n} I, \quad (6.23)$$

where  $i_1, \dots, i_n \in \{0, 1\}$ .

**Example 6.4.** Consider the case of 3-qubit state, meaning we work in the algebra  $\mathbb{C}_6$  with dimension  $2^6 = 64$ . Primitive idempotent  $I = f_1 f_1^\dagger f_2 f_2^\dagger f_3 f_3^\dagger$  is defined. Basis states can be then written in the form

$$\begin{aligned} |000\rangle &= I, & |100\rangle &= f_1^\dagger I, & |010\rangle &= f_2^\dagger I, & |001\rangle &= f_3^\dagger I, \\ |110\rangle &= f_1^\dagger f_2^\dagger I, & |101\rangle &= f_1^\dagger f_3^\dagger I, & |011\rangle &= f_2^\dagger f_3^\dagger I, & |111\rangle &= f_1^\dagger f_2^\dagger f_3^\dagger I. \end{aligned}$$

Multiple qubit gates will be now constructed. Basic gates will be written in the matrix form. The C-NOT gate will serve as an example. The matrices representing gates are usually sparse. Therefore it is not difficult to write the matrix in the form of sum, where every matrix has just one nonzero element.

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Every matrix in the sum is now written in the form of outer products. And outer products can be translated to GA very simply. We also know that the state  $\langle i_1 i_2 |$  is just a Hermitian conjugation of state  $|i_1 i_2\rangle$ . Hence we can derive

$$\begin{aligned} |00\rangle \langle 00| + |01\rangle \langle 01| + |11\rangle \langle 10| + |10\rangle \langle 11| &= II^\dagger + f_2^\dagger II^\dagger f_2 + f_1^\dagger f_2^\dagger II f_1 + f_1^\dagger II^\dagger f_2 f_1 = \\ &= I + f_2^\dagger I f_2 + f_1^\dagger f_2^\dagger I f_1 + f_1^\dagger I f_2 f_1 = \\ &= f_1 f_1^\dagger f_2 f_2^\dagger + f_1 f_1^\dagger f_2^\dagger f_2 - f_1^\dagger f_1 f_2^\dagger - f_1^\dagger f_1 f_2 = \\ &= f_1 f_1^\dagger - f_1^\dagger f_1 (f_2^\dagger + f_2) \end{aligned}$$

The form of other multiple qubit gates is obtained in the same way.

$$\begin{aligned} \lambda_{\text{C-NOT}} &= f_1 f_1^\dagger - f_1^\dagger f_1 (f_2^\dagger + f_2) \\ \lambda_{\text{CZ}} &= f_1 f_1^\dagger + f_1^\dagger f_1 (f_2 f_2^\dagger - f_2^\dagger f_2) \\ \lambda_{\text{SWAP}} &= f_1 f_1^\dagger f_2 f_2^\dagger + f_1^\dagger f_1 f_2^\dagger f_2 + f_1^\dagger f_2 - f_1 f_2^\dagger \end{aligned} \quad (6.24)$$

This construction can be used for any given gate. Matrix representation is written in the form of outer products and then translated into GA.

## 6.6 Parallel gates

Similarly to the matrix representation, also in the GA representation needs to be discussed how to represent parallel gates acting on the multiple qubit. Tensor product will be used.

**Definition 6.4** (Gates tensor product). A tensor product  $\lambda_1 \otimes \dots \otimes \lambda_n$ , where  $\lambda_k \in \{f_k f_k^\dagger, f_k^\dagger f_k, f_k, f_k^\dagger\}$  for each  $k = 1, \dots, n$ , is represented by a gemotric product  $(-1)^s \lambda_1 \dots \lambda_n$ , where the sign is determined by the cardinality of the sets  $S_i$ , such that  $s = \sum_i |S_i|$ , where

$$S_i = \left\{ l < i : \lambda_l = f_l \text{ or } \lambda_l = f_l^\dagger f_l \right\} \text{ in the case if } \lambda_i = f_i \text{ or } \lambda_i = f_i^\dagger.$$

Although this definition might seem a bit complicated at the first glance. Examples are provided for a proper understanding.

