

## Chapter 3

# Virtual Realizations of Quantum Computers

- March 20, 2021 (v1.6)

In the previous chapter, we have seen how quantum computation works under the assumption that the elementary quantum logic gate operations are available. But how can one build a quantum computer, a machine, that allows such quantum logic gates? Quantum computers are physical systems and the implementation of all the quantum logic gates are governed by the laws of physics. In this section, we discuss the basic physical principles that are directly involved in the implementation of quantum logic gates. Through the course of discussion, it will become clearer what basic conditions and requirements should be fulfilled to build a quantum computer.

By now there are many quantum computer architectures that are not only proposed and tested at the research level but also actually running. However, to understand each architecture requires a certain level of knowledge about the physical systems. For example, to understand a quantum computer based on superconducting circuits, one has to first understand the superconductivity, the Josephson effect, the flux quantization, the Josephson inductance (a sort of non-linear kinetic inductance), and the interaction of superconducting circuits to electromagnetic fields. Such discussions often hinder access to the essential part of the operating principle of the quantum computer, and are out of the scope of this workbook.

Here we consider an idealistic and minimal quantum system that is suitable for quantum computation and discuss how to control it to implement the desired quantum logic gates. It is certainly impractical. Nevertheless it will point out the crucial requirements when one wants to actually develop an quantum computer based on realistic systems and devices. Through the discussions, we will indicate how the relevant parts are related to actual quantum computer architectures.



Figure 3.1: An internal view of IBQ quantum computer based on superconducting circuits.

### 3.1 Quantum Bits

We have already noted several times that the building blocks and basic computational units of a quantum computer are qubits. Ideally, a qubit is associated with a two-dimensional Hilbert space. In reality, the Hilbert space for any realisitc system is infinite-dimensional, and a qubit usually refers to *certain degrees of freedom* that are relatively independent of other degrees of freedom. For example, the spin of electron or the polarization of photon has exactly two-dimensional Hilbert space. In many cases, a qubit may also refer to a *certain two-dimensional subspace* of a larger Hilbert space that are decoupled or separated relatively well from the rest. For example, a superconducting charge qubit refers to a two lowest-energy charnging states in a small—hundreds nanometers in lateral size—superconducting island.

However, a well-defined two-dimensional Hilbert space (or subspace) does not necessarily mean that the degrees of freedom in question qualifies as a qubit. For example, consider the spin of neutron. Although its Hilbert space dimension is certainly two, you recognize that it can hardly be used for quantum computatoin. It is hard to isolate a neutron, and even more so to manipulate its spin in a reliable and tunable manner. Then what requirements should qubits—individually and as a whole collection—meet to build a practical quantum computer? Apart from specific technical issues in specific systems, these are the basic requirements—the so-called DiVincenzo criteria—commonly rated to assess the potential of a particular architecture in consideration ([DiVincenzo, 2000](#)):

- (a) The qubits should be well characterized and form a scalable system. For each qubit, the Hilbert space should be well defined in the sense mentioned above and its internal Hamiltonian including the parameters needs to be accurately known. The qubits must also admit genuine interactions among

them and maintain the characteristics up to a sufficiently large scale for practical computation.

- (b) The qubits should allow initialization to a fixed logical basis state. Even though any quantum algorithm assumes superposition in the middle of the process, all computations must start from a known value. This straightforward requirement is the same even for classical computers. One of the common approaches for initialization is to cool down the system and wait for it to relax to the ground state. Another method is to perform a projection measurement in the logical basis so as for the state to collapse to the logical basis state corresponding to the measurement outcome.
- (c) The qubits should maintain coherence long enough for the desired gate operations. The superposition between different logical basis states is a crucial difference distinguishing quantum computers from classical computers. Unfortunately, qubits are subject to various decoherence effects due to external control circuits and measuring devices and eventually lose quantumness. The system should maintain the coherence during the desired gate operations to get a reliable result out of the computation.
- (d) The system of qubits should allow a *universal* set of quantum gate operations. As discussed in Chapter 2, quantum computation is to achieve a desired unitary transformation with a combination of certain elementary gate operations that are acting on a single qubit or two qubits at a time. Below we will discuss the physical implementations of those elementary quantum logic gates.
- (e) The system should allow measurements in the logical basis. At the end of a computation, the result needs to be read out, and it is achieved by performing measurements on specific qubits. The capability of accurate measurement is called the *quantum efficiency*. Ideal measurement has 100% quantum efficiency. Less than 100 % quantum efficiency in measurements leads to a tradeoff with other resources. For example, if a computation is desired with 97 % reliability while measurements have only 90 % quantum efficiency, then one must repeat the computation three times or more.

In the rest of the chapter, let us now focus on the manipulation of quantum states of qubits, which naturally forms the largest part of quantum computation.

Consiser a quantum computer consisting of  $n$  qubits. Let  $\mathcal{S}_j$  ( $j = 1, \dots, n$ ) be the 2-dimentional Hilbert space associated with the  $j$ th single qubit. An ideal quantum computer would realize a Hamiltonian on  $\mathcal{S}_1 \otimes \dots \otimes \mathcal{S}_n$  of the form

$$\hat{H}(t) = \sum_j \sum_{\mu} B_j^{\mu}(t) \hat{S}_j^{\mu} + \sum_{ij} \sum_{\mu\nu} J_{ij}^{\mu\nu}(t) \hat{S}_i^{\mu} \hat{S}_j^{\nu}, \quad (3.1)$$

where  $\hat{S}_j^{\mu}$  ( $\mu = x, y, z$ ) are the Pauli operators—see Section 2.1.1—on  $\mathcal{S}_j$ .

