

```
!pip install cirq numpy
```

```

Downloading cirq_google-1.4.1-py3-none-any.whl (532 kB)
532.7/532.7 kB 27.9 MB/s eta 0:00:00
Downloading cirq_ionq-1.4.1-py3-none-any.whl (60 kB)
60.5/60.5 kB 3.6 MB/s eta 0:00:00
Downloading cirq_pasqal-1.4.1-py3-none-any.whl (31 kB)
Downloading cirq_rigetti-1.4.1-py3-none-any.whl (69 kB)
69.3/69.3 kB 5.6 MB/s eta 0:00:00
Downloading cirq_web-1.4.1-py3-none-any.whl (596 kB)
596.5/596.5 kB 34.0 MB/s eta 0:00:00
Downloading numpy-1.26.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (18.3 MB)
18.3/18.3 MB 52.7 MB/s eta 0:00:00
Downloading duet-0.2.9-py3-none-any.whl (29 kB)
Downloading protobuf-4.25.6-cp37-abi3-manylinux2014_x86_64.whl (294 kB)
294.6/294.6 kB 20.3 MB/s eta 0:00:00
Downloading pyquil-4.16.1-py3-none-any.whl (203 kB)
203.2/203.2 kB 13.7 MB/s eta 0:00:00
Downloading grpcio_status-1.62.3-py3-none-any.whl (14 kB)
Downloading packaging-23.2-py3-none-any.whl (53 kB)
53.0/53.0 kB 4.2 MB/s eta 0:00:00
Downloading qcs_sdk_python-0.21.18-cp311-cp311-manylinux_2_28_x86_64.whl (6.7 MB)
6.7/6.7 MB 79.7 MB/s eta 0:00:00
Downloading quil-0.16.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.3 MB)
2.3/2.3 MB 55.4 MB/s eta 0:00:00
Downloading types_deprecated-1.2.15.20250304-py3-none-any.whl (8.6 kB)
Downloading qcs_api_client_common-0.11.8-cp311-cp311-manylinux_2_28_x86_64.whl (2.7 MB)
2.7/2.7 MB 67.3 MB/s eta 0:00:00
Downloading python_rapidjson-1.20-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.7 MB)
1.7/1.7 MB 56.5 MB/s eta 0:00:00
Downloading ruamel.yaml-0.18.10-py3-none-any.whl (117 kB)
117.7/117.7 kB 9.1 MB/s eta 0:00:00
Downloading ruamel.yaml.clib-0.2.12-cp311-cp311-manylinux2014_x86_64.whl (739 kB)
739.1/739.1 kB 31.9 MB/s eta 0:00:00
Building wheels for collected packages: rpcq
Building wheel for rpcq (setup.py) ... done
Created wheel for rpcq: filename=rpcq-3.11.0-py3-none-any.whl size=45969 sha256=75bebce0d52dd2cdd4c8ecb563f96ddef5bfff72019c
Stored in directory: /root/.cache/pip/wheels/b3/b6/ad/8f05ac735661e828024c764a6e6fc3eb6f84434e408aad5eab
Successfully built rpcq
Installing collected packages: types-deprecated, ruamel.yaml.clib, python-rapidjson, protobuf, packaging, numpy, duet, ruamel
Attempting uninstall: protobuf
Found existing installation: protobuf 5.29.4
Uninstalling protobuf-5.29.4:
Successfully uninstalled protobuf-5.29.4
Attempting uninstall: packaging
Found existing installation: packaging 24.2
Uninstalling packaging-24.2:
Successfully uninstalled packaging-24.2
Attempting uninstall: numpy
Found existing installation: numpy 2.0.2
Uninstalling numpy-2.0.2:
Successfully uninstalled numpy-2.0.2
Attempting uninstall: grpcio-status
Found existing installation: grpcio-status 1.71.0
Uninstalling grpcio-status-1.71.0:
Successfully uninstalled grpcio-status-1.71.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is
google-cloud-bigquery 3.31.0 requires packaging>=24.2.0, but you have packaging 23.2 which is incompatible.
Successfully installed cirq-1.4.1 cirq-aqt-1.4.1 cirq-core-1.4.1 cirq-google-1.4.1 cirq-ionq-1.4.1 cirq-pasqal-1.4.1 cirq-rig

```

```

import cirq
import numpy as np
import matplotlib.pyplot as plt

# Define 5 qubits
qubits = [cirq.LineQubit(i) for i in range(5)]

# Create a circuit
circuit = cirq.Circuit()

# Apply Hadamard gate to every qubit
circuit.append(cirq.H.on_each(*qubits))

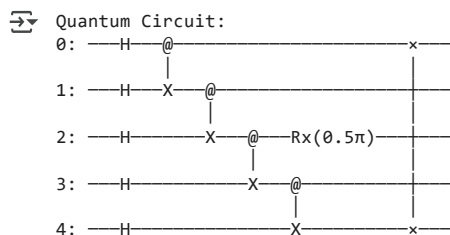
# Apply CNOT gates
circuit.append([
    cirq.CNOT(qubits[0], qubits[1]),
    cirq.CNOT(qubits[1], qubits[2]),
    cirq.CNOT(qubits[2], qubits[3]),
    cirq.CNOT(qubits[3], qubits[4])
])

# Apply SWAP between qubits 0 and 4
circuit.append(cirq.SWAP(qubits[0], qubits[4]))

```

```
# Apply Rotate-X ( $\pi/2$ ) on any qubit (choosing qubit 2)
circuit.append(cirq.rx(np.pi/2)(qubits[2]))
```

```
# Print and visualize the circuit
print("Quantum Circuit:")
print(circuit)
```



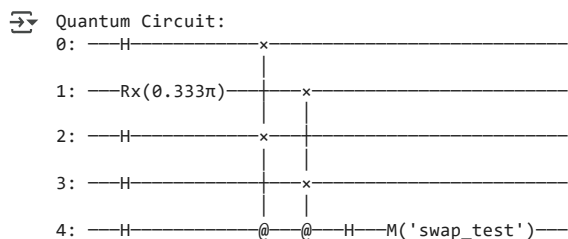
```
qubits = [cirq.LineQubit(i) for i in range(5)]
```

```
circuit = cirq.Circuit()
circuit.append(cirq.H(qubits[0]))
circuit.append(cirq.rx(np.pi/3)(qubits[1]))
circuit.append(cirq.H(qubits[2]))
circuit.append(cirq.H(qubits[3]))
```

```
circuit.append(cirq.H(qubits[4]))
circuit.append(cirq.CSWAP(qubits[4], qubits[0], qubits[2]))
circuit.append(cirq.CSWAP(qubits[4], qubits[1], qubits[3]))
circuit.append(cirq.H(qubits[4]))
```

```
circuit.append(cirq.measure(qubits[4], key='swap_test'))
print("Quantum Circuit:")
print(circuit)
```

```
simulator = cirq.Simulator()
result = simulator.run(circuit, repetitions=1000)
print("\nSwap Test Results:")
print(result.histogram(key='swap_test'))
```



```
Swap Test Results:
Counter({0: 746, 1: 254})
```