# The Logic in Computer Science Column

by

Yuri Gurevich

Microsoft Research
One Microsoft Way, Redmond WA 98052, USA
gurevich@microsoft.com

# From Reversible Logic Gates to Universal Quantum Bases

Alex Bocharov, Krysta M. Svore[*]

## Abstract

With the anticipated end of Moore's law for integrated circuits [3, 4] fast approaching and continued advances in low-power electronics, interest in quantum computing has increased. This shifts the focus from deterministic logical circuits to potentially more powerful circuits based on controllable quantum systems. In this column, we present a mathematical tour of the quantum circuit model, beginning with reversible logic circuits and expanding to quantum circuits, gates, and measurement. We highlight quantum mechanical phenomena such as superposition, entanglement, and measurement, review the Gottesman-Knill theorem, which states that some subclasses of quantum operations can be simulated efficiently on a classical computer, and describe sets of quantum gates that are universal for quantum computation.

[*] Quantum Architectures and Computation Group, Microsoft Research, Redmond, WA 98052, USA. {alexeib,ksvore}@microsoft.com

*Bulletin of the EATCS no 110, pp. 52–75, June 2013*

# 1 Introduction

There is a growing list of problems for which a quantum algorithm delivers superpolynomial speedup over the corresponding classical algorithms. Most notably, *integer factorization* can be solved exponentially faster using a quantum algorithm than using the best-known classical algorithms [10]. Other problems include *Pell's equation* [11], computing the *unit group and class group* of a number field [12, 13], finding the *hidden shift* of a boolean function [14], solving *linear systems* of equations [15], *group order and membership* [16], *group isomorphism* [17], and *knot invariants* [18, 19]. These algorithms are based on the quantum circuit model of computation.

At the core of the quantum circuit model is *unitary evolution*, which by nature is physically reversible. According to Landauer's principle [2], in order for a computational process to be physically reversible it must be logically reversible. We begin in Section 2 with a short introductory discourse on reversible logic gates which are a special case of classical Boolean gates [5]. We then introduce in Section 3 controllable quantum states and observe that any reversible logic gate generates a unitary quantum gate, however the converse is not true. Quantum state spaces are incomparably richer than boolean logic, as is the hierarchy of controllable gates that we will present.

Since quantum gates are richer than boolean gates, it comes as a surprise that a subclass of quantum circuits can in fact be simulated efficiently on a classical computer [6]. This subclass is commonly called stabilizer circuits or *Clifford* circuits. In Section 4, we introduce the Gottesman-Knill theorem and review some of its key implications.

In Section 5, we introduce the notion of a *universal quantum basis*. After reviewing several such universal bases, we highlight a key result: the quantum analog of the classical *Toffoli* gate, with some help in the form of *measurement and classical feedback*, is universal when added to the group of Clifford circuits. Finally, we conclude in Section 6 with directions for future work.

# 2 Reversible Logic Gates

Consider an $n$-bit space $\{0,1\}^n$ and the complete set of Boolean functions of the form $f : \{0,1\}^n \rightarrow \{0,1\}^m$, where $m$ is a positive integer. It was realized very early that any Boolean function can be represented as a nested composition of *logic gates*. In fact, in 1881, C.S. Pierce claimed that with just *one* gate, a NOR, or alternatively a NAND, any Boolean function can be realized. For NAND, this was first proven by H.M. Sheffer in 1913 (see [20]). Nested compositions of logic gates are called *logic circuits*.

In this column, we focus on *reversible* Boolean functions. A reversible function $f$ is a computation that can be 'undone', that is, an arbitrary bit vector input $x$ can be reconstructed from the corresponding output vector $f(x)$ and so no input information is erased during the computation of $f$. We define a *reversible Boolean function* with $n$ arguments as a bijection $\{0, 1\}^n \to \{0, 1\}^n$. If we consider $\{0, 1\}^n$ as a set with $2^n$ elements, we can also say that such a bijection is an arbitrary permutation of $2^n$. Thus there is a one-to-one correspondence between reversible boolean functions and elements of the symmetric group $S_{2^n}$.

Interestingly, neither $\mathrm{NAND}(a, b) = \neg(a \wedge b)$ nor $\mathrm{NOR}(a, b) = \neg(a \vee b)$ is reversible. In fact, among those gates commonly appearing in disjunctive or conjunctive normal forms, only NOT: $\{0, 1\} \to \{0, 1\}$ is reversible. Naturally, for any $n$, the identity map $I_n$ is also reversible. We denote the identity gate by $I$.

A binary reversible gate that plays a key role in reversible (and subsequently in quantum) logic is the controlled-NOT gate, written as CNOT, which maps $\{0, 1\}^2 \to \{0, 1\}^2$ and is defined as $\mathrm{CNOT}(a, b) = (a, a \oplus b)$, where $\oplus$ is the exclusive OR (which can be alternatively viewed as addition modulo 2). Intuitively, the first input bit, or the *control* bit, controls the application of the NOT operation on the second bit, called the *target* bit. Evidently, when $a = 0$, $b$ remains unchanged, and when $a = 1$ the second bit is flipped.

We can use simple gates such as NOT and CNOT during synthesis of $n \times n$ reversible Boolean functions $\{0, 1\}^n \to \{0, 1\}^n$.

To this end, $\mathrm{NOT}[i], i = 1, \ldots, n$ denotes the NOT gate applied to the $i$-th input argument, i.e., $\mathrm{NOT}[i]$ replaces the $i$-th bit of the bit vector $(x_1, \ldots, x_n)$ with $\mathrm{NOT}(x_i)$.

We define $\mathrm{CNOT}[i, j], 1 \leq i, j \leq n, i \neq j$ as follows:

$\mathrm{CNOT}[i, j]$ replaces the $j$-th bit of the bit vector $(x_1, \ldots, x_n)$ with $x_i \oplus x_j$.

In general, given a $k \times k$ Boolean function $f : \{0, 1\}^k \to \{0, 1\}^k$, some $n \geq k$, and a *multi-index* $\mathbf{i} = i_1, \ldots, i_k, 1 \leq i_l \leq n, l = 1, \ldots k$, we define the *extension* $f[\mathbf{i}] : \{0, 1\}^n \to \{0, 1\}^n$ to $n$-bit logic as follows: if $f(x_{i_1}, \ldots, x_{i_k}) = (y_{i_1}, \ldots, y_{i_k})$ then $f[\mathbf{i}](x_1, \ldots x_n) = replace((x_1, \ldots x_n), x_{i_l}, y_{i_l}, l = 1, \ldots, k)$.

Throughout, we use the $\circ$ symbol to denote the composition of reversible Boolean functions of the same arity, and replace it with a single space when this does not lead to ambiguities. We also use the term 'wire' to refer to logical input bits in the computation.

We now consider several examples of function composition.

**Example 1.**

1. *$NOT[i]\, NOT[i] = I$.*

2. *$CNOT[i, j]\, CNOT[i, j] = I$.*

**Example 2.**

1. *We introduce the two-bit SWAP gate that swaps the two logical input bits:*

$$SWAP = CNOT[1,2]\, CNOT[2,1]\, CNOT[1,2]$$

.

2. *Verify that for $n \geq 2, 1 \leq i, j \leq n, i \neq j$,*

$$SWAP[i,j] = CNOT[i,j]\, CNOT[j,i]\, CNOT[i,j].$$

3. *Verify that $SWAP[i,j] = SWAP[j,i]$ and $SWAP[i,j]\, SWAP[i,j] = I$.*

Conjugated composition of a reversible function with SWAP is equivalent to re-indexing the arguments of that function.

**Example 3.**

1. $NOT[j] = SWAP[i,j]\, NOT[i]\, SWAP[i,j]$.