**Example 6.5.**

$$f_1 \otimes f_2^\dagger = (-1)^s f_1 f_2^\dagger \quad s = \sum_i |S_i|$$

$$S_1 = \emptyset, \quad S_2 = \{1\}, \quad S_3 = \emptyset; \quad S_i \text{ is empty for all } i > 2$$

$$s = 1; \quad \text{leading to: } f_1 \otimes f_2^\dagger = (-1)^1 f_1 f_2^\dagger = -f_1 f_2^\dagger$$

This definition of a tensor product is not commutative. For  $\lambda_k \in \{f_k f_k^\dagger, f_k^\dagger f_k, f_k, f_k^\dagger\}$  is  $\lambda_i \otimes \lambda_j$  not necessarily equal to  $\lambda_j \otimes \lambda_i$ .

**Example 6.6.**

$$f_1^\dagger \otimes f_2 = (-1)^s f_1^\dagger f_2 \quad s = \sum_i |S_i|$$

$$S_1 = \emptyset, \quad S_2 = \emptyset, \quad S_3 = \emptyset; \quad S_i \text{ is empty for all } i > 2$$

$$s = 1; \quad \text{leading to: } f_1^\dagger \otimes f_2 = (-1)^0 f_1^\dagger f_2 = f_1^\dagger f_2$$

Now consider two parallel gates. Knowing the rules of the tensor product, it is easy to calculate a particular example. It is important to keep in mind, that the tensor product is not commutative and the order of the gates matter.

**Example 6.7.**

$$\begin{aligned} \lambda_X \otimes \lambda_Y &= (f_1^\dagger + f_1) \otimes (i f_2^\dagger + i f_2) \\ &= i((f_1^\dagger \otimes f_2^\dagger - f_1^\dagger \otimes f_2 + f_1 \otimes f_2^\dagger - f_1 \otimes f_2)) \\ &= i(f_1^\dagger f_2^\dagger - f_1^\dagger f_2 - f_1 f_2^\dagger + f_1 f_2) \end{aligned}$$

The elegance of this approach lays in the fact, that the multiple qubit gates can be obtained via tensor product and they do not have to be derived from their matrix representation. The CNOT gate was derived from the matrix representation. Now will be shown, how the same result can be deduced from the tensor product. The idempotent  $I$  does not change the sign in contrast to the idempotent  $K$ . Their proper combination leads to the following example.

**Example 6.8.**

$$\begin{aligned} \lambda_{CNOT} &= I_1 \otimes 1 + K_1 \otimes X_2 \\ &= f_1 f_1^\dagger \otimes 1 + f_1^\dagger f_1 \otimes (f_2^\dagger + f_2) \\ &= f_1 f_1^\dagger + f_1^\dagger f_1 \otimes f_2^\dagger + f_1^\dagger f_1 \otimes f_2 \\ &= f_1 f_1^\dagger - f_1^\dagger f_1 f_2^\dagger - f_1^\dagger f_1 f_2 \\ &= f_1 f_1^\dagger - f_1^\dagger f_1 (f_2^\dagger + f_2) \end{aligned}$$

Which after some modification truly is the desired form of CNOT gate. In this case we supposed, that the first qubit is the control qubit. The construction for the case, when the control qubit is the second one is analogical. Other gates can be obtained in the same manner.

