Chapter 6

Phase Estimation and its Applications

A *decision* problem is a problem in which it has only two possible outputs (yes or no) on any input of n bits. An output "yes" in the decision problem is to the number of solutions not to be zero and another output "no" in the decision problem is to the number of solutions to be zero. An example of a decision problem is deciding whether a given Boolean formula, $F(x_1, x_2) = x_1 \wedge x_2$, has solutions that satisfy $F(x_1, x_2)$ to have a *true* value or not, where the value of two Boolean variables x_1 and x_2 is either true (1) or false (0) and "$\wedge$" is the ***AND*** operation of two operands. For the convenience of the presentation, Boolean variable x_1^0 is to represent the value 0 (zero) of Boolean variable x_1 and Boolean variable x_1^1 is to represent the value 1 (one) of Boolean variable x_1. Boolean variable x_2^0 is to represent the value 0 (zero) of Boolean variable x_2 and Boolean variable x_2^1 is to represent the value 1 (one) of Boolean variable x_2.

A decision procedure is in the form of an algorithm to solve a decision problem. A decision procedure for the decision problem "a given Boolean formula, $F(x_1, x_2) = x_1 \wedge x_2$, does it have solutions that satisfy $F(x_1, x_2)$ to have a *true* value?" would implement $x_1 \wedge x_2$ (the ***AND*** operation of two operands) of four times according to four different inputs $x_1^0\ x_2^0$, $x_1^0\ x_2^1$, $x_1^1\ x_2^0$ and $x_1^1\ x_2^1$. After it executes each ***AND*** operation, it finds the fourth input $x_1^1\ x_2^1$ that satisfies $F(x_1, x_2)$ to have a *true* value. Finally, it gives an output "yes" to the decision problem. This implies that the number of solutions is not equal to zero. If time complexity of a decision procedure to solve a decision problem with the input of n bits is $O(2^n)$, then the decision problem is a NP-Complete problem.

We assume that a $(2^n \times 2^n)$ unitary matrix (operator) U has a $(2^n \times 1)$ eigenvector $|u>$ with eigenvalue $e^{\sqrt{-1}\times 2\times\pi\times\theta}$ such that $U \times |u> = e^{\sqrt{-1}\times 2\times\pi\times\theta} \times |u>$, where the value of θ is *unknown* and is real. The purpose of the phase estimate algorithm is to estimate the value of θ. Deciding whether there exist solutions for a problem with the input of n bits is equivalent to estimate the value of θ. In this chapter, we first describe how the phase estimate algorithm works on quantum computers and various kinds of real applications. We illustrate how to write quantum programs to compute and estimate the value of θ to that any given a $(2^n \times 2^n)$ unitary matrix (operator) U has a $(2^n \times 1)$ eigenvector $|u>$ with eigenvalue $(e^{\sqrt{-1}\times 2\times\pi\times\theta})$. Next, we explain the reason of why deciding whether there exist solutions for a problem with the input of n bits is equivalent to estimate the value of θ. We also explain how the quantum-counting algorithm determines the number of solutions for a decision problem with the input of n bits. Next, we introduce how to write quantum algorithms to implement the quantum-counting algorithm that is a real application of the phase estimate algorithm for computing the number of solutions for various kinds of real applications with the input of n bits.

6.1 Phase Estimation

We use the quantum circuit shown in Figure 6.1 to implement the phase estimation algorithm. It uses two quantum registers. At the left top in Figure 6.1, the first register ($\otimes_{k=t}^{1}|y_k^0\rangle$) contains t quantum bits initially in the state |0>. Quantum bit $|y_t^0>$ is the most significant bit. Quantum bit $|y_1^0>$ is the least significant bit. The corresponding decimal value of the first register is $(|y_t^0> \times 2^{t-1}) + \ldots + (|y_2^0> \times 2^{2-1}) + (|y_1^0> \times 2^{1-1})$. How we select t that is dependent on two things. The first thing is to that the number of bits of accuracy we wish to have in our estimation for the value of θ. The second thing is to that with what probability we wish the phase estimation algorithm to be successful. The dependence of t on these quantities appear naturally from the following analysis.

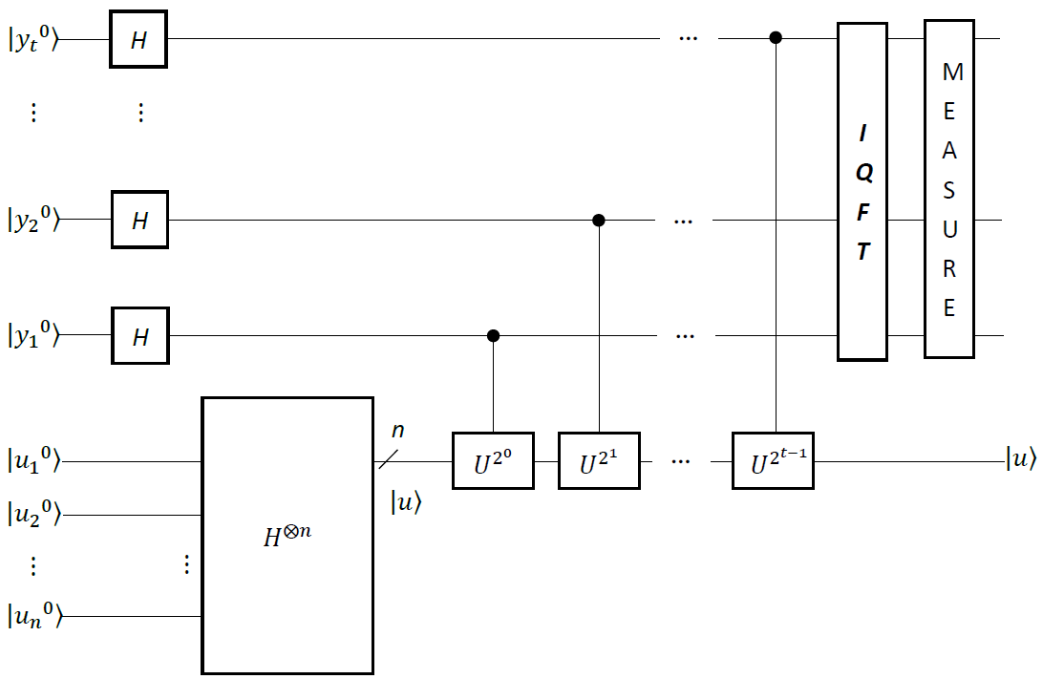

Figure 6.1: Quantum circuit of calculating the phase.

At the left bottom in Figure 6.1, the second register ($\otimes_{j=1}^{n}|u_j^0\rangle$) contains n quantum bits initially in the state |0>. Quantum bit $|u_1^0>$ is the most significant bit. Quantum bit $|u_n^0>$ is the least significant bit. The corresponding decimal value of the second register is $(|u_1^0> \times 2^{n-1}) + (|u_2^0> \times 2^{n-2}) + \ldots + (|y_n^0> \times 2^{n-n})$. How we select n that is dependent on a thing. The thing is to the size of the input for various kinds of real applications. This means that we select *n* that actually is to the number of bits of input for a problem. For the convenience of the presentation, the following initial state vector is

$$|\varphi_0\rangle = (\otimes_{k=t}^{1}|y_k^0\rangle) \otimes (\otimes_{j=1}^{n}|u_j^0\rangle). \quad (6.1)$$

6.1.1 Initialization of Phase Estimation

In Figure 6.1, the circuit begins by means of using a Hadamard transform on the *first* register $(\otimes_{k=t}^{1}|y_k^0\rangle)$ and another Hadamard transform on the *second* register $(\otimes_{j=1}^{n}|u_j^0\rangle)$. A superposition of the first register is $(\frac{1}{\sqrt{2^t}}\ (\otimes_{k=t}^{1}\ (|y_k^0\rangle + |y_k^1\rangle)))$. Another superposition of the second register is $(|u\rangle\ =\ \frac{1}{\sqrt{2^n}}\ (\otimes_{j=1}^{n}\ (|u_j^0\rangle + |u_j^1\rangle)))$. This is to say that the superposition of the second register begins in the new state vector $(|u\rangle\ =\ \frac{1}{\sqrt{2^n}}\ (\otimes_{j=1}^{n}\ (|u_j^0\rangle + |u_j^1\rangle)))$, and consists of *n* quantum bits as is necessary to store $(|u\rangle)$. The new state vector $(|u\rangle)$ is an eigenstate (eigenvector) of *U*. Therefore, this gives that the following new state vector is

$$|\varphi_1\rangle = (\frac{1}{\sqrt{2^t}}\ (\otimes_{k=t}^{1}\ (|y_k^0\rangle + |y_k^1\rangle))) \otimes (\frac{1}{\sqrt{2^n}}\ (\otimes_{j=1}^{n}\ (|u_j^0\rangle + |u_j^1\rangle)))$$

$$= (\frac{1}{\sqrt{2^t}}\ (\otimes_{k=t}^{1}\ (|y_k^0\rangle + |y_k^1\rangle))) \otimes (|u\rangle). \quad (6.2)$$

6.1.2 Controlled-*U* Operations on the Superposition of the Second Register to Phase Estimation

Next, in Figure 6.1, the circuit implements application of controlled-*U* operations on the superposition of the second register that is the state $(|u\rangle)$, with *U* raised to successive powers of two. Because the effect of one application of unitary operator *U* on its eigenvector (eigenstate) $(|u\rangle)$ is $(U \times |u\rangle\ =\ e^{\sqrt{-1}\times 2\times\pi\times\theta} \times |u\rangle)$, the effect of repeated application of unitary operator *U* on its eigenvector (eigenstate) $(|u\rangle)$ is

$$U^a\ |u\rangle\ =\ U^{a-1}\ U\ |u\rangle\ =\ U^{a-1}(e^{\sqrt{-1}\times 2\times\pi\times\theta} \times |u\rangle)\ =\ e^{\sqrt{-1}\times 2\times\pi\times\theta} \times (U^{a-1}|u\rangle)\ = e^{\sqrt{-1}\times 2\times\pi\times\theta} \times e^{\sqrt{-1}\times 2\times\pi\times\theta} \times \ldots \times e^{\sqrt{-1}\times 2\times\pi\times\theta}\ |u\rangle = e^{\sqrt{-1}\times 2\times\pi\times\theta\times a}\ |u\rangle. \quad (6.3)$$

Implementing one controlled-*U* operation that has its eigenvector (eigenstate) $(|u\rangle)$ and its eigenvalue $e^{\sqrt{-1}\times 2\times\pi\times\theta}$ is to that if the controlled quantum bit is the state |1>, then it completes one application of unitary operator *U*, ($U \times |u> = e^{\sqrt{-1}\times 2\times\pi\times\theta} \times |u>$). Otherwise, it does not complete one application of unitary operator *U*.

Similarly, implementing repeated application of one controlled-*U* operation that has its eigenvector (eigenstate) ($|u\rangle$) and its eigenvalue $e^{\sqrt{-1}\times 2\times\pi\times\theta}$ is to that if the controlled quantum bit is the state |1>, then it completes repeated application of unitary operator *U*, ($U^a \times |u> = e^{\sqrt{-1}\times 2\times\pi\times\theta\times a} \times |u>$). Otherwise, it does not complete repeated application of unitary operator *U*.

In the new state vector $|\varphi_1>$ in (6.2), each quantum bit in the first register is currently in its superposition. A superposition ($\frac{1}{\sqrt{2}}$ ($|y_1{}^0> + |y_1{}^1>$)) at the weighted position 2^0 is the controlled quantum bit of implementing controlled-U^{2^0} operations on the superposition of the second register that is the state ($|u\rangle$). This gives that the following new state vector is

$$|\varphi_2> = (\frac{1}{\sqrt{2^t}} (\otimes_{k=t}^{2} (|y_k^0\rangle + |y_k^1\rangle))) \otimes (|y_1^0\rangle|u\rangle + e^{\sqrt{-1}\times 2\times\pi\times\theta\times 2^0}|y_1^1\rangle|u\rangle)$$

$$= (\frac{1}{\sqrt{2^t}} (\otimes_{k=t}^{2} (|y_k^0\rangle + |y_k^1\rangle))) \otimes (|y_1^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\theta\times 2^0}|y_1^1\rangle) \otimes (|u\rangle). \quad (6.4)$$

Altering the phase of the state $|y_1{}^1>$ is from one (1) to become ($e^{\sqrt{-1}\times 2\times\pi\times\theta\times 2^0}$). We call it as *phase kickback*.

Next, in the new state vector $|\varphi_2>$ in (6.4), a superposition ($\frac{1}{\sqrt{2}}$ ($|y_2{}^0> + |y_2{}^1>$)) at the weighted position 2^1 is the controlled quantum bit of implementing controlled-U^{2^1} operations on the superposition of the second register that is the state ($|u\rangle$). This means that the following new state vector is

$$|\varphi_3> = (\frac{1}{\sqrt{2^t}} (\otimes_{k=t}^{3} (|y_k^0\rangle + |y_k^1\rangle))) \otimes (|y_2^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\theta\times 2^1}|y_2^1\rangle)$$

$$\otimes (|y_1^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\theta\times 2^0}|y_1^1\rangle) \otimes (|u\rangle). \quad (6.5)$$

Because of *phase kickback*, the phase of the state $|y_2{}^1>$ is from one (1) to become ($e^{\sqrt{-1}\times 2\times\pi\times\theta\times 2^1}$).

Next, in the new state vector $|\varphi_3>$ in (6.5), a superposition ($\frac{1}{\sqrt{2}}$ ($|y_3{}^0> + |y_3{}^1>$)) at the weighted position 2^2 through a superposition ($\frac{1}{\sqrt{2}}$ ($|y_t{}^0> + |y_t{}^1>$)) at the weighted position 2^{t-1} are the controlled quantum bits of implementing controlled-U^{2^2} operations through controlled-$U^{2^{t-1}}$ operations on the superposition of the second register that is the state ($|u\rangle$). This gives that the following new state vector is

$$|\varphi_4> = (\frac{1}{\sqrt{2^t}} (|y_t^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\theta\times 2^{t-1}}|y_t^1\rangle) \otimes (|y_{t-1}^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\theta\times 2^{t-2}}|y_{t-1}^1\rangle) \otimes \ldots$$
$$\otimes (|y_2^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\theta\times 2^1}|y_2^1\rangle) \otimes (|y_1^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\theta\times 2^0}|y_1^1\rangle)) \otimes (|u\rangle)$$
$$= (\frac{1}{\sqrt{2^t}} (\sum_{Y=0}^{2^t-1} e^{\sqrt{-1}\times 2\times\pi\times\theta\times Y} |Y\rangle)) \otimes (|u\rangle). \quad (6.6)$$

Because of *phase kickback*, the phase of the state $|Y>$ for $0 \le Y \le 2^t - 1$ is from one (1) to become ($e^{\sqrt{-1}\times 2\times\pi\times\theta\times Y}$). From this description above, the second quantum register stays in the state ($|u>$) through the computation.

