CHAPTER 5

Quantum Gates and Circuits

In quantum circuits, gates are mathematically represented as transformation matrices or linear operators that must be *unitary*. This is a required condition because the norm of a system wave function must be equal to 1 at all times; unitary transformations preserve the norm. Each unitary transformation U has an inverse transformation $U^{-1} = U^{\dagger}$, where $U^{\dagger}$ is the conjugate of U. Thus, quantum computations are reversible.

All computations are performed by applying a succession of these unitary matrices on single or multi-qubit systems. A single qubit is represented as a 2×1 matrix, 2 qubits as a 4×1 matrix, and 3 qubits as a 8×1 matrix. If a gate acts on a single qubit then it is called a *single-qubit gate*; a 2-qubit gate and a 3-qubit code can be defined in a similar manner. Every unitary operator (U) is represented by a 2×2 matrix, in fact every unitary operator is valid single-qubit gate [1,5].

5.1 *X* Gate

It is the quantum equivalent of a classical NOT gate, that is, if $|k\rangle$ is the input to an *X* gate, the output of the gate is $|k'\rangle$. Since the states of a qubit $|0\rangle$ and $|1\rangle$ are represented by the column vectors

$$\begin{pmatrix}1\\0\end{pmatrix} \quad \text{and} \quad \begin{pmatrix}0\\1\end{pmatrix}$$

respectively, therefore

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

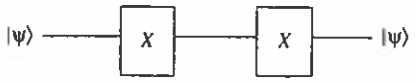
$$X\,|1\rangle \quad \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}=|0\rangle$$

the *X* gate is also called *bit-flip* gate because it inverts each input bit.

The matrix for the X gate is shown below and is the same as Pauli matrix σ_x; it is called X gate because of this reason.

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

Suppose two X gates are connected in series to form a quantum circuit as shown below



A line in the circuit is considered as a *quantum wire* and basically represents a single qubit, thus an input at the input of the first X gate is transformed to $X|\psi>$ and the second X gate acts on it to form $XX|\psi>$. Replacing each X by its matrix representation results in an *identity matrix I:*

$$X \cdot X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I$$

Thus two NOT gates in series is basically equivalent to a *quantum wire,* that is, nothing happens and the output is the same as the original input.

The matrix representation of the state of a superposed qubit $\alpha|0> + \beta|1>$ is given by

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Therefore, if a superposed qubit goes through an X gate, the result will be

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} = \alpha|1> + \beta|0>$$

It should be mentioned here that the X gate "negates" the computational basis states $|0>$ and $|1>$ correctly, it cannot correctly negate an arbitrary superposition state.

5.2 Y Gate

This gate is represented by the Pauli matrix σ_y. It maps $|0>$ to $i|1>$ and $|1>$ to $-i|0>$.

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$Y|0> \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ i \end{pmatrix} = i|1>$$

$$Y|1> \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -i \\ 0 \end{pmatrix} = -i|0>$$

Therefore, the matrix Y defines the transformation:

$$Y(\alpha|0> + \beta|1>) = \alpha Y|0> + \beta Y|1> = i\alpha|1> - i\beta|0>$$

5.3 *Z* Gate

This gate maps input $|k>$ to

$$(-1)^k |k>$$

Thus for an input $|0>$ the output of the Z gate is not changed, that is, also $|0>$ and for an input $|1>$ the output is $-|1>$.

It should be clear from the above definition that the matrix for the Z gate can be written as

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

that is, the same as the Pauli matrix σ_z.

The mapping of $|0>$ and $|1>$ using the matrix for the Z gate shown below:

$$Z|0> = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0>$$

$$Z|1> = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} = -|1>$$

Thus the matrix Z define the transformation:

$$Z(\alpha|0> + \beta|1>) = \alpha Z|0> + \beta Z|1> = \alpha|0> - \beta|1>$$

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} = \alpha|1> + \beta|0>$$

Due to this nature, this transformation is sometimes called a *phase-flip*.

5.4 $\sqrt{NOT}$ (Square Root of NOT) Gate

A square-root NOT gate is a 1-qubit gate that is designed to implement the expression:

$$\sqrt{\text{NOT}} \cdot \sqrt{\text{NOT}} = \text{NOT}$$

There is no such operation in classical logic. The $\sqrt{\text{NOT}}$ gate is a good example of how a gate can exist even though Boolean algebra cannot be used to describe its operation.