2. *By direct verification on the $i, j, k, l$ bits,*

$$CNOT[k,l] = SWAP[i,k] SWAP[j,l]\, CNOT[i,j]\, SWAP[j,l]\, SWAP[i,k].$$

Regarding individual bits as logical wires as in Example 3(2.), we can view the SWAP gate as a mechanism to move both the control and the target bits of a CNOT gate from wire to wire.

Given a set of elementary gates, we can define a $n$-bit logical *circuit* over that set of gates as a composition of a finite number of extensions of these gates to $n$-bit logic.

**Example 4.** *For $n = 3$ , $SWAP[1,2]\, SWAP[2,3]$ is a circuit implementing a cyclic permutation of bits in a 3-bit vector: $(x_1, x_2, x_3) \to (x_2, x_3, x_1)$.*

How do we implement a Boolean function with a logical circuit? Example 4 describes a specific implementation where the resulting function is a simple composition of all gates in the circuit.

In general, however, we may need to implement a function with a given number of arguments with a circuit of *greater* arity. Let $f : \{0,1\}^k \to \{0,1\}^k$ be a Boolean function, and suppose $n > k$. Consider the extension $f[1,\ldots,k]$ of the function $f$ to the $n$-bit space. Given an $n$-bit circuit $c$ such that the composition of its gates is equal to $f[1,\ldots,k]$, we say that *circuit $c$ implements $f$ using $n-k$ ancillary bits*.

To get a taste of the constructive side of Boolean function synthesis, let us first characterize Boolean functions that are implementable using only the CNOT gate and no ancillary bits. To this end, let us view the elements of the $\{0, 1\}^n$ as bit vectors and interpret the exclusive OR operation, $\oplus$, as bitwise addition of the vectors mod 2. We say that a reversible boolean function $f : \{0, 1\}^n \to \{0, 1\}^n$ performs a *linear transformation* $f(x \oplus y) = f(x) \oplus f(y)$ for any bit vectors of appropriate dimension.

**Theorem 1.** *A reversible Boolean function can be represented by a circuit constructed entirely of CNOTs if and only if the function performs a reversible linear transformation.*

Obviously, the identity map is a linear transformation, as is any extension CNOT$[i, j]$ to any bit space. Therefore the 'only if' part of the above theorem is straightforward: composition of any number of linear transformations is a linear transformation. A proof of the less trivial 'if' part can be found, for example, in [21].

**Example 5.** *None of the extensions of the NOT gate are linear transformations. This can be seen from the fact that when an extension of the NOT gate is applied to the zero bit vector the result is non-zero. Therefore the NOT gate cannot be implemented by a CNOT circuit using any number of ancillary bits.*

**Example 6.** *Consider a controlled-CNOT gate, CCNOT, which is a ternary gate with the property CCNOT$(x, y, z) = (x, y, (x \wedge y) \oplus z)$. This gate is also called a* Toffoli *gate after its inventor Tommaso Toffoli [22]. Toffoli gate is not a linear transformation (e.g., compare $CCNOT((1, 0, 0) \oplus (1, 1, 0))$ and $CCNOT(1, 0, 0) \oplus CCNOT(1, 1, 0)$). Therefore it cannot be implemented as a CNOT circuit.*

Even more surprising is that the Toffoli gate cannot be implemented as a circuit combining NOTs and CNOTs. We omit the proof in this column. However, if we add a Toffoli gate to our small library of reversible gates and allow ancillary bits, we can represent all the reversible Boolean functions, as outlined in the following definition and theorem. We refer the reader to [21] for the corresponding proof.

**Definition 1.** *A composition of various extensions of the CNOT, NOT, and Toffoli gates is called a $CNT$*-circuit.

**Theorem 2.** *Any reversible Boolean function can be implemented by a $CNT$-circuit using at most one ancillary bit.*

**Exercise 1.** *Consider the 4-bit reversible boolean function $f$ defined by the following rules:*

$$f(0, 0, 0, 0) = (1, 1, 1, 1),$$

$$f(0,1,0,1) = (0,0,0,0),$$

$$f(1,0,1,0) = (0,1,0,1),$$

$$f(1,1,1,1) = (1,0,1,0),$$

*and $f$ is identity on the remaining 12 bit vectors of* $\{0,1\}^4$.

*Show that the function $f$ cannot be implemented with a CNT-circuit using no ancillary bits.*

*Hint* As shown in [21], a reversible function can be implemented by a *CNT*-circuit without ancillary bits if and only if the function performs an even permutation of the bit space. Establish that the function $f$ from the above exercise performs an odd permutation of the $\{0,1\}^4$.

# 3 Quantum States and Quantum Gates

Armed with an understanding of reversible logic gates and circuits, we can now introduce quantum gates and circuits. We begin by describing the principal information unit, the quantum bit, as a quantum system with two basis states. (On a physical level, such states may be represented, for example, by polarizations of a single photon or spin directions of a single electron.)

## 3.1 Quantum States

In a quantum computation, information is stored in a quantum bit, or *qubit*, which extends the concept of the classical bit. Whereas a classical bit has a state value $s \in \{0,1\}$, a state of a qubit $|\psi\rangle$ is actually a linear *superposition* of basis states:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1}$$

where the $\{0,1\}$ basis state vectors are represented in Dirac notation (ket vectors) as $|0\rangle = (1,0)^T$ and $|1\rangle = (0,1)^T$ , respectively. The *amplitudes* $\alpha$ and $\beta$ are complex numbers that satisfy the normalization condition: $|\alpha|^2 + |\beta|^2 = 1$. Upon *measurement* of the quantum state $|\psi\rangle$, either state $|0\rangle$ or $|1\rangle$ is observed with probability $|\alpha|^2$ or $|\beta|^2$, respectively.

Note that a $n$-qubit quantum state is a $2^n \times 1$-dimensional state vector, where each entry represents the amplitude of the corresponding basis state. Therefore, $n$ qubits live in a $2^n$-dimensional Hilbert space, and we can represent a superposition over $2^n$ states as:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i\,|i\rangle\,, \tag{2}$$

where $\alpha_i$ are complex amplitudes that satisfy the condition $\sum_i |\alpha_i|^2 = 1$, and $i$ is the binary representation of integer $i$. Note, for example, that the state $|0000\rangle$ is equivalent to writing the tensor product of the four states: $|0\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle = |0\rangle^{\otimes 4} = (1, 0, 0, 0, 0, 0, 0, 0)^T$. The ability to represent a superposition over exponentially many states with only a linear number of qubits is one of the essential ingredients of a quantum circuit — an innate massive parallelism.

**Example 7.**

1. *The two-qubit state* $(1/2)(|00\rangle - i\,|01\rangle + i\,|10\rangle + |11\rangle)$ *is a product of the single-qubit state* $(1/\sqrt{2})(|0\rangle + i\,|1\rangle)$ *on the first qubit and the* $(1/\sqrt{2})(|0\rangle - i\,|1\rangle)$ *state on the second qubit.*

2. *The two-qubit state* $(1/\sqrt{2})(|00\rangle + |11\rangle)$ *is not a product of two individual single-qubit states*

The state given in Example 7(2.) possesses a non-classical *entanglement* property. When multi-qubit state cannot be represented as a product of individual single-qubit states, we say that the state is *entangled*. Intuitively, this means that for at least two qubits in the system we cannot in principle identify, or separate out, their individual states.

It follows from the principles of quantum mechanics that two quantum states are indistinguishable if they differ only by a *phase factor* of the form $e^{i\theta}, \theta \in \mathbb{R}$. Thus, we can rewrite a qubit as $\cos(\theta)\,|0\rangle + e^{i\phi}\sin(\theta)\,|1\rangle$, where $0 \leq \phi < 2\pi, 0 \leq \theta \leq \pi/2$. We can interpret $2\theta$ and $\phi$ as spherical angle coordinates and map the state onto a point on the unit sphere, allowing a geometrical interpretation of single-qubit states, as originally proposed by Felix Bloch [38].

