

Оглавление

От издательства	14
Предисловие	15
Структура книги	15
Типографские соглашения.....	17
Благодарности.....	17
Глава 1. Введение.....	19
Необходимая подготовка	19
Что такое QPU?	21
Практический подход	22
Учебник QCEngine	22
Отладка	24
Низкоуровневые команды QPU	25
Ограничения моделирования	28
Аппаратные ограничения	28
QPU и GPU: некоторые общие характеристики.....	29

ЧАСТЬ I. ПРОГРАММИРОВАНИЕ ДЛЯ QPU

Глава 2. Один кубит.....	32
Краткий обзор физических кубитов	34
Круговая запись	37
Размер круга	38
Поворот	39
Первые операции QPU.....	41
Команда QPU: NOT	41

Команда QPU: HAD	42
Команда QPU: READ	43
Команда QPU: WRITE	43
Практический пример: идеально случайный бит	45
Пример кода	46
Пример кода	48
Команды QPU: PHASE(θ)	48
Команды QPU: ROTX(θ) и ROTY(θ)	49
COPY: недостающая операция	50
Объединение операций QPU	50
Команда QPU: ROOT-NOT	51
Пример кода	52
Практический пример: квантовая проверка защиты	53
Пример кода	54
Итоги	57

Глава 3. Группы кубитов 58

Круговая запись для многокубитных регистров	58
Пример кода	60
Изображение многокубитного регистра	61
Однокубитные операции в многокубитных регистрах	62
Чтение кубита в многокубитном регистре	64
Наглядное представление большого количества кубитов	65
Команды QPU: CNOT	67
Практический пример: использование пар Белла для реализации совместной случайности	70
Пример кода	71
Команды QPU: CPHASE и CZ	71
Приемы программирования QPU: фазовый откат	73
Пример кода	75
Команда QPU: CCNOT (вентиль Тоффоли)	75
Команды QPU: SWAP и CSWAP	76
Проверка обменом	77
Пример кода	78
Построение произвольной условной операции	81

Пример кода	82
Практический пример: дистанционно управляемая случайность.....	84
Пример кода	84
Итоги.....	87
Глава 4. Квантовая телепортация	88
Практический пример: первые шаги в телепортации.....	88
Пример кода	90
Анализ программы.....	94
Шаг 1: создание запутанной пары.....	95
Шаг 2: подготовка данных.....	95
Шаг 3.1: связывание данных с запутанной парой.....	96
Шаг 3.2: перевод кубита данных в суперпозицию	97
Шаг 3.3: чтение обоих кубитов Алисы	97
Шаг 4: получение и преобразование	98
Шаг 5: проверка результата	99
Интерпретация результатов	101
Как используется телепортация?	102
Известные проблемы при телепортации	102

ЧАСТЬ II. ПРИМИТИВЫ QPU

Глава 5. Квантовая арифметика и логика	106
Странно и необычно.....	106
Арифметика с QPU.....	108
Практический пример: построение операторов инкремента и декремента....	109
Пример кода	110
Сложение двух квантовых целых чисел	112
Пример кода	113
Отрицательные числа.....	114
Практический пример: более сложные вычисления.....	115
Пример кода	116
Переход на квантовый уровень	116
Квантовое условное выполнение.....	116
Пример кода	117

Фазовое кодирование результата	118
Пример кода	118
Обратимость и служебные кубиты	119
Отмена вычислений	122
Отображение булевой логики на операции QPU	125
Базовая квантовая логика	126
Пример кода	127
Итоги	128
Глава 6. Усиление комплексной амплитуды	129
Практический пример: преобразование между фазой и амплитудой	129
Пример кода	130
Итерация усиления комплексной амплитуды	132
Больше итераций?	133
Пример кода	134
Инвертирование нескольких элементов	136
Пример кода	137
Использование усиления комплексной амплитуды	142
AA и QFT при оценке суммы	142
Ускорение традиционных алгоритмов с применением AA	143
Внутри QPU	143
Интуитивное понимание	143
Итоги	146
Глава 7. QFT: квантовое преобразование Фурье	147
Скрытые закономерности	147
Пример кода	148
QFT, DFT и FFT	149
Частоты в регистре QPU	150
Пример кода	151
Пример кода	154
DFT	154
Вещественные и комплексные входные данные для DFT	156
Подробнее о DFT	158
Пример кода	160
Практическое применение QFT	163