**Example 6.9.**

$$\begin{aligned}
 \lambda_{SWAP} &= I_1 \otimes 1 + K_1 \otimes Z_2 \\
 &= f_1 f_1^\dagger \otimes 1 + f_1^\dagger f_1 \otimes (f_2 f_2^\dagger + f_2^\dagger f_2) \\
 &= f_1 f_1^\dagger + f_1^\dagger f_1 \otimes f_2 f_2^\dagger + f_1^\dagger f_1 \otimes f_2^\dagger f_2 \\
 &= f_1 f_1^\dagger - f_1^\dagger f_1 f_2 f_2^\dagger - f_1^\dagger f_1 f_2^\dagger f_2 \\
 &= f_1 f_1^\dagger - f_1^\dagger f_1 (f_2 f_2^\dagger - f_2^\dagger f_2)
 \end{aligned}$$

## 6.7 Measurement

The last thing, that needs to be introduced is the measurement. For calculating the probability of certain outcome in matrix representation we just took the inner product of the outcome with the transformed qubit. And the square root of the product was the particular probability. In the GA will be used a so called scalar projection, denoted  $[ ]_0$ . The scalar projection takes just the scalar part of a multivector. We can than use the following formula

$$\langle \rho | \psi \rangle = 2^n [ \rho^\dagger \psi ]_0 \quad (6.25)$$

where  $n$  is the number of qubits. This factor needs to be introduced due to our choice of basis and due to the spinorial nature of the representation, which will not be discussed in more details. Let us see on an example how this scalar projection works.

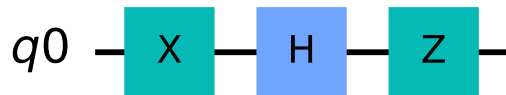
$$\begin{aligned}
 \langle 0|1 \rangle &= 2^1 [ f f^\dagger f^\dagger ]_0 = 0 \\
 \langle 0|0 \rangle &= 2^1 [ f f^\dagger ]_0 = 1
 \end{aligned} \quad (6.26)$$

The outcomes are as expected. And just a quick recall that  $f f^\dagger = \frac{1}{2} + f \wedge f^\dagger$  and therefore  $[ f f^\dagger ]_0 = \frac{1}{2}$ .

## 6.8 Circuit example

In this chapter will be given some simple examples how to represent the circuits in GA. We will start with the serial setup, where we do not encounter the tensor product. We will always compare the matrix representation and GA representation.

**Example 6.10** (Serial gates). The first example shows the circuit with all gates on one wire, see Figure 6.1.



**Figure 6.1:** Serial gates

We will start with the matrix representation. All matrices are multiplied and we obtain a matrix representing the circuit.

$$M = ZHX = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \quad (6.27)$$

For any given qubit can be now calculated the probability of the qubit ending in the basis state. The qubit is multiplied by the final matrix and then is taken the inner product with each basis state. We will use the basis qubits  $|\psi\rangle = |0\rangle$  and  $|\rho\rangle = |1\rangle$ . We compute the transformed qubit and then take the inner product with basis states to obtain the probabilities

$$|\psi'\rangle = M|\psi\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad (6.28)$$

$$|\rho'\rangle = M|\rho\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (6.29)$$

$$p(|0\rangle) = |\langle 0|\psi'\rangle|^2 = \left| \frac{1}{\sqrt{2}} (1 \ 0) \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right|^2 = \left| -\frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2} \quad (6.30)$$

$$p(|1\rangle) = |\langle 1|\psi'\rangle|^2 = \left| \frac{1}{\sqrt{2}} (0 \ 1) \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right|^2 = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2} \quad (6.31)$$

$$p(|0\rangle) = |\langle 0|\rho'\rangle|^2 = \left| \frac{1}{\sqrt{2}} (1 \ 0) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right|^2 = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2} \quad (6.32)$$

$$p(|1\rangle) = |\langle 1|\rho'\rangle|^2 = \left| \frac{1}{\sqrt{2}} (0 \ 1) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right|^2 = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2} \quad (6.33)$$

For both vectors the sum of probabilities is 1. The constraint given by the second axiom of probability is satisfied and we can come to the conclusion, that our computation is correct.

$$p(|0\rangle) + p(|1\rangle) = \frac{1}{2} + \frac{1}{2} = 1$$

Now the alternative is shown. The circuit is represented in the apparatus of geometric algebra.