The parameter  $B_j^\mu$  directly controls the  $j$ th qubit; physically, it plays the same role as the magnetic field on a spin. In realistic systems, it may be hard to address single qubits individually. How freely single qubits can be manipulated strongly depends on how many of the parameters  $B_j^\mu$  the system allows to be tunable accurately. See, for example, Section 3.2 below.

The control parameters  $J_{ij}^{\mu\nu}$  describe the (hypothetical) exchange coupling between the  $i$ th and  $j$ th qubits. In principle, any type of interaction between two qubits can be used to implement CNOT gate (see Section 3.2 for examples) although the actual implementation may require more than one interactions and many additional single-qubit operations depending on the particular type of coupling. Therefore, an accurate control of the coupling parameters  $J_{ij}^{\mu\nu}$  between a specific pair of qubits is essential for universal quantum computation. In realistic systems, the coupling parameters  $J_{ij}^{\mu\nu}$  are even more difficult to realize. First of all, in many architectures the connectivity of qubits that are not in direct proximity of each other is seriously limited. Further, dynamically turning on and off the coupling is often forbidden. In many cases, in order to achieve a sizable strength, the exchange couplings are kept turned on throughout the whole computation processes. Such difficulties and imperfections all contribute to the errors in the computational outputs.

---

We will be denoting each qubit by the symbol S and accompanying indices .

**Let[Qubit, S]**

The Pauli operators are specified by the last index. For example, the Pauli operator  $S_j^x$  is denoted by  $S[j,1]$ .

*In[1]:=*  $S[j, 1]$

*Out[1]=*  $S_j^x$

*In[2]:=*  $S[j, 1] ** S[j, 2]$

$S[j, 1] ** S[k, 2]$

*Out[2]=*  $i S_j^z$

*Out[3]=*  $S_j^x S_k^y$

## 3.2 Dynamical Scheme

The dynamic scheme implements the desired quantum gate operations by means of the time-evolution operator governing the dynamics of the physical qubits in a quantum computer; hence the name. It is the most common scheme of quantum computation and a majority of quantum computers demonstrated so far are based on it.

### 3.2.1 Implementation of Single-Qubit Gates

Conceptually, the most straightforward way to control a single qubit is to apply the static parameters  $B^x$ ,  $B^y$ , and  $B^z$  for a certain period  $\tau$  of operation time. We refer to them collectively by a vector  $\mathbf{B} = (B^x, B^y, B^z)$ , which can be regarded as a fictitious magnetic field—here, the dimension of  $\mathbf{B}$  is energy unlike the real magnetic field. The Hamiltonian for the qubit is given by

$$\hat{H} = \frac{1}{2} \mathbf{B} \cdot \hat{\mathbf{S}}, \quad (3.2)$$

where  $\hat{\mathbf{S}} := (\hat{S}^x, \hat{S}^y, \hat{S}^z)$  is the vector of the Pauli operators on  $\mathcal{S}$ . The time evolution is then governed by the unitary operator

$$\hat{U}(t) = \exp(-it\hat{H}) = \exp(-it\mathbf{B} \cdot \hat{\mathbf{S}}/2). \quad (3.3)$$

It has the same form as the single-qubit rotation operator in Eq. (2.24), Section 2.1.3: It describes the rotation in the Pauli space—or the Bloch sphere—around the axis parallel to the vector  $\mathbf{B}$  by the angle  $\phi(t) = |\mathbf{B}|t$ . When the involved two-level system is a true spin, such a rotation in the Pauli space is called the *Larmor precession*.

---

Consider a qubit. Let us apply a (fictitious) magnetic field.

```
In[1]:= Let[Real, B]
opH = Dot[B@{1, 2, 3}, S@{1, 2, 3}] / 2
Out[1]=  $\frac{1}{2} (B_1 S^x + B_2 S^y + B_3 S^z)$ 
```

To be specific, we consider the case with the magnetic field between the x- and z-axis in the xz-plane. The factor of  $1/\sqrt{2}$  is to set the energy (frequency) scale to unit.

```
In[2]:= B[1] = B[3] = 1/Sqrt[2]; B[2] = 0;
opH
```

```
Out[2]=  $\frac{1}{2} \left( \frac{S^x}{\sqrt{2}} + \frac{S^z}{\sqrt{2}} \right)$ 
```

```
In[3]:= opU[t_] = MultiplyExp[-I t opH]
opU[t] // Elaborate
```

```
Out[3]=  $e^{-\frac{1}{2} I t \left( \frac{S^x}{\sqrt{2}} + \frac{S^z}{\sqrt{2}} \right)}$ 
```

```
Out[4]=  $\cos\left(\frac{t}{2}\right) - \frac{i S^x \sin\left(\frac{t}{2}\right)}{\sqrt{2}} - \frac{i S^z \sin\left(\frac{t}{2}\right)}{\sqrt{2}}$ 
```

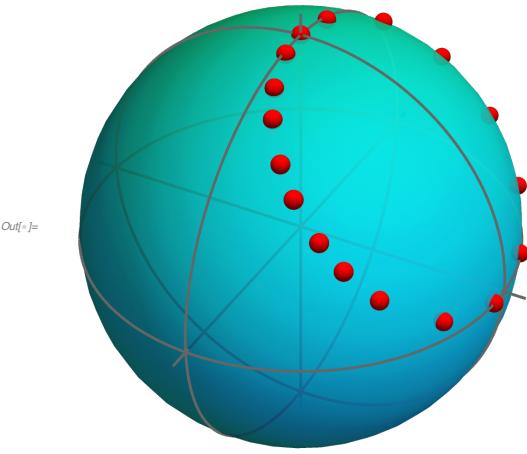
```
In[5]:= vec[t_] = opU[t] ** Ket[] // Elaborate
```

```
Out[5]=  $\frac{i |1s\rangle \sin\left(\frac{t}{2}\right)}{\sqrt{2}} + |-\rangle \left( \cos\left(\frac{t}{2}\right) - \frac{i \sin\left(\frac{t}{2}\right)}{\sqrt{2}} \right)$ 
```

This illustrates the Larmor precession (with the qubit regarded as a “spin”).

