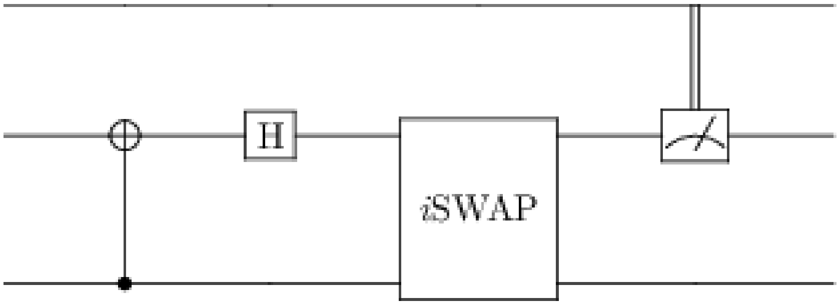
Testing MatRenderer: Matplotlib Based Circuit Renderer

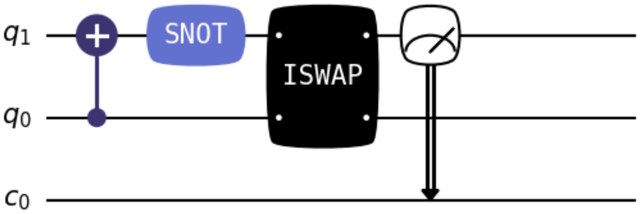
```
from qutip_qip.circuit import QubitCircuit, MatRenderer
import numpy as np
```

Current LaTeX

```
qc1 = QubitCircuit(2, num_cbits=1)
qc1.add_gate("CNOT", controls=0, targets=1)
qc1.add_gate("SNOT", targets=1)
qc1.add_gate("ISWAP", targets=[0,1])
qc1.add_measurement("M0", targets=1, classical_store=0)
qc1
```

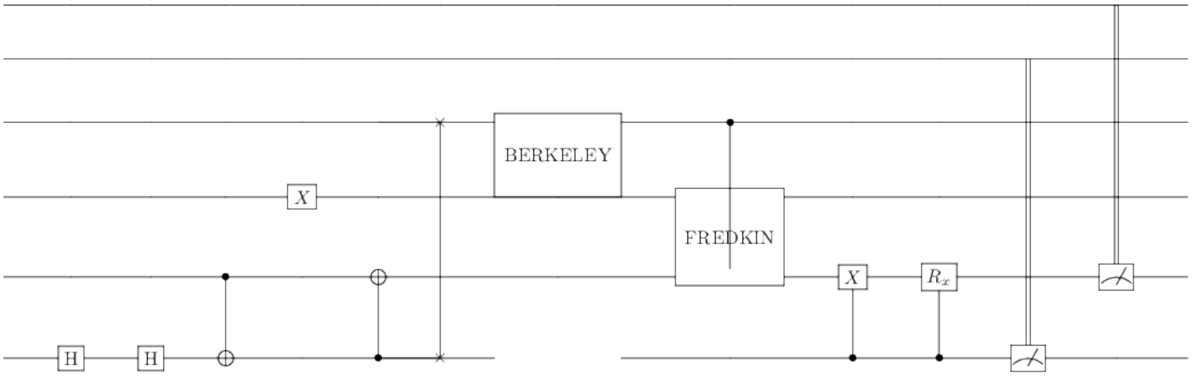


```
MatRenderer(qc1)
```

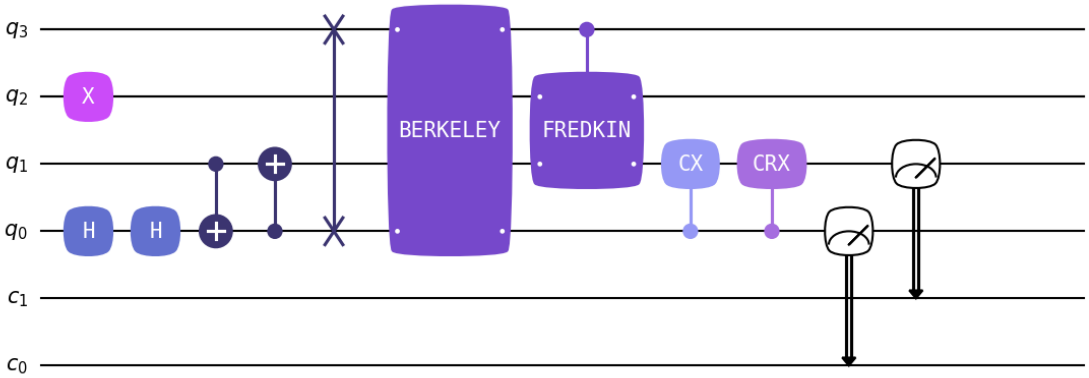


Some bugs with current LaTeX implementation - Multi-Qubit Gates Rendering over non-connected wires (eg. BERKELEY in circuit below)

```
qc = QubitCircuit(4, num_cbits=2)
qc.add_gate("H", targets=[0])
qc.add_gate("H", targets=[0])
qc.add_gate("CNOT", controls=[1], targets=[0])
qc.add_gate("X", targets=[2])
qc.add_gate("CNOT", controls=[0], targets=[1])
qc.add_gate("SWAP", targets=[0, 3])
qc.add_gate("BERKELEY", targets=[0, 3])
qc.add_gate("FREDKIN", controls=[3], targets=[1, 2])
qc.add_gate("CX", controls=[0], targets=[1])
qc.add_gate("CRX", controls=[0], targets=[1], arg_value=0.5)
qc.add_measurement("M", targets=[0], classical_store=0)
qc.add_measurement("M", targets=[1], classical_store=1)
qc
```

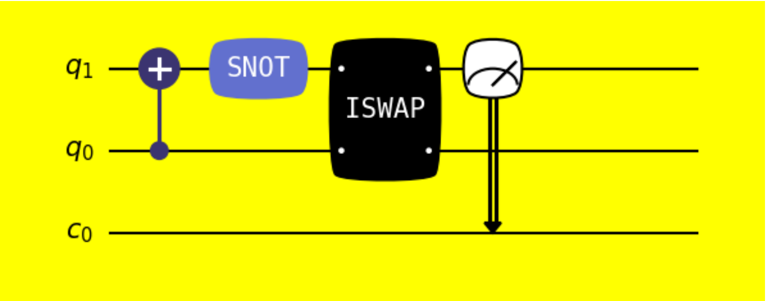


```
MatRenderer(qc)
```

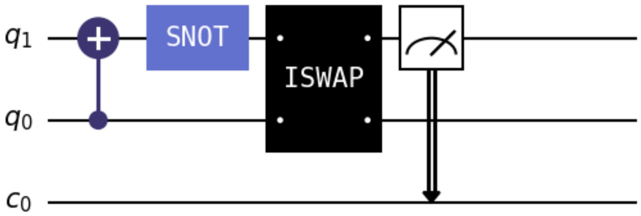


Circuit Layout Level Customization Options

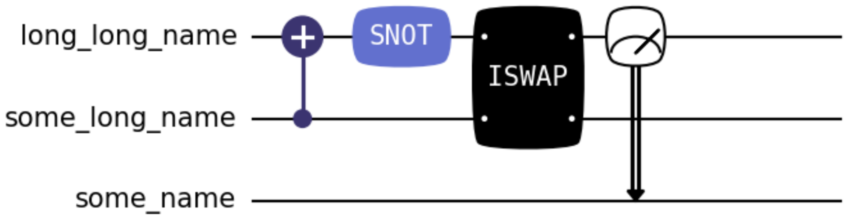
```
MatRenderer(qc1, bgcolor="yellow")
```



