T2 - Entanglement - Superdense Coding - ParsaVARES

November 1, 2024

1 Entanglement in Action

2 Superdense Coding

[2]:
```
%pip install qiskit[visualization]
```

```
Requirement already satisfied: qiskit[visualization] in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages
(1.2.4)
Requirement already satisfied: rustworkx>=0.15.0 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit[visualization]) (0.15.1)
Requirement already satisfied: numpy<3,>=1.17 in /opt/conda/lib/python3.11/site-
packages (from qiskit[visualization]) (1.26.4)
Requirement already satisfied: scipy>=1.5 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit[visualization]) (1.14.1)
Requirement already satisfied: sympy>=1.3 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit[visualization]) (1.13.3)
Requirement already satisfied: dill>=0.3 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit[visualization]) (0.3.9)
Requirement already satisfied: python-dateutil>=2.8.0 in
/opt/conda/lib/python3.11/site-packages (from qiskit[visualization]) (2.9.0)
Requirement already satisfied: stevedore>=3.0.0 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit[visualization]) (5.3.0)
Requirement already satisfied: typing-extensions in
/opt/conda/lib/python3.11/site-packages (from qiskit[visualization]) (4.11.0)
Requirement already satisfied: symengine<0.14,>=0.11 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit[visualization]) (0.13.0)
Requirement already satisfied: matplotlib>=3.3 in
/opt/conda/lib/python3.11/site-packages (from qiskit[visualization]) (3.9.2)
Requirement already satisfied: pydot in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
```

```
qiskit[visualization]) (3.0.2)
Requirement already satisfied: Pillow>=4.2.1 in /opt/conda/lib/python3.11/site-
packages (from qiskit[visualization]) (11.0.0)
Requirement already satisfied: pylatexenc>=1.4 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit[visualization]) (2.10)
Requirement already satisfied: seaborn>=0.9.0 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit[visualization]) (0.13.2)
Requirement already satisfied: contourpy>=1.0.1 in
/opt/conda/lib/python3.11/site-packages (from
matplotlib>=3.3->qiskit[visualization]) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.11/site-
packages (from matplotlib>=3.3->qiskit[visualization]) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/conda/lib/python3.11/site-packages (from
matplotlib>=3.3->qiskit[visualization]) (4.54.1)
Requirement already satisfied: kiwisolver>=1.3.1 in
/opt/conda/lib/python3.11/site-packages (from
matplotlib>=3.3->qiskit[visualization]) (1.4.7)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/lib/python3.11/site-packages (from
matplotlib>=3.3->qiskit[visualization]) (24.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/opt/conda/lib/python3.11/site-packages (from
matplotlib>=3.3->qiskit[visualization]) (3.2.0)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.11/site-
packages (from python-dateutil>=2.8.0->qiskit[visualization]) (1.16.0)
Requirement already satisfied: pandas>=1.2 in /opt/conda/lib/python3.11/site-
packages (from seaborn>=0.9.0->qiskit[visualization]) (2.2.3)
Requirement already satisfied: pbr>=2.0.0 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
stevedore>=3.0.0->qiskit[visualization]) (6.1.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
sympy>=1.3->qiskit[visualization]) (1.3.0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.11/site-
packages (from pandas>=1.2->seaborn>=0.9.0->qiskit[visualization]) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.11/site-
packages (from pandas>=1.2->seaborn>=0.9.0->qiskit[visualization]) (2024.2)
Note: you may need to restart the kernel to use updated packages.
```