A $\sqrt{NOT}$ gate can be represented by the following matrix:

$$\sqrt{NOT} = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix}$$

When a $|0>$ is applied to a $\sqrt{NOT}$ gate, the output is

$$\sqrt{NOT}\,|0> = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix} \begin{vmatrix} 1 \\ 0 \end{vmatrix} = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 \\ 1 \end{vmatrix} = \frac{1}{\sqrt{2}} \left(\begin{vmatrix} 1 \\ 0 \end{vmatrix} + \begin{vmatrix} 0 \\ 1 \end{vmatrix} \right)$$

$$= \frac{1}{\sqrt{2}} (|0> + |1>)$$

Similarly, when a $|1>$ is applied to a $\sqrt{NOT}$ gate, the output is

$$\sqrt{NOT}\,|1> = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix} \begin{vmatrix} 0 \\ 1 \end{vmatrix} = \frac{1}{\sqrt{2}} \begin{vmatrix} -1 \\ 1 \end{vmatrix} = \frac{1}{\sqrt{2}} \left(\begin{vmatrix} 0 \\ 1 \end{vmatrix} - \begin{vmatrix} 1 \\ 0 \end{vmatrix} \right)$$

$$= \frac{1}{\sqrt{2}} (|1> - |0>)$$

Thus when a qubit at state $|0>$ or state $|1>$ is applied to a $\sqrt{NOT}$ gate, it leaves the qubit in an equal superposition of states $|0>$ and $|1$. If the output is applied to another $\sqrt{NOT}$ gate, a superposition of $|0>$ and $|1>$ states will be the input to the second $\sqrt{NOT}$ gate since the unmeasured output of the first gate cannot be assigned any definite value. Thus, the output of the second gate when two $\sqrt{NOT}$ gates are connected in series can be derived as follows from the unmeasured output states of $\sqrt{NOT}\,|0>$ and $\sqrt{NOT}\,|1>$.

When the output of the first $\sqrt{NOT}$ gate $\frac{1}{\sqrt{2}} (|0> + |1>)$, the output of the second $\sqrt{NOT}$ gate will be

$$\frac{1}{\sqrt{2}} \begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix} \begin{vmatrix} 1 \\ 0 \end{vmatrix} \frac{1}{\sqrt{2}} (|0> + |1>)$$

$$= \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix} \begin{vmatrix} 1 \\ 0 \end{vmatrix} \frac{1}{\sqrt{2}} \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right]$$

$$= \frac{1}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$= \frac{1}{2} \left[\begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right] + \frac{1}{2} \left[\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right]$$

$$= \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1>$$

Similarly if the output of the first $\sqrt{NOT}$ gate is $\frac{1}{\sqrt{2}}(|1> - |0>)$, the output of the second $\sqrt{NOT}$ gate is

$$\frac{1}{\sqrt{2}}\begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix}\begin{vmatrix} 1 \\ 0 \end{vmatrix}\frac{1}{\sqrt{2}}(|1> - |0>)$$

$$= \frac{1}{\sqrt{2}}\begin{vmatrix} 1 & -1 \\ 1 & 1 \end{vmatrix}\frac{1}{\sqrt{2}}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right]$$

$$= \frac{1}{2}\begin{pmatrix} -1 \\ 1 \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$= \frac{1}{2}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right] - \frac{1}{2}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix}\right]$$

$$= \frac{1}{2}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right]$$

$$= -\frac{1}{2} \cdot 2\begin{pmatrix} 1 \\ 0 \end{pmatrix} = -1\,|0> = |0>$$

The quantum NOT gate operates exactly like its classical counterpart. If the NOT gate is implemented using two unknown gates in series then these two gates can be assumed to performing as $\sqrt{NOT}$ gates. Thus the logic operation of the NOT gate can be written as

$$\sqrt{NOT} \cdot \sqrt{NOT} = NOT$$

However, no single input and single output classical gate can reproduce the function of a traditional NOT gate when connected in series with another similar gate. In other words, a $\sqrt{NOT}$ gate is a truly nonclassical gate.