Now consider the evolution of a quantum state. Such evolution would need to preserve the norm of the complex vectors representing the states and would also need to transform a superposition of states into a superposition of transformed states. A physically motivated operation would be, for example, a linear operator $\mathbb{C}^2 \rightarrow \mathbb{C}^2$ that preserves the inner product $\langle(\alpha, \beta), (\gamma, \delta)\rangle = \alpha\gamma^* + \beta\delta^*$, where $*$ is complex conjugate transpose. Such operetors are known as *unitary operators* (c.f., [43]).

A quantum computation proceeds through the *unitary* evolution of a quantum state; in turn, quantum operations are necessarily *reversible*. We refer to quantum unitary operations as quantum *gates*. In the multi-qubit case, an $n$-qubit quantum gate is a $2^n \times 2^n$ unitary matrix acting on an $n$-qubit quantum state.

We can more formally define a unitary operator $U$. For a given invertible linear transformation $U : \mathbb{C}^N \rightarrow \mathbb{C}^N$, we introduce

$$U^\dagger : \mathbb{C}^N \rightarrow \mathbb{C}^N$$

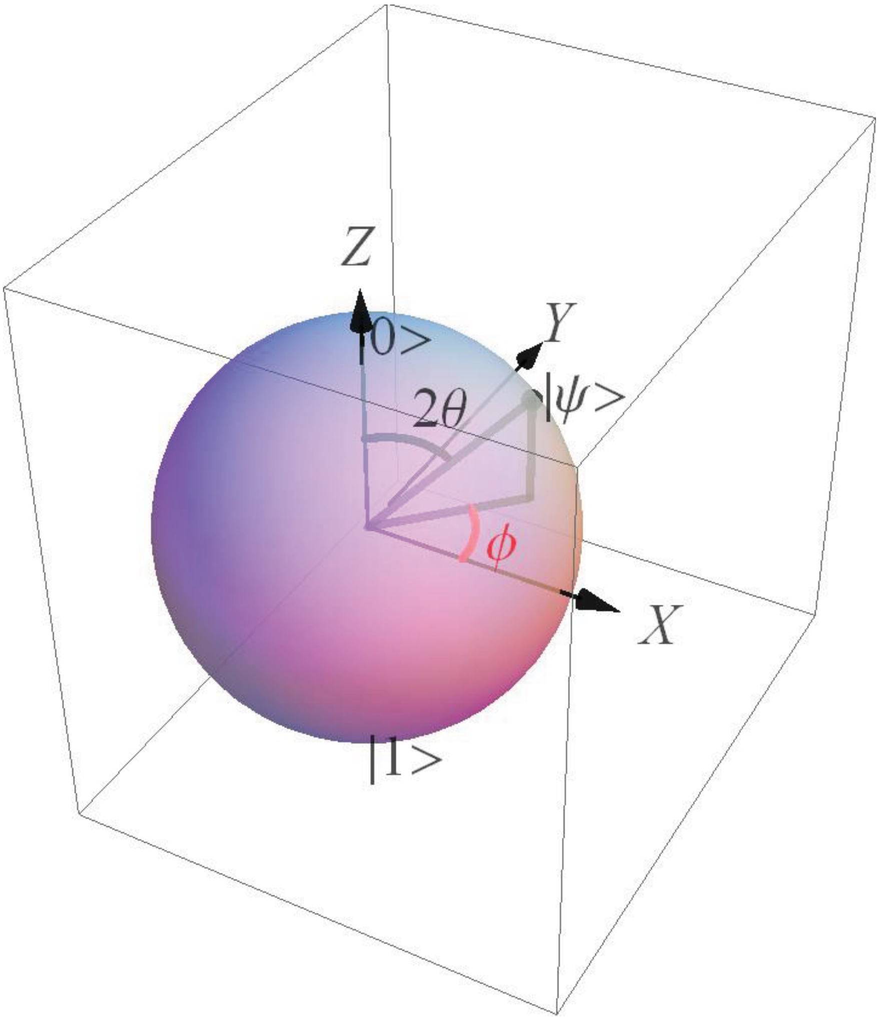

Figure 1: Bloch sphere representation of a single-qubit state. The two basis states, $|0\rangle$ and $|1\rangle$, sit at the two poles.

as the transformation defined by transposition and complex conjugation of the matrix of $U$.

**Definition 2.** *A linear operator $U : \mathbb{C}^N \to \mathbb{C}^N$ is* unitary *if $UU^\dagger = I$ (or in other words $U^\dagger$ is the inverse of $U$).*

Returning to the single-qubit Bloch sphere interpretation, because a unitary operator $A : \mathbb{C}^2 \to \mathbb{C}^2$ transforms valid single-qubit states into valid single-qubit states and because the states map onto points on the Bloch sphere (see Fig. 1), a single-qubit unitary can be interpreted as some transformation of the Bloch sphere. It is not difficult to see that this transformation is, in fact, an isometry. Conversely, each isometry from the special orthogonal group SO(3) corresponds to an equivalence class of single-qubit unitary operators. Because superposition states are defined up to am arbitrary global phase factor, two unitary operators that differ only by a multiplicative $e^{i\theta}$ are considered equivalent, in particular any unitary is equivalent to one with determinant equal to 1.

## 3.2 Reversible 'Classical' Gates

Unitary operations on an $n$-qubit system are, by definition, reversible. Any reversible $n$-bit classical Boolean function $f$ can be converted into the $n$-qubit unitary operator $U_f$ by defining the action on the standard basis $|s\rangle$ as follows:

$$U_f(|s\rangle) = |f(s)\rangle \,.$$

We call a unitary operator $U_f$, where $f$ is a reversible Boolean function, a *classical unitary gate*. From the definition, it follows that $U_*$ is a functor preserving, for reversible Boolean functions $f$ and $g$, the composition: $U_{f\circ g} = U_f \circ U_g$.

Boolean gates of the universal $CNT$ basis generate the following unitary gates: $X = U_{\text{NOT}}$ is a single-qubit gate with the matrix:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

$U_{\text{CNOT}}$ is a two-qubit gate, denoted by $\Lambda(X)$, and called CNOT. Its matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Similarly, $U_{\text{CNOT}[2,1]}$ is denoted by $\Lambda(X)[2,1]$ and its matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Since we can compute $U_{f\circ g}$, we can obtain the $U_{\text{SWAP}}$ gate:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$U_{\text{CCNOT}}$ is a three-qubit, denoted by $\Lambda^2(X)$, and referred to as the Toffoli gate. Its matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The $\Lambda$ symbol surreptitiously introduced above is actually the functor of adding a control qubit to a unitary. For an $n$-qubit unitary $G$, $\Lambda(G)$ is the $n+1$-qubit unitary defined as follows in the standard basis:

$$\Lambda(G)\,|0\,s_1 \ldots s_n\rangle \equiv |0 s_1 \ldots s_n\rangle\,,$$

$$\Lambda(G)\,|1\,s_1 \ldots s_n\rangle \equiv |1\rangle\, G\,|s_1 \ldots s_n\rangle\,.$$

Unlike the classical case where the control bit is a logical switch on the application of the target gate, the meaning of the control qubit is more complex, since the qubit can be in superposition. Nevertheless, at the matrix level, $U_{C(f)} = \Lambda(U_f)$ for any reversible Boolean function $f$.

An important non-classical single-qubit gate is the *Hadamard* gate $H$, which maps a quantum state into a quantum superposition state, as follows:

$$H\,|0\rangle = (1/\sqrt{2})(|0\rangle + |1\rangle),$$

$$H\,|1\rangle = (1/\sqrt{2})(|0\rangle - |1\rangle),$$

where the unitary matrix is given by

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{3}$$

There are $2^n!$ reversible Boolean functions and as many 'classical' gates on $n$ qubits. Although the size of this set is double-exponential in $n$, the set turns out to be very sparse in the infinite continuous group of unitary operators.

