Q3 SYMBOL

See Also Related Guides Tech Notes

URL

Q3`

AmplitudeEmbedding

NEW IN 13.3

AmplitudeEmbedding[$\{x_1, x_2, \ldots, x_{2^n}\}, \{s_1, s_2, \ldots, s_n\}$]

returns a quantum state on qubits $\{s_1, s_2, \ldots, s_n\}$, the amplitudes of which encode classical input data $\{x_1, x_2, \ldots, x_{2^n}\}$.

Details and Options

- The *amplitude embedding* is mapping, $\{x_1, x_2, \ldots, x_{2^n}\} \mapsto \sum_{k=1}^{2^n} |k-1\rangle x_k$, where $|a\rangle := |a_1\rangle \otimes |a_2\rangle \otimes \ldots \otimes |a_n\rangle$ and $a := (a_1 a_2 \ldots a_n)_2$ is the binary-digit representation of integer a.

Examples (1)

In[1]:=
```
Needs["Q3`"]
```

Basic Examples (1)

In[1]:=
```
Let[Qubit, S]
```

We consider a quantum register of *n* qubits.

In[2]:=
```
$n = 4;
kk = Range[$n];
SS = S[kk, $]
```

Out[2]= $\{S_1, S_2, S_3, S_4\}$

Suppose that we have a classical input data.

In[3]:=
```
Let[Real, x]
xx = x[Range@Power[2, $n]]
```

Out[3]= $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}\}$

The amplitude embedding constructs the following quantum state.

In[4]:=
```
AmplitudeEmbedding[xx, SS]
```

Out[4]= $|0_{s_1} 0_{s_2} 0_{s_3} 0_{s_4}\rangle x_1 + |0_{s_1} 0_{s_2} 0_{s_3} 1_{s_4}\rangle x_2 + |0_{s_1} 0_{s_2} 1_{s_3} 0_{s_4}\rangle x_3 + |0_{s_1} 0_{s_2} 1_{s_3} 1_{s_4}\rangle x_4 + |0_{s_1} 1_{s_2} 0_{s_3} 0_{s_4}\rangle x_5 + |0_{s_1} 1_{s_2} 0_{s_3} 1_{s_4}\rangle x_6 + |0_{s_1} 1_{s_2} 1_{s_3} 0_{s_4}\rangle x_7 + |0_{s_1} 1_{s_2} 1_{s_3} 1_{s_4}\rangle x_8 + |1_{s_1} 0_{s_2} 0_{s_3} 0_{s_4}\rangle x_9 + |1_{s_1} 0_{s_2} 0_{s_3} 1_{s_4}\rangle x_{10} + |1_{s_1} 0_{s_2} 1_{s_3} 0_{s_4}\rangle x_{11} + |1_{s_1} 0_{s_2} 1_{s_3} 1_{s_4}\rangle x_{12} + |1_{s_1} 1_{s_2} 0_{s_3} 0_{s_4}\rangle x_{13} + |1_{s_1} 1_{s_2} 0_{s_3} 1_{s_4}\rangle x_{14} + |1_{s_1} 1_{s_2} 1_{s_3} 0_{s_4}\rangle x_{15} + |1_{s_1} 1_{s_2} 1_{s_3} 1_{s_4}\rangle x_{16}$

On an actual quantum computer, the above quantum state must be achieved through a unitary gate. This is represented by the AmplitudeEmbedding function.

©1988–2023 Wolfram Research, Inc. All rights reserved. http://reference.wolfram.com/language

In[5]:=
```
op = AmplitudeEmbeddingGate[xx, SS]
```

Out[5]=
$$\text{AmplitudeEmbeddingGate}[\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}\}, \{S_1, S_2, S_3, S_4\}]$$

The above unitary gate is decomposed into two uniformly controlled rotations. See AmplitudeEmbeddingGate and UniformlyControlledRotation.

See Also

AmplitudeEmbeddingGate ▪ UniformlyControlledRotation ▪ BasisEmbedding ▪ BasisEmbeddingGate

Tech Notes

- Multi–Control Unitary Gates
- Quantum Information Systems with Q3
- Quick Quantum Computing with Q3
- Q3: Quick Start

Related Guides

- Quantum Information Systems
- Q3

Related Links

- M. Nielsen and I. L. Chuang (2022) , Quantum Computation and Quantum Information (Cambridge University Press, 2011).
- Mahn–Soo Choi (2022) , A Quantum Computation Workbook (Springer, 2022).

