**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА КОРОЛЕВА**

**МОСКОВСКОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГИМНАЗИЯ №5 (МБОУ ГИМНАЗИЯ №5)**

**КВАНТОВЫЙ КОМПЬЮТЕР.**

**СИМУЛЯЦИЯ.**

проект (исследовательская работа)

по предмету информатика

Ученика 10 класса В

Арутюняна Андрея

Научный руководитель

Климович Лидия Николаевна

Королев

2020-2021 учебный год

Содержание . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2 Введение . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3 Глава I Теория . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4  
1.1 Квантовая механика . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4   
1.2 Квантовые объекты . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4 1.3 Состояние квантовой системы . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7 1.4 Квантовые операторы . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8 1.5 Измерение элементов квантовой системы . . . . . . . . . . . . . . . . . . . . . . . . . .13 1.6 Нюанс в производительности . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15 1.7 Квантовые алгоритмы . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15   
Глава II Практика. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18  
2.1 Тензорное произведение. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .18 2.2 Инициализация симулятора . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18 2.3 Преобразование квантовой системы . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20 2.4 Измерение квантовых объектов . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23 2.5 Квантовая телепортация . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24 Итоги . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27  
Заключение . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28 Список использованных источников и литературы . . . . . . . . . . . . . . . . . . . . . . 29

**Введение**

Актуальность темы:

В ежегодном отчете аналитической компании Gartner «Цикл зрелости технологий» (Hype Cycle for Emerging Technologies) квантовые компьютеры — один из основных трендов 2019 года наряду с искусственным интеллектом и блокчейном.

По версии Gartner, технологии проходят несколько циклов зрелости — триггер, пик чрезмерных ожиданий, разочарование, работа над ошибками, плато продуктивности. Согласно [графику](https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/) за 2018 год, плодотворная работа с квантовыми вычислительными системами начнется через 5-10 лет, а это значит, что стартаперам и инвесторам стоит быть наготове, чтобы не упустить момент, когда вкладываться в квантовые технологии будет поздно.

Несколько лет назад о намерении построить работающие прототипы квантовых компьютеров заявила не только давно изучавшая эту сферу компания IBM, но и Google, Intel и Microsoft.

При этом канадская компания D-Wave с 2011 года уже выпускает и продает так называемые адиабатические компьютеры — сегодня компания [предлагает](https://www.rbc.ru/magazine/2018/01/5a393ff49a794761b9379d07) уже 2000-кубитные компьютеры, а через два года обещает 20 тысяч кубит.

Объект исследования – Квантовый компьютер

Предмет исследования – Квантовое превосходство

Метод исследования – Моделирование и анализ

Цель – Понять является ли квантовый компьютер универсальной заменой обычному.

Задачи:

1. Изучить литературу в области квантовой механики, линейной алгебры
2. Написать симуляцию квантового регистра
3. Протестировать квантовый алгоритм
4. Сделать выводы о работе квантового компьютера
5. Сравнить квантовый компьютер с обычным компьютером
6. Подтвердить или опровергнуть гипотезу проекта

Гипотеза – Квантовый компьютер выполняет операции быстрее, чем обычный компьютер.

**Глава I. Теория**

**1.1 Квантовая механика**

Квантовый компьютер использует такие явления квантовой механики, как квантовая суперпозиция и квантовая запутанность.

Квантовая суперпозиция – явление в квантовой механике, при котором субатомная частица способна находится одновременно в двух и более состояниях. В этом случае состояние субатомной частицы представляется как сумма вероятностей всех ее возможных состояний.

Субатомный мир – мир, частицы которого меньше атома (субатомные частицы).

Квантовая запутанность – явление в квантовой механике, при котором состояния субатомных частиц взаимозависимы.

Квантовые вычисления – вычисления, выполняющиеся на квантовом компьютере, преимущество которых в том, что они могут выполнять параллельно работающие алгоритмы гораздо быстрее обычного компьютера, благодаря квантовой механике.

**1.2 Квантовые объекты**

Кубит – вычислительная единица квантового компьютера. Кубит в может быть в состояниях:

* |0〉 и |1〉 (обозначения Дирака в соответствии с рисунком 1) – бинарное состояние
* Квантовая суперпозиция
* Квантовая запутанность с другими кубитами

Рисунок 1 - Обозначение Дирака

Состояние кубита в суперпозиции выражается следующим образом:

α|0〉 + β|1〉

где α и β - амплитуды вероятностей, комплексные числа. α - амплитуда вероятности 0, а β соответственно амплитуды вероятности 1. В этом выражении состояния кубита важно то, что

.

Амплитуды вероятности α и β возникают из модели состояний кубита в трехмерном пространстве: если изобразить модель состояний бита в трехмерном пространстве, то мы получим две точки 0 и 1 в соответствии с рисунком 2.



Рисунок 2 – Модель состояния бита в трехмерном пространстве

Кубит может существовать в этих состояниях, однако он может быть и в состояниях между ними (квантовая суперпозиция). Следовательно, в трехмерном пространстве кубит будет выглядеть как сфера в вершинах двух точек, полученных с помощью модели состояний бита в соответствии с рисунком 3.



Рисунок 3 – Модель кубита в состоянии суперпозиции

Кубит имеет спин – вектор поворота частицы. Именно спин определяет текущее состояние кубита. На сфере Блоха спин будет выглядеть как радиус вектор сферы, проведенный к точке на ее поверхности и полученный из модели кубита в трехмерном пространстве. Вершины 0 и 1 изменятся на |0〉 и |1〉 соответственно. Расположим данную сферу в начале трехмерной системы координат xyz. Проведем радиус вектор |ψ〉 в точку на поверхности данной сферы. Этот вектор – спин кубита. Отметим угол θ, как угол между вектором |ψ〉 и осью z. Далее спроецируем вектор |ψ〉 на окружность, в которой лежат оси x и y. Получим угол ϕ, как угол между проекцией вектора |ψ〉 и осью x в соответствии с рисунком 4.



Рисунок 4 – Сфера Блоха. Модель спина кубита в состоянии суперпозиции

Теперь мы можем выразить амплитуды вероятностей α и β:

Теперь кубит можно описать как нормированный комплексный вектор в двухмерном пространстве:

〈ψ|ψ〉 = 1

|ψ〉 =

**1.3 Состояние квантовой системы**

Пусть n – число кубит в квантовом регистре. В таком случае каждый из них изначально в состоянии |0〉. Каждое такое состояние мы можем разложить как вектор . Далее производя тензорное произведение всех этих состояний и получим инициализированное состояние квантовой системы, как комплексный вектор в длиной 2n.

Примеры:

Если n = 2

|ψ〉 = 1|00〉 + 0|01〉 + 0|10〉 + 0|11〉

Если n = 3

|ψ〉 = 1|000〉 + 0|001〉 + 0|010〉 + 0|011〉 + 0|100〉 + 0|101〉 + 0|110〉 + 0|111〉

**1.4 Квантовые операторы**

Квантовыми операторами в квантовом компьютере являются матрицы, на которые мы с помощью тензорного произведения перемножаем амплитуды вероятностей.

Тензорное произведение – способ перемножения векторов, матриц и т.д. Оно обозначается символом: ⊗. Матрицы квантовых операторов уникальны тем, что они унитарны, т.е. всегда можно получить обратные к ним матрицы, которые вернут квантовое состояние в исходное состояние.