5.5 Hadamard Gate

The Hadamard gate is a truly quantum gate and is one of the most important in quantum computing. It has some similar characteristics to the $\sqrt{NOT}$ gate. However, the Hadamard gate, unlike the $\sqrt{NOT}$ gate, is *self-inverse*. It maps input $|m>$ to

$$H\,|m> = \frac{1}{\sqrt{2}}\sum_{k=0,1}(-1)^{mk}\,|k>$$

$$= \frac{|0> + (-1)^m\,|1>)}{\sqrt{2}}$$

The effect of the Hadamard gate on the computational basis states $|0>$ and $|1>$ are as follows, assuming $m = 0$ and 1, respectively:

$$H|0> = H\begin{pmatrix}1\\0\end{pmatrix} = \frac{0> + (-1)^0|1>)}{\sqrt{2}}$$
$$= \frac{0> + |1>)}{\sqrt{2}} = |+>$$
$$H|1> = H\begin{pmatrix}0\\1\end{pmatrix} = \frac{|0> + (-1)^m|1>)}{\sqrt{2}}$$
$$= \frac{|0> - |1>)}{\sqrt{2}} = |->$$

Hence the matrix for the Hadamard gate can be defined as

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$$

Notice that a Hadamard gate converts a $|0>$ state and a $|1>$ to a superposition state of $|0>$ and $|1>$. When the superposed qubit is measured, there is an equal probability of it being in the state $|1>$ or $|0>$.

A Hadamard gate acts on a superposed state $\alpha|0> + \beta|1>$ linearly:

$$H(\alpha|0> + \beta|1>) = H\alpha|0> + H\beta|1>$$
$$= \alpha H|0> + \beta H|1>$$
$$= \alpha \cdot \frac{|0> + |1>)}{\sqrt{2}} + \beta \cdot \frac{|0> - |1>)}{\sqrt{2}}$$
$$= \frac{\alpha + \beta}{\sqrt{2}}|0> + \frac{\alpha - \beta}{\sqrt{2}}|1>$$

A quantum register of size n can not only store an individual string but all possible combinations of n qubit at the same time. A 2-qubit register, for example, can store $|01>$ or $|10>$ at any time:

$$|01> = |0> \otimes |1>$$

$$|11> = |1> \otimes |1>$$

However, the 2-qubit register can also store both strings simultaneously. This can be done by setting the first qubit to a superposition of $|0>$ and $|1>$ instead of just $|0>$ or $|1>$; this can be done by applying the qubit to a Hadamard gate:

$$(\frac{1}{\sqrt{2}}|0> + |1>) \otimes |1>$$
$$= \frac{1}{\sqrt{2}}|01> + |11>$$

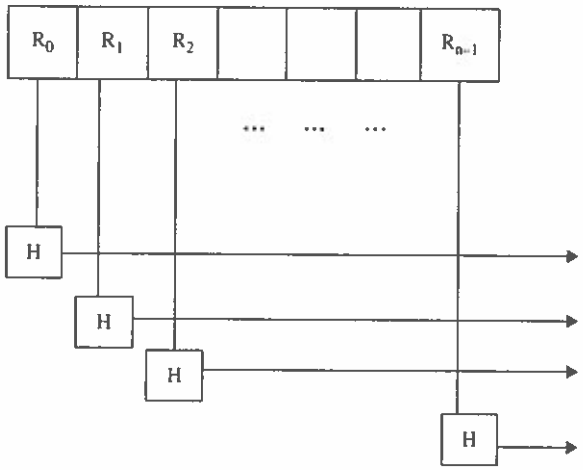


FIGURE 5.1 Generation of all possible bit strings corresponding to n qubits.