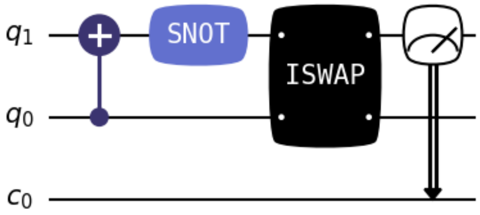
```
MatRenderer(qc1, bulge=False)
```



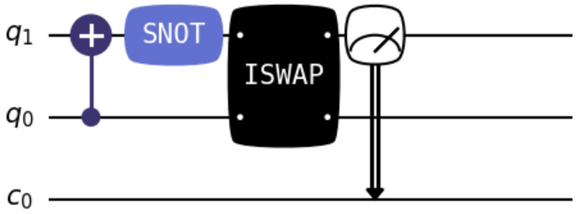
```
MatRenderer(qc1, wire_label=["some_name", "some_long_name", "long_long_name"])
```



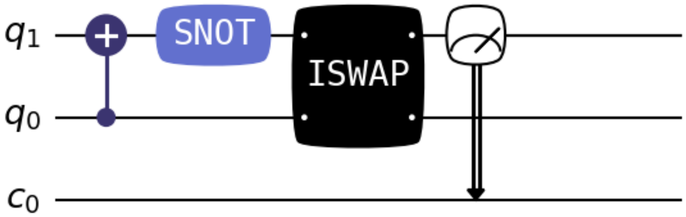
```
MatRenderer(qc1, end_wire_ext=0)
```



```
MatRenderer(qc1, condense=0.1)
```



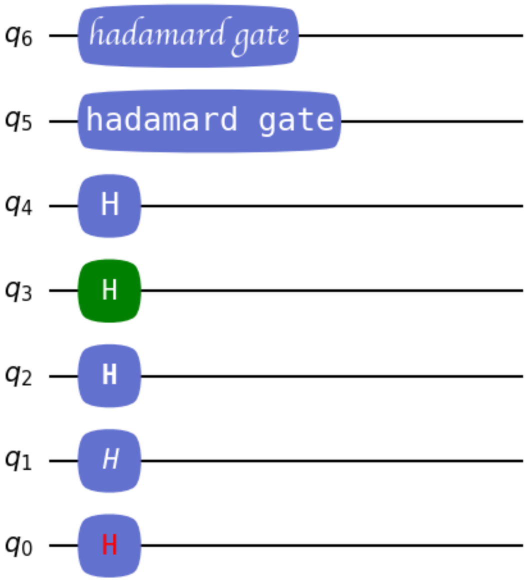
```
MatRenderer(qc1, fontsize=13)
```



Gate Level Customization Options

```
qc = QubitCircuit(7)
qc.add_gate("H", targets=[0], style={'fontcolor': 'red'})
qc.add_gate("H", targets=[1], style={'fontstyle': 'italic'})
qc.add_gate("H", targets=[2], style={'fontweight': 'bold'})
qc.add_gate("H", targets=[3], style={'color': 'green'})
qc.add_gate("H", targets=[4], style={'fontsize': 12})
qc.add_gate("H", targets=[5], style={'text': 'hadamard gate'})
qc.add_gate("H", targets=[6], style={'text': 'hadamard gate', 'fontfamily': 'cursive'})
```

