Quantum Computing Assignment 8 - Group 18

Rallabhandi, Anand Krishna Mustafa, Syed Husain
, Mohammed Kamran

January 13, 2021

Exercise 8.1

(**Decomposition of Controlled-U Gates**)

(a) Consider $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, & $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

$\mathbf{XYX} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix} = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} = \mathbf{-Y}$

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

$\mathbf{X}\boldsymbol{R_y(\theta)}\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \cos(\frac{\theta}{2}) & \sin(\frac{\theta}{2}) \\ -\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} = \boldsymbol{R_y(-\theta)}$

Properties Used : $\cos(\phi) = \cos(-\phi)$ & $-\sin(\phi) = \sin(-\phi)$

$\mathbf{X}\boldsymbol{R_z(\theta)}\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} e^{\frac{-i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & e^{\frac{-i\theta}{2}} \\ e^{\frac{i\theta}{2}} & 0 \end{pmatrix} = \begin{pmatrix} e^{\frac{i\theta}{2}} & 0 \\ 0 & e^{\frac{-i\theta}{2}} \end{pmatrix} = \boldsymbol{R_z(-\theta)}$

Properties Used : $e^{i\phi} = \cos(\phi) + i\sin(\phi)$ & $e^{-i\phi} = \cos(\phi) - i\sin(\phi)$

(b)

$$ABC = R_z(\beta) * R_y(\frac{\gamma}{2}) * R_y(\frac{-\gamma}{2}) * R_z(\frac{-(\delta+\beta)}{2}) * R_z(\frac{(\delta-\beta)}{2})$$

$$R_y(\frac{\gamma}{2}) * R_y(\frac{-\gamma}{2}) = I$$

$$ABC = \begin{pmatrix} e^{\frac{-i\beta}{2}} & 0 \\ 0 & e^{\frac{i\beta}{2}} \end{pmatrix} * \begin{pmatrix} e^{\frac{i(\delta+\beta)}{4}} & 0 \\ 0 & e^{\frac{-i(\delta+\beta)}{4}} \end{pmatrix} * \begin{pmatrix} e^{\frac{-i(\delta-\beta)}{4}} & 0 \\ 0 & e^{\frac{i(\delta-\beta)}{4}} \end{pmatrix}$$

$$= \begin{pmatrix} e^{\frac{i(\delta-\beta)}{4}} & 0 \\ 0 & e^{\frac{-i(\delta-\beta)}{4}} \end{pmatrix} * \begin{pmatrix} e^{\frac{-i(\delta-\beta)}{4}} & 0 \\ 0 & e^{\frac{i(\delta-\beta)}{4}} \end{pmatrix} = I$$

To prove: $U = e^{i\alpha} AXBXC$

Taking the right-hand side of the equation:

$$e^{i\alpha}[R_z(\beta) * R_y(\frac{\gamma}{2})] * X * R_y(\frac{-\gamma}{2}) * R_z(\frac{-(\delta+\beta)}{2}) * X * R_z(\frac{(\delta-\beta)}{2})$$

$$= e^{i\alpha}[R_z(\beta) * R_y(\frac{\gamma}{2})] * [X * R_y(\frac{-\gamma}{2}) * X] * [X * R_z(\frac{-(\delta+\beta)}{2}) * X] * R_z(\frac{(\delta-\beta)}{2})$$

$$= e^{i\alpha}[R_z(\beta) * R_y(\frac{\gamma}{2})] * R_y(\frac{\gamma}{2}) * R_z(\frac{(\delta+\beta)}{2}) * R_z(\frac{(\delta-\beta)}{2})$$

$$= e^{i\alpha}[R_z(\beta) * R_y(\frac{\gamma}{2})] * R_y(\frac{\gamma}{2}) * \begin{pmatrix} e^{\frac{-i\delta}{2}} & 0 \\ 0 & e^{\frac{i\delta}{2}} \end{pmatrix}$$

$$= e^{i\alpha} R_z(\beta) * \begin{pmatrix} \cos\frac{\gamma}{2} & -\sin\frac{\gamma}{2} \\ \sin\frac{\gamma}{2} & \cos\frac{\gamma}{2} \end{pmatrix} * \begin{pmatrix} e^{\frac{-i\delta}{2}} & 0 \\ 0 & e^{\frac{i\delta}{2}} \end{pmatrix}$$

$$= e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta) = U$$

(c) On solving the quantum circuit using matrix representation we get,

$$\begin{pmatrix} A & 0 \\ 0 & Ae^{i\alpha} \end{pmatrix} * \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} * \begin{pmatrix} B & 0 \\ 0 & B \end{pmatrix} * \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} * \begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix}$$

$$= \begin{pmatrix} A & 0 \\ 0 & AXe^{i\alpha} \end{pmatrix} * \begin{pmatrix} B & 0 \\ 0 & B \end{pmatrix} * \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} * \begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix}$$

$$= \begin{pmatrix} AB & 0 \\ 0 & AXBe^{i\alpha} \end{pmatrix} * \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} * \begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix}$$

$$= \begin{pmatrix} AB & 0 \\ 0 & AXBXe^{i\alpha} \end{pmatrix} * \begin{pmatrix} C & 0 \\ 0 & C \end{pmatrix}$$

$$= \begin{pmatrix} ABC & 0 \\ 0 & e^{i\alpha}AXBXC \end{pmatrix}$$

$$\begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix} = U - controlled$$

(d) Any Unitary Operation can be decomposed as follows:

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta) = \begin{pmatrix} e^{i(\alpha-\frac{\beta}{2}-\frac{\delta}{2})}\cos(\frac{\gamma}{2}) & -e^{i(\alpha-\frac{\beta}{2}+\frac{\delta}{2})}\sin(\frac{\gamma}{2}) \\ e^{i(\alpha+\frac{\beta}{2}-\frac{\delta}{2})}\sin(\frac{\gamma}{2}) & e^{i(\alpha+\frac{\beta}{2}+\frac{\delta}{2})}\cos(\frac{\gamma}{2}) \end{pmatrix}$$

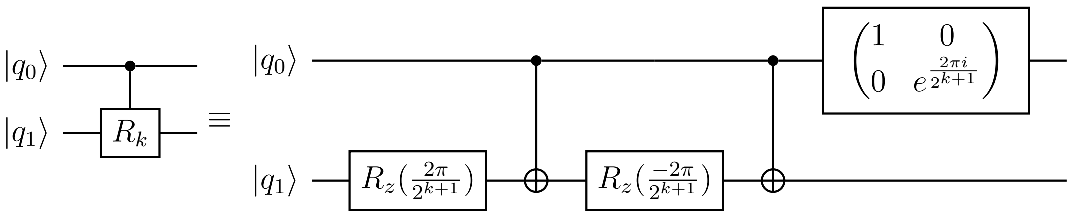
Choosing $\beta = \gamma = 0$

$\implies U = \begin{pmatrix} e^{i(\alpha - \frac{\delta}{2})} & 0 \\ 0 & e^{i(\alpha + \frac{\delta}{2})} \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix} \implies \alpha - \frac{\delta}{2} = 0$ & $\alpha + \frac{\delta}{2} = \frac{2\pi}{2^k} \implies \alpha = \frac{2\pi}{2^{k+1}}$ & $\delta = \frac{2\pi}{2^k}$

Suppose U is a Unitary Gate on a single qubit. Then there exists unitary operators A, B, & C on a single qubit such that ABC=I & $U = e^{i\alpha} AXBXC$, where α is some overall phase vector.

$A = R_z(\beta) R_y(\frac{\gamma}{2})$, $B = R_y(\frac{-\gamma}{2}) R_z(\frac{-(\delta+\beta)}{2})$, & $C = R_z(\frac{\delta-\beta}{2})$

$\implies R_k = e^{\frac{2\pi i}{2^{k+1}}} R_z(0) R_y(0) X R_y(0) R_z(\frac{-2\pi}{2^{k+1}}) X R_z(\frac{2\pi}{2^{k+1}}) = e^{\frac{2\pi i}{2^{k+1}}} X R_z(\frac{-2\pi}{2^{k+1}}) X R_z(\frac{2\pi}{2^{k+1}})$



Exercise 8.2

(Three Qubit Quantum Fourier Transform Implementation)

Exercise 8.2 Solution

In [1]:
```
import numpy as np

def reorder_gate(G, perm):
    """Adapt gate 'G' to an ordering of the qubit as specified in 'perm'.

    Example, given G = np.kron(np.kron(A, B), C):
        reorder_gate(G, [1, 2, 0]) == np.kron(np.kron(B, C), A)
    """
    perm = list(perm)
    # number of qubits
    n = len(perm)
    # reoder both input and output dimensions
    perm2 = perm + [n + i for i in perm]
    return np.reshape(np.transpose(np.reshape(G, 2*n*[2]), perm2), (2**n, 2**n))
```

Implementing Gates ¶

In [2]:
```
H = np.array([
    [1/np.sqrt(2), 1/np.sqrt(2)],
    [1/np.sqrt(2), -1/np.sqrt(2)],
])
Controlled_S = np.array([
    [1, 0, 0, 0],
    [0, 1, 0, 0],
    [0, 0, 1, 0],
    [0, 0, 0, 1j]
])
Controlled_T = np.array([
    [1, 0, 0, 0],
    [0, 1, 0, 0],
    [0, 0, 1, 0],
    [0, 0, 0, np.exp(1j*np.pi/4)]
])
CNOT = np.array([
    [1, 0, 0, 0],
    [0, 1, 0, 0],
    [0, 0, 0, 1],
    [0, 0, 1, 0]
])
SWAP = np.array([
    [1, 0, 0, 0],
    [0, 0, 1, 0],
    [0, 1, 0, 0],
    [0, 0, 0, 1]
])
I = np.eye(2)
```

Fourier Transform Matrx

In [3]:
```
F_Transform = np.array([[np.exp(2*np.pi*1j*j*k/8)/np.sqrt(8) for j in range(8)] for k in range(8)])
```

Circuit Implementation

In [4]:
```
F_Transform_Circuit = np.kron(np.kron(H, I), I) @\
                      reorder_gate(np.kron(Controlled_S, I), [1, 0, 2]) @\
                      reorder_gate(np.kron(Controlled_T, I), [1, 2, 0]) @\
                      np.kron(np.kron(I, H), I) @\
                      reorder_gate(np.kron(I, Controlled_S), [0, 2, 1]) @\
                      np.kron(np.kron(I, I), H) @\
                      reorder_gate(np.kron(SWAP, I), [0, 2, 1])
```

Equivalency Test

In [5]:
```
if(np.round(F_Transform) == np.round(F_Transform_Circuit)).all():
    print(True)
else:
    print(False)
```

```
True
```