$$\begin{aligned} \lambda_M &= \lambda_Z \lambda_H \lambda_X = (ff^\dagger - f^\dagger f) \frac{1}{\sqrt{2}} (ff^\dagger - f^\dagger f + f + f^\dagger) (f^\dagger + f) \\ &= \frac{1}{\sqrt{2}} (ff^\dagger ff^\dagger - ff^\dagger f^\dagger f + ff^\dagger f + ff^\dagger f^\dagger) (f^\dagger + f) \\ &= \frac{1}{\sqrt{2}} (ff^\dagger + f + f^\dagger f - f^\dagger) (f^\dagger + f) \\ &= \frac{1}{\sqrt{2}} (ff^\dagger f^\dagger + ff^\dagger f + ff^\dagger + ff = f^\dagger ff^\dagger + f^\dagger ff - f^\dagger f^\dagger - f^\dagger f) \\ &= \frac{1}{\sqrt{2}} (f^\dagger - f^\dagger f + f + ff^\dagger) \end{aligned} \quad (6.34)$$

This expression can be thought as an equivalent to the final matrix obtained in the matrix representation. Now will be calculated the transformed qubits of  $|\psi\rangle = |0\rangle$  and  $|\rho\rangle = |1\rangle$ .

$$\begin{aligned}
 |\psi'\rangle &= \lambda_M |\psi\rangle = \frac{1}{\sqrt{2}}(f^\dagger - f^\dagger f + f + f f^\dagger)(f f^\dagger) \\
 &= \frac{1}{\sqrt{2}}(f^\dagger f f^\dagger - f^\dagger f f f^\dagger + f f f^\dagger + f f^\dagger f f^\dagger) \\
 &= \frac{1}{\sqrt{2}}(f^\dagger + f f^\dagger)
 \end{aligned} \tag{6.35}$$

$$\begin{aligned}
 |\rho'\rangle &= \lambda_M |\rho\rangle = \frac{1}{\sqrt{2}}(f^\dagger - f^\dagger f + f + f f^\dagger)f^\dagger \\
 &= \frac{1}{\sqrt{2}}(f^\dagger f^\dagger - f^\dagger f f^\dagger + f f^\dagger + f f^\dagger f^\dagger) \\
 &= \frac{1}{\sqrt{2}}(f f^\dagger - f^\dagger)
 \end{aligned}$$

The probabilities of the qubit ending in particular basis state are yet to be calculated.

$$\begin{aligned}
 p(|0\rangle) &= |\langle 0|\psi'\rangle|^2 = \left| 2 \left[ f f^\dagger \frac{1}{\sqrt{2}}(f^\dagger + f f^\dagger) \right]_0 \right|^2 = \left| \frac{2}{\sqrt{2}} [f f^\dagger f^\dagger + f f^\dagger f f^\dagger]_0 \right|^2 \\
 &= \left| \frac{2}{\sqrt{2}} [f f^\dagger]_0 \right|^2 = \left| \frac{2}{\sqrt{2}} \frac{1}{2} \right|^2 = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2}
 \end{aligned} \tag{6.36}$$

Other probabilities are calculated in the same way, therefore only a short version will be shown.

$$p(|1\rangle) = |\langle 1|\psi'\rangle|^2 = \left| 2 \left[ f \frac{1}{\sqrt{2}}(f^\dagger + f f^\dagger) \right]_0 \right|^2 = \frac{1}{2} \tag{6.37}$$

$$p(|0\rangle) = |\langle 0|\rho'\rangle|^2 = \left| 2 \left[ f f^\dagger \frac{1}{\sqrt{2}}(f f^\dagger - f^\dagger) \right]_0 \right|^2 = \frac{1}{2} \tag{6.38}$$

$$p(|1\rangle) = |\langle 1|\rho'\rangle|^2 = \left| 2 \left[ f \frac{1}{\sqrt{2}}(f f^\dagger - f^\dagger) \right]_0 \right|^2 = \frac{1}{2} \tag{6.39}$$

Now can be verified, that the sum of probabilities is 1 for both vectors. Probabilities calculated via GA representation are same as the probabilities calculated via matrix representation. GA representation proves to be correct.

