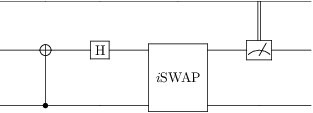
Testing MatRenderer: Matplotlib Based Circuit Renderer

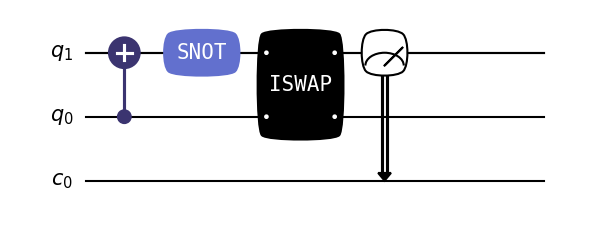
from qutip\_qip.circuit import QubitCircuit, MatRenderer  
import numpy as np

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

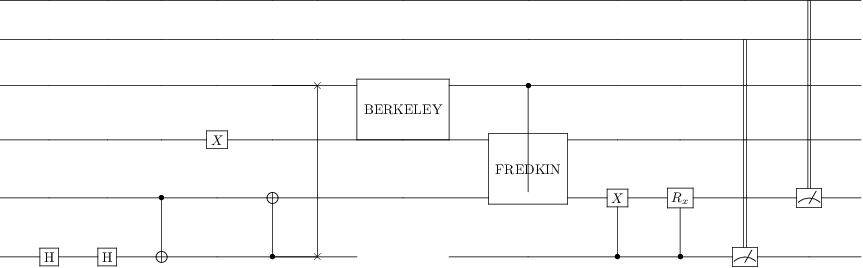


MatRenderer(qc1)

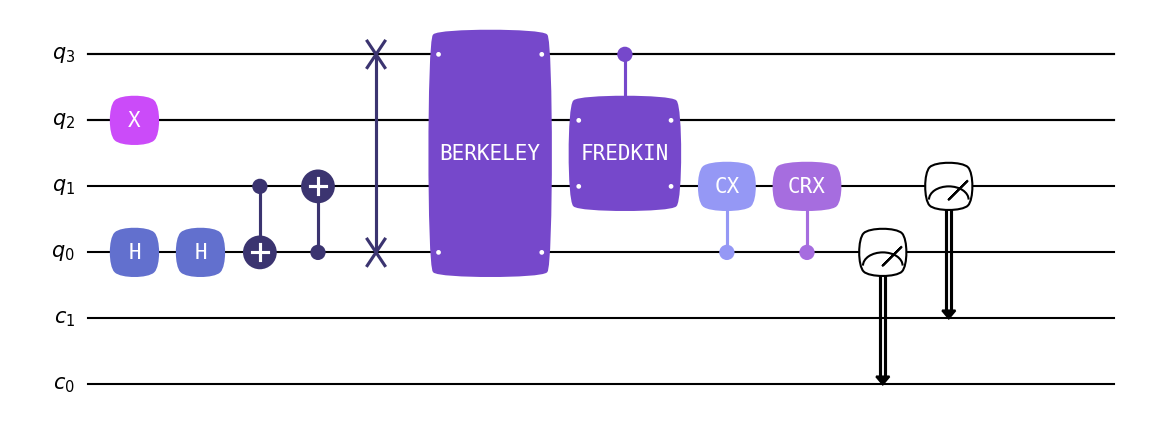


Some bugs with current LaTeX implementation - Multi-Qubit Gates Rendering over non-connected wires (eg. BERKELEY in circuit below)

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

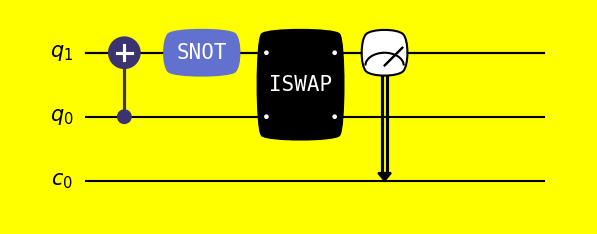


MatRenderer(qc)