## 3.3 Pauli Gates

The single-qubit *Pauli group* is generated by compositions of the following unitary gates, called *Pauli gates*:

$$I = U_I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X = U_{\mathrm{NOT}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Of these generating gates, only $X$ and $I$ are classical. Note that $X^2 = Y^2 = Z^2 = I$.

An easy matrix algebra exercise shows that the Pauli gates generate a group of 16 elements with a 4-element center $z = \{\pm I, \pm i\, I\}$. Since $-1 = e^{i\pi}$ and $\pm\, i = e^{\pm i\,\pi/2}$ are phase factors, each element of $z$ is equivalent to the identity as a single-qubit unitary gate and each element of the Pauli group is equivalent to one of the four gates in $\{I, X, Y, Z\}$

Recall that a single-qubit unitary can be interpreted as a rotation of the Bloch sphere and apply this interpretation to the Pauli gates. Then it is easy to verify that $\{X, Y, Z\}$ are rotations by angle $\pi$ about axes $\{x, y, z\}$, respectively. Proceeding with the Bloch sphere exercise, we find an easy recipe for writing out the rotation corresponding to any single-qubit unitary $A$.

**Exercise 2.** *(1) Prove that given a single-qubit unitary $A$ and $P \in \{X, Y, Z\}$ , $A\,P\,A^\dagger = a_P\,X + b_P\,Y + c_P\,Z$ where $a_P, b_P, c_P$ are uniquely defined by* real *coefficients.*

Hint. *Due to the unitary condition $A\,A^\dagger = I$ , matrix $A$ is defined up to a phase factor by its first row. More specifically, $A = e^{i\alpha} * B$, where $\alpha \in \mathbb{R}$ and $B =$* $[[u, v], [-v^*, u^*]]$. *It is easy to see that $B = Re(u)\,I + i(Im(v)\,X + Re(v)\,Y + Im(u)\,Z)$.*

*(2) Prove that in the context of (1)*

$$\begin{bmatrix} a_X & a_Y & a_Z \\ b_X & b_Y & b_Z \\ c_X & c_Y & c_Z \end{bmatrix}$$

*is a special orthogonal matrix.*

*(3) Prove that the matrix in (2) defines the rotation of the Bloch sphere corresponding to $A$.*

The multi-qubit Pauli group is, again, generated by the $\{I, X, Y, Z\}$ using tensor products along with the compositions. Let us start with a formal description of the tensor product of unitary operators. Let $n = k + m$ be an integer partition of the natural integer $n$. We note that the complex vector space $\mathbb{C}^{2^n}$ is represented as tensor product $\mathbb{C}^{2^k} \otimes \mathbb{C}^{2^m}$. One specific tensor representation is written in terms of standard bases in $\mathbb{C}^{2^k}$ and $\mathbb{C}^{2^m}$ by noting that if $\{a_1, \ldots, a_{2^k}\}$ is a basis in the former and $\{b_1, \ldots, b_{2^m}\}$ is a basis in the latter, then the formal pairs $(a_j, b_l), 1 \leq j \leq 2^k, 1 \leq l \leq 2^m$ form a basis in a complex vector space of dimension $(2^k)\times(2^m) = 2^n$. For the purposes of the tensor product representation we denote the new basis element $(a_j, b_l)$ as $a_j \otimes b_l$ or simply $a_j b_l$.

**Definition 3.** *Given two linear operators $A : \mathbb{C}^M \to \mathbb{C}^M$ and $B : \mathbb{C}^N \to \mathbb{C}^N$, $M, N \in \mathbb{N}$, the* tensor product *of $A$ and $B$ is the linear operator $A \otimes B : \mathbb{C}^M \otimes \mathbb{C}^N \to \mathbb{C}^M \otimes \mathbb{C}^N$ uniquely defined by the property*

$$(A \otimes B)(a \otimes b) = (A\,a) \otimes (B\,b).$$

Iterating the construction and the definition, we can represent the $n$-qubit state space $\mathbb{C}^{2^n}$ as a tensor product of $n$ copies of the one-qubit state space $\mathbb{C}^2$. In particular for any set of unitaries $A_j : \mathbb{C}^2 \to \mathbb{C}^2, j = 1, \ldots, n$ , the tensor product $A_1 \otimes \cdots \otimes A_n$ is the unitary operator uniquely defined by the property

$$(A_1 \otimes \cdots \otimes A_n)(|\psi_1\rangle \ldots |\psi_n\rangle) = A_1(|\psi_1\rangle) \ldots A_n(|\psi_n\rangle).$$

If $A_1 = \ldots = A_n = A$, then the above tensor product is written as $A^{\otimes n}$.

It is easy to see that, in the context of the above definition, given pairs of operators $A, C : \mathbb{C}^M \to \mathbb{C}^M$ and $B, D : \mathbb{C}^N \to \mathbb{C}^N$,

$$(A \circ C) \otimes (B \circ D) = (A \otimes B) \circ (C \otimes D).$$

We are now ready to define the *n-qubit Pauli group* $P_n$ as the group with respect to composition generated by $\{P_1 \otimes \ldots \otimes P_n | P_i \in \{I, X, Y, Z\}, i = 1, \ldots, n\}$. It is known (c.f., [24]) that this group has $2^{2n+2}$ elements. For any $n$ , $P_n$ has a 4-element center. Introducing $I_n = I^{\otimes n}$, we can describe the center as $\{\pm I_n, \pm i\, I_n\}$.

Pauli groups are important in quantum information theory. This is due to the fact that it forms the core of the so-called *Heisenberg* representation of quantum computing (see [6]), where quantum information is encoded in eigenstates of certain Pauli operators (we refer the reader to [6] for details.)

## 3.4 Clifford Group

Consider the group $U(2^n)$ of the $n$-qubit unitaries. The Pauli group $P_n \subset U(2^n)$ is a subgroup, and is tiny compared to the continuous infinite group $U(2^n)$. We want to look for meaningful ways to extend this group of operations into a larger and more powerful set.

One might logically ask what is a set of unitary operations that *preserves* the Pauli group? (This would be a set of operations that would preserve the Heisenberg computational model mentioned in the previous subsection.) In group theory language, the question would be: what is the *normalizer* of the Pauli group?

Before answering this question, we note that the center of the $U(2^n)$ consisting of the scalar operators of the form $e^{i\theta} I, \theta \in \mathbb{R}$ stabilizes all the elements of $U(2^n)$ and is trivially a part of the normalizer for $P_n$. This is not at all interesting. The question then must be: what other operators outside the center and the $P_n$ are in the normalizer of $P_n$?

Here, the *Hadamard* gate $H$ comes to prominence. Recall that $H$ is a single-qubit gate defined as

$$H\,|0\rangle = (1/\sqrt{2})(|0\rangle + |1\rangle),$$
$$H\,|1\rangle = (1/\sqrt{2})(|0\rangle - |1\rangle).$$

It is easy to see that $H^2 = I; H X H = Z; H Y H = (-1) Y; H Z H = X$, therefore $H$ is in the normalizer of the single-qubit Pauli group. Obviously, using $H$ in a tensor product with other normalizer gates generates a normalizer element of the respective multi-qubit Pauli group. For example, $I \otimes H, H \otimes I$ and $H \otimes H$ are in the normalizer of the two-qubit Pauli group.

Another important gate that we must now introduce is the *phase gate* $S$ defined as

$$S |0\rangle = |0\rangle ; S |1\rangle = i |1\rangle .$$

Unlike other gates considered thus far, $S$ is not involutive, rather we see immediately that $S^2 = Z$ so $S^4 = I$, making it an order-4 group element. The inverse is given by $S^\dagger = S^3$.