This can be extended to all qubits in a quantum register of size n by first preparing each qubit in the state $|0>$ then applying the Hadamard gate to each of the n qubits in parallel as shown in Fig. 5.1. The resulting state of the register is an n-qubit superposition containing 2^n component states; these states are all the possible bit strings corresponding to n qubits.

$$H|0>\otimes H|0>\otimes\cdots\otimes H|0>=\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1}|k>$$

5.6 Phase Gate

This gate turns a $|0>$ into $|0>$, and a $|1>$ into $i|1>$. It is represented by the following matrix:

$$S=\begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}$$

The mapping of $|0>$ and $|1>$ using the matrix for the S gate shown below:

$$S|0>=\begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}=|0>$$

$$S|1>=\begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}0\\-i\end{pmatrix}=i|1>$$

Thus, the matrix S defines the transformation:

$$S(\alpha|0>+\beta|1>)=\alpha S|0>+\beta S|1>$$
$$=\alpha|0>+i\beta|1>$$

Note that a Z gate can be obtained by connecting two S gates in series:

$$S^2=\begin{pmatrix}1&0\\0&i\end{pmatrix}\begin{pmatrix}1&0\\0&i\end{pmatrix}=\begin{pmatrix}1&0\\0&-1\end{pmatrix}=Z$$

5.7 *T* Gate

The matrix for the T gate, also known as the $\frac{\pi}{8}$ gate, is

$$T=\begin{pmatrix}1&0\\0&\exp\left(\frac{i\pi}{4}\right)\end{pmatrix}=\begin{pmatrix}1&0\\0&\frac{(1+i)}{\sqrt{2}}\end{pmatrix}$$

Note that an S gate can be formed by connecting two T gates in series:

$$T^2=\begin{pmatrix}1&0\\0&\frac{(1+i)}{\sqrt{2}}\end{pmatrix}\begin{pmatrix}1&0\\0&\frac{(1+i)}{\sqrt{2}}\end{pmatrix}=\begin{pmatrix}1&0\\0&i\end{pmatrix}=S$$

5.8 Reversible Logic

Loss of energy is a major problem in digital circuits designed using classical gates. In classical gates energy dissipation results due to information loss. Landauer [2] showed that the loss of each bit of information leads to at least

$$kT\log(2)$$

amount of energy release in the form of heat, where k is the Boltzman's constant and T is temperature at which the computation is performed. Although this is a trivial amount of heat, it becomes significant in any major computing system. Such a system carries out many millions of operations per second; as a result it becomes noticeably hot in a short amount of time. The loss of information and the resulting heat generation can be avoided by replacing classical gates with *reversible gates*.

In a reversible gate it is possible to unambiguously determine bits on the inputs of the gate from the outputs of the gate. Classical gates are *irreversible;* all possible input combinations in such a gate are mapped to one of two possible outputs 0 and 1. For example, in two-input AND gates the possible input combinations are 00, 01, 10, and 11. The combinations 00, 01, and 10 map to output 0, only the combination 11 produces an output 1. Thus, when an AND gate produces a 0, the input could be one of the three possibilities and it is not possible to infer what the actual input was. Since in an AND gate a complete knowledge about the input cannot be determined from the output, the operation of the AND gate is not reversible. In a similar manner in all classical gates, the input values cannot be determined from the output value. The NOT gate is an exception. A straightforward way explaining this is that since in classical gates such as two-input AND, OR, NAND, NOR gates a 2-bit input maps into a 1-bit output, one-bit of information is lost on every operation and it is not possible to recover. Thus, classical gates are irreversible. Reversible gates do not lose information. Hence, a reversible gate must have exactly the same number of inputs and outputs; this allows the input value to be uniquely determined from the output value and vice versa. This is the reason a NOT gate with a single input and a single output is reversible. Bennett [3] showed that any computing system can be made reversible by replacing each classical gate in the system with its reversible equivalent. An extremely useful reversible gate for quantum computing circuits is the two-input/two-output controlled-NOT or *CNOT* gate.

5.9 CNOT Gate

A CNOT gate basically implements a reversible EX-OR. It can be used to generate entanglement. The CNOT gate can be graphically represented as in Fig. 5.2, The control and the target inputs are shown as two horizontal lines. The dependence of output y on control a is shown by a vertical line from a to one of the inputs of the EX-OR gate,

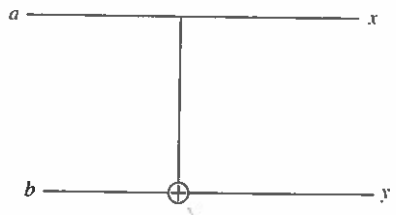


Figure 5.2 CNOT gate.

a	b	x	y
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Table 5.1 Truth Table of CNOT Gate

the other input of the gate is driven by target input b. The truth table of CNOT is shown in Table 5.1.

