Практическое задание 2

Задача: Реализовать схему для задачи сверхплотного кодирования. Проверить программу для получения всех классических значений для двух кубит.

Решение

Ссылка на GitHub: [nAglTI/QuantumComputations: Python scripts for Quantum Computing Assignments. (github.com)](https://github.com/nAglTI/QuantumComputations)

Алгоритм:

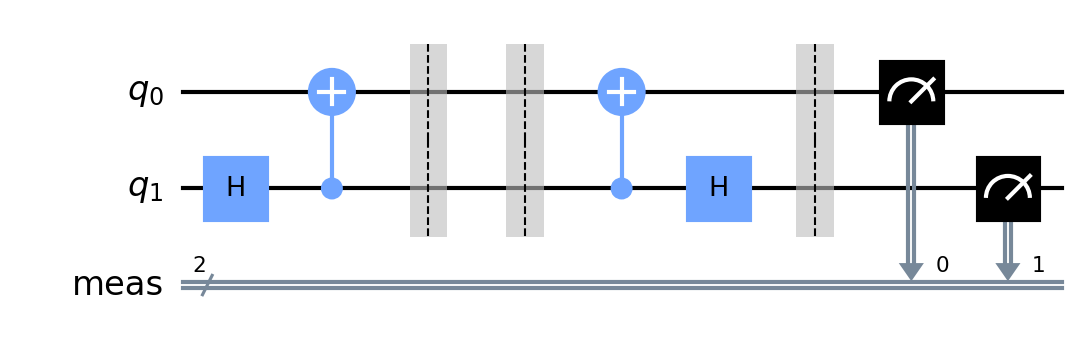
1. Реализация функции создания пары Белла
2. Реализация функций кодирования и декодирования сообщения
3. Запуск алгоритма для каждого из состояний (00, 01, 10, 11)
   1. Адресат A кодирует сообщение в кубит 1
   2. Декодирование этого кубита адресатом B
   3. Измерение кубита
   4. Вычисление результата
   5. Сохранение схемы и гистограммы

Код решения:

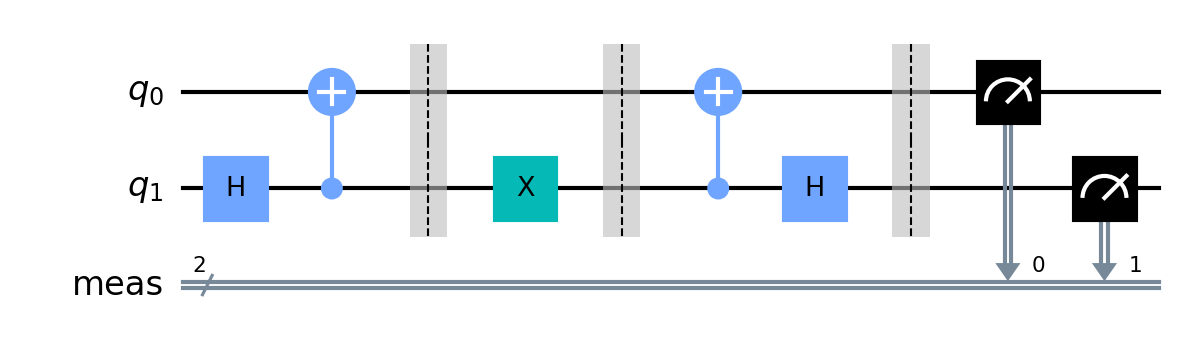
# Cоздание пары Белла в виде схемы  
def bellPairInit():  
 qc = QuantumCircuit(2)  
 qc.h(1)  
 qc.cx(1, 0)  
 return qc  
  
  
# Энкодинг  
def encodeMessage(qc, qubit, msg):  
 if len(msg) != 2 or not set(msg).issubset({"0", "1"}):  
 raise ValueError(f"message '{msg}' is invalid")  
 if msg[1] == "1":  
 qc.x(qubit)  
 if msg[0] == "1":  
 qc.z(qubit)  
 return qc  
  
  
# Декодинг  
def decodeMessage(qc):  
 qc.cx(1, 0)  
 qc.h(1)  
 return qc  
  
  
def task2():  
 calcResult('00')  
 calcResult('01')  
 calcResult('10')  
 calcResult('11')  
  
  
def calcResult(message):  
 # Получение пары Белла  
 qc = bellPairInit()  
 qc.barrier()  
 # Кубит 0 переходит к A, а кубит 1 переходит к B  
 # A кодирует сообщение в кубит 1.  
 qc = encodeMessage(qc, 1, message)  
 qc.barrier()  
 # A отправляет кубит адресату B  
 # B применяет функцию декодирования  
 qc = decodeMessage(qc)  
 # B измеряет кубит  
 qc.measure\_all()  
 qc.draw()  
 # Вычисление результата, сохранение схемы и гистограммы  
 qc.draw(output='mpl', filename=f'task2\_{message}.png')  
 aer\_sim = Aer.get\_backend('aer\_simulator')  
 result = aer\_sim.run(qc).result()  
 counts = result.get\_counts(qc)  
 print("Task 2 result:", counts)  
 plot\_histogram(counts)  
 plt.savefig('histogram\_task2.png')

Результаты для каждого сообщения:

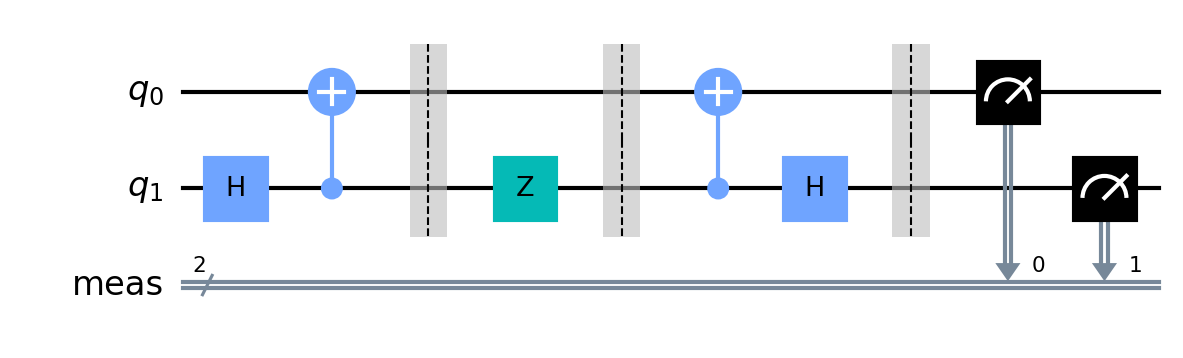
* 00:



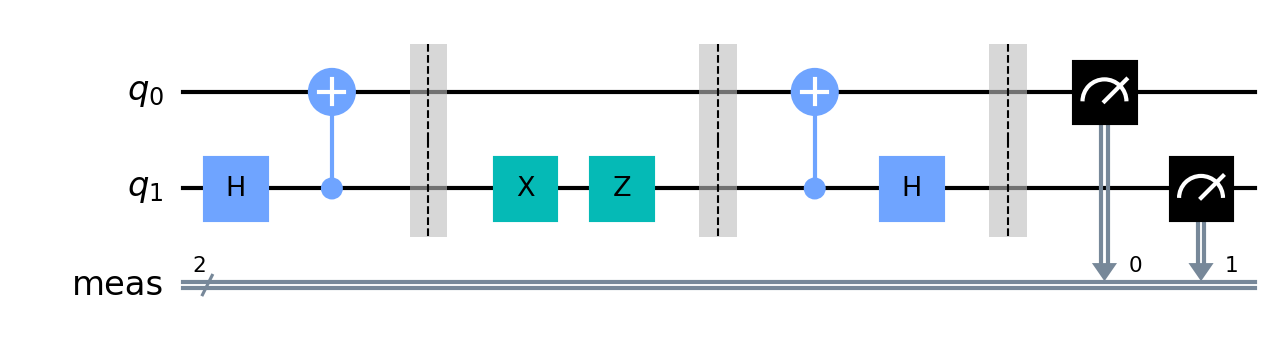
* 01:

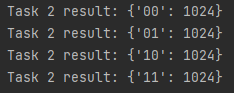


* 10:



* 11:





Для каждого сообщения гистограмма совпадает:

