

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

Кафедра ИИТ

ОТЧЁТ
По лабораторной работе №4
«Сумматор в квантовых схемах»

Выполнил:
Студент группы ИИ-22
Нестерчук Д.Н.
Проверил: Хацкевич
А.С.

2024

Цель работы: ознакомление с выполнением простого сложения с помощью квантовых схем.

Задачи:

1. Изучить теоретический материал.
2. Произвести регистрацию на сайте <https://quantum.ibm.com/> для получения токена API или дальнейшей работы с интерактивной средой IBM Quantum Experience
3. Средствами Qiskit или используя средства интерактивной среды IBM Quantum Experience <https://quantum.ibm.com/composer/>, создать квантовую схему полного сумматора.

Полный сумматор принимает на вход два двоичных числа плюс бит переполнения, который мы назовем X. Создайте полный сумматор с входными данными:

A=1, B=0, X=1.

Ход работы

Код программы:

```
from qiskit import QuantumRegister, ClassicalRegister,
QuantumCircuit
from qiskit.visualization import plot_histogram
from qiskit_aer import Aer
import matplotlib.pyplot as plt

def run_full_adder_simulation(a, b, c_in, display_circuit=False):
    A = QuantumRegister(1, 'A')
    B = QuantumRegister(1, 'B')
    C_in = QuantumRegister(1, 'C_in')
    S = QuantumRegister(1, 'S')
    C_out = QuantumRegister(1, 'C_out')

    cr = ClassicalRegister(2, 'cr')
    qc = QuantumCircuit(A, B, C_in, S, C_out, cr)

    # Устанавливаем значения входных кубитов
    if a:
        qc.x(A[0])
    if b:
        qc.x(B[0])
    if c_in:
        qc.x(C_in[0])

    # Реализация схемы полного сумматора
    qc.cx(A[0], S[0]) # S = A XOR B
    qc.cx(B[0], S[0]) # S = A XOR B
    qc.cx(C_in[0], S[0]) # S = A XOR B XOR C_in

    # Перенос: C_out = (A AND B) OR (C_in AND (A OR B))
    qc.ccx(A[0], B[0], C_out[0]) # C_out = A AND B
    qc.ccx(A[0], C_in[0], C_out[0]) # C_out = (A AND B) OR (C_in
AND A)
    qc.ccx(B[0], C_in[0], C_out[0]) # C_out = (A AND B) OR (C_in
AND B)

    qc.measure(S[0], cr[0])
```

```

qc.measure(C_out[0], cr[1])

# Если это последняя итерация, показываем квантовую схему
if display_circuit:
    print("Квантовая схема для A=1, B=1, C_in=1:")
    qc.draw(output='mpl')
    plt.show()

# Выполняем симуляцию
simulator = Aer.get_backend('aer_simulator')
job = simulator.run(qc, shots=1)
result = job.result()
counts = result.get_counts(qc)

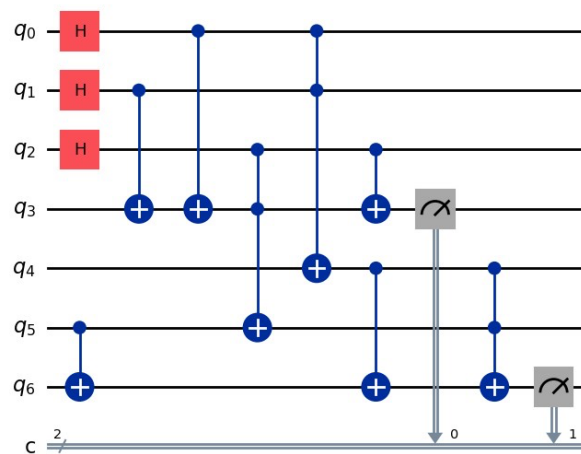
# Извлекаем результаты измерений
measured_result = list(counts.keys())[0]
s_result = measured_result[1] # S (сумма)
c_out_result = measured_result[0] # C_out (перенос)

return s_result, c_out_result

def main():
    print("A B C_in | S C_out")
    print("-----")
    for a in [0, 1]:
        for b in [0, 1]:
            for c_in in [0, 1]:
                display_circuit = (a == 1 and b == 1 and c_in == 1)
                s_result, c_out_result =
run_full_adder_simulation(a, b, c_in, display_circuit)
                print(f"{a} {b}   {c_in}   | {s_result}
{c_out_result}")

if __name__ == "__main__":
    main()

```



Результат работы:

Вывод: ознакомился с выполнением простого сложения с помощью квантовых схем.