**Example 6.11** (Parallel gates). In this example is examined the behaviour of parallel gates. On one wire is Hadamard gate and on second wire is the X-gate. The CNOT gate is then applied with the first qubit being a control qubit, see Figure 6.2.

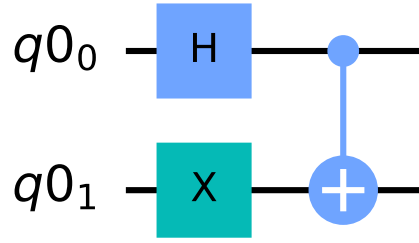


Figure 6.2: CNOT gate circuit

The matrix representation will be shown first. The tensor product of Hadamard gate and X-gate is taken. The form of CNOT gate is known, 6.24.

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (6.40)$$

$$\begin{aligned} H \otimes X &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 * \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & 1 * \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ 1 * \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & -1 * \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{pmatrix} \end{aligned} \quad (6.41)$$

The final matrix is obtained by multiplying CNOT gate with  $H \otimes X$ .

$$M = \text{CNOT}(H \otimes X) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \quad (6.42)$$

We will demonstrate the action of the circuit on two example multiple qubits, namely  $\psi = |00\rangle$  and  $\rho = |01\rangle$ .

$$|\psi'\rangle = M|\psi\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \quad (6.43)$$

$$|\rho'\rangle = M|\rho\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (6.44)$$

We obtained the transformed qubits and the measurement can be carried out.

$$\begin{aligned}
p(|00\rangle) &= |\langle 00|\psi'\rangle|^2 = \left| \frac{1}{\sqrt{2}} (1 \ 0 \ 0 \ 0) \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \right|^2 = 0 \\
p(|01\rangle) &= |\langle 01|\psi'\rangle|^2 = \left| \frac{1}{\sqrt{2}} (0 \ 1 \ 0 \ 0) \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \right|^2 = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2} \\
p(|10\rangle) &= |\langle 01|\psi'\rangle|^2 = \left| \frac{1}{\sqrt{2}} (0 \ 0 \ 1 \ 0) \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \right|^2 = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2} \\
p(|11\rangle) &= |\langle 11|\psi'\rangle|^2 = \left| \frac{1}{\sqrt{2}} (0 \ 0 \ 0 \ 1) \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \right|^2 = 0
\end{aligned} \tag{6.45}$$

The measurement for the second qubit will be shorten

$$\begin{aligned}
p(|00\rangle) &= |\langle 00|\rho'\rangle|^2 = \frac{1}{2}, & p(|01\rangle) &= |\langle 01|\rho'\rangle|^2 = 0, \\
p(|10\rangle) &= |\langle 10|\rho'\rangle|^2 = 0, & p(|11\rangle) &= |\langle 11|\rho'\rangle|^2 = \frac{1}{2}.
\end{aligned} \tag{6.46}$$

We can easily verify that the sum of probabilities is equal to 1 for both multiple qubits.

The same circuit is now realized in the language of GA. The tensor product of the Hadamard gate and X-gate is taken. The form of the CNOT gate is known, (6.24).

$$\begin{aligned}
\lambda_{HX} &= \lambda_H \otimes \lambda_X = \frac{1}{\sqrt{2}} (f_1 f_1^\dagger - f_1^\dagger f_1 + f_1 + f_1^\dagger) (f_2^\dagger + f_2) \\
&= \frac{1}{\sqrt{2}} (f_1 f_1^\dagger f_2^\dagger + f_1 f_1^\dagger f_2 + f_1^\dagger f_1 f_2^\dagger + f_1^\dagger f_1 f_2 - f_1 f_2^\dagger - f_1 f_2 + f_1^\dagger f_2^\dagger + f_1^\dagger f_2) \tag{6.47}
\end{aligned}$$

$$\lambda_{CNOT} = f_1 f_1^\dagger - f_1^\dagger f_1 f_2^\dagger - f_1^\dagger f_1 f_2$$