**Technical Note:** You need Chop because numerical errors sometimes give  $0.+Ket[...]$ , which cannot be handled properly by BlochVector.

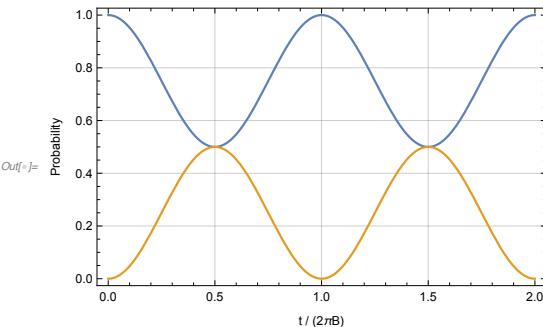
```
In[5]:= vv = BlochVector /@ Table[Chop@vec[t], {t, 0, 8, 0.5}];
BlochSphere[{Red, Bead /@ vv}]
```



Let us now examine the probabilities for the qubit to be found in the logical basis states.

```
In[6]:= prob[t_] = Abs[Normal@Matrix@vec[t]]^2
Out[6]= {Cos[t/2] - I Sin[t/2]^2, 1/2 |Sin[t/2]|^2}
```

```
In[7]:= Plot[Evaluate@prob[2 Pi t], {t, 0, 2},
FrameLabel -> {"t / (2πB)", "Probability"}]
```



Although conceptionally simple, the above method has limited applications in many realistic systems. For example, in the presence of other levels, one cannot apply the method selectively between the two levels in question. More widely applicable method is to apply an oscillating field: Suppose that

$$B^x = B_{\perp} \cos(t\omega), \quad B^y = B_{\perp} \sin(t\omega), \quad B^z = B_0. \quad (3.4)$$

One can regaard it as a fictitious magnetic field precessing around the  $z$ -axis with frequency  $\omega$ . The Hamiltonian now depends on time and is given by

$$\hat{H} = \frac{1}{2} B_{\perp} [\cos(t\omega) \hat{S}^x + \sin(t\omega) \hat{S}^y] + \frac{1}{2} B_0 \hat{S}^z. \quad (3.5)$$

Recalling the property in Eq. (2.26) of the Pauli operators as the generators of rotation, we observe that

$$\hat{U}_z^\dagger(t\omega)\hat{H}(t)\hat{U}_z(t\omega) = \frac{1}{2}B_\perp\hat{S}^x + \frac{1}{2}B_0\hat{S}^z \quad (3.6)$$

does not depend on time any longer. This observation suggests that the dynamics looks simpler in the rotating frame. As the rotating frame is not an inertial frame, one has to take into account the non-inertial effect—corresponding to the classical *inertial force*: The state vectors  $|\psi(t)\rangle$  and  $|\psi_R(t)\rangle$  in the lab and rotating frame, respectively, are related by

$$|\psi_R(t)\rangle = \hat{U}_z^\dagger(t\omega)|\psi(t)\rangle. \quad (3.7)$$

Putting it into the original Schrödinger equation for  $|\psi(t)\rangle$ ,

$$i\partial_t|\psi\rangle = \hat{H}(t)|\psi\rangle, \quad (3.8)$$

and operating  $\hat{U}_z^\dagger(t)$  from the left on both sides, one can get the Schrödinger equation in the rotating frame,

$$i\partial_t|\psi_R\rangle = \hat{H}_R|\psi_R\rangle, \quad (3.9)$$

where the Hamiltonian in the rotating frame is given by

$$\hat{H}_R(t) := U_z^\dagger(t)\hat{H}(t)\hat{U}_z(t) - \hat{U}_z^\dagger(t)[i\partial_t\hat{U}_z(t)]. \quad (3.10)$$

The second term in  $\hat{H}_R$  is responsible for the non-inertial effect. As already expected in the above observation, the rotating-frame Hamiltonian does not depend on time any longer,

$$\hat{H}_R = \frac{1}{2}B_\perp\hat{S}^x + \frac{1}{2}(B_0 - \omega)\hat{S}^z. \quad (3.11)$$

The time-evolution operator in the rotating frame,

$$\hat{U}_R(t) = \exp(-it\hat{H}_R) \quad (3.12)$$

is *formally* the same as the one (3.3) in the lab frame. It describes a precession around the axis parallel to  $\mathbf{B}_R := (B_\perp, 0, B_0 - \omega)$  with the frequency