By direct computation,

$$S X S^\dagger = Y; S Y S^\dagger = (-1)X; S Z S^\dagger = Z.$$

Thus $S$ is a normalizer element and so are its compositions and tensor products with other normalizer elements. For example, $S \otimes H, S \otimes S$ and $H \otimes S$ are all in the normalizer of the two-qubit Paulis.

We note that the already familiar CNOT gate $\Lambda(X)$ preserves the two-qubit Pauli group (as does any controlled-Pauli gate $\Lambda(P), P \in \{X, Y, Z, (-1)I, \pm i I\}$ ; this is easy to check by direct computation). We also note the following amusing two-qubit identity:

$$(H \otimes H) \circ \Lambda(X) \circ (H \otimes H) = \Lambda(X)[2, 1].$$

**Theorem 3.** *The normalizer of the n-qubit Pauli group in $U(2^n)$ is generated by the center $z(U(2^n))$ (the subgroup of scalar unitaries), tensor products of $I, H, S$ operators, and various CNOT operators $\Lambda(X)[j, l], 1 \leq j < l \leq n$.*

A proof of an equivalent theorem can be found in Chapter 10 of [23].

**Definition 4.** *The* Clifford group *is the group of unitary operators, generated by*

(1) $H$ and $S$ in the single-qubit case

(2) Tensor products of $I, H, S$ and all $\Lambda(X)[j, l], 1 \leq j < l \leq n$ in the $n$-qubit case, $n > 1$.

For any number of qubits, the Clifford group is the "non-trivial part" of the normalizer of the Pauli group.[1] Interestingly, although the term *Clifford group* is

[1] It is commonly claimed that the Clifford group *is* the normalizer of the Pauli group. Strictly speaking, this claim is incorrect. It is only meaningful 'modulo scalar operators'. More precisely, the central quotient of the Clifford group is the normalizer of the central quotient of the Pauli group in the central quotient $PU(2^n)$ of the unitary group.

universally accepted, its origin is not entirely clear (it is not directly related to the class of Clifford algebras). According to D. Gottesman, the first use of the term is attributed to Eric M. Rains.

If we view Clifford operators as "instructions" on a quantum computer, we get a rather large instruction set. As per [24], the $n$-qubit Clifford group has $2^{n^2+2n+3}\prod_{j=1}^{n} 4^j - 1$ distinct elements. For example, this amounts to 92,160 elements in the two-qubit case. Regrettably, this instruction set does not provide any speedups for quantum computers over classical computers. This remarkable result is the subject of the next section.

# 4 Gottesman-Knill Theorem

In order to harness the power of a quantum computer, we need to first step away from the unitary operator paradise and introduce some non-unitary operations. We start this section by discussing *quantum measurement* and *classical feedback* operations.

## 4.1 Measurement and Classical Feedback

It is one of the great mysteries of quantum mechanics that measurements are parameterized by Hermitian operators. An operator $M : \mathbb{C}^N \to \mathbb{C}^N$ is called *Hermitian* if $M = M^\dagger$. Obviously all the eigenvalues of such an operator are real. It follows that if $M$ is both unitary and Hermitian, then it is also involutive, i.e., $M^2 = I$, with eignevalues ±1. Note, for example, that a generator of the Clifford group is Hermitian iff it does not explicitly contain the $S$ gate.

If $\{m_1, \ldots, m_l\}$ is the list of distinct eigenvalues of a Hermitian operator $M$, then it is a simple algebraic fact that

$$M = \sum_j m_j \, Pr_j,$$

where $Pr_j$ is the projector onto the eigenspace of $M$ corresponding to the eigenvalue $m_j$. Conceptually, according to the postulates of quantum mechanics, a *measurement* of the operator $M$ on a quantum state $|\psi\rangle$ must produce one of the eigenvalues of $M$. Any of the eigenvalues may randomly result from the measurement.

We need a way to compute the probability $p_j$ of observing a certain eigenvalue $m_j$ in the measurement. To this end, consider the projection $Pr_j\,|\psi\rangle$ of a state vector $|\psi\rangle$ on the $j$th eigenspace. Note that the scalar product $\langle\psi|\,Pr_j\,|\psi\rangle$ is a measure of proximity of the state vector $|\psi\rangle$ to the eigenspace, very similar to the cosine of the angle between a vector and a plane in Euclidean space (the larger the cosine, the smaller the angle).

It is the *measurement postulate* of quantum mechanics that defines $p_j$ as $p_j = \langle\psi| Pr_j |\psi\rangle$. It is easy to see that $\sum_j p_j = \langle\psi| |\psi\rangle = 1$.

The fundamental *state reduction* principle also states that *if the eigenvalue $m_j$ is observed in measuring the operator $M$, the quantum state $|\psi\rangle$ is changed ('collapsed') into $Pr_j |\psi\rangle$ post-measurement.*

The probability of observing an eigenvalue of $M$ can thus be also understood as the probability of the quantum state being forced into the corresponding eigenspace. It makes intuitive sense that such a probability is proportional to a proximity measure between the state and the eigenspace.

**Example 8.** *The simplest scenario is measuring the Pauli operator $Z$ on the single-qubit state $\alpha|0\rangle + \beta|1\rangle$ . The eigenvalues of $Z$ are $+1$ and $-1$ and the probabilities of observing each one are $|\alpha|^2$ and $|\beta|^2$ respectively. Post-measurement the state collapses to either basis state $|0\rangle$ or to basis state $|1\rangle$. The standard notation for this measurement procedure is $M_Z$. In multi-qubit case we will use the notation $M_Z[i]$ for the Z-measurement applied to $i^{th}$ qubit only.*

One key application of measurements in quantum computation is the feeding of the measurement results back into quantum circuits to be used as *classical control bits*. The non-unitary primitive that makes such feedback possible is called a *classically controlled gate*. Given an operator $G \in U(N)$, the classically controlled gate

$$BC(G) : (\{0,1\} \times \mathbb{C}^N) \to (\{0,1\} \times \mathbb{C}^N)$$

is defined by

$$BC(G)((0,v)) = (0,v);\ BC(G)((1,v)) = (1,G\,v).$$

### 4.1.1 An Important Toffoli-based Construction

We now consider an important measurement example that is significantly more sophisticated than Example 8, and introduces the concept of a classical feedback loop. We will implement a certain single-qubit rotation that will turn out to be important in the next section.

Consider a single-qubit state $|\psi\rangle$ and add two ancillary qubits prepared in state $|0\rangle$. To simplify notations, assign indices 1 and 2 to the ancillary qubits and assign index 3 to the qubit in the state $|\psi\rangle$. The resulting 3-qubit system is initially in the product state $|00\rangle|\psi\rangle$.

Consider operator $H \otimes H \otimes I$ that performs the Hadamard gate on qubits 1 and 2 and leaves qubit 3 unchanged; consider operator $I \otimes I \otimes S$ that leaves qubits 1 and 2 unchanged while performing the phase gate $S$ on the third qubit. Build the

3-qubit circuit $U = (H \otimes H \otimes I)\,\Lambda^2(X)\,(I \otimes I \otimes S)\,\Lambda^2(X)\,(H \otimes H \otimes I)$, apply it to the 3-qubit system prepared in state $|00\rangle\,|\psi\rangle$, then apply $M_Z$ measurement operator to qubits 1 and 2.