Основные квантовые операторы:

* Паули X (X) =
* Паули Y (Y) =
* Паули Z (Z) =
* Адамара (H) =
* Фазовый сдвиг (S / P) =
* Перестановка (SWAP) =
* Контролируемый X (CX) =
* Контролируемый Y (CY) =
* Контролируемый Z (CZ) =

Можно заметить, что контролируемые квантовые операторы (т.е. результат тензорного произведения на которых зависит от состояния кубита-контроллера) строятся с помощью формулы:

Контролируемые операторы имеют кубит-контроллер и кубит-цель. Оператор G выполняется на кубит-контроллер, оставляя неизменным кубит-контроллер в случае если кубит-котроллер в базисном состоянии равен 1

В качестве примера можно рассмотреть квантовый оператор Паули X. Этот оператор является аналогом логического оператора NOT на обычном компьютере. Итак, возьмем кубит в бинарном состоянии, который изначально находится в состоянии |0〉. Далее раскладываем это состояния, как комплексный вектор:

Теперь можно применить к этому кубиту квантовый оператор Паули X, используя тензорное произведение:

Далее необходимо преобразовать матрицу в вектор:

Теперь можем получить новое состояние кубита, после применения квантового оператора Паули X:

Итоговое преобразование выглядит так:

В случае если кубит будет находится в состоянии суперпозиции в роли комплексного вектора выступает вектор амплитуд вероятностей:

В этом случае преобразование выглядит аналогично:

В итоге получается преобразование:

Но в квантовом регистре больше, чем 1 кубит. Поэтому нужно влиять на все квантовое состояние. Это делается с помощью матрицы унитарной эволюции. Формируется она следующим способом (на примере CX(0, 1), n = 3)

CX =

Пусть изначальное квантовое состояние равно |ψ〉:

|ψ〉 = 0|000〉 + 0|001〉 + 0|010〉 + 0|011〉 + 1|100〉 + 0|101〉 + 0|110〉 + 0|111〉

Формируем список базисных состояний, где базис по индексу кубита-контроллера равен 1:

[|100〉, |101〉, |110〉, |111〉]

Далее в соответствии с оператором формируем выходные базисные состояния:

|000〉 => |000〉

|001〉 => |001〉

|010〉 => 010〉

|011〉 => 011〉

|100〉 => 110〉

|101〉 => 111〉

|110〉 => 100〉

|111〉 => 101〉

[|110〉, |111〉, |100〉, |101〉]

После этого создаем матрицу размером (2n, 2n), состоящую из нулей. Каждый индекс матрицы обозначает базисные состояния квантовой системы. В нашем случае:

0 = |000〉 4 = |100〉

1 = |001〉 5 = |101〉

2 = |010〉 6 = |110〉

3 = |011〉 7 = |111〉

Тогда, исходя из этих индексов, можно определить, как именно переходят базисные состояния. В итоге мы получим следующую матрицу унитарной эволюции:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |000〉 | |001〉 | |010〉 | |011〉 | |100〉 | |101〉 | |110〉 | |111〉 |
|  |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| |000〉 | 0 | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| |001〉 | 1 | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** |
| |010〉 | 2 | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** |
| |011〉 | 3 | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** |
| |100〉 | 4 | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** |
| |101〉 | 5 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** |
| |110〉 | 6 | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| |111〉 | 7 | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |

Таким образом, применив эту матрицу унитарной эволюции, получаются следующие состояния:

|ψ〉 = 0|000〉 + 0|001〉 + 0|010〉 + 0|011〉 + 1|100〉 + 0|101〉 + 0|110〉 + 0|111〉

CX|ψ〉= 0|000〉 + 0|001〉 + 0|010〉 + 0|011〉 + 0|100〉 + 0|101〉 + 1|110〉 + 0|111〉

Но квантовые операторы не всегда состоят только из 0 или 1. Если применить контролируемый квантовый оператор CH (0, 1) на это же квантовое состояние:

|ψ〉 = 0|000〉 + 0|001〉 + 0|010〉 + 0|011〉 + 1|100〉 + 0|101〉 + 0|110〉 + 0|111〉

CH =

Формируются переходы базисных состояний:

|000〉 => |000〉

|001〉 => |001〉

|010〉 => |010〉

|011〉 => |011〉

|100〉 =>

|101〉 =>

|110〉 =>

|111〉 =>

В этом случае матрица унитарной эволюции будет выглядеть так:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |000〉 | |001〉 | |010〉 | |011〉 | |100〉 | |101〉 | |110〉 | |111〉 |
|  |  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| |000〉 | 0 | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| |001〉 | 1 | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** |
| |010〉 | 2 | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** |
| |011〉 | 3 | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** |
| |100〉 | 4 | **0** | **0** | **0** | **0** |  | **0** |  | **0** |
| |101〉 | 5 | **0** | **0** | **0** | **0** | **0** |  | **0** |  |
| |110〉 | 6 | **0** | **0** | **0** | **0** |  | **0** |  | **0** |
| |111〉 | 7 | **0** | **0** | **0** | **0** | **0** |  | **0** | **-** |

**1.5 Измерение элементов квантовой системы**

Состояние квантовой системы - комплексный вектор. Применяя квантовые операторы, лишь изменяется этот вектор. Но для вычислений нужно преобразовать этот вектор в базовые состояния для обычных компьютеров (0 или 1).

Такой процесс называется измерением кубита в квантовой системе. Измерение включает в себя 2 шага:

1. Формирование вероятностей
2. Выбор вектора и его нормализация

В формировании вероятностей создается 2 вероятности. 1-ая вероятность является суммой квадратов модулей амплитуд вероятностей базисных состояний, в которых выбранный кубит в состоянии 0. 2-ая вероятность является суммой квадратов модулей амплитуд вероятностей базисных состояний, в которых выбранный кубит в состоянии 1.

Далее создается 2 вектора. В 1-ом исключаются все базисные состояния, в которых избранный кубит = 0. Во 2-ом исключаются все базисные состояния, в которых избранный кубит = 1. После получения вероятностей, случайно выбирается состояние кубита. В соответствии с ним выбирается один из этих двух векторов. Нормализуя этот вектор, получается новое состояние квантовой системы.

Пример:

Пусть состояние квантовой системы равно |ψ〉 при n = 2:

|ψ〉 =

i – индекс измеряемого кубита. Пусть i = 0. Тогда формируются 2 вероятности. Prob0 – вероятность 0. Prob1 – вероятность 1. Тогда:

Prob0 =

Prob1 =

Получаются 2 вектора. Пусть вектор |ψ*0* обозначает квантовое состояние в случае, если кубит будет в состоянии 0 и пусть вектор |ψ*1* обозначает квантовое состояние в случае, если кубит будет в состоянии 1. Тогда:

|ψ*0*〉 =

|ψ*1*〉 =

Допустим, что кубит в состоянии 0. Тогда новое состояние квантовой системы будет равно нормированному вектору |ψ*0*〉. Тогда:

|ψ〉 =

**1.6 Нюанс в производительности**

Квантовый компьютер быстрее выполняет параллельные алгоритмы, но классические алгоритм будут выполнятся намного медленнее, ввиду вероятностного принципа. Квантовый компьютер не является универсальной заменой обычному компьютеру. Я предполагаю, что он будет в роли сопроцессора для выполнения определенных задач.

**1.7 Квантовые алгоритмы**

**Алгоритм Шора**

Алгоритм Шора – алгоритм факторизации (разложения числа на простые множители), позволяющий разложить число M на время O(log3M), используя O(log M) логических кубитов. В основе его работы лежит то, что gcd(tr/2 1, M) – простые делители M. t – случайное число от 1 до M, а r – период функции f(x) = tx mod M. Алгоритм в будущем будет способен взломать RSA шифрование, однако на данный момент это невозможно ввиду слишком маленького количества кубит даже на самом мощном квантовом компьютере. Отличительной чертой алгоритма является то, что в нем присутствуют классические вычисления на обычном компьютере.

**Алгоритм Дойча-Йожи**

Алгоритм Дойча-Йожи – алгоритм, который способен выполнять задачу Дойча-Йожи намного быстрее классических компьютеров. Задача Дойча — Йожи заключается в определении, является ли функция нескольких двоичных переменных f(x1,x2,…,xn)постоянной (принимает либо значение 0, либо 1 при любых аргументах) или сбалансированной (для половины области определения принимает значение 0, для другой половины 1). При этом считается априорно известным, что функция либо является константой, либо сбалансирована.

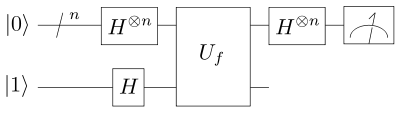


Рисунок 25 – Схема квантового алгоритма Дойча-Йожи

Так выглядит реализация алгоритма Дойча-Йожи с помощью моего симулятора:

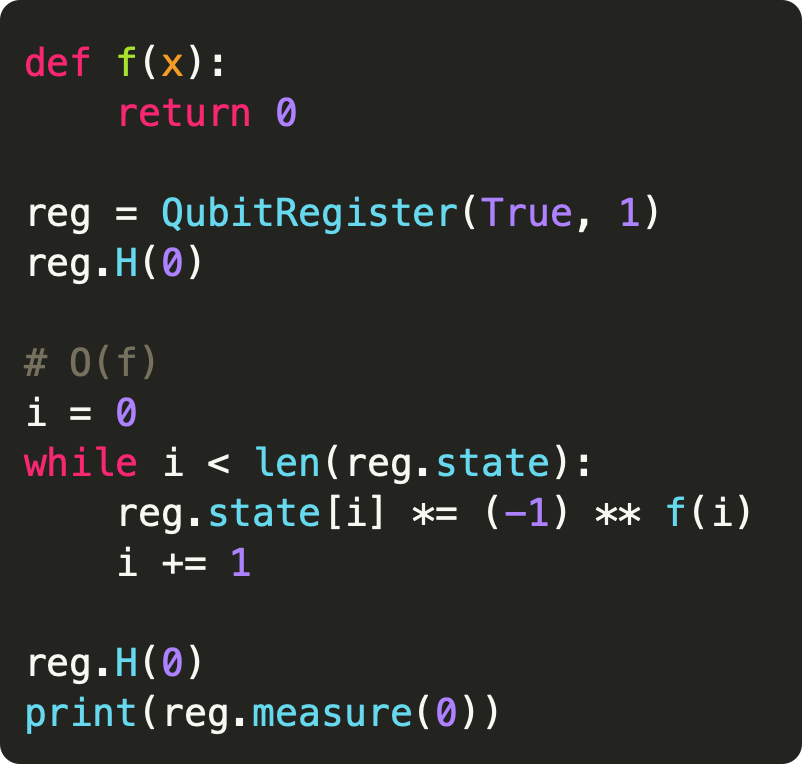


Рисунок 26 – Реализация алгоритм Дойча-Йожи

**Алгоритм Гровера**

Алгоритм Гровера – алгоритм, способный найти решение уравнения f(x) = 1 за обращений у функции f, где f – функция от n переменных, следовательно у классического компьютера поиск решения занял бы N = 2n обращений к функции f.

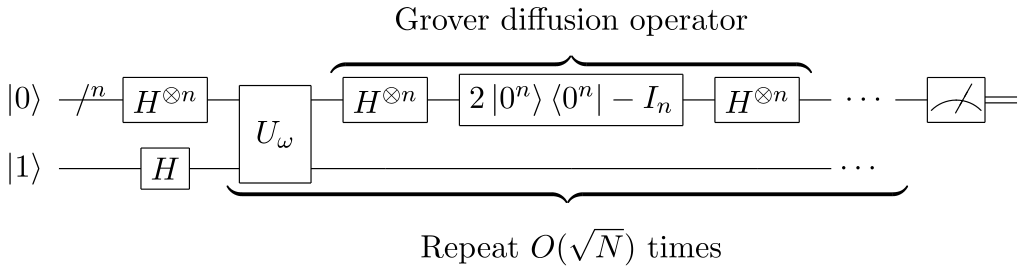


Рисунок 27 – Схема квантового алгоритма Гровера

**“Квантовое превосходство Google”**

В конце августа 2019 года специалисты Google подготовили доклад «Quantum Supremacy Using a Programmable Superconducting Processor» о своем открытии при использовании нового квантового компьютера — достижении квантового превосходства. Материал был опубликован в сентябре 2019 года на сайте НАСА, но позже удален. Дело в том, что Google выбрали задачу запуска квантовых схем, которая естественно будет работать быстрее на квантовом компьютере, чем на его симуляции. Для демонстрации квантового превосходства необходимо использование квантовых алгоритмов, поэтому то, что показали Google это лишь начало для демонстрации полноценного квантового превосходства.

**Глава II. Практика**

**2.1 Тензорное произведение**

Первым шагом к понимаю работы симулятора квантового компьютера является понимания реализации тензорного произведения. Вот так выглядит моя реализация тензорного произведения на низкоуровневом языке С:

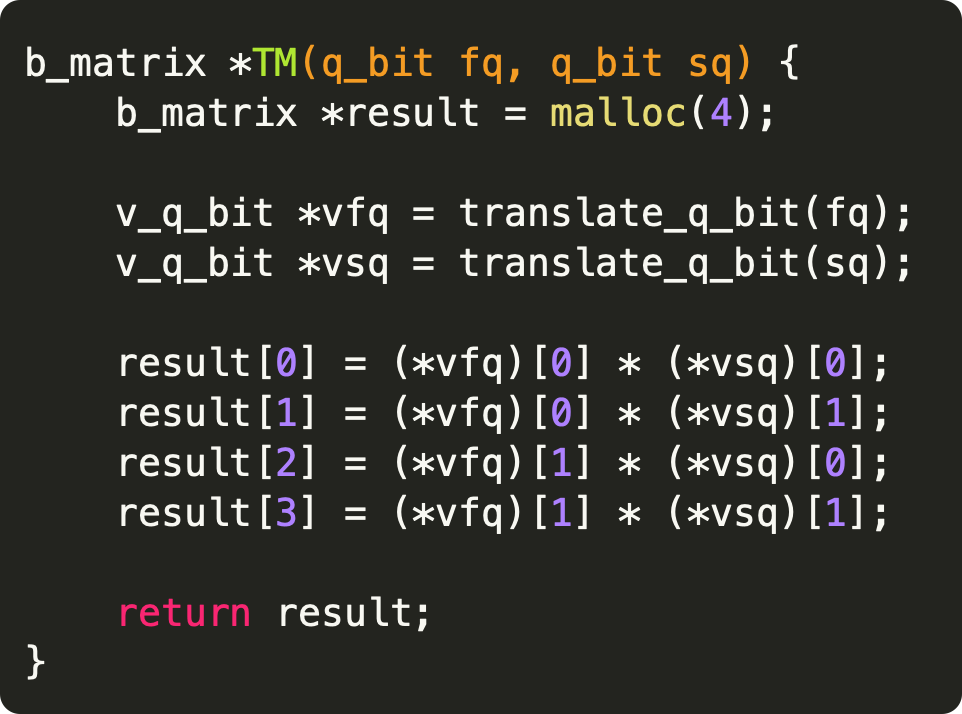
****

Рисунок 5 – Функция тензорного произведения

**2.2 Инициализация симулятора**

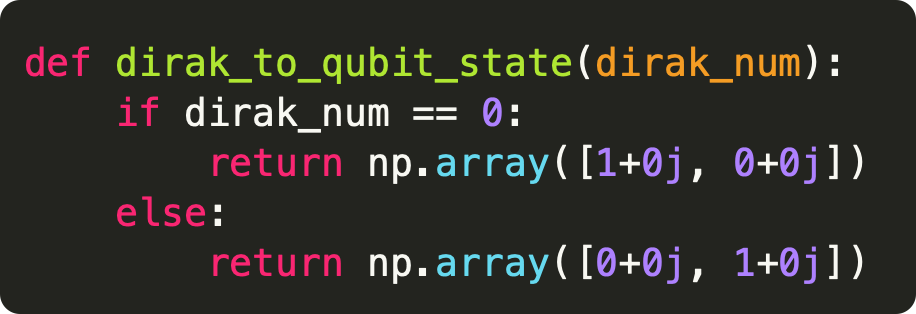


Рисунок 6 – Функция перевода обозначения Дирака в комплексный вектор

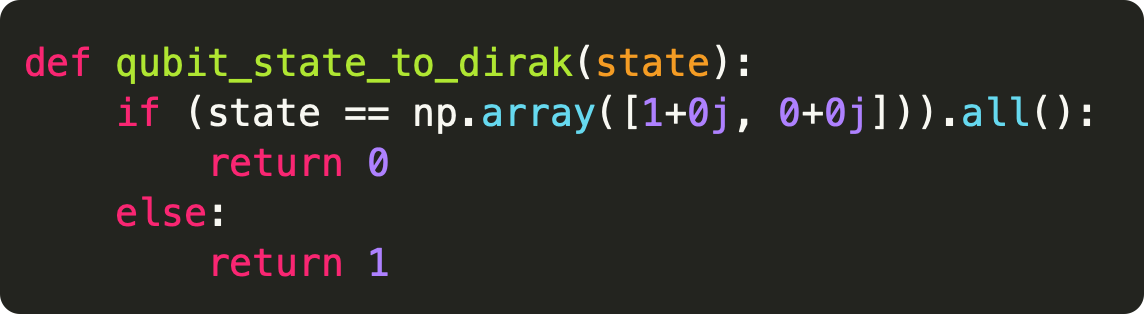


Рисунок 7 – Функция перевода комплексного вектора в обозначение Дирака

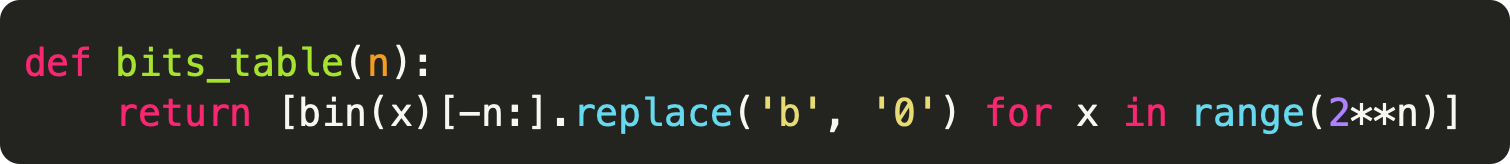


Рисунок 8 – Функция формирования таблицы базисных состояний квантового регистра размером n