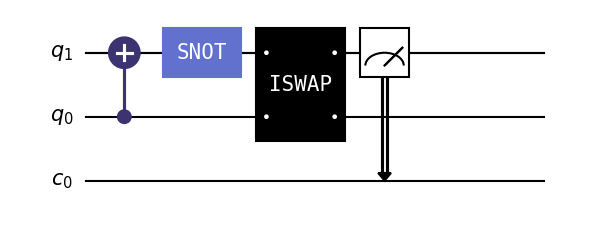


#### Circuit Layout Level Customization Options

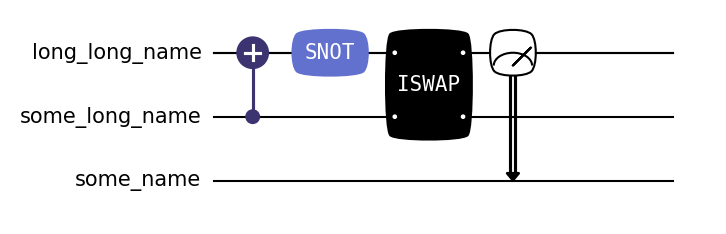
MatRenderer(qc1, bgcolor="yellow")



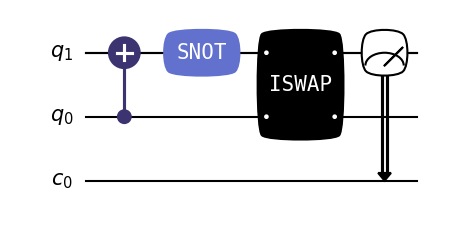
MatRenderer(qc1, bulge=False)



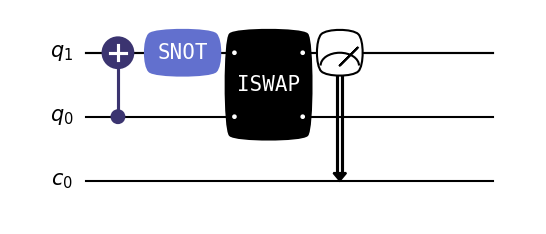
MatRenderer(qc1, wire\_label=["some\_name", "some\_long\_name", "long\_long\_name"])



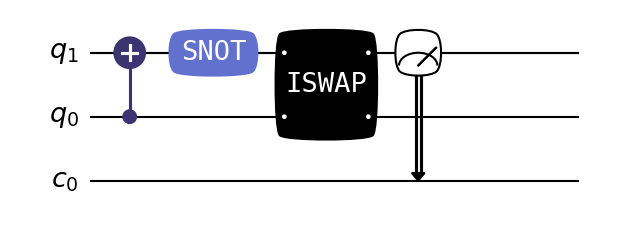
MatRenderer(qc1, end\_wire\_ext=0)



MatRenderer(qc1, condense=0.1)



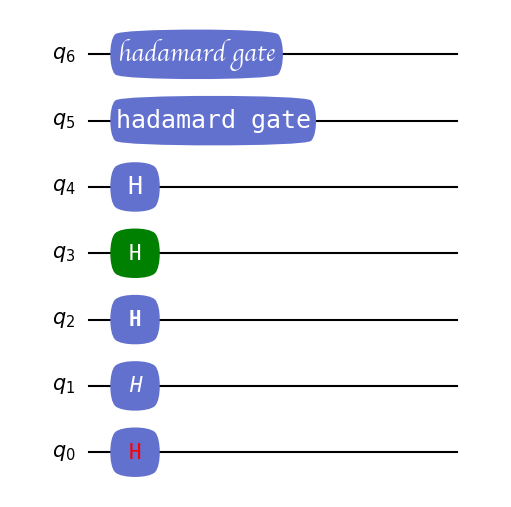
MatRenderer(qc1, fontsize=13)



#### 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'})

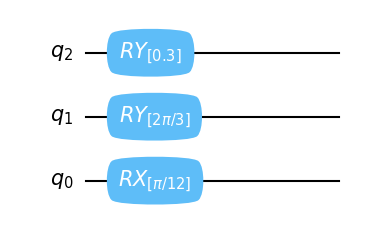
MatRenderer(qc)