Note that the Toffoli gate $\Lambda^2(X)$ does not belong to the Clifford group and neither does the composite $U$. Given $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$, $|a|^2 + |b|^2 = 1$, by direct computation we obtain

$$U\,|00\rangle\,|\psi\rangle = (1/4)\,((3+i)\,a\,|000\rangle + (1+3i)\,b\,|001\rangle + \\ (1-i)\,a\,|010\rangle - (1-i)\,b\,|011\rangle + (1-i)\,a\,|100\rangle - \\ (1-i)\,b\,|101\rangle + (i-1)\,a\,|110\rangle - (i-1)\,b\,|111\rangle).$$

We introduce the Clifford gate $IIZ = I \otimes I \otimes Z$ that leaves qubits 1 and 2 unchanged and performs a Pauli-$Z$ gate on the third qubit and introduce the classically controlled gate $BC(IIZ)(M_Z[1] \vee M_Z[2], *)$, where the notation reads: apply the $IIZ$ gate *unless* $M_Z[1] = |0\rangle$ and $M_Z[2] = |0\rangle$. Now apply the composition $BC(IIZ)(M_Z[1] \vee M_Z[2], *) \circ U$ to the $|00\rangle\,|\psi\rangle$ state.

As per the state reduction principle, when $M_Z[1] = |0\rangle$ and $M_Z[2] = |0\rangle$, the $U\,|00\rangle\,|\psi\rangle$ is projected to the $+1$ eigenspace of $Z \otimes I \otimes I$, then to $+1$ eigenspace of $I \otimes Z \otimes I$. In other words, the state gets projected onto the two-dimensional space spanned by $|000\rangle$ and $|001\rangle$. From the above expression for $U\,|00\rangle\,|\psi\rangle$, we derive that the projected state vector is proportional to $(3+i)\,a\,|000\rangle + (1+3\,i)\,b\,|001\rangle$ and thus it is equivalent to $a\,|000\rangle + ((1+3\,i)/(3+i))\,b\,|001\rangle = a\,|000\rangle + (\frac{3+4\,i}{5})\,b\,|001\rangle$.

To summarize, the case when measurement outcomes are $|0\rangle$ is equivalent to applying the $V = [[1,0],[0, \frac{3+4\,i}{5}]]$ gate to the third qubit. Looking at the remainder of the expression for $U\,|00\rangle\,|\psi\rangle$, it is easy to see that all other outcomes are equivalent to applying the Pauli-$Z$ gate to the third qubit, which is then canceled out by the $BC(IIZ)(M_Z[1] \vee M_Z[2], *)$ operator.

Finally, we estimate the probability of measurement outcomes being simultaneously $|0\rangle$. As per the measurement postulate above, that probability is $p_{00} = |(3+i)/4|^2\,|a|^2 + |(1+3\,i)/4|^2\,|b|^2 = 5/8\,(|a|^2 + |b|^2) = 5/8$. Note, for now, that using the above protocol, the probability of performing the gate $V$ on the third qubit is higher than the probability of leaving the third qubit state unchanged.

## 4.2 The Theorem and Discussion

Daniel Gottesman [6] and, independently, Emmanuel Knill, conjectured (and later proved) that certain quantum circuits, when containing only a subset of quantum operations and measurements, could be efficiently computed on a classical computer. Informally, if the computer uses only, for example, the gates within the Clifford group and measurements in the computational basis, then it is no more powerful (and in fact, more restricted) than a classical computer. One of the most common versions of this result is articulated in the following theorem.

**Theorem 4.** *The result of applying a sequence of Clifford gates followed by a Pauli measurement to the input state $|\mathbf{0}\rangle = |0\rangle^{\otimes N}$ can be simulated in polynomial time on a probabilistic classical computer.*

The practical corollary is that if we only use Clifford gates for both preparation of a quantum state and the evolution of the quantum state, then this computation will not have an exponential speed-up over the corresponding classical computation.

In [8], Maarten Van den Nest established a slightly more general result regarding the 'classicality' of certain circuits. Recall that the quantum Toffoli gate $\Lambda^2(X)$ does not belong to the Clifford group. (As an exercise, check $\Lambda^2(X)(X \otimes I \otimes I)\Lambda^2(X)$.) Consider, however, a type of circuit called *H-Toffoli* that consists of two decoupled parts: the first part is a multi-qubit Hadamard gate $H^{\otimes N}$ and the second part is an arbitrary $N$-qubit circuit composed of the NOT, CNOT and Toffoli gates. Because classical {NOT, CNOT, Toffoli} constitute a universal basis in the the group of reversible classical circuits, we note that the second part is a quantum wrapper around *arbitrary* reversible boolean function, i.e., it is of the form $U_f$, where $f$ is the reversible boolean function computed by the classical circuit replicating the {NOT, CNOT, Toffoli} part of the quantum circuit.

Then any H-Toffoli circuit followed by a Pauli measurement has the same computational power as a probabilistic classical computation. Intuitively, the result may be not so unexpected given the circuit is $H^{\otimes N}\, U_f$, where $f$ is classical. This is surprising though: if we allow the Hadamard gate and the quantum Toffoli gate to *interleave*, then we get a 'universal' circuit group that goes beyond classical computation and delivers the famous exponential speedups observed in some quantum algorithms. This phenomenon is discussed in the next section.

# 5 Universal Quantum Bases

Since any constructive set of operations is going to be finite or countably infinite, we need a different notion of universality and a different concept of circuit synthesis. Both are based on the notion of a *dense subgroup* of the unitary group $U(N)$.

**Definition 5.** *A subgroup $G \subset U(N)$ is everywhere* dense *if for every $u \in U(N)$ and for every $\epsilon > 0$ there exists a $g_\epsilon \in G$ such that the distance between $g_\epsilon$ and $u$ is less than $\epsilon$.*

There are several ways to define the distance on an operator group, for the purposes of this definition, they are equivalent. The distance most often used in quantum computing literature is the *trace distance* and is defined on $U(N)$ as

$$dist(U,V) = \sqrt{(N - |tr(U\,V^{\dagger})|)/N},$$

where $tr$ stands for the operator trace. Since $tr(I_N) = N$, each operator is at zero distance with itself. It is not difficult to prove that the distance as defined is non-negative real and satisfies the triangle inequality.

*Concept:* A finite set of quantum gates forms a *pure universal quantum basis* in $n$-qubit space if they generate an everywhere dense subgroup of $U(2^n)$.

The best fault-tolerant implementations of operations in a pure universal quantum basis are not entirely unitary; they also require the use of non-unitary operations, including (but not limited to) *state preparation*, *measurement*, and *classical feedback*. These non-unitary operations may use varying numbers of *ancillary qubits*. (To get some taste of the amount of 'non-unitary help' required, an inquisitive reader may consult [33] or the appendix of [35].) As in the reversible logic world, allowing ancillary qubits may be more desirable than increasing the number of operations required for implementation, which relates to the following definition.

**Definition 6.** *We say that a k-qubit unitary $u \in U(2^k)$ is approximated to precision $\epsilon > 0$ by a circuit $c \in U(2^n)$, where $n \geq k$,* using $n - k$ ancillary qubits *if either $I_{n-k} \otimes u$ is at a distance less than $\epsilon$ from c or u is at a distance less than $\epsilon$ from a projection of c onto $U(2^k)$.*

In this definition, the term *projection* refers to a factorization map $U(2^n) \to U(2^k)$ related to some non-unitary operation(s).

We are finally ready to discuss universal quantum bases, which enable quantum computations that cannot be simulated classically. Exact and effective unitary reduction leads to the following result first published in [25]:

**Theorem 5.** *The circuit group generated by CNOT and all single-qubit unitary operators is purely universal in the multi-qubit space.*

This particular reduction puts an onus on implementing any single-qubit gate $G$ that is universal in single-qubit space, possibly in combination with the single-qubit Clifford group. In fact, *any gate G has this property, unless the eigenvalues of $G^2$ are* $\pm 1$.

In light of this it would seem that the gate $T = [[1,0],[0,\sqrt{i}]]$ is the simplest and most logical choice, since the phase gate $S = T^2$ has one eigenvalue equal to $i$. The gate $T$, commonly known as the $\pi/8$-*gate*, was originally proposed in

[26] (albeit with a different rationale). While $\{H, S\}$ generate the finite single-qubit Clifford group, $\{H, T\}$ is a universal single-qubit basis and hence generates an infinite group, everywhere dense in $U(2)$, that, of course, contains $S = T^2$ and thus contains the entire Clifford group. Research based on this 'Clifford+$T$' basis has generated a steady stream of both theoretical and practical results over recent years (an incomplete selection includes [27, 28, 29, 30, 31]).