6.1.3 Inverse Quantum Fourier Transform on the Superposition of the First Register to Phase Estimation

Next, in Figure 6.1, the circuit implements the **inverse Quantum Fourier transform** on the superposition of the first register. It takes the new state vector ($|\varphi_4>$) in (6.6) as its input state vector. The output state of the **inverse Quantum Fourier transform** on the superposition of the first register is

$$|\varphi_5> = (\sum_{Y=0}^{2^t-1} \frac{1}{\sqrt{2^t}} e^{\sqrt{-1}\times 2\times\pi\times\theta\times Y} \frac{1}{\sqrt{2^t}} \sum_{i=0}^{2^t-1} e^{-\sqrt{-1}\times 2\times\pi\times\frac{i}{2^t}\times Y} |i\rangle) \otimes (|u\rangle)$$
$$= (\frac{1}{2^t} (\sum_{Y=0}^{2^t-1} \sum_{i=0}^{2^t-1} e^{\sqrt{-1}\times 2\times\pi\times Y\times(\theta-\frac{i}{2^t})} |i\rangle)) \otimes (|u\rangle)$$
$$= (\sum_{i=0}^{2^t-1} \sum_{Y=0}^{2^t-1} \frac{1}{2^t} (e^{\sqrt{-1}\times 2\times\pi\times(\theta-\frac{i}{2^t})})^Y |i\rangle) \otimes (|u\rangle). \quad (6.7)$$

From this description above, the second quantum register still stays in the state ($|u>$) through the computation. From the new state vector ($|\varphi_5>$) in (6.7), the probability amplitude of $|i>$ is

$$\phi_i = \frac{1}{2^t} \times (\sum_{Y=0}^{2^t-1}(e^{\sqrt{-1}\times 2\times\pi\times(\theta-\frac{i}{2^t})})^Y). \quad (6.8)$$

6.1.4 Idealistic Phase Estimation

The probability amplitude of $|i>$ is simply the sum of a geometrical sequence with quotient $q = (e^{\sqrt{-1}\times 2\times\pi\times(\theta-\frac{i}{2^t})})$. On one hand if the value of θ may be expressed in t bits in the first quantum register, as $\theta = 0.y_t\, y_{t-1} \ldots y_2\, y_1 = (y_t\, y_{t-1} \ldots y_2\, y_1 / 2^t)$. Then the value of θ actually is equal to $(i / 2^t)$ for $0 \le i \le 2^t - 1$ and is an integer multiple of $(1 / 2^t)$. This gives that the quotient q is $e^{\sqrt{-1}\times 2\times\pi\times(\frac{i}{2^t}-\frac{i}{2^t})} = e^{\sqrt{-1}\times 2\times\pi\times 0} = 1$, the probability amplitude of $|i>$ is $\frac{1}{2^t} \times (\sum_{Y=0}^{2^t-1} 1^Y) = \frac{1}{2^t} \times (\sum_{Y=0}^{2^t-1} 1) = \frac{1}{2^t} \times 2^t = 1$ and any other probability amplitudes disappear. This is the *ideal case* of phase estimation. Finally, in Figure 6.1, after a measurement on the output state of the inverse quantum Fourier transform to the superposition of the first register is completed, we obtain the computational basis state $|i>$ with the successful probability 1 (100%). This indicates that the value of θ is equal to $(i / 2^t)$ with the successful probability 1 (100%). Therefore, we obtain the eigenvalue $(e^{\sqrt{-1}\times 2\times\pi\times\frac{i}{2^t}})$ with the successful probability 1 (100%).

6.1.5 Phase Estimation in Practical Cases

On the other hand if the value of θ may not be expressed in t bits in the first quantum register. This is to say that $\theta \ne 0.y_t\, y_{t-1} \ldots y_2\, y_1 \ne (y_t\, y_{t-1} \ldots y_2\, y_1 / 2^t)$. Then the quotient q is $e^{\sqrt{-1}\times 2\times\pi\times(\theta - \frac{i}{2^t})} \ne 1$ and we can rewrite the probability amplitude of $|i>$ in (6.8) as follows

$$\phi_i = \frac{1}{2^t} \times \frac{1-q^{2^t}}{1-q} = \frac{1}{2^t} \times \frac{1-(e^{\sqrt{-1}\times 2\times\pi\times(\theta - \frac{i}{2^t})})^{2^t}}{1-e^{\sqrt{-1}\times 2\times\pi\times(\theta - \frac{i}{2^t})}} = \frac{1}{2^t} \times \frac{1-e^{\sqrt{-1}\times 2\times\pi\times(2^t\times\theta - i)}}{1-e^{\sqrt{-1}\times 2\times\pi\times(\theta - \frac{i}{2^t})}}. \quad (6.9)$$

This gives another good explanation of uncertainty and thus appearing inaccuracy when measuring the output of the **inverse quantum Fourier transform** in Figure 6.1. The probability of measuring a suitable state $|i>$ on the first register in Figure 6.1 is

$$|\phi_i|^2 = \frac{1}{2^{2\times t}} \times \frac{|1-e^{\sqrt{-1}\times 2\times\pi\times(2^t\times\theta - i)}|^2}{|1-e^{\sqrt{-1}\times 2\times\pi\times\left(\theta - \frac{i}{2^t}\right)}|^2}. \quad (6.10)$$

Because $|1 - e^{\sqrt{-1}\times\gamma}|^2 = 4 \times sin^2(\gamma / 2)$, we can rewrite $|\phi_i|^2$ in (6.10) as follows

$$|\phi_i|^2 = \frac{1}{2^{2\times t}} \times \frac{4\times\sin^2(\frac{2\times\pi\times(2^t\times\theta - i)}{2})}{4\times\sin^2(\frac{2\times\pi\times\left(\theta - \frac{i}{2^t}\right)}{2})} = \frac{1}{2^{2\times t}} \times \frac{\sin^2(\frac{2\times\pi\times(2^t\times\theta - i)}{2})}{\sin^2(\frac{2\times\pi\times\left(\theta - \frac{i}{2^t}\right)}{2})}. \quad (6.11)$$

This is the *practical case* of phase estimation. Finally, in Figure 6.1, after a measurement on the output state of the inverse quantum Fourier transform to the superposition of the first register is completed, we obtain the computational basis state $|i>$ with the probability $(\frac{1}{2^{2\times t}} \times \frac{\sin^2(\frac{2\times\pi\times(2^t\times\theta - i)}{2})}{\sin^2(\frac{2\times\pi\times\left(\theta - \frac{i}{2^t}\right)}{2})})$. Because $(i / 2^t) = (y_t\, y_{t-1} \ldots y_2\, y_1 / 2^t) = 0.y_t\, y_{t-1} \ldots y_2\, y_1$, $(i / 2^t)$ is an estimated value to the value of θ with the probability $(\frac{1}{2^{2\times t}} \times \frac{\sin^2(\frac{2\times\pi\times(2^t\times\theta - i)}{2})}{\sin^2(\frac{2\times\pi\times\left(\theta - \frac{i}{2^t}\right)}{2})})$. Hence, we only obtain an *estimated* eigenvalue $(e^{\sqrt{-1}\times 2\times\pi\times\frac{i}{2^t}})$ with the probability $(\frac{1}{2^{2\times t}} \times \frac{\sin^2(\frac{2\times\pi\times(2^t\times\theta - i)}{2})}{\sin^2(\frac{2\times\pi\times\left(\theta - \frac{i}{2^t}\right)}{2})})$.

This is to say that if more than one $|\phi_i|^2$ differs from zero then there is a nonzero probability of receiving different estimated phases (eigenvalues) after the measurement when repeating to execute the circuit of phase estimation in Figure 6.1.

6.1.6 Performance and Requirement to Phase Estimation

The phase estimation algorithm allows one to estimate the value of the phase θ to an eigenvalue $(e^{\sqrt{-1}\times 2\times\pi\times\theta})$ of a unitary operator U with its eigenvector ($|u>$). From the analysis in subsection 6.1.4, if the value of the phase θ is to $\theta = 0.y_t\, y_{t-1} \ldots y_2\, y_1 = (y_t\, y_{t-1} \ldots y_2\, y_1 / 2^t)$ that is to a t bit binary expansion of the first quantum register, then in the circuit of Figure 6.1 the outcome of the final measurement is $|i>$ with the probability 100%. Because $|i>$ is a t bit binary expansion of the first quantum register, we obtain that the value of the phase θ is equal to $(i / 2^t)$ with the probability 100%. This is the *ideal* case.

On the other hand, from the analysis in subsection 6.1.5, if the value of the phase θ is not a t bit binary expansion of the first quantum register, then the outcome of the final measurement is $|i>$ with the probability $(\frac{1}{2^{2\times t}} \times \frac{\sin^2(\frac{2\times\pi\times(2^t\times\theta - i)}{2})}{\sin^2(\frac{2\times\pi\times\left(\theta - \frac{i}{2^t}\right)}{2})})$. Let Y be the integer in the range 0 to $2^t - 1$ so that $(Y / 2^t) = (y_t\, y_{t-1} \ldots y_2\, y_1 / 2^t) = (0.y_t\, y_{t-1} \ldots y_2\, y_1)$ is the best t bit approximation to the value of the phase θ and $(Y / 2^t)$ is less than the value of the phase θ. This indicates that the difference $\delta = \theta - (Y / 2^t)$ between θ and $(Y / 2^t)$ satisfies $0 \leq \delta \leq (1 / 2^t)$. We assume that the outcome of the final measurement in the circuit of Figure 6.1 is $|i>$. We aim to bound the probability of obtaining a value of i such that $|i - Y| > \varepsilon$, where ε is a positive integer characterizing our desired tolerance to error. The probability of measuring such a state $|i>$ is

$$\boldsymbol{P}(|i - Y| > \varepsilon) \leq \frac{1}{2\times(\varepsilon-1)}. \tag{6.12}$$

We assume that we would like to approximate the value of the phase θ to an accuracy 2^{-t}, that is, we select $\varepsilon = 2^{t-n} - 1$. By means of using $t = n + q$ quantum bits in the circuit of Figure 6.1, we see from (6.12) that the probability of obtaining an approximation correct to this accuracy is at least

$$\boldsymbol{P}(|i - Y| \leq \varepsilon) = 1 - \boldsymbol{P}(|i - Y| > \varepsilon) = 1 - \frac{1}{2\times(\varepsilon-1)}$$

$$= 1 - \frac{1}{2\times(2^{t-n}-1-1)} = 1 - \frac{1}{2\times(2^{t-n}-2)}. \tag{6.13}$$

Therefore to successfully obtain the value of the phase θ accurate to t bits with probability of success at least $1 - \alpha = 1 - \frac{1}{2\times(2^{t-n}-2)}$, we select

$$t = n + \lceil \log_2(2 + (1 / (2 \times \alpha))) \rceil. \tag{6.14}$$

Because $\alpha = \frac{1}{2\times(2^{t-n}-2)}$, we obtain $\alpha \times (2 \times (2^{t-n} - 2)) = 1$. This is to say that $2^{t-n} - 2 = (1 / (2 \times \alpha))$ and $2^{t-n} = (1 / (2 \times \alpha)) + 2$ and $\log_2(2^{t-n}) = \log_2(2 + (1 / (2 \times \alpha)))$ and $t = n + \lceil \log_2(2 + (1 / (2 \times \alpha))) \rceil$. This is the result in (6.14).

6.1.7 Assessment to Complexity of Phase Estimation

In the circuit of Figure 6.1, the number of quantum bits to the *first* register ($\otimes_{k=t}^{1}|y_k^0\rangle$) is t quantum bits and the number of quantum bits to the *second* register ($\otimes_{j=1}^{n}|u_j^0\rangle$) is n quantum bits. Therefore, space complexity of phase estimation is $O(t + n)$ quantum bits. The *first* stage in the circuit of Figure 6.1 is to implement $(t + n)$ Hadamard gates.

Next, the *second* stage in the circuit of Figure 6.1 is to implement application of controlled-U operations on the superposition of the second register that is the state ($|u\rangle$), with U raised to successive powers of two. The $U1(\lambda)$ gate is $U1(\lambda) = U1(\text{lambda}) = \begin{pmatrix} 1 & 0 \\ 0 & e^{\sqrt{-1}\times\lambda} \end{pmatrix}$ for that λ (lambda) is a real value. If the value of λ is equal to $(2 \times \pi \times \theta \times 2^{k-1})$ to $1 \leq k \leq t$, then it can implement a controlled-$U^{2^{k-1}}$ operation to $1 \leq k \leq t$. This is to say that a total cost of completing the second stage is to implement t $U1(\lambda)$ gates.

Next, the *third* stage in the circuit of Figure 6.1 is to implement the inverse quantum Fourier transform on the superposition of the first register. A total cost of completing the inverse quantum Fourier transform is to implement $O(t^2)$ quantum gates. Finally, reading out the output state of the inverse quantum Fourier transform on the superposition of the first register is to implement one measurement. Because from the statements above a total cost of completing phase estimation is $O(t^2 + n)$ quantum gates, time complexity of phase estimation is to $O(t^2 + n)$ quantum gates.

6.2 Computing Eigenvalue of a $(2^2 \times 2^2)$ Unitary Matrix U with a $(2^2 \times 1)$ Eigenvector $|u>$ in Phase Estimation

We use the circuit in Figure 6.2 to compute eigenvalue of a $(2^2 \times 2^2)$ unitary matrix

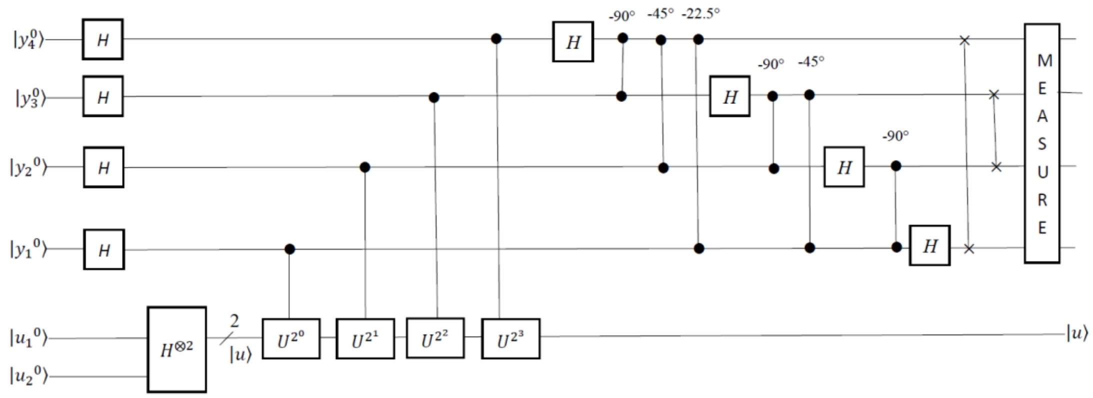

Figure 6.2: Quantum circuit for calculating eigenvalue of a ($2^2 \times 2^2$) unitary matrix U with a ($2^2 \times 1$) eigenvector $|u>$.