$$\Omega_R := \sqrt{B_\perp^2 + (B_0 - \omega)^2}. \quad (3.13)$$

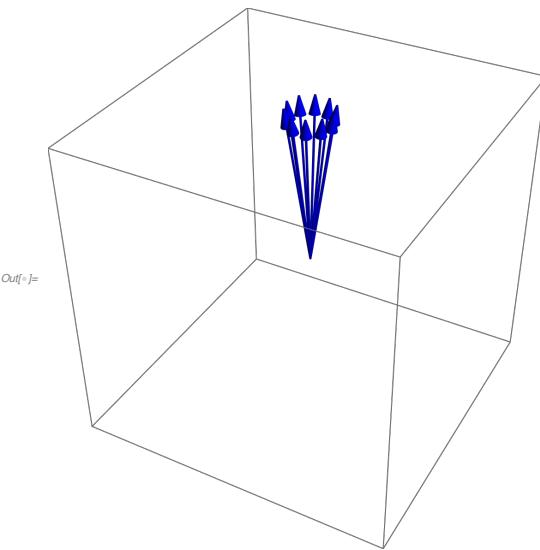
The precession in the rotating frame is called the *Rabi oscillation* and the frequency in (3.13) is called the *Rabi frequency*. The fictitious magnetic field in the rotating frame—compare Eqs. (3.2) and (3.10)—is almost along the  $x$ -axis for  $\omega \approx B_0$ , that is, the time-evolution  $\hat{U}_R(t)$  in the rotating frame corresponds to the rotation around the  $x$ -axis in the Bloch sphere. In this case, the qubit can make a full transition from the initial state  $|0\rangle$  to the orthogonal state  $|1\rangle$  by properly tuning

the operation time and/or the parameter  $B_{\perp}$ . In this sense, when  $\omega \approx B_0$ , the system is said to be at *resonance*. As the driving frequency  $\omega$  gets away off the resonance, the maximum transition probability becomes smaller and smaller. This resonance behavior allows to induce transitions between a selected pair of two levels among many others.

---

Let us now apply a time-dependent field. It precesses around the z-axis. Note that typically,  $B_3 \gg B_1, B_2$ .

```
In[7]:= ω = 2 Pi;
B[1] = Cos[ω t];
B[2] = Sin[ω t];
B[3] = (11/10) ω;
Graphics3D[{Blue, Table[Arrow@Tube@{{0, 0, 0}, B@{1, 2, 3}}, {t, 0, 1, 0.1}]},
PlotRange → ω {{-1, 1}, {-1, 1}, {-1, 1}}]
```



```
In[8]:= opH = Dot[B@{1, 2, 3}, S@{1, 2, 3}] / 2
Out[8]= 1/2 (Cos[2 π t] Sx + 11 π Sz/5 + Sy Sin[2 π t])
```

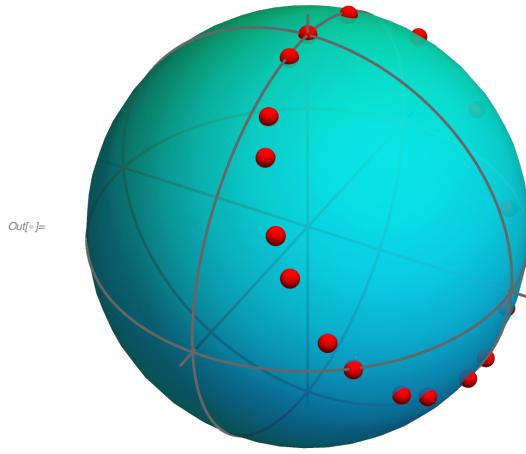
```
In[9]:= opHR = Rotation[-ω t, S[3]] ** opH ** Rotation[ω t, S[3]] - ω S[3] / 2 // Elaborate
Out[9]= Sx/2 + π Sz/10
```

```
In[10]:= opUR[t_] = MultiplyExp[-I t opHR]
Out[10]= E^{-I t (Sx/2 + π Sz/10)}
```

```
vec[t_] = opUR[t] ** Ket[] // Elaborate;
```

This illustrates the precession in the rotating frame, which is called the Rabi oscillation.

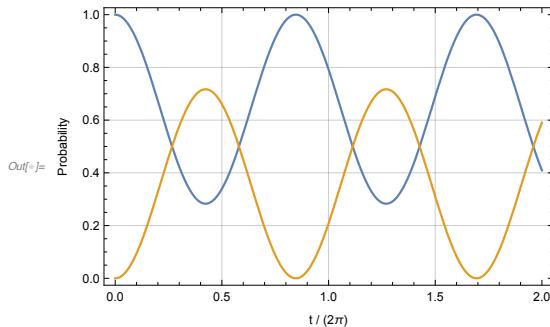
```
In[5]:= vv = BlochVector /@ Table[Chop@vec[t], {t, 0, 8, 0.5}];
BlochSphere[{Red, Bead /@ vv}]
```



This shows the transition probabilities in the logical basis states. Note that the probabilities are the same both in the lab and rotating frame. The Rabi oscillation frequency is in this particular example is close to one---it is exactly one at resonance.

```
prob[t_] = Abs[Normal@Matrix@vec[t]]^2;
```

```
In[6]:= Plot[Evaluate@prob[2 Pi t], {t, 0, 2},
FrameLabel -> {"t / (2π)", "Probability"}]
```



Rotating-wave approximation ...

.....

### 3.2.2 Implementation of CNOT

CNOT is a vital gate operation in any quantum algorithms. Seemingly simple, it is not trivial to physically realize in realistic system. A typical obstacle is that the Hamiltonian, in particular the coupling between two qubits, is restricted. Here we take a few examples of the qubit-qubit interaction and see how one can use it to implement the CNOT gate. These examples should be enough to give general idea

about what is required for the implementation of CNOT or similar gate operations in a given realistic physical system.

$$\text{CNOT} = \sum_{z=0}^1 |z\rangle\langle z| \otimes \hat{X}^z = i \sum_z |z\rangle\langle z| \otimes e^{-i\pi z \hat{X}} \quad (3.14)$$

As  $z = 0, 1$  are eigenvalues of  $|1\rangle\langle 1| = (\hat{I} - \hat{Z})/2$ ,

$$\text{CNOT} = i \exp \left[ -i \frac{\pi}{2} (\hat{I} - \hat{Z}) \otimes \hat{X} \right] \quad (3.15)$$