[3]:
```
%pip install qiskit_aer
```

```
Requirement already satisfied: qiskit_aer in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages
(0.15.1)
Requirement already satisfied: qiskit>=1.1.0 in
```

```
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit_aer) (1.2.4)
Requirement already satisfied: numpy>=1.16.3 in /opt/conda/lib/python3.11/site-
packages (from qiskit_aer) (1.26.4)
Requirement already satisfied: scipy>=1.0 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit_aer) (1.14.1)
Requirement already satisfied: psutil>=5 in /opt/conda/lib/python3.11/site-
packages (from qiskit_aer) (5.9.8)
Requirement already satisfied: rustworkx>=0.15.0 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit>=1.1.0->qiskit_aer) (0.15.1)
Requirement already satisfied: sympy>=1.3 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit>=1.1.0->qiskit_aer) (1.13.3)
Requirement already satisfied: dill>=0.3 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit>=1.1.0->qiskit_aer) (0.3.9)
Requirement already satisfied: python-dateutil>=2.8.0 in
/opt/conda/lib/python3.11/site-packages (from qiskit>=1.1.0->qiskit_aer) (2.9.0)
Requirement already satisfied: stevedore>=3.0.0 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit>=1.1.0->qiskit_aer) (5.3.0)
Requirement already satisfied: typing-extensions in
/opt/conda/lib/python3.11/site-packages (from qiskit>=1.1.0->qiskit_aer)
(4.11.0)
Requirement already satisfied: symengine<0.14,>=0.11 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
qiskit>=1.1.0->qiskit_aer) (0.13.0)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.11/site-
packages (from python-dateutil>=2.8.0->qiskit>=1.1.0->qiskit_aer) (1.16.0)
Requirement already satisfied: pbr>=2.0.0 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
stevedore>=3.0.0->qiskit>=1.1.0->qiskit_aer) (6.1.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/opt/.qbraid/environments/qbraid_000000/pyenv/lib/python3.11/site-packages (from
sympy>=1.3->qiskit>=1.1.0->qiskit_aer) (1.3.0)
Note: you may need to restart the kernel to use updated packages.
```

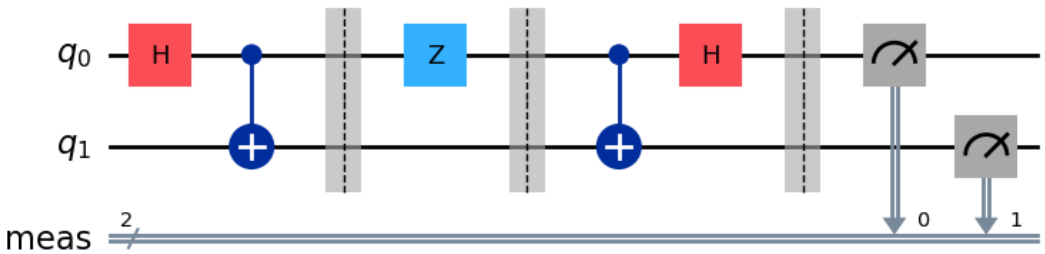
[4]:
```
# Required imports
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister
from qiskit_aer.primitives import Sampler
from qiskit_aer import AerSimulator
from qiskit.visualization import plot_histogram
```

Here is a simple implementation of superdense coding where we specify the circuit itself depending on the bits to be transmitted. First let's specify the bits to be transmitted. (Try changing the bits to see that it works correctly.)

[5]:
```
c = "1"
d = "0"
```

Now we'll build the circuit accordingly. Here we'll just allow Qiskit to use the default names for the qubits: q0 for the top qubit and q1 for the bottom one.

[6]:
```
protocol = QuantumCircuit(2)

# Prepare ebit used for superdense coding
# Replace the ?
protocol.h(0)           # Apply Hadamard gate on the first qubit (q0)
protocol.cx(0, 1)       # Apply CNOT gate with q0 as control and q1 as target
protocol.barrier()

# Alice's operations
if d == "1":
    protocol.x(0)       # Apply X gate if d is 1
if c == "1":
    protocol.z(0)       # Apply Z gate if c is 1
protocol.barrier()

# Bob's actions
protocol.cx(0, 1)       # Apply CNOT with q0 as control and q1 as target
protocol.h(0)           # Apply Hadamard on q0
protocol.measure_all()

display(protocol.draw('mpl'))
```