The final circuit representation is obtained simply by multiplying the  $\lambda_{HX}$  and  $\lambda_{CNOT}$ . Using the rules mentioned in (6.11), (6.12) and (6.13) we can simplify to the final form

$$\begin{aligned}
\lambda_M &= \lambda_{CNOT} \lambda_{HX} \\
&= \frac{1}{\sqrt{2}} (f_1 f_1^\dagger f_2^\dagger + f_1 f_1^\dagger f_2 - f_1 f_2^\dagger - f_1 f_2 - f_1^\dagger f_1 f_2^\dagger f_2 + f_1^\dagger f_2^\dagger f_2 + f_1^\dagger f_1 f_2 f_2^\dagger + f_1^\dagger f_2 f_2^\dagger). \tag{6.48}
\end{aligned}$$

Note that in the matrix representation the final matrix has 8 nonzero elements. The GA representation has also 8 elements. To verify the correctness we will calculate the probabilities of same qubits  $\psi = |00\rangle$  and  $\rho = |01\rangle$ .

$$\begin{aligned}
|\psi'\rangle &= \lambda_M |\psi\rangle \\
&= \frac{1}{\sqrt{2}}(f_1 f_1^\dagger f_2^\dagger + f_1 f_1^\dagger f_2 - f_1 f_2^\dagger - f_1 f_2 - f_1^\dagger f_1 f_2^\dagger f_2 + f_1^\dagger f_2^\dagger f_2 + f_1^\dagger f_1 f_2 f_2^\dagger + f_1^\dagger f_2 f_2^\dagger)(f_1 f_1^\dagger f_2 f_2^\dagger) \\
&= \frac{1}{\sqrt{2}}(f_1 f_1^\dagger f_2^\dagger + f_1^\dagger f_2 f_2^\dagger) \\
|\rho'\rangle &= \lambda_M |\rho\rangle \\
&= \frac{1}{\sqrt{2}}(f_1 f_1^\dagger f_2^\dagger + f_1 f_1^\dagger f_2 - f_1 f_2^\dagger - f_1 f_2 - f_1^\dagger f_1 f_2^\dagger f_2 + f_1^\dagger f_2^\dagger f_2 + f_1^\dagger f_1 f_2 f_2^\dagger + f_1^\dagger f_2 f_2^\dagger)(f_1 f_1^\dagger f_2 f_2^\dagger) \\
&= \frac{1}{\sqrt{2}}(f_1 f_1^\dagger f_2 f_2^\dagger + f_1^\dagger f_2^\dagger)
\end{aligned} \tag{6.49}$$

To calculate the probabilities of particular outcomes we use the scalar projection. It is important to change the factor before the scalar projection. This example works with two qubits, therefore we know  $2^n = 2^2 = 4$ .

$$\begin{aligned}
p(|00\rangle) &= |\langle 00|\psi'\rangle|^2 = \left| 4 \left[ f_1 f_1^\dagger f_2 f_2^\dagger \frac{1}{\sqrt{2}}(f_1 f_1^\dagger f_2^\dagger + f_1^\dagger f_2 f_2^\dagger) \right]_0 \right|^2 = \left| \frac{4}{\sqrt{2}} [0]_0 \right|^2 = 0 \\
p(|01\rangle) &= |\langle 00|\psi'\rangle|^2 = \left| 4 \left[ f_1 f_1^\dagger f_2 \frac{1}{\sqrt{2}}(f_1 f_1^\dagger f_2^\dagger + f_1^\dagger f_2 f_2^\dagger) \right]_0 \right|^2 = \left| \frac{4}{\sqrt{2}} [f_1 f_1^\dagger f_2 f_2^\dagger]_0 \right|^2 \\
&= \left| \frac{4}{\sqrt{2}} \frac{1}{4} \right|^2 = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2}
\end{aligned} \tag{6.50}$$

$$\begin{aligned}
p(|10\rangle) &= |\langle 10|\psi'\rangle|^2 = \frac{1}{2} \\
p(|11\rangle) &= |\langle 11|\psi'\rangle|^2 = 0
\end{aligned}$$