#### Argument Value

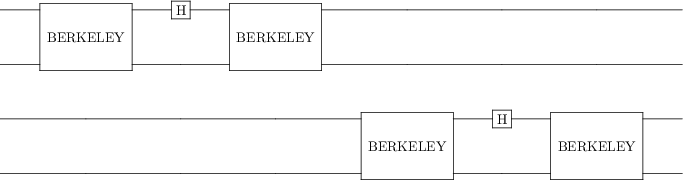
qc = QubitCircuit(3)  
qc.add\_gate("RX", targets=[0], arg\_value=np.pi/12, style={'showarg': True})  
qc.add\_gate("RY", targets=[1], arg\_value=2\*np.pi/3, style={'showarg': True})  
qc.add\_gate("RY", targets=[2], arg\_value=0.3, style={'showarg': True})

MatRenderer(qc)

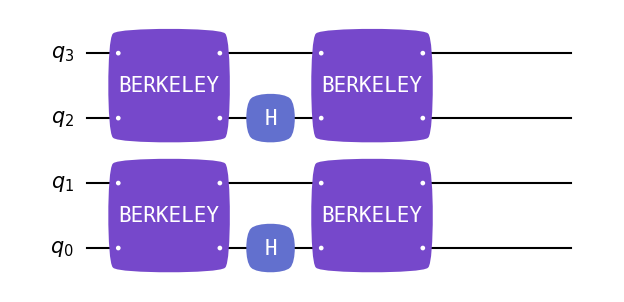


#### Some more reandom circuit for testing

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

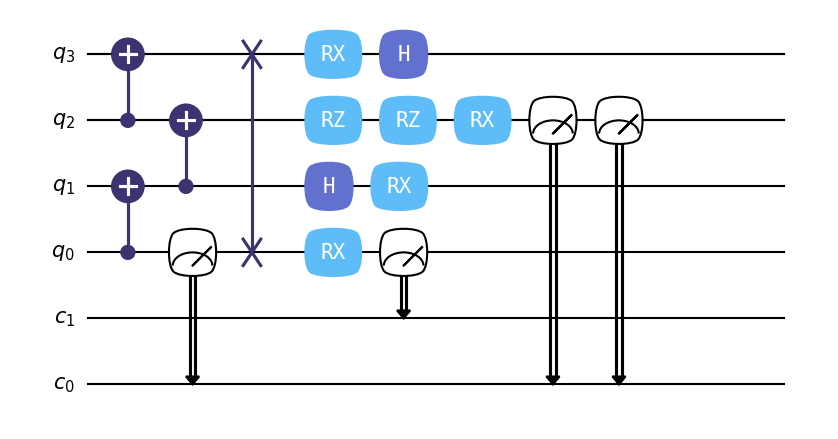


MatRenderer(qc)



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)



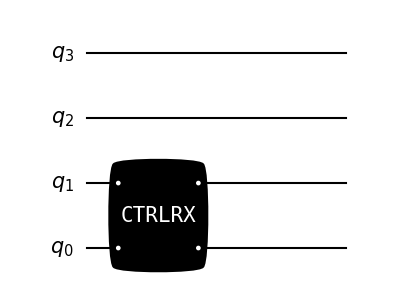
#### With User Custom Gates

from qutip\_qip.operations import Gate, rx  
import numpy as np  
from qutip import Qobj

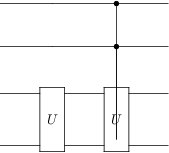
def user\_gate1(arg\_value):  
 # controlled rotation X  
 mat = np.zeros((4, 4), dtype=np.complex)  
 mat[0, 0] = mat[1, 1] = 1.  
 mat[2:4, 2:4] = rx(arg\_value).full()  
 return Qobj(mat, dims=[[2, 2], [2, 2]])  
  
  
def user\_gate2():  
 # S gate  
 mat = np.array([[1., 0],  
 [0., 1.j]])  
 return 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)