The input a is typically called the *source*, and input b is known as the *target*. Output $x = a$, that is, x takes the value of source s. The source is also called the *control input* and controls the application of the NOT operation on the target input. Output $y = a \oplus b$, that is, y is the inverse of target b when source is 1 otherwise $y = b$. In other words, whether y gets the inverted value of the target b (or not) is *controlled* by source a. For this reason, CNOT is known as a *controlled NOT gate*. It can be seen from the truth table that in CNOT gate the inputs can be uniquely determined from the outputs, thus verifying the reversibility of the gate and is represented by the matrix:

	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	0	1
11	0	0	1	0

As discussed above, if the target qubit is $|0\rangle$ and the control qubit is either $|0\rangle$ or $|1\rangle$ then the target takes the value of the control qubit, that is, becomes a copy of the control qubit, but the control qubit itself does not change. However, a superposition in the control qubit results in the entanglement of control and target qubits. To illustrate, assume the control input is $|1\rangle$ and the target input is $|0\rangle$. Then the combined state of the control and the target inputs is

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

$$= \frac{1}{\sqrt{2}}(|10\rangle + |00\rangle)$$

Thus the combined input state is a superposition of states $|10\rangle$ and $|00\rangle$. However, when the control input of the CNOT gate is a superposition of $|0\rangle$ and $|1\rangle$, that is, $\alpha|1\rangle+\beta|0\rangle$ and the target qubit $|0\rangle$, the output of the gate is an entangled state that is a superposition of the output states of the two individual inputs $\frac{1}{\sqrt{2}}|10\rangle$ and $\frac{1}{\sqrt{2}}|00\rangle$.

Thus the combined output state of the CNOT gate is

$$\left(\frac{1}{\sqrt{2}}|10\rangle+\frac{1}{\sqrt{2}}|00\rangle\right)$$

Since for a CNOT gate

$$|00\rangle \rightarrow |00\rangle,\ |10\rangle \rightarrow |11\rangle$$

the combined output state is

$$\frac{1}{\sqrt{2}}|11\rangle+\frac{1}{\sqrt{2}}|00\rangle$$

Note that this output state cannot be separated as a product of two single states unlike the combined input state. In other words, the output state is *entangled*.

5.10 Controlled-*U* Gate

A CNOT gate can be extended in a way that it can work on two qubits based upon a single control qubit. Assume U is a single qubit gate that can be represented with a unitary matrix:

$$U=\begin{vmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{vmatrix}$$

Then the *controlled-U gate* can operate on two qubits in such a way that the first qubit serves as a control. The schematic of the gate is shown in Fig. 5.3.

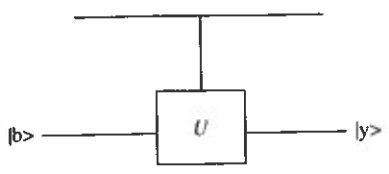


Figure 5.3 Controlled-*U* gate.

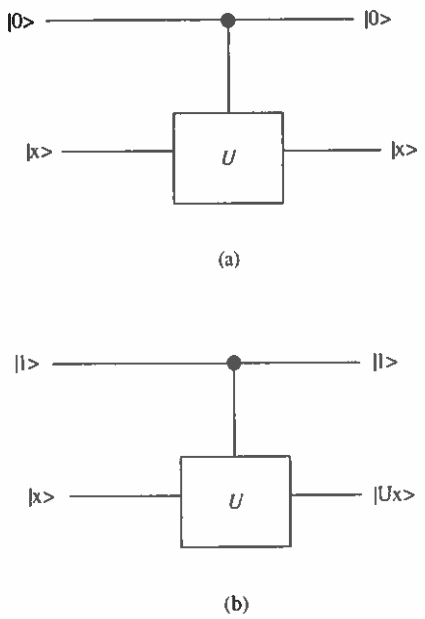


Figure 5.4 (*a*) $|0x\rangle \rightarrow |0x\rangle$; (*b*) $|1x\rangle \rightarrow |1, Ux\rangle$.