In fact, the use of this basis is so common that the research community has focused on developing fault-tolerant implementations of the $T$ gate, while perhaps overlooking other more convenient universal bases. The best fault-tolerant implementations of $T$ are based on the so-called *magic state distillation protocol* that consumes a number of ancillary qubits while also requiring non-unitary steps such as measurement and classical feedback (see, for example, [32, 33, 34]). Nevertheless, the $T$ gate provides a convenient abstraction, where the non-unitary techniques needed for implementation are separate from the group-theoretical guarantees.

To give a taste of alternative universal bases, we will briefly sketch a more recent proposal, borrowing directly from reversible logic circuits. The alternative is based on the quantum Toffoli gate $\Lambda^2(X)$ and Subsection 4.1.1.

In 2002, Shi [9] offered an elegant proof that the Toffoli gate in combination with the Hadamard gate form a universal quantum basis when *one* ancillary qubit is allowed. However, the proof does not yield a constructive algorithm to perform the actual approximation of a unitary gate by a synthesized Toffoli/Hadamard circuit to a desired precision.

In contrast, the task of synthesizing Clifford+Toffoli circuits has recently become algorithmic. In [35], an algorithm for synthesizing efficient Clifford+$V$ circuits, where $V = [[1, 0], [0, \frac{3+4i}{5}]]$ is the gate constructed in subsection 4.1.1, is presented. (Action of the $T$ and $V$ gates on the Bloch sphere are shown schematically in Fig. 2.) It shows that any single-qubit unitary can be effectively approximated to precision $\epsilon$ by a Clifford+$V$ circuit containing no more than $4 \log_5(2/\epsilon)$ occurrences of the $V$ gate.

By iterating over the circuit from 4.1.1, we can perform the $V$ gate with probability 1. The actual number of iterations needed is a random variable, however its expected value is $5/8 * \sum_k k\,(3/8)^{k-1} = 8/5$. Thus a Toffoli-based circuit approximating a single-qubit target to precision $\epsilon$ will have on average $(64/5) \log_5(2/\epsilon)$ occurrences of the Toffoli gate.

As per Theorem 5, the ability to effectively approximate any single-qubit gate with a Toffoli-based circuit, combined with the two-qubit Clifford gate $CNOT$, implies the ability to effectively approximate any multi-qubit unitary by such a circuit. Note that we also have a specific upper bound on the number of occurrences of the Toffoli gate in the resulting approximation.

As defined, this solution currently consumes more resources than the most

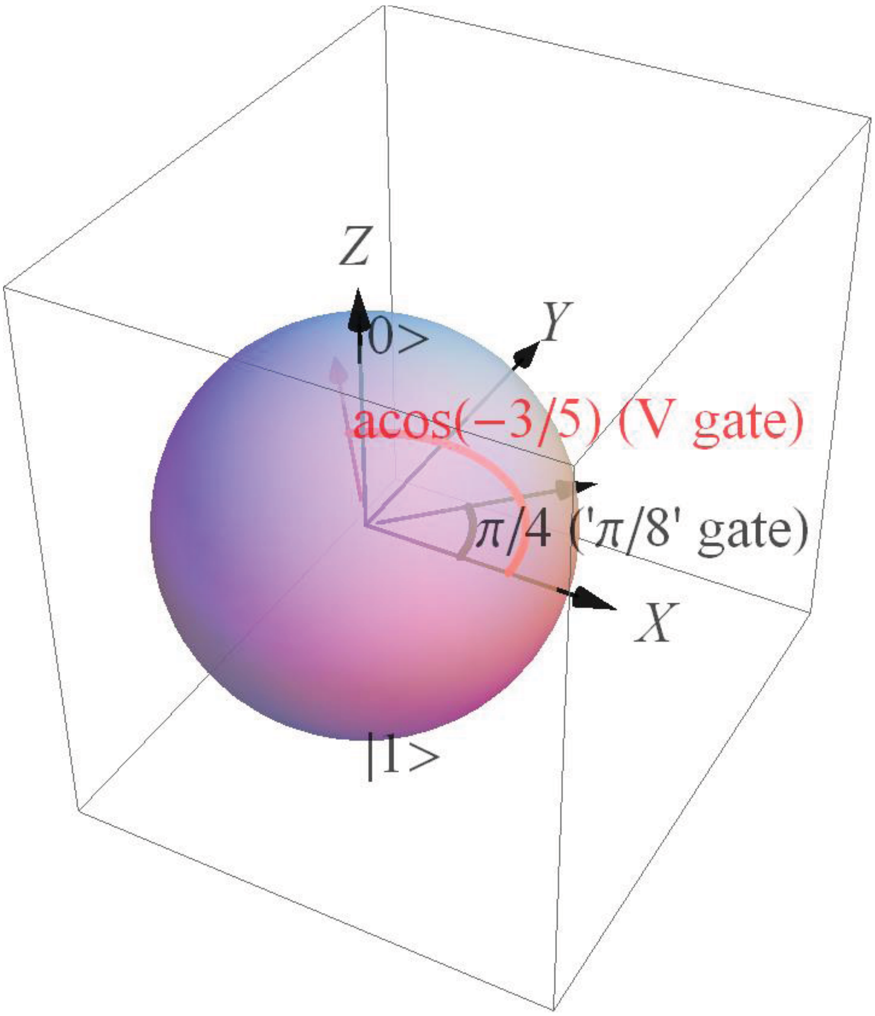

Figure 2: Action of the '$\pi/8$' and $V$ gates on the Bloch sphere. The gates perform rotations about $Z$ axis by the angles of $\pi/4$ and $cos^{-1}(-3/5)$ respectively (the latter angle is an irrational multiple of $\pi$).

recent solutions based on the Clifford+$T$ basis (primarily because known fault-tolerant implementations of the Toffoli gate are even more expensive than those of the $T$ gate [41, 42]). However we point out this alternative not just to prove the algorithmic feasibility of the universal Toffoli-based quantum circuits. There is evidence that in the multi-qubit space such circuits can be more aggressively optimized than those based on the Clifford+$T$ approach.

## 6 Future Directions

At this stage of research, circuits based on universal quantum bases constitute the most popular framework for implementing quantum algorithms. The implementation of an algorithm begins with a definition of the required high-level unitary and non-unitary operators, followed by a *quantum compilation* step, where the high-level operators are represented by circuits in a chosen basis.

Interestingly, the year 2012 could likely be referred to as the "Year of Quantum Circuit Decomposition". Until early 2012, the most popular and most efficient method for decomposing (or compiling) a high-level quantum circuit, in particular the single-qubit gates, into implementable and fault-tolerant quantum gates was the Dawson-Nielsen version of the Solovay-Kitaev theorem [36, 37]. Given a target unitary gate and a compilation precision $\epsilon$, this method delivers circuits of depth $O(\log^{3.97}(1/\epsilon))$. A handful of theoretical results published over the last decade (c.f., [39]), however, suggested that much more efficient circuit depths $O(\log(1/\epsilon))$ could be achieved for some bases, but no constructive compilation algorithms to achieve these asymptotics were yet known.

Remarkably, in the course of 2012, *efficient* circuit compilation algorithms achieving circuit depths of $O(\log(1/\epsilon))$ have been discovered for two universal bases. Compared to the previous solution in [36, 37], the cost of a circuit implementing a typical single-qubit rotation in an algorithm such as Shor's factorization [10] has come down from millions of basis gates to mere dozens of gates. The latest compilation algorithms (see, for example, [35, 31, 29]) not only address the asymptotic circuit growth rate, but also come with specific upper bounds on the circuit depth.

We now look to an upcoming year to label as the "Year of Multi-qubit Decomposition". Depth upper bounds for multi-qubit circuits is the next research frontier for circuit compilation. Most of the algorithms referenced in this column exploit the following two facts:

1. any single-qubit unitary can be decomposed, effectively and exactly, into at most three axial rotations,

2. any controlled single-qubit unitary can be decomposed, effectively and exactly, into at most three uncontrolled single-qubit unitaries (interleaved with at most two CNOTs).

Sidestepping either of these two intermediate decomposition steps would slash the depth of a circuit implementing a general controlled unitary by a factor of 3 (bypassing both has the potential of reducing the constant coefficient in front of the $\log(1/\epsilon)$ by a factor of 9). In this respect, using multi-qubit primitive gates (such as Toffoli) hold much promise for the future of practical compilation of quantum algorithms.

## 7 Acknowledgments

The Authors wish to thank Andreas Blass, Yuri Gurevich, and Nathan Cody Jones for very useful discussions.

## References

[1] A. Lubotsky, R. Phillips, P. Sarnak. Hecke operators and distributing points on $S^2$, I and II. *Comm. Pure and Appl. Math., 34, 149–186, and 40, 401–420*, 1986,1987.

[2] R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development, 5, 183–191*, 1961.

[3] G. E. Moore. Cramming more components onto integrated circuits. *Electronics Magazine, p.4*, 1965.

[4] R. Kurzweil. The Singularity is Near. *Penguin Books*, 2005.

[5] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development, vol. 17, no. 6, 525–532*, 1973.

[6] D. Gottesman. The Heisenberg Representation of Quantum Computers. arXiv:quant-ph/9807006v1, 1998. (http://arxiv.org/abs/quant-ph/9807006v1).

[7] S. Aaronson, D. Gottesman. Improved Simulation of Stabilizer Circuits . *Phys. Rev. A, 70, 052328* , 2004. arXiv:quant-ph/0406196, 2004.

[8] M. Van den Nest. Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond. arXiv:quant-ph/0811.0898, 2008. (http://arxiv.org/abs/0811.0898).

[9] Y. Shi. Both Toffoli and Controlled-NOT need little help to do universal quantum computation. arXiv:quant-ph/0205115, 2002. (http://arxiv.org/abs/quant-ph/0205115).

[10] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing, 26(5):1484-1509*, 2005. (also, arXiv:quant-ph/9508027)

[11] S. Hallgren. Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. *Proceedings of the 34th ACM Symposium on Theory of Computing*, 2002.

[12] S. Hallgren. Fast quantum algorithms for computing the unit group and class group of a number field. *Proceedings of the 37th ACM Symposium on Theory of Computing*, 2005

[13] A. Schmidt, U. Vollmer. Polynomial time quantum algorithm for the computation of the unit group of a number field. *Proceedings of the 37th Symposium on the Theory of Computing, pg. 475-480*, 2005.

[14] W. van Dam, S. Hallgren, L. Ip. Quantum algorithms for some hidden shift problems. *SIAM Journal on Computing, 36(3):763-778*, 2006. (also, arXiv:quant-h/0211140)

[15] A. Harrow, A. Hassidim, S. Lloyd. Quantum algorithm for solving linear systems of equations. *Physical Review Letters 15(103):150502*, 2009. (also, arXiv:0811.3171)

[16] J. Watrous. Quantum algorithms for solvable groups. *Proceedings of the 33rd ACM Symposium on Theory of Computing, pages 60-67*, 2001. (also, arXiv:quant-ph/0011023)

[17] F. Le Gall. An efficient quantum algorithm for some instances of the group isomorphism problem. Proceedings of STACS 2010. (also, arXiv:1001.0608)

[18] M. Freedman, A. Kitaev, Z. Wang. Simulation of topological field theories by quantum computers. *Communications in Mathematical Physics, 227:587-603*, 2002.

[19] M. Freedman, M. Larsen, Z. Wang. A modular functor which is universal for quantum computation. *Comm. Math. Phys. 227(3):605-622*, 2002. (also, arXiv:quant-ph/0001108)

[20] H. Buning, T. Lettmann. Propositional logic: deduction and algorithms. *Cambridge University Press, Ltd*, 1999.

[21] A. Shende, A. Prasad, I. Markov, J. Hayes. Synthesis of Reversible Logic Circuits. *IEEE Trans. CAD 22(6):710 - 722*, 2003.

[22] T. Toffoli. Reversible Computing. newblock *ICALP: 632-644* , 1980

[23] M. Nielsen, I. Chuang. Quantum Computation and Quantum Information. *Cambridge University Press*, 2000.

[24] S. Clark, R. Jozsa, N. Linden. Generalized Clifford groups and simulation of associated quantum circuits. *Quant Inf Comp 8:106 - 126*, 2008.

[25] A. Barenco et al. Elementary gates for quantum computation. *Physical Review A 52(5):3457-3467*, 1995.

[26] P. Boykin et al. A new universal and fault-tolerant quantum basis. *Information Processing Letters, 75(3):101-107*, 2000.

[27] A. Bocharov, K. Svore. A Depth-Optimal Canonical Form for Single-qubit Quantum Circuits. *Physical Review Letters 109:190501*, 2012. (also, arXiv:1206.3223)

[28] V. Kliuchnikov, D. Maslov, M. Mosca. Asymptotically optimal approximation of single qubit unitaries by Clifford and T circuits using a constant number of ancillary qubits. `arXiv:1212.0822`, 2012.

[29] V. Kliuchnikov, D. Maslov, M. Mosca. Practical approximation of single-qubit unitaries by single-qubit quantum Clifford and T circuits. `arXiv:1212.6964`, 2012.

[30] B. Giles, P. Selinger. Exact synthesis of multiqubit Clifford+T circuits. `arXiv:1212.0506`, 2012.

[31] P. Selinger. Efficient Clifford+T approximation of single-qubit operators. arXiv:1212.6253, 2012.

[32] S. Bravyi, A. Kitaev. Universal Quantum Computation with ideal Clifford gates and noisy ancillas. *Physical Review A 71:022316*, 2005. also, arXiv:quant-ph/0403025, 2004.

[33] B. Reichardt. Improved magic states distillation for quantum universality. http://arxiv.org/pdf/quant-ph/0411036.pdf, 2004.

[34] A. Meier, B. Eastin, E. Knill. Magic-state distillation with the four-qubit code. arXiv:1204.4221, 2012.

[35] A. Bocharov, Y. Gurevich, K. Svore. Efficient Decomposition of Single-Qubit Gates into $V$ Basis Circuits. arXiv:1303.1411, 2013.

[36] C. Dawson, M. Nielsen. The Solovay-Kitaev Algorithm. arXiv:quant-ph/0505030, 2005. (http://arxiv.org/abs/quant-ph/0505030).

[37] A. Y. Kitaev. Quantum computations: algorithms and error correction. *Russ. Math. Surv., 52(6):1191-1249*, 1997.

[38] F. Bloch. Nuclear induction. Phys. Rev. 70(7-8) (460), 1946.

[39] A. Harrow, B. Recht, I. Chuang. Efficient Discrete Approximations of Quantum Gates. *J. Math. Phys. 43:4445*, 2002. (also, arXiv:quant-ph/0111031), 2001

[40] B. Eastin. Distilling one-qubit magic states into Toffoli states . arXiv:1212.4872, 2012

[41] Cody Jones. Novel constructions for the fault-tolerant Toffoli gate. *Phys. Rev. A 87:022328* , 2013. (also, arXiv:1212.5069), 2013

[42] Cody Jones. Composite Toffoli gate with two-round error detection. arXiv:1303.6971, 2013

[43] J. Conway. A Course in Functional Analysis. Springer, 1990.