U with a ($2^2 \times 1$) eigenvector $|u>$. It makes use of two quantum registers. At the left top in Figure 6.2, the first register ($\otimes_{k=4}^{1}|y_k^0\rangle$) contains *four* quantum bits initially in the state $|0>$. Quantum bit $|y_4^0>$ is the most significant bit. Quantum bit $|y_1^0>$ is the least significant bit. The corresponding decimal value of the first register is ($|y_4^0> \times 2^{4-1}$) + ($|y_3^0> \times 2^{3-1}$) + ($|y_2^0> \times 2^{2-1}$) + ($|y_1^0> \times 2^{1-1}$). At the left bottom in Figure 6.2, the second register ($\otimes_{j=1}^{2}|u_j^0\rangle$) contains *two* quantum bits initially in the state $|0>$. Quantum bit $|u_1^0>$ is the most significant bit. Quantum bit $|u_2^0>$ is the least significant bit. The corresponding decimal value of the second register is ($|u_1^0> \times 2^{2-1}$) + ($|u_2^0> \times 2^{2-2}$). For the convenience of the presentation, the following initial state vector is

$$|\varphi_0> = (\otimes_{k=4}^{1}|y_k^0\rangle) \otimes (\otimes_{j=1}^{2}|u_j^0\rangle). \qquad (6.15)$$

6.2.1 Initialize Quantum Registers to Calculate Eigenvalue of a ($2^2 \times 2^2$) Unitary Matrix *U* with a ($2^2 \times 1$) Eigenvector $|u>$ in Phase Estimation

In Listing 6.1, the program is in the backend that is *simulator* of Open QASM with *thirty-two* quantum bits in **IBM**'s quantum computer. The program is to compute eigenvalue of a ($2^2 \times 2^2$) unitary matrix U with a ($2^2 \times 1$) eigenvector $|u>$ in phase estimation. Figure 6.3 is the corresponding quantum circuit of the program in Listing 6.1 and is to implement the quantum circuit of Figure 6.2 to compute eigenvalue of a ($2^2 \times 2^2$) unitary matrix U with a ($2^2 \times 1$) eigenvector $|u>$ in phase estimation.

```
1. OPENQASM 2.0;
2. include "qelib1.inc";

3. qreg q[6];
4. creg c[4];
```

Listing 6.1: The program of computing eigenvalue of a ($2^2 \times 2^2$) unitary matrix U with a ($2^2 \times 1$) eigenvector $|u>$ in phase estimation.

The statement "OPENQASM 2.0;" on line one of Listing 6.1 is to point out that the program is written with version 2.0 of Open QASM. Next, the statement "include "qelib1.inc";" on line two of Listing 6.1 is to continue parsing the file "qelib1.inc" as if the contents of the file were pasted at the location of the include statement, where the file “qelib1.inc” is **Quantum Experience (QE) Standard Header** and the path is specified relative to the current working directory.

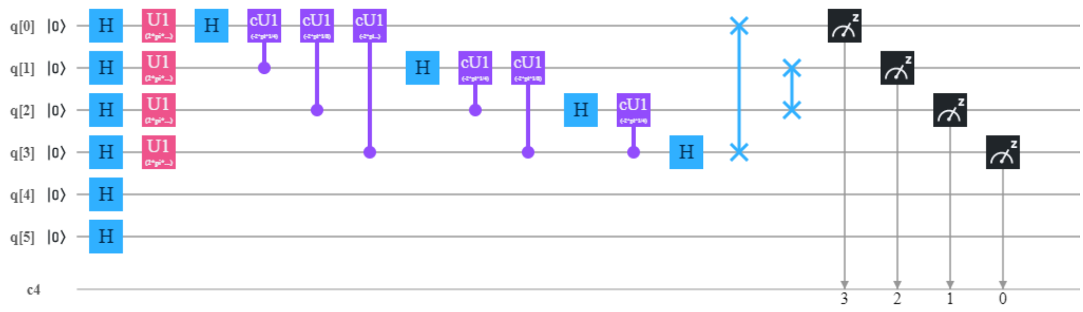

Figure 6.3: Implementing quantum circuits of Figure 6.2 to compute eigenvalue of a $(2^2 \times 2^2)$ unitary matrix U with a $(2^2 \times 1)$ eigenvector $|u>$ in phase estimation.

Then, the statement “qreg q[6];” on line three of Listing 6.1 is to declare that in the program there are *six* quantum bits. In the left top of Figure 6.3, six quantum bits are subsequently q[0], q[1], q[2], q[3], q[4] and q[5]. The initial value of each quantum bit is set to state |0>. We make use of four quantum bits q[0], q[1], q[2] and q[3] to respectively encode four quantum bits $|y_4>$, $|y_3>$, $|y_2>$ and $|y_1>$ in Figure 6.2. We use two quantum bits q[4] and q[5] to respectively encode two quantum bits $|u_1>$ and $|u_2>$ in Figure 6.2. For the convenience of our explanation, $q[k]^0$ for $0 \leq k \leq 5$ is to represent the value 0 of q[k] and $q[k]^1$ for $0 \leq k \leq 5$ is to represent the value 1 of q[k]. Because quantum bit $|y_4{}^0>$ is the most significant bit and quantum bit $|y_1{}^0>$ is the least significant bit, quantum bit $|q[0]^0>$ is the most significant bit and quantum bit $|q[3]^0>$ is the least significant bit. The corresponding decimal value of the first register in Figure 6.3 is $(|q[0]^0> \times 2^{4-1}) + (|q[1]^0> \times 2^{3-1}) + (|q[2]^0> \times 2^{2-1}) + (|q[3]^0> \times 2^{1-1})$.

Next, the statement “creg c[4];” on line four of Listing 6.1 is to declare that there are four classical bits in the program. In the left bottom of Figure 6.3, four classical bits are subsequently c[0], c[1], c[2] and c[3]. The initial value of each classical bit is set to zero (0). For the convenience of our explanation, $c[k]^0$ for $0 \leq k \leq 3$ is to represent the value 0 of c[k] and $c[k]^1$ for $0 \leq k \leq 3$ is to represent the value 1 of c[k]. The corresponding decimal value of the four initial classical bits $c[3]^0\ c[2]^0\ c[1]^0\ c[0]^0$ is $2^3 \times c[3]^0 + 2^2 \times c[2]^0 + 2^1 \times c[1]^0 + 2^0 \times c[0]^0$. This indicates that classical bit $c[3]^0$ is the most significant bit and classical bit $c[0]^0$ is the least significant bit. For the convenience of our explanation, we can rewrite the initial state vector $|\varphi_0> = (\otimes_{k=4}^{1} |y_k^0\rangle) \otimes (\otimes_{j=1}^{2} |u_j^0\rangle)$ in (6.15) in Figure 6.2 as follows

$$|\varphi_0\rangle = |q[0]^0\rangle\ |q[1]^0\rangle\ |q[2]^0\rangle\ |q[3]^0\rangle\ |q[4]^0\rangle\ |q[5]^0\rangle. \qquad (6.16)$$

6.2.2 Superposition of Quantum Registers to Calculate Eigenvalue of a ($2^2 \times 2^2$) Unitary Matrix *U* with a ($2^2 \times 1$) Eigenvector |*u*> in Phase Estimation

In Figure 6.2, the first stage of the circuit is to implement a Hadamard transform with four Hadamard gates on the *first* register ($\otimes_{k=4}^{1}|y_k^0\rangle$) and another Hadamard transform with two Hadamrad gates on the *second* register ($\otimes_{j=1}^{2}|u_j^0\rangle$). The *six* statements "h q[0];", "h q[1];", "h q[2];", "h q[3];", "h q[4];" and "h q[5];" on line *five* of Listing 6.1 through line *ten* of Listing 6.1 is to implement *six* Hadamrad gates on the first register and the second register. They complete each Hadamrad gate in the first time slot of Figure 6.3 and perform the first stage of the circuit in Figure 6.2.

Listing 6.1 continued…

```
//Implement a Hadamard transform on two registers.

5.  h q[0];
6.  h q[1];
7.  h q[2];
8.  h q[3];
9.  h q[4];
10. h q[5];
```

A superposition of the first register is $(\frac{1}{\sqrt{2^4}}\ (\otimes_{k=4}^{1}(|y_k^0\rangle + |y_k^1\rangle))) = (\frac{1}{\sqrt{2^4}}\ (\otimes_{a=0}^{3}(|q[a]^0\rangle + |q[a]^1\rangle)))$. Another superposition of the second register is $(|u\rangle = \frac{1}{\sqrt{2^2}}\ (\otimes_{j=1}^{2}(|u_j^0\rangle + |u_j^1\rangle)) = \frac{1}{\sqrt{2^2}}\ (\otimes_{b=4}^{5}(|q[b]^0\rangle + |q[b]^1\rangle)))$. This is to say that the superposition of the second register begins in the new state vector $(|u\rangle = \frac{1}{\sqrt{2^2}}\ (\otimes_{j=1}^{2}(|u_j^0\rangle + |u_j^1\rangle = \frac{1}{\sqrt{2^2}}\ (\otimes_{b=4}^{5}(|q[b]^0\rangle + |q[b]^1\rangle)))$ and contains *two* quantum bits as is necessary to store ($|u\rangle$). The new state vector ($|u\rangle$) is an eigenstate (eigenvector) of *U*. Therefore, this gives that the following new state vector is

$$|\varphi_1> = (\frac{1}{\sqrt{2^4}}\ (\otimes_{k=4}^{1}\ (|y_k^0\rangle + |y_k^1\rangle))) \otimes (\frac{1}{\sqrt{2^2}}\ (\otimes_{j=1}^{2}\ (|u_j^0\rangle + |u_j^1\rangle)))$$

$$= (\frac{1}{\sqrt{2^4}}\ (\otimes_{k=4}^{1}\ (|y_k^0\rangle + |y_k^1\rangle))) \otimes (|u\rangle)$$

$$= (\frac{1}{\sqrt{2^4}}\ (\otimes_{a=0}^{3}\ (|q[a]^0\rangle + |q[a]^1\rangle)))\ \otimes (\frac{1}{\sqrt{2^2}}\ (\otimes_{b=4}^{5}\ (|q[b]^0\rangle + |q[b]^1\rangle)))$$

$$= (\frac{1}{\sqrt{2^4}}\ (\otimes_{a=0}^{3}\ (|q[a]^0\rangle + |q[a]^1\rangle)))\ \otimes (|u\rangle). \quad (6.17)$$

6.2.3 Controlled-*U* Operations on the Superposition of the Second Register to Determine Eigenvalue of a ($2^2 \times 2^2$) Unitary Matrix *U* with a ($2^2 \times 1$) Eigenvector |*u*> in Phase Estimation

In the new state vector $|\varphi_1>$ in (6.17), each quantum bit in the first register is currently in its superposition. The value of the first register is from state $(\otimes_{k=4}^{1}|y_k^0\rangle)$ (zero) encoded by state $(\otimes_{a=0}^{3}|q[a]^0\rangle)$ through state $(\otimes_{k=4}^{1}|y_k^1\rangle)$ (fifteen) encoded by state $(\otimes_{a=0}^{3}|q[a]^1\rangle)$. The circuit of Figure 6.2 can precisely estimate sixteen phases. This is to say that the first register with four quantum bits can precisely represent sixteen phases. Sixteen phases are subsequently ($0 / 2^4$), ($1 / 2^4$), ($2 / 2^4$), ($3 / 2^4$), ($4 / 2^4$), ($5 / 2^4$), ($6 / 2^4$), ($7 / 2^4$), ($8 / 2^4$), ($9 / 2^4$), ($10 / 2^4$), ($11 / 2^4$), ($12 / 2^4$), ($13 / 2^4$), ($14 / 2^4$) and ($15 / 2^4$). The corresponding sixteen phase angles are subsequently ($2 \times \pi \times 0 / 2^4$), ($2 \times \pi \times 1 / 2^4$), ($2 \times \pi \times 2 / 2^4$), ($2 \times \pi \times 3 / 2^4$), ($2 \times \pi \times 4 / 2^4$), ($2 \times \pi \times 5 / 2^4$), ($2 \times \pi \times 6 / 2^4$), ($2 \times \pi \times 7 / 2^4$), ($2 \times \pi \times 8 / 2^4$), ($2 \times \pi \times 9 / 2^4$), ($2 \times \pi \times 10 / 2^4$), ($2 \times \pi \times 11 / 2^4$), ($2 \times \pi \times 12 / 2^4$), ($2 \times \pi \times 13 / 2^4$), ($2 \times \pi \times 14 / 2^4$) and ($2 \times \pi \times 15 / 2^4$).

Say that we are trying to determine an eigenvalue of 90°. This is to say that the effect of one application of unitary operator *U* on its eigenvector (eigenstate) ($|u\rangle$) is ($U \times |u> = e^{\sqrt{-1}\times 2\times\pi\times\theta} \times |u> = e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{2^4}} \times |u>$). So, the effect of repeated application of unitary operator *U* on its eigenvector (eigenstate) ($|u\rangle$) is

$$U^a\ |u> = e^{\sqrt{-1}\times 2\times\pi\times\theta\times a}\ |u> = e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{2^4}\times a} \times |u>. \quad (6.18)$$

A superposition $(\frac{1}{\sqrt{2}}\ (|y_1{}^0> + |y_1{}^1>))$ that is encoded by $(\frac{1}{\sqrt{2}}\ (|q[3]^0> + |q[3]^1>))$ at the weighted position 2^0 is the controlled quantum bit of implementing controlled-U^{2^0} operations on the superposition of the second register that is the state ($|u\rangle$). Similarly, a superposition ($\frac{1}{\sqrt{2}}$ ($|y_2^0>$ + $|y_2^1>$)) that is encoded by ($\frac{1}{\sqrt{2}}$ ($|q[2]^0>$ + $|q[2]^1>$)) at the weighted position 2^1 is the controlled quantum bit of implementing controlled-U^{2^1} operations on the superposition of the second register that is the state ($|u\rangle$). Next, a superposition ($\frac{1}{\sqrt{2}}$ ($|y_3^0>$ + $|y_3^1>$)) that is encoded by ($\frac{1}{\sqrt{2}}$ ($|q[1]^0>$ + $|q[1]^1>$)) at the weighted position 2^2 is the controlled quantum bit of implementing controlled-U^{2^2} operations on the superposition of the second register that is the state ($|u\rangle$). Next, a superposition ($\frac{1}{\sqrt{2}}$ ($|y_4^0>$ + $|y_4^1>$)) that is encoded by ($\frac{1}{\sqrt{2}}$ ($|q[0]^0>$ + $|q[0]^1>$)) at the weighted position 2^3 is the controlled quantum bit of implementing controlled-U^{2^3} operations on the superposition of the second register that is the state ($|u\rangle$).

