PHYS440 - Homework 1+2

John Hurst

April 2024

Note: As far as I can tell, it is currently not possible to run most circuits created in the IBM Quantum Composer directly on IBM Quantum hardware, because the Quantum Composer does not support automatic transpilation. For this reason, I ran all my circuits using Python Qiskit programs, where it is possible to transpile the circuits. I've included the programs, and images of the original circuits and the transpiled circuits in the submitted files.

---

1. Bell-state realization and correlation measurement on *IBM Quantum* (12 marks)

The *Qiskit* textbook chapter on Multiple Qubits and Entangled States explains how to generate the entangled two qubit state

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{1}$$

and gives the creation of the Bell state

$$\left|\Psi^+\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \tag{2}$$

as an exercise. Read this/remind yourself of the relevant material. Furthermore, read up of how to perform measurements of $Z_0 = \mathbb{1} \otimes Z$ and $Z_1 = Z \otimes \mathbb{1}$ to find the correlator $\langle Z_1 Z_0 \rangle = P_{00} + P_{11} - P_{01} - P_{10}$. Use this preparation to answer the following questions.

(a) Design the quantum circuits that, starting from the two-qubit state $|00\rangle$, create the remaining two Bell states

$$\left|\Phi^-\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{3}$$

$$\left|\Psi^-\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \tag{4}$$

Show (by using matrix representations of the quantum gates) that the circuits that you have designed indeed yield the Bell states as their output. Submit the circuit diagrams and state-vector simulations from *IBM Quantum* as part of your solution.

(b) Consider the two operators $X_0 = \mathbb{1} \otimes X$ and $X_1 = X \otimes \mathbb{1}$. Design the circuits that allow you to measure $\langle \Phi^+ | X_1 X_0 | \Phi^+ \rangle$, where $\left|\Phi^+\right\rangle$ is defined in Eq. (1). Run the circuit first on the simulator and then on one of the available quantum processors. Discuss the result of your experiment. Submit screenshots of your *IBM Quantum* jobs as part of your solution.

(a) There are a number of different circuits possible for each Bell state. These two simple examples are perhaps the simplest circuits for $\left|\Phi^-\right\rangle$ and $\left|\Psi^-\right\rangle$:

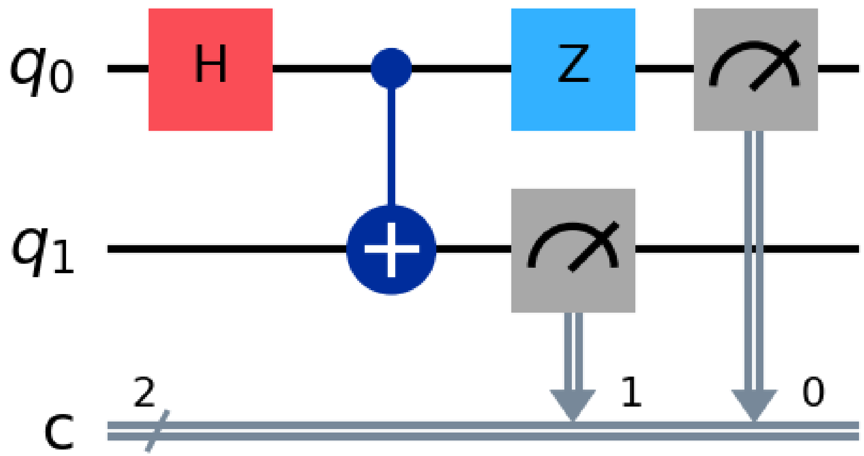

Figure 1: Bell state $\left|\Phi^-\right\rangle$ with Qiskit

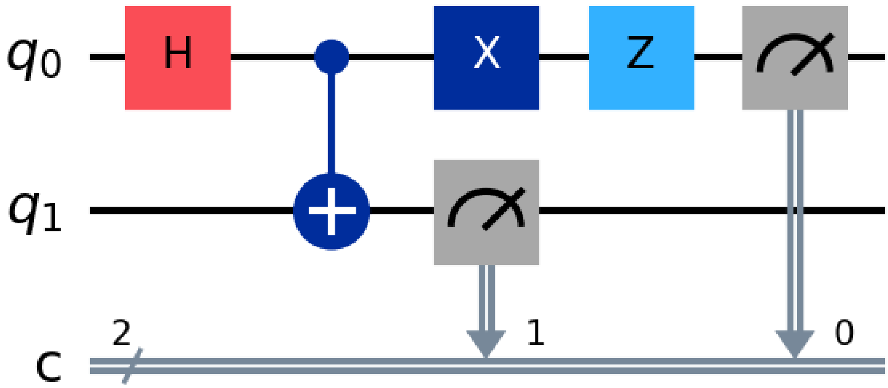

Figure 2: Bell state $|\Psi^{-}\rangle$ with Qiskit

I ran simple circuits for all four Bell states in the IBM Quantum simulator using the web circuit composer. The screen shots are included in the submitted files, and the ones for $|\Phi^{-}\rangle$ and $|\Psi^{-}\rangle$ are shown below.

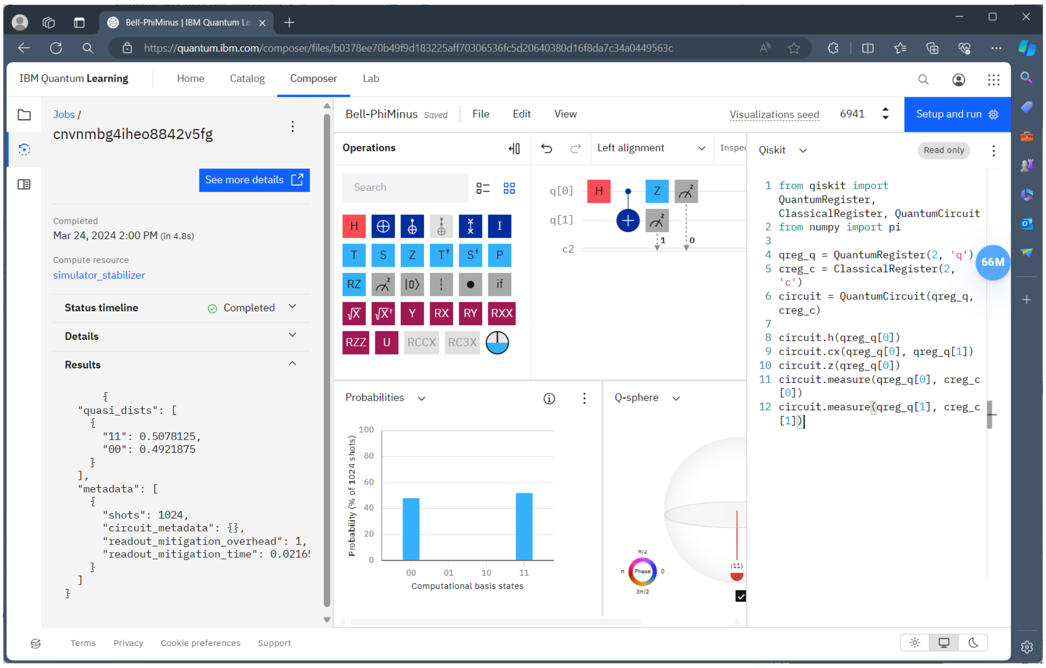

Figure 3: Bell state $|\Phi^{-}\rangle$ on IBM Quantum

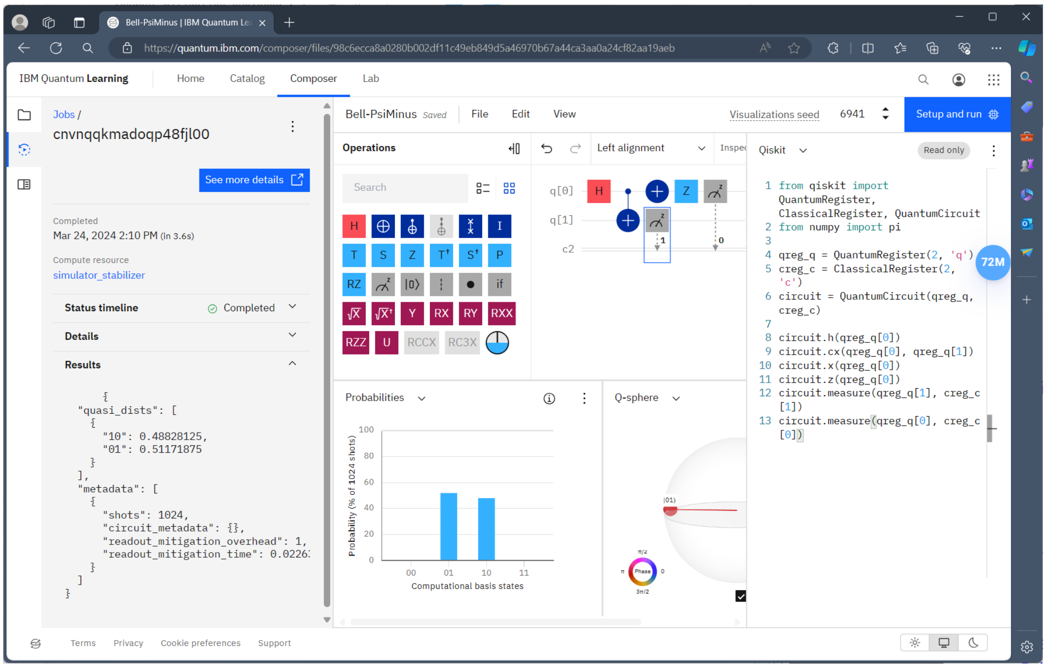

Figure 4: Bell state $\left|\Psi^{-}\right\rangle$ on IBM Quantum

The IBM Quantum systems follow several architectures, each of which supports only a subset of possible gates. The Hadamard and CNOT gates are generally not available, and circuits must be designed or transpiled to use only the available gates.

The attached Mathematica notebook Homework_1+2_Q1a_BellStates.nb shows each Bell state created with simple "canonical" circuits (using H and CNOT), and also with circuits using only the gates available on the IBM Quantum systems. For each circuit the diagram, probability plot, matrix form and output state are shown.

I wrote Python programs using both IBM's Qiskit and Google's Cirq libraries to generate the circuits and simulate them. The programs are included in the submitted files.

The output of the Qiskit program is:

```
bin/homework12_q1a_qiskit.py --simulate --state=phiplus
00: 0.4893
11: 0.5107
bin/homework12_q1a_qiskit.py --simulate --state=phiminus
00: 0.5195
11: 0.4805
bin/homework12_q1a_qiskit.py --simulate --state=psiplus
01: 0.4717
10: 0.5283
bin/homework12_q1a_qiskit.py --simulate --state=psiminus
01: 0.5029
```

```
10: 0.4971
```

The Qiskit version of the program also supports running on the real IBM Quantum hardware, using the Qiskit `Sampler` feature. This includes a transpilation step to convert the circuit to use only the available gates. I captured the transpiled circuits and the results of running them on the IBM Quantum hardware. Because the transpiled circuits include a very large number of ancilla bits, I have cropped the images to show only the relevant parts. I've included the images in the submitted files.

Once the jobs are completed, I queried the results:

```
bin/qiskit_job_status.py cr48fcd0dz600086ntg0
Status: JobStatus.DONE
Backend: ibm_osaka
Tags: ['homework12', 'bell', 'phiplus', 'ibm_osaka']
Sample data for pub 0: {'11': 498, '00': 492, '01': 13, '10': 21}

bin/qiskit_job_status.py cr499xz8091g008jnn80
Status: JobStatus.DONE
Backend: ibm_osaka
Tags: ['homework12', 'bell', 'phiminus', 'ibm_osaka']
Sample data for pub 0: {'11': 469, '01': 35, '00': 491, '10': 29}

bin/qiskit_job_status.py cr49ba52yk500082p5f0
Status: JobStatus.DONE
Backend: ibm_osaka
Tags: ['homework12', 'bell', 'psiplus', 'ibm_osaka']
Sample data for pub 0: {'10': 424, '00': 84, '01': 447, '11': 69}

bin/qiskit_job_status.py cr49c6g0dz600086nwk0
Status: JobStatus.DONE
Backend: ibm_osaka
Tags: ['homework12', 'bell', 'psiminus', 'ibm_osaka']
Sample data for pub 0: {'10': 391, '01': 398, '00': 147, '11': 88}
```

The results are summarised in the table below, as proportions:

Bell state	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
$\left\|\Phi^+\right\rangle$	48%	1%	2%	49%
$\left\|\Phi^-\right\rangle$	48%	3%	3%	46%
$\left\|\Psi^+\right\rangle$	8%	44%	41%	7%
$\left\|\Psi^-\right\rangle$	14%	39%	38%	9%

The results illustrate the presence of noise, because there are a significant number of counts for the states that should not be present.

(b) We can compute X_0 and X_1 and the correlator as:

$$X_0 = \mathbb{1} \otimes X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$X_1 = X \otimes \mathbb{1} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\begin{aligned} X_1 X_0 &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

We can compute $\langle \Phi^+ | X_1 X_0 | \Phi^+ \rangle$ as:

$$\begin{aligned} \langle \Phi^+ | X_1 X_0 | \Phi^+ \rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \\ &= \frac{1}{2}(1+1) \\ &= 1 \end{aligned}$$

The included Mathematica notebook Homework_1+2_Q1b_CorrelationExpectation.nb verifies this analytic calculation.

It also includes a simple quantum circuit to measure the correlator $\langle \Phi^+ | X_1 X_0 | \Phi^+ \rangle$.

I also wrote a Python program using Qiskit Sampler feature to generate the circuit and simulate it.

The circuit run by the program is shown below:

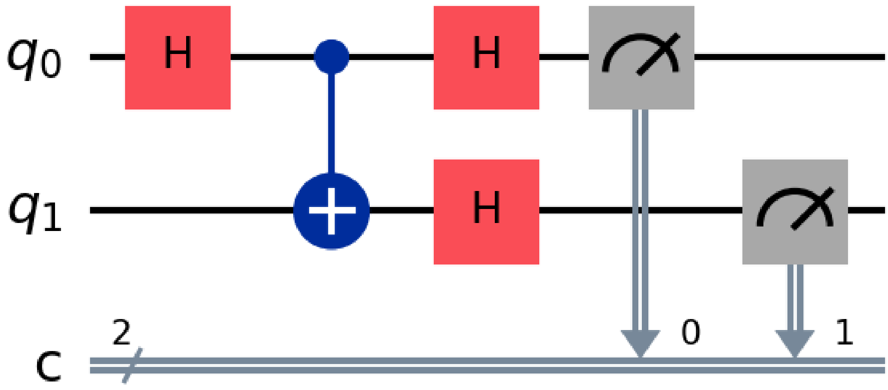

Figure 5: Bell state $\left|\Phi^-\right\rangle$ with Qiskit

Running the program in simulation mode:

```
bin/homework12_q1b_qiskit.py --simulate --filename=xcorrelator_qiskit.png
00: 0.4746
11: 0.5254
```

This shows a clean result with perfect expectation of 1, as expected.

I also ran the program on the IBM Quantum hardware, and the results are shown below:

```
bin/homework12_q1b_qiskit.py --run --pass-manager --filename=
    xcorrelator_qiskit_ibmquantum_full.png
# ... wait for job to complete
bin/qiskit_job_status.py cr6z43r95ey00083jtf0
Status: JobStatus.DONE
Backend: ibm_brisbane
Tags: ['homework12', 'xcorrelator', 'ibm_brisbane']
Sample data for pub 0: {'11': 504, '00': 454, '01': 45, '10': 21}
```

The results are summarised in the table below, as proportions:

State vector	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
Frequency	44%	4%	2%	49%

These results also show the presence of noise, with a significant number of counts for the states that should not be present.

We can also do this experiment more simply in Qiskit using the Estimator feature instead of the Sampler feature.

With the Estimator, we only need to do the circuit to prepare the input state, $\left|\Phi^+\right\rangle$:

```
circuit = QuantumCircuit(2)
circuit.h(0)
circuit.cx(0, 1)
```

Then we can define the observable and run the experiment:

```
# Define observable as 1 * X_1 * X_0.
observable = SparsePauliOp.from_list([("XX", 1)])
```

My program can run this in the simulator:

```
bin/homework12_q1b_qiskit_estimator.py --simulate
[1.]
```

This again gives a perfect result of 1.0.

I also ran the program on the IBM Quantum hardware, and the results are shown below:

```
bin/homework12_q1b_qiskit_estimator.py --run \
  --filename=xcorrelator_qiskit_estimator_ibmquantum_full.png
# ... wait for job to complete
bin/qiskit_job_status.py crct7hpdjmqg008kemeg
Status: JobStatus.DONE
Backend: ibm_osaka
Tags: None
Estimate data for pub 0: [[1.0078125]]
```

The result is 1.0078125, which is very close to the expected value of 1.0, but shows the presence of noise.

---

2. Testing Bell's Inequality on *IBM Quantum* (28 marks)

To answer the questions below, it will be useful to first familiarise yourself with the CHSH inequality, e.g. by reading about it in textbooks and/or the document *A Brief Introduction to Entanglement & the CHSH inequality* from our Lit folder. You will also have to make use of the techniques learned from Question 1 above.

Consider the state $\left|\Psi^-\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ and the following observables

$$A = Z \otimes \mathbb{1} \quad ; \qquad\qquad A' = X \otimes \mathbb{1}$$
$$B = \mathbb{1} \otimes \frac{1}{\sqrt{2}}(X+Z) \quad ; \qquad\qquad B' = \mathbb{1} \otimes \frac{1}{\sqrt{2}}(X-Z).$$

(a) Calculate analytically, using the matrix representation of the relevant quantum gates, the following correlators:

$$\langle\Psi^-|AB|\Psi^-\rangle, \quad \langle\Psi^-|AB'|\Psi^-\rangle, \quad \langle\Psi^-|A'B|\Psi^-\rangle, \quad \langle\Psi^-|A'B'|\Psi^-\rangle, \tag{5}$$

as well as the quantity

$$C = \langle\Psi^-|AB|\Psi^-\rangle - \langle\Psi^-|AB'|\Psi^-\rangle + \langle\Psi^-|A'B|\Psi^-\rangle + \langle\Psi^-|A'B'|\Psi^-\rangle. \tag{6}$$

(b) Design the four quantum circuits that will allow you to measure the expectation values given in Eq. (5). Submit the circuit diagrams from *IBM Quantum* as part of your solution.

(c) Run the circuits on the simulator, evaluate the expectation values from Eq. (5), and check whether the CHSH inequality is violated. Submit screenshots of your *IBM Quantum* jobs as part of your solution.

(d) Now run your circuits on one of the available quantum processors and evaluate the expectation values from Eq. (5) as well as $|C|$, with C given in Eq. (6).

(e) Submit screenshots of your *IBM Quantum* jobs as part of your solution.

(f) Comment on your result, discussing specifically whether—and if so, at what level of confidence—it rules out the possibility of a hidden variable theory.

(a) The included Mathematica notebook Homework_1+2_Q2_CHSH.nb calculates the correlators and the CHSH quantity C analytically.

It gives a result of $C = -2\sqrt{2}$, or $|C| = 2\sqrt{2}$, which violates the CHSH inequality. This is actually the maximum violation predicted by quantum mechanics, according to the Tsirelson bound.

(b) I used the four circuits shown below. Each circuit prepares the input state of $|\Psi^-\rangle$ and then performs the necessary transformation of the quantum state into the computational (Z) basis to perform the measurement of the desired observable.

- ✿ To measure $A = Z_0$ requires no transformation, because Z_0 is already in the computational basis.
- ✿ To measure $A' = X_0$ requires a Hadamard gate on qubit 0 before the measurement.
- ✿ To measure $B = \frac{1}{\sqrt{2}}(X_1 + Z_1)$ requires a Y rotation of $-\frac{\pi}{4}$, achieved by the RY($-\frac{\pi}{4}$) gate.
- ✿ I was not completely sure about the transformation required to measure $B' = \frac{1}{\sqrt{2}}(X_1 - Z_1)$, but it appears that RY($-\frac{\pi}{4}$) followed by a Hadamard gate is the correct transformation.

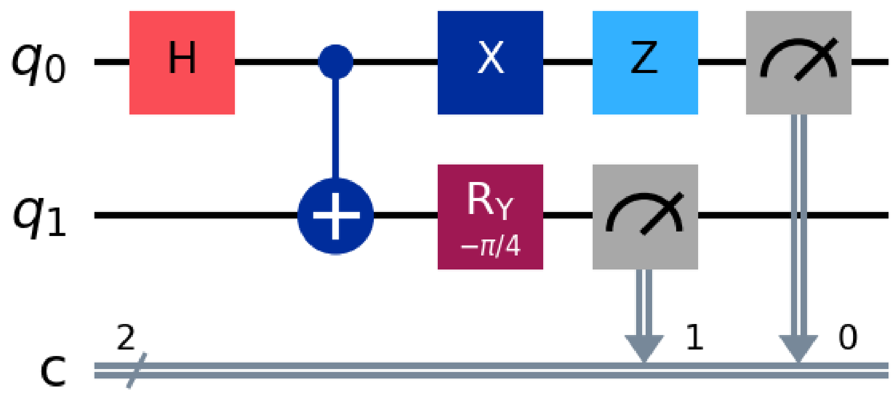

Figure 6: Circuit for measuring $\langle \Psi^- | AB | \Psi^- \rangle$

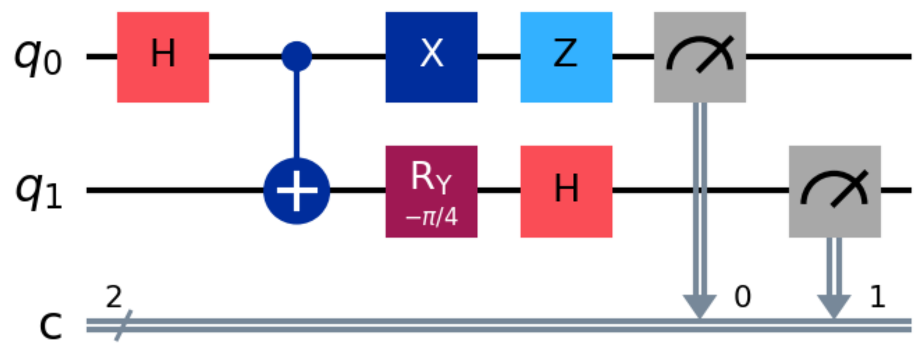

Figure 7: Circuit for measuring $\langle \Psi^- | AB' | \Psi^- \rangle$

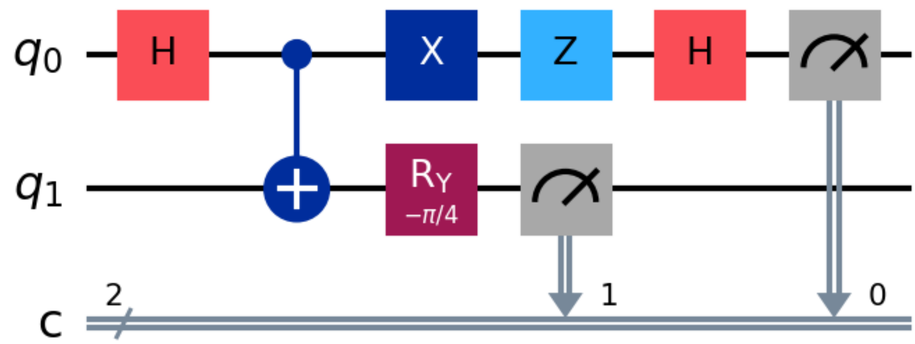

Figure 8: Circuit for measuring $\langle \Psi^- | A'B | \Psi^- \rangle$

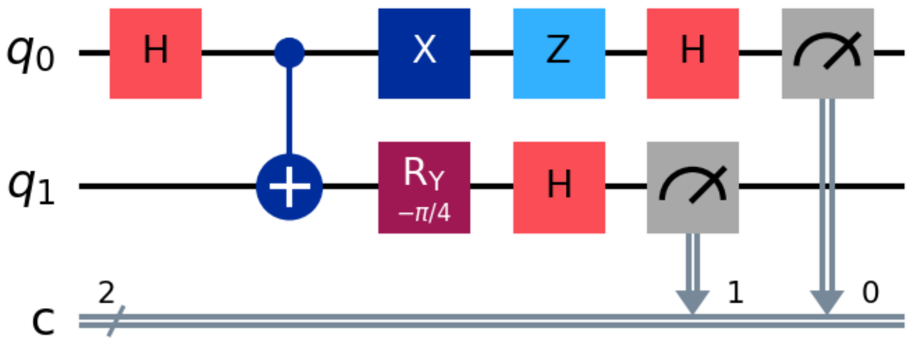

Figure 9: Circuit for measuring $\langle \Psi^- | A' B' | \Psi^- \rangle$

The Mathematica notebook referenced above also evaluates the expectations for the observables using these circuits, and shows that the circuits give the same result as the analytic calculation.

(c) I wrote a Python program using the Qiskit Sampler feature to generate the circuits and simulate them.

The program runs the circuits and prints the statevector frequency proportions:

```
bin/homework12_q2_qiskit.py --obs=ab --simulate
00: 0.0811
01: 0.4160
10: 0.4336
11: 0.0693
bin/homework12_q2_qiskit.py --obs=abp --simulate
00: 0.4326
01: 0.0684
10: 0.0742
11: 0.4248
bin/homework12_q2_qiskit.py --obs=apb --simulate
00: 0.0723
01: 0.4336
10: 0.4219
11: 0.0723
bin/homework12_q2_qiskit.py --obs=apbp --simulate
00: 0.0654
01: 0.4287
10: 0.4355
11: 0.0703
```

We use these proportions to calculate the expectation values for the observables using

$$\langle AB \rangle = P_{00} - P_{01} - P_{10} + P_{11}$$

as shown in this table:

Observable	P_{00}	P_{01}	P_{10}	P_{11}	$P_{00} - P_{01} - P_{10} + P_{11}$
AB	0.0811	0.4160	0.4336	0.0693	-0.6992
AB’	0.4326	0.0684	0.0742	0.4248	0.7148
A’B	0.0723	0.4336	0.4219	0.0723	-0.7109
A’B’	0.0654	0.4287	0.4355	0.0703	-0.7285

The CHSH quantity C is then calculated as:

$$\begin{aligned} C &= \langle AB \rangle - \langle AB' \rangle + \langle A'B \rangle + \langle A'B' \rangle \\ &= -0.6992 - 0.7148 - 0.7109 - 0.7285 \\ &= -2.8534 \end{aligned}$$

This result is close to the analytic result of $-2\sqrt{2} \approx -2.8284$.

The Qiskit Estimator feature is a simpler way to compute the expectations required in this question, because it requires only the definition of a single circuit (for the input state only), and the Estimator takes care of the rest.

I wrote a separate Python program to use the Estimator feature to compute the expectations. The program is included in the submitted files, but the important parts are shown below.

We prepare the qubits in the entangled state $\left|\Psi^-\right\rangle$:

```
chsh_circuit = QuantumCircuit(2)
chsh_circuit.h(0)
chsh_circuit.cx(0, 1)
chsh_circuit.x(0)
chsh_circuit.z(0)
```

Then we define the observables that we are interested in:

$$
\begin{aligned}
AB &= Z_0 \times \frac{1}{\sqrt{2}}(X_1 + Z_1) = \frac{1}{\sqrt{2}}(X_1 Z_0 + Z_1 Z_0) \\
AB' &= Z_0 \times \frac{1}{\sqrt{2}}(X_1 - Z_1) = \frac{1}{\sqrt{2}}(X_1 Z_0 - Z_1 Z_0) \\
A'B &= X_0 \times \frac{1}{\sqrt{2}}(X_1 + Z_1) = \frac{1}{\sqrt{2}}(X_1 X_0 + Z_1 X_0) \\
A'B' &= X_0 \times \frac{1}{\sqrt{2}}(X_1 - Z_1) = \frac{1}{\sqrt{2}}(X_1 X_0 - Z_1 X_0) \\
C &= AB - AB' + A'B + A'B' \\
&= \frac{1}{\sqrt{2}}(X_1 Z_0 + Z_1 Z_0) - \frac{1}{\sqrt{2}}(X_1 Z_0 - Z_1 Z_0) \\
&\quad + \frac{1}{\sqrt{2}}(X_1 X_0 + Z_1 X_0) + \frac{1}{\sqrt{2}}(X_1 X_0 - Z_1 X_0) \\
&= \sqrt{2} Z_1 Z_0 + \sqrt{2} X_1 X_0
\end{aligned}
$$

Technically it would be sufficient to define only C, but I've included the others as required by the question and to show more detail.

The Python code to define these observables is:

```
observable_ab =
    SparsePauliOp.from_list([("XZ", 1/sqrt(2)), ("ZZ", 1/sqrt(2))])
observable_abp =
    SparsePauliOp.from_list([("XZ", 1/sqrt(2)), ("ZZ", -1/sqrt(2))])
observable_apb =
    SparsePauliOp.from_list([("XX", 1/sqrt(2)), ("ZX", 1/sqrt(2))])
observable_apbp =
    SparsePauliOp.from_list([("XX", 1/sqrt(2)), ("ZX", -1/sqrt(2))])
observable_c =
    SparsePauliOp.from_list([("ZZ", sqrt(2)), ("XX", sqrt(2))])
```

We run the circuit using the Qiskit simulator, and evaluate the expectation values:

```
bin/homework12_q2_qiskit_estimator.py --simulate
[-0.70710678  0.70710678 -0.70710678 -0.70710678 -2.82842712]
```

These results agree perfectly with the analytic results:

$$\langle\Psi^-|AB|\Psi^-\rangle = -\frac{1}{\sqrt{2}}$$
$$\langle\Psi^-|AB'|\Psi^-\rangle = \frac{1}{\sqrt{2}}$$
$$\langle\Psi^-|A'B|\Psi^-\rangle = -\frac{1}{\sqrt{2}}$$
$$\langle\Psi^-|A'B'|\Psi^-\rangle = -\frac{1}{\sqrt{2}}$$
$$C = -2\sqrt{2}$$

(d) I ran the individual circuits using the Sampler feature on IBM Quantum hardware. The transpiled circuit diagrams are included in the submitted files.

The first run generated these counts:

Observable	Backend	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
AB	ibm_kyoto	178	306	304	236
AB’	ibm_kyoto	307	170	158	389
A’B	ibm_osaka	115	412	413	84
A’B’	ibm_osaka	119	414	403	88

We convert these counts to proportions and calculate the expectation values:

Observable	P_{00}	P_{01}	P_{10}	P_{11}	$P_{00}-P_{01}-P_{10}+P_{11}$
AB	0.1738	0.2988	0.2969	0.2305	-0.1914
AB’	0.2998	0.1660	0.1543	0.3799	0.3594
A’B	0.1123	0.4023	0.4033	0.0820	-0.6113
A’B’	0.1162	0.4043	0.3936	0.0859	-0.5957

The CHSH quantity C is then calculated as:

$$\begin{aligned} C &= \langle AB\rangle - \langle AB'\rangle + \langle A'B\rangle + \langle A'B'\rangle \\ &= -0.1914 - 0.3594 - 0.6113 - 0.5957 \\ &= -1.7578 \end{aligned}$$

So, from this experiment we get $|C| < 2$, which does not violate the CHSH inequality. However, this result might have been affected by noise in the IBM Quantum systems. I ran the same circuits a second time, and happened to get them run on a different combination of IBM Quantum backends:

Observable	Backend	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
AB	ibm_osaka	118	399	427	80
AB’	ibm_osaka	434	98	98	394
A’B	ibm_osaka	98	425	423	78
A’B’	ibm_kyoto	151	343	345	185

This time the proportions were more consistent with the analytic and simulator results:

Observable	P_{00}	P_{01}	P_{10}	P_{11}	$P_{00}-P_{01}-P_{10}+P_{11}$
AB	0.1152	0.3896	0.4170	0.0781	-0.6133
AB’	0.4238	0.0957	0.0957	0.3848	0.6172
A’B	0.0957	0.4150	0.4131	0.0762	-0.6563
A’B’	0.1475	0.3350	0.3369	0.1807	-0.3438

Now, the CHSH quantity C is calculated as:

$$
\begin{aligned}
C &= -0.6133 - 0.6172 - 0.6563 - 0.3438 \\
&= -2.2306
\end{aligned}
$$

This result does violate the CHSH inequality, but it is still less than the maximum violation predicted by quantum mechanics.

It appears that the ibm_koyoto backend did not give very good results for these circuits at the time I ran them, while the ibm_osaka backend gave more consistent results.

I also ran the Estimator version of the program on the IBM Quantum hardware:

```
bin/homework12_q2_qiskit_estimator.py --run \
    --filename=chsh_qiskit_estimator_ibmquantum_full.png
# ... wait for job to complete
bin/qiskit_job_status.py crcwnppdqdn000834ksg
Status: JobStatus.DONE
Backend: ibm_osaka
Tags: ['homework12', 'chsh', 'ibm_osaka']
Estimate data for pub 0: [[-0.65312916  0.77907694 -0.73409559
  -0.7376941  -2.90399579]]
```

The numbers are similar to the simulator results, but again show the presence of noise in the IBM Quantum system.

(e) Interestingly, the result from the IBM Quantum system using the Estimator has $|C| = 2.90$, which is actually greater than the $2\sqrt{2} \approx 2.83$ maximum violation predicted by quantum mechanics.

> According to theory, a result greater than 2 implies that the system is not compatible with a local hidden variable theory. However, given the obvious noise issues, I am not sure what level of confidence we can assign to this result. Although the deviation is large (in the Estimator case larger even than quantum mechanics predicted!), our experiments on IBM Quantum have displayed a large amount of unexpected results due to noise. I'm not sure we can apply any formal statistical analysis to this result, without accounting for the statistical properties of the noise.

I was interested in the noise effect from the different IBM Quantum systems, and the fact that the ibm_osaka backend results are more consistent with theory than those of the ibm_kyoto backend. It turns out that Qiskit supports simulation using noise models for the IBM Quantum systems, based on the calibration data that IBM provides. There are two ways to use this feature: we can use static `FakeKyoto` and `FakeOsaka` backends, which are pre-calibrated noise models based on data from some time in the past, or we can use the latest calibration results via the `AerSimulator` class.

I thought it would be interesting to run a large number of trials and compare the distributions for the estimate of C from the different backends.

I tried with the `AerSimulator` class, but it was very slow, I guess because of needing to collect live data from the IBM Quantum backends. So I decided to use the `FakeKyoto` and `FakeOsaka` backends. I ran 100 trials for each backend, each time estimating the observables $\langle AB \rangle$, $\langle AB' \rangle$, $\langle A'B \rangle$, $\langle A'B' \rangle$ and C. I then plotted histograms of the results, shown below:

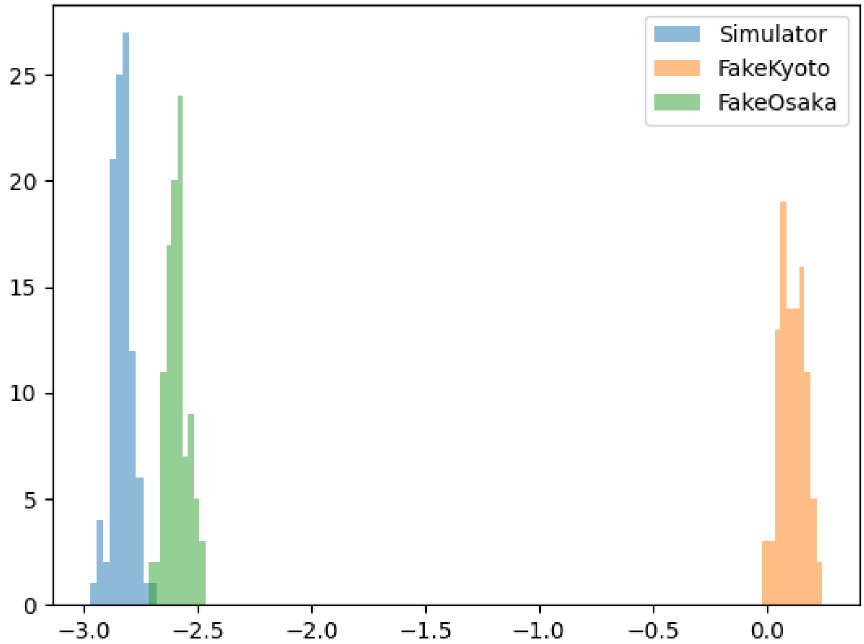

Figure 10: Distributions of the estimate of C

This graph shows a pretty bad result for the ibm_kyoto backend. It would be interesting to know whether the current calibration results give as bad a result, but in my own CHSH runs I did see that the ibm_kyoto backend gave considerably worse results than the ibm_osaka backend.