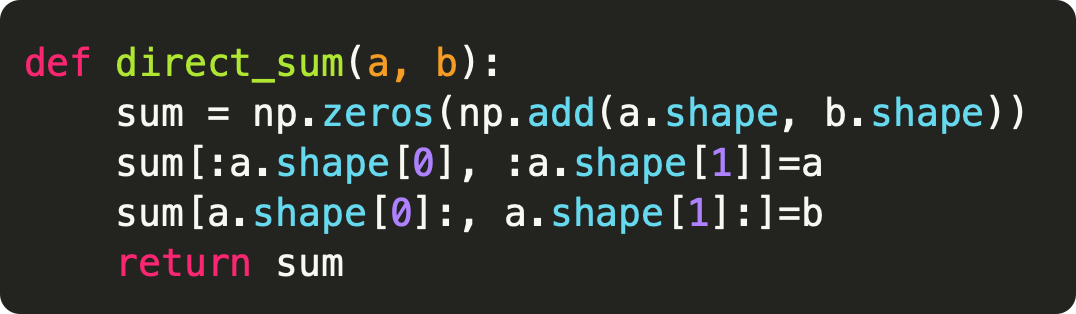


Рисунок 9 – Функция прямой суммы матриц

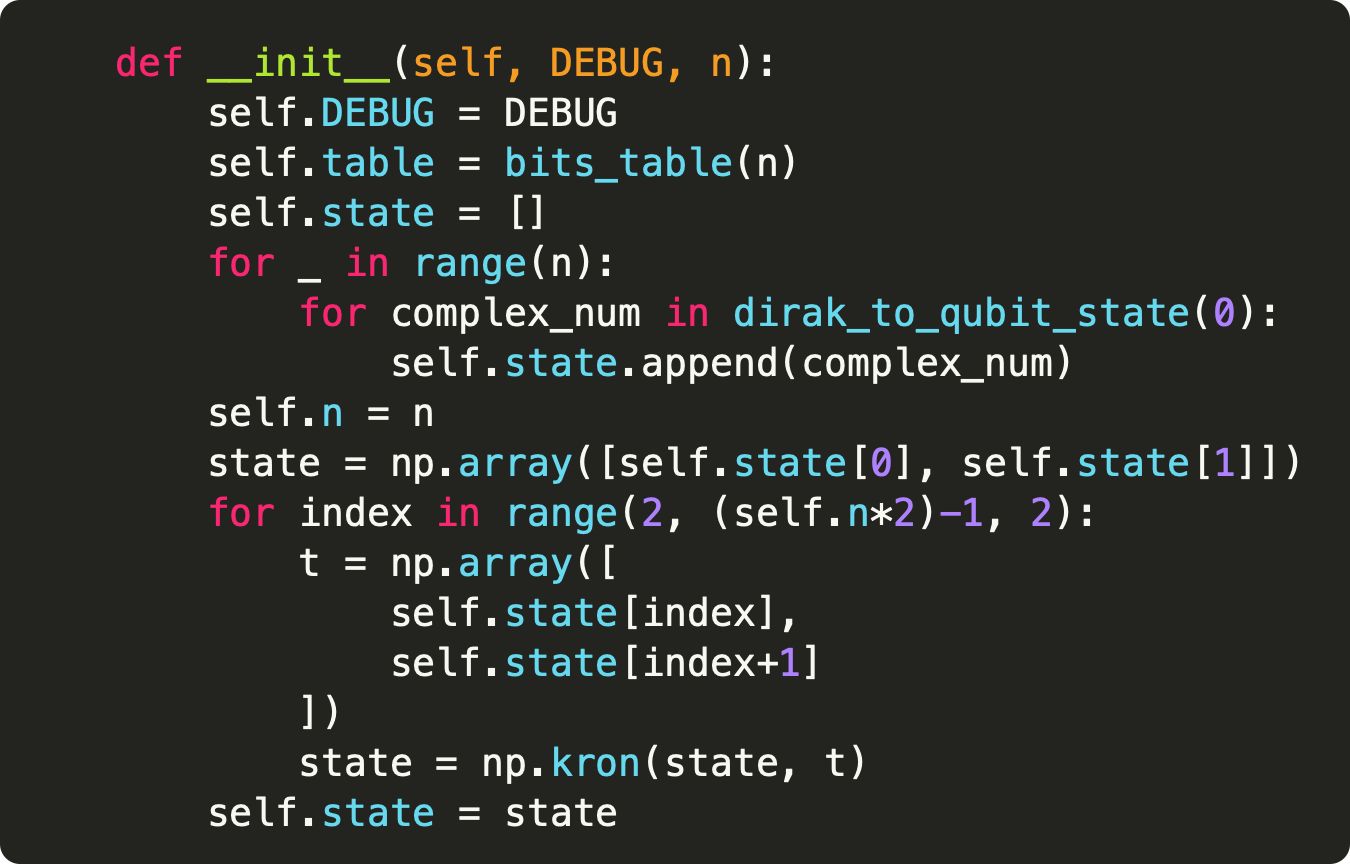


Рисунок 10 – Функция инициализации состояния квантового регистра, как комплексный вектор

**2.3 Преобразование квантовой системы**



Рисунок 11 – Функция формирования и применения матрицы унитарной эволюции

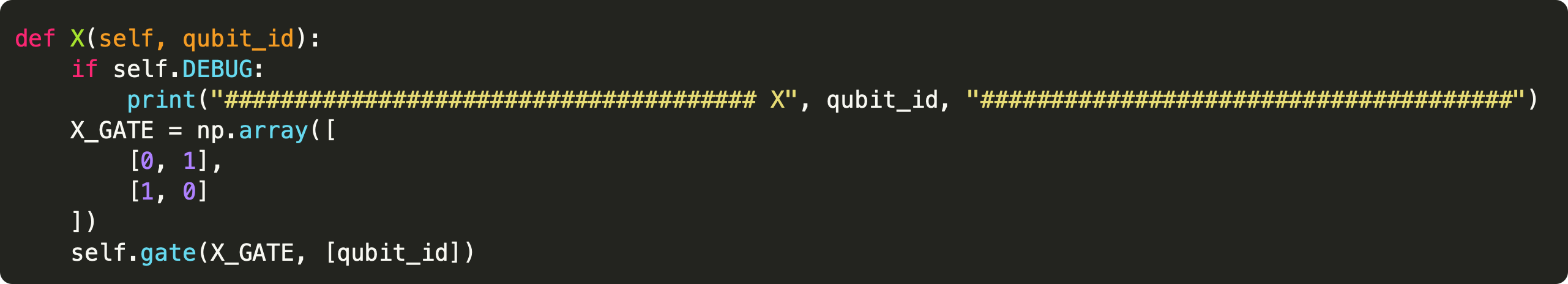


Рисунок 12 – Функция применения квантового оператора X

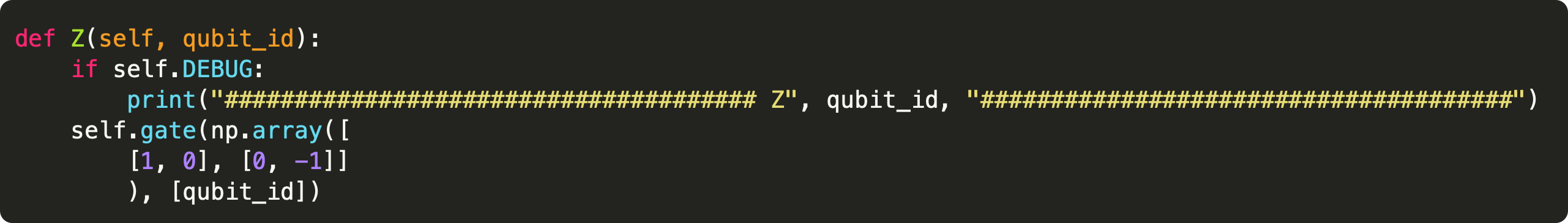


Рисунок 13 – Функция применения квантового оператора Z

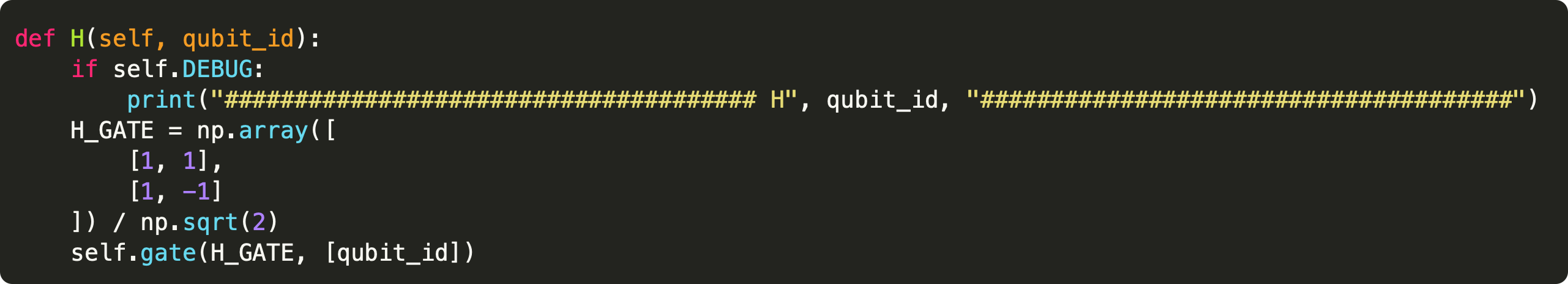


Рисунок 14 – Функция применения квантового оператор H

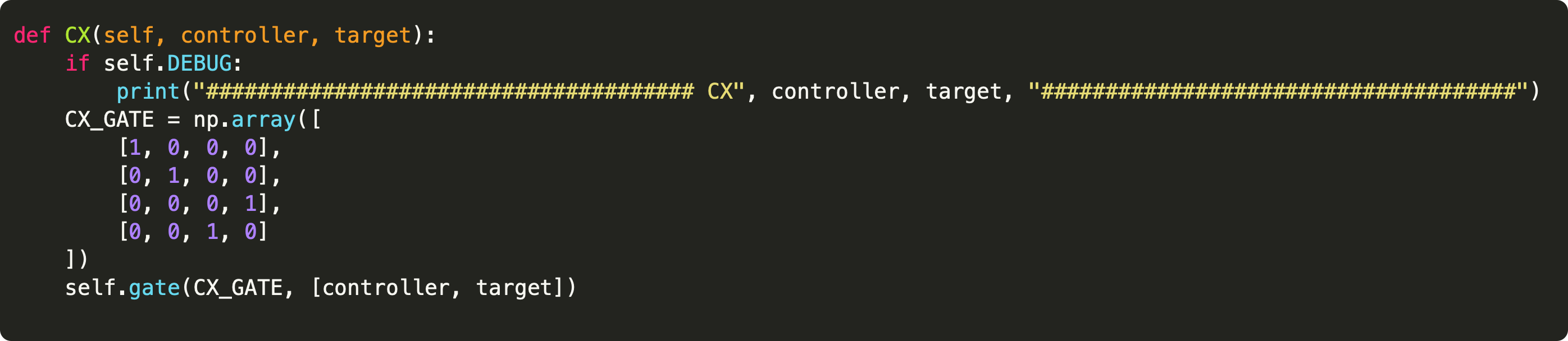


Рисунок 15 – Функция применения квантового оператора CX

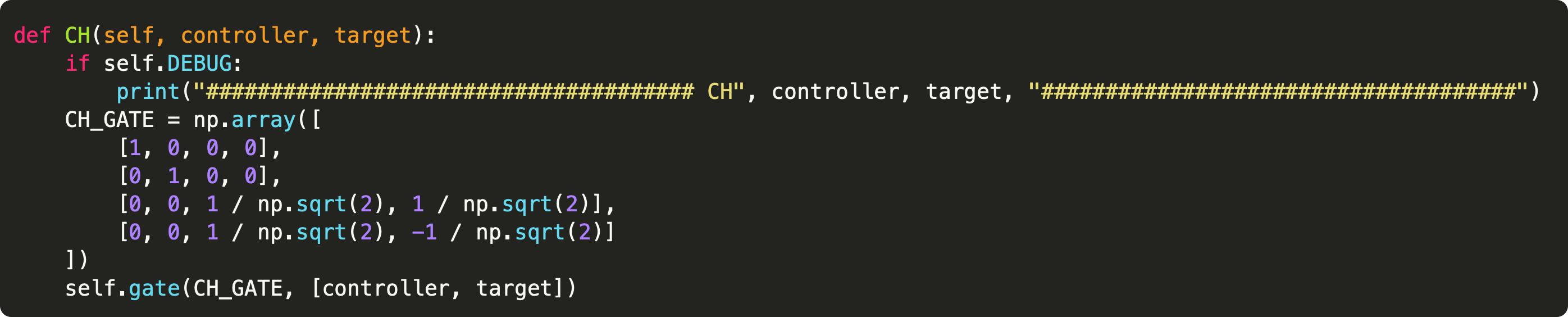


Рисунок 16 – Функция применения квантового оператора CH

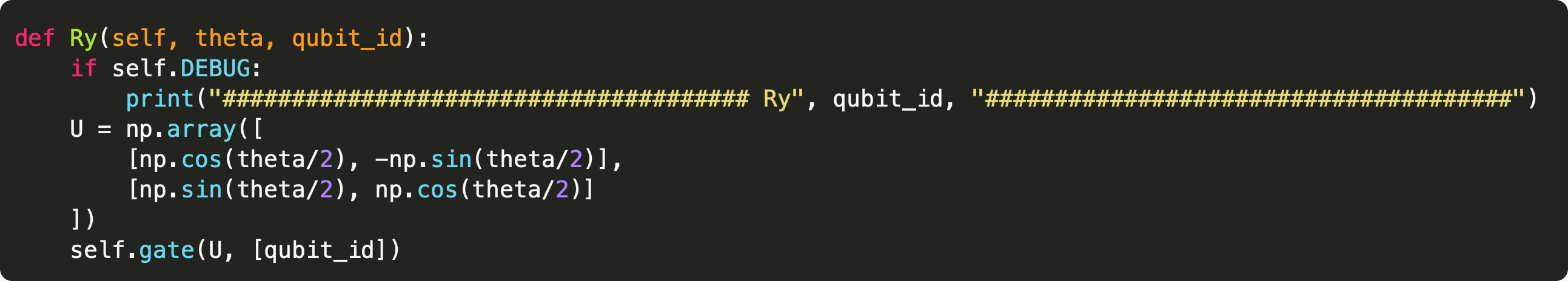


Рисунок 17 – Функция применения квантового оператора Ry

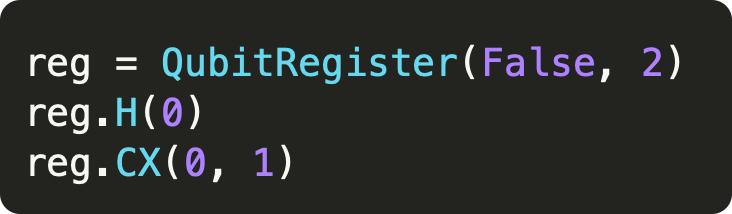


Рисунок 18 – Квантовая суперпозиция в двухкубитной системе