The outputs of the gate corresponding to the control bit $a = |0\rangle$ or $|1\rangle$ are shown in Figs. 5.4(*a*) and 5.4(*b*), respectively.

Thus the input-to-output mapping of the gate can be represented as

a	b	x	y
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1,U\rangle$ $\|0\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1,U\rangle$ $\|1\rangle$

Since,

$$U\rangle|0\rangle = \begin{vmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{vmatrix} \begin{vmatrix} 1 \\ 0 \end{vmatrix} = \begin{vmatrix} u_{00} \\ u_{10} \end{vmatrix}$$

$$= (u_{00}|0\rangle + u_{10}|1\rangle)$$

Therefore,

$$|1\rangle|U\rangle|0\rangle = |1\rangle(u_{00}|0\rangle + u_{10}|1\rangle)$$

Similarly,

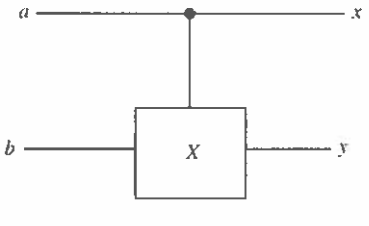
$$|U>|1> = \begin{vmatrix} u_{01} \\ u_{11} \end{vmatrix} = (u_{01}|0> + u_{10}|1>)$$

$$|1>|U>|1> = |1>(u_{01}|0> + u_{10}|1>$$

Hence, the matrix representing the controlled-U gate can be written as

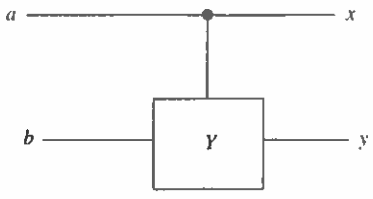
$$\begin{array}{c} \\ 00 \\ 01 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix} \end{array}$$

This indicates that the controlled version of a U-gate matrix can be obtained by substituting the lower two-by-two submatrix of a four-by-four identity matrix with the single qubit matrix of the chosen gate. If U is a Pauli matrix X, Y, or Z, the resulting controlled gates are identified as *controlled-X*, *controlled-Y*, and *controlled-Z* gates, respectively, as depicted in Fig. 5.5[4].



	xy			
a b	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	0	1
11	0	0	1	0

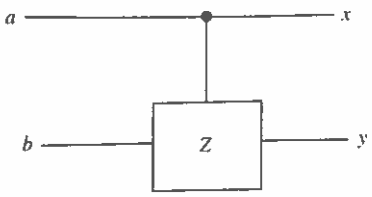
(a)



	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	0	$-i$
11	0	0	i	0

(b)

Figure 5.5 Controlled-U gate. (*a*) Controlled-X gate. (*b*) Controlled-Y gate. (*c*) Controlled-Z gate.



	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	1	0
11	0	0	0	−1

(c)

FIGURE 5.5 *(Continued)*

5.11 Reversible Gates

Landauer [2] showed that three-input/three-output reversible logic gates are extremely useful for classical reversible computation. Two such gates are *Fredkin gate* and *Toffoli gates;* these have been proven to be *universal gates* for irreversible computation. A universal gate allows the building of a circuit corresponding to any Boolean function. Both NAND and NOR gates have been used as universal gates in classical digital circuit design. Fredkin gates and Toffoli gates have been shown to function as NAND gates.

5.11.1 Fredkin Gate (Controlled Swap Gate)

Unlike the quantum gates discussed so far, the Fredkin gate is a three-input gate. It can be seen from the truth table of the gate shown in Fig. 5.6 that when $a = 0$, b is transferred to x and c to y. Alternatively, when $a = 1$, outputs x and y are *swapped*, that is, b and c are transferred to y and x, respectively. Because of this feature, a Fredkin gate is also known as a *controlled swap* (CSWAP) *gate*.

The mapping from input to output is

$$\begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix} \rightarrow \begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 110 \\ 101 \\ 111 \end{pmatrix}$$

a	b	c	w	x	y
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

FIGURE 5.6 Truth table of Fredkin gate.