©1988–2023 Wolfram Research, Inc. All rights reserved. http://reference.wolfram.com/language

Q3 SYMBOL

See Also Related Guides Tech Notes URL

Q3`

AmplitudeEmbeddingGate

NEW IN 13.3

AmplitudeEmbeddingGate [$\{x_1, x_2, \ldots, x_{2^n}\}, \{s_1, s_2, \ldots, s_n\}$]

represents the gate to encode classical input data $\{x_1, x_2, \ldots, x_{2^n}$ into the amplitudes of a quantum state of qubits $s_1, s_2, \ldots, s_n$.

Details and Options

- The *amplitude embedding* is mapping, $\{x_1, x_2, \ldots, x_{2^n}\} \mapsto \sum_{k=1}^{2^n} |k-1\rangle x_k$, where $|a\rangle := |a_1\rangle \otimes |a_2\rangle \otimes \ldots \otimes |a_n\rangle$ and $a := (a_1 a_2 \ldots a_n)_2$ is the binary-digit representation of integer a.

Examples (1)

```
In[1]:= Needs["Q3`"]
```

Basic Examples (1)

```
In[1]:= Let[Qubit, S]
```

Consider a system of n qubits.

```
In[2]:= $n = 3;
        $N = Power[2, $n];
        kk = Range[$n];
        SS = S[kk, $]
```

```
Out[2]= {S1, S2, S3}
```

We want to embed a classical input data of the form.

```
In[3]:= xx = Normalize@RandomVector[$N]
```

```
Out[3]= {-0.259904 - 0.427261 i, 0.444914 - 0.227347 i, -0.140951 + 0.0979235 i, 0.122408 + 0.459967 i,
  -0.116309 + 0.123089 i, -0.020741 + 0.0522983 i, -0.314258 + 0.153372 i, -0.288961 - 0.081422 i}
```

We assume that the data is normalized.

```
In[4]:= Norm[xx]
```

```
Out[4]= 1.
```

This is the desired quantum state embedding the classical data above.

©1988–2023 Wolfram Research, Inc. All rights reserved. http://reference.wolfram.com/language

In[5]:=
```
vec = AmplitudeEmbedding[xx, SS]
```

Out[5]= $(-0.259904 - 0.427261\,i)\,|0_{S_1}0_{S_2}0_{S_3}\rangle + (0.444914 - 0.227347\,i)\,|0_{S_1}0_{S_2}1_{S_3}\rangle - (0.140951 - 0.0979235\,i)\,|0_{S_1}1_{S_2}0_{S_3}\rangle + (0.122408 + 0.459967\,i)\,|0_{S_1}1_{S_2}1_{S_3}\rangle - (0.116309 - 0.123089\,i)\,|1_{S_1}0_{S_2}0_{S_3}\rangle - (0.020741 - 0.0522983\,i)\,|1_{S_1}0_{S_2}1_{S_3}\rangle - (0.314258 - 0.153372\,i)\,|1_{S_1}1_{S_2}0_{S_3}\rangle - (0.288961 + 0.081422\,i)\,|1_{S_1}1_{S_2}1_{S_3}\rangle$

On a quantum machine, the above quantum state must be achieved through a unitary gate starting from the usual initial state $|0\rangle := |0\rangle^{\otimes n}$.

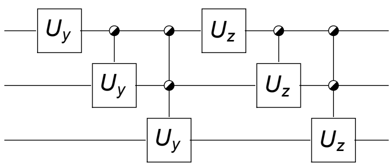
In[6]:=
```
op = AmplitudeEmbeddingGate[xx, SS]
```

Out[6]=
```
AmplitudeEmbeddingGate[{-0.259904 - 0.427261 i, 0.444914 - 0.227347 i,
  -0.140951 + 0.0979235 i, 0.122408 + 0.459967 i, -0.116309 + 0.123089 i,
  -0.020741 + 0.0522983 i, -0.314258 + 0.153372 i, -0.288961 - 0.081422 i}, {S1, S2, S3}]
```

In quantum circuit, it is depicted as follows, which consists of two uniformly controlled rotations.

In[7]:=
```
qc = QuantumCircuit[op]
```

Out[7]=



Obtain the output state from the above unitary gate.