Therefore, one can implement a CNOT gate with the first and second qubit as the control and target qubit, respectively, in terms of the Hamiltonian (up to an irrelevant global phase factor  $i$ )

$$\hat{H} = \frac{\pi}{2} \hat{S}_2^x - \frac{\pi}{2} \hat{S}_1^z \hat{S}_2^x . \quad (3.16)$$

### 3.3 Geometric/Topological Scheme

When a system undergoes a cyclic adiabatic process starting, the system remains in the same energy level without making a transition to other energy levels. However, the quantum state of the system acquires a phase factor from two contributions. One is responsible for the usual dynamical accumulation and the other results from the geometric properties of the parameter space. The latter is called the *geometric phase* of the cyclic adiabatic process (Berry, 1984). When the energy level is degenerate and associated with a multi-dimensional eigen-subspace, the geometric phase becomes non-Abelian, that is, the quantum state evolves to another state within the subspace. The unitary transformation between the initial and final state is called the *non-Abelian geometric phase* (Wilczek & Zee, 1984). Non-Abelian geometric phase can be extended to any *cyclic evolution*, without restriction by the adiabatic condition (Aharonov & Anandan, 1987; Anandan, 1988).

The *geometric scheme* of quantum computation (or simply *geometric quantum computation* for short) is to implement the unitary gate operations by means of the non-Abelian geometric phases (Zanardi & Rasetti, 1999; Sjöqvist *et al.*, 2012). The geometric scheme is stable against random fluctuations of the parameters as it depends on the geometric path in the parameter space rather than the detailed time-dependence of the parameters.

In this section, we will give a brief overview of the geometric scheme. Consider a cyclic process where the Hamiltonian changes in time through the control parameters  $\lambda_\mu$  with  $\mu = 1, 2, \dots$

$$\hat{H}(t) = \hat{H}(\boldsymbol{\lambda}(t)) \quad (3.17)$$

where we adopted a row-vector notation

$$\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_\mu, \dots) \quad (3.18)$$

to collectively denote the control parameters. When the Hamiltonian  $\hat{H}(t)$  is the idealistic one in (3.1) for qubits, of course, the control parameters  $\boldsymbol{\lambda}$  refer to the parameters  $B_j^\mu$  and  $J_{ij}^\mu$ . However, the Hamiltonian for a geometric quantum computation is usually much more general and does not contain elements manifesting peculiar form of qubits. The logical qubits of a geometric scheme is often implicitly encoded into the subspace undergoing the cyclic evolution.

Let the states  $|\alpha_j(t)\rangle$  ( $j = 1, 2, \dots$ ) form an *instantaneous basis* of the Hilbert space  $\mathcal{H}$ . These states are often chosen to be the instantaneous eigenstates of the Hamiltonian  $\hat{H}(t)$ , but the choice is completely arbitrary as long as they are *cyclic*,  $|\alpha_j(\tau)\rangle = |\alpha_j(0)\rangle$ , and *smooth* enough to be differentiable with respect to time (and hence the parameters  $\lambda_\mu$ ) and . Suppose that a physical state  $|\psi(t')\rangle$  at a given time  $t'$  is expanded in the instantaneous basis  $|\alpha_j(t')\rangle$  as

$$|\psi(t')\rangle = \sum_j |\alpha_j(t')\rangle \langle \alpha_j(t')| \psi(t') \rangle. \quad (3.19)$$

At later time  $t > t'$ , it must be expanded in another instantaneous basis  $|\alpha_i(t)\rangle$  in the form

$$|\psi(t)\rangle = \sum_i |\alpha_i(t)\rangle U_{ij}(t, t') \langle \alpha_j(t')| \psi(t') \rangle, \quad (3.20)$$

where  $U_{ij}(t, t')$  is a unitary matrix which describes the basis change to  $|\alpha_i(t)\rangle$  from  $|\alpha_j(t')\rangle$ . Putting (3.20) into the Schrödinger equation, one can see that the unitary matrix  $U(t, t')$  satisfies the dynamical equation

$$i \frac{\partial}{\partial t} U(t, t') = [H(t) - iA(t)]U(t, t'), \quad (3.21)$$

where  $H(t)$  is the matrix representation of the Hamiltonian  $\hat{H}(t)$  in the instantaneous basis  $|\alpha_j(t)\rangle$  and the matrix  $A(t)$  has been defined by

$$A_{ij}(t) := \langle \alpha_i(t) | \frac{d}{dt} |\alpha_j(t)\rangle. \quad (3.22)$$

The solution to the equation (3.21) is formally given by

$$U(t, t') = T \exp \left[ - \int_{t'}^t ds \{ A(s) + iH(s) \} \right], \quad (3.23)$$

where  $T$  denotes the time ordering.

So far, everything has been completely general. A geometric quantum computation requires to identify a dynamic subspace  $\mathcal{K}(t) \subset \mathcal{H}$  such that

- (a)  $\mathcal{K}(t)$  undergoes a *cyclic evolution*,  $\mathcal{K}(\tau) = \mathcal{K}(0)$ ;

(b)  $\hat{H}(t)$  vanishes within  $\mathcal{K}(t)$  at all time  $0 < t < \tau$ .

With these two conditions satisfied, the unitary matrix governing the evolution of the state vector  $|\psi(t)\rangle$  in (3.20) is determined solely by the matrix  $A(t)$ . Furthermore, since

$$A_{ij}(t) = \sum_{\mu} \frac{d\lambda_{\mu}}{dt} \langle \alpha_i | \partial_{\mu} \alpha_j \rangle , \quad (3.24)$$

the unitary matrix  $U(\tau, 0)$  for the whole cycle is completely determined by the closed path in the parameter space that is traversed by the parameters  $\lambda_{\mu}(t)$ . That is, for a given loop  $\mathcal{C}$  in the parameter space, the unitary matrix  $U(\tau, 0) = U(\mathcal{C})$  is given by

$$U(\mathcal{C}) = P \exp \left[ - \sum_{\mu} \oint_{\mathcal{C}} d\lambda_{\mu} A^{\mu} \right] , \quad (3.25)$$

where  $P$  denotes the path ordering.

In short, the geometric quantum computation is to implement the quantum gate operations by means of the unitary transformation  $U(\mathcal{C})$  in (3.25). The computational space is identified by the subspace  $\mathcal{K}$  undergoing a cyclic evolution. Different choices of the closed path  $\mathcal{C}$  results in different quantum logic gates.

As an example, let us consider a toy mode

$$\hat{H} = \epsilon |0\rangle\langle 0| - \sum_{j=1}^3 (\Omega_j |j\rangle\langle j| + h.c.) \quad (3.26)$$

consisting of four levels  $|j\rangle$  ( $j = 0, 1, 2, 3$ ). For simplicity, assume  $\Omega_j \in \mathbb{R}$ . There is one “bright state” (unnormalized)

$$|\Omega\rangle := \sum_{j=1}^3 |j\rangle \Omega_j \quad (3.27)$$

and two “dark states”  $|D_j\rangle$  ( $j = 1, 2$ ) that are orthogonal to both  $|\Omega\rangle$  and  $|0\rangle$ . In this toy model, the two dark states form a degenerate subspace  $\mathcal{K}$  of our interest. It is convenient to take the parameterization

$$\Omega_1 = \sin \theta \cos \phi, \quad \Omega_2 = \sin \theta \sin \phi, \quad \Omega_3 = \cos \theta \quad (3.28)$$

which leads to

$$|\Omega\rangle = |1\rangle \sin \theta \cos \phi + |2\rangle \sin \theta \sin \phi + |3\rangle \cos \theta \quad (3.29)$$

and

$$|D_1\rangle = |1\rangle \sin \phi - |2\rangle \cos \phi \quad (3.30a)$$

$$|D_2\rangle = |1\rangle \cos \theta \cos \phi + |2\rangle \cos \theta \sin \phi - |3\rangle \sin \theta \quad (3.30b)$$

Now consider the non-Abelian geometric phase for the path with fixed  $\Omega_3$  (i.e.,  $\theta$ ). We see that

$$A_{jk}(\phi) := \langle D_j | \partial_\phi D_k \rangle = -i \cos \theta \sigma_{jk}^y, \quad (3.31)$$

where  $\sigma^y$  is a Pauli matrix. As a result, the cyclic evolution gives rise to the following unitary transformation

$$U = \text{Pexp} \left( - \oint d\phi A \right) = \exp(i\sigma^y 2\pi \cos \theta) = U_y(4\pi \cos \theta), \quad (3.32)$$

where  $U_y(\varphi)$  is the rotation around the  $y$ -axis by angle  $\varphi$ .

The *topological scheme* of quantum computation is a peculiar case of geometric scheme.

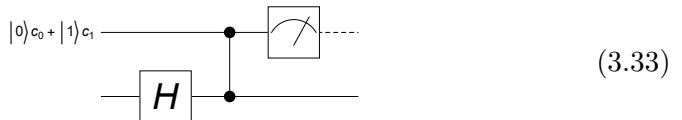
...

## 3.4 Measurement-Based Scheme

The fundamental postulates of quantum mechanics discussed at the very beginning of the book states that the quantum state of a system changes as a consequence of two difference causes. One is the time-evolution governed by the Schrödinger equation. The dynamical and geometric scheme of quantum computation are both ultimately based on the time evolution. The other cause for the change of quantum states is the collapse of quantum states after measurement (see Postulate 3). Can we also use it for quantum computation? At a first glance, it look difficult because of the uncontrollable nature of the quantum state collapse. Recently it have been found that a quantum computation is possible completely based on measurement provided the system is prepared in a special quantum state called the *cluster state* or *graph state* (Raussendorf & Briegel, 2001; Raussendorf *et al.*, 2002). Such a scheme of quantum computation is called the *measurement-based quantum computation* or *one-way quantum computation*—“one-way” for the reason to be clear below.

...

**Example 9** Consider the following quantum circuit



Show that the circuit transforms the product state into an entangled state of the form

$$(|0\rangle c_0 + |1\rangle c_1) \otimes |0\rangle \mapsto |0\rangle \otimes (\hat{H}|0\rangle)c_0 + |1\rangle \otimes (\hat{H}|1\rangle)c_1. \quad (3.34)$$

Unlike a single CNOT gate, which copies the state of the control qubit and generates the entanglement as shown in Eq. (2.33), the Hadamard gate is operated on the target qubit. Hint: Note that  $\hat{Z}\hat{H} = \hat{H}\hat{X}$ .