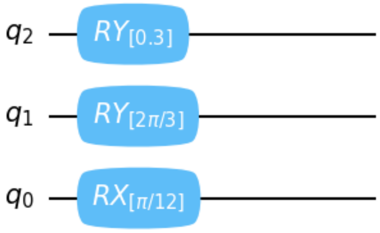
```
MatRenderer(qc)
```



Argument Value

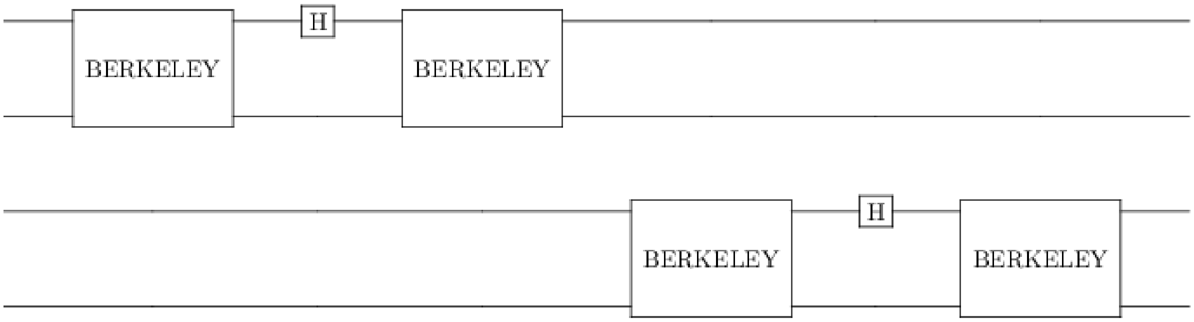
```
qc = QubitCircuit(3)
qc.add_gate("RX", targets=[0], arg_value=np.pi/12, style={'showarg': True})
qc.add_gate("RY", targets=[1], arg_value=2*np.pi/3, style={'showarg': True})
qc.add_gate("RY", targets=[2], arg_value=0.3, style={'showarg': True})
```

```
MatRenderer(qc)
```

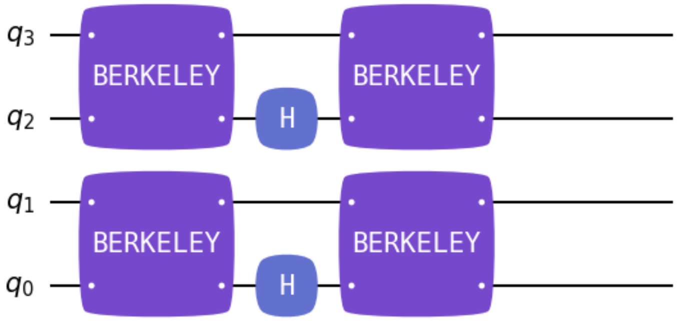


Some more reandom circuit for testing

```
qc = QubitCircuit(4, reverse_states=False)
qc.add_gate("BERKELEY", targets=[0, 1])
qc.add_gate("H", targets=[0])
qc.add_gate("BERKELEY", targets=[0, 1])
qc.add_gate("BERKELEY", targets=[2, 3])
qc.add_gate("H", targets=[2])
qc.add_gate("BERKELEY", targets=[2, 3])
qc
```

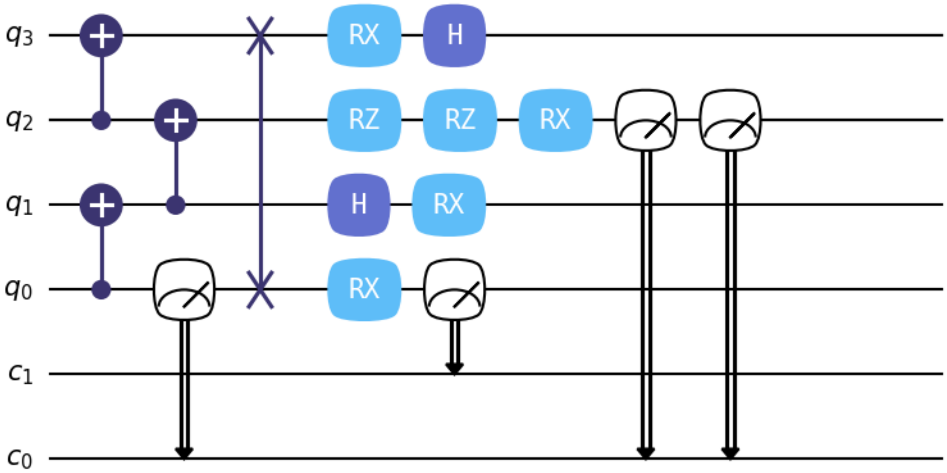


```
MatRenderer(qc)
```



```
qc = QubitCircuit(4, num_cbits=2)
qc.add_gate("CNOT", controls=[0], targets=[1])
qc.add_gate("CNOT", controls=[2], targets=[3])
qc.add_gate("CNOT", controls=[1], targets=[2])
qc.add_measurement("M", targets=[0], classical_store=0)
qc.add_gate("SWAP", targets=[0, 3])
qc.add_gate("H", targets=[1])
qc.add_gate("RZ", targets=[2], arg_value=0.5)
qc.add_gate("RZ", targets=[2], arg_value=0.5)
qc.add_1q_gate("RX")
qc.add_measurement("M", targets=[0], classical_store=1)
qc.add_measurement("M", targets=[2], classical_store=0)
qc.add_measurement("M", targets=[2], classical_store=0)
qc.add_gate("H", targets=[3])
```

```
MatRenderer(qc)
```