In[8]:=
```
out = qc ** Ket[]
```

Out[8]= $(-0.468873 - 0.173953\,i)\,|0_{S_1}0_{S_2}0_{S_3}\rangle + (0.207993 - 0.454285\,i)\,|0_{S_1}0_{S_2}1_{S_3}\rangle - (0.04983 - 0.164235\,i)\,|0_{S_1}1_{S_2}0_{S_3}\rangle + (0.381304 + 0.284887\,i)\,|0_{S_1}1_{S_2}1_{S_3}\rangle - (0.0148924 - 0.168692\,i)\,|1_{S_1}0_{S_2}0_{S_3}\rangle + (0.0161672 + 0.0538881\,i)\,|1_{S_1}0_{S_2}1_{S_3}\rangle - (0.151385 - 0.31522\,i)\,|1_{S_1}1_{S_2}0_{S_3}\rangle - (0.277167 - 0.115353\,i)\,|1_{S_1}1_{S_2}1_{S_3}\rangle$

Apparently, it looks different from the desired state `vec` shown above. However, this is because of a physically irrelevant global phase. Indeed, the fidelity between the two states are unity.

In[9]:=
```
Fidelity[out, vec]
```

Out[9]= 1.

As already mentioned, the amplitude embedding gate consists of two uniformly controlled rotations. This is made explicitly using the Expand function.

In[10]:=
```
Expand[op]
```

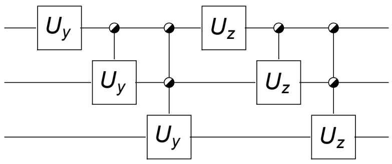
Out[10]=
```
Sequence[Rotation[1.03387, {0, 1, 0}, S1],
 UniformlyControlledRotation[{S1}, {1.24243, 2.40276}, {0, 1, 0}, S2],
 UniformlyControlledRotation[{S1, S2}, {1.56986, 2.44945, 0.641499, 1.41884}, {0, 1, 0}, S3],
 Rotation[0.710362, {0, 0, 1}, S1], UniformlyControlledRotation[{S1}, {3.21741, -2.22781}, {0, 0, 1}, S2],
 UniformlyControlledRotation[{S1, S2}, {1.6449, -1.22371, -0.379524, -5.55449}, {0, 0, 1}, S3]]
```

©1988–2023 Wolfram Research, Inc. All rights reserved. http://reference.wolfram.com/language

In[11]:=

```
qc = QuantumCircuit[Expand@op]
```

Out[11]=

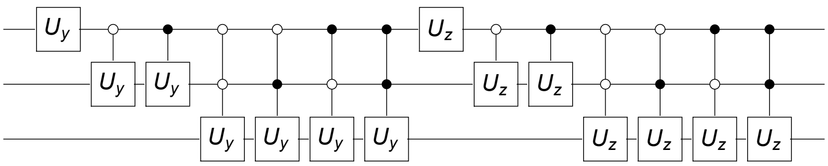


The uniformly controlled rotations themselves may be decomposed further into products of more elementary gates as follows (notice the ExpandAll command).

In[12]:=

```
qc = QuantumCircuit[ExpandAll@op]
```

Out[12]=



Check again the above statement is true by examining the output state.

In[13]:=

```
more = qc ** Ket[SS]
```

Out[13]=

$$(-0.468873 - 0.173953\,i)\,|0_{S_1}0_{S_2}0_{S_3}\rangle + (0.207993 - 0.454285\,i)\,|0_{S_1}0_{S_2}1_{S_3}\rangle - (0.04983 - 0.164235\,i)\,|0_{S_1}1_{S_2}0_{S_3}\rangle + (0.381304 + 0.284887\,i)\,|0_{S_1}1_{S_2}1_{S_3}\rangle - (0.0148924 - 0.168692\,i)\,|1_{S_1}0_{S_2}0_{S_3}\rangle + (0.0161672 + 0.0538881\,i)\,|1_{S_1}0_{S_2}1_{S_3}\rangle - (0.151385 - 0.31522\,i)\,|1_{S_1}1_{S_2}0_{S_3}\rangle - (0.277167 - 0.115353\,i)\,|1_{S_1}1_{S_2}1_{S_3}\rangle$$

In[14]:=

```
Fidelity[more, vec]
```

Out[14]=

```
1.
```

Note that the above quantum circuit may be simplified even further. See the examples in UniformlyControlledRotation.

See Also

AmplitudeEmbedding ▪ UniformlyControlledRotation ▪ BasisEmbedding ▪ BasisEmbeddingGate

Tech Notes

- Multi–Control Unitary Gates
- Quantum Information Systems with Q3
- Quick Quantum Computing with Q3
- Q3: Quick Start

©1988–2023 Wolfram Research, Inc. All rights reserved. http://reference.wolfram.com/language

Related Guides

- Quantum Information Systems
- Q3

Related Links

- M. Möttönen *et al.*, Quantum Information and Computation 5, 467 (2005) , "Transformation of quantum states using uniformly controlled rotations."
- M. Nielsen and I. L. Chuang (2022) , Quantum Computation and Quantum Information (Cambridge University Press, 2011).
- Mahn–Soo Choi (2022) , A Quantum Computation Workbook (Springer, 2022).

