Quantum Circuits in 15 Minutes

Mahn-Soo Choi (Korea University)
choims@korea.ac.kr

Three Basic Layers

Consider a quantum register of qubits, which we refer to them by symbol S.

In[ ]:=
```
Let[Qubit, S]
```

For example, we will consider a quantum register of *n* qubits. The qubits are referred to by S[*k*,$]. for *k* = 1, 2, …, *n*.

In[ ]:=
```
$n = 3;
kk = Range[$n];
SS = S[kk, $]
```

Out[ ]=

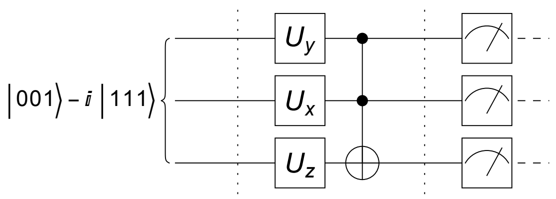
$\{S_1, S_2, S_3\}$

The following example shows the three basic layers of quantum circuits for typical VQA; the initialization, quantum operation, and measurement.

In[ ]:=
```
in = Ket[SS → {0, 0, 1}] - I * Ket[SS → {1, 1, 1}];
mm = Measurement[S[kk, 3]];
```

In[ ]:=
```
qc = QuantumCircuit[
  in, "Separator",
  {Rotation[ϕ[1], S[1, 2]],
   Rotation[ϕ[2], S[2, 1]],
   Rotation[ϕ[3], S[3, 3]]},
  CNOT[S@{1, 2}, S[3]],
  "Separator",
  mm,
  "PortSize" → {2.1, 1}]
```

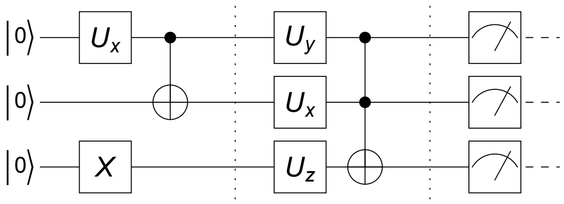
Out[ ]=



The initialization may be specified in further details.

In[∘]:=
```
qc = QuantumCircuit[
  Ket[SS],
  {Rotation[Pi / 2, S[1, 1]], S[3, 1]}, CNOT[S[1], S[2]],
  "Separator",
  {Rotation[ϕ[1], S[1, 2]],
   Rotation[ϕ[2], S[2, 1]],
   Rotation[ϕ[3], S[3, 3]]},
  CNOT[S@{1, 2}, S[3]],
  "Separator",
  mm]
```

Out[∘]=



The VQA requires a step to calculate the expectation value of various physical quantities including the Hamiltonian. In this particular example, we take the statistical average with a small number of measurements.

In[∘]:=
```
Quiet[
   data = Table[Elaborate[qc]; Readout[Measurements@qc], {50}],
   Measurement::nonum
  ]; // EchoTiming
```

1.65784

In[∘]:=
```
avg = Mean[data]
```

Out[∘]=

$$\left\{\frac{17}{25}, \frac{18}{25}, \frac{23}{50}\right\}$$

Single-Qubit Gates

The initialization layer is relatively simple, at least, conceptually.

For the quantum operations layer, one needs to know what operations are available for a given quantum machine.

For the measurements layer, one has to figure out how to implement the measurement of the physical quantities given that a quantum computer can directly measure only the Pauli Z operators on individual qubits. However, the question is essentially the same as for the quantum operations layer.

Therefore, we focus on the second layer here.

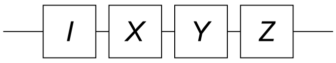
In[∘]:=
```
Let[Qubit, S]
```

Pauli gates

In[ ]:=
```
pauli = S[Full]
```
Out[ ]=

$\{S^0, S^X, S^Y, S^Z\}$

In[ ]:=
```
QuantumCircuit[Sequence @@ pauli]
```
Out[ ]=



In[ ]:=
```
PauliForm[pauli]
```
Out[ ]=

{I, X, Y, Z}

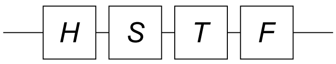
In[ ]:=
```
RL = S[{4, 5}]
```
Out[ ]=

$\{S^+, S^-\}$

In[ ]:=
```
extra = S[{6, 7, 8, 9}]
```
Out[ ]=

$\{S^H, S^S, S^T, S^F\}$

In[ ]:=
```
QuantumCircuit[Sequence @@ extra]
```
Out[ ]=



In[ ]:=
```
MatrixForm /@ Matrix[extra]
```
Out[ ]=

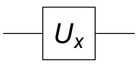
$$\left\{\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{8}} \end{pmatrix}\right\}$$

Rotations

In[ ]:=
```
Let[Real, ϕ]
op = Rotation[ϕ, S[1, 1]]
```
Out[ ]=

Rotation$[\phi, S_1^X]$

In[ ]:=
```
QuantumCircuit[op]
```
Out[ ]=



In[∘]:=
```
new = Elaborate[op]
PauliForm[new]
```
Out[∘]=

$$\mathrm{Cos}\left[\frac{\phi}{2}\right] - i\, S_1^X\, \mathrm{Sin}\left[\frac{\phi}{2}\right]$$

Out[∘]=

$$\mathrm{I}\ \mathrm{Cos}\left[\frac{\phi}{2}\right] - i\, X\, \mathrm{Sin}\left[\frac{\phi}{2}\right]$$

In[∘]:=
```
Dagger[new] ** new // Simplify
```
Out[∘]=

1

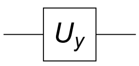
In[∘]:=
```
Matrix[op] // ExpToTrig // MatrixForm
```
Out[∘]//MatrixForm=

$$\begin{pmatrix} \mathrm{Cos}\left[\frac{\phi}{2}\right] & -i\, \mathrm{Sin}\left[\frac{\phi}{2}\right] \\ -i\, \mathrm{Sin}\left[\frac{\phi}{2}\right] & \mathrm{Cos}\left[\frac{\phi}{2}\right] \end{pmatrix}$$

In[∘]:=
```
op = Rotation[ϕ, S[1, 2]]
```
Out[∘]=

$$\mathrm{Rotation}\left[\phi,\ S_1^Y\right]$$

In[∘]:=
```
QuantumCircuit[op]
```
Out[∘]=



In[∘]:=
```
new = Elaborate[op]
PauliForm[new]
```
Out[∘]=

$$\mathrm{Cos}\left[\frac{\phi}{2}\right] - i\, S_1^Y\, \mathrm{Sin}\left[\frac{\phi}{2}\right]$$

Out[∘]=

$$\mathrm{I}\ \mathrm{Cos}\left[\frac{\phi}{2}\right] - i\, Y\, \mathrm{Sin}\left[\frac{\phi}{2}\right]$$

In[∘]:=
```
Dagger[new] ** new // Simplify
```
Out[∘]=

1

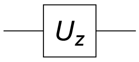
In[∘]:=
```
Matrix[op] // ExpToTrig // MatrixForm
```
Out[∘]//MatrixForm=

$$\begin{pmatrix} \mathrm{Cos}\left[\frac{\phi}{2}\right] & -\mathrm{Sin}\left[\frac{\phi}{2}\right] \\ \mathrm{Sin}\left[\frac{\phi}{2}\right] & \mathrm{Cos}\left[\frac{\phi}{2}\right] \end{pmatrix}$$

In[∘]:=
```
op = Rotation[ϕ, S[1, 3]]
```
Out[∘]=

$$\mathrm{Rotation}\left[\phi,\ S_1^Z\right]$$

In[ ]:=
```
QuantumCircuit[op]
```
Out[ ]=



In[ ]:=
```
new = Elaborate[op]
PauliForm[new]
```
Out[ ]=

$$\mathrm{Cos}\left[\frac{\phi}{2}\right] - i\, S_1^z\, \mathrm{Sin}\left[\frac{\phi}{2}\right]$$

Out[ ]=

$$\mathrm{I}\,\mathrm{Cos}\left[\frac{\phi}{2}\right] - i\, \mathrm{Z}\, \mathrm{Sin}\left[\frac{\phi}{2}\right]$$

In[ ]:=
```
Dagger[new] ** new // Simplify
```
Out[ ]=

1

In[ ]:=
```
Matrix[op] // MatrixForm
```
Out[ ]//MatrixForm=

$$\begin{pmatrix} e^{-\frac{i\phi}{2}} & 0 \\ 0 & e^{\frac{i\phi}{2}} \end{pmatrix}$$

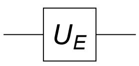
Euler rotations

$$U_E(\{\alpha, \beta, \gamma\}) = U_Z(\alpha)\, U_Y(\beta)\, U_Z(\gamma)$$

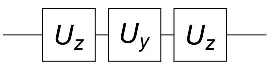
In[ ]:=
```
op = EulerRotation[ϕ@{1, 2, 3}, S[1]]
```
Out[ ]=

EulerRotation[$\{\phi_1, \phi_2, \phi_3\}$, S_1]

In[ ]:=
```
qc1 = QuantumCircuit[op]
qc2 = QuantumCircuit[
   Rotation[ϕ[3], S[1, 3]],
   Rotation[ϕ[2], S[1, 2]],
   Rotation[ϕ[1], S[1, 3]]
  ]
```
Out[ ]=



Out[ ]=



In[ ]:=
```
qc1 - qc2 // Elaborate
```
Out[ ]=

0

In[ ]:= `new = Elaborate[op]`

Out[ ]=

$$\text{Cos}\left[\frac{\phi_2}{2}\right] \text{Cos}\left[\frac{1}{2}(\phi_1+\phi_3)\right] - i\,\text{Cos}\left[\frac{1}{2}(\phi_1-\phi_3)\right] S_1^Y\, \text{Sin}\left[\frac{\phi_2}{2}\right] + i\, S_1^X\, \text{Sin}\left[\frac{\phi_2}{2}\right] \text{Sin}\left[\frac{1}{2}(\phi_1-\phi_3)\right] - i\,\text{Cos}\left[\frac{\phi_2}{2}\right] S_1^Z\, \text{Sin}\left[\frac{1}{2}(\phi_1+\phi_3)\right]$$

In[ ]:= `PauliForm[new]`

Out[ ]=

$$I\, \text{Cos}\left[\frac{\phi_2}{2}\right] \text{Cos}\left[\frac{1}{2}(\phi_1+\phi_3)\right] - i\,\text{Cos}\left[\frac{1}{2}(\phi_1-\phi_3)\right] Y\, \text{Sin}\left[\frac{\phi_2}{2}\right] + i\, X\, \text{Sin}\left[\frac{\phi_2}{2}\right] \text{Sin}\left[\frac{1}{2}(\phi_1-\phi_3)\right] - i\,\text{Cos}\left[\frac{\phi_2}{2}\right] Z\, \text{Sin}\left[\frac{1}{2}(\phi_1+\phi_3)\right]$$

In[ ]:= `Dagger[new] ** new // Simplify`

Out[ ]=

1

In[ ]:= `Matrix[new] // MatrixForm`

Out[ ]//MatrixForm=

$$\begin{pmatrix} \frac{1}{2} e^{-\frac{1}{2} i \phi_2 - \frac{1}{2} i (\phi_1+\phi_3)} + \frac{1}{2} e^{\frac{1}{2} i \phi_2 - \frac{1}{2} i (\phi_1+\phi_3)} & -\frac{1}{2} i\, e^{-\frac{1}{2} i \phi_1 - \frac{1}{2} i \phi_2 + \frac{1}{2} i \phi_3} + \frac{1}{2} i\, e^{-\frac{1}{2} i \phi_1 + \frac{1}{2} i \phi_2 + \frac{1}{2} i \phi_3} \\ \frac{1}{2} i\, e^{\frac{1}{2} i \phi_1 - \frac{1}{2} i \phi_2 - \frac{1}{2} i \phi_3} - \frac{1}{2} i\, e^{\frac{1}{2} i \phi_1 + \frac{1}{2} i \phi_2 - \frac{1}{2} i \phi_3} & \frac{1}{2} e^{-\frac{1}{2} i \phi_2 + \frac{1}{2} i (\phi_1+\phi_3)} + \frac{1}{2} e^{\frac{1}{2} i \phi_2 + \frac{1}{2} i (\phi_1+\phi_3)} \end{pmatrix}$$

Problems

1. Why are they called “rotations”?

2. Given a unitary matrix (operator) *U*, can you decompose it into a product of rotations?

Two-Qubit Gates

In[ ]:= `Let[Qubit, S]`

CNOT (CX, Controlled-X)

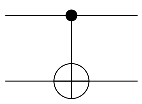
In[ ]:= `op = CNOT[S[1], S[2]]`

Out[ ]=

$\text{CNOT}[\{S_1\} \to \{1\}, \{S_2\}]$

In[ ]:= `qc = QuantumCircuit[op]`

Out[ ]=



In[ ]:=
```
new = Elaborate[qc]
new // PauliForm
```

Out[ ]=
$$\frac{1}{2} - \frac{1}{2} S_1^Z S_2^X + \frac{1}{2} S_1^Z + \frac{1}{2} S_2^X$$

Out[ ]=
$$\frac{I \otimes I}{2} + \frac{I \otimes X}{2} + \frac{Z \otimes I}{2} - \frac{Z \otimes X}{2}$$

In[ ]:=
```
in = Basis[S@{1, 2}]
```

Out[ ]=
$$\{ |0_{S_1} 0_{S_2}\rangle, |0_{S_1} 1_{S_2}\rangle, |1_{S_1} 0_{S_2}\rangle, |1_{S_1} 1_{S_2}\rangle \}$$

In[ ]:=
```
out = op ** in
```

Out[ ]=
$$\{ |0_{S_1} 0_{S_2}\rangle, |0_{S_1} 1_{S_2}\rangle, |1_{S_1} 1_{S_2}\rangle, |1_{S_1} 0_{S_2}\rangle \}$$

In[ ]:=
```
Thread[in → out] // TableForm
```

Out[ ]//TableForm=
$$|0_{S_1} 0_{S_2}\rangle \to |0_{S_1} 0_{S_2}\rangle$$
$$|0_{S_1} 1_{S_2}\rangle \to |0_{S_1} 1_{S_2}\rangle$$
$$|1_{S_1} 0_{S_2}\rangle \to |1_{S_1} 1_{S_2}\rangle$$
$$|1_{S_1} 1_{S_2}\rangle \to |1_{S_1} 0_{S_2}\rangle$$

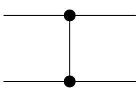
CZ (Controlled-Z)

In[ ]:=
```
op = CZ[S[1], S[2]]
```

Out[ ]=
$$CZ[\{S_1\}, \{S_2\}]$$

In[ ]:=
```
qc = QuantumCircuit[op]
```

Out[ ]=



In[ ]:=
```
new = Elaborate[qc]
new // PauliForm
```

Out[ ]=
$$\frac{1}{2} - \frac{1}{2} S_1^Z S_2^Z + \frac{1}{2} S_1^Z + \frac{1}{2} S_2^Z$$

Out[ ]=
$$\frac{I \otimes I}{2} + \frac{I \otimes Z}{2} + \frac{Z \otimes I}{2} - \frac{Z \otimes Z}{2}$$

In[ ]:=
```
in = Basis[S@{1, 2}]
```
Out[ ]=

$\{|0_{S_1} 0_{S_2}\rangle, |0_{S_1} 1_{S_2}\rangle, |1_{S_1} 0_{S_2}\rangle, |1_{S_1} 1_{S_2}\rangle\}$

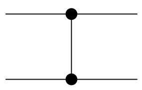
In[ ]:=
```
out = op ** in
```
Out[ ]=

$\{|0_{S_1} 0_{S_2}\rangle, |0_{S_1} 1_{S_2}\rangle, |1_{S_1} 0_{S_2}\rangle, -|1_{S_1} 1_{S_2}\rangle\}$

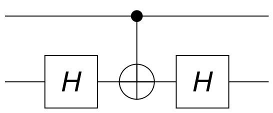
In[ ]:=
```
Thread[in → out] // TableForm
```
Out[ ]//TableForm=

$|0_{S_1} 0_{S_2}\rangle \to |0_{S_1} 0_{S_2}\rangle$
$|0_{S_1} 1_{S_2}\rangle \to |0_{S_1} 1_{S_2}\rangle$
$|1_{S_1} 0_{S_2}\rangle \to |1_{S_1} 0_{S_2}\rangle$
$|1_{S_1} 1_{S_2}\rangle \to -|1_{S_1} 1_{S_2}\rangle$

In[ ]:=
```
qc = QuantumCircuit[CZ[S[1], S[2]]]
new = QuantumCircuit[S[2, 6], CNOT[S[1], S[2]], S[2, 6]]
```
Out[ ]=



Out[ ]=

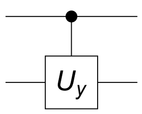


In[ ]:=
```
new - qc // Elaborate
```
Out[ ]=

0

Controlled-Unitary Gates

In[ ]:=
```
qc = QuantumCircuit[ControlledGate[S[1], Rotation[ϕ, S[2, 2]]]]
```
Out[ ]=



Universal Set of Quantum Gates

In[ ]:=
```
Let[Qubit, S]
```

In[ ]:=
```
$n = 2;
kk = Range[$n];
SS = S[kk, $]
```

Out[ ]=

$\{S_1, S_2\}$

In[ ]:=
```
mat = RandomUnitary[Power[2, $n]];
mat // MatrixForm
```

Out[ ]//MatrixForm=

```
( 0.6591 - 0.160089 i      0.375249 + 0.131382 i    -0.246705 + 0.351941 i   0.351712 - 0.2
  0.154807 + 0.18311 i     0.745781 - 0.153574 i    0.0156201 - 0.354704 i   -0.462193 + 0.:
  0.605769 - 0.268087 i    -0.314039 + 0.185861 i   0.488427 - 0.319079 i    -0.121779 + 0.:
  0.0911371 + 0.187965 i   -0.0901809 - 0.347643 i  0.102929 - 0.581591 i    0.324045 - 0.6
```

In[ ]:=
```
op = Elaborate@ExpressionFor[mat, S@{1, 2}];
op // PauliForm
```

Out[ ]=

```
(0.554338 - 0.310977 i) I⊗I + (0.127802 + 0.00068167 i) I⊗X -
 (0.199922 + 0.00106654 i) I⊗Y + (0.0194256 + 0.0713936 i) I⊗Z -
 (0.0483273 + 0.0279913 i) X⊗I + (0.0361078 - 0.0629773 i) X⊗X -
 (0.0203923 + 0.0172709 i) X⊗Y + (0.227859 + 0.0699182 i) X⊗Z -
 (0.279874 + 0.306121 i) Y⊗I + (0.24989 + 0.147559 i) Y⊗X -
 (0.185317 + 0.0214445 i) Y⊗Y - (0.0301401 + 0.120116 i) Y⊗Z +
 (0.148102 + 0.154146 i) Z⊗I + (0.137227 + 0.156564 i) Z⊗X +
 (0.225786 + 0.111288 i) Z⊗Y - (0.0627658 + 0.0746514 i) Z⊗Z
```

In[ ]:=
```
twl = TwoLevelDecomposition[mat];
MatrixForm /@ Matrix /@ twl
```

Out[ ]=

$$\left\{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1.+0.\,i & 0 \\ 0 & 0 & 0 & -0.823834-0.566831\,i \end{pmatrix},\right.$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.872117-0.385961\,i & 0.261485-0.148565\,i \\ 0 & 0 & -0.261485-0.148565\,i & 0.872117+0.385961\,i \end{pmatrix},$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.210674-0.249191\,i & 0.945262 & 0 \\ 0 & -0.945262 & -0.210674+0.249191\,i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -0.957198+0.0437554\,i & 0.258899-0.121773\,i \\ 0 & 0 & -0.258899-0.121773\,i & -0.957198-0.0437554\,i \end{pmatrix},$$

$$\begin{pmatrix} 0.6591-0.160089\,i & 0.734818 & 0 & 0 \\ -0.734818 & 0.6591+0.160089\,i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

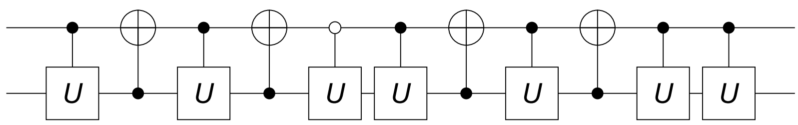
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.510669+0.178795\,i & 0.840981 & 0 \\ 0 & -0.840981 & 0.510669-0.178795\,i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\left.\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -0.399219+0.569512\,i & 0.569143-0.438583\,i \\ 0 & 0 & -0.569143-0.438583\,i & -0.399219-0.569512\,i \end{pmatrix}\right\}$$

In[ ]:=
```
gates = FromTwoLevelU[#, SS] & /@ twl;
```

In[ ]:=
```
qc = QuantumCircuit[Sequence @@ Reverse[Flatten@gates]]
```

Out[ ]=



Problems

1. Given a unitary matrix (operator) U on more than one qubit, can you decompose it into a product of elementary gates?

Summary

Keywords

- Single-qubit gates
- Rotation, Euler rotation
- Two-qubit gates
- CNOT, Controlled-unitary gates

Related Links

- Q3 Tutorial: Single-Qubit Gates
- Q3 Tutorial: Two-Qubit Gates
- A Quantum Workbook (Springer, 2022), Chapter 2.