Therefore, the matrix representing the gate is

	000	001	010	011	100	101	110	111
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	0	0
101	0	0	0	0	0	0	1	0
110	0	0	0	0	0	1	0	0
111	0	0	0	0	0	0	0	1

The structural representation of the gate is shown in Fig. 5.7. It can be seen from the diagram that

$$w = a$$

$$x = a'b \oplus ac$$

$$y = a'c \oplus ab$$

This gate is universal as well as *conservative*; in a conservative gate the number of 0s and 1s remain constant as signals pass through from the input to the output of the gate.

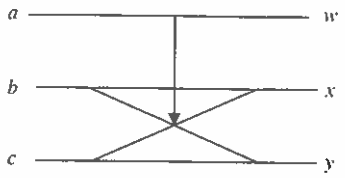


FIGURE 5.7 Fredkin gate.

a	b	c	w	x	y
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Figure 5.8 Truth table for Toffoli gate.

5.11.2 Toffoli Gate (Controlled-Controlled-NOT)

A Toffoli gate also known as a CCNOT gate. It has three inputs; the outputs are the same as the inputs except the third qubit which flips only if the first two qubits are both 1s. The truth table of the Toffoli gate is shown in Fig. 5.8.

It can be seen from the truth table that the mapping from input to output is

$$\begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix} \rightarrow \begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 111 \\ 110 \end{pmatrix}$$

Thus, the matrix representing the Toffoli gate can be represented as shown in Fig. 5.9.

	000	001	010	011	100	101	110	111
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	1	0
101	0	0	0	0	0	1	0	0
110	0	0	0	0	0	0	0	1
111	0	0	0	0	0	0	1	0

Figure 5.9 Matrix corresponding the Toffoli gate.

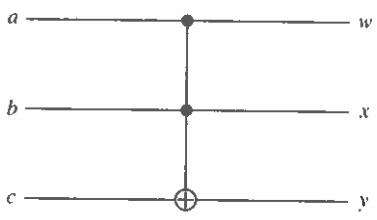


Figure 5.10 Toffoli gate.

Figure 5.10 shows the structural representation of the Toffoli gate that indicates:

$$w = a$$

$$x = b$$

$$y = ab \oplus c$$

5.11.3 Peres Gate

The Peres gate is a modified form of the Toffoli gate. The truth table of the gate is shown Fig. 5.11.

It can be seen from the truth table that the mapping from input to output is

$$\begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix} \rightarrow \begin{pmatrix} 000 \\ 001 \\ 010 \\ 011 \\ 110 \\ 111 \\ 101 \\ 100 \end{pmatrix}$$

a	b	c	w	x	y
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Figure 5.11 Truth table for Peres gate.

	000	001	010	011	100	101	110	111
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	0	0	1	0
101	0	0	0	0	0	0	0	1
110	0	0	0	0	0	1	0	0
111	0	0	0	0	1	0	0	0

Figure 5.12 Matrix for Peres gate.

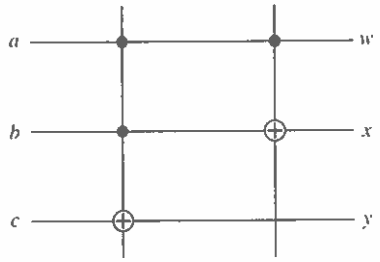


Figure 5.13 Peres gate.

Thus the matrix for the Peres gate is as shown in Fig. 5.12.

The structural representation of the gate is shown in Fig. 5.13. As can be seen in the diagram

$$w = a$$
$$x = a \oplus b$$
$$y = ab \oplus c$$

References

1. Michael Nielsen and Issac Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.
2. R. Landauer, Irreversibility and Heat Generation in the Computational Process, *IBM J.Res.& Dev.*, 5, 183–191, 1961.
3. C. H. Bennett, Logical Reversibility of Computation, *IBM J.Res.& Dev.* 30, 525–532, 1973.
4. Quantum logic gate, Root edit on August 1, 2018. https://en.wikipedia.org/wiki/Quantum_logic_gate#Controlled_(cX_cY_cZ)_gates.
5. David McMohan, *Quantum Computing Explained*, Wiley Interscience, 2008.