The sum of all probabilities is 1, which is correct. We can now compare the results to the one obtained by matrix representation. All probabilities are same. For the second multiple qubit is the process analogical and just the final probabilities are shown.

$$\begin{aligned}
p(|00\rangle) &= |\langle 00|\rho'\rangle|^2 = \frac{1}{2} \\
p(|01\rangle) &= |\langle 01|\rho'\rangle|^2 = 0 \\
p(|10\rangle) &= |\langle 10|\rho'\rangle|^2 = 0 \\
p(|11\rangle) &= |\langle 11|\rho'\rangle|^2 = \frac{1}{2}
\end{aligned} \tag{6.51}$$

In the case of the second multiple qubit the probabilities also add up to 1. They are equal to the matrix representation and we can come to the conclusion, that the GA representation works correctly.



It can be clearly seen, that with the higher number of input qubits the dimension of the matrices increases exponentially. The computation becomes very demanding, because we are dealing with huge matrices. Representation in GA is very intuitive and allows for many simplifications during the calculation. The number of elements representing quantum states or quantum gates also increases, but they usually cancel out during the calculation and simplify the expression. It is also worth noticing, that the geometric product of many elements in GA is equal to 0. This property simplifies the calculation and also saves a lot of memory.

The biggest problem with GA representation is, that there is no software for a direct implementation. GAALOPWeb offers some opportunities but with many limitations. With proper implementation can be QC realised in the language of GA very efficiently. The equivalence of the matrix and GA representation also allows for using more suitable representation for particular problems.

## 7 Conclusion

The first chapter of this thesis defines key concepts from algebra and places them into context. Special attention is given to bilinear and quadratic forms and also to algebraic products. General description of tensor product and tensor spaces is given and then shown on examples. Clifford algebra is derived and all important properties are discussed, [6].

In order to understand the concept of qubit properly, the second chapter focuses on quantum physics. Mathematical description of the electromagnetic wave is given and is shown how to decompose the vector representing the wave. The contradiction between classical wave theory and quantum understanding is discussed. The chapter is mostly theoretical and works with the following sources mainly, [1, 10].

All the information is then used to define a qubit. The difference between a classical bit and a qubit is emphasized. In order to address the problem with a qubit visualisation, the number of degrees of freedom is reduced to two and then the definition of qubit is transformed to spherical coordinates. We then define so called Bloch sphere, which is used to visualise different qubits and their transformations. The definition of qubit is extended to multiple qubits using the Kronecker product. In a simple manner is also described how the measurement is carried out. This chapter is based on the knowledge acquired from [2, 7, 13].

Next chapter shows how to derive a general quantum gate and what properties must be satisfied. Basic gates are defined and their action is demonstrated on examples. To further illustrate how quantum gates work, the software Qiskit [11] is used for the visualisation. The definition of a single qubit gate is extended to a multiple qubits gate using the Kronecker product. Simple quantum circuit is then constructed and described with all necessities.

The main part applies the knowledge and constructs an apparatus of geometric algebra (GA) suitable for quantum computing. We start with the definition of GA over the real numbers. Special attention is paid to geometric product this algebra is equipped with. This product plays a crucial role throughout the thesis and therefore is carefully derived from the inner and exterior product. Important properties of geometric product are highlighted. However, in order to properly represent qubits in GA we need to bring complex numbers to our definition. Setting the ground field to the complex numbers is not enough, because the complex behaviour needs to be brought directly into the algebra. Therefore an orthogonal linear transformation is introduced. Basis written with the use of this transformation is called a Witt basis and plays very important role. We examine interesting properties of the Witt basis for the purpose of further computing. More information about this approach to the construction of GA can be found in [5].