**(solution)** Via the controlled-Z gate (CZ),

$$(|0\rangle c_0 + |1\rangle c_1) \otimes \hat{H} |0\rangle \xrightarrow{\text{CZ}} |0\rangle c_0 \otimes \hat{H} |0\rangle + |1\rangle c_1 \otimes \hat{Z}\hat{H} |0\rangle = |0\rangle c_0 \otimes \hat{H} |0\rangle + |1\rangle c_1 \otimes \hat{H} |1\rangle \quad (\dagger 1)$$

Namely,

$$(|0\rangle c_0 + |1\rangle c_1) \otimes \hat{H} |0\rangle \xrightarrow{\text{CZ}} |0\rangle \otimes \hat{H} |0\rangle c_0 + |1\rangle \otimes \hat{H} |1\rangle c_1 \quad (\dagger 2)$$

Consider an arbitrary state that the first qubit can have.

```
In[7]:= vec = Ket[] <| c[0] + Ket[S[1] -> 1] <| c[1];
vec // LogicalForm
Out[7]= c[0] |0_{S[1]}> + c[1] |1_{S[1]}>
```

This shows the most elementary quantum circuit model that is used for a measurement-based quantum computation.

```
In[8]:= qc = QuissoCircuit[vec, LogicalForm[Ket[], S[2]], S[2, 6], CZ[S[1], S[2]], "PortSize" -> {2, 0.5}];
qc
Out[8]= |0> + c[1] |1> ---> |0> -> H --->
```

Here is the result of the operations of the gates.

```
In[9]:= out = Elaborate[qc]
out
Out[9]= c[0] |-> -> c[0] |1_{S[1]S[2]}> + c[1] |1_{S[1]}> + c[0] |1_{S[2]}> /> sqrt[2]
```

It is useful to rewrite the resulting state in the following form.

```
In[10]:= new = QuissoFactor[out, S[1]];
new // LogicalForm
new
Out[10]= |0_{S[1]}> ⊗ (c[0] |0_{S[2]}> /> sqrt[2] + c[0] |1_{S[2]}> /> sqrt[2]) + |1_{S[1]}> ⊗ (c[1] |0_{S[2]}> /> sqrt[2] - c[1] |1_{S[2]}> /> sqrt[2])
```

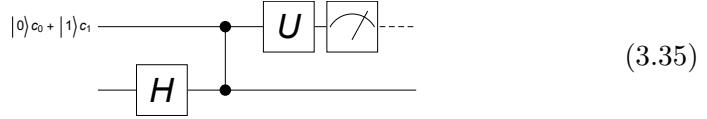
Using the relation between the CNOT and CZ gate, the following is an equivalent quantum circuit model.

```
In[11]:= Row@{QuissoCircuit[CNOT[S[1], S[2]]], "==" , QuissoCircuit[S[2, 6], CZ[S[1], S[2]], S[2, 6]]}
Out[11]= |0> ---> |0> ⊕ |1> = |0> ---> H ---> |0> ⊕ H --->
```

```
In[12]:= new = QuissoCircuit[vec, LogicalForm[Ket[], S[2]], CNOT[S[1], S[2]], S[2, 6], "PortSize" -> {2, 0.5}];
new
Out[12]= |0> + c[1] |1> ---> |0> ⊕ H --->
```

In[ $\cdot$ ] := Elaborate[qc] -> Elaborate[new]  
 Out[ $\cdot$ ] =  $\emptyset$

**Example 10** Consider the quantum circuit



with an arbitrary unitary operator  $\hat{U}$ . Show that the most general state  $|\psi\rangle_{\text{out}}$  resulting in the second qubit has the form

$$|\psi\rangle_{\text{out}} = \hat{H}\hat{U}_z(\phi)|\psi\rangle_{\text{in}}, \quad \hat{U}_z(\phi) := e^{-i\sigma^z\phi/2}, \quad \phi \in \mathbb{C} \quad (3.36)$$

up to a global phase factor.

Note that the angle  $\phi \in \mathbb{C}$  is complex in general and depends on the measurement outcome as well as the unitary gate  $U$  (or, equivalently, the measurement basis).

**(solution)** According to Example 9,

$$(|0\rangle c_0 + |1\rangle c_1) \otimes \hat{H}|0\rangle \xrightarrow{\text{CZ}} |0\rangle \otimes \hat{H}|0\rangle c_0 + |1\rangle \otimes \hat{H}|1\rangle c_1 \quad (\dagger 3)$$

$\hat{U}$  on qubit 0:

$$\begin{aligned} \hat{U}|0\rangle c_0 \otimes \hat{H}|0\rangle + \hat{U}|1\rangle c_1 \otimes \hat{H}|1\rangle &= \sum_{mj} |m\rangle U_{mj} c_j \otimes \hat{H}|j\rangle \\ &= \sum_m |m\rangle \otimes \sum_j \hat{H}|j\rangle U_{mj} c_j \\ &= \sum_m |m\rangle \otimes \sum_{ij} \hat{H}|i\rangle W_{ij}^{(m)} c_j, \end{aligned}$$

where

$$W_{ij}^{(m)} := U_{mj} \delta_{ij} \quad (\dagger 4)$$

$$\hat{U}|0\rangle c_0 \otimes \hat{H}|0\rangle + \hat{U}|1\rangle c_1 \otimes \hat{H}|1\rangle = \sum_m |m\rangle \otimes \sum_j \hat{H}W^{(m)}|j\rangle c_j \quad (\dagger 5)$$

Therefore, upon the measurement

$$|\psi\rangle_2 = \hat{H}\hat{W}^{(m)}|\psi\rangle_1 \quad (\dagger 6)$$

depending on the measurement outcome  $m$ . Now recall from ( $\dagger 4$ ) that  $\hat{W}^{(m)}$  is diagonal, and hence

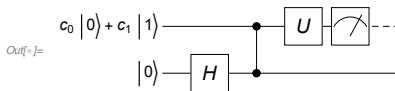
$$|\Psi_{\text{out}}\rangle = \hat{H}\hat{U}_z|\psi\rangle_{\text{in}} \quad (\dagger 7)$$

Consider a quantum circuit model of the following form.

```
In[ $\cdot$ ] := opU = Rotation[ $\phi$ , S[1, 1], "Label" -> "U"];  

  qc = QuissoCircuit[vec, LogicalForm[Ket[], S[2]], S[2, 6],  

    CZ[S[1], S[2]], opU, Measurement[S[1]], "PortSize" -> {2, 0.5}]
```



```
In[5]:= out = Elaborate[qc];
QuissoFactor[out] // LogicalForm
```