The *four* statements from line *eleven* through line *fourteen* in Listing 6.1 are "u1(2*pi*4/16*1) q[3];", "u1(2*pi*4/16*2) q[2];", "u1(2*pi*4/16*4) q[1];" and "u1(2*pi*4/16*8) q[0];". They take the new state vector ($|\varphi_1>$) in (6.17) as their input

Listing 6.1 continued…

```
//Implement controlled-U operations on the superposition of the second register.

11. u1(2*pi*4/16*1) q[3];
12. u1(2*pi*4/16*2) q[2];
13. u1(2*pi*4/16*4) q[1];
14. u1(2*pi*4/16*8) q[0];
```

state vector and implement each controlled-*U* operation on the superposition of the second register in the *second* time slot of Figure 6.3 and in the *second* stage of Figure 6.2. They alert the phase of the state $|y_1^1>$ ($|q[3]^1>$) is from one (1) to become ($e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 2^0}$) = ($e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 1}$). They alert the phase of the state $|y_2^1>$ ($|q[2]^1>$) is from one (1) to become ($e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 2^1}$) = ($e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 2}$). They alert the phase of the state $|y_3^1>$ ($|q[1]^1>$) is from one (1) to become ($e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 2^2}$) = ($e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 4}$) and alert the phase of the state $|y_4^1>$ ($|q[0]^1>$) is from one (1) to become ($e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 2^3}$)

= $(e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 8})$. This gives that the following new state vector is

$$|\varphi_2\rangle = (\frac{1}{\sqrt{2^4}}\ (|y_4^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 2^3}|y_4^1\rangle) \otimes (|y_3^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 2^2}|y_3^1\rangle) \otimes$$

$$(|y_2^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 2^1}|y_2^1\rangle) \otimes (|y_1^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 2^0}|y_1^1\rangle)) \otimes (|u\rangle)$$

$$= (\frac{1}{\sqrt{2^4}}\ (|y_4^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 8}|y_4^1\rangle) \otimes (|y_3^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 4}|y_3^1\rangle) \otimes$$

$$(|y_2^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 2}|y_2^1\rangle) \otimes (|y_1^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 1}|y_1^1\rangle)) \otimes (|u\rangle)$$

$$= (\frac{1}{\sqrt{2^4}}\ (|q[0]^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 8}|q[0]^1\rangle) \otimes (|q[1]^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 4}|q[1]^1\rangle) \otimes$$

$$(|q[2]^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 2}|q[2]^1\rangle) \otimes (|q[3]^0\rangle + e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times 1}|q[3]^1\rangle)) \otimes (|u\rangle)$$

$$= (\frac{1}{\sqrt{2^4}}\ (\sum_{Y=0}^{2^4-1} e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times Y}\ |Y\rangle)) \otimes (|u\rangle). \qquad (6.19)$$

From this description above, the second quantum register stays in the state ($|u>$) through the computation. Because of *phase kickback*, the phase of the state $|Y>$ for $0 \leq Y \leq 2^4 - 1$ is from one (1) to become $(e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times Y})$. In the state vector ($|\varphi_2>$) in (6.19), it contains sixteen phase angles from state |0> through state |15>. The front eight phase angles are (90° × 0 = 0°), (90° × 1 = 90°), (90° × 2 = 180°), (90° × 3 = 270°), (90° × 4 = 360° = 0°), (90° × 5 = 450° = 90°), (90° × 6 = 540° = 180°) and (90° × 7 = 630° = 270°). The last eight phase angles are (90° × 8 = 720° = 0°), (90° × 9 = 810° = 90°), (90° × 10 = 900° = 180°), (90° × 11 = 990° = 270°), (90° × 12 = 1080° = 0°), (90° × 13 = 1170° = 90°), (90° × 14 = 1260° = 180°) and (90° × 15 = 1350° = 270°). The phase angle rotates back to its starting value 0° *four* times.

6.2.4 The Inverse Quantum Fourier Transform on the Superposition of the First Register to Compute Eigenvalue of a ($2^2 \times 2^2$) Unitary Matrix *U* with a ($2^2 \times 1$) Eigenvector |*u*> in Phase Estimation

Hidden patterns and information stored in the state vector ($|\varphi_2>$) in (6.19) are to that its phase angle rotates back to its starting value 0° *four* times. This implies that the number of the period per sixteen phase angles is *four* and the frequency is equal to *four*

(16 / 4). The twelve statements from line *fifteen* through line *twenty-six* in Listing 6.1

Listing 6.1 continued…

```
//Implement one inverse quantum Fourier transform on the superposition of the first
// register.

15. h q[0];
16. cu1(-2*pi*1/4) q[1],q[0];
17. cu1(-2*pi*1/8) q[2],q[0];
18. cu1(-2*pi*1/16) q[3],q[0];

19. h q[1];
20. cu1(-2*pi*1/4) q[2],q[1];
21. cu1(-2*pi*1/8) q[3],q[1];

22. h q[2];
23. cu1(-2*pi*1/4) q[3],q[2];

24. h q[3];

25. swap q[0],q[3];
26. swap q[1],q[2];
```

implement each quantum operation from the *third* time slot through the *fourteenth* time slot in Figure 6.3. They actually implement each quantum operation of completing an **inverse quantum Fourier transform** on the superposition of the first register in Figure 6.2. They take the state vector ($|\varphi_2>$) in (6.19) as their input state vector. Because the **inverse quantum Fourier transform** effectively transforms the state of the first register into a superposition of the *periodic* signal's component frequencies, they produce the following state vector

$$|\varphi_3> = \left(\sum_{Y=0}^{2^4-1} \frac{1}{\sqrt{2^4}} e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{2^4}\times Y} \frac{1}{\sqrt{2^4}} \sum_{i=0}^{2^4-1} e^{-\sqrt{-1}\times 2\times\pi\times\frac{i}{2^4}\times Y} |i\rangle\right) \otimes (|u\rangle)$$

$$= \left(\frac{1}{2^4} \left(\sum_{Y=0}^{2^4-1} \sum_{i=0}^{2^4-1} e^{\sqrt{-1}\times 2\times\pi\times Y\times(\frac{4}{2^4}-\frac{i}{2^4})} |i\rangle\right)\right) \otimes (|u\rangle)$$

$$= \left(\sum_{i=0}^{2^4-1} \sum_{Y=0}^{2^4-1} \frac{1}{2^4} \left(e^{\sqrt{-1}\times 2\times\pi\times(\frac{4}{2^4}-\frac{i}{2^4})}\right)^Y |i\rangle)\right) \otimes (|u\rangle). \quad (6.20)$$

6.2.5 Read the Quantum Result to Figure out Eigenvalue of a ($2^2 \times 2^2$) Unitary Matrix *U* with a ($2^2 \times 1$) Eigenvector |*u*> in Phase Estimation

Finally, the four statements "measure q[0] -> c[3];", "measure q[1] -> c[2];", "measure q[2] -> c[1];" and "measure q[3] -> c[0];" from line *twenty-seven* through line *thirty* in Listing 6.1 implement a measurement. They measure the output state of the inverse quantum Fourier transform to the superposition of the first register in Figure 6.3 and in Figure 6.2. This is to say that they measure four quantum bits q[0], q[1], q[2] and q[3] of the first register and record the measurement outcome by overwriting four classical bits c[3], c[2], c[1] and c[0].

Listing 6.1 continued…

```
//Complete a measurement on the first register.

27. measure q[0] -> c[3];
28. measure q[1] -> c[2];
29. measure q[2] -> c[1];
30. measure q[3] -> c[0];
```

In the backend *simulator* with thirty-two quantum bits in **IBM**'s quantum computers, we use the command "run" to execute the program in Listing 6.1. Figure 6.4 shows the measured result. From Figure 6.4, we obtain that a computational basis state 0100 (c[3] = 0 = q[0] = |0>, c[2] = 1 = q[1] = |1>, c[1] = 0 = q[2] = |0> and c[0] = 0 = q[3] = |0>) has the probability 100%. This is to say that the value of θ is equal to (4 / 16). Therefore, we obtain that eigenvalue of a ($2^2 \times 2^2$) unitary matrix *U* with a ($2^2 \times 1$) eigenvector |*u*> is equal to ($e^{\sqrt{-1} \times 2 \times \pi \times \frac{4}{2^4}}$) with the probability 100%.

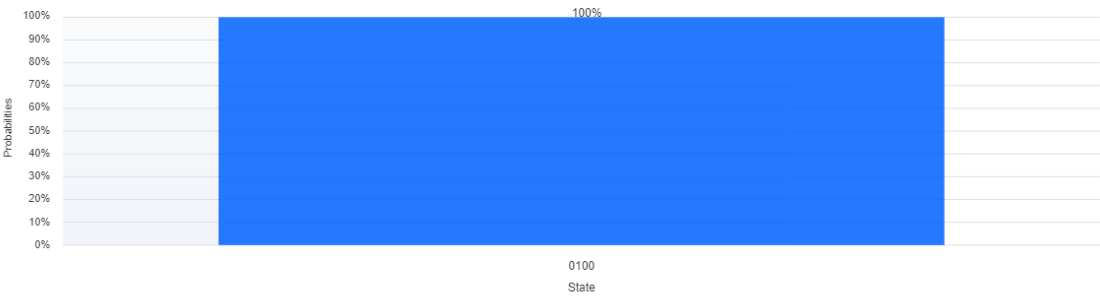

Figure 6.4: A computational basis state 0100 has the probability 100%.

6.3 Quantum Counting to a Decision Problem with Any Input of *n* Bits in Real Applications of Phase Estimation

A *decision* problem is a problem in which it has only two possible outputs (yes or no) on any input of n bits. An output "yes" in the decision problem is to the number of solutions not to be zero and another output "no" in the decision problem is to the number of solutions to be zero. Solving a decision problem with any input of n bits is equivalent to solve one interesting problem with any input of n bits that is to from an unsorted database including 2^n items with each item has n bits how many items satisfy any given condition and we would like to find the number of solutions. If the number of solutions is not equal to zero, then there is an output "yes" in the decision problem with any input of n bits. Otherwise, there is an output "no" in the decision problem with any input of n bits.

A common formulation of a decision problem with any input of n bits is as follows. For any given oracular function O_f: $\{u_1\ u_2\ \ldots\ u_{n-1}\ u_n \mid \forall\ u_j \in \{0, 1\}$ for $1 \leq j \leq n\} \rightarrow \{0, 1\}$, its domain is $\{u_1\ u_2\ \ldots\ u_{n-1}\ u_n \mid \forall\ u_j \in \{0, 1\}$ for $1 \leq j \leq n\}$ and its range is $\{0, 1\}$. The decision problem with any input of n bits is asking to how many elements from its domain satisfy the condition $O_f(u_1\ u_2\ \ldots\ u_{n-1}\ u_n) = 1$. If the number of elements from its domain that satisfy $O_f(u_1\ u_2\ \ldots\ u_{n-1}\ u_n)$ to have a true value (1) is not equal to zero, then an output is "yes" to the decision problem with any input of n bits. Otherwise, an output is "no" for the decision problem with any input of n bits.

6.3.1 Binary Search Trees for Representing the Domain of a Decision Problem with Any Input of *n* Bits

A *tree* is a finite set of one or more nodes such that there is a specially designated node called the *root* and the remaining nodes are partitioned into $v \geq 0$ disjoint sets T_1, …, T_v, where each of these sets is a tree. T_1, …, T_v are called the subtrees of the root. A *binary tree* is a finite set of nodes that is either empty or contains a root and two disjoint binary trees called the *left* subtree and the *right* subtree.

For any given oracular function O_f: $\{u_1\ u_2\ \ldots\ u_{n-1}\ u_n \mid \forall\ u_j \in \{0, 1\}$ for $1 \leq j \leq n\} \rightarrow \{0, 1\}$, its domain is $\{u_1\ u_2\ \ldots\ u_{n-1}\ u_n \mid \forall\ u_j \in \{0, 1\}$ for $1 \leq j \leq n\}$ and its range is $\{0, 1\}$. A decision problem with any input of n bits is asking to how many elements from its domain satisfy the condition $O_f(u_1\ u_2\ \ldots\ u_{n-1}\ u_n)$ to have a true value (1). We make use of a binary tree in Figure 6.5 to represent the *structure* of the domain that is $\{u_1\ u_2\ \ldots\ u_{n-1}\ u_n \mid \forall\ u_j \in \{0, 1\}$ for $1 \le j \le n\}$. In the binary tree in Figure 6.5, a node stands for a bit of one element in $\{u_1\ u_2\ \ldots\ u_{n-1}\ u_n \mid \forall\ u_j \in \{0, 1\}$ for $1 \le j \le n\}$. The root of the binary tree in Figure 6.5 is u_1. The value of the *left* branch of each node represents that the value of the corresponding bit is equal to zero (0) and the value of the *right* branch of each node stands for that the value of the corresponding bit is equal to one (1). Since the value of the left branch of each node is less than the value of the right branch of each node, we regard the binary tree in Figure 6.5 as a binary search tree.

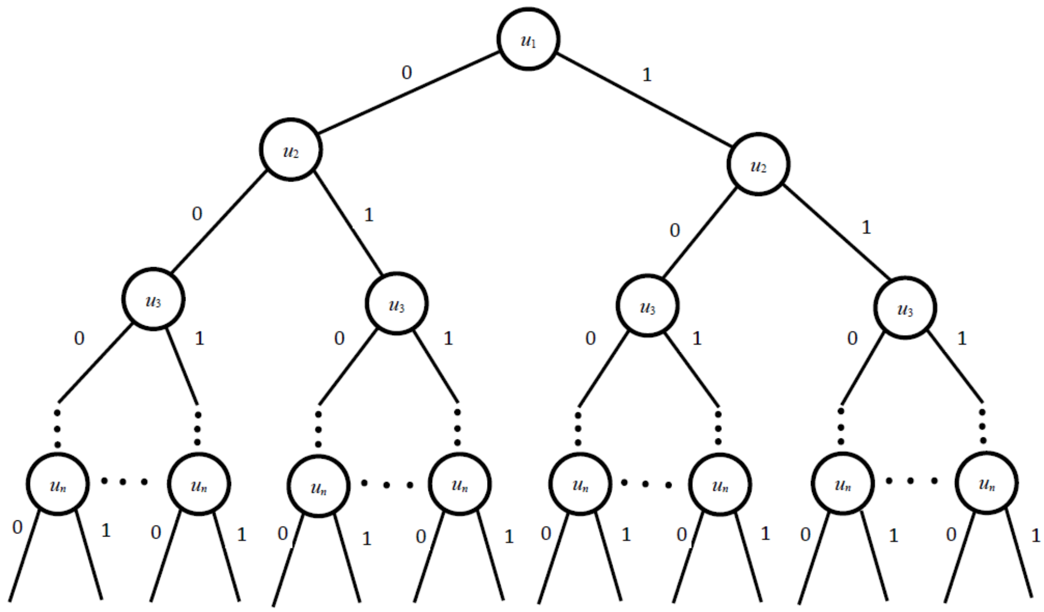

Figure 6.5: A binary search tree for representing the domain of a decision problem with any input of *n* bits.