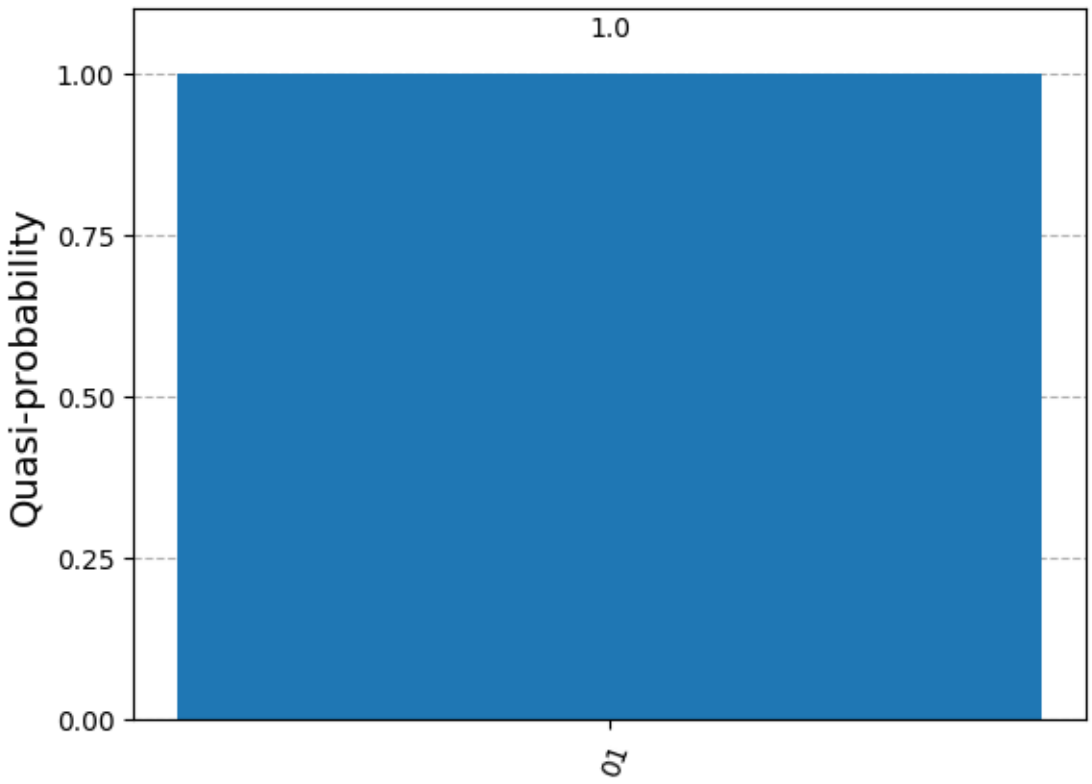
Remark the `measure_all` function, which measures all of the qubits and puts the results into a single classical register (therefore having two bits in this case).

Running the Aer simulator produces the expected output.

[7]:
```
# Replace ?
result = Sampler().run(protocol).result()
statistics = result.quasi_dists[0].binary_probabilities()
```

```
for outcome, frequency in statistics.items():
    print(f"Measured {outcome} with frequency {frequency}")

display(plot_histogram(statistics))
```

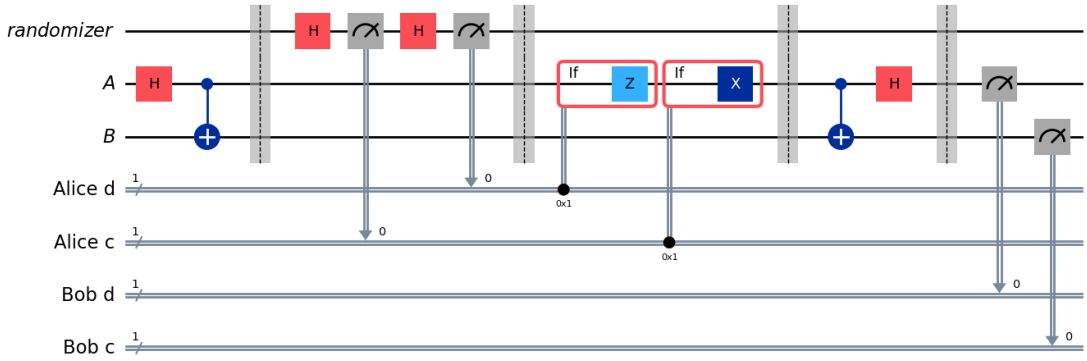
```
Measured 01 with frequency 1.0
```



Just for fun, we can use an additional qubit as a random bit generator to randomly choose c and d, then run the superdense coding protocol to see that these bits are transmitted correctly.

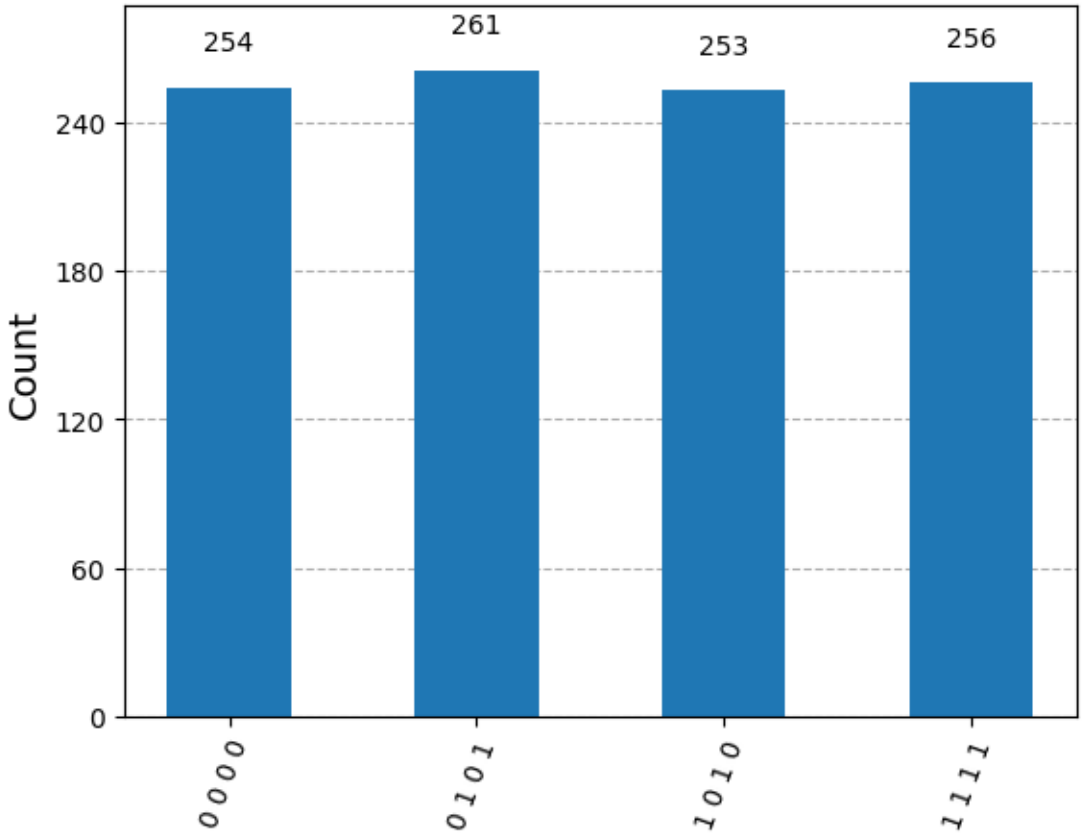
[8]:
```
rbg = QuantumRegister(1, "randomizer")
ebit0 = QuantumRegister(1, "A")
ebit1 = QuantumRegister(1, "B")

Alice_c = ClassicalRegister(1, "Alice c")
Alice_d = ClassicalRegister(1, "Alice d")

test = QuantumCircuit(rbg, ebit0, ebit1, Alice_d, Alice_c)

# Initialize the ebit
# Replace ?
```

```
test.h(ebit0)           # Prepare entanglement
test.cx(ebit0, ebit1)   # Create entanglement between Alice's and Bob's qubits
test.barrier()

# Use the 'randomizer' qubit twice to generate Alice's bits c and d.
# Use the Hadamard Gate to create the random bit, with a measurement.
# Replace ?
test.h(rbg)
test.measure(rbg, Alice_c)# Measure for Alice's c bit
test.h(rbg)# Reapply Hadamard to create another random bit
test.measure(rbg, Alice_d)# Measure for Alice's d bit
test.barrier()

# Now the protocol runs, starting with Alice's actions, which depend
# on her bits.
# Replace ?
with test.if_test((Alice_d, 1), label="Z"):
    test.z(ebit0)# Apply Z gate if Alice's d is 1
with test.if_test((Alice_c, 1), label="X"):
    test.x(ebit0)# Apply X gate if Alice's c is 1
test.barrier()

# Bob's actions
# Replace
test.cx(ebit0, ebit1)# Apply CNOT to decode
test.h(ebit0)# Apply Hadamard to decode
test.barrier()

Bob_c = ClassicalRegister(1, "Bob c")
Bob_d = ClassicalRegister(1, "Bob d")
test.add_register(Bob_d)
test.add_register(Bob_c)
test.measure(ebit0, Bob_d)
test.measure(ebit1, Bob_c)

display(test.draw('mpl'))
```



Running the Aer simulator shows the results: Alice and Bob's classical bits always agree.

[9]:
```
# Replace ?
result = AerSimulator().run(test).result()
statistics = result.get_counts()
display(plot_histogram(statistics))
```



3 End of Notebook

[ ]: