

4

Квантовая телепортация

В этой главе будет представлена программа для QPU, позволяющая немедленно телепортировать объект¹ на расстояние 3,1 мм! А при наличии правильного оборудования тот же код будет работать и на межзвездных расстояниях.

Хотя слово «телепортация» может ассоциироваться с магией и фокусами, вы увидите, что разновидность квантовой телепортации, выполняемая при помощи QPU, выглядит столь же впечатляюще, но намного более практично — собственно, она является важнейшим концептуальным компонентом программирования QPU.

Практический пример: первые шаги в телепортации

Если вы хотите освоить телепортацию, лучше всего опробовать ее в деле. Учтите, что на протяжении всей человеческой истории до момента написания книги лишь несколько тысяч людей выполняло физическую телепортацию какого-либо рода, так что даже выполнение следующего кода ставит вас в число первопроходцев.

Для данного примера вместо системы моделирования мы воспользуемся реальным пятикубитным QPU компании IBM (рис. 4.1). Вы сможете скопировать пример кода из листинга 4.1 на сайт IBM Q Experience, щелкнуть на кнопке и убедиться в том, что телепортация прошла успешно.

¹ В данном случае телепортируется квантовое состояние объекта, а не сам объект. Квантовая телепортация — это обмен информацией, а не перемещение. — *Примеч. науч. ред.*

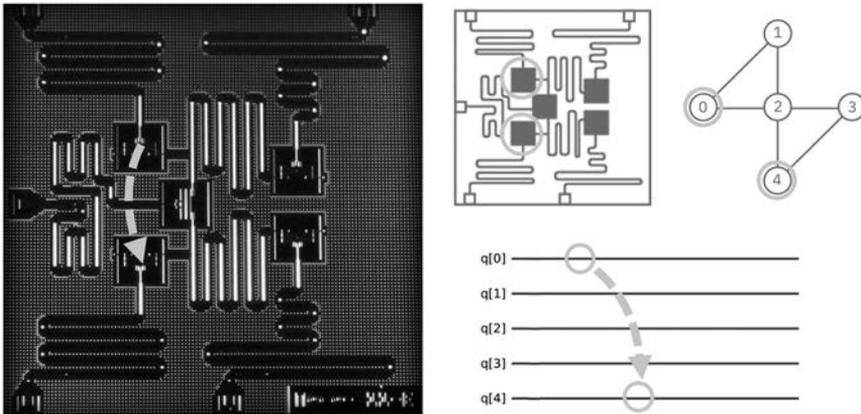


Рис. 4.1. Микросхема IBM очень мала, так что перемещение кубита будет довольно коротким; на иллюстрации и схеме выделены части QPU, между которыми будет происходить телепортация¹

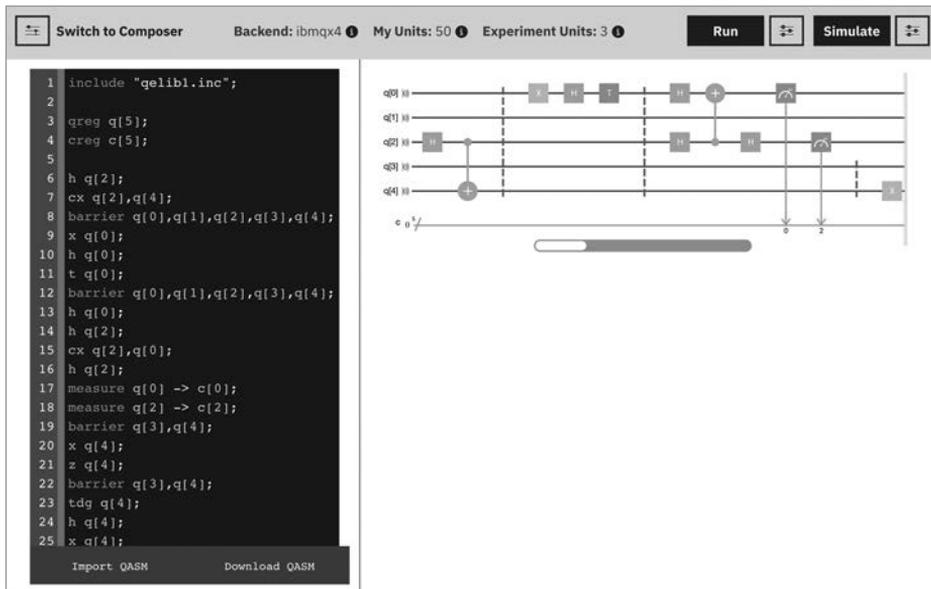


Рис. 4.2. Редактор IBM Q Experience (QX) OpenQASM²

¹ Публикуется с разрешения International Business Machines Corporation, © International Business Machines Corporation.

² Публикуется с разрешения International Business Machines Corporation, © International Business Machines Corporation.

Для программирования IBM Q Experience можно использовать OpenQASM¹ и Qiskit². Обратите внимание: в листинге 4.1 содержится не код JavaScript, предназначенный для выполнения в QSEngine, а код OpenQASM, предназначенный для выполнения в облачном интерфейсе IBM (рис. 4.2). Выполнение этого кода позволит вам не смоделировать, а реально *выполнить* телепортацию кубита в исследовательском центре IBM в Йорктаун Хайтс (Нью-Йорк). Мы подробно опишем, как это делается. Подробное рассмотрение этого кода также поможет вам точно понять, как работает квантовая телепортация.

Пример кода

Этот пример можно выполнить онлайн по адресу <http://oreilly-qc.github.io?p=4-1>.

Листинг 4.1. Телепортация с проверкой

```
include "qelib1.inc";
qreg q[5];
creg c[5];

// Шаг 1: создать запутанную пару
h q[2];
cx q[2],q[4];
barrier q[0],q[1],q[2],q[3],q[4];

// Шаг 2: подготовить данные
x q[0];
h q[0];
t q[0];
barrier q[0],q[1],q[2],q[3],q[4];

// Шаг 3: отправить
h q[0];
h q[2];
cx q[2],q[0];
h q[2];
measure q[0] -> c[0];
measure q[2] -> c[2];
barrier q[3],q[4];

// Шаг 4: получить
x q[4];
z q[4];
barrier q[3],q[4];
```

¹ OpenQASM — квантовый язык ассемблера, поддерживаемый IBM Q Experience.

² Qiskit — пакет разработки с открытым кодом для работы с квантовыми процессорами IBM Q.

```
// Шаг 5: проверить
tdg q[4];
h q[4];
x q[4];
measure q[4] -> c[4];
```

Прежде чем углубляться в подробности, сначала проясним ряд моментов. Под *квантовой телепортацией* мы подразумеваем возможность переноса точного состояния (то есть амплитуд и относительных фаз) из одного кубита в другой. Задача — взять всю информацию, содержащуюся в одном кубите, и поместить ее в другой кубит. Напомним, что квантовая информация не может реплицироваться; следовательно, информация в первом кубите обязательно уничтожается при телепортации во второй кубит. Так как квантовое описание является самым точным описанием физического объекта, в действительности именно такое явление мы называем телепортацией, только на квантовом уровне.

Разобравшись с этим, займемся телепортацией! В учебниках введение в квантовую телепортацию обычно начинается с истории, которая выглядит примерно так: пара кубитов, находящихся в *запутанном* состоянии, находится в распоряжении двух сторон, Алисы и Боба (физики обожают имена в алфавитном порядке). Запутанные кубиты — ресурс, который используется Алисой для телепортации состояния некоторого *другого* кубита Бобу. Таким образом, в телепортации задействованы три кубита: кубит *данных*, который хочет телепортировать Алиса, и запутанная пара кубитов, которые есть у нее и Боба (эти кубиты образуют некое подобие квантового кабеля Ethernet). Алиса готовит свои данные, а затем при помощи операций **HAD** и **CNOT** она запутывает кубит данных с другим кубитом (который, в свою очередь, уже запутан с кубитом Боба). Затем она разрушает свой кубит данных и запутанный кубит операциями **READ**. Результаты этих операций **READ** дают два традиционных бита информации, которые она отправляет Бобу. Так как на этот раз речь идет о *битах*, а не о кубитах, можно воспользоваться традиционным кабелем Ethernet. Используя эти два бита, Боб выполняет некие однокубитные операции со своей половиной запутанной пары, которая изначально принадлежала ему и Алисе — и получает кубит данных, который собиралась отправить Алиса.

Прежде чем перейти к описанному здесь¹ протоколу квантовых операций, необходимо разобраться с одной проблемой. *«Постойте-ка... эта часть рождена нашим воображением. Если Алиса отправляет Бобу традицион-*

¹ Полный исходный код версий этого примера для QASM и QCEngine доступен по адресу <http://oreilly-qc.github.io?p=4-1>.

ную информацию по кабелю Ethernet, — продолжаете вы, — то такая телепортация не особо впечатляет». Отличное наблюдение! Успех квантовой телепортации действительно зависит от передачи традиционных (цифровых) битов. Мы уже видели, что амплитуды и относительные фазы должны полностью описывать произвольное состояние, которое может принимать кубит в континууме значений. Принципиально здесь то, что протокол телепортации работает даже в том случае, когда Алиса не знает состояние своего кубита. И это особенно важно, потому что невозможно определить амплитуду и относительную фазу одного кубита в неизвестном состоянии. Тем не менее — с помощью запутанной пары кубитов — для эффективной передачи точной конфигурации кубита Алисы (независимо от его комплексных амплитуд) достаточно всего двух традиционных битов. Эта конфигурация будет верной при потенциально бесконечном количестве битов точности!

Как же добиться этого волшебного эффекта? Ниже описан весь протокол. На рис. 4.3 изображены операции, которые должны быть выполнены с тремя задействованными кубитами. Все эти операции были представлены в главах 2 и 3.

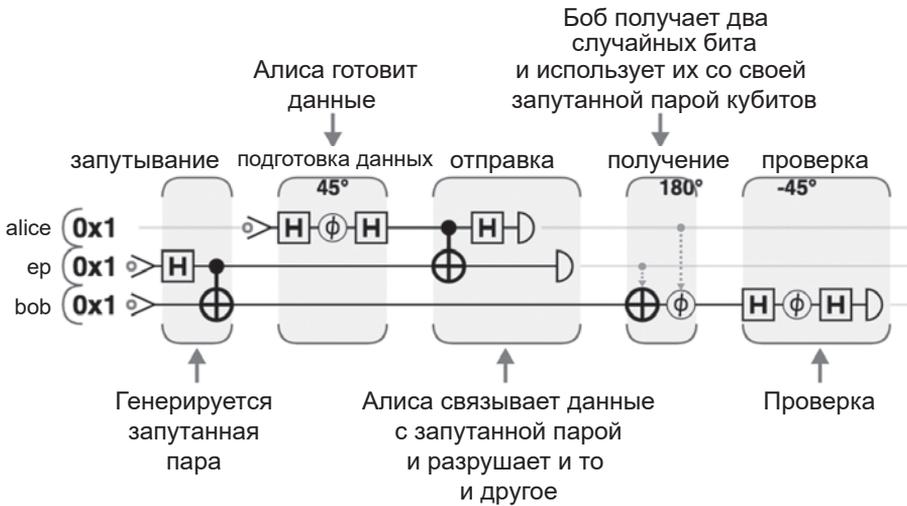


Рис. 4.3. Полная схема телепортации: на стороне Алисы находятся кубиты alice и er, а на стороне Боба — кубит bob

Если вставить код из этой схемы из листинга 4.1 в систему IBM QX, пользовательский интерфейс IBM покажет схему на рис. 4.4. Обратите внимание: это точно такая же программа, как на рис. 4.3, только представленная несколько иным образом. Система обозначений с квантовыми вентилями,

которую мы использовали, считается стандартизированной; несомненно, она встретится вам за пределами книги¹.

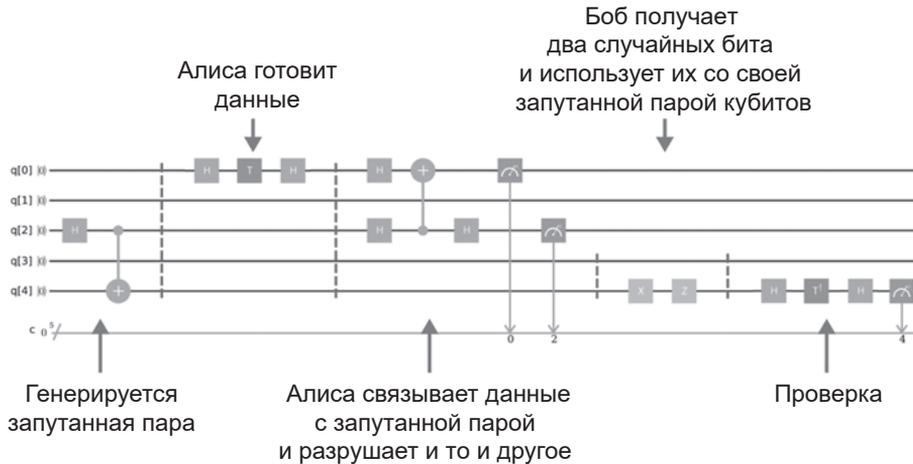


Рис. 4.4. Схема телепортации в IBM QX²

При нажатии кнопки Run IBM выполнит вашу программу 1024 раза (это число можно настроить), а затем выведет статистику по запуску. После выполнения программы результат будет выглядеть примерно (но не в точности) так, как показано на гистограмме на рис. 4.5.

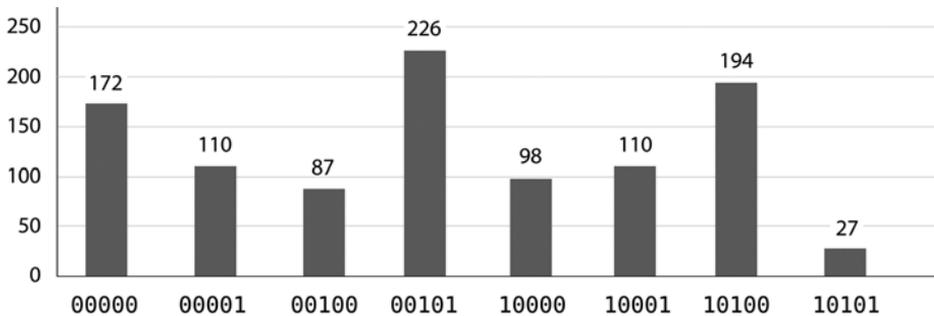


Рис. 4.5. Результаты запуска программы (успешная телепортация?)

¹ Такие вентили, как CNOT и HAD, могут комбинироваться разными способами для получения того же результата. Для некоторых операций в IBM QX и QCEngine используются разные разложения.

² Публикуется с разрешения International Business Machines Corporation, © International Business Machines Corporation.

Успех? Возможно! Чтобы продемонстрировать, как читать и интерпретировать эти результаты, мы подробнее разберем каждый шаг программы QPU, используя круговую запись для наглядного представления всего, что происходит с нашими кубитами¹.



На момент написания книги схемы и результаты, отображаемые IBM QX, показывают, что происходит со всеми пятью кубитами, доступными для QPU, даже если они не использовались. Это объясняет, почему в схеме на рис. 4.4 присутствуют две пустые линии кубитов и почему столбцы с результатами для каждого вывода на рис. 4.5 помечаются пятиразрядными двоичными числами (вместо трехразрядных) — несмотря на то что фактически показаны только столбцы, соответствующие $2^3 = 8$ возможным конфигурациям трех использованных кубитов.

Анализ программы

Так как в нашем примере телепортации используются три кубита, для их полного описания потребуются $2^3 = 8$ кругов (по одному для каждой возможной комбинации 3 битов). Мы расположим эти восемь кругов в две строки, что поможет вам понять, как операции влияют на три составляющих кубита. На рис. 4.6 все строки и столбцы кругов помечены конкретными значениями каждого кубита. Чтобы проверить правильность меток, обратитесь к двоичному значению соответствующего регистра.

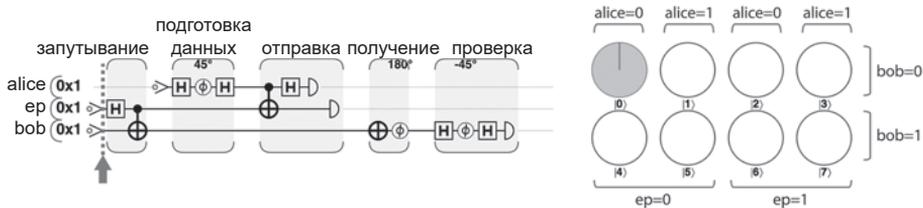


Рис. 4.6. Полная программа телепортации с проверкой



При работе с многокубитными регистрами круговая запись часто размещается по строкам и столбцам, как на рис. 4.6. Такое размещение всегда помогает быстрее выявить закономерности в поведении отдельных кубитов: вы всегда сразу замечаете важные строки или столбцы.

В начале программы все три кубита инициализируются в состоянии $|0\rangle$, как видно из рис. 4.6, — единственным возможным значением является значение, при котором $alice=0$, $ep=0$ и $bob=0$.

¹ Полный исходный код этого примера доступен по адресу <http://oreilly-qc.github.io?p=4-1>.

Шаг 1: создание запутанной пары

Первым шагом телепортации должна стать установка связи через запутанность. Комбинация HAD и CNOT, решающая эту задачу, была использована в главе 3 для создания запутанного состояния двух кубитов — так называемой *пары Белла*. Из круговой записи на рис. 4.7 видно, что при чтении bob и ep значения будут случайными с распределением 50/50, но они будут гарантированно совпадать друг с другом.

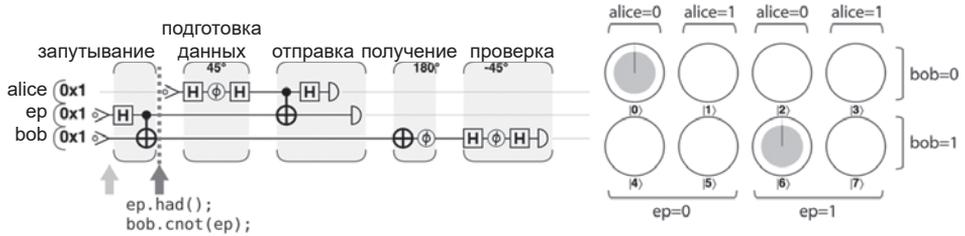


Рис. 4.7. Шаг 1: создание запутанной пары

Шаг 2: подготовка данных

После того как связь квантовой запутанности будет установлена, Алиса может подготовить данные для отправки. Конечно, способ подготовки зависит от природы (квантовой) информации, которую Алиса хочет отправить Бобу. Она может записать значение в кубит данных, ввести в состояние запутанности с другими данными QPU или даже получить из предыдущих вычислений в совершенно отдельной части QPU.

Для нашего примера мы разочаруем Алису и предложим ей подготовить особенно простой кубит данных с использованием операций HAD и PHASE. Преимуществом такого решения является получение данных с легко расшифровываемой закономерностью круговой записи (рис. 4.8).

Мы видим, что кубиты bob и ep продолжают зависеть друг от друга (только круги, соответствующие кубитам bob и ep с одинаковыми значениями, имеют ненулевые амплитуды).

Также видно, что значение alice не зависит ни от одного из двух других кубитов; более того, в результате подготовки данных был получен кубит с вероятностями 85,4% $|0\rangle$ и 14,6% $|1\rangle$ и относительной фазой -90° (круги, соответствующие alice=1, повернуты на 90° по часовой стрелке относительно кругов alice=0, что соответствует отрицательным значениям по нашим соглашениям).

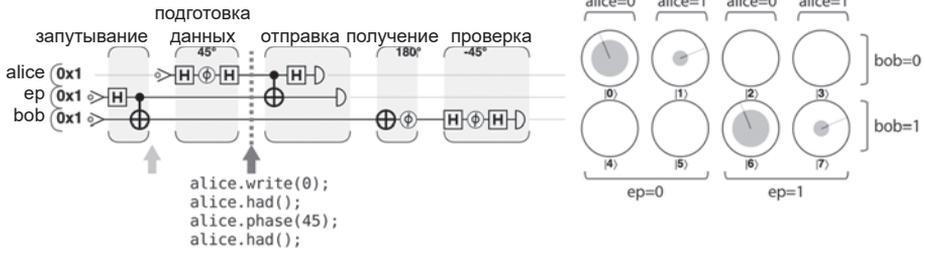


Рис. 4.8. Шаг 2: подготовка данных

Шаг 3.1: связывание данных с запутанной парой

В главе 2 было показано, что условная природа операции CNOT позволяет задействовать состояния двух кубитов. Теперь Алиса использует этот факт, чтобы запутать кубит данных со своей половиной запутанной пары (другая половина находится у Боба). В круговой записи это действие меняет круги местами, как показано на рис. 4.9.

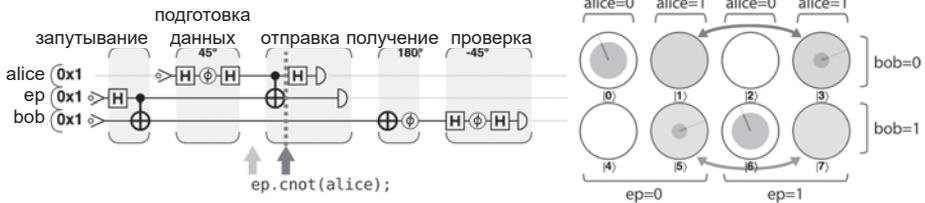


Рис. 4.9. Шаг 3.1: связывание данных с запутанной парой

Появление *множественных* запутанных состояний усложняет ситуацию, поэтому стоит пояснить происходящее. И у Алисы, и у Боба уже имеется один из двух запутанных кубитов (полученных на шаге 1). Теперь Алиса запутывает другой кубит (данные) со своей половиной (уже запутанной) пары. На интуитивном уровне понятно, что Алиса в каком-то смысле косвенно связала свои данные с половиной запутанной пары, принадлежащей Бобу, хотя ее кубит данных остался неизменным. Результаты READ для ее данных будут логически связаны с этими двумя другими кубитами. Эта связь видна в круговой записи, поскольку состояние регистра QPU на рис. 4.9 содержит только те конфигурации, у которых результат XOR всех трех кубитов равен 0. Ранее это условие выполнялось для ep и bob, но теперь оно выполняется для всех трех кубитов, образующих *трехкубитную* запутанную группу.