The binary search tree in Figure 6.5 includes 2^n subtrees and each subtree encodes one element in $\{u_1\ u_2\ \ldots\ u_{n-1}\ u_n \mid \forall\ u_j \in \{0, 1\}$ for $1 \le j \le n\}$. For example, the first subtree $(u_1)\text{--}^0\text{--}(u_2)\text{--}^0\text{--}\ \ldots\ (u_{n-1})\text{--}^0\text{--}(u_n)\text{--}^0\text{--}$ encodes the first element $\{u_1{}^0\ u_2{}^0\ \ldots\ u_{n-1}{}^0\ u_n{}^0\}$. The second subtree $(u_1)\text{--}^0\text{--}(u_2)\text{--}^0\text{--}\ \ldots\ (u_{n-1})\text{--}^0\text{--}(u_n)\text{--}^1\text{--}$ encodes the second element $\{u_1{}^0\ u_2{}^0\ \ldots\ u_{n-1}{}^0\ u_n{}^1\}$. The last subtree $(u_1)\text{--}^1\text{--}(u_2)\text{--}^1\text{--}\ \ldots\ (u_{n-1})\text{--}^1\text{--}(u_n)\text{--}^1\text{--}$ encodes the last element $\{u_1{}^1\ u_2{}^1\ \ldots\ u_{n-1}{}^1\ u_n{}^1\}$.

6.3.2 Flowchart of Solving a Decision Problem with Any Input of *n* Bits

Figure 6.6 is flowchart of solving a decision problem with any input of *n* bits. On the execution of the first statement, S_1, it sets the initial value of $u_1\ u_2\ \ldots\ u_{n-1}\ u_n$ to zero

(0). Next, on the execution of the *second* statement, S_2, it judges whether $O_f(u_1\ u_2\ \ldots\ u_{n-1}\ u_n)$ has a true value (1) or not. If it returns a true value, then on the execution of the *third* statement, S_3, it generates that an output is "yes". Next, on the execution of the *fourth* statement, S_4, it executes one "End" instruction to terminate the processing of solving a decision problem with any input of n bits. Otherwise, on the execution the *fifth* statement, S_5, it increases the value of $u_1\ u_2\ \ldots\ u_{n-1}\ u_n$. Next, on the execution of the *sixth* statement, S_6, it judges whether the value of $u_1\ u_2\ \ldots\ u_{n-1}\ u_n$ is greater than 2^n or not. If it returns a true value, then on the execution of the *seventh* statement, S_7, it produces that an output is "no". Next, on the execution of the *eighth* statement, S_8, it executes one "End" instruction to terminate the processing of solving a decision problem with any input of n bits. Otherwise, it goes to statement S_2 and continues to execute statement S_2.

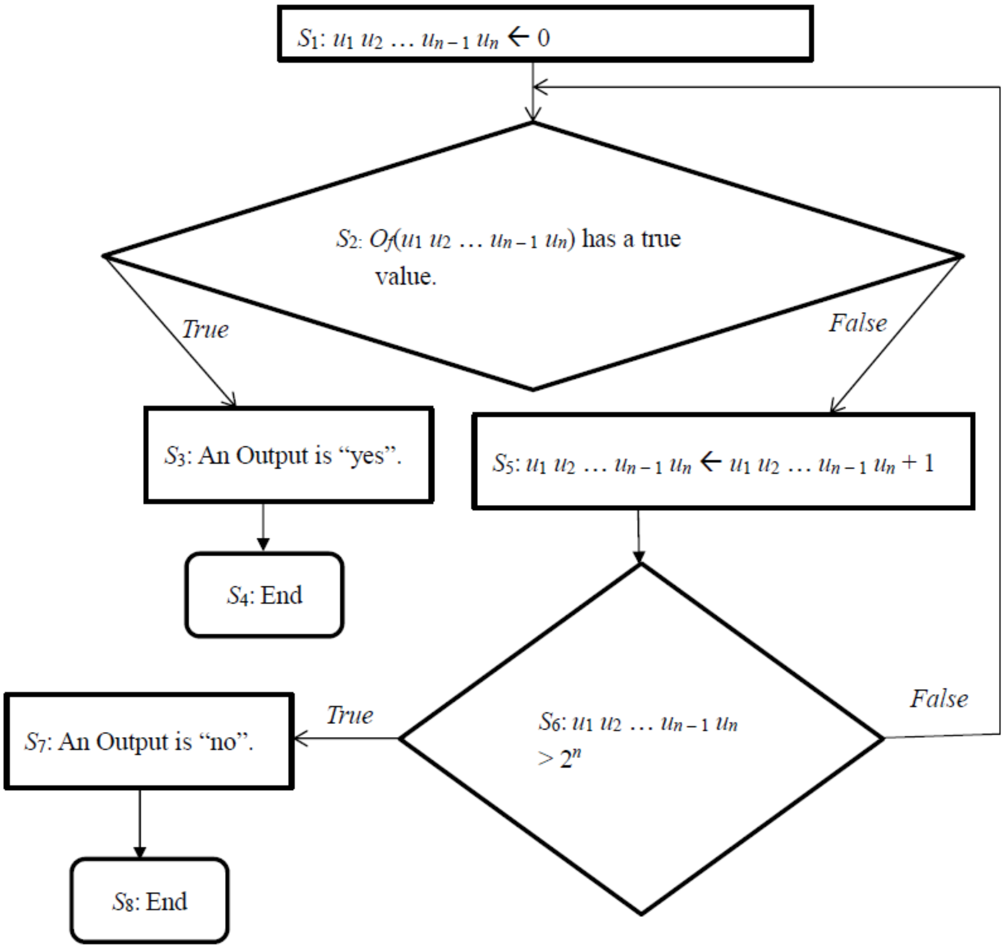

Figure 6.6: Logical flowchart of solving a decision problem with any input of n bits.

6.3.3 Geometrical Interpretation to Solve a Decision Problem with Any Input of *n* Bits

Binary search trees in Figure 6.5 encode $\{u_1\, u_2 \ldots u_{n-1}\, u_n \mid \forall\, u_j \in \{0, 1\}$ for $1 \le j \le n\}$ that is the domain of a decision problem with any input of n bits. We assume that an initial state vector ($|\phi_0\rangle$) is ($\otimes_{j=1}^{n} |u_j^0\rangle$). We begin to make use of a Hadamard transform ($\otimes_{j=1}^{n} H$) on the initial state vector ($|\phi_0\rangle$) that is the register ($\otimes_{j=1}^{n} |u_j^0\rangle$). A superposition of the register is

$$|\phi_1\rangle = \frac{1}{\sqrt{2^n}} \left(\otimes_{j=1}^{n} \left(|u_j^0\rangle + |u_j^1\rangle\right)\right). \quad (6.21)$$

The new state vector ($|\phi_1\rangle$) encodes each subtree in Figure 6.5 with that the amplitude of each subtree is ($\frac{1}{\sqrt{2^n}}$). This is to say that it encodes each element of the domain to a decision problem with any input of n bits.

In the state vector ($|\phi_1\rangle$) in (6.21), subtrees (elements) that satisfy $O_f(u_1\, u_2 \ldots u_{n-1}\, u_n)$ to have a true value (1) are referred as *marked* states and ones that do not result in a solution are referred as *unmarked* states. We assume that N is equal to 2^n. We also assume that in the state vector ($|\phi_1\rangle$) in (6.21), S stands for the number of solution(s) and $(N - S)$ stands for the number of non-solution(s) to a decision problem with any input of n bits. We build two superpositions comprising uniformly distributed computational basis states

$$|\varphi\rangle = \frac{1}{\sqrt{N-S}} \left(\sum_{O_f(u_1\, u_2 \cdots u_n)=0} |u_1\, u_2 \cdots u_n\rangle\right), \quad (6.22)$$

$$|\lambda\rangle = \frac{1}{\sqrt{S}} \left(\sum_{O_f(u_1\, u_2 \cdots u_n)=1} |u_1\, u_2 \cdots u_n\rangle\right). \quad (6.23)$$

Because the inner product of $|\varphi\rangle$ and $|\lambda\rangle$ is equal to zero and the length of $|\varphi\rangle$ and $|\lambda\rangle$ is equal to one, $|\varphi\rangle$ and $|\lambda\rangle$ form an orthonormal basis of a two-dimensional Hilbert space which is depicted in Figure 6.7. In Figure 6.7, Point D is the *original* point of the two-dimensional Hilbert space and its coordinate is (0, 0).

The state vector ($|\phi_1\rangle$) in 6.21 can be expressed as a linear combination of ($|\varphi\rangle$) and ($|\lambda\rangle$) in a two-dimensional Hilbert space of Figure 6.7 in the following way

$$|\phi_1\rangle = \frac{1}{\sqrt{N}} \left(\sum_{O_f(u_1\, u_2 \cdots u_n)=0} |u_1\, u_2 \cdots u_n\rangle + \sum_{O_f(u_1\, u_2 \cdots u_n)=1} |u_1\, u_2 \cdots u_n\rangle\right)$$

$= (\frac{\sqrt{N-S}}{\sqrt{N}}|\varphi\rangle + \frac{\sqrt{S}}{\sqrt{N}}|\lambda\rangle)$. (6.24)

From (6.24), coordinate of ($|\phi_1\rangle$) in a two-dimensional Hilbert space of Figure 6.7 is ($\frac{\sqrt{N-S}}{\sqrt{N}}$, $\frac{\sqrt{S}}{\sqrt{N}}$) and is strictly related to the angle between ($|\phi_1\rangle$) and ($|\varphi\rangle$) denoted by ($\frac{\theta}{2}$) which is depicted in Figure 6.7. Point *B* is coordinate point of ($|\phi_1\rangle$).

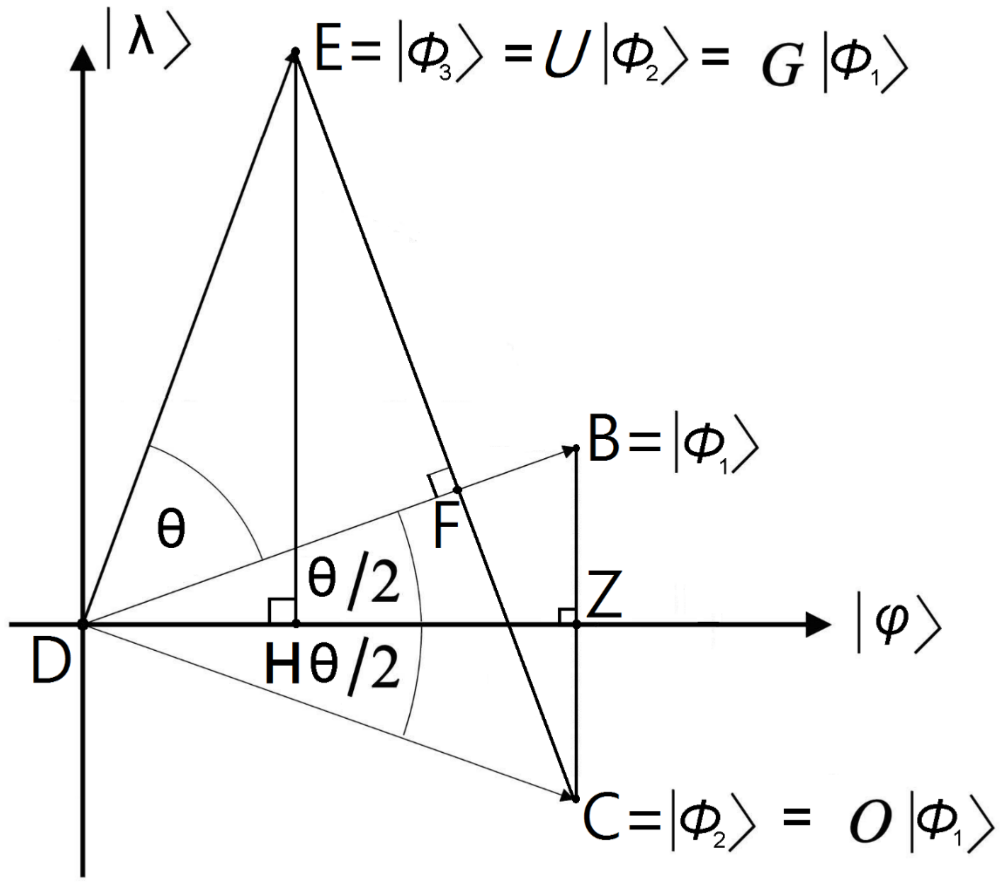

Figure 6.7: Geometrical interpretation of solving a decision problem with any input of *n* bits in a two-dimensional Hilbert space spanned by ($|\varphi\rangle$) and ($|\lambda\rangle$).

In the quantum search algorithm introduced in the third Chapter, the Oracle *O* multiplies the probability amplitude of the answer(s) by −1 and leaves any other amplitude unchanged. We use the Oracle *O* to operate on the state vector ($|\phi_1\rangle$) in (6.21) and obtain the new state vector $|\phi_2\rangle = O(|\phi_1\rangle)$ that can be expressed as a linear combination of ($|\varphi\rangle$) and ($|\lambda\rangle$) in a two-dimensional Hilbert space of Figure 6.7 in the following way

$$|\phi_2\rangle = (\frac{\sqrt{N-S}}{\sqrt{N}}|\varphi\rangle + (-\frac{\sqrt{S}}{\sqrt{N}}|\lambda\rangle)). \quad (6.25)$$

From (6.25), coordinate of ($|\phi_2\rangle$) in a two-dimensional Hilbert space of Figure 6.7 is ($\frac{\sqrt{N-S}}{\sqrt{N}}$, $-\frac{\sqrt{S}}{\sqrt{N}}$) and is depicted in Figure 6.7. Point *C* is coordinate point of ($|\phi_2\rangle$). The angle between ($|\phi_2\rangle$) and ($|\varphi\rangle$) is actually equal to ($\frac{\theta}{2}$) that is depicted in Figure 6.7. The Oracle O is equivalent to a reflection about axis $|\varphi\rangle$ in the two-dimensional geometrical interpretation of Figure 6.7. Because in Figure 6.7 point Z is the intersection of line $\overline{BC}$ and axis $|\varphi\rangle$ in which they are vertical each other, we obtain its coordinate to be ($\frac{\sqrt{N-S}}{\sqrt{N}}$, 0).

In the quantum search algorithm introduced in the third Chapter, the unitary operator *U* is the inversion about the average. The Grover operator *G* consists of two transformations on the index register that are *U* and *O*. We apply the unitary operator *U* to operate on the state vector ($|\phi_2\rangle$) in (6.25) and get the new state vector $|\phi_3\rangle$ = *U* ($|\phi_2\rangle$) = (*U*)(*O*) ($|\phi_1\rangle$) = *G* ($|\phi_1\rangle$). The new state vector ($|\phi_3\rangle$) can be expressed as a linear combination of ($|\varphi\rangle$) and ($|\lambda\rangle$) in a two-dimensional Hilbert space of Figure 6.7 in the following way

$$|\phi_3\rangle = (\frac{\sqrt{N-S}}{\sqrt{N}} \times (\frac{N-4\times S}{N})\ |\varphi\rangle + \frac{\sqrt{S}}{\sqrt{N}} \times (\frac{3\times N-4\times S}{N})\ |\lambda\rangle). \quad (6.26)$$

From (6.26), coordinate of ($|\phi_3\rangle$) in a two-dimensional Hilbert space of Figure 6.7 is ($\frac{\sqrt{N-S}}{\sqrt{N}} \times (\frac{N-4\times S}{N})$, $\frac{\sqrt{S}}{\sqrt{N}} \times (\frac{3\times N-4\times S}{N})$) and is depicted in Figure 6.7. Point *E* is coordinate point of ($|\phi_3\rangle$). The angle between ($|\phi_3\rangle$) and ($|\phi_1\rangle$) is actually equal to (θ) that is depicted in Figure 6.7. The unitary operator *U* (the inversion about the average) in Figure 6.7 reflects its input state ($|\phi_2\rangle$) over ($|\phi_1\rangle$) to ($|\phi_3\rangle$) in the two-dimensional geometrical interpretation of Figure 6.7. In Figure 6.7, point *F* is the intersection of line $\overline{EC}$ and line $\overline{DB}$ in which they are vertical each other and point *H* is the intersection of line $\overline{EH}$ and axis $|\varphi\rangle$ in which they are vertical each other.