With User Custom Gates

```
from qutip_qip.operations import Gate, rx
import numpy as np
from qutip import Qobj
```

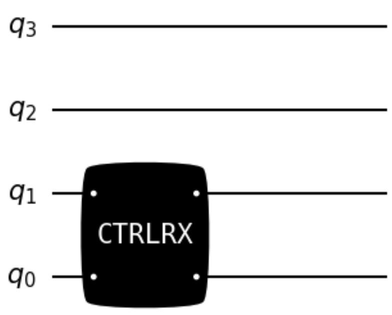
```
def user_gate1(arg_value):
    # controlled rotation X
    mat = np.zeros((4, 4), dtype=np.complex)
    mat[0, 0] = mat[1, 1] = 1.
    mat[2:4, 2:4] = rx(arg_value).full()
    return Qobj(mat, dims=[[2, 2], [2, 2]])


def user_gate2():
    # S gate
    mat = np.array([[1.,   0],
                    [0., 1.j]])
    return Qobj(mat, dims=[[2], [2]])
```

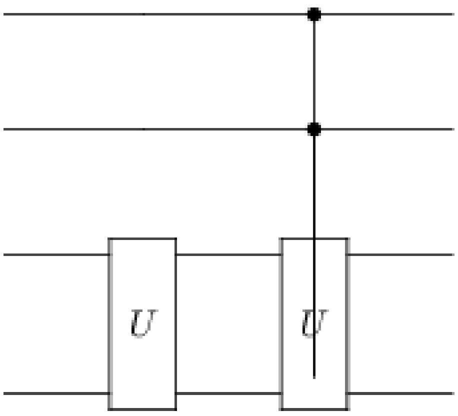
```
qc = QubitCircuit(4)
qc.user_gates = {"CTRLRX": user_gate1,
                 "S"     : user_gate2}
```

```
# qubit 1 controls qubit 0
qc.add_gate("CTRLRX", targets=[1,0], arg_value=np.pi/2)
```

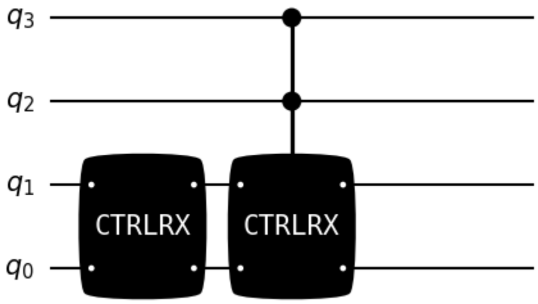
```
MatRenderer(qc)
```



```
# qubit 0 controls qubit 1
qc.add_gate("CTRLRX", targets=[0,1], controls=[2,3], arg_value=np.pi/2)
qc
```



```
MatRenderer(qc)
```


q_3
q_2
q_1
q_0
CTRLRX
CTRLRX