**Measurement** : Cannot perform a measurement on a nonnumeric state vector. Probability half is assumed.

$$\text{Out}[5]= \frac{c_0 \cos\left[\frac{\phi}{2}\right] |0_{S_2}\rangle + c_0 \cos\left[\frac{\phi}{2}\right] |1_{S_2}\rangle - i c_1 |0_{S_2}\rangle \sin\left[\frac{\phi}{2}\right] + i c_1 |1_{S_2}\rangle \sin\left[\frac{\phi}{2}\right]}{\sqrt{2}}$$

To make the structure of the resulting state, let us examine the state before the measurement.  
In addition, remove the expected Hadamard gate on the second qubit.

```
In[6]:= opU = Rotation[\phi, S[1, 1], "Label" \rightarrow "U"];
qc = QuissoCircuit[vec, LogicalForm[Ket[], S[2]],
S[2, 6], CZ[S[1], S[2]], opU, S[2, 6], "PortSize" \rightarrow {2, 0.5}]

Out[6]= 
```

```
In[7]:= out = Elaborate[qc];
QuissoFactor[out, S[1]] // LogicalForm
```

$$\text{Out}[7]= |0_{S_1}\rangle \otimes \left( c_0 \cos\left[\frac{\phi}{2}\right] |0_{S_2}\rangle - i c_1 |1_{S_2}\rangle \sin\left[\frac{\phi}{2}\right] \right) + \\ |1_{S_1}\rangle \otimes \left( c_1 \cos\left[\frac{\phi}{2}\right] |1_{S_2}\rangle - i c_0 |0_{S_2}\rangle \sin\left[\frac{\phi}{2}\right] \right)$$

### 3.5 Spin-Boson Model\*

In many architectures of quantum computers, the coupling between qubits are indirectly achieved through a bosonic mode shared by the qubits. Common examples include the quantum computers based on superconducting qubits and trapped ions. The bosonic mode can also be used for a quantum non-demolition (QND) measurement of quantum states as in the quantum computers based on superconducting circuits. Here we discuss the properties of such a spin-boson model at the elementary level.

Let us first examine the elementary properties of the model with a single-qubit coupled to a bosonic mode. The Hamiltonian is given by

$$\hat{H} = \omega \hat{a}^\dagger \hat{a} + \frac{1}{2} \Omega \hat{S}^z + g(\hat{a}^\dagger + \hat{a}) \hat{S}^x, \quad (3.37)$$

where  $\hat{a}$  and  $\hat{a}^\dagger$  are the annihilation and creation operators, respectively, of the bosonic mode.

---

Denote the qubits by S and the cavity mode by a.

```
Let[Qubit, S]
Let[Boson, a]
```

This is the Hamiltonian.

```
In[5]:= opH = Dagger[a] ** a + Ω S[3] / 2 + g (Dagger[a] + a) ** S[1]
Out[5]= a† a + g (a Sx + a† Sx) +  $\frac{\Omega S^z}{2}$ 
```

Here is a *truncated* basis.

```
In[6]:= bs = Basis[{a, S}];  
LogicalForm@bs
Out[6]= { |0a0s>, |0a1s>, |1a0s>, |1a1s>, |2a0s>,
          |2a1s>, |3a0s>, |3a1s>, |4a0s>, |4a1s>, |5a0s>, |5a1s>}
```

Here is the matrix representation of the Hamiltonian (showing only part of it), and the eigen-energies for a particular set of parameters. The matrix representation is not exact as the basis has been truncated.

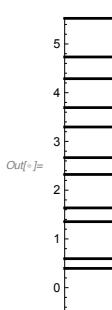
```
In[7]:= Ω = 1;
g = 1 / 10;
math = Matrix[opH];
math[[;; 6, ;; 6]] // MatrixForm
val = N@Eigenvalues[math]
Out[7]//MatrixForm=

$$\begin{pmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{10} & 0 & 0 \\ 0 & -\frac{1}{2} & \frac{1}{10} & 0 & 0 & 0 \\ 0 & \frac{1}{10} & \frac{3}{2} & 0 & 0 & \frac{1}{5\sqrt{2}} \\ \frac{1}{10} & 0 & 0 & \frac{1}{2} & \frac{1}{5\sqrt{2}} & 0 \\ 0 & 0 & 0 & \frac{1}{5\sqrt{2}} & \frac{5}{2} & 0 \\ 0 & 0 & \frac{1}{5\sqrt{2}} & 0 & 0 & \frac{3}{2} \end{pmatrix}
Out[7]= \{ 5.52494, 4.7328, 4.2874, 3.69409, 3.29609, 2.66746,
          2.32239, 1.63601, 1.35389, 0.594847, -0.505013, 0.395102 \}$$

```

Here you can see the (approximate) energy levels of the spin-boson model.

```
In[8]:= levels = Line@Thread@{Thread@{0, val}, Thread@{1, val}};
Graphics[levels,
  BaseStyle → Thick,
  Axes → {None, True}]
```



...

Next we demonstrate the QND measurement of quantum states using the bosonic mode.

...

We demonstrate a basic method to induce a coupling between two qubits by sharing the single bosonic mode.

$$\hat{H} = \omega \hat{a}^\dagger \hat{a} + \frac{1}{2} \Omega \sum_{j=1}^2 \hat{S}_j^z + g(\hat{a}^\dagger + \hat{a}) \sum_j \hat{S}_j^x \quad (3.38)$$

...