6.3.4 Determine the Matrix of the Grover Operator in Geometrical Interpretation to Solve a Decision Problem with Any Input of *n* Bits

From Figure 6.7, point B is $(\frac{\sqrt{N-S}}{\sqrt{N}}, \frac{\sqrt{S}}{\sqrt{N}})$, point D is (0, 0) and point Z is $(\frac{\sqrt{N-S}}{\sqrt{N}}, 0)$. The length of line $\overline{DB}$ is one (1), the length of line $\overline{DZ}$ is $(\sqrt{\frac{N-S}{N}})$ and the length of line $\overline{BZ}$ is $(\sqrt{\frac{S}{N}})$. Therefore, we obtain that $\sin(\theta / 2) = (\sqrt{\frac{S}{N}} / 1) = (\sqrt{\frac{S}{N}})$ and $\cos(\theta / 2) = (\sqrt{\frac{N-S}{N}} / 1) = (\sqrt{\frac{N-S}{N}})$. Because coordinate of $(|\phi_1\rangle)$ in Figure 6.7 is $(\frac{\sqrt{N-S}}{\sqrt{N}}, \frac{\sqrt{S}}{\sqrt{N}})$, its coordinate is also equal to $(\cos(\theta / 2), \sin(\theta / 2))$ in the basis of $(|\varphi\rangle)$ and $(|\lambda\rangle)$. From Figure 6.7, $\sin(\theta + (\theta / 2)) = (\frac{\sqrt{S}}{\sqrt{N}} \times (\frac{3 \times N - 4 \times S}{N}))$ and $\cos(\theta + (\theta / 2)) = (\frac{\sqrt{N-S}}{\sqrt{N}} \times (\frac{N - 4 \times S}{N}))$ are obtained. Since coordinate of $|\phi_3\rangle$ in Figure 6.7 is $(\frac{\sqrt{N-S}}{\sqrt{N}} \times (\frac{N-4 \times S}{N}), \frac{\sqrt{S}}{\sqrt{N}} \times (\frac{3 \times N - 4 \times S}{N}))$, its coordinate is also equal to $(\cos(\theta + (\theta / 2)), \sin(\theta + (\theta / 2)))$ in the basis of $(|\varphi\rangle)$ and $(|\lambda\rangle)$. From Figure 6.7, the matrix of the Grover operator G in the basis of $(|\varphi\rangle)$ and $(|\lambda\rangle)$ is

$$G = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}_{2 \times 2}. \tag{6.27}$$

The matrix of the Grover operator G in the basis of $(|\varphi\rangle)$ and $(|\lambda\rangle)$ is a unitary matrix (a unitary operator) because of $(\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}_{2 \times 2} \times \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}_{2 \times 2} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}_{2 \times 2} \times \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}_{2 \times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{2 \times 2})$. The eigenvalues of the Grover operator G in the basis of $(|\varphi\rangle)$ and $(|\lambda\rangle)$ are

$$(e^{\sqrt{-1} \times \theta}) \text{ and } (e^{-\sqrt{-1} \times \theta}). \tag{6.28}$$

The value of θ is a real. The corresponding eigenvectors of the Grover operator G in the basis of $(|\varphi\rangle)$ and $(|\lambda\rangle)$ are

$$|V_1\rangle = \frac{e^{\sqrt{-1} \times \gamma}}{\sqrt{2}} \begin{bmatrix} \sqrt{-1} \\ 1 \end{bmatrix}_{2 \times 1} \text{ and } |V_2\rangle = \frac{e^{\sqrt{-1} \times \gamma}}{\sqrt{2}} \begin{bmatrix} -\sqrt{-1} \\ 1 \end{bmatrix}_{2 \times 1}. \tag{6.29}$$

The value of γ is a real.

6.3.5 Quantum Counting Circuit to Solve a Decision Problem with Any Input of *n* Bits

From Figure 6.7, we can figure out the projection of $|\phi_1\rangle$ onto axis $|\varphi>$ that is $\sin(\theta / 2) = (\sqrt{\frac{S}{N}} / 1) = (\sqrt{\frac{S}{N}})$. The value of S is to the number of solutions that is how many elements in the domain $\{u_1\ u_2\ \ldots\ u_{n-1}\ u_n \mid \forall\ u_j \in \{0, 1\}$ for $1 \le j \le n\}$ satisfy $O_f(u_1\ u_2\ \ldots\ u_{n-1}\ u_n)$ to have a true value. Because $S = (\sin(\theta / 2))^2 \times N$ and the value of N is known, if we can determine the value of θ, then we can compute the value of S that is the number of solutions. If the value of S is not equal to zero, then an output is "yes" to a decision problem with any input of n bits. Otherwise, an output is "no" to the decision problem with any input of n bits.

Figure 6.8 is quantum-counting circuits that are a real application of phase estimation. In Figure 6.8, if an eigenvalue generated from controlled Grover operations

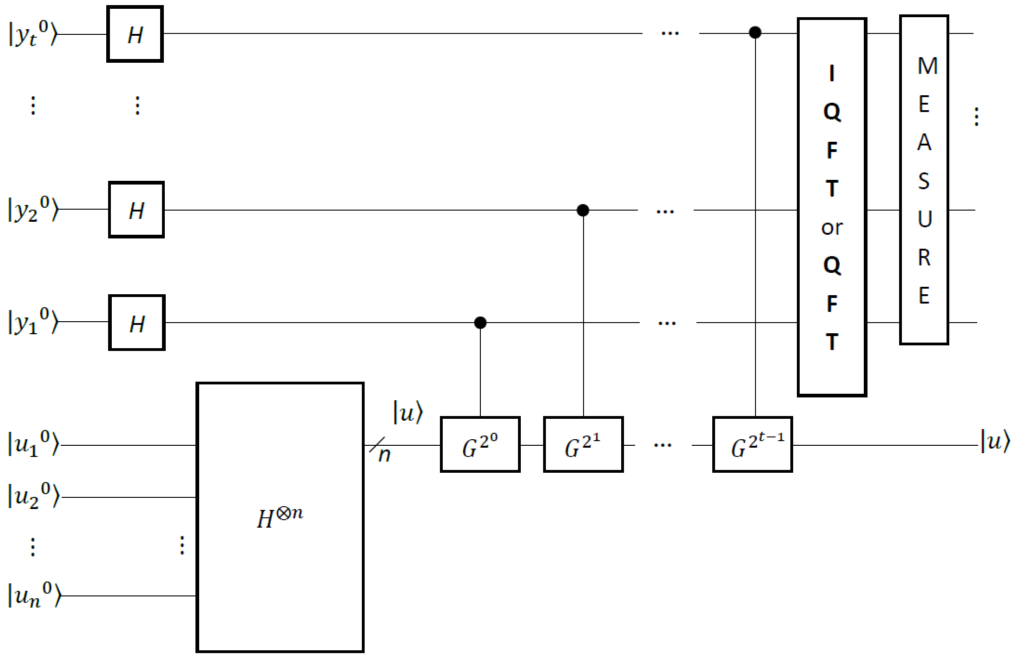

Figure 6.8: Quantum counting circuits to calculate the number of solutions to a decision problem with the input of n bits.

is $(e^{\sqrt{-1}\times\theta})$, then we use controlled Grover operations followed by **inverse quantum Fourier transform** to find the best approximation of t bits to the value of θ. Otherwise, we use controlled Grover operations followed by **quantum Fourier transform** to find the best approximation of t bits to the value of θ. In Figure 6.8, a superposition of the second register is the state vector $|u>$. The state vector $|u>$ is a superposition of $(|\varphi\rangle)$ in (6.22) and $(|\lambda\rangle)$ in (6.23). Because $|V_1\rangle$ and $|V_2\rangle$ in (6.29) form an orthonormal basis of the space spanned by $(|\varphi\rangle)$ in (6.22) and $(|\lambda\rangle)$ in (6.23), the state vector $|u>$ in Figure 6.8 can be expressed as a linear combination of $|V_1\rangle$ and $|V_2\rangle$ in (6.29).

6.4 Determine the Number of Solutions to the Independent-set Problem in a Graph with Two Vertices and One Edge in Phase Estimation

We assume that graph G has is a set V of vertices and a set E of edges. We also suppose that V is $\{v_1, \ldots, v_n\}$ in which each element v_j for $1 \le j \le n$ is a vertex in graph G. We assume that E is $\{(v_a, v_b)| \; v_a \in V \text{ and } v_b \in V\}$. We use $G = (V, E)$ to represent it. We assume that $|V|$ is the number of vertices in V and $|E|$ is the number of edges in E. We also suppose that $|V|$ is equal to n and $|E|$ is equal to m. The value of m is at most equal to $((n \times (n - 1)) / 2)$. For graph $G = (V, E)$, its *complementary* graph is $\bar{G} = (V, \bar{E})$ in which each edge in $\bar{E}$ is out of E. This is to say that $\bar{E}$ is $\{(v_c, v_d)| \; v_c \in V \text{ and } v_d \in V \text{ and } (v_c, v_d) \notin E\}$. We assume that $|\bar{E}|$ is the number of edges in $\bar{E}$. The number of edges in $\bar{E}$ is $(((n \times (n - 1)) / 2) - m)$. An *independent-set* of graph G with n vertices and m edges is a subset $V^1 \subseteq V$ of vertices such that for all $v_c, v_d \in V^1$, the edge (v_c, v_d) is *not* in E. The independent-set problem of graph G with n vertices and m edges is to find a *maximum-sized* independent set in G.

Consider that in Figure 6.9, a graph G^1 contains two vertices $\{v_1, v_2\}$ and one edge

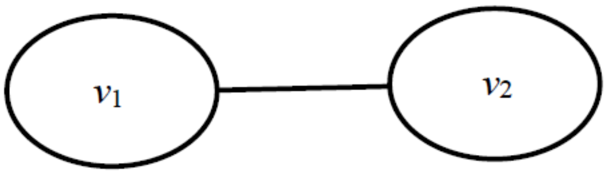

Figure 6.9: A graph G^1 has two vertices and one edge.

$\{(v_1, v_2)\}$ and its complementary graph $\overline{G^1}$ includes the same vertices and zero edge. This is an example of a decision problem that is deciding whether a graph G^1 in Figure 6.9 has a *maximum-sized* independent set or not. All of the subsets of vertex are $\{\}$ that is an empty set, $\{v_1\}$, $\{v_2\}$ and $\{v_1, v_2\}$. Because in $\{v_1, v_2\}$, the edge (v_1, v_2) is one edge of graph G^1, $\{v_1, v_2\}$ does not satisfy definition of an independent set. For other three subsets of vertex that are $\{\}$ that is an empty set, $\{v_1\}$ and $\{v_2\}$, there is no edge in them to connect to other *distinct* vertex. Therefore, they satisfy definition of an independent set. So, all of the independent sets in graph G^1 are {} that is an empty set, $\{v_1\}$ and $\{v_2\}$. Since the number of vertex in them are subsequently zero, one and one, the *maximum-sized* independent set for graph G^1 is $\{v_1\}$ and $\{v_2\}$. Finally, for the decision problem "a graph G^1 in Figure 6.9, does it have a *maximum-sized* independent set?" it gives an output "yes".

For any graph G with n vertices and m edges, all possible independent sets are 2^n possible choices consisting of legal and illegal independent sets in G. Each possible choice corresponds to a subset of vertices in G. Hence, we assume that Y is a set of 2^n possible choices and Y is equal to $\{u_1 u_2 \ldots u_{n-1} u_n \mid \forall u_j \in \{0, 1\} \text{ for } 1 \leq j \leq n\}$. This indicates that the length of each element in Y is n bits and each element represents one of 2^n possible choices. For the sake of presentation, we suppose that u_j^0 is that the value of u_j is zero and u_j^1 is that the value of u_j is one. If an element $u_1 u_2 \ldots u_{n-1} u_n$ in Y is a legal independent set and the value of u_j for $1 \leq j \leq n$ is one, then u_j^1 represents that the j^{th} vertex is within the legal independent set. If an element $u_1 u_2 \ldots u_{n-1} u_n$ in Y is a legal independent set and the value of u_j for $1 \leq j \leq n$ is zero, then u_j^0 represents that the j^{th} vertex is not within the legal independent set. We use superposition of a register with n quantum bits $\left(\frac{1}{\sqrt{2^n}}\left(\otimes_{j=1}^{n}\left(\left|u_j^0\right\rangle + \left|u_j^1\right\rangle\right)\right)\right)$ to encode a set of 2^n possible choices, $Y = \{u_1 u_2 \ldots u_{n-1} u_n \mid \forall u_j \in \{0, 1\} \text{ for } 1 \leq j \leq n\}$.