With the suitable apparatus of GA we can come down to the definition of a qubit. The key idea is to identify qubits with the elements of GA. General definition of a qubit is given and then extended to multiple qubits. Quantum gates are also identified with the elements of GA and that allows for a computation within one algebraic structure. The relation between matrix and GA representation is highlighted. We then carefully derive the form of quantum gates written in GA. The universal way to construct any multiple qubit gate is shown. Everything in this chapter is accompanied with examples to provide an intuitive understanding of computing in GA. We also briefly discuss, how to perform the measurement in GA.

In the final part it is shown how to write a simple circuit in matrix and GA representation. The first example shows a serial gate circuit and the second one shows a parallel gate circuit. We also confirm the correctness of GA representation by comparing the results of measurement with the results obtained in the matrix representation.

We were able to construct an apparatus of GA, which allows for an intuitive and straightforward computing. All obtained results confirmed the correctness of the approach. An elegant definition of the basis allows many simplifications during the computing, making it faster and potentially

---

also less memory consuming. The matrix representation has a problem with the increase of the dimension of the matrices. This increase is exponential and even for a small number of qubits the computing can become very demanding. The dimension of the expressions representing quantum gates in GA also increases, but it is not so dramatic and can often be simplified. The biggest problem is the implementation. Complex numbers are difficult to represent and there is currently no software suitable for the computing in GA. GAALOPWeb offers an alternative, while using a slightly different approach and working with the GA over real numbers.

## References

- [1] BROWNE, M. E.; Schaum's outlines physics for engineering and Science; McGraw-Hill: New York, 2010.
- [2] DE LIMA MARQUEZINO, F.; et al. A Primer on Quantum Computing, SpringerBriefs in Computer Science, 2019.
- [3] DORST, L.; FONTIJNE, D.H.F. and MANN, S.; Geometric algebra for computer science: an object-oriented approach to geometry. Amsterdam: Elsevier ; Morgan Kaufmann, 2007, xxxv, 626 s. : il. ISBN 978-0-12-369465-2.
- [4] HILDENBRAND, D.; Introduction to Geometric Algebra Computing (1st ed.). Chapman and Hall/CRC. <https://doi.org/10.1201/9781315152172>.
- [5] HRDINA, J.; NÁVRAT A. and VAŠÍK, P.; Quantum computing based on complex Clifford algebras. 2022, [online]. Available on: <https://doi.org/10.1007/s1128-022-03648-w>.
- [6] LANE, M. S.; BIRKHOFF, G., Algebra; American Mathematical Society: Providence, 1999, ISBN 9780821816462
- [7] NIELSEN, M. A.; CHUANG, I. L.; Quantum Computation and Quantum Information; Cambridge University Press: Cambridge, 2010.
- [8] PERWASS, C.; Geometric Algebra with Applications in Engineering. 1. vyd. Springer, 2009. ISBN 978-3-540-89067-6.
- [9] PROCHÁZKA, L.; Inverzní kinematika robotického ramene s předepsanou trajektorií efektoru pomocí geometrické algebry. Brno, 2022.
- [10] PURCELL, E. M.; MORIN, D.; Electricity and magnetism; Cambridge University Press: Cambridge, 2018.
- [11] Open-source quantum development <https://qiskit.org/> (accessed Apr 28, 2023).
- [12] RUE, J.; XAMBO, S.; Mathematical essentials of quantum computing, Lecture notes UPC, <https://web.mat.upc.edu/sebastia.xambo/QC/qc.pdf>.
- [13] Theory of computing; ACM Press: New York, 1993.