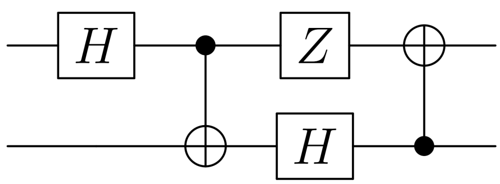
Homework #2

Due by 10/29/2025 before class

1. In HW#1, we consider the following quantum circuit C.



Now we simulate it using Qiskit.

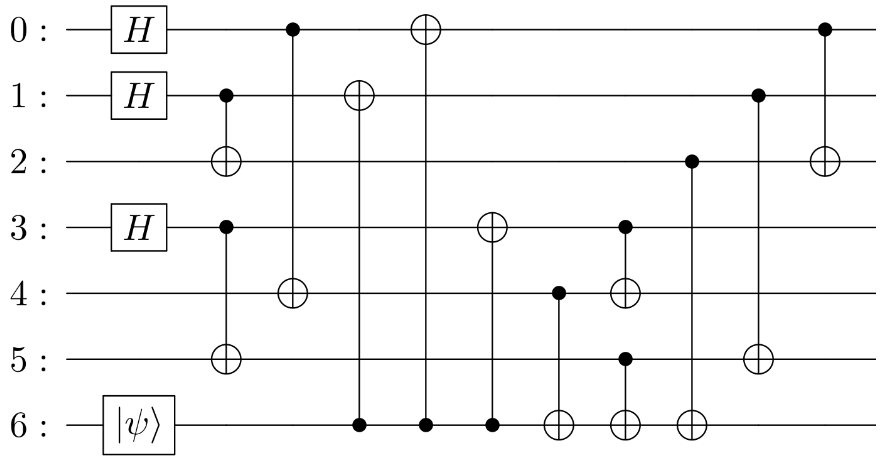
(a) [10 pt] Let Qiskit compute the unitary U for this circuit C. Note that in Qiskit the qubits are ordered from right to left.

(b) [10 pt] Apply the circuit C to the initial state $|0\rangle|0\rangle$ and measure in the computational basis. Get the distribution of the measurement outcome for 1000 trials and plot the histogram.

2. [15 pt] Use the 1-bit quantum full adder circuit in the class to construct an n-bit quantum full adder. It should accept two n-qubit registers for input and one $(n+1)$-qubit register as ancilla. Test your n-bit quantum full adder for randomly generated 5-bit inputs in computational basis.

3. Consider 7 physical qubits encoding the Steane code with stabilizers $X_4X_5X_6X_7$, $X_2X_3X_6X_7$, $X_1X_3X_5X_7$, $Z_4Z_5Z_6Z_7$, $Z_2Z_3Z_6Z_7$ and $Z_1Z_3Z_5Z_7$. Suppose all the encoding, stabilizer measurement and error correction operations are perfect, while single-qubit errors are added by hand after encoding.

(a) [10 pt] The following circuit can encode the logical state for general single-qubit input state $|\psi\rangle$. Run this circuit and give the expressions for logical states $|0_L\rangle$ and $|1_L\rangle$. [Hint: For the state vector psi you get, you can use psi.reverse_qargs() to recover our order convention.] Also construct a circuit for decoding of the logical state. (Note that in practice, we will not use these circuits for encoding and decoding, but will directly perform fault-tolerant operations on the physical qubits.)

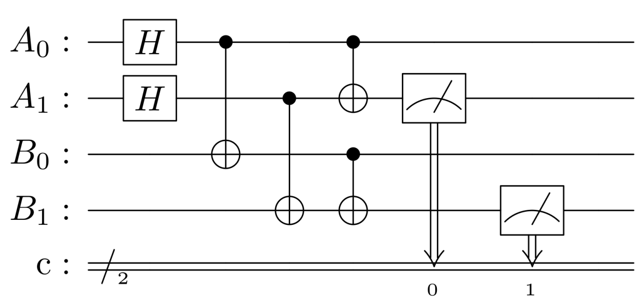


(b) [10 pt] Construct circuits to measure the stabilizers of the Steane code. Each stabilizer measurement should use a different ancilla qubit. Combine these circuits with the encoding circuit to verify that the logical states $|0_L\rangle$ and $|1_L\rangle$ are both simultaneous eigenstates of all the stabilizers with the eigenvalue of one.

(c) [10 pt] In HW#1, we obtain the error syndromes for all the possible single-qubit errors. Use them to construct an error correction circuit for all the possible error syndromes. [Hint: You may use *if_test* or *switch* to deal with different error syndromes. For the Steane code or the general CSS code, we can treat the X or Z types of errors separately.]

(d) [10 pt] Combine your encoding, stabilizer measurement, error correction and decoding circuits together. Insert single-qubit error between encoding and stabilizer measurement by hand, and verify that we still get correct states after decoding. Specifically, consider the logical state $\frac{1}{\sqrt{2}}(|0\rangle_L + i|1\rangle_L)$, and four possible single-qubit errors X_1, Z_2, Y_3 and $[I + i(X_4 + Y_4 + Z_4)]/2$.

4. Entanglement purification is a process where we start from a few copies of imperfect EPR states and try to generate an EPR state with higher fidelity. The following circuit demonstrates entanglement purification for two EPR states between Alice and Bob. First we prepare two EPR pairs (A_0B_0 and A_1B_1) where Alice and Bob can be far apart. Then Alice and Bob perform local CNOT gates on their two qubits, respectively, and measure one qubit each. Conditioned on the two measurement outcomes being 00 or 11, the remained two qubits will be in an EPR state between Alice and Bob.



(a) [10 pt] Run this circuit and show that in the ideal case, the measurement outcome is either 00 or 11, and that after the measurement we always get a perfect EPR state. Specifically, run the circuit for 1000 trials, plot the distribution for the measurement outcome and compute the average fidelity between the state of the remaining qubits and a perfect EPR state conditioned on the measurement outcome being 00 or 11. [Hint: To simulate the final state vector conditioned on the classical measurement outcome, you can use *QuantumCircuit.save_statevector(label='psi', conditional=True)*. In the result of your simulation, you can use e.g. *result.data()['psi']['0x0']*, *result.data()['psi']['0x1']*, etc. to access the state vector conditioned on different measurement outcome.]

(b) [15 pt] Now suppose the generation of distant EPR states is subjected to error, e.g. due to the long-distance transmission. On the other hand, assume all the local operations by Alice or Bob are still perfect. You can simulate this error model by the following code

```
from qiskit_aer.noise import NoiseModel, depolarizing_error
p = 0.1
err = depolarizing_error(p, 2)
noise_model = NoiseModel(basis_gates=['rx', 'ry', 'rz', 'cx'])
noise_model.add_quantum_error(err, 'cx', [0, 2])
noise_model.add_quantum_error(err, 'cx', [1, 3])
```

where we consider an error rate of $p = 10\%$ to generate distant entanglement between qubits 0 and 2, and between qubits 1 and 3. Again, run this circuit for 1000 trials, plot the distribution for the measurement outcome and compute the average fidelity between the state of the remaining qubits and a perfect EPR state conditioned on the measurement outcome being 00 or 11. [Hint: To simulate the final density matrix conditioned on the classical measurement outcome, you can use *QuantumCircuit.save_density_matrix(label='rho', conditional=True)* in a similar way as *QuantumCircuit.save_statevector()*. In addition, compute the average fidelity for the initially prepared EPR states. You should see an improvement in the fidelity after entanglement purification.