Deciding whether a graph G^1 with two vertices and one edge in Figure 6.9 has a *maximum-sized* independent set or not is equivalent to compute the number of solution to the same problem. Therefore, we make use of the circuit in Figure 6.10 to determine

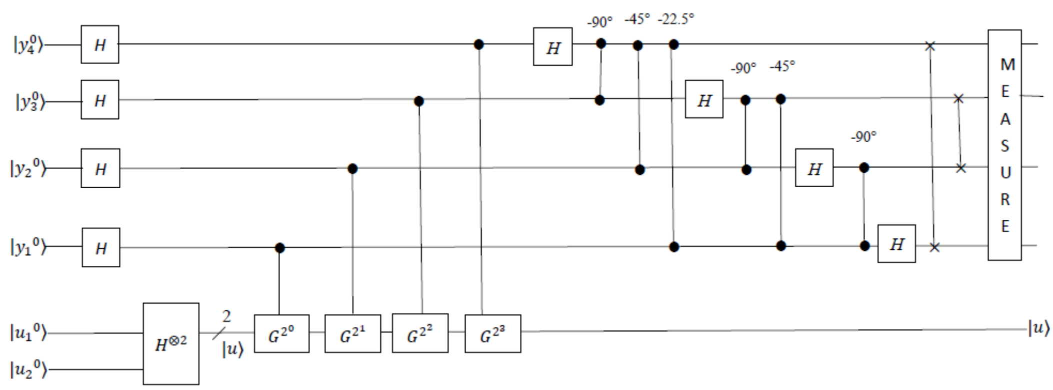

Figure 6.10: Quantum circuit for deciding whether a graph G^1 with two vertices and one edge in Figure 6.9 has a *maximum-sized* independent set or not.

the number of solution to the independent set problem in a graph G^1 with two vertices and one edge in Figure 6.9. It uses two quantum registers. At the left top in Figure 6.10, the first register ($\otimes_{k=4}^{1}|y_k^0\rangle$) includes *four* quantum bits initially in the state |0>. Quantum bit $|y_4{}^0>$ is the most significant bit. Quantum bit $|y_1{}^0>$ is the least significant bit. The corresponding decimal value of the first register is ($|y_4{}^0> \times 2^{4-1}$) + ($|y_3{}^0> \times 2^{3-1}$) + ($|y_2{}^0> \times 2^{2-1}$) + ($|y_1{}^0> \times 2^{1-1}$). At the left bottom in Figure 6.10, the second register ($\otimes_{j=1}^{2}|u_j^0\rangle$) contains *two* quantum bits initially in the state |0>. Quantum bit $|u_1>$ encodes the *first* vertex v_1 in graph G^1 in Figure 6.9 and is the most significant bit. Quantum bit $|u_2>$ encodes the *second* vertex v_2 in graph G^1 in Figure 6.9 and is the least significant bit. Quantum bits $|u_1{}^1> |u_2{}^1>$ encodes $\{v_1, v_2\}$ that is a subset of two vertices. Quantum bits $|u_1{}^1> |u_2{}^0>$ encodes $\{v_1\}$ that is a subset of one vertex. Quantum bits $|u_1{}^0> |u_2{}^1>$ encodes $\{v_2\}$ that is a subset of one vertex. Quantum bits $|u_1{}^0> |u_2{}^0>$ encodes {} that is an empty subset without any vertex. Of course, the corresponding decimal value of the second register is ($|u_1{}^0> \times 2^{2-1}$) + ($|u_2{}^0> \times 2^{2-2}$). For the convenience of the presentation, the following initial state vector is

$$|\varphi_0> = (\otimes_{k=4}^{1}|y_k^0\rangle) \otimes (\otimes_{j=1}^{2}|u_j^0\rangle). \tag{6.30}$$

6.4.1 Initialize Quantum Registers to Calculate the Number of Solutions to the Independent-set Problem in a Graph with Two Vertices and One Edge in Phase Estimation

In Listing 6.2, the program is in the backend that is *simulator* of Open QASM with *thirty-two* quantum bits in **IBM**'s quantum computer. The program is to calculate the number of solutions to the independent-set problem in graph G^1 with two vertices and one edge in Figure 6.9. Figure 6.11 is the corresponding quantum circuit of the program in Listing 6.2 and is to implement the quantum circuit of Figure 6.10 to calculate the number of solutions to the independent-set problem in graph G^1 with two vertices and one edge in Figure 6.9.

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[6];
creg c[4];
```

Listing 6.2: The program of computing the number of solutions to the independent-set problem in graph G^1 with two vertices and one edge in Figure 6.9.

The statement “OPENQASM 2.0;” on line one of Listing 6.2 is to indicate that the program is written with version 2.0 of Open QASM. Then, the statement “include "qelib1.inc";” on line two of Listing 6.2 is to continue parsing the file “qelib1.inc” as if the contents of the file were pasted at the location of the include statement, where the file “qelib1.inc” is **Quantum Experience (QE) Standard Header** and the path is specified relative to the current working directory.

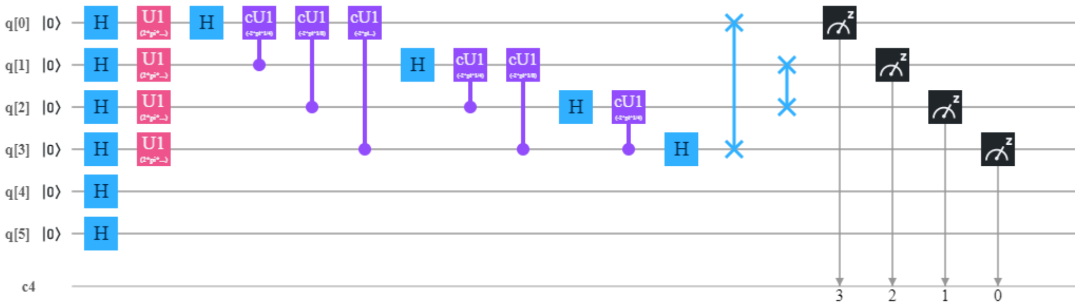

Figure 6.11: Implementing quantum circuits of Figure 6.10 to compute the number of solutions to the independent-set problem in graph G^1 with two vertices and one edge in Figure 6.9.

Next, the statement “qreg q[6];” on line three of Listing 6.2 is to declare that in the program there are *six* quantum bits. In the left top of Figure 6.11, six quantum bits are respectively q[0], q[1], q[2], q[3], q[4] and q[5]. The initial value of each quantum bit is set to state |0>. We use four quantum bits q[0], q[1], q[2] and q[3] to subsequently encode four quantum bits $|y_4>$, $|y_3>$, $|y_2>$ and $|y_1>$ in Figure 6.10. We apply two quantum bits q[4] and q[5] to respectively encode two quantum bits $|u_1>$ and $|u_2>$ in Figure 6.10. For the convenience of our explanation, $q[k]^0$ for $0 \le k \le 5$ is to represent the value 0 of q[k] and $q[k]^1$ for $0 \le k \le 5$ is to represent the value 1 of q[k]. Since quantum bit $|y_4{}^0>$ is the most significant bit and quantum bit $|y_1{}^0>$ is the least significant bit, quantum bit $|q[0]^0>$ is the most significant bit and quantum bit $|q[3]^0>$ is the least significant bit. The corresponding decimal value of the first register in Figure 6.11 is $(|q[0]^0> \times 2^{4-1}) + (|q[1]^0> \times 2^{3-1}) + (|q[2]^0> \times 2^{2-1}) + (|q[3]^0> \times 2^{1-1})$.

Then, the statement “creg c[4];” on line four of Listing 6.2 is to declare that there are four classical bits in the program. In the left bottom of Figure 6.11, four classical bits are respectively c[0], c[1], c[2] and c[3]. The initial value of each classical bit is set to zero (0). For the convenience of our explanation, $c[k]^0$ for $0 \le k \le 3$ is to represent the value 0 of c[k] and $c[k]^1$ for $0 \le k \le 3$ is to represent the value 1 of c[k]. The corresponding decimal value of the four initial classical bits $c[3]^0$ $c[2]^0$ $c[1]^0$ $c[0]^0$ is 2^3

$\times$ c[3]^0 + 2^2 $\times$ c[2]^0 + 2^1 $\times$ c[1]^0 + 2^0 $\times$ c[0]^0. This is to say that classical bit c[3]^0 is the most significant bit and classical bit c[0]^0 is the least significant bit. For the convenience of our explanation, we can rewrite the initial state vector $|\varphi_0>$ = $(\otimes_{k=4}^{1}|y_k^0\rangle) \otimes (\otimes_{j=1}^{2}|u_j^0\rangle)$ in (6.30) in Figure 6.10 as follows

$|\varphi_0> = (\otimes_{k=4}^{1}|y_k^0\rangle) \otimes (\otimes_{j=1}^{2}|u_j^0\rangle) = |q[0]^0> |q[1]^0> |q[2]^0> |q[3]^0> |q[4]^0> |q[5]^0>$. (6.31)

6.4.2 Superposition of Quantum Registers to Compute the Number of Solutions to the Independent-set Problem in a Graph with Two Vertices and One Edge in Phase Estimation

In Figure 6.10, the first stage of the circuit is to implement a Hadamard transform with four Hadamard gates on the *first* register $(\otimes_{k=4}^{1}|y_k^0\rangle)$ and another Hadamard transform with two Hadamrad gates on the *second* register $(\otimes_{j=1}^{2}|u_j^0\rangle)$. The *six* statements "h q[0];", "h q[1];", "h q[2];", "h q[3];", "h q[4];" and "h q[5];" on line *five* of Listing 6.2 through line *ten* of Listing 6.2 is to implement *six* Hadamrad gates on the first register and the second register. They perform each Hadamrad gate in the first time slot of Figure 6.11 and complete the first stage of the circuit in Figure 6.10.

Listing 6.2 continued…

```
//Implement a Hadamard transform on two registers.

5.   h q[0];
6.   h q[1];
7.   h q[2];
8.   h q[3];
9.   h q[4];
10. h q[5];
```

A superposition of the first register is $(\frac{1}{\sqrt{2^4}}\ (\otimes_{k=4}^{1}(|y_k^0\rangle + |y_k^1\rangle)))$ = $(\frac{1}{\sqrt{2^4}}\ (\otimes_{a=0}^{3}(|q[a]^0\rangle + |q[a]^1\rangle)))$. Another superposition of the second register is $(|u\rangle = \frac{1}{\sqrt{2^2}}\ (\otimes_{j=1}^{2}(|u_j^0\rangle + |u_j^1\rangle)) = \frac{1}{\sqrt{2^2}}\ (\otimes_{b=4}^{5}(|q[b]^0\rangle + |q[b]^1\rangle)))$. This implies that the superposition of the second register begins in the new state vector $(|u\rangle = \frac{1}{\sqrt{2^2}}$

$(\otimes_{j=1}^{2}(|u_j^0\rangle + |u_j^1\rangle)) = \frac{1}{\sqrt{2^2}}(\otimes_{b=4}^{5}(|q[b]^0\rangle + |q[b]^1\rangle)))$ and contains *two* quantum bits as is necessary to store $(|u\rangle)$. In superposition of the second register $(|u\rangle)$, state $(|u_1^1\rangle|u_2^1\rangle)$ that is encoded by state $(|q[4]^1\rangle|q[5]^1\rangle)$ with the amplitude (1/2) encodes $\{v_1, v_2\}$ that is a subset of two vertices. State $(|u_1^1\rangle|u_2^0\rangle)$ that is encoded by state $(|q[4]^1\rangle|q[5]^0\rangle)$ with the amplitude (1/2) encodes $\{v_1\}$ that is a subset of one vertex. State $(|u_1^0\rangle|u_2^1\rangle)$ that is encoded by state $(|q[4]^0\rangle|q[5]^1\rangle)$ with the amplitude (1/2) encodes $\{v_2\}$ that is a subset of one vertex. State $(|u_1^0\rangle|u_2^0\rangle)$ that is encoded by state $(|q[4]^0\rangle|q[5]^0\rangle)$ with the amplitude (1/2) encodes {} that is an empty subset without vertex. The new state vector $(|u\rangle)$ is an eigenstate (eigenvector) of *G* that is the Grover operator and is a unitary operator. Thus, this gives that the following new state vector is

$$|\varphi_1\rangle = (\frac{1}{\sqrt{2^4}}(\otimes_{k=4}^{1}(|y_k^0\rangle + |y_k^1\rangle))) \otimes (\frac{1}{\sqrt{2^2}}(\otimes_{j=1}^{2}(|u_j^0\rangle + |u_j^1\rangle)))$$

$$= (\frac{1}{\sqrt{2^4}}(\otimes_{k=4}^{1}(|y_k^0\rangle + |y_k^1\rangle))) \otimes (|u\rangle)$$

$$= (\frac{1}{\sqrt{2^4}}(\otimes_{a=0}^{3}(|q[a]^0\rangle + |q[a]^1\rangle))) \otimes (\frac{1}{\sqrt{2^2}}(\otimes_{b=4}^{5}(|q[b]^0\rangle + |q[b]^1\rangle)))$$

$$= (\frac{1}{\sqrt{2^4}}(\otimes_{a=0}^{3}(|q[a]^0\rangle + |q[a]^1\rangle))) \otimes (|u\rangle). \quad (6.32)$$

6.4.3 Controlled-*G* Operations on the Superposition of the Second Register to Determine the Number of Solutions to the Independent-set Problem in a Graph with Two Vertices and One Edge in Phase Estimation

In the new state vector $|\varphi_1\rangle$ in (6.32), each quantum bit in the first register is currently in its superposition. The value of the first register is from state $(\otimes_{k=4}^{1}|y_k^0\rangle)$ (zero) encoded by state $(\otimes_{a=0}^{3}|q[a]^0\rangle)$ through state $(\otimes_{k=4}^{1}|y_k^1\rangle)$ (fifteen) encoded by state $(\otimes_{a=0}^{3}|q[a]^1\rangle)$ with that the amplitude of each state is (1 / 4). The circuit of Figure 6.10 can precisely estimate sixteen phases. This indicates that the first register with four quantum bits can precisely represent sixteen phases. Sixteen phases are respectively $(0 / 2^4)$, $(1 / 2^4)$, $(2 / 2^4)$, $(3 / 2^4)$, $(4 / 2^4)$, $(5 / 2^4)$, $(6 / 2^4)$, $(7 / 2^4)$, $(8 / 2^4)$, $(9 / 2^4)$, $(10 / 2^4)$, $(11 / 2^4)$, $(12 / 2^4)$, $(13 / 2^4)$, $(14 / 2^4)$ and $(15 / 2^4)$. The corresponding sixteen phase angles are respectively $(2 \times \pi \times 0 / 2^4)$, $(2 \times \pi \times 1 / 2^4)$, $(2 \times \pi \times 2 / 2^4)$, $(2 \times \pi \times 3 / 2^4)$, $(2 \times \pi \times 4 / 2^4)$, $(2 \times \pi \times 5 / 2^4)$, $(2 \times \pi \times 6 / 2^4)$, $(2 \times \pi \times 7 / 2^4)$, $(2 \times \pi \times 8 / 2^4)$, $(2 \times \pi \times 9 / 2^4)$, $(2 \times \pi \times 10 / 2^4)$, $(2 \times \pi \times 11 / 2^4)$, $(2 \times \pi \times 12 / 2^4)$, $(2 \times \pi \times 13 / 2^4)$, $(2 \times \pi \times$

$14 / 2^4$) and ($2 \times \pi \times 15 / 2^4$).

Say that we are trying to compute an eigenvalue of 90°. The number of solutions for the independent-set problem in a graph G^1 with two vertices and one edge in Figure 6.9 is $S = N \times (\sin(\theta / 2))^2 = 4 \times (\sin(90° / 2))^2 = 4 \times (1 / 2) = 2$. This gives that the answer is two for determining the number of solutions for the independent-set problem in a graph G^1 with two vertices and one edge in Figure 6.9. Therefore, the effect of one application of the Grover operator G on its eigenvector (eigenstate) ($|u\rangle$) is ($G \times |u> = e^{\pm\sqrt{-1}\times 2\times\pi\times\theta} \times |u> = e^{\pm\sqrt{-1}\times 2\times\pi\times\frac{4}{2^4}} \times |u>$). So, the effect of repeated application of the Grover operator G on its eigenvector (eigenstate) ($|u\rangle$) is

$$G^a |u> = e^{\pm\sqrt{-1}\times 2\times\pi\times\theta\times a} |u> = e^{\pm\sqrt{-1}\times 2\times\pi\times\frac{4}{2^4}\times a} \times |u>. \quad (6.33)$$

A superposition ($\frac{1}{\sqrt{2}}$ ($|y_1{}^0> + |y_1{}^1>$)) that is encoded by ($\frac{1}{\sqrt{2}}$ ($|q[3]^0> + |q[3]^1>$)) at the weighted position 2^0 is the controlled quantum bit of implementing controlled-G^{2^0} operations on the superposition of the second register that is the state ($|u\rangle$). Similarly, a superposition ($\frac{1}{\sqrt{2}}$ ($|y_2{}^0> + |y_2{}^1>$)) that is encoded by ($\frac{1}{\sqrt{2}}$ ($|q[2]^0> + |q[2]^1>$)) at the weighted position 2^1 is the controlled quantum bit of implementing controlled-G^{2^1} operations on the superposition of the second register that is the state ($|u\rangle$). Then, a superposition ($\frac{1}{\sqrt{2}}$ ($|y_3{}^0> + |y_3{}^1>$)) that is encoded by ($\frac{1}{\sqrt{2}}$ ($|q[1]^0> + |q[1]^1>$)) at the weighted position 2^2 is the controlled quantum bit of implementing controlled-G^{2^2} operations on the superposition of the second register that is the state ($|u\rangle$). Next, a superposition ($\frac{1}{\sqrt{2}}$ ($|y_4{}^0> + |y_4{}^1>$)) that is encoded by ($\frac{1}{\sqrt{2}}$ ($|q[0]^0> + |q[0]^1>$)) at the weighted position 2^3 is the controlled quantum bit of implementing controlled-G^{2^3} operations on the superposition of the second register that is the state ($|u\rangle$).

The Grover operator G has two eigenvalues ($e^{\sqrt{-1}\times 2\times\pi\times\theta}$) and ($e^{-\sqrt{-1}\times 2\times\pi\times\theta}$). We assume that it generates the eigenvalue ($e^{\sqrt{-1}\times 2\times\pi\times\theta}$) = ($e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{2^4}}$). The *four* statements from line *eleven* through line *fourteen* in Listing 6.2 are "u1(2*pi*4/16*1) q[3];", "u1(2*pi*4/16*2) q[2];", "u1(2*pi*4/16*4) q[1];" and "u1(2*pi*4/16*8) q[0];".

Listing 6.2 continued…

```
//Implement controlled-G operations on the superposition of the second register.