Скорость QFT	163
Пример кода	167
Пример кода	167
Пример кода	168
Внутри QPU	169
Интуитивное объяснение.....	170
Последовательность операций	172
Пример кода	173
Итоги.....	176
Глава 8. Квантовая оценка фазы	177
Получение информации об операциях QPU	177
Собственные фазы предоставляют полезную информацию	178
Что делает оценка фазы.....	180
Как пользоваться оценкой фазы.....	180
Ввод.....	182
Пример кода	182
Вывод.....	184
Предостережения	184
Выбор размера выходного регистра	185
Сложность.....	186
Условные операции	186
Оценка фазы на практике.....	186
Внутри QPU	187
Пример кода	187
Интуитивное объяснение.....	188
Операция за операцией.....	190
Итоги.....	192

ЧАСТЬ III. ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ QPU

Глава 9. Реальные данные.....	194
Нецелые данные.....	195
QRAM	196

Пример кода	198
Пример кода	200
Кодирование векторов	201
Пример кода	203
Ограничения комплексного амплитудного кодирования	204
Комплексное амплитудное кодирование и круговая запись	206
Кодирование матриц	207
Как матрицы могут представляться операциями QPU?	208
Квантовое моделирование	209
Глава 10. Квантовый поиск	215
Фазовая логика	216
Построение элементарных операций фазовой логики	218
Построение сложных команд фазовой логики	219
Пример кода	221
Решение логических головоломок	222
О котятках и тиграх	223
Пример кода	226
Общий рецепт для решения задач выполнимости булевых формул	227
Практический пример: задача выполнимости 3-SAT	229
Пример кода	229
Практический пример: невыполнимая задача 3-SAT	231
Пример кода	232
Ускорение традиционных алгоритмов	234
Глава 11. Квантовая избыточная выборка	236
Применение QPU в компьютерной графике	236
Традиционная избыточная выборка	237
Практический пример: вычисление изображений с фазовым кодированием	239
Пиксельный шейдер	240
Использование операции PHASE для рисования	241
Пример кода	242
Рисование кривых	243
Пример кода	244

Выборка в изображениях с фазовым кодированием.....	245
Пример кода	247
Более интересное изображение.....	247
Избыточная выборка	248
Пример кода	250
QSS и традиционная выборка методом Монте-Карло	251
Как работает QSS.....	252
Пример кода	255
Добавление цветов.....	257
Итоги.....	259

Глава 12. Алгоритм Шора..... 260

Практический пример: алгоритм Шора на QPU	261
Пример кода	262
Что делает алгоритм Шора.....	263
Для чего вообще нужен QPU?.....	264
Пример кода	264
Квантовое решение.....	266
Шаг за шагом: разложение числа 15 на простые множители	268
Пример кода	268
Шаг 1: инициализация регистров QPU.....	269
Шаг 2: перевод в квантовую суперпозицию	270
Шаг 3: условное умножение на 2	272
Шаг 4: условное умножение на 4	274
Шаг 5: квантовое преобразование Фурье	276
Шаг 6: чтение квантового результата.....	276
Шаг 7: цифровая логика.....	278
Пример кода	279
Шаг 8: проверка результата	281
Важные подробности.....	281
Вычисление остатка	281
Время и память	283
Другие значения переменной <code>coprimes</code>	283

Глава 13. Квантовое машинное обучение	284
Решение систем линейных уравнений.....	285
Описание и решение систем линейных уравнений.....	286
Решение линейных уравнений на QPU.....	288
Квантовый анализ главных компонент.....	298
Традиционный анализ главных компонент	299
PCA с QPU	301
Квантовый метод опорных векторов.....	305
Традиционный метод опорных векторов.....	306
SVM с QPU.....	310
Другие применения машинного обучения.....	315

ЧАСТЬ IV. ПЕРСПЕКТИВЫ

Глава 14. Обзор литературы	318
От круговой записи к комплексным векторам	318
Некоторые нюансы и примечания по терминологии	321
Измерительный базис.....	322
Разложение и компиляция вентиляей.....	324
Телепортация вентиляей.....	326
Зал славы QPU.....	326
Гонка между квантовыми и традиционными компьютерами.....	327
Алгоритмы на базе оракулов	328
Алгоритм Дойча—Джозы	329
Задача Бернштейна—Вазирани	329
Задача Саймона	330
Языки квантового программирования	330
Перспективы квантового моделирования.....	332
Исправление ошибок и устройства NISQ.....	332
Что дальше?	333
Книги	333
Конспекты лекций	334
Сетевые источники информации	334

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.*

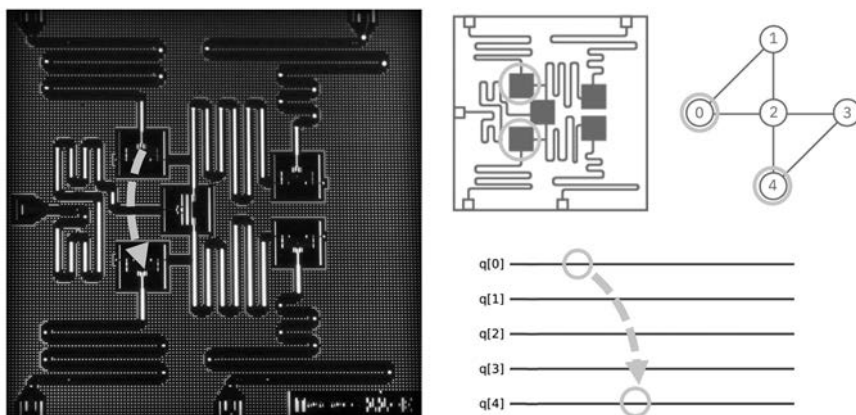


Рис. 4.1. Микросхема IBM очень мала, так что перемещение кубита будет довольно коротким; на иллюстрации и схеме выделены части QPU, между которыми будет происходить телепортация¹

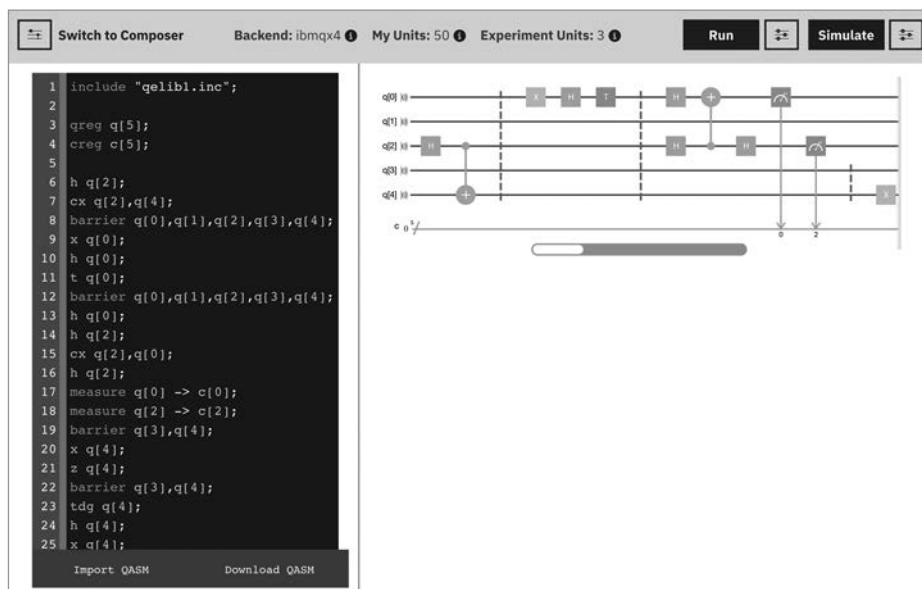


Рис. 4.2. Редактор IBM Q Experience (QX) OpenQASM²

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.

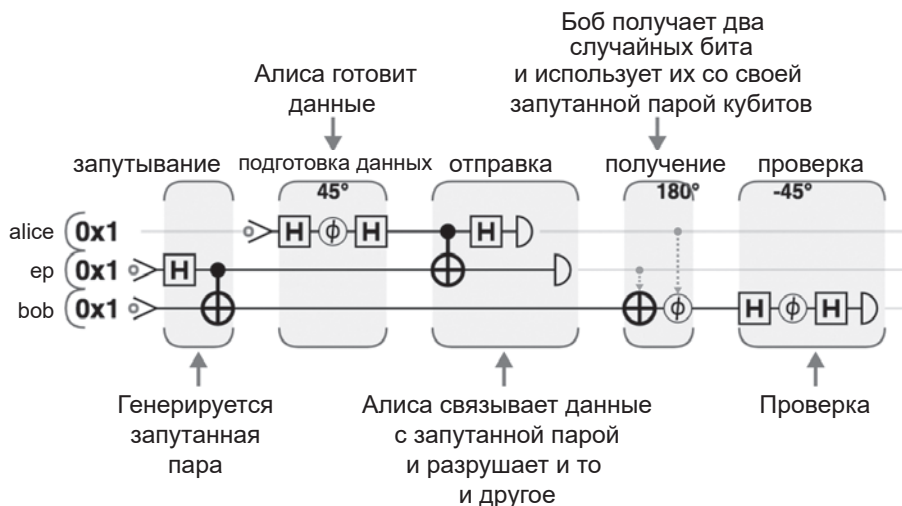


Рис. 4.3. Полная схема телепортации: на стороне Алисы находятся кубиты alice и er, а на стороне Боба — кубит bob

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.

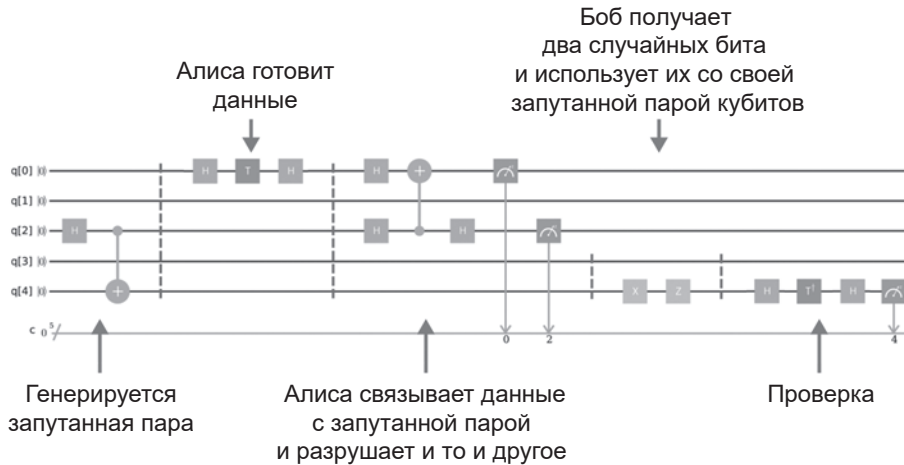


Рис. 4.4. Схема телепортации в IBM QX²

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.

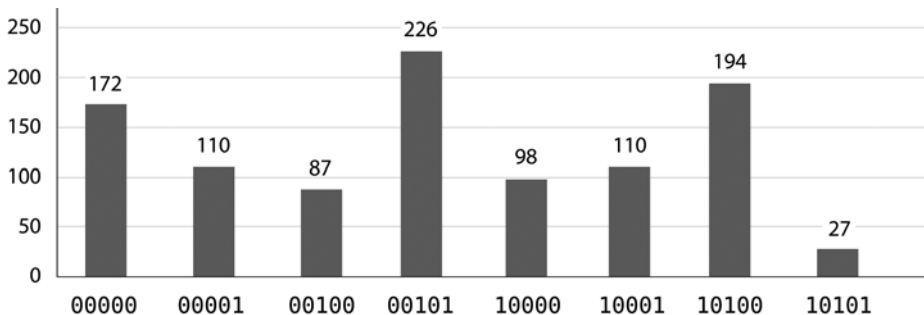


Рис. 4.5. Результаты запуска программы (успешная телепортация?)

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.

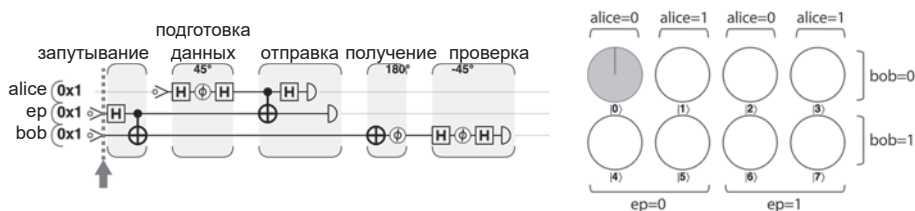


Рис. 4.6. Полная программа телепортации с проверкой



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 $\text{alice}=0$, $\text{ep}=0$ и $\text{bob}=0$.

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

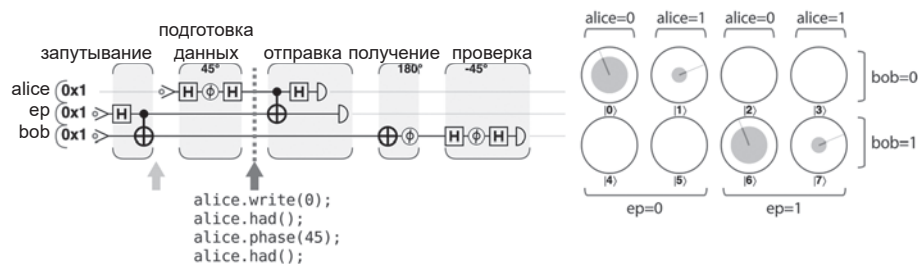


Рис. 4.8. Шаг 2: подготовка данных

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.

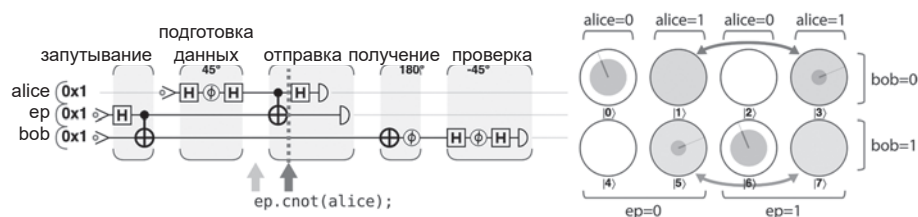


Рис. 4.9. Шаг 3.1: связывание данных с запутанной парой

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.