

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
Кафедра интеллектуальных информационных технологий

Отчет
по дисциплине
«Современные методы защиты информации»
по лабораторной работе № 4
«Сумматор в квантовых схемах»

Выполнил:
студент 4 курса
группы ИИ-22
Полиенко В.Э.
Проверила:
Хацкевич А.С.

Брест 2024

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

Постановка задачи:

- Изучить теоретический материал.
- Средствами Qiskit или используя средства интерактивной среды IBM Quantum Experience, создать квантовую схему полного сумматора.

A(input)	B(input)	X(carry input)	S(sum)	C(carry out)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Ход работы:

```
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()

```

Получили следующую схему квантовых вентилей:

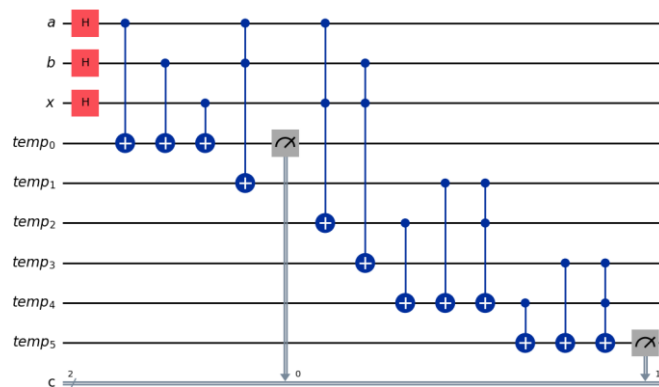


Рисунок 2 – Схема полного сумматора на квантовых вентилях

Вывод программы:

A	B	X	S	C	
0	0	0	0	0	0.1249
0	0	1	1	0	0.3749
0	1	0	1	0	0.3749
0	1	1	0	1	0.3754
1	0	0	1	0	0.3749
1	0	1	0	1	0.3754
1	1	0	0	1	0.3754
1	1	1	1	1	0.1248

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