©1988–2023 Wolfram Research, Inc. All rights reserved. http://reference.wolfram.com/language

Q3 SYMBOL

See Also Related Guides Tech Notes

URL

Q3`

UniformlyControlledRotation

NEW IN 13.3

UniformlyControlledRotation[$\{c_1, c_2, \ldots, c_n\}$, $\{\theta_1, \theta_2, \ldots, \theta_{2^n}\}$, $\{x, y, z\}$, s]
represents the uniformly controlled rotation on qubit s around axis $\{x, y, z\}$ by angles $\theta_1, \theta_2, \ldots, \theta_{2^n}$ depending on all possible bit sequences of control qubits $c_1, c_2, \ldots, c_n$.

UniformlyControlledRotation[$\{c_1, c_2, \ldots, c_n\}$, $\{\theta_1, \theta_2, \ldots, \theta_n\}$, $s[\ldots, k]$]
uses the k–axis as the rotation axis.

Details and Options

- In general, UniformlyControlledRotation [$\{c_1, c_2, \ldots, c_n\}$, $\{\theta_1, \theta_2, \ldots, \theta_{2^n}\}$, $\{x, y, z\}$, s] is a product of 2^n two-level matrices.
- Note also that UniformlyControlledRotation [$\{c_1, c_2, \ldots, c_n\}$, $\{\theta_1, \theta_2, \ldots, \theta_{2^n}\}$, $\{x, y, z\}$, s] equals to $R_{\mathbf{v}}(\theta_1) \oplus R_{\mathbf{v}}(\theta_2) \oplus \ldots \oplus R_{\mathbf{v}}(\theta_{2^n})$, where $\mathbf{v} \equiv \{x, y, z\}$, and hence is block diagonal.
- An arbitrary multi-qubit unitary matrix U can be decomposed into a product of uniformly controlled rotations; see Möttönen *et al*. (2004).

Examples (4)

```
In[1]:= Needs["Q3`"]
```

Basic Examples (2)

```
In[1]:= Let[Qubit, S, T]
```

```
In[2]:= $n = 3;
        $N = Power[2, $n];
        kk = Range[$n];
        SS = S[kk, $]
Out[2]= {S1, S2, S3}
```

Define a series of rotations on qubit `T[1,$]`.

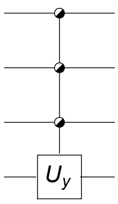
```
In[3]:= aa = RandomReal[{0, 1}, $N] * Pi
Out[3]= {1.81603, 3.04719, 2.86853, 0.160244, 1.0859, 1.78237, 2.22677, 0.516285}
```

```
In[4]:= op = UniformlyControlledRotation[SS, aa, T[2]]
Out[4]= UniformlyControlledRotation[{S1, S2, S3},
         {1.81603, 3.04719, 2.86853, 0.160244, 1.0859, 1.78237, 2.22677, 0.516285}, {0, 1, 0}, T]
```

©1988–2023 Wolfram Research, Inc. All rights reserved. http://reference.wolfram.com/language

In[5]:= **qc = QuantumCircuit[op]**

Out[5]=



In[6]:= **op ** Ket[SS]**

Out[6]= $0.615313\ |0_{S_1}0_{S_2}0_{S_3}0_T\rangle + (0.788283 + 0.\ i)\ |0_{S_1}0_{S_2}0_{S_3}1_T\rangle$

In[7]:= **op ** S[1, 3] // Chop**

Out[7]=

$$-0.137222 - 0.049277\, S_1^Z S_2^Z - 0.0737621\, S_1^Z S_3^Z - (0. + 0.663108\, i)\, S_1^Z T^Y + 0.0844846\, S_2^Z S_3^Z - (0. + 0.071805\, i)\, S_2^Z T^Y - (0. + 0.0399121\, i)\, S_3^Z T^Y + 0.272719\, S_1^Z S_2^Z S_3^Z - (0. + 0.107305\, i)\, S_1^Z S_2^Z T^Y - (0. + 0.135102\, i)\, S_1^Z S_3^Z T^Y + (0. + 0.0272742\, i)\, S_2^Z S_3^Z T^Y + (0. + 0.253041\, i)\, S_1^Z S_2^Z S_3^Z T^Y + 0.586071\, S_1^Z - 0.068323\, S_2^Z + 0.000622466\, S_3^Z - (0. + 0.0513669\, i)\, T^Y$$

In[8]:= **Elaborate[op] // Chop**

Out[8]=

$$0.586071 - 0.068323\, S_1^Z S_2^Z + 0.000622466\, S_1^Z S_3^Z - (0. + 0.0513669\, i)\, S_1^Z T^Y + 0.272719\, S_2^Z S_3^Z - (0. + 0.107305\, i)\, S_2^Z T^Y - (0. + 0.135102\, i)\, S_3^Z T^Y + 0.0844846\, S_1^Z S_2^Z S_3^Z - (0. + 0.071805\, i)\, S_1^Z S_2^Z T^Y - (0. + 0.0399121\, i)\, S_1^Z S_3^Z T^Y + (0. + 0.253041\, i)\, S_2^Z S_3^Z T^Y + (0. + 0.0272742\, i)\, S_1^Z S_2^Z S_3^Z T^Y - 0.137222\, S_1^Z - 0.049277\, S_2^Z - 0.0737621\, S_3^Z - (0. + 0.663108\, i)\, T^Y$$

In[9]:= **Matrix[op] // Chop // MatrixForm**

Out[9]//MatrixForm=

0.615313	-0.788283	0	0	0	0	0	0	0	0	
0.788283	0.615313	0	0	0	0	0	0	0	0	
0	0	0.0471855	-0.998886	0	0	0	0	0	0	
0	0	0.998886	0.0471855	0	0	0	0	0	0	
0	0	0	0	0.136106	-0.990694	0	0	0	0	
0	0	0	0	0.990694	0.136106	0	0	0	0	
0	0	0	0	0	0	0.996792	-0.0800364	0	0	
0	0	0	0	0	0	0.0800364	0.996792	0	0	
0	0	0	0	0	0	0	0	0.856189	-0.516664	
0	0	0	0	0	0	0	0	0.516664	0.856189	
0	0	0	0	0	0	0	0	0	0	0.
0	0	0	0	0	0	0	0	0	0	0.
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	

In[10]:=

Expand[op]

Out[10]=

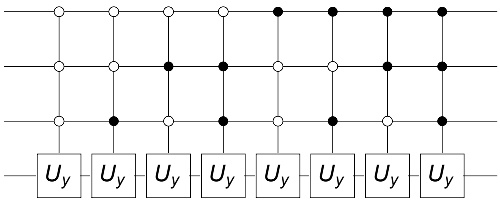
Sequence[ControlledGate[{S_1, S_2, S_3} → {0, 0, 0}, Rotation[1.81603, {0, 1, 0}, T]],
ControlledGate[{S_1, S_2, S_3} → {0, 0, 1}, Rotation[3.04719, {0, 1, 0}, T]],
ControlledGate[{S_1, S_2, S_3} → {0, 1, 0}, Rotation[2.86853, {0, 1, 0}, T]],
ControlledGate[{S_1, S_2, S_3} → {0, 1, 1}, Rotation[0.160244, {0, 1, 0}, T]],
ControlledGate[{S_1, S_2, S_3} → {1, 0, 0}, Rotation[1.0859, {0, 1, 0}, T]],
ControlledGate[{S_1, S_2, S_3} → {1, 0, 1}, Rotation[1.78237, {0, 1, 0}, T]],
ControlledGate[{S_1, S_2, S_3} → {1, 1, 0}, Rotation[2.22677, {0, 1, 0}, T]],
ControlledGate[{S_1, S_2, S_3} → {1, 1, 1}, Rotation[0.516285, {0, 1, 0}, T]]]

©1988–2023 Wolfram Research, Inc. All rights reserved. http://reference.wolfram.com/language

In[11]:=

```
qc = QuantumCircuit[Expand@op]
```

Out[11]=



For more examples, see the Scope section below.

Scope (2)

Dagger (1)

In[1]:=

```
Let[Qubit, S, T]
```

In[2]:=

```
$n = 3;
$N = Power[2, $n];
kk = Range[$n];
SS = S[kk, $]
```

Out[2]= $\{S_1, S_2, S_3\}$

In[3]:=

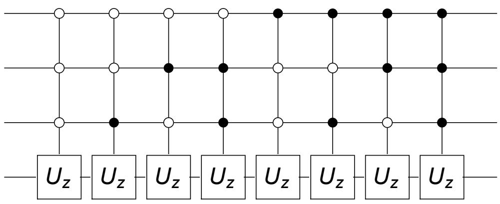
```
aa = RandomReal[{0, 1}, $N] * Pi
```

Out[3]= {2.36348, 2.92257, 1.13025, 2.12854, 1.40961, 1.17361, 2.45454, 1.58282}

In[4]:=

```
op = UniformlyControlledRotation[SS, aa, T[3]];
qc = QuantumCircuit[Expand@op]
```

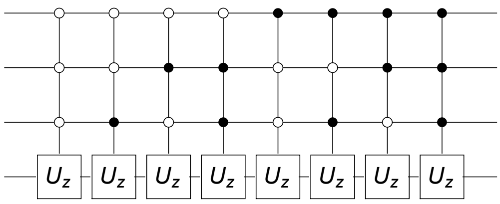
Out[4]=



In[5]:=

```
new = QuantumCircuit[Expand@Dagger@op]
```

Out[5]=



©1988–2023 Wolfram Research, Inc. All rights reserved. http://reference.wolfram.com/language

In[6]:=
```
Matrix[qc].Matrix[new] // Chop // MatrixForm
```

Out[6]//MatrixForm=

1.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1.	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1.	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1.	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1.	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1.	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1.	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1.	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1.	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1.	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1.	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1.	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1.	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1.	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.

Simplification (1)

As long as the rotation axis is *not* parallel to the x-axis, uniformly controlled rotation may be reduced to a simpler and more efficient gate sequence.

In[1]:=
```
Let[Qubit, S, T]
```

In[2]:=
```
$n = 3;
$N = Power[2, $n];
kk = Range[$n];
SS = S[kk, $]
```

Out[2]= $\{S_1, S_2, S_3\}$

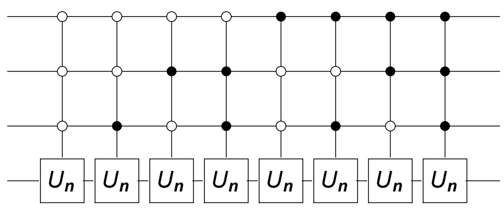
In[3]:=
```
aa = RandomReal[{0, 2}, $N] * Pi
```

Out[3]= {5.77497, 0.816812, 3.64532, 1.99685, 1.32353, 5.2546, 5.48638, 3.91312}

Note that the rotation axis is not parallel to the x-axis.

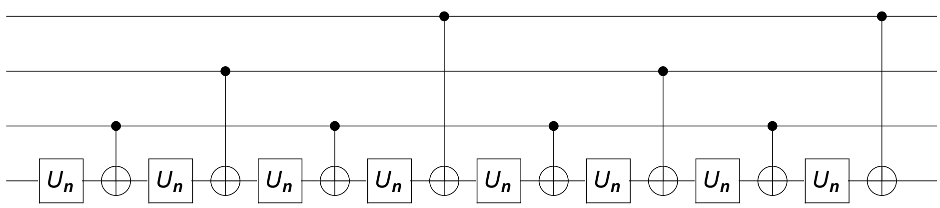
In[4]:=
```
op = UniformlyControlledRotation[SS, aa, {0, 1, 1}, T];
qc = QuantumCircuit[Expand@op]
```

Out[4]=



In[5]:=
```
new = QuantumCircuit[GateFactor@op]
```

Out[5]=



©1988–2023 Wolfram Research, Inc. All rights reserved. http://reference.wolfram.com/language

In[6]:=
```
Matrix[new] - Matrix[qc] // Chop // MatrixForm
```

Out[6]//MatrixForm=

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

See Also

ControlledGate ▪ ControlledPower ▪ CNOT ▪ CZ ▪ AmplitudeEmbeddingGate

Tech Notes

- Multi-Control Unitary Gates
- Quantum Information Systems with Q3
- Quick Quantum Computing with Q3
- Q3: Quick Start

Related Guides

- Quantum Information Systems
- Q3

Related Links

- M. Möttönen *et al.*, Physical Review Letters 93, 130502 (2004) , "Quantum Circuits for General Multiqubit Gates."
- M. Nielsen and I. L. Chuang (2022) , Quantum Computation and Quantum Information (Cambridge University Press, 2011).
- Mahn-Soo Choi (2022) , A Quantum Computation Workbook (Springer, 2022).

©1988–2023 Wolfram Research, Inc. All rights reserved. http://reference.wolfram.com/language