11. u1(2*pi*4/16*1) q[3];
12. u1(2*pi*4/16*2) q[2];
13. u1(2*pi*4/16*4) q[1];
14. u1(2*pi*4/16*8) q[0];
```

They take the new state vector (|φ_1>) in (6.32) as their input state vector and implement each controlled-*G* operation on the superposition of the second register in the *second* time slot of Figure 6.11 and in the *second* stage of Figure 6.10. They alert the phase of the state |y_1^1> (|q[3]^1>) is from one (1) to become ($e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times2^0}$) = ($e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times1}$). They alert the phase of the state |y_2^1> (|q[2]^1>) is from one (1) to become ($e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times2^1}$) = ($e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times2}$). They alert the phase of the state |y_3^1> (|q[1]^1>) is from one (1) to become ($e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times2^2}$) = ($e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times4}$) and alert the phase of the state |y_4^1> (|q[0]^1>) is from one (1) to become ($e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times2^3}$) = ($e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times8}$). This gives that the following new state vector is

$$|\varphi_2\rangle = (\frac{1}{\sqrt{2^4}}\ (|y_4^0\rangle + e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times2^3}|y_4^1\rangle) \otimes (|y_3^0\rangle + e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times2^2}|y_3^1\rangle) \otimes$$

$$(|y_2^0\rangle + e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times2^1}|y_2^1\rangle) \otimes (|y_1^0\rangle + e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times2^0}|y_1^1\rangle)) \otimes (|u\rangle)$$

$$= (\frac{1}{\sqrt{2^4}}\ (|y_4^0\rangle + e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times8}|y_4^1\rangle) \otimes (|y_3^0\rangle + e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times4}|y_3^1\rangle) \otimes$$

$$(|y_2^0\rangle + e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times2}|y_2^1\rangle) \otimes (|y_1^0\rangle + e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times1}|y_1^1\rangle)) \otimes (|u\rangle)$$

$$= (\frac{1}{\sqrt{2^4}}\ (|q[0]^0\rangle + e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times8}|q[0]^1\rangle) \otimes (|q[1]^0\rangle + e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times4}|q[1]^1\rangle) \otimes$$

$$(|q[2]^0\rangle + e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times2}|q[2]^1\rangle) \otimes (|q[3]^0\rangle + e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times1}|q[3]^1\rangle)) \otimes (|u\rangle)$$

$$= (\frac{1}{\sqrt{2^4}}\ (\sum_{Y=0}^{2^4-1} e^{\sqrt{-1}\times2\times\pi\times\frac{4}{16}\times Y}\ |Y\rangle)) \otimes (|u\rangle). \quad (6.34)$$

From this description above, the second quantum register stays in the state ($|u>$) through the computation. Because of *phase kickback*, the phase of the state $|Y>$ for $0 \leq Y \leq 2^4 - 1$ is from one (1) to become $(e^{\sqrt{-1}\times 2\times\pi\times\frac{4}{16}\times Y})$. In the state vector ($|\varphi_2>$) in (6.34), it includes sixteen phase angles from state |0> through state |15>. The front eight phase angles are (90° × 0 = 0°), (90° × 1 = 90°), (90° × 2 = 180°), (90° × 3 = 270°), (90° × 4 = 360° = 0°), (90° × 5 = 450° = 90°), (90° × 6 = 540° = 180°) and (90° × 7 = 630° = 270°). The last eight phase angles are (90° × 8 = 720° = 0°), (90° × 9 = 810° = 90°), (90° × 10 = 900° = 180°), (90° × 11 = 990° = 270°), (90° × 12 = 1080° = 0°), (90° × 13 = 1170° = 90°), (90° × 14 = 1260° = 180°) and (90° × 15 = 1350° = 270°). The phase angle rotates back to its starting value 0° *four* times.

6.4.4 The Inverse Quantum Fourier Transform on the Superposition of the First Register to Compute the Number of Solutions to the Independent-set Problem in a Graph with Two Vertices and One Edge in Phase Estimation

Hidden patterns and information stored in the state vector ($|\varphi_2>$) in (6.34) are to that its phase angle rotates back to its starting value 0° *four* times. This is to say that the number of the period per sixteen phase angles is *four* and the frequency is equal to *four* (16 / 4). The twelve statements from line *fifteen* through line *twenty-six* in Listing 6.2

Listing 6.2 continued…

```
//Implement one inverse quantum Fourier transform on the superposition of the first
// register.

15. h q[0];
16. cu1(-2*pi*1/4) q[1],q[0];
17. cu1(-2*pi*1/8) q[2],q[0];
18. cu1(-2*pi*1/16) q[3],q[0];

19. h q[1];
20. cu1(-2*pi*1/4) q[2],q[1];
21. cu1(-2*pi*1/8) q[3],q[1];

22. h q[2];
```

```
23. cu1(-2*pi*1/4) q[3],q[2];

24. h q[3];

25. swap q[0],q[3];
26. swap q[1],q[2];
```

complete each quantum operation from the *third* time slot through the *fourteenth* time slot in Figure 6.11. They actually implement each quantum operation of performing an **inverse quantum Fourier transform** on the superposition of the first register in Figure 6.10. They take the state vector ($|\varphi_2>$) in (6.34) as their input state vector. Since the **inverse quantum Fourier transform** effectively transforms the state of the first register into a superposition of the *periodic* signal's component frequencies, they generate the following state vector

$$|\varphi_3> = (\sum_{Y=0}^{2^4-1} \frac{1}{\sqrt{2^4}} e^{\sqrt{-1} \times 2 \times \pi \times \frac{4}{2^4} \times Y} \frac{1}{\sqrt{2^4}} \sum_{i=0}^{2^4-1} e^{-\sqrt{-1} \times 2 \times \pi \times \frac{i}{2^4} \times Y} |i\rangle) \otimes (|u\rangle)$$

$$= (\frac{1}{2^4} (\sum_{Y=0}^{2^4-1} \sum_{i=0}^{2^4-1} e^{\sqrt{-1} \times 2 \times \pi \times Y \times (\frac{4}{2^4} - \frac{i}{2^4})} |i\rangle)) \otimes (|u\rangle)$$

$$= (\sum_{i=0}^{2^4-1} \sum_{Y=0}^{2^4-1} \frac{1}{2^4} (e^{\sqrt{-1} \times 2 \times \pi \times (\frac{4}{2^4} - \frac{i}{2^4})})^Y |i\rangle)) \otimes (|u\rangle). \quad (6.35)$$

6.4.5 Read the Quantum Result to Figure out the Number of Solutions to the Independent-set Problem in a Graph with Two Vertices and One Edge in Phase Estimation

Finally, the four statements "measure q[0] -> c[3];", "measure q[1] -> c[2];", "measure q[2] -> c[1];" and "measure q[3] -> c[0];" from line *twenty-seven* through line *thirty* in Listing 6.2 implement a measurement. They measure the output state of the inverse quantum Fourier transform to the superposition of the first register in Figure 6.11 and in Figure 6.10. This is to say that they measure four quantum bits q[0], q[1], q[2] and q[3] of the first register and record the measurement outcome by overwriting four classical bits c[3], c[2], c[1] and c[0].

Listing 6.2 continued…

```
//Complete a measurement on the first register.
```

```
27. measure q[0] -> c[3];
28. measure q[1] -> c[2];
29. measure q[2] -> c[1];
30. measure q[3] -> c[0];
```

In the backend *simulator* with thirty-two quantum bits in **IBM**'s quantum computers, we use the command "run" to execute the program in Listing 6.2. Figure 6.12 shows the measured result. From Figure 6.12, we get that a computational basis state 0100 (c[3] = 0 = q[0] = |0>, c[2] = 1 = q[1] = |1>, c[1] = 0 = q[2] = |0> and c[0] = 0 = q[3] = |0>) has the probability 100%. This indicates that the phase angle is $\theta = 2 \times \pi \times (4 / 16) = 90^{\circ}$ with the probability 100%. Hence, The number of solutions for the independent-set problem in a graph G^1 with two vertices and one edge in Figure 6.9 is $S = N \times (\sin(\theta / 2))^2 = 4 \times (\sin(90° / 2))^2 = 4 \times (1 /2) = 2$. This is to say that the answer with the probability 100% is two for computing the number of solutions for the independent-set problem in a graph G^1 with two vertices and one edge in Figure 6.9. Therefore, an output is "yes" to a decision problem that is deciding whether a graph G^1 in Figure 6.9 has a *maximum-sized* independent set or not.

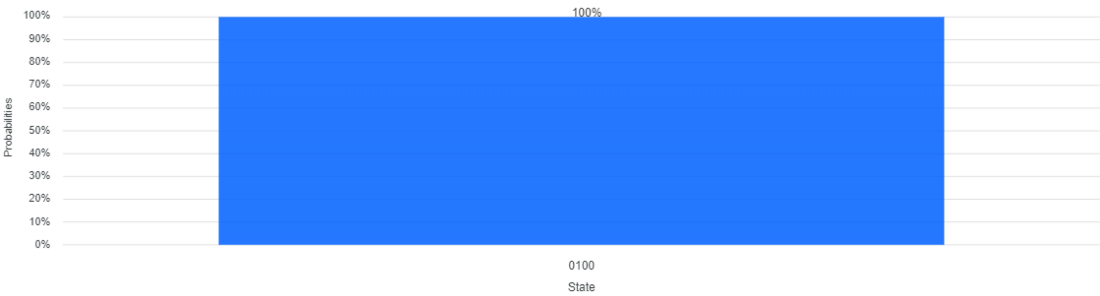

Figure 6.12: A computational basis state 0100 has the probability 100%.

6.5 Summary

In this chapter, we illustrated that a *decision* problem is a problem in which it has only two possible outputs (yes or no) on any input of *n* bits. An output "yes" in the decision problem on any input of *n* bits is to the number of solutions not to be zero and another output "no" in the decision problem on any input of *n* bits is to the number of solutions to be zero. Next, we described that a $(2^n \times 2^n)$ unitary matrix (operator) U has a $(2^n \times 1)$ eigenvector |*u*> with eigenvalue $e^{\sqrt{-1}\times 2\times\pi\times\theta}$ such that $U \times |u> = e^{\sqrt{-1}\times 2\times\pi\times\theta} \times |u>$, where the value of θ is *unknown* and is real. We then illustrated how the phase estimate algorithm with the what possibility estimates the value of θ. We also described time complexity, space complexity and performance of the phase estimate algorithm. Next, we introduced how to design quantum circuits and write quantum programs for computing eigenvalue of a ($2^2 \times 2^2$) unitary matrix U with a ($2^2 \times 1$) eigenvector $|u\rangle$. Next, we described how the quantum-counting algorithm determines the number of solutions for a decision problem with the input of n bits. We also illustrated time complexity, space complexity and performance of the quantum-counting algorithm. We then introduced how to design quantum circuits and write quantum programs to determine the number of solution to the independent set problem in a graph G^1 with two vertices and one edge.

6.6 Bibliographical Notes

In this chapter for more details about an introduction of the phase estimation algorithm, the recommended books are [Nielsen and Chuang 2000; Imre and Balazs 2005; Lipton and Regan 2014; Silva 2018; Johnston et al 2019]. For a more detailed description to binary search trees, the recommended book is [Horowitz et al 2003]. For a more detailed introduction to the discrete Fourier transform and the inverse discrete Fourier transform, the recommended books are [Cormen et al 2009; Nielsen and Chuang 2000; Imre and Balazs 2005; Lipton and Regan 2014; Silva 2018; Johnston et al 2019]. The two famous articles [Coppersmith 1994; Shor 1994] gave the original version of the Quantum Fourier transform and the inverse quantum Fourier transform. A good illustration for the product state decomposition of the quantum Fourier transform and the inverse quantum Fourier transform is the two famous articles in [Griffiths and Niu 1996; Cleve et al 1998]. For a more detailed description to the quantum-counting algorithm, the recommended article and books are [Brassard et al 1998; Nielsen and Chuang 2000; Imre and Balazs 2005; Lipton and Regan 2014; Silva 2018; Johnston et al 2019]. A good introduction to the instructions of Open QASM is the famous article in [Cross et al 2017].

6.7 Exercises

6.1 Prove that the transformation of the Oracle is $O = I_{2^n, 2^n} - 2 \times |x_0\rangle \langle x_0|$, where x_0 is one element in the domain of the Oracle and x_0 satisfies $O(x_0) = 1$.

6.2 Determine the matrix of the Oracle that is $O = I_{2^2, 2^2} - 2 \times |x_0\rangle \langle x_0|$, where $x_0 = 2$ and x_0 satisfies $O(x_0) = 1$.

6.3 Show that the unitary operator U (inversion about the average) is equivalent to reflect its input state $|\phi_2\rangle$ over $|\phi_1\rangle$ to $|\phi_3\rangle$ that is a reflection about $|\phi_1\rangle$ in the two-dimensional geometrical interpretation of Figure 6-7.

6.4 Compute the matrix of the Grover operator *G* in the basis of $(|\varphi\rangle)$ and $(|\lambda\rangle)$ in Figure 6.7.

6.5 Calculate the eigenvalues and corresponding eigenvectors of the Grover operator *G* in the basis of $(|\varphi\rangle)$ and $(|\lambda\rangle)$ in Figure 6.7.