**2.4 Измерение квантовых объектов**

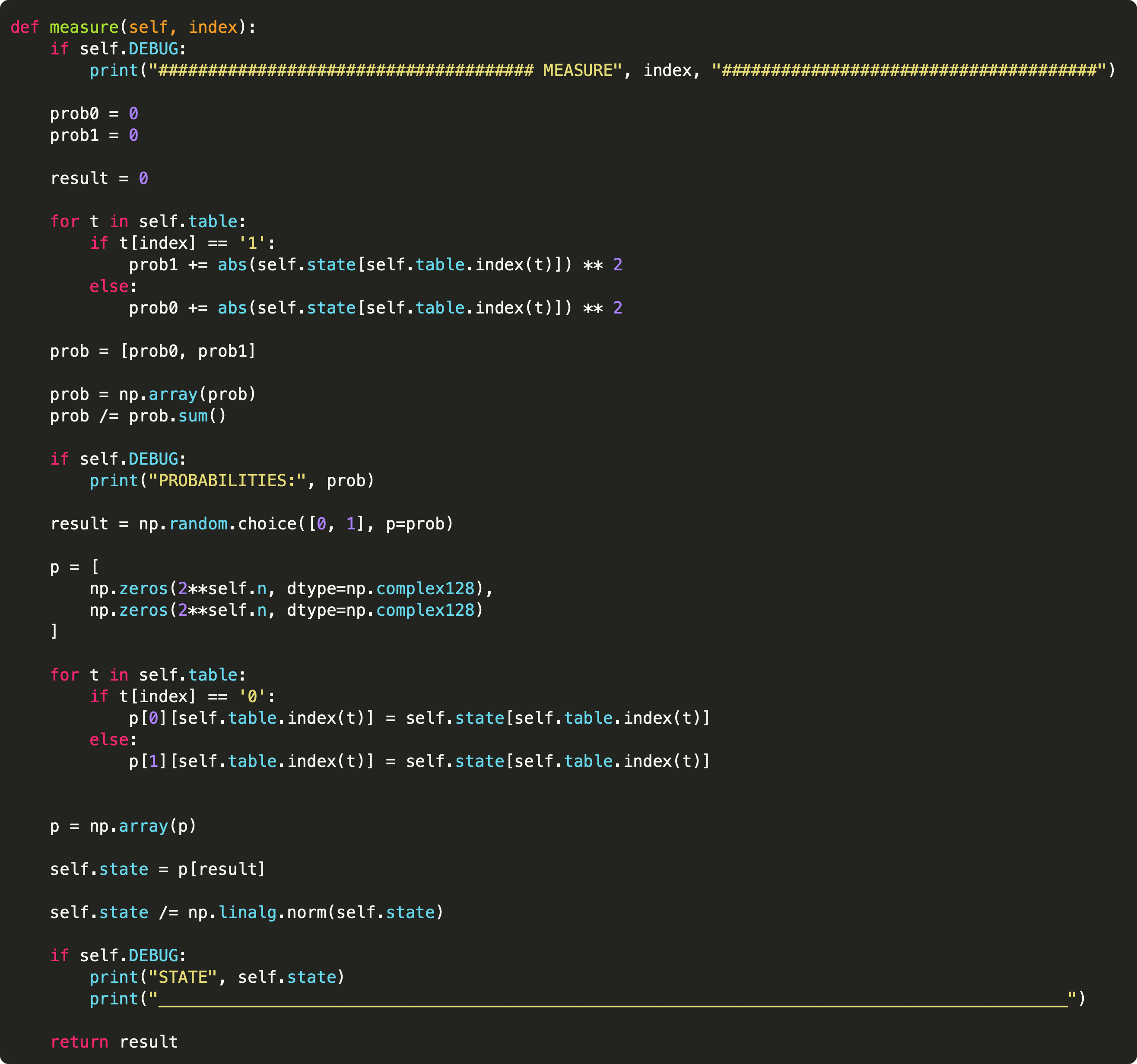


Рисунок 19 – Функция измерения кубита в квантовом регистре

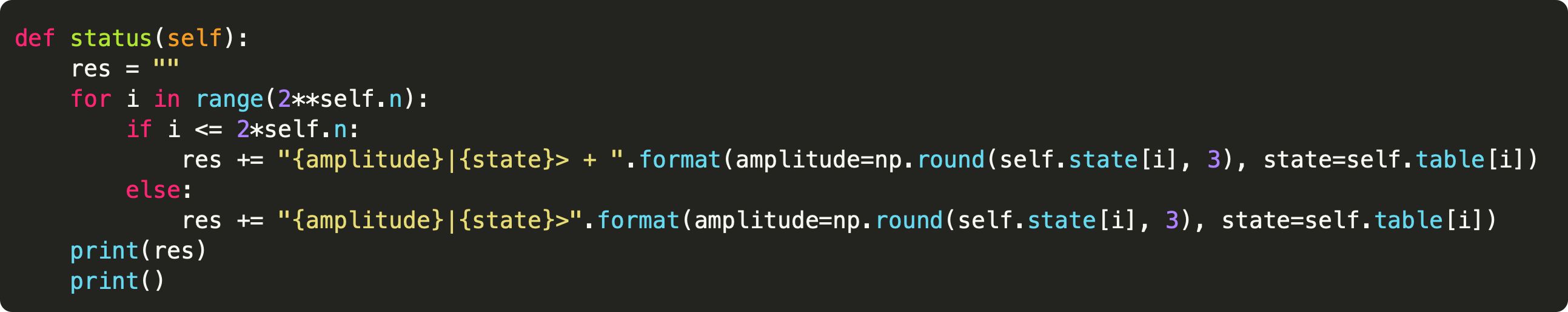


Рисунок 20 – Функция консольного вывода текущего состояния квантовой системы

**2.5 Квантовая телепортация**

Для тестирования симулятора я выбрал самый простой квантовый алгоритм – алгоритм квантовой телепортации. Так выглядит ее схема:

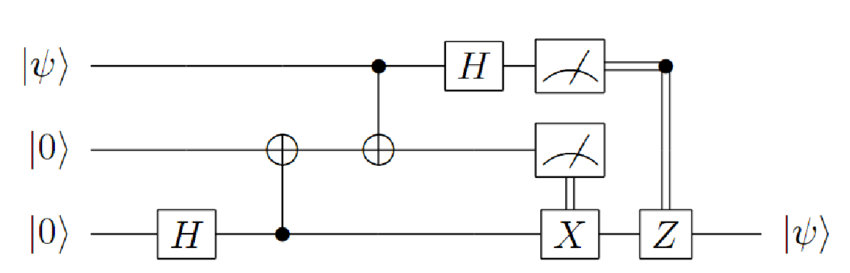


Рисунок 21 – Схема квантовой телепортации

У отправителя есть один кубит, обладающий информацией и второй кубит, который предварительно запутан с кубитом получателя. Отправитель производит измерения всего квантового состояний и передает результаты измерений по классическому каналу связи. После этого отправитель корректирует полученное состояния и получает состояние отправителя.

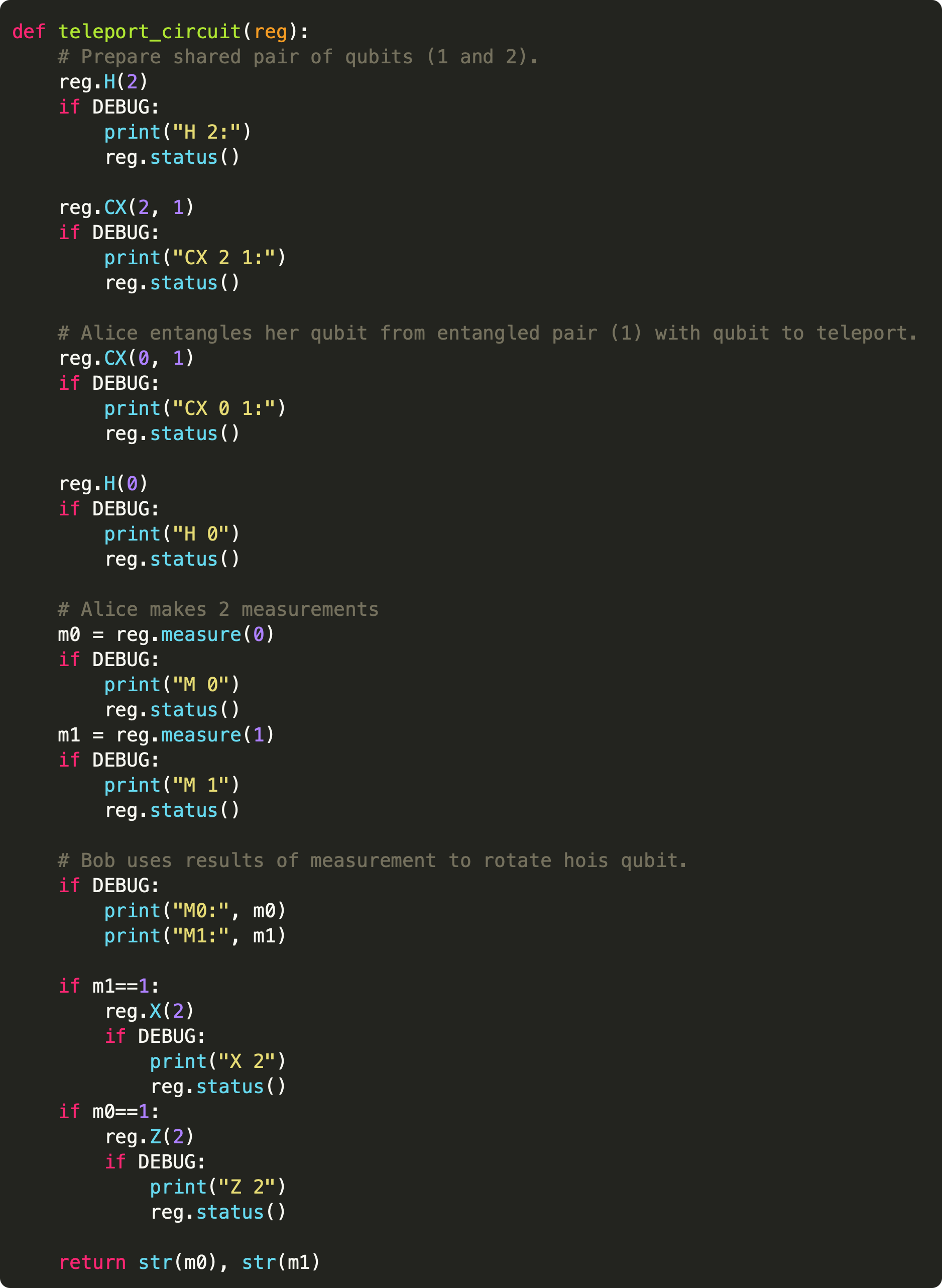


Рисунок 22 – Функция применения схемы квантовой телепортации на квантовый регистр

Функция для тестирования квантовой телепортации должна инициализировать квантовый регистр размером n = 3. Она принимает количество повторений алгоритма и угол, на который кубит будет повернут (информацию).



Рисунок 23 – Функция для тестирования квантовой телепортации

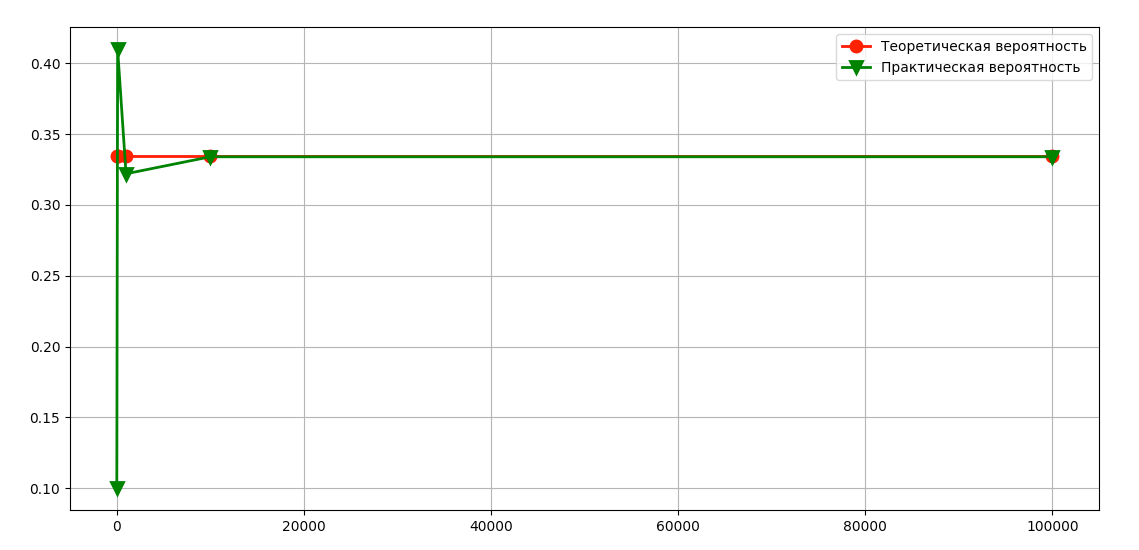
Я протестировал квантовую телепортацию на разных количествах повторений. В качестве показателя эффективности используется разность между предполагаемой и получаемой вероятностью. Данные тестирования представлены на рисунке 24:

Рисунок 24 – Тестирование квантовой телепортации

Из таблицы можно сделать вывод, что эмпирическая вероятность измерить единицу стремится к теоретической, из чего можно сделать вывод, что телепортация произошла успешно.

**Итоги**

**3.1 Сравнение с обычным компьютером**

|  |  |  |
| --- | --- | --- |
|  | **Обычный компьютер** | **Квантовый компьютер** |
| **Логика** | 0 / 1 | α|0〉 + β|1〉 |
| **Физика** | Полупроводниковый транзистор | Квантовый объект |
| **Носитель информации** | Уровни напряжения | Спин |
| **Взаимосвязь** | Полупроводниковый чип | Квантовая запутанность |
| **Алгоритмы** | Стандартные | Специальные (Шора, Гровера и т.д) |
| **Принцип** | Цифровой, детерминированный | Аналоговый, вероятностный |
| **Влияние внешней среды** | Низкое | Высокое |
| **Роль в вычислениях** | Процесс | Сопроцесс |

Таблица 1 – Сравнение квантового компьютера с обычным компьютером

**Заключение**

Выводы:

1. Симулятор квантового компьютера работает корректно ввиду результатов тестирования квантовой телепортации.
2. Квантовое превосходство выражается в успешном использовании квантовых алгоритмов.
3. Квантовый компьютер быстрее выполняет параллельные задачи.
4. Вероятностное устройство квантового компьютера является причиной квантового превосходства
5. Линейные задачи будут выполняться медленнее на квантовом компьютере, чем на обычном компьютере ввиду присутствия процентов ошибок в вероятностном устройстве квантового компьютера.
6. Взлом RSA шифрования квантовым компьютером невозможно на данный момент ввиду длины ключа RSA шифрования.

Итоги:

Прирост в производительности у квантового компьютера доступен лишь при использовании специальных алгоритмов. Квантовый компьютер не является универсальной заменой обычному компьютеру.

**Список использованных источников и литературы**

* Исследовано [Wikipedia, Quantum Computing]: the free encyclopedia. Режим доступа к энциклопедии: <https://en.wikipedia.org/wiki/Quantum_computing>
* Исследовано [Wikipedia, Quantum Logic Gate]: the free encyclopedia. Режим доступа к энциклопедии: <https://en.wikipedia.org/wiki/Quantum_logic_gate>
* Исследовано [Quantum Computing: Lecture Notes]: Ronald de Wolf, QuSoft, CWI and University of Amsterdam. Режим доступа к заметкам лекций: <https://homepages.cwi.nl/~rdewolf/qcnotes.pdf>
* Исследовано [Quantum Mechanics and Quantum Computation]: BerkeleyX CS-191x. Режим доступа к курсу: <https://www.edx.org/course/quantum-mechanics-and-quantum-computation>
* Исследовано [Quantum Gate Decomposition Algorithms]: Alexander Slepoy. Режим доступа к книге: <https://core.ac.uk/download/pdf/193939351.pdf>
* Беклемишев Д.В. – 13 изд – “Курс аналитической геометрии и линейной алгебры”: Санкт-Петербург, Москва, Краснодар: 2015.
* Mikio Nakahara, Tetsuo Ohmi – 1st Edition– “Quantum Computing. From Linear Algebra to Physical Realizations”: CRC Press, Taylor & Francis Group, 6000 Broken Sound Parkway NW, 300.
* Исследовано [Сколько денег страны тратят на квантовые технологии]: РБК. Режим доступа к cтатье: <https://trends.rbc.ru/trends/industry/606ad5239a79474c50841023>
* Исследовано [Пора задуматься об инвестировании в квантовые технологии – какие у них перспективы]: РБ. Режим доступа к cтатье: <https://rb.ru/opinion/